

---

# AWS Resource Access Manager

## User Guide



## **AWS Resource Access Manager: User Guide**

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

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

## Table of Contents

What is AWS RAM?	1
Benefits	1
What about cross-account access with resource-based policies?	1
How resource sharing works	2
Sharing your resources	2
Using shared resources	3
Service quotas	3
Accessing AWS RAM	4
Pricing	4
Compliance and international standards	4
PCI DSS	4
FedRAMP	4
SOC and ISO	5
Getting started	6
Sharing your resources	6
Enable resource sharing within AWS Organizations	6
Create a resource share	7
Using shared resources	11
Respond to the resource share invitation	11
Use the resources that are shared with you	13
Working with shared resources	14
Regional and global resources	14
What are the differences between Regional and global resources?	14
Resource shares and their Regions	15
Resources owned by you	16
Viewing resource shares you created	16
Creating a resource share	18
Updating a resource share	21
Viewing your shared resources	25
Viewing principals you share with	26
Viewing AWS RAM managed permissions	28
Updating managed permission versions	30
Deleting a resource share	31
Resources shared with you	32
Accepting and rejecting invitations	32
Viewing resource shares shared with you	34
Viewing resources shared with you	36
View principals sharing with you	37
Leaving a resource share	37
Availability Zone IDs	40
Shareable resources	43
AWS App Mesh	43
Amazon Aurora	44
AWS Certificate Manager Private Certificate Authority	44
AWS CodeBuild	45
Amazon EC2	45
EC2 Image Builder	46
AWS Glue	47
AWS License Manager	48
AWS Migration Hub Refactor Spaces	49
AWS Network Firewall	49
AWS Outposts	50
Amazon S3 on Outposts	51
AWS Resource Groups	51

Amazon Route 53 .....	52
Amazon SageMaker .....	53
AWS Systems Manager Incident Manager .....	53
Amazon VPC .....	54
AWS Cloud WAN .....	57
AWS RAM managed permissions .....	59
How AWS RAM managed permissions work .....	59
Types of AWS RAM managed permissions .....	59
Security .....	61
Data protection .....	61
Identity and access management .....	62
How AWS RAM works with IAM .....	62
AWS managed policies .....	64
Using Service-Linked Roles .....	67
Example IAM policies .....	68
Example SCPs .....	70
Disable sharing with Organizations .....	73
Logging and monitoring .....	73
Monitoring using CloudWatch Events .....	73
Logging AWS RAM API calls with AWS CloudTrail .....	75
Resilience .....	76
Infrastructure security .....	76
Using the AWS SDKs .....	78
Document history .....	79

# What is AWS Resource Access Manager?

AWS Resource Access Manager (AWS RAM) helps you securely share the AWS resources that you create in one AWS account with other AWS accounts. If you have multiple AWS accounts, you can create a resource once and use AWS RAM to make that resource usable by those other accounts. If your account is managed by AWS Organizations, then you can share resources with all the other accounts in the organization, or only those accounts contained by one or more specified organizational units (OUs). You can also share with specific AWS accounts by account ID, regardless of whether the account is part of an organization. [Some supported resource types \(p. 43\)](#) also let you share them with specified IAM roles and users.

## Contents

- [Benefits \(p. 1\)](#)
- [How resource sharing works \(p. 2\)](#)
- [Service quotas \(p. 3\)](#)
- [Accessing AWS RAM \(p. 4\)](#)
- [Pricing \(p. 4\)](#)
- [Compliance and international standards \(p. 4\)](#)

## Benefits

Why use AWS RAM? It offers the following benefits:

- **Reduces your operational overhead** – Create a resource once, and then use AWS RAM to share that resource with other accounts. This eliminates the need to provision duplicate resources in every account, which reduces operational overhead.
- **Provides security and consistency** – Simplify security management for your shared resources by using a single set of policies and permissions. If you were to instead create duplicate resources in all your separate accounts, you would have the task of implementing identical policies and permissions, and then have to keep them identical across all those accounts. Instead, all users of an AWS RAM resource share are managed by a single set of policies and permissions. AWS RAM offers a consistent experience for sharing different types of AWS resources.
- **Provides visibility and auditability** – View the usage details for your shared resources through the integration of AWS RAM with Amazon CloudWatch and AWS CloudTrail. AWS RAM provides comprehensive visibility into shared resources and accounts.

## What about cross-account access with resource-based policies?

You can share some types of AWS resources with other AWS accounts by [attaching a resource-based permission policy](#) that identifies principals outside of your AWS account. However, sharing a resource by

attaching a policy doesn't take advantage of the additional benefits that AWS RAM provides. By using AWS RAM you get the following features:

- You can share with an [organization or an organizational unit \(OU\)](#) without having to enumerate every one of the AWS account IDs. All principals in the relevant AWS accounts automatically get access to the resources in such a resource share.
- Users can see the resources shared with them directly in the originating AWS service console and API operations as if those resources were directly in the user's account. For example, if you share a Amazon VPC subnet with another account, users in that account can see the subnet in the Amazon VPC console and in the results of Amazon VPC API operations performed in that account. Resources shared by policy aren't visible this way; instead, you have to discover and explicitly refer to the resource by its ARN.
- The owners of a resource can see which principals have access to each individual resource that they have shared.
- If you share resources with an account that isn't part of your organization, then AWS RAM initiates an invitation process. The recipient must accept the invitation before that principal can access the shared resources. Sharing within an organization doesn't require an invitation.

If you have resources that you have shared by using a resource-based permission policy, you can "promote" those resources to fully AWS RAM-managed resources by using the [PromoteResourceShareCreatedFromPolicy](#) API operation, or its CLI equivalent, [promote-resource-share-created-from-policy](#).

## How resource sharing works

When you share a resource with another AWS account, you are granting access to principals in that account to the shared resource. Any policies and permissions that apply to the account you shared the resource with also apply to the shared resource. The resources in the share look like they're native resources in the AWS accounts you shared them with.

You can share both global and Regional resources. For more information, see [Sharing Regional resources compared to global resources \(p. 14\)](#).

## Sharing your resources

With AWS RAM, you share resources that you own by creating a *resource share*. To create a resource share, you specify the following:

- The AWS Region in which you want to create the resource share. In the console, you choose from the **Region** drop-down menu in the upper-right corner of the console. In the AWS CLI, you use the `--region` parameter.
- A resource share can contain only Regional resources that are in the same AWS Region as the resource share.
- A resource share can contain global resources only if the resource share is in the designated home Region, US East (N. Virginia), `us-east-1`.
- A name for the resource share.
- The list of resources that you want to grant access to as part of this resource share.
- The principals to which you grant access to the resource share. Principals can be individual AWS accounts, the accounts in an organization or an organizational unit (OU) in AWS Organizations, or individual AWS Identity and Access Management (IAM) roles or users.

### Note

Not all resource types can be shared with IAM roles and users. For information about resources that you can share with these principals, see [Shareable AWS resources \(p. 43\)](#).

- The AWS RAM permission to associate with each resource type. This is an AWS managed permission policy that determines what the principals in the other accounts can do with the resources in the resource share.

Your account retains full ownership of the resources that you share.

## Using shared resources

When the owner of a resource shares it with your account, you can access the shared resource just as you would if your account owned it. You can access the resource by using the relevant service's console, AWS Command Line Interface (AWS CLI) commands, and API operations. The API operations that principals in your account are allowed to perform vary depending on the resource type and are specified by the AWS RAM permission attached to the resource share. All IAM policies and service control policies configured in your account also continue to apply, which enables you to make use of your existing investments in security and governance controls.

When you access a shared resource using that resource's service, you have the same abilities and limitations as the AWS account that owns the resource.

- If the resource is Regional, then you can access it from only the AWS Region in which it exists in the owning account.
- If the resource is global, then you can access it from any AWS Region that the resource's service console and tools support. Note that you can view and manage the resource share and its global resources in the AWS RAM console and tools only in the designated home Region, US East (N. Virginia), `us-east-1`.

## Service quotas

Your AWS account has the following limits related to AWS RAM. You can request an increase for some of these limits. To request a limit increase, contact [AWS Support](#).

Resource	Default limit
Maximum number of resource shares per AWS Region in an account	5,000
Maximum number of shared principals per AWS Region in an account	5,000
Maximum number of shared resources per AWS Region in an account	5,000
Maximum number of pending invitations per sharing account <ul style="list-style-type: none"><li>• <i>This quota applies to only <b>sending</b> accounts who are sharing with accounts that are not part of the same AWS Organization.</i></li><li>• <i>There is no quota to limit how many pending invitations a receiving account can have.</i></li><li>• <i>Invitations are not used when sharing between two accounts that are part of the same AWS Organization and resource sharing within the AWS Organization is enabled.</i></li></ul>	20

## Accessing AWS RAM

You can work with AWS RAM in any of the following ways:

### AWS RAM console

AWS RAM provides a web-based user interface, the AWS RAM console. If you've signed up for an AWS account, you can access the AWS RAM console by signing into the [AWS Management Console](#) and choosing AWS RAM from the console home page.

You can also navigate in your browser directly to the [AWS RAM console](#). If you aren't already signed in, then you're asked to do so before the console appears.

### AWS CLI and Tools for Windows PowerShell

The AWS CLI and Tools for PowerShell provide direct access to the AWS RAM public API operations. AWS supports these tools 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 AWS RAM, see the [AWS CLI Command Reference](#) or the [AWS Tools for Windows PowerShell Cmdlet Reference](#).

### AWS SDKs

AWS provides API commands for a broad set of programming languages. For more information about getting started, see [AWS SDKs and Tools Reference Guide](#).

### Query API

If you don't use one of the supported programming languages, then the AWS RAM HTTPS Query API gives you programmatic access to AWS RAM and AWS. With the AWS RAM API, you can issue HTTPS requests directly to the service. When you use the AWS RAM API, you must include code to digitally sign requests using your credentials. For more information, see the [AWS RAM API Reference](#).

## Pricing

There are no additional charges for using AWS RAM or for creating resource shares and sharing your resources across accounts. Resource usage charges vary depending on the resource type. For more information about how AWS bills shareable resources, refer to the documentation for the resource's owning service.

## Compliance and international standards

### PCI DSS

AWS Resource Access Manager (AWS RAM) supports the processing, storage, and transmission of credit card data by a merchant or service provider, and has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).

For more information about PCI DSS, including how to request a copy of the AWS PCI Compliance Package, see [PCI DSS Level 1](#).

### FedRAMP

AWS Resource Access Manager is authorized as FedRAMP Moderate in the following AWS Regions: US East (N. Virginia), US East (Ohio), US West (N. California), and US West (Oregon).



AWS RAM is authorized as FedRAMP High in the following AWS Regions: AWS GovCloud (US-West) and AWS GovCloud (US-East).

The Federal Risk and Authorization Management Program (FedRAMP) is a US government-wide program that delivers a standard approach to the security assessment, authorization, and continuous monitoring for cloud products and services.

For more information about FedRAMP compliance, see [FedRAMP](#).

## SOC and ISO

AWS Resource Access Manager can be used for workloads subject to Service Organization Control (SOC) compliance and International Organization for Standardization (ISO) ISO 9001, ISO 27001, ISO 27017, ISO 27018 and ISO 27701 standards. Customers in finance, healthcare, and other regulated sectors can get insights into the security processes and controls that protect customer data which can be found in the SOC reports, AWS ISO and CSA STAR certificates in [AWS Artifact](#).

For more information about SOC compliance, see [SOC](#).

For more information about ISO compliance, see [ISO 9001](#), [ISO 27001](#), [ISO 27017](#), [ISO 27018](#) and [ISO 27701](#).

# Getting started with AWS RAM

With AWS Resource Access Manager, you can share resources that you own with other individual AWS accounts. If your account is managed by AWS Organizations, you can also share resources with the other accounts in your organization. You can also use resources that were shared with you by other AWS accounts.

If you don't enable sharing within AWS Organizations, you can't share resources with your organization or with the organizational units (OU) in your organization. However, you can still share resources with individual AWS accounts in your organization. For [supported resource types \(p. 43\)](#), you can also share resources with individual AWS Identity and Access Management (IAM) roles or users in your organization. In this case, these principals are treated as if they were external accounts, rather than as part of your organization. They receive an invitation to join the resource share, and they must accept the invitation to gain access to the shared resources.

## Contents

- [Sharing your AWS resources \(p. 6\)](#)
- [Using shared AWS resources \(p. 11\)](#)

## Sharing your AWS resources

To share a resource that you own by using AWS RAM, do the following:

- [Enable resource sharing within AWS Organizations \(p. 6\)](#) (optional)
- [Create a resource share \(p. 7\)](#)

## Notes

- Sharing a resource makes it available for use by principals outside of the AWS account that created the resource. Sharing doesn't change any permissions or quotas that apply to the resource in the account that created it.
- AWS RAM is a Regional service. The principals that you share with can access resource shares in only the AWS Regions in which they were created.
- Some resources have special considerations and prerequisites for sharing. For more information, see [Shareable AWS resources \(p. 43\)](#).

## Enable resource sharing within AWS Organizations

When your account is managed by AWS Organizations, you can take advantage of that to share resources more easily. With or without Organizations, a user can share with individual accounts. However, if your account is in an organization, then you can share with individual accounts, or with all accounts in the organization or in an OU without having to enumerate each account.

To share resources within an organization, you must first use the AWS RAM console or AWS Command Line Interface (AWS CLI) to enable sharing with AWS Organizations. When you share resources in your organization, AWS RAM doesn't send invitations to principals. Principals in your organization gain access to shared resources without exchanging invitations.

When you enable resource sharing within your organization, AWS RAM creates a service-linked role called `AWSServiceRoleForResourceAccessManager`. This role can be assumed by only the AWS RAM service, and grants AWS RAM permission to retrieve information about the organization it is a member of, by using the AWS managed policy `AWSResourceAccessManagerServiceRolePolicy`.

If you no longer need to share resources with your entire organization or OUs, you can disable resource sharing. For more information, see [Disabling resource sharing with AWS Organizations \(p. 73\)](#).

### Minimum permissions

To run the procedures below, you must have the following permissions:

- `ram:EnableSharingWithAwsOrganization`
- `iam:CreateServiceLinkedRole`
- `organizations:DescribeOrganization`

### Requirements

- You can perform these steps only while signed in as a principal in the organization's management account.
- The organization must have all features enabled. For more information, see [Enabling all features in your organization](#) in the *AWS Organizations User Guide*.

### Important

You must enable sharing with AWS Organizations by using the AWS RAM console or the [enable-sharing-with-aws-organization](#) AWS CLI command. This ensures that the `AWSServiceRoleForResourceAccessManager` service-linked role is created. If you enable trusted access with AWS Organizations by using the AWS Organizations console or the [enable-aws-service-access](#) AWS CLI command, the `AWSServiceRoleForResourceAccessManager` service-linked role isn't created, and you can't share resources within your organization.

### Console

#### To enable resource sharing within your organization

1. Open the [Settings](#) page in the AWS RAM console.
2. Choose **Enable sharing with AWS Organizations**, and then choose **Save settings**.

### AWS CLI

#### To enable resource sharing within your organization

Use the [enable-sharing-with-aws-organization](#) command.

This command can be used in any AWS Region, and it enables sharing with AWS Organizations in all Regions in which AWS RAM is supported.

```
$ aws ram enable-sharing-with-aws-organization
{
  "returnValue": true
}
```

## Create a resource share

To share resources that you own, create a resource share. When you create a resource share, you do the following:

1. Add the resources that you want to share.

2. For each resource type that you include in the share, specify the permission to use for that resource type.
  - If only the *default permission* is available for a resource type, then AWS RAM automatically associates that permission with the resource type and there is no action for you.
  - If more than the default AWS RAM managed permission is available for a resource type, then you must choose the permission to associate with that resource type.
3. Specify the principals that you want to have access to the resources.

## Considerations

- The resource types that you can include in a resource share are listed at [Shareable AWS resources \(p. 43\)](#).
- You can share a resource only if you own it. You can't share a resource that's shared with you.
- AWS RAM is a Regional service. When you share a resource with principals in other AWS accounts, they must access each resource from the same AWS Region that it was created in. For supported global resources, you can access those resources from any AWS Region that's supported by that resource's service console and tools. Note that you can view such resource shares and their global resources in the AWS RAM console and tools only in the designated home Region, US East (N. Virginia), `us-east-1`. For more information about AWS RAM and global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
- If the account you're sharing from is part of an organization in AWS Organizations and sharing within your organization is enabled, any principals in the organization that you share with are automatically granted access to the shared resources without the use of invitations. A principal in an account with whom you share outside of the context of an organization receives an invitation to join the resource share and is granted access to the shared resources only after they accept the invitation.
- For the following resource types you have seven days to accept the invitation to join the share for the following resource types. If you don't accept the invitation before it expires, the invitation is automatically declined.

### Important

For shared resource types **not** on the following list, you have **12 hours** to accept the invitation to join the resource share. If you try to accept the invitation after 12 hours, RAM fails to process the invitation and the originating account must share the resources again to generate a new invitation.

- Amazon Aurora – DB clusters
- Amazon EC2 – capacity reservations and dedicated hosts
- AWS License Manager – License configurations
- AWS Outposts – Local gateway route tables, outposts, and sites
- Amazon Route 53 – Forwarding rules
- Amazon VPC – Customer-owned IPv4 addresses, prefix lists, subnets, traffic mirror targets, transit gateways, transit gateway multicast domains
- After you add an organization or an organization unit (OU) to a resource share, changes to the accounts that are in an OU or accounts that join or leave an organization dynamically affect the resource share. For example, if you add a new account to an OU that has access to a resource share, then the new member account automatically receives access to the shared resources.
- You can add only the organization your account is a member of, and OUs from that organization to your resource shares. You can't add OUs or organizations from outside your own organization to a resource share as principals. However, you can add individual AWS accounts, IAM users, and IAM roles from outside your organization as principals to a resource share.

