Fleet Hub for AWS IoT Device Management Guide

Fleet Hub for AWS IoT Device Management

Fleet Hub for AWS IoT Device Management: Fleet Hub for AWS IoT Device Management Guide

Table of Contents

What is Fleet Hub for AWS IoT Device Management?

With Fleet Hub for AWS IoT Device Management (Fleet Hub), you can build standalone web applications for monitoring the health of your device fleets. You can make these applications available to users in your organization, even if they don't have AWS accounts. Use Fleet Hub to manage common fleet-wide tasks such as investigating and remediating operational and security issues.

Fleet Hub provides the following capabilities.

- Monitor device fleets in near-real time.
- Set alerts to notify your technicians about unusual behavior.
- Running jobs.

> **Note**
>
> For Fleet Hub to index connectivity status data, your Things must connect to AWS IoT Core with client ID equal to Thing name.

How Fleet Hub for AWS IoT Device Management works

Administrators can use Fleet Hub for AWS IoT Device Management to create secure web applications in a few minutes without provisioning any resources or writing any code. Web applications that you create by using Fleet Hub integrate with your existing identity systems, such as Active Directory. This allows your administrators to apply their own authentication and authorization models.

Fleet Hub web applications integrate with AWS IoT Core fleet indexing and device monitoring. These integrations provide the ability to monitor device health data and create alarms when devices in your fleet reach a specified state.

Fleet Hub applications use the `AWSIoTFleetHubFederationAccess` managed policy. For more information, see ???.

Example use cases:

- Visualize device connectivity issues - You can see the number of disconnected devices in your fleet, the last connection status for a device, and the reason or reasons why devices disconnected.
- Set alarms - You can set thresholds that trigger alarms when a particular number of devices disconnect. Alarms can also notify you when a device or devices disconnect for a particular reason. You can then look at detailed device data to investigate and troubleshoot.
- Run jobs - You can run remote operations (such as firmware updates) on one or more devices.

How Fleet Hub data indexing works

You can use the Fleet Hub console to activate fleet indexing for your device fleet. When you activate fleet indexing in Fleet Hub, you activate it for the entire fleet and make it available to all Fleet Hub applications.

When it's enabled, fleet indexing indexes all AWS IoT Core-managed fields automatically. You can also use fleet indexing to add custom data that you can use to query and aggregate data in Fleet Hub applications.

How Fleet Hub alarms work

Fleet Hub web applications provide an interface that allows your users to create alarms. The following steps show how users create alarms in Fleet Hub.

1. Create a query to aggregate data - Specify a query that aggregates the devices your users want to target by using searchable fields.
2. Configure threshold - Set a threshold that triggers the alarms when a condition in the indexed data (such as connectivity status over a specified interval) is reached.
3. Configure notification - Specify a group of recipients whom Fleet Hub notifies when the specified devices are in alarm.

How Fleet Hub jobs work

You can use the Fleet Hub console to run remote operations on devices.

When job templates are enabled, you can create specific jobs from the templates in your Fleet Hub applications.

Fleet Hub for AWS IoT Device Management for administrators

This section contains guidance for administrators for how to create and manage Fleet Hub for AWS IoT Device Management web applications.

Topics

Getting started

This section explains how to create and set up Fleet Hub for AWS IoT Device Management web applications.

Topics

Create your first Fleet Hub application

Prerequisites

The following list contains the resources you need to create a Fleet Hub web application.

- An AWS account.
- AWS IAM Identity Center turned on for your account. (If you haven't already activated this service, the AWS IoT Core console (https://console.aws.amazon.com/iot/) prompts you to do so.)

Create your first Fleet Hub web application

The following steps describe how to create Fleet Hub for AWS IoT Device Management web applications.

1. Navigate to the AWS IoT Core console (https://console.aws.amazon.com/iot/), and in the left panel, choose **Fleet Hub**, and then **Applications**.
2. On the applications page, choose **Create application**.
3. On the **Set up IAM Identity Center** page, if you haven't activated AWS IAM Identity Center (IAM Identity Center), follow the steps to activate it. AWS Organizations sends you an email. Choose the link in the email to finish activating IAM Identity Center.

   > **Note**
   >
   > You can connect your own identity provider to IAM Identity Center. For more information, see What Is AWS IAM Identity Center? and Connect to your external identity provider.
   > When creating a Fleet Hub application, you must create an organization instance of IAM Identity Center if you don't already have one. The Fleet Hub application you create must also be in the same AWS Region of the organization instance of IAM Identity Center. For more information see Enabling IAM Identity Center and Organization instances of IAM Identity Center.

   The page tells you if you have already activated IAM Identity Center.

   Choose **Next**.
4. On the **Index AWS IoT data** page, review the information in the **How the data flow works from AWS IoT to Fleet Hub** section. This page links you to the pages in the AWS IoT Core console where you can activate and manage AWS IoT Core fleet indexing. You can use this service to index, search, and aggregate your registry data, shadow data, device connectivity data (device lifecycle events), and device violations data. You can also create custom fields in addition to the managed fields that AWS IoT Core fleet indexing indexes by default.

   - If you've activated fleet indexing, this pages displays your fleet indexing settings and custom fields.

   - If you haven't enabled thing indexing and thing connectivity, you must do so to use Fleet Hub.

When you're done managing and reviewing your fleet indexing settings, choose **Next**.

For more information about how to activate fleet indexing for Fleet Hub applications, see Managing fleet indexing for Fleet Hub applications.

5. On the **Configure application** page, in the **Application role** section, create a new service role or select an existing service role. Your Fleet Hub web application assumes this role when it uses Fleet Hub resources. Federated users have the same permissions as the role when they use the web application.

   - If you create a new role, the role name must begin with the following string: `AWSIotFleetHub_`*`random_string`*.

   - If you select an existing role, make sure that it has the permissions that are in the policy document. To see the permissions that your Fleet Hub web application needs, choose **View role details**. A window opens that shows you the policy document that the service applies to any new role that you create from this page.

6. On the **Configure application** page, in the **Application properties** section, enter a name for your application. Optionally, you can also enter an application description.

   Choose **Create application**.

7. On the **Applications** page, select the application that you created and choose **View details**. Review the details of the application.

> **Note**
>
> For more information about possible solutions for resolving issues as an administrator of Fleet Hub, see Troubleshooting.

Manage fleet indexing for Fleet Hub applications

You can use the AWS IoT Core console or the AWS CLI to activate fleet indexing and configure the following data sources to index: AWS IoT registry data, AWS IoT Device Shadow data, AWS IoT connectivity data, and AWS IoT Device Defender violations data. The following steps describe how

to activate fleet indexing for Fleet Hub for AWS IoT Device Management applications in AWS IoT Core console. To view the steps using AWS CLI, see Managing thing indexing.

> **Important**
>
> July 20th, 2022 is the General Availability release of AWS IoT Device Management fleet indexing's integration with AWS IoT Core named shadows and AWS IoT Device Defender detect violations. With this GA release, you can index specific named shadows by specifying shadow names. If you added your named shadows for indexing during this feature's public preview period from November 30, 2021 to July 19, 2022, we encourage you to reconfigure your fleet indexing settings and choose specific shadow names to reduce indexing cost and optimize performance. For more information about how to reconfigure your fleet indexing settings, see Managing fleet indexing.

1. Navigate to the AWS IoT Core console (https://console.aws.amazon.com/iot/), and in the left panel, choose **Settings**.
2. On the **Settings** page, navigate to the **Fleet indexing** section, then choose **Manage indexing**.
3. On the **Manage fleet indexing** page, in the **Configuration** section, choose **Thing indexing** and the data sources that you want AWS IoT to index. You must activate thing indexing and thing connectivity to use Fleet Hub.
4. (Optional) On the **Manage fleet indexing** page, in the **Custom fields for aggregation-optional** section, create custom fields in addition to the managed fields that fleet indexing indexes by default.

   When you're done managing and reviewing your fleet indexing settings, choose **Next**.

   It can take a moment for fleet indexing to update the settings. For more information about how to manage fleet indexing, see Managing fleet indexing.

Add users to Fleet Hub applications

Your Fleet Hub for AWS IoT Device Management web application doesn't contain any users when it's newly created. You must add users before you and members of your organization can use the application. The steps in this topic describe how to add users to your application.

You add users from your existing identity system by setting up AWS IAM Identity Center (IAM Identity Center) for your account. You can connect your own identity provider to IAM Identity Center. For more information, see What Is IAM Identity Center?.

1. On the **Applications** page, choose your web application from the **Fleet Hub applications** list. Choose **View details**.
2. On the application details page, choose **Add user**.
3. In the **Add Fleet Hub users** window, select the users from your organization that you want to have access to the application. Choose **Add selected users**.
4. On the application details page, verify that you see the users you selected in the **Fleet Hub users** list.

AWS and AWS IoT Core services that interact Fleet Hub for AWS IoT Device Management

This topic explains how the features in Fleet Hub for AWS IoT Device Management interact with other AWS services to deliver the capabilities in your Fleet Hub web applications.

The following table indicates what AWS services Fleet Hub for AWS IoT Device Management uses to implement each feature.

Capability	AWS service	Description
Integrate existing identity systems, such as Active Directory.	AWS IAM Identity Center (IAM Identity Center)	You add users from your existing identity system by setting up AWS IAM Identity Center (IAM Identity Center) for your account. You can connect your own identity provider to IAM Identity Center. For more information, see What Is AWS IAM Identity Center? and Workforce identities.

Capability	AWS service	Description
Create queries by using AWS-managed fields, custom fields, and any attributes in your indexed data sources.	AWS IoT fleet indexing	Use the fleet indexing service to index, search, and aggregate your registry data, shadow data, and device connectivity data (device lifecycle events). You can also create custom fields for aggregation in addition to the managed fields that AWS IoT fleet indexing indexes by default. For more information about fleet indexing, see Fleet indexing.

Capability	AWS service	Description
Create alarms for a set of devices specified by a query.	Amazon CloudWatch (CloudWatch)	Fleet Hub dashboards expose CloudWatch metrics that you can use in combinati on with searchable fields to create alarming threshold s. For example, you can create CloudWatch alarm that generates an Amazon Simple Notification Service (Amazon SNS) notificat ion whenever the number of connected devices falls beneath a specified quantity. For information about CloudWatch, see What Is Amazon CloudWatch? For information about how AWS IoT Core works with CloudWatch to create metrics and alarms, see Monitor AWS IoT alarms and metrics using CloudWatch.

Troubleshooting

This section provides troubleshooting information and possible solutions to help resolve issues as an administrator of Fleet Hub.

Symptom	Solution
My web application link doesn't work.	It might take a few hours after you've created your application for the link to work.

