User Guide

AWS Resource Access Manager

AWS Resource Access Manager: User Guide

Table of Contents

What is AWS Resource Access Manager?

AWS Resource Access Manager (AWS RAM) helps you securely share your resources across AWS accounts, within your organization or organizational units (OUs), and with AWS Identity and Access Management (IAM) roles and users for supported resource types. 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, 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 also let you share them with specified IAM roles and users.

Contents

Video overviews

The following video provides a brief introduction to AWS RAM and describes how to create a resource share. For more information, see ???.

The following video demonstrates how to apply AWS managed permissions to your AWS resources. For more information, see ???.

This video demonstrates how to author and associate customer managed permissions following the best practice of least privilege. For more information see, ???.

Benefits of AWS RAM

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. Within the account that owns the resource, AWS RAM simplifies granting access to every role and user in that account without having to use identity-based permission policies.
- **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 policy that identifies AWS Identity and Access Management (IAM) principals (IAM roles and users) 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.
- 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 use AWS RAM to share an 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 attaching a resource-based policy aren't visible this way; instead, you have to discover and explicitly refer to the resource by its Amazon Resource Name (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. After you turn on the ability to share within your organization, sharing with accounts in the organization doesn't require invitations.

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 doing either of the following:

- Use the PromoteResourceShareCreatedFromPolicy API operation.
- Use the API operation's equivalent, which is the AWS Command Line Interface (AWS CLI) promote-resource-share-created-from-policy command.

How resource sharing works

When you share a resource in the *owning account* with another AWS account, the *consuming account*, you are granting access for principals in the consuming account to the shared resource. Any policies and permissions that apply to roles and users in the consuming account 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.

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** dropdown 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 for global resources, 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.

- A managed permission to associate with each resource type that you include in a resource share. The managed permission determines what the principals in the other accounts can do with the resources in the resource share.

  The behavior of the permission depends on the type of principal:

  - If the principal is in a different account from the one that owns the resource, then the permissions attached to the resource share are the maximum permissions available to be granted to roles and users in those accounts. The administrator of those accounts must then grant individual roles and users access to the shared resource with IAM identity-based policies. The permissions granted in those policies can't exceed those defined in permissions attached to the resource share.

The resource owning account retains full ownership of the resources that it shares.

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 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. 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`.

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 AWS 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 the 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 for AWS RAM

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, see the documentation for the resource's owning service.

Compliance and international standards

PCI DSS

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 RAM 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 RAM 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, and 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, 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

Terms and concepts for AWS RAM

The following concepts help explain how you can use AWS Resource Access Manager (AWS RAM) to share your resources.

Resource share

You share resources using AWS RAM by creating a *resource share*. A resource share has the following three elements:

- A list of one or more AWS resources to be shared.
- A list of one or more principals to whom access to the resources is granted.
- A managed permission for each type of resource that you include in the share. Each managed permission applies to all resources of that type in that resource share.

After you use AWS RAM to create a resource share, the principals specified in the resource share can be granted access to the share's resources.

- If you turn on AWS RAM sharing with AWS Organizations, and your principals that you share with are in the same organization as the sharing account, those principals can receive access as soon as their account administrator grants them permissions to use the resources using an AWS Identity and Access Management (IAM) permission policy.
- If you don't turn on AWS RAM sharing with Organizations, you can still share resources with individual AWS accounts that are in your organization. The administrator in the consuming account receives an invitation to join the resource share, and they must accept the invitation before the principals specified in the resource share can access the shared resources.
- You can also share with accounts outside of your organization, if the resource type supports it. The administrator in the consuming account receives an invitation to join the resource share, and they must accept the invitation before the principals specified in the resource share can access the shared resources. For information about which resource types support this type of sharing, see Shareable AWS resources and view the **Can share with accounts outside its organization** column.

Sharing account

The *sharing account* contains the resource that is shared and in which the AWS RAM administrator creates the AWS resource share by using AWS RAM.

An AWS RAM administrator is an IAM principal who has permissions to create and configure resource shares in the AWS account. Because AWS RAM works by attaching a resource-based policy to the resources in a resource share, the AWS RAM administrator also must have permissions to call the `PutResourcePolicy` operation in the AWS service for each resource type included in a resource share.

Consuming principals

The *consuming account* is the AWS account to which a resource is shared. The resource share can specify an entire account as the principal, or for some resource types, individual roles or users in the account. For information about which resource types support this type of sharing, see Shareable AWS resources and view the **Can share with IAM roles & users** column.

AWS RAM also supports service principals as consumers of resource shares. For information about which resource types support this type of sharing, see Shareable AWS resources and view the **Can share with service principals** column.

The principals in the consuming account can perform only those actions allowed by ***both*** of the following permissions:

- The managed permissions attached to the resource share. These specify the *maximum* permissions that can be granted to the principals in the consuming account.
- The IAM identity-based policies attached to individual roles or users by the IAM administrator in the consuming account. Those policies must grant `Allow` access to specified actions and to the Amazon Resource Name (ARN) of a resource in the sharing account.

AWS RAM supports the following IAM principal types as consumers of resource shares:

- **Another AWS account** – The resource share makes the included resources in the sharing account available to the consuming account.
- **Individual IAM roles or users in another account** – Some resource types support sharing directly with individual IAM roles or users. Specify this principal type by its ARN.
  - **IAM role** – `arn:aws:iam::123456789012:role/rolename`
  - **IAM user** – `arn:aws:iam::123456789012:user/username`
- **Service principal** – Share a resource with an AWS service to grant the service access to a resource share. Service principal sharing allows an AWS service to take actions on your behalf to ease the operational burden.

  To share with a service principal, choose to allow sharing with anyone, and then, under **Select principal type**, choose **Service principal** from the dropdown list. Specify the service principal's name in the following format:

  - `*service-id*.amazonaws.com`

  To mitigate the risk of confused deputy, the resource policy shows the resource owner's account ID in the `aws:SourceAccount` condition key.
- **Accounts in an organization** – If the sharing account is managed by AWS Organizations, then the resource share can specify the organization’s ID to share with all of the accounts in the organization. The resource share can alternatively specify an organizational unit (OU) ID to share with all of the accounts in that OU. A sharing account can share only with its own organization

or OU IDs within its own organization. Specify accounts in an organization by the ARN of the organization or the OU.

- **All accounts in an organization** – Following is an example ARN of an organization in AWS Organizations:

  ```
  arn:aws:organizations::123456789012:organization/o-<orgid>
  ```

- **All accounts in an organizational unit**– Following is an example ARN of an OU ID:

  ```
  arn:aws:organizations::123456789012:organization/o-<orgid>/ou-<rootid>-<ouid>
  ```

> ⚠ **Important**
>
> When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.
> Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

Resource-based policy

Resource-based policies are JSON text documents that implement the IAM policy language. Unlike identity-based policies that you attach to the principal, such as an IAM role or user, you attach resource-based policies to the resource. AWS RAM authors resource-based policies on your behalf based on the information you provide for your resource share. You must specify a `Principal` policy element that determines who can access the resource. For more information, see Identity-based policies and resource-based policies in the *IAM User Guide*.

The resource-based policies generated by AWS RAM are evaluated along with all other IAM policy types. This includes any IAM identity-based policies attached to the principals who are attempting to access the resource, and service control policies (SCPs) for AWS Organizations that might apply to the AWS account. Resource-based policies generated by AWS RAM participate in the same policy evaluation logic as all other IAM policies. For complete details of policy evaluation and how to determine the resulting permissions, see Policy evaluation logic in the *IAM User Guide*.

AWS RAM provides a simple and secure resource sharing experience by providing easy-to-use abstraction resource-based policies.

For those resource types that support resource-based policies, AWS RAM automatically constructs and manages the resource-based policies for you. For a given resource, AWS RAM builds the resource-based policy by combining the information from all of the resource shares that include that resource. For example, consider an Amazon SageMaker AI pipeline that you share by using AWS RAM and include in two different resource shares. You could use one resource share to provide read-only access to your entire organization. You could then use the other resource share to grant only SageMaker AI execution permissions to a single account. AWS RAM automatically combines those two different sets of permissions into a single resource policy with multiple statements. It then attaches the combined resource-based policy to the pipeline resource. You can view this underlying resource policy by calling the GetResourcePolicy operation. AWS services then use that resource-based policy to authorize any principal who attempts to perform an action on the shared resource.

Although you can manually create the resource-based policies and attach them to your resources by calling `PutResourcePolicy`, we recommend that you use AWS RAM because it provides the following advantages:

- **Discoverability for share consumers** – If you share resources by using AWS RAM, users can see all of the resources shared with them directly in the resource owning service's console and API operations as if those resources were directly in the user's account. For example, if you share an AWS CodeBuild project with another account, users in the consuming account can see the project in the CodeBuild console and in the results of CodeBuild API operations performed. Resources shared by directly attaching a resource-based policy aren't visible this way. Instead, you must discover and explicitly refer to the resource by its ARN.
- **Manageability for share owners** – If you share resources by using AWS RAM, resource owners in the sharing account can centrally see which other accounts have access to their resources. If you share a resource using a resource-based policy, you can see the consuming accounts only by examining the policy for individual resources in the relevant service console or API.
- **Efficiency** – If you share resources by using AWS RAM, you can share multiple resources and manage them as a unit. Resources shared by using only resource-based policies require individual policies attached to every resource that you share.