### Note

Not all resource types can be shared with IAM roles and users. For information about resources that you can share with these principals, see [Shareable AWS resources \(p. 43\)](#).

## Console

### To create a resource share

1. Open the [AWS RAM console](#).
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#). If you want to include global resources in the resource share, then you must choose the designated home Region, US East (N. Virginia), us-east-1.
3. If you're new to AWS RAM, choose **Create a resource share** from the home page. Otherwise, choose **Create resource share** from the [Shared by me : Resource shares](#) page.
4. In **Step 1: Specify resource share details**, do the following:
  - a. For **Name**, enter a descriptive name for the resource share.
  - b. Under **Resources**, choose resources to add to the resource share as follows:
    - For **Select resource type**, choose the type of resource to share. This filters the list of shareable resources to only those resources of the selected type.
    - In the resulting list of resources, select the check boxes next to the individual resources that you want to share. The selected resources move under **Selected resources**.

If you're sharing resources that are associated with a specific availability zone, then using the Availability Zone ID (AZ ID) helps you determine the relative location of these resources across accounts. For more information, see [Availability Zone IDs for your AWS resources \(p. 40\)](#).
  - c. (Optional) To [attach tags](#) to the resource share, under **Tags**, enter a tag key and value. Add others by choosing **Add new tag**. Repeat this step as needed. These tags apply to only the resource share itself, not to the resources in the resource share.
5. Choose **Next**.
6. In **Step 2: Associate a permission with each resource type**, if more than the default AWS RAM managed permission is available, then you can choose which permission to associate with the resource type. If only the default permission is available, then AWS RAM automatically associates this permission with the resource type. For more information, see [Types of AWS RAM managed permissions \(p. 59\)](#).

To display the actions that the permission allows, expand **View the actions that are allowed by this permission**.

7. Choose **Next**.
8. In **Step 3: Choose principals to grant access**, do the following:
  - a. By default, **Allow sharing with external principals** is selected, which means that, for those resource types that support it, you can share resources with AWS accounts that are outside of your organization. This doesn't affect resource types that can be shared *only* within an organization, such as Amazon VPC subnets. You can also share some [supported resource types \(p. 43\)](#) with IAM roles and users.

To restrict resource sharing to only accounts and principals in your organization, choose **Allow sharing with principals in your organization only**.
  - b. For **Principals**, do the following:
    - To add the organization, an organizational unit (OU), or an AWS account that is part of an organization, turn on **Display organizational structure**. This displays a tree view of your organization. Then, select the check box next to each principal that you want to add.

- If you select the organization (the ID begins with o-), then all AWS accounts in the organization can access the resource share.
- If you select an OU (the ID begins with ou-), then all AWS accounts in that OU and its child OUs can access the resource share.
- If you select an individual AWS account, then only that account can access the resource share.

**Note**

The **Display organizational structure** toggle appears only if sharing with AWS Organizations is enabled and you're signed in to the management account for the organization.

You can't use this method to specify an AWS account outside your organization, or an IAM role or IAM user. Instead, you must turn off **Display organizational structure** and use the dropdown list and text box to enter the ID or ARN.

- To specify a principal by ID or ARN, including principals that are outside of the organization, then for each principal, select the principal type. Next, enter the ID (for an AWS account, organization, or OU) or ARN (for an IAM user or role), and then choose **Add**. The available principal types and ID and ARN formats are as follows:

- **AWS account** – To add an AWS account, enter the 12-digit account ID. For example:

123456789012

- **Organization** – To add all of the AWS accounts in your organization, enter the ID of the organization. For example:

o-abcd1234

- **Organizational unit (OU)** – To add an OU, enter the ID of the OU. For example:

ou-abcd-1234efgh

- **IAM role** – To add an IAM role, enter the ARN of the role. Use the following syntax.

arn:*partition*:iam::*account*:role/*role-name*

For example:

arn:aws:iam::123456789012:role/MyS3AccessRole

**Note**

To obtain the unique ARN for an IAM role, [view the list of roles in the IAM console](#), use the [get-role](#) AWS CLI command or the [GetRole](#) API action.

- **IAM user** – To add an IAM user, enter the ARN of the user. Use the following syntax.

arn:*partition*:iam::*account*:user/*user-name*

For example:

arn:aws:iam::123456789012:user/JohnDoe

**Note**

To obtain the unique ARN for an IAM user, [view the list of users in the IAM console](#), use the [get-user](#) AWS CLI command or the [GetUser](#) API action.

- c. For **Selected principals**, verify that the principals you specified appear in the list.

9. Choose **Next**.

10. In **Step 4: Review and create**, review the configuration details for your resource share. To change the configuration for any step, choose the link that corresponds to the step you want to go back to and make the required changes.

11. After you finish reviewing the resource share, choose **Create resource share**.

It can take a few minutes for the resource and principal associations to complete. Allow this process to complete before you try to use the resource share.

12. You can add and remove resources and principals or apply custom tags to your resource share at any time. You can change permission for resource types that are included in your resource share, for those types that support more than the default permission. You can delete your resource share when you no longer want to share the resources. For more information, see [Share AWS resources owned by you \(p. 16\)](#).

## AWS CLI

### To create a resource share

Use the [create-resource-share](#) command. The following command creates a resource share that is shared with all of the AWS accounts in the organization. The share contains an AWS License Manager license configuration, and it grants the default permissions for that resource type.

```
$ aws ram create-resource-share \
  --region us-east-1 \
  --name MyLicenseConfigShare \
  --permission-arns arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionLicenseConfiguration \
  --resource-arns arn:aws:license-manager:us-east-1:123456789012:license-
configuration:lic-abc123 \
  --principals arn:aws:organizations::123456789012:organization/o-1234abcd
{
  "resourceShare": {
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/12345678-abcd-09876543",
    "name": "MyLicenseConfigShare",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": true,
    "status": "ACTIVE",
    "creationTime": "2021-09-14T20:42:40.266000-07:00",
    "lastUpdatedTime": "2021-09-14T20:42:40.266000-07:00"
  }
}
```

## Using shared AWS resources

To start using resources that were shared with your account using AWS Resource Access Manager, complete the following tasks.

### Tasks

- [Respond to the resource share invitation \(p. 11\)](#)
- [Use the resources that are shared with you \(p. 13\)](#)

## Respond to the resource share invitation

If you receive an invitation to join a resource share, you must accept it to gain access to the shared resources. If you're part of an organization in AWS Organizations and sharing in your organization is enabled, principals in your organization are automatically granted access to the shared resources. Those principals don't receive invitations.

## Console

### To respond to invitations

1. Open the [Shared with me : Resource shares](#) page in the AWS RAM console.

#### Note

A resource share is visible in only the AWS Region in which it was created. If an expected resource share doesn't appear in the console, you might need to switch to a different AWS Region using the drop-down control in the upper-right corner.

2. Review the list of resource shares to which you have been granted access.

The **Status** column indicates your current participation status for the resource share. The Pending status indicates that you have been added to a resource share, but you have not yet accepted or rejected the invitation.

3. To respond to the resource share invitation, select the resource share ID and choose **Accept resource share** to accept the invitation, or **Reject resource share** to decline the invitation. If you reject the invitation, you don't get access to the resources. If you accept the invitation, you gain access to the resources.

## AWS CLI

To start, get a list of the resource share invitations that are available to you. The following example command was run in the `us-west-2` Region, and shows one resource share is available in the PENDING state.

```
$ aws ram get-resource-share-invitations
{
  "resourceShareInvitations": [
    {
      "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111122223333:resource-
share-invitation/1234abcd-ef12-9876-5432-aaaaaa11111",
      "resourceShareName": "MyNewResourceShare",
      "resourceShareArn": "arn:aws:ram:us-west-2:111122223333:resource-
share/1234abcd-ef12-9876-5432-bbbbbb22222",
      "senderAccountId": "111122223333",
      "receiverAccountId": "444455556666",
      "invitationTimestamp": "2021-09-15T15:00:32.568000-07:00",
      "status": "PENDING"
    }
  ]
}
```

You can use the Amazon Resource Name (ARN) of the invitation from the previous command as a parameter in the next command to accept that invitation.

```
$ aws ram accept-resource-share-invitation \
--resource-share-invitation-arn arn:aws:ram:us-west-2:111122223333:resource-share-
invitation/1234abcd-ef12-9876-5432-aaaaaa11111
{
  "resourceShareInvitation": {
    "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111122223333:resource-
share-invitation/1234abcd-ef12-9876-5432-aaaaaa11111",
    "resourceShareName": "MyNewResourceShare",
    "resourceShareArn": "arn:aws:ram:us-west-2:111122223333:resource-
share/1234abcd-ef12-9876-5432-bbbbbb22222",
    "senderAccountId": "111122223333",
    "receiverAccountId": "444455556666",
    "invitationTimestamp": "2021-09-15T15:14:12.580000-07:00",
    "status": "ACCEPTED"
  }
}
```



```
}  
}
```

The output shows that the `status` has changed to `ACCEPTED`. The resources that are included in that resource share are now available to principals in the accepting account.

## Use the resources that are shared with you

After you accept the invitation to join a resource share, you can perform specific actions on the shared resources. These actions vary by resource type. For more information, see [Shareable AWS resources \(p. 43\)](#). The resources are available directly in each resource's service console and API/CLI operations. If the resource is regional, then you must use the correct AWS Region in the service console or API/CLI command. If the resource is global, then you must use the designated home Region, US East (N. Virginia), `us-east-1`. To view the resource in AWS RAM, you must open the AWS RAM console to the AWS Region that the resource share was created in.

# Working with shared AWS resources

You can use AWS Resource Access Manager (AWS RAM) to share AWS resources that you own and access AWS resources that are shared with you.

## Contents

- [Sharing Regional resources compared to global resources \(p. 14\)](#)
  - [What are the differences between Regional and global resources? \(p. 14\)](#)
  - [Resource shares and their Regions \(p. 15\)](#)
- [Share AWS resources owned by you \(p. 16\)](#)
  - [Viewing resource shares you created in AWS RAM \(p. 16\)](#)
  - [Creating a resource share in AWS RAM \(p. 18\)](#)
  - [Update a resource share in AWS RAM \(p. 21\)](#)
  - [Viewing your shared resources in AWS RAM \(p. 25\)](#)
  - [Viewing the principals you share resources with in AWS RAM \(p. 26\)](#)
  - [Viewing AWS RAM managed permissions \(p. 28\)](#)
  - [Updating AWS RAM managed permissions to a newer version \(p. 30\)](#)
  - [Deleting a resource share in AWS RAM \(p. 31\)](#)
- [Access AWS resources shared with you \(p. 32\)](#)
  - [Accepting and rejecting resource share invitations \(p. 32\)](#)
  - [Viewing resource shares shared with you \(p. 34\)](#)
  - [Viewing resources shared with you \(p. 36\)](#)
  - [View principals sharing with you \(p. 37\)](#)
  - [Leaving a resource share \(p. 37\)](#)
    - [Prerequisites for leaving a resource share \(p. 38\)](#)
    - [How to leave a resource share \(p. 39\)](#)
- [Availability Zone IDs for your AWS resources \(p. 40\)](#)

## Sharing Regional resources compared to global resources

This topic discusses the differences in how AWS Resource Access Manager (AWS RAM) works with Regional and global resources.

### What are the differences between Regional and global resources?

#### Regional resources

Most resources that you can share with AWS RAM are *Regional*. You create them in a specified AWS Region, and then they exist in that Region. To see or interact with those resources, you must direct your operations to that Region. For example, to create an Amazon Elastic Compute Cloud (Amazon EC2) instance with the AWS Management Console, you [choose the AWS Region](#) that you want to create the instance in. If you use the AWS Command Line Interface (AWS CLI) to create the instance, then you include the `--region` parameter. The AWS SDKs each have their own equivalent mechanism to specify the Region that the operation uses.

There are several reasons for using Regional resources. One good reason is to ensure that the resources, and the service endpoints that you use to access them, are as close to the customer as possible. This improves performance by minimizing latency. Another reason is to provide an isolation boundary. This lets you create independent copies of resources in multiple Regions to distribute the load and improve scalability. At the same time, it isolates the resources from each other to improve availability.

If you specify a different AWS Region in the console or in an AWS CLI command, then you can no longer see or interact with the resources you could see in the previous Region.

When you look at the [Amazon Resource Name \(ARN\)](#) for a Regional resource, the Region that contains the resource is specified as the fourth field in the ARN. For example, an Amazon EC2 instance is a Regional resource. Such resources have ARNs that looks similar to the following sample for a VPC that exists in the `us-east-1` Region.

```
arn:aws:ec2:us-east-1:123456789012:instance/i-0a6f30921424d3eee
```

### Global resources

Some AWS services support resources that you can access *globally*, meaning that you can use the resource from *anywhere*. You don't specify an AWS Region in a global service's console. To access a global resource, you don't specify a `--region` parameter when using the service's AWS CLI and AWS SDK operations.

Global resources support cases where it's critical that only one instance of a particular resource can exist at a time. In such scenarios, replication or synchronization between copies in different Regions isn't adequate. Having to access a single global endpoint, with the possible increase in latency, is considered acceptable to ensure that any changes are instantaneously visible to consumers of the resource. For example, when you create an AWS Cloud WAN core network as a global resource, it's consistent to all users. It appears as a single, contiguous global network across all Regions.

The [Amazon Resource Name \(ARN\)](#) for a global resource doesn't include a Region. The fourth field of such an ARN is empty, such as the following sample ARN for a Cloud WAN core network.

```
arn:aws:networkmanager::123456789012:core-network/core-network-0514d38fa6f796cea
```

## Resource shares and their Regions

AWS RAM is a Regional service, and a resource share is Regional. Therefore, a resource share can contain resources from the same AWS Region as the resource share, and any supported global resources. The Region in which you create the resource share is the resource share's *home Region*.

### Important

Currently, you can create resource shares with global resources **only in the designated home Region** US East (N. Virginia) Region, `us-east-1`. Although you can create the resource share only in that single home Region, any shared global resource appears as a standard global resource when viewed in that service's console or CLI and SDK operations. The restriction to the home Region applies only to the resource share, not the resources it contains.

To share a Regional resource that you created in the `us-west-2` Region, you must configure the AWS RAM console to use `us-west-2` and create the resource share there. You can't create a resource share that includes Regional resources from different AWS Regions. This means that to share resources from both `us-west-2` and `eu-north-1`, you must create two different resource shares. You can't combine resources from two different Regions into a single resource share.

To share a global resource in the AWS RAM console, you must configure the AWS RAM console to use the designated home Region, US East (N. Virginia) `us-east-1`. Then, create the resource share in the

designated home Region. You can mix global resources in a resource share only with resources from the `us-east-1` Region.

Even though the global resource is viewable in an AWS RAM resource share in only the designated home Region, it's still a global resource after you share it. You can access it in the shared AWS accounts from any Region from which you could access it in the original AWS account.

### Considerations

- To create a resource share in the AWS RAM console, you must use the Region that contains the resources that you want to share. If you want to include a global resource, then you must use the designated home Region to create the share. For example, to share an AWS Cloud WAN core network, you must create the resource share in the `us-east-1` Region.
- To view or modify a resource share in the AWS RAM console, you must use the Region that contains the resource share. Similarly, the AWS RAM AWS CLI and SDK operations let you interact with only resource shares that are in the Region that you specify in your operation. To view or modify resource shares that contain global resources, you must use the designated home Region, US East (N. Virginia), `us-east-1`.
- To view a Regional resource in the AWS RAM console to include it in a resource share, you must use the Region that contains the Regional resource.
- To view a global resource in the AWS RAM console to include it in a resource share, you must use the designated home Region, US East (N. Virginia), `us-east-1`.
- You can create a resource share with **both** Regional and global resources in only the designated home Region, US East (N. Virginia), `us-east-1`.

## Share AWS resources owned by you

You can use AWS Resource Access Manager (AWS RAM) to share the resources that you specify with the principals that you specify. This section describes how you can create new resource shares, modify existing resource shares, and delete resource shares that you no longer need.

### Topics

- [Viewing resource shares you created in AWS RAM \(p. 16\)](#)
- [Creating a resource share in AWS RAM \(p. 18\)](#)
- [Update a resource share in AWS RAM \(p. 21\)](#)
- [Viewing your shared resources in AWS RAM \(p. 25\)](#)
- [Viewing the principals you share resources with in AWS RAM \(p. 26\)](#)
- [Viewing AWS RAM managed permissions \(p. 28\)](#)
- [Updating AWS RAM managed permissions to a newer version \(p. 30\)](#)
- [Deleting a resource share in AWS RAM \(p. 31\)](#)

## Viewing resource shares you created in AWS RAM

You can view a list of the resource shares that you have created. You can see which resources you're sharing and the principals with whom they're shared.

### Console

#### To view your resource shares

1. Open the [Shared by me : Resource shares](#) page in the AWS RAM console.

2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. (Optional) Apply a filter to find specific resource shares. You can apply multiple filters to narrow your search. You can type a keyword, such as part of a resource share name to list only those resource shares that include that text in the name. Choose the text box to see a dropdown list of suggested attribute fields. After you choose one, you can choose from the list of available values for that field. You can add other attributes or keywords until you find the resource you want.
4. Choose the name of the resource share to review. The console displays the following information about the resource share:
  - **Summary** – Lists the resource share name, ID, owner, Amazon Resource Name (ARN), creation date, whether it allows sharing with external accounts, and its current status.
  - **Permissions** – Lists the AWS RAM managed permissions that are attached to this resource share. There can be at most one permission per resource type included in the resource share.
  - **Shared resources** – Lists the individual resources that are included in the resource share. Choose the ID of a resource to open a new browser tab to view the resource in its native service's console.
  - **Shared principals** – Lists the principals with whom the resources are shared.
  - **Tags** – Lists the tag key-value pairs that are attached to the resource share itself; these are not the tags attached to the individual resources included in the resource share.