Symptom	Solution
I can't log in to my web application.	Make sure that you have added at least one user to your application. Make sure that your role has the appropriate trust relationship such as the following: `{"Version": "2012-10-17", "Statement": [ { "Effect": "Allow", "Principal": { "Service": "iotfleethub.amazonaws.com" }, "Action": "sts:AssumeRole" } ] }` For more information about how to edit IAM trust relationship, see Editing the trust relationship for an existing role.
I can't create a web application.	Make sure that you haven't reached your limit of total number of web applications.
I'm not seeing a custom field that I'm expecting.	Check to make sure that you've set up fleet indexing correctly. For more information about fleet indexing, see Fleet indexing.

Fleet Hub for AWS IoT Device Management for users

This section contains information for users of Fleet Hub for AWS IoT Device Management web applications. For information about creating Fleet Hub applications and adding users to them, see *Fleet Hub for AWS IoT Device Management for administrators*.

Topics

Getting started

This section contains information about getting started with using the features of Fleet Hub for AWS IoT Device Management web applications.

Topics

Create your first query

This topic walks you through the steps to create a simple Fleet Hub for AWS IoT Device Management query. The queries are specified using search query syntax.

Prerequisites

- A Fleet Hub application associated with an AWS IoT Core account that contains devices (things).
- An account in your organization that has permissions to use the Fleet Hub application.

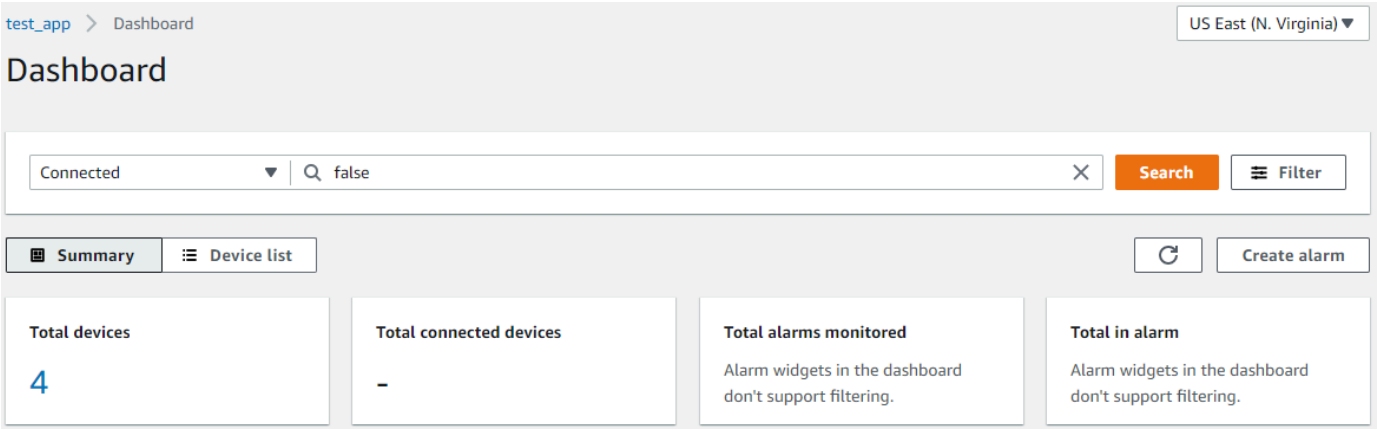
Create your first Fleet Hub query

Create your first Fleet Hub query

1. Navigate to your Fleet Hub application.

   The default dashboard view displays a list of all devices that contains the managed and custom attributes. The attributes that contain the **attributes** prefix are custom attributes.

2. On the menu at the top of the page, choose **Connected** from **All fields**. Enter **`false`** in the text box next to the dropdown menu.



3. To perform the search, choose **Search**. You see a list of all devices that aren't connected to AWS IoT Core.

For more information about the query syntax and example queries, see Query syntax, Example thing queries, and Example thing group queries.

Create your first alarm

This topic walks you through the steps to create a simple Fleet Hub for AWS IoT Device Management alarm.

Prerequisites

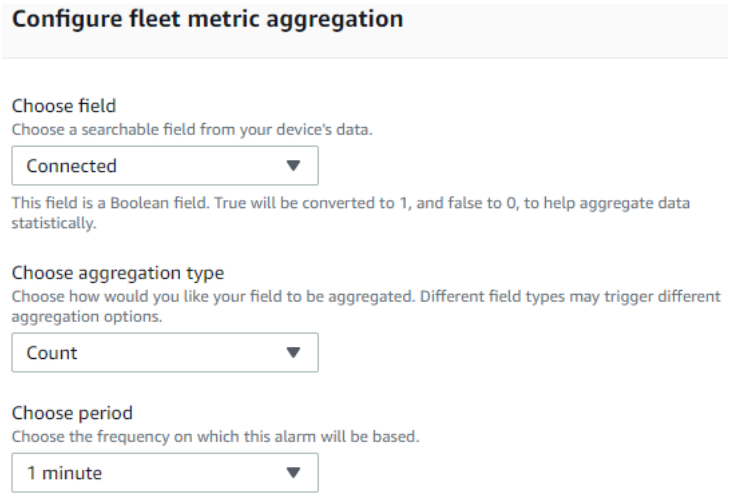
- A Fleet Hub application associated with an AWS IoT Core account that contains devices (things).
- An account in your organization that has permissions to use the Fleet Hub application.

Creating your first alarm

Create your first Fleet Hub alarm

1. Navigate to your Fleet Hub application.
2. If you want to target a specific set of devices, create a query. For instructions on how to create a simple query, see the section called “Create your first query”. If you don't create a query, your alarm will apply to all of the devices in your fleet.
3. On the default dashboard page, choose **Create alarm**.
4. On the **Build aggregation metric** page, verify that your query appears under **Target query**. In the **Configure fleet metric aggregation** section, on the **Choose field** menu, choose **Connected**. This AWS-managed field indicates whether a device is connected to AWS IoT Core. The **Choose field** menu contains the AWS-managed fields and the custom fields that your administrator has created in AWS IoT fleet indexing.
5. For **Choose aggregation type**, choose any one of the following options.
   - **Maximum** -- Configure a maximum threshold.
   - **Count** -- Configure a specific count as the threshold.
   - **Sum** -- Configure a sum as the threshold.
   - **Minimum** -- Configure a minimum threshold.
   - **Average** -- Configure an average threshold.
6. For **Choose period**, choose the duration of the condition specified in the preceding menus that will trigger the alarm.

   An example setting for **Configure fleet metric aggregation** can look like the following:



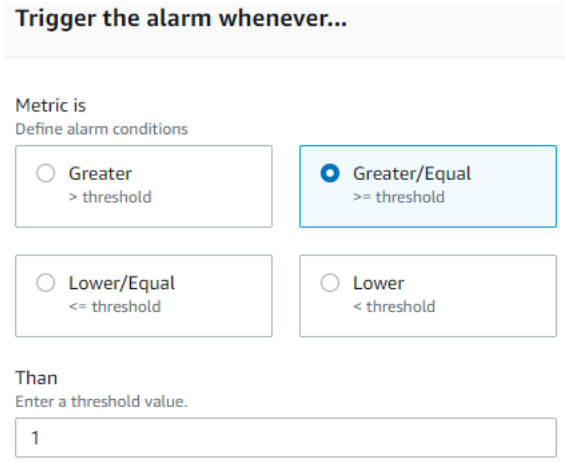
Choose **Next**.

7. On the **Set threshold** page, in the **Trigger the alarm whenever...** section, choose one of any of the following options.

   - **Greater** -- Alarms when the aggregation metric and type exceeds the specified value.
   - **Greater/Equal** -- Alarms when the aggregation metric and type equals or exceeds the specified value.
   - **Lower** -- Alarms when the aggregation metric and type falls below the specified value.
   - **Lower/Equal** -- Alarms when the aggregation metric and type equals or falls below the specified value.

8. In the **Than** text box, specify the value to use as the threshold for the alarm.

   An example setting for **Set threshold** can look like the following:



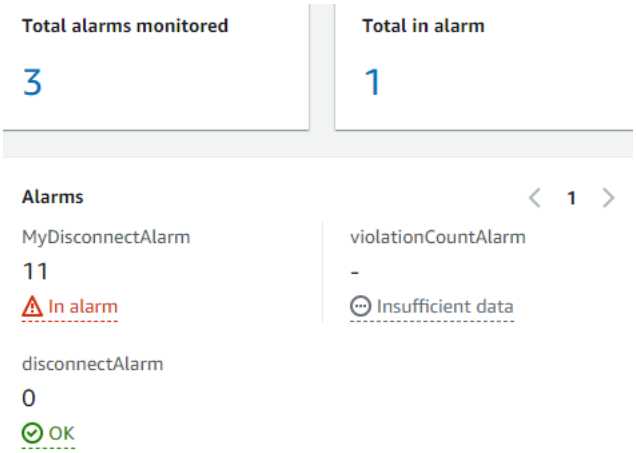
   Choose **Next**.

9. On the **Notify user** page, in the **Notify -- optional** section, enter a name for the email list that contains the users in your organization who receive notifications when the alarm is active. Enter a comma-separated list of email addresses to populate this list.

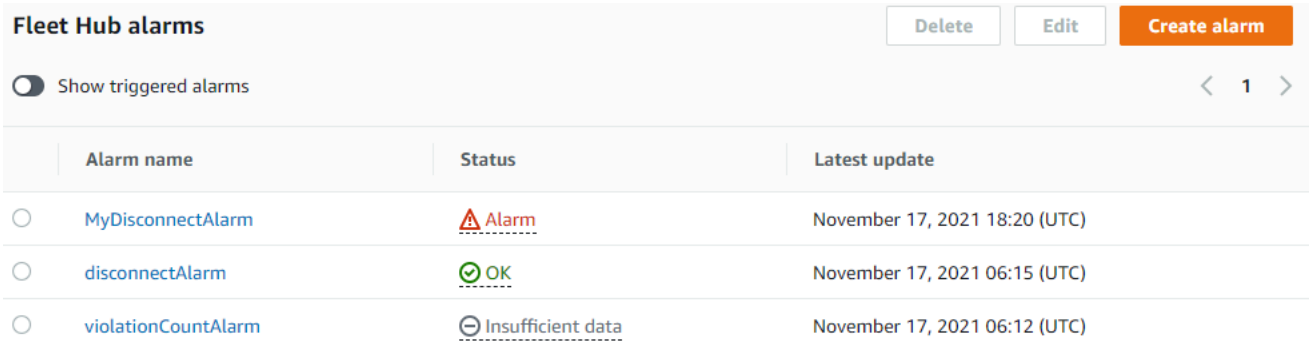
10. In the **Alarm details** section, enter a name for your alarm, and optionally enter a description for your alarm. Choose **Next**.

11. On the **Review** page, verify the information that you entered on the previous pages. Choose **Submit**. You return to the default dashboard.

12. On the default dashboard, the alarms widgets display information of all the alarms you created.



To see details of the alarms that you created, in the left navigation panel, choose **Fleet Hub alarms**.