- **Simplicity** – With AWS RAM, you don't need to understand the JSON-based IAM policy language. AWS RAM provides ready-to-use AWS managed permissions that you can choose from to attach to your resource shares.

By using AWS RAM, you can even share some resource types that don’t support resource-based policies yet. For such resource types, AWS RAM automatically generates a resource-based policy as a representation of the actual permissions. Users can view this representation by calling GetResourcePolicy. This includes the following resource types:

- 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 Virtual Private Cloud – Customer-owned IPv4 addresses, prefix lists, subnets, traffic mirror targets, transit gateways, and transit gateway multicast domains

Examples of AWS RAM generated resource-based policies

If you share an EC2 Image Builder image resource with an individual ***account***, AWS RAM generates a policy that looks like the following example and attaches it to any image resources that are included in the resource share.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:root"
            },
            "Action": [
                "imagebuilder:GetImage",
                "imagebuilder:ListImages"
            ],
            "Resource": "arn:aws:imagebuilder:us-east-1:123456789012:image/
testimage/1.0.0/44"
        }
    ]
}
```

If you share an EC2 Image Builder image resource with an ***IAM role or user*** in a different AWS account, AWS RAM generates a policy that looks like the following example and attaches it to any image resources that are included in the resource share.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::123456789012:role/MySampleRole"
            },
            "Action": [
                "imagebuilder:GetImage",
                "imagebuilder:ListImages"
            ],
            "Resource": "arn:aws:imagebuilder:us-east-1:123456789012:image/
testimage/1.0.0/44"
        }
    ]
}
```

If you share an EC2 Image Builder image resource with all of the accounts in an organization or with the accounts an OU, AWS RAM generates a policy that looks like the following example and attaches it to any image resources that are included in the resource share.

Note

This policy uses "Principal": "*" and then uses the "Condition" element to restrict permissions to identities that match the specified PrincipalOrgID. For more information, see Implications of using "Principal": "*" in a resource-based policy.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "imagebuilder:GetImage",
                "imagebuilder:ListImages"
            ],
            "Resource": "arn:aws:imagebuilder:us-east-1:123456789012:image/
testimage/1.0.0/44",
            "Condition": {
                "StringEquals": {
                    "aws:PrincipalOrgID": "o-123456789"
                }
            }
        }
    ]
}
```

Implications of using "Principal": "*" in a resource-based policy

When you include "Principal": "*" in a resource-based policy, the policy grants access to all IAM principals in the account that contains the resource, subject to any restrictions imposed by a Condition element, if it exists. Explicit Deny statements in any policy that applies to the calling principal overrides the permissions granted by this policy. However, an ***implicit*** Deny (meaning the lack of an *explicit* Allow) in any applicable identity policies, permissions boundary policies, or session policies does ***not*** result in a Deny to the principals granted access to an action by such a resource-based policy.

If this behavior isn't desirable for your scenario, then you can limit this behavior by adding an ***explicit*** `Deny` statement to an identity policy, permissions boundary, or session policy that affects the relevant roles and users.

Managed permissions

Managed permissions define what actions principals can perform under which conditions on supported resource types in a resource share. When you create a resource share, you must specify which managed permission to use for each resource type included in the resource share. A managed permission lists the set of `actions` and *conditions* that principals can perform with the resource shared using AWS RAM.

You can attach only one managed permission for each resource type in a resource share. You can't create a resource share in which some resources of a certain type use one managed permission and other resources of the same type use a different managed permission. To do that, you would need to create two different resource shares and split the resources among them, giving each set a different managed permission. There are two different types of managed permissions:

AWS managed permissions

AWS managed permissions are created and maintained by AWS and grant permissions for common customer scenarios. AWS RAM defines at least one AWS managed permission for every supported resource type. Some resource types support more than one AWS managed permission, with one managed permission designated as the AWS default. The default AWS managed permission is associated unless you specify otherwise.

Customer managed permissions

Customer managed permissions are managed permissions that you author and maintain by precisely specifying which actions can be performed under which conditions with resources shared using AWS RAM. For example, you want to limit read access for your Amazon VPC IP Address Manager (IPAM) pools, which help you manage your IP addresses at scale. You can create customer managed permissions for your developers to assign IP addresses, but not view the range of IP addresses other developer accounts assign. You can follow the best practice of least privilege, granting only the permissions required to perform tasks on shared resources.

You define your own permission for a resource type in a resource share with the option to add conditions such as Global Context Keys and service specific keys to specify the conditions under which principals have access to the resource. These permissions can be used in one or more AWS RAM shares. Customer managed permissions are Region specific.

AWS RAM takes managed permissions as an input to author the resource-based policies for the resources you share.

Managed permission version

Any change to a managed permission is represented as a new version of that managed permission. The new version is the default for all new resource shares. Each managed permission always has one version designated as the default version. When you or AWS creates a new managed permission version, you must explicitly update the managed permission for each existing resource share. You can evaluate the changes before you apply them to your resource share in this step. All new resource shares will automatically use the new version of the managed permission for the corresponding resource type.

AWS managed permission versions

> AWS handles all changes to AWS managed permissions. Such changes address new functionality or remove discovered shortcomings. You can only apply the default managed permission version to your resource shares.

Customer managed permission versions

> You handle all changes to customer managed permissions. You can create a new default version, set an older version as the default, or delete versions that are no longer associated with any resource shares. Each customer managed permission can have up to five versions.

When you create or update a resource share, you can attach only the default version of the specified managed permission. For more information, see Updating AWS managed permissions to a newer version.

Sharing your AWS resources

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

- Enable resource sharing within AWS Organizations (optional)
- Create a resource share

Notes

- Sharing a resource with principals outside of the AWS account that owns the resource doesn't change the permissions or quotas that apply to the resource within 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 the resources were created.
- Some resources have special considerations and prerequisites for sharing. For more information, see Shareable AWS resources.

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`.

Note

By default, when you enable sharing with AWS Organizations, resource sharing within your organization restricts access to consumers within the same organization. If a consumer account leaves the organization, that account loses access to resources in the resource share. This restriction applies whether you share resources with an OU, the entire organization, or an individual account in the organization.

For account-to-account sharing within your organization, you can retain sharing access when accounts leave by setting `RetainSharingOnAccountLeaveOrganization` to `True` when you create a new resource share. With this setting enabled, AWS RAM sends an invitation to the consuming account (similar to sharing with external accounts). The account retains access to shared resources even if it leaves the organization.
The `RetainSharingOnAccountLeaveOrganization` setting has the following requirements and limitations:

- Requires `allowExternalPrincipals` to be `True`
- Can only be set when creating new resource shares
- Does not apply to sharing with OUs or the entire organization
- When `RetainSharingOnAccountLeaveOrganization` is set to `True`, you cannot use resource shares to share resources that can only be shared within an organization.

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.

Minimum permissions

To run the procedures below, you must sign in as a principal in the organization's management account that has the following permissions:

- `ram:EnableSharingWithAwsOrganization`
- `iam:CreateServiceLinkedRole`
- `organizations:enableAWSServiceAccess`
- `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. Here's an overview of the process:

1. Add the resources that you want to share.

2. For each resource type that you include in the share, specify the managed permission to use for that resource type.
   - You can choose from one of the available AWS managed permissions, an existing customer managed permission, or create a new customer managed permission.
   - AWS managed permissions are created by AWS to cover standard use cases.
   - Customer managed permissions allow you to tailor your own managed permissions to meet your security and business needs.

> **Note**
>
> If the selected managed permission has multiple versions, then AWS RAM automatically attaches the default version. You can attach ***only*** the version that is designated as the default.

3. Specify the principals that you want to have access to the resources.

Considerations

- If you later need to delete an AWS resource that you included in a share, we recommend that you first either remove the resource from any resource share that includes it, or delete the resource share.
- The resource types that you can include in a resource share are listed at Shareable AWS resources.
- 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, those principals 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. 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.
- 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 resource shares 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.

- If you share with a service principal, you can't associate any other principals with the resource share.
- If the sharing is between accounts or principals that are part of an organization, then any changes to organization membership dynamically affect access to the resource share.
  - If you add an AWS account to the organization or an OU that has access to a resource share, then that new member account automatically gets access to the resource share. The administrator of the account you shared with can then grant individual principals in that account access to the resources in that share.
  - If you remove an account from the organization or an OU that has access to a resource share, then any principals in that account automatically lose access to resources that were accessed through that resource share.
  - If you shared directly with a member account or with IAM roles or users in the member account and then remove that account from the organization, then any principals in that account lose access to the resources that were accessed through that resource share.

> ⚠ **Important**
>
> When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.
> Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

- 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 or, for supported services, IAM roles and users 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.

- 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. After 12 hours, the invitation expires and the end user principal in the resource share is disassociated. The invitation can no longer be accepted by end users.

  - 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

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. 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 checkboxes 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.

   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 managed permission with each resource type**, you can choose to associate a managed permission created by AWS with the resource type, choose an existing customer managed permission, or you can create your own customer managed permission for supported resource types. For more information, see Types of managed permissions.

   Choose **Create customer managed permission** to construct a customer managed permission that meets the requirements of your sharing use case. For more information see Create a customer managed permission. After completing the process, choose

   ↻

   and then you can select your new customer managed permission from the **Managed permissions** dropdown list.

> **Note**
>
> If the selected managed permission has multiple versions, then AWS RAM automatically attaches the default version. You can attach ***only*** the version designated as the default.

To display the actions that the managed permission allows, expand **View the policy template for this managed permission**.

7. Choose **Next**.
8. In **Step 3: Grant access to principals**, do the following:

   a. By default, **Allow sharing with anyone** 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 with IAM roles and users.

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

   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 checkbox next to each principal that you want to add.

        > **Important**
        >
        > When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.

Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

  - If you select the organization (the ID begins with `o-`), then principals in all AWS accounts in the organization can access the resource share.
  - If you select an OU (the ID begins with `ou-`), then principals in all AWS accounts in that OU and its child OUs can access the resource share.
  - If you select an individual AWS account, then only principals in 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 user. Instead, you must turn off **Display organizational structure** and use the drop down 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 role or user), 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/bob
  ```

  > **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.

