aws

Developer Guide

Amazon Quick Sight

Amazon Quick Sight: Developer Guide

Table of Contents

Overview

Quick Sight is a cloud-scale business intelligence (BI) service that you can use to deliver easy-to-understand insights to the people who you work with, wherever they are.

The *Amazon Quick Sight Developer Guide* provides usage examples of API operations for Amazon Quick Sight and procedural walkthroughs of common tasks. The guide also provides examples showing how to work with Quick Sight using AWS software development kits (SDKs). By using AWS SDKs, you can access Amazon Quick Sight from your preferred programming language.

Currently, you can use the Amazon Quick Sight API to manage users and groups. In Enterprise Edition, you can also use the API to embed dashboards in your webpage or app.

To monitor the calls made to the Amazon Quick Sight API for your account, use AWS CloudTrail. CloudTrail can monitor calls made by the AWS Management Console, command line tools, and other services. For more information, see the AWS CloudTrail User Guide.

Following, you can find out how to get started using the Amazon Quick Sight API:

- Terminology and concepts
- Get started with the Amazon Quick Sight API
- Amazon Resource Names (ARNs) in Quick Sight
- Operations

Are you a first-time Amazon Quick Sight user?

If you are a first-time user of Amazon Quick Sight, we recommend that you begin by reading the following sections in the *Amazon Quick Suite User Guide*:

- How Amazon Quick Suite Works
- Getting Started with Quick Suite
- AWS Security in Quick Suite

Terminology and concepts

Following, you can find a list of terms and concepts used to describe Amazon Quick Sight development in the *Amazon Quick Sight Developer Guide*.

Anonymous Quick Sight user – A temporary Quick Sight user identity that virtually belongs to a namespace and that you can use only with embedding. You can use tag-based rules to implement row-level security for such users.

Caller identity – The identity of the IAM user making an API request. The identity of the caller is determined by Quick Sight using the signature attached to the request. Through the use of our provided SDK clients, no manual steps are necessary to generate the signature or attach it to the requests. However, you can do it manually if you want to.

Invoker identity – In addition to the caller identity, but not as a replacement for it, you can assume a caller's identity through the IAM `AssumeRole` API operation when making calls to Quick Sight. AWS approves callers through their invoker's identity. This approval means that you can avoid having to explicitly add multiple accounts belonging to the same Quick Sight subscription.

Namespace – A logical container that you can use to isolate user pools so that you can organize clients, subsidiaries, teams, and so on.

Quick Sight ARN – Amazon Resource Name (ARN). Quick Sight resources are identified using their name or ARN. For example, the following are ARNs for a group named `MyGroup1`, a user named `User1`, and a dashboard with the ID `1a1ac2b2-3fc3-4b44-5e5d-c6db6778df89`.

```
arn:aws:quicksight:us-east-1:111122223333:group/default/MyGroup1
  arn:aws:quicksight:us-east-1:111122223333:user/default/User1
  arn:aws:quicksight:us-west-2:111122223333:dashboard/1a1ac2b2-3fc3-4b44-5e5d-
c6db6778df89
```

The following examples show ARNs for a template named `MyTemplate` and a dashboard named `MyDashboard`.

- The following is the sample ARN for a template.

  ```
  arn:aws:quicksight:us-east-1:111122223333:template/MyTemplate
  ```

- The following is the sample ARN for a template, referencing a specific version of the template.

```
arn:aws:quicksight:us-east-1:111122223333:template/MyTemplate/version/10
```

- The following is the sample ARN for a template alias.

```
arn:aws:quicksight:us-east-1:111122223333:template/MyTemplate/alias/STAGING
```

- The following is the sample ARN for a dashboard.

```
arn:aws:quicksight:us-east-1:111122223333:dashboard/MyDashboard
```

- The following is the sample ARN for a dashboard, referencing a specific version of the dashboard.

```
arn:aws:quicksight:us-east-1:111122223333:dashboard/MyDashboard/version/10
```

Depending on the scenario, you might need to provide an entity's name, ID, or ARN. You can retrieve the ARN if you have the name, using some of the Quick Sight API operations.

Quick Sight dashboard – An entity that identifies Quick Sight reports, created from analyses or templates. You can share Quick Sight dashboards. With the right permissions, you can create scheduled email reports from them. The `CreateDashboard` and `DescribeDashboard` API operations act on the dashboard entity.

Quick Sight template – An entity that encapsulates the metadata required to create an analysis or a dashboard. It abstracts the dataset associated with the analysis by replacing it with placeholders. You can use templates to create dashboards by replacing dataset placeholders with datasets. These datasets need to follow the same schema that was used to create the source analysis and template.

Quick Sight user – This is an Quick Sight user identity acted on by your API call. This user isn't identical to the caller identity but might be the one that maps to the user in Quick Sight.

Get started with the Amazon Quick Sight API

You can manage most aspects of your deployment with the AWS SDKs to access an API that's tailored to the programming language or platform that you're using. For more information, see AWS SDKs. For more information about specific API operations, see Quick Sight API Reference.

Use the topics in this section to get started with the Quick Sight API and SDKs.

Topics

Prerequisites

If you plan to access Quick Sight through its API, make sure you're familiar with the following:

- JSON
- Web services
- HTTP requests
- One or more programming languages, such as JavaScript, Java, Python, or C#

We recommend visiting the AWS Getting Started Resource Center for a tour of what AWS SDKs and toolkits have to offer.

Although you can use a terminal and your favorite text editor, you might benefit from the more visual UI experience you get in an integrated development environment (IDE). We provide a list of IDEs in the AWS Getting Started Resource Center in the IDE and IDE Toolkits section. This site provides AWS toolkits that you can download for your preferred IDE. Some IDEs also offer tutorials to help you learn more about programming languages.

Before you can call the Quick Sight API operations, make sure that you have the `quicksight:`*`operation-name`* permission in an IAM policy attached to your IAM identity. For example, to call `list-users`, you need the permission `quicksight:ListUsers`. The same pattern applies to all operations.

If you're not sure what the necessary permission is, you can attempt to make a call. The client then tells you what the missing permission is. You can use asterisk (*) in the Resource field of your permission policy instead of specifying explicit resources. However, we recommend that you restrict each permission as much as possible. You can restrict user access by specifying or excluding resources in the policy, using their Quick Sight Amazon Resource Name (ARN) identifier.

For more information, see the following:

- IAM Policy Examples in the *Amazon Quick Sight User Guide*
- Actions, Resources, and Condition Keys in the *IAM User Guide*
- IAM JSON Policy Elements in the *IAM User Guide*

To retrieve the ARN of a user or a group, use the `Describe` operation on the relevant resource. You can also add conditions in IAM to further restrict access to an API in some scenarios. For instance, when adding `User1` to `Group1`, the main resource is `Group1`, so you can allow or deny access to certain groups. However, you can also add a condition by using the Quick Sight IAM key `quicksight:UserName` to allow or prevent certain users from being added to that group.

Following is an example policy. It means that the caller with this policy attached can invoke the `CreateGroupMembership` operation for any group, if the user name they are adding to the group isn't `user1`.

```
{
 "Effect": "Allow",
 "Action": "quicksight:CreateGroupMembership",
 "Resource": "arn:aws:quicksight:us-east-1:aws-account-id:group/default/*",
 "Condition": {
  "StringNotEquals": {
   "quicksight:UserName": "user1"
  }
 }
}
```

AWS CLI

The following procedure explains how to interact with Quick Sight API operations through the AWS Command Line Interface (AWS CLI). The following instructions have been tested in Bash but should be identical or similar in other command-line environments.

To use Quick Sight API operations through the AWS CLI

1. Install AWS SDK in your environment. For instructions, see AWS Command line Interface.
2. Set up your AWS CLI identity and AWS Region using the following command and follow-up instructions. Use the credentials for an IAM identity or role that has the proper permissions.

```
aws configure
```

3. Look at Quick Sight SDK help by running the following command.

```
aws quicksight help
```

4. To get detailed instructions on how to use an API, enter its name followed by `help`, as follows.

```
aws quicksight list-users help
```

5. Call an Quick Sight API operation. The following example returns a list of Quick Sight users in your account.

```
aws quicksight list-users \
--aws-account-id aws-account-id \
--namespace default \
--region us-east-1
```

Java SDK

Use the following procedure to set up a Java app that interacts with Quick Sight.

To set up a Java app that works with Quick Sight

1. Create a Java project in your IDE.
2. Import the Quick Sight SDK into your new project, for example: `AWSQuickSightJavaClient-1.11.x.jar`
3. After your IDE indexes the Quick Sight SDK, add an import line as follows.

```
import com.amazonaws.services.quicksight.AmazonQuickSight;
```

If your IDE doesn't recognize line this as valid, verify that you imported the SDK.

4. Download and import external dependencies for the Quick Sight SDK.

   Like other AWS SDKs, Quick Sight SDK requires external dependencies to perform many of its functions. Make sure to download and import those into the same project. The following dependencies are required:

   - `aws-java-sdk-1.11.402.jar` (AWS Java SDK and credentials setup) – To download, see Set up the AWS SDK for Java in the SDK for Java documentation.
   - `commons-logging-1.2.jar` – To download, see Download Apache Commons Logging on the Apache Commons website.
   - `jackson-annotations-2.9.6.jar`, `jackson-core-2.9.6.jar`, and `jackson-databind-2.9.6.jar` – To download, see the Maven repository.
   - `httpclient-4.5.6.jar`, `httpcore-4.4.10.jar` – To download, see the Apache site.
   - `joda-time-2.1.jar` – To download, see the MVNrepository Joda Time site.

5. Create an Quick Sight client.

   You can use a default public endpoint that the client can communicate with, or you can reference the endpoint explicitly. There are multiple ways to provide your AWS credentials. In the following example, we provide a direct, simple approach.

   The following client method is used to make all the API calls that follow.

```
private static AmazonQuickSight getClient() {
 final AWSCredentialsProvider credsProvider = new AWSCredentialsProvider() {
 @Override
 public AWSCredentials getCredentials() {
 // provide actual IAM access key and secret key here
 return new BasicAWSCredentials("access-key", "secret-key");
 }

 @Override
 public void refresh() {}
 };

 return AmazonQuickSightClientBuilder
 .standard()
 .withRegion(Regions.US_EAST_1.getName())
 .withCredentials(credsProvider)
 .build();
```

```
}
```

6. Use the client that you just created to list all the users in our Quick Sight account.

   Provide the AWS account ID that you used to subscribe to Quick Sight. This ID must match the AWS account ID of the caller's identity. Cross-account calls aren't supported at this time. Also, make sure that the required parameter `namespace` is set to *`default`*.

```
getClient().listUsers(new ListUsersRequest()
 .withAwsAccountId("relevant_AWS_account_ID")
 .withNamespace("default"))
 .getUserList().forEach(user -> {
  System.out.println(user.getArn());
 });
```

7. See a list of all possible API operations and the request objects they use by choosing the CTRL key and clicking the client object in your IDE view of the Quick Sight interface. Or find this list in the `com.amazonaws.services.quicksight` package in the Quick Sight JavaClient .jar file.

JavaScript (Node.js) SDK

Use the following procedure to interact with Quick Sight using Node.js.

To work with Quick Sight using Node.js

1. Set up your node environment using the following commands:

   - `npm install aws-sdk`
   - `npm install aws4`
   - `npm install request`
   - `npm install url`

   For information on configuring the Node.js with AWS SDK and setting your credentials, see the AWS SDK for JavaScript Developer Guide for SDK v2.

2. Use the following code example to test your setup. HTTPS is required. The example displays a full listing of Quick Sight operations along with their URL request parameters, followed by a list of Quick Sight users in your account.

```
const AWS = require('aws-sdk');
const https = require('https');

var quicksight = new AWS.Service({
 apiConfig: require('./quicksight-2018-04-01.min.json'),
 region: 'us-east-1',
});

console.log(quicksight.config.apiConfig.operations);

quicksight.listUsers({
 // Enter your actual AWS account ID
 'AwsAccountId': 'relevant_AWS_account_ID',
 'Namespace': 'default',
}, function(err, data) {
 console.log('---');
 console.log('Errors: ');
 console.log(err);
 console.log('---');
 console.log('Response: ');
 console.log(data);
});
```

Python3 SDK

Use the following procedure to create a custom-built `botocore` package to interact with Quick Sight.

To create a custom botocore package to work with Quick Sight

1. Create a credentials file in the AWS directory for your environment. In a Linux- or macOS-based environment, that file is called `~/.aws/credentials` and looks like the following.

```
[default]
aws_access_key_id = Your_IAM_access_key
aws_secret_access_key = Your_IAM_secret_key
```

2. Unzip the folder `botocore-1.12.10`. Change directory into `botocore-1.12.10` and enter the Python3 interpreter environment.

Each response comes back as a dictionary object. They each have a `ResponseMetadata` entry that contains request IDs and response status. Other entries are based on what type of operation you run.

3. As a test, use the following example code, a sample app that first creates, deletes, and lists groups. Then it lists users in a Quick Sight account.

```
import botocore.session
default_namespace = 'default'
account_id = 'relevant_AWS_Account'

session = botocore.session.get_session()
client = session.create_client("quicksight", region_name='us-east-1')

print('Creating three groups: ')
client.create_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup1')
client.create_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup2')
client.create_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup3')

print('Retrieving the groups and listing them: ')
response = client.list_groups(AwsAccountId = account_id,
 Namespace=default_namespace)
for group in response['GroupList']:
 print(group)

print('Deleting our groups: ')
client.delete_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup1')
client.delete_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup2')
client.delete_group(AwsAccountId = account_id, Namespace=default_namespace,
 GroupName='MyGroup3')

response = client.list_users(AwsAccountId = account_id,
 Namespace=default_namespace)
for user in response['UserList']:
 print(user)
```

.NET/C# SDK

Use the following procedure to interact with Quick Sight using C#.NET. This example is constructed on Microsoft Visual for Mac; the instructions can vary slightly based on your IDE and platform.

To work with Quick Sight using C#.NET

1. Unzip the `nuget.zip` file into a folder called `nuget`.
2. Create a new **Console app** project in Visual Studio.
3. Under your solution, locate app **Dependencies**, then open the context (right-click) menu and choose **Add Packages**.
4. In the sources list, choose **Configure Sources**.
5. Choose **Add**, and name the source `QuickSightSDK`. Browse to the `nuget` folder and choose **Add Source**.
6. Choose **OK**. Then, with `QuickSightSDK` selected, select all three Quick Sight packages:
   - `AWSSDK.QuickSight`
   - `AWSSDK.Extensions.NETCore.Setup`
   - `AWSSDK.Extensions.CognitoAuthentication`
7. Choose **Add Package**.
8. Copy and paste the following sample app into your console app editor.

```
using System;
using Amazon.QuickSight.Model;
using Amazon.QuickSight;

namespace DotNetQuickSightSDKTest
{
 class Program
 {
  private static readonly string AccessKey = "insert_your_access_key";
  private static readonly string SecretAccessKey = "insert_your_secret_key";
  private static readonly string AccountID = "AWS_account_ID";
  private static readonly string Namespace = "default";  // leave this as
 default

  static void Main(string[] args)
  {
```

```
    var client = new AmazonQuickSightClient(
     AccessKey,
     SecretAccessKey,
     Amazon.RegionEndpoint.USEast1);

    var listUsersRequest = new ListUsersRequest
    {
     AwsAccountId = AccountID,
     Namespace = Namespace
    };

    client.ListUsersAsync(listUsersRequest).Result.UserList.ForEach(
     user => Console.WriteLine(user.Arn)
    );

    var listGroupsRequest = new ListGroupsRequest
    {
     AwsAccountId = AccountID,
     Namespace = Namespace
    };

    client.ListGroupsAsync(listGroupsRequest).Result.GroupList.ForEach(
     group => Console.WriteLine(group.Arn)
    );
   }
  }
}
```

Make API requests with the Quick Sight SDKs

You can use API operations for Amazon Quick Sight and AWS SDKs to access Quick Sight from your preferred programming language. Currently, you can use the Amazon Quick Sight API to manage users and groups. In Enterprise Edition, you can also use the API to embed dashboards in your webpage or app.

To monitor the calls made to the Quick Sight API for your account, including calls made by the AWS Management Console, command line tools, and other services, use AWS CloudTrail. For more information, see the AWS CloudTrail User Guide.

AWS provides libraries, sample code, tutorials, and other resources for software developers who prefer to build applications using language-specific API operations instead of submitting a request

over HTTPS. These libraries provide basic functions that automatically take care of tasks such as cryptographically signing your requests, retrying requests, and handling error responses. These libraries help make it easier for you to get started.

For more information about downloading the AWS SDKs, see AWS SDKs and Tools. The following links are a sample of the language-specific API documentation available.

AWS Command Line Interface

- AWS CLI Quick Sight Command Reference
- AWS CLI User Guide
- AWS CLI Command Reference

AWS SDK for .NET

- Amazon.Quicksight
- Amazon.Quicksight.Model

AWS SDK for C++

- Aws::QuickSight::QuickSightClient Class Reference

AWS SDK for Go

- quicksight

AWS SDK for Java

- QuickSightClient
- QuickSightModel

AWS SDK for JavaScript

- AWS.QuickSight

AWS SDK for PHP

- QuickSightClient

AWS SDK for Python (Boto3)

- QuickSight

AWS SDK for Ruby

- Aws::QuickSight

Use CLI skeleton files

To run AWS CLI commands that require long and complicated strings, you can generate CLI skeleton files . A *CLI skeleton* is a JSON file that provides you with an outline, or skeleton, of the command that you want to run. You can use a CLI skeleton file for every Quick Sight command, but skeleton files are most useful when using `Create` or `Update` commands.

Generate a CLI skeleton file

To generate a CLI skeleton file, enter the following command into your terminal.

```
aws quicksight OPERATION --generate-cli-skeleton
```

A JSON file that contains a skeleton of the command that you want to run then appears in your terminal. Enter the required input values and save the file. The following example shows a cli example that is generated for the `UpdateDashboardPermissions` API.

```
$ aws quicksight update-dashboard-permissions --generate-cli-skeleton
{
 "AwsAccountId": "",
 "DashboardId": "",
 "GrantPermissions":[
  {
   "Principal": "",
   "Actions": [
    ""
```

```
   ]
  }
 ],
 "RevokePermissions": [
  {
   "Principal": "",
   "Actions": [
    ""
   ]

  }
 ]

}
```

Enter the following to make a CLI command using the saved skeleton file.

```
aws quicksight COMMAND --cli-input-json file://filename.json
```

You can update and reuse CLI skeleton files to run future commands.

Operations that skeleton files are most useful for

You can use a CLI skeleton file for every command in Quick Sight. However, skeleton files are most useful for commands that require long or complicated string inputs, such as a permissions update.