View device details

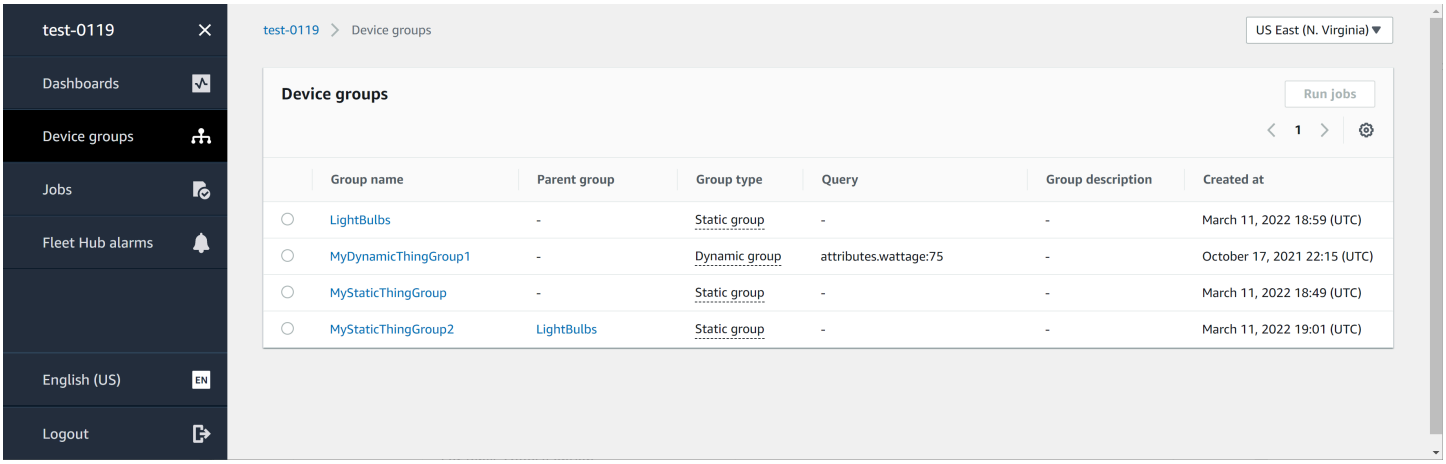
This topic walks you through the steps to view details about your device groups and your devices.

Prerequisites

- A Fleet Hub application associated with an AWS IoT Core account that contains devices (things).
- An account in your organization that has permissions to use the Fleet Hub application.

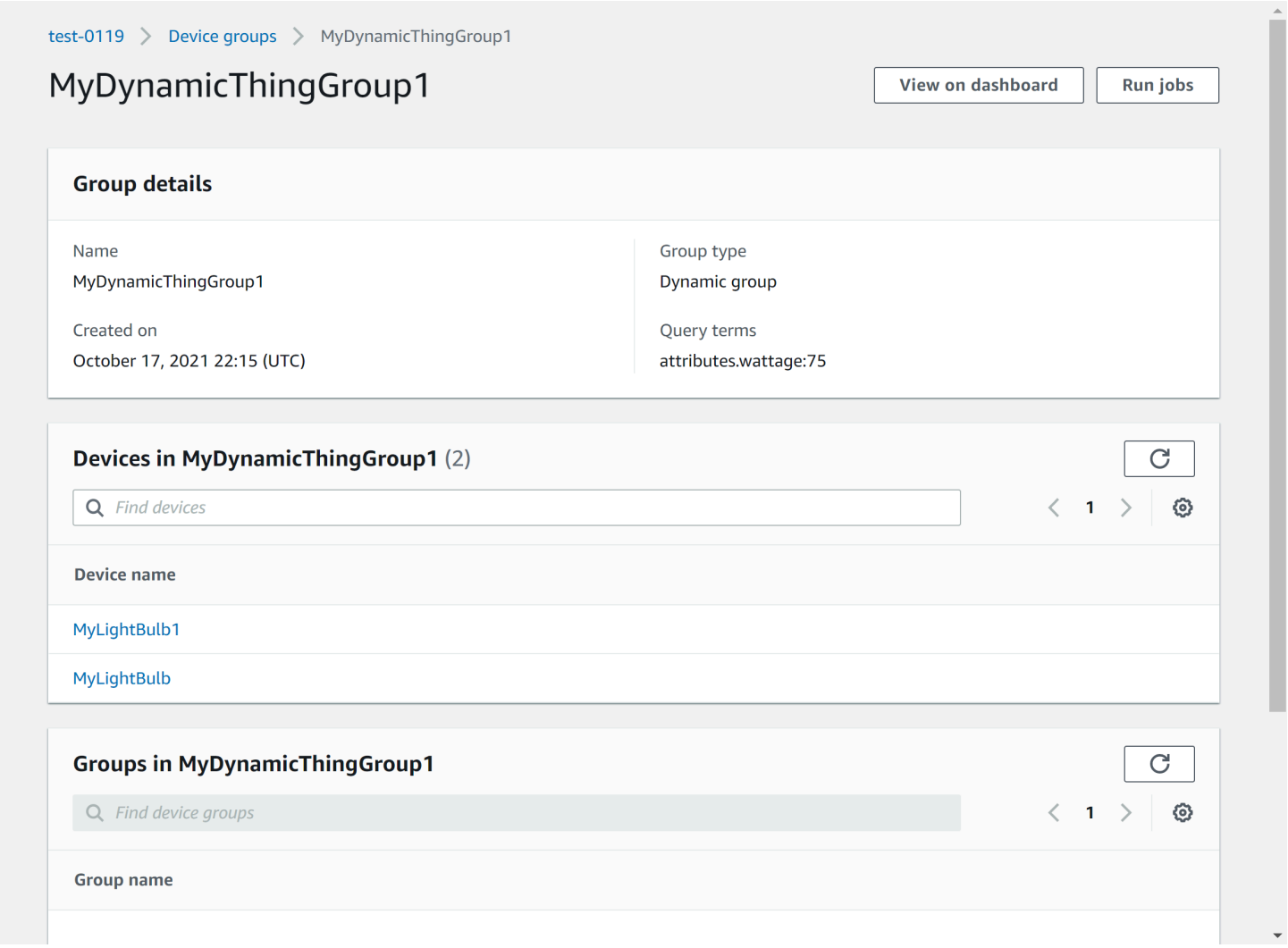
Device groups

When you log in to your Fleet Hub web application, you see **Device groups** on the left navigation panel. The **Device groups** page lists all the device groups in your Fleet Hub web application. To view the details of a device group, choose a specific device group from the **Group name** column.



Device group details

The **Device group details** page contains information about your selected device group. To view the details of a device, choose a specific device from the **Device name** column of the **Devices in *XXX*** section.



Device details

The **Device details** page contains information about your selected device.

> **Note**
>
> If your client is using a different client ID from Thing Name when connecting to AWS IoT, the connectivity status of your "thing" won't be indexed by Fleet Indexing.

Details

The **Details** section contains the following information about your device:

- **Device name** – The name of the thing resource that represents your device. For more information, see How to manage things with the registry.

- **Thing type** – The thing type that's associated with your device. You can use the thing type to store information that's common to all things with the same thing type. For more information, see Thing types.

- **Last connection timestamp** – The timestamp for when your device last connected to AWS IoT.

- **Shareable device link** – A shareable link that points to the **Device details** page of the selected device.

- **Last connection status** – The connection status of your device to AWS IoT. If your device is connected, the value is `true`. If it's not connected, the value is `false`.

- **Disconnect reason** – The reason why your device is disconnected.

Reported data

The **Reported data** section contains information about your device's registry data, device shadows data, and thing groups.

- **Device fields** – The indexed fields of your device in AWS IoT fleet indexing. For more information, see Managing fleet indexing.

- **Device shadows** – The shadows that are associated with your device. The device shadows can include both classic unnamed shadows and named shadows. For more information, see AWS IoT device shadow.

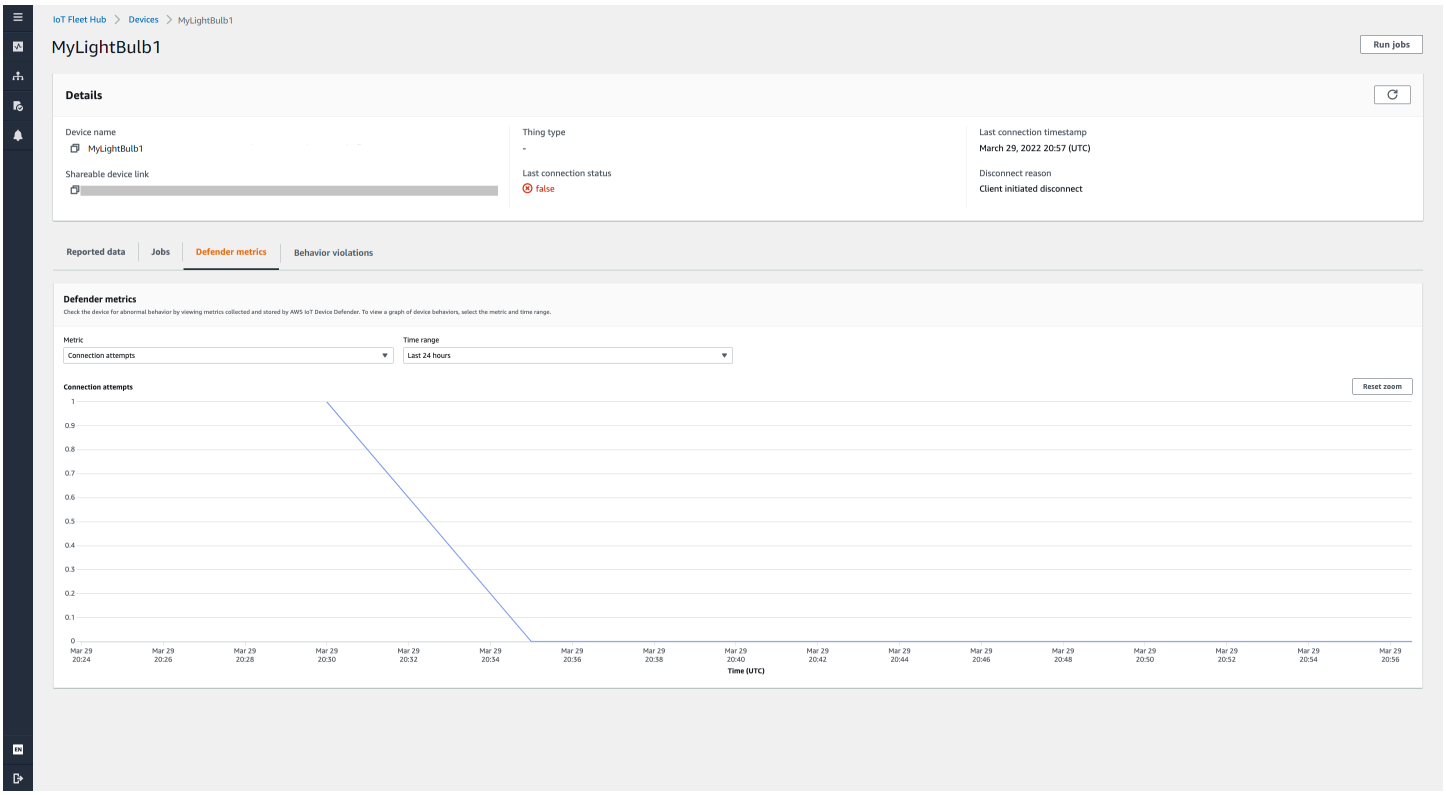
- **Device groups** – The device groups that are associated with your device. The device groups can include both static thing groups and dynamic thing groups. For more information, see Static thing groups and Dynamic thing groups.

