aws

User Guide

# AWS Resource Explorer

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

# AWS Resource Explorer: User Guide

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

# Table of Contents

**Resource Explorer ..... 1**
First time user ..... 2
Features of Resource Explorer ..... 2
Supported Regions ..... 3
Related services ..... 7
Pricing ..... 8
**Getting started ..... 9**
Accessing Resource Explorer ..... 9
Terms and concepts ..... 10
Resource Explorer administrator ..... 13
Resource Explorer user ..... 14
Index ..... 14
View ..... 16
Resource ..... 17
Unified search in the AWS Management Console ..... 18
Multi-account search ..... 19
Prerequisites ..... 19
Sign up for an AWS account ..... 19
Create a user with administrative access ..... 20
Setting up Resource Explorer ..... 21
Quick setup ..... 22
Advanced setup ..... 23
**Identify Resource Explorer status in AWS Regions ..... 29**
Checking the Resource Explorer status in a Region ..... 29
**Turning on a Region ..... 31**
Create a Resource Explorer index in a Region ..... 32
About opt-in Regions ..... 34
Opt-out behaviors ..... 34
**Turning on cross-Region search ..... 36**
About the aggregator index ..... 36
Creating the aggregator index ..... 38
Demoting the aggregator index ..... 39
**Turning on multi-account search ..... 42**
Prerequisites ..... 42

Enable multi-account search ... 42
Multi-account Quick Setup ... 43
Effect of account actions on multi-account search ... 44
Resource Explorer disabled ... 44
Member account is removed from an organization ... 44
Account is suspended ... 44
Account is closed ... 45
Account opt-out ... 45
**Supporting console unified search ... 46**
**Deploying to an organization ... 47**
Prerequisites ... 47
Creating the stack sets for Resource Explorer ... 48
Sample AWS CloudFormation templates ... 49
**Configuring Resource Explorer views ... 53**
Default views ... 55
Creating views ... 56
Granting access to views ... 60
Using tag-based authorization to control access to your views ... 61
Setting a default view ... 63
Tagging views ... 64
Add tags to your views ... 64
Controlling permissions with tags ... 66
Referencing tags in an ABAC policy ... 66
Sharing views ... 67
Permissions policy to share view with AWS accounts ... 68
Deleting views ... 70
**AWS managed views ... 72**
About managed views ... 72
Listing managed views ... 73
Deleting managed views ... 75
**Supported resource types ... 76**
Supported services and resource types ... 77
Amazon API Gateway ... 82
AWS Amplify ... 82
AWS App Runner ... 82
AWS AppConfig ... 82

Amazon AppFlow ... 83
AppIntegrations ... 83
AWS App Mesh ... 83
Amazon AppStream ... 83
AWS AppSync ... 83
AWS Application Discovery Service ... 83
Amazon Application Recovery Controller (ARC) ... 83
Amazon Athena ... 84
AWS Audit Manager ... 84
AWS Backup ... 84
AWS Batch ... 84
AWS Certificate Manager ... 84
AWS Cloud Map ... 84
AWS Cloud9 ... 85
AWS CloudFormation ... 85
Amazon CloudFront ... 85
AWS CloudTrail ... 85
Amazon CloudWatch ... 85
Amazon CloudWatch Evidently ... 86
Amazon CloudWatch Logs ... 86
Amazon CloudWatch Observability Access Manager ... 86
Amazon CloudWatch RUM ... 86
Amazon CloudWatch Synthetics ... 86
AWS CodeArtifact ... 86
AWS CodeBuild ... 87
AWS CodeCommit ... 87
AWS CodeConnections ... 87
AWS CodeDeploy ... 87
Amazon CodeGuru Profiler ... 87
Amazon CodeGuru Reviewer ... 87
AWS CodePipeline ... 87
AWS CodeStar Connections ... 87
Amazon Cognito Identity ... 88
Amazon Cognito IdentityPool ... 88
Amazon Comprehend ... 88
Amazon Connect ... 88

Amazon Connect Wisdom ... 88
AWS Cost Explorer ... 88
AWS Data Exchange ... 89
AWS Data Pipeline ... 89
AWS DataSync ... 89
AWS Database Migration Service ... 89
Amazon Detective ... 89
AWS Device Farm ... 89
Amazon DynamoDB ... 89
DynamoDB Accelerator ... 90
EC2 Image Builder ... 90
Amazon EMR ... 90
Amazon EMR Serverless ... 90
Amazon EMR on EKS ... 90
Amazon ElastiCache ... 90
AWS Elastic Beanstalk ... 91
Amazon Elastic Compute Cloud (Amazon EC2) ... 91
Amazon Elastic Container Registry ... 93
Amazon Elastic Container Registry Public ... 93
Amazon Elastic Container Service ... 93
Amazon Elastic File System ... 94
Amazon Elastic Kubernetes Service (Amazon EKS) ... 94
Elastic Load Balancing ... 94
AWS Elemental MediaPackage ... 94
AWS Elemental MediaPackage VoD ... 95
AWS Elemental MediaTailor ... 95
Amazon CloudWatch Events ... 95
Amazon EventBridge Pipes ... 95
Amazon EventBridge Scheduler ... 95
Amazon EventBridge Schemas ... 95
Amazon FSx ... 95
AWS Fault Injection Service ... 96
Amazon FinSpace ... 96
Firehose ... 96
Amazon Forecast ... 96
Amazon Fraud Detector ... 96

Amazon GameLift Servers ... 97
AWS Global Accelerator ... 97
AWS Glue ... 97
AWS Glue DataBrew ... 97
AWS Ground Station ... 98
Amazon GuardDuty ... 98
AWS HealthLake ... 98
AWS HealthOmics ... 98
IAM Access Analyzer ... 98
Amazon IVS ... 98
AWS Identity and Access Management ... 99
Amazon Inspector ... 99
Amazon Interactive Video Service ... 99
AWS IoT ... 99
AWS IoT Analytics ... 100
AWS IoT Core Device Advisor ... 100
AWS IoT Events ... 100
AWS IoT FleetWise ... 100
AWS IoT Greengrass ... 100
AWS IoT SiteWise ... 101
AWS IoT TwinMaker ... 101
AWS IoT Wireless ... 101
Amazon Kendra ... 101
AWS Key Management Service ... 102
Amazon Kinesis ... 102
Amazon Managed Service for Apache Flink ... 102
Amazon Kinesis Video Streams ... 102
AWS Lambda ... 102
Amazon Lex ... 102
Amazon Location Service ... 102
Amazon Lookout for Metrics ... 103
Amazon Lookout for Vision ... 103
Amazon MQ ... 103
AWS Mainframe Modernization ... 103
Amazon Managed Blockchain ... 103
Amazon Managed Grafana ... 103

Amazon Managed Service for Prometheus ..... 103
Amazon Managed Streaming for Apache Kafka ..... 103
Amazon Managed Workflows for Apache Airflow ..... 104
Amazon MemoryDB ..... 104
AWS Migration Hub Refactor Spaces ..... 104
AWS Mobile Targeting ..... 104
AWS Network Firewall ..... 104
AWS Network Manager ..... 105
Amazon OpenSearch Service ..... 105
AWS Outposts ..... 105
AWS Panorama ..... 105
Amazon Personalize ..... 105
AWS Private Certificate Authority ..... 105
AWS Proton ..... 106
Amazon Quantum Ledger Database (Amazon QLDB) ..... 106
Amazon QuickSight ..... 106
Amazon Redshift ..... 106
Amazon Rekognition ..... 106
Amazon Relational Database Service (Amazon RDS) ..... 107
AWS Resilience Hub ..... 107
AWS Resource Access Manager ..... 107
AWS Resource Groups ..... 107
AWS Resource Explorer ..... 108
Amazon Route 53 ..... 108
Amazon Route 53 Recovery Readiness ..... 108
Amazon Route 53 Resolver ..... 108
Amazon S3 Glacier ..... 108
Amazon SageMaker AI ..... 108
AWS Secrets Manager ..... 109
AWS Service Catalog ..... 109
AWS Signer ..... 109
Amazon Simple Email Service ..... 109
Amazon Simple Notification Service ..... 109
Amazon Simple Queue Service ..... 110
Amazon Simple Storage Service (Amazon S3) ..... 110
AWS Step Functions States Language ..... 110

Storage Gateway ... 110
AWS Systems Manager ... 110
AWS Transfer Family ... 110
Amazon WorkSpaces ... 111
Programmatically accessing the list of supported resource types ... 111
Resource types that appear as other types ... 112
**Searching for resources ... 114**
Quick filters ... 117
Search query templates ... 118
**Search query syntax ... 119**
How queries work in Resource Explorer ... 119
Query string syntax ... 119
Basics ... 119
Filters ... 120
Filter operators ... 125
**Example queries ... 129**
Untagged resources ... 129
Tagged resources ... 130
Missing tags ... 130
Invalid tags ... 130
Subset of Regions ... 131
Global resources ... 131
Multiple filters ... 131
Using quotation marks for multi-word terms ... 132
AWS CloudFormation stack members ... 132
**Viewing resource details ... 133**
Overview ... 133
Relationships ... 134
Timeline ... 134
Compliance ... 135
Resource shares ... 135
Tags ... 136
Additional properties ... 136
**Managing resources ... 137**
Resource Explorer console integrations with other AWS services ... 137
Console resource Actions ... 137

Manage tags ... 137
Create application ... 139
Add to application ... 141
Export resources to a .csv file ... 142
**Unified search ... 144**
Checking if unified search is enabled ... 144
Turning on unified search ... 145
**Working with CloudFormation ... 146**
Resource Explorer and CloudFormation templates ... 146
Learn more about AWS CloudFormation ... 149
**Using Amazon Q Developer in chat applications ... 150**
AWS resource questions ... 150
Prerequisites ... 150
Commonly asked resource questions ... 150
**Turning off Resource Explorer ... 152**
Turning off Resource Explorer in one AWS Region ... 152
Turning off all AWS Regions ... 154
**Security ... 157**
Upgrade IAM policies to IPv6 ... 158
Customers impacted by upgrade from IPv4 to IPv6 ... 158
What is IPv6? ... 158
Updating an IAM policy for IPv6 ... 159
Verify your client can support IPv6 ... 160
Identity and access management ... 161
Audience ... 162
Authenticating with identities ... 163
Managing access using policies ... 166
Resource Explorer and IAM ... 168
Identity-based policy examples ... 175
Example SCPs ... 180
AWS managed policies ... 182
Using service-linked roles ... 216
Troubleshooting permissions ... 218
Data protection ... 219
Encryption at rest ... 220
Encryption in transit ... 221

Compliance validation ... 221
Resilience ... 222
Infrastructure security ... 222
AWS PrivateLink ... 223
Considerations ... 223
Create an interface endpoint ... 223
Create an endpoint policy ... 224
**Monitoring ... 225**
CloudTrail logs ... 225
Resource Explorer information in CloudTrail ... 225
Understanding Resource Explorer log file entries ... 227
**Troubleshooting ... 237**
General issues ... 237
A link to Resource Explorer is missing the AWS Region ... 237
Unified search CloudTrail errors ... 238
Setup issues ... 239
I get an "access denied" message when I make a request to Resource Explorer ... 239
I get an "access denied" message when I make a request with temporary security credentials ... 240
Search issues ... 241
Why are some resources missing from my Resource Explorer search results? ... 241
Why are some searches limited to 1,000 results? ... 243
Why are my resources not appearing in unified search results in the console? ... 243
Why do unified search in the console and Resource Explorer sometimes give different results? ... 244
What permissions do I need to be able to search for resources? ... 244
**Quotas ... 246**
**Working with AWS SDKs ... 247**
**Document history ... 249**

# What is AWS Resource Explorer?

AWS Resource Explorer is a resource search and discovery service. With Resource Explorer, you can explore your resources, such as Amazon Elastic Compute Cloud instances, Amazon Kinesis streams, or Amazon DynamoDB tables, using an internet search engine-like experience. Resource Explorer allows you to easily search for your resources using resource metadata like names, tags, and IDs, and displays additional resource details from other AWS services, such as AWS Config and AWS Cloud Control. You can add resource metadata using tags, and collectively manage resources in an application. Resource Explorer works across AWS Regions in your account to simplify your cross-Region workloads.

Resource Explorer provides fast responses to your search queries by using indexes that are created and maintained by the AWS Resource Explorer service. Resource Explorer uses a variety of data sources to gather information about resources in your AWS account. Resource Explorer stores that information in the indexes for Resource Explorer to search.

**We want your feedback about this documentation**

Our goal is to help you get everything you can from Resource Explorer. If this guide helps you to do that, then let us know. If the guide isn't helping you, then we want to hear from you so we can address the issue. Use the **Feedback** link that's in the upper-right corner of every page. That sends your comments directly to the writers of this guide. We review every submission, looking for opportunities to improve the documentation. Thank you in advance for your help!

**Topics**


- Are you a first-time Resource Explorer user?
- Features of Resource Explorer
- Resource Explorer supported Regions
- Related AWS services
- Pricing

# Are you a first-time Resource Explorer user?

If you're a first-time user of Resource Explorer, we recommend that you begin by reading the following topics in the *Getting started* section:

- Terms and concepts for Resource Explorer
- Setting up Resource Explorer using Quick setup

# Features of Resource Explorer

Resource Explorer provides the following features:

- Users can ***search for resources*** in their AWS Region or across Regions in their AWS account.
- Users can use keywords, search operators, and attributes like tags to ***filter the search results to only matching resources***.
- When users find a resource in the search results, they can immediately ***go to the resource's native console*** to work with that resource.
- Users can view additional information about a selected resource from other AWS services, such as AWS Config and AWS Cloud Control, directly in the Resource Explorer console.
- Users can manage their resources using quick Actions in the Resource Explorer console to manage tags and add resources to new or existing applications.
- Administrators can ***create views that define which resources are available in search results***. Administrators can create different views for different groups of users based on their tasks, and grant permissions to views to only those users who need them.
- Resource Explorer, like many other AWS services, is ***eventually consistent***. Resource Explorer achieves high availability by replicating data across multiple servers within Amazon data centers around the world. If a request to change some data is successful, the change is committed and safely stored. However, then the change must be replicated across Resource Explorer, which can take some time. As an example, this includes Resource Explorer finding a resource in one Region, and replicating that to the Region that contains the aggregator index for the account.

# Resource Explorer supported Regions

| Region Name | Region | Endpoint | Protocol |
|---|---|---|---|
| US East (Ohio) | us-east-2 | resource-explorer-2.us-east-2.amazonaws.com<br>resource-explorer-2-fips.us-east-2.amazonaws.com<br>resource-explorer-2-fips.us-east-2.api.aws | HTTPS<br>HTTPS<br>HTTPS |
| US East (N. Virginia) | us-east-1 | resource-explorer-2.us-east-1.amazonaws.com<br>resource-explorer-2-fips.us-east-1.amazonaws.com<br>resource-explorer-2-fips.us-east-1.api.aws | HTTPS<br>HTTPS<br>HTTPS |
| US West (N. California) | us-west-1 | resource-explorer-2.us-west-1.amazonaws.com<br>resource-explorer-2-fips.us-west-1.amazonaws.com<br>resource-explorer-2-fips.us-west-1.api.aws | HTTPS<br>HTTPS<br>HTTPS |
| US West (Oregon) | us-west-2 | resource-explorer-2.us-west-2.amazonaws.com<br>resource-explorer-2-fips.us-west-2.amazonaws.com<br>resource-explorer-2-fips.us-west-2.api.aws | HTTPS<br>HTTPS<br>HTTPS |
| Africa (Cape Town) | af-south-1 | resource-explorer-2.af-south-1.amazonaws.com | HTTPS |

| Region Name | Region | Endpoint | Protocol |
|---|---|---|---|
| Asia Pacific (Hong Kong) | ap-east-1 | resource-explorer-2.ap-east-1.amazonaws.com | HTTPS |
| Asia Pacific (Hyderabad) | ap-south-2 | resource-explorer-2.ap-south-2.amazonaws.com | HTTPS |
| Asia Pacific (Jakarta) | ap-southeast-3 | resource-explorer-2.ap-southeast-3.amazonaws.com | HTTPS |
| Asia Pacific (Malaysia) | ap-southeast-5 | resource-explorer-2.ap-southeast-5.amazonaws.com | HTTPS |
| Asia Pacific (Melbourne) | ap-southeast-4 | resource-explorer-2.ap-southeast-4.amazonaws.com | HTTPS |
| Asia Pacific (Mumbai) | ap-south-1 | resource-explorer-2.ap-south-1.amazonaws.com | HTTPS |
| Asia Pacific (Osaka) | ap-northeast-3 | resource-explorer-2.ap-northeast-3.amazonaws.com | HTTPS |
| Asia Pacific (Seoul) | ap-northeast-2 | resource-explorer-2.ap-northeast-2.amazonaws.com | HTTPS |

| Region Name | Region | Endpoint | Protocol |
|---|---|---|---|
| Asia Pacific (Singapore) | ap-southeast-1 | resource-explorer-2.ap-southeast-1.amazonaws.com | HTTPS |
| Asia Pacific (Sydney) | ap-southeast-2 | resource-explorer-2.ap-southeast-2.amazonaws.com | HTTPS |
| Asia Pacific (Thailand) | ap-southeast-7 | resource-explorer-2.ap-southeast-7.amazonaws.com | HTTPS |
| Asia Pacific (Tokyo) | ap-northeast-1 | resource-explorer-2.ap-northeast-1.amazonaws.com | HTTPS |
| Canada (Central) | ca-central-1 | resource-explorer-2.ca-central-1.amazonaws.com<br>resource-explorer-2-fips.ca-central-1.amazonaws.com<br>resource-explorer-2-fips.ca-central-1.api.aws | HTTPS<br>HTTPS<br>HTTPS |
| Canada West (Calgary) | ca-west-1 | resource-explorer-2.ca-west-1.amazonaws.com<br>resource-explorer-2-fips.ca-west-1.amazonaws.com<br>resource-explorer-2-fips.ca-west-1.api.aws | HTTPS<br>HTTPS<br>HTTPS |

| Region Name | Region | Endpoint | Protocol |
|---|---|---|---|
| Europe (Frankfurt) | eu-central-1 | resource-explorer-2.eu-central-1.amazonaws.com | HTTPS |
| Europe (Ireland) | eu-west-1 | resource-explorer-2.eu-west-1.amazonaws.com | HTTPS |
| Europe (London) | eu-west-2 | resource-explorer-2.eu-west-2.amazonaws.com | HTTPS |
| Europe (Milan) | eu-south-1 | resource-explorer-2.eu-south-1.amazonaws.com | HTTPS |
| Europe (Paris) | eu-west-3 | resource-explorer-2.eu-west-3.amazonaws.com | HTTPS |
| Europe (Spain) | eu-south-2 | resource-explorer-2.eu-south-2.amazonaws.com | HTTPS |
| Europe (Stockholm) | eu-north-1 | resource-explorer-2.eu-north-1.amazonaws.com | HTTPS |
| Europe (Zurich) | eu-central-2 | resource-explorer-2.eu-central-2.amazonaws.com | HTTPS |
| Israel (Tel Aviv) | il-central-1 | resource-explorer-2.il-central-1.amazonaws.com | HTTPS |
| Mexico (Central) | mx-central-1 | resource-explorer-2.mx-central-1.amazonaws.com | HTTPS |
| Middle East (Bahrain) | me-south-1 | resource-explorer-2.me-south-1.amazonaws.com | HTTPS |

| Region Name | Region | Endpoint | Protocol | |
|---|---|---|---|---|
| Middle East (UAE) | me-central-1 | resource-explorer-2.me-central-1.amazonaws.com | HTTPS | |
| South America (São Paulo) | sa-east-1 | resource-explorer-2.sa-east-1.amazonaws.com | HTTPS | |

## Related AWS services

The following are the other AWS services whose primary purpose is to help you manage your AWS resources:

myApplications in the AWS Management Console

myApplications is an extension of the AWS Management Console that helps you manage and monitor the cost, health, security posture, and performance of your applications in AWS. Applications allow you to group resources and metadata. From the AWS Management Console, you can access all of the applications in your account, view metrics across all applications, and review cost, security, and operations metrics from multiple AWS services in a single view. myApplications includes resource information from the following AWS services:

- AWS Resource Access Manager (AWS RAM)

  Share the resources in one AWS account with other AWS accounts. If your account is managed by AWS Organizations, you can use AWS RAM to share resources with the accounts in an organizational unit, or all of the accounts in the organization. The shared resources work for users in those accounts just like they would if they were created in the local account.

- AWS Resource Groups

  Create groups for your AWS resources. Then, you can use and manage each group as a unit instead of having to reference every resource individually. Your groups can consist of resources that are part of the same AWS CloudFormation stack, or that are tagged with the same tags.

Some resource types also support applying a configuration to a resource group to affect all relevant resources in that group.

- AWS Resource Groups Tagging API

  Tags are customer-defined metadata that you can attach to your resources. You can categorize your resources for purposes like cost allocation and attribute-based access control.

# Pricing

There are no charges to search for resources by using AWS Resource Explorer, including creating views, turning on Regions, or searching for resources. In the process of building your resource inventory, Resource Explorer calls APIs on your behalf that may result in charges. Interacting with the resources that you find in your search results can result in usage charges that vary depending on the resource type and its AWS service. Some sources of additional data available in the Resource Explorer console are from other AWS services that can result in usage charges, such as AWS Config. These sources are only used if you explicitly enable them in your account. For more information about how AWS bills for the normal use of a specific resource type, refer to the documentation for that resource type's owning service.

# Getting started with Resource Explorer

Use the topics in this section to get a basic understanding of the concepts and terms used by AWS Resource Explorer. Learn about the prerequisites that you must satisfy to successfully use Resource Explorer and how to turn on Resource Explorer in your AWS account.

## Accessing Resource Explorer

You can interact with Resource Explorer in the following ways:

### Resource Explorer console

Resource Explorer provides a web-based user interface, the Resource Explorer console. If you signed up for an AWS account, you can access the Resource Explorer console by signing into the AWS Management Console and choosing **Resource Explorer** from the console home page.

You can also navigate in your browser directly to the **Resource Explorer dashboard** page, or to the **Resource search** page. If you aren't already signed in, then you're asked to do so before the console appears.

> **Note**
>
> The Resource Explorer console is a *global* console, meaning that you don't have to select an AWS Region to work in. However, when you use Resource Explorer to create an index or a view, you need to specify which Region the index or view is stored in. When you use Resource Explorer to search, you can choose any view you have access to. The results automatically come from the Region associated with the selected view. If the view is from the Region that contains the aggregator index, the results include resources from all Regions where you created Resource Explorer indexes.

### AWS Management Console unified search

At the top of every page in the AWS Management Console, there is a search bar. You can configure Resource Explorer to participate in unified search. Then, your users can use Resource Explorer search query syntax in the unified search text box, and see matching resources in those search results. By turning this feature on, users can search for resources from the console of any AWS service without having to first switch to the Resource Explorer console.

> **Important**
>
> Unified search always searches using the default view in the AWS Region that contains the aggregator index.

**Resource Explorer commands in the AWS CLI and Tools for Windows PowerShell**

The AWS CLI and Tools for PowerShell provide direct access to the Resource Explorer public API operations. These tools work on Windows, macOS, and Linux. For more information about getting started, see the AWS Command Line Interface User Guide, or the AWS Tools for Windows PowerShell User Guide. For more information about the commands for Resource Explorer, see the AWS CLI Command Reference or the AWS Tools for Windows PowerShell Cmdlet Reference.

**Resource Explorer operations in the AWS SDKs**

AWS provides API commands for a broad set of programming languages. For more information about getting started, see Using AWS Resource Explorer with an AWS SDK.

**Query API**

If you don't use one of the supported programming languages, the Resource Explorer HTTPS Query API gives you programmatic access to Resource Explorer. With the Resource Explorer API, you can issue HTTPS requests directly to the service. When you use the Resource Explorer API, you must include code that can digitally sign your requests using your AWS credentials. For more information, see the AWS Resource Explorer API Reference.

# Terms and concepts for Resource Explorer

AWS Resource Explorer is a resource search and discovery service. With Resource Explorer, you can explore your resources by using an internet search engine-like experience. You can search for your resources, such as Amazon Elastic Compute Cloud instances, Amazon Kinesis streams, or Amazon DynamoDB tables by using resource metadata like names, tags, and IDs. Resource Explorer works across AWS Regions in your account to simplify your cross-Region workloads.

Resource Explorer provides fast responses to your search queries by using indexes that are created and maintained by the AWS Resource Explorer service. Resource Explorer uses a variety of data sources to gather information about resources in your AWS account. Resource Explorer stores that information in the indexes for Resource Explorer to search.

You should understand the following concepts to successfully administer and configure AWS Resource Explorer for your users.

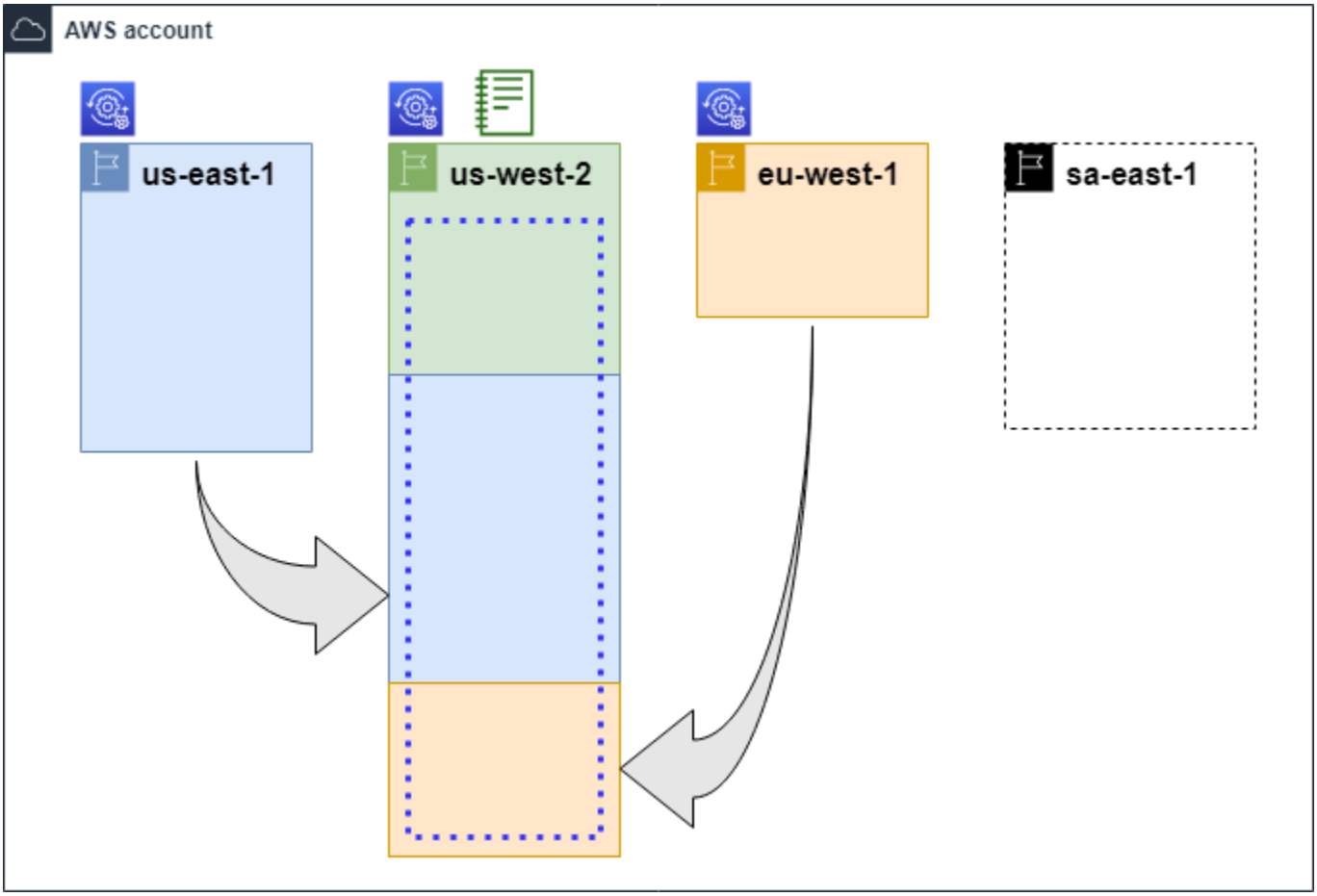
**Concepts**

- Resource Explorer administrator
- Resource Explorer user
- Index
- View
- Resource
- Unified search in the AWS Management Console
- Multi-account search

The following diagram shows three AWS Regions in which the administrator turned on Resource Explorer, and one Region the administrator chose not to turn on. The Region where Resource Explorer isn't turned on doesn't have an index. Therefore, its resources can't be searched by Resource Explorer queries.

In this example scenario, the administrator chose the US West (Oregon) Region (`us-west-2`) to contain the aggregator index for the account. All Regions that you turn on replicate their local indexes to the Region with the aggregator index.

The default view created by Resource Explorer doesn't have any filters. Therefore, results from searching with this view can include resources of any type in all Regions in the account where Resource Explorer is turned on.




| Legend | |
|---|---|
|  | Resource Explorer is turned on in this AWS Region and information about the Region's resources is stored in a local index in that Region. Every Region's local index is also replicated (indicated by the arrows) to the Region that contains the aggregator index. |
|  | The index in this AWS Region is configured to be the aggregator index for the account. Resource Explorer replicates the resource information collected in the local indexes of all other Regions where Resource Explorer is turned on into the aggregator index in this Region. Searches made in this Region can include results from all Regions in the account. |
|  | The default view created by **Quick Setup** includes all resources in all AWS Regions. |

# Resource Explorer administrator

A Resource Explorer *administrator* is an AWS Identity and Access Management (IAM) principal who has the permission to manage Resource Explorer and its settings in the AWS account. The Resource Explorer administrator can configure the following features:

- Turn on Resource Explorer for individual AWS Regions in the AWS account by creating indexes in those Regions. This lets Resource Explorer discover resources and populate the index with information about those resources so that users can search for resources in that Region.
- Update the index type in one AWS Region to make it the aggregator index for its AWS account. The aggregator index in this Region receives replicated copies of the resource information from all other Regions in the account where Resource Explorer is turned on.
- Create views that define the subset of indexed information users can search and discover in Resource Explorer.
- While not part of the Resource Explorer actions, the Resource Explorer administrator must also be able to grant search permissions to the principals in the account. The administrator can grant these permissions to principals by adding the relevant permissions to existing IAM permission policies, or by using the Resource Explorer read only AWS managed policy.

  To provide access, add permissions to your users, groups, or roles:

  - Users and groups in AWS IAM Identity Center:

    Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.

  - Users managed in IAM through an identity provider:

    Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

  - IAM users:

    - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
    - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

The administrator typically has all Resource Explorer permissions (`resource-explorer-2:*`) on all Resource Explorer resources, including the indexes and views. These permissions can be granted by using the Resource Explorer full access AWS managed policy.

## Resource Explorer user

A Resource Explorer *user* is an IAM principal that has permission to do one or more of the following tasks:

- Perform a search for resources by using a view to query Resource Explorer. A Resource Explorer user wants to discover and find AWS resources and typically uses the Resource Explorer console, or the Resource Explorer `Search` operations provided by the AWS SDKs or the AWS CLI.

  A role or user can use IAM get permission to search with one of two methods:

  - The Resource Explorer read only AWS managed policy to the IAM role, group, or user.
  - An IAM permission policy with a statement containing the following minimum permissions to the IAM role, group, or user.

    ```
    {
        "Effect": "Allow",
        "Action": [
            "resource-explorer-2:Search",
            "resource-explorer-2:GetView",
        "Resource": "<ARN of the view>"
    }
    ```

- Although typically considered an administrator task, you can delegate to trusted users the ability to define create views. To do this, the administrator can grant permission to call the `resource-explorer-2:CreateView` operation in an IAM permission policy attached to the relevant roles, groups, or users. If the view requires specific permissions, then provision for adding or modifying the IAM policies for the relevant users must be made.

For information about how to search for resources using Resource Explorer, see Using AWS Resource Explorer to search for resources.

## Index

An *index* is the collection of information maintained by Resource Explorer about all of the AWS resources in one AWS Region in your AWS account. Resource Explorer maintains an index in each

Region in which you turn on Resource Explorer. Resource Explorer updates the index automatically as you create and delete resources in your AWS account. In the earlier diagram, the boxes under the AWS Region names represent the Resource Explorer indexes maintained in each AWS Region. The index in a Region is the source of information for any views created in that Region. Users can't directly query the index. Instead, they must always query using a view.

There are two types of indexes:

**Local index**

There is one *local index* in every AWS Region in which you turn on Resource Explorer. A local index contains information about only the resources in the same Region.

**Aggregator index**

The Resource Explorer administrator can also designate the index in one AWS Region to be the *aggregator index* for the AWS account. The aggregator index receives and stores a copy of the index for every other Region where Resource Explorer is turned on in the account. The aggregator index also receives and stores information about the resources in its own Region. In the earlier diagram, the Region `us-west-2` contains the aggregator index for the account. The primary reason to designate an aggregator index for the account is so that you can create views that can include resources from all Regions in the account. There can be ***only one*** aggregator index in an AWS account.

When you turn on Resource Explorer, you can specify which AWS Region is to contain the aggregator index. You can also change the AWS Region used for the aggregator index later. For information about how to promote a local index to make it the aggregator index for its AWS account, see Turning on cross-Region search by creating an aggregator index.