## AWS CLI

### To view your resource shares

You can use the [get-resource-shares](#) command with the parameter `--resource-owner` set to `SELF` to display details of the resource shares created in your AWS account.

The following example shows the resource shares that are shared in the current AWS Region (us-east-1) for the calling AWS account. To get the resource shares created in a different Region, use the `--region <region-code>` parameter. To get resource shares that include global resources, you must specify the Region US East (N. Virginia), us-east-1.

```
$ aws ram get-resource-shares \
  --resource-owner SELF
{
  "resourceShares": [
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/2ebe77d7-4156-4a93-87a4-228568d04425",
      "name": "MySubnetShare",
      "owningAccountId": "123456789012",
      "allowExternalPrincipals": true,
      "status": "ACTIVE",
      "creationTime": "2021-09-10T15:38:54.449000-07:00",
      "lastUpdatedTime": "2021-09-10T15:38:54.449000-07:00",
      "featureSet": "STANDARD"
    },
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/818d71dd-7512-4f71-99c6-2ae57aa010bc",
      "name": "MyLicenseConfigShare",
      "owningAccountId": "123456789012",
      "allowExternalPrincipals": true,
      "status": "ACTIVE",
```

```
        "creationTime": "2021-09-14T20:42:40.266000-07:00",  
        "lastUpdatedTime": "2021-09-14T20:42:40.266000-07:00",  
        "featureSet": "STANDARD"  
      }  
    ]  
  }  
}
```

## Creating a resource share in AWS RAM

To share resources that you own, create a resource share. When you create a resource share, you do the following:

1. Add the resources that you want to share.
2. For each resource type that you include in the share, specify the permission to use for that resource type.
  - If only the *default permission* is available for a resource type, then AWS RAM automatically associates that permission with the resource type and there is no action for you.
  - If more than the default AWS RAM managed permission is available for a resource type, then you must choose the permission to associate with that resource type.
3. Specify the principals that you want to have access to the resources.

### Considerations

- The resource types that you can include in a resource share are listed at [Shareable AWS resources \(p. 43\)](#).
- You can share a resource only if you own it. You can't share a resource that's shared with you.
- AWS RAM is a Regional service. When you share a resource with principals in other AWS accounts, they must access each resource from the same AWS Region that it was created in. For supported global resources, you can access those resources from any AWS Region that's supported by that resource's service console and tools. Note that you can view such resource shares and their global resources in the AWS RAM console and tools only in the designated home Region, US East (N. Virginia), `us-east-1`. For more information about AWS RAM and global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
- If the account you're sharing from is part of an organization in AWS Organizations and sharing within your organization is enabled, any principals in the organization that you share with are automatically granted access to the shared resources without the use of invitations. A principal in an account with whom you share outside of the context of an organization receives an invitation to join the resource share and is granted access to the shared resources only after they accept the invitation.
- For the following resource types you have seven days to accept the invitation to join the share for the following resource types. If you don't accept the invitation before it expires, the invitation is automatically declined.

#### Important

For shared resource types **not** on the following list, you have **12 hours** to accept the invitation to join the resource share. If you try to accept the invitation after 12 hours, RAM fails to process the invitation and the originating account must share the resources again to generate a new invitation.

- Amazon Aurora – DB clusters
- Amazon EC2 – capacity reservations and dedicated hosts
- AWS License Manager – License configurations
- AWS Outposts – Local gateway route tables, outposts, and sites
- Amazon Route 53 – Forwarding rules

- Amazon VPC – Customer-owned IPv4 addresses, prefix lists, subnets, traffic mirror targets, transit gateways, transit gateway multicast domains
- After you add an organization or an organization unit (OU) to a resource share, changes to the accounts that are in an OU or accounts that join or leave an organization dynamically affect the resource share. For example, if you add a new account to an OU that has access to a resource share, then the new member account automatically receives access to the shared resources.
- You can add only the organization your account is a member of, and OUs from that organization to your resource shares. You can't add OUs or organizations from outside your own organization to a resource share as principals. However, you can add individual AWS accounts, IAM users, and IAM roles from outside your organization as principals to a resource share.

**Note**

Not all resource types can be shared with IAM roles and users. For information about resources that you can share with these principals, see [Shareable AWS resources \(p. 43\)](#).

Console

**To create a resource share**

1. Open the [AWS RAM console](#).
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#). If you want to include global resources in the resource share, then you must choose the designated home Region, US East (N. Virginia), us-east-1.
3. If you're new to AWS RAM, choose **Create a resource share** from the home page. Otherwise, choose **Create resource share** from the [Shared by me : Resource shares](#) page.
4. In **Step 1: Specify resource share details**, do the following:
  - a. For **Name**, enter a descriptive name for the resource share.
  - b. Under **Resources**, choose resources to add to the resource share as follows:
    - For **Select resource type**, choose the type of resource to share. This filters the list of shareable resources to only those resources of the selected type.
    - In the resulting list of resources, select the check boxes next to the individual resources that you want to share. The selected resources move under **Selected resources**.

If you're sharing resources that are associated with a specific availability zone, then using the Availability Zone ID (AZ ID) helps you determine the relative location of these resources across accounts. For more information, see [Availability Zone IDs for your AWS resources \(p. 40\)](#).
  - c. (Optional) To [attach tags](#) to the resource share, under **Tags**, enter a tag key and value. Add others by choosing **Add new tag**. Repeat this step as needed. These tags apply to only the resource share itself, not to the resources in the resource share.
5. Choose **Next**.
6. In **Step 2: Associate a permission with each resource type**, if more than the default AWS RAM managed permission is available, then you can choose which permission to associate with the resource type. If only the default permission is available, then AWS RAM automatically associates this permission with the resource type. For more information, see [Types of AWS RAM managed permissions \(p. 59\)](#).

To display the actions that the permission allows, expand **View the actions that are allowed by this permission**.
7. Choose **Next**.

8. In **Step 3: Choose principals to grant access**, do the following:

- a. By default, **Allow sharing with external principals** is selected, which means that, for those resource types that support it, you can share resources with AWS accounts that are outside of your organization. This doesn't affect resource types that can be shared *only* within an organization, such as Amazon VPC subnets. You can also share some [supported resource types](#) (p. 43) with IAM roles and users.

To restrict resource sharing to only accounts and principals in your organization, choose **Allow sharing with principals in your organization only**.

- b. For **Principals**, do the following:
  - To add the organization, an organizational unit (OU), or an AWS account that is part of an organization, turn on **Display organizational structure**. This displays a tree view of your organization. Then, select the check box next to each principal that you want to add.
  - If you select the organization (the ID begins with o-), then all AWS accounts in the organization can access the resource share.
  - If you select an OU (the ID begins with ou-), then all AWS accounts in that OU and its child OUs can access the resource share.
  - If you select an individual AWS account, then only that account can access the resource share.

**Note**

The **Display organizational structure** toggle appears only if sharing with AWS Organizations is enabled and you're signed in to the management account for the organization.

You can't use this method to specify an AWS account outside your organization, or an IAM role or IAM user. Instead, you must turn off **Display organizational structure** and use the dropdown list and text box to enter the ID or ARN.

- To specify a principal by ID or ARN, including principals that are outside of the organization, then for each principal, select the principal type. Next, enter the ID (for an AWS account, organization, or OU) or ARN (for an IAM user or role), and then choose **Add**. The available principal types and ID and ARN formats are as follows:

- **AWS account** – To add an AWS account, enter the 12-digit account ID. For example:

123456789012

- **Organization** – To add all of the AWS accounts in your organization, enter the ID of the organization. For example:

o-abcd1234

- **Organizational unit (OU)** – To add an OU, enter the ID of the OU. For example:

ou-abcd-1234efgh

- **IAM role** – To add an IAM role, enter the ARN of the role. Use the following syntax.

arn:*partition*:iam::*account*:role/*role-name*

For example:

arn:aws:iam::123456789012:role/MyS3AccessRole

**Note**

To obtain the unique ARN for an IAM role, [view the list of roles in the IAM console](#), use the [get-role](#) AWS CLI command or the [GetRole](#) API action.

- **IAM user** – To add an IAM user, enter the ARN of the user. Use the following syntax.

arn:*partition*:iam::*account*:user/*user-name*



For example:

```
arn:aws:iam::123456789012:user/JohnDoe
```

#### Note

To obtain the unique ARN for an IAM user, [view the list of users in the IAM console](#), use the [get-user](#) AWS CLI command or the [GetUser](#) API action.

- c. For **Selected principals**, verify that the principals you specified appear in the list.
9. Choose **Next**.
10. In **Step 4: Review and create**, review the configuration details for your resource share. To change the configuration for any step, choose the link that corresponds to the step you want to go back to and make the required changes.
11. After you finish reviewing the resource share, choose **Create resource share**.

It can take a few minutes for the resource and principal associations to complete. Allow this process to complete before you try to use the resource share.

12. You can add and remove resources and principals or apply custom tags to your resource share at any time. You can change permission for resource types that are included in your resource share, for those types that support more than the default permission. You can delete your resource share when you no longer want to share the resources. For more information, see [Share AWS resources owned by you \(p. 16\)](#).

## AWS CLI

### To create a resource share

Use the [create-resource-share](#) command. The following command creates a resource share that is shared with all of the AWS accounts in the organization. The share contains an AWS License Manager license configuration, and it grants the default permissions for that resource type.

```
$ aws ram create-resource-share \
  --region us-east-1 \
  --name MyLicenseConfigShare \
  --permission-arns arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionLicenseConfiguration \
  --resource-arns arn:aws:license-manager:us-east-1:123456789012:license-
configuration:lic-abc123 \
  --principals arn:aws:organizations::123456789012:organization/o-1234abcd
{
  "resourceShare": {
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/12345678-abcd-09876543",
    "name": "MyLicenseConfigShare",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": true,
    "status": "ACTIVE",
    "creationTime": "2021-09-14T20:42:40.266000-07:00",
    "lastUpdatedTime": "2021-09-14T20:42:40.266000-07:00"
  }
}
```

## Update a resource share in AWS RAM

You can update a resource share in AWS RAM at any time in the following ways:

- You can add principals, resources, or tags to a resource share that you created.

- For resource types that support more than the default AWS RAM managed permission, you can choose which permission applies to resources of each type.
- You can revoke access to shared resources by removing principals or resources from a resource share. If you revoke access, principals no longer have access to the shared resources.

#### Note

Principals with whom you share resources can leave your resource share if the share is empty or contains only resource types that support leaving a resource share. If the resource share contains resource types that don't support leaving, a message appears to inform principals that they must contact the share owner. In this case, you, as the owner of the resource share, must remove the principals from your resource share. For a list of resource types that don't support this action, see [Prerequisites for leaving a resource share \(p. 38\)](#).

#### Console

##### To update a resource share

1. Navigate to the [Shared by me : Resource shares](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. Select the resource share and then choose **Modify**.
4. In **Step 1: Specify resource share details**, review the resource share details, and if required, update any of the following:
  - a. (Optional) To change the name of the resource share, edit **Name**.
  - b. (Optional) To add a resource to the resource share, under **Resources**, choose the type of resource and then select the check box next to the resource to add it to the resource share. Global resources appear only if you set the Region to US East (N. Virginia), (us-east-1) in the AWS Management Console.
  - c. (Optional) To remove a resource from the resource share, locate the resource under **Selected resources**, and then choose the **X** next to the resource's ID.
  - d. (Optional) To add a tag to the resource share, under **Tags**, enter a tag key and value in the empty text boxes. To add more than one tag key and value pair, choose **Add new tag**. You can add up to 50 tags.
  - e. To remove a tag from the resource share, under **Tags**, locate the tag and choose **Remove** next to it.
5. Choose **Next**.
6. (Optional) In **Step 2: Associate a permission with each resource type**, if more than the default AWS RAM managed permission is available, you can choose which permission to associate with the resource type. For more information, see [Types of AWS RAM managed permissions \(p. 59\)](#). If only the default AWS RAM managed permission is available, then you can't alter anything for this resource type.

To display the actions that the AWS RAM managed permission allows, choose **View the actions that are allowed by this permission** to expand it and display the list.

7. Choose **Next**.
8. In **Step 3: Choose principals that are allowed to access**, review the selected principals, and if required, update any of the following:
  - a. (Optional) To change whether sharing is enabled with principals inside or outside your organization, choose one of the following options:

- To share resources with AWS accounts, IAM users, and IAM roles that are outside of your organization, choose **Allow sharing with external principals**.
  - To restrict resource sharing to only principals in your organization in AWS Organizations, choose **Allow sharing with principals in your organization only**.
- b. For **Principals**, do the following:
- (Optional) To add an organization, organizational unit (OU), or member AWS account inside your organization, turn on **Display organizational structure** to display a tree view of your organization. Then select the check box next to each principal that you want to add.
- Note**  
The **Display organizational structure** toggle appears only if sharing with AWS Organizations is enabled and you are signed in as a principal in the organization's management account.  
You can't use this method to specify an AWS account outside your organization, or an IAM role or IAM user. Instead, you must add these principals by entering their identifiers, which are shown in the text box below the **Display organizational structure** switch. See the next bullet point.
- (Optional) To add a principal by its identifier, choose the principal type from the dropdown list, and then enter the ID or ARN for the principal. Finally, choose **Add**.
- The addition immediately appears in the **Selected principals** list.
- You can then add additional accounts, OUs, or your organization by repeating this step.
- (Optional) To remove a principal, locate it under **Selected principals**, select its check box, and then choose **Deselect**.
9. Choose **Next**.
10. In **Step 4: Review and update**, review the configuration details for your resource share. To change the configuration for any step, choose the link that corresponds to the step you want to go back to, and then make the required changes.
11. Choose **Update resource share** when you're done making changes.

## AWS CLI

### To update a resource share

You can use the following AWS CLI commands to modify a resource share:

- To rename a resource share, or to change whether external principals are allowed, use the command `update-resource-share`. The following example renames the specified resource share and sets it to allow only principals from its organization. You must use the service endpoint for the AWS Region that contains the resource share.

```
$ aws ram update-resource-share \
  --region us-east-1 \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE \
  --name "my-renamed-resource-share" \
  --no-allow-external-principals
{
  "resourceShare": {
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
    "name": "my-renamed-resource-share",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": false,
```

```
    "status": "ACTIVE",  
    "creationTime": 1565295733.282,  
    "lastUpdatedTime": 1565303080.023  
  }  
}
```

- To add a resource to a resource share, use the command `associate-resource-share`. The following example adds a subnet to the specified resource share.

```
$ aws ram associate-resource-share \  
  --region us-east-1 \  
  --resource-arns arn:aws:ec2:us-east-1:123456789012:subnet/  
subnet-0250c25a1f4e15235 \  
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/7ab63972-  
b505-7e2a-420d-6f5d3EXAMPLE  
{  
  "resourceShareAssociations": [  
    {  
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-  
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",  
      "associatedEntity": "arn:aws:ec2:us-east-1:123456789012:subnet/  
subnet-0250c25a1f4e15235",  
      "associationType": "RESOURCE",  
      "status": "ASSOCIATING",  
      "external": false  
    }  
  ]  
}
```

- To add or replace a AWS RAM managed permission for a resource type in a resource share, use the commands `list-permissions` and `associate-resource-share-permission`. You can assign only one permission per resource type in a resource share. If you try to add a permission to a resource type that already has a permission, you must include the `--replace` option or the command fails with an error.

The following example command lists the ARNs for the permissions available for an Amazon Elastic Compute Cloud (Amazon EC2) subnet, and then uses one of those ARNs to replace the currently assigned permission for that resource type in the specified resource share.

```
$ aws ram list-permissions \  
  --resource-type ec2:Subnet  
{  
  "permissions": [  
    {  
      "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionSubnet",  
      "version": "1",  
      "defaultVersion": true,  
      "name": "AWSRAMDefaultPermissionSubnet",  
      "resourceType": "ec2:Subnet",  
      "creationTime": "2020-02-27T11:38:26.727000-08:00",  
      "lastUpdatedTime": "2020-02-27T11:38:26.727000-08:00"  
    }  
  ]  
}  
$ aws ram associate-resource-share-permission \  
  --region us-east-1 \  
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/f1d72a60-  
da19-4765-b4f9-e27b658b15b8 \  
  --permission-arn arn:aws:ram::aws:permission/AWSRAMDefaultPermissionSubnet  
{  
  "returnValue": true  
}
```

- To remove a resource from a resource share, use the command [disassociate-resource-share](#). The following example removes the Amazon EC2 subnet with the specified ARN from the specified resource share.

```
$ aws ram disassociate-resource-share \
  --region us-east-1 \
  --resource-arns arn:aws:ec2:us-east-1:123456789012:subnet/
subnet-0250c25a1f4e15235 \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/7ab63972-
b505-7e2a-420d-6f5d3EXAMPLE
{
  "resourceShareAssociations": [
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/7ab63972-b505-7e2a-420d-6f5d3EXAMPLE",
      "associatedEntity": "arn:aws:ec2:us-east-1:ubnet/subnet-0250c25a1f4e15235",
      "associationType": "RESOURCE",
      "status": "DISASSOCIATING",
      "external": false
    }
  ]
}
```

- To modify the tags attached to a resource share, use the commands [tag-resource](#) and [untag-resource](#). The following example adds the tag `project=lima` to the specified resource share.

```
$ aws ram tag-resource \
  --region us-east-1 \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/f1d72a60-
da19-4765-b4f9-e27b658b15b8 \
  --tags key=project,value=lima
```

The following example removes the tag with a key of `project` from the specified resource share.

```
$ aws ram untag-resource \
  --region us-east-1 \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-share/f1d72a60-
da19-4765-b4f9-e27b658b15b8 \
  --tag-keys=project
```

The tagging commands produce no output when successful.

## Viewing your shared resources in AWS RAM

You can view the list of individual resources that you've shared, across all resource shares. The list helps you to determine which resources you're currently sharing, the number of resource shares that they're included in, and the number of principals that have access to them.