Jobs

The **Jobs** section displays all of the jobs running on the device. Each job has a details page that displays summary information about the job, including target and runtime information. For more information, see Working with jobs and job templates in Fleet Hub for AWS IoT Device Management, and Jobs.

Defender metrics

The **Defender metrics** section displays AWS IoT Device Defender metrics that are associated with your currently selected device. You can use the displayed metrics data to visualize your device operation across a time frame you choose. To view the defender metrics data from your Fleet Hub application, your Fleet Hub administrator must first set up AWS IoT Device Defender metrics that are associated with the selected device. For more information about how to create and set up AWS IoT Device Defender metrics for your devices, see Custom metrics, Device-side metrics, and Cloud-side metrics.



Behavior violations

The **Behavior violations** section displays the indexed AWS IoT Device Defender detect violations data that are associated with your currently selected device. The behavior violations data can include violation count, last violation time, and last violation metric value. To view the behavior

violations data from your Fleet Hub application, your Fleet Hub administrator should set up AWS IoT Device Defender behavior violations in a security profile and configure AWS IoT Device Defender violations in fleet indexing. For more information about how to set up behavior violations in an AWS IoT Device Defender security profile, see AWS IoT Device Defender Detect. For more information about how to configure AWS IoT Device Defender violations, see Manage fleet indexing for Fleet Hub applications and Managing thing indexing.

Queries and filters

You can use Fleet Hub for AWS IoT Device Management queries to create and view lists of things in your device fleet. All AWS-managed fields, custom fields, and any attributes in your indexed data sources are available to you as query filters. You can also create custom fields to activate aggregation for the section called "Alarms" by using AWS IoT fleet indexing. For more information about fleet indexing, see Fleet indexing.

Topics

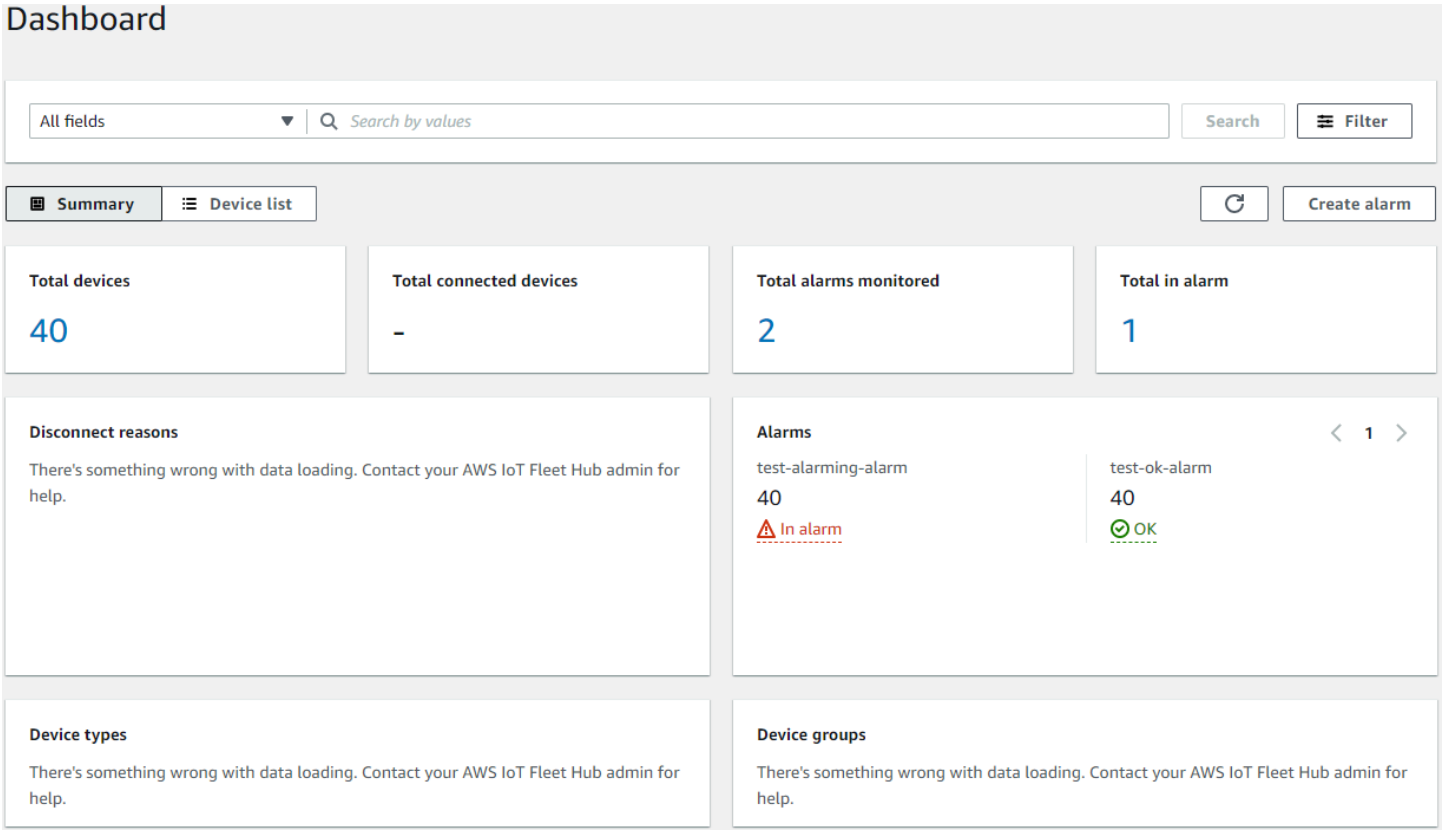
View the dashboard

When you log in to your Fleet Hub for AWS IoT Device Management web application, you see a dashboard that presents two views of data about the devices in your fleet.

Summary

The **summary** view displays a rolled-up view of data about all of the devices in your fleet. It provides the following information.

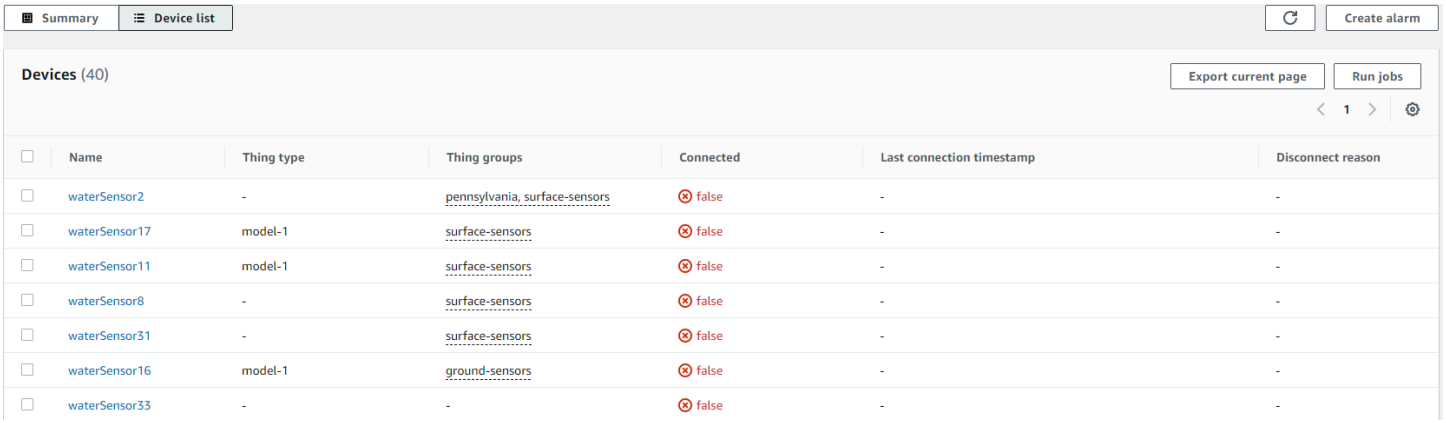
- Total number of devices
- Number of connected devices
- A list of reasons why devices have disconnected
- The thing types that you have created for your fleet and the number of devices for each type
- The thing groups that you have created for your fleet and the number of devices in each group



Device list

The **Device list** view displays a table that lists the devices in your fleet. The table provides the following information for each device in the list.

- The device name
- The device's connection status
- The timestamp for the device's last connection
- For a device that isn't connected, the reason why it disconnected
- The device's thing type
- The device's thing group
- The custom fields that you've created in the fleet indexing service



To download a CSV file that contains the devices displayed on the page, on the device list, choose **Export current page**. Note that if the list is paginated, this only downloads data displayed on the current page, not on subsequent pages.

You can use queries and filters to narrow the number of devices that generate the summary data in the first view and that appear in the device list. For more information about using queries and filters to get more specific information about devices in your fleet, see the section called “Creating queries”.

Create queries with filters

This topic explains how Fleet Hub for AWS IoT Device Management queries work and walks you through the steps required to create a query with filters.

You can control the number and types of devices that display on your dashboard summary and list views by using queries. You filter queries by using AWS-managed fields, custom fields, and any attributes from your indexed data sources from AWS IoT fleet indexing. For more information about the fleet indexing, see Fleet indexing.

You can also add keywords to your queries. Keywords apply across all searchable fields. They also count against the limit of three filters that you can apply in a single query.

The following section describes the steps required to create a typical query.

Creating queries

The following steps describe how to create a typical query.

Prerequisites

- A Fleet Hub application tied to an AWS IoT Core account that contains several devices (things)

- An account that has permissions to use the Fleet Hub application

Create your first Fleet Hub query with a filter in the console

1. Navigate to your Fleet Hub application.
2. On the default dashboard, verify that you can see the **Device list** tab and the total number of devices (things) in the associate AWS IoT Core account.

   The default dashboard contains navigation tabs, including one for the device list. It displays the total number of devices in the associated AWS IoT Core account and the total number of connected devices.
3. On the default dashboard, choose the **Device list** tab. Verify that you see a list of all devices that contain the managed and custom attributes. The custom attributes contain the **attributes** prefix.

   By default, the device list dashboard displays custom and managed attributes for all devices in the associated AWS IoT Core account.
4. At the top of the page, enter any keyword you want to include in your query. Keyword queries apply to all fields.
5. At the top of the page, choose **Filter**.
6. In the **Filter** modal, under **Field**, choose the field that you want to use as a filter. Under **Operator**, choose an option. Finally, for **Value**, choose the field value to use in your filter.