An index is a resource with an Amazon resource name (ARN). However, you can use this ARN only in permission policies to grant access to operations that interact directly with the index. With those operations, you can create views and set them as the default in a Region, turn on or turn off Resource Explorer in a Region, and create an aggregator index for the account. The ARN of an index looks similar to the following example:

```
arn:aws:resource-explorer-2:us-east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-
abcd11111111
```

# View

A *view* is the mechanism used to query the resources listed in an index. The view defines what information in the index is visible and available for search and discovery purposes. A user never directly queries the Resource Explorer index. Instead, queries must always go through a view which lets the view creator limit which resources the user can see in search results.

When you create a view, you specify filters that restrict which resources are included in search results. For example, you could choose to include only resources of a few specified resource types that are used by those to whom you grant access to this view. Results from queries that users make with a view are always automatically filtered to include only those resources that match the view's criteria.

To grant access to use a view, you can use assign permissions using one of the following methods.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

Grant permission to allow your roles, groups, or users to invoke the `resource-explorer-2:GetView` and `resource-explorer-2:Search` operations on a view identified by its Amazon resource name (ARN). Alternatively, you can use the Resource Explorer read only AWS managed policy for all principals who need to use the view to search. You can create multiple views that have different filters and scopes and thus return different subsets of your resource information. Then, you can grant permissions for each view to those users who need to see the information included by that view's results.

To search with Resource Explorer, each user must have permission to use at least one view. You can't perform a search in Resource Explorer without using a view.

Views are stored on a per-Region basis. A view can access only the Resource Explorer index in that AWS Region. To access account-wide search results, you must use a view in the Region that contains the *aggregator index* for the account. The **Quick setup** option creates a default view in the AWS Region with the aggregator index and with filters that include all resources in all AWS Regions used by the account.

For information about how to create views, see Configuring an Resource Explorer view to provide access to resource searches. For information about how to use views in a query, see Using AWS Resource Explorer to search for resources.

Every view has an Amazon resource name (ARN) that you can reference in permission policies to grant access to individual views. You can also pass a view's ARN as a parameter to any API or AWS CLI operation that interacts with a view. The ARN of a view looks similar to the following example.

```
arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-View-
Name/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
```

**Note**

Every view ARN includes an AWS generated UUID at the end. This helps to ensure that users who might have had access to views with a specific name that was deleted can't automatically access a new view created with the same name.

## Resource

A *resource* is an entity in AWS that you can work with. Resources are created by AWS services as you use the features of the service. Examples include an Amazon EC2 instance, an Amazon S3 bucket, or an AWS CloudFormation stack. Some resource types can contain customer data. All resource types have attributes or metadata to describe the resource, including a name, description, and the Amazon resource name (ARN) that you use to uniquely reference a resource. Most resource types also support tags. Tags are custom metadata that you can attach to your resources for a variety of purposes, such as cost allocation in your billing, security authorization using attribute-based access control, or to support your other categorization needs.

The primary purpose of Resource Explorer is to help you find the resources that exist in your AWS account. Resource Explorer uses a variety of techniques to discover all of your resources and place information about them in an index. Then, you can query the index through whatever views that your administrator makes available to you.

> **Important**
>
> Resource Explorer excludes intentionally those resources types whose inclusion would expose customer data. The following resource types are ***not*** indexed by Resource Explorer and are therefore never returned in search results.
>
> - Amazon S3 objects that are contained *within* a bucket
> - Amazon DynamoDB table items
> - DynamoDB attribute values

## Unified search in the AWS Management Console

At the top of the AWS Management Console, in every AWS service, there is a search bar that you can use to search for a variety of AWS related things. You can search for services and features, and get links directly to the relevant page in that service's console. You can also search for documentation and blog articles related to your search term.

After you turn on Resource Explorer and create an aggregator index and a default view, unified search can also include your account's resources in the search results. *Unified search* automatically uses the default view in the AWS Region that contains the aggregator index for the account. This lets you search for a resource from any page in the AWS Management Console, without having to first open Resource Explorer. If you don't promote a local index to be the aggregator index for the account, or if you don't create a default view in the aggregator index Region, unified search doesn't include resources in its search results. Also, any principal performing a search must have permission to use the default view in the Region that contains the aggregator index or unified search doesn't include resources in its search results.

> **Important**
>
> Unified search automatically inserts a wildcard character (*) operator at the end of the first keyword in the string. This means that unified search results include resources that match any string that starts with the specified keyword.

The search performed by the **Query** text box on the Resource search page in the Resource Explorer console does ***not*** automatically append a wildcard character. You can insert a * manually after any term in the search string.

For more information about unified search and its integration with Resource Explorer, see Using unified search in the AWS Management Console.

### Multi-account search

With multi-account search, you can search and discover resources across AWS Organizations and AWS Regions with a single keyword search.

For more information about multi-account search and how to enable it for Resource Explorer, see Turning on multi-account search.

## Prerequisites to using Resource Explorer

Before you use AWS Resource Explorer for the first time, complete the following tasks as required.

**Tasks**

- Sign up for an AWS account
- Create a user with administrative access

### Sign up for an AWS account

If you do not have an AWS account, complete the following steps to create one.

**To sign up for an AWS account**

1. Open https://portal.aws.amazon.com/billing/signup.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call or text message and entering a verification code on the phone keypad.

   When you sign up for an AWS account, an *AWS account root user* is created. The root user has access to all AWS services and resources in the account. As a security best practice, assign

administrative access to a user, and use only the root user to perform tasks that require root user access.

AWS sends you a confirmation email after the sign-up process is complete. At any time, you can view your current account activity and manage your account by going to https://aws.amazon.com/ and choosing **My Account**.

## Create a user with administrative access

After you sign up for an AWS account, secure your AWS account root user, enable AWS IAM Identity Center, and create an administrative user so that you don't use the root user for everyday tasks.

### Secure your AWS account root user

1. Sign in to the AWS Management Console as the account owner by choosing **Root user** and entering your AWS account email address. On the next page, enter your password.

   For help signing in by using root user, see Signing in as the root user in the *AWS Sign-In User Guide*.

2. Turn on multi-factor authentication (MFA) for your root user.

   For instructions, see Enable a virtual MFA device for your AWS account root user (console) in the *IAM User Guide*.

### Create a user with administrative access

1. Enable IAM Identity Center.

   For instructions, see Enabling AWS IAM Identity Center in the *AWS IAM Identity Center User Guide*.

2. In IAM Identity Center, grant administrative access to a user.

   For a tutorial about using the IAM Identity Center directory as your identity source, see Configure user access with the default IAM Identity Center directory in the *AWS IAM Identity Center User Guide*.

**Sign in as the user with administrative access**

- To sign in with your IAM Identity Center user, use the sign-in URL that was sent to your email address when you created the IAM Identity Center user.

  For help signing in using an IAM Identity Center user, see Signing in to the AWS access portal in the *AWS Sign-In User Guide*.

**Assign access to additional users**

1. In IAM Identity Center, create a permission set that follows the best practice of applying least-privilege permissions.

   For instructions, see Create a permission set in the *AWS IAM Identity Center User Guide*.
2. Assign users to a group, and then assign single sign-on access to the group.

   For instructions, see Add groups in the *AWS IAM Identity Center User Guide*.

# Setting up and configuring Resource Explorer

Before you can set up and configure AWS Resource Explorer, first ensure that you meet the prerequisites. After that, sign in as an IAM role or user that has the permissions required to perform the Resource Explorer operations for the following procedure.

You can use this set up and configuration procedure to set up Resource Explorer in existing accounts, and in any new accounts added to your organization.

There are two ways to set up Resource Explorer:

- **Quick setup**
- **Advanced setup**

> **Important**
>
> If you choose to set up Resource Explorer using any option that says "all AWS Regions", it activates only those AWS Regions that exist and that are enabled in the AWS account *at the time you perform the procedure*. Resource Explorer does ***not*** automatically turn on in any AWS Regions that AWS adds in the future. When AWS introduces a new Region, you

can choose to turn on Resource Explorer in the Region manually when it appears in the **Settings** page of the Resource Explorer console, or by calling the CreateIndex operation.

> **Note**
>
> Setting up Resource Explorer can also turn on the ability to search for resources by using the unified search bar on the AWS Management Console. For users to see resources in the unified search results, you must configure Resource Explorer with a cross-Region aggregator index and a default view. For details, see the following procedures. You must also ensure that your searching users have permission to use the default view in the AWS Region that contains the aggregator index. For more information, see Using unified search in the AWS Management Console.

## Setting up Resource Explorer using Quick setup

If you choose the Quick setup option, Resource Explorer does the following:

- Creates an index in every AWS Region in your AWS account.
- Updates the index in the Region you specify to be the aggregator index for the account.
- Creates a default view in the aggregator index Region. This view has no filters so it returns all resources found in the index.

### Minimum permissions

To perform the steps in the following procedure, you must have the following permissions:

- **Action**: `resource-explorer-2:*` – **Resource**: no specific resource (*)
- **Action**: `iam:CreateServiceLinkedRole` – **Resource**: no specific resource (*)

AWS Management Console

**To set up Resource Explorer using Quick setup**

1. Open the AWS Resource Explorer console at https://console.aws.amazon.com/resource-explorer.

2. Choose **Turn on Resource Explorer**.
3. On the **Turn on Resource Explorer** page, choose **Quick setup**.
4. Choose which AWS Region you want to contain the aggregator index. You should select the Region that is appropriate for the geographic location for your users.
5. At the bottom of the page, choose **Turn on Resource Explorer**.
6. On the **Progress** page, you can monitor each AWS Region as Resource Explorer creates its index. The page displays the status of creating the aggregator index and creating the default view.

After all steps show that they completed successfully, you and your users can navigate to the **Resource search** page and begin searching for resources.

> **Note**
>
> Tagged resources local to the index appear in search results within a few minutes. Untagged resources typically take less than two hours to appear, but can take longer when there is heavy demand. It can also can take up to an hour to complete the initial replication to a new aggregator index from all of the existing local indexes.

**Next steps:** Before your users can search with the default view you just created, you must grant them permissions to search with it. For more information, see Granting access to Resource Explorer views for search.

AWS CLI

Setting up Resource Explorer in your AWS account by using the AWS CLI is, by definition, equivalent to the **Advanced setup** option. This is because the Resource Explorer CLI operations don't perform any of the steps for you automatically like the Resource Explorer console does. See the AWS CLI tab on the Setting up Resource Explorer using Advanced setup to see what commands are the equivalent of using the console.

## Setting up Resource Explorer using Advanced setup

If you choose the Advanced setup option, you can do the following:

- Choose the AWS Regions in which to turn on Resource Explorer.

- Choose whether to configure one Region with an aggregator index. If you do, you specify the AWS Region to place it in. This index allows you to create views that can include resources from all Regions in the account. For more information, see Turning on cross-Region search by creating an aggregator index.
- Choose whether to create a default view. That view allows searching automatically for any AWS resource in the Regions in which you turn on Resource Explorer. You must ensure that any principals who need to use the default view to search in Resource Explorer have permissions on the view. For more information, see Granting access to Resource Explorer views for search.

> **Note**
>
> You can configure Resource Explorer to include your resources in the search results provided by the unified search feature on the AWS Management Console. To turn on this feature, you must configure Resource Explorer with an aggregator index and a default view that all roles and users can search with. The Quick setup option creates both the aggregator index and default view and is the way we recommend that you turn on Resource Explorer.

### Minimum permissions

To perform the steps in the following procedure, you must have the following permissions:

- **Action**: `resource-explorer-2:*` – **Resource**: no specific resource (*)
- **Action**: `iam:CreateServiceLinkedRole` – **Resource**: no specific resource (*)

AWS Management Console

#### To turn on Resource Explorer using Advanced setup

1. Open the AWS Resource Explorer console at https://console.aws.amazon.com/resource-explorer.
2. Choose **Turn on Resource Explorer**.
3. On the **Turn on Resource Explorer** page, choose **Advanced setup**.
4. In the **AWS Regions** box, under **Regions**, choose whether you want to turn on Resource Explorer in all AWS Regions, or only specific Regions.

If you choose **Turn on Resource Explorer in only the specified AWS Regions in this account**, select each Region whose resources you want to include in search results.

5. For **Aggregator index**, choose whether you want to create an aggregator index. If you choose to create an aggregator index, all other AWS Regions replicate their indexes to this Region. This lets users search for resources across all selected Regions in the AWS account. Choose the AWS Region that contains the aggregator index. We recommend that you specify the Region where your users spend most of their time, or at least where you expect them to perform most of their resource searches.
6. In the **Default view** box, under **View creation**, choose whether to create a default view. This option is available only if you chose to create an aggregator index. If you choose to create a default view, Resource Explorer places this view in the same AWS Region as the aggregator index. This lets the default view include results from all AWS Regions in which you registered Resource Explorer. Whenever a user performs a search in a Region with a default view and doesn't explicitly specify a view, the search uses the default view for that Region.

> **Note**
>
> Before your users can search with a view, you must grant them permissions to use that view. For more information, see Granting access to Resource Explorer views for search.

7. Choose **Activate Resource Explorer**.

> **Note**
>
> Tagged resources local to the index appear in search results within a few minutes. Untagged resources typically take less than two hours to appear, but can take longer when there is heavy demand. It can also can take up to an hour to complete the initial replication to a new aggregator index from all of the existing local indexes.

AWS CLI

**To set up Resource Explorer using Advanced setup**

The Resource Explorer console performs many API operation calls on your behalf based on the choices you make. The following example AWS CLI commands illustrate how to perform the same basic procedures outside of the console using the AWS CLI.

**Example Step 1: Turn on Resource Explorer by creating indexes in the desired AWS Regions**

Run the following command in each AWS Region in which you want to activate Resource Explorer. The following example command turns on Resource Explorer in the AWS Region that is the default for the AWS CLI.

```
$ aws resource-explorer-2 create-index
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-07-27T16:17:12.130000+00:00",
    "State": "CREATING"
}
```

**Example Step 2: Update the index in one AWS Region to be the aggregator index for the account**

Run the following command in the AWS Region in which you want Resource Explorer to update the local index to the aggregator index for the account. The following example command updates the aggregator index in the US East (N. Virginia) (`us-east-1`).

```
$ aws resource-explorer-2 update-index-type \
    --arn arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111 \
    --type AGGREGATOR
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "LastUpdatedAt": "2022-07-27T16:29:49.231000+00:00",
    "State": "UPDATING",
    "Type": "AGGREGATOR"
}
```

**Example Step 3: Create a view in the AWS Region that contains the aggregator index**

Run the following command in the AWS Region in which you created the aggregator index. The following example command creates a view identical to the one created by the Resource

Explorer console setup process. This new view includes tags attached to the resource as part of the indexed information and supports searching for resources by tag key or value.

```
$ aws resource-explorer-2 create-view \
    --view-name My-New-View \
    --included-properties Name=tags
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-07-27T16:34:14.960000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-New-
View/1a2b3c4d-5d6e-7f8a-9b0c-abcd22222222"
    }
}
```

**Example Step 4: Set your new view as the default for its AWS Region**

The following example sets the view you created in the previous step as the default for the Region. You must run the following command in the same AWS Region in which you created the default view.

```
$ aws resource-explorer-2 associate-default-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-New-
View/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-New-
View/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
}
```

Before your users can search with a view, you must grant them permissions to use that view. For more information, see Granting access to Resource Explorer views for search.

After you run those commands, Resource Explorer is running in the specified Regions in your AWS account. Resource Explorer builds and maintains an index in each Region with details of the resources located there. Resource Explorer replicates each of the individual Region indexes to the aggregator index in the specified Region. That Region also contains a view that allows any IAM role or user in the account to search for resources across all indexed Regions.

> **Note**
>
> Tagged resources local to the index appear in search results within a few minutes. Untagged resources typically take less than two hours to appear, but can take longer when there is heavy demand. It can also can take up to an hour to complete the initial replication to a new aggregator index from all of the existing local indexes.

# Identify which AWS Regions have Resource Explorer turned on

You can identify which AWS Regions have AWS Resource Explorer turned on by verifying if the Region contain an index for Resource Explorer. To view which Regions have an index, use the procedures on this page.

> **Important**
>
> Users can search for resources in *only* those Regions that have Resource Explorer turned on. You can also create an aggregator index in one Region to support searching for resources in all of your Regions. Resource Explorer replicates resource information to the Region with the aggregator index from all other Regions that contain a Resource Explorer index. Users can't use Resource Explorer to discover resources in Regions that don't have an index.

## Checking the Resource Explorer status in a Region

You can check which Regions have indexes for Resource Explorer by using the AWS Management Console, by using commands in the AWS Command Line Interface (AWS CLI), or by using API operations in an AWS SDK.

AWS Management Console

**To check which Regions have indexes for Resource Explorer**

1. Open the **Settings** page in the Resource Explorer console.
2. The list in the **Indexes** section includes only those Regions that contain a Resource Explorer index. The value in the **Type** column indicates whether the index is a **Local** index for its Region, or the **Aggregator** index for the AWS account.
3. To see which Regions don't contain a Resource Explorer, choose **Create indexes**. If a Region is not present, then the Region does not contain Resource Explorer.

AWS CLI

**To check which Regions have indexes for Resource Explorer**

Run the following command to see which AWS Regions have indexes for Resource Explorer.

```
$ aws resource-explorer-2 list-indexes
{
    "Indexes": [
        {
            "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
            "Region": "us-east-1",
            "Type": "AGGREGATOR"
        },
        {
            "Arn": "arn:aws:resource-explorer-2:us-
west-2:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd22222222",
            "Region":"us-west-2",
            "Type":"LOCAL"
        }
    ]
}
```

# Turning on Resource Explorer in an AWS Region to index your resources

When you initially turn on AWS Resource Explorer in your AWS account, you created indexes for the service in one or more AWS Regions. If you used the **Quick setup** option, Resource Explorer automatically created indexes in all AWS Regions that are turned on in your AWS account. The Resource Explorer service also promoted the index in the specified Region to be the aggregator index for the account. If you used the **Advanced setup** option, you specified the Regions in which to create indexes.

**Topics**


- Create a Resource Explorer index in a Region
- Considerations for AWS opt-in Regions


When you turn on Resource Explorer in an AWS Region, the service performs the following actions:

- When you start Resource Explorer in the ***first*** Region in an AWS account, Resource Explorer creates a service-linked role in the account named `AWSServiceRoleForResourceExplorer`. This role grants permissions for Resource Explorer to discover and index the resources in your account by using services such as AWS CloudTrail and the tagging service. Creation of the service-linked role happens only when you register the ***first*** AWS Region in the account. Resource Explorer uses the same service-linked role for all additional Regions that you add later.
- Resource Explorer creates an index in the specified Region to store the details about that Region's resources.
- Resource Explorer begins discovering the resources in the specified Region and adds the information it finds about them to that Region's index.
- If your account already contains an aggregator index in a different Region, Resource Explorer starts replicating the information from the new Region's index to the aggregator index to support cross-Region search.

When those steps are complete, information about your resources is available to be discovered by users. They can search by using one of the views defined in either the same Region or the Region that contains the aggregator index.

# Create a Resource Explorer index in a Region

You can create a Resource Explorer index in an additional AWS Region by using the AWS Management Console, by using commands in the AWS Command Line Interface (AWS CLI), or by using API operations in an AWS SDK. You can create only one index in a Region.

### Minimum permissions

To perform the steps in the following procedure, you must have the following permissions:

- **Action**: `resource-explorer-2:*` – **Resource**: no specific resource (*)
- **Action**: `iam:CreateServiceLinkedRole` – **Resource**: no specific resource (*)

AWS Management Console

#### To create a Resource Explorer index in an AWS Region

1. On the Resource Explorer **Settings** page.
2. In the **Indexes** section, choose **Create indexes**.
3. On the **Create indexes** page, select the check boxes next to the AWS Regions in which you want to create an index to support searching that Region's resources. Unavailable check boxes indicate Regions that already contain a Resource Explorer index.
4. (Optional) In the **Tags** section, you can specify tag key and value pairs to the index.
5. Choose **Create indexes**.

   Resource Explorer displays a green banner at the top of the page to indicate success, or a red banner if there is an error creating an index in one or more of the selected Regions.

> **Note**
>
> Tagged resources local to the index appear in search results within a few minutes. Untagged resources typically take less than two hours to appear, but can take longer when there is heavy demand. It can also can take up to an hour to complete the initial replication to a new aggregator index from all of the existing local indexes.

**Next step** – If you already created an aggregator index, then the new Regions automatically begin to replicate their index information to the aggregator index. If that is where your users do all of their searching, then the resources in the new Region appear in those search results and you're done.

However, if you want users to be able to search for resources in ***only*** the newly indexed Region, then you must also create a view for users in that Region and grant your users permissions to that view. For instructions on how to create a view, see Configuring an Resource Explorer view to provide access to resource searches.

AWS CLI

**To create a Resource Explorer index in an AWS Region**

Run the following command for each AWS Region in which you want to create an index to support searching that Region's resources. The following example command registers Resource Explorer in the US East (N. Virginia) (`us-east-1`).

```
$ aws resource-explorer-2 create-index \
    --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-11-01T20:00:59.149Z",
    "State": "CREATING"
}
```

Repeat this command for each Region in which you want to turn on Resource Explorer, substituting the appropriate Region code for the `--region` parameter.

Because Resource Explorer performs some of the index creation as asynchronous tasks in the background, the response can be `CREATING`, which indicates that the background processes are not yet complete.

> **Note**
>
> Tagged resources local to the index appear in search results within a few minutes. Untagged resources typically take less than two hours to appear, but can take longer when there is heavy demand. It can also can take up to an hour to complete the initial replication to a new aggregator index from all of the existing local indexes.

You can check for final completion by running the following command, and checking for the ACTIVE state.

```
$ aws resource-explorer-2 get-index \
    --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingFrom": [],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "LOCAL"
}
```

**Next step** – If you already created an aggregator index, then the new Regions automatically begin to replicate their index information to the aggregator index. If that is where your users do all of their searching, then the resources in the new Region appear in those search results and you're done.

However, if you want users to be able to search for resources in ***only*** the newly indexed Region, then you must also create a view for users in that Region and grant your users permissions to that view. For instructions on how to create a view, see Configuring an Resource Explorer view to provide access to resource searches.

# Considerations for AWS opt-in Regions

Opt-in Regions have higher security requirements than commercial Regions as it pertains to sharing IAM data through accounts in opt-in Regions. All of the data managed through the IAM service is considered identity data.

You can activate opt-in Regions using the AWS Resource Explorer console. See Turning on Resource Explorer in an AWS Region to index your resources for more information.

## Opt-out behaviors

Consider the following behaviors before you opt-out of an opt-in Region:

**Important**

Before you opt-out of a Region with an aggregator index, we suggest that you delete the aggregator index or demote it to a local index. Resource Explorer supports one aggregator index across all Regions within the partition.

- Your index isn't deleted, it's only disabled. If you choose to opt-in again later, your settings will revert.
- IAM disables IAM access to resources in the Region.
- Resource Explorer disables the index for the opted-out Region and stops ingesting data. The `ListIndexes` API won't show the Region index anymore.
- If your aggregator index is in a different Region, Resource Explorer stops data replication from the opted-out Region and cleans up the data within 24 hours.
- If you opt-out of your aggregator index Region, you will have to opt-in again to delete or demote the index.
- If you opt-in to the Region again, Resource Explorer re-enables the index and starts to ingest data.
- Any changes to the status of an opt-in Region takes about 24 hours to go into effect.

# Turning on cross-Region search by creating an aggregator index

With cross-region search enabled, you can search for resources across all of the Regions in your AWS account.

**Topics**


- About the aggregator index
- Promoting a local index to be the aggregator index for the account
- Demoting the aggregator index to a local index


## About the aggregator index

AWS Resource Explorer stores the information it collects about the resources in an AWS Region to a *local index* that Resource Explorer creates and maintains in that Region. For example, assume that you have an Amazon EC2 instance in the US West (Oregon) Region. Resource Explorer stores the details about that resource in the local index in the US West (Oregon) Region.

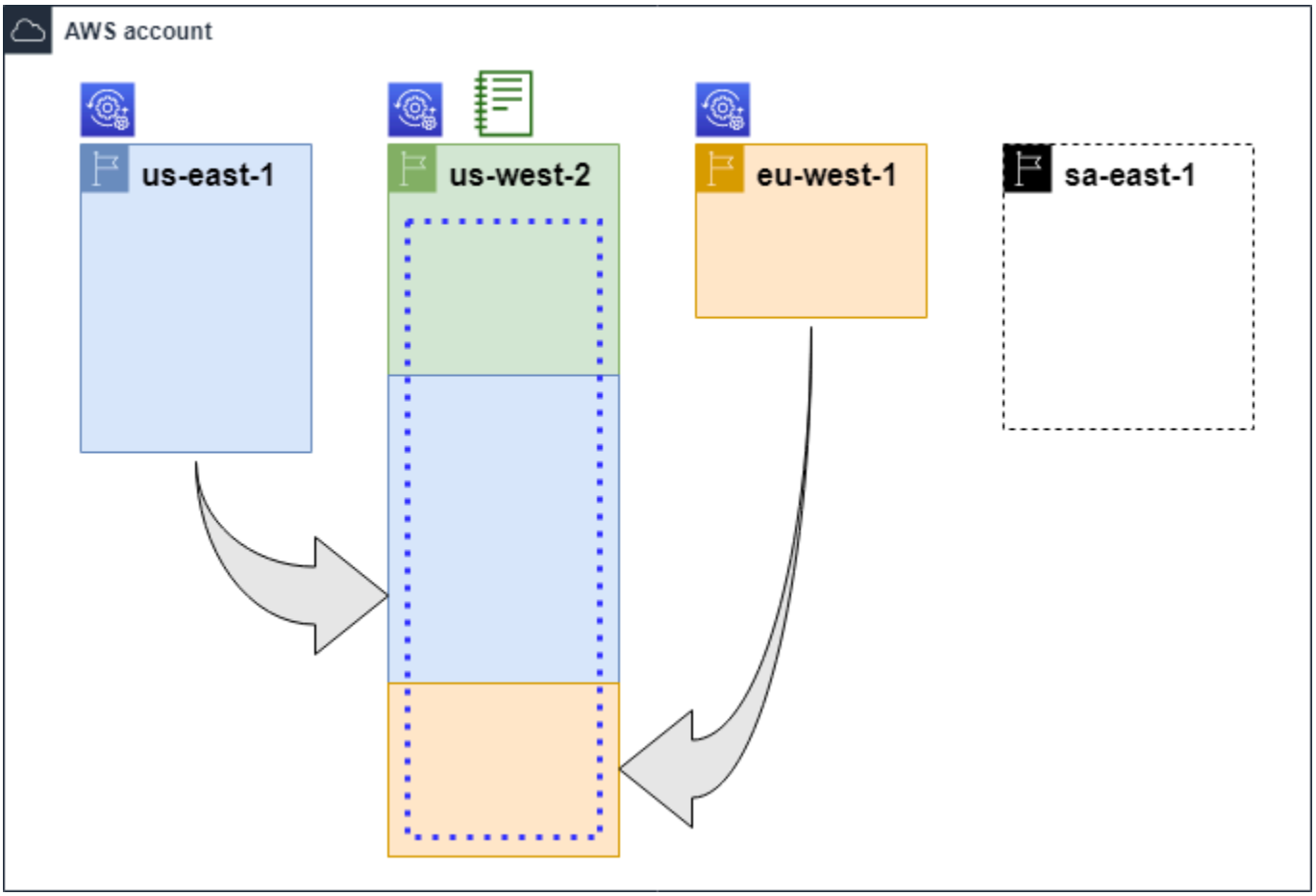
To support searching for resources across all AWS Regions in your account, you can convert the local index in *one* Region to be the aggregator index for your account.

The aggregator index contains a replicated copy of the local index in every other Region where you turned on Resource Explorer. This lets you create views in the Region that contains the aggregator index whose results can include resources from all AWS Regions in the account.

The following diagram shows an example of how the aggregator index works. In this example AWS account, the administrator does the following:

- Turns on Resource Explorer in three AWS Regions (`us-east-1`, `us-west-2`, and `eu-west-1`) by creating indexes in those Regions. Each Region contains its own local index.
- Chooses *not* to create an index in the `sa-east-1` Region. Users can't perform searches in `sa-east-1`, and no resources from that Region appear in any search results.
- Creates the aggregator index for the account in the `us-west-2` Region. This causes Resource Explorer to replicate information from the local indexes in all other Regions where Resource Explorer is turned on to the aggregator index. This allows searches performed in `us-west-2` to include resources from all three Regions in which Resource Explorer is turned on.

This configuration means that a user can perform cross-Region searches in ***only*** `us-west-2`, which contains the aggregator index. Only views from that Region can return results from all Regions in the account.




| **Legend** | |
|---|---|
|  | Resource Explorer is turned on in this AWS Region, and its resources are catalogued into an index in that Region. This Region's index is also replicated (indicated by the arrows) to the AWS Region that contains the aggregator index. |
|  | This AWS Region contains the aggregator index. Resource Explorer replicates the resource information collected in all other AWS Regions into this Region. |
|  | The default view created by **Quick Setup** includes all resources in all AWS Regions. |

# Promoting a local index to be the aggregator index for the account

You have the option to create an aggregator index in one AWS Region when you first set up AWS Resource Explorer. For more information, see Setting up and configuring Resource Explorer. This procedure is about promoting one of the local indexes to be the aggregator index for the account if you didn't do it at initial set up.

**Important**

- You can have only one aggregator index in an AWS account. If the account already has an aggregator index, you must first either demote it to a local index or delete it.
- After deleting or changing which Region contains the aggregator index, you must wait 24 hours before you can promote another index to be the aggregator index.

AWS Management Console

**To promote a local index to be the aggregator index for the account**

1. Open the Resource Explorer **Settings** page.
2. In the **Indexes** section, select the check box next to the index that you want to promote, and then choose **Change index type**.
3. In the **Change index type for <*Region name*>** dialog, choose **aggregator index**, and then choose **Save changes**.

AWS CLI

**To promote a local index to be the aggregator index for the account**

The following example command updates the index in the specified AWS Region from type LOCAL to type AGGREGATOR. You must call the operation from the AWS Region that you want to contain the aggregator index.

```
$ aws resource-explorer-2 update-index-type \
    --arn arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111 \
```

```
    --type AGGREGATOR \
    --region us-east-1
{
    "Arn":"arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "LastUpdatedAt":"2022-07-13T18:41:58.799Z",
    "State":"UPDATING",
    "Type":"AGGREGATOR"
}
```

The operation works asynchronously and starts with `State` set to `UPDATING`. To check if the operation has completed, you can run the following command and look for the value `ACTIVE` in the `State` response field. You must run this command in the Region the contains the index you want to check.

```
$ aws resource-explorer-2 get-index --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-10-12T21:31:37.277000+00:00",
    "LastUpdatedAt": "2022-10-12T21:31:37.677000+00:00",
    "ReplicatingFrom": [
        "us-west-2",
        "us-east-2",
        "us-west-1"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "AGGREGATOR"
}
```

# Demoting the aggregator index to a local index

You can demote an aggregator index to a local index, such as when you want to move the aggregator index to a different AWS Region.

When you demote an aggregator index to a local index, Resource Explorer stops replicating the indexes from other AWS Regions. It also starts an asynchronous background task to delete any replicated information from other Regions. Until that asynchronous task completes, some cross-Region results can continue to appear in search results.

**Notes**

- After you demote an aggregator index, you must wait 24 hours before you can promote either the same index or the index in a different Region to be the new aggregator index for the account.
- After demoting an aggregator index, it can take up to 36 hours for the background processes to complete and for all resource information from other Regions to disappear from results from searches performed in this Region.
- If you demote a member account within an organization wide view, the member may be removed from multi-account search.

You can check the status of the background task by viewing the list of indexes on the **Settings** page or by using the GetIndex operation. When the asynchronous tasks complete, the `Status` field from the index changes from `UPDATING` to `ACTIVE`. At that time, only results from the local Region appear in query results.

AWS Management Console

**To demote an aggregator index to a local index**

1. Open the Resource Explorer **Settings** page.
2. In the **Indexes** section, select the check box next to the Region that contains the aggregator index that you want to demote to a local index, and then choose **Change index type**.
3. In the **Change index type for <*Region name*>** dialog, choose **Local index**, and then choose **Save changes**.

AWS CLI

**To demote an aggregator index to a local index**

The following example demotes the specified aggregator index to a local index. You must call the operation in the AWS Region that currently contains the aggregator index.

```
$ aws resource-explorer-2 update-index-type \
    --arn arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111 \
    --type LOCAL \
```

```
    --region us-east-1
{
    "Arn":"arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "LastUpdatedAt":"2022-07-13T18:41:58.799Z",
    "State":"UPDATING",
    "Type":"LOCAL"
}
```

The operation works asynchronously and starts with `State` set to `UPDATING`. To check if the operation has completed, you can run the following command and look for the value `ACTIVE` in the `State` response field. You must run this command in the Region the contains the index you want to check.

```
$ aws resource-explorer-2 get-index --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-10-12T21:31:37.277000+00:00",
    "LastUpdatedAt": "2022-10-12T21:31:37.677000+00:00",
    "ReplicatingFrom": [
        "us-west-2",
        "us-east-2",
        "us-west-1"
    ],
    "State": "ACTIVE",
    "Tags": {},
    "Type": "LOCAL"
}
```

# Turning on multi-account search

With multi-account search, you can search for resources across accounts with active indexes in your AWS Organizations or organizational unit (OU).

