AWS Resource Groups

User Guide

AWS Resource Groups: User Guide

Table of Contents

What Is AWS Resource Groups?

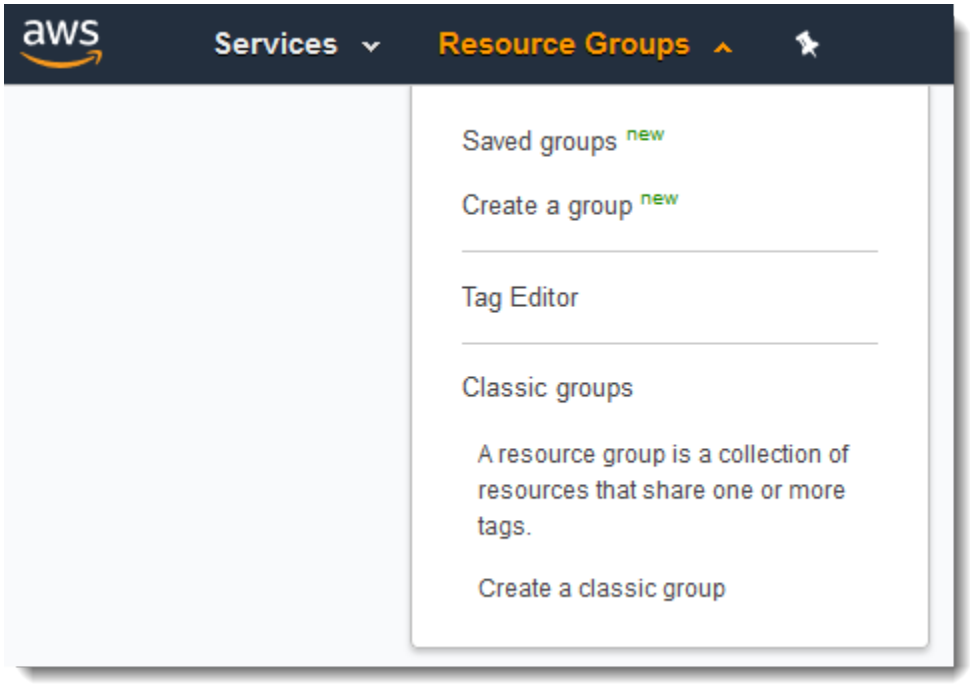
You can use *resource groups* to organize your AWS resources. Resource groups make it easier to manage and automate tasks on large numbers of resources at one time. This guide shows you how to create and manage resource groups in AWS Resource Groups.

You can access Resource Groups through any of the following entry points.

- On the navigation bar of the AWS Management Console.
- In the AWS Systems Manager console, from the left navigation pane entry for Resource Groups.
- By using the Resource Groups API, in AWS CLI commands or AWS SDK programming languages.

To work with resource groups on the AWS Management Console home

1. Sign in to the AWS Management Console.
2. On the navigation bar, choose **Resource Groups**.



3. Choose a resource group from **Saved Groups**, or choose **Create a Group**.

To work with resource groups in AWS Systems Manager

1. Sign in to the AWS Management Console.
2. On the console home page, in **Management Tools**, choose **AWS Systems Manager**.
3. On the **AWS Systems Manager** home page, choose **Explore Resource Groups**.

What Are Resource Groups?

In AWS, a *resource* is an entity that you can work with. Examples include an Amazon EC2 instance, an AWS CloudFormation stack, or an Amazon S3 bucket. If you work with multiple resources, you might find it useful to manage them as a group rather than move from one AWS service to another for each task. If you manage large numbers of related resources, such as EC2 instances that make up an application layer, you likely need to perform bulk actions on these resources at one time. Examples of bulk actions include:

- Applying updates or security patches.
- Upgrading applications.
- Opening or closing ports to network traffic.
- Collecting specific log and monitoring data from your fleet of instances.

A *resource group* is a collection of AWS resources that are all in the same AWS region, that match criteria provided in a query, and that share one or more *tags* or portions of tags. You build queries in the Resource Groups console or pass them as arguments to Resource Groups commands in the AWS CLI. *Queries* include lists of resources that are specified in the following format `AWS::service::resource`, and tags. *Tags* are keys that help identify and sort your resources within your organization. Optionally, tags include values for keys.

Resource groups can be *nested*; a resource group can contain existing resource groups in the same region.

By default, the AWS Management Console is organized by AWS service. But with Resource Groups, you can create a custom console that organizes and consolidates information based on criteria that you specify in tags. The following list describes some of the cases in which tagging and resource grouping can help organize your resources.

- An application that has different phases, such as development, staging, and production.
- Projects managed by multiple departments or individuals.
- A set of AWS resources that you use together for a common project or that you want to manage or monitor as a group.
- A set of resources related to applications that run on a specific platform, such as Android or iOS.

For example, you are developing a web application, and you are maintaining separate sets of resources for your alpha, beta, and release stages. Each version runs on Amazon EC2 with an Amazon Elastic Block Store storage volume. You use Elastic Load Balancing to manage traffic and Route 53 to manage your domain. Without Resource Groups, you might have to access multiple consoles just to check the status of your services or modify the settings for one version of your application.

With Resource Groups, you use a single page to view and manage your resources. For example, let's say you use the tool to create a resource group for each version—alpha, beta, and release—of your application. To check your resources for the alpha version of your application, open your resource group. Then view the consolidated information on your resource group page. To modify a specific resource, choose the resource's links on your resource group page to access the service console that has the settings that you need.

Differences Between AWS Resource Groups and Legacy Resource Groups

The following table describes key differences between the AWS Resource Groups service, and the older, classic Resource Groups.

	AWS Resource Groups	Legacy Resource Groups
API support	Has a public API. For more information about the AWS Resource Groups API, see the AWS Resource Groups API Reference.	No public API. You can create and manage legacy resource groups in the AWS Management Console only.
Region support	Regional; all resources in a group that you create with AWS Resource Groups are located in the same region.	Cross-regional
Permissions	Per AWS account	Per user
Requirements	To create a group, you must choose resource types that have at least one tag key assigned, and specify at least a tag key value.	To create a group, you must choose a tag key from a drop-down list. Specifying resource types is optional.
Entry points	You can open AWS Resource Groups from the upper left of the AWS Management Console. This opens AWS Systems Manager, where you work with AWS Resource Groups. When you choose AWS Resource Groups, the URL is https://console.aws.amazon.com/resource-groups/groups in Systems Manager. You can also create and manage groups by using the AWS CLI and API.	You can open legacy Resource Groups from the upper left of the AWS Management Console. When you choose legacy Resource Groups, the URL is https://resources.console.aws.amazon.com/r/group.
Purposes	Perform tasks such as Systems Manager Automation on multiple resources at one time; view insights and monitoring information about grouped resources.	Get monitoring data about resources, such as CloudWatch alarms.
Support for nested groups (a resource group of other resource groups)	Yes. You can create a resource group that contains other resource groups in the same region that were created in the new service.	No.

AWS Resource Groups and Permissions

The new Resource Groups feature permissions (the feature that is covered by this guide) are at the account level. As long as users who are sharing your account have the correct IAM permissions, they can work with resource groups that you create.

In the older, classic Resource Groups, however, if you use AWS Identity and Access Management (IAM) to create multiple users in the same account, each of those users has their own, individual resource groups.

These groups are not visible to other users. For information about creating IAM users, see Creating an IAM User in the *IAM User Guide*.

Tags are properties of a resource, so they are shared across your entire account. Users in a department or specialized group can draw from a common vocabulary (tags) to create resource groups that are meaningful to their roles and responsibilities. Having a common pool of tags also means that when users share a resource group, they don't have to worry about missing or conflicting tag information.

AWS Resource Groups Resources

In Resource Groups, the only available resource is a group. Groups have unique Amazon Resource Names (ARNs) associated with them. For more information about ARNs, see Amazon Resource Names (ARN) and AWS Service Namespaces in the *Amazon Web Services General Reference*.

Resource Type	ARN Format
Resource Group	`arn:aws:resource-groups:`*`region`*`:`*`account`*`:group/`*`group-name`*

How Tagging Works

Tags are key and value pairs that act as metadata for organizing your AWS resources. With most AWS resources, you have the option of adding tags when you create the resource, whether it's an Amazon EC2 instance, an Amazon S3 bucket, or other resource. However, you can also add tags to multiple, supported resources at once by using Tag Editor. You build a query for resources of various types, and then add, remove, or replace tags for the resources in your search results. Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query.

For more information about tagging, see Working with Tag Editor (p. 21) in this guide. You can tag supported resources (p. 4) by using Tag Editor, and some additional resources by using tagging functionality in the service console in which you create and manage the resource.

Supported Resources

You can use the AWS Management Console or the AWS CLI to add tags to many AWS resources. This topic describes resources that are currently supported by tags.

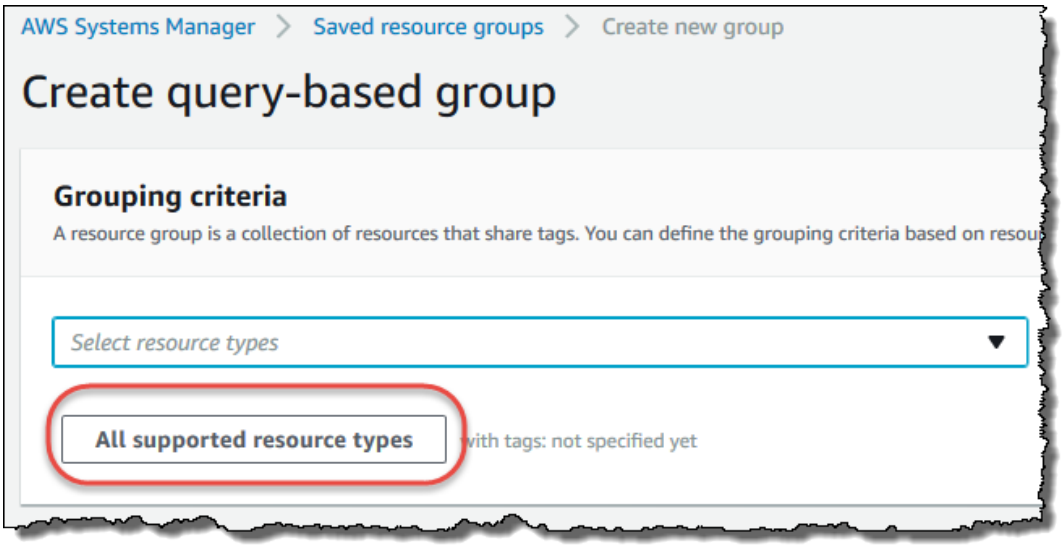
Topics

Supported Resources for Resource Groups

You can use the Resource Groups tool in the AWS Management Console to create groups for the following tagged AWS resources.

> **Important**
> A resource group based on a query for **All supported resource types** can add members automatically over time, as new resources are supported by Resource Groups. When you run automations or other bulk tasks on an existing resource group based on **All supported resource types**, be aware that the actions might be run on many more resources than were in the group when you first created it. This might also mean that automations or tasks that you created for

other resources are applied to unintended resources, or resources on which the tasks cannot be completed.



