aws

User Guide

AWS Resource Explorer

AWS Resource Explorer: User Guide

Table of Contents

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.

AWS Resource Explorer provides fast responses to your search queries by using indexes that are created and maintained by the AWS Resource Explorer service. These indexes come in two types: Resource Explorer-owned indexes that provide immediate partial results, and user-owned indexes that provide complete results with automatic updates. 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.

Beginning October 6, 2025, Resource Explorer is enabled by default in your AWS account with no setup steps required to begin searching for and finding resources. When you first access Resource Explorer through the AWS Management Console, Unified Search, or CLI/SDK/API, you'll automatically receive search functionality based on your IAM permissions, as explained later in this page under the section called “First-time user”. Resource Explorer provides immediate search results in each AWS Region you access, with Unified Search showing results from your current AWS Region without requiring any additional configuration.

For enhanced functionality, you can search across multiple AWS Regions by enabling cross-Region search with a single click or API call to select an aggregator AWS Region. Organizations can set up multi-account search by enabling trusted access and using AWS CloudFormation deployment. If you previously configured Resource Explorer with cross-Region search, your existing setup remains unchanged. You can now search for resources across hundreds of AWS services, such as Amazon Elastic Compute Cloud instances, Amazon Kinesis streams, Amazon DynamoDB tables, and more, without any initial configuration, making it easier to manage resources and troubleshoot issues from the moment you access your account.

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?

As a first-time user of Resource Explorer, you can start searching for resources immediately without any setup steps. You can access Resource Explorer through the AWS Management Console, Unified Search, or CLI/SDK/API to begin finding your resources.

Your search experience is automatically enabled based on your IAM permissions. If you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, you can immediately search in partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release). Resource Explorer uses a service-linked channel to receive AWS CloudTrail events on your behalf, and this visibility is available in the CloudTrail console across all supported Regions. For complete resource inventory with automatic updates, you'll also need the `iam:CreateServiceLinkedRole` permission (included in the `AWSResourceExplorerFullAccess` managed policy). After the service-linked role is created in your account by any user, subsequent users only need the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to get complete results on first search in subsequent Regions.

To get the most from Resource Explorer, we recommend starting with search and then exploring these topics as needed:

- Using AWS Resource Explorer to search for resources
- Terms and concepts for Resource Explorer
- Setting up and configuring Resource Explorer

Features of Resource Explorer

Resource Explorer provides the following features:

- Automatic features (available immediately):
  - Resource Explorer is automatically enabled when you first access the service or search in Unified Search, providing immediate search functionality. Depending on your IAM permissions, you can view either partial results immediately or complete resource inventory with automatic updates.
  - Users can search for resources in their current AWS Region through the Resource Explorer console, Unified Search, or CLI/SDK/API.
  - 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.
- Enhanced features (optional configuration):
  - Cross-Region search capability when you select an aggregator AWS Region.
  - Multi-account search through AWS Organizations integration.
  - Additional resource details from other AWS services, such as AWS Config and AWS Cloud Control, directly in the Resource Explorer console.
  - Resource management using quick Actions in the Resource Explorer console to manage tags and add resources to new or existing applications.
  - Custom views that administrators can create to define which resources are available in search results, with different views for different user groups based on their needs.

  Manual configuration is only needed in specific cases:

  - When you need to add missing permissions

- When you want enhanced features like cross-Region search
- When you want to set up multi-account configurations
- When you want to customize your Resource Explorer setup

- 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 AWS Region, and replicating that to the AWS Region that contains the aggregator index for the account.

Resource Explorer supported Regions

The following table provides information about the AWS Regions where Resource Explorer is available.

Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	resource-explorer-2.us-east-2.amazonaws.com resource-explorer-2-fips.us-east-2.amazonaws.com resource-explorer-2-fips.us-east-2.api.aws	HTTPS HTTPS HTTPS
US East (N. Virginia)	us-east-1	resource-explorer-2.us-east-1.amazonaws.com resource-explorer-2-fips.us-east-1.amazonaws.com resource-explorer-2-fips.us-east-1.api.aws	HTTPS HTTPS HTTPS
US West (N. California)	us-west-1	resource-explorer-2.us-west-1.amazonaws.com resource-explorer-2-fips.us-west-1.amazonaws.com	HTTPS HTTPS HTTPS

Region Name	Region	Endpoint	Protocol
		resource-explorer-2-fips.us-west-1.api.aws	
US West (Oregon)	us-west-2	resource-explorer-2.us-west-2.amazonaws.com resource-explorer-2-fips.us-west-2.amazonaws.com resource-explorer-2-fips.us-west-2.api.aws	HTTPS HTTPS HTTPS
Africa (Cape Town)	af-south-1	resource-explorer-2.af-south-1.amazonaws.com	HTTPS
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

Region Name	Region	Endpoint	Protocol
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 (New Zealand)	ap-southeast-6	resource-explorer-2.ap-southeast-6.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
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 (Taipei)	ap-east-2	resource-explorer-2.ap-east-2.amazonaws.com	HTTPS

Region Name	Region	Endpoint	Protocol
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 resource-explorer-2-fips.ca-central-1.amazonaws.com resource-explorer-2-fips.ca-central-1.api.aws	HTTPS HTTPS HTTPS
Canada West (Calgary)	ca-west-1	resource-explorer-2.ca-west-1.amazonaws.com resource-explorer-2-fips.ca-west-1.amazonaws.com resource-explorer-2-fips.ca-west-1.api.aws	HTTPS HTTPS HTTPS
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

Region Name	Region	Endpoint	Protocol
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
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

Automatic setup and basic search functionality are available at no additional cost. There are no charges to search for resources by using AWS Resource Explorer, including creating views, completing setup in AWS 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 how Resource Explorer is automatically enabled based on your permissions and how to access enhanced features for your AWS account.

Accessing Resource Explorer

Resource Explorer is automatically enabled when you access the service through any of the following methods, with your experience determined by your IAM permissions:

- **Full Experience:** Users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) get complete search results with automatic infrastructure creation on a Regional basis for indexes and views.
- **Enhanced Experience:** Users with only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy get immediate access to partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release).
- **No Access:** Users without the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy receive access denied errors, respecting IAM boundaries.

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. Simply navigating to or accessing the Resource Explorer console automatically triggers the setup process based on your permissions. You can access the Resource Explorer console by signing into the AWS Management Console and searching for it by name or choosing **View all services** and selecting **Resource Explorer** from the list.

You can also navigate in your browser directly to the **Resource Explorer home 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. Resource Explorer returns resource search results in Unified Search when you have the appropriate permissions. You can use Resource Explorer search query syntax in the Unified Search text box, and see matching resources in those search results. You can search for resources from the console of any AWS service without having to first switch to the Resource Explorer console.

> **Important**
>
> Unified Search uses the default view in the aggregator Region if configured, or the default view in the current Region for regional results.

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 and automatically trigger setup based on your permissions when you invoke search 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 that automatically enable Resource Explorer functionality when you have appropriate permissions. 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.

Getting started immediately

You can start using Resource Explorer immediately with just the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. This provides instant access to search functionality with partial results while automatic setup completes in the background.

> **Note**
>
> Automatic setup can complete only if you have the necessary permissions. You must have, at minimum, the permissions in the `AWSResourceExplorerFullAccess` managed policy.

To start searching immediately

1. Ensure you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. This permission is included in the `ResourceExplorerFullAccess` managed policy.
2. Navigate to the Resource Explorer console or use Unified Search from any AWS console page.
3. Begin searching immediately. You'll receive partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release) while indexing completes.
4. (Optional) For complete results, obtain the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). Once any user in your

account creates the service-linked role, all users with search permission searching in a new Region can create an index and a view for full results.

Enhanced setup for cross-Region search

While Resource Explorer provides immediate regional search functionality, you can optionally configure enhanced features like cross-Region search and custom views.

- **Cross-Region search:** Create an aggregator index to search across all Regions from a single location.
- **Custom views:** Create filtered views for specific resource types or access control requirements.
- **Multi-account search:** Configure organization-wide resource discovery (requires management account or delegated administrator permissions).

For detailed setup instructions, see Setting up and configuring Resource Explorer.

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 is available immediately when you have the appropriate permissions. Users with the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy can start searching for resources right away without any setup. Users with both the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) get complete search results with automatic infrastructure creation (index and view) on first search in a Region. The `iam:CreateServiceLinkedRole` permission is needed only by one user initially to create the service-linked role for the account. After the service-linked role exists in the account, all users with search permission searching in a new Region can create an index and a view for full results.

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.

Resource Explorer operates in two modes: automatic setup and manual setup. With automatic setup, Resource Explorer creates the necessary infrastructure (indexes and views) when you first search in a Region, provided you have the required permissions. Manual setup allows administrators to pre-configure Resource Explorer infrastructure before users begin searching.

You should understand the following concepts to successfully use AWS Resource Explorer .

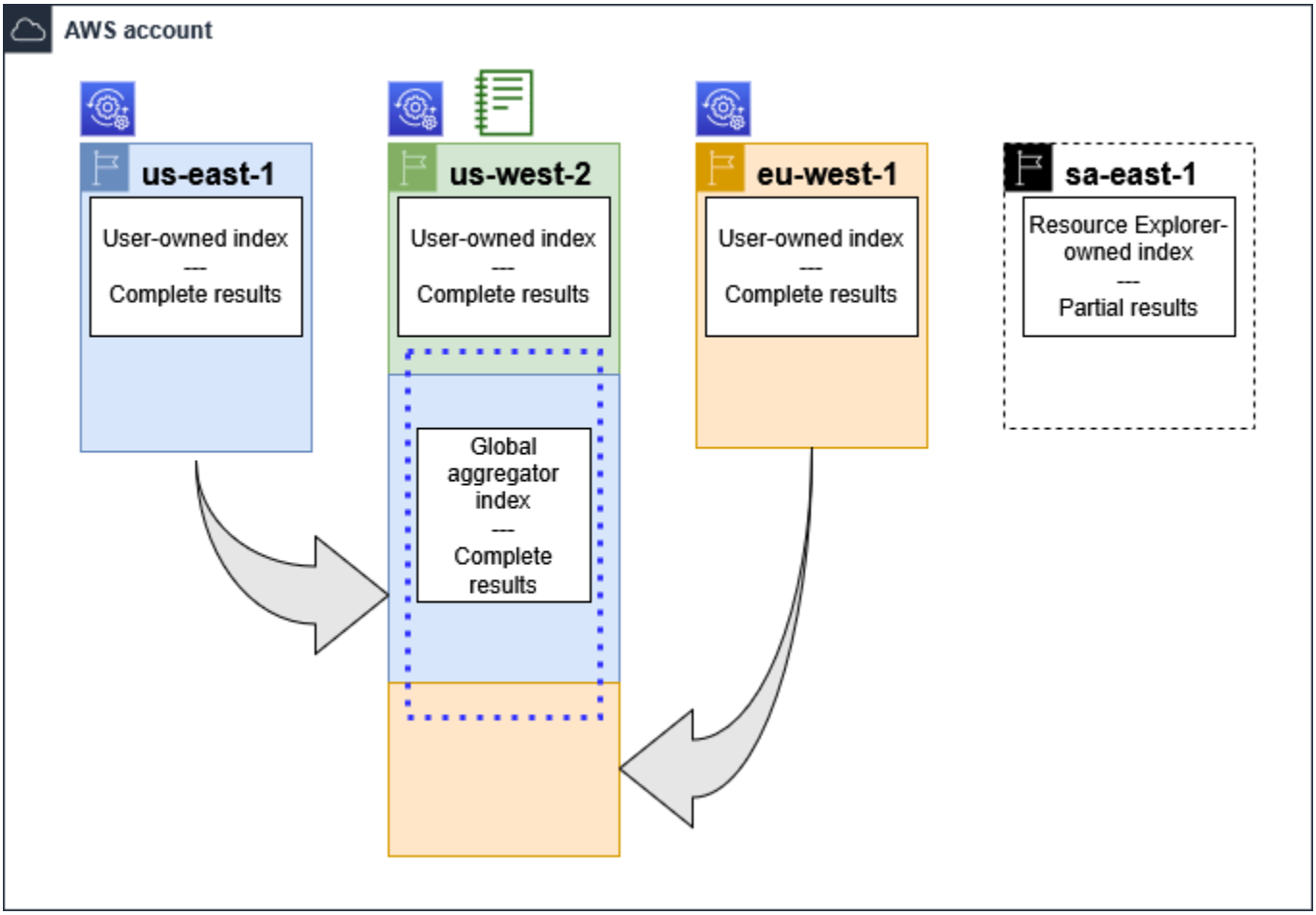
Concepts

- 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 users have searched for resources, and one Region where no search has occurred yet. Regions with user-owned (local) indexes provide complete search results, while Regions with only Resource Explorer owned indexes provide partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release).