**Topics**


- Prerequisites
- Enable multi-account search
- Multi-account Quick Setup
- Effect of account actions on Resource Explorer multi-account search


## Prerequisites

To turn on multi-account search for your organization, complete the following:

- For opt-in Regions, verify your management account is also opted-in where you are turning on multi-account search.
- Create an administrative user.
- Create a service-linked role in the administrator account with `aws iam create-service-linked-role --aws-service-name resource-explorer-2.amazonaws.com`.
- Enable trusted access in AWS Organizations. This allows full integration with Resource Explorer to list resources across all accounts in your organization.
- Assign a delegated administrator (*recommended*). For more information, see Delegated administrator for AWS services that work with Organizations in the *AWS Organizations User Guide*.
  - Resource Explorer supports only 1 delegated administrator who performs similar actions to the management account.
  - Removing or changing the delegated administrator for your organization results in the removal of all multi-account views created in their account.

## Enable multi-account search

To search and discover resources across your organization's accounts, you must complete the following steps:

1. Activate AWS Resource Explorer in one or more accounts in your AWS Organizations.
2. Register one Region to contain the aggregator index.
3. Choose a Region in which to create an aggregator index. This Region must be consistent across your AWS Organizations.
4. Create a Resource Explorer view that's scoped to your AWS Organizations or organizational unit. Create this view in the aggregator Region from the preceding step.
5. Share the view with accounts across your organization.

## Multi-account Quick Setup

Enable Resource Explorer across multiple accounts in your organization with the Quick Setup.

> **Note**
>
> This process does not deploy any resources in the management account. If you are using the management account and you want indexes in the account, you must manually add them with the Resource Explorer onboarding flow.

1. Navigate to Quick Setup for Resource Explorer in the Systems Manager console.
2. Choose your **Aggregator index Region**. This allows you to search for resources located in all Regions in the selected target accounts. If any of the selected target accounts already have an aggregator index configured in another Region, the existing aggregator index will be automatically replaced with this new Region.
3. Choose your account **Targets**. You can enable Resource Explorer for your entire organization or for specific organizational units (OUs).

   > **Note**
   >
   > You can deploy to a maximum of 50,000 AWS CloudFormation stacks at a time. If you have a large organization that spans multiple Regions, you should deploy at the OU level in smaller batches.

4. Read through the summary of acknowledgements before you choose **Create**.

# Effect of account actions on Resource Explorer multi-account search

> **Note**
>
> It takes up to 24 hours to remove accounts and resources from multi-account search results.

Account actions have the following effects on AWS Resource Explorer multi-account search.

## Resource Explorer disabled

When you disable Resource Explorer for an account, it is disabled only for that account in the AWS Region that is selected when you disable it.

You must disable Resource Explorer separately in each Region where it's enabled.

After 24 hours, resources from this account won't appear in search results.

Other Resource Explorer data and settings are not removed.

## Member account is removed from an organization

When a member account is removed from an organization, the Resource Explorer administrator account loses permissions to view resources in the member account.

If the removed account is an administrator or delegated administrator account, all the multi-account views previously created by these accounts will also be removed.

Resource Explorer continues to run in both accounts.

Resource search results no longer include resources from this account.

## Account is suspended

When an account is suspended in AWS, the account loses permissions to view resources in Resource Explorer. The administrator account for a suspended account can view the existing resources.

For an organization account, the member account status can also change to **Account Suspended**. This happens if the account is suspended at the same time that the administrator account attempts

to enable the account. The administrator account for an **Account Suspended** account cannot view resources for that account.

Otherwise, the suspended status doesn't affect the member account status.

After 90 days, the account is either deactivated or reactivated. When the account is reactivated, its Resource Explorer permissions are restored. If the member account status is **Account Suspended**, the administrator account must enable the account manually.

## Account is closed

When an AWS account is closed, Resource Explorer responds to the closure as follows:

- Resource Explorer retains the resources for the account for 90 days from the effective date of the account closure. At the end of the 90 day period, Resource Explorer permanently deletes all resources for the account.
- To retain resources for more than 90 days, you can use a custom action with an EventBridge rule to store the resources in an Amazon S3 bucket. As long as Resource Explorer retains the resources, when you reopen the closed account, Resource Explorer restores the resources for the account.
- If the account is a Resource Explorer administrator account, it is removed as an administrator and all the member accounts are removed. If the account is a member account, it is disassociated and removed as a member from the Resource Explorer administrator account.
- For more information, see Closing an account.

## Account opt-out

If an account opts-out of a Region, you will still see their resources in search results for up to 24 hours.

After 24 hours, resources from this account won't appear in search results. For more information, see Opt-out behaviors.

# Supporting unified search in the AWS Management Console

The AWS Management Console has a search bar at the top of every console page. This provides a *unified search* experience across all AWS services. Unified search results can include such things as:

- AWS service and feature console pages.
- AWS documentation pages.
- AWS blog and Knowledge Base articles
- Resources in your accounts — if you follow the steps below.

To see your account's resources in your unified search results, you must perform the following steps. You can do this during initial setup of AWS Resource Explorer. It all happens automatically if you use the **Quick setup** option.

- You must create an aggregator index in one AWS Region for the AWS account.
- You must create a default view in the AWS Region that contains the aggregator index.
- You must grant all principals that need to search for resources in the unified search bar permission to search using that default view.

Unified search always uses the default view in the AWS Region that contains the aggregator index to perform all searches.

# Deploying Resource Explorer to the accounts in an organization

By using AWS CloudFormation StackSets, you can define and deploy to all of the accounts managed in an organization by AWS Organizations. When you define a stack set, you specify AWS resources that you want created across your AWS Regions and across all of the target accounts that you specify. When all of the accounts are part of the same organization, you can take advantage of AWS CloudFormation integration with Organizations and let those services handle the cross-account role creation. You can enable automatic deployment in an organization, which automatically deploys stack instances to new accounts that you might add to the target organization or an organizational unit (OU) in the future. If you remove an account from the organization, then AWS CloudFormation automatically deletes any resources that were deployed as part of an organization stack instance. For more information about StackSets, see Working with AWS CloudFormation StackSets in the *AWS CloudFormation User Guide*.

You can use AWS CloudFormation StackSets to turn on and configure AWS Resource Explorer in all of the accounts in your organization, creating indexes in each enabled Region, and creating views where you need them.

> **Important**
>
> If you try to setup an aggregator index in a Region, you must make sure the account doesn't have an existing aggregator index in any other Regions. After you demote an aggregator index to a local index, you must wait 24 hours before you can promote another index to be the new aggregator index for the account.

## Prerequisites

To use AWS CloudFormation StackSets to deploy Resource Explorer to the accounts in your organization, you, or the administrator of your organization, must first perform the following steps to enable stacks with service-managed permissions:

1. The organization must have all features enabled. If the organization has only consolidated billing features enabled, you can't create a stack set with service-managed permissions.

2. Turn on trusted access between AWS CloudFormation and Organizations. This grants AWS CloudFormation permission to create the roles needed in the organization's management account and the member accounts AWS CloudFormation will deploy Resource Explorer indexes and views.

Now you can create stack sets with service-managed permissions.

> **Important**
>
> You must create the stack sets in the organization's management account. AWS CloudFormation is a Regional service, so you can view and manage the stack sets you create from only the Region you originally created them in.

## Creating the stack sets for Resource Explorer

The fully deploy Resource Explorer, you must deploy two stack sets.

- The first stack set creates the aggregator index and default view that lets users search for resources across all of the Regions in the account.

  Deploy this stack set to ***only*** the single Region in which you want to create the aggregator index.
- The second stack sets creates a local index and default view. The local index replicates its content to the aggregator index.

  Deploy this stack set to every enabled Region in the account ***except*** the Region that contains the aggregator index. Don't choose any Regions that aren't enabled in the accounts to which you deploy the stack. If you do, the deployment fails.

Sample templates for each of these are in the following section. For step-by-step instructions on how to create a stack set using these templates, see Create a stack set with service-managed permissions in the *AWS CloudFormation User Guide*.

After you deploy these stack sets to your organization, every account within the scope you selected, organization or organizational unit, has an aggregator index in the specified Region, and local indexes in every other Region.

# Sample AWS CloudFormation templates

The following sample template creates the account's aggregator index and a default view that can search for resources across all Regions in the account where you deploy an index.

YAML

```
Description: >-
  CFN Stack setting up ResourceExplorer with an Aggregator Index, and a new Default
 View.
Resources:
  Index:
    Type: 'AWS::ResourceExplorer2::Index'
    Properties:
      Type: AGGREGATOR
      Tags:
        Purpose: ResourceExplorer CFN Stack
  View:
    Type: 'AWS::ResourceExplorer2::View'
    Properties:
      ViewName: DefaultView
      IncludedProperties:
        - Name: tags
      Tags:
        Purpose: ResourceExplorer CFN Stack
    DependsOn: Index
  DefaultViewAssociation:
    Type: 'AWS::ResourceExplorer2::DefaultViewAssociation'
    Properties:
      ViewArn: !Ref View
```

JSON

```
{
    "Description": "CFN Stack setting up ResourceExplorer with an Aggregator Index,
 and a new Default View.",
    "Resources": {
        "Index": {
            "Type": "AWS::ResourceExplorer2::Index",
            "Properties": {
                "Type": "AGGREGATOR",
                "Tags": {
                    "Purpose": "ResourceExplorer CFN Stack"
```

```
                }
            }
        },
        "View": {
            "Type": "AWS::ResourceExplorer2::View",
            "Properties": {
                "ViewName": "DefaultView",
                "IncludedProperties": [{
                    "Name": "tags"
                }],
                "Tags": {
                    "Purpose": "ResourceExplorer CFN Stack"
                }
            },
            "DependsOn": "Index"
        },
        "DefaultViewAssociation": {
            "Type": "AWS::ResourceExplorer2::DefaultViewAssociation",
            "Properties": {
                "ViewArn": {
                    "Ref": "View"
                }
            }
        }
    }
}
```

The following sample template creates a local index in each enabled Region in all accounts other than the one with the aggregator index. It also creates a default view that users can search for resources in only that Region. Users must search with a view in the aggregator Region to search for resource across all Regions.

YAML

```
Description: >-
  CFN Stack setting up ResourceExplorer with a Local Index, and a new Default View.
Resources:
  Index:
    Type: 'AWS::ResourceExplorer2::Index'
    Properties:
      Type: LOCAL
      Tags:
```

```
        Purpose: ResourceExplorer CFN Stack
  View:
    Type: 'AWS::ResourceExplorer2::View'
    Properties:
      ViewName: DefaultView
      IncludedProperties:
        - Name: tags
      Tags:
        Purpose: ResourceExplorer CFN Stack
    DependsOn: Index
  DefaultViewAssociation:
    Type: 'AWS::ResourceExplorer2::DefaultViewAssociation'
    Properties:
      ViewArn: !Ref View
```

JSON

```
{
    "Description": "CFN Stack setting up ResourceExplorer with a Local Index, and a
 new Default View.",
    "Resources": {
        "Index": {
            "Type": "AWS::ResourceExplorer2::Index",
            "Properties": {
                "Type": "LOCAL",
                "Tags": {
                    "Purpose": "ResourceExplorer CFN Stack"
                }
            }
        },
        "View": {
            "Type": "AWS::ResourceExplorer2::View",
            "Properties": {
                "ViewName": "DefaultView",
                "IncludedProperties": [{
                    "Name": "tags"
                }],
                "Tags": {
                    "Purpose": "ResourceExplorer CFN Stack"
                }
            },
            "DependsOn": "Index"
        },
```

```
        "DefaultViewAssociation": {
            "Type": "AWS::ResourceExplorer2::DefaultViewAssociation",
            "Properties": {
                "ViewArn": {
                    "Ref": "View"
                }
            }
        }
    }
}
```

# Configuring an Resource Explorer view to provide access to resource searches

Views are the key to searching for your resources. Every AWS Resource Explorer search operation must use a view. Views are the method the administrator can use to control access to the information about resources in your AWS account.

A view can be accessed by only principals (IAM roles or users) that have permission to use that view. To search successfully with Resource Explorer, a principal must have `Allow` access to both the `resource-explorer-2:GetView` and `resource-explorer-2:Search` operations on the view's ARN.

Views contain built-in filters that the administrator can use to limit results to only items of interest. For example, you can create a view that includes only resources related to a certain project. Users who don't need to see information about other projects can use this view to see only those resources of interest.

A view is a Regional resource. The view is created and stored in a specific AWS Region and returns in its results only information from the index in that Region. To include results from across all Regions in the account, the view must reside in the Region that contains the aggregator index. That Region contains a replica of the indexes from all other Regions in the account.

There are several key elements to every view:

**Permissions to search**

You can use standard AWS permission policies to control who can use each view. This is provided by identity-based permission policies attached to the principals that give you granular control over who can see the information provided by each view. For example, you can grant access to the `Production-resources` view to allow searching only by the engineers that operate your production services. Then, you can grant different permissions to the `Pre-production-resources` view to allow searching for pre-production resources by your developers.

If you use the AWS managed policy named `AWSResourceExplorerReadOnlyAccess` with your principals, it grants them the ability to search using any view in the account.

Alternatively, you can create your own permissions policy and grant the following permissions for only specified views:

- `resource-explorer-2:GetView`
- `resource-explorer-2:Search`

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

For more information about permissions related to views, see Granting access to Resource Explorer views for search.

**Filtering the search**

A view serves as a virtual window through which the user can see the resources in the account. You can create multiple views, each presenting a different view of the larger picture. For example, you can create a view that allows searching only resources associated with your pre-production environment, as identified by tags attached to your resources. Then, you could create a separate view that allows searching only resources in your production environment, based on different values in the tags. If you configure multiple views with different `FilterString` values, you don't have to re-enter those query parameters every time you Search.

Views also can specify which optional pieces of information about the resources to include in the results. The default list of fields is always included in results. In addition to the default list, you can request that the view also include any tags attached to the resource.

**Scope of the search**

- **Region scope** – When you search in an AWS Region with Resource Explorer, the results can include only resources that are indexed in that Region. The index in most Regions is labelled

LOCAL because it contains information about resources within only that Region. Searches in those Regions can return only those resources.

- **Account scope** – You can promote one local index to be the aggregator index for the account. When you do this, all other Regions where Resource Explorer is turned on replicate their index information to the Region with the aggregator index. If you search in that Region, those results include resources from all Regions in the account. When you use the **Quick setup** option to configure the server, Resource Explorer automatically creates an aggregator index in the Region you specify. Also, the **Quick Setup** option creates a default view in that Region to support searching all resources in the account across all Regions.

# Default views

If a user attempts to search without explicitly specifying a view, Resource Explorer uses the *default view* defined for that AWS Region.

If a default view doesn't exist for that Region and the user didn't specify a view to use, then the search fails and generates an exception.

Resource Explorer automatically creates a default view as follows:

- If you turn on Resource Explorer using the AWS Management Console and choose the **Quick setup** option, you must specify which Region contains the aggregator index for the account. Resource Explorer automatically creates a default view in the specified aggregator index Region.
- If you register Resource Explorer using the AWS Management Console and choose the **Advanced setup** option, you can *optionally* choose to create the aggregator index for the account in a specified Region. If you do this, Resource Explorer creates a default view automatically in the aggregator index Region.
- If you register Resource Explorer by using the console and choose *not* to register an aggregator index Region, Resource Explorer creates a default view for the local index in each Region.
- If you register Resource Explorer by using the AWS CLI or the API operations, Resource Explorer doesn't automatically create a default view. Instead, you must configure the default view manually for each Region where you expect users to search from.

# Creating Resource Explorer views to use for search

All searches must use a *view*. A view defines filters that determine which resources can be returned by queries that use the view. Views also control who can search for resources.

A view is stored in an AWS Region, and returns search results from only that Region's index. If the Region contains the aggregator index, then the view returns search results from the index in every Region in the account.

Multi-account views allow you to search for resources in accounts across your organization. Any account you wish to search requires indexes. Only the management account, or a delegated administrator for the organization, can create a multi-account view.

AWS Resource Explorer can create a default view for you during initial set up if you chose the relevant options in either Quick Setup for Resource Explorer in the Systems Manager console or **Advanced setup**. At any later time, you can create additional views that have different filters for different sets of users.

You can create a view by using the AWS Management Console or by running AWS CLI commands or their equivalent API operations in an AWS SDK.

**Minimum permissions**

To run this procedure, you must have the following permissions:

- **Action:** `resource-explorer-2:CreateView`

  **Resource:** This can be `*` to allow creation of a view in any AWS Region in the account.

AWS Management Console

**To create a view**

1. Open the Resource Explorer console **Views** page and choose **Create view**.
2. On the **Create view** page, for **Name**, enter a name for the view.

   The name must be no more than 64 characters long, and can include letters, digits, and the hyphen (-) character. The name must be unique within its AWS Region.

3. Choose the AWS Region in which you want to create the view. To create a view that returns resources from all Regions in the account, choose the AWS Region that contains the aggregator index.
4. (Optional) For **Scope**, choose whether your search returns multi-account resources, or returns resources only from your account. Account level scope is the default.

   Only the management account or delegated administrator can see the option to create a multi-account view.
5. Choose whether to filter the results.

   - **Include all resources**

     No query filters are included. All resources in the index associated with the view can be returned in search results.
   - **Include only resources that match a specified filter**

     Turns on the **Resource filters** check box where you can choose filter *names* and *operators*. For an explanation of each of the available filter names and operators, see Filters.
   - Choose the optional resource attributes to include in results from this view. Select the check box next to **Tags** to let users search for resources based on their tag key names and values. If you don't include tags in the view then users can't make search requests that use tag keys and values to further filter the results.
   - Optionally, you can attach tags to the view. Expand the **Tags** box, and enter up to 50 tag key/value pairs. You can use tags to categorize resources, or as part of an attribute-based access control (ABAC) security permission strategy. For more information, see Adding tags to views.
   - Choose **Create view**.

   The console returns to the **Search** page where you can use your new view to perform a search.

   **Next step:** Grant the principals in your account permissions to search with your new view. For more information, see Granting access to Resource Explorer views for search

AWS CLI

**To create a view**

Run the following command to create a view in the specified AWS Region. The following example creates a view that returns only resources related to the Amazon EC2 service that are tagged with a `Stage` key and the value `prod`.

```
$ aws resource-explorer-2 create-view \
    --region us-west-2 \
    --view-name "My-EC2-Prod-Resources" \
    --filters FilterString="service:ec2 tag:stage=prod" \
    --included-properties Name=tags
{
    "View": {
        "Filters": {
            "FilterString": "service:ec2 tag:stage=prod"
        },
        "IncludedProperties": [
            {
                "Name": "tags"
            }
        ],
        "LastUpdatedAt": "2022-08-03T16:13:37.625000+00:00",
        "Owner": "123456789012",
        "Scope": "arn:aws:iam::123456789012:root",
        "ViewArn": "arn:aws:resource-explorer-2:us-west-2:123456789012:view/My-EC2-
Prod-Resources/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
    }
}
```

**To create an organization level view**

The following example creates a view that returns resources from across your organization. This must be performed by the organization's management account, or a delegated administrator account.

1. Run the `aws organizations describe-organization` command to get your organization ARN.
2. Run the following command to create a view for the specified organization.

```
$ aws resource-explorer-2 create-view \
    --region us-west-2 \
    --view-name entire-org-view \
    --scope "arn:aws:organizations::111111111111:organization/o-exampleorgid"
{
```

```
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [],
        "LastUpdatedAt": "2022-08-03T16:13:37.625000+00:00",
        "Owner": "111111111111",
        "Scope": "arn:aws:organizations::111111111111:organization/o-
exampleorgid",
        "ViewArn": "arn:aws:resource-explorer-2:us-west-2:111111111111:view/
entire-org-view/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
    }
}
```

**To create an organizational unit level view**

The following example creates a view that returns resources from all members of this organizational unit. This view behaves similarly to an organizational level view. This must be performed by the organization's management account, or a delegated administrator account.

1. Run the `aws organizations describe-organizational-unit` command to get your organization ARN.
2. Run the following command to create a view for the specified organizational unit.

```
$ aws resource-explorer-2 create-view \
    --region us-west-2 \
    --view-name entire-ou-view \
    --scope "arn:aws:organizations::222222222222:ou/o-exampleorgid/ou-
exampleouid"
{
    "View": {
        "Filters": {
            "FilterString": ""
        },
        "IncludedProperties": [],
        "LastUpdatedAt": "2022-08-03T16:13:37.625000+00:00",
        "Owner": "222222222222",
        "Scope": "arn:aws:organizations::222222222222:ou/o-exampleorgid/ou-
exampleouid",
        "ViewArn": "arn:aws:resource-explorer-2:us-west-2:222222222222:view/
entire-ou-view/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
```

```
    }
}
```

**Next step:** Grant the principals in your account permissions to search with your new view. For more information, see Granting access to Resource Explorer views for search

# Granting access to Resource Explorer views for search

Before users can search with a new view, you must grant access to AWS Resource Explorer views. To do this, use an identity-based permission policy to the AWS Identity and Access Management (IAM) principals that need to search with the view.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

You can use either of the following methods:

- **Use an existing AWS managed policy**. Resource Explorer provides several pre-defined AWS managed policies for your use. For details of all of the available AWS managed policies, see AWS managed policies for AWS Resource Explorer.

  For example, you could use the `AWSResourceExplorerReadOnlyAccess` policy to grant search permissions to all views in the account.

- **Create your own permission policy and assign it to the principals**. If you create your own policy, you can restrict access to a single view, or a subset of the available views by specifying the Amazon resource name (ARN) of each view in the `Resource` element of the policy statement. For example, You can use the following example policy to grant that principal the ability to search using only that one view.

  JSON

  ```
  {
      "Version": "2012-10-17",
      "Statement": [
          {
              "Effect": "Allow",
              "Action": [
                  ":Search",
                  ":GetView"
              ],
              "Resource": "arn:aws:::123456789012:view/MyTestView/"
          }
      ]
  }
  ```

  Use the IAM console to create the permission policies and to use them with the principals that need those permissions. For more information about IAM permission policies, see the following topics:

  - Policies and permissions in IAM
  - Adding and removing IAM identity permissions
  - Understanding permissions granted by a policy

## Using tag-based authorization to control access to your views

If you choose to create multiple views with filters that return results with only certain resources, then you might also want to restrict access to those views to only the principals who need to see those resources. You can provide this type of security for the views in your account by using an attribute-based access control (ABAC) strategy. The *attributes* used by ABAC are the tags attached both to the principals attempting to perform operations in AWS and to the resources they attempt to access.

ABAC uses standard IAM permission policies attached to the principals. The policies use `Condition` elements in the policy statements to allow access only when both the tags attached to the requesting principal and the tags attached to the affected resource match the requirements in the policy.

For example, you could attach a tag `"Environment" = "Production"` to all of the AWS resources that support your company's production application. To ensure that only principals that are authorized to access the production environment can see those resources, create a Resource Explorer view that uses that tag as a filter. Then, to restrict access to the view to only the appropriate principals, you grant permissions using a policy that has a condition similar to the following example elements.

```
{
    "Effect": "Allow",
    "Action": [ "service:Action1", "service:Action2" ],
    "Resource": "arn:aws:arn-of-a-resource",
    "Condition": { "StringEquals": {"aws:ResourceTag/Environment":
 "${aws:PrincipalTag/Environment}"} }
}
```

That `Condition` in the previous example specifies that the request is allowed ***only*** if the `Environment` tag attached to the principal making the request matches the `Environment` tag attached to the resource specified in the request. If those two tags don't exactly match, or if either tag is missing, then the Resource Explorer denies the request.

> **Important**
>
> To successfully use ABAC to secure access to your resources, you must first restrict access to the ability to add or modify the tags attached to your principals and resources. If a user can add or modify the tags attached an AWS principal or resource then that user can affect the permissions controlled by those tags. In a secure ABAC environment, only approved security administrators have permission to add or modify the tags attached to principals, and only security administrators and resource owners can add or modify the tags attached to resources.

For more information about how to successfully implement an ABAC strategy, see the following topics in the *IAM User Guide*:

- IAM tutorial: Define permissions to access AWS resources based on tags
- Controlling access to AWS resources using tags

After you have the necessary ABAC infrastructure in place, you can use start using tags to control who can search using the Resource Explorer views in your account. For example policies that illustrate the principle, see the following example permission policies:

- Granting access to a view based on tags
- Granting access to create a view based on tags

## Setting a default view in an AWS Region

In AWS Resource Explorer, you can define many views in an AWS Region, where each view addresses different search requirements. We recommend that you set ***one*** view in each Region as the *default view* for that Region.

Resource Explorer uses the default view whenever a user performs a search and doesn't explicitly specify which view to use. The unified search bar at the top of every AWS Management Console page also automatically uses the default view in the Region that contains the aggregator index to find resources that match the user's search query.

You can select only a view that exists in the Region to be that Region's default view. If another Region has a view that you want to use, you must first create a copy of that view in the Region in which you want to make it the default view.

> **Tip**
>
> There is no *copy view* operation. You must create a view in the target Region and then copy the settings from the existing view to the new view.

You can specify a view as the default for its Region by using the AWS Management Console or by running AWS CLI commands or their equivalent API operations in an AWS SDK.

AWS Management Console

**To set a default view**

1. On the Resource Explorer **Views** page, choose the option button next to the view that you want to make the default for its Region.
2. Choose **Actions**, then choose **Set as default**.

AWS CLI

**To set a default view**

Run the following command to set the specified view as the default for its Region. The following example sets the specified view to be the default for all searches performed in the us-east-1 Region. That view must exist in the Region in which you run the command.

```
$ aws resource-explorer-2 associate-default-view \
    --region us-east-1 \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/
MyViewName/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/
MyViewName/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
}
```

# Adding tags to views

You can add tags to your views to categorize them. Tags are customer-supplied metadata that take the form of a key name string and an associated optional value string. For general information about tagging AWS resources, see Tagging AWS Resources in the *Amazon Web Services General Reference*.

## Add tags to your views

You can add tags to your Resource Explorer views by using the AWS Management Console or by running AWS CLI commands or their equivalent API operations in an AWS SDK.

AWS Management Console

**To add tags to a view**

1. Open the Resource Explorer **Views** page and choose the name of the view that you want to tag to display its **Details** page.
2. Under **Tags**, choose **Manage tags**.
3. To add a tag, choose **Add tag** and then enter a tag key name and optional value.

   > **Note**
   >
   > You can also delete a tag by choosing the **X** next to the tag.

   You can attach up to 50 user-defined tags to a resource. Any tags that are created and managed automatically by AWS don't count against this quota.
4. When you're done with all tag changes, choose **Save changes**.

AWS CLI

**To add tags to a view**

Run the following command to add tags to a view. The following example add tags with the key name `environment` and the value `production` to the specified view.

```
$ aws resource-explorer-2 tag-resource \
    --resource-id arn:aws:resource-explorer-2:us-east-1:123456789012:view/
MyViewName/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111 \
    --tags environment=production
```

The preceding command produces no output if it succeeds.

> **Note**
>
> To remove an existing tag from a view, use the `untag-resource` command.

## Controlling permissions with tags

One key use of tagging is to support an attribute-based access control (ABAC) strategy. ABAC can help simplify permission management by letting you tag resources. Then, you grant permission to users for resources that are tagged a certain way.

For example, consider this scenario. For a view called `ViewA`, you attach the tag `environment=prod` (*key name=value*). Another `ViewB` might be tagged `environment=beta`. You tag your roles and users with the same tags and values, based on which environment each role or user should be able to access.

Then, you could assign an AWS Identity and Access Management (IAM) permission policy to your IAM roles, groups, and users. The policy grants permission to access and search using a view only if the role or user making the search request has an `environment` tag with the same value as the `environment` tag attached to the view.

The benefit to this approach is that it's dynamic and doesn't require you to maintain a list of who has access to which resources. Instead, you ensure that all resources (your views) and principals (IAM roles and users) are tagged properly. Then, the permissions update automatically without you having to change any policies.

## Referencing tags in an ABAC policy

After your views are tagged, you can choose to use those tags to control access dynamically to those views. The following example policy assumes that both your IAM principals and your views are tagged with the tag key `environment` and some value. When that is done, you can attach the following example policy to your principals. Your roles and users can then `Search` using any views that are tagged with an `environment` tag value that exactly matches the `environment` tag attached to the principal.

JSON

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

```
                ":GetView",
                ":Search"
            ],
            "Resource": "arn:aws:::123456789012:view/*",
            "Condition": {
                "ForAnyValue:StringEquals": {
                    "aws:ResourceTag/environment":
 "${aws:PrincipalTag/environment}"
                }
            }
        }
    ]
}
```

If both the principal and view have the `environment` tag but the values don't match, or if either is missing the `environment` tag then Resource Explorer denies the search request.

For more information about using ABAC to securely grant access to your resources, see What is ABAC for AWS?

## Sharing Resource Explorer views

Views in AWS Resource Explorer primarily use resource-based policies to grant access. Similar to Amazon S3 bucket policies, these policies are attached to the view and specify who can use the view. This is in contrast to AWS Identity and Access Management (IAM) identity-based policies. An IAM identity-based policy is assigned to a role, group, or user, and it specifies which actions and resources that role, group, or user can access. You can use either type of policy with Resource Explorer views, as follows:

- Within the management account or delegated administrator account that owns the resource, use ***either*** policy type to grant access, provided that no other policy explicitly denies access to the view for that principal.
- Across accounts, you must use ***both*** policy types. The resource-based policy attached to the view in the *sharing* account turns on sharing with another *consuming* account. However, that policy doesn't grant access to the individual users or roles in the consuming account. The administrator in the consuming account must also assign an identity-based policy to the desired roles and users in the consuming account. That policy grants access to the Amazon resource name (ARN) of the view.

To share views with other accounts, you must use AWS Resource Access Manager (AWS RAM). AWS RAM handles the complexity of resource-based policies for you. Before you can share, you must perform the following tasks:

- Turn on multi-account search.
- Ensure that your resource-based policy or the IAM identity-based policy you use to share and unshare views includes the `resource-explorer-2:GetResourcePolicy`, `resource-explorer-2:PutResourcePolicy` and `resource-explorer-2:DeleteResourcePolicy` permissions.

To share a view, you must be the organization's management account or a delegated administrator. You specify the accounts or identities that you want to share the resource with. AWS RAM fully supports Resource Explorer views. AWS RAM uses policies similar to those described in the following sections, based on the types of the principals that you choose to share with. For instructions on how to share resources, see Sharing your AWS resources in the *AWS Resource Access Manager User Guide*.

Administrators and delegated administrators can create and share 3 types of views: organization scope view, organizational unit (OU) scope views, and account-level scope views. They can share with organizations, OUs, or accounts. When accounts join or leave the organization, AWS RAM automatically grants or revokes the shared view.

## Permissions policy to share view with AWS accounts

The following example policy shows how you can make a view available to the principals in two different AWS accounts:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": [ "111122223333", "444455556666" ]
            },
            "Action": [
                "resource-explorer-2:Search",
```

```
                "resource-explorer-2:GetView",
            ],
            "Resource": "arn:aws:resource-explorer-2:us-
east-1:123456789012:view/policy-name/",
            "Condition": {"StringEquals": {"aws:PrincipalOrgID":
 "o-123456789012"},
                "StringNotEquals": {"aws:PrincipalAccount": "123456789012"}
            }
        }
    ]
    }"
}
```

The administrator in each of the specified accounts must now specify which roles and users can access the view by attaching identity-based permissions policies to the roles, groups, and users. The administrators of accounts 111122223333 or 444455556666 can create the following example policy. Then, they can assign the policy to roles, groups, and users in those accounts who are to be allowed to search using the view shared from the originating account.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                ":Search",
                ":GetView",
            "Resource": "arn:aws:resource-explorer-2:us-
east-1:123456789012:view/policy-name/"
        }
    ]
}
```

You can use these IAM identity-based policies as part of an attribute-based access control (ABAC) security strategy. In that paradigm, you make sure that all of your resources and all of your identities are tagged. Then, you specify in your policies which tag keys and values must match between the identity and the resource for access to be allowed. For information about tagging the

views in your account, see Adding tags to views. For more information about attribute-based access control, see What is ABAC for AWS? and Controlling access to AWS resources using tags, both in the *IAM User Guide*.

# Deleting views in Resource Explorer

When you no longer need an AWS Resource Explorer view, you can delete it. You can delete views by using the AWS Management Console or by running AWS CLI commands or their equivalent API operations in an AWS SDK.

> **Note**
>
> You can't delete a view that is currently designated as the default for its AWS Region. To delete the view, you must remove the view as the default. To do this, you can run the DisassociateDefaultView API operation in that Region.

### Minimum permissions

To run this procedure, you must have the following permissions:

- **Action:** `resource-explorer-2:DeleteView`

  **Resource:** The ARN of the view to delete

AWS Management Console

**To delete a view**

1. On the Resource Explorer console **Views** page, choose the option button next to the view that you want to delete.
2. Choose **Actions**, and then choose **Delete**.
3. In the confirmation dialog box, type the name of the view, and then choose **Delete**.

AWS CLI

**To delete a view**

Run the following command to delete the view with the specified Amazon Resource Name (ARN).

```
$ aws resource-explorer-2 delete-view \
    --view-arn arn:aws:resource-explorer-2:us-east-1:123456789012:view/
MyViewName/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
{
    "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/
MyViewName/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
}
```

# AWS managed views

A *managed view* is how other AWS services can access resource information indexed by Resource Explorer for your AWS account or organization with your consent.