Service	Resources
Amazon API Gateway	• AWS::ApiGateway::Stage
AWS Certificate Manager	• AWS::CertificateManager::Certificate
Amazon CloudFront	• AWS::CloudFront::Distribution • AWS::CloudFront::StreamingDistribution
AWS CloudTrail	• AWS::CloudTrail::Trail
Amazon CloudWatch Logs	• AWS::Logs::LogGroup
AWS Database Migration Service	• AWS::DMS::Endpoint • AWS::DMS::ReplicationInstance
AWS Data Pipeline	• AWS::DataPipeline::Pipeline
Amazon DynamoDB	• AWS::DynamoDB::Table
Amazon Elastic Compute Cloud (Amazon EC2)	• AWS::EC2::CustomerGateway • AWS::EC2::DHCPOptions • AWS::EC2::EIP • AWS::EC2::Instance • AWS::EC2::InternetGateway • AWS::EC2::LaunchTemplate • AWS::EC2::NatGateway • AWS::EC2::NetworkAcl • AWS::EC2::NetworkInterface • AWS::EC2::RouteTable • AWS::EC2::SecurityGroup • AWS::EC2::Snapshot • AWS::EC2::Subnet

Service	Resources
	• AWS::EC2::VPCPeeringConnection • AWS::EC2::VPC • AWS::EC2::VPNConnection • AWS::EC2::VPNGateway • AWS::EC2::Volume
AWS Elastic Beanstalk	• AWS::ElasticBeanstalk::Environment
Amazon ElastiCache	• AWS::ElastiCache::CacheCluster • AWS::ElastiCache::Snapshot
Amazon Elastic File System	• AWS::EFS::FileSystem
Elastic Load Balancing	• AWS::ElasticLoadBalancing::LoadBalancer • AWS::ElasticLoadBalancingV2::LoadBalancer • AWS::ElasticLoadBalancingV2::TargetGroup
Amazon EMR	• AWS::EMR::Cluster
Amazon Elasticsearch Service	• AWS::Elasticsearch::Domain
Amazon Inspector	• AWS::Inspector::AssessmentTemplate
AWS Key Management Service	• AWS::KMS::Key
Amazon Kinesis	• AWS::Kinesis::Stream
AWS Lambda	• AWS::Lambda::Function
AWS OpsWorks	• AWS::OpsWorks::Instance • AWS::OpsWorks::Layer • AWS::OpsWorks::Stack
Amazon Relational Database Service (Amazon RDS)	• AWS::RDS::DBClusterParameterGroup • AWS::RDS::DBCluster • AWS::RDS::DBInstance • AWS::RDS::DBParameterGroup • AWS::RDS::DBSecurityGroup • AWS::RDS::DBSnapshot • AWS::RDS::DBSubnetGroup • AWS::RDS::OptionGroup
Amazon Redshift	• AWS::Redshift::ClusterParameterGroup • AWS::Redshift::ClusterSecurityGroup • AWS::Redshift::ClusterSubnetGroup • AWS::Redshift::Cluster
AWS Resource Groups	• AWS::ResourceGroups::Group (Lets you create a *nested* resource group, or a resource group that contains other resource groups)

Service	Resources
Amazon Route 53	(Supported only in the US East (N. Virginia) Region, `us-east-1`.) • AWS::Route53::HealthCheck • AWS::Route53::HostedZone
Amazon SageMaker	• AWS::SageMaker::EndpointConfig • AWS::SageMaker::Endpoint • AWS::SageMaker::HyperParameterTuningJob • AWS::SageMaker::Model • AWS::SageMaker::NotebookInstance • AWS::SageMaker::TrainingJob
AWS Service Catalog	• AWS::ServiceCatalog::CloudFormationProduct • AWS::ServiceCatalog::Portfolio
Amazon Simple Queue Service	• AWS::SQS::Queue
Amazon Simple Storage Service (Amazon S3)	• AWS::S3::Bucket
AWS Systems Manager	• AWS::SSM::Document • AWS::SSM::MaintenanceWindow • AWS::SSM::ManagedInstance • AWS::SSM::Parameter • AWS::SSM::PatchBaseline
AWS Storage Gateway	• AWS::StorageGateway::Gateway • AWS::StorageGateway::Volume
Amazon WorkSpaces	• AWS::WorkSpaces::Workspace

Supported Resources for Tag Editor Tagging

You can use Tag Editor in the AWS Management Console to tag the following AWS resources. For more information, see Working with Tag Editor (p. 21).

Service	Resources
Amazon ElastiCache	• Cache cluster • Snapshot
Amazon Elastic Compute Cloud (Amazon EC2)	• AMI • Instance • Network interface • Reserved Instance • Security group • Snapshot • Spot Instance request • Volume

Service	Resources
Elastic Load Balancing	• Classic load balancer (CLB)
Amazon EMR	• Cluster
Amazon Glacier	• Vault
Kinesis	• Stream
Amazon Relational Database Service (Amazon RDS)	• Database instance • Database option group • Database parameter group • Database security group • Database snapshot • Database subnet group • Event subscription • Reserved database instance
Amazon Redshift	• Cluster • Hardware security module (HSM) client certificate • HSM connection • Parameter group • Snapshot • Subnet group
Amazon Route 53	• Domain • Health check • Hosted zone
Amazon Simple Storage Service (Amazon S3)	• Bucket
AWS Storage Gateway	• Gateway
Amazon Virtual Private Cloud	• Customer gateway • DHCP option set • Internet gateway • Network access conrol list (ACL) • Route table • Subnet • Virtual private gateway • VPC • VPN connection

Getting Started with AWS Resource Groups

In AWS, a *resource* is an entity that you can work with. Examples include an Amazon EC2 instance, an Amazon S3 bucket, or an Amazon Route 53 hosted zone. If you work with multiple resources, you might find it useful to manage them as a group rather than move from one AWS service to another for each task.

This section shows you how to get started with AWS Resource Groups. First, organize AWS resources by tagging them in Tag Editor. Then build queries in Resource Groups that include the resource types you want in a group, and tags that you've applied to resources.

After you've created resource groups in Resource Groups, use AWS Systems Manager tools such as Automation to simplify management tasks on your groups of resources. You can also use groups as the basis for viewing monitoring and configuration insights in AWS Systems Manager.

For more information about getting started with AWS Systems Manager features and tools, see the *AWS Systems Manager User Guide*.

Topics

Prerequisites for Working with AWS Resource Groups

Before you get started working with resource groups, be sure you have an active AWS account with existing resources and appropriate rights to tag resources and create groups.

Topics

Sign Up for AWS

If you do not have an AWS account, use the following procedure to create one.

To sign up for AWS

1. Open https://aws.amazon.com/, and then choose **Create an AWS Account**.

   Note
   If you previously signed in to the AWS Management Console using AWS account root user credentials, choose **Sign in to a different account**. If you previously signed in to the console using IAM credentials, choose **Sign-in using root account credentials**. Then choose **Create a new AWS account**.
2. Follow the online instructions.

   Part of the sign-up procedure involves receiving a phone call and entering a verification code using the phone keypad.

Create Resources

You can create an empty resource group, but won't be able to see insights or perform any tasks on resource group members until there are resources in the group. For more information about the supported resource types, see Supported Resources (p. 4).

Set Up Permissions

To make full use of Resource Groups and Tag Editor, you might need additional permissions to tag resources or to see a resource's tag keys and values. These permissions fall into the following categories:

- Permissions for individual services so that you can tag resources from those services and include them in resource groups.
- Permissions that are are required to use the Tag Editor console and API.
- Permissions that are required to use the new AWS Resource Groups console and API.

> **Note**
> The managed policies that were used for legacy Resource Groups, `ResourceGroupsandTagEditorFullAccess` and `ResourceGroupsandTagEditorReadOnlyAccess`, do not grant access to AWS Resource Groups.

If you are an administrator, you can provide permissions for your users by creating policies through the AWS Identity and Access Management (IAM) service. You first create IAM users or groups, and then apply the policies with the permissions that they need. For information about creating and attaching IAM policies, see Working with Policies.

Permissions for Individual Services

> **Important**
> This section describes permissions that are required if you want to tag resources from other service consoles and APIs, and add those resources to resource groups.

As described in What Are Resource Groups? (p. 2), each resource group represents a collection of resources of specified types that share one or more tag keys or values. To add tags to 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, such as those listed in the Amazon EC2 User Guide.

To make full use of the Resource Groups feature, you need other permissions that allow you to access a service's console and interact with the resources there. For examples of such policies for Amazon EC2, see Example Policies for Working in the Amazon EC2 Console in the *Amazon EC2 User Guide for Linux Instances*.

Granting Permissions for Using Tag Editor

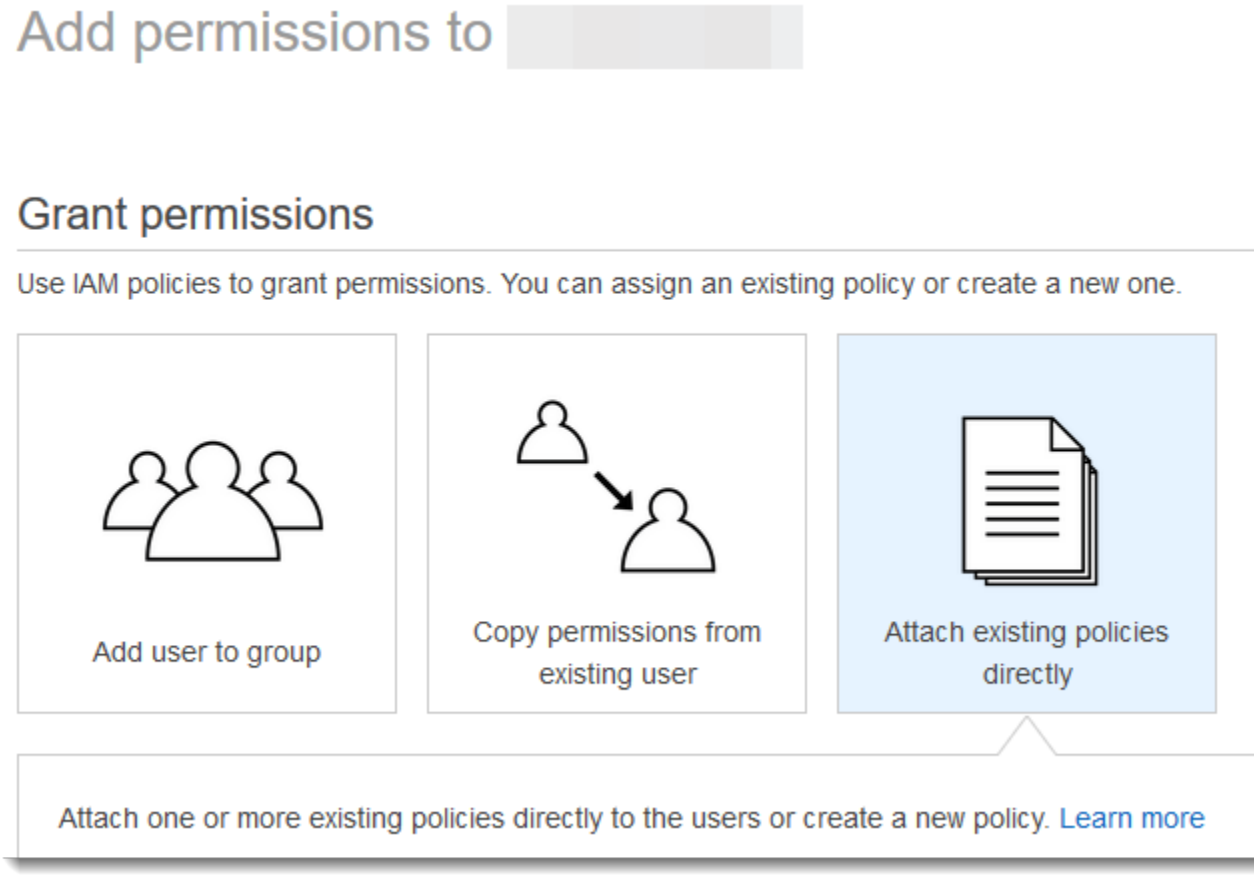
For information about how to grant permissions for Tag Editor and the legacy Resource Groups console, see Obtaining Permissions for Resource Groups and Tag Editor in the AWS Management Console Help. Permissions shown in this topic are for using the new AWS Resource Groups service.