Following is a list of Quick Sight operations where we recommend using a CLI skeleton file:

Account customization operations

- CreateAccountCustomization
- UpdateAccountCustomization
- UpdateAccountSettings

Analysis operations

- CreateAnalysis
- UpdateAnalysis
- UpdateAnalysisPermissions

Dashboard operations

- CreateDashboard
- UpdateDashboard
- UpdateDashboardPermissions
- UpdateDashboardPublishedVersion

Dataset operations

- CreateDataSet
- UpdateDataSet
- UpdateDataSetPermissions

Data source operations

- CreateDataSource
- UpdateDataSource
- UpdateDataSourcePermissions

Folder operations

- CreateFolder
- CreateFolderMembership
- UpdateFolder
- UpdateFolderPermissions

Group operations

- CreateGroup
- CreateGroupMembership
- UpdateGroup

IAM policy assignment operations

- CreateIAMPolicyAssignment
- UpdateIAMPolicyAssignment

Ingestion operations

- CreateIngestion

Namespace operations

- CreateNamespace

Template operations

- CreateTemplate
- UpdateTemplate
- UpdateTemplatePermissions

Template alias operations

- CreateTemplateAlias
- UpdateTemplateAlias

Theme operations

- CreateTheme
- UpdateTheme
- UpdateThemePermissions

Theme alias permissions

- CreateThemeAlias
- UpdateTemplateAlias

User operations

- RegisterUser
- UpdateUser

Use the Quick Sight Dev Portal

The Dev Portal helps you learn by example how to use the Quick Sight API in your website or application. Currently, the Dev Portal focuses on API operations for embedded analytics.

The portal provides easy-to-use code samples to get you started. You can choose from the following three different use cases:

- Displaying embedded dashboards to everyone (nonauthenticated users)
- Personalizing dashboards for your users
- Embedding dashboard authoring

The portal itself displays dashboards by using embedding for everyone.

To get started with the Dev Portal

1. Open the Dev Portal and choose **Try it** on the use case that you want to view.
2. View code examples by choosing **How to embed it** in the menu bar. Then choose each of the following from the navigation pane at left:
   - Configure permissions
   - Get embedding URL (code samples in Java, JavaScript, and Python)
   - Embed URL in your application
3. Choose **Download all code** to download all of the code in a .zip file.
4. Choose **How to customize it** to customize the dashboard, This screen is interactive, so you can choose any item in the navigation pane to view the changes live.
5. View and download the HTML code at bottom left.
6. Choose the Quick Sight icon at upper left to return to the start page.

Embed assets with Quick Sight

Quick Sight Embedded Analytics allows users to integrate business analytics into their own applications and web portals. This empowers end users to gain deeper and faster insights through the embedded assets to allow for for easier in-depth analysis of your data. Use Quick Sight embedded analytics to provide critical business insights from your Software as a Service (SaaS) product, scale and modernize your enterprise's application on the cloud, and leverage Generative Business Intelligence (BI) to enable end users to use natural language and discover actionable insights.

Businesses can embed rich data visuals, interactive dashboards, and advanced ML-powered analytics in minutes. You can embed the full dashboard-building experience within a portal or application, including the Quick Sight home page, search, and data experiences. This allows you to provide ad-hoc data exploration and author capabilities to your application's power users who want to explore usage data, create specific views as dashboards, and share their creations with other users or groups in their organization.

Specialized expertise is not required for your team to develop, maintain, and evolve the analytics components for your applications. Teams can easily manage and scale your analytics servers, manage complex data engineering pipelines and infrastructure as your applications' popularity grows.

Improve your Business Intelligence with Quick Sight Embedded Analytics

The use cases below highlight how an embedded analytics tool benefits independent software vendors as well as the program managers and developers at your enterprise.

- Add advanced analytics capabilities and natural language processing within your Software as a Service (SaaS) product to improve the overall user experience for the end-users.
- Integrate scheduled business intelligence reports directly into your enterprise, which streamlines the availability of data analytics.

How Embedded Analytics Can Transform Your Application

The list below describes 9 ways embedded analytics can be used to transform your applications:

1. **Leverage Generative BI and Natural Language Processing to your business data**

   Embed powerful AI tools into your application to empower your users with deeper insights and accelerate decision making for your business.

2. **Save development time and resources by using enterprise-ready interactive dashboards and visuals**

   Embedded analytics saves valuable engineering time by providing enterprise-ready dashboards and visuals that can be used right away. Once they ae embedded, you can use these embedded dashboards to analyze data and derive actionable insights.

3. **Supports highly advanced analytics**

   You can unlock a multitude of advanced features powered by Amazon Quick Sight for your embedded analytics, with new features being released by Quick Sight frequently.

4. **Discover, scale, and securely share valuable business insights at a rapid pace**

   Costly development efforts needed to integrate, manage, analyze, secure, and share data are minimized with Quick Sight embedded analytics. Quick Sight embedded analytics can also automatically scale up and down depending on your organization's infrastructural needs.

5. **Embed analytics seamlessly to your application**

   You can customize and personalize the look, feel, and functionality of your Quick Sight embedded analytics to best meet your requirements. This customization capability allows you to refine available analytics features to align with your brand style. You can customize UX elements like theming and styling of your analytics to match your brand needs.

6. **Embedding analytics into your application is easy with AWS cloud technologies**

   Embedded analytics are designed to be used easily. Quick Sight provides multiple ways to embed feature rich analytics to your application to meet your business requirements, ranging from 1-click copy and paste to advanced API integration with AWS cloud technologies support.

7. **Self-Service BI Capabilities**

   Everyone in your organization can create custom dashboards and reports without the need to rely on your IT professionals and dedicated data analysts.

8. **Centralize your analytics**

Embedded analytics increases operational efficiency and boosts productivity by allowing all analytics to be available in one interface. This eliminates the need to toggle between different platforms to find what you need.

9. **Distribute up to date and extensive customer facing reports directly from your application**

   You can schedule automated delivery of reports to ensure that stakeholders receive timely updates without the need to access your application.

What to consider when using embedded analytics

Review the considerations below before you get started with Quick Sight embedded analytics:

1. **How much analytic capability would you like to give your users?**

   You can embed different Quick Sight experiences in your application that are tailored to how users want to interact with their business intelligence. Quick Sight console embedding gives end users the ability to author dashboards and visuals from your application. Quick Sight dashboard embedding gives users the power to filter data and create reports. Visual embedding allows you to place individual analytics anywhere on your page and create your own interactive, customizable inline view.

2. **How do you want to manage users and govern their access to embedded analytics?**

   Quick Sight offers you flexibility with user management by allowing you to serve embedded content with or without the need to provison users in Quick Sight. User management through Quick Sight allows you to delegate the responsibility of capturing user preferences and managing user permissions to Quick Sight. Alternatively, if your application already has its own user context, you can choose to embed anonymously without the overhead of user provisioning. With this, you can still govern data access using runtime tag-based row-level security (RLS).

3. **How unique do you want your embedded experience to be?**

   Embedded Analytics supports extensive customization features to ensure that the data visualization experience matches your brand. You can customize Quick Sight to match the style of your brand, create custom controls from your application that interact with embedded content, and update your application based on events that happen in the embedded experience.

4. **What type of application do you want to embed into?**

Quick Sight embedded analytics can securely integrate and scale into all application types. Enterprise applications can have personalized, authenticated user experiences which can integrate into your directory system. Independent Software Vendors (ISVs) can serve embedded dashboards and visuals to end users without the need to sign in to Quick Sight.

5. **Should you use natural language insights to visualize data or interactive dashboards?**

   With embedded analytics and Amazon Q, you can embed Natural Language Processing (NLP) for your business data into your application. End users use an embedded search bar to ask questions and visualize data.

Get Started with Quick Sight embedded analytics

Amazon Quick Sight is a scalable, embeddable, ML-powered BI Service built for the cloud. Embedded Analytics allows you to easily extend your visuals, dashboards, and Natural Language Queries (Q) to your apps, websites, and portals.

Topics

Prerequisites

Before you get started, familiarize yourself with the list of technologies Quick Sight uses to create an embedding experience. Check that the technologies listed below are compatible with your application:

- Embedding utilizes Iframes to display your content and MessageChannels to communicate.
- If you're developing a JavaScript-based front-end application, we ecommend you use the Embedding Quick Sight data dashboards for registered users in your application to leverage performance, customization, and interactivity capabilities offered through the SDK for your embedded content.
- A backend service that is compatible with one of the languages supported in the AWS SDK.
- Many web applications use CSP to add security on what can be loaded within the application. Ensure you have ability to allowlist Quick Sight domains in your CSP.

- Make sure you are using one of our supported browsers.

After you confirm that your application is compatible with Quick Sight embedding, complete the steps listed in Getting started with Amazon Quick Sight.

Choose the right embedding solution

Quick Sight offers multiple options and mechanisms of embedding. You can embed a visual, a dashboard, the complete Quick Sight console, or the natural language component Q in your application. Furthermore, based on your organizations authentication setup, you can choose between anonymous and registered user embedding. All of these options are possible when you onboard the Quick Sight Embedding APIs.

Anonymous user embedding

Anonymous user embedding offers a lightweight option for you to bring insights to your users without the need to register them in Quick Sight. You can use anonymous embedding to share your dashboards to any number of users in your application and scale as you go. With row-level security, you can further customize your data based on pre-determined context and showcase relevant insights to your users.

To get started with anonymous embedding, see Embedding Quick Sight data dashboards for anonymous (unregistered) users.

Registered user embedding

Use registered user embedding to bring Quick Sight to every user in your organization. Use Quick Sight APIs to share dashboards with users or grant them access to build dashboards from scratch, all within the context of your application. Use comprehensive user context based features like row-level security or column-level-security for fine grained data access control. You can also use features like bookmarks to deliver personalized experiences.

To get started with registered user embedding, see Embedding Quick Sight data dashboards for registered users.

1-click Enterprise embedding

1-click Enterprise embedding is focused toward enterprises that have Quick Sight accounts set up for all of their users. With 1-click embedding, developers embed Quick Sight visuals and dashboards with a static embed code from Quick Sight that is added to an <iframe>. When a user

accesses a dashboard on your intranet enterprise applications, they are required to sign in to Quick Sight.

To get started with 1-click embedding, see 1-click Enterprise embedding.

1-click public embedding

Use 1-click public embedding to take your insights to public websites with a few clicks. Developers receive an embed code from Quick Sight and add it to an <iframe>. You have full control over when dashboards go public with sharing settings that are available on the dashboard. In case of an emergency, you can revoke public sharing within seconds.

To get started with 1-click public embedding, see Turn on public access to visuals and dashboards with a 1-click embed code.

Use the table below to compare the different features of each embedding option:

Embedd option	Embed in SaaS application	Embed in an interna applica on	Embed in a public portal	Embeddable content	RLS suppor	Require user setup	Coding and insfrastr ucture required
Anonym embedc	✓	✓	Maybe*	Dashboards, Visuals, Q	✓ (with RLS Tags)	—	✓
Registe d user embedc	✓	✓	—	Dashboards, Visuals, Q	✓	✓	✓
1-click Enterpri e embedc	Not recommended	✓	—	Dashboards, Visuals	✓	✓	—
1-click	—	—	✓	Dashboards, Visuals	—	—	—

Embedd option	Embed in SaaS application	Embed in an interna applica on	Embed in a public portal	Embeddable content	RLS suppor	Require user setup	Coding and insfrastr ucture required
public embedc							

*Anonymous embedding delegates the responsibility of authentication to the 3P application. If the 3P application does not have user authentication, the embedded content becomes publicly accessible.

Create your first embedding application

The quickest and most flexible way to embed a dashboard in your web application is to get an embed URL through the Quick Sight API and load that onto your application using the Quick Sight Embedding SDK, in an iFrame. To do so, you will need:

- A backend service to generate the embed URL
- An endpoint to pass the embed url to your front end application
- (Optional) A Javascript-based front-end application that leverages the embedding SDK to load the dashboard within an iFrame
- (Optional) Front-end methods to customize and integrate with the embedded dashboard seamlessly with your application using functions in the embedding SDK

To embed a dashboard in a React application, see this example.

To set up your first embedded dashboard, see Embedding Quick Sight data dashboards for registered users.

To set up a different kind of embedded asset, shooce one of the following options:

Embedding options for registered users

- Embedding Quick Sight visuals for registered users
- Embed the full functionality of the Quick Sight console for registered users

- Embed the Generative Q&A experience for registered users

Embedding options for anonymous (unregistered) users

- Embed data dashboards for anonymous (unregistered) users
- Embed visuals for anonymous (unregistered) users
- Embed the Generative Q&A experience for anonymous (unregistered) users

1-click embedding options

- Turn on public access to visuals and dashboards with a 1-click embed code
- Embedding visuals and dashboards for registered users with a 1-click embed code

Customize embedded assets

Use Quick Sight embedded analytics to embed custom Quick Sight assets into your application that are tailored to meet your business needs. For embedded dashboards and visuals, Quick Sight authors can add filters and drill downs that readers can access as they navigate the dashboard or visual. Quick Sight developers can also use the Quick Sight SDKs to build tighter integrations between their SaaS applications and their Quick Sight embedded assets to add datapoint callback actions to visuals on a dashboard at runtime.

For more information about the Quick Sight SDKs, see the `amazon-Quick Sight-embedding-sdk` on GitHub or NPM.

Use the sections listed below to find descriptions of how to use the Quick Sight SDKs to customize your Quick Sight embedded analytics:

Topics

- Control the look and feel of embedded assets
- Add interactivity to your embedded content
- Personalization

Control the look and feel of embedded assets

Amazon Quick Sight embedded experiences offer solutions to developers who want to make embedded content integrate seamlessly with their application. With the embedding SDK, developers can have control over the dimensions, locale, and theming of their embedded dashboards, visuals, and Q Search Bars. These customizations reduce friction for users and make Quick Sight feel like a part of your app.

To simplify user experience, you can control which toolbar options are available and whether the Q search bar topic can be changed. Embedded content dimensions can be configured and resized to fit into your app and custom themes can be applied to better suit your brand identity.

To learn more, see Runtime theming.

Add interactivity to your embedded content

Amazon Quick Sight embedded content offers options for developers to interact with their embedded experiences to create a cohesive experience between their application and Quick Sight with the embedding SDK. Quick Sight allows application developers to control the experience given to users and to react to user actions taken within their embedded analytics.

Processes can be made more efficient by giving users the information that they need when they need it, such as by selecting the relevant sheet or filtering the data to what is relevant to the user. For exmaple, users can use preconfigured visual actions to focus on individual datapoints and carry out tasks in your application without needing to leave Quick Sight.

The embedding SDK allows for bi-directional interactions between your app and Quick Sight. Developers can respond to users changing parameters, switching sheets, clicking on datapoints, and more. From your app to Quick Sight, you can set parameters, Q search bar questions, and undo or reset the content state.

To learn more about custom assets that can be configured in the embedding SDK, see Dynamic filtering – set parameters, set filters and Respond to events – callbacks

Personalization

Amazon Quick Sight enhances user experience with state persistence to ensure a seamless transition across sessions with the maintenance of the current dashboard state. This feature allows for uninterrupted workflows and a consistent analytics experience. Furthermore, users can optimize their productivity with bookmarks to save and revisit specific views of their dashboard.

Personalization features including state persistence and bookmarks are only available for registered users. To integrate these capabilities into embedded analytics, developers can leverage the Quick Sight APIs. This API-driven approach allows developers to customize personalization features within embedded analytics. The Quick Sight APIs provide developers with the tools they need to create a tailored and efficient user experience to meet their business needs.

State Persistence

Use state persistence to ensure a continuous user experience that maintains the current state of a dashboard across different sessions. This means that Quick Sight retains information about filters, selected tabs, and other configurations When a user revisits a dashboard, they can pick up where they left off, which eliminates the need to recreate the view each time. State persistence can be utilized to improve user productivity, enhance collaboration, and promote efficient data exploration.

Bookmarks

Users can utilize bookmarks to save and revisit specific views within Quick Sight dashboards to enhance the efficiency and flexibility of data exploration. Bookmarks can be used to improve user productivity, enmahce collaboration, and create user defined views of a Quick Sight dashboard.

To learn more about bookmarks, see Bookmarking views of a dashboard and GenerateEmbedUrlForRegisteredUser.

Embedding security

Amazon Quick Sight provides a secure platform that allows you to distribute dashboards and insights to tens of thousands of users with multiple-region availability and built-in redundancy. 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.

Quick Sight manages who sees content

By default, Quick Sight only allows users who have access to content in the console see that same content in an embedded view. For anonymous (unregistered) users, content access can be governed with row level security (RLS) tags. Additionally, Quick Sight has the capability to share assets to anyone on the internet with 1-click public embedding.

Quick Sight manages where you see content

Quick Sight offers a variety of solutions to control where embedding can take place. To ensure embedding is only done intentionally, Quick Sight will only embed on domains that are allow-listed. You can add static domains to your allow-list through the Quick Sight console, or you can dynamically add a domain at runtime. Additionally, you can limit access to your organization's Quick Sight account to a predefined list of Internet Protocol (IP) address ranges.

Quick Sight manages what you see

Quick Sight allows you to restrict access to a dataset. You can do this before or after you have shared the dataset. When a dataset owner views the content, they can still see all the data. When you share the dataset with readers, they can only see the data applicable to them individually, as restricted by the permission dataset rules.

Amazon Resource Names (ARNs) in Quick Sight

Amazon Resource Names (ARNs) uniquely identify AWS resources. An ARN identifies a resource unambiguously across all of AWS, for example in IAM policies, Amazon Relational Database Service (Amazon RDS) tags, and API calls. To retrieve the ARN of an Quick Sight resource, you can use the `Describe` operation on the relevant resource.

You can use this section to learn how ARNs work. The material here provides examples are geared specifically for Quick Sight.

Topics

ARN formats

ARNs are delimited by colons, and composed of segments, which are the parts separated by colons (`:`). The specific components and values used in the segments of an ARN depend on which AWS service the ARN is for. The following example shows how ARNs are constructed.

```
arn:partition:service:region:account-id:resource-id
        arn:partition:service:region:account-id:resource-type/resource-id
        arn:partition:service:region:account-id:resource-type:resource-id
```

These ARNs contain the following segments:

`partition` – The partition that the resource is in. For standard AWS Regions, the partition is *`aws`*. If you have resources in other partitions, the partition is *`aws-partitionname`*. For example, the partition for resources in the China (Beijing) Region is `aws-cn`.

`service` – The service namespace that identifies the AWS product. For example, `quicksight` identifies Quick Sight, `s3` identifies Amazon S3, `iam` identifies IAM, and so on.

region – The AWS Region that the resource resides in. The ARNs for some resources don't require an AWS Region, so this component might be omitted in some cases, like in the case of S3. Quick Sight ARNs require an AWS Region.

account-id – The ID of the AWS account that owns the resource. When you use the account number in an ARN or an API operation, you omit the hyphens (for example, 123456789012). The ARNs for some resources don't require an account number, so this component might be omitted. Quick Sight ARNs require an AWS account number. However, the account number and the AWS Region are omitted from S3 bucket ARNs, as shown following.

```
arn:aws:s3:::bucket_name
arn:aws:s3:::bucket_name/key_name
```

resource or *resource-type* – The content of this part of the ARN varies by service. A resource identifier can be the name or ID of the resource (for example, `user/Bob` or `instance/i-1234567890abcdef0`) or a resource path. For example, some resource identifiers include a parent resource (*sub-resource-type/parent-resource/sub-resource*) or a qualifier such as a version (*resource-type:resource-name:qualifier*).