In this example scenario, a user selected the US West (Oregon) Region (`us-west-2`) to contain the aggregator index for the account. All Regions with user-owned (local) indexes 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 including Tags.



Legend	
	Resource Explorer is set up with a user-owned (local) index in this AWS Region. Information about the Region's resources is stored in a local index in that Region. Every Region's user-owned (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 user-owned (local) indexes of all other Regions into the aggregator index in this Region. Searches made in this Region can include results from all Regions with user-owned (local) indexes in the account.
	The default view created by **Quick Setup** includes all resources in all AWS Regions with user-owned (local) indexes.

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. With Resource Explorer functionality available in an account by default, manual administrator setup is optional for basic functionality. Users with appropriate permissions can start searching immediately and Resource Explorer will automatically create the necessary infrastructure. The Resource Explorer administrator can configure the following features:

- Complete setup for individual AWS Regions in the AWS account by creating user-owned indexes in those Regions by searching or in **Settings**. This provides complete search results and lets Resource Explorer discover all resources and populate the index with comprehensive information about those resources.
- Enable cross-Region search by updating 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 user-owned indexes exist.
- 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

Resource Explorer provides three permission-based experience tiers for users:

Full Experience

Permissions: At minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. If the service-linked role doesn't exist in the account, one user needs the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) to create it initially

Experience: Complete single-Region resource search results with automatic updates

Enhancement: Can optionally enable cross-Region search by selecting an aggregator index

Enhanced Experience

Permissions: At minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy

Experience: Partial results immediately (all tagged resources and supported untagged resources created after the immediate resource discovery release)

Enhancement: Can upgrade to full experience by obtaining service-linked role creation permission or having another user with permissions create the service-linked role in the account

No Access

Permissions: Missing the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy

Experience: No resource search access

Enhancement: Must obtain proper permissions to access the service

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 get 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": "*"
    }
    ```

- 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 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 three types of indexes:

Resource Explorer-owned index

A *Resource Explorer owned index* exists in every AWS Region and is managed by the Resource Explorer service. These indexes cannot be deleted or modified by users. Resource Explorer owned indexes provide partial search results, including all tagged resources and supported untagged resources created after the immediate resource discovery release. Users with only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy access resources through these indexes.

User-owned (local) index

There is one *user-owned (local) index* in every AWS Region in which you complete setup for Resource Explorer. A user-owned index contains complete information about all resources in the same Region and provides full search results.

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 user-owned indexes exist 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. Using an aggregator index is optional but recommended for cross-region search capabilities. There can be ***only one*** aggregator index in an AWS account.

When you complete setup for Resource Explorer, you can specify which AWS Region contains 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 Enabling cross-Region search by creating an aggregator index.

After the service-linked role has been created in the account (created by a user with the `iam:CreateServiceLinkedRole` permission, which is included in the AWSResourceExplorerFullAccess managed policy), automatic index creation occurs when users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy perform their first search in a Region that doesn't have a user-index set up already. If the service-linked role doesn't exist in the account, the user needs the `iam:CreateServiceLinkedRole` permission to create it. After the service-linked role exists in the account, any user with, at

minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy can trigger automatic index creation for complete search results.

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, enable or disable 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.

For more information about views in Resource Explorer, see *Views*.

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 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.

Unified Search automatically uses the default view in the AWS Region that contains the aggregator index for the account or the default or service view per Region. This lets you search for a resource from any page in the AWS Management Console, without having to first open Resource Explorer.

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

AWS Resource Explorer is available immediately when you have the appropriate permissions. Users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy can start searching for resources right away without any setup. Users with the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and `iam:CreateServiceLinkedRole` permissions (included in the AWSResourceExplorerFullAccess managed policy) get complete search results with automatic infrastructure creation on first search.

> **Note**
>
> After the service-linked role is created in your account when any user with the `iam:CreateServiceLinkedRole` permission accesses Resource Explorer, subsequent users need only, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to create an index and view for full results in a Region on first search.

Your search experience is automatically enabled based on your IAM permissions. For enhanced functionality like cross-Region search, multi-account configurations, or more control over your Resource Explorer configuration, you can use the manual setup options below.

Quick Setup and Advanced Setup options remain available for customers who want cross-Region search or more control over their Resource Explorer configuration.

Note

Multi-account search requires that your account is part of an AWS Organizations organization.

There are two ways to enhance your Resource Explorer configuration:

- **Enable Cross-Region Search**
- **Enhanced Configuration Options**

Important

If you choose to create user-owned indexes using any option that says "all AWS Regions", it creates indexes only in those AWS Regions that exist and that are enabled in the AWS account *at the time you perform the procedure*. User-owned indexes are ***not*** automatically created in any AWS Regions that AWS adds in the future. When AWS introduces a new Region, you can choose to create user-owned indexes in the Region manually when it appears in the **Settings** page of the Resource Explorer console, or by calling the CreateIndex operation.

Note

Configuring Resource Explorer can enhance the ability to search for resources using the Unified Search bar on the AWS Management Console. Unified Search works with local Region indexes and does not require an aggregator index. For cross-Region search capabilities, you can optionally configure an aggregator index and default view. For more information, see Using Unified Search in the AWS Management Console.

Enabling cross-Region search

To enable cross-Region search capabilities, you can complete setup to create user-owned indexes and configure an aggregator index. This procedure does the following:

- Creates user-owned indexes in every AWS Region in your AWS account for complete search results.
- 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 enable cross-Region search

1. Open the AWS Resource Explorer console at https://console.aws.amazon.com/resource-explorer.
2. If you see the **Complete setup and enable cross-Region search** banner, proceed to the next step. Otherwise, navigate to **Settings** to access setup options. You can also access **Complete Setup** from the left navigation when available.
3. In the **Complete setup and enable cross-Region search** banner, select your preferred aggregator index from the list. Choose the Region that is appropriate for the geographic location of your users.
4. Choose **Enable cross-Region search in all Regions**. Alternatively, you can choose **Customize Region setup** for more granular control over which Regions to include.
5. Monitor the indexing progress.
6. Wait for the setup to complete. The indexing process creates user-owned indexes in all or selected Regions and configures the aggregator index in your selected Region.

After setup completes, you and your users can search for resources across all Regions. The cross-Region search capability will be fully available after indexing is 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.

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 Using enhanced configuration options to see what commands are the equivalent of using the console.

Using enhanced configuration options

For more granular control over your Resource Explorer configuration, you can use Advanced setup options to:

- Choose the AWS Regions in which to create user-owned indexes for complete search results.
- 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 Enabling 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 where you have user-owned indexes. 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.

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 configure Resource Explorer with enhanced options

1. Open the AWS Resource Explorer console at https://console.aws.amazon.com/resource-explorer.
2. Navigate to **Settings** to access enhanced configuration options, or choose **Customize Region setup** from the cross-Region setup banner. You can also access **Complete Setup** from the left navigation when available.
3. Select the specific Regions where you want to create user-owned indexes, or configure custom view settings as needed.
4. If enabling cross-Region search, review the "Confirm cross-Region setup" modal that explains: "By enabling cross-Region search, AWS performs the following steps:" followed by details about creating indexes in all AWS Regions, creating the aggregator index, and creating default view with filter.
5. Choose **Cancel** to return to the previous screen, or **Confirm and enable** to proceed with the cross-Region setup.
6. Monitor the setup progress and wait for indexing to complete. To continue using Resource Explorer with partial results during this process, choose **Proceed to Resource Search**.

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: Create user-owned 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 enables 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.

Understanding the immediate resource discovery experience

Immediate resource discovery experience

Beginning October 6, 2025, Resource Explorer provides immediate resource discovery functionality without requiring manual setup. When you first access Resource Explorer through the console, Unified Search, CLI, or API, the service automatically enables search capabilities based on your IAM permissions. Setup occurs when you use the Search or ListResources APIs from Resource Explorer either directly or through Unified Search. This eliminates the traditional setup barrier and provides immediate value while maintaining all existing functionality for customers who have already configured Resource Explorer.

The automatic experience provides different levels of functionality based on your permissions:

- **Immediate search access:** If you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, you can immediately search all tagged resources and supported untagged resources created after the immediate resource discovery release.
- **Complete resource inventory:** For complete resource inventory with automatic updates, you'll also need the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). Once the service-linked role is created in your account by any user, subsequent users need only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to create an index and view for full results on first search in each Region .
- **Enhanced functionality:** Cross-Region search capabilities remain available as optional enhancements that can be enabled with a single click.

This approach differs from the previous manual setup approach where users had to explicitly configure indexes and views before they could search for resources. Now, basic search functionality is available immediately, with enhanced features accessible through progressive permission-based upgrades.

Permission tiers and user experiences

Resource Explorer provides three distinct user experiences based on your IAM permissions:

Tier 1: Full search experience

Required permissions: At minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (needed only for initial service-linked role creation per account).

Experience: Complete resource search results with automatic infrastructure creation. On first search, Resource Explorer automatically creates the service-linked role and user-owned indexes and views in the Region, providing full search functionality including all tagged and supported untagged resources with ongoing automatic updates. On search in subsequent Regions, Resource Explorer automatically creates user-owned indexes and views per Region. After the service-linked role is created in your account by any user, subsequent users need only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to create an index and view for full results on first search in each Region.

Available through managed policies: `AdministratorAccess`, `AWSResourceExplorerFullAccess`, or custom policies with both permissions.

Tier 2: Partial search experience

Required permissions: At minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy only (missing `iam:CreateServiceLinkedRole` and service-linked role does not already exist in account)

Experience: If no service-linked role exists in your account, you get immediate partial search results from Resource Explorer-owned indexes. Results include all tagged resources and supported untagged resources created after the immediate resource discovery release, but not complete historical data. If the service-linked role already exists in your account (created by another user), you can get full results in each Region you search with just the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy.

Available through managed policies: `ReadOnlyAccess`, `AWSResourceExplorerReadOnlyAccess`, or custom policies with search and list indexes permissions only.

Upgrade path: To get complete results when no service-linked role exists, obtain `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) from your administrator or sign in with a role that has this permission.

Tier 3: No search access

Permissions: No `AWSResourceExplorerReadOnlyAccess` managed policy permissions

Experience: Access denied errors when attempting to search. This tier respects IAM boundaries and provides complete access control.

Upgrade path: Obtain, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy from your administrator to access basic search functionality.

Troubleshooting permission issues

If you encounter permission-related issues:

- **Access denied errors:** You need, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. Contact your administrator to obtain the necessary permissions or sign in with a role that has this permission.
- **Partial results only:** You have search permission but lack `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). You can continue with partial results or request additional permissions for complete functionality.
- **Service-linked role creation errors:** If you see errors about creating service-linked roles, you need `iam:CreateServiceLinkedRole` permission or must sign in with a role that has this permission.

Understanding indexing progress and completion

When Resource Explorer automatically creates infrastructure for complete search functionality, indexing happens in the background. Understanding the progress indicators helps you know what to expect.

Timeline expectations

- **Immediate partial results:** Available instantly when you first access Resource Explorer with appropriate permissions.

- **Complete resource inventory:** Full indexing typically completes within minutes to hours, depending on the number of resources in your account and Regions.
- **Ongoing updates:** Once complete indexing finishes, resource updates are reflected in search results within minutes.

Completion criteria and indicators

You can identify indexing progress and completion through several indicators:

- **Console banners:** Blue banners indicate "indexing in progress" while green banners show "setup completed successfully."
- **Search result completeness:** Partial results include all tagged resources and supported untagged resources created after the immediate resource discovery release. Complete results include your full supported resource inventory with historical data and automatic updates.
- **Index status:** You can check index status on the Settings page or using the `GetIndex` API operation. Active status indicates completed indexing.

Complete indexing means Resource Explorer has discovered and indexed all supported resource types in your account, providing comprehensive search results with ongoing automatic updates.

Identifying which Regions display full versus partial resource results

AWS Resource Explorer provides different levels of search results depending on your permissions and the type of index in each Region. All Regions have Resource Explorer-owned indexes that provide partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release. Accessing partial results requires that you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy). Regions with user-owned indexes provide complete search results. You can identify which type of results you'll receive in each Region by checking the index types using the procedures on this page.