- **Service principal** – To add a service principal, choose **Service principal** from the **Select principal type** dropbox. Enter the AWS service principal's name. Use the following syntax:

  - `service-id.amazonaws.com`

    For example:

    ```
    pca-connector-ad.amazonaws.com
    ```

   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 the managed permission for resource types that are included in your resource share, for those types that support more than the default managed 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.

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 managed permissions for that resource type.

> **Note**
>
> If you want to use a customer managed permission with a resource type in this resource share, you can either use an existing customer managed permission or create a new customer managed permission. Make note of the ARN for the customer managed permission, and then create the resource share. For more information, see Create a customer managed permission.

```
$ 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
- Use the resources that are shared with you

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.

Invitations aren't used in the following scenarios:

- If you're part of an organization in AWS Organizations and sharing in your organization is enabled, then principals in the organization automatically get access to the shared resources without invitations.
- If you share with the AWS account that owns the resource, then the principals in that account automatically get access to the shared resources without 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-aaaaaa111111",
            "resourceShareName": "MyNewResourceShare",
            "resourceShareArn": "arn:aws:ram:us-west-2:111122223333:resource-
share/1234abcd-ef12-9876-5432-bbbbbb222222",
            "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-aaaaaa111111
{
    "resourceShareInvitation": {
        "resourceShareInvitationArn": "arn:aws:ram:us-west-2:111122223333:resource-
share-invitation/1234abcd-ef12-9876-5432-aaaaaa111111",
        "resourceShareName": "MyNewResourceShare",
        "resourceShareArn": "arn:aws:ram:us-west-2:111122223333:resource-
share/1234abcd-ef12-9876-5432-bbbbbb222222",
        "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. 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

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

Resources are either Regional or global. You can use the fourth field in the Amazon Resource Name (ARN) to identify whether a resource is Regional or global. Regional resources show the AWS Region. If it's blank, then the resource is global.

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 the principals you share resources with in AWS RAM
- Deleting a resource share in AWS RAM

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.
3. If any of the managed permissions used by the resource shares in the results have a new version of the managed permission that is designated as the default, then the page displays a banner to alert you. You can choose to update all managed permission versions at once by choosing **Review and update all** at the top of the page.

   Alternatively, for individual resource shares with one or more new versions of managed permissions, the **Status** column displays **Update available**. Choosing that link begins the process of reviewing the updated managed permission versions and letting you assign them as the versions for the relevant resource types in that one resource share.
4. (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.
5. 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.
- **Managed Permissions** – Lists the managed permissions that are attached to this resource share. There can be at most one managed permission per resource type included in the resource share. Each managed permission displays the version of that managed permission that is associated with the resource share. If it is not the default version, then the console displays an **Update to default version** link. If you choose that link, then you are provided with the opportunity to update the resource share to use the default version.
- **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 include resource shares that contain 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. Here's an overview of the process:

1. Add the resources that you want to share.
2. For each resource type that you include in the share, specify the managed permission to use for that resource type.
   - You can choose from one of the available AWS managed permissions, an existing customer managed permission, or create a new customer managed permission.
   - AWS managed permissions are created by AWS to cover standard use cases.
   - Customer managed permissions allow you to tailor your own managed permissions to meet your security and business needs.

> **Note**
>
> If the selected managed permission has multiple versions, then AWS RAM automatically attaches the default version. You can attach ***only*** the version that is designated as the default.

3. Specify the principals that you want to have access to the resources.

Considerations

- If you later need to delete an AWS resource that you included in a share, we recommend that you first either remove the resource from any resource share that includes it, or delete the resource share.
- The resource types that you can include in a resource share are listed at Shareable AWS resources.
- 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, those principals 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. 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.
- 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 resource shares 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.
- If you share with a service principal, you can't associate any other principals with the resource share.
- If the sharing is between accounts or principals that are part of an organization, then any changes to organization membership dynamically affect access to the resource share.
  - If you add an AWS account to the organization or an OU that has access to a resource share, then that new member account automatically gets access to the resource share. The administrator of the account you shared with can then grant individual principals in that account access to the resources in that share.
  - If you remove an account from the organization or an OU that has access to a resource share, then any principals in that account automatically lose access to resources that were accessed through that resource share.
  - If you shared directly with a member account or with IAM roles or users in the member account and then remove that account from the organization, then any principals in that account lose access to the resources that were accessed through that resource share.

> ⚠ **Important**
>
> When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.
> Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

- 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 or, for supported services, IAM roles and users 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.

- 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. After 12 hours, the invitation expires and the end user principal in the resource share is disassociated. The invitation can no longer be accepted by end users.

- 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

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. 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 checkboxes 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.

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 managed permission with each resource type**, you can choose to associate a managed permission created by AWS with the resource type, choose an existing customer managed permission, or you can create your own customer managed permission for supported resource types. For more information, see Types of managed permissions.

   Choose **Create customer managed permission** to construct a customer managed permission that meets the requirements of your sharing use case. For more information see Create a customer managed permission. After completing the process, choose

   ↻

   and then you can select your new customer managed permission from the **Managed permissions** dropdown list.

   > **Note**
   >
   > If the selected managed permission has multiple versions, then AWS RAM automatically attaches the default version. You can attach ***only*** the version designated as the default.

   To display the actions that the managed permission allows, expand **View the policy template for this managed permission**.

7. Choose **Next**.

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

   a. By default, **Allow sharing with anyone** 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 with IAM roles and users.

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

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 checkbox next to each principal that you want to add.

  ⚠ **Important**

  When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.
  Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

  - If you select the organization (the ID begins with `o-`), then principals in all AWS accounts in the organization can access the resource share.
  - If you select an OU (the ID begins with `ou-`), then principals in all AWS accounts in that OU and its child OUs can access the resource share.
  - If you select an individual AWS account, then only principals in 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 user. Instead, you must turn off **Display organizational structure** and use the drop down 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 role or user), 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/bob
  ```

  > **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.

- **Service principal** – To add a service principal, choose **Service principal** from the **Select principal type** dropbox. Enter the AWS service principal's name. Use the following syntax:
  - `service-id.amazonaws.com`

    For example:

    ```
    pca-connector-ad.amazonaws.com
    ```

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 the managed permission for resource types that are included in your resource share, for those types that support more than the default managed 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.

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 managed permissions for that resource type.