Some resource ARNs can include a path, variable, or wildcard.

You can use wildcard characters (* and ?) within any ARN segment. An asterisk (*) represents any combination of zero or more characters, and a question mark (?) represents any single character. You can use multiple * or ? characters in each segment. If you're using the ARN for permissions, avoid using * wildcards if possible, to limit access to only the required elements. Following are some examples of using paths, wildcards, and variables.

For the following example, we use an S3 ARN. You might use this when you give permissions to S3 in an IAM policy. This S3 ARN shows a path and file are specified.

> **Note**
>
> The term key name is used to describe what looks like a path and file after `bucketname/`. These are called key names because a bucket doesn't actually contain folder structures like those used in your computer's file system. Instead the slash (/) is a delimiter that helps to make the organization of the bucket more intuitive. In this case, the bucket name is `amzn-s3-demo-bucket`, and the key name is `developers/design_info.doc`.

```
arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export-2019-q4.json
```

To identify all the objects in the bucket, you can use a wildcard to indicate that all key names (or paths and files) are included in the ARN, as follows.

```
arn:aws:s3:::amzn-s3-demo-bucket/*
```

You can use part of a key name plus the wildcard to identify all the objects that begin with a specific pattern. In this case, it resembles a folder name plus a wildcard, as shown following. However, this ARN also includes any "subfolders" inside of `my-data`.

```
arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export*
```

In this case, specifying using this wildcard includes the objects with names like the following:

- `my-data/sales-export-1.xlsx`
- `my-data/sales-export-new.txt`
- `my-data/sales-export-2019/file1.txt`

You can use wildcards of both types (asterisks and question marks) in combination or separately, as shown following.

```
arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export-2019-q?.*

arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export-20??-q?.*
```

Or, if you want to future-proof the ARN, you can replace the entire year with a wildcard, rather than just using wildcards for the last two digits.

```
arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export-????-q?.*
arn:aws:s3:::amzn-s3-demo-bucket/my-data/sales-export-*-q?.*
```

To read more about S3 ARNs, see Specifying Resources in a Policy and Object Key and Metadata in the *Amazon S3 User Guide*.

Quick Sight resource ARNs

The following resource types are defined by Quick Sight: user, group, and dashboard. These are used in Quick Sight API calls and as elements of IAM permission statements. To find up-to-date information for Quick Sight (service prefix: quicksight) resources, actions, and condition context keys for use in IAM permission policies, see Actions, Resources, and Condition Keys for Quick Sight in the *IAM User Guide*.

Resource type	ARN format	Condition keys
user	arn:${Partition}:quicksight:${Region}:${Account}:user/${ResourceId}	N/A
group	arn:${Partition}:quicksight:${Region}:${Account}:group/${ResourceId}	N/A
dashboard	arn:${Partition}:quicksight:${Region}:${Account}:dashboard/${ResourceId}	N/A

Resource ARNs are constructed from the segments that describe your resource. For example, a resource ARN for an analysis consists of the following segments.

```
arn:<partition>:quicksight:<aws-region>:<aws-account-id>:<resource-type>/<resource-id>
```

The segments are defined as follows:

- *partition* – for example, `aws` or `aws-cn`.
- *aws-region* – The AWS Region that contains the resource.
- *aws-account-id* – The AWS account that contains the resource. This ID excludes the hyphens.
- *resource-type* – The type of resource. For this example, this is `analyses`. For a dashboard, it's `dashboard`.
- *resource-id* – The unique identifier for a specific resource.

The AWS Region, resource type, and resource ID are identified in the URL of the resource when you are using the Quick Sight console. For example, let's say this is the URL of the analysis that you want an ARN for.

```
https://us-east-2.quicksight.aws.amazon.com/sn/analysis/4036e682-7de6-4c05-8a76-be51b9ec9b29
```

The AWS Region is `us-east-2`. The resource-type is `analysis`. The resource ID in this URL is `4036e682-7de6-4c05-8a76-be51b9ec9b29`. If your account number is `111122223333`, then the ARN for this analysis is as follows.

```
arn:aws:quicksight:us-east-2:111122223333:analysis/4036e682-7de6-4c05-8a76-be51b9ec9b29
```

To get your AWS account number, contact your system administrator.

Permissions for Quick Sight resources

If you're not sure what the necessary permission is, you can attempt to make a call. The client then tells you what the missing permission is. You can use asterisk (*) in the Resource field of your permission policy instead of specifying explicit resources. However, we highly recommend that you restrict each permission as much as possible. You can restrict user access by specifying or excluding resources in the policy, using their Quick Sight ARN. To retrieve the ARN of an Quick Sight resource, use the `Describe` operation on the relevant resource.

Before you can call the Quick Sight API operations, you need the `quicksight:`*`operation-name`* permission in a policy attached to your IAM identity. For example, to call `list-users`, you need the permission `quicksight:ListUsers`. The same pattern applies to all operations. If you attempt to make the call you don't have permissions to call, the resulting error shows you what the missing permission is. We highly recommend that you restrict each permission as much as possible.

You can add conditions in IAM to further restrict access to an API in some scenarios. For example, when you add `User1` to `Group1`, the main resource is `Group1`. You can allow or deny access to certain groups. Or you can also edit the Quick Sight IAM key `quicksight:UserName` to add a condition to allow or prevent certain users from being added to that group.

For more information, see the following:

- Actions, Resources, and Condition Keys
- IAM JSON Policy Elements

Best practices

By working with Quick Sight, you can share analyses, dashboards, templates, and themes with up to 100 principals. A *principal* can be one of the following:

- The Amazon Resource Name (ARN) of an Quick Sight user or group associated with a data source or dataset. (This is common.)
- The ARN of an Quick Sight user, group, or namespace associated with an analysis, dashboard, template, or theme. (This is common.)
- The ARN of an AWS account root: This is an IAM ARN rather than a Quick Sight ARN. Use this option only to share resources (templates) across AWS accounts. (This is less common.)

To share these resources with more principals, consider assigning resource permissions at the group or namespace level. For example, if you add users into a group and share a resource to the group, the group counts as one principal. This is true even though it's shared to everyone in the group.

Quick Sight API errors

Quick Sight has two types of error codes:

- **Client errors** – These errors are usually caused by something the client did. An example is specifying an incorrect or invalid parameter in the request. Another is using an action or resource for a user that doesn't have permission to use the action or resource. These errors are accompanied by a 400-series HTTP response code.
- **Server errors** – These errors are usually caused by an AWS server-side issue. These errors are accompanied by a 500-series HTTP response code.

Topics

Common client errors

Following, you can find a list of the common client errors that all operations can return.

Error code	Description
`AuthFailure`	The provided credentials couldn't be validated . You might not be authorized to carry out the request. Ensure that your account is authorized to use the Quick Sight service and that your credit card details are correct. Also ensure that you're using the correct access keys.
`Blocked`	Your account is currently blocked. Contact Support if you have questions.
`DryRunOperation`	The user has the required permissions, so the request would have succeeded, but the `DryRun` parameter was used.
`IdempotentParameterMismatch`	The request uses the same client token as a previous, but nonidentical request. Don't reuse a client token with different requests, unless the requests are identical.
`IncompleteSignature`	The request signature doesn't conform to AWS standards.
`InvalidAction`	The action or operation requested isn't valid. Verify that the action is typed correctly.
`InvalidCharacter`	A specified character isn't valid.
`InvalidClientTokenId`	The X.509 certificate or AWS access key ID provided doesn't exist in our records.
`InvalidPaginationToken`	The specified pagination token isn't valid or is expired.
`InvalidParameter`	A parameter specified in a request isn't valid, is unsupported, or can't be used. The returned message provides an explanation of the error value.

Error code	Description
`InvalidParameterCombination`	A value that indicates an incorrect combination of parameters, or a missing parameter.
`InvalidParameterValue`	A value specified in a parameter isn't valid, is unsupported, or can't be used. Ensure that you specify a resource by using its full ID. The returned message provides an explanation of the error value.
`InvalidQueryParameter`	The AWS query string is malformed or doesn't adhere to AWS standards.
`MalformedQueryString`	The query string contains a syntax error.
`MissingAction`	The request is missing an action or a required parameter.
`MissingAuthenticationToken`	The request must contain either a valid (registered) AWS access key ID or X.509 certificate.
`MissingParameter`	The request is missing a required parameter. Ensure that you have supplied all the required parameters for the request, for example the resource ID.
`OptInRequired`	You aren't authorized to use the requested service. Ensure that you have subscribed to the service you are trying to use. If you are new to AWS, your account might take some time to be activated while your credit card details are being verified.
`PendingVerification`	Your account is pending verification. Until the verification process is complete, you might not be able to carry out requests with this account. If you have questions, contact AWS Support.

Error code	Description
`RequestExpired`	The request reached the service more than 15 minutes after the date stamp on the request or the request expiration date (such as for presigned URLs). Or the date stamp on the request is more than 15 minutes in the future. If you're using temporary security credentials, this error can also occur if the credentials have expired. For more information, see Temporary Security Credentials in the *IAM User Guide*.
`UnauthorizedOperation`	You aren't authorized to perform this operation. Check your IAM policies, and ensure that you are using the correct access keys.
`UnknownParameter`	An unknown or unrecognized parameter was supplied. Requests that can cause this error include supplying a misspelled parameter or a parameter that isn't supported for the specified API version.
`UnsupportedInstanceAttribute`	The specified attribute can't be modified.
`UnsupportedOperation`	The specified request includes an unsupport ed operation. The returned message provides details of the unsupported operation.
`UnsupportedProtocol`	The protocol that you used is unsupported.
`ValidationError`	The input fails to satisfy the constraints specified by an AWS service.

Client errors

Following, you can find a list of client errors that are specific to Quick Sight API operations.

Error code	Description
AccessDeniedException	You don't have access to this. The provided credentials can't be validated. You might not be authorized to carry out the request. Ensure that your account is authorized to use the Amazon Quick Sight service and that your policies have the correct permissions. Also, ensure that you are using the correct access keys.
DomainNotWhiteListedException	The domain specified isn't on the allow list. Make sure that all domains for embedded dashboards are added to the approved list by an Amazon Quick Sight administrator.
IdentityTypeNotSupportedException	The identity type specified isn't supported . Supported identity types include: IAM and QUICKSIGHT.
InvalidNextTokenException	The `NextToken` value isn't valid.
InvalidParameterValueException	One or more parameters doesn't have a valid value.
PreconditionNotMetException	One or more preconditions aren't met.
QuickSightUserNotFoundException	The user isn't found. This can happen in any operation that requires finding a user based on the provided user name, such as `DeleteUser` , `DescribeUser` , and so on.
ResourceExistsException	The resource specified doesn't exist.
ResourceNotFoundException	One or more resources couldn't be found.
SessionLifetimeInMinutesInvalidException	The number of minutes specified for the lifetime of a session is invalid. The session lifetime must be 15–600 minutes.

Error code	Description
ThrottlingException	Access is throttled.
UnsupportedUserEditionException	Indicates that you are calling an operation on an Amazon Quick Sight subscription where the edition doesn't include support for that operation. Amazon Quick Sight currently has Standard Edition and Enterprise Edition. Not every operation and capability is available in every edition.

Common causes of client errors

There are a number of reasons that you might encounter an error while performing a request. Some errors can be prevented or easily solved by following these guidelines:

- **Specify the AWS account ID and namespace** – Make sure that the relevant AWS account ID are provided with each request. The namespace must be set to `default`.
- **Allow for eventual consistency** – Some errors are caused because a previous request hasn't yet propagated through the system.
- **Use a sleep interval between request rates** – Quick Sight API requests are throttled to help maintain the performance of the service. If your requests have been throttled, you get an error.
- **Use the full ID of the resource** – When specifying a resource, ensure that you use its full ID, and not its user-supplied name or description.
- **Check your services** – Ensure that you have signed up for all the services you are attempting to use. You can check which services you're signed up for by going to the **My Account** section of the AWS home page.
- **Check your permissions** – Ensure that you have the required permissions to carry out the request.
- **Check your VPC** – Some resources can't be shared between virtual private clouds (VPCs), for example security groups.
- **Check your credentials** – Ensure that you provide your access keys when you are making requests and that you have entered the credentials correctly. Also, if you have more than one

account, ensure that you are using the correct credentials for a particular account. If the provided credentials are incorrect, you might get the following error: `Client.AuthFailure`.

Server errors

Following, you can find a list of errors that can be returned by the server.

Error code	Description
BatchClientRequestTokensNotDistinctException	The batch client request tokens aren't unique.
EmptyBatchRequestException	The batch request was empty.
InternalFailureException	An internal failure occurred.
InternalServiceError	There was an internal error from the service.
InvalidBatchClientRequestTokenException	The AWS request token for this client batch request is invalid.
InvalidParameterException	One or more parameters has a value that isn't valid.
LimitExceededException	A limit is exceeded.
ResourceUnavailableException	This resource is currently unavailable.
TooManyEntriesInBatchRequestException	There are too many entries in this batch request.

Operations

To find Quick Sight API operations by category, use the following list.

Topics

Account customization operations

With account customization API operations, you can update and customize Amazon Quick Sight account settings. For more information, see the following API operations.

Topics

- CreateAccountCustomization
- DeleteAccountCustomization
- DescribeAccountCustomization
- UpdateAccountCustomization

Account settings

With account settings operations, you can perform actions on Quick Sight account settings. For more information, see the following API operations.

Topics

- DescribeAccountSettings
- UpdateAccountSettings

DescribeAccountSettings

Use the `DescribeAccountSettings` API operation to describe the settings that were used when your Amazon Quick Sight subscription was first created in this AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-account-settings
    --aws-account-id 555555555555
```

For more information about the `DescribeAccountSettings` API operation, see DescribeAccountSettings in the *Amazon Quick Sight API Reference*.

UpdateAccountSettings

Use the `UpdateAccountSettings` API operation to update the Amazon Quick Sight settings in your AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-account-settings
    --aws-account-id 555555555555
```

```
    --default-namespace NAMESPACE
    --notification-email EMAIL
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-account-settings
    --cli-input-json file://updateaccountsettings.json
```

For more information about the `UpdateAccountSettings` API operation, see UpdateAccountSettings in the *Amazon Quick Sight API Reference*.

CreateAccountCustomization

Use the `CreateAccountCustomization` API operation to create Amazon Quick Sight customizations in the current AWS Region. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-account-customization
    --aws-account-id 555555555555
    --account-customization DEFAULTTHEME
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-account-customization
    --cli-input-json file://createaccountcustomization.json
```

For more information about the `CreateAccountCustomization` API operation, see CreateAccountCustomization in the *Amazon Quick Sight API Reference*.

DeleteAccountCustomization

Use the `DeleteAccountCustomization` API operation to delete all Amazon Quick Sight customizations in this AWS Region for the specified AWS account and Quick Sight namespace. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-account-customization
    --aws-account-id 555555555555
```

For more information about the `DeleteAccountCustomization` API operation, see DeleteAccountCustomization in the *Amazon Quick Sight API Reference*.

DescribeAccountCustomization

Use the `DescribeAccountCustomization` API operation to describe the customizations associated with the provided AWS account and Amazon Quick Sight namespace in an AWS Region. Following is an example AWS CLI command for this operation.

AWS CLI

```
quicksight describe-account-customization
   --aws-account-id 555555555555
```

For more information about the `DescribeAccountCustomization` API operation, see DescribeAccountCustomization in the *Amazon Quick Sight API Reference*.

UpdateAccountCustomization

Use the `UpdateAccountCustomization` API operation to update Amazon Quick Sight customizations in the current AWS Region. Currently, the only customization that you can use is a theme.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-account-customization
    --aws-account-id 555555555555
    --namespace NAMESPACE
    --account-customization DEFAULTTHEME
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-account-customization
    --cli-input-json file://updateaccountcustomization.json
```

For more information about the `UpdateAccountCustomization` API operation, see UpdateAccountCustomization in the *Amazon Quick Sight API Reference*.

Analysis operations

With analysis API operations, you can perform actions on Amazon Quick Sight analyses. For more information, see the following API operations.

Topics

- Analysis permissions operations
- CreateAnalysis
- DeleteAnalysis
- DescribeAnalysis
- ListAnalyses
- RestoreAnalysis
- SearchAnalyses
- UpdateAnalysis

Analysis permissions operations

With analysis permissions API operations, you can view and update permissions for analyses. For more information, see the following API operations.

Topics

- DescribeAnalysisPermissions
- UpdateAnalysisPermissions

DescribeAnalysisPermissions

Use the `DescribeAnalysisPermissions` API operation to view the read and write permissions for an analysis. To use this operation, you need the ID of the analysis whose permissions you

want to view. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-analysis-permissions
    --aws-account-id 555555555555
    --analysis-id ANALYSISID
```

For more information about the `DescribeAnalysisPermissions` API operation, see DescribeAnalysisPermissions in the *Amazon Quick Sight API Reference*.

UpdateAnalysisPermissions

Use the `UpdateAnalysisPermissions` API operation to update the read and write permissions for an analysis. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the analysis whose permissions you want to update. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-analysis-permissions
    --aws-account-id 555555555555
    --analysis-id ANALYSISID
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:RestoreAnalysis,quicksight:UpdateAnalysisPermissions,quicksight:
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:555555555555:user/
default/
USERNAME,Actions=quicksight:RestoreAnalysis,quicksight:UpdateAnalysisPermissions,quicksight:
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-analysis-permissions
    --cli-input-json file://updateanalysispermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateAnalysisPermissions` API operation, see UpdateAnalysisPermissions in the *Amazon Quick Sight API Reference*.

CreateAnalysis

Use the `CreateAnalysis` API operation to create an analysis in Amazon Quick Sight for a specified user. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-analysis
    --aws-account-id AWSACCOUNTID
    --analysis-id ANALYSISID
    --name NAME
    --source-entity SOURCEENTITY
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-analysis
    --cli-input-json file://createanalysis.json
```

For more information about the `CreateAnalysis` API operation, see CreateAnalysis in the *Amazon Quick Sight API Reference*.

DeleteAnalysis

Use the `DeleteAnalysis` API operation to delete an analysis from Amazon Quick Sight for a specified user. To use this operation, you need the ID of the analysis that you want to delete. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-analysis
    --aws-account-id 555555555555
    --analysis-id ANALYSISID
```

For more information about the `DeleteAnalysis` API operation, see DeleteAnalysis in the *Amazon Quick Sight API Reference*.