Console

### To view the resources that you're currently sharing

1. Open the [Shared by me : Shared resources](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (`us-east-1`). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. For each shared resource, the following information is available:

- **Resource ID** – The ID of the resource. Choose the ID of a resource to open a new browser tab to view the resource in its native service console.
- **Resource type** – The type of resource.
- **Last share date** – The date on which the resource was last shared.
- **Resource shares** – The number of resource shares that include the resource. To see the list of the resource shares, choose the number.
- **Principals** – The number of principals who can access the resource. Choose the value to view the principals.

## AWS CLI

### To view the resources that you're currently sharing

You can use the [list-resources](#) command with the parameter `--resource-owner` set to `SELF` to display details of the resources that you currently share.

The following example shows the resources that are included in resource shares in the AWS Region (us-east-1) for the calling AWS account. To get the resources that you share in a different Region, use the `--region <region-code>` parameter.

```
$ aws ram list-resources \
  --region us-east-1 \
  --resource-owner SELF
{
  "resources": [
    {
      "arn": "arn:aws:license-manager:us-east-1:123456789012:license-configuration:lic-ecbd5574fd92cb0d312baea260e4cece",
      "type": "license-manager:LicenseConfiguration",
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-share/818d71dd-7512-4f71-99c6-2ae57aa010bc",
      "creationTime": "2021-09-14T20:42:40.266000-07:00",
      "lastUpdatedTime": "2021-09-14T20:42:41.081000-07:00"
    },
    {
      "arn": "arn:aws:license-manager:us-east-1:123456789012:license-configuration:lic-ecbd5574fd92cb0d312baea260e4cece",
      "type": "license-manager:LicenseConfiguration",
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-share/a477f3b2-4001-4dcb-bd54-7c8d23b4f07d",
      "creationTime": "2021-07-22T11:48:11.104000-07:00",
      "lastUpdatedTime": "2021-07-22T11:48:11.971000-07:00"
    }
  ]
}
```

## Viewing the principals you share resources with in AWS RAM

You can view the principals you share your resources with, across all resource shares. Viewing this list of principals helps you determine who has access to your shared resources.

## Console

### To view the principals you're sharing resources with

1. Navigate to the [Shared by me : Principals](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. Apply a filter to find specific principals. You can apply multiple filters to narrow your search. Choose the text box to see a dropdown list of suggested attribute fields. After you choose one, you can choose from the list of available values for that field. You can add other attributes or keywords until you find the resource you want.
4. For each principal in the list, the console displays the following information:
  - **Principal ID** – The ID of the principal. Choose the ID to open a new browser tab to view the principal in its native console.
  - **Resource shares** – The number of resource shares you shared with the specified principal. Choose the number to view the list of resource shares.
  - **Resources** – The number of resources you shared with the principal. Choose the number to view the list of shared resources.

## AWS CLI

### To view the principals you're sharing resources with

You can use the [list-principals](#) command to get a list of the principals you reference in resource shares that you created in the current AWS Region for the calling account.

The following example lists the principals that have access to shares created in the default Region for the calling account. In this example, the principals are the calling account's organization and a separate AWS account, as part of two different resource shares. You must use the service endpoint for the AWS Region that contains the resource share.

```
$ aws ram list-principals \
  --region us-east-1 \
  --resource-owner SELF
{
  "principals": [
    {
      "id": "arn:aws:organizations::123456789012:organization/o-alb2c3dr",
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-share/a477f3b2-4001-4dcb-bd54-7c8d23b4f07d",
      "creationTime": "2021-09-14T20:40:58.532000-07:00",
      "lastUpdatedTime": "2021-09-14T20:40:59.610000-07:00",
      "external": false
    },
    {
      "id": "111111111111",
      "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-share/6405fa7c-0786-4e15-8c9f-8aec02802f18",
      "creationTime": "2021-09-15T15:00:31.601000-07:00",
      "lastUpdatedTime": "2021-09-15T15:14:13.618000-07:00",
      "external": true
    }
  ]
}
```

## Viewing AWS RAM managed permissions

You can view details about AWS RAM managed permissions that are available to assign to resource types in your resource shares. You can identify the managed permissions that are assigned to resource shares. To see these details, use the **Permissions library** in the AWS RAM console.

Console

### To view details about AWS RAM managed permissions

1. Navigate to the [Permissions library](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#). Although all Regions share the same available managed permissions, this affects the list of associated resource shares in step 5.
3. In the **Permissions** list, choose the managed permission for which you want to view details. You can use the search box to filter the list of permissions by entering part of a name or a resource type.
4. (Optional) To change the display preferences, choose the gear icon in the upper right of the **Permissions** panel. You can change the following preferences:
  - **Page size** – The number of resources displayed on each page.
  - **Wrap lines** – Whether to wrap lines in table rows.
  - **Columns** – Whether to display or hide information about the resource type and associated shares.

After you finish setting display preferences, choose **Confirm**.

5. For each permission, the list displays the following information:
  - **Permission name** – The name of the AWS managed permission.
  - **Resource type** – The resource type that is associated with the managed permission.
  - **Associated shares** – The number of resource shares that are associated with the managed permission. If a number appears, then you can choose the number to display a table of resource shares with the following information:
    - **Resource share name** – The name of the resource share that is associated with the managed permission.
    - **Owner** – The AWS account number of the resource share owner.
    - **Allow external principals** – Whether that resource share allows sharing with principals outside the organization in AWS Organizations.
    - **Status** – The current status of the association between the resource share and the managed permission.

You can choose the managed permission's name to display more information about that permission. The details page for a permission displays the following information:

- **Resource type** – The type of AWS resource to which this managed permission applies.
- **Last updated time** – The date and time when the managed permission was last updated.
- **Creation time** – The date and time when the managed permission was created.
- **ARN** – The [Amazon Resource Name \(ARN\)](#) of the managed permission. The ARNs for managed permissions follow this format:



```
arn:aws:ram::aws:permission/  
AWSRAM[DefaultPermission]ShareableResourceType
```

The substring `[DefaultPermission]` is present in the name of only the one managed permission for that resource type that is designated the default.

- **Allowed actions** – The list of AWS service actions that principals are allowed to perform on the associated resource type.

## AWS CLI

### To view details about AWS RAM managed permissions

You can use the [list-permissions](#) command to get a list of the permissions available to use on resource shares in the current AWS Region for the calling account.

```
$ aws ram list-permissions  
{  
  "permissions": [  
    {  
      "arn": "arn:aws:ram::aws:permission/  
AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",  
      "version": "1",  
      "defaultVersion": true,  
      "name":  
"AWSRAMBlankEndEntityCertificateAPICSRPassthroughIssuanceCertificateAuthority",  
      "resourceType": "acm-pca:CertificateAuthority",  
      "creationTime": "2021-06-09T09:22:57.427000-07:00",  
      "lastUpdatedTime": "2021-06-09T09:22:57.427000-07:00"  
    },  
    {  
      "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionAppMesh",  
      "version": "1",  
      "defaultVersion": true,  
      "name": "AWSRAMDefaultPermissionAppMesh",  
      "resourceType": "appmesh:Mesh",  
      "creationTime": "2020-05-12T11:12:54.068000-07:00",  
      "lastUpdatedTime": "2020-05-12T11:12:54.068000-07:00"  
    },  
    ... TRUNCATED FOR BREVITY ... RUN COMMAND TO SEE COMPLETE LIST OF  
    PERMISSIONS ...  
    {  
      "arn": "arn:aws:ram::aws:permission/  
AWSRAMRevokeCertificateCertificateAuthority",  
      "version": "1",  
      "defaultVersion": true,  
      "name": "AWSRAMRevokeCertificateCertificateAuthority",  
      "resourceType": "acm-pca:CertificateAuthority",  
      "creationTime": "2021-06-09T09:23:16.668000-07:00",  
      "lastUpdatedTime": "2021-06-09T09:23:16.668000-07:00"  
    },  
    {  
      "arn": "arn:aws:ram::aws:permission/  
AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",  
      "version": "1",  
      "defaultVersion": true,  
      "name":  
"AWSRAMSubordinateCACertificatePathLen0IssuanceCertificateAuthority",  
      "resourceType": "acm-pca:CertificateAuthority",  
      "creationTime": "2021-06-09T09:23:11.462000-07:00",  
    }  
  ]  
}
```

```
        "lastUpdatedTime": "2021-06-09T09:23:11.462000-07:00"
      }
    ]
  }
}
```

After you find the ARN of a specific permission you're interested in, you can retrieve its details, including the JSON policy text, by running the command [get-permission](#).

```
$ aws ram get-permission \
  --permission-arn arn:aws:ram::aws:permission/AWSRAMDefaultPermissionResourceGroup
{
  "permission": {
    "arn": "arn:aws:ram::aws:permission/AWSRAMDefaultPermissionResourceGroup",
    "version": "1",
    "defaultVersion": true,
    "name": "AWSRAMDefaultPermissionResourceGroup",
    "resourceType": "resource-groups:Group",
    "permission": "{\n\"Effect\": \"Allow\", \"Action\": [\n\"resource-groups:GetGroup\",
      \n\"resource-groups:GetGroupConfiguration\", \"resource-
groups:ListGroupResources\"]\n}",
    "creationTime": 1582832306.525,
    "lastUpdatedTime": 1582832306.525,
    "isResourceTypeDefault": true
  }
}
```

## Updating AWS RAM managed permissions to a newer version

From time to time, AWS updates the managed permissions you can attach to a resource share for a specific resource type. When AWS does this, it creates a new version of the managed permission. Resource shares that include the specified resource type don't automatically use the latest version of the managed permission. You must explicitly update the managed permission for each resource share yourself. This helps ensure you can evaluate the changes before you apply them to your resource shares.

### Console

At this time, you can perform this task only by using the AWS CLI operations for AWS RAM, or their AWS SDK equivalents.

### AWS CLI

#### To update the version of an AWS RAM managed permission

1. Run the command [get-resource-shares](#) with the `--permission-arn` parameter to specify the [Amazon Resource Name \(ARN\)](#) of the managed permission that you want to update. This results in the command returning only those resource shares that use that managed permission.

For example, the following sample command returns details for every resource share that uses the default AWS RAM managed permission for Amazon EC2 capacity reservations.

```
$ aws ram get-resource-shares \
  --resource-owner SELF \
  --permission-arn arn:aws:ram::aws:permission/
AWSRAMDefaultPermissionCapacityReservation
```

The output includes the ARN of every resource share with at least one resource whose access is controlled by that managed permission.

2. For each resource share specified in the previous command, run the command [associate-resource-share-permission](#). Include the `--resource-share-arn` to specify the resource share to update, the `--permission-arn` to specify which AWS RAM managed permission you're updating, and the `--replace` parameter to specify that you want to update the share to use the latest version of that managed permission.

```
$ aws ram associate-resource-share-permission \
  --resource-share-arn < ARN of one of the shares from the output of the previous
  command > \
  --permission-arn arn:aws:ram::aws:permission/
  AWSRAMDefaultPermissionCapacityReservation \
  --replace
```

3. Repeat the command in step 2 for each `ResourceShareArn` you received in the results from the command in step 1.

## Deleting a resource share in AWS RAM

You can delete a resource share at any time. When you delete a resource share, all principals that were associated with the resource share lose access to the shared resources. Deleting a resource share doesn't delete the shared resources.

The deleted resource share remains visible in the AWS RAM console for a short period after deletion, but its status changes to `Deleted`.

### Console

#### To delete a resource share

1. Open the [Shared by me : Resource shares](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (`us-east-1`). For more information about sharing global resources, see [Sharing Regional resources compared to global resources](#) (p. 14).
3. Select the resource share you want to delete.

#### Warning

Be sure to select the correct resource share. You can't recover a resource share after you delete it.

4. Choose **Delete**, then in the confirmation message, choose **Delete**.

### AWS CLI

#### To delete a resource share

You can use the [delete-resource-share](#) command to delete a resource share that you no longer need.

The following example first uses the [get-resource-shares](#) command to get the Amazon Resource Name (ARN) of the resource share that you want to delete. Then it uses [delete-resource-share](#) to delete the specified resource share.

```
$ aws ram get-resource-shares \
  --region us-east-1 \
  --resource-owner SELF
{
```

```
"resourceShares": [
  {
    "resourceShareArn": "arn:aws:ram:us-east-1:123456789012:resource-
share/2ebe77d7-4156-4a93-87a4-228568d04425",
    "name": "MySubnetShare",
    "owningAccountId": "123456789012",
    "allowExternalPrincipals": true,
    "status": "ACTIVE",
    "creationTime": "2021-09-10T15:38:54.449000-07:00",
    "lastUpdatedTime": "2021-09-10T15:38:54.449000-07:00",
    "featureSet": "STANDARD"
  }
]
}
$ aws ram delete-resource-share \
  --region us-east-1 \
  --resource-share-arn arn:aws:ram:us-east-1:123456789012:resource-
share/2ebe77d7-4156-4a93-87a4-228568d04425
{
  "returnValue": true
}
```

## Access AWS resources shared with you

With AWS Resource Access Manager (AWS RAM), you can view the resource shares to which you have been added, the shared resources that you can access, and the AWS accounts that have shared resources with you. You can also leave a resource share when you no longer require access to its shared resources.

### Contents

- [Accepting and rejecting resource share invitations \(p. 32\)](#)
- [Viewing resource shares shared with you \(p. 34\)](#)
- [Viewing resources shared with you \(p. 36\)](#)
- [View principals sharing with you \(p. 37\)](#)
- [Leaving a resource share \(p. 37\)](#)

## Accepting and rejecting resource share invitations

To access shared resources, the owner of the resource share must add you as a principal.

If you're added to the resource share through an AWS account that is in an organization in AWS Organizations, and sharing within the organization is enabled, then you automatically get access to the shared resources without having to accept an invitation.

If you're added to a resource share by one of the following, you receive an invitation to join the resource share:

- An account outside of your organization in AWS Organizations
- An account inside your organization when sharing with AWS Organizations is not enabled

If you receive an invitation to join a resource share, you must accept it to access its shared resources. If you decline the invitation, you can't access the shared resources.

You have seven days to accept an invitation to join a resource share. If you don't accept the invitation within seven days, the invitation expires and is automatically declined.

## Console

### To respond to an invitation to a resource share

1. Navigate to the [Shared with me : Resource shares](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources](#) (p. 14).
3. Review the list of resource shares to which you have been added.

The **Status** column indicates your current participation status for the resource share. The **Pending** status indicates that you have been added to a resource share, but you have not yet accepted or rejected the invitation.

4. To respond to the resource share invitation, select the resource share ID and choose **Accept resource share** to accept the invitation, or **Reject resource share** to decline the invitation. If you reject the invitation, you don't get access to the resources. If you accept the invitation, you gain access to the resources.

## AWS CLI

### To respond to an invitation to a resource share

You can use the following commands to accept or reject invitations to a resource share:

- [get-resource-share-invitations](#)
- [accept-resource-share-invitation](#)
- [reject-resource-share-invitation](#)

1. The following example starts by using the [get-resource-share-invitations](#) command to retrieve a list of all of the invitations available to the user's AWS account. The AWS CLI query parameter lets you restrict the output to only those invitations with its **status** set to **PENDING**. This example shows one invitation from account 111111111111 is currently **PENDING** for the current account 123456789012 in the specified AWS Region.

```
$ aws ram get-resource-share-invitations \
  --region us-east-1 \
  --query 'resourceShareInvitations[?status==`PENDING`]'
{
  "resourceShareInvitations": [
    {
      "resourceShareInvitationArn": "arn:aws:ram:us-east-1:111111111111:resource-share-invitation/3b3bc051-fbf6-4336-8377-06c559dfee49",
      "resourceShareName": "Test TrngAcct Resource Share",
      "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-share/c4506c70-df75-4e6c-ac30-42ca03295a37",
      "senderAccountId": "111111111111",
      "receiverAccountId": "123456789012",
      "invitationTimestamp": "2021-09-21T08:56:24.977000-07:00",
      "status": "PENDING"
    }
  ]
}
```

2. After you find the invitation that you want to accept, make note of the **resourceShareInvitationArn** in the output to use in the next command to accept the invitation.

```
$ aws ram accept-resource-share-invitation \
  --region us-east-1 \
  --resource-share-invitation-arn arn:aws:ram:us-east-1:111111111111:resource-
share-invitation/3b3bc051-fbf6-4336-8377-06c559dfee49
{
  "resourceShareInvitation": {
    "resourceShareInvitationArn": "arn:aws:ram:us-east-1:111111111111:resource-
share-invitation/3b3bc051-fbf6-4336-8377-06c559dfee49",
    "resourceShareName": "Test TrngAcct Resource Share",
    "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-share/
c4506c70-df75-4e6c-ac30-42ca03295a37",
    "senderAccountId": "111111111111",
    "receiverAccountId": "123456789012",
    "invitationTimestamp": "2021-09-21T09:18:24.545000-07:00",
    "status": "ACCEPTED"
  }
}
```

If successful, note that the response shows that the status has changed from `PENDING` to `ACCEPTED`.

If you instead wanted to reject the invitation, run the [reject-resource-share-invitation](#) command, with the same parameters.

```
$ aws ram reject-resource-share-invitation \
  --region us-east-1 \
  --resource-share-invitation-arn arn:aws:ram:us-east-1:111111111111:resource-share-
invitation/3b3bc051-fbf6-4336-8377-06c559dfee49
{
  "resourceShareInvitation": {
    "resourceShareInvitationArn": "arn:aws:ram:us-east-1:111111111111:resource-
share-invitation/3b3bc051-fbf6-4336-8377-06c559dfee49",
    "resourceShareName": "Test TrngAcct Resource Share",
    "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-share/
c4506c70-df75-4e6c-ac30-42ca03295a37",
    "senderAccountId": "111111111111",
    "receiverAccountId": "123456789012",
    "invitationTimestamp": "2021-09-21T09:18:24.545000-07:00",
    "status": "REJECTED"
  }
}
```

## Viewing resource shares shared with you

You can view the resource shares to which you have access. You can see which principals are sharing resources with you and which resources they're sharing.