**Topics**


- About managed views
- Listing managed views
- Deleting managed views


## About managed views

Managed views can only be updated or deleted by the service that created the managed view. An AWS service creates a managed view using IAM forward access sessions (FAS) or a service-linked role (SLR).

Resource Explorer uses a resource-based policy to control access to the managed view. When an AWS service creates a managed view, Resource Explorer attaches the resource-based policy to the view. This policy allows the managing AWS service to use and delete the view and allows view's resource owners to list and retrieve details about the view. The following is an example resource-based policy attached to a managed view:

JSON

```
{
  Version:"2012-10-17",
  Statement:[
    {
      Sid:"view_UUID_ACCESS_TO_SERVICE_PRINCIPAL",
      Effect:"Allow",
      Principal:{
        Service:"sampleservice.amazonaws.com"
      },
      Action:[
        "resource-explorer-2:GetManagedView",
        "resource-explorer-2:DeleteManagedView",
        "resource-explorer-2:Search"
      ],
```

```
      Resource:"managed_view_ARN",
      Condition:{
        StringEquals:{
          'aws:SourceAccount':"owner_accountID"
        }
      }
    },
    {
      Sid:"view_UUID_DENY_ACCESS_TO_NON_SERVICE_PRINCIPAL",
      Effect:"Deny",
      Principal:"*",
      Condition:{
        'ForAllValues:StringNotEquals':{
          'aws:PrincipalServiceNamesList':[
            "sampleservice.amazonaws.com"
          ]
        }
      },
      NotAction:[
        "resource-explorer-2:GetManagedView"
      ],
      Resource:"managed_view_ARN"
    }
  ]
}
```

# Listing managed views

You can see which managed views you have access to on the **Views** page in the Resource Explorer console. You can also run AWS CLI commands or their equivalent API operations in an AWS SDK to list the managed views you have access to in your currently selected AWS Region and retrieve view details.

To run these commands, you must have the following permissions:

- **Action**: `resource-explorer-2:GetManagedView`

  **Resource**: The ARN of the specified view.
- **Action**: `resource-explorer-2:ListManagedViews`

  **Resource**: The ARN of the specified view.

**To list your available managed views**

Run the following command to list managed views in the specified AWS Region:

```
aws resource-explorer-2 list-managed-views  --region region
```

The command output is a list of ARNs.

```
{
  "ManagedViews": [
    "arn:aws:resource-explorer-2:us-east-1:111122223333:managed-view/
ManagedViewNameA/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "arn:aws:resource-explorer-2:us-east-1:444455556666:managed-view/
ManagedViewNameB/1a2b3c4d-5d6e-7f8a-9b0c-abcd22222222"
  ]
}
```

**To retrieve managed view details**

Run the following command to retrieve details about a specified managed view using the view's ARN:

```
aws resource-explorer-2 get-managed-view \
    --managed-view-arn arn:aws:resource-explorer-2:us-east-1:111122223333:managed-view/
ManagedViewNameA/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
```

The command output is details about the specified managed view.

```
{
  "ManagedView": {
    "ManagedViewArn": "arn:aws:resource-explorer-2:us-east-1:111122223333:managed-view/
ManagedViewNameA/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "ManagedViewName": "ManagedViewNameA",
    "TrustedService": "sampleservice.amazonaws.com",
    "LastUpdatedAt": "2024-01-01T01:01:01.100000+00:00",
    "Owner": "111111111111",
    "Scope": "arn:aws:iam::111111111111:root",
    "Filters": {
      "FilterString": ""
    },
    "IncludedProperties": [
```

```
      {
        "Name": "tags"
      }
    ],
    "ResourcePolicy": "{\"Version\":\"2012-10-17\",\"Statement\":[{\"Sid\":
\"EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111_ACCESS_TO_SERVICE_PRINCIPAL\",\"Effect\":
\"Allow\",\"PrincipalGroup\":{\"AWS\":\"sservicea.amazonaws.com\"},\"Action\":
[\"resource-explorer-2:GetManagedView\",\"resource-explorer-2:DeleteManagedView
\",\"resource-explorer-2:Search\"],\"Resource\":\"arn:aws:resource-
explorer-2:us-east-1:111122223333:managed-view/ExampleManagedViewName/
EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111\",\"Condition\":{\"StringEquals\":
{\"aws:SourceAccount\":\"111122223333\"}}},{\"Sid\":\"EXAMPLE8-90ab-cdef-
fedc-EXAMPLE11111_DENY_ACCESS_TO_NON_SERVICE_PRINCIPAL\",\"Effect\":\"Deny
\",\"Principal\":\"*\",\"NotAction\":\"resource-explorer-2:GetManagedView\",
\"Resource\":\"arn:aws:resource-explorer-2:us-east-1:111122223333:managed-
view/ExampleManagedViewName/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111\",\"Condition
\":{\"ForAllValues:StringNotEquals\":{\"aws:PrincipalServiceNamesList\":
\"servicea.amazonaws.com\"}}}]}",
    "Version": "1"
  }
}
```

# Deleting managed views

Managed views can only be deleted by the AWS service that manages them. Before the managing service can delete the view, you may need to perform service-specific tasks to remove a managed view from your account.

Resource Explorer managed views use the AWS Systems Manager `AWSManagedViewForSSM` unified console resource, which allows Systems Manager to access resource information indexed by Resource Explorer for your organization. If you want to delete the managed view, you must disable the unified console in Systems Manager. For instructions, see Disabling the Systems Manager unified console in the *AWS Systems Manager User Guide*.

# Resource types you can search for with Resource Explorer

Resource Explorer supports resource types across numerous AWS services.

**Topics**


- Supported services and resource types
- Programmatically accessing the list of supported resource types
- Resource types that appear as other types


Some resource types are identified by Amazon resource name (ARN) strings that share a common format with another resource type. When this happens, Resource Explorer can report such resources as that other resource type. For list of resource types affected by this issue, see Resource types that appear as other types.

At this time, tags attached to AWS Identity and Access Management (IAM) resources, such as roles or users, can't be used for searching.

If you have encrypted access to some of your resources, Resource Explorer is unable to discover them. You will not see these resources in your search results.

The following tables list the resource types that are supported for searching in AWS Resource Explorer.

**Note**

As of July 9, 2024, Resource Explorer no longer supports the following resource types:

- **Amazon Elastic Container Service** — `ecs:task`
- **AWS Systems Manager** — `ssm:automation-execution`
- **AWS Systems Manager** — `ssm:patchbaseline`

You can still use these resource types in their own services, but they are no longer indexed or searchable in Resource Explorer.

# Supported services and resource types

### Supported AWS services

- Amazon API Gateway
- AWS Amplify
- AWS App Runner
- AWS AppConfig
- Amazon AppFlow
- AppIntegrations
- AWS App Mesh
- Amazon AppStream
- AWS AppSync
- AWS Application Discovery Service
- Amazon Application Recovery Controller (ARC)
- Amazon Athena
- AWS Audit Manager
- AWS Backup
- AWS Batch
- AWS Certificate Manager
- AWS Cloud Map
- AWS Cloud9
- AWS CloudFormation
- Amazon CloudFront
- AWS CloudTrail
- Amazon CloudWatch
- Amazon CloudWatch Evidently
- Amazon CloudWatch Logs
- Amazon CloudWatch Observability Access Manager
- Amazon CloudWatch RUM

- Amazon CloudWatch Synthetics
- AWS CodeArtifact
- AWS CodeBuild
- AWS CodeCommit
- AWS CodeConnections
- AWS CodeDeploy
- Amazon CodeGuru Profiler
- Amazon CodeGuru Reviewer
- AWS CodePipeline
- AWS CodeStar Connections
- Amazon Cognito Identity
- Amazon Cognito IdentityPool
- Amazon Comprehend
- Amazon Connect
- Amazon Connect Wisdom
- AWS Cost Explorer
- AWS Data Exchange
- AWS Data Pipeline
- AWS DataSync
- AWS Database Migration Service
- Amazon Detective
- AWS Device Farm
- Amazon DynamoDB
- DynamoDB Accelerator
- EC2 Image Builder
- Amazon EMR
- Amazon EMR Serverless
- Amazon EMR on EKS

- Amazon ElastiCache
- AWS Elastic Beanstalk
- Amazon Elastic Compute Cloud (Amazon EC2)
- Amazon Elastic Container Registry
- Amazon Elastic Container Registry Public
- Amazon Elastic Container Service
- Amazon Elastic File System
- Amazon Elastic Kubernetes Service (Amazon EKS)
- Elastic Load Balancing
- AWS Elemental MediaPackage
- AWS Elemental MediaPackage VoD
- AWS Elemental MediaTailor
- Amazon CloudWatch Events
- Amazon EventBridge Pipes
- Amazon EventBridge Scheduler
- Amazon EventBridge Schemas
- Amazon FSx
- AWS Fault Injection Service
- Amazon FinSpace
- Firehose
- Amazon Forecast
- Amazon Fraud Detector
- Amazon GameLift Servers
- AWS Global Accelerator
- AWS Glue
- AWS Glue DataBrew
- AWS Ground Station
- Amazon GuardDuty
- AWS HealthLake

- AWS HealthOmics
- IAM Access Analyzer
- Amazon IVS
- AWS Identity and Access Management
- Amazon Inspector
- Amazon Interactive Video Service
- AWS IoT
- AWS IoT Analytics
- AWS IoT Core Device Advisor
- AWS IoT Events
- AWS IoT FleetWise
- AWS IoT Greengrass
- AWS IoT SiteWise
- AWS IoT TwinMaker
- AWS IoT Wireless
- Amazon Kendra
- AWS Key Management Service
- Amazon Kinesis
- Amazon Managed Service for Apache Flink
- Amazon Kinesis Video Streams
- AWS Lambda
- Amazon Lex
- Amazon Location Service
- Amazon Lookout for Metrics
- Amazon Lookout for Vision
- Amazon MQ
- AWS Mainframe Modernization
- Amazon Managed Blockchain
- Amazon Managed Grafana

- Amazon Managed Service for Prometheus
- Amazon Managed Streaming for Apache Kafka
- Amazon Managed Workflows for Apache Airflow
- Amazon MemoryDB
- AWS Migration Hub Refactor Spaces
- AWS Mobile Targeting
- AWS Network Firewall
- AWS Network Manager
- Amazon OpenSearch Service
- AWS Outposts
- AWS Panorama
- Amazon Personalize
- AWS Private Certificate Authority
- AWS Proton
- Amazon Quantum Ledger Database (Amazon QLDB)
- Amazon QuickSight
- Amazon Redshift
- Amazon Rekognition
- Amazon Relational Database Service (Amazon RDS)
- AWS Resilience Hub
- AWS Resource Access Manager
- AWS Resource Groups
- AWS Resource Explorer
- Amazon Route 53
- Amazon Route 53 Recovery Readiness
- Amazon Route 53 Resolver
- Amazon S3 Glacier
- Amazon SageMaker AI
- AWS Secrets Manager

- AWS Service Catalog
- AWS Signer
- Amazon Simple Email Service
- Amazon Simple Notification Service
- Amazon Simple Queue Service
- Amazon Simple Storage Service (Amazon S3)
- AWS Step Functions States Language
- Storage Gateway
- AWS Systems Manager
- AWS Transfer Family
- Amazon WorkSpaces

## Amazon API Gateway

- `apigateway:restapis`
- `apigateway:vpclinks`

## AWS Amplify

- `amplify:apps/branches`
- `amplify:apps/domains`

## AWS App Runner

- `apprunner:service`
- `apprunner:vpcconnector`

## AWS AppConfig

- `appconfig:application`
- `appconfig:deploymentstrategy`

## Amazon AppFlow

- appflow:flow

## AppIntegrations

- app-integrations:event-integration

## AWS App Mesh

- appmesh:mesh
- appmesh:mesh/virtualNode
- appmesh:mesh/virtualService

## Amazon AppStream

- appstream:app-block
- appstream:application
- appstream:fleet
- appstream:image-builder
- appstream:stack

## AWS AppSync

- appsync:apis

## AWS Application Discovery Service

- ds:directory

## Amazon Application Recovery Controller (ARC)

- route53-recovery-control:cluster

- route53-recovery-control:controlpanel/safetyrule

## Amazon Athena

- athena:datacatalog
- athena:workgroup

## AWS Audit Manager

- auditmanager:assessment

## AWS Backup

- backup:backup-plan
- backup:backup-vault
- backup:report-plan

## AWS Batch

- batch:compute-environment
- batch:job-queue
- batch:scheduling-policy

## AWS Certificate Manager

- acm:certificate

## AWS Cloud Map

- servicediscovery:service

## AWS Cloud9

- cloud9:environment

## AWS CloudFormation

- cloudformation:stack
- cloudformation:stackset

## Amazon CloudFront

- cloudfront:cache-policy
- cloudfront:continuous-deployment-policy
- cloudfront:distribution
- cloudfront:field-level-encryption-config
- cloudfront:field-level-encryption-profile
- cloudfront:function
- cloudfront:origin-access-control
- cloudfront:origin-access-identity
- cloudfront:origin-request-policy
- cloudfront:realtime-log-config
- cloudfront:response-headers-policy

## AWS CloudTrail

- cloudtrail:channel
- cloudtrail:eventdatastore
- cloudtrail:trail

## Amazon CloudWatch

- cloudwatch:alarm

- cloudwatch:dashboard
- cloudwatch:insight-rule
- cloudwatch:metric-stream

## Amazon CloudWatch Evidently

- evidently:project
- evidently:project/experiment
- evidently:project/feature
- evidently:project/launch

## Amazon CloudWatch Logs

- logs:destination
- logs:log-group

## Amazon CloudWatch Observability Access Manager

- oam:sink

## Amazon CloudWatch RUM

- rum:appmonitor

## Amazon CloudWatch Synthetics

- synthetics:group

## AWS CodeArtifact

- codeartifact:domain
- codeartifact:repository

## AWS CodeBuild

- codebuild:project

## AWS CodeCommit

- codecommit:repository

## AWS CodeConnections

- codeconnections:connection

## AWS CodeDeploy

- codedeploy:application
- codedeploy:deploymentconfig

## Amazon CodeGuru Profiler

- codeguru-profiler:profilingGroup

## Amazon CodeGuru Reviewer

- codeguru-reviewer:association

## AWS CodePipeline

- codepipeline:pipeline
- codepipeline:webhook

## AWS CodeStar Connections

- codestar-connections:connection

## Amazon Cognito Identity

- cognito-identity:identitypool

## Amazon Cognito IdentityPool

- cognito-idp:userpool

## Amazon Comprehend

- comprehend:document-classifier
- comprehend:entity-recognizer

## Amazon Connect

- connect:instance
- connect:instance/agent
- connect:instance/rule
- connect:instance/task-template
- connect:instance/transfer-destination
- connect:phone-number

## Amazon Connect Wisdom

- wisdom:assistant
- wisdom:association
- wisdom:knowledge-base

## AWS Cost Explorer

- ce:anomalymonitor
- ce:anomalysubscription

## AWS Data Exchange

- dataexchange:data-sets

## AWS Data Pipeline

- datapipeline:pipeline

## AWS DataSync

- datasync:location
- datasync:task

## AWS Database Migration Service

- dms:endpoint
- dms:es
- dms:rep
- dms:subgrp
- dms:task

## Amazon Detective

- detective:graph

## AWS Device Farm

- devicefarm:project
- devicefarm:testgrid-project

## Amazon DynamoDB

- dynamodb:table

## DynamoDB Accelerator

- dax:cache

## EC2 Image Builder

- imagebuilder:component
- imagebuilder:container-recipe
- imagebuilder:distribution-configuration
- imagebuilder:image
- imagebuilder:image-pipeline
- imagebuilder:image-recipe
- imagebuilder:infrastructure-configuration

## Amazon EMR

- elasticmapreduce:cluster

## Amazon EMR Serverless

- emr-serverless:applications

## Amazon EMR on EKS

- emr-containers:virtualclusters

## Amazon ElastiCache

- elasticache:cluster
- elasticache:globalreplicationgroup
- elasticache:parametergroup
- elasticache:replicationgroup
- elasticache:reserved-instance

- elasticache:snapshot
- elasticache:subnetgroup
- elasticache:user
- elasticache:usergroup

## AWS Elastic Beanstalk

- elasticbeanstalk:application
- elasticbeanstalk:applicationversion
- elasticbeanstalk:configurationtemplate
- elasticbeanstalk:environment

## Amazon Elastic Compute Cloud (Amazon EC2)

- ec2:capacity-reservation
- ec2:capacity-reservation-fleet
- ec2:carrier-gateway
- ec2:client-vpn-endpoint
- ec2:customer-gateway
- ec2:dedicated-host
- ec2:dhcp-options
- ec2:egress-only-internet-gateway
- ec2:elastic-ip
- ec2:fleet
- ec2:fpga-image
- ec2:host-reservation
- ec2:image
- ec2:instance
- ec2:instance-event-window
- ec2:internet-gateway

- ec2:ipam
- ec2:ipam-pool
- ec2:ipam-resource-discovery
- ec2:ipam-resource-discovery-association
- ec2:ipam-scope
- ec2:ipv4pool-ec2
- ec2:key-pair
- ec2:launch-template
- ec2:natgateway
- ec2:network-acl
- ec2:network-insights-access-scope
- ec2:network-insights-access-scope-analysis
- ec2:network-insights-analysis
- ec2:network-insights-path
- ec2:network-interface
- ec2:placement-group
- ec2:prefix-list
- ec2:reserved-instances
- ec2:route-table
- ec2:security-group
- ec2:security-group-rule
- ec2:snapshot
- ec2:spot-fleet-request
- ec2:spot-instances-request
- ec2:subnet
- ec2:subnet-cidr-reservation
- ec2:traffic-mirror-filter
- ec2:traffic-mirror-filter-rule
- ec2:traffic-mirror-session

- ec2:traffic-mirror-target
- ec2:transit-gateway
- ec2:transit-gateway-attachment
- ec2:transit-gateway-connect-peer
- ec2:transit-gateway-multicast-domain
- ec2:transit-gateway-policy-table
- ec2:transit-gateway-route-table
- ec2:transit-gateway-route-table-announcement
- ec2:verified-access-endpoint
- ec2:verified-access-group
- ec2:verified-access-instance
- ec2:verified-access-trust-provider
- ec2:volume
- ec2:vpc
- ec2:vpc-endpoint
- ec2:vpc-flow-log
- ec2:vpc-peering-connection
- ec2:vpn-connection
- ec2:vpn-gateway

## Amazon Elastic Container Registry

- ecr:repository

## Amazon Elastic Container Registry Public

- ecr-public:repository

## Amazon Elastic Container Service

- ecs:capacity-provider

- ecs:cluster
- ecs:container-instance
- ecs:service
- ecs:task-definition
- ecs:task-set

## Amazon Elastic File System

- elasticfilesystem:access-point
- elasticfilesystem:file-system

## Amazon Elastic Kubernetes Service (Amazon EKS)

- eks:cluster

## Elastic Load Balancing

- elasticloadbalancing:listener
- elasticloadbalancing:listener-rule
- elasticloadbalancing:listener-rule/app
- elasticloadbalancing:listener-rule/net
- elasticloadbalancing:listener/app
- elasticloadbalancing:listener/net
- elasticloadbalancing:loadbalancer
- elasticloadbalancing:loadbalancer/app
- elasticloadbalancing:loadbalancer/net
- elasticloadbalancing:targetgroup

## AWS Elemental MediaPackage

- mediapackage:channels
- mediapackage:origin_endpoints

## AWS Elemental MediaPackage VoD

- mediapackage-vod:packaging-configurations
- mediapackage-vod:packaging-groups

## AWS Elemental MediaTailor

- mediatailor:playbackConfiguration

## Amazon CloudWatch Events

- events:archive
- events:connection
- events:endpoint
- events:event-bus
- events:rule

## Amazon EventBridge Pipes

- pipes:pipe

## Amazon EventBridge Scheduler

- scheduler:schedule-group

## Amazon EventBridge Schemas

- schemas:discoverer

## Amazon FSx

- fsx:file-system

## AWS Fault Injection Service

- fis:experiment-template

## Amazon FinSpace

- finspace:environment

## Firehose

- firehose:deliverystream

## Amazon Forecast

- forecast:dataset
- forecast:dataset-group
- forecast:dataset-import-job
- forecast:forecast
- forecast:forecast-export-job
- forecast:predictor
- forecast:predictor-backtest-export-job

## Amazon Fraud Detector

- frauddetector:detector
- frauddetector:entity-type
- frauddetector:event-type
- frauddetector:external-model
- frauddetector:label
- frauddetector:model
- frauddetector:outcome
- frauddetector:variable

## Amazon GameLift Servers

- gamelift:alias
- gamelift:build
- gamelift:gamesessionqueue
- gamelift:location
- gamelift:matchmakingconfiguration
- gamelift:matchmakingruleset

## AWS Global Accelerator

- globalaccelerator:accelerator
- globalaccelerator:accelerator/listener
- globalaccelerator:accelerator/listener/endpoint-group

## AWS Glue

- glue:crawler
- glue:database
- glue:job
- glue:mlTransform
- glue:table
- glue:trigger

## AWS Glue DataBrew

- databrew:dataset
- databrew:job
- databrew:project
- databrew:recipe
- databrew:ruleset
- databrew:schedule

## AWS Ground Station

- groundstation:config
- groundstation:dataflow-endpoint-group
- groundstation:mission-profile

## Amazon GuardDuty

- guardduty:detector/filter
- guardduty:detector/ipset
- guardduty:detector/threatintelset

## AWS HealthLake

- healthlake:datastore/fhir

## AWS HealthOmics

- omics:referenceStore
- omics:runGroup
- omics:workflow

## IAM Access Analyzer

- access-analyzer:analyzer

## Amazon IVS

- ivschat:logging-configuration
- ivschat:room

## AWS Identity and Access Management

- iam:group
- iam:instance-profile
- iam:mfa
- iam:oidc-provider
- iam:policy
- iam:role
- iam:saml-provider
- iam:server-certificate
- iam:user

## Amazon Inspector

- inspector:target/template

## Amazon Interactive Video Service

- ivs:channel
- ivs:recording-configuration
- ivs:stream-key

## AWS IoT

- iot:authorizer
- iot:cacert
- iot:cert
- iot:jobtemplate
- iot:mitigationaction
- iot:policy
- iot:provisioningtemplate
- iot:rolealias

- iot:rule
- iot:ruledestination
- iot:securityprofile
- iot:thing

## AWS IoT Analytics

- iotanalytics:channel
- iotanalytics:dataset
- iotanalytics:datastore
- iotanalytics:pipeline

## AWS IoT Core Device Advisor

- iotdeviceadvisor:suitedefinition

## AWS IoT Events

- iotevents:alarmModel
- iotevents:detectorModel
- iotevents:input

## AWS IoT FleetWise

- iotfleetwise:decoder-manifest
- iotfleetwise:model-manifest
- iotfleetwise:signal-catalog
- iotfleetwise:vehicle

## AWS IoT Greengrass

- greengrass:components:versions

- greengrass:connectorsDefinition
- greengrass:coresDefinition
- greengrass:devicesDefinition
- greengrass:functionsDefinition
- greengrass:groups
- greengrass:loggersDefinition
- greengrass:resourcesDefinition
- greengrass:subscriptionsDefinition

## AWS IoT SiteWise

- iotsitewise:asset
- iotsitewise:asset-model
- iotsitewise:dashboard
- iotsitewise:gateway

## AWS IoT TwinMaker

- iottwinmaker:workspace
- iottwinmaker:workspace/component-type
- iottwinmaker:workspace/entity

## AWS IoT Wireless

- iotwireless:ServiceProfile

## Amazon Kendra

- kendra:index

## AWS Key Management Service

- kms:key

## Amazon Kinesis

- kinesis:stream

## Amazon Managed Service for Apache Flink

- kinesisanalytics:application

## Amazon Kinesis Video Streams

- kinesisvideo:stream

## AWS Lambda

- lambda:code-signing-config
- lambda:event-source-mapping
- lambda:function

## Amazon Lex

- lex:bot-alias

## Amazon Location Service

- geo:map
- geo:place-index
- geo:tracker

## Amazon Lookout for Metrics

- lookoutmetrics:Alert
- lookoutmetrics:AnomalyDetector

## Amazon Lookout for Vision

- lookoutvision:project

## Amazon MQ

- mq:broker

## AWS Mainframe Modernization

- m2:env

## Amazon Managed Blockchain

- managedblockchain:accessors

## Amazon Managed Grafana

- grafana:workspaces

## Amazon Managed Service for Prometheus

- aps:rulegroupsnamespace
- aps:workspace

## Amazon Managed Streaming for Apache Kafka

- kafka:cluster

- kafka:configuration

## Amazon Managed Workflows for Apache Airflow

- airflow:environment

## Amazon MemoryDB

- memorydb:acl
- memorydb:cluster
- memorydb:parametergroup
- memorydb:subnetgroup
- memorydb:user

## AWS Migration Hub Refactor Spaces

- refactor-spaces:environment
- refactor-spaces:environment/application
- refactor-spaces:environment/application/route
- refactor-spaces:environment/application/service

## AWS Mobile Targeting

- mobiletargeting:apps/campaigns
- mobiletargeting:apps/segments

## AWS Network Firewall

- network-firewall:firewall
- network-firewall:firewall-policy

## AWS Network Manager

- networkmanager:attachment
- networkmanager:core-network
- networkmanager:device
- networkmanager:global-network
- networkmanager:link

## Amazon OpenSearch Service

- es:domain

## AWS Outposts

- outposts:site

## AWS Panorama

- panorama:package

## Amazon Personalize

- personalize:dataset
- personalize:dataset-group
- personalize:schema
- personalize:solution

## AWS Private Certificate Authority

- acm-pca:certificate-authority

## AWS Proton

- proton:environment-account-connection

## Amazon Quantum Ledger Database (Amazon QLDB)

- qldb:ledger
- qldb:stream

## Amazon QuickSight

- quicksight:dataset
- quicksight:datasource
- quicksight:template
- quicksight:theme

## Amazon Redshift

- redshift:cluster
- redshift:eventsubscription
- redshift:parametergroup
- redshift:snapshot
- redshift:snapshotcopygrant
- redshift:snapshotschedule
- redshift:subnetgroup
- redshift:usagelimit

## Amazon Rekognition

- rekognition:project

## Amazon Relational Database Service (Amazon RDS)

- rds:auto-backup
- rds:cev
- rds:cluster
- rds:cluster-endpoint
- rds:cluster-pg
- rds:cluster-snapshot
- rds:db
- rds:db-proxy
- rds:db-proxy-endpoint
- rds:deployment
- rds:es
- rds:global-cluster
- rds:og
- rds:pg
- rds:ri
- rds:secgrp
- rds:snapshot
- rds:subgrp

## AWS Resilience Hub

- resiliencehub:resiliency-policy

## AWS Resource Access Manager

- ram:resource-share

## AWS Resource Groups

- resource-groups:group

## AWS Resource Explorer

- resource-explorer-2:index
- resource-explorer-2:view

## Amazon Route 53

- route53:domain
- route53:healthcheck
- route53:hostedzone

## Amazon Route 53 Recovery Readiness

- route53-recovery-readiness:readiness-check
- route53-recovery-readiness:recovery-group
- route53-recovery-readiness:resource-set

## Amazon Route 53 Resolver

- route53resolver:firewall-domain-list
- route53resolver:firewall-rule-group
- route53resolver:firewall-rule-group-association
- route53resolver:resolver-endpoint
- route53resolver:resolver-query-log-config
- route53resolver:resolver-rule

## Amazon S3 Glacier

- glacier:vaults

## Amazon SageMaker AI

- sagemaker:app-image-config

- sagemaker:domain
- sagemaker:endpoint
- sagemaker:feature-group
- sagemaker:image
- sagemaker:model
- sagemaker:notebook-instance
- sagemaker:pipeline

## AWS Secrets Manager

- secretsmanager:secret

## AWS Service Catalog

- servicecatalog:applications
- servicecatalog:attribute-groups

## AWS Signer

- signer:signing-profiles

## Amazon Simple Email Service

- ses:configuration-set
- ses:contact-list
- ses:identity

## Amazon Simple Notification Service

- sns:topic

## Amazon Simple Queue Service

- sqs:queue

## Amazon Simple Storage Service (Amazon S3)

- s3:accesspoint
- s3:bucket
- s3:storage-lens

## AWS Step Functions States Language

- states:activity
- states:stateMachine

## Storage Gateway

- storagegateway:gateway

## AWS Systems Manager

- ssm:association
- ssm:document
- ssm:maintenancewindow
- ssm:managed-instance
- ssm:parameter
- ssm:resource-data-sync
- ssm:windowtarget
- ssm:windowtask

## AWS Transfer Family

- transfer:agreement

- transfer:certificate
- transfer:connector
- transfer:profile
- transfer:workflow

### Amazon WorkSpaces

- workspaces:connectionalias
- workspaces:workspace

## Programmatically accessing the list of supported resource types

To access the list of supported resource types from code, you can invoke the ListSupportedResourceTypes operation from any AWS SDK.

For example, you can run the list-supported-resource-types AWS Command Line Interface (AWS CLI) command, as shown in the following example.

```
$ aws resource-explorer-2 list-supported-resource-types
{
    "ResourceTypes": [
        {
            "ResourceType": "acm-pca:certificate-authority",
            "Service": "acm-pca"
        },
        {
            "ResourceType": "airflow:environment",
            "Service": "airflow"
        },
        {
            "ResourceType": "amplify:branches",
            "Service": "amplify"
        },
... truncated for brevity ...
```

# Resource types that appear as other types

Some resource types are identified by Amazon resource name (ARN) strings that share a common format with another resource type. When this happens, Resource Explorer can report such resources as that other resource type. This affects the resource types in the following table.

| Actual resource type | Reported as resource type |
| --- | --- |
| `ec2:securitygroupegress`<br>`ec2:securitygroupingress` | `ec2:security-group-rule` |
| `elasticloadbalancingv2:loadbalancer` | `elasticloadbalancing:loadbalancer` |
| `docdb:dbcluster`<br>`neptune:dbcluster`<br>`rds:dbcluster` | `rds:cluster` |
| `docdb:dbclusterparametergroup`<br>`neptune:dbclusterparametergroup`<br>`rds:dbclusterparametergroup` | `rds:cluster-pg` |
| `docdb:clustersnapshot`<br>`neptune:dbclustersnapshot`<br>`rds:clustersnapshot` | `rds:cluster-snapshot` |
| `docdb:dbinstance`<br>`neptune:dbinstance`<br>`rds:dbinstance` | `rds:db` |
| `docdb:eventsubscription`<br>`neptune:eventsubscription` | `rds:es` |

| Actual resource type | Reported as resource type |
| --- | --- |
| rds:eventsubscription | |
| docdb:globalcluster<br><br>rds:globalcluster | rds:global-cluster |
| neptune:dbparametergroup<br><br>rds:dbparametergroup | rds:pg |
| docdb:dbsubnetgroup<br><br>neptune:dbsubnetgroup<br><br>rds:dbsubnetgroup | rds:subgrp |

# Using AWS Resource Explorer to search for resources

The primary purpose of enabling AWS Resource Explorer in your AWS account is to allow your users to search for resources in the account, and to use the Resource Explorer console to quickly act on those resources. You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to search for resources using Resource Explorer.

The following are some of the main characteristics of Resource Explorer search.

- **Every search must use a view.**

  The view is what Resource Explorer uses to determine who has permissions to see which resources. To use a view in a Resource Explorer search operation, the user must have an `Allow` on the `resource-explorer-2:Search` operation for the specified view. This permission comes from an identity-based permission policy attached to the principal making the request.

  The view can include a filter that limits which resources can be included in the results. By creating different views that use filters and by granting different principals access to different views, you can configure an environment where each group of users can view only the resources relevant to them.

  For more information about views, see Configuring an Resource Explorer view to provide access to resource searches.
- **Resource Explorer uses asynchronous background processes to maintain its indexes.**

  It can take Resource Explorer some time for its indexing processes to discover newly created or modified resources and add them to the local index. It can take additional time for Resource Explorer to replicate changes in the local indexes to the aggregator index.

  The same applies to resources that you delete. It can take some time after you delete a resource for that deletion to be discovered by the indexing process and that resource's information to be removed from the local index. Additional time is needed for Resource Explorer to replicate that deletion from the local index to the account's aggregator index.

  Most resource modifications and deletions are visible in search results within minutes in all Regions where you've activated Resource Explorer. In some cases, modifications or deletions may take up to two weeks to be visible.
- **A search in Resource Explorer occurs within an AWS Region.**

Each Region where you turn on Resource Explorer contains an index of only the resources stored in that Region. Views are also associated with Regions, and can return only the resources found in that Region's index. The one exception to this is the aggregator index, that receives a replicated copy of all of the local indexes to support searching across all Regions in the account.

- **Cross-Region search requires an aggregator index for the account.**

  To let users search for resources across all AWS Regions, the administrator must designate one Region to contain the aggregator index for the account. A copy of every local index is automatically replicated to the aggregator index.

  Because of this, only views in the aggregator index Region can return results that include resources from all AWS Regions in the account.