DescribeAnalysis

Use the `DescribeAnalysis` API operation to view a summary of the metadata for an analysis for a specified user. To use this operation, you need the ID of the analysis that you want to describe. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-analysis
    --analysis-id ANALYSISID
    --aws-account-id 555555555555
```

For more information about the `DescribeAnalysis` API operation, see DescribeAnalysis in the *Amazon Quick Sight API Reference*.

ListAnalyses

Use the `ListAnalyses` API operation to list Amazon Quick Sight analyses that exist in the specified AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-analyses
    --aws-account-id 555555555555
    --page-size 10
```

```
    --max-items 10
```

For more information about the `ListAnalyses` API operation, see ListAnalyses in the *Amazon Quick Sight API Reference*.

RestoreAnalysis

Use the `RestoreAnalysis` API operation to restore an analysis for a specified user. To use this operation, you need the ID of the analysis that you want to restore. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight restore-analysis
    --analysis-id ANALYSISID
    --aws-account-id 555555555555
```

For more information about the `RestoreAnalysis` API operation, see RestoreAnalysis in the *Amazon Quick Sight API Reference*.

SearchAnalyses

Use the `SearchAnalyses` API operation to search for analyses that belong to the specified user. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight search-analyses
    --aws-account-id 555555555555
    --filters
 Operator=StringEquals,Name=QUICKSIGHT_USER,Value=arn:aws:quicksight:us-
east-1:555555555555:user/default/USERNAME
    --page-size 10
    --max-items 100
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `SearchAnalyses` API operation, see SearchAnalyses in the *Amazon Quick Sight API Reference*.

UpdateAnalysis

Use the `UpdateAnalysis` API operation to update an analysis in Amazon Quick Sight. To use this operation, you need the ID of the analysis that you want to update. The analysis ID is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-analysis
    --aws-account-id 555555555555
    --analysis-id ANALYSISID
    --name NAME
    --source-entity '{"SourceTemplate":{"DataSetReferences":
[{"DataSetPlaceholder":"PLACEHOLDER","DataSetArn":"arn:aws:quicksight:us-
west-2:555555555555:dataset/DATASETID"}],"Arn":"arn:aws:quicksight:us-
west-2:555555555555:template/TEMPLATEID"}}'
    --theme-arn THEMEARN
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-analysis
    --cli-input-json file://updateanalysis.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateAnalysis` API operation, see UpdateAnalysis in the *Amazon Quick Sight API Reference*.

Asset bundle operations

Use Quick Sight asset bundle API operations to export Quick Sight assets from one account to another. Asset bundle operations can be used to back up or restore deleted work, promote new work into a production account, or to duplicate assets within an account or across multiple accounts.

Asset bundle import and export operations support the following asset types:

- Analyses
- Dashboards
- Data sources
- Datasets
- Shared folders
- Restricted folders
- Refresh schedules
- Themes
- VPC connections

The following data sources and dataset types aren't supported by the asset bundle APIs.

- Adobe Analytics
- File
- GitHub
- JIRA
- Salesforce
- ServiceNow
- Twitter

Topics

Permissions

Before you begin, verify that you have an AWS Identity and Access Management role that grants the CLI user access to call the Quick Sight asset bundle API operations.

Quick Sight recommends that you use the `AWSQuickSightAssetBundleExportPolicy` and `AWSQuickSightAssetBundleImportPolicy` IAM managed policies to streamline your API usage. You can also choose to explicitly define your oen IAM policy to fit your use case. For more information about IAM managed policies in Quick Sight, see AWS managed policies for Quick Sight.

The following example shows an IAM policy that you can add to an existing IAM role to use the `StartAssetBundleExportJob` operation.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "quicksight:DescribeAssetBundleExportJob",
        "quicksight:ListAssetBundleExportJobs",
        "quicksight:StartAssetBundleExportJob",
        "quicksight:DescribeAnalysis",
        "quicksight:DescribeDashboard",
        "quicksight:DescribeDataSet",
        "quicksight:DescribeDataSetRefreshProperties",
        "quicksight:DescribeDataSource",
        "quicksight:DescribeRefreshSchedule",
        "quicksight:DescribeTheme",
        "quicksight:DescribeVPCConnection",
        "quicksight:ListRefreshSchedules",
        "quicksight:DescribeAnalysisPermissions",
        "quicksight:DescribeDashboardPermissions",
        "quicksight:DescribeDataSetPermissions",
        "quicksight:DescribeDataSourcePermissions",
        "quicksight:DescribeThemePermissions",
        "quicksight:ListTagsForResource"
      ],
      "Resource": "*"
```

```
    }
  ]
}
```

The following example shows an IAM policy that you can add to an existing IAM role to use the `StartAssetBundleImportJob` operation.

JSON

```
{
  "Version":"2012-10-17",
  "Statement": [
      {
        "Effect": "Allow",
        "Action": [
            "quicksight:DescribeAssetBundleImportJob",
            "quicksight:ListAssetBundleImportJobs",
            "quicksight:StartAssetBundleImportJob",
            "quicksight:CreateAnalysis",
            "quicksight:DeleteAnalysis",
            "quicksight:DescribeAnalysis",
            "quicksight:UpdateAnalysis",
            "quicksight:CreateDashboard",
            "quicksight:DeleteDashboard",
            "quicksight:DescribeDashboard",
            "quicksight:UpdateDashboard",
            "quicksight:UpdateDashboardLinks",
            "quicksight:UpdateDashboardPublishedVersion",
            "quicksight:CreateDataSet",
            "quicksight:DeleteDataSet",
            "quicksight:DescribeDataSet",
            "quicksight:PassDataSet",
            "quicksight:UpdateDataSet",
            "quicksight:DeleteDataSetRefreshProperties",
            "quicksight:DescribeDataSetRefreshProperties",
            "quicksight:PutDataSetRefreshProperties",
            "quicksight:CreateRefreshSchedule",
            "quicksight:DescribeRefreshSchedule",
            "quicksight:DeleteRefreshSchedule",
            "quicksight:ListRefreshSchedules",
            "quicksight:UpdateRefreshSchedule",
```

```
            "quicksight:CreateDataSource",
            "quicksight:DescribeDataSource",
            "quicksight:DeleteDataSource",
            "quicksight:PassDataSource",
            "quicksight:UpdateDataSource",
            "quicksight:CreateTheme",
            "quicksight:DeleteTheme",
            "quicksight:DescribeTheme",
            "quicksight:UpdateTheme",
            "quicksight:CreateVPCConnection",
            "quicksight:DescribeVPCConnection",
            "quicksight:DeleteVPCConnection",
            "quicksight:UpdateVPCConnection",
            "quicksight:DescribeAnalysisPermissions",
            "quicksight:DescribeDashboardPermissions",
            "quicksight:DescribeDataSetPermissions",
            "quicksight:DescribeDataSourcePermissions",
            "quicksight:DescribeThemePermissions",
            "quicksight:UpdateAnalysisPermissions",
            "quicksight:UpdateDashboardPermissions",
            "quicksight:UpdateDataSetPermissions",
            "quicksight:UpdateDataSourcePermissions",
            "quicksight:UpdateThemePermissions",
            "quicksight:ListTagsForResource",
            "quicksight:TagResource",
            "quicksight:UntagResource",
            "s3:GetObject",
            "iam:PassRole"
        ],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "iam:PassedToService": "quicksight.amazonaws.com"
            }
        }
      }
  ]
}
```

Asset bundle export operations

Quick Sight developers can use asset bundle export operations to export existing Quick Sight assets and download their definitions to be saved in your own storage. Quick Sight doesn't export data contained within the asset. These exported assets can be imported back into Quick Sight whenever you want. To export Quick Sight assets, start a named asynchronous export job. Then, poll for the job's completion, and then download the asset bundle with the download URL that's provided by the Quick Sight API.

Assets that are exported with the Quick Sight asset bundle APIs can be exported as a Quick Sight JSON bundle or a CloudFormation JSON file.

When you run an export job that generates a Quick Sight JSON file, the job returns a `.qs` zip file. The file can be unzipped to access the exported asset definitions.

Use the following sections to learn more about the asset bundle export API operations.

StartAssetBundleExportJob

Export jobs are configured with the `StartAssetBundleExportJobRequest` object.

Export jobs are identified by an `AssetBundleExportJobId` that you provide when you create the new export job. This ID is unique while the job is running. After the job is completed, you can reuse this ID for another job.

Export jobs include a list of Quick Sight asset ARNs to be exported. You can choose to have all dependencies of the specified asset ARNs to be exported automatically with the rest of the job. For example, if you're creating a job to export a Quick Sight dashboard, you can also choose to export the dashboard's theme, dataset, and data source.

Developers can also choose to have all assets in a folder and its subfolders exported automatically to preserve folder hierarchy and folder memberships. Parent folders are considered to be dependencies of subfolders and are included in the export if the `IncludeAllDependencies` parameter is set to `True`. Assets in a folder are considered folder members and are included in the export if the `IncludeFolderMembers` parameter is set to `ONE_LEVEL/RECURSE`. When assets are exported directly, folder membership information can be preserved when the `IncludeFolderMemberships` parameter is set to `True`. To include the folder in the direct export, set `IncludeAllDependencies` to `True`.

All export jobs run asynchronously after they are started. Poll the status of an export job with a `DescribeAssetBundleExportJob` call to know is the current status of the job. Callers must

have read-only permissions for all of the resource types that are exported, including the optional dependencies that are included in the export job.

In some cases, certain Quick Sight assets contain anomalies that don't impact the end user's experience. By default, the `StartAssetBundleExportJob` API operates in `Lenient` mode, which ignores these anomalies. Callers can choose to enforce stricter validations with `Strict` mode during export. To do so, set the value of the optional `StrictModeForAllResources` parameter to `"True”`. The `StartAssetBundleImportJob` API operation follows the validation strategy that is defined in the exported bundle. To import an existing bundle with `Lenient` mode, run a new export job with the optional `StrictModeForAllResources` parameter set to `"False"`.

For more information about the `StartAssetBundleExportJob` operation, see StartAssetBundleExportJob in the *Quick Sight API Reference*.

DescribeAssetBundleExportJob

Use the `DescribeAssetBundleExportJob` operation to obtain the current status of an existing export job that's up to 14 days old. You can also use this operation to review a specified job's configuration.

Export jobs that have succeeded return a download URL for the asset bundle file in their description. Failed export jobs return error information in their description. Poll this operation until the export job that you want the status of has succeeded or failed.

For more information about the `DescribeAssetBundleExportJob` operation, see DescribeAssetBundleExportJob in the *Quick Sight API Reference*.

ListAssetBundleExportJobs

Use the `ListAssetBundleExportJobs` operation to retrieve a list of all export jobs that were created in the last 14 days. Export jobs are listed in the order that they were started, starting with the most recently started job. To have multiple lists by this operation, you can choose to specify a maximum page size to be returned and use a pagination token.

For more information about the `ListAssetBundleImportJobs` operation, see ListAssetBundleExportJobs in the *Quick Sight API Reference*.

Examples

The following example uses a `StartAssetBundleExportJob` API call to create a CloudFormation JSON file with override parameters.

```
# configure and start the export job
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--cloud-formation-override-property-configuration '{"Dashboards": [{"Arn":
 "arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/RESOURCEID","Properties":
 ["Name"]}]}' \
--include-all-dependencies \
--export-format CLOUDFORMATION_JSON

# poll job description until status is success
aws quicksight describe-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID

# download the provided bundle (wget used here - any tool or browser works as well)
wget -O ~/qs-bundle.qs 'https://the-long-url-from-your-job-description...'
```

The following example uses a `StartAssetBundleExportJob` API call to create a Quick Sight asset bundle file.

```
# configure and start the export job
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--include-all-dependencies \
--export-format QUICKSIGHT_JSON

# poll job description until status is success
aws quicksight describe-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID

# download the provided bundle (wget used here - any tool or browser works as well)
wget -O ~/qs-bundle.qs 'https://the-long-url-from-your-job-description...'
```

The following example uses a `StartAssetBundleExportJob` API call to include information for tags. By default, tag information is not exported.

```
# Export in QuickSight format
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--include-all-dependencies \
--include-tags \
--export-format QUICKSIGHT_JSON

# Export in CloudFormation format, with optional tags overrides
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--include-all-dependencies \
--include-tags \
--export-format CLOUDFORMATION_JSON
```

The following example uses a `StartAssetBundleExportJob` API call to include permissions information. By default, permission information is not exported. Permission overrides are not supported for the CloudFormation format. To import permissions for a CloudFormation format file, make sure that the source and target accounts are the same or have the same principal names for users, groups, and namespaces.

```
# Export in QuickSight format
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--include-all-dependencies \
--include-permissions \
--export-format QUICKSIGHT_JSON
```

```
# Export in CloudFormation format
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
--include-all-dependencies \
--include-permissions \
--export-format CLOUDFORMATION_JSON
```

The following example recusrively exports a folder along with all subfolders and the parent folder tree.

```
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns '["arn:aws:quicksight:REGION:AWSACCOUNTID:folder/RESOURCEID"]' \
--include-all-dependencies \
--include-folder-members RECURSE \
--export-format QUICKSIGHT_JSON"
```

The following example runs a dashboard export job that preserves the dashboard's folder memberships.

```
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/RESOURCEID"]' \
--include-folder-memberships \
--export-format QUICKSIGHT_JSON
```

The following example calls the `StartAssetBundleExportJob` API with `Strict` mode.

```
aws quicksight start-asset-bundle-export-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-export-job-id JOBID \
--resource-arns
 '["arn:aws:quicksight:REGION:AWSACCOUNTID:dashboard/
RESOURCEID","arn:aws:quicksight:REGION:AWSACCOUNTID:analysis/RESOURCEID"]' \
```

```
--include-all-dependencies \
--export-format QUICKSIGHT_JSON
```

Asset bundle import operations

Use asset bundle import operations to import Quick Sight assets from an Quick Sight bundle file that's generated by an earlier export job. The following statements apply to asset bundle import operations.

- You can only import Quick Sight JSON bundles with the Quick Sight asset bundle APIs. CloudFormation JSON files can only be imported using the CloudFormation console or APIs. Both Quick Sight JSON and CloudFormation JSON files support property value overrides. If you want to generate a Quick Sight JSON file, property overrides are specified when you use an import API call. If you want to generate a CloudFormation JSON file, property overrides are configured with the `cloud-formation-override-property-configuration` parameter when you create or update the CloudFormation stack. You can import files that were created from your account, or you can import asset bundle files that were generated from other Quick Sight accounts. When you create a new import job, you can choose to provide overrides when you configure the import job.
- The asset bundle import operations only support `.qs` format zip files. The `.qs` format file that contains the asset bundle that you want to import is in an Amazon S3 bucket or in a BASE64 encoded file that you can add to the import job directly. The S3 bucket exists in the same AWS account as your QuickSight account.
- All import jobs run asynchronously after they are started. Poll the status of an import job with a `DescribeAssetBundleImportJob` API call to know the current status of the job. If an asset bundle import job fails, you can choose to have all assets that were successfully imported during the failed job rollback. Information about the error that caused the job to fail is returned in the job description of a `DescribeAssetBundleImportJob` API call.
- All of an imported assets' dependencies must be present for an asset import job to succeed. You can include all dependencies of the asset when you export it. Alternatively, you can configure all dependencies in the QuickSight account that you want to move the asset into. For example, to import a dashboard, the dataset, data source, and theme that the dashboard uses must exist in the account that you're importing the asset into. The caller must have permissions to describe, create, and update all Quick Sight resources located in the asset that you want to import.
- After an import job succeeds, grant permissions to all users or user groups that need to access the newly created resource. If you want to override the properties of the `QUICKSIGHT JSON`

format export, provide the new values when you start an import job. If you want to override properties in a `CLOUDFORMATION_JSON` format export, provide the property names to override when you start an export job. Then and add the new values when the stack is created in the CloudFormation console or with the CloudFormation APIs.

Permissions are not propagated through the asset bundles. You can update asset permissions with an `UpdateDashboardPermissions` API call.

Use the following sections to learn more about the asset bundle import API operations.

StartAssetBundleImportJob

Import jobs are configured with the `StartAssetBundleImportJobRequest` object.

Import jobs are identified by an `AssetBundleImportJobId` that you provide when you create the new import job. This ID is unique while the job is running. After the job is completed, you can reuse this ID for another job.

Provide an Amazon S3 uri or a base64-encoded ZIP file to the request. If you use an Amazon S3 uri, the caller must have `GetObject` permissions. All assets contained in the file are imported into the target account.

You can choose to configure override values to be applied to specific assets when they are imported. All imported data sources must have credential overrides. You can store asset credentials in AWS Secrets Manager or you can set a username and password directly into an override. If you use Secrets Manager, provide the secret ARN in the data source override. The caller must have `GetSecretValue` and `DescribeSecret` permissions to configure the Secrets Manager secret to the override.

All import jobs run asynchronously after they are started. Poll the status of an export job with a `DescribeAssetBundleImportJob` call to know the current status of the job. Callers must have read and write permissions for all of the resource types that are exported, including the optional dependencies that are included in the export job.

When an asset import job fails, you can choose to have all assets that were successfully imported during the failed job roll back automatically. If you don't choose to roll back the assets, successfully imported assets will still exist in the account that they are imported to. Information about the error that caused the job to fail is returned in the job description of a `DescribeAssetBundleImportJob` API call.

For more information about the `StartAssetBundleImportJob` operation, see StartAssetBundleImportJob in the *Quick Sight API Reference*.

VPC overrides

When you make a `StartAssetBundleImportJob` API call, provide an override parameter for the VPC connection that's configured to your Quick Sight account. You can find the `OverrideParameters` value in the asset bundle file that was created when the asset was exported. The following example shows an `OverrideParameters` structure that uses the `PrefixForAllResources` value.

```
"OverrideParameters": {
  "VPCConnections": [
    {
        "VPCConnectionId": "<PrefixForAllResources<VPCConnectionId in asset bundle
 file"
        "DnsResolvers": [ "string" ],
        "Name": "string",
        "RoleArn": "string",
        "SecurityGroupIds": [ "string" ],
        "SubnetIds": [ "string" ]

    }
  ]
 }
```

For more information about setting up a VPC connection in Quick Sight, see Configuring the VPC connection with the Quick Sight CLI.

Permissions

The following statements apply to asset bundle import permissions.

- Asset bundle import operations support up to 64 principals.
- The final state of an asset bundle's permissions are determined by the following.
  - If the `OverridePermissions` parameter is provided in the input, all existing permissions are replaced by the permissions that are specified in the `OverridePermissions` parameter.
  - If the asset bundle was exported with permissions, all existing permissions are replaced by the permissions that are in the exported asset bundle's file.
  - If neither of the above conditions are met, no changes are made to the asset's permissions.

- If the caller executes an asset bundle import job from a different account than the account that the asset bundle was exported from, there are differences in the the user, group, and namespace principal ARNs. When this happens, provide the correct ARN values in the `OverridePermissions` parameter.

Tags

The final state of an asset's tags are determined by the following.