Granting Permissions for Using AWS Resource Groups

This section describes required permissions for the new AWS Resource Groups service.
For information about how to assign permissions for using legacy Resource Groups, see Obtaining Permissions for Resource Groups and Tag Editor. The managed policies that were used for legacy Resource Groups, `ResourceGroupsandTagEditorFullAccess` and `ResourceGroupsandTagEditorReadOnlyAccess`, do not grant access to AWS Resource Groups.

To add a policy for using AWS Resource Groups to a user, do the following.

1. Open the IAM console.
2. In the navigation pane, choose **Users**.
3. Find the user to whom you want to grant AWS Resource Groups permissions. Choose the user's name to open the user properties page.
4. Choose **Add permissions**.
5. Choose **Attach existing policies directly**.



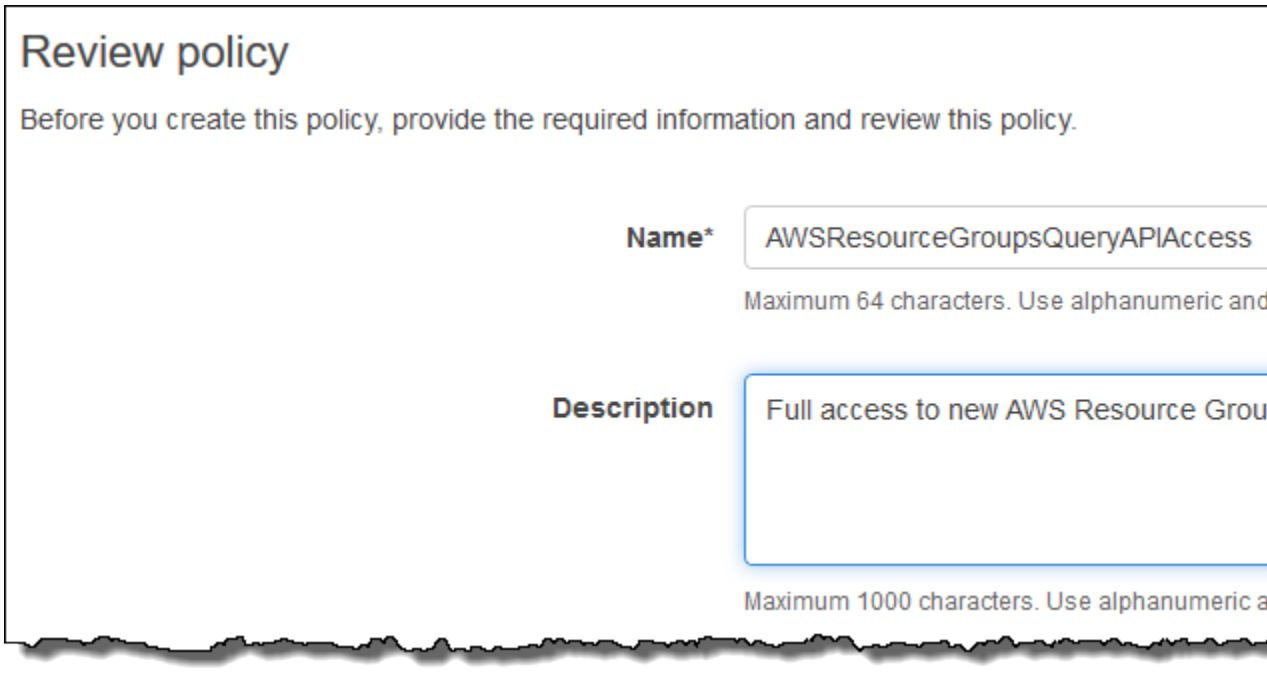
6. Choose **Create policy**.
7. On the **JSON** tab, paste the following policy statement.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "ec2:DescribeInstances",
        "ec2:DescribeSecurityGroups",
        "ec2:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeVpcs",
        "elasticache:DescribeCacheClusters",
        "elasticache:DescribeSnapshots",
        "elasticache:ListTagsForResource",
        "elasticloadbalancing:DescribeLoadBalancers",
        "elasticloadbalancing:DescribeTags",
        "elasticmapreduce:DescribeCluster",
        "elasticmapreduce:ListClusters",
        "kinesis:DescribeStream",
        "kinesis:ListStreams",
        "kinesis:ListTagsForStream",
        "opsworks:DescribeStacks",
        "opsworks:ListTags",
        "rds:DescribeDBInstances",
        "rds:DescribeDBSnapshots",
        "rds:ListTagsForResource",
```