The search results you receive depend on both your permissions and the index type:

- **Full results:** Available in Regions with user-owned indexes when you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy (and the service-linked role exists in the account)
- **Partial results:** Available in all Regions through Resource Explorer-owned indexes when you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, or in Regions where user-owned indexes have been deleted

> **Important**
>
> Users can search for resources in all Regions, but the completeness of results varies by index type. Regions with user-owned indexes provide complete search results and replicate to the aggregator index for cross-Region search (when enabled). Regions with only Resource Explorer-owned indexes provide partial results (all tagged resources and supported untagged resources created after the immediate resource discovery release) but are not included in cross-Region search replication.

Checking index types and result completeness in Regions

You can check which Regions have user-owned indexes (providing full results) versus Resource Explorer-owned indexes (providing partial results) 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. All Regions have Resource Explorer-owned indexes by default, while user-owned indexes are created when users with appropriate permissions complete setup in a Region.

AWS Management Console

To check index types and result completeness by Region

1. Open the **Settings** page in the Resource Explorer console.
2. The list in the **Indexes** section shows user-owned indexes that provide complete search results. 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. Regions not listed have only Resource Explorer-owned indexes, which provide partial results.
3. To see which Regions have only Resource Explorer-owned indexes (partial results), choose **Create indexes**. Available Regions in this list have only Resource Explorer-owned indexes and will provide partial search results until you create user-owned indexes for full results.

AWS CLI

To check index types and result completeness by Region

Run the following command to see which AWS Regions have user-owned indexes (providing complete results). Regions not listed have only Resource Explorer-owned indexes (providing partial results).

```
$ 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"
        }
    ]
```

```
}
```

Creating user-owned indexes for enhanced Resource Explorer functionality

AWS Resource Explorer automatically enables basic search functionality when you search with appropriate permissions. However, you may need to manually complete setup in specific scenarios, such as when you lack required permissions, have previously deleted an index in a Region, or need to manage existing aggregator configurations. For enhanced functionality like cross-Region search, you can use the **Quick setup** option or one-click cross-Region banner to create indexes in all AWS Regions that are turned on in your AWS account. When you use the Quick Setup option, Resource Explorer promotes the specified Region to be the aggregator index for the account. If you use the **Advanced setup** option, you can specify the Regions in which to create indexes.

Topics

When you complete setup for Resource Explorer in an AWS Region, the service performs the following actions:

- When the first user with appropriate permissions accesses Resource Explorer in the first Region in an AWS account, Resource Explorer automatically 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. Resource Explorer uses a service-linked channel to receive CloudTrail events on your behalf. 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 automatically creates an index in the specified Region to store the details about that Region's resources. Once the service-linked role exists in the account, subsequent Regions are automatically enabled when users with search permissions invoke search operations in those Regions.
- 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

While Resource Explorer automatically enables basic search functionality, you may need to manually create indexes in specific scenarios. The Resource Explorer console provides banner notifications to guide you through setup completion, and you can access enhanced setup options through the "Complete Setup" option in the left navigation or on the **Settings** page.

Manual index creation is typically needed when:

- You lack the required `iam:CreateServiceLinkedRole` permission for automatic setup
- You previously deleted an index in a Region and want to restore full functionality
- You need to manage existing aggregator configurations or create cross-Region search capabilities
- You want enhanced control over index configuration and tagging

During manual setup, you may see indexing progress indicators in the console. A blue banner displays "Completing AWS Resource Explorer setup" while indexing is in progress, which changes to a green completion banner when setup is finished.

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. **Primary method - Banner workflow:** If you see a **Complete setup and enable cross-Region search** banner, you can use these guided workflows:

   - For cross-Region setup, choose **Enable cross-Region search in all Regions** in the banner

     -or-

     For customized setup, choose **Customize Region setup** to access enhanced configuration options.

2. **Alternative method - Settings page:** If banner workflows are not available or you need manual control:

   a. On the Resource Explorer **Settings** page.

   b. In the **Indexes** section, choose **Create indexes**.

   c. 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 user-owned index.

   d. (Optional) In the **Tags** section, you can specify tag key and value pairs to the index.

   e. 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 can also create a view for users in that Region and grant your users permissions to that view or users can search using the service view in that Region.. For instructions on how to create a view, see Configuring a 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 complete setup for 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 can also create a view for users in that Region and grant your users permissions to that view or users can search using the service view in that Region. For instructions on how to create a view, see Configuring a 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 Completing setup for 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.

Enabling 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

AWS Resource Explorer stores the information it collects about the resources in an AWS Region to a *user-owned (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 user-owned (local) index in the US West (Oregon) Region.

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

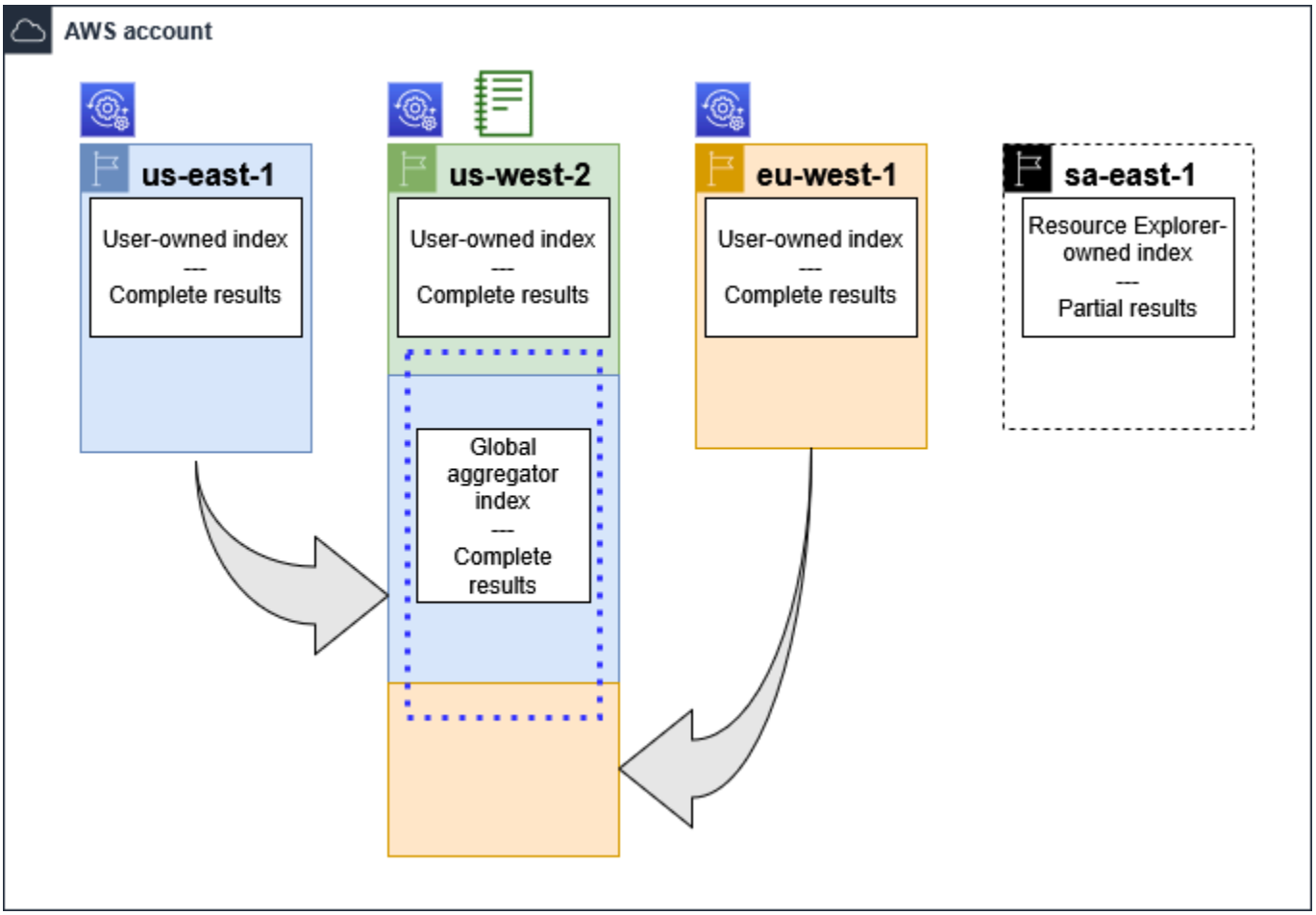
Resource Explorer automatically creates user-owned indexes when users with appropriate permissions perform search operations in a Region. This automatic creation occurs when users have both, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission, or when the service-linked role already exists in the account and users have search permissions.

The aggregator index contains a replicated copy of the user-owned (local) index in every other Region where you have a user-owned index for 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:

- Completes setup of Resource Explorer in three AWS Regions (`us-east-1`, `us-west-2`, and `eu-west-1`) by searching in or directly creating indexes in those Regions. Each Region contains its own user-owned (local) index.
- Does not create a user-owned index in the `sa-east-1` Region. Resources from `sa-east-1` will not appear in cross-region search results from other Regions.
- Creates the aggregator index for the account in the `us-west-2` Region. This causes Resource Explorer to replicate information from the user-owned (local) indexes in all other Regions where Resource Explorer where a user-owned (local) index exists. This allows searches performed in `us-west-2` to include resources from all three Regions in which Resource Explorer setup is complete.

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 set up with a user-owned (local) index in this AWS Region. Information about the Region's resources is stored in a local index in that Region. Every Region's user-owned (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 user-owned (local) indexes of all other Regions into the aggregator index in this Region. Searches made in this Region can include results from all Regions with user-owned (local) indexes in the account.
	The default view created by **Quick Setup** includes all resources in all AWS Regions with user-owned (local) indexes.

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

For cross-region search functionality, Resource Explorer provides a streamlined banner workflow that allows you to enable cross-region search with a single click. If you see a **Complete setup and enable cross-Region search** banner, you can choose **Enable cross-Region search in all Regions** for automatic setup or **Customize Region setup** for granular control. Alternatively, you have the option to manually create an aggregator index in one AWS Region by promoting an existing user-owned (local) index from the **Settings** page . This procedure describes the manual process for promoting a user-owned (local) index to be the aggregator index for the account.

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 user-owned (local) index to be the aggregator index for the account

1. **Primary method - Banner workflow:** If you see a **Complete setup and enable cross-Region search** banner or banner notification, you can use the guided workflow:
   a. For automatic setup across all regions, choose **Enable cross-Region search in all Regions**, or
   b. For customized configuration, choose **Customize Region setup** to access granular setup options.
2. **Alternative method:** The **Settings** page.
   a. Open the Resource Explorer **Settings** page.
   b. In the **Indexes** section, select the check box next to the index that you want to promote, and then choose **Change index type**.
   c. In the **Change index type for <*Region name*>** dialog, choose **aggregator index**, and then choose **Save changes**.

AWS CLI

To promote a user-owned (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. For information about the banner workflow option, see the Console tab above.

```
$ 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

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 have at minimum read-only access.

The account resources you can view in your Unified Search results depend on the permissions assigned to you.

- Partial Regional results: With, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, you can can immediately search all tagged resources and supported untagged resources created after the immediate resource discovery release in a Region.
- Full Regional results: With at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission, you can search full results, including all tagged and untagged resources with ongoing automatic updates and historical backfill, in a Region.
- Full cross-Region results: If you create an aggregator index, cross-Region results are available in Unified Search.

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

For more information about resourcer views, see the section called “Permission tiers and user experiences”.

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 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 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 CloudFormation StackSets to configure AWS Resource Explorer for organization-wide search 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 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 CloudFormation and Organizations. This grants CloudFormation permission to create the roles needed in the organization's management account and the member accounts 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. 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 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 Resource Explorer 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"
                }
            }
        }
    }
}
```

Working with views

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.

Granting access to a view

Resource Explorer provides three types of views: User views, AWS managed views, and AWS service views.

User views

User views are created and managed by users or administrators. When automatic setup occurs, Resource Explorer creates user-owned default views that include tags for comprehensive filtering capabilities.

AWS service views

A service view is a pre-defined view owned and managed by AWS services (not customer accounts) in AWS Resource Explorer that enables controlled access to resource data.

> **Note**
>
> A Resource Explorer-owned view is a type of service view that acts as a fallback when no user-owned default view exists in a Region. These views cannot be modified or deleted by users and provide basic search functionality through Resource Explorer-owned indexes.