- If the `OverrideTags` parameter is provided in the API input, all existing tags are replaced by the tags that are specified in the `OverrideTags` parameter.
- If the asset bundle file is exported with tags, all existing tags are replaced by the tags that are in the asset bundle's file.
- If neither of the above statements aren't met, no changes are made to the asset's tags.

DescribeAssetBundleImportJob

Use the `DescribeAssetBundleImportJob` operation to obtain the current status of an existing export job that's up to 14 days old. You can also use this operation to review a specified job's configuration.

Failed import jobs return error information in their description. Poll this operation until the import job that you want the status of has succeeded or failed.

For more information about the `DescribeAssetBundleImportJob` operation, see DescribeAssetBundleImportJob in the *Quick Sight API Reference*.

ListAssetBundleImportJobs

Use the `ListAssetBundleImportJobs` operation to retrieve a list of all import jobs that were created in the last 14 days. Import jobs are listed in the order that they were started, starting with the most recently started job. If you expect to have multiple lists by this operation, you can choose to specify a maximum page size to be returned and use a pagination token.

For more information about the `ListAssetBundleImportJobs` operation, see ListAssetBundleImportJobs in the *Quick Sight API Reference*.

Examples

The following example creates an asset bundle import job for a file that is located in the caller's Amazon S3 bucket.

```
# upload your bundle to an S3 bucket in your account
aws s3 cp ~/qs-bundle.qs s3://bucket/key/qs-bundle.qs

aws quicksight start-asset-bundle-import-job
  --aws-account-id AWSACCOUNTID \
  --asset-bundle-import-job-id JOBID \
  --asset-bundle-import-source '{"S3Uri": "s3://bucket/key/qs-bundle.qs"}' \
  --failure-action ROLLBACK

# poll job description until status is success (or failed)
aws quicksight describe-asset-bundle-import-job
  --aws-account-id AWSACCOUNTID \
  --asset-bundle-import-job-id JOBID

# grant yourself or others permissions to view/modify the imported resources (for more
 information, see UpdateDashboardPermissions in the Amazon Quick Sight API Reference)

# open your Quick Sight site in your browser and confirm the imported resources
 (important)
```

The following example creates an asset bundle import job with a bundle file that's uploaded directly. This example also uses data source credential overrides.

```
aws quicksight start-asset-bundle-import-job
  --aws-account-id AWSACCOUNTID \
  --asset-bundle-import-job-id JOBID \
  --asset-bundle-import-source-bytes fileb://~/qs-bundle.qs \
  --asset-bundle-import-source-bytes fileb://~/qs-bundle.qs \
  --override-parameters '{"DataSources": [{"DataSourceId": "some-data-source-id",
 "Credentials": {"CredentialPair": {"Username": "some-username", "Password": "some-
password"}}}]}' \
  --failure-action ROLLBACK

# poll job description until status is success (or failed)
aws quicksight describe-asset-bundle-import-job
  --aws-account-id AWSACCOUNTID \
  --asset-bundle-import-job-id JOBID
```

```
# grant yourself or others permissions to view/modify the imported resources (for more
 information, see UpdateDashboardPermissions in the Amazon Quick Sight API Reference)

# open your Quick Sight site in your browser and confirm the imported resources
 (important)
```

The `Override` parameters also accept local files, as shown in the example below.

```
--override-parameters file://import-override-parameter-prod.json \
--override-permissions file://import-override-permission-prod.json \
--override-tags file://import-override-tags-prod.json \
```

If callers want to assign different permissions to exported assets, they can provide an override object at import. There are two ways that this can be done.

- Explicitly specify the resource IDs. If a prefix ID is specified, include the prefix in the resource ID.
- Use the wildcard `"*"` to represent all resources of a specific type in the asset bundle files.

In the example below, all dashboards that are included in the asset bundle file are imported with specified permissions.

```
// import-override-permission-prod.json
{
    "DataSources": [
        {
            "DataSourceIds": ["DATASOURCEID"],
            "Permissions": {
                "Principals": ["arn:aws:quicksight:REGION:AWSACCOUNTID:user/
default/USERIR"],
                "Actions": [
                    "quicksight:UpdateDataSourcePermissions",
                    "quicksight:DescribeDataSourcePermissions",
                    "quicksight:PassDataSource",
                    "quicksight:DescribeDataSource",
                    "quicksight:DeleteDataSource",
                    "quicksight:UpdateDataSource"
                ]
            }
        }
    ],
    "DataSets": [
```

```
        {
            "DataSetIds": ["DATASETID"],
            "Permissions": {
                "Principals": ["arn:aws:quicksight:REGION:AWSACCOUNTID:user/
default/USERIR"],
                "Actions": [
                    "quicksight:DeleteDataSet",
                    "quicksight:UpdateDataSetPermissions",
                    "quicksight:PutDataSetRefreshProperties",
                    "quicksight:CreateRefreshSchedule",
                    "quicksight:CancelIngestion",
                    "quicksight:PassDataSet",
                    "quicksight:ListRefreshSchedules",
                    "quicksight:UpdateRefreshSchedule",
                    "quicksight:DeleteRefreshSchedule",
                    "quicksight:DescribeDataSetRefreshProperties",
                    "quicksight:DescribeDataSet",
                    "quicksight:CreateIngestion",
                    "quicksight:DescribeRefreshSchedule",
                    "quicksight:ListIngestions",
                    "quicksight:DescribeDataSetPermissions",
                    "quicksight:UpdateDataSet",
                    "quicksight:DeleteDataSetRefreshProperties",
                    "quicksight:DescribeIngestion"
                ]
            }
        }
    ],
    "Themes": [
        {
            "ThemeIds": ["THEMEID"],
            "Permissions": {
                "Principals": ["arn:aws:quicksight:REGION:AWSACCOUNTID:user/
default/USERIR"],
                "Actions": [
                    "quicksight:ListThemeVersions",
                    "quicksight:UpdateThemeAlias",
                    "quicksight:DescribeThemeAlias",
                    "quicksight:UpdateThemePermissions",
                    "quicksight:DeleteThemeAlias",
                    "quicksight:DeleteTheme",
                    "quicksight:ListThemeAliases",
                    "quicksight:DescribeTheme",
                    "quicksight:CreateThemeAlias",
```

```
                    "quicksight:UpdateTheme",
                    "quicksight:DescribeThemePermissions"
                ]
            }
        }
    ],
    "Analyses": [
        {
            "AnalysisIds": ["ANALYSISIDS"],
            "Permissions": {
                "Principals": ["arn:aws:quicksight:REGION:AWSACCOUNTID:user/
default/USERIR"],
                "Actions": [
                    "quicksight:RestoreAnalysis",
                    "quicksight:UpdateAnalysisPermissions",
                    "quicksight:DeleteAnalysis",
                    "quicksight:DescribeAnalysisPermissions",
                    "quicksight:QueryAnalysis",
                    "quicksight:DescribeAnalysis",
                    "quicksight:UpdateAnalysis"
                ]
            }
        }
    ],
    "Dashboards": [
        {
            "DashboardIds": ["*"],
            "Permissions": {
                "Principals": ["arn:aws:quicksight:REGION:AWSACCOUNTID:user/
default/USERIR"],
                "Actions": [
                    "quicksight:DescribeDashboard",
                    "quicksight:ListDashboardVersions",
                    "quicksight:UpdateDashboardPermissions",
                    "quicksight:QueryDashboard",
                    "quicksight:UpdateDashboard",
                    "quicksight:DeleteDashboard",
                    "quicksight:DescribeDashboardPermissions",
                    "quicksight:UpdateDashboardPublishedVersion"
                ]
            }
        }
    ]
```

```
}
```

If callers want to assign different tags to imported assets, they can provide an override object at import. There are two ways that this can be done.

- Explicitly specify the resource IDs. If a prefix ID is specified, include the prefix in the resource ID.
- Use the wildcard "*" to represent all resources of a specific type in the asset bundle files.

In the example below, all dashboards that are included in the asset bundle file are imported with specified tags.

```
// import-override-tags-prod.json

{
    "DataSources": [
        {
            "DataSourceIds": ["DATASOURCEID"],
            "Tags": [
                {
                    "Key": "tagkey_datasource",
                    "Value": "tagvalue_datasource"
                },
                {
                    "Key": "tagkey2_datasource",
                    "Value": "tagvalue2_datasource"
                }
            ]
        }
    ],
    "DataSets": [
        {
            "DataSetIds": ["*"],
            "Tags": [
                {
                    "Key": "tagkey_dataset",
                    "Value": "tagvalue_dataset"
                },
                {
                    "Key": "tagkey2_dataset",
                    "Value": "tagvalue2_dataset"
                }
            ]
```

```
        }
    ],
    "Themes": [
        {
            "ThemeIds": ["*"],
            "Tags": [
                {
                    "Key": "tagkey_theme",
                    "Value": "tagvalue_theme"
                },
                {
                    "Key": "tagkey2_theme",
                    "Value": "tagvalue2_theme"
                }
            ]
        }
    ],
    "Analyses": [
        {
            "AnalysisIds": ["*"],
            "Tags": [
                {
                    "Key": "tagkey_analysis",
                    "Value": "tagvalue_analysis"
                },
                {
                    "Key": "tagkey2_analysis",
                    "Value": "tagvalue2_analysis"
                }
            ]
        }
    ],
    "Dashboards": [
        {
            "DashboardIds": ["*"],
            "Tags": [
                {
                    "Key": "tagkey_dashboard",
                    "Value": "tagvalue_dashboard"
                },
                {
                    "Key": "tagkey2_dashboard",
                    "Value": "tagvalue2_dashboard"
                }
```

```
            ]
        }
    ]
}
```

If you want to import an asset bundle file with `Strict` mode, use the `OverrideValidationStrategy` parameter and set `StrictModeForAllResources` to `True`. The following example calls the `StartAssetBundleImportJob` API with `Strict` mode.

```
aws quicksight start-asset-bundle-import-job
--aws-account-id AWSACCOUNTID \
--asset-bundle-import-job-id JOBID \
--asset-bundle-import-source-bytes fileb://~/qs-bundle.qs \
--override-validation-strategy '{"StrictModeForAllResources":true}'
```

Dashboard operations

With dashboard API operations, you can perform actions on Amazon Quick Sight dashboards. For more information, see the following API operations.

Topics

- Dashboard permissions
- CreateDashboard
- DeleteDashboard
- DescribeDashboard
- ListDashboards
- ListDashboardVersions
- SearchDashboards
- UpdateDashboard
- UpdateDashboardPublishedVersion

Dashboard permissions

With dashboard permissions API operations, you can view and update permissions for dashboards. For more information, see the following API operations.

Topics

DescribeDashboardPermissions

Use the `DescribeDashboardPermissions` API operation to view the read and write permissions for a dashboard. To use this operation, you need the ID of the dashboard whose permissions you want to view. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-dashboard-permissions
    --aws-account-id 555555555555
    --dashboard-id 111122223333
```

For more information about the `DescribeDashboardPermissions` API operation, see DescribeDashboardPermissions in the *Amazon Quick Sight API Reference*.

UpdateDashboardPermissions

Use the `UpdateDashboardPermissions` API operation to update read and write permissions for a dashboard. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the dashboard whose permissions you want to update. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-dashboard-permissions
    --aws-account-id 555555555555
    --dashboard-id DASHBOARDID
```

```
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:555555555555:user/
default/
USERNAME,Actions=quicksight:DescribeDashboard,quicksight:QueryDashboard,quicksight:ListDashb
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:555555555555:user/
default/
USERNAME,Actions=quicksight:DescribeDashboard,quicksight:QueryDashboard,quicksight:ListDashb
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-dashboard-permisisons
    --cli-input-json file://updatedashboardpermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateDashboardPermissions` API operation, see UpdateDashboardPermissions in the *Amazon Quick Sight API Reference*.

CreateDashboard

Use the `CreateDashboard` API operation to create a dashboard. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-dashboard
    --aws-account-id 555555555555
    --dashboard-id newDash
    --name Dashboard1
    --source-entity '{"SourceTemplate":{"DataSetReferences":
[{"DataSetPlaceholder":"PLACEHOLDER","DataSetArn":"arn:aws:quicksight:REGION:555555555555:da
DATASETID"}],"Arn":"arn:aws:quicksight:REGION:555555555555:template/TEMPLATEID"}}'
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-dashbord
    --cli-input-json file://createdashboard.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `CreateDashboard` API operation, see CreateDashboard in the *Amazon Quick Sight API Reference*.

DeleteDashboard

Use the `DeleteDashboard` API operation to delete a dashboard. To use this operation, you need the ID of the dashboard that you want to delete. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

You can add a `VersionNumber` parameter to this operation to only delete the specified version of the dashboard.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-dashboard
    --aws-account-id 555555555555
    --dashboard-id DASHBOARDID
```

For more information about the `DeleteDashboard` API operation, see DeleteDashboard in the *Amazon Quick Sight API Reference*.

DescribeDashboard

Use the `DescribeDashboard` API operation to view the summary of a dashboard. To use this operation, you need the ID of the dashboard that you want to view. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-dashboard
    --aws-account-id 555555555555
    --dashboard-id DASHBOARDID
```

For more information about the `DescribeDashboard` API operation, see DescribeDashboard in the *Amazon Quick Sight API Reference*.

ListDashboards

Use the `ListDashboards` API operation to list dashboards in an AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-dashboards
    --aws-account-id 555555555555
    --page-size 10
    --max-items 100
```

For more information about the `ListDashboards` API operation, see ListDashboards in the *Amazon Quick Sight API Reference*.

ListDashboardVersions

Use the `ListDashboardVersions` API operation to list all the versions of a dashboard in an AWS account. To use this operation, you need the ID of the dashboard that you want to update. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-dashboard-versions
```

```
    --aws-account-id AWSACCOUNTID
    --dashboard-id DASHBOARD
    --page-size 10
    --max-items 100
```

For more information about the `ListDashboardVersions` API operation, see ListDashboardVersions in the *Amazon Quick Sight API Reference*.

SearchDashboards

Use the `SearchDashboards` API operation to search for dashboards in an AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight search-dashboards
    --aws-account-id 555555555555
    --filters
 Operator=StringEquals,Name=QUICKSIGHT_USER,Value=arn:aws:quicksight:us-
east-1:555555555555:user/default/USERNAME
    --page-size 10
    --max-items 100
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `SearchDashboards` API operation, see SearchDashboards in the *Amazon Quick Sight API Reference*.

UpdateDashboard

Use the `UpdateDashboard` API operation to update a dashboard in an AWS account. To use this operation, you need the ID of the dashboard that you want to update. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-dashboard
    --aws-account-id 555555555555
    --dashboard-id DASHBOARDID
    --name Dashboard
    --source-entity '{"SourceTemplate":{"DataSetReferences":[{"DataSetPlaceholder":
 "PLACEHOLDER","DataSetArn": "arn:aws:quicksight:<region>:<awsaccountid>:dataset/
<datasetid>"}],"Arn": "arn:aws:quicksight:<region>:<awsaccountid>:template/
<templateid>"}}'
    --version-description VERSION
    --dashboard-publish-options
 AdHocFilteringOption={AvailabilityStatus=ENABLED},ExportToCSVOption={AvailabilityStatus=ENA
    --theme-arn THEMEARN
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-dashboard
    --cli-input-json file://updatedashboard.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateDashboard` API operation, see UpdateDashboard in the *Amazon Quick Sight API Reference*.

UpdateDashboardPublishedVersion

Use the `UpdateDashboardPublishedVersion` API operation to update the published version of a dashboard. To use this operation, you need the ID of the published dashboard that you want to update. The dashboard ID is part of the dashboard URL in Quick Sight. You can also use the `ListDashboards` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-dashboard-published-version
    --aws-account-id 555555555555
    --dashboard-id DASHBOARDID
    --dashboard-version-number VERSION
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-dashboard-published-version
    --cli-input-json file://updatedashboardpublishedversion.json
```

For more information about the `UpdateDashboardPublishedVersion` API operation, see UpdateDashboardPublishedVersion in the *Amazon Quick Sight API Reference*.

Data source operations

With data source operations, you can perform actions on data sources. For more information, see the following API operations.

Topics

- Data source permissions
- CreateDataSource
- DeleteDataSource
- DescribeDataSource
- ListDataSources
- UpdateDataSource

Data source permissions

With data source permissions API operations, you can view and update permissions for a data source. For more information, see the following API operations.

Topics

- DescribeDataSourcePermissions
- UpdateDataSourcePermissions

DescribeDataSourcePermissions

Use the `DescribeDataSourcePermissions` API operation to describe the resource permissions for a data source. To use this operation, you need the ID of the data source whose permissions you want to view. The data source ID is part of the data source URL in Quick Sight. You can also use the `ListDataSources` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-data-source-permissions
    --aws-account-id AWSACCOUNTID
    --data-source-id DATASOURCEID
```

For more information about the `DescribeDataSourcePermissions` API operation, see DescribeDataSourcePermissions in the *Amazon Quick Sight API Reference*.

UpdateDataSourcePermissions

Use the `UpdateDataSourcePermissions` API operation to update the resource permissions for a data source. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the data source whose permissions you want to update. The data source ID is part of the data source URL in Quick Sight. You can also use the `ListDataSources` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-data-source-permissions
    --aws-account-id AWSACCOUNTID
    --data-source-id DATASOURCEID
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:DescribeDataSource,quicksight:DescribeDataSourcePermissions,quic
```

```
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:DescribeDataSource,quicksight:DescribeDataSourcePermissions,quic
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-data-source-permissions
    --cli-input-json file://updatedatasourcepermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateDataSourcePermissions` API operation, see UpdateDataSourcePermissions in the *Amazon Quick Sight API Reference*.

CreateDataSource

Use the `CreateDataSource` API operation to create a data source. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-data-source
    --aws-account-id AWSACCOUNTID
    --data-source-id DATASOURCEID
    --name NAME
    --type ATHENA
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-data-source
    --cli-input-json file://createdatasource.json
```

For more information about the `CreateDataSource` API operation, see CreateDataSource in the *Amazon Quick Sight API Reference*.

DeleteDataSource

Use the `DeleteDataSource` API operation to permanently delete a data source from Amazon Quick Sight. To use this operation, you need the ID of the data source that you want to delete. The data source ID is part of the data source URL in Quick Sight. You can also use the `ListDataSources` API operation to get the ID.

> **Note**
>
> Deleting a data source breaks all datasets that reference it.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-data-source
    --aws-account-id AWSACCOUNTID
    --data-source-id DATASOURCEID
```

For more information about the `DeleteDataSource` API operation, see DeleteDataSource in the *Amazon Quick Sight API Reference*.

DescribeDataSource

Use the `DescribeDataSource` API operation to describe a data source. To use this operation, you need the ID of the data source that you want to view. The data source ID is part of the data source URL in Quick Sight. You can also use the `ListDataSources` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-data-source
    --aws-account-id AWSACCOUNTID
```

```
    --data-source-id DATASOURCEID
```

For more information about the `DescribeDataSource` API operation, see DescribeDataSource in the *Amazon Quick Sight API Reference*.