```
        "redshift:DescribeClusters",
        "redshift:DescribeTags",
        "resource-groups:*",
        "route53:GetHealthCheck",
        "route53:GetHostedZone",
        "route53:ListHealthChecks",
        "route53:ListHostedZones",
        "route53:ListTagsForResource",
        "route53domains:ListDomains",
        "s3:GetBucketTaggingConfiguration",
        "s3:ListBuckets",
        "storagegateway:DescribeGatewayInformation",
        "storagegateway:ListGateways",
        "storagegateway:ListTagsForResource",
        "tag:GetResources"
      ],
      "Resource": "*"
    }
  ]
}
```

Note
This policy statement grants permissions only for AWS Resource Groups actions. It does not allow access to AWS Systems Manager tasks in the AWS Resource Groups console. For example, this policy does not grant permissions for you to use Systems Manager Automation commands. To perform Systems Manager tasks on resource groups, you must have Systems Manager permissions attached to your policy (such as `ssm:*`). For more information about granting access to Systems Manager, see Configuring Access to Systems Manager in the *Systems Manager User Guide*.

8. Choose **Review policy**.
9. Give the new policy a name and description. To distinguish this policy from any policies for legacy Resource Groups, the name should be different from `ResourceGroupsandTagEditorFullAccess` (for example, `AWSResourceGroupsQueryAPIAccess`).



10. Choose **Create policy**.

11. Now that the policy is saved in IAM, you can attach it to other users. For more information about how to add a policy to a user, see Adding Permissions by Attaching Policies Directly to the User in the *IAM User Guide*.

Finding Resources in AWS Resource Groups

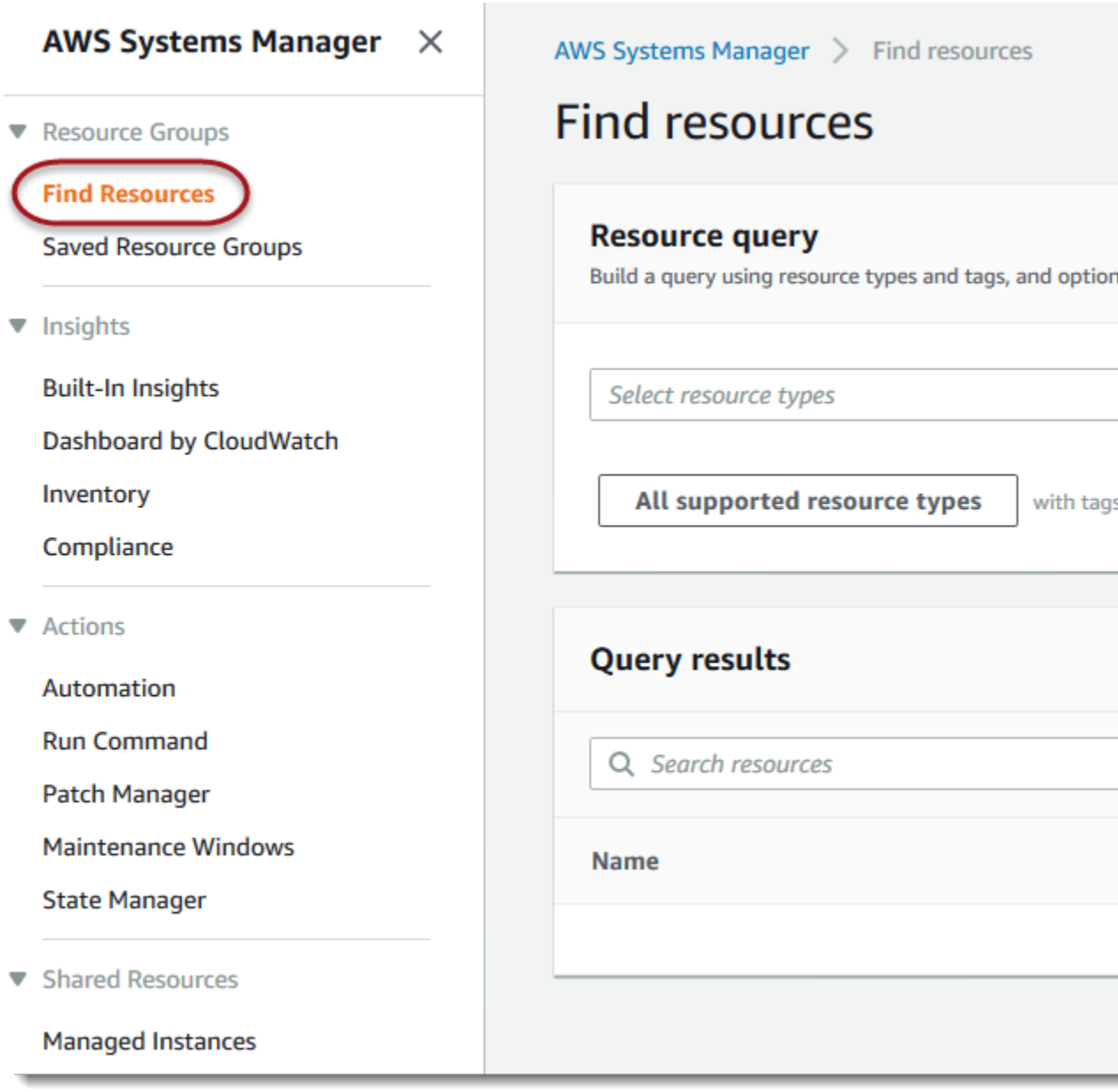
You can use Resource Groups to search or query for resources in the current region within your account, and optionally create a group of resources that are returned as search results. You can also run the AWS CLI command **search-resources** to find resources of specific types that have specific tags. To search for resources and create a group in a single AWS CLI command, run **create-group**.

Find Resources Using the AWS Management Console

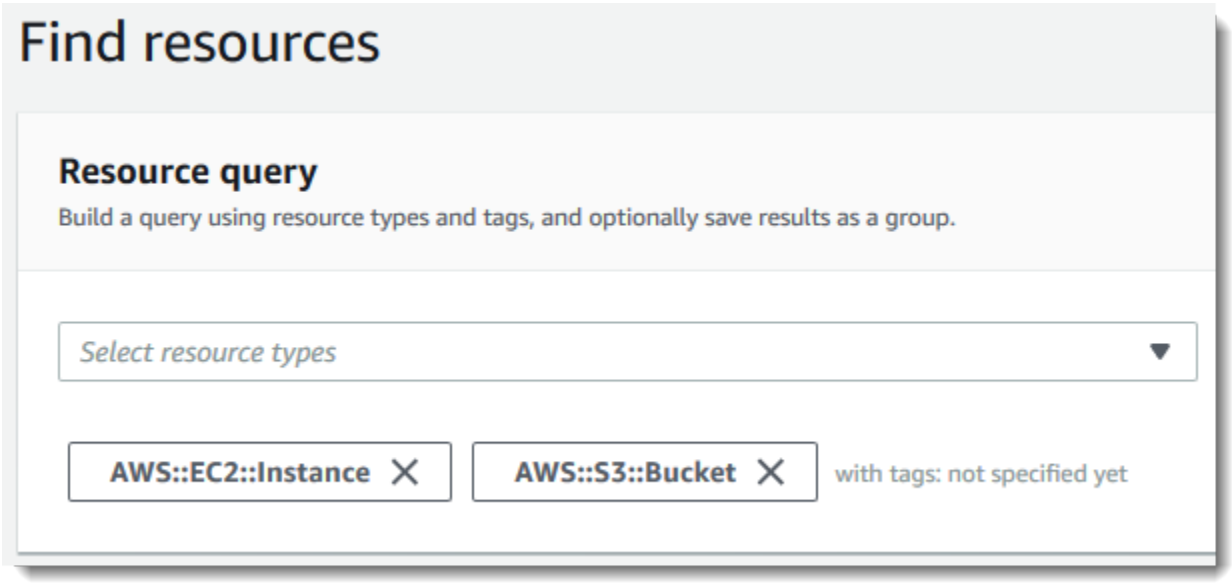
The **Find resources** page lets you build queries by selecting resource types, and specifying tags that are shared by resources of those types.

To build a query of resources in Resource Groups in the AWS Management Console

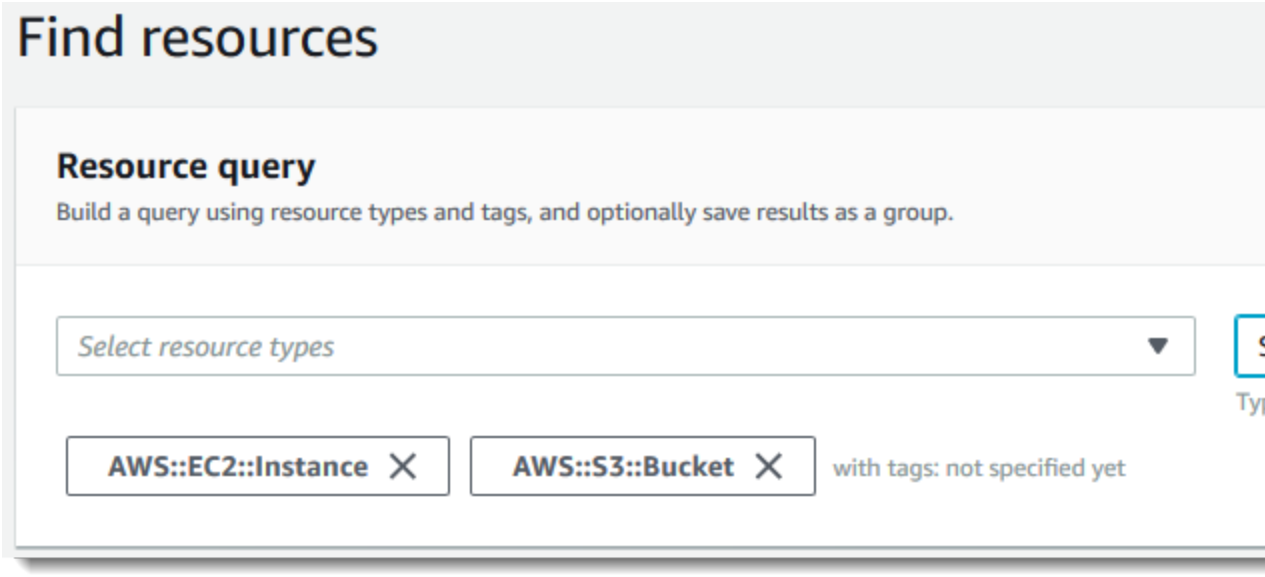
1. Open Resource Groups from the AWS Systems Manager console or from the top left of the AWS Management Console.
2. In the navigation pane, choose **Find resources**.



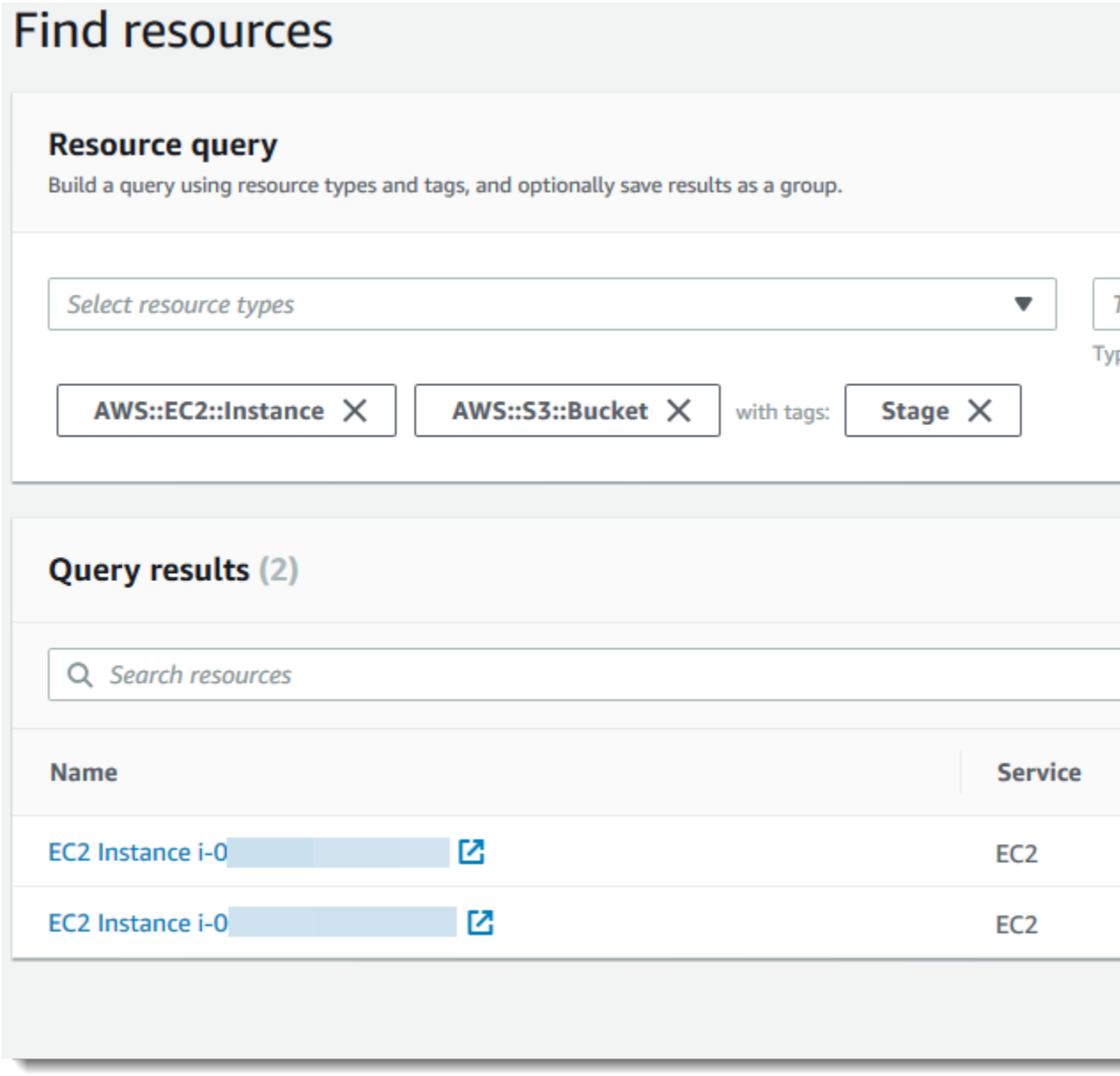
3. In the **Resource query** area, choose resource types in the **Select resource types** drop-down list. You can have a maximum of 20 resource types in a query. For this walkthrough, choose **AWS::EC2::Instance** and **AWS::S3::Bucket**.



4. In the tag string boxes, specify a tag key, or a tag key and value pair, to limit the EC2 instances and S3 buckets in the current region to only those that are tagged with your specified values. Choose **+** or press **Enter** when you've finished your tag. In this example, filter for resources that have a tag key of **Stage**. The tag value is optional, but narrows the results of the query further. To add more tags, choose **+**. Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query.



5. Choose **View query results** to return the list of EC2 instances and S3 buckets in the current region that match the specified tag key or keys.



If you only wanted to find resources by building a query, you have finished the procedure. To create a resource group based on your query, choose **Save query as group**, and go on to the next step.

6. On the **Create query-based group** page, in the **Group details** area, specify a group name, and optionally, add a description.

   a. In the **Group name** box, type a name for your resource group.

      A resource group name can have a maximum of 127 characters, including letters, numbers, hyphens, dots, and underscores. The name cannot start with `AWS` or `aws`. These are reserved. A resource group name must be unique in the current region in your account.

   b. (Optional) In the **Group description** box, type a description of your group.

7. (Optional) In the **Group tags** area, add tag key and value pairs that apply only to the resource group, not the member resources in the group.

Group tags are useful if you plan to make this group a member of a larger group. Because specifying at least a tag key is required to create a group, be sure to add at least a tag key in **Group tags** to groups that you plan to nest into larger groups.

8. To create your group, choose **Create group**.

Find Resources Using the AWS CLI

The procedure in this section builds queries by using the **search-resources** command to find resources of specific types that have specific tags. To search for resources and create a group in a single AWS CLI command, run **create-group**. For more information about how to build a query and create a group in a single command, see Build Queries and Groups in AWS Resource Groups (p. 18).

Note
The `SearchResources` API currently does not support finding untagged resources. You can use it only to find resources within a single region that are already tagged with at least one tag key.

To build a query and find resources in Resource Groups (AWS CLI)

1. In an AWS CLI session, type the following, and then press **Enter**, replacing the values for resource query type, resource type filters, tag keys, and tag values with your own. The `Type` attribute of `--resource-query` currently has one valid value, `TAG_FILTERS_1_0`. Both `ResourceTypeFilters` and `TagFilters` are required in the `Query` attribute. To search for all resource types that have a specific tag, specify `[\"AWS::AllSupported\"]` as the value of `ResourceTypeFilters`. You can have a maximum of 20 resource types in a query. At least one tag key is required, but specifying tag values is optional.