- **A query consists of any number of free-form text keywords and filters.**

  Free-form keywords are combined in the query using logical **OR** operators. Filters that use Resource Explorer defined filter names are combined in the query using logical **AND** operators. Consider the following example query.

```
test instance service:EC2 region:us-west-2
```

  This is evaluated by Resource Explorer as follows.

```
test OR instance AND service:EC2 AND region:us-west-2
```

  This query requires that matching resources must be Amazon EC2 resources in the US West (Oregon) Region, and have at least one of the keywords (*test*, *instance*) attached in some way, such as in the name, description, or tags.

> **Note**
>
> Because of the implicit AND, you can successfully use only one filter for an attribute that can have only one value associated with the resource. For example, a resource can be part of only one AWS Region. Therefore, the following query returns no results.
>
> ```
> region:us-east-1 region:us-west-1
> ```

This limitation does ***not*** apply to the filters for attributes that can have multiple values at the same time, such as `tag:`, `tag.key:`, and `tag.value:`.

- **A search can return only the first 1,000 results if you include free-form text.**

  If your query includes free-form text, Resource Explorer uses the `Search` API operation, but if your query does not include free-form text, Resource Explorer uses the `ListResources` operation. `Search` operations are limited to 1,000 results that are sorted by relevancy, while the `ListResources` operation has no upper limit and are *not* sorted by relevancy. To view query resources beyond 1,000 results when using free-form text (the `Search` operation), you must use additional filters to restrict matching results to those you want to see.

- **There is a per-account quota on the number of search operations that you can perform.**

  Quotas limit how many queries you can make per second, and how many queries you can make each month. For specific quota numbers, see Quotas for Resource Explorer. Quota usage depends on if Resource Explorer performs resource queries using the `Search` or `ListResources` operations on your behalf based on the logic described in the previous list item.

AWS Management Console

**To search for resources using Resource Explorer**

1. On the **Resource search** page, start by choosing the view that you want to use. You can choose from among only those views that you have permissions to access.
2. (Optional) Choose a **Query template**.
   a. For templates that require a specified resource type or application, **choose a value**.
   b. Choose **Apply**.
3. (Optional) In the **Quick filters** menu, choose one or more filters to apply to the search query.
4. (Optional) For **Query**, enter the search terms and filters that identify the resources you want to see. For information about all of the available syntax options, see Search query syntax reference for Resource Explorer.
5. Resource Explorer displays all of the results that match both the `Filter` defined in the view and the **Query** that you provide. If your query includes free-form text, the results are

sorted by relevance, with those resources that match more of your query terms appearing higher in the list and resources that match fewer terms appearing further down the list.

6. You can view details about the selected resource from within Resource Explorer by selecting the checkbox in the table.

   Alternatively, you can choose the identifier of a resource to navigate to that resource type's native console, where you can interact with the resource in all of the ways supported by that AWS service.

After submitting your search query, Resource Explorer displays a results table. You can use the AWS CLI

**To search for resources using Resource Explorer**

Run the following command to search for resources using the specified view. That view must exist in the Region in which you run the operation. The following example searches for Amazon EC2 instances that are tagged `env=production` in the US East (Ohio) (us-east-2). For information about all of the available syntax options for the `query-string` parameter, see Search query syntax reference for Resource Explorer.

```
$ aws resource-explorer-2 search \
    --region us-east-1 \
    --query-string "resourcetype:AWS::EC2::Instance tag:env=production"
    --view-arn arn:aws:resource-explorer-2:us-east-2:123456789012:view/My-Resources-View/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
```

# Quick filters

The Resource Explorer console provides Quick filters so you can quickly and easily apply filters like Region, Resource type, or Tag keys and values to your resource query. You can use Quick filters independently, or in addition to the free-form keyword and defined filter query.

The Quick filters menu only displays category filter values matching loaded resource data. For accounts with more than 1,000 resources, you can choose **Load more** at the top of the Resources table to view additional resources and filter values.

For example, by default the Region category displays five Regions for the first 1,000 loaded resources. After you load more data, the Region category displays a total of 12 Regions across 2,000 resources.

## Search query templates

The Resource Explorer console provides search query templates, which are predefined query configurations for common queries. Query templates allow you to quickly perform a search and better understand how to customize your own queries. For some templates, you must specify the desired resource type or application in the template filter. After selecting a query template, you can add additional query strings and filters.

You can choose from the following query templates:

- **Tagged resources** — This template returns resources with user or system tags, including tagged resource types that are not supported by Resource Explorer.
- **All untagged resources** — This template returns resources with no user or system tags.
- **All non-taggable resources** — This template returns resources that do not support tagging.
- **All untagged resources of [`type`]** — This template returns resources with no user tags of the specified type.
- **Resources not in [`application`]** — This template returns resources that do not belong in the specified application.
- **All resources in [`application`]** — This template returns resources that belong to the specified application.
- **Amazon EC2 resources that are not instances in [`application`]** — This template returns Amazon EC2 resources that are *not* the `ec2:instance` resource type and that belong in the specified application.

# Search query syntax reference for Resource Explorer

AWS Resource Explorer helps you to find individual AWS resources in your AWS accounts. To help you find the exact resources you're looking for, Resource Explorer accepts search query strings that support the syntax described in this topic. For example queries that demonstrate how to use the features described here, see Example Resource Explorer search queries.

> **Note**
>
> At this time, tags attached to AWS Identity and Access Management (IAM) resources, such as roles or users, are not indexed.

## How queries work in Resource Explorer

Search queries always use a view. If you don't explicitly specify one, Resource Explorer uses the view designated as the default for the AWS Region that you're working in.

Views determine which resources are available for you to query. You can create different views that each return a different set of resources.

For example, you could create a view that includes only those resources that are tagged with the key `Environment` and the value `Production`. Then, you could choose to grant access for that view to only those users who have a business reason to view those resources. A separate view that includes the `Alpha` or `Beta` environment resources could be accessed by different users who need to view those resources. For information about controlling who gets access to which views, see Granting access to Resource Explorer views for search.

## Query string syntax

This section provides information about basic aspects of query syntax, filters, and filter operators.

### Basics

At its most basic, a `QueryString` is a set of free-form text keywords that are *implicitly* joined by a logical **OR** operator. Separate each keyword from the others by using a space, as shown in the following example:

```
ec2 billing test gamma
```

Resource Explorer evaluates this list of keywords to mean:

```
ec2 OR billing OR test OR gamma
```

Resource Explorer sorts results by relevance, giving higher preference to resources that match a greater number of the search terms. Resources that don't match one or more of the terms aren't excluded from the results. However, Resource Explorer considers them of lower relevance and pushes them further down in the search results.

If you specify an empty string for the `QueryString` parameter, your query returns the first 1,000 resources that are available through the view used for the operation. The maximum number of resources that can be returned by any query is 1,000.

> **Note**
>
> AWS reserves the right to update the matching logic and relevance algorithms for evaluating free-form text keywords so that we can provide customers with the most relevant results. Therefore, results returned for the same queries using free-form text keywords might change over time. Where you require more deterministic results, we recommend that you use filters. Filter matching logic does not change over time.

## Filters

You can limit the results of your query more strictly by including ***filters***. Unlike text keywords, filters are evaluated in the query with the **AND** operator. For example, consider the following query that consists of two free-form keywords and two filters:

```
test instance service:EC2 region:us-west-2
```

This query is evaluated as follows:

```
( test OR instance ) AND service:EC2 AND region:us-west-2
```

Filters are always evaluated using **AND** logical operators. If a resource doesn't match the filter, that resource is not included in the results. The example query's results include any resources that are associated with Amazon EC2 and are in the US West (Oregon) AWS Region and have at least one of the keywords attached in some way.

**Note**

Because of the implicit AND, you can successfully use only one filter for an attribute that can have only one value associated with the resource. For example, a resource can be part of only one AWS Region. Therefore, the following query returns no results.

```
region:us-east-1 region:us-west-1
```

This limitation does ***not*** apply to the filters for attributes that can have multiple values at the same time, such as `tag:`, `tag.key:`, and `tag.value:`.

The following table lists the available filter names that you can use in a Resource Explorer search query.

| Filter name | Description and example |
| --- | --- |
| `accountid:` | The AWS account that owns the resource. Resource Explorer includes in the results only the resources that are owned by the specified account.<br><br>`accountid:123456789012` |
| `application:` | This filter enables you to search for resources with an `awsApplication` tag key and a resource group value. You can search by application name or the application resource group ARN.<br><br>`application:MyApplicationName`<br><br>`application:arn:aws:resource-groups:  us-east-1 :123456789012:group/MyApplicationName/123456789abced`<br><br>`arn:aws:resource-groups:  us-east-1 :123456789012:group/MyApplicationName/123456789abced`<br><br>**Note**<br>To use this filter, your view must have access to tagging data. |

| Filter name | Description and example |
| --- | --- |
| `id:` | The identifier of an individual resource, expressed as an Amazon resource name (ARN).<br><br>`id:arn:aws:license-manager:  us-east-1 :12345678 9012:license-configuration:lic-ecbd5574fd92cb 0d312baea26EXAMPLE` |
| `region:` | The AWS Region where the resource is located. Resource Explorer includes in the results only the resources that reside in the specified AWS Region.<br><br>`region:us-east-1`<br><br>**Note**<br>Typing only the Region code (without a filter, such as `us-east-1` ) doesn't return the same results as `region:us-east-1` . This outcome is because, as a free-form text keyword that isn't a filter, the Region code is broken down into its individual pieces. For example, `us-east-1` is searched as `us`, `east`, and `1`. This breakdown into components doesn't occur when you use the `region:` prefix. |
| `region:global` | A special case for the `region:` filter that you can use to find resources that are not associated with an individual AWS Region but are considered to be global in scope.<br><br>`region:global`<br><br>**Note**<br>Typing only the keyword `global` doesn't return the same results as `region:global` because the literal word "global" is not attached to global resources. Typing `global` as a keyword returns only those resources that have that literal string associated with the resource. |

| Filter name | Description and example |
| --- | --- |
| `resourcetype:` | The resource type in *`service`*`:`*`type`* notation. Resource Explorer includes in the results only the resources of the specified type.<br><br>`resourcetype:ec2:instance` |
| `resourcetype.supports:` | This filter enables you to search for resources that support tags. `tags` is the only supported value. Resource Explorer includes in the results only the resources that are taggable.<br><br>`resourcetype.supports:tags` |
| `service:` | The AWS service that is associated with the type of the resource. Resource Explorer includes in the results only the resources that are created and managed by the specified service.<br><br>`service:ec2` |
| `tag:` | A tag key/value pair expressed as `<key>=<value>` . Resource Explorer includes in the results only the resources that have a tag with both a matching key and the specified value.<br><br>`tag:environment=production`<br><br>To use this filter, your view must have an `IncludeProperty` with the **Name** parameter specified as `tags`. This configuration displays the `tag` property and value in resource search results. |

| Filter name | Description and example |
| --- | --- |
| `tag:all` | A special case of the `tag:` filter that enables you to search for resources that have one or more user-created tags attached, even if the resource type is not supported in Resource Explorer.<br><br>To use this filter, your view must have an `IncludeProperty` with the **Name** parameter specified as `tags`. This configuration displays the `tag` property and value in resource search results.<br><br>**Note**<br>Resources with *AWS service-created* tags still appear in results for this filter. |
| `tag:none` | A special case of the `tag:` filter that enables you to search for any resources that don't have any user-created tags attached.<br><br>To use this filter, your view must have an `IncludeProperty` with the **Name** parameter specified as `tags`. This configuration displays the `tag` property and value in resource search results.<br><br>**Note**<br>Resources with *AWS service-created* tags still appear in results for this filter. |
| `tag.key:` | A tag key. Resource Explorer includes in the results only the resources that have a tag with a matching key, regardless of value.<br><br>`tag.key:environment`<br><br>To use this filter, your view must have an `IncludeProperty` with the **Name** parameter specified as `tags`. This configuration displays the `tag` property and value in resource search results. |

| Filter name | Description and example |
| --- | --- |
| `tag.value:` | A tag value. Resource Explorer includes in the results only the resources that have a tag with a matching value, regardless of the key name.<br><br>`tag.value:production`<br><br>To use this filter, your view must have an `IncludeProperty` with the **Name** parameter specified as `tags`. This configuration displays the `tag` property and value in resource search results. |

## Filter operators

You can modify your keywords and filters by including one of the operators shown in the following table as part of the string.

| Operator | Description and example |
| --- | --- |
| `"multiple word phrase"`<br><br>or<br><br>`"hyphenated-phrase "` | Surround a multi-word phrase that should be treated as a single keyword with double quotation marks characters (" "). Resource Explorer includes only those resources that match the entire phrase, with all words together, and in the specified order.<br><br>If you don't use the double quotation marks, Resource Explorer breaks up the phrase into its components by spaces or hyphens, and includes resources that match the individual components, even if they're not together or in a different order. Quotations should be around everything after the operator.<br><br>`"This matches only resources with the whole sentence."`<br><br>`This matches resources with any of the words.`<br><br>`"us-east-1"` – matches only resources that associated with that exact Region.<br><br>`us-east-1` – matches any resource that contain "us" or "east" or "1".<br><br>`-tag:"environment=production"` |

| Operator | Description and example |
| --- | --- |
| *keyword** | Prefix wildcard matching. You can place a wildcard character (an asterisk *) at only the end of the string. Resource Explorer includes in the results only the resources with values that start with the prefix text before the *. The following example matches all AWS Regions that begin with us-east.<br><br>`region:us-east*`<br><br>**Important**<br>Unified search automatically inserts a wildcard character (*) operator at the end of the first keyword in the string. This means that unified search results include resources that match any string that starts with the specified keyword.<br>The search performed by the **Query** text box on the Resource search page in the Resource Explorer console does ***not*** automatically append a wildcard character. You can insert a * manually after any term in the search string. |

| Operator | Description and example |
| --- | --- |
| `-keyword` | `Not` operator. You can place a hyphen (`-`) at the beginning of its keyword or filter to invert the search results. Resource Explorer ***excludes*** from the results any resources that match the keyword or filter that follows this operator. The following example causes all resources associated with the Amazon EC2 service to be excluded from the results.<br><br>`-service:ec2`<br><br>**Important**<br><br>If you use the AWS CLI `search` command and your `--query-string` parameter value has the `-` operator as the first character, you must separate the parameter name from its value with an equal sign character (=) instead of the usual space character. If you use the space character, the CLI misinterprets the string. For example, the following query fails.<br><br>`aws resource-explorer-2 search --query-string "-tag:none region:us-east-1"`<br><br>The following corrected query string, with an = replacing the space, works as expected.<br><br>`aws resource-explorer-2 search --query-string ="-tag:none region:us-east-1"`<br><br>If you change the order of the filters in the query string so that the `-` isn't the first character in the parameter value, you can use the standard space character. The following query string works.<br><br>`aws resource-explorer-2 search --query-string "region:us-east-1 -tag:none"` |

| Operator | Description and example |
| --- | --- |
| `\<special character>` | You can escape special characters that must be included exactly as shown rather than interpreted. If your text includes one of the special characters ( `* " - : = \`), you must precede that character with a backslash (\) to ensure that the character is taken literally. The following example shows how to use a free-form text keyword that includes the hyphen (-) character (`"my-key-word"` ).<br><br>Also, to prevent Resource Explorer from breaking up the expression at the hyphens into three separate keywords, you can surround the entire phrase in double quotation marks.<br><br>`"my\-key\-word"`<br><br>To insert a literal backslash, insert two backslash characters in a row. The first backslash is interpreted as the escape and the second backslash is the literal character to insert.<br><br>`"some_text\\some_more_text"` |

**Note**

If the view includes the tags attached to the resources, then the `Search` operation doesn't throw validation errors for search strings, because a filter that's not valid could also be interpreted as a free-form text search. For example, even though `cat:blue` ***looks*** like a filter, Resource Explorer can't parse it as one because `cat:` isn't one of the valid, defined filters. Instead Resource Explorer interprets the whole string as a free-form search string to allow it to match things like a tag key name or a piece of an ARN.
The operation does throw a validation error if either of the following is true:

- The view doesn't include information about tags
- The search query explicitly uses a tag filter (`tag.key:`, `tag.value:`, or `tag:`)

# Example Resource Explorer search queries

The following examples show the syntax for common types of queries that you can use in AWS Resource Explorer.

**Important**

If you use the AWS CLI `search` command and your `--query-string` parameter value has the `-` operator as the first character, you must separate the parameter name from its value with an equal sign character (=) instead of the usual space character. If you use the space character, the CLI misinterprets the string. For example, the following query fails.

```
aws resource-explorer-2 search --query-string "-tag:none region:us-east-1"
```

The following corrected query, with an = replacing the space, works as expected.

```
aws resource-explorer-2 search --query-string="-tag:none region:us-east-1"
```

If you change the order of the filters in the query string so that the `-` isn't the first character in the parameter value, you can use the standard space character. The following query works.

```
aws resource-explorer-2 search --query-string "region:us-east-1 -tag:none"
```

## Search for untagged resources

If you want to use attribute-based access control (ABAC) in your account, use cost-based allocation, or perform tag-based automation against your resources, you need to know which resources in your account might be missing tags. The following example query uses the special case filter tag:none to return all resources that are missing user-generated tags.

The `tag:none` filter applies to only tags that are *created by the user*. Tags that are generated and maintained by AWS are exempt from this filter and still appear in the results.

```
tag:none
```

To also exclude all AWS created system tags, add a second filter as shown in the following example. The first element in the query string duplicates the previous example by filtering out all user-created tags. AWS created system tags *always* begin with the letters `aws`. Therefore, you can use the logical NOT operator ( - ) with the tag.key filter to also exclude any resources that have a tag with a key name that begins with `aws`.

```
tag:none -tag.key:aws*
```

## Search for tagged resources

To find all resources that have a tag of any type, you can use the logical NOT operator ( - ) with the special case tag: none filter as follows.

```
-tag:none
```

## Search for resources that are missing a specific tag

Also related to ABAC, you might want to search for all resources that don't have a tag with a specified key. The following example uses the logical NOT operator - to return all resources that are missing a tag with the key name `Department`.

```
-tag.key:Department
```

## Search for resources that have invalid tag values

For compliance reasons, you might want to search for all resources that have missing or misspelled tag values on important tags. The following example returns all resources that have a tag with the key name `environment`. However, the query filters out any resource that has one of the valid values `prod`, `integ`, or `dev`. Any results that appear from this query have some other value that you should investigate and correct.

> **Important**
>
> Resource Explorer searches are ***not*** case sensitive and can't distinguish between key names and values that differ only by how they're capitalized. For example, the values in the following example match `PR0D`, `prod`, `Pr0d`, or any variation. However, some applications use tags in case-sensitive ways. We recommend that you standardize on a capitalization

strategy for your organization, such as using only lower-case tag key names and values. A consistent approach can help avoid the confusion that can be caused by having tags that differ only by how they're capitalized.

```
tag.key:environment -tag:environment=prod -tag:environment=integ -tag:environment=dev
```

## Search for resources in a subset of AWS Regions

Use the `'*'` wildcard operator to match all Regions in a certain area of the world. The following example returns all resources that are in Regions in Europe (EU).

```
region:eu-*
```

## Search for global resources

Use the special case `global` value for the `region:` filter to find your resources that are considered to be global and not associated with an individual Region.

```
region:global
```

## Search for resources of a certain type that are located in a specific Region

When you use multiple filters, Resource Explorer evaluates the expression by combining the prefixes with implicit logical `AND` operators. The following example returns all resources that are in the Asia Pacific (Hong Kong) Region `AND` are Amazon EC2 instances.

```
region:ap-east-1 resourcetype:ec2:instance
```

**Note**

Because of the implicit `AND`, you can successfully use only one filter for an attribute that can have only one value associated with the resource. For example, a resource can be part of only one AWS Region. Therefore, the following query returns no results.

```
region:us-east-1 region:us-west-1
```

This limitation does ***not*** apply to the filters for attributes that can have multiple values at the same time, such as `tag:`, `tag.key:`, and `tag.value:`.

## Search for resources that have a multi-word term

Surround a multi-word term with double quotation marks (") to return only results that have the entire term in the specified order. Without double quotation marks, Resource Explorer returns resources that match any individual words that make up the term. For example, the following query uses the double quotation marks to return only resources that match the term `"west wing"`. The query does ***not*** match resources in the `us-west-2` AWS Region (or any other Region that includes `west` in its code) or resources that match the word "wing" without the word "west".

```
"west wing"
```

## Search for resources that are part of a specified CloudFormation stack

When you create a resource as part of an AWS CloudFormation stack, they are all tagged with the stack's name *automatically*. The following example returns all resources that were created as part of the specified stack.

```
tag:aws:cloudformation:stack-name=my-stack-name
```

# Viewing resource details

In the Resource Explorer console, you can view the details of a selected resource, including information about the resource provided by other AWS services. To view a resource's details, open the **Resources** page, and select the checkbox of the desired resource. Review each of the tabs to learn more about the resource.

**Topics**


- Overview
- Relationships
- Timeline
- Compliance
- Resource shares
- Tags
- Additional properties


## Overview

The Overview tab displays basic information about the selected resource, including the **Resource type**, **Amazon Resource Name (ARN)**, the **AWS Region** where the resource resides, the **Owner account**, and the **Last indexed** date and time stamp. For some resource types, this tab also displays resource properties sourced from the AWS Cloud Control API, like public access settings for Amazon S3 buckets or instance state for Amazon EC2 instances. This tab also includes a link to the resource's native console.

If additional AWS services are enabled in your AWS account, the Overview tab also displays the following information:

- **AWS Security Hub** — This integration displays a total number of Security findings and a total number of Critical and High ranked findings. If available, choosing the **Total findings** link directs you to the Security Hub console.
- **AWS Cost Explorer** — This integration displays the resource's **Cost** over the past 14 days. Choosing the **cost value** link directs you to the resource's cost details page in the AWS Cost Explorer console.

- **AWS Config** — This integration displays the resource's **Compliance** status with rules from AWS Config. Choosing the compliance status link directs you to the resource details **Compliance** tab.

#### Minimum permissions

To view all of the available resource details in this tab, which includes details from other AWS services, you must have the following permissions:

- **Action**: `ce:GetCostAndUsageWithResources`
- **Action**: `cloudformation:GetResource`
- **Action**: `config:DescribeComplianceByResource`
- **Action**: `config:DescribeConfigurationRecorders`
- **Action**: `config:DescribeConfigurationRecorderStatus`
- **Action**: `config:GetComplianceDetailsByResource`
- **Action**: `securityhub:GetAdhocInsightResults`

## Relationships

The Relationships tab displays the selected resource's single-level relationships to other resources. This tab only displays relationships for supported resource types.

#### Minimum permissions

To view resource relationships in this tab, you must have at least read-only permissions for all resource types and underlying AWS services you want to visualize. Resource Explorer recommends using the `ReadOnlyAccess` general AWS managed policy. You can attach this policy to your users, groups, and roles to provide read-only access to AWS services and resources.

## Timeline

If you have AWS Config enabled in your account, the Timeline tab displays a resource's history of events over the past 60 days. You can filter by Configuration events, Compliance events, or CloudTrail Events.

You can learn more about Viewing compliance history timeline for resources and rules in the *AWS Config developer guide*.

For accounts without AWS Config enabled, the Timeline tab displays AWS CloudTrail events. You can learn more about Understanding CloudTrail events in the *AWS CloudTrail User Guide*.

**Minimum permissions**

To view a resource's event history in this tab, you must have the following permissions:

- **Action**: `config:DescribeConfigurationRecorderStatus`
- **Action**: `cloudtrail:LookupEvents`
- **Action**: `config:GetResourceConfigHistory`

## Compliance

If your account already has AWS Config enabled, but does not include any rules, you can choose **Add rule** to create new rules in the AWS Config console.

If the selected resource does include rules, this tab displays the resource's **Compliant** and **Non-compliant** rules, including any known fixes for non-compliant rules. Choosing an individual rule directs you to the rule in the AWS Config console.

You can learn more about Evaluating resources with AWS Config rules in the *AWS Config developer guide*.

**Minimum permissions**

To view resource details in this tab, you must have AWS Config enabled in your AWS account and have the following permissions:

- **Action**: `config:DescribeComplianceByResource`
- **Action**: `config:DescribeConfigurationRecorders`
- **Action**: `config:DescribeConfigurationRecorderStatus`
- **Action**: `config:DescribeRemediationConfigurations`
- **Action**: `config:GetComplianceDetailsByResource`

## Resource shares

The Resource shares tab displays any resource shares that include this resource. Use AWS Resource Access Manager to create resource shares that make the resource available to other individual AWS

accounts, or to the accounts in an organization or an organizational unit. Review Sharing your AWS resources in the AWS Resource Access Manager user guide for more information.

**Minimum permissions**

To view resource shares in this tab, you must have the following permissions:

- **Action**: `ram:ListResources`
- **Action**: `ram:GetResourceShares`

## Tags

The Tags tab displays a list of tags attached to the selected resource. Each tag contains a key name and an associated value that you can use to categorize your resources.

**Minimum permissions**

To view a resource's tags in this tab, you must have the following permissions:

- **Action**: `tag:GetResources`

## Additional properties

The Additional properties tab displays resource details obtained by AWS Cloud Control API, including the availability zone, block device mappings, and more.

**Minimum permissions**

To view a resource's additional properties in this tab, you must have the following permissions:

- **Action**: `cloudformation:GetResource`

# Managing resources in the Resource Explorer console

The Resource Explorer console supports resource quick-actions and integrations with several other AWS services, allowing you to perform the most common resource management tasks and providing additional resource information from one console.

## Resource Explorer console integrations with other AWS services

**Amazon Q Developer Ask Amazon Q** — When you select one or more resources, choosing **Actions**, **Help me with this resource** opens a chat panel where you can ask Amazon Q for more information about those resources. For example, you can ask for details about a specific resource or list resources based on criteria such as AWS Region or state. To learn more, review Chatting about your resources in the *Amazon Q Developer User Guide*.

## Resource Actions in the Resource Explorer console

The Resource **Actions** menu enables you to perform common resource management tasks on a selection of up to 400 resources from within the Resource Explorer console.

**Topics**

- Manage resource tags
- Create application
- Add to application
- Export resources to a .csv file

## Manage resource tags

You can select up to 400 resources and apply *tags* to them. Tags are key and value pairs that act as metadata for organizing your AWS resources and can help you manage, identify, organize, search for, and filter resources.

> **Note**
>
> Tags are not encrypted and should not be used to store sensitive data, such as personally identifiable information (PII) or personal health information (PHI).

Each tag has two parts:

- A *tag key* (for example, CostCenter, Environment, or Project). Tag keys are case sensitive.
- A *tag value* (for example, 111122223333 or Production). Like tag keys, tag values are case sensitive.

The resources in your selection do not need to all reside in the AWS Region you currently have selected. The following behaviors apply when tagging resources in the Resource Explorer console:

- Global resources, such as `iam::Role`, are resources you can use from anywhere. In the Resource Explorer console, global resources do not display a region.
- If any of the selected resources already have a tag key and you specify a new value for that key, the newly specified tag key-value pair is applied to all of the selected resources.
- After updating tags with bulk tagging in the Resource Explorer console, the tag changes are not immediately reflected in the resources' tag count. After bulk tagging changes are applied, new resource searches may take up to 30 seconds to reflect new tagging details.

> **Note**
>
> AWS recommends not including AWS CloudFormation stacks in your resource selection when managing tags in the Resource Explorer console. Instead, you should manage tags on AWS CloudFormation stacks only using AWS CloudFormation. Tagging AWS CloudFormation stacks from Resource Explorer can cause unexpected tagging behavior, resulting in downtime or other issues.

### Minimum permissions

To add or remove tags from a resource, you need the permissions required for the service to which the resource belongs. For example, to tag Amazon EC2 instances, your must have permissions to the tagging actions in that service's API. For more information, review Grant permission to tag Amazon EC2 resources during creation in the *Amazon EC2 User Guide*.

To perform the steps in the following procedure, you must have the following permissions:

- **Action**: `tag:GetTagKeys`
- **Action**: `tag:GetTagValues`

- **Action**: `tag:TagResources`
- **Action**: `tag:UntagResources`

**To manage tags for a selection of resources**

1. On the **Resources** page, start by choosing the view that you want to use. You can choose from among only those views that you have permissions to access.
2. (Optional) Submit a Resource query.
3. In **Resources**, select up to 400 resources.
4. For **Actions**, choose **Manage tags**.
5. Select an **existing tag** or **create a new tag** to apply to all of the selected resources.
6. Choose **Apply**.

## Create application

You can select up to 400 resources and create a new application that includes those resources. The application is visible in myApplications in the AWS Management Console. All resources in the selection must meet the following requirements to be successfully added to a new application:

- Resources must be in the same AWS Region because an application can only exist in a single region.
- Global resources can only be added to an application that resides in the global resource's home region. To add a global resource to an application, apply the `awsApplication` tag to the resource. You can learn more about global AWS services and their resources in the Global services AWS whitepaper.
- Resources must be supported by the Resource Groups Tagging API.
- Resources must reside in the same AWS account.
- Resources must not already be in an application.

> **Note**
>
> AWS recommends not including AWS CloudFormation stacks in your resource selection when creating an application in the Resource Explorer console. Creating an application that includes a AWS CloudFormation stack requires a stack update because all resources added

to your application are tagged with the `awsApplication` tag. Manual configurations performed after the stack was last updated may not be reflected after this update. This can cause downtime or other application issues. For more information, see Update behaviors of stack resources in the *AWS CloudFormation User Guide*.

**Minimum permissions**

To create a new application, or add resources to an existing application, you need additional permissions to tag resources and to perform application actions in Resource Groups and AWS Service Catalog.

To perform the steps in the following procedure, you must have the following permissions:

- **Action:** `tag:TagResources`
- **Action:** `resource-groups:Tag`
- **Action:** `resource-groups:CreateGroup`
- **Action:** `resource-groups:GroupResources`
- **Action:** `servicecatalog:CreateApplication`
- **Action:** `servicecatalog:TagResource`

**To create an application from a selection of resources**

1. On the **Resource search** page, start by choosing the view that you want to use. You can choose from among only those views that you have permissions to access.
2. (Optional) Submit a Resource query.
3. In **Resources**, select up to 400 resources.
4. For **Actions**, choose **Create application**.
5. In **Create application**, enter the **application name** and select a **Region**.
6. (Optional) Add **Tags** and **Attribute groups**.
7. Choose **Create**.

After creating the new application, resource searches may take several minutes to reflect new tagging details.

# Add to application

You can select up to 400 resources and add those resources to an existing application. All resources in the selection must meet the following requirements to be successfully added to an application:

- Resources must be in the same AWS Region because an application can only exist in a single region.
- Global resources can only be added to an application that resides in the global resource's home region. To add a global resource to an application, apply the `awsApplication` tag to the resource. You can learn more about global AWS services and their resources in the Global services AWS whitepaper.
- Resources must be supported by the Resource Groups Tagging API.
- Resources must reside in the same AWS account.
- Resources must not already be in an application.

> **Note**
>
> AWS recommends not including AWS CloudFormation stacks in your resource selection when adding resources to an application in the Resource Explorer console. Adding a AWS CloudFormation stack to the application requires a stack update because all resources added to your application are tagged with the `awsApplication` tag. Manual configurations performed after the stack was last updated may not be reflected after this update. This can cause downtime or other application issues. For more information, see Update behaviors of stack resources in the *AWS CloudFormation User Guide*.

### Minimum permissions

To create a new application, or add resources to an existing application, you need additional permissions to tag resources and to perform application actions in Resource Groups and AWS Service Catalog.

To perform the steps in the following procedure, you must have the following permissions:

- **Action:** `tag:TagResources`
- **Action:** `resource-groups:Tag`

- **Action:** `resource-groups:CreateGroup`
- **Action:** `resource-groups:GroupResources`
- **Action:** `servicecatalog:CreateApplication`
- **Action:** `servicecatalog:TagResource`

**To add a selection of resources to an application**

1. On the **Resource search** page, start by choosing the view that you want to use. You can choose from among only those views that you have permissions to access.
2. (Optional) Submit a Resource query.
3. In **Resources**, select up to 400 resources.
4. For **Actions**, choose **Add to application**.
5. In **Applications**, select the desired application.
6. Choose **Next**.
7. (Optional) If necessary, choose **Remove resources from their application** to remove resources from their current application and add them to your newly selected application.
8. Choose **Confirm**.
9. Select the final acknowledgement about removing resources from their current application to your newly selected application, and then choose **Confirm**.

After creating the new application, resource searches may take several minutes to reflect new tagging details.

## Export resources to a .csv file

You can export the results of a **Resource query** to a comma-separated values (.csv) file. The .csv file includes the identifier, resource type, Region, AWS account, the total number of tags, and a column for each unique tag key in the collection. The .csv file can help you configure your AWS resources in your organization, or determine where there are overlaps or inconsistencies in tagging across resources.

1. In the results of your **Resources** query, choose **Actions**, **Export CSV**.

   For searches using search operators (calling the `ListResources` API) where results may return more than 1,000 matches, pagination is progressive and loads pages in groups of 10.

For example, exporting to CSV from page 10 exports 1,000 results. Exporting from page 11 paginates through page 20, exports up to 2,000 results.

2. If prompted by your browser, choose to open the .csv file, or save it to a convenient location.

# Using unified search in the AWS Management Console

The AWS Management Console includes a search bar at the top of every AWS console page. This search bar can search the AWS service documentation and blog topics, and take you directly to AWS service console pages. It can also return the resources in your AWS account, if you turn on the unified search feature by turning on the required Resource Explorer features.

With unified search, your users can search for resources from ***any*** AWS service console without having to first navigate to the AWS Resource Explorer console.