Console

### To view the resource shares

1. Navigate to the [Shared with me : Resource shares](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (`us-east-1`). For more information about sharing global resources, see [Sharing Regional resources compared to global resources](#) (p. 14).

3. (Optional) Apply a filter to find specific resource shares. You can apply multiple filters to narrow your search. You can type a keyword, such as part of a resource share name to list only those resource shares that include that text in the name. Choose the text box to see a dropdown list of suggested attribute fields. After you choose one, you can choose from the list of available values for that field. You can add other attributes or keywords until you find the resource you want.
4. The AWS RAM console displays the following information:
  - **Name** – The name of the resource share.
  - **ID** – The ID of the resource share. Choose the ID to view the details page for the resource share.
  - **Owner** – The ID of the AWS account that created the resource share.
  - **Status** – The current status of the resource share. Possible values include:
    - **Active** – The resource share is active and available for use.
    - **Deleted** – The resource share is deleted and is no longer available for use.
    - **Pending** – An invitation to accept the resource share is waiting for a response.

## AWS CLI

### To view the resource shares

Use the [get-resource-shares](#) command with the `--resource-owner` parameter set to `OTHER-ACCOUNTS`.

The following example shows the list of resource shares shared in the specified AWS Region with the calling account by other AWS accounts.

```
$ aws ram get-resource-shares \
  --region us-east-1 \
  --resource-owner OTHER-ACCOUNTS
{
  "resourceShares": [
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-
share/8b831ba0-63df-4608-be3c-19096blee16e",
      "name": "Prod Env Shared Licenses",
      "owningAccountId": "111111111111",
      "allowExternalPrincipals": true,
      "status": "ACTIVE",
      "creationTime": "2021-09-21T08:50:41.308000-07:00",
      "lastUpdatedTime": "2021-09-21T08:50:41.308000-07:00",
      "featureSet": "STANDARD"
    },
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:222222222222:resource-share/
c4506c70-df75-4e6c-ac30-42ca03295a37",
      "name": "Prod Env Shared Subnets",
      "owningAccountId": "222222222222",
      "allowExternalPrincipals": true,
      "status": "ACTIVE",
      "creationTime": "2021-09-21T08:56:24.737000-07:00",
      "lastUpdatedTime": "2021-09-21T08:56:24.737000-07:00",
      "featureSet": "STANDARD"
    }
  ]
}
```

## Viewing resources shared with you

You can view the shared resources that you can access. You can see which principals shared the resources with you and which resource shares include the resources.

### Console

#### To view resources shared with you

1. Navigate to the [Shared with me : Shared resources](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. Apply a filter to find specific shared resources. You can apply multiple filters to narrow your search.
4. The following information is available:
  - **Resource ID** – The ID of the resource. Choose the ID of the resource to view it in its service console.
  - **Resource type** – The type of resource.
  - **Last share date** – The date on which the resource was shared with you.
  - **Resource shares** – The number of resource shares in which the resource is included. Choose the value to view the resource shares.
  - **Owner ID** – The ID of the principal who owns the resource.

### AWS CLI

#### To view resources shared with you

You can use the [list-resources](#) command to view resources that are shared with you.

The following example command displays details about the resource accessible through a resource share in the specified AWS Region from another AWS account.

```
$ aws ram list-resources \
  --region us-east-1 \
  --resource-owner OTHER-ACCOUNTS
{
  "resources": [
    {
      "arn": "arn:aws:license-manager:us-east-1:111111111111:license-configuration:lic-36be0485f5ae379cc74cf8e9242ab143",
      "type": "license-manager:LicenseConfiguration",
      "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-share/8b831ba0-63df-4608-be3c-19096b1ee16e",
      "status": "AVAILABLE",
      "creationTime": "2021-09-21T08:50:41.308000-07:00",
      "lastUpdatedTime": "2021-09-21T08:50:42.517000-07:00"
    }
  ]
}
```



## View principals sharing with you

You can view a list of all the principals that are sharing resources with you. You can see which resources and resource shares they're sharing with you.

### Console

#### To view the principals that are sharing resources with you

1. Open the AWS RAM console at <https://console.aws.amazon.com/ram>.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources](#) (p. 14).
3. In the navigation pane, choose **Shared with me, Principals**.
4. (Optional) You can apply a filter to find specific principals. You can apply multiple filters to narrow your search.
5. The console displays the following information:
  - **Principal ID** – The ID of the principal who is sharing with you.
  - **Resource shares** – The number of resource shares to which the principal has added you. Choose the number to view the list of resource shares.
  - **Resources** – The number of resources the principal is sharing with you. Choose the value to view the list of resources.

### AWS CLI

#### To view the principals that are sharing resources with you

You can use the `list-principals` command to retrieve the list of principals that are sharing resources with your AWS account.

The following example command displays details about the AWS account that shared a resource share with the account used to call the operation in the specified AWS Region.

```
$ aws ram list-principals \
  --region us-east-1 \
  --resource-owner OTHER-ACCOUNTS
{
  "principals": [
    {
      "id": "111111111111",
      "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-
share/8b831ba0-63df-4608-be3c-19096b1ee16e",
      "creationTime": "2021-09-21T08:50:41.308000-07:00",
      "lastUpdatedTime": "2021-09-21T09:06:25.545000-07:00",
      "external": true
    }
  ]
}
```

## Leaving a resource share

If you no longer need access to resources that are shared with you, you can leave a resource share at any time. When you leave a resource share, you lose access to the shared resources.

## Prerequisites for leaving a resource share

- You can leave a resource share only if it was shared with you as an AWS account not in the context of an organization. You can't leave a resource share if you were added to it by an AWS account inside your organization and sharing with AWS Organizations is enabled. Access to resource shares within an organization is automatic.
- To leave a resource share, verify that the resource share is either empty or that it contains only resource types that support leaving a share.

The following resource types **do not** support leaving a resource share. If the resource share contains one or more of these, ask the owner of the resource share to remove your principal from those with permission to the share.

Service	Resource type
AWS App Mesh	appmesh:Mesh
AWS Certificate Manager Private Certificate Authority	acm-pca:CertificateAuthority
AWS CodeBuild	codebuild:Project codebuild:ReportGroup
EC2 Image Builder	imagebuilder:Component imagebuilder:ContainerRecipe imagebuilder:Image imagebuilder:ImageRecipe
AWS Glue	glue:Catalog glue:Database glue:Table
AWS Network Firewall	network-firewall:FirewallPolicy network-firewall:StatefulRuleGroup network-firewall:StatelessRuleGroup
AWS Resource Groups	resource-groups:Group
Amazon Route 53	route53resolver:FirewallRuleGroup route53resolver:ResolverQueryLogConfig
AWS Systems Manager Incident Manager	ssm-contacts:Contact ssm-incidents:ResponsePlan

## How to leave a resource share

### Console

#### To leave a resource share

1. Navigate to the [Shared with me : Resource shares](#) page in the AWS RAM console.
2. Because AWS RAM resource shares exist in specific AWS Regions, choose the appropriate AWS Region from the dropdown list in the upper-right corner of the console. To see resource shares that contain global resources, you must set the AWS Region to US East (N. Virginia), (us-east-1). For more information about sharing global resources, see [Sharing Regional resources compared to global resources \(p. 14\)](#).
3. Select the resource share you want to leave.
4. Choose **Leave resource share**, and in the confirmation dialog box, choose **Leave**.

### AWS CLI

#### To leave a resource share

You can use the [disassociate-resource-share](#) command to leave a resource share.

The following example commands causes the AWS account that calls the command to lose access to the resources shared by the resource share specified by the ARN. You must direct the request to the service endpoint in the AWS Region that contains the resource share that you want to leave.

1. First, retrieve the list of resource shares to retrieve the ARN of the resource share that you want to leave.

```
$ aws ram get-resource-shares \
  --region us-east-1 \
  --resource-owner OTHER-ACCOUNTS
{
  "resourceShares": [
    {
      "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-
share/8b831ba0-63df-4608-be3c-19096b1ee16e",
      "name": "Prod Environment Shared Licenses",
      "owningAccountId": "111111111111",
      "allowExternalPrincipals": true,
      "status": "ACTIVE",
      "creationTime": "2021-09-21T08:50:41.308000-07:00",
      "lastUpdatedTime": "2021-09-21T08:50:41.308000-07:00",
      "featureSet": "STANDARD"
    }
  ]
}
```

2. Then, you can run the command to leave that resource share. Note that you must also specify your account ID, 123456789012, as the principal to disassociate from the specified resource share, which is shared by account 111111111111.

```
$ aws ram disassociate-resource-share \
  --region us-east-1 \
  --resource-share-arn arn:aws:ram:us-east-1:111111111111:resource-
share/8b831ba0-63df-4608-be3c-19096b1ee16e \
  --principals 123456789012
{
  "resourceShareAssociations": [
    {

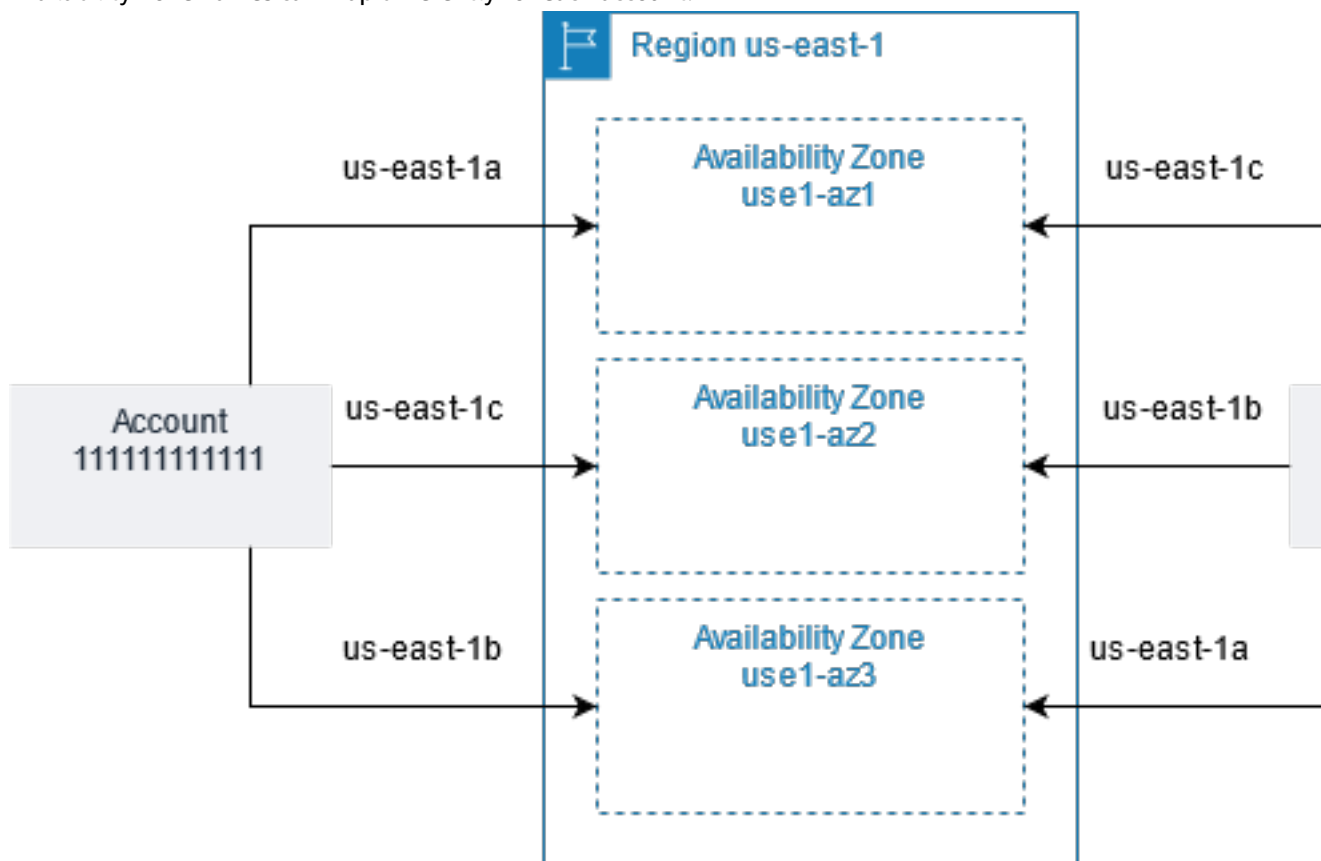
```

```
    "resourceShareArn": "arn:aws:ram:us-east-1:111111111111:resource-  
share/8b831ba0-63df-4608-be3c-19096b1ee16e",  
    "associatedEntity": "123456789012",  
    "associationType": "PRINCIPAL",  
    "status": "DISASSOCIATING",  
    "external": false  
  }  
]  
}
```

## Availability Zone IDs for your AWS resources

AWS maps the physical Availability Zones *randomly* to the available zone names for each AWS account. This approach helps to distribute resources across the Availability Zones in an AWS Region, instead of resources likely being concentrated in Availability Zone "a" for each Region. As a result, the Availability Zone `us-east-1a` for *your* AWS account might not represent the same physical location as `us-east-1a` for a different AWS account. For more information, see [Regions and Availability Zones](#) in the *Amazon EC2 User Guide*.

The following illustration shows how the AZ IDs are the same for every account even though the Availability Zone names can map differently for each account.



For some resources, you must identify not only the AWS Region, but also the Availability Zone. For example, an Amazon VPC subnet. Within a single account, the mapping of an Availability Zone to a specific name isn't important. But, when you use AWS RAM to share such a resource with other AWS accounts, the mapping *is* important. This random mapping complicates the ability of the account accessing the shared resource to know which Availability Zone to reference. To help with this, such

resources also allow you to identify the actual location of your resources relative to your accounts by using the *AZ ID*. An AZ ID is a unique and consistent identifier for an Availability Zone across all AWS accounts. For example, `use1-az1` is an AZ ID for an Availability Zone in the `us-east-1` Region and it represents the same physical location in every AWS account.

You can use AZ IDs to determine the location of resources in one account relative to the resources in another account. For example, if you share a subnet in the Availability Zone with the AZ ID `use1-az2` with another account, this subnet is available to that account in the Availability Zone whose AZ ID is also `use1-az2`. The AZ ID for each subnet is displayed in the Amazon VPC console, and can be queried using the AWS CLI.

#### Console

##### To view the AZ IDs for the Availability Zones in your account

1. Navigate to the [AWS RAM console home](#) page in the AWS RAM console.
2. You can view the AZ IDs for the current AWS Region under **Your AZ ID**.

#### AWS CLI

##### To view the AZ IDs for the Availability Zones in your account

The following example command shows the AZ IDs for the Availability Zones in the `us-west-2` Region and how they are mapped for the calling AWS account.

```
$ aws ec2 describe-availability-zones \
  --region us-west-2
{
  "AvailabilityZones": [
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2a",
      "ZoneId": "usw2-az2",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2",
      "ZoneType": "availability-zone"
    },
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2b",
      "ZoneId": "usw2-az1",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2",
      "ZoneType": "availability-zone"
    },
    {
      "State": "available",
      "OptInStatus": "opt-in-not-required",
      "Messages": [],
      "RegionName": "us-west-2",
      "ZoneName": "us-west-2c",
      "ZoneId": "usw2-az3",
      "GroupName": "us-west-2",
      "NetworkBorderGroup": "us-west-2",
      "ZoneType": "availability-zone"
    }
  ],
}
```

```
{
  "State": "available",
  "OptInStatus": "opt-in-not-required",
  "Messages": [],
  "RegionName": "us-west-2",
  "ZoneName": "us-west-2d",
  "ZoneId": "usw2-az4",
  "GroupName": "us-west-2",
  "NetworkBorderGroup": "us-west-2",
  "ZoneType": "availability-zone"
}
```

# Shareable AWS resources

With AWS Resource Access Manager (AWS RAM), you can share resources that are created and managed by other AWS services. You can share resources with individual AWS accounts. You can also share resources with the accounts in an organization or organizational units (OUs) in AWS Organizations. Some supported resource types also let you share resources with individual AWS Identity and Access Management (IAM) roles and users.

The following sections list the services that work with AWS RAM, and the resources that support sharing. Also identified are which resource types can be shared with individual IAM users and roles.

## Services with resources that you can share by using AWS RAM

- [AWS App Mesh \(p. 43\)](#)
- [Amazon Aurora \(p. 44\)](#)
- [AWS Certificate Manager Private Certificate Authority \(p. 44\)](#)
- [AWS CodeBuild \(p. 45\)](#)
- [Amazon EC2 \(p. 45\)](#)
- [EC2 Image Builder \(p. 46\)](#)
- [AWS Glue \(p. 47\)](#)
- [AWS License Manager \(p. 48\)](#)
- [AWS Migration Hub Refactor Spaces \(p. 49\)](#)
- [AWS Network Firewall \(p. 49\)](#)
- [AWS Outposts \(p. 50\)](#)
- [Amazon S3 on Outposts \(p. 51\)](#)
- [AWS Resource Groups \(p. 51\)](#)
- [Amazon Route 53 \(p. 52\)](#)
- [Amazon SageMaker \(p. 53\)](#)
- [AWS Systems Manager Incident Manager \(p. 53\)](#)
- [Amazon VPC \(p. 54\)](#)
- [AWS Cloud WAN \(p. 57\)](#)

## AWS App Mesh



You can share the following AWS App Mesh resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Mesh <code>appmesh:Mesh</code>	Create and manage a mesh centrally, and share it with other AWS accounts or your organization. A shared mesh allows resources created by different AWS accounts to communicate with each other in the same mesh. For more	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
	information, see <a href="#">Working with shared meshes</a> in the <i>AWS App Mesh User Guide</i> .		