```
aws resource-groups search-resources --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"resource_type1\",
\"resource_type2\"],\"TagFilters\":[{\"Key\":\"Key1\",\"Values\":[\"Value1\",
\"Value2\"]},{\"Key\":\"Key2\",\"Values\":[\"Value1\",\"Value2\"]}]}"}'
```

   The following command finds all EC2 instances in the current region within your account that have a tag key of `Stage` and a value of `Test`.

```
aws resource-groups search-resources --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],
\"TagFilters\":[{\"Key\":\"Stage\",\"Values\":[\"Test\"]}]}"}'
```

   The following command is an example that does not filter for any specific resource types (all resource types are included in the query), and that specifies only a tag key of `Stage`, but no tag values.

```
aws resource-groups search-resources --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"AWS::AllSupported\"],
\"TagFilters\":[{\"Key\":\"Stage\"}]}"}'
```

2. The following are returned in the response to the command.

   - The ARNs of each resource that the query returns.
   - The type of each resource that the query returns.

   The results of the **search-resources** command look similar to the following example.

```
HTTP/1.1 200
```

```
Content-type: application/json

{
   "NextToken": "NextToken value, if provided to paginate results",
   "ResourceIdentifiers": [
      {
         "ResourceArn": "arn:aws:ec2:us-west-2:123456789012:instance/*",
         "ResourceType": "AWS::EC2::Instance"
      },
      {
         "ResourceArn": "arn:aws:ec2:us-west-2:123456789000:instance/*",
         "ResourceType": "AWS::EC2::Instance"
      }
   ]
}
```

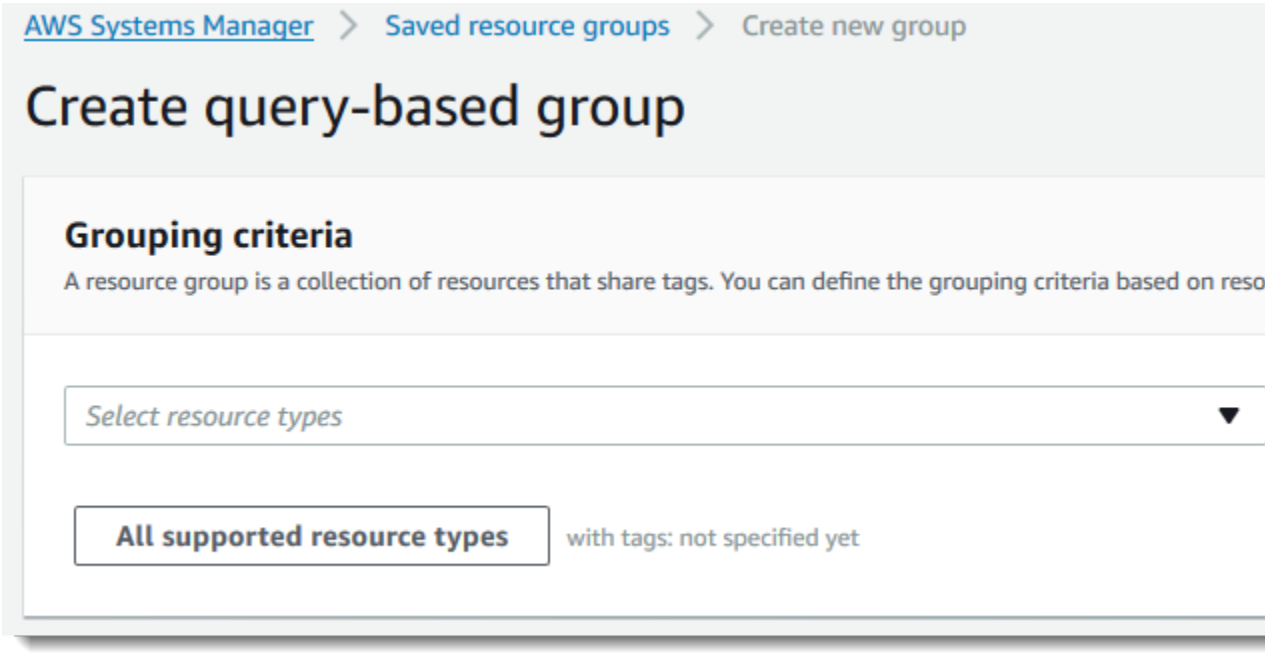
Build Queries and Groups in AWS Resource Groups

In AWS Resource Groups, a *query* is the foundation of a group. To build a query, choose the types of resources that you want to be part of the group, and then specify the tags that are shared by the resources that you want to be members of the group. For example, if you want to create a resource group that has all of the Amazon EC2 instances and Amazon S3 buckets that you are using to run the testing stage of an application, and you have instances and buckets that are tagged this way, choose the `AWS::EC2::Instance` and `AWS::S3::Bucket` resource types from the drop-down list, and then specify the tag key **`Stage`**, with a tag value of **`Test`**.

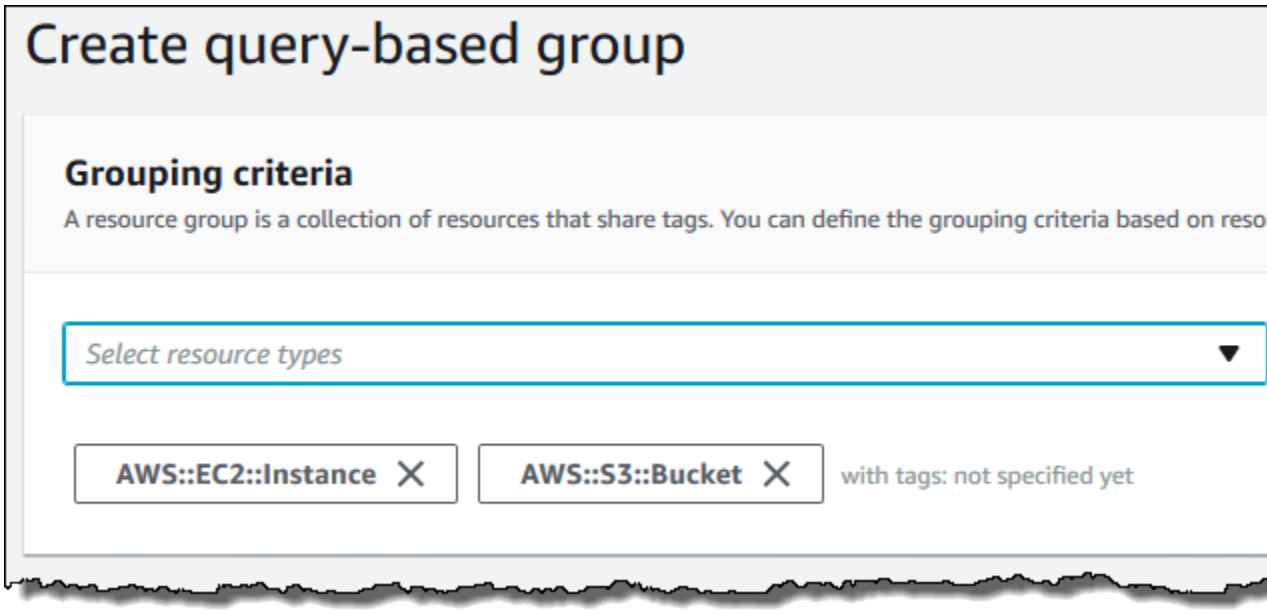
In the AWS CLI, you build a query and create your resource group in the same command. The AWS CLI command shown in this topic creates a group.

To build a query and create a group in Resource Groups (AWS Management Console)

1. Open Resource Groups from the AWS Systems Manager console or from the top left of the AWS Management Console.
2. Choose **Create a resource group**.

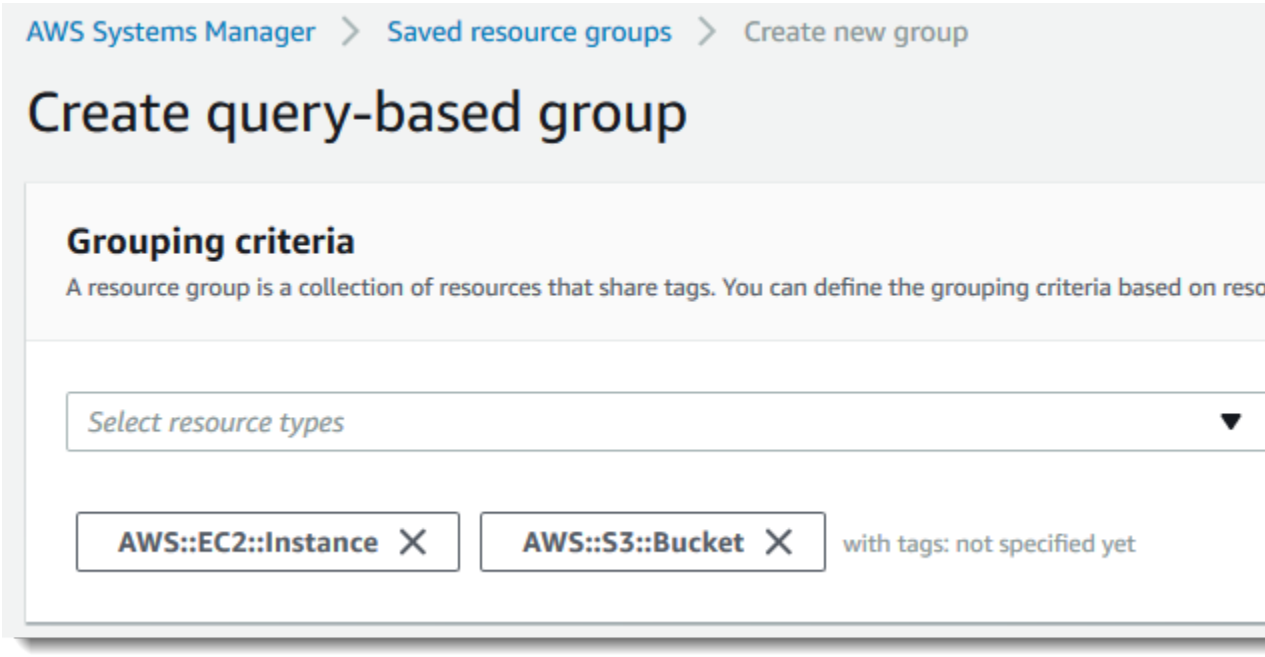


3. On the **Create a resource group** page, choose the resource types that you want to be in your resource group. You can have a maximum of 20 resource types in a query. For the purposes of this walkthrough, choose **AWS::EC2::Instance** and **AWS::S3::Bucket**.



4. In the tag string boxes, specify a tag key, or a tag key and value pair, to limit the EC2 instances and S3 buckets in your account to only those that are tagged with your specified values. Choose **+** or press **Enter** when you've finished your tag. In this example, filter for resources that have a tag key of **Stage**. The tag value is optional, but narrows the results of the query further. To add more tags, choose **+**.

   Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query. If you specify multiple values in your query for a tag key, the query returns resources that match *any* of the values of the tag key. However, when there is more than one tag key, the query returns resources that match *all* tag keys, and at least one value of each specified tag key.



5. Choose **View group resources** to return the list of EC2 instances and S3 buckets in your account that match the specified tag key or keys.
6. To create a resource group based on your query, specify a name, and optionally, add a description.

   a. In the **Group name** box, type a name for your resource group.

      A resource group name can have a maximum of 127 characters, including letters, numbers, hyphens, dots, and underscores. The name cannot start with `AWS` or `aws`. These are reserved. A resource group name must be unique in the current region in your account.

   b. (Optional) In the **Group description** box, type a description of your group.

   c. (Optional) In the **Group tags** area, add tag key and value pairs that apply only to the resource group, not the member resources in the group.

      Group tags are useful if you plan to make this group a member of a larger group. Because specifying at least a tag key is required to create a group, be sure to add at least a tag key in **Group tags** to groups that you plan to nest into larger groups.
7. When you are finished, choose **Create group**.

To create a group in Resource Groups (AWS CLI)

In an AWS CLI command, you build a query and create a resource group based on the query in a single command.

1. In an AWS CLI session, type the following, and then press **Enter**, replacing the values for group name, resource types, tag keys, and tag values with your own. You can have a maximum of 20 resource types in a query. A resource group name can have a maximum of 127 characters, including letters, numbers, hyphens, dots, and underscores. The name cannot start with `AWS` or `aws`. These are reserved. A resource group name must be unique in your account. For more information about the syntax of the members of the `ResourceQuery` API, see ResourceQuery in the *AWS Resource Groups API Reference*.

   Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query. If you specify multiple values in your query for a tag

key, the query returns resources that match *any* of the values of the tag key. However, when there is more than one tag key, the query returns resources that match *all* tag keys, and at least one value of each specified tag key.