> **Note**
>
> If you want to use a customer managed permission with a resource type in this resource share, you can either use an existing customer managed permission or create a new customer managed permission. Make note of the ARN for the customer managed permission, and then create the resource share. For more information, see Create a customer managed permission.

```
$ 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 managed permission, you can choose which managed permission applies to resources of each type.
- When a managed permission attached to the resource share has a new default version, you can update the managed permission to use the new version.
- 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.

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.
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 checkbox 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 managed permission with each resource type**, you can choose to associate a managed permission created by AWS with the resource type, choose an existing customer managed permission, or you can create your own customer managed permission. For more information, see Types of managed permissions.

   You can also choose **Create customer managed permission** to construct a customer managed permission that meets the requirements of your sharing use case. For more information, see Create a customer managed permission. After completing the process, choose

   ⟳ ,

   and then you can select your new customer managed permission from the **Managed permission** dropdown list.

   To display the actions that the managed permission allows, expand **View the policy template for this managed permission**.

7. If the version of the managed permission currently assigned to the resource share isn't the current default version, then you can update to the default version by choosing **Update to default version**.

   Note

   Until you save your changes to the resource share after the final step, you can cancel the version update by choosing **Revert to previous version**. However, for

AWS managed permissions, after you save the resource share, the change is final and you can no longer return to the previous version.

8. Choose **Next**.

9. 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 or individual IAM roles or users 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 checkbox next to each principal that you want to add.

        > ⚠ **Important**
        >
        > When you share with an organization or an OU, and that scope includes the account that owns the resource share, all principals in the sharing account automatically get access to the resources in the share. The access granted is defined by the managed permissions associated with the share. This is because the resource-based policy that AWS RAM attaches to each resource in the share uses `"Principal": "*"`. For more information, see Implications of using "Principal": "*" in a resource-based policy.
        > Principals in the other consuming accounts don't immediately get access to the share's resources. The other accounts' administrators must first attach identity-based permission policies to the appropriate principals. Those policies must grant `Allow` access to the ARNs of individual resources in the resource share. The permissions in those policies can't exceed those specified in the managed permission associated with the resource share.

> **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 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**.

  If you select an individual AWS account, then only that account can access the resource share. You can choose either of the following options.

  - **Another AWS account (other than the resource owner)** – Makes the resource available to the other account. The administrator of that account must complete the process by granting access to the shared resource using identity-based permission policies to individual roles and users. Those permissions can't exceed those defined in the managed permissions attached to the resource share.
  - **This AWS account (resource owner)** – All roles and users in the resource owning account automatically receive the access defined by the managed permissions attached to the resource share.
- 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 checkbox, and then choose **Deselect**.

10. Choose **Next**.
11. In **Step 4: Review and update**, review the configuration details for your resource share.
12. 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.

If any managed permissions are still using versions other than the default, you have another opportunity to address that by choosing **Update to default version**.

13. 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 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 managed permission per resource type in a resource share. If you try to add a managed permission to a resource type that already has a managed permission, you must include the `--replace` option or the command fails with an error.

  The following example command lists the ARNs for the managed permissions available for an Amazon Elastic Compute Cloud (Amazon EC2) subnet, and then uses one of those ARNs to replace the currently assigned AWS managed 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.
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.
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-a1b2c3dr",
            "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
        }
    ]
}
```

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.

To delete an AWS resource

If you need to delete an AWS resource that you included in a resource share, AWS recommends that you first ensure that you either remove the resource from any resource share that includes it, or delete the resource share.

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.
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**.
5. The deleted resource share disappears after two hours. Until then, it remains visible in the console with a deleted status.

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

To access shared resources, the owner of the resource share must add you as a principal. The owner can add any of the following as a principal to the resource share.

- The organization of which your account is a member
- An organizational unit (OU) that contains your account
- Your individual account
- For supported resource types, your specific IAM role or user

If you're added to the resource share through an AWS account that is a member of 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. Service principals also get automatic access to shared resources without accepting an invitation. If the account through which you receive access is later removed from the organization, then any principals in that account automatically lose access to the resources that were accessed through that resource share.

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.

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. After 12 hours, the invitation expires and the end user principal in the resource share is disassociated. The invitation can no longer be accepted by end users.

- 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

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.
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.
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-19096b1ee16e",
            "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.
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/home.
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.
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 individual AWS account and 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 are the only resource types that support leaving a resource share.

Service	Resource type
Amazon Aurora	`rds:Cluster`

Service	Resource type
Amazon EC2	`ec2:CapacityReservation` `ec2:DedicatedHost`
AWS License Manager	`license-manager:LicenseConfiguration`

AWS Outposts	`ec2:LocalGatewayRouteTable`
	`outposts:Outpost`
	`outposts:Site`

Amazon Route 53	`route53resolver:ResolverRule`

Amazon VPC	`ec2:CoipPool`
	`ec2:PrefixList`
	`ec2:Subnet`
	`ec2:TrafficMirrorTarget`
	`ec2:TransitGateway`
	`ec2:TransitGatewayMulticastDomain`

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.

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"
        }
    ]
}
```

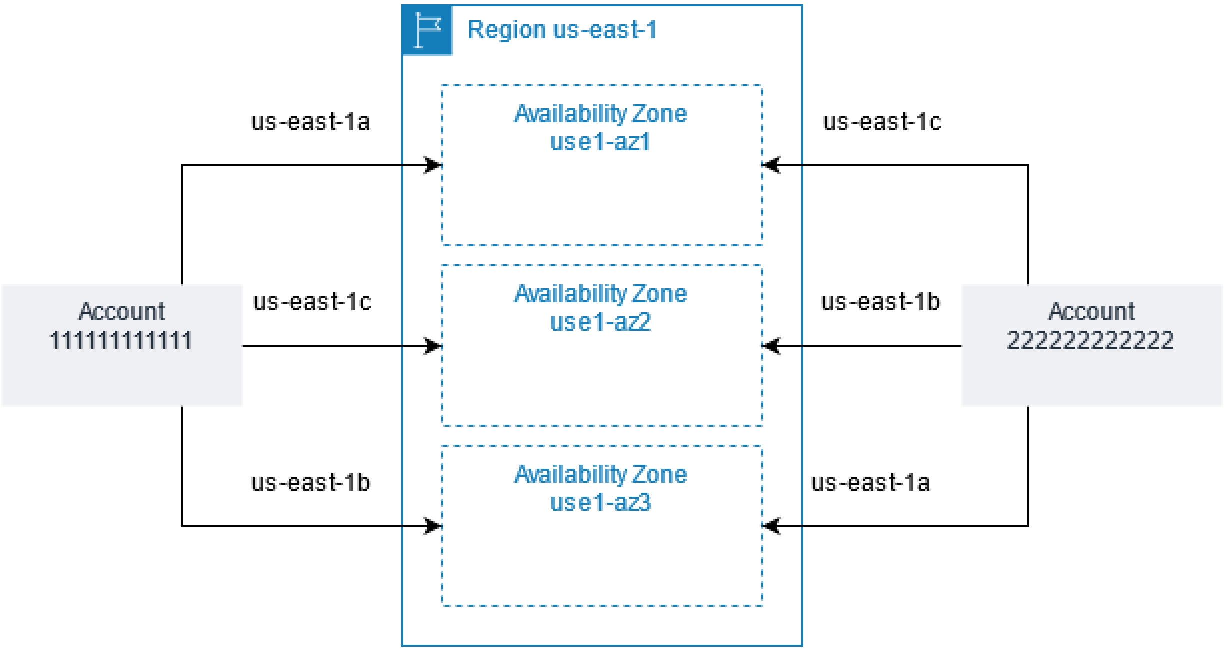
2. Then, you can run the command to leave that resource share. Note that you must also specify your account ID, 123456789012, as the principal to disassociate from the specified resource share, which is shared by account 111111111111.

```
$ 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 Availability 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 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 resource types, grouped by AWS service, that you can share by using AWS RAM. The columns in the tables specify which features each resource type supports:

Can share with IAM users and roles	✓ Yes – you can share resources of this type with individual AWS Identity and Access Management (IAM) roles and users, in addition to accounts. ✗ No – you can share resources of this type with only accounts.
Can share with accounts outside its organizat ion	✓ Yes – you may only share resources of this type with individua l accounts, inside or outside of its organization. See Considerations for more information. ✗ No – you can share resources of this type with only accounts that are members of the same organization.

Can use customer managed permissions	All resource types supported by AWS RAM support AWS managed permissions, but a Yes in this column means that customer managed permissions is also supported for this resource type. ✓ Yes – resources of this type support the use of customer managed permissions. ✗ No – resources of this type do not support the use of customer managed permissions.
Can share with service principals	✓ Yes – you can share resources of this type with AWS services. ✗ No – you can't share resources of this type with AWS services.

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	Can use customer managed permissions	Can share with service principals
Meshes `appmesh:Mesh`	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 information, see [Working with shared meshes](#) in the *AWS App Mesh User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

AWS AppSync GraphQL API

You can share the following AWS AppSync GraphQL API 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	Can use customer managed permissions	Can share with service principals
AppSync GraphyQL APIs `appsync:Apis`	Manage AWS AppSync GraphQL APIs centrally, and share them with other AWS accounts or your organization. This lets multiple accounts share AWS AppSync APIs as part of creating a unified AWS AppSync Merged API which can access data from multiple subschema APIs across different accounts in the same Region. For more information, see [Merged APIs]() in the *AWS AppSync Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Amazon API Gateway

You can share the following Amazon API Gateway 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	Can use customer managed permissions	Can share with service principals
API Gateway Private Custom Domains `apigateway:Domainnames`	Create and manage domain names centrally, and share them with other AWS accounts or your organization. This lets multiple accounts invoke your domain names that are mapped to private APIs. For more information, see [Custom domain names for private APIs in API Gateway](#) in the *Amazon API Gateway Developer Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Amazon Application Recovery Controller (ARC)

You can share the following Amazon Application Recovery Controller (ARC) 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	Can use customer managed permissions	Can share with service principals
Route 53 ARC Clusters `route53-recovery-control:Cluster`	Create and manage ARC clusters centrally, and share them with other AWS accounts or your organization. This lets multiple accounts create control panels and routing controls in a single shared cluster, reducing complexity and the total number of clusters an organization requires. For more information, see [Sharing clusters across accounts](#) in the *Amazon Application Recovery Controller (ARC) Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No
ARC Region switch plans `arc-region-switch:Plan`	Create and manage plans centrally, and share them with other AWS accounts or your organization. This lets multiple accounts use resources from an account that is	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	different from the account that hosts the plan. For more information, see Region switch in the *Amazon Application Recovery Controller (ARC) Developer Guide*.				