> **Tip**
>
> When you want to use the unified search bar to search specifically for resources, begin your search query by typing **`/Resources`**. This causes AWS resources to be ranked higher in the search results than results that do not represent resources.

**Topics**


- Checking if unified search is enabled
- Turning on unified search


> **Important**
>
> Unified search automatically inserts a wildcard character (*) operator at the end of the first keyword in the string. This means that unified search results include resources that match any string that starts with the specified keyword.
> The search performed by the **Query** text box on the Resource search page in the Resource Explorer console does ***not*** automatically append a wildcard character. You can insert a * manually after any term in the search string.

## Checking if unified search is enabled

To see if unified search is enabled in your AWS account, look at the top of the **Settings** page. Resource Explorer displays the current status of each requirement there. The requirements for unified search are as follows:

- You must turn on Resource Explorer in at least one AWS Region. Only resources in Regions with Resource Explorer indexes can appear in unified search results.
- You must create an aggregator index in the Region of your choice. Searches performed in this Region return results from all registered Regions in the account.
- You must create a default view in the Region that contains the aggregator index. All users who need to use unified search for resources must have permission to use this default view.
- Users must have an AWS Identity and Access Management (IAM) permissions policy assigned to their IAM principal that grants permission to perform the `resource-explorer-2:Get*`, `resource-explorer-2:List*`, `resource-explorer-2:Describe*`,`resource-explorer-2:Search` actions. You can grant these permissions by using your own custom IAM policies. These permissions are already included as part of the following AWS managed policies that are available for your use:
  - AWSResourceExplorerReadOnlyAccess
  - AWSResourceExplorerFullAccess

## Turning on unified search

To enable including your account's resources in the search results for unified search from any AWS console, you must complete the following steps:

1. Activate AWS Resource Explorer in one or more AWS Regions in your account.
2. Register one Region to contain the aggregator index.
3. Create a default view in the Region with the aggregator index.

# Creating Resource Explorer resources with CloudFormation

AWS Resource Explorer is integrated with AWS CloudFormation, a service that helps you model and set up your AWS resources. This integration helps you spend less time creating and managing your resources and infrastructure. You create a template that describes all the AWS resources that you want, and CloudFormation provisions and configures those resources for you. Examples of resources include indexes, views, or the assignment of a default view for an AWS Region.

When you use CloudFormation, you can reuse your template to set up your Resource Explorer resources consistently and repeatedly. Just describe your resources once, and then provision the same resources over and over in multiple AWS accounts and Regions.

**Using AWS CloudFormation to deploy Resource Explorer to AWS Organizations**

You can use AWS CloudFormation StackSets to deploy Resource Explorer to all of the accounts in your organization. When you add or create member accounts in your organization, StackSets can automatically configure indexes in each AWS Region, including an aggregator index where you specify, to each new member account. For instructions, see Deploying Resource Explorer to the accounts in an organization.

## Resource Explorer and CloudFormation templates

To provision and configure resources for Resource Explorer and related services, you must understand AWS CloudFormation templates. Templates are formatted text files in JSON or YAML. These templates describe the resources that you want to provision in your CloudFormation stacks. If you're unfamiliar with JSON or YAML, you can use AWS CloudFormation Designer to help you get started with CloudFormation templates. For more information, see What is AWS CloudFormation Designer? in the *AWS CloudFormation User Guide*.

Resource Explorer supports creating the following resource types in CloudFormation:

- **Index** – Creates an index in a Region and turns on Resource Explorer in that Region. You can specify that the index be either local or the aggregator index for the AWS account. For more information, see Turning on Resource Explorer in an AWS Region to index your resources and Turning on cross-Region search by creating an aggregator index.

- **View** – Creates a view that determines what results can appear when a user performs a search. Every search operation must specify a view. You must grant users permission to use the views that you want them to access. For more information, see Configuring an Resource Explorer view to provide access to resource searches.

  > **Note**
  >
  > You must create an index in a Region before you can create a view in that same Region. If you create an index and view as part of the same stack, use the `DependsOn` attribute on the view, as shown in the following example template, to ensure that the index is created first.

- **DefaultViewAssociation** – Assigns the specified view to be the default in its Region. When a user doesn't explicitly specify the view to use for a search operation, Resource Explorer attempts to use the default view associated with the Region in which the user performs the search. For more information, see Setting a default view in an AWS Region

The following example illustrates how you might create one index and a view in the same Region, and set the view to be the default for the Region.

**YAML**

```
Description: >-
  Sample CFN Stack setting up Resource Explorer with an aggregator index and a default
 view
Resources:
  SampleIndex:
    Type: 'AWS::ResourceExplorer2::Index'
    Properties:
      Type: AGGREGATOR
      Tags:
        Purpose: ResourceExplorer Sample CFN Stack
  SampleView:
    Type: 'AWS::ResourceExplorer2::View'
    Properties:
      ViewName: mySampleView
      IncludedProperties:
        - Name: tags
      Tags:
        Purpose: ResourceExplorer Sample CFN Stack
```

```
    DependsOn: SampleIndex
  SampleDefaultViewAssociation:
    Type: 'AWS::ResourceExplorer2::DefaultViewAssociation'
    Properties:
      ViewArn: !Ref SampleView
```

### JSON

```
{
    "Description": "Sample CFN Stack setting up Resource Explorer with an aggregator
 index and a default view ",
    "Resources": {
        "SampleIndex": {
            "Type": "AWS::ResourceExplorer2::Index",
            "Properties": {
                "Type": "AGGREGATOR",
                "Tags": {
                    "Purpose": "ResourceExplorer Sample Stack"
                }
            }
        },
        "SampleView": {
            "Type": "AWS::ResourceExplorer2::View",
            "Properties": {
                "ViewName": "mySampleView",
                "IncludedProperties": [
                    {
                        "Name": "tags"
                    }
                ],
                "Tags": {
                    "Purpose": "ResourceExplorer Sample CFN Stack"
                }
            },
            "DependsOn": "SampleIndex"
        },
        "SampleDefaultViewAssociation": {
            "Type": "AWS::ResourceExplorer2::DefaultViewAssociation",
            "Properties": {
                "ViewArn": {
                    "Ref": "SampleView"
                }
            }
```

```
        }
    }
}
```

For more information, including examples of JSON and YAML templates for Resource Explorer indexes and views, see the ResourceExplorer2 resource type reference in the *AWS CloudFormation User Guide*.

## Learn more about AWS CloudFormation

To learn more about CloudFormation, see the following resources:

- AWS CloudFormation
- AWS CloudFormation User Guide
- AWS CloudFormation Command Line Interface User Guide

# Using Amazon Q Developer in chat applications to search for resources

You can search and discover information about AWS services and your AWS resources by asking Amazon Q Developer in chat applications natural language questions. Amazon Q Developer in chat applications answers service-related questions directly in your chat channels with relevant AWS documentation and support article excerpts. Amazon Q Developer in chat applications uses Resource Explorer to search and find answers to your resource related questions.

For more information, see What is Amazon Q Developer in chat applications? in the *Amazon Q Developer in chat applications Administrator Guide*.

## AWS resource questions

Amazon Q Developer in chat applications uses Resource Explorer to search and discover your resources. Amazon Q Developer in chat applications displays these search results in a list. This list shows the top five matching resources and includes the ability to filter results further by resource type, AWS Region, and tag.

### Prerequisites

To ask Amazon Q Developer in chat applications resource related questions you must:

- Make sure you have active indexes and views with at least one default view in your AWS Region. Indexes and views allow Resource Explorer to catalog and query your resources. See Terms and concepts for Resource Explorer for more information.
- Add the AWSResourceExplorerReadOnlyAccess policy to your channel role or each appropriate user role, depending on your channel's permission scheme.
- Verify that your channel guardrail policies allow AWSResourceExplorerReadOnlyAccess permissions.

### Commonly asked resource questions

You can ask these questions directly from your chat channels. Replace the words with red text with your own information.

```
@aws What services am I using in Region?

@aws What are the resources in my account with tags?

@aws What lambda functions do I have?
```

# Turning off Resource Explorer

When you no longer need to search for resources in a specific AWS Region, you can turn off AWS Resource Explorer in only that Region by deleting its index, or you can delete Resource Explorer in all AWS Regions. When you do this, Resource Explorer stops scanning for new or updated resources in that Region. If your account contains an aggregator index, then replication from the deleted index stops, and the information from the deleted index is removed from the aggregator index and stops appearing in search results. It can take up to 24 hours for all resources from the deleted index to disappear from search results in the Region with the aggregator index.

> **Note**
>
> When you register the first AWS Region, Resource Explorer creates a service linked role (SLR) named `AWSServiceRoleForResourceExplorer` in the AWS account. Resource Explorer ***doesn't*** delete this SLR automatically. After you delete the Resource Explorer index in every Region in the account, you can use the IAM console to delete the SLR if you won't use Resource Explorer in the future. If you do delete the role and you then choose to turn on Resource Explorer again in at least one AWS Region, Resource Explorer re-creates the service-linked role automatically.

## Turning off Resource Explorer in one AWS Region

You can turn off Resource Explorer in an AWS Region by using the AWS Management Console, by using commands in the AWS Command Line Interface (AWS CLI), or by using API operations in an AWS SDK.

If you turn off Resource Explorer for a member account, and the member is in an organization wide view, it will be removed from the multi-account search results.

If your account includes a managed view (a view managed by an AWS service), the managed view must be deleted before you can turn off Resource Explorer. Review AWS Managed views for instructions on removing a managed view from your account and prompting the managing service to delete the view.

If you no longer want to support searching for resources in one or more of the AWS Regions in your account, perform the steps in the following procedure.

> **Note**
>
> If the index you delete is the aggregator index for the AWS account, you must wait 24 hours before you can promote another local index to be the aggregator index for the account. Users can't perform account-wide searches using Resource Explorer until another aggregator index is configured.

AWS Management Console

**To delete the Resource Explorer index in an AWS Region**

1. Open the Resource Explorer **Settings** page.
2. In the **Indexes** section, select the check boxes next to the AWS Regions with the indexes that you want to delete, and then choose **Delete**.
3. On the **Delete indexes** page, verify that you selected only indexes that you want to delete. Type **`delete`** in the **Confirm** text box, and then choose **Delete indexes**.

   Resource Explorer displays a green banner at the top of the page to indicate success, or a red banner if there is an error with one or more of the selected Regions.

AWS CLI

**To delete the Resource Explorer index in an AWS Region**

If you no longer want to support searching for resources in one or more of the AWS Regions in your account, run the following commands.

Run the following command for each Region with the indexes that you want to delete. You must run the command in the Region with the index you want to delete. The following example command deletes the Resource Explorer index in the US West (Oregon) (`us-west-2`).

```
$ aws resource-explorer-2 delete-index \
    --arn arn:aws:resource-explorer-2:us-
west-2:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd22222222 \
    --region us-west-2
{
    "Arn": "arn:aws:resource-explorer-2:us-
west-2:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd22222222",
    "State": "DELETING"
```

```
}
```

Because Resource Explorer performs some of the deletion cleanup work as asynchronous tasks in the background, the response might indicate that the operation is `DELETING`. This status indicates that the background processes are not yet complete. You can check for final completion by running the following command, and checking for the `State` to change to `DELETED`.

```
$ aws resource-explorer-2 get-index \
    --region us-west-2
{
    "Arn": "arn:aws:resource-explorer-2:us-
west-2:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingFrom": [],
    "State": "DELETED",
    "Tags": {},
    "Type": "LOCAL"
}
```

## Turning off Resource Explorer in all AWS Regions

If you want to turn off AWS Resource Explorer completely, perform the following procedure.

> **Note**
>
> Resource Explorer creates a service linked role named AWSServiceRoleForResourceExplorer in the account when you create an index in the first AWS Region for an account. Resource Explorer *does not* automatically delete this service linked role. After you delete the Resource Explorer index in every Region, you can then use the IAM console to delete the role if you're sure you won't be using Resource Explorer again in the future. If you do delete the role and you then choose to start Resource Explorer in at least one AWS Region, Resource Explorer recreates the service-linked role.

If your account includes a managed view (a view managed by an AWS service), the managed view must be deleted before you can turn off Resource Explorer. Review AWS Managed views for

instructions on removing a managed view from your account and prompting the managing service to delete the view.

You can turn off Resource Explorer by using the AWS Management Console, by using commands in the AWS Command Line Interface (AWS CLI), or by using API operations in an AWS SDK.

AWS Management Console

If you no longer want to support searching for resources in any AWS Region in your AWS account, perform the steps in the following procedure.

**To turn off Resource Explorer in all AWS Regions**

1. Open the Resource Explorer **Settings** page.
2. In the **Indexes** section, select the check boxes next to all registered AWS Regions, and then choose **Delete**.

   > **Tip**
   >
   > You can check the box in the table header row next to **Index** to check the boxes for all Regions in a single step.

3. On the **Delete indexes** page, verify that you want to delete all indexes. Type **`delete`** in the **Confirm** text box, and then choose **Delete indexes**.

   Resource Explorer displays a green banner at the top of the page to indicate success, or a red banner if there is an error with one or more of the selected Regions.

AWS CLI

**To turn off Resource Explorer in all AWS Regions**

If you no longer want to support searching for resources in any AWS Regions in your account, run the following command to find the ARN of every index in each AWS Region in which you previously turned on Resource Explorer.

```
$ aws resource-explorer-2 list-indexes --query Indexes[*].Arn[
"arn:aws:resource-explorer-2:us-east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-
abcd11111111",
"arn:aws:resource-explorer-2:us-west-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-
abcd22222222",
```

```
"arn:aws:resource-explorer-2:us-west-2:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-
abcd33333333"
]
```

For each response, run the following command to delete the Resource Explorer index in that Region.

```
$ aws resource-explorer-2 delete-index \
    --arn arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111 \
    --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "State": "DELETING"
}
```

Repeat the previous command in each additional Region.

Because Resource Explorer performs some of the cleanup as asynchronous tasks in the background, the response might indicate that the operation is `DELETING`. This status indicates that the background processes are not yet complete. You can check for final completion by running the following command, and checking for the status to change to `DELETED`.

```
$ aws resource-explorer-2 get-index \
    --region us-east-1
{
    "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
    "CreatedAt": "2022-07-12T18:59:10.503000+00:00",
    "LastUpdatedAt": "2022-07-13T18:41:58.799000+00:00",
    "ReplicatingFrom": [],
    "State": "DELETED",
    "Tags": {},
    "Type": "LOCAL"
}
```

# Security in AWS Resource Explorer

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

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

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

This documentation helps you understand how to apply the shared responsibility model when using AWS Resource Explorer. It shows you how to configure Resource Explorer to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your Resource Explorer resources.

**Contents**


- Upgrade IAM policies to IPv6
- Identity and access management for AWS Resource Explorer
- Data protection in AWS Resource Explorer
- Compliance validation for AWS Resource Explorer
- Resilience in AWS Resource Explorer
- Infrastructure security in AWS Resource Explorer
- Access AWS Resource Explorer using an interface endpoint (AWS PrivateLink)

# Upgrade IAM policies to IPv6

AWS Resource Explorer customers use IAM policies to set an allowed range of IP addresses and prevent any IP addresses outside the configured range from being able to access Resource Explorer APIs.

The *resource-explorer-2.region.api.aws* domain where Resource Explorer APIs are hosted is being upgraded to support IPv6 in addition to IPv4.

IP address filtering policies that are not updated to handle IPv6 addresses might result in clients losing access to the resources on the Resource Explorer API domain.

## Customers impacted by upgrade from IPv4 to IPv6

Customers who are using dual addressing with policies containing *aws:sourceIp* are impacted by this upgrade. Dual addressing means that the network supports both IPv4 and IPv6.

If you are using dual addressing, you must update your IAM policies that are currently configured with IPv4 format addresses to include IPv6 format addresses.

For help with access issues, contact Support.

> **Note**
>
> The following customers are *not* impacted by this upgrade:
>
> - Customers who are on *only* IPv4 networks.
> - Customers who are on *only* IPv6 networks.

## What is IPv6?

IPv6 is the next generation IP standard intended to eventually replace IPv4. The previous version, IPv4, uses a 32-bit addressing scheme to support 4.3 billion devices. IPv6 instead uses 128-bit addressing to support approximately 340 trillion trillion trillion (or 2 to the 128th power) devices.

```
2001:cdba:0000:0000:0000:0000:3257:9652
2001:cdba:0:0:0:0:3257:9652
2001:cdba::3257:965
```

# Updating an IAM policy for IPv6

IAM policies are currently used to set an allowed range of IP addresses using the `aws:SourceIp` filter.

Dual addressing supports both IPv4 and IPV6 traffic. If your network uses dual addressing, you must ensure that any IAM polices that are used for IP address filtering are updated to include IPv6 address ranges.

For example, this Amazon S3 bucket policy identifies allowed IPv4 address ranges `192.0.2.0.*` and `203.0.113.0.*` in the `Condition` element.

```
# https://docs.aws.amazon.com/IAM/latest/UserGuide/
reference_policies_examples_aws_deny-ip.html
{
    "Version": "2012-10-17",
    "Statement": {
        "Effect": "Deny",
        "Action": "*",
        "Resource": "*",
        "Condition": {
            "NotIpAddress": {
                "*aws:SourceIp*": [
                    "*192.0.2.0/24*",
                    "*203.0.113.0/24*"
                ]
            },
            "Bool": {
                "aws:ViaAWSService": "false"
            }
        }
    }
}
```

To update this policy, the policy's `Condition` element is updated to include IPv6 address ranges `2001:DB8:1234:5678::/64` and `2001:cdba:3257:8593::/64`.

> **Note**
>
> DO NOT REMOVE the existing IPv4 addresses because they are needed for backward compatibility.

```
"Condition": {
            "NotIpAddress": {
                "*aws:SourceIp*": [
                    "*192.0.2.0/24*", <<DO NOT REMOVE existing IPv4 address>>
                    "*203.0.113.0/24*", <<DO NOT REMOVE existing IPv4 address>>
                    "*2001:DB8:1234:5678::/64*", <<New IPv6 IP address>>
                    "*2001:cdba:3257:8593::/64*" <<New IPv6 IP address>>
                ]
            },
            "Bool": {
                "aws:ViaAWSService": "false"
            }
        }
```

For more information about managing access permissions with IAM, see Managed policies and inline policies in the *AWS Identity and Access Management User Guide*.

## Verify your client can support IPv6

Customers using the *resource-explorer-2.{region}.api.aws* endpoint are advised to verify if their clients can access other AWS service Endpoints that are already IPv6 enabled. The following steps describe how to verify those endpoints.

This examples uses Linux and curl version 8.6.0 and uses the Amazon Athena service endpoints which has IPv6 enabled endpoints located at the *api.aws domain*.

> **Note**
>
> Switch the AWS Region to the same Region where the client is located. In this example, we use the US East (N. Virginia) – `us-east-1` endpoint.

1. Determine if the endpoint resolves with an IPv6 address using the following curl command.

```
dig +short AAAA athena.us-east-1.api.aws
2600:1f18:e2f:4e05:1a8a:948e:7c08:d2d6
2600:1f18:e2f:4e03:4a1e:83b0:8823:4ce5
2600:1f18:e2f:4e04:34c3:6e9a:2b0d:dc79
```

2. Determine if the client network can make a connection using IPv6 using the following curl command.

```
curl --ipv6 -o /dev/null --silent -w "\nremote ip: %{remote_ip}\nresponse code:
 %{response_code}\n" https://athena.us-east-1.api.aws

remote ip: 2600:1f18:e2f:4e05:1a8a:948e:7c08:d2d6
response code: 404
```

If a remote IP was identified **and** the response code is not `0`, a network connection was successfully made to the endpoint using IPv6.

If the remote IP is blank or the response code is `0`, the client network or the network path to the endpoint is IPv4-only. You can verify this configuration with the following curl command.

```
curl -o /dev/null --silent -w "\nremote ip: %{remote_ip}\nresponse code:
 %{response_code}\n" https://athena.us-east-1.api.aws

remote ip: 3.210.103.49
response code: 404
```

If a remote IP was identified **and** the response code is not `0`, a network connection was successfully made to the endpoint using IPv4. The remote IP should be an IPv4 address because the operating system should select the protocol that is valid for the client. If the remote IP is not an IPv4 address, use the following command to force curl to use IPv4.

```
curl --ipv4 -o /dev/null --silent -w "\nremote ip: %{remote_ip}\nresponse code:
 %{response_code}\n" https://athena.us-east-1.api.aws

remote ip: 35.170.237.34
response code: 404
```

# Identity and access management for AWS Resource Explorer

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

**Topics**

- Audience
- Authenticating with identities
- Managing access using policies
- How Resource Explorer works with IAM
- AWS Resource Explorer identity-based policy examples
- Example service control policies for AWS Organizations and Resource Explorer
- AWS managed policies for AWS Resource Explorer
- Using service-linked roles for Resource Explorer
- Troubleshooting AWS Resource Explorer permissions


## Audience

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

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

**Service administrator** – If you're in charge of Resource Explorer resources at your company, you probably have full access to Resource Explorer. It's your job to determine which Resource Explorer features and resources your service users should access. You must then submit requests to your IAM administrator to change the permissions of your service users. Review the information on this page to understand the basic concepts of IAM. To learn more about how your company can use IAM with Resource Explorer, see How Resource Explorer works with IAM.

**IAM administrator** – If you're an IAM administrator, you might want to learn details about how you can write policies to manage access to Resource Explorer. To view example Resource Explorer identity-based policies that you can use in IAM, see AWS Resource Explorer identity-based policy examples.

# Authenticating with identities

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

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

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

If you access AWS programmatically, AWS provides a software development kit (SDK) and a command line interface (CLI) to cryptographically sign your requests by using your credentials. If you don't use AWS tools, you must sign requests yourself. For more information about using the recommended method to sign requests yourself, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

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

## AWS account root user

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

## Users and groups

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

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

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

## Roles

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

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

- **Federated user access** – To assign permissions to a federated identity, you create a role and define permissions for the role. When a federated identity authenticates, the identity is associated with the role and is granted the permissions that are defined by the role. For information about roles for federation, see Create a role for a third-party identity provider (federation) in the *IAM User Guide*. If you use IAM Identity Center, you configure a permission set. To control what your identities can access after they authenticate, IAM Identity Center correlates the permission set to a role in IAM. For information about permissions sets, see Permission sets in the *AWS IAM Identity Center User Guide*.
- **Temporary IAM user permissions** – An IAM user or role can assume an IAM role to temporarily take on different permissions for a specific task.

- **Cross-account access** – You can use an IAM role to allow someone (a trusted principal) in a different account to access resources in your account. Roles are the primary way to grant cross-account access. However, with some AWS services, you can attach a policy directly to a resource (instead of using a role as a proxy). To learn the difference between roles and resource-based policies for cross-account access, see Cross account resource access in IAM in the *IAM User Guide*.
- **Cross-service access** – Some AWS services use features in other AWS services. For example, when you make a call in a service, it's common for that service to run applications in Amazon EC2 or store objects in Amazon S3. A service might do this using the calling principal's permissions, using a service role, or using a service-linked role.
  - **Forward access sessions (FAS)** – When you use an IAM user or role to perform actions in AWS, you are considered a principal. When you use some services, you might perform an action that then initiates another action in a different service. FAS uses the permissions of the principal calling an AWS service, combined with the requesting AWS service to make requests to downstream services. FAS requests are only made when a service receives a request that requires interactions with other AWS services or resources to complete. In this case, you must have permissions to perform both actions. For policy details when making FAS requests, see Forward access sessions.
  - **Service role** – A service role is an IAM role that a service assumes to perform actions on your behalf. An IAM administrator can create, modify, and delete a service role from within IAM. For more information, see Create a role to delegate permissions to an AWS service in the *IAM User Guide*.
  - **Service-linked role** – A service-linked role is a type of service role that is linked to an AWS service. The service can assume the role to perform an action on your behalf. Service-linked roles appear in your AWS account and are owned by the service. An IAM administrator can view, but not edit the permissions for service-linked roles.
- **Applications running on Amazon EC2** – You can use an IAM role to manage temporary credentials for applications that are running on an EC2 instance and making AWS CLI or AWS API requests. This is preferable to storing access keys within the EC2 instance. To assign an AWS role to an EC2 instance and make it available to all of its applications, you create an instance profile that is attached to the instance. An instance profile contains the role and enables programs that are running on the EC2 instance to get temporary credentials. For more information, see Use an IAM role to grant permissions to applications running on Amazon EC2 instances in the *IAM User Guide*.

# Managing access using policies

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

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

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

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

## Identity-based policies

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

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

## Resource-based policies

Resource-based policies are JSON policy documents that you attach to a resource. Examples of resource-based policies are IAM *role trust policies* and Amazon S3 *bucket policies*. In services that

support resource-based policies, service administrators can use them to control access to a specific resource. For the resource where the policy is attached, the policy defines what actions a specified principal can perform on that resource and under what conditions. You must specify a principal in a resource-based policy. Principals can include accounts, users, roles, federated users, or AWS services.

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

AWS Resource Explorer doesn't support resource-based policies.

### Access control lists (ACLs)

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

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

AWS Resource Explorer doesn't support ACLs.

### Other policy types

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

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

any or all of your accounts. The SCP limits permissions for entities in member accounts, including each AWS account root user. For more information about Organizations and SCPs, see Service control policies in the *AWS Organizations User Guide*.

- **Resource control policies (RCPs)** – RCPs are JSON policies that you can use to set the maximum available permissions for resources in your accounts without updating the IAM policies attached to each resource that you own. The RCP limits permissions for resources in member accounts and can impact the effective permissions for identities, including the AWS account root user, regardless of whether they belong to your organization. For more information about Organizations and RCPs, including a list of AWS services that support RCPs, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Session policies are advanced policies that you pass as a parameter when you programmatically create a temporary session for a role or federated user. The resulting session's permissions are the intersection of the user or role's identity-based policies and the session policies. Permissions can also come from a resource-based policy. An explicit deny in any of these policies overrides the allow. For more information, see Session policies in the *IAM User Guide*.

### Multiple policy types

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

## How Resource Explorer works with IAM

Before you use IAM to manage access to AWS Resource Explorer, you should understand what IAM features are available to use with Resource Explorer. To get a high-level view of how Resource Explorer and other AWS services work with IAM, see AWS services that work with IAM in the *IAM User Guide*.

**Topics**

- Resource Explorer identity-based policies
- Authorization based on Resource Explorer tags
- Resource Explorer IAM roles

Like any other AWS service, Resource Explorer requires permissions to use its operations to interact with your resources. To search, users must have permission to retrieve the details about a view, and

also to search using the view. To create indexes or views, or to modify them or any other Resource Explorer settings, you must have additional permissions.

Assign IAM identity-based policies that grant those permissions to the appropriate IAM principals. Resource Explorer provides several managed policies that pre-define common sets of permissions. You can assign these to your IAM principals.

## Resource Explorer identity-based policies

With IAM identity-based policies, you can specify allowed or denied actions against specific resources and the conditions under which those actions are allowed or denied. Resource Explorer supports specific actions, resources, and condition keys. To learn about all of the elements that you use in a JSON policy, see IAM JSON policy elements reference in the *IAM User Guide*.

### Actions

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

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

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

Policy actions in Resource Explorer use the `resource-explorer-2` service prefix before the action. For example, to grant someone permission to search using a view, with the Resource Explorer `Search` API operation, you include the `resource-explorer-2:Search` action in a policy assigned to that principal. Policy statements must include either an `Action` or `NotAction` element. Resource Explorer defines its own set of actions that describe tasks that you can perform with this service. These align with the Resource Explorer API operations.

To specify multiple actions in a single statement, separate them with commas as shown in the following example.

```
"Action": [
      "resource-explorer-2:action1",
```

```
      "resource-explorer-2:action2"
]
```

You can specify multiple actions using wildcard characters (*). For example, to specify all actions that begin with the word `Describe`, include the following action.

```
"Action": "resource-explorer-2:Describe*"
```

For a list of Resource Explorer actions, see Actions Defined by AWS Resource Explorer in the *AWS Service Authorization Reference*.

**Resources**

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

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

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

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

**View**

The primary Resource Explorer resource type is the *view*.

The Resource Explorer view resource has the following ARN format.

```
arn:${Partition}:resource-explorer-2:${Region}:${Account}:view/${ViewName}/${unique-id}
```

The Resource Explorer ARN format is shown in the following example.

```
arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-Search-
View/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
```

> **Note**
>
> The ARN for a view includes a unique identifier at the end to ensure that every view is unique. This helps ensure that an IAM policy that granted access to an old, deleted view can't be used to accidentally grant access to a new view that happens to have the same name as the old view. Every new view receives a new, unique ID at the end to ensure that ARNs are never reused.

For more information about the format of ARNs, see Amazon Resource Names (ARNs).

You use IAM identity-based policies assigned to the IAM principals and specify the view as the `Resource`. Doing this lets you grant search access through one view to one set of principals, and access through a completely different view to a different set of principals.

For example, to grant permission to a single view named `ProductionResourcesView` in an IAM policy statement, first get the Amazon resource name (ARN) of the view. You can use the **Views** page in the console to view the details of a view, or invoke the `ListViews` operation to retrieve the full ARN of the view you want. Then, include it in a policy statement, like that shown in the following example that grants permission to modify the definition of only one view.

```
"Effect": "Allow",
"Action": "UpdateView",
"Resource": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/
ProductionResourcesView/<unique-id>"
```

To allow the actions on ***all*** views that belong to a specific account, use the wildcard character (*) in the relevant part of the ARN. The following example grants search permission to all views in a specified AWS Region and account.

```
"Effect": "Allow",
"Action": "Search",
"Resource": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/*"
```

Some Resource Explorer actions, such as `CreateView`, aren't performed against a specific resource, because, as in the following example, the resource doesn't exist yet. In such cases, you must use the wildcard character (*) for the entire resource ARN.

```
"Effect": "Allow",
```

```
"Action": "resource-explorer-2:CreateView"
"Resource": "*"
```

If you specify a path that ends in a wildcard character, then you can restrict the `CreateView` operation to creating views with only the approved path. The following example policy piece shows how to allow the principal to create views only in the path `view/ProductionViews/`.

```
"Effect": "Allow",
"Action": "resource-explorer-2:CreateView"
"Resource": "arn:aws:resource-explorer-2:us-
east-1:123456789012:view/ProductionViews/*""
```

**Index**

Another resource type that you can use to control access to Resource Explorer functionality is the index.

The primary way that you interact with the index is to turn on Resource Explorer in an AWS Region by creating an index in that Region. After that, you do almost everything else by interacting with the view.

One thing that you can do with the index is to control who can ***create*** views in each Region.

> **Note**
>
> After you create a view, IAM authorizes all other view actions against only the ARN of the view, and not the index.

The index has an ARN that you can reference in a permission policy. A Resource Explorer index ARN has the following format.

```
arn:${Partition}:resource-explorer-2:${Region}:${Account}:index/${unique-id}
```

See the following example of an Resource Explorer index ARN.

```
arn:aws:resource-explorer-2:us-east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-
abcd22222222
```

Some Resource Explorer actions check authentication against multiple resource types. For example, the CreateView operation authorizes against both the ARN of the index and the ARN of the view as it will be after Resource Explorer creates it. To grant administrators permission to manage the Resource Explorer service, you can use `"Resource": "*"` to authorize actions for any resource, index, or view.

Alternatively, you can restrict a principal to only being able to work with specified Resource Explorer resources. For example, to limit actions to only Resource Explorer resources in a specified Region, you can include an ARN template that matches both the index and the view, but calls out only a single Region. In the following example, the ARN matches both indexes or views in only the `us-west-2` Region of the specified account. Specify the Region in the third field of the ARN, but use a wildcard character (*) in the final field to match any resource type.

```
"Resource": "arn:aws:resource-explorer-2:us-west-2:123456789012:*
```

For more information, see Resources Defined by AWS Resource Explorer in the *AWS Service Authorization Reference*. To learn with which actions you can specify the ARN of each resource, see Actions Defined by AWS Resource Explorer.

**Condition keys**

Resource Explorer doesn't provide any service-specific condition keys, but it does support using some global condition keys. To see all AWS global condition keys, see AWS global condition context keys in the *IAM User Guide*.

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

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

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

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

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

To see a list of the condition keys that you can use with Resource Explorer, see Condition Keys for AWS Resource Explorer in the *AWS Service Authorization Reference*. To learn which actions and resources you can use a condition key with, see Actions Defined by AWS Resource Explorer.

#### Examples

To view examples of Resource Explorer identity-based policies, see AWS Resource Explorer identity-based policy examples.

### Authorization based on Resource Explorer tags

You can attach tags to Resource Explorer views or pass tags in a request to Resource Explorer. To control access based on tags, you provide tag information in the condition element of a policy using the `resource-explorer-2:ResourceTag/`*`key-name`*, `aws:RequestTag/`*`key-name`*, or `aws:TagKeys` condition keys. For more information about tagging Resource Explorer resources, see Adding tags to views. For using tag-based authorization in Resource Explorer, see Using tag-based authorization to control access to your views.

### Resource Explorer IAM roles

An IAM role is a principal within your AWS account that has specific permissions.

#### Using temporary credentials with Resource Explorer

You can use temporary credentials to sign in with federation, assume an IAM role, or to assume a cross-account role. You obtain temporary security credentials by calling AWS Security Token Service (AWS STS) API operations such as AssumeRole or GetFederationToken.

Resource Explorer supports using temporary credentials.