   You can add up to three filters. A keyword query counts against this number.
7. To perform your query, choose **Apply filters**. The results show all the devices that match your query.

Working with jobs and job templates in Fleet Hub for AWS IoT Device Management

> **Note**
>
> The job templates feature is in preview and subject to change.

A job is a remote operation that is sent to and run 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. You can run preconfigured jobs from Fleet Hub for AWS IoT Device Management web applications. Your organization's administrators create job templates in the AWS IoT console and attach policies that make the templates available to Fleet Hub users. In your Fleet Hub application, you specify the devices or a device group on which the job runs.

Administrators also create device groups that you can view in your application. To see these groups, choose **Device groups** in the navigation pane. When you specify a device group as a target, you can specify one of the following two types of options for how the job runs.

- snapshot: The job runs once.
- continuous: After its initial run, the job runs on any device that is added to the group.

For more information about creating and managing job templates, see Job templates. For more information about how jobs work, see Jobs.

Running jobs

You can run a job from several locations in a Fleet Hub application, but the following steps are always the same.

1. Select a group or one or more devices as the target.
2. Choose **Run job**.
3. Under **Job target selection**, choose either **continuous** or **snapshot**.
4. Select a job template. Verify that the text under **Job summary** describes the type of job that you want to run.
5. Optionally, enter a name for the job.
6. Choose **Run**.

You can select targets and follow these steps from the following locations in your Fleet Hub application.

- The device list tab on the dashboard.
- The details page of a specific device.

- Device groups page.
- The details page of a specific device group.

Viewing and managing jobs

You can see jobs that are running in your fleet in the following locations.

- The jobs list page -- This page displays all of the jobs running in your fleet. To see this page, choose **Jobs** in the navigation pane.
- The details page for a specific device -- This page displays all of the jobs running on the device.

Each job has a details page that displays summary information about the job, including target and runtime information. This page shows the runtime status of the job on each device. It also displays the following totals.

- Number of runs.
- Number of canceled runs.
- Number of successful runs.
- Number of failed runs.
- Number of rejected runs.
- Number of queued runs.
- Number of in progress runs.
- Number of removed runs.
- Number of timed out runs.

To cancel a job, select the job and choose **Cancel**.

Alarms

This section explains how Fleet Hub for AWS IoT Device Management alarms work and walks you through the steps required to create an alarm.

When you create a Fleet Hub alarm, it applies to all of the devices currently showing in your dashboard. If you apply no query, the alarm applies to all devices in your fleet. For information about using your dashboard and creating queries, see the section called “Queries and filters”.

Alarms use Amazon CloudWatch (CloudWatch) metrics in combination with searchable fields from the AWS IoT fleet indexing service to create CloudWatch alarms. For example, you can create an alarm that generates an Amazon Simple Notification Service (Amazon SNS) message whenever the average battery level of the devices in your fleet falls below 50%.

Fleet Hub alarms use the GetStatistics and GetPercentiles capabilities of the fleet indexing service to query aggregate data. For example, when you create an alarm that tracks a custom numerical field, you can create alarming thresholds that apply to the following values of the specified attribute.

- Maximum
- Count
- Sum
- Minimum
- Average
- Values in the 10th, 50th, 90th, 95th, or 99th percentile

For more information about querying aggregate data in the fleet indexing service, see Querying for aggregate data.

The following table lists some examples of the aggregation types that are available for AWS-managed and custom fields.

Field	Aggregation type
Thing type (AWS-managed string field)	Count
Thing group (AWS-managed string field)	Count
Connected (AWS-managed Boolean field) The value of `true` is 1. The value of `false` is 0.	• Maximum • Count • Sum • Minimum • Average

Field	Aggregation type
shadow.reported.batterylevel (numerical aggregation field created in the fleet indexing service)	• Maximum • Count • Sum • Minimum • Average • p10 (10th percentile) • p50 (50th percentile) • p90 (90th percentile) • p95 (95th percentile) • p99 (99th percentile)

In addition to specifying aggregation fields and types, you also specify the following values.

- The duration of time (1 minute or 5 minutes) required for your specified alarming threshold to trigger the alarm.
- One of the following comparison operators to apply to your specified aggregation field and type.
  - Greater
  - Greater/Equal
  - Lower
  - Lower/Equal
- The value to use with your specified comparison operator.
- A list of email addresses of people in your organization who receive Amazon SNS messages whenever your alarm is triggered.
- An alarm name.

To create a Fleet Hub alarm, see the section called “Creating alarms”.

Creating alarms

This topic walks you through the steps required to create a Fleet Hub for AWS IoT Device Management alarm. It assumes that your administrator has created an aggregation field out of a

device shadow field named **shadow.reported.batterylevel**. This custom field indicates the battery level of a device. You need to ask your administrator to create searchable custom fields in the AWS IoT fleet indexing service.

The alarm that you create sends an Amazon Simple Notification Service (Amazon SNS) message to a list of people in your organization whenever the average battery level of devices in your fleet falls below 50% during a period of 1 minute.

Create a Fleet Hub query

1. Navigate to your Fleet Hub application.
2. If you want to target a specific set of devices, create a query. For instructions on how to create a simple query, see the section called “Create queries with filters”. If you don't create a query, your alarm applies to all of the devices in your fleet.
3. On the default dashboard page, choose **Create alarm**.
4. On the **Build aggregation metric** page, verify that your query appears under **Target query**. In the **Configure fleet metric aggregation** section, for **Choose field**, choose **shadow.reported.batterylevel**. This menu contains the AWS-managed fields and the custom fields that your administrator has created in the AWS IoT fleet indexing service.
5. For **Choose aggregation type**, choose **Average**. This choice bases the alarm on the average battery level value in your device fleet.
6. For **Choose period**, choose **1 minute**. This triggers the alarm when your device fleet remains in the specified alarming state for one minute.

   Choose **Next**.
7. On the **Set threshold** page, in the **Trigger the alarm whenever...** section, choose **Lower/Equal**. This triggers the alarm when the average battery level value falls below a value that you specify.
8. In the **Than** text box, enter 50.

   Choose **Next**.
9. On the **Notify user** page, in the **Notify -- optional** section, enter a name for the email list that contains the users in your organization who receive notifications when the alarm is active. Enter a comma-separated list of email addresses to populate this list.
10. In the **Alarm details** section, enter a name for your alarm, and optionally enter a description for your alarm. Choose **Next**.

11. On the **Review** page, verify the information that you entered on the previous pages. Choose **Submit**. You return to the default dashboard.
12. On the default dashboard, in the left navigation panel, choose **Fleet Hub alarms**. Verify that you see the alarm that you created.

Troubleshooting

This section provides troubleshooting information and possible solutions to help resolve issues as a user of Fleet Hub.

Symptom	Solution
I can't add more filters or terms to my query.	Make sure that you haven't reached the limit of four query terms and filters.
I can't find a custom metric.	Ask your administrator to create the metric in the fleet indexing service.
My alarm isn't showing any data.	Alarm data takes a few minutes to load.
I need to change the devices that my alarm targets.	Go to your dashboard and change the query.
I see an error when I change the Region in my dashboard.	Ask your administrator to make sure that fleet indexing is activated in the Region you selected.
The connectivity status of my "thing" is not indexed by Fleeting Indexing.	Make sure your client is using the same client ID as Thing Name when connecting to AWS IoT. If your client is using a different ID from Thing Name when connecting to AWS IoT, the connectivity status of your "thing" won't be indexed by Fleet Indexing.

Monitoring Fleet Hub for AWS IoT Device Management

Monitoring is an important part of maintaining the reliability, availability, and performance of Fleet Hub and your other AWS solutions. AWS provides the following monitoring tools to watch Fleet Hub, report when something is wrong, and take automatic actions when appropriate.

- *AWS CloudTrail* captures API calls and related events made by or on behalf of your AWS account and delivers the log files to an Amazon S3 bucket that you specify. You can identify which users and accounts called AWS, the source IP address from which the calls were made, and when the calls occurred. For more information, see the AWS CloudTrail User Guide.

Topics

Logging Fleet Hub for AWS IoT Device Management API calls with AWS CloudTrail

Fleet Hub for AWS IoT Device Management is integrated with AWS CloudTrail. The CloudTrail service provides a record of actions that a user, role, or an AWS service takes in Fleet Hub. CloudTrail captures all API calls for Fleet Hub as events. Captured calls include those from the Fleet Hub console and code calls to the Fleet Hub API operations.

If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Fleet Hub. 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 that CloudTrail collects, you can determine the request that was made to Fleet Hub, the IP address from which the request was made, who made the request and when, and more details.

To learn more about CloudTrail, see the AWS CloudTrail User Guide.

Fleet Hub information in CloudTrail

AWS CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Fleet Hub, that activity is recorded in a CloudTrail event 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 Fleet Hub, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon Simple Storage Service (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 Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify.

You can also configure other AWS services to further analyze and act on the event data collected in CloudTrail logs. For more information, see the following:

- Overview for creating a trail
- CloudTrail supported services and integrations
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions
- Receiving CloudTrail log files from multiple accounts

CloudTrail logs all Fleet Hub actions. They're documented in the AWS IoT API Reference. For example, calls to the `CreateApplication` and `UpdateApplication` actions 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 AWS Identity and Access Management 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.