## Amazon Aurora

You can share the following Amazon Aurora resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
DB clusters <code>rds:Cluster</code>	Create and manage a DB cluster centrally, and share it with other AWS accounts or your organization. This lets multiple AWS accounts clone a shared, centrally managed DB cluster. For more information, see <a href="#">Cross-account cloning with AWS RAM and Amazon Aurora</a> in the <i>Amazon Aurora User Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.

## AWS Certificate Manager Private Certificate Authority

You can share the following ACM Private CA resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Private certificate authority (CA) <code>acm-pca:CertificateAuthority</code>	Create and manage private certificate authorities (CAs) for your organization's internal public key infrastructure (PKI), and share those CAs with other AWS accounts or your organization. This lets AWS Certificate Manager users in other accounts issue X.509 certificates signed by your shared CA. For more information, see <a href="#">Controlling access to a private CA</a> in the <i>AWS Certificate Manager Private Certificate Authority User Guide</i> .	 Yes	 Yes  Can share with <b>any</b> AWS account.



## AWS CodeBuild



You can share the following AWS CodeBuild resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Project <code>codebuild:Project</code>	Create a project, and use it to run builds. Share the project with other AWS accounts or your organization. This lets multiple AWS accounts and users view information about a project and analyze its builds. For more information, see <a href="#">Working with shared projects</a> in the <i>AWS CodeBuild User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.
Report group <code>codebuild:ReportGroup</code>	Create a report group, and use it to create reports when you build a project. Share the report group with other AWS accounts or your organization. This lets multiple AWS accounts and users view the report group and its reports, and the test case results for each report. A report can be viewed for 30 days after it's created, and then it expires and is no longer available to view. For more information, see <a href="#">Working with shared projects</a> in the <i>AWS CodeBuild User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.

## Amazon EC2

You can share the following Amazon EC2 resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Capacity reservations <code>ec2:CapacityReservation</code>	Create and manage capacity reservations centrally, and share the reserved capacity with other AWS accounts or your organization. This lets multiple AWS accounts launch their Amazon EC2 instances into centrally managed reserved capacity. For more information, see <a href="#">Working with shared Capacity</a>	✘ No	✔ Yes  Can share with <b>any</b> AWS account.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
	<p><a href="#">Reservations</a> in the <i>Amazon EC2 User Guide for Linux Instances</i>.</p> <p><b>Important</b> If you don't meet all of the <a href="#">prerequisites for sharing a capacity reservation</a>, then the sharing operation can fail. If this happens and a user attempts to launch an Amazon EC2 instance into that capacity reservation, it launches as an on-demand instance that can accrue higher costs. We recommend that you verify that you can access the shared capacity reservation by attempting to <a href="#">view it in the Amazon EC2 console</a>. You can also monitor for failed resource shares so that you can take corrective action before users launch instances in ways that raise your costs. For more information, see <a href="#">Example: Alerting on resource share failures</a> (p. 74).</p>		
<p>Dedicated hosts</p> <p><code>ec2:DedicatedHost</code></p>	<p>Allocate and manage Amazon EC2 dedicated hosts centrally, and share the host's instance capacity with other AWS accounts or your organization. This lets multiple AWS accounts launch their Amazon EC2 instances on to centrally managed dedicated hosts. For more information, see <a href="#">Working with shared Dedicated Hosts</a> in the <i>Amazon EC2 User Guide for Linux Instances</i>.</p>	<p> No</p>	<p> Yes</p> <p>Can share with <b>any</b> AWS account.</p>







## EC2 Image Builder

You can share the following EC2 Image Builder resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Components <code>imagebuilder:Component</code>	Create and manage components centrally, and share them with other AWS accounts or your organization. Manage who can use predefined build and test components in their image recipes. For more information, see <a href="#">Share EC2 Image Builder resources</a> in the <i>EC2 Image Builder User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.
Container recipes <code>imagebuilder:ContainerRecipe</code>	Create and manage your container recipes centrally, and share them with other AWS accounts or your organization. This allows you to manage who can use predefined documents to duplicate container image builds. For more information, see <a href="#">Share EC2 Image Builder resources</a> in the <i>EC2 Image Builder User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.
Images <code>imagebuilder:Image</code>	Create and manage your golden images centrally, and share them with other AWS accounts or your organization. Manage who can use images created with EC2 Image Builder across your organization. For more information, see <a href="#">Share EC2 Image Builder resources</a> in the <i>EC2 Image Builder User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.
Image recipes <code>imagebuilder:ImageRecipe</code>	Create and manage your image recipes centrally, and share them with other AWS accounts or your organization. This allows you to manage who can use predefined documents to duplicate AMI builds. For more information, see <a href="#">Share EC2 Image Builder resources</a> in the <i>EC2 Image Builder User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.



## AWS Glue

You can share the following AWS Glue resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Data catalogs <code>glue:Catalog</code>	Manage a central data catalog, and share metadata about databases and tables with AWS accounts or your organization. This enables users to run queries on data across multiple accounts. For more information, see <a href="#">Sharing Data Catalog Tables and Databases Across AWS Accounts</a> in the <i>AWS Lake Formation Developer Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.
Databases <code>glue:Database</code>	Create and manage data catalog databases centrally, and share them with AWS accounts or your organization. Databases are collections of data catalog tables. This enables users to run queries and extract, transform, and load (ETL) jobs that can join and query data across multiple accounts. For more information, see <a href="#">Sharing Data Catalog Tables and Databases Across AWS Accounts</a> in the <i>AWS Lake Formation Developer Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.
Tables <code>glue:Table</code>	Create and manage data catalog tables centrally, and share them with AWS accounts or your organization. Data catalog tables contain metadata about data tables in Amazon S3, JDBC data sources, Amazon Redshift, streaming sources, and other data stores. This enables users to run queries and ETL jobs that can join and query data across multiple accounts. For more information, see <a href="#">Sharing Data Catalog Tables and Databases Across AWS Accounts</a> in the <i>AWS Lake Formation Developer Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.



## AWS License Manager

You can share the following AWS License Manager resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
License configurations <code>license-manager:LicenseConfiguration</code>	Create and manage license configurations centrally, and share them with other AWS accounts or your organization. This lets you enforce centrally managed licensing rules that are based on the terms of your enterprise agreements across multiple AWS accounts. For more information, see <a href="#">License configurations in License Manager</a> in the <i>License Manager User Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.



## AWS Migration Hub Refactor Spaces

You can share the following Migration Hub Refactor Spaces resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Refactor Spaces Environment <code>refactor-spaces:Environment</code>	Create a Refactor Spaces environment, and use it to contain your Refactor Spaces applications. Share the environment with other AWS accounts or all of the accounts in your organization. This lets multiple AWS accounts and users view information about the environment and the applications in it. For more information, see <a href="#">Sharing Refactor Spaces environments using AWS RAM</a> in the <i>AWS Migration Hub Refactor Spaces User Guide</i> .	 Yes	 Yes  Can share with <b>any</b> AWS account.

## AWS Network Firewall

You can share the following AWS Network Firewall resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Firewall policies	Create and manage firewall policies centrally, and share them with	 Yes	 Yes

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
network-firewall:FirewallPolicy	other AWS accounts or your organization. This enables multiple accounts in an organization to share a common set of network monitoring, protection, and filtering behaviors. For more information, see <a href="#">Sharing firewall policies and rule groups</a> in the <i>AWS Network Firewall Developer Guide</i> .		Can share with <b>any</b> AWS account.
Rule groups network-firewall:StatefulRuleGroup network-firewall:StatelessRuleGroup	Create and manage stateless and stateful rule groups centrally, and share them with other AWS accounts or your organization. This enables multiple accounts in an organization in AWS Organizations to share a set of criteria for inspecting and handling network traffic. For more information, see <a href="#">Sharing firewall policies and rule groups</a> in the <i>AWS Network Firewall Developer Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.

## AWS Outposts

You can share the following AWS Outposts resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Outposts outposts:Outpost	Create and manage Outposts centrally, and share them with other AWS accounts in your organization. This lets multiple accounts create subnets and EBS volumes on your shared, centrally managed Outposts. For more information, see <a href="#">Working with shared AWS Outposts resources</a> in the <i>AWS Outposts User Guide</i> .	✘ No	✘ No  Can share with <b>only</b> AWS accounts in its own organization.
Local gateway route table ec2:LocalGatewayRouteTable	Create and manage VPC associations to a local gateway centrally, and share them with other AWS accounts in your organization. This lets multiple accounts create VPC associations to a local gateway,	✘ No	✘ No  Can share with <b>only</b> AWS accounts

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
	and view route table and virtual interface configuration. For more information, see <a href="#">Shareable Outpost resources</a> in the <i>AWS Outposts User Guide</i> .		in its own organization.
Sites outposts:Site	Create and manage Outpost sites and share them with other AWS accounts in your organization. This lets multiple accounts create and manage Outposts at the shared site and supports split control between the Outpost resources and the site. For more information, see <a href="#">Working with shared AWS Outposts resources</a> in the <i>AWS Outposts User Guide</i> .	⊗ No	✔ Yes  Can share with <b>any</b> AWS account.



## Amazon S3 on Outposts

You can share the following Amazon S3 on Outposts resource by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
S3 on Outpost s3-outposts:Outpost	Create and manage Amazon S3 buckets, access points, and endpoints on the Outpost. This lets multiple accounts create and manage Outposts at the shared site and supports split control between the Outpost resources and the site. For more information, see <a href="#">Working with shared AWS Outposts resources</a> in the <i>AWS Outposts User Guide</i> .	⊗ No	⊗ No  Can share with <b>only</b> AWS accounts in its own organization.





## AWS Resource Groups

You can share the following AWS Resource Groups resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Resource groups <code>resource-groups:Group</code>	Create and manage a host resource group centrally, and share it with other AWS accounts in your organization. This lets multiple AWS accounts share a group of Amazon EC2 Dedicated Hosts created using AWS License Manager. For more information, see <a href="#">Host resource groups in AWS License Manager</a> in the <i>AWS License Manager User Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.

## Amazon Route 53

You can share the following Amazon Route 53 resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Route 53 Resolver DNS Firewall rule groups <code>route53resolver:FirewallRuleGroup</code>	Create and manage Route 53 Resolver DNS Firewall rule groups centrally, and share them with other AWS accounts or your organization. This enables multiple accounts to share a set of criteria for inspecting and handling outbound DNS queries that go through Route 53 Resolver. For more information, see <a href="#">Sharing Route 53 Resolver DNS Firewall rule groups between AWS accounts</a> in the <i>Amazon Route 53 Developer Guide</i> .	 Yes	 Yes  Can share with <b>any</b> AWS account.
Forwarding rules <code>route53resolver:ResolverRule</code>	Create and manage forwarding rules centrally, and share them with other AWS accounts or your organization. This lets multiple accounts forward DNS queries from their virtual private clouds (VPCs) to the target IP addresses defined in shared, centrally managed resolver rules. For more information, see <a href="#">Sharing forwarding rules with other AWS accounts and using shared rules</a> in the <i>Amazon Route 53 Developer Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.



Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Query logs <code>route53resolver:ResolverQueryLogConfig</code>	Create and manage query logs centrally, and share them with other AWS accounts in your organization. This enables multiple AWS accounts to log DNS queries that originate in their VPCs to a centrally managed query log. For more information, see <a href="#">Sharing Resolver query logging configurations with other AWS accounts</a> in the <i>Amazon Route 53 Developer Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.

## Amazon SageMaker

You can share the following Amazon SageMaker resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Lineage group <code>sagemaker:LineageGroup</code>	Amazon SageMaker lets you create lineage groups of your pipeline metadata to get a deeper understanding of its history and relationships. Share the lineage group with other AWS accounts or the accounts in your organization. This lets multiple AWS accounts and users view information about the lineage group and query the tracking entities within it. For more information, see <a href="#">Cross-Account Lineage Tracking</a> in the <i>Amazon SageMaker Developer Guide</i> .	✘ No	✔ Yes  Can share with <b>any</b> AWS account.

## AWS Systems Manager Incident Manager

You can share the following AWS Systems Manager Incident Manager resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Contacts	Create and manage contacts and escalation plans centrally, and share	✔ Yes	✔ Yes





Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
ssm-contacts:Contact	the contact details with other AWS accounts or your organization. This lets many AWS accounts view engagements occurring during an incident. For more information, see <a href="#">Working with shared contacts and response plans</a> in the <i>AWS Systems Manager Incident Manager User Guide</i> .		Can share with <b>any</b> AWS account.
Response plans ssm-incidents:ResponsePlan	Create and manage response plans centrally, and share them with other AWS accounts or your organization. This lets those AWS accounts connect Amazon CloudWatch alarms and Amazon EventBridge event rules to response plans, automatically creating an incident when it's detected. The incident also has access to the metrics of these other AWS accounts. For more information, see <a href="#">Working with shared contacts and response plans</a> in the <i>AWS Systems Manager Incident Manager User Guide</i> .	✔ Yes	✔ Yes  Can share with <b>any</b> AWS account.





## Amazon VPC

You can share the following Amazon Virtual Private Cloud (Amazon VPC) resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Customer-owned IPv4 addresses ec2:CoipPool	During the AWS Outposts installation process, AWS creates an address pool, known as a <i>customer-owned IP address pool</i> , based on information that you provide about your on-premises network.  Customer-owned IP addresses provide local, or external connectivity to resources in your Outposts subnets through your on-premises network. You can assign these addresses to resources on your Outpost, such as EC2	✘ No	✘ No  Can share with <b>only</b> AWS accounts in its own organization.



Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
	instances, using Elastic IP addresses or using the subnet setting that automatically assigns customer-owned IP addresses. For more information, see <a href="#">Customer-owned IP addresses</a> in the <i>AWS Outposts User Guide</i> .		
IP Address Manager (IPAM) ec2:IpamPool	Share IPAM pools centrally with other AWS accounts, IAM roles or users, or an entire organization or organizational unit (OU) in AWS Organizations. This lets those principals allocate CIDRs from the pool to AWS resources, such as VPCs, in their respective accounts. For more information, see <a href="#">Share an IPAM pool using AWS RAM</a> in the <i>Amazon VPC IP Address Manager User Guide</i> . For more information, see <a href="#">Work with VPCs and subnets</a> in the <i>Amazon VPC User Guide</i> .	✔ Yes	✘ No  Can share with <b>only</b> AWS accounts in its own organization.
Prefix lists ec2:PrefixList	Create and manage prefix lists centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts reference prefix lists in their resources, such as VPC security groups and subnet route tables. For more information, see <a href="#">Working with shared prefix lists</a> in the <i>Amazon VPC User Guide</i> .	✘ No	✔ Yes  Can share with <b>any</b> AWS account.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Subnets <code>ec2:Subnet</code>	<p>Create and manage subnets centrally, and share them with AWS accounts within your organization. This lets multiple AWS accounts launch their application resources into centrally managed VPCs. These resources include Amazon EC2 instances, Amazon Relational Database Service (RDS) databases, Amazon Redshift clusters, and AWS Lambda functions. For more information, see <a href="#">Working with VPC sharing</a> in the <i>Amazon VPC User Guide</i>.</p> <p><b>Note</b> To include a subnet when you create a resource share, you must have the <code>ec2:DescribeSubnets</code> and <code>ec2:DescribeVpcs</code> permissions, in addition to <code>ram:CreateResourceShare</code>.</p>	 No	<p> No</p> <p>Can share with <b>only</b> AWS accounts in its own organization.</p>
Traffic mirror targets <code>ec2:TrafficMirrorTarget</code>	<p>Create and manage traffic mirror targets centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts send mirrored network traffic from traffic mirror sources in their accounts to a shared, centrally managed traffic mirror target. For more information, see <a href="#">Cross-account traffic mirroring targets</a> in the <i>Traffic Mirroring Guide</i>.</p>	 No	<p> Yes</p> <p>Can share with <b>any</b> AWS account.</p>

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Transit gateways <code>ec2:TransitGateway</code>	Create and manage transit gateways centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts route traffic between their VPCs and on-premises networks through a shared, centrally managed transit gateway. For more information, see <a href="#">Sharing a transit gateway</a> in the <i>Amazon VPC Transit Gateways</i> .  <b>Note</b> To include a transit gateway when you create a resource share, you must have the <code>ec2:DescribeTransitGateway</code> permission in addition to <code>ram:CreateResourceShare</code> .	 No	 Yes  Can share with <b>any</b> AWS account.
Transit gateway multicast domains <code>ec2:TransitGatewayMulticastDomain</code>	Create and manage transit gateway multicast domains centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts register and deregister group members or group sources in the multicast domain. For more information, see <a href="#">Working with shared multicast domains</a> in the <i>Transit Gateways Guide</i> .	 No	 Yes  Can share with <b>any</b> AWS account.

## AWS Cloud WAN

You can share the following AWS Cloud WAN resources by using AWS RAM.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
Cloud WAN core network <code>networkmanager:CoreNetwork</code>	Create and manage a Cloud WAN core network centrally, and share it with other AWS accounts. This lets multiple AWS accounts access and provision hosts on a single Cloud WAN core network. For more information, see <a href="#">Share a core</a>	 Yes	 Yes  Can share with <b>any</b> AWS account.

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization
	<a href="#">network</a> in the <i>AWS Cloud WAN User Guide</i> .		

# AWS RAM managed permissions

For each shareable resource type, there is at least one AWS Resource Access Manager (AWS RAM) managed permission that defines the actions that principals with access to the resources in a resource share are allowed to perform on those resources. Some resource types have only one AWS RAM managed permission, and it's used automatically by default, with no action required by you. Some resource types define more than one managed permission, and you can choose which one to use in a resource share.

You can retrieve the list of the available managed permissions at any time. For more information, see [Viewing AWS RAM managed permissions \(p. 28\)](#).

## Topics

- [How AWS RAM managed permissions work \(p. 59\)](#)
- [Types of AWS RAM managed permissions \(p. 59\)](#)

## How AWS RAM managed permissions work

AWS RAM managed permissions are similar to [IAM resource-based policies](#). When you create a resource share, you associate a AWS RAM managed permission with each resource type that you want to share.