AWS managed views

A managed view provides other AWS services with the ability to access resource information indexed by Resource Explorer for your AWS account or organization with your consent.

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 a 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.

Comparison of view types

The following table compares the different types of views available in Resource Explorer:

Feature	User Views	Managed Views	Service Views
Created by	User	AWS Service (per account)	AWS Service (global)
Can user modify	Yes	No	No
Can user delete	Yes	Via service only	No
Requires user setup	Yes	Service manages	No
Access to config/relationship data	No	Yes	Yes

Feature	User Views	Managed Views	Service Views
Streaming support	No	No	Yes

User views

User views are created and managed by users or administrators. When automatic setup occurs, Resource Explorer creates user-owned default views that include tags for comprehensive filtering capabilities.

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.

Configuring a 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 with user-owned (local) indexes 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 with user-owned (local) indexes.

Default views

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

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.

Topics

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.

> **Resource Explorer-owned views**
>
> Resource Explorer provides Resource Explorer-owned views that are service-managed and cannot be modified or deleted by users. These views serve as automatic fallbacks to ensure search functionality remains available even when no user-owned views exist in a Region. Resource Explorer-owned views do not include resource tags in search results.
> When automatic setup occurs, Resource Explorer creates a default user-owned view in each Region that includes Tags. The view hierarchy follows this priority: user-owned views are used first, with automatic fallback to Resource Explorer-owned views when no user-owned view is available.

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 user-owned index in every Region in the account.

Multi-account views allow you to search for resources in accounts across your organization. Only the management account, or a delegated administrator for the organization, can create a multi-account view. 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. 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. Note that with Resource Explorer's automatic setup, view creation is optional since Resource Explorer-owned views provide fallback search functionality:

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

  Resource: This can be `*` to allow creation of a view in any AWS Region in the account.
- **When view creation is needed:** While Resource Explorer automatically creates default views during setup and provides Resource Explorer-owned views as fallbacks, you may want to create custom views for specific filtering requirements, tag-based searches, or to control access permissions for different user groups.

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": [
                "resource-explorer-2:Search",
                "resource-explorer-2:GetView"
            ],
            "Resource": "arn:aws:resource-explorer-2:us-
east-1:123456789012:view/MyTestView/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
        }
    ]
}
```

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 automatically creates default user-owned views during setup that include Tags for enhanced search functionality. If no user-owned view exists in a Region, Resource Explorer automatically falls back to using a Resource Explorer-owned view to ensure search functionality remains available. Resource Explorer-owned views are service-managed and cannot be modified or deleted, and they do not include resource tags in search results.

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 uses the default view in the aggregator Region if an aggregator index exists, or the default view in the current Region if no aggregator is configured. This provides regional search results without requiring cross-region setup.

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** or set as default in the **Default** column.

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.

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/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ]
    }
    ]
    }
```

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": [
    "resource-explorer-2:Search",
    "resource-explorer-2:GetView"
    ],
    "Resource": [
    "arn:aws:resource-explorer-2:us-east-1:123456789012:view/policy-
name/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    ]
    }
    ]
    }
```

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"
}
```

What are AWS service views

AWS service views are pre-defined views (cannot be modified or deleted) that enable controlled resource data access by AWS service teams. AWS Resource Explorer is a platform service that other service teams can take a dependency on to provide value-added services to customers, while providing access and usage transparency to customers. Service views represent one of three integration patterns (user views, AWS managed views, and AWS service views) that other AWS services can use to integrate with Resource Explorer.

There are two types of service views:

- **Resource Explorer-defined service views:** Used by Resource Explorer itself to power features like the default search functionality.
- **Service-defined service views:** Created by an AWS service during onboarding to access specific resource data. Customers cannot use this view directly to view resources.

Key characteristics of service views

AWS service views have the following key characteristics:

Service-defined service view

Service views are created during AWS service onboarding and cannot be modified or deleted by customers.

Pre-defined configuration

Service views include specific filters and properties defined during service onboarding to meet the service's integration requirements.

Global availability

Service views are automatically available to authorized callers without setup as a global resource.

How service views work

Service views support two primary use cases:

- **Search and discovery**:
  - **Resource Explorer-defined service views**: Customers can use this view to discover resources in the default search functionality.
  - **Service-defined service views**: Services can search for customer resources with customer credentials.
- **Resource streaming:** Services receive real-time resource change notifications through event streams

Customers can manage service views through the following actions:

Customer opt-in is required for streaming access through service views. Customers must explicitly grant permission through the Resource Explorer `CreateStreamingAccessForService` API action. AWS services must create their own service views and can only use the service views they have created.

Customer experience

Customers can manage service views through the following actions:

Viewing available service views

How can customers see what service views exist?

Customers can view all available service views by using the `ListServiceViews` API action. This API action returns a list of all service views that are available in their account, including both Resource Explorer-defined and service-defined views. The response includes the view name, ARN, and configuration details.

Monitoring service access

How can customers see which services currently have access?

Customers can monitor which services have streaming access to their resources by using the `ListStreamingAccessForServices` API action. This action provides a complete list of all services that are currently authorized to receive resource updates, allowing customers to maintain visibility over their resource data sharing.

Permissions and security

Service views maintain strong security controls:

- **Customer control:** Customers retain control over which services can access their resources (resource streaming only)
  - **Service-linked role-based access limitations:** When AWS services use SLRs with Resource Explorer permissions, customers must accept the predefined permissions or choose not to use the service
  - **Customer options:** To revoke Search*/List* access granted through SLRs, customers must disable the entire service integration
- **IAM integration:** Works with existing IAM policies and Resource Explorer permissions
- **Service principal allowlisting:** Only pre-approved AWS services can create and use service views

Related topics

- Configuring a Resource Explorer view to provide access to resource searches
- Identity and access management for AWS Resource Explorer
- Terms and concepts for Resource Explorer

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 be updated or deleted only 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:

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: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": "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": "arn:aws:resource-explorer-2:us-east-1:111122223333:managed-view/
ExampleManagedViewName/EXAMPLE8-90ab-cdef-fedc-EXAMPLE11111"
  }
  ]
  }
```

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 provides details about the specified managed view, similar to the following:

```
{
  "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\":\"YYYY-MM-DD\",\"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*.

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. Resource discovery happens automatically when you access Resource Explorer with appropriate permissions. If you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, you can immediately search all tagged resources and supported untagged resources created after the immediate resource discovery release. For complete resource discovery with automatic updates, you'll also need the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). After the service-linked role is created in your account by any user, subsequent users need only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to get complete results.

Topics

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 February 10, 2026, Resource Explorer no longer supports the following resource types:
>
> - **Amazon Chime**— `chime:media-pipeline`

As of February 10, 2026, Resource Explorer no longer supports the following resource types:

- **Amazon Location Service**— `geo:map`
- **Amazon Location Service**— `geo:place-index`
- **Amazon Location Service**— `geo:tracker`

As of December 5, 2025, Resource Explorer no longer supports the following resource types:

- **AWS IoT Analytics**— `iotanalytics:channel`
- **AWS IoT Analytics**— `iotanalytics:dataset`
- **AWS IoT Analytics**— `iotanalytics:datastore`
- **AWS IoT Analytics**— `iotanalytics:pipeline`

As of November 21, 2025, Resource Explorer no longer supports the following resource types:

- **AWS IoT FleetWise**— `iotfleethub:application`

As of November 3, 2025, Resource Explorer no longer supports the following resource types:

- **Amazon Lookout for Metrics**— `lookoutmetrics:Alert`
- **Amazon Lookout for Metrics**— `lookoutmetrics:AnomalyDetector`
- **Amazon Lookout for Vision**— `lookoutvision:project`

As of October 13, 2025, Resource Explorer no longer supports the following resource types:

- **Amazon CloudWatch Evidently**— `evidently:project`
- **Amazon CloudWatch Evidently**— `evidently:project/experiment`
- **Amazon CloudWatch Evidently**— `evidently:project/feature`
- **Amazon CloudWatch Evidently**— `evidently:project/launch`

As of October 1, 2025, Resource Explorer no longer supports the following resource types:

- **Amazon Quantum Ledger Database (Amazon QLDB)** — `qldb:ledger`
- **Amazon Quantum Ledger Database (Amazon QLDB)** — `qldb:stream`

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`

Supported services and resource types

Supported AWS services

- Amazon API Gateway
- Direct Connect
- AWS Partner Network
- AWS Shield
- AWS Systems Manager Incident Manager
- AWS WAFV2
- Amazon Macie
- OpenSearch Service Serverless Service
- Amazon S3 Express
- Amazon VPC Lattice
- Amazon Verified Permissions
- Amazon WorkSpaces Web
- 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 Backup gateway
- AWS Batch
- Amazon Bedrock
- AWS Certificate Manager
- Amazon Chime
- AWS Cloud Map
- AWS Cloud9
- CloudFormation
- Amazon CloudFront
- AWS CloudTrail
- Amazon CloudWatch
- 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
- AWS Config
- Amazon Connect
- Amazon Connect Customer Profiles
- 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
- Amazon EC2 Auto Scaling
- 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 MediaStore
- 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 Inspector
- Amazon Interactive Video Service
- AWS IoT
- 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
- AWS License Manager
- 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 Quick
- 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 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:apis
- apigateway:apis/integrations
- apigateway:apis/routes
- apigateway:apis/stages
- apigateway:restapis
- apigateway:restapis/deployments
- apigateway:restapis/resources
- apigateway:restapis/resources/methods
- apigateway:restapis/stages
- apigateway:vpclinks

Direct Connect

- directconnect:dx-gateway

AWS Partner Network

- partnercentral:catalog/engagement
- partnercentral:catalog/engagement-invitation
- partnercentral:catalog/opportunity
- partnercentral:catalog/resource-snapshot-job
- partnercentral:resourcesnapshot

AWS Shield

- shield:protection

- shield:protection-group

AWS Systems Manager Incident Manager

- ssm-incidents:response-plan

AWS WAFV2

- wafv2:ipset
- wafv2:regexpatternset
- wafv2:rulegroup
- wafv2:webacl

Amazon Macie

- macie2:allow-list
- macie2:custom-data-identifier
- macie2:findings-filter

OpenSearch Service Serverless Service

- aoss:collection

Amazon S3 Express

- s3express:bucket

Amazon VPC Lattice

- vpc-lattice:service
- vpc-lattice:service/listener
- vpc-lattice:servicenetwork
- vpc-lattice:servicenetworkserviceassociation

- vpc-lattice:targetgroup

Amazon Verified Permissions

- verifiedpermissions:policy-store

Amazon WorkSpaces Web

- workspaces-web:portal

AWS Amplify

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

AWS App Runner

- apprunner:autoscalingconfiguration
- apprunner:connection
- apprunner:service
- apprunner:vpcconnector

AWS AppConfig

- appconfig:application
- appconfig:application/environment
- appconfig:deploymentstrategy
- appconfig:extensionassociation

Amazon AppFlow

- appflow:flow

AppIntegrations

- app-integrations:application
- app-integrations:event-integration

AWS App Mesh

- appmesh:mesh
- appmesh:mesh/virtualGateway
- appmesh:mesh/virtualGateway/gatewayRoute
- appmesh:mesh/virtualNode
- appmesh:mesh/virtualRouter
- appmesh:mesh/virtualRouter/route
- 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/routingcontrol
- 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 Backup gateway

- backup-gateway:hypervisor

AWS Batch

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

Amazon Bedrock

- bedrock:agent
- bedrock:agent-alias
- bedrock:application-inference-profile
- bedrock:data-automation-project
- bedrock:flow
- bedrock:guardrail
- bedrock:knowledge-base
- bedrock:prompt
- bedrock:prompt-router

AWS Certificate Manager

- acm:certificate

Amazon Chime

- chime:app-instance
- chime:app-instance/bot
- chime:app-instance/user
- chime:media-insights-pipeline-configuration
- chime:media-pipeline-kinesis-video-stream-pool
- chime:sma
- chime:vc

AWS Cloud Map

- servicediscovery:service

AWS Cloud9

- cloud9:environment

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:dashboard
- cloudtrail:eventdatastore
- cloudtrail:trail

Amazon CloudWatch

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

Amazon CloudWatch Logs

- logs:destination
- logs:log-group

Amazon CloudWatch Observability Access Manager

- oam:sink

Amazon CloudWatch RUM

- rum:appmonitor

Amazon CloudWatch Synthetics

- synthetics:canary
- 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
- comprehend:flywheel