#### Service-linked roles

Service-linked roles allow AWS services to access resources in other services to complete an action on your behalf. Service-linked roles appear in your IAM account and are owned by the service. An IAM administrator can view but not edit the permissions for service-linked roles.

Resource Explorer uses service-linked roles to perform its work. For details about Resource Explorer service-linked roles, see Using service-linked roles for Resource Explorer.

## AWS Resource Explorer identity-based policy examples

By default, AWS Identity and Access Management (IAM) principals, such as roles, groups, and users, don't have permission to create or modify Resource Explorer resources. They also can't perform tasks using the AWS Management Console, AWS Command Line Interface (AWS CLI), or AWS API. An IAM administrator must create IAM policies that grant principals permission to perform specific API operations on the specified resources they need. Then, the administrator must assign those policies to the IAM principals that require those permissions.

To provide access, add permissions to your users, groups, or roles:

- Users and groups in AWS IAM Identity Center:

  Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.
- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.
- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

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

**Topics**


- Policy best practices
- Using the Resource Explorer console
- Granting access to a view based on tags
- Granting access to create a view based on tags
- Allow principals to view their own permissions

## Policy best practices

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

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

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

## Using the Resource Explorer console

For principals to search in the AWS Resource Explorer console, they must have a minimum set of permissions. If you don't create an identity-based policy with the minimum required permissions, then the Resource Explorer console doesn't function as intended for principals in the account.

You can use the AWS managed policy named `AWSResourceExplorerReadOnlyAccess` to grant the ability to use the Resource Explorer console to search using any view in the account. To grant permissions to search with only a single view, see Granting access to Resource Explorer views for search, and the examples in the following two sections.

You don't need to allow minimum console permissions for principals that are making calls only to the AWS CLI or the AWS API. Instead, you can choose to grant access to only those actions that match the API operations that the principals need to perform.

## Granting access to a view based on tags

In this example, you want to grant access to a Resource Explorer view in your AWS account to principals in the account. To do this you assign IAM identity-based policies to the principals that you want to be able to search in Resource Explorer. The following example IAM policy grants access to any request where the `Search-Group` tag attached to the calling principal exactly matches the value for that same tag attached to the view used in the request.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                ":GetView",
                ":Search"
            ],
            "Resource": "arn:aws::*:*:view/*",
            "Condition": {
                "StringEquals": {"aws:ResourceTag/Search-Group":
 "${aws:PrincipalTag/Search-Group}"}
            }
        }
```

```
    ]
}
```

You can assign this policy to the IAM principals in your account. If a principal with the tag `Search-Group=A` attempts to search using a Resource Explorer view, the view must also be tagged `Search-Group=A`. If it's not, then the principal is denied access. The condition tag key `Search-Group` matches both `Search-group` and `search-group` because condition key names are not case-sensitive. For more information, see IAM JSON Policy Elements: Condition in the *IAM User Guide*.

> **Important**
>
> To see your resources in unified search results in the AWS Management Console, principals must have both `GetView` and `Search` permissions for the default view in the AWS Region that contains the aggregator index. The simplest way to grant those permissions is to leave the default resource-based permission that was attached to the view when you turned on Resource Explorer using Quick or Advanced setup.
> For this scenario, you could consider setting the default view to filter out sensitive resources and then setting up additional views to which you grant tag-based access as described in the previous example.

## Granting access to create a view based on tags

In this example, you want to allow only principals that are tagged the same as the index to be able to create views in the AWS Region that contains the index. To do this, create identity-based permissions to allow the principals to search with views.

Now you're ready to grant permissions to create a view. You can add the statements in this example to the same permission policy that you use to grant `Search` permissions to appropriate principals. The actions are allowed or denied based on the tags attached to the principals calling the operations and index that the view is to be associated with. The following example IAM policy denies any request to create a view when the value of the `Allow-Create-View` tag attached to the caller's principal doesn't exactly match the value for that same tag attached to the index in the Region in which the view is created.

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": ":CreateView",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {"aws:ResourceTag/Allow-Create-View":
 "${aws:PrincipalTag/Allow-Create-View}"}
            }
        }
    ]
}
```

## Allow principals to view their own permissions

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

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

```
            "Action": [
                "iam:GetGroupPolicy",
                "iam:GetPolicyVersion",
                "iam:GetPolicy",
                "iam:ListAttachedGroupPolicies",
                "iam:ListGroupPolicies",
                "iam:ListPolicyVersions",
                "iam:ListPolicies",
                "iam:ListUsers"
            ],
            "Resource": "*"
        }
    ]
}
```

## Example service control policies for AWS Organizations and Resource Explorer

AWS Resource Explorer supports service control policies (SCPs). SCPs are policies that you attach to elements in an organization to manage permissions within that organization. An SCP applies to all AWS accounts in an organization under the element to which you attach the SCP. SCPs offer central control over the maximum available permissions for all accounts in your organization. They can help you to ensure your AWS accounts stay within your organization's access control guidelines. For more information, see Service control policies in the *AWS Organizations User Guide*.

### Prerequisites

To use SCPs, you must first do the following:

- Enable all features in your organization. For more information, see Enabling all features in your organization in the *AWS Organizations User Guide*.
- Enable SCPs for use within your organization. For more information, see Enabling and disabling policy types in the *AWS Organizations User Guide*.
- Create the SCPs that you need. For more information about creating SCPs, see Creating and updating SCPs in the *AWS Organizations User Guide*.

## Example service control policies

The following example shows how you can use attribute-based access control (ABAC) to control access to the administrative operations of Resource Explorer. This example policy denies access to all Resource Explorer operations except the two permissions required to search, `resource-explorer-2:Search` and `resource-explorer-2:GetView`, unless the IAM principal making the request is tagged `ResourceExplorerAdmin=TRUE`. For a more complete discussion of using ABAC with Resource Explorer, see Using tag-based authorization to control access to your views.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        ":AssociateDefaultView",
        ":BatchGetView",
        ":CreateIndex",
        ":CreateView",
        ":DeleteIndex",
        ":DeleteView",
        ":DisassociateDefaultView",
        ":GetDefaultView",
        ":GetIndex",
        ":ListIndexes",
        ":ListSupportedResourceTypes",
        ":ListTagsForResource",
        ":ListViews",
        ":TagResource",
        ":UntagResource",
        ":UpdateIndexType",
        ":UpdateView""
      ],
      "Resource": [
        "*"
      ],
      "Condition": {
          "StringNotEqualsIgnoreCase": {"aws:PrincipalTag/ResourceExplorerAdmin":
 "TRUE"}
      }
```

```
    ]
}
```

## AWS managed policies for AWS Resource Explorer

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

### General AWS managed policies that include Resource Explorer permissions

- AdministratorAccess – Grants full access to AWS services and resources.
- ReadOnlyAccess – Grants read-only access to AWS services and resources.
- ViewOnlyAccess – Grants permissions to view resources and basic metadata for AWS services.

> **Note**
>
> The Resource Explorer `Get*` permissions included in the `ViewOnlyAccess` policy perform like `List` permissions although they return only a single value, because a Region can contain only one index and one default view.

### AWS managed policies for Resource Explorer

- AWSResourceExplorerFullAccess

- AWSResourceExplorerReadOnlyAccess
- AWSResourceExplorerServiceRolePolicy

## AWS managed policy: AWSResourceExplorerFullAccess

You can assign the `AWSResourceExplorerFullAccess` policy to your IAM identities.

This policy grants permissions that allow full administrative control of the Resource Explorer service. You can perform all tasks involved in turning on and managing Resource Explorer in the AWS Regions in your account. With this policy, the Resource Explorer console shows information from other integrated AWS services and allows you to perform actions such as creating an application.

### Permissions details

This policy includes permissions that allow all actions for Resource Explorer, including turning on and turning off Resource Explorer in AWS Regions, creating or deleting an aggregator index for the account, creating, updating, and deleting views, and searching. This policy also includes permissions that are not part of Resource Explorer:

- `ec2:DescribeRegions` – allows Resource Explorer to access the details about the Regions in your account.
- `ram:ListResources` – allows Resource Explorer to list the resource shares that resources are part of.
- `ram:GetResourceShares` – allows Resource Explorer to identify details about the resource shares that you own or that are shared with you.
- `iam:CreateServiceLinkedRole` – allows Resource Explorer to create the required service-linked role when you turn on Resource Explorer by creating the first index.
- `organizations:DescribeOrganization` – allows Resource Explorer to access information about your organization.

To see the latest version of this AWS managed policy, see `AWSResourceExplorerFullAccess` in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: AWSResourceExplorerReadOnlyAccess

You can assign the `AWSResourceExplorerReadOnlyAccess` policy to your IAM identities.

This policy grants read-only permissions that allows users to discover their resources with basic search access, and access other integrated AWS services in the Resource Explorer console.

**Permissions details**

This policy includes permissions that allow users to perform the Resource Explorer `Get*`, `List*`, and `Search` operations to view information about Resource Explorer components and configuration settings, but doesn't allow users to change them. Users can also search. This policy also includes two permissions that are not part of Resource Explorer:

- `ec2:DescribeRegions` – allows Resource Explorer to access the details about the Regions in your account.
- `ram:ListResources` – allows Resource Explorer to list the resource shares that resources are part of.
- `ram:GetResourceShares` – allows Resource Explorer to identify details about the resource shares that you own or that are shared with you.
- `organizations:DescribeOrganization` – allows Resource Explorer to access information about your organization.

To see the latest version of this AWS managed policy, see `AWSResourceExplorerReadOnlyAccess` in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: AWSResourceExplorerServiceRolePolicy

You can't attach `AWSResourceExplorerServiceRolePolicy` to any IAM entities yourself. This policy can be attached only to a service-linked role that allows Resource Explorer to perform actions on your behalf. For more information, see Using service-linked roles for Resource Explorer.

This policy grants the permissions required for Resource Explorer to retrieve information about your resources. Resource Explorer populates the indexes it maintains in each AWS Region that you register.

To see the latest version of this AWS managed policy, see `AWSResourceExplorerServiceRolePolicy` in the IAM console or `AWSResourceExplorerServiceRolePolicy` in the *AWS Managed Policy Reference Guide*.

## AWS managed policy: AWSResourceExplorerOrganizationsAccess

You can assign `AWSResourceExplorerOrganizationsAccess` to your IAM identities.

This policy grants administrative permissions to Resource Explorer and grants read-only permissions to other AWS services to support this access. The AWS Organizations administrator needs these permissions to set up and manage multi-account search in the console.

**Permissions details**

This policy includes permissions that allow administrators to set up multi-account search for the organization:

- `ec2:DescribeRegions` – Allows Resource Explorer to access the details about the Regions in your account.
- `ram:ListResources` – Allows Resource Explorer to list the resource shares that resources are part of.
- `ram:GetResourceShares` – Allows Resource Explorer to identify details about the resource shares that you own or that are shared with you.
- `organizations:ListAccounts` – Allows Resource Explorer to identify the accounts within an organization.
- `organizations:ListRoots` – Allows Resource Explorer to identify the root accounts within an organization.
- `organizations:ListOrganizationalUnitsForParent` – Allows Resource Explorer to identify the organizational units (OUs) in a parent organizational unit or root.
- `organizations:ListAccountsForParent` – Allows Resource Explorer to identify the accounts in an organization that are contained by the specified target root or an OU.
- `organizations:ListDelegatedAdministrators` – Allows Resource Explorer to identify the AWS accounts that are designated as delegated administrators in this organization.
- `organizations:ListAWSServiceAccessForOrganization` – Allows Resource Explorer to identify a list of the AWS services that are enabled to integrate with your organization.
- `organizations:DescribeOrganization` – Allows Resource Explorer to retrieve information about the organization that the user's account belongs to.
- `organizations:EnableAWSServiceAccess` – Allows Resource Explorer to enable the integration of an AWS service (the service that is specified by `ServicePrincipal`) with AWS Organizations.
- `organizations:DisableAWSServiceAccess` – Allows Resource Explorer to disable the integration of an AWS service (the service that is specified by ServicePrincipal) with AWS Organizations.

- `organizations:RegisterDelegatedAdministrator` – Allows Resource Explorer to enable the specified member account to administer the organization's features of the specified AWS service.
- `organizations:DeregisterDelegatedAdministrator` – Allows Resource Explorer to remove the specified member AWS account as a delegated administrator for the specified AWS service.
- `iam:GetRole` – Allows Resource Explorer to retrieve information about the specified role, including the role's path, GUID, ARN, and the role's trust policy that grants permission to assume the role.
- `iam:CreateServiceLinkedRole` – Allows Resource Explorer to create the required service-linked role when you turn on Resource Explorer by creating the first index.

To see the latest version of this AWS managed policy, see `AWSResourceExplorerOrganizationsAccess` in the IAM console.

## Resource Explorer updates to AWS managed policies

View details about updates to AWS managed policies for Resource Explorer since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the Resource Explorer Document history page.

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to allow Resource Explorer to manage indexes and views | Resource Explorer modified the permissions in the service-linked role policy `AWSResourceExplorerServiceRolePolicy`. The following permissions were added that allow Resource Explorer to create, manage, and delete indexes and views:<br>• `UpdateIndexType`<br>• `CreateIndex` | July 23, 2025 |

| Change | Description | Date |
| --- | --- | --- |
| | • `CreateView`<br>• `AssociateDefaultView`<br>• `DeleteIndex` | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer modified the permissions in the service-linked role policy AWSResourceExplorerServiceRolePolicy. Permissions were added that allows Resource Explorer to view additional resource types:<br>• `appconfig:listapplications`<br>• `appconfig:listdeploymentstrategies`<br>• `ce:getanomalymonitors`<br>• `ce:getanomalysubscriptions`<br>• `cloudformation:listresources`<br>• `cloudfront:listcontinuousdeploymentpolicies`<br>• `cloudtrail:listchannels`<br>• `codedeploy:listapplications`<br>• `codedeploy:listdeploymentconfigs`<br>• `events:listarchives`<br>• `events:listendpoints` | May 7, 2025 |

| Change | Description | Date |
| --- | --- | --- |
| | • gamelift:listlocations<br>• groundstation:listmissionprofiles<br>• inspector:listassessmenttemplates<br>• iot:listcacertificates<br>• iot:listcertificates<br>• iotdeviceadvisor:listsuitedefinitions<br>• iotfleetwise:listdecodermanifests<br>• iotfleetwise:listmodelmanifests<br>• iotfleetwise:listsignalcatalogs<br>• lightsail:getbuckets<br>• lightsail:getcertificates<br>• managedblockchain:listaccessors<br>• oam:listsinks<br>• omics:listreferencestores<br>• omics:listrungroups<br>• omics:listworkflows | |

| Change | Description | Date |
| --- | --- | --- |
| | • personalize:listsolutions<br>• pipes:listpipes<br>• scheduler:listschedulegroups<br>• scheduler:listschedules<br>• schemas:listdiscoverers<br>• transfer:listcertificates<br>• transfer:listconnectors<br>• transfer:listprofiles | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows Resource Explorer to view additional resource types:<br>• cloud9:environment<br>• cloudtrail:eventdatastore<br>• connect:instance/rule<br>• connect:instance/task-template<br>• connect:phone-number<br>• datapipeline:pipeline<br>• dax:cache<br>• devicefarm:project<br>• devicefarm:testgrid-project<br>• ds:directory<br>• ec2:ipam-resource-discovery<br>• ec2:ipam-resource-discovery-association<br>• elasticloadbalancing:listener-rule/net | March 21, 2025 |

| Change | Description | Date |
| --- | --- | --- |
| | • events:connection<br>• forecast:dataset-import-job<br>• forecast:forecast<br>• forecast:forecast-export-job<br>• forecast:predictor<br>• forecast:predictor-backtest-export-job<br>• geo:map<br>• grafana:workspaces<br>• groundstation:dataflow-endpoint-group<br>• iot:ruledestination<br>• iotfleetwise:vehicle<br>• ivschat:logging-configuration<br>• ivschat:room<br>• lookoutmetrics:AnomalyDetector<br>• m2:env<br>• outposts:site<br>• quicksight:theme<br>• route53-recovery-control:cluster<br>• route53-recovery-control:controlpanel/safetyrule | |

| Change | Description | Date |
| --- | --- | --- |
| | • route53-recovery-readiness:readiness-check<br>• route53resolver:firewall-rule-group-association<br>• rum:appmonitor<br>• sagemaker:app-image-config<br>• servicediscovery:service<br>• synthetics:group<br>• transfer:agreement<br>• transfer:profile<br>• workspaces:connectionalias | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows Resource Explorer to view additional resource types:<br>• cloud9:listenvironments<br>• cloudtrail:listeventdatastores<br>• codeconnections:listconnections<br>• connect:listphonenumbersv2<br>• connect:listrules<br>• connect:listtasktemplates<br>• datapipeline:listpipelines<br>• dax:describeclusters<br>• devicefarm:listprojects<br>• devicefarm:listtestgridprojects<br>• ec2:describeipamresourcediscoveries<br>• ec2:describeipamresourcediscoveryassociations | March 28, 2025 |

| Change | Description | Date |
| --- | --- | --- |
| | • events:listapidestinations<br>• events:listconnections<br>• forecast:listdatasetimportjobs<br>• forecast:listforecastexportjobs<br>• forecast:listforecasts<br>• forecast:listpredictorbacktestexportjobs<br>• forecast:listpredictors<br>• geo:listmaps<br>• grafana:listworkspaces<br>• groundstation:listdatafllowendpointgroups<br>• iotfleetwise:listvehicles<br>• iottwinmaker:listsyncjobs<br>• ivschat:listloggingconfigurations<br>• ivschat:listrooms<br>• kms:listaliases | |

| Change | Description | Date |
| --- | --- | --- |
| | • license-manager:listdistributedgrants<br>• lightsail:getcontainerservices<br>• lightsail:getdisks<br>• lookoutmetrics:listanomalydetectors<br>• m2:listenvironments<br>• macie2:listallowlists<br>• organizations:listorganizationalunitsforparent<br>• organizations:listroots<br>• outposts:listsites<br>• quicksight:listthemes<br>• route53-recovery-control-config:listclusters<br>• route53-recovery-control-config:listcontrolpanels<br>• route53-recovery-control-config:listsafetyrules<br>• route53-recovery-readiness:listreadinesschecks | |

| Change | Description | Date |
| --- | --- | --- |
| | • route53resolver:listfirewallrulegroupassociations<br>• rum:listappmonitors<br>• s3:listmultiregionaccesspoints<br>• sagemaker:listappimageconfigs<br>• servicediscovery:listservices<br>• synthetics:listgroups<br>• timestream:listscheduledqueries<br>• transfer:listagreements<br>• transfer:listservers<br>• workspaces:describeconnectionaliases | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows Resource Explorer to view additional resource types:<br>• acm:certificate<br>• codepipeline:webhook<br>• comprehend:document-classifier<br>• comprehend:entity-recognizer<br>• databrew:job<br>• databrew:project<br>• dataexchange:data-sets<br>• dms:es<br>• dms:subgrp<br>• elasticmapreduce:cluster<br>• emr-containers:virtualclusters<br>• frauddetector:external-model<br>• frauddetector:model<br>• fsx:file-system<br>• glacier:vaults<br>• glue:crawler | January 6, 2025 |

| Change | Description | Date |
| --- | --- | --- |
| | • greengrass:connectorsDefinition<br>• greengrass:coresDefinition<br>• greengrass:devicesDefinition<br>• greengrass:functionsDefinition<br>• greengrass:loggersDefinition<br>• greengrass:resourcesDefinition<br>• greengrass:subscriptionsDefinition<br>• mq:broker<br>• route53:domain<br>• ses:contact-list<br>• ses:configuration-set<br>• ses:identity<br>• storagegateway:gateway | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows Resource Explorer to view additional resource types:<br>• IamAction(value=apigateway:Get)<br>• airflow:ListEnvironments<br>• appflow:ListFlows<br>• appmesh:ListMeshes<br>• appmesh:ListVirtualNodes<br>• appmesh:ListVirtualServices<br>• auditmanager:GetAccountStatus<br>• auditmanager:ListAssessments<br>• backup:ListBackupVaults<br>• codeguru-reviewer:ListRepositoryAssociations<br>• connect:ListInstances<br>• connect:ListQuickConnects<br>• connect:ListUsers | November 21, 2024 |

| Change | Description | Date |
| --- | --- | --- |
| | • databrew:ListSchedules<br>• datasync:ListLocations<br>• datasync:ListTasks<br>• dms:DescribeEndpoints<br>• dms:DescribeReplicationInstances<br>• dms:DescribeReplicationTasks<br>• eks:ListClusters<br>• gamelift:DescribeGameSessionQueues<br>• gamelift:DescribeMatchmakingConfigurations<br>• gamelift:DescribeMatchmakingRuleSets<br>• gamelift:ListBuilds<br>• glue:ListMLTransforms<br>• groundstation:ListConfigs<br>• guardduty:ListDetectors<br>• guardduty:ListFilters<br>• guardduty:ListIPSets | |

| Change | Description | Date |
| --- | --- | --- |
| | • guardduty:ListThreatIntelSets<br>• iotsitewise:ListDashboards<br>• iotsitewise:ListPortals<br>• iotsitewise:ListProjects<br>• iotwireless:ListServiceProfiles<br>• ivs:ListRecordingConfigurations<br>• kendra:ListIndices<br>• macie2:ListCustomDataIdentifiers<br>• macie2:ListFindingsFilters<br>• memorydb:DescribeSubnetGroups<br>• mobiletargeting:GetCampaigns<br>• proton:ListEnvironmentAccountConnections<br>• quicksight:DescribeAccountSubscription<br>• quicksight:ListDataSets<br>• quicksight:ListDataSources | |

| Change | Description | Date |
| --- | --- | --- |
| | • quicksight:ListTemplates<br>• ram:GetResourceShares<br>• robomaker:ListRobotApplications<br>• robomaker:ListSimulationApplications<br>• route53resolver:ListResolverQueryLogConfigs<br>• sagemaker:ListDomains<br>• sagemaker:ListEndpoints<br>• sagemaker:ListFeatureGroups<br>• sagemaker:ListImages<br>• sagemaker:ListPipelines<br>• transfer:ListWorkflows<br>• workspaces:DescribeWorkspaces | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy - Updated policy permissions to view additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows Resource Explorer to view additional resource types:<br>• apprunner:ListVpcConnectors<br>• backup:ListReportPlans<br>• emr-serverless:ListApplications<br>• events:ListEventBuses<br>• geo:ListPlaceIndexes<br>• geo:ListTrackers<br>• greengrass:ListComponents<br>• greengrass:ListComponentVersions<br>• iot:ListRoleAliases<br>• iottwinmaker:ListComponentTypes<br>• iottwinmaker:ListEntities<br>• iottwinmaker:ListScenes<br>• kafka:ListConfigurations | December 12, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| | • kms:ListKeys<br>• kinesisanalytics:ListApplications<br>• lex:ListBots<br>• lex:ListBotAliases<br>• mediapackage-vod:ListPackagingConfigurations<br>• mediapackage-vod:ListPackagingGroups<br>• mq:ListBrokers<br>• personalize:ListDatasetGroups<br>• personalize:ListDatasets<br>• personalize:ListSchemas<br>• route53:ListHealthChecks<br>• route53:ListHostedZones<br>• secretsmanager:ListSecrets | |
| New managed policy | Resource Explorer added the following AWS managed policy:<br>• AWSResourceExplorerOrganizationsAccess | November 14, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| Updated managed policies | Resource Explorer updated the following AWS managed policies to support multi-account search:<br>• AWSResourceExplorerFullAccess<br>• AWSResourceExplorerReadOnlyAccess | November 14, 2023 |
| AWSResourceExplorerServiceRolePolicy – Updated policy to support multi-account search with Organizations | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows the Resource Explorer to support multi-account search with Organizations:<br>• `organizations:ListAWSServiceAccessForOrganization`<br>• `organizations:DescribeAccount`<br>• `organizations:DescribeOrganization`<br>• `organizations:ListAccounts`<br>• `organizations:ListDelegatedAdministrators` | November 14, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy – Updated policy to support additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows the service to index the following resource types:<br>• accessanalyzer:analyzer<br>• acmpca:certificateauthority<br>• amplify:app<br>• amplify:backendenvironment<br>• amplify:branch<br>• amplify:domainassociation<br>• amplifyuibuilder:component<br>• amplifyuibuilder:theme<br>• appintegrations:eventintegration<br>• apprunner:service<br>• appstream:appblock<br>• appstream:application<br>• appstream:fleet<br>• appstream:imagebuilder<br>• appstream:stack<br>• appsync:graphqlapi<br>• aps:rulegroupsnamespace<br>• aps:workspace<br>• apigateway:restapi<br>• apigateway:deployment | October 17, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| | • athena:datacatalog<br>• athena:workgroup<br>• autoscaling:autosc alinggroup<br>• backup:backupplan<br>• batch:computeenvironment<br>• batch:jobqueue<br>• batch:schedulingpolicy<br>• cloudformation:stack<br>• cloudformation:stackset<br>• cloudfront:fieldlevelencryp tionconfig<br>• cloudfront:fieldlevelencryp tionprofile<br>• cloudfront:originaccesscont rol<br>• cloudtrail:trail<br>• codeartifact:domain<br>• codeartifact:repository<br>• codecommit:repository<br>• codeguruprofiler:profilingg roup<br>• codestarconnection s:connection<br>• databrew:dataset<br>• databrew:recipe<br>• databrew:ruleset<br>• detective:graph<br>• directoryservices:directory<br>• ec2:carriergateway | |

| Change | Description | Date |
| --- | --- | --- |
| | • ec2:verifiedaccessendpoint<br>• ec2:verifiedaccessgroup<br>• ec2:verifiedaccessinstance<br>• ec2:verifiedaccesstrustprovider<br>• ecr:repository<br>• elasticache:cachesecuritygroup<br>• elasticfilesystem:accesspoint<br>• events:rule<br>• evidently:experiment<br>• evidently:feature<br>• evidently:launch<br>• evidently:project<br>• finspace:environment<br>• firehose:deliverystream<br>• faultinjectionsimulator:experimenttemplate<br>• forecast:datasetgroup<br>• forecast:dataset<br>• frauddetector:detector<br>• frauddetector:entitytype<br>• frauddetector:eventtype<br>• frauddetector:label<br>• frauddetector:outcome<br>• frauddetector:variable<br>• gamelift:alias<br>• globalaccelerator:accelerator | |

| Change | Description | Date |
| --- | --- | --- |
| | • globalaccelerator: endpointgroup<br>• globalaccelerator:listener<br>• glue:database<br>• glue:job<br>• glue:table<br>• glue:trigger<br>• greengrass:group<br>• healthlake:fhirdatastore<br>• iam:virtualmfadevice<br>• imagebuilder:compo nentbuildversion<br>• imagebuilder:component<br>• imagebuilder:conta inerrecipe<br>• imagebuilder:distributionco nfiguration<br>• imagebuilder:image buildversion<br>• imagebuilder:imagepipeline<br>• imagebuilder:imagerecipe<br>• imagebuilder:image<br>• imagebuilder:infrastructure configuration<br>• iot:authorizer<br>• iot:jobtemplate<br>• iot:mitigationaction<br>• iot:provisioningtemplate<br>• iot:securityprofile<br>• iot:thing | |

| Change | Description | Date |
| --- | --- | --- |
| | • iot:topicruledestination<br>• iotanalytics:channel<br>• iotanalytics:dataset<br>• iotanalytics:datastore<br>• iotanalytics:pipeline<br>• iotevents:alarmmodel<br>• iotevents:detectormodel<br>• iotevents:input<br>• iotsitewise:assetmodel<br>• iotsitewise:asset<br>• iotsitewise:gateway<br>• iottwinmaker:workspace<br>• ivs:channel<br>• ivs:streamkey<br>• kafka:cluster<br>• kinesisvideo:stream<br>• lambda:alias<br>• lambda:layerversion<br>• lambda:layer<br>• lookoutmetrics:alert<br>• lookoutvision:project<br>• mediapackage:channel<br>• mediapackage:origi nendpoint<br>• mediatailor:playba ckconfiguration<br>• memorydb:acl<br>• memorydb:cluster<br>• memorydb:parametergroup | |

| Change | Description | Date |
| --- | --- | --- |
| | • memorydb:user<br>• mobiletargeting:app<br>• mobiletargeting:segment<br>• mobiletargeting:template<br>• networkfirewall:firewallpolicy<br>• networkfirewall:firewall<br>• networkmanager:globalnetwork<br>• networkmanager:device<br>• networkmanager:link<br>• networkmanager:attachment<br>• networkmanager:corenetwork<br>• panorama:package<br>• qldb:journalkinesisstreamsforledger<br>• qldb:ledger<br>• rds:bluegreendeployment<br>• refactorspaces:application<br>• refactorspaces:environment<br>• refactorspaces:route<br>• refactorspaces:service<br>• rekognition:project<br>• resiliencehub:app<br>• resiliencehub:resiliencypolicy<br>• resourcegroups:group<br>• route53:recoverygroup | |

| Change | Description | Date |
| --- | --- | --- |
| | • route53:resourceset<br>• route53:firewalldomain<br>• route53:firewallrulegroup<br>• route53:resolverendpoint<br>• route53:resolverrule<br>• sagemaker:model<br>• sagemaker:notebook instance<br>• signer:signingprofile<br>• ssmincidents:responseplan<br>• ssm:inventoryentry<br>• ssm:resourcedatasync<br>• states:activity<br>• timestream:database<br>• wisdom:assistant<br>• wisdom:assistantas sociation<br>• wisdom:knowledgebase | |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy – Updated policy to support additional resource types | Resource Explorer added permissions to the service-linked role policy `AWSResourceExplorerServiceRolePolicy` that allows the service to index the following resource types:<br>• codebuild:project<br>• codepipeline:pipeline<br>• cognito:identitypool<br>• cognito:userpool<br>• ecr:repository<br>• efs:filesystem<br>• elasticbeanstalk:application<br>• elasticbeanstalk:applicationversion<br>• elasticbeanstalk:environment<br>• iot:policy<br>• iot:topicrule<br>• stepfunctions:statemachine<br>• s3:bucket | August 1, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| AWSResourceExplorerServiceRolePolicy – Updated policy to support additional resource types | Resource Explorer added permissions to the service-linked role policy AWSResourceExplorerServiceRolePolicy that allows the service to index the following resource types:<br>• elasticache:cluster<br>• elasticache:globalreplicationgroup<br>• elasticache:parametergroup<br>• elasticache:replicationgroup<br>• elasticache:reserved-instance<br>• elasticache:snapshot<br>• elasticache:subnetgroup<br>• elasticache:user<br>• elasticache:usergroup<br>• lambda:code-signing-config<br>• lambda:event-source-mapping<br>• sqs:queue | March 7, 2023 |

| Change | Description | Date |
| --- | --- | --- |
| New managed policies | Resource Explorer added the following AWS managed policies:<br>• AWSResourceExplorerFullAccess<br>• AWSResourceExplorerReadOnlyAccess<br>• AWSResourceExplorerServiceRolePolicy | November 7, 2022 |
| Resource Explorer started tracking changes | Resource Explorer started tracking changes for its AWS managed policies. | November 7, 2022 |

## Using service-linked roles for Resource Explorer

AWS Resource Explorer uses AWS Identity and Access Management (IAM) service-linked roles. A service-linked role is a unique type of IAM role that is linked directly to Resource Explorer. Service-linked roles are predefined by Resource Explorer and include all the permissions that the service requires to call other AWS services on your behalf.

A service-linked role makes configuring Resource Explorer easier because you don’t have to manually add the necessary permissions. Resource Explorer defines the permissions of its service-linked roles, and unless defined otherwise, only Resource Explorer can assume its roles. The defined permissions include both the trust policy and the permissions policy, and that permissions policy can't be assigned to any other IAM entity.

For information about other services that support service-linked roles, see AWS services that work with IAM in the *IAM User Guide*. There, look for the services that have **Yes** in the **Service-linked roles** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-linked role permissions for Resource Explorer

Resource Explorer uses the service-linked role named `AWSServiceRoleForResourceExplorer`. This role grants permissions to the Resource Explorer service to view resources and AWS CloudTrail events in your AWS account on your behalf and to index those resources to support searching.

The `AWSServiceRoleForResourceExplorer` service-linked role trusts only the service with the following service principal to assume the role:

- `resource-explorer-2.amazonaws.com`

The role permissions policy named AWSResourceExplorerServiceRolePolicy allows Resource Explorer read-only access to retrieve resource names and properties for supported AWS resources. To view the services and resources that Resource Explorer supports, see Resource types you can search for with Resource Explorer. For the complete list of all actions this role can perform, you can view the `AWSResourceExplorerServiceRolePolicy` policy in the IAM console.

A principal is an IAM entity such as a user, group, or role. If you let Resource Explorer create the service-linked role for you when it creates the index in the first Region of the account, then the principal performing the task needs only the permissions required to create the Resource Explorer index. To create the service-linked role manually using IAM, then the principal performing the task must have permission to create a service-linked role. For more information, see Service-linked role permissions in the *IAM User Guide*.

## Creating a service-linked role for Resource Explorer

You don't need to manually create a service-linked role. When you turn on Resource Explorer in the AWS Management Console, or run CreateIndex in the first AWS Region in your account using the AWS CLI or an AWS API, Resource Explorer creates the service-linked role for you.

If you delete this service-linked role, and then need to create it again, you can use the same process to re-create the role in your account. When you RegisterResourceExplorer in the first Region in your account, Resource Explorer creates the service-linked role for you again.

## Editing a service-linked role for Resource Explorer

Resource Explorer doesn't allow you to edit the `AWSServiceRoleForResourceExplorer` service-linked role. After you create a service-linked role, you can't change the name of the role because various entities might reference the role. However, you can edit the description of the role using IAM. For more information, see Editing a service-linked role in the *IAM User Guide*.

### Deleting a service-linked role for Resource Explorer

You can use the IAM console, the AWS CLI, or the AWS API to manually delete the service-linked role. To do this, you must first remove the Resource Explorer indexes from every AWS Region in your account and then you can manually delete the service-linked role.

> **Note**
>
> If the Resource Explorer service is using the role when you try to delete the resources, the deletion fails. If that happens, ensure that all indexes from all Regions are deleted, then wait for a few minutes and try the operation again.

#### To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSServiceRoleForResourceExplorer` service-linked role. For more information, see Deleting a service-linked role in the *IAM User Guide*.

### Supported Regions for Resource Explorer service-linked roles

Resource Explorer supports using service-linked roles in all of the Regions where the service is available. For more information, see AWS service endpoints in the *Amazon Web Services General Reference*.

## Troubleshooting AWS Resource Explorer permissions

Use the following information to help you diagnose and fix common issues that you might encounter when working with Resource Explorer and AWS Identity and Access Management (IAM).

#### Topics

- I am not authorized to perform an action in Resource Explorer
- I want to allow people outside of my AWS account to access my Resource Explorer resources

### I am not authorized to perform an action in Resource Explorer

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

For example, the following error occurs when someone assumes the IAM role `MyExampleRole` tries to use the console to view details about a view but does not have `resource-explorer-2:GetView` permission.