After you create the resource share, AWS RAM provides the managed permission that you associate with each resource type to the respective resource-owning service, such as AWS Certificate Manager Private Certificate Authority. The permissions are then attached to each of the resources in the resource share.

AWS RAM managed permissions specify the following:

### Effect

Indicates whether to `Allow` or `Deny` the principal permission to perform an operation on a shared resource. For an AWS RAM managed permission, the effect is always `Allow`. For more information, see [Effect](#) in the *IAM User Guide*.

### Principal

The ID number of an AWS account, or the [Amazon Resource Name \(ARN\)](#) of an organization or organizational unit (OU) in AWS Organizations, or the Amazon Resource Name (ARN) of an AWS Identity and Access Management (IAM) role or user to whom you want to grant access the shared resource. For more information, see [Principal](#) in the *IAM User Guide*.

### Note

Not all resource types can be shared with IAM roles and users. For information about resources that you can share with these principals, see [Shareable AWS resources \(p. 43\)](#).

### Action

The operation that the principal is granted permission to perform. This can be an action in the AWS Management Console or an operation in the AWS Command Line Interface (AWS CLI) or AWS API. The actions are defined by the AWS RAM permission. For more information, see [Action](#) in the *IAM User Guide*.

## Types of AWS RAM managed permissions

When you create a resource share, you choose the AWS RAM permission to associate with each resource type that you include in the resource share. Managed permissions are defined by the resource-owning service and managed by AWS RAM.

- **Default managed permission** – There is one default managed permission available for each resource type that AWS RAM supports. The default managed permission allows principals to perform specific actions that are defined by the service for the resource type. For example, for the Amazon VPC `ec2:Subnet` resource type, the default managed permission allows principals to perform the following actions:

- `ec2:RunInstances`
- `ec2:CreateNetworkInterface`
- `ec2:DescribeSubnets`

The names of default managed permissions use the following format:

`AWSRAMDefaultPermission`*ShareableResourceType*. For example, for the `ec2:Subnet` resource type, the name of the default AWS RAM managed permission is `AWSRAMDefaultPermissionSubnet`.

- **Additional managed permissions** – Some resource types support additional choices for the permission you can attach to a resource type in a resource share. Examples include read-only access or full access (Read and Write access). These additional managed permissions provide you with more flexibility to choose the permissions to grant to specific principals for supported resource types. For example, when you share a resource type that supports both a full access (Read and Write) managed permission and a read-only managed permission, you can share the resources with the full access managed permission granted to an administrator. You can then share the resources with other team members using the read-only managed permission to follow the [security best practice of granting least privilege](#).

#### Note

Currently, only some AWS services that work with AWS RAM support additional managed permissions beyond the default. You can view the available permissions for each AWS service on the [Permissions library](#) page. This page provides details about each available managed permission, including any resource shares that are currently associated with the permission and whether sharing with external principals is allowed, if applicable. For more information, see [Viewing AWS RAM managed permissions \(p. 28\)](#).

For services that don't support additional managed permissions, when you create a resource share, AWS RAM automatically applies the default permission defined for the resource type that you choose.



# Security in AWS RAM

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 AWS Resource Access Manager (AWS RAM), 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 RAM. The following topics show you how to configure AWS RAM to meet your security and compliance objectives. You also learn how to use other AWS services that help you to monitor and secure your AWS RAM resources.

## Topics

- [Data protection in AWS RAM \(p. 61\)](#)
- [Identity and access management for AWS RAM \(p. 62\)](#)
- [Logging and monitoring in AWS RAM \(p. 73\)](#)
- [Resilience in AWS RAM \(p. 76\)](#)
- [Infrastructure security in AWS RAM \(p. 76\)](#)

## Data protection in AWS RAM

The AWS [shared responsibility model](#) applies to data protection in AWS Resource Access Manager. 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. This content includes 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 user accounts with 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 recommend TLS 1.2 or later.
- Set up API and user activity logging with AWS CloudTrail.

- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing personal data that is stored in Amazon S3.
- If you require FIPS 140-2 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see [Federal Information Processing Standard \(FIPS\) 140-2](#).

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form fields such as a **Name** field. This includes when you work with AWS RAM or other AWS services using the console, API, AWS CLI, or AWS SDKs. Any data that you enter into tags or free-form 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.

## Identity and access management for AWS RAM

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 AWS resources. IAM enables you to create users and groups under your AWS account. You control the permissions that users have to perform tasks using AWS resources. You can use IAM for no additional charge. For more information about managing and creating custom IAM policies, see [Managing IAM policies](#).

### Topics

- [How AWS RAM works with IAM \(p. 62\)](#)
- [AWS managed policies for AWS RAM \(p. 64\)](#)
- [Using Service-Linked Roles for AWS RAM \(p. 67\)](#)
- [Example IAM policies for AWS RAM \(p. 68\)](#)
- [Example service control policies for AWS Organizations and AWS RAM \(p. 70\)](#)
- [Disabling resource sharing with AWS Organizations \(p. 73\)](#)

## How AWS RAM works with IAM

By default, IAM users don't have permission to create or modify AWS RAM resources. To allow IAM users to create or modify resources and perform tasks, you must either attach an AWS managed policy or create and attach new IAM policies that grant permission to use specific resources and API actions. You then attach those policies to the IAM users or groups that require those permissions.

AWS RAM provides several AWS managed policies that you can use that will address the needs of many users. For more information about these, see [AWS managed policies for AWS RAM \(p. 64\)](#).

If you need finer control over the permissions you grant to your users, you can construct your own policies in the IAM console. For information about creating policies and attaching them to your IAM users and roles, see [Policies and permissions in IAM](#) in the *AWS Identity and Access Management User Guide*.

The following sections provide the AWS RAM specific details for building an IAM permission policy.

### Contents

- [Policy structure \(p. 63\)](#)
  - [Effect \(p. 63\)](#)
  - [Action \(p. 63\)](#)

- [Resource \(p. 63\)](#)
- [Condition \(p. 63\)](#)

## Policy structure

An IAM policy is a JSON document that includes the following statements: Effect, Action, Resource, and Condition. An IAM policy typically takes the following form.

```
{
  "Statement": [{
    "Effect": "effect",
    "Action": "action",
    "Resource": "arn",
    "Condition": {
      "condition": {
        "key": "value"
      }
    }
  }]
}
```

### Effect

The *Effect* statement indicates whether the policy allows or denies a user permission to perform an action. The possible values include: Allow and Deny.

### Action

The *Action* statement specifies the AWS RAM API actions for which the policy is allowing or denying permission. For a complete list of the allowed actions, see [Actions defined by AWS Resource Access Manager](#) in the *IAM User Guide*.

### Resource

The *Resource* statement specifies the AWS RAM resources that are affected by the policy. To specify a resource in the statement, you need to use its unique Amazon Resource Name (ARN). For a complete list of the allowed resources, see [Resources defined by AWS Resource Access Manager](#) in the *IAM User Guide*.

### Condition

*Condition* statements are optional. They can be used to further refine the conditions under which the policy applies. AWS RAM supports the following condition keys:

- `aws:RequestTag/${TagKey}` – Specifies a tag key and value pair that must be used when creating or tagging a resource share.
- `aws:ResourceTag/${TagKey}` – Indicates that the action can be performed only on resources that have the specified tag key and value pair.
- `aws:TagKeys` – Specifies the tag keys that can be used when creating or tagging a resource share.
- `ram:AllowsExternalPrincipals` – Indicates that the action can be performed only on resource shares that allow or deny sharing with external principals. An external principal is an AWS account outside of your organization in AWS Organizations.
- `ram:Principal` – Indicates that the action can be performed only on the specified principal.
- `ram:RequestedResourceType` – Indicates that the action can be performed only on the specified resource type. You must specify resource types using the format shown in the list of [shareable resource types \(p. 43\)](#).

- `ram:ResourceArn` – Indicates that the action can be performed only on a resource with the specified ARN.
- `ram:ResourceShareName` – Indicates that the action can be performed only on a resource share with the specified name.
- `ram:ShareOwnerAccountId` – Indicates that the action can be performed only on resource shares owned by a specific account.

## AWS managed policies for AWS RAM

AWS Resource Access Manager currently provides several AWS RAM managed policies, which are described in this topic.

### AWS managed policies

- [AWS managed policy: `AWSResourceAccessManagerReadOnlyAccess` \(p. 64\)](#)
- [AWS managed policy: `AWSResourceAccessManagerFullAccess` \(p. 65\)](#)
- [AWS managed policy: `AWSResourceAccessManagerResourceShareParticipantAccess` \(p. 65\)](#)
- [AWS managed policy: `AWSResourceAccessManagerServiceRolePolicy` \(p. 66\)](#)
- [AWS RAM updates to AWS managed policies \(p. 67\)](#)

In the preceding list, you can attach the first three policies to your IAM users and roles to grant permissions. The last policy in the list is reserved for the AWS RAM service's service-linked role.

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

AWS services maintain and update AWS managed policies. You can't change the permissions in AWS managed policies. Services occasionally add additional permissions to an AWS managed policy to support new features. This type of update affects all identities (users, groups, and roles) where the policy is attached. Services are most likely to update an AWS managed policy when a new feature is launched or when new operations become available. Services do not remove permissions from an AWS managed policy, so policy updates won't break your existing permissions.

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

## AWS managed policy: `AWSResourceAccessManagerReadOnlyAccess`

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

This policy provides read-only permissions to the resource shares that are owned by your AWS account.

It does this by granting permission to run any of the `Get*` or `List*` operations. It doesn't provide any ability to modify any resource share.

### Permissions details

This policy includes the following permissions.

- `ram` – Allows principals to view details about resource shares owned by the account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ram:Get*",
        "ram:List*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

## AWS managed policy: `AWSResourceAccessManagerFullAccess`

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

This policy provides full administrative access to view or modify the resource shares that are owned by your AWS account.

It does this by granting permission to run any `ram` operations.

### Permissions details

This policy includes the following permissions.

- `ram` – Allows principals to view or modify any information about the resource shares that are owned by the AWS account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ram:*"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

## AWS managed policy: `AWSResourceAccessManagerResourceShareParticipantAccess`

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

This policy provides principals the ability to accept or reject resource shares that are shared with this AWS account, and to view details about these resource shares. It doesn't provide any ability to modify those resource shares.

It does this by granting permission to run some `ram` operations.

### Permissions details

This policy includes the following permissions.

- **ram** – Allows principals to accept or reject resource share invitations and to view details about the resource shares that are shared with the account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Action": [
        "ram:AcceptResourceShareInvitation",
        "ram:GetResourcePolicies",
        "ram:GetResourceShareInvitations",
        "ram:GetResourceShares",
        "ram:ListPendingInvitationResources",
        "ram:ListPrincipals",
        "ram:ListResources",
        "ram:RejectResourceShareInvitation"
      ],
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

## AWS managed policy: AWSResourceAccessManagerServiceRolePolicy

The AWS managed policy `AWSResourceAccessManagerServiceRolePolicy` can be used only with the service-linked role for AWS RAM. You can't attach, detach, modify, or delete this policy.

This policy provides AWS RAM with read-only access to your organization's structure. When you enable integration between AWS RAM and AWS Organizations, AWS RAM automatically creates a service-linked role named [AWSServiceRoleForResourceAccessManager](#) that the service assumes when it needs to look up information about your organization and its accounts, for example, when you view the organization's structure in the AWS RAM console.

It does this by granting read-only permission to run the `organizations:Describe` and `organizations:List` operations that provide details of the organization's structure and accounts.

### Permissions details

This policy includes the following permissions.

- **organizations** – Allows principals to view information about the organization's structure, including the organizational units, and the AWS accounts they contain.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "organizations:DescribeAccount",
        "organizations:DescribeOrganization",
        "organizations:DescribeOrganizationalUnit",

```

```
        "organizations:ListAccounts",
        "organizations:ListAccountsForParent",
        "organizations:ListChildren",
        "organizations:ListOrganizationalUnitsForParent",
        "organizations:ListParents",
        "organizations:ListRoots"
    ],
    "Resource": "*"
  },
  {
    "Sid": "AllowDeletionOfServiceLinkedRoleForResourceAccessManager",
    "Effect": "Allow",
    "Action": [
      "iam:DeleteRole"
    ],
    "Resource": [
      "arn:aws:iam::*:role/aws-service-role/ram.amazonaws.com/*"
    ]
  }
]
```

## AWS RAM updates to AWS managed policies

View details about updates to AWS managed policies for AWS RAM since this service began tracking these changes. For automatic alerts about changes to this page, subscribe to the RSS feed on the [AWS RAM Document history](#) page.

Change	Description	Date
AWS Resource Access Manager started tracking changes	AWS RAM documented its existing managed policies and started tracking changes.	September 16, 2021

## Using Service-Linked Roles for AWS RAM

AWS Resource Access Manager 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 the AWS RAM service. Service-linked roles are predefined by AWS and include all the permissions that AWS RAM needs to call other AWS services on your behalf.

A service-linked role makes configuring AWS RAM easier because you don't have to manually add the necessary permissions. AWS RAM defines the permissions of its service-linked roles, and unless defined otherwise, only AWS RAM can assume its service-linked roles. The defined permissions include both a trust policy and a permissions policy, and that permissions policy cannot be attached to any other IAM entity.

For information about other services that support service-linked roles, see [AWS Services That Work with IAM](#) and look for the services that have **Yes** in the **Service-Linked Role** column. Choose a **Yes** with a link to view the service-linked role documentation for that service.

## Service-Linked Role Permissions for AWS RAM

AWS RAM uses the service-linked role named `AWSServiceRoleForResourceAccessManager` when you enable sharing with AWS Organizations. This role grants permissions to the AWS RAM service to view organization details, such as the list of member accounts and which organizational units each account is in.

This service-linked role trusts the following service to assume the role:

- `ram.amazonaws.com`

The role permissions policy named `AWSResourceAccessManagerServiceRolePolicy` is attached to this service-linked role, and allows AWS RAM to complete the following actions on the specified resources:

- Actions: read-only actions that retrieve details about your organization's structure.  
For the complete list of actions, you can view the policy in the IAM console: [AWSResourceAccessManagerServiceRolePolicy](#).

For a principal to turn on AWS RAM sharing within your organization, that principal (an IAM entity such as a user, group, or role), 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 AWS RAM

You don't need to manually create a service-linked role. When you turn on AWS RAM sharing within your organization in the AWS Management Console, or run the [EnableSharingWithAwsOrganization](#) in your account using the AWS CLI or an AWS API, AWS RAM creates the service-linked role for you.

If you delete this service-linked role, then AWS RAM no longer has permissions to view the details of your organization's structure.

## Editing a service-linked role for AWS RAM

AWS RAM does not allow you to edit the `AWSResourceAccessManagerServiceRolePolicy` service-linked role. After you create a service-linked role, you cannot 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 AWS RAM

You can use the IAM console, the AWS CLI or the AWS API to manually delete the service-linked role.

### To manually delete the service-linked role using IAM

Use the IAM console, the AWS CLI, or the AWS API to delete the `AWSResourceAccessManagerServiceRolePolicy` service-linked role. For more information, see [Deleting a Service-Linked Role](#) in the *IAM User Guide*.

## Supported Regions for AWS RAM Service-Linked Roles

AWS RAM supports using service-linked roles in all of the Regions where the service is available. For more information, see [AWS Regions and Endpoints](#) in the *Amazon Web Services General Reference*.

## Example IAM policies for AWS RAM

This topic includes examples of IAM policies for AWS RAM that demonstrate sharing specific resources and resource types and restricting sharing.

### Examples of IAM policies

- [Example 1: Allow sharing of specific resources \(p. 69\)](#)
- [Example 2: Allow sharing of specific resource types \(p. 69\)](#)



- [Example 3: Restrict sharing with external AWS accounts \(p. 69\)](#)

## Example 1: Allow sharing of specific resources

You can use an IAM policy to restrict principals to associating only specific resources with resource shares.

For example, the following policy limits principals to sharing only the resolver rule with the specified Amazon Resource Name (ARN). The operator `StringEqualsIfExists` allows a request if either the request doesn't include a `ResourceArn` parameter, or if it does include that parameter, that its value exactly matches the specified ARN.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["ram:CreateResourceShare", "ram:AssociateResourceShare"],
    "Resource": "*",
    "Condition": {
      "StringEqualsIfExists": {
        "ram:ResourceArn": "arn:aws:route53resolver:us-
west-2:123456789012:resolver-rule/rslvr-rr-5328a0899aexample"
      }
    }
  }]
}
```

## Example 2: Allow sharing of specific resource types

You can use an IAM policy to limit principals to associating only specific resource types with resource shares.