```
aws resource-groups create-group --name resource-group-name --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"resource_type1\",
\"resource_type2\"],\"TagFilters\":[{\"Key\":\"Key1\",\"Values\":[\"Value1\",
\"Value2\"]},{\"Key\":\"Key2\",\"Values\":[\"Value1\",\"Value2\"]}]}"}'
```

The following command is an example.

```
aws resource-groups create-group --name my-resource-group --resource-query
 '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"AWS::EC2::Instance\"],
\"TagFilters\":[{\"Key\":\"Stage\",\"Values\":[\"Test\"]}]}"}'
```

2. The following are returned in the response to the command.

   - A full description of the group you have created.
   - The resource query that you used to create the group.
   - The tags that are associated with the group.

After you have created a resource group, you can use the group to view insights about (p. 31) or perform tasks on the resources in the group.

Working with Tag Editor

Tags are words or phrases that act as metadata for identifying and organizing your AWS resources. The tag limit varies with the resource, but most can have up to 50 tags. Each tag consists of a key and a value. For more about tagging, see Using Cost Allocation Tags in the *AWS Billing and Cost Management User Guide*.

You can add tags to resources when you create the resource or add, change, or remove those tags one resource at a time within each resource's console. To add to multiple resources at once, you need to use Tag Editor. With Tag Editor, you search for the resources that you want to tag, and then add, remove, or edit tags for the resources in your search results.

To start Tag Editor

1. Sign in to the AWS Management Console.
2. On the navigation bar, choose **Resource Groups**, and then choose **Tag Editor**.

Not all resources can have tags applied. To see if a resource supports tagging, consult the documentation for that resource's service.

For information about permissions and roles that are required to tag resources, see Set Up Permissions (p. 9).

Topics

- Scenario: Implementing a New Tagging Strategy (p. 25)

Searching for Resources to Tag

With Tag Editor, you can locate all the resources that are available for tagging. For more information, see Supported Resources for Tag Editor Tagging (p. 7).

To search for resources to tag

1. Sign in to the AWS Management Console, choose **Resource Groups**, and then choose **Tag Editor**.
2. For **Regions**, choose the regions that you want to search in. Repeat for as many regions as you want. To remove a region, choose the **x** by its name.
3. For **Resource types**, choose the kind of resources that you want to locate. Repeat for as many resource types as you want. To remove a region or resource type, choose the **x** by its name. To search for all resource types or all regions, select **All resource types**.
4. (Optional) To limit your search to resources that already have certain tag keys or values, in the first **Tags** box, choose the name of a tag key. Type in the box to search for a key based on characters that it contains.
5. (Optional) In the next **Tags** box, do any of the following:

   - Leave the box empty to search for all resources with the specified key and any value.
   - Select **Not tagged** to search for resources that do *not* have the specified tag key.
   - Select **Empty value** to search for resources that have the specified tag key, but no value.
   - Type one or more characters to find resources with the values that you are looking for. Select a value from the list to find an exact match or select the **Contains:** option to find values that contain the characters that you typed.

     If you don't see any values listed, you might not have permissions to view available tags. In that case, you can simply type in a complete value and press **Enter** to start searching.
   - Choose the **x** next to an item that you added to remove it from the search criteria.

   You can add multiple values for each tag key. Doing so potentially increases the number of resources in the search results because the results include resources tagged with any of the selected values. The search is case sensitive.

   Note
   Before a key and its values appear in this list, they must have been applied to at least one resource in the current account. If you don't see a tag that you just applied to a resource, try refreshing your browser window.
6. (Optional) To further refine your group, continue using the **Tags** boxes at the bottom to specify more tag keys and values. The search results contain only those resources that have all the specified tags, so the more tags you specify, the fewer resources Tag Editor finds.
7. When you have the settings that you want, choose **Find resources**.

Finding Untagged Resources

You might find it useful to know which resources in your account have yet to be tagged. You might also want to know which resources have tag keys, but no tag values. You can use Tag Editor to find these resources.

1. Sign into the AWS Management Console and open Tag Editor at https://resources.console.aws.amazon.com/r/tags.
2. For **Regions**, select the regions that you want to include.
3. For **Resource types**, select the resource types that you want to search for.
4. For **Tags**, select a tag key that you want to apply to resources that do not already have it.
5. In the next box, do either or both of the following, depending on your goal:
   - Choose **Not tagged** to find resources with no specified tags.
   - Choose **Empty value** to find resources that are tagged with your specified key, but that have no tag value.
6. Choose **Find Resources** to have Tag Editor find all such resources and list them at the bottom of the page.
7. (Optional) Use any of the methods described in Tagging Resources (p. 24) to add tags to the resources in the search results.

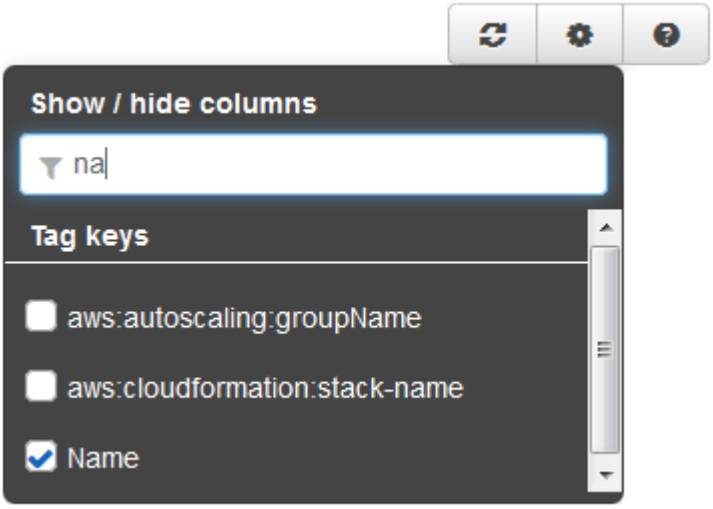
Customizing Tag Search Results

You can sort and filter the results of your tag search to find the tags and resources that you need to work with.

To customize tag search results

Do any of the following:

- To limit the display to resources that contain a keyword, type the keyword in the filter box above the table. For example, you could type **`elasticbeanstalk`** to see all resources whose ID indicates that they are associated with the Elastic Beanstalk service.
- To sort the list by any column, choose the column heading name. To reverse the sort order, choose the name again.
- To change the width of a table column, drag the divider between the column headings.
- To add or remove columns of existing tag keys to the results, choose the cog icon () above the table and select or clear a tag key. To search for and limit the items that appear in the list, type a full or partial keyword or words in the box at the top.



Tagging Resources

After you have located the resources (p. 22) that you want to tag, you can add, remove, or edit the tags for all or some of your search results as a group.

To apply a new tag to one or more resources, you first create a tag key.

To create a tag key

1. Search for the resources (p. 22) whose tags you want to manage.
2. Choose **Create a new tag key**.
3. Type the name of your new key, and then choose **Add key**. Repeat for as many tag keys as you want to create.

   Note
   The new tag key does not actually exist until a value for the new key is applied to a resource. A tag does not exist unless it is applied to a resource.

To edit or apply tags for a single resource

1. Search for the resources (p. 22) whose tags you want to manage.
2. Customize the search results (p. 23) table to make it easy to find the resource that you want.
3. Do any of the following:

   - To add a tag to a resource, choose the **+** () icon, type a value, and choose the check mark icon (). You can apply a tag to a resource without specifying a value.

   - To edit the tag value of a resource, choose the pencil icon () by its value and edit the value. Choose the check mark icon or press **Enter**.

- To remove an existing tag value, choose the **X** icon () next to the value.

- To apply an existing tag key, choose the cog icon () and select the key that you want, as described in Customizing Tag Search Results (p. 23). Then choose the **+** icon in the column for that key, type a new value, and either choose the check mark icon or press **Enter**.

- To modify a resource's tag in its own console, choose the blue icon (). Then use the settings in that console to edit or apply tags.

To edit or apply tags for multiple resources

1. Search for the resources (p. 22) whose tags you want to manage.
2. Customize the search results (p. 23) table to make it easy to find the resource that you want.
3. Select the check box for each resource whose tags you want to modify. To manage tags for all the resources in the list, select the check box in the column heading row.
4. Choose **Edit tags for selected**. Modify the keys or values, and then choose **Apply changes**.
5. Repeat the preceding step as needed.

Scenario: Implementing a New Tagging Strategy

Consider a situation where you have a medium to large working environment with multiple resources used by various employees. You decide to use tagging to help you organize and get better oversight of your account's resources. But how to proceed when there are dozens of resources to tag? Fortunately, Tag Editor can simplify the process.

1. **Make a plan.**