ListDataSources

Use the `ListDataSources` API operation to list all data sources in the current AWS Region that belong to a particular AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-data-sources
    --aws-account-id AWSACCOUNTID
    --page-size 10
    --max-items 100
```

For more information about the `ListDataSources` API operation, see ListDataSources in the *Amazon Quick Sight API Reference*.

UpdateDataSource

Use the `UpdateDataSource` API operation to update a data source. To use this operation, you need the ID of the data source that you want to update. The data source ID is part of the data source URL in Quick Sight. You can also use the `ListDataSources` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-data-source
    --aws-account-id AWSACCOUNTID
    --data-source-id DATASOURCEID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-data-source
```

```
    --cli-input-json file://updatedatasource.json
```

For more information about the `UpdateDataSource` API operation, see UpdateDataSource in the *Amazon Quick Sight API Reference*.

Dataset operations

With dataset operations, you can perform actions on Amazon Quick Sight datasets. For more information, see the following API operations.

Topics

- Dataset permissions operations
- CreateDataSet
- DeleteDataSet
- DescribeDataSet
- ListDataSets
- UpdateDataSet

Dataset permissions operations

With dataset permissions API operations, you can view and update permissions on a dataset. For more information, see the following API operations.

Topics

- DescribeDataSetPermissions
- UpdateDataSetPermissions

DescribeDataSetPermissions

Use the `DescribeDataSetPermissions` API operation to describe the permissions on a dataset. To use this operation, you need the ID of the dataset whose permissions that you want to view. The dataset ID is part of the dataset URL in Quick Sight. You can also use the `ListDataSets` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-data-set-permissions
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
```

For more information about the `DescribeDataSetPermissions` API operation, see DescribeDataSetPermissions in the *Amazon Quick Sight API Reference*.

UpdateDataSetPermissions

Use the `UpdateDataSetPermissions` API operation to update the permissions on a dataset. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the dataset whose permissions that you want to update. The dataset ID is part of the dataset URL in Quick Sight. You can also use the `ListDataSets` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-data-set-permissions
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:DescribeDataSet,quicksight:DescribeDataSetPermissions,quicksight
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:DescribeDataSet,quicksight:DescribeDataSetPermissions,quicksight
```

If your `region` has already been configured with the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-data-set-permissions
    --cli-input-json file://updatedatasetpermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information about the `UpdateDataSetPermissions` API operation, see UpdateDataSetPermissions in the *Amazon Quick Sight API Reference*.

CreateDataSet

Use the `CreateDataSet` API operation to create a dataset. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-data-set
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
    --name NAME
    --physical-table-map '{"PhysicalTableMap":{"string":{"CustomSql":{"Columns":
[{"Name":"string","Type":"string"}],"DataSourceArn":"string","Name":"string","SqlQuery":"str
{"Catalog":"string","DataSourceArn":"string","InputColumns":
[{"Name":"string","Type":"string"}],"Name":"string","Schema":"string"},"S3Source":
{"DataSourceArn":"string","InputColumns":
[{"Name":"string","Type":"string"}],"UploadSettings":
{"ContainsHeader":boolean,"Delimiter":"string","Format":"string","StartFromRow":number,"Text
    --import-mode DIRECT_QUERY
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-data-set
    --cli-input-json file://createdataset.json
```

For more information about the `CreateDataSet` API operation, see CreateDataSet in the *Amazon Quick Sight API Reference*.

DeleteDataSet

Use the `DeleteDataSet` API operation to delete a dataset. To use this operation, you need the ID of the dataset that you want to delete. The dataset ID is part of the dataset URL in Quick Sight. You can also use the `ListDataSets` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-data-set
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
```

For more information about the `DeleteDataSet` API operation, see DeleteDataSet in the *Amazon Quick Sight API Reference*.

DescribeDataSet

Use the `DescribeDataSet` API operation to describe a dataset. To use this operation, you need the ID of the dataset that you want to describe. The dataset ID is part of the dataset URL in Quick Sight. You can also use the `ListDataSets` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-data-set
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
```

For more information about the `DescribeDataSet` API operation, see DescribeDataSet in the *Amazon Quick Sight API Reference*.

ListDataSets

Use the `ListDataSets` API operation to list all of the datasets that belong to a particular AWS account in an AWS Region. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-data-sets
    --aws-account-id AWSACCOUNTID
    --page-size 10
```

```
    --max-items 100
```

For more information about the `ListDataSets` API operation, see ListDataSets in the *Amazon Quick Sight API Reference*.

UpdateDataSet

Use the `UpdateDataSet` API operation to update a dataset. To use this operation, you need the ID of the dataset that you want to update. The dataset ID is part of the dataset URL in Quick Sight. You can also use the `ListDataSets` API operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-data-set
   --aws-account-id AWSACCOUNTID
   --data-set-id DATASETID
   --name NAME
   --physical-table-map PHYSICALTABLEMAP
   --import-mode IMPORTMODE
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-data-set
    --cli-input-json file://updatedataset.json
```

For more information about the `UpdateDataSet` API operation, see UpdateDataSet in the *Amazon Quick Sight API Reference*.

Folder operations

In Quick Sight Enterprise Edition, you can create personal and shared folders to add hierarchical structure to Quick Sight asset management. Using folders, people can more easily organize, navigate through, and discover dashboards, analyses, and datasets. Within a folder, you can still use your usual tools to search for assets or to add assets to your favorites list.

For more information about folders, see Organizing Assets into Folders for Quick Sight in the *Amazon Quick Sight User Guide*.

Using the AWS CLI, you can use the following operations to create, search, update, and delete folders in your Quick Sight account:

Topics

Folder membership operations

With folder membership API operations, you can view and update assets, such as a dashboard, analysis, or dataset, to a folder. For more information, see the following API operations:

- CreateFolderMembership
- DeleteFolderMembership
- ListFolderMembers

CreateFolderMembership

Use the `CreateFolderMembership` to add an asset, such as a dashboard, analysis, or dataset, to a folder. To use this operation, you need the member ID of the asset that you want to add to a folder. The member ID is either the dashboard, analysis, or dataset ID of the analysis, dashboard, or dataset that you want to add to a folder. The member ID is part of the analysis, dashboard, or dataset URL in Quick Sight. You can also use the `ListAnalyses`, `ListDashboards`, or `ListDataSets` operations to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-folder-membership
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
    --member-id 444455556666
    --member-type DASHBOARD
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-folder-membership
    --cli-input-json file://createfoldermembership.json
```

For more information about the `CreateFolderMembership` operation, see CreateFolderMembership in the *Quick Sight API Reference*.

DeleteFolderMembership

Use the `DeleteFolderMembership` to delete an asset, such as a dashboard, analysis, or dataset, from a folder. To use this operation, you need the member ID of the asset that you want to add to a folder. The member ID is either the dashboard, analysis, or dataset ID of the analysis, dashboard, or dataset that you want to add to a folder. The member ID is part of the analysis, dashboard, or dataset URL in Quick Sight. You can also use the `ListAnalyses`, `ListDashboards`, or `ListDataSets` operations to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-folder-membership
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
    --member-id 444455556666
    --member-type DASHBOARD
```

For more information about the `DeleteFolderMembership` operation, see DeleteFolderMembership in the *Quick Sight API Reference*.

ListFolderMembers

Use the `ListFolderMembers` operation to list all assets (`DASHBOARD`, `ANALYSIS`, and `DATASET`) that are in a folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-folder-members
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
    --page-size 10
    --max-items 100
```

For more information about the `ListFolderMembers` operation, see ListFolderMembers in the *Quick Sight API Reference*.

Folder permissions operations

With folder permission API operations, you can view and update permissions for folders. For more information, see the following API operations:

- UpdateFolderPermissions
- DescribeFolderPermissions
- DescribeFolderResolvedPermissions

DescribeFolderPermissions

Use the `DescribeFolderPermissions` operation to describe the permissions of a folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-folder-permissions
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
```

You can find the folder ID by using a `ListFolders` operation or through the URL in the Quick Sight user interface.

For more information about the `DescribeFolderPermissions` operation, see DescribeFolderPermissions in the *Quick Sight API Reference*.

DescribeFolderResolvedPermissions

Use the `DescribeFolderResolvedPermissions` operation to describe the resolved permissions of a folder. Permissions consist of both folder direct permissions and the inherited permissions from the ancestor folders. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-folder-resolved-permissions
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
```

For more information about the `DescribeFolderResolvedPermissions` operation, see DescribeFolderResolvedPermissions in the *Quick Sight API Reference*.

UpdateFolderPermissions

Use the `UpdateFolderPermissions` operation to update the permissions of a folder. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-folder-permissions --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:CreateFolder,quicksight:DescribeFolder,quicksight:UpdateFolder,q
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/
USERNAME,Actions=quicksight:CreateFolder,quicksight:DescribeFolder,quicksight:UpdateFolder,q
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-folder-permissions
    --cli-input-json file://updatefolderpermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information on the `UpdateFolderPermissions` operation, see UpdateFolderPermissions in the *Quick Sight API Reference*.

CreateFolder

The `CreateFolder` operation creates an empty shared folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-folder
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-folder
    --cli-input-json file://createfolder.json
```

For more information about the `CreateFolder` operation, see CreateFolder in the *Quick Sight API Reference*.

DeleteFolder

Use the `DeleteFolder` operation to delete an empty folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-folder
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
```

For more information about the `DeleteFolder` operation, see DeleteFolder in the *Quick Sight API Reference*.

DescribeFolder

Use the `DescribeFolder` operation to describe a folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-folder
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
```

For more information about the `DescribeFolder` operation, see DescribeFolder in the *Quick Sight API Reference*.

ListFolders

Use the `ListFolders` operation to list all folders in an Quick Sight account.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-folders
    --aws-account-id AWSACCOUNTID
    --page-size 10
    --max-items 100
```

For more information about the `ListFolders` operation, see ListFolders in the *Quick Sight API Reference*.

SearchFolders

Use the `SearchFolders` operation to search the subfolders of a folder.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight search-folders
    --aws-account-id AWSACCOUNTID
    --filters
 Operator=StringEquals,Name=QUICKSIGHT_USER,Value=arn:aws:quicksight:us-
east-1:AWSACCOUNTID:user/default/USERNAME
    --max-results 100
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information on the SearchFolders operation, see SearchFolders in the *Quick Sight API Reference*.

UpdateFolder

Use the `UpdateFolder` operation to update the name of a folder. To use this operation, you need the ID of the folder whose permissions you want to view. The folder ID is part of the folder URL in Quick Sight. You can also use the `ListFolders` operation to get the ID.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-folder
    --aws-account-id AWSACCOUNTID
    --folder-id FOLDERID
    --name NAME
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-folder
    --cli-input-json file://updatefolder.json
```

For more information about the `UpdateFolder` operation, see UpdateFolder in the *Quick Sight API Reference*.

Group operations

With group API operations, you can perform actions on groups. For more information, see the following API operations.

Topics

- Group membership operations
- CreateGroup
- DeleteGroup
- DescribeGroup

- ListGroups
- SearchGroups
- UpdateGroup

Group membership operations

With group membership API operations, you can view and update permissions for members in a group. For more information, see the following API operations.

Topics

- CreateGroupMembership
- DeleteGroupMembership
- DescribeGroupMembership
- ListGroupMemberships

CreateGroupMembership

Use the `CreateGroupMembership` API operation to add an Amazon Quick Sight user to a Quick Sight group. You can find users in a certain group by calling the `ListGroups` API operation, and then the `ListGroupMemberships` API operation on the group of your choice.

Following is an example AWS CLI command for this operation. In the following examples, the member *USERNAME* is added to the group *GROUPNAME*.

AWS CLI

```
aws quicksight create-group-membership
    --namespace default
    --aws-account-id AWSACCOUNTID
    --group-name GROUPNAME
    --member-name USERNAME
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-group-membership
    --cli-input-json file://creategroupmembership.json
```

For more information about the `CreateGroupMembership` API operation, see CreateGroupMembership in the *Amazon Quick Sight API Reference*.

DeleteGroupMembership

Use the `DeleteGroupMembership` API operation to remove a user from a group so that the user is no longer a member of the group. You can find users in a certain group by calling the `ListGroups` API operation, and then the `ListGroupMemberships` operation on the group that you choose.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-group-membership
    --member-name USERNAME
    --group-name GROUPNAME
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE
```

For more information about the `DeleteGroupMembership` API operation, see DeleteGroupMembership in the *Amazon Quick Sight API Reference*.

DescribeGroupMembership

Use the `DescribeGroupMembership` API operation to determine if a user is a member of the specified group. If the user exists and is a member of the specified group, an associated `GroupMember` object is returned.

Following is an example AWS CLI command for this operation.

AWS CLI

CLI Input:

```
aws quicksight describe-group-membership
    --region us-west-2
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE
    --group-name Marketing-East
```

```
    --member-name MEMBERNAME
```

For more information about the `ListGroups` API operation, see DescribeGroupMembership in the *Amazon Quick Sight API Reference*.

ListGroupMemberships

Use the `ListGroupMemberships` API operation to list member users in a group. To view a list of user groups in Amazon Quick Sight, call the `ListGroups` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-group-memberships
    --group-name GROUPNAME
    --max-results 100
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE
```

For more information about the `ListGroupMemberships` API operation, see ListGroupMemberships in the *Amazon Quick Sight API Reference*.

CreateGroup

Use the `CreateGroup` API operation to create a user group in Amazon Quick Sight. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-group
    --namespace NAMESPACE
    --aws-account-id AWSACCOUNTID
    --group-name GROUPNAME
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-group
```

```
    --cli-input-json file://creategroup.json
```

For more information about the `CreateGroup` API operation, see CreateGroup in the *Amazon Quick Sight API Reference*.

DeleteGroup

Use the `DeleteGroup` API operation to remove a user group from Amazon Quick Sight. You can find a group name by calling the `ListGroups` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-group
    --group-name GROUPNAME
    --aws-account-id AWSACCOUNTID
    --namespace default
```

For more information about the `DeleteGroup` API operation, see DeleteGroup in the *Amazon Quick Sight API Reference*.

DescribeGroup

Use the `DescribeGroup` API operation to view an Amazon Quick Sight group's description and Amazon Resource Name (ARN). You can find a group name by calling the `ListGroups` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-group
    --group-name GROUPNAME
    --aws-account-id AWSACCOUNTID
    --namespace default
```

For more information about the `DescribeGroup` API operation, see DescribeGroup in the *Amazon Quick Sight API Reference*.

ListGroups

Use the `ListGroups` API operation to list all user groups in Amazon Quick Sight. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-groups
    --aws-account-id AWSACCOUNTID
    --max-results 100
    --namespace default
```

For more information about the `ListGroups` API operation, see ListGroups in the *Amazon Quick Sight API Reference*.

SearchGroups

Use the `SearchGroups` operation to search groups in a specified Quick Sight namespace using the supplied filters.

Following is an example AWS CLI command for this operation.

AWS CLI

CLI Input:

```
aws quicksight search-groups
    --region us-west-2
    --aws-account-id AWSACCOUNTID
    --namespace default
    --filters "[{\"Operator\": \"StringLike\", \"Name\": \"GROUP_NAME\", \"Value\":
 \"Mar\"}]"
```

For more information about the `SearchGroups` API operation, see SearchGroups in the *Amazon Quick Sight API Reference*.

UpdateGroup

Use the `UpdateGroup` API operation to change a group description. You can find a group name by calling the `ListGroups` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-group
    --group-name GROUPNAME
    --description "NEW DESCRIPTION"
    --aws-account-id AWSACCOUNTID
    --namespace default
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-group
    --cli-input-json file://updategroup.json
```

For more information about the `UpdateGroup` API operation, see UpdateGroup in the *Amazon Quick Sight API Reference*.

IAM policy assignment operations

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator to securely control access to AWS resources. Administrators control who can be authenticated (signed in) and authorized (have permissions) to use Amazon Quick Sight resources. For more information about using Quick Sight with IAM, see Using AWS Identity and Access Management (IAM) in the *Amazon Quick Sight User Guide*.

With IAM policy assignment operations, you can create, update, and delete IAM policy assignments. For more information, see the following API operations.

Topics

- CreateIAMPolicyAssignment
- DeleteIAMPolicyAssignment
- DescribeIAMPolicyAssignment
- ListIAMPolicyAssignments
- ListIAMPolicyAssignmentsForUser
- UpdateIAMPolicyAssignment

CreateIAMPolicyAssignment

Use the `CreateIAMPolicyAssignment` API operation to create an assignment with one specified IAM policy, identified by its Amazon Resource Name (ARN). This policy assignment is attached to the specified groups or users of Amazon Quick Sight. Assignment names are unique for each AWS account. To avoid overwriting rules in other namespaces, use assignment names that are unique.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-iam-policy-assignment
    --aws-account-id AWSACCOUNTID
    --assignment-name ASSIGNMENT
    --assignment-status ENABLED
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-iam-policy-assignment
    --cli-input-json file://createiampolicyassignment.json
```

For more information about the `CreateIAMPolicyAssignment` API operation, see CreateIAMPolicyAssignment in the *Amazon Quick Sight API Reference*.

DeleteIAMPolicyAssignment

Use the `DeleteIAMPolicyAssignment` API operation to delete an existing IAM policy assignment.

To find a policy assignment name, call the `ListIAMPolicyAssignments` or `ListIAMPolicyAssignmentsForUser` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-iam-policy-assignment
    --aws-account-id AWSACCOUNTID
```

```
    --assignment-name ASSIGNMENT
    --namespace default
```

For more information about the `DeleteIAMPolicyAssignment` API operation, see DeleteIAMPolicyAssignment in the *Amazon Quick Sight API Reference*.

DescribeIAMPolicyAssignment

Use the `DescribeIAMPolicyAssignment` API operation to describe an existing IAM policy assignment.

To find a policy assignment name, call the `ListIAMPolicyAssignments` or `ListIAMPolicyAssignmentsForUser` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-iam-policy-assignment
    --aws-account-id AWSACCOUNTID
    --assignment-name ASSIGNMENT
    --namespace default
```

For more information about the `DescribeIAMPolicyAssignment` API operation, see DescribeIAMPolicyAssignment in the *Amazon Quick Sight API Reference*.

ListIAMPolicyAssignments

Use the `ListIAMPolicyAssignments` API operation to list IAM policy assignments in the current Amazon Quick Sight account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-iam-policy-assignments
    --aws-account-id AWSACCOUNTID
    --assignment-status ENABLED
    --namespace default
    --max-results 100
```

For more information about the `ListIAMPolicyAssignments` API operation, see ListIAMPolicyAssignments in the *Amazon Quick Sight API Reference*.

ListIAMPolicyAssignmentsForUser

Use the `ListIAMPolicyAssignmentsForUser` API operation to list all the IAM policy assignments, including the Amazon Resource Names (ARNs), for the IAM policies assigned to the specified user and the groups that the user belongs to.