AWS Config

- config:config-rule

Amazon Connect

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

Amazon Connect Customer Profiles

- profile:domains
- profile:domains/integrations
- profile:domains/object-types

Amazon Connect Wisdom

- wisdom:assistant

- wisdom:association
- wisdom:content
- 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:cert
- dms:endpoint
- dms:es
- dms:rep
- dms:subgrp
- dms:task

Amazon Detective

- detective:graph

AWS Device Farm

- devicefarm:instanceprofile
- devicefarm:project
- devicefarm:testgrid-project

Amazon DynamoDB

- dynamodb:table

DynamoDB Accelerator

- dax:cache

Amazon EC2 Auto Scaling

- autoscaling:autoScalingGroup

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:jobtemplates
- emr-containers:securityconfigurations
- emr-containers:virtualclusters
- emr-containers:virtualclusters/endpoints

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
- eks:daemonset
- eks:deployment
- eks:eks-anywhere-subscription
- eks:endpointslice
- eks:ingress
- eks:namespace
- eks:persistentvolume
- eks:podidentityassociation
- eks:replicaset
- eks:service
- eks:statefulset

Elastic Load Balancing

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

AWS Elemental MediaPackage

- mediapackage:channels
- mediapackage:origin_endpoints

AWS Elemental MediaPackage VoD

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

AWS Elemental MediaStore

- mediastore:container

AWS Elemental MediaTailor

- mediatailor:channel
- mediatailor:liveSource
- mediatailor:playbackConfiguration
- mediatailor:vodSource

Amazon CloudWatch Events

- events:api-destination
- 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:backup
- 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
- gamelift:script

AWS Global Accelerator

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

AWS Glue

- glue:crawler
- glue:dataQualityRuleset
- glue:database

- glue:job
- glue:mlTransform
- glue:registry
- 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
- guardduty:detector/filter
- guardduty:detector/ipset
- guardduty:detector/publishingDestination
- guardduty:detector/threatintelset
- guardduty:malware-protection-plan

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
- inspector2:filter

Amazon Inspector

- inspector:target/template
- inspector2:filter

Amazon Interactive Video Service

- ivs:channel
- ivs:encoder-configuration
- ivs:ingest-configuration
- ivs:playback-key
- ivs:playback-restriction-policy
- ivs:recording-configuration
- ivs:storage-configuration
- ivs:stream-key

AWS IoT

- iot:authorizer
- iot:billinggroup
- iot:cacert
- iot:cert
- iot:fleetmetric
- iot:jobtemplate
- iot:mitigationaction
- iot:policy

- iot:provisioningtemplate
- iot:rolealias
- iot:rule
- iot:ruledestination
- iot:scheduledaudit
- iot:securityprofile
- iot:thing
- iot:thinggroup
- iot:thingtype

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:access-policy
- iotsitewise:asset
- iotsitewise:asset-model
- iotsitewise:dashboard
- iotsitewise:gateway
- iotsitewise:portal
- iotsitewise:project

AWS IoT TwinMaker

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

AWS IoT Wireless

- iotwireless:Destination
- iotwireless:DeviceProfile
- iotwireless:FuotaTask
- iotwireless:MulticastGroup
- iotwireless:ServiceProfile

- iotwireless:SidewalkAccount
- iotwireless:WirelessDevice
- iotwireless:WirelessGateway
- iotwireless:WirelessGatewayTaskDefinition

Amazon Kendra

- kendra:index
- kendra:index/access-control-configuration
- kendra:index/data-source
- kendra:index/experience
- kendra:index/faq
- kendra:index/featured-results-set
- kendra:index/query-suggestions-block-list
- kendra:index/thesaurus

AWS Key Management Service

- kms:key

Amazon Kinesis

- kinesis:stream

Amazon Managed Service for Apache Flink

- kinesisanalytics:application

Amazon Kinesis Video Streams

- kinesisvideo:channel
- kinesisvideo:stream

AWS Lambda

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

Amazon Lex

- lex:bot
- lex:bot-alias

AWS License Manager

- license-manager:grant

Amazon MQ

- mq:broker
- mq:configuration

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:snapshot
- 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
- mobiletargeting:templates/EMAIL
- mobiletargeting:templates/PUSH
- mobiletargeting:templates/SMS

AWS Network Firewall

- network-firewall:firewall
- network-firewall:firewall-policy
- network-firewall:stateful-rulegroup
- network-firewall:stateless-rulegroup

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
- proton:environment-template
- proton:service-template

Amazon Quick

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

Amazon Redshift

- redshift:cluster
- redshift:eventsubscription
- redshift:hsmclientcertificate
- 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:app
- 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:cell
- 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 Glacier

- glacier:vaults

Amazon SageMaker AI

- sagemaker:action
- sagemaker:algorithm
- sagemaker:app
- sagemaker:app-image-config
- sagemaker:artifact
- sagemaker:cluster
- sagemaker:code-repository
- sagemaker:context
- sagemaker:domain
- sagemaker:endpoint
- sagemaker:endpoint-config
- sagemaker:experiment
- sagemaker:experiment-trial
- sagemaker:experiment-trial-component
- sagemaker:feature-group

- sagemaker:flow-definition
- sagemaker:hub
- sagemaker:human-task-ui
- sagemaker:image
- sagemaker:image-version
- sagemaker:inference-component
- sagemaker:inference-experiment
- sagemaker:mlflow-tracking-server
- sagemaker:model
- sagemaker:model-card
- sagemaker:model-package
- sagemaker:model-package-group
- sagemaker:monitoring-schedule
- sagemaker:notebook-instance
- sagemaker:notebook-instance-lifecycle-config
- sagemaker:pipeline
- sagemaker:project
- sagemaker:space
- sagemaker:studio-lifecycle-config
- sagemaker:user-profile
- sagemaker:workforce
- sagemaker:workteam

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:dedicated-ip-pool
- 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:multiregionaccesspoint
- s3:storage-lens
- s3:storage-lens-group

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:session
- ssm:windowtarget
- ssm:windowtask

AWS Transfer Family

- transfer:agreement
- transfer:certificate
- transfer:connector
- transfer:profile
- transfer:server
- transfer:user
- 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	ec2:security-group-rule
ec2:securitygroupingress	

Actual resource type	Reported as resource type
elasticloadbalancingv2:loadbalancer	elasticloadbalancing:loadbalancer
docdb:dbcluster neptune:dbcluster rds:dbcluster	rds:cluster
docdb:dbclusterparametergroup neptune:dbclusterparametergroup rds:dbclusterparametergroup	rds:cluster-pg
docdb:clustersnapshot neptune:dbclustersnapshot rds:clustersnapshot	rds:cluster-snapshot
docdb:dbinstance neptune:dbinstance rds:dbinstance	rds:db
docdb:eventsubscription neptune:eventsubscription rds:eventsubscription	rds:es
docdb:globalcluster rds:globalcluster	rds:global-cluster
neptune:dbparametergroup rds:dbparametergroup	rds:pg

Actual resource type	Reported as resource type

`docdb:dbsubnetgroup`
 	
`neptune:dbsubnetgroup`
 	
`rds:dbsubnetgroup`
	`rds:subgrp`

Using AWS Resource Explorer to search for resources

AWS Resource Explorer provides immediate search capabilities for your AWS resources without requiring manual setup. When you access Resource Explorer with the appropriate permissions, the service automatically enables resource search functionality in your account. You can use the AWS Management Console or the AWS Command Line Interface (AWS CLI) to search for resources using Resource Explorer.

The search experience you receive depends on your IAM permissions. With basic (Read-Only) search permissions, you get immediate access to partial results. With additional permissions, you receive complete resource inventory and enhanced functionality.

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

- **Resource Explorer automatically enables search functionality based on your permissions.**

  When you access Resource Explorer with the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, the service automatically provides search capabilities. Users with both the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) receive complete resource inventory. The `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) is only needed until the first user creates the service-linked role for the account. Once created, users with only the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy will also receive complete results in subsequent Regions where they search. Users without the service-linked role receive partial results immediately (all tagged resources plus untagged resources created after the feature launch).

- **Every search uses 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.

Resource Explorer provides both user-owned views (which you create and manage) and Resource Explorer-owned views (which are service-managed and cannot be modified or deleted). Resource Explorer-owned views do not include resource tags in search results. If no user-owned view is available, Resource Explorer automatically falls back to using a Resource Explorer-owned view to ensure search functionality remains available.

For more information about views, see Configuring a 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 completed setup for Resource Explorer. In some cases, modifications or deletions may take up to two weeks to be visible.

- **Search results may be partial during initial indexing.**

  During the initial indexing process after automatic setup, you may see partial results while the complete resource inventory is being built. Users with appropriate permissions will see indexing progress indicators in the console, and results will become more complete as indexing progresses.

- **A search in Resource Explorer occurs within an AWS Region.**

  Each Region where you complete setup for 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.

Search query string

A `QueryString` is a set of free-form text keywords that Resource Explorer implicitly joins using `OR` operator logic. You seperate 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
```

The search results for this example are based on the following behavior:

- Resource Explorer sorts results by relevance and gives higher preference to resources that match a greater number of the search terms. Resources that match more of the terms are pushed higher in the search results.
- Resources that do not match one or more of the terms aren't excluded from the results, but Resource Explorer considers them of lower relevance. Resources that don't match the terms are pushed 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***.

If you use more than one filter in the search query, Resource Explorer joins the filters using the `AND` operator. If a filter contains multiple values, Resource Explorer uses the `OR` operator between each value.

For example, consider the following query that consists of two free-form keywords and two specified filters:

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

Resource Explorer evaluates the query as follows:

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

The query string `test instance` follows the logical `OR` syntax of a keyword query string while the specified `service:` and `region:` filters follow the implicit `AND` syntax. This 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` operator, 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:`.

Specifying multiple filter values

Resource Explorer allows you to specify multiple values for each filter, which Resource Explorer joins with the `OR` operator. You can use this operator to scope down results to multiple resource types, accounts, tags, and more with a simple syntax. When specifying multiple values in the same filter, separate each value with a comma.

For example, consider the following query that consists of two filters, where one filter includes multiple possible values separated by a comma:

```
service:EC2 region:us-west-2,us-east-1
```

Given the comma-separated syntax of the `region:` filter, Resource Explorer evaluates the query as follows:

```
(service:EC2) AND (region:us-west-2 OR region:us-east-1)
```

Resource Explorer applies the `OR` operator to the comma-separated `region:` filter values.

This example query's results include any resources that are associated with Amazon EC2 and are in the US West (Oregon) or US East (N. Virginia) AWS Region.

> **Note**
>
> You can escape or quote the commas used to specify the OR operator like other special characters. For example, `tag.key:comma\,literal` or `tag.key:"comma,literal"`.

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. `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.
	
	`application:MyApplicationName`
	
	`application:arn:aws:resource-groups: us-east-1 :123456789012:group/MyApplicationName/123456789abced`
	
	`arn:aws:resource-groups: us-east-1 :123456789012:group/MyApplicationName/123456789abced`
	
	Note
	To use this filter, your view must have access to tagging data.

`id:`	The identifier of an individual resource, expressed as an Amazon resource name (ARN).

Filter name	Description and example
	`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. `region:us-east-1` **Note** 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:gl obal`	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. `region:global` **Note** 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. `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. `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. `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. `tag:environment=production` 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. 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. **Note** 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. 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. **Note** 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. `tag.key:environment` 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. `tag.value:production` 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"`
 	
or
 	
`"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.
	
	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.
	
	`"This matches only resources with the whole sentence."`
	
	`This matches resources with any of the words.`
	
	`"us-east-1"` – matches only resources that associated with that exact Region.
	
	`us-east-1` – matches any resource that contain "us" or "east" or "1".
	
	`-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. `region:us-east*` **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.
value,*value*	OR operator. You can place a comma (,) between multiple values in a filter. The following example causes Resource Explorer to return search results that match any of the values in the region: filter. `service:EC2 region:us-west-2,us-east-1`

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.
	
	`-service:ec2`
	
	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 string, 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 string works.
	
	`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"`).
	
	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.
	
	`"my\-key\-word"`
	
	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.
	
	`"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 `PROD`, `prod`, `Prod`, 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 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

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 CSPM** — 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 CSPM 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 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 CloudFormation. Tagging 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 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 *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 CloudFormation stacks in your resource selection when adding resources to an application in the Resource Explorer console. Adding a 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 *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 when you have appropriate Resource Explorer permissions.

With Unified Search, users can search for resources from ***any*** AWS service console without having to first navigate to the AWS Resource Explorer console. Unified Search returns regional results from the current Region by default, or cross-region results if an aggregator index is configured.

Access to resource results in Unified Search is permission-based. Users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy can immediately search for resources in any Region where they have access, with results varying based on their permission level and the type of indexes available in the Region.

> **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

> **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 resource search is enabled

To see if resource search is enabled in your AWS account, verify that the following requirements for Resource Explorer are met:

- Users must have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. Resource Explorer automatically provides search functionality, with results varying based on permission level and index type (user-owned indexes provide complete results, Resource Explorer-owned indexes provide partial results).
- (Optional) For cross-region search results, you can create an aggregator index in a Region of your choice. Without an aggregator, Unified Search returns regional results from the current Region.

Enabling Unified Search

Resource search in Unified Search is automatically available when users have appropriate Resource Explorer permissions. To enhance Unified Search functionality with full results, cross-Region results, or custom views, you can optionally complete the following steps:

1. (Optional) ??? to create user-owned indexes for complete search results.
2. (Optional) ??? to enable cross-Region search results.
3. (Optional) ??? for specific filtering requirements or access control.

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 CloudFormation to deploy Resource Explorer to AWS Organizations

You can use 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 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 CloudFormation Designer to help you get started with CloudFormation templates. For more information, see What is 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 Creating user-owned indexes for enhanced Resource Explorer functionality and Enabling 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 a Resource Explorer view to provide access to resource searches.

  > **Note**
  >
  > You must create a user-owned index in a Region before you can create a view in that same Region. If you create a user-owned 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 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 access 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 your account has streaming access enabled for an AWS service, the streaming access must be removed before you can turn off Resource Explorer. To remove streaming access, follow the

offboarding workflow for each service that has streaming access enabled. You can view which services have streaming access by using the ListStreamingAccessForServices API operation.

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.

If your account has streaming access enabled for an AWS service, the streaming access must be removed before you can turn off Resource Explorer. To remove streaming access, follow the offboarding workflow for each service that has streaming access enabled. You can view which services have streaming access by using the ListStreamingAccessForServices API operation.

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

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.

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

How you use AWS Identity and Access Management (IAM) differs based on your role:

- **Service user** - request permissions from your administrator if you cannot access features (see Troubleshooting AWS Resource Explorer permissions)
- **Service administrator** - determine user access and submit permission requests (see How Resource Explorer works with IAM)
- **IAM administrator** - write policies to manage access (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 as the AWS account root user, an IAM user, or by assuming an IAM role.

You can sign in as a federated identity using credentials from an identity source like AWS IAM Identity Center (IAM Identity Center), single sign-on authentication, or Google/Facebook credentials. For more information about signing in, see How to sign in to your AWS account in the *AWS Sign-In User Guide*.

For programmatic access, AWS provides an SDK and CLI to cryptographically sign requests. For more information, see AWS Signature Version 4 for API requests in the *IAM User Guide*.

AWS account root user

When you create an AWS account, you begin with one sign-in identity called the AWS account *root user* that has complete access to all AWS services and resources. We strongly recommend that you don't use the root user for everyday tasks. For tasks that require root user credentials, see Tasks that require root user credentials in the *IAM User Guide*.

Users and groups

An *IAM user* is an identity with specific permissions for a single person or application. We recommend using temporary credentials instead of IAM users with long-term credentials. For more information, see Require human users to use federation with an identity provider to access AWS using temporary credentials in the *IAM User Guide*.

An *IAM group* specifies a collection of IAM users and makes permissions easier to manage for large sets of users. For more information, see Use cases for IAM users in the *IAM User Guide*.

Roles

An *IAM role* is an identity with specific permissions that provides temporary credentials. You can assume a role by switching from a user to an IAM role (console) or by calling an AWS CLI or AWS API operation. For more information, see Methods to assume a role in the *IAM User Guide*.

IAM roles are useful for federated user access, temporary IAM user permissions, cross-account access, cross-service access, and applications running on Amazon EC2. For more information, see Cross account resource access in IAM 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 defines permissions when associated with an identity or resource. AWS evaluates these policies when a principal makes a request. Most policies are stored in AWS as JSON documents. For more information about JSON policy documents, see Overview of JSON policies in the *IAM User Guide*.

Using policies, administrators specify who has access to what by defining which **principal** can perform **actions** on what **resources**, and under what **conditions**.

By default, users and roles have no permissions. An IAM administrator creates IAM policies and adds them to roles, which users can then assume. IAM policies define permissions regardless of the method used to perform the operation.

Identity-based policies

Identity-based policies are JSON permissions policy documents that you attach to an identity (user, group, or role). These policies control what actions identities 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 *inline policies* (embedded directly into a single identity) or *managed policies* (standalone policies attached to multiple identities). To learn how to choose between managed and inline policies, 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 include 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. You must specify a principal in a resource-based policy.

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 policy types that can set the maximum permissions granted by more common policy types:

- **Permissions boundaries** – Set the maximum permissions that an identity-based policy can grant to an IAM entity. For more information, see Permissions boundaries for IAM entities in the *IAM User Guide*.
- **Service control policies (SCPs)** – Specify the maximum permissions for an organization or organizational unit in AWS Organizations. For more information, see Service control policies in the *AWS Organizations User Guide*.
- **Resource control policies (RCPs)** – Set the maximum available permissions for resources in your accounts. For more information, see Resource control policies (RCPs) in the *AWS Organizations User Guide*.
- **Session policies** – Advanced policies passed as a parameter when creating a temporary session for a role or federated user. 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 create indexes or views, or to modify them or any other Resource Explorer settings, you must have additional permissions.

Your search experience is automatically enabled based on your IAM permissions. If you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy, you can immediately search all tagged resources and supported untagged resources created after the immediate resource discovery release. For complete resource inventory with automatic updates, you'll also need the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). After the service-linked role is created in your account by any user, subsequent users need only need, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to create an index and view for full results in a Region on first search. Organizations can control access by denying the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy to prevent all search access, or denying `iam:CreateServiceLinkedRole` to limit users to partial results only when a service-linked role does not already exist in an account.

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. 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. As a best practice, specify a resource using its Amazon Resource Name (ARN). For actions that don't support resource-level permissions, 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 create 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 specifies when statements execute based on defined criteria. 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. 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. When users with both, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) initiate their first resource search, Resource Explorer automatically creates the service-linked role at the account level. Once the service-linked role exists, subsequent regions are automatically enabled when users with search permissions invoke search operations. For details about Resource Explorer service-linked roles, see Using service-linked roles for Resource Explorer.

Troubleshooting service-linked role creation: If users lack the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy), they will receive an error when attempting to create the service-linked role. To resolve this issue, users must either get permission from an administrator or sign in with a role that has the required permission.

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

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 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": [
                "resource-explorer-2:GetView",
                "resource-explorer-2:Search"
            ],
            "Resource": "arn:aws:resource-explorer-2:*:*: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": "resource-explorer-2:CreateView",
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {"aws:ResourceTag/Allow-Create-View":
 "${aws:PrincipalTag/Allow-Create-View}"}
            }
        }
    ]
}
```