For example, the following policy limits principals to sharing only resolver rules.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["ram:CreateResourceShare", "ram:AssociateResourceShare"],
    "Resource": "*",
    "Condition": {
      "StringEqualsIfExists": {
        "ram:RequestedResourceType": "route53resolver:ResolverRule"
      }
    }
  }]
}
```

## Example 3: Restrict sharing with external AWS accounts

You can use an IAM policy to prevent principals from sharing resources with AWS accounts that are outside of its AWS organization.

For example, the following IAM policy prevents principals from adding external AWS accounts to resource shares.

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

```
"Statement": [{
  "Effect": "Allow",
  "Action": "ram:CreateResourceShare",
  "Resource": "*",
  "Condition": {
    "Bool": {
      "ram:RequestedAllowsExternalPrincipals": "false"
    }
  }
}]
}
```

## Example service control policies for AWS Organizations and AWS RAM

AWS RAM 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 [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

### Contents

- [Example 1: Prevent external sharing \(p. 70\)](#)
- [Example 2: Prevent users from accepting resource share invitations from external accounts outside your organization \(p. 71\)](#)
- [Example 3: Allow specific accounts to share specific resource types \(p. 71\)](#)
- [Example 4: Prevent sharing with the entire organization or with organizational units \(p. 72\)](#)
- [Example 5: Allow sharing with only specific principals \(p. 72\)](#)

The following examples show how you can control various aspects of resource sharing in an organization.

### Example 1: Prevent external sharing

The following SCP prevents users from creating resource shares that allow sharing with principals that are outside of the sharing user's organization.

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

```
"Statement": [
  {
    "Effect": "Deny",
    "Action": [
      "ram:CreateResourceShare",
      "ram:UpdateResourceShare"
    ],
    "Resource": "*",
    "Condition": {
      "Bool": {
        "ram:RequestedAllowsExternalPrincipals": "true"
      }
    }
  }
]
```

### Example 2: Prevent users from accepting resource share invitations from external accounts outside your organization

The following SCP blocks any principal in an affected account from accepting an invitation to use a resource share. Resource shares that are shared to other accounts in the same organization as the sharing account don't generate invitations and are therefore not affected by this SCP.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "ram:AcceptResourceShareInvitation",
      "Resource": "*"
    }
  ]
}
```

### Example 3: Allow specific accounts to share specific resource types

The following SCP allows *only* accounts 111111111111 and 222222222222 to create new resource shares that share Amazon EC2 prefix lists or to associate prefix lists with existing resource shares.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ram:AssociateResourceShare",
        "ram:CreateResourceShare"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEquals": {
          "aws:PrincipalAccount": [
            "111111111111",
            "222222222222"
          ]
        },
        "StringEqualsIfExists": {
          "ram:RequestedResourceType": "ec2:PrefixList"
        }
      }
    }
  ]
}
```

```
}
]
}
```

## Example 4: Prevent sharing with the entire organization or with organizational units

The following SCP prevents users from creating resource shares that share resources with an entire organization or with any organizational units. Users *can* share with individual AWS accounts in the organization, or with IAM roles or users.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ram:CreateResourceShare",
        "ram:AssociateResourceShare"
      ],
      "Resource": "*",
      "Condition": {
        "StringLike": {
          "ram:Principal": [
            "arn:aws:organizations::*:organization/*",
            "arn:aws:organizations::*:ou/*"
          ]
        }
      }
    }
  ]
}
```

## Example 5: Allow sharing with only specific principals

The following example SCP allows users to share resources with *only* organization o-12345abcdef, organizational unit ou-98765fedcba, and AWS account 111111111111.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": [
        "ram:AssociateResourceShare",
        "ram:CreateResourceShare"
      ],
      "Resource": "*",
      "Condition": {
        "StringNotEqualsIfExists": {
          "ram:Principal": [
            "arn:aws:organizations::123456789012:organization/o-12345abcdef",
            "arn:aws:organizations::123456789012:ou/o-12345abcdef/",
            "ou-98765fedcba",
            "111111111111"
          ]
        }
      }
    }
  ]
}
```

## Disabling resource sharing with AWS Organizations

If you previously enabled sharing with AWS Organizations and you no longer need to share resources with your entire organization or organizational units (OUs), you can disable sharing. When you disable sharing with AWS Organizations, all organizations or OUs are removed from the resource shares that you have created and they lose access to the shared resources.

### To disable sharing with AWS Organizations

1. Disable trusted access to AWS Organizations using the AWS Organizations [disable-aws-service-access](#) AWS CLI command.

```
$ aws organizations disable-aws-service-access --service-principal ram.amazonaws.com
```

#### Important

When you disable trusted access to AWS Organizations, principals within your organizations are removed from all resource shares and lose access to those shared resources.

2. Use the IAM console, the AWS CLI, or the IAM API operations to delete the **AWSServiceRoleForResourceAccessManager** service-linked role. For more information, see [Deleting a service-linked role](#) in the *IAM User Guide*.

## Logging and monitoring in AWS RAM

Monitoring is an important part of maintaining the reliability, availability, and performance of AWS RAM and your AWS solutions. You should collect monitoring data from all parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs. AWS provides several tools for monitoring your AWS RAM resources and responding to potential incidents:

### Amazon CloudWatch Events

Delivers a near-real-time stream of system events that describe changes in AWS resources. CloudWatch Events enables automated event-driven computing, as you can write rules that watch for certain events and trigger automated actions in other AWS services when these events happen. For more information, see [Monitoring AWS RAM using CloudWatch Events \(p. 73\)](#).

### 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 RAM API calls with AWS CloudTrail \(p. 75\)](#).

## Monitoring AWS RAM using CloudWatch Events

Using Amazon CloudWatch Events, you can set up automatic notifications for specific events in AWS RAM. Events from AWS RAM are delivered to CloudWatch Events in near-real time. You can configure CloudWatch Events to monitor events and invoke targets in response to events that indicate changes to your resource shares. Changes to a resource share trigger events for both the owner of the resource share and the principals that were granted access to the resource share.

When you create an event pattern, the source is `aws.ram`.

#### Note

Take care writing code that depends on these events. These events are not guaranteed, but are emitted on a best effort basis. If an error occurs when AWS RAM attempts to emit an event, the

service tries several more times. However, it can time out and result in the loss of that specific event.

For more information, see the [Amazon CloudWatch Events User Guide](#).

## Example: Alerting on resource share failures

Consider the scenario where you want to share Amazon EC2 capacity reservations with other accounts in your organization. Doing this is a good way to reduce your costs.

However, if you don't meet all of the [prerequisites for sharing a capacity reservation](#), then it can silently fail performing the asynchronous tasks involved in sharing resources. If the share operation fails, and your users in other accounts attempt to launch instances with one of those capacity reservations, then Amazon EC2 acts as if the capacity reservation was full and launches the instance as an on-demand instance instead. This can result in higher than expected costs.

To monitor for resource share failures, set up an Amazon CloudWatch Events rule that alerts you whenever an AWS RAM resource share fails. The following tutorial procedure uses an Amazon Simple Notification Service (SNS) topic to notify all topic subscribers whenever EventBridge discovers a resource sharing failure. For more information about Amazon SNS, see the [Amazon Simple Notification Service Developer Guide](#).

### To create a rule that notifies you when resource sharing fails

1. Open the [Amazon EventBridge console](#).
2. In the navigation pane, choose **Rules**, and then in the **Rules** list, choose **Create rule**.
3. Enter a name and optional description for your rule, then choose **Next**.
4. Scroll down to the **Event pattern** box, and choose **Custom patterns (JSON editor)**.
5. Copy and paste the following event pattern:

```
{
  "source": ["aws.ram"],
  "detail-type": ["Resource Sharing State Change"],
  "detail": {
    "event": ["Resource Share Association"],
    "status": ["failed"]
  }
}
```

6. Choose **Next**.
7. For **Target 1**, under **Target type**, choose **AWS service**.
8. Under **Select a target**, choose **SNS topic**.
9. For **Topic**, choose the SNS topic to which you want to publish the notification. This topic must already exist.
10. Choose **Next**, and then choose **Next** again to see to review your configuration.
11. When you're satisfied with your options, choose **Create rule**.
12. Back on the **Rules** page, ensure that your new rule is marked **Enabled**. If necessary, choose the radio button next to your rule name, and then choose **Enable**.

As long as that rule is enabled, any AWS RAM resource share that fails generates an SNS alert to the recipients of the topic you published to.

You can also confirm that shared capacity reservations are accessible to the accounts you shared them with by attempting to [view them in the Amazon EC2 console from those accounts](#).

## Logging AWS RAM API calls with AWS CloudTrail

AWS RAM is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS RAM. CloudTrail captures all API calls for AWS RAM as events. The calls captured include calls from the AWS RAM console and code calls to the AWS RAM API operations. If you create a trail, you can enable continuous delivery of CloudTrail events to an Amazon S3 bucket that you specify, including events for AWS RAM. If you don't configure a trail, you can still view the most recent events in the CloudTrail console in **Event history**. Use the information collected by CloudTrail to determine the request that was made to AWS RAM, the requesting IP address, the requester, when it was made, and additional details.

For more information about CloudTrail, see the [AWS CloudTrail User Guide](#).

### AWS RAM information in CloudTrail

CloudTrail is enabled on your AWS account when you create the account. When activity occurs in AWS RAM, 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 AWS RAM, 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:

- [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](#) and [Receiving CloudTrail log files from multiple accounts](#)

All AWS RAM actions are logged by CloudTrail and are documented in the [AWS RAM API Reference](#). For example, calls to the `CreateResourceShare`, `AssociateResourceShare`, and `EnableSharingWithAwsOrganization` actions generate entries in the CloudTrail log files.

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

- Whether the request was made with root or AWS Identity and Access Management (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 AWS RAM 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.

The following example shows a CloudTrail log entry for the `CreateResourceShare` action.

```
{
  "eventVersion": "1.05",
  "userIdentity": {
    "type": "IAMUser",
    "principalId": "NOPIOSFODNN7EXAMPLE",
    "arn": "arn:aws:iam::111122223333:user/admin",
    "accountId": "111122223333",
    "accessKeyId": "BCDIOSFODNN7EXAMPLE",
    "userName": "admin"
  },
  "eventTime": "2018-11-03T04:23:19Z",
  "eventSource": "ram.amazonaws.com",
  "eventName": "CreateResourceShare",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "192.0.1.0",
  "userAgent": "aws-cli/1.16.2 Python/2.7.10 Darwin/16.7.0 botocore/1.11.2",
  "requestParameters": {
    "name": "foo"
  },
  "responseElements": {
    "resourceShare": {
      "allowExternalPrincipals": true,
      "name": "foo",
      "owningAccountId": "111122223333",
      "resourceShareArn": "arn:aws:ram:us-east-1:111122223333:resource-share/EXAMPLE0-1234-abcd-1212-987656789098",
      "status": "ACTIVE"
    }
  },
  "requestID": "EXAMPLE0-abcd-1234-mnop-987654567876",
  "eventID": "EXAMPLE0-1234-abcd-hijk-543234565434",
  "readOnly": false,
  "eventType": "AwsApiCall",
  "recipientAccountId": "111122223333"
}
```

## Resilience in AWS RAM

The AWS global infrastructure is built around AWS Regions and Availability Zones. AWS Regions provide multiple physically separated and isolated Availability Zones, which are connected with low-latency, high-throughput, and highly redundant networking. With Availability Zones, you can design and operate applications and databases that automatically fail over between Availability 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 RAM

As a managed service, AWS RAM is protected by the AWS global network security procedures that are described in the [Amazon Web Services: Overview of Security Processes](#) whitepaper.

You use AWS published API calls to access AWS RAM through the network. Clients must support Transport Layer Security (TLS) 1.0 or later. We recommend TLS 1.2 or later. Clients must also support cipher suites with perfect forward secrecy (PFS) such as Ephemeral Diffie-Hellman (DHE) or Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). Most modern systems such as Java 7 and later support these modes.



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

# Using AWS RAM 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 help developers to build applications in their preferred language.

SDK documentation	Code examples
<a href="#">AWS SDK for C++</a>	<a href="#">AWS SDK for C++ code examples</a>
<a href="#">AWS SDK for Go</a>	<a href="#">AWS SDK for Go code examples</a>
<a href="#">AWS SDK for Java</a>	<a href="#">AWS SDK for Java code examples</a>
<a href="#">AWS SDK for JavaScript</a>	<a href="#">AWS SDK for JavaScript code examples</a>
<a href="#">AWS SDK for .NET</a>	<a href="#">AWS SDK for .NET code examples</a>
<a href="#">AWS SDK for PHP</a>	<a href="#">AWS SDK for PHP code examples</a>
<a href="#">AWS SDK for Python (Boto3)</a>	<a href="#">AWS SDK for Python (Boto3) code examples</a>
<a href="#">AWS SDK for Ruby</a>	<a href="#">AWS SDK for Ruby code examples</a>

**Example availability**

Can't find what you need? Request a code example with the feedback link.

# Document history for the AWS RAM User Guide

The following table describes important additions to the AWS Resource Access Manager documentation. We also update the documentation to address the feedback that you send us.

For notification about these updates, you can subscribe to the AWS RAM RSS feed.

update-history-change	update-history-description	update-history-date
<a href="#">AWS Resource Access Manager receives SOC and ISO certification.</a>	AWS RAM has been validated as being compliant with Service Organization Control (SOC) and International Organization for Standardization (ISO) ISO 9001, ISO 27001, ISO 27017, ISO 27018 and ISO 27701 standards.	May 31, 2022
<a href="#">AWS Resource Access Manager receives FedRAMP certification.</a>	AWS RAM has been validated as being compliant with the Federal Risk and Authorization Management Program (FedRAMP).	April 8, 2022
<a href="#">AWS Resource Access Manager receives PCI DSS certification.</a>	AWS RAM has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).	February 27, 2022
<a href="#">Added support for sharing global resources</a>	You can now share global resources with other AWS accounts.	December 2, 2021
<a href="#">Added support for AWS Cloud WAN core networks as shareable global resources.</a>	You can now share Cloud WAN core networks with other AWS accounts.	December 2, 2021
<a href="#">Support for sharing Amazon VPC IP Address Manager (IPAM) pools (p. 79)</a>	You can use AWS RAM to share Amazon VPC IPAM pools. For more information, see <a href="#">Sharable AWS resources</a> in the <i>AWS RAM User Guide</i> .	December 1, 2021
<a href="#">Support for sharing Amazon SageMaker resources (p. 79)</a>	You can use AWS RAM to share SageMaker lineage groups. For more information, see <a href="#">Sharable AWS resources</a> in the <i>AWS RAM User Guide</i> .	November 30, 2021
<a href="#">Support for sharing AWS Migration Hub Refactor Spaces resources (p. 79)</a>	You can use AWS RAM to share Migration Hub environments. For more information, see <a href="#">Sharable AWS resources</a> in the <i>AWS RAM User Guide</i> .	November 29, 2021

<a href="#">Added information about AWS RAM AWS-managed IAM permission policies.</a>	Published details about the available AWS-managed permission policies that you can access in the IAM console and attach to the IAM users and roles in your AWS account.	September 16, 2021
<a href="#">Added support for sharing S3 on Outposts resources</a>	You can now use AWS RAM to share S3 on Outposts with other AWS accounts.	August 5, 2021
<a href="#">Added support for additional managed permissions and sharing resources with IAM roles and IAM users</a>	For supported resource types, you can choose from additional AWS RAM managed permissions and share resources with IAM roles and IAM users.	June 10, 2021
<a href="#">Added support for sharing AWS Systems Manager Incident Manager resources</a>	You can now use AWS RAM to share AWS Systems Manager Incident Manager contacts and response plans with other AWS accounts.	May 10, 2021
<a href="#">Added support for sharing Amazon Route 53 resources</a>	You can now use AWS RAM to share Amazon Route 53 Resolver DNS Firewall rule groups with other AWS accounts.	March 31, 2021
<a href="#">Added support for sharing AWS Transit Gateway resources</a>	You can now use AWS RAM to share transit gateway multicast domains with other AWS accounts.	December 10, 2020
<a href="#">Added support for sharing AWS Network Firewall resources</a>	You can now use AWS RAM to share AWS Network Firewall firewall policies and rule groups with other AWS accounts.	November 17, 2020
<a href="#">Added support for sharing for Outposts and local gateway route tables</a>	You can now use AWS RAM to share Outposts and local gateway route tables with other AWS accounts.	October 15, 2020
<a href="#">Added support for sharing Route 53 query logs</a>	You can now use AWS RAM to share Route 53 query logs with other AWS accounts.	September 7, 2020
<a href="#">Added support for sharing AWS Certificate Manager Private Certificate Authority resources.</a>	You can now use AWS RAM to share ACM Private CA private certificate authorities (CAs) with other AWS accounts.	August 17, 2020
<a href="#">Added support for sharing AWS Glue data catalogs, databases, and tables.</a>	You can now use AWS RAM to share AWS Glue data catalogs, databases, and tables with other AWS accounts.	July 7, 2020
<a href="#">Added support for sharing Amazon VPC prefix lists.</a>	You can now use AWS RAM to share prefix lists.	June 29, 2020

<a href="#">Added support for sharing AWS Outposts customer-owned IPv4 addresses.</a>	You can now use AWS RAM to share AWS Outposts customer-owned IPv4 addresses with other AWS accounts.	April 22, 2020
<a href="#">Added support for sharing AWS App Mesh meshes</a>	You can now use AWS RAM to share meshes with other AWS accounts.	January 17, 2020
<a href="#">Added support for sharing AWS CodeBuild projects and report groups</a>	You can now use AWS RAM to share AWS CodeBuild projects and report groups with other AWS accounts.	December 13, 2019
<a href="#">Added support for sharing additional resources</a>	You can now use AWS RAM to share Amazon EC2 Dedicated Hosts, AWS Resource Groups resource groups, and Amazon EC2 Image Builder components, images, and image recipes with other AWS accounts.	December 2, 2019
<a href="#">Added support for sharing On-Demand Capacity Reservations</a>	You can now use AWS RAM to share On-Demand Capacity Reservations with other AWS accounts.	July 29, 2019
<a href="#">Added support for sharing Aurora DB clusters</a>	You can now use AWS RAM to share Aurora DB clusters with other AWS accounts.	July 2, 2019
<a href="#">Added support for sharing Traffic Mirroring targets</a>	You can now use AWS RAM to share Traffic Mirroring targets with other AWS accounts.	June 25, 2019
<a href="#">Added support for sharing license configurations</a>	You can now use AWS RAM to share AWS License Manager license configurations with other AWS accounts.	December 5, 2018
<a href="#">Added support for sharing subnets</a>	You can now use AWS RAM to share Amazon VPC subnets with other AWS accounts.	November 27, 2018
<a href="#">Added support for sharing transit gateways</a>	You can now use AWS RAM to share Amazon VPC transit gateways with other AWS accounts.	November 26, 2018
<a href="#">Added support for sharing forwarding rules</a>	You can now use AWS RAM to share Route 53 forwarding rules with other AWS accounts.	November 20, 2018