To find a user name, call the `ListUsers` API operation.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-iam-policy-assignments-for-user
    --aws-account-id AWSACCOUNTID
    --user-name USER
    --max-results 100
    --namespace default
```

For more information about the `ListIAMPolicyAssignmentsForUser` API operation, see ListIAMPolicyAssignmentsForUser in the *Amazon Quick Sight API Reference*.

UpdateIAMPolicyAssignment

Use the `UpdateIAMPolicyAssignment` API operation to update an existing IAM policy assignment. This operation updates only the optional parameters that are specified in the request. It overwrites all of the users included in `Identities`.

To find a policy assignment name, call the `ListIAMPolicyAssignments` or `ListIAMPolicyAssignmentsForUser` API operations.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-iam-policy-assignment
    --aws-account-id AWSACCOUNTID
    --assignment-name NAME
    --namespace default
```

```
    --assignment-status ENABLED
    --policy-arn 222244446666
    --identities KEY=VALUE,VALUE,KEY=VALUE,VALUE
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-iam-policy-assignment
    --cli-input-json file://updateiampolicyassignment.json
```

For more information about the `UpdateIAMPolicyAssignment` API operation, see UpdateIAMPolicyAssignment in the *Amazon Quick Sight API Reference*.

Ingestion operations

With ingestion API operations, you can perform actions on Quick Sight ingestions. For more information, see the following API operations.

Topics

- CancelIngestion
- CreateIngestion
- DescribeIngestion
- ListIngestions

CancelIngestion

Use the `CancelIngestion` operation to cancel an ongoing ingestion of data into SPICE.

To use this operation, you need the ID of the dataset that is undergoing the ingestion that you want to cancel and the ID of the ingestion you want to cancel. You can use the `ListDataSets` operation to list all datasets and their corresponding dataset IDs. You can use the `ListIngestions` operation to list all ingestion IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight cancel-ingestion
```

```
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
    --ingestion-id INGESTIONID
```

For more information about the `CancelIngestion` operation, see CancelIngestion in the *Quick Sight API Reference*.

CreateIngestion

Use the `CreateIngestion` to create and start a new SPICE ingestion on a dataset.

Any ingestions operating on tagged datasets inherit the same tags automatically for use in access control. For an example, see How do I create an IAM policy to control access to Amazon EC2 resources using tags? in the AWS Knowledge Center. Tags are visible on the tagged dataset, but not on the ingestion resource.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-ingestion
    --data-set-id DATASETID
    --ingestionid INGESTIONID
    --aws-account-id AWSACCOUNTID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-ingestion
    --cli-input-json file://createingestion.json
```

For more information about the `CreateIngestion` operation, see CreateIngestion in the *Quick Sight API Reference*.

DescribeIngestion

Use the `DescribeIngestion` operation to describe a SPICE ingestion.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-ingestion
    --aws-account-id AWSACCOUNTID
    --data-set-id DATASETID
    --ingestion-id INGESTIONID
```

For more information about the `DescribeIngestion` operation, see DescribeIngestion in the *Quick Sight API Reference*.

ListIngestions

Use the `ListIngestions` operation to list the history of SPICE ingestioned for a dataset.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-ingestions
    --data-set-id DATASETID
    --aws-account-id AWSACCOUNTID
    --page-size 10
    --max-items 100
```

For more information about the `ListIngestions` operation, see ListIngestions in the *Quick Sight API Reference*.

IP and VPC endpoint restriction operations

With IP and VPC endpoint restriction API operations, you can perform actions on Quick Sight IP and VPC endpoint restrictions. For more information, see the following API operations.

Topics

- DescribeIpRestriction
- UpdateIpRestriction
- Quick and interface VPC endpoints (AWS PrivateLink)

DescribeIpRestriction

Use the `DescribeIpRestriction` operation to get a summary and status of IP rules.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-ip-restriction \
    --aws-account-id AWSACCOUNTID
```

For more information about the `DescribeIpRestriction` operation, see DescribeIpRestriction in the *Quick Sight API Reference*.

UpdateIpRestriction

Use the `UpdateIpRestriction` operation to update the content and status of IP rules. To use this operation, provide the entire map of rules. You can use the `DescribeIpRestriction` operation to get the current rule map.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-ip-restriction \
    --aws-account-id AWSACCOUNTID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-ip-restriction \
    --cli-input-json file://updateiprestriction.json
```

For more information about the `UpdateIpRestriction` operation, see UpdateIpRestriction in the *Quick Sight API Reference*.

Quick and interface VPC endpoints (AWS PrivateLink)

You can establish a private connection between your VPC and Quick by creating an *interface VPC endpoint*. Interface endpoints are powered by AWS PrivateLink, a technology that enables you to privately access the Quick website and APIs without leaving the Amazon network. Instances in your VPC don't need public IP addresses to communicate with Quick website and APIs, but still need access to certain domains other than Quick so that static assets, reports, and other files can be downloaded. For a list of domains that Quick needs to access, see Domains accessed by Quick.

Each interface endpoint is represented by one or more Elastic Network Interfaces in your subnets.

For more information, see Interface VPC endpoints (AWS PrivateLink) in the *Amazon VPC User Guide*.

Considerations for Quick VPC endpoints

Before you set up an interface VPC endpoint for Quick, ensure that you review Interface endpoint properties and limitations in the *Amazon VPC User Guide*.

The following considerations apply to VPC endpoint restrictions in Quick:

- Quick supports data sources from AWS services including Amazon S3, Amazon Redshift, and Athena. Quick needs access to the resources from your AWS accounts to retrieve this data. If you want traffic to other AWS services to be routed through the VPC endpoint, you need to create VPC endpoint connections for each service that your Quick account is configured to. For more information about connecting to a VPC connection with Quick, see Connecting to a VPC with Quick.
- IP and VPC endpoint rules precede all other rules in Quick. If you have embedded dashboards or visuals that are visible to the public (anyone on the internet) and restrict traffic to the Quick website through a VPC endpoint, public dashboards can only be shared through the VPC endpoint. For more information on public embedding, see Turning on public access to visuals and dashboards with a 1-click embed code.
- Quick Website VPC endpoints are not available in China regions & Govcloud.

Creating an interface VPC endpoint for Quick Website

You can create a VPC endpoint for the Quick website using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Create VPC endpoints for Quick using the following service names:

- `com.amazonaws.region.quicksight-website` - For Quick website access

The private DNS names for the Quick website are not same as the public URL for Quick. To reach Quick through the public URL, create an A record for the website in the format `<region>.quicksight.aws.amazon.com` and point it to the VPC endpoint. For more information about routing to a VPC endpoint, see Routing traffic to an Amazon Virtual Private Cloud interface endpoint by using your domain name.

The management of certain administrator features require that an administrator sign in to Quick as an IAM user. If you sign in through the VPC endpoint, you need to create the following VPC endpoints for the AWS Management Console.

- `com.amazonaws.region.console`
- `com.amazonaws.region.signin`

For more information about VPC endpoints for the AWS Management Console, see Required VPC endpoints and DNS configuration.

Creating a VPC endpoint policy for Quick Website

You can attach an endpoint policy to your VPC endpoint to restrict usage of the endpoint to specific Quick accounts or to accounts under specific AWS organizations. The AWS account IDs that are allow–listed or deny–listed are the AWS accounts in which the Quick account is created. In most cases, this is the same account ID in which the VPC endpoint is created. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Quick Website actions

The following is an example of an endpoint policy for Quick. When attached to an endpoint, this policy grants access to all Quick actions for all principals on all resources.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": "*",
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": [
                        "012345678901"
                    ]
                }
            }
        }
    ]
}
```

Policies for the Quick website must have the values of the `Principal`, `Action`, and `Resource` fields set to `"*"`. A condition may be specified only against the `aws:PrincipalAccount` or the `aws:OrgId attributes`. These conditions are evaluated on all requests to the Quick website and API calls after the user signs in.

Creating an interface VPC endpoint for Quick APIs

You can create a VPC endpoint for the Quick APIs using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Creating an interface endpoint in the *Amazon VPC User Guide*.

Create VPC endpoints for Quick using the following service names:

- `com.amazonaws.region.quicksight`
- For Quick API access through FIPS endpoint- `com.amazonaws.region.quicksight-fips`

When you create a VPC endpoint for Quick APIs, the private DNS resolution automatically routes API calls to the VPC endpoint. No additional DNS configuration is required - your existing API calls to `quicksight.<region>.amazonaws.com` will automatically use the VPC endpoint when private DNS is enabled.

For more information about VPC endpoints for the AWS Management Console, see Required VPC endpoints and DNS configuration.

Following APIs are not supported via interface VPC endpoint Quick API:

API Name
CreateActionConnector
DeleteActionConnector
DescribeActionConnector
DescribeActionConnectorPermissions
ListActionConnectors
SearchActionConnectors
UpdateActionConnector
UpdateActionConnectorPermissions
GetFlowMetadata
GetFlowPermissions
ListFlows
SearchFlows
UpdateFlowPermissions

Creating a VPC endpoint policy for Quick APIs

You can attach an endpoint policy to your VPC endpoint to restrict usage of the endpoint to specific Quick accounts or to accounts under specific AWS organizations. The AWS account IDs that are allow–listed or deny–listed are the AWS accounts in which the Quick account is created. In most cases, this is the same account ID in which the VPC endpoint is created. The policy specifies the following information:

- The principal that can perform actions.
- The actions that can be performed.
- The resources on which actions can be performed.

For more information, see Controlling access to services with VPC endpoints in the *Amazon VPC User Guide*.

Example: VPC endpoint policy for Quick API actions

The following is an example of an endpoint policy for Quick APIs. When attached to an endpoint, this policy grants access to all Quick actions for specific Quick actions and conditions.

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "quicksight:DescribeUser",
                "quicksight:ListUsers"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalAccount": [
                        "012345678901"
                    ]
                }
            }
        }
    ]
```

```
}
```

Restricting access to the Quick website

You can choose to restrict access to your Quick account to only allow traffic from an approved VPC endpoint. This prevents general internet users from accessing your Quick account. Before you can make this change, make sure that you're an IAM user with the `UpdateIpRestriction` permission. For more information on the permissions that are required to restrict access with a VPC endpoint, see Turning on IP and VPC endpoint restrictions in Quick.

Use the following procedure to restrict access with a VPC endpoint in Quick.

1. Open the Quick console.
2. Choose **Manage Quick**, and then choose **Security & permissions**.
3. On the **Security & permissions** page that opens, navigate to **IP and VPC endpoint restrictions** and choose **Manage**.
4. Turn on the **Enforce restrictions** switch to turn on your VPC endpoint restrictions.

You can also perform this action with the Quick APIs. The following example turns on the enforcement of a VPC endpoint restriction.

```
aws quicksight update-ip-restriction \
--aws-account-id AWSACCOUNTID \
--region REGION \
--enabled \
--vpc-endpoint-id-restriction-rule-map vpce-001122def=MyVpcEndpointAllowed
```

Domains accessed by Quick

The table below lists all URLs that are accessed by Quick from your browser. Make sure that you have established connectivity for all of domains listed in the table.

URL	Reason	Has VPC endpoint support?
region.quicksight.aws.amazon.com	The bulk of traffic to Quick flows through this domain.	Yes

URL	Reason	Has VPC endpoint support?
quicksight.region.amazonaws.com	Quick public API calls.	Yes
signin.aws.amazon.com	To sign in to the AWS console if the account uses IAM identities.	Yes
region.signin.aws	To sign in to the AWS console if the account uses or Quick native users for identity management.	No
*.cloudfront.net	To download static assets, for example CSS or JS.	No
*.s3.region.amazonaws.com	To download reports and thumbnails.	Yes
*.execute-api.region.amazonaws.com	To access client-side metrics.	No
https://*.kinesisvideo.amazonaws.com	To allow live streaming of automation workflows	No
https://apis.google.com/js/api.js	To allow google drive file picker	NA
https://*.officeapps.live.com	To allow Quick side panel extenstion	NA
https://outlook.cloud.microsoft	To allow Quick side panel extenstion	NA
https://*.sharepoint.com	To allow Quick side panel extenstion	NA

URL	Reason	Has VPC endpoint support?
https://*.office.com	To allow Quick side panel extenstion	NA
https://*.office365.com	To allow Quick side panel extenstion	NA

Key management operations

Use Quick Sight key management APIs to list and update customer managed keys (CMKs) that are registered to a Quick Sight account. For more information about key management in Quick Sight, see Key management in the Quick Sight User Guide.

Permissons

Before you begin, create or update an IAM role that contains a user permission to access and use all CMKs that are registered to your Quick Sight account. At minimum, the IAM policy must contain the `kms:CreateGrant`, `quicksight:UpdateKeyRegistration`, and `quicksight:DescribeKeyRegistration` permissions. To see a list of IAM policy examples that can be used to grant different degrees of access to the CMKs in a account, see IAM identity-based policies for Amazon Quick Sight: using the admin key management console.

CMK API Examples

The example below lists all customer managed keys that are registered to a Quick Sight account.

```
aws quicksight describe-key-registration \
--aws-account-id AWSACCOUNTID \
--region REGION
```

The example below updates a CMK registration and designates a default key.

```
aws quicksight update-key-registration \
--aws-account-id AWSACCOUNTID \
--key-registration '[{"KeyArn": "KEYARN", "DefaultKey": true}]'
--region REGION
```

The example below updates the registration of two CMKs in a Quick Sight account and designates one of the two updated keys as the new default key.

```
aws quicksight update-key-registration \
--aws-account-id AWSACCOUNTID \
--key-registration '[{"KeyArn": "KEYARN", "DefaultKey": true}, {"KeyArn": "KEYARN",
 "DefaultKey": false}]'
--region REGION
```

The example below clears all CMK registrations from a Quick Sight account. Instead, Quick Sight uses AWS owned keys to encrypt your resources.

```
aws quicksight update-key-registration \
--aws-account-id AWSACCOUNTID \
--key-registration '[]'
--region REGION
```

Namespace operations

An Amazon Quick Sight *namespace* is a logical container that you can use to organize clients, subsidiaries, teams, and so on. By using a namespace, you can isolate the Amazon Quick Sight users and groups that are registered for that namespace. Users that access the namespace can share assets only with other users or groups in the same namespace. They can't see users and groups in other namespaces. For more information about namespaces, see Supporting multitenancy with isolated namespaces in the *Quick Sight User guide*.

To implement namespaces, you use the following Quick Sight API operations.

Topics

- CreateNamespace
- DeleteNamespace
- DescribeNamespace
- ListNamespaces

CreateNamespace

Use the `CreateNamespace` API operation to create a new namespace for you to use with Amazon Quick Sight.

You can create a namespace after your AWS account is subscribed to Amazon Quick Sight. The namespace must be unique within the AWS account. By default, there is a limit of 100 namespaces per AWS account. To increase your limit, create a ticket with AWS Support.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-namespace
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE \
    --identity-store QUICKSIGHT
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-namespace
    --cli-input-json file://createnamespace.json
```

For more information about the `CreateNamespace` API operation, see CreateNamespace in the *Amazon Quick Sight API Reference*.

DeleteNamespace

Use the `DeleteNamespace` API operation to delete a namespace and the users and groups that are associated with the namespace. This is an asynchronous process. Assets including dashboards, analyses, datasets, and data sources are not deleted. To delete these assets, you use the relevant API operations for each asset, such as `DeleteDashboard` or `DeleteDataSet`.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-namespace
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE
```

Find a namespace by running the `ListNamespaces` operation.

For more information about the `DeleteNamespace` API operation, see DeleteNamespace in the *Amazon Quick Sight API Reference*.

DescribeNamespace

Use the `DescribeNamespace` API operation to describe a specified namespace. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-namespace
    --aws-account-id AWSACCOUNTID
    --namespace NAMESPACE
```

For more information about the `DescribeNamespace` API operation, see DescribeNamespace in the *Amazon Quick Sight API Reference*.

ListNamespaces

Use the `ListNamespaces` API operation to list namespaces for a specified AWS account. Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-namespaces
    --aws-account-id AWSACCOUNTID
    --page-size 10
    --max-items 100
```

For more information about the `ListNamespaces` API operation, see ListNamespaces in the *Amazon Quick Sight API Reference*.

Tag operations

Tags can help you categorize and allocate costs incurred by your Quick Sight resources. For more information about tags, see User-defined cost allocation tags. You can visualize costs of tagged resources that have consumption-based pricing in AWS cost and usage reports. For more information on cost and usage reports, see What are AWS Cost and Usage Reports.

You can also use tags to scope user permissions by granting a user permission to access or change only resources with certain tag values. You can use the TagResource API operation with a resource that already has tags. If you specify a new tag key for the resource, this tag is appended to the list of tags associated with the resource. If you specify a tag key that is already associated with the resource, the new tag value that you specify replaces the previous value for that tag. You can tag a new Quick Sight managed user or IAM user at creation with a RegisterUser API call.

You can associate as many as 50 tags with a resource. Amazon Quick Sight supports tagging for a data sets, data sources, dashboards, users, and templates.

Tagging for Quick Sight works in a similar way to tagging for other AWS services. Quick Sight doesn't currently support the tag editor for AWS Resource Groups.

Tags that are used for Admin Pro, Author Pro, or Reader Pro users can't be used as cost allocation tags.

For more information about the Tag API operations, see the following topics.

Topics

- ListTagsForResource
- TagResource
- UntagResource

- RegisterUser

ListTagsForResource

Use the `ListTagsForResource` API operation to list tags assigned to a resource.

Following is an example AWS CLI command for this operation. To find a resource's Amazon Resource Name (ARN), use the `List` operation for the resource. For example, `ListDashboards`.

AWS CLI

```
aws quicksight list-tags-for-resource
    --resource-arn 444455556666
```

For more information about the `ListTagsForResource` API operation, see ListTagsForResource in the *Amazon Quick Sight API Reference*.

TagResource

Use the `TagResource` API operation to assign one or more tags (key-value pairs) to the specified Amazon Quick Sight resource.

Following is an example AWS CLI command for this operation. To find a resource's Amazon Resource Name (ARN), use the `List` operation for the resource, for example `ListDashboards`.

AWS CLI

```
aws quicksight tag-resource
    --resource-arn 777788889999
    --tags Key=NewDashboard,Value=True
```

For more information about the `TagResource` API operation, see TagResource in the *Amazon Quick Sight API Reference*.

UntagResource

Use the `UntagResource` API operation to remove a tag from a resource. Before you do so, you can call the `ListTagsForResource` API operation to list the tags assigned to a resource.

Following is an example AWS CLI command for this operation. To find a resource's Amazon Resource Name (ARN), use the `List` operation for the resource, for example `ListDashboards`.

AWS CLI

```
aws quicksight untag-resource
    --resource-arn 777788889999
    --tag-keys NewDashboard,ExampleDashboard
```

For more information about the `UntagResource` API operation, see UntagResource in the *Amazon Quick Sight API Reference*.