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	Can use customer managed permissions	Can share with service principals
Aurora DB Clusters `rds:Cluster`	Create and manage a DB cluster centrally , and share it with other AWS accounts or your organization. This lets multiple AWS accounts clone	✗ N	✓ Y Can share with **any**	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	a shared, centrally managed DB cluster. For more information, see [Cross-account cloning with AWS RAM and Amazon Aurora](#) in the *Amazon Aurora User Guide*.		AWS account.		

AWS Backup

You can share the following AWS Backup 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	Can use customer managed permissions	Can share with service principals
Backup Vaults `backup:BackupVault`	Create and manage logically air-gapped vaults centrally and share them with other AWS accounts or your organization. This option lets multiple	✓ Y	✓ Y Can share with **any**	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	accounts access and restore backups from the vault(s). For more information, see [Overview of logically air gapped vaults](#) in the *AWS Backup Developer Guide*.		AWS account.		

Amazon Bedrock

You can share the following Amazon Bedrock 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	Can use customer managed permissions	Can share with service principals
Bedrock Custom Model `bedrock:CustomModel`	Create and manage custom model centrally, and share it with other AWS accounts or your organization. This lets multiple accounts use the same custom	✓ Y	✗ N Can share with **only** AWS	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	model for generative AI applications. For more information, see Share a model for another account in the *Amazon Bedrock User Guide*.		accounts in its own organization.		

Billing and Cost Management

You can share the following Billing and Cost Management 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	Can use customer managed permissions	Can share with service principals
BCM Dashboards `bcm-dashboards:dashboard`	Create and manage Billing and Cost Management dashboards and share them with other AWS accounts within or outside your organization. When you share	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	a dashboard, only dashboard configurations are shared, not the underlying data. Recipients receive access to the dashboard layout and widget configurations, and will see data based on their own access permissions. This sharing capability allows organizations to establish common cost reporting practices and helps different teams view cost data consistently. For more information, see [Sharing dashboards]() in the *Billing and Cost Management User Guide*.				

AWS Billing View Service

You can share the following AWS Billing View Service 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	Can use customer managed permissions	Can share with service principals
Billing Views `billing:billingview`	Create and manage custom billing views centrally, and share them with other AWS accounts or your organization. This lets application and business unit owners access business unit-level AWS spend from a member account. For more information, see [Sharing custom billing views](#) in the *AWS Cost Management User Guide*.	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	✅ Y	❌ No

AWS Cloud Map

You can share the following AWS Cloud Map 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	Can use customer managed permissions	Can share with service principals
AWS Cloud Map Namespaces `servicediscovery:Namespace`	Create and manage namespaces centrally, and share them with other AWS accounts within your organization. This lets multiple AWS accounts discover services and instances in the shared namespace without the need for temporary credentials. For more information, see Shared AWS Cloud Map namespaces in the *AWS Cloud Map Developer Guide*.	✓ Y	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No

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	Can use customer managed permissions	Can share with service principals
Core networks `networkmanager:CoreNetwork`	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 [Share a core network](#) in the *AWS Cloud WAN User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Amazon CloudFront

You can share the following Amazon CloudFront 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	Can use customer managed permissions	Can share with service principals
Amazon CloudFront VpcOrigin `cloudfront:VpcOrigin`	Create and manage CloudFront VPC origins centrally, and share it with other AWS accounts or your organization. This lets multiple AWS accounts use a shared VPC origins for CloudFront distributions. For more information, see Working with shared resources in CloudFront in the *Amazon CloudFront Developer Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

AWS CloudHSM

You can share the following AWS CloudHSM 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	Can use customer managed permissions	Can share with service principals
AWS CloudHSM Backups `cloudhsm:Backup`	Manage AWS CloudHSM Backups centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts and users view information about the Backup and use it to restore a AWS CloudHSM Cluster. For more information, see [Managing AWS CloudHSM backups](#) in the *AWS CloudHSM User Guide*.	✓ Y	✓ Y	✓ Y	✗ No

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	Can use customer managed permissions	Can share with service principals
CodeBuild Projects `codebuild:Project`	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 [Working with shared projects](#) in the *AWS CodeBuild User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No
CodeBuild Report groups `codebuild:ReportGroup`	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	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	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 [Working with shared projects]() in the *AWS CodeBuild User Guide*.				

AWS CodeConnections

You can share the following CodeConnections 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	Can use customer managed permissions	Can share with service principals
Code Connections `codeconnections:Connection`	Manage the reuse of code connections in multiple accounts. In other words, sharing code connections reduces the administr	✗ N	✓ Y Can share with **any**	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	ator burden and need for administrator access in every account that requires a code connection. For more information, see [Share connections with AWS accounts](#) in the *Developer Tools Console User Guide*.		AWS account.		

Amazon DataZone

You can share the following DataZone 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	Can use customer managed permissions	Can share with service principals
DataZone Domains `datazone:Domain`	Create and manage domains centrally, and share it with other AWS accounts or your organization.	✗ N	✓ Y	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	This lets multiple accounts create Amazon DataZone domains. For more information, see What is Amazon DataZone in the *Amazon DataZone User Guide*.		Can share with **any** 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	Can use customer managed permissions	Can share with service principals
Capacity reservations `ec2:CapacityReservation`	Create and manage capacity reservations centrally, and share the reserved capacity with other AWS accounts or your organization. This lets multiple AWS	❌ N	Yes for Capacity Reservations (Can share with **any** AWS account).	❌ N	❌ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	accounts launch their Amazon EC2 instances into centrally managed reserved capacity. For more information, see [Working with shared Capacity Reservations](#) in the *Amazon EC2 User Guide*. Share Capacity Blocks for ML (UltraServer CBs are not yet supported) with other AWS accounts or your organization. This capability enables workloads running in different AWS accounts to launch Amazon EC2 instances into Capacity Blocks you own, helping you better utilize your reserved capacity and save costs. For more information, see [Working with shared Capacity Blocks](#) in		No for Capacity Blocks (Can share with **only** 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	Can use customer managed permissions	Can share with service principals
	the *Amazon EC2 User Guide*. **Important** If you don't meet all of the prerequisites for sharing a capacity reservation, 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				

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	that you can access the shared capacity reservation by attempting to view it in the Amazon EC2 console. 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 Example: Alerting on resource share failures.				

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Dedicated hosts `ec2:DedicatedHost`	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 [Working with shared Dedicated Hosts](#) in the *Amazon EC2 User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Placement groups `ec2:PlacementGroup`	Share the placement groups you own across your AWS accounts, both within and outside your organization. You can launch Amazon EC2 instances from any of the accounts you share with into a shared placement group. For more information, see, [Share a placement group](#) in the *Amazon EC2 User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

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	Can use customer managed permissions	Can share with service principals
Image Builder Components `imagebuilder:Component`	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 [Share EC2 Image Builder resources](#) in the *EC2 Image Builder User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No
Image Builder Container recipes `imagebuilder:ContainerRecipe`	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 [Share EC2 Image Builder](#)	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	resources in the *EC2 Image Builder User Guide*.				
Image Builder Images `imagebuilder:Image`	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 Share EC2 Image Builder resources in the *EC2 Image Builder User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Image Builder Image recipes `imagebuilder:ImageRecipe`	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 [Share EC2 Image Builder resources]() in the *EC2 Image Builder User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Elastic Load Balancing

You can share the following Elastic Load Balancing 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	Can use customer managed permissions	Can share with service principals
ELB Trust stores `elasticloadbalancing:TrustStore`	Create and manage Elastic Load Balancing trust stores centrally, and share them with other AWS accounts or your organization. Security admins can maintain a single or smaller number of trust stores and enable Mutual TLS configurations across Application Load Balancers. For more information, see [Share your Elastic Load Balancing trust store for Application Load Balancers](#) in the *User Guide for Application Load Balancers*.	✓ Y	✓ Y	✗ N	✗ No

AWS End User Messaging SMS

You can share the following AWS End User Messaging SMS 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	Can use customer managed permissions	Can share with service principals
AWS SMS Voice Opt out lists `sms-voice:OptOutList`	Create an opt-out list and share it with other AWS accounts in your organization. You can share the opt-out list so the other applications can opt out user's phone numbers from different AWS accounts or they can check the status of the user's phone number. For more information, see [Working with shared resources]() in the in AWS End User Messaging SMS User Guide.	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No
AWS SMS Voice Phone numbers `sms-voice:PhoneNumber`	Create and manage phone numbers to share them with other AWS accounts or your organization. This lets multiple AWS accounts send messages using the shared phone number. For more informati	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✓ Yes

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	on, see Working with shared resources in the in AWS End User Messaging SMS User Guide.				
AWS SMS Voice Pool `sms-voice:Pool`	Create and manage pools to share them with other AWS accounts or your organization. This lets multiple AWS accounts send messages using the shared pool. For more information, see Working with shared resources in the in AWS End User Messaging SMS User Guide.	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✓ Yes