Understanding Fleet Hub for AWS IoT Device Management 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 aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

Example

The following CloudTrail log entry shows information about the `CreateApplication` action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "principal-id",
        "arn": "arn:aws:sts::123456789012:assumed-role/Admin/test-user-name",
        "accountId": "123456789012",
        "accessKeyId": "access-key",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "principal-id",
                "arn": "arn:aws:iam::123456789012:role/Admin",
                "accountId": "123456789012",
                "userName": "Admin"
            },
            "webIdFederationData": {},
            "attributes": {
                "mfaAuthenticated": "false",
                "creationDate": "2020-12-04T19:59:53Z"
            }
        }
    },
    "eventTime": "2020-12-04T20:02:38Z",
    "eventSource": "iotfleethub.amazonaws.com",
    "eventName": "CreateApplication",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "72.22.186.61",
    "userAgent": "console.amazonaws.com",
    "requestParameters": {
        "applicationDescription": "Test application description",
```

```
            "applicationName": "Test application name",
            "clientToken": "c9bc7f45-3737-4ee9-9b0f-5de1aab169b2"
        },
        "responseElements": {
            "applicationUrl": "https://application-id.app.iotfleethub.aws",
            "applicationArn": "arn:aws:iotfleethub:us-
east-1:123456789012:application/application-id",
            "applicationId": "application-id"
        },
        "requestID": "5456304e-31c5-4336-9bbe-a375e3728eee",
        "eventID": "9ffb5d72-9267-4f4e-88e6-d26051133c8c",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "recipientAccountId": "123456789012"
    }
```

Security in Fleet Hub for AWS IoT Device Management

Cloud security at AWS is the highest priority. As an AWS customer, you benefit from a data center and network architecture that is built to meet the requirements of the most security-sensitive organizations.

Security is a shared responsibility between AWS and you. The shared responsibility model describes this as security *of* the cloud and security *in* the cloud:

- **Security of the cloud** – AWS is responsible for protecting the infrastructure that runs AWS services in the AWS Cloud. AWS also provides you with services that you can use securely. Third-party auditors regularly test and verify the effectiveness of our security as part of the AWS Compliance Programs. To learn about the compliance programs that apply to Fleet Hub, see AWS Services in Scope by Compliance Program.
- **Security in the cloud** – Your responsibility is determined by the AWS service that you use. You're also responsible for other factors including the sensitivity of your data, your company's requirements, and applicable laws and regulations.

This documentation helps you understand how to apply the shared responsibility model when using Fleet Hub for AWS IoT Device Management. The following topics show you how to configure Fleet Hub to meet your security and compliance objectives. You will also learn how to use other AWS services that help you to monitor and secure your Fleet Hub resources.

Topics

Data protection in Fleet Hub

The AWS shared responsibility model applies to data protection in Fleet Hub for AWS IoT Device Management. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with Fleet Hub or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Encryption at Rest

Fleet Hub protects data at rest through server-side encryption. For more information, see Data encryption in AWS IoT in the *AWS IoT Developer Guide*.

Encryption in transit

In cloud deployments of flows, Fleet Hub protects data in transit by using the Transport Layer Security (TLS) protocol. For more information, see Transport security in AWS IoT in the *AWS IoT Developer Guide*.

Identity and Access Management for Fleet Hub for AWS IoT Device Management

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. IAM administrators control who can be *authenticated* (signed in) and *authorized* (have permissions) to use Fleet Hub resources. IAM is an AWS service that you can use with no additional charge.

Topics

Audience

How you use AWS Identity and Access Management (IAM) differs, depending on the work that you do in Fleet Hub.

Service user – If you use the Fleet Hub service to do your job, then your administrator provides you with the credentials and permissions that you need. As you use more Fleet Hub features to do your work, you might need additional permissions. Understanding how access is managed can help you request the right permissions from your administrator. If you cannot access a feature in Fleet Hub, see Troubleshooting Fleet Hub for AWS IoT Device Management identity and access.

Service administrator – If you're in charge of Fleet Hub resources at your company, you probably have full access to Fleet Hub. It's your job to determine which Fleet Hub features and resources

your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Fleet Hub, see How Fleet Hub for AWS IoT Device Management works with IAM.

IAM administrator – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Fleet Hub. To view example Fleet Hub identity-based policies that you can use in IAM, see Identity-based policy examples for Fleet Hub for AWS IoT Device Management.

Authenticating with identities

Authentication is how you sign in to AWS using your identity credentials. You must be *authenticated* (signed in to AWS) as the AWS account root user, as an IAM user, or by assuming an IAM role.

You can sign in to AWS as a federated identity by using credentials provided through an identity source. AWS IAM Identity Center (IAM Identity Center) users, your company's single sign-on authentication, and your Google or Facebook credentials are examples of federated identities. When you sign in as a federated identity, your administrator previously set up identity federation using IAM roles. When you access AWS by using federation, you are indirectly assuming a role.

Depending on the type of user you are, you can sign in to the AWS Management Console or the AWS access portal. For more information about signing in to AWS, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see Signing AWS API requests in the *IAM User Guide*.

Regardless of the authentication method that you use, you might be required to provide additional security information. For example, AWS recommends that you use multi-factor authentication (MFA) to increase the security of your account. To learn more, see Multi-factor authentication in the *AWS IAM Identity Center User Guide* and Using multi-factor authentication (MFA) in AWS in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS account *root user* and is accessed by signing in with the email address and password that you used to create the account. We strongly recommend that you don't use the root user for your everyday tasks. Safeguard your root user credentials and use them to perform the tasks that only the root user can perform. For the complete list of tasks that require you to sign in as the root user, see Tasks that require root user credentials in the *IAM User Guide*.

Federated identity

As a best practice, require human users, including users that require administrator access, to use federation with an identity provider to access AWS services by using temporary credentials.

A *federated identity* is a user from your enterprise user directory, a web identity provider, the AWS Directory Service, the Identity Center directory, or any user that accesses AWS services by using credentials provided through an identity source. When federated identities access AWS accounts, they assume roles, and the roles provide temporary credentials.

For centralized access management, we recommend that you use AWS IAM Identity Center. You can create users and groups in IAM Identity Center, or you can connect and synchronize to a set of users and groups in your own identity source for use across all your AWS accounts and applications. For information about IAM Identity Center, see What is IAM Identity Center? in the *AWS IAM Identity Center User Guide*.

IAM users and groups

An *IAM user* is an identity within your AWS account that has specific permissions for a single person or application. Where possible, we recommend relying on temporary credentials instead of creating IAM users who have long-term credentials such as passwords and access keys. However, if you have specific use cases that require long-term credentials with IAM users, we recommend that you rotate access keys. For more information, see Rotate access keys regularly for use cases that require long-term credentials in the *IAM User Guide*.

An *IAM group* is an identity that specifies a collection of IAM users. You can't sign in as a group. You can use groups to specify permissions for multiple users at a time. Groups make permissions easier to manage for large sets of users. For example, you could have a group named *IAMAdmins* and give that group permissions to administer IAM resources.

Users are different from roles. A user is uniquely associated with one person or application, but a role is intended to be assumable by anyone who needs it. Users have permanent long-term credentials, but roles provide temporary credentials. To learn more, see When to create an IAM user (instead of a role) in the *IAM User Guide*.

IAM roles

An *IAM role* is an identity within your AWS account that has specific permissions. It is similar to an IAM user, but is not associated with a specific person. You can temporarily assume an IAM role in the AWS Management Console by switching roles. You can assume a role by calling an AWS CLI or AWS API operation or by using a custom URL. For more information about methods for using roles, see Using IAM roles in the *IAM User Guide*.

IAM roles with temporary credentials are useful in the following situations:

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Creating a role for a third-party Identity Provider in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.
- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the

principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

- **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.
- **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Using an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

To learn whether to use IAM roles or IAM users, see When to create an IAM role (instead of a user) in the *IAM User Guide*.

Managing access using policies

You control access in AWS by creating policies and attaching them to AWS identities or resources. A policy is an object in AWS that, when associated with an identity or resource, defines their permissions. AWS evaluates these policies when a principal (user, root user, or role session) makes a request. Permissions in the policies determine whether the request is allowed or denied. Most policies are stored in AWS as JSON documents. For more information about the structure and contents of JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

IAM policies define permissions for an action regardless of the method that you use to perform the operation. For example, suppose that you have a policy that allows the `iam:GetRole` action. A user with that policy can get role information from the AWS Management Console, the AWS CLI, or the AWS API.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

Identity-based policies can be further categorized as *inline policies* or *managed policies*. Inline policies are embedded directly into a single user, group, or role. Managed policies are standalone policies that you can attach to multiple users, groups, and roles in your AWS account. Managed policies include AWS managed policies and customer managed policies. To learn how to choose between a managed policy or an inline policy, see Choosing between managed policies and inline policies in the *IAM User Guide*.

Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

Resource-based policies are inline policies that are located in that service. You can't use AWS managed policies from IAM in a resource-based policy.

Access control lists (ACLs)

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Amazon S3, AWS WAF, and Amazon VPC are examples of services that support ACLs. To learn more about ACLs, see Access control list (ACL) overview in the *Amazon Simple Storage Service Developer Guide*.

Other policy types

AWS supports additional, less-common policy types. These policy types can set the maximum permissions granted to you by the more common policy types.

- **Permissions boundaries** – A permissions boundary is an advanced feature in which you set the maximum permissions that an identity-based policy can grant to an IAM entity (IAM user or role). You can set a permissions boundary for an entity. The resulting permissions are the intersection of an entity's identity-based policies and its permissions boundaries. Resource-based policies that specify the user or role in the `Principal` field are not limited by the permissions boundary. An explicit deny in any of these policies overrides the allow. For more information about permissions boundaries, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – SCPs are JSON policies that specify the maximum permissions for an organization or organizational unit (OU) in AWS Organizations. AWS Organizations is a service for grouping and centrally managing multiple AWS accounts that your business owns. If you enable all features in an organization, then you can apply service control policies (SCPs) to any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see How SCPs work in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

Multiple policy types

When multiple types of policies apply to a request, the resulting permissions are more complicated to understand. To learn how AWS determines whether to allow a request when multiple policy types are involved, see Policy evaluation logic in the *IAM User Guide*.

How Fleet Hub for AWS IoT Device Management works with IAM

Before you use IAM to manage access to Fleet Hub, learn what IAM features are available to use with Fleet Hub.

IAM features you can use with Fleet Hub for AWS IoT Device Management

IAM feature	Fleet Hub support
Identity-based policies	Yes
Resource-based policies	No
Policy actions	Yes
Policy resources	Yes
Policy condition keys	Yes
ACLs	No
ABAC (tags in policies)	Yes
Temporary credentials	Yes
Principal permissions	Yes
Service roles	Yes
Service-linked roles	No

To get a high-level view of how Fleet Hub and other AWS services work with most IAM features, see AWS services that work with IAM in the *IAM User Guide*.

Identity-based policies for Fleet Hub

Supports identity-based policies	Yes

Identity-based policies are JSON permissions policy documents that you can attach to an identity, such as an IAM user, group of users, or role. These policies control what actions users and roles can perform, on which resources, and under what conditions. To learn how to create an identity-based policy, see Creating IAM policies in the *IAM User Guide*.

With IAM identity-based policies, you can specify allowed or denied actions and resources as well as the conditions under which actions are allowed or denied. You can't specify the principal in an identity-based policy because it applies to the user or role to which it is attached. To learn about all of the elements that you can use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

Identity-based policy examples for Fleet Hub

To view examples of Fleet Hub identity-based policies, see Identity-based policy examples for Fleet Hub for AWS IoT Device Management.

Resource-based policies within Fleet Hub

Supports resource-based policies	No

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

To enable cross-account access, you can specify an entire account or IAM entities in another account as the principal in a resource-based policy. Adding a cross-account principal to a resource-based policy is only half of establishing the trust relationship. When the principal and the resource are in different AWS accounts, an IAM administrator in the trusted account must also grant

the principal entity (user or role) permission to access the resource. They grant permission by attaching an identity-based policy to the entity. However, if a resource-based policy grants access to a principal in the same account, no additional identity-based policy is required. For more information, see Cross account resource access in IAM in the *IAM User Guide*.

Policy actions for Fleet Hub

> **Note**
>
> Fleet Hub applications use the `AWSIoTFleetHubFederationAccess` managed policy. For more information, see ???.

Supports policy actions	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Action` element of a JSON policy describes the actions that you can use to allow or deny access in a policy. Policy actions usually have the same name as the associated AWS API operation. There are some exceptions, such as *permission-only actions* that don't have a matching API operation. There are also some operations that require multiple actions in a policy. These additional actions are called *dependent actions*.

Include actions in a policy to grant permissions to perform the associated operation.

To see a list of Fleet Hub actions, see Actions defined by Fleet Hub for AWS IoT Device Management in the *Service Authorization Reference*.

Policy actions in Fleet Hub use the following prefix before the action:

```
iotfleethub
```

To specify multiple actions in a single statement, separate them with commas.