Allow principals to view their own permissions

JSON

```
{
    "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": [
    "resource-explorer-2:AssociateDefaultView",
    "resource-explorer-2:BatchGetView",
    "resource-explorer-2:CreateIndex",
    "resource-explorer-2:CreateView",
    "resource-explorer-2:DeleteIndex",
    "resource-explorer-2:DeleteView",
    "resource-explorer-2:DisassociateDefaultView",
    "resource-explorer-2:GetDefaultView",
    "resource-explorer-2:GetIndex",
    "resource-explorer-2:ListIndexes",
    "resource-explorer-2:ListSupportedResourceTypes",
    "resource-explorer-2:ListTagsForResource",
    "resource-explorer-2:ListViews",
    "resource-explorer-2:TagResource",
    "resource-explorer-2:UntagResource",
    "resource-explorer-2:UpdateIndexType",
    "resource-explorer-2: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` (included in the AWSResourceExplorerFullAccess managed policy) – 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, `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` (included in the AWSResourceExplorerFullAccess managed policy) – 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 *AWS Managed Policy Reference Guide*.

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 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: • `kinesis:ListTagsForResource`	February 04, 2026
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: • `events:ListTagsForResource`	December 16, 2025
AWSResourceExplorerServiceRolePolicy - Updated	Resource Explorer modified the permissions in the	November 17, 2025

Change	Description	Date
policy permissions to view additional resource types	service-linked role policy AWSResourceExplorerServiceRolePolicy. Permissions were added that allows Resource Explorer to view additional resource types: • `dynamodb:DescribeTimeToLive` • `lambda:GetRuntimeManagementConfig` • `logs:DescribeResourcePolicies`	

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: • `organizations:DescribeResourcePolicy` • `elasticloadbalancing:DescribeLoadBalancerPolicies` • `elasticloadbalancing:DescribeLoadBalancerPolicyTypes` • `elasticloadbalancing:DescribeTargetGroupAttributes` • `elasticloadbalancing:DescribeTargetHealth` • `kinesis:DescribeStreamSummary` • `kinesis:ListTagsForStream`	October 13, 2025

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: • `logs:ListTagsForResource` • `cloudformation:GetResource` • `cloudfront:GetDistribution` • `cloudfront:GetDistributionConfig` • `dynamodb:DescribeContinuousBackups` • `dynamodb:DescribeContributorInsights` • `dynamodb:DescribeKinesisStreamingDestination` • `dynamodb:DescribeTable` • `dynamodb:GetResourcePolicy` • `dynamodb:ListTagsOfResource`	September 24, 2025

Change	Description	Date
	• ecs:ListTagsForResource • elasticloadbalancing:DescribeCapacityReservation • elasticloadbalancing:DescribeLoadBalancerAttributes • elasticloadbalancing:DescribeTags • events:DescribeRule • events:ListTargetsByRule • iam:GetRole • iam:GetRolePolicy • iam:ListAttachedRolePolicies • iam:ListRolePolicies • lambda:GetEventSourceMapping • lambda:GetFunction • lambda:GetFunctionCodeSigningConfig • lambda:GetFunctionRecursionConfig • lambda:ListTags • logs:DescribeIndexPolicies • logs:GetDataProtectionPolicy	

Change	Description	Date
	• s3:GetAccelerateConfiguration • s3:GetAnalyticsConfiguration • s3:GetBucketCORS • s3:GetBucketLogging • s3:GetBucketMetadataTableConfiguration • s3:GetBucketNotification • s3:GetBucketObjectLockConfiguration • s3:GetBucketOwnershipControls • s3:GetBucketPublicAccessBlock • s3:GetBucketTagging • s3:GetBucketVersioning • s3:GetBucketWebsite • s3:GetEncryptionConfiguration • s3:GetIntelligentTieringConfiguration • s3:GetInventoryConfiguration • s3:GetLifecycleConfiguration	

Change	Description	Date
	• s3:GetMetricsConfiguration • s3:GetReplicationConfiguration • sns:GetDataProtectionPolicy • sns:GetTopicAttributes • sns:ListSubscriptionsByTopic • sns:ListTagsForResource • sqs:GetQueueAttributes • sqs:ListQueueTags	

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: • `aoss:ListCollections` • `app-integrations:ListApplications` • `appconfig:ListEnvironments` • `appconfig:ListExtensionAssociations` • `bedrock:ListAgentAliases` • `cloudtrail:ListDashboards` • `comprehend:ListFlywheels` • `connect:ListEvaluationForms` • `connect:ListInstanceAttributes` • `connect:ListPrompts` • `connect:ListRoutingProfileQueues`	September 15, 2025

Change	Description	Date
	• connect:ListRoutingProfiles • connect:ListSecurityProfiles • devicefarm:ListInstanceProfiles • directconnect:DescribeDirectConnectGateways • ec2:DescribeInstanceConnectEndpoints • ec2:DescribeVpcBlockPublicAccessExclusions • eks:DescribeAccessEntry • eks:DescribeAddon • eks:DescribeFargateProfile • eks:DescribeIdentityProviderConfig • eks:DescribeNodegroup • eks:ListAccessEntries • eks:ListAddons • eks:ListFargateProfiles • eks:ListIdentityProviderConfigs • eks:ListNodegroups	

Change	Description	Date
	• fms:ListPolicies • fms:ListProtocolsLists • fsx:DescribeBackups • glue:ListDataQualityRulesets • glue:ListRegistries • iottwinmaker:ListSyncJobs • ivs:ListEncoderConfigurations • ivs:ListIngestConfigurations • ivs:ListPlaybackRestrictionPolicies • ivs:ListStorageConfigurations • lightsail:GetBuckets • lightsail:GetCertificates • lightsail:GetContainerServices • macie2:ListAllowLists • macie2:ListCustomDataIdentifiers • macie2:ListFindingsFilters • mediatailor:ListVodSources	

Change	Description	Date
	• network-firewall:ListRuleGroups • organizations:ListPolicies • partnercentral:ListEngagementInvitations • partnercentral:ListEngagements • partnercentral:ListOpportunities • partnercentral:ListResourceSnapshotJobs • partnercentral:ListResourceSnapshots • profile:ListIntegrations • profile:ListProfileObjectTypes • resiliencehub:ListApps • route53-recovery-control-config:ListRoutingControls • route53-recovery-readiness:ListCells • s3:ListStorageLensGroups • s3express:ListAllMyDirectoryBuckets	

Change	Description	Date
	• sagemaker:ListHubContents • sagemaker:ListMonitoringSchedules • ssm-incidents:ListResponsePlans • verifiedpermissions:ListPolicyStores • vpc-lattice:ListListeners • vpc-lattice:ListServiceNetworkServiceAssociations • vpc-lattice:ListServiceNetworks • vpc-lattice:ListServices • vpc-lattice:ListTargetGroups • wafv2:ListIPSets • wafv2:ListRegexPatternSets • wafv2:ListRuleGroups • wisdom:ListContents • workspaces-web:ListPortals	

Change	Description	Date
AWSResourceExplorerServiceRolePolicy - Updated policy permissions for AWS policy best practices	Resource Explorer modified the permissions in the service-linked role policy AWSResourceExplorerServiceRolePolicy. Permissions were removed for resource types that are not currently supported by Resource Explorer. For the latest version of this policy, see AWSResourceExplorerServiceRolePolicy in the *AWS Managed Policy Reference Guide*. The following permissions were removed for unsupported resource types: • `amplify:ListBackendEnvironments` • `amplifyuibuilder:ListComponents` • `amplifyuibuilder:ListThemes` • `backup:ListFrameworks` • `bedrock:ListBlueprints` • `bedrock:ListCustomModels` • `directconnect:DescribeConnections`	September 5, 2025

Change	Description	Date
	• directconnect:DescribeDirectConnectGateways • directconnect:DescribeLags • directconnect:DescribeVirtualInterfaces • dynamodb:ListStreams • ec2:DescribeElasticGpus • ec2:DescribeExportImageTasks • ec2:DescribeExportTasks • ec2:DescribeImportImageTasks • ec2:DescribeImportSnapshotTasks • ec2:DescribeVpcEndpointServices • ecs:ListTasks • eks:ListAccessEntries • eks:ListAddons • eks:ListFargateProfiles • eks:ListIdentityProviderConfigs • eks:ListNodegroups	

Change	Description	Date
	• elasticache:DescribeCacheSecurityGroups • fms:ListPolicies • fms:ListProtocolsLists • fsx:DescribeStorageVirtualMachines • fsx:DescribeVolumes • iottwinmaker:ListScenes • iottwinmaker:ListSyncJobs • kinesis:ListStreamConsumers • kms:ListAliases • lambda:ListAliases • lambda:ListLayerVersions • lambda:ListLayers • lex:ListBots • lightsail:GetBuckets • lightsail:GetCertificates • lightsail:GetContainerServices • logs:DescribeLogStreams • macie2:ListAllowLists	

Change	Description	Date
	• macie2:ListCustomDataIdentifiers • macie2:ListFindingsFilters • redshift:DescribeScheduledActions • resiliencehub:ListApps • resource-explorer-2:ListIndexes • robomaker:ListRobotApplications • robomaker:ListSimulationApplications • sagemaker:ListDataQualityJobDefinitions • sagemaker:ListModelBiasJobDefinitions • sagemaker:ListModelExplainabilityJobDefinitions • sagemaker:ListModelQualityJobDefinitions • sagemaker:ListMonitoringSchedules • scheduler:ListSchedules • ssm-incidents:ListResponsePlans	

Change	Description	Date
	• ssm:DescribeAutomationExecutions • ssm:DescribePatchBaselines • ssm:ListInventoryEntries • timestream:ListDatabases • timestream:ListScheduledQueries • xray:GetGroups	

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: • `apigateway:restapis/deployments` • `apigateway:restapis/resources/methods` • `apigateway:restapis/resources` • `apigateway:restapis/stages` • `apigateway:apis` • `apigateway:apis/routes` • `apigateway:apis/stages` • `appmesh:mesh/virtualGateway/gatewayRoute` • `appmesh:mesh/virtualRouter/route` • `appmesh:mesh/virtualGateway`	August 4, 2025

Change	Description	Date
	• appmesh:mesh/virtualRouter • apprunner:autoscalingconfiguration • apprunner:connection • autoscaling:autoScalingGroup • backup-gateway:hypervisor • batch:job-definition • bedrock:agent • bedrock:application-inference-profile • bedrock:data-automation-project • bedrock:flow • bedrock:guardrail • bedrock:knowledge-base • bedrock:prompt • bedrock:prompt-router • chime:app-instance • chime:app-instance/bot • chime:app-instance/user	

Change	Description	Date
	• chime:media-insights-pipeline-configuration • chime:media-pipeline • chime:media-pipeline-kinesis-video-stream-pool • chime:sma • chime:vc • config:config-rule • connect:instance/operating-hours • dms:cert • eks:eks-anywhere-subscription • eks:podidentityassociation • emr-containers:jobtemplates • emr-containers:virtualclusters/endpoints • emr-containers:securityconfigurations • events:api-destination • gamelift:script • guardduty:detector	

Change	Description	Date
	• guardduty:malware-protection-plan • guardduty:detector/publishingDestination • inspector2:filter • iot:billinggroup • iot:fleetmetric • iot:scheduledaudit • iot:thinggroup • iot:thingtype • iotfleethub:application • iotsitewise:access-policy • iotsitewise:portal • iotsitewise:project • iotwireless:Destination • iotwireless:DeviceProfile • iotwireless:FuotaTask • iotwireless:MulticastGroup • iotwireless:SidewalkAccount • iotwireless:WirelessDevice • iotwireless:WirelessGateway	

Change	Description	Date
	• iotwireless:WirelessGatewayTaskDefinition • ivs:playback-key • kendra:index/access-control-configuration • kendra:index/data-source • kendra:index/faq • kendra:index/featured-results-set • kendra:index/query-suggestions-block-list • kendra:index/thesaurus • kendra:index/experience • kinesisvideo:channel • license-manager:grant • mediapackage-vod:assets • mediastore:container • mediatailor:channel • mediatailor:liveSource • memorydb:snapshot	

Change	Description	Date
	• mobiletargeting:templates/SMS • mobiletargeting:templates/PUSH • mobiletargeting:templates/EMAIL • mq:configuration • profile:domains • proton:environment-template • proton:service-template • redshift:hsmclientcertificate • s3:multiregionaccesspoint • sagemaker:action • sagemaker:algorithm • sagemaker:app • sagemaker:artifact • sagemaker:code-repository • sagemaker:context • sagemaker:endpoint-config • sagemaker:experiment • sagemaker:experiment-trial • sagemaker:experiment-trial-component	

Change	Description	Date
	• sagemaker:human-task-ui • sagemaker:image-version • sagemaker:inference-component • sagemaker:inference-experiment • sagemaker:model-package-group • sagemaker:model-package • sagemaker:model-card • sagemaker:notebook-instance-lifecycle-config • sagemaker:project • sagemaker:space • sagemaker:user-profile • sagemaker:workforce • sagemaker:cluster • sagemaker:flow-definition • sagemaker:hub • sagemaker:mlflow-tracking-server • sagemaker:studio-lifecycle-config • sagemaker:workteam	

Change	Description	Date
	• ses:dedicated-ip-pool • ssm:session • synthetics:canary • transfer:server • transfer:user	
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: • UpdateIndexType • CreateIndex • CreateView • AssociateDefaultView • DeleteIndex	July 23, 2025

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: • `appconfig:applications` • `appconfig:deploymentstrategies` • `ce:getanomalymonitors` • `ce:getanomalysubscriptions` • `cloudformation:resources` • `cloudfront:continuousdeploymentpolicies` • `cloudtrail:channels` • `codedeploy:applications` • `codedeploy:deploymentconfigs` • `events:archives` • `events:endpoints` • `gamelift:locations`	May 7, 2025

Change	Description	Date
	• groundstation:missionprofiles • inspector:assessmenttemplates • iot:cacertificates • iot:certificates • iotdeviceadvisor:suitedefinitions • iotfleetwise:decodermanifests • iotfleetwise:modelmanifests • iotfleetwise:signalcatalogs • managedblockchain:accessors • oam:sink • omics:referencestores • omics:rungroups • omics:workflows • personalize:solutions • pipes:pipes • scheduler:schedulegroups • scheduler:schedules • schemas:discoverers • transfer:certificates • transfer:connectors	

Change	Description	Date
	• `transfer:profiles`	

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: • cloud9:environment • cloudtrail:eventdatastore • connect:instance/rule • connect:instance/task-template • connect:phone-number • datapipeline:pipeline • dax:cache • devicefarm:project • devicefarm:testgrid-project • ds:directory • ec2:ipam-resource-discovery • ec2:ipam-resource-discovery-association • elasticloadbalancing:listener-rule/net	March 21, 2025

Change	Description	Date
	• events:connection • forecast:dataset-import-job • forecast:forecast • forecast:forecast-export-job • forecast:predictor • forecast:predictor-backtest-export-job • geo:map • grafana:workspaces • groundstation:dataflow-endpoint-group • iot:ruledestination • iotfleetwise:vehicle • ivschat:logging-configuration • ivschat:room • lookoutmetrics:AnomalyDetector • m2:env • outposts:site • quicksight:theme • route53-recovery-control:cluster • route53-recovery-control:controlpanel/safetyrule	

Change	Description	Date
	• route53-recovery-readiness:readiness-check • route53resolver:firewall-rule-group-association • rum:appmonitor • sagemaker:app-image-config • servicediscovery:service • synthetics:group • transfer:agreement • transfer:profile • 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: • acm:certificate • codepipeline:webhook • comprehend:document-classifier • comprehend:entity-recognizer • databrew:job • databrew:project • dataexchange:data-sets • dms:es • dms:subgrp • elasticmapreduce:cluster • emr-containers:virtualclusters • frauddetector:external-model • frauddetector:model • fsx:file-system • glacier:vaults • glue:crawler	January 6, 2025

Change	Description	Date
	• greengrass:connectorsDefinition • greengrass:coresDefinition • greengrass:devicesDefinition • greengrass:functionsDefinition • greengrass:loggersDefinition • greengrass:resourcesDefinition • greengrass:subscriptionsDefinition • mq:broker • route53:domain • ses:contact-list • ses:configuration-set • ses:identity • 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: • IamAction(value=apigateway:Get) • airflow:ListEnvironments • appflow:ListFlows • appmesh:ListMeshes • appmesh:ListVirtualNodes • appmesh:ListVirtualServices • auditmanager:GetAccountStatus • auditmanager:ListAssessments • backup:ListBackupVaults • codeguru-reviewer:ListRepositoryAssociations • connect:ListInstances • connect:ListQuickConnects • connect:ListUsers	November 21, 2024

Change	Description	Date
	• databrew:ListSchedules • datasync:ListLocations • datasync:ListTasks • dms:DescribeEndpoints • dms:DescribeReplicationInstances • dms:DescribeReplicationTasks • eks:ListClusters • gamelift:DescribeGameSessionQueues • gamelift:DescribeMatchmakingConfigurations • gamelift:DescribeMatchmakingRuleSets • gamelift:ListBuilds • glue:ListMLTransforms • groundstation:ListConfigs • guardduty:ListDetectors • guardduty:ListFilters • guardduty:ListIPSets	