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
AWS SMS Voice Sender IDs `sms-voice:SenderId`	Create and manage sender IDs and share them with other AWS accounts or your organization. This lets multiple AWS accounts send messages using the shared sender ID. For more information, see [Working with shared resources]() in the in AWS End User Messaging SMS User Guide.	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✓ Yes

Amazon FSx for OpenZFS

You can share the following Amazon FSx for OpenZFS 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	Can use customer managed permissions	Can share with service principals
FSx Volumes `fsx:Volume`	Create and manage FSx for OpenZFS volumes centrally, and share them with other AWS accounts or your organization. This lets multiple accounts perform data replication using OpenZfs snapshots under shared volumes through FSx APIs `CreateVolume` or `CopySnapshotAndUpdateVolume` . For more information, see [On-demand data replication](#) in the *Amazon FSx for OpenZFS User Guide*.	✓ Yes	✓ Yes Can share with **any** AWS account.	✓ Yes	✗ No

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	Can use customer managed permissions	Can share with service principals
AWS Glue Catalog `glue:Catalog`	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 [Sharing Data Catalog Tables and Databases Across AWS Accounts]() in the *AWS Lake Formation Developer Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No
AWS Glue Databases `glue:Database`	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	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	, and load (ETL) jobs that can join and query data across multiple accounts. For more information, see [Sharing Data Catalog Tables and Databases Across AWS Accounts](#) in the *AWS Lake Formation Developer Guide*.				

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
AWS Glue Tables `glue:Table`	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 Sharing Data Catalog Tables and Databases Across AWS Accounts in the *AWS Lake Formation Developer Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

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	Can use customer managed permissions	Can share with service principals
License configurations `license-manager:LicenseConfiguration`	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 [License configurations in License Manager](#) in the *License Manager User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

AWS Marketplace

You can share the following AWS Marketplace 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	Can use customer managed permissions	Can share with service principals
Marketplace Catalog Entities `aws-marketplace:Entity`	Create, manage, and share entities across AWS accounts or in your organization in AWS Marketplace. For more information, see [Resource sharing in AWS RAM](#) in the *AWS Marketplace Catalog API Reference*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

AWS Migration Hub Refactor Spaces

You can share the following AWS 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	Can use customer managed permissions	Can share with service principals
Refactor Spaces Environment	Create a Refactor Spaces environment, and use it to contain your Refactor Spaces applications. Share	✓ Y	✓ Y	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
`refactor-spaces:Environment`	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 Sharing Refactor Spaces environments using AWS RAM in the *AWS Migration Hub Refactor Spaces User Guide*.		Can share with **any** AWS account.		

Multi-party approval

You can share the following Multi-party approval 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	Can use customer managed permissions	Can share with service principals
Multi-Party Approval Team `mpa:ApprovalTeam`	Create and manage approval teams and share them with other AWS accounts or your organization. This allows other AWS accounts to use an approval team associated with a protected operation. A protected operation is a predefined list of operations that require team approval before it can be executed. For more information, see [Terms and Concepts](#) in the *Multi-party approval User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

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	Can use customer managed permissions	Can share with service principals
Network Firewalls `network-firewall:Firewall`	Create and manage firewalls centrally, and share them with other AWS accounts so they can create firewall endpoints. This enables multiple accounts to use the protections of a single firewall. For more information, see [Sharing AWS Network Firewall resources](#) in the *AWS Network Firewall Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No
Network Firewall Policies `network-firewall:FirewallPolicy`	Create and manage firewall policies centrally, and share them with other AWS accounts or your organization. This enables multiple accounts in an organization to share a common set of network monitoring, protection, and	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	filtering behaviors. For more information, see [Sharing AWS Network Firewall resources](#) in the *AWS Network Firewall Developer Guide*.				

Network Firewall 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 [Sharing AWS Network Firewall resources](#) in the *AWS Network Firewall Developer Guide*.	✓ Y	✓ Y
			Can share with **any** AWS account.
				✗ N	✗ No

Oracle Database@AWS

You can share the following Oracle Database@AWS 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	Can use customer managed permissions	Can share with service principals
Oracle Database@AWS Exadata Infrastructure `odb:CloudExadataInfrastructure`	With Oracle Database@AWS, you can share your Exadata infrastructure and your ODB network across multiple AWS accounts in the same AWS organization. This enables you to provision infrastructure once and reuse it across trusted accounts, allowing you to reduce costs while separating responsibilities. For more information, see Resource sharing in Oracle Database@AWS in the *Oracle Database@AWS User Guide*.	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	❌ N	❌ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Oracle Database@AWS ODB network `odb:OdbNetwork`	With Oracle Database@AWS, you can share your Exadata infrastructure and ODB network across multiple AWS accounts in the same AWS organization. This enables you to provision infrastructure once and reuse it across trusted accounts, allowing you to reduce costs while separating responsibilities. For more information, see [Resource sharing in Oracle Database@AWS]() in the *Oracle Database@AWS User Guide*.	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	❌ N	❌ No

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	Can use customer managed permissions	Can share with service principals
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 [Working with shared AWS Outposts resources](#) in the *AWS Outposts User Guide*.	✗ N	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No
Local gateway route tables `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, and view route table and virtual interface configuration. For	✗ N	✗ N Can share with **only** AWS accounts in its own organization.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	more information, see Shareable Outpost resources in the *AWS Outposts User Guide*.				
Outposts 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 Working with shared AWS Outposts resources in the *AWS Outposts User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

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	Can use customer managed permissions	Can share with service principals
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 [Working with shared AWS Outposts resources](#) in the *AWS Outposts User Guide*.	✗ N	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No

AWS Private Certificate Authority

You can share the following AWS 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	Can use customer managed permissions	Can share with service principals
Private certificate authority (CAs) `acm-pca:CertificateAuthority`	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 [Controlling access to a private CA](#) in the *AWS Private Certificate Authority User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✓ Yes

AWS Resource Explorer

You can share the following AWS Resource Explorer 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	Can use customer managed permissions	Can share with service principals
Resource Explorer Views `resource-explorer-2:View`	Create and configure Resource Explorer views centrally, and share them with other AWS accounts in your organization. This lets roles and users in multiple AWS accounts search for and discover the resources accessible through the view. For more information, see [Sharing Resource Explorer views](#) in the *AWS Resource Explorer User Guide*.	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	❌ N	❌ No

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	Can use customer managed permissions	Can share with service principals
Resource groups `resource-groups:Group`	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 [Host resource groups in AWS License Manager](#) in the *AWS License Manager User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

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	Can use customer managed permissions	Can share with service principals
Route 53 Resolver Firewall rule groups `route53resolver:FirewallRuleGroup`	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 [Sharing Route 53 Resolver DNS Firewall rule groups between AWS accounts]() in the *Amazon Route 53 Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No
Route 53 Profiles `route53profiles:Profile`	Create and manage Route 53 Profiles centrally, and share them with other AWS accounts or	✓ Y	✓ Y Can share with **any**	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	your organization. This lets multiple accounts apply the DNS configurations specified in the Route 53 Profiles to multiple VPCs. For more information, see [Amazon Route 53 Profiles](#) in the *Amazon Route 53 Developer Guide*.		AWS account.		

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Resolver rules `route53resolver:ResolverRule`	Create and manage Resolver 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 [Sharing Resolver rules with other AWS accounts and using shared rules](#) in the *Amazon Route 53 Developer Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Resolver Query Logging Configurations `route53resolver:ResolverQueryLogConfig`	Create and manage query logs centrally, and share them with other AWS accounts or 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 [Sharing Resolver query logging configurations with other AWS accounts](#) in the *Amazon Route 53 Developer Guide*.	✓ Yes	✓ Yes Can share with **any** AWS account.	✓ Yes	✗ No

Amazon Simple Storage Service

You can share the following Amazon Simple Storage Service 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	Can use customer managed permissions	Can share with service principals
S3 Access Grants `s3:AccessGrants`	Create and manage S3 Access Grants Instance centrally, and share them with other AWS accounts or your organization. This lets multiple accounts view and delete shared resources. For more information, see [S3 Access Grants Cross-account Access](#) in the *Amazon Simple Storage Service User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✓ Yes