```
"Action": [
      "iotfleethub:action1",
```

```
      "iotfleethub:action2"
         ]
```

To view examples of Fleet Hub identity-based policies, see Identity-based policy examples for Fleet Hub for AWS IoT Device Management.

Policy resources for Fleet Hub

Supports policy resources	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Resource` JSON policy element specifies the object or objects to which the action applies. Statements must include either a `Resource` or a `NotResource` element. As a best practice, specify a resource using its Amazon Resource Name (ARN). You can do this for actions that support a specific resource type, known as *resource-level permissions*.

For actions that don't support resource-level permissions, such as listing operations, use a wildcard (*) to indicate that the statement applies to all resources.

```
"Resource": "*"
```

To see a list of Fleet Hub resource types and their ARNs, see Resources defined by Fleet Hub for AWS IoT Device Management in the *Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions defined by Fleet Hub for AWS IoT Device Management.

To view examples of Fleet Hub identity-based policies, see Identity-based policy examples for Fleet Hub for AWS IoT Device Management.

Policy condition keys for Fleet Hub

Supports service-specific policy condition keys	Yes

Administrators can use AWS JSON policies to specify who has access to what. That is, which **principal** can perform **actions** on what **resources**, and under what **conditions**.

The `Condition` element (or `Condition` *block*) lets you specify conditions in which a statement is in effect. The `Condition` element is optional. You can create conditional expressions that use condition operators, such as equals or less than, to match the condition in the policy with values in the request.

If you specify multiple `Condition` elements in a statement, or multiple keys in a single `Condition` element, AWS evaluates them using a logical `AND` operation. If you specify multiple values for a single condition key, AWS evaluates the condition using a logical `OR` operation. All of the conditions must be met before the statement's permissions are granted.

You can also use placeholder variables when you specify conditions. For example, you can grant an IAM user permission to access a resource only if it is tagged with their IAM user name. For more information, see IAM policy elements: variables and tags in the *IAM User Guide*.

AWS supports global condition keys and service-specific condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

To see a list of Fleet Hub condition keys, see Condition keys for Fleet Hub for AWS IoT Device Management in the *Service Authorization Reference*. To learn with which actions and resources you can use a condition key, see Actions defined by Fleet Hub for AWS IoT Device Management.

To view examples of Fleet Hub identity-based policies, see Identity-based policy examples for Fleet Hub for AWS IoT Device Management.

Access control lists (ACLs) in Fleet Hub

Supports ACLs	No

Access control lists (ACLs) control which principals (account members, users, or roles) have permissions to access a resource. ACLs are similar to resource-based policies, although they do not use the JSON policy document format.

Attribute-based access control (ABAC) with Fleet Hub

Supports ABAC (tags in policies)	Yes

Attribute-based access control (ABAC) is an authorization strategy that defines permissions based on attributes. In AWS, these attributes are called *tags*. You can attach tags to IAM entities (users or roles) and to many AWS resources. Tagging entities and resources is the first step of ABAC. Then you design ABAC policies to allow operations when the principal's tag matches the tag on the resource that they are trying to access.

ABAC is helpful in environments that are growing rapidly and helps with situations where policy management becomes cumbersome.

To control access based on tags, you provide tag information in the condition element of a policy using the `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, or `aws:TagKeys` condition keys.

If a service supports all three condition keys for every resource type, then the value is **Yes** for the service. If a service supports all three condition keys for only some resource types, then the value is **Partial**.

For more information about ABAC, see What is ABAC? in the *IAM User Guide*. To view a tutorial with steps for setting up ABAC, see Use attribute-based access control (ABAC) in the *IAM User Guide*.

Using Temporary credentials with Fleet Hub

Supports temporary credentials	Yes

Some AWS services don't work when you sign in using temporary credentials. For additional information, including which AWS services work with temporary credentials, see AWS services that work with IAM in the *IAM User Guide*.

You are using temporary credentials if you sign in to the AWS Management Console using any method except a user name and password. For example, when you access AWS using your company's single sign-on (SSO) link, that process automatically creates temporary credentials. You also automatically create temporary credentials when you sign in to the console as a user and then switch roles. For more information about switching roles, see Switching to a role (console) in the *IAM User Guide*.

You can manually create temporary credentials using the AWS CLI or AWS API. You can then use those temporary credentials to access AWS. AWS recommends that you dynamically generate

temporary credentials instead of using long-term access keys. For more information, see Temporary security credentials in IAM.

Cross-service principal permissions for Fleet Hub

Supports forward access sessions (FAS)	Yes

When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.

Service roles for Fleet Hub

Supports service roles	Yes

A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*.

> **Warning**
>
> Changing the permissions for a service role might break Fleet Hub functionality. Edit service roles only when Fleet Hub provides guidance to do so.

Service-linked roles for Fleet Hub

Supports service-linked roles	No

A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS

account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.

For details about creating or managing service-linked roles, see AWS services that work with IAM. Find a service in the table that includes a Yes in the **Service-linked role** column. Choose the **Yes** link to view the service-linked role documentation for that service.

Identity-based policy examples for Fleet Hub for AWS IoT Device Management

By default, users and roles don't have permission to create or modify Fleet Hub resources. They also can't perform tasks by using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. To grant users permission to perform actions on the resources that they need, an IAM administrator can create IAM policies. The administrator can then add the IAM policies to roles, and users can assume the roles.

To learn how to create an IAM identity-based policy by using these example JSON policy documents, see Creating IAM policies in the *IAM User Guide*.

For details about actions and resource types defined by Fleet Hub, including the format of the ARNs for each of the resource types, see Actions, resources, and condition keys for Fleet Hub for AWS IoT Device Management in the *Service Authorization Reference*.

Topics

Policy best practices

Identity-based policies determine whether someone can create, access, or delete Fleet Hub resources in your account. These actions can incur costs for your AWS account. When you create or edit identity-based policies, follow these guidelines and recommendations:

- **Get started with AWS managed policies and move toward least-privilege permissions** – To get started granting permissions to your users and workloads, use the *AWS managed policies* that grant permissions for many common use cases. They are available in your AWS account. We recommend that you reduce permissions further by defining AWS customer managed policies

that are specific to your use cases. For more information, see AWS managed policies or AWS managed policies for job functions in the *IAM User Guide*.

- **Apply least-privilege permissions** – When you set permissions with IAM policies, grant only the permissions required to perform a task. You do this by defining the actions that can be taken on specific resources under specific conditions, also known as *least-privilege permissions*. For more information about using IAM to apply permissions, see Policies and permissions in IAM in the *IAM User Guide*.
- **Use conditions in IAM policies to further restrict access** – You can add a condition to your policies to limit access to actions and resources. For example, you can write a policy condition to specify that all requests must be sent using SSL. You can also use conditions to grant access to service actions if they are used through a specific AWS service, such as AWS CloudFormation. For more information, see IAM JSON policy elements: Condition in the *IAM User Guide*.
- **Use IAM Access Analyzer to validate your IAM policies to ensure secure and functional permissions** – IAM Access Analyzer validates new and existing policies so that the policies adhere to the IAM policy language (JSON) and IAM best practices. IAM Access Analyzer provides more than 100 policy checks and actionable recommendations to help you author secure and functional policies. For more information, see IAM Access Analyzer policy validation in the *IAM User Guide*.
- **Require multi-factor authentication (MFA)** – If you have a scenario that requires IAM users or a root user in your AWS account, turn on MFA for additional security. To require MFA when API operations are called, add MFA conditions to your policies. For more information, see Configuring MFA-protected API access in the *IAM User Guide*.

For more information about best practices in IAM, see Security best practices in IAM in the *IAM User Guide*.

Using the Fleet Hub console

To access the Fleet Hub for AWS IoT Device Management console, you must have a minimum set of permissions. These permissions must allow you to list and view details about the Fleet Hub resources in your AWS account. If you create an identity-based policy that is more restrictive than the minimum required permissions, the console won't function as intended for entities (users or roles) with that policy.

You don't need to allow minimum console permissions for users that are making calls only to the AWS CLI or the AWS API. Instead, allow access to only the actions that match the API operation that they're trying to perform.

To ensure that users and roles can still use the Fleet Hub console, also attach the Fleet Hub `ConsoleAccess` or `ReadOnly` AWS managed policy to the entities. For more information, see Adding permissions to a user in the *IAM User Guide*.

Allow users to view their own permissions

This example shows how you might create a policy that allows IAM users to view the inline and managed policies that are attached to their user identity. This policy includes permissions to complete this action on the console or programmatically using the AWS CLI or AWS API.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "ViewOwnUserInfo",
            "Effect": "Allow",
            "Action": [
                "iam:GetUserPolicy",
                "iam:ListGroupsForUser",
                "iam:ListAttachedUserPolicies",
                "iam:ListUserPolicies",
                "iam:GetUser"
            ],
            "Resource": ["arn:aws:iam::*:user/${aws:username}"]
        },
        {
            "Sid": "NavigateInConsole",
            "Effect": "Allow",
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

Troubleshooting Fleet Hub for AWS IoT Device Management identity and access

Use the following information to help you diagnose and fix common issues that you might encounter when working with Fleet Hub and IAM.

Topics

I am not authorized to perform an action in Fleet Hub

If the AWS Management Console tells you that you're not authorized to perform an action, then you must contact your administrator for assistance. Your administrator is the person that provided you with your sign-in credentials.

> **Note**
>
> Fleet Hub applications use the `AWSIoTFleetHubFederationAccess` managed policy. For more information, see ???.

The following example error occurs when the `mateojackson` IAM user tries to use the console to view details about a fictional *`my-example-widget`* resource but does not have the fictional `iotfleethub:`*`GetWidget`* permissions.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 iotfleethub:GetWidget on resource: my-example-widget
```

In this case, Mateo asks his administrator to update his policies to allow him to access the *`my-example-widget`* resource using the `iotfleethub:`*`GetWidget`* action.

I am not authorized to perform iam:PassRole

If you receive an error that you're not authorized to perform the `iam:PassRole` action, your policies must be updated to allow you to pass a role to Fleet Hub.

Some AWS services allow you to pass an existing role to that service instead of creating a new service role or service-linked role. To do this, you must have permissions to pass the role to the service.

The following example error occurs when an IAM user named `marymajor` tries to use the console to perform an action in Fleet Hub. However, the action requires the service to have permissions that are granted by a service role. Mary does not have permissions to pass the role to the service.