Template alias operations

A *template alias* is a reference to a version of a template. For example, suppose that you create the template alias `exampleAlias` for version 1 of the template `exampleTemp`. You can use the template alias `exampleAlias` to reference version 1 of template `exampleTemp` in a `DescribeTemplate` API operation, as in the following example.

```
aws quicksight describe-template
    --aws-account-id AWSACCOUNTID
    --template-id exampleTempID
    --alias-name exampleAlias
```

With template alias API operations, you can perform actions on Quick Sight template aliases. For more information, see the following API operations.

Topics

- CreateTemplateAlias
- DeleteTemplateAlias
- DescribeTemplateAlias
- ListTemplateAliases
- UpdateTemplateAlias

CreateTemplateAlias

Use the `CreateTemplateAlias` operation to create a template alias for a template. To use this operation, you need the ID of the template that you want to create an alias for. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-template-alias
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --alias-name ALIAS
    --template-version-number VERSION
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-template-alias
    --cli-input-json file://createtemplatealias.json
```

For more information about the `CreateTemplateAlias` operation, see CreateTemplateAlias in the *Quick Sight API Reference*.

DeleteTemplateAlias

Use the `DeleteTemplateAlias` operation to delete the item that the specified template alias points to. If you provide a specific alias, you delete the version of the template that the alias points to. To use this operation, you need the ID of the template that is using the alias you want to delete. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-template-alias
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --alias-name ALIAS
```

For more information about the `DeleteTemplateAlias` operation, see DeleteTemplateAlias in the *Quick Sight API Reference*.

DescribeTemplateAlias

Use the `DescribeTemplateAlias` operation to describe the template alias for a template. To use this operation, you need the ID of the template that is using the alias that you want to describe. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-template-alias
    --aws-account-id AWSACCOUNTID
    --template-id 222244446666
    --alias-name ALIAS
```

The parameter value for `alias-name` can be `$LATEST`.

For more information about the `DescribeTemplateAlias` operation, see DescribeTemplateAlias in the *Quick Sight API Reference*.

ListTemplateAliases

Use the `ListTemplateAliases` operation to list all the aliases of a template. To use this operation, you need the ID of the template that is using the aliases that you want to list. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-template-aliases
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --page-size 10
    --max-items 100
```

For more information about the `ListTemplateAliases` operation, see ListTemplateAliases in the *Quick Sight API Reference*.

UpdateTemplateAlias

Use the `UpdateTemplateAlias` operation to update the template alias of a template. To use this operation, you need the ID of the template that is using the alias that you want to update. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-template-alias
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --alias-name ALIAS
    --template-version-number VERSION
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-template-alias
    --cli-input-json file://updatetemplatealias.json
```

The parameter value for `alias-name` can be `$LATEST`.

For more information about the `UpdateTemplateAlias` operation, see UpdateTemplateAlias in the *Quick Sight API Reference*.

Template operations

A *template* is a resource in Quick Sight that holds the information necessary to create an analysis or dashboard. You can use templates to migrate dashboards and analyses across accounts.

With template API operations, you can perform actions on Quick Sight templates. For more information, see the following API operations.

Topics

- Template permissions
- CreateTemplate
- DeleteTemplate
- DescribeTemplate
- ListTemplates
- ListTemplateVersions
- UpdateTemplate

Template permissions

With template permissions API operations, you can view and update permissions for templates. For more information, see the following API operations.

- DescribeTemplatePermissions
- UpdateTemplatePermissions

DescribeTemplatePermissions

Use the `DescribeTemplatePermissions` operation to describe read and write permissions for a template. To use this operation, you need the ID of the template that you want to describe the permissions of. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-template-permissions
    --aws-account-id AWSACCOUNTID
    --template-id 222244446666
```

For more information about the `DescribeTemplatePermissions` operation, see DescribeTemplatePermissions in the *Quick Sight API Reference*.

UpdateTemplatePermissions

Use the `UpdateTemplatePermissions` operation updates the resource permissions for a template. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the template that you want to update the permissions of. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-template-permissions
    --aws-account-id AWSACCOUNTID
```

```
    --template-id TEMPLATEID
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/USERNAME,Actions=DescribeTemplate
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/USERNAME,Actions=DescribeTemplate
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-template-permissions
    --cli-input-json file://updatetemplatepermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information on the `UpdateTemplatePermissions` operation, see UpdateTemplatePermissions in the *Quick Sight API Reference*.

CreateTemplate

Use the `CreateTemplate` operation to create a template from an existing Quick Sight analysis or template. You can use the resulting template to create a dashboard.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-template
    --aws-account-id 555555555555
    --template-id TEMPLATEID
    --source-entity SOURCEENTITY
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-template
    --cli-input-json file://createtemplate.json
```

You can to get the ID the dataset ID by using a `DescribeAnalysis`operation. The `ANALYSISID`is part of the analysis URL in Quick Sight. You can also use the `ListAnalyses` operation to get the ID.

For more information about the `CreateTemplate` operation, see CreateTemplate in the *Quick Sight API Reference*.

DeleteTemplate

Use the `DeleteTemplate` operation to delete a template. To use this operation, you need the ID of the template that you want to delete. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-template
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
```

For more information about the `DeleteTemplate` operation, see DeleteTemplate in the *Quick Sight API Reference*.

DescribeTemplate

Use the `DescribeTemplate` operation to describe a template's metadata. To use this operation, you need the ID of the template that you want to describe. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-template
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --version-number VERSION
    --alias-name ALIAS
```

The parameter value for `alias-name` can be `$LATEST`.

For more information about the `DescribeTemplate` operation, see DescribeTemplate in the *Quick Sight API Reference*.

ListTemplates

Use the `ListTemplates` operation to list all the templates in the current Quick Sight account.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-templates
    --aws-account-id AWSACCOUNTID
    --page-size 10
    --max-items 100
```

For more information about the `ListTemplates` operation, see ListTemplates in the *Quick Sight API Reference*.

ListTemplateVersions

Use the `ListTemplateVersions` operation to list all the versions of the templates in the current Quick Sight account. To use this operation to list the versions of a template, you need that template's ID. You can use the `ListTemplates` operation to list all templates and their corresponding template IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-template-versions
    --aws-account-id AWSACCOUNTID
    --template-id TEMPLATEID
    --page-size 10
    --max-items 100
```

For more information about the `ListTemplateVersions` operation, see ListTemplateVersions in the *Quick Sight API Reference*.

UpdateTemplate

Use the `UpdateTemplate` operation to update a template from an existing Quick Sight analysis or another template.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-template
    --aws-account-id 555555555555
    --template-id TEMPLATEID
    --source-entity SOURCEENTITY
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-template
    --cli-input-json file://updatetemplate.json
```

For more information about the `UpdateTemplate` operation, see UpdateTemplate in the *Quick Sight API Reference*.

Theme operations

A *theme* is a collection of settings that you can apply to analyses and dashboards in Amazon Quick Sight. You can apply themes to modify the appearance of dashboards and analyses.

With theme operations, you can perform actions on Quick Sight themes. For more information, see the following API operations.

Topics

- Theme permissions
- CreateTheme
- DeleteTheme
- DescribeTheme

- ListThemes
- ListThemeVersions
- UpdateTheme

Theme permissions

With theme permissions API operations, you can view and update permissions for themes. For more information, see the following API operations.

- DescribeThemePermissions
- UpdateThemePermissions

DescribeThemePermissions

Use the `DescribeThemePermissions` operation to describe the read and write permissions for a theme. To use this operation, you need the ID of the theme that you want to describe. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-theme-permissions
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
```

For more information about the `UpdateThemePermissions` operation, see UpdateThemePermissions in the*Quick Sight API Reference*.

UpdateThemePermissions

Use the `UpdateThemePermissions` operation to update the resource permissions for a template. You can grant or revoke permissions in the same command. To use this operation, you need the ID of the theme that you want to update. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-theme-permissions
    --aws-account-id 555555555555
    --theme-id 111122223333
    --grant-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/USERNAME,Actions=quicksight:ListThemeVersions, quicksight:UpdateThemeAlias,
 quicksight: DescribeThemeAlias, quicksight:UpdateThemePermissions,
 quicksight:DeleteThemeAlias, quicksight: DeleteTheme, quicksight:ListThemeAliases,
 quicksight:DescribeTheme, quicksight: CreateThemeAlias, quicksight:UpdateTheme,
 quicksight: DescribeThemePermissions
    --revoke-permissions Principal=arn:aws:quicksight:us-east-1:AWSACCOUNTID:user/
default/USERNAME,Actions=quicksight:ListThemeVersions, quicksight:UpdateThemeAlias,
 quicksight: DescribeThemeAlias, quicksight:UpdateThemePermissions,
 quicksight:DeleteThemeAlias, quicksight: DeleteTheme, quicksight:ListThemeAliases,
 quicksight:DescribeTheme, quicksight: CreateThemeAlias, quicksight:UpdateTheme,
 quicksight: DescribeThemePermissions
```

If your `region` has already been configured within the CLI, it doesn't need to be included as an argument.

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-theme-permissions
    --cli-input-json file//:updatethemepermissions.json
```

If your region has already been configured with the CLI, it does not need to be included in an argument.

For more information on the `UpdateThemePermissions` operation, see UpdateThemePermissions in the *Quick Sight API Reference*.

CreateTheme

Use the `CreateTheme` operation to create a theme. The `base-theme-id` is the ID of the theme that you want to base the new theme off of. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight create-theme
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --name NAME
    --base-theme-id THEMEID
    --configuration '{"Configuration":{"DataColorPalette":{"Colors":
[""],"MinMaxGradient":[""],"EmptyFillColor":""},"UIColorPalette":
{"PrimaryForeground":"","PrimaryBackground":
 "","SecondaryForeground":"","SecondaryBackground":"","Accent":"","AccentForeground":"","Dan
{"Tile":{"Border":{"Show":true}},"TileLayout":{"Gutter":{"Show":true},"Margin":
{"Show":true}}}}'
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-theme
    --cli-input-json file//:createtheme.json
```

For more information about the `CreateTheme` operation, see CreateTheme in the*Quick Sight API Reference*.

DeleteTheme

Use the `DeleteTheme` operation to delete a theme. To use this operation, you need the ID of the theme that you want to delete. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-theme
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
```

For more information about the `DeleteTheme` operation, see DeleteTheme in the *Quick Sight API Reference*.

DescribeTheme

Use the `DescribeTheme` operation to describe a theme. To use this operation, you need the ID of the theme that you want to describe. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-theme
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --version-number 1
    --alias-name ALIAS
```

The parameter value for `alias-name` can be `$LATEST`.

For more information about the `DescribeTheme` operation, see DescribeTheme in the *Quick Sight API Reference*.

ListThemes

Use the `ListThemes` operation to list all the themes in the current AWS account.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-themes
    --aws-account-id AWSACCOUNTID
    --type QUICKSIGHT
    --page-size 10
    --max-items 100
```

For more information about the `ListThemes` operation, see ListThemes in the *Quick Sight API Reference*.

ListThemeVersions

Use the `ListThemeVersions` operation to list all the versions of the themes in the current AWS account. To use this operation to list the versions of a theme, you need that theme's ID. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-theme-version
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --page-size 10
    --max-items 100
```

To list all themes and their theme IDs, call the `ListThemes` operation.

For more information about the `ListThemeVersions` operation, see ListThemeVersions in the *Quick Sight API Reference*.

UpdateTheme

Use the `UpdateTheme` operation to update a theme.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-theme
    --aws-account-id 555555555555
    --theme-id THEMEID
    --base-theme-id BASETHEMEID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-theme
    --cli-input-json file//:updatetheme.json
```

For more information about the UpdateTheme operation, see UpdateTheme in the*Quick Sight API Reference*.

Theme alias operations

A *theme alias* is a reference to a version of a theme. For example, suppose that you create the theme alias *exampleAlias* for version 1 of the theme exampleTheme. You can use the theme alias *exampleAlias* to reference version 1 of theme exampleTheme in a DescribeTheme API operation, as in the following example.

Example

```
aws quicksight describe-theme
    --aws-account-id AWSACCOUNTID
    --theme-id exampleThemeID
    --alias-name exampleAlias
```

With theme alias operations, you can perform actions on Quick Sight theme aliases. For more information, see the following API operations.

Topics

- CreateThemeAlias
- DeleteThemeAlias
- DescribeThemeAlias
- ListThemeAliases
- UpdateThemeAlias

CreateThemeAlias

The CreateThemeAlias operation creates a theme alias for a theme. To use this operation, you need the ID of the theme that you want to create an alias for. You can use the ListThemes operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight
```

```
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --alias-name ALIAS
    --theme-version-number VERSION
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight create-theme-alias
    --cli-input-json file://create-theme-alias.json
```

For more information about the `CreateThemeAlias` operation, see CreateThemeAlias in the *Quick Sight API Reference*.

DeleteThemeAlias

Use the `DeleteThemeAlias` operation to delete the version of the theme that the specified theme alias points to. If you provide a specific alias, you delete the version of the theme that the alias points to. To use this operation, you need the ID of the theme that is using the alias that you want to delete. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-theme-alias
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --alias-name ALIAS
```

For more information about the `DeleteThemeAlias` operation, see DeleteThemeAlias in the *Quick Sight API Reference*.

DescribeThemeAlias

Use the `DescribeThemeAlias` operation to describe the alias for a theme. To use this operation, you need the ID of the theme that is using the alias that you want to describe. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-theme-alias
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --alias-name ALIAS
```

For more information about the `DescribeThemeAlias` operation, see DescribeThemeAlias in the *Quick Sight API Reference*.

ListThemeAliases

Use the `ListThemeAliases` operation to list all the aliases of a theme. To use this operation, you need the ID of the theme that is using the aliases that you want described. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-theme-aliases
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --max-results 100
```

For more information about the `ListThemeAliases` operation, see ListThemeAliases in the *Quick Sight API Reference*.

UpdateThemeAlias

Use the `UpdateThemeAlias` operation to update an alias of a theme. To use this operation, you need the ID of the theme that is using the alias that you want to update. You can use the `ListThemes` operation to list all themes and their corresponding theme IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-theme-alias
    --aws-account-id AWSACCOUNTID
    --theme-id THEMEID
    --alias-name ALIAS
    --theme-version-number VERSION
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-theme-alias
    --cli-input-json file://updatethemealias.json
```

For more information about the `UpdateThemeAlias` operation, see UpdateThemeAlias in the *Quick Sight API Reference*.

User operations

With user API operations, you can perform actions on Amazon Quick Sight account users. For more information, see the following API operations.

Topics

- DeleteUser
- DeleteUserByPrincipalTitle
- DescribeUser
- ListUserGroups
- ListUsers
- RegisterUser
- UpdateUser

DeleteUser

Use the `DeleteUser` operation to delete the Quick Sight user that is associated with the identity of the IAM user or role that's making the call. The IAM user isn't deleted as a result of this call.

To use this operation, you need the ID of the user that you want to delete. You can also use the `ListUsers` operation to list all users and their corresponding user IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-user
    --user-name USERNAME
    --aws-account-id AWSACCOUNTID
    --namespace default
```

For more information about the `DeleteUser` operation, see DeleteUser in the *Quick Sight API Reference*.

DeleteUserByPrincipalTitle

The `DeleteUserByPrincipalTitle` operation deletes a user identified by a principal ID. Following is an example AWS CLI command for this operation. To use this operation, you need the ID of the user that you want to delete. You can also use the `ListUsers` operation to list all users and their corresponding user IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight delete-user-by-principal-id
    --principal-id PRINCIPALID
    --aws-account-id AWSACCOUNTID
    --namespace default
```

For more information about the `DeleteUserByPrincipalTitle` operation, see DeleteUserByPrincipalTitle in the *Quick Sight API Reference*.

DescribeUser

Use the `DescribeUser` operation to return information about a user, given the user name. Following is an example AWS CLI command for this operation. To use this operation, you need the

ID of the user that you want to describe. You can also use the `ListUsers` operation to list all users and their corresponding user IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight describe-user
    --aws-account-id AWSACCOUNTID
    --namespace default
```

For more information about the `DescribeUser` operation, see DescribeUser in the *Quick Sight API Reference*.

ListUserGroups

Use the `ListUserGroups` operation to list the Quick Sight groups that an Quick Sight user is a member of. To use this operation, you need the ID of the user whose group memberships you want to know. You can use the `ListUsers` operation to list all users and their corresponding user IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-user-groups
    --user-name USERNAME
    --aws-account-id AWSACCOUNTID
    --namespace default
    --max-results 100
```

For more information about `ListUserGroups` operation, see ListUserGroupsin the *Quick Sight API Reference*.

ListUsers

Use the `ListUsers` operation to return a list of all of the Quick Sight users belonging to this account.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight list-users
    --aws-account-id AWSACCOUNTID
    --max-results 100
    --namespace default
```

For more information about `ListUsers` operation, see ListUsers in the *Quick Sight API Reference*.

RegisterUser

Use the `RegisterUser` operation to create an Quick Sight user whose identity is associated with the IAM identity or role specified in the request. When you register a new user from the Amazon Quick Sight API, Amazon Quick Sight generates a registration URL. The user accesses this registration URL to create their account. Amazon Quick Sight does not send a registration email to users who are registered from the Amazon Quick Sight API.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight register-user
    --identity-type QUICKSIGHT
    --email EMAIL
    --user-role AUTHOR
    --iam-arn 222233334444
    --session-name SESSION
    --aws-account-id AWSACCOUNTID
    --namespace default
    --user-name USERNAME
    --external-login-federation-provider-type CUSTOM_OIDC
    --custom-federation-provider-url www.example.com/
    --external-login-id EXTERNALLOGINID
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight register-user
    --cli-input-json file://registeruser.json
```

After using this operation, you get a response that includes a link labeled **Invitation URL**. Click the **Invitation URL** to set up a password and activate the new account. The new user then appear in the Quick Sight UI. You can use the `ListUsers` operation to list all users and their corresponding user IDs.

For more information about `RegisterUser` operation, see RegisterUserin the *Quick Sight API Reference*.

UpdateUser

Use the `UpdateUser` operation to update an Quick Sight user. To use this operation, you need the ID of the user that you want to delete. You can use the `ListUsers` operation to list all users and their corresponding user IDs.

Following is an example AWS CLI command for this operation.

AWS CLI

```
aws quicksight update-user
    --aws-account-id 555555555555
    --username USERNAME
    --namespace NAMESPACE
    --email johndoe@example.com
    --role ROLE
```

You can also make this command using a CLI skeleton file with the following command. For more information about CLI skeleton files, see Use CLI skeleton files.

```
aws quicksight update-user
    --cli-input-json file://updateuser.json
```

For more information about `UpdateUser` operation, see UpdateUserin the *Quick Sight API Reference*.

Document history for the Amazon Quick Sight Developer Guide

The following table describes important changes in each *Quick Sight Developer Guide* release.

Change	Description	Date
Initial release	Initial release of the *Amazon Quick Sight Developer Guide*	January 10, 2022