   Before you begin, sketch out a plan of the tag keys and values that will help you organize your resources. For example, you might want all resources to have tag keys like *Project*, *Cost Center*, and *Environment*. Remember, too, that each resource cannot have more than 50 tags.

2. **Open Tag Editor.**

   Sign into the AWS Management Console and open Tag Editor at https://resources.console.aws.amazon.com/r/tags.

3. **Find all resources in your account.**

   For **Regions**, select all regions that apply. For **Resource types**, select **All resource types**. Leave both **Tags** boxes empty. Then choose **Find resources**. For more information, see Searching for Resources to Tag (p. 22).

4. **Select all the found resources.**

   The Tag Editor search results appear at the bottom of the page. When the list shows the resources that you want to tag, select the top check box to select all resources. Choose **Edit tags for selected**.

5. **Apply tag keys with empty elements.**

   In **Add/edit tags**, under **Add tags**, in the space provided, type the key name that you want to add, such as **`Project`**. Repeat for your other new keys, such as **`Cost Center`** and **`Environment`**. Choose **Apply changes**.

   Tip
   If any of your selected resources have reached the maximum of 50 tags, a message warns you before you choose **Apply changes**. You can pause the pointer over the number of affected resources in the message to see a pop-up list of the specific resources.

6. **Add values for each tag key.**

   The next step is to add tag values that will help you distinguish individual resources that share tag keys. There are a couple of ways to do this depending on whether you plan on adding the same values to many resources or just a few.

   a. **Bulk add values.**

      Start by selecting the check box at the top of the table again to clear all the check boxes. Then select individual check boxes for just those resources that need a specific tag value.

      Choose **Edit tags for selected**. In **Add/edit tags**, under **Applied tags**, type a new value in the **Value** column next to a tag key. For example, you might add a billing code in the value for the Cost Center key or type **`Production`** for the Environment key.

      If the **Value** column shows **Multiple values**, you can still type in a new value. However, your new value will replace all the key's existing values for the selected resources.

      When you're done, choose **Apply changes**.

   b. **Add individual tag values.**

      If you want each resource to have its own unique value, you can edit tag values right in the search results table. Start by choosing the cog icon above the table and selecting the check boxes for your new keys. To continue our example, you might select the check boxes for **Project**, **Cost Center**, and **Environment**. This makes your keys appear as columns in the search results table.

      For a given resource, locate the column that displays the tag key whose value you want to edit. Choose the pencil icon, and then type the new value in the box. Press **Enter** to complete the editing.

7. Repeat Step 6 for other resources in your list.

Update Groups in AWS Resource Groups

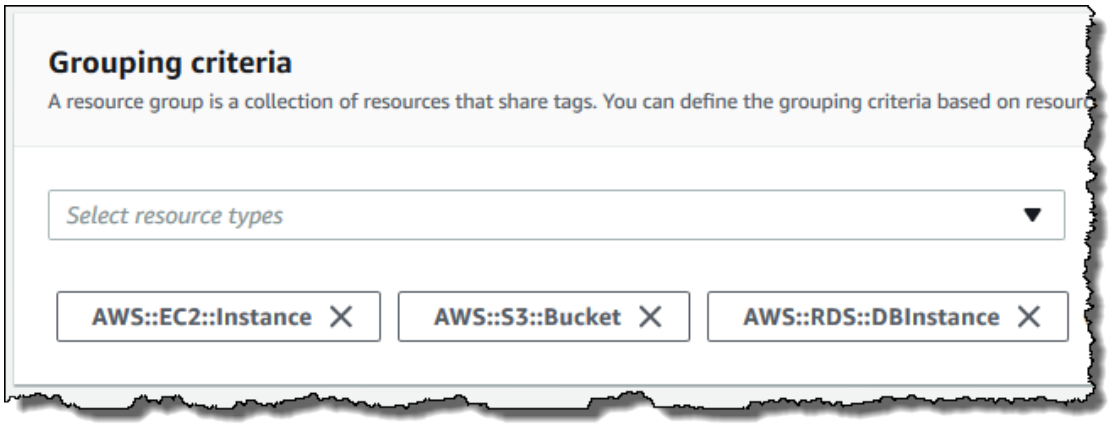
To update a resource group in Resource Groups, you can edit the query and tags that are the basis of your group. You can add and remove resources from your group only by applying changes to the query or tags. You cannot select specific resources to add to or remove from your group. The best way to add or remove a specific resource from a group is to edit the resource's tags, then verify that your resource group tag query either includes or omits the tag, depending on whether you want the resource in your group.

In the AWS CLI, you update groups in two commands;

- `update-group`, which you run to update a group's description.
- `update-group-query`, which you run to update the resource query and tags that determine the group's member resources.

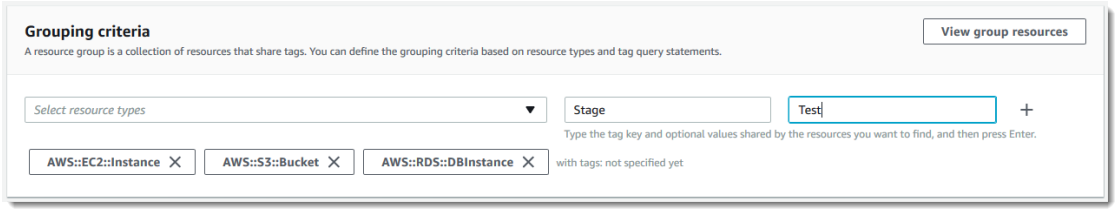
To update a query and group in Resource Groups (AWS Management Console)

1. Open Resource Groups from the AWS Systems Manager console, or from the top left of the AWS Management Console.
2. Under **Saved resource groups** in the navigation pane, choose an existing group, and then choose **Edit**.
3. On the **Edit group** page, in the **Grouping criteria** area, add or remove resource types as desired. You can have a maximum of 20 resource types in a query. To remove a resource type, choose **X** on the resource type's label. Choose **View group resources** to see how the changes affect your group's resource members. In this walkthrough, we add the resource type **AWS::RDS::DBInstance** to the query.

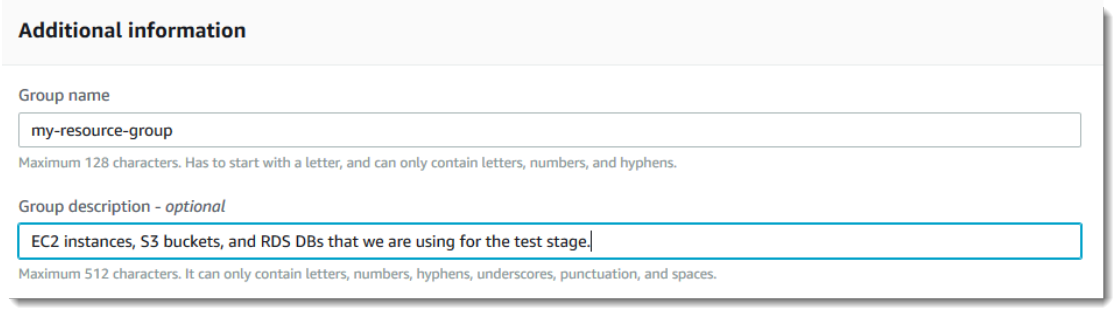


4. Edit tags, if necessary. In this example, we filter for resources that have a tag key of **Stage** and add a tag value of **Test**. The tag value is optional, but narrows the results of the query further.

   Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query. If you specify multiple values in your query for a tag key, the query returns resources that match *any* of the values of the tag key. However, when there is more than one tag key, the query returns resources that match *all* tag keys, and at least one value of each specified tag key.



5. In the **Additional information** area, you can edit the group description. You cannot edit an existing group's name.



6. In the **Group tags** area, add or remove tags as desired. Group tags are metadata about your resource group. They do not affect member resources. To change the resources that are returned by the resource group's query, edit tags in the **Grouping criteria** area.

   Group tags are useful if you plan to make this group a member of a larger group. Because specifying at least a tag key is required to create a group, be sure to add at least a tag key in **Group tags** to groups that you plan to nest into larger groups.
7. Choose **View query results** to return the updated list of EC2 instances, S3 buckets, and Amazon RDS database instances in your account that match the specified tag keys. If you do not see resources in the list that you expect, be sure that the resources are tagged with tags that you specified in the **Grouping criteria** area.
8. When you are finished, choose **Save changes**.

To update a group in Resource Groups (AWS CLI)

In the AWS CLI, you update a group's query and update a resource group's description by using two different commands. You cannot edit an existing group's name.

1. If you do not want to change the description of your group, skip this step and go on to the next. In an AWS CLI session, type the following, and then press **Enter**, replacing the values for group name and description with your own.

   ```
   aws resource-groups update-group --group-name resource-group-name --description
    "description_text"
   ```

   The following command is an example.

   ```
   aws resource-groups update-group --group-name my-resource-group --description "EC2
    instances, S3 buckets, and RDS DBs that we are using for the test stage."
   ```

   The command returns a full, updated description of the group.
2. To update the query and tags of a group, type the following command, and then press **Enter**, replacing the values for group name, resource types, tag keys, and tag values with your own. You can have a maximum of 20 resource types in a query. For more information about the syntax of the members of the `ResourceQuery` API, see ResourceQuery in the *AWS Resource Groups API Reference*.

   Queries assign an `AND` operator to tags, so any resource that matches the specified resource types and all specified tags is returned by the query. If you specify multiple values in your query for a tag key, the query returns resources that match any of the values of the tag key. However, when there is more than one tag key, the query returns resources that match *all* tag keys, and at least one value of each specified tag key.

   ```
   aws resource-groups update-group-query --name resource-group-name --resource-query
    '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"resource_type1\",
   \"resource_type2"],\"TagFilters\":[{\"Key\":\"Key1\",\"Values\":[\"Value1\",
   \"Value2\"]},{\"Key\":\"Key2\",\"Values\":[\"Value1\",\"Value2\"]}]}"}'
   ```

   The following command is an example.

   ```
   aws resource-groups update-group-query --name my-resource-group --resource-query
    '{"Type":"TAG_FILTERS_1_0","Query":"{\"ResourceTypeFilters\":[\"AWS::EC2::Instance
   \",\"AWS::S3::Bucket\",\"AWS::RDS::DBInstance\"],\"TagFilters\":[{\"Key\":\"Stage\",
   \"Values\":[\"Test\"]}]}"}'
   ```

   The command returns the updated query as a result.

Delete Groups from AWS Resource Groups

You can use the Resource Groups console or the AWS CLI to delete resource groups from AWS Resource Groups. Deleting a resource group does not delete the resources that are members of the group or tags on member resources. It deletes only the group structure and any group-level tags.

To delete resource groups (AWS Management Console)

1. In the AWS Systems Manager console navigation pane, or from the Resource Groups drop-down menu on the AWS home page, choose **Saved Resource Groups**.
2. Choose the resource group that you want to delete.
3. On the group's detail page, choose **Delete**.
4. When you are prompted to confirm the deletion, choose **Delete**.

To delete resource groups (AWS CLI)

1. Type the following command, replacing *resource_group_name* with the name of your group, and then press **Enter**.