Amazon SageMaker AI

You can share the following Amazon SageMaker AI 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	Can use customer managed permissions	Can share with service principals
SageMaker AI Resource Catalogs `sagemaker:SagemakerCatalog`	For **discoverability** – allows account owners to grant discoverability permissions to other accounts, for all feature group resources in the SageMaker AI catalog. Once granted access, users of those accounts can view the feature groups that have been shared with them from the catalog. For more information, see [Cross-account feature group discoverability and access](#) in the *Amazon SageMaker AI Developer Guide*. **Note** Discoverability and access are separate	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	permissions in SageMaker AI.				
SageMaker AI Feature groups `sagemaker:FeatureGroup`	For **access** – allows account owners to grant access permissions to other accounts, for select feature group resources. Once granted access, users of those accounts can use the feature groups that have been shared with them. For more information, see [Cross-account feature group discoverability and access](#) in the *Amazon SageMaker AI Developer Guide*. **Note** Discoverability and access are separate permissions in SageMaker AI.	✓ Yes	✓ Yes Can share with **any** AWS account.	✓ Yes	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI Hubs `sagemaker:Hub`	With Amazon SageMaker AI JumpStart, you can create and manage `sagemaker:Hub` centrally, and share them with other AWS accounts in the same organization. For more information, see [Control foundation model access using private curated hubs in Amazon SageMaker AI JumpStart]() in the *Amazon SageMaker AI Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI Lineage groups `sagemaker:LineageGroup`	Amazon SageMaker AI 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 [Cross-Account Lineage Tracking](#) in the *Amazon SageMaker AI Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI Model Cards `sagemaker:ModelCard`	Amazon SageMaker AI creates Model Cards to document critical details about your machine learning (ML) models in a single place for streamlined governance and reporting. Share your Model Cards with other AWS accounts or the accounts in your organization to achieve a multi-account strategy for your machine learning operations. This allows AWS accounts to share the model cards access for their ML activities to other accounts. For more information, see [Amazon SageMaker AI Model Cards]() in the *Amazon SageMaker AI Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI Model package groups `sagemaker:model-package-group`	With Amazon SageMaker AI Model Registry, you can create and manage `sagemaker:model-package-group` centrally, and share them with other AWS accounts to register model versions. For more information, see [Amazon SageMaker AI Model Registry](#) in the *Amazon SageMaker AI Developer Guide*.	✓ Y	✓ Y	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI Partner Apps `sagemaker:PartnerApp`	With SageMaker AI Partner AI Apps, you can create and manage SageMaker AI Partner AI Apps centrally, and share access to them with other AWS accounts. For more information, see [Setting up cross-account sharing for Amazon SageMaker AI partner AI apps]() in the *Amazon SageMaker AI Developer Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
SageMaker AI pipelines `sagemaker:Pipeline`	With Amazon SageMaker AI Model Building Pipelines, you can create, automate, and manage end-to-end machine learning workflows at scale. Share your pipelines with other AWS accounts or the accounts in your organization to achieve a multi-account strategy for your machine learning operations. This lets multiple AWS accounts and users view information about a pipeline and its executions with optional access to start, stop, and retry pipelines from other accounts. For more information, see Cross-Account Support for SageMaker	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	AI Pipelines in the *Amazon SageMaker AI Developer Guide*.				

AWS Service Catalog AppRegistry

You can share the following AWS Service Catalog AppRegistry 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	Can use customer managed permissions	Can share with service principals
AppRegistry Applications `servicecatalog:Applications`	Create an application, and use it to track the resources belonging to that application throughout your AWS environment. Share the application with other AWS accounts or your organization. This lets multiple AWS accounts and users view information about	✗ N	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	the application and associated resources with it locally. For more information, see [Creating applications](#) in the *Service Catalog User Guide*.				
AppRegistry Attribute Groups `servicecatalog:AttributeGroups`	Create an attribute group, and use it to store meta-data relating to your applications. Share the attribute groups with other AWS accounts or your organization. This lets multiple AWS accounts and users view information about the attribute groups. For more information, see [Creating attribute groups](#) in the *Service Catalog User Guide*.	✗ N	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No

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	Can use customer managed permissions	Can share with service principals
Incident Manager Contacts `ssm-contacts:Contact`	Create and manage contacts and escalation plans centrally, and share the contact details with other AWS accounts or your organization. This lets many AWS accounts view engagements occurring during an incident. **Note** Currently, the ability to add a contact that's shared from another account to an incident response plan is not supported. For more information, see [Working with	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	shared contacts and response plans in the *AWS Systems Manager Incident Manager User Guide*.				

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Incident Manager 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 Working with shared contacts and response plans in the *AWS Systems Manager Incident Manager User Guide.*	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

AWS Systems Manager

You can share the following AWS Systems 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	Can use customer managed permissions	Can share with service principals
SSM JITNA Auto-Deny policies `ssm:Document`	Create an approval policy for just-in-time node access with Systems Manager. A deny-access policy explicitly prevents the auto-approval of access requests to the nodes you specify. Share the deny-access policy with other AWS accounts or your organization. This ensures your deny-access policy for just-in-time node access applies to all accounts in your organization. For more information, see Just-in-time node access using Systems Manager in the *AWS Systems Manager User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Parameter Store Advanced Parameters `ssm:Parameter`	Create a parameter, and use it to store configuration data that you can reference in your scripts, commands, SSM documents, and configuration and automation workflows. Share the parameter with other AWS accounts or your organization. This lets multiple AWS accounts and users view information about the string and improve security by separating your data from your code. For more information, see [Working with shared parameters](#) in the *AWS Systems Manager User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

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	Can use customer managed permissions	Can share with service principals
Customer-owned IPv4pool `ec2:CoipPool`	During the AWS Outposts installation process, AWS creates an address pool, known as a *customer-owned IP address pool*, 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 instances, using Elastic IP addresses or using the subnet	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	❌ N	❌ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	setting that automatically assigns customer-owned IP addresses. For more information, see [Customer-owned IP addresses]() in the *AWS Outposts User Guide*.				
IPAM pools `ec2:IpamPool`	Share Amazon VPC 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 [Share an IPAM pool using AWS RAM]() in the *Amazon VPC IP Address Manager User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
IPAM resource discoveries `ec2:IpamResourceDiscovery`	Share resource discoveries with other AWS accounts. A resource discovery is an Amazon VPC IPAM component that enables IPAM to manage and monitor resources that belong to the owning account. For more information, see Work with resource discoveries in the *Amazon VPC IPAM User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
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 [Working with shared prefix lists](#) in the *Amazon VPC User Guide*.	❌ N	✅ Y Can share with **any** AWS account.	❌ N	❌ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Subnets `ec2:Subnet`	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 [Working with VPC sharing](#) in the *Amazon VPC User Guide*. **Note** To include a subnet when you create a resource share, you must have	❌ N	❌ N Can share with **only** AWS accounts in its own organization.	❌ N	❌ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	the `ec2:DescribeSubnets` and `ec2:DescribeVpcs` permissions, in addition to `ram:CreateResourceShare` . Default subnets are not shareable. You can share only subnets you create yourself.				

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Security groups `ec2:SecurityGroup`	Create and manage security groups centrally, and share them with other AWS accounts or your organization. This lets multiple AWS accounts associate the security group with their Elastic network interfaces. For more information, see [Share a security group](#) in the *Amazon VPC User Guide*.	✓ Y	✗ N Can share with **only** AWS accounts in its own organization.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Traffic mirror targe `ec2:TrafficMirrorTarget`	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 [Cross-account traffic mirroring targets](#) in the *Traffic Mirroring Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Transit gateways `ec2:TransitGateway`	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 [Sharing a transit gateway](#) in the *Amazon VPC Transit Gateways*. **Note** To include a transit gateway when you create a resource share, you must have the `ec2:DescribeTransitGateway`	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
	permission in addition to `ram:CreateResourceShare` .				
Transit gateway multicast domains `ec2:TransitGatewayMulticastDomain`	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 [Working with shared multicast domains]() in the *Transit Gateways Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
AWS Verified Access groups `ec2:VerifiedAccessGroup`	Create and manage AWS Verified Access groups centrally, and then share them with other AWS accounts or your organization. This lets applications in multiple accounts use a single, shared set of AWS Verified Access endpoints. For more information, see [Share your AWS Verified Access group through AWS Resource Access Manager](#) in the *AWS Verified Access User Guide*.	✓ Y	✓ Y Can share with **any** AWS account.	✗ N	✗ No

Amazon VPC Lattice

You can share the following Amazon VPC Lattice 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	Can use customer managed permissions	Can share with service principals
Amazon VPC Lattice resource configuration `vpc-lattice:ResourceConfiguration`	Create a resource configuration in Amazon VPC Lattice to share VPC resources across accounts and VPCs. In the resource configuration, you identify who can access that resource and specify the resource gateway through which you want to share the resource. Consumers can access the VPC resource through a resource VPC endpoint that they create in AWS PrivateLink. For more information, see Access VPC resources through AWS PrivateLink in the *AWS PrivateLink User Guide* and Resource configuration for VPC resources in the *VPC Lattice User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Amazon VPC Lattice services `vpc-lattice:Service`	Create and manage Amazon VPC Lattice services centrally, and share them with individual AWS accounts or your organization. This allows service owners to connect, secure, and observe service-to-service communication in a multi-account environment. For more information, see [Working with shared resources](#) in the *VPC Lattice User Guide*.	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Resource type and code	Use case	Can share with IAM users and roles	Can share with accounts outside its organization	Can use customer managed permissions	Can share with service principals
Amazon VPC Lattice service network `vpc-lattice:ServiceNetwork`	Create and manage Amazon VPC Lattice service networks centrally, and share them with individual AWS accounts or your organization. This allows service network owners to connect, secure, and observe service-to-service communication in a multi-account environment. For more information, see [Working with shared resources](#) in the *Amazon VPC Lattice User Guide.*	✗ N	✓ Y Can share with **any** AWS account.	✓ Y	✗ No

Managing permissions in AWS RAM

In AWS RAM, there are two types of managed permissions, AWS managed permissions and customer managed permissions.