```
User: arn:aws:iam::123456789012:role/MyExampleRole is not authorized to perform:
 resource-explorer-2:GetView on resource: arn:aws:resource-explorer-2:us-
east-1:123456789012:view/EC2-Only-View/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111
```

In this case, the person using the role must ask the administrator to update the role's permission policies to allow access to the view using the `resource-explorer-2:GetView` action.

### I want to allow people outside of my AWS account to access my Resource Explorer resources

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

To learn more, consult the following:

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

## Data protection in AWS Resource Explorer

The AWS shared responsibility model applies to data protection in AWS Resource Explorer. As described in this model, AWS is responsible for protecting the global infrastructure that runs all of the AWS Cloud. You are responsible for maintaining control over your content that is hosted on

this infrastructure. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

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

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

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

## Encryption at rest

Data that is stored by Resource Explorer includes the indexed list of the resources and their associated ARNs that are used by the customer and the views to access them.

This data is encrypted when at rest by using AWS Key Management Service (AWS KMS) symmetric encryption keys that implement the Advanced Encryption Standard (AES) in Galois Counter Mode (GCM) with 256-bit keys (AES-256-GCM).

## Encryption in transit

Customer requests and all associated data is encrypted in transit using Transport Later Security (TLS) 1.2 or later. All Resource Explorer endpoints support HTTPS for encrypting data in transit. For a list of Resource Explorer service endpoints, see AWS Resource Explorer endpoints and quotas in the *AWS General Reference*.

# Compliance validation for AWS Resource Explorer

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading reports in AWS Artifact in the *AWS Artifact User Guide*.

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

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

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

- AWS Compliance Resources – This collection of workbooks and guides might apply to your industry and location.
- Evaluating Resources with Rules in the *AWS Config Developer Guide* – AWS Config assesses how well your resource configurations comply with internal practices, industry guidelines, and regulations. In the Resource Explorer console, the details view of a selected resource shows its compliance with AWS Config Compliance Rules.
- AWS Security Hub – This AWS service provides a comprehensive view of your security state within AWS that helps you check your compliance with security industry standards and best practices. In

the Resource Explorer console, the details view of a selected resource shows findings from AWS Security Hub.

## Resilience in AWS Resource Explorer

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

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

## Infrastructure security in AWS Resource Explorer

As a managed service, AWS Resource Explorer is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well-Architected Framework*.

You use AWS published API calls to access Resource Explorer through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

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

For more information about AWS global network security procedures, see the Amazon Web Services: Overview of Security Processes whitepaper.

# Access AWS Resource Explorer using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and AWS Resource Explorer. You can access Resource Explorer as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Instances in your VPC don't need public IP addresses to access Resource Explorer.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for Resource Explorer.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

## Considerations for Resource Explorer

Before you set up an interface endpoint for Resource Explorer, review Considerations in the *AWS PrivateLink Guide*.

Resource Explorer supports making calls to all of its API actions through the interface endpoint.

## Create an interface endpoint for Resource Explorer

You can create an interface endpoint for Resource Explorer using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create an interface endpoint for Resource Explorer using the following service name:

```
aws.api.region.resource-explorer-2
```

```
aws.api.region.resource-explorer-2-fips
```

If you enable private DNS for the interface endpoint, you can make API requests to Resource Explorer using its default Regional DNS name. For example, `resource-explorer-2.us-east-1.amazonaws.com` and `resource-explorer-2.us-east-1.api.aws`.

# Create an endpoint policy for your interface endpoint

An endpoint policy is an IAM resource that you can attach to an interface endpoint. The default endpoint policy allows full access to Resource Explorer through the interface endpoint. To control the access allowed to Resource Explorer from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals that can perform actions (AWS accounts, IAM users, and IAM roles).
- The actions that can be performed.
- The resources on which the actions can be performed.

For more information, see Control access to services using endpoint policies in the *AWS PrivateLink Guide*.

**Example: VPC endpoint policy for Resource Explorer actions**

The following is an example of a custom endpoint policy. When you attach this policy to your interface endpoint, it grants access to the listed Resource Explorer actions for all principals on all resources.

```
{
  "Statement": [
    {
      "Principal": "*",
      "Effect": "Allow",
      "Action": [
        "resource-explorer-2:*"
      ],
      "Resource": "*"
    }
  ]
}
```

# Monitoring AWS Resource Explorer

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

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

# Logging AWS Resource Explorer API calls using AWS CloudTrail

AWS Resource Explorer is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Resource Explorer. CloudTrail captures all API calls for Resource Explorer as events. The calls captured include calls from the Resource Explorer console and code calls to the Resource Explorer API operations.

If you create a *trail*, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Resource Explorer. A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Using the information collected by CloudTrail, you can determine the request that was made to Resource Explorer, the IP address from which the request was made, who made the request, when it was made, and additional details.

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

## Resource Explorer information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Resource Explorer, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing events with CloudTrail Event history.

**Important**

You can find all Resource Explorer events by searching for **Event source** = **resource-explorer-2.amazonaws.com**

For an ongoing record of events in your AWS account, including events for Resource Explorer, create a trail. A *trail* enables CloudTrail to deliver log files to an Amazon S3 bucket. By default, when you create a trail in the console, the trail applies to all AWS Regions. The trail logs events from all Regions in the AWS partition and delivers the log files to the Amazon S3 bucket that you specify. Additionally, you can configure other AWS services to further analyze and act upon the event data collected in CloudTrail logs. For more information, see the following topics in the *AWS CloudTrail User Guide*:

- Creating a trail for your AWS account
- AWS service integrations with CloudTrail Logs
- Configuring Amazon SNS notifications for CloudTrail
- Receiving CloudTrail log files from multiple Regions
- Receiving CloudTrail log files from multiple accounts

All Resource Explorer actions are logged by CloudTrail and are documented in the AWS Resource Explorer API Reference. For example, calls to the `CreateIndex`, `DeleteIndex`, and `UpdateIndex` actions generate entries in the CloudTrail log files.

Every event or log entry contains information that helps you determine who made the request.

- AWS account root credentials
- Temporary security credentials from an AWS Identity and Access Management (IAM) role or federated user.
- Long-term security credentials from an IAM user.
- Another AWS service.

**Important**

For security reasons, all `Tags`, `Filters`, and `QueryString` values are redacted from the CloudTrail trail entries.

For more information, see the CloudTrail userIdentity element.

## Understanding Resource Explorer log file entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files aren't an ordered stack trace of the public API calls, so they don't appear in any specific order.

**Topics**

- CreateIndex
- DeleteIndex
- UpdateIndexType
- Search
- CreateView
- DeleteView
- DisassociateDefaultView

### CreateIndex

The following example shows a CloudTrail log entry that demonstrates the `CreateIndex` action.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-166EXAMPLE",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-166EXAMPLE",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
```

```
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T19:13:59Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-08-23T19:13:59Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "CreateIndex",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.create-index",
    "requestParameters": {
        "ClientToken": "792ee665-58af-423c-bfdb-d7c9aEXAMPLE"
    },
    "responseElements": {
        "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
        "State": "CREATING",
        "CreatedAt": "2022-08-23T19:13:59.775Z"
    },
    "requestID": "a193afe9-17ff-4f30-ae0a-73bb0EXAMPLE",
    "eventID": "2ec50598-4de6-474d-bd0e-f5c00EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


## DeleteIndex

The following example shows a CloudTrail long entry that demonstrates the `DeleteIndex` action.

**Note**

This action also asynchronously deletes all views for the account in that Region, which results in a `DeleteView` event for each deleted view.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:My-Role-Name",
        "arn": "arn:aws:sts::123456789012:assumed-role/My-Admin-Role/My-Delegated-
Role",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/My-Admin-Role",
                "accountId": "123456789012",
                "userName": "My-Admin-Role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T18:33:06Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-08-23T19:04:06Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "DeleteIndex",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.delete-index",
    "requestParameters": {
        "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
    },
    "responseElements": {
```

```
        "Access-Control-Expose-Headers": "x-amzn-RequestId,x-amzn-ErrorType,x-amzn-
ErrorMessage,Date",
        "State": "DELETING",
        "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
    },
    "requestID": "d7d80bd2-cd2d-47fb-88d6-5133aEXAMPLE",
    "eventID": "675eab39-c514-4d32-989d-0ea98EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

## UpdateIndexType

The following example shows a CloudTrail log entry that demonstrates the `UpdateIndexType` action to promote an index from type `LOCAL` to `AGGREGATOR`.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661282039",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661282039",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T19:13:59Z",
                "mfaAuthenticated": "false"
            }
        }
```

```
    },
    "eventTime": "2022-08-23T19:21:18Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "UpdateIndexType",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.update-index-type",
    "requestParameters": {
        "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
        "Type": "AGGREGATOR"
    },
    "responseElements": {
        "Type": "AGGREGATOR",
        "Arn": "arn:aws:resource-explorer-2:us-
east-1:123456789012:index/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111",
        "LastUpdatedAt": "2022-08-23T19:21:17.924Z",
        "State": "UPDATING"
    },
    "requestID": "a145309d-df14-4c2e-a9f6-8ed45EXAMPLE",
    "eventID": "ed33ab96-f5c6-4a77-a69a-8585aEXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


## Search

The following example shows a CloudTrail log entry that demonstrates the `Search` action.

> **Note**
>
> For security reasons, all references to `Tag`, `Filters`, and `QueryString` parameters are redacted in the CloudTrail trail entries.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
```

```
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661282039",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661282039",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T19:13:59Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-08-03T16:50:11Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "Search",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.search",
    "requestParameters": {
        "QueryString": "***"
    },
    "responseElements": null,
    "requestID": "22320db5-b194-446f-b9f4-e603bEXAMPLE",
    "eventID": "addb3bca-0c41-46bf-a5e6-42299EXAMPLE",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

## CreateView

The following example shows a CloudTrail log entry that demonstrates the `CreateView` action.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661282039",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661282039",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T19:13:59Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2023-01-20T21:54:48Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "CreateView",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.create-view",
    "requestParameters": {
        "ViewName": "CTTagsTest",
        "Tags": "***"
    },
    "responseElements": {
        "View": {
            "Filters": "***",
            "IncludedProperties": [],
            "LastUpdatedAt": "2023-01-20T21:54:48.079Z",
```

```
            "Owner": "123456789012",
            "Scope": "arn:aws:iam::123456789012:root",
            "ViewArn": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/
CTTest/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
        }
    },
    "requestID": "b22d8ced-4905-42c4-b1aa-ef713EXAMPLE",
    "eventID": "f62e339f-1070-41a8-a6ec-12491EXAMPLE",
    "readOnly": false,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

## DeleteView

The following example shows a CloudTrail log entry that demonstrates the event that can occur when the `DeleteView` action starts automatically because of a `DeleteIndex` operation in the same AWS Region.

> **Note**
>
> If the deleted view is the default view for the Region, this action asynchronously also disassociates the view as the default. This produces a `DisassociateDefaultView` event.

```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "AssumedRole",
        "principalId": "AROAEXAMPLEEXAMPLE:botocore-session-1661282039",
        "arn": "arn:aws:sts::123456789012:assumed-role/cli-role/botocore-
session-1661282039",
        "accountId": "123456789012",
        "accessKeyId": "AKIAIOSFODNN7EXAMPLE",
        "sessionContext": {
            "sessionIssuer": {
                "type": "Role",
                "principalId": "AROAEXAMPLEEXAMPLE",
                "arn": "arn:aws:iam::123456789012:role/cli-role",
                "accountId": "123456789012",
```

```
                "userName": "cli-role"
            },
            "webIdFederationData": {},
            "attributes": {
                "creationDate": "2022-08-23T19:13:59Z",
                "mfaAuthenticated": "false"
            }
        }
    },
    "eventTime": "2022-09-16T19:33:27Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "DeleteView",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.delete-view",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "cd174d1e-0a24-4b47-8b67-d024aEXAMPLE",
    "readOnly": false,
    "resources": [{
        "accountId": "334026708824",
        "type": "AWS::ResourceExplorer2::View",
        "ARN": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/
CTTest/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
    }],
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```


## DisassociateDefaultView

The following example shows a CloudTrail log entry that demonstrates the event that can occur when the `DisassociateDefaultView` action starts automatically because of a `DeleteView` operation on the current default view.


```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "accountId": "123456789012",
        "invokedBy": "resource-explorer-2.amazonaws.com"
```

```
    },
    "eventTime": "2022-09-16T19:33:26Z",
    "eventSource": "resource-explorer-2.amazonaws.com",
    "eventName": "DisassociateDefaultView",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "10.24.34.15",
    "userAgent": "aws-cli/2.7.14 Python/3.9.11 Windows/10 exe/AMD64 prompt/off command/
resource-explorer-2.disassociate-default-view",
    "requestParameters": null,
    "responseElements": null,
    "eventID": "d8016cb1-5c23-4ea4-bda2-70b03EXAMPLE",
    "readOnly": false,
    "eventType": "AwsServiceEvent",
    "managementEvent": true,
    "recipientAccountId": "123456789012",
    "eventCategory": "Management"
}
```

# Troubleshooting Resource Explorer

If you encounter issues when working with Resource Explorer, consult the topics in this section. Also see Troubleshooting AWS Resource Explorer permissions in the **Security** section of this guide.

**Topics**


- General issues (*this page*)
- Troubleshooting Resource Explorer setup and configuration issues
- Troubleshooting Resource Explorer search issues


# General issues

**Topics**


- I received a link to Resource Explorer but when I open it, the console shows only an error.
- Why does unified search in the console cause "access denied" errors in my CloudTrail logs?


## I received a link to Resource Explorer but when I open it, the console shows only an error.

Some third-party tools produce link URLs to pages in Resource Explorer. In some cases, those URLs don't include the parameter that directs the console to a specific AWS Region. If you open such a link, the Resource Explorer console isn't told which Region to use, and defaults to using the last Region the user signed in to. If the user doesn't have permissions to access Resource Explorer in that Region, then the console attempts to use US East (N. Virginia) (`us-east-1`) Region, or US West (Oregon) (`us-west-2`) if the console can't reach `us-east-1`.

If the user doesn't have permission to access the index in any of those Regions, then the Resource Explorer console returns an error.

You can prevent this issue by ensuring that all users have the following permissions:

- `ListIndexes` – no specific resource; use `*`.
- `GetIndex` for the ARN of the each index created in the account. To avoid having to redo permission policies if you delete and recreate an index, we recommend that you use `*`.

The minimum policy to achieve this might look like this example:

JSON

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "resource-explorer-2:GetIndex",
                "resource-explorer-2:ListIndexes"
            ],
            "Resource": "*"
        }
    ]
}
```

Alternatively, you might consider attaching the AWS managed permission `AWSResourceExplorerReadOnlyAccess` to all users who need to use Resource Explorer. That grants these required permissions, plus the permissions needed see the available views in the Region and search using those views.

## Why does unified search in the console cause "access denied" errors in my CloudTrail logs?

Unified search in the AWS Management Console lets principals search from any page in the AWS Management Console. The results can include resources from the principal's account if Resource Explorer is turned on and configured to support unified search. Whenever you start typing in the unified search bar, unified search attempts to call `resource-explorer-2:ListIndexes` operation to check whether it can include resources from the user's account in the results.

Unified search uses the currently signed-in user's permissions to perform this check. If that user doesn't have permission to call `resource-explorer-2:ListIndexes` granted in an attached AWS Identity and Access Management (IAM) permission policy, then the check fails. That failure is added as an `Access denied` entry in your CloudTrail logs.

This CloudTrail log entry has the following characteristics:

- **Event source:** `resource-explorer-2.amazonaws.com`
- **Event name:** `ListIndexes`
- **Error code:** `403` (Access denied)

The following AWS managed policies include permission to call `resource-explorer-2:ListIndexes`. If you assign any of these to the principal, or any other policy that includes this permission, then this error does not occur:

- AWSResourceExplorerReadOnlyAccess
- AWSResourceExplorerFullAccess
- ReadOnlyAccess
- ViewOnlyAccess

# Troubleshooting Resource Explorer setup and configuration issues

Use the information here to help you diagnose and fix issues that can occur when you initially set up or configure AWS Resource Explorer.

**Topics**


- I get an "access denied" message when I make a request to Resource Explorer
- I get an "access denied" message when I make a request with temporary security credentials


## I get an "access denied" message when I make a request to Resource Explorer

- Verify that you have permissions to call the action and resource that you requested. An administrator can grant permissions by assigning an AWS Identity and Access Management (IAM) permission policy to your IAM principal, such as a role, group, or user.

  To provide access, add permissions to your users, groups, or roles:

  - Users and groups in AWS IAM Identity Center:

Create a permission set. Follow the instructions in Create a permission set in the *AWS IAM Identity Center User Guide*.

- Users managed in IAM through an identity provider:

  Create a role for identity federation. Follow the instructions in Create a role for a third-party identity provider (federation) in the *IAM User Guide*.

- IAM users:
  - Create a role that your user can assume. Follow the instructions in Create a role for an IAM user in the *IAM User Guide*.
  - (Not recommended) Attach a policy directly to a user or add a user to a user group. Follow the instructions in Adding permissions to a user (console) in the *IAM User Guide*.

  The policy must allow the requested `Action` on the `Resource` that you want to access.

  If the policy statements that grant those permissions include any conditions, such as time-of-day or IP address restrictions, you also must meet those requirements when you send the request. For information about viewing or modifying policies for an IAM principal, see Managing IAM policies in the *IAM User Guide*.
- If you're signing API requests manually (without using the AWS SDKs), verify that you signed the request correctly.

## I get an "access denied" message when I make a request with temporary security credentials

- Verify that the IAM principal that you're using to make the request has the correct permissions. Permissions for temporary security credentials are derived from a principal defined in IAM, so the permissions are limited to those granted to the principal. For more information about how permissions for temporary security credentials are determined, see Controlling permissions for temporary security credentials in the *IAM User Guide*.
- Verify that your requests are being signed correctly and that the request is well formed. For details, see the toolkit documentation for your chosen SDK or Using temporary credentials with AWS resources in the *IAM User Guide*.
- Verify that your temporary security credentials haven't expired. For more information, see Requesting temporary security credentials in the *IAM User Guide*.

# Troubleshooting Resource Explorer search issues

Use the information here to help you diagnose and fix common errors that can occur when you search for resources by using Resource Explorer.

**Topics**


- Why are some resources missing from my Resource Explorer search results?
- Why are some searches limited to 1,000 results?
- Why are my resources not appearing in unified search results in the console?
- Why do unified search in the console and Resource Explorer sometimes give different results?
- What permissions do I need to be able to search for resources?


## Why are some resources missing from my Resource Explorer search results?

The following list provides reasons why some resources might not appear in your search results as expected:

**Initial indexing isn't complete**

After you initially turn on Resource Explorer in an AWS Region, it can take up to 36 hours for indexing and replication to the aggregator index to complete. Try your search again later.

**The resource is new**

It can take a few minutes for a new resource to be discovered by Resource Explorer and added to the local index. Try again in a few minutes.

**Information about a new resource in one Region hasn’t yet been propagated to the aggregator index**

It can take some time for details about a new resource discovered in one Region to be indexed in its own Region and then replicated to the aggregator index for the account. The new resource can appear in cross-Region search results only after replication is complete. Try your search again later.

**The Region with the resource doesn't have Resource Explorer turned on**

Your administrator determines which AWS Regions that Resource Explorer can operate in. The **Settings** page shows which Regions have Resource Explorer turned on and contain an index.

If the Region with your resource is not turned on, ask your administrator to turn on Resource Explorer in that Region.

**The resource exists in a different Region, and the searched Region doesn't contain the aggregator index**

You can search for resources across all Regions in the account only by using a view in the Region that contains the aggregator index. Searches in any other Region return resources from only the Region in which you perform the search.

**Filters on the view exclude that resource**

Every view can include filters in the configuration that restrict which results can be included in search results made with that view. Ensure that the resource you're looking for matches the filters in the view that you're using to search. For more about filters, see Filters.

**The resource type is not supported by Resource Explorer**

Some resource types aren't supported by Resource Explorer. For more information, see Resource types you can search for with Resource Explorer.

**Indexes or views aren't configured in the console Region**

If the indexes or views aren't configured in the Regions expected by the console consuming the widget, you will not see the results you expect. For more information, see Turning on cross-Region search by creating an aggregator index.

**Your views don't include tags**

Tags are required by the Resource Explorer widget. If your views don't include tags, the resources won't be included in your results. For more information, see Adding tags to views.

**Your search uses the wrong search query syntax**

Search in Resource Explorer is unique to this service. Without the correct syntax, you won't find the resources you expect. For more information, see Search query syntax reference for Resource Explorer.

**You have recently tagged your resources**

After you tag a resource, there may be a 30 second delay before the resource appears in your search results.

**The resource type doesn't support tag filters**

If tag filters aren't supported by the resource type, they won't display in the Resource Explorer widget. Resource types that don't support tag filters are:

- cloudfront:cache-policy
- cloudfront:origin-access-identity
- cloudfront:function
- cloudfront:origin-request-policy
- cloudfront:realtime-log-config
- cloudfront:response-headers-policy
- cloudwatch:dashboard
- docdb:globalcluster
- elasticache:globalreplicationgroup
- iam:group
- lambda:code-signing-config
- lambda:event-source-mapping
- ssm:windowtarget
- ssm:windowtask
- rds:auto-backup
- rds:global-cluster
- s3:accesspoint

## Why are some searches limited to 1,000 results?

If your query includes free-form text, the Resource Explorer console will use the `Search` operation, but if your query does not include free-form text, Resource Explorer uses the `ListResources` operation. `Search` operations are limited to 1,000 results that are sorted by relevancy, while `ListResource` operations have no upper limit and are not sorted by relevancy. To see resources beyond 1,000 results when using free-form text (and the `Search` operation), you must use additional filters.

## Why are my resources not appearing in unified search results in the console?

Unified search results are available in the search bar at the top of every AWS Management Console page. However, the search can return resources that match the query in search results *only* after the following configuration options are complete:

- There must be an aggregator index in one of the Regions in the account.
- There must be a default view in the Region that contains the aggregator index.
- All principals (IAM roles and users) must have permission to search using that default view.

## Why do unified search in the console and Resource Explorer sometimes give different results?

Unified search results are available in the search bar at the top of every AWS Management Console page. When you use unified search, the unified search process automatically inserts a wildcard character (*) to the end of the first term that you type in the query string. That wildcard character isn't visible in the unified search box, but it does affect the results.

> **Important**
>
> Unified search automatically inserts a wildcard character (*) operator at the end of the first keyword in the string. This means that unified search results include resources that match any string that starts with the specified keyword.
> The search performed by the **Query** text box on the Resource search page in the Resource Explorer console does ***not*** automatically append a wildcard character. You can insert a * manually after any term in the search string.

## What permissions do I need to be able to search for resources?

To search, you must have permission to perform *both* of the following operations on a view that resides in the Region in which you call the operation:

- `resource-explorer-2:GetView`
- `resource-explorer-2:Search`
- `resource-explorer-2:ListResources`

This can be done by adding a statement similar to the following example to a policy assigned to your IAM principal.

```
        {
            "Effect": "Allow",
```

```
            "Action": [
                "resource-explorer-2:GetView",
                "resource-explorer-2:Search"
            ],
            "Resource": "arn:aws:resource-explorer-2:us-east-1:123456789012:view/My-
View-Name/1a2b3c4d-5d6e-7f8a-9b0c-abcd11111111"
        }
```

You can replace the Amazon Resource Number (ARN) of a specific view with an ARN that includes a wildcard (*) to grant permission to all matching views.

If you don't specify a view in your request, Resource Explorer automatically uses the *default view* for the Region in which you made the request. If you don't have permissions to use the default view, talk to your administrator.

> **Note**
>
> Even if you see a resource in the results of a Resource Explorer search query, you need permissions on the resource itself to be able to interact with that resource.

# Quotas for Resource Explorer

Your AWS account has default quotas for each AWS service. Unless otherwise noted, quotas are Region-specific. You can request increases for some quotas, and other quotas cannot be increased.

To view the quotas for AWS Resource Explorer, open the Service Quotas console. In the navigation pane, choose **AWS services** and select **Resource Explorer**.

To request a quota increase, see Requesting a quota increase in the *Service Quotas User Guide*. If the quota is not yet available in Service Quotas, use the limit increase form.

The following quotas are the defaults and maximums for Resource Explorer.

| Maximum value quotas | Default value | Maximum value |
|---|---|---|
| Number of views in an AWS Region | 10 | 10 |

| Rate limits for operations | Default value | Maximum value |
|---|---|---|
| Maximum `Search` operations per second | 5 | 5 |
| Maximum non-`Search` operations per second | 3 | 3 |

# Using AWS Resource Explorer with an AWS SDK

AWS software development kits (SDKs) are available for many popular programming languages. Each SDK provides an API, code examples, and documentation that make it easier for developers to build applications in their preferred language.

| SDK documentation | Code examples |
| --- | --- |
| AWS SDK for C++ | AWS SDK for C++ code examples |
| AWS CLI | AWS CLI code examples |
| AWS SDK for Go | AWS SDK for Go code examples |
| AWS SDK for Java | AWS SDK for Java code examples |
| AWS SDK for JavaScript | AWS SDK for JavaScript code examples |
| AWS SDK for Kotlin | AWS SDK for Kotlin code examples |
| AWS SDK for .NET | AWS SDK for .NET code examples |
| AWS SDK for PHP | AWS SDK for PHP code examples |
| AWS Tools for PowerShell | AWS Tools for PowerShell code examples |
| AWS SDK for Python (Boto3) | AWS SDK for Python (Boto3) code examples |
| AWS SDK for Ruby | AWS SDK for Ruby code examples |
| AWS SDK for Rust | AWS SDK for Rust code examples |
| AWS SDK for SAP ABAP | AWS SDK for SAP ABAP code examples |
| AWS SDK for Swift | AWS SDK for Swift code examples |

**Example availability**

Can't find what you need? Request a code example by using the **Provide feedback** link at the bottom of this page.

# Document history for the Resource Explorer User Guide

The following table describes the documentation releases for AWS Resource Explorer. For notification about updates to this documentation, you can subscribe to an RSS feed.

| Change | Description | Date |
| --- | --- | --- |
| Updated managed policy | The `AWSResourceExplorerServiceRolePolicy` AWS managed policy has been updated to add policy permissions allowing Resource Explorer to manage indexes and views. | July 23, 2025 |
| Updated managed policy | Resource Explorer added support to view additional resource types. The `AWSResourceExplorerServiceRolePolicy` AWS managed policy has been updated to grant Resource Explorer access to view additional resource types. | May 7, 2025 |
| Added support for new resource types | Resource Explorer added support for 41 new resources from AWS services including AWS Device Farm and Amazon Managed Grafana. | May 7, 2025 |
| Added support for new resource types | Resource Explorer added support for 32 new resources from AWS services including AWS HealthOmics and Amazon Personalize. | March 21, 2025 |

| | | |
|---|---|---|
| Added support for new resource types | Resource Explorer added support for 34 new resources from AWS services including AWS IoT FleetWise and Amazon Lightsail. | January 28, 2025 |
| Added support for new resource types | Resource Explorer added support for 29 new resources from AWS services including Amazon EMR and Amazon MQ. | January 6, 2025 |
| Added support for new resource types | Resource Explorer added support for 59 new resources from AWS services including AWS DataSync, Amazon GuardDuty, and Amazon SageMaker AI. | November 21, 2024 |
| New console features | Resource Explorer added new console features including bulk resource actions like managing tags and creating an application, and added new resource details from other AWS services like AWS Config and AWS Security Hub. | November 20, 2024 |
| Managed views | Resource Explorer added managed views, allowing other AWS services to access resource information indexed by Resource Explorer for your AWS account or organization with your consent. | November 8, 2024 |

| | | |
|---|---|---|
| New search filter added | Resource Explorer added a new `tag:all` search query filter, enabling you to search for resources that have one or more user-created tags attached, even if the resource type is not supported in Resource Explorer. | September 6, 2024 |
| Content organization improvements | Updated topic titles and reorganized content to improve readability and discoverability. | August 29, 2024 |
| Notice to upgrade IAM policies to IPv6 | Customers who are using dual addressing with ASPEN policies containin g `aws:sourceIp` are impacted by this upgrade. Dual addressing means that the network supports both IPv4 and IPv6. | July 15, 2024 |
| Discontinued support for three resource types | Resource Explorer discontin ued support for the following three resource types: `ecs:task`, `ssm:autom ation-execution` , and `ssm:patchbaseline` . | July 9, 2024 |
| Added support for new resource types | Resource Explorer added support for 65 new resources from AWS services including AWS Key Management Service, Amazon Route 53, and Amazon Fraud Detector. | February 20, 2024 |

| | | |
|---|---|---|
| Updated managed policy | Resource Explorer added support to view additiona l resource types. The AWSResourceExplore rServiceRolePolicy _ AWS managed policy has been updated to grant Resource Explorer access to view additional resource types. | December 12, 2023 |
| New search filter added | Resource Explorer now supports searching your resources by application. | November 16, 2023 |
| Added support for new resource types | Resource Explorer added support for 86 new resources from AWS services including AWS CloudFormation, AWS Glue, and Amazon SageMaker AI. | November 15, 2023 |
| Resource Explorer supports multi-account search | You can now use Resource Explorer to search and discover resources across AWS accounts within your organization or organizational unit. For more information, see Turning on multi-account search. | November 14, 2023 |

| | | |
|---|---|---|
| New and updated managed policies | Resource Explorer added support for AWS Organizations. The AWS managed policies have been added and updated to grant Resource Explorer access to your organization, organizational structure, accounts, and delegated administrators. | November 14, 2023 |
| Added support for new resource types | Resource Explorer added support for AWS Organizations. The AWS managed policies have been updated to grant Resource Explorer access to your organization, organizational structure, accounts, and delegated administrators. | November 14, 2023 |
| Added support for new resource types | Resource Explorer now supports 12 new resource types from services including Amazon Cognito, AWS Elastic Beanstalk, and Amazon Elastic File System. | October 18, 2023 |
| Added support for new resource types | Resource Explorer added support for 164 resources. The AWS managed policies that grant Resource Explorer access to index resources were updated to include those new resource types. | October 17, 2023 |

| | | |
|---|---|---|
| Resource Explorer is now available in certain opt-in Regions | Customers in BAH and CGK can now opt in to Resource Explorer. | October 5, 2023 |
| Added support for new resource types | Resource Explorer added support for resources from the following AWS services: AWS CodeBuild, AWS CodePipeline, Amazon Cognito, Amazon Elastic Container Registry, AWS Elastic Beanstalk, Amazon Elastic File System, AWS IoT, and AWS Step Functions. The AWS managed policies that grant Resource Explorer access to index resources were updated to include those new resource types. | August 1, 2023 |
| Resource Explorer now supports exporting search results to a CSV | You can now export the results of your search on the **Resource search** page to a CSV-formatted file. | April 4, 2023 |
| Use Amazon Q Developer in chat applications to search and discover your AWS resources | You can now use Amazon Q Developer in chat applications to search your resources using natural language questions . For more information, see Using Amazon Q Developer in chat applications to search for resources. | March 30, 2023 |

| | | |
|---|---|---|
| Added support for new resource types | Resource Explorer added support for resources from the following AWS services: Amazon ElastiCache, AWS Lambda, and Amazon Simple Queue Service (Amazon SQS). The AWS managed policies that grant Resource Explorer access to index resources were updated to include those new resource types. | March 7, 2023 |
| IAM best practices update | Updated guide to align with the IAM best practices . For more information, see Security best practices in IAM. | December 6, 2022 |
| New AWS managed policies | Resource Explorer adds AWSResourceExplorerFullAccess, AWSResourceExplorerReadOnlyAccess, and AWSResourceExplorerServiceRolePolicy managed policies. | November 7, 2022 |
| Initial release | Initial release of the Resource Explorer User Guide | November 7, 2022 |