Change	Description	Date
	• guardduty:ListThreatIntelSets • iotsitewise:ListDashboards • iotsitewise:ListPortals • iotsitewise:ListProjects • iotwireless:ListServiceProfiles • ivs:ListRecordingConfigurations • kendra:ListIndices • macie2:ListCustomDataIdentifiers • macie2:ListFindingsFilters • memorydb:DescribeSubnetGroups • mobiletargeting:GetCampaigns • proton:ListEnvironmentAccountConnections • quicksight:DescribeAccountSubscription • quicksight:ListDataSets • quicksight:ListDataSources	

Change	Description	Date
	• quicksight:ListTemplates • ram:GetResourceShares • robomaker:ListRobotApplications • robomaker:ListSimulationApplications • route53resolver:ListResolverQueryLogConfigs • sagemaker:ListDomains • sagemaker:ListEndpoints • sagemaker:ListFeatureGroups • sagemaker:ListImages • sagemaker:ListPipelines • transfer:ListWorkflows • 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: • apprunner:ListVpcConnectors • backup:ListReportPlans • emr-serverless:ListApplications • events:ListEventBuses • geo:ListPlaceIndexes • geo:ListTrackers • greengrass:ListComponents • greengrass:ListComponentVersions • iot:ListRoleAliases • iottwinmaker:ListComponentTypes • iottwinmaker:ListEntities • iottwinmaker:ListScenes • kafka:ListConfigurations	December 12, 2023

Change	Description	Date
	• kms:ListKeys • kinesisanalytics:ListApplications • lex:ListBots • lex:ListBotAliases • mediapackage-vod:ListPackagingConfigurations • mediapackage-vod:ListPackagingGroups • mq:ListBrokers • personalize:ListDatasetGroups • personalize:ListDatasets • personalize:ListSchemas • route53:ListHealthChecks • route53:ListHostedZones • secretsmanager:ListSecrets	
New managed policy	Resource Explorer added the following AWS managed policy: • AWSResourceExplorerOrganizationsAccess	November 14, 2023

Change	Description	Date
Updated managed policies	Resource Explorer updated the following AWS managed policies to support multi-account search: • AWSResourceExplorerFullAccess • 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: • `organizations:ListAWSServiceAccessForOrganization` • `organizations:DescribeAccount` • `organizations:DescribeOrganization` • `organizations:ListAccounts` • `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: • accessanalyzer:analyzer • acmpca:certificateauthority • amplify:app • amplify:backendenvironment • amplify:branch • amplify:domainassociation • amplifyuibuilder:component • amplifyuibuilder:theme • appintegrations:eventintegration • apprunner:service • appstream:appblock • appstream:application • appstream:fleet • appstream:imagebuilder • appstream:stack • appsync:graphqlapi • aps:rulegroupsnamespace • aps:workspace • apigateway:restapi • apigateway:deployment	October 17, 2023

Change	Description	Date
	• athena:datacatalog • athena:workgroup • autoscaling:autosc alinggroup • backup:backupplan • batch:computeenvironment • batch:jobqueue • batch:schedulingpolicy • cloudformation:stack • cloudformation:stackset • cloudfront:fieldlevelencryp tionconfig • cloudfront:fieldlevelencryp tionprofile • cloudfront:originaccesscont rol • cloudtrail:trail • codeartifact:domain • codeartifact:repository • codecommit:repository • codeguruprofiler:profilingg roup • codestarconnection s:connection • databrew:dataset • databrew:recipe • databrew:ruleset • detective:graph • directoryservices:directory • ec2:carriergateway	

Change	Description	Date
	• ec2:verifiedaccessendpoint • ec2:verifiedaccessgroup • ec2:verifiedaccessinstance • ec2:verifiedaccesstrustprovider • ecr:repository • elasticache:cachesecuritygroup • elasticfilesystem:accesspoint • events:rule • evidently:experiment • evidently:feature • evidently:launch • evidently:project • finspace:environment • firehose:deliverystream • faultinjectionsimulator:experimenttemplate • forecast:datasetgroup • forecast:dataset • frauddetector:detector • frauddetector:entitytype • frauddetector:eventtype • frauddetector:label • frauddetector:outcome • frauddetector:variable • gamelift:alias • globalaccelerator:accelerator	

Change	Description	Date
	• globalaccelerator: endpointgroup • globalaccelerator:listener • glue:database • glue:job • glue:table • glue:trigger • greengrass:group • healthlake:fhirdatastore • iam:virtualmfadevice • imagebuilder:compo nentbuildversion • imagebuilder:component • imagebuilder:conta inerrecipe • imagebuilder:distributionco nfiguration • imagebuilder:image buildversion • imagebuilder:imagepipeline • imagebuilder:imagerecipe • imagebuilder:image • imagebuilder:infrastructure configuration • iot:authorizer • iot:jobtemplate • iot:mitigationaction • iot:provisioningtemplate • iot:securityprofile • iot:thing	

Change	Description	Date
	• iot:topicruledestination • iotanalytics:channel • iotanalytics:dataset • iotanalytics:datastore • iotanalytics:pipeline • iotevents:alarmmodel • iotevents:detectormodel • iotevents:input • iotsitewise:assetmodel • iotsitewise:asset • iotsitewise:gateway • iottwinmaker:workspace • ivs:channel • ivs:streamkey • kafka:cluster • kinesisvideo:stream • lambda:alias • lambda:layerversion • lambda:layer • lookoutmetrics:alert • lookoutvision:project • mediapackage:channel • mediapackage:origi nendpoint • mediatailor:playba ckconfiguration • memorydb:acl • memorydb:cluster • memorydb:parametergroup	

Change	Description	Date
	• memorydb:user • mobiletargeting:app • mobiletargeting:segment • mobiletargeting:template • networkfirewall:firewallpolicy • networkfirewall:firewall • networkmanager:globalnetwork • networkmanager:device • networkmanager:link • networkmanager:attachment • networkmanager:corenetwork • panorama:package • qldb:journalkinesisstreamsforledger • qldb:ledger • rds:bluegreendeployment • refactorspaces:application • refactorspaces:environment • refactorspaces:route • refactorspaces:service • rekognition:project • resiliencehub:app • resiliencehub:resiliencypolicy • resourcegroups:group • route53:recoverygroup	

Change	Description	Date
	• route53:resourceset • route53:firewalldomain • route53:firewallrulegroup • route53:resolverendpoint • route53:resolverrule • sagemaker:model • sagemaker:notebook instance • signer:signingprofile • ssmincidents:responseplan • ssm:inventoryentry • ssm:resourcedatasync • states:activity • timestream:database • wisdom:assistant • wisdom:assistantas sociation • 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: • codebuild:project • codepipeline:pipeline • cognito:identitypool • cognito:userpool • ecr:repository • efs:filesystem • elasticbeanstalk:application • elasticbeanstalk:applicationversion • elasticbeanstalk:environment • iot:policy • iot:topicrule • stepfunctions:statemachine • 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: • elasticache:cluster • elasticache:globalreplicationgroup • elasticache:parametergroup • elasticache:replicationgroup • elasticache:reserved-instance • elasticache:snapshot • elasticache:subnetgroup • elasticache:user • elasticache:usergroup • lambda:code-signing-config • lambda:event-source-mapping • sqs:queue	March 7, 2023

Change	Description	Date
New managed policies	Resource Explorer added the following AWS managed policies: • AWSResourceExplorerFullAccess • AWSResourceExplorerReadOnlyAccess • 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. To see the latest version of this AWS managed policy, `AWSResourceExplorerServiceRolePolicy` in the *AWS Managed Policy Reference Guide*. To see permission changes to this policy, see Resource Explorer updates to AWS managed policies.

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 first access Resource Explorer with appropriate permissions, 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 CSPM – 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 CSPM.

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.

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 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

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

Topics

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

Troubleshooting permission-based access issues

Resource Explorer provides different user experiences based on your IAM permissions. Use this section to troubleshoot issues related to permission-based access and search results.

I'm getting partial search results instead of complete results

If you're receiving partial search results, this indicates you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy but lack `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy), or the service-linked role hasn't been created in your account yet.

- **To get complete results:** Obtain `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy) from your administrator, or sign in with a role that has this permission. Once you initiate a search with both permissions, Resource Explorer will automatically create the service-linked role and provide complete results.
- **If the service-linked role already exists:** Verify you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. Users with search permission get complete results after searching in a Region once the service-linked role exists in the account.

> **Note**
>
> Automatic setup may not happen in this case if an index was previously deleted or the aggregator index already exists

- **Regional differences:** Results may vary by Region based on index types. Regions with user-owned indexes provide complete results, while Regions with only Resource Explorer-owned indexes provide partial results.

Service-linked role creation issues

If you receive an error when Resource Explorer attempts to create the service-linked role during your first search, this indicates you lack the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy).

- **Resolution:** Get permission from your administrator OR sign in with a role that has the `iam:CreateServiceLinkedRole` permission.

> **Note**
>
> **Note:** The service-linked role only needs to be created once per account. After it's created by any user with the appropriate permission, all users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy are able to create an index and view for full results in a Region on first search.

I can't access Resource Explorer search functionality

If you receive access denied errors when trying to use Resource Explorer search, you lack at minimum the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy.

- **Resolution:** Contact your administrator to obtain the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. These permissions are also a subset of the `ResourceExplorerFullAccess` managed policy
- **Organizational control:** If your organization wants to prevent access to Resource Explorer search functionality, administrators can disallow the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy.

Indexing progress and completion issues

When Resource Explorer automatically creates indexes and views, you may see indexing progress indicators in the console.

- **Blue banner "Completing Resource Explorer setup":** This indicates indexing is in progress. You can search immediately and receive partial results while indexing completes in the background.
- **Green completion banner:** This indicates that the user-index is setup. Refresh to view full results.
- **Timeline expectations:** Initial indexing typically completes within a few hours, depending on the number of resources in your account. You can use Resource Explorer immediately while indexing continues.
- **If indexing appears stuck:** Indexing runs automatically in the background. If you don't see progress after several hours, verify your permissions and try refreshing the console.

I get an "access denied" message when I make a request to Resource Explorer

Access denied errors can occur when accessing Resource Explorer search functionality or when trying to configure enhanced features like custom views or cross-Region search.

- **For basic search access:** Verify you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. This permission provides immediate access to search functionality.
- **For complete search results:** Verify you have both the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy and the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy), or that the service-linked role already exists in your account.

- **For enhanced features:** 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?

The following list provides reasons why some resources might not appear in your search results as expected:

Initial indexing isn't complete

After you first access Resource Explorer, the service automatically enables search capabilities based on your IAM permissions. It can then 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 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.

User indexes aren't configured in the console Region

If a user index isn't configured in a Region, you will see partial results. For more information, see Understanding the immediate resource discovery experience.

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 when you have, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy. If you don't have access to resource results, obtain permission from your administrator or sign in with a role that has this permission.

Why does 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
Added support for new resource types	Resource Explorer added support for 9 new resources from Amazon Elastic Kubernetes Service.	March 31, 2026
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.	February 4, 2026
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.	December 16, 2025
Updated managed policy	Resource Explorer added support to view additional resource types. The	November 17, 2025

	AWSResourceExplorerServiceRolePolicy AWS managed policy has been updated to grant Resource Explorer access to view additional resource types.	
Added support for new resource types	Resource Explorer added support for 47 new resources from AWS services including AWS Amplify and AWS WAFV2.	October 27, 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.	October 13, 2025

Immediate resource discovery feature release	Resource Explorer now provides automatic resource search functionality without requiring manual setup. When you first access Resource Explorer through the console, Unified Search, CLI, or API, the service automatically enables search capabilities based on your IAM permissions. Users with, at minimum, the permissions in the `AWSResourceExplorerReadOnlyAccess` managed policy can immediately search all tagged resources and supported untagged resources created after the immediate resource discovery release. For complete resource inventory with automatic updates, users also need the `iam:CreateServiceLinkedRole` permission (included in the AWSResourceExplorerFullAccess managed policy). This feature eliminates the traditional setup barrier and provides immediate value while maintaining all existing functionality for customers who have already configured Resource Explorer. Cross-Region search capabilities remain	October 6, 2025

	available as an optional enhancement.	
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.	September 24, 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.	September 15, 2025

Updated managed policy	The AWSResourceExplorerServiceRolePolicy AWS managed policy has been updated to remove unnecessary permissions for resource types that are not currently supported by Resource Explorer. This update ensures the policy complies with AWS policy best practices. Permissions for resource types that are currently supported by Resource Explorer are unchanged.	September 5, 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.	August 4, 2025
Added support for new resource types	Resource Explorer added support for 121 new resources from AWS services including Amazon MQ and AWS Shield.	August 4, 2025

Added support for new search query filter operator	Resource Explorer added support for a new OR filter operator that you can use to separate multiple values in a single filter. If you use a comma-separated list of values in a filter, Resource Explorer returns search results matching any of those filter values.	July 31, 2025
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 CSPM.	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 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