Managed permissions define how a consumer can act on the resources in a resource share. When you create a resource share, you must specify which managed permission to use for each resource type that is included in the resource share. The policy template in the managed permission contains everything needed for a resource-based policy except for the principal and the resource. The resource's Amazon Resource Name (ARN) and the ARN of the principals associated with the resource share complete the elements of a resource-based policy. AWS RAM then authors the resource-based policy that it attaches to all resources in that resource share.

Each managed permission can have one or more versions. One version is designated as the *default* version for that managed permission. Occasionally, AWS updates an AWS managed permission for a resource type by creating a new version and designating that new version as the default. You can also update your customer managed permissions by creating new versions. Managed permissions that are already attached to a resource share are ***not*** automatically updated. The AWS RAM console does indicate when a new default version is available, and you can review the changes in the new default version compared to the previous one.

Note

We recommend that you update to the new version of the AWS managed permission as soon as possible. These updates typically add support for new or updated AWS services that can share additional resource types using AWS RAM. A new default version can also address and correct security vulnerabilities.

Important

You can only attach the default version of the managed permission to a new resource share.

You can retrieve the list of the available managed permissions at any time. For more information, see Viewing managed permissions.

Topics

Viewing managed permissions

You can view details about 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 **Managed permissions library** in the AWS RAM console.

Console

To view details about managed permissions available in AWS RAM

1. Navigate to the **Managed 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. Although all Regions share the same available AWS managed permissions, this affects the number of associated resource shares displayed for each managed permission in Step 5. Customer managed permissions are only available in the Region that they were created in.
3. In the **Managed permissions** list, choose the managed permission for which you want to view details. You can use the search box to filter the list of managed permissions by entering part of a name or a resource type, or choosing a managed permission type from the dropdown list.
4. (Optional) To change the display preferences, choose the gear icon in the upper right of the **Managed 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 managed permission, the list displays the following information:

- **Managed permission name** – The name of the managed permission.
- **Resource type** – The resource type that is associated with the managed permission.
- **Managed permission type** – Whether the managed permission is an AWS managed permission or a customer 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.
  - **Managed permission version** – The version of the managed permission that is attached to this resource share.
  - **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.
- **Status** – Describes whether the managed permission is:
  - **Attachable** – You can attach the managed permission to your resource shares.
  - **Unattachable** – You can't attach the managed permission to your resource shares.
  - **Deleting** – The managed permission is no longer active and will soon be deleted.
  - **Deleted** – The managed permission has been deleted. It remains visible for two hours before it disappears from the **Managed permission library**.

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

- **Resource type** – The type of AWS resource to which this managed permission applies.
- **Number of versions** – You can have up to five versions of a customer managed permission.
- **Default version** – Specifies which version is the default and therefore assigned automatically to all new resource shares that use this managed permission. Any existing resource shares that use different versions display a prompt for you to update the resource share to the default version.
- **ARN** – The Amazon Resource Name (ARN) of the managed permission. The ARNs for AWS managed permissions use the following format:

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

  The substring *[DefaultPermission]* (without the brackets in an actual ARN) is present in the name of only the one managed permission for that resource type that is designated the default.
- **Managed permission versions** – You can choose which version's information to display in the tabs below this dropdown list.
  - **Details** tab:
    - **Creation time** – The date and time when this version of the managed permission was created.
    - **Last updated time** – The date and time when this version of the managed permission was last updated.
  - **Policy template** tab – The list of service actions and conditions, if applicable, that this version of the managed permission allows principals to perform on the associated resource type.
  - **Associated resource shares** – The list of resource shares that use this version of the managed permission.

AWS CLI

To view details about managed permissions available in AWS RAM

You can use the list-permissions command to get a list of the managed 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",
            "status": "ATTACHABLE",
            "creationTime": "2022-06-30T13:03:31.732000-07:00",
            "lastUpdatedTime": "2022-06-30T13:03:31.732000-07:00",
            "isResourceTypeDefault": false,
            "permissionType": "AWS_MANAGED"
        },
        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMBlankEndEntityCertificateAPIPassthroughIssuanceCertificateAuthority",
            "version": "1",
            "defaultVersion": true,
            "name":
 "AWSRAMBlankEndEntityCertificateAPIPassthroughIssuanceCertificateAuthority",
            "resourceType": "acm-pca:CertificateAuthority",
            "status": "ATTACHABLE",
            "creationTime": "2022-11-18T07:05:46.976000-08:00",
            "lastUpdatedTime": "2022-11-18T07:05:46.976000-08:00",
            "isResourceTypeDefault": false,
            "permissionType": "AWS_MANAGED"
        },

        ... TRUNCATED FOR BREVITY ... RUN COMMAND TO SEE COMPLETE LIST OF
 PERMISSIONS ...

        {
            "arn": "arn:aws:ram::aws:permission/
AWSRAMVPCPermissionsNetworkManagerCoreNetwork",
            "version": "1",
            "defaultVersion": true,
            "name": "AWSRAMVPCPermissionsNetworkManagerCoreNetwork",
            "resourceType": "networkmanager:CoreNetwork",
            "status": "ATTACHABLE",
            "creationTime": "2022-06-30T13:03:46.557000-07:00",
            "lastUpdatedTime": "2022-06-30T13:03:46.557000-07:00",
            "isResourceTypeDefault": false,
            "permissionType": "AWS_MANAGED"
        },        {
            "arn": "arn:aws:ram:us-east-1:123456789012:permission/My-Test-CMP",
            "version": "1",
            "defaultVersion": true,
            "name": "My-Test-CMP",
            "resourceType": "ec2:IpamPool",
            "status": "ATTACHABLE",
            "creationTime": "2023-03-08T06:54:10.038000-08:00",
            "lastUpdatedTime": "2023-03-08T06:54:10.038000-08:00",
            "isResourceTypeDefault": false,
            "permissionType": "CUSTOMER_MANAGED"
        }
    ]
}
```

You can also find the ARN of a specific managed permission by its name in the `--query` parameter of the `list-permissions` AWS CLI command. The following example filters the output to include only elements in the `permissions` array results that match the specified name. We also specify that we want to see only the ARN field in the results, and in plain text format instead of the default JSON.

```
$ aws ram list-permissions \
    --query "permissions[?name == 'My-Test-CMP'].arn \
    --output text
arn:aws:ram:us-east-1:123456789012:permission/My-Test-CMP
```

After you find the ARN of the specific managed permission you're interested in, you can retrieve its details, including its JSON policy text, by running the command get-permission.

```
$ aws ram get-permission \
    --permission-arn arn:aws:ram:us-east-1:123456789012:permission/My-Test-CMP
{
    "permission": {
        "arn": "arn:aws:ram:us-east-1:123456789012:permission/My-Test-CMP",
        "version": "1",
        "defaultVersion": true,
        "name": "My-Test-CMP",
        "resourceType": "ec2:IpamPool",
        "permission": "{\n\t\"Effect\": \"Allow\",\n\t\"Action\": [\n
\t\t\"ec2:GetIpamPoolAllocations\",\n\t\t\"ec2:GetIpamPoolCidrs\",\n\t
\t\"ec2:AllocateIpamPoolCidr\",\n\t\t\"ec2:AssociateVpcCidrBlock\",\n
\t\t\"ec2:CreateVpc\",\n\t\t\"ec2:ProvisionPublicIpv4PoolCidr\",\n\t\t
\"ec2:ReleaseIpamPoolAllocation\"\n\t]\n}",
        "creationTime": "2023-03-08T06:54:10.038000-08:00",
        "lastUpdatedTime": "2023-03-08T06:54:10.038000-08:00",
        "isResourceTypeDefault": false,
        "permissionType": "CUSTOMER_MANAGED",
        "featureSet": "STANDARD",
        "status": "ATTACHABLE"
    }
}
```

Creating and using customer managed permissions in AWS RAM

AWS Resource Access Manager (AWS RAM) provides at least one AWS managed permission for every resource type that you can share. However, those managed permissions might not provide least privilege access for your sharing use case. When one of the provided AWS managed permissions doesn't work, you can create your own *customer managed permission*.

Customer managed permissions are managed permissions that you author and maintain by precisely specifying which actions can be performed under which conditions with resources shared using AWS RAM. For example, you want to limit read access for your Amazon VPC IP Address Manager (IPAM) pools, which help you manage your IP addresses at scale. You can create customer managed permissions for your developers to assign IP addresses, but not view the range of IP addresses other developer accounts assign. You can follow the best practice of least privilege, granting only the permissions required to perform tasks on shared resources.

In addition, you can update or delete customer managed permissions as needed.

Topics

- Create a customer managed permission
- Create a new version of a customer managed permission
- Choose a different version to be the default for a customer managed permission
- Delete a customer managed permission version
- Delete a customer managed permission

Create a customer managed permission

Customer managed permissions are specific to an AWS Region. Make sure that you create this customer managed permission in the appropriate Region.

Console

To create a customer managed permission

1. Do one of the following:

   - Navigate to the Managed permissions library, and choose **Create a customer managed permission**.
   - Navigate directly to the Create