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

In this case, Mary's policies must be updated to allow her to perform the `iam:PassRole` action.

If you need help, contact your AWS administrator. Your administrator is the person who provided you with your sign-in credentials.

I want to allow people outside of my AWS account to access my Fleet Hub resources

You can create a role that users in other accounts or people outside of your organization can use to access your resources. You can specify who is trusted to assume the role. For services that support resource-based policies or access control lists (ACLs), you can use those policies to grant people access to your resources.

To learn more, consult the following:

- To learn whether Fleet Hub supports these features, see How Fleet Hub for AWS IoT Device Management works with IAM.
- To learn how to provide access to your resources across AWS accounts that you own, see Providing access to an IAM user in another AWS account that you own in the *IAM User Guide*.
- To learn how to provide access to your resources to third-party AWS accounts, see Providing access to AWS accounts owned by third parties in the *IAM User Guide*.
- To learn how to provide access through identity federation, see Providing access to externally authenticated users (identity federation) in the *IAM User Guide*.
- To learn the difference between using roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.

Compliance validation for Fleet Hub for AWS IoT Device Management

Third-party auditors assess the security and compliance of Fleet Hub as part of multiple AWS compliance programs. These include SOC, PCI, FedRAMP, HIPAA, and others.

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. AWS provides the following resources to help with compliance:

- Security and Compliance Quick Start Guides – These deployment guides discuss architectural considerations and provide steps for deploying baseline environments on AWS that are security and compliance focused.
- Architecting for HIPAA Security and Compliance on Amazon Web Services – This whitepaper describes how companies can use AWS to create HIPAA-eligible applications.

  > **Note**
  >
  > Not all AWS services are HIPAA eligible. For more information, see the HIPAA Eligible Services Reference.

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- AWS Customer Compliance Guides – Understand the shared responsibility model through the lens of compliance. The guides summarize the best practices for securing AWS services and map the guidance to security controls across multiple frameworks (including National Institute of Standards and Technology (NIST), Payment Card Industry Security Standards Council (PCI), and International Organization for Standardization (ISO)).
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – The AWS Config service assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations.

- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS. Security Hub uses security controls to evaluate your AWS resources and to check your compliance against security industry standards and best practices. For a list of supported services and controls, see Security Hub controls reference.
- Amazon GuardDuty – This AWS service detects potential threats to your AWS accounts, workloads, containers, and data by monitoring your environment for suspicious and malicious activities. GuardDuty can help you address various compliance requirements, like PCI DSS, by meeting intrusion detection requirements mandated by certain compliance frameworks.
- AWS Audit Manager – This AWS service helps you continuously audit your AWS usage to simplify how you manage risk and compliance with regulations and industry standards.

Resilience in Fleet Hub for AWS IoT Device Management

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between zones without interruption. Availability Zones are more highly available, fault tolerant, and scalable than traditional single or multiple data center infrastructures.

For more information about AWS Regions and Availability Zones, see AWS Global Infrastructure.

AWS managed policies for Fleet Hub for AWS IoT Device Management

To add permissions to users, groups, and roles, it is easier to use AWS managed policies than to write policies yourself. It takes time and expertise to create IAM customer managed policies that provide your team with only the permissions they need. To get started quickly, you can use our AWS managed policies. These policies cover common use cases and are available in your AWS account. For more information about AWS managed policies, see AWS managed policies in the *IAM User Guide*.

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles)

where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

Additionally, AWS supports managed policies for job functions that span multiple services. For example, the **ReadOnlyAccess** AWS managed policy provides read-only access to all AWS services and resources. When a service launches a new feature, AWS adds read-only permissions for new operations and resources. For a list and descriptions of job function policies, see AWS managed policies for job functions in the *IAM User Guide*.

AWS managed policy: AWSIoTFleetHubFederationAccess

You can attach the `AWSIoTFleetHubFederationAccess` policy to your IAM identities.

This policy grants Fleet Hub for AWS IoT Device Management federated users the permissions they need to take actions in AWS IoT and other AWS services from Fleet Hub web applications.

For more information about adding users to Fleet Hub web applications, see ???.

View this policy in the AWS console.

Permissions details

This policy includes the following permissions:

- `iot` - Retrieve AWS IoT device data and perform fleet-level actions.
- `iotfleethub` - Retrieve Fleet Hub app metadata.
- `cloudwatch` - Retrieve CloudWatch alarm and metric data. Also allows create and delete actions scoped to Fleet Hub alarms.
- `sns` - Perform create, read, delete, subscribe, and unsubscribe operations. These operations are scoped to Fleet Hub SNS topics.

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

```
            "Effect": "Allow",
            "Action": [
                "iot:DescribeIndex",
                "iot:DescribeThingGroup",
                "iot:GetBucketsAggregation",
                "iot:GetCardinality",
                "iot:GetIndexingConfiguration",
                "iot:GetPercentiles",
                "iot:GetStatistics",
                "iot:SearchIndex",
                "iot:CreateFleetMetric",
                "iot:ListFleetMetrics",
                "iot:DeleteFleetMetric",
                "iot:DescribeFleetMetric",
                "iot:UpdateFleetMetric",
                "iot:DescribeCustomMetric",
                "iot:ListCustomMetrics",
                "iot:ListDimensions",
                "iot:ListMetricValues",
                "iot:ListThingGroups",
                "iot:ListThingsInThingGroup",
                "iot:ListJobTemplates",
                "iot:DescribeJobTemplate",
                "iot:ListJobs",
                "iot:CreateJob",
                "iot:CancelJob",
                "iot:DescribeJob",
                "iot:ListJobExecutionsForJob",
                "iot:ListJobExecutionsForThing",
                "iot:DescribeJobExecution",
                "iot:ListSecurityProfiles",
                "iot:DescribeSecurityProfile",
                "iot:ListActiveViolations",
                "iot:GetThingShadow",
                "iot:ListNamedShadowsForThing",
                "iot:CancelJobExecution",
                "iot:DescribeEndpoint",
                "iotfleethub:DescribeApplication",
                "cloudwatch:DescribeAlarms",
                "cloudwatch:GetMetricData",
                "cloudwatch:ListMetrics",
                "sns:ListTopics"
            ],
            "Resource": "*"
```

```
        },
        {
            "Effect": "Allow",
            "Action": [
                "sns:CreateTopic",
                "sns:DeleteTopic",
                "sns:ListSubscriptionsByTopic",
                "sns:Subscribe",
                "sns:Unsubscribe"
            ],
            "Resource": "arn:aws:sns:*:*:iotfleethub*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:PutMetricAlarm",
                "cloudwatch:DeleteAlarms",
                "cloudwatch:DescribeAlarmHistory"
            ],
            "Resource": "arn:aws:cloudwatch:*:*:iotfleethub*"
        }
    ]
}
```

Fleet Hub updates to AWS managed policies

View details about updates to AWS managed policies for Fleet Hub since this service began tracking these changes. For more information, see the Fleet Hub Documentation history page.

Change	Description	Date
AWSIoTFleetHubFederationAccess – Update to an existing policy	Fleet Hub added new permissions to allow app users to retrieve AWS IoT Device Defender metric data in Fleet Hub apps.	April 4, 2022

Change	Description	Date
AWSIoTFleetHubFederationAccess – Update to an existing policy	Fleet Hub added new permissions to allow app users to retrieve additional data sources for indexing. A permission is also added to allow app users to cancel an AWS IoT job execution within the app.	November 15, 2021
AWSIoTFleetHubFederationAccess – Update to an existing policy	Fleet Hub added new permissions for app users to retrieve Thing Group data and perform CRUD operations on AWS IoT jobs.	May 24, 2021
AWSIoTFleetHubFederationAccess – Update to an existing policy	Fleet Hub removed permissions for unsupported Fleet Hub dashboard APIs.	April 12, 2021
AWSIoTFleetHubFederationAccess – New policy	Fleet Hub added a new policy that grants permissions that are needed for Fleet Hub application users to retrieve device data and perform AWS IoT actions.	April 12, 2021
Fleet Hub started tracking changes	Fleet Hub started tracking changes for its AWS managed policies.	April 12, 2021

Infrastructure security in Fleet Hub for AWS IoT Device Management

As a managed service, Fleet Hub for AWS IoT Device Management is protected by the AWS global network security procedures that are described in the Amazon Web Services: Overview of Security Processes whitepaper.

You use AWS published API calls to access Fleet Hub through the network. Clients must support Transport Layer Security (TLS) 1.2 or later. We recommend using TLS 1.3. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.

Additionally, requests must be signed by using an access key ID and a secret access key that is associated with an IAM principal. Or you can use the AWS Security Token Service (AWS STS) to generate temporary security credentials to sign requests.

Cross-service confused deputy prevention

The confused deputy problem is a security issue where an entity that doesn't have permission to perform an action can coerce a more-privileged entity to perform the action. In AWS, cross-service impersonation can result in the confused deputy problem. Cross-service impersonation can occur when one service (the *calling service*) calls another service (the *called service*). The calling service can be manipulated to use its permissions to act on another customer's resources in a way it shouldn't otherwise have permission to access. To prevent this, AWS provides tools that help you protect your data for all services with service principals that have been given access to resources in your account.

To limit the permissions that Fleet Hub gives another service to the resource, we recommend using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in resource policies. If you use both global condition context keys, the `aws:SourceAccount` value and the account in the `aws:SourceArn` value must use the same account ID when used in the same policy statement.

The most effective way to protect against the confused deputy problem is to use the `aws:SourceArn` global condition context key with the full Amazon Resource Name (ARN) of the resource. For Fleet Hub, your `aws:SourceArn` must comply with the format: `arn:aws:iot:`*`region`*`:`*`account-id`*`:*`. Make sure that the *`region`* matches your Fleet Hub Region and the *`account-id`* matches your customer account ID.

The following example shows how to prevent the confused deputy problem by using the `aws:SourceArn` and `aws:SourceAccount` global condition context keys in the Fleet Hub role trust policy. To find your Fleet Hub role ARN, go to the Fleet Hub section in the AWS IoT console and select your Fleet Hub application to view the application details page.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "iotfleethub.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "123456789012"
        },
        "ArnLike": {
          "aws:SourceArn": "arn:aws:iot:us-east-1:123456789012:*"
        }
      }
    }
  ]
}
```

Documentation history

The following table describes the updates to the documentation for Fleet Hub. For changes in AWS managed policies for Fleet Hub, see AWS managed policies for Fleet Hub for AWS IoT Device Management.

Change	Description	Date
Fleet Hub for AWS IoT Device Management general availability release	Updated content to reflect improvements made to Fleet Hub for AWS IoT Device Management during the preview period.	May 25, 2021.
Preview release of Fleet Hub for AWS IoT Device Management	Published the preview release version of the *Fleet Hub for AWS IoT Device Management User Guide*.	December 16, 2020.