```
aws resource-groups delete-group --group-name resource_group_name
```

2. When you are prompted to confirm the deletion, type `yes`, and then press **Enter**.

Logging AWS Resource Groups API Calls with AWS CloudTrail

AWS Resource Groups is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in Resource Groups. CloudTrail captures all API calls for Resource Groups as events, including calls from the Resource Groups console and from code calls to the Resource Groups APIs. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket, including events for Resource Groups. 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 Groups, 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 Groups Information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in Resource Groups, that activity is recorded in a CloudTrail event along with other AWS service events in **Event history**. You can view, search, and download recent events in your AWS account. For more information, see Viewing Events with CloudTrail Event History.

For an ongoing record of events in your AWS account, including events for Resource Groups, 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 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:

- Overview for Creating a Trail
- CloudTrail Supported Services and Integrations
- Configuring Amazon SNS Notifications for CloudTrail
- Receiving CloudTrail Log Files from Multiple Regions and Receiving CloudTrail Log Files from Multiple Accounts

All Resource Groups actions are logged by CloudTrail and are documented in the AWS Resource Groups API Reference. For example, calls to the `CreateGroup`, `GetGroup`, and `UpdateGroupQuery` actions generate entries in the CloudTrail log files.

Every event or log entry contains information about who generated the request. The identity information helps you determine the following:

- Whether the request was made with root or IAM user credentials.
- Whether the request was made with temporary security credentials for a role or federated user.
- Whether the request was made by another AWS service.

For more information, see the CloudTrail `userIdentity` Element.

Understanding Resource Groups Log File Entries

A trail is a configuration that enables delivery of events as log files to an Amazon S3 bucket that you specify. CloudTrail log files contain one or more log entries. An event represents a single request from any source and includes information about the requested action, the date and time of the action, request parameters, and so on. CloudTrail log files are not an ordered stack trace of the public API calls, so they do not appear in any specific order.

The following example shows a CloudTrail log entry that demonstrates the action `CreateGroup`.


```
{"eventVersion":"1.05",
"userIdentity":{
    "type":"AssumedRole",
    "principalId":"ID number:AWSResourceGroupsUser",
    "arn":"arn:aws:sts::831000000000:assumed-role/Admin/AWSResourceGroupsUser",
    "accountId":"831000000000","accessKeyId":"ID number",
    "sessionContext":{
        "attributes":{
            "mfaAuthenticated":"false",
            "creationDate":"2018-06-05T22:03:47Z"
            },
        "sessionIssuer":{
            "type":"Role",
            "principalId":"ID number",
            "arn":"arn:aws:iam::831000000000:role/Admin",
            "accountId":"831000000000",
            "userName":"Admin"
            }
        }
    },
"eventTime":"2018-06-05T22:18:23Z",
"eventSource":"resource-groups.amazonaws.com",
"eventName":"CreateGroup",
"awsRegion":"us-west-2",
"sourceIPAddress":"100.25.190.51",
"userAgent":"console.amazonaws.com",
"requestParameters":{
    "Description": "EC2 instances that we are using for application staging.",
```

```
    "Name": "Staging",
    "ResourceQuery": {
      "Query": "string",
      "Type": "TAG_FILTERS_1_0"
      },
    "Tags": {
      "Key":"Phase",
      "Value":"Stage"
      }
    },
"responseElements":{
    "Group": {
      "Description":"EC2 instances that we are using for application staging.",
      "groupArn":"arn:aws:resource-groups:us-west-2:831000000000:group/Staging"
      "Name":"Staging"
     },
    "resourceQuery": {
      "Query":"string",
      "Type":"TAG_FILTERS_1_0"
     }
    },
"requestID":"de7z64z9-d394-12ug-8081-7zz0386fbcb6",
"eventID":"8z7z18dz-6z90-47bz-87cf-e8346428zzz3",
"eventType":"AwsApiCall",
"recipientAccountId":"831000000000"
}
```

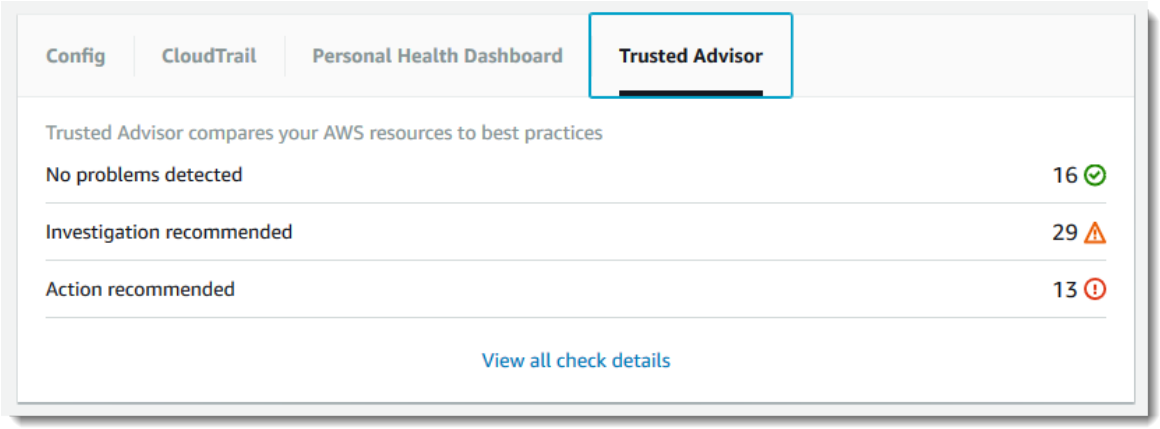

Viewing Insights about AWS Resource Groups

Insights show detailed information about the resources in your groups, such as AWS CloudTrail logs, and results of evaluations against AWS Config rules. AWS Trusted Advisor reports and the Personal Health Dashboard show events and recommendations at the account level. The AWS CloudTrail and AWS Config views show information about a single, selected resource group at a time.

Topics

Included Insights

Choosing **Insights** in the AWS Systems Manager left navigation pane shows monitoring views that are built-in, or included by default. Choose a resource group, and the page displays group members' rule compliance from AWS Config, or event log data from CloudTrail. The Personal Health Dashboard shows events by region in your account. Recommendations from Trusted Advisor are also per account. The following shows an example of a Trusted Advisor view for an account.



On the AWS Config tab, data includes rule compliance, compliance by resource type, and a history of configuration changes on group resources.

For more information about how to drill down, interpret, and use the data shown by these insights, see the following AWS User Guides.

- *AWS Config Developer Guide*
- *AWS CloudTrail User Guide*
- *AWS Trusted Advisor User Guide*

Amazon CloudWatch Dashboards

The Dashboards by CloudWatch page in AWS Systems Manager shows data directly from the CloudWatch service.

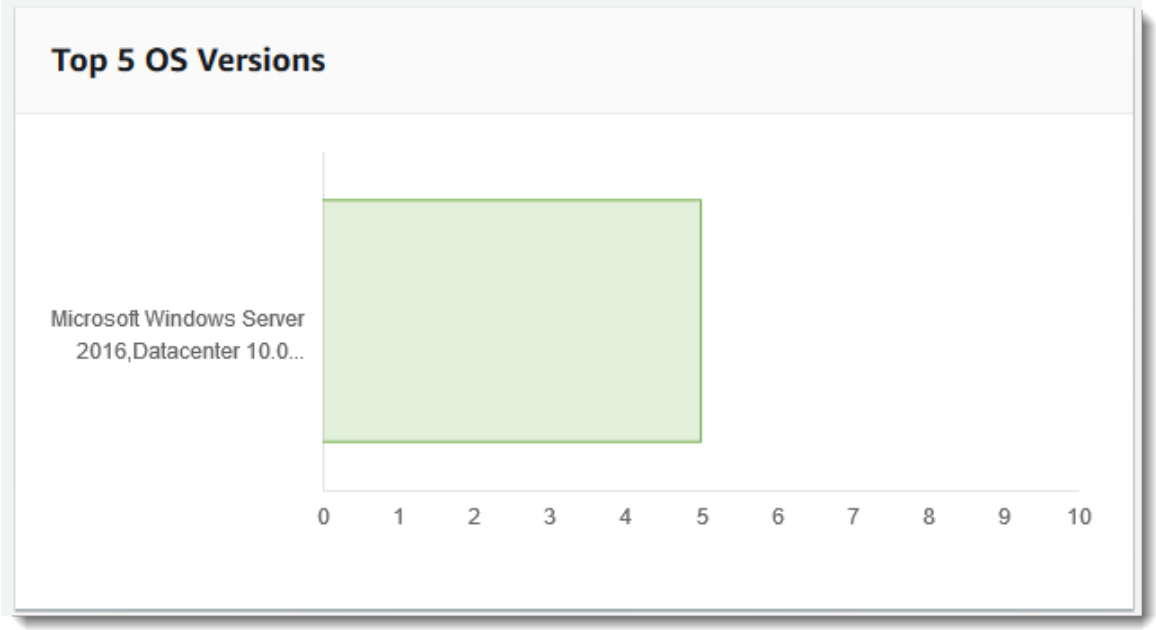
To create a CloudWatch dashboard in the AWS Systems Manager console

1. On the **Dashboard by CloudWatch** page, choose **Create new dashboard**.
2. Enter a name for the dashboard, such as the name of the service for which you want to view CloudWatch data.
3. Choose **Create dashboard**.
4. Choose the graphical format in which you want CloudWatch displayed, and then choose **Configure**.

For more information about how to use and change CloudWatch metrics, see the *Amazon CloudWatch User Guide*.

AWS Systems Manager Inventory and Compliance

The **Inventory** insights show software and packages (excluding AWS components) that are installed on instances in a group that you are managing with AWS Systems Manager, or that are tagged with a tag that you specify in the search box at the top of the page. The following screenshot shows an example of operating systems that are installed in the inventory of a group's instances.



Inventory data comes from AWS Systems Manager. For more information about how to work with inventory data, see Systems Manager Inventory Management in the *AWS Systems Manager User Guide*.

The **Compliance** insights show compliance with patch and configuration standards that you set by using AWS Systems Manager. For more information about how to set up patch baselines and configuration associations against which tagged or grouped resources can be measured, see Systems Manager Configuration Compliance in the *AWS Systems Manager User Guide*.

AWS Resource Groups Document History

update-history-change	update-history-description	update-history-date
Resource Groups and CloudTrail (p. 34)	Resource Groups now offers AWS CloudTrail support. You can view and work with logs of all Resource Groups API calls in CloudTrail.	June 29, 2018

- **API version:** 2017-11-27
- **Latest documentation update:** June 29, 2018

Earlier Updates

The following table describes important changes in each release of the *AWS Resource Groups User Guide* before June 2018.

Change	Description	Date
Initial release	Initial release of the next generation of AWS Resource Groups	November 29, 2017

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.