

---

# AWS IoT

## Developer Guide



## AWS IoT: Developer Guide

Copyright © 2019 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

## Table of Contents

What Is AWS IoT .....	1
AWS IoT Components .....	1
How to Get Started with AWS IoT .....	2
Accessing AWS IoT .....	2
Related Services .....	3
How AWS IoT Works .....	3
Getting Started with AWS IoT .....	5
Sign in to the AWS IoT Console .....	5
Register a Device in the Registry .....	6
Create and Activate a Device Certificate .....	15
Create an AWS IoT Policy .....	17
Attach an AWS IoT Policy to a Device Certificate .....	20
Attach a Certificate to a Thing .....	22
Configure Your Device .....	25
View Device MQTT Messages with the AWS IoT MQTT Client .....	25
Configure and Test Rules .....	27
Create an SNS Topic .....	27
Subscribe to an Amazon SNS Topic .....	29
Create a Rule .....	30
Test the Amazon SNS Rule .....	36
Next Steps .....	37
Create and Track an AWS IoT Job .....	37
Connect Your Device to AWS IoT .....	37
Run the Jobs Sample .....	37
Create a Job Document .....	38
Create a Job .....	38
Execute the Job on a Device .....	45
Tracking Job Progress with Job and Job Execution Events .....	46
AWS IoT Rules Tutorials .....	50
Creating an Amazon DynamoDB Rule .....	50
Testing an Amazon DynamoDB Rule .....	60
Creating an AWS Lambda Rule .....	61
Create a Lambda Function .....	61
Test Your Lambda Function .....	66
Create a Lambda Rule .....	68
Test Your Lambda Rule .....	78
Troubleshooting Lambda Rules .....	79
Creating an Amazon SNS Rule .....	80
AWS IoT SDK Tutorials .....	89
Prerequisites .....	89
Create an AWS IoT Thing for Your Raspberry Pi .....	89
Using the AWS IoT Embedded C SDK .....	102
Set Up the Runtime Environment for the AWS IoT Embedded C SDK .....	102
Sample App Configuration .....	102
Run Sample Applications .....	104
Using the AWS IoT Device SDK for JavaScript .....	105
Set Up the Runtime Environment for the AWS IoT Device SDK for JavaScript .....	105
Install the AWS IoT Device SDK for JavaScript .....	106
Prepare to Run the Sample Applications .....	106
Run the Sample Applications .....	107
AWS IoT Additional Tutorials .....	109
AWS IoT Plant Watering Sample .....	109
Module 1: Setting Up AWS IoT and Sending Data with Your Development Computer .....	109
Module 2: Sending Data with the Raspberry Pi .....	129

Cleaning Up .....	148
Next Steps .....	152
Managing Devices with AWS IoT .....	154
How to Manage Things with the Registry .....	154
Create a Thing .....	154
List Things .....	155
Search for Things .....	155
Update a Thing .....	156
Delete a Thing .....	157
Attach a Principal to a Thing .....	157
Detach a Principal from a Thing .....	157
Thing Types .....	157
Create a Thing Type .....	158
List Thing Types .....	158
Describe a Thing Type .....	158
Associate a Thing Type with a Thing .....	159
Deprecate a Thing Type .....	159
Delete a Thing Type .....	160
Thing Groups .....	160
Create a Thing Group .....	161
Describe a Thing Group .....	162
Add a Thing to a Thing Group .....	163
Remove a Thing from a Thing Group .....	163
List Things in a Thing Group .....	163
List Thing Groups .....	164
List Groups for a Thing .....	165
Update a Thing Group .....	166
Delete a Thing Group .....	166
Attach a Policy to a Thing Group .....	166
Detach a Policy from a Thing Group .....	167
List the Policies Attached to a Thing Group .....	167
List the Groups for a Policy .....	167
Get Effective Policies for a Thing .....	168
Test Authorization for MQTT Actions .....	168
Dynamic Thing Groups .....	169
Create a Dynamic Thing Group .....	170
Describe a Dynamic Thing Group .....	171
Update a Dynamic Thing Group .....	171
Delete a Dynamic Thing Group .....	172
Limitations and Conflicts .....	172
Tagging Your AWS IoT Resources .....	174
Tag Basics .....	174
Tag Restrictions and Limitations .....	175
Using Tags with IAM Policies .....	175
Billing Groups .....	176
Viewing Cost Allocation and Usage Data .....	177
Limits .....	178
Security and Identity .....	179
AWS IoT Authentication .....	179
X.509 Certificates .....	180
IAM Users, Groups, and Roles .....	187
Amazon Cognito Identities .....	187
Custom Authentication .....	187
Custom Authorizers .....	188
Configure a Custom Authorizer .....	189
Custom Authorizer Workflow .....	190
Authorization .....	192

AWS IoT Policies .....	193
IAM IoT Policies .....	220
Authorizing Direct Calls to AWS Services .....	226
How to Use a Certificate to Get a Security Token .....	227
Cross Account Access .....	230
Transport Security .....	231
TLS Cipher Suite Support .....	231
Message Broker .....	233
Protocols .....	233
Protocol/Port Mappings .....	233
MQTT .....	234
HTTP .....	238
MQTT over the WebSocket Protocol .....	239
Rules .....	243
Granting AWS IoT the Required Access .....	243
Pass Role Permissions .....	245
Creating an AWS IoT Rule .....	245
Viewing Your Rules .....	249
Deleting a Rule .....	249
AWS IoT Rule Actions .....	249
CloudWatch Alarm Action .....	250
CloudWatch Metric Action .....	251
DynamoDB Action .....	252
DynamoDBv2 Action .....	253
Elasticsearch Action .....	254
Firehose Action .....	255
IoT Analytics Action .....	256
IoT Events Action .....	257
Kinesis Action .....	258
Lambda Action .....	259
Republish Action .....	260
S3 Action .....	261
Salesforce Action .....	262
SNS Action .....	263
SQS Action .....	263
Step Functions Action .....	264
Troubleshooting a Rule .....	265
Error Handling (Error Action) .....	265
Error Action Message Format .....	266
Error Action Example .....	266
AWS IoT SQL Reference .....	267
Data Types .....	268
Operators .....	271
Functions .....	276
SELECT Clause .....	310
FROM Clause .....	312
WHERE Clause .....	313
Literals .....	313
Case Statements .....	314
JSON Extensions .....	314
Substitution Templates .....	315
SQL Versions .....	316
What's New in the 2016-03-23 SQL Rules Engine Version .....	316
Basic Ingest .....	319
To use Basic Ingest .....	319
Device Shadow Service .....	321
Device Shadow Service Data Flow .....	321

Detecting a Thing Is Connected .....	327
Device Shadow Service Documents .....	328
Document Properties .....	329
Versioning of a Device Shadow .....	329
Client Token .....	330
Example Document .....	330
Empty Sections .....	330
Arrays .....	331
Using Shadows .....	332
Protocol Support .....	332
Updating a Shadow .....	332
Retrieving a Shadow Document .....	333
Deleting Data .....	336
Deleting a Shadow .....	336
Delta State .....	337
Observing State Changes .....	338
Message Order .....	339
Trim Shadow Messages .....	340
RESTful API .....	340
GetThingShadow .....	341
UpdateThingShadow .....	341
DeleteThingShadow .....	342
MQTT Pub/Sub Topics .....	343
/update .....	343
/update/accepted .....	344
/update/documents .....	345
/update/rejected .....	345
/update/delta .....	346
/get .....	347
/get/accepted .....	347
/get/rejected .....	348
/delete .....	348
/delete/accepted .....	349
/delete/rejected .....	349
Document Syntax .....	350
Request State Documents .....	350
Response State Documents .....	350
Error Response Documents .....	351
Error Messages .....	352
Jobs .....	354
Jobs Key Concepts .....	354
Managing Jobs .....	356
Creating and Managing Jobs (Console) .....	357
Creating and Managing Jobs (CLI) .....	358
Devices and Jobs .....	366
Programming Devices to Work with Jobs .....	368
Using the AWS IoT Jobs APIs .....	378
Job Management and Control API .....	378
Jobs Device MQTT and HTTPS APIs .....	434
Job Rollout and Abort Configuration .....	458
Using Job Rollout Rates .....	458
Using Job Rollout Abort Configurations .....	459
Job Limits .....	460
Device Provisioning .....	461
Provisioning Templates .....	461
Parameters Section .....	461
Resources Section .....	462

Template Example .....	466
Programmatic Provisioning .....	467
Just-in-Time Provisioning .....	468
Bulk Provisioning .....	471
Fleet Indexing Service .....	472
Managing Thing Indexing .....	472
Enabling Thing Indexing .....	472
Describing a Thing Index .....	473
Querying a Thing Index .....	474
Restrictions and Limitations .....	475
Authorization .....	476
Managing Thing Group Indexing .....	477
Enabling Thing Group Indexing .....	477
Describing Group Indexes .....	477
Querying a Thing Group Index .....	478
Authorization .....	478
Getting Statistics About Your Device Fleet .....	478
Query Syntax .....	479
Example Thing Queries .....	480
Example Thing Group Queries .....	481
AWS IoT Device Defender .....	483
Audit .....	483
Audit checks .....	483
How To Perform Audits .....	501
Notifications .....	501
Permissions .....	504
Service Limits .....	508
Audit Commands .....	508
Manage Audit Settings .....	508
Schedule Audits .....	512
Run An On-Demand Audit .....	521
Manage Audit Instances .....	522
Check Audit Results .....	528
Detect .....	534
Concepts .....	535
Behaviors .....	536
Metrics .....	537
How To Use AWS IoT Device Defender Detect .....	548
Permissions .....	549
Service Limits .....	550
Sending Metrics from Devices .....	550
Detect Commands .....	555
AttachSecurityProfile .....	555
CreateSecurityProfile .....	556
DeleteSecurityProfile .....	561
DescribeSecurityProfile .....	562
DetachSecurityProfile .....	567
ListActiveViolations .....	568
ListSecurityProfiles .....	573
ListSecurityProfilesForTarget .....	574
ListTargetsForSecurityProfile .....	576
ListViolationEvents .....	577
UpdateSecurityProfile .....	583
ValidateSecurityProfileBehaviors .....	591
Device Agent Integration with AWS IoT Greengrass .....	595
Security Best Practices for Device Agents .....	597
AWS IoT Device Defender Troubleshooting Guide .....	598

Event Messages .....	602
Registry Events .....	603
Jobs Events .....	609
Lifecycle Events .....	612
Connect/Disconnect Events .....	612
Subscribe/Unsubscribe Events .....	613
AWS IoT SDKs .....	615
AWS Mobile SDK for Android .....	615
Arduino Yún SDK .....	615
AWS IoT Device SDK for Embedded C .....	615
AWS IoT C++ Device SDK .....	616
AWS Mobile SDK for iOS .....	616
AWS IoT Device SDK for Java .....	616
AWS IoT Device SDK for JavaScript .....	616
AWS IoT Device SDK for Python .....	617
Monitoring .....	618
Monitoring Tools .....	618
Automated Tools .....	619
Manual Tools .....	619
Monitoring with Amazon CloudWatch .....	619
Metrics and Dimensions .....	620
Using AWS IoT Metrics .....	627
Creating CloudWatch Alarms .....	627
Monitoring with CloudWatch Logs .....	629
Create a Logging Role .....	629
Log Level .....	630
Configure AWS IoT Logging .....	631
CloudWatch Log Entry Format .....	634
Viewing Logs .....	649
Logging AWS IoT API Calls with AWS CloudTrail .....	649
AWS IoT Information in CloudTrail .....	650
Understanding AWS IoT Log File Entries .....	650
Troubleshooting .....	652
Diagnosing Connectivity Issues .....	652
Authentication .....	652
Authorization .....	652
Diagnosing Rules Issues .....	652
Diagnosing Problems with Shadows .....	653
Diagnosing Salesforce Action Issues .....	654
Execution Trace .....	654
Action Success and Failure .....	655
AWS IoT Limits .....	655
AWS IoT Errors .....	655
IOT Commands .....	657
AcceptCertificateTransfer .....	661
AddThingToBillingGroup .....	662
AddThingToThingGroup .....	663
AssociateTargetsWithJob .....	664
AttachPolicy .....	666
AttachPrincipalPolicy .....	667
AttachSecurityProfile .....	668
AttachThingPrincipal .....	669
CancelAuditTask .....	670
CancelCertificateTransfer .....	671
CancelJob .....	672
CancelJobExecution .....	674
ClearDefaultAuthorizer .....	676

CreateAuthorizer .....	677
CreateBillingGroup .....	679
CreateCertificateFromCsr .....	680
CreateDynamicThingGroup .....	682
CreateJob .....	686
CreateKeysAndCertificate .....	691
CreateOTAUpdate .....	692
CreatePolicy .....	698
CreatePolicyVersion .....	699
CreateRoleAlias .....	701
CreateScheduledAudit .....	702
CreateSecurityProfile .....	705
CreateStream .....	709
CreateThing .....	712
CreateThingGroup .....	714
CreateThingType .....	717
CreateTopicRule .....	719
DeleteAccountAuditConfiguration .....	733
DeleteAuthorizer .....	734
DeleteBillingGroup .....	735
DeleteCACertificate .....	736
DeleteCertificate .....	737
DeleteDynamicThingGroup .....	738
DeleteJob .....	739
DeleteJobExecution .....	741
DeleteOTAUpdate .....	743
DeletePolicy .....	744
DeletePolicyVersion .....	745
DeleteRegistrationCode .....	746
DeleteRoleAlias .....	747
DeleteScheduledAudit .....	748
DeleteSecurityProfile .....	749
DeleteStream .....	750
DeleteThing .....	751
DeleteThingGroup .....	752
DeleteThingShadow .....	753
DeleteThingType .....	754
DeleteTopicRule .....	755
DeleteV2LogLevel .....	756
DeprecateThingType .....	757
DescribeAccountAuditConfiguration .....	758
DescribeAuditTask .....	760
DescribeAuthorizer .....	762
DescribeBillingGroup .....	764
DescribeCACertificate .....	766
DescribeCertificate .....	768
DescribeDefaultAuthorizer .....	771
DescribeEndpoint .....	773
DescribeEventConfigurations .....	774
DescribeIndex .....	775
DescribeJob .....	777
DescribeJobExecution .....	782
DescribeJobExecution .....	785
DescribeRoleAlias .....	788
DescribeScheduledAudit .....	790
DescribeSecurityProfile .....	791
DescribeStream .....	796

DescribeThing .....	798
DescribeThingGroup .....	800
DescribeThingRegistrationTask .....	803
DescribeThingType .....	805
DetachPolicy .....	807
DetachPrincipalPolicy .....	808
DetachSecurityProfile .....	809
DetachThingPrincipal .....	810
DisableTopicRule .....	811
EnableTopicRule .....	812
GetEffectivePolicies .....	813
GetIndexingConfiguration .....	814
GetJobDocument .....	816
GetLoggingOptions .....	817
GetOTAUpdate .....	818
GetPendingJobExecutions .....	823
GetPolicy .....	825
GetPolicyVersion .....	827
GetRegistrationCode .....	829
GetStatistics .....	829
GetThingShadow .....	831
GetTopicRule .....	832
GetV2LoggingOptions .....	847
ListActiveViolations .....	848
ListAttachedPolicies .....	853
ListAuditFindings .....	854
ListAuditTasks .....	860
ListAuthorizers .....	862
ListBillingGroups .....	864
ListCACertificates .....	865
ListCertificates .....	867
ListCertificatesByCA .....	869
ListIndices .....	871
ListJobExecutionsForJob .....	872
ListJobExecutionsForThing .....	875
ListJobs .....	877
ListOTAUpdates .....	880
ListOutgoingCertificates .....	882
ListPolicies .....	883
ListPolicyPrincipals .....	885
ListPolicyVersions .....	887
ListPrincipalPolicies .....	888
ListPrincipalThings .....	890
ListRoleAliases .....	891
ListScheduledAudits .....	893
ListSecurityProfiles .....	894
ListSecurityProfilesForTarget .....	896
ListStreams .....	898
ListTagsForResource .....	899
ListTargetsForPolicy .....	900
ListTargetsForSecurityProfile .....	902
ListThingGroups .....	903
ListThingGroupsForThing .....	905
ListThingPrincipals .....	907
ListThingRegistrationTaskReports .....	908
ListThingRegistrationTasks .....	909
ListThingTypes .....	911

ListThings .....	913
ListThingsInBillingGroup .....	915
ListThingsInThingGroup .....	916
ListTopicRules .....	918
ListV2LoggingLevels .....	919
ListViolationEvents .....	921
Publish .....	926
RegisterCACertificate .....	927
RegisterCertificate .....	929
RegisterThing .....	931
RejectCertificateTransfer .....	932
RemoveThingFromBillingGroup .....	933
RemoveThingFromThingGroup .....	934
ReplaceTopicRule .....	935
SearchIndex .....	950
SetDefaultAuthorizer .....	953
SetDefaultPolicyVersion .....	955
SetLoggingOptions .....	956
SetV2LogLevel .....	957
SetV2LoggingOptions .....	958
StartNextPendingJobExecution .....	958
StartOnDemandAuditTask .....	961
StartThingRegistrationTask .....	963
StopThingRegistrationTask .....	964
TagResource .....	965
TestAuthorization .....	966
TestInvokeAuthorizer .....	970
TransferCertificate .....	972
UntagResource .....	973
UpdateAccountAuditConfiguration .....	974
UpdateAuthorizer .....	976
UpdateBillingGroup .....	978
UpdateCACertificate .....	979
UpdateCertificate .....	981
UpdateDynamicThingGroup .....	982
UpdateEventConfigurations .....	985
UpdateIndexingConfiguration .....	986
UpdateJob .....	987
UpdateJobExecution .....	990
UpdateRoleAlias .....	994
UpdateScheduledAudit .....	995
UpdateSecurityProfile .....	997
UpdateStream .....	1005
UpdateThing .....	1007
UpdateThingGroup .....	1009
UpdateThingGroupsForThing .....	1011
UpdateThingShadow .....	1012
ValidateSecurityProfileBehaviors .....	1014

# What Is AWS IoT?

AWS IoT provides secure, bi-directional communication between Internet-connected devices such as sensors, actuators, embedded micro-controllers, or smart appliances and the AWS Cloud. This enables you to collect telemetry data from multiple devices, and store and analyze the data. You can also create applications that enable your users to control these devices from their phones or tablets.

## AWS IoT Components

AWS IoT consists of the following components:

### **Device gateway**

Enables devices to securely and efficiently communicate with AWS IoT.

### **Message broker**

Provides a secure mechanism for devices and AWS IoT applications to publish and receive messages from each other. You can use either the MQTT protocol directly or MQTT over WebSocket to publish and subscribe. You can use the HTTP REST interface to publish.

### **Rules engine**

Provides message processing and integration with other AWS services. You can use an SQL-based language to select data from message payloads, and then process and send the data to other services, such as Amazon S3, Amazon DynamoDB, and AWS Lambda. You can also use the message broker to republish messages to other subscribers.

### **Security and Identity service**

Provides shared responsibility for security in the AWS Cloud. Your devices must keep their credentials safe in order to securely send data to the message broker. The message broker and rules engine use AWS security features to send data securely to devices or other AWS services.

### **Registry**

Organizes the resources associated with each device in the AWS Cloud. You register your devices and associate up to three custom attributes with each one. You can also associate certificates and MQTT client IDs with each device to improve your ability to manage and troubleshoot them.

### **Group registry**

Groups allow you to manage several devices at once by categorizing them into groups. Groups can also contain groups—you can build a hierarchy of groups. Any action you perform on a parent group will apply to its child groups, and to all the devices in it and in all of its child groups as well. Permissions given to a group will apply to all devices in the group and in all of its child groups.

### **Device shadow**

A JSON document used to store and retrieve current state information for a device.

### **Device Shadow service**

Provides persistent representations of your devices in the AWS Cloud. You can publish updated state information to a device's shadow, and your device can synchronize its state when it connects. Your devices can also publish their current state to a shadow for use by applications or other devices.

### Device Provisioning service

Allows you to provision devices using a template that describes the resources required for your device: a *thing*, a certificate, and one or more policies. A thing is an entry in the registry that contains attributes that describe a device. Devices use certificates to authenticate with AWS IoT. Policies determine which operations a device can perform in AWS IoT.

The templates contain variables that are replaced by values in a dictionary (map). You can use the same template to provision multiple devices just by passing in different values for the template variables in the dictionary.

### Custom Authentication service

You can define custom authorizers that allow you to manage your own authentication and authorization strategy using a custom authentication service and a Lambda function. Custom authorizers allow AWS IoT to authenticate your devices and authorize operations using bearer token authentication and authorization strategies.

Custom authorizers can implement various authentication strategies (for example, JSON Web Token verification, OAuth provider callout, and so on) and must return policy documents that are used by the device gateway to authorize MQTT operations.

### Jobs service

Allows you to define a set of remote operations that are sent to and executed on one or more devices connected to AWS IoT. For example, you can define a job that instructs a set of devices to download and install application or firmware updates, reboot, rotate certificates, or perform remote troubleshooting operations.

To create a job, you specify a description of the remote operations to be performed and a list of targets that should perform them. The targets can be individual devices, groups or both.

For information about AWS IoT limits, see [AWS IoT Limits](#).

## How to Get Started with AWS IoT

- To learn more about AWS IoT, see [How AWS IoT Works \(p. 3\)](#).
- To learn how to connect a device to AWS IoT, see [Getting Started with AWS IoT \(p. 5\)](#).

## Accessing AWS IoT

AWS IoT provides the following interfaces to create and interact with your devices:

- **AWS Command Line Interface (AWS CLI)**—Run commands for AWS IoT on Windows, macOS, and Linux. These commands allow you to create and manage things, certificates, rules, and policies. To get started, see the [AWS Command Line Interface User Guide](#). For more information about the commands for AWS IoT, see [iot](#) in the [AWS CLI Command Reference](#).
- **AWS IoT API**—Build your IoT applications using HTTP or HTTPS requests. These API actions allow you to programmatically create and manage things, certificates, rules, and policies. For more information about the API actions for AWS IoT, see [Actions](#) in the [AWS IoT API Reference](#).
- **AWS SDKs**—Build your IoT applications using language-specific APIs. These SDKs wrap the HTTP/HTTPS API and allow you to program in any of the supported languages. For more information, see [AWS SDKs and Tools](#).
- **AWS IoT Device SDKs**—Build applications that run on devices that send messages to and receive messages from AWS IoT. For more information see, [AWS IoT SDKs](#).

# Related Services

AWS IoT integrates directly with the following AWS services:

- **Amazon Simple Storage Service**—Provides scalable storage in the AWS Cloud. For more information, see [Amazon S3](#).
- **Amazon DynamoDB**—Provides managed NoSQL databases. For more information, see [Amazon DynamoDB](#).
- **Amazon Kinesis**—Enables real-time processing of streaming data at a massive scale. For more information, see [Amazon Kinesis](#).
- **AWS Lambda**—Runs your code on virtual servers from Amazon EC2 in response to events. For more information, see [AWS Lambda](#).
- **Amazon Simple Notification Service**—Sends or receives notifications. For more information, see [Amazon SNS](#).
- **Amazon Simple Queue Service**—Stores data in a queue to be retrieved by applications. For more information, see [Amazon SQS](#).

# How AWS IoT Works

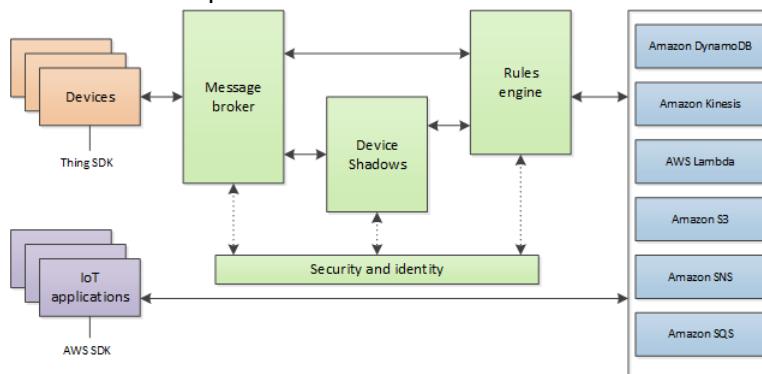
AWS IoT enables Internet-connected devices to connect to the AWS Cloud and lets applications in the cloud interact with Internet-connected devices. Common IoT applications either collect and process telemetry from devices or enable users to control a device remotely.

Devices report their state by publishing messages, in JSON format, on MQTT topics. Each MQTT topic has a hierarchical name that identifies the device whose state is being updated. When a message is published on an MQTT topic, the message is sent to the AWS IoT MQTT message broker, which is responsible for sending all messages published on an MQTT topic to all clients subscribed to that topic.

Communication between a device and AWS IoT is protected through the use of X.509 certificates. AWS IoT can generate a certificate for you or you can use your own. In either case, the certificate must be registered and activated with AWS IoT, and then copied onto your device. When your device communicates with AWS IoT, it presents the certificate to AWS IoT as a credential.

We recommend that all devices that connect to AWS IoT have an entry in the registry. The registry stores information about a device and the certificates that are used by the device to secure communication with AWS IoT.

You can create rules that define one or more actions to perform based on the data in a message. For example, you can insert, update, or query a DynamoDB table or invoke a Lambda function. Rules use expressions to filter messages. When a rule matches a message, the rules engine triggers the action using the selected properties. Rules also contain an IAM role that grants AWS IoT permission to the AWS resources used to perform the action.



Each device has a shadow that stores and retrieves state information. Each item in the state information has two entries: the state last reported by the device and the desired state requested by an application. An application can request the current state information for a device. The shadow responds to the request by providing a JSON document with the state information (both reported and desired), metadata, and a version number. An application can control a device by requesting a change in its state. The shadow accepts the state change request, updates its state information, and sends a message to indicate the state information has been updated. The device receives the message, changes its state, and then reports its new state.

# Getting Started with AWS IoT

This tutorial shows you how to create resources required to send, receive, and process MQTT messages from devices using AWS IoT. You will be using an MQTT client to emulate an IoT device.

For more information about AWS IoT, see [What Is AWS IoT \(p. 1\)](#).

## Topics

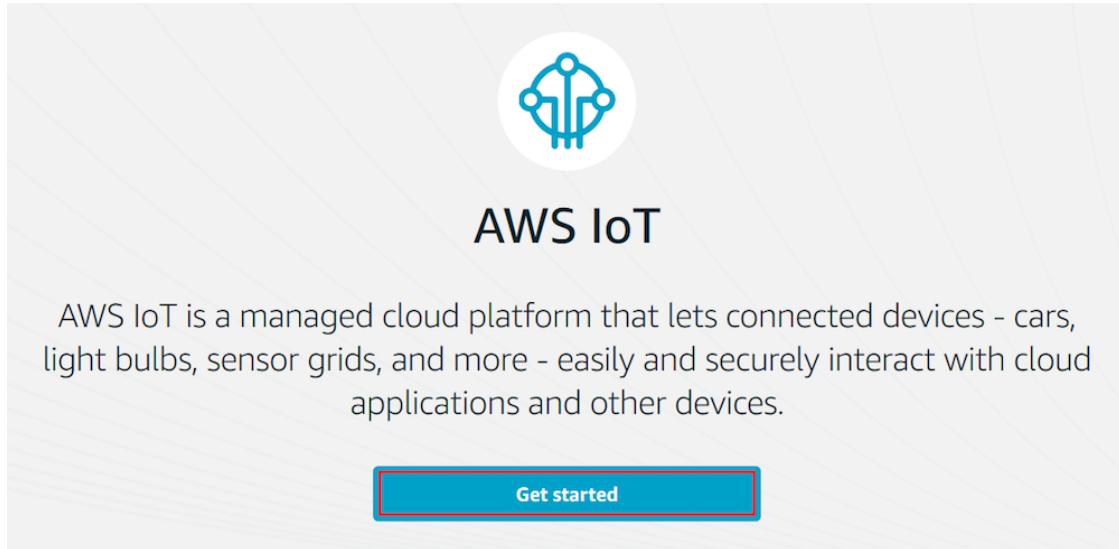
- [Sign in to the AWS IoT Console \(p. 5\)](#)
- [Register a Device in the Registry \(p. 6\)](#)
- [Configure Your Device \(p. 25\)](#)
- [View Device MQTT Messages with the AWS IoT MQTT Client \(p. 25\)](#)
- [Configure and Test Rules \(p. 27\)](#)
- [Create and Track an AWS IoT Job \(p. 37\)](#)

## Sign in to the AWS IoT Console

If you do not have an AWS account, create one.

### To create an AWS account:

1. Open the [AWS home page](#) and choose **Create an AWS Account**.
2. Follow the online instructions. Part of the sign-up procedure involves receiving a phone call and entering a PIN using your phone's keypad.
3. Sign in to the AWS Management Console and open the [AWS IoT console](#).
4. On the **Welcome** page, choose **Get started**.



If this is your first time using the AWS IoT console, you see the **Welcome to the AWS IoT Console** page.

# Register a Device in the Registry

Devices connected to AWS IoT are represented by *IoT things* in the AWS IoT registry. The registry allows you to keep a record of all of the devices that are registered to your AWS IoT account.

## To register your device in the registry:

1. On the **Welcome to the AWS IoT Console** page, in the navigation pane, choose **Manage**.

The screenshot shows the AWS IoT Console's 'Welcome to the AWS IoT Console' page. The left sidebar has a 'Manage' button highlighted with a red box. The main area contains three cards: 'See how AWS IoT works' (with a 'Start the tutorial' button), 'Connect to AWS IoT' (with a 'View connection options' button), and 'Explore documentation' (with a 'Go to documentation' button). The top right corner has a bell icon and a checkmark icon.

2. On the **You don't have any things yet** page, choose **Register a thing**.

The screenshot shows the 'You don't have any things yet' page. The left sidebar has 'Manage' and 'Things' selected. The main area features a central illustration of a car, a windmill, and a thermometer, with the text 'You don't have any things yet' and 'A thing is the representation of a device in the cloud.' Below the illustration are 'Learn more' and 'Register a thing' buttons, with 'Register a thing' highlighted with a red box.

3. On the **Creating AWS IoT things** page, choose **Create a single thing**.

**Creating AWS IoT things**

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. [Learn more.](#)

**Register a single AWS IoT thing**  
Create a thing in your registry

**Create a single thing**

---

**Bulk register many AWS IoT things**  
Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

**Create many things**

---

**Create a single thing**

4. On the **Create a thing** page, in the **Name** field, type a name for your thing, such as **MyIoTThing**. Choose **Next**.

**Note**

We do not recommended using personally identifiable information in your thing name.

CREATE A THING

## Add your device to the thing registry

STEP  
1/3

This step creates an entry in the thing registry and a thing shadow for your device.

Name

Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

Create a type

Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups /

Create group Change

Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Value

Clear

Add another

Show thing shadow ▾

Cancel

Back

Next

5. On the **Add a certificate for your thing** page, choose **Create certificate**. This generates an X.509 certificate and key pair.

CREATE A THING

## Add a certificate for your thing

A certificate is used to authenticate your device's connection to AWS IoT.

### One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

[Create certificate](#)

### Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

[Create with CSR](#)

### Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

[Get started](#)

### Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

[Create thing without certificate](#)

6. On the **Certificate created!** page, download your public and private keys, certificate, and root certificate authority (CA):

- a. Choose **Download** for your certificate.
- b. Choose **Download** for your private key.
- c. Choose **Download** for the Amazon root CA. This will display a new web page. Choose **RSA 2048 bit key: Amazon Root CA 1**. This opens another web page with the text of the root CA certificate. Copy this text and paste it into a file named `Amazon_Root_CA_1.pem`.

Most web browsers save downloaded files into a Downloads directory. You will copy these files to a different directory when you run the sample applications. Choose **Activate** to activate the X.509 certificate, and then choose **Attach a policy**.

## Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	0488f55a11.cert.pem	<a href="#">Download</a>
A public key	0488f55a11.public.key	<a href="#">Download</a>
A private key	0488f55a11.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

[Activate](#)

[Cancel](#)

[Done](#)

[Attach a policy](#)

7. On the **Add a policy for your thing** page, choose **Register Thing**.

After you register your thing, you will need to create and attach a new policy to the certificate.

CREATE A THING

### Add a policy for your thing

STEP 3/3

Select a policy to attach to this certificate:

No match found  
There are no policies in your account.

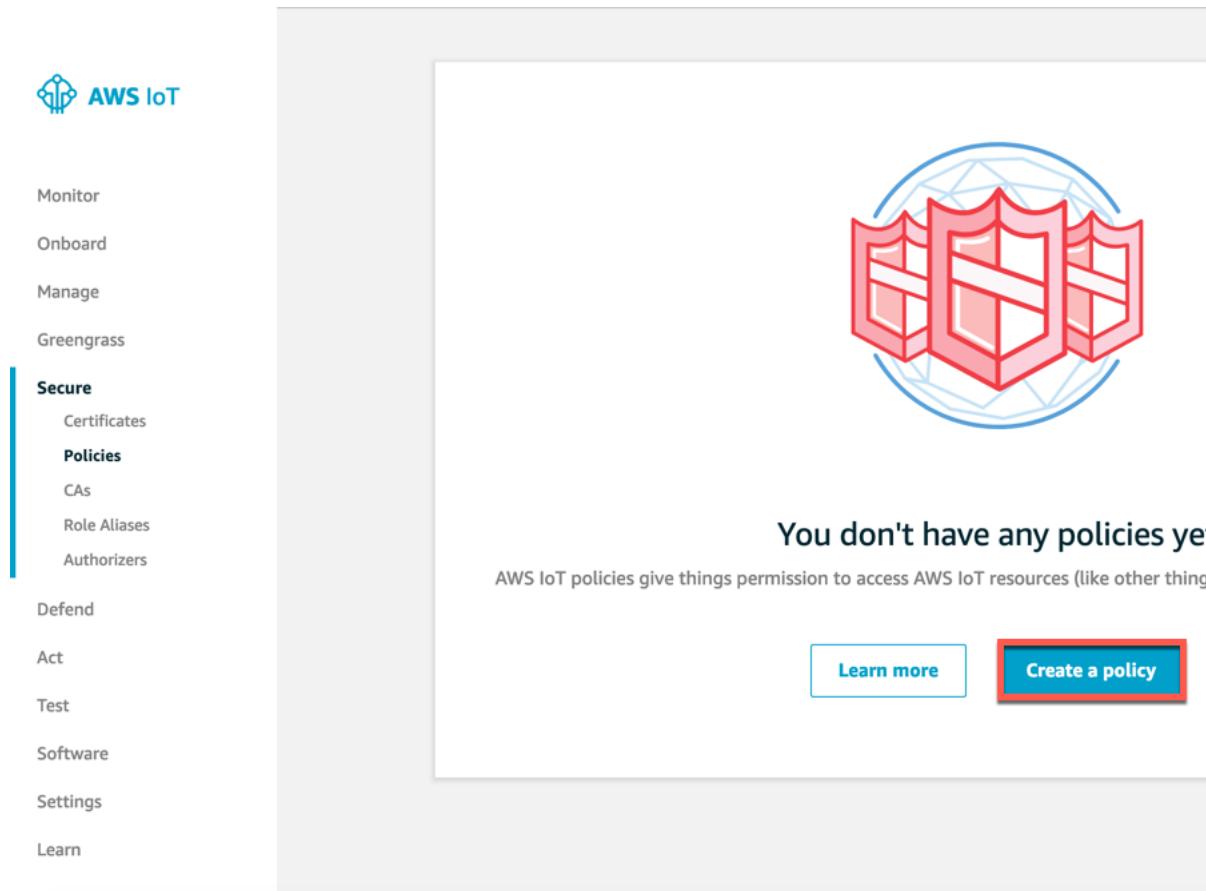
[View more](#)

0 policies selected

[Register Thing](#)

8. On the AWS IoT console, in the navigation pane, choose **Secure** and **Policies**.

Choose **Create**.



9. On the **Create a policy** page:

- a. Enter a **Name** for the policy, such as **MyIoTPolicy**.
- b. For **Action**, enter **iot:\***. For **Resource ARN**, enter **\***.
- c. Under **Effect**, choose **Allow**, and then choose **Create**.

This policy allows your device to perform all AWS IoT actions on all AWS IoT resources.

**Important**

These settings are overly permissive. In a production environment narrow the scope of the permissions to that which are required by your device. For more information, see [Authorization \(p. 192\)](#).

## Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name

MyIotPolicy

### Add statements

Policy statements define the types of actions that can be performed by a resource.

Advanced

Action

iot:\*

Resource ARN

\*

Effect

Allow  Deny

Remove

Add statement

Create

10. Choose **Manage** and then choose your AWS IoT thing.

THING  
**MyIotThing**  
NO TYPE

**Actions**

**Details**

**Security**  

Thing Groups  
Billing Groups  
Shadow  
Interact  
Activity  
Jobs  
Violations

**Thing ARN**

A thing Amazon Resource Name uniquely identifies this thing.  
arn:aws:iot:us-west-2:...:thing/MyIotThing

**Type**

Q   No type ...

11. Choose **Security**

THING  
**MyIotThing**  
NO TYPE

**Actions**

**Details**

**Security**  

Thing Groups  
Billing Groups  
Shadow  
Interact  
Activity  
Jobs  
Violations

**Certificates**

Create certificate View other options

c290b87d4ece080f3a...

12. Choose your certificate.

13. In the certificate detail page, choose **Actions** and then **Attach policy**.

The screenshot shows the AWS IoT Certificate Details page for a certificate with ARN: arn:aws:iot:us-west-2:030714055129:cert/c290b87d4ece080f3a6d3085. The 'Actions' menu is open, and the 'Attach policy' option is highlighted with a red box.

**CERTIFICATE**  
**c290b87d4ece080f3a6d3085f9971e939b300c20d8503766a92aecc4bcdf75f7**  
**ACTIVE**

**Details**      Certificate ARN  
arn:aws:iot:us-west-2:030714055129:cert/c290b87d4ece080f3a6d3085

Policies      A certificate Amazon Resource Name (ARN) uniquely identifies this certificate. [Learn more](#)

Things

Non-compliance

**Details**

**Issuer**  
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

**Subject**  
CN=AWS IoT Certificate

**Create date**  
Feb 18, 2019 2:41:59 PM -0800

**Effective date**  
Feb 18, 2019 2:39:59 PM -0800

**Expiration date**  
Dec 31, 2049 3:59:59 PM -0800

**Actions**

- Activate
- Deactivate
- Revoke
- Accept transfer
- Reject transfer
- Revoke transfer
- Start transfer
- Attach policy**
- Attach thing
- Download
- Delete

14. Choose the policy you created (MyIotPolicy) and choose **Attach**.

The screenshot shows the 'Attach policies to certificate(s)' dialog box. It lists the selected policy 'MyIotPolicy' and provides options to search for other policies or cancel the operation.

**Attach policies to certificate(s)**

Policies will be attached to the following certificate(s):  
**c290b87d4ece080f3a6d3085f9971e939b300c20d8503766a92aecc4bcdf75f7**

Choose one or more policies

Search policies

**MyIotPolicy** [View](#)

1 policy selected [Cancel](#) **Attach**

# Create and Activate a Device Certificate

Communication between your device and AWS IoT is protected through the use of X.509 certificates. AWS IoT can generate a certificate for you or you can use your own X.509 certificate. In this tutorial, AWS IoT generates the X.509 certificate for you. Certificates must be activated prior to use.

1. Choose **Create certificate**.



A certificate is used to authenticate your device's connection to AWS IoT.

## One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

**Create certificate**

## Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

**Create with CSR**

## Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

**Get started**

## Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

**Create thing without certificate**

2. On the **Certificate created!** page, choose **Download** for the certificate, private key, and the root CA for AWS IoT (the public key need not be downloaded). Save each of them to your computer, and then choose **Activate** to continue.

### Note

The root CA for AWS IoT **Download** link takes you to the [X.509 Certificates and AWS IoT \(p. 180\)](#) page where you choose a CA certificate. Unlike the other **Download** links on the page, it doesn't directly download a file.

## Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	09eb9ae91d.cert.pem	<a href="#">Download</a>
A public key	09eb9ae91d.public.key	<a href="#">Download</a>
A private key	09eb9ae91d.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

[Activate](#)

[Cancel](#)

[Done](#)

[Attach a policy](#)

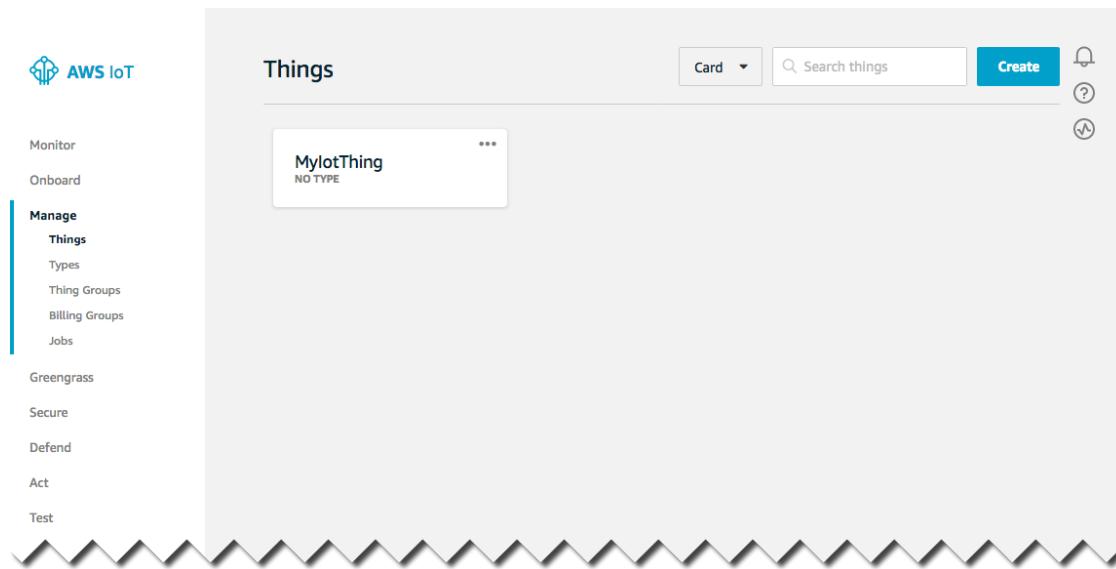
Be aware that the downloaded filenames may be different than those listed on the **Certificate created!** page. For example:

- 2a540e2346-certificate.pem.crt
- 2a540e2346-private.pem.key
- 2a540e2346-public.pem.key

### Note

Although it is unlikely, root CA certificates are subject to expiration and/or revocation. If this should occur, you must copy new a root CA certificate onto your device.

3. Choose **Done** to return to the AWS IoT console main page.



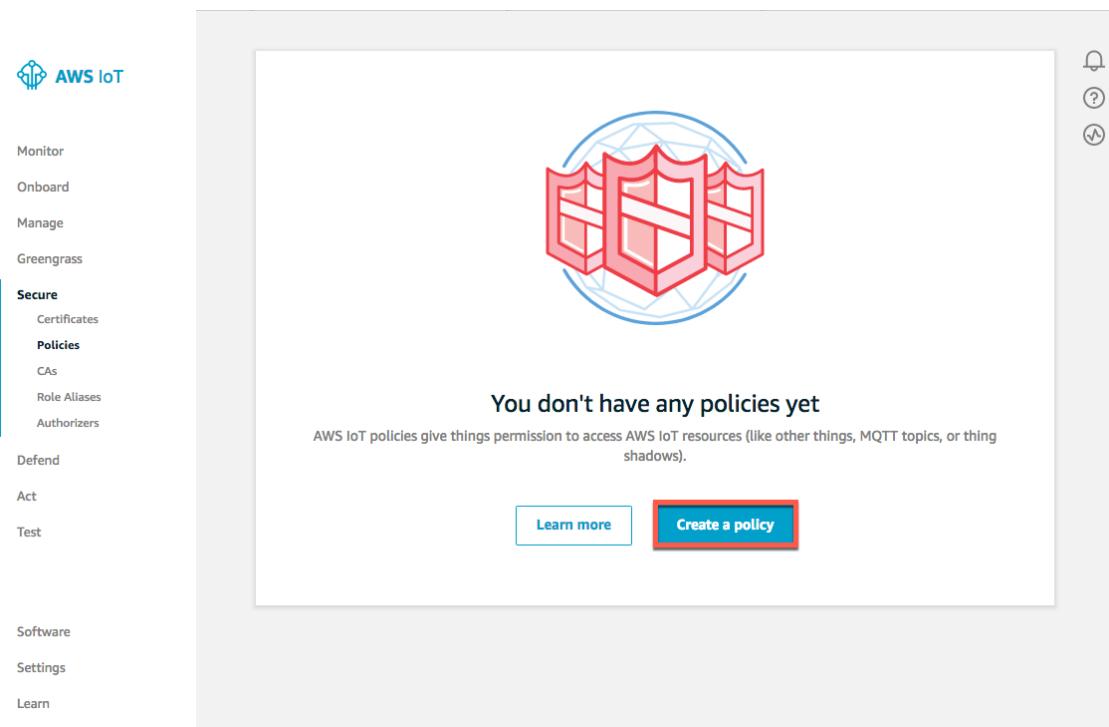
When working with a device you will need to copy the private key and rootCA certificate onto your device. This guide assumes you are not using a device and are trying out AWS IoT using the console.

## Create an AWS IoT Policy

X.509 certificates are used to authenticate your device with AWS IoT. AWS IoT policies are used to authorize your device to perform AWS IoT operations, such as subscribing or publishing to MQTT topics. Your device presents its certificate when sending messages to AWS IoT. To allow your device to perform AWS IoT operations, you must create an AWS IoT policy and attach it to your device certificate.

### To create an AWS IoT policy:

1. In the left navigation pane, choose **Secure**, and then **Policies**. On the **You don't have a policy yet** page, choose **Create a policy**.



2. On the **Create a policy** page, in the **Name** field, type a name for the policy (for example, **MyIoTPolicy**).

**Note**

We do not recommended using personally identifiable information in your policy names.

In the **Action** field, type **iot:Connect**. In the **Resource ARN** field, type **\***. Select the **Allow** check box. This allows all clients to connect to AWS IoT.

## Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name  
MyIotPolicy

Add statements

Policy statements define the types of actions that can be performed by a resource. [Advanced mode](#)

Action  
iot:Connect

Resource ARN  
\*

Effect  
 Allow  Deny

Remove

Add statement



### Note

You can restrict which clients (devices) are able to connect by specifying a client ARN as the resource. The client ARNs follow this format:

`arn:aws:iot:<your-region>:<your-aws-account>:client/<my-client-id>`

Select the **Add Statement** button to add another policy statement. In the **Action** field, type **iot:Publish**. In the **Resource ARN** field, type the ARN of the topic to which your device will publish.

### Note

The topic ARN follows this format:

`arn:aws:iot:<your-region>:<your-aws-account>:topic/<your/topic>`

For example:

`arn:aws:iot:us-east-1:123456789012:topic/my/topic`

Finally, select the **Allow** check box. This allows your device to publish messages to the specified topic.

3. After you have entered the information for your policy, choose **Create**.

The screenshot shows the AWS IoT Policy creation interface. At the top, there's a decorative wavy line graphic. Below it, a section titled "Add statements" contains a note: "Policy statements define the types of actions that can be performed by a resource." To the right is a link to "Advanced mode".

The main area displays two policy statements:

- Statement 1:** Action: iot:Connect; Resource ARN: \*; Effect:  Allow  Deny. A "Remove" button is to the right.
- Statement 2:** Action: iot:Publish; Resource ARN: arn:aws:iot:us-west-2:123456789012:topic/mytopic; Effect:  Allow  Deny. A "Remove" button is to the right.

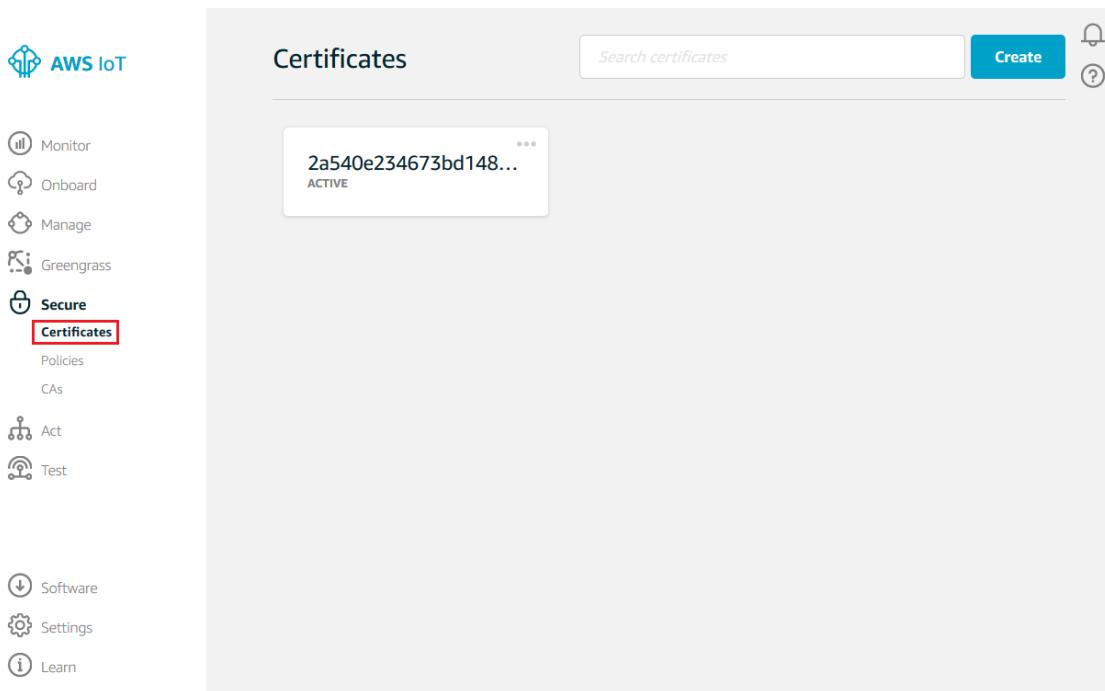
At the bottom left is a "Add statement" button, and at the bottom right is a prominent blue "Create" button.

For more information, see [Managing AWS IoT Policies](#).

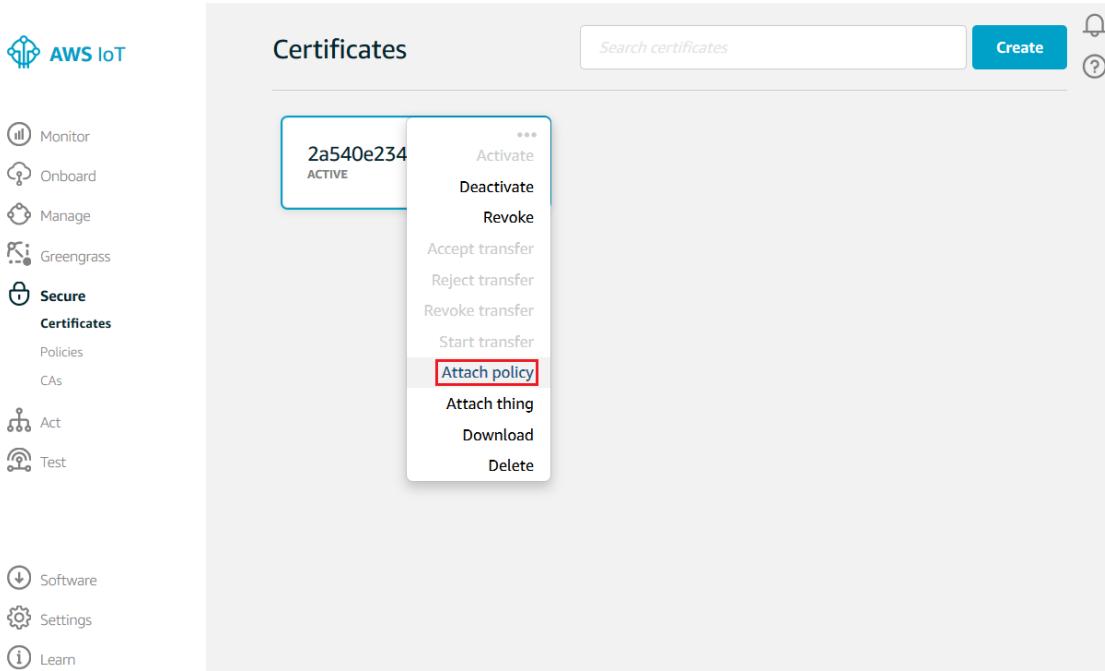
## Attach an AWS IoT Policy to a Device Certificate

Now that you have created a policy, you must attach it to your device certificate. Attaching an AWS IoT policy to a certificate gives the device the permissions specified in the policy.

1. In the left navigation pane, choose **Secure**, and then **Certificates**.



2. In the box for the certificate you created, choose ... to open a drop-down menu, and then choose **Attach policy**.



3. In the **Attach policies to certificate(s)** dialog box, select the check box next to the policy you created in the previous step, and then choose **Attach**.

## Attach policies to certificate(s)

Policies will be attached to the following certificate(s):  
**09eb9ae91d6f9bdf423d4b491aa50fdbd925c2fefb56f63dab8435ecfc08b18a**

Choose one or more policies

Search policies

<input checked="" type="checkbox"/> MylotPolicy	<a href="#">View</a>
---	----------------------

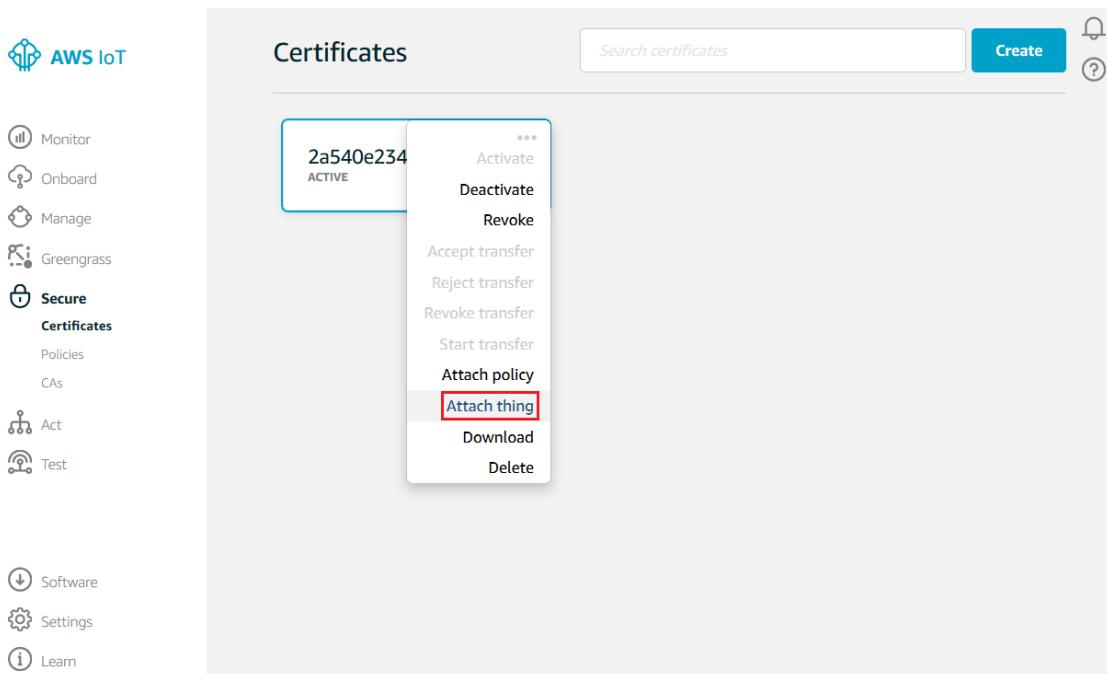
1 policy selected [Cancel](#) [Attach](#)

## Attach a Certificate to a Thing

A device must have a certificate, private key and root CA certificate to authenticate with AWS IoT. We recommend that you also attach the device certificate to the IoT thing that represents your device in AWS IoT. This allows you to create AWS IoT policies that grant permissions based on certificates attached to your things. For more information. see [Thing Policy Variables \(p. 201\)](#)

### To attach a certificate to the thing representing your device in the registry:

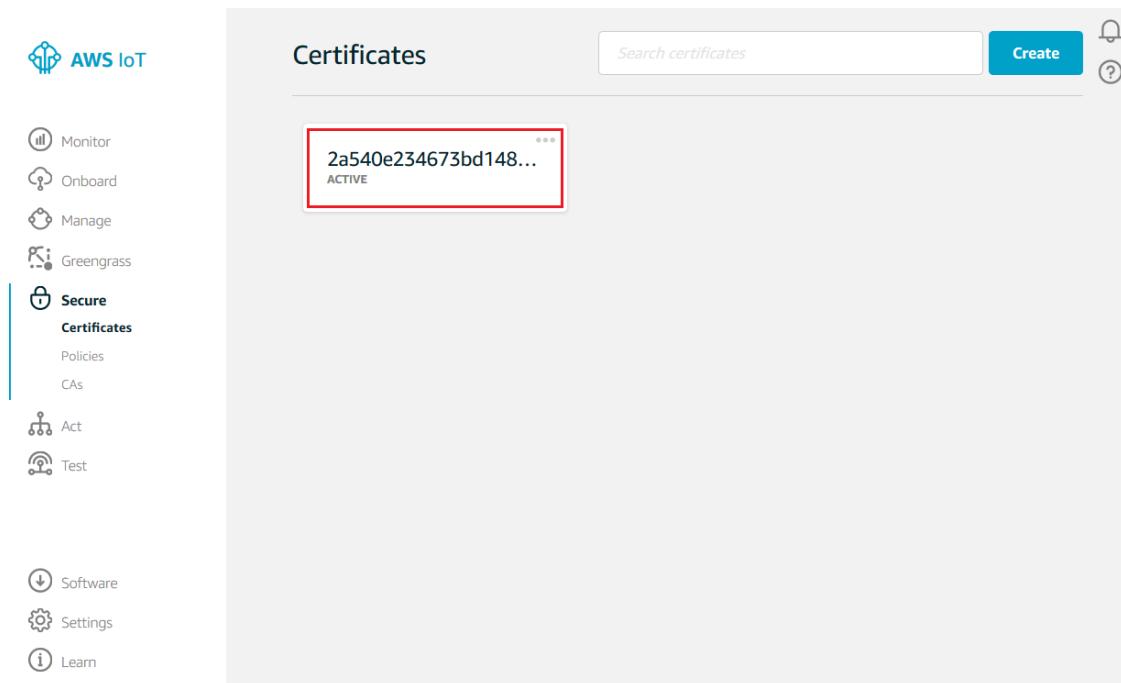
1. In the box for the certificate you created, choose ... to open a drop-down menu, and then choose **Attach thing**.



2. In the **Attach things to certificate(s)** dialog box, select the check box next to the thing you registered, and then choose **Attach**.

The screenshot shows the "Attach things to certificate(s)" dialog box. It displays the message: "Things will be attached to the following certificate(s):" followed by the certificate ID "09eb9ae91d6f9bdf423d4b491aa50fdbd925c2fefb56f63dab8435ecfc08b18a". Below this, it says "Choose one or more things" and shows a search bar with "Search things" placeholder text. A list of things is shown, with "MylotThing" selected (indicated by a checked checkbox). At the bottom, it says "1 thing selected" and has "Cancel" and "Attach" buttons, with "Attach" being highlighted with a red box.

3. To verify the thing is attached, select the box representing the certificate.



4. On the **Details** page for the certificate, in the left navigation pane, choose **Things**.

The screenshot shows the "Details" page for a certificate. The top header includes the word "CERTIFICATE" and the certificate ID "09eb9ae91d6f9bdf423d4b491aa50fdbd925c2fefb56f63dab8435ecfc08b18a". Below this, the status is listed as "ACTIVE". On the right, there is an "Actions" dropdown menu. The left navigation pane has tabs for "Details", "Policies", "Things" (which is selected and highlighted in blue), and "Non-compliance". The main content area is titled "Things" and lists a single item: "MylotThing".

5. To verify the policy is attached, on the **Details** page for the certificate, in the left navigation pane, choose **Policies**.

The screenshot shows the "Details" page for a certificate, identical to the previous one but with the "Policies" tab selected in the left navigation pane instead of "Things". The main content area is titled "Policies" and lists a single item: "MylotPolicy".

# Configure Your Device

All devices must have a device certificate, private key, and root CA certificate installed in order to communicate with AWS IoT. Consult your device's documentation to connect to it and copy your device certificate, private key, and root CA certificate onto your device.

If you don't have an IoT-ready device, you can use the MQTT client, the AWS IoT Device SDKs or the AWS CLI. For more information, see the [AWS IoT SDK Tutorials \(p. 89\)](#) section. The tutorials there use a Raspberry Pi but can easily be adapted for use with other types of computers.

# View Device MQTT Messages with the AWS IoT MQTT Client

You can use the AWS IoT MQTT client to better understand the MQTT messages sent by a device.

Devices publish MQTT messages on topics. You can use the AWS IoT MQTT client to subscribe to these topics to see these messages.

## To view MQTT messages:

1. In the [AWS IoT console](#), in the left navigation pane, choose **Test**.



2. Subscribe to the topic on which your IoT thing publishes. Continuing with this example, in **Subscribe to a topic**, in the **Subscription topic** field, type `my/topic`, and then choose **Subscribe to topic**.

## AWS IoT Developer Guide

### View Device MQTT Messages with the AWS IoT MQTT Client

The screenshot shows the AWS IoT MQTT client interface. On the left, a sidebar lists navigation options: Monitor, Onboard, Manage, Greengrass, Secure, Defend, Act, and Test. The 'Test' option is selected and highlighted in blue. The main area is titled 'MQTT client' and shows the 'Connected as iotconsole-1549405147548-4' status. A 'Subscriptions' tab is active, displaying two buttons: 'Subscribe to a topic' and 'Publish to a topic'. Below these buttons, there is a 'Subscription topic' input field containing 'my/topic', a 'Subscribe to topic' button, and a 'Max message capture' input field set to '100'. Under 'Quality of Service', the '0 - This client will not acknowledge to the Device Gateway that messages are received' radio button is selected. In the 'MQTT payload display' section, the 'Auto-format JSON payloads (improves readability)' radio button is selected. The interface has a light gray background with decorative wavy patterns at the bottom.

Choosing **Subscribe to topic** above, results in the topic **my/topic** appearing in the **Subscriptions** column.

The screenshot shows the AWS IoT MQTT client interface. The sidebar and top navigation are identical to the previous screenshot. The 'Subscriptions' tab is still active. Now, the 'Publish' section is visible, featuring a 'Specify a topic and a message to publish...' input field with 'my/topic' entered, and a 'Publish to topic' button. Below this, a dark text area displays a JSON message payload:  
1 {  
2 "message": "Hello from AWS IoT console"  
3 }

#### To emulate an IoT thing sending a message

- On the MQTT client page, in the **Publish** section, in the **Specify a topic and a message to publish...** field, type **my/topic**.

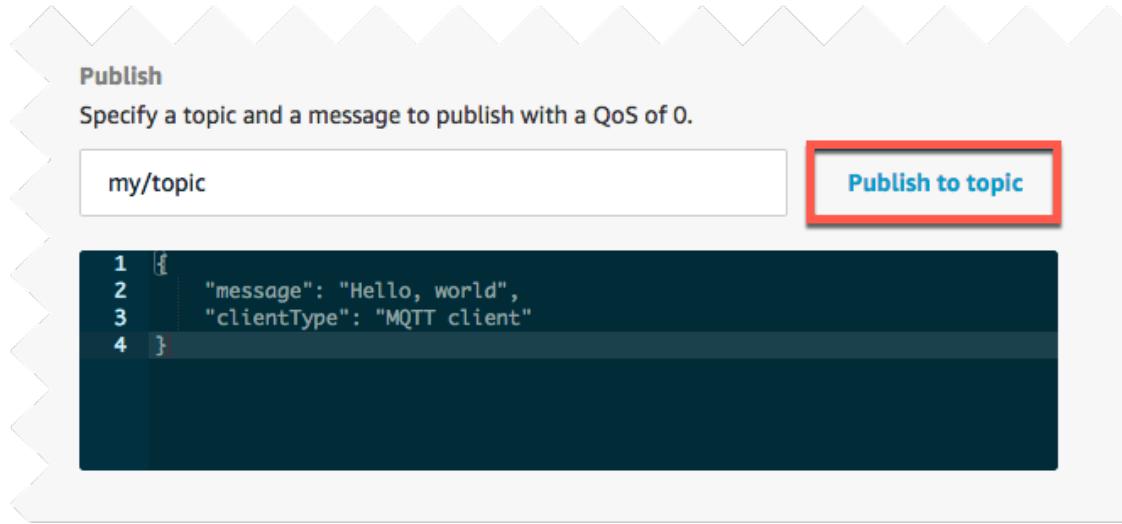
##### Note

We do not recommend using personally identifiable information in topic names.

In the message payload section, type the following JSON:

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

Choose **Publish to topic**. You should see the message in the AWS IoT MQTT client (choose **my/topic** in the **Subscription** column to see the message).



## Configure and Test Rules

The AWS IoT rules engine listens for incoming MQTT messages that match a rule. When a matching message is received, the rule takes some action with the data in the MQTT message (for example, writing data to an Amazon S3 bucket, invoking a Lambda function, or sending a message to an Amazon SNS topic). In this step, you will create and configure a rule to send the data received from a device to an Amazon SNS topic. Specifically, you will:

- Create an Amazon SNS topic.
- Subscribe to the Amazon SNS topic using a cell phone number.
- Create a rule that will send a message to the Amazon SNS topic when a message is received from your device.
- Test the rule using the MQTT client.

## Create an SNS Topic

Use the Amazon SNS console to create an Amazon SNS topic.

### Note

Amazon SNS is not available in all AWS regions.

1. Open the [Amazon SNS console](#).
2. On the left pane, choose **Topics**.

The screenshot shows the AWS SNS dashboard. On the left, a sidebar menu includes 'Topics' (which is selected and highlighted in red), 'Applications', 'Subscriptions', and 'Text messaging (SMS)'. The main content area has a header 'SNS dashboard'. Under 'Common actions', there are five items: 'Create topic' (with a description: 'Create a communication channel to send messages and subscribe to notifications'), 'Create platform application' (with a description: 'Create a platform application for mobile devices'), 'Create subscription' (with a description: 'Subscribe an endpoint to a topic to receive messages published to that topic'), 'Publish message' (with a description: 'Publish a message to a topic or as a direct publish to a platform endpoint'), and 'Publish text message (SMS)' (with a description: 'Publish a text message to a phone number'). To the right, under 'Resources', it says 'You are using the following Amazon SNS resources in the us-west-2 region:' followed by a table with four rows: Topic (0), Subscriptions (0), Applications (0), and Endpoints (0). Below this is a 'More info' section with links to 'Getting started', 'Documentation', 'API reference', 'Forums', and 'Service health'.

3. Choose **Create new topic**.

The screenshot shows the 'Topics' list page. The sidebar menu on the left is identical to the one in the previous screenshot. The main content area has a header 'Topics'. At the top, there are three buttons: 'Publish to topic' (disabled), 'Create new topic' (highlighted with a red box), and 'Actions ▾'. Below this is a 'Filter' input field. A table lists topics with columns 'Name' and 'ARN'. One row is shown: 'My\_SNS\_Topic' with ARN 'arn:aws:sns:us-west-2::My\_SNS\_Topic'. At the bottom of the table, it says 'Total Items: 1' and 'Selected Items: 0'.

4. Type a topic name and a display name, and then choose **Create topic**.

The screenshot shows the 'Create new topic' dialog. It has two input fields: 'Topic name' containing 'MylotSNSTopic' and 'Display name' containing 'SNS Topic'. At the bottom right, there are 'Cancel' and 'Create topic' buttons, with 'Create topic' being highlighted with a red box.

**Note**

We do not recommend using personally identifiable information in Amazon SNS topic names.

5. Make a note of the ARN for the topic you just created.

The screenshot shows the AWS SNS Topics page. On the left, there's a navigation sidebar with links for SNS dashboard, Topics (which is selected), Applications, Subscriptions, and Text messaging (SMS). The main area has a title 'Topics' and three buttons: 'Publish to topic', 'Create new topic', and 'Actions ▾'. Below these are 'Filter' and 'Name' columns. Two topics are listed:

Name	ARN
My_SNS_Topic	arn:aws:sns:us-west-2::My_SNS_Topic
MyIoTNSTopic	arn:aws:sns:us-west-2::MyIoTNSTopic

## Subscribe to an Amazon SNS Topic

To receive SMS messages on your cell phone, subscribe to the Amazon SNS topic.

1. In the Amazon SNS console, select the check box next to the topic you just created. From the Actions menu, choose **Subscribe to topic**.

The screenshot shows the same AWS SNS Topics page as before. A context menu is open over the 'MyIoTNSTopic' row. The menu items are: 'Edit topic display name', 'Subscribe to topic' (which is highlighted with a red box), 'Confirm a subscription', 'Edit topic policy', 'Edit topic delivery policy', 'Delivery status', and 'Delete topics'.

2. On **Create subscription**, from the **Protocol** drop-down list, choose **SMS**.

In the **Endpoint** field, type the phone number of an SMS-enabled cell phone, and then choose **Create subscription**.

**Note**

Enter the phone number using numbers and dashes only.

Create subscription

Topic ARN: arn:aws:sns:us-west-2:...:MyIotSNSTopic

Protocol: SMS

Endpoint: 1-206-555-0100

**Create subscription**

The Amazon SNS console will display the following message, but you may not receive a confirmation message.

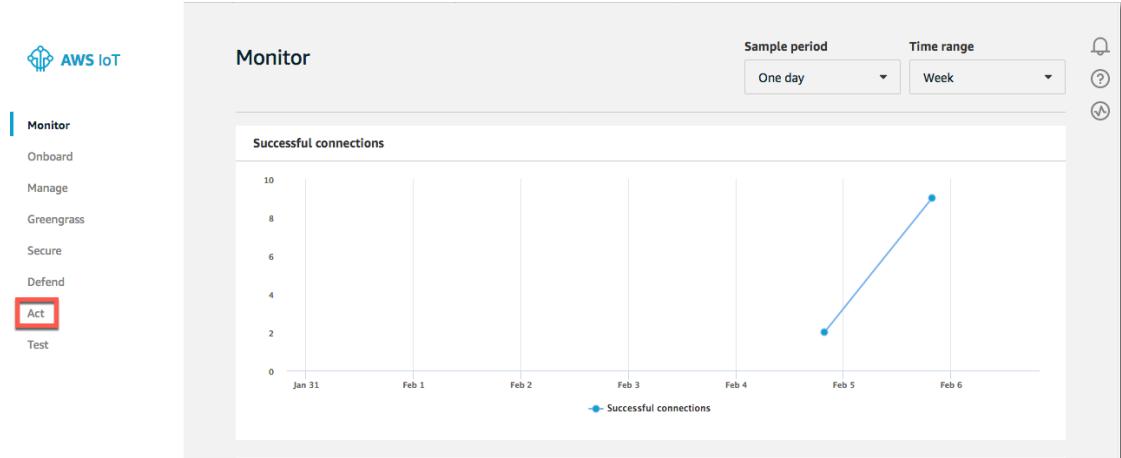
A confirmation message will be sent to the subscribed endpoint.  
Once the subscription has been confirmed, the endpoint will receive notifications from this topic.

## Create a Rule

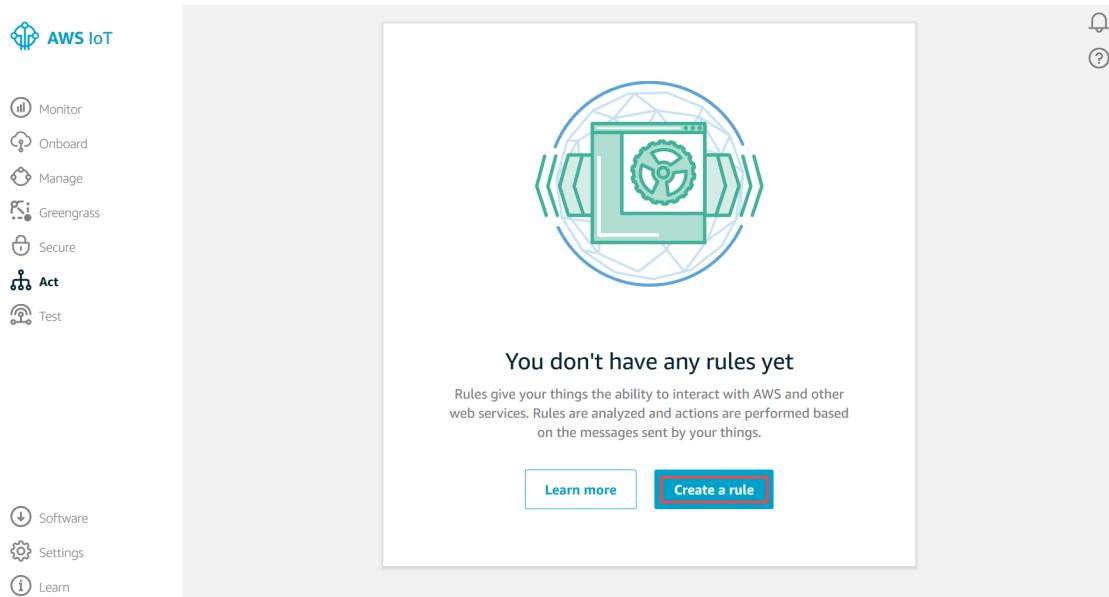
AWS IoT rules consist of a topic filter, a rule action, and, in most cases, an IAM role. Messages published on topics that match the topic filter trigger the rule. The rule action defines which action to take when the rule is triggered. The IAM role contains one or more IAM policies that determine which AWS services the rule can access. You can create multiple rules that listen on a single topic. Likewise, you can create a single rule that is triggered by multiple topics. The AWS IoT rules engine continuously processes messages published on topics that match the topic filters defined in the rules.

In this example, you will create a rule that uses Amazon SNS to send an SMS notification to a cell phone number.

1. In the AWS IoT console, in the left navigation pane, choose **Act**.



2. On the **Act** page, choose **Create a rule**.



3. On the **Create a rule** page, in the **Name** field, type a name for your rule.

**Note**

We do not recommended using personally identifiable information in your rule name.

In the **Description** field, type a description for the rule.

### Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

**Name**

MyiotRule

**Description**

A rule for AWS IoT Getting Started

4. Scroll down to **Rule query statement**. Choose the latest version from the **Using SQL version** drop-down list. In the **Rule query statement** field, type **SELECT \* FROM 'my/topic'**.

**SELECT \*** specifies that you want to send the entire MQTT message that triggered the rule. **FROM 'my/topic'** is the topic filter. The rules engine uses the topic filter to determine which rules to trigger when an MQTT message is received.

**Rule query statement**

Indicate the source of the messages you want to process with this rule.

Using SQL version

2016-03-23 ▾

**Rule query statement**

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

1 SELECT \* FROM 'my/topic'

5. In **Set one or more actions**, choose **Add action**.

**Set one or more actions**

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

**Add action**

6. On the **Select an action** page, select **Send a message as an SNS push notification**, and then choose **Configure action**.

**Select an action**

Select an action.



Insert a message into a DynamoDB table  
DYNAMODB



Split message into multiple columns of a DynamoDB table (DynamoDBv2)  
DYNAMODBV2



Send a message to a Lambda function  
LAMBDA



Send a message as an SNS push notification  
SNS



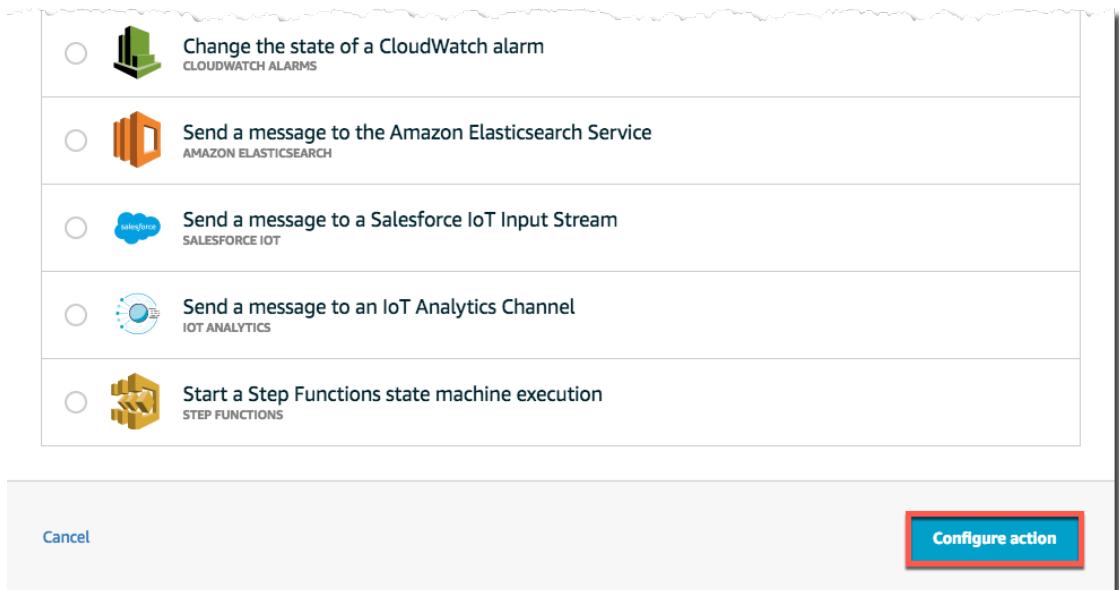
Send a message to an SQS queue  
SQS



Send a message to an Amazon Kinesis Stream  
AMAZON KINESIS



Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH



7. On the **Configure action** page, under **SNS target**, choose **Select** to expand the SNS topic drop down. Then choose **Select** next to the Amazon SNS topic you created earlier. Under **Message format** select **JSON**.

**Configure action**

Send a message as an SNS push notification  
SNS

\*SNS target

MyIotSNSTopic	Create	Clear	Select
---------------	--------	-------	--------

Message format

JSON
------

8. Now give AWS IoT permission to publish to the Amazon SNS topic on your behalf when the rule is triggered. Choose **Create a new role**. Enter a name for your new role in the **IAM role name** field. After you have entered the name, choose **Create a new role**.

Configure action

 Send a message as an SNS push notification  
SNS

\*SNS target  
MyiotSNSTopic [Create](#) [Clear](#) [Select](#)

Message format  
JSON

Choose or create a role to grant AWS IoT access to perform this action.

\*IAM role name  
Choose a role [Update role](#) [Create a new role](#)

[Cancel](#) [Add action](#)

- Under **IAM role name**, choose **Create a new role** to apply the permissions to the newly created role, choose the newly created role, and choose **Add action**.

Configure action

 Send a message as an SNS push notification  
SNS

---

\*SNS target

MyiotSNSTopic	<a href="#">Create</a>	<a href="#">Clear</a>	<a href="#">Select</a>
---------------	------------------------	-----------------------	------------------------

Message format

JSON
------

---

Choose or create a role to grant AWS IoT access to perform this action.

\*IAM role name

MyiotRuleRole	<a href="#">Update role</a>	<a href="#">Create a new role</a>
---------------	-----------------------------	-----------------------------------

[Cancel](#) Add action

10. On the **Create a Rule** page, choose **Create rule**.

**Set one or more actions**

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

 Send a message as an SNS push notification  
MyiotSNSTopic      [Remove](#) [Edit](#)

[Add action](#)

**Error action**

Optionally set an action that will be executed when something goes wrong with processing your rule.

[Add action](#)

**Tags**

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. [Learn more](#) about tagging your AWS resources.

Tag name	Value	Clear
Provide a tag name, e.g. Manufacturer	Provide a tag value, e.g. Acme-Corporation	<a href="#">Clear</a>

[Add another](#)

[Cancel](#) [Create rule](#)

For more information about creating rules, see [AWS IoT Rules](#).

## Test the Amazon SNS Rule

You can test your rule by using the AWS IoT MQTT client:

1. In the [AWS IoT console](#), in the left navigation pane, choose **Test**.
2. On the MQTT client page, in the **Publish** section, in the **Specify a topic and a message to publish...** field, type **my/topic** or the topic you used in the rule. In the message payload section, type the following JSON:

```
{  
    "default": "Hello, from AWS IoT console",  
    "message": "Hello, from AWS IoT console"  
}
```

3. Choose **Publish to topic**. You should receive an Amazon SNS message on your cell phone.

Congratulations! You have successfully created and configured a rule that sends data received from a device to an Amazon SNS topic.

## Next Steps

For more information about AWS IoT rules, see [AWS IoT Rule Tutorials \(p. 50\)](#) and [AWS IoT Rules \(p. 243\)](#).

## Create and Track an AWS IoT Job

AWS IoT jobs enable you to deploy and track management tasks in your device fleet. You can use jobs to send remote actions to one or many devices at once, control the deployment of jobs to your devices, and track the current and past status of job executions for each device.

This topic shows how to create and deploy a sample job to a device. It walks you through the steps required to create a job and track its events on a device that is configured to communicate with AWS IoT. These instructions assume that you're using a Raspberry Pi, but they could be adapted for other Linux-based devices.

Here are some possible scenarios for using jobs:

- Updating device firmware, software, or files, such as security certificates.
- Performing administrative tasks, such as restarting devices or performing diagnostics.
- Restoring devices to factory settings or other known good states.

## Connect Your Device to AWS IoT

Perform the following steps to connect a Raspberry Pi to AWS IoT.

1. Complete the [Connecting Your Raspberry Pi](#) tutorial. When you're done, you'll have an AWS IoT thing registered in your AWS account named *MyRaspberryPi*. You'll also have fully configured security certificates on your device.
2. Complete the [Using the AWS IoT Device SDK for JavaScript](#) tutorial. When you're done, your device will be connected to AWS IoT, and you'll be able to run the sample code that comes with the AWS IoT Device SDK for JavaScript.

Now your device is ready to use AWS IoT jobs.

## Run the Jobs Sample

The AWS IoT Device SDK for JavaScript includes a sample named `jobs-example.js`. This sample can receive messages from the [AWS IoT console](#) to verify connectivity with the AWS IoT platform. It can also receive and process job executions that originate from the AWS IoT jobs service.

You can run this sample by using the following command. Use the REST endpoint of your Raspberry Pi as the value of the `-H` parameter.

```
node examples/jobs-example.js -f ~/certs -H <PREFIX>.iot.<REGION>.amazonaws.com -  
T thingName
```

If you've created a configuration file that contains the thing name and the host endpoint (the REST endpoint of your device), you can use the following command.

```
node examples/jobs-example.js -f ./certs -F your config file name.json
```

## Create a Job Document

A job document is a JSON document that provides all of the information that your device needs to execute a job. The AWS IoT Device SDK for JavaScript uses a property named `operation` to route job documents to specific handlers. The `jobs-example.js` program has a sample handler for an operation named `customJob`. We're going to create a job document named `example-job.json` for this handler. The `example-job.json` file should contain the following JSON object.

```
{  
  "operation": "customJob",  
  "otherInfo": "someValue"  
}
```

For more sample job documents, see the documentation for the [jobs-agent.js](#) sample.

## Create a Job

Now you're ready to create a job that delivers the job document to all of the devices that you specify. To create a job, you can use the AWS IoT console, the AWS IoT SDK, or the AWS IoT CLI.

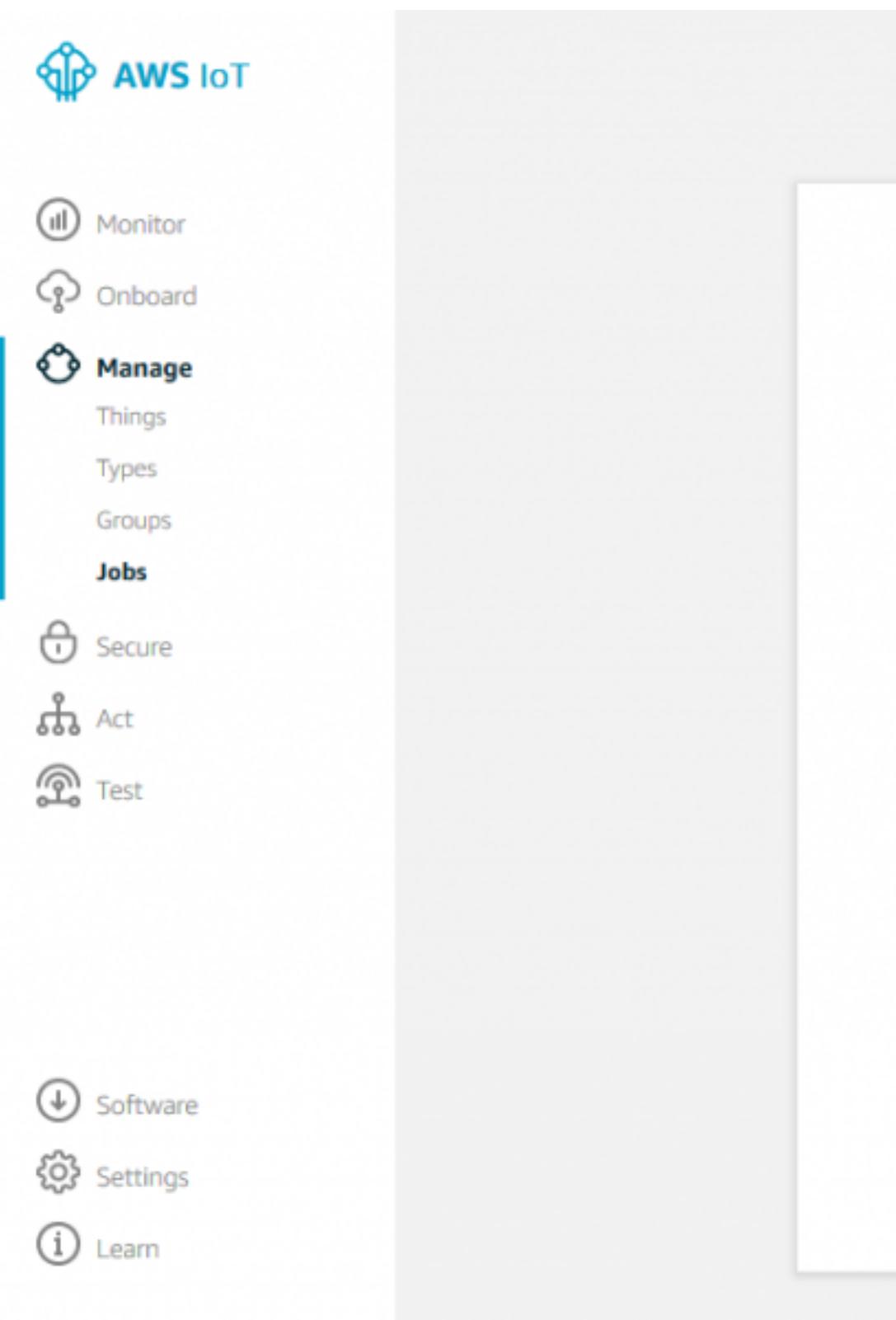
The following example shows how to create a job by using the [AWS IoT CLI](#).

```
aws iot create-job \  
--job-id "example-job-01" \  
--targets "arn:aws:iot::::thing/MyRaspberryPi" \  
--document file:///example-job.json \  
--description "My First test job" \  
--target-selection SNAPSHOT
```

If you store your job document in Amazon Simple Storage Service, use the `--document-source` parameter instead of the `--document` parameter to specify the Amazon S3 URL for the job document.

If you prefer to use the AWS IoT console, follow these steps to create a job:

1. Upload the job document to an Amazon S3 bucket. For information about uploading documents to Amazon S3, see [How Do I Upload Files and Folders to an S3 Bucket?](#) in the *Amazon Simple Storage Service Console User Guide*.
2. In the AWS IoT console, choose **Manage**, and then choose **Jobs**.
3. Choose **Create a job**.



4. On the **Select a job** page, choose **Create custom job**.

The screenshot shows a modal window titled "CREATE JOB" with a large "X" button in the top-left corner. The main title "Select a job" is displayed prominently. Below it, there are four listed options:

- Create a custom job**: Send a request to acquire an executable job file from one of your devices connected to AWS IoT.
- Create an Amazon FreeRTOS Over-the-air (OTA) update job**: This OTA update job will send your firmware image securely over FreeRTOS-based devices.
- Create a Greengrass Core update job**: Create a snapshot job to update one or more Greengrass Core or OTA agent version.

5. On the **Create a job** page, enter a unique job ID.

**Note**

We do not recommended using personally identifiable information in your job ID.

Under **Select devices to update**, select the device that you connected to AWS IoT.

The screenshot shows the 'Create a job' page in the AWS IoT Developer Guide. At the top, there is a back arrow icon and the text 'CREATE JOB'. Below this, the main title 'Create a job' is displayed. The first section, 'Job ID', contains a text input field with the value 'example-job-01', which is highlighted with a red border. The next section, 'Description', has an empty text input field. Below these, a heading 'Select devices to update' is followed by the instruction 'Browse and select the devices you want to include in this job.' A message indicates '1 thing(s) and 0 thing group(s) selected.' The 'Things' tab is active, showing a search bar and a list item 'MyRaspberryPi' with a checked checkbox, also highlighted with a red border.

CREATE JOB

## Create a job

Job ID

example-job-01

Description

Select devices to update

Browse and select the devices you want to include in this job.

1 thing(s) and 0 thing group(s) selected.

**Things** Thing Groups Summary

Search

MyRaspberryPi

6. Scroll down to **Add a job file** and choose the job document file that you uploaded to Amazon S3. Under **Job type**, select **Your job will complete after deploying to the selected devices/groups (snapshot)**. (The other option, **Your job will continue deploying to any devices added to the selected groups (continuous)**, is for deploying a job to groups of devices as devices are added to each group.) Leave the **Job executions rollout configuration** unchanged. Choose **Create**.

**Add a job file**

Upload a job file that defines what your job should do.

`example-job.json`

**Pre-sign resource URLs**

For an extra layer of security, you can pre-sign URLs that refer to resources in your job file.

Cannot find pre-sign url placeholder in the job file. Skip pre-sign configuration.

**Job type**

A job can run on the devices and/or groups selected, or remain open until completed.

Your job will complete after deploying to the selected devices

Your job will continue deploying to any devices added to the job

**Job executions rollout configuration**

Specify how quickly devices will be notified of a pending job execution.

**Maximum per minute (1-1000)**

1000

7. Your new job appears on the **Jobs** page.

The screenshot shows the AWS IoT console interface. On the left, there's a sidebar with icons for Monitor, Onboard, Manage (which is selected), Things, Types, Groups, and Jobs. The main area is titled 'Jobs' and displays a card for a job named 'example-job-01'. The card has a 'SNAPSHOT' button and three vertical dots for more options. The job status is shown as 'Completed'.

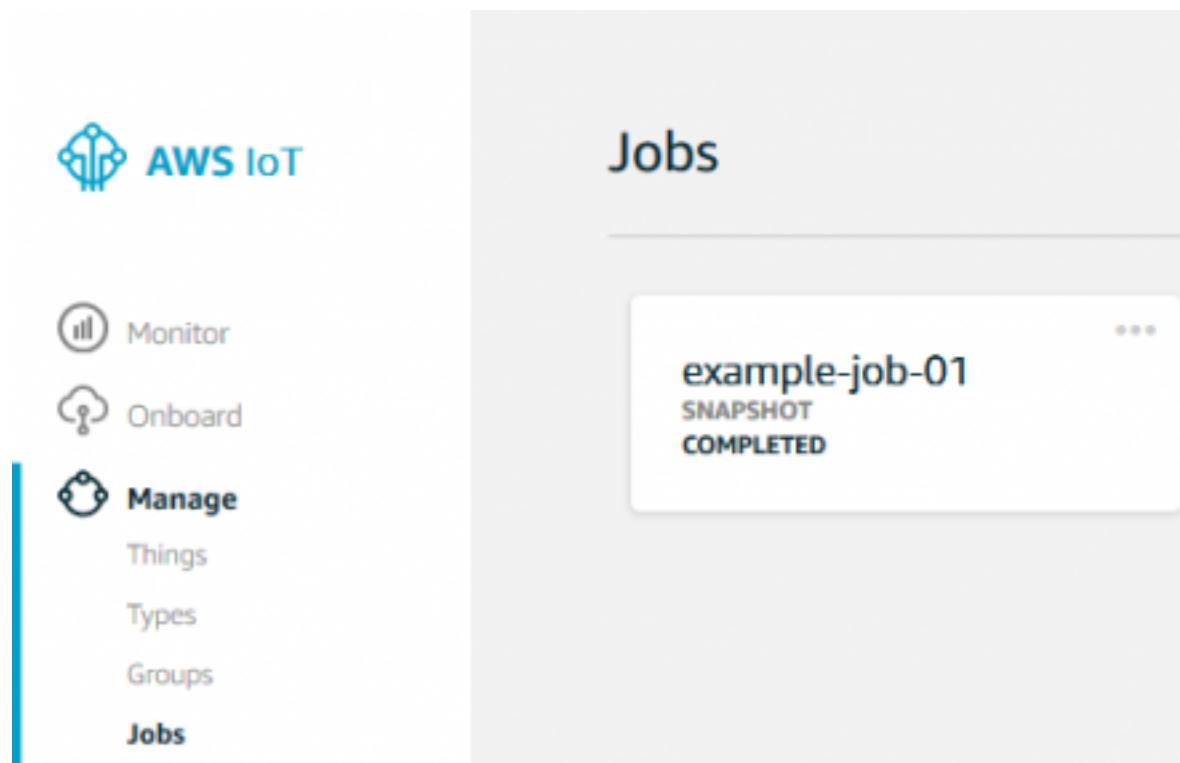
For more information about creating and deploying jobs, see the [AWS IoT Jobs documentation](#).

## Execute the Job on a Device

After your job is created, the jobs service sends a notification of a pending job to your device. Your device gets the job details and job document through the [NextJobExecutionChanged](#) API. The `jobs-example.js` sample that you've already run executes the job on the device. When the job is complete, the sample publishes its completed status by using the [UpdateJobExecution](#) API. When you run the sample on your device, you'll see the following output.

```
node examples/jobs-example.js -f ./certs -F config.json
connect
startJobNotifications completed for thing: MyRaspberryPi
customJob operation handler invoked, jobId: example-job-01
```

When you refresh the **Jobs** page, you can see that your job has successfully completed.



## Tracking Job Progress with Job and Job Execution Events

You can use `Job` events and `JobExecution` events to track the progress of your job.

This is a helpful way to alert users, system administrators, or other members of your system that a job is complete or that a job execution has changed its status. For example, you can alert a user about a firmware update on a device or inform a system administrator about an issue in their device fleet that needs to be investigated and resolved.

Job events for the job in this example are published to the following topics when the job is completed or canceled.

```
$aws/events/job/example-job-01/completed  
$aws/events/job/example-job-01/canceled
```

Job execution events for the job in this example are sent to the following topics when the job execution reaches one of the possible final statuses.

```
$aws/events/jobExecution/example-job-01/succeeded  
$aws/events/jobExecution/example-job-01/failed  
$aws/events/jobExecution/example-job-01/rejected  
$aws/events/jobExecution/example-job-01/canceled  
$aws/events/jobExecution/example-job-01/removed
```

When the job execution on your device succeeds, AWS IoT publishes a [JobExecutionSucceeded](#) event. You can see this event by navigating to the AWS IoT **Test** page and subscribing to the `$aws/events/jobExecution/example-job-01/succeeded` topic in the MQTT client.

The screenshot shows two side-by-side interfaces. On the left is the AWS IoT Test page, which has a sidebar with icons for Monitor, Onboard, Manage, Greengrass, Secure, Act, and Test. The 'Test' icon is highlighted with a red box. On the right is an 'MQTT client' interface with a 'Subscriptions' section containing 'Subscribe to a topic' and 'Publish to a topic'. A red box highlights the topic path '\$aws/events/jobExecution/example-job-01/succeeded' in the 'Publish to a topic' section. The 'Max message payload' field is set to 100.

The following message appears when the job execution for your device has successfully completed.

## Publish

Specify a topic and a message to publish with a QoS of 0.

```
$aws/events/jobExecution/example-job-01/succeeded
```

```
1  [
2    "message": "Hello from AWS IoT console"
3  ]
```

\$aws/events/jobExecution/example-job-01/...

Dec 19, 2017

```
{
  "eventType": "JOB_EXECUTION",
  "eventId": "af479061-a800-4d1f-a557-bbcd71243f7e",
  "timestamp": "1513709703",
  "operation": "succeeded",
  "jobId": "example-job-01",
  "thingArn": "arn:aws:iot:us-east-1:xxxxxxxxxxxx:thing",
  "status": "SUCCEEDED",
  "statusDetails": {
    "key": "value"
  }
}
```

AWS IoT also publishes a completed job event. You can see this event by subscribing to the \$aws/events/job/example-job-01/completed topic in the MQTT client.

## Publish

Specify a topic and a message to publish with a QoS of 0.

```
$aws/events/job/example-job-01/completed
```

```
1  {
2    "message": "Hello from AWS IoT console"
3 }
```

**\$aws/events/job/example-job-01/completed**

Dec 19, 2017

```
{
  "eventType": "JOB",
  "eventId": "a1817d74-0fld-42b3-b03f-acfc90af41e",
  "timestamp": "1513709645",
  "operation": "completed",
  "jobId": "example-job-01",
  "status": "COMPLETED",
  "targetSelection": "SNAPSHOT",
  "targets": [
    "arn:aws:iot:us-east-1:[REDACTED]:thing/MyRaspbe"
  ],
  "description": "Job example-job-01 for Thing MyRaspb",
  "completedAt": "1513709645105",
  "createdAt": "1513709615488",
  "lastUpdatedAt": "1513709645105",
  "jobProgressDetails": {
    "numberOfCanceledThings": 0,
    "numberOfRejectedThings": 0,
    "numberOfFailedThings": 0,
    "numberOfRemovedThings": 0,
    "numberOfSucceededThings": 1
  }
}
```

# AWS IoT Rules Tutorials

The following tutorials show you how to create and test AWS IoT rules. Before you begin, be sure to complete the [AWS IoT Getting Started Tutorial \(p. 5\)](#). It shows you how to create an AWS account and register a device in AWS IoT, which are prerequisites for these tutorials.

The scenario in this tutorial is a greenhouse with rows of plants. Each plant has a moisture sensor. At a predetermined interval, the moisture sensor sends its data to AWS IoT. The AWS IoT rules engine receives this data and writes it to a DynamoDB table. You create a rule to write data to DynamoDB and emulate the sensors using the AWS IoT MQTT client.

An AWS IoT rule consists of an SQL SELECT statement, a topic filter, and a rule action. Devices send information to AWS IoT by publishing messages to MQTT topics. The SQL SELECT statement allows you to extract data from an incoming MQTT message. The topic filter of an AWS IoT rule specifies one or more MQTT topics. The rule is triggered when an MQTT message is received on a topic that matches the topic filter. Rule actions allow you to take the information extracted from an MQTT message and send it to another AWS service. Rule actions are defined for AWS services like Amazon DynamoDB, AWS Lambda, Amazon SNS, and Amazon S3. By using a Lambda rule, you can call other AWS or third-party web services. For a complete list of rule actions, see [AWS IoT Rule Actions \(p. 249\)](#).

In these tutorials, we assume that you're using the AWS IoT MQTT client and that you are using `my/greenhouse` as the topic filter in the rules.

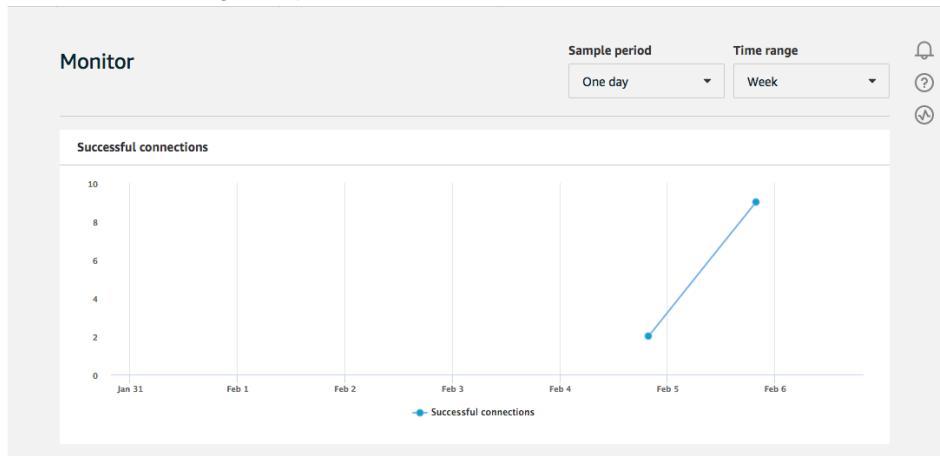
You can also use your own device, but you must know on which MQTT topic your device publishes so you can specify it as the topic filter in the rule. For more information, see [AWS IoT Rules \(p. 243\)](#).

## Creating an Amazon DynamoDB Rule

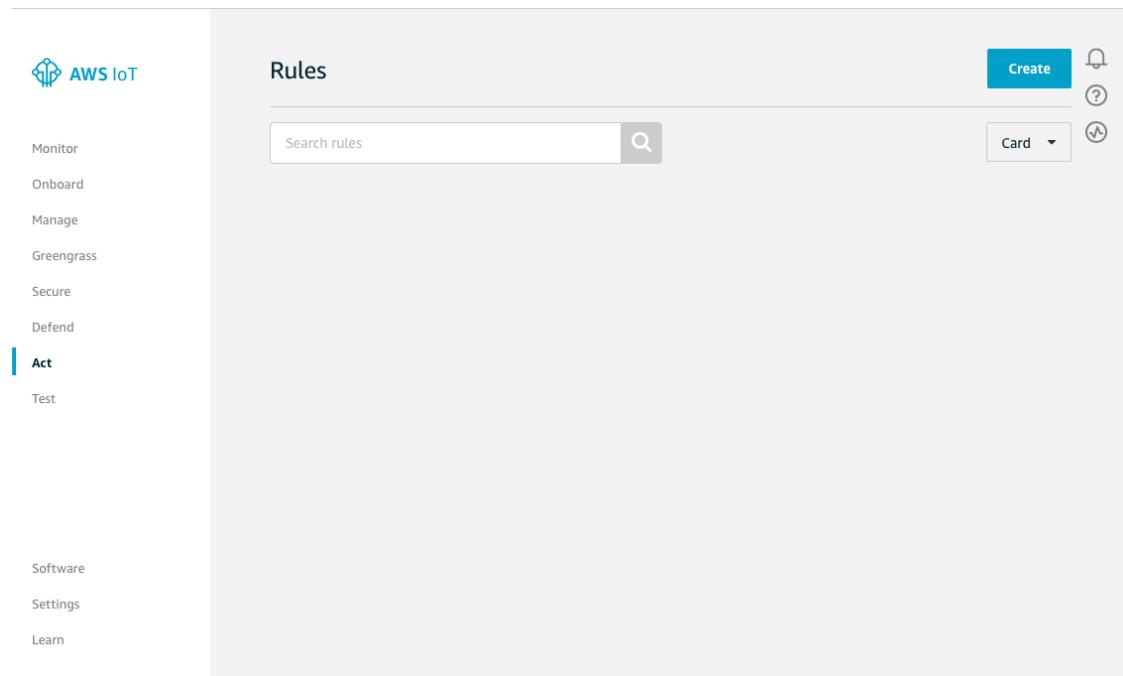
DynamoDB rules allow you to take information from an incoming MQTT message and write it to a DynamoDB table.

### To create a DynamoDB rule

1. In the [AWS IoT console](#), in the navigation pane, choose **Act**.



2. On the **Rules** page, choose **Create**.



3. On the **Create a rule** page, enter a name and description for your rule.

**Note**

We do not recommend the use of personally identifiable information in your rule names or descriptions.

### Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name  
GreenhouseRule

Description  
A DynamoDB rule for a greenhouse

4. Under **Rule query statement**, choose the latest version from the **Using SQL version** list. For **Rule query statement**, enter:

```
SELECT * FROM 'my/greenhouse'
```

("SELECT \*" specifies that you want to send the entire MQTT message that triggered the rule. "FROM 'my/greenhouse'" tells the rules engine to trigger this rule when an MQTT message is received whose topic matches this topic filter. Choose **Add action**.

#### Rule query statement

Indicate the source of the messages you want to process with this rule.

Using SQL version

2016-03-23 ▾

#### Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT * FROM 'my/greenhouse'
```

#### Set one or more actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

Add action

5. On the **Select an action** page, choose **Insert a message into a DynamoDB table**, and then choose **Configure action**.

Select an action

Select an action.

-  Insert a message into a DynamoDB table  
DYNAMODB
-  Split message into multiple columns of a DynamoDB table (DynamoDBv2)  
DYNAMODBV2
-  Send a message to a Lambda function  
LAMBDA
-  Send a message as an SNS push notification  
SNS
-  Send a message to an SQS queue  
SQS
-  Send a message to an Amazon Kinesis Stream  
AMAZON KINESIS
-  Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH
-  Store a message in an Amazon S3 bucket  
S3
-  Send a message to an Amazon Kinesis Firehose stream  
AMAZON KINESIS FIREHOSE
-  Send message data to CloudWatch  
CLOUDWATCH METRICS
-  Change the state of a CloudWatch alarm  
CLOUDWATCH ALARMS
-  Send a message to the Amazon Elasticsearch Service  
AMAZON ELASTICSEARCH
-  Send a message to a Salesforce IoT Input Stream  
SALESFORCE IOT
-  Send a message to an IoT Analytics Channel  
IOT ANALYTICS
-  Start a Step Functions state machine execution  
STEP FUNCTIONS

[Cancel](#) Configure action

6. On the **Configure action** page, choose **Create a new resource**.

Configure action

 Insert a message into a DynamoDB table  
DYNAMODB

The table must contain Partition and Sort keys.

\*Table name  
Choose a resource ▾  Create a new resource

\*Partition key  
Required field does not exist

\*Partition key type  
Required field does not exist

\*Partition key value

Sort key  
Optional field does not exist

Sort key type  
Optional field does not exist

Sort key value

Write message data to this column

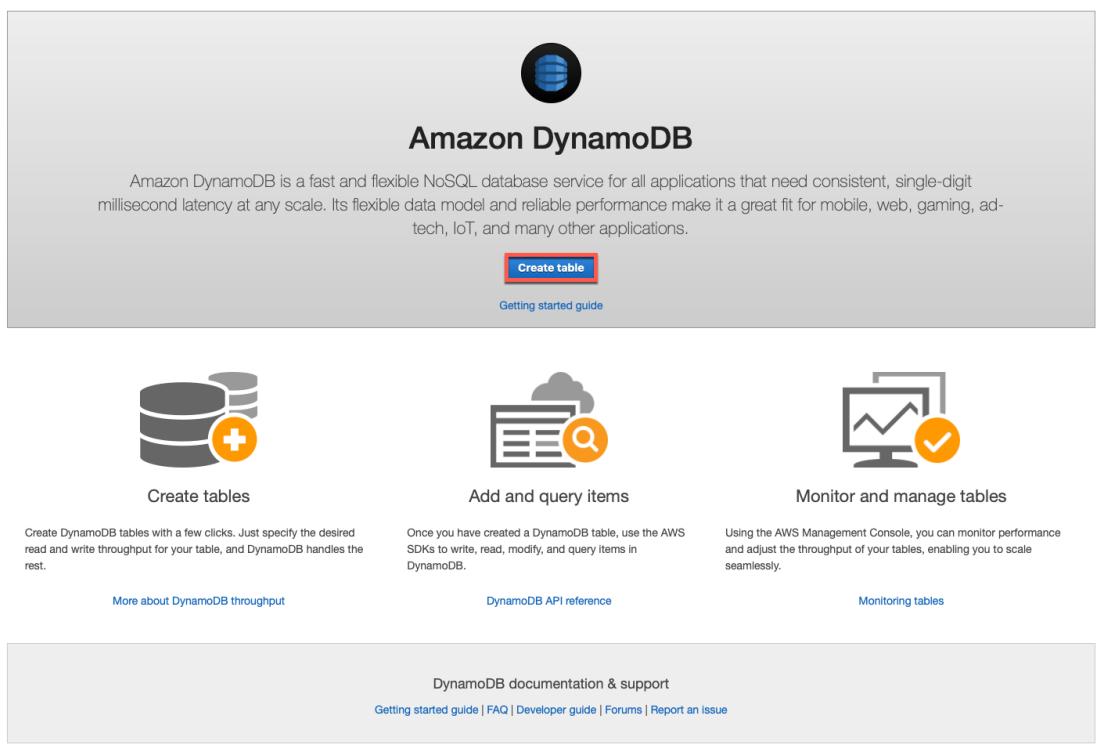
Operation 

Choose or create a role to grant AWS IoT access to perform this action.

No role selected 

7. On the **Amazon DynamoDB** page, choose **Create table**.



8. On the **Create DynamoDB table** page, enter a name in **Table name**. In **Partition key**, enter **Row**. Select **Add sort key**, and then enter **PositionInRow** in the **Sort key** field. Row represents a row of plants in a greenhouse. PositionInRow represents the position of a plant in the row. Choose **String** for both the partition and sort keys, and then choose **Create**. It takes a few seconds to create your DynamoDB table. Close the browser tab where the Amazon DynamoDB console is open. If you don't close the tab, your DynamoDB table is not displayed in the **Table name** list on the **Configure action** page of the AWS IoT console.

**Create DynamoDB table**

DynamoDB is a schema-less database that only requires a table name and primary key. The table's primary key is made up of one or two attributes that uniquely identify items, partition the data, and sort data within each partition.

<b>Table name*</b>	GreenhouseTable	<a href="#">Tutorial</a>	<a href="#">?</a>	
<b>Primary key*</b>	Partition key			
	Row	Type: String	<a href="#">Edit</a>	<a href="#">?</a>
<input checked="" type="checkbox"/> Add sort key	PositionInRow	Type: String	<a href="#">Edit</a>	<a href="#">?</a>
<b>Table settings</b>				
Default settings provide the fastest way to get started with your table. You can modify these default settings now or after your table has been created.				
<input checked="" type="checkbox"/> Use default settings <ul style="list-style-type: none"> <li>No secondary indexes.</li> <li>Provisioned capacity set to 5 reads and 5 writes.</li> <li>Basic alarms with 80% upper threshold using SNS topic "dynamodb".</li> <li>Encryption at Rest with DEFAULT encryption type <a href="#">NEW!</a></li> </ul>				
<b><span style="color: #0072bc;">i</span> You do not have the required role to enable Auto Scaling by default.</b> <small>Please refer to <a href="#">documentation</a>.</small>				
<small>Additional charges may apply if you exceed the AWS Free Tier levels for CloudWatch or Simple Notification Service. Advanced alarm settings are available in the CloudWatch management console.</small>				
<a href="#">Cancel</a> <a href="#" style="background-color: #0072bc; color: white; border: 1px solid #0072bc; padding: 2px 10px; border-radius: 5px;">Create</a>				

9. On the **Configure action** page, choose your new table from the **Table name** list. In **Partition key value**, enter  **\${row}** . This instructs the rule to take the value of the **row** attribute from the MQTT

message and write it into the **Row** column in the DynamoDB table. In **Sort key value**, enter `#{row}`. This writes the value of the `index` attribute into the **PositionInRow** column. Leave **Write message data to this column** blank. By default, the entire message is written to a column in the table named `Payload`. Choose **Create a new role**.

Configure action

Insert a message into a DynamoDB table  
DYNAMODB

The table must contain Partition and Sort keys.

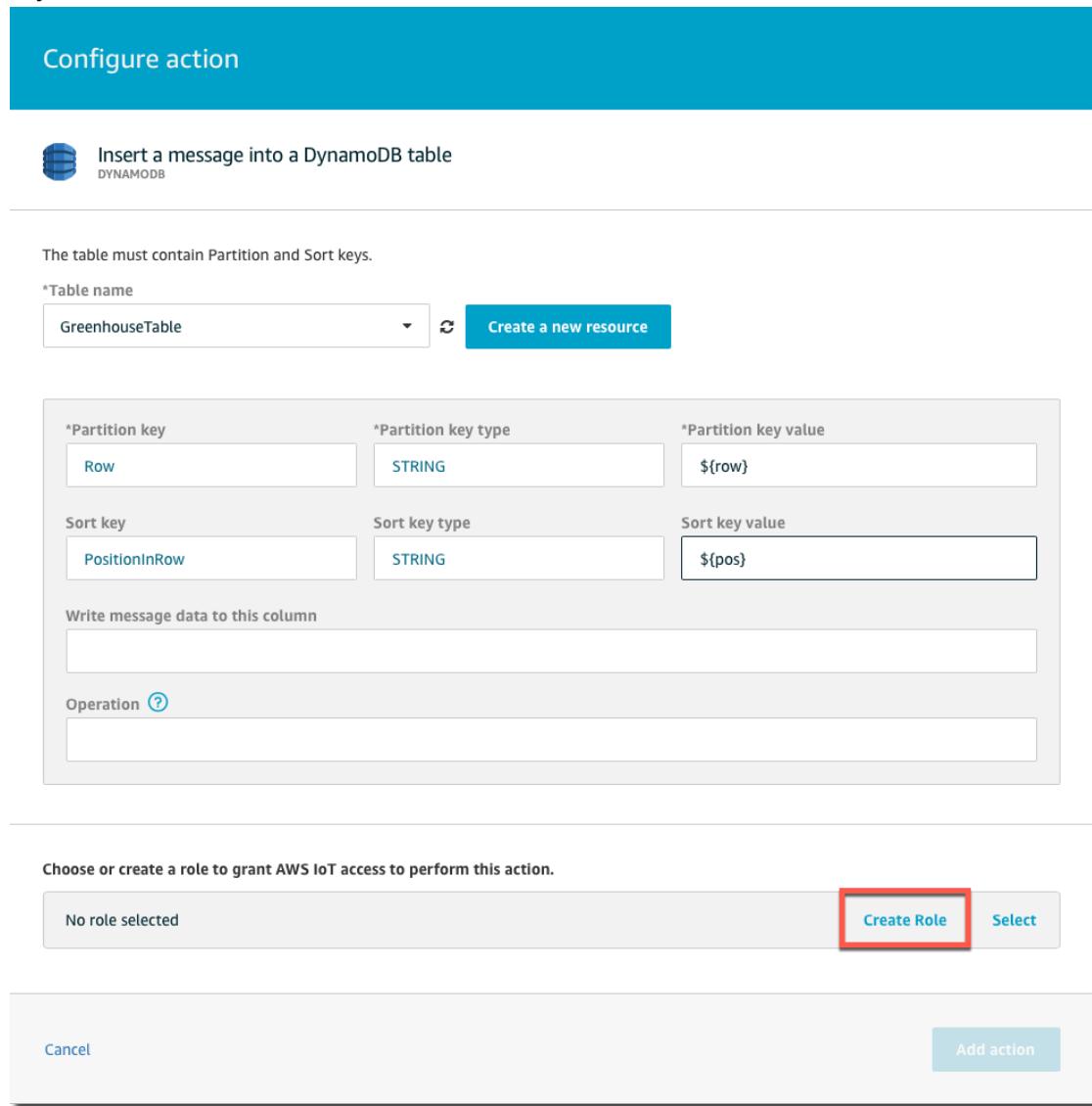
\*Table name: GreenhouseTable [Create a new resource](#)

*Partition key: Row	*Partition key type: STRING	*Partition key value: \${row}
Sort key: PositionInRow	Sort key type: STRING	Sort key value: \${pos}
Write message data to this column:		
Operation:		

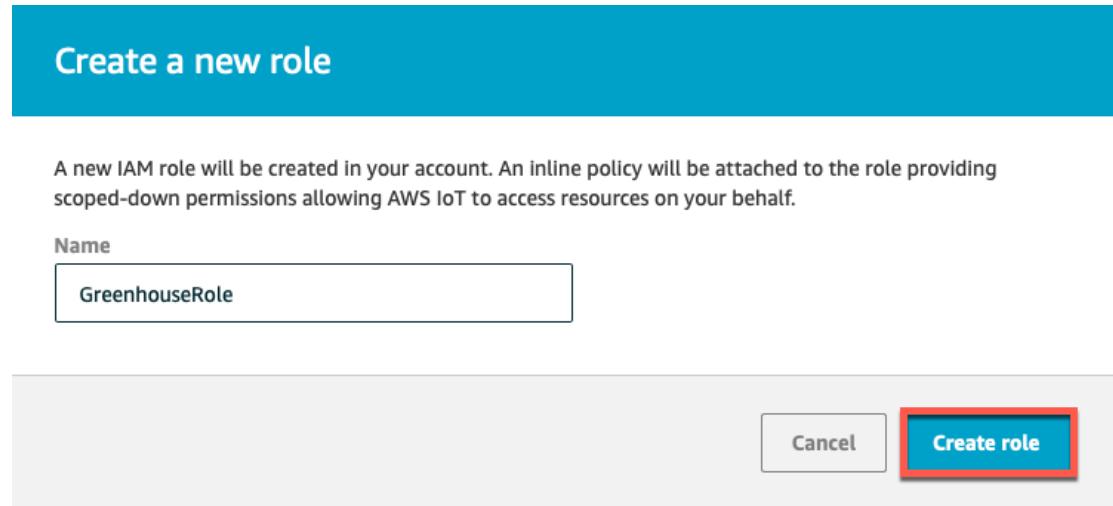
Choose or create a role to grant AWS IoT access to perform this action.

No role selected [Create Role](#) [Select](#)

[Cancel](#) [Add action](#)



10. In **Create a new role**, enter a unique name in **Name**, and then choose **Create role**.



11. Choose **Add action**.

Configure action

 Insert a message into a DynamoDB table  
DYNAMODB

The table must contain Partition and Sort keys.

\*Table name  
GreenhouseTable ▼ Create a new resource

\*Partition key  
Row ▼ STRING \${row}

Sort key  
PositionInRow ▼ STRING \${pos}

Write message data to this column

Operation ?

Choose or create a role to grant AWS IoT access to perform this action.  
GreenhouseRole Policy Attached ✓ Create Role Select

Add action Cancel

12. Choose **Create rule** to create your rule.

## Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

GreenhouseRule

Description

A DynamoDB rule for a greenhouse

### Rule query statement

Indicate the source of the messages you want to process with this rule.

Using SQL version

2016-03-23

Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT * FROM 'my/greenhouse'
```

### Set one or more actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)



Insert a message into a DynamoDB table  
GreenhouseTable

Remove Edit ▾

Add action

### Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.

Add action

### Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. [Learn more](#) about tagging your AWS resources.

Tag name

Provide a tag name, e.g. Manufacturer

Value

Provide a tag value, e.g. Acme-Corporation

Clear

Add another

## Testing an Amazon DynamoDB Rule

1. To test the rule, open the AWS IoT console and from the navigation pane, choose **Test**.
2. Choose **Publish to a topic**. In the **Publish** section, enter **my/greenhouse** as the topic. In the message area, enter the following JSON:

```
{  
    "row" : "0",  
    "index" : "0",  
    "moisture" : "75"  
}
```

The screenshot shows the AWS IoT MQTT client interface. At the top, it says "Connected as iotconsole-". The main area is titled "MQTT client" with a "Subscriptions" tab selected. Below it, there's a "Publish to topic" section. A red box highlights the "Subscribe to topic" input field, which is empty and has a placeholder "Specify a topic to subscribe to, e.g. myTopic/1". A red box also highlights the "Subscribe to topic" button. Further down, there are fields for "Max message capture" (set to 100) and "Quality of Service" (set to 0). The "MQTT payload display" section has three options: "Auto-format JSON payloads (improves readability)" (selected), "Display payloads as strings (more accurate)", and "Display raw payloads (in hexadecimal)". In the "Publish" section, there's a "Publish" input field containing "my/greenhouse" and a "Publish to topic" button (also highlighted with a red box). Below the publish input, there's a code editor window showing the JSON message: { "row" : "0", "index" : "0", "moisture" : "75" }.

Return to the DynamoDB console and choose **Tables**.

The screenshot shows the AWS DynamoDB console. On the left, a sidebar menu includes 'DynamoDB' (selected), 'Dashboard', 'Tables' (highlighted with a red box), 'Backups', 'Reserved capacity', and 'Preferences'. Under 'DAX', it lists 'Dashboard', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main content area is titled 'Create table' and contains a brief description of DynamoDB. A large blue button says 'Create table'. Below this is a section for 'Recent alerts' with a note 'No CloudWatch alarms have been triggered.' and a link 'View all in CloudWatch'. A section for 'Total capacity for US West (Oregon)' shows 'Provisioned read capacity' at 5 and 'Provisioned write capacity' at 5, with 'Reserved' columns at 0. Another section for 'Service health' shows a single entry: 'Amazon DynamoDB (Oregon)' with a green checkmark and the status 'Service is operating normally'. A link 'View complete service health details' is provided.

Select the **GreenhouseTable** and then choose the **Items** tab. Your data is displayed on the **Items** tab.

The screenshot shows the 'Items' tab for the 'GreenhouseTable' in the AWS DynamoDB console. The left sidebar shows 'DynamoDB' selected, along with 'Dashboard', 'Tables' (highlighted with a red box), 'Backups', 'Reserved capacity', and 'Preferences'. Under 'DAX', it lists 'Dashboard', 'Clusters', 'Subnet groups', 'Parameter groups', and 'Events'. The main area shows a table with one item. The table has two columns: 'Row' and 'PositionInRow'. The first row has values '0' and '1'. The payload column contains the JSON object: { "moisture": { "S": "75" }, "pos": { "S": "1" }, "row": { "S": "0" } }.

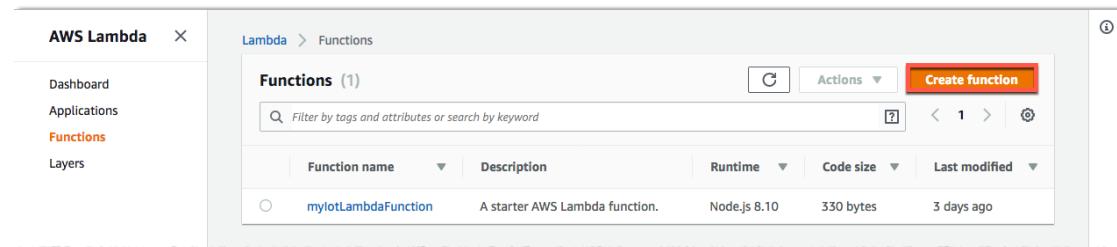
## Creating an AWS Lambda Rule

You can define a rule that calls a Lambda function, passing in data from the MQTT message that triggered the rule. This allows you to extract data from the incoming message and then call another AWS or third-party service. In this tutorial, we assume you have completed the [AWS IoT Getting Started Tutorial \(p. 5\)](#) in which you create and subscribe to an Amazon SNS topic. Now you create a Lambda function that publishes a message to the Amazon SNS topic you created in the [AWS IoT Getting Started Tutorial \(p. 5\)](#). You also create a Lambda rule that calls the Lambda function, passing in some data from the MQTT message that triggered the rule.

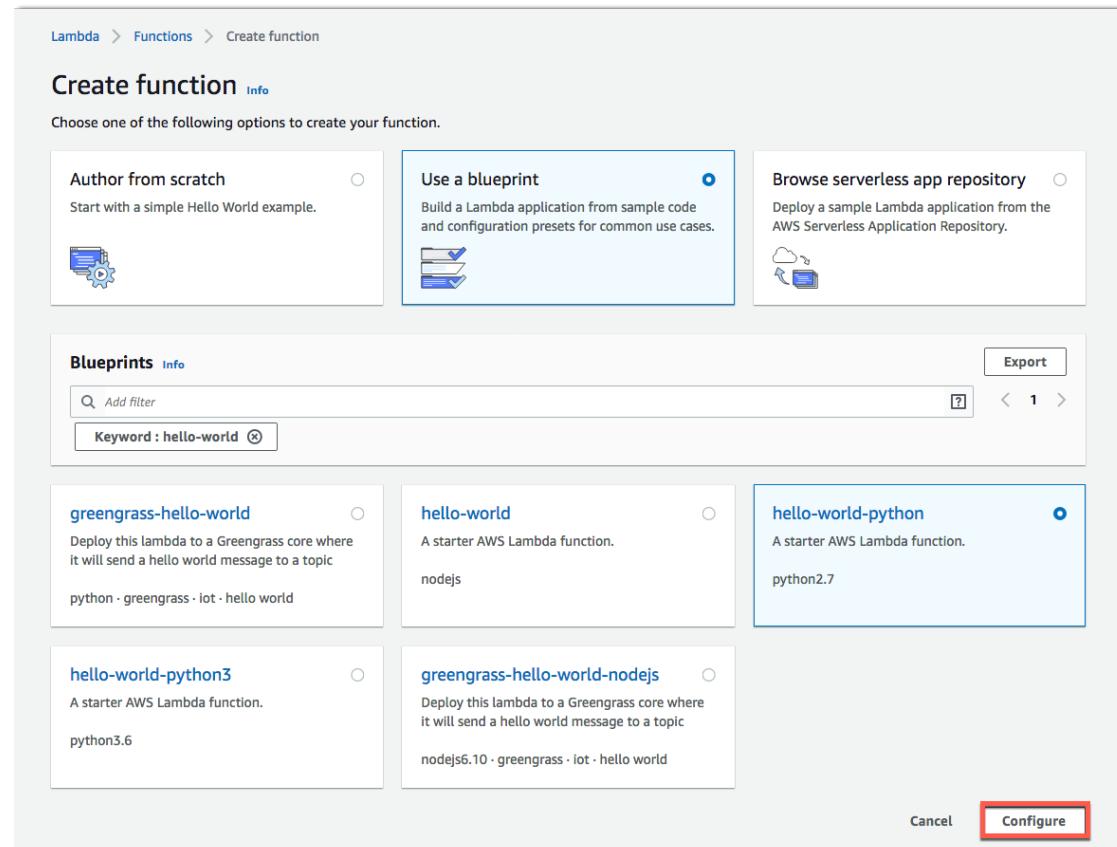
In this tutorial, you use the AWS IoT MQTT client to send a message that triggers the rule.

## Create a Lambda Function

1. In the [AWS Lambda console](#), choose **Create function**.



2. On the **Create function** page, choose **Use a blueprint**. In the **Blueprints** filter field, enter **hello-world**, and then press **Enter**. Choose the **hello-world-python** blueprint, and then choose **Configure**.



3. In **Basic information**, enter a name for your function.

**Note**

We do not recommend the use of personally identifiable information in your rule names or descriptions.

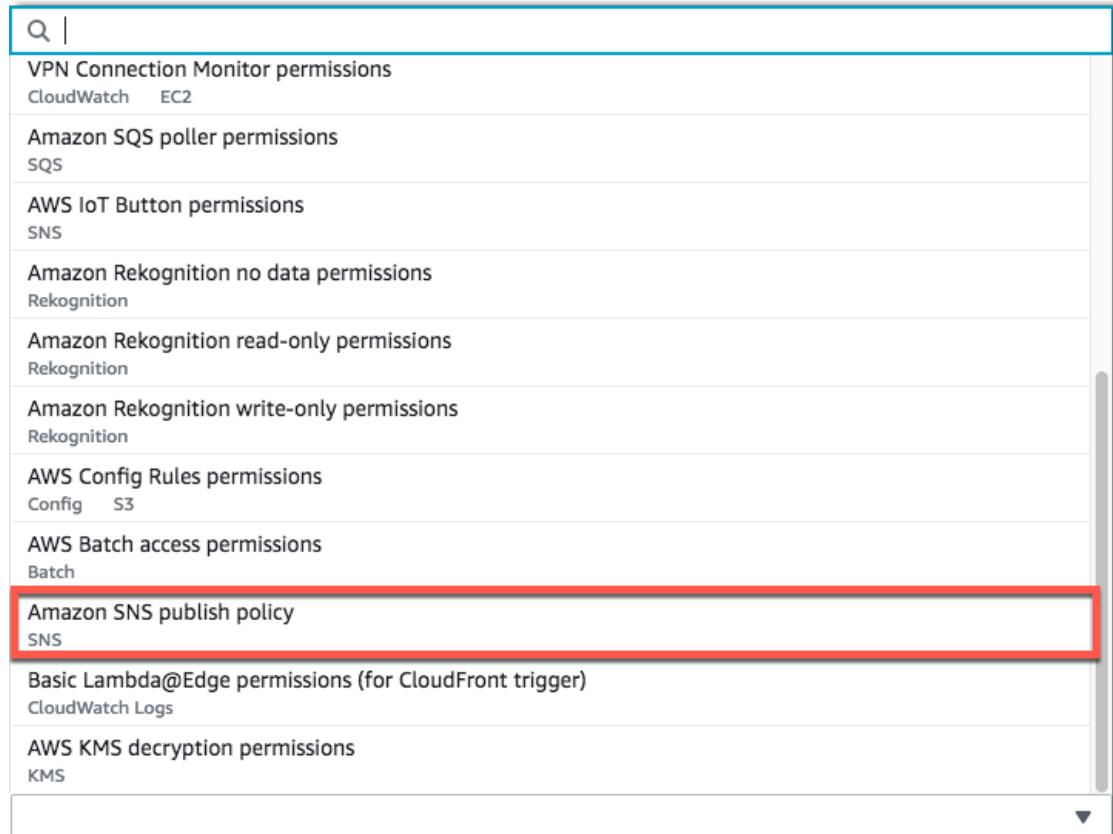
**Basic information** Info

Function name

Execution role  
Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Existing role  
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

4. From **Execution Role**, choose **Create a new role from AWS policy templates**. Under **Role name**, enter a name for the role. From **Policy templates**, choose **Amazon SNS publish policy**. Click outside of the drop-down menu to dismiss it.



VPN Connection Monitor permissions  
CloudWatch EC2

Amazon SQS poller permissions  
SQS

AWS IoT Button permissions  
SNS

Amazon Rekognition no data permissions  
Rekognition

Amazon Rekognition read-only permissions  
Rekognition

Amazon Rekognition write-only permissions  
Rekognition

AWS Config Rules permissions  
Config S3

AWS Batch access permissions  
Batch

**Amazon SNS publish policy** SNS

Basic Lambda@Edge permissions (for CloudFront trigger)  
CloudWatch Logs

AWS KMS decryption permissions  
KMS

5. Choose **Create function**.

Basic information [Info](#)

Function name

myLambdaFunction

Execution role

Choose a role that defines the permissions of your function. To create a custom role, go to the [IAM console](#).

Create a new role from AWS policy templates ▾

i Role creation might take a few minutes. The new role will be scoped to the current function. To use it with other functions, you can modify it in the IAM console.

Role name

Enter a name for your new role.

myiotLambdaFunction2-role

Use only letters, numbers, hyphens, or underscores with no spaces.

Policy templates [Info](#)

Choose one or more policy templates.

Amazon SNS publish policy X  
SNS

Lambda function code

Code is preconfigured by the chosen blueprint. You can configure it after you create the function.

Runtime

Python 2.7

```
1  from __future__ import print_function
2
3  import json
4
5  print('Loading function')
6
7
8  def lambda_handler(event, context):
9      #print("Received event: " + json.dumps(event, indent=2))
10     print("value1 = " + event['key1'])
11     print("value2 = " + event['key2'])
12     print("value3 = " + event['key3'])
13     return event['key1'] # Echo back the first key value
14     #raise Exception('Something went wrong')
15
```

\* These fields are required.

[Cancel](#)

[Previous](#)

[Create function](#)

- In the Lambda console, choose the name of your Lambda function. Information about your Lambda function is displayed. Scroll down to the **Function code** section and replace the existing code with the following:

```
from __future__ import print_function

import json
import boto3

print('Loading function')

def lambda_handler(event, context):

    # Parse the JSON message
    eventText = json.loads(event)

    # Print the parsed JSON message to the console; you can view this text in the Monitoring tab in the Lambda console or in the CloudWatch Logs console
    print('Received event: ', eventText)

    # Create an SNS client
    sns = boto3.client('sns')

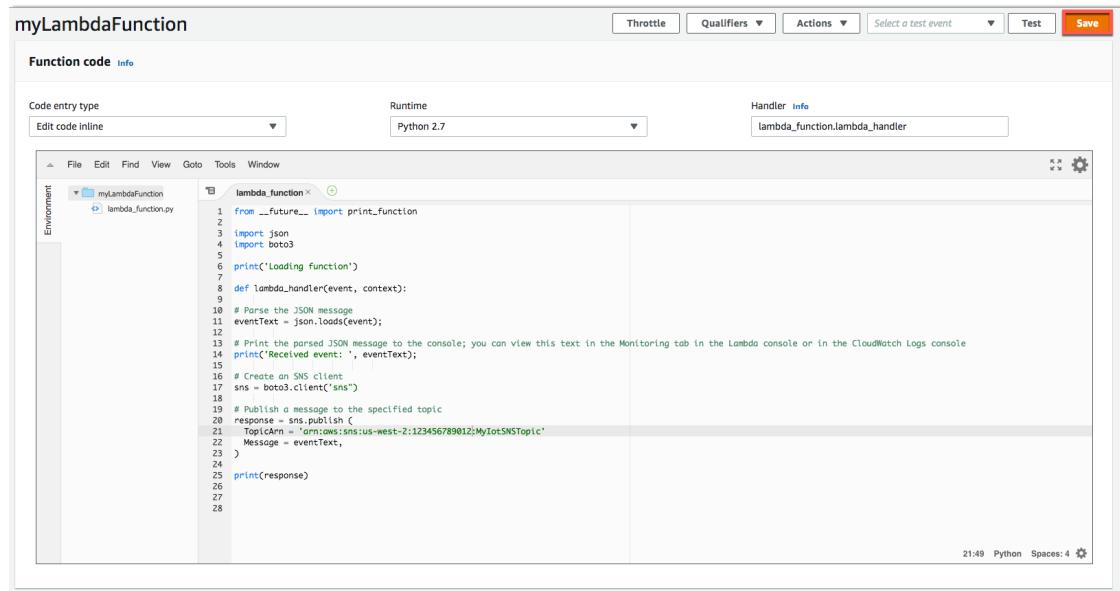
    # Publish a message to the specified topic
    response = sns.publish (
        TopicArn = 'arn:aws:iam::123456789012:role/service-role/myLambdaFunctionRole',
        Message = eventText
    )

    print(response)
```

### Note

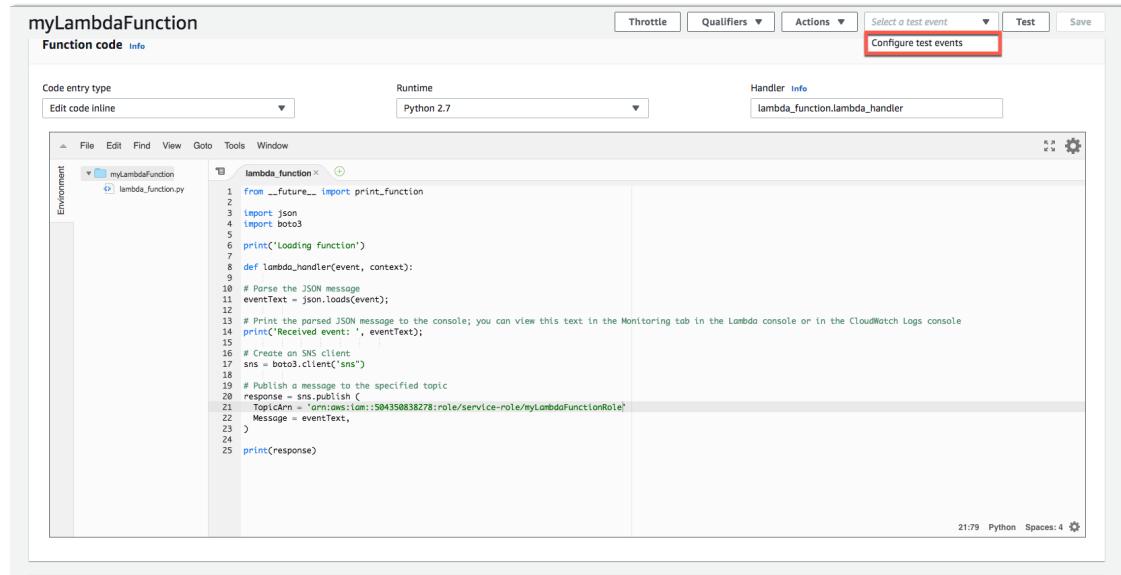
Replace the value of `TopicArn` with the ARN of the Amazon SNS topic you created earlier.

### Choose Save.



## Test Your Lambda Function

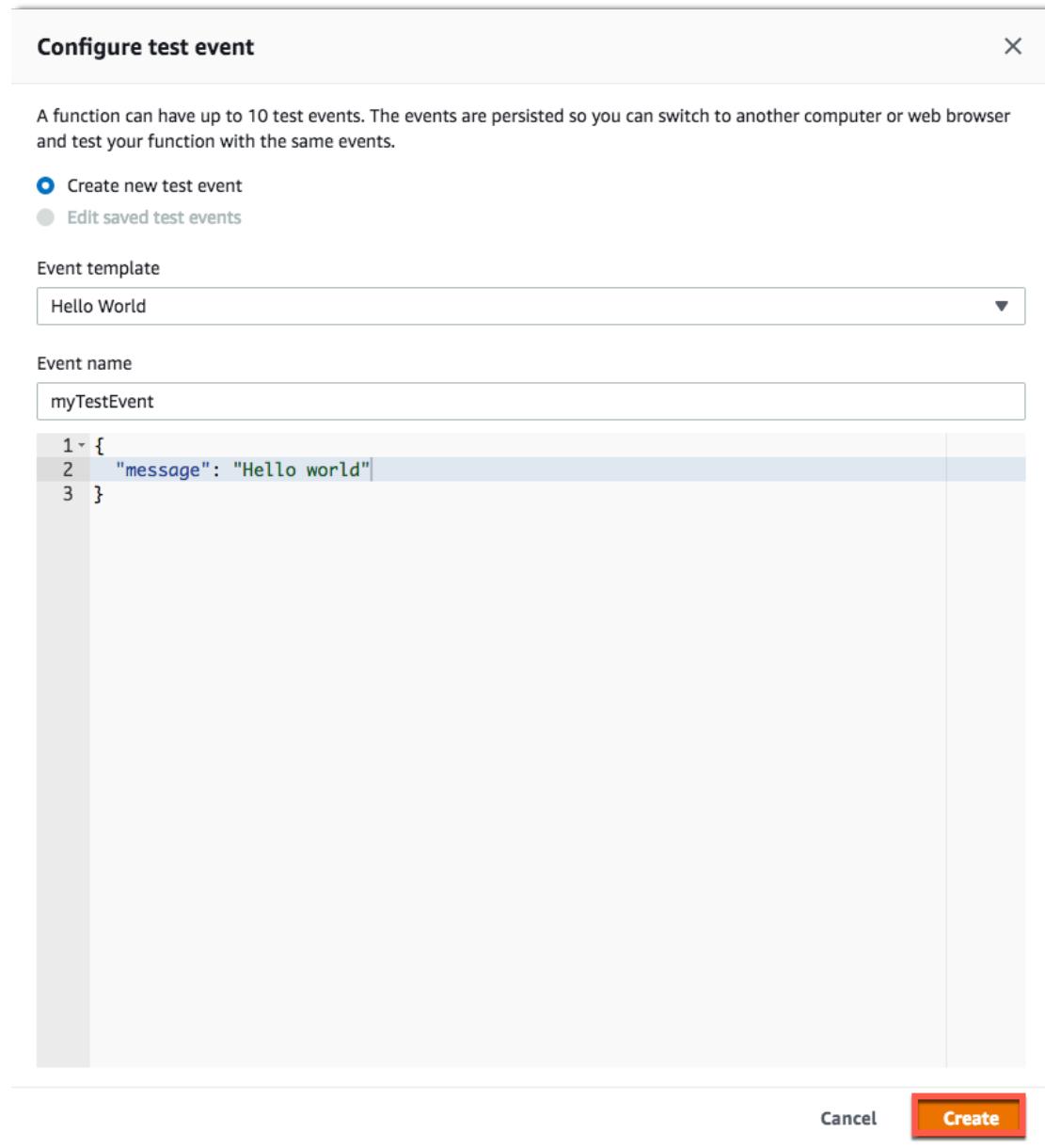
1. In the upper right of the Lambda function detail page, from **Select a test event**, choose **Configure test events**.



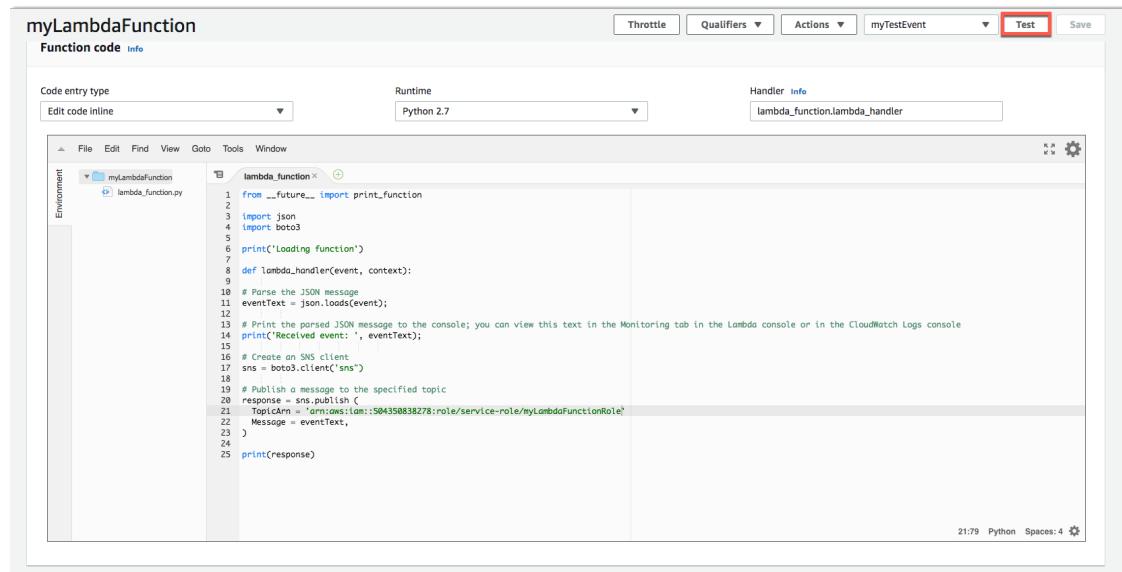
2. On **Configure test event**, enter a name for your test event and replace the message JSON with the following:

```
{  
  "message" : "Hello, world"  
}
```

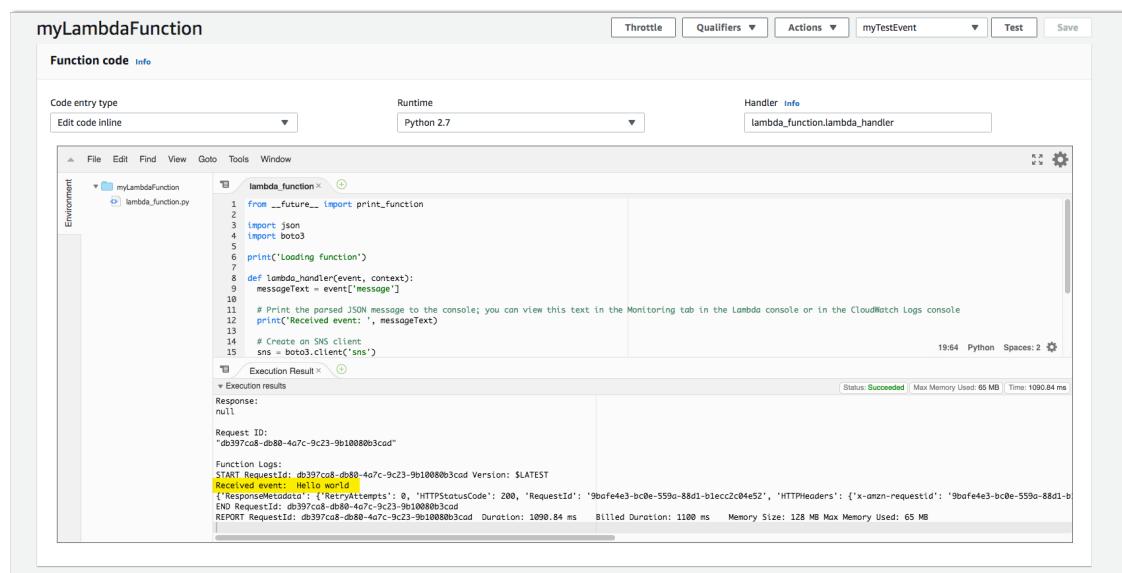
Choose **Create**.



3. In the upper right of the Lambda function detail page, choose **Test** to test your Lambda function with the message you specified in the test event.



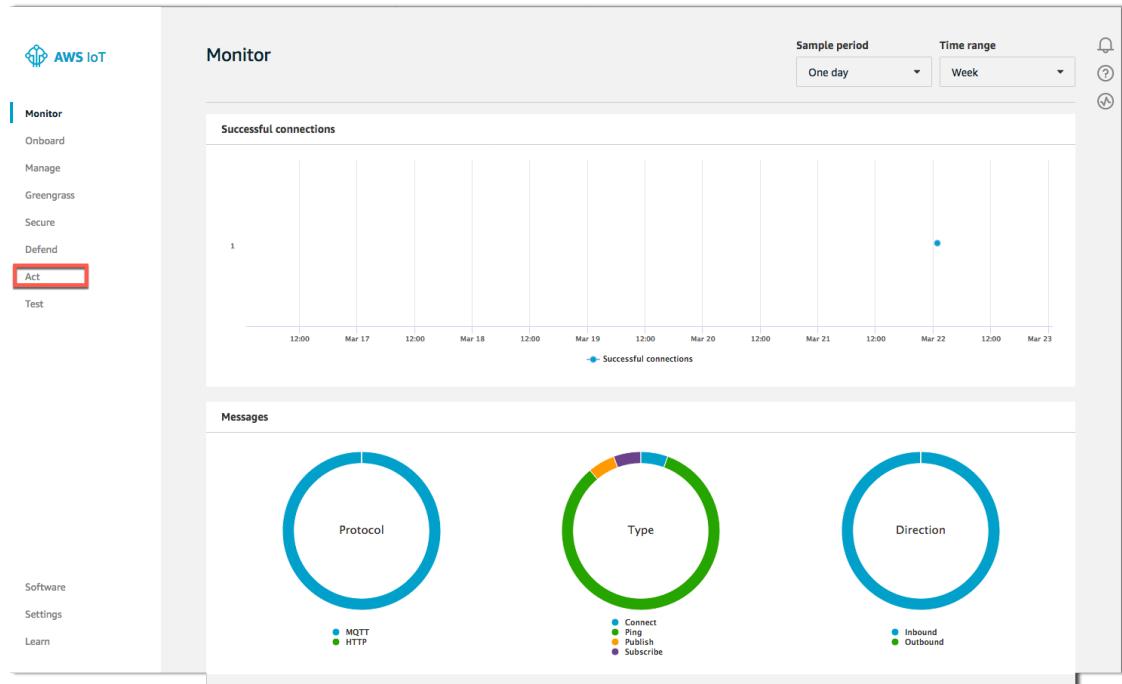
- Under your Lambda function code, on the **Execution result** tab, you see the output from the Lambda function.



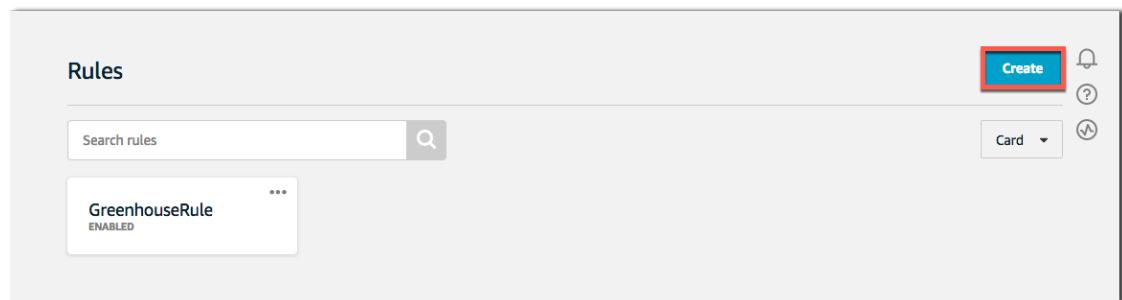
## Create a Lambda Rule

This section provides steps for creating a rule with a Lambda action and an error action. The Lambda action calls your Lambda function. If an error occurs when calling the Lambda function, the error action publishes a message to the `lambda/error` MQTT topic. This is useful when you are testing the rule.

- Browse to the AWS IoT console, and from the navigation pane, choose **Act**.



2. Choose **Create** to create an AWS IoT rule.



3. On the **Create a rule** page, enter a name for your rule.

The screenshot shows the 'Create a rule' page. The top bar is blue with the title 'Create a rule'. Below it, a descriptive text states: 'Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function)'. There are two input fields: 'Name' containing 'myLambdaRule' and 'Description' which is currently empty.

4. In **Rule query statement**, enter the following query:

```
SELECT * FROM "my/lambda/topic"
```

**Rule query statement**

Indicate the source of the messages you want to process with this rule.

Using SQL version

2016-03-23 ▾

**Rule query statement**

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT * FROM "my/lambda/topic"
```

5. In **Set one or more actions**, choose **Add action**.

**Set one or more actions**

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

**Add action**

6. Under **Select an action**, choose **Send a message to a Lambda function**, and then choose **Configure action**.

Select an action

Select an action.

<input type="radio"/>	 Insert a message into a DynamoDB table DYNAMODB
<input type="radio"/>	 Split message into multiple columns of a DynamoDB table (DynamoDBv2) DYNAMODBV2
<input checked="" type="radio"/>	 Send a message to a Lambda function LAMBDA
<input type="radio"/>	 Send a message as an SNS push notification SNS
<input type="radio"/>	 Send a message to an SQS queue SQS
<input type="radio"/>	 Send a message to an Amazon Kinesis Stream AMAZON KINESIS
<input type="radio"/>	 Republish a message to an AWS IoT topic AWS IOT REPUBLISH
<input type="radio"/>	 Store a message in an Amazon S3 bucket S3
<input type="radio"/>	 Send a message to an Amazon Kinesis Firehose stream AMAZON KINESIS FIREHOSE
<input type="radio"/>	 Send message data to CloudWatch CLOUDWATCH METRICS
<input type="radio"/>	 Change the state of a CloudWatch alarm CLOUDWATCH ALARMS
<input type="radio"/>	 Send a message to the Amazon Elasticsearch Service AMAZON ELASTICSEARCH
<input type="radio"/>	 Send a message to a Salesforce IoT Input Stream SALESFORCE IOT
<input type="radio"/>	 Send a message to an IoT Analytics Channel IOT ANALYTICS
<input type="radio"/>	 Start a Step Functions state machine execution STEP FUNCTIONS

[Cancel](#) Configure action

7. On **Configure action**, choose **Select**.

### Configure action

 Send a message to a Lambda function  
LAMBDA

We'll set [the permissions](#) on the Lambda function for you. [Create a new Lambda function](#)

\*Function name

No lambda function selected Select

[Cancel](#) Add action

8. Choose your Lambda function.

### Configure action

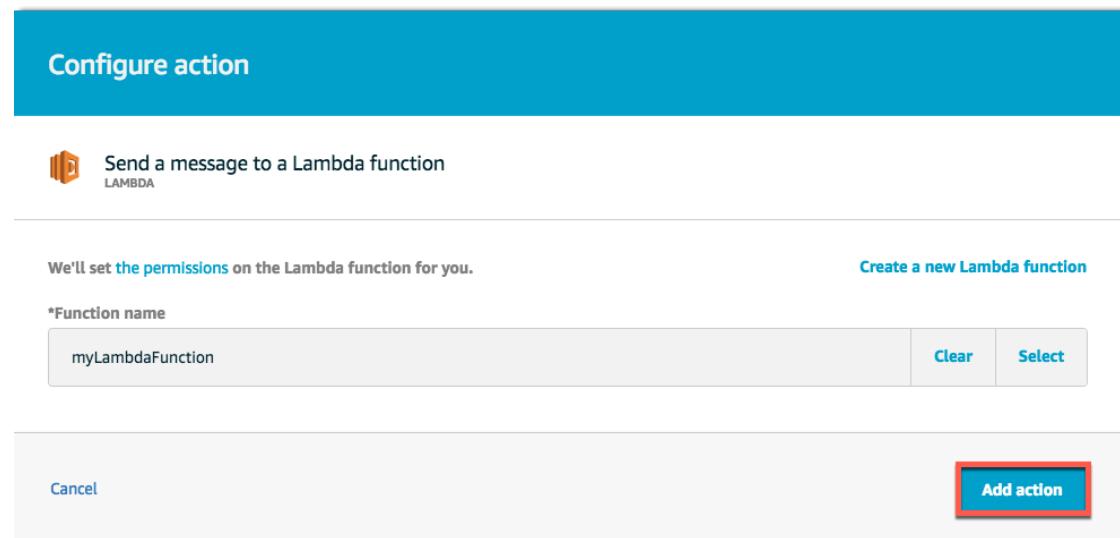
 Send a message to a Lambda function  
LAMBDA

We'll set [the permissions](#) on the Lambda function for you. [Create a new Lambda function](#)

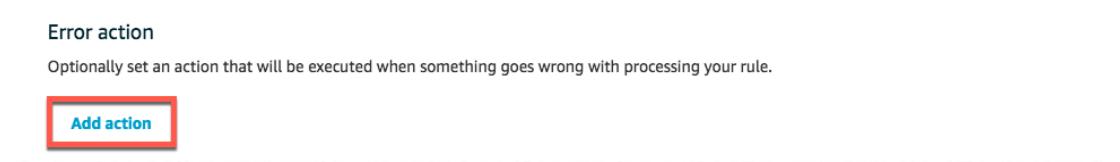
\*Function name

No lambda function selected	<a href="#">Refresh</a>	<a href="#">Close</a>
<input type="text" value="Search for lambda functions"/>		
myLambdaFunction <span style="border: 2px solid red; padding: 2px;">Select</span>		
<a href="#">Cancel</a> <span style="background-color: #0070C0; color: white; padding: 5px 10px;">Add action</span>		

9. Choose **Add action**.



10. On the **Create a rule** page, in **Error action**, choose **Add action**.



11. On the **Set an action as error action** page, choose **Republish a message to an AWS IoT topic**, and then choose **Configure action**.

### Set an action as error action

Set an action as error action.

-  Insert a message into a DynamoDB table  
DYNAMODB
-  Split message into multiple columns of a DynamoDB table (DynamoDBv2)  
DYNAMODBV2
-  Send a message to a Lambda function  
LAMBDA
-  Send a message as an SNS push notification  
SNS
-  Send a message to an SQS queue  
SQS
-  Send a message to an Amazon Kinesis Stream  
AMAZON KINESIS
-  Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH
-  Store a message in an Amazon S3 bucket  
S3
-  Send a message to an Amazon Kinesis Firehose stream  
AMAZON KINESIS FIREHOSE
-  Send message data to CloudWatch  
CLOUDWATCH METRICS
-  Change the state of a CloudWatch alarm  
CLOUDWATCH ALARMS
-  Send a message to the Amazon Elasticsearch Service  
AMAZON ELASTICSEARCH
-  Send a message to a Salesforce IoT Input Stream  
SALESFORCE IOT
-  Send a message to an IoT Analytics Channel  
IOT ANALYTICS
-  Start a Step Functions state machine execution  
STEP FUNCTIONS

[Cancel](#) Configure action

12. On the **Configure action** page, under **Topic**, enter `lambda/error`.

### Configure action

 Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

\*Topic [?](#)

Choose or create a role to grant AWS IoT access to perform this action.

No role selected [Create Role](#) [Select](#)

[Cancel](#) [Add action](#)

13. Under **Choose or create a role to grant AWS IoT access to perform this action**, choose **Create Role**.

### Configure action

 Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

\*Topic [?](#)

Choose or create a role to grant AWS IoT access to perform this action.

No role selected [Create Role](#) [Select](#)

[Cancel](#) [Add action](#)

14. In **Create a new role**, enter a name for the role, and then choose **Create role**.

## Create a new role

A new IAM role will be created in your account. An inline policy will be attached to the role providing scoped-down permissions allowing AWS IoT to access resources on your behalf.

Name

myLambdaErrorActionRole

Cancel Create role

15. In **Configure action**, choose **Add action**.

## Configure action

 Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH

This action will republish the message to another AWS IoT topic.

\*Topic [?](#)

lambda/error

Choose or create a role to grant AWS IoT access to perform this action.

myLambdaErrorActionRolePolicy Attached[Create Role](#)[Select](#)

Cancel Add action

16. In **Create a rule**, choose **Create rule**.

## Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name

Description

### Rule query statement

Indicate the source of the messages you want to process with this rule.

Using SQL version

#### Rule query statement

SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT * FROM "my/lambda/topic"
```

### Set one or more actions

Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)



Send a message to a Lambda function  
myLambdaFunction

[Remove](#) [Edit](#)

[Add action](#)

### Error action

Optionally set an action that will be executed when something goes wrong with processing your rule.



Republish a message to an AWS IoT topic  
lambda/error

[Remove](#) [Edit](#)

### Tags

Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. [Learn more](#) about tagging your AWS resources.

Tag name

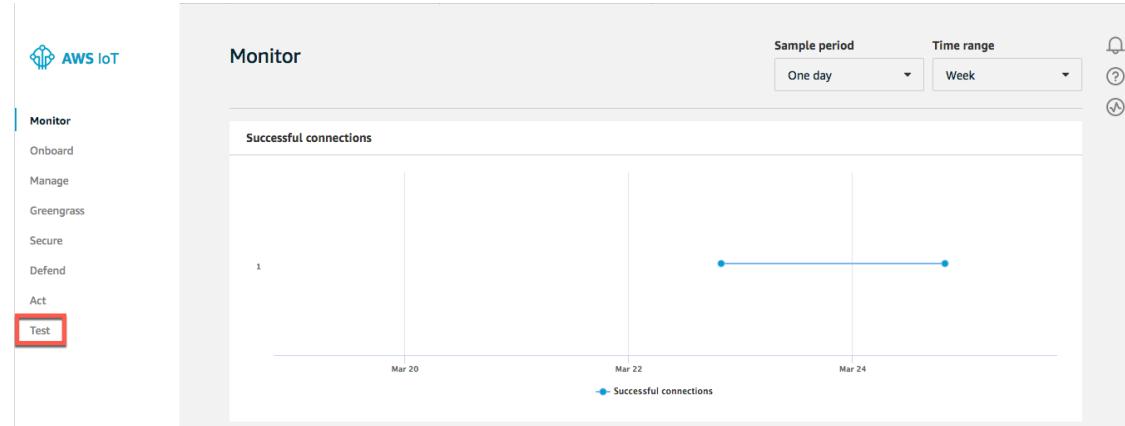
Value

[Clear](#)

[Add another](#)

## Test Your Lambda Rule

1. To test your Lambda rule, open the AWS IoT console, and from the navigation pane, choose **Test**.

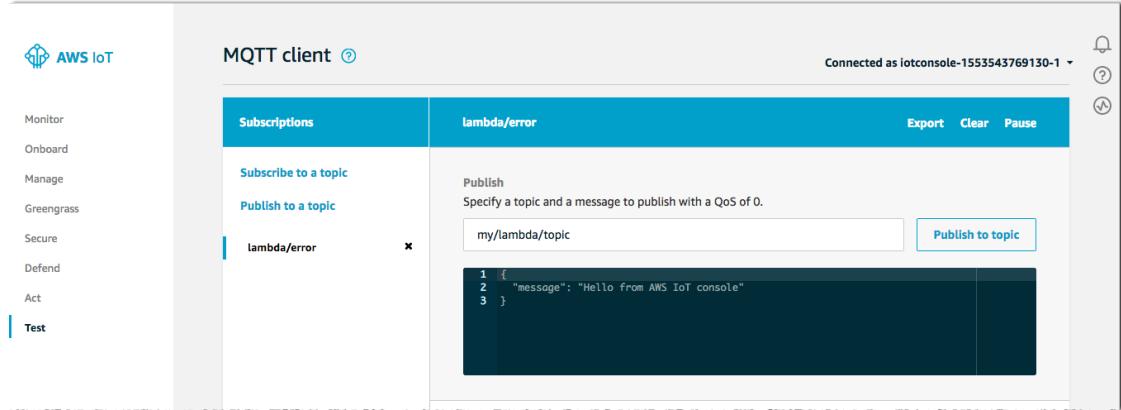


2. In the MQTT client, under **Subscription topic**, enter `lambda/error`, and then choose **Subscribe to topic**.

The screenshot shows the AWS IoT MQTT client interface. The left sidebar has 'Subscriptions' and 'Publish to a topic'. The main area has 'Subscribe to a topic' with a 'lambda/error' input field and a 'Subscribe to topic' button (highlighted with a red box). Below it is 'Max message capture' set to 100. Under 'Quality of Service', radio buttons show '0 - This client will not acknowledge to the Device Gateway that messages are received' (selected) and '1 - This client will acknowledge to the Device Gateway that messages are received'. Under 'MQTT payload display', radio buttons show 'Auto-format JSON payloads (improves readability)' (selected) and 'Display payloads as strings (more accurate)' and 'Display raw payloads (in hexadecimal)'. The bottom section is titled 'Publish' with a note 'Specify a topic and a message to publish with a QoS of 0.' It has a 'Specify a topic to publish to, e.g. myTopic/1' input field and a 'Publish to topic' button. A code editor window shows the following JSON message:

```
1 {
2   "message": "Hello from AWS IoT console"
3 }
```

3. Under **Publish**, enter `my/lambda/topic`, and then choose **Publish to topic** to publish the default JSON message.



Publishing this message should trigger the rule and call your Lambda function. Your Lambda function pushes an Amazon SNS message to a phone number subscribed to your Amazon SNS topic. If you do not get a text message, in the MQTT client, check to see if any messages were published to `lambda/error`.

## Troubleshooting Lambda Rules

If your Lambda function is called, but you do not receive a text message, make sure your phone number is subscribed to your Amazon SNS topic. If your phone number is subscribed, check the CloudWatch logs for your Lambda function. AWS Lambda writes logs to CloudWatch, which makes it possible for you to see output from your Lambda function.

### To view CloudWatch Logs

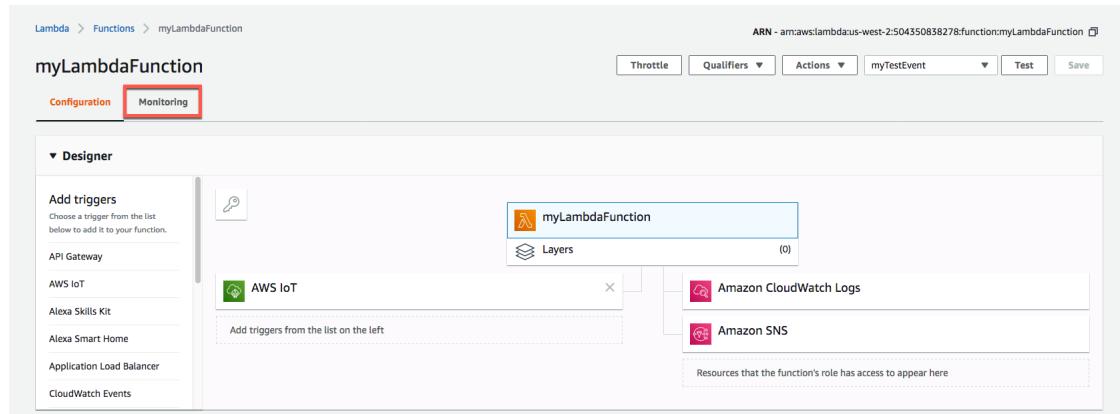
1. In the Lambda console, from the navigation pane, choose **Functions**.



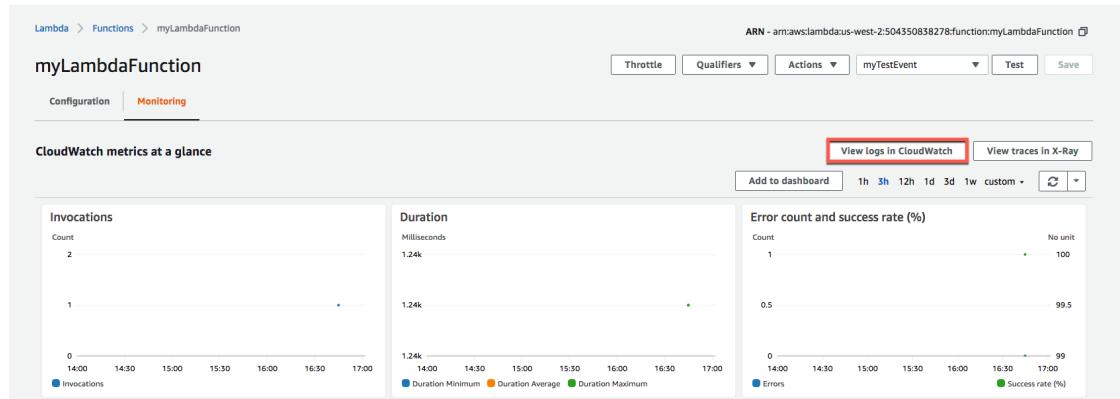
2. Choose your Lambda function.



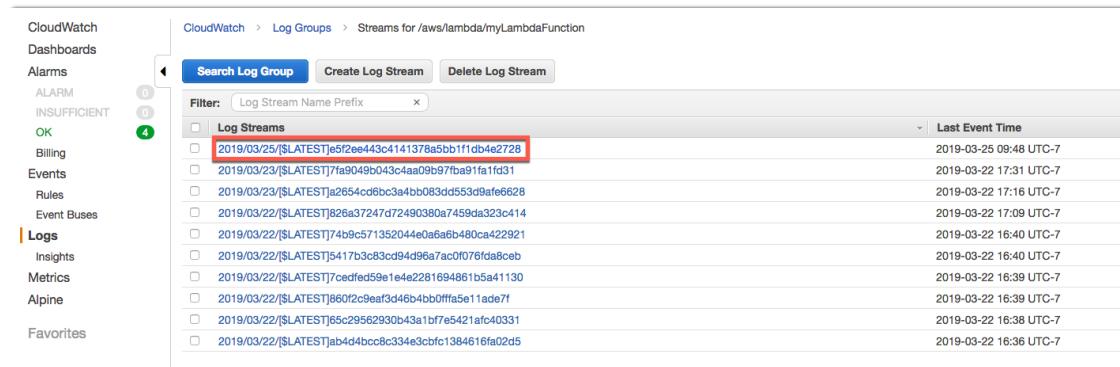
3. On the Lambda function detail page, choose the **Monitoring** tab.



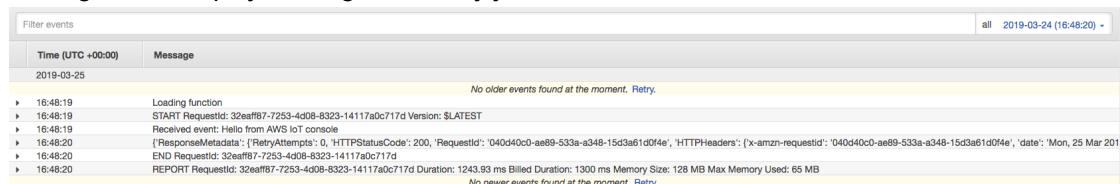
#### 4. Choose View logs in CloudWatch.



#### 5. Choose the latest log stream.



#### 6. The log stream displays the logs written by your Lambda function.



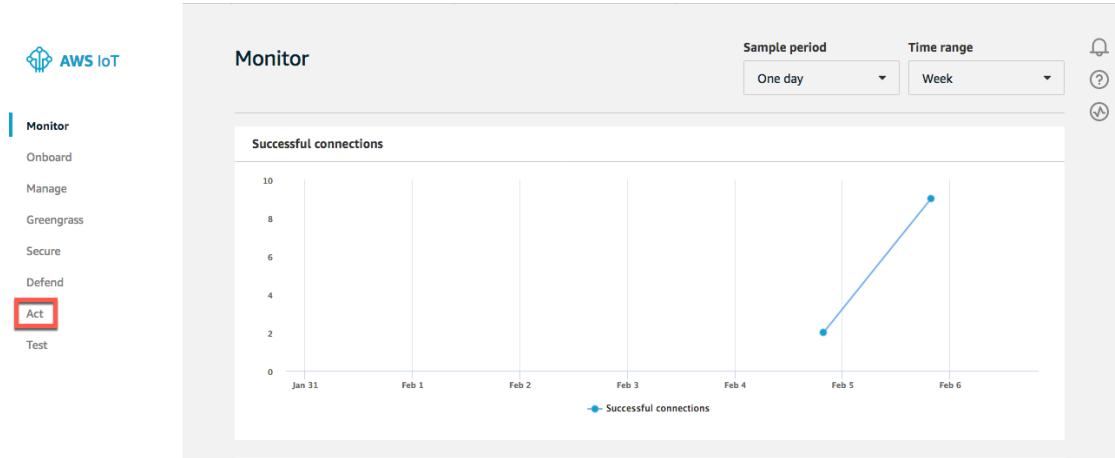
## Creating an Amazon SNS Rule

You can define a rule that sends message data to an Amazon SNS topic.

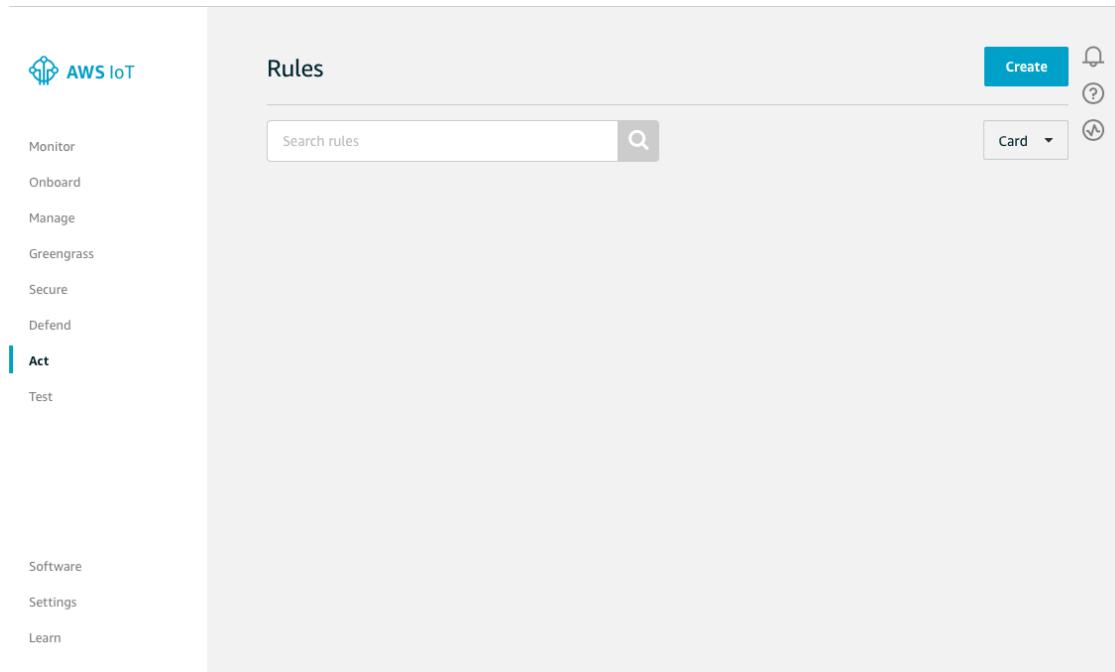
In this tutorial, you create a rule that sends the name of the AWS IoT thing that triggered the rule to all subscribers of an Amazon SNS topic.

### To create a rule with an SNS action

1. In the [AWS IoT console](#), in the navigation pane, choose **Act**.



2. On the **Rules** page, choose **Create**.



3. Enter a name and brief description for your rule.

**Note**

We do not recommend using personally identifiable information in your rule names or descriptions.

## Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

Name  
MySNSRule

Description  
A more complex SNS rule

4. In the **Rule query statement** editor, enter the following:

```
SELECT *, topic(3) as thing FROM '$aws/things/+/shadow/update/accepted'
```

(The topic filter following the "FROM" specifies the topics that trigger the rule's action when a message is published to them. The plus sign (+) used in the topic filter is a wildcard character that matches any thing name. The "topic(3)" attribute following "SELECT" appends the thing name, which is the third topic field, onto the message contents.)

Rule query statement  
Indicate the source of the messages you want to process with this rule.

Using SQL version  
2016-03-23

Rule query statement  
SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT *, topic(3) as thing FROM '$aws/things/+/shadow/update/accepted'
```

5. In **Set one or more actions**, choose **Add action**.

Set one or more actions  
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

Add action

6. Under **Select an action**, choose **Send a message as an SNS push notification**, and then choose **Configure action**.

### Select an action

Select an action.

-  Insert a message into a DynamoDB table  
DYNAMODB
-  Split message into multiple columns of a DynamoDB table (DynamoDBv2)  
DYNAMODBV2
-  Send a message to a Lambda function  
LAMBDA
-  Send a message as an SNS push notification  
SNS
-  Send a message to an SQS queue  
SQS
-  Send a message to an Amazon Kinesis Stream  
AMAZON KINESIS
-  Republish a message to an AWS IoT topic  
AWS IOT REPUBLISH
-  Store a message in an Amazon S3 bucket  
S3
-  Send a message to an Amazon Kinesis Firehose stream  
AMAZON KINESIS FIREHOSE
-  Send message data to CloudWatch  
CLOUDWATCH METRICS
-  Change the state of a CloudWatch alarm  
CLOUDWATCH ALARMS
-  Send a message to the Amazon Elasticsearch Service  
AMAZON ELASTICSEARCH
-  Send a message to a Salesforce IoT Input Stream  
SALESFORCE IOT
-  Send a message to IoT Analytics  
IOT ANALYTICS
-  Start a Step Functions state machine execution  
STEP FUNCTIONS

[Cancel](#) Configure action

7. On the **Configure action** page, for **SNS target**, choose **Create**.

**Configure action**

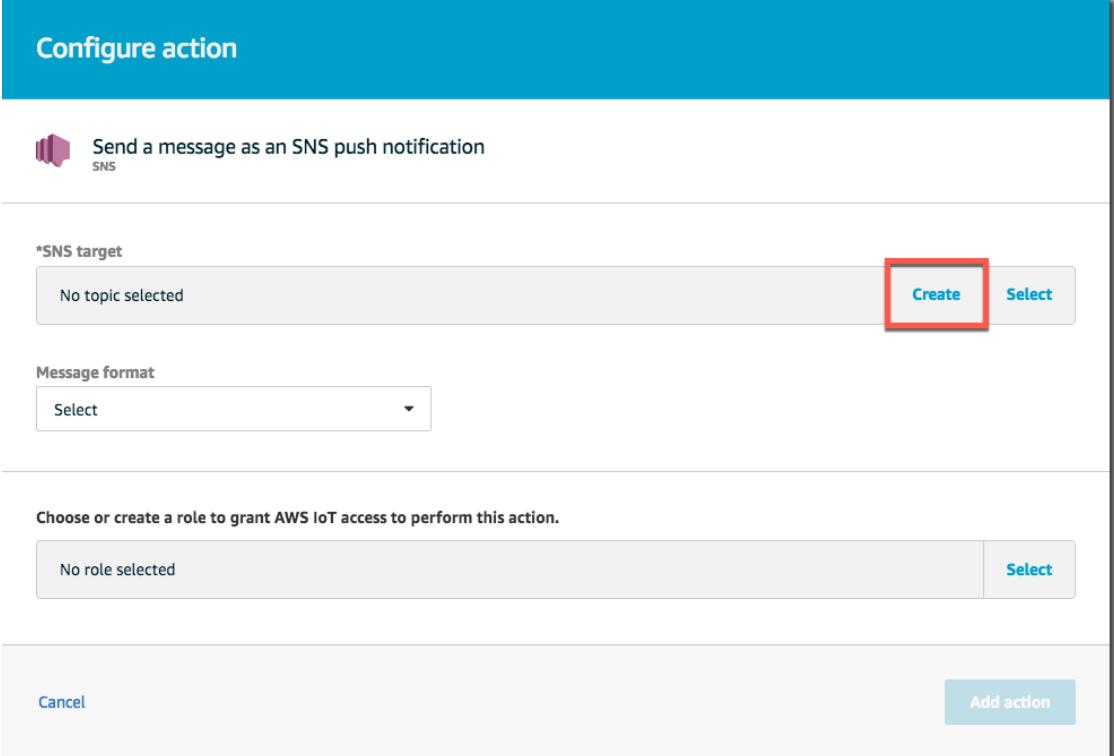
 Send a message as an SNS push notification  
SNS

\***SNS target**  
No topic selected Create Select

**Message format**  
Select

Choose or create a role to grant AWS IoT access to perform this action.  
No role selected Select

Cancel Add action

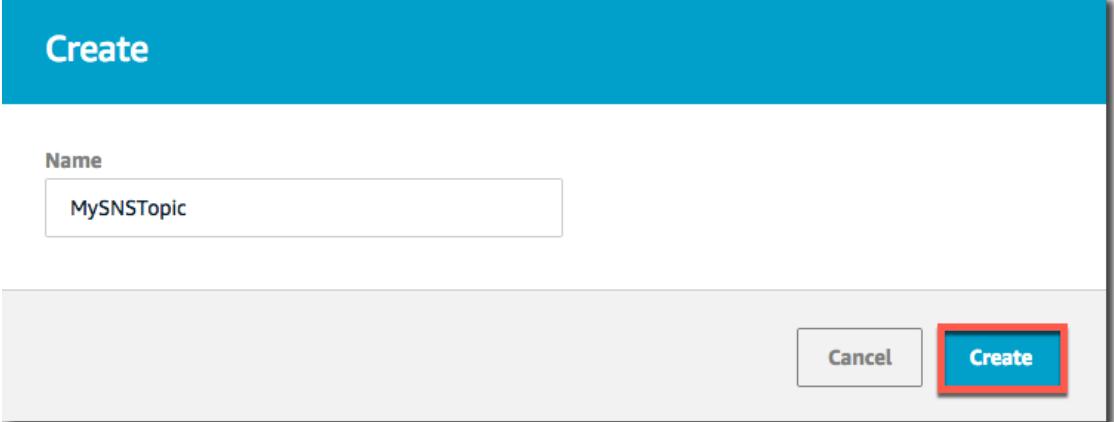


8. Enter a topic name in the dialog box that opens, and then choose **Create**.

**Create**

**Name**  
MySNSTopic

Cancel Create



9. On the **Configure action** page, for **SNS target**, choose the SNS topic you just created. For **Message format**, choose **JSON**. Under **Choose or create a role to grant AWS IoT access to perform this action**, choose **Create Role**.

### Configure action

 Send a message as an SNS push notification  
SNS

\*SNS target  
MySNSTopic [Create](#) [Clear](#) [Select](#)

Message format  
JSON ▾

Choose or create a role to grant AWS IoT access to perform this action.  
No role selected [Create Role](#) [Select](#)

[Cancel](#) [Add action](#)

10. Enter a name for the role, and then choose **Create role**.

### Create a new role

A new IAM role will be created in your account. An inline policy will be attached to the role providing scoped-down permissions allowing AWS IoT to access resources on your behalf.

Name  
MySNSRole

[Cancel](#) [Create role](#)

11. In **Configure action** choose **Add action**.

Configure action

SNS target

Send a message as an SNS push notification  
SNS

MySNSTopic      Create      Clear      Select

Message format

JSON

Choose or create a role to grant AWS IoT access to perform this action.

MySNSRole      Policy Attached      Create Role      Select

Cancel      Add action

The screenshot shows the 'Configure action' step of creating an Amazon SNS rule. At the top, there's a blue header bar with the title 'Configure action'. Below it, a section for 'SNS target' shows a purple icon and the text 'Send a message as an SNS push notification' followed by 'SNS'. A text input field contains 'MySNSTopic' with three buttons to its right: 'Create', 'Clear', and 'Select'. Underneath is a 'Message format' dropdown set to 'JSON'. The next section, 'Choose or create a role to grant AWS IoT access to perform this action.', shows a role named 'MySNSRole' with a green button indicating 'Policy Attached'. To the right are 'Create Role' and 'Select' buttons. At the bottom right of the configuration area is a large blue button with white text labeled 'Add action', which is outlined with a red box.

12. Choose **Create rule**.

## Create a rule

Create a rule to evaluate messages sent by your things and specify what to do when a message is received (for example, write data to a DynamoDB table or invoke a Lambda function).

**Name**  
mySNSRule

**Description**

---

**Rule query statement**  
Indicate the source of the messages you want to process with this rule.

**Using SQL version**  
2016-03-23

**Rule query statement**  
SELECT <Attribute> FROM <Topic Filter> WHERE <Condition>. For example: SELECT temperature FROM 'iot/topic' WHERE temperature > 50. To learn more, see [AWS IoT SQL Reference](#).

```
1 SELECT *, topic(3) as thing FROM '$aws/things/+shadow/update/accepted'
```

---

**Set one or more actions**  
Select one or more actions to happen when the above rule is matched by an inbound message. Actions define additional activities that occur when messages arrive, like storing them in a database, invoking cloud functions, or sending notifications. (\*.required)

 Send a message as an SNS push notification  
MyIoTTopic Remove Edit >

**Add action**

---

**Error action**  
Optionally set an action that will be executed when something goes wrong with processing your rule.

**Add action**

---

**Tags**  
Apply tags to your resources to help organize and identify them. A tag consists of a case-sensitive key-value pair. [Learn more](#) about tagging your AWS resources.

**Tag name**  
Provide a tag name, e.g. Manufacturer

**Value**  
Provide a tag value, e.g. Acme-Corporation

**Clear**

**Add another**

---

87

**Cancel** **Create rule**

To test the rule, add a subscription to the SNS topic you created, and update the shadow of any AWS IoT thing.

You can use the AWS IoT console to find a thing, open its detail page, and change the device's shadow. When the Device Shadow service is notified of the change, it publishes a message on `$aws/things/MySNSThing/shadow/update/accepted`. Your rule is triggered and all subscribers to your SNS topic receive a message that contains your thing's name.

# AWS IoT SDK Tutorials

The AWS IoT Device SDKs help you to connect your devices to AWS IoT easily and quickly. The AWS IoT Device SDKs include open-source libraries, developer guides with samples, and porting guides so that you can build innovative IoT products or solutions on your choice of hardware platforms.

## Important

Before going through this tutorial, please go through the [Getting Started with AWS IoT \(p. 5\)](#).

These tutorials provide step-by-step instructions for connecting your Raspberry Pi to the [Message Broker for AWS IoT \(p. 233\)](#), using with the AWS IoT Device SDK for Embedded C and the AWS IoT Device SDK for JavaScript. After following these instructions, you can connect to the AWS IoT platform and run the sample applications included with the AWS IoT Device SDKs.

## Contents

- [Prerequisites \(p. 89\)](#)
- [Create an AWS IoT Thing for Your Raspberry Pi \(p. 89\)](#)
- [Using the AWS IoT Embedded C SDK \(p. 102\)](#)
- [Using the AWS IoT Device SDK for JavaScript \(p. 105\)](#)

## Prerequisites

This tutorial requires the following:

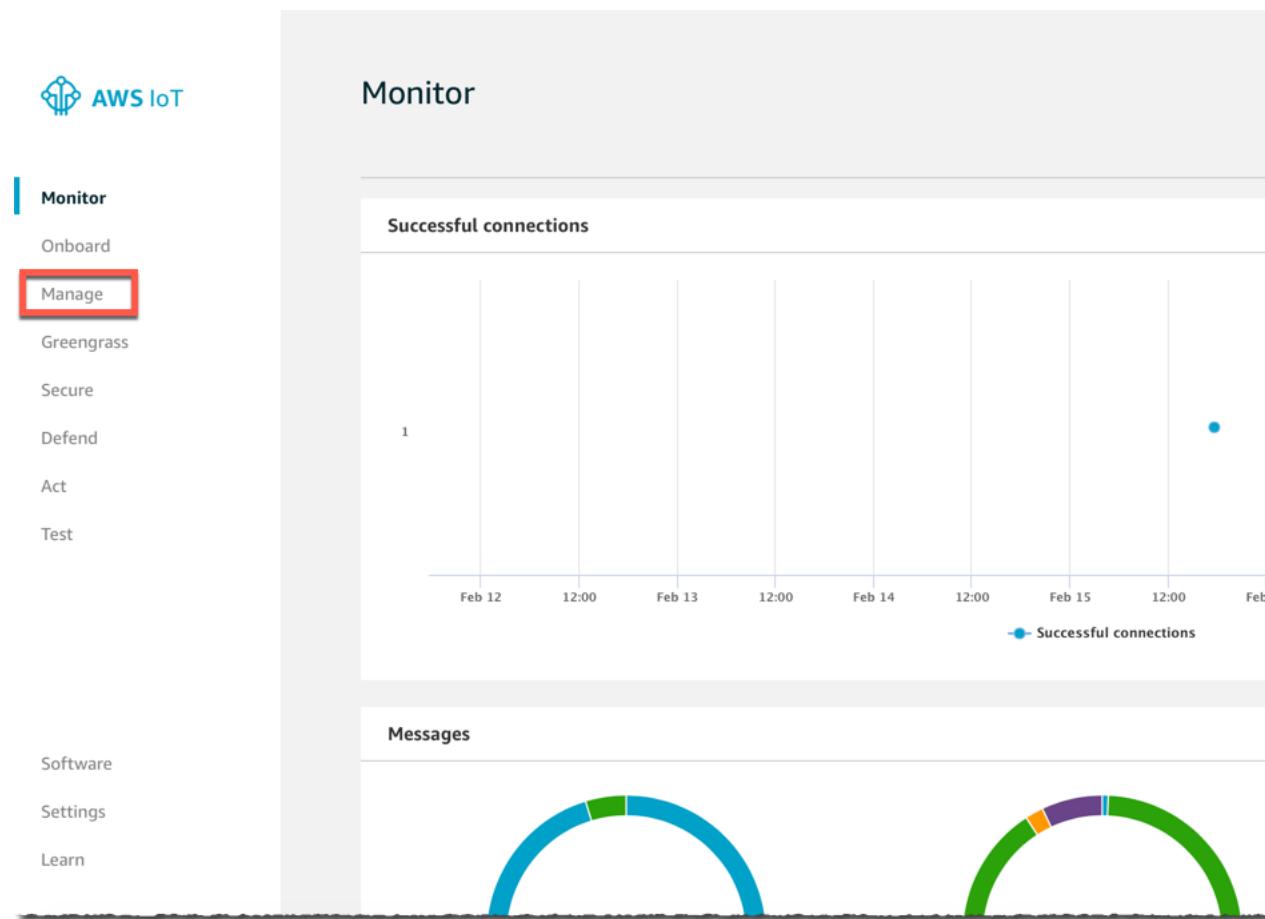
- A [Raspberry Pi 2 Model B](#), or newer.
- [Raspbian Wheezy OS](#), or newer.
- [Chromium](#) or [Iceweasel](#) web browser.
- Your Raspberry Pi must be connected to the internet using a WiFi or ethernet connection.
- An AWS account. If you don't already have an AWS account, you can get one for free by going to the [Amazon AWS Getting Started Resource Center](#).

## Create an AWS IoT Thing for Your Raspberry Pi

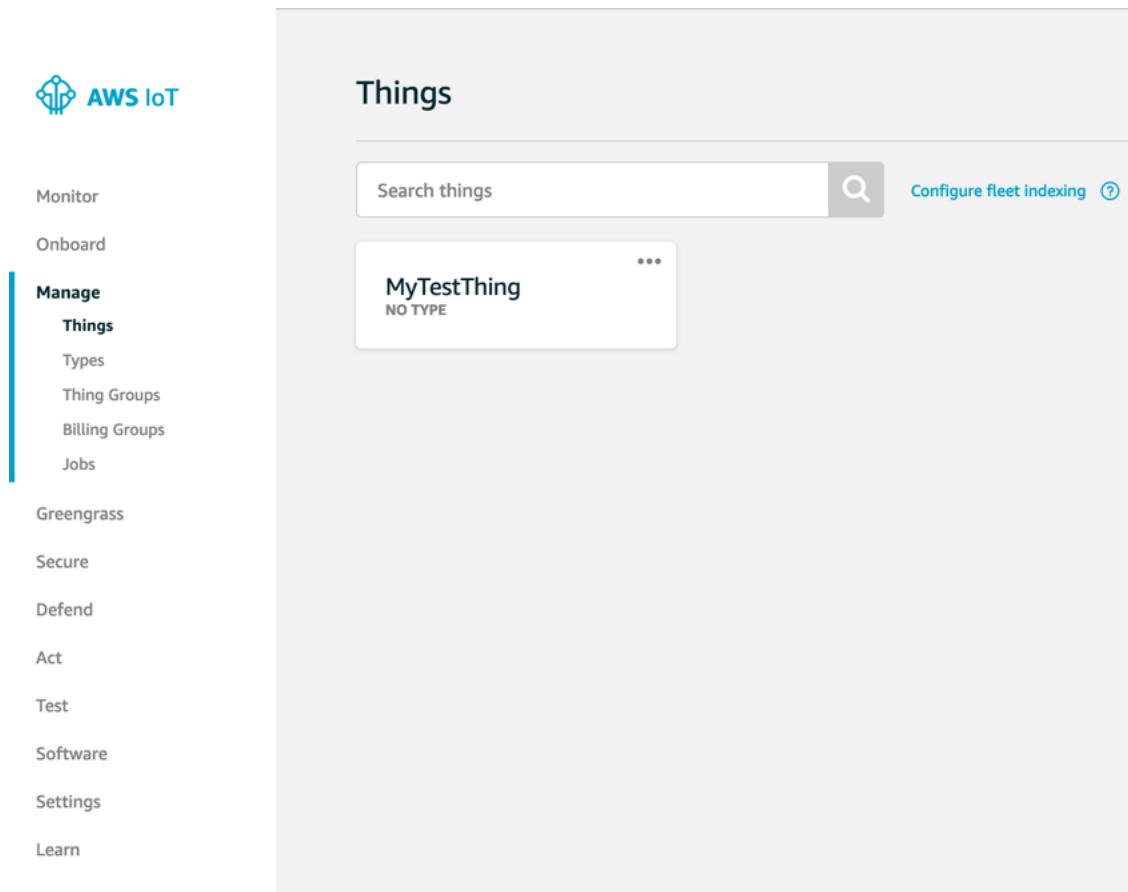
A *thing* represents a device whose status or data is stored in the AWS cloud. The device's status or data is stored in a JSON document known as the device's *shadow*. The shadow is used to both store and retrieve state information. The [Device Shadow service](#) maintains a shadow for each device that is connected to AWS IoT.

### Create an AWS IoT Thing

1. On your Raspberry Pi, open a web browser and go to the [AWS IoT console](#). You may be prompted to sign in.
2. In the [AWS IoT console](#), you see the **Monitor** page. In the navigation pane, choose **Manage**.



3. Choose **Create**.



4. On the **Creating AWS IoT things** page, choose **Create a single thing**.

## Creating AWS IoT things

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. [Learn more.](#)

### Register a single AWS IoT thing

Create a thing in your registry

[Create a single thing](#)

### Bulk register many AWS IoT things

Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

[Create many things](#)

[Cancel](#)

[Create a single thing](#)

5. On the **Add your device to the device registry** page, enter **MyRaspberryPi** for the device **Name**. Leave the default values for all the other fields, and then choose **Next**.

CREATE A THING

## Add your device to the thing registry

This step creates an entry in the thing registry and a thing shadow for your device.

Name

MyFirstThing

### Apply a type to this thing

Using a thing type simplifies device management by providing consistent registry data for things that share a type. Types provide things with a common set of attributes, which describe the identity and capabilities of your device, and a description.

Thing Type

No type selected

[Create a type](#)

### Add this thing to a group

Adding your thing to a group allows you to manage devices remotely using jobs.

Thing Group

Groups /

[Create group](#) [Change](#)

### Set searchable thing attributes (optional)

Enter a value for one or more of these attributes so that you can search for your things in the registry.

Attribute key

Provide an attribute key, e.g. Manufacturer

Value

Provide an attribute value, e.g. Acme-Corporation

[Clear](#)

[Add another](#)

[Show thing shadow ▾](#)

[Cancel](#)

[Back](#)

[Next](#)

6. On the **Add a certificate for your thing** page, choose **Create certificate**. This generates an X.509 certificate and key pair.

CREATE A THING

## Add a certificate for your thing

A certificate is used to authenticate your device's connection to AWS IoT.

### One-click certificate creation (recommended)

This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

[Create certificate](#)

### Create with CSR

Upload your own certificate signing request (CSR) based on a private key you own.

[Create with CSR](#)

### Use my certificate

Register your CA certificate and use your own certificates for one or many devices.

[Get started](#)

### Skip certificate and create thing

You will need to add a certificate to your thing later before your device can connect to AWS IoT.

[Create thing without certificate](#)

7. On the **Certificate created!** page, download your public and private keys, certificate, and root certificate authority (CA). Save them on your Raspberry Pi, you will copy them to a different directory later on in this tutorial. Choose **Activate** to activate the X.509 certificate, and then choose **Attach a policy**.

## Certificate created!

Download these files and save them in a safe place. Certificates can be retrieved at any time, but the private and public keys cannot be retrieved after you close this page.

In order to connect a device, you need to download the following:

A certificate for this thing	0488f55a11.cert.pem	<a href="#">Download</a>
A public key	0488f55a11.public.key	<a href="#">Download</a>
A private key	0488f55a11.private.key	<a href="#">Download</a>

You also need to download a root CA for AWS IoT:

A root CA for AWS IoT [Download](#)

[Activate](#)

[Cancel](#)

[Done](#)

[Attach a policy](#)

8. On the **Add a policy for your thing** page, choose **Register Thing**.

After you register your thing, you will need to create and attach a new policy to the certificate.

CREATE A THING

### Add a policy for your thing

STEP 3/3

Select a policy to attach to this certificate:

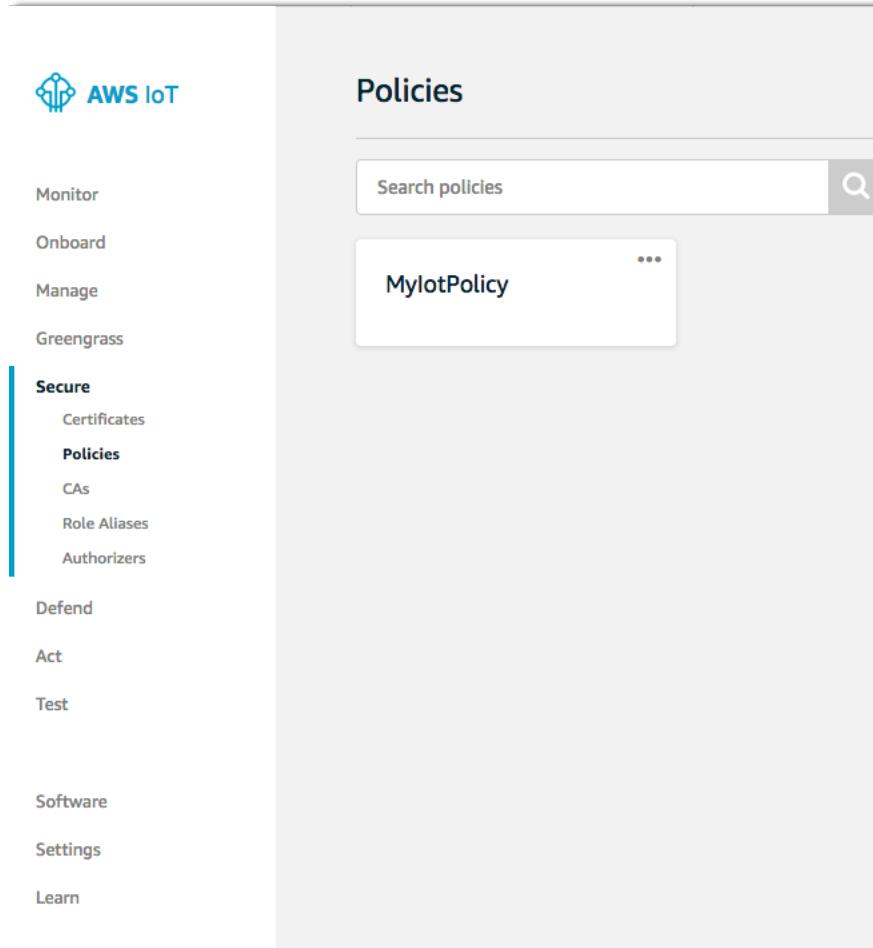
No match found  
There are no policies in your account.

[View more](#)

0 policies selected

[Register Thing](#)

9. On the AWS IoT console, in the navigation pane, choose **Secure** and **Policies**. In the **Policies** page, choose **Create a policy**.



10. On the **Create a policy** page:

- Enter a **Name** for the policy. For **Action**,
- enter **iot:\***. For **Resource ARN**, enter **\***.
- Under **Effect**, choose **Allow**, and then choose **Create**.

This policy allows your Raspberry Pi to publish messages to AWS IoT.

**Important**

These settings are overly permissive. In a production environment narrow the scope of the permissions to that which are required by your device. For more information, see [Authorization \(p. 192\)](#).

The screenshot shows the 'Create a policy' page in the AWS IoT console. At the top, there's a blue header bar with the title 'Create a policy'. Below it, a note says 'Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#)'. A 'Name' field contains 'MyRaspberryPiPolicy'. Under the heading 'Add statements', there's a note 'Policy statements define the types of actions that can be performed by a resource.' and an 'Advanced mode' link. The main statement configuration area has fields for 'Action' (containing 'iot:\*'), 'Resource ARN' (containing '\*'), 'Effect' (with 'Allow' checked and 'Deny' uncheckable), and a 'Remove' button. An 'Add statement' button is below this. At the bottom right is a large red-bordered 'Create' button.

11. In the AWS IoT console, choose **Manage** and **Things**. On the **Things** page, choose **MyRaspberryPi**.

The screenshot shows the AWS IoT Things interface. On the left, a navigation sidebar lists various categories: Monitor, Onboard, Manage (with sub-options Things, Types, Thing Groups, Billing Groups, Jobs), Greengrass, Secure, Defend, Act, Test, Software, Settings, and Learn. The 'Manage Things' option is currently selected. The main content area is titled 'Things' and contains a search bar labeled 'Search things'. A single item, 'MyRaspberryPi' (NO TYPE), is listed, and it is highlighted with a red rectangular box. A 'Create' button is located in the top right corner of the main area.

12. On the thing's **Details** page, in the left navigation pane, choose **Interact**.

The screenshot shows the AWS IoT Thing details page for a thing named 'MyRaspberryPi'. The top navigation bar includes 'THING', 'MyRaspberryPi', 'NO TYPE', and 'Actions'. On the left, a navigation pane lists 'Details', 'Security', 'Thing Groups', 'Billing Groups', 'Shadow', 'Interact' (which is highlighted with a red box), 'Activity', 'Jobs', 'Violations', and 'Defender metrics' (with a 'new' badge). The main content area shows 'Thing ARN' as 'arn:aws:iot:us-west-2:...:thing/MyRaspberryPi' and 'Type' as 'No type'. There is also a search bar and an 'Edit' link.

13. Make a note of the REST API endpoint. You will need it to connect to AWS IoT. In the navigation pane, choose **Security**.

THING  
**MyRaspberryPi**  
NO TYPE

**Actions**

Details

**Security**

Thing Groups

Billing Groups

Shadow

**Interact**

Activity

Jobs

Violations

Defender metrics **new**

This thing already appears to be connected.

**Connect a device**

**HTTPS**

Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

`[REDACTED]-ats.iot.us-west-2.amazonaws.com`

**MQTT**

Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow)

[Learn more](#)

**Update to this thing shadow**

`$aws/things/MyRaspberryPi/shadow/update`

**Update to this thing shadow was accepted**

`$aws/things/MyRaspberryPi/shadow/update/accepted`

**Update this thing shadow documents**

`$aws/things/MyRaspberryPi/shadow/update/documents`

14. Choose the certificate that you created earlier.

THING  
**MyRaspberryPi**  
NO TYPE

Actions

Details

Security

Thing Groups

Billing Groups

Shadow

Interact

Activity

Jobs

Violations

Defender metrics

**new**

Certificates

Create certificate

View other options

511058e40bda25f70...

15. On the certificate's **Details** page, in **Actions**, choose **Attach policy**.

CERTIFICATE  
**511058e40bda25f705c2f16e29479bb59c1f7264ad8e4d89396004ef6899af24**  
ACTIVE

Actions

Details

Policies

Things

Non-compliance

Certificate ARN

A certificate Amazon Resource Name (ARN) uniquely identifies this certificate. [Learn more](#)

arn:aws:iot:us-west-2:...:cert/511058e40bda25f705c2f16e29479bb59c1f7264ad8e4d89396004ef6899af24

Details

Issuer  
OU=Amazon Web Services O=Amazon.com Inc. L=Seattle ST=Washington C=US

Subject  
CN=AWS IoT Certificate

Create date  
Feb 22, 2019 3:01:24 PM -0800

Effective date  
Feb 22, 2019 2:59:24 PM -0800

Expiration date  
Dec 31, 2049 3:59:59 PM -0800

Activate

Deactivate

Revoke

Accept transfer

Reject transfer

Revoke transfer

Start transfer

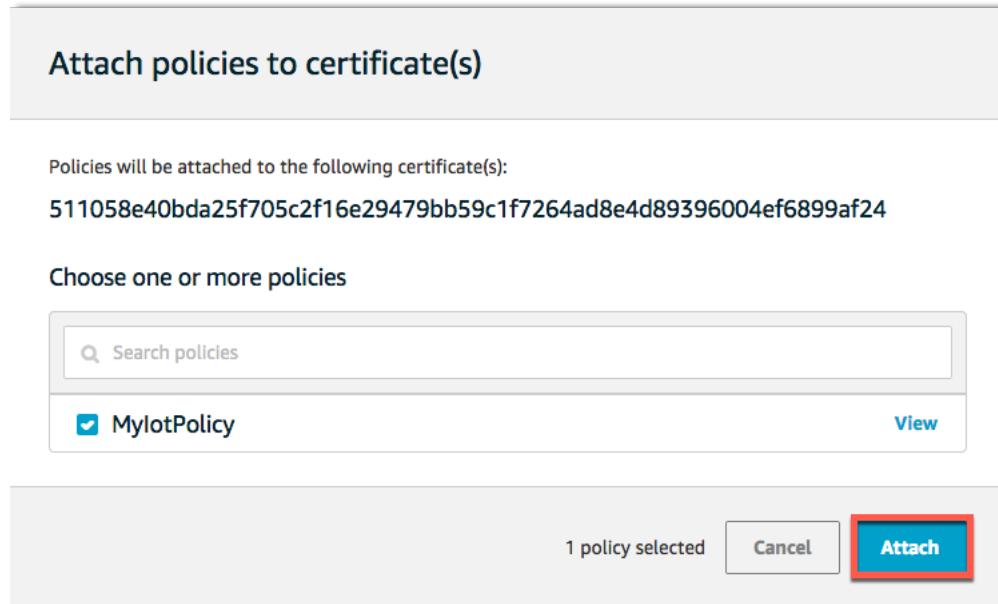
**Attach policy**

Attach thing

Download

Delete

16. On the **Attach policies to certificate(s)** page, choose the policy you created, and then choose **Attach**.



## Using the AWS IoT Embedded C SDK

### Set Up the Runtime Environment for the AWS IoT Embedded C SDK

Download the AWS IoT Embedded C SDK to your Raspberry Pi from [GitHub](#):

```
git clone https://github.com/aws/aws-iot-device-sdk-embedded-C.git -b release
```

This will create a directory called `aws-iot-device-sdk-embedded-C` in the current directory. By default you will be in your user's home directory (`/home/pi`).

Download mbed TLS to your Raspberry Pi from [GitHub](#).

#### Note

This link will bring up the potentially unstable development branch by default. We do not recommend using the development branch. For more information about officially released branches see [arm MBED](#).

Copy the contents of the mbed TLS directory into the `aws-iot-device-sdk-embedded-C/external_libs/mbedtls` directory.

### Sample App Configuration

The AWS IoT Embedded C SDK includes sample applications for you to try. For simplicity, we are going to run the `subscribe_publish_sample` application. This application illustrates how to connect to the AWS IoT Message Broker and subscribe and publish to MQTT topics.

1. Follow the instructions on [Getting Started with AWS IoT \(p. 5\)](#) to create an IoT thing, certificate, private key, and IoT policy.

2. Copy your certificate, private key, and root CA certificate into the `aws-iot-device-sdk-embedded-C/certs` directory.

**Note**

Device and root CA certificates are subject to expiration or revocation. If this should occur, you must copy a new CA certificate or private key and device certificate onto your device.

3. Navigate to the `aws-iot-device-sdk-embedded-C/samples/linux/subscribe_publish_sample` directory. You must configure your personal AWS IoT endpoint, private key, and certificate. The personal endpoint is the REST API endpoint you noted earlier. If you don't remember the endpoint and you have access to a machine with the AWS CLI installed, you can use the **aws iot describe-endpoint** command to find your personal endpoint URL. Or, go to the AWS IoT console:
  - a. Choose **Registry**.
  - b. Choose **Things**.
  - c. choose the thing that represents your Raspberry Pi. On the **Details** page for the thing, in the left navigation pane, choose **Interact**.
  - d. Copy everything, including ".com", from **REST API endpoint**.

THING

**MyRaspberryPi**

NO TYPE

**Actions**

Details	Thing ARN	Edit
Security	A thing Amazon Resource Name uniquely identifies this thing.	
Thing Groups		
Billing Groups		
Shadow		
<b>Interact</b>	<code>arn:aws:iot:us-west-2:...:thing/MyRaspberryPi</code>	
Activity		
Jobs		
Violations		
Defender metrics	<b>new</b>	

4. Open the `aws_iot_config.h` file and, in the `//Get from console` section, update the values for the following:

`AWS_IOT_MQTT_HOST`

Your personal endpoint.

`AWS_IOT_MY_THING_NAME`

Your thing name.

AWS\_IOT\_ROOT\_CA\_FILENAME

## Your root CA certificate.

AWS\_IOT\_CERTIFICATE\_FILENAME

## Your certificate.

AWS\_IOT\_PRIVATE\_KEY\_FILENAME

Your private key.

For example:

```
// Get from console
// =====
#define AWS_IOT_MQTT_HOST           "a22j5sm6o3yzc5.iot.us-east-1.amazonaws.com"
#define AWS_IOT_MQTT_PORT          8883
#define AWS_IOT_MQTT_CLIENT_ID      "MyRaspberryPi"
#define AWS_IOT_MY_THING_NAME       "MyRaspberryPi"
#define AWS_IOT_ROOT_CA_FILENAME    "root-CA.crt"
#define AWS_IOT_CERTIFICATE_FILENAME "4bbdc778b9-certificate.pem.crt"
#define AWS_IOT_PRIVATE_KEY_FILENAME "4bbdc778b9-private.pem.key"
// =====
```

# Run Sample Applications

**Run the AWS IoT Device SDK for Embedded C sample applications**

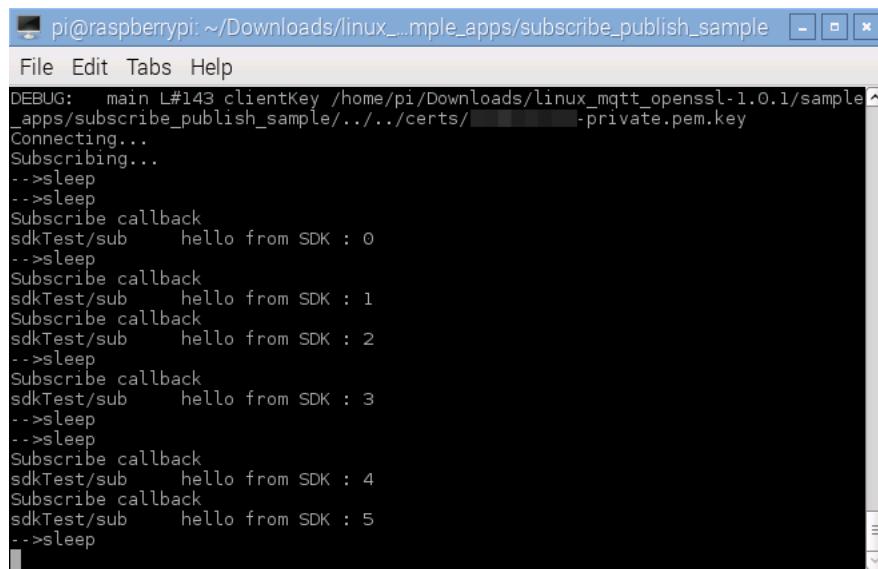
1. Compile the `subscribe publish sample app` using the included makefile.

```
make -f Makefile
```

This generates an executable file.

```
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample - □ ×
File Edit Tabs Help
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ ls
aws_iot_config.h Makefile subscribe_publish_sample.c
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ make
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $ ls
aws_iot_config.h subscribe_publish_sample
Makefile subscribe_publish_sample.c
pi@raspberrypi: ~/Downloads/linux_mqtt_openssl-1.0.1/sample_apps/subscribe_publish_sample
h_sample $
```

2. Run the `subscribe_publish_sample_app`. You should see output similar to the following:



The screenshot shows a terminal window titled "pi@raspberrypi: ~/Downloads/linux...". The window displays the output of a sample application. The log includes messages like "DEBUG: main L#143 clientKey /home/pi/Downloads/linux\_mqtt\_openssl-1.0.1/sample...\_apps/subscribe\_publish\_sample/.../certs/ -private.pem.key", "Connecting...", "Subscribing...", "Subscribe callback", "sdkTest/sub hello from SDK : 0", "Subscribe callback", "sdkTest/sub hello from SDK : 1", "Subscribe callback", "sdkTest/sub hello from SDK : 2", "Subscribe callback", "sdkTest/sub hello from SDK : 3", "Subscribe callback", "sdkTest/sub hello from SDK : 4", "Subscribe callback", "sdkTest/sub hello from SDK : 5", and several "--->sleep" entries.

Your Raspberry Pi is now connected to AWS IoT using the AWS IoT Device SDK for Embedded C.

## Using the AWS IoT Device SDK for JavaScript

This tutorial shows you how to install Node.js, the npm package manager, and the AWS IoT Device SDK for JavaScript on a Raspberry Pi and run the device SDK sample applications.

### Set Up the Runtime Environment for the AWS IoT Device SDK for JavaScript

To use the AWS IoT Device SDK for JavaScript, install Node and the npm package manager on your Raspberry Pi.

1. To add the Node repository, open a terminal and run the following command:

```
curl -sL https://deb.nodesource.com/setup_11.x | sudo -E bash -
```

2. To install Node and npm, run the following command:

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

3. To verify the installation of Node and npm, run the following commands:

```
node -v
```

and

```
npm -v
```

If a version number is displayed by these commands, node and npm are installed correctly.

## Install the AWS IoT Device SDK for JavaScript

To install the AWS IoT Device SDK for JavaScript on your Raspberry Pi, create a `~/deviceSDK` directory using the following command:

```
mkdir deviceSDK
```

Use npm to install the SDK:

```
npm install aws-iot-device-sdk
```

After the installation is complete, you should find a `node_modules` directory in your `~/deviceSDK` directory.

## Prepare to Run the Sample Applications

Follow the instructions at [Getting Started with AWS IoT \(p. 5\)](#) to register your Raspberry Pi with AWS IoT. Under `aws-device-sdk-js`, create a `certs` directory and copy your private key, certificate, and root CA files to the `certs` directory.

- Rename your private key `node-private-key.pem`.
- Rename your certificate `node-cert.pem`.

To run the AWS IoT Device SDK for JavaScript samples, you need the following information:

### Your AWS Region

You can find the region you are using by browsing to the AWS IoT console and looking at the URL. The region appears immediately after the `https://` in the URL. For example:

```
https://us-west-2.console.aws.amazon.com/iot/home?region=us-west-2#/dashboard
```

The AWS Region also appears after the `?` in the URL. For more information, see [AWS Regions and Endpoints](#). For region information specific to AWS IoT, see [AWS IoT Regions and Endpoints](#).

### A client ID

An arbitrary ID used to identify a device or application connecting to AWS IoT.

### Your private key

The fully-qualified path to your private key on your Raspberry Pi. This is the key that is generated when you registered your Raspberry Pi with AWS IoT.

### Your AWS IoT X.509 certificate

The fully-qualified path to your AWS IoT certificate on your Raspberry Pi. This is the certificate that is generated when you registered your Raspberry Pi with AWS IoT.

### The STS Amazon root CA

The fully-qualified path to the Amazon Root CA on your Raspberry Pi.

### Your AWS IoT endpoint

You can find your endpoint by running the `describe-endpoint` CLI command:

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

You can also find your endpoint by browsing to the AWS IoT console, choosing the IoT thing for your Raspberry Pi, and then choosing **Interact**. Your endpoint is displayed under **HTTPS** and **Update your Device Shadow using this Rest API endpoint**.

The port on which the AWS IoT message broker listens

This is always 8883.

Your IoT thing name

This is the name you specified when you registered your Raspberry Pi with AWS IoT.

## Run the Sample Applications

The AWS IoT Device SDK for JavaScript contains a number of samples in the `aws-iot-device-sdk-js/examples` directory. We recommend that you start with `device-example.js`. This example runs in two modes. In mode 1, it subscribes to the MQTT topic `topic_1` and publishes a message every 4 seconds on `topic_2`. In mode 2, it subscribes to `topic_2` and publishes a message every 4 seconds on `topic_1`. You can run two instances of `device-example.js`, one in mode 1 and one in mode 2, and see the messages being sent and received.

From the `aws-iot-device-sdk-js/examples` directory, run the following command to start an instance of the sample:

```
node device-example -k "../certs/node-private-key.pem" -c "../certs/node-cert.pem" -i "client-id-1" -H "<your-iot-endpoint>" -p 8883 -T "your-thing-name" --test-mode 1
```

Start another instance of `device-example.js` running in mode 2:

```
node device-example -k "../certs/node-private-key.pem" -c "../certs/node-cert.pem" -i "client-id-2" -H "<your-iot-endpoint>" -p 8883 -T "your-thing-name" --test-mode 2
```

### Important

Make sure that you use different client IDs when you run the two instances of `device-example.js`. No two clients (devices or applications) can connect to AWS IoT using the same client ID. The first client's connection is terminated and the second client connection is established.

The thing name is important only when you create a policy specific to an IoT thing. In the AWS IoT Getting Started tutorial, you do not create such a policy, so you can use the same thing name for both instances.

Your Raspberry Pi is now connected to AWS IoT using the AWS IoT SDK for JavaScript.

If the sample instances are running correctly, the output from the instance running in mode 1 should look like this:

```
substituting 250ms delay for true...
connect
message topic_1 {"mode1Process":1}
message topic_1 {"mode1Process":2}
message topic_1 {"mode1Process":3}
message topic_1 {"mode1Process":4}
...
```

The output from the instance running in mode 2 should look like this:

```
substituting 250ms delay for true...
connect
message topic_2 {"mode2Process":1}
message topic_2 {"mode2Process":2}
message topic_2 {"mode2Process":3}
message topic_2 {"mode2Process":4}
```

...

If the sample does not run correctly, try adding the `-d` option when running the sample to display debug information.

# AWS IoT Additional Tutorials

The tutorials in this section focus on using multiple AWS IoT services together to accomplish a particular task. These tutorials focus more on the integration of the services than on going deep on specific AWS IoT features. The tutorials in this section may build upon one another, demonstrating how you can start small and evolve your solutions and your needs change or grow.

Each tutorial will provide a list of prerequisites, including any specific hardware needs. Where possible, the tutorials will provide alternatives if you don't have all of the needed hardware.

## AWS IoT Plant Watering Sample

This hands-on sample demonstrates how to use AWS IoT to continually detect the current soil moisture level for a common houseplant. Whenever the moisture level gets too low, an email alert is sent to the houseplant's owner as a reminder to water it.

To get real soil moisture readings, this sample uses hardware such as a [Raspberry Pi](#) and a soil moisture sensor kit, and a common houseplant. If you don't have this hardware or the houseplant, you can simulate the soil moisture readings by generating random readings from your development computer instead. This sample shows you both approaches.

### Contents

- [Module 1: Setting Up AWS IoT and Sending Data with Your Development Computer \(p. 109\)](#)
  - [Prerequisites for Steps 1–5 \(p. 110\)](#)
  - [Step 1: Create the AWS IoT Policy \(p. 110\)](#)
  - [Step 2: Create the Thing \(p. 112\)](#)
  - [Step 3: Send and Receive Test Data for the Thing \(p. 115\)](#)
  - [Step 4: Set Up Email Alerts for Low Moisture Readings \(p. 122\)](#)
  - [Step 5: Simulate Random Moisture Levels \(p. 126\)](#)
- [Module 2: Sending Data with the Raspberry Pi \(p. 129\)](#)
  - [Prerequisites for Steps 6–12 \(p. 129\)](#)
    - [Prerequisites for Steps 6–12 \(p. 129\)](#)
  - [Step 6: Begin Preparing the microSDHC Card \(p. 131\)](#)
  - [Step 7: Download Raspbian to the microSDHC Card \(p. 132\)](#)
  - [Step 8: Finish Preparing the microSDHC card \(p. 134\)](#)
  - [Step 9: Connect to the Raspberry Pi and Set Up Raspbian \(p. 135\)](#)
  - [Step 10: Set Up the Soil Moisture Sensor Kit \(p. 139\)](#)
  - [Step 11: Capture Data from the Soil Moisture Sensor Kit \(p. 145\)](#)
  - [Step 12: Send Soil Moisture Sensor Readings to AWS IoT \(p. 145\)](#)
- [Cleaning Up \(p. 148\)](#)
- [Next Steps \(p. 152\)](#)

## Module 1: Setting Up AWS IoT and Sending Data with Your Development Computer

In the first part of this walkthrough (Steps 1–5), you set up AWS IoT to begin receiving and storing soil moisture readings coming from your development computer as a device simulator, or from a Raspberry

Pi. You then set up AWS IoT to send email alerts through Amazon Simple Notification Service (Amazon SNS) that are based on those readings.

Finally, you use your development computer to simulate soil moisture readings by generating random data. You then push those readings into AWS IoT . When the readings get too low, Amazon SNS automatically sends an email alert.

In [Module 2 \(p. 129\)](#), you can generate real soil moisture readings with a Raspberry Pi and then push those real readings into AWS IoT .

## Prerequisites for Steps 1–5

To complete the first five steps of this tutorial, you need the following:

- An AWS account
- An IAM administrator user in the AWS account. (You can use the AWS account root user instead of an IAM administrator user. However, we don't recommend it.)
- A desktop or laptop development computer to work with the [AWS IoT console](#) from a web browser, and to push simulated soil moisture readings into AWS IoT . This computer can be running a Windows, macOS, Linux, or Unix operating system. (This sample was tested with a laptop computer running Windows 10 Enterprise edition.)

## Step 1: Create the AWS IoT Policy

In this step, to allow the Raspberry Pi, or your development computer as a substitute simulator, to perform AWS IoT operations, you create an AWS IoT policy.

X.509 certificates are used to authenticate devices with AWS IoT . AWS IoT policies are used to authorize devices to perform AWS IoT operations, such as subscribing or publishing to MQTT topics.

If you use the Raspberry Pi hardware for this sample, the Raspberry Pi presents its certificate when sending messages to AWS IoT . If you use your development computer to simulate the soil moisture readings, your computer presents its certificate when sending messages to AWS IoT instead.

Later on, you attach the policy to a device certificate.

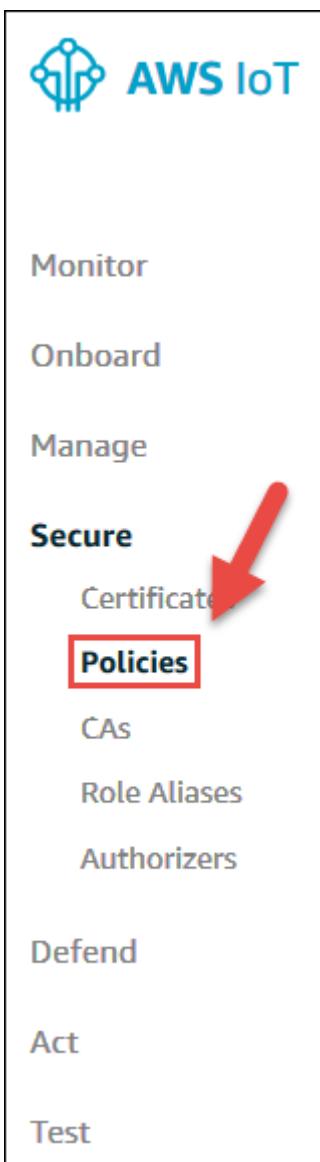
1. Use your operating system's web browser to sign in to the AWS Management Console, at <https://aws.amazon.com>.

### Note

For this sample walkthrough, we recommend that you sign in using the credentials of an [IAM administrator user](#) in your AWS account.

2. On the AWS navigation bar, choose the AWS Region where you want to create the AWS IoT resources in your AWS account. This sample was tested with the *US East (N. Virginia)* Region.
3. Open the [AWS IoT console](#). To do this, on the AWS navigation bar, choose **Services**. In the **Find a service by name or feature** box, enter **IoT Core**, and then press **Enter**.
4. In the AWS IoT console, if a **Get started** button appears, choose it.
5. In the service navigation pane, expand **Secure**, and then choose **Policies**.

6.



7. If a **You don't have any policies yet** dialog box appears, choose **Create a policy**. Otherwise, choose **Create**.
8. Provide a **Name** that represents this policy, for example **PlantWateringPolicy**.

**Note**  
If you choose to use a different name, be sure to substitute it throughout this sample.
9. For **Action**, enter **iot:\***.
10. For **Resource ARN**, replace the suggested value with an asterisk (\*).
11. For **Effect**, choose **Allow**.
12. Choose **Create**.

Create a policy

Create a policy to define a set of authorized actions. You can authorize actions on one or more resources (things, topics, topic filters). To learn more about IoT policies go to the [AWS IoT Policies documentation page](#).

Name ① PlantWateringPolicy

Add statements

Policy statements define the types of actions that can be performed by a resource. Advanced mode

Action
<span style="border: 1px solid red; padding: 2px;">② iot:*</span>

Resource ARN ③ \*

Effect ④  Allow  Deny

Remove

⑤ Create

Add statement

## Step 2: Create the Thing

In this step, you create a *thing* in AWS IoT to represent the Raspberry Pi (or your development computer as a device simulator).

Devices connected to AWS IoT are represented by things in the AWS IoT registry. The registry enables you to keep a record of all of the devices that are connected to your AWS account in AWS IoT.

1. With the [AWS IoT console](#) open, in the service navigation pane, choose **Manage**.
2. If an **Introducing AWS IoT Device Management** dialog box is displayed, choose **Show me later**, or press **Esc**.
3. In the service navigation pane, with **Manage** expanded, choose **Things**.



4. If a **You don't have any things yet** dialog box is displayed, choose **Register a thing**. Otherwise, choose **Create**.
5. On the **Creating AWS IoT things** page, for **Register a single AWS IoT thing**, choose **Create a single thing**.

## Creating AWS IoT things

An IoT thing is a representation and record of your physical device in the cloud. Any physical device needs a thing record in order to work with AWS IoT. [Learn more.](#)

**Register a single AWS IoT thing**  
Create a thing in your registry

**Bulk register many AWS IoT things**  
Create things in your registry for a large number of devices already using AWS IoT, or register devices so they are ready to connect to AWS IoT.

**Create a single thing**

**Create many things**

**Cancel**

**Create a single thing**



6. On the **Add your device to the device registry** page, provide a **Name** that represents your Raspberry Pi (or your development computer as a device simulator), for example, *MyRPi*.

**Note**

If you choose to use a different name, be sure to substitute it throughout this sample.

7. Leave the rest of this page unchanged, and then choose **Next**.
8. On the **Add a certificate for your thing** page, choose **Create certificate**.

### CREATE A THING

## Add a certificate for your thing

STEP 2/3

A certificate is used to authenticate your device's connection to AWS IoT.

**One-click certificate creation (recommended)**  
This will generate a certificate, public key, and private key using AWS IoT's certificate authority.

**Create with CSR**  
Upload your own certificate signing request (CSR) based on a private key you own.

**Create certificate**

**Get started**

**Use my certificate**  
Register your CA certificate and use your own certificates for one or many devices.

**Get started**

**Skip certificate and create thing**  
You will need to add a certificate to your thing later before your device can connect to AWS IoT.

**Create thing without certificate**



9. For **A certificate for this thing**, choose **Download**. Then follow your web browser's onscreen directions to save the file ending in `certificate.pem.crt.txt` to your local development computer.

**Note**

Although the dialog box shows a file ending in `cert.pem`, the file that you download ends in `certificate.pem.crt.txt`.

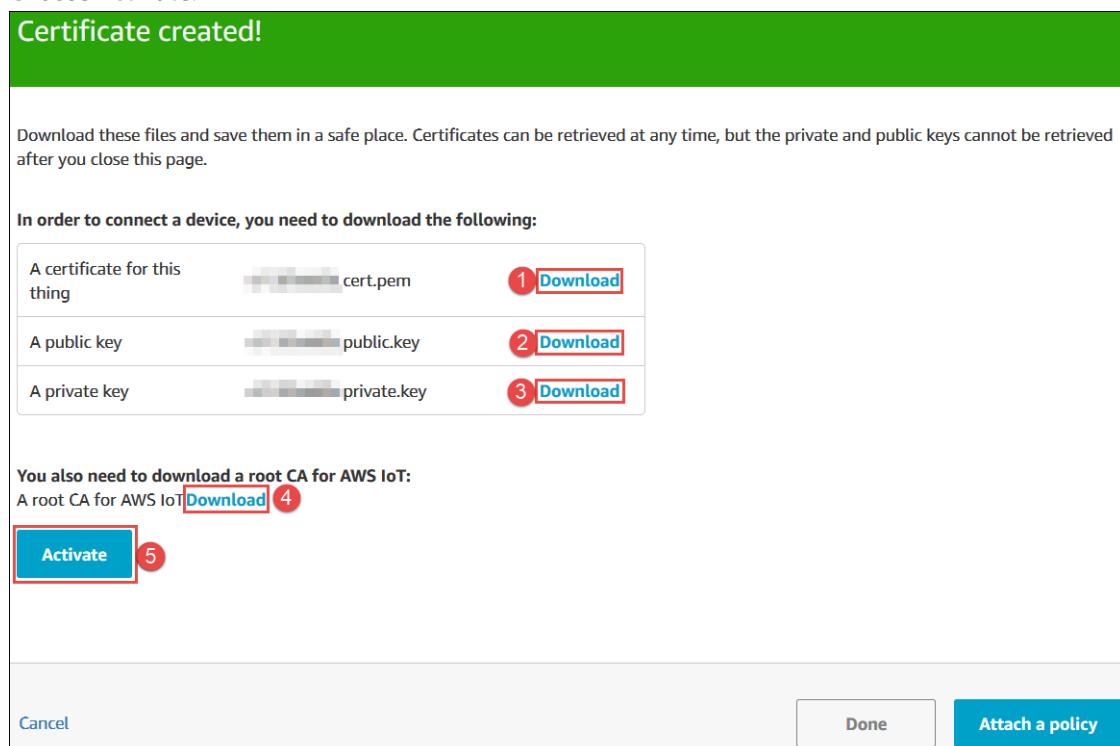
10. Repeat the previous step in this section for **A public key**, **A private key**, and **A root CA for AWS IoT**. Save the files ending in `public.pem.key`, `private.pem.key`, and `.pem`, respectively, to your development computer.

When you choose the **Download** link next to **A root CA for AWS IoT**, the **Server Authentication (p. 181)** section of the *AWS IoT Developer Guide* is displayed. From there, to get the root CA for AWS IoT, click the **Amazon Root CA 1** link in that section, which downloads the RSA 2048 bit key for the Amazon Trust Services endpoint.

**Important**

You can download the files for **A certificate for this thing** and **A root CA for AWS** at any time. However, this is your only opportunity to download the files for **A public key** and **A private key for this thing**.

11. Choose **Activate**.



12. Choose **Attach a policy**.
13. For **Add a policy for your thing**, select **PlantWateringPolicy (0 policies selected changes to 1 policy selected)**. Then choose **Register Thing**.
14. If an **Introducing AWS IoT Device Management** dialog box is displayed again, choose **Show me later**, or press **Esc**.

## Step 3: Send and Receive Test Data for the Thing

In this step, you practice sending test data to the device shadow for the Raspberry Pi (or your development computer as a device simulator). In a later step, you either send real data from the soil moisture sensor kit through the Raspberry Pi to its device shadow, or you send simulated data from your development computer to its device shadow.

A device's shadow is a JSON document, stored in AWS IoT , that AWS IoT uses to save and retrieve current state information for a device. The [Device Shadow Service for AWS IoT \(p. 321\)](#) maintains a shadow for each device you connect to AWS IoT . You can use the shadow to get and set the state of a device, regardless of whether the device is connected to the internet.

1. In the [AWS IoT console](#), on the **Things** page, choose **MyRPi**.

The screenshot shows the AWS IoT Things page. On the left, there is a sidebar with 'Monitor', 'Onboard', 'Manage', and 'Things' buttons. The 'Things' button is highlighted. On the right, a list of devices is shown, with 'MyRPi' selected. A red arrow points to the 'MyRPi' button. Below the list, there is a '...' button.

2. Choose **Interact**.

The screenshot shows the AWS IoT Thing details page for 'MyRPi'. At the top, it says 'THING MyRPi NO TYPE'. On the left, there is a sidebar with 'Details', 'Security', 'Groups', 'Shadow', 'Interact' (which is highlighted with a red box), 'Activity', 'Jobs', and 'Violations'. A red arrow points to the 'Interact' button. On the right, there is a 'Thing ARN' section with the value 'arn:aws:iot:us-east-...:thing/MyRPi' and a 'Type' section with the value 'No type'.

3. For MQTT, make a note of the value for each of the following MQTT topics, which enable you to set and get updates to the shadow:

- **Update to this device shadow** (for example, `$aws/things/MyRPi/shadow/update`)
- **Get this device shadow** (for example, `$aws/things/MyRPi/shadow/get`)
- **Get this device shadow accepted** (for example, `$aws/things/MyRPi/shadow/get/accepted`)

MyRPi

NO TYPE

Actions ▾

Details This thing already appears to be connected. Connect a device

Security

Groups

Shadow Update your Thing Shadow using this Rest API Endpoint. [Learn more](#)

Interact 1

Activity

Jobs

Violations

MQTT

Use topics to enable applications and things to get, update, or delete the state information for a Thing (Thing Shadow) [Learn more](#)

Update to this thing shadow 2 `$aws/things/MyRPi/shadow/update`

Update to this thing shadow was accepted `$aws/things/MyRPi/shadow/update/accepted`

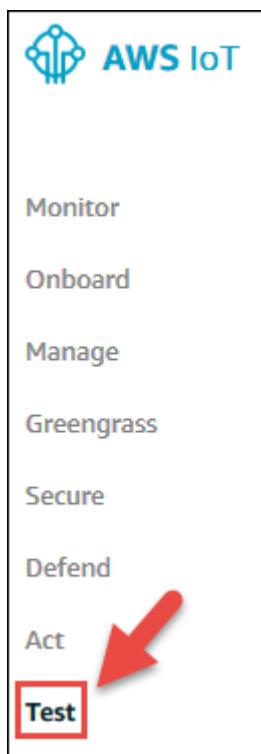
Update this thing shadow documents `$aws/things/MyRPi/shadow/update/documents`

Update to this thing shadow was rejected `$aws/things/MyRPi/shadow/update/rejected`

Get this thing shadow 3 `$aws/things/MyRPi/shadow/get`

Get this thing shadow accepted 4 `$aws/things/MyRPi/shadow/get/accepted`

4. Choose the back button.
5. If an **Introducing AWS IoT Device Management** dialog box is displayed, choose **Show me later**, or press **Esc**.
6. In the service navigation pane, choose **Test**.



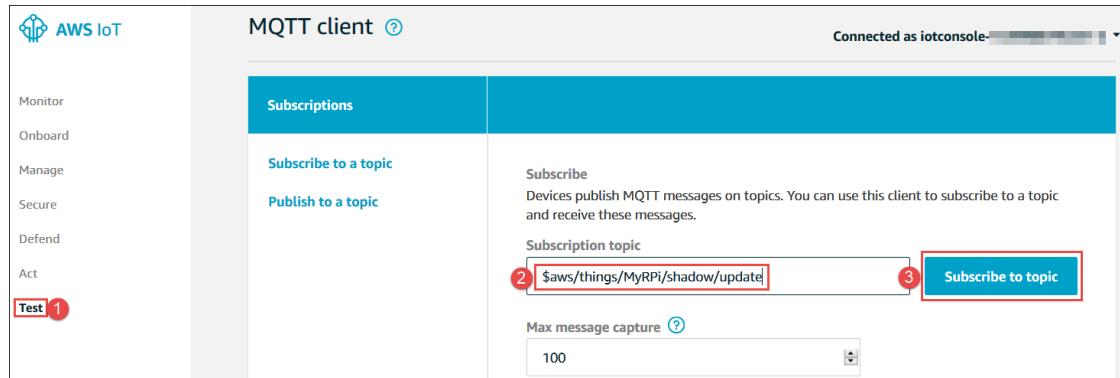
7. For **Subscription topic**, enter the MQTT topic value that you noted in step 3 of this procedure for **Update to this device shadow** (for example, `$aws/things/MyRPi/shadow/update`), and then choose **Subscribe to topic**.

The image shows the 'MQTT client' section of the AWS IoT console. On the left, there's a sidebar with the same navigation menu as the main page. In the center, under the 'Subscriptions' tab, there are two buttons: 'Subscribe to a topic' and 'Publish to a topic'. Below these buttons is a text input field labeled 'Subscription topic' containing the value '\$aws/things/MyRPi/shadow/update'. To the right of the input field is a 'Subscribe' button. At the bottom of the 'Subscriptions' section, there's a 'Max message capture' input field set to '100'. Three numbered callouts point to specific elements: 1 points to the 'Test' button in the sidebar; 2 points to the 'Subscription topic' input field; and 3 points to the 'Subscribe to topic' button.

**Important**

If you named your thing something other than MyRPi, be sure to substitute your thing's name for MyRPi in the preceding MQTT topic name and other MQTT topic names throughout Step 3. Otherwise, the subscribed MQTT topic won't display any activity.

8. Choose **Subscribe to a topic**.

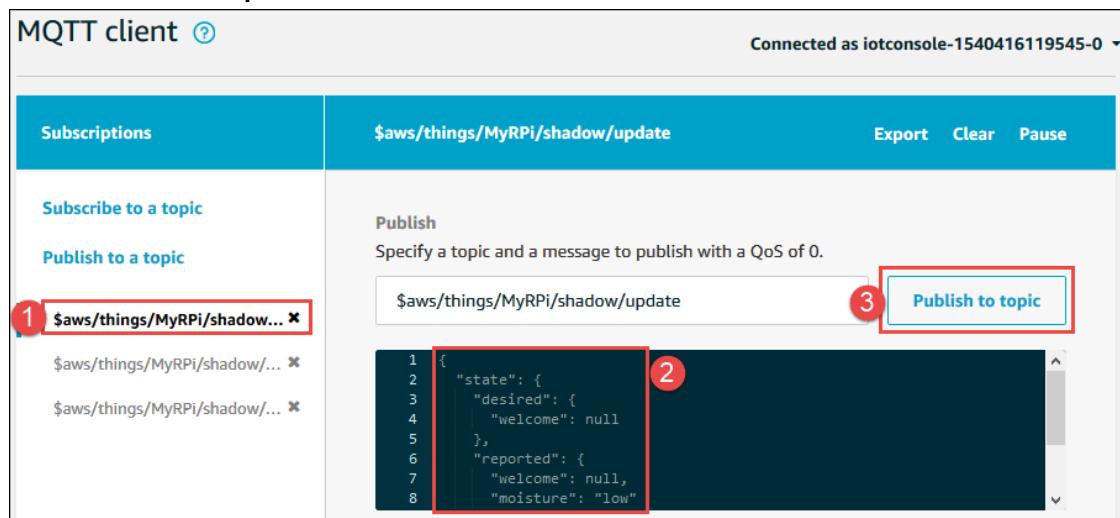


9. Repeat steps 7 and 8 in this procedure for the MQTT topic values that you noted for **Get this device shadow** (for example, `$aws/things/MyRPi/shadow/get`) and **Get this device shadow accepted** (for example, `$aws/things/MyRPi/shadow/get/accepted`).
10. Now push some test data into the shadow. To do this, in the MQTT client navigation pane, choose the MQTT topic value for **Update to this device shadow** (for example, `$aws/things/MyRPi/shadow/update`). You might need to pause your mouse over a truncated topic value to see its full value.
11. In the message payload area, replace the current payload with the following payload:

```
{
  "state": {
    "desired": {
      "welcome": null
    },
    "reported": {
      "welcome": null,
      "moisture": "low"
    }
  }
}
```

The preceding payload removes the default welcome value for the shadow and adds a moisture value with the value low to the shadow.

12. Choose **Publish to topic**.



13. To get that data from the shadow, choose the MQTT topic value for **Get this device shadow** (for example, `$aws/things/MyRPi/shadow/get`).
14. In the message payload area, replace the current payload with the following payload:

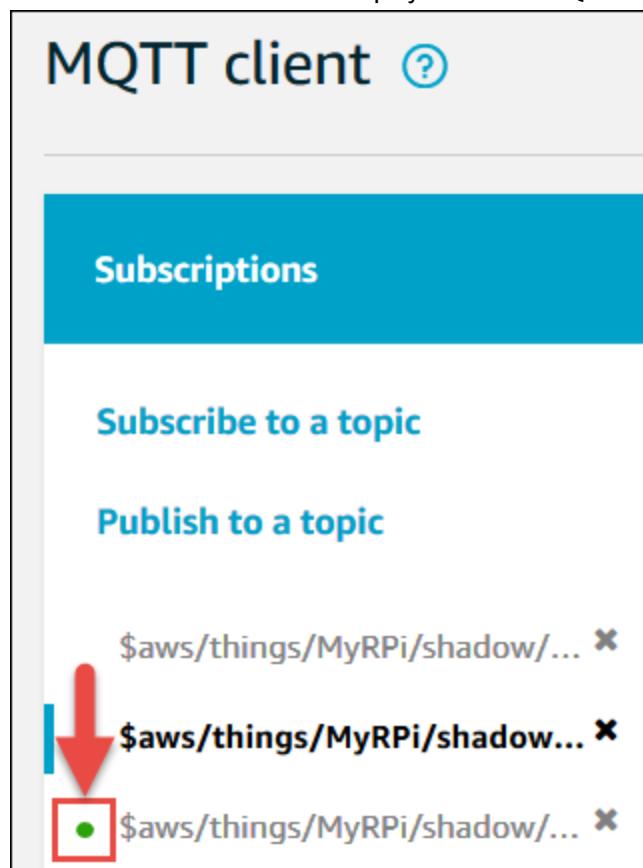
```
{ }
```

You specify empty curly braces here because the **Get this thing shadow** MQTT topic takes only an empty payload.

15. Choose **Publish to topic**.



A green dot is displayed next to the MQTT value for **Get this device shadow accepted**. This means that there is new information displayed for that MQTT topic.



16. Choose the MQTT topic value for **Get this device shadow accepted** (for example, `$aws/things/MyRPi/shadow/get/accepted`), and note the output, for example:

```
{  
    "state": {  
        "reported": {  
            "moisture": "low"  
        }  
    },  
    "metadata": {  
        "reported": {  
            "moisture": {  
                "timestamp": 1539272338  
            }  
        }  
    },  
    "version": 19,  
    "timestamp": 1539272436  
}
```

In the preceding output, the `moisture` value that was reported earlier is shown, along with the time each corresponding event happened and the current shadow document version.

17. Make another update to the shadow. To do this, in the MQTT client navigation pane, choose the MQTT topic value for **Update to this device shadow** (for example, `$aws/things/MyRPi/shadow/update`).
18. In the message payload area, replace the current payload with the following payload to change the current moisture value:

```
{  
    "state": {  
        "reported": {  
            "moisture": "okay"  
        }  
    }  
}
```

19. Choose **Publish to topic**.
20. Choose the MQTT topic value for **Get this device shadow** (for example, `$aws/things/MyRPi/shadow/get`).
21. In the message payload area, replace the current payload with the following payload:

```
{}
```

22. Choose **Publish to topic**. A green dot is displayed next to the MQTT value for **Get this device shadow accepted**.
23. Choose the MQTT topic value for **Get this device shadow accepted** (for example, `$aws/things/MyRPi/shadow/get/accepted`), and note the output, for example:

```
{  
    "state": {  
        "reported": {  
            "moisture": "okay"  
        }  
    }  
}
```

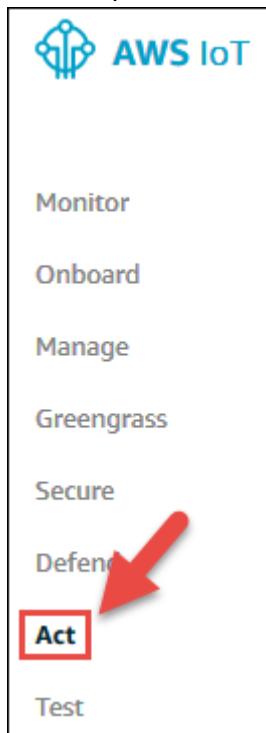
```
        },
      },
      "metadata": {
        "reported": {
          "moisture": {
            "timestamp": 1539272823
          }
        }
      },
      "version": 20,
      "timestamp": 1539272827
    }
```

In the preceding output, the `moisture` value that was just changed is shown, along with the time that each corresponding event happened and the new current shadow document version.

## Step 4: Set Up Email Alerts for Low Moisture Readings

In this step, you set up Amazon Simple Notification Service (Amazon SNS) to automatically send an email alert to the houseplant's owner as a reminder to water it whenever the soil moisture level is too low.

1. Create an AWS IoT rule to trigger the email alert through Amazon SNS. To do this, with the [AWS IoT console](#) open, in the service navigation pane, choose **Act**.



2. If a **You don't have any rules yet** dialog box appears, choose **Create a rule**. Otherwise, choose **Create**.
3. On the **Create a rule** page, enter a **Name** for this rule, for example, **MyRPiLowMoistureAlertRule**. If you use a different name, be sure to substitute it throughout this sample.
4. For **Description**, provide a meaningful description for this rule, for example, **Sends an alert whenever soil moisture level readings are too low**.

5. For **Rule query statement**, with **Using SQL version** set to **2016-03-23**, in the **Rule query statement** box, enter the following AWS IoT SQL statement as a single line, without any line breaks:

```
SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE state.reported.moisture = 'low'
```

### Rule query statement

Indicate the source of the messages you want to process with this rule.

#### Using SQL version

2016-03-23 ▾

#### Rule query statement

To learn more about constructing a SQL statement, see [AWS IoT SQL Reference](#).

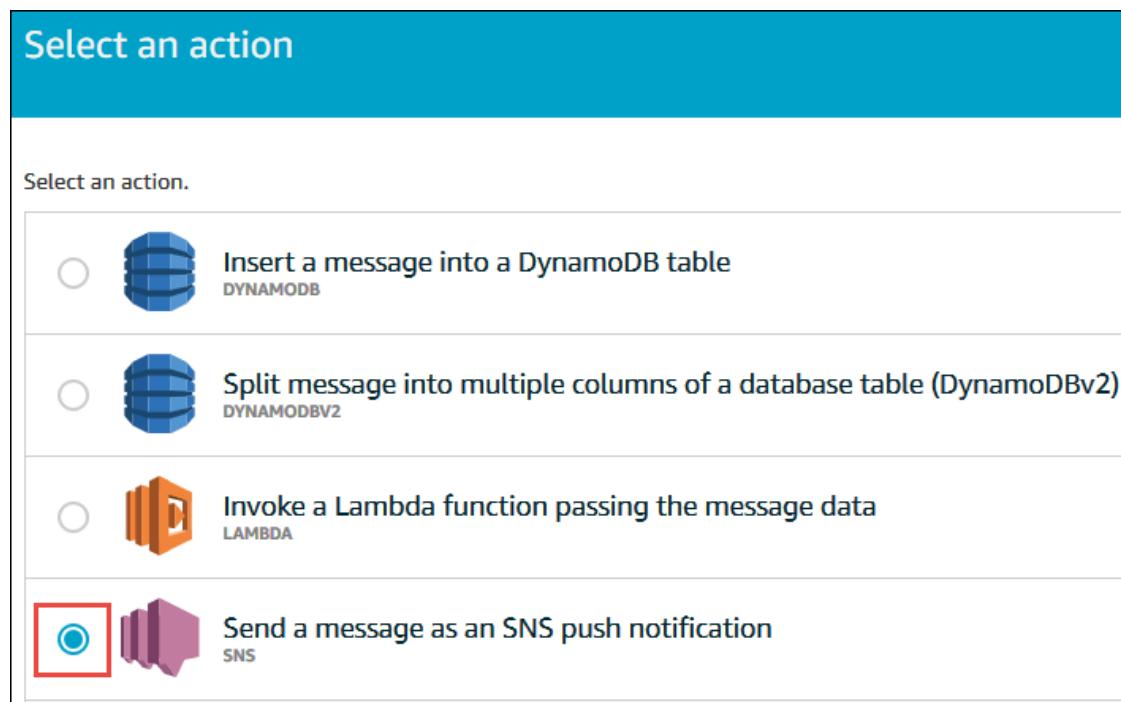
```
1 SELECT * FROM '$aws/things/MyRPi/shadow/update/accepted' WHERE state.reported.moisture = 'low'
```

This statement triggers the rule whenever the `moisture` value is reported as `low` for the specified MQTT topic.

#### Important

If you named your thing something other than `MyRPi`, be sure to substitute your thing's name in the preceding AWS IoT SQL statement. Otherwise, the rule might never be triggered.

6. For **Set one or more actions**, choose **Add action**.
7. On the **Select an action** page, choose **Send a message as an SNS push notification**.



8. Choose **Configure action**.
9. On the **Configure action** page, for **SNS target**, choose **Create**. Enter a name for the SNS topic, for example, **MyRPiLowMoistureTopic**, and then choose **Create**. If you choose to use a different name, be sure to substitute it throughout this sample.
10. For **Message format**, choose **RAW**.
11. For **IAM role name**, choose **Create a new role**, and then enter a name for the new role, for example, **MyRPiLowMoistureTopicRole**. If you choose to use a different name, be sure to substitute it throughout this sample.
12. Choose **Create a new role**.
13. For **IAM role name**, choose **MyRPiLowMoistureTopicRole**.
14. Choose **Add action**.

Configure action

SNS Send a message as an SNS push notification

\*SNS target MyRPiLowMoistureTopic 2 1 Create Clear Select

Message format RAW 3

Choose or create a role to grant AWS IoT access to perform this action.

\*IAM role name MyRPiLowMoistureTopicRole 5 4 Update role Create a new role

Cancel 6 Add action

15. Choose **Create rule**.
16. Set up Amazon SNS to send the messages through your Amazon SNS topic to your email inbox. On the AWS navigation bar, choose **Services**. In the **Find a service by name or feature** box, enter **SNS**, and then press **Enter**.
17. In the service navigation pane, choose **Topics**.



18. Select the check box next to **MyRPiLowMoistureTopic**.

Topics		
		Publish to topic
		Create new topic
Actions ▾		
Filter		
	Name	ARN
<input type="checkbox"/>	[REDACTED]	[REDACTED]
<input checked="" type="checkbox"/>	MyRPiLowMoistureTopic	arn:aws:sns:us-east-[REDACTED]:MyRPiLowMoistureTopic

19. For **Actions**, choose **Subscribe to topic**.

Topics		
		Publish to topic
		Create new topic
Actions ▾		
Filter		
	Name	ARN
<input type="checkbox"/>	[REDACTED]	[REDACTED]
<input type="checkbox"/>	[REDACTED]	arn:aws:sns:[REDACTED]
<input checked="" type="checkbox"/>	MyRPiLowMoistureTopic	arn:aws:sns:[REDACTED]

1
2

- Edit topic display name
- Subscribe to topic**
- Confirm a subscription
- Edit topic policy
- Edit topic delivery policy
- Delivery status

20. In the **Create subscription** dialog box, for **Protocol**, choose **Email**.  
 21. For **Endpoint**, enter your email address.  
 22. Choose **Create subscription**.  
 23. Watch your inbox for a subscription confirmation email titled **AWS Notification – Subscription Confirmation from no-reply@sns.amazonaws.com**. When it arrives, open it, and then click the **Confirm subscription** link. After you click the link, a confirmation page is displayed in your web browser. You can close this confirmation page.

**Important**

Until you confirm this subscription, you won't receive any email alerts from this Amazon SNS topic, even though AWS IoT might be sending email alerts to it.

## Step 5: Simulate Random Moisture Levels

In this step, you use your development computer to simulate soil moisture readings by generating random data. You then push those readings into the related shadow in AWS IoT. When the readings get too low, Amazon SNS automatically sends an email alert to the houseplant's owner.

1. Make sure that your development computer has [Python](#) and [pip](#) installed.

[pip](#) is included with Python versions 3.4 and later. To check your installed version of Python, run the command **python --version** from the command prompt running in Admin mode for Windows, or

from a terminal session running in macOS, Linux, or Unix. To check whether **pip** is also installed, run the command **pip --version**.

2. Use **pip** to install the AWS IoT Device SDK for Python on your development computer. To do this, run the command **pip install AWSIoTPythonSDK**.
3. Use a text editor to create a new file on your development computer with the following code:

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient
import random, time

# A random programmatic shadow client ID.
SHADOW_CLIENT = "myShadowClient"

# The unique hostname that &IoT; generated for
# this device.
HOST_NAME = "yourhostname-ats.iot.us-east-1.amazonaws.com"

# The relative path to the correct root CA file for &IoT;;
# which you have already saved onto this device.
ROOT_CA = "AmazonRootCA1.pem"

# The relative path to your private key file that
# &IoT; generated for this device, which you
# have already saved onto this device.
PRIVATE_KEY = "yourkeyid-private.pem.key"

# The relative path to your certificate file that
# &IoT; generated for this device, which you
# have already saved onto this device.
CERT_FILE = "yourkeyid-certificate.pem.crt"

# A programmatic shadow handler name prefix.
SHADOW_HANDLER = "MyRPi"

# Automatically called whenever the shadow is updated.
def myShadowUpdateCallback(payload, responseStatus, token):
    print()
    print('UPDATE: $aws/things/' + SHADOW_HANDLER +
          '/shadow/update/#')
    print("payload = " + payload)
    print("responseStatus = " + responseStatus)
    print("token = " + token)

# Create, configure, and connect a shadow client.
myShadowClient = AWSIoTMQTTShadowClient(SHADOW_CLIENT)
myShadowClient.configureEndpoint(HOST_NAME, 8883)
myShadowClient.configureCredentials(ROOT_CA, PRIVATE_KEY,
                                   CERT_FILE)
myShadowClient.configureConnectDisconnectTimeout(10)
myShadowClient.configureMQTTOperationTimeout(5)
myShadowClient.connect()

# Create a programmatic representation of the shadow.
myDeviceShadow = myShadowClient.createShadowHandlerWithName(
    SHADOW_HANDLER, True)

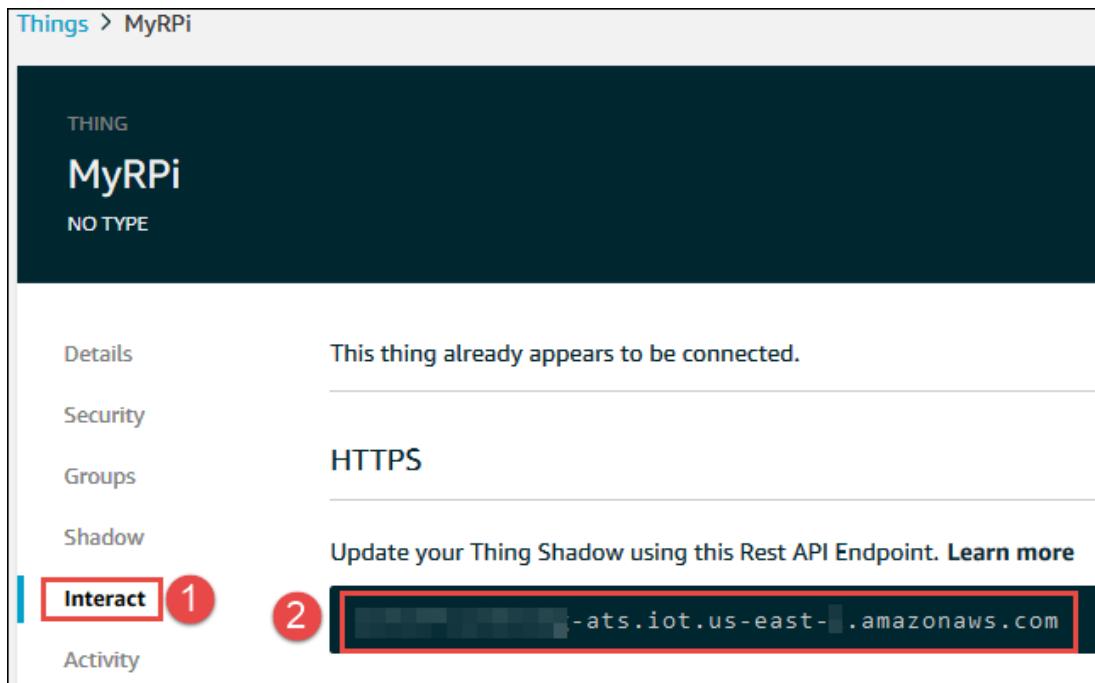
# Keep generating random test data until this script
# stops running.
# To stop running this script, press Ctrl+C.
while True:
    # Generate random True or False test data to represent
    # okay or low moisture levels, respectively.
    moisture = random.choice([True, False])
```

```
if moisture:
    myDeviceShadow.shadowUpdate(
        '{"state":{"reported":{"moisture":"okay"}}}',
        myShadowUpdateCallback, 5)
else:
    myDeviceShadow.shadowUpdate(
        '{"state":{"reported":{"moisture":"low"}}}',
        myShadowUpdateCallback, 5)

# Wait for this test value to be added.
time.sleep(60)
```

In the preceding code, replace the following values:

- *yourhostname-ats.iot.us-east-1.amazonaws.com* with the REST API endpoint that AWS IoT generated for you. To get this endpoint, in the [AWS IoT console](#) navigation pane, expand **Manage**, choose **Things**, and then choose your thing's name (for example, **MyRPi**). Choose **Interact**, and then look in the **HTTPS** area.



- *AmazonRootCA1.pem* with the name of the root CA for AWS IoT, which you saved earlier to your development computer.
  - *yourkeyid-private.pem.key* with the name of the private key for your device in AWS IoT, which you saved earlier to your development computer.
  - *yourkeyid-certificate.pem.crt.txt* with the name of the root certificate file for your device in AWS IoT, which you saved earlier to your development computer.
  - The *60* in `time.sleep(60)` with the number of seconds to wait for each new random simulated reading to be generated. The lower the value that you use for this number, the more frequently you might get email alerts.
4. Save the file with the extension `.py`, for example, `moisture.py`, in the same directory where you saved your root CA for AWS IoT, the private key file for your device in AWS IoT, and the root certificate file for your device in AWS IoT. If you choose to use a different name for the `.py` file, be sure to substitute it throughout this sample.

5. From the command prompt running in Admin mode for Windows, or from a terminal session in macOS, Linux, or Unix, switch to the directory that contains the `moisture.py` file. Then run the command `python moisture.py` for Python to start running the `moisture.py` script.

Every **60** seconds, the script generates a random `True` or `False` value. If the value is `True`, Python sends a "moisture okay" reading to AWS IoT. If the value is `False`, Python sends a low moisture reading to AWS IoT. Whenever AWS IoT receives a low moisture reading, it triggers a rule that sends an alert to your email address through Amazon SNS.

6. When you are done, press **Ctrl+C** to stop running the script.

If you don't have a Raspberry Pi, you have reached the end of this sample walkthrough, and you can skip ahead to [Cleaning Up \(p. 148\)](#). Otherwise, continue to [Module 2: Sending Data with the Raspberry Pi \(p. 129\)](#) to begin preparing your Raspberry Pi.

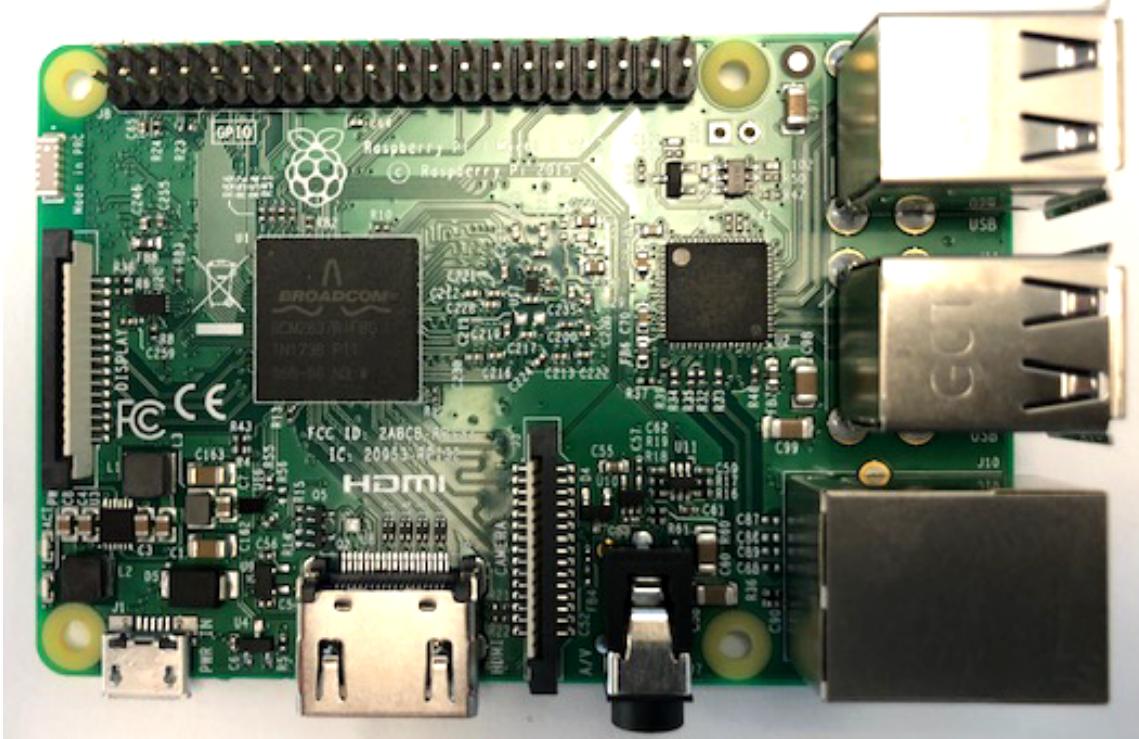
## Module 2: Sending Data with the Raspberry Pi

### Prerequisites for Steps 6–12

In [Module 1: Setting Up AWS IoT and Sending Data with Your Development Computer \(p. 109\)](#), you used your development computer to simulate soil moisture readings by generating random data, and then pushed those simulated readings into AWS IoT. In Part 2 (Steps 6–12), you generate real soil moisture readings with a Raspberry Pi and then push those real readings into AWS IoT instead.

#### Prerequisites for Steps 6–12

- Complete all of the steps in [Module 1: Setting Up AWS IoT and Sending Data with Your Development Computer \(p. 109\)](#).
- A Raspberry Pi 3. This sample was tested with a [Raspberry Pi 3 Model B](#).



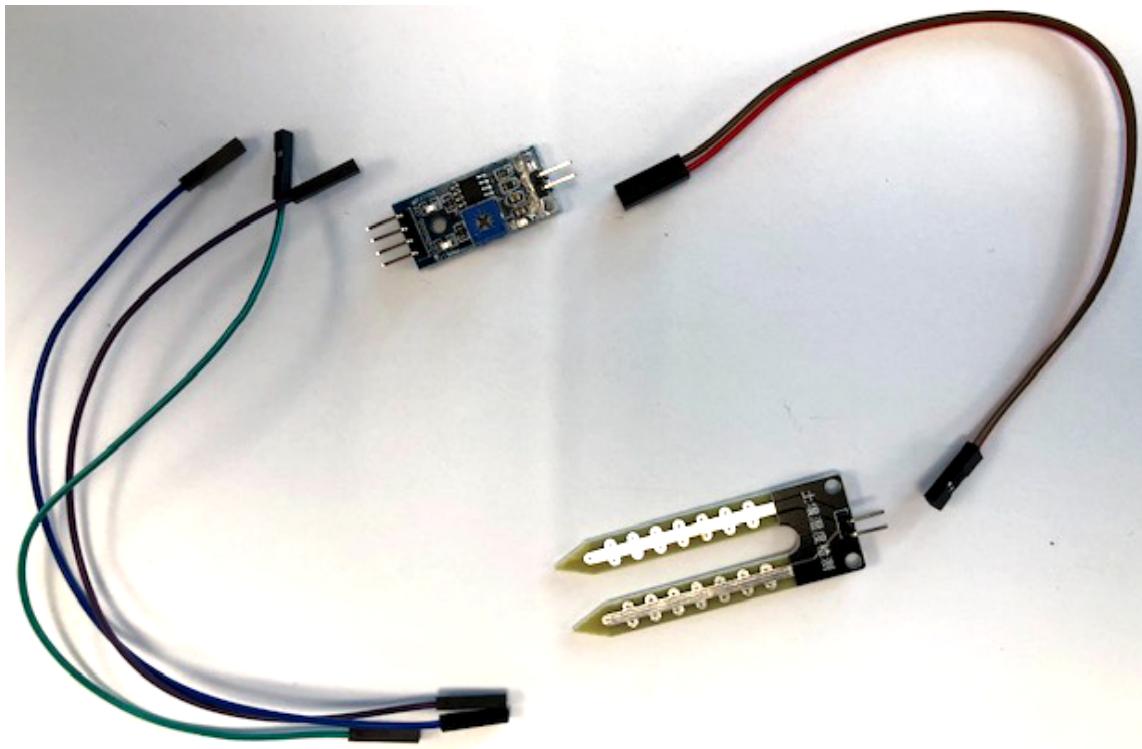
- A Raspberry Pi 3 [micro USB adapter power supply](#) with at least 5V 2.5A. This sample was tested with a 5V 2.5A power supply.



- A [micro SD card](#) with at least 16 GB. This sample was tested with a microSDHC 16 GB card.



- A desktop or laptop development computer with either a slot for the micro SD card or a micro SD card reader that can connect to the computer. This sample was tested with a laptop running Windows 10 Enterprise that has a built-in SD card reader.
- A network to connect the Raspberry Pi to AWS IoT and, optionally, to connect your development computer to the Raspberry Pi. This setup can be either a wireless network, or a physical network router that you can connect to with Ethernet cables. The Raspberry Pi 3 Model B provides both built-in Wi-Fi and an Ethernet port. This sample was tested with a wireless network.
- If you don't want to access the Raspberry Pi from your development computer, you need to connect the Raspberry Pi to a separate keyboard, mouse, and monitor. The Raspberry Pi 3 Model B provides four USB ports and a full-size HDMI port. This sample was tested with a USB keyboard, a USB mouse, and a monitor with HDMI input.
- A [soil moisture sensor kit](#) compatible with Raspberry Pi. This includes the sensor module (the "probe") and a microcontroller. You also need two female-to-female connection wires from the sensor module to the microcontroller, and three female-to-female connection wires from the microcontroller to the Raspberry Pi's onboard GPIO pins. This sample was tested with a Gikfun soil moisture sensor.



- A glass of water.
- A common houseplant.

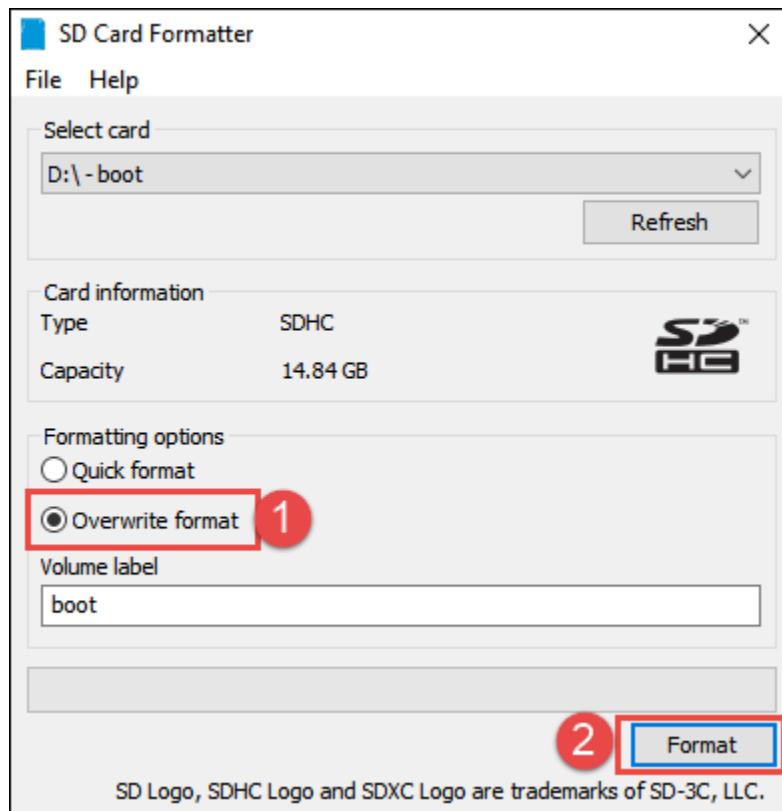
## Step 6: Begin Preparing the microSDHC Card

In this step, you prepare a microSDHC card to store the Raspberry Pi's operating system.

If you already have a Raspberry Pi with an operating system such as Raspbian installed, then connect to your Raspberry Pi and skip ahead to [Step 10: Set Up the Soil Moisture Sensor Kit \(p. 139\)](#).

Insert a blank microSDHC card into your desktop or laptop computer. Use an SD card formatting app or utility to format the SD card. For example, if you are using Windows or macOS, complete the following steps:

1. a. Download and install [SD Card Formatter](#).
2. Run SD Card Formatter.
3. **Select card** should already be set to the drive letter for the microSDHC card. If not, choose it.
4. For **Formatting options**, choose **Overwrite format**.
5. Choose **Format**.



When prompted to continue formatting the microSDHC card, choose **Yes**. The format operation can take a while.

## Step 7: Download Raspbian to the microSDHC Card

In this step you download Raspbian onto the formatted microSDHC card. Raspbian is an operating system based on Debian and optimized for the Raspberry Pi hardware.

1. Go to the [Raspberry Pi Downloads](#) page.
2. Choose **Raspbian**.
3. On the [Download Raspbian](#) page, for **Raspbian Stretch with Desktop**, choose **Download ZIP**.

The screenshot shows the official Raspbian website. At the top, there's a navigation bar with links for Blog, Downloads, Community, Help, Forums, and Education. A small Raspberry Pi logo is on the left, and a green shopping bag icon with the letter 'B' is on the right. Below the navigation, a red banner with the word "RASPBIAN" in white is visible. The main content area has a light gray background. It starts with a paragraph about Raspbian being the Foundation's official supported operating system, mentioning installation options like NOOBS or the installation guide. It then describes Raspbian's pre-installed software, including Python, Scratch, Sonic Pi, Java, and more. Following this, it discusses the size of the Desktop image (over 4GB) and provides links for unzipping tools like 7Zip and The Unarchiver. Two download options are shown: "RASPBIAN STRETCH WITH DESKTOP" and "RASPBIAN STRETCH LITE". Each option includes a small image of a Raspberry Pi board, version information (e.g., Version: October 2018, Release date: 2018-10-09), kernel version (4.14), release notes (link), and download links for Torrent and ZIP files. SHA-256 checksums are also provided.

**RASPBIAN**

**RASPBIAN STRETCH WITH DESKTOP**  
Image with desktop based on Debian Stretch  
Version: October 2018  
Release date: 2018-10-09  
Kernel version: 4.14  
Release notes: [Link](#)  
[Download Torrent](#) [Download ZIP](#)

**RASPBIAN STRETCH LITE**  
Minimal image based on Debian Stretch  
Version: October 2018  
Release date: 2018-10-09  
Kernel version: 4.14  
Release notes: [Link](#)  
[Download Torrent](#) [Download ZIP](#)

4. Use an image flashing app or utility to flash the .zip file that you just downloaded onto the microSDHC card. For example, for Windows, macOS, or Linux:
  1. Download and install [Etcher](#).
  2. Run Etcher.
  3. Choose **Select image**.
  4. Choose the .zip file that you just downloaded.
  5. Next to the picture of the drive, if the microSDHC card is not selected, choose it.
  6. Choose **Flash!**



The flashing operation can take a while.

## Step 8: Finish Preparing the microSDHC card

In this step, you add several files to the microSDHC card. These files enable you to connect to the Raspberry Pi from your desktop or laptop computer and enable the Raspberry Pi to communicate with AWS IoT.

1. If you plan to connect to the Raspberry Pi from your desktop or laptop computer, create a blank file named `ssh` in the root of the microSDHC card. This file allows you to connect to the Raspberry Pi from an SSH connection tool (such as PuTTY for Windows, the SSH utility in GitBash for Windows, or the SSH utility for macOS, Linux, or Unix) after the Raspberry Pi boots.

For example, for Windows, in the command prompt running in Admin mode, run the following command, which creates a blank file named `ssh` in the root of the microSDHC card. This command assumes the microSDHC card is connected as drive letter D.

```
fsutil file createnew D:\ssh 0
```

2. If you plan to connect to the Raspberry Pi from your desktop or laptop computer, create a blank file named `wpa_supplicant.conf` in the root of the microSDHC card. This file enables the Raspberry Pi to connect to a wireless network.

For example, for Windows, in the same command prompt, run the following command, which creates a blank file named `wpa_supplicant.conf` in the root of the microSDHC card. This command assumes the microSDHC card is connected as drive letter D.

```
fsutil file createnew D:\wpa_supplicant.conf 0
```

3. If you created the blank `wpa_supplicant.conf` file, open a text editor and add the following content to the `wpa_supplicant.conf` file. Then save the file.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
network={
    ssid="MyWirelessNetworkName"
    psk="MyWirelessNetworkPassword"
    key_mgmt=WPA-PSK
}
```

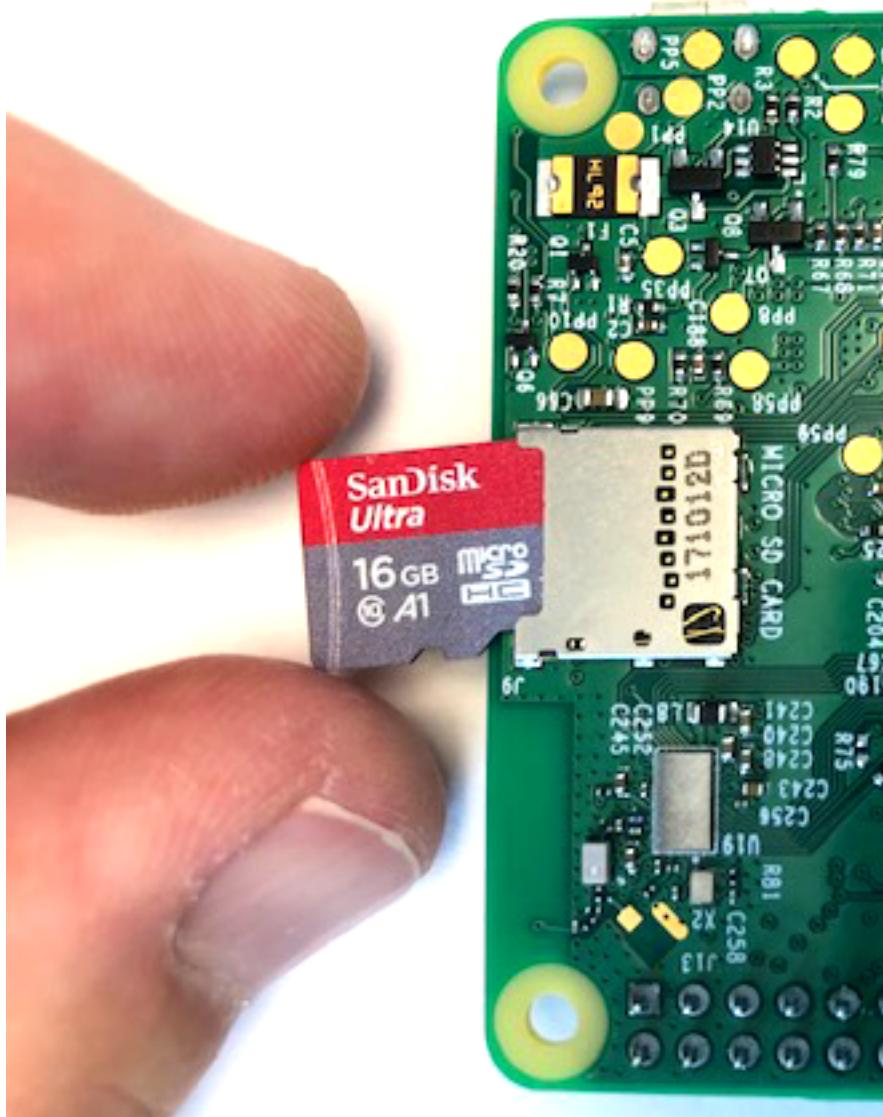
In the preceding content, replace *MyWirelessNetworkName* with the name of the wireless network. Replace *MyWirelessNetworkPassword* with the password for the wireless network. For more information, see [Wireless Connectivity](#) on the Raspberry Pi website.

4. Create a folder in the root of the microSDHC card named deviceSDK.
5. Copy the files that AWS IoT generated for you earlier ending in .certificate.pem.ct.txt (the root certificate file for your device in AWS IoT), private.pem.key (the private key for your device in AWS IoT), and .pem (the root CA for AWS IoT) into this new deviceSDK folder.
6. Copy the file named moisture.py from [Step 5: Simulate Random Moisture Levels \(p. 126\)](#) into this new deviceSDK folder.
7. Eject the microSDHC card from your desktop or laptop computer.

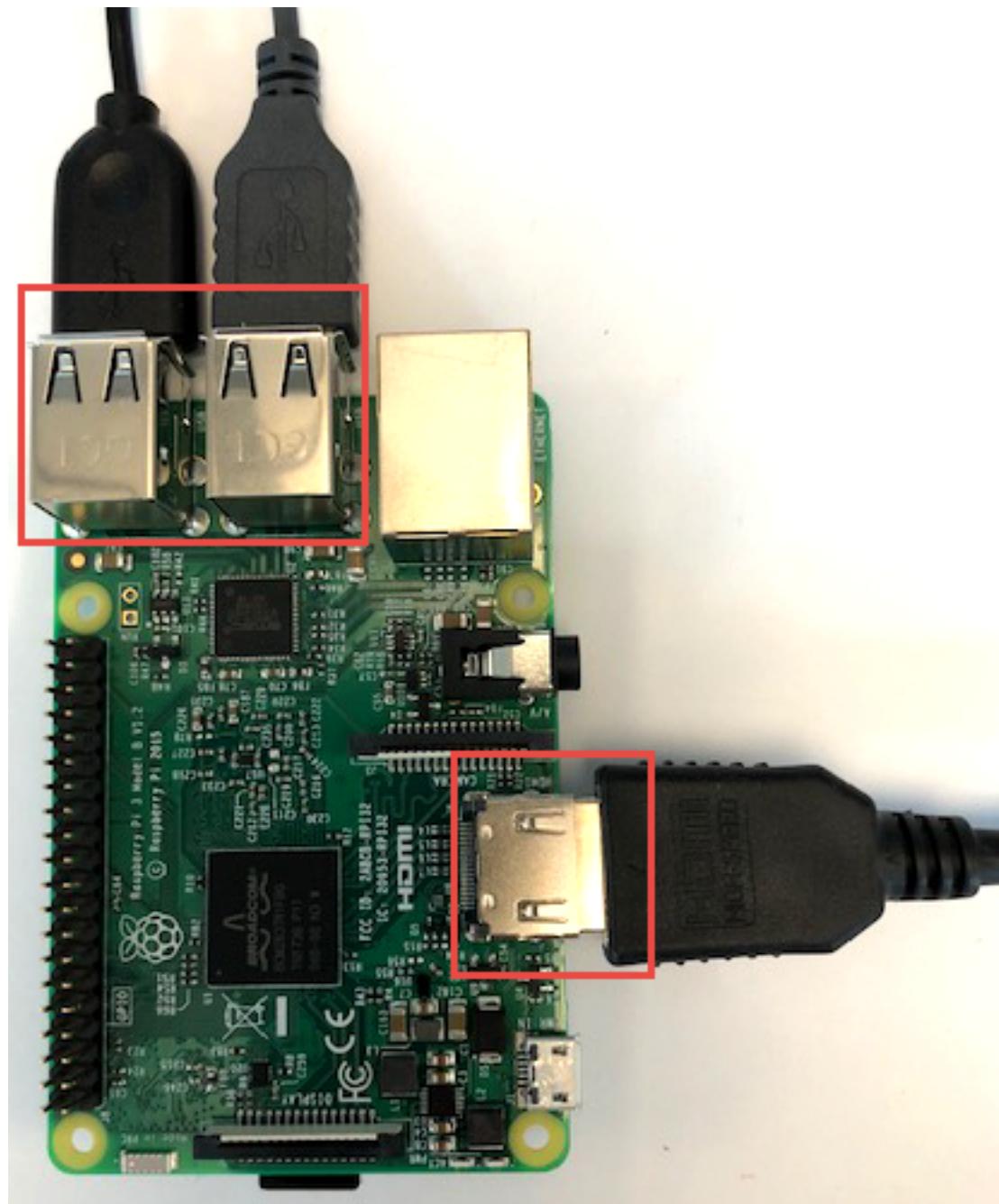
## Step 9: Connect to the Raspberry Pi and Set Up Raspbian

In this step, you start the Raspberry Pi. You connect to it directly or from your desktop or laptop computer. If you connect directly to the Raspberry Pi, you then set up Raspbian on it.

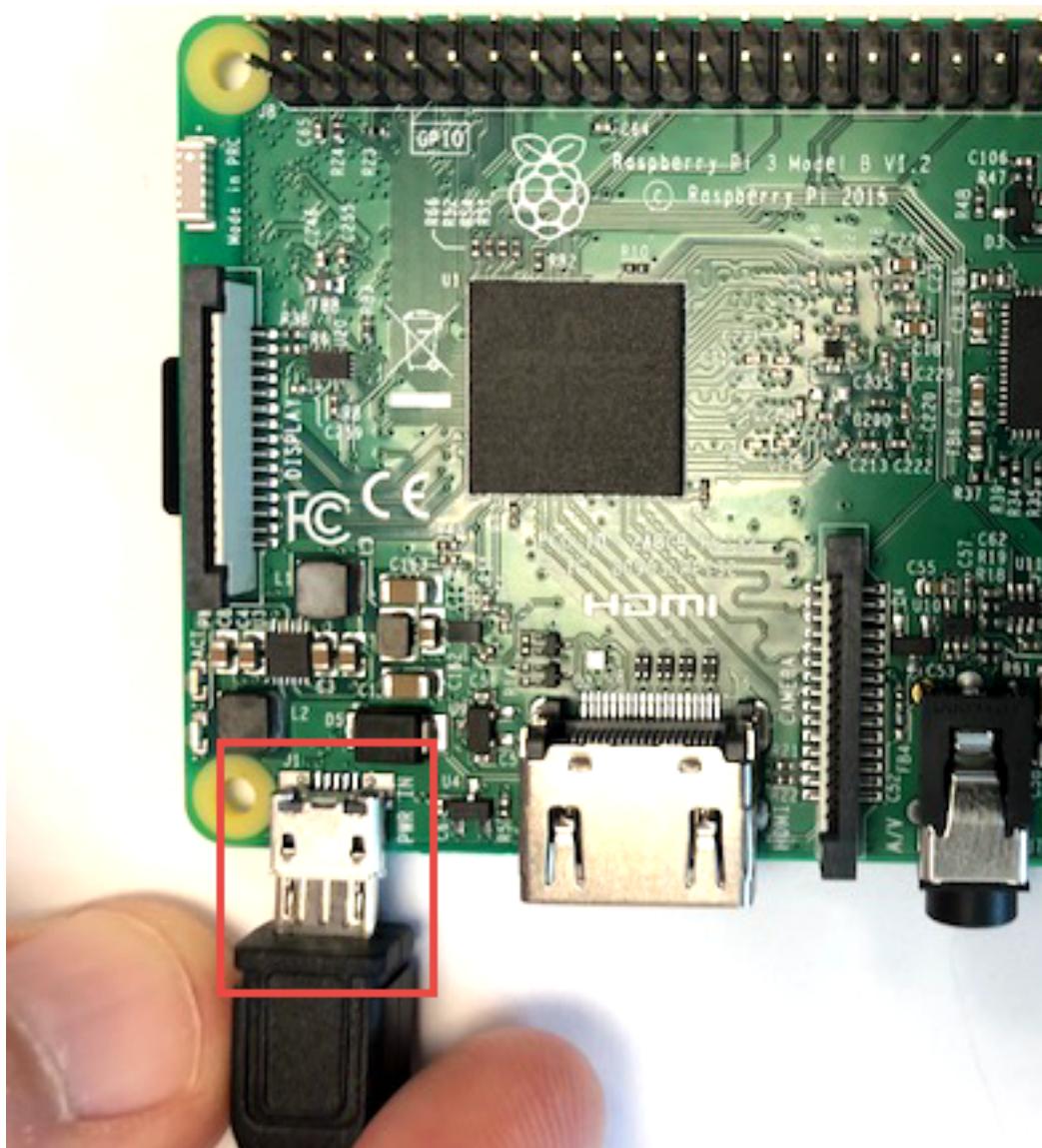
1. Insert the microSDHC card into the Raspberry Pi. The card slot is on the underside of the motherboard. The card goes in the slot only one way, typically with the writing on the card facing down.



2. If you want to access the Raspberry Pi directly, instead of from your development computer, connect a separate keyboard, mouse, and monitor directly to the Raspberry Pi, for example, by using the USB and HDMI ports. Although the Raspberry Pi 3 provides Bluetooth connectivity, you won't be able to connect via Bluetooth until you boot the Raspberry Pi for the first time.



3. Insert the prongs of the micro USB adapter power supply into your power source, and then plug the micro USB end into its slot in the Raspberry Pi.



The Raspberry Pi red power LED light turns on, the green activity LED light begins flickering, and the Raspbian operating system automatically boots. If you plan to access the Raspberry Pi from your development computer, the Raspberry Pi attempts to connect to the wireless network using the password that you specified earlier in the `wpa_supplicant.conf` file.

4. If you don't want to access the Raspberry Pi from your development computer, skip ahead to step 5 in this procedure.

To access the Raspberry Pi from your development computer, get the IP address of the Raspberry Pi that is now connected to the wireless network. For example, if you can log in to your wireless network router as an administrator, you should be able to look up the IP address from there. Otherwise, you could use a utility such as **ping** or an application such as [Nmap for Windows](#).

After you get the Raspberry Pi's IP address, connect from your Windows desktop or laptop computer to the Raspberry Pi using an SSH connection tool such as [PuTTY](#), or the SSH utility in [Git Bash for Windows](#).

If you use PuTTY, for Host Name (or IP address), use the format `pi@X.X.X.X`. For SSH, use `ssh pi@X.X.X.X`. In either case, `pi` is the default user name, and `X.X.X.X` is the Raspberry Pi's IP address. When prompted for a password, use the default password, `raspberry`.

Skip ahead to [Step 10: Set Up the Soil Moisture Sensor Kit \(p. 139\)](#).

5. To access the Raspberry Pi directly instead of from your development computer, turn on your monitor to the correct input source (for example, HDMI input).

A dialog box might appear, notifying you that SSH is enabled on the Raspberry Pi and the default password for the user `pi` has not been changed. You can close this dialog box now, because you get an opportunity to change this password later in this step.

6. The first time you start the Raspberry Pi, a **Welcome to Raspberry Pi** dialog box is displayed. Choose **Next**.
7. On the **Set Country** page, choose the **Country**, **Language**, and **Timezone** you want. If you have a US keyboard, select the **US keyboard** box.
8. Choose **Next**.
9. On the **Change Password** page, enter a new password for the default user `pi`.

**Note**

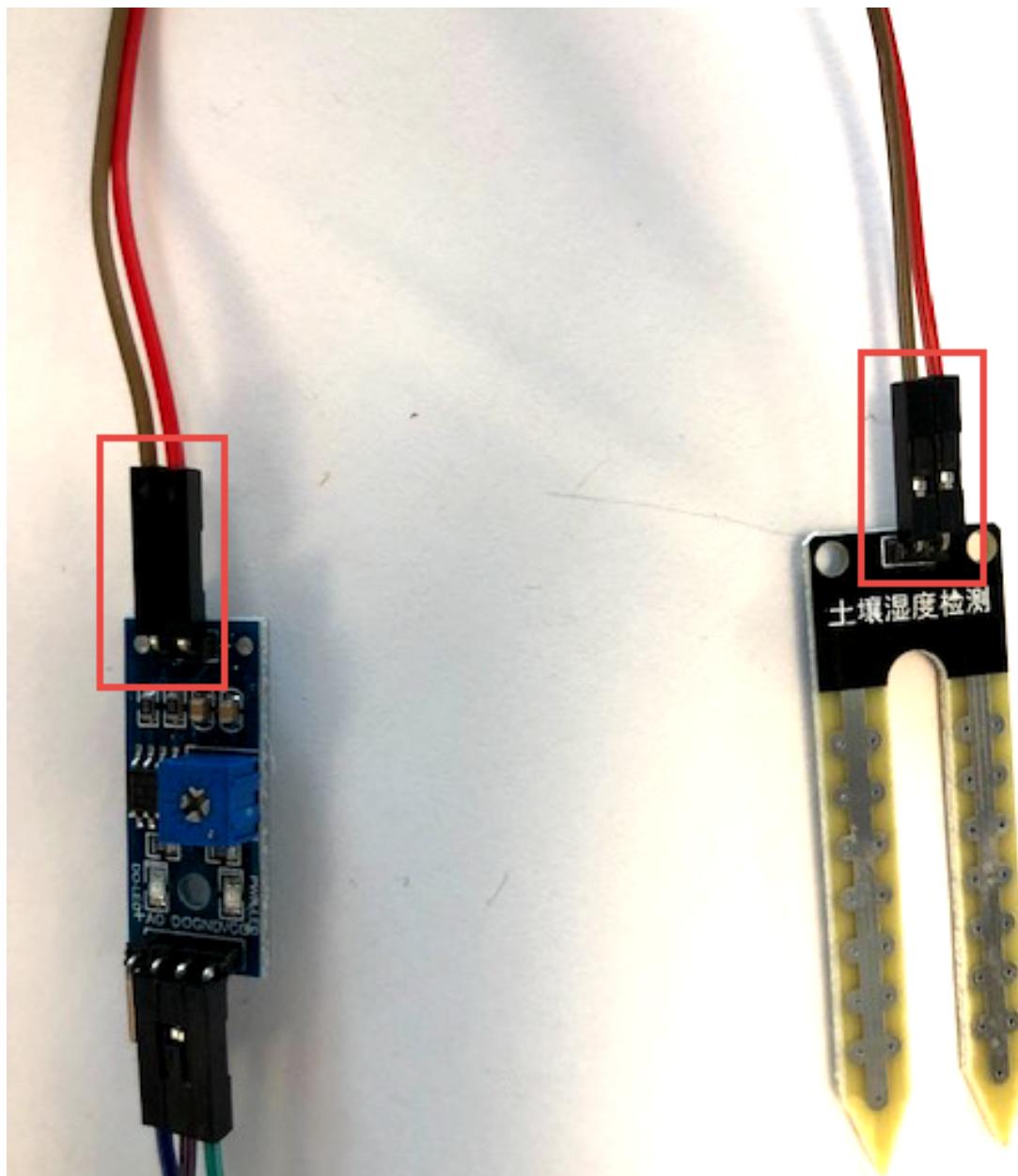
You can complete this procedure whether or not you set a new password, but setting a new password helps keep your device secure.

10. Choose **Next**.
11. On the **Select WiFi Network** page, choose the wireless network to connect to, and then choose **Next**.
12. On the **Update Software** page, choose **Next**. When the update is complete, choose **OK**.
13. On the **Setup Complete** page, choose **Reboot**, and wait while the Raspberry Pi restarts.

## Step 10: Set Up the Soil Moisture Sensor Kit

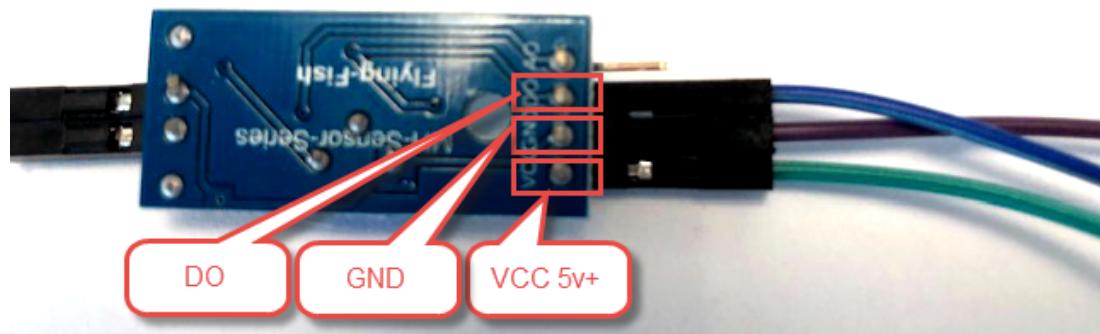
In this step, you connect the soil moisture sensor kit to the running Raspberry Pi and test it to make sure that the sensor kit works.

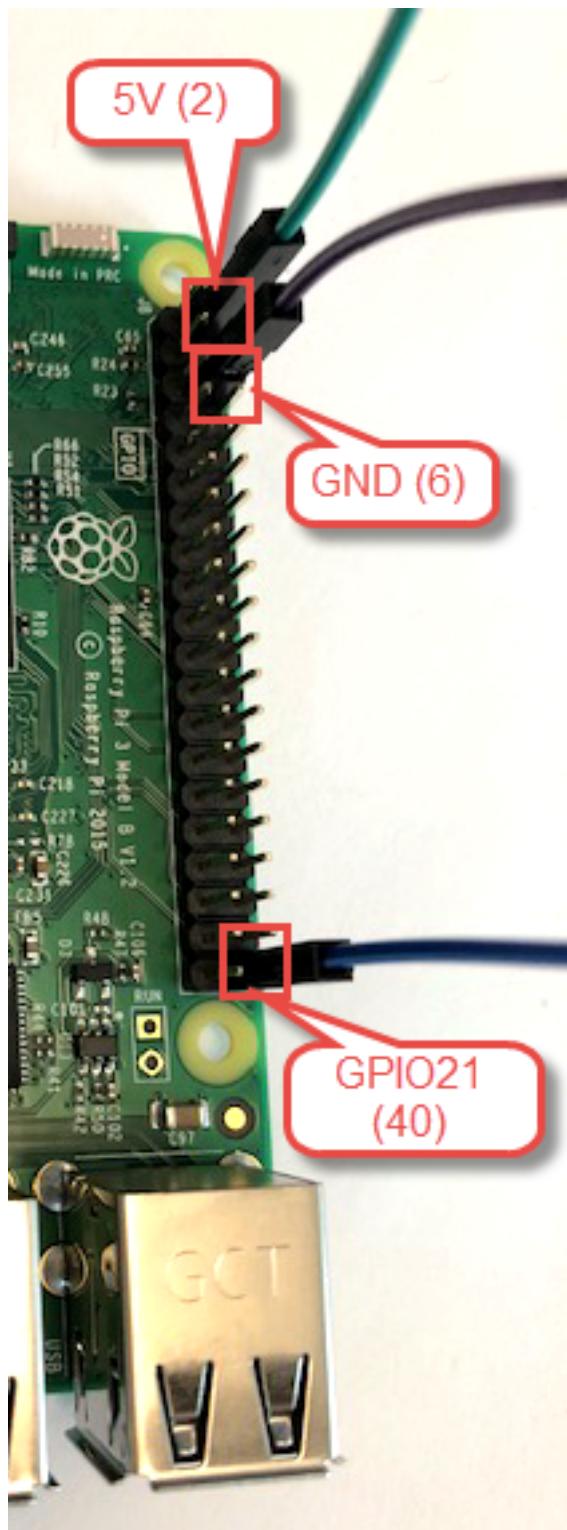
1. Connect two female-to-female connector wires from the two pins on the sensor module to the two pins on the microcontroller. The connector wires can be connected in either order, but be sure to connect to the two and only two pins on the one side of the microcontroller, and not to any of the four pins on the other side of the microcontroller.



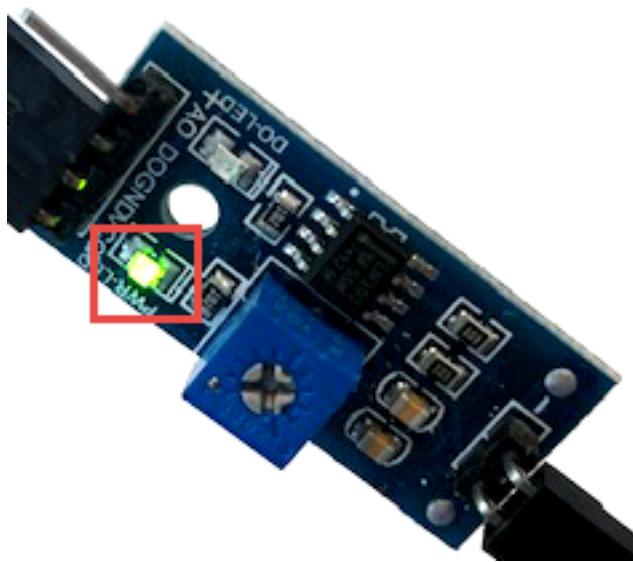
2. Connect three female-to-female connector wires from three specific pins on the microcontroller to three specific pins on the Raspberry Pi, as follows:
  1. Connect from the VCC 5v+ power pin on the microcontroller to one of the 5V power pins on the Raspberry Pi (for example, pin 2).
  2. Connect from the GND ground pin on the microcontroller to one of the GND ground pins on the Raspberry Pi (for example, pin 6).
  3. Connect from the DO digital data pin on the microcontroller to one of the GPIO pins on the Raspberry Pi (for example, GPIO21 on pin 40).
  4. Do not connect anything to the AO analog data pin on the microcontroller.

To see a graphical list of pins and their labels, you can run the `pinout` command on the Raspberry Pi, or go to <https://pinout.xyz>.



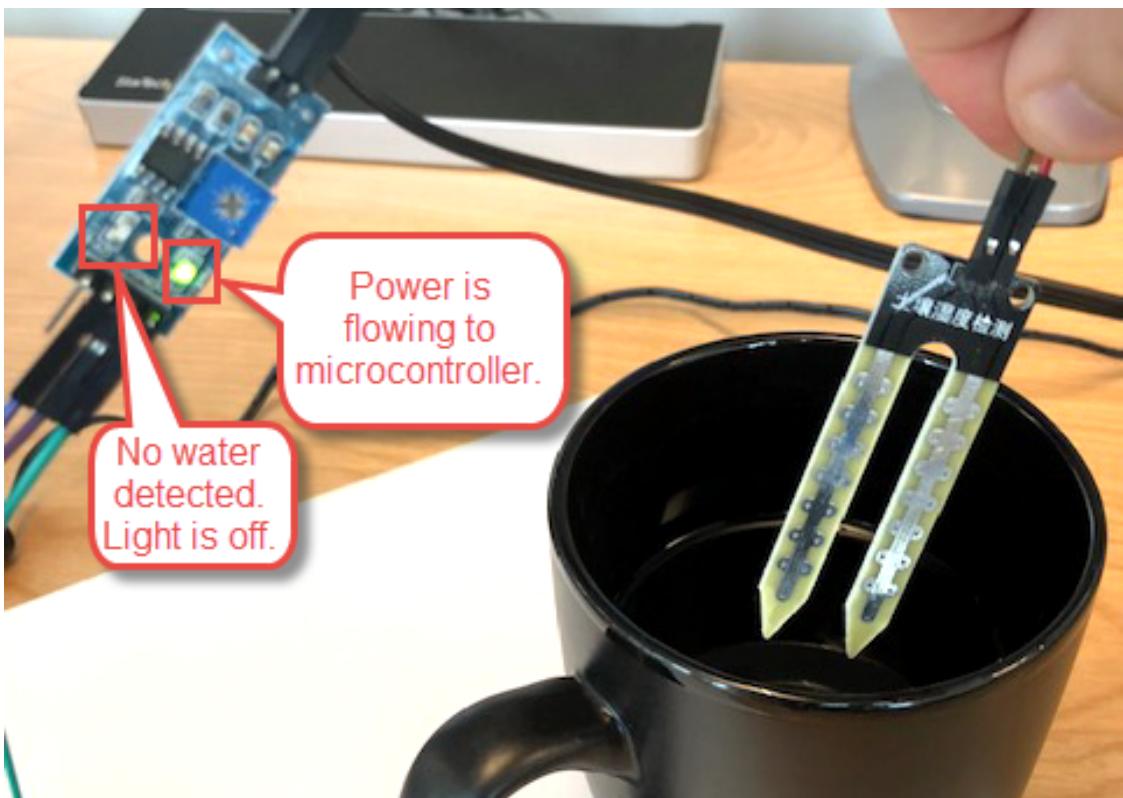


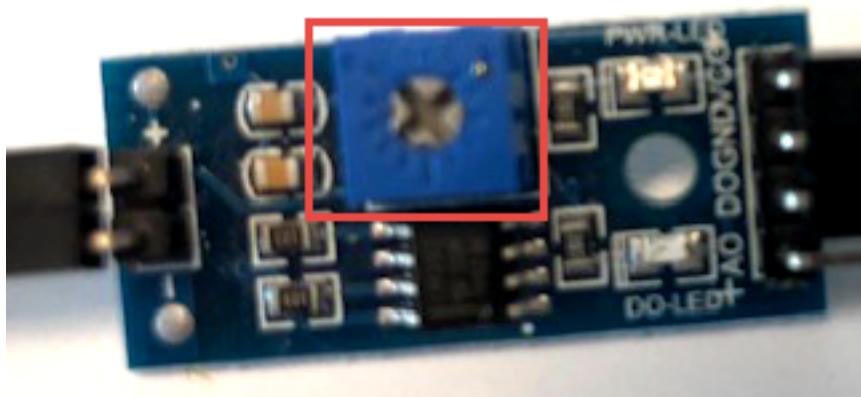
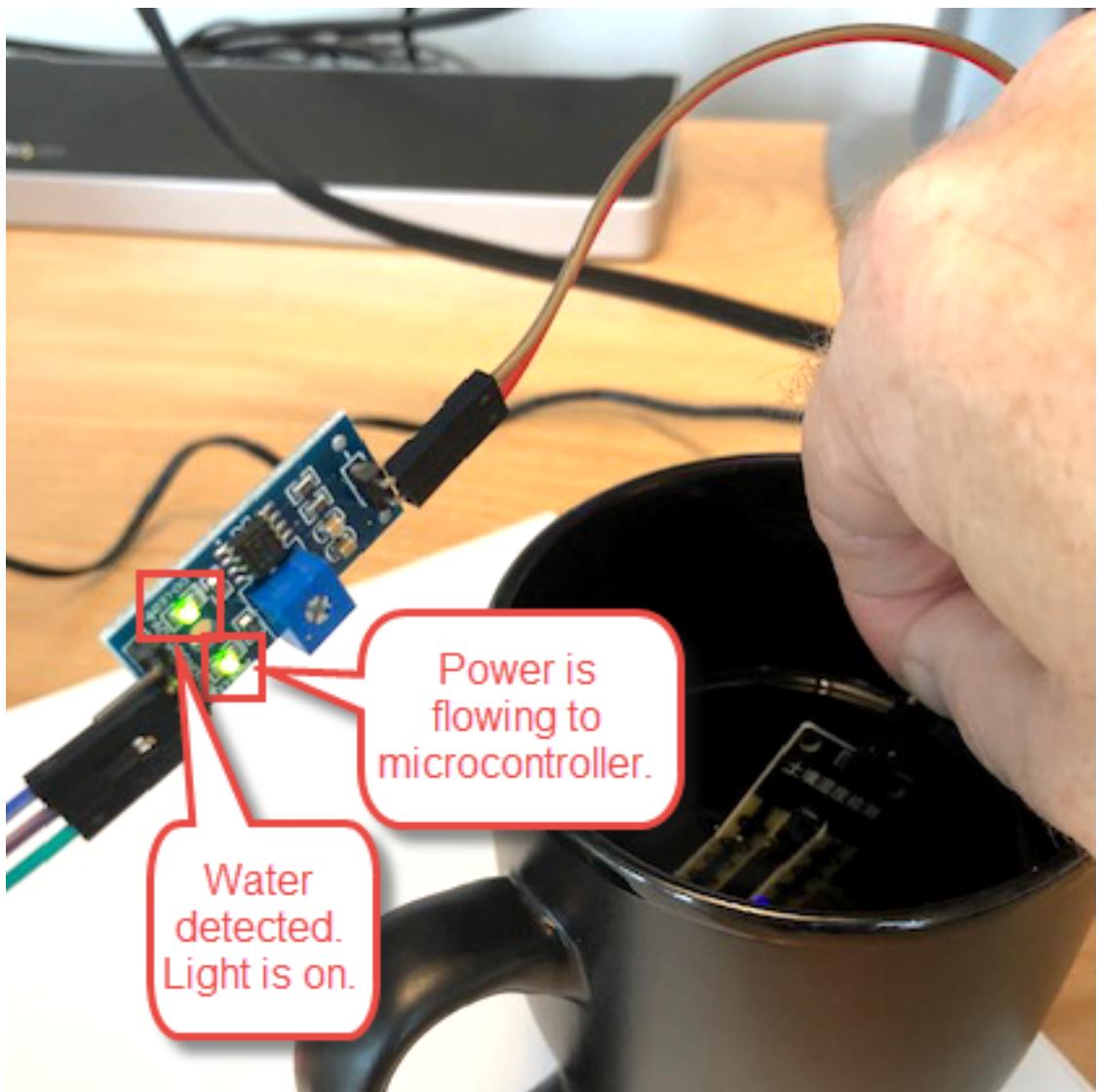
If the connections are correct, the PWR power LED light is displayed on the microcontroller.



The LED lights on some microcontrollers might display in green while others might display in red. Both colors mean the same thing.

3. Place the prongs on the sensor module into a mug of water. If the sensor detects water, the DO digital data LED light is displayed on the microcontroller. If the DO LED light isn't displayed, with the prongs still in the water, use a slotted screwdriver to change the sensitivity of the potentiometer on the microcontroller until the DO light is displayed. Move the prongs in and out of the water to check whether the DO light goes on and off, adjusting the potentiometer on the microcontroller as needed.





## Step 11: Capture Data from the Soil Moisture Sensor Kit

In this step, you use the Python programming language to run some code on the Raspberry Pi to capture data from the soil moisture sensor kit.

1. Use an available code editor on the Raspberry Pi (for example, nano, IDLE, or vi) to create a file with the following code.

```
import RPi.GPIO as GPIO
import time

# Represents the GPIO21 pin.
channel = 21

# Use the GPIO BCM pin numbering scheme.
GPIO.setmode(GPIO.BCM)

# Receive input signals through the pin.
GPIO.setup(channel, GPIO.IN)

# Infinite loop to keep this script running.
while True:
    # 'No water' = 1/True (sensor's microcontroller light is off).
    if GPIO.input(channel):
        print("No water detected")
    else:
        # 'Water' = 0/False (microcontroller light is on).
        print("Water detected!")

    # Wait 5 seconds before checking again.
    time.sleep(5)

# Clean things up if for any reason we get to this
# point before script stops.
GPIO.cleanup()
```

This code listens every five seconds for input from the soil moisture sensor kit on the GPIO21 BCM pin (the 40th pin) on the Raspberry Pi. If the sensor's microcontroller light is off, a value of 1 is reported. If the light is on, a value of 0 is reported.

2. Save the file with the extension .py, for example, `gpio.py`, in the `deviceSDK` folder. If you choose to use a different name for the .py file, be sure to substitute it throughout this sample.
3. From the command prompt in PuTTY or SSH, or from the terminal in Raspbian, run commands to switch to the `deviceSDK` folder, and then use Python to run the `gpio.py` file, for example, `cd /deviceSDK && python gpio.py`.
4. Every 5–10 seconds, place the prongs on the sensor module into a glass of water, or remove the prongs from the water. Every 5 seconds, Python prints `No water detected` or `Water detected!`, depending on whether the prongs are in the water.
5. When you're done, stop running the script by pressing **Ctrl+C**.

## Step 12: Send Soil Moisture Sensor Readings to AWS IoT

In this step, you use the Python programming language to run some code on the Raspberry Pi that captures data from the soil moisture sensor kit and then sends it to AWS IoT.

To do this, you integrate some of the code that you wrote in the previous step into the code that you wrote in `moisture.py` for [Step 5: Simulate Random Moisture Levels \(p. 126\)](#).

1. Use an available code editor on the Raspberry Pi to open the `moisture.py` file that you created in [Step 5: Simulate Random Moisture Levels \(p. 126\)](#).
2. Add some of the code from the `gpio.py` file that you wrote into the `moisture.py` file, and make a few changes to the existing code in the `moisture.py` file, as indicated between the `### BEGIN` and `### END` comments, as follows. The final code is listed immediately after the following code to be changed.

```
from AWSIoTPythonSDK.MQTTLib import AWSIoTMQTTShadowClient
import RPi.GPIO as GPIO
import time

# A random programmatic shadow client ID.
SHADOW_CLIENT = "myShadowClient"

# The unique hostname that AWS IoT generated for
# this device.
HOST_NAME = "yourhostname-ats.iot.us-east-1.amazonaws.com"

# The relative path to the correct root CA file for AWS IoT,
# that you have already saved onto this device.
ROOT_CA = "AmazonRootCA1.pem"

# The relative path to your private key file that
# AWS IoT generated for this device, that you
# have already saved onto this device.
PRIVATE_KEY = "yourkeyid-private.pem.key"

# The relative path to your certificate file that
# AWS IoT generated for this device, that you
# have already saved onto this device.
CERT_FILE = "yourkeyid-certificate.pem.crt.txt"

# A programmatic shadow handler name prefix.
SHADOW_HANDLER = "MyRPi"

# Automatically called whenever the shadow is updated.
def myShadowUpdateCallback(payload, responseStatus, token):
    print()
    print('UPDATE: $aws/things/' + SHADOW_HANDLER +
          '/shadow/update/#')
    print("payload = " + payload)
    print("responseStatus = " + responseStatus)
    print("token = " + token)

# Create, configure, and connect a shadow client.
myShadowClient = AWSIoTMQTTShadowClient(SHADOW_CLIENT)
myShadowClient.configureEndpoint(HOST_NAME, 8883)
myShadowClient.configureCredentials(ROOT_CA, PRIVATE_KEY,
                                   CERT_FILE)
myShadowClient.configureConnectDisconnectTimeout(10)
myShadowClient.configureMQTTOperationTimeout(5)
myShadowClient.connect()

# Create a programmatic representation of the shadow.
myDeviceShadow = myShadowClient.createShadowHandlerWithName(
    SHADOW_HANDLER, True)

# Represents the GPIO21 pin on the Raspberry Pi.
channel = 21

# Use the GPIO BCM pin numbering scheme.
GPIO.setmode(GPIO.BCM)
```

```
# Receive input signals through the pin.  
GPIO.setup(channel, GPIO.IN)  
  
while True:  
  
    if GPIO.input(channel):  
        myDeviceShadow.shadowUpdate(  
            '{"state":{"reported":{"moisture":"low"}}}',  
            myShadowUpdateCallback, 5)  
    else:  
        myDeviceShadow.shadowUpdate(  
            '{"state":{"reported":{"moisture":"okay"}}}',  
            myShadowUpdateCallback, 5)  
  
    # Wait for this test value to be added.  
    time.sleep(60)
```

### Note

In the preceding code, note that the following values will not match your code:

1. *yourhostname-ats.iot.us-east-1.amazonaws.com* will instead be the REST API endpoint that AWS IoT generated for you.
2. *AmazonRootCA1.pem* will instead be name of the root CA for AWS IoT.
3. *yourkeyid-private.pem.key* will instead be the name of the private key for your device in AWS IoT.
4. *yourkeyid-certificate.pem.crt.txt* will instead be the name of the root certificate file for your device in AWS IoT.
5. The *60* in `time.sleep(60)` will be the number of seconds you want to wait for each new reading to be generated. The lower this number, the more frequently you might get email alerts.
3. Save your changes to the `moisture.py` file.
4. From the command prompt in PuTTY or SSH, or from the terminal in Raspbian, run the following command to use the `pip` program to install the AWS IoT Device SDK for Python on the Raspberry Pi:

```
pip install AWSIoTPythonSDK
```

5. Run commands to switch to the `deviceSDK` folder if you're not already there. Then use Python to run the `moisture.py` file, for example, `cd /deviceSDK && python moisture.py`.
6. Every 45–60 seconds, place the prongs on the sensor module into a glass of water, or remove the prongs from the water. Every 60 seconds, Python reports "moisture": "low" or "moisture": "okay" to AWS IoT, depending on whether the prongs are in the water. Whenever AWS IoT receives a low moisture reading, it triggers a rule that sends an alert to your email address through Amazon SNS.
7. When you're done, press **Ctrl+C** to stop running the script.
8. You can now replace the glass of water with a common houseplant. Place the prongs on the sensor module into the houseplant's soil. Adjust the potentiometer on the sensor's microcontroller to get the right sensitivity of soil moisture that you want.
9. In the `moisture.py` file, change the *60* in `time.sleep(60)` to the number of seconds between times you want to check the soil. For example, to check once an hour, change *60* to *3600*. To check once a day, change *60* to *86400*.
10. Restart the `moisture.py` script by running the command `python moisture.py`.
11. After each interval of seconds in `time.sleep` passes, Python reports "moisture": "low" or "moisture": "okay" to AWS IoT. This depends on whether the DO light on the sensor's

microcontroller is off or on, the soil moisture level, and the potentiometer's sensitivity setting. Whenever AWS IoT receives a low moisture reading, it triggers a rule that sends an alert to your email address through Amazon SNS.

- When you're done, stop running the script by pressing **Ctrl+C**.

## Cleaning Up

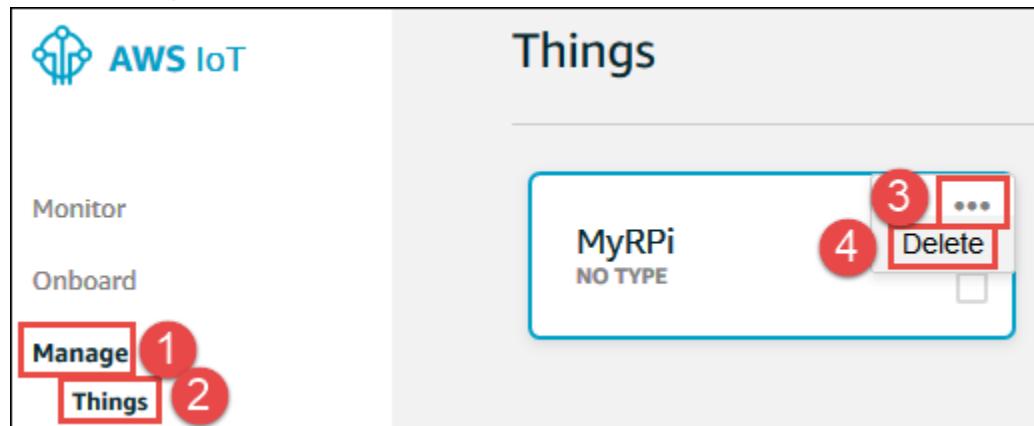
If you no longer want to use the AWS resources in AWS IoT, Amazon SNS, and IAM that you created for this sample, you can delete those resources by following the instructions in this section.

**Note**

If you don't delete these AWS resources, then any ongoing usage of those resources might start generating charges to your AWS account.

### To delete the thing in AWS IoT

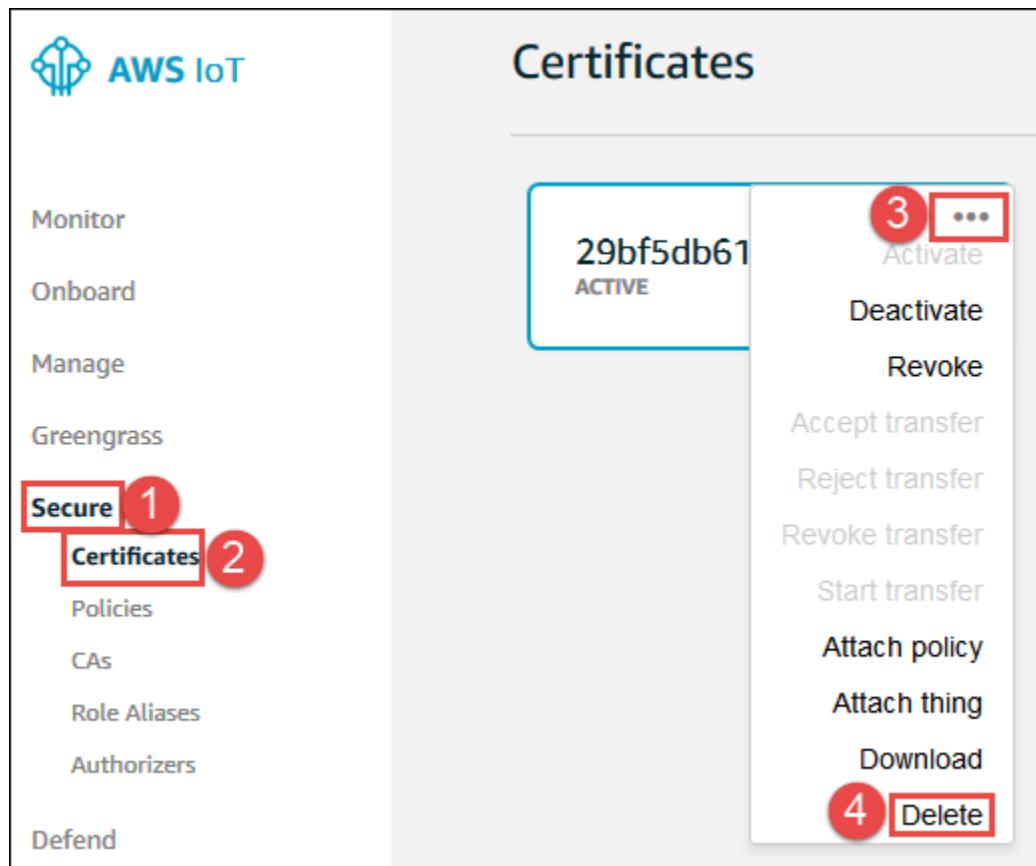
- In the AWS Management Console, open the [AWS IoT console](#), if it isn't already open. To do this, on the AWS navigation bar, choose **Services**. In the **Find a service by name or feature** box, enter **IoT Core**, and then press **Enter**.
- In the service navigation pane, expand **Manage**.
- If an **Introducing AWS IoT Device Management** dialog box is displayed, choose **Show me later**, or press **Esc**.
- Choose **Things**.
- In the list of things, in the card for the name of the thing you want to delete (for example, **MyRPi**), choose the ellipsis (...), and then choose **Delete**.



- When prompted, choose **Yes, continue with delete**.

### To delete the thing's certificate in AWS IoT

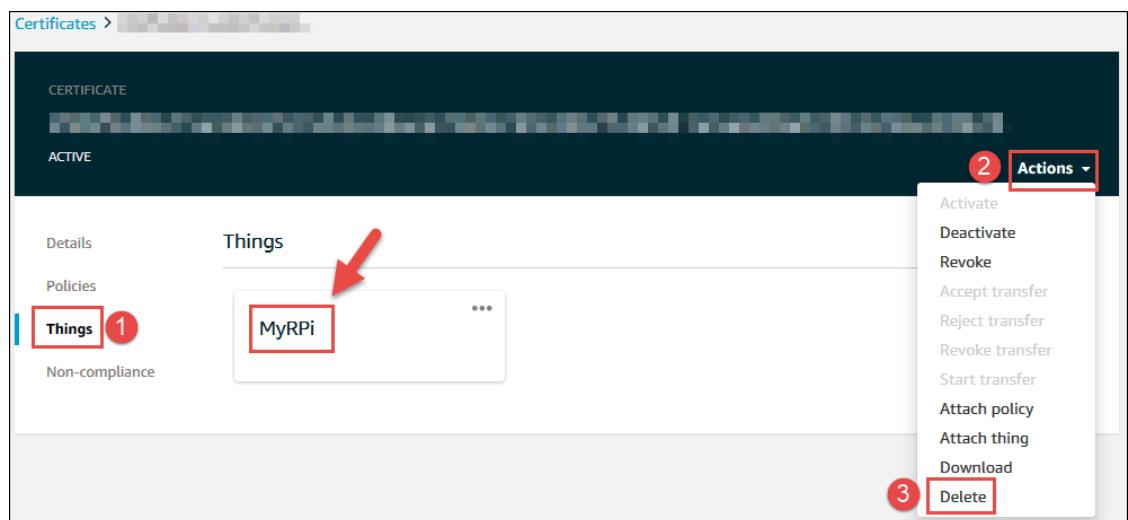
- In the [AWS IoT console](#), in the service navigation pane, expand **Secure**, and then choose **Certificates**.
- In the list of certificates, in the card for the thing's certificate, choose the ellipsis (...), and then choose **Delete**.



If you're not sure which certificate to delete, then in the list of certificates, choose the ID for the certificate that you think is the correct one. Then choose **Things**.

If your thing's name is displayed, this is the correct certificate to delete. Choose **Actions**, and then choose **Delete**.

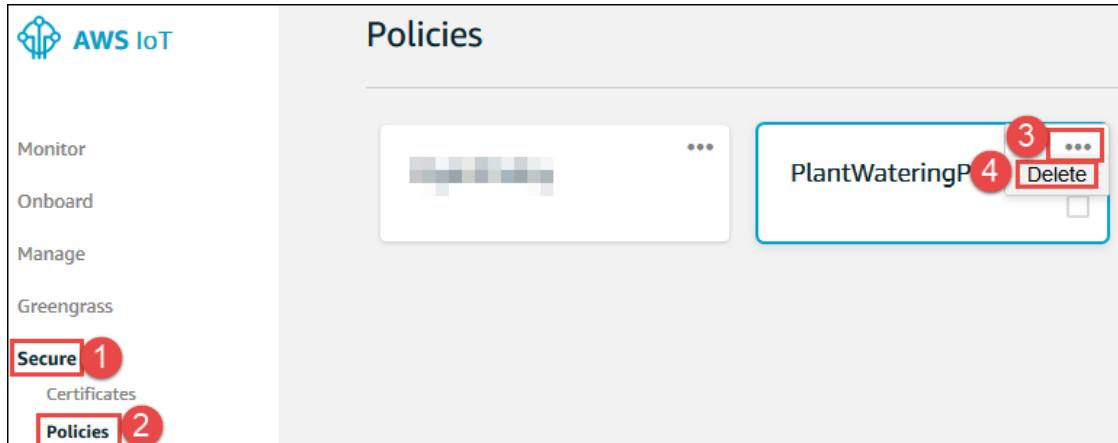
If, however, your thing isn't displayed, this isn't the correct certificate. Choose the back button to return to the list of certificates.



- When prompted, choose **Yes, continue with delete**.

### To delete the thing's policy in AWS IoT

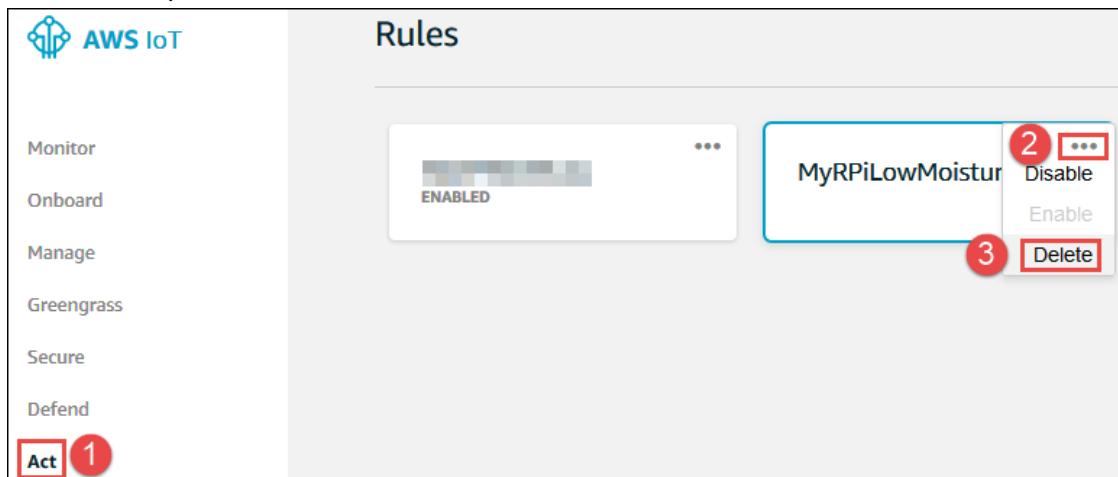
1. In the [AWS IoT console](#), in the service navigation pane, expand **Secure**, and then choose **Policies**.
2. In the list of policies, in the card for the thing's policy (for example, **PlantWateringPolicy**), choose the ellipsis (...), and then choose **Delete**.



3. When prompted, choose **Yes, continue with delete**.

### To delete the thing's rule in AWS IoT

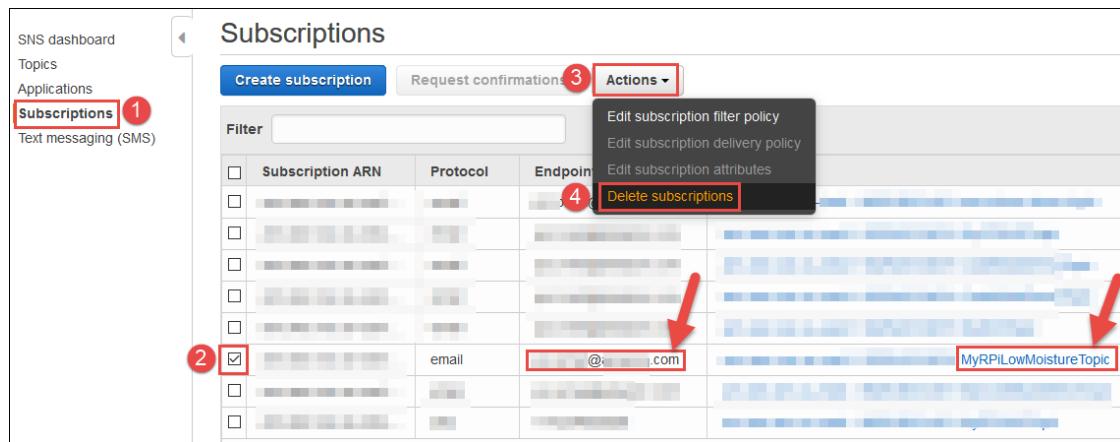
1. In the [AWS IoT console](#), in the service navigation pane, choose **Act**.
2. In the list of rules, in the card for the thing's rule (for example, **MyRPiLowMoistureAlertRule**), choose the ellipsis (...), and then choose **Delete**.



3. When prompted, choose **Yes, continue with delete**.

### To delete the related subscription in Amazon SNS

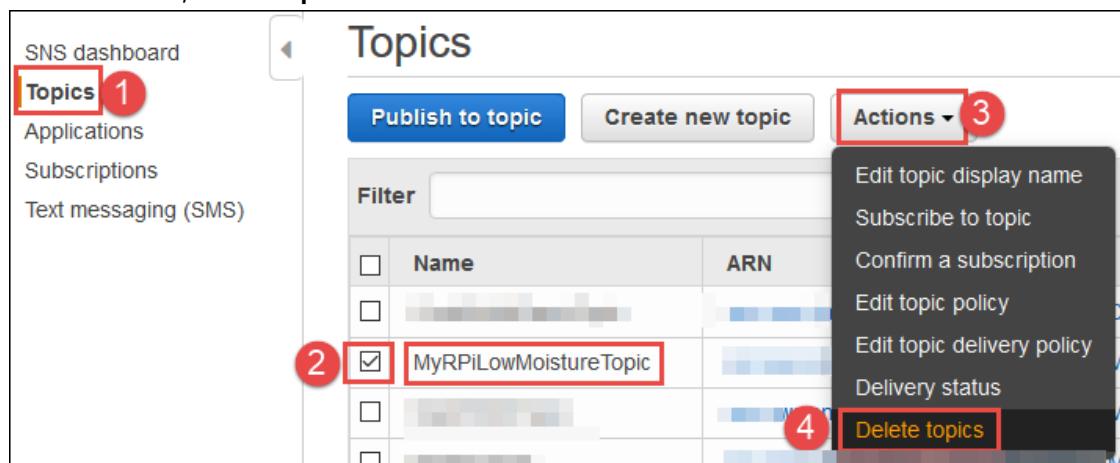
1. Open the Amazon SNS console. To do this, on the AWS navigation bar, choose **Services**. In the **Find a service by name or feature** box, enter **SNS**, and then press **Enter**.
2. In the service navigation pane, choose **Subscriptions**.
3. In the list of subscriptions, select the box for the row where **Topic ARN** contains the name of the related topic (for example, **MyRPiLowMoistureTopic**) and **Endpoint** contains your email address.
4. Choose **Actions, Delete subscriptions**.



- When prompted, choose **Delete**.

#### To delete the related topic in Amazon SNS

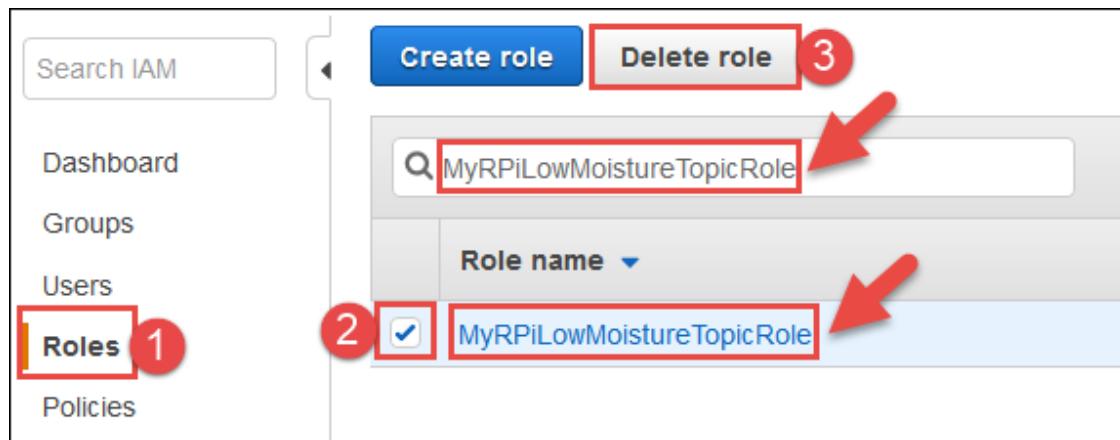
- In the Amazon SNS console, in the service navigation pane, choose **Topics**.
- In the list of topics, select the box for the row where **Name** contains the name of the related topic (for example, **MyRPiLowMoistureTopic**).
- Choose **Actions, Delete topics**.



- When prompted, choose **Delete**.

#### To delete the related Amazon SNS role in IAM

- Open the IAM console. To do this, on the AWS navigation bar, choose **Services**. In the **Find a service by name or feature** box, enter **IAM**, and then press **Enter**.
- In the service navigation pane, choose **Roles**.
- In the list of roles, select the box where **Role name** contains the name of the related role (for example, **MyRPiLowMoistureTopicRole**). If you can't find the role, then in the **Search** box, enter the name of the role. Then press **Enter**.
- Choose **Delete role**.



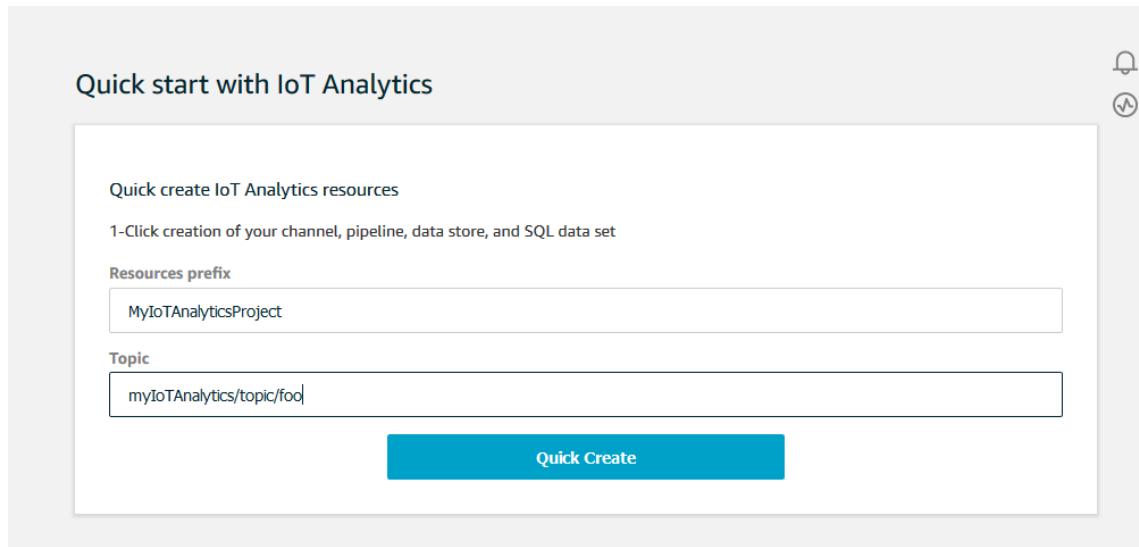
- When prompted, choose **Yes, delete**.

## Next Steps

Learn more about how to use AWS IoT. See the following resources.

- [AWS IoT Developer Guide](#) – Learn more about the concepts in this sample walkthrough, including how to work more extensively with things, rules, and shadows. Then learn about thing types, thing groups, jobs, and services such as AWS IoT Device Defender.
- [AWS IoT Greengrass Developer Guide](#) – Learn how AWS IoT Greengrass enables you to securely run local compute, messaging, data caching, sync, and machine language inference capabilities for connected devices.
- [AWS IoT Analytics User Guide](#) – Learn how AWS IoT Analytics, a fully managed service, makes it easy to run and operationalize sophisticated analytics on massive volumes of IoT data. With AWS IoT Analytics, you don't have to worry about the cost and complexity typically required to build an IoT analytics platform.

The AWS IoT Analytics console also has a **Quick start** feature that allows you to create a channel, data store, pipeline, and data store with one click. Look for this page when you enter the AWS IoT Analytics console:



4. [AWS IoT 1-Click Developer Guide](#) – Learn how to use AWS IoT 1-Click, a service that enables simple devices to trigger [AWS Lambda](#) functions to execute actions.
5. [Amazon FreeRTOS User Guide](#) – Learn how to use Amazon FreeRTOS, an operating system for microcontrollers that makes small, low-power edge devices easy to program, deploy, secure, connect, and manage.
6. [AWS IoT Events Developer Guide](#) – Learn how to use AWS IoT Events to monitor your equipment and device fleets for failures or changes in operation, and to trigger actions when such events occur.

# Managing Devices with AWS IoT

AWS IoT provides a registry that helps you manage *things*. A thing is a representation of a specific device or logical entity. It can be a physical device or sensor (for example, a light bulb or a switch on a wall). It can also be a logical entity like an instance of an application or physical entity that does not connect to AWS IoT but is related to other devices that do (for example, a car that has engine sensors or a control panel).

Information about a thing is stored in the registry as JSON data. Here is an example thing:

```
{  
    "version": 3,  
    "thingName": "MyLightBulb",  
    "defaultClientId": "MyLightBulb",  
    "thingTypeName": "LightBulb",  
    "attributes": {  
        "model": "123",  
        "wattage": "75"  
    }  
}
```

Things are identified by a name. Things can also have attributes, which are name-value pairs you can use to store information about the thing, such as its serial number or manufacturer.

A typical device use case involves the use of the thing name as the default MQTT client ID. Although we do not enforce a mapping between a thing's registry name and its use of MQTT client IDs, certificates, or shadow state, we recommend you choose a thing name and use it as the MQTT client ID for both the registry and the Device Shadow service. This provides organization and convenience to your IoT fleet without removing the flexibility of the underlying device certificate model or shadows.

You do not need to create a thing in the registry to connect a device to AWS IoT. Adding things to the registry allows you to manage and search for devices more easily.

## How to Manage Things with the Registry

You use the AWS IoT console or the AWS CLI to interact with the registry. The following sections show how to use the CLI to work with the registry.

### Create a Thing

The following command shows how to use the AWS IoT **CreateThing** command from the CLI to create a thing:

```
$ aws iot create-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\": {\"wattage\":\"75\", \"model\":\"123\"}}"
```

The **CreateThing** command displays the name and ARN of your new thing:

```
{  
    "thingArn": "arn:aws:iot:us-east-1:123456789012:thing/MyLightBulb",  
    "thingName": "MyLightBulb",  
    "thingId": "12345678abcdefghijklmnopqrstuvwxyz"
```

```
}
```

**Note**

We do not recommend using personally identifiable information in your thing names.

## List Things

You can use the **ListThings** command to list all things in your account:

```
$ aws iot list-things
{
    "things": [
        {
            "attributes": {
                "model": "123",
                "wattage": "75"
            },
            "version": 1,
            "thingName": "MyLightBulb"
        },
        {
            "attributes": {
                "numOfStates": "3"
            },
            "version": 11,
            "thingName": "MyWallSwitch"
        }
    ]
}
```

## Search for Things

You can use the **DescribeThing** command to list information about a thing:

```
$ 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"
    }
}
```

You can use the **ListThings** command to search for all things associated with a thing type name:

```
$ 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"
}
]
```

You can use the **ListThings** command to search for all things that have an attribute with a specific value:

```
$ 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"
        }
    ]
}
```

## Update a Thing

You can use the **UpdateThing** command to update a thing:

```
$ aws iot update-thing --thing-name "MyLightBulb" --attribute-payload "{\"attributes\": {\"wattage\":\"150\", \"model\":\"456\"}}"
```

The **UpdateThing** command does not produce output. You can use the **DescribeThing** command to see the result:

```
$ aws iot describe-thing --thing-name "MyLightBulb"
{
    "attributes": {
        "model": "456",
        "wattage": "150"
    },
    "version": 2,
    "thingName": "MyLightBulb"
}
```

## Delete a Thing

You can use the **DeleteThing** command to delete a thing:

```
$ aws iot delete-thing --thing-name "MyThing"
```

This command returns successfully with no error if the deletion is successful or you specify a thing that doesn't exist.

## Attach a Principal to a Thing

A physical device must have an X.509 certificate to communicate with AWS IoT. You can associate the certificate on your device with the thing in the registry that represents your device. To attach a certificate to your thing, use the **AttachThingPrincipal** command:

```
$ aws iot attach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

The **AttachThingPrincipal** command does not produce any output.

## Detach a Principal from a Thing

You can use the **DetachThingPrincipal** command to detach a certificate from a thing:

```
$ aws iot detach-thing-principal --thing-name "MyLightBulb" --principal "arn:aws:iot:us-east-1:123456789012:cert/a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

The **DetachThingPrincipal** command does not produce any output.

## Thing Types

Thing types allow you to store description and configuration information that is common to all things associated with the same thing type. This simplifies the management of things in the registry. For example, you can define a LightBulb thing type. All things associated with the LightBulb thing type share a set of attributes: serial number, manufacturer, and wattage. When you create a thing of type LightBulb (or change the type of an existing thing to LightBulb) you can specify values for each of the attributes defined in the LightBulb thing type.

Although thing types are optional, their use provides better discovery of things.

- Things with a thing type can have up to 50 attributes.

- Things without a thing type can have up to three attributes.
- A thing can be associated with only one thing type.
- There is no limit on the number of thing types you can create in your account.

Thing types are immutable. You cannot change a thing type name after it has been created. You can deprecate a thing type at any time to prevent new things from being associated with it. You can also delete thing types that have no things associated with them.

## Create a Thing Type

You can use the **CreateThingType** command to create a thing type:

```
$ aws iot create-thing-type  
      --thing-type-name "LightBulb" --thing-type-properties  
      "thingTypeDescription=light bulb type, searchableAttributes=wattage,model"
```

The **CreateThingType** command returns a response that contains the thing type and its ARN:

```
{  
    "thingTypeName": "LightBulb",  
    "thingTypeId": "df9c2d8c-894d-46a9-8192-9068d01b2886",  
    "thingTypeArn": "arn:aws:iot:us-west-2:123456789012:thingtype/LightBulb"  
}
```

## List Thing Types

You can use the **ListThingTypes** command to list thing types:

```
$ aws iot list-thing-types
```

The **ListThingTypes** command returns a list of the thing types defined in your AWS account:

```
{  
    "thingTypes": [  
        {  
            "thingTypeName": "LightBulb",  
            "thingTypeProperties": {  
                "searchableAttributes": [  
                    "wattage",  
                    "model"  
                ],  
                "thingTypeDescription": "light bulb type"  
            },  
            "thingTypeMetadata": {  
                "deprecated": false,  
                "creationDate": 1468423800950  
            }  
        }  
    ]  
}
```

## Describe a Thing Type

You can use the **DescribeThingType** command to get information about a thing type:

```
$ aws iot describe-thing-type --thing-type-name "LightBulb"
```

The **DescribeThingType** command returns information about the specified type:

```
{  
    "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  
    }  
}
```

## Associate a Thing Type with a Thing

You can use the **CreateThing** command to specify a thing type when you create a thing:

```
$ aws iot create-thing --thing-name "MyLightBulb" --thing-type-name "LightBulb" --  
attribute-payload "{\"attributes\": {\"wattage\":\"75\", \"model\":\"123\"}}"
```

You can use the **UpdateThing** command at any time to change the thing type associated with a thing:

```
$ aws iot update-thing --thing-name "MyLightBulb"  
          --thing-type-name "LightBulb" --attribute-payload "{\"attributes\"::  
          {\"wattage\":\"75\", \"model\":\"123\"}}"
```

You can also use the **UpdateThing** command to disassociate a thing from a thing type.

## Deprecate a Thing Type

Thing types are immutable. They cannot be changed after they are defined. You can, however, deprecate a thing type to prevent users from associating any new things with it. All existing things associated with the thing type are unchanged.

To deprecate a thing type, use the **DeprecateThingType** command:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType"
```

You can use the **DescribeThingType** command to see the result:

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{  
    "thingTypeName": "StopLight",  
    "thingTypeProperties": {  
        "searchableAttributes": [  
            "wattage",  
            "model"  
        ]  
    }  
}
```

```
        "numOfLights",
        "model"
    ],
    "thingTypeDescription": "traffic light type",
},
"thingTypeMetadata": {
    "deprecated": true,
    "creationDate": 1468425854308,
    "deprecationDate": 1468446026349
}
}
```

Deprecating a thing type is a reversible operation. You can undo a deprecation by using the --undo-deprecate flag with the **DeprecateThingType** CLI command:

```
$ aws iot deprecate-thing-type --thing-type-name "myThingType" --undo-deprecate
```

You can use the **DescribeThingType** CLI command to see the result:

```
$ aws iot describe-thing-type --thing-type-name "StopLight":
```

```
{
    "thingTypeName": "StopLight",
    "thingTypeArn": "arn:aws:iot:us-east-1:123456789012:thingtype/StopLight",
    "thingTypeId": "12345678abcdefghijklmnop12345678",
    "thingTypeProperties": {
        "searchableAttributes": [
            "wattage",
            "numOfLights",
            "model"
        ],
        "thingTypeDescription": "traffic light type"
    },
    "thingTypeMetadata": {
        "deprecated": false,
        "creationDate": 1468425854308,
    }
}
```

## Delete a Thing Type

You can delete thing types only after they have been deprecated. To delete a thing type, use the **DeleteThingType** command:

```
$ aws iot delete-thing-type --thing-type-name "StopLight"
```

### Note

You must wait five minutes after you deprecate a thing type before you can delete it.

## Thing Groups

Thing groups allow you to manage several things at once by categorizing them into groups. Groups can also contain groups — you can build a hierarchy of groups. You can attach a policy to a parent group and it will be inherited by its child groups, and by all of the things in the group and in its child groups as well. This makes control of permissions easy for large numbers of things.

Here are the things you can do with thing groups:

- Create, describe or delete a group.
  - Add a thing to a group, or to more than one group.
  - Remove a thing from a group.
  - List the groups you have created.
  - List all child groups of a group (its direct and indirect descendants.)
  - List the things in a group, including all the things in its child groups.
  - List all ancestor groups of a group (its direct and indirect parents.)
  - Add, delete or update the attributes of a group. (Attributes are name-value pairs you can use to store information about a group.)
  - Attach or detach a policy to or from a group.
  - List the policies attached to a group.
  - List the policies inherited by a thing (by virtue of the policies attached to its group, or one of its parent groups.)
  - Configure logging options for things in a group (see [Configure AWS IoT Logging \(p. 631\)](#).)
  - Create jobs that will be sent to and executed on every thing in a group and its child groups (see [Jobs \(p. 354\)](#).)

Here are some limitations of thing groups:

- A group can have at most one direct parent.
  - If a group will be a child of another group, this must be specified at the time it is created.
  - You can't change a group's parent later. (So be sure to plan your group hierarchy and create a parent group before creating any child groups it will contain.)
  - You cannot add a thing to more than 10 groups.
  - You cannot add a thing to more than one group in the same hierarchy. (In other words, you cannot add a thing to two groups which share a common parent.)
  - You cannot rename a group.
  - Thing group names can't contain international characters, such as û, é and ñ.

Attaching and detaching policies to groups can enhance the security of your AWS IoT operations in a number of significant ways. The per device method of attaching a policy to a certificate, which is then attached to a thing, is time consuming and makes it difficult to quickly update or change policies across a fleet of devices. Having a policy attached to the thing's group saves steps when it is time to rotate the certificates on a thing. And policies are dynamically applied to things when they change group membership, so you aren't required to re-create a complex set of permissions each time a device changes membership in a group.

## Create a Thing Group

You can use the **CreateThingGroup** command to create a thing group:

```
$ aws iot create-thing-group --thing-group-name LightBulbs
```

The **CreateThingGroup** command returns a response that contains the thing group, its ID, and ARN:

```
{  
    "thingGroupName": "LightBulbs",  
    "thingGroupId": "abcdefghijklmnopqrstuvwxyz12345678qrstuvwxyz",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
```

}

**Note**

We do not recommend using personally identifiable information in your thing group names.

Here is an example that specifies a parent of the thing group when it is created:

```
$ aws iot create-thing-group --thing-group-name RedLights --parent-group-name LightBulbs
```

As before, the **CreateThingGroup** command returns a response that contains the thing group, its ID, and ARN:

```
{  
    "thingGroupName": "RedLights",  
    "thingGroupId": "abcdefghijklmnopqrstuvwxyz12345678ijklmnop12345678qrstuvwxyz",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",  
}
```

**Important**

Keep in mind the following limits when creating group hierarchies:

- A group can have at most one direct parent.
- A group may have no more than 100 direct child groups.
- The maximum depth of a group hierarchy is 7.
- A group can have up to 50 attributes. (Attributes are name-value pairs you can use to store information about a group.) Each attribute name can contain up to 128 characters and each value up to 800 characters.

## Describe a Thing Group

You can use the **DescribeThingGroup** command to get information about a thing group:

```
$ aws iot describe-thing-group --thing-group-name RedLights
```

The **DescribeThingGroup** command returns information about the specified group:

```
{  
    "thingGroupName": "RedLights",  
    "thingGroupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights",  
    "thingGroupId": "12345678abcdefghijklmnopqrstuvwxyz12345678ijklmnop12345678",  
    "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": {  
        "attributeCount": 0  
    }  
}
```

```
"attributePayload": {  
    "attributes": {  
        "brightness": "3400_lumens"  
    },  
    "thingGroupDescription": "string"  
},  
}
```

## Add a Thing to a Thing Group

You can use the **AddThingToThingGroup** command to add a thing to a group:

```
$ aws iot add-thing-to-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

The **AddThingToThingGroup** command does not produce any output.

**Important**

You can add a thing to a maximum of 10 groups. But you cannot add a thing to more than one group in the same hierarchy. (In other words, you cannot add a thing to two groups which share a common parent.)

If a thing belongs to 10 thing groups, and one or more of those groups is a dynamic thing group, you can use the [overrideDynamicGroups](#) flag to make static groups take priority over dynamic groups.

## Remove a Thing from a Thing Group

You can use the **RemoveThingFromThingGroup** command to remove a thing from a group:

```
$ aws iot remove-thing-from-thing-group --thing-name MyLightBulb --thing-group-name RedLights
```

The **RemoveThingFromThingGroup** command does not produce any output.

## List Things in a Thing Group

You can use the **ListThingsInThingGroup** command to list the things that belong to a group:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs
```

The **ListThingsInThingGroup** command returns a list of the things in the given group:

```
{  
    "things": [  
        "TestThingA"  
    ]  
}
```

With the **--recursive** parameter, you can list things belonging to a group and those in any of its child groups as well:

```
$ aws iot list-things-in-thing-group --thing-group-name LightBulbs --recursive
```

```
{
```

```
    "things": [
        "TestThingA",
        "MyLightBulb"
    ]}
```

**Note**

This operation is [eventually consistent](#). In other words, changes to the thing group might not be reflected immediately.

## List Thing Groups

You can use the **ListThingGroups** command to list groups you have created:

```
$ aws iot list-thing-groups
```

The **ListThingGroups** command returns a list of the groups defined in your AWS account:

```
{
    "thingGroups": [
        {
            "groupName": "LightBulbs",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"
        },
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        },
        {
            "groupName": "RedLEDLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLEDLights"
        },
        {
            "groupName": "RedIncandescentLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/
RedIncandescentLights"
        },
        {
            "groupName": "ReplaceableObjects",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/ReplaceableObjects"
        }
    ]
}
```

Use the optional filters to list those groups that have a given group as parent (`--parent-group`) or groups whose name begins with a given prefix (`--name-prefix-filter`.) The `--recursive` parameter allows you to list all children groups, not just direct child groups of a thing group:

```
$ aws iot list-thing-groups --parent-group LightBulbs
```

In this case, the **ListThingGroups** command returns a list of the direct child groups of the thing group defined in your AWS account:

```
{
    "childGroups": [
        {
            "groupName": "RedLights",
            "groupArn": "arn:aws:iot:us-west-2:123456789012:thinggroup/RedLights"
        }
    ]
}
```

```
    ]  
}
```

Use the `--recursive` parameter with the **ListThingGroups** command to list all child groups of a thing group, not just direct children:

```
$ aws iot list-thing-groups --parent-group LightBulbs --recursive
```

The **ListThingGroups** command returns a list of all child groups of the thing group:

```
{
  "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

This operation is [eventually consistent](#). In other words, changes to the thing group might not be reflected immediately.

## List Groups for a Thing

You can use the **ListThingGroupsForThing** command to list the groups a thing belongs to, including any parent groups:

```
$ aws iot list-thing-groups-for-thing --thing-name MyLightBulb
```

The **ListThingGroupsForThing** command returns a list of the thing groups this thing belongs to, including any parent groups:

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

## Update a Thing Group

You can use the **UpdateThingGroup** command to update the attributes of a thing group:

```
$ aws iot update-thing-group --thing-group-name "LightBulbs" --thing-group-properties "thingGroupDescription=\"this is a test group\", attributePayload=\\"{\\"attributes \\"=\\\"Owner\\\"=\\\"150\\\",\\\"modelNames\\\"=\\\"456\\\"}\\\""
```

The **UpdateThingGroup** command returns a response that contains the group's version number after the update:

```
{  
    "version": 4  
}
```

**Note**

A group can have up to 50 attributes.

## Delete a Thing Group

To delete a thing group, use the **DeleteThingGroup** command:

```
$ aws iot delete-thing-group --thing-group-name "RedLights"
```

The **DeleteThingGroup** command does not produce any output.

**Important**

If you try to delete a thing group that has child thing groups, you receive an 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.
```

You must delete any child groups first before you delete the group.

You can delete a group that has child things, but any permissions granted to the things by membership in the group no longer apply. Before deleting a group that has a policy attached, check carefully that removing those permissions would not stop the things in the group from being able to function properly. Also, note that commands that show which groups a thing belongs to (for example, **ListGroupsForThing**) might continue to show the group while records in the cloud are being updated.

## Attach a Policy to a Thing Group

You can use the **AttachPolicy** command to attach a policy to a thing group and so, by extension, to all things in that group and things in any of its child groups:

```
$ aws iot attach-policy \  
--target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs" \  
--policy-name "myLightBulbPolicy"
```

The **AttachPolicy** command does not produce any output

**Important**

You can attach a maximum number of two policies to a group.

**Note**

We do not recommend using personally identifiable information in your policy names.

The `--target` parameter can be a thing group ARN (as above), a certificate ARN, or an Amazon Cognito Identity. For more information about policies, certificates and authentication, see [Security and Identity for AWS IoT \(p. 179\)](#).

## Detach a Policy from a Thing Group

You can use the **DetachPolicy** command to detach a policy from a group and so, by extension, to all things in that group and things in any of its child groups:

```
$ aws iot detach-policy --target "arn:aws:iot:us-west-2:123456789012:thinggroup/LightBulbs"  
--policy-name "myLightBulbPolicy"
```

The **DetachPolicy** command does not produce any output.

## List the Policies Attached to a Thing Group

You can use the **ListAttachedPolicies** command to list the policies attached to a group:

```
$ aws iot list-attached-policies --target "arn:aws:iot:us-west-2:123456789012:thinggroup/  
RedLights"
```

The `--target` parameter can be a thing group ARN (as above), a certificate ARN, or an Amazon Cognito Identity.

Add the optional `--recursive` parameter to include all policies attached to the group's parent groups as well.

The **ListAttachedPolicies** command returns a list of policies:

```
{  
    "policies": [  
        "MyLightBulbPolicy"  
        ...  
    ]  
}
```

## List the Groups for a Policy

You can use the **ListTargetsForPolicy** command to list the targets, including any groups, that a policy is attached to:

```
$ aws iot list-targets-for-policy --policy-name "MyLightBulbPolicy"
```

Add the optional `--page-size number` parameter to specify the maximum number of results to be returned for each query, and the `--marker string` parameter on subsequent calls to retrieve the next set of results, if any.

The **ListTargetsForPolicy** command returns a list of targets and the token to use to retrieve more results:

```
{  
    "nextMarker": "string",  
    "targets": [ "string" ... ]  
}
```

## Get Effective Policies for a Thing

You can use the **GetEffectivePolicies** command to list the policies in effect for a thing, including the policies attached to any groups the thing belongs to (whether the group is a direct parent or indirect ancestor):

```
$ aws iot get-effective-policies \
--thing-name "MyLightBulb" \
--principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847"
```

Use the `--principal` parameter to specify the ARN of the certificate attached to the thing. If you are using Amazon Cognito Identity authentication, use the `--cognito-identity-pool-id` parameter and, optionally, add the `--principal` parameter to specify a specific Cognito Identity. (If you specify only the `--cognito-identity-pool-id` then the policies associated with that identity pool's role for unauthenticated users are returned. If you use both, the policies associated with that identity pool's role for authenticated users are returned.)

The `--thing-name` parameter is optional and may be used instead of the `--principal` parameter. When used, the policies attached to any group the thing belongs to, and the policies attached to any parent groups of these groups (up to the root group in the hierarchy) will be returned.

The **GetEffectivePolicies** command returns a list of policies:

```
{
  "effectivePolicies": [
    {
      "policyArn": "string",
      "policyDocument": "string",
      "policyName": "string"
    }
    ...
  ]
}
```

## Test Authorization for MQTT Actions

You can use the **TestAuthorization** command to test whether an MQTT action is allowed for a thing:

```
aws iot test-authorization \
--principal "arn:aws:iot:us-east-1:123456789012:cert/
a0c01f5835079de0a7514643d68ef8414ab739a1e94ee4162977b02b12842847" \
--auth-infos "{\"actionType\": \"PUBLISH\", \"resources\": [ \"arn:aws:iot:us-
east-1:123456789012:topic/my/topic\"]}"
```

Use the `--principal` parameter to specify the ARN of the certificate attached to the thing. If using Amazon Cognito Identity authentication, specify a Cognito Identity as the `--principal` or use the `--cognito-identity-pool-id` parameter, or both. (If you specify only the `--cognito-identity-pool-id` then the policies associated with that identity pool's role for unauthenticated users are considered. If you use both, the policies associated with that identity pool's role for authenticated users are considered.)

Specify one or more MQTT actions you want to test by listing sets of resources and action types following the `--auth-infos` parameter. The `actionType` field should contain "PUBLISH", "SUBSCRIBE", "RECEIVE", or "CONNECT". The `resources` field should contain a list of resource ARNs. See [AWS IoT Policies \(p. 193\)](#) for more information.

You can test the effects of adding policies by specifying them with the `--policy-names-to-add` parameter. Or you can test the effects of removing policies by them with the `--policy-names-to-skip` parameter.

You can use the optional `--client-id` parameter to further refine your results.

The **TestAuthorization** command returns details on actions that were allowed or denied for each set of `--auth-infos` queries you specified:

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

## Dynamic Thing Groups

Dynamic thing groups update group membership through search queries. Using dynamic thing groups, you can change the way you interact with things depending on their connectivity, registry, or shadow data.

You can apply a policy to a dynamic thing group. A thing can belong to at most 10 dynamic groups. If a search query string defines a dynamic thing group that contains a thing that already belongs to 10 dynamic thing groups, the policy is not applied to that thing.

Because dynamic thing groups are tied to your fleet index, you must enable the fleet indexing service to use them. You can preview the things in a dynamic thing group before you create the group with a

fleet indexing search query. For more information, see [Fleet Indexing Service](#) and [Fleet Indexing Service: Query Syntax](#).

You can specify a dynamic thing group as a target for a job. Only things that meet the criteria that define the dynamic thing group perform the job.

For example, suppose that you want to update the firmware on your devices, but, to minimize the chance that the update is interrupted, you only want to update firmware on devices with battery life greater than 80%. You can create a dynamic thing group that only includes devices with a reported battery life above 80%, and you can use that dynamic thing group as the target for your firmware update job. Only devices that meet your battery life criteria receive the firmware update. As devices reach the 80% battery life criteria, they are added to the dynamic thing group and receive the firmware update.

For more information about specifying thing groups as job targets, see [Using the AWS IoT APIs](#).

Dynamic thing groups differ from static thing groups in the following ways:

- Thing membership is not explicitly defined. To create a dynamic thing group, you must define a query string that defines group membership.
- Dynamic thing groups cannot be part of a hierarchy.
- You use a different set of commands to create, update, and delete dynamic thing groups. For all other operations, the same commands that you use to interact with static thing groups can be used to interact with dynamic thing groups.
- A single account can have no more than 100 dynamic thing groups defined.

For more information about static thing groups, see [Thing Groups \(p. 160\)](#).

As an example, suppose we create a dynamic group that contains all rooms in a data warehouse whose temperature is greater than 60 degrees Fahrenheit. When a room's temperature is 61 degrees or higher, it is added to the RoomTooWarm dynamic thing group. All rooms in the RoomTooWarm dynamic thing group have cooling fans turned on. When a room's temperature falls to 60 degrees or lower, it is removed from the dynamic thing group and its fan would be turned off.

## Create a Dynamic Thing Group

Use the **CreateDynamicThingGroup** command to create a dynamic thing group. To create a dynamic thing group for the room too warm scenario you would use the **create-dynamic-thing-group** CLI command:

```
$ aws iot create-dynamic-thing-group --thing-group-name "RoomTooWarm" --query-string "attributes.temperature>60"
```

### Note

We do not recommend using personally identifiable information in your dynamic thing group names.

The **CreateDynamicThingGroup** command returns a response that contains the index name, query string, query version, thing group name, thing group ID, and thing group ARN:

```
{  
    "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": "abcdefghijklmnopqrstuvwxyz"  
}
```

Dynamic thing group creation is not instantaneous. The dynamic thing group backfill takes time to complete. When a dynamic thing group is created, the status of the group is set to **BUILDING**. When the backfill is complete, the status changes to **ACTIVE**. To check the status of your dynamic thing group, use the [DescribeThingGroup](#) command.

## Describe a Dynamic Thing Group

Use the **DescribeThingGroup** command to get information about a dynamic thing group:

```
$ aws iot describe-thing-group --thing-group-name "RoomTooWarm"
```

The **DescribeThingGroup** command returns information about the specified group:

```
{  
    "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"  
}
```

Running **DescribeThingGroup** on a dynamic thing group returns attributes that are specific to dynamic thing groups, such as the `queryString` and the `status`.

The status of a dynamic thing group can take the following values:

**ACTIVE**

The dynamic thing group is ready for use.

**BUILDING**

The dynamic thing group is being created, and thing membership is being processed.

**REBUILDING**

The dynamic thing group's membership is being updated, following the adjustment of the group's search query.

### Note

After you create a dynamic thing group, you can use the group, regardless of its status. Only dynamic thing groups with an **ACTIVE** status include all of the things that match the search query for that dynamic thing group. Dynamic thing groups with **BUILDING** and **REBUILDING** statuses might not include all of the things that match the search query.

## Update a Dynamic Thing Group

Use the **UpdateDynamicThingGroup** command to update the attributes of a dynamic thing group, including the group's search query. The following command updates the thing group description and the query string changing the membership criteria to temperature > 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"
```

The **UpdateDynamicThingGroup** command returns a response that contains the group's version number after the update:

```
{  
    "version": 2  
}
```

Dynamic thing group updates are not instantaneous. The dynamic thing group backfill takes time to complete. When a dynamic thing group is updated, the status of the group changes to REBUILDING while the group updates its membership. When the backfill is complete, the status changes to ACTIVE. To check the status of your dynamic thing group, use the **DescribeThingGroup** command.

## Delete a Dynamic Thing Group

Use the **DeleteDynamicThingGroup** command to delete a dynamic thing group:

```
$ aws iot delete-dynamic-thing-group --thing-group-name "RoomTooWarm"
```

The **DeleteDynamicThingGroup** command does not produce any output.

Before you delete a group that has a policy attached, be sure that removing those permissions does not stop the things in the group from functioning properly. Commands that show which groups a thing belongs to (for example, **ListGroupsForThing**) might continue to show the group while records in the cloud are being updated.

## Limitations and Conflicts

Dynamic thing groups share some of the same limitations as static thing groups:

- A thing group can have up to 50 attributes.
- A thing can belong up to a maximum of 10 thing groups.
- Thing groups cannot be renamed.
- Thing group names cannot contain international characters, such as ü, é, and ñ.

When using dynamic thing groups, keep the following in mind.

### Older dynamic thing groups take priority over newer ones

By default, if a thing belongs to 10 thing groups, you cannot add it to additional groups. If a conflict in membership arises among dynamic thing groups when you create or update a dynamic thing group, older dynamic thing groups take priority over newer ones.

### With `overrideDynamicGroups` enabled, static groups take priority over dynamic groups

By default, if a thing belongs to 10 thing groups, you cannot add the thing to additional groups. If you are updating thing membership with the [AddThingToThingGroup](#) or [UpdateThingGroupsForThing](#) commands, you can use the `overrideDynamicGroups` flag to make static thing groups take priority

over dynamic thing groups. With `overrideDynamicGroups` enabled, if a thing belongs to 10 thing groups, and one or more of those groups is dynamic, adding the thing to a static thing group removes it from the newest dynamic thing group.

For example, suppose that you create a dynamic thing group named `DynamicGroup1`, and then you create nine more dynamic thing groups, with `DynamicGroup10` being the last group that you created. If `Thing1` belongs to all 10 dynamic thing groups, manually adding `Thing1` to a static group with `OverrideDynamicGroups` enabled removes the thing from `DynamicGroup10`.

## Applying Policies to Members of a Dynamic Thing Group

A policy can be applied to a dynamic thing group. A thing can belong to at most 10 dynamic groups. If a search query string defines a dynamic thing group that contains a thing that already belongs to 10 dynamic thing groups, the policy is not applied to that thing.

## Dynamic thing group membership is eventually consistent

Only the final state of a thing is evaluated for the registry. Intermediary states can be skipped if states are updated rapidly. Avoid associating a rule, job, or policy with a dynamic thing group whose membership depends on an intermediary state.

## The fleet indexing service must be enabled

The fleet indexing service must be enabled and the fleet indexing backfill must be complete before you can create and use dynamic thing groups. Expect a delay after you enable the fleet indexing service. The backfill can take some time to complete. The more things that you have registered, the longer the backfill process takes. After you enable the fleet indexing service for dynamic thing groups, you cannot disable it until you delete all of your dynamic thing groups.

### Note

If you have permissions to query the fleet index, you can access the data of things across the entire fleet.

# Tagging Your AWS IoT Resources

To help you manage and organize your thing groups, thing types, topic rules, jobs, scheduled audits and security profiles you can optionally assign your own metadata to each of these resources in the form of tags. This section describes tags and shows you how to create them.

To help you manage your costs related to things, you can create [billing groups \(p. 176\)](#) that contain things. You can then assign tags containing your metadata to each of these billing groups. This section also discusses billing groups and the commands available to create and manage them.

## Tag Basics

Tags enable you to categorize your AWS IoT resources in different ways, for example, by purpose, owner, or environment. This is useful when you have many resources of the same type — you can quickly identify a specific resource based on the tags you've assigned to it. Each tag consists of a key and optional value, both of which you define. For example, you could define a set of tags for your thing types that helps you track devices by type. We recommend that you create a set of tag keys that meets your needs for each kind of resource. Using a consistent set of tag keys makes it easier for you to manage your resources.

You can search for and filter resources based on the tags you add or apply. You can also use billing group tags to categorize and track your costs. You can also use tags to control access to your resources as described in [Using Tags with IAM Policies \(p. 176\)](#).

For ease of use, the Tag Editor in the AWS Management Console provides a central, unified way to create and manage your tags. For more information, see [Working with Tag Editor](#) in [Working with the AWS Management Console](#).

You can also work with tags using the AWS CLI and the AWS IoT API. You can associate tags with thing groups, thing types, topic rules, jobs, security profiles, and billing groups when you create them by using the "Tags" field in the following commands:

- [CreateBillingGroup \(p. 679\)](#)
- [CreateDynamicThingGroup \(p. 682\)](#)
- [CreateJob \(p. 686\)](#)
- [CreateOTAUpdate \(p. 692\)](#)
- [CreateScheduledAudit \(p. 702\)](#)
- [CreateSecurityProfile \(p. 705\)](#)
- [CreateStream \(p. 709\)](#)
- [CreateThingGroup \(p. 714\)](#)
- [CreateThingType \(p. 717\)](#)
- [CreateTopicRule \(p. 719\)](#)

You can add, modify, or delete tags for existing resources that support tagging by using the following commands:

- [TagResource \(p. 965\)](#)
- [ListTagsForResource \(p. 899\)](#)

- [UntagResource \(p. 973\)](#)

You can edit tag keys and values, and you can remove tags from a resource at any time. You can set the value of a tag to an empty string, but you can't set the value of a tag to null. If you add a tag that has the same key as an existing tag on that resource, the new value overwrites the old value. If you delete a resource, any tags associated with the resource are also deleted.

Additional information is available in [AWS Tagging Strategies](#).

## Tag Restrictions and Limitations

The following basic restrictions apply to tags:

- Maximum number of tags per resource — 50
- Maximum key length — 127 Unicode characters in UTF-8
- Maximum value length — 255 Unicode characters in UTF-8
- Tag keys and values are case-sensitive.
- Do not use the "aws:" prefix in your tag names or values because it's reserved for AWS use. You can't edit or delete tag names or values with this prefix. Tags with this prefix don't count against your tags per resource limit.
- If your tagging schema is used across multiple services and resources, remember that other services may have restrictions on allowed characters. Generally, allowed characters are: letters, spaces, and numbers representable in UTF-8, and the following special characters: + - = . \_ : / @.

## Using Tags with IAM Policies

You can apply tag-based resource-level permissions in the IAM policies you use for AWS IoT API actions. This gives you better control over what resources a user can create, modify, or use. You use the Condition element (also called the Condition block) with the following condition context keys and values in an IAM policy to control user access (permissions) based on a resource's tags:

- Use `aws:ResourceTag/tag-key: tag-value` to allow or deny user actions on resources with specific tags.
- Use `aws:RequestTag/tag-key: tag-value` to require that a specific tag be used (or not used) when making an API request to create or modify a resource that allows tags.
- Use `aws:TagKeys: [tag-key, ...]` to require that a specific set of tag keys be used (or not used) when making an API request to create or modify a resource that allows tags.

### Note

The condition context keys and values in an IAM policy apply only to those AWS IoT actions where an identifier for a resource capable of being tagged is a required parameter. For example, the use of [DescribeEndpoint \(p. 773\)](#) will not be allowed or denied on the basis of condition context keys and values because no taggable resource (thing groups, thing types, topic rules, jobs, or security profile) is referenced in this request.

[Controlling Access Using Tags](#) in the *AWS Identity and Access Management User Guide* has additional information on using tags. The [IAM JSON Policy Reference](#) section of that guide has detailed syntax, descriptions, and examples of the elements, variables, and evaluation logic of JSON policies in IAM.

The following example policy applies two tag-based restrictions. An IAM user restricted by this policy:

- Cannot give a resource the tag "env=prod" (in the example, see the line "aws:RequestTag/env" : "prod")

- Cannot modify or access a resource that has an existing tag "env=prod" (in the example, see the line "aws:ResourceTag/env" : "prod").

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

You can also specify multiple tag values for a given tag key by enclosing them in a list, like this:

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

**Note**

If you allow or deny users access to resources based on tags, you must consider explicitly denying users the ability to add those tags to or remove them from the same resources. Otherwise, it's possible for a user to circumvent your restrictions and gain access to a resource by modifying its tags.

## Billing Groups

AWS IoT doesn't allow you to directly apply tags to individual things, but it does allow you to place things in billing groups and to apply tags to these. For AWS IoT, allocation of cost and usage data based on tags is limited to billing groups.

The following commands are available:

- [AddThingToBillingGroup \(p. 662\)](#) adds a thing to a billing group.

- [CreateBillingGroup \(p. 679\)](#) creates a billing group.
- [DeleteBillingGroup \(p. 735\)](#) deletes the billing group.
- [DescribeBillingGroup \(p. 764\)](#) returns information about a billing group.
- [ListBillingGroups \(p. 864\)](#) lists the billing groups you have created.
- [ListThingsInBillingGroup \(p. 915\)](#) lists the things you have added to the given billing group.
- [RemoveThingFromBillingGroup \(p. 933\)](#) removes the given thing from the billing group.
- [UpdateBillingGroup \(p. 978\)](#) updates information about the billing group.
- [CreateThing \(p. 712\)](#) allows you to specify a billing group for the thing when you create it.
- [DescribeThing \(p. 798\)](#) returns the description of a thing including the billing group the thing belongs to, if any.

## Viewing Cost Allocation and Usage Data

You can use billing group tags to categorize and track your costs. When you apply tags to billing groups (and so to the things they include), AWS generates a cost allocation report as a comma-separated value (CSV) file with your usage and costs aggregated by your tags. You can apply tags that represent business categories (such as cost centers, application names, or owners) to organize your costs across multiple services. For more information about using tags for cost allocation, see [Use Cost Allocation Tags](#) in the [AWS Billing and Cost Management User Guide](#).

### Note

To accurately associate usage and cost data with those things you have placed in billing groups, each device or application must:

- Be registered as a thing in AWS IoT (see [Managing Devices with AWS IoT \(p. 154\)](#)).
- Connect to the AWS IoT message broker via MQTT using only the thing's name as the client ID (see [Message Broker for AWS IoT \(p. 233\)](#)).
- Authenticate using a client certificate associated with the thing.

The following pricing dimensions are available for billing groups (based on the activity of things associated with the billing group):

- Connectivity (based on the thing name used as the client ID to connect)
- Messaging (based on messages inbound from, and outbound to, a thing; MQTT only)
- Shadow operations (based on the thing whose message triggered a shadow update)
- Rules triggered (based on the thing whose inbound message triggered the rule; does not apply to those rules triggered by MQTT lifecycle events)
- Thing index updates (based on the thing that was added to the index)
- Remote actions (based on the thing updated)
- [Detect \(p. 534\)](#) reports (based on the thing whose activity is reported)

Cost and usage data based on tags (and reported for a billing group) doesn't reflect the following activities:

- Device registry operations (including updates to things, thing groups, and thing types; see [Managing Devices with AWS IoT \(p. 154\)](#))
- Thing group index updates (when adding a thing group)
- Index search queries
- [Device Provisioning \(p. 461\)](#)
- [Audit \(p. 483\)](#) reports

## Limits

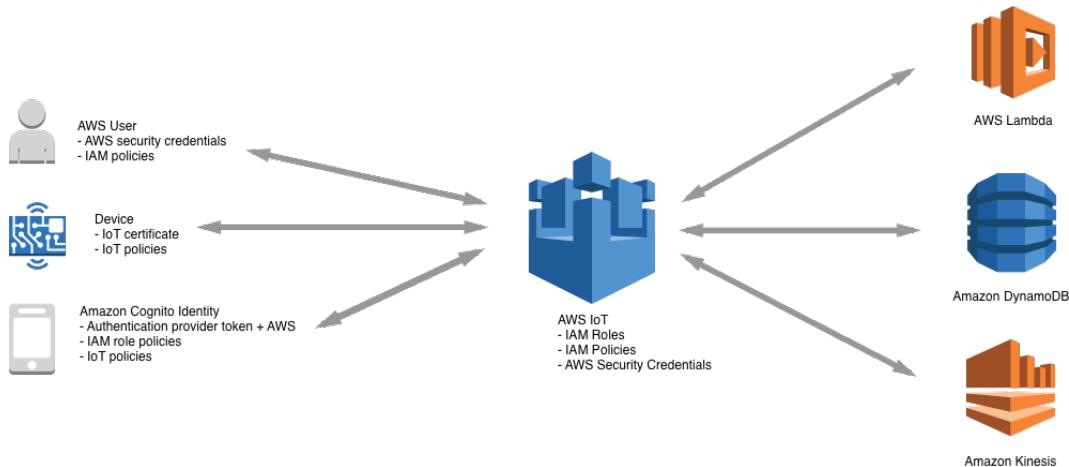
- A thing can belong to exactly one billing group.
- Unlike thing groups, billing groups cannot be organized into hierarchies.
- In order for its usage to be registered for tagging or billing purposes, a device must:
  - Be registered as a thing in AWS IoT.
  - Communicate with AWS IoT using MQTT only.
  - Authenticate with AWS IoT using only its thing name as the clientID.
  - Use only an X.509 certificate or Amazon Cognito Identity to authenticate.

Additional information can be found in [Managing Devices with AWS IoT \(p. 154\)](#), [Security and Identity for AWS IoT \(p. 179\)](#), and [Device Provisioning \(p. 461\)](#). The API command [AttachThingPrincipal \(p. 669\)](#), can be used to attach a certificate or other credential to a thing.

- The maximum number of billing groups per account is 20,000.

# Security and Identity for AWS IoT

Each connected device must have a credential to access the message broker or the Device Shadow service. All traffic to and from AWS IoT must be encrypted over Transport Layer Security (TLS). Device credentials must be kept safe in order to send data securely to the message broker. AWS Cloud security mechanisms protect data as it moves between AWS IoT and other devices or AWS services.



- You are responsible for managing device credentials (X.509 certificates, AWS credentials) on your devices and policies in AWS IoT. You are responsible for assigning unique identities to each device and managing the permissions for a device or group of devices.
- Your devices should connect using X.509 certificates or Amazon Cognito identities over a secure connection according to the AWS IoT connection model. During research and development, and for some applications that make API calls or use WebSockets you can also use IAM users and groups, or custom authentication tokens.
- When using AWS IoT authentication, the message broker authenticates and authorizes all actions in your account. The message broker is responsible for authenticating your devices, securely ingesting device data, and adhering to the access permissions you place on devices using policies.
- When using custom authentication, a custom authorizer is responsible for authenticating your devices and providing an AWS IoT/IAM policy to authorize actions in your account.
- The AWS IoT rules engine forwards device data to other devices and other AWS services according to rules you define. It uses AWS access management systems to securely transfer data to its final destination.

## AWS IoT Authentication

AWS IoT supports four types of identity principals for authentication:

- X.509 certificates
- IAM users, groups, and roles
- Amazon Cognito identities
- Federated identities

These identities can be used with mobile applications, web applications, or desktop applications. They can even be used by a user typing AWS IoT CLI commands. Typically, AWS IoT devices use X.509

certificates, while mobile applications use Amazon Cognito identities. Web and desktop applications use IAM or federated identities. CLI commands use IAM.

## X.509 Certificates

X.509 certificates are digital certificates that use the X.509 public key infrastructure standard to associate a public key with an identity contained in a certificate. X.509 certificates are issued by a trusted entity called a certification authority (CA). The CA maintains one or more special certificates called CA certificates that it uses to issue X.509 certificates. Only the certification authority has access to CA certificates.

### Note

We recommend that each device be given a unique certificate to enable fine-grained management including certificate revocation.

Devices must support rotation and replacement of certificates in order to ensure smooth operation as certificates expire.

AWS IoT supports the following certificate-signing algorithms:

- SHA256WITHRSA
- SHA384WITHRSA
- SHA384WITHRSA
- SHA512WITHRSA
- RSASSAPSS
- DSA\_WITH\_SHA256
- ECDSA-WITH-SHA256
- ECDSA-WITH-SHA384
- ECDSA-WITH-SHA512

Certificates provide several benefits over other identification and authentication mechanisms. Certificates enable asymmetric keys to be used with devices. This means you can burn private keys into secure storage on a device. This way, sensitive cryptographic material never leaves the device. Certificates provide stronger client authentication over other schemes, such as user name and password or bearer tokens, because the secret key never leaves the device.

AWS IoT authenticates certificates using the TLS protocol's client authentication mode. TLS is available in many programming languages and operating systems and is commonly used for encrypting data. In TLS client authentication, AWS IoT requests a client X.509 certificate and validates the certificate's status and AWS account against a registry of certificates. It then challenges the client for proof of ownership of the private key that corresponds to the public key contained in the certificate.

To use AWS IoT certificates, clients must support all of the following in their TLS implementation:

- TLS 1.2.
- SHA-256 RSA certificate signature validation.
- One of the cipher suites from the TLS cipher suite support section.

## X.509 Certificates and AWS IoT

### Client Authentication

AWS IoT can use AWS IoT-generated certificates or certificates signed by a CA certificate for device authentication. Certificates generated by AWS IoT are long-lived (but will expire at

2049-12-31T23:59:59Z, that is at midnight GMT on December 31, 2049.) The expiry date and time for certificates signed by a CA certificate are set when the certificate is created.

**Note**

We recommend that each device be given a unique certificate to enable fine-grained management including certificate revocation. Devices must support rotation and replacement of certificates in order to ensure smooth operation as certificates expire.

To use a certificate that is not created by AWS IoT, you must register a CA certificate. All device certificates must be signed by the CA certificate you register.

You can use the AWS IoT console or CLI to perform the following operations:

- Create and register an AWS IoT certificate.
- Register a CA certificate.
- Register a device certificate.
- Activate or deactivate a device certificate.
- Revoke a device certificate.
- Transfer a device certificate to another AWS account.
- List all CA certificates registered to your AWS account.
- List all device certificates registered to your AWS account.

For more information about the CLI commands to use to perform these operations, see [AWS IoT CLI Reference](#).

For more information about using the AWS IoT console to create certificates, see [Create and Activate a Device Certificate](#).

## Server Authentication

Server certificates allow your devices to verify that they're communicating with AWS IoT and not another server impersonating AWS IoT. Service certificates need to be copied onto your device and referenced when connecting to AWS IoT. For more information, see the [AWS IoT Device SDKs \(p. 615\)](#). AWS IoT server certificates are signed by one of the following CA certificates:

### VeriSign Endpoints (legacy)

- RSA 2048 bit key: [VeriSign Class 3 Public Primary G5 root CA certificate](#)

### Amazon Trust Services Endpoints (preferred)

- RSA 2048 bit key: [Amazon Root CA 1](#)
- RSA 4096 bit key: Amazon Root CA 2 - Reserved for future use.
- ECC 256 bit key: Amazon Root CA 3 - Comming soon.
- ECC 384 bit key: Amazon Root CA 4 - Reserved for future use.

We recommend that all customers create an Amazon Trust Services (ATS) endpoint and load these CA certificates onto their devices to avoid any issues with the widespread distrust by browsers of Symantec CAs (including VeriSign) that occurred in October 2018. For backward-compatibility reasons, we still support customers using these endpoints. Customers can create an ATS endpoint by calling the `describe-endpoint` API with the `iot:Data-ATS` endpointType. Devices operating on ATS endpoints are fully interoperable with devices operating on Symantec endpoints in the same account and do not require any reregistration.

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

Storing all of these certificates on your device can take up valuable memory space. If your devices implement RSA-based validation, you can omit the [Amazon Root CA 3](#) and [Amazon Root CA 4](#) ECC certificates. If your devices implement ECC-based certificate validation, you can omit the [Amazon Root CA 1](#) and [Amazon Root CA 2](#) RSA certificates.

All new AWS IoT Core regions, beginning with the May 9, 2018 launch of AWS IoT Core in the Asia Pacific (Mumbai) Region, will serve only ATS certificates.

**Note**

CA certificates have an expiration date after which they cannot be used to validate a server's certificate. CA certificates might have to be replaced before their expiration date. Make sure that you can update the root CA certificates on all of your devices to ensure ongoing connectivity and to keep up-to-date with security best practices.

For a complete list of CA certificates used by AWS IoT see, [Amazon Trust Services](#).

## Create and Register an AWS IoT Device Certificate

You can use the AWS IoT console or the AWS IoT CLI to create an AWS IoT certificate.

### To create a certificate (console)

1. Sign in to the AWS Management Console and open the [AWS IoT console](#).
2. In the left navigation pane, choose **Security** to expand the choices, and then choose **Certificates**. Choose **Create**.
3. Choose **One-click certificate creation - Create certificate**. Alternatively, to generate a certificate with a certificate signing request (CSR), choose **Create with CSR**.
4. Use the links to the public key, private key, and certificate to download each to a secure location.
5. Choose **Activate**.

### To create a certificate (CLI)

The AWS IoT CLI provides two commands to create certificates:

- [create-keys-and-certificate](#)

The [CreateKeysAndCertificate](#) API creates a private key, public key, and X.509 certificate.

- [create-certificate-from-csr](#)

The [CreateCertificateFromCSR](#) API creates a certificate given a CSR.

## Use Your Own Certificate

To use your own X.509 certificates, you must register a CA certificate with AWS IoT. The CA certificate can then be used to sign device certificates. You can register up to 10 CA certificates with the same subject field per AWS account per region. This allows you to have more than one CA sign your device certificates.

**Note**

Device certificates must be signed by the registered CA certificate. It is common for a CA certificate to be used to create an intermediate CA certificate. If you are using an intermediate certificate to sign your device certificates, you must register the intermediate CA certificate. Use the AWS IoT root CA certificate when you connect to AWS IoT even if you register your own root CA certificate. The AWS IoT root CA certificate is used by a device to verify the identity of the AWS IoT servers.

## Contents

- [Registering Your CA Certificate \(p. 183\)](#)
- [Creating a Device Certificate Using Your CA Certificate \(p. 184\)](#)
- [Registering a Device Certificate \(p. 185\)](#)
- [Registering Device Certificates Manually \(p. 185\)](#)
- [Using Automatic/Just-in-Time Registration for Device Certificates \(p. 185\)](#)
- [Deactivate the CA Certificate \(p. 186\)](#)
- [Revoke the Device Certificate \(p. 186\)](#)

If you do not have a CA certificate, you can use [OpenSSL](#) tools to create one.

### To create a CA certificate

1. Generate a key pair.

```
openssl genrsa -out rootCA.key 2048
```

2. Use the private key from the key pair to generate a CA certificate.

```
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem
```

### Registering Your CA Certificate

To register your CA certificate, you must:

- Get a registration code from AWS IoT.
- Sign a private key verification certificate with your CA certificate.
- Pass your CA certificate and a private key verification certificate to the `register-ca-certificate` CLI command.

The `Common Name` field in the private key verification certificate must be set to the registration code generated by the `get-registration-code` CLI command. A single registration code is generated per AWS account. You can use the `register-ca-certificate` command or the AWS IoT console to register CA certificates.

#### Note

A CA certificate cannot be registered to more than one account in the same region. However, a CA certificate can be registered to more than one account if the accounts are in different regions.

### To register a CA certificate

1. Get a registration code from AWS IoT. This code is used as the `Common Name` of the private key verification certificate.

```
aws iot get-registration-code
```

2. Generate a key pair for the private key verification certificate.

```
openssl genrsa -out verificationCert.key 2048
```

3. Create a CSR for the private key verification certificate. Set the Common Name field of the certificate to your registration code.

```
openssl req -new -key verificationCert.key -out verificationCert.csr
```

You are prompted for some information, including the Common Name, for the certificate.

```
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) []:XXXXXXXXXXXXMYREGISTRATIONCODEXXXXXX  
Email Address []:
```

4. Use the CSR to create a private key verification certificate.

```
openssl x509 -req -in verificationCert.csr -CA rootCA.pem -CAkey rootCA.key -  
CAcreateserial -out verificationCert.pem -days 500 -sha256
```

5. Register the CA certificate with AWS IoT. Pass in the CA certificate and the private key verification certificate to the register-ca-certificate CLI command.

```
aws iot register-ca-certificate --ca-certificate file://rootCA.pem --verification-cert  
file://verificationCert.pem
```

6. Use the update-certificate CLI command to activate the CA certificate.

```
aws iot update-ca-certificate --certificate-id XXXXXXXXXXXX --new-status ACTIVE
```

## Creating a Device Certificate Using Your CA Certificate

You can use a CA certificate registered with AWS IoT to create a device certificate. The device certificate must be registered with AWS IoT before use.

### To create a device certificate

1. Generate a key pair.

```
openssl genrsa -out deviceCert.key 2048
```

2. Create a CSR for the device certificate.

```
openssl req -new -key deviceCert.key -out deviceCert.csr
```

You are prompted for some information, as shown here.

```
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 []:
```

3. Create a device certificate from the CSR.

```
openssl x509 -req -in deviceCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out deviceCert.pem -days 500 -sha256
```

**Note**

You must use the CA certificate registered with AWS IoT to create device certificates. If you have more than one CA certificate (with the same subject field and public key) registered in your AWS account, you must specify the CA certificate used to create the device certificate when you register your device certificate.

4. Register a device certificate.

```
aws iot register-certificate --certificate-pem file://deviceCert.pem --ca-certificate-pem file://rootCA.pem
```

5. Use the update-certificate CLI command to activate the device certificate.

```
aws iot update-certificate --certificate-id xxxxxxxxxxxx --new-status ACTIVE
```

## Registering a Device Certificate

You must use the CA certificate registered with AWS IoT to sign device certificates. If you have more than one CA certificate (with the same subject field and public key) registered in your AWS account, you must specify the CA certificate used to sign the device certificate when you register your device certificate. You can register each device certificate manually, or you can use automatic registration, which allows devices to register their certificate when they connect to AWS IoT for the first time.

### Registering Device Certificates Manually

Use the following CLI command to register a device certificate:

```
aws iot register-certificate --certificate-pem file://deviceCert.crt --ca-certificate-pem file://caCert.crt
```

### Using Automatic/Just-in-Time Registration for Device Certificates

To register device certificates automatically when devices first connect to AWS IoT, you must enable automatic registration for your CA certificate. This registers any device certificate signed by your CA certificate when it connects to AWS IoT.

#### Enable Automatic Registration

Use the update-ca-certificate API to set the auto-registration-status of the CA certificate to ENABLE:

```
$ aws iot update-ca-certificate --certificate-id caCertificateId --new-auto-registration-status ENABLE
```

You can also set the auto-registration-status to ENABLE when you use the register-ca-certificate API to register your CA certificate:

```
aws iot register-ca-certificate --ca-certificate file://rootCA.pem --verification-cert file://privateKeyVerificationCert.crt --allow-auto-registration
```

When a device first attempts to connect to AWS IoT, as part of the TLS handshake, it must present a registered CA certificate and a device certificate. AWS IoT recognizes the CA certificate as a

registered CA certificate and automatically registers the device certificate and sets its status to PENDING\_ACTIVATION. This means that the device certificate was automatically registered and is awaiting activation. A certificate must be in the ACTIVE state before it can be used to connect to AWS IoT. When AWS IoT automatically registers a certificate or when a certificate in PENDING\_ACTIVATION status connects, AWS IoT publishes a message to the following MQTT topic:

```
$aws/events/certificates/registered/caCertificateID
```

Where *caCertificateID* is the ID of the CA certificate that issued the device certificate.

The message published to this topic has the following structure:

```
{  
    "certificateId": "certificateID",  
    "caCertificateId": "caCertificateId",  
    "timestamp": timestamp,  
    "certificateStatus": "PENDING_ACTIVATION",  
    "awsAccountId": "awsAccountId",  
    "certificateRegistrationTimestamp": "certificateRegistrationTimestamp"  
}
```

You can create a rule that listens on this topic and performs some actions. We recommend that you create a Lambda rule that verifies the device certificate is not on a certificate revocation list (CRL), activates the certificate, and creates and attaches a policy to the certificate. The policy determines which resources the device is able to access. For more information about how to create a Lambda rule that listens on the \$aws/events/certificates/registered/*caCertificateID* topic and performs these actions, see [Just-in-Time Registration](#).

### Deactivate the CA Certificate

When you register a device certificate, AWS checks if the associated CA certificate is ACTIVE. If the CA certificate is INACTIVE, AWS IoT does not allow the device certificate to be registered. By marking the CA certificate as INACTIVE, you prevent any new device certificates issued by the compromised CA to be registered in your account. You can use the update-ca-certificate API to deactivate the CA certificate:

```
$ aws iot update-ca-certificate --certificate-id certificateId --new-status INACTIVE
```

#### Note

Any registered device certificates that were signed by the compromised CA certificate continue to work until you explicitly revoke them.

Use the ListCertificatesByCA API to get a list of all registered device certificates that were signed by the compromised CA. For each device certificate signed by the compromised CA certificate, use the UpdateCertificate API to revoke the device certificate to prevent it from being used.

### Revoke the Device Certificate

If you detect suspicious activity on a registered device certificate, you can use the update-certificate API to revoke it:

```
$ aws iot update-certificate --certificate-id certificateId  
--new-status REVOKED
```

If any error or exception occurs during the auto-registration of the device certificates, AWS IoT sends events or messages to your logs in CloudWatch Logs. For more information about setting up the logs for your account, see the [Amazon CloudWatch documentation](#).

## IAM Users, Groups, and Roles

IAM users, groups, and roles are the standard mechanisms for managing identity and authentication in AWS. You can use them to connect to AWS IoT HTTP interfaces using the AWS SDK and CLI.

IAM roles also allow AWS IoT to access other AWS resources in your account on your behalf. For example, if you want to have a device publish its state to a DynamoDB table, IAM roles allow AWS IoT to interact with Amazon DynamoDB. For more information, see [IAM Roles](#).

For message broker connections over HTTP, AWS IoT authenticates IAM users, groups, and roles using the Signature Version 4 signing process. For information, see [Signing AWS API Requests](#).

When using AWS Signature Version 4 with AWS IoT, clients must support the following in their TLS implementation:

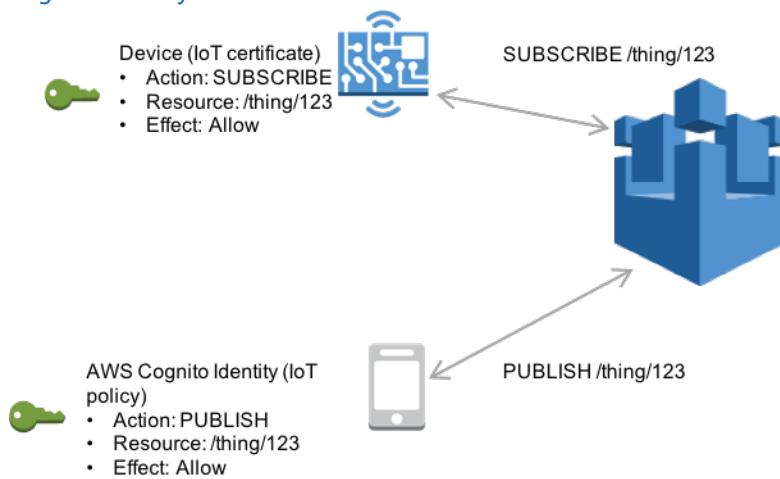
- TLS 1.2, TLS 1.1, TLS 1.0.
- SHA-256 RSA certificate signature validation.
- One of the cipher suites from the TLS cipher suite support section.

For information, see the [IAM User Guide](#).

## Amazon Cognito Identities

Amazon Cognito Identity allows you to use your own identity provider or other popular identity providers, such as Login with Amazon, Facebook, or Google. You exchange a token from your identity provider for AWS security credentials. The credentials represent an IAM role and can be used with AWS IoT.

You can attach an AWS IoT policy to an Amazon Cognito identity using the [AttachPrincipalPolicy](#) API and give fine-grained permissions to an individual user of your AWS IoT application. In this way, you can assign permissions between specific customers and their devices. For more information, see [Amazon Cognito Identity](#).



## Custom Authentication

AWS IoT allows you to define custom authorizers that allow you to manage your own authentication and authorization strategy using a custom authentication service and a Lambda function. Custom authorizers

allow AWS IoT to authenticate your devices and authorize operations using bearer token authentication and authorization strategies.

When an HTTP connection is established (and optionally upgraded to a WebSocket connection) and Signature Version 4 headers are not present, the AWS IoT device gateway checks if a custom authorizer is configured for the endpoint, and if so, it is used to authenticate the connection and authorize the device. Custom authorizers can implement various authentication strategies (for example: JWT verification, OAuth provider callout, and so on) and must return policy documents that are used by the device gateway to authorize MQTT operations.

## Custom Authorizers

Custom authorizers consist of:

Name

A unique arbitrary string that identifies the authorizer.

Lambda function ARN

An ARN of a Lambda function that implements the authentication logic and returns authorization policies.

Public key

The public key from a key pair that is used to prevent unauthorized calls to the authorizer's Lambda function.

Use the following command to generate a key pair:

```
openssl genrsa  
          -out myKeyPair.pem 2048
```

Use the following command to extract the public key from the key pair:

```
openssl rsa -in myKeyPair.pem  
          -pubout > mykey.pub
```

Token key name

The key name used to extract tokens from the WebSocket connection headers.

The logic that performs the authentication is implemented in a Lambda function.

**Note**

AWS Lambda usage is [billed](#). For more information about Lambda, see [AWS Lambda Developer Guide](#).

This function takes a token presented by a device, authenticates the device, and returns the following information:

isAuthenticated

A Boolean value that indicates whether the token was authenticated. If this is `false`, the rest of the response fields should be ignored.

principalId

The principal that is getting this permission.

policyDocuments

A list of policies that specifies which operations the token bearer can perform.

DisconnectAfterInSecs

The length of time, in seconds, to keep the WebSocket connection open.

RefreshAfterInSecs

The length of time, in seconds, after which the Lambda function is invoked to refresh the policies.

Context

Additional information derived after validating the token. This information is made available in [AWS IoT rules engine SQL statements](#) and [IAM/AWS IoT policy variables](#).

You must grant permission to the AWS IoT service principal to invoke the Lambda function that implements the custom authentication/authorization logic. You can do this with the following CLI command:

```
aws lambda add-permission --function-name <lambda_function_name>
--statement-id <unique_identifier_string>
--action 'lambda:InvokeFunction'
--principal iot.amazonaws.com
--source-arn arn:aws:iot:<your-aws-region>:<account_id>:authorizer/<authorizer-name>
```

function-name

The name of the Lambda function to which you are granting invocation permission.

statement-id

A statement identifier

action

The Lambda action you are granting permission to perform.

principal

The principal to which you are granting permission.

source-arn

The ARN of the custom authorizer, specifying this value ensures your Lambda function can only be invoked by the intended custom authorizer.

For more information about granting permission to call Lambda functions, see [AWS Lambda Permissions](#).

You can set a default authorizer that is used when authorizer information is not included in a connection request:

```
aws iot set-default-authorizer --authorizer-name <my-authorizer>
```

## Configure a Custom Authorizer

1. Create a Lambda function that implements your authentication/authorization logic (for example, the `MyAuthorizerFunction` in the following step). The following is an example of what a custom authorizing Lambda function might return:

```
{
    "isAuthenticated": true,
    "principalId": "xxxxxxxx",
    "disconnectAfterInSeconds": 86400,
    "refreshAfterInSeconds": 300,
    "policyDocuments": [
        "{\"Version\":\"2012-10-17\", \"Statement\":[{\"Action\":\"...\", \"Effect\":
        \"Allow/Deny\", \"Resource\":\"...\"}]}"
    ],
    "context": {
        "username": "johnDoe123",
        "city": "Seattle",
        "country": "USA"
    }
}
```

The return value of the Lambda function should be similar to the above. It can be either a JSON serialized or non-serialized object.

2. Register a custom authorizer with AWS IoT using the `create-authorizer` API.

```
aws iot create-authorizer --authorizer-name MyAuthorizer
    --authorizer-function-arn arn:aws:lambda:us-
west-2:<account_id>:function:MyAuthorizerFunction // Lambda ARN
    --token-key-name MyAuthorizerToken // Key use to
    extract token from headers
    --token-signing-public-keys FIRST_KEY= // Public key used
    to verify token signature
    -----BEGIN PUBLIC KEY-----
[...insert your public key here...]
    -----END PUBLIC KEY-----
    --status ACTIVE // Authorizer
    status - must be ACTIVE
    --region us-west-2 // AWS region
    --endpoint https://us-west-2.iot.amazonaws.com // IoT endpoint
```

The `test-invoke-authorizer` API can be used to test if the custom authorizer Lambda function has been configured correctly, as shown:

```
aws iot test-invoke-authorizer --authorizer-name <NAME_OF_AUTHORIZER> --token
<TOKEN_VALUE> --token-signature <TOKEN_SIGNATURE>
```

#### Note

`<TOKEN_SIGNATURE>` must be signed with the private key of the public/private key pair uploaded to AWS IoT used in the `create-authorizer` call. One method of locally creating `<TOKEN_SIGNATURE>` from a UNIX-like command line is as follows:

```
echo -n <TOKEN_VALUE> | openssl dgst -sha256 -sign <PRIVATE_KEY> | openssl
base64
```

You must trim all newline characters from the result of the prior command before passing the `<TOKEN_SIGNATURE>` value to the `test-invoke-authorizer` API.

## Custom Authorizer Workflow

For a device to authenticate with the AWS IoT device gateway using a custom authorizer, it needs both a token and a signature used by AWS to validate the tokens before invoking the authorizer.

When a device attempts to connect to AWS IoT, it sends the following information in HTTP headers:

- A token generated by your authentication service.
- The signature generated by your authentication service.
- The authorizer used to authenticate the token. If omitted, the default authorizer is used.

The following is an example HTTP request to connect to AWS IoT over the WebSocket protocol.

```
GET /mqtt HTTP/1.1
Host: <your-iot-endpoint>
Upgrade: WebSocket
Connection: Upgrade
x-amz-customauthorizer-name: <authorizer-name>
x-amz-customauthorizer-signature: <token-signature>
<token-key-name>: <some-token>
sec-WebSocket-Key: <any random base64 value>
sec-websocket-protocol: mqtt
sec-WebSocket-Version: <websocket version>
```

In this example, the `x-amz-customauthorizer-name` header specifies the custom authorizer to use, the `x-amz-customauthorizer-signature` header contains the digital signature used to verify the token, and the `token-key-name` is the token key name specified by the `--token-key-name` passed to the `create-authorizer` API.

**Note**

Some web browsers may not support custom HTTP headers.

The AWS IoT device gateway validates the digital signature and if valid, calls the specified authorizer. The following is an example payload AWS IoT sends to the custom authenticator's Lambda function.

```
{
  "token": "some-token"
}
```

The authorizer validates the token and returns a principal ID, its associated AWS IoT/IAM policy, and time-to-live (TTL) information for the connection.

The following is an example of the response from a custom authorizer.

```
{
  "isAuthenticated": true,
  "principalId": "xxxxxxxx",
  "disconnectAfterInSeconds": 86400,
  "refreshAfterInSeconds": 300,
  "policyDocuments": [
    {
      "Version": "2012-10-17",
      "Statement": [
        {
          "Action": "...",
          "Effect": "Allow/Deny",
          "Resource": "..."
        }
      ]
    }
  ]
}
```

The return value of the Lambda function should be similar to the above and can be either a JSON serialized or non-serialized object.

The AWS IoT device gateway then establishes the WebSocket connection. AWS IoT caches the policies associated with the principal so subsequent calls can be authorized without having to reauthenticate the device. Any failure that occurs during custom authentication results in authentication failure and connection termination.

For an end-to-end example of this workflow, see [How to Use Your Own Identity and Access Management Systems to Control Access to AWS IoT Resources](#).

# Authorization

Policies determine what an authenticated identity can do. An authenticated identity is used by devices, mobile applications, web applications, and desktop applications. An authenticated identity can even be a user typing AWS IoT CLI commands. The identity can execute AWS IoT operations only if it has a policy that grants it permission.

Both AWS IoT policies and IAM policies are used with AWS IoT to control the operations an identity (also called a *principal*) can perform. The policy type you use depends on the type of identity you are using to authenticate with AWS IoT. The following table shows the identity types, the protocols they use, and the policy types that can be used for authorization.

AWS IoT operations are divided into two groups:

- Control plane API allows you to perform administrative tasks like creating or updating certificates, things, rules, and so on.
- Data plane API allows you send data to and receive data from AWS IoT.

The type of policy you use depends on whether you are using control plane or data plane API.

## AWS IoT Data Plane API and Policy Types

Protocol and Authentication Mechanism	SDK	Identity Type	Policy Type		
MQTT over mutual authentication (port 8883 or 443 <sup>†</sup> <a href="#">(p. 233)</a> )	AWS IoT Device SDK	X.509 certificates	AWS IoT policy		
MQTT over WebSocket (port 443)	AWS Mobile SDK	Amazon Cognito, IAM, or federated identity	AWS IoT policy for Amazon Cognito identities  IAM policy for other identities		
HTTP over server authentication (port 443)	AWS CLI	Amazon Cognito, IAM, or federated identity	AWS IoT policy for Amazon Cognito identities  IAM policy for other identities		
HTTP over mutual authentication (port 8443)	No SDK support	X.509 certificates	AWS IoT policy		

## AWS IoT Control Plane API and Policy Types

Protocol and Authentication Mechanism	SDK	Identity Type	Policy Type		
HTTP over server authentication (port 443)	AWS CLI	Amazon Cognito, IAM, or federated identity	AWS IoT policy for Amazon Cognito identities  IAM policy for other identities		

AWS IoT policies are attached to X.509 certificates or Amazon Cognito identities. IAM policies are attached to an IAM user, group, or role. If you use the AWS IoT console or the AWS IoT CLI to attach the policy (to a certificate or Amazon Cognito Identity), you use an AWS IoT policy. Otherwise, you use an IAM policy.

Policy-based authorization is a powerful tool. It gives you complete control over what a device, user, or application can do in AWS IoT. For example, consider a device connecting to AWS IoT with a certificate. You can allow the device to access all MQTT topics, or you can restrict its access to a single topic. In another example, consider a user typing CLI commands at the command line. By using a policy, you can allow or deny access to any command or AWS IoT resource for the user. You can also control an application's access to AWS IoT resources.

## AWS IoT Policies

AWS IoT policies are JSON documents. They follow the same conventions as IAM policies. AWS IoT supports named policies so many identities can reference the same policy document. Named policies are versioned so they can be easily rolled back.

AWS IoT defines a set of policy actions that describe the operations and resources to which you can grant or deny access. For example:

- `iot:Connect` represents permission to connect to the AWS IoT message broker.
- `iot:Subscribe` represents permission to subscribe to an MQTT topic or topic filter.
- `iot:GetThingShadow` represents permission to get a device's shadow.

AWS IoT policies allow you to control access to the AWS IoT data plane. The AWS IoT data plane consists of operations that allow you to connect to the AWS IoT message broker, send and receive MQTT messages, and get or update a device's shadow. For more information, see [AWS IoT Policy Actions \(p. 194\)](#).

An AWS IoT policy is a JSON document that contains one or more policy statements. Each statement contains an `Effect`, an `Action`, and a `Resource`. The `Effect` specifies whether the action is allowed or denied. The `Action` specifies the action the policy is allowing or denying. The `Resource` specifies the resource or resources on which the action is allowed or denied.

registered devices (1)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT with a client id matching the thing name and restricts the device to publishing on a client-id/thing name specific MQTT topic. In order for a connection to be successful, the thing name must be registered in the AWS IoT Registry and be authenticated using an identity/principal attached to the thing:

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

#### unregistered devices (1)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client id "client1" and restricts the device to publishing on a client-id specific MQTT topic:

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

## AWS IoT Policy Actions

The following policy actions are defined by AWS IoT:

### MQTT Policy Actions

**iot:Connect**

Represents the permission to connect to the AWS IoT message broker. The `iot:Connect` permission is checked every time a `CONNECT` request is sent to the broker. The message broker does not allow two clients with the same client ID to stay connected at the same time. After the second client connects, the broker detects this case and disconnects one of the clients. The `iot:Connect` permission can be used to ensure only authorized clients can connect using a specific client ID.

**iot:Publish**

Represents the permission to publish on an MQTT topic. This permission is checked every time a `PUBLISH` request is sent to the broker. This can be used to allow clients to publish to specific topic patterns.

**Note**

You must also grant `iot:Connect` permission to grant `iot:Publish` permission.

`iot:Receive`

Represents the permission to receive a message from AWS IoT. The `iot:Receive` permission is checked every time a message is delivered to a client. Because this permission is checked on every delivery, it can be used to revoke permissions to clients that are currently subscribed to a topic.

`iot:Subscribe`

Represents the permission to subscribe to a topic filter. This permission is checked every time a `SUBSCRIBE` request is sent to the broker. This can be used to allow clients to subscribe to topics that match specific topic patterns.

**Note**

You must also grant `iot:Connect` permission to grant `iot:Subscribe` permission.

## Shadow Policy Actions

`iot:DeleteThingShadow`

Represents the permission to delete a device's shadow. The `iot:DeleteThingShadow` permission is checked every time a request is made to delete the shadow's contents.

`iot:GetThingShadow`

Represents the permission to retrieve a device's shadow. The `iot:GetThingShadow` permission is checked every time a request is made to retrieve the shadow's contents.

`iot:UpdateThingShadow`

Represents the permission to update a device's shadow. The `iot:UpdateThingShadow` permission is checked every time a request is made to update the shadow's contents.

**Note**

The job execution policy actions apply only for the HTTP TLS endpoint. If you use the MQTT endpoint, you must use MQTT policy actions defined in this topic.

## Job Executions Policy Actions

`iot:DescribeJobExecution`

Represents the permission to retrieve a job execution for a given thing. The `iot:DescribeJobExecution` permission is checked every time a request is made to get the job execution.

`iot:GetPendingJobExecutions`

Represents the permission to retrieve the list of jobs that are not in a terminal status for a thing. The `iot:GetPendingJobExecutions` permission is checked every time a request is made to retrieve the list.

`iot:UpdateJobExecution`

Represents the permission to update a job execution. The `iot:UpdateJobExecution` permission is checked every time a request is made to update the state of a job execution.

`iot:StartNextPendingJobExecution`

Represents the permission to get and start the next pending job execution for a thing. (That is, to update a job execution with status `QUEUED` or `IN_PROGRESS` to `IN_PROGRESS`.) The `iot:StartNextPendingJobExecution` permission is checked every time a request is made to start the next pending job execution.

## Action Resources

To specify a resource for an AWS IoT policy action, you must use the ARN of the resource. All resource ARNs are of the following form:

```
arn:aws:iot:<region>:<AWS account ID>:<resource type>/<resource name>
```

The following table shows the resource to specify for each action type:

Action	Resource
iot:DeleteThingShadow	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:Connect	A client ID ARN - arn:aws:iot:us-east-1:123456789012:client/myClientId
iot:Publish	A topic ARN - arn:aws:iot:us-east-1:123456789012:topic/myTopicName
iot:Subscribe	A topic filter ARN - arn:aws:iot:us-east-1:123456789012:topicfilter/myTopicFilter
iot:Receive	A topic ARN - arn:aws:iot:us-east-1:123456789012:topic/myTopicName
iot:UpdateThingShadow	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:GetThingShadow	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:DescribeJobExecution	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:GetPendingJobExecutions	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:UpdateJobExecution	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne
iot:StartNextPendingJobExecution	A thing ARN - arn:aws:iot:us-east-1:123456789012:thing/thingOne

## AWS IoT Policy Variables

AWS IoT defines policy variables that can be used in AWS IoT policies within the resource or condition block. When a policy is evaluated, the policy variables are replaced by actual values. For example, if a device connected to the AWS IoT message broker with a client ID of "100-234-3456," the `iot:ClientId` policy variable would be replaced in the policy document by "100-234-3456." For more information about policy variables, see [IAM Policy Variables](#) and [Multi-Value Conditions](#).

### Basic Policy Variables

AWS IoT defines the following basic policy variables:

- `iot:ClientId`: The client ID used to connect to the AWS IoT message broker.

- **aws:SourceIp**: The IP address of the client connected to the AWS IoT message broker.

The following AWS IoT policy shows the use of policy variables:

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

In these examples,  `${iot:ClientId}`  is replaced by the ID of the client connected to the AWS IoT message broker when the policy is evaluated. When you use policy variables like  `${iot:ClientId}` , you can inadvertently open access to unintended topics. For example, if you use a policy that uses  `${iot:ClientId}`  to specify a topic filter:

```
{
    "Effect": "Allow",
    "Action": ["iot:Subscribe"],
    "Resource": [
        "arn:aws:iot:us-east-1:123456789012:topicfilter/my/${iot:ClientId}/topic"
    ]
}
```

A client can connect using  `+`  as the client ID. This would allow the user to subscribe to any topic matching the topic filter  `my/+ /topic` . To protect against such security gaps, use the  `iot:Connect`  policy action to control which client IDs are able to connect. For example, this policy allows only clients whose client ID is  `clientid1`  to connect:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Connect"],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/clientid1"
            ]
        }
    ]
}
```

## X.509 Certificate Policy Variables

X.509 certificate policy variables allow you to write AWS IoT policies that grant permissions based on X.509 certificate attributes. The following sections describe how you can use these certificate policy variables.

### Issuer Attributes

The following AWS IoT policy variables allow you to allow or deny permissions based on certificate attributes set by the certificate issuer.

- `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`

### Subject Attributes

The following AWS IoT policy variables allow you to grant or deny permissions based on certificate subject attributes set by the certificate issuer.

- `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`

X.509 certificates allow these attributes to contain one or more values. By default, the policy variables for each multi-value attribute return the first value. For example, the

`Certificate.Subject.Country` attribute might contain a list of country names, but when evaluated in a policy, `iot:Certificate.Subject.Country` is replaced by the first country name. You can request a specific attribute value other than the first by using a zero-based index. For example, `iot:Certificate.Subject.Country.1` is replaced by the second country name in the `Certificate.Subject.Country` attribute. If you specify an index value that does not exist (for example, if you ask for a third value when there are only two values assigned to the attribute), no substitution is made and authorization fails. You can use the `.List` suffix on the policy variable name to specify all values of the attribute.

#### registered devices (2)

For devices registered as things in the AWS IoT Registry, the following policy allows clients with a thing name registered in the AWS IoT Registry to connect, but restricts the right to publish to a thing name specific topic to those clients with certificates whose `Certificate.Subject.Organization` attribute is set to "Example Corp" or "AnyCompany". This restriction is accomplished by using a "Condition" field that specifies a condition which must be met in order to allow the preceding action. In this case the condition is that the `Certificate.Subject.Organization` attribute associated with the certificate must include one of the values listed:

```
{  
    "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)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client ids "client1", "client2", and "client3", but restricts the right to publish to a client-id specific topic to those clients with certificates whose `Certificate.Subject.Organization` attribute is set to "Example Corp" or "AnyCompany". This restriction is accomplished by using a "Condition" field that specifies a condition which must be met in order to allow the preceding action. In this case the condition is that the

`Certificate.Subject.Organization` attribute associated with the certificate must include one of the values listed:

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

### Issuer Alternate Name Attributes

The following AWS IoT policy variables allow you to grant or deny permissions based on issuer alternate name attributes set by the certificate issuer.

- `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`

### Subject Alternate Name Attributes

The following AWS IoT policy variables allow you to grant or deny permissions based on subject alternate name attributes set by the certificate issuer.

- `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`

### Other Attributes

You can use `iot:Certificate.SerialNumber` to allow or deny access to AWS IoT resources based on the serial number of a certificate. The `iot:Certificate.AvailableKeys` policy variable contains the name of all certificate policy variables that contain values.

### X.509 Certificate Policy Variable Limitations

The following limitations apply to X.509 certificate policy variables:

#### Wildcards

If wildcard characters are present in certificate attributes, the policy variable is not replaced by the certificate attribute value, leaving the  `${policy-variable}` text in the policy document. This might cause authorization failure.

#### Array fields

Certificate attributes that contain arrays are limited to five items. Additional items are ignored.

#### String length

All string values are limited to 1024 characters. If a certificate attribute contains a string longer than 1024 characters, the policy variable is not replaced by the certificate attribute value, leaving the  `${policy-variable}` in the policy document. This might cause authorization failure.

## Thing Policy Variables

Thing policy variables allow you to write AWS IoT policies that grant or deny permissions based on thing properties like thing names, thing types, and thing attribute values. The thing name is obtained from the client ID in the MQTT Connect message sent when a thing connects to AWS IoT. The thing policy variables are replaced when a thing connects to AWS IoT over MQTT using TLS mutual authentication or MQTT over the WebSocket protocol using authenticated Amazon Cognito identities. Thing policy variables are also replaced when a certificate or authenticated Amazon Cognito identity is attached to a thing. You can use the [AttachThingPrincipal](#) API to attach certificates and authenticated Amazon Cognito identities to a thing.

The following thing policy variables are available:

- `iot:Connection.Thing.ThingName`
- `iot:Connection.Thing.ThingTypeName`
- `iot:Connection.Thing.Attributes[attributeName]`
- `iot:Connection.Thing.IsAttached`

### `iot:Connection.Thing.ThingName`

This resolves to the name of the thing for which the policy is being evaluated. The thing name is set to the client ID of the MQTT/WebSocket connection. This policy variable is available only when connecting over MQTT or MQTT over the WebSocket protocol.

### `iot:Connection.Thing.ThingTypeName`

This resolves to the thing type associated with the thing for which the policy is being evaluated. The thing name is set to the client ID of the MQTT/WebSocket connection. The thing type name is obtained by a call to the `DescribeThing` API. This policy variable is available only when connecting over MQTT or MQTT over the WebSocket protocol.

### iot:Connection.Thing.Attributes[*attributeName*]

This resolves to the value of the specified attribute associated with the thing for which the policy is being evaluated. A thing can have up to 50 attributes. Each attribute is available as a policy variable: `iot:Connection.Thing.Attributes[attributeName]` where `attributeName` is the name of the attribute. The thing name is set to the client ID of the MQTT/WebSocket connection. This policy variable is only available when connecting over MQTT or MQTT over the WebSocket protocol.

### iot:Connection.Thing.IsAttached

This resolves to `true` if the thing for which the policy is being evaluated has a certificate or Amazon Cognito identity attached.

## Example Policies

AWS IoT policies are specified in a JSON document. These are the components of an AWS IoT policy:

### *Version*

Must be set to "2012-10-17".

### *Effect*

Must be set to "Allow" or "Deny".

### *Action*

Must be set to "iot:*operation-name*" where *operation-name* is one of the following:

"iot:Connect": Connect to AWS IoT

"iot:Receive": Receive messages from AWS IoT

"iot:Publish": MQTT publish.

"iot:Subscribe": MQTT subscribe.

"iot:UpdateThingShadow": Update a device's shadow.

"iot:GetThingShadow": Retrieve a device's shadow.

"iot>DeleteThingShadow": Delete a device's shadow.

### *Resource*

Must be set to one of the following:

Client - `arn:aws:iot:region:account-id:client/client-id`

Topic ARN - `arn:aws:iot:region:account-id:topic/topic-name`

Topic filter ARN - `arn:aws:iot:region:account-id:topicfilter/topic-filter`

## Connect Policy Examples

The following policy grants permission to connect to AWS IoT with client id "client1":

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

```

        "Effect": "Allow",
        "Action": [
            "iot:Connect"
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:client/client1"
        ]
    }
]
}

```

The following policy denies permission to connect to AWS IoT with client ids "client1" and "client2", while allowing devices to connect using a client id that matches the name of a thing registered in the AWS IoT Registry:

```

{
    "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}"
            ]
        }
    ]
}

```

### registered devices (3)

The following policy grants permission for a device to connect using its thing name as the client ID and to subscribe to the topic filter `my/topic/filter`. The device must be registered with AWS IoT. When connecting to AWS IoT, the device must provide the certificate associated with the IoT thing in the AWS IoT Registry:

```

{
    "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:topic/my/topic/filter"
            ]
        }
    ]
}

```

```
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topicfilter/my/topic/filter"
        ]
    }
}
```

#### unregistered devices (3)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect using client id "client1" and to subscribe to topic filter 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"
            ]
        }
    ]
}
```

## Publish/Subscribe Policy Examples

The policy you use depends on how you are connecting to AWS IoT. You can connect to AWS IoT using an MQTT client, HTTP, or WebSocket. When you connect with an MQTT client, you are authenticating with an X.509 certificate. When you connect over HTTP or the WebSocket protocol, you are authenticating with Signature Version 4 and Amazon Cognito.

### Policies for MQTT Clients

When you specify topic filters in AWS IoT policies for MQTT clients, MQTT wildcard characters "+" and "#" are treated as literal characters. Their use might result in unexpected behavior.

#### registered devices (4)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT with the client id that matches the thing name, and to subscribe to the topic filter some/+/topic only:

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

```

```

        "Action": [
            "iot:Connect"
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "iot:Subscribe"
        ],
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topicfilter/some/+/topic"
        ]
    }
]
}

```

#### unregistered devices (4)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client id "client1" and subscribe to the topic filter some/+/topic only:

```

{
    "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

Within a policy, the MQTT wildcard character '+' is treated as a literal, not a wildcard. Attempts to subscribe to topic filters that match the pattern some/+/topic will fail and cause the client to disconnect.

You can use "\*" as a wildcard in the resource attribute of the policy. For example, if each device in your account must publish on a unique topic reserved for it alone, use the following policy:

#### registered devices (5)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using a client id that matches the thing name and to publish to any topic prefixed by the thing name:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}",
            ]
        }
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Connection.Thing.ThingName}/*"
            ]
        }
    ]
}
```

#### unregistered devices (5)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client id "client1", "client2", or "client3" and to publish to any topic prefixed by the client id:

```
{
    "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}/*"
            ]
        }
    ]
}
```

You can also use the "\*" wildcard at the end of a topic filter. Using wildcard characters could lead to granting unintended privileges, so they should only be used after careful consideration. One situation in which they might be useful is when devices must subscribe to messages with many different topics, for example if a device must subscribe to reports from temperature sensors in multiple locations.

### registered devices (6)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using the device's thing name as the client id, and to subscribe to a topic prefixed by the thing name, followed by "room", followed by any string. (It is expected that these topics will be, for example, "thing1/room1", "thing1/room2"...):

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

### unregistered devices (6)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client ids "client1", "client2", "client3", and to subscribe to a topic prefixed by the client id, followed by "room", followed by any string. (It is expected that these topics will be, for example, "client1/room1", "client1/room2"...):

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

```
    ]  
}
```

### registered devices (7)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using the device's thing name as the client id, and to subscribe to the topics "my/topic" and "my/othersTopic":

```
{  
    "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/othersTopic"  
            ]  
        }  
    ]  
}
```

### unregistered devices (7)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client id "client1", and to subscribe to the topics "my/topic" and "my/othersTopic":

```
{  
    "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/othersTopic"  
            ]  
        }  
    ]  
}
```

```
        ]
    }
}
```

### registered devices (8)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using the device's thing name as the client id and to subscribe to a topic unique to that thing name/client id:

```
{
    "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)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client id "client1", and to publish to a topic unique to that client id:

```
{
    "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)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using the device's thing name as the client id and to publish to any topic prefixed by that thing name/client except for one topic ending with "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)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client ids "client1" and "client2" and to publish to any topic prefixed by the client id used to connect, except for one topic ending with "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)

For devices registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using the device's thing name as the client id and to subscribe to the topic "my/topic":

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/
${iot:Connection.Thing.ThingName}"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
            ]
        },
        {
            "Effect": "Deny",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/${iot:Thing.ThingName}/bar"
            ]
        }
    ]
}

```

unregistered devices (10)

For devices not registered as things in the AWS IoT Registry, the following policy grants permission to connect to AWS IoT using client id "client1" and to subscribe to the topic "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"  
            ]  
        }  
    ]  
}
```

Thing policy variables are also replaced when a certificate or authenticated Amazon Cognito identity is attached to a thing. The following policy grants permission to connect to AWS IoT with client id "client1" and to publish and subscribe to topic "iotmonitor/provisioning/987654321098". It also allows the certificate holder to subscribe to this same 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",  
                "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/my/topic"  
            ]  
        }  
    ]  
}
```

```
        "Resource": [
            "arn:aws:iot:us-east-1:123456789012:topicfilter/iotmonitor/
provisioning/987654321098"
        ]
    }
}
```

## Policies for HTTP and WebSocket Clients

For the following operations, AWS IoT uses AWS IoT policies attached to Amazon Cognito identities (through the [AttachPolicy API](#)) to scope down the permissions attached to the Amazon Cognito identity pool with authenticated identities. That means an Amazon Cognito identity needs permission from the IAM role policy attached to the pool and the AWS IoT policy attached to the Amazon Cognito identity through the AWS IoT [AttachPolicy API](#).

- `iot:Connect`
- `iot:Publish`
- `iot:Subscribe`
- `iot:Receive`
- `iot:GetThingShadow`
- `iot:UpdateThingShadow`
- `iot:DeleteThingShadow`

### Note

For other AWS IoT operations or for unauthenticated identities, AWS IoT does not scope down the permissions attached to the Amazon Cognito identity pool role. For both authenticated and unauthenticated identities, this is the most permissive policy that we recommend attaching to the Amazon Cognito pool role.

## HTTP

To allow unauthenticated Amazon Cognito identities to publish messages over HTTP on a topic specific to the Cognito identity, attach the following policy to the Amazon Cognito identity pool role:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish",
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-
identity.amazonaws.com:sub}"]
        }
    ]
}
```

To allow authenticated users, attach the preceding policy to the Amazon Cognito identity pool role and to the Cognito identity using the AWS IoT [AttachPrincipalPolicy API](#).

### Note

When authorizing Cognito identities, AWS IoT will consider both these policies and grant the least privileges specified. An action is only allowed if both the policies allow the requested action, and if either one of these policies disallow an action, that action will be unauthorized.

## MQTT

To allow unauthenticated Amazon Cognito identities to publish MQTT messages over WebSockets on a topic specific to the Cognito identity in your account, attach the following policy to the Amazon Cognito identity pool role:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Publish",  
            ],  
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-  
identity.amazonaws.com:sub}"]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:Connect",  
            ],  
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/${cognito-  
identity.amazonaws.com:sub}"]  
        }  
    ]  
}
```

To allow authenticated users, attach the preceding policy to the Amazon Cognito identity pool role and to the Cognito identity using the AWS IoT [AttachPrincipalPolicy](#) API.

**Note**

When authorizing Cognito identities, AWS IoT will consider both these policies and grant the least privileges specified. An action is only allowed if both the policies allow the requested action, and if either one of these policies disallow an action, that action will be unauthorized.

## Receive Policy Examples

registered devices (11)

For devices registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with a client id that matches the thing name and to subscribe to and receive messages on one 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)

For devices not registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client id "client1" and to subscribe to and receive messages on one 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"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/my/topic"
            ]
        }
    ]
}
```

## Certificate Policy Examples

#### registered devices (12)

For devices registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with a client id that matches a thing name, and to publish to a topic whose name is equal to the certificateId of the certificate the device used to authenticate itself:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:topic/certificateId"
            ]
        }
    ]
}
```

```

        ],
        "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}"]
    }
]
}

```

#### unregistered devices (12)

For devices not registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client ids "client1", "client2", and "client3" and to publish to a topic whose name is equal to the `certificateId` of the certificate the device used to authenticate itself:

```

{
    "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"
            ]
        }
    ]
}

```

#### registered devices (13)

For devices registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with a client id that matches a thing name, and to publish to a topic whose name is equal to the subject's common name field of the certificate the device used to authenticate itself:

```

{
    "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**

In this example, the certificate's subject common name is used as the topic identifier, with the assumption that the subject common name is unique for each registered certificate. If the certificates are shared across multiple devices, the subject common name will be the same for all the devices sharing this certificate, thereby allowing publish privileges to the same topic from multiple devices (not recommended).

unregistered devices (13)

For devices not registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client ids "client1", "client2", and "client3" and to publish to a topic whose name is equal to the subject's common name field of the certificate the device used to authenticate itself:

```

{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/
${iot:Certificate.Subject.CommonName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        }
    ]
}

```

#### **Note**

In this example, the certificate's subject common name is used as the topic identifier, with the assumption that the subject common name is unique for each registered certificate. If the certificates are shared across multiple devices, the subject common name will be the same for all the devices sharing this certificate, thereby allowing publish privileges to the same topic from multiple devices (not recommended).

registered devices (14)

For devices registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with a client id that matches a thing name, and to publish to a topic whose name is prefixed with

"admin/" when the certificate used to authenticate the device has its `Subject.CommonName.2` field set to "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"
                }
            }
        }
    ]
}
```

#### unregistered devices (14)

For devices not registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client ids "client1", "client2", and "client3" and to publish to a topic whose name is prefixed with "admin/" when the certificate used to authenticate the device has its `Subject.CommonName.2` field set to "Administrator":

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": [
                "arn:aws:iot:us-east-1:123456789012:client/client1",
                "arn:aws:iot:us-east-1:123456789012:client/client2",
                "arn:aws:iot:us-east-1:123456789012:client/client3"
            ]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/admin/*"],
            "Condition": {
                "StringEquals": {
                    "iot:Certificate.Subject.CommonName.2": "Administrator"
                }
            }
        }
    ]
}
```

```
}
```

### registered devices (15)

For devices registered in AWS IoT Registry, the following policy allows a device to use a thing name registered with AWS IoT to publish on a specific topic consisting of "admin/" followed by the ThingName when the certificate used to authenticate the device has any one of its `Subject.CommonName` fields set to "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"
                }
            }
        }
    ]
}
```

### unregistered devices (15)

For devices not registered in AWS IoT Registry, the following policy grants permission to connect to AWS IoT with client ids "client1", "client2", and "client3" and to publish to the topic "admin" when the certificate used to authenticate the device has any one of its `Subject.CommonName` fields set to "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"]
        }
    ]
}
```

```

        ],
        "Resource": [ "arn:aws:iot:us-east-1:123456789012:topic/admin" ],
        "Condition": {
            "ForAnyValue:StringEquals": {
                "iot:Certificate.Subject.CommonName.List": "Administrator"
            }
        }
    ]
}

```

## Thing Policy Examples

The following policy allows a device to connect if the certificate used to authenticate with AWS IoT is attached to the thing for which the policy is being evaluated:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Connect"],
            "Resource": [ "*" ],
            "Condition": {
                "Bool": {
                    "iot:Connection.Thing.IsAttached": ["true"]
                }
            }
        }
    ]
}
```

## IAM IoT Policies

AWS Identity and Access Management defines a policy action for each operation defined by AWS IoT, including control plane and data plane APIs.

## AWS IoT API Permissions

The following table lists the AWS IoT API, the IAM permissions required, and the resource the API manipulates.

API	Required permission (policy actions)	Resources
AcceptCertificateTransfers	AcceptCertificateTransfers	<code>arn:aws:iot:&lt;region&gt;:&lt;account-id&gt;:cert/&lt;cert-id&gt;</code>
		<b>Note</b> The AWS account specified in the ARN must be the account to which the certificate is being transferred.
AddThingToThingGroup	AddThingToThingGroup	<code>arn:aws:iot:&lt;region&gt;:&lt;account-id&gt;:thinggroup/&lt;thing-group-name&gt;</code> <code>arn:aws:iot:&lt;region&gt;:&lt;account-id&gt;:thing/&lt;thing-name&gt;</code>
AssociateTargetsWithJob	AssociateTargetsWithJob	

API	Required permission (policy actions)	Resources
AttachPolicy	iot:AttachPolicy	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>  or  arn:aws:iot: <i>region:account-id:cert/cert-id</i>
AttachPrincipalPolicy	iot:AttachPrincipalPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
AttachThingPrincipal	iot:AttachThingPrincipal	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
CancelCertificateTransfer	iot:CancelCertificateTransfer	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
		<b>Note</b> The AWS account specified in the ARN must be the account to which the certificate is being transferred.
CancelJob	iot:CancelJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
CancelJobExecution	iot:CancelJobExecution	arn:aws:iot: <i>region:account-id:job/job-id</i>  arn:aws:iot: <i>region:account-id:thing/thing-name</i>
ClearDefaultAuthorizer	iot:ClearDefaultAuthorizer	
CreateAuthorizer	iot>CreateAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i>
CreateCertificateFromCsr	iot>CreateCertificateFromCsr	
CreateJob	iot:CreateJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
CreateKeysAndCertificate	iot>CreateKeysAndCertificate	
CreatePolicy	iot:CreatePolicy	*
CreatePolicyVersion	iot:CreatePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
		<b>Note</b> This must be an AWS IoT policy, not an IAM policy.
CreateRoleAlias	iot:CreateRoleAlias	(parameter: roleAlias)  arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
CreateThing	iot:CreateThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
CreateThingGroup	iot:CreateThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>  for group being created and for parent group, if used
CreateThingType	iot:CreateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
CreateTopicRule	iot:CreateTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>

API	Required permission (policy actions)	Resources
DeleteAuthorizer	iot:DeleteAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-name</i>
DeleteCACertificate	iot:DeleteCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
DeleteCertificate	iot:DeleteCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
DeleteJob	iot:DeleteJob	arn:aws:iot: <i>region:account-id:job/job-id</i>
DeleteJobExecution	iot:DeleteJobExecution	arn:aws:iot: <i>region:account-id:job/job-id</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>
DeletePolicy	iot:DeletePolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
DeletePolicyVersion	iot:DeletePolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
DeleteRegistrationCode	iot:DeleteRegistrationCode	
DeleteRoleAlias	iot:DeleteRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
DeleteThing	iot:DeleteThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
DeleteThingGroup	iot:DeleteThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
DeleteThingType	iot:DeleteThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
DeleteTopicRule	iot:DeleteTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
DeleteV2LoggingLevel	iot:DeleteV2LoggingLevel	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
DeprecateThingType	iot:DeprecateThingType	arn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>
DescribeAuthorizer	iot:DescribeAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i>  (parameter: authorizerName) none
DescribeCACertificate	iot:DescribeCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
DescribeCertificate	iot:DescribeCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
DescribeDefaultAuthorizer	iot:DescribeDefaultAuthorizer	
DescribeEndpoint	iot:DescribeEndpoint*	
DescribeEventConfiguration	iot:DescribeEventConfiguration	
DescribeIndex	iot:DescribeIndex	arn:aws:iot: <i>region:account-id:index/index-name</i>
DescribeJob	iot:DescribeJob	arn:aws:iot: <i>region:account-id:job/job-id</i>

API	Required permission (policy actions)	Resources
DescribeJobExecution	iot:DescribeJobExecution	
DescribeRoleAlias	iot:DescribeRoleAliasarn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>	
DescribeThing	iot:DescribeThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
DescribeThingGroup	iot:DescribeThingGrouparn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>	
DescribeThingRegistrationTask	iot:DescribeThingRegistrationTask	
DescribeThingType	iot:DescribeThingTypearn:aws:iot: <i>region:account-id:thingtype/thing-type-name</i>	
DetachPolicy	iot:DetachPolicy	arn:aws:iot: <i>region:account-id:cert/cert-id</i> or arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
DetachPrincipalPolicy	iot:DetachPrincipalPolicyarn:aws:iot: <i>region:account-id:cert/cert-id</i>	
DetachThingPrincipal	iot:DetachThingPrincipalarn:aws:iot: <i>region:account-id:cert/cert-id</i>	
DisableTopicRule	iot:DisableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
EnableTopicRule	iot:EnableTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
GetEffectivePolicies	iot:GetEffectivePolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
GetIndexingConfiguration	iot:GetIndexingConfiguration	
GetJobDocument	iot:GetJobDocument	arn:aws:iot: <i>region:account-id:job/job-id</i>
GetLoggingOptions	iot:GetLoggingOptions	
GetPolicy	iot:GetPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
GetPolicyVersion	iot:GetPolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
GetRegistrationCode	iot:GetRegistrationCode	
GetTopicRule	iot:GetTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
ListAttachedPolicies	iot>ListAttachedPolicies	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> or arn:aws:iot: <i>region:account-id:cert/cert-id</i>
ListAuthorizers	iot>ListAuthorizers	none
ListCACertificates	iot>ListCACertificates*	
ListCertificates	iot>ListCertificates	*

API	Required permission (policy actions)	Resources
ListCertificatesByCA	iot:ListCertificatesByCA	
ListIndices	iot:ListIndices	none
ListJobExecutionsForJob	iot:ListJobExecutionsForJob	
ListJobExecutionsForThing	iot:ListJobExecutionsForThing	
ListJobs	iot:ListJobs	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> if thingGroupName parameter used
ListOutgoingCertificates	iot:ListOutgoingCertificates	
ListPolicies	iot:ListPolicies	*
ListPolicyPrincipals	iot:ListPolicyPrincipals	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
ListPolicyVersions	iot:ListPolicyVersions	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
ListPrincipalPolicies	iot:ListPrincipalPolicies	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
ListPrincipalThings	iot:ListPrincipalThings	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
ListRoleAliases	iot:ListRoleAliases	none
ListTargetsForPolicy	iot:ListTargetsForPolicy	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
ListThingGroups	iot:ListThingGroups	none
ListThingGroupsForThing	iot:ListThingGroupsForThing	iot: <i>region:account-id:thing/thing-name</i>
ListThingPrincipals	iot:ListThingPrincipals	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
ListThingRegistrationTasks	iot:ListThingRegistrationTasks	Task Reports
ListThingRegistrationTasks	iot:ListThingRegistrationTasks	Tasks
ListThingTypes	iot:ListThingTypes	*
ListThings	iot:ListThings	*
ListThingsInThingGroup	iot:ListThingsInThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
ListTopicRules	iot:ListTopicRules	*
ListV2LoggingLevels	iot:ListV2LoggingLevels	none
RegisterCACertificate	iot:RegisterCACertificate	
RegisterCertificate	iot:RegisterCertificate	*
RegisterThing	iot:RegisterThing	none
RejectCertificateTransfer	iot:RejectCertificateTransfer	arn:aws:iot: <i>region:account-id:cert/cert-id</i>

API	Required permission (policy actions)	Resources
RemoveThingFromThingGroup	iot:RemoveThingFromThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i> arn:aws:iot: <i>region:account-id:thing/thing-name</i>
ReplaceTopicRule	iot:ReplaceTopicRule	arn:aws:iot: <i>region:account-id:rule/rule-name</i>
SearchIndex	iot:SearchIndex	arn:aws:iot: <i>region:account-id:index/index-id</i>
SetDefaultAuthorizer	iot:SetDefaultAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/authorizer-function-name</i>
SetDefaultPolicyVersion	iot:SetDefaultPolicyVersion	arn:aws:iot: <i>region:account-id:policy/policy-name</i>
SetLoggingOptions	iot:SetLoggingOptions	arn:aws:iot: <i>region:account-id:role/role-name</i>
SetV2LoggingLevel	iot:SetV2LoggingLevel	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
SetV2LoggingOptions	iot:SetV2LoggingOptions	arn:aws:iot: <i>region:account-id:role/role-name</i>
StartThingRegistrationTask	iot:StartThingRegistrationTask	
StopThingRegistrationTask	iot:StopThingRegistrationTask	
TestAuthorization	iot:TestAuthorization	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
TestInvokeAuthorizer	iot:TestInvokeAuthorizer	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
TransferCertificate	iot:TransferCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
UpdateAuthorizer	iot:UpdateAuthorizer	arn:aws:iot: <i>region:account-id:authorizer/function/authorizer-function-name</i>
UpdateCACertificate	iot:UpdateCACertificate	arn:aws:iot: <i>region:account-id:cacert/cert-id</i>
UpdateCertificate	iot:UpdateCertificate	arn:aws:iot: <i>region:account-id:cert/cert-id</i>
UpdateEventConfigurations	iot:UpdateEventConfigurations	
UpdateIndexingConfiguration	iot:UpdateIndexingConfiguration	
UpdateRoleAlias	iot:UpdateRoleAlias	arn:aws:iot: <i>region:account-id:rolealias/role-alias-name</i>
UpdateThing	iot:UpdateThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>
UpdateThingGroup	iot:UpdateThingGroup	arn:aws:iot: <i>region:account-id:thinggroup/thing-group-name</i>
UpdateThingGroupsForThing	iot:UpdateThingGroupsForThing	arn:aws:iot: <i>region:account-id:thing/thing-name</i>

## IAM Policy Templates

AWS IoT provides a set of IAM policy templates you can either use as-is or as a starting point for creating custom IAM policies. These templates allow access to configuration and data operations. Configuration

operations allow you to create things, certificates, policies, and rules. Data operations send data over MQTT or HTTP protocols. The following table describes these templates.

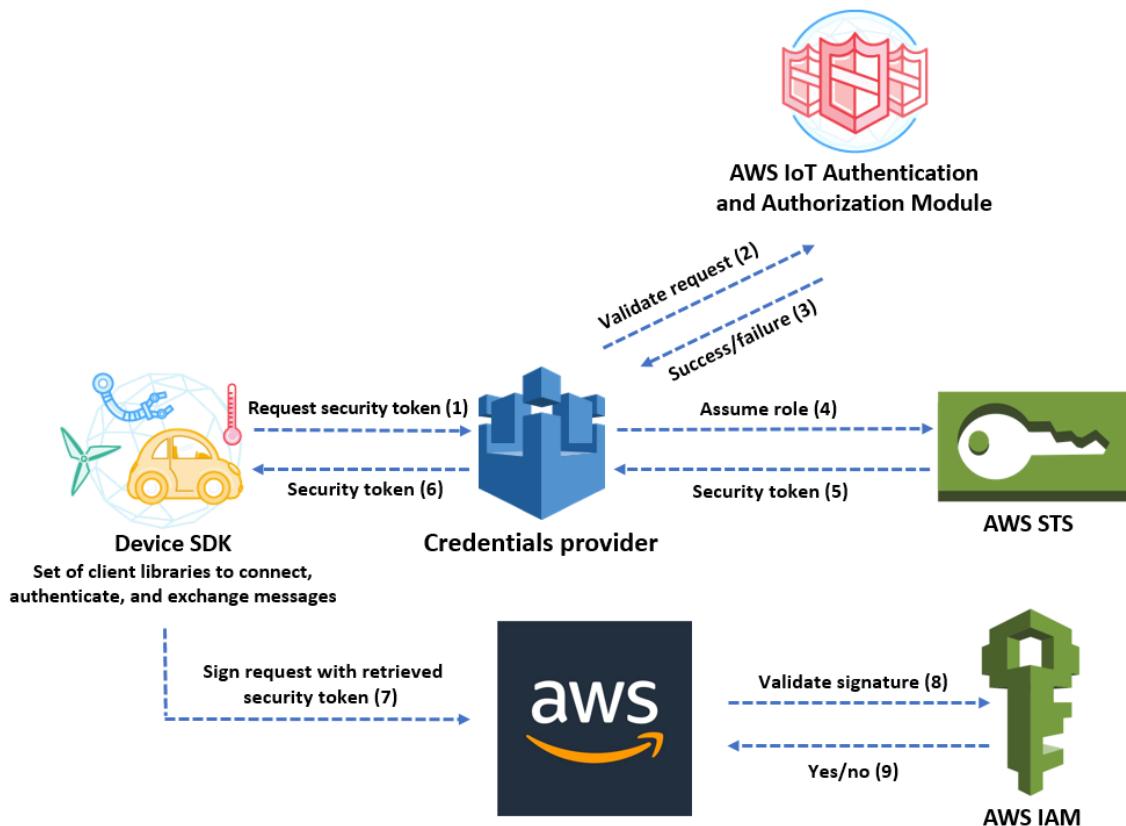
Policy Template	Description
AWSIoTConfigAccess	Allows the associated identity access to all AWS IoT configuration operations. This policy can affect data processing and storage.
AWSIoTConfigReadOnlyAccess	Allows the associated identity to call read-only configuration operations.
AWSIoTDataAccess	Allows the associated identity full access to all AWS IoT data operations. Data operations send data over MQTT or HTTP protocols.
AWSIoTFullAccess	Allows the associated identity full access to all AWS IoT configuration and data operations.
AWSIoTOTUpdate	Allows the associated identity access to create AWS IoT jobs and AWS IoT code signing jobs.
AWSIoTRuleActions	Allows the associated identity access to all AWS services supported in AWS IoT rule actions.
AWSIoTThingsRegistration	Allows the associated identity to register things in bulk using <a href="#">StartThingRegistrationTask</a> . This policy can affect data processing and storage.

## Authorizing Direct Calls to AWS Services

Devices can use X.509 certificates to connect to AWS IoT using TLS mutual authentication protocols. Other AWS services do not support certificate-based authentication, but they can be called using AWS credentials in [AWS Signature Version 4 format](#). The [Signature Version 4 algorithm](#) normally requires the caller to have an access key ID and a secret access key. AWS IoT has a credentials provider that allows you to use the built-in [X.509 certificate](#) as the unique device identity to authenticate AWS requests. This eliminates the need to store an access key ID and a secret access key on your device.

The credentials provider authenticates a caller using an X.509 certificate and issues a temporary, limited-privilege security token. The token can be used to sign and authenticate any AWS request. This way of authenticating your AWS requests requires you to create and configure an [AWS Identity and Access Management \(IAM\) role](#) and attach appropriate IAM policies to the role so that the credentials provider can assume the role on your behalf.

The following diagram illustrates the credentials provider workflow.



1. The AWS IoT device makes an HTTPS request to the credentials provider for a security token. The request includes the device X.509 certificate for authentication.
2. The credentials provider forwards the request to the AWS IoT authentication and authorization module to validate the certificate and verify that it has permission to request the security token.
3. If the certificate is valid and has permission to request a security token, the AWS IoT authentication and authorization module returns success. Otherwise, it sends an exception to the device.
4. After successfully validating the certificate, the credentials provider invokes the [AWS Security Token Service \(AWS STS\)](#) to assume the IAM role that you created for it.
5. AWS STS returns a temporary, limited-privilege security token to the credentials provider.
6. The credentials provider returns the security token to the device.
7. The device uses the security token to sign an AWS request with AWS Signature Version 4.
8. The requested service invokes IAM to validate the signature and authorize the request against access policies attached to the IAM role that you created for the credentials provider.
9. If IAM validates the signature successfully and authorizes the request, the request succeeds. Otherwise, IAM sends an exception.

The following section describes how to use a certificate to get a security token. It assumes that you have already [registered a device](#) and [created and activated your own certificate](#) for it.

## How to Use a Certificate to Get a Security Token

1. Configure the IAM role that the credentials provider assumes on behalf of your device. Attach the following trust policy to the role.

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

For each AWS service that you want to call, attach an access policy to the role. The credentials provider supports the following policy variables:

- `credentials-iot:ThingName`
- `credentials-iot:ThingTypeName`
- `credentials-iot:AwsCertificateId`

When the device provides the thing name in its request to an AWS service, the credentials provider adds `credentials-iot:ThingName` and `credentials-iot:ThingTypeName` as context variables to the security token. The credentials provider provides `credentials-iot:AwsCertificateId` as a context variable even if the device doesn't provide the thing name in the request. You pass the thing name as the value of the `x-amzn-iot-thingname` HTTP request header.

These three variables work for IAM policies only, not AWS IoT policies.

2. Make sure that the user who performs the next step (creating a role alias) has permission to pass this newly created role to AWS IoT. The following policy gives both `iam:GetRole` and `iam:PassRole` permissions to an AWS user. The `iam:GetRole` permission enables the user to get information about the role that you've just created. The `iam:PassRole` permission enables the user to pass the role to another AWS service.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Effect": "Allow",  
         "Action": ["  
             "iam:GetRole",  
             "iam:PassRole"  
         ],  
         "Resource": "arn:aws:iam::your aws account id:role/your role name"  
     }  
]
```

3. Create an AWS IoT role alias. The device that is going to make direct calls to AWS services must know which role ARN to use when connecting to AWS IoT. Hard-coding the role ARN is not a good solution because it requires you to update the device whenever the role ARN changes. A better solution is to use the `CreateRoleAlias` API to create a role alias that points to the role ARN. If the role ARN changes, you simply update the role alias. No change is required on the device. This API takes the following parameters:

`roleAlias`

Mandatory. An arbitrary string identifying the role alias. It serves as the primary key in the role alias data model. It contains 1-128 characters and must include only alphanumeric characters and the =,@, and - symbols. Uppercase and lowercase alphabetic characters are allowed.

`roleArn`

Mandatory. The ARN of the role to which the role alias refers.

credentialDurationInSeconds

Optional. How long (in seconds) the credential is valid. The minimum value is 900 seconds (15 minutes). The maximum value is 3,600 seconds (60 minutes). The default value is 3,600 seconds.

For more information about this API, see [CreateRoleAlias](#).

4. Attach a policy to the device certificate. The policy attached to the device certificate must grant the device permission to assume the role. You do this by granting permission for the `iot:AssumeRoleWithCertificate` action to the role alias, as in the following example.

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "iot:AssumeRoleWithCertificate",  
        "Resource": "arn:aws:iot:your_region:your_aws_account_id:rolealias/your_role  
alias"  
    }  
}
```

5. Make an HTTPS request to the credentials provider to get a security token. Supply the following information.

- *Certificate*: Because this is an HTTP request over TLS mutual authentication, you have to provide the certificate and the corresponding private key to your client while making the request. Use the same certificate and private key that you used when you registered your certificate with AWS IoT.

To make sure your device is communicating with AWS IoT (and not a service impersonating it), refer to the documentation on [Server Authentication in AWS IoT Core](#), follow the links to download the appropriate CA certificates, and then copy them to your device.

- *RoleAlias*: The name of the role alias that you created for the credentials provider.
- *ThingName*: The thing name that you created when you registered your AWS IoT thing. This is passed as the value of the `x-amzn-iot-thingname` HTTP header. This value is required only if you are using thing attributes as policy variables in AWS IoT or IAM policies.

Run the following command in the AWS CLI to obtain the credentials provider endpoint for your AWS account. For more information about this API, see [DescribeEndpoint](#).

```
aws iot describe-endpoint --endpoint-type iot:CredentialProvider
```

The following JSON object is sample output of the `describe-endpoint` command. It contains the `endpointAddress` that you use to request a security token.

```
{  
    "endpointAddress": "your_aws_account_specific_prefix.credentials.iot.your  
region.amazonaws.com"  
}
```

Use the endpoint to make an HTTPS request to the credentials provider to return a security token. The following sample uses `curl` but you can use any HTTP client.

```
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
```

This command returns a security token object that contains an `accessKeyId`, a `secretAccessKey`, a `sessionToken`, and an expiration. The following JSON object is sample output of the `curl` command.

```
{"credentials": {"accessKeyId": "access key", "secretAccessKey": "secret access key", "sessionToken": "session token", "expiration": "2018-01-18T09:18:06Z"}}
```

You can then use the `accessKeyId`, `secretAccessKey`, and `sessionToken` values to sign requests to AWS services. For an end-to-end demonstration of a specific use case, see [How to Eliminate the Need for Hard-Coded AWS Credentials in Devices by Using the AWS IoT Credential Provider](#).

## Cross Account Access

AWS IoT allows you to enable a principal to publish or subscribe to a topic that is defined in an AWS account not owned by the principal. You configure cross account access by creating an IAM policy and IAM role and then attaching the policy to the role.

First, create a customer managed IAM policy as described in [Creating IAM Policies](#), just like you would for other users and certificates in your AWS account.

registered devices (16)

For devices registered in AWS IoT Registry, the following policy grants permission to devices to use thing names registered in your account's (123456789012) AWS IoT Registry to connect to AWS IoT and to publish to a `thingName` specific topic whose name is prefixed with "my/topic/":

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Connect"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:client/${iot:Connection.Thing.ThingName}"]
        },
        {
            "Effect": "Allow",
            "Action": [
                "iot:Publish"
            ],
            "Resource": ["arn:aws:iot:us-east-1:123456789012:topic/my/topic/${iot:Connection.Thing.ThingName}"],
        }
    ]
}
```

unregistered devices (16)

For devices not registered in AWS IoT Registry, the following policy grants permission to a device to use the thing name "client1" registered in your account's (123456789012) AWS IoT Registry to

connect to AWS IoT and to publish to a client id specific topic whose name is prefixed with "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}"]
        }
    ]
}
```

Next, follow the steps in [Creating a Role to Delegate Permissions to an IAM User](#). Enter the account id of the AWS account with which you want to share access. Then, in the final step, attach the policy you just created to the role. If, at a later time, you need to modify the AWS account ID to which you are granting access, you can use the following trust policy format to do so:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": { "AWS": "arn:aws:iam:us-east-1:567890123456:user:MyUser" },
            "Action": "sts:AssumeRole",
        }
    ]
}
```

## Transport Security

The AWS IoT message broker and Device Shadow service encrypt all communication with [TLS version 1.2](#). TLS is used to ensure the confidentiality of the application protocols (MQTT, HTTP) supported by AWS IoT. TLS is available in a number of programming languages and operating systems.

For MQTT, TLS encrypts the connection between the device and the broker. TLS client authentication is used by AWS IoT to identify devices. For HTTP, TLS encrypts the connection between the device and the broker. Authentication is delegated to AWS Signature Version 4.

## TLS Cipher Suite Support

AWS IoT supports the following cipher suites:

- ECDHE-ECDSA-AES128-GCM-SHA256 (recommended)

- ECDHE-RSA-AES128-GCM-SHA256 (recommended)
- 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

# Message Broker for AWS IoT

The AWS IoT message broker is a publish/subscribe broker service that enables the sending and receiving of messages to and from AWS IoT. When communicating with AWS IoT, a client sends a message addressed to a topic like `Sensor/temp/room1`.

**Note**

We do not recommend using personally identifiable information in your topics.

The message broker, in turn, sends the message to all clients that have registered to receive messages for that topic. The act of sending the message is referred to as *publishing*. The act of registering to receive messages for a topic filter is referred to as *subscribing*.

The topic namespace is isolated for each AWS account and region pair. For example, the `Sensor/temp/room1` topic for an AWS account is independent from the `Sensor/temp/room1` topic for another AWS account. This is true of regions, too. The `Sensor/temp/room1` topic in the same AWS account in `us-east-1` is independent from the same topic in `us-east-2`. AWS IoT does not support sending and receiving messages across AWS accounts and regions.

The message broker maintains a list of all client sessions and the subscriptions for each session. When a message is published on a topic, the broker checks for sessions with subscriptions that map to the topic. The broker then forwards the publish message to all sessions that have a currently connected client.

If your use case does not require IoT, see [AWS Messaging](#) for information about other AWS messaging services that better fit your requirements.

## Protocols

The message broker supports the use of the MQTT protocol to publish and subscribe and the HTTPS protocol to publish. Both protocols are supported through IP version 4 and IP version 6. The message broker also supports MQTT over the WebSocket protocol.

## Protocol/Port Mappings

The following table shows each protocol supported by AWS IoT, the authentication method, and port used for each protocol.

### Protocol, Authentication, and Port Mappings

Protocol	Authentication	Port	ALPN ProtocolName
MQTT	X.509 client certificate	8883, 443 <sup>†</sup>	x-amzn-mqtt-ca
HTTPS	X.509 client certificate	8443, 443 <sup>†</sup>	x-amzn-http-ca
HTTPS	SigV4	443	N/A
MQTT over WebSocket	SigV4	443	N/A

<sup>†</sup>Clients that connect on port 443 with X.509 client certificate authentication must implement the [Application Layer Protocol Negotiation \(ALPN\)](#) TLS extension and use the [ALPN ProtocolName](#) listed above in the [ALPN ProtocolNameList](#) sent by the client as part of the `ClientHello` message.

## MQTT

MQTT is a widely adopted, lightweight messaging protocol designed for constrained devices. For more information, see [MQTT](#).

Although the AWS IoT message broker implementation is based on MQTT version 3.1.1, it deviates from the specification as follows:

- In AWS IoT, subscribing to a topic with Quality of Service (QoS) 0 means a message is delivered zero or more times. A message might be delivered more than once. Messages delivered more than once might be sent with a different packet ID. In these cases, the DUP flag is not set.
- AWS IoT does not support publishing and subscribing with QoS 2. The AWS IoT message broker does not send a PUBACK or SUBACK when QoS 2 is requested.
- When responding to a connection request, the message broker sends a CONNACK message. This message contains a flag to indicate if the connection is resuming a previous session. The value of this flag might be incorrect if two MQTT clients connect with the same client ID simultaneously.
- When a client subscribes to a topic, there might be a delay between the time the message broker sends a SUBACK and the time the client starts receiving new matching messages.
- The MQTT specification provides a provision for the publisher to request that the broker retain the last message sent to a topic and send it to all future topic subscribers. AWS IoT does not support retained messages. If a request is made to retain messages, the connection is disconnected.
- The message broker uses the client ID to identify each client. The client ID is passed in from the client to the message broker as part of the MQTT payload. Two clients with the same client ID are not allowed to be connected concurrently to the message broker. When a client connects to the message broker using a client ID that another client is using, a CONNACK message is sent to both clients and the currently connected client is disconnected.
- On rare occasions, the message broker might resend the same logical PUBLISH message with a different packet ID.
- The message broker does not guarantee the order in which messages and ACK are received.

## Topics

The message broker uses topics to route messages from publishing clients to subscribing clients. The forward slash (/) is used to separate topic hierarchy.

### Note

We do not recommend using personally identifiable information in your topics.

The following table lists the wildcards that can be used in the topic filter when you subscribe.

### Topic Wildcards

Wildcard	Description
#	Must be the last character in the topic to which you are subscribing. Works as a wildcard by matching the current tree and all subtrees. For example, a subscription to Sensor/# receives messages published to Sensor/, Sensor/temp, Sensor/temp/room1, but not the messages published to Sensor.
+	Matches exactly one item in the topic hierarchy. For example, a subscription to Sensor/+/room1 receives messages published to Sensor/temp/room1, Sensor/moisture/room1, and so on.

## Reserved Topics

Except for those topics listed here, any topics beginning with \$ are considered reserved and are not supported for publishing and subscribing. Any attempts to publish or subscribe to topics beginning with \$ result in a terminated connection.

Topic	Allowed Operations	Description
\$aws/events/presence/connected/ <i>clientId</i>	Subscribe	AWS IoT publishes to this topic when an MQTT client with the specified client ID connects to AWS IoT. For more information, see <a href="#">Connect/Disconnect Events (p. 612)</a> .
\$aws/events/presence/disconnected/ <i>clientId</i>	Subscribe	AWS IoT publishes to this topic when an MQTT client with the specified client ID disconnects to AWS IoT. For more information, see <a href="#">Connect/Disconnect Events (p. 612)</a> .
\$aws/events/subscriptions/subscribed/ <i>clientId</i>	Subscribe	AWS IoT publishes to this topic when an MQTT client with the specified client ID subscribes to an MQTT topic. For more information, see <a href="#">Subscribe/Unsubscribe Events (p. 613)</a> .
\$aws/events/subscriptions/unsubscribed/ <i>clientId</i>	Subscribe	AWS IoT publishes to this topic when an MQTT client with the specified client ID unsubscribes to an MQTT topic. For more information, see <a href="#">Subscribe/Unsubscribe Events (p. 613)</a> .
\$aws/rules/ <i>ruleName</i>	Publish	A device or an application publishes to this topic to trigger rules directly. For more information, see <a href="#">Basic Ingest (p. 319)</a> .
\$aws/things/ <i>thingName</i> /shadow/delete	Publish/Subscribe	A device or an application publishes to this topic to delete a shadow. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#delete-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#delete-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/delete/accepted	Subscribe	The Device Shadow service sends messages to this

Topic	Allowed Operations	Description
		topic when a shadow is deleted. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#delete-accepted-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#delete-accepted-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/delete/rejected	Subscribe	The Device Shadow service sends messages to this topic when a request to delete a shadow is rejected. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#delete-rejected-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#delete-rejected-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/get	Publish/Subscribe	An application or a thing publishes an empty message to this topic to get a shadow. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html">https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html</a> .
\$aws/things/ <i>thingName</i> /shadow/get/accepted	Subscribe	The Device Shadow service sends messages to this topic when a request for a shadow is made successfully. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#get-accepted-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#get-accepted-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/get/rejected	Subscribe	The Device Shadow service sends messages to this topic when a request for a shadow is rejected. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#get-rejected-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide//device-shadow-mqtt.html#get-rejected-pub-sub-topic</a> .

Topic	Allowed Operations	Description
\$aws/things/ <i>thingName</i> /shadow/update	Publish/Subscribe	A thing or application publishes to this topic to update a shadow. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/update/accepted	Subscribe	The Device Shadow service sends messages to this topic when an update is successfully made to a shadow. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-accepted-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-accepted-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/update/rejected	Subscribe	The Device Shadow service sends messages to this topic when an update to a shadow is rejected. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-rejected-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-rejected-pub-sub-topic</a> .
\$aws/things/ <i>thingName</i> /shadow/update/delta	Subscribe	The Device Shadow service sends messages to this topic when a difference is detected between the reported and desired sections of a shadow. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-delta-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-delta-pub-sub-topic</a> .

Topic	Allowed Operations	Description
\$aws/things/ <i>thingName</i> /shadow/update/documents	Subscribe	AWS IoT publishes a state document to this topic whenever an update to the shadow is successfully performed. For more information, see <a href="https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-documents-pub-sub-topic">https://docs.aws.amazon.com/iot/latest/developerguide/device-shadow-mqtt.html#update-documents-pub-sub-topic</a> .

## MQTT Persistent Sessions

A persistent session represents an ongoing connection to an MQTT message broker. When a client connects to the AWS IoT message broker using a persistent session, the message broker saves all subscriptions the client makes during the connection. When the client disconnects, the message broker stores unacknowledged QoS 1 messages and new QoS 1 messages published to topics to which the client is subscribed. When the client reconnects to the persistent session, all subscriptions are reinstated and all stored messages are sent to the client at a maximum rate of 10 messages per second.

You create an MQTT persistent session by sending a CONNECT message setting the `cleanSession` flag to 0. If no session exists for the client sending the CONNECT message, a new persistent session is created. If a session already exists for the client, it is resumed.

Devices need to look at the `sessionPresent` attribute in the CONNACK (Connection Acknowledged) message to determine if a persistent session is present. If `sessionPresent` is set to 1, a persistent session is present and stored messages are delivered to the client. If `sessionPresent` is set to 0, no persistent session is present and the client must re-subscribe to its topic filters.

Persistent sessions have a default expiry period of 1 hour. The expiry period begins when the message broker detects that a client disconnects (MQTT disconnect or timeout). The persistent session expiry period can be increased through the standard limit increase process. If a client has not resumed its session within the expiry period, the session is terminated and any associated stored messages are discarded. The expiry period is approximate, sessions might be persisted for up to 30 minutes longer (but not less) than the configured duration. For more information, see [AWS Service Limits](#). Any messages stored for persistent sessions will be billed at the standard messaging rate. For more information see, [AWS IoT Pricing](#).

## HTTP

The message broker supports clients connecting with the HTTP protocol using a REST API. Clients can publish by sending a POST message to `<AWS IoT Endpoint>/topics/<url_encoded_topic_name>?qos=1`".

For example, you can use `curl` to emulate sending a message. For example:

```
curl --tlsv1.2 --cacert root-CA.crt --cert 4b7828d2e5-certificate.pem.crt --key 4b7828d2e5-private.pem.key -X POST -d "{\"message\": \"Hello, world\" }" "https://a1pn10j0v8htvw.iot.us-east-1.amazonaws.com:8443/topics/my/topic"
```

--tlsv1.2

Use TLSv1.2 (SSL). `curl` must be installed with OpenSSL and you must use version 1.2 of TLS.

--cacert <filename>

The filename of the CA certificate to verify the peer.

--cert <filename>

The client certificate filename.

--key <filename>

The private key filename.

-X POST

The type of request (in this case, POST).

-d <data>

The HTTP POST data you want to publish.

"https://..."

The URL. In this case, the REST API endpoint for the thing.

To find the endpoint for a thing, in the AWS IoT console, choose **Registry** to expand your choices. Choose **Things**, choose the thing, and then choose **Interact**.) After the endpoint add the port (:8443) followed by the keyword "topics", the topic and, finally, specify the quality of service in a query string (?qos=1).

## MQTT over the WebSocket Protocol

AWS IoT supports MQTT over the [WebSocket](#) protocol to enable browser-based and remote applications to send and receive data from AWS IoT-connected devices using AWS credentials. AWS credentials are specified using [AWS Signature Version 4](#). WebSocket support is available on TCP port 443, which allows messages to pass through most firewalls and web proxies.

A WebSocket connection is initiated on a client by sending an HTTP GET request. The URL you use is of the following form:

wss://<endpoint>.iot.<region>.amazonaws.com/mqtt

wss

Specifies the WebSocket protocol.

endpoint

Your AWS account-specific AWS IoT endpoint. You can use the AWS IoT CLI [describe-endpoint](#) command to find this endpoint.

region

The AWS Region of your AWS account.

mqtt

Specifies you are sending MQTT messages over the WebSocket protocol.

When the server responds, the client sends an upgrade request to indicate to the server it will communicate using the WebSocket protocol. After the server acknowledges the upgrade request, all

communication is performed using the WebSocket protocol. The WebSocket implementation you use acts as a transport protocol. The data you send over the WebSocket protocol are MQTT messages.

## Using the WebSocket Protocol in a Web Application

The WebSocket implementation provided by most web browsers does not allow the modification of HTTP headers, so you must add the Signature Version 4 information to the query string. For more information, see [Adding Signing Information to the Query String](#).

The following JavaScript defines some utility functions used in generating a Signature Version 4 request.

```
/***
 * utilities to do sigv4
 * @class SigV4Utils
 */
function SigV4Utils() {}

SigV4Utils.getSignatureKey = function (key, date, region, service) {
    var kDate = AWS.util.crypto.hmac('AWS4' + key, date, 'buffer');
    var kRegion = AWS.util.crypto.hmac(kDate, region, 'buffer');
    var kService = AWS.util.crypto.hmac(kRegion, service, 'buffer');
    var kCredentials = AWS.util.crypto.hmac(kService, 'aws4_request', 'buffer');
    return kCredentials;
};

SigV4Utils.getSignedUrl = function(host, region, credentials) {
    var datetime = AWS.util.date.iso8601(new Date()).replace(/[:\-\]\|\.\d{3}]/g, '');
    var date = datetime.substr(0, 8);

    var method = 'GET';
    var protocol = 'wss';
    var uri = '/mqtt';
    var service = 'iotdevicegateway';
    var algorithm = 'AWS4-HMAC-SHA256';

    var credentialScope = date + '/' + region + '/' + service + '/' + 'aws4_request';
    var canonicalQueryString = 'X-Amz-Algorithm=' + algorithm;
    canonicalQueryString += '&X-Amz-Credential=' +
    encodeURIComponent(credentials.accessKeyId + '/' + credentialScope);
    canonicalQueryString += '&X-Amz-Date=' + datetime;
    canonicalQueryString += '&X-Amz-SignedHeaders=host';

    var canonicalHeaders = 'host:' + host + '\n';
    var payloadHash = AWS.util.crypto.sha256('', 'hex')
    var canonicalRequest = method + '\n' + uri + '\n' + canonicalQueryString + '\n' +
    canonicalHeaders + '\nhost\n' + payloadHash;

    var stringToSign = algorithm + '\n' + datetime + '\n' + credentialScope + '\n' +
    AWS.util.crypto.sha256(canonicalRequest, 'hex');
    var signingKey = SigV4Utils.getSignatureKey(credentials.secretAccessKey, date, region,
    service);
    var signature = AWS.util.crypto.hmac(signingKey, stringToSign, 'hex');

    canonicalQueryString += '&X-Amz-Signature=' + signature;
    if (credentials.sessionToken) {
        canonicalQueryString += '&X-Amz-Security-Token=' +
    encodeURIComponent(credentials.sessionToken);
    }

    var requestUrl = protocol + '://' + host + uri + '?' + canonicalQueryString;
    return requestUrl;
};
```

## To create a Signature Version 4 request

1. Create a canonical request for Signature Version 4.

The following JavaScript code creates a canonical request:

```
var datetime = AWS.util.date.iso8601(new Date()).replace(/[:\ -]|\.\d{3}/g, '');
var date = datetime.substr(0, 8);

var method = 'GET';
var protocol = 'wss';
var uri = '/mqtt';
var service = 'iotdevicegateway';
var algorithm = 'AWS4-HMAC-SHA256';

var credentialScope = date + '/' + region + '/' + service + '/' + 'aws4_request';
var canonicalQueryString = 'X-Amz-Algorithm=' + algorithm;
canonicalQueryString += '&X-Amz-Credential=' +
    encodeURIComponent(credentials.accessKeyId + '/' + credentialScope);
canonicalQueryString += '&X-Amz-Date=' + datetime;
canonicalQueryString += '&X-Amz-SignedHeaders=host';

var canonicalHeaders = 'host:' + host + '\n';
var payloadHash = AWS.util.crypto.sha256('', 'hex')
var canonicalRequest = method + '\n' + uri + '\n' + canonicalQueryString + '\n' +
    canonicalHeaders + '\nhost\n' + payloadHash;
```

2. Create a string to sign, generate a signing key, and sign the string.

Take the canonical URL you created in the previous step and assemble it into a string to sign. You do this by creating a string composed of the hashing algorithm, the date, the credential scope, and the SHA of the canonical request. Next, generate the signing key and sign the string, as shown in the following JavaScript code.

```
var stringToSign = algorithm + '\n' + datetime + '\n' + credentialScope + '\n' +
    AWS.util.crypto.sha256(canonicalRequest, 'hex');
var signingKey = SigV4Utils.getSignatureKey(credentials.secretAccessKey, date, region,
    service);
var signature = AWS.util.crypto.hmac(signingKey, stringToSign, 'hex');
```

3. Add the signing information to the request.

The following JavaScript code shows how to add the signing information to the query string.

```
canonicalQueryString += '&X-Amz-Signature=' + signature;
```

4. If you have session credentials (from an STS server, AssumeRole, or Amazon Cognito), append the session token to the end of the URL string after signing:

```
canonicalQueryString += '&X-Amz-Security-Token=' +
    encodeURIComponent(credentials.sessionToken);
```

5. Prepend the protocol, host, and URI to the canonicalQueryString:

```
var requestUrl = protocol + '://' + host + uri + '?' + canonicalQueryString;
```

6. Open the WebSocket.

The following JavaScript code shows how to create a Paho MQTT client and call CONNECT to AWS IoT. The endpoint argument is your AWS account-specific endpoint. The clientId is a text identifier that is unique among all clients simultaneously connected in your AWS account.

```
var client = new Paho.MQTT.Client(requestUrl, clientId);
var connectOptions = {
    onSuccess: function(){
        // connect succeeded
    },
    useSSL: true,
    timeout: 3,
    mqttVersion: 4,
    onFailure: function() {
        // connect failed
    }
};
client.connect(connectOptions);
```

## Using the WebSocket Protocol in a Mobile Application

We recommend using one of the AWS IoT Device SDKs to connect your device to AWS IoT when making a WebSocket connection. The following AWS IoT Device SDKs support WebSocket-based MQTT connections to AWS IoT:

- [Node.js](#)
- [iOS](#)
- [Android](#)

For a reference implementation for connecting a web application to AWS IoT using MQTT over the WebSocket protocol, see [AWS Labs WebSocket sample](#).

If you are using a programming or scripting language that is not currently supported, any existing WebSocket library can be used as long as the initial WebSocket upgrade request (HTTP POST) is signed using Signature Version 4. Some MQTT clients, such as [Eclipse Paho for JavaScript](#), support the WebSocket protocol natively.

# Rules for AWS IoT

Rules give your devices the ability to interact with AWS services. Rules are analyzed and actions are performed based on the MQTT topic stream. You can use rules to support tasks like these:

- Augment or filter data received from a device.
- Write data received from a device to an Amazon DynamoDB database.
- Save a file to Amazon S3.
- Send a push notification to all users using Amazon SNS.
- Publish data to an Amazon SQS queue.
- Invoke a Lambda function to extract data.
- Process messages from a large number of devices using Amazon Kinesis.
- Send data to the Amazon Elasticsearch Service.
- Capture a CloudWatch metric.
- Change a CloudWatch alarm.
- Send the data from an MQTT message to Amazon Machine Learning to make predictions based on an Amazon ML model.
- Send a message to a Salesforce IoT Input Stream.
- Send message data to an AWS IoT Analytics channel.
- Start execution of a Step Functions state machine.
- Send message data to an AWS IoT Events input.

Your rules can use MQTT messages that pass through the publish/subscribe [Message Broker for AWS IoT \(p. 233\)](#) or, using the [Basic Ingest \(p. 319\)](#) feature, you can securely send device data to the AWS services listed above without incurring [messing costs](#). (The [Basic Ingest \(p. 319\)](#) feature optimizes data flow by removing the publish/subscribe message broker from the ingestion path, so it is more cost effective while keeping the security and data processing features of AWS IoT.)

Before AWS IoT can perform these actions, you must grant it permission to access your AWS resources on your behalf. When the actions are performed, you incur the standard charges for the AWS services you use.

## Granting AWS IoT the Required Access

You use IAM roles to control the AWS resources to which each rule has access. Before you create a rule, you must create an IAM role with a policy that allows access to the required AWS resources. AWS IoT assumes this role when executing a rule.

### To create an IAM role (AWS CLI)

1. Save the following trust policy document, which grants AWS IoT permission to assume the role, to a file called `iot-role-trust.json`:

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

```
        "Service": "iot.amazonaws.com"
    },
    "Action": "sts:AssumeRole"
}
}
```

Use the [create-role](#) command to create an IAM role specifying the `iot-role-trust.json` file:

```
aws iam create-role --role-name my-iot-role --assume-role-policy-document file://iot-role-trust.json
```

The output of this command looks like the following:

```
{
  "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. Save the following JSON into a file named `iot-policy.json`.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "dynamodb:*",
      "Resource": "*"
    }
  ]
}
```

This JSON is an example policy document that grants AWS IoT administrator access to DynamoDB.

Use the [create-policy](#) command to grant AWS IoT access to your AWS resources upon assuming the role, passing in the `iot-policy.json` file:

```
aws iam create-policy --policy-name my-iot-policy --policy-document file://my-iot-policy.json
```

For more information about how to grant access to AWS services in policies for AWS IoT, see [Creating an AWS IoT Rule \(p. 245\)](#).

The output of the [create-policy](#) command contains the ARN of the policy. You need to attach the policy to a role.

```
{
  "Policy": {
    "PolicyName": "my-iot-policy",
    "CreateDate": "2015-09-30T19:31:18.620Z",
    "AttachmentCount": 0,
    "IsAttachable": true,
    "PolicyId": "ZXR6A36LTYANPAI7NJ5UV",
    "DefaultVersionId": "v1",
    "Path": "/",
    "Arn": "arn:aws:iam::123456789012:policy/my-iot-policy",
  }
}
```

```
        "UpdateDate": "2015-09-30T19:31:18.620Z"
    }
}
```

3. Use the [attach-role-policy](#) command to attach your policy to your role:

```
aws iam attach-role-policy --role-name my-iot-role --policy-arn  
"arn:aws:iam::123456789012:policy/my-iot-policy"
```

## Pass Role Permissions

Part of a rule definition is an IAM role that grants permission to access resources specified in the rule's action. The rules engine assumes that role when the rule's action is triggered. The role must be defined in the same AWS account as the rule.

When creating or replacing a rule you are, in effect, passing a role to the rules engine. The user performing this operation requires the `iam:PassRole` permission. To ensure you have this permission, create a policy that grants the `iam:PassRole` permission and attach it to your IAM user. The following policy shows how to allow `iam:PassRole` permission for a role.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1",
      "Effect": "Allow",
      "Action": [
        "iam:PassRole"
      ],
      "Resource": [
        "arn:aws:iam::123456789012:role/myRole"
      ]
    }
  ]
}
```

In this policy example, the `iam:PassRole` permission is granted for the role `myRole`. The role is specified using the role's ARN. You must attach this policy to your IAM user or role to which your user belongs. For more information, see [Working with Managed Policies](#).

**Note**

Lambda functions use resource-based policy, where the policy is attached directly to the Lambda function itself. When creating a rule that invokes a Lambda function, you do not pass a role, so the user creating the rule does not need the `iam:PassRole` permission. For more information about Lambda function authorization, see [Granting Permissions Using a Resource Policy](#).

## Creating an AWS IoT Rule

You configure rules to route data from your connected things. Rules consist of the following:

Rule name

The name of the rule.

**Note**

We do not recommend using personally identifiable information in your rule names.

### Optional description

A textual description of the rule.

#### Note

We do not recommend using personally identifiable information in your rule descriptions.

### SQL statement

A simplified SQL syntax to filter messages received on an MQTT topic and push the data elsewhere. For more information, see [AWS IoT SQL Reference \(p. 267\)](#).

### SQL version

The version of the SQL rules engine to use when evaluating the rule. Although this property is optional, we strongly recommend that you specify the SQL version. If this property is not set, the default, 2015-10-08, is used. For more information, see [SQL Versions \(p. 316\)](#).

### One or more actions

The actions AWS IoT performs when executing the rule. For example, you can insert data into a DynamoDB table, write data to an Amazon S3 bucket, publish to an Amazon SNS topic, or invoke a Lambda function.

### An error action

The action AWS IoT performs when it is unable to perform a rule's action.

When you create a rule, be aware of how much data you are publishing on topics. If you create rules that include a wildcard topic pattern, they might match a large percentage of your messages, and you might need to increase the capacity of the AWS resources used by the target actions. Also, if you create a republish rule that includes a wildcard topic pattern, you can end up with a circular rule that causes an infinite loop.

#### Note

Creating and updating rules are administrator-level actions. Any user who has permission to create or update rules is able to access data processed by the rules.

### To create a rule (AWS CLI)

Use the [create-topic-rule](#) command to create a rule:

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

The following is an example payload file with a rule that inserts all messages sent to the `iot/test` topic into the specified DynamoDB table. The SQL statement filters the messages and the role ARN grants AWS IoT permission to write to the DynamoDB table.

```
{
    "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()}"
        }
    ]
}
```

The following is an example payload file with a rule that inserts all messages sent to the `iot/test` topic into the specified S3 bucket. The SQL statement filters the messages, and the role ARN grants AWS IoT permission to write to the Amazon S3 bucket.

```
{
  "awsIotSqlVersion": "2016-03-23",
  "sql": "SELECT * FROM 'iot/test'",
  "ruleDisabled": false,
  "actions": [
    {
      "s3": {
        "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",
        "bucketName": "my-bucket",
        "key": "myS3Key"
      }
    }
  ]
}
```

The following is an example payload file with a rule that pushes data to Amazon Elasticsearch Service:

```
{
  "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()}"
      }
    }
  ]
}
```

The following is an example payload file with a rule that invokes a Lambda function:

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "lambda": {
        "functionArn": "arn:aws:lambda:us-west-2:123456789012:function:my-lambda-function"
      }
    }
  ]
}
```

The following is an example payload file with a rule that publishes to an Amazon SNS topic:

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "sns": {
        "targetArn": "arn:aws:sns:us-west-2:123456789012:my-sns-topic",
        "message": "Hello from AWS IoT!"
      }
    }
  ]
}
```

```

        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
    }]
}

```

The following is an example payload file with a rule that republishes on a different MQTT topic:

```
{
    "sql": "expression",
    "ruleDisabled": false,
    "awsIotSqlVersion": "2016-03-23",
    "actions": [
        {
            "republish": {
                "topic": "my-mqtt-topic",
                "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
            }
        }
    ]
}
```

The following is an example payload file with a rule that pushes data to an Amazon Kinesis Data Firehose stream:

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

The following is an example payload file with a rule that uses the Amazon Machine Learning `machinelearning_predict` function to republish to a topic if the data in the MQTT payload is classified as a 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"
            }
        }
    ]
}
```

The following is an example payload file with a rule that publishes messages to a Salesforce IoT Cloud input stream.

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

The following is an example payload file with a rule that starts an execution of a Step Functions state machine.

```
{
  "sql": "expression",
  "ruleDisabled": false,
  "awsIotSqlVersion": "2016-03-23",
  "actions": [
    {
      "stepFunctions": {
        "stateMachineName": "myCoolStateMachine",
        "executionNamePrefix": "coolRunning",
        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
    }
  ]
}
```

## Viewing Your Rules

Use the [list-topic-rules](#) command to list your rules:

```
aws iot list-topic-rules
```

Use the [get-topic-rule](#) command to get information about a rule:

```
aws iot get-topic-rule --rule-name my-rule
```

## Deleting a Rule

When you are finished with a rule, you can delete it.

### To delete a rule (AWS CLI)

Use the [delete-topic-rule](#) command to delete a rule:

```
aws iot delete-topic-rule --rule-name my-rule
```

## AWS IoT Rule Actions

AWS IoT rule actions are used to specify what to do when a rule is triggered. You can define actions to write data to a DynamoDB database or a Kinesis stream or to invoke a Lambda function, and more. The following actions are supported:

- `cloudwatchAlarm` to change a CloudWatch alarm.

- `cloudwatchMetric` to capture a CloudWatch metric.
- `dynamoDB` to write data to a DynamoDB database.
- `dynamoDBv2` to write data to a DynamoDB database.
- `elasticsearch` to write data to an Amazon Elasticsearch Service domain.
- `firehose` to write data to an Amazon Kinesis Data Firehose stream.
- `iotAnalytics` to send data to an AWS IoT Analytics channel.
- `iotEvents` to send data to an AWS IoT Events input.
- `kinesis` to write data to a Kinesis stream.
- `lambda` to invoke a Lambda function.
- `republish` to republish the message on another MQTT topic.
- `s3` to write data to an Amazon S3 bucket.
- `salesforce` to write a message to a Salesforce IoT input stream.
- `sns` to write data as a push notification.
- `sqs` to write data to an SQS queue.
- `stepFunctions` to start execution of a Step Functions state machine.

**Note**

The AWS IoT rules engine might make multiple attempts to perform an action in case of intermittent errors. If all attempts fail, the message is discarded and the error is available in your CloudWatch logs. You can specify an error action for each rule that is invoked after a failure occurs. For more information, see [Error Handling \(Error Action\) \(p. 265\)](#).

Each action is described in detail.

## CloudWatch Alarm Action

### CloudWatch Alarm Action

The CloudWatch alarm action allows you to change CloudWatch alarm state. You can specify the state change reason and value in this call.

[more info \(1\)](#)

When creating an AWS IoT rule with a CloudWatch alarm action, you must specify the following information:

`roleArn`

The IAM role that allows access to the CloudWatch alarm.

`alarmName`

The CloudWatch alarm name.

`stateReason`

Reason for the alarm change.

`stateValue`

The value of the alarm state. Acceptable values are `OK`, `ALARM`, `INSUFFICIENT_DATA`.

**Note**

Make sure the role associated with the rule has a policy that grants the `cloudwatch:SetAlarmState` permission.

The following JSON example shows how to define a CloudWatch alarm action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [{  
            "cloudwatchAlarm": {  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw",  
                "alarmName": "IoTAlarm",  
                "stateReason": "Temperature stabilized.",  
                "stateValue": "OK"  
            }  
        }]  
    }  
}
```

For more information, see [CloudWatch Alarms](#).

## CloudWatch Metric Action

### CloudWatch Metric Action

The CloudWatch metric action allows you to capture a CloudWatch metric. You can specify the metric namespace, name, value, unit, and timestamp.

[more info \(2\)](#)

When creating an AWS IoT rule with a CloudWatch metric action, you must specify the following information:

**roleArn**

The IAM role that allows access to the CloudWatch metric.

**metricNamespace**

CloudWatch metric namespace name.

**metricName**

The CloudWatch metric name.

**metricValue**

The CloudWatch metric value.

**metricUnit**

The metric unit supported by CloudWatch.

**metricTimestamp**

An optional Unix timestamp.

#### Note

Make sure that the role associated with the rule has a policy granting the `cloudwatch:PutMetricData` permission.

The following JSON example shows how to define a CloudWatch metric action in an AWS IoT rule:

```
{
```

```
"topicRulePayload": {  
    "sql": "SELECT * FROM 'some/topic'",  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23",  
    "actions": [  
        {"  
            "cloudwatchMetric": {  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_cw",  
                "metricNamespace": "Iotnamespace",  
                "metricName": "IotMetric",  
                "metricValue": "1",  
                "metricUnit": "Count",  
                "metricTimestamp": "1456821314"  
            }  
        }  
    ]  
}
```

For more information, see [CloudWatch Metrics](#).

## DynamoDB Action

### DynamoDB Action

The **dynamoDB** action allows you to write all or part of an MQTT message to a DynamoDB table.

[more info \(3\)](#)

When creating a DynamoDB rule, you must specify the following information:

#### hashKeyType

The data type of the hash key (also called the partition key). Valid values are: "STRING" or "NUMBER".

#### hashKeyField

The name of the hash key (also called the partition key).

#### hashKeyValue

The value of the hash key.

#### rangeKeyType

Optional. The data type of the range key (also called the sort key). Valid values are: "STRING" or "NUMBER".

#### rangeKeyField

Optional. The name of the range key (also called the sort key).

#### rangeKeyValue

Optional. The value of the range key.

#### operation

Optional. The type of operation to be performed. This follows the substitution template, so it can be \${operation}, but the substitution must result in one of the following: **INSERT**, **UPDATE**, or **DELETE**.

#### payloadField

Optional. The name of the field where the payload is written. If this value is omitted, the payload is written to the **payload** field.

table

The name of the DynamoDB table.

roleARN

The IAM role that allows access to the DynamoDB table. At a minimum, the role must allow the `dynamoDB:PutItem` IAM action.

The data written to the DynamoDB table is the result from the SQL statement of the rule. The `hashKeyValue` and `rangeKeyValue` fields are usually composed of expressions (for example, `"${topic()}"` or `"${timestamp()}"`).

**Note**

Non-JSON data is written to DynamoDB as binary data. The DynamoDB console displays the data as Base64-encoded text.

Make sure that the role associated with the rule has a policy granting the `dynamodb:PutItem` permission.

The following JSON example shows how to define a `dynamoDB` action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "ruleDisabled": false,  
        "sql": "SELECT * AS message FROM 'some/topic'",  
        "description": "A test Dynamo DB rule",  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {"  
                "dynamoDB": {  
                    "hashKeyField": "key",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDB",  
                    "tableName": "my_ddb_table",  
                    "hashKeyValue": "${topic()",  
                    "rangeKeyValue": "${timestamp()",  
                    "rangeKeyField": "timestamp"  
                }  
            }]  
    }  
}
```

For more information, see the [Amazon DynamoDB Getting Started Guide](#).

## DynamoDBv2 Action

### DynamoDBv2 Action

The `dynamoDBv2` action allows you to write all or part of an MQTT message to a DynamoDB table. Each attribute in the payload is written to a separate column in the DynamoDB database.

[more info \(4\)](#)

When creating a DynamoDB rule, you must specify the following information:

roleARN

The IAM role that allows access to the DynamoDB table. At a minimum, the role must allow the `dynamoDB:PutItem` IAM action.

tableName

The name of the DynamoDB table.

**Note**

The MQTT message payload must contain a root-level key that matches the table's primary partition key and a root-level key that matches the table's primary sort key, if one is defined.

The data written to the DynamoDB table is the result from the SQL statement of the rule.

**Note**

Make sure that the role associated with the rule has a policy granting the `dynamodb:PutItem` permission.

The following JSON example shows how to define a `dynamoDB` action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * AS message FROM 'some/topic'",  
        "ruleDisabled": false,  
        "description": "A test DynamoDBv2 rule",  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {"  
                "dynamoDBv2": {  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_dynamoDBv2",  
                    "putItem": {  
                        "tableName": "my_ddb_table"  
                    }  
                }  
            }  
        ]  
    }  
}
```

For more information, see the [Amazon DynamoDB Getting Started Guide](#).

## Elasticsearch Action

### Elasticsearch Action

The `elasticsearch` action allows you to write data from MQTT messages to an Amazon Elasticsearch Service domain. Data in Elasticsearch can then be queried and visualized by using tools like Kibana.

[more info \(5\)](#)

When you create an AWS IoT rule with an `elasticsearch` action, you must specify the following information:

`endpoint`

The endpoint of your Amazon Elasticsearch Service domain.

`index`

The Elasticsearch index where you want to store your data.

`type`

The type of document you are storing.

`id`

The unique identifier for each document.

**Note**

Make sure that the role associated with the rule has a policy granting the `es:ESHttpPut` permission.

The following JSON example shows how to define an `elasticsearch` action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "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()}"  
            }  
        }]  
    }  
}
```

For more information, see the [Amazon Elasticsearch Service Developer Guide](#).

## Firehose Action

### Firehose Action

A firehose action sends data from an MQTT message that triggered the rule to a Kinesis Data Firehose stream.

[more info \(6\)](#)

When creating a rule with a firehose action, you must specify the following information:

`deliveryStreamName`

The Kinesis Data Firehose stream to which to write the message data.

`roleArn`

The IAM role that allows access to Kinesis Data Firehose.

`separator`

A character separator that is used to separate records written to the Kinesis Data Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).

**Note**

Make sure that the role associated with the rule has a policy that grants the `firehose:PutRecord` permission.

The following JSON example shows how to create an AWS IoT rule with a firehose action:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {"firehose": {  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_fh",  
                "deliveryStreamName": "my-stream"  
            }}  
        ]  
    }  
}
```

```
        "actions": [{
            "firehose": {
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_firehose",
                "deliveryStreamName": "my_firehose_stream"
            }
        }]
    }
```

For more information, see the [Kinesis Data Firehose Developer Guide](#).

## IoT Analytics Action

### IoT Analytics Action

An `iotAnalytics` action sends data from the MQTT message that triggered the rule to an AWS IoT Analytics channel.

[more info \(7\)](#)

When creating a rule with an `iotAnalytics` action, you must specify the following information:

`channelName`

The name of the AWS IoT Analytics channel to which to write the data.

`roleArn`

The IAM role that allows access to the AWS IoT Analytics channel.

The policy attached to the role you specify should look like this:

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

and have a trust relationship that looks like this:

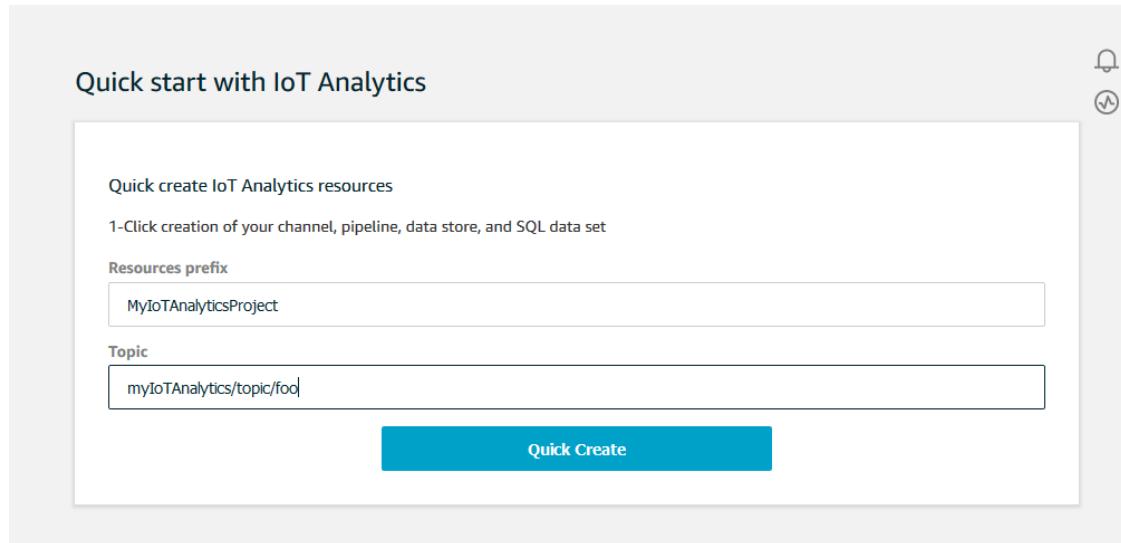
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "Service": "iot.amazonaws.com"
            },
            "Action": "sts:AssumeRole",
        }
    ]
}
```

The following JSON example shows how to create an AWS IoT rule with an `iotAnalytics` action:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [{  
            "iotAnalytics": {  
                "channelName": "mychannel",  
                "roleArn": "arn:aws:iam::123456789012:role/analyticsRole",  
            }  
        }]  
    }  
}
```

For more information, see the [AWS IoT Analytics User Guide](#).

The AWS IoT Analytics console also has a **Quick start** feature that allows you to create a channel, data store, pipeline, and data store with one click. Look for this page when you enter the AWS IoT Analytics console:



## IoT Events Action

### IoT Events Action

An `iotEvents` action sends data from the MQTT message that triggered the rule to an AWS IoT Events input.

[more info \(8\)](#)

When creating a rule with a `iotEvents` action, you must specify the following information:

`inputName`

The name of the AWS IoT Events input.

`messageId`

Optional. Use this to ensure that only one input (message) with a given messageId is processed by an AWS IoT Events detector.

#### roleArn

The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action": "iotevents:BatchPutMessage").

Here is an example trust policy that should be attached to the role:

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "iotevents:BatchPutMessage",  
        "Resource": [ * ]  
    }  
}
```

The following JSON example shows how to create an AWS IoT rule with an `iotEvents` action:

```
{  
    "topicRulePayload": {  
        "sql": "expression",  
        "ruleDisabled": false,  
        "description": "An AWS IoT Events test rule",  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [ {  
            "iotEvents": {  
                "inputName": "MyIoTEventsInput",  
                "messageId": "1234567890",  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_events"  
            },  
            }]  
    }  
}
```

For more information, see the [AWS IoT Events Developer Guide](#).

## Kinesis Action

### Kinesis Action

The `kinesis` action allows you to write data from MQTT messages into a Kinesis stream.  
[more info \(9\)](#)

When creating an AWS IoT rule with a `kinesis` action, you must specify the following information:

#### stream

The Kinesis stream to which to write data.

#### partitionKey

The partition key used to determine to which shard the data is written. The partition key is usually composed of an expression (for example, "\${topic()}" or "\${timestamp()}").

#### Note

Ensure that the policy associated with the rule has the `kinesis:PutRecord` permission.

The following JSON example shows how to define a `kinesis` action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {"  
                "kinesis": {  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_kinesis",  
                    "streamName": "my_kinesis_stream",  
                    "partitionKey": "${topic()}"  
                }  
            }  
        ]  
    }  
}
```

For more information, see the [Kinesis Developer Guide](#).

## Lambda Action

### Lambda Action

A lambda action calls a Lambda function, passing in the MQTT message that triggered the rule.  
[more info \(10\)](#)

In order for AWS IoT to call a Lambda function, you must configure a policy granting the `lambda:InvokeFunction` permission to AWS IoT. Lambda functions use resource-based policies, so you must attach the policy to the Lambda function itself. Use the following CLI command to attach a policy granting `lambda:InvokeFunction` permission:

```
aws lambda add-permission --function-name "function_name" --region "region" --principal iot.amazonaws.com --source-arn arn:aws:iot:us-east-2:account_id:rule/rule_name --source-account "account_id" --statement-id "unique_id" --action "lambda:InvokeFunction"
```

The following are the arguments for the `add-permission` command:

`--function-name`

Name of the Lambda function whose resource policy you are updating by adding a new permission.

`--region`

The AWS Region of your account.

`--principal`

The principal who is getting the permission. This should be `iot.amazonaws.com` to allow AWS IoT permission to call a Lambda function.

`--source-arn`

The ARN of the rule. You can use the `get-topic-rule` CLI command to get the ARN of a rule.

`--source-account`

The AWS account where the rule is defined.

`--statement-id`

A unique statement identifier.

--action

The Lambda action you want to allow in this statement. To allow AWS IoT to invoke a Lambda function, specify `lambda:InvokeFunction`.

**Note**

If you add a permission for an AWS IoT principal without providing the source ARN, any AWS account that creates a rule with your Lambda action can trigger rules to invoke your Lambda function from AWS IoT.

For more information, see [Lambda Permission Model](#).

When creating a rule with a `lambda` action, you must specify the Lambda function to invoke when the rule is triggered.

The following JSON example shows a rule that calls a Lambda function:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [{  
            "lambda": {  
                "functionArn": "arn:aws:lambda:us-  
east-2:123456789012:function:myLambdaFunction"  
            }  
        }]  
    }  
}
```

If you do not specify a version or alias for your Lambda function, the most recent version of the function is executed. You can specify a version or alias if you want to execute a specific version of your Lambda function. To specify a version or alias, append the version or alias to the ARN of the Lambda function. For example:

```
"arn:aws:lambda:us-east-2:123456789012:function:myLambdaFunction:someAlias"
```

For more information about versioning and aliases see [AWS Lambda Function Versioning and Aliases](#). For more information about AWS Lambda, see the [AWS Lambda Developer Guide](#).

## Republish Action

### Republish Action

The `republish` action allows you to republish the message that triggered the role to another MQTT topic.

[more info \(11\)](#)

When creating a rule with a `republish` action, you must specify the following information:

topic

The MQTT topic to which to republish the message. If you are republishing to a reserved topic, one that begins with \$ use \$\$ instead. For example, if you are republishing to a device shadow

topic like `$aws/things/MyThing/shadow/update`, specify the topic as `$$aws/things/MyThing/shadow/update`.

**roleArn**

The IAM role that allows publishing to the MQTT topic.

**Note**

Make sure that the role associated with the rule has a policy granting the `iot:Publish` permission.

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [{  
            "republish": {  
                "topic": "another/topic",  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"  
            }  
        }]  
    }  
}
```

## S3 Action

**S3 Action**

An `s3` action writes the data from the MQTT message that triggered the rule to an Amazon S3 bucket.

[more info \(12\)](#)

When creating an AWS IoT rule with an `s3` action, you must specify the following information, except `cannedacl`, which is optional:

**bucket**

The Amazon S3 bucket to which to write data.

**cannedacl**

Optional. The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, including allowed values, see [S3 Canned ACLs](#).

**key**

The path to the file where the data is written. For example, if the value of this argument is `"${topic()}/${timestamp()}"`, the topic the message was sent to is "this/is/my/topic.". If the current timestamp is 1460685389, the data is written to a file called "1460685389" in the "this/is/my/topic" folder on Amazon S3.

**Note**

Using a static key results in a single file in Amazon S3 being overwritten for each invocation of the rule. More common use cases are to use the message timestamp or another unique message identifier so that a new file is saved in Amazon S3 for each message received.

**roleArn**

The IAM role that allows access to the Amazon S3 bucket.

**Note**

Make sure that the role associated with the rule has a policy granting the `s3:PutObject` permission.

The following JSON example shows how to define an `s3` action in an AWS IoT rule:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "s3": {  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",  
                    "bucketName": "my-bucket",  
                    "key": "${topic()}/${timestamp()}"  
                    "cannedAcl": "public-read"  
                }  
            }  
        ]  
    }  
}
```

For more information, see the [Amazon S3 Developer Guide](#).

## Salesforce Action

### Salesforce Action

A **salesforce** action sends data from the MQTT message that triggered the rule to a Salesforce IoT input stream.

[more info \(13\)](#)

When creating a rule with a **salesforce** action, you must specify the following information:

**url**

The URL exposed by the Salesforce IoT input stream. The URL is available from the Salesforce IoT platform when you create an input stream. For more information, see the [Salesforce IoT documentation](#).

**token**

The token used to authenticate access to the specified Salesforce IoT input stream. The token is available from the Salesforce IoT platform when you create an input stream. For more information, see the [Salesforce IoT documentation](#).

**Note**

These parameters do not support substitution.

The following JSON example shows how to create an AWS IoT rule with a **salesforce** action:

```
{  
    "topicRulePayload": {  
        "sql": "expression",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "salesforce": {  
                    "token": "ABCDEFGHI123456789abcdefghi123456789",  
                    "url": "https://your-salesforce-iot-input-stream-url.com"  
                }  
            }  
        ]  
    }  
}
```

```
        "url": "https://ingestion-cluster-id.my-env.sfdcnow.comstreams/stream-
id/connection-id/my-event"
    }
}
}
```

For more information, refer to the [Salesforce IoT documentation](#).

## SNS Action

### SNS Action

A `sns` action sends the data from the MQTT message that triggered the rule as an SNS push notification.

[more info \(14\)](#)

When creating a rule with an `sns` action, you must specify the following information:

`messageFormat`

The message format. Accepted values are "JSON" and "RAW." The default value of the attribute is "RAW." SNS uses this setting to determine if the payload should be parsed and relevant platform-specific parts of the payload should be extracted.

`roleArn`

The IAM role that allows access to SNS.

`targetArn`

The SNS topic or individual device to which the push notification is sent.

#### Note

Make sure that the policy associated with the rule has the `sns:Publish` permission.

The following JSON example shows how to define an `sns` action in an AWS IoT rule:

```
{
    "topicRulePayload": {
        "sql": "SELECT * FROM 'some/topic'",
        "ruleDisabled": false,
        "awsIotSqlVersion": "2016-03-23",
        "actions": [
            {
                "sns": {
                    "targetArn": "arn:aws:sns:us-east-2:123456789012:my_sns_topic",
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sns"
                }
            }
        ]
    }
}
```

For more information, see the [Amazon SNS Developer Guide](#).

## SQS Action

### SQS Action

An `sqs` action sends data from the MQTT message that triggered the rule to an SQS queue.

[more info \(15\)](#)

When creating a rule with an `sqs` action, you must specify the following information:

`queueUrl`

The URL of the SQS queue to which to write the data.

`useBase64`

Set to `true` if you want the MQTT message data to be Base64-encoded before writing to the SQS queue. Otherwise, set to `false`.

`roleArn`

The IAM role that allows access to the SQS queue.

**Note**

Make sure that the role associated with the rule has a policy granting the `sqs:SendMessage` permission.

The following JSON example shows how to create an AWS IoT rule with an `sqs` action:

```
{  
    "topicRulePayload": {  
        "sql": "SELECT * FROM 'some/topic'",  
        "ruleDisabled": false,  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [  
            {  
                "sqs": {  
                    "queueUrl": "https://sqs.us-east-2.amazonaws.com/123456789012/  
my_sqs_queue",  
                    "roleArn": "arn:aws:iam::123456789012:role/aws_iot_sqs",  
                    "useBase64": false  
                }  
            }  
        ]  
    }  
}
```

SQS action does not support [SQS FIFO Queues](#). Because the Rules Engine is a fully distributed service, there is no guarantee of message order when the SQS action is triggered.

For more information, see the [Amazon SQS Developer Guide](#).

## Step Functions Action

[Step Functions Action](#)

A `stepFunctions` action starts execution of a Step Functions state machine.

[more info \(16\)](#)

When creating a rule with a `stepFunctions` action, you must specify the following information:

`executionNamePrefix`

Optional. The name given to the state machine execution consists of this prefix followed by a UUID. Step Functions creates a unique name for each state machine execution if one is not provided.

stateMachineName

The name of the Step Functions state machine whose execution will be started.

roleArn

The ARN of the role that grants AWS IoT permission to start execution of a state machine ("Action":"states:StartExecution").

Here is an example trust policy that should be attached to the role:

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Allow",  
        "Action": "states:StartExecution",  
        "Resource": [ * ]  
    }  
}
```

The following JSON example shows how to create an AWS IoT rule with a `stepFunctions` action:

```
{  
    "topicRulePayload": {  
        "sql": "expression",  
        "ruleDisabled": false,  
        "description": "A step functions test rule",  
        "awsIotSqlVersion": "2016-03-23",  
        "actions": [{  
            "stepFunctions": {  
                "executionNamePrefix": "myExecution",  
                "stateMachineName": "myStateMachine",  
                "roleArn": "arn:aws:iam::123456789012:role/aws_iot_step_functions"  
            }  
        }]  
    }  
}
```

For more information, see the [Step Functions Developer Guide](#).

## Troubleshooting a Rule

If you are having an issue with your rules, you should enable CloudWatch Logs. By analyzing your logs, you can determine whether the issue is authorization or whether, for example, a WHERE clause condition did not match. For more information, see [Setting Up CloudWatchLogs](#).

## Error Handling (Error Action)

When AWS IoT receives a message from a device, the rules engine checks to see if the message matches a rule. If so, the rule's SQL statement is evaluated and the rule's actions are triggered, passing the SQL statement's result.

If a problem occurs when triggering an action, the rules engine triggers an error action, if one is specified for the rule. This might happen when:

- A rule doesn't have permission to access an Amazon S3 bucket.
- A user error causes DynamoDB provisioned throughput to be exceeded.

## Error Action Message Format

A single message is generated per rule and message. For example, if two rule actions in the same rule fail, the error action receives one message that contains both errors.

The error action message looks like this:

```
{  
    "ruleName": "TestAction",  
    "topic": "testme/action",  
    "cloudwatchTraceId": "7e146a2c-95b5-6caf-98b9-50e3969734c7",  
    "clientId": "iotconsole-1511213971966-0",  
    "base64OriginalPayload": "ewogICJtZXNzYWdIjogIkhlbGxvIHZyb20gQVdTIElvVCBjb25zb2xIgp9",  
    "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

The name of the rule that triggered the error action.

### topic

The topic on which the original message was received.

### cloudwatchTraceld

A unique identity referring to the error logs in CloudWatch.

### clientId

The client ID of the message publisher.

### base64OriginalPayload

The original message payload Base64-encoded.

### failures

#### failedAction

The name of the action that failed to complete (for example, "S3Action").

#### failedResource

The name of the resource (for example, the name of an S3 bucket).

#### errorMessage

The description and explanation of the error.

## Error Action Example

Here is an example of a rule with an added error action. The following rule has an action that writes message data to a DynamoDB table and an error action that writes data to an Amazon S3 bucket:

```
{  
    "sql" : "SELECT * FROM ..."  
    "actions" : [{  
        "dynamoDB" : {  
            "table" : "PoorlyConfiguredTable",  
            "hashKeyField" : "AConstantString",  
            "hashKeyValue" : "AHashKey"}  
    ],  
    "errorAction" : {  
        "s3" : {  
            "roleArn": "arn:aws:iam::123456789012:role/aws_iot_s3",  
            "bucket" : "message-processing-errors",  
            "key" : "${replace(topic(), '/', '-') + '-' + timestamp() + '-' + newuuid()}"  
        }  
    }  
}
```

You can use any function or substitution in an error action's SQL statement, except for external functions (for example, `get_thing_shadow`, `aws_lambda`, and `machinelearning_predict`.)

For more information about rules and how to specify an error action, see [Creating an AWS IoT Rule](#).

For more information about using CloudWatch to monitor the success or failure of rules, see [AWS IoT Metrics and Dimensions \(p. 620\)](#).

## AWS IoT SQL Reference

In AWS IoT, rules are defined using an SQL-like syntax. SQL statements are composed of three types of clauses:

### **SELECT**

Required. Extracts information from the incoming message payload and performs transformations.

### **FROM**

The MQTT message topic filter. The rule is triggered for each message sent to an MQTT topic that matches the filter specified here. Required for rules that are triggered by messages that pass through the message broker. Optional for rules that are only triggered using the [Basic Ingest \(p. 319\)](#) feature.

### **WHERE**

Optional. Adds conditional logic that determines if the actions specified by a rule are carried out.

An example SQL statement looks like this:

```
SELECT color AS rgb FROM 'a/b' WHERE temperature > 50
```

An example MQTT message (also called an incoming payload) looks like this:

```
{  
    "color":"red",  
    "temperature":100  
}
```

If this message is published on the 'a/b' topic, the rule is triggered and the SQL statement is evaluated. The SQL statement extracts the value of the color property if the "temperature" property is greater than 50. The WHERE clause specifies the condition temperature > 50. The AS keyword renames the "color" property to "rgb". The result (also called an *outgoing payload*) looks like this:

```
{
    "rgb": "red"
}
```

This data is then forwarded to the rule's action, which sends the data for more processing. For more information about rule actions, see [AWS IoT Rule Actions \(p. 249\)](#).

## Data Types

The AWS IoT rules engine supports all JSON data types.

### Supported Data Types

Type	Meaning
Int	A discrete Int. 34 digits maximum.
Decimal	A Decimal with a precision of 34 digits, with a minimum non-zero magnitude of 1E-999 and a maximum magnitude 9.999...E999. <b>Note</b> Some functions return Decimals with double precision rather than 34-digit precision.
Boolean	True or False.
String	A UTF-8 string.
Array	A series of values that don't have to have the same type.
Object	A JSON value consisting of a key and a value. Keys must be strings. Values can be any type.
Null	Null as defined by JSON. It's an actual value that represents the absence of a value. You can explicitly create a Null value by using the Null keyword in your SQL statement. For example: "SELECT NULL AS n FROM 'a/b'"
Undefined	Not a value. This isn't explicitly representable in JSON except by omitting the value. For example, in the object { "foo": null}, the key "foo" returns NULL, but the key "bar" returns Undefined. Internally, the SQL language treats Undefined as a value, but it isn't representable in JSON, so when serialized to JSON, the results are Undefined. <div style="border: 1px solid black; padding: 5px; margin-top: 10px;"> <pre>{"foo":null, "bar":undefined}</pre> </div>
	is serialized to JSON as:

Type	Meaning
	<pre>{"foo":null}</pre> <p>Similarly, <code>Undefined</code> is converted to an empty string when serialized by itself. Functions called with invalid arguments (for example, wrong types, wrong number of arguments, and so on) return <code>Undefined</code>.</p>

## Conversions

The following table lists the results when a value of one type is converted to another type (when a value of the incorrect type is given to a function). For example, if the absolute value function "abs" (which expects an `Int` or `Decimal`) is given a `String`, it attempts to convert the `String` to a `Decimal`, following these rules. In this case, `'abs("-5.123")'` is treated as `'abs(-5.123)'`.

### Note

There are no attempted conversions to `Array`, `Object`, `Null`, or `Undefined`.

### To Decimal

Argument Type	Result
<code>Int</code>	A <code>Decimal</code> with no decimal point.
<code>Decimal</code>	The source value.
<code>Boolean</code>	<code>Undefined</code> . (You can explicitly use the <code>cast</code> function to transform <code>true = 1.0</code> , <code>false = 0.0</code> .)
<code>String</code>	The SQL engine tries to parse the string as a <code>Decimal</code> . AWS IoT attempts to parse strings matching the regular expression: <code>^-?\d+(\.\d+)?((?i)E-?\d+)?\$. "0", "-1.2", "5E-12"</code> are all examples of strings that are converted automatically to <code>Decimals</code> .
<code>Array</code>	<code>Undefined</code> .
<code>Object</code>	<code>Undefined</code> .
<code>Null</code>	<code>Null</code> .
<code>Undefined</code>	<code>Undefined</code> .

### To Int

Argument Type	Result
<code>Int</code>	The source value.
<code>Decimal</code>	The source value rounded to the nearest <code>Int</code> .
<code>Boolean</code>	<code>Undefined</code> . (You can explicitly use the <code>cast</code> function to transform <code>true = 1.0</code> , <code>false = 0.0</code> .)

Argument Type	Result
String	The SQL engine tries to parse the string as a Decimal. AWS IoT attempts to parse strings matching the regular expression: <code>^-?\d+(\.\d+)?((?i)E-?\d+)?\$. "0", "-1.2", "5E-12"</code> are all examples of strings that are converted automatically to Decimals. AWS IoT attempts to convert the String to a Decimal, and then truncates the decimal places of that Decimal to make an Int.
Array	Undefined.
Object	Undefined.
Null	Null.
Undefined	Undefined.

## To Boolean

Argument Type	Result
Int	Undefined. (You can explicitly use the cast function to transform 0 = False, any_nonzero_value = True.)
Decimal	Undefined. (You can explicitly use the cast function to transform 0 = False, any_nonzero_value = True.)
Boolean	The original value.
String	"true"=True and "false"=False (case insensitive). Other string values are Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## To String

Argument Type	Result
Int	A string representation of the Int in standard notation.
Decimal	A string representing the Decimal value, possibly in scientific notation.
Boolean	"true" or "false". All lowercase.
String	The original value.

Argument Type	Result
Array	The Array serialized to JSON. The resultant string is a comma-separated list, enclosed in square brackets. A String is quoted. A Decimal, Int, Boolean, and Null is not.
Object	The object serialized to JSON. The resultant string is a comma-separated list of key-value pairs and begins and ends with curly braces. A String is quoted. A Decimal, Int, Boolean, and Null is not.
Null	Undefined.
Undefined	Undefined.

## Operators

The following operators can be used in SELECT, FROM, and WHERE clauses.

### AND operator

Returns a Boolean result. Performs a logical AND operation. Returns true if left and right operands are true. Otherwise, returns false. Boolean operands or case insensitive "true" or "false" string operands are required.

Syntax: `expression AND expression`.

#### AND Operator

Left Operand	Right Operand	Output
Boolean	Boolean	Boolean. True if both operands are true. Otherwise, false.
String/Boolean	String/Boolean	If all strings are "true" or "false" (case insensitive), they are converted to Boolean and processed normally as <code>boolean AND boolean</code> .
Other Value	Other Value	Undefined.

### OR operator

Returns a Boolean result. Performs a logical OR operation. Returns true if either the left or the right operands are true. Otherwise, returns false. Boolean operands or case insensitive "true" or "false" string operands are required.

Syntax: `expression OR expression`.

#### OR Operator

Left Operand	Right Operand	Output
Boolean	Boolean	Boolean. True if either operand is true. Otherwise, false.

Left Operand	Right Operand	Output
String/Boolean	String/Boolean	If all strings are "true" or "false" (case insensitive), they are converted to Booleans and processed normally as <code>boolean</code> OR <code>boolean</code> .
Other Value	Other Value	Undefined.

## NOT operator

Returns a Boolean result. Performs a logical NOT operation. Returns true if the operand is false. Otherwise, returns true. A Boolean operand or case insensitive "true" or "false" string operand is required.

Syntax: NOT `expression`.

### NOT Operator

Operand	Output
Boolean	Boolean. True if operand is false. Otherwise, true.
String	If string is "true" or "false" (case insensitive), it is converted to the corresponding boolean value, and the opposite value is returned.
Other Value	Undefined.

## > operator

Returns a Boolean result. Returns true if the left operand is greater than the right operand. Both operands are converted to a Decimal, and then compared.

Syntax: `expression > expression`.

### > Operator

Left Operand	Right Operand	Output
Int/Decimal	Int/Decimal	Boolean. True if the left operand is greater than the right operand. Otherwise, false.
String/Int/ Decimal	String/Int/ Decimal	If all strings can be converted to Decimal, then Boolean. Returns true if the left operand is greater than the right operand. Otherwise, false.
Other Value	Undefined.	Undefined.

## >= operator

Returns a Boolean result. Returns true if the left operand is greater than or equal to the right operand. Both operands are converted to a Decimal, and then compared.

Syntax: `expression >= expression`.

### >= Operator

Left Operand	Right Operand	Output
Int/Decimal	Int/Decimal	Boolean. True if the left operand is greater than or equal to the right operand. Otherwise, false.
String/Int/ Decimal	String/Int/ Decimal	If all strings can be converted to Decimal, then Boolean. Returns true if the left operand is greater than or equal to the right operand. Otherwise, false.
Other Value	Undefined.	Undefined.

### < operator

Returns a Boolean result. Returns true if the left operand is less than the right operand. Both operands are converted to a Decimal, and then compared.

Syntax: *expression < expression*.

### < Operator

Left Operand	Right Operand	Output
Int/Decimal	Int/Decimal	Boolean. True if the left operand is less than the right operand. Otherwise, false.
String/Int/ Decimal	String/Int/ Decimal	If all strings can be converted to Decimal, then Boolean. Returns true if the left operand is less than the right operand. Otherwise, false.
Other Value	Undefined	Undefined

### <= operator

Returns a Boolean result. Returns true if the left operand is less than or equal to the right operand. Both operands are converted to a Decimal, and then compared.

Syntax: *expression <= expression*.

### >= Operator

Left Operand	Right Operand	Output
Int/Decimal	Int/Decimal	Boolean. True if the left operand is less than or equal to the right operand. Otherwise, false.
String/Int/ Decimal	String/Int/ Decimal	If all strings can be converted to Decimal, then Boolean. Returns true if the left operand is less than or equal to the right operand. Otherwise, false.
Other Value	Undefined	Undefined

### <> operator

Returns a Boolean result. Returns true if both left and right operands are not equal. Otherwise, returns false.

Syntax: `expression <> expression`.

### <> Operator

Left Operand	Right Operand	Output
Int	Int	True if left operand is not equal to right operand. Otherwise, false.
Decimal	Decimal	True if left operand is not equal to right operand. Otherwise, false. Int is converted to Decimal before being compared.
String	String	True if left operand is not equal to right operand. Otherwise, false.
Array	Array	True if the items in each operand are not equal and not in the same order. Otherwise, false
Object	Object	True if the keys and values of each operand are not equal. Otherwise, false. The order of keys/values is unimportant.
Null	Null	False.
Any Value	Undefined	Undefined.
Undefined	Any Value	Undefined.
Mismatched Type	Mismatched Type	True.

## = operator

Returns a Boolean result. Returns true if both left and right operands are equal. Otherwise, returns false.

Syntax: `expression = expression`.

### = Operator

Left Operand	Right Operand	Output
Int	Int	True if left operand is equal to right operand. Otherwise, false.
Decimal	Decimal	True if left operand is equal to right operand. Otherwise, false. Int is converted to Decimal before being compared.
String	String	True if left operand is equal to right operand. Otherwise, false.
Array	Array	True if the items in each operand are equal and in the same order. Otherwise, false.
Object	Object	True if the keys and values of each operand are equal. Otherwise, false. The order of keys/values is unimportant.
Any Value	Undefined	Undefined.
Undefined	Any Value	Undefined.
Mismatched Type	Mismatched Type	False.

## + operator

The "+" is an overloaded operator. It can be used for string concatenation or addition.

*Syntax:* `expression + expression`.

### + Operator

Left Operand	Right Operand	Output
String	Any Value	Converts the right operand to a string and concatenates it to the end of the left operand.
Any Value	String	Converts the left operand to a string and concatenates the right operand to the end of the converted left operand.
Int	Int	Int value. Adds operands together.
Int/Decimal	Int/Decimal	Decimal value. Adds operands together.
Other Value	Other Value	Undefined.

## - operator

Subtracts the right operand from the left operand.

*Syntax:* `expression - expression`.

### - Operator

Left Operand	Right Operand	Output
Int	Int	Int value. Subtracts right operand from left operand.
Int/Decimal	Int/Decimal	Decimal value. Subtracts right operand from left operand.
String/Int/ Decimal	String/Int/ Decimal	If all strings convert to decimals correctly, a Decimal value is returned. Subtracts right operand from left operand. Otherwise, returns Undefined.
Other Value	Other value	Undefined.
Other Value	Other Value	Undefined.

## \* operator

Multiplies the left operand by the right operand.

*Syntax:* `expression * expression`.

### \* Operator

Left Operand	Right Operand	Output
Int	Int	Int value. Multiplies the left operand by the right operand.

Left Operand	Right Operand	Output
Int/Decimal	Int/Decimal	Decimal value. Multiplies the left operand by the right operand.
String/Int/ Decimal	String/Int/ Decimal	If all strings convert to decimals correctly, a Decimal value is returned. Multiplies the left operand by the right operand. Otherwise, returns Undefined.
Other Value	Other value	Undefined.

## / operator

Divides the left operand by the right operand.

Syntax: *expression / expression*.

### / Operator

Left Operand	Right Operand	Output
Int	Int	Int value. Divides the left operand by the right operand.
Int/Decimal	Int/Decimal	Decimal value. Divides the left operand by the right operand.
String/Int/ Decimal	String/Int/ Decimal	If all strings convert to decimals correctly, a Decimal value is returned. Divides the left operand by the right operand. Otherwise, returns Undefined.
Other Value	Other value	Undefined.

## % operator

Returns the remainder from dividing the left operand by the right operand.

Syntax: *expression % expression*.

### % Operator

Left Operand	Right Operand	Output
Int	Int	Int value. Returns the remainder from dividing the left operand by the right operand.
String/Int/ Decimal	String/Int/ Decimal	If all strings convert to decimals correctly, a Decimal value is returned. Returns the remainder from dividing the left operand by the right operand. Otherwise, Undefined.
Other Value	Other value	Undefined.

## Functions

You can use the following built-in functions in the SELECT or WHERE clauses of your SQL expressions.

## abs(Decimal)

Returns the absolute value of a number. Supported by SQL version 2015-10-8 and later.

Example: `abs(-5)` returns 5.

Argument Type	Result
Int	Int, the absolute value of the argument.
Decimal	Decimal, the absolute value of the argument.
Boolean	Undefined.
String	Decimal. The result is the absolute value of the argument. If the string cannot be converted, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## accountid()

Returns the ID of the account that owns this rule as a String. Supported by SQL version 2015-10-8 and later.

Example:

```
accountid() = "123456789012"
```

## acos(Decimal)

Returns the inverse cosine of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `acos(0) = 1.5707963267948966`

Argument Type	Result
Int	Decimal (with double precision), the inverse cosine of the argument. Imaginary results are returned as Undefined.
Decimal	Decimal (with double precision), the inverse cosine of the argument. Imaginary results are returned as Undefined.
Boolean	Undefined.
String	Decimal, the inverse cosine of the argument. If the string cannot be converted, the result is Undefined. Imaginary results are returned as Undefined.

Argument Type	Result
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## asin(Decimal)

Returns the inverse sine of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `asin(0) = 0.0`

Argument Type	Result
Int	Decimal (with double precision), the inverse sine of the argument. Imaginary results are returned as Undefined.
Decimal	Decimal (with double precision), the inverse sine of the argument. Imaginary results are returned as Undefined.
Boolean	Undefined.
String	Decimal (with double precision), the inverse sine of the argument. If the string cannot be converted, the result is Undefined. Imaginary results are returned as Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## atan(Decimal)

Returns the inverse tangent of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `atan(0) = 0.0`

Argument Type	Result
Int	Decimal (with double precision), the inverse tangent of the argument. Imaginary results are returned as Undefined.
Decimal	Decimal (with double precision), the inverse tangent of the argument. Imaginary results are returned as Undefined.

Argument Type	Result
Boolean	Undefined.
String	Decimal, the inverse tangent of the argument. If the string cannot be converted, the result is Undefined. Imaginary results are returned as Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## atan2(Decimal, Decimal)

Returns the angle, in radians, between the positive x-axis and the (x, y) point defined in the two arguments. The angle is positive for counter-clockwise angles (upper half-plane,  $y > 0$ ), and negative for clockwise angles (lower half-plane,  $y < 0$ ). Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `atan2(1, 0) = 1.5707963267948966`

Argument Type	Argument Type	Result
Int/Decimal	Int/Decimal	Decimal (with double precision)
Int/Decimal/String	Int/Decimal/String	Decimal, the inverse tangent of the argument. If the string cannot be converted, the result is Undefined.
Other Value	Other Value	Undefined.

## aws\_lambda(functionArn, inputJson)

Calls the specified Lambda function passing `inputJson` to the Lambda function and returns the JSON generated by the Lambda function.

### Arguments

Argument	Description
<code>functionArn</code>	The ARN of the Lambda function to call. The Lambda function must return JSON data.
<code>inputJson</code>	The JSON input passed to the Lambda function.

You must grant AWS IoT `lambda:InvokeFunction` permissions to invoke the specified Lambda function. The following example shows how to grant the `lambda:InvokeFunction` permission using the 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"
```

The following are the arguments for the **add-permission** command:

**--function-name**

Name of the Lambda function whose resource policy you are updating by adding a new permission.

**--region**

The AWS Region of your account.

**--principal**

The principal who is getting the permission. This should be `iot.amazonaws.com` to allow AWS IoT permission to call a Lambda function.

**--source-arn**

The ARN of the rule. You can use the **get-topic-rule** CLI command to get the ARN of a rule.

**--source-account**

The AWS account where the rule is defined.

**--statement-id**

A unique statement identifier.

**--action**

The Lambda action you want to allow in this statement. To allow AWS IoT to invoke a Lambda function, specify `lambda:InvokeFunction`.

### Note

If you add a permission for an AWS IoT principal without providing the source ARN, any AWS account that creates a rule with your Lambda action can trigger rules to invoke your Lambda function from AWS IoT. For more information, see [Lambda Permission Model](#).

Given a JSON message payload like:

```
{  
    "attribute1": 21,  
    "attribute2": "value"  
}
```

The `aws_lambda` function can be used to call Lambda function as follows:

```
SELECT  
aws_lambda("arn:aws:lambda:us-east-1:account_id:function/lambda_function",  
{"payload":attribute1} as output FROM 'a/b'
```

If you want to pass the full MQTT message payload, you can specify the JSON payload using `*`. For example:

```
SELECT  
aws_lambda("arn:aws:lambda:us-east-1:account_id:function/lambda_function", *) as output  
FROM 'a/b'
```

`payload.inner.element` selects data from message published on topic 'a/b'.

`some.value` selects data from the output that is generated by the Lambda function.

**Note**

The rules engine limits the execution duration of Lambda functions. Lambda function calls from rules should be completed within 2000 milliseconds.

## bitand(Int, Int)

Performs a bitwise AND on the bit representations of the two `Int`(-converted) arguments. Supported by SQL version 2015-10-8 and later.

Example: `bitand(13, 5) = 5`

Argument Type	Argument Type	Result
<code>Int</code>	<code>Int</code>	<code>Int</code> , a bitwise AND of the two arguments.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Int</code> , a bitwise AND of the two arguments. If either argument is a decimal number, it is converted to an integer. If both arguments are rounded integers, the result is an integer. If one argument is a decimal number and the other is not, the result is a decimal number. If the arguments cannot be converted to integers, the result is <code>Undefined</code> .
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Int</code> , a bitwise AND of the two arguments. If either argument is a decimal number or a string, it is converted to an integer. If both arguments are rounded integers, the result is an integer. If one argument is a decimal number and the other is not, the result is a decimal number. If the conversion fails, the result is <code>Undefined</code> .
Other Value	Other Value	<code>Undefined</code> .

## bitor(Int, Int)

Performs a bitwise OR of the bit representations of the two arguments. Supported by SQL version 2015-10-8 and later.

Example: `bitor(8, 5) = 13`

Argument Type	Argument Type	Result
<code>Int</code>	<code>Int</code>	<code>Int</code> , the bitwise OR of the two arguments.
<code>Int/Decimal</code>	<code>Int/Decimal</code>	<code>Int</code> , the bitwise OR of the two arguments. If either argument is a decimal number, it is converted to an integer. If both arguments are rounded integers, the result is an integer. If one argument is a decimal number and the other is not, the result is a decimal number. If the conversion fails, the result is <code>Undefined</code> .
<code>Int/Decimal/String</code>	<code>Int/Decimal/String</code>	<code>Int</code> , the bitwise OR of the two arguments. If either argument is a decimal number or a string, it is converted to an integer. If both arguments are rounded integers, the result is an integer. If one argument is a decimal number and the other is not, the result is a decimal number. If the conversion fails, the result is <code>Undefined</code> .
Other Value	Other Value	<code>Undefined</code> .

## bitxor(Int, Int)

Performs a bitwise XOR on the bit representations of the two `Int`(-converted) arguments. Supported by SQL version 2015-10-8 and later.

Example:`bitor(13, 5) = 8`

Argument Type	Argument Type	Result
Int	Int	Int, a bitwise XOR or
Int/Decimal	Int/Decimal	Int, a bitwise XOR or numbers are rounded
Int/Decimal/String	Int/Decimal/String	Int, a bitwise XOR or converted to decimal Int. If any conversion
Other Value	Other Value	Undefined.

## bitnot(Int)

Performs a bitwise NOT on the bit representations of the `Int`(-converted) argument. Supported by SQL version 2015-10-8 and later.

Example:`bitnot(13) = 2`

Argument Type	Result
Int	Int, a bitwise NOT of the argument.
Decimal	Int, a bitwise NOT of the argument. The <code>Decimal</code> value is rounded down to the nearest <code>Int</code> .
String	Int, a bitwise NOT of the argument. Strings are converted to decimals and rounded down to the nearest <code>Int</code> . If any conversion fails, the result is <code>Undefined</code> .
Other Value	Other value.

## cast()

Converts a value from one data type to another. Cast behaves mostly like the standard conversions, with the addition of the ability to cast numbers to or from Booleans. If AWS IoT cannot determine how to cast one type to another, the result is `Undefined`. Supported by SQL version 2015-10-8 and later. Format: `cast(value as type)`.

Example:

`cast(true as Decimal) = 1.0`

The following keywords might appear after "as" when calling `cast`:

**For SQL version 2015-10-8 and 2016-03-23**

Keyword	Result
Decimal	Casts value to <code>Decimal</code> .
Bool	Casts value to <code>Boolean</code> .

Keyword	Result
Boolean	Casts value to Boolean.
String	Casts value to String.
Nvarchar	Casts value to String.
Text	Casts value to String.
Ntext	Casts value to String.
varchar	Casts value to String.
Int	Casts value to Int.
Integer	Casts value to Int.

#### Additionally, for SQL version 2016-03-23

Keyword	Result
Decimal	Casts value to Decimal.
Bool	Casts value to Boolean.
Boolean	Casts value to Boolean.

Casting rules:

#### Cast to Decimal

Argument Type	Result
Int	A Decimal with no decimal point.
Decimal	The source value.
Boolean	true = 1.0, false = 0.0.
String	Tries to parse the string as a Decimal. AWS IoT attempts to parse strings matching the regex: ^-?\d+(\.\d+)?((?i)E-?\d+)?\$."0", "-1.2", "5E-12" are all examples of strings that are converted automatically to decimals.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

#### Cast to Int

Argument Type	Result
Int	The source value.

Argument Type	Result
Decimal	The source value, rounded down to the nearest Int.
Boolean	true = 1.0, false = 0.0.
String	Tries to parse the string as a Decimal. AWS IoT attempts to parse strings matching the regex: ^-?\d+(\.\d+)?((? <i>i</i> )E-?\d+)?\$". "0", "-1.2", "5E-12" are all examples of strings that are converted automatically to decimals. AWS IoT attempts to convert the string to a Decimal and round down to the nearest Int.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

### Cast to Boolean

Argument Type	Result
Int	0 = False, any_nonzero_value = True.
Decimal	0 = False, any_nonzero_value = True.
Boolean	The source value.
String	"true" = True and "false" = False (case insensitive). Other string values = Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

### Cast to String

Argument Type	Result
Int	A string representation of the Int, in standard notation.
Decimal	A string representing the Decimal value, possibly in scientific notation.
Boolean	"true" or "false", all lowercase.
String	"true"=True and "false"=False (case-insensitive). Other string values = Undefined.
Array	The array serialized to JSON. The result string is a comma-separated list enclosed in square brackets. String is quoted. Decimal, Int, and Boolean are not.

Argument Type	Result
Object	The object serialized to JSON. The JSON string is a comma-separated list of key-value pairs and begins and ends with curly braces. String is quoted. Decimal, Int, Boolean, and Null are not.
Null	Undefined.
Undefined	Undefined.

## ceil(Decimal)

Rounds the given Decimal up to the nearest Int. Supported by SQL version 2015-10-8 and later.

Examples:

```
ceil(1.2) = 2
ceil(11.2) = -1
```

Argument Type	Result
Int	Int, the argument value.
Decimal	Int, the Decimal value rounded up to the nearest Int.
String	Int. The string is converted to Decimal and rounded up to the nearest Int. If the string cannot be converted to a Decimal, the result is Undefined.
Other Value	Undefined.

## chr(String)

Returns the ASCII character that corresponds to the given Int argument. Supported by SQL version 2015-10-8 and later.

Examples:

```
chr(65) = "A".
chr(49) = "1".
```

Argument Type	Result
Int	The character corresponding to the specified ASCII value. If the argument is not a valid ASCII value, the result is Undefined.
Decimal	The character corresponding to the specified ASCII value. The Decimal argument is rounded down to the nearest Int. If the argument is not a valid ASCII value, the result is Undefined.
Boolean	Undefined.

Argument Type	Result
String	If the String can be converted to a Decimal, it is rounded down to the nearest Int. If the argument is not a valid ASCII value, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Other Value	Undefined.

## clientid()

Returns the ID of the MQTT client sending the message, or n/a if the message wasn't sent over MQTT. Supported by SQL version 2015-10-8 and later.

Example:

```
clientid() = "123456789012"
```

## concat()

Concatenates arrays or strings. This function accepts any number of arguments and returns a String or an Array. Supported by SQL version 2015-10-8 and later.

Examples:

```
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]
```

Number of Arguments	Result
0	Undefined.
1	The argument is returned unmodified.
2+	If any argument is an Array, the result is a single array containing all of the arguments. If no arguments are arrays, and at least one argument is a String, the result is the concatenation of the String representations of all the arguments. Arguments are converted to strings using the standard conversions listed above. •

## cos(Decimal)

Returns the cosine of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example:

$\cos(0) = 1.$

Argument Type	Result
Int	Decimal (with double precision), the cosine of the argument. Imaginary results are returned as Undefined.
Decimal	Decimal (with double precision), the cosine of the argument. Imaginary results are returned as Undefined.
Boolean	Undefined.
String	Decimal (with double precision), the cosine of the argument. If the string cannot be converted to a Decimal, the result is Undefined. Imaginary results are returned as Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## cosh(Decimal)

Returns the hyperbolic cosine of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example:  $\cosh(2.3) = 5.037220649268761.$

Argument Type	Result
Int	Decimal (with double precision), the hyperbolic cosine of the argument. Imaginary results are returned as Undefined.
Decimal	Decimal (with double precision), the hyperbolic cosine of the argument. Imaginary results are returned as Undefined.
Boolean	Undefined.
String	Decimal (with double precision), the hyperbolic cosine of the argument. If the string cannot be converted to a Decimal, the result is Undefined. Imaginary results are returned as Undefined.
Array	Undefined.

Argument Type	Result
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## encode(value, encodingScheme)

Use the `encode` function to encode the payload, which potentially might be non-JSON data, into its string representation based on the encoding scheme. Supported by SQL version 2016-03-23 and later.

`value`

Any of the valid expressions, as defined in [AWS IoT SQL Reference \(p. 267\)](#). You can specify \* to encode the entire payload, regardless of whether it's in JSON format. If you supply an expression, the result of the evaluation is converted to a string before it is encoded.

`encodingScheme`

A literal string representing the encoding scheme you want to use. Currently, only 'base64' is supported.

## endswith(String, String)

Returns a Boolean indicating whether the first `String` argument ends with the second `String` argument. If either argument is `Null` or `Undefined`, the result is `Undefined`. Supported by SQL version 2015-10-8 and later.

Example: `endswith("cat", "at") = true.`

argument Type 1	argument Type 2	Result
<code>String</code>	<code>String</code>	True if the first argument ends in the second argument. Otherwise, false.
Other Value	Other Value	Both arguments are converted to strings using standard conversion rules. The result is true if the first argument ends in the second argument. Either argument is <code>Null</code> or <code>Undefined</code> .

## exp(Decimal)

Returns e raised to the `Decimal` argument. `Decimal` arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `exp(1) = e.`

Argument Type	Result
<code>Int</code>	<code>Decimal</code> (with double precision), $e^{\text{argument}}$ .
<code>Decimal</code>	<code>Decimal</code> (with double precision), $e^{\text{argument}}$ .

Argument Type	Result
String	Decimal (with double precision), e ^ argument. If the String cannot be converted to a Decimal, the result is Undefined.
Other Value	Undefined.

## get

Extracts a value from a collection-like type (Array, String, Object). No conversion is applied to the first argument. Conversion applies as documented in the table to the second argument. Supported by SQL version 2015-10-8 and later.

Examples:

```
get(["a", "b", "c"], 1) = "b"
get({"a": "b"}, "a") = "b"
get("abc", 1) = "b"
```

argument Type 1	argument Type 2	Result
Array	Any Type (converted to Int)	The item at the 0-based index specified by the second argument. If the conversion is unsuccessful or the index is outside the bounds of the array (array.length), the result is Undefined.
String	Any Type (converted to Int)	The character at the 0-based index specified by the second argument. If the conversion is unsuccessful or the index is outside the bounds of the string (string.length), the result is Undefined.
Object	String (no conversion is applied)	The value stored in the object corresponding to the key argument.
Other Value	Any Value	Undefined.

## get\_thing\_shadow(thingName, roleARN)

Returns the shadow of the specified thing. Supported by SQL version 2016-03-23 and later.

thingName

String: The name of the thing whose shadow you want to retrieve.

roleArn

String: A role ARN with `iot:GetThingShadow` permission.

Example:

```
SELECT * from 'a/b'  
  
WHERE get_thing_shadow("MyThing", "arn:aws:iam::123456789012:role/  
AllowsThingShadowAccess") .state.reported.alarm = 'ON'
```

## Hashing Functions

AWS IoT provides the following hashing functions:

- md2
- md5
- sha1
- sha224
- sha256
- sha384
- sha512

All hash functions expect one string argument. The result is the hashed value of that string. Standard string conversions apply to non-string arguments. All hash functions are supported by SQL version 2015-10-8 and later.

Examples:

```
md2("hello") = "a9046c73e00331af68917d3804f70655"  
md5("hello") = "5d41402abc4b2a76b9719d911017c592"
```

## indexof(String, String)

Returns the first index (0-based) of the second argument as a substring in the first argument. Both arguments are expected as strings. Arguments that are not strings are subjected to standard string conversion rules. This function does not apply to arrays, only to strings. Supported by SQL version 2015-10-8 and later.

Examples:

```
indexof("abcd", "bc") = 1
```

## isNull()

Returns true if the argument is the Null value. Supported by SQL version 2015-10-8 and later.

Examples:

```
isNull(5) = false.
```

```
isNull(null) = true.
```

Argument Type	Result
Int	false
Decimal	false
Boolean	false

Argument Type	Result
String	false
Array	false
Object	false
Null	true
Undefined	false

## isUndefined()

Returns true if the argument is Undefined. Supported by SQL version 2016-03-23 and later.

Examples:

```
isUndefined(5) = false.
```

```
isNull(floor([1,2,3]))) = true.
```

Argument Type	Result
Int	false
Decimal	false
Boolean	false
String	false
Array	false
Object	false
Null	false
Undefined	true

## length(String)

Returns the number of characters in the provided string. Standard conversion rules apply to non-String arguments. Supported by SQL version 2015-10-8 and later.

Examples:

```
length("hi") = 2
```

```
length(false) = 5
```

## ln(Decimal)

Returns the natural logarithm of the argument. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example:  $\ln(e) = 1$ .

Argument Type	Result
Int	Decimal (with double precision), the natural log of the argument.
Decimal	Decimal (with double precision), the natural log of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the natural log of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## log(Decimal)

Returns the base 10 logarithm of the argument. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `log(100) = 2.0`.

Argument Type	Result
Int	Decimal (with double precision), the base 10 log of the argument.
Decimal	Decimal (with double precision), the base 10 log of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the base 10 log of the argument. If the String cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## lower(String)

Returns the lowercase version of the given String. Non-string arguments are converted to strings using the standard conversion rules. Supported by SQL version 2015-10-8 and later.

Examples:

```
lower("HELLO") = "hello".
lower(["HELLO"]) = ["\"hello\"].
```

## lpad(String, Int)

Returns the `String` argument, padded on the left side with the number of spaces specified by the `Int` argument. The `Int` argument must be between 0 and 1000. If the provided value is outside of this valid range, the argument is set to the nearest valid value (0 or 1000). Supported by SQL version 2015-10-8 and later.

Examples:

```
lpad("hello", 2) = " hello".
lpad(1, 3) = " 1"
```

argument Type 1	argument Type 2	Result
String	Int	String, the provided String with a number of spaces.
String	Decimal	The Decimal argument is converted to Int and the String is padded with the specified number of spaces.
String	String	The second argument is rounded down to the specified number of digits and padded with the specified number of spaces. The result is Undefined.
Other Value	Int/Decimal/String	The first value is converted to standard conversions and applied on that String. The result is Undefined.
Any Value	Other Value	Undefined.

## ltrim(String)

Removes all leading whitespace (tabs and spaces) from the provided `String`. Supported by SQL version 2015-10-8 and later.

Example:

```
Ltrim(" h i ") = "hi".
```

Argument Type	Result
Int	The String representation of the Int with all leading whitespace removed.
Decimal	The String representation of the Decimal with all leading whitespace removed.

Argument Type	Result
Boolean	The String representation of the boolean ("true" or "false") with all leading whitespace removed.
String	The argument with all leading whitespace removed.
Array	The String representation of the Array (using standard conversion rules) with all leading whitespace removed.
Object	The String representation of the Object (using standard conversion rules) with all leading whitespace removed.
Null	Undefined.
Undefined	Undefined.

## machinelearning\_predict(modelId)

Use the `machinelearning_predict` function to make predictions using the data from an MQTT message based on an Amazon Machine Learning (Amazon ML) model. Supported by SQL version 2015-10-8 and later. The arguments for the `machinelearning_predict` function are:

`modelId`

The ID of the model against which to run the prediction. The real-time endpoint of the model must be enabled.

`roleArn`

The IAM role that has a policy with `machinelearning:Predict` and `machinelearning:GetMLModel` permissions and allows access to the model against which the prediction is run.

`record`

The data to be passed into the Amazon ML Predict API. This should be represented as a single layer JSON object. If the record is a multi-level JSON object, the record is flattened by serializing its values. For example, the following JSON:

```
{ "key1": {"innerKey1": "value1"}, "key2": 0}
```

would become:

```
{ "key1": "{\"innerKey1\": \"value1\"}", "key2": 0}
```

The function returns a JSON object with the following fields:

`predictedLabel`

The classification of the input based on the model.

`details`

Contains the following attributes:

`PredictiveModelType`

The model type. Valid values are REGRESSION, BINARY, MULTICLASS.

### Algorithm

The algorithm used by Amazon ML to make predictions. The value must be SGD.  
`predictedScores`

Contains the raw classification score corresponding to each label.  
`predictedValue`

The value predicted by Amazon ML.

## mod(Decimal, Decimal)

Returns the remainder of the division of the first argument by the second argument. Supported by SQL version 2015-10-8 and later. You can also use "%" as an infix operator for the same modulo functionality. Supported by SQL version 2015-10-8 and later.

Example: `mod(8, 3) = 2.`

Left Operand	Right Operand	Output
Int	Int	Int, the first argument.
Int/Decimal	Int/Decimal	Decimal, the first argument.
String/Int/Decimal	String/Int/Decimal	If all strings convert to Int, then Int. Otherwise, argument modulo the second argument. Undefined.
Other Value	Other Value	Undefined.

## nanvl(AnyValue, AnyValue)

Returns the first argument if it is a valid `Decimal`. Otherwise, the second argument is returned. Supported by SQL version 2015-10-8 and later.

Example: `Nanvl(8, 3) = 8.`

argument Type 1	argument Type 2	Output
Undefined	Any Value	The second argument.
Null	Any Value	The second argument.
Decimal (NaN)	Any Value	The second argument.
Decimal (not NaN)	Any Value	The first argument.
Other Value	Any Value	The first argument.

## newuuid()

Returns a random 16-byte UUID. Supported by SQL version 2015-10-8 and later.

Example: `newuuid() = 123a4567-b89c-12d3-e456-789012345000`

## numbytes(String)

Returns the number of bytes in the UTF-8 encoding of the provided string. Standard conversion rules apply to non-String arguments. Supported by SQL version 2015-10-8 and later.

Examples:

`numbytes("hi") = 2`

`numbytes("€") = 3`

## principal()

Returns the X.509 certificate fingerprint or thing name, depending on which endpoint, MQTT or HTTP, received the request. Supported by SQL version 2015-10-8 and later.

Example:

`principal() = "ba67293af50bf2506f5f93469686da660c7c844e7b3950fb16813e0d31e9373"`

## parse\_time(String, Long, [String])

Use the `parse_time` function to format a timestamp into a human-readable date/time format. Supported by SQL version 2016-03-23 and later. The arguments for the `parse_time` function are:

pattern

(String) A date/time pattern that conforms to the [ISO 8601](#) standard format. (Specifically, the function supports [Joda-Time formats](#).)

timestamp

(Long) The time to be formatted in milliseconds since Unix epoch. See function [timestamp\(\) \(p. 308\)](#).

timezone

(String) [Optional] The time zone of the formatted date/time. The default is "UTC". The function supports [Joda-Time time zones](#).

Examples:

When this message is published to the topic 'A/B', the payload `{"ts": "1970.01.01 AD at 21:46:40 CST"}` is sent to the S3 bucket:

```
{  
    "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"  
    }  
}
```

```
    }
}
```

When this message is published to the topic 'A/B', a payload similar to `{"ts": "2017.06.09 AD at 17:19:46 UTC"}` (but with the current date/time) is sent to the S3 bucket:

```
{
  "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()` can also be used as a substitution template. For example, when this message is published to the topic 'A/B', the payload is sent to the S3 bucket with key = "2017":

```
{
  "ruleArn": "arn:aws:iot:us-east-2:ACCOUNT_ID:rule/RULE_NAME",
  "topicRulePayload": {
    "sql": "SELECT * FROM 'A/B'",
    "awsIotSqlVersion": "2016-03-23",
    "ruleDisabled": false,
    "actions": [
      {
        "s3": {
          "roleArn": "arn:aws:iam::ACCOUNT_ID:rule:role/ROLE_NAME",
          "bucketName": BUCKET_NAME,
          "key": "${parse_time('yyyy", timestamp(), "UTC")}"
        }
      }
    ],
    "ruleName": "RULE_NAME"
  }
}
```

## power(Decimal, Decimal)

Returns the first argument raised to the second argument. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later. Supported by SQL version 2015-10-8 and later.

Example: `power(2, 5) = 32.0.`

argument Type 1	argument Type 2	Output
Int/Decimal	Int/Decimal	A Decimal (with double precision) raised to the second argument.

argument Type 1	argument Type 2	Output
Int/Decimal/String	Int/Decimal/String	A Decimal (with double raised to the second argument) is converted to decimal and converted to Decimal.
Other Value	Other Value	Undefined.

## rand()

Returns a pseudorandom, uniformly distributed double between 0.0 and 1.0. Supported by SQL version 2015-10-8 and later.

Example:

```
rand() = 0.8231909191640703
```

## regexp\_matches(String, String)

Returns true if the string (first argument) contains a match for the regular expression (second argument).

Example:

```
Regexp_matches("aaaa", "a{2,}") = true.
```

```
Regexp_matches("aaaa", "b") = false.
```

### First argument:

Argument Type	Result
Int	The String representation of the Int.
Decimal	The String representation of the Decimal.
Boolean	The String representation of the boolean ("true" or "false").
String	The String.
Array	The String representation of the Array (using standard conversion rules).
Object	The String representation of the Object (using standard conversion rules).
Null	Undefined.
Undefined	Undefined.

### Second argument:

Must be a valid regex expression. Non-string types are converted to String using the standard conversion rules. Depending on the type, the resultant string might not be a valid regular expression. If the (converted) argument is not valid regex, the result is Undefined.

### Third argument:

Must be a valid regex replacement string. (Can reference capture groups.) Non-string types are converted to `String` using the standard conversion rules. If the (converted) argument is not a valid regex replacement string, the result is `Undefined`.

## regexp\_replace(String, String, String)

Replaces all occurrences of the second argument (regular expression) in the first argument with the third argument. Reference capture groups with `"$"`. Supported by SQL version 2015-10-8 and later.

**Example:**

```
Regexp_replace("abcd", "bc", "x") = "axd".
```

```
Regexp_replace("abcd", "b(.*)d", "$1") = "ac".
```

### First argument:

Argument Type	Result
<code>Int</code>	The <code>String</code> representation of the <code>Int</code> .
<code>Decimal</code>	The <code>String</code> representation of the <code>Decimal</code> .
<code>Boolean</code>	The <code>String</code> representation of the boolean ("true" or "false").
<code>String</code>	The source value.
<code>Array</code>	The <code>String</code> representation of the <code>Array</code> (using standard conversion rules).
<code>Object</code>	The <code>String</code> representation of the <code>Object</code> (using standard conversion rules).
<code>Null</code>	<code>Undefined</code> .
<code>Undefined</code>	<code>Undefined</code> .

### Second argument:

Must be a valid regex expression. Non-string types are converted to `Strings` using the standard conversion rules. Depending on the type, the resultant string might not be a valid regular expression. If the (converted) argument is not a valid regex expression, the result is `Undefined`.

### Third argument:

Must be a valid regex replacement string. (Can reference capture groups.) Non-string types will be converted to `Strings` using the standard conversion rules. If the (converted) argument is not a valid regex replacement string, the result is `Undefined`.

## regexp\_substr(String, String)

Finds the first match of the 2nd parameter (regex) in the first parameter. Reference capture groups with `"$"`. Supported by SQL version 2015-10-8 and later.

**Example:**

```
regexp_substr("hihihello", "hi") => "hi"
```

```
regexp_substr("hihihello", "(hi)*") => "hihi".
```

**First argument:**

Argument Type	Result
Int	The String representation of the Int.
Decimal	The String representation of the Decimal.
Boolean	The String representation of the boolean ("true" or "false").
String	The String argument.
Array	The String representation of the Array (using standard conversion rules).
Object	The String representation of the Object (using standard conversion rules).
Null	Undefined.
Undefined	Undefined.

*Second argument:*

Must be a valid regex expression. Non-string types are converted to Strings using the standard conversion rules. Depending on the type, the resultant string might not be a valid regular expression. If the (converted) argument is not a valid regex expression, the result is Undefined.

*Third argument:*

Must be a valid regex replacement string. (Can reference capture groups.) Non-string types are converted to String using the standard conversion rules. If the argument is not a valid regex replacement string, the result is Undefined.

## rpad(String, Int)

Returns the string argument, padded on the right side with the number of spaces specified in the second argument. The Int argument must be between 0 and 1000. If the provided value is outside of this valid range, the argument is set to the nearest valid value (0 or 1000). Supported by SQL version 2015-10-8 and later.

**Examples:**

`rpad("hello", 2) = "hello "`.

`rpad(1, 3) = "1 "`.

argument Type 1	argument Type 2	Result
String	Int	The String is padded on the right side with a number

argument Type 1	argument Type 2	Result
		of spaces equal to the provided Int.
String	Decimal	The Decimal argument is rounded down to the nearest Int and the string is padded on the right side with a number of spaces equal to the provided Int.

argument Type 1	argument Type 2	Result
String	String	The second argument is converted to a Decimal, which is rounded down to the nearest Int. The String is padded on the right side with a number of spaces equal to the Int value.

argument Type 1	argument Type 2	Result
Other Value	Int/Decimal/String	The first value is converted to a String using the standard conversions, and the rpad function is applied on that String. If it cannot be converted, the result is Undefined.
Any Value	Other Value	Undefined.

## round(Decimal)

Rounds the given Decimal to the nearest Int. If the Decimal is equidistant from two Int values (for example, 0.5), the Decimal is rounded up. Supported by SQL version 2015-10-8 and later.

Example: `Round(1.2) = 1.`

`Round(1.5) = 2.`

`Round(1.7) = 2.`

`Round(-1.1) = -1.`

`Round(-1.5) = -2.`

Argument Type	Result
Int	The argument.
Decimal	Decimal is rounded down to the nearest Int.
String	Decimal is rounded down to the nearest Int. If the string cannot be converted to a Decimal, the result is Undefined.

Argument Type	Result
Other Value	Undefined.

## rtrim(String)

Removes all trailing whitespace (tabs and spaces) from the provided `String`. Supported by SQL version 2015-10-8 and later.

Examples:

```
rtrim(" h i ") = " h i"
```

Argument Type	Result
<code>Int</code>	The <code>String</code> representation of the <code>Int</code> .
<code>Decimal</code>	The <code>String</code> representation of the <code>Decimal</code> .
<code>Boolean</code>	The <code>String</code> representation of the boolean ("true" or "false").
<code>Array</code>	The <code>String</code> representation of the <code>Array</code> (using standard conversion rules).
<code>Object</code>	The <code>String</code> representation of the <code>Object</code> (using standard conversion rules).
<code>Null</code>	Undefined.
<code>Undefined</code>	Undefined

## sign(Decimal)

Returns the sign of the given number. When the sign of the argument is positive, 1 is returned. When the sign of the argument is negative, -1 is returned. If the argument is 0, 0 is returned. Supported by SQL version 2015-10-8 and later.

Examples:

```
sign(-7) = -1.
```

```
sign(0) = 0.
```

```
sign(13) = 1.
```

Argument Type	Result
<code>Int</code>	<code>Int</code> , the sign of the <code>Int</code> value.
<code>Decimal</code>	<code>Int</code> , the sign of the <code>Decimal</code> value.
<code>String</code>	<code>Int</code> , the sign of the <code>Decimal</code> value. The string is converted to a <code>Decimal</code> value, and the sign of the <code>Decimal</code> value is returned. If the <code>String</code> cannot be converted to a <code>Decimal</code> , the result is <code>Undefined</code> . Supported by SQL version 2015-10-8 and later.

Argument Type	Result
Other Value	Undefined.

## sin(Decimal)

Returns the sine of a number in radians. Decimal arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `sin(0) = 0.0`

Argument Type	Result
Int	Decimal (with double precision), the sine of the argument.
Decimal	Decimal (with double precision), the sine of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the sine of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## sinh(Decimal)

Returns the hyperbolic sine of a number. Decimal values are rounded to double precision before function application. The result is a Decimal value of double precision. Supported by SQL version 2015-10-8 and later.

Example: `sinh(2.3) = 4.936961805545957`

Argument Type	Result
Int	Decimal (with double precision), the hyperbolic sine of the argument.
Decimal	Decimal (with double precision), the hyperbolic sine of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the hyperbolic sine of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.

Argument Type	Result
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## substring(String, Int [, Int])

Expects a `String` followed by one or two `Int` values. For a `String` and a single `Int` argument, this function returns the substring of the provided `String` from the provided `Int` index (0-based, inclusive) to the end of the `String`. For a `String` and two `Int` arguments, this function returns the substring of the provided `String` from the first `Int` index argument (0-based, inclusive) to the second `Int` index argument (0-based, exclusive). Indices that are less than zero are set to zero. Indices that are greater than the `String` length are set to the `String` length. For the three argument version, if the first index is greater than (or equal to) the second index, the result is the empty `String`.

If the arguments provided are not (`String, Int`), or (`String, Int, Int`), the standard conversions are applied to the arguments to attempt to convert them into the correct types. If the types cannot be converted, the result of the function is `Undefined`. Supported by SQL version 2015-10-8 and later.

Examples:

```
substring("012345", 0) = "012345".
substring("012345", 2) = "2345".
substring("012345", 2.745) = "2345".
substring(123, 2) = "3".
substring("012345", -1) = "012345".
substring(true, 1.2) = "rue".
substring(false, -2.411E247) = "false".
substring("012345", 1, 3) = "12".
substring("012345", -50, 50) = "012345".
substring("012345", 3, 1) = "".
```

## sqrt(Decimal)

Returns the square root of a number. `Decimal` arguments are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `sqrt(9) = 3.0`.

Argument Type	Result
<code>Int</code>	The square root of the argument.
<code>Decimal</code>	The square root of the argument.

Argument Type	Result
Boolean	Undefined.
String	The square root of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## startswith(String, String)

Returns Boolean, whether the first string argument starts with the second string argument. If either argument is Null or Undefined, the result is Undefined. Supported by SQL version 2015-10-8 and later.

Example:

```
startswith("ranger", "ran") = true
```

argument Type 1	argument Type 2	Result
String	String	Whether the first string starts with the second string
Other Value	Other Value	Both arguments are converted to strings. If either argument is Null or Undefined, the result is Undefined.

## tan(Decimal)

Returns the tangent of a number in radians. Decimal values are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `tan(3) = -0.1425465430742778`

Argument Type	Result
Int	Decimal (with double precision), the tangent of the argument.
Decimal	Decimal (with double precision), the tangent of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the tangent of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.

Argument Type	Result
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## tanh(Decimal)

Returns the hyperbolic tangent of a number in radians. Decimal values are rounded to double precision before function application. Supported by SQL version 2015-10-8 and later.

Example: `tanh(2.3) = 0.9800963962661914`

Argument Type	Result
Int	Decimal (with double precision), the hyperbolic tangent of the argument.
Decimal	Decimal (with double precision), the hyperbolic tangent of the argument.
Boolean	Undefined.
String	Decimal (with double precision), the hyperbolic tangent of the argument. If the string cannot be converted to a Decimal, the result is Undefined.
Array	Undefined.
Object	Undefined.
Null	Undefined.
Undefined	Undefined.

## timestamp()

Returns the current timestamp in milliseconds from 00:00:00 Coordinated Universal Time (UTC), Thursday, 1 January 1970, as observed by the AWS IoT rules engine. Supported by SQL version 2015-10-8 and later.

Example: `timestamp() = 1481825251155`

## topic(Decimal)

Returns the topic to which the message that triggered the rule was sent. If no parameter is specified, the entire topic is returned. The Decimal parameter is used to specify a specific topic segment, with 1 designating the first segment. For the topic `foo/bar/baz`, `topic(1)` returns `foo`, `topic(2)` returns `bar`, and so on. Supported by SQL version 2015-10-8 and later.

Examples:

`topic() = "things/myThings/thingOne"`

`topic(1) = "things"`

When [Basic Ingest \(p. 319\)](#) is used, the initial prefix of the topic (`$aws/rules/rule-name`) is not available to the `topic()` function. For example, given the topic:

```
$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights
topic() = "Buildings/Building5/Floor2/Room201/Lights"
topic(3) = "Floor2"
```

## traceid()

Returns the trace ID (UUID) of the MQTT message, or `Undefined` if the message wasn't sent over MQTT. Supported by SQL version 2015-10-8 and later.

Example:

```
traceid() = "12345678-1234-1234-1234-123456789012"
```

## trunc(Decimal, Int)

Truncates the first argument to the number of `Decimal` places specified by the second argument. If the second argument is less than zero, it is set to zero. If the second argument is greater than 34, it is set to 34. Trailing zeroes are stripped from the result. Supported by SQL version 2015-10-8 and later.

Examples:

```
trunc(2.3, 0) = 2.
trunc(2.3123, 2) = 2.31.
trunc(2.888, 2) = 2.88.
trunc(2.00, 5) = 2.
```

argument Type 1	argument Type 2	Result
Int	Int	The source value.
Int/Decimal	Int/Decimal	The first argument is converted to a Decimal by the second argument. If the second argument is less than zero, it is set to zero. If the second argument is greater than 34, it is set to 34. Trailing zeroes are stripped from the result.
Int/Decimal/String	Int/Decimal	The first argument is converted to a Decimal by the second argument. If the second argument is less than zero, it is set to zero. If the second argument is greater than 34, it is set to 34. Trailing zeroes are stripped from the result. If the conversion fails, the result is <code>Undefined</code> .
Other Value		<code>Undefined</code> .

## trim(String)

Removes all leading and trailing whitespace from the provided `String`. Supported by SQL version 2015-10-8 and later.

Example:

```
Trim(" hi ") = "hi"
```

Argument Type	Result
Int	The String representation of the Int with all leading and trailing whitespace removed.
Decimal	The String representation of the Decimal with all leading and trailing whitespace removed.
Boolean	The String representation of the Boolean ("true" or "false") with all leading and trailing whitespace removed.
String	The String with all leading and trailing whitespace removed.
Array	The String representation of the Array using standard conversion rules.
Object	The String representation of the Object using standard conversion rules.
Null	Undefined.
Undefined	Undefined.

## upper(String)

Returns the uppercase version of the given String. Non-String arguments are converted to String using the standard conversion rules. Supported by SQL version 2015-10-8 and later.

Examples:

```
upper("hello") = "HELLO"
upper(["hello"]) = "[\"HELLO\"]"
```

## SELECT Clause

The AWS IoT SELECT clause is essentially the same as the ANSI SQL SELECT clause, with some minor differences.

You can use the SELECT clause to extract information from incoming MQTT messages. `SELECT *`  can be used to retrieve the entire incoming message payload. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}
SQL statement: SELECT * FROM 'a/b'
Outgoing payload: {"color":"red", "temperature":50}
```

If the payload is a JSON object, you can reference keys in the object. Your outgoing payload contains the key-value pair. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}
SQL statement: SELECT color FROM 'a/b'
Outgoing payload: {"color":"red"}
```

You can use the AS keyword to rename keys. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}
```

```
SQL:SELECT color AS my_color FROM 'a/b'  
Outgoing payload: {"my_color":"red"}
```

You can select multiple items by separating them with a comma. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}  
SQL: SELECT color as my_color, temperature as farenheit FROM 'a/b'  
Outgoing payload: {"my_color":"red", "farenheit":50}
```

You can select multiple items including '\*' to add items to the incoming payload. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}  
SQL: SELECT *, 15 as speed FROM 'a/b'  
Outgoing payload: {"color":"red", "temperature":50, "speed":15}
```

You can use the "VALUE" keyword to produce outgoing payloads that are not JSON objects. You may only select one item. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}  
SQL: SELECT VALUE color FROM 'a/b'  
Outgoing payload: "red"
```

You can use '.' syntax to drill into nested JSON objects in the incoming payload. For example:

```
Incoming payload published on topic 'a/b': {"color":{"red":255,"green":0,"blue":0},  
"temperature":50}  
SQL: SELECT color.red as red_value FROM 'a/b'  
Outgoing payload: {"red_value":255}
```

You can use functions (see [Functions \(p. 276\)](#)) to transform the incoming payload. Parentheses can be used for grouping. For example:

```
Incoming payload published on topic 'a/b': {"color":"red", "temperature":50}  
SQL: SELECT (temperature - 32) * 5 / 9 AS celsius, upper(color) as my_color FROM 'a/b'  
Outgoing payload: {"celsius":10, "my_color":"RED"}
```

## Working with Binary Payloads

When the message payload should be handled as raw binary data (rather than a JSON object), you can use the '\*' operator to refer to it in a `SELECT` clause.

These rules must be followed to use '\*' to refer to the message payload as raw binary data:

1. The SQL statement and templates must not refer to JSON names other than '\*'.
2. The `SELECT` statement must have '\*' as the only item, or must have only functions, for example:

```
SELECT * FROM 'a/b'
```

```
SELECT encode(*, 'base64') AS data, timestamp() AS ts FROM 'a/b'
```

## Binary Payload Examples

The following `SELECT` clause can be used with binary payloads because it doesn't refer to any JSON names.

```
SELECT * FROM 'a/b'
```

The following SELECT cannot be used with binary payloads because it refers to device\_type in the WHERE clause.

```
SELECT * FROM 'a/b' WHERE device_type = 'thermostat'
```

The following SELECT cannot be used with binary payloads because it violates rule #2.

```
SELECT *, timestamp() AS timestamp FROM 'a/b'
```

The following SELECT can be used with binary payloads because it doesn't violate rule #1 or rule #2.

```
SELECT * FROM 'a/b' WHERE timestamp() % 12 = 0
```

The following AWS IoT rule cannot be used with payloads because it violates rule #1.

```
{
    "sql": "SELECT * FROM 'a/b'"
    "actions": [
        {
            "republish": {
                "topic": "device/${device_id}"
            }
        }
    ]
}
```

## FROM Clause

The FROM clause subscribes your rule to a topic or topic filter. The rule is triggered for each message sent to an MQTT topic that matches the topic filter specified here. A topic filter allows you to subscribe to a group of similar topics.

### Example:

Incoming payload published on topic 'a/b': {temperature: 50}

Incoming payload published on topic 'a/c': {temperature: 50}

SQL: "SELECT temperature AS t FROM 'a/b'".

The rule is subscribed to 'a/b', so the incoming payload is passed to the rule, and the outgoing payload (passed to the rule actions) is: {t: 50}. The rule is not subscribed to 'a/c', so the rule is not triggered for the message published on 'a/c'.

### # Wildcard Example:

You can use the '#' (multi-level) wildcard character to match one or more particular path elements:

Incoming payload published on topic 'a/b': {temperature: 50}.

Incoming payload published on topic 'a/c': {temperature: 60}.

Incoming payload published on topic 'a/e/f': {temperature: 70}.

Incoming payload published on topic 'b/x': {temperature: 80}.

SQL: "SELECT temperature AS t FROM 'a/#'".

The rule is subscribed to any topic beginning with 'a', so it is executed three times, sending outgoing payloads of {t: 50} (for a/b), {t: 60} (for a/c), and {t: 70} (for a/e/f) to its actions. It is not subscribed to 'b/x', so the rule is not triggered for the {temperature: 80} message.

**+ Wildcard Example:**

You can use the '+' (single-level) wildcard character to match any one particular path element:

Incoming payload published on topic 'a/b': {temperature: 50}.

Incoming payload published on topic 'a/c': {temperature: 60}.

Incoming payload published on topic 'a/e/f': {temperature: 70}.

Incoming payload published on topic 'b/x': {temperature: 80}.

SQL: "SELECT temperature AS t FROM 'a/+'".

The rule is subscribed to all topics with two path elements where the first element is 'a'. The rule is executed for the messages sent to 'a/b' and 'a/c', but not 'a/e/f' or 'b/x'.

## WHERE Clause

The WHERE clause determines if the actions specified by a rule are carried out. If the WHERE clause evaluates to true, the rule actions are performed. Otherwise, the rule actions are not performed.

Example:

Incoming payload published on a/b: {"color": "red", "temperature": 40}.

SQL: SELECT color AS my\_color FROM 'a/b' WHERE temperature > 50 AND color <> 'red'.

In this case, the rule would be triggered, but the actions specified by the rule would not be performed. There would be no outgoing payload.

You can use functions and operators in the WHERE clause. However, you cannot reference any aliases created with the AS keyword in the SELECT. (The WHERE clause is evaluated first, to determine if SELECT is evaluated.)

## Literals

You can directly specify literal objects in the SELECT and WHERE clauses of your rule SQL, which can be useful for passing information.

**Note**

Literals are available only when using SQL version 2016-03-23 or later.

JSON object syntax is used (key-value pairs, comma-separated, where keys are strings and values are JSON values, wrapped in curly brackets {}). For example:

Incoming payload published on topic a/b: {"lat\_long": [47.606, -122.332]}

SQL statement: SELECT {'latitude': get(lat\_long, 0), 'longitude': get(lat\_long, 1)} as lat\_long FROM 'a/b'

The resulting outgoing payload would be: {"lat\_long": {"latitude": 47.606, "longitude": -122.332}}.

You can also directly specify arrays in the SELECT and WHERE clauses of your rule SQL, which allows you to group information. JSON syntax is used (wrap comma-separated items in square brackets [] to create an array literal). For example:

Incoming payload published on topic a/b: {"lat": 47.696, "long": -122.332}

SQL statement: `SELECT [lat,long] as lat_long FROM 'a/b'`

The resulting output payload would be: {"lat\_long": [47.606, -122.332]}.

## Case Statements

Case statements can be used for branching execution, like a switch statement, or if/else statements.

Syntax:

```
CASE v WHEN t[1] THEN r[1]
        WHEN t[2] THEN r[2] ...
        WHEN t[n] THEN r[n]
        ELSE r[e] END
```

The expression `v` is evaluated and matched for equality against each `t[i]` expression. If a match is found, the corresponding `r[i]` expression becomes the result of the case statement. If there is more than one possible match, the first match is selected. If there are no matches, the else statement's `r[e]` is used as the result. If there is no match and no else statement, the result of the case statement is `Undefined`. For example:

Incoming payload published on topic a/b: {"color": "yellow"}

SQL statement: `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 'a/b'`

The resulting output payload would be: {"instructions": "caution"}.

Case statements require at least one WHEN clause. An ELSE clause is not required.

**Note**

If `v` is `Undefined`, the result of the case statement is `Undefined`.

## JSON Extensions

You can use the following extensions to ANSI SQL syntax to make it easier to work with nested JSON objects.

### ":" Operator

This operator accesses members in embedded JSON objects and functions identically to ANSI SQL and JavaScript. For example:

```
SELECT foo.bar AS bar.baz FROM 'a/b'
```

### \*

This functions in the same way as the \* wildcard in ANSI SQL. It's used in the SELECT clause only and creates a new JSON object containing the message data. If the message payload is not in JSON format, \* returns the entire message payload as raw bytes. For example:

```
SELECT * FROM 'a/b'
```

### Applying a Function to an Attribute Value

The following is an example JSON payload that might be published by a device:

```
{
  "deviceid" : "iot123",
  "temp" : 54.98,
  "humidity" : 32.43,
  "coords" : {
    "latitude" : 47.615694,
    "longitude" : -122.3359976
  }
}
```

The following example applies a function to an attribute value in a JSON payload:

```
SELECT temp, md5(deviceid) AS hashed_id FROM topic/#
```

The result of this query is the following JSON object:

```
{
  "temp": 54.98,
  "hashed_id": "e37f81fb397e595c4aeb5645b8cbbbd1"
}
```

## Substitution Templates

You can use a substitution template to augment the JSON data returned when a rule is triggered and AWS IoT performs an action. The syntax for a substitution template is `#{expression}`, where *expression* can be any expression supported by AWS IoT in a SELECT or WHERE clause. This includes functions, operators, and information present in the original message payload. Because an expression in a substitution template is evaluated separately from the "SELECT ..." statement, you cannot reference an alias created using the AS clause. For more information about supported expressions, see [AWS IoT SQL Reference \(p. 267\)](#).

Substitution templates appear in the SELECT clause within a rule:

```
{
  "sql": "SELECT *, topic() AS topic FROM 'my/iot/topic'",
  "ruleDisabled": false,
  "actions": [
    {
      "republish": {
        "topic": "${topic()}/republish",
        "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
      }
    }
  ]
}
```

If this rule is triggered by the following JSON:

```
{
  "deviceid" : "iot123",
  "temp" : 54.98,
  "humidity" : 32.43,
  "coords" : {
```

```
        "latitude" : 47.615694,  
        "longitude" : -122.3359976  
    }  
}
```

Here is the output of the rule:

```
{  
    "coords":{  
        "longitude": -122.3359976,  
        "latitude": 47.615694  
    },  
    "humidity": 32.43,  
    "temp": 54.98,  
    "deviceid": "iot123",  
    "topic": "my/iot/topic"  
}
```

## SQL Versions

The AWS IoT rules engine uses an SQL-like syntax to select data from MQTT messages. The SQL statements are interpreted based on an SQL version specified with the `awsIotSqlVersion` property in a JSON document that describes the rule. For more information about the structure of JSON rule documents, see [Creating a Rule \(p. 245\)](#). The `awsIotSqlVersion` property allows you to specify which version of the AWS IoT SQL rules engine you want to use. When a new version is deployed, you can continue to use an older version or change your rule to use the new version. Your current rules continue to use the version with which they were created.

The following JSON example shows how to specify the SQL version using the `awsIotSqlVersion` property:

```
{  
    "sql": "expression",  
    "ruleDisabled": false,  
    "awsIotSqlVersion": "2016-03-23",  
    "actions": [  
        {  
            "republish": {  
                "topic": "my-mqtt-topic",  
                "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"  
            }  
        }  
    ]  
}
```

Current supported versions are:

- 2015-10-08, the original SQL version built on 2015-10-08.
- 2016-03-23, the SQL version built on 2016-03-23.
- beta, the most recent beta SQL version. The use of this version might introduce breaking changes to your rules.

## What's New in the 2016-03-23 SQL Rules Engine Version

- Fixes for selecting nested JSON objects.

- Fixes for array queries.
- Inter-object query support.
- Support to output an array as a top-level object.
- Addition of the encode (*value, encodingScheme*) function, which can be applied on both JSON and non-JSON format data.

## Inter-Object Queries

This feature allows you to query for an attribute in a JSON object. For example, given the following MQTT message:

```
{
  "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 }
  ]
}
```

And the following rule:

```
SELECT (SELECT v FROM e WHERE n = 'temperature') as temperature FROM 'my/topic'
```

The rule generates the following output:

```
{"temperature": [{"v":22.5}]}
```

Using the same MQTT message, given a slightly more complicated rule such as:

```
SELECT get((SELECT v FROM e WHERE n = 'temperature'),1).v as temperature FROM 'topic'
```

The rule generates the following output:

```
{"temperature":22.5}
```

## Output an Array as a Top-Level Object

This feature allows a rule to return an array as a top-level object. For example, given the following MQTT message:

```
{
  "a": {"b":"c"},
  "arr":[1,2,3,4]
}
```

And the following rule:

```
SELECT VALUE arr FROM 'topic'
```

The rule generates the following output:

```
[1,2,3,4]
```

## Encode Function

Encodes the payload, which potentially might be non-JSON data, into its string representation based on the specified encoding scheme.

# Basic Ingest

Basic Ingest enables you to securely send device data to the AWS services supported by [AWS IoT Rule Actions \(p. 249\)](#) without incurring [messaging costs](#). Basic Ingest optimizes data flow by removing the publish/subscribe message broker from the ingestion path, so it's more cost effective while keeping the security and data processing features of AWS IoT.

To use Basic Ingest, you send messages from your devices or applications with topic names that start with `$aws/rules/rule-name` as their first three levels, where *rule-name* is the name of your AWS IoT Rule to trigger.

You can continue to use an existing rule with Basic Ingest by just adding the Basic Ingest prefix (`$aws/rules/rule-name`) to the message topic by which you normally trigger the rule. For example, if you have a rule named "BuildingManager" that is triggered by messages with topics like "Buildings/Building5/Floor2/Room201/Lights" ("sql": "SELECT \* FROM 'Buildings/#'"), you can trigger the same rule with Basic Ingest by sending a message with topic `$aws/rules/BuildingManager/Buildings/Building5/Floor2/Room201/Lights`.

Be aware that:

- Your devices and rules cannot subscribe to Basic Ingest reserved topics. (See [Reserved Topics \(p. 235\)](#) for details.)
- If you need a publish/subscribe broker to distribute messages to multiple subscribers (for example, to deliver messages to other devices as well as the Rules Engine), you should continue to use the AWS IoT message broker to handle the message distribution. Just publish your messages on topics other than Basic Ingest topics.

## To use Basic Ingest

Make sure your device or application is using a [policy \(p. 193\)](#) that has publish permissions on `$aws/rules/*`. Or you can specify permission for individual rules with `$aws/rules/rule-name/*` in the policy. Otherwise, your devices and applications can continue to use their existing connections with AWS IoT Core.

When the message reaches the Rules Engine, there is no difference in execution or error handling between rules triggered from Basic Ingest and those triggered through message broker subscriptions.

You can, of course, create rules for use with Basic Ingest. Keep in mind the following:

- The initial prefix of a Basic Ingest topic (`$aws/rules/rule-name`) isn't available to the [topic\(Decimal\) \(p. 308\)](#) function.
- If you define a rule that is triggered only with Basic Ingest, the `FROM` clause is optional in the `sql` field of the `rule` definition. It's still required if the rule will also be triggered by other messages that must be sent through the message broker (for example, because those other messages must be distributed to multiple subscribers). See [AWS IoT SQL Reference \(p. 267\)](#).
- The first three levels of the Basic Ingest topic (`$aws/rules/rule-name`) are not counted toward the eight segment length limit or toward the 256 total character limit for a topic. Otherwise, the same restrictions apply as documented in [AWS IoT Limits](#).
- If a message is received with a Basic Ingest topic that specifies an inactive rule or a rule that doesn't exist, an error log is created in a CloudWatch log to help you with debugging (See [Rules Engine Logs \(p. 640\)](#)). A "RuleNotFound" metric is indicated and you can create alarms on this metric. See Rule Metrics in [AWS IoT Metrics \(p. 620\)](#).

- You can still publish with QoS 1 on Basic Ingest topics. You will receive a PUBACK once the message is successfully delivered to the Rules Engine. Receiving a PUBACK does not mean that your rule actions completed successfully. You can configure an error action to handle errors during action execution. See [Error Handling \(Error Action\) \(p. 265\)](#).

# Device Shadow Service for AWS IoT

A device's *shadow* is a JSON document that is used to store and retrieve current state information for a device. The Device Shadow service maintains a shadow for each device you connect to AWS IoT. You can use the shadow to get and set the state of a device over MQTT or HTTP, regardless of whether the device is connected to the Internet. Each device's shadow is uniquely identified by the name of the corresponding thing.

## Contents

- [Device Shadow Service Data Flow \(p. 321\)](#)
- [Device Shadow Service Documents \(p. 328\)](#)
- [Using Shadows \(p. 332\)](#)
- [Device Shadow RESTful API \(p. 340\)](#)
- [Shadow MQTT Topics \(p. 343\)](#)
- [Shadow Document Syntax \(p. 350\)](#)
- [Shadow Error Messages \(p. 352\)](#)

## Device Shadow Service Data Flow

The Device Shadow service acts as an intermediary, allowing devices and applications to retrieve and update a device's shadow.

To illustrate how devices and applications communicate with the Device Shadow service, this section walks you through the use of the AWS IoT MQTT client and the AWS CLI to simulate communication between an internet-connected light bulb, an application, and the Device Shadow service.

The Device Shadow service uses MQTT topics to facilitate communication between applications and devices. To see how this works, use the AWS IoT MQTT client to subscribe to the following MQTT topics with QoS 1:

\$aws/things/myLightBulb/shadow/update/accepted

The Device Shadow service sends messages to this topic when an update is successfully made to the device's shadow.

\$aws/things/myLightBulb/shadow/update/rejected

The Device Shadow service sends messages to this topic when an update to the device's shadow is rejected.

\$aws/things/myLightBulb/shadow/update/delta

The Device Shadow service sends messages to this topic when a difference is detected between the reported and desired sections of the device's shadow. For more information, see [/update/delta \(p. 346\)](#).

\$aws/things/myLightBulb/shadow/get/accepted

The Device Shadow service sends messages to this topic when a request for the device's shadow is made successfully.

\$aws/things/myLightBulb/shadow/get/rejected

The Device Shadow service sends messages to this topic when a request for the device's shadow is rejected.

`$aws/things/myLightBulb/shadow/delete/accepted`

The Device Shadow service sends messages to this topic when the device's shadow is deleted.

`$aws/things/myLightBulb/shadow/delete/rejected`

The Device Shadow service sends messages to this topic when a request to delete the device's shadow is rejected.

`$aws/things/myLightBulb/shadow/update/documents`

The Device Shadow service publishes a state document to this topic whenever an update to the device's shadow is successfully performed.

To learn more about all of the MQTT topics used by the Device Shadow service, see [Shadow MQTT Topics \(p. 343\)](#).

**Note**

We recommend that you subscribe to the `.../rejected` topics to see any errors sent by the Device Shadow service.

When the light bulb comes online, it sends its current state to the Device Shadow service by sending an MQTT message to the `$aws/things/myLightBulb/shadow/update` topic.

**Note**

Device Shadows are created the first time an attempt is made to update it. The Device Shadow service will detect that a shadow doesn't exist and will create one. If the shadow exists, it will be updated.

To simulate this, use the AWS IoT MQTT client to publish the following message to the `$aws/things/myLightBulb/shadow/update` topic:

```
{  
    "state": {  
        "reported": {  
            "color": "red"  
        }  
    }  
}
```

This message sets the color of the light bulb to "red."

The Device Shadow service responds by sending the following message to the `$aws/things/myLightBulb/shadow/update/accepted` topic:

```
{  
    "messageNumber": 4,  
    "payload": {  
        "state": {  
            "reported": {  
                "color": "red"  
            }  
        },  
        "metadata": {  
            "reported": {  
                "color": {  
                    "timestamp": 1469564492  
                }  
            }  
        },  
        "version": 1,  
        "timestamp": 1469564492  
    },  
}
```

```

    "qos": 0,
    "timestamp": 1469564492848,
    "topic": "$aws/things/myLightBulb/shadow/update/accepted"
}

```

This message indicates the Device Shadow service received the UPDATE request and updated the device's shadow. If the shadow doesn't exist, it is created. Otherwise, the shadow is updated with the data in the message. If you don't see a message published to `$aws/things/myLightBulb/shadow/update/accepted`, check the subscription to `$aws/things/myLightBulb/shadow/update/rejected` to see any error messages.

In addition, the Device Shadow service publishes the following message to the `$aws/things/myLightBulb/shadow/update/documents` topic.

```

{
    "previous": null,
    "current": {
        "state": {
            "reported": {
                "color": "red"
            }
        },
        "metadata": {
            "reported": {
                "color": {
                    "timestamp": 1483467764
                }
            }
        },
        "version": 1
    },
    "timestamp": 1483467764
}

```

Messages are published to the `/update/documents` topic whenever an update to the device's shadow is successfully performed. For more information of the contents of messages published to this topic, see [Shadow MQTT Topics \(p. 343\)](#).

An application that interacts with the light bulb comes online and requests the light bulb's current state. The application sends an empty message to the `$aws/things/myLightBulb/shadow/get` topic. To simulate this, use the AWS IoT MQTT client to publish an empty message ("") to the `$aws/things/myLightBulb/shadow/get` topic.

The Device Shadow service responds by publishing the requested shadow to the `$aws/things/myLightBulb/shadow/get/accepted` topic:

```

{
    "messageNumber": 1,
    "payload": {
        "state": {
            "reported": {
                "color": "red"
            }
        },
        "metadata": {
            "reported": {
                "color": {
                    "timestamp": 1469564492
                }
            }
        },
        "version": 1,

```

```

        "timestamp": 1469564571
    },
    "qos": 0,
    "timestamp": 1469564571533,
    "topic": "$aws/things/myLightBulb/shadow/get/accepted"
}

```

If you don't see a message on the `$aws/things/myLightBulb/shadow/get/accepted` topic, check the `$aws/things/myLightBulb/shadow/get/rejected` topic for any error messages.

The application displays this information to the user, and the user requests a change to the light bulb's color (from red to green). To do this, the application publishes a message on the `$aws/things/myLightBulb/shadow/update` topic:

```
{
    "state": {
        "desired": {
            "color": "green"
        }
    }
}
```

To simulate this, use the AWS IoT MQTT client to publish the preceding message to the `$aws/things/myLightBulb/shadow/update` topic.

The Device Shadow service responds by sending a message to the `$aws/things/myLightBulb/shadow/update/accepted` topic:

```
{
    "messageNumber": 5,
    "payload": {
        "state": {
            "desired": {
                "color": "green"
            }
        },
        "metadata": {
            "desired": {
                "color": {
                    "timestamp": 1469564658
                }
            }
        },
        "version": 2,
        "timestamp": 1469564658
    },
    "qos": 0,
    "timestamp": 1469564658286,
    "topic": "$aws/things/myLightBulb/shadow/update/accepted"
}
```

and to the `$aws/things/myLightBulb/shadow/update/delta` topic:

```
{
    "messageNumber": 1,
    "payload": {
        "version": 2,
        "timestamp": 1469564658,
        "state": {
            "color": "green"
        }
    },

```

```

    "metadata": {
        "color": {
            "timestamp": 1469564658
        }
    },
    "qos": 0,
    "timestamp": 1469564658309,
    "topic": "$aws/things/myLightBulb/shadow/update/delta"
}

```

The Device Shadow service publishes a message to this topic when it accepts a shadow update and the resulting shadow contains different values for desired and reported states.

The Device Shadow service also publishes a message to the `$aws/things/myLightBulb/shadow/update/documents` topic:

```

{
    "previous": {
        "state": {
            "reported": {
                "color": "red"
            }
        },
        "metadata": {
            "reported": {
                "color": {
                    "timestamp": 1483467764
                }
            }
        },
        "version": 1
    },
    "current": {
        "state": {
            "desired": {
                "color": "green"
            },
            "reported": {
                "color": "red"
            }
        },
        "metadata": {
            "desired": {
                "color": {
                    "timestamp": 1483468612
                }
            },
            "reported": {
                "color": {
                    "timestamp": 1483467764
                }
            }
        },
        "version": 2
    },
    "timestamp": 1483468612
}

```

The light bulb is subscribed to the `$aws/things/myLightBulb/shadow/update/delta` topic, so it receives the message, changes its color, and publishes its new state. To simulate this, use the AWS IoT MQTT client to publish the following message to the `$aws/things/myLightBulb/shadow/update` topic to update the shadow state:

```
{
  "state": {
    "reported": {
      "color": "green"
    },
    "desired": null
  }
}
```

In response, the Device Shadow service sends a message to the `$aws/things/myLightBulb/shadow/update/accepted` topic:

```
{
  "messageNumber": 6,
  "payload": {
    "state": {
      "reported": {
        "color": "green"
      },
      "desired": null
    },
    "metadata": {
      "reported": {
        "color": {
          "timestamp": 1469564801
        }
      },
      "desired": {
        "timestamp": 1469564801
      }
    },
    "version": 3,
    "timestamp": 1469564801
  },
  "qos": 0,
  "timestamp": 1469564801673,
  "topic": "$aws/things/myLightBulb/shadow/update/accepted"
}
```

and to the `$aws/things/myLightBulb/shadow/update/documents` topic:

```
{
  "previous": {
    "state": {
      "reported": {
        "color": "red"
      }
    },
    "metadata": {
      "reported": {
        "color": {
          "timestamp": 1483470355
        }
      }
    },
    "version": 3
  },
  "current": {
    "state": {
      "reported": {
        "color": "green"
      }
    }
  }
}
```

```
{  
    "metadata": {  
        "reported": {  
            "color": {  
                "timestamp": 1483470364  
            }  
        }  
    },  
    "version": 4  
},  
"+timestamp": 1483470364  
}
```

The app requests the current state from the Device Shadow service and displays the most recent state data. To simulate this, run the following command:

```
aws iot-data get-thing-shadow --thing-name "myLightBulb" "output.txt" && cat "output.txt"
```

**Note**

On Windows, omit the `&& cat "output.txt"`, which displays the contents of `output.txt` to the console. You can open the file in Notepad or any text editor to see the contents of the shadow.

The Device Shadow service returns the shadow document:

```
{  
    "state": {  
        "reported": {  
            "color": "green"  
        }  
    },  
    "metadata": {  
        "reported": {  
            "color": {  
                "timestamp": 1469564801  
            }  
        }  
    },  
    "version": 3,  
    "timestamp": 1469564864  
}
```

To delete the device's shadow, publish an empty message to the `$aws/things/myLightBulb/shadow/delete` topic. AWS IoT responds by publishing a message to the `$aws/things/myLightBulb/shadow/delete/accepted` topic:

```
{  
    "version" : 1,  
    "timestamp" : 1488565234  
}
```

## Detecting a Thing Is Connected

To determine if a device is currently connected, include a connected setting in the shadow and use an MQTT Last Will and Testament (LWT) message that sets the connected setting to `false` if a device is disconnected due to error.

**Note**

Currently, LWT messages sent to AWS IoT reserved topics (topics that begin with \$) are ignored by the AWS IoT Device Shadow service, but are still processed by subscribed clients and by the AWS IoT rules engine. If you want the Device Shadow service to receive LWT messages, register

an LWT message to a non-reserved topic and create a rule that republishes the message on the reserved topic. The following example shows how to create a republish rule that listens for messages from the `my/things/myLightBulb/update` topic and republishes it to `$aws/things/myLightBulb/shadow/update`.

```
{
  "rule": {
    "ruleDisabled": false,
    "sql": "SELECT * FROM 'my/things/myLightBulb/update'",
    "description": "Turn my/things/ into $aws/things/",
    "actions": [
      {
        "republish": {
          "topic": "$aws/things/myLightBulb/shadow/update",
          "roleArn": "arn:aws:iam::123456789012:role/aws_iot_republish"
        }
      }
    ]
  }
}
```

When a device connects, it registers an LWT that sets the connected setting to `false`:

```
{
  "state": {
    "reported": {
      "connected": "false"
    }
  }
}
```

It also publishes a message on its update topic (`$aws/things/myLightBulb/shadow/update`), setting its connected state to `true`:

```
{
  "state": {
    "reported": {
      "connected": "true"
    }
  }
}
```

When the device disconnects gracefully, it publishes a message on its update topic and sets its connected state to `false`:

```
{
  "state": {
    "reported": {
      "connected": "false"
    }
  }
}
```

If the device disconnects due to an error, its LWT message is posted automatically to the update topic.

## Device Shadow Service Documents

The Device Shadow service respects all rules of the JSON specification. Values, objects, and arrays are stored in the device's shadow document.

## Contents

- [Document Properties \(p. 329\)](#)
- [Versioning of a Device Shadow \(p. 329\)](#)
- [Client Token \(p. 330\)](#)
- [Example Document \(p. 330\)](#)
- [Empty Sections \(p. 330\)](#)
- [Arrays \(p. 331\)](#)

# Document Properties

A device's shadow document has the following properties:

`state`

`desired`

The desired state of the thing. Applications can write to this portion of the document to update the state of a thing without having to directly connect to a thing.

`reported`

The reported state of the thing. Things write to this portion of the document to report their new state. Applications read this portion of the document to determine the state of a thing.

`metadata`

Information about the data stored in the `state` section of the document. This includes timestamps, in Epoch time, for each attribute in the `state` section, which enables you to determine when they were updated.

### Note

Metadata do not contribute to the document size for service limits or pricing. For more information, see [AWS IoT Service Limits](#).

`timestamp`

Indicates when the message was transmitted by AWS IoT. By using the timestamp in the message and the timestamps for individual attributes in the `desired` or `reported` section, a thing can determine how old an updated item is, even if it doesn't feature an internal clock.

`clientToken`

A string unique to the device that enables you to associate responses with requests in an MQTT environment.

`version`

The document version. Every time the document is updated, this version number is incremented. Used to ensure the version of the document being updated is the most recent.

For more information, see [Shadow Document Syntax \(p. 350\)](#).

## Versioning of a Device Shadow

The Device Shadow service supports versioning on every update message (both request and response), which means that with every update of a device's shadow, the version of the JSON document is incremented. This ensures two things:

- A client can receive an error if it attempts to overwrite a shadow using an older version number. The client is informed it must resync before it can update a device's shadow.
- A client can decide not to act on a received message if the message has a lower version than the version stored by the client.

In some cases, a client might bypass version matching by not submitting a version.

## Client Token

You can use a client token with MQTT-based messaging to verify the same client token is contained in a request and request response. This ensures the response and request are associated.

### Note

The client token can be no longer than 64 bytes. A client token that is longer than 64 bytes will cause a 400 (Bad Request) response and an *Invalid clientToken* error message.

## Example Document

Here is an example shadow document:

```
{  
    "state" : {  
        "desired" : {  
            "color" : "RED",  
            "sequence" : [ "RED", "GREEN", "BLUE" ]  
        },  
        "reported" : {  
            "color" : "GREEN"  
        }  
    },  
    "metadata" : {  
        "desired" : {  
            "color" : {  
                "timestamp" : 12345  
            },  
            "sequence" : {  
                "timestamp" : 12345  
            }  
        },  
        "reported" : {  
            "color" : {  
                "timestamp" : 12345  
            }  
        }  
    },  
    "version" : 10,  
    "clientToken" : "UniqueClientToken",  
    "timestamp": 123456789  
}
```

## Empty Sections

A shadow document contains a desired section only if it has a desired state. For example, the following is a valid state document with no desired section:

```
{  
    "reported" : { "temp": 55 }
```

```
}
```

The `reported` section can also be empty:

```
{
    "desired" : { "color" : "RED" }
}
```

If an update causes the `desired` or `reported` sections to become null, the section is removed from the document. To remove the `desired` section from a document (in response, for example, to a device updating its state), set the `desired` section to `null`:

```
{
    "state": {
        "reported": {
            "color": "red"
        },
        "desired": null
    }
}
```

It is also possible a shadow document will not contain `desired` or `reported` sections. In that case, the shadow document is empty. For example, this is a valid document:

```
{
}
```

## Arrays

Shadows support arrays, but treat them as normal values in that an update to an array replaces the whole array. It is not possible to update part of an array.

Initial state:

```
{
    "desired" : { "colors" : [ "RED", "GREEN", "BLUE" ] }
}
```

Update:

```
{
    "desired" : { "colors" : [ "RED" ] }
}
```

Final state:

```
{
    "desired" : { "colors" : [ "RED" ] }
}
```

Arrays can't have null values. For example, the following array is not valid and will be rejected.

```
{
    "desired" : {
        "colors" : [ null, "RED", "GREEN" ]
    }
}
```

```
}
```

## Using Shadows

AWS IoT provides three methods for working with a device's shadow:

### UPDATE

Creates a device's shadow if it doesn't exist, or updates the contents of a device's shadow with the data provided in the request. The data is stored with timestamp information to indicate when it was last updated. Messages are sent to all subscribers with the difference between desired or reported state (delta). Things or apps that receive a message can perform an action based on the difference between desired or reported states. For example, a device can update its state to the desired state, or an app can update its UI to show the change in the device's state.

### GET

Retrieves the latest state stored in the device's shadow (for example, during start-up of a device to retrieve configuration and the last state of operation). This method returns the full JSON document, including metadata.

### DELETE

Deletes a device's shadow, including all of its content. This removes the JSON document from the data store. You can't restore a device's shadow you deleted, but you can create a new shadow with the same name.

## Protocol Support

These methods are supported through both [MQTT](#) and a RESTful API over HTTPS. Because MQTT is a publish/subscribe communication model, AWS IoT implements a set of reserved topics. Things or applications subscribe to these topics before publishing on a request topic in order to implement a request-response behavior. For more information, see [Shadow MQTT Topics \(p. 343\)](#) and [Device Shadow RESTful API \(p. 340\)](#).

## Updating a Shadow

You can update a device's shadow by using the [UpdateThingShadow \(p. 341\)](#) RESTful API or by publishing to the [/update \(p. 343\)](#) topic. Updates affect only the fields specified in the request.

Initial state:

```
{
  "state": {
    "reported" : {
      "color" : { "r" :255, "g": 255, "b": 0 }
    }
  }
}
```

An update message is sent:

```
{
  "state": {
```

```

        "desired" : {
            "color" : { "r" : 10 },
            "engine" : "ON"
        }
    }
}

```

The device receives the desired state on the `/update/delta` topic that is triggered by the previous `/update` message and then executes the desired changes. When finished, the device should confirm its updated state through the `reported` section in the shadow JSON document.

Final state:

```
{
    "state": {
        "reported" : {
            "color" : { "r" : 10, "g" : 255, "b": 0 },
            "engine" : "ON"
        }
    }
}
```

## Retrieving a Shadow Document

You can retrieve a device's shadow by using the [GetThingShadow \(p. 341\)](#) RESTful API or by subscribing and publishing to the [/get \(p. 347\)](#) topic. This retrieves the entire document plus the delta between the `desired` or `reported` states.

Example document:

```
{
    "state": {
        "desired": {
            "lights": {
                "color": "RED"
            },
            "engine": "ON"
        },
        "reported": {
            "lights": {
                "color": "GREEN"
            },
            "engine": "ON"
        }
    },
    "metadata": {
        "desired": {
            "lights": {
                "color": {
                    "timestamp": 123456
                },
                "engine": {
                    "timestamp": 123456
                }
            }
        },
        "reported": {
            "lights": {
                "color": {
                    "timestamp": 789012
                }
            }
        }
    }
}
```

```
        },
        "engine": {
            "timestamp": 789012
        }
    },
    "version": 10,
    "timestamp": 123456789
}
}
```

Response:

```
{
    "state": {
        "desired": {
            "lights": {
                "color": "RED"
            },
            "engine": "ON"
        },
        "reported": {
            "lights": {
                "color": "GREEN"
            },
            "engine": "ON"
        },
        "delta": {
            "lights": {
                "color": "RED"
            }
        }
    },
    "metadata": {
        "desired": {
            "lights": {
                "color": {
                    "timestamp": 123456
                }
            },
            "engine": {
                "timestamp": 123456
            }
        },
        "reported": {
            "lights": {
                "color": {
                    "timestamp": 789012
                }
            },
            "engine": {
                "timestamp": 789012
            }
        },
        "delta": {
            "lights": {
                "color": {
                    "timestamp": 123456
                }
            }
        }
    },
    "version": 10,
    "timestamp": 123456789
}
```

## Optimistic Locking

You can use the state document version to ensure you are updating the most recent version of a device's shadow document. When you supply a version with an update request, the service rejects the request with an HTTP 409 conflict response code if the current version of the state document does not match the version supplied.

For example:

Initial document:

```
{  
    "state": {  
        "desired": { "colors": [ "RED", "GREEN", "BLUE" ] }  
    },  
    "version": 10  
}
```

Update: (version doesn't match; request will be rejected)

```
{  
    "state": {  
        "desired": {  
            "colors": [  
                "BLUE"  
            ]  
        }  
    },  
    "version": 9  
}
```

Result:

```
409 Conflict
```

Update: (version matches; this request will be accepted)

```
{  
    "state": {  
        "desired": {  
            "colors": [  
                "BLUE"  
            ]  
        }  
    },  
    "version": 10  
}
```

Final state:

```
{  
    "state": {  
        "desired": {  
            "colors": [  
                "BLUE"  
            ]  
        }  
    },  
    "version": 11  
}
```

```
}
```

## Deleting Data

You can delete data from a device's shadow by publishing to the [/update \(p. 343\)](#) topic, setting the fields to be deleted to null. Any field with a value of `null` is removed from the document.

Initial state:

```
{
  "state": {
    "desired" : {
      "lights": { "color": "RED" },
      "engine" : "ON"
    },
    "reported" : {
      "lights" : { "color": "GREEN" },
      "engine" : "OFF"
    }
  }
}
```

An update message is sent:

```
{
  "state": {
    "desired": null,
    "reported": {
      "engine": null
    }
  }
}
```

Final state:

```
{
  "state": {
    "reported" : {
      "lights" : { "color" : "GREEN" }
    }
  }
}
```

You can delete all data from a device's shadow by setting its state to `null`. For example, sending the following message deletes all of the state data, but the device's shadow remains.

```
{
  "state": null
}
```

The device's shadow still exists even if its state is `null`. The version of the shadow is incremented when the next update occurs.

## Deleting a Shadow

You can delete a device's shadow document by using the [DeleteThingShadow \(p. 342\)](#) RESTful API or by publishing to the [/delete \(p. 348\)](#) topic.

**Note**

Deleting a device's shadow does not delete the thing. Deleting a thing does not delete the corresponding device's shadow.

Initial state:

```
{  
  "state": {  
    "desired" : {  
      "lights": { "color": "RED" },  
      "engine" : "ON"  
    },  
    "reported" : {  
      "lights" : { "color": "GREEN" },  
      "engine" : "OFF"  
    }  
  }  
}
```

An empty message is published to the /delete topic.

Final state:

```
HTTP 404 - resource not found
```

## Delta State

Delta state is a virtual type of state that contains the difference between the desired and reported states. Fields in the desired section that are not in the reported section are included in the delta. Fields that are in the reported section and not in the desired section are not included in the delta. The delta contains metadata, and its values are equal to the metadata in the desired field. For example:

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

```

        "timestamp": 12345
    }
},
"delta": {
    "color": {
        "timestamp": 12345
    },
    "state": {
        "timestamp": 12345
    }
}
},
"version": 17,
"timestamp": 123456789
}
}

```

When nested objects differ, the delta contains the path all the way to the root.

```

{
    "state": {
        "desired": {
            "lights": {
                "color": {
                    "r": 255,
                    "g": 255,
                    "b": 255
                }
            }
        }
    },
    "reported": {
        "lights": {
            "color": {
                "r": 255,
                "g": 0,
                "b": 255
            }
        }
    },
    "delta": {
        "lights": {
            "color": {
                "g": 255
            }
        }
    }
},
"version": 18,
"timestamp": 123456789
}
}

```

The Device Shadow service calculates the delta by iterating through each field in the `desired` state and comparing it to the `reported` state.

Arrays are treated like values. If an array in the `desired` section doesn't match the array in the `reported` section, then the entire desired array is copied into the delta.

## Observing State Changes

When a device's shadow is updated, messages are published on two MQTT topics:

- `$aws/things/thing-name/shadow/update/accepted`

- \$aws/things/*thing-name*/shadow/update/delta

The message sent to the update/delta topic is intended for the thing whose state is being updated. This message contains only the difference between the desired and reported sections of the device's shadow document. Upon receiving this message, the device should decide whether to make the requested change. If the device's state is changed, it should publish its new current state to the \$aws/things/*thing-name*/shadow/update topic.

Devices and applications can subscribe to either of these topics to be notified when the state of the document has changed.

Here is an example of that flow:

1. A device reports its state.
2. The system updates the state document in its persistent data store.
3. The system publishes a delta message, which contains only the delta and is targeted at the subscribed devices. Devices should subscribe to this topic to receive updates.
4. The device's shadow publishes an accepted message, which contains the entire received document, including metadata. Applications should subscribe to this topic to receive updates.

## Message Order

There is no guarantee that messages from the AWS IoT service will arrive at the device in any specific order.

Initial state document:

```
{  
  "state" : {  
    "reported" : { "color" : "blue" }  
  },  
  "version" : 10,  
  "timestamp": 123456777  
}
```

Update 1:

```
{  
  "state": { "desired" : { "color" : "RED" } },  
  "version": 10,  
  "timestamp": 123456777  
}
```

Update 2:

```
{  
  "state": { "desired" : { "color" : "GREEN" } },  
  "version": 11,  
  "timestamp": 123456778  
}
```

Final state document:

```
{  
  "state": {  
    "reported": { "color" : "GREEN" }  
  }  
}
```

```
{
    },
    "version": 12,
    "timestamp": 123456779
}
```

This results in two delta messages:

```
{
    "state": {
        "color": "RED"
    },
    "version": 11,
    "timestamp": 123456778
}
```

```
{
    "state": { "color" : "GREEN" },
    "version": 12,
    "timestamp": 123456779
}
```

The device might receive these messages out of order. Because the state in these messages is cumulative, a device can safely discard any messages that contain a version number older than the one it is tracking. If the device receives the delta for version 12 before version 11, it can safely discard the version 11 message.

## Trim Shadow Messages

To reduce the size of shadow messages sent to your device, define a rule that selects only the fields your device needs then republishes the message on an MQTT topic to which your device is listening.

The rule is specified in JSON and should look like the following:

```
{
    "sql": "SELECT state, version FROM '$aws/things/+/_shadow/update/delta'",
    "ruleDisabled": false,
    "actions": [
        {
            "republish": {
                "topic": "${topic(2)}/delta",
                "roleArn": "arn:aws:iam::123456789012:role/my-iot-role"
            }
        }
    ]
}
```

The SELECT statement determines which fields from the message will be republished to the specified topic. A "+" wild card is used to match all shadow names. The rule specifies that all matching messages should be republished to the specified topic. In this case, the "topic()" function is used to specify the topic on which to republish. topic(2) evaluates to the thing name in the original topic. For more information about creating rules, see [Rules](#).

## Device Shadow RESTful API

A shadow exposes the following URI for updating state information:

```
https://endpoint/things/thingName/shadow
```

The endpoint is specific to your AWS account. To retrieve your endpoint, use the [describe-endpoint](#) command. The format of the endpoint is as follows:

```
identifier.iot.region.amazonaws.com
```

The shadow RESTful API follows the same HTTPS protocols/port mappings as described in [AWS IoT Protocols](#).

#### API Actions

- [GetThingShadow \(p. 341\)](#)
- [UpdateThingShadow \(p. 341\)](#)
- [DeleteThingShadow \(p. 342\)](#)

## GetThingShadow

Gets the shadow for the specified thing.

The response state document includes the delta between the desired and reported states.

#### Request

The request includes the standard HTTP headers plus the following URI:

```
HTTP GET https://endpoint/things/thingName/shadow
```

#### Response

Upon success, the response includes the standard HTTP headers plus the following code and body:

```
HTTP 200
BODY: response state document
```

For more information, see [Example Response State Document \(p. 350\)](#).

#### Authorization

Retrieving a shadow requires a policy that allows the caller to perform the `iot:GetThingShadow` action. The Device Shadow service accepts two forms of authentication: Signature Version 4 with IAM credentials or TLS mutual authentication with a client certificate.

The following is an example policy that allows a caller to retrieve a device's shadow:

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

## UpdateThingShadow

Updates the shadow for the specified thing.

Updates affect only the fields specified in the request state document. Any field with a value of `null` is removed from the device's shadow.

### Request

The request includes the standard HTTP headers plus the following URI and body:

```
HTTP POST https://endpoint/things/thingName/shadow
BODY: request state document
```

For more information, see [Example Request State Document \(p. 350\)](#).

### Response

Upon success, the response includes the standard HTTP headers plus the following code and body:

```
HTTP 200
BODY: response state document
```

For more information, see [Example Response State Document \(p. 350\)](#).

### Authorization

Updating a shadow requires a policy that allows the caller to perform the `iot:UpdateThingShadow` action. The Device Shadow service accepts two forms of authentication: Signature Version 4 with IAM credentials or TLS mutual authentication with a client certificate.

The following is an example policy that allows a caller to update a device's shadow:

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

## DeleteThingShadow

Deletes the shadow for the specified thing.

### Request

The request includes the standard HTTP headers plus the following URI:

```
HTTP DELETE https://endpoint/things/thingName/shadow
```

### Response

Upon success, the response includes the standard HTTP headers plus the following code and body:

```
HTTP 200
BODY: Empty response state document
```

### Authorization

Deleting a device's shadow requires a policy that allows the caller to perform the `iot:DeleteThingShadow` action. The Device Shadow service accepts two forms of authentication: Signature Version 4 with IAM credentials or TLS mutual authentication with a client certificate.

The following is an example policy that allows a caller to delete a device's shadow:

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

## Shadow MQTT Topics

The Device Shadow service uses reserved MQTT topics to enable applications and devices to get, update, or delete the state information for a device (shadow). The names of these topics start with `$aws/things/thingName/shadow`. Publishing and subscribing on shadow topics requires topic-based authorization. AWS IoT reserves the right to add new topics to the existing topic structure. For this reason, we recommend that you avoid wild card subscriptions to shadow topics. For example, avoid subscribing to topic filters like `$aws/things/thingName/shadow/#` because the number of topics that match this topic filter might increase as AWS IoT introduces new shadow topics. For examples of the messages published on these topics see [Device Shadow Service Data Flow \(p. 321\)](#).

The following are the MQTT topics used for interacting with shadows.

### Topics

- [/update \(p. 343\)](#)
- [/update/accepted \(p. 344\)](#)
- [/update/documents \(p. 345\)](#)
- [/update/rejected \(p. 345\)](#)
- [/update/delta \(p. 346\)](#)
- [/get \(p. 347\)](#)
- [/get/accepted \(p. 347\)](#)
- [/get/rejected \(p. 348\)](#)
- [/delete \(p. 348\)](#)
- [/delete/accepted \(p. 349\)](#)
- [/delete/rejected \(p. 349\)](#)

## /update

Publish a request state document to this topic to update the device's shadow:

```
$aws/things/thingName/shadow/update
```

A client attempting to update the state of a thing would send a JSON request state document like this:

```
{  
    "state" : {
```

```
        "desired" : {
            "color" : "red",
            "power" : "on"
        }
    }
}
```

A device updating its shadow would send a JSON request state document like this:

```
{
    "state" : {
        "reported" : {
            "color" : "red",
            "power" : "on"
        }
    }
}
```

AWS IoT responds by publishing to either [/update/accepted \(p. 344\)](#) or [/update/rejected \(p. 345\)](#).

For more information, see [Request State Documents \(p. 350\)](#).

## Example Policy

The following is an example of the required policy:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": ["iot:Publish"],
            "Resource": ["arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/update"]
        }
    ]
}
```

## /update/accepted

AWS IoT publishes a response state document to this topic when it accepts a change for the device's shadow:

```
$aws/things/thingName/shadow/update/accepted
```

For more information, see [Response State Documents \(p. 350\)](#).

## Example Policy

The following is an example of the required policy:

```
{
    "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 publishes a state document to this topic whenever an update to the shadow is successfully performed:

```
$aws/things/thingName/shadow/update/documents
```

The JSON document will contain two primary nodes: `previous` and `current`. The `previous` node will contain the contents of the full shadow document before the update was performed while `current` will contain the full shadow document after the update is successfully applied. When the shadow is updated (created) for the first time, the `previous` node will contain `null`.

## Example Policy

The following is an example of the required policy:

```
{
    "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/accepted"]
        }
    ]
}
```

## /update/rejected

AWS IoT publishes an error response document to this topic when it rejects a change for the device's shadow:

```
$aws/things/thingName/shadow/update/rejected
```

For more information, see [Error Response Documents \(p. 351\)](#).

## Example Policy

The following is an example of the required policy:

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

## /update/delta

AWS IoT publishes a response state document to this topic when it accepts a change for the device's shadow and the request state document contains different values for desired and reported states:

```
$aws/things/thingName/shadow/update/delta
```

For more information, see [Response State Documents \(p. 350\)](#).

## Publishing Details

- A message published on `update/delta` includes only the desired attributes that differ between the `desired` and `reported` sections. It contains all of these attributes, regardless of whether these attributes were contained in the current update message or were already stored in AWS IoT. Attributes that do not differ between the `desired` and `reported` sections are not included.
- If an attribute is in the `reported` section but has no equivalent in the `desired` section, it is not included.
- If an attribute is in the `desired` section but has no equivalent in the `reported` section, it is included.
- If an attribute is deleted from the `reported` section but still exists in the `desired` section, it is included.

## Example Policy

The following is an example of the required policy:

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

```
        ]
    }
```

## /get

Publish an empty message to this topic to get the device's shadow:

```
$aws/things/thingName/shadow/get
```

AWS IoT responds by publishing to either [/get/accepted \(p. 347\)](#) or [/get/rejected \(p. 348\)](#).

## Example Policy

The following is an example of the required policy:

```
{
    "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 publishes a response state document to this topic when returning the device's shadow:

```
$aws/things/thingName/shadow/get/accepted
```

For more information, see [Response State Documents \(p. 350\)](#).

## Example Policy

The following is an example of the required policy:

```
{
    "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 publishes an error response document to this topic when it can't return the device's shadow:

```
$aws/things/thingName/shadow/get/rejected
```

For more information, see [Error Response Documents \(p. 351\)](#).

## Example Policy

The following is an example of the required policy:

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

## /delete

To delete a device's shadow, publish an empty message to the delete topic:

```
$aws/things/thingName/shadow/delete
```

The content of the message is ignored.

AWS IoT responds by publishing to either [/delete/accepted \(p. 349\)](#) or [/delete/rejected \(p. 349\)](#).

## Example Policy

The following is an example of the required policy:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": ["iot:Subscribe"],  
            "Resource": ["arn:aws:iot:region:account:topicfilter/$aws/things/thingName/shadow/delete"]  
        },  
        {  
            "Effect": "Allow",  
            "Action": ["iot:Receive"],  
            "Resource": ["arn:aws:iot:region:account:topic/$aws/things/thingName/shadow/delete"]  
        }  
    ]  
}
```

```
    ]  
}
```

## /delete/accepted

AWS IoT publishes a message to this topic when a device's shadow is deleted:

```
$aws/things/thingName/shadow/delete/accepted
```

### Example Policy

The following is an example of the required policy:

```
{  
    "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 publishes an error response document to this topic when it can't delete the device's shadow:

```
$aws/things/thingName/shadow/delete/rejected
```

For more information, see [Error Response Documents \(p. 351\)](#).

### Example Policy

The following is an example of the required policy:

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

```
    ]  
}
```

## Shadow Document Syntax

The Device Shadow service uses the following documents in UPDATE, GET, and DELETE operations using the [RESTful API \(p. 340\)](#) or [MQTT Pub/Sub Messages \(p. 343\)](#). For more information, see [Device Shadow Service Documents \(p. 328\)](#).

### Examples

- [Request State Documents \(p. 350\)](#)
- [Response State Documents \(p. 350\)](#)
- [Error Response Documents \(p. 351\)](#)

## Request State Documents

Request state documents have the following format:

```
{  
  "state": {  
    "desired": {  
      "attribute1": "integer2",  
      "attribute2": "string2",  
      ...  
      "attributeN": "boolean2"  
    },  
    "reported": {  
      "attribute1": "integer1",  
      "attribute2": "string1",  
      ...  
      "attributeN": "boolean1"  
    }  
  },  
  "clientToken": "token",  
  "version": version  
}
```

- **state** — Updates affect only the fields specified.
- **clientToken** — If used, you can verify that the request and response contain the same client token.
- **version** — If used, the Device Shadow service processes the update only if the specified version matches the latest version it has.

## Response State Documents

Response state documents have the following format:

```
{  
  "state": {  
    "desired": {  
      "attribute1": "integer2",  
      "attribute2": "string2",  
      ...  
      "attributeN": "boolean2"
```

```

},
"reported": {
    "attribute1": integer1,
    "attribute2": "string1",
    ...
    "attributeN": boolean1
},
"delta": {
    "attribute3": integerX,
    "attribute5": "stringY"
}
},
"metadata": {
    "desired": {
        "attribute1": {
            "timestamp": timestamp
        },
        "attribute2": {
            "timestamp": timestamp
        },
        ...
        "attributeN": {
            "timestamp": timestamp
        }
    },
    "reported": {
        "attribute1": {
            "timestamp": timestamp
        },
        "attribute2": {
            "timestamp": timestamp
        },
        ...
        "attributeN": {
            "timestamp": timestamp
        }
    }
},
"timestamp": timestamp,
"clientToken": "token",
"version": version
}
}

```

- **state**
  - **reported** — Only present if a thing reported any data in the `reported` section and contains only fields that were in the request state document.
  - **desired** — Only present if a thing reported any data in the `desired` section and contains only fields that were in the request state document.
- **metadata** — Contains the timestamps for each attribute in the `desired` and `reported` sections so that you can determine when the state was updated.
- **timestamp** — The Epoch date and time the response was generated by AWS IoT.
- **clientToken** — Present only if a client token was used when publishing valid JSON to the `/update` topic.
- **version** — The current version of the document for the device's shadow shared in AWS IoT. It is increased by one over the previous version of the document.

## Error Response Documents

Error response documents have the following format:

```
{
  "code": error-code,
  "message": "error-message",
  "timestamp": timestamp,
  "clientToken": "token"
}
```

- **code** — An HTTP response code that indicates the type of error.
- **message** — A text message that provides additional information.
- **timestamp** — The date and time the response was generated by AWS IoT.
- **clientToken** — Present only if a client token was used when publishing valid JSON to the /update topic.

For more information, see [Shadow Error Messages \(p. 352\)](#).

## Shadow Error Messages

The Device Shadow service publishes a message on the error topic (over MQTT) when an attempt to change the state document fails. This message is only emitted as a response to a publish request on one of the reserved \$aws topics. If the client updates the document using the REST API, then it receives the HTTP error code as part of its response, and no MQTT error messages are emitted.

HTTP Error Code	Error Messages
400 (Bad Request)	<ul style="list-style-type: none"> <li>• Invalid JSON</li> <li>• Missing required node: state</li> <li>• State node must be an object</li> <li>• Desired node must be an object</li> <li>• Reported node must be an object</li> <li>• Invalid version</li> <li>• Invalid clientToken</li> </ul> <p><b>Note</b> A client token that is longer than 64 bytes will cause this response.</p> <ul style="list-style-type: none"> <li>• JSON contains too many levels of nesting; maximum is 6</li> <li>• State contains an invalid node</li> </ul>
401 (Unauthorized)	<ul style="list-style-type: none"> <li>• Unauthorized</li> </ul>
403 (Forbidden)	<ul style="list-style-type: none"> <li>• Forbidden</li> </ul>
404 (Not Found)	<ul style="list-style-type: none"> <li>• Thing not found</li> </ul>
409 (Conflict)	<ul style="list-style-type: none"> <li>• Version conflict</li> </ul>
413 (Payload Too Large)	<ul style="list-style-type: none"> <li>• The payload exceeds the maximum size allowed</li> </ul>
415 (Unsupported Media Type)	<ul style="list-style-type: none"> <li>• Unsupported documented encoding; supported encoding is UTF-8</li> </ul>
429 (Too Many Requests)	<ul style="list-style-type: none"> <li>• The Device Shadow service will generate this error message when there are more than 10 in-flight requests.</li> </ul>

HTTP Error Code	Error Messages
500 (Internal Server Error)	<ul style="list-style-type: none"><li>• Internal service failure</li></ul>

# Jobs

AWS IoT jobs can be used to define a set of remote operations that are sent to and executed on one or more devices connected to AWS IoT.

## Jobs Key Concepts

### job

A job is a remote operation that is sent to and executed on one or more devices connected to AWS IoT. For example, you can define a job that instructs a set of devices to download and install application or firmware updates, reboot, rotate certificates, or perform remote troubleshooting operations.

### job document

To create a job, you must first create a job document that is a description of the remote operations to be performed by the devices.

Job documents are UTF-8 encoded JSON documents and should contain any information your devices need to perform a job. A job document will most likely contain one or more URLs where the device can download an update or some other data. The job document itself can be stored in an Amazon S3 bucket, or be included inline with the command that creates the job.

### target

When you create a job, you specify a list of targets that are the devices that should perform the operations. The targets can be things or [thing groups \(p. 160\)](#) or both. AWS IoT Jobs sends a message to each target to inform it that a job is available.

### job execution

A job execution is an instance of a job on a target device. The target starts an execution of a job by downloading the job document. It then performs the operations the document specifies, and reports its progress to AWS IoT. An execution number is a unique identifier of a specific job execution on a specific target. The Jobs service provides commands to track the progress of a job execution on a specific target and the progress of a job generally across all the targets of the job.

### snapshot job

By default, a job is sent to all targets that you specify when you create the job. After those targets complete the job (or report that they are unable to do so), the job is complete.

### continuous job

A continuous job is sent to all targets that you specify when you create the job, but continues to run and will be sent to any new devices (things) that are added to the target group. For example, a continuous job can be used to onboard or upgrade devices as they are added to a group. You can make a job continuous by setting an optional parameter when you create the job.

### rollouts

You can specify how quickly targets are notified of a pending job execution. This allows you to create a staged rollout to better manage updates, reboots, and other operations.

The following field can be added to the `CreateJob` request to specify the maximum number of targets that will be informed of the job per minute. This example sets a static rollout rate.

```
"jobExecutionRolloutConfig": {  
    "maximumPerMinute": "integer"  
}
```

You can also set a variable rollout rate with the `exponentialRate` field. The following example creates a rollout that has an exponential rate.

```
"jobExecutionsRolloutConfig": {  
    "exponentialRate": {  
        "baseRatePerMinute": integer,  
        "incrementFactor": integer,  
        "rateIncreaseCriteria": {  
            "numberOfNotifiedThings": integer, // Set one or the other  
            "numberOfSucceededThings": integer // of these two values.  
        },  
        "maximumPerMinute": integer  
    }  
}
```

For more information about configuring job rollouts, see [Job Rollout and Abort Configuration](#).

#### abort

You can create a set of conditions to abort rollouts once certain criteria that you specify have been met. For more information about configuring job abort conditions, see [Job Rollout and Abort Configuration](#).

#### presigned URLs

To allow a device secure, time-limited access to data beyond that included in the job document itself, you can use presigned Amazon S3 URLs. You can place your data in an Amazon S3 bucket and add a placeholder link to the data in the job document. When the Jobs service receives a request for the job document, it parses the job document looking for placeholder links and it replaces them with presigned Amazon S3 URLs.

The placeholder link is of the following form:

```
 ${aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/key}
```

where `bucket` is your bucket name and `key` is the object in the bucket to which you are linking.

#### timeouts

##### Note

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

Job timeouts enable you to be notified whenever a job execution gets stuck in the `IN_PROGRESS` state for an unexpectedly long period of time. There are two types of timers: in progress timers and step timers.

When you create a job, you can set a value for the `inProgressTimeoutInMinutes` property of the optional `TimeoutConfig` object. The in progress timer can't be updated and will apply to all job executions for the job. Whenever a job execution remains in the `IN_PROGRESS` status for longer

than this interval, the job execution will fail and switch to the terminal `TIMED_OUT` status. AWS IoT will also publish an MQTT notification.

You can also set a step timer for a specific job execution by setting a value for `stepTimeoutInMinutes` when you call [UpdateJobExecution](#). The step timer applies only to the job execution that you update, and you can set a new value for this timer each time you update a job execution. You can also create a step timer when you call [StartNextPendingJobExecution](#). If the job execution remains in the `IN_PROGRESS` status for longer than the step timer interval, it will fail and switch to the terminal `TIMED_OUT` status. The step timer has no effect on the in progress timer that you set when you create a job.

The following diagram and description illustrate the ways in which in progress timeouts and step timeouts interact with each other.

**Job Creation:** `CreateJob` sets an in progress timer that expires in twenty minutes. This timer applies to all job executions and can't be updated.

**12:00 PM:** The job execution starts and switches to `IN_PROGRESS` status. The in progress timer starts to run.

**12:05 PM:** `UpdateJobExecution` creates a step timer with a value of 7 minutes. If a new step timer isn't created, the job execution will time out at 12:12 PM.

**12:10 PM:** `UpdateJobExecution` creates a new step timer with a value of 5 minutes. The previous step timer is discarded. If a new step timer isn't created, the job execution will time out at 12:15 PM.

**12:13 PM:** `UpdateJobExecution` creates a new step timer with a value of 9 minutes. The job execution times out at 12:20 because the in progress timer expires at 12:20. The step timer can't exceed the absolute bound created by the in progress timer.

`UpdateJobExecution` can also discard a step timer that has already been created by creating a new step timer with a value of -1.

## Managing Jobs

You can use the [AWS IoT console](#), the Jobs HTTPS API, the AWS Command Line Interface, or the AWS SDKs to create and manage jobs. For more information, see [Job Management and Control API \(p. 378\)](#), [AWS CLI Command Reference: iot](#) or [AWS SDKs and Tools](#).

The primary purpose of jobs is to notify devices of a software or firmware update. When sending code to devices, the best practice is to sign the code file. This allows devices to detect if the code has been modified in transit. The instructions in the following section are written with the assumption that you want to code-sign the software update you are sending to your devices.

For more information about code-signing, see [What Is Code Signing for AWS IoT?](#)

Before you create a job, you must create a job document. If you are using Code-signing for AWS IoT, you must upload your job document to a versioned Amazon S3 bucket. For more information about creating an Amazon S3 bucket and uploading files to it, see [Getting Started with Amazon Simple Storage Service](#) in the [Amazon S3 Getting Started Guide](#).

Your job document can contain a presigned Amazon S3 URL that points to your code file (or other file). Presigned Amazon S3 URLs are valid for a limited amount of time and so are not generated until a device requests a job document. Because the presigned URL has not been created when you are creating the job document, you put a placeholder URL in your job document instead.

A placeholder URL looks like the following: \${aws:iot:s3-presigned-url:https://s3.region.amazonaws.com/<bucket>/<code file>} where **bucket** is the Amazon S3 bucket that contains the code file and **code file** is the Amazon S3 key of the code file.

When a device requests the job document, AWS IoT generates the presigned URL and replaces the placeholder URL with the presigned URL. Your job document is then sent to the device.

When you create a job that uses presigned Amazon S3 URLs, you must provide an IAM role that grants permission to download files from the Amazon S3 bucket where the data or updates are stored. The role must also grant permission for AWS IoT to assume the role.

You can specify an optional timeout for the presigned URL. For more information, see [CreateJob \(p. 393\)](#).

### To grant Jobs permission to assume your role

1. Sign in to the AWS Management Console and open the IAM console at <https://console.aws.amazon.com/iam/>.
2. In the navigation pane, choose **Roles**.
3. Search for your role and choose it.
4. Choose the **Trust Relationships** tab, and then choose **Edit Trust Relationship**.
5. On the **Edit Trust Relationship** page, replace the policy document with the following JSON:

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

6. Choose **Update Trust Policy**.
7. If your job uses a job document that is an Amazon S3 object, choose **Permissions** and with the following JSON, add a policy that grants permission to download files from your Amazon S3 bucket:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::your_S3_bucket/*"  
        }  
    ]  
}
```

## Creating and Managing Jobs (Console)

If you are using Code-signing for AWS IoT, you must add two placeholder URLs in your job document:

A placeholder for the code file should look like this: \${aws:iot:s3-presigned-url:https://s3.amazonaws.com/<my-s3-bucket>/<my-code-file>}.

**Note**

Currently, versions are not supported for presigned URL placeholders for jobs. If you update your code file and copy it to the same Amazon S3 location, you must create a new signature and then reference the new signature version in your job document.

A placeholder for the signature should look like this: \${aws:iot:code-sign-signature:s3://<region>.<my-s3-bucket>/<my-code-file>@<code-file-version-id>}.

**To create a job**

1. Browse to the [AWS IoT console](#).
2. In the navigation pane, choose **Manage**, and then choose **Jobs**.
3. Choose **Create a job**.
4. Choose **Create a custom job**.
5. Enter an alphanumeric ID for your job and an optional description.

**Note**

We do not recommend using personally identifiable information in your job IDs or descriptions.

6. Select the device or device groups that you want to update.
7. Under **Add a job file**, choose **Select**, and then select your job document.
8. Choose **Sign image for me**. If you are not code-signing your update, you can skip this step.
9. Create or choose a code-signing profile. If you are not code-signing your update, you can skip this step.
10. Under **Pre-sign resource URLs**, choose **I want to pre-sign my URLs and have configured my job file**. If you are not code-signing your update, you can skip this step.
11. Choose a role and an expiry time for the presigned URL.
12. Under **Job type**, choose the appropriate option for your update, and then choose **Next**.
13. Specify values for any advanced configurations, and then choose **Create**.

After you create the job, the console generates a JSON signature and places it in your job document.

You can use the [AWS IoT console](#) to view the status, cancel, or delete a job.

1. Browse to the [AWS IoT console](#).
2. In the navigation pane, choose **Manage**, and then choose **Jobs**.

## Creating and Managing Jobs (CLI)

This section describes how to create and manage jobs.

### Create Jobs

You use the **CreateJob** command to create an AWS IoT job. The job is queued for execution on the targets (things or thing groups) that you specify. To create an AWS IoT job, you need a job document that can be included in the body of the request or as a link to an Amazon S3 document. If the job includes downloading files using presigned Amazon S3 URLs, you need an IAM role ARN that has permission to download the file and grants permission to AWS IoT Jobs to assume the role.

## Code-signing with Jobs

If you are using Code-signing for AWS IoT, you must start a code-signing job and include the output in your job document. Use the [start-signing-job](#) command to create a code-signing job. `start-signing-job` returns a job ID. Use the [describe-signing-job](#) command to get the Amazon S3 location where the signature is stored. You can then download the signature from Amazon S3. For more information about code-signing jobs, see [Code-signing for AWS IoT](#).

Your job document must contain a pre-signed URL placeholder for your code file and the JSON signature output placed in an Amazon S3 bucket using the `start-signing-job` command, enclosed in a `codesign` element:

```
{
    "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>
            }
        }
    }
}
```

## Creating a Job with a Job Document

The following command shows how to create a job using a job document (`job-document.json`) stored in an Amazon S3 bucket (`jobBucket`) and a role with permission to download files from Amazon S3 (`S3DownloadRole`).

```
aws iot create-job \
    --job-id 010 \
    --targets arn:aws:iot:us-east-1:123456789012:thing/thingOne \
    --document-source https://s3.amazonaws.com/my-s3-bucket/job-document.json \
    --timeout-config inProgressTimeoutInMinutes=100 \
    --job-executions-rollout-config "{\"exponentialRate\": { \"baseRatePerMinute\": 50, \
    \"incrementFactor\": 2, \"rateIncreaseCriteria\": { \"numberOfNotifiedThings\": 1000, \
    \"numberOfSucceededThings\": 1000}, \"maximumPerMinute\": 1000}}\" \
    --abort-config "{ \"criteriaList\": [ { \"action\": \"CANCEL\", \"failureType\": \
    \"FAILED\", \"minNumberOfExecutedThings\": 100, \"thresholdPercentage\": 20}, { \"action\": \
    \"CANCEL\", \"failureType\": \"TIMED_OUT\", \"minNumberOfExecutedThings\": 200, \
    \"thresholdPercentage\": 50}]}\" \
    --presigned-url-config "{\"roleArn\":\"arn:aws:iam::123456789012:role/ \
    S3DownloadRole\", \"expiresInSec\":3600}"
```

The job is executed on `thingOne`.

The optional `timeout-config` parameter specifies the amount of time each device has to finish its execution of the job. The timer starts when the job execution status is set to `IN_PROGRESS`. If the job execution status is not set to another terminal state before the time expires, it is set to `TIMED_OUT`.

### Note

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

The in-progress timer can't be updated and applies to all job executions for the job. Whenever a job execution remains in the `IN_PROGRESS` status for longer than this interval, the job execution fails and switches to the terminal `TIMED_OUT` status. AWS IoT also publishes an MQTT notification.

For more information about creating configurations about job rollouts and aborts, see [Job Rollout and Abort Configuration](#).

**Note**

Job documents that are specified as Amazon S3 files are retrieved at the time you create the job. Changing the contents of the Amazon S3 file you used as the source of your job document after you have created the job does not change what is sent to the targets of the job.

## Update a Job

You use the **UpdateJob** command to update a job. You can update the `description`, `presignedUrlConfig`, `jobExecutionsRolloutConfig`, `abortConfig`, and `timeoutConfig` fields of a job.

```
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}"
```

For more information about configuring job rollouts and aborts, see [Job Rollout and Abort Configuration](#).

## Cancel a Job

You use the **CancelJob** command to cancel a job. Canceling a job stops AWS IoT from rolling out any new job executions for the job. It also cancels any job executions that are in a `QUEUED` state. AWS IoT leaves any job executions in a terminal state untouched because the device has already completed the job. If the status of a job execution is `IN_PROGRESS`, it also remains untouched unless you use the optional `--force` parameter.

The following command shows how to cancel a job with ID 010.

```
aws iot cancel-job --job-id 010
```

The command displays the following output:

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

When you cancel a job, job executions that are in a `QUEUED` state are canceled. Job executions that are in an `IN_PROGRESS` state are canceled if you specify the optional `--force` parameter. Job executions in a terminal state are not canceled.

**Warning**

Canceling a job that is in the `IN_PROGRESS` state (by setting the `--force` parameter) cancels any job executions that are in progress and causes the device that is executing the job to

be unable to update the job execution status. Use caution and make sure that each device executing a canceled job is able to recover to a valid state.

The status of a canceled job or of one of its job executions is eventually consistent. AWS IoT stops scheduling new job executions and `QUEUED` job executions for that job to devices as soon as possible. Changing the status of a job execution to `CANCELED` might take some time, depending on the number of devices and other factors.

If a job is cancelled because it has met the criteria defined by an `AbortConfig` object, the service will add auto-populated values for the `comment` and `reasonCode` fields. Customers can create their own values for `reasonCode` when the job cancellation is user-driven.

## Cancel a Job Execution

You use the **CancelJobExecution** command to cancel a job execution on a device. It cancels a job execution that is in a `QUEUED` state. If you want to cancel a job execution that is in progress, you must use the `--force` parameter.

The following command shows how to cancel the job execution from job 010 running on `myThing`.

```
aws iot cancel-job-execution --job-id 010 --thing-name myThing
```

The command displays no output.

A job execution that is in a `QUEUED` state is canceled. A job execution that is in an `IN_PROGRESS` state is canceled if you specify the optional `--force` parameter. Job executions in a terminal state cannot be canceled.

### Warning

When you cancel a job execution that is in the `IN_PROGRESS` state, the device is unable to update the job execution status. Use caution and ensure that the device is able to recover to a valid state.

If the job execution is in a terminal state or if the job execution is in an `IN_PROGRESS` state and the `--force` parameter is not set to `true`, this command causes an `InvalidStateException`.

The status of a canceled job execution is eventually consistent. Changing the status of a job execution to `CANCELED` might take some time, depending on various factors.

## Delete a Job

You use the **DeleteJob** command to delete a job and its job executions. By default, you can only delete a job that is in a terminal state (`SUCCEEDED` or `CANCELED`). Otherwise, an exception occurs. You can delete a job in the `IN_PROGRESS` state if the `force` parameter is set to `true`.

Run the following command to delete a job:

```
aws iot delete-job --job-id 010 --force|--no-force
```

The command displays no output.

### Warning

When you delete a job that is in the `IN_PROGRESS` state, the device that is executing the job cannot access job information or update the job execution status. Use caution and make sure that each device executing a job that has been deleted is able to recover to a valid state.

It can take some time to delete a job, depending on the number of job executions created for the job and various other factors. While the job is being deleted, `DELETION_IN_PROGRESS` appears as

the status of the job. An error results if you attempt to delete or cancel a job whose status is already `DELETION_IN_PROGRESS`.

Only 10 jobs can have a status of `DELETION_IN_PROGRESS` at the same time. Otherwise, a `LimitExceededException` occurs.

## Get a Job Document

You use the **GetJobDocument** command to retrieve a job document for a job. A job document is a description of the remote operations to be performed by the devices.

Run the following command to get a job document:

```
aws iot get-job-document --job-id 010
```

The command returns the job document for the specified job:

```
{  
    "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

When you use this command to retrieve a job document, placeholder URLs are not replaced by presigned Amazon S3 URLs. When a device calls the [GetPendingJobExecutions \(p. 437\)](#) MQTT API, the placeholder URLs are replaced by presigned Amazon S3 URLs in the job document.

## List Jobs

You use the **ListJobs** command to get a list of all jobs in your AWS account. Job data and job execution data are purged after 90 days. Run the following command to list all jobs in your AWS account:

```
aws iot list-jobs
```

The command returns all jobs in your account, sorted by the job status:

```
{  
    "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": "PENDING",  
            "lastUpdatedAt": 1486685868.444,  
            "jobArn": "arn:aws:iot:us-east-1:123456789012:job/011",  
            "createdAt": 1486685868.444,  
            "targetSelection": "SNAPSHOT",  
            "jobId": "011"  
        }  
    ]  
}
```

```

        "status": "CANCELED",
        "lastUpdatedAt": 1486678850.575,
        "jobArn": "arn:aws:iot:us-east-1:123456789012:job/011",
        "createdAt": 1486678850.575,
        "targetSelection": "SNAPSHOT",
        "jobId": "011"
    }
}

```

## Describe a Job

Run the **DescribeJob** command to get the status of a specific job. The following command shows how to describe a job:

```
$ aws iot describe-job --job-id 010
```

The command returns the status of the specified job. For example:

```
{
    "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
    }
}
}

```

## List Executions for a Job

A job running on a specific device is represented by a job execution object. Run the **ListJobExecutionsForJob** command to list all job executions for a job. The following shows how to list the executions for a job:

```
aws iot list-job-executions-for-job --job-id 010
```

The command returns a list of job executions:

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

## List Job Executions for a Thing

Run the **ListJobExecutionsForThing** command to list all job executions running on a thing. The following shows how to list job executions for a thing:

```
aws iot list-job-executions-for-thing --thing-name thingOne
```

The command returns a list of job executions that are running or have run on the specified thing:

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

## Describe Job Execution

Run the **DescribeJobExecution** command to get the status of a job execution. You must specify a job ID and thing name and, optionally, an execution number to identify the job execution. The following shows how to describe a job execution:

```
aws iot describe-job-execution --job-id 017 --thing-name thingOne
```

The command returns the [JobExecution \(p. 383\)](#). For example:

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

## Delete Job Execution

Run the **DeleteJobExecution** command to delete a job execution. You must specify a job ID and a thing name and, optionally, an execution number to identify the job execution. The following shows how to delete a job execution:

```
aws iot delete-job-execution --job-id 017 --thing-name thingOne --execution-number
1234567890 --force|--no-force
```

The command displays no output.

By default, the status of the job execution must be `QUEUED` or in a terminal state (`SUCCEEDED`, `FAILED`, `REJECTED`, `TIMED_OUT`, `REMOVED` or `CANCELED`). Otherwise, an error occurs. To delete a job execution with a status of `IN_PROGRESS`, you can set the `force` parameter to `true`.

### Warning

When you delete a job execution with a status of `IN_PROGRESS`, the device that is executing the job cannot access job information or update the job execution status. Use caution and make sure that the device is able to recover to a valid state.

## Devices and Jobs

### device communication with Jobs

Devices can communicate with the AWS IoT Jobs service through these methods:

- MQTT
- HTTP Signature Version 4
- HTTP TLS

### using the MQTT protocol

Communication between the AWS IoT Jobs service and your devices can occur over the MQTT protocol. Devices subscribe to MQTT topics to be notified of new jobs and to receive responses from the AWS IoT Jobs service. Devices publish on MQTT topics to query or update the state of a job execution. Each device has its own general MQTT topic. For more information about publishing and subscribing to MQTT topics, see [Message Broker for AWS IoT \(p. 233\)](#).

### Note

You must use the correct endpoint when you communicate with the AWS IoT Jobs service through MQTT. Use the **DescribeEndpoint** command to find it. For example, if you run this command:

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

you get a result similar to the following:

```
{  
    "endpointAddress": "a1b2c3d4e5f6g7.iot.us-west-2.amazonaws.com"  
}
```

With this method, your device uses its device-specific certificate and private key to authenticate with the AWS IoT Jobs service.

Devices can:

- Be notified when a job execution is added or removed from the list of pending job executions by subscribing to the `$aws/things/thing-name/jobs/notify` MQTT topic, where *thing-name* is the name of the thing associated with the device.
- Be notified when the next pending job execution has changed by subscribing to the `$aws/things/thing-name/jobs/notify-next` MQTT topic, where *thing-name* is the name of the thing associated with the device.
- Update the status of a job execution by calling the [UpdateJobExecution \(p. 451\)](#) API.
- Query the status of a job execution by calling the [DescribeJobExecution \(p. 446\)](#) API.
- Retrieve a list of pending job executions by calling the [GetPendingJobExecutions \(p. 437\)](#) API.
- Retrieve the next pending job execution by calling the [DescribeJobExecution \(p. 446\)](#) API with jobId `$next`.
- Get and start the next pending job execution by calling the [StartNextPendingJobExecution \(p. 441\)](#) API.

The AWS IoT Jobs service publishes success and failure messages on an MQTT topic. The topic is formed by appending `accepted` or `rejected` to the topic used to make the request. For example, if a request message is published on the `$aws/things/myThing/jobs/get` topic, the AWS IoT Jobs service publishes success messages on the `$aws/things/myThing/jobs/get/accepted` topic and publishes rejected messages on the `$aws/things/myThing/jobs/get/rejected` topic.

using HTTP Signature Version 4

Communication between the AWS IoT Jobs service and your devices can occur over HTTP Signature Version 4 on port 443. This is the method used by the AWS SDKs and CLI. For more information about those tools, see [AWS CLI Command Reference: iot-jobs-data](#) or [AWS SDKs and Tools](#) and refer to the `iotJobsDataPlane` section for your preferred language.

**Note**

You must use the correct endpoint when you communicate with the AWS IoT Jobs service through HTTP Signature Version 4 or using an AWS SDK or CLI `iotJobsDataPlane` command. Use the `DescribeEndpoint` command to find it. For example, if you run this command:

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

you get a result similar to the following:

```
{  
    "endpointAddress": "a1b2c3d4e5f6g7.jobs.iot.us-west-2.amazonaws.com"  
}
```

With this method of communication, your device uses IAM credentials to authenticate with the AWS IoT Jobs service.

The following commands are available using this method:

- **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 ...
```

using HTTP TLS

Communication between the AWS IoT Jobs service and your devices can occur over HTTP TLS on port 8443 using a third-party software client that supports this protocol.

**Note**

You must use the correct endpoint when you communicate with the AWS IoT Jobs service through HTTP TLS. Use the **DescribeEndpoint** command to find it. For example, if you run this command:

```
aws iot describe-endpoint --endpoint-type iot:Jobs
```

you get a result similar to the following:

```
{  
    "endpointAddress": "a1b2c3d4e5f6g7.jobs.iot.us-west-2.amazonaws.com"  
}
```

With this method, your device uses X.509 certificate-based authentication (for example, its device-specific certificate and private key).

The following commands are available using this method:

- **DescribeJobExecution**
- **GetPendingJobExecutions**
- **StartNextPendingJobExecution**
- **UpdateJobExecution**

## Programming Devices to Work with Jobs

The examples in this section use MQTT to illustrate how a device works with the AWS IoT Jobs service. Alternatively, you could use the corresponding API or CLI commands. For these examples, we assume a device called *MyThing* subscribes to the following MQTT topics:

- \$aws/things/*MyThing*/jobs/notify (or \$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`

If you are using Code-signing for AWS IoT your device code must verify the signature of your code file. The signature will be in the job document within the `codesign` property. For more information about verifying a code file signature, see [Device Agent Sample](#).

## Device Workflow

There are two ways a device can handle the jobs it is given to execute.

### Option A: Get the next job

1. When a device first comes online, it should subscribe to the device's `notify-next` topic.
2. Call the [DescribeJobExecution \(p. 446\)](#) MQTT API with `jobId $next` to get the next job, its job document, and other details, including any state saved in `statusDetails`. If the job document has a code file signature, you must verify the signature before proceeding with processing the job request.
3. Call the [UpdateJobExecution \(p. 451\)](#) MQTT API to update the job status. Or, to combine this and the previous step in one call, the device can call [StartNextPendingJobExecution \(p. 441\)](#).
4. Optionally, you can add a step timer by setting a value for `stepTimeoutInMinutes` when you call either [UpdateJobExecution \(p. 451\)](#) or [StartNextPendingJobExecution \(p. 441\)](#).
5. Perform the actions specified by the job document using the [UpdateJobExecution \(p. 451\)](#) MQTT API to report on the progress of the job.
6. Continue to monitor the job execution by calling the [DescribeJobExecution \(p. 446\)](#) MQTT API with this `jobId`. If the job execution is canceled or deleted while the device is running the job, the device should be capable of recovering to a valid state.
7. Call the [UpdateJobExecution \(p. 451\)](#) MQTT API when finished with the job to update the job status and report success or failure.
8. Because this job's execution status has been changed to a terminal state, the next job available for execution (if any) changes. The device is notified that the next pending job execution has changed. At this point, the device should continue as described in step 2.

If the device remains online, it continues to receive notifications of the next pending job execution, including its job execution data, when it completes a job or a new pending job execution is added. When this occurs, the device continues as described in step 2.

### Option B: Pick from available jobs

1. When a device first comes online, it should subscribe to the thing's `notify` topic.
2. Call the [GetPendingJobExecutions \(p. 437\)](#) MQTT API to get a list of pending job executions.
3. If the list contains one or more job executions, pick one.
4. Call the [DescribeJobExecution \(p. 446\)](#) MQTT API to get the job document and other details, including any state saved in `statusDetails`.
5. Call the [UpdateJobExecution \(p. 451\)](#) MQTT API to update the job status. If the `includeJobDocument` field is set to `true` in this command, the device can skip the previous step and retrieve the job document at this point.
6. Optionally, you can add a step timer by setting a value for `stepTimeoutInMinutes` when you call [UpdateJobExecution \(p. 451\)](#).
7. Perform the actions specified by the job document using the [UpdateJobExecution \(p. 451\)](#) MQTT API to report on the progress of the job.
8. Continue to monitor the job execution by calling the [DescribeJobExecution \(p. 446\)](#) MQTT API with this `jobId`. If the job execution is canceled or deleted while the device is running the job, the device should be capable of recovering to a valid state.

9. Call the [UpdateJobExecution \(p. 451\)](#) MQTT API when finished with the job to update the job status and to report success or failure.

If the device remains online, it is notified of all pending job executions when a new pending job execution becomes available. When this occurs, the device can continue as described in step 2.

If the device is unable to execute the job, it should call the [UpdateJobExecution \(p. 451\)](#) MQTT API to update the job status to REJECTED.

## Starting a New Job

### new job notification

When a new job is created, the AWS IoT Jobs service publishes a message on the `$aws/things/thing-name/jobs/notify` topic for each target device.

[more info \(1\)](#)

The message contains the following information:

```
{
  "timestamp":1476214217017,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "0001",
        "queuedAt": 1476214216981,
        "lastUpdatedAt": 1476214216981,
        "versionNumber": 1
      }
    ]
  }
}
```

The device receives this message on the '`$aws/things/thingName/jobs/notify`' topic when the job execution is queued.

### get job information

To get more information about a job execution, the device calls the [DescribeJobExecution \(p. 446\)](#) MQTT API with the `includeJobDocument` field set to `true` (the default).

[more info \(2\)](#)

If the request is successful, the AWS IoT Jobs service publishes a message on the `$aws/things/MyThing/jobs/0023/get/accepted` topic:

```
{
  "clientToken": "client-001",
  "timestamp": 1489097434407,
  "execution": {
    "approximateSecondsBeforeTimedOut": number,
    "jobId": "023",
    "status": "QUEUED",
    "queuedAt": 1489097374841,
    "lastUpdatedAt": 1489097374841,
    "versionNumber": 1,
    "jobDocument": {
      < contents of job document >
    }
  }
}
```

}

**Note**

If the request fails, the AWS IoT Jobs service publishes a message on the `$aws/things/MyThing/jobs/0023/get/rejected` topic.

The device now has the job document that it can use to perform the remote operations for the job. If the job document contains an Amazon S3 presigned URL, the device can use that URL to download any required files for the job.

## Report Job Execution Status

update execution status

As the device is executing the job, it can call the [UpdateJobExecution \(p. 451\)](#) MQTT API to update the status of the job execution.

more info (3)

For example, a device can update the job execution status to `IN_PROGRESS` by publishing the following message on the `$aws/things/MyThing/jobs/0023/update` topic:

```
{  
    "status": "IN_PROGRESS",  
    "statusDetails": {  
        "progress": "50%"  
    },  
    "expectedVersion": "1",  
    "clientToken": "client001"  
}
```

Jobs responds by publishing a message to the `$aws/things/MyThing/jobs/0023/update/accepted` or `$aws/things/MyThing/jobs/0023/update/rejected` topic:

```
{  
    "clientToken": "client001",  
    "timestamp": 1476289222841  
}
```

The device can combine the two previous requests by calling [StartNextPendingJobExecution \(p. 441\)](#). That gets and starts the next pending job execution and allows the device to update the job execution status. This request also returns the job document when there is a job execution pending.

If the job contains a [TimeoutConfig](#), the in-progress timer starts running. You can also set a step timer for a job execution by setting a value for `stepTimeoutInMinutes` when you call [UpdateJobExecution](#). The step timer applies only to the job execution that you update. You can set a new value for this timer each time you update a job execution. You can also create a step timer when you call [StartNextPendingJobExecution](#). If the job execution remains in the `IN_PROGRESS` status for longer than the step timer interval, it fails and switches to the terminal `TIMED_OUT` status. The step timer has no effect on the in-progress timer that you set when you create a job.

**Note**

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

The `status` field can be set to `IN_PROGRESS`, `SUCCEEDED`, or `FAILED`. You cannot update the status of a job execution that is already in a terminal state.

report execution completed

When the device is finished executing the job, it calls the [UpdateJobExecution \(p. 451\)](#) MQTT API. If the job was successful, set `status` to `SUCCEEDED` and, in the message payload, in `statusDetails`, add other information about the job as name-value pairs. The in-progress and step timers end when the job execution is complete.

more info (4)

For example:

```
{  
    "status": "SUCCEEDED",  
    "statusDetails": {  
        "progress": "100%"  
    },  
    "expectedVersion": "2",  
    "clientToken": "client-001"  
}
```

If the job was not successful, set `status` to `FAILED` and, in `statusDetails`, add information about the error that occurred:

```
{  
    "status": "FAILED",  
    "statusDetails": {  
        "errorCode": "101",  
        "errorMsg": "Unable to install update"  
    },  
    "expectedVersion": "2",  
    "clientToken": "client-001"  
}
```

#### Note

The `statusDetails` attribute can contain any number of name-value pairs.

When the AWS IoT Jobs service receives this update, it publishes a message on the `$aws/things/MyThing/jobs/notify` topic to indicate the job execution is complete:

```
{  
    "timestamp": 1476290692776,  
    "jobs": {}  
}
```

## Additional Jobs

additional jobs

If there are other job executions pending for the device, they are included in the message published to `$aws/things/MyThing/jobs/notify`.

more info (5)

For example:

```
{  
    "timestamp": 1476290692776,  
    "jobs": {}  
}
```

```

    "QUEUED": [
        {
            "jobId": "0002",
            "queuedAt": 1476290646230,
            "lastUpdatedAt": 1476290646230
        }
    ],
    "IN_PROGRESS": [
        {
            "jobId": "0003",
            "queuedAt": 1476290646230,
            "lastUpdatedAt": 1476290646230
        }
    ]
}

```

## Jobs Notifications

The AWS IoT Jobs service publishes MQTT messages to reserved topics when jobs are pending or when the first job execution in the list changes. Devices can keep track of pending jobs by subscribing to these topics.

Job notifications are published to MQTT topics as JSON payloads. There are two kinds of notifications:

- A **ListNotification** contains a list of no more than 10 pending job executions. The job executions in this list have status values of either `IN_PROGRESS` or `QUEUED`. They are sorted by status (`IN_PROGRESS` job executions before `QUEUED` job executions) and then by the times when they were queued.

A **ListNotification** is published whenever one of the following criteria is met.

- A new job execution is queued or changes to a non-terminal status (`IN_PROGRESS` or `QUEUED`).
- An old job execution changes to a terminal status (`FAILED`, `SUCCEEDED`, `CANCELED`, `TIMED_OUT`, `REJECTED`, or `REMOVED`).
- A **NextNotification** contains summary information about the one job execution that is next in the queue.

A **NextNotification** is published whenever the first job execution in the list changes.

- A new job execution is added to the list as `QUEUED`, and it is the first one in the list.
- The status of an existing job execution that was not the first one in the list changes from `QUEUED` to `IN_PROGRESS` and becomes the first one in the list. (This happens when there are no other `IN_PROGRESS` job executions in the list or when the job execution whose status changes from `QUEUED` to `IN_PROGRESS` was queued earlier than any other `IN_PROGRESS` job execution in the list.)
- The status of the job execution that is first in the list changes to a terminal status and is removed from the list.

For more information about publishing and subscribing to MQTT topics, see [Message Broker for AWS IoT \(p. 233\)](#).

### Note

Notifications are not available when you use HTTP Signature Version 4 or HTTP TLS to communicate with jobs.

job pending

The AWS IoT Jobs service publishes a message on an MQTT topic when a job is added to or removed from the list of pending job executions for a thing or the first job execution in the list changes:

- `$aws/things/thingName/jobs/notify`
- `$aws/things/thingName/jobs/notify-next`

[more info \(6\)](#)

The messages contain the following example payloads:

\$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" }
  }
}
```

Possible job execution status values are QUEUED, IN\_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED\_OUT, REJECTED, and REMOVED.

The following series of examples show the notifications that are published to each topic as job executions are created and change from one state to another.

First, one job, called *job1*, is created. This notification is published to the jobs/notify topic:

```
{
  "timestamp": 1517016948,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

```
    }
}
```

This notification is published to the `jobs/notify-next` topic:

```
{
  "timestamp": 1517016948,
  "execution": {
    "jobId": "job1",
    "status": "QUEUED",
    "queuedAt": 1517016947,
    "lastUpdatedAt": 1517016947,
    "versionNumber": 1,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

When another job is created (`job2`), this notification is published to the `jobs/notify` topic:

```
{
  "timestamp": 1517017192,
  "jobs": {
    "QUEUED": [
      {
        "jobId": "job1",
        "queuedAt": 1517016947,
        "lastUpdatedAt": 1517016947,
        "executionNumber": 1,
        "versionNumber": 1
      },
      {
        "jobId": "job2",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "executionNumber": 1,
        "versionNumber": 1
      }
    ]
  }
}
```

A notification is not published to the `jobs/notify-next` topic because the next job in the queue (`job1`) has not changed. When `job1` starts to execute, its status changes to `IN_PROGRESS`. No notifications are published because the list of jobs and the next job in the queue have not changed.

When a third job (`job3`) is added, this notification is published to the `jobs/notify` topic:

```
{
  "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
        }
    ]
}

```

A notification is not published to the `jobs/notify-next` topic because the next job in the queue is still `job1`.

When `job1` is complete, its status changes to `SUCCEEDED`, and this notification is published to the `jobs/notify` topic:

```

{
    "timestamp": 1517186269,
    "jobs": {
        "QUEUED": [
            {
                "jobId": "job2",
                "queuedAt": 1517017191,
                "lastUpdatedAt": 1517017191,
                "executionNumber": 1,
                "versionNumber": 1
            },
            {
                "jobId": "job3",
                "queuedAt": 1517017905,
                "lastUpdatedAt": 1517017905,
                "executionNumber": 1,
                "versionNumber": 1
            }
        ]
    }
}

```

At this point, `job1` has been removed from the queue, and the next job to be executed is `job2`. This notification is published to the `jobs/notify-next` topic:

```

{
    "timestamp": 1517186269,
    "execution": {
        "jobId": "job2",
        "status": "QUEUED",
        "queuedAt": 1517017191,
        "lastUpdatedAt": 1517017191,
        "versionNumber": 1,
        "executionNumber": 1,
        "jobDocument": {
            "operation": "test"
        }
    }
}

```

```
    }
}
```

If `job3` must begin executing before `job2` (which is not recommended), the status of `job3` can be changed to `IN_PROGRESS`. If this happens, `job2` is no longer next in the queue, and this notification is published to the `jobs/notify-next` topic:

```
{
  "timestamp": 1517186779,
  "execution": {
    "jobId": "job3",
    "status": "IN_PROGRESS",
    "queuedAt": 1517017905,
    "startedAt": 1517186779,
    "lastUpdatedAt": 1517186779,
    "versionNumber": 2,
    "executionNumber": 1,
    "jobDocument": {
      "operation": "test"
    }
  }
}
```

No notification is published to the `jobs/notify` topic because no job has been added or removed.

If the device rejects `job2` and updates its status to `REJECTED`, this notification is published to the `jobs/notify` topic:

```
{
  "timestamp": 1517189392,
  "jobs": {
    "IN_PROGRESS": [
      {
        "jobId": "job3",
        "queuedAt": 1517017905,
        "lastUpdatedAt": 1517186779,
        "startedAt": 1517186779,
        "executionNumber": 1,
        "versionNumber": 2
      }
    ]
  }
}
```

If `job3` (which is still in progress) is force deleted, this notification is published to the `jobs/notify` topic:

```
{
  "timestamp": 1517189551,
  "jobs": {}
}
```

At this point, the queue is empty. This notification is published to the `jobs/notify-next` topic:

```
{
  "timestamp": 1517189551
}
```

# Using the AWS IoT Jobs APIs

There are two categories of API used in the AWS IoT Jobs service:

- Those used for management and control of jobs.
- Those used by the devices executing those jobs.

In general, job management and control uses an HTTPS protocol API. Devices can use either an MQTT or an HTTPS protocol API. (The HTTPS API is designed for a low volume of calls typical when creating and tracking jobs. It usually opens a connection for a single request, and then closes the connection after the response is received. The MQTT API allows long polling. It is designed for large amounts of traffic that can scale to millions of devices.)

## Note

Each AWS IoT Jobs HTTPS API has a corresponding command that allows you to call the API from the AWS CLI. The commands are lowercase, with hyphens between the words that make up the name of the API. For example, you can invoke the `CreateJob` API on the CLI by typing:

```
aws iot create-job ...
```

## Job Management and Control API

### Job Management and Control Data Types

The following data types are used by management and control applications to communicate with the AWS IoT Jobs service.

#### Job

##### Job data type

The `Job` object contains details about a job.

##### syntax (1)

```
{  
    "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
    }
}

```

**description (1)**

**jobArn**

An ARN identifying the job with the format "arn:aws:iot:*region:account*:job/*jobId*".

**jobId**

The unique identifier you assigned to this job when it was created.

**status**

The status of the job, one of IN\_PROGRESS, CANCELED, or SUCCEEDED.

**targetSelection**

Specifies whether the job continues to run (CONTINUOUS) or is complete after those things specified as targets have completed the job (SNAPSHOT). If CONTINUOUS, the job might also be run on a thing when a change is detected in a target. For example, a job runs on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.

**comment**

If the job was updated, describes the reason for the update.

**targets**

A list of AWS IoT things and thing groups to which the job should be sent.

**description**

A short text description of the job.

`createdAt`

The time, in seconds since the epoch, when the job was created.

`lastUpdatedAt`

The time, in seconds since the epoch, when the job was last updated.

`completedAt`

The time, in seconds since the epoch, when the job was completed.

`jobProcessDetails`

Details about the job process:

`processingTargets`

A list of AWS IoT things and thing groups that are currently executing the job.

`numberOfCanceledThings`

The number of AWS IoT things that canceled the job.

`numberOfSucceededThings`

The number of AWS IoT things that successfully completed the job.

`numberOfFailedThings`

The number of AWS IoT things that failed to complete the job.

`numberOfRejectedThings`

The number of AWS IoT things that rejected the job.

`numberOfQueuedThings`

The number of AWS IoT things that are awaiting execution of the job.

`numberOfInProgressThings`

The number of AWS IoT things that are currently executing the job.

`numberOfRemovedThings`

The number of AWS IoT things that are no longer scheduled to execute the job because they have been deleted or removed from the group that was a target of the job.

`numberOfTimedOutThings`

The number of things whose job execution status is `TIMED_OUT`.

`presignedUrlConfig`

Configuration information for presigned Amazon S3 URLs.

`expiresInSec`

How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.

`roleArn`

The ARN of an IAM role that grants permission to download files from an Amazon S3 bucket. The role must also grant permission for AWS IoT to download the files. For more information about how to create and configure the role, see [Create Jobs \(p. 358\)](#).

`jobExecutionRolloutConfig`

Optional. Allows you to create a staged rollout of a job.

`maximumPerMinute`

The maximum number of things (devices) to which the job is sent for execution, per minute.

`exponentialRate`

Allows you to create an exponential rate of rollout for a job.

`baseRatePerMinute`

The minimum number of things that are notified of a pending job, per minute, at the start of job rollout. This parameter allows you to define the initial rate of rollout.

`incrementFactor`

The exponential factor to increase the rate of rollout for a job.

`rateIncreaseCriteria`

The criteria to initiate the increase in rate of rollout for a job. You can specify either `numberOfNotifiedThings` or `numberOfSucceededThing`, but not both.

`numberOfNotifiedThings`

The threshold for number of notified things that initiate the increase in rate of rollout.

`numberOfSucceededThings`

The threshold for number of succeeded things that initiate the increase in rate of rollout.

`abortConfig`

Optional. Details of abort criteria to abort the job.

`criteriaList`

The list of abort criteria to define rules to abort the job.

`action`

The type of abort action to initiate a job abort.

`failureType`

The type of job execution failure to define a rule to initiate a job abort.

`minNumberOfExecutedThings`

Minimum number of executed things before evaluating an abort rule.

`thresholdPercentage`

The threshold as a percentage of the total number of executed things that initiate a job abort.

`timeoutConfig`

Optional. Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to `IN_PROGRESS`. If the job execution status is not set to another terminal state before the time expires, it is set to `TIMED_OUT`.

**Note**

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

**inProgressTimeoutInMinutes**

Specifies the amount of time, in minutes, this device has to finish execution of this job. A timer is started, or restarted, whenever this job's execution status is specified as IN\_PROGRESS with this field populated. If the job execution status is not set to a terminal state before the timer expires, or before another job execution status update is sent with this field populated, the status is set to TIMED\_OUT.

## JobSummary

JobSummary data type

The JobSummary object contains a job summary.

**syntax (2)**

```
{  
    "jobArn": "string",  
    "jobId": "string",  
    "status": "IN_PROGRESS|CANCELED|SUCCEEDED",  
    "targetSelection": "CONTINUOUS|SNAPSHOT",  
    "thingGroupId": "string",  
    "createdAt": timestamp,  
    "lastUpdatedAt": timestamp,  
    "completedAt": timestamp  
}
```

**description (2)**

**jobArn**

An ARN that identifies the job.

**jobId**

The unique identifier you assigned to this job when it was created.

**status**

The job status. Can be one of IN\_PROGRESS, CANCELED, or SUCCEEDED.

**targetSelection**

Specifies whether the job continues to run (CONTINUOUS) or is complete after all those things specified as targets have completed the job (SNAPSHOT). If CONTINUOUS, the job might also be run on a thing when a change is detected in a target. For example, a job runs on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.

**thingGroupId**

The ID of the thing group.

**createdAt**

The UNIX timestamp for when the job was created.

**lastUpdatedAt**

The UNIX timestamp for when the job was last updated.

**completedAt**

The UNIX timestamp for when the job was completed.

## JobExecution

**JobExecution** data type

The **JobExecution** object represents the execution of a job on a device.

**syntax (3)**

```
{  
    "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  
}
```

**description (3)**

**approximateSecondsBeforeTimedOut**

The estimated number of seconds that remain before the job execution status is changed to **TIMED\_OUT**. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The actual job execution timeout can occur up to 60 seconds later than the estimated duration.

**jobId**

The unique identifier you assigned to this job when it was created.

**executionNumber**

A number that identifies this job execution on this device. It can be used later in commands that return or update job execution information.

**thingArn**

The AWS IoT thing ARN.

**queuedAt**

The time, in seconds since the epoch, when the job execution was queued.

**lastUpdatedAt**

The time, in seconds since the epoch, when the job execution was last updated.

**startedAt**

The time, in seconds since the epoch, when the job execution was started.

**status**

The status of the job execution. Can be one of QUEUED, IN\_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED\_OUT, REJECTED, or REMOVED.

**statusDetails**

A collection of name-value pairs that describe the status of the job execution.

## JobExecutionSummary

### JobExecutionSummary data type

The `JobExecutionSummary` object contains job execution summary information:

#### syntax (4)

```
{  
    "executionNumber": 1234567890,  
    "queuedAt": timestamp,  
    "lastUpdatedAt": timestamp,  
    "startedAt": timestamp,  
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|REMOVED"  
}
```

#### description (4)

**executionNumber**

A number that identifies a job execution on a device. It can be used later in commands that return or update job execution information.

**queuedAt**

The time, in seconds since the epoch, when the job execution was queued.

**lastUpdatedAt**

The time, in seconds since the epoch, when the job execution was last updated.

**startAt**

The time, in seconds since the epoch, when the job execution was started.

**status**

The status of the job execution: QUEUED, IN\_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED\_OUT, REJECTED, or REMOVED.

## JobExecutionSummaryForJob

### JobExecutionSummaryForJob data type

The `JobExecutionSummaryForJob` object contains a summary of information about job executions for a specific job.

#### syntax (5)

```
{  
    "executionSummaries": [  
        {
```

```
        "thingArn": "arn:aws:iot:us-west-2:123456789012:thing/MyThing",
        "jobExecutionSummary": [
            {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            }
        ],
        ...
    }
```

description (5)

**thingArn**

The AWS IoT thing ARN.

**jobExecutionSummary**

An [JobExecutionSummary \(p. 384\)](#) object.

## JobExecutionSummaryForThing

JobExecutionSummaryForThing data type

The `JobExecutionSummaryForThing` object contains a summary of information about a job execution on a specific thing.

syntax (6)

```
{
    "executionSummaries": [
        {
            "jobExecutionSummary": {
                "status": "IN_PROGRESS",
                "lastUpdatedAt": 1549395301.389,
                "queuedAt": 1541526002.609,
                "executionNumber": 1
            },
            "jobId": "MyThingJob"
        },
        ...
    ]
}
```

description (6)

**jobId**

The unique identifier you assigned to this job when it was created.

**jobExecutionSummary**

A [JobExecutionSummary \(p. 384\)](#) object.

## Job Management and Control HTTPS Commands

The following commands are available for management and control applications over the HTTPS protocol.

## AssociateTargetsWithJob

### AssociateTargetsWithJob command

Associates a group with a continuous job. For more information, see [CreateJob \(p. 393\)](#). The following criteria must be met:

- The job must have been created with the `targetSelection` field set to `CONTINUOUS`.
- The job status must currently be `IN_PROGRESS`.
- The total number of targets associated with a job must not exceed 100.

### HTTPS (1)

Request:

```
POST /jobs/jobId/targets
{
    "targets": [ "string" ],
    "comment": "string"
}
```

**jobId**

The unique identifier you assigned to this job when it was created.  
**targets**

A list of thing group ARNs that define the targets of the job.

**comment**

Optional. A comment string that describes why the job was associated with the targets.

Response:

```
{
    "jobArn": "string",
    "jobId": "string",
    "description": "string"
}
```

**jobArn**

An ARN identifying the job.  
**jobId**

The unique identifier you assigned to this job when it was created.  
**description**

A short text description of the job.

### CLI (1)

**Synopsis:**

```
aws iot associate-targets-with-job \
--targets <value> \
--job-id <value> \
```

```
[--comment <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "targets": [
    "string"
  ],
  "jobId": "string",
  "comment": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
targets	list  member: TargetArn	A list of thing group ARNs that define the targets of the job.
TargetArn	string	
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
comment	string  length max:2028  pattern: [^\p{C}]+	An optional string that describes why the job was associated with the targets.

**Output:**

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

**cli output fields:**

Name	Type	Description
jobArn	string	An ARN identifying the job.
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
description	string  length max:2028	A short text description of the job.

Name	Type	Description
	pattern: [^\\p{C}]+	

## MQTT (1)

Not available.

## CancelJob

CancelJob command

Cancels a job.

### HTTPS (2)

Request:

```
PUT /jobs/jobId/cancel
{
    "force": boolean,
    "comment": "string",
    "reasonCode": "string"
}
```

**jobId**

The unique identifier you assigned to this job when it was created.

**force**

[Optional] If true, job executions with status IN\_PROGRESS and QUEUED are canceled. Otherwise, only job executions with status QUEUED are canceled. The default is false.

### Warning

Canceling a job with a status of IN\_PROGRESS causes a device that is executing the job to be unable to update the job execution status. Use caution and make sure that each device executing a job that is canceled is able to recover to a valid state.

**comment**

[Optional] A comment string that describes why the job was canceled.

**reasonCode**

[Optional] A reason code string that explains why the job was canceled. If a job is cancelled because it meets the conditions defined by an abortConfig, this field is auto-populated.

Response:

```
{
    "jobArn": "string",
    "jobId": "string",
    "description": "string"
}
```

**jobArn**

The job ARN.

**jobId**

The unique identifier you assigned to this job when it was created.

**description**

A short text description of the job.

**CLI (2)**

**Synopsis:**

```
aws iot cancel-job \
--job-id <value> \
[--force <value>] \
[--comment <value>] \
[--reasonCode <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "jobId": "string",
  "force": boolean,
  "comment": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
force	boolean	If true, jobs with status QUEUED and IN_PROGRESS are canceled. Otherwise, only jobs with status QUEUED are canceled.  <b>Warning</b> Canceling a job with a status of IN_PROGRESS causes a device that is executing the job to be unable to update the job execution status. Use caution and make sure that each device executing a job that is canceled is able to recover to a valid state.

Name	Type	Description
comment	string length max:2028 pattern: [^\p{C}]+	An optional string that describes why the job was canceled.
reasonCode	string length max:128 pattern: [\p{Upper}\p{Digit}_]+	An optional string that explains why the job was canceled. If the job is canceled because it meets the criteria defined by the <code>abortConfig</code> , this field is auto-populated.

Output:

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

**cli output fields:**

Name	Type	Description
jobArn	string	The job ARN.
jobId	string length max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
description	string length max:2028 pattern: [^\p{C}]+	A short text description of the job.

MQTT (2)

Not available.

## CancelJobExecution

CancelJobExecution command

Cancels a job execution on a device.

HTTPS (3)

Request:

```
PUT /things/thingName/jobs/jobId/cancel
{
```

```
        "force": boolean,  
        "expectedVersion": "string",  
        "statusDetails": {  
            "string": "string"  
        }  
    }  
}
```

**thingName**

The name of the thing whose job execution will be canceled.

**jobId**

The unique identifier you assigned to the job when it was created.

**force**

Optional. If `true`, a job execution with a status of `IN_PROGRESS` or `QUEUED` can be canceled. Otherwise, only a job execution with status of `QUEUED` can be canceled. If you attempt to cancel a job execution with a status of `IN_PROGRESS` and you do not set `force` to `true`, an `InvalidStateTransitionException` is thrown. The default is `false`.

**Warning**

Cancelling a job with a status of `IN_PROGRESS` causes a device that is executing the job to be unable to update the job execution status. Use caution and make sure that each device executing a job that is canceled is able to recover to a valid state.

**expectedVersion**

Optional. The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in the AWS IoT Jobs service does not match, the update is rejected with a `VersionConflictException` error, and an `ErrorResponse` that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate `DescribeJobExecution` request to obtain the job execution status data.)

**statusDetails**

Optional. A collection of name-value pairs that describe the status of the job execution.

**Response:**

```
{  
}
```

**CLI (3)**

**Synopsis:**

```
aws iot cancel-job-execution \  
  --job-id <value> \  
  --thing-name <value> \  
  [--force | --no-force] \  
  [--expected-version <value>] \  
  [--status-details <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

**cli-input-json format:**

```
{  
    "jobId": "string",
```

```

    "thingName": "string",
    "force": boolean,
    "expectedVersion": long,
    "statusDetails": {
        "string": "string"
    }
}

```

**cli-input-json fields:**

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The job to be canceled.
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the thing whose execution of the job will be canceled.
force	boolean	<p>Optional. If <code>true</code>, the job execution is canceled if it has status of IN_PROGRESS or QUEUED. Otherwise, the job execution is canceled only if it has status of QUEUED. However, if you attempt to cancel a job execution that has a status of IN_PROGRESS, and you do not set <code>--force</code> to <code>true</code>, an <code>InvalidStateException</code> is thrown. The default is <code>false</code>.</p> <p><b>Warning</b> Canceling a job that has a status of IN_PROGRESS, causes a device that is executing the job to be unable to update the job execution status. Use caution and make sure that each device executing a job that is canceled is able to recover to a valid state.</p>
expectedVersion	long  java class: <code>java.lang.Long</code>	Optional. The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in the

Name	Type	Description
		AWS IoT Jobs service does not match, the update is rejected with a <code>VersionMismatch</code> error, and an <code>ErrorResponse</code> that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate <code>DescribeJobExecution</code> request to obtain the job execution status data.)
statusDetails	map key: DetailsKey value: DetailsValue	A collection of name-value pairs that describe the status of the job execution. If not specified, the <code>statusDetails</code> are unchanged.
DetailsKey	string length max:128 min:1 pattern: [a-zA-Z0-9:_-]+	
DetailsValue	string length max:1024 min:1 pattern: [^\p{C}]*+	

Output:

None

MQTT (3)

Not available.

## CreateJob

CreateJob command

Creates a job. You can provide the job document as a link to a file in an Amazon S3 bucket (`documentSource` parameter) or in the body of the request (`document` parameter).

A job can be made *continuous* by setting the optional `targetSelection` parameter to `CONTINUOUS`. (The default is `SNAPSHOT`.) A continuous job can be used to onboard or upgrade devices as they are added to a group because it continues to run and is executed on newly added things, even after the things in the group at the time the job was created have completed the job.

A job can have an optional `TimeoutConfig`, which sets the value of the in-progress timer. The in-progress timer can't be updated and applies to all executions of the job.

### Note

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

The following validations are performed on arguments to the CreateJob API:

- The `targets` argument must be a list of valid thing or thing group ARNs. All things and thing groups must be in your AWS account.
- The `documentSource` argument must be a valid Amazon S3 URL to a job document. Amazon S3 URLs are of the form: <https://s3.amazonaws.com/bucketName/objectName>.
- The document stored in the URL specified by the `documentSource` argument must be a UTF-8 encoded JSON document.
- The size of a job document is limited to 32 KB due to the limit on the size of an MQTT message(128 KB) and encryption.
- The `jobId` must be unique in your AWS account.

#### HTTPS (4)

Request:

```
PUT /jobs/jobId

{
    "targets": [ "string" ],
    "document": "string",
    "documentSource": "string",
    "description": "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
    }
}
```

**jobId**

A job identifier, which must be unique for your AWS account. We recommend using a UUID. Alphanumeric characters, "-", and "\_" can be used here.

**targets**

A list of thing or thing group ARNs that defines the targets of the job.

`document`

Optional. The job document.

`documentSource`

Optional. An Amazon S3 link to the job document.

`description`

Optional. A short text description of the job.

`presignedUrlConfigData`

Optional. Configuration information for presigned Amazon S3 URLs.

`roleArn`

The ARN of the IAM role that contains permissions to access the Amazon S3 bucket. This is the bucket that contains the data that devices download with the presigned Amazon S3 URLs. This role must also grant AWS IoT permission to assume the role. For more information, see [Create Jobs \(p. 358\)](#).

`expiresInSec`

How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.

`targetSelection`

Optional. Specifies whether the job continues to run (CONTINUOUS) or is complete after all those things specified as targets have completed the job (SNAPSHOT). If CONTINUOUS, the job might also be scheduled to run on a thing when a change is detected in a target. For example, a job is scheduled to run on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.

`jobExecutionRolloutConfig`

Optional. Allows you to create a staged rollout of a job.

`maximumPerMinute`

The maximum number of things on which the job is sent for execution, per minute. Valid values are 1 to 1000. If not specified, the default is 1000. The actual number of things that receive the job might be less during any particular minute interval (due to system latency), but is not more than the specified value.

`exponentialRate`

Allows you to create an exponential rate of rollout for a job.

`baseRatePerMinute`

The minimum number of things that are notified of a pending job, per minute, at the start of job rollout. This parameter allows you to define the initial rate of rollout.

`incrementFactor`

The exponential factor to increase the rate of rollout for a job.

`rateIncreaseCriteria`

The criteria to initiate the increase in rate of rollout for a job. Set values for either the `numberOfNotifiedThings` or `numberOfSucceededThings`, but not both.

`numberOfNotifiedThings`

The threshold for number of notified things that initiate the increase in rate of rollout.

`numberOfSucceededThings`

The threshold for number of succeeded things that initiate the increase in rate of rollout.

`abortConfig`

Optional. Details of abort criteria to abort the job.

`criteriaList`

The list of abort criteria to define rules to abort the job.

`action`

The type of abort action to initiate a job abort.

`failureType`

The type of job execution failure to define a rule to initiate a job abort.

`minNumberOfExecutedThings`

Minimum number of executed things before evaluating an abort rule.

`thresholdPercentage`

The threshold as a percentage of the total number of executed things that initiate a job abort.

`timeoutConfig`

Optional. Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to `IN_PROGRESS`. If the job execution status is not set to another terminal state before the time expires, it is set to `TIMED_OUT`.

**Note**

The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.

`inProgressTimeoutInMinutes`

Specifies the amount of time, in minutes, this device has to finish execution of this job. A timer is started, or restarted, whenever this job's execution status is specified as `IN_PROGRESS` with this field populated. If the job execution status is not set to a terminal state before the timer expires, or before another job execution status update is sent with this field populated, the status is set to `TIMED_OUT`.

Response:

```
{  
    "jobArn": "string",  
    "jobId": "string",  
    "description": "string"  
}
```

`jobArn`

The ARN of the job.

`jobId`

The unique identifier you assigned to this job.

**description**

An optional short text description of the job.

**CLI (4)**

**Synopsis:**

```
aws iot create-job \
--job-id <value> \
--targets <value> \
[--document-source <value>] \
[--document <value>] \
[--description <value>] \
[--presigned-url-config <value>] \
[--target-selection <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--document-parameters <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "jobId": "string",
  "targets": [
    "string"
  ],
  "documentSource": "string",
  "document": "string",
  "description": "string",
  "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"
    }
}
```

**cli-input-json fields:**

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	A job identifier which must be unique for your AWS account. We recommend using a UUID. Alphanumeric characters, "-" and "_" are valid for use here.
targets	list  member: TargetArn	A list of things and thing groups to which the job should be sent.
TargetArn	string	
documentSource	string  length max:1350 min:1	An S3 link to the job document.
document	string  length max:32768	The job document.
description	string  length max:2028  pattern: [^\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration information for presigned S3 URLs.
roleArn	string  length max:2048 min:20	The ARN of an IAM role that grants permission to download files from the Amazon S3 bucket where the job data or updates are stored. The role must also grant permission for AWS IoT to download the files.
expiresInSec	long  java class: java.lang.Long  range- max:3600 min:60	How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.
targetSelection	string  enum: CONTINUOUS   SNAPSHOT	Specifies whether the job continues to run (CONTINUOUS), or is complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous,

Name	Type	Description
		the job can also be run on a thing when a change is detected in a target. For example, a job runs on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of the job.
maximumPerMinute	integer  java class: java.lang.Integer  range- max:1000 min:1	The maximum number of things that are notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	java class: java.lang.Integer	The minimum number of things that will be notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.
incrementFactor	java class: java.lang.Double	The exponential factor to increase the rate of rollout for a job.
rateIncreaseCriteria	RateIncreaseCriteria	Allows you to define a criteria to initiate the increase in rate of rollout for a job. Set a value for either <code>numberOfNotifiedThings</code> or <code>numberOfSucceededThings</code> , but not both.
numberOfNotifiedThings	java class: java.lang.Double	The threshold for number of notified things that will initiate the increase in rate of rollout.
numberOfSucceededThings	java class: java.lang.Double	The threshold for number of succeeded things that will initiate the increase in rate of rollout.
abortConfig	AbortConfig	Allows you to create criteria to abort a job.
criteriaList	AbortCriteria	The list of abort criteria to define rules to abort the job.

Name	Type	Description
action	java class: java.lang.String (CANCEL)	The type of abort action to initiate a job abort.
failureType	java class: java.lang.String (FAILED   REJECTED   TIMED_OUT   ALL)	The type of job execution failure to define a rule to initiate a job abort.
minNumberOfExecutedThings	java class: java.lang.Integer	Minimum number of executed things before evaluating an abort rule.
thresholdPercentage	java class: java.lang.Double	<p>The threshold as a percentage of the total number of executed things that will initiate a job abort.</p> <p>AWS IoT supports up to two digits after the decimal (for example, 10.9 and 10.99, but not 10.999).</p>
timeoutConfig	TimeoutConfig	<p>Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to IN_PROGRESS. If the job execution status is not set to another terminal state before the time expires, it is set to TIMED_OUT.</p> <p><b>Note</b> The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.</p>
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. A timer is started, or restarted, whenever this job's execution status is specified as IN_PROGRESS with this field populated. If the job execution status is not set to a terminal state before the timer expires, or before another job execution status update is sent with this field populated, the status is set to TIMED_OUT.

Name	Type	Description
documentParameters	map  key: ParameterKey  value: ParameterValue	Parameters for the job document.
ParameterKey	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	
ParameterValue	string  length max:1024 min:1  pattern: [^\p{C}]+	

Output:

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

**cli output fields:**

Name	Type	Description
jobArn	string	The job ARN.
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job.
description	string  length max:2028  pattern: [^\p{C}]+	The job description.

## MQTT (4)

Not available.

## DeleteJob

### DeleteJob command

Deletes a job and its related job executions.

Deleting a job can take time, depending on the number of job executions created for the job and various other factors. While the job is being deleted, the status of the job is shown

as "DELETION\_IN\_PROGRESS". Attempting to delete or cancel a job whose status is already "DELETION\_IN\_PROGRESS" results in an error.

#### HTTPS (5)

##### **Request syntax:**

```
DELETE /jobs/jobId?force=force
```

##### **URI Request Parameters:**

Name	Type	Req?	Description
jobId	JobId	yes	The ID of the job to be deleted.
force	ForceFlag	no	(Optional) When true, you can delete a job with a status of "IN_PROGRESS". Otherwise, you can only delete a job that is in a terminal state ("SUCCEEDED" or "CANCELED") or an exception occurs. The default is false.

##### **Note**

Deleting a job with a status of "IN\_PROGRESS", causes a device that is executing the job to be unable to access job information or update the job execution status. Use caution and make sure that each device executing a job that is deleted is able to recover to a valid state.

##### **Errors:**

#### InvalidRequestException

The contents of the request were invalid. For example, this code is returned when an UpdateJobExecution request contains invalid status details. The message contains details about the error.

HTTP response code: 400

#### InvalidStateException

An update attempted to change the job or job execution to a state that is invalid because of its current state (for example, an attempt to change a request in state SUCCEEDED to state IN\_PROGRESS). In this case, the body of the error message also contains the executionState field.

HTTP response code: 409

#### ResourceNotFoundException

The specified resource does not exist.

HTTP response code: 404

#### ThrottlingException

The rate exceeds the limit.

HTTP response code: 429

#### ServiceUnavailableException

The service is temporarily unavailable.

HTTP response code: 503

### CLI (5)

#### Synopsis:

```
aws iot delete-job \
--job-id <value> \
[--force | --no-force] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{  
    "jobId": "string",  
    "force": boolean  
}
```

#### cli-input-json fields:

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the job to be deleted.

Name	Type	Description
force	boolean	<p>(Optional) When true, you can delete a job with a status of IN_PROGRESS. Otherwise, you can only delete a job that is in a terminal state (SUCCEEDED or CANCELED) or an exception occurs. The default is false.</p> <p><b>Note</b> Deleting a job with a status of IN_PROGRESS, causes a device that is executing the job to be unable to access job information or update the job execution status. Use caution and make sure that each device executing a job that is deleted is able to recover to a valid state.</p>

Output:

None

MQTT (5)

Not available.

## DeleteJobExecution

DeleteJobExecution command

Deletes a job execution.

HTTPS (6)

**Request syntax:**

```
DELETE /things/thingName/jobs/jobId/executionNumber/executionNumber?force=force
```

### URI Request Parameters:

Name	Type	Req?	Description
jobId	JobId	yes	The ID of the job whose execution will be deleted.
thingName	ThingName	yes	The name of the thing whose execution of the job will be deleted.

Name	Type	Req?	Description
executionNumber	ExecutionNumber	yes	The ID of the job execution to be deleted.
force	ForceFlag	no	<p>When true, you can delete a job execution with a status of IN_PROGRESS. Otherwise, you can only delete a job execution that is in a terminal state (SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED, or CANCELED) or an exception occurs. The default is false.</p> <p><b>Note</b>            Deleting a job execution with a status of IN_PROGRESS causes the device to be unable to access job information or update the job execution status. Use caution and make sure that the device is able to recover to a valid state.</p>

**Errors:**

**InvalidRequestException**

The contents of the request were invalid. For example, this code is returned when an UpdateJobExecution request contains invalid status details. The message contains details about the error.

HTTP response code: 400

**InvalidStateTransitionException**

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state SUCCEEDED to state IN\_PROGRESS). In this case, the body of the error message also contains the executionState field.

HTTP response code: 409  
**ResourceNotFoundException**  
The specified resource does not exist.

HTTP response code: 404  
**ThrottlingException**  
The rate exceeds the limit.

HTTP response code: 429  
**ServiceUnavailableException**  
The service is temporarily unavailable.

HTTP response code: 503

## CLI (6)

### Synopsis:

```
aws iot delete-job-execution \
--job-id <value> \
--thing-name <value> \
--execution-number <value> \
[--force | --no-force] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "jobId": "string",
  "thingName": "string",
  "executionNumber": long,
  "force": boolean
}
```

### **cli-input-json** fields:

Name	Type	Description
jobId	string length max:64 min:1 pattern: [a-zA-Z0-9_-]+	The ID of the job whose execution will be deleted.
thingName	string length max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the thing whose execution of the job will be deleted.
executionNumber	long java class: java.lang.Long	The ID of the job execution to be deleted.
force	boolean	When true, you can delete a job execution with a status

Name	Type	Description
		<p>of IN_PROGRESS. Otherwise, you can only delete a job execution that is in a terminal state (SUCCEEDED, FAILED, TIMED_OUT, REJECTED, REMOVED, or CANCELED) or an exception occurs. The default is false.</p> <p><b>Note</b> Deleting a job execution with a status of IN_PROGRESS causes the device to be unable to access job information or update the job execution status. Use caution and make sure that the device is able to recover to a valid state.</p>

Output:

None

MQTT (6)

Not available.

## DescribeJob

DescribeJob command

Gets the details of the specified job.

HTTPS (7)

Request:

```
GET /jobs/jobId
```

**jobId**

The unique identifier you assigned to this job when it was created.

Response:

```
{
  "documentSource": "string",
  "job": Job
}
```

`documentSource`

An Amazon S3 link to the job document.

`job`

A [Job \(p. 378\)](#) object.

CLI (7)

**Synopsis:**

```
aws iot describe-job \
--job-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
  "jobId": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
<code>jobId</code>	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.

Output:

```
{
  "documentSource": "string",
  "job": {
    "jobArn": "string",
    "jobId": "string",
    "targetSelection": "string",
    "status": "string",
    "forceCanceled": boolean,
    "comment": "string",
    "targets": [
      "string"
    ],
    "description": "string",
    "presignedUrlConfig": {
      "roleArn": "string",
      "expiresInSec": long
    },
    "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
            }
        ]
    },
    "createdAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "completedAt": "timestamp",
    "jobProcessDetails": {
        "processingTargets": [
            "string"
        ],
        "numberOfCanceledThings": "integer",
        "numberOfSucceededThings": "integer",
        "numberOfFailedThings": "integer",
        "numberOfRejectedThings": "integer",
        "numberOfQueuedThings": "integer",
        "numberOfInProgressThings": "integer",
        "numberOfRemovedThings": "integer",
        "numberOfTimedOutThings": "integer"
    },
    "documentParameters": {
        "string": "string"
    },
    "timeoutConfig": {
        "inProgressTimeoutInMinutes": number
    }
}
}

```

**cli output fields:**

Name	Type	Description
documentSource	string  length max:1350 min:1	An Amazon S3 link to the job document.
job	Job	Information about the job.
jobArn	string	An ARN identifying the job with format "arn:aws:iot:region:account:job/jobId".
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
targetSelection	string  enum: CONTINUOUS   SNAPSHOT	Specifies whether the job continues to run (CONTINUOUS), or is complete after all those things specified as targets have completed the

Name	Type	Description
		job (SNAPSHOT). If continuous, the job can also be run on a thing when a change is detected in a target. For example, a job runs on a device when the thing representing the device is added to a target group, even after the job was completed by all things originally in the group.
status	string  enum: IN_PROGRESS   CANCELED   SUCCEEDED	The status of the job, one of IN_PROGRESS, CANCELED, or SUCCEEDED.
forceCanceled	boolean  java class: java.lang.Boolean	Is true if the job was canceled with the optional force parameter set to true.
comment	string  length max:2028  pattern: [^\p{C}]+	If the job was updated, describes the reason for the update.
targets	list  member: TargetArn	A list of AWS IoT things and thing groups to which the job should be sent.
TargetArn	string	
description	string  length max:2028  pattern: [^\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration for presigned Amazon S3 URLs.
roleArn	string  length max:2048 min:20	The ARN of an IAM role that grants permission to download files from the Amazon S3 bucket where the job data or updates are stored. The role must also grant permission for tAWS IoT Jobs service to download the files.
expiresInSec	long  java class: java.lang.Long  range- max:3600 min:60	How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.

Name	Type	Description
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of the job.
maximumPerMinute	integer  java class: java.lang.Integer  range- max:1000 min:1	The maximum number of things that are notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	java class: java.lang.Integer	The minimum number of things that are notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.
incrementFactor	java class: java.lang.Double	The exponential factor to increase the rate of rollout for a job.
rateIncreaseCriteria	RateIncreaseCriteria	Allows you to define a criteria to initiate the increase in rate of rollout for a job. Set a value for either <code>numberOfNotifiedThings</code> or <code>numberOfSucceededThings</code> , but not both.
numberOfNotifiedThings	java class: java.lang.Double	The threshold for number of notified things that initiate the increase in rate of rollout.
numberOfSucceededThings	java class: java.lang.Double	The threshold for number of succeeded things that initiate the increase in rate of rollout.
abortConfig	AbortConfig	Allows you to create criteria to abort a job.
criteriaList	AbortCriteria	The list of abort criteria to define rules to abort the job.
action	java class: java.lang.String (CANCEL)	The type of abort action to initiate a job abort.
failureType	java class: java.lang.String (FAILED   REJECTED   TIMED_OUT   ALL)	The type of job execution failure to define a rule to initiate a job abort.

Name	Type	Description
minNumberOfExecutedThings	java class: java.lang.Integer)	Minimum number of executed things before evaluating an abort rule.
thresholdPercentage	java class: java.lang.Double)	The threshold as a percentage of the total number of executed things that initiate a job abort.  AWS IoT supports up to two digits after the decimal (for example, 10.9 and 10.99, but not 10.999).
createdAt	timestamp	The time, in seconds since the epoch, when the job was created.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job was last updated.
completedAt	timestamp	The time, in seconds since the epoch, when the job was completed.
jobProcessDetails	JobProcessDetails	Details about the job process.
processingTargets	list  member: ProcessingTargetName  java class: java.util.List	The devices on which the job is executing.
ProcessingTargetName	string	
numberOfCanceledThings	integer  java class: java.lang.Integer	The number of things that canceled the job.
numberOfSucceededThings	integer  java class: java.lang.Integer	The number of things that successfully completed the job.
numberOfFailedThings	integer  java class: java.lang.Integer	The number of things that failed executing the job.
numberOfRejectedThings	integer  java class: java.lang.Integer	The number of things that rejected the job.
numberOfQueuedThings	integer  java class: java.lang.Integer	The number of things that are awaiting execution of the job.

Name	Type	Description
numberOfInProgressThings	integer java class: java.lang.Integer	The number of things currently executing the job.
numberOfRemovedThings	integer java class: java.lang.Integer	The number of things that are no longer scheduled to execute the job because they have been deleted or have been removed from the group that was a target of the job.
numberOfTimedOutThings	integer java class: java.lang.Integer	The number of things whose job execution status is <b>TIMED_OUT</b> .
documentParameters	map key: ParameterKey value: ParameterValue	The parameters specified for the job document.
ParameterKey	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
ParameterValue	string  length max:1024 min:1  pattern: [^\p{C}]+	
timeoutConfig	TimeoutConfig	<p>Specifies the amount of time each device has to finish its execution of the job. A timer is started when the job execution status is set to <b>IN_PROGRESS</b>. If the job execution status is not set to another terminal state before the timer expires, it is set to <b>TIMED_OUT</b>.</p> <p><b>Note</b> The job timeout feature isn't currently available in the PDT (us-gov-west-1) Region.</p>

Name	Type	Description
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The in-progress timer can't be updated and applies to all job executions for the job. Whenever a job execution remains in the IN_PROGRESS status for longer than this interval, the job execution fails and switches to the terminal TIMED_OUT status.

## MQTT (7)

Not available.

## DescribeJobExecution

DescribeJobExecution command

Gets details of a job execution. The job's execution status must be SUCCEEDED or FAILED.

### HTTPS (8)

Request:

```
GET /things/thingName/jobs/jobId?executionNumber=executionNumber
```

**jobId**

The unique identifier you assigned to this job when it was created.

**thingName**

The thing name associated with the device the job execution is running on.

**executionNumber**

Optional. A number that is used to specify a job execution on a device. (See [JobExecution \(p. 383\)](#).) If not specified, the latest job execution is returned.

Response:

```
{
  "execution": { JobExecution }
}
```

**execution**

A [JobExecution \(p. 383\)](#) object.

## CLI (8)

### Synopsis:

```
aws iot describe-job-execution \
  --job-id <value> \
  --thing-name <value> \
  [--execution-number <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "jobId": "string",
  "thingName": "string",
  "executionNumber": long
}
```

### cli-input-json fields:

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the thing on which the job execution is running.
executionNumber	long  java class: java.lang.Long	A string (consisting of the digits "0" through "9") that is used to specify a particular job execution on a particular device.

### Output:

```
{
  "execution": {
    "approximateSecondsBeforeTimedOut": "number"
    "jobId": "string",
    "status": "string",
    "forceCanceled": boolean,
    "statusDetails": {
      "detailsMap": {
        "string": "string"
      }
    },
    "thingArn": "string",
    "queuedAt": "timestamp",
    "startedAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "executionNumber": long,
```

```
        "versionNumber": long
    }
}
```

**cli output fields:**

Name	Type	Description
execution	JobExecution	Information about the job execution.
approximateSecondsBeforeTimedOut	long	The estimated number of seconds that remain before the job execution status is changed to <code>TIMED_OUT</code> . The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The actual job execution timeout can occur up to 60 seconds later than the estimated duration. This value is not included if the job execution has reached a terminal status.
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to the job when it was created.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution (IN_PROGRESS, QUEUED, FAILED, SUCCEEDED, TIMED_OUT, CANCELED, or REJECTED).
forceCanceled	boolean  java class: java.lang.Boolean	Is <code>true</code> if the job execution was canceled with the optional <code>force</code> parameter set to <code>true</code> .
statusDetails	JobExecutionStatusDetails	A collection of name-value pairs that describe the status of the job execution.
detailsMap	map  key: DetailsKey  value: DetailsValue	The job execution status.
DetailsKey	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	
DetailsValue	string	

Name	Type	Description
	length max:1024 min:1 pattern: [^\p{C}]*+	
thingArn	string	The ARN of the thing on which the job execution is running.
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long java class: java.lang.Long	A string (consisting of the digits "0" through "9") that identifies this job execution on this device. It can be used in commands that return or update job execution information.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.

#### MQTT (8)

Not available.

### GetJobDocument

GetJobDocument command

Gets the job document for a job.

**Note**

Placeholder URLs are not replaced with presigned Amazon S3 URLs in the document returned. Presigned URLs are generated only when the AWS IoT Jobs service receives a request over MQTT.

#### HTTPS (9)

Request:

```
GET /jobs/jobId/job-document
```

*jobId*

The unique identifier you assigned to this job when it was created.

Response:

```
{  
    "document": "string"  
}
```

document

The job document content.

CLI (9)

**Synopsis:**

```
aws iot get-job-document \  
    --job-id <value> \  
    [--cli-input-json <value>] \  
    [--generate-cli-skeleton]
```

cli-input-json format:

```
{  
    "jobId": "string"  
}
```

**cli-input-json fields:**

Name	Type	Description
jobId	string length max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.

Output:

```
{  
    "document": "string"  
}
```

**cli output fields:**

Name	Type	Description
document	string length max:32768	The job document content.

MQTT (9)

Not available.

## ListJobExecutionsForJob

### ListExecutionsForJob command

Gets a list of job executions for a job.

HTTPS (10)

Request:

```
GET /jobs/jobId/things?status=status&maxResults=maxResults&nextToken=nextToken
```

**jobId**

The unique identifier you assigned to this job when it was created.

**status**

Optional. A filter that lets you search for jobs that have the specified status: QUEUED, IN\_PROGRESS, SUCCEEDED, FAILED, TIMED\_OUT, REJECTED, REMOVED, or CANCELED.

**maxResults**

Optional. The maximum number of results to be returned per request.

**nextToken**

Optional. The token to retrieve the next set of results.

Response:

```
{  
    "executionSummaries": [ JobExecutionSummary ... ]  
}
```

**executionSummaries**

A list of [JobExecutionSummary \(p. 384\)](#) objects associated with the specified job ID.

CLI (10)

**Synopsis:**

```
aws iot list-job-executions-for-job \  
  --job-id <value> \  
  [--status <value>] \  
  [--max-results <value>] \  
  [--next-token <value>] \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

**cli-input-json format:**

```
{  
    "jobId": "string",  
    "status": "string",  
    "maxResults": "integer",  
    "nextToken": "string"  
}
```

**cli-input-json fields:**

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job.
maxResults	integer  java class: java.lang.Integer  range- max:250 min:1	The maximum number of results to be returned per request.
nextToken	string	The token to retrieve the next set of results.

**Output:**

```
{
  "executionSummaries": [
    {
      "thingArn": "string",
      "jobExecutionSummary": {
        "status": "string",
        "queuedAt": "timestamp",
        "startedAt": "timestamp",
        "lastUpdatedAt": "timestamp",
        "executionNumber": long
      }
    }
  ],
  "nextToken": "string"
}
```

**cli output fields:**

Name	Type	Description
executionSummaries	list  member: JobExecutionSummaryForJob  java class: java.util.List	A list of job execution summaries.
JobExecutionSummaryForJob	JobExecutionSummaryForJob	
thingArn	string	The ARN of the thing on which the job execution is running.

Name	Type	Description
jobExecutionSummary	JobExecutionSummary	Contains a subset of information about a job execution.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution.
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long  java class: java.lang.Long	A string (consisting of the digits "0" through "9") that identifies this job execution on this device. It can be used later in commands that return or update job execution information.
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

## MQTT (10)

Not available.

## ListJobExecutionsForThing

### ListJobExecutionsForThing command

Gets a list of job executions for a thing.

## HTTPS (11)

Request:

```
GET /things/thingName/jobs?status=status&maxResults=maxResults&nextToken=nextToken
```

**thingName**

The name of the thing for which JobExecutions will be listed.

**status**

An optional filter that lets you search for jobs that have the specified status: QUEUED, IN\_PROGRESS, SUCCEEDED, FAILED, TIMED\_OUT, REJECTED, REMOVED, or CANCELED.

**maxResults**

The maximum number of results to be returned per request.

**nextToken**

The token for the next set of results, or `null` if there are no additional results.

**Response:**

```
{
    "executionSummaries": [ JobExecutionSummary ... ]
}
```

**executionSummaries**

A list of the [JobExecutionSummary \(p. 384\)](#) objects for the job executions associated with the specified thing.

**CLI (11)**

**Synopsis:**

```
aws iot list-job-executions-for-thing \
    --thing-name <value> \
    [--status <value>] \
    [--max-results <value>] \
    [--next-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
    "thingName": "string",
    "status": "string",
    "maxResults": "integer",
    "nextToken": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The thing name.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT	An optional filter that lets you search for jobs that have the specified status.

Name	Type	Description
	REJECTED   REMOVED   CANCELED	
maxResults	integer  java class: java.lang.Integer  range- max:250 min:1	The maximum number of results to be returned per request.
nextToken	string	The token to retrieve the next set of results.

Output:

```
{
  "executionSummaries": [
    {
      "jobId": "string",
      "jobExecutionSummary": {
        "status": "string",
        "queuedAt": "timestamp",
        "startedAt": "timestamp",
        "lastUpdatedAt": "timestamp",
        "executionNumber": long
      }
    }
  ],
  "nextToken": "string"
}
```

**CLI output fields:**

Name	Type	Description
executionSummaries	list  member: JobExecutionSummaryForThing  java class: java.util.List	A list of job execution summaries.
JobExecutionSummaryForThing	JobExecutionSummaryForThing	
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
jobExecutionSummary	JobExecutionSummary	Contains a subset of information about a job execution.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT	The status of the job execution.

Name	Type	Description
	REJECTED   REMOVED   CANCELED	
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long  java class: java.lang.Long	A string (consisting of the digits "0" through "9") that identifies this job execution on this device. It can be used later in commands that return or update job execution information.
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

## MQTT (11)

Not available.

## ListJobs

### ListJobs command

Gets a list of the jobs in your AWS account.

### HTTPS (12)

Request:

```
GET /jobs?
status=status&targetSelection=targetSelection&thingGroupName=thingGroupName&thingGroupId=thingGroup
```

**status**

Optional. A filter that lets you search for jobs that have the specified status: IN\_PROGRESS, CANCELED, or SUCCEEDED.

**targetSelection**

Optional. A filter that lets you search for jobs that have the specified targetSelection value: CONTINUOUS or SNAPSHOT.

**thingGroupName**

Optional. A filter that lets you search for jobs that have the specified thing group name as a target.

`thingGroupId`

Optional. A filter that lets you search for jobs that have the specified thing group ID as a target.

`maxResults`

Optional. The maximum number of results to be returned per request.

`nextToken`

Optional. The token to retrieve the next set of results.

**Response:**

```
{
    "jobs": [ JobSummary ... ],
}
```

`jobs`

A list of [JobSummary \(p. 382\)](#) objects, one for each job in your AWS account that matches the specified filtering criteria.

**CLI (12)**

**Synopsis:**

```
aws iot list-jobs \
[--status <value>] \
[--target-selection <value>] \
[--max-results <value>] \
[--next-token <value>] \
[--thing-group-name <value>] \
[--thing-group-id <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
    "status": "string",
    "targetSelection": "string",
    "maxResults": "integer",
    "nextToken": "string",
    "thingGroupName": "string",
    "thingGroupId": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
<code>status</code>	string  enum: IN_PROGRESS   CANCELED   SUCCEEDED	An optional filter that lets you search for jobs that have the specified status.
<code>targetSelection</code>	string  enum: CONTINUOUS   SNAPSHOT	Specifies whether the job continues to run (CONTINUOUS), or is complete after all those things specified

Name	Type	Description
		as targets have completed the job (SNAPSHOT). If continuous, the job can also be run on a thing when a change is detected in a target. For example, a job runs on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.
maxResults	integer  java class: java.lang.Integer  range- max:250 min:1	The maximum number of results to return per request.
nextToken	string	The token to retrieve the next set of results.
thingGroupName	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	A filter that limits the returned jobs to those for the specified group.
thingGroupId	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	A filter that limits the returned jobs to those for the specified group.

Output:

```
{
  "jobs": [
    {
      "jobArn": "string",
      "jobId": "string",
      "thingGroupId": "string",
      "targetSelection": "string",
      "status": "string",
      "createdAt": "timestamp",
      "lastUpdatedAt": "timestamp",
      "completedAt": "timestamp"
    }
  ],
  "nextToken": "string"
}
```

**cli output fields:**

Name	Type	Description
jobs	list  member: JobSummary	A list of jobs.

Name	Type	Description
	java class: java.util.List	
JobSummary	JobSummary	
jobArn	string	The job ARN.
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingGroupId	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the thing group.
targetSelection	string  enum: CONTINUOUS   SNAPSHOT	Specifies whether the job continues to run (CONTINUOUS), or is complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous, the job can also be run on a thing when a change is detected in a target. For example, a job runs on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.
status	string  enum: IN_PROGRESS   CANCELED   SUCCEEDED	The job summary status.
createdAt	timestamp	The time, in seconds since the epoch, when the job was created.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job was last updated.
completedAt	timestamp	The time, in seconds since the epoch, when the job completed.
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

## MQTT (12)

Not available.

## UpdateJob

### UpdateJob command

Updates supported fields of the specified job. Updated values for `timeoutConfig` take effect for only newly in-progress executions. Currently in-progress executions continue to execute with the old timeout configuration.

HTTPS (13)

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
    }
}
```

`jobId`

A job identifier that must be unique for your AWS account. We recommend using a UUID. Alphanumeric characters, "-", and "\_" can be used here.

`description`

Optional. A short text description of the job.

`presignedUrlConfigData`

Optional. Configuration information for presigned Amazon S3 URLs.

`roleArn`

The ARN of the IAM role that contains permissions to access the Amazon S3 bucket. This is the bucket that contains the data that devices download with the presigned Amazon S3 URLs. This role must also grant AWS IoT permission to assume the role. For more information, see [Create Jobs \(p. 358\)](#).

`expiresInSec`

How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.

`jobExecutionRolloutConfig`

Optional. Allows you to create a staged rollout of a job.

`maximumPerMinute`

The maximum number of things on which the job is sent for execution, per minute. Valid values are 1 to 1000. If not specified, the default is 1000. The actual number of things that receive the job might be less during any particular minute interval (due to system latency), but are not more than the specified value.

`exponentialRate`

Allows you to create an exponential rate of rollout for a job.

`baseRatePerMinute`

The minimum number of things that are notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.

`incrementFactor`

The exponential factor to increase the rate of rollout for a job.

`rateIncreaseCriteria`

The criteria to initiate the increase in rate of rollout for a job. Set values for either the `numberOfNotifiedThings` or `numberOfSucceededThings`, but not both.

`numberOfNotifiedThings`

The threshold for number of notified things that initiate the increase in rate of rollout.

`numberOfSucceededThings`

The threshold for number of succeeded things that initiate the increase in rate of rollout.

`abortConfig`

Optional. Details of abort criteria to abort the job.

`criteriaList`

The list of abort criteria to define rules to abort the job.

`action`

The type of abort action to initiate a job abort.

`failureType`

The type of job execution failure to define a rule to initiate a job abort.

`minNumberOfExecutedThings`

Minimum number of executed things before evaluating an abort rule.

`thresholdPercentage`

The threshold as a percentage of the total number of executed things that initiate a job abort.

#### `timeoutConfig`

Optional. Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to `IN_PROGRESS`. If the job execution status is not set to another terminal state before the time expires, it is set to `TIMED_OUT`.

#### `inProgressTimeoutInMinutes`

Specifies the amount of time, in minutes, this device has to finish execution of this job. A timer is started, or restarted, whenever this job's execution status is specified as `IN_PROGRESS` with this field populated. If the job execution status is not set to a terminal state before the timer expires, or before another job execution status update is sent with this field populated, the status is set to `TIMED_OUT`.

#### Response:

```
HTTP/1.1 200
```

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

CLI (13)

#### Synopsis:

```
aws iot update-job \
--job-id <value> \
[--description <value>] \
[--presigned-url-config <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format:

```
{
  "description": "string",
  "presignedUrlConfig": {
    "expiresInSec": number,
    "roleArn": "string"
  },
  "jobExecutionsRolloutConfig": {
    "exponentialRate": {
      "baseRatePerMinute": number,
      "incrementFactor": number,
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": number,
        "numberOfSucceededThings": number
      }
    },
    "maximumPerMinute": number
  },
  "abortConfig": {
    "criteriaList": [
      {
        "action": "string",
        "failureType": "string",
        "minNumberOfExecutedThings": number,
        "thresholdPercentage": number
      }
    ]
  },
}
```

```

    "timeoutConfig": {
        "inProgressTimeoutInMinutes": number
    }
}

```

**cli-input-json fields:**

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	A job identifier that must be unique for your AWS account. We recommend using a UUID. Alphanumeric characters, "-" and "_" are valid for use here.
description	string  length max:2028  pattern: [^\\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration information for presigned S3 URLs.
roleArn	string  length max:2048 min:20	The ARN of an IAM role that grants permission to download files from the S3 bucket where the job data or updates are stored. The role must also grant permission for AWS IoT to download the files.
expiresInSec	long  java class: java.lang.Long  range- max:3600 min:60	How long (in seconds) presigned URLs are valid. Valid values are 60 - 3600. The default value is 3600 seconds. Presigned URLs are generated when the AWS IoT Jobs service receives an MQTT request for the job document.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of the job.
maximumPerMinute	integer  java class: java.lang.Integer  range- max:1000 min:1	The maximum number of things that are notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	java class: java.lang.Integer	The minimum number of things that are notified of a pending job, per minute at the start of job rollout. This

Name	Type	Description
		parameter allows you to define the initial rate of rollout.
incrementFactor	java class: java.lang.Double	The exponential factor to increase the rate of rollout for a job.
rateIncreaseCriteria	RateIncreaseCriteria	Allows you to define a criteria to initiate the increase in rate of rollout for a job. Set a value for either <code>numberOfNotifiedThings</code> or <code>numberOfSucceededThings</code> , but not both.
numberOfNotifiedThings	java class: java.lang.Double	The threshold for number of notified things that initiate the increase in rate of rollout.
numberOfSucceededThings	java class: java.lang.Double	The threshold for number of succeeded things that initiate the increase in rate of rollout.
abortConfig	AbortConfig	Allows you to create criteria to abort a job.
criteriaList	AbortCriteria	The list of abort criteria to define rules to abort the job.
action	java class: java.lang.String (CANCEL)	The type of abort action to initiate a job abort.
failureType	java class: java.lang.String (FAILED   REJECTED   TIMED_OUT   ALL)	The type of job execution failure to define a rule to initiate a job abort.
minNumberOfExecutedThings	java class: java.lang.Integer	Minimum number of executed things before evaluating an abort rule.
thresholdPercentage	java class: java.lang.Double	The threshold as a percentage of the total number of executed things that initiate a job abort.  AWS IoT supports up to two digits after the decimal (for example, 10.9 and 10.99, but not 10.999).

Name	Type	Description
timeoutConfig	TimeoutConfig	<p>Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to <code>IN_PROGRESS</code>. If the job execution status is not set to another terminal state before the time expires, it is set to <code>TIMED_OUT</code>.</p> <p><b>Note</b>            The job timeout feature isn't currently available in the PDT (<code>us-gov-west-1</code>) Region.</p>
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. A timer is started, or restarted, whenever this job's execution status is specified as <code>IN_PROGRESS</code> with this field populated. If the job execution status is not set to a terminal state before the timer expires, or before another job execution status update is sent with this field populated, the status is set to <code>TIMED_OUT</code> .
documentParameters	map  key: ParameterKey  value: ParameterValue	Parameters for the job document.
ParameterKey	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
ParameterValue	string  length max:1024 min:1  pattern: [^\p{C}]+	

Output:

```
HTTP/1.1 200
```

If the action is successful, the service sends back an HTTP 200 response with an empty HTTP body.

## MQTT (13)

Not available.

# Jobs Device MQTT and HTTPS APIs

## Device MQTT and HTTPS Data Types

The following data types are used to communicate with the AWS IoT Jobs service over the MQTT and HTTPS protocols.

### JobExecution

#### JobExecution data type

Contains data about a job execution.

##### syntax (7)

```
{  
    "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  
}
```

##### description (7)

###### jobId

The unique identifier you assigned to this job when it was created.

###### thingName

The name of the thing that is executing the job.

###### jobDocument

The content of the job document.

###### status

The status of the job execution. Can be one of: QUEUED, IN\_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED\_OUT, REJECTED, or REMOVED.

###### statusDetails

A collection of name-value pairs that describe the status of the job execution.

###### queuedAt

The time, in seconds since the epoch, when the job execution was enqueued.

`startedAt`

The time, in seconds since the epoch, when the job execution was started.

`lastUpdatedAt`

The time, in seconds since the epoch, when the job execution was last updated.

`versionNumber`

The version of the job execution. Job execution versions are incremented each time they are updated by a device.

`executionNumber`

A number that identifies a job execution on a device. It can be used later in commands that return or update job execution information.

## JobExecutionState

JobExecutionState data type

Contains data about the state of a job execution.

syntax (8)

```
{  
    "status": "QUEUED|IN_PROGRESS|FAILED|SUCCEEDED|CANCELED|TIMED_OUT|REJECTED|  
    REMOVED",  
    "statusDetails": {  
        "string": "string"  
        ...  
    }  
    "versionNumber": "number"  
}
```

description (8)

`status`

The status of the job execution. Can be one of: QUEUED, IN\_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED\_OUT, REJECTED, or REMOVED.

`statusDetails`

A collection of name-value pairs that describe the status of the job execution.

`versionNumber`

The version of the job execution. Job execution versions are incremented each time they are updated by a device.

## JobExecutionSummary

JobExecutionSummary data type

Contains a subset of information about a job execution.

syntax (9)

```
{  
    "jobId": "string",
```

```
        "queuedAt": timestamp,  
        "startedAt": timestamp,  
        "lastUpdatedAt": timestamp,  
        "versionNumber": "number",  
        "executionNumber": long  
    }
```

description (9)

**jobId**

The unique identifier you assigned to this job when it was created.

**queuedAt**

The time, in seconds since the epoch, when the job execution was enqueued.

**startedAt**

The time, in seconds since the epoch, when the job execution started.

**lastUpdatedAt**

The time, in seconds since the epoch, when the job execution was last updated.

**versionNumber**

The version of the job execution. Job execution versions are incremented each time the AWS IoT Jobs service receives an update from a device.

**executionNumber**

A number that identifies a job execution on a device.

## ErrorResponse

ErrorResponse data type

Contains information about an error that occurred during an AWS IoT Jobs service operation.

syntax (10)

```
{  
    "code": "ErrorCode",  
    "message": "string",  
    "clientToken": "string",  
    "timestamp": timestamp,  
    "executionState": JobExecutionState  
}
```

description (10)

**code**

ErrorCode can be set to:

**InvalidTopic**

The request was sent to a topic in the AWS IoT Jobs namespace that does not map to any API.

**InvalidJson**

The contents of the request could not be interpreted as valid UTF-8-encoded JSON.

#### InvalidRequest

The contents of the request were invalid. For example, this code is returned when an `UpdateJobExecution` request contains invalid status details. The message contains details about the error.

#### InvalidStateTransition

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state `SUCCEEDED` to state `IN_PROGRESS`). In this case, the body of the error message also contains the `executionState` field.

#### ResourceNotFound

The `JobExecution` specified by the request topic does not exist.

#### VersionMismatch

The expected version specified in the request does not match the version of the job execution in the AWS IoT Jobs service. In this case, the body of the error message also contains the `executionState` field.

#### InternalError

There was an internal error during the processing of the request.

#### RequestThrottled

The request was throttled.

#### TerminalStateReached

Occurs when a command to describe a job is performed on a job that is in a terminal state.

#### message

An error message string.

#### clientToken

An arbitrary string used to correlate a request with its reply.

#### timestamp

The time, in seconds since the epoch.

#### executionState

A [JobExecutionState \(p. 435\)](#) object. This field is included only when the `code` field has the value `InvalidStateTransition` or `VersionMismatch`. This makes it unnecessary in these cases to perform a separate `DescribeJobExecution` request to obtain the current job execution status data.

## Device Commands

The following commands are available over the MQTT and HTTPS protocols.

### GetPendingJobExecutions

#### GetPendingJobExecutions command

Gets the list of all jobs for a thing that are not in a terminal state.

## MQTT (12)

To invoke this API, publish a message on `$aws/things/thingName/jobs/get`.

Request payload:

```
{ "clientToken": "string" }
```

**clientToken**

Optional. A client token used to correlate requests and responses. Enter an arbitrary value here and it is reflected in the response.

To receive the response, subscribe to:

- `$aws/things/thingName/jobs/get/accepted` and
- `$aws/things/thingName/jobs/get/rejected` or
- `$aws/things/thingName/jobs/get/#` for both.

Response payload:

```
{
  "inProgressJobs" : [ JobExecutionSummary ... ],
  "queuedJobs" : [ JobExecutionSummary ... ],
  "timestamp" : 1489096425069,
  "clientToken" : "client-001"
}
```

**inProgressJobs**

A list of [JobExecutionSummary \(p. 435\)](#) objects with status IN\_PROGRESS.

**queuedJobs**

A list of [JobExecutionSummary \(p. 435\)](#) objects with status QUEUED.

**clientToken**

A client token used to correlate requests and responses.

**timestamp**

The time, in seconds since the epoch, when the message was sent.

## HTTPS (12)

Request:

```
GET /things/thingName/jobs
```

**thingName**

The name of the thing associated with the device.

Response:

```
{
    "inProgressJobs" : [ JobExecutionSummary ... ],
    "queuedJobs" : [ JobExecutionSummary ... ]
}
```

#### inProgressJobs

A list of [JobExecutionSummary \(p. 435\)](#) objects.

#### queuedJobs

A list of [JobExecutionSummary \(p. 435\)](#) objects.

### CLI (12)

#### Synopsis:

```
aws iot-jobs-data get-pending-job-executions \
    --thing-name <value> \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

#### cli-input-json format:

```
{
    "thingName": "string"
}
```

#### cli-input-json fields:

Name	Type	Description
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing that is executing the job.

#### Output:

```
{
    "inProgressJobs": [
        {
            "jobId": "string",
            "queuedAt": long,
            "startedAt": long,
            "lastUpdatedAt": long,
            "versionNumber": long,
            "executionNumber": long
        }
    ],
    "queuedJobs": [
        {
            "jobId": "string",
            "queuedAt": long,
            "startedAt": long,
            "lastUpdatedAt": long,
            "versionNumber": long,
            "executionNumber": long
        }
    ]
}
```

```

        "executionNumber": long
    }
]
}
}
```

**cli output fields:**

Name	Type	Description
inProgressJobs	list  member: JobExecutionSummary  java class: java.util.List	A list of JobExecutionSummary objects with status IN_PROGRESS.
JobExecutionSummary	JobExecutionSummary	
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long  java class: java.lang.Long	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time the AWS IoT Jobs service receives an update from a device.
executionNumber	long  java class: java.lang.Long	A number that identifies a job execution on a device.
queuedJobs	list  member: JobExecutionSummary  java class: java.util.List	A list of JobExecutionSummary objects with status QUEUED.
JobExecutionSummary	JobExecutionSummary	
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.

Name	Type	Description
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long java class: java.lang.Long	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time the AWS IoT Jobs service receives an update from a device.
executionNumber	long java class: java.lang.Long	A number that identifies a job execution on a device.

## StartNextPendingJobExecution

StartNextPendingJobExecution command

Gets and starts the next pending job execution for a thing (status IN\_PROGRESS or QUEUED).

- Any job executions with status IN\_PROGRESS are returned first.
- Job executions are returned in the order in which they were created.
- If the next pending job execution is QUEUED, its state is changed to IN\_PROGRESS and the job execution's status details are set as specified.
- If the next pending job execution is already IN\_PROGRESS, its status details are not changed.
- If no job executions are pending, the response does not include the execution field.
- You can optionally create a step timer by setting a value for the stepTimeoutInMinutes property. If you don't update the value of this property by running UpdateJobExecution, the job execution times out when the step timer expires.

MQTT (13)

To invoke this API, publish a message on \$aws/things/*thingName*/jobs/start-next.

Request payload:

```
{
    "statusDetails": {
        "string": "job-execution-state"
        ...
    },
    "stepTimeoutInMinutes": long,
    "clientToken": "string"
}
```

**statusDetails**

A collection of name-value pairs that describe the status of the job execution. If not specified, the `statusDetails` are unchanged.

**stepTimeOutInMinutes**

Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by calling `UpdateJobExecution`, setting the status to `IN_PROGRESS` and specifying a new timeout value in field `stepTimeoutInMinutes`) the job execution status is set to `TIMED_OUT`. Setting this timeout has no effect on that job execution timeout that might have been specified when the job was created (`CreateJob` using the `timeoutConfig` field).

**clientToken**

A client token used to correlate requests and responses. Enter an arbitrary value here and it is reflected in the response.

To receive the response, subscribe to:

- `$aws/things/thingName/jobs/start-next/accepted` and
- `$aws/things/thingName/jobs/start-next/rejected` or
- `$aws/things/thingName/jobs/start-next/#` for both.

Response payload:

```
{  
    "execution" : JobExecutionData,  
    "timestamp" : timestamp,  
    "clientToken" : "string"  
}
```

**execution**

A [JobExecution \(p. 434\)](#) object. For example:

```
{  
    "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,  
}
```

**timestamp**

The time, in milliseconds since the epoch, when the message was sent to the device.

**clientToken**

A client token used to correlate requests and responses.

## HTTPS (13)

Request:

```
PUT /things/thingName/jobs/$next
{
    "statusDetails": {
        "string": "string"
        ...
    },
    "stepTimeoutInMinutes": long
}
```

**thingName**

The name of the thing associated with the device.

**statusDetails**

A collection of name-value pairs that describe the status of the job execution. If not specified, the **statusDetails** are unchanged.

**stepTimeOutInMinutes**

Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by calling `UpdateJobExecution`, setting the status to `IN_PROGRESS` and specifying a new timeout value in field `stepTimeoutInMinutes`) the job execution status is set to `TIMED_OUT`. Setting this timeout has no effect on that job execution timeout that might have been specified when the job was created (`CreateJob` using the `timeoutConfig` field).

Response:

```
{
    "execution" : JobExecution
}
```

**execution**

A [JobExecution](#) (p. 434) object. For example:

```
{
    "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,
}
```

## CLI (13)

**Synopsis:**

```
aws iot-jobs-data start-next-pending-job-execution \
--thing-name <value> \
[--step-timeout-in-minutes <value>] \
[--status-details <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "thingName": "string",
  "statusDetails": {
    "string": "string"
  },
  "stepTimeoutInMinutes": long
}
```

**cli-input-json fields:**

Name	Type	Description
thingName	string length max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name of the thing associated with the device.
statusDetails	map key: DetailsKey value: DetailsValue	A collection of name-value pairs that describe the status of the job execution. If not specified, the statusDetails are unchanged.
stepTimeOutInMinutes	long	Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by calling <a href="#">UpdateJobExecution</a> , setting the status to <a href="#">IN_PROGRESS</a> and specifying a new timeout value in field <code>stepTimeoutInMinutes</code> ) the job execution status is set to <a href="#">TIMED_OUT</a> . Setting this timeout has no effect on that job execution timeout that might have been specified when the job was created ( <a href="#">CreateJob</a> using the <code>timeoutConfig</code> field).
DetailsKey	string length max:128 min:1 pattern: [a-zA-Z0-9:_-]+	

Name	Type	Description
DetailsValue	string  length max:1024 min:1  pattern: [^\p{C}]*+	

Output:

```
{
  "execution": {
    "jobId": "string",
    "thingName": "string",
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "queuedAt": long,
    "startedAt": long,
    "lastUpdatedAt": long,
    "versionNumber": long,
    "executionNumber": long,
    "jobDocument": "string"
  }
}
```

**cli output fields:**

Name	Type	Description
execution	JobExecution	A JobExecution object.
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing that is executing the job.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution. Can be one of: QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED_OUT, REJECTED, or REMOVED.
statusDetails	map  key: DetailsKey  value: DetailsValue	A collection of name-value pairs that describe the status of the job execution.
DetailsKey	string	

Name	Type	Description
	length max:128 min:1 pattern: [a-zA-Z0-9_-]+	
DetailsValue	string  length max:1024 min:1  pattern: [^\p{C}]*+	
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long  java class: java.lang.Long	The time, in seconds since the epoch, when the job execution was started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
executionNumber	long  java class: java.lang.Long	A number that identifies a job execution on a device. It can be used later in commands that return or update job execution information.
jobDocument	string  length max:32768	The content of the job document.

## DescribeJobExecution

DescribeJobExecution command

Gets detailed information about a job execution.

You can set the `jobId` to `$next` to return the next pending job execution for a thing (status `IN_PROGRESS` or `QUEUED`).

MQTT (14)

To invoke this API, publish a message on `$aws/things/thingName/jobs/jobId/get`.

Request payload:

```
{
  "executionNumber": long,
  "includeJobDocument": boolean,
  "clientToken": "string"
}
```

`thingName`

The name of the thing associated with the device.

`jobId`

The unique identifier assigned to this job when it was created.

Or use `$next` to return the next pending job execution for a thing (status IN\_PROGRESS or QUEUED). In this case, any job executions with status IN\_PROGRESS are returned first. Job executions are returned in the order in which they were created.

`executionNumber`

Optional. A number that identifies a job execution on a device. If not specified, the latest job execution is returned.

`includeJobDocument`

Optional. Unless set to `false`, the response contains the job document. The default is `true`.

`clientToken`

A client token used to correlate requests and responses. Enter an arbitrary value here and it is reflected in the response.

To receive the response, subscribe to:

- `$aws/things/thingName/jobs/jobId/get/accepted` and
- `$aws/things/thingName/jobs/jobId/get/rejected` or
- `$aws/things/thingName/jobs/jobId/get/#` for both.

Response payload:

```
{  
    "execution" : JobExecutionData,  
    "timestamp": "timestamp",  
    "clientToken": "string"  
}
```

`execution`

A [JobExecution \(p. 434\)](#) object.

`timestamp`

The time, in seconds since the epoch, when the message was sent.

`clientToken`

A client token used to correlate requests and responses.

## HTTPS (14)

The job's execution status must be QUEUED or IN\_PROGRESS.

Request:

```
GET /things/thingName/jobs/jobId?  
executionNumber=executionNumber&includeJobDocument=includeJobDocument
```

`thingName`

The name of the thing associated with the device.

`jobId`

The unique identifier assigned to this job when it was created.

Or use `$next` to return the next pending job execution for a thing (status IN\_PROGRESS or QUEUED). In this case, any job executions with status IN\_PROGRESS are returned first. Job executions are returned in the order in which they were created.

`includeJobDocument`

Optional. Unless set to `false`, the response contains the job document. The default is `true`.

`executionNumber`

Optional. A number that identifies a job execution on a device. If not specified, the latest job execution is returned.

**Response:**

```
{
    "execution" : JobExecution,
}
```

`execution`

A [JobExecution \(p. 434\)](#) object.

## CLI (14)

The job's execution status must be QUEUED or IN\_PROGRESS.

### Synopsis:

```
aws iot-jobs-data describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format:

```
{
    "jobId": "string",
    "thingName": "string",
    "includeJobDocument": boolean,
    "executionNumber": long
}
```

### **cli-input-json** fields:

Name	Type	Description
<code>jobId</code>	string  pattern: [a-zA-Z0-9_-]+ +\$next	The unique identifier assigned to this job when it was created, or <code>\$next</code> to return the next

Name	Type	Description
		pending job execution for a thing (status IN_PROGRESS or QUEUED). In this case, any job executions with status IN_PROGRESS are returned first. Job executions are returned in the order in which they were created.
thingName	string length max:128 min:1 pattern: [a-zA-Z0-9:_]+	The thing name associated with the device the job execution is running on.
includeJobDocument	boolean java class: java.lang.Boolean	Optional. Unless set to false, the response contains the job document. The default is true.
executionNumber	long java class: java.lang.Long	Optional. A number that identifies a job execution on a device. If not specified, the latest job execution is returned.

Output:

```
{
  "execution": {
    "jobId": "string",
    "thingName": "string",
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "queuedAt": long,
    "startedAt": long,
    "lastUpdatedAt": long,
    "versionNumber": long,
    "executionNumber": long,
    "jobDocument": "string"
  }
}
```

**cli output fields:**

Name	Type	Description
execution	JobExecution	Contains data about a job execution.
jobId	string length max:64 min:1 pattern: [a-zA-Z0-9:_]+	The unique identifier you assigned to this job when it was created.
thingName	string	The name of the thing that is executing the job.

Name	Type	Description
	length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution. Can be one of: QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED_OUT, REJECTED, or REMOVED.
statusDetails	map  key: DetailsKey  value: DetailsValue	A collection of name-value pairs that describe the status of the job execution.
DetailsKey	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
DetailsValue	string  length max:1024 min:1  pattern: [^\p{C}]*+	
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long  java class: java.lang.Long	The time, in seconds since the epoch, when the job execution was started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
executionNumber	long  java class: java.lang.Long	A number that identifies a job execution on a device. It can be used later in commands that return or update job execution information.
jobDocument	string  length max:32768	The content of the job document.

## UpdateJobExecution

### UpdateJobExecution command

Updates the status of a job execution. You can optionally create a step timer by setting a value for the `stepTimeoutInMinutes` property. If you don't update the value of this property by running `UpdateJobExecution` again, the job execution times out when the step timer expires.

MQTT (15)

To invoke this API, publish a message on `$aws/things/thingName/jobs/jobID/update`.

Request payload:

```
{  
    "status": "job-execution-state",  
    "statusDetails": {  
        "string": "string"  
        ...  
    },  
    "expectedVersion": "number",  
    "executionNumber": long,  
    "includeJobExecutionState": boolean,  
    "includeJobDocument": boolean,  
    "stepTimeoutInMinutes": long,  
    "clientToken": "string"  
}
```

**status**

The new status for the job execution (IN\_PROGRESS, FAILED, SUCCEEDED, or REJECTED). This must be specified on every update.

**statusDetails**

A collection of name-value pairs that describe the status of the job execution. If not specified, the `statusDetails` are unchanged.

**expectedVersion**

The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in the AWS IoT Jobs service does not match, the update is rejected with a `VersionMismatch` error, and an [ErrorResponse \(p. 436\)](#) that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate `DescribeJobExecution` request to obtain the job execution status data.)

**executionNumber**

Optional. A number that identifies a job execution on a device. If not specified, the latest job execution is used.

**includeJobExecutionState**

Optional. When included and set to `true`, the response contains the `JobExecutionState` field. The default is `false`.

**includeJobDocument**

Optional. When included and set to `true`, the response contains the `JobDocument`. The default is `false`.

**stepTimeoutInMinutes**

Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by

again calling `UpdateJobExecution`, setting the status to `IN_PROGRESS` and specifying a new timeout value in this field) the job execution status is set to `TIMED_OUT`. Setting or resetting this timeout has no effect on the job execution timeout that might have been specified when the job was created (by using `CreateJob` with the `timeoutConfig`).

`clientToken`

A client token used to correlate requests and responses. Enter an arbitrary value here and it is reflected in the response.

To receive the response, subscribe to:

- `$aws/things/thingName/jobs/jobId/update/accepted` and
- `$aws/things/thingName/jobs/jobId/update/rejected` or
- `$aws/things/thingName/jobs/jobId/update/#` for both.

Response payload:

```
{  
    "executionState": JobExecutionState,  
    "jobDocument": "string",  
    "timestamp": timestamp,  
    "clientToken": "string"  
}
```

`executionState`

A [JobExecutionState \(p. 435\)](#) object.

`jobDocument`

A [job document \(p. 354\)](#) object.

`timestamp`

The time, in seconds since the epoch, when the message was sent.

`clientToken`

A client token used to correlate requests and responses.

## HTTPS (15)

Request:

```
POST /things/thingName/jobs/jobId  
{  
    "status": "job-execution-state",  
    "statusDetails": {  
        "string": "string"  
        ...  
    },  
    "expectedVersion": "number",  
    "includeJobExecutionState": boolean,  
    "includeJobDocument": boolean,  
    "stepTimeoutInMinutes": long,  
    "executionNumber": long  
}
```

`thingName`

The name of the thing associated with the device.

`jobId`

The unique identifier assigned to this job when it was created.

`status`

The new status for the job execution (IN\_PROGRESS, FAILED, SUCCEEDED, or REJECTED). This must be specified on every update.

`statusDetails`

Optional. A collection of name-value pairs that describe the status of the job execution. If not specified, the `statusDetails` are unchanged.

`expectedVersion`

Optional. The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in the AWS IoT Jobs service does not match, the update is rejected with a `VersionMismatch` error, and an [ErrorResponse \(p. 436\)](#) that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate `DescribeJobExecution` request to obtain the job execution status data.)

`includeJobExecutionState`

Optional. When included and set to `true`, the response contains the `JobExecutionState` data. The default is `false`.

`includeJobDocument`

Optional. When set to `true`, the response contains the job document. The default is `false`.

`stepTimeoutInMinutes`

Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by again calling `UpdateJobExecution`, setting the status to IN\_PROGRESS and specifying a new timeout value in this field) the job execution status is set to TIMED\_OUT. Setting or resetting this timeout has no effect on the job execution timeout that might have been specified when the job was created (by using `CreateJob` with the `timeoutConfig`).

`executionNumber`

Optional. A number that identifies a job execution on a device.

Response:

```
{  
    "executionState": JobExecutionState,  
    "jobDocument": "string"  
}
```

`executionState`

A [JobExecutionState \(p. 435\)](#) object.

`jobDocument`

The contents of the [job document \(p. 354\)](#).

## CLI (15)

### Synopsis:

```
aws iot-jobs-data update-job-execution \
    --job-id <value> \
    --thing-name <value> \
    --status <value> \
    [--status-details <value>] \
    [--expected-version <value>] \
    [--include-job-execution-state | --no-include-job-execution-state] \
    [--include-job-document | --no-include-job-document] \
    [--execution-number <value>] \
    [--cli-input-json <value>] \
    [--step-timeout-in-minutes <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
    "jobId": "string",
    "thingName": "string",
    "status": "string",
    "statusDetails": {
        "string": "string"
    },
    "stepTimeoutInMinutes": number,
    "expectedVersion": long,
    "includeJobExecutionState": boolean,
    "includeJobDocument": boolean,
    "executionNumber": long
}
```

**cli-input-json** fields:

Name	Type	Description
jobId	string  length max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier assigned to this job when it was created.
thingName	string  length max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the thing associated with the device.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The new status for the job execution (IN_PROGRESS, FAILED, SUCCEEDED, or REJECTED). This must be specified on every update.
statusDetails	map  key: DetailsKey  value: DetailsValue	Optional. A collection of name-value pairs that describe the status of the job execution. If

Name	Type	Description
		not specified, the statusDetails are unchanged.
DetailsKey	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
DetailsValue	string  length max:1024 min:1  pattern: [^\p{C}]*+	
stepTimeoutInMinutes  long  Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by again calling <code>UpdateJobExecution</code> , setting the status to <code>IN_PROGRESS</code> and specifying a new timeout value in this field) the job execution status is set to <code>TIMED_OUT</code> . Setting or resetting this timeout has no effect on the job execution timeout that might have been specified when the job was created (by using <code>CreateJob</code> with the <code>timeoutConfig</code> ).		
expectedVersion	long  java class: java.lang.Long	Optional. The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in the AWS IoT Jobs service does not match, the update is rejected with a <code>VersionMismatch</code> error, and an <code>ErrorResponse</code> that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate <code>DescribeJobExecution</code> request to obtain the job execution status data.)

Name	Type	Description
includeJobExecutionState	boolean java class: java.lang.Boolean	Optional. When included and set to true, the response contains the JobExecutionState data. The default is false.
includeJobDocument	boolean java class: java.lang.Boolean	Optional. When set to true, the response contains the job document. The default is false.
executionNumber	long java class: java.lang.Long	Optional. A number that identifies a job execution on a device.

Output:

```
{
  "executionState": {
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "versionNumber": long
  },
  "jobDocument": "string"
}
```

#### cli output fields:

Name	Type	Description
executionState	JobExecutionState	A JobExecutionState object.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution. Can be one of: QUEUED, IN_PROGRESS, FAILED, SUCCEEDED, CANCELED, TIMED_OUT, REJECTED, or REMOVED.
statusDetails	map  key: DetailsKey  value: DetailsValue	A collection of name-value pairs that describe the status of the job execution.
DetailsKey	string  length max:128 min:1  pattern: [a-zA-Z0-9:_-]+	
DetailsValue	string  length max:1024 min:1  pattern: [^\p{C}]*+	

Name	Type	Description
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
jobDocument	string length max:32768	The contents of the job documents.

## JobExecutionsChanged

JobExecutionsChanged message

Sent whenever a job execution is added to or removed from the list of pending job executions for a thing.

MQTT (16)

Topic: \$aws/things/*thingName*/jobs/notify

Message payload:

```
{
  "jobs" : [
    "JobExecutionState": [ JobExecutionSummary \(p. 384\) ... ]
  ],
  "timestamp": timestamp,
}
```

HTTPS (16)

Not available.

CLI (16)

Not available.

## NextJobExecutionChanged

NextJobExecutionChanged message

Sent whenever there is a change to which job execution is next on the list of pending job executions for a thing, as defined for [DescribeJobExecution \(p. 446\)](#) with jobId \$next. This message is not sent when the next job's execution details change, only when the next job that would be returned by [DescribeJobExecution](#) with jobId \$next has changed. Consider job executions J1 and J2 with state QUEUED. J1 is next on the list of pending job executions. If the state of J2 is changed to IN\_PROGRESS while the state of J1 remains unchanged, then this notification is sent and contains details of J2.

MQTT (17)

Topic: \$aws/things/*thingName*/jobs/notify-next

Message payload:

```
{
  "execution" : JobExecution \(p. 383\),
```

```
    "timestamp": timestamp,  
}
```

#### HTTPS (17)

Not available.

#### CLI (17)

Not available.

## Job Rollout and Abort Configuration

AWS IoT jobs can be deployed using variable rollout rates as various criteria and thresholds are met. Job rollouts can also be aborted if the number of failed jobs matches a set of criteria. These rollout configurations give you more granular control over a job's blast radius. Job rollout rate criteria are set at job creation through the [JobExecutionsRolloutConfig](#) object. Job abort criteria are set at job creation through the [AbortConfig](#) object.

## Using Job Rollout Rates

You set a job rollout rate by configuring the [ExponentialRolloutRate](#) property of the [JobExecutionsRolloutConfig](#) object when you run the [CreateJob](#) API. The following example sets a variable rollout rate by using the `exponentialRate` parameter.

```
{  
...  
    "jobExecutionsRolloutConfig": {  
        "exponentialRate": {  
            "baseRatePerMinute": 50,  
            "incrementFactor": 2,  
            "rateIncreaseCriteria": {  
                "numberOfNotifiedThings": 1000, // Set one or the other  
                "numberOfSucceededThings": 1000 // of these two values.  
            },  
            "maximumPerMinute": 1000  
        }  
    }  
}
```

The `baseRatePerMinute` parameter specifies the rate at which the jobs are executed until the `numberOfNotifiedThings` or `numberOfSucceededThings` threshold has been met.

The `incrementFactor` parameter specifies the exponential factor by which the rollout rate increases after the `numberOfNotifiedThings` or `numberOfSucceededThings` threshold has been met.

The `rateIncreaseCriteria` parameter is an object that specifies either the `numberOfNotifiedThings` or `numberOfSucceededThings` threshold.

The `maximumPerMinute` parameter specifies the upper limit of the rate at which job executions can occur. Valid values range from 1 to 1000. This parameter is required when you pass an [ExponentialRate](#) object. In a variable rate rollout, this value establishes the upper limit of a job rollout rate.

A job rollout with the configuration above would start at a rate of 50 job executions per minute. It would continue at that rate until either 1000 things have received job execution notifications (if a value for

`numberOfNotifiedThings` has been specified) or 1000 successful job executions have occurred (if a value for `numberOfSucceededThings` has been specified).

The following table illustrates how the rollout would proceed over the first four increments.

Rollout rate per minute	50	100	200	400
Number of notified devices or successful executions	1000	2000	3000	4000

The following configuration sets a static rollout rate.

```
{
...
"jobExecutionsRolloutConfig": {
    "maximumPerMinute": 1000
}
}
```

The `maximumPerMinute` parameter specifies the upper limit of the rate at which job executions can occur. Valid values range from 1 to 1000. This parameter is optional. If you don't specify a value, the default value of 1000 is used.

## Using Job Rollout Abort Configurations

You set up a job abort condition by configuring the optional `AbortConfig` object when you run the `CreateJob` API. This section describes the effect that the following sample configuration would have on a job rollout that was experiencing multiple failed executions.

```
"abortConfig": {
    "criteriaList": [
        {
            "action": "CANCEL",
            "failureType": "FAILED",
            "minNumberOfExecutedThings": 100,
            "thresholdPercentage": 20
        },
        {
            "action": "CANCEL",
            "failureType": "TIMED_OUT",
            "minNumberOfExecutedThings": 200,
            "thresholdPercentage": 50
        }
    ]
}
```

The `action` parameter specifies the action to take when the abort criteria have been met. This parameter is required, and `CANCEL` is the only valid value.

The `failureType` parameter specifies which failure types should trigger a job abort. Valid values are `FAILED`, `REJECTED`, `TIMED_OUT`, and `ALL`.

The `minNumberOfExecutedThings` parameter specifies the number of completed job executions that must occur before the service checks to see if the job abort criteria have been met. In this example, AWS IoT doesn't check to see if a job abort should occur until at least 100 devices have completed job executions.

The `thresholdPercentage` parameter specifies the total number of executed things that initiate a job abort. In this example, AWS IoT initiates a job abort and cancels the job rollout if at least 20% of all completed executions have failed in any way after 100 executions have completed.

**Note**

Deletion of job executions affects the computation value of the total completed execution. When a job aborts, the service creates an automated comment and `reasonCode` to differentiate a user-driven cancellation from a job abort cancellation.

## Job Limits

For job limit information, see [AWS IoT Job Limits](#) in the AWS General Reference.

# Device Provisioning

AWS IoT device provisioning involves the creation and registration of the following entities:

- A certificate. You can provision a device with an existing certificate or have AWS IoT create and register one for you.
- A policy attached to the certificate.
- A unique identifier for the thing (device).
- A set of attributes for the thing, including existing thing types and groups.

To provision a device, create a template that describes the resources required for your device. Devices require a thing, a certificate, and one or more policies. A *thing* is an entry in the registry that contains attributes that describe the device. Devices use certificates to authenticate with AWS IoT. Policies determine which operations a device can perform in AWS IoT.

Templates contain variables that are replaced when the template is used to provision a device. A dictionary (map) is used to provide values for the variables used in a template. You can use the same template to provision multiple devices. You simply pass in different values for the template variables in the dictionary.

AWS IoT provides three ways to provision devices:

- Single-thing provisioning with a provisioning template.

This is a good option if you only need to provision devices one at a time.

- Just-in-time provisioning (JITP) with a template that registers and provisions a device when it first connects to AWS IoT.

This is a good option if you need to register large numbers of devices, but you don't have information about them that you can assemble into a bulk provisioning list.

- Bulk provisioning.

This option allows you to specify a list of single-thing provisioning template values that are stored in a file in an S3 bucket. This approach works well if you have a large number of known devices whose desired characteristics you can assemble into a list.

Just-in-time provisioning and bulk provisioning are better options if you need to provision large numbers of devices. AWS IoT also provides a [RegisterThing](#) API that you can use to provision single devices programmatically.

## Provisioning Templates

A provisioning template is a JSON document that uses parameters to describe the resources your device must use to interact with AWS IoT. A template contains two sections: **Parameters** and **Resources**.

### Parameters Section

The **Parameters** section declares the parameters used in the **Resources** section. Each parameter declares a name, a type, and an optional default value. The default value is used when the dictionary

passed in with the template does not contain a value for the parameter. The `Parameters` section of a template document looks like the following:

```
{  
    "Parameters" : {  
        "ThingName" : {  
            "Type" : "String"  
        },  
        "SerialNumber" : {  
            "Type" : "String"  
        },  
        "Location" : {  
            "Type" : "String",  
            "Default" : "WA"  
        },  
        "CSR" : {  
            "Type" : "String"  
        }  
    }  
}
```

This template snippet declares four parameters: `ThingName`, `SerialNumber`, `Location`, and `CSR`. All of these parameters are of type `String`. The `Location` parameter declares a default value of "WA".

## Resources Section

The `Resources` section of the template declares the resources required for your device to communicate with AWS IoT: a thing, a certificate, and one or more policies. Each resource specifies a logical name, a type, and a set of properties.

A logical name allows you to refer to a resource elsewhere in the template.

The type specifies the kind of resource you are declaring. Valid types are:

- `AWS::IoT::Thing`
- `AWS::IoT::Certificate`
- `AWS::IoT::Policy`

The properties you specify depend on the type of resource you are declaring.

### Thing Resources

Thing resources are declared using the following properties:

- `ThingName`: String.
- `AttributePayload`: Optional. A list of name/value pairs.
- `ThingTypeName`: Optional. String for an associated thing type for the thing.
- `ThingGroups`: Optional. A list of groups to which the thing belongs.

### Certificate Resources

Certificates can be specified in one of the following ways:

- A certificate signing request (CSR).

- A certificate ID of an existing device certificate.
- A device certificate created with a CA certificate registered with AWS IoT. If you have more than one CA certificate registered with the same subject field, you must also pass in the CA certificate used to sign the device certificate.

**Note**

When you declare a certificate in a template, use only one of these methods. For example, if you use a CSR, you cannot also specify a certificate ID or a device certificate.

For more information, see [AWS IoT and Certificates](#).

Certificate resources are declared using the following properties:

- `CertificateSigningRequest`: String.
- `CertificateID`: String.
- `CertificatePem`: String.
- `CACertificatePem`: String.
- `Status`: Optional. String that can be one of: ACTIVE, INACTIVE, PENDING\_ACTIVATION. Defaults to ACTIVE.

Examples:

- Certificate specified with a CSR:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate",  
        "Properties" : {  
            "CertificateSigningRequest": {"Ref" : "CSR"},  
            "Status" : "ACTIVE"  
        }  
    }  
}
```

- Certificate specified with an existing certificate ID:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate",  
        "Properties" : {  
            "CertificateId": {"Ref" : "CertificateId"}  
        }  
    }  
}
```

- Certificate specified with an existing certificate .pem and CA certificate .pem:

```
{  
    "certificate" : {  
        "Type" : "AWS::IoT::Certificate"  
        "Properties" : {  
            "CACertificatePem": {"Ref" : "CACertificatePem"},  
            "CertificatePem": {"Ref" : "CertificatePem"}  
        }  
    }  
}
```

## Policy Resources

Policy resources are declared using one of the following properties:

- **PolicyName**: Optional. String. Defaults to a hash of the policy document. If you are using an existing IoT policy, enter the name of the policy for the **PolicyName** property, and do not include the **PolicyDocument** property.
- **PolicyDocument**: Optional. A JSON object specified as an escaped string. If **PolicyDocument** is not provided, the policy must already be created.

### Note

If a **Policy** section is present, **PolicyName** or **PolicyDocument**, but not both, must be specified.

## Override Settings

If a template specifies a resource that already exists, the **OverrideSettings** section allows you to specify the action to take:

**DO NOTHING**

Leave the resource as is.

**REPLACE**

Replace the resource with the resource specified in the template.

**FAIL**

Cause the request to fail with a **ResourceConflictsException**.

**MERGE**

Valid only for the **ThingGroups** and **AttributePayload** properties of a thing. Merge the existing attributes or group memberships of the thing with those specified in the template.

When you declare a thing resource, you can specify **OverrideSettings** for the following properties:

- **ATTRIBUTE\_PAYLOAD**
- **THING\_TYPE\_NAME**
- **THING\_GROUPS**

When you declare a certificate resource, you can specify **OverrideSettings** for the **Status** property.

**OverrideSettings** are not available for policy resources.

## Resource Example

The following template snippet declares a thing, a certificate, and a policy:

```
{  
    "Resources" : {  
        "thing" : {  
            "Type" : "AWS::IoT::Thing",  
            "Properties" : {  
                "Name" : "MyThing"  
            }  
        },  
        "cert" : {  
            "Type" : "AWS::IoT::Certificate",  
            "Properties" : {  
                "SubjectFingerprint" : "A1B2C3D4E5F6G7H8I9J0K1L2M3N4O5P6Q7R8S9T0U1V2W3X4Y5Z6",  
                "Subject" : "arn:aws:iot:us-east-1:123456789012:thing/MyThing"  
            }  
        },  
        "policy" : {  
            "Type" : "AWS::IoT::Policy",  
            "Properties" : {  
                "Name" : "MyPolicy",  
                "PolicyDocument" : "...."  
            }  
        }  
    }  
}
```

```
        "ThingName" : {"Ref" : "ThingName"},  
        "AttributePayload" : { "version" : "v1", "serialNumber" : {"Ref" :  
"SerialNumber"}},  
        "ThingType" : {"Ref" : "ThingType"},  
        "ThingGroups" : ["v1-lightbulbs", {"Ref" : "Location"}]  
    },  
    "OverrideSettings" : {  
        "AttributePayload" : "MERGE",  
        "ThingType" : "REPLACE",  
        "ThingGroups" : "DO_NOTHING"  
    }  
},  
"certificate" : {  
    "Type" : "AWS::IoT::Certificate",  
    "Properties" : {  
        "CertificateSigningRequest": {"Ref" : "CSR"},  
        "Status" : "ACTIVE"  
    },  
    "OverrideSettings" : {  
        "Status" : "DO_NOTHING"  
    }  
},  
"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\"] }]}  
    }  
}
```

The thing is declared with:

- The logical name "thing".
  - The type `AWS::IoT::Thing`.
  - A set of thing properties.

The thing properties include the thing name, a set of attributes, an optional thing type name, and an optional list of thing groups to which the thing belongs.

Parameters are referenced by `{"Ref": "<parameter-name>"}`. When the template is evaluated, the parameters are replaced with the parameter's value from the dictionary passed in with the template.

The certificate is declared with:

- The logical name "certificate".
  - The type AWS::IoT::Certificate.
  - A set of properties.

The properties include the CSR for the certificate, and setting the status to ACTIVE. The CSR text is passed as a parameter in the dictionary passed with the template.

The policy is declared with:

- The logical name "policy".
  - The type AWS::IoT::Policy.
  - Either the name of an existing policy or a policy document.

## Template Example

The following JSON file is an example of a complete provisioning template that specifies the certificate with a CSR:

(The `PolicyDocument` field value must be a JSON object specified as an escaped string.)

```
{
    "Parameters" : {
        "ThingName" : {
            "Type" : "String"
        },
        "SerialNumber" : {
            "Type" : "String"
        },
        "Location" : {
            "Type" : "String",
            "Default" : "WA"
        },
        "CSR" : {
            "Type" : "String"
        }
    },
    "Resources" : {
        "thing" : {
            "Type" : "AWS::IoT::Thing",
            "Properties" : {
                "ThingName" : {"Ref" : "ThingName"},
                "AttributePayload" : { "version" : "v1", "serialNumber" : {"Ref" : "SerialNumber"} },
                "ThingTypeName" : "lightBulb-versionA",
                "ThingGroups" : [ "v1-lightbulbs", {"Ref" : "Location"} ]
            }
        },
        "certificate" : {
            "Type" : "AWS::IoT::Certificate",
            "Properties" : {
                "CertificateSigningRequest": {"Ref" : "CSR"},
                "Status" : "ACTIVE"
            }
        },
        "policy" : {
            "Type" : "AWS::IoT::Policy",
            "Properties" : {
                "PolicyDocument" : "{ \"Version\": \"2012-10-17\", \"Statement\": [{ \"Effect\": \"Allow\", \"Action\": [\"iot:Publish\"], \"Resource\": [\"arn:aws:iot:us-east-1:123456789012:topic/foo/bar\"] }] }"
            }
        }
    }
}
```

The following JSON file is an example of a complete provisioning template that specifies an existing certificate with a certificate ID:

```
{
    "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"}
            },
            "OverrideSettings" : {
                "Status" : "DO_NOTHING"
            }
        },
        "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\"] }] }"
            }
        }
    }
}

```

## Programmatic Provisioning

To provision a thing, use the [RegisterThing](#) API or the `register-thing` CLI command. The `register-thing` CLI command takes the following arguments:

`--template-body`

The provisioning template.

`--parameters`

A list of name/value pairs for the parameters used in the provisioning template, in JSON format (for example, `{"ThingName" : "MyProvisionedThing", "CSR" : "<csr-text>"}`).

See [Provisioning Templates \(p. 461\)](#).

[RegisterThing](#) or `register-thing` returns the ARNs for the resources and the text of the certificate it created:

{

```

    "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"
    }
}

```

If a parameter is omitted from the dictionary, the default value is used. If no default value is specified, the parameter is not replaced with a value.

## Just-in-Time Provisioning

You can have your devices provisioned when they first attempt to connect to AWS IoT. Just-in-time provisioning (JITP) settings are made on CA certificates. To provision the device, you must enable automatic registration and associate a provisioning template with the CA certificate used to sign the device certificate.

You can make these settings when you register a CA certificate with the [RegisterCACertificate API](#) or the `register-ca-certificate` CLI command:

```
aws iot register-ca-certificate --ca-certificate <your-ca-cert> --verification-cert
<your-verification-cert> --set-as-active --allow-auto-registration --
registration-config file://<your-template>
```

For more information, see [Registering a CA certificate](#).

You can also use the [UpdateCACertificate API](#) or the `update-ca-certificate` CLI command to update the settings for a CA certificate:

```
$ aws iot update-ca-certificate --cert-id <caCertificateId> --new-auto-registration-status
ENABLE --registration-config file://<your-template>
```

When a device attempts to connect to AWS IoT by using a certificate signed by a registered CA certificate, AWS IoT loads the template from the CA certificate and calls [RegisterThing](#) using the template. The JITP workflow first registers a certificate with a status value of PENDING\_ACTIVATION. When the device provisioning flow is complete, the status of the certificate is changed to ACTIVE.

AWS IoT defines the following parameters that you can declare and reference in provisioning templates:

- `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`

The values for these provisioning template parameters are limited to what JITP can extract from the subject field of the certificate of the device being provisioned. The `AWS::IoT::Certificate::Id`

parameter refers to an internally generated ID, not an ID that is contained in the certificate. You can get the value of this ID using the `principal()` function inside an AWS IoT rule.

The following JSON file is an example of a complete JITP template. The value of the `templateBody` field must be a JSON object specified as an escaped string and can use only the values in the preceding list. You can use a variety of tools to create the stringified JSON object, such as `json.dumps` (Python) or `JSON.stringify` (Node). The value of the `roleARN` field must be the ARN of a role that has the `AWSIoTThingsRegistration` attached to it. Also, your template can use an existing `PolicyName` instead of the inline `PolicyDocument` in the example. (The first example adds line breaks for readability, but you can copy and paste the template that appears directly below it.)

```
{
    "templateBody" : "{
        \r\n        \"Parameters\" : { \r\n            \\"AWS::IoT::Certificate::CommonName\\": { \r\n                \\"Type\\": \"String
        \\"}, \r\n            \\"AWS::IoT::Certificate::SerialNumber\\": { \r\n                \\"Type\\": \"String
        \\"}, \r\n            \\"AWS::IoT::Certificate::Country\\": { \r\n                \\"Type\\": \"String\\\" \r
        \n            }, \r\n            \\"AWS::IoT::Certificate::Id\\": { \r\n                \\"Type\\": \"String\\\" \r\n
        \r\n            }, \r\n            \\"Resources\\": { \r\n                \\"thing\\": { \r\n                    \\"Type\\": \"AWS::IoT::Thing\\\", \r\n                    \\"Properties\\": { \r\n                        \\"ThingName\\": { \r\n                            \\"Ref\\": "
        \"AWS::IoT::Certificate::CommonName\\\" \r\n                        }, \r\n                        \\"AttributePayload\\": { \r\n                            \\"version\\": \"v1\\\", \r\n                            \\"serialNumber\\": { \r\n                                \\"Ref\\": \"AWS::IoT::Certificate::SerialNumber\\\" \r
        \r\n                            }, \r\n                            \\"ThingType\\": \"lightBulb-versionA\\\", \r\n                            \\"ThingGroups\\": [ \r\n                                \"v1-lightbulbs\\\", \r\n                                \\"Ref\\": \"AWS::IoT::Certificate::Country\\\" \r\n                            ], \r\n                            \\"OverrideSettings\\": { \r\n                                \\"AttributePayload\\": \"MERGE\\\", \r\n                                \\"ThingType\\": \"REPLACE\\\", \r\n                                \\"ThingGroups\\": \"DO_NOTHING\\\" \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                                \\"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\""
    }
}
```

}

Version you can copy and paste:

```
{
  "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                \"Properties\": {\r\n                    \"ThingName\": {\r\n                        \"Ref\": \"AWS::IoT::Certificate::CommonName\"\r\n                    },\r\n                    \"version\": \"v1\", \"AttributePayload\": {\r\n                        \"serialNumber\": {\r\n                            \"Ref\": \"AWS::IoT::Certificate::SerialNumber\"\r\n                        },\r\n                            \"ThingType\": \"lightBulb-versionA\", \"v1-lightbulbs\": {\r\n                                \"ThingGroups\": [\r\n                                    {\r\n                                        \"Ref\": \"AWS::IoT::Certificate::Country\"\r\n                                    },\r\n                                    {\r\n                                        \"AttributePayload\": \"MERGE\", \"OverrideSettings\": {\r\n                                            \"ThingType\": \"REPLACE\", \"ThingGroups\": \"DO NOTHING\"\r\n                                        },\r\n                                        \"certificate\": {\r\n                                            \"Type\": \"AWS::IoT::Certificate\", \"Properties\": {\r\n                                                \"CertificateId\": {\r\n                                                    \"Ref\": \"AWS::IoT::Certificate::Id\"\r\n                                                },\r\n                                                \"Status\": \"ACTIVE\", \"OverrideSettings\": {\r\n                                                    \"Status\": \"DO NOTHING\", \"Policy\": {\r\n                                                        \"Type\": \"AWS::IoT::Policy\", \"Properties\": {\r\n                                                            \"PolicyDocument\": \"{\r\n                                                                \"Statement\": [{\r\n                                                                    \"Effect\": \"Allow\", \"Action\": \"iot:Publish\", \"Resource\": \"arn:aws:iot:us-east-1:123456789012:topic/foo/bar\"}\r\n                                                                }\r\n                                                            }\r\n                                                        }\r\n                                                    }\r\n                                                }\r\n                                            }\r\n                                        }\r\n                                    }\r\n                                }\r\n                            }\r\n                        }\r\n                    }\r\n                }\r\n            }\r\n        }\r\n    }\r\n}
```

This sample template declares values for the `AWS::IoT::Certificate::CommonName`, `AWS::IoT::Certificate::SerialNumber`, `AWS::IoT::Certificate::Country` and `AWS::IoT::Certificate::Id` provisioning parameters that are extracted from the certificate and used in the Resources section. The JITP workflow then uses this template to perform the following actions:

- Register a certificate and set its status to PENDING\_ACTIVE.
- Create one thing resource.
- Create one policy resource.
- Attach the policy to the certificate.
- Attach the certificate to the thing.
- Update the certificate status to ACTIVE.

You should be able to see the certificate registration as a logged event (`RegisterCACertificate`) in AWS CloudTrail . You can also use CloudTrail to troubleshoot issues with your JITP template.

# Bulk Provisioning

You can use the [start-thing-registration-task](#) command to provision things in bulk. This command takes a provisioning template, an Amazon S3 bucket name, a key name, and a role ARN that allows access to the file in the Amazon S3 bucket. The file in the Amazon S3 bucket contains the values used to replace the parameters in the template. The file must be a newline-delimited JSON file. Each line contains all of the parameter values for provisioning a single device. For example:

```
{"ThingName": "foo", "SerialNumber": "123", "CSR": "csr1"}  
{"ThingName": "bar", "SerialNumber": "456", "CSR": "csr2"}
```

The following bulk provisioning-related APIs might be useful:

- [ListThingRegistrationTasks](#) – Lists the current bulk thing provisioning tasks.
- [DescribeThingRegistrationTask](#) – Provides information about a specific bulk thing provisioning task.
- [StopThingRegistrationTask](#) – Stops a bulk thing provisioning task.
- [ListThingRegistrationTaskReports](#) – Used to check the results and/or failures for a bulk thing provisioning task.

**Note**

Only one bulk provisioning operation task can run at a time (per account).

# Fleet Indexing Service

Fleet Indexing is a managed service that you can use to index and search your registry data, shadow data, and device connectivity data (device lifecycle events) in the cloud. After you set up your fleet index, the service manages the indexing of updates for your thing groups, thing registries, and device shadows. You can use a simple query language to search across this data. You can also create a [dynamic thing group](#) with a search query.

When you enable indexing, AWS IoT creates an index for your things or thing groups. After it's active, you can run queries on your index, such as finding all devices that are handheld and have more than 70 percent battery life. AWS IoT keeps it continuously updated with your latest data.

`AWS_Things` is the index created for all of your *things*. `AWS_ThingGroups` is the index that contains all of your *thing groups*.

You can use the [AWS IoT console](#) to manage your indexing configuration and run your search queries. Choose the indexes you would like to use in the console settings page. If you prefer programmatic access, you can use the AWS SDKs or the AWS CLI.

For information about pricing this and other services, see the [AWS IoT Device Management Pricing](#) page.

## Managing Thing Indexing

`AWS_Things` is the index created for all of your things. You can control what to index: registry data, shadow data, and device connectivity status data (driven by device lifecycle events).

### Enabling Thing Indexing

You can create the `AWS_Things` index and control its configuration by using the `--thing-indexing-configuration` parameter in the [UpdateIndexingConfiguration](#) API. You can retrieve the current indexing configuration by using the [GetIndexingConfiguration](#) API.

The following command shows how to use the `get-indexing-configuration` CLI command to retrieve the current thing indexing configuration. (In this example, thing indexing is currently disabled.)

```
aws iot get-indexing-configuration
{
    "thingIndexingConfiguration": {
        "thingConnectivityIndexingMode": "OFF"
        "thingIndexingMode": "OFF"
    }
}
```

The following table lists the allowed combinations of `thingIndexingMode` and `thingConnectivityIndexingMode`, and their associated effects. The required `thingIndexingMode` parameter specifies if the `AWS_Things` index contains just registry data or registry and shadow data. The optional `thingConnectivityIndexingMode` parameter specifies whether the index also contains connectivity status data (that is, when devices connected and disconnected).

<code>thingIndexingMode</code>	<code>thingConnectivityIndexingMode</code>	Result
OFF	<i>Not specified.</i>	No indexing or delete an index.

thingIndexingMode	thingConnectivityIndexingMode	Result
OFF	OFF	Equivalent to the previous entry.
REGISTRY	<i>Not specified.</i>	Create or configure the AWS_Things index to index registry data only.
REGISTRY	OFF	Equivalent to the previous entry. (Only registry data is indexed.)
REGISTRY_AND_SHADOW	<i>Not specified.</i>	Create or configure the AWS_Things index to index registry data and shadow data.
REGISTRY_AND_SHADOW	OFF	Equivalent to the previous entry. (Registry data and shadow data are indexed.)
REGISTRY	STATUS	Create or configure the AWS_Things index to index registry data and thing connectivity status data (REGISTRY_AND_CONNECTIVITY_STATUS).
REGISTRY_AND_SHADOW	STATUS	Create or configure the AWS_Things index to index registry data, shadow data, and thing connectivity status data (REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS).

Use the **update-indexing-configuration** CLI command to update the thing indexing configuration. In the following example, you can search registry data, shadow data, and thing connectivity status data using the AWS\_Things index (after it is built, as described in the next section).

```
aws iot update-indexing-configuration --thing-indexing-configuration
    thingIndexingMode=REGISTRY_AND_SHADOW,thingConnectivityIndexingMode=STATUS
```

## Describing a Thing Index

The following command shows you how to use the **describe-index** CLI command to retrieve the current status of the thing index.

```
aws iot describe-index --index-name "AWS_Things"
{
    "indexName": "AWS_Things",
    "indexStatus": "BUILDING",
    "schema": "REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS"
}
```

The first time you enable indexing, AWS IoT builds your index. You can't query the index if `indexStatus` is in the `BUILDING` state. The schema for the things index indicates which type of data (`REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS`) is indexed.

Changing the configuration of your index causes the index to be rebuilt. During this process, the `indexStatus` is `REBUILDING`. You can execute queries on data in the things index while it is being rebuilt. For example, if you change the index configuration from `REGISTRY` to `REGISTRY_AND_SHADOW`,

while the index is being rebuilt, you can query registry data, including the latest updates. However, you can't query the shadow data until the rebuild is complete. The amount of time it takes to build or rebuild the index depends on the amount of data.

## Querying a Thing Index

Use the **search-index** CLI command to query data in the index.

```
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
            }
        },
        {
            "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"
}

```

In the JSON response, "connectivity" (as enabled by the `thingConnectivityIndexingMode=STATUS` setting) provides a Boolean value and a timestamp that indicates if the device is connected to AWS IoT Core. The device "mything1" disconnected (`false`) at POSIX time 1556649874716:

```

"connectivity": {
    "connected": false,
    "timestamp": 1556649874716
}

```

The device "mything2" connected (`true`) at POSIX time 1556649855046:

```

"connectivity": {
    "connected": true,
    "timestamp": 1556649855046
}

```

Timestamps are given in milliseconds since epoch, so 1556649855046 represents 6:44:15.046 PM on Tuesday, April 30, 2019 (GMT).

### Important

If a device has been disconnected for approximately an hour, the connectivity status `"timestamp"` value might be absent. For persistent sessions, the value might be absent after a client has been disconnected longer than the configured time-to-live (TTL) for the persistent session. The connectivity status data is indexed only for connections where the client ID has a matching thing name. (The client ID is the value used to connect a device to AWS IoT Core.)

## Restrictions and Limitations

These are the restrictions and limitations for AWS\_Things.

### Shadow fields with complex types

A shadow field is indexed only if the value of the field is a simple type or an array that consists entirely of simple types. (Simple type means a string, number, or one of the literals `true` or `false`). If a field's value is itself a JSON object, or an array that contains an object, indexing is not performed on that field. For example, given the following shadow state, the value of field `"palette"` is not indexed because it's an array whose items are objects. The value of field `"colors"` is indexed because each value in the array is a string.

```
{

```

```

    "state": {
        "reported": {
            "switched": "ON",
            "colors": [ "RED", "GREEN", "BLUE" ],
            "palette": [
                {
                    "name": "RED",
                    "intensity": 124
                },
                {
                    "name": "GREEN",
                    "intensity": 68
                },
                {
                    "name": "BLUE",
                    "intensity": 201
                }
            ]
        }
    }
}

```

### Shadow metadata

A field in a shadow's metadata section is indexed, but only if the corresponding field in the shadow's "state" section is indexed. (In the previous example, the "palette" field in the shadow's metadata section is not indexed either.)

### Unregistered shadows

If you use [CreateThing](#) to create a shadow using a thing name that hasn't been registered in your AWS IoT account, fields in this shadow are not indexed.

### Numeric values

If any registry or shadow data is recognized by the service as a numeric value, it's indexed as such. You can form queries involving ranges and comparison operators on numeric values (for example, "attribute.foo<5" or "shadow.reported.foo:[75 TO 80]"). To be recognized as numeric, the value of the data must be a valid JSON "number" type literal (an integer in the range -2<sup>53</sup>...2<sup>53</sup>-1 or a double-precision floating point with optional exponential notation) or part of an array that contains only such values.

### Null values

Null values are not indexed.

## Authorization

You can specify the things index as a resource ARN in an AWS IoT policy action, as follows.

Action	Resource
iot:SearchIndex	An index ARN (for example, arn:aws:iot:<your-aws-region>:index/AWS_Things).
iot:DescribeIndex	An index ARN (for example, arn:aws:iot:<your-aws-region>:index/AWS_Things).

**Note**

If you have permissions to query the fleet index, you can access the data of things across the entire fleet.

## Managing Thing Group Indexing

`AWS_ThingGroups` is the index that contains all of your thing groups. You can use this index to search for groups based on group name, description, attributes, and all parent group names.

### Enabling Thing Group Indexing

You can create the `AWS_ThingGroups` index and control its configuration by using the `thing-group-indexing-configuration` setting in the [UpdateIndexingConfiguration](#) API. You can retrieve the current indexing configuration by using the [GetIndexingConfiguration](#) API.

Use the **get-indexing-configuration** CLI command to retrieve the current thing and thing group indexing configurations.

```
aws iot get-indexing-configuration
{
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "ON"
    }
}
```

Use the **update-indexing-configuration** CLI command to update the thing group indexing configurations.

```
aws iot update-indexing-configuration --thing-group-indexing-configuration
    thingGroupIndexingMode=ON
```

**Note**

You can also update configurations for both thing and thing group indexing in a single command, as follows.

```
aws iot update-indexing-configuration --thing-indexing-configuration
    thingIndexingMode=REGISTRY --thing-group-indexing-configuration
    thingGroupIndexingMode=ON
```

The following are valid values for `thingGroupIndexingMode`.

OFF

No indexing/delete index.

ON

Create or configure the `AWS_ThingGroups` index.

### Describing Group Indexes

Use the **describe-index** CLI command to retrieve the current status of the `AWS_ThingGroups` index.

```
aws iot describe-index --index-name "AWS_ThingGroups"
{
```

```
    "indexStatus": "ACTIVE",
    "indexName": "AWS_ThingGroups",
    "schema": "THING_GROUPS"
}
```

The first time you enable indexing, AWS IoT builds your index. You can't query the index if the `indexStatus` is `BUILDING`.

## Querying a Thing Group Index

Use the `search-index` CLI command to query data in the index:

```
aws iot search-index --index-name "AWS_ThingGroups" --query-string
"thingGroupName:mythinggroup*"
```

## Authorization

You can specify the thing groups index as a resource ARN in an AWS IoT policy action, as follows.

Action	Resource
<code>iot:SearchIndex</code>	An index ARN (for example, <code>arn:aws:iot:&lt;your-aws-region&gt;:index/AWS_ThingGroups</code> ).
<code>iot:DescribeIndex</code>	An index ARN (for example, <code>arn:aws:iot:&lt;your-aws-region&gt;:index/AWS_ThingGroups</code> ).

## Getting Statistics About Your Device Fleet

You can use the `get-statistics` CLI command or the `GetStatistics` API to search an index for aggregate data. For example, you might want to find the number of devices that are currently connected to AWS IoT:

```
aws iot get-statistics --index-name AWS_Things --query-string "connectivity.connected:true".
```

This command returns the number of things that have a property called `connectivity.connected` set to `true` in their device shadow:

```
{
  "statistics" : {
    "count" : 1000
  }
}
```

The `get-statistics` CLI command takes the following parameters:

`index-name`

The name of the index to search. The default value is `AWS_Things`.

`query-string`

The query used to search the index. You can specify "\*" to get the count of all indexed things in your AWS account.

#### query-version

The version of the query to use. The default value is 2017-09-30.

The **get-statistics** CLI command returns data in a JSON object. Currently, the only statistic returned is count:

```
{  
  "statistics" : {  
    "count" : 1000  
  }  
}
```

## Query Syntax

Queries are specified using a query syntax.

The query syntax supports the following features.

- Terms and phrases
- Searching fields
- Prefix search
- Range search
- Boolean operators AND, OR, NOT and – (The hyphen is used to exclude something from search results (for example, thingName:(tv\* AND -plasma)))
- Grouping
- Field grouping
- Escaping special characters

The query syntax does not support the following features:

- Leading wildcard search (such as "\*xyz"). (Searching for "\*" matches all things, however.)
- Regular expressions
- Boosting
- Ranking
- Fuzzy searches
- Proximity search
- Sorting
- Aggregation

A few things to note about the query language:

- The default operator is AND. A query for “thingName:abc thingType:xyz” is equivalent to “thingName:abc AND thingType:xyz”.
- If a field isn't specified, AWS IoT searches for the term in all fields.
- All field names are case sensitive.
- Search is case insensitive. Words are separated by whitespace characters as defined by Java's `Character.isWhitespace(int)`.
- Indexing of device shadow data includes reported, desired, delta, and metadata sections.
- Device shadow and registry versions are not searchable, but are present in the response.

- The maximum number of terms in a query is 5.

## Example Thing Queries

Queries are specified in a query string using a query syntax and passed to the [SearchIndex API](#). The following table lists some example query strings.

Query String	Result
abc	Queries for "abc" in any registry or shadow field.
thingName:myThingName	Queries for a thing with name "myThingName".
thingName:my*	Queries for things with names that begin with "my".
thingName:ab?	Queries for things with names that have "ab" plus one additional character, for example: "aba", "abb", "abc", etc.
attributes.myAttribute:75	Queries for things with an attribute named "myAttribute" that has the value 75.
attributes.myAttribute:[75 TO 80]	Queries for things with an attribute named "myAttribute" whose value falls within a numeric range (75–80, inclusive).
attributes.myAttribute:{75 TO 80}	Queries for things with an attribute named "myAttribute" whose value falls within the numeric range (>75 and <=80).
attributes.serialNumber:["abcd" TO "abcf"]	Queries for things with an attribute named "serialNumber" whose value is within an alphanumeric string range. This query returns things with a "serialNumber" attribute with values "abcd", "abce", or "abcf".
attributes.myAttribute:i*t	Queries for things with an attribute named "myAttribute" whose value is 'i', followed by any number of characters, followed by 't'.
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Queries for things that combine terms using Boolean expressions. This query returns things that have an attribute named "attr1" with a value "abc", an attribute named "attr2" that is less than 5, and an attribute named "attr3" that is not greater than 10.
shadow.hasDelta:true	Queries for things whose shadow has a delta element.
NOT attributes.model:legacy	Queries for things where the attribute named "model" is not "legacy".
shadow.reported.stats.battery:{70 TO 100} (v2 OR v3) NOT attributes.model:legacy	Queries for things with the following: <ul style="list-style-type: none"> <li>The thing's shadow stats.battery attribute has a value between 70 and 100.</li> </ul>

Query String	Result
	<ul style="list-style-type: none"> <li>The text "v2" or "v3" occurs in a thing's name, type name, or attribute values.</li> <li>The thing's <code>model</code> attribute is not set to "legacy".</li> </ul>
<code>shadow.reported.myvalues:2</code>	Queries for things where the <code>myvalues</code> array in the shadow's reported section contains a value of 2.
<code>shadow.reported.location:* NOT shadow.desired.stats.battery:*</code>	Queries for things with the following: <ul style="list-style-type: none"> <li>The <code>location</code> attribute exists in the shadow's <code>reported</code> section.</li> <li>The <code>stats.battery</code> attribute does not exist in the shadow's <code>desired</code> section.</li> </ul>
<code>connectivity.connected:true</code>	Queries for all connected devices.
<code>connectivity.connected:false</code>	Queries for all disconnected devices.
<code>connectivity.connected:true AND connectivity.timestamp : [1557651600000 TO 1557867600000]</code>	Queries for all connected devices with a connect timestamp $\geq$ 1557651600000 and $\leq$ 1557867600000. Timestamps are given in milliseconds since epoch.
<code>connectivity.connected:false AND connectivity.timestamp : [1557651600000 TO 1557867600000]</code>	Queries for all disconnected devices with a disconnect timestamp $\geq$ 1557651600000 and $\leq$ 1557867600000. Timestamps are given in milliseconds since epoch.
<code>connectivity.connected:true AND connectivity.timestamp &gt; 1557651600000</code>	Queries for all connected devices with a connect timestamp $>$ 1557651600000. Timestamps are given in milliseconds since epoch.
<code>connectivity.connected:*</code>	Queries for all devices with connectivity information present.

## Example Thing Group Queries

Queries are specified in a query string using a query syntax and passed to the [SearchIndex API](#). The following table lists some example query strings.

Query String	Result
<code>abc</code>	Queries for "abc" in any field.
<code>thingGroupName:myGroupThingName</code>	Queries for a thing group with name "myGroupThingName".
<code>thingGroupName:my*</code>	Queries for thing groups with names that begin with "my".
<code>thingGroupName:ab?</code>	Queries for thing groups with names that have "ab" plus one additional character, for example: "aba", "abb", "abc", etc.

Query String	Result
attributes.myAttribute:75	Queries for thing groups with an attribute named "myAttribute" that has the value 75.
attributes.myAttribute:[75 TO 80]	Queries for thing groups with an attribute named "myAttribute" whose value falls within a numeric range (75–80, inclusive).
attributes.myAttribute:[75 TO 80]	Queries for thing groups with an attribute named "myAttribute" whose value falls within the numeric range (>75 and <=80).
attributes.myAttribute:["abcd" TO "abcf"]	Queries for thing groups with an attribute named "myAttribute" whose value is within an alphanumeric string range. This query returns thing groups with a "serialNumber" attribute with values "abcd", "abce", or "abcf".
attributes.myAttribute:i*t	Queries for thing groups with an attribute named "myAttribute" whose value is 'i', followed by any number of characters, followed by 't'.
attributes.attr1:abc AND attributes.attr2<5 NOT attributes.attr3>10	Queries for thing groups that combine terms using Boolean expressions. This query returns thing groups that have an attribute named "attr1" with a value "abc", an attribute named "attr2" that is less than 5, and an attribute named "attr3" that is not greater than 10.
NOT attributes.myAttribute:cde	Queries for thing groups where the attribute named "myAttribute" is not "cde".
parentGroupNames:(myParentThingGroupName)	Queries for thing groups whose parent group name matches "myParentThingGroupName".
parentGroupNames:(myParentThingGroupName OR myRootThingGroupName)	Queries for thing groups whose parent group name matches "myParentThingGroupName" or "myRootThingGroupName".
parentGroupNames:(myParentThingGroupNa*)	Queries for thing groups whose parent group name begins with "myParentThingGroupNa".

# AWS IoT Device Defender

AWS IoT Device Defender is an IoT security service that allows you to audit the configuration of your devices, monitor connected devices to detect abnormal behavior, and to mitigate security risks. It gives you the ability to enforce consistent security policies across your AWS IoT device fleet and respond quickly when devices are compromised.

IoT fleets may consist of large numbers of devices that have diverse capabilities, are long-lived, and are geographically distributed. These characteristics make fleet setup complex and error-prone. And since devices are often constrained in computational power, memory, and storage capabilities, this limits the use of encryption and other forms of security on the devices themselves. Also, devices often use software with known vulnerabilities. The combination of these factors makes IoT fleets an attractive target for hackers and make it difficult to secure your device fleet on an ongoing basis.

AWS IoT Device Defender addresses these challenges by providing tools to identify security issues and deviations from best practices. AWS IoT Device Defender can audit device fleets to ensure they adhere to security best practices and detect abnormal behavior on devices.

**Note**

AWS IoT Device Defender is not available in the China (Beijing) Region.

## Audit

An AWS IoT Device Defender audit looks at account and device-related settings and policies to ensure security measures are in place. An audit can help you detect any drifts from security best practices or proper access policies, such as multiple devices using the same identity, or overly-permissive policies that allow one device to read and update data for many other devices. You can run audits as needed ("on-demand audits") or schedule them to be run periodically ("scheduled audits").

An AWS IoT Device Defender audit runs a set of pre-defined checks ([Audit checks \(p. 483\)](#)) for common IoT security best practices and device vulnerabilities. Examples of pre-defined checks include policies that grant permission to read or update data on multiple device's, devices that share an identity (X.509 certificate), or certificates that are expiring or have been revoked but are still active.

## Audit checks

**Note**

When a check is enabled, data collection begins immediately. If you have a large amount of data in your account to be collected, then results of the check may not be available for some time after you have enabled it.

The following audit checks are supported:

REVOKED\_CA\_CERT\_CHECK

A CA certificate was revoked but is still active in AWS IoT.

**Severity: Critical**

[more info \(1\)](#)

A CA certificate is marked as revoked in the Certificate Revocation List maintained by the issuing authority, but is still marked as "ACTIVE" or "PENDING\_TRANSFER" in the AWS IoT system.

The following reason codes are returned when this check finds a non-compliant CA certificate:

- CERTIFICATE\_REVOKED\_BY\_ISSUER

#### why it matters (1)

A revoked CA certificate should no longer be used to sign device certificates. It may have been revoked because it was compromised, and newly added devices with certificates signed using this CA certificate might pose a security threat.

#### how to fix it (1)

1. Mark the CA certificate as "INACTIVE" in the AWS IoT system using [UpdateCACertificate](#).
2. Review the device certificate registration activity for the time after the CA certificate was revoked and consider revoking any device certificates that may have been issued with it during this time. (Use [ListCertificatesByCA](#) to list the device certificates signed by the CA certificate and [UpdateCertificate](#) to revoke a device certificate.)

### DEVICE\_CERTIFICATE\_SHARED\_CHECK

Multiple, concurrent connections use the same X.509 certificate to authenticate with the AWS IoT service.

Severity: **Critical**

#### more info (2)

When this check is enabled, data collection begins immediately, but results of the check are not available for at least two hours.

When performed as part of an on-demand audit, this check looks at the certificates and clientIDs that were used by devices to connect during the 31 days prior to the start of the audit. For scheduled audits, this check looks at data from the last time the audit was run to the time this instance of the audit started. If you have taken steps to mitigate this condition during the time checked, note when the concurrent connections were made to determine if the problem persists.

The following reason codes are returned when this check finds a non-compliant certificate:

- CERTIFICATE\_SHARED\_BY\_MULTIPLE\_DEVICES

In addition, the findings returned by this check include the ID of the shared certificate, the IDs of the clients using the certificate to connect, and the connect/disconnect times. Results are listed in order of most recent first.

#### why it matters (2)

Each device should have a unique certificate to authenticate with AWS IoT. When multiple devices use the same certificate this may indicate that a device has been compromised and its identity cloned in order to further compromise the system.

#### how to fix it (2)

Verify that the device certificate has not been compromised. If it has, follow your security best practices to mitigate the situation.

If you are using the same certificate on multiple devices, you may want to:

1. Provision new, unique certificates and attach them to each device.
2. Verify that the new certificates are valid and the devices are able to use them to connect.
3. Mark the old certificate as "REVOLED" in the AWS IoT system using [UpdateCertificate](#)
4. Detach the old certificate from each of the devices.

#### UNAUTHENTICATED\_COGNITO\_ROLE\_OVERLY\_PERMISSIVE\_CHECK

A policy attached to an unauthenticated Cognito identity pool role is considered overly-permissive because it grants permission to perform any of the following AWS IoT actions:

- manage or modify things
- read thing administrative data
- manage non-thing related data or resources

Or, because it grants permission to perform the following AWS IoT actions on a broad set of devices:

- use MQTT to connect/publish/subscribe to reserved topics (including shadow or job execution data)
- use API commands to read or modify shadow or job execution data

In general, devices which connect using an unauthenticated Cognito identity pool role should have only limited permission to publish/subscribe to thing-specific MQTT topics or use the API commands to read/modify thing-specific data related to shadow or job execution data.

**Severity: Critical**

##### manage or modify things (3)

The following AWS IoT API actions are used to manage or modify things so permission to perform these should not be granted to devices connecting via an unauthenticated Cognito identity pool:

- AddThingToThingGroup
- AttachThingPrincipal
- CreateThing
- DeleteThing
- DetachThingPrincipal
- ListThings
- ListThingsInThingGroup
- RegisterThing
- RemoveThingFromThingGroup
- UpdateThing
- UpdateThingGroupsForThing

Any role that grants permission to perform these actions on even a single resource is considered non-compliant.

##### read thing administrative data (3)

The following AWS IoT API actions are used to read or modify thing data so devices connecting via an unauthenticated Cognito identity pool should not be given permission to perform these:

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing
- ListThingPrincipals

**Example:**

- non-compliant:

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

This allows the device to perform the specified actions even though it is granted for only one specific thing.

#### manage non-things (3)

Devices connecting via an unauthenticated Cognito identity pool should not be given permission to perform any other AWS IoT API actions other than those discussed in these sections. In order to manage your account with an application that connects via an unauthenticated Cognito identity pool, create a separate identity pool not used by devices.

#### subscribe/publish to MQTT topics (3)

MQTT messages are sent through the AWS IoT message broker and are used by devices to perform many different actions, including accessing and modifying shadow state and job execution state. A policy that grants permission to a device to connect, publish or subscribe to MQTT messages should restrict these actions to specific resources as follows:

##### Connect

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:client/*
```

The wildcard \* allows any device to connect to AWS IoT.

```
arn:aws:iot:<region>:<account-id>:client/${iot:ClientId}
```

Unless "iot:Connection.Thing.IsAttached" is set to true in the condition keys, this is equivalent to the wildcard \* as in the previous example.

- compliant:

```
{
    "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" }
        }
    ]
}

```

The resource specification contains a variable that matches the device name used to connect, and the condition statement further restricts the permission by checking that the certificate used by the MQTT client matches that attached to the thing with the name used.

#### Publish

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*/shadow/update
```

This allows the device to update the shadow of any device (\* = all devices).

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*
```

This allows the device to read/update/delete the shadow of any device.

- compliant:

```
{
    "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/*"
            ],
        }
    ]
}
```

The resource specification contains a wildcard, but it only matches any shadow-related topic for the device whose thing name is used to connect.

#### Subscribe

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This allows the device to subscribe to reserved shadow or job topics for all devices.

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

The same as the previous example, but using the # wildcard.

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/+/shadow/update
```

This allows the device to see shadow updates on any device (+ = all devices).

- compliant:

```
{
    "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/*"
            ],
        }
    ]
}
```

The resource specifications contain wildcards but they only match any shadow-related topic and any job-related topic for the device whose thing name is used to connect.

#### Receive

- compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This is okay because the device can only receive messages from topics on which it has permission to subscribe.

#### read/modify shadow or job data (3)

A policy that grants permission to a device to perform an API action to access or modify device shadows or job execution data should restrict these actions to specific resources. The API actions are:

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

#### Examples:

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:thing/*
```

This allows the device to perform the specified action on any thing.

- compliant:

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

```
        "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"
        ]
    }
}
```

This allows the device to perform the specified actions on only two specific things.

#### more info (3)

For this check, AWS IoT Device Defender audits all Cognito identity pools that have been used to connect to the AWS IoT message broker during the past 31 days prior to the audit execution. All Cognito identity pools from which either an authenticated or unauthenticated Cognito identity connected are included in the audit.

The following reason codes are returned when this check finds a non-compliant unauthenticated Cognito identity pool role:

- `ALLOWS_ACCESS_TO_IOT_ADMIN_ACTIONS`
- `ALLOWS_BROAD_ACCESS_TO_IOT_DATA_PLANE_ACTIONS`

#### why it matters (3)

Because unauthenticated identities are never authenticated by the user, they pose a much greater risk than authenticated Cognito identities. If an unauthenticated identity is compromised it could use administrative actions to modify account settings, delete resources or gain access to sensitive data. Or, with broad access to device settings, it could access or modify shadows and jobs for all devices in your account. A guest user might use the permissions to compromise your entire fleet or launch a DDOS attack with messages.

#### how to fix it (3)

A policy attached to an unauthenticated Cognito identity pool role should grant only those permissions required for a device to do its job. We recommend the following steps:

1. Create a new compliant role.
2. Create a new Cognito identity pool and attach the compliant role to it.
3. Verify that your identities can access AWS IoT using the new pool.
4. Once verification is complete, attach the new, compliant role to the Cognito identity pool that was flagged as non-compliant.

#### AUTHENTICATED\_COGNITO\_ROLE\_OVERLY\_PERMISSIVE\_CHECK

A policy attached to an authenticated Cognito identity pool role is considered overly-permissive because it grants permission to perform the following AWS IoT actions:

- manage or modify things
- manage non-thing related data or resources

Or, because it grants permission to perform the following AWS IoT actions on a broad set of devices:

- read thing administrative data
- use MQTT to connect/publish/subscribe to reserved topics (including shadow or job execution data)
- use API commands to read or modify shadow or job execution data

In general, devices that connect using an authenticated Cognito identity pool role should have only limited permission to read thing-specific administrative data, publish/subscribe to thing-specific MQTT topics or use the API commands to read/modify thing-specific data related to shadow or job execution data.

Severity: **Critical**

#### manage or modify things (4)

The following AWS IoT API actions are used to manage or modify things so permission to perform these should not be granted to devices connecting via an authenticated Cognito identity pool:

- AddThingToThingGroup
- AttachThingPrincipal
- CreateThing
- DeleteThing
- DetachThingPrincipal
- ListThings
- ListThingsInThingGroup
- RegisterThing
- RemoveThingFromThingGroup
- UpdateThing
- UpdateThingGroupsForThing

Any role that grants permission to perform these actions on even a single resource is considered non-compliant.

#### manage non-things (4)

Devices connecting via an authenticated Cognito identity pool should not be given permission to perform any other AWS IoT API actions other than those discussed in these sections. In order to manage your account with an application that connects via an authenticated Cognito identity pool, create a separate identity pool not used by devices.

#### read thing administrative data (4)

The following AWS IoT API actions are used to read thing data so devices connecting via an authenticated Cognito identity pool should be given permission to perform these on only a limited set of things:

- DescribeThing
- ListJobExecutionsForThing
- ListThingGroupsForThing

- ListThingPrincipals

**Examples:**

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:thing/*
```

This allows the device to perform the specified action on any thing.

- compliant:

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

This allows the device to perform the specified actions on only one specific thing.

- compliant:

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

This is compliant because, although the resource is specified with a wildcard (\*), this is preceded by a specific string, and that limits the set of things accessed to those with names that have the given prefix.

**subscribe/publish to MQTT topics (4)**

MQTT messages are sent through the AWS IoT message broker and are used by devices to perform many different actions, including accessing and modifying shadow state and job execution state. A

policy that grants permission to a device to connect, publish or subscribe to MQTT messages should restrict these actions to specific resources as follows:

#### Connect

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:client/*
```

The wildcard \* allows any device to connect to AWS IoT.

```
arn:aws:iot:<region>:<account-id>:client/${iot:ClientId}
```

Unless "iot:Connection.Thing.IsAttached" is set to true in the condition keys, this is equivalent to the wildcard \* as in the previous example.

- compliant:

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

The resource specification contains a variable that matches the device name used to connect, and the condition statement further restricts the permission by checking that the certificate used by the MQTT client matches that attached to the thing with the name used.

#### Publish

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*/shadow/update
```

This allows the device to update the shadow of any device (\* = all devices).

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*
```

This allows the device to read/update/delete the shadow of any device.

- compliant:

```
{
  "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/*"
    ],
}
]
}

```

The resource specification contains a wildcard, but it only matches any shadow-related topic for the device whose thing name is used to connect.

#### Subscribe

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This allows the device to subscribe to reserved shadow or job topics for all devices.

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/#
```

The same as the previous example, but using the # wildcard.

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/+shadow/update
```

This allows the device to see shadow updates on any device (+ = all devices).

- compliant:

```
{
  "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/*"
      ],
    }
  ]
}
```

The resource specifications contain wildcards but they only match any shadow-related topic and any job-related topic for the device whose thing name is used to connect.

#### Receive

- compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This is okay because the device can only receive messages from topics on which it has permission to subscribe.

#### read/modify shadow or job data (4)

A policy that grants permission to a device to perform an API action to access or modify device shadows or job execution data should restrict these actions to specific resources. The API actions are:

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

#### Examples:

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:thing/*
```

This allows the device to perform the specified action on any thing.

- compliant:

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

This allows the device to perform the specified actions on only two specific things.

#### more info (4)

For this check, AWS IoT Device Defender audits all Cognito identity pools that have been used to connect to the AWS IoT message broker during the past 31 days prior to the audit execution. All Cognito identity pools from which either an authenticated or unauthenticated Cognito identity connected are included in the audit.

The following reason codes are returned when this check finds a non-compliant authenticated Cognito identity pool role:

- 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

#### why it matters (4)

If an authenticated identity is compromised it could use administrative actions to modify account settings, delete resources or gain access to sensitive data.

#### how to fix it (4)

A policy attached to an authenticated Cognito identity pool role should grant only those permissions required for a device to do its job. We recommend the following steps:

1. Create a new compliant role.
2. Create a new Cognito identity pool and attach the compliant role to it.
3. Verify that your identities can access AWS IoT using the new pool.
4. Once verification is complete, attach the new, compliant role to the Cognito identity pool that was flagged as non-compliant.

## IOT\_POLICY\_OVERLY\_PERMISSIVE\_CHECK

An AWS IoT policy gives permissions that are too broad/unrestricted. It grants permission to send or receive MQTT messages for a broad set of devices, or grants permission to access or modify shadow and job execution data for a broad set of devices.

In general, a policy for a device should grant access to very specific resources associated with just that device and no, or very few, other devices. With certain exceptions, using a wildcard (for example, "\*") to specify resources in such a policy is considered too broad/unrestricted.

Severity: **Critical**

#### MQTT permissions (5)

MQTT messages are sent through the AWS IoT message broker and are used by devices to perform many different actions, including accessing and modifying shadow state and job execution state. A policy that grants permission to a device to connect, publish or subscribe to MQTT messages should restrict these actions to specific resources as follows:

##### Connect

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:client/*
```

The wildcard \* allows any device to connect to AWS IoT.

```
arn:aws:iot:<region>:<account-id>:client/${iot:ClientId}
```

Unless "iot:Connection.Thing.IsAttached" is set to true in the condition keys, this is equivalent to the wildcard \* as in the previous example.

- compliant:

```
{
    "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" }
        }
    }
]
}

```

The resource specification contains a variable that matches the device name used to connect, and the condition statement further restricts the permission by checking that the certificate used by the MQTT client matches that attached to the thing with the name used.

#### Publish

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*/shadow/update
```

This allows the device to update the shadow of any device (\* = all devices).

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/*
```

This allows the device to read/update/delete the shadow of any device.

- compliant:

```
{
    "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/*"
            ],
        }
    ]
}
```

The resource specification contains a wildcard, but it only matches any shadow-related topic for the device whose thing name is used to connect.

#### Subscribe

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This allows the device to subscribe to reserved shadow or job topics for all devices.

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

The same as the previous example, but using the # wildcard.

```
arn:aws:iot:<region>:<account-id>:topic/$aws/things/+shadow/update
```

This allows the device to see shadow updates on any device (+ = all devices).

- compliant:

```
{  
    "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/*"  
            ],  
        }  
    ]  
}
```

The resource specifications contain wildcards but they only match any shadow-related topic and any job-related topic for the device whose thing name is used to connect.

#### Receive

- compliant:

```
arn:aws:iot:<region>:<account-id>:topicfilter/$aws/things/*
```

This is okay because the device can only receive messages from topics on which it has permission to subscribe.

#### shadow and job permissions (5)

A policy that grants permission to a device to perform an API action to access or modify device shadows or job execution data should restrict these actions to specific resources. The API actions are:

- DeleteThingShadow
- GetThingShadow
- UpdateThingShadow
- DescribeJobExecution
- GetPendingJobExecutions
- StartNextPendingJobExecution
- UpdateJobExecution

#### Examples:

- non-compliant:

```
arn:aws:iot:<region>:<account-id>:thing/*
```

This allows the device to perform the specified action on any thing.

- compliant:

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

This allows the device to perform the specified actions on only two specific things.

#### more info (5)

The following reason code is returned when this check finds a non-compliant IoT policy:

- ALLOWS\_BROAD\_ACCESS\_TO\_IOT\_DATA\_PLANE\_ACTIONS

#### why it matters (5)

A certificate, Cognito identity or thing group with an overly-permissive policy can, if compromised, impact the security of your entire account. An attacker could use such broad access to read or modify shadows, jobs or job executions for all your devices. Or an attacker could use a compromised certificate to connect malicious devices or launch a DDOS attack on your network.

#### how to fix it (5)

Follow these steps to fix any non-compliant policies attached to things, thing groups, or other entities:

1. Create a new, compliant version of the policy using [CreatePolicyVersion](#) with the "setAsDefault" flag set to true (making this new version operative for all entities that use the policy).
2. Get a list of targets (certificates, thing groups) that the policy is attached to using [ListTargetsForPolicy](#) and determine which devices are included in the groups or which use the certificates to connect.
3. Verify that all associated devices are able to connect to AWS IoT. If a device is unable to connect, roll back the default policy to the previous version using [SetPolicyVersion](#), revise the policy and try again.

Use [AWS IoT Policy Variables](#) to dynamically reference specific AWS IoT resources in your policies.

## CA\_CERT\_APPROACHING\_EXPIRATION\_CHECK

A CA certificate is expiring within 30 days or has expired.

**Severity: Medium**

[more info \(6\)](#)

This check applies to CA certificates that are "ACTIVE" or "PENDING\_TRANSFER".

The following reason codes are returned when this check finds a non-compliant CA certificate:

- CERTIFICATE\_APPROACHING\_EXPIRATION
- CERTIFICATE\_PAST\_EXPIRATION

[why it matters \(6\)](#)

An expired CA certificate should not be used to sign new device certificates.

[how to fix it \(6\)](#)

Consult your security best practices for how to proceed. You may want to:

1. Register a new CA certificate with AWS IoT.
2. Verify that you are able to sign device certificates using the new CA certificate.
3. Mark the old CA certificate as "INACTIVE" in the AWS IoT system using [UpdateCACertificate](#).

## CONFLICTING\_CLIENT\_IDS\_CHECK

Multiple devices connect using the same client ID.

**Severity: High**

[more info \(7\)](#)

Multiple connections were made using the same client ID, causing an already connected device to be disconnected. The MQTT specification allows only one active connection per client ID, so when another device connects using the same client ID, it knocks the previous one off the connection.

When performed as part of an on-demand audit, this check looks at how clientIDs were used to connect during the 31 days prior to the start of the audit. For scheduled audits, this check looks at data from the last time the audit was run to the time this instance of the audit started. If you have taken steps to mitigate this condition during the time checked, note when the connections/disconnections were made to determine if the problem persists.

The following reason codes are returned when this check finds non-compliance:

- DUPLICATE\_CLIENT\_ID\_ACROSS\_CONNECTIONS

In addition, the findings returned by this check include the clientID used to connect, principal IDs, and disconnect times. Results are listed in order of most recent first.

[why it matters \(7\)](#)

Devices with conflicting IDs are forced to constantly reconnect, which may result in lost messages or leave a device unable to connect.

This may indicate that a device or a device's credentials have been compromised, and might be part of a DDoS attack. It is also possible that devices are misconfigured in the account or a device has a bad connection and is forced to reconnect several times per minute.

#### how to fix it (7)

Register each device as a unique thing in AWS IoT, and use the thing name as the client ID to connect. Or use a UUID as the client ID when connecting the device over MQTT.

#### DEVICE\_CERT\_APPROACHING\_EXPIRATION\_CHECK

A device certificate is expiring within 30 days or has expired.

**Severity: Medium**

#### more info (8)

This check applies to device certificates that are "ACTIVE" or "PENDING\_TRANSFER".

The following reason codes are returned when this check finds a non-compliant device certificate:

- CERTIFICATE\_APPROACHING\_EXPIRATION
- CERTIFICATE\_PAST\_EXPIRATION

#### why it matters (8)

A device certificate should not be used after it expires.

#### how to fix it (8)

Consult your security best practices for how to proceed. You may want to:

1. Provision a new certificate and attach it to the device.
2. Verify that the new certificate is valid and the device is able to use it to connect.
3. Mark the old certificate as "INACTIVE" in the AWS IoT system using [UpdateCertificate](#).
4. Detach the old certificate from the device. (See [DetachThingPrincipal](#).)

#### REVOKED\_DEVICE\_CERT\_CHECK

A revoked device certificate is still active.

**Severity: Medium**

#### more info (9)

A device certificate is in its CA's [Certificate revocation list](#) but it is still active in AWS IoT.

This check applies to device certificates that are "ACTIVE" or "PENDING\_TRANSFER".

The following reason codes are returned when this check finds non-compliance:

- CERTIFICATE\_REVOKED\_BY\_ISSUER

#### why it matters (9)

A device certificate is usually revoked because it has been compromised. It is possible that it has not yet been revoked in AWS IoT due to an error or an oversight.

#### how to fix it (9)

Verify that the device certificate has not been compromised. If it has, follow your security best practices to mitigate the situation. You may want to:

1. Provision a new certificate for the device.
2. Verify that the new certificate is valid and the device is able to use it to connect.
3. Mark the old certificate as "REVOKED" in the AWS IoT system using [UpdateCertificate](#).
4. Detach the old certificate from the device. (See [DetachThingPrincipal](#).)

#### LOGGING\_DISABLED\_CHECK

Amazon IoT CloudWatch logs are not enabled.

Severity: **Low**

[more info \(10\)](#)

The following reason codes are returned when this check finds non-compliance:

- LOGGING\_DISABLED

[why it matters \(10\)](#)

Amazon IoT CloudWatch logs provide visibility into behaviors within AWS IoT, including authentication failures and unexpected connects and disconnects, that may indicate that a device has been compromised.

[how to fix it \(10\)](#)

Enable Amazon IoT CloudWatch logs. See [Monitoring Tools](#).

## How To Perform Audits

1. Configure audit settings for your account. Use [UpdateAccountAuditConfiguration \(p. 508\)](#) to enable those checks you want to be available for audits, set up optional notifications, and configure permissions.

For certain checks, AWS IoT begins collecting data as soon as the check is enabled.

2. Create one or more audit schedules. Use [CreateScheduledAudit \(p. 512\)](#) to specify the checks you want to perform during an audit and how often these audits should be run.

Or, you can run an on-demand audit when necessary. Use [StartOnDemandAuditTask \(p. 521\)](#) to specify the checks you want to perform and start an audit running right away. (Results may not be ready for some time if you have recently enabled a check that is included in the on-demand audit.)

3. You can use the [AWS IoT console](#) to view the results of your audits.

Or, you can see the results of your audits with [ListAuditFindings \(p. 528\)](#). With this command, you can filter the results by the type of check, a specific resource, or the time of the audit. Armed with this information, you can take the appropriate steps to mitigate any problems found.

## Notifications

When an audit is completed, an SNS notification can be sent with a summary of the results of each audit check that was performed, including details about the number of non-compliant resources that were found. Use the `auditNotificationTargetConfigurations` field in the input to the [UpdateAccountAuditConfiguration \(p. 508\)](#) command. The SNS notification has the following payload:

[payload example](#)

```
{
    "accountId": "123456789012",
    "taskId": "4e2bcd1ccbc2a5dd15292a82ab80c380",
    "taskStatus": "FAILED|CANCELED|COMPLETED",
    "taskType": "ON_DEMAND_AUDIT_TASK|SCHEDULED_AUDIT_TASK",
    "scheduledAuditName": "myWeeklyAudit",
    "failedChecksCount": 0,
    "canceledChecksCount": 0,
    "nonCompliantChecksCount": 1,
    "compliantChecksCount": 0,
    "totalChecksCount": 1,
    "taskStartTime": 1524740766191,
    "auditDetails": [
        {
            "checkName": "DEVICE_CERT_APPROACHING_EXPIRATION_CHECK | REVOKED_DEVICE_CERT_CHECK | CA_CERT_APPROACHING_EXPIRATION_CHECK | REVOKED_CA_CERT_CHECK | DEVICE_CERTIFICATE_SHARED_CHECK | IOT_POLICY_UNRESTRICTED_CHECK | UNAUTHENTICATED_COGNITO_IDENTITY_UNRESTRICTED_ACCESS_CHECK | AUTHENTICATED_COGNITO_IDENTITY_UNRESTRICTED_ACCESS_CHECK | CONFLICTING_CLIENT_IDS_CHECK | LOGGING_DISABLED_CHECK",
            "checkRunStatus": "FAILED | CANCELED | COMPLETED_COMPLIANT | COMPLETED_NON_COMPLIANT",
            "nonCompliantResourcesCount": 1,
            "totalResourcesCount": 1,
            "message": "optional message if an error occurred",
            "errorCode": "INSUFFICIENT_PERMISSIONS|AUDIT_CHECK_DISABLED"
        }
    ]
}
```

[payload JSON schema](#)

```
{
    "$schema": "http://json-schema.org/draft-07/schema#",
    "$id": "arn:aws:iot:::schema:auditnotification/1.0",
    "type": "object",
    "properties": {
        "accountId": {
            "type": "string"
        },
        "taskId": {
            "type": "string"
        },
        "taskStatus": {
            "type": "string",
            "enum": [
                "FAILED",
                "CANCELED",
                "COMPLETED"
            ]
        },
        "taskType": {
            "type": "string"
        }
    }
}
```

```

    "type": "string",
    "enum": [
        "SCHEDULED_AUDIT_TASK",
        "ON_DEMAND_AUDIT_TASK"
    ]
},
"scheduledAuditName": {
    "type": "string"
},
"failedChecksCount": {
    "type": "integer"
},
"canceledChecksCount": {
    "type": "integer"
},
"nonCompliantChecksCount": {
    "type": "integer"
},
"compliantChecksCount": {
    "type": "integer"
},
"totalChecksCount": {
    "type": "integer"
},
"taskStartTime": {
    "type": "integer"
},
"auditDetails": {
    "type": "array",
    "items": [
        {
            "type": "object",
            "properties": {
                "checkName": {
                    "type": "string",
                    "enum": [
                        "DEVICE_CERT_APPROACHING_EXPIRATION_CHECK",
                        "REVOKED_DEVICE_CERT_CHECK",
                        "CA_CERT_APPROACHING_EXPIRATION_CHECK",
                        "REVOKED_CA_CERT_CHECK",
                        "LOGGING_DISABLED_CHECK"
                    ]
                },
                "checkRunStatus": {
                    "type": "string",
                    "enum": [
                        "FAILED",
                        "CANCELED",
                        "COMPLETED_COMPLIANT",
                        "COMPLETED_NON_COMPLIANT"
                    ]
                }
            },
            "nonCompliantResourcesCount": {
                "type": "integer"
            },
            "totalResourcesCount": {
                "type": "integer"
            },
            "message": {
                "type": "string",
            },
            "errorCode": {
                "type": "string",
                "enum": [
                    "INSUFFICIENT_PERMISSIONS",
                    "AUDIT_CHECK_DISABLED"
                ]
            }
        }
    ]
}

```

```

        ]
    }
},
"required": [
    "checkName",
    "checkRunStatus",
    "nonCompliantResourcesCount",
    "totalResourcesCount"
]
}
],
}
},
"required": [
    "accountId",
    "taskId",
    "taskStatus",
    "taskType",
    "failedChecksCount",
    "canceledChecksCount",
    "nonCompliantChecksCount",
    "compliantChecksCount",
    "totalChecksCount",
    "taskStartTime",
    "auditDetails"
]
}
}

```

Notifications can also be viewed in the AWS IoT console, along with additional information about the device, device statistics (for example, last connection time, number of active connections, data transfer rate), and historical alerts for the device.

## Permissions

This section contains information on how to set up the AWS IAM roles and policies required to create, run and manage AWS IoT Device Defender audits. For more information on AWS IAM, see the [AWS Identity and Access Management User Guide](#).

### Give AWS IoT Device Defender permission to collect your data in order to run an audit.

When you call [UpdateAccountAuditConfiguration \(p. 508\)](#) you must specify an IAM Role with two policies, a permissions policy and a trust policy. The permissions policy grants AWS IoT Device Defender permission to access your account data, using AWS IoT APIs, when it runs an audit. The trust policy grants AWS IoT Device Defender permission to assume the required role.

[permissions policy](#)

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:GetLoggingOptions",
                "iot:GetV2LoggingOptions",
                "iot>ListCACertificates",
                "iot>ListCertificates",
                "iotDescribeCACertificate",
                "iotDescribeCACertificate"
            ]
        }
    ]
}
```

```
    "iot:DescribeCertificate",
    "iot>ListPolicies",
    "iot:GetPolicy",
    "iot:GetEffectivePolicies",
    "cognito-identity:GetIdentityPoolRoles",
    "iam>ListRolePolicies",
    "iam>ListAttachedRolePolicies",
    "iam:GetPolicy",
    "iam:GetPolicyVersion",
    "iam:GetRolePolicy"
],
"Resource":[
    "*"
]
}
]
```

trust policy

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

## Give AWS IoT Device Defender permission to publish notifications to an SNS topic.

If you use the `auditNotificationTargetConfigurations` parameter in [UpdateAccountAuditConfiguration \(p. 508\)](#), you must specify an IAM Role with two policies, a permissions policy and a trust policy. The permissions policy grants permission to AWS IoT Device Defender to publish notifications to your SNS topic. The trust policy grants AWS IoT Device Defender permission to assume the required role.

permissions policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "sns:Publish"
            ],
            "Resource": [
                "arn:aws:sns:region:account-id:your-topic-name"
            ]
        }
    ]
}
```

trust policy

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

## Give IAM users or groups permission to run AWS IoT Device Defender audit commands

To allow IAM users or groups to manage, run, or view the results of AWS IoT Device Defender, you must create and assign roles with attached policies that grant permission to run the appropriate commands. What each specific policy contains depends on what commands you want the user or group to run.

- **UpdateAccountAuditConfiguration**

policy

NOTE: You must create the IAM Role with the attached policy in same account from which this command is run. Cross account access is not allowed. The policy should have "iam:PassRole" permissions (permissions to pass this role).

In the following policy template `audit-permissions-role-arn` is the `roleArn` that you pass to AWS IoT Device Defender in the `UpdateAccountAuditConfiguration` request using the `roleArn` parameter. `audit-notifications-permissions-role-arn` is the `roleArn` that you pass to AWS IoT Device Defender in the `UpdateAccountAuditConfiguration` request using the `auditNotificationTargetConfigurations` parameter.

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iot:UpdateAccountAuditConfiguration"  
            ],  
            "Resource": [  
                "*"  
            ]  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "iam:PassRole"  
            ],  
            "Resource": [  
                "arn:aws:iam::account-id:role/audit-permissions-role-arn",  
                "arn:aws:iam::account-id:role/audit-notifications-permissions-role-arn"  
            ]  
        }  
    ]  
}
```

```
    ]  
}
```

- **DescribeAccountAuditConfiguration**
- **DeleteAccountAuditConfiguration**
- **StartOnDemandAuditTask**
- **CancelAuditTask**
- **DescribeAuditTask**
- **ListAuditTasks**
- **ListScheduledAudits**
- **ListAuditFindings**

[policy](#)

NOTE: all these commands require \* in the Resource field of the policy.

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot:DescribeAccountAuditConfiguration",  
        "iot:DeleteAccountAuditConfiguration",  
        "iot:StartOnDemandAuditTask",  
        "iot:CancelAuditTask",  
        "iot:DescribeAuditTask",  
        "iot>ListAuditTasks",  
        "iot>ListScheduledAudits",  
        "iot>ListAuditFindings"  
      ],  
      "Resource": [  
        "*"  
      ]  
    }  
  ]  
}
```

- **CreateScheduledAudit**
- **UpdateScheduledAudit**
- **DeleteScheduledAudit**
- **DescribeScheduledAudit**

[policy](#)

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "iot>CreateScheduledAudit",  
        "iot:UpdateScheduledAudit",  
        "iot:DeleteScheduledAudit",  
        "iot:DescribeScheduledAudit"  
      ],  
      "Resource": [  
        "*"  
      ]  
    }  
  ]  
}
```

```

    "Resource": [
        "arn:aws:iot:region:account-id:scheduledaudit/scheduled-audit-name"
    ]
}
]
}

```

The format for an AWS IoT Device Defender scheduled audit roleArn is:

```
arn:aws:iot:region:account-id:scheduledaudit/scheduled-audit-name
```

## Service Limits

Resource	Limit	Description
scheduled audits	5 max.	You can create up to 5 scheduled audits before a LimitExceeded Exception occurs.
simultaneous in progress "on-demand" audits	10 max.	You can create up to 10 "on-demand" audits before a LimitExceeded Exception occurs.

## Audit Commands

### Manage Audit Settings

Configure audit settings for your account using `UpdateAccountAuditConfiguration`. This command allows you to enable those checks you want to be available for audits, set up optional notifications, and configure permissions.

Check these settings with `DescribeAccountAuditConfiguration`.

Use `DeleteAccountAuditConfiguration` to delete your audit settings. This restores all default values, and effectively disables audits since all checks are disabled by default.

### UpdateAccountAuditConfiguration

Configures or reconfigures the Device Defender audit settings for this account. Settings include how audit notifications are sent and which audit checks are enabled or disabled.

#### Synopsis

```
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]
```

cli-input-json format

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

### cli-input-json fields

Name	Type	Description
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to AWS IoT to access information about your devices, policies, certificates and other items as necessary when performing an audit.
auditNotificationTargetConfigurations	map	Information about the targets to which audit notifications are sent.
targetArn	string	The ARN of the target (SNS topic) to which audit notifications are sent.
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to send notifications to the target.
enabled	boolean	True if notifications to the target are enabled.
auditCheckConfigurations	map	<p>Specifies which audit checks are enabled and disabled for this account. Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are currently enabled.</p> <p>Note that some data collection may begin immediately when certain checks are enabled. When a check is disabled, any data collected so far in relation to the check is deleted.</p> <p>You cannot disable a check if it is used by any scheduled audit. You must first delete the check</p>

Name	Type	Description
		from the scheduled audit or delete the scheduled audit itself.  On the first call to <code>UpdateAccountAuditConfiguration</code> this parameter is required and must specify at least one enabled check.
<code>enabled</code>	<code>boolean</code>	True if this audit check is enabled for this account.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DescribeAccountAuditConfiguration

Gets information about the Device Defender audit settings for this account. Settings include how audit notifications are sent and which audit checks are enabled or disabled.

#### Synopsis

```
aws iot describe-account-audit-configuration \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

#### Output

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": [
    {
      "string": {
        "targetArn": "string",
        "roleArn": "string",
        "enabled": "boolean"
      }
    },
  ]}
```

```
"auditCheckConfigurations": {
    "string": {
        "enabled": "boolean"
    }
}
```

### CLI output fields

Name	Type	Description
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to AWS IoT to access information about your devices, policies, certificates and other items as necessary when performing an audit.  On the first call to <code>UpdateAccountAuditConfiguration</code> this parameter is required.
auditNotificationTargetConfigurations	map	Information about the targets to which audit notifications are sent for this account.
targetArn	string	The ARN of the target (SNS topic) to which audit notifications are sent.
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to send notifications to the target.
enabled	boolean	True if notifications to the target are enabled.
auditCheckConfigurations	map	Which audit checks are enabled and disabled for this account.
enabled	boolean	True if this audit check is enabled for this account.

### Errors

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DeleteAccountAuditConfiguration

Restores the default settings for Device Defender audits for this account. Any configuration data you entered is deleted and all audit checks are reset to disabled.

### Synopsis

```
aws iot delete-account-audit-configuration \
[--delete-scheduled-audits | --no-delete-scheduled-audits] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "deleteScheduledAudits": "boolean"  
}
```

### cli-input-json fields

Name	Type	Description
deleteScheduledAudits	boolean	If true, all scheduled audits are deleted.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## Schedule Audits

Create one or more scheduled audits using `CreateScheduledAudit`. This command allows you to specify the checks you want to perform during an audit and how often the audit should be run.

Keep track of your scheduled audits with `ListScheduledAudits` and `DescribeScheduledAudit`.

Change an existing scheduled audit with `UpdateScheduledAudit` or delete it with `DeleteScheduledAudit`.

## CreateScheduledAudit

Creates a scheduled audit that is run at a specified time interval.

### Synopsis

```
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]
```

#### cli-input-json format

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "scheduledAuditName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. Can be one of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] [12][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Can be "1" through "31" or "LAST". This field is required if the "frequency" parameter is set to "MONTHLY". If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. Can be one of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT". This field is required if the "frequency" parameter is set to "WEEKLY" or "BIWEEKLY".  enum: SUN   MON   TUE   WED   THU   FRI   SAT

Name	Type	Description
targetCheckNames	list  member: AuditCheckName	Which checks are performed during the scheduled audit. Checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage the scheduled audit.
Key	string	The tag's key.
Value	string	The tag's value.
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name you want to give to the scheduled audit. (Max. 128 chars)

## Output

```
{
  "scheduledAuditArn": "string"
}
```

## CLI output fields

Name	Type	Description
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## ListScheduledAudits

Lists all of your scheduled audits.

### Synopsis

```
aws iot list-scheduled-audits \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time. The default is 25.

### Output

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "string",
      "scheduledAuditArn": "string",
      "frequency": "string",
      "dayOfMonth": "string",
      "dayOfWeek": "string"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
scheduledAudits	list member: ScheduledAuditMetadata java class: java.util.List	The list of scheduled audits.
scheduledAuditName	string length- max:128 min:1	The name of the scheduled audit.

Name	Type	Description
	pattern: [a-zA-Z0-9_-]+	
scheduledAuditArn	string	The ARN of the scheduled audit.
frequency	string	How often the scheduled audit takes place.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] 1[2][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit is run (if the frequency is "MONTHLY"). If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit is run (if the frequency is "WEEKLY" or "BIWEEKLY").  enum: SUN   MON   TUE   WED   THU   FRI   SAT
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

## Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DescribeScheduledAudit

Gets information about a scheduled audit.

### Synopsis

```
aws iot describe-scheduled-audit \
--scheduled-audit-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{
  "scheduledAuditName": "string"
}
```

### cli-input-json fields

Name	Type	Description
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit whose information you want to get.

### Output

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

### CLI output fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. One of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] [12][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Will be "1" through "31" or "LAST". If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. One of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT".  enum: SUN   MON   TUE   WED   THU   FRI   SAT

Name	Type	Description
targetCheckNames	list member: AuditCheckName	Which checks are performed during the scheduled audit. (Note that checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
scheduledAuditName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit.
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### InternalFailureException

An unexpected error has occurred.

## UpdateScheduledAudit

Updates a scheduled audit, including what checks are performed and how often the audit takes place.

### Synopsis

```
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]
```

cli-input-json format

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
```

```

    "targetCheckNames": [
        "string"
    ],
    "scheduledAuditName": "string"
}

```

#### cli-input-json fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. Can be one of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] [12][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Can be "1" through "31" or "LAST". This field is required if the "frequency" parameter is set to "MONTHLY". If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. Can be one of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT". This field is required if the "frequency" parameter is set to "WEEKLY" or "BIWEEKLY".  enum: SUN   MON   TUE   WED   THU   FRI   SAT
targetCheckNames	list  member: AuditCheckName	Which checks are performed during the scheduled audit. Checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit. (Max. 128 chars)

## Output

```
{
  "scheduledAuditArn": "string"
}
```

## CLI output fields

Name	Type	Description
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DeleteScheduledAudit

Deletes a scheduled audit.

### Synopsis

```
aws iot delete-scheduled-audit \
  --scheduled-audit-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "scheduledAuditName": "string"
}
```

### cli-input-json fields

Name	Type	Description
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit you want to delete.

## Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## Run An On-Demand Audit

Use `StartOnDemandAuditTask` to specify the checks you want to perform and start an audit running right away.

### StartOnDemandAuditTask

Starts an on-demand Device Defender audit.

#### Synopsis

```
aws iot start-on-demand-audit-task \
  --target-check-names <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

#### cli-input-json fields

Name	Type	Description
<code>targetCheckNames</code>	list  member: <code>AuditCheckName</code>	Which checks are performed during the audit. The checks you specify must be enabled for your account or an exception occurs. Use <code>DescribeAccountAuditConfiguration</code> to see the list of all checks including those that are enabled or <code>UpdateAccountAuditConfiguration</code> to select which checks are enabled.

## Output

```
{  
    "taskId": "string"  
}
```

## CLI output fields

Name	Type	Description
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9-]+	The ID of the on-demand audit you started.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## Manage Audit Instances

Use `DescribeAuditTask` to get information about a specific audit instance. If it has already run, the results include which checks failed and which passed, those that the system was unable to complete, and if the audit is still in progress, those it is still working on.

Use `ListAuditTasks` to find the audits that were run during a specific time interval.

Use `CancelAuditTask` to halt an audit in progress.

## DescribeAuditTask

Gets information about a Device Defender audit.

### Synopsis

```
aws iot describe-audit-task \  
  --task-id <value> \  
  [--cli-input-json <value>] \  
  [--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
    "taskId": "string"
```

}

### cli-input-json fields

Name	Type	Description
taskId	string length- max:40 min:1 pattern: [a-zA-Z0-9-]+	The ID of the audit whose information you want to get.

### Output

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

### CLI output fields

Name	Type	Description
taskStatus	string	The status of the audit: one of "IN_PROGRESS", "COMPLETED", "FAILED", or "CANCELED".  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED
taskType	string	The type of audit: "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK

Name	Type	Description
taskStartTime	timestamp	The time the audit started.
taskStatistics	TaskStatistics	Statistical information about the audit.
totalChecks	integer	The number of checks in this audit.
inProgressChecks	integer	The number of checks in progress.
waitingForDataCollectionChecks	integer	The number of checks waiting for data collection.
compliantChecks	integer	The number of checks that found compliant resources.
nonCompliantChecks	integer	The number of checks that found non-compliant resources.
failedChecks	integer	The number of checks
canceledChecks	integer	The number of checks that did not run because the audit was canceled.
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit (only if the audit was a scheduled audit).
auditDetails	map	Detailed information about each check performed during this audit.
checkRunStatus	string  enum: IN_PROGRESS   WAITING_FOR_DATA_COLLECTION   CANCELED   COMPLETED_COMPLIANT   COMPLETED_NON_COMPLIANT   FAILED	The completion status of this check, one of "IN_PROGRESS", "WAITING_FOR_DATA_COLLECTION", "CANCELED", "COMPLETED_COMPLIANT", "COMPLETED_NON_COMPLIANT", or "FAILED".  enum: IN_PROGRESS   WAITING_FOR_DATA_COLLECTION   CANCELED   COMPLETED_COMPLIANT   COMPLETED_NON_COMPLIANT   FAILED
checkCompliant	boolean	True if the check completed and found all resources compliant.
totalResourcesCount	long	The number of resources on which the check was performed.

Name	Type	Description
nonCompliantResourcesCount	long	The number of resources that the check found non-compliant.
errorCode	string	The code of any error encountered when performing this check during this audit. One of "INSUFFICIENT_PERMISSIONS", or "AUDIT_CHECK_DISABLED".
message	string length- max:2048	The message associated with any error encountered when performing this check during this audit.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## ListAuditTasks

Lists the Device Defender audits that have been performed during a given time period.

### Synopsis

```
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]
```

`cli-input-json` format

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
```

```

    "nextToken": "string",
    "maxResults": "integer"
}

```

### cli-input-json fields

Name	Type	Description
startTime	timestamp	The beginning of the time period. Note that audit information is retained for a limited time (180 days). Requesting a start time prior to what is retained results in an "InvalidRequestException".
endTime	timestamp	The end of the time period.
taskType	string	A filter to limit the output to the specified type of audit: can be one of "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK
taskStatus	string	A filter to limit the output to audits with the specified completion status: can be one of "IN_PROGRESS", "COMPLETED", "FAILED" or "CANCELED".  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time. The default is 25.

### Output

```

{
  "tasks": [
    {
      "taskId": "string",
      "taskStatus": "string",
      "taskType": "string"
    }
  ],
  "nextToken": "string"
}

```

## CLI output fields

Name	Type	Description
tasks	list  member: AuditTaskMetadata  java class: java.util.List	The audits that were performed during the specified time period.
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9-]+	The ID of this audit.
taskStatus	string  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED	The status of this audit: one of "IN_PROGRESS", "COMPLETED", "FAILED" or "CANCELED".
taskType	string  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK	The type of this audit: one of "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## CancelAuditTask

Cancels an audit that is in progress. The audit can be either scheduled or on-demand. If the audit is not in progress, an "InvalidRequestException" occurs.

### Synopsis

```
aws iot cancel-audit-task \
--task-id <value> \
```

```
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "taskId": "string"
}
```

### cli-input-json fields

Name	Type	Description
taskId	string length- max:40 min:1 pattern: [a-zA-Z0-9-]+	The ID of the audit you want to cancel. You can only cancel an audit that is "IN_PROGRESS".

### Output

None

### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## Check Audit Results

Use `ListAuditFindings` to see the results of an audit. You can filter the results by the type of check, a specific resource, or the time of the audit. Armed with this information, you can take the appropriate steps to mitigate any problems that were found.

### ListAuditFindings

Lists the findings (results) of a Device Defender audit or of the audits performed during a specified time period. (Findings are retained for 180 days.)

#### Synopsis

```
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]
```

#### cli-input-json format

```
{
    "taskId": "string",
    "checkName": "string",
    "resourceIdentifier": {
        "deviceCertificateId": "string",
        "caCertificateId": "string",
        "cognitoIdentityPoolId": "string",
        "clientId": "string",
        "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
        },
        "account": "string"
    },
    "maxResults": "integer",
    "nextToken": "string",
    "startTime": "timestamp",
    "endTime": "timestamp"
}
```

#### cli-input-json fields

Name	Type	Description
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9]+	A filter to limit results to the audit with the specified ID. You must specify either the taskId or the startTime and endTime, but not both.
checkName	string	A filter to limit results to the findings for the specified audit check.
resourceIdentifier	ResourceIdentifier	Information identifying the non-compliant resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.

Name	Type	Description
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
maxResults	integer  range- max:250 min:1	The maximum number of results to return at one time. The default is 25.
nextToken	string	The token for the next set of results.
startTime	timestamp	A filter to limit results to those found after the specified time. You must specify either the startTime and endTime or the taskId, but not both.
endTime	timestamp	A filter to limit results to those found before the specified time. You must specify either the startTime and endTime or the taskId, but not both.

## Output

```
{
  "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",
        "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
        },
        "account": "string"
    },
    "additionalInfo": {
        "string": "string"
    }
}
],
"reasonForNonCompliance": "string",
"reasonForNonComplianceCode": "string"
}
],
"nextToken": "string"
}
}

```

## CLI output fields

Name	Type	Description
findings	list  member: AuditFinding	The findings (results) of the audit.
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9-]+	The ID of the audit that generated this result (finding)
checkName	string	The audit check that generated this result.
taskStartTime	timestamp	The time the audit started.
findingTime	timestamp	The time the result (finding) was discovered.
severity	string  enum: CRITICAL   HIGH   MEDIUM   LOW	The severity of the result (finding).

Name	Type	Description
nonCompliantResource	NonCompliantResource	The resource that was found to be non-compliant with the audit check.
resourceType	string	The type of the non-compliant resource.  enum: DEVICE_CERTIFICATE   CA_CERTIFICATE   IOT_POLICY   COGNITO_IDENTITY_POOL   CLIENT_ID   ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Information identifying the non-compliant resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
additionalInfo	map	Additional information about the non-compliant resource.
relatedResources	list  member: RelatedResource	The list of related resources.

Name	Type	Description
resourceType	string	The type of resource.  enum: DEVICE_CERTIFICATE   CA_CERTIFICATE   IOT_POLICY   COGNITO_IDENTITY_POOL   CLIENT_ID   ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Information identifying the resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
additionalInfo	map	Additional information about the resource.
reasonForNonCompliance	string	The reason the resource was non-compliant.
reasonForNonComplianceCode	string	A code which indicates the reason that the resource was non-compliant.

Name	Type	Description
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## Detect

AWS IoT Device Defender Detect allows you to identify unusual behavior that may indicate a compromised device by monitoring the behavior of your devices. Using a combination of cloud-side metrics (from AWS IoT) and device-side metrics (from agents you install on your devices) you can detect changes in connection patterns, devices that communicate to unauthorized or unrecognized endpoints, and changes in inbound and outbound device traffic patterns. You create security profiles, which contain definitions of expected device behaviors, and assign them to a group of devices or to all the devices in your fleet. AWS IoT Device Defender Detect uses these security profiles to detect anomalies and send alerts via Amazon CloudWatch metrics and Amazon SNS notifications.

AWS IoT Device Defender Detect is capable of detecting a variety of security issues frequently found in connected devices:

- Traffic from a device to a known malicious IP address or to an unauthorized endpoint that indicates a potential malicious command and control channel.
- Anomalous traffic, such as a spike in outbound traffic, that indicates a device is participating in a DDoS.
- Devices with remote management interfaces and ports that are remotely accessible.
- A spike in the rate of messages sent to your account— so that a rogue device does not end up costing you in per-message charges.

#### Use cases:

Measure attack surface

You can use AWS IoT Device Defender Detect to measure the attack surface of your devices. For example, you can identify devices with service ports which are often the target of attack campaigns (telnet service running on ports 23/2323, SSH service running on port 22, HTTP/S services running on ports 80/443/8080/8081). While these service ports might have legitimate reasons to be used on the devices, they are also usually part of the attack surface for adversaries and carry associated risks. After Detect alerts you to the attack surface, you can either decide to minimize it (by eliminating unused network services) or run additional assessments to identify security weaknesses (for example, telnet configured with common, default or weak passwords).

## Detect device behavioral anomalies with possible security root causes

You can use AWS IoT Device Defender Detect to alert you to unexpected device behavioral metrics (the number of open ports, number of connections, an unexpected open port, connections to unexpected IP addresses) which might indicate a security breach. For example, a higher than expected number of TCP connections may indicate a device is being used for a DDoS attack. A process listening on a port other than the one you expect may indicate a backdoor installed on a device for remote control. You can use Detect to probe the health of your device fleets and verify your security assumptions (for example, no device is listening on port 23 or 2323).

## Detect a mis-configured device

A spike in the number or size of messages sent from a device to your account might indicate a mis-configured device. Such a device could increase your per-message charges. Similarly, a device with many authorization failures could require a re-configured policy.

# Concepts

## metric

AWS IoT Device Defender Detect uses metrics to detect anomalous behavior. Detect compares the reported value of a metric with the expected value you provide. These metrics can be taken from two sources: (a) cloud-side metrics, and (b) device-side metrics:

- (a) Abnormal behavior on the AWS IoT network is detected by using cloud-side metrics such as the number of authorization failures, or the number or size of messages a device sends or receives via AWS IoT.
- (b) AWS IoT Device Defender Detect can also collect, aggregate, and monitor metrics data generated by AWS IoT devices, for example, the ports a device is listening on, the number of bytes or packets sent, or the device's TCP connections.

You can use AWS IoT Device Defender Detect with cloud-side metrics alone. To use device-side metrics you must first deploy an agent on your AWS IoT connected devices or device gateways to collect the metrics and send them to AWS IoT. See [Sending Metrics from Devices \(p. 550\)](#)

## security profile

A security profile defines anomalous behaviors for a group of devices (a [thing group](#)) or for all devices in your account, and specifies what actions to take when an anomaly is detected. You can create a security profile and associate it with a group of devices using the console or with AWS IoT API commands. AWS IoT Device Defender Detect starts recording security-related data and uses the behaviors defined in the security profile to detect anomalies in the behavior of the devices.

## behavior

A behavior tells AWS IoT Device Defender Detect how to recognize when a device is doing something abnormal. Each behavior consists of a name, a metric, an operator, and a value or a statistical threshold. For some metrics, a time period (`durationSeconds`) is also required. Any device action that does not match a defined behavior statement triggers an alert.

## alert

When an anomaly is detected, an alert notification can be sent via a CloudWatch metric (see [AWS IoT Metrics \(p. 620\)](#)) or an SNS notification. An alert notification is also displayed in the AWS IoT CDM console along with additional information about the alert, and a history of alerts for the device. An alert is also sent when a monitored device stops exhibiting anomalous behavior or when it had been causing an alert but stops reporting for an extended period.

## Behaviors

A security profile contains a set of behaviors. Each behavior contains a metric that specifies the normal behavior for a group of devices or for all devices in your account (see [Metrics \(p. 537\)](#) and [CreateSecurityProfile \(p. 556\)](#)).

The following describes some of the fields that are used in the definition of a behavior:

**name**

The name you wish to give to the behavior.

**metric**

The name of the metric used. That is, what is measured by the behavior.

**criteria**

The criteria that determine if a device is behaving normally in regard to the **metric**.

**comparisonOperator**

The operator that relates the thing measured (**metric**) to the criteria (**value** or **statisticalThreshold**).

Possible values are: "less-than", "less-than-equals", "greater-than", "greater-than-equals", "in-cidr-set", "not-in-cidr-set", "in-port-set", and "not-in-port-set". Not all operators are valid for every metric-- operators for CIDR sets and ports are only for use with metrics involving such entities.

**value**

The value to be compared with the **metric**. Depending on the type of metric, this should contain a count (**value**), **cids** (a list of CIDRs) or **ports** (a list of ports).

**statisticalThreshold**

The statistical threshold by which a behavior violation is determined. This field contains a **statistic** field which has the following possible values: "p0", "p0.1", "p0.01", "p1", "p10", "p50", "p90", "p99", "p99.9", "p99.99", or "p100".

This **statistic** indicates a percentile. It resolves to a value by which compliance with the behavior is determined. Metrics are collected one or more times over the specified duration (**durationSeconds**) from all reporting devices associated with this security profile, and percentiles are calculated based on that data. After that, measurements are collected for a device and accumulated over the same duration. If the resulting value for the device falls above or below (**comparisonOperator**) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise it is in violation of the behavior.

A **percentile** indicates the percentage of all the measurements considered which fall below the associated value. For example, if the value associated with "p90" (the 90th percentile) is 123, then 90% of all measurements were below 123.

**durationSeconds**

Use this to specify the period of time over which the behavior is evaluated, for those criteria which have a time dimension (for example, **NUM\_MESSAGES\_SENT**). For a **statisticalThreshold** metric comparison, this is the time period during which measurements are collected for all devices to determine the **statisticalThreshold** values, and then for each device to determine how its behavior ranks in comparison.

#### consecutiveDatapointsToAlarm

If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1. (Note that this differs from the IoT console where a value of 3 is presented by default, but can be overridden.)

#### consecutiveDatapointsToClear

If an alert has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1. (Note that this differs from the IoT console where a value of 3 is presented by default, but can be overridden.)

## Metrics

### aws:message-byte-size

The number of bytes in a message.

[more info \(1\)](#)

Use this metric to specify the maximum or minimum size (in bytes) of each message transmitted from a device to AWS IoT.

Source: cloud-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: bytes

Example:

```
{  
  "name": "Max Message Size",  
  "metric": "aws:message-byte-size",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "value": {  
      "count": 1024  
    },  
    "consecutiveDatapointsToAlarm": 3,  
    "consecutiveDatapointsToClear": 3  
  }  
}
```

Example using a statisticalThreshold:

```
{  
  "name": "Large Message Size",  
  "metric": "aws:message-byte-size",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "statisticalThreshold": {  
      "statistic": "p90"  
    },  
    "durationSeconds": 300,  
    "consecutiveDatapointsToAlarm": 3,  
    "consecutiveDatapointsToClear": 3  
  }  
}
```

```
}
```

An alarm occurs for a device if, during three consecutive 5 minute periods, it transmits messages whose cumulative size is more than that measured for 90 percent of all other devices reporting for this security profile behavior.

**aws:num-messages-received / aws:num-messages-sent**

The number of messages received or sent by a device during a given time period.

[more info \(2\)](#)

Use this metric to specify the maximum or minimum number of messages that may be sent or received between AWS IoT and each device in a given period of time.

Source: cloud-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: messages

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{
  "name": "Out bound message count",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than",
    "value": {
      "count": 50
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 2,
    "consecutiveDatapointsToClear": 2
  }
}
```

Example using a statisticalThreshold:

```
{
  "name": "Out bound message rate",
  "metric": "aws:num-messages-sent",
  "criteria": {
    "comparisonOperator": "less-than",
    "statisticalThreshold": {
      "statistic": "p99"
    },
    "durationSeconds": 300,
    "consecutiveDatapointsToAlarm": 2,
    "consecutiveDatapointsToClear": 2
  }
}
```

**aws:all-bytes-out**

The number of outbound bytes from a device during a given time period.

[more info \(3\)](#)

Use this metric to specify the maximum or minimum amount of out-bound traffic that a device should send, measured in bytes in a given period of time.

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: bytes

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{  
  "name": "TCP outbound traffic",  
  "metric": "aws:all-bytes-out",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "value": {  
      "count": 4096  
    },  
    "durationSeconds": 300,  
    "consecutiveDatapointsToAlarm": 5,  
    "consecutiveDatapointsToClear": 4  
  }  
}
```

Example using a statisticalThreshold:

```
{  
  "name": "TCP outbound traffic",  
  "metric": "aws:all-bytes-out",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "statisticalThreshold": {  
      "statistic": "p50"  
    },  
    "durationSeconds": 900,  
    "consecutiveDatapointsToAlarm": 5,  
    "consecutiveDatapointsToClear": 4  
  }  
}
```

**aws:all-bytes-in**

The number of inbound bytes to a device during a given time period.

[more info \(4\)](#)

Use this metric to specify the maximum or minimum amount of in-bound traffic that a device should receive, measured in bytes in a given period of time.

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: bytes

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{  
    "name": "TCP inbound traffic",  
    "metric": "aws:all-bytes-in",  
    "criteria": {  
        "comparisonOperator": "less-than",  
        "value": {  
            "count": 4096  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 3  
    }  
}
```

Example using a statisticalThreshold:

```
{  
    "name": "TCP inbound traffic",  
    "metric": "aws:all-bytes-in",  
    "criteria": {  
        "comparisonOperator": "less-than",  
        "statisticalThreshold": {  
            "statistic": "p90"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 3  
    }  
}
```

aws:all-packets-out

The number of outbound packets from a device during a given time period.

[more info \(5\)](#)

Use this metric to specify the maximum or minimum amount of total outbound traffic that a device should send in a given period of time.

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: packets

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{  
    "name": "TCP outbound traffic",  
    "metric": "aws:all-packets-out",  
    "criteria": {  
        "comparisonOperator": "less-than",  
        "value": {  
            "count": 100  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 3  
    }  
}
```

Example using a statisticalThreshold:

```
{  
    "name": "TCP outbound traffic",  
    "metric": "aws:all-packets-out",  
    "criteria": {  
        "comparisonOperator": "less-than",  
        "statisticalThreshold": {  
            "statistic": "p90"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 3  
    }  
}
```

## aws:all-packets-in

The number of inbound packets to a device during a given time period.

[more info \(6\)](#)

Use this metric to specify the maximum or minimum amount of total inbound traffic that a device should receive in a given period of time.

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: packets

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{  
    "name": "TCP inbound traffic",  
    "metric": "aws:all-packets-in",
```

```

    "criteria": {
        "comparisonOperator": "less-than",
        "value": {
            "count": 100
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 2,
        "consecutiveDatapointsToClear": 1
    }
}

```

Example using a statisticalThreshold:

```

{
    "name": "TCP inbound traffic",
    "metric": "aws:all-packets-in",
    "criteria": {
        "comparisonOperator": "less-than",
        "statisticalThreshold": {
            "statistic": "p90"
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 2,
        "consecutiveDatapointsToClear": 1
    }
}

```

#### aws:num-authorization-failures

The number of authorization failures during a given time period.

[more info \(7\)](#)

Use this metric to specify the maximum number of authorization failures allowed for each device in a given period of time. An authorization failure occurs when a request from a device to AWS IoT is denied, for example, if a device attempts to publish to a topic for which it does not have sufficient permissions.

Source: cloud-side

Unit: failures

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: failures

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```

{
    "name": "Authorization Failures",
    "metric": "aws:num-authorization-failures",
    "criteria": {
        "comparisonOperator": "less-than",
        "value": {
            "count": 5
        }
    }
}

```

```
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 2,
        "consecutiveDatapointsToClear": 1
    }
}
```

Example using a statisticalThreshold:

```
{
    "name": "Authorization Failures",
    "metric": "aws:num-authorization-failures",
    "criteria": {
        "comparisonOperator": "less-than",
        "statisticalThreshold": {
            "statistic": "p50"
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 2,
        "consecutiveDatapointsToClear": 1
    }
}
```

#### aws:source-ip-address

The IP address from which a device has connected to AWS IoT.

[more info \(8\)](#)

Use this metric to specify a set of whitelisted or blacklisted CIDRs from which each device must or must not connect to AWS IoT.

Source: cloud-side

Operators: in-cidr-set | not-in-cidr-set

Values: a list of CIDRs

Units: n/a

Example:

```
{
    "name": "Blacklisted 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" ]
        }
    }
}
```

#### aws:destination-ip-addresses

A set of IP destinations.

[more info \(9\)](#)

Use this metric to specify a set of whitelisted or blacklisted CIDRs that each device must or must not communicate with.

Source: device-side

Operators: in-cidr-set | not-in-cidr-set

Values: a list of CIDRs

Units: n/a

Example:

```
{  
    "name": "Blacklisted destination IPs",  
    "metric": "aws:destination-ip-addresses",  
    "criteria": {  
        "comparisonOperator": "not-in-cidr-set",  
        "value": {  
            "cidrs": [ "12.8.0.0/16", "15.102.16.0/24" ]  
        }  
    }  
}
```

[aws:listening-tcp-ports / aws:listening-udp-ports](#)

The TCP or UDP ports that the device is listening on.

[more info \(10\)](#)

Use this metric to specify a set of whitelisted or blacklisted TCP/UDP ports that each device must or must not listen on.

Source: device-side

Operators: in-port-set | not-in-port-set

Values: a list of ports

Units: n/a

Example:

```
{  
    "name": "Listening TCP Ports",  
    "metric": "aws:listening-tcp-ports",  
    "criteria": {  
        "comparisonOperator": "in-port-set",  
        "value": {  
            "ports": [ 443, 80 ]  
        }  
    }  
}
```

**aws:num-listening-tcp-ports / aws:num-listening-udp-ports**

The number of TCP or UDP ports the device is listening on.

[more info \(11\)](#)

Use this metric to specify the maximum or minimum number of TCP or UDP ports that each device should listen on.

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: ports

Example:

```
{  
  "name": "Max TCP Ports",  
  "metric": "aws:num-listening-tcp-ports",  
  "criteria": {  
    "comparisonOperator": "less-than-equals",  
    "value": {  
      "count": 4  
    },  
    "consecutiveDatapointsToAlarm": 2,  
    "consecutiveDatapointsToClear": 1  
  }  
}
```

Example using a statisticalThreshold:

```
{  
  "name": "Max TCP Ports",  
  "metric": "aws:num-listening-tcp-ports",  
  "criteria": {  
    "comparisonOperator": "less-than-equals",  
    "statisticalThreshold": {  
      "statistic": "p90"  
    },  
    "durationSeconds": 300,  
    "consecutiveDatapointsToAlarm": 2,  
    "consecutiveDatapointsToClear": 1  
  }  
}
```

**aws:num-established-tcp-connections**

The number of TCP connections for a device.

[more info \(12\)](#)

Use this metric to specify the maximum or minimum number of active TCP connections that each device should have. (All TCP states)

Source: device-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: connections

Example:

```
{  
  "name": "TCP Connection Count",  
  "metric": "aws:num-established-tcp-connections",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "value": {  
      "count": 3  
    },  
    "consecutiveDatapointsToAlarm": 3,  
    "consecutiveDatapointsToClear": 3  
  }  
}
```

Example using a statisticalThreshold:

```
{  
  "name": "TCP Connection Count",  
  "metric": "aws:num-established-tcp-connections",  
  "criteria": {  
    "comparisonOperator": "less-than",  
    "statisticalThreshold": {  
      "statistic": "p90"  
    },  
    "durationSeconds": 900,  
    "consecutiveDatapointsToAlarm": 3,  
    "consecutiveDatapointsToClear": 3  
  }  
}
```

### aws:num-connection-attempts

The number of times a device has attempted to make a connection in a given time period.

[more info \(13\)](#)

Use this metric to specify the maximum or minimum number of connection attempts for each device. Both successful and unsuccessful attempts are counted.

Source: cloud-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: connection attempts

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```
{  
  "name": "Connection Attempts",  
  "metric": "aws:num-connection-attempts",  
  "criteria": {
```

```

    "comparisonOperator": "greater-than",
    "value": {
        "count": 5
    },
    "durationSeconds": 600,
    "consecutiveDatapointsToAlarm": 1,
    "consecutiveDatapointsToClear": 2
}
}

```

Example using a statisticalThreshold:

```

{
    "name": "Connection Attempts",
    "metric": "aws:num-connection-attempts",
    "criteria": {
        "comparisonOperator": "greater-than",
        "statisticalThreshold": {
            "statistic": "p10"
        },
        "durationSeconds": 300,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 2
    }
}

```

#### aws:num-disconnects

The number of times a device has disconnected from AWS IoT during a given time period.  
[more info \(14\)](#)

Use this metric to specify the maximum or minimum number of times a device has disconnected from AWS IoT during a given time period.

Source: cloud-side

Operators: less-than | less-than-equals | greater-than | greater-than-equals

Value: a non-negative integer

Units: disconnects

Duration: a non-negative integer, valid values are 300, 600, 900, 1800 or 3600 seconds

Example:

```

{
    "name": "Disconnects",
    "metric": "aws:num-disconnects",
    "criteria": {
        "comparisonOperator": "greater-than",
        "value": {
            "count": 5
        },
        "durationSeconds": 600,
        "consecutiveDatapointsToAlarm": 1,
        "consecutiveDatapointsToClear": 2
    }
}

```

Example using a statisticalThreshold:

```
{  
    "name": "Disconnections",  
    "metric": "aws:num-disconnects",  
    "criteria": {  
        "comparisonOperator": "greater-than",  
        "statisticalThreshold": {  
            "statistic": "p10"  
        },  
        "durationSeconds": 300,  
        "consecutiveDatapointsToAlarm": 1,  
        "consecutiveDatapointsToClear": 2  
    }  
}
```

## How To Use AWS IoT Device Defender Detect

1. You can use AWS IoT Device Defender Detect with just cloud-side metrics, but if you plan to use device reported metrics, you must first deploy an agent on your AWS IoT connected devices or device gateways. See [Sending Metrics from Devices \(p. 550\)](#).
2. You should consider viewing the actual metrics that your devices generate before you define behaviors and before you generate alarms. You can have AWS IoT collect metrics from your devices to see what might be usual, or unusual behavior for a group of devices, or for all devices in your account, without generating alarms. Use [CreateSecurityProfile \(p. 556\)](#) but specify only those additionalMetricsToRetain in which you are interested. Do not specify any behaviors at this point.
3. Create a set of behaviors for your security profile. Behaviors contain metrics that specify normal behavior for a group of devices or for all devices in your account. [Metrics \(p. 537\)](#) has more information, including examples. After you have created a set of behaviors, you can validate them with [ValidateSecurityProfileBehaviors \(p. 591\)](#).
4. Use [CreateSecurityProfile \(p. 556\)](#) to create a security profile that includes your behaviors. You can have alerts sent to a target (an SNS topic) when a device violates a behavior by using the alertTargets parameter. (If you do send alerts using SNS be aware that these will count against your account's SNS limit. It is possible for a large burst of violations to exhaust your capacity.) You can also check for violations using CloudWatch metrics (see [AWS IoT Metrics \(p. 620\)](#)).
5. Use [AttachSecurityProfile \(p. 555\)](#) to attach the security profile to a group of devices (a thing group) or to all the devices in your account. AWS IoT Device Defender Detect will begin checking for abnormal behavior and will send alerts if any behavior violations are detected.

To attach a security profile to a group of devices, you must specify the ARN of the thing group which contains them. A thing group ARN has the following format:

```
arn:aws:iot:<region>:<accountid>:thinggroup/<thing-group-name>
```

To attach a security profile to all of the devices in an account, you must specify an ARN with the following format:

```
arn:aws:iot:<region>:<accountid>:all/things
```

6. You can also keep track of violations with [ListActiveViolations \(p. 568\)](#) which allows you to see what violations have been detected for a given security profile or target device.

Use [ListViolationEvents \(p. 577\)](#) to see what violations have been detected during a specified time period. You can filter these results by a given security profile or device.

7. If your devices violate the behaviors you have defined too often, or not often enough, you will want to fine-tune the behavior definitions.
  8. To review the security profiles you have set up and the devices that are being monitored, use [ListSecurityProfiles \(p. 573\)](#), [ListSecurityProfilesForTarget \(p. 574\)](#), and [ListTargetsForSecurityProfile \(p. 576\)](#).
- You can get more details about a particular security profile with [DescribeSecurityProfile \(p. 562\)](#).
9. To make changes to a security profile, use [UpdateSecurityProfile \(p. 583\)](#). You may detach a security profile from an account or target thing group using [DetachSecurityProfile \(p. 567\)](#) or delete a security profile entirely with [DeleteSecurityProfile \(p. 561\)](#).

## Permissions

This section contains information on how to set up the AWS IAM roles and policies required to manage AWS IoT Device Defender Detect. For more information on AWS IAM, see the [AWS Identity and Access Management User Guide](#).

### Give AWS IoT Device Defender Detect permission to publish alerts to an SNS topic.

If you use the `alertTargets` parameter in [CreateSecurityProfile \(p. 556\)](#), you must specify an IAM Role with two policies, a permissions policy and a trust policy. The permissions policy grants permission to AWS IoT Device Defender to publish notifications to your SNS topic. The trust policy grants AWS IoT Device Defender permission to assume the required role.

[permissions policy](#)

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "sns:Publish"  
            ],  
            "Resource": [  
                "arn:aws:sns:region:account-id:your-topic-name"  
            ]  
        }  
    ]  
}
```

[trust policy](#)

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

```
}
```

```
pass role policy
```

You also need an IAM permissions policy attached to the IAM user that allows the user to pass roles. See [Granting a User Permissions to Pass a Role to an AWS Service](#).

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "",
            "Effect": "Allow",
            "Action": [
                "iam:GetRole",
                "iam:PassRole"
            ],
            "Resource": "arn:aws:iam::>account-id<:role/Role_To_Pass"
        }
    ]
}
```

## Service Limits

- The maximum number of security profiles per target (thing group or user account) is 5.
- The maximum number of behaviors per security profile is 100.
- The maximum number of value elements (counts, IP addresses, ports) per security profile is 1000.
- Device metric reporting may be throttled to one metric report per 5 minutes per device. So limit your device reporting to one metric report every 5 minutes to avoid throttling.
- AWS IoT Device Defender Detect violations are stored for 90 days after they have been generated.

## Sending Metrics from Devices

AWS IoT Device Defender Detect can collect, aggregate, and monitor metrics data generated by AWS IoT devices to identify devices that are exhibiting abnormal behavior. This section contains information on how to send metrics from a device to AWS IoT Device Defender.

You must securely deploy an agent on your AWS IoT connected devices or device gateways to collect device-side metrics. AWS IoT Device Defender provides sample agents to use as examples when creating your own. If you are unable to provide device metrics, you can still get limited functionality based on cloud-side metrics.

Please note the following:

- A sample device metric reporting agent is currently available in C at <https://github.com/aws-samples/aws-iot-device-defender-agent-c>. There is also a sample device metric reporting agent available in Python on GitHub at <https://github.com/aws-samples/aws-iot-device-defender-agent-sdk-python> (specifically [here](#)).
- To use the sample agents or create your own custom agent, you must install the AWS IoT Device SDK. To see the links for your development language of choice, go to [AWS IoT Core Resources](#).
- All agents must create a connection to AWS IoT and publish metrics to one of these reserved Device Defender MQTT topics:

```
$aws/things/<THING_NAME>/Defender/metrics/json
```

or

```
$aws/things/THING_NAME/Defender/metrics/cbor
```

Device Defender will reply with the receipt status of your metrics reports using one of these topics:

```
$aws/things/THING_NAME/Defender/metrics/json/accepted
```

```
$aws/things/THING_NAME/Defender/metrics/cbor/accepted
```

```
$aws/things/THING_NAME/Defender/metrics/json/rejected
```

```
$aws/things/THING_NAME/Defender/metrics/cbor/rejected
```

- In order to report metrics, a device must be registered as a thing in AWS IoT.
- A device should, generally, send a metric report once every 5 minutes. Devices are throttled to one metric report every 5 minutes (a device may not make more than one metric report every 5 minutes).
- Devices must report current metrics; historical metrics reporting is not supported.
- You can use [Jobs \(p. 354\)](#) to set the metrics reporting interval of your device metric reporting agent instances. An example of how to do this is included with the AWS Device Defender Agent C samples. See the [README.md](#) for more information.

## Device Metrics Document Specification

### Overall Structure:

Long Name	Short Name	Required	Type	Constraints	Notes
header	hed	Y	Object		Complete block required for well-formed report.
metrics	met	Y	Object		Complete block required for well-formed report.

### Header Block:

Long Name	Short Name	Required	Type	Constraints	Notes
report_id	rid	Y	Integer		Monotonically increasing value, epoch timestamp recommended
version	v	Y	String	Major.Minor	minor increments with addition

Long Name	Short Name	Required	Type	Constraints	Notes
					of field, major increments if metrics removed

#### Metrics Block:

##### TCP Connections:

Long Name	Short Name	Parent Element	Required	Type	Constraints	Notes
tcp_connections	tc	metrics	N	Object		
established_connections	conne	tcp_connections	N	List<Connection>		ESTABLISHED TCP State
connections	cs	established_connections		List<Object>		
remote_addr	rad	connections	Y	Number	ip:port	ip can be ipv6 or ipv4
local_port	lp	connections	N	Number	>= 0	
local_interface	li	connections	N	String		interface name
total	t	established_connections		Number	>= 0	Number of established connections

##### Listening TCP Ports:

Long Name	Short Name	Parent Element	Required	Type	Constraints	Notes
listening_tcp_ports	pts	metrics	N	Object		
ports	pts	listening_tcp_ports	N	List<Port>	> 0	
port	pt	ports	N	Number	> 0	ports should be numbers > 0
interface	if	ports	N	String		interface name
total	t	listening_tcp_ports		Number	>= 0	

##### Listening UDP Ports:

Long Name	Short Name	Parent Element	Required	Type	Constraints	Notes
listening_udp_ports	ports	metrics	N	Object		

Long Name	Short Name	Parent Element	Required	Type	Constraints	Notes
ports	pts	listening_udp_ports		List<Port>	> 0	
port	pt	ports	N	Number	> 0	ports should be numbers > 0
interface	if	ports	N	String		interface name
total	t	listening_udp_ports		Number	>= 0	

### Network Stats:

Long Name	Short Name	Parent Element	Required	Type	Constraints	Notes
network_stats	ns	metrics	N	Object		
bytes_in	bi	network_stats	N	Number	Delta Metric, >= 0	
bytes_out	bo	network_stats	N	Number	Delta Metric, >= 0	
packets_in	pi	network_stats	N	Number	Delta Metric, >= 0	
packets_out	po	network_stats	N	Number	Delta Metric, >= 0	

Example JSON structure using long names:

```
{
  "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
    }
}
}
```

Example JSON structure using short names:

```
{
    "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
    }
}
}
```

## Detect Commands

### AttachSecurityProfile

Associates a Device Defender security profile with a thing group or with this account. Each thing group or account can have up to five security profiles associated with it.

#### Synopsis:

```
aws iot attach-security-profile \
--security-profile-name <value> \
--security-profile-target-arn <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format:

```
{
    "securityProfileName": "string",
    "securityProfileTargetArn": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile that is attached.
securityProfileTargetArn	string	The ARN of the target (thing group) to which the security profile is attached.

**Output:**

None

**Errors:**

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**LimitExceededException**

A limit has been exceeded.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## CreateSecurityProfile

Creates a Device Defender security profile.

**Synopsis:**

```
aws iot create-security-profile \
--security-profile-name <value> \
[--security-profile-description <value>] \
[--behaviors <value>] \
[--alert-targets <value>] \
[--additional-metrics-to-retain <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "securityProfileName": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ],
  "alertTargets": {
    "string": {
      "alertTargetArn": "string",
      "roleArn": "string"
    }
  },
  "additionalMetricsToRetain": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you are giving to the security profile.
securityProfileDescription	string  length- max:1000  pattern: [\p{Graph} ]*	A description of the security profile.
behaviors	list  member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.

Name	Type	Description
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cidrs	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.

Name	Type	Description
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Specifies the destinations to which alerts are sent. (Alerts are always sent to the console.) Alerts are generated when a device (thing) violates a behavior.
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage the security profile.
Key	string	The tag's key.
Value	string	The tag's value.

Output:

```
{
  "securityProfileName": "string",
```

```

    "securityProfileArn": "string"
}
```

**cli output fields:**

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name you gave to the security profile.
securityProfileArn	string	The ARN of the security profile.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ResourceAlreadyExistsException`

The resource already exists.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DeleteSecurityProfile

Deletes a Device Defender security profile.

**Synopsis:**

```
aws iot delete-security-profile \
--security-profile-name <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "securityProfileName": "string",
  "expectedVersion": "long"
}
```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string length- max:128 min:1	The name of the security profile to be deleted.

Name	Type	Description
	pattern: [a-zA-Z0-9:_]+	
expectedVersion	long	The expected version of the security profile. A new version is generated whenever the security profile is updated. If you specify a value that is different than the actual version, a <code>VersionConflictException</code> is thrown.

**Output:**

None

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

## DescribeSecurityProfile

Gets information about a Device Defender security profile.

**Synopsis:**

```
aws iot describe-security-profile \
--security-profile-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "securityProfileName": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string	The name of the security profile whose information you want to get. length- max:128 min:1

Name	Type	Description
	pattern: [a-zA-Z0-9:_-]+	

Output:

```
{
    "securityProfileName": "string",
    "securityProfileArn": "string",
    "securityProfileDescription": "string",
    "behaviors": [
        {
            "name": "string",
            "metric": "string",
            "criteria": {
                "comparisonOperator": "string",
                "value": {
                    "count": "long",
                    "cidrs": [
                        "string"
                    ],
                    "ports": [
                        "integer"
                    ]
                },
                "durationSeconds": "integer",
                "consecutiveDatapointsToAlarm": "integer",
                "consecutiveDatapointsToClear": "integer",
                "statisticalThreshold": {
                    "statistic": "string"
                }
            }
        }
    ],
    "alertTargets": {
        "string": {
            "alertTargetArn": "string",
            "roleArn": "string"
        }
    },
    "additionalMetricsToRetain": [
        "string"
    ],
    "version": "long",
    "creationDate": "timestamp",
    "lastModifiedDate": "timestamp"
}
```

**cli output fields:**

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name of the security profile.
securityProfileArn	string	The ARN of the security profile.
securityProfileDescription	string	A description of the security profile (associated with the

Name	Type	Description
	length- max:1000 pattern: [\p{Graph} ]*	security profile when it was created or updated).
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.

Name	Type	Description
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.
version	long	The version of the security profile. A new version is generated whenever the security profile is updated.
creationDate	timestamp	The time the security profile was created.
lastModifiedDate	timestamp	The time the security profile was last modified.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DetachSecurityProfile

Disassociates a Device Defender security profile from a thing group or from this account.

### Synopsis:

```
aws iot detach-security-profile \
  --security-profile-name <value> \
  --security-profile-target-arn <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "securityProfileName": "string",
  "securityProfileTargetArn": "string"
}
```

### cli-input-json fields:

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile that is detached.
securityProfileTargetArn	string	The ARN of the thing group from which the security profile is detached.

### Output:

None

### Errors:

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## ListActiveViolations

Lists the active violations for a given Device Defender security profile.

### Synopsis:

```
aws iot list-active-violations \
[--thing-name <value>] \
[--security-profile-name <value>] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "thingName": "string",
  "securityProfileName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields:

Name	Type	Description
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing whose active violations are listed.
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the Device Defender security profile for which violations are listed.
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

### Output:

```
{
  "activeViolations": [
    {
      "id": "string",
      "severity": "string",
      "lastUpdated": "string",
      "violations": [
        {
          "id": "string",
          "rule": "string",
          "lastUpdated": "string",
          "status": "string"
        }
      ]
    }
  ]
}
```

```

    "violationId": "string",
    "thingName": "string",
    "securityProfileName": "string",
    "behavior": {
        "name": "string",
        "metric": "string",
        "criteria": {
            "comparisonOperator": "string",
            "value": {
                "count": "long",
                "cidrs": [
                    "string"
                ],
                "ports": [
                    "integer"
                ]
            },
            "durationSeconds": "integer",
            "consecutiveDatapointsToAlarm": "integer",
            "consecutiveDatapointsToClear": "integer",
            "statisticalThreshold": {
                "statistic": "string"
            }
        }
    },
    "lastViolationValue": {
        "count": "long",
        "cidrs": [
            "string"
        ],
        "ports": [
            "integer"
        ]
    },
    "lastViolationTime": "timestamp",
    "violationStartTime": "timestamp"
},
],
"nextToken": "string"
}

```

#### cli output fields:

Name	Type	Description
activeViolations	list  member: ActiveViolation	The list of active violations.
violationId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-]+	The ID of the active violation.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name of the thing responsible for the active violation.
securityProfileName	string  length- max:128 min:1	The security profile whose behavior is in violation.

Name	Type	Description
	pattern: [a-zA-Z0-9:_]+	
behavior	Behavior	The behavior which is being violated.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long  range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list  member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list  member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.

Name	Type	Description
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
lastViolationValue	MetricValue	The value of the metric (the measurement) which caused the most recent violation.
count	long  range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cids	list  member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list  member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
lastViolationTime	timestamp	The time the most recent violation occurred.
violationStartTime	timestamp	The time the violation started.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## ListSecurityProfiles

Lists the Device Defender security profiles you have created. You can use filters to list only those security profiles associated with a thing group or only those associated with your account.

### Synopsis:

```
aws iot list-security-profiles \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields:

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

### Output:

```
{
  "securityProfileIdentifiers": [
    {
      "name": "string",
      "arn": "string"
    }
  ],
  "nextToken": "string"
}
```

**cli output fields:**

Name	Type	Description
securityProfileIdentifiers	list  member: SecurityProfileIdentifier  java class: java.util.List	A list of security profile identifiers (names and ARNs).
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the security profile.
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## ListSecurityProfilesForTarget

Lists the Device Defender security profiles attached to a target (thing group).

**Synopsis:**

```
aws iot list-security-profiles-for-target \
[--next-token <value>] \
[--max-results <value>] \
[--recursive | --no-recursive] \
--security-profile-target-arn <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format:**

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "recursive": "boolean",
  "securityProfileTargetArn": "string"
}
```

**cli-input-json fields:**

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.
recursive	boolean	If true, return child groups as well.
securityProfileTargetArn	string	The ARN of the target (thing group) whose attached security profiles you want to get.

**Output:**

```
{
  "securityProfileTargetMappings": [
    {
      "securityProfileIdentifier": {
        "name": "string",
        "arn": "string"
      },
      "target": {
        "arn": "string"
      }
    }
  ],
  "nextToken": "string"
}
```

**cli output fields:**

Name	Type	Description
securityProfileTargetMappings	list member: SecurityProfileTargetMapping java class: java.util.List	A list of security profiles and their associated targets.
securityProfileIdentifier	SecurityProfileIdentifier	Information that identifies the security profile.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the security profile.
arn	string	The ARN of the security profile.
target	SecurityProfileTarget	Information about the target (thing group) associated with the security profile.

Name	Type	Description
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## ListTargetsForSecurityProfile

Lists the targets (thing groups) associated with a given Device Defender security profile.

**Synopsis:**

```
aws iot list-targets-for-security-profile \
--security-profile-name <value> \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "securityProfileName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile.
nextToken	string	The token for the next set of results.

Name	Type	Description
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

Output:

```
{
  "securityProfileTargets": [
    {
      "arn": "string"
    }
  ],
  "nextToken": "string"
}
```

#### cli output fields:

Name	Type	Description
securityProfileTargets	list member: SecurityProfileTarget java class: java.util.List	The thing groups to which the security profile is attached.
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

#### Errors:

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## ListViolationEvents

Lists the Device Defender security profile violations discovered during the given time period. You can use filters to limit the results to those alerts issued for a particular security profile, behavior or thing (device).

#### Synopsis:

```
aws iot list-violation-events \
--start-time <value> \
--end-time <value> \
[--thing-name <value>] \
[--security-profile-name <value>] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "thingName": "string",
  "securityProfileName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

**cli-input-json fields:**

Name	Type	Description
startTime	timestamp	The start time for the alerts to be listed.
endTime	timestamp	The end time for the alerts to be listed.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter to limit results to those alerts caused by the specified thing.
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter to limit results to those alerts generated by the specified security profile.
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

**Output:**

```
{
  "violationEvents": [
    {
      "violationId": "string",
      "thingName": "string",
      "securityProfileName": "string",
      "behavior": {
```

```

    "name": "string",
    "metric": "string",
    "criteria": {
        "comparisonOperator": "string",
        "value": {
            "count": "long",
            "cidrs": [
                "string"
            ],
            "ports": [
                "integer"
            ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
            "statistic": "string"
        }
    },
    "metricValue": {
        "count": "long",
        "cidrs": [
            "string"
        ],
        "ports": [
            "integer"
        ]
    },
    "violationEventType": "string",
    "violationEventTime": "timestamp"
},
],
"nextToken": "string"
}

```

**cli output fields:**

Name	Type	Description
violationEvents	list  member: ViolationEvent	The security profile violation alerts issued for this account during the given time frame, potentially filtered by security profile, behavior violated, or thing (device) violating.
violationId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-]+	The ID of the violation event.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing responsible for the violation event.
securityProfileName	string  length- max:128 min:1	The name of the security profile whose behavior was violated.

Name	Type	Description
	pattern: [a-zA-Z0-9:_]+	
behavior	Behavior	The behavior which was violated.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long  range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list  member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list  member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.

Name	Type	Description
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
metricValue	MetricValue	The value of the metric (the measurement).
count	long  range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cids	list  member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list  member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
violationEventType	string  enum: in-alarm   alarm-cleared   alarm-invalidated	The type of violation event.
violationEventTime	timestamp	The time the violation event occurred.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

**Errors:**

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## UpdateSecurityProfile

Updates a Device Defender security profile.

### Synopsis:

```
aws iot update-security-profile \
--security-profile-name <value> \
[--security-profile-description <value>] \
[--behaviors <value>] \
[--alert-targets <value>] \
[--additional-metrics-to-retain <value>] \
[--delete-behaviors | --no-delete-behaviors] \
[--delete-alert-targets | --no-delete-alert-targets] \
[--delete-additional-metrics-to-retain | --no-delete-additional-metrics-to-retain] \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "securityProfileName": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ],
  "alertTargets": {
    "string": {
      "alertTargetArn": "string",
      "region": "string"
    }
  }
}
```

```

        "roleArn": "string"
    },
},
"additionalMetricsToRetain": [
    "string"
],
"deleteBehaviors": "boolean",
"deleteAlertTargets": "boolean",
"deleteAdditionalMetricsToRetain": "boolean",
"expectedVersion": "long"
}

```

**cli-input-json fields:**

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the security profile you want to update.
securityProfileDescription	string length- max:1000 pattern: [\p{Graph} ]*	A description of the security profile.
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use

Name	Type	Description
		this to specify that numeric value to be compared with the metric.
cidrs	list member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, <code>NUM_MESSAGES_SENT</code> ). For a <code>statisticalThreshold</code> metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.
deleteBehaviors	boolean	If true, delete all behaviors defined for this security profile. If any behaviors are defined in the current invocation an exception occurs.
deleteAlertTargets	boolean	If true, delete all alertTargets defined for this security profile. If any alertTargets are defined in the current invocation an exception occurs.

Name	Type	Description
deleteAdditionalMetricsToRetain	boolean	If true, delete all additionalMetricsToRetain defined for this security profile. If any additionalMetricsToRetain are defined in the current invocation an exception occurs.
expectedVersion	long	The expected version of the security profile. A new version is generated whenever the security profile is updated. If you specify a value that is different than the actual version, a VersionConflictException is thrown.

**Output:**

```
{
  "securityProfileName": "string",
  "securityProfileArn": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ],
  "alertTargets": {
    "string": {
      "alertTargetArn": "string",
      "roleArn": "string"
    }
  },
  "additionalMetricsToRetain": [
    "string"
  ],
  "version": "long",
  "creationDate": "timestamp",
  "lastModifiedDate": "timestamp"
}
```

}

**cli output fields:**

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the security profile that was updated.
securityProfileArn	string	The ARN of the security profile that was updated.
securityProfileDescription	string length- max:1000 pattern: [\p{Graph} ]*	The description of the security profile.
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use

Name	Type	Description
		this to specify that set to be compared with the metric.
ports	list member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the security profile's behaviors but it is also retained for any metric specified here.
version	long	The updated version of the security profile.
creationDate	timestamp	The time the security profile was created.
lastModifiedDate	timestamp	The time the security profile was last modified.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

#### `VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

#### `ThrottlingException`

The rate exceeds the limit.

#### `InternalFailureException`

An unexpected error has occurred.

## ValidateSecurityProfileBehaviors

Validates a Device Defender security profile behaviors specification.

### Synopsis:

```
aws iot validate-security-profile-behaviors \
--behaviors <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format:

```
{
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ]
}
```

**cli-input-json fields:**

Name	Type	Description
behaviors	list  member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.

Name	Type	Description
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long  range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cidrs	list  member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list  member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.

Name	Type	Description
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.

**Output:**

```
{
  "valid": "boolean",
  "validationErrors": [
    {
      "errorMessage": "string"
    }
  ]
}
```

**cli output fields:**

Name	Type	Description
valid	boolean	True if the behaviors were valid.
validationErrors	list  member: ValidationError	The list of any errors found in the behaviors.
errorMessage	string  length- max:2048	The description of an error found in the behaviors.

**Errors:**

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## Device Agent Integration with AWS IoT Greengrass

AWS IoT Device Defender can be used in conjunction with AWS IoT Greengrass. Device agent integration follows the standard AWS IoT Greengrass Lambda function deployment model, allowing you to add AWS IoT Device Defender security to your AWS IoT Greengrass Core devices. To integrate a device agent, follow the steps outlined in this section.

Prerequisites:

- Have your AWS IoT Greengrass environment set up.
- Have your AWS IoT Greengrass core configured and running.
- Ensure you can successfully deploy and run a Lambda function on your AWS IoT Greengrass core.

In general, the process described here follows the [Create and Package a Lambda Function](#) section of the AWS IoT Greengrass Developers Guide.

### Create A Lambda Package

1. Clone the AWS IoT Device Defender Python Samples Repository.

```
git clone https://github.com/aws-samples/aws-iot-device-defender-agent-sdk-python.git
```

2. Create and activate a virtual environment (optional but recommended).

```
pip install virtualenv
virtualenv metrics_lambda_environment
source metrics_lambda_environment/bin/activate
```

3. Install the AWS IoT Device Defender sample agent in the virtual environment Install from PyPi.

```
pip install AWSIoTDeviceDefenderAgentSDK
```

4. Install the downloaded source.

```
cd aws-iot-device-defender-agent-sdk-python
#This must be run from the same directory as setup.py
pip install .
```

5. Create an empty directory to assemble your Lambda function, we will refer to this as your “Lambda directory”.

```
mkdir metrics_lambda
cd metrics_lambda
```

6. Complete steps 1-4 from this section: [Create and Package a Lambda Function](#).
7. Unzip the AWS IoT Greengrass python sdk into your Lambda directory.

```
unzip ../aws_greengrass_core_sdk/sdk/python_sdk_1_1_0.zip
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrass_common .
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrasssdk .
```

```
cp -R ../aws_greengrass_core_sdk/examples/HelloWorld/greengrass_ipc_python_sdk .
```

8. Copy the AWSIoTDeviceDefenderAgentSDK module to the root level of your Lambda directory.

```
cp -R ../aws-iot-device-defender-agent-sdk-python/AWSIoTDeviceDefenderAgentSDK .
```

9. Copy the AWS IoT Greengrass agent to the root level of your Lambda directory.

```
cp ../aws-iot-device-defender-agent-sdk-python/samples/greengrass/greengrass_core_metrics_agent/greengrass_defender_agent.py .
```

10. Customize the AWS IoT Greengrass agent to include the name of your AWS IoT Greengrass Core device, and the desired metrics sample rate:

- Replace GREENGASS\_CORENAME with the name of your AWS IoT Greengrass Core.
- Set the SAMPLE\_RATE\_SECONDS to your desired metrics reporting interval. (Note: 5 minutes (300 seconds) is the shortest reporting interval supported by AWS IoT Device Defender.)

11. Copy the dependencies from your virtual environment (or your system) into the the root level of your Lambda directory.

```
cp -R ../metrics_lambda_environment/lib/python2.7/site-packages/psutil .
cp -R ../metrics_lambda_environment/lib/python2.7/site-packages/cbor .
```

12. Create your Lambda function zipfile. (Note: you should perform this command in the root level of your Lambda directory.)

```
rm *.zip
zip -r greengrass_defender_metrics_lambda.zip *
```

## Configure and deploy your AWS IoT Greengrass Lambda function

1. [Upload your lambda zip file](#).
2. Select the Python 2.7 runtime, and enter `greengrass_defender_agent.function_handler` in the Handler field.
3. [Configure your lambda as a long-lived lambda](#).
4. [Configure a subscription from your Lambda function to the AWS IoT Cloud](#). (Note: For AWS IoT Device Defender, a subscription from the AWS IoT cloud to your Lambda function is not required.)
5. Create a local resource to allow your Lambda function to collect metrics from your AWS IoT Greengrass Core host:
  - Follow the instructions [here](#) using the following parameters:
    - Resource Name: Core\_Proc
    - Type: Volume
    - Source Path: /proc
    - Destination Path: /host\_proc
    - Group owner file access permission: "Automatically add OS group permissions of the Linux group that owns the resource"
    - Associate the resource with your metrics Lambda function.
6. Deploy your Lambda function to your AWS IoT Greengrass Group.

## Review your AWS IoT Device Defender device metrics using the AWS IoT Console

1. Temporarily modify the publish topic in your AWS IoT Greengrass Lambda function to "metrics/test".

2. Deploy the Lambda function.
3. Add a subscription to the temporary topic ("metrics/test") in the "Test" section of the IoT console in order to see the metrics your AWS IoT Greengrass Core is emitting.

## Security Best Practices for Device Agents

### Least Privilege

The agent process should be granted only the minimum permissions necessary to perform its duties.

#### Basic Mechanisms

- Agent should be run as non-root user.
- Agent should run as a dedicated user, in its own group.
- User/groups should be granted Read-Only permissions on the resources required to gather and transmit metrics.
- Example: read-only on /proc /sys for the sample agent.
- For an example of how to setup a process to run with reduced permissions, see the Setup Instructions which are included with the Python Sample Agent.

There are a number of well-known Linux mechanisms that can help you further restrict/isolate your agent process:

#### Advanced Mechanisms

- [CGroups](#)
- [SELinux](#)
- [Chroot](#)
- [Linux Namespaces](#)

### Operational Resiliency

An agent process must be resilient to unexpected operational errors and exceptions and must not crash or exit permanently. The code needs to gracefully handle exceptions and, as a precaution, it must be configured to automatically restart in case of unexpected termination (for example, due to system restarts or uncaught exceptions).

### Least Dependencies

An agent must use the least possible number of dependencies (i.e. third-party libraries) in its implementation. If use of a library is justified due to the complexity of a task (for example, transport layer security) only use well-maintained dependencies and establish a mechanism to keep dependencies up to date. If the added dependencies contain functionality not used by the agent and active by default (for example, opening ports, domain sockets), disable them in your code or by means of the library's configuration files.

### Process Isolation

An agent process must only contain functionality required for performing device metric gathering and transmission. It must not piggyback on other system processes as a container or implement functionality for other out of scope use-cases. Additionally, the agent process must refrain from creating inbound communication channels such as domain socket and network service ports which would allow local or remote processes to interfere with its operation and impact its integrity and isolation.

### Stealthiness

An agent process must not be named with keywords such as security, monitoring, or audit indicating its purpose and security value. Generic code names or random and unique-per-device process names must be preferred. The same principle must be followed in naming the directory in which the agent's binaries reside and any names and values of process arguments.

### Least Information Shared

Any agent artifacts deployed to devices must not contain sensitive information such as privileged credentials, debugging and dead code, or inline comments or documentation files which reveal details about server-side processing of agent-gathered metrics or other details about backend systems.

### Transport Layer Security

To establish TLS secure channels for data transmission, an agent process must enforce all client-side validations, such as certificate chain and domain name validation, at the application level, if not enabled by default. Furthermore, an agent must use a root certificate store which contains trusted authorities and does not contain certificates belonging to compromised certificate issuers.

### Secure Deployment

Any agent deployment mechanism, such as code push or sync and repositories containing its binaries, source code and any configuration files (including trusted root certificates), must be access controlled to prevent unauthorized code injection or tampering. If the deployment mechanism relies on network communication, then cryptographic methods must be utilized to protect the integrity of deployment artifacts in transit.

### Further Reading

- [Security and Identity for AWS IoT](#)
- [Understanding the AWS IoT Security Model](#)
- [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](#)

## AWS IoT Device Defender Troubleshooting Guide

### General

Q: Are there any prerequisites to using AWS IoT Device Defender?

A: You can use the AWS IoT Device Defender Detect feature for devices that are registered as things with AWS IoT. To use the cloud-side metrics, devices must connect using the thing name as their client Id.

### Audit

Q: I enabled a check and my audit has been showing "In-Progress" for a long time. Is something wrong? When will I get results?

A: When a check is enabled, data collection begins immediately. However, if you have a large amount of data in your account to be collected (certificates, things, policies, etc.) then the results of the check may not be available for some time after you have enabled it.

## Detect

Q: How do I know what thresholds should be set in an AWS IoT Device Defender security profile behavior?

A: Start by creating a security profile behavior with low thresholds and attach it to a thing group consisting of a representative set of devices. Device defender will alert you with the metric datapoints emitted by the device for the behaviors that are violated. You can then fine-tune the device behavior thresholds to match your use case.

Q: I created a behavior, but it is not triggering a violation when I expect it to. How should I fix it?

A: Keep in mind that when you define a behavior you are indicating how you expect your device to behave normally. For example, if you have a security camera that only connects to one central server on TCP port 8888, you don't expect it to make any other connections. To be alerted if the camera makes a connection on another port, you could define a behavior such as:

```
{  
  "name": "Listening TCP Ports",  
  "metric": "aws:listening-tcp-ports",  
  "criteria": {  
    "comparisonOperator": "in-port-set",  
    "value": {  
      "ports": [ 8888 ]  
    }  
  }  
}
```

Once the behavior has been created, if the camera makes a TCP connection on TCP Port 443, the device behavior would be violated and would trigger an alert.

Q: One or more of my behaviors are in violation. How do I clear the violation?

A: Alarms will clear once the device returns to expected behavior, as defined by the behavior profiles you have defined. Behavior profiles are evaluated upon receipt of metrics data for your device.

Q: I deleted a behavior that was in violation, how do I stop the alerts?

A: Deleting a behavior will stop all future violations and alerts for that behavior. Prior alerts will need to be drained from your notification mechanism as normal. However, when a behavior is deleted, the record of violations of that behavior will be retained for the same time period as all other violations in your account.

## Device Metrics

Q: I'm submitting metrics reports that I know violate my behaviors, but no violations are being triggered. What's wrong?

A: Check that your metrics reports are being accepted by subscribing to the following MQTT topics:

```
$aws/things/THING_NAME/defender/metrics/FORMAT/rejected  
$aws/things/THING_NAME/defender/metrics/FORMAT/accepted
```

where "THING\_NAME" is the name of the thing reporting the metric and FORMAT is either "json" or "cbor" depending on the format of the metrics report the thing submits.

After you have subscribed, you should receive messages on these topics for each metric report submitted. A "rejected" message indicates that there was a problem parsing the metric report. An error message is included in the message payload to help you correct any errors in your metric report. An "accepted" message indicates the metric report was parsed properly.

Q: What happens if I send an empty metric in my metric report?

A: An empty list of ports or IP addresses is always considered in conformity with the corresponding behavior. If the corresponding behavior was in violation, then the violation will be cleared.

Q: My device metric reports are getting rejected with an error message of "XXXX Not a Thing". How can I fix it?

A: Make sure you are submitting metrics for a thing which is registered in your AWS IoT account. AWS IoT Device Defender requires things to be registered in your account to evaluate reported metrics against defined behaviors.

If you are submitting metrics from a registered thing, verify that you are sending a well-formed metrics report using one of the supported formats. ([Device Metrics Document Specification \(p. 551\)](#))

Q: What value should I supply in the report id field in my device metrics report?

A: This should be a unique value for each metric report, expressed as a positive integer. A common practice is to use a [Unix epoch timestamp](#).

Q: Should I create a dedicated MQTT connection for AWS IoT Device Defender metrics?

A: A separate MQTT connection is not required.

Q: What client id should I use when connecting to publish device metrics?

You should use a registered thing name as the clientId when connecting to AWS IoT.

Q: Can I publish metrics for a device with a different clientId?

It is possible to publish metrics on behalf of another thing, as long as that thing is registered in your AWS account. This can be accomplished by publishing the metrics to the AWS IoT Device Defender reserved topic for that device. For example, "Thing-1" would like to publish metrics for itself and also on behalf of "Thing-2". "Thing-1" collects its own metrics and publishes them on the MQTT topic:

```
$aws/things/Thing-1/defender/metrics/json
```

"Thing-1" will then obtain metrics from "Thing-2" and publish these metrics on the MQTT topic:

```
$aws/things/Thing-2/defender/metrics/json
```

As long as "Thing-1" and "Thing-2" are registered in your account, metric submission and evaluation will work properly.

Q: How many security profiles and behaviors can I have in my account?

A: Refer to the [Detect Service Limits \(p. 550\)](#) section of the AWS IoT Device Defender Developer Guide.

Q: What does a prototypical target role for an alert target look like?

A: A role that allows AWS IoT Device Defender to publish alerts on an alert target (SNS topic) requires 2 things:

1. a trust relationship specifying `iot.amazonaws.com` as the trusted entity and
2. an attached policy which grants AWS IoT permission to publish on a specified SNS topic, for example:

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

```
        "Action": "sns:Publish",
        "Resource": "<sns-topic-arn>"  
    }  
}  
]
```

# Event Messages

## Note

This section contains information about messages published by AWS IoT when things or jobs are updated or changed. For information about the AWS IoT Events service that allows you to create detectors to monitor your devices for failures or changes in operation, and to trigger specific actions when they occur, see the [AWS IoT Events](#) service overview.

AWS IoT publishes event messages when certain events occur. For example, events are generated by the registry when things are added, updated, or deleted. Each event causes a single event message to be sent. Event messages are published over MQTT with a JSON payload. The content of the payload depends on the type of event.

## Note

Event messages are guaranteed to be published once. It is possible for them to be published more than once. The ordering of event messages is not guaranteed.

To receive event messages, your device must use an appropriate policy that allows it to connect to the AWS IoT device gateway and subscribe to MQTT event topics. You must also subscribe to the appropriate topic filters.

The following is an example of the policy required for receiving lifecycle events:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iot:Subscribe",
                "iot:Receive"
            ],
            "Resource": [
                "arn:aws:iot:region:account:/$aws/events/*"
            ]
        }
    ]
}
```

You control which event types are published by calling the [UpdateEventConfigurations](#) API or by using the **update-event-configurations** CLI command. For example:

```
aws iot update-event-configurations --event-configurations "{\"THING\":{\"Enabled\": true}}"
```

## Note

All quotation marks ("") are escaped with a backslash (\).

You can get the current event configuration by calling the [DescribeEventConfigurations](#) API or by using the **describe-event-configurations** CLI command. For example:

```
aws iot describe-event-configurations
```

The output of the **describe-event-configurations** command looks like the following:

```
{
    "lastModifiedDate": 1552671347.841,
    "eventConfigurations": {
        "THING_TYPE": {
            "Enabled": false
        },
    }
}
```

```

    "JOB_EXECUTION": {
        "Enabled": false
    },
    "THING_GROUP_HIERARCHY": {
        "Enabled": false
    },
    "CERTIFICATE": {
        "Enabled": false
    },
    "THING_TYPE_ASSOCIATION": {
        "Enabled": false
    },
    "THING_GROUP_MEMBERSHIP": {
        "Enabled": false
    },
    "CA_CERTIFICATE": {
        "Enabled": false
    },
    "THING": {
        "Enabled": true
    },
    "JOB": {
        "Enabled": false
    },
    "POLICY": {
        "Enabled": false
    },
    "THING_GROUP": {
        "Enabled": false
    }
},
"creationDate": 1552671347.84
}

```

## Registry Events

The registry publishes event messages when things, thing types, and thing groups are created, updated, or deleted. The registry currently supports the following event types:

### Thing Created/Updated/Deleted

The registry publishes the following event messages when things are created, updated, or deleted:

- \$aws/events/thing/<thingName>/created
- \$aws/events/thing/<thingName>/updated
- \$aws/events/thing/<thingName>/deleted

The messages contain the following example payload:

```
{
    "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"
    }
```

The payloads contain the following attributes:

**eventType**

Set to "THING\_EVENT".

**eventId**

A unique event ID (string).

**timestamp**

The UNIX timestamp of when the event occurred.

**operation**

The operation that triggered the event. Valid values are:

- CREATED
- UPDATED
- DELETED

**accountId**

Your AWS account ID.

**thingId**

The ID of the thing being created, updated, or deleted.

**thingName**

The name of the thing being created, updated, or deleted.

**versionNumber**

The version of the thing being created, updated, or deleted. This value is set to 1 when a thing is created. It is incremented by 1 each time the thing is updated.

**thingTypeName**

The thing type associated with the thing, if one exists. Otherwise, null.

**attributes**

A collection of name-value pairs associated with the thing.

#### Thing Type Created/Deprecated/Undeleted/Deleted

The registry publishes the following event messages when thing types are created, deprecated, undeleted, or deleted:

- \$aws/events/thingType/<thingTypeName>/created
- \$aws/events/thingType/<thingTypeName>/updated
- \$aws/events/thingType/<thingTypeName>/deleted

The message contains the following example payload:

```
{
    "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"
}
```

The payloads contain the following attributes:

eventType

Set to "THING\_TYPE\_EVENT".

eventId

A unique event ID (string).

timestamp

The UNIX timestamp of when the event occurred.

operation

The operation that triggered the event. Valid values are:

- CREATED
- UPDATED
- DELETED

accountId

Your AWS account ID.

thingTypeId

The ID of the thing type being created, deprecated, or deleted.

thingTypeName

The name of the thing type being created, deprecated, or deleted.

isDeprecated

true if the thing type is deprecated. Otherwise, false.

deprecationDate

The UNIX timestamp for when the thing type was deprecated.

searchableAttributes

A collection of name-value pairs associated with the thing type that can be used for searching.

description

A description of the thing type.

Thing Type Associated or Disassociated with a Thing

The registry publishes the following event messages when a thing type is associated or disassociated with a thing.

- \$aws/events/thingTypeAssociation/thing/<thingName>/<typeName>

The messages contain the following example payload:

```
{
    "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,
}

```

The payloads contain the following attributes:

**eventId**

A unique event ID (string).

**eventType**

Set to "THING\_TYPE\_ASSOCIATION\_EVENT".

**operation**

The operation that triggered the event. Valid values are:

- CREATED
- DELETED

**thingId**

The ID of the thing whose type association was changed.

**thingName**

The name of the thing whose type association was changed.

**thingTypeName**

The thing type associated with, or no longer associated with, the thing.

**timestamp**

The UNIX timestamp of when the event occurred.

#### Thing Group Created/Updated/Deleted

The registry publishes the following event messages when a thing group is created, updated, or deleted.

- \$aws/events/thingGroup/<groupName>/created
- \$aws/events/thingGroup/<groupName>/updated
- \$aws/events/thingGroup/<groupName>/deleted

The messages contain the following example payload:

```

{
    "eventType" : "THING_GROUP_EVENT",
    "eventId" : "87f8e095-531c-47b3-aab5-5171364d138d",
    "timestamp" : 1234567890123,
    "operation" : "CREATED|UPDATED|DELETED",
    "accountId" : "123456789012",
    "thingGroupId" : "8f82a106-6b1d-4331-8984-a84db5f6f8cb",
    "thingGroupName" : "MyRootThingGroup",
    "versionNumber" : 1,
    "parentGroupName" : null,
    "parentGroupId" : null,
    "description" : "My root thing group",
    "rootToParentThingGroups" : null,
    "attributes" : {
        "attribute1" : "value1",
        "attribute3" : "value3",
        "attribute2" : "value2"
    }
}

```

```
    }
```

The payloads contain the following attributes:

**eventType**

Set to "THING\_GROUP\_EVENT".

**eventId**

A unique event ID (string).

**timestamp**

The UNIX timestamp of when the event occurred.

**operation**

The operation that triggered the event. Valid values are:

- CREATED
- UPDATED
- DELETED

**accountId**

Your AWS account ID.

**thingGroupId**

The ID of the thing group being created, updated, or deleted.

**thingGroupName**

The name of the thing group being created, updated, or deleted.

**versionNumber**

The version of the thing group. This value is set to 1 when a thing group is created. It is incremented by 1 each time the thing group is updated.

**parentGroupName**

The name of the parent thing group, if one exists.

**parentGroupId**

The ID of the parent thing group, if one exists.

**description**

A description of the thing group.

**rootToParentThingGroups**

An array of information about the parent thing group. There is one entry for each parent thing group, starting with the parent of the current thing group and continuing until the root thing group has been reached. Each entry contains the thing group name and the thing group ARN.

**attributes**

A collection of name-value pairs associated with the thing group.

#### Thing Added to or Removed from a Thing Group

The registry publishes the following event messages when a thing is added to or removed from a thing group.

- \$aws/events/thingGroupMembership/thingGroup/<thingGroupName>/thing/<thingName>/added

- \$aws/events/thingGroupMembership/thingGroup/<thingGroupName>/thing/<thingName>/removed

The messages contain the following example payload:

```
{
    "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"
}
```

The payloads contain the following attributes:

**eventType**

Set to "THING\_GROUP\_MEMBERSHIP\_EVENT".

**eventId**

The event ID.

**timestamp**

The UNIX timestamp for when the event occurred.

**operation**

ADDED when a thing is added to a thing group. REMOVED when a thing is removed from a thing group.

**accountId**

Your AWS account ID.

**groupArn**

The ARN of the thing group.

**groupId**

The ID of the group.

**thingArn**

The ARN of the thing that was added or removed from the thing group.

**thingId**

The ID of the thing that was added or removed from the thing group.

**membershipId**

An ID that represents the relationship between the thing and the thing group. This value is generated when you add a thing to a thing group.

### Thing Group Added to or Deleted from a Thing Group

The registry publishes the following event messages when a thing group is added to or removed from another thing group.

- \$aws/events/thingGroupHierarchy/thingGroup/<parentThingGroupName>/childThingGroup/<childThingGroupName>/added

- `$aws/events/thingGroupHierarchy/thingGroup/<parentThingGroupName>/childThingGroup/<childThingGroupName>/removed`

The message contains the following example payload:

```
{  
    "eventType" : "THING_GROUP_HIERARCHY_EVENT",  
    "eventId" : "264192c7-b573-46ef-ab7b-489fc47da41",  
    "timestamp" : 1234567890123,  
    "operation" : "ADDED|REMOVED",  
    "accountId" : "123456789012",  
    "thingGroupId" : "8f82a106-6b1d-4331-8984-a84db5f6f8cb",  
    "thingGroupName" : "MyRootThingGroup",  
    "childGroupId" : "06838589-373f-4312-b1f2-53f2192291c4",  
    "childGroupName" : "MyChildThingGroup"  
}
```

The payloads contain the following attributes:

`eventType`

Set to "THING\_GROUP\_HIERARCHY\_EVENT".

`eventId`

The event ID.

`timestamp`

The UNIX timestamp for when the event occurred.

`operation`

ADDED when a thing is added to a thing group. REMOVED when a thing is removed from a thing group.

`accountId`

Your AWS account ID.

`thingGroupId`

The ID of the parent thing group.

`thingGroupName`

The name of the parent thing group.

`childGroupId`

The ID of the child thing group.

`childGroupName`

The name of the child thing group.

## Jobs Events

Jobs publishes to reserved topics on the MQTT protocol when jobs are pending, completed, or canceled, and when a device reports success or failure when executing a job. Devices or management and monitoring applications can keep track of the status of jobs by subscribing to these topics. You need to use the [UpdateEventConfigurations](#) API to control what kinds of job events you receive.

Because it can take some time to cancel or delete a job, two messages are sent to indicate the start and end of a request. For example, when a cancellation request starts, a message is sent to the `$aws/events/job/jobID/cancellation_in_progress` topic. When the cancellation request is complete,

a message is sent to the `$aws/events/job/jobID/canceled` topic. A similar process occurs for a job deletion request. Management and monitoring applications can subscribe to these topics to keep track of the status of jobs.

For more information about publishing and subscribing to MQTT topics, see [Message Broker for AWS IoT \(p. 233\)](#).

#### Job Completed/Canceled/Deleted

The AWS IoT Jobs service publishes a message on an MQTT topic when a job is completed, canceled, deleted, or cancellation or deletion are in progress:

- `$aws/events/job/jobID/completed`
- `$aws/events/job/jobID/canceled`
- `$aws/events/job/jobID/deleted`
- `$aws/events/job/jobID/cancellation_in_progress`
- `$aws/events/job/jobID/deletion_in_progress`

The completed message contains the following example payload:

```
{
  "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
  }
}
```

The canceled message contains the following example payload:

```
{
  "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
}
```

The deleted message contains the following example payload:

```
{
  "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"
}
```

The cancellation\_in\_progress message contains the following example payload:

```
{
  "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"
}
```

The deletion\_in\_progress message contains the following example payload:

```
{
  "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"  
}
```

### Job Execution Terminal Status

The AWS IoT Jobs service publishes a message when a device updates a job execution to terminal status:

- \$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

The message contains the following example payload:

```
{  
  "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"  
  }  
}
```

## Lifecycle Events

AWS IoT publishes lifecycle events on the MQTT topics discussed in the following sections. These messages allow you to be notified of lifecycle events from the message broker.

### Note

Lifecycle messages might be sent out of order. You might receive duplicate messages.

## Connect/Disconnect Events

AWS IoT publishes a message to the following MQTT topics when a client connects or disconnects:

- \$aws/events/presence/connected/*clientId*: A client connected to the message broker.
- \$aws/events/presence/disconnected/*clientId*: A client disconnected from the message broker.

The following is a list of JSON elements that are contained in the connection/disconnection messages published to the \$aws/events/presence/connected/*clientId* topic.

clientId

The client ID of the connecting or disconnecting client.

**Note**

Client IDs that contain # or + do not receive lifecycle events.

clientInitiatedDisconnect

Only found in disconnection messages. True if the client initiated the disconnect. Otherwise, False.

eventType

The type of event. Valid values are connected or disconnected.

principalIdentifier

The credential used to authenticate. For TLS mutual authentication certificates, this is the certificate ID. For other connections, this is IAM credentials.

sessionIdentifier

A globally unique identifier in AWS IoT that exists for the life of the session.

timestamp

An approximation of when the event occurred, expressed in milliseconds since the Unix epoch. The accuracy of the timestamp is +/- 2 minutes.

versionNumber

The version number for the lifecycle event. This is a monotonically increasing long integer value for each client ID connection. The version number can be used by a subscriber to infer the order of lifecycle events.

**Note**

The Connect and Disconnect messages for a client connection have the same version number.

The version number might skip values and is not guaranteed to be consistently increasing by 1 for each event.

If a client is not connected for approximately one hour, the version number is reset to 0. For persistent sessions, the version number is reset to 0 after a client has been disconnected longer than the configured time-to-live (TTL) for the persistent session.

## Handling Client Disconnections

Best practice is to always have a wait state implemented for lifecycle events, including Last Will and Testament (LWT) messages. When a disconnect message is received, your code should wait a period of time and verify a device is still offline before taking action. One way to do this is by using [SQS Delay Queues](#). When a client receives a LWT or a lifecycle event, you can enqueue a message (for example, for 5 seconds). When that message becomes available and is processed (by Lambda or another service), you can first check if the device is still offline before taking further action.

## Subscribe/Unsubscribe Events

AWS IoT publishes a message to the following MQTT topic when a client subscribes or unsubscribes to an MQTT topic:

```
$aws/events/subscriptions/subscribed/clientId
```

or

```
$aws/events/subscriptions/unsubscribed/clientId
```

Where *clientId* is the MQTT client ID that connects to the AWS IoT message broker.

The message published to this topic has the following structure:

```
{  
    "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"]  
}
```

The following is a list of JSON elements that are contained in the subscribed and unsubscribed messages published to the `$aws/events/subscriptions/subscribed/clientId` and `$aws/events/subscriptions/unsubscribed/clientId` topics.

#### clientId

The client ID of the subscribing or unsubscribing client.

##### Note

Client IDs that contain # or + do not receive lifecycle events.

#### eventType

The type of event. Valid values are `subscribed` or `unsubscribed`.

#### principalIdentifier

The credential used to authenticate. For TLS mutual authentication certificates, this is the certificate ID. For other connections, this is IAM credentials.

#### sessionIdentifier

A globally unique identifier in AWS IoT that exists for the life of the session.

#### timestamp

An approximation of when the event occurred, expressed in milliseconds since the Unix epoch. The accuracy of the timestamp is +/- 2 minutes.

#### topics

An array of the MQTT topics to which the client has subscribed.

##### Note

Lifecycle messages might be sent out of order. You might receive duplicate messages.

# AWS IoT SDKs

## Contents

- [AWS Mobile SDK for Android \(p. 615\)](#)
- [Arduino Yún SDK \(p. 615\)](#)
- [AWS IoT Device SDK for Embedded C \(p. 615\)](#)
- [AWS IoT C++ Device SDK \(p. 616\)](#)
- [AWS Mobile SDK for iOS \(p. 616\)](#)
- [AWS IoT Device SDK for Java \(p. 616\)](#)
- [AWS IoT Device SDK for JavaScript \(p. 616\)](#)
- [AWS IoT Device SDK for Python \(p. 617\)](#)

The AWS IoT Device SDKs help you to easily and quickly connect your devices to AWS IoT. The AWS IoT Device SDKs include open-source libraries, developer guides with samples, and porting guides so that you can build innovative IoT products or solutions on your choice of hardware platforms.

## AWS Mobile SDK for Android

The AWS SDK for Android contains a library, samples, and documentation for developers to build connected mobile applications using AWS. This SDK also includes support for calling AWS IoT APIs. For more information, see the following:

- [AWS Mobile SDK for Android on GitHub](#)
- [AWS Mobile SDK for Android Readme](#)
- [AWS Mobile SDK for Android Samples](#)

## Arduino Yún SDK

The AWS IoT Arduino Yún SDK makes it possible for developers to connect their Arduino Yún-compatible boards to AWS IoT. By connecting a device to AWS IoT, users can securely work with the message broker, rules, and shadows provided by AWS IoT and with other AWS services like AWS Lambda, Kinesis, and Amazon S3. For more information, see the following:

- [Arduino Yún SDK on GitHub](#)
- [Arduino Yún SDK Readme](#)

## AWS IoT Device SDK for Embedded C

The AWS IoT Device SDK for Embedded C is a collection of C source files that can be used in embedded applications to securely connect to the AWS IoT platform. It includes transport clients, TLS implementations, and examples for their use. It also supports AWS IoT-specific features such as an API to access the Device Shadow service. It is distributed as source code and is intended to be built into

customer firmware along with application code, other libraries, and RTOS. For more information, see the following:

- [AWS IoT Device SDK for Embedded C GitHub](#)
- [AWS IoT Device SDK for Embedded C Readme](#)
- [AWS IoT Device SDK for Embedded C Porting Guide](#)

## AWS IoT C++ Device SDK

The AWS IoT C++ Device SDK allows developers to build connected applications using AWS and the AWS IoT APIs. Specifically, this SDK was designed for devices that are not resource constrained and required advanced features such as message queuing, multi-threading support, and the latest language features. For more information, see the following:

- [AWS IoT C++ Device SDK GitHub](#)
- [AWS IoT C++ Device SDK Readme](#)

## AWS Mobile SDK for iOS

The AWS SDK for iOS is an open-source software development kit, distributed under an Apache Open Source license. The SDK for iOS provides a library, code samples, and documentation to help developers build connected mobile applications using AWS. This SDK also includes support for calling the AWS IoT API.

- [AWS SDK for iOS on GitHub](#)
- [AWS SDK for iOS Readme](#)
- [AWS SDK for iOS Samples](#)

## AWS IoT Device SDK for Java

The AWS IoT Device SDK for Java makes it possible for Java developers to access the AWS IoT platform through MQTT or MQTT over the WebSocket protocol. The SDK is built with shadow support. You can access shadows by using HTTP methods, including GET, UPDATE, and DELETE. The SDK also supports a simplified shadow access model, which allows developers to exchange data with shadows by just using getter and setter methods, without having to serialize or deserialize any JSON documents. For more information, see the following:

- [AWS IoT Device SDK for Java on GitHub](#)
- [AWS IoT Device SDK for Java readme](#)

## AWS IoT Device SDK for JavaScript

The aws-iot-device-sdk.js package makes it possible for developers to write JavaScript applications that access AWS IoT using MQTT or MQTT over the WebSocket protocol. It can be used in Node.js environments and browser applications. For more information, see the following:

- [AWS IoT Device SDK for JavaScript on GitHub](#)
- [AWS IoT Device SDK for JavaScript readme](#)

## AWS IoT Device SDK for Python

The AWS IoT Device SDK for Python makes it possible for developers to write Python scripts to use their devices to access the AWS IoT platform through MQTT or MQTT over the WebSocket protocol. By connecting their devices to AWS IoT, users can securely work with the message broker, rules, and shadows provided by AWS IoT and with other AWS services like AWS Lambda, Kinesis, and Amazon S3, and more.

- [AWS IoT Device SDK for Python on GitHub](#)
- [AWS IoT Device SDK for Python readme](#)

# Monitoring AWS IoT

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS IoT and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure, if one occurs. Before you start monitoring AWS IoT, you should create a monitoring plan that includes answers to the following questions:

- What are your monitoring goals?
- Which resources will you monitor?
- How often will you monitor these resources?
- Which monitoring tools will you use?
- Who will perform the monitoring tasks?
- Who should be notified when something goes wrong?

The next step is to establish a baseline for normal AWS IoT performance in your environment, by measuring performance at various times and under different load conditions. As you monitor AWS IoT, store historical monitoring data so that you can compare it with current performance data, identify normal performance patterns and performance anomalies, and devise methods to address issues.

For example, if you're using Amazon EC2, you can monitor CPU utilization, disk I/O, and network utilization for your instances. When performance falls outside your established baseline, you might need to reconfigure or optimize the instance to reduce CPU utilization, improve disk I/O, or reduce network traffic.

To establish a baseline you should, at a minimum, monitor the following items:

- PublishIn.Success
- PublishOut.Success
- Subscribe.Success
- Ping.Success
- Connect.Success
- GetThingShadow.Accepted
- UpdateThingShadow.Accepted
- DeleteThingShadow.Accepted
- RulesExecuted

## Topics

- [Monitoring Tools \(p. 618\)](#)
- [Monitoring with Amazon CloudWatch \(p. 619\)](#)
- [Monitoring with CloudWatch Logs \(p. 629\)](#)
- [Logging AWS IoT API Calls with AWS CloudTrail \(p. 649\)](#)

## Monitoring Tools

AWS provides tools that you can use to monitor AWS IoT. You can configure some of these tools to do the monitoring for you. Some of the tools require manual intervention. We recommend that you automate monitoring tasks as much as possible.

## Automated Monitoring Tools

You can use the following automated monitoring tools to watch AWS IoT and report when something is wrong:

- **Amazon CloudWatch Alarms** – Watch a single metric over a time period that you specify, and perform one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon Simple Notification Service (Amazon SNS) topic or Amazon EC2 Auto Scaling policy. CloudWatch alarms do not invoke actions simply because they are in a particular state. The state must have changed and been maintained for a specified number of periods. For more information, see [Monitoring with Amazon CloudWatch \(p. 619\)](#).
- **Amazon CloudWatch Logs** – Monitor, store, and access your log files from AWS CloudTrail or other sources. For more information, see [Monitoring Log Files](#) in the *Amazon CloudWatch User Guide*.
- **Amazon CloudWatch Events** – Match events and route them to one or more target functions or streams to make changes, capture state information, and take corrective action. For more information, see [What Is Amazon CloudWatch Events](#) in the *Amazon CloudWatch User Guide*.
- **AWS CloudTrail Log Monitoring** – Share log files between accounts, monitor CloudTrail log files in real time by sending them to CloudWatch Logs, write log processing applications in Java, and validate that your log files have not changed after delivery by CloudTrail. For more information, see [Working with CloudTrail Log Files](#) in the *AWS CloudTrail User Guide*.

## Manual Monitoring Tools

Another important part of monitoring AWS IoT involves manually monitoring those items that the CloudWatch alarms don't cover. The AWS IoT, CloudWatch, and other AWS service console dashboards provide an at-a-glance view of the state of your AWS environment. We recommend that you also check the log files on AWS IoT.

- AWS IoT dashboard shows:
  - CA certificates
  - Certificates
  - Policies
  - Rules
  - Things
- CloudWatch home page shows:
  - Current alarms and status.
  - Graphs of alarms and resources.
  - Service health status.

You can use CloudWatch to do the following:

- Create [customized dashboards](#) to monitor the services you care about.
- Graph metric data to troubleshoot issues and discover trends.
- Search and browse all your AWS resource metrics.
- Create and edit alarms to be notified of problems.

## Monitoring with Amazon CloudWatch

You can monitor AWS IoT using CloudWatch, which collects and processes raw data from AWS IoT into readable, near real-time metrics. These statistics are recorded for a period of two weeks, so that you can

access historical information and gain a better perspective on how your web application or service is performing. By default, AWS IoT metric data is sent automatically to CloudWatch in one minute periods. For more information, see [What Are Amazon CloudWatch, Amazon CloudWatch Events, and Amazon CloudWatch Logs?](#) in the [Amazon CloudWatch User Guide](#).

### Topics

- [AWS IoT Metrics and Dimensions \(p. 620\)](#)
- [How Do I Use AWS IoT Metrics? \(p. 627\)](#)
- [Creating CloudWatch Alarms to Monitor AWS IoT \(p. 627\)](#)

## AWS IoT Metrics and Dimensions

When you interact with AWS IoT, the service sends the following metrics and dimensions to CloudWatch every minute. You can use the following procedures to view the metrics for AWS IoT.

### To view metrics (CloudWatch console)

Metrics are grouped first by the service namespace, and then by the various dimension combinations within each namespace.

1. Open the CloudWatch console at <https://console.aws.amazon.com/cloudwatch/>.
2. In the navigation pane, choose **Metrics**.
3. In the **CloudWatch Metrics by Category** pane, under the metrics category for AWS IoT, choose a metrics category, and then in the upper pane, scroll down to view the full list of metrics.

### To view metrics (CLI)

- At a command prompt, use the following command:

```
aws cloudwatch list-metrics --namespace "AWS/IoT"
```

CloudWatch displays the following metrics for AWS IoT:

## AWS IoT Metrics

The `AWS/IoT` namespace includes the following metrics. AWS IoT sends the following metrics to CloudWatch once per received request.

### AWS IoT Metrics

Metric	Description
RulesExecuted	The number of AWS IoT rules executed.
NumLogEventsFailedToPublishThrottled	The number of log events within the batch that have failed to publish due to throttling errors.
NumLogBatchesFailedToPublishThrottled	The singular batch of log events that has failed to publish due to throttling errors.

## Rule Metrics

Metric	Description
TopicMatch	The number of incoming messages published on a topic on which a rule is listening. The <code>RuleName</code> dimension contains the name of the rule.
ParseError	The number of JSON parse errors that occurred in messages published on a topic on which a rule is listening. The <code>RuleName</code> dimension contains the name of the rule.
RuleNotFound	The rule to be triggered could not be found. The <code>RuleName</code> dimension contains the name of the rule.
RuleMessageThrottled	The number of messages throttled by the rules engine because of malicious behavior or because the number of messages exceeds the rules engine's throttle limit. The <code>RuleName</code> dimension contains the name of the rule to be triggered.

## Rule Action Metrics

Metric	Description
Success	The number of successful rule action invocations. The <code>RuleName</code> dimension contains the name of the rule that specifies the action. The <code>ActionType</code> dimension contains the type of action that was invoked.
Failure	The number of failed rule action invocations. The <code>RuleName</code> dimension contains the name of the rule that specifies the action. The <code>RuleName</code> dimension contains the name of the rule that specifies the action. The <code>ActionType</code> dimension contains the type of action that was invoked.

## Message Broker Metrics

Metric	Description
Connect.AuthError	The number of connection requests that could not be authorized by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the CONNECT message.
Connect.ClientError	The number of connection requests rejected because the MQTT message did not meet the requirements defined in <a href="#">AWS IoT Limits</a> . The <code>Protocol</code> dimension contains the protocol used to send the CONNECT message.
Connect.ServerError	The number of connection requests that failed because an internal error occurred. The <code>Protocol</code> dimension contains the protocol used to send the CONNECT message.

Metric	Description
Connect.Success	The number of successful connections to the message broker. The <code>Protocol</code> dimension contains the protocol used to send the CONNECT message.
Connect.Throttle	The number of connection requests that were throttled because the client exceeded the allowed connect request rate. This can be the account-level connect rate or the number of connections from the same client ID. The <code>Protocol</code> dimension contains the protocol used to send the CONNECT message.
Ping.Success	The number of ping messages received by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the ping message.
PublishIn.AuthError	The number of publish requests the message broker was unable to authorize. The <code>Protocol</code> dimension contains the protocol used to publish the message.
PublishIn.ClientError	The number of publish requests rejected by the message broker because the message did not meet the requirements defined in <a href="#">AWS IoT Limits</a> . The <code>Protocol</code> dimension contains the protocol used to publish the message.
PublishIn.ServerError	The number of publish requests the message broker failed to process because an internal error occurred. The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.
PublishIn.Success	The number of publish requests successfully processed by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.
PublishIn.Throttle	The number of publish request that were throttled because the client exceeded the allowed inbound message rate. The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.
PublishOut.AuthError	The number of publish requests made by the message broker that could not be authorized by AWS IoT. The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.
PublishOut.ClientError	The number of publish requests made by the message broker that were rejected because the message did not meet the requirements defined in <a href="#">AWS IoT Limits</a> . The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.
PublishOut.Success	The number of publish requests successfully made by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the PUBLISH message.

Metric	Description
Subscribe.AuthError	The number of subscription requests made by a client that could not be authorized. The <code>Protocol</code> dimension contains the protocol used to send the <code>SUBSCRIBE</code> message.
Subscribe.ClientError	The number of subscribe requests that were rejected because the <code>SUBSCRIBE</code> message did not meet the requirements defined in <a href="#">AWS IoT Limits</a> . The <code>Protocol</code> dimension contains the protocol used to send the <code>SUBSCRIBE</code> message.
Subscribe.ServerError	The number of subscribe requests that were rejected because an internal error occurred. The <code>Protocol</code> dimension contains the protocol used to send the <code>SUBSCRIBE</code> message.
Subscribe.Success	The number of subscribe requests that were successfully processed by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the <code>SUBSCRIBE</code> message.
Subscribe.Throttle	The number of subscribe requests that were throttled because the client exceeded the allowed subscribe request rate. The <code>Protocol</code> dimension contains the protocol used to send the <code>SUBSCRIBE</code> message.
Unsubscribe.ClientError	The number of unsubscribe requests that were rejected because the <code>UNSUBSCRIBE</code> message did not meet the requirements defined in <a href="#">AWS IoT Limits</a> . The <code>Protocol</code> dimension contains the protocol used to send the <code>UNSUBSCRIBE</code> message.
Unsubscribe.ServerError	The number of unsubscribe requests that were rejected because an internal error occurred. The <code>Protocol</code> dimension contains the protocol used to send the <code>UNSUBSCRIBE</code> message.
Unsubscribe.Success	The number of unsubscribe requests that were successfully processed by the message broker. The <code>Protocol</code> dimension contains the protocol used to send the <code>UNSUBSCRIBE</code> message.
Unsubscribe.Throttle	The number of unsubscribe requests that were rejected because the client exceeded the allowed unsubscribe request rate. The <code>Protocol</code> dimension contains the protocol used to send the <code>UNSUBSCRIBE</code> message.

#### Note

The message broker metrics are displayed in the AWS IoT console under **Protocol Metrics**.

## Device Shadow Metrics

Metric	Description
DeleteThingShadow.Accepted	The number of DeleteThingShadow requests processed successfully. The <code>Protocol</code> dimension contains the protocol used to make the request.
GetThingShadow.Accepted	The number of GetThingShadow requests processed successfully. The <code>Protocol</code> dimension contains the protocol used to make the request.
UpdateThingShadow.Accepted	The number of UpdateThingShadow requests processed successfully. The <code>Protocol</code> dimension contains the protocol used to make the request.

### Note

The device shadow metrics are displayed in the AWS IoT console under **Protocol Metrics**.

## Jobs Metrics

Metric	Description
ServerError	The number of server errors generated while executing the job. The <code>JobId</code> dimension contains the ID of the job.
ClientError	The number of client errors generated while executing the job. The <code>JobId</code> dimension contains the ID of the job.
QueuedJobExecutionTotalCount	The total number of job executions whose status is <code>QUEUED</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
InProgressJobExecutionTotalCount	The total number of job executions whose status is <code>IN_PROGRESS</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
FailedJobExecutionTotalCount	The total number of job executions whose status is <code>FAILED</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
SucceededJobExecutionTotalCount	The total number of job executions whose status is <code>SUCCESS</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
CanceledJobExecutionTotalCount	The total number of job executions whose status is <code>CANCELED</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
RejectedJobExecutionTotalCount	The total number of job executions whose status is <code>REJECTED</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.
RemovedJobExecutionTotalCount	The total number of job executions whose status is <code>REMOVED</code> for the given job. The <code>JobId</code> dimension contains the ID of the job.

Metric	Description
QueuedJobExecutionCount	The number of job executions whose status has changed to <code>QUEUED</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
InProgressJobExecutionCount	The number of job executions whose status has changed to <code>IN_PROGRESS</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
FailedJobExecutionCount	The number of job executions whose status has changed to <code>FAILED</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
SucceededJobExecutionCount	The number of job executions whose status has changed to <code>SUCCESS</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
CanceledJobExecutionCount	The number of job executions whose status has changed to <code>CANCELED</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
RejectedJobExecutionCount	The number of job executions whose status has changed to <code>REJECTED</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.
RemovedJobExecutionCount	The number of job executions whose status has changed to <code>REMOVED</code> within a time period that is determined by CloudWatch. (For more information about CloudWatch metrics, see <a href="#">Amazon CloudWatch Metrics</a> .) The <code>JobId</code> dimension contains the ID of the job.

## Device Defender Audit Metrics

Metric	Description
NonCompliantResources	The number of resources that were found to be noncompliant with a check. The system reports the

Metric	Description
	number of resources that were out of compliance for each check of each audit performed.
ResourcesEvaluated	The number of resources that were evaluated for compliance. The system reports the number of resources that were evaluated for each check of each audit performed.

### Device Defender Detect Metrics

Metric	Description
Violations	The number of new violations of security profile behaviors that have been found since the last time an evaluation was performed. The system reports the number of new violations for the account, for a specific security profile, and for a specific behavior of a specific security profile.
ViolationsCleared	The number of violations of security profile behaviors that have been resolved since the last time an evaluation was performed. The system reports the number of resolved violations for the account, for a specific security profile, and for a specific behavior of a specific security profile.
ViolationsInvalidated	The number of violations of security profile behaviors for which information is no longer available since the last time an evaluation was performed (because the reporting device stopped reporting, or is no longer being monitored for some reason). The system reports the number of invalidated violations for the entire account, for a specific security profile, and for a specific behavior of a specific security profile.

## Dimensions for Metrics

Metrics use the namespace and provide metrics for the following dimensions:

Dimension	Description
ActionType	The <a href="#">action type</a> specified by the rule that triggered the request.
Protocol	The protocol used to make the request. Valid values are: MQTT or HTTP
RuleName	The name of the rule triggered by the request.
JobId	The ID of the job whose progress or message connection success/failure is being monitored.
CheckName	The name of the Device Defender audit check whose results are being monitored.

Dimension	Description
ScheduledAuditName	The name of the Device Defender scheduled audit whose check results are being monitored. This has the value "OnDemand" if the results reported are for an audit that was performed on demand.
SecurityProfileName	The name of the Device Defender Detect security profile whose behaviors are being monitored.
BehaviorName	The name of the Device Defender Detect security profile behavior that is being monitored.

## How Do I Use AWS IoT Metrics?

The metrics reported by AWS IoT provide information that you can analyze in different ways. The following use cases are based on a scenario where you have ten things that connect to the internet once a day. Each day:

- Ten things connect to AWS IoT at roughly the same time.
- Each thing subscribes to a topic filter, and then waits for an hour before disconnecting. During this period, things communicate with one another and learn more about the state of the world.
- Each thing publishes some perception it has based on its newly found data using `UpdateThingShadow`.
- Each thing disconnects from AWS IoT.

These are suggestions to get you started, not a comprehensive list.

- [How can I be notified if my things do not connect successfully each day? \(p. 627\)](#)
- [How can I be notified if my things are not publishing data each day? \(p. 628\)](#)
- [How can I be notified if my thing's shadow updates are being rejected each day? \(p. 629\)](#)

## Creating CloudWatch Alarms to Monitor AWS IoT

You can create a CloudWatch alarm that sends an Amazon SNS message when the alarm changes state. An alarm watches a single metric over a time period you specify and performs one or more actions based on the value of the metric relative to a given threshold over a number of time periods. The action is a notification sent to an Amazon SNS topic or Auto Scaling policy. Alarms trigger actions for sustained state changes only. CloudWatch alarms do not trigger actions simply because they are in a particular state; the state must have changed and been maintained for a specified number of periods.

### How can I be notified if my things do not connect successfully each day?

1. Create an Amazon SNS topic, `arn:aws:sns:us-east-1:123456789012:things-not-connecting-successfully`.

For more information, see [Set Up Amazon Simple Notification Service](#).

2. Create the alarm.

```
Prompt>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 \
--unit Count \
--evaluation-periods 1 \
--alarm-actions arn:aws:sns:us-east-1:1234567890:things-not-connecting-successfully
```

3. Test the alarm.

```
Prompt>aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
"initializing" --state-value OK
```

```
Prompt>aws cloudwatch set-alarm-state --alarm-name ConnectSuccessAlarm --state-reason
"initializing" --state-value ALARM
```

## How can I be notified if my things are not publishing data each day?

1. Create an Amazon SNS topic, arn:aws:sns:us-east-1:123456789012:things-not-publishing-data.

For more information, see [Set Up Amazon Simple Notification Service](#).

2. Create the alarm.

```
Prompt>aws cloudwatch put-metric-alarm \
--alarm-name PublishInSuccessAlarm\
--alarm-description "Alarm when my Things don't publish their data" \
--namespace AWS/IoT \
--metric-name Publish.In.Success \
--dimensions Name=Protocol,Value=MQTT \
--statistic Sum \
--threshold 10 \
--comparison-operator LessThanThreshold \
--period 86400 \
--unit Count \
--evaluation-periods 1 \
--alarm-actions arn:aws:sns:us-east-1:1234567890:things-not-publishing-data
```

3. Test the alarm.

```
Prompt>aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
"initializing" --state-value OK
```

```
Prompt>aws cloudwatch set-alarm-state --alarm-name PublishInSuccessAlarm --state-reason
"initializing" --state-value ALARM
```

## How can I be notified if my thing's shadow updates are being rejected each day?

1. Create an Amazon SNS topic, arn:aws:sns:us-east-1:1234567890:things-shadow-updates-rejected.

For more information, see [Set Up Amazon Simple Notification Service](#).

2. Create the alarm.

```
Prompt>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 arn:aws:sns:us-east-1:1234567890:things-shadow-updates-rejected
```

3. Test the alarm.

```
Prompt>aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-reason "initializing" --state-value OK
```

```
Prompt>aws cloudwatch set-alarm-state --alarm-name UpdateThingShadowSuccessAlarm --state-reason "initializing" --state-value ALARM
```

## Monitoring with CloudWatch Logs

AWS IoT sends progress events about each message as it passes from your devices through the message broker and rules engine. To view these logs, you must configure AWS IoT to generate the logs used by CloudWatch.

For more information about CloudWatch Logs, see [CloudWatch Logs](#). For information about supported AWS IoT CloudWatch Logs, see [CloudWatch Log Entry Format \(p. 634\)](#).

To enable AWS IoT logging, you must create an IAM role, register the role with AWS IoT, and then configure AWS IoT logging.

### Note

Before you enable AWS IoT logging, make sure you understand the CloudWatch Logs access permissions. Users with access to CloudWatch Logs can see debugging information from your devices. For more information, see [Authentication and Access Control for Amazon CloudWatch Logs](#).

## Create a Logging Role

Use the [IAM console](#) to create a logging role.

1. From the navigation pane, choose **Roles**, and then choose **Create new role**.
2. Choose **AWS Service Role**, and then for service role type, choose **AWS IoT**.

3. Choose the **AWSIoTLogging** role, and then choose **Next Step**.
4. Enter a name and description for the role, and then choose **Create role**.

## Logging Role Policy

The following policy documents provide the role policy and trust policy that allow AWS IoT to submit logs to CloudWatch on your behalf.

### Note

These documents were created for you when you created the logging role.

Role policy:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:CreateLogGroup",  
                "logs:CreateLogStream",  
                "logs:PutLogEvents",  
                "logs:PutMetricFilter",  
                "logs:PutRetentionPolicy"  
            ],  
            "Resource": [  
                "*"  
            ]  
        }  
    ]  
}
```

Trust policy:

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

## Log Level

The log level specifies which types of logs are generated.

ERROR

Any error that causes an operation to fail.

Logs include ERROR information only.

**WARN**

Anything that can potentially cause inconsistencies in the system, but might not cause the operation to fail.

Logs include ERROR and WARN information.

**INFO**

High-level information about the flow of things.

Logs include INFO, ERROR, and WARN information.

**DEBUG**

Information that might be helpful when debugging a problem.

Logs include DEBUG, INFO, ERROR, and WARN information.

**DISABLED**

All logging is disabled.

## Configure AWS IoT Logging

You can use the AWS IoT console, the [set-v2-logging-options](#) CLI command, or the `SetV2LoggingOptions` API to enable logging. The principal used to make the API call must have [Pass Role Permissions \(p. 245\)](#) for your logging role. The logging role is passed to [set-v2-logging-options](#) or `SetV2LoggingOptions` as the `roleARN` parameter.

You can configure logging to be global or fine-grained. Global logging sets one logging level for all logs no matter what resource triggered the logs. Fine-grained logging allows you to set a logging level for a specific resource or set of resources. Currently, only thing groups are supported. You can use the AWS IoT console, the CLI, or the API to enable global logging. You must use the CLI or API to enable fine-grained logging.

### Global Logging

Use the `set-v2-logging-options` CLI command to set the logging options for your account. `set-v2-logging-options` takes three arguments:

`--role-arn`

Your logging role ARN. The logging role grants AWS IoT permission to write to your logs in CloudWatch Logs.

`--default-log-level`

The log level to use. Valid values are: ERROR, WARN, INFO, DEBUG, or DISABLED

`--disable-all-logs | --no-disable-all-logs`

When set to true (`--disable-all-logs`) disables all logs. The default (parameter not used) is false.

For example:

```
aws iot set-v2-logging-options \
--role-arn arn:aws:iam::<your-aws-account-num>:role/<IoTLoggingRole> \
--default-log-level <INFO>
```

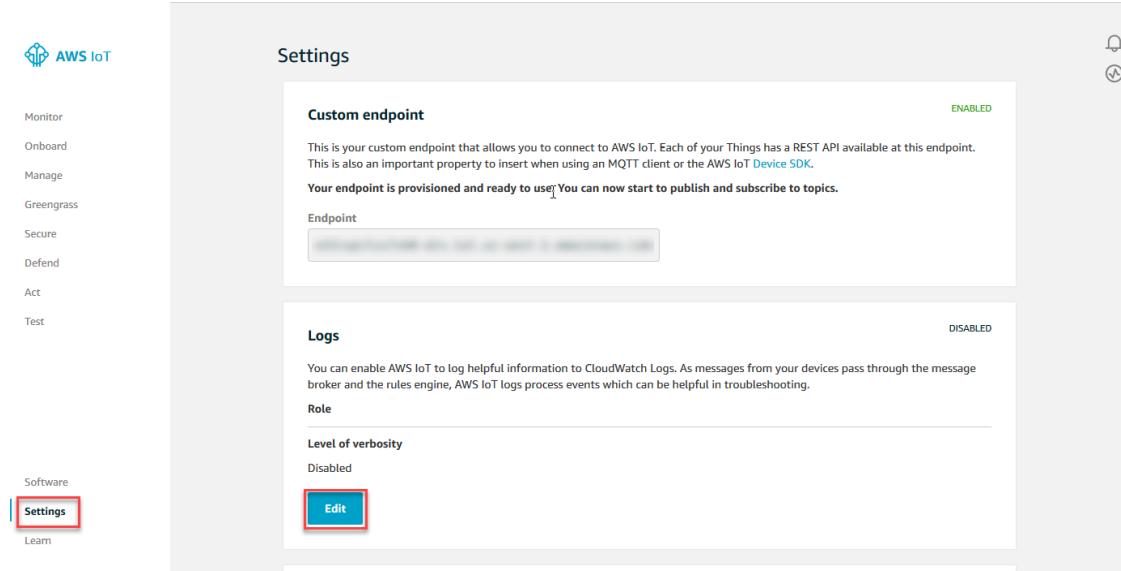
You can use the `get-v2-logging-options` CLI command to get the current logging options.

**Note**

AWS IoT continues to support older commands (`set-logging-options` and `get-logging-options`) to set and get global logging on your account. Be aware that when these commands are used, the resulting logs contain plain-text, rather than JSON payloads and logging latency is generally higher. No further improvements will be made to the implementation of these older commands. We recommend that you use the "v2" versions to configure your logging options and, when possible, change legacy applications that use the older versions.

**To configure global logging by using the AWS IoT console**

1. Sign in to the AWS IoT console. For more information, see [Sign in to the AWS IoT Console \(p. 5\)](#).
2. In the left navigation pane, choose **Settings**.



3. In the **Logs** section of the **Settings** page, choose **Edit**. The **Logs** section displays your settings for role and level of verbosity.
4. On the **Configure role setting** page, choose the level of verbosity that describes the level of detail that you want to appear in the CloudWatch logs.

The screenshot shows the 'Configure role setting' page. 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 red box highlights this dropdown. Underneath is a 'Set role' section with a note about selecting a role for CloudWatch Logs. It shows 'No role selected' and two buttons: 'Create Role' and 'Select'. A red box highlights the 'Select' button. At the bottom, there are 'Cancel' and 'Update' buttons, with a red box highlighting the 'Update' button.

5. Choose **Select** to specify a role that you created previously, or **Create Role** to create a role to use for logging.
6. Choose **Update** to save your changes.

Review your CloudWatch logs to see if you are satisfied with the level of collected information. If not, you can always change the logging level later.

## Fine-Grained Logging

Fine-grained logging allows you to specify a logging level for a target. A target is defined by a resource type and a resource name. Currently, AWS IoT supports thing groups as targets. Fine-grained logging allows you to set a logging level for a specific thing group. Say we have a thing group called "Phones" that contains things that represent different kinds of phones. We then create another thing group called "MobilePhones" and make it a child of the "Phones" thing group. Fine-grained logging allows you to configure one logging level for all things in the "Phones" group (and any child groups) and another logging level for things in the "MobilePhones" group. In this example, we have assigned two different logging levels to things in the "MobilePhones" group — one from the logging level for the "Phones" thing group and another from the "MobilePhones" thing group — but the logging level specified for the child thing group will override the logging level specified for the parent thing group.

Use the `set-v2-logging-options` CLI command to enable fine-grained logging and set the default logging level. It takes the following optional arguments:

`--role-arn`

An IAM role that allows AWS IoT to write to your CloudWatch Logs. If not specified, AWS IoT uses the logging role associated with your account. The logging role is associated with your account when it is created. For more information, see [Create a Logging Role \(p. 629\)](#).

`--default-log-level`

The logging level used if not specified. Valid values are: DEBUG, INFO, ERROR, WARN, and DISABLED.

`--disable-all-logs | --no-disable-all-logs`

When set to true (`--disable-all-logs`), disables all logs. The default (parameter not used) is false.

The `get-v2-logging-options` CLI command returns the configured IAM logging role, the default logging level, and the `disableAllLogs` value.

Use the `set-v2-logging-level` CLI command to configure fine-grained logging for a target. It takes the following arguments:

`--log-target`

A JSON object that contains the resource type (field `targetType`) and name (field `targetName`) of the entity for which you are configuring logging. AWS IoT currently supports `THING_GROUP` for the resource type. You can configure up to 10 logging targets.

`--log-level`

The logging level used when generating logs for the specified resource. Valid values are: DEBUG, INFO, ERROR, WARN, and DISABLED

Use the `list-v2-logging-levels` CLI command to get a list of the currently configured fine-grained logging levels. Call the `delete-v2-logging-level` CLI command to delete a logging level. Use the `delete-v2-logging-level` command to delete a fine-grained logging level.

## CloudWatch Log Entry Format

Each component of AWS IoT generates its own logs. Each log entry has an `eventType` that indicates which operation caused the log to be generated. This section describes the logs generated by the following AWS IoT components:

- [Message broker \(p. 634\)](#)
- [Device Shadow service \(p. 638\)](#)
- [Rules engine \(p. 640\)](#)
- [Jobs \(p. 645\)](#)

All CloudWatch Logs have the following common attributes:

`timestamp`

The UNIX timestamp of when the client connected to the AWS IoT message broker.

`logLevel`

The log level being used. For more information, see [the section called "Log Level" \(p. 630\)](#).

`traceId`

A randomly generated identifier that can be used to correlate all logs for a specific request.

`accountId`

Your AWS account ID.

`status`

The status of the request.

`eventType`

The event type for which the log was generated. The value of the event type for each event is listed in the following sections.

## Message Broker Logs

The AWS IoT message broker generates logs for the following events:

[Connect Log](#)

The AWS IoT message broker generates a Connect log when an MQTT client connects.

[more info \(1\)](#)

For example:

```
{  
    "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
```

}

In addition to the attributes common to CloudWatch Logs, Connect log entries contain the following attributes:

**eventType**

Connect for connection logs.

**protocol**

The protocol used when making the request. Valid values are MQTT or HTTP.

**clientId**

The ID of the client making the request.

**principalId**

The ID of the principal making the request.

**sourcelp**

The IP address where the request originated.

**sourcePort**

The port where the request originated.

## Subscribe Log

The AWS IoT message broker generates a Subscribe log when an MQTT client subscribes to a topic.  
[more info \(2\)](#)

For example:

```
{  
  "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  
}
```

In addition to the attributes common to CloudWatch Logs, Subscribe log entries contain the following attributes:

**eventType**

Subscribe for subscription logs.

**protocol**

The protocol used when making the request. Valid values are MQTT or HTTP.

**topicName**

The name of the subscribed topic.

**clientId**

The ID of the client making the request.

**principalId**

The ID of the principal making the request.

**sourceIp**

The IP address where the request originated.

**sourcePort**

The port where the request originated.

**Publish-In Log**

When the AWS IoT message broker receives an MQTT message, it generates a Publish-In log.

[more info \(3\)](#)

For example:

```
{  
    "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  
}
```

In addition to the attributes common to CloudWatch Logs, Publish-In log entries contain the following attributes:

**eventType**

Publish-In when the message broker receives a message.

**status**

The status of the request.

**protocol**

The protocol used when making the request. Valid values are MQTT or HTTP.

**topicName**

The name of the subscribed topic.

**clientId**

The ID of the client making the request.

**principalId**

The ID of the principal making the request.

sourcelp

The IP address where the request originated.

sourcePort

The port where the request originated.

### Publish-Out Log

When the message broker publishes an MQTT message, it generates a Publish-Out log.

[more info \(4\)](#)

For example:

```
{  
    "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  
}
```

In addition to the attributes common to CloudWatch Logs, Publish-Out log entries contain the following attributes:

eventType

Publish-Out when the message broker publishes a message.

status

The status of the request.

protocol

The protocol used when making the request. Valid values are MQTT or HTTP.

topicName

The name of the subscribed topic.

clientId

The ID of the client making the request.

principalId

The ID of the principal making the request.

sourcelp

The IP address where the request originated.

sourcePort

The port where the request originated.

## Disconnect Log

The AWS IoT message broker generates a **Disconnect** log when an MQTT client disconnects.  
[more info \(5\)](#)

For example:

```
{  
    "timestamp": "2017-08-10 15:37:23.476",  
    "logLevel": "INFO",  
    "traceId": "20b23f3f-d7f1-faea-169f-82263394fbdb",  
    "accountId": "123456789012",  
    "status": "Success",  
    "eventType": "Disconnect",  
    "protocol": "MQTT",  
    "clientId": "abf27092886e49a8a5c1922749736453",  
    "principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",  
    "sourceIp": "205.251.233.181",  
    "sourcePort": 13490  
}
```

In addition to the attributes common to CloudWatch Logs, **Disconnect** log entries contain the following attributes:

**eventType**

Disconnect for connection logs.

**protocol**

The protocol used when making the request. Valid values are **MQTT** or **HTTP**.

**clientId**

The ID of the client making the request.

**principalId**

The ID of the principal making the request.

**sourceIp**

The IP address where the request originated.

**sourcePort**

The port where the request originated.

## Device Shadow Logs

The AWS IoT Device Shadow service generates logs for the following events:

### GetThingShadow Logs

The Device Shadow service generates a **GetThingShadow** log when a get request for a shadow is received.

[more info \(6\)](#)

For example:

```
{  
    "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"
}
```

In addition to the attributes common to CloudWatch Logs, GetThingShadow log entries contain the following attributes:

**eventType**

GetThingShadow for GetThingShadow logs.

**protocol**

The protocol used when making the request. Valid values are **MQTT** or **HTTP**.

**deviceShadowName**

The name of the requested shadow.

**topicName**

The name of the topic on which the request was published.

### UpdateThingShadow Logs

The Device Shadow service generates a UpdateThingShadow log when a request to update a device's shadow is received.

[more info \(7\)](#)

For example:

```
{
    "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"
}
```

In addition to the attributes common to CloudWatch Logs, UpdateThingShadow log entries contain the following attributes:

**eventType**

UpdateThingShadow for update shadow logs.

**protocol**

The protocol used when making the request. Valid values are **MQTT** or **HTTP**.

**deviceShadowName**

The name of the shadow to update.

**topicName**

The name of the topic on which the request was published.

## DeleteThingShadow Logs

The Device Shadow service generates a DeleteThingShadow log when a request to delete a device's shadow is received.

[more info \(8\)](#)

For example:

```
{  
    "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"  
}
```

In addition to the attributes common to CloudWatch Logs, DeleteThingShadow log entries contain the following attributes:

**eventType**

DeleteThingShadow for DeleteThingShadow logs.

**protocol**

The protocol used when making the request. Valid values are MQTT or HTTP.

**deviceShadowName**

The name of the shadow to update.

**topicName**

The name of the topic on which the request was published.

## Rules Engine Logs

The AWS IoT Rules Engine service generates logs for the following events:

### Rule Match Logs

The AWS IoT rules engine generates a RuleMatch log when the message broker receives a message that matches a rule.

[more info \(9\)](#)

For example:

```
{  
    "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"  
}
```

```
}
```

In addition to the attributes common to CloudWatch Logs, RuleMatch log entries contain the following attributes:

**eventType**

RuleMatch for rule match logs.

**clientId**

The ID of the client making the request.

**topicName**

The name of the subscribed topic.

**ruleName**

The name of the matching rule.

**principalId**

The ID of the principal making the request.

### Function Execution Logs

The rules engine generates a FunctionExecution log when a rule's SQL query calls an external function. An external function is called when a rule's action makes an HTTP request to AWS IoT or another web service (for example, calling `get_thing_shadow` or `machinelearning_predict`).

[more info \(10\)](#)

A FunctionExecution log looks like the following:

```
{
    "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"
}
```

In addition to the attributes common to CloudWatch Logs, FunctionExecution log entries contain the following attributes:

**eventType**

FunctionExecution for rule match logs.

**clientId**

N/A for FunctionExecution logs.

**topicName**

The name of the subscribed topic.

**ruleName**

The name of the matching rule.

**resources**

A collection of resources used by the rule's actions.

**principalId**

The ID of the principal making the request.

## Starting Execution Logs

When the AWS IoT rules engine starts to trigger a rule's action, it generates a `StartingExecution` log.

[more info \(11\)](#)

For example:

```
{  
    "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"  
}
```

In addition to the attributes common to CloudWatch Logs, `StartingExecution` log entries contain the following attributes:

**eventType**

`StartingRuleExecution` for starting rule execution logs.

**clientId**

The ID of the client making the request.

**topicName**

The name of the subscribed topic.

**ruleName**

The name of the matching rule.

**ruleAction**

The name of the action triggered.

**principalId**

The ID of the principal making the request.

## Rule Execution Logs

When the AWS IoT rules engine triggers a rule's action, it generates a `RuleExecution` log.

[more info \(12\)](#)

For example:

```
{  
    "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"  
}
```

In addition to the attributes common to CloudWatch Logs, RuleExecution log entries contain the following attributes:

**eventType**

RuleExecution for rule execution logs.

**clientId**

The ID of the client making the request.

**topicName**

The name of the subscribed topic.

**ruleName**

The name of the matching rule.

**ruleAction**

The name of the action triggered.

**resources**

A collection of resources used by the rule's actions.

**principalId**

The ID of the principal making the request.

## Rule Not Found Logs

When the AWS IoT rules engine cannot find a rule with a given name, it generates a RuleNotFound error log.

[more info \(13\)](#)

For example:

```
{  
    "timestamp": "2017-10-04 19:25:46.070",  
    "logLevel": "ERROR",  
    "traceId": "30aa7ccc-1d23-0b97-aa7b-76196d83537e",  
}
```

```
"accountId": "123456789012",
"status": "Failure",
"eventType": "RuleNotFound",
"clientId": "abf27092886e49a8a5c1922749736453",
"topicName": "$aws/rules/example_rule",
"ruleName": "example_rule",
"principalId": "145179c40e2219e18a909d896a5340b74cf97a39641beec2fc3eeafc5a932167",
"reason": "RuleNotFound",
"details": "Rule example_rule not found"
}
```

In addition to the attributes common to CloudWatch Logs, `RuleNotFound` log entries contain the following attributes:

`eventType`

The string `RuleNotFound` for rule-not-found logs.

`clientId`

The ID of the client making the request.

`topicName`

The name of the topic that was published.

`ruleName`

The name of the rule that could not be found.

`principalId`

The ID of the principal making the request.

`reason`

The string `"RuleNotFound"`.

`details`

A brief explanation of the error.

### Rule Message Throttled Logs

When a message is throttled, the AWS IoT rules engine generates a `RuleMessageThrottled` error log.

[more info \(14\)](#)

For example:

```
{
  "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"
}
```

In addition to the attributes common to CloudWatch Logs, `RuleMessageThrottled` log entries contain the following attributes:

`eventType`

`RuleMessageThrottled` for rule message throttled logs.

`clientId`

The ID of the client making the request.

`topicName`

The name of the topic that was published.

`ruleName`

The name of the rule to be triggered.

`principalId`

The ID of the principal making the request.

`reason`

The string "RuleMessageThrottled".

`details`

A brief explanation of the error.

## Job Logs

The AWS IoT Job service generates logs for the following events. Logs are generated when an MQTT or HTTP request is received from the device.

### Get Pending Job Execution Logs

The AWS IoT Jobs service generates a `GetJobExecution` log when the service receives a job execution request.

[more info \(16\)](#)

For example:

```
{  
    "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."  
}
```

In addition to the attributes common to CloudWatch Logs, `GetPendingJobExecution` log entries contain the following attributes:

`eventType`

`GetPendingJobExecution` for get pending job execution logs.

protocol

The protocol used when making the request. Valid values are `MQTT` or `HTTP`.  
`clientId`

The ID of the client making the request.

`topicName`

The name of the subscribed topic.  
`clientToken`

A unique, case sensitive identifier to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

`details`

Other information from the Jobs service.

## Describe Job Execution Logs

The AWS IoT Jobs service generates a `DescribeJobExecution` log when the service receives a request to describe a job execution.

[more info \(17\)](#)

For example:

```
{  
    "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."  
}
```

In addition to the attributes common to CloudWatch Logs, `GetJobExecution` log entries contain the following attributes:

`eventType`

`DescribeJobExecution` for describe job execution logs.  
`protocol`

The protocol used when making the request. Valid values are `MQTT` or `HTTP`.  
`clientId`

The ID of the client making the request.

`jobId`

The job ID for the job execution.  
`topicName`

The topic used to make the request.

**clientToken**

A unique, case sensitive identifier to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

**details**

Other information from the Jobs service.

### Update Job Execution Logs

The AWS IoT Jobs service generates an `UpdateJobExecution` log when the service receives a request to update a job execution.

[more info \(18\)](#)

For example:

```
{  
    "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."  
}
```

In addition to the attributes common to CloudWatch Logs, `UpdateJobExecution` log entries contain the following attributes:

**eventType**

`UpdateJobExecution` for update job execution logs.

**protocol**

The protocol used when making the request. Valid values are `MQTT` or `HTTP`.

**clientId**

The ID of the client making the request.

**jobId**

The job ID for the job execution.

**topicName**

The topic used to make the request.

**clientToken**

A unique, case sensitive identifier to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

**versionNumber**

The version of the job execution.

## details

Other information from the Jobs service.

### Start Next Pending Job Execution Logs

When it receives a request to start the next pending job execution, the AWS IoT Jobs service generates a `StartNextPendingJobExecution` log.

[more info \(19\)](#)

For example:

```
{  
    "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."  
}
```

In addition to the attributes common to CloudWatch Logs, `StartNextPendingJobExecution` log entries contain the following attributes:

`eventType`

`StartNextPendingJobExecution` for start-next-pending-job execution logs.

`protocol`

The protocol used when making the request. Valid values are `MQTT` or `HTTP`.

`clientId`

The ID of the client making the request.

`topicName`

The topic used to make the request.

`clientToken`

A unique, case sensitive identifier to ensure the idempotency of the request. For more information, see [How to Ensure Idempotency](#).

`details`

Other information from the Jobs service.

### Report Final Job Execution Count Logs

The AWS IoT Jobs service generates a `ReportFinalJobExecutionCount` log when a job is completed.

[more info \(20\)](#)

For example:

```
{  
    "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"
}
```

In addition to the attributes common to CloudWatch Logs, `ReportFinalJobExecutionCount` log entries contain the following attributes:

`eventType`

`ReportFinalJobExecutionCount` for report-final-job-execution-count logs.

`jobId`

The job ID for the job execution.

`details`

Other information from the Jobs service.

## Viewing Logs

### To view your logs

1. Browse to <https://console.aws.amazon.com/cloudwatch/>. In the navigation pane, choose **Logs**.
2. In the **Filter** text box, enter **AWSIoTLogsV2**, and then press Enter.
3. Double-click the `AWSIoTLogsV2` log group.
4. Choose **Search Log Group**. A complete list of the AWS IoT logs generated for your account is displayed.
5. Choose the expand icon to look at an individual stream.

You can also enter a query in the **Filter events** text box. Here are some interesting queries to try:

- `{ $.logLevel = "INFO" }`

Find all logs that have a log level of `INFO`.

- `{ $.status = "Success" }`

Find all logs that have a status of `Success`.

- `{ $.status = "Success" && $.eventType = "GetThingShadow" }`

Find all logs that have a status of `Success` and an event type of `GetThingShadow`.

For more information about creating filter expressions, see [CloudWatch Logs Queries](#).

## Logging AWS IoT API Calls with AWS CloudTrail

AWS IoT is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS IoT. CloudTrail captures all API calls for AWS IoT as events, including

calls from the AWS IoT console and from code calls to the AWS IoT APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for AWS IoT. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to AWS IoT, the IP address from which the request was made, who made the request, when it was made, and other details.

To learn more about CloudTrail, see the [AWS CloudTrail User Guide](#).

## AWS IoT Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS IoT, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see [Viewing Events with CloudTrail Event History](#).

For an ongoing record of events in your AWS account, including events for AWS IoT, create a trail. A trail enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all AWS Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. You can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see:

- [Overview for Creating a Trail](#)
- [CloudTrail Supported Services and Integrations](#)
- [Configuring Amazon SNS Notifications for CloudTrail](#)
- [Receiving CloudTrail Log Files from Multiple Regions](#) and [Receiving CloudTrail Log Files from Multiple Accounts](#)

### Note

AWS IoT data plane actions (device side) are not logged by CloudTrail. Use CloudWatch to monitor these.

AWS IoT control plane actions *are* logged by CloudTrail. For example, calls to the **CreateThing**, **ListThings**, and **ListTopicRules** sections generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the [CloudTrail userIdentity Element](#). AWS IoT actions are documented in the [AWS IoT API Reference](#).

## Understanding AWS IoT Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the **AttachPolicy** action.

```
{  
    "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": ""  
}
```

# Troubleshooting AWS IoT

The following information might help you troubleshoot common issues in AWS IoT.

## Tasks

- [Diagnosing Connectivity Issues \(p. 652\)](#)
- [Diagnosing Rules Issues \(p. 652\)](#)
- [Diagnosing Problems with Shadows \(p. 653\)](#)
- [Diagnosing Salesforce IoT Input Stream Action Issues \(p. 654\)](#)
- [AWS IoT Limits \(p. 655\)](#)
- [AWS IoT Errors \(p. 655\)](#)

## Diagnosing Connectivity Issues

### Authentication

How do my devices authenticate AWS IoT endpoints?

Add the AWS IoT CA certificate to your client's trust store. Refer to the documentation on [Server Authentication in AWS IoT Core](#) and then follow the links to download the appropriate CA certificate.

How can I validate a correctly configured certificate?

Use the OpenSSL `s_client` command to test a connection to the AWS IoT endpoint:

```
openssl s_client -connect custom_endpoint.iot.us-east-1.amazonaws.com:8443 -CAfile CA.pem -cert cert.pem -key privateKey.pem
```

### Authorization

I received a PUBNACK or SUBNACK response from the broker. What do I do?

Make sure that there is a policy attached to the certificate you are using to call AWS IoT. All publish/subscribe operations are denied by default.

## Diagnosing Rules Issues

CloudWatch Logs are the best way to debug issues you are having with rules. For more information about using CloudWatch Logs with AWS IoT, see [Monitoring with CloudWatch Logs \(p. 629\)](#). When you enable CloudWatch Logs for AWS IoT, you can see which rules are triggered and their success or failure. You also get information about whether WHERE clause conditions match.

The most common rules issue is authorization. The logs show if your role is not authorized to perform `AssumeRole` on the resource. Here is an example log generated by [fine-grained logging \(p. 633\)](#):

```
{
    "timestamp": "2017-12-09 22:49:17.954",
    "logLevel": "ERROR",
    "traceId": "ff563525-6469-506a-e141-78d40375fc4e",
    "accountId": "123456789012",
    "status": "Failure",
    "eventType": "RuleExecution",
    "clientId": "iotconsole-123456789012-3",
    "topicName": "test-topic",
    "ruleName": "rule1",
    "ruleAction": "DynamoAction",
    "resources": {
        "ItemHashKeyField": "id",
        "Table": "trashbin",
        "Operation": "Insert",
        "ItemHashKeyValue": "id",
        "IsPayloadJSON": "true"
    },
    "principalId": "ABCDEFG1234567ABCD890:outis",
    "details": "User: arn:aws:sts::123456789012:assumed-role/dynamo-testbin/5aUMInJH is not authorized to perform: dynamodb:PutItem on resource: arn:aws:dynamodb:us-east-1:123456789012:table/testbin (Service: AmazonDynamoDBv2; Status Code: 400; Error Code: AccessDeniedException; Request ID: AKQJ987654321AKQJ123456789AKQJ987654321AKQJ987654321)"
}
```

Here is a similar example log generated by [global logging \(p. 631\)](#):

```
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.
```

For more information, see [the section called “Viewing Logs” \(p. 649\)](#).

External services are controlled by the end user. Before rule execution, make sure external services are set up with enough throughput and capacity units.

## Diagnosing Problems with Shadows

### Diagnosing Shadows

Issue	Troubleshooting Guidelines
A device's shadow document is rejected with "Invalid JSON document."	If you are unfamiliar with JSON, modify the examples provided in this guide for your own use. For more information, see <a href="#">Device Shadow Document Syntax</a> .
I submitted correct JSON, but none or only parts of it are stored in the device's shadow document.	Be sure you are following the JSON formatting guidelines. Only JSON fields in the desired and

Issue	Troubleshooting Guidelines
	reported sections are stored. JSON content (even if formally correct) outside of those sections is ignored.
I received an error that the device's shadow exceeds the allowed size.	The device's shadow supports 8 KB of data only. Try shortening field names inside of your JSON document or simply create more shadows by creating more <i>things</i> . A device can have an unlimited number of things/shadows associated with it. The only requirement is that each thing name must be unique in your account.
When I receive a device's shadow, it is larger than 8 KB. How can this happen?	Upon receipt, the AWS IoT service adds metadata to the device's shadow. The service includes this data in its response, but it does not count toward the limit of 8 KB. Only the data for desired and reported state inside the state document sent to the device's shadow counts toward the limit.
My request has been rejected due to incorrect version. What should I do?	Perform a GET operation to sync to the latest state document version. When using MQTT, subscribe to the ./update/accepted topic to be notified about state changes and receive the latest version of the JSON document.
The timestamp is off by several seconds.	The timestamp for individual fields and the whole JSON document is updated when the document is received by the AWS IoT service or when the state document is published onto the ./update/accepted and ./update/delta message. Messages can be delayed over the network, which can cause the timestamp to be off by a few seconds.
My device can publish and subscribe on the corresponding shadow topics, but when I attempt to update the shadow document over the HTTP REST API, I get HTTP 403.	Be sure you have created policies in IAM to allow access to these topics and for the corresponding action (UPDATE/GET/DELETE) for the credentials you are using. IAM policies and certificate policies are independent.
Other issues.	The Device Shadow service logs errors to CloudWatch Logs. To identify device and configuration issues, enable CloudWatch Logs and view the logs for debug information.

## Diagnosing Salesforce IoT Input Stream Action Issues

### Execution Trace

How do I see the execution trace of a Salesforce action?

If CloudWatch Logs are not set up, see the [Monitoring with CloudWatch Logs \(p. 629\)](#) section. After you have activated the logs, you are able to see the execution trace of the Salesforce action.

## Action Success and Failure

How do I check that messages have been sent successfully to a Salesforce IoT input stream?

View the logs generated by execution of the Salesforce action in CloudWatch Logs. If you see "Action executed successfully," then it means that the AWS IoT rules engine received confirmation from the Salesforce IoT that the message was successfully pushed to the targeted input stream.

If you are experiencing problems with the Salesforce IoT platform, contact Salesforce IoT support.

What do I do if messages have not been sent successfully to a Salesforce IoT input stream?

View the logs generated by execution of the Salesforce action in CloudWatch Logs. Depending on the log entry, you can try the following actions:

**Failed to locate the host**

Check that the `url` parameter of the action is correct and that your Salesforce IoT input stream exists.

**Received Internal Server Error from Salesforce**

Retry. If the problem persists, contact Salesforce IoT Support.

**Received Bad Request Exception from Salesforce**

Check the payload you are sending for errors.

**Received Unsupported Media Type Exception from Salesforce**

Salesforce IoT does not support a binary payload at this time. Check that you are sending a JSON payload.

**Received Unauthorized Exception from Salesforce**

Check that the `token` parameter of the action is correct and that your token is still valid.

**Received Not Found Exception from Salesforce**

Check that the `url` parameter of the action is correct and that your Salesforce IoT input stream exists.

If you receive an error that is not listed here, contact AWS Support.

## AWS IoT Limits

AWS IoT limit information and values are provided in the [AWS IoT Limits](#) section of the *Amazon Web Services General Reference*.

## AWS IoT Errors

This section lists the error codes sent by AWS IoT.

### Message Broker Error Codes

Error Code	Error Description
400	Bad request.

Error Code	Error Description
401	Unauthorized.
403	Forbidden.
503	Service unavailable.

### Identity and Security Error Codes

Error Code	Error Description
401	Unauthorized.

### Device Shadow Error Codes

Error Code	Error Description
400	Bad request.
401	Unauthorized.
403	Forbidden.
404	Not found.
409	Conflict.
413	Request too large.
422	Failed to process request.
429	Too many requests.
500	Internal error.
503	Service unavailable.

# IOT Commands

This chapter contains the following sections:

- [AcceptCertificateTransfer \(p. 661\)](#)
- [AddThingToBillingGroup \(p. 662\)](#)
- [AddThingToThingGroup \(p. 663\)](#)
- [AssociateTargetsWithJob \(p. 664\)](#)
- [AttachPolicy \(p. 666\)](#)
- [AttachPrincipalPolicy \(p. 667\)](#)
- [AttachSecurityProfile \(p. 668\)](#)
- [AttachThingPrincipal \(p. 669\)](#)
- [CancelAuditTask \(p. 670\)](#)
- [CancelCertificateTransfer \(p. 671\)](#)
- [CancelJob \(p. 672\)](#)
- [CancelJobExecution \(p. 674\)](#)
- [ClearDefaultAuthorizer \(p. 676\)](#)
- [CreateAuthorizer \(p. 677\)](#)
- [CreateBillingGroup \(p. 679\)](#)
- [CreateCertificateFromCsr \(p. 680\)](#)
- [CreateDynamicThingGroup \(p. 682\)](#)
- [CreateJob \(p. 686\)](#)
- [CreateKeysAndCertificate \(p. 691\)](#)
- [CreateOTAUpdate \(p. 692\)](#)
- [CreatePolicy \(p. 698\)](#)
- [CreatePolicyVersion \(p. 699\)](#)
- [CreateRoleAlias \(p. 701\)](#)
- [CreateScheduledAudit \(p. 702\)](#)
- [CreateSecurityProfile \(p. 705\)](#)
- [CreateStream \(p. 709\)](#)
- [CreateThing \(p. 712\)](#)
- [CreateThingGroup \(p. 714\)](#)
- [CreateThingType \(p. 717\)](#)
- [CreateTopicRule \(p. 719\)](#)
- [DeleteAccountAuditConfiguration \(p. 733\)](#)
- [DeleteAuthorizer \(p. 734\)](#)
- [DeleteBillingGroup \(p. 735\)](#)
- [DeleteCACertificate \(p. 736\)](#)
- [DeleteCertificate \(p. 737\)](#)
- [DeleteDynamicThingGroup \(p. 738\)](#)
- [DeleteJob \(p. 739\)](#)
- [DeleteJobExecution \(p. 741\)](#)

- [DeleteOTAUpdate \(p. 743\)](#)
- [DeletePolicy \(p. 744\)](#)
- [DeletePolicyVersion \(p. 745\)](#)
- [DeleteRegistrationCode \(p. 746\)](#)
- [DeleteRoleAlias \(p. 747\)](#)
- [DeleteScheduledAudit \(p. 748\)](#)
- [DeleteSecurityProfile \(p. 749\)](#)
- [DeleteStream \(p. 750\)](#)
- [DeleteThing \(p. 751\)](#)
- [DeleteThingGroup \(p. 752\)](#)
- [DeleteThingShadow \(p. 753\)](#)
- [DeleteThingType \(p. 754\)](#)
- [DeleteTopicRule \(p. 755\)](#)
- [DeleteV2LoggingLevel \(p. 756\)](#)
- [DeprecateThingType \(p. 757\)](#)
- [DescribeAccountAuditConfiguration \(p. 758\)](#)
- [DescribeAuditTask \(p. 760\)](#)
- [DescribeAuthorizer \(p. 762\)](#)
- [DescribeBillingGroup \(p. 764\)](#)
- [DescribeCACertificate \(p. 766\)](#)
- [DescribeCertificate \(p. 768\)](#)
- [DescribeDefaultAuthorizer \(p. 771\)](#)
- [DescribeEndpoint \(p. 773\)](#)
- [DescribeEventConfigurations \(p. 774\)](#)
- [DescribeIndex \(p. 775\)](#)
- [DescribeJob \(p. 777\)](#)
- [DescribeJobExecution \(p. 782\)](#)
- [DescribeJobExecution \(p. 785\)](#)
- [DescribeRoleAlias \(p. 788\)](#)
- [DescribeScheduledAudit \(p. 790\)](#)
- [DescribeSecurityProfile \(p. 791\)](#)
- [DescribeStream \(p. 796\)](#)
- [DescribeThing \(p. 798\)](#)
- [DescribeThingGroup \(p. 800\)](#)
- [DescribeThingRegistrationTask \(p. 803\)](#)
- [DescribeThingType \(p. 805\)](#)
- [DetachPolicy \(p. 807\)](#)
- [DetachPrincipalPolicy \(p. 808\)](#)
- [DetachSecurityProfile \(p. 809\)](#)
- [DetachThingPrincipal \(p. 810\)](#)
- [DisableTopicRule \(p. 811\)](#)
- [EnableTopicRule \(p. 812\)](#)
- [GetEffectivePolicies \(p. 813\)](#)
- [GetIndexingConfiguration \(p. 814\)](#)

- [GetJobDocument \(p. 816\)](#)
- [GetLoggingOptions \(p. 817\)](#)
- [GetOTAUpdate \(p. 818\)](#)
- [GetPendingJobExecutions \(p. 823\)](#)
- [GetPolicy \(p. 825\)](#)
- [GetPolicyVersion \(p. 827\)](#)
- [GetRegistrationCode \(p. 829\)](#)
- [GetStatistics \(p. 829\)](#)
- [GetThingShadow \(p. 831\)](#)
- [GetTopicRule \(p. 832\)](#)
- [GetV2LoggingOptions \(p. 847\)](#)
- [ListActiveViolations \(p. 848\)](#)
- [ListAttachedPolicies \(p. 853\)](#)
- [ListAuditFindings \(p. 854\)](#)
- [ListAuditTasks \(p. 860\)](#)
- [ListAuthorizers \(p. 862\)](#)
- [ListBillingGroups \(p. 864\)](#)
- [ListCACertificates \(p. 865\)](#)
- [ListCertificates \(p. 867\)](#)
- [ListCertificatesByCA \(p. 869\)](#)
- [ListIndices \(p. 871\)](#)
- [ListJobExecutionsForJob \(p. 872\)](#)
- [ListJobExecutionsForThing \(p. 875\)](#)
- [ListJobs \(p. 877\)](#)
- [ListOTAUpdates \(p. 880\)](#)
- [ListOutgoingCertificates \(p. 882\)](#)
- [ListPolicies \(p. 883\)](#)
- [ListPolicyPrincipals \(p. 885\)](#)
- [ListPolicyVersions \(p. 887\)](#)
- [ListPrincipalPolicies \(p. 888\)](#)
- [ListPrincipalThings \(p. 890\)](#)
- [ListRoleAliases \(p. 891\)](#)
- [ListScheduledAudits \(p. 893\)](#)
- [ListSecurityProfiles \(p. 894\)](#)
- [ListSecurityProfilesForTarget \(p. 896\)](#)
- [ListStreams \(p. 898\)](#)
- [ListTagsForResource \(p. 899\)](#)
- [ListTargetsForPolicy \(p. 900\)](#)
- [ListTargetsForSecurityProfile \(p. 902\)](#)
- [ListThingGroups \(p. 903\)](#)
- [ListThingGroupsForThing \(p. 905\)](#)
- [ListThingPrincipals \(p. 907\)](#)
- [ListThingRegistrationTaskReports \(p. 908\)](#)
- [ListThingRegistrationTasks \(p. 909\)](#)
- [ListThingTypes \(p. 911\)](#)

- [ListThings \(p. 913\)](#)
- [ListThingsInBillingGroup \(p. 915\)](#)
- [ListThingsInThingGroup \(p. 916\)](#)
- [ListTopicRules \(p. 918\)](#)
- [ListV2LoggingLevels \(p. 919\)](#)
- [ListViolationEvents \(p. 921\)](#)
- [Publish \(p. 926\)](#)
- [RegisterCACertificate \(p. 927\)](#)
- [RegisterCertificate \(p. 929\)](#)
- [RegisterThing \(p. 931\)](#)
- [RejectCertificateTransfer \(p. 932\)](#)
- [RemoveThingFromBillingGroup \(p. 933\)](#)
- [RemoveThingFromThingGroup \(p. 934\)](#)
- [ReplaceTopicRule \(p. 935\)](#)
- [SearchIndex \(p. 950\)](#)
- [SetDefaultAuthorizer \(p. 953\)](#)
- [SetDefaultPolicyVersion \(p. 955\)](#)
- [SetLoggingOptions \(p. 956\)](#)
- [SetV2LogLevel \(p. 957\)](#)
- [SetV2LoggingOptions \(p. 958\)](#)
- [StartNextPendingJobExecution \(p. 958\)](#)
- [StartOnDemandAuditTask \(p. 961\)](#)
- [StartThingRegistrationTask \(p. 963\)](#)
- [StopThingRegistrationTask \(p. 964\)](#)
- [TagResource \(p. 965\)](#)
- [TestAuthorization \(p. 966\)](#)
- [TestInvokeAuthorizer \(p. 970\)](#)
- [TransferCertificate \(p. 972\)](#)
- [UntagResource \(p. 973\)](#)
- [UpdateAccountAuditConfiguration \(p. 974\)](#)
- [UpdateAuthorizer \(p. 976\)](#)
- [UpdateBillingGroup \(p. 978\)](#)
- [UpdateCACertificate \(p. 979\)](#)
- [UpdateCertificate \(p. 981\)](#)
- [UpdateDynamicThingGroup \(p. 982\)](#)
- [UpdateEventConfigurations \(p. 985\)](#)
- [UpdateIndexingConfiguration \(p. 986\)](#)
- [UpdateJob \(p. 987\)](#)
- [UpdateJobExecution \(p. 990\)](#)
- [UpdateRoleAlias \(p. 994\)](#)
- [UpdateScheduledAudit \(p. 995\)](#)
- [UpdateSecurityProfile \(p. 997\)](#)
- [UpdateStream \(p. 1005\)](#)
- [UpdateThing \(p. 1007\)](#)

- [UpdateThingGroup \(p. 1009\)](#)
- [UpdateThingGroupsForThing \(p. 1011\)](#)
- [UpdateThingShadow \(p. 1012\)](#)
- [ValidateSecurityProfileBehaviors \(p. 1014\)](#)

## AcceptCertificateTransfer

Accepts a pending certificate transfer. The default state of the certificate is INACTIVE.

To check for pending certificate transfers, call `ListCertificates` to enumerate your certificates.

### Synopsis

```
aws iot accept-certificate-transfer \
  --certificate-id <value> \
  [--set-as-active | --no-set-as-active] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateId": "string",
  "setAsActive": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)
setAsActive	boolean	Specifies whether the certificate is active.

### Output

None

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`TransferAlreadyCompletedException`

You can't revert the certificate transfer because the transfer is already complete.

`InvalidRequestException`

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## AddThingToBillingGroup

Adds a thing to a billing group.

### Synopsis

```
aws iot add-thing-to-billing-group \
[--billing-group-name <value>] \
[--billing-group-arn <value>] \
[--thing-name <value>] \
[--thing-arn <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "billingGroupName": "string",
  "billingGroupArn": "string",
  "thingName": "string",
  "thingArn": "string"
}
```

### cli-input-json fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.
billingGroupArn	string	The ARN of the billing group.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing to be added to the billing group.
thingArn	string	The ARN of the thing to be added to the billing group.

## Output

None

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

# AddThingToThingGroup

Adds a thing to a thing group.

## Synopsis

```
aws iot add-thing-to-thing-group \
[--thing-group-name <value>] \
[--thing-group-arn <value>] \
[--thing-name <value>] \
[--thing-arn <value>] \
[--override-dynamic-groups | --no-override-dynamic-groups] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

## cli-input-json format

```
{
  "thingGroupName": "string",
  "thingGroupArn": "string",
  "thingName": "string",
  "thingArn": "string",
  "overrideDynamicGroups": "boolean"
}
```

## cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the group to which you are adding a thing.
thingGroupArn	string	The ARN of the group to which you are adding a thing.

Name	Type	Description
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing to add to a group.
thingArn	string	The ARN of the thing to add to a group.
overrideDynamicGroups	boolean	Override dynamic thing groups with static thing groups when 10-group limit is reached. If a thing belongs to 10 thing groups, and one or more of those groups are dynamic thing groups, adding a thing to a static group removes the thing from the last dynamic group.

#### Output

None

#### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## AssociateTargetsWithJob

Associates a group with a continuous job. The following criteria must be met:

- The job must have been created with the targetSelection field set to "CONTINUOUS".
- The job status must currently be "IN\_PROGRESS".
- The total number of targets associated with a job must not exceed 100.

#### Synopsis

```
aws iot associate-targets-with-job \
--targets <value> \
--job-id <value> \
[--comment <value>] \
```

```
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "targets": [
    "string"
  ],
  "jobId": "string",
  "comment": "string"
}
```

#### cli-input-json fields

Name	Type	Description
targets	list  member: TargetArn	A list of thing group ARNs that define the targets of the job.
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
comment	string  length- max:2028  pattern: [^\p{C}]+	An optional comment string describing why the job was associated with the targets.

#### Output

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

#### CLI output fields

Name	Type	Description
jobArn	string	An ARN identifying the job.
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
description	string  length- max:2028  pattern: [^\p{C}]+	A short text description of the job.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`LimitExceededException`

A limit has been exceeded.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

# AttachPolicy

Attaches a policy to the specified target.

## Synopsis

```
aws iot attach-policy \
--policy-name <value> \
--target <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "target": "string"
}
```

## cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy to attach.
target	string	The <a href="#">identity</a> to which the policy is attached.

## Output

None

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

# AttachPrincipalPolicy

Attaches the specified policy to the specified principal (certificate or other credential).

**Note:** This API is deprecated. Please use `AttachPolicy` instead.

## Synopsis

```
aws iot attach-principal-policy \
--policy-name <value> \
--principal <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "principal": "string"
}
```

## cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.

Name	Type	Description
principal	string	The principal, which can be a certificate ARN (as returned from the CreateCertificate operation) or an Amazon Cognito ID.

#### Output

None

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## AttachSecurityProfile

Associates a Device Defender security profile with a thing group or with this account. Each thing group or account can have up to five security profiles associated with it.

#### Synopsis

```
aws iot attach-security-profile \
--security-profile-name <value> \
--security-profile-target-arn <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "securityProfileName": "string",
  "securityProfileTargetArn": "string"
```

}

### cli-input-json fields

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile that is attached.
securityProfileTargetArn	string	The ARN of the target (thing group) to which the security profile is attached.

### Output

None

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`LimitExceededException`

A limit has been exceeded.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the `--version` parameter.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## AttachThingPrincipal

Attaches the specified principal to the specified thing. A principal can be X.509 certificates, IAM users, groups, and roles, Amazon Cognito identities or federated identities.

### Synopsis

```
aws iot attach-thing-principal \
--thing-name <value> \
--principal <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string",
  "principal": "string"
}
```

#### **cli-input-json** fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.
principal	string	The principal, such as a certificate or other credential.

#### Output

None

#### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## CancelAuditTask

Cancels an audit that is in progress. The audit can be either scheduled or on-demand. If the audit is not in progress, an "InvalidRequestException" occurs.

#### Synopsis

```
aws iot cancel-audit-task \
--task-id <value> \
```

```
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "taskId": "string"
}
```

#### cli-input-json fields

Name	Type	Description
taskId	string length- max:40 min:1 pattern: [a-zA-Z0-9-]+	The ID of the audit you want to cancel. You can only cancel an audit that is "IN_PROGRESS".

#### Output

None

#### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## CancelCertificateTransfer

Cancels a pending transfer for the specified certificate.

**Note** Only the transfer source account can use this operation to cancel a transfer. (Transfer destinations can use RejectCertificateTransfer instead.) After transfer, AWS IoT returns the certificate to the source account in the INACTIVE state. After the destination account has accepted the transfer, the transfer cannot be cancelled.

After a certificate transfer is cancelled, the status of the certificate changes from PENDING\_TRANSFER to INACTIVE.

#### Synopsis

```
aws iot cancel-certificate-transfer \
--certificate-id <value> \
[--cli-input-json <value>]
```

```
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "certificateId": "string"  
}
```

#### cli-input-json fields

Name	Type	Description
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)

#### Output

None

#### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**TransferAlreadyCompletedException**

You can't revert the certificate transfer because the transfer is already complete.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## CancelJob

Cancels a job.

#### Synopsis

```
aws iot cancel-job \  
    --job-id <value> \  
    ...
```

```
[--reason-code <value>] \
[--comment <value>] \
[--force | --no-force] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string",
  "reasonCode": "string",
  "comment": "string",
  "force": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
reasonCode	string length- max:128 pattern: [\p{Upper}p Digit_]+	(Optional)A reason code string that explains why the job was canceled.
comment	string length- max:2028 pattern: [^\p{C}]+	An optional comment string describing why the job was canceled.
force	boolean	(Optional) If true job executions with status "IN_PROGRESS" and "QUEUED" are canceled, otherwise only job executions with status "QUEUED" are canceled. The default is false.  Canceling a job which is "IN_PROGRESS", will cause a device which is executing the job to be unable to update the job execution status. Use caution and ensure that each device executing a job which is canceled is able to recover to a valid state.

### Output

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
```

}

## CLI output fields

Name	Type	Description
jobArn	string	The job ARN.
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
description	string length- max:2028 pattern: [^\p{C}]+	A short text description of the job.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### ServiceUnavailableException

The service is temporarily unavailable.

# CancelJobExecution

Cancels the execution of a job for a given thing.

## Synopsis

```
aws iot cancel-job-execution \
--job-id <value> \
--thing-name <value> \
[--force | --no-force] \
[--expected-version <value>] \
[--status-details <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

## cli-input-json format

```
{
  "jobId": "string",
  "thingName": "string",
  "force": "boolean",
  "expectedVersion": "long",
```

```

    "statusDetails": {
        "string": "string"
    }
}

```

### cli-input-json fields

Name	Type	Description
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the job to be canceled.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing whose execution of the job will be canceled.
force	boolean	(Optional) If true the job execution will be canceled if it has status IN_PROGRESS or QUEUED, otherwise the job execution will be canceled only if it has status QUEUED. If you attempt to cancel a job execution that is IN_PROGRESS, and you do not set force to true, then an InvalidStateTransitionException will be thrown. The default is false.  Canceling a job execution which is "IN_PROGRESS", will cause the device to be unable to update the job execution status. Use caution and ensure that the device is able to recover to a valid state.
expectedVersion	long	(Optional) The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in Jobs does not match, the update is rejected with a VersionMismatch error, and an ErrorResponse that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate DescribeJobExecution request in order to obtain the job execution status data.)

Name	Type	Description
statusDetails	map	A collection of name/value pairs that describe the status of the job execution. If not specified, the statusDetails are unchanged. You can specify at most 10 name/value pairs.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`InvalidStateException`

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state `SUCCESS` to state `IN_PROGRESS`). In this case, the body of the error message also contains the `executionState` field.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the `--version` parameter.

## ClearDefaultAuthorizer

Clears the default authorizer.

#### Synopsis

```
aws iot clear-default-authorizer \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

#### Output

None

## Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

# CreateAuthorizer

Creates an authorizer.

## Synopsis

```
aws iot create-authorizer \
--authorizer-name <value> \
--authorizer-function-arn <value> \
--token-key-name <value> \
--token-signing-public-keys <value> \
[--status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "authorizerName": "string",
  "authorizerFunctionArn": "string",
  "tokenKeyName": "string",
  "tokenSigningPublicKeys": {
    "string": "string"
  },
  "status": "string"
}
```

## cli-input-json fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.

Name	Type	Description
authorizerFunctionArn	string	The ARN of the authorizer's Lambda function.
tokenKeyName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the token key used to extract the token from the HTTP headers.
tokenSigningPublicKeys	map	The public keys used to verify the digital signature returned by your custom authentication service.
status	string	The status of the create authorizer request.  enum: ACTIVE   INACTIVE

## Output

```
{
  "authorizerName": "string",
  "authorizerArn": "string"
}
```

## CLI output fields

Name	Type	Description
authorizerName	string length- max:128 min:1 pattern: [w=,@-]+	The authorizer's name.
authorizerArn	string	The authorizer ARN.

## Errors

`ResourceAlreadyExistsException`

The resource already exists.

`InvalidRequestException`

The contents of the request were invalid.

`LimitExceededException`

A limit has been exceeded.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

```
ServiceUnavailableException
The service is temporarily unavailable.

InternalFailureException
An unexpected error has occurred.
```

## CreateBillingGroup

Creates a billing group.

### Synopsis

```
aws iot create-billing-group \
--billing-group-name <value> \
[--billing-group-properties <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "billingGroupName": "string",
  "billingGroupProperties": {
    "billingGroupDescription": "string"
  },
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### cli-input-json fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you wish to give to the billing group.
billingGroupProperties	BillingGroupProperties	The properties of the billing group.
billingGroupDescription	string  length- max:2028  pattern: [\p{Graph}]*	The description of the billing group.
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage the billing group.

Name	Type	Description
Key	string	The tag's key.
Value	string	The tag's value.

#### Output

```
{
  "billingGroupName": "string",
  "billingGroupArn": "string",
  "billingGroupId": "string"
}
```

#### CLI output fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you gave to the billing group.
billingGroupArn	string	The ARN of the billing group.
billingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-]+	The ID of the billing group.

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceAlreadyExistsException`

The resource already exists.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## CreateCertificateFromCsr

Creates an X.509 certificate using the specified certificate signing request.

**Note:** The CSR must include a public key that is either an RSA key with a length of at least 2048 bits or an ECC key from NIST P-256 or NIST P-384 curves.

**Note:** Reusing the same certificate signing request (CSR) results in a distinct certificate.

You can create multiple certificates in a batch by creating a directory, copying multiple .csr files into that directory, and then specifying that directory on the command line. The following commands show how to create a batch of certificates given a batch of CSRs.

Assuming a set of CSRs are located inside of the directory my-csr-directory:

On Linux and OS X, the command is:

```
$ ls my-csr-directory/ | xargs -l aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/
```

This command lists all of the CSRs in my-csr-directory and pipes each CSR file name to the aws iot create-certificate-from-csr AWS CLI command to create a certificate for the corresponding CSR.

The aws iot create-certificate-from-csr part of the command can also be run in parallel to speed up the certificate creation process:

```
$ ls my-csr-directory/ | xargs -P 10 -l aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/
```

On Windows PowerShell, the command to create certificates for all CSRs in my-csr-directory is:

```
> ls -Name my-csr-directory | % aws iot create-certificate-from-csr --certificate-signing-request file://my-csr-directory/$_
```

On a Windows command prompt, the command to create certificates for all CSRs in my-csr-directory is:

```
> forfiles /p my-csr-directory /c "cmd /c aws iot create-certificate-from-csr --certificate-signing-request file://@path"
```

### Synopsis

```
aws iot create-certificate-from-csr \
--certificate-signing-request <value> \
[--set-as-active | --no-set-as-active] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateSigningRequest": "string",
  "setAsActive": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
certificateSigningRequest	string length- min:1	The certificate signing request (CSR).
setAsActive	boolean	Specifies whether the certificate is active.

## Output

```
{  
    "certificateArn": "string",  
    "certificateId": "string",  
    "certificatePem": "string"  
}
```

## CLI output fields

Name	Type	Description
certificateArn	string	The Amazon Resource Name (ARN) of the certificate. You can use the ARN as a principal for policy operations.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. Certificate management operations only take a certificateId.
certificatePem	string length- max:65536 min:1	The certificate data, in PEM format.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### InternalFailureException

An unexpected error has occurred.

# CreateDynamicThingGroup

Creates a dynamic thing group.

## Synopsis

```
aws iot create-dynamic-thing-group \  
    --thing-group-name <value> \  

```

```
[--thing-group-properties <value>] \
[--index-name <value>] \
--query-string <value> \
[--query-version <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "thingGroupName": "string",
  "thingGroupProperties": {
    "thingGroupDescription": "string",
    "attributePayload": {
      "attributes": {
        "string": "string"
      },
      "merge": "boolean"
    }
  },
  "indexName": "string",
  "queryString": "string",
  "queryVersion": "string",
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

#### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group name to create.
thingGroupProperties	ThingGroupProperties	The dynamic thing group properties.
thingGroupDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The thing group description.
attributePayload	AttributePayload	The thing group attributes in JSON format.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\"attributes\":\n{\\"string1\\":\n\"string2\\\"}</code>

Name	Type	Description
merge	boolean	<p>Specifies whether the list of attributes provided in the <code>AttributePayload</code> is merged with the attributes stored in the registry, instead of overwriting them.</p> <p>To remove an attribute, call <code>UpdateThing</code> with an empty attribute value.</p> <p><b>Note</b> The <code>merge</code> attribute is only valid when calling <code>UpdateThing</code>.</p>
indexName	<p>string</p> <p>length- max:128 min:1</p> <p>pattern: [a-zA-Z0-9:_]+</p>	<p>The dynamic thing group index name.</p> <p><b>Note</b> Currently one index is supported: "AWS_Things".</p>
queryString	<p>string</p> <p>length- min:1</p>	<p>The dynamic thing group search query string.</p> <p>See <a href="#">Query Syntax</a> for information about query string syntax.</p>
queryVersion	string	<p>The dynamic thing group query version.</p> <p><b>Note</b> Currently one query version is supported: "2017-09-30". If not specified, the query version defaults to this value.</p>
tags	<p>list</p> <p>member: Tag</p> <p>java class: java.util.List</p>	Metadata which can be used to manage the dynamic thing group.
Key	string	The tag's key.
Value	string	The tag's value.

## Output

```
{
  "thingGroupName": "string",
  "thingGroupArn": "string",
  "thingGroupId": "string",
```

```

    "indexName": "string",
    "queryString": "string",
    "queryVersion": "string"
}

```

### CLI output fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group name.
thingGroupArn	string	The dynamic thing group ARN.
thingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group ID.
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group index name.
queryString	string  length- min:1	The dynamic thing group search query string.
queryVersion	string	The dynamic thing group query version.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceAlreadyExistsException`

The resource already exists.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`InvalidQueryException`

The query is invalid.

`LimitExceededException`

A limit has been exceeded.

# CreateJob

Creates a job.

## Synopsis

```
aws iot create-job \
--job-id <value> \
--targets <value> \
[--document-source <value>] \
[--document <value>] \
[--description <value>] \
[--presigned-url-config <value>] \
[--target-selection <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{
  "jobId": "string",
  "targets": [
    "string"
  ],
  "documentSource": "string",
  "document": "string",
  "description": "string",
  "presignedUrlConfig": {
    "roleArn": "string",
    "expiresInSec": "long"
  },
  "targetSelection": "string",
  "jobExecutionsRolloutConfig": {
    "maximumPerMinute": "integer",
    "exponentialRate": {
      "baseRatePerMinute": "integer",
      "incrementFactor": "double",
      "rateIncreaseCriteria": {
        "numberOfNotifiedThings": "integer",
        "numberOfSucceededThings": "integer"
      }
    }
  },
  "abortConfig": {
    "criteriaList": [
      {
        "failureType": "string",
        "action": "string",
        "thresholdPercentage": "double",
        "minNumberOfExecutedThings": "integer"
      }
    ]
  },
  "timeoutConfig": {
    "inProgressTimeoutInMinutes": "long"
  },
  "tags": [
    {
      "key": "string",
      "value": "string"
    }
  ]
}
```

```

        "Key": "string",
        "Value": "string"
    }
]
}

```

### cli-input-json fields

Name	Type	Description
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	A job identifier which must be unique for your AWS account. We recommend using a UUID. Alpha-numeric characters, "-" and "_" are valid for use here.
targets	list member: TargetArn	A list of things and thing groups to which the job should be sent.
documentSource	string length- max:1350 min:1	An S3 link to the job document.
document	string length- max:32768	The job document. <b>Note</b> If the job document resides in an S3 bucket, you must use a placeholder link when specifying the document. The placeholder link is of the following form: <code>\$ aws:iot:s3-presigned-url:https://s3.amazonaws.com/bucket/key</code> where <i>bucket</i> is your bucket name and <i>key</i> is the object in the bucket to which you are linking.
description	string length- max:2028 pattern: [^\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration information for pre-signed S3 URLs.
roleArn	string length- max:2048 min:20	The ARN of an IAM role that grants permission to download files from the S3 bucket where the job data/updates are stored. The role

Name	Type	Description
		must also grant permission for IoT to download the files.
expiresInSec	long range- max:3600 min:60	How long (in seconds) pre-signed URLs are valid. Valid values are 60 - 3600, the default value is 3600 seconds. Pre-signed URLs are generated when Jobs receives an MQTT request for the job document.
targetSelection	string	Specifies whether the job will continue to run (CONTINUOUS), or will be complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous, the job may also be run on a thing when a change is detected in a target. For example, a job will run on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.  enum: CONTINUOUS   SNAPSHOT
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of the job.
maximumPerMinute	integer range- min:1	The maximum number of things that will be notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	integer range- max:1000 min:1	The minimum number of things that will be notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.
rateIncreaseCriteria	RateIncreaseCriteria	The criteria to initiate the increase in rate of rollout for a job.  AWS IoT supports up to one digit after the decimal (for example, 1.5, but not 1.55).

Name	Type	Description
numberOfNotifiedThings	integer range- min:1	The threshold for number of notified things that will initiate the increase in rate of rollout.
numberOfSucceededThings	integer range- min:1	The threshold for number of succeeded things that will initiate the increase in rate of rollout.
abortConfig	AbortConfig	Allows you to create criteria to abort a job.
criteriaList	list member: AbortCriteria java class: java.util.List	The list of abort criteria to define rules to abort the job.
failureType	string	The type of job execution failure to define a rule to initiate a job abort.  enum: FAILED   REJECTED   TIMED_OUT   ALL
action	string	The type of abort action to initiate a job abort.  enum: CANCEL
minNumberOfExecutedThings	integer range- min:1	Minimum number of executed things before evaluating an abort rule.
timeoutConfig	TimeoutConfig	Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to IN_PROGRESS. If the job execution status is not set to another terminal state before the time expires, it will be automatically set to TIMED_OUT.

Name	Type	Description
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The in progress timer can't be updated and will apply to all job executions for the job. Whenever a job execution remains in the IN_PROGRESS status for longer than this interval, the job execution will fail and switch to the terminal TIMED_OUT status.
tags	list member: Tag java class: java.util.List	Metadata which can be used to manage the job.
Key	string	The tag's key.
Value	string	The tag's value.

## Output

```
{
  "jobArn": "string",
  "jobId": "string",
  "description": "string"
}
```

## CLI output fields

Name	Type	Description
jobArn	string	The job ARN.
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job.
description	string length- max:2028 pattern: [^\p{C}]+	The job description.

## Errors

### InvalidRequestException

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ResourceAlreadyExistsException`

The resource already exists.

`LimitExceededException`

A limit has been exceeded.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

## CreateKeysAndCertificate

Creates a 2048-bit RSA key pair and issues an X.509 certificate using the issued public key.

**Note** This is the only time AWS IoT issues the private key for this certificate, so it is important to keep it in a secure location.

### Synopsis

```
aws iot create-keys-and-certificate \
[--set-as-active | --no-set-as-active] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "setAsActive": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
setAsActive	boolean	Specifies whether the certificate is active.

### Output

```
{
  "certificateArn": "string",
  "certificateId": "string",
  "certificatePem": "string",
  "keyPair": {
    "PublicKey": "string",
    "PrivateKey": "string"
  }
}
```

### CLI output fields

Name	Type	Description
certificateArn	string	The ARN of the certificate.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. AWS IoT issues a default subject name for the certificate (for example, AWS IoT Certificate).
certificatePem	string length- max:65536 min:1	The certificate data, in PEM format.
keyPair	KeyPair	The generated key pair.
PublicKey	string length- min:1	The public key.
PrivateKey	string length- min:1	The private key.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## CreateOTAUpdate

Creates an AWS IoT OTAUpdate on a target group of things or groups.

### Synopsis

```
aws iot create-ota-update \
--ota-update-id <value> \
[--description <value>] \
--targets <value> \
[--target-selection <value>] \
[--aws-job-executions-rollout-config <value>] \
--files <value> \
--role-arn <value> \
[--additional-parameters <value>] \
```

```
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
    "otaUpdateId": "string",
    "description": "string",
    "targets": [
        "string"
    ],
    "targetSelection": "string",
    "awsJobExecutionsRolloutConfig": {
        "maximumPerMinute": "integer"
    },
    "files": [
        {
            "fileName": "string",
            "fileVersion": "string",
            "fileLocation": {
                "stream": {
                    "streamId": "string",
                    "fileId": "integer"
                },
                "s3Location": {
                    "bucket": "string",
                    "key": "string",
                    "version": "string"
                }
            },
            "codeSigning": {
                "awsSignerJobId": "string",
                "startSigningJobParameter": {
                    "signingProfileParameter": {
                        "certificateArn": "string",
                        "platform": "string",
                        "certificatePathOnDevice": "string"
                    },
                    "signingProfileName": "string",
                    "destination": {
                        "s3Destination": {
                            "bucket": "string",
                            "prefix": "string"
                        }
                    }
                },
                "customCodeSigning": {
                    "signature": {
                        "inlineDocument": "blob"
                    },
                    "certificateChain": {
                        "certificateName": "string",
                        "inlineDocument": "string"
                    },
                    "hashAlgorithm": "string",
                    "signatureAlgorithm": "string"
                },
                "attributes": {
                    "string": "string"
                }
            }
        ],
        "roleArn": "string",
    ]
}
```

```

    "additionalParameters": {
        "string": "string"
    },
    "tags": [
        {
            "Key": "string",
            "Value": "string"
        }
    ]
}

```

#### **cli-input-json fields**

Name	Type	Description
otaUpdateId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the OTA update to be created.
description	string  length- max:2028  pattern: [^\p{C}]+	The description of the OTA update.
targets	list  member: Target	The targeted devices to receive OTA updates.
targetSelection	string  Specifies whether the update will continue to run (CONTINUOUS), or will be complete after all the things specified as targets have completed the update (SNAPSHOT). If continuous, the update may also be run on a thing when a change is detected in a target. For example, an update will run on a thing when the thing is added to a target group, even after the update was completed by all things originally in the group. Valid values: CONTINUOUS   SNAPSHOT.  enum: CONTINUOUS   SNAPSHOT	
awsJobExecutionsRolloutConfig	AwsJobExecutionsRolloutConfig	Configuration for the rollout of OTA updates.
maximumPerMinute	integer  range- max:1000 min:1	The maximum number of OTA update job executions started per minute.
files	list	The files to be streamed by the OTA update.

Name	Type	Description
	member: OTAUpdateFile	
fileName	string	The name of the file.
fileVersion	string	The file version.
fileLocation	FileLocation	The location of the updated firmware.
stream	Stream	The stream that contains the OTA update.
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
fileId	integer  range- max:255 min:0	The ID of a file associated with a stream.
s3Location	S3Location	The location of the updated firmware in S3.
bucket	string  length- min:1	The S3 bucket.
key	string  length- min:1	The S3 key.
version	string	The S3 bucket version.
codeSigning	CodeSigning	The code signing method of the file.
awsSignerJobId	string	The ID of the AWSSignerJob which was created to sign the file.
startSigningJobParameter	StartSigningJobParameter	Describes the code-signing job.
signingProfileParameter	SigningProfileParameter	Describes the code-signing profile.
certificateArn	string	Certificate ARN.
platform	string	The hardware platform of your device.
certificatePathOnDevice	string	The location of the code-signing certificate on your device.
signingProfileName	string	The code-signing profile name.
destination	Destination	The location to write the code-signed file.

Name	Type	Description
s3Destination	S3Destination	Describes the location in S3 of the updated firmware.
bucket	string length- min:1	The S3 bucket that contains the updated firmware.
prefix	string	The S3 prefix.
customCodeSigning	CustomCodeSigning	A custom method for code signing a file.
signature	CodeSigningSignature	The signature for the file.
inlineDocument	blob	A base64 encoded binary representation of the code signing signature.
certificateChain	CodeSigningCertificateChain	The certificate chain.
certificateName	string	The name of the certificate.
inlineDocument	string	A base64 encoded binary representation of the code signing certificate chain.
hashAlgorithm	string	The hash algorithm used to code sign the file.
signatureAlgorithm	string	The signature algorithm used to code sign the file.
attributes	map	A list of name/attribute pairs.
roleArn	string length- max:2048 min:20	The IAM role that allows access to the AWS IoT Jobs service.
additionalParameters	map	A list of additional OTA update parameters which are name-value pairs.
tags	list member: Tag java class: java.util.List	Metadata which can be used to manage updates.
Key	string	The tag's key.
Value	string	The tag's value.

## Output

```
{
  "otaUpdateId": "string",
  "awsIotJobId": "string",
  "otaUpdateArn": "string",
  "awsIotJobArn": "string",
```

```
    "otaUpdateStatus": "string"
}
```

### CLI output fields

Name	Type	Description
otaUpdateId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The OTA update ID.
awsIoTJobId	string	The AWS IoT job ID associated with the OTA update.
otaUpdateArn	string	The OTA update ARN.
awsIoTJobArn	string	The AWS IoT job ARN associated with the OTA update.
otaUpdateStatus	string	The OTA update status.  enum: CREATE_PENDING   CREATE_IN_PROGRESS   CREATE_COMPLETE   CREATE_FAILED

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`LimitExceededException`

A limit has been exceeded.

`ResourceNotFoundException`

The specified resource does not exist.

`ResourceAlreadyExistsException`

The resource already exists.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`InternalFailureException`

An unexpected error has occurred.

`ServiceUnavailableException`

The service is temporarily unavailable.

# CreatePolicy

Creates an AWS IoT policy.

The created policy is the default version for the policy. This operation creates a policy version with a version identifier of **1** and sets **1** as the policy's default version.

## Synopsis

```
aws iot create-policy \
  --policy-name <value> \
  --policy-document <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "policyDocument": "string"
}
```

## cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyDocument	string	The JSON document that describes the policy. <b>policyDocument</b> must have a minimum length of 1, with a maximum length of 2048, excluding whitespace.

## Output

```
{
  "policyName": "string",
  "policyArn": "string",
  "policyDocument": "string",
  "policyVersionId": "string"
}
```

## CLI output fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.

Name	Type	Description
policyArn	string	The policy ARN.
policyDocument	string	The JSON document that describes the policy.
policyVersionId	string pattern: [0-9]+	The policy version ID.

### Errors

`ResourceAlreadyExistsException`

The resource already exists.

`MalformedPolicyException`

The policy documentation is not valid.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## CreatePolicyVersion

Creates a new version of the specified AWS IoT policy. To update a policy, create a new policy version. A managed policy can have up to five versions. If the policy has five versions, you must use `DeletePolicyVersion` to delete an existing version before you create a new one.

Optionally, you can set the new version as the policy's default version. The default version is the operative version (that is, the version that is in effect for the certificates to which the policy is attached).

### Synopsis

```
aws iot create-policy-version \
--policy-name <value> \
--policy-document <value> \
[--set-as-default | --no-set-as-default] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
  "policyName": "string",
```

```

    "policyDocument": "string",
    "setAsDefault": "boolean"
}

```

#### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyDocument	string	The JSON document that describes the policy. Minimum length of 1. Maximum length of 2048, excluding whitespace.
setAsDefault	boolean	Specifies whether the policy version is set as the default. When this parameter is true, the new policy version becomes the operative version (that is, the version that is in effect for the certificates to which the policy is attached).

#### Output

```
{
    "policyArn": "string",
    "policyDocument": "string",
    "policyVersionId": "string",
    "isDefaultVersion": "boolean"
}
```

#### CLI output fields

Name	Type	Description
policyArn	string	The policy ARN.
policyDocument	string	The JSON document that describes the policy.
policyVersionId	string  pattern: [0-9]+	The policy version ID.
isDefaultVersion	boolean	Specifies whether the policy version is the default.

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

**MalformedPolicyException**

The policy documentation is not valid.

**VersionsLimitExceededException**

The number of policy versions exceeds the limit.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## CreateRoleAlias

Creates a role alias.

### Synopsis

```
aws iot create-role-alias \
--role-alias <value> \
--role-arn <value> \
[--credential-duration-seconds <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{  
    "roleAlias": "string",  
    "roleArn": "string",  
    "credentialDurationSeconds": "integer"  
}
```

### cli-input-json fields

Name	Type	Description
roleAlias	string  length- max:128 min:1  pattern: [w=,@-]+	The role alias that points to a role ARN. This allows you to change the role without having to update the device.
roleArn	string  length- max:2048 min:20	The role ARN.

Name	Type	Description
credentialDurationSeconds	integer range- max:3600 min:900	How long (in seconds) the credentials will be valid.

#### Output

```
{
  "roleAlias": "string",
  "roleAliasArn": "string"
}
```

#### CLI output fields

Name	Type	Description
roleAlias	string length- max:128 min:1 pattern: [w=,@-]+	The role alias.
roleAliasArn	string	The role alias ARN.

#### Errors

`ResourceAlreadyExistsException`

The resource already exists.

`InvalidRequestException`

The contents of the request were invalid.

`LimitExceedededException`

A limit has been exceeded.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## CreateScheduledAudit

Creates a scheduled audit that is run at a specified time interval.

#### Synopsis

```
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]
```

#### cli-input-json format

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "scheduledAuditName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. Can be one of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] [12][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Can be "1" through "31" or "LAST". This field is required if the "frequency" parameter is set to "MONTHLY". If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. Can be one of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT". This field is required if the "frequency" parameter is set to "WEEKLY" or "BIWEEKLY".

Name	Type	Description
		enum: SUN   MON   TUE   WED   THU   FRI   SAT
targetCheckNames	list  member: AuditCheckName	Which checks are performed during the scheduled audit. Checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage the scheduled audit.
Key	string	The tag's key.
Value	string	The tag's value.
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name you want to give to the scheduled audit. (Max. 128 chars)

## Output

```
{
  "scheduledAuditArn": "string"
}
```

## CLI output fields

Name	Type	Description
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**LimitExceeded**

A limit has been exceeded.

## CreateSecurityProfile

Creates a Device Defender security profile.

### Synopsis

```
aws iot create-security-profile \
  --security-profile-name <value> \
  [--security-profile-description <value>] \
  [--behaviors <value>] \
  [--alert-targets <value>] \
  [--additional-metrics-to-retain <value>] \
  [--tags <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "securityProfileName": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ],
  "alertTargets": {
    "string": {
      "alertTargetArn": "string",
      "roleArn": "string"
    }
  },
  "additionalMetricsToRetain": [
    "string"
  ],
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

```

        "Value": "string"
    }
]
}

```

### cli-input-json fields

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you are giving to the security profile.
securityProfileDescription	string length- max:1000 pattern: [\p{Graph} ]*	A description of the security profile.
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use

Name	Type	Description
		this to specify that set to be compared with the metric.
ports	list member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the metric.
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, <code>NUM_MESSAGES_SENT</code> ). For a <code>statisticalThreshold</code> metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Specifies the destinations to which alerts are sent. (Alerts are always sent to the console.) Alerts are generated when a device (thing) violates a behavior.
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage the security profile.
Key	string	The tag's key.
Value	string	The tag's value.

#### Output

```
{
  "securityProfileName": "string",
```

```
    "securityProfileArn": "string"
}
```

### CLI output fields

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you gave to the security profile.
securityProfileArn	string	The ARN of the security profile.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceAlreadyExistsException`

The resource already exists.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## CreateStream

Creates a stream for delivering one or more large files in chunks over MQTT. A stream transports data bytes in chunks or blocks packaged as MQTT messages from a source like S3. You can have one or more files associated with a stream. The total size of a file associated with the stream cannot exceed more than 2 MB. The stream will be created with version 0. If a stream is created with the same streamID as a stream that existed and was deleted within last 90 days, we will resurrect that old stream by incrementing the version by 1.

### Synopsis

```
aws iot create-stream \
--stream-id <value> \
[--description <value>] \
--files <value> \
--role-arn <value> \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "streamId": "string",
  "description": "string",
  "files": [
```

```
{
  "fileId": "integer",
  "s3Location": {
    "bucket": "string",
    "key": "string",
    "version": "string"
  }
},
"roleArn": "string",
"tags": [
  {
    "Key": "string",
    "Value": "string"
  }
]
}
```

### cli-input-json fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
description	string  length- max:2028  pattern: [^\p{C}]+	A description of the stream.
files	list  member: StreamFile	The files to stream.
fileId	integer  range- max:255 min:0	The file ID.
s3Location	S3Location	The location of the file in S3.
bucket	string  length- min:1	The S3 bucket.
key	string  length- min:1	The S3 key.
version	string	The S3 bucket version.
roleArn	string  length- max:2048 min:20	An IAM role that allows the IoT service principal assumes to access your S3 files.
tags	list  member: Tag  java class: java.util.List	Metadata which can be used to manage streams.

Name	Type	Description
Key	string	The tag's key.
Value	string	The tag's value.

### Output

```
{
  "streamId": "string",
  "streamArn": "string",
  "description": "string",
  "streamVersion": "integer"
}
```

### CLI output fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
streamArn	string	The stream ARN.
description	string  length- max:2028  pattern: [^\p{C}]+	A description of the stream.
streamVersion	integer  range- max:65535 min:0	The version of the stream.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### LimitExceededException

A limit has been exceeded.

#### ResourceNotFoundException

The specified resource does not exist.

#### ResourceAlreadyExistsException

The resource already exists.

#### ThrottlingException

The rate exceeds the limit.

#### UnauthorizedException

You are not authorized to perform this operation.

#### ServiceUnavailableException

The service is temporarily unavailable.

#### InternalFailureException

An unexpected error has occurred.

## CreateThing

Creates a thing record in the registry. If this call is made multiple times using the same thing name and configuration, the call will succeed. If this call is made with the same thing name but different configuration a `ResourceAlreadyExistsException` is thrown.

#### Note

This is a control plane operation. See [Authorization](#) for information about authorizing control plane actions.

#### Synopsis

```
aws iot create-thing \
  --thing-name <value> \
  [--thing-type-name <value>] \
  [--attribute-payload <value>] \
  [--billing-group-name <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "thingName": "string",
  "thingTypeName": "string",
  "attributePayload": {
    "attributes": {
      "string": "string"
    },
    "merge": "boolean"
  },
  "billingGroupName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing to create.
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing type associated with the new thing.

Name	Type	Description
attributePayload	AttributePayload	The attribute payload, which consists of up to three name/value pairs in a JSON document. For example:  <code>\"attributes\":\n{\\"string1\\":\n\\"string2\\\"}</code>
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\"attributes\":\n{\\"string1\\":\n\\"string2\\\"}</code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.  To remove an attribute, call UpdateThing with an empty attribute value.  <b>Note</b> The merge attribute is only valid when calling UpdateThing.
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group the thing will be added to.

## Output

```
{
  "thingName": "string",
  "thingArn": "string",
  "thingId": "string"
}
```

## CLI output fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the new thing.

Name	Type	Description
thingArn	string	The ARN of the new thing.
thingId	string	The thing ID.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceAlreadyExistsException`

The resource already exists.

`ResourceNotFoundException`

The specified resource does not exist.

## CreateThingGroup

Create a thing group.

### Note

This is a control plane operation. See [Authorization](#) for information about authorizing control plane actions.

### Synopsis

```
aws iot create-thing-group \
--thing-group-name <value> \
[--parent-group-name <value>] \
[--thing-group-properties <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "thingGroupName": "string",
  "parentGroupName": "string",
```

```

"thingGroupProperties": {
    "thingGroupDescription": "string",
    "attributePayload": {
        "attributes": {
            "string": "string"
        },
        "merge": "boolean"
    }
},
"tags": [
    {
        "Key": "string",
        "Value": "string"
    }
]
}

```

#### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing group name to create.
parentGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the parent thing group.
thingGroupProperties	ThingGroupProperties	The thing group properties.
thingGroupDescription	string  length- max:2028  pattern: [\p{Graph}]*	The thing group description.
attributePayload	AttributePayload	The thing group attributes in JSON format.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\ "attributes\": {\ \"string1\": \"string2\"}</code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.  To remove an attribute, call UpdateThing with an empty attribute value.

Name	Type	Description
		<b>Note</b> The <code>merge</code> attribute is only valid when calling <code>UpdateThing</code> .
tags	list member: Tag java class: <code>java.util.List</code>	Metadata which can be used to manage the thing group.
Key	string	The tag's key.
Value	string	The tag's value.

## Output

```
{
  "thingGroupName": "string",
  "thingGroupArn": "string",
  "thingGroupId": "string"
}
```

## CLI output fields

Name	Type	Description
thingGroupName	string length- max:128 min:1 pattern: [a-zA-Z0-9-_]+	The thing group name.
thingGroupArn	string	The thing group ARN.
thingGroupId	string length- max:128 min:1 pattern: [a-zA-Z0-9-_]+	The thing group ID.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceAlreadyExistsException

The resource already exists.

### ThrottlingException

The rate exceeds the limit.

### InternalFailureException

An unexpected error has occurred.

# CreateThingType

Creates a new thing type.

## Synopsis

```
aws iot create-thing-type \
--thing-type-name <value> \
[--thing-type-properties <value>] \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingTypeName": "string",
  "thingTypeProperties": {
    "thingTypeDescription": "string",
    "searchableAttributes": [
      "string"
    ]
  },
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

## cli-input-json fields

Name	Type	Description
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing type.
thingTypeProperties	ThingTypeProperties	The ThingTypeProperties for the thing type to create. It contains information about the new thing type including a description, and a list of searchable thing attribute names.
thingTypeDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The description of the thing type.
searchableAttributes	list  member: AttributeName  java class: java.util.List	A list of searchable thing attribute names.

Name	Type	Description
tags	list member: Tag java class: java.util.List	Metadata which can be used to manage the thing type.
Key	string	The tag's key.
Value	string	The tag's value.

## Output

```
{
  "thingTypeName": "string",
  "thingTypeArn": "string",
  "thingTypeId": "string"
}
```

## CLI output fields

Name	Type	Description
thingTypeName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing type.
thingTypeArn	string	The Amazon Resource Name (ARN) of the thing type.
thingTypeld	string	The thing type ID.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceAlreadyExistsException`

The resource already exists.

# CreateTopicRule

Creates a rule. Creating rules is an administrator-level action. Any user who has permission to create rules will be able to access data processed by the rule.

## Synopsis

```
aws iot create-topic-rule \
--rule-name <value> \
--topic-rule-payload <value> \
[--tags <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{
  "ruleName": "string",
  "topicRulePayload": {
    "sql": "string",
    "description": "string",
    "actions": [
      {
        "dynamoDB": {
          "tableName": "string",
          "roleArn": "string",
          "operation": "string",
          "hashKeyField": "string",
          "hashKeyValue": "string",
          "hashKeyType": "string",
          "rangeKeyField": "string",
          "rangeKeyValue": "string",
          "rangeKeyType": "string",
          "payloadField": "string"
        },
        "dynamoDBv2": {
          "roleArn": "string",
          "putItem": {
            "tableName": "string"
          }
        },
        "lambda": {
          "functionArn": "string"
        },
        "sns": {
          "targetArn": "string",
          "roleArn": "string",
          "messageFormat": "string"
        },
        "sqs": {
          "roleArn": "string",
          "queueUrl": "string",
          "useBase64": "boolean"
        },
        "kinesis": {
          "roleArn": "string",
          "streamName": "string",
          "partitionKey": "string"
        },
        "republish": {
          "roleArn": "string",
          "topic": "string"
        }
    ],
    "retentionPeriod": {
      "unit": "seconds",
      "value": 123
    }
  }
}
```

```

"s3": {
    "roleArn": "string",
    "bucketName": "string",
    "key": "string",
    "cannedAcl": "string"
},
"firehose": {
    "roleArn": "string",
    "deliveryStreamName": "string",
    "separator": "string"
},
"cloudwatchMetric": {
    "roleArn": "string",
    "metricNamespace": "string",
    "metricName": "string",
    "metricValue": "string",
    "metricUnit": "string",
    "metricTimestamp": "string"
},
"cloudwatchAlarm": {
    "roleArn": "string",
    "alarmName": "string",
    "stateReason": "string",
    "stateValue": "string"
},
"elasticsearch": {
    "roleArn": "string",
    "endpoint": "string",
    "index": "string",
    "type": "string",
    "id": "string"
},
"salesforce": {
    "token": "string",
    "url": "string"
},
"iotAnalytics": {
    "channelArn": "string",
    "channelName": "string",
    "roleArn": "string"
},
"iotEvents": {
    "inputName": "string",
    "messageId": "string",
    "roleArn": "string"
},
"stepFunctions": {
    "executionNamePrefix": "string",
    "stateMachineName": "string",
    "roleArn": "string"
}
},
],
"ruleDisabled": "boolean",
"awsIotSqlVersion": "string",
"errorAction": {
    "dynamoDB": {
        "tableName": "string",
        "roleArn": "string",
        "operation": "string",
        "hashKeyField": "string",
        "hashKeyValue": "string",
        "hashKeyType": "string",
        "rangeKeyField": "string",
        "rangeKeyValue": "string",
        "rangeKeyType": "string",
    }
}
]
}

```

```
        "payloadField": "string"
    },
    "dynamoDBv2": {
        "roleArn": "string",
        "putItem": {
            "tableName": "string"
        }
    },
    "lambda": {
        "functionArn": "string"
    },
    "sns": {
        "targetArn": "string",
        "roleArn": "string",
        "messageFormat": "string"
    },
    "sqs": {
        "roleArn": "string",
        "queueUrl": "string",
        "useBase64": "boolean"
    },
    "kinesis": {
        "roleArn": "string",
        "streamName": "string",
        "partitionKey": "string"
    },
    "republish": {
        "roleArn": "string",
        "topic": "string"
    },
    "s3": {
        "roleArn": "string",
        "bucketName": "string",
        "key": "string",
        "cannedAcl": "string"
    },
    "firehose": {
        "roleArn": "string",
        "deliveryStreamName": "string",
        "separator": "string"
    },
    "cloudwatchMetric": {
        "roleArn": "string",
        "metricNamespace": "string",
        "metricName": "string",
        "metricValue": "string",
        "metricUnit": "string",
        "metricTimestamp": "string"
    },
    "cloudwatchAlarm": {
        "roleArn": "string",
        "alarmName": "string",
        "stateReason": "string",
        "stateValue": "string"
    },
    "elasticsearch": {
        "roleArn": "string",
        "endpoint": "string",
        "index": "string",
        "type": "string",
        "id": "string"
    },
    "salesforce": {
        "token": "string",
        "url": "string"
    }
},
```

```

    "iotAnalytics": {
        "channelArn": "string",
        "channelName": "string",
        "roleArn": "string"
    },
    "iotEvents": {
        "inputName": "string",
        "messageId": "string",
        "roleArn": "string"
    },
    "stepFunctions": {
        "executionNamePrefix": "string",
        "stateMachineName": "string",
        "roleArn": "string"
    }
},
"tags": "string"
}

```

#### cli-input-json fields

Name	Type	Description
ruleName	string  length- max:128 min:1  pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.
topicRulePayload	TopicRulePayload	The rule payload.
sql	string	The SQL statement used to query the topic. For more information, see <a href="#">AWS IoT SQL Reference</a> in the <a href="#">AWS IoT Developer Guide</a> .
description	string	The description of the rule.
actions	list  member: Action	The actions associated with the rule.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
operation	string	The type of operation to be performed. This follows the substitution template, so it can be <code>\$ operation</code> , but the substitution must result in one of the following: <code>INSERT</code> , <code>UPDATE</code> , or <code>DELETE</code> .
hashKeyField	string	The hash key name.

Name	Type	Description
hashKeyValue	string	The hash key value.
hashKeyType	string	The hash key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
putItem	PutItemInput	Specifies the DynamoDB table to which the message data will be written. For example:  <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> Each attribute in the message payload will be written to a separate column in the DynamoDB database.
tableName	string	The table where the message data will be written.
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.

Name	Type	Description
roleArn	string	The ARN of the IAM role that grants access.
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.

Name	Type	Description
key	string	The object key.
cannedAcl	string	The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a> .
		enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.
separator	string pattern: ([ ] (  ) ())	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.

Name	Type	Description
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.(sfdc-matrix.net) (sfdcnow.com))/streams/w <a href="#">1, 20</a> /w <a href="#">1, 20</a> /event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	iotAnalyticsAction	Sends message data to an AWS IoT Analytics channel.
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).

Name	Type	Description
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").
ruleDisabled	boolean	Specifies whether the rule is disabled.
awsIoTSqlVersion	string	The version of the SQL rules engine to use when evaluating the rule.
errorAction	Action	The action to take when an error occurs.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.

Name	Type	Description
operation	string	The type of operation to be performed. This follows the substitution template, so it can be <code>#\$ operation</code> , but the substitution must result in one of the following: INSERT, UPDATE, or DELETE.
hashKeyField	string	The hash key name.
hashKeyValue	string	The hash key value.
hashKeyType	string	The hash key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
putItem	PutItemInput	Specifies the DynamoDB table to which the message data will be written. For example:  <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> Each attribute in the message payload will be written to a separate column in the DynamoDB database.
tableName	string	The table where the message data will be written.

Name	Type	Description
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.
roleArn	string	The ARN of the IAM role that grants access.
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.

Name	Type	Description
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.
key	string	The object key.
cannedAcl	string	<p>The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a>.</p> <p>enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write</p>
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.
separator	string pattern: ([ ]) ( ) ()	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ';' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .

Name	Type	Description
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.(sfdc-matrix.net) (sfdcnow.com))/streams/w <b>1, 20</b> /w <b>1, 20</b> /event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	IotAnalyticsAction	Sends message data to an AWS IoT Analytics channel.

Name	Type	Description
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").

Name	Type	Description
tags	string	<p>Metadata which can be used to manage the topic rule.</p> <p><b>Note</b>            For URI Request parameters use format: ...key1=value1&amp;key2=value2...            For the CLI command-line parameter use format: --tags "key1=value1&amp;key2=value2..."            For the cli-input-json file use format: "tags": "key1=value1&amp;key2=value2..."</p>

#### Output

None

#### Errors

`SqlParseException`

The Rule-SQL expression can't be parsed correctly.

`InternalException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`ResourceAlreadyExistsException`

The resource already exists.

`ServiceUnavailableException`

The service is temporarily unavailable.

`ConflictingResourceUpdateException`

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

## DeleteAccountAuditConfiguration

Restores the default settings for Device Defender audits for this account. Any configuration data you entered is deleted and all audit checks are reset to disabled.

#### Synopsis

```
aws iot delete-account-audit-configuration \
[--delete-scheduled-audits | --no-delete-scheduled-audits] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "deleteScheduledAudits": "boolean"
}
```

#### **cli-input-json fields**

Name	Type	Description
deleteScheduledAudits	boolean	If true, all scheduled audits are deleted.

#### **Output**

None

#### **Errors**

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DeleteAuthorizer

Deletes an authorizer.

#### **Synopsis**

```
aws iot delete-authorizer \
  --authorizer-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

#### **cli-input-json format**

```
{
  "authorizerName": "string"
}
```

#### **cli-input-json fields**

Name	Type	Description
authorizerName	string length- max:128 min:1 pattern: [w=,@-]+	The name of the authorizer to delete.

**Output**

None

**Errors**

**DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DeleteBillingGroup

Deletes the billing group.

**Synopsis**

```
aws iot delete-billing-group \
  --billing-group-name <value> \
  [--expected-version <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json format**

```
{
  "billingGroupName": "string",
  "expectedVersion": "long"
}
```

**cli-input-json fields**

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.

Name	Type	Description
expectedVersion	long	The expected version of the billing group. If the version of the billing group does not match the expected version specified in the request, the <code>DeleteBillingGroup</code> request is rejected with a <code>VersionConflictException</code> .

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DeleteCACertificate

Deletes a registered CA certificate.

#### Synopsis

```
aws iot delete-ca-certificate \
--certificate-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateId": "string"
}
```

#### cli-input-json fields

Name	Type	Description
certificateId	string	The ID of the certificate to delete. (The last part of the

Name	Type	Description
	length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	certificate ARN contains the certificate ID.)

#### Output

None

#### Errors

**InvalidRequestException**

The contents of the request were invalid.

**CertificateStateException**

The certificate operation is not allowed.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## DeleteCertificate

Deletes the specified certificate.

A certificate cannot be deleted if it has a policy or IoT thing attached to it or if its status is set to ACTIVE. To delete a certificate, first use the DetachPrincipalPolicy API to detach all policies. Next, use the UpdateCertificate API to set the certificate to the INACTIVE status.

#### Synopsis

```
aws iot delete-certificate \
--certificate-id <value> \
[--force-delete | --no-force-delete] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateId": "string",
  "forceDelete": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)
forceDelete	boolean	Forces the deletion of a certificate if it is inactive and is not attached to an IoT thing.

### Output

None

### Errors

`CertificateStateException`

The certificate operation is not allowed.

`DeleteConflictException`

You can't delete the resource because it is attached to one or more resources.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## DeleteDynamicThingGroup

Deletes a dynamic thing group.

### Synopsis

```
aws iot delete-dynamic-thing-group \
--thing-group-name <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingGroupName": "string",  
    "expectedVersion": "long"  
}
```

**cli-input-json fields**

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the dynamic thing group to delete.
expectedVersion	long	The expected version of the dynamic thing group to delete.

**Output**

None

**Errors**

**InvalidRequestException**

The contents of the request were invalid.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DeleteJob

Deletes a job and its related job executions.

Deleting a job may take time, depending on the number of job executions created for the job and various other factors. While the job is being deleted, the status of the job will be shown as "DELETION\_IN\_PROGRESS". Attempting to delete or cancel a job whose status is already "DELETION\_IN\_PROGRESS" will result in an error.

Only 10 jobs may have status "DELETION\_IN\_PROGRESS" at the same time, or a LimitExceededException will occur.

**Synopsis**

```
aws iot delete-job \
```

```
--job-id <value> \
[--force | --no-force] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string",
  "force": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
jobId	string  length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The ID of the job to be deleted.  After a job deletion is completed, you may reuse this jobId when you create a new job. However, this is not recommended, and you must ensure that your devices are not using the jobId to refer to the deleted job.
force	boolean	(Optional) When true, you can delete a job which is "IN_PROGRESS". Otherwise, you can only delete a job which is in a terminal state ("COMPLETED" or "CANCELED") or an exception will occur. The default is false.  <b>Note</b> Deleting a job which is "IN_PROGRESS", will cause a device which is executing the job to be unable to access job information or update the job execution status. Use caution and ensure that each device executing a job which is deleted is able to recover to a valid state.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

#### `InvalidStateException`

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state SUCCESS to state IN\_PROGRESS). In this case, the body of the error message also contains the executionState field.

#### `ResourceNotFoundException`

The specified resource does not exist.

#### `LimitExceededException`

A limit has been exceeded.

#### `ThrottlingException`

The rate exceeds the limit.

#### `ServiceUnavailableException`

The service is temporarily unavailable.

## DeleteJobExecution

Deletes a job execution.

### Synopsis

```
aws iot delete-job-execution \
--job-id <value> \
--thing-name <value> \
--execution-number <value> \
[--force | --no-force] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "jobId": "string",
  "thingName": "string",
  "executionNumber": "long",
  "force": "boolean"
}
```

### `cli-input-json` fields

Name	Type	Description
<code>jobId</code>	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the job whose execution on a particular device will be deleted.
<code>thingName</code>	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the thing whose job execution will be deleted.

Name	Type	Description
executionNumber	long	<p>The ID of the job execution to be deleted. The <code>executionNumber</code> refers to the execution of a particular job on a particular device.</p> <p>Note that once a job execution is deleted, the <code>executionNumber</code> may be reused by IoT, so be sure you get and use the correct value here.</p>
force	boolean	<p>(Optional) When true, you can delete a job execution which is "IN_PROGRESS". Otherwise, you can only delete a job execution which is in a terminal state ("SUCCEEDED", "FAILED", "REJECTED", "REMOVED" or "CANCELED") or an exception will occur. The default is false.</p> <p><b>Note</b>            Deleting a job execution which is "IN_PROGRESS", will cause the device to be unable to access job information or update the job execution status. Use caution and ensure that the device is able to recover to a valid state.</p>

## Output

None

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`InvalidStateTransitionException`

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state `SUCCESS` to state `IN_PROGRESS`). In this case, the body of the error message also contains the `executionState` field.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

#### **ServiceUnavailableException**

The service is temporarily unavailable.

## DeleteOTAUpdate

Delete an OTA update.

### Synopsis

```
aws iot delete-ota-update \
--ota-update-id <value> \
[--delete-stream | --no-delete-stream] \
[--force-delete-aws-job | --no-force-delete-aws-job] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "otaUpdateId": "string",
  "deleteStream": "boolean",
  "forceDeleteAWSJob": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
otaUpdateId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The OTA update ID to delete.
deleteStream	boolean	Specifies if the stream associated with an OTA update should be deleted when the OTA update is deleted.
forceDeleteAWSJob	boolean	Specifies if the AWS Job associated with the OTA update should be deleted with the OTA update is deleted.

### Output

None

### Errors

#### **InvalidRequestException**

The contents of the request were invalid.

#### **ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**InternalFailureException**

An unexpected error has occurred.

**ServiceUnavailableException**

The service is temporarily unavailable.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

## DeletePolicy

Deletes the specified policy.

A policy cannot be deleted if it has non-default versions or it is attached to any certificate.

To delete a policy, use the DeletePolicyVersion API to delete all non-default versions of the policy; use the DetachPrincipalPolicy API to detach the policy from any certificate; and then use the DeletePolicy API to delete the policy.

When a policy is deleted using DeletePolicy, its default version is deleted with it.

### Synopsis

```
aws iot delete-policy \
  --policy-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "policyName": "string"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy to delete.

### Output

None

## Errors

`DeleteConflictException`

You can't delete the resource because it is attached to one or more resources.

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# DeletePolicyVersion

Deletes the specified version of the specified policy. You cannot delete the default version of a policy using this API. To delete the default version of a policy, use `DeletePolicy`. To find out which version of a policy is marked as the default version, use `ListPolicyVersions`.

## Synopsis

```
aws iot delete-policy-version \
--policy-name <value> \
--policy-version-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "policyVersionId": "string"
}
```

## cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.

Name	Type	Description
policyVersionId	string pattern: [0-9]+	The policy version ID.

#### Output

None

#### Errors

`DeleteConflictException`

You can't delete the resource because it is attached to one or more resources.

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## DeleteRegistrationCode

Deletes a CA certificate registration code.

#### Synopsis

```
aws iot delete-registration-code \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

#### Output

None

## Errors

**ThrottlingException**

The rate exceeds the limit.

**ResourceNotFoundException**

The specified resource does not exist.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

# DeleteRoleAlias

Deletes a role alias

## Synopsis

```
aws iot delete-role-alias \
--role-alias <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "roleAlias": "string"  
}
```

## cli-input-json fields

Name	Type	Description
roleAlias	string  length- max:128 min:1  pattern: [w=@-]+	The role alias to delete.

## Output

None

## Errors

**DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## DeleteScheduledAudit

Deletes a scheduled audit.

### Synopsis

```
aws iot delete-scheduled-audit \
--scheduled-audit-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "scheduledAuditName": "string"  
}
```

### cli-input-json fields

Name	Type	Description
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit you want to delete.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DeleteSecurityProfile

Deletes a Device Defender security profile.

### Synopsis

```
aws iot delete-security-profile \
--security-profile-name <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "securityProfileName": "string",
  "expectedVersion": "long"
}
```

### cli-input-json fields

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the security profile to be deleted.
expectedVersion	long	The expected version of the security profile. A new version is generated whenever the security profile is updated. If you specify a value that is different than the actual version, a <code>VersionConflictException</code> is thrown.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

## DeleteStream

Deletes a stream.

### Synopsis

```
aws iot delete-stream \
--stream-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "streamId": "string"
}
```

### cli-input-json fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.

### Output

None

### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**DeleteConflictException**

You can't delete the resource because it is attached to one or more resources.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DeleteThing

Deletes the specified thing. Returns successfully with no error if the deletion is successful or you specify a thing that doesn't exist.

### Synopsis

```
aws iot delete-thing \
  --thing-name <value> \
  [--expected-version <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
  "thingName": "string",  
  "expectedVersion": "long"  
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing to delete.
expectedVersion	long	The expected version of the thing record in the registry. If the version of the record in the registry does not match the expected version specified in the request, the DeleteThing request is rejected with a VersionConflictException.

### Output

None

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# DeleteThingGroup

Deletes a thing group.

## Synopsis

```
aws iot delete-thing-group \
--thing-group-name <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingGroupName": "string",  
    "expectedVersion": "long"  
}
```

## cli-input-json fields

Name	Type	Description
thingGroupName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing group to delete.

Name	Type	Description
expectedVersion	long	The expected version of the thing group to delete.

#### Output

None

#### Errors

**InvalidRequestException**

The contents of the request were invalid.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DeleteThingShadow

Deletes the shadow for the specified thing.

For more information, see [DeleteThingShadow](#) in the AWS IoT Developer Guide.

#### Synopsis

```
aws iot-data delete-thing-shadow \
--thing-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.

## Output

```
{  
    "payload": "blob"  
}
```

## CLI output fields

Name	Type	Description
payload	blob	The state information, in JSON format.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`MethodNotAllowedException`

The specified combination of HTTP verb and URI is not supported.

`UnsupportedDocumentEncodingException`

The encoding is not supported.

# DeleteThingType

Deletes the specified thing type. You cannot delete a thing type if it has things associated with it. To delete a thing type, first mark it as deprecated by calling `DeprecateThingType`, then remove any associated things by calling `UpdateThing` to change the thing type on any associated thing, and finally use `DeleteThingType` to delete the thing type.

## Synopsis

```
aws iot delete-thing-type \  
    --thing-type-name <value> \  
    [--cli-input-json <value>] \  

```

```
[--generate-cli-skeleton]
```

cli-input-json format

```
{  
    "thingTypeName": "string"  
}
```

#### cli-input-json fields

Name	Type	Description
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing type.

#### Output

None

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## DeleteTopicRule

Deletes the rule.

#### Synopsis

```
aws iot delete-topic-rule \  
[--rule-name <value>] \  
[--cli-input-json <value>] \  

```

[--generate-cli-skeleton]

**cli-input-json** format

```
{
  "ruleName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
ruleName	string  length- max:128 min:1  pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.

#### Output

None

#### Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

**UnauthorizedException**

You are not authorized to perform this operation.

**ConflictingResourceUpdateException**

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

## DeleteV2LoggingLevel

Deletes a logging level.

#### Synopsis

```
aws iot delete-v2-logging-level \
--target-type <value> \
--target-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "targetType": "string",
  "targetName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
targetType	string	The type of resource for which you are configuring logging. Must be THING_Group. enum: DEFAULT   THING_GROUP
targetName	string	The name of the resource for which you are configuring logging.

#### Output

None

#### Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

## DeprecateThingType

Deprecates a thing type. You can not associate new things with deprecated thing type.

#### Synopsis

```
aws iot deprecate-thing-type \
  --thing-type-name <value> \
  [--undo-deprecate | --no-undo-deprecate] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "thingTypeName": "string",
  "undoDeprecate": "boolean"
}
```

### **cli-input-json fields**

Name	Type	Description
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing type to deprecate.
undoDeprecate	boolean	Whether to undelete a deprecated thing type. If <b>true</b> , the thing type will not be deprecated anymore and you can associate it with things.

### **Output**

None

### **Errors**

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DescribeAccountAuditConfiguration

Gets information about the Device Defender audit settings for this account. Settings include how audit notifications are sent and which audit checks are enabled or disabled.

### **Synopsis**

```
aws iot describe-account-audit-configuration \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format**

```
{
}
```

### Output

```
{
    "roleArn": "string",
    "auditNotificationTargetConfigurations": {
        "string": {
            "targetArn": "string",
            "roleArn": "string",
            "enabled": "boolean"
        }
    },
    "auditCheckConfigurations": {
        "string": {
            "enabled": "boolean"
        }
    }
}
```

### CLI output fields

Name	Type	Description
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to AWS IoT to access information about your devices, policies, certificates and other items as necessary when performing an audit.  On the first call to <code>UpdateAccountAuditConfiguration</code> this parameter is required.
auditNotificationTargetConfigurations	map	Information about the targets to which audit notifications are sent for this account.
targetArn	string	The ARN of the target (SNS topic) to which audit notifications are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send notifications to the target.
enabled	boolean	True if notifications to the target are enabled.
auditCheckConfigurations	map	Which audit checks are enabled and disabled for this account.
enabled	boolean	True if this audit check is enabled for this account.

### Errors

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DescribeAuditTask

Gets information about a Device Defender audit.

### Synopsis

```
aws iot describe-audit-task \
  --task-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "taskId": "string"
}
```

### cli-input-json fields

Name	Type	Description
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9-]+	The ID of the audit whose information you want to get.

### Output

```
{
  "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",
      "resourceArn": "string"
    }
  }
}
```

```

        "errorCode": "string",
        "message": "string"
    }
}
}
```

### CLI output fields

Name	Type	Description
taskStatus	string	The status of the audit: one of "IN_PROGRESS", "COMPLETED", "FAILED", or "CANCELED".  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED
taskType	string	The type of audit: "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK
taskStartTime	timestamp	The time the audit started.
taskStatistics	TaskStatistics	Statistical information about the audit.
totalChecks	integer	The number of checks in this audit.
inProgressChecks	integer	The number of checks in progress.
waitingForDataCollectionChecks	integer	The number of checks waiting for data collection.
compliantChecks	integer	The number of checks that found compliant resources.
nonCompliantChecks	integer	The number of checks that found non-compliant resources.
failedChecks	integer	The number of checks
canceledChecks	integer	The number of checks that did not run because the audit was canceled.
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit (only if the audit was a scheduled audit).
auditDetails	map	Detailed information about each check performed during this audit.

Name	Type	Description
checkRunStatus	string	The completion status of this check, one of "IN_PROGRESS", "WAITING_FOR_DATA_COLLECTION", "CANCELED", "COMPLETED_COMPLIANT", "COMPLETED_NON_COMPLIANT", or "FAILED".  enum: IN_PROGRESS   WAITING_FOR_DATA_COLLECTION   CANCELED   COMPLETED_COMPLIANT   COMPLETED_NON_COMPLIANT   FAILED
checkCompliant	boolean	True if the check completed and found all resources compliant.
totalResourcesCount	long	The number of resources on which the check was performed.
nonCompliantResourcesCount	long	The number of resources that the check found non-compliant.
errorCode	string	The code of any error encountered when performing this check during this audit. One of "INSUFFICIENT_PERMISSIONS", or "AUDIT_CHECK_DISABLED".
message	string length- max:2048	The message associated with any error encountered when performing this check during this audit.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### InternalFailureException

An unexpected error has occurred.

## DescribeAuthorizer

Describes an authorizer.

## Synopsis

```
aws iot describe-authorizer \
--authorizer-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "authorizerName": "string"
}
```

## cli-input-json fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The name of the authorizer to describe.

## Output

```
{
  "authorizerDescription": {
    "authorizerName": "string",
    "authorizerArn": "string",
    "authorizerFunctionArn": "string",
    "tokenKeyName": "string",
    "tokensigningPublicKeys": {
      "string": "string"
    },
    "status": "string",
    "creationDate": "timestamp",
    "lastModifiedDate": "timestamp"
  }
}
```

## CLI output fields

Name	Type	Description
authorizerDescription	AuthorizerDescription	The authorizer description.
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.
authorizerArn	string	The authorizer ARN.
authorizerFunctionArn	string	The authorizer's Lambda function ARN.
tokenKeyName	string	The key used to extract the token from the HTTP headers.

Name	Type	Description
	length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	
tokenSigningPublicKeys	map	The public keys used to validate the token signature returned by your custom authentication service.
status	string	The status of the authorizer. enum: ACTIVE   INACTIVE
creationDate	timestamp	The UNIX timestamp of when the authorizer was created.
lastModifiedDate	timestamp	The UNIX timestamp of when the authorizer was last updated.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# DescribeBillingGroup

Returns information about a billing group.

## Synopsis

```
aws iot describe-billing-group \
--billing-group-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
```

```

    "billingGroupName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.

#### Output

```
{
  "billingGroupName": "string",
  "billingGroupId": "string",
  "billingGroupArn": "string",
  "version": "long",
  "billingGroupProperties": {
    "billingGroupDescription": "string"
  },
  "billingGroupMetadata": {
    "creationDate": "timestamp"
  }
}
```

#### CLI output fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.
billingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The ID of the billing group.
billingGroupArn	string	The ARN of the billing group.
version	long	The version of the billing group.
billingGroupProperties	BillingGroupProperties	The properties of the billing group.
billingGroupDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The description of the billing group.
billingGroupMetadata	BillingGroupMetadata	Additional information about the billing group.

Name	Type	Description
creationDate	timestamp	The date the billing group was created.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## DescribeCACertificate

Describes a registered CA certificate.

### Synopsis

```
aws iot describe-ca-certificate \
--certificate-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "certificateId": "string"
}
```

### cli-input-json fields

Name	Type	Description
<code>certificateId</code>	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The CA certificate identifier.

### Output

```
{
  "certificateDescription": {
    "certificateArn": "string",
    "certificateId": "string",
    "status": "string",
    "certificatePem": "string",
```

```

    "ownedBy": "string",
    "creationDate": "timestamp",
    "autoRegistrationStatus": "string",
    "lastModifiedDate": "timestamp",
    "customerVersion": "integer",
    "generationId": "string",
    "validity": {
        "notBefore": "timestamp",
        "notAfter": "timestamp"
    },
    "registrationConfig": {
        "templateBody": "string",
        "roleArn": "string"
    }
}

```

### CLI output fields

Name	Type	Description
certificateDescription	CACertificateDescription	The CA certificate description.
certificateArn	string	The CA certificate ARN.
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The CA certificate ID.
status	string  enum: ACTIVE   INACTIVE	The status of a CA certificate.
certificatePem	string  length- max:65536 min:1	The CA certificate data, in PEM format.
ownedBy	string  length- max:12 min:12  pattern: [0-9]+	The owner of the CA certificate.
creationDate	timestamp	The date the CA certificate was created.
autoRegistrationStatus	string  enum: ENABLE   DISABLE	Whether the CA certificate configured for auto registration of device certificates. Valid values are "ENABLE" and "DISABLE"
lastModifiedDate	timestamp	The date the CA certificate was last modified.
customerVersion	integer  range- min:1	The customer version of the CA certificate.

Name	Type	Description
generationId	string	The generation ID of the CA certificate.
validity	CertificateValidity	When the CA certificate is valid.
notBefore	timestamp	The certificate is not valid before this date.
notAfter	timestamp	The certificate is not valid after this date.
registrationConfig	RegistrationConfig	Information about the registration configuration.
templateBody	string	The template body.
roleArn	string length- max:2048 min:20	The ARN of the role.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## DescribeCertificate

Gets information about the specified certificate.

### Synopsis

```
aws iot describe-certificate \
--certificate-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format**

```
{
```

```

    "certificateId": "string"
}
```

#### cli-input-json fields

Name	Type	Description
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)

#### Output

```
{
  "certificateDescription": {
    "certificateArn": "string",
    "certificateId": "string",
    "caCertificateId": "string",
    "status": "string",
    "certificatePem": "string",
    "ownedBy": "string",
    "previousOwnedBy": "string",
    "creationDate": "timestamp",
    "lastModifiedDate": "timestamp",
    "customerVersion": "integer",
    "transferData": {
      "transferMessage": "string",
      "rejectReason": "string",
      "transferDate": "timestamp",
      "acceptDate": "timestamp",
      "rejectDate": "timestamp"
    },
    "generationId": "string",
    "validity": {
      "notBefore": "timestamp",
      "notAfter": "timestamp"
    }
  }
}
```

#### CLI output fields

Name	Type	Description
certificateDescription	CertificateDescription	The description of the certificate.
certificateArn	string	The ARN of the certificate.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate.
caCertificateId	string length- max:64 min:64	The certificate ID of the CA certificate used to sign this certificate.

Name	Type	Description
	pattern: (0x)?[a-fA-F0-9]+	
status	string	The status of the certificate.  enum: ACTIVE   INACTIVE   REVOKED   PENDING_TRANSFER   REGISTER_INACTIVE   PENDING_ACTIVATION
certificatePem	string  length- max:65536 min:1	The certificate data, in PEM format.
ownedBy	string  length- max:12 min:12  pattern: [0-9]+	The ID of the AWS account that owns the certificate.
previousOwnedBy	string  length- max:12 min:12  pattern: [0-9]+	The ID of the AWS account of the previous owner of the certificate.
creationDate	timestamp	The date and time the certificate was created.
lastModifiedDate	timestamp	The date and time the certificate was last modified.
customerVersion	integer  range- min:1	The customer version of the certificate.
transferData	TransferData	The transfer data.
transferMessage	string  length- max:128	The transfer message.
rejectReason	string  length- max:128	The reason why the transfer was rejected.
transferDate	timestamp	The date the transfer took place.
acceptDate	timestamp	The date the transfer was accepted.
rejectDate	timestamp	The date the transfer was rejected.
generationId	string	The generation ID of the certificate.
validity	CertificateValidity	When the certificate is valid.
notBefore	timestamp	The certificate is not valid before this date.

Name	Type	Description
notAfter	timestamp	The certificate is not valid after this date.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## DescribeDefaultAuthorizer

Describes the default authorizer.

### Synopsis

```
aws iot describe-default-authorizer \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

### Output

```
{
  "authorizerDescription": {
    "authorizerName": "string",
    "authorizerArn": "string",
    "authorizerFunctionArn": "string",
    "tokenKeyName": "string",
    "tokenSigningPublicKeys": {
      "string": "string"
    },
  },
```

```

        "status": "string",
        "creationDate": "timestamp",
        "lastModifiedDate": "timestamp"
    }
}

```

## CLI output fields

Name	Type	Description
authorizerDescription	AuthorizerDescription	The default authorizer's description.
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.
authorizerArn	string	The authorizer ARN.
authorizerFunctionArn	string	The authorizer's Lambda function ARN.
tokenKeyName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The key used to extract the token from the HTTP headers.
tokenSigningPublicKeys	map	The public keys used to validate the token signature returned by your custom authentication service.
status	string  enum: ACTIVE   INACTIVE	The status of the authorizer.
creationDate	timestamp	The UNIX timestamp of when the authorizer was created.
lastModifiedDate	timestamp	The UNIX timestamp of when the authorizer was last updated.

## Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DescribeEndpoint

Returns a unique endpoint specific to the AWS account making the call.

### Synopsis

```
aws iot describe-endpoint \
[--endpoint-type <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "endpointType": "string"  
}
```

### cli-input-json fields

Name	Type	Description
endpointType	string	<p>The endpoint type. Valid endpoint types include:</p> <ul style="list-style-type: none"><li>• <b>iot:Data</b> - Returns a VeriSign signed data endpoint.</li><li>• <b>iot:Data-ATS</b> - Returns an ATS signed data endpoint.</li><li>• <b>iot:CredentialProvider</b> - Returns an AWS IoT credentials provider API endpoint.</li><li>• <b>iot:Jobs</b> - Returns an AWS IoT device management Jobs API endpoint.</li></ul>

### Output

```
{  
    "endpointAddress": "string"  
}
```

### CLI output fields

Name	Type	Description
endpointAddress	string	The endpoint. The format of the endpoint is as follows: <i>identifier.iot.region.amazonaws.com</i> .

### Errors

`InternalFailureException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`UnauthorizedException`

You are not authorized to perform this operation.

`ThrottlingException`

The rate exceeds the limit.

## DescribeEventConfigurations

Describes event configurations.

### Synopsis

```
aws iot describe-event-configurations \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

### Output

```
{
  "eventConfigurations": {
    "string": {
      "Enabled": "boolean"
    },
    "creationDate": "timestamp",
    "lastModifiedDate": "timestamp"
  }
}
```

### CLI output fields

Name	Type	Description
eventConfigurations	map	The event configurations.

Name	Type	Description
Enabled	boolean	True to enable the configuration.
creationDate	timestamp	The creation date of the event configuration.
lastModifiedDate	timestamp	The date the event configurations were last modified.

### Errors

`InternalFailureException`

An unexpected error has occurred.

`ThrottlingException`

The rate exceeds the limit.

## DescribeIndex

Describes a search index.

### Synopsis

```
aws iot describe-index \
  --index-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "indexName": "string"
}
```

### cli-input-json fields

Name	Type	Description
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The index name.

### Output

```
{
  "indexName": "string",
  "indexStatus": "string",
  "schema": "string"
```

}

## CLI output fields

Name	Type	Description
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The index name.
indexStatus	string	The index status.  enum: ACTIVE   BUILDING   REBUILDING
schema	string	Contains a value that specifies the type of indexing performed. Valid values are: <ul style="list-style-type: none"><li>• REGISTRY – Your thing index contains only registry data.</li><li>• REGISTRY_AND_SHADOW<ul style="list-style-type: none"><li>- Your thing index contains registry data and shadow data.</li></ul></li><li>• REGISTRY_AND_CONNECTIVITY_STATUS<ul style="list-style-type: none"><li>- Your thing index contains registry data and thing connectivity status data.</li></ul></li><li>• REGISTRY_AND_SHADOW_AND_CONNECTIVITY_STATUS<ul style="list-style-type: none"><li>- Your thing index contains registry data, shadow data, and thing connectivity status data.</li></ul></li></ul>

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

# DescribeJob

Describes a job.

## Synopsis

```
aws iot describe-job \
--job-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string"
}
```

## cli-input-json fields

Name	Type	Description
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.

## Output

```
{
  "documentSource": "string",
  "job": {
    "jobArn": "string",
    "jobId": "string",
    "targetSelection": "string",
    "status": "string",
    "forceCanceled": "boolean",
    "reasonCode": "string",
    "comment": "string",
    "targets": [
      "string"
    ],
    "description": "string",
    "presignedUrlConfig": {
      "roleArn": "string",
      "expiresInSec": "long"
    },
    "jobExecutionsRolloutConfig": {
      "maximumPerMinute": "integer",
      "exponentialRate": {
        "baseRatePerMinute": "integer",
        "incrementFactor": "double",
        "rateIncreaseCriteria": {
          "numberOfNotifiedThings": "integer",
          "numberOfSucceededThings": "integer"
        }
      }
    },
    "abortConfig": {
      "criteriaList": [
        "string"
      ]
    }
  }
}
```

```
{
    "failureType": "string",
    "action": "string",
    "thresholdPercentage": "double",
    "minNumberOfExecutedThings": "integer"
}
],
},
"createdAt": "timestamp",
"lastUpdatedAt": "timestamp",
"completedAt": "timestamp",
"jobProcessDetails": {
    "processingTargets": [
        "string"
    ],
    "numberOfCanceledThings": "integer",
    "numberOfSucceededThings": "integer",
    "numberOfFailedThings": "integer",
    "numberOfRejectedThings": "integer",
    "numberOfQueuedThings": "integer",
    "numberOfInProgressThings": "integer",
    "numberOfRemovedThings": "integer",
    "numberOfTimedOutThings": "integer"
},
"timeoutConfig": {
    "inProgressTimeoutInMinutes": "long"
}
}
}
```

## CLI output fields

Name	Type	Description
documentSource	string  length- max:1350 min:1	An S3 link to the job document.
job	Job	Information about the job.
jobArn	string	An ARN identifying the job with format "arn:aws:iot:region:account:job/jobId".
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
targetSelection	string	Specifies whether the job will continue to run (CONTINUOUS), or will be complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous, the job may also be run on a thing when a change is detected in a target. For example, a job will run on a device when the thing representing the device is added

Name	Type	Description
		to a target group, even after the job was completed by all things originally in the group.  enum: CONTINUOUS   SNAPSHOT
status	string	The status of the job, one of IN_PROGRESS, CANCELED, DELETION_IN_PROGRESS or COMPLETED.  enum: IN_PROGRESS   CANCELED   COMPLETED   DELETION_IN_PROGRESS
forceCanceled	boolean	Will be true if the job was canceled with the optional force parameter set to true.
reasonCode	string  length- max:128  pattern: [\p{Upper}p Digit_]+	If the job was updated, provides the reason code for the update.
comment	string  length- max:2028  pattern: [^\p{C}]+	If the job was updated, describes the reason for the update.
targets	list  member: TargetArn	A list of IoT things and thing groups to which the job should be sent.
description	string  length- max:2028  pattern: [^\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration for pre-signed S3 URLs.
roleArn	string  length- max:2048 min:20	The ARN of an IAM role that grants permission to download files from the S3 bucket where the job data/updates are stored. The role must also grant permission for IoT to download the files.

Name	Type	Description
expiresInSec	long range- max:3600 min:60	How long (in seconds) pre-signed URLs are valid. Valid values are 60 - 3600, the default value is 3600 seconds. Pre-signed URLs are generated when Jobs receives an MQTT request for the job document.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of a job.
maximumPerMinute	integer range- min:1	The maximum number of things that will be notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	integer range- max:1000 min:1	The minimum number of things that will be notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.
rateIncreaseCriteria	RateIncreaseCriteria	The criteria to initiate the increase in rate of rollout for a job.  AWS IoT supports up to one digit after the decimal (for example, 1.5, but not 1.55).
numberOfNotifiedThings	integer range- min:1	The threshold for number of notified things that will initiate the increase in rate of rollout.
numberOfSucceededThings	integer range- min:1	The threshold for number of succeeded things that will initiate the increase in rate of rollout.
abortConfig	AbortConfig	Configuration for criteria to abort the job.
criteriaList	list member: AbortCriteria java class: java.util.List	The list of abort criteria to define rules to abort the job.

Name	Type	Description
failureType	string	The type of job execution failure to define a rule to initiate a job abort.  enum: FAILED   REJECTED   TIMED_OUT   ALL
action	string	The type of abort action to initiate a job abort.  enum: CANCEL
minNumberOfExecutedThings	integer  range- min:1	Minimum number of executed things before evaluating an abort rule.
createdAt	timestamp	The time, in seconds since the epoch, when the job was created.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job was last updated.
completedAt	timestamp	The time, in seconds since the epoch, when the job was completed.
jobProcessDetails	JobProcessDetails	Details about the job process.
processingTargets	list  member: ProcessingTargetName  java class: java.util.List	The target devices to which the job execution is being rolled out. This value will be null after the job execution has finished rolling out to all the target devices.
numberOfCanceledThings	integer	The number of things that cancelled the job.
numberOfSucceededThings	integer	The number of things which successfully completed the job.
numberOfFailedThings	integer	The number of things that failed executing the job.
numberOfRejectedThings	integer	The number of things that rejected the job.
numberOfQueuedThings	integer	The number of things that are awaiting execution of the job.
numberOfInProgressThings	integer	The number of things currently executing the job.

Name	Type	Description
numberOfRemovedThings	integer	The number of things that are no longer scheduled to execute the job because they have been deleted or have been removed from the group that was a target of the job.
numberOfTimedOutThings	integer	The number of things whose job execution status is <code>TIMED_OUT</code> .
timeoutConfig	TimeoutConfig	Specifies the amount of time each device has to finish its execution of the job. A timer is started when the job execution status is set to <code>IN_PROGRESS</code> . If the job execution status is not set to another terminal state before the timer expires, it will be automatically set to <code>TIMED_OUT</code> .
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The in progress timer can't be updated and will apply to all job executions for the job. Whenever a job execution remains in the <code>IN_PROGRESS</code> status for longer than this interval, the job execution will fail and switch to the terminal <code>TIMED_OUT</code> status.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### ServiceUnavailableException

The service is temporarily unavailable.

## DescribeJobExecution

Gets details of a job execution.

## Synopsis

```
aws iot-jobs-data describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--include-job-document | --no-include-job-document] \
[--execution-number <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string",
  "thingName": "string",
  "includeJobDocument": "boolean",
  "executionNumber": "long"
}
```

## cli-input-json fields

Name	Type	Description
jobId	string pattern: [a-zA-Z0-9_-]+ ^\$next	The unique identifier assigned to this job when it was created.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The thing name associated with the device the job execution is running on.
includeJobDocument	boolean	Optional. Unless set to false, the response contains the job document. The default is true.
executionNumber	long	Optional. A number that identifies a particular job execution on a particular device. If not specified, the latest job execution is returned.

## Output

```
{
  "execution": {
    "jobId": "string",
    "thingName": "string",
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "queuedAt": "long",
    "startedAt": "long",
    "lastUpdatedAt": "long",
    "approximateSecondsBeforeTimedOut": "long",
    "versionNumber": "long",
    "executionNumber": "long",
    "jobDocument": "string"
  }
}
```

```
}
```

### CLI output fields

Name	Type	Description
execution	JobExecution	Contains data about a job execution.
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the thing that is executing the job.
status	string	The status of the job execution. Can be one of: "QUEUED", "IN_PROGRESS", "FAILED", "SUCCESS", "CANCELED", "TIMED_OUT", "REJECTED", or "REMOVED".  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
statusDetails	map	A collection of name/value pairs that describe the status of the job execution.
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long	The time, in seconds since the epoch, when the job execution was started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
approximateSecondsBeforeTimedOut	long	The estimated number of seconds that remain before the job execution status will be changed to <code>TIMED_OUT</code> . The actual job execution timeout can occur up to 60 seconds later than the estimated duration.
versionNumber	long	The version of the job execution. Job execution versions are

Name	Type	Description
		incremented each time they are updated by a device.
executionNumber	long	A number that identifies a particular job execution on a particular device. It can be used later in commands that return or update job execution information.
jobDocument	string length- max:32768	The content of the job document.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

`CertificateValidationException`

The certificate is invalid.

`TerminalStateException`

The job is in a terminal state.

## DescribeJobExecution

Describes a job execution.

### Synopsis

```
aws iot describe-job-execution \
--job-id <value> \
--thing-name <value> \
[--execution-number <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "jobId": "string",
  "thingName": "string",
  "executionNumber": "long"
}
```

### cli-input-json fields

Name	Type	Description
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the thing on which the job execution is running.
executionNumber	long	A string (consisting of the digits "0" through "9" which is used to specify a particular job execution on a particular device.

### Output

```
{
  "execution": {
    "jobId": "string",
    "status": "string",
    "forceCanceled": "boolean",
    "statusDetails": {
      "detailsMap": {
        "string": "string"
      }
    },
    "thingArn": "string",
    "queuedAt": "timestamp",
    "startedAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "executionNumber": "long",
    "versionNumber": "long",
    "approximateSecondsBeforeTimedOut": "long"
  }
}
```

### CLI output fields

Name	Type	Description
execution	JobExecution	Information about the job execution.
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to the job when it was created.
status	string	The status of the job execution (IN_PROGRESS, QUEUED, FAILED, SUCCEEDED, TIMED_OUT, CANCELED, or REJECTED).

Name	Type	Description
		enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
forceCanceled	boolean	Will be true if the job execution was canceled with the optional force parameter set to true.
statusDetails	JobExecutionStatusDetails	A collection of name/value pairs that describe the status of the job execution.
detailsMap	map	The job execution status.
thingArn	string	The ARN of the thing on which the job execution is running.
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long	A string (consisting of the digits "0" through "9") which identifies this particular job execution on this particular device. It can be used in commands which return or update job execution information.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
approximateSecondsBeforeTimedOut	long	The estimated number of seconds that remain before the job execution status will be changed to TIMED_OUT. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The actual job execution timeout can occur up to 60 seconds later than the estimated duration. This value will not be included if the job execution has reached a terminal status.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

# DescribeRoleAlias

Describes a role alias.

## Synopsis

```
aws iot describe-role-alias \
  --role-alias <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
  "roleAlias": "string"  
}
```

## cli-input-json fields

Name	Type	Description
roleAlias	string  length- max:128 min:1  pattern: [w=,@-]+	The role alias to describe.

## Output

```
{  
  "roleAliasDescription": {  
    "roleAlias": "string",  
    "roleAliasArn": "string",  
    "roleArn": "string",  
    "owner": "string",  
    "credentialDurationSeconds": "integer",  
    "creationDate": "timestamp",  
    "lastModifiedDate": "timestamp"
```

```
}
```

## CLI output fields

Name	Type	Description
roleAliasDescription	RoleAliasDescription	The role alias description.
roleAlias	string  length- max:128 min:1  pattern: [w=,@-]+	The role alias.
roleAliasArn	string	The ARN of the role alias.
roleArn	string  length- max:2048 min:20	The role ARN.
owner	string  length- max:12 min:12  pattern: [0-9]+	The role alias owner.
credentialDurationSeconds	integer  range- max:3600 min:900	The number of seconds for which the credential is valid.
creationDate	timestamp	The UNIX timestamp of when the role alias was created.
lastModifiedDate	timestamp	The UNIX timestamp of when the role alias was last modified.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

# DescribeScheduledAudit

Gets information about a scheduled audit.

## Synopsis

```
aws iot describe-scheduled-audit \
--scheduled-audit-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "scheduledAuditName": "string"
}
```

## cli-input-json fields

Name	Type	Description
scheduledAuditName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit whose information you want to get.

## Output

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
  "targetCheckNames": [
    "string"
  ],
  "scheduledAuditName": "string",
  "scheduledAuditArn": "string"
}
```

## CLI output fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. One of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string pattern: ^([1-9] 1[2][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Will be "1" through "31" or "LAST". If days 29-31 are

Name	Type	Description
		specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. One of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT".  enum: SUN   MON   TUE   WED   THU   FRI   SAT
targetCheckNames	list  member: AuditCheckName	Which checks are performed during the scheduled audit. (Note that checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit.
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## DescribeSecurityProfile

Gets information about a Device Defender security profile.

### Synopsis

```
aws iot describe-security-profile \
```

```
--security-profile-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "securityProfileName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the security profile whose information you want to get.

#### Output

```
{
  "securityProfileName": "string",
  "securityProfileArn": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ],
  "alertTargets": {
    "string": {
      "alertTargetArn": "string",
      "roleArn": "string"
    }
  },
  "additionalMetricsToRetain": [
    "string"
  ],
  "version": "long",
  "creationDate": "timestamp",
  "lastModifiedDate": "timestamp"
}
```

```

        "lastModifiedDate": "timestamp"
    }
}
```

### CLI output fields

Name	Type	Description
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the security profile.
securityProfileArn	string	The ARN of the security profile.
securityProfileDescription	string length- max:1000 pattern: [\p{Graph} ]*	A description of the security profile (associated with the security profile when it was created or updated).
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cids	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use

Name	Type	Description
		this to specify that set to be compared with the metric.
ports	list member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.
version	long	The version of the security profile. A new version is generated whenever the security profile is updated.
creationDate	timestamp	The time the security profile was created.
lastModifiedDate	timestamp	The time the security profile was last modified.

## Errors

### InvalidRequestException

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DescribeStream

Gets information about a stream.

### Synopsis

```
aws iot describe-stream \
  --stream-id <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
  "streamId": "string"  
}
```

### cli-input-json fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.

### Output

```
{  
  "streamInfo": {  
    "streamId": "string",  
    "streamArn": "string",  
    "streamVersion": "integer",  
    "description": "string",  
    "files": [  
      {  
        "fileId": "integer",  
        "s3Location": {  
          "bucket": "string",  
          "key": "string",  
          "version": "string"  
        }  
      }  
    ],  
  },  
}
```

```

    "createdAt": "timestamp",
    "lastUpdatedAt": "timestamp",
    "roleArn": "string"
}
}

```

### CLI output fields

Name	Type	Description
streamInfo	StreamInfo	Information about the stream.
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
streamArn	string	The stream ARN.
streamVersion	integer  range- max:65535 min:0	The stream version.
description	string  length- max:2028  pattern: [^\p{C}]+	The description of the stream.
files	list  member: StreamFile	The files to stream.
fileId	integer  range- max:255 min:0	The file ID.
s3Location	S3Location	The location of the file in S3.
bucket	string  length- min:1	The S3 bucket.
key	string  length- min:1	The S3 key.
version	string	The S3 bucket version.
createdAt	timestamp	The date when the stream was created.
lastUpdatedAt	timestamp	The date when the stream was last updated.
roleArn	string  length- max:2048 min:20	An IAM role AWS IoT assumes to access your S3 files.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DescribeThing

Gets information about the specified thing.

### Synopsis

```
aws iot describe-thing \
  --thing-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingName": "string"  
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.

### Output

```
{  
    "defaultClientId": "string",  
    "thingName": "string",  
    "thingId": "string",  
    "thingArn": "string",  
    "thingTypeName": "string",  
    "attributes": {
```

```

        "string": "string"
    },
    "version": "long",
    "billingGroupName": "string"
}

```

### CLI output fields

Name	Type	Description
defaultClientId	string	The default client ID.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.
thingId	string	The ID of the thing to describe.
thingArn	string	The ARN of the thing to describe.
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing type name.
attributes	map	The thing attributes.
version	long	The current version of the thing record in the registry.  <b>Note</b> To avoid unintentional changes to the information in the registry, you can pass the version information in the <code>expectedVersion</code> parameter of the <code>UpdateThing</code> and <code>DeleteThing</code> calls.
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group the thing belongs to.

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DescribeThingGroup

Describe a thing group.

### Synopsis

```
aws iot describe-thing-group \
  --thing-group-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingGroupName": "string"  
}
```

### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing group.

### Output

```
{  
    "thingGroupName": "string",  
    "thingGroupId": "string",  
    "thingGroupArn": "string",  
    "version": "long",  
    "thingGroupProperties": {  
        "thingGroupDescription": "string",  
        "attributePayload": {  
            "attributes": {  
                "string": "string"  
            },  
            "merge": "boolean"  
        }  
    }  
}
```

```

},
"thingGroupMetadata": {
    "parentGroupName": "string",
    "rootToParentThingGroups": [
        {
            "groupName": "string",
            "groupArn": "string"
        }
    ],
    "creationDate": "timestamp"
},
"indexName": "string",
"queryString": "string",
"queryVersion": "string",
"status": "string"
}

```

### CLI output fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name of the thing group.
thingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The thing group ID.
thingGroupArn	string	The thing group ARN.
version	long	The version of the thing group.
thingGroupProperties	ThingGroupProperties	The thing group properties.
thingGroupDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The thing group description.
attributePayload	AttributePayload	The thing group attributes in JSON format.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\ "attributes\ ":" {\ \"string1\ ":" \"string2\ "}</code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.

Name	Type	Description
		To remove an attribute, call <code>UpdateThing</code> with an empty attribute value.  <b>Note</b> The <code>merge</code> attribute is only valid when calling <code>UpdateThing</code> .
thingGroupMetadata	ThingGroupMetadata	Thing group metadata.
parentGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The parent thing group name.
rootToParentThingGroups	list  member: GroupNameAndArn  java class: java.util.List	The root parent thing group.
groupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The group name.
groupArn	string	The group ARN.
creationDate	timestamp	The UNIX timestamp of when the thing group was created.
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group index name.
queryString	string  length- min:1	The dynamic thing group search query string.
queryVersion	string	The dynamic thing group query version.
status	string  enum: ACTIVE   BUILDING   REBUILDING	The dynamic thing group status.

## Errors

### InvalidRequestException

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## DescribeThingRegistrationTask

Describes a bulk thing provisioning task.

### Synopsis

```
aws iot describe-thing-registration-task \
--task-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "taskId": "string"
}
```

### cli-input-json fields

Name	Type	Description
taskId	string  length- max:40	The task ID.

### Output

```
{
  "taskId": "string",
  "creationDate": "timestamp",
  "lastModifiedDate": "timestamp",
  "templateBody": "string",
  "inputFileBucket": "string",
  "inputFileKey": "string",
  "roleArn": "string",
  "status": "string",
  "message": "string",
  "successCount": "integer",
  "failureCount": "integer",
  "percentageProgress": "integer"
}
```

### CLI output fields

Name	Type	Description
taskId	string	The task ID.

Name	Type	Description
	length- max:40	
creationDate	timestamp	The task creation date.
lastModifiedDate	timestamp	The date when the task was last modified.
templateBody	string	The task's template.
inputFileBucket	string  length- max:256 min:3  pattern: [a-zA-Z0-9._-]+	The S3 bucket that contains the input file.
inputFileKey	string  length- max:1024 min:1  pattern: [a-zA-Z0-9!_*()'/-]+	The input file key.
roleArn	string  length- max:2048 min:20	The role ARN that grants access to the input file bucket.
status	string  enum: InProgress   Completed   Failed   Cancelled   Cancelling	The status of the bulk thing provisioning task.
message	string  length- max:2048	The message.
successCount	integer	The number of things successfully provisioned.
failureCount	integer	The number of things that failed to be provisioned.
percentageProgress	integer  range- max:100 min:0	The progress of the bulk provisioning task expressed as a percentage.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## DescribeThingType

Gets information about the specified thing type.

### Synopsis

```
aws iot describe-thing-type \
  --thing-type-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingTypeName": "string"
}
```

### cli-input-json fields

Name	Type	Description
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing type.

### Output

```
{
  "thingTypeName": "string",
  "thingTypeId": "string",
  "thingTypeArn": "string",
  "thingTypeProperties": {
    "thingTypeDescription": "string",
    "searchableAttributes": [
      "string"
    ]
  },
  "thingTypeMetadata": {
    "deprecated": "boolean",
    "deprecationDate": "timestamp",
    "creationDate": "timestamp"
  }
}
```

### CLI output fields

Name	Type	Description
thingTypeName	string	The name of the thing type.

Name	Type	Description
	length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	
thingTypeId	string	The thing type ID.
thingTypeArn	string	The thing type ARN.
thingTypeProperties	ThingTypeProperties	The ThingTypeProperties contains information about the thing type including description, and a list of searchable thing attribute names.
thingTypeDescription	string length- max:2028 pattern: [\p{Graph} ]*	The description of the thing type.
searchableAttributes	list member: AttributeName java class: java.util.List	A list of searchable thing attribute names.
thingTypeMetadata	ThingTypeMetadata	The ThingTypeMetadata contains additional information about the thing type including: creation date and time, a value indicating whether the thing type is deprecated, and a date and time when it was deprecated.
deprecated	boolean	Whether the thing type is deprecated. If <b>true</b> , no new things could be associated with this type.
deprecationDate	timestamp	The date and time when the thing type was deprecated.
creationDate	timestamp	The date and time when the thing type was created.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DetachPolicy

Detaches a policy from the specified target.

### Synopsis

```
aws iot detach-policy \
--policy-name <value> \
--target <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "target": "string"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy to detach.
target	string	The target from which the policy will be detached.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## DetachPrincipalPolicy

Removes the specified policy from the specified certificate.

**Note:** This API is deprecated. Please use `DetachPolicy` instead.

### Synopsis

```
aws iot detach-principal-policy \
  --policy-name <value> \
  --principal <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
  "policyName": "string",  
  "principal": "string"  
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy to detach.
principal	string	The principal.  If the principal is a certificate, specify the certificate ARN. If the principal is an Amazon Cognito identity, specify the identity ID.

### Output

None

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# DetachSecurityProfile

Disassociates a Device Defender security profile from a thing group or from this account.

## Synopsis

```
aws iot detach-security-profile \
  --security-profile-name <value> \
  --security-profile-target-arn <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
    "securityProfileName": "string",  
    "securityProfileTargetArn": "string"  
}
```

## cli-input-json fields

Name	Type	Description
<code>securityProfileName</code>	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile that is detached.
<code>securityProfileTargetArn</code>	string	The ARN of the thing group from which the security profile is detached.

## Output

None

**Errors**

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## DetachThingPrincipal

Detaches the specified principal from the specified thing. A principal can be X.509 certificates, IAM users, groups, and roles, Amazon Cognito identities or federated identities.

**Note**

This call is asynchronous. It might take several seconds for the detachment to propagate.

**Synopsis**

```
aws iot detach-thing-principal \
  --thing-name <value> \
  --principal <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json format**

```
{
  "thingName": "string",
  "principal": "string"
}
```

**cli-input-json fields**

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.
principal	string	If the principal is a certificate, this value must be ARN of the certificate. If the principal is an Amazon Cognito identity, this value must be the ID of the Amazon Cognito identity.

**Output**

None

**Errors**

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## DisableTopicRule

Disables the rule.

**Synopsis**

```
aws iot disable-topic-rule \
--rule-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format**

```
{  
    "ruleName": "string"  
}
```

**cli-input-json fields**

Name	Type	Description
ruleName	string  length- max:128 min:1  pattern: ^[a-zA-Z0-9_]+\$	The name of the rule to disable.

**Output**

None

## Errors

`InternalException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`ServiceUnavailableException`

The service is temporarily unavailable.

`UnauthorizedException`

You are not authorized to perform this operation.

`ConflictingResourceUpdateException`

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

# EnableTopicRule

Enables the rule.

## Synopsis

```
aws iot enable-topic-rule \
  --rule-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "ruleName": "string"  
}
```

## cli-input-json fields

Name	Type	Description
ruleName	string  length- max:128 min:1  pattern: ^[a-zA-Z0-9_]+\$	The name of the topic rule to enable.

## Output

None

## Errors

`InternalException`

An unexpected error has occurred.

#### `InvalidRequestException`

The contents of the request were invalid.

#### `ServiceUnavailableException`

The service is temporarily unavailable.

#### `UnauthorizedException`

You are not authorized to perform this operation.

#### `ConflictingResourceUpdateException`

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

## GetEffectivePolicies

Gets a list of the policies that have an effect on the authorization behavior of the specified device when it connects to the AWS IoT device gateway.

### Synopsis

```
aws iot get-effective-policies \
[--principal <value>] \
[--cognito-identity-pool-id <value>] \
[--thing-name <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "principal": "string",
  "cognitoIdentityPoolId": "string",
  "thingName": "string"
}
```

### `cli-input-json` fields

Name	Type	Description
principal	string	The principal.
cognitoidentityPoolId	string	The Cognito identity pool ID.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing name.

### Output

```
{
  "effectivePolicies": [
    {
```

```

        "policyName": "string",
        "policyArn": "string",
        "policyDocument": "string"
    }
]
}

```

### CLI output fields

Name	Type	Description
effectivePolicies	list  member: EffectivePolicy  java class: java.util.List	The effective policies.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
policyDocument	string	The IAM policy document.

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## GetIndexingConfiguration

Gets the search configuration.

### Synopsis

```
aws iot get-indexing-configuration \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
}
```

**Output**

```
{
    "thingIndexingConfiguration": {
        "thingIndexingMode": "string",
        "thingConnectivityIndexingMode": "string"
    },
    "thingGroupIndexingConfiguration": {
        "thingGroupIndexingMode": "string"
    }
}
```

### CLI output fields

Name	Type	Description
thingIndexingConfiguration	ThingIndexingConfiguration	Thing indexing configuration.
thingIndexingMode	string	<p>Thing indexing mode. Valid values are:</p> <ul style="list-style-type: none"> <li>• REGISTRY – Your thing index contains registry data only.</li> <li>• REGISTRY_AND_SHADOW – Your thing index contains registry and shadow data.</li> <li>• OFF - Thing indexing is disabled.</li> </ul> <p>enum: OFF   REGISTRY   REGISTRY_AND_SHADOW</p>
thingConnectivityIndexingMode	string	<p>Thing connectivity indexing mode. Valid values are:</p> <ul style="list-style-type: none"> <li>• STATUS – Your thing index contains connectivity status. To enable thing connectivity indexing, thingIndexMode must not be set to OFF.</li> <li>• OFF - Thing connectivity status indexing is disabled.</li> </ul> <p>enum: OFF   STATUS</p>
thingGroupIndexingConfiguration	ThingGroupIndexingConfiguration	The index configuration.

Name	Type	Description
thingGroupIndexingMode	string	Thing group indexing mode. enum: OFF   ON

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## GetJobDocument

Gets a job document.

### Synopsis

```
aws iot get-job-document \
--job-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string"
}
```

### cli-input-json fields

Name	Type	Description
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.

### Output

```
{
  "document": "string"
```

}

### CLI output fields

Name	Type	Description
document	string length- max:32768	The job document content.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

## GetLoggingOptions

Gets the logging options.

NOTE: use of this command is not recommended. Use `GetV2LoggingOptions` instead.

### Synopsis

```
aws iot get-logging-options \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

### Output

```
{  
  "roleArn": "string",  
  "logLevel": "string"  
}
```

### CLI output fields

Name	Type	Description
roleArn	string	The ARN of the IAM role that grants access.

Name	Type	Description
logLevel	string	The logging level. enum: DEBUG   INFO   ERROR   WARN   DISABLED

### Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

## GetOTAUpdate

Gets an OTA update.

### Synopsis

```
aws iot get-ota-update \
--ota-update-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "otaUpdateId": "string"
}
```

### cli-input-json fields

Name	Type	Description
otaUpdateId	string	The OTA update ID.  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+

### Output

```
{
  "otaUpdateInfo": {
    "otaUpdateId": "string",
    "otaUpdateArn": "string",
    "creationDate": "timestamp",
```

```

    "lastModifiedDate": "timestamp",
    "description": "string",
    "targets": [
        "string"
    ],
    "awsJobExecutionsRolloutConfig": {
        "maximumPerMinute": "integer"
    },
    "targetSelection": "string",
    "otaUpdateFiles": [
        {
            "fileName": "string",
            "fileVersion": "string",
            "fileLocation": {
                "stream": {
                    "streamId": "string",
                    "fileId": "integer"
                },
                "s3Location": {
                    "bucket": "string",
                    "key": "string",
                    "version": "string"
                }
            },
            "codeSigning": {
                "awsSignerJobId": "string",
                "startSigningJobParameter": {
                    "signingProfileParameter": {
                        "certificateArn": "string",
                        "platform": "string",
                        "certificatePathOnDevice": "string"
                    },
                    "signingProfileName": "string",
                    "destination": {
                        "s3Destination": {
                            "bucket": "string",
                            "prefix": "string"
                        }
                    }
                },
                "customCodeSigning": {
                    "signature": {
                        "inlineDocument": "blob"
                    },
                    "certificateChain": {
                        "certificateName": "string",
                        "inlineDocument": "string"
                    },
                    "hashAlgorithm": "string",
                    "signatureAlgorithm": "string"
                }
            },
            "attributes": {
                "string": "string"
            }
        }
    ],
    "otaUpdateStatus": "string",
    "awsIotJobId": "string",
    "awsIotJobArn": "string",
    "errorInfo": {
        "code": "string",
        "message": "string"
    },
    "additionalParameters": {
        "string": "string"
    }
}

```

```
    }
}
```

## CLI output fields

Name	Type	Description
otaUpdateInfo	OTAUpdateInfo	The OTA update info.
otaUpdateId	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The OTA update ID.
otaUpdateArn	string	The OTA update ARN.
creationDate	timestamp	The date when the OTA update was created.
lastModifiedDate	timestamp	The date when the OTA update was last updated.
description	string length- max:2028 pattern: [^\p{C}]+	A description of the OTA update.
targets	list member: Target	The targets of the OTA update.
awsJobExecutionsRolloutConfig	AwsJobExecutionsRolloutConfig	Configuration for the rollout of OTA updates.
maximumPerMinute	integer range- max:1000 min:1	The maximum number of OTA update job executions started per minute.
targetSelection	string	Specifies whether the OTA update will continue to run (CONTINUOUS), or will be complete after all those things specified as targets have completed the OTA update (SNAPSHOT). If continuous, the OTA update may also be run on a thing when a change is detected in a target. For example, an OTA update will run on a thing when the thing is added to a target group, even after the OTA update was completed by all things originally in the group.  enum: CONTINUOUS   SNAPSHOT

Name	Type	Description
otaUpdateFiles	list member: OTAUpdateFile	A list of files associated with the OTA update.
fileName	string	The name of the file.
fileVersion	string	The file version.
fileLocation	FileLocation	The location of the updated firmware.
stream	Stream	The stream that contains the OTA update.
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
fileId	integer  range- max:255 min:0	The ID of a file associated with a stream.
s3Location	S3Location	The location of the updated firmware in S3.
bucket	string  length- min:1	The S3 bucket.
key	string  length- min:1	The S3 key.
version	string	The S3 bucket version.
codeSigning	CodeSigning	The code signing method of the file.
awsSignerJobId	string	The ID of the AWSSignerJob which was created to sign the file.
startSigningJobParameter	StartSigningJobParameter	Describes the code-signing job.
signingProfileParameter	SigningProfileParameter	Describes the code-signing profile.
certificateArn	string	Certificate ARN.
platform	string	The hardware platform of your device.
certificatePathOnDevice	string	The location of the code-signing certificate on your device.
signingProfileName	string	The code-signing profile name.

Name	Type	Description
destination	Destination	The location to write the code-signed file.
s3Destination	S3Destination	Describes the location in S3 of the updated firmware.
bucket	string length- min:1	The S3 bucket that contains the updated firmware.
prefix	string	The S3 prefix.
customCodeSigning	CustomCodeSigning	A custom method for code signing a file.
signature	CodeSigningSignature	The signature for the file.
inlineDocument	blob	A base64 encoded binary representation of the code signing signature.
certificateChain	CodeSigningCertificateChain	The certificate chain.
certificateName	string	The name of the certificate.
inlineDocument	string	A base64 encoded binary representation of the code signing certificate chain.
hashAlgorithm	string	The hash algorithm used to code sign the file.
signatureAlgorithm	string	The signature algorithm used to code sign the file.
attributes	map	A list of name/attribute pairs.
otaUpdateStatus	string	The status of the OTA update.  enum: CREATE_PENDING   CREATE_IN_PROGRESS   CREATE_COMPLETE   CREATE_FAILED
awsIoTJobId	string	The AWS IoT job ID associated with the OTA update.
awsIoTJobArn	string	The AWS IoT job ARN associated with the OTA update.
errorInfo	ErrorInfo	Error information associated with the OTA update.
code	string	The error code.
message	string	The error message.
additionalParameters	map	A collection of name/value pairs

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`InternalFailureException`

An unexpected error has occurred.

`ServiceUnavailableException`

The service is temporarily unavailable.

`ResourceNotFoundException`

The specified resource does not exist.

# GetPendingJobExecutions

Gets the list of all jobs for a thing that are not in a terminal status.

## Synopsis

```
aws iot-jobs-data get-pending-job-executions \
  --thing-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingName": "string"  
}
```

## cli-input-json fields

Name	Type	Description
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing that is executing the job.

## Output

```
{  
    "inProgressJobs": [  
        {  
            "jobId": "string",
```

```

        "queuedAt": "long",
        "startedAt": "long",
        "lastUpdatedAt": "long",
        "versionNumber": "long",
        "executionNumber": "long"
    }
],
"queuedJobs": [
{
    "jobId": "string",
    "queuedAt": "long",
    "startedAt": "long",
    "lastUpdatedAt": "long",
    "versionNumber": "long",
    "executionNumber": "long"
}
]
}

```

### CLI output fields

Name	Type	Description
inProgressJobs	list member: JobExecutionSummary java class: java.util.List	A list of JobExecutionSummary objects with status IN_PROGRESS.
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time AWS IoT Jobs receives an update from a device.
executionNumber	long	A number that identifies a particular job execution on a particular device.
queuedJobs	list member: JobExecutionSummary java class: java.util.List	A list of JobExecutionSummary objects with status QUEUED.

Name	Type	Description
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time AWS IoT Jobs receives an update from a device.
executionNumber	long	A number that identifies a particular job execution on a particular device.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

`CertificateValidationException`

The certificate is invalid.

## GetPolicy

Gets information about the specified policy with the policy document of the default version.

### Synopsis

```
aws iot get-policy \
--policy-name <value> \
```

```
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.

### Output

```
{
  "policyName": "string",
  "policyArn": "string",
  "policyDocument": "string",
  "defaultVersionId": "string",
  "creationDate": "timestamp",
  "lastModifiedDate": "timestamp",
  "generationId": "string"
}
```

### CLI output fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
policyDocument	string	The JSON document that describes the policy.
defaultVersionId	string  pattern: [0-9]+	The default policy version ID.
creationDate	timestamp	The date the policy was created.
lastModifiedDate	timestamp	The date the policy was last modified.
generationId	string	The generation ID of the policy.

### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## GetPolicyVersion

Gets information about the specified policy version.

### Synopsis

```
aws iot get-policy-version \
--policy-name <value> \
--policy-version-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "policyVersionId": "string"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The policy version ID.

### Output

```
{
  "policyArn": "string",
```

```

    "policyName": "string",
    "policyDocument": "string",
    "policyVersionId": "string",
    "isDefaultVersion": "boolean",
    "creationDate": "timestamp",
    "lastModifiedDate": "timestamp",
    "generationId": "string"
}

```

### CLI output fields

Name	Type	Description
policyArn	string	The policy ARN.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyDocument	string	The JSON document that describes the policy.
policyVersionId	string  pattern: [0-9]+	The policy version ID.
isDefaultVersion	boolean	Specifies whether the policy version is the default.
creationDate	timestamp	The date the policy version was created.
lastModifiedDate	timestamp	The date the policy version was last modified.
generationId	string	The generation ID of the policy version.

### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## GetRegistrationCode

Gets a registration code used to register a CA certificate with AWS IoT.

### Synopsis

```
aws iot get-registration-code \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{  
}
```

### Output

```
{  
    "registrationCode": "string"  
}
```

### CLI output fields

Name	Type	Description
registrationCode	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The CA certificate registration code.

### Errors

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

## GetStatistics

Gets statistics about things that match the specified query.

### Synopsis

```
aws iot get-statistics \
[--index-name <value>] \
--query-string <value> \
[--aggregation-field <value>] \
[--query-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "indexName": "string",
  "queryString": "string",
  "aggregationField": "string",
  "queryVersion": "string"
}
```

#### cli-input-json fields

Name	Type	Description
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the index to search. The default value is AWS_Things.
queryString	string  length- min:1	The query used to search. You can specify "*" for the query string to get the count of all indexed things in your AWS account.
aggregationField	string  length- min:1	The aggregation field name. Currently not supported.
queryVersion	string	The version of the query used to search.

#### Output

```
{
  "statistics": {
    "count": "integer"
  }
}
```

#### CLI output fields

Name	Type	Description
statistics	Statistics	The statistics returned by the Fleet Indexing service based on the query and aggregation field.
count	integer	The count of things that match the query.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidQueryException`

The query is invalid.

`InvalidAggregationException`

The aggregation is invalid.

`IndexNotReadyException`

The index is not ready.

# GetThingShadow

Gets the shadow for the specified thing.

For more information, see [GetThingShadow](#) in the AWS IoT Developer Guide.

## Synopsis

```
aws iot-data get-thing-shadow \
  --thing-name <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "thingName": "string"  
}
```

## cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.

## Output

```
{  
    "payload": "blob"  
}
```

## CLI output fields

Name	Type	Description
payload	blob	The state information, in JSON format.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`MethodNotAllowedException`

The specified combination of HTTP verb and URI is not supported.

`UnsupportedDocumentEncodingException`

The encoding is not supported.

# GetTopicRule

Gets information about the rule.

## Synopsis

```
aws iot get-topic-rule \  
    --rule-name <value> \  
    [--cli-input-json <value>] \  
    [--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
    "ruleName": "string"
```

}

### cli-input-json fields

Name	Type	Description
ruleName	string length- max:128 min:1 pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.

### Output

```
{
  "ruleArn": "string",
  "rule": {
    "ruleName": "string",
    "sql": "string",
    "description": "string",
    "createdAt": "timestamp",
    "actions": [
      {
        "dynamoDB": {
          "tableName": "string",
          "roleArn": "string",
          "operation": "string",
          "hashKeyField": "string",
          "hashKeyValue": "string",
          "hashKeyType": "string",
          "rangeKeyField": "string",
          "rangeKeyValue": "string",
          "rangeKeyType": "string",
          "payloadField": "string"
        },
        "dynamoDBv2": {
          "roleArn": "string",
          "putItem": {
            "tableName": "string"
          }
        },
        "lambda": {
          "functionArn": "string"
        },
        "sns": {
          "targetArn": "string",
          "roleArn": "string",
          "messageFormat": "string"
        },
        "sqs": {
          "roleArn": "string",
          "queueUrl": "string",
          "useBase64": "boolean"
        },
        "kinesis": {
          "roleArn": "string",
          "streamName": "string",
          "partitionKey": "string"
        },
        "republish": {
          "roleArn": "string",
          "topic": "string"
        }
      }
    ]
  }
}
```

```

"s3": {
    "roleArn": "string",
    "bucketName": "string",
    "key": "string",
    "cannedAcl": "string"
},
"firehose": {
    "roleArn": "string",
    "deliveryStreamName": "string",
    "separator": "string"
},
"cloudwatchMetric": {
    "roleArn": "string",
    "metricNamespace": "string",
    "metricName": "string",
    "metricValue": "string",
    "metricUnit": "string",
    "metricTimestamp": "string"
},
"cloudwatchAlarm": {
    "roleArn": "string",
    "alarmName": "string",
    "stateReason": "string",
    "stateValue": "string"
},
"elasticsearch": {
    "roleArn": "string",
    "endpoint": "string",
    "index": "string",
    "type": "string",
    "id": "string"
},
"salesforce": {
    "token": "string",
    "url": "string"
},
"iotAnalytics": {
    "channelArn": "string",
    "channelName": "string",
    "roleArn": "string"
},
"iotEvents": {
    "inputName": "string",
    "messageId": "string",
    "roleArn": "string"
},
"stepFunctions": {
    "executionNamePrefix": "string",
    "stateMachineName": "string",
    "roleArn": "string"
}
},
],
"ruleDisabled": "boolean",
"awsIotSqlVersion": "string",
"errorAction": {
    "dynamoDB": {
        "tableName": "string",
        "roleArn": "string",
        "operation": "string",
        "hashKeyField": "string",
        "hashKeyValue": "string",
        "hashKeyType": "string",
        "rangeKeyField": "string",
        "rangeKeyValue": "string",
        "rangeKeyType": "string",
    }
}
]
}

```

```

        "payloadField": "string"
    },
    "dynamoDBv2": {
        "roleArn": "string",
        "putItem": {
            "tableName": "string"
        }
    },
    "lambda": {
        "functionArn": "string"
    },
    "sns": {
        "targetArn": "string",
        "roleArn": "string",
        "messageFormat": "string"
    },
    "sqs": {
        "roleArn": "string",
        "queueUrl": "string",
        "useBase64": "boolean"
    },
    "kinesis": {
        "roleArn": "string",
        "streamName": "string",
        "partitionKey": "string"
    },
    "republish": {
        "roleArn": "string",
        "topic": "string"
    },
    "s3": {
        "roleArn": "string",
        "bucketName": "string",
        "key": "string",
        "cannedAcl": "string"
    },
    "firehose": {
        "roleArn": "string",
        "deliveryStreamName": "string",
        "separator": "string"
    },
    "cloudwatchMetric": {
        "roleArn": "string",
        "metricNamespace": "string",
        "metricName": "string",
        "metricValue": "string",
        "metricUnit": "string",
        "metricTimestamp": "string"
    },
    "cloudwatchAlarm": {
        "roleArn": "string",
        "alarmName": "string",
        "stateReason": "string",
        "stateValue": "string"
    },
    "elasticsearch": {
        "roleArn": "string",
        "endpoint": "string",
        "index": "string",
        "type": "string",
        "id": "string"
    },
    "salesforce": {
        "token": "string",
        "url": "string"
    }
},

```

```
        "iotAnalytics": {
            "channelArn": "string",
            "channelName": "string",
            "roleArn": "string"
        },
        "iotEvents": {
            "inputName": "string",
            "messageId": "string",
            "roleArn": "string"
        },
        "stepFunctions": {
            "executionNamePrefix": "string",
            "stateMachineName": "string",
            "roleArn": "string"
        }
    }
}
```

## CLI output fields

Name	Type	Description
ruleArn	string	The rule ARN.
rule	TopicRule	The rule.
ruleName	string length- max:128 min:1 pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.
sql	string	The SQL statement used to query the topic. When using a SQL query with multiple lines, be sure to escape the newline characters.
description	string	The description of the rule.
createdAt	timestamp	The date and time the rule was created.
actions	list member: Action	The actions associated with the rule.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
operation	string	The type of operation to be performed. This follows the substitution template, so it can be <code>\\$ operation</code> , but the substitution must result in

Name	Type	Description
		one of the following: INSERT, UPDATE, or DELETE.
hashKeyField	string	The hash key name.
hashKeyValue	string	The hash key value.
hashKeyType	string	The hash key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
putItem	PutItemInput	<p>Specifies the DynamoDB table to which the message data will be written. For example:</p> <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> <p>Each attribute in the message payload will be written to a separate column in the DynamoDB database.</p>
tableName	string	The table where the message data will be written.
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.

Name	Type	Description
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.
roleArn	string	The ARN of the IAM role that grants access.
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.

Name	Type	Description
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.
key	string	The object key.
cannedAcl	string	<p>The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a>.</p> <p>enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write</p>
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.
separator	string pattern: ([ ] (  ) (),)	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.

Name	Type	Description
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.((sfdc-matrix.net) (sfcdnnow.com))/streams/w <i>1,20</i> /w <i>1,20</i> /event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	IoTAnalyticsAction	Sends message data to an AWS IoT Analytics channel.
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.

Name	Type	Description
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").
ruleDisabled	boolean	Specifies whether the rule is disabled.
awsIoTSqlVersion	string	The version of the SQL rules engine to use when evaluating the rule.
errorAction	Action	The action to perform when an error occurs.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.

Name	Type	Description
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
operation	string	The type of operation to be performed. This follows the substitution template, so it can be \$ <i>operation</i> , but the substitution must result in one of the following: INSERT, UPDATE, or DELETE.
hashKeyField	string	The hash key name.
hashKeyValue	string	The hash key value.
hashKeyType	string	The hash key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.

Name	Type	Description
putItem	PutItemInput	<p>Specifies the DynamoDB table to which the message data will be written. For example:</p> <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> <p>Each attribute in the message payload will be written to a separate column in the DynamoDB database.</p>
tableName	string	The table where the message data will be written.
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.
roleArn	string	The ARN of the IAM role that grants access.
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.

Name	Type	Description
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.
key	string	The object key.
cannedAcl	string	The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a> .  enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.

Name	Type	Description
separator	string pattern: ([ ] ( ) ( .) ( ,))	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.

Name	Type	Description
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.(sfdc-matrix.net) (sfdcnow.com))/streams/w <i>1, 20</i> /w <i>1, 20</i> /event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	IoTAnalyticsAction	Sends message data to an AWS IoT Analytics channel.
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.

Name	Type	Description
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").

### Errors

`InternalException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`ServiceUnavailableException`

The service is temporarily unavailable.

`UnauthorizedException`

You are not authorized to perform this operation.

## GetV2LoggingOptions

Gets the fine grained logging options.

### Synopsis

```
aws iot get-v2-logging-options \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
}
```

### Output

```
{
```

```

    "roleArn": "string",
    "defaultLogLevel": "string",
    "disableAllLogs": "boolean"
}

```

### CLI output fields

Name	Type	Description
roleArn	string	The IAM role ARN AWS IoT uses to write to your CloudWatch logs.
defaultLogLevel	string	The default log level.  enum: DEBUG   INFO   ERROR   WARN   DISABLED
disableAllLogs	boolean	Disables all logs.

### Errors

`InternalException`

An unexpected error has occurred.

`NotConfiguredException`

The resource is not configured.

`ServiceUnavailableException`

The service is temporarily unavailable.

## ListActiveViolations

Lists the active violations for a given Device Defender security profile.

### Synopsis

```

aws iot list-active-violations \
[--thing-name <value>] \
[--security-profile-name <value>] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]

```

`cli-input-json` format

```

{
    "thingName": "string",
    "securityProfileName": "string",
    "nextToken": "string",
    "maxResults": "integer"
}

```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing whose active violations are listed.
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the Device Defender security profile for which violations are listed.
nextToken	string	The token for the next set of results.
maxResults	integer  range- max:250 min:1	The maximum number of results to return at one time.

### Output

```
{
  "activeViolations": [
    {
      "violationId": "string",
      "thingName": "string",
      "securityProfileName": "string",
      "behavior": {
        "name": "string",
        "metric": "string",
        "criteria": {
          "comparisonOperator": "string",
          "value": {
            "count": "long",
            "cidrs": [
              "string"
            ],
            "ports": [
              "integer"
            ]
          },
          "durationSeconds": "integer",
          "consecutiveDatapointsToAlarm": "integer",
          "consecutiveDatapointsToClear": "integer",
          "statisticalThreshold": {
            "statistic": "string"
          }
        }
      },
      "lastViolationValue": {
        "count": "long",
        "cidrs": [
          "string"
        ],
        "ports": [
          "integer"
        ]
      },
      "lastViolationTime": "timestamp",
      "lastViolationValueTime": "timestamp"
    }
  ]
}
```

```

        "violationStartTime": "timestamp"
    }
],
"nextToken": "string"
}

```

### CLI output fields

Name	Type	Description
activeViolations	list  member: ActiveViolation	The list of active violations.
violationId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-]+	The ID of the active violation.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing responsible for the active violation.
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The security profile whose behavior is in violation.
behavior	Behavior	The behavior which is being violated.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.

Name	Type	Description
count	long range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cidrs	list member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, <code>NUM_MESSAGES_SENT</code> ). For a <code>statisticalThreshold</code> metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
lastViolationValue	MetricValue	The value of the metric (the measurement) which caused the most recent violation.
count	long  range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cids	list  member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list  member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
lastViolationTime	timestamp	The time the most recent violation occurred.
violationStartTime	timestamp	The time the violation started.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## ListAttachedPolicies

Lists the policies attached to the specified thing group.

### Synopsis

```
aws iot list-attached-policies \
--target <value> \
[--recursive | --no-recursive] \
[--marker <value>] \
[--page-size <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "target": "string",
  "recursive": "boolean",
  "marker": "string",
  "pageSize": "integer"
}
```

### cli-input-json fields

Name	Type	Description
target	string	The group for which the policies will be listed.
recursive	boolean	When true, recursively list attached policies.
marker	string pattern: [A-Za-z0-9+/]+={0,2}	The token to retrieve the next set of results.
pageSize	integer range- max:250 min:1	The maximum number of results to be returned per request.

### Output

```
{
  "policies": [
    {
      "policyName": "string",
      "policyArn": "string"
```

```

        },
        "nextMarker": "string"
    }
}

```

### CLI output fields

Name	Type	Description
policies	list  member: Policy  java class: java.util.List	The policies.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
nextMarker	string  pattern: [A-Za-z0-9+/]+={0,2}	The token to retrieve the next set of results, or null if there are no more results.

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## ListAuditFindings

Lists the findings (results) of a Device Defender audit or of the audits performed during a specified time period. (Findings are retained for 180 days.)

### Synopsis

```
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]
```

#### cli-input-json format

```
{
  "taskId": "string",
  "checkName": "string",
  "resourceIdentifier": {
    "deviceCertificateId": "string",
    "caCertificateId": "string",
    "cognitoIdentityPoolId": "string",
    "clientId": "string",
    "policyVersionIdentifier": {
      "policyName": "string",
      "policyVersionId": "string"
    },
    "account": "string"
  },
  "maxResults": "integer",
  "nextToken": "string",
  "startTime": "timestamp",
  "endTime": "timestamp"
}
```

#### cli-input-json fields

Name	Type	Description
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9]+	A filter to limit results to the audit with the specified ID. You must specify either the taskId or the startTime and endTime, but not both.
checkName	string	A filter to limit results to the findings for the specified audit check.
resourceIdentifier	ResourceIdentifier	Information identifying the non-compliant resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.

Name	Type	Description
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
maxResults	integer  range- max:250 min:1	The maximum number of results to return at one time. The default is 25.
nextToken	string	The token for the next set of results.
startTime	timestamp	A filter to limit results to those found after the specified time. You must specify either the startTime and endTime or the taskId, but not both.
endTime	timestamp	A filter to limit results to those found before the specified time. You must specify either the startTime and endTime or the taskId, but not both.

## Output

```
{
  "findings": [
    {
      "taskId": "string",
      "checkName": "string",
      "taskStartTime": "timestamp",
      "findingTime": "timestamp",
      "severity": "string",
      "nonCompliantResource": {
        "resourceType": "string",
        "resourceIdentifier": {
          "deviceCertificateId": "string",
          "caCertificateId": "string",
          "resourceArn": "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",
        "policyVersionIdentifier": {
            "policyName": "string",
            "policyVersionId": "string"
        },
        "account": "string"
    },
    "additionalInfo": {
        "string": "string"
    }
}
],
"reasonForNonCompliance": "string",
"reasonForNonComplianceCode": "string"
}
],
"nextToken": "string"
}
}

```

### CLI output fields

Name	Type	Description
findings	list member: AuditFinding	The findings (results) of the audit.
taskId	string length- max:40 min:1 pattern: [a-zA-Z0-9-]+	The ID of the audit that generated this result (finding)
checkName	string	The audit check that generated this result.
taskStartTime	timestamp	The time the audit started.
findingTime	timestamp	The time the result (finding) was discovered.
severity	string	The severity of the result (finding).

Name	Type	Description
		enum: CRITICAL   HIGH   MEDIUM   LOW
nonCompliantResource	NonCompliantResource	The resource that was found to be non-compliant with the audit check.
resourceType	string	The type of the non-compliant resource.  enum: DEVICE_CERTIFICATE   CA_CERTIFICATE   IOT_POLICY   COGNITO_IDENTITY_POOL   CLIENT_ID   ACCOUNT_SETTINGS
resourceIdentifier	ResourceIdentifier	Information identifying the non-compliant resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
additionalInfo	map	Additional information about the non-compliant resource.

Name	Type	Description
relatedResources	list  member: RelatedResource	The list of related resources.
resourceType	string  enum: DEVICE_CERTIFICATE   CA_CERTIFICATE   IOT_POLICY   COGNITO_IDENTITY_POOL   CLIENT_ID   ACCOUNT_SETTINGS	The type of resource.
resourceIdentifier	ResourceIdentifier	Information identifying the resource.
deviceCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate attached to the resource.
caCertificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate used to authorize the certificate.
cognitoIdentityPoolId	string	The ID of the Cognito Identity Pool.
clientId	string	The client ID.
policyVersionIdentifier	PolicyVersionIdentifier	The version of the policy associated with the resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The name of the policy.
policyVersionId	string  pattern: [0-9]+	The ID of the version of the policy associated with the resource.
account	string  length- max:12 min:12  pattern: [0-9]+	The account with which the resource is associated.
additionalInfo	map	Additional information about the resource.
reasonForNonCompliance	string	The reason the resource was non-compliant.

Name	Type	Description
reasonForNonComplianceCode	string	A code which indicates the reason that the resource was non-compliant.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## ListAuditTasks

Lists the Device Defender audits that have been performed during a given time period.

### Synopsis

```
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]
```

`cli-input-json` format

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "taskType": "string",
  "taskStatus": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
startTime	timestamp	The beginning of the time period. Note that audit

Name	Type	Description
		information is retained for a limited time (180 days). Requesting a start time prior to what is retained results in an "InvalidRequestException".
endTime	timestamp	The end of the time period.
taskType	string	A filter to limit the output to the specified type of audit: can be one of "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK
taskStatus	string	A filter to limit the output to audits with the specified completion status: can be one of "IN_PROGRESS", "COMPLETED", "FAILED" or "CANCELED".  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time. The default is 25.

## Output

```
{
  "tasks": [
    {
      "taskId": "string",
      "taskStatus": "string",
      "taskType": "string"
    }
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
tasks	list member: AuditTaskMetadata java class: java.util.List	The audits that were performed during the specified time period.

Name	Type	Description
taskId	string  length- max:40 min:1  pattern: [a-zA-Z0-9-]+	The ID of this audit.
taskStatus	string	The status of this audit: one of "IN_PROGRESS", "COMPLETED", "FAILED" or "CANCELED".  enum: IN_PROGRESS   COMPLETED   FAILED   CANCELED
taskType	string	The type of this audit: one of "ON_DEMAND_AUDIT_TASK" or "SCHEDULED_AUDIT_TASK".  enum: ON_DEMAND_AUDIT_TASK   SCHEDULED_AUDIT_TASK
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## ListAuthorizers

Lists the authorizers registered in your account.

### Synopsis

```
aws iot list-authorizers \
[--page-size <value>] \
[--marker <value>] \
[--ascending-order | --no-ascending-order] \
[--status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{
  "pageSize": "integer",
  "marker": "string",
  "ascendingOrder": "boolean",
  "status": "string"
}
```

#### cli-input-json fields

Name	Type	Description
pageSize	integer range- max:250 min:1	The maximum number of results to return at one time.
marker	string pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.
ascendingOrder	boolean	Return the list of authorizers in ascending alphabetical order.
status	string	The status of the list authorizers request. enum: ACTIVE   INACTIVE

#### Output

```
{
  "authorizers": [
    {
      "authorizerName": "string",
      "authorizerArn": "string"
    }
  ],
  "nextMarker": "string"
}
```

#### CLI output fields

Name	Type	Description
authorizers	list member: AuthorizerSummary java class: java.util.List	The authorizers.
authorizerName	string length- max:128 min:1 pattern: [w=,@-]+	The authorizer name.
authorizerArn	string	The authorizer ARN.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# ListBillingGroups

Lists the billing groups you have created.

## Synopsis

```
aws iot list-billing-groups \
[--next-token <value>] \
[--max-results <value>] \
[--name-prefix-filter <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "namePrefixFilter": "string"
}
```

## cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return per request.
namePrefixFilter	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	Limit the results to billing groups whose names have the given prefix.

## Output

```
{
  "billingGroups": [
    {
      "groupName": "string",
      "groupArn": "string"
    }
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
billingGroups	list  member: GroupNameAndArn  java class: java.util.List	The list of billing groups.
groupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The group name.
groupArn	string	The group ARN.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

# ListCACertificates

Lists the CA certificates registered for your AWS account.

The results are paginated with a default page size of 25. You can use the returned marker to retrieve additional results.

## Synopsis

```
aws iot list-ca-certificates \
[--page-size <value>] \
[--marker <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "pageSize": "integer",
  "marker": "string",
  "ascendingOrder": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
pageSize	integer range- max:250 min:1	The result page size.
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
ascendingOrder	boolean	Determines the order of the results.

## Output

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "status": "string",
      "creationDate": "timestamp"
    }
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
certificates	list member: CACertificate java class: java.util.List	The CA certificates registered in your AWS account.
certificateArn	string	The ARN of the CA certificate.

Name	Type	Description
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate.
status	string	The status of the CA certificate.  The status value REGISTER_INACTIVE is deprecated and should not be used.  enum: ACTIVE   INACTIVE
creationDate	timestamp	The date the CA certificate was created.
nextMarker	string pattern: [A-Za-z0-9+/]+={0,2}	The current position within the list of CA certificates.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListCertificates

Lists the certificates registered in your AWS account.

The results are paginated with a default page size of 25. You can use the returned marker to retrieve additional results.

### Synopsis

```
aws iot list-certificates \
[--page-size <value>] \
[--marker <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>]
```

```
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "pageSize": "integer",
  "marker": "string",
  "ascendingOrder": "boolean"
}
```

#### cli-input-json fields

Name	Type	Description
pageSize	integer range- max:250 min:1	The result page size.
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
ascendingOrder	boolean	Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

#### Output

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "status": "string",
      "creationDate": "timestamp"
    }
  ],
  "nextMarker": "string"
}
```

#### CLI output fields

Name	Type	Description
certificates	list member: Certificate java class: java.util.List	The descriptions of the certificates.
certificateArn	string	The ARN of the certificate.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)

Name	Type	Description
status	string	The status of the certificate.  The status value REGISTER_INACTIVE is deprecated and should not be used.  enum: ACTIVE   INACTIVE   REVOKED   PENDING_TRANSFER   REGISTER_INACTIVE   PENDING_ACTIVATION
creationDate	timestamp	The date and time the certificate was created.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results, or null if there are no additional results.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

# ListCertificatesByCA

List the device certificates signed by the specified CA certificate.

## Synopsis

```
aws iot list-certificates-by-ca \
    --ca-certificate-id <value> \
    [--page-size <value>] \
    [--marker <value>] \
    [--ascending-order | --no-ascending-order] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
    "caCertificateId": "string",
```

```

    "pageSize": "integer",
    "marker": "string",
    "ascendingOrder": "boolean"
}

```

#### cli-input-json fields

Name	Type	Description
caCertificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the CA certificate. This operation will list all registered device certificate that were signed by this CA certificate.
pageSize	integer range- max:250 min:1	The result page size.
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
ascendingOrder	boolean	Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

#### Output

```
{
  "certificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "status": "string",
      "creationDate": "timestamp"
    }
  ],
  "nextMarker": "string"
}
```

#### CLI output fields

Name	Type	Description
certificates	list member: Certificate java class: java.util.List	The device certificates signed by the specified CA certificate.
certificateArn	string	The ARN of the certificate.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)
status	string	The status of the certificate.

Name	Type	Description
		The status value REGISTER_INACTIVE is deprecated and should not be used.  enum: ACTIVE   INACTIVE   REVOKED   PENDING_TRANSFER   REGISTER_INACTIVE   PENDING_ACTIVATION
creationDate	timestamp	The date and time the certificate was created.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results, or null if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListIndices

Lists the search indices.

### Synopsis

```
aws iot list-indices \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
nextToken	string	The token used to get the next set of results, or null if there are no additional results.
maxResults	integer range- max:500 min:1	The maximum number of results to return at one time.

### Output

```
{
  "indexNames": [
    "string"
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
indexNames	list member: IndexName java class: java.util.List	The index names.
nextToken	string	The token used to get the next set of results, or null if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListJobExecutionsForJob

Lists the job executions for a job.

## Synopsis

```
aws iot list-job-executions-for-job \
    --job-id <value> \
    [--status <value>] \
    [--max-results <value>] \
    [--next-token <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "jobId": "string",
  "status": "string",
  "maxResults": "integer",
  "nextToken": "string"
}
```

## cli-input-json fields

Name	Type	Description
jobId	string length- max:64 min:1 pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
status	string	The status of the job.  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
maxResults	integer range- max:250 min:1	The maximum number of results to be returned per request.
nextToken	string	The token to retrieve the next set of results.

## Output

```
{
  "executionSummaries": [
    {
      "thingArn": "string",
      "jobExecutionSummary": {
        "status": "string",
        "queuedAt": "timestamp",
        "startedAt": "timestamp",
        "lastUpdatedAt": "timestamp",
        "executionNumber": "long"
      }
    }
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
executionSummaries	list  member: JobExecutionSummaryForJob  java class: java.util.List	A list of job execution summaries.
thingArn	string	The ARN of the thing on which the job execution is running.
jobExecutionSummary	JobExecutionSummary	Contains a subset of information about a job execution.
status	string	The status of the job execution.  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long	A string (consisting of the digits "0" through "9") which identifies this particular job execution on this particular device. It can be used later in commands which return or update job execution information.
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### ServiceUnavailableException

The service is temporarily unavailable.

## ListJobExecutionsForThing

Lists the job executions for the specified thing.

### Synopsis

```
aws iot list-job-executions-for-thing \
--thing-name <value> \
[--status <value>] \
[--max-results <value>] \
[--next-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string",
  "status": "string",
  "maxResults": "integer",
  "nextToken": "string"
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing name.
status	string  An optional filter that lets you search for jobs that have the specified status.  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	An optional filter that lets you search for jobs that have the specified status.  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
maxResults	integer  range- max:250 min:1	The maximum number of results to be returned per request.
nextToken	string	The token to retrieve the next set of results.

### Output

```
{
```

```

"executionSummaries": [
  {
    "jobId": "string",
    "jobExecutionSummary": {
      "status": "string",
      "queuedAt": "timestamp",
      "startedAt": "timestamp",
      "lastUpdatedAt": "timestamp",
      "executionNumber": "long"
    }
  }
],
"nextToken": "string"
}

```

### CLI output fields

Name	Type	Description
executionSummaries	list  member: JobExecutionSummaryForThing  java class: java.util.List	A list of job execution summaries.
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
jobExecutionSummary	JobExecutionSummary	Contains a subset of information about a job execution.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution.
queuedAt	timestamp	The time, in seconds since the epoch, when the job execution was queued.
startedAt	timestamp	The time, in seconds since the epoch, when the job execution started.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job execution was last updated.
executionNumber	long	A string (consisting of the digits "0" through "9") which identifies this particular job execution on this particular device. It can be used later in commands which return or update job execution information.

Name	Type	Description
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**ServiceUnavailableException**

The service is temporarily unavailable.

## ListJobs

Lists jobs.

### Synopsis

```
aws iot list-jobs \
[--status <value>] \
[--target-selection <value>] \
[--max-results <value>] \
[--next-token <value>] \
[--thing-group-name <value>] \
[--thing-group-id <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "status": "string",
  "targetSelection": "string",
  "maxResults": "integer",
  "nextToken": "string",
  "thingGroupName": "string",
  "thingGroupId": "string"
}
```

### cli-input-json fields

Name	Type	Description
status	string	An optional filter that lets you search for jobs that have the specified status.

Name	Type	Description
		enum: IN_PROGRESS   CANCELED   COMPLETED   DELETION_IN_PROGRESS
targetSelection	string	Specifies whether the job will continue to run (CONTINUOUS), or will be complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous, the job may also be run on a thing when a change is detected in a target. For example, a job will run on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.  enum: CONTINUOUS   SNAPSHOT
maxResults	integer range- max:250 min:1	The maximum number of results to return per request.
nextToken	string	The token to retrieve the next set of results.
thingGroupName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter that limits the returned jobs to those for the specified group.
thingGroupId	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter that limits the returned jobs to those for the specified group.

## Output

```
{
  "jobs": [
    {
      "jobArn": "string",
      "jobId": "string",
      "thingGroupId": "string",
      "targetSelection": "string",
      "status": "string",
      "createdAt": "timestamp",
      "lastUpdatedAt": "timestamp",
      "completedAt": "timestamp"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
jobs	list  member: JobSummary  java class: java.util.List	A list of jobs.
jobArn	string	The job ARN.
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the thing group.
targetSelection	string	Specifies whether the job will continue to run (CONTINUOUS), or will be complete after all those things specified as targets have completed the job (SNAPSHOT). If continuous, the job may also be run on a thing when a change is detected in a target. For example, a job will run on a thing when the thing is added to a target group, even after the job was completed by all things originally in the group.  enum: CONTINUOUS   SNAPSHOT
status	string	The job summary status.  enum: IN_PROGRESS   CANCELED   COMPLETED   DELETION_IN_PROGRESS
createdAt	timestamp	The time, in seconds since the epoch, when the job was created.
lastUpdatedAt	timestamp	The time, in seconds since the epoch, when the job was last updated.
completedAt	timestamp	The time, in seconds since the epoch, when the job completed.

Name	Type	Description
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**ServiceUnavailableException**

The service is temporarily unavailable.

## ListOTAUpdates

Lists OTA updates.

### Synopsis

```
aws iot list-ota-updates \
[--max-results <value>] \
[--next-token <value>] \
[--ota-update-status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "maxResults": "integer",
  "nextToken": "string",
  "otaUpdateStatus": "string"
}
```

### cli-input-json fields

Name	Type	Description
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.
nextToken	string	A token used to retrieve the next set of results.

Name	Type	Description
otaUpdateStatus	string	The OTA update job status.  enum: CREATE_PENDING   CREATE_IN_PROGRESS   CREATE_COMPLETE   CREATE_FAILED

### Output

```
{
    "otaUpdates": [
        {
            "otaUpdateId": "string",
            "otaUpdateArn": "string",
            "creationDate": "timestamp"
        }
    ],
    "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
otaUpdates	list  member: OTAUpdateSummary	A list of OTA update jobs.
otaUpdateId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The OTA update ID.
otaUpdateArn	string	The OTA update ARN.
creationDate	timestamp	The date when the OTA update was created.
nextToken	string	A token to use to get the next set of results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### ThrottlingException

The rate exceeds the limit.

#### UnauthorizedException

You are not authorized to perform this operation.

#### InternalFailureException

An unexpected error has occurred.

### ServiceUnavailableException

The service is temporarily unavailable.

## ListOutgoingCertificates

Lists certificates that are being transferred but not yet accepted.

### Synopsis

```
aws iot list-outgoing-certificates \
[--page-size <value>] \
[--marker <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "pageSize": "integer",
  "marker": "string",
  "ascendingOrder": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
pageSize	integer range- max:250 min:1	The result page size.
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
ascendingOrder	boolean	Specifies the order for results. If True, the results are returned in ascending order, based on the creation date.

### Output

```
{
  "outgoingCertificates": [
    {
      "certificateArn": "string",
      "certificateId": "string",
      "transferredTo": "string",
      "transferDate": "timestamp",
      "transferMessage": "string",
      "creationDate": "timestamp"
    }
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
outgoingCertificates	list  member: OutgoingCertificate  java class: java.util.List	The certificates that are being transferred but not yet accepted.
certificateArn	string	The certificate ARN.
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The certificate ID.
transferredTo	string  length- max:12 min:12  pattern: [0-9]+	The AWS account to which the transfer was made.
transferDate	timestamp	The date the transfer was initiated.
transferMessage	string  length- max:128	The transfer message.
creationDate	timestamp	The certificate creation date.
nextMarker	string  pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListPolicies

Lists your policies.

## Synopsis

```
aws iot list-policies \
[--marker <value>] \
[--page-size <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "marker": "string",
  "pageSize": "integer",
  "ascendingOrder": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
marker	string pattern: [A-Za-z0-9+/]+={0,2}	The marker for the next set of results.
pageSize	integer range- max:250 min:1	The result page size.
ascendingOrder	boolean	Specifies the order for results. If true, the results are returned in ascending creation order.

## Output

```
{
  "policies": [
    {
      "policyName": "string",
      "policyArn": "string"
    }
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
policies	list member: Policy java class: java.util.List	The descriptions of the policies.
policyName	string length- max:128 min:1	The policy name.

Name	Type	Description
	pattern: [w+=,.@-]+	
policyArn	string	The policy ARN.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results, or null if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListPolicyPrincipals

Lists the principals associated with the specified policy.

**Note:** This API is deprecated. Please use `ListTargetsForPolicy` instead.

### Synopsis

```
aws iot list-policy-principals \
  --policy-name <value> \
  [--marker <value>] \
  [--page-size <value>] \
  [--ascending-order | --no-ascending-order] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "policyName": "string",
  "marker": "string",
  "pageSize": "integer",
  "ascendingOrder": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string	The policy name.

Name	Type	Description
	length- max:128 min:1 pattern: [w+=,.@-]+	
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
pageSize	integer range- max:250 min:1	The result page size.
ascendingOrder	boolean	Specifies the order for results. If true, the results are returned in ascending creation order.

## Output

```
{
  "principals": [
    "string"
  ],
  "nextMarker": "string"
}
```

## CLI output fields

Name	Type	Description
principals	list member: PrincipalArn java class: java.util.List	The descriptions of the principals.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results, or null if there are no additional results.

## Errors

### ResourceNotFoundException

The specified resource does not exist.

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### `InternalFailureException`

An unexpected error has occurred.

## ListPolicyVersions

Lists the versions of the specified policy and identifies the default version.

### Synopsis

```
aws iot list-policy-versions \
    --policy-name <value> \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.

### Output

```
{
  "policyVersions": [
    {
      "versionId": "string",
      "isDefaultVersion": "boolean",
      "createDate": "timestamp"
    }
  ]
}
```

### CLI output fields

Name	Type	Description
policyVersions	list  member: PolicyVersion  java class: java.util.List	The policy versions.
versionId	string  pattern: [0-9]+	The policy version ID.

Name	Type	Description
isDefaultVersion	boolean	Specifies whether the policy version is the default.
createDate	timestamp	The date and time the policy was created.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListPrincipalPolicies

Lists the policies attached to the specified principal. If you use an Cognito identity, the ID must be in [AmazonCognito Identity format](#).

**Note:** This API is deprecated. Please use `ListAttachedPolicies` instead.

### Synopsis

```
aws iot list-principal-policies \
--principal <value> \
[--marker <value>] \
[--page-size <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "principal": "string",
  "marker": "string",
  "pageSize": "integer",
  "ascendingOrder": "boolean"
```

}

### cli-input-json fields

Name	Type	Description
principal	string	The principal.
marker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results.
pageSize	integer range- max:250 min:1	The result page size.
ascendingOrder	boolean	Specifies the order for results. If true, results are returned in ascending creation order.

### Output

```
{
  "policies": [
    {
      "policyName": "string",
      "policyArn": "string"
    }
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
policies	list member: Policy java class: java.util.List	The policies.
policyName	string length- max:128 min:1 pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	The marker for the next set of results, or null if there are no additional results.

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## ListPrincipalThings

Lists the things associated with the specified principal. A principal can be X.509 certificates, IAM users, groups, and roles, Amazon Cognito identities or federated identities.

### Synopsis

```
aws iot list-principal-things \
[--next-token <value>] \
[--max-results <value>] \
--principal <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "principal": "string"
}
```

### cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return in this operation.
principal	string	The principal.

### Output

```
{
  "things": [
    "string"
```

```
    ],
    "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
things	list member: ThingName	The things.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## ListRoleAliases

Lists the role aliases registered in your account.

### Synopsis

```
aws iot list-role-aliases \
[--page-size <value>] \
[--marker <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "pageSize": "integer",
  "marker": "string",
  "ascendingOrder": "boolean"
```

}

### cli-input-json fields

Name	Type	Description
pageSize	integer range- max:250 min:1	The maximum number of results to return at one time.
marker	string pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.
ascendingOrder	boolean	Return the list of role aliases in ascending alphabetical order.

### Output

```
{
  "roleAliases": [
    "string"
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
roleAliases	list member: RoleAlias java class: java.util.List	The role aliases.
nextMarker	string pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### ThrottlingException

The rate exceeds the limit.

#### UnauthorizedException

You are not authorized to perform this operation.

#### ServiceUnavailableException

The service is temporarily unavailable.

#### InternalFailureException

An unexpected error has occurred.

# ListScheduledAudits

Lists all of your scheduled audits.

## Synopsis

```
aws iot list-scheduled-audits \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

## cli-input-json fields

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time. The default is 25.

## Output

```
{
  "scheduledAudits": [
    {
      "scheduledAuditName": "string",
      "scheduledAuditArn": "string",
      "frequency": "string",
      "dayOfMonth": "string",
      "dayOfWeek": "string"
    }
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
scheduledAudits	list member: ScheduledAuditMetadata java class: java.util.List	The list of scheduled audits.

Name	Type	Description
scheduledAuditName	string length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit.
scheduledAuditArn	string	The ARN of the scheduled audit.
frequency	string	How often the scheduled audit takes place.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string pattern: ^([1-9] 1[2][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit is run (if the frequency is "MONTHLY"). If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit is run (if the frequency is "WEEKLY" or "BIWEEKLY").  enum: SUN   MON   TUE   WED   THU   FRI   SAT
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### InternalFailureException

An unexpected error has occurred.

## ListSecurityProfiles

Lists the Device Defender security profiles you have created. You can use filters to list only those security profiles associated with a thing group or only those associated with your account.

### Synopsis

```
aws iot list-security-profiles \
```

```
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "nextToken": "string",
  "maxResults": "integer"
}
```

#### cli-input-json fields

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

#### Output

```
{
  "securityProfileIdentifiers": [
    {
      "name": "string",
      "arn": "string"
    }
  ],
  "nextToken": "string"
}
```

#### CLI output fields

Name	Type	Description
securityProfileIdentifiers	list member: SecurityProfileIdentifier java class: java.util.List	A list of security profile identifiers (names and ARNs).
name	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name you have given to the security profile.
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or null if there are no additional results.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

# ListSecurityProfilesForTarget

Lists the Device Defender security profiles attached to a target (thing group).

## Synopsis

```
aws iot list-security-profiles-for-target \
[--next-token <value>] \
[--max-results <value>] \
[--recursive | --no-recursive] \
--security-profile-target-arn <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "recursive": "boolean",
  "securityProfileTargetArn": "string"
}
```

## cli-input-json fields

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.
recursive	boolean	If true, return child groups as well.
securityProfileTargetArn	string	The ARN of the target (thing group) whose attached security profiles you want to get.

## Output

```
{
  "securityProfileTargetMappings": [
```

```
{
    "securityProfileIdentifier": {
        "name": "string",
        "arn": "string"
    },
    "target": {
        "arn": "string"
    }
},
],
"nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
securityProfileTargetMappings	list  member: SecurityProfileTargetMapping  java class: java.util.List	A list of security profiles and their associated targets.
securityProfileIdentifier	SecurityProfileIdentifier	Information that identifies the security profile.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the security profile.
arn	string	The ARN of the security profile.
target	SecurityProfileTarget	Information about the target (thing group) associated with the security profile.
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### ThrottlingException

The rate exceeds the limit.

#### InternalFailureException

An unexpected error has occurred.

#### ResourceNotFoundException

The specified resource does not exist.

# ListStreams

Lists all of the streams in your AWS account.

## Synopsis

```
aws iot list-streams \
[--max-results <value>] \
[--next-token <value>] \
[--ascending-order | --no-ascending-order] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "maxResults": "integer",
  "nextToken": "string",
  "ascendingOrder": "boolean"
}
```

## cli-input-json fields

Name	Type	Description
maxResults	integer range- max:250 min:1	The maximum number of results to return at a time.
nextToken	string	A token used to get the next set of results.
ascendingOrder	boolean	Set to true to return the list of streams in ascending order.

## Output

```
{
  "streams": [
    {
      "streamId": "string",
      "streamArn": "string",
      "streamVersion": "integer",
      "description": "string"
    }
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
streams	list member: StreamSummary	A list of streams.
streamId	string	The stream ID.

Name	Type	Description
	length- max:128 min:1 pattern: [a-zA-Z0-9_-]+	
streamArn	string	The stream ARN.
streamVersion	integer range- max:65535 min:0	The stream version.
description	string length- max:2028 pattern: [^\p{C}]+	A description of the stream.
nextToken	string	A token used to get the next set of results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## ListTagsForResource

Lists the tags (metadata) you have assigned to the resource.

### Synopsis

```
aws iot list-tags-for-resource \
--resource-arn <value> \
[--next-token <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "resourceArn": "string",
  "nextToken": "string"
}
```

### cli-input-json fields

Name	Type	Description
resourceArn	string	The ARN of the resource.
nextToken	string	The token to retrieve the next set of results.

### Output

```
{
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
tags	list member: Tag java class: java.util.List	The list of tags assigned to the resource.
Key	string	The tag's key.
Value	string	The tag's value.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### InternalFailureException

An unexpected error has occurred.

#### ResourceNotFoundException

The specified resource does not exist.

#### ThrottlingException

The rate exceeds the limit.

## ListTargetsForPolicy

List targets for the specified policy.

## Synopsis

```
aws iot list-targets-for-policy \
--policy-name <value> \
[--marker <value>] \
[--page-size <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "policyName": "string",
  "marker": "string",
  "pageSize": "integer"
}
```

### cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
marker	string  pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.
pageSize	integer  range- max:250 min:1	The maximum number of results to return at one time.

## Output

```
{
  "targets": [
    "string"
  ],
  "nextMarker": "string"
}
```

### CLI output fields

Name	Type	Description
targets	list  member: PolicyTarget  java class: java.util.List	The policy targets.
nextMarker	string  pattern: [A-Za-z0-9+/]{0,2}	A marker used to get the next set of results.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

## ListTargetsForSecurityProfile

Lists the targets (thing groups) associated with a given Device Defender security profile.

### Synopsis

```
aws iot list-targets-for-security-profile \
--security-profile-name <value> \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "securityProfileName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The security profile.

Name	Type	Description
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

### Output

```
{
  "securityProfileTargets": [
    {
      "arn": "string"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
securityProfileTargets	list member: SecurityProfileTarget java class: java.util.List	The thing groups to which the security profile is attached.
arn	string	The ARN of the security profile.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### ResourceNotFoundException

The specified resource does not exist.

#### ThrottlingException

The rate exceeds the limit.

#### InternalFailureException

An unexpected error has occurred.

## ListThingGroups

List the thing groups in your account.

## Synopsis

```
aws iot list-thing-groups \
[--next-token <value>] \
[--max-results <value>] \
[--parent-group <value>] \
[--name-prefix-filter <value>] \
[--recursive | --no-recursive] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

## cli-input-json format

```
{
    "nextToken": "string",
    "maxResults": "integer",
    "parentGroup": "string",
    "namePrefixFilter": "string",
    "recursive": "boolean"
}
```

## cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.
parentGroup	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter that limits the results to those with the specified parent group.
namePrefixFilter	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter that limits the results to those with the specified name prefix.
recursive	boolean	If true, return child groups as well.

## Output

```
{
    "thingGroups": [
        {
            "groupName": "string",
            "groupArn": "string"
        }
    ],
    "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
thingGroups	list  member: GroupNameAndArn  java class: java.util.List	The thing groups.
groupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The group name.
groupArn	string	The group ARN.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

**InvalidRequestException**

The contents of the request were invalid.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

# ListThingGroupsForThing

List the thing groups to which the specified thing belongs.

## Synopsis

```
aws iot list-thing-groups-for-thing \
--thing-name <value> \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing name.
nextToken	string	The token to retrieve the next set of results.
maxResults	integer  range- max:250 min:1	The maximum number of results to return at one time.

### Output

```
{
  "thingGroups": [
    {
      "groupName": "string",
      "groupArn": "string"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
thingGroups	list  member: GroupNameAndArn  java class: java.util.List	The thing groups.
groupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The group name.
groupArn	string	The group ARN.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`InternalFailureException`

An unexpected error has occurred.

#### ResourceNotFoundException

The specified resource does not exist.

## ListThingPrincipals

Lists the principals associated with the specified thing. A principal can be X.509 certificates, IAM users, groups, and roles, Amazon Cognito identities or federated identities.

#### Synopsis

```
aws iot list-thing-principals \
--thing-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.

#### Output

```
{
  "principals": [
    "string"
  ]
}
```

#### CLI output fields

Name	Type	Description
principals	list  member: PrincipalArn  java class: java.util.List	The principals associated with the thing.

#### Errors

##### InvalidRequestException

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## ListThingRegistrationTaskReports

Information about the thing registration tasks.

### Synopsis

```
aws iot list-thing-registration-task-reports \
  --task-id <value> \
  --report-type <value> \
  [--next-token <value>] \
  [--max-results <value>] \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "taskId": "string",
  "reportType": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
taskId	string length- max:40	The id of the task.
reportType	string	The type of task report. enum: ERRORS   RESULTS
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return per request.

## Output

```
{
  "resourceLinks": [
    "string"
  ],
  "reportType": "string",
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
resourceLinks	list  member: S3FileUrl	Links to the task resources.
reportType	string	The type of task report.  enum: ERRORS   RESULTS
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### InternalFailureException

An unexpected error has occurred.

# ListThingRegistrationTasks

List bulk thing provisioning tasks.

## Synopsis

```
aws iot list-thing-registration-tasks \
[--next-token <value>] \
[--max-results <value>] \
[--status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
```

```

    "nextToken": "string",
    "maxResults": "integer",
    "status": "string"
}

```

### cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.
status	string	The status of the bulk thing provisioning task.  enum: InProgress   Completed   Failed   Cancelled   Cancelling

### Output

```

{
    "taskIds": [
        "string"
    ],
    "nextToken": "string"
}

```

### CLI output fields

Name	Type	Description
taskIds	list member: TaskId	A list of bulk thing provisioning task IDs.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**InternalFailureException**

An unexpected error has occurred.

# ListThingTypes

Lists the existing thing types.

## Synopsis

```
aws iot list-thing-types \
[--next-token <value>] \
[--max-results <value>] \
[--thing-type-name <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "thingTypeName": "string"
}
```

## cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return in this operation.
thingTypeName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing type.

## Output

```
{
  "thingTypes": [
    {
      "thingTypeName": "string",
      "thingTypeArn": "string",
      "thingTypeProperties": {
        "thingTypeDescription": "string",
        "searchableAttributes": [
          "string"
        ]
      },
      "thingTypeMetadata": {
        "deprecated": "boolean",
        "deprecationDate": "timestamp",
        "creationDate": "timestamp"
      }
    }
  ],
  "nextToken": "string"
```

}

## CLI output fields

Name	Type	Description
thingTypes	list  member: ThingTypeDefinition  java class: java.util.List	The thing types.
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name of the thing type.
thingTypeArn	string	The thing type ARN.
thingTypeProperties	ThingTypeProperties	The ThingTypeProperties for the thing type.
thingTypeDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The description of the thing type.
searchableAttributes	list  member: AttributeName  java class: java.util.List	A list of searchable thing attribute names.
thingTypeMetadata	ThingTypeMetadata	The ThingTypeMetadata contains additional information about the thing type including: creation date and time, a value indicating whether the thing type is deprecated, and a date and time when it was deprecated.
deprecated	boolean	Whether the thing type is deprecated. If <b>true</b> , no new things could be associated with this type.
deprecationDate	timestamp	The date and time when the thing type was deprecated.
creationDate	timestamp	The date and time when the thing type was created.
nextToken	string	The token for the next set of results, or <b>null</b> if there are no additional results.

## Errors

#### `InvalidRequestException`

The contents of the request were invalid.

#### `ThrottlingException`

The rate exceeds the limit.

#### `UnauthorizedException`

You are not authorized to perform this operation.

#### `ServiceUnavailableException`

The service is temporarily unavailable.

#### `InternalFailureException`

An unexpected error has occurred.

## ListThings

Lists your things. Use the `attributeName` and `attributeValue` parameters to filter your things. For example, calling `ListThings` with `attributeName=Color` and `attributeValue=Red` retrieves all things in the registry that contain an attribute **Color** with the value **Red**.

### Synopsis

```
aws iot list-things \
[--next-token <value>] \
[--max-results <value>] \
[--attribute-name <value>] \
[--attribute-value <value>] \
[--thing-type-name <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "nextToken": "string",
  "maxResults": "integer",
  "attributeName": "string",
  "attributeValue": "string",
  "thingTypeName": "string"
}
```

### cli-input-json fields

Name	Type	Description
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return in this operation.

Name	Type	Description
attributeName	string length- max:128 pattern: [a-zA-Z0-9_,@/:#-]+	The attribute name used to search for things.
attributeValue	string length- max:800 pattern: [a-zA-Z0-9_,@/:#-]*	The attribute value used to search for things.
thingTypeName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name of the thing type used to search for things.

### Output

```
{
  "things": [
    {
      "thingName": "string",
      "thingTypeName": "string",
      "thingArn": "string",
      "attributes": {
        "string": "string"
      },
      "version": "long"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
things	list member: ThingAttribute java class: java.util.List	The things.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name of the thing.
thingTypeName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_-]+	The name of the thing type, if the thing has been associated with a type.
thingArn	string	The thing ARN.

Name	Type	Description
attributes	map	A list of thing attributes which are name-value pairs.
version	long	The version of the thing record in the registry.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### InternalFailureException

An unexpected error has occurred.

## ListThingsInBillingGroup

Lists the things you have added to the given billing group.

### Synopsis

```
aws iot list-things-in-billing-group \
--billing-group-name <value> \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "billingGroupName": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
billingGroupName	string	The name of the billing group.

Name	Type	Description
	length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	
nextToken	string	The token to retrieve the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return per request.

## Output

```
{
  "things": [
    "string"
  ],
  "nextToken": "string"
}
```

## CLI output fields

Name	Type	Description
things	list member: ThingName	A list of things in the billing group.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

# ListThingsInThingGroup

Lists the things in the specified group.

## Synopsis

```
aws iot list-things-in-thing-group \
--thing-group-name <value> \
[--recursive | --no-recursive] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingGroupName": "string",
  "recursive": "boolean",
  "nextToken": "string",
  "maxResults": "integer"
}
```

#### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing group name.
recursive	boolean	When true, list things in this thing group and in all child groups as well.
nextToken	string	The token to retrieve the next set of results.
maxResults	integer  range- max:250 min:1	The maximum number of results to return at one time.

#### Output

```
{
  "things": [
    "string"
  ],
  "nextToken": "string"
}
```

#### CLI output fields

Name	Type	Description
things	list  member: ThingName	The things in the specified thing group.
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

# ListTopicRules

Lists the rules for the specific topic.

## Synopsis

```
aws iot list-topic-rules \
[--topic <value>] \
[--max-results <value>] \
[--next-token <value>] \
[--rule-disabled | --no-rule-disabled] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "topic": "string",  
    "maxResults": "integer",  
    "nextToken": "string",  
    "ruleDisabled": "boolean"  
}
```

## cli-input-json fields

Name	Type	Description
topic	string	The topic.
maxResults	integer range- max:10000 min:1	The maximum number of results to return.
nextToken	string	A token used to retrieve the next value.
ruleDisabled	boolean	Specifies whether the rule is disabled.

## Output

```
{
```

```

"rules": [
  {
    "ruleArn": "string",
    "ruleName": "string",
    "topicPattern": "string",
    "createdAt": "timestamp",
    "ruleDisabled": "boolean"
  }
],
"nextToken": "string"
}

```

### CLI output fields

Name	Type	Description
rules	list  member: TopicRuleListItem	The rules.
ruleArn	string	The rule ARN.
ruleName	string  length- max:128 min:1  pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.
topicPattern	string	The pattern for the topic names that apply.
createdAt	timestamp	The date and time the rule was created.
ruleDisabled	boolean	Specifies whether the rule is disabled.
nextToken	string	A token used to retrieve the next value.

### Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

## ListV2LoggingLevels

Lists logging levels.

### Synopsis

```
aws iot list-v2-logging-levels \
[--target-type <value>] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "targetType": "string",
  "nextToken": "string",
  "maxResults": "integer"
}
```

### cli-input-json fields

Name	Type	Description
targetType	string	The type of resource for which you are configuring logging. Must be THING_Group.  enum: DEFAULT   THING_GROUP
nextToken	string	The token used to get the next set of results, or null if there are no additional results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

### Output

```
{
  "logTargetConfigurations": [
    {
      "logTarget": {
        "targetType": "string",
        "targetName": "string"
      },
      "logLevel": "string"
    }
  ],
  "nextToken": "string"
}
```

### CLI output fields

Name	Type	Description
logTargetConfigurations	list member: LogTargetConfiguration	The logging configuration for a target.
logTarget	LogTarget	A log target

Name	Type	Description
targetType	string	The target type. enum: DEFAULT   THING_GROUP
targetName	string	The target name.
logLevel	string	The logging level. enum: DEBUG   INFO   ERROR   WARN   DISABLED
nextToken	string	The token used to get the next set of results, or <b>null</b> if there are no additional results.

### Errors

**InternalException**

An unexpected error has occurred.

**NotConfiguredException**

The resource is not configured.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

## ListViolationEvents

Lists the Device Defender security profile violations discovered during the given time period. You can use filters to limit the results to those alerts issued for a particular security profile, behavior or thing (device).

### Synopsis

```
aws iot list-violation-events \
--start-time <value> \
--end-time <value> \
[--thing-name <value>] \
[--security-profile-name <value>] \
[--next-token <value>] \
[--max-results <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "startTime": "timestamp",
  "endTime": "timestamp",
  "thingName": "string",
```

```

    "securityProfileName": "string",
    "nextToken": "string",
    "maxResults": "integer"
}

```

### cli-input-json fields

Name	Type	Description
startTime	timestamp	The start time for the alerts to be listed.
endTime	timestamp	The end time for the alerts to be listed.
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter to limit results to those alerts caused by the specified thing.
securityProfileName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	A filter to limit results to those alerts generated by the specified security profile.
nextToken	string	The token for the next set of results.
maxResults	integer range- max:250 min:1	The maximum number of results to return at one time.

### Output

```
{
  "violationEvents": [
    {
      "violationId": "string",
      "thingName": "string",
      "securityProfileName": "string",
      "behavior": {
        "name": "string",
        "metric": "string",
        "criteria": {
          "comparisonOperator": "string",
          "value": {
            "count": "long",
            "cidrs": [
              "string"
            ],
            "ports": [
              "integer"
            ]
          }
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ]
}
```

```

        },
        "metricValue": {
            "count": "long",
            "cidrs": [
                "string"
            ],
            "ports": [
                "integer"
            ]
        },
        "violationEventType": "string",
        "violationEventTime": "timestamp"
    }
],
"nextToken": "string"
}

```

### CLI output fields

Name	Type	Description
violationEvents	list  member: ViolationEvent	The security profile violation alerts issued for this account during the given time frame, potentially filtered by security profile, behavior violated, or thing (device) violating.
violationId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The ID of the violation event.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name of the thing responsible for the violation event.
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name of the security profile whose behavior was violated.
behavior	Behavior	The behavior which was violated.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9-_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.

Name	Type	Description
comparisonOperator	string	The operator that relates the thing measured ( <code>metric</code> ) to the criteria (containing a <code>value</code> or <code>statisticalThreshold</code> ).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the <code>metric</code> .
count	long  range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cidrs	list  member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list  member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, <code>NUM_MESSAGES_SENT</code> ). For a <code>statisticalThreshold</code> metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer  range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.

Name	Type	Description
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.
statistic	string pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
metricValue	MetricValue	The value of the metric (the measurement).
count	long range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cids	list member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .

Name	Type	Description
violationEventType	string	The type of violation event. enum: in-alarm   alarm-cleared   alarm-invalidated
violationEventTime	timestamp	The time the violation event occurred.
nextToken	string	A token that can be used to retrieve the next set of results, or <code>null</code> if there are no additional results.

### Errors

#### InvalidRequestException

The contents of the request were invalid.

#### ThrottlingException

The rate exceeds the limit.

#### InternalFailureException

An unexpected error has occurred.

## Publish

Publishes state information.

For more information, see [HTTP Protocol](#) in the AWS IoT Developer Guide.

### Synopsis

```
aws iot-data publish \
[--topic <value>] \
[--qos <value>] \
[--payload <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "topic": "string",
  "qos": "integer",
  "payload": "blob"
}
```

### cli-input-json fields

Name	Type	Description
topic	string	The name of the MQTT topic.

Name	Type	Description
qos	integer range- max:1 min:0	The Quality of Service (QoS) level.
payload	blob	The state information, in JSON format.

#### Output

None

#### Errors

`InternalFailureException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`UnauthorizedException`

You are not authorized to perform this operation.

`MethodNotAllowedException`

The specified combination of HTTP verb and URI is not supported.

## RegisterCACertificate

Registers a CA certificate with AWS IoT. This CA certificate can then be used to sign device certificates, which can be then registered with AWS IoT. You can register up to 10 CA certificates per AWS account that have the same subject field. This enables you to have up to 10 certificate authorities sign your device certificates. If you have more than one CA certificate registered, make sure you pass the CA certificate when you register your device certificates with the RegisterCertificate API.

#### Synopsis

```
aws iot register-ca-certificate \
--ca-certificate <value> \
--verification-certificate <value> \
[--set-as-active | --no-set-as-active] \
[--allow-auto-registration | --no-allow-auto-registration] \
[--registration-config <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "caCertificate": "string",
  "verificationCertificate": "string",
  "setAsActive": "boolean",
  "allowAutoRegistration": "boolean",
  "registrationConfig": {
    "templateBody": "string",
    "roleArn": "string"
  }
}
```

```
}
```

### cli-input-json fields

Name	Type	Description
caCertificate	string length- max:65536 min:1	The CA certificate.
verificationCertificate	string length- max:65536 min:1	The private key verification certificate.
setAsActive	boolean	A boolean value that specifies if the CA certificate is set to active.
allowAutoRegistration	boolean	Allows this CA certificate to be used for auto registration of device certificates.
registrationConfig	RegistrationConfig	Information about the registration configuration.
templateBody	string	The template body.
roleArn	string length- max:2048 min:20	The ARN of the role.

### Output

```
{
  "certificateArn": "string",
  "certificateId": "string"
}
```

### CLI output fields

Name	Type	Description
certificateArn	string	The CA certificate ARN.
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The CA certificate identifier.

### Errors

`ResourceAlreadyExistsException`

The resource already exists.

`RegistrationCodeValidationException`

The registration code is invalid.

**InvalidRequestException**

The contents of the request were invalid.

**CertificateValidationException**

The certificate is invalid.

**ThrottlingException**

The rate exceeds the limit.

**LimitExceededException**

A limit has been exceeded.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## RegisterCertificate

Registers a device certificate with AWS IoT. If you have more than one CA certificate that has the same subject field, you must specify the CA certificate that was used to sign the device certificate being registered.

### Synopsis

```
aws iot register-certificate \
--certificate-pem <value> \
[--ca-certificate-pem <value>] \
[--set-as-active | --no-set-as-active] \
[--status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "certificatePem": "string",  
    "caCertificatePem": "string",  
    "status": "string"  
}
```

### cli-input-json fields

Name	Type	Description
certificatePem	string length- max:65536 min:1	The certificate data, in PEM format.
caCertificatePem	string length- max:65536 min:1	The CA certificate used to sign the device certificate being registered.

Name	Type	Description
status	string	The status of the register certificate request.  enum: ACTIVE   INACTIVE   REVOKED   PENDING_TRANSFER   REGISTER_INACTIVE   PENDING_ACTIVATION

#### Output

```
{
  "certificateArn": "string",
  "certificateId": "string"
}
```

#### CLI output fields

Name	Type	Description
certificateArn	string	The certificate ARN.
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The certificate identifier.

#### Errors

**ResourceAlreadyExistsException**

The resource already exists.

**InvalidRequestException**

The contents of the request were invalid.

**CertificateValidationException**

The certificate is invalid.

**CertificateStateException**

The certificate operation is not allowed.

**CertificateConflictException**

Unable to verify the CA certificate used to sign the device certificate you are attempting to register. This happens when you have registered more than one CA certificate that has the same subject field and public key.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

# RegisterThing

Provisions a thing.

## Synopsis

```
aws iot register-thing \
--template-body <value> \
[--parameters <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "templateBody": "string",  
    "parameters": {  
        "string": "string"  
    }  
}
```

### cli-input-json fields

Name	Type	Description
templateBody	string	The provisioning template. See <a href="#">Programmatic Provisioning</a> for more information.
parameters	map	The parameters for provisioning a thing. See <a href="#">Programmatic Provisioning</a> for more information.

## Output

```
{  
    "certificatePem": "string",  
    "resourceArns": {  
        "string": "string"  
    }  
}
```

### CLI output fields

Name	Type	Description
certificatePem	string	.
resourceArns	map	ARNs for the generated resources.

## Errors

`InternalFailureException`

An unexpected error has occurred.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InvalidRequestException`

The contents of the request were invalid.

`UnauthorizedException`

You are not authorized to perform this operation.

`ThrottlingException`

The rate exceeds the limit.

`ConflictingResourceUpdateException`

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

`ResourceRegistrationFailureException`

The resource registration failed.

# RejectCertificateTransfer

Rejects a pending certificate transfer. After AWS IoT rejects a certificate transfer, the certificate status changes from **PENDING\_TRANSFER** to **INACTIVE**.

To check for pending certificate transfers, call `ListCertificates` to enumerate your certificates.

This operation can only be called by the transfer destination. After it is called, the certificate will be returned to the source's account in the **INACTIVE** state.

## Synopsis

```
aws iot reject-certificate-transfer \
--certificate-id <value> \
[--reject-reason <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateId": "string",
  "rejectReason": "string"
}
```

## cli-input-json fields

Name	Type	Description
<code>certificateId</code>	string  length- max:64 min:64	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)

Name	Type	Description
	pattern: (0x)?[a-fA-F0-9]+	
rejectReason	string length- max:128	The reason the certificate transfer was rejected.

#### Output

None

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`TransferAlreadyCompletedException`

You can't revert the certificate transfer because the transfer is already complete.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## RemoveThingFromBillingGroup

Removes the given thing from the billing group.

#### Synopsis

```
aws iot remove-thing-from-billing-group \
[--billing-group-name <value>] \
[--billing-group-arn <value>] \
[--thing-name <value>] \
[--thing-arn <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "billingGroupName": "string",
  "billingGroupArn": "string",
  "thingName": "string",
  "thingArn": "string"
```

}

### **cli-input-json fields**

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.
billingGroupArn	string	The ARN of the billing group.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing to be removed from the billing group.
thingArn	string	The ARN of the thing to be removed from the billing group.

### **Output**

None

### **Errors**

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## RemoveThingFromThingGroup

Remove the specified thing from the specified group.

### **Synopsis**

```
aws iot remove-thing-from-thing-group \
[--thing-group-name <value>] \
[--thing-group-arn <value>] \
[--thing-name <value>] \
[--thing-arn <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format**

```
{
  "thingGroupName": "string",
  "thingGroupArn": "string",
  "thingName": "string",
  "thingArn": "string"
}
```

#### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The group name.
thingGroupArn	string	The group ARN.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing to remove from the group.
thingArn	string	The ARN of the thing to remove from the group.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## ReplaceTopicRule

Replaces the rule. You must specify all parameters for the new rule. Creating rules is an administrator-level action. Any user who has permission to create rules will be able to access data processed by the rule.

#### Synopsis

```
aws iot replace-topic-rule \
```

```
--rule-name <value> \
--topic-rule-payload <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

cli-input-json format

```
{
  "ruleName": "string",
  "topicRulePayload": {
    "sql": "string",
    "description": "string",
    "actions": [
      {
        "dynamoDB": {
          "tableName": "string",
          "roleArn": "string",
          "operation": "string",
          "hashKeyField": "string",
          "hashKeyValue": "string",
          "hashKeyType": "string",
          "rangeKeyField": "string",
          "rangeKeyValue": "string",
          "rangeKeyType": "string",
          "payloadField": "string"
        },
        "dynamoDBv2": {
          "roleArn": "string",
          "putItem": {
            "tableName": "string"
          }
        },
        "lambda": {
          "functionArn": "string"
        },
        "sns": {
          "targetArn": "string",
          "roleArn": "string",
          "messageFormat": "string"
        },
        "sqs": {
          "roleArn": "string",
          "queueUrl": "string",
          "useBase64": "boolean"
        },
        "kinesis": {
          "roleArn": "string",
          "streamName": "string",
          "partitionKey": "string"
        },
        "republish": {
          "roleArn": "string",
          "topic": "string"
        },
        "s3": {
          "roleArn": "string",
          "bucketName": "string",
          "key": "string",
          "cannedAcl": "string"
        },
        "firehose": {
          "roleArn": "string",
          "deliveryStreamName": "string",
          "separator": "string"
        },
      ]
    ]
  }
}
```

```

    "cloudwatchMetric": {
        "roleArn": "string",
        "metricNamespace": "string",
        "metricName": "string",
        "metricValue": "string",
        "metricUnit": "string",
        "metricTimestamp": "string"
    },
    "cloudwatchAlarm": {
        "roleArn": "string",
        "alarmName": "string",
        "stateReason": "string",
        "stateValue": "string"
    },
    "elasticsearch": {
        "roleArn": "string",
        "endpoint": "string",
        "index": "string",
        "type": "string",
        "id": "string"
    },
    "salesforce": {
        "token": "string",
        "url": "string"
    },
    "iotAnalytics": {
        "channelArn": "string",
        "channelName": "string",
        "roleArn": "string"
    },
    "iotEvents": {
        "inputName": "string",
        "messageId": "string",
        "roleArn": "string"
    },
    "stepFunctions": {
        "executionNamePrefix": "string",
        "stateMachineName": "string",
        "roleArn": "string"
    }
},
],
"ruleDisabled": "boolean",
"awsIotSqlVersion": "string",
"errorAction": {
    "dynamoDB": {
        "tableName": "string",
        "roleArn": "string",
        "operation": "string",
        "hashKeyField": "string",
        "hashKeyValue": "string",
        "hashKeyType": "string",
        "rangeKeyField": "string",
        "rangeKeyValue": "string",
        "rangeKeyType": "string",
        "payloadField": "string"
    },
    "dynamoDBv2": {
        "roleArn": "string",
        "putItem": {
            "tableName": "string"
        }
    },
    "lambda": {
        "functionArn": "string"
    }
},

```

```

"sns": {
    "targetArn": "string",
    "roleArn": "string",
    "messageFormat": "string"
},
"sqs": {
    "roleArn": "string",
    "queueUrl": "string",
    "useBase64": "boolean"
},
"kinesis": {
    "roleArn": "string",
    "streamName": "string",
    "partitionKey": "string"
},
"republish": {
    "roleArn": "string",
    "topic": "string"
},
"s3": {
    "roleArn": "string",
    "bucketName": "string",
    "key": "string",
    "cannedAcl": "string"
},
"firehose": {
    "roleArn": "string",
    "deliveryStreamName": "string",
    "separator": "string"
},
"cloudwatchMetric": {
    "roleArn": "string",
    "metricNamespace": "string",
    "metricName": "string",
    "metricValue": "string",
    "metricUnit": "string",
    "metricTimestamp": "string"
},
"cloudwatchAlarm": {
    "roleArn": "string",
    "alarmName": "string",
    "stateReason": "string",
    "stateValue": "string"
},
"elasticsearch": {
    "roleArn": "string",
    "endpoint": "string",
    "index": "string",
    "type": "string",
    "id": "string"
},
"salesforce": {
    "token": "string",
    "url": "string"
},
"iotAnalytics": {
    "channelArn": "string",
    "channelName": "string",
    "roleArn": "string"
},
"iotEvents": {
    "inputName": "string",
    "messageId": "string",
    "roleArn": "string"
},
"stepFunctions": {

```

```

        "executionNamePrefix": "string",
        "stateMachineName": "string",
        "roleArn": "string"
    }
}
}
}
```

#### cli-input-json fields

Name	Type	Description
ruleName	string length- max:128 min:1 pattern: ^[a-zA-Z0-9_]+\$	The name of the rule.
topicRulePayload	TopicRulePayload	The rule payload.
sql	string	The SQL statement used to query the topic. For more information, see <a href="#">AWS IoT SQL Reference</a> in the <a href="#">AWS IoT Developer Guide</a> .
description	string	The description of the rule.
actions	list member: Action	The actions associated with the rule.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
operation	string	The type of operation to be performed. This follows the substitution template, so it can be \$ <i>operation</i> , but the substitution must result in one of the following: <b>INSERT</b> , <b>UPDATE</b> , or <b>DELETE</b> .
hashKeyField	string	The hash key name.
hashKeyValue	string	The hash key value.
hashKeyType	string enum: STRING   NUMBER	The hash key type. Valid values are "STRING" or "NUMBER"
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.

Name	Type	Description
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
putItem	PutItemInput	<p>Specifies the DynamoDB table to which the message data will be written. For example:</p> <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> <p>Each attribute in the message payload will be written to a separate column in the DynamoDB database.</p>
tableName	string	The table where the message data will be written.
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.
roleArn	string	The ARN of the IAM role that grants access.

Name	Type	Description
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.
key	string	The object key.

Name	Type	Description
cannedAcl	string	The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a> .  enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.
separator	string  pattern: ([ ] (  ) (.))	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ',' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.

Name	Type	Description
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.(sfdc-matrix.net) (sfdcnow.com))/streams/w <a href="#">1, 20</a> /w <a href="#">1, 20</a> /event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	iotAnalyticsAction	Sends message data to an AWS IoT Analytics channel.
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).

Name	Type	Description
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").
ruleDisabled	boolean	Specifies whether the rule is disabled.
awsIoTSqlVersion	string	The version of the SQL rules engine to use when evaluating the rule.
errorAction	Action	The action to take when an error occurs.
dynamoDB	DynamoDBAction	Write to a DynamoDB table.
tableName	string	The name of the DynamoDB table.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.

Name	Type	Description
operation	string	The type of operation to be performed. This follows the substitution template, so it can be <code>#\$ operation</code> , but the substitution must result in one of the following: INSERT, UPDATE, or DELETE.
hashKeyField	string	The hash key name.
hashKeyValue	string	The hash key value.
hashKeyType	string	The hash key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
rangeKeyField	string	The range key name.
rangeKeyValue	string	The range key value.
rangeKeyType	string	The range key type. Valid values are "STRING" or "NUMBER" enum: STRING   NUMBER
payloadField	string	The action payload. This name can be customized.
dynamoDBv2	DynamoDBv2Action	Write to a DynamoDB table. This is a new version of the DynamoDB action. It allows you to write each attribute in an MQTT message payload into a separate DynamoDB column.
roleArn	string	The ARN of the IAM role that grants access to the DynamoDB table.
putItem	PutItemInput	Specifies the DynamoDB table to which the message data will be written. For example:  <pre>{ "dynamoDBv2": { "roleArn": "aws:iam:12341251:my-role" "putItem": { "tableName": "my-table" } } }</pre> Each attribute in the message payload will be written to a separate column in the DynamoDB database.
tableName	string	The table where the message data will be written.

Name	Type	Description
lambda	LambdaAction	Invoke a Lambda function.
functionArn	string	The ARN of the Lambda function.
sns	SnsAction	Publish to an Amazon SNS topic.
targetArn	string	The ARN of the SNS topic.
roleArn	string	The ARN of the IAM role that grants access.
messageFormat	string	(Optional) The message format of the message to publish. Accepted values are "JSON" and "RAW". The default value of the attribute is "RAW". SNS uses this setting to determine if the payload should be parsed and relevant platform-specific bits of the payload should be extracted. To read more about SNS message formats, see <a href="https://docs.aws.amazon.com/sns/latest/dg/json-formats.html">https://docs.aws.amazon.com/sns/latest/dg/json-formats.html</a> refer to their official documentation.  enum: RAW   JSON
sqs	SqsAction	Publish to an Amazon SQS queue.
roleArn	string	The ARN of the IAM role that grants access.
queueUrl	string	The URL of the Amazon SQS queue.
useBase64	boolean	Specifies whether to use Base64 encoding.
kinesis	KinesisAction	Write data to an Amazon Kinesis stream.
roleArn	string	The ARN of the IAM role that grants access to the Amazon Kinesis stream.
streamName	string	The name of the Amazon Kinesis stream.
partitionKey	string	The partition key.
republish	RepublishAction	Publish to another MQTT topic.
roleArn	string	The ARN of the IAM role that grants access.

Name	Type	Description
topic	string	The name of the MQTT topic.
s3	S3Action	Write to an Amazon S3 bucket.
roleArn	string	The ARN of the IAM role that grants access.
bucketName	string	The Amazon S3 bucket.
key	string	The object key.
cannedAcl	string	<p>The Amazon S3 canned ACL that controls access to the object identified by the object key. For more information, see <a href="#">S3 canned ACLs</a>.</p> <p>enum: private   public-read   public-read-write   aws-exec-read   authenticated-read   bucket-owner-read   bucket-owner-full-control   log-delivery-write</p>
firehose	FirehoseAction	Write to an Amazon Kinesis Firehose stream.
roleArn	string	The IAM role that grants access to the Amazon Kinesis Firehose stream.
deliveryStreamName	string	The delivery stream name.
separator	string pattern: ([ ]) ( ) ()	A character separator that will be used to separate records written to the Firehose stream. Valid values are: '\n' (newline), '\t' (tab), '\r\n' (Windows newline), ';' (comma).
cloudwatchMetric	CloudwatchMetricAction	Capture a CloudWatch metric.
roleArn	string	The IAM role that allows access to the CloudWatch metric.
metricNamespace	string	The CloudWatch metric namespace name.
metricName	string	The CloudWatch metric name.
metricValue	string	The CloudWatch metric value.
metricUnit	string	The <a href="#">metric unit</a> supported by CloudWatch.
metricTimestamp	string	An optional <a href="#">Unix timestamp</a> .

Name	Type	Description
cloudwatchAlarm	CloudwatchAlarmAction	Change the state of a CloudWatch alarm.
roleArn	string	The IAM role that allows access to the CloudWatch alarm.
alarmName	string	The CloudWatch alarm name.
stateReason	string	The reason for the alarm change.
stateValue	string	The value of the alarm state. Acceptable values are: OK, ALARM, INSUFFICIENT_DATA.
elasticsearch	ElasticsearchAction	Write data to an Amazon Elasticsearch Service domain.
roleArn	string	The IAM role ARN that has access to Elasticsearch.
endpoint	string pattern: https?://.*	The endpoint of your Elasticsearch domain.
index	string	The Elasticsearch index where you want to store your data.
type	string	The type of document you are storing.
id	string	The unique identifier for the document you are storing.
salesforce	SalesforceAction	Send a message to a Salesforce IoT Cloud Input Stream.
token	string length- min:40	The token used to authenticate access to the Salesforce IoT Cloud Input Stream. The token is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
url	string length- max:2000 pattern: https://ingestion-[a-zA-Z0-9]{1,12}.[a-zA-Z0-9]+.(sfdc-matrix.net) (sfdcnow.com))/streams/w 1, 20/w 1, 20/event	The URL exposed by the Salesforce IoT Cloud Input Stream. The URL is available from the Salesforce IoT Cloud platform after creation of the Input Stream.
iotAnalytics	IotAnalyticsAction	Sends message data to an AWS IoT Analytics channel.

Name	Type	Description
channelArn	string	(deprecated) The ARN of the IoT Analytics channel to which message data will be sent.
channelName	string	The name of the IoT Analytics channel to which message data will be sent.
roleArn	string	The ARN of the role which has a policy that grants IoT Analytics permission to send message data via IoT Analytics (iotanalytics:BatchPutMessage).
iotEvents	IoTEventsAction	Sends an input to an AWS IoT Events detector.
inputName	string length- max:128 min:1	The name of the AWS IoT Events input.
messageId	string length- max:128	[Optional] Use this to ensure that only one input (message) with a given messageId will be processed by an AWS IoT Events detector.
roleArn	string	The ARN of the role that grants AWS IoT permission to send an input to an AWS IoT Events detector. ("Action":"iotevents:BatchPutMessage").
stepFunctions	StepFunctionsAction	Starts execution of a Step Functions state machine.
executionNamePrefix	string	(Optional) A name will be given to the state machine execution consisting of this prefix followed by a UUID. Step Functions automatically creates a unique name for each state machine execution if one is not provided.
stateMachineName	string	The name of the Step Functions state machine whose execution will be started.
roleArn	string	The ARN of the role that grants IoT permission to start execution of a state machine ("Action":"states:StartExecution").

## Output

None

## Errors

`SqlParseException`

The Rule-SQL expression can't be parsed correctly.

`InternalException`

An unexpected error has occurred.

`InvalidRequestException`

The contents of the request were invalid.

`ServiceUnavailableException`

The service is temporarily unavailable.

`UnauthorizedException`

You are not authorized to perform this operation.

`ConflictingResourceUpdateException`

A conflicting resource update exception. This exception is thrown when two pending updates cause a conflict.

# SearchIndex

The query search index.

## Synopsis

```
aws iot search-index \
[--index-name <value>] \
--query-string <value> \
[--next-token <value>] \
[--max-results <value>] \
[--query-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "indexName": "string",
  "queryString": "string",
  "nextToken": "string",
  "maxResults": "integer",
  "queryVersion": "string"
}
```

## cli-input-json fields

Name	Type	Description
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The search index name.
queryString	string	The search query string.

Name	Type	Description
	length- min:1	
nextToken	string	The token used to get the next set of results, or null if there are no additional results.
maxResults	integer range- max:500 min:1	The maximum number of results to return at one time.
queryVersion	string	The query version.

### Output

```
{
  "nextToken": "string",
  "things": [
    {
      "thingName": "string",
      "thingId": "string",
      "thingTypeName": "string",
      "thingGroupNames": [
        "string"
      ],
      "attributes": {
        "string": "string"
      },
      "shadow": "string",
      "connectivity": {
        "connected": "boolean",
        "timestamp": "long"
      }
    }
  ],
  "thingGroups": [
    {
      "thingGroupName": "string",
      "thingGroupId": "string",
      "thingGroupDescription": "string",
      "attributes": {
        "string": "string"
      },
      "parentGroupNames": [
        "string"
      ]
    }
  ]
}
```

### CLI output fields

Name	Type	Description
nextToken	string	The token used to get the next set of results, or null if there are no additional results.
things	list member: ThingDocument	The things that match the search query.

Name	Type	Description
	java class: java.util.List	
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing name.
thingId	string	The thing ID.
thingTypeName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing type name.
thingGroupNames	list  member: ThingGroupName  java class: java.util.List	Thing group names.
attributes	map	The attributes.
shadow	string	The shadow.
connectivity	ThingConnectivity	Indicates whether the thing is connected to the AWS IoT service.
connected	boolean	True if the thing is connected to the AWS IoT service; false if it is not connected.
timestamp	long	The epoch time (in milliseconds) when the thing last connected or disconnected. If the thing has been disconnected for more than a few weeks, the time value might be missing.
thingGroups	list  member: ThingGroupDocument  java class: java.util.List	The thing groups that match the search query.
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing group name.
thingGroupId	string  length- max:128 min:1  pattern: [a-zA-Z0-9-]+	The thing group ID.

Name	Type	Description
thingGroupDescription	string length- max:2028 pattern: [\p{Graph} ]*	The thing group description.
attributes	map	The thing group attributes.
parentGroupNames	list member: ThingGroupName java class: java.util.List	Parent group names.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidQueryException`

The query is invalid.

`IndexNotReadyException`

The index is not ready.

## SetDefaultAuthorizer

Sets the default authorizer. This will be used if a websocket connection is made without specifying an authorizer.

### Synopsis

```
aws iot set-default-authorizer \
--authorizer-name <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
    "authorizerName": "string"
}
```

#### cli-input-json fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.

#### Output

```
{
    "authorizerName": "string",
    "authorizerArn": "string"
}
```

#### CLI output fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.
authorizerArn	string	The authorizer ARN.

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`ResourceAlreadyExistsException`

The resource already exists.

# SetDefaultPolicyVersion

Sets the specified version of the specified policy as the policy's default (operative) version. This action affects all certificates to which the policy is attached. To list the principals the policy is attached to, use the ListPrincipalPolicy API.

## Synopsis

```
aws iot set-default-policy-version \
--policy-name <value> \
--policy-version-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "policyName": "string",  
    "policyVersionId": "string"  
}
```

## cli-input-json fields

Name	Type	Description
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyVersionId	string  pattern: [0-9]+	The policy version ID.

## Output

None

## Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## SetLoggingOptions

Sets the logging options.

NOTE: use of this command is not recommended. Use `SetV2LoggingOptions` instead.

### Synopsis

```
aws iot set-logging-options \
  --logging-options-payload <value> \
  [--cli-input-json <value>] \
  [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "loggingOptionsPayload": {
    "roleArn": "string",
    "logLevel": "string"
  }
}
```

### cli-input-json fields

Name	Type	Description
loggingOptionsPayload	LoggingOptionsPayload	The logging options payload.
roleArn	string	The ARN of the IAM role that grants access.
logLevel	string	The log level.  enum: DEBUG   INFO   ERROR   WARN   DISABLED

### Output

None

### Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

# SetV2LogLevel

Sets the logging level.

## Synopsis

```
aws iot set-v2-logging-level \
--log-target <value> \
--log-level <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "logTarget": {
    "targetType": "string",
    "targetName": "string"
  },
  "logLevel": "string"
}
```

## cli-input-json fields

Name	Type	Description
logTarget	LogTarget	The log target.
targetType	string	The target type.  enum: DEFAULT   THING_GROUP
targetName	string	The target name.
logLevel	string	The log level.  enum: DEBUG   INFO   ERROR   WARN   DISABLED

## Output

None

## Errors

**InternalException**

An unexpected error has occurred.

**NotConfiguredException**

The resource is not configured.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

# SetV2LoggingOptions

Sets the logging options for the V2 logging service.

## Synopsis

```
aws iot set-v2-logging-options \
[--role-arn <value>] \
[--default-log-level <value>] \
[--disable-all-logs | --no-disable-all-logs] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "roleArn": "string",
  "defaultLogLevel": "string",
  "disableAllLogs": "boolean"
}
```

## cli-input-json fields

Name	Type	Description
roleArn	string	The ARN of the role that allows IoT to write to Cloudwatch logs.
defaultLogLevel	string	The default logging level.  enum: DEBUG   INFO   ERROR   WARN   DISABLED
disableAllLogs	boolean	If true all logs are disabled. The default is false.

## Output

None

## Errors

**InternalException**

An unexpected error has occurred.

**InvalidRequestException**

The contents of the request were invalid.

**ServiceUnavailableException**

The service is temporarily unavailable.

# StartNextPendingJobExecution

Gets and starts the next pending (status IN\_PROGRESS or QUEUED) job execution for a thing.

## Synopsis

```
aws iot-jobs-data start-next-pending-job-execution \
    --thing-name <value> \
    [--status-details <value>] \
    [--step-timeout-in-minutes <value>] \
    [--cli-input-json <value>] \
    [--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingName": "string",
  "statusDetails": {
    "string": "string"
  },
  "stepTimeoutInMinutes": "long"
}
```

## cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing associated with the device.
statusDetails	map	A collection of name/value pairs that describe the status of the job execution. If not specified, the statusDetails are unchanged.
stepTimeoutInMinutes	long	<p>Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by calling <code>UpdateJobExecution</code>, setting the status to <code>IN_PROGRESS</code>, and specifying a new timeout value in field <code>stepTimeoutInMinutes</code>) the job execution status will be automatically set to <code>TIMED_OUT</code>. Note that setting the step timeout has no effect on the in progress timeout that may have been specified when the job was created (<code>CreateJob</code> using field <code>timeoutConfig</code>).</p> <p>Valid values for this parameter range from 1 to 10080 (1 minute to 7 days).</p>

## Output

```
{
  "execution": {
    "jobId": "string",
    "thingName": "string",
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "queuedAt": "long",
    "startedAt": "long",
    "lastUpdatedAt": "long",
    "approximateSecondsBeforeTimedOut": "long",
    "versionNumber": "long",
    "executionNumber": "long",
    "jobDocument": "string"
  }
}
```

## CLI output fields

Name	Type	Description
execution	JobExecution	A JobExecution object.
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier you assigned to this job when it was created.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_-]+	The name of the thing that is executing the job.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The status of the job execution. Can be one of: "QUEUED", "IN_PROGRESS", "FAILED", "SUCCESS", "CANCELED", "TIMED_OUT", "REJECTED", or "REMOVED".  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
statusDetails	map	A collection of name/value pairs that describe the status of the job execution.
queuedAt	long	The time, in seconds since the epoch, when the job execution was enqueued.
startedAt	long	The time, in seconds since the epoch, when the job execution was started.

Name	Type	Description
lastUpdatedAt	long	The time, in seconds since the epoch, when the job execution was last updated.
approximateSecondsBeforeTimedOut	long	The estimated number of seconds that remain before the job execution status will be changed to <code>TIMED_OUT</code> . The actual job execution timeout can occur up to 60 seconds later than the estimated duration.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
executionNumber	long	A number that identifies a particular job execution on a particular device. It can be used later in commands that return or update job execution information.
jobDocument	string length- max:32768	The content of the job document.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### ResourceNotFoundException

The specified resource does not exist.

### ThrottlingException

The rate exceeds the limit.

### ServiceUnavailableException

The service is temporarily unavailable.

### CertificateValidationException

The certificate is invalid.

## StartOnDemandAuditTask

Starts an on-demand Device Defender audit.

### Synopsis

```
aws iot start-on-demand-audit-task \
```

```
--target-check-names <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "targetCheckNames": [
    "string"
  ]
}
```

#### cli-input-json fields

Name	Type	Description
targetCheckNames	list member: AuditCheckName	Which checks are performed during the audit. The checks you specify must be enabled for your account or an exception occurs. Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.

#### Output

```
{
  "taskId": "string"
}
```

#### CLI output fields

Name	Type	Description
taskId	string length- max:40 min:1 pattern: [a-zA-Z0-9-]+	The ID of the on-demand audit you started.

#### Errors

##### InvalidRequestException

The contents of the request were invalid.

##### ThrottlingException

The rate exceeds the limit.

##### InternalFailureException

An unexpected error has occurred.

### LimitExceeded**Exception**

A limit has been exceeded.

## StartThingRegistrationTask

Creates a bulk thing provisioning task.

### Synopsis

```
aws iot start-thing-registration-task \
--template-body <value> \
--input-file-bucket <value> \
--input-file-key <value> \
--role-arn <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "templateBody": "string",
  "inputFileBucket": "string",
  "inputFileKey": "string",
  "roleArn": "string"
}
```

### cli-input-json fields

Name	Type	Description
templateBody	string	The provisioning template.
inputFileBucket	string  length- max:256 min:3  pattern: [a-zA-Z0-9._-]+	The S3 bucket that contains the input file.
inputFileKey	string  length- max:1024 min:1  pattern: [a-zA-Z0-9!_.*'()-/]+	The name of input file within the S3 bucket. This file contains a newline delimited JSON file. Each line contains the parameter values to provision one device (thing).
roleArn	string  length- max:2048 min:20	The IAM role ARN that grants permission the input file.

### Output

```
{
  "taskId": "string"
}
```

### CLI output fields

Name	Type	Description
taskId	string length- max:40	The bulk thing provisioning task ID.

### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`InternalFailureException`

An unexpected error has occurred.

## StopThingRegistrationTask

Cancels a bulk thing provisioning task.

### Synopsis

```
aws iot stop-thing-registration-task \
--task-id <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "taskId": "string"
}
```

### cli-input-json fields

Name	Type	Description
taskId	string length- max:40	The bulk thing provisioning task ID.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

## TagResource

Adds to or modifies the tags of the given resource. Tags are metadata which can be used to manage a resource.

### Synopsis

```
aws iot tag-resource \
--resource-arn <value> \
--tags <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "resourceArn": "string",
  "tags": [
    {
      "Key": "string",
      "Value": "string"
    }
  ]
}
```

### cli-input-json fields

Name	Type	Description
resourceArn	string	The ARN of the resource.
tags	list member: Tag java class: java.util.List	The new or modified tags for the resource.
Key	string	The tag's key.
Value	string	The tag's value.

**Output**

None

**Errors**

`InvalidRequestException`

The contents of the request were invalid.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`LimitExceededException`

A limit has been exceeded.

## TestAuthorization

Tests if a specified principal is authorized to perform an AWS IoT action on a specified resource. Use this to test and debug the authorization behavior of devices that connect to the AWS IoT device gateway.

**Synopsis**

```
aws iot test-authorization \
[--principal <value>] \
[--cognito-identity-pool-id <value>] \
--auth-infos <value> \
[--client-id <value>] \
[--policy-names-to-add <value>] \
[--policy-names-to-skip <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "principal": "string",
  "cognitoIdentityPoolId": "string",
  "authInfos": [
    {
      "actionType": "string",
      "resources": [
        "string"
      ]
    }
  ],
  "clientId": "string",
  "policyNamesToAdd": [
    "string"
  ],
  "policyNamesToSkip": [
    "string"
  ]}
```

```
    ]
}
```

### cli-input-json fields

Name	Type	Description
principal	string	The principal.
cognitoIdentityPoolId	string	The Cognito identity pool ID.
authInfos	list member: AuthInfo	A list of authorization info objects. Simulating authorization will create a response for each authInfo object in the list.
actionType	string	The type of action for which the principal is being authorized.  enum: PUBLISH   SUBSCRIBE   RECEIVE   CONNECT
resources	list member: Resource	The resources for which the principal is being authorized to perform the specified action.
clientId	string	The MQTT client ID.
policyNamesToAdd	list member: PolicyName java class: java.util.List	When testing custom authorization, the policies specified here are treated as if they are attached to the principal being authorized.
policyNamesToSkip	list member: PolicyName java class: java.util.List	When testing custom authorization, the policies specified here are treated as if they are not attached to the principal being authorized.

### Output

```
{
  "authResults": [
    {
      "authInfo": {
        "actionType": "string",
        "resources": [
          "string"
        ]
      },
      "allowed": {
        "policies": [
          {
            "policyName": "string",
            "policyArn": "string"
          }
        ]
      }
    }
  ]
}
```

```

    "denied": {
      "implicitDeny": {
        "policies": [
          {
            "policyName": "string",
            "policyArn": "string"
          }
        ]
      },
      "explicitDeny": {
        "policies": [
          {
            "policyName": "string",
            "policyArn": "string"
          }
        ]
      }
    },
    "authDecision": "string",
    "missingContextValues": [
      "string"
    ]
  }
]
}
}

```

### CLI output fields

Name	Type	Description
authResults	list  member: AuthResult	The authentication results.
authInfo	AuthInfo	Authorization information.
actionType	string  enum: PUBLISH   SUBSCRIBE   RECEIVE   CONNECT	The type of action for which the principal is being authorized.
resources	list  member: Resource	The resources for which the principal is being authorized to perform the specified action.
allowed	Allowed	The policies and statements that allowed the specified action.
policies	list  member: Policy  java class: java.util.List	A list of policies that allowed the authentication.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.

Name	Type	Description
denied	Denied	The policies and statements that denied the specified action.
implicitDeny	ImplicitDeny	Information that implicitly denies the authorization. When a policy doesn't explicitly deny or allow an action on a resource it is considered an implicit deny.
policies	list member: Policy java class: java.util.List	Policies that don't contain a matching allow or deny statement for the specified action on the specified resource.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
explicitDeny	ExplicitDeny	Information that explicitly denies the authorization.
policies	list member: Policy java class: java.util.List	The policies that denied the authorization.
policyName	string  length- max:128 min:1  pattern: [w+=,.@-]+	The policy name.
policyArn	string	The policy ARN.
authDecision	string	The final authorization decision of this scenario. Multiple statements are taken into account when determining the authorization decision. An explicit deny statement can override multiple allow statements.  enum: ALLOWED   EXPLICIT_DENY   IMPLICIT_DENY
missingContextValues	list member: MissingContextValue java class: java.util.List	Contains any missing context values found while evaluating policy.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`LimitExceededException`

A limit has been exceeded.

# TestInvokeAuthorizer

Tests a custom authorization behavior by invoking a specified custom authorizer. Use this to test and debug the custom authorization behavior of devices that connect to the AWS IoT device gateway.

## Synopsis

```
aws iot test-invoke-authorizer \
--authorizer-name <value> \
--token <value> \
--token-signature <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{  
    "authorizerName": "string",  
    "token": "string",  
    "tokenSignature": "string"  
}
```

## `cli-input-json` fields

Name	Type	Description
authorizerName	string  length- max:128 min:1	The custom authorizer name.

Name	Type	Description
	pattern: [w=,@-]+	
token	string length- max:6144 min:1	The token returned by your custom authentication service.
tokenSignature	string length- max:2560 min:1 pattern: [A-Za-z0-9+/]{0,2}	The signature made with the token and your custom authentication service's private key.

## Output

```
{
  "isAuthenticated": "boolean",
  "principalId": "string",
  "policyDocuments": [
    "string"
  ],
  "refreshAfterInSeconds": "integer",
  "disconnectAfterInSeconds": "integer"
}
```

## CLI output fields

Name	Type	Description
isAuthenticated	boolean	True if the token is authenticated, otherwise false.
principalId	string length- max:128 min:1 pattern: [a-zA-Z0-9]+	The principal ID.
policyDocuments	list member: PolicyDocument	IAM policy documents.
refreshAfterInSeconds	integer	The number of seconds after which the temporary credentials are refreshed.
disconnectAfterInSeconds	integer	The number of seconds after which the connection is terminated.

## Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

**InvalidResponseException**

The response is invalid.

## TransferCertificate

Transfers the specified certificate to the specified AWS account.

You can cancel the transfer until it is acknowledged by the recipient.

No notification is sent to the transfer destination's account. It is up to the caller to notify the transfer target.

The certificate being transferred must not be in the ACTIVE state. You can use the UpdateCertificate API to deactivate it.

The certificate must not have any policies attached to it. You can use the DetachPrincipalPolicy API to detach them.

### Synopsis

```
aws iot transfer-certificate \
--certificate-id <value> \
--target-aws-account <value> \
[--transfer-message <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "certificateId": "string",  
    "targetAwsAccount": "string",  
    "transferMessage": "string"  
}
```

### cli-input-json fields

Name	Type	Description
certificateId	string length- max:64 min:64 pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)

Name	Type	Description
targetAwsAccount	string  length- max:12 min:12  pattern: [0-9]+	The AWS account.
transferMessage	string  length- max:128	The transfer message.

#### Output

```
{
  "transferredCertificateArn": "string"
}
```

#### CLI output fields

Name	Type	Description
transferredCertificateArn	string	The ARN of the certificate.

#### Errors

##### InvalidRequestException

The contents of the request were invalid.

##### ResourceNotFoundException

The specified resource does not exist.

##### CertificateStateException

The certificate operation is not allowed.

##### TransferConflictException

You can't transfer the certificate because authorization policies are still attached.

##### ThrottlingException

The rate exceeds the limit.

##### UnauthorizedException

You are not authorized to perform this operation.

##### ServiceUnavailableException

The service is temporarily unavailable.

##### InternalFailureException

An unexpected error has occurred.

## UntagResource

Removes the given tags (metadata) from the resource.

### Synopsis

```
aws iot untag-resource \
--resource-arn <value> \
--tag-keys <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "resourceArn": "string",
  "tagKeys": [
    "string"
  ]
}
```

### cli-input-json fields

Name	Type	Description
resourceArn	string	The ARN of the resource.
tagKeys	list member: TagKey java class: java.util.List	A list of the keys of the tags to be removed from the resource.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

## UpdateAccountAuditConfiguration

Configures or reconfigures the Device Defender audit settings for this account. Settings include how audit notifications are sent and which audit checks are enabled or disabled.

### Synopsis

```
aws iot update-account-audit-configuration \
[--role-arn <value>] \
[--audit-notification-target-configurations <value>] \
```

```
[--audit-check-configurations <value>] \n
[--cli-input-json <value>] \n
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "roleArn": "string",
  "auditNotificationTargetConfigurations": {
    "string": {
      "targetArn": "string",
      "roleArn": "string",
      "enabled": "boolean"
    }
  },
  "auditCheckConfigurations": {
    "string": {
      "enabled": "boolean"
    }
  }
}
```

#### cli-input-json fields

Name	Type	Description
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to AWS IoT to access information about your devices, policies, certificates and other items as necessary when performing an audit.
auditNotificationTargetConfigurations	map	Information about the targets to which audit notifications are sent.
targetArn	string	The ARN of the target (SNS topic) to which audit notifications are sent.
roleArn	string length- max:2048 min:20	The ARN of the role that grants permission to send notifications to the target.
enabled	boolean	True if notifications to the target are enabled.
auditCheckConfigurations	map	<p>Specifies which audit checks are enabled and disabled for this account. Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are currently enabled.</p> <p>Note that some data collection may begin immediately when certain checks are enabled. When a check is disabled, any</p>

Name	Type	Description
		<p>data collected so far in relation to the check is deleted.</p> <p>You cannot disable a check if it is used by any scheduled audit. You must first delete the check from the scheduled audit or delete the scheduled audit itself.</p> <p>On the first call to <code>UpdateAccountAuditConfiguration</code> this parameter is required and must specify at least one enabled check.</p>
enabled	boolean	True if this audit check is enabled for this account.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

## UpdateAuthorizer

Updates an authorizer.

#### Synopsis

```
aws iot update-authorizer \
--authorizer-name <value> \
[--authorizer-function-arn <value>] \
[--token-key-name <value>] \
[--token-signing-public-keys <value>] \
[--status <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "authorizerName": "string",
  "authorizerFunctionArn": "string",
  "tokenKeyName": "string",
  "tokenSigningPublicKeys": {
```

```

    "string": "string"
},
"status": "string"
}
}
```

### cli-input-json fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.
authorizerFunctionArn	string	The ARN of the authorizer's Lambda function.
tokenKeyName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The key used to extract the token from the HTTP headers.
tokenSigningPublicKeys	map	The public keys used to verify the token signature.
status	string  enum: ACTIVE   INACTIVE	The status of the update authorizer request.

### Output

```
{
  "authorizerName": "string",
  "authorizerArn": "string"
}
```

### CLI output fields

Name	Type	Description
authorizerName	string  length- max:128 min:1  pattern: [w=,@-]+	The authorizer name.
authorizerArn	string	The authorizer ARN.

### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`InvalidRequestException`

The contents of the request were invalid.

**LimitExceeded**

A limit has been exceeded.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## UpdateBillingGroup

Updates information about the billing group.

### Synopsis

```
aws iot update-billing-group \
--billing-group-name <value> \
--billing-group-properties <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "billingGroupName": "string",
  "billingGroupProperties": {
    "billingGroupDescription": "string"
  },
  "expectedVersion": "long"
}
```

### cli-input-json fields

Name	Type	Description
billingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the billing group.
billingGroupProperties	BillingGroupProperties	The properties of the billing group.
billingGroupDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The description of the billing group.

Name	Type	Description
expectedVersion	long	The expected version of the billing group. If the version of the billing group does not match the expected version specified in the request, the <code>UpdateBillingGroup</code> request is rejected with a <code>VersionConflictException</code> .

#### Output

```
{
  "version": "long"
}
```

#### CLI output fields

Name	Type	Description
version	long	The latest version of the billing group.

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`VersionConflictException`

An exception thrown when the version of a thing passed to a command is different than the version specified with the `--version` parameter.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## UpdateCACertificate

Updates a registered CA certificate.

#### Synopsis

```
aws iot update-ca-certificate \
--certificate-id <value> \
[--new-status <value>] \
[--new-auto-registration-status <value>] \
[--registration-config <value>]
```

```
[--remove-auto-registration | --no-remove-auto-registration] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "certificateId": "string",
  "newStatus": "string",
  "newAutoRegistrationStatus": "string",
  "registrationConfig": {
    "templateBody": "string",
    "roleArn": "string"
  },
  "removeAutoRegistration": "boolean"
}
```

#### cli-input-json fields

Name	Type	Description
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The CA certificate identifier.
newStatus	string	The updated status of the CA certificate.  <b>Note:</b> The status value REGISTER_INACTIVE is deprecated and should not be used.  enum: ACTIVE   INACTIVE
newAutoRegistrationStatus	string	The new value for the auto registration status. Valid values are: "ENABLE" or "DISABLE".  enum: ENABLE   DISABLE
registrationConfig	RegistrationConfig	Information about the registration configuration.
templateBody	string	The template body.
roleArn	string  length- max:2048 min:20	The ARN of the role.
removeAutoRegistration	boolean	If true, removes auto registration.

#### Output

None

#### Errors

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## UpdateCertificate

Updates the status of the specified certificate. This operation is idempotent.

Moving a certificate from the ACTIVE state (including REVOKED) will not disconnect currently connected devices, but these devices will be unable to reconnect.

The ACTIVE state is required to authenticate devices connecting to AWS IoT using a certificate.

### Synopsis

```
aws iot update-certificate \
--certificate-id <value> \
--new-status <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "certificateId": "string",
  "newStatus": "string"
}
```

### cli-input-json fields

Name	Type	Description
certificateId	string  length- max:64 min:64  pattern: (0x)?[a-fA-F0-9]+	The ID of the certificate. (The last part of the certificate ARN contains the certificate ID.)
newStatus	string	The new status.  <b>Note:</b> Setting the status to PENDING_TRANSFER will result in an exception being thrown.

Name	Type	Description
		<p>PENDING_TRANSFER is a status used internally by AWS IoT. It is not intended for developer use.</p> <p><b>Note:</b> The status value REGISTER_INACTIVE is deprecated and should not be used.</p> <p>enum: ACTIVE   INACTIVE   REVOKED   PENDING_TRANSFER   REGISTER_INACTIVE   PENDING_ACTIVATION</p>

#### Output

None

#### Errors

`ResourceNotFoundException`

The specified resource does not exist.

`CertificateStateException`

The certificate operation is not allowed.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## UpdateDynamicThingGroup

Updates a dynamic thing group.

#### Synopsis

```
aws iot update-dynamic-thing-group \
--thing-group-name <value> \
--thing-group-properties <value> \
[--expected-version <value>] \
[--index-name <value>] \
[--query-string <value>] \
[--query-version <value>] \
[--cli-input-json <value>]
```

[--generate-cli-skeleton]

#### cli-input-json format

```
{
  "thingGroupName": "string",
  "thingGroupProperties": {
    "thingGroupDescription": "string",
    "attributePayload": {
      "attributes": {
        "string": "string"
      },
      "merge": "boolean"
    }
  },
  "expectedVersion": "long",
  "indexName": "string",
  "queryString": "string",
  "queryVersion": "string"
}
```

#### cli-input-json fields

Name	Type	Description
thingGroupName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the dynamic thing group to update.
thingGroupProperties	ThingGroupProperties	The dynamic thing group properties to update.
thingGroupDescription	string  length- max:2028  pattern: [\p{Graph}]*	The thing group description.
attributePayload	AttributePayload	The thing group attributes in JSON format.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\ "attributes\ ":" {\ \"string1\ ":" \"string2\ "}</code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.  To remove an attribute, call UpdateThing with an empty attribute value.

Name	Type	Description
		<b>Note</b> The <code>merge</code> attribute is only valid when calling <code>UpdateThing</code> .
expectedVersion	long	The expected version of the dynamic thing group to update.
indexName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The dynamic thing group index to update. <b>Note</b> Currently one index is supported: 'AWS_Things'.
queryString	string  length- min:1	The dynamic thing group search query string to update.
queryVersion	string	The dynamic thing group query version to update. <b>Note</b> Currently one query version is supported: "2017-09-30". If not specified, the query version defaults to this value.

## Output

```
{
  "version": "long"
}
```

## CLI output fields

Name	Type	Description
version	long	The dynamic thing group version.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### VersionConflictException

An exception thrown when the version of a thing passed to a command is different than the version specified with the `--version` parameter.

### ThrottlingException

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

**ResourceNotFoundException**

The specified resource does not exist.

**InvalidQueryException**

The query is invalid.

## UpdateEventConfigurations

Updates the event configurations.

### Synopsis

```
aws iot update-event-configurations \
[--event-configurations <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "eventConfigurations": {
    "string": {
      "Enabled": "boolean"
    }
  }
}
```

### cli-input-json fields

Name	Type	Description
eventConfigurations	map	The new event configuration values.
Enabled	boolean	True to enable the configuration.

### Output

None

### Errors

**InvalidRequestException**

The contents of the request were invalid.

**InternalFailureException**

An unexpected error has occurred.

**ThrottlingException**

The rate exceeds the limit.

# UpdateIndexingConfiguration

Updates the search configuration.

## Synopsis

```
aws iot update-indexing-configuration \
[--thing-indexing-configuration <value>] \
[--thing-group-indexing-configuration <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "thingIndexingConfiguration": {
    "thingIndexingMode": "string",
    "thingConnectivityIndexingMode": "string"
  },
  "thingGroupIndexingConfiguration": {
    "thingGroupIndexingMode": "string"
  }
}
```

## cli-input-json fields

Name	Type	Description
thingIndexingConfiguration	ThingIndexingConfiguration	Thing indexing configuration.
thingIndexingMode	string	<p>Thing indexing mode. Valid values are:</p> <ul style="list-style-type: none"> <li>• REGISTRY – Your thing index contains registry data only.</li> <li>• REGISTRY_AND_SHADOW - Your thing index contains registry and shadow data.</li> <li>• OFF - Thing indexing is disabled.</li> </ul> <p>enum: OFF   REGISTRY   REGISTRY_AND_SHADOW</p>
thingConnectivityIndexingMode	string	<p>Thing connectivity indexing mode. Valid values are:</p> <ul style="list-style-type: none"> <li>• STATUS – Your thing index contains connectivity status. To enable thing connectivity indexing, thingIndexMode must not be set to OFF.</li> <li>• OFF - Thing connectivity status indexing is disabled.</li> </ul> <p>enum: OFF   STATUS</p>

Name	Type	Description
thingGroupIndexingConfiguration	ThingGroupIndexingConfiguration	Thing group indexing configuration.
thingGroupIndexingMode	string	Thing group indexing mode. enum: OFF   ON

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

## UpdateJob

Updates supported fields of the specified job.

#### Synopsis

```
aws iot update-job \
--job-id <value> \
[--description <value>] \
[--presigned-url-config <value>] \
[--job-executions-rollout-config <value>] \
[--abort-config <value>] \
[--timeout-config <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "jobId": "string",
  "description": "string",
  "presignedUrlConfig": {
    "roleArn": "string",
    "expiresInSec": "long"
  },
  "jobExecutionsRolloutConfig": {
    "maximumPerMinute": "integer",
```

```

    "exponentialRate": {
        "baseRatePerMinute": "integer",
        "incrementFactor": "double",
        "rateIncreaseCriteria": {
            "numberOfNotifiedThings": "integer",
            "numberOfSucceededThings": "integer"
        }
    },
    "abortConfig": {
        "criteriaList": [
            {
                "failureType": "string",
                "action": "string",
                "thresholdPercentage": "double",
                "minNumberOfExecutedThings": "integer"
            }
        ]
    },
    "timeoutConfig": {
        "inProgressTimeoutInMinutes": "long"
    }
}

```

#### **cli-input-json fields**

Name	Type	Description
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The ID of the job to be updated.
description	string  length- max:2028  pattern: [^\p{C}]+	A short text description of the job.
presignedUrlConfig	PresignedUrlConfig	Configuration information for pre-signed S3 URLs.
roleArn	string  length- max:2048 min:20	The ARN of an IAM role that grants permission to download files from the S3 bucket where the job data/updates are stored. The role must also grant permission for IoT to download the files.
expiresInSec	long  range- max:3600 min:60	How long (in seconds) pre-signed URLs are valid. Valid values are 60 - 3600, the default value is 3600 seconds. Pre-signed URLs are generated when Jobs receives an MQTT request for the job document.
jobExecutionsRolloutConfig	JobExecutionsRolloutConfig	Allows you to create a staged rollout of the job.

Name	Type	Description
maximumPerMinute	integer range- min:1	The maximum number of things that will be notified of a pending job, per minute. This parameter allows you to create a staged rollout.
exponentialRate	ExponentialRolloutRate	The rate of increase for a job rollout. This parameter allows you to define an exponential rate for a job rollout.
baseRatePerMinute	integer range- max:1000 min:1	The minimum number of things that will be notified of a pending job, per minute at the start of job rollout. This parameter allows you to define the initial rate of rollout.
rateIncreaseCriteria	RateIncreaseCriteria	The criteria to initiate the increase in rate of rollout for a job.  AWS IoT supports up to one digit after the decimal (for example, 1.5, but not 1.55).
numberOfNotifiedThings	integer range- min:1	The threshold for number of notified things that will initiate the increase in rate of rollout.
numberOfSucceededThings	integer range- min:1	The threshold for number of succeeded things that will initiate the increase in rate of rollout.
abortConfig	AbortConfig	Allows you to create criteria to abort a job.
criteriaList	list member: AbortCriteria java class: java.util.List	The list of abort criteria to define rules to abort the job.
failureType	string	The type of job execution failure to define a rule to initiate a job abort.  enum: FAILED   REJECTED   TIMED_OUT   ALL
action	string	The type of abort action to initiate a job abort.  enum: CANCEL

Name	Type	Description
minNumberOfExecutedThings	integer range- min:1	Minimum number of executed things before evaluating an abort rule.
timeoutConfig	TimeoutConfig	Specifies the amount of time each device has to finish its execution of the job. The timer is started when the job execution status is set to IN_PROGRESS. If the job execution status is not set to another terminal state before the time expires, it will be automatically set to TIMED_OUT.
inProgressTimeoutInMinutes	long	Specifies the amount of time, in minutes, this device has to finish execution of this job. The timeout interval can be anywhere between 1 minute and 7 days (1 to 10080 minutes). The in progress timer can't be updated and will apply to all job executions for the job. Whenever a job execution remains in the IN_PROGRESS status for longer than this interval, the job execution will fail and switch to the terminal TIMED_OUT status.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`ServiceUnavailableException`

The service is temporarily unavailable.

## UpdateJobExecution

Updates the status of a job execution.

#### Synopsis

```
aws iot-jobs-data update-job-execution \
--job-id <value> \
--thing-name <value> \
--status <value> \
[--status-details <value>] \
[--step-timeout-in-minutes <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>] \
[--generate-cli-skeleton]
```

#### cli-input-json format

```
{
  "jobId": "string",
  "thingName": "string",
  "status": "string",
  "statusDetails": {
    "string": "string"
  },
  "stepTimeoutInMinutes": "long",
  "expectedVersion": "long",
  "includeJobExecutionState": "boolean",
  "includeJobDocument": "boolean",
  "executionNumber": "long"
}
```

#### cli-input-json fields

Name	Type	Description
jobId	string  length- max:64 min:1  pattern: [a-zA-Z0-9_-]+	The unique identifier assigned to this job when it was created.
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the thing associated with the device.
status	string  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED	The new status for the job execution (IN_PROGRESS, FAILED, SUCCESS, or REJECTED). This must be specified on every update.
statusDetails	map	Optional. A collection of name/value pairs that describe the status of the job execution. If

Name	Type	Description
		not specified, the statusDetails are unchanged.
stepTimeoutInMinutes	long	<p>Specifies the amount of time this device has to finish execution of this job. If the job execution status is not set to a terminal state before this timer expires, or before the timer is reset (by again calling <code>UpdateJobExecution</code>, setting the status to <code>IN_PROGRESS</code>, and specifying a new timeout value in this field) the job execution status will be automatically set to <code>TIMED_OUT</code>. Note that setting or resetting the step timeout has no effect on the in progress timeout that may have been specified when the job was created (<code>CreateJob</code> using field <code>timeoutConfig</code>).</p> <p>Valid values for this parameter range from 1 to 10080 (1 minute to 7 days). A value of -1 is also valid and will cancel the current step timer (created by an earlier use of <code>UpdateJobExecutionRequest</code>).</p>
expectedVersion	long	<p>Optional. The expected current version of the job execution. Each time you update the job execution, its version is incremented. If the version of the job execution stored in Jobs does not match, the update is rejected with a <code>VersionMismatch</code> error, and an <code>ErrorResponse</code> that contains the current job execution status data is returned. (This makes it unnecessary to perform a separate <code>DescribeJobExecution</code> request in order to obtain the job execution status data.)</p>
includeJobExecutionState	boolean	<p>Optional. When included and set to true, the response contains the <code>JobExecutionState</code> data. The default is false.</p>
includeJobDocument	boolean	<p>Optional. When set to true, the response contains the job document. The default is false.</p>

Name	Type	Description
executionNumber	long	Optional. A number that identifies a particular job execution on a particular device.

#### Output

```
{
  "executionState": {
    "status": "string",
    "statusDetails": {
      "string": "string"
    },
    "versionNumber": "long"
  },
  "jobDocument": "string"
}
```

#### CLI output fields

Name	Type	Description
executionState	JobExecutionState	A JobExecutionState object.
status	string	The status of the job execution. Can be one of: "QUEUED", "IN_PROGRESS", "FAILED", "SUCCESS", "CANCELED", "TIMED_OUT", "REJECTED", or "REMOVED".  enum: QUEUED   IN_PROGRESS   SUCCEEDED   FAILED   TIMED_OUT   REJECTED   REMOVED   CANCELED
statusDetails	map	A collection of name/value pairs that describe the status of the job execution.
versionNumber	long	The version of the job execution. Job execution versions are incremented each time they are updated by a device.
jobDocument	string length- max:32768	The contents of the Job Documents.

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**ServiceUnavailableException**

The service is temporarily unavailable.

**CertificateValidationException**

The certificate is invalid.

**InvalidStateTransitionException**

An update attempted to change the job execution to a state that is invalid because of the job execution's current state (for example, an attempt to change a request in state SUCCESS to state IN\_PROGRESS). In this case, the body of the error message also contains the executionState field.

## UpdateRoleAlias

Updates a role alias.

### Synopsis

```
aws iot update-role-alias \
--role-alias <value> \
[--role-arn <value>] \
[--credential-duration-seconds <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{  
    "roleAlias": "string",  
    "roleArn": "string",  
    "credentialDurationSeconds": "integer"  
}
```

### cli-input-json fields

Name	Type	Description
roleAlias	string  length- max:128 min:1  pattern: [w=,@-]+	The role alias to update.
roleArn	string  length- max:2048 min:20	The role ARN.
credentialDurationSeconds	integer  range- max:3600 min:900	The number of seconds the credential will be valid.

### Output

```
{
```

```

    "roleAlias": "string",
    "roleAliasArn": "string"
}

```

**CLI output fields**

Name	Type	Description
roleAlias	string  length- max:128 min:1  pattern: [w=,@-]+	The role alias.
roleAliasArn	string	The role alias ARN.

**Errors****ResourceNotFoundException**

The specified resource does not exist.

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## UpdateScheduledAudit

Updates a scheduled audit, including what checks are performed and how often the audit takes place.

**Synopsis**

```
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]
```

**cli-input-json format**

```
{
  "frequency": "string",
  "dayOfMonth": "string",
  "dayOfWeek": "string",
```

```

    "targetCheckNames": [
        "string"
    ],
    "scheduledAuditName": "string"
}

```

#### cli-input-json fields

Name	Type	Description
frequency	string	How often the scheduled audit takes place. Can be one of "DAILY", "WEEKLY", "BIWEEKLY" or "MONTHLY". The actual start time of each audit is determined by the system.  enum: DAILY   WEEKLY   BIWEEKLY   MONTHLY
dayOfMonth	string  pattern: ^([1-9] [12][0-9] 3[01])\$ ^LAST\$	The day of the month on which the scheduled audit takes place. Can be "1" through "31" or "LAST". This field is required if the "frequency" parameter is set to "MONTHLY". If days 29-31 are specified, and the month does not have that many days, the audit takes place on the "LAST" day of the month.
dayOfWeek	string	The day of the week on which the scheduled audit takes place. Can be one of "SUN", "MON", "TUE", "WED", "THU", "FRI" or "SAT". This field is required if the "frequency" parameter is set to "WEEKLY" or "BIWEEKLY".  enum: SUN   MON   TUE   WED   THU   FRI   SAT
targetCheckNames	list  member: AuditCheckName	Which checks are performed during the scheduled audit. Checks must be enabled for your account. (Use <a href="#">DescribeAccountAuditConfiguration</a> to see the list of all checks including those that are enabled or <a href="#">UpdateAccountAuditConfiguration</a> to select which checks are enabled.)
scheduledAuditName	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The name of the scheduled audit. (Max. 128 chars)

## Output

```
{
  "scheduledAuditArn": "string"
}
```

## CLI output fields

Name	Type	Description
scheduledAuditArn	string	The ARN of the scheduled audit.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

# UpdateSecurityProfile

Updates a Device Defender security profile.

## Synopsis

```
aws iot update-security-profile \
--security-profile-name <value> \
[--security-profile-description <value>] \
[--behaviors <value>] \
[--alert-targets <value>] \
[--additional-metrics-to-retain <value>] \
[--delete-behaviors | --no-delete-behaviors] \
[--delete-alert-targets | --no-delete-alert-targets] \
[--delete-additional-metrics-to-retain | --no-delete-additional-metrics-to-retain] \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "securityProfileName": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
```

```

    "criteria": [
        "comparisonOperator": "string",
        "value": {
            "count": "long",
            "cidrs": [
                "string"
            ],
            "ports": [
                "integer"
            ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
            "statistic": "string"
        }
    }
],
"alertTargets": {
    "string": {
        "alertTargetArn": "string",
        "roleArn": "string"
    }
},
"additionalMetricsToRetain": [
    "string"
],
"deleteBehaviors": "boolean",
"deleteAlertTargets": "boolean",
"deleteAdditionalMetricsToRetain": "boolean",
"expectedVersion": "long"
}

```

#### **cli-input-json fields**

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the security profile you want to update.
securityProfileDescription	string  length- max:1000  pattern: [\p{Graph}]*	A description of the security profile.
behaviors	list  member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.

Name	Type	Description
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the metric.
count	long  range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cidrs	list  member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list  member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.

Name	Type	Description
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the profile's behaviors but it is also retained for any metric specified here.

Name	Type	Description
deleteBehaviors	boolean	If true, delete all behaviors defined for this security profile. If any behaviors are defined in the current invocation an exception occurs.
deleteAlertTargets	boolean	If true, delete all alertTargets defined for this security profile. If any alertTargets are defined in the current invocation an exception occurs.
deleteAdditionalMetricsToRetain	boolean	If true, delete all additionalMetricsToRetain defined for this security profile. If any additionalMetricsToRetain are defined in the current invocation an exception occurs.
expectedVersion	long	The expected version of the security profile. A new version is generated whenever the security profile is updated. If you specify a value that is different than the actual version, a VersionConflictException is thrown.

## Output

```
{
  "securityProfileName": "string",
  "securityProfileArn": "string",
  "securityProfileDescription": "string",
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  }
}
```

```
],
"alertTargets": {
  "string": {
    "alertTargetArn": "string",
    "roleArn": "string"
  }
},
"additionalMetricsToRetain": [
  "string"
],
"version": "long",
"creationDate": "timestamp",
"lastModifiedDate": "timestamp"
}
```

## CLI output fields

Name	Type	Description
securityProfileName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the security profile that was updated.
securityProfileArn	string	The ARN of the security profile that was updated.
securityProfileDescription	string  length- max:1000  pattern: [\p{Graph} ]*	The description of the security profile.
behaviors	list  member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.
comparisonOperator	string  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set	The operator that relates the thing measured (metric) to the criteria (containing a value or statisticalThreshold).

Name	Type	Description
value	MetricValue	The value to be compared with the metric.
count	long range- min:0	If the comparisonOperator calls for a numeric value, use this to specify that numeric value to be compared with the metric.
cidrs	list member: Cidr	If the comparisonOperator calls for a set of CIDRs, use this to specify that set to be compared with the metric.
ports	list member: Port	If the comparisonOperator calls for a set of ports, use this to specify that set to be compared with the metric.
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, NUM_MESSAGES_SENT). For a statisticalThreshold metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.

Name	Type	Description
statistic	string  pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.
alertTargets	map	Where the alerts are sent. (Alerts are always sent to the console.)
alertTargetArn	string	The ARN of the notification target to which alerts are sent.
roleArn	string  length- max:2048 min:20	The ARN of the role that grants permission to send alerts to the notification target.
additionalMetricsToRetain	list  member: BehaviorMetric	A list of metrics whose data is retained (stored). By default, data is retained for any metric used in the security profile's behaviors but it is also retained for any metric specified here.
version	long	The updated version of the security profile.
creationDate	timestamp	The time the security profile was created.
lastModifiedDate	timestamp	The time the security profile was last modified.

## Errors

`InvalidRequestException`

The contents of the request were invalid.

`ResourceNotFoundException`

The specified resource does not exist.

**VersionConflictException**

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.

## UpdateStream

Updates an existing stream. The stream version will be incremented by one.

### Synopsis

```
aws iot update-stream \
--stream-id <value> \
[--description <value>] \
[--files <value>] \
[--role-arn <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "streamId": "string",
  "description": "string",
  "files": [
    {
      "fileId": "integer",
      "s3Location": {
        "bucket": "string",
        "key": "string",
        "version": "string"
      }
    }
  ],
  "roleArn": "string"
}
```

### cli-input-json fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
description	string  length- max:2028  pattern: [^\p{C}]+	The description of the stream.

Name	Type	Description
files	list  member: StreamFile	The files associated with the stream.
fileId	integer  range- max:255 min:0	The file ID.
s3Location	S3Location	The location of the file in S3.
bucket	string  length- min:1	The S3 bucket.
key	string  length- min:1	The S3 key.
version	string	The S3 bucket version.
roleArn	string  length- max:2048 min:20	An IAM role that allows the IoT service principal assumes to access your S3 files.

## Output

```
{
  "streamId": "string",
  "streamArn": "string",
  "description": "string",
  "streamVersion": "integer"
}
```

## CLI output fields

Name	Type	Description
streamId	string  length- max:128 min:1  pattern: [a-zA-Z0-9_-]+	The stream ID.
streamArn	string	The stream ARN.
description	string  length- max:2028  pattern: [^\p{C}]+	A description of the stream.
streamVersion	integer  range- max:65535 min:0	The stream version.

## Errors

**InvalidRequestException**

The contents of the request were invalid.

**ResourceNotFoundException**

The specified resource does not exist.

**ThrottlingException**

The rate exceeds the limit.

**UnauthorizedException**

You are not authorized to perform this operation.

**ServiceUnavailableException**

The service is temporarily unavailable.

**InternalFailureException**

An unexpected error has occurred.

## UpdateThing

Updates the data for a thing.

**Synopsis**

```
aws iot update-thing \
--thing-name <value> \
[--thing-type-name <value>] \
[--attribute-payload <value>] \
[--expected-version <value>] \
[--remove-thing-type | --no-remove-thing-type] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json format**

```
{
  "thingName": "string",
  "thingTypeName": "string",
  "attributePayload": {
    "attributes": {
      "string": "string"
    },
    "merge": "boolean"
  },
  "expectedVersion": "long",
  "removeThingType": "boolean"
}
```

**cli-input-json fields**

Name	Type	Description
thingName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing to update.

Name	Type	Description
thingTypeName	string length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	The name of the thing type.
attributePayload	AttributePayload	A list of thing attributes, a JSON string containing name-value pairs. For example:  <code>\"attributes\": {\"name1\":\"value2\"}</code>  This data is used to add new attributes or update existing attributes.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\"attributes\": {\"string1\": \"string2\"}</code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.  To remove an attribute, call UpdateThing with an empty attribute value.  <b>Note</b> The merge attribute is only valid when calling UpdateThing.
expectedVersion	long	The expected version of the thing record in the registry. If the version of the record in the registry does not match the expected version specified in the request, the UpdateThing request is rejected with a VersionConflictException.
removeThingType	boolean	Remove a thing type association. If true, the association is removed.

## Output

None

## Errors

### InvalidRequestException

The contents of the request were invalid.

### VersionConflictException

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

### ThrottlingException

The rate exceeds the limit.

### UnauthorizedException

You are not authorized to perform this operation.

### ServiceUnavailableException

The service is temporarily unavailable.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

# UpdateThingGroup

Update a thing group.

## Synopsis

```
aws iot update-thing-group \
--thing-group-name <value> \
--thing-group-properties <value> \
[--expected-version <value>] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

## cli-input-json format

```
{
    "thingGroupName": "string",
    "thingGroupProperties": {
        "thingGroupDescription": "string",
        "attributePayload": {
            "attributes": {
                "string": "string"
            },
            "merge": "boolean"
        }
    },
    "expectedVersion": "long"
}
```

## cli-input-json fields

Name	Type	Description
thingGroupName	string	The thing group to update.

Name	Type	Description
	length- max:128 min:1 pattern: [a-zA-Z0-9:_]+	
thingGroupProperties	ThingGroupProperties	The thing group properties.
thingGroupDescription	string  length- max:2028  pattern: [\p{Graph} ]*	The thing group description.
attributePayload	AttributePayload	The thing group attributes in JSON format.
attributes	map	A JSON string containing up to three key-value pair in JSON format. For example:  <code>\ "attributes\": {\ "string1\": \"string2\"} </code>
merge	boolean	Specifies whether the list of attributes provided in the AttributePayload is merged with the attributes stored in the registry, instead of overwriting them.  To remove an attribute, call UpdateThing with an empty attribute value.  <b>Note</b> The merge attribute is only valid when calling UpdateThing.
expectedVersion	long	The expected version of the thing group. If this does not match the version of the thing group being updated, the update will fail.

#### Output

```
{
  "version": "long"
}
```

#### CLI output fields

Name	Type	Description
version	long	The version of the updated thing group.

## Errors

### InvalidRequestException

The contents of the request were invalid.

### VersionConflictException

An exception thrown when the version of a thing passed to a command is different than the version specified with the --version parameter.

### ThrottlingException

The rate exceeds the limit.

### InternalFailureException

An unexpected error has occurred.

### ResourceNotFoundException

The specified resource does not exist.

## UpdateThingGroupsForThing

Updates the groups to which the thing belongs.

### Synopsis

```
aws iot update-thing-groups-for-thing \
[--thing-name <value>] \
[--thing-groups-to-add <value>] \
[--thing-groups-to-remove <value>] \
[--override-dynamic-groups | --no-override-dynamic-groups] \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

### cli-input-json format

```
{
  "thingName": "string",
  "thingGroupsToAdd": [
    "string"
  ],
  "thingGroupsToRemove": [
    "string"
  ],
  "overrideDynamicGroups": "boolean"
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The thing whose group memberships will be updated.
thingGroupsToAdd	list  member: ThingGroupName	The groups to which the thing will be added.

Name	Type	Description
thingGroupsToRemove	list member: ThingGroupName	The groups from which the thing will be removed.
overrideDynamicGroups	boolean	Override dynamic thing groups with static thing groups when 10-group limit is reached. If a thing belongs to 10 thing groups, and one or more of those groups are dynamic thing groups, adding a thing to a static group removes the thing from the last dynamic group.

#### Output

None

#### Errors

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`InternalFailureException`

An unexpected error has occurred.

`ResourceNotFoundException`

The specified resource does not exist.

## UpdateThingShadow

Updates the shadow for the specified thing.

For more information, see [UpdateThingShadow](#) in the AWS IoT Developer Guide.

#### Synopsis

```
aws iot-data update-thing-shadow \
--thing-name <value> \
--payload <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

`cli-input-json` format

```
{
  "thingName": "string",
  "payload": "blob"
}
```

### cli-input-json fields

Name	Type	Description
thingName	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name of the thing.
payload	blob	The state information, in JSON format.

### Output

```
{
  "payload": "blob"
}
```

### CLI output fields

Name	Type	Description
payload	blob	The state information, in JSON format.

### Errors

`ConflictException`

The specified version does not match the version of the document.

`RequestEntityTooLargeException`

The payload exceeds the maximum size allowed.

`InvalidRequestException`

The contents of the request were invalid.

`ThrottlingException`

The rate exceeds the limit.

`UnauthorizedException`

You are not authorized to perform this operation.

`ServiceUnavailableException`

The service is temporarily unavailable.

`InternalFailureException`

An unexpected error has occurred.

`MethodNotAllowedException`

The specified combination of HTTP verb and URI is not supported.

`UnsupportedDocumentEncodingException`

The encoding is not supported.

# ValidateSecurityProfileBehaviors

Validates a Device Defender security profile behaviors specification.

## Synopsis

```
aws iot validate-security-profile-behaviors \
--behaviors <value> \
[--cli-input-json <value>] \
[--generate-cli-skeleton]
```

**cli-input-json** format

```
{
  "behaviors": [
    {
      "name": "string",
      "metric": "string",
      "criteria": {
        "comparisonOperator": "string",
        "value": {
          "count": "long",
          "cidrs": [
            "string"
          ],
          "ports": [
            "integer"
          ]
        },
        "durationSeconds": "integer",
        "consecutiveDatapointsToAlarm": "integer",
        "consecutiveDatapointsToClear": "integer",
        "statisticalThreshold": {
          "statistic": "string"
        }
      }
    }
  ]
}
```

## cli-input-json fields

Name	Type	Description
behaviors	list member: Behavior	Specifies the behaviors that, when violated by a device (thing), cause an alert.
name	string  length- max:128 min:1  pattern: [a-zA-Z0-9:_]+	The name you have given to the behavior.
metric	string	What is measured by the behavior.
criteria	BehaviorCriteria	The criteria that determine if a device is behaving normally in regard to the metric.

Name	Type	Description
comparisonOperator	string	The operator that relates the thing measured ( <code>metric</code> ) to the criteria (containing a <code>value</code> or <code>statisticalThreshold</code> ).  enum: less-than   less-than-equals   greater-than   greater-than-equals   in-cidr-set   not-in-cidr-set   in-port-set   not-in-port-set
value	MetricValue	The value to be compared with the <code>metric</code> .
count	long  range- min:0	If the <code>comparisonOperator</code> calls for a numeric value, use this to specify that numeric value to be compared with the <code>metric</code> .
cidrs	list  member: Cidr	If the <code>comparisonOperator</code> calls for a set of CIDRs, use this to specify that set to be compared with the <code>metric</code> .
ports	list  member: Port	If the <code>comparisonOperator</code> calls for a set of ports, use this to specify that set to be compared with the <code>metric</code> .
durationSeconds	integer	Use this to specify the time duration over which the behavior is evaluated, for those criteria which have a time dimension (for example, <code>NUM_MESSAGES_SENT</code> ). For a <code>statisticalThreshold</code> metric comparison, measurements from all devices are accumulated over this time duration before being used to calculate percentiles, and later, measurements from an individual device are also accumulated over this time duration before being given a percentile rank.
consecutiveDatapointsToAlarm	integer  range- max:10 min:1	If a device is in violation of the behavior for the specified number of consecutive datapoints, an alarm occurs. If not specified, the default is 1.

Name	Type	Description
consecutiveDatapointsToClear	integer range- max:10 min:1	If an alarm has occurred and the offending device is no longer in violation of the behavior for the specified number of consecutive datapoints, the alarm is cleared. If not specified, the default is 1.
statisticalThreshold	StatisticalThreshold	A statistical ranking (percentile) which indicates a threshold value by which a behavior is determined to be in compliance or in violation of the behavior.
statistic	string pattern: (p0 p0.1 p0.01 p1 p10 p50 p90 p99 p99.9 p99.99 p100)	The percentile which resolves to a threshold value by which compliance with a behavior is determined. Metrics are collected over the specified period ( <code>durationSeconds</code> ) from all reporting devices in your account and statistical ranks are calculated. Then, the measurements from a device are collected over the same period. If the accumulated measurements from the device fall above or below ( <code>comparisonOperator</code> ) the value associated with the percentile specified, then the device is considered to be in compliance with the behavior, otherwise a violation occurs.

## Output

```
{
  "valid": "boolean",
  "validationErrors": [
    {
      "errorMessage": "string"
    }
  ]
}
```

## CLI output fields

Name	Type	Description
valid	boolean	True if the behaviors were valid.
validationErrors	list member: ValidationError	The list of any errors found in the behaviors.

Name	Type	Description
errorMessage	string length- max:2048	The description of an error found in the behaviors.

## Errors

**InvalidRequestException**

The contents of the request were invalid.

**ThrottlingException**

The rate exceeds the limit.

**InternalFailureException**

An unexpected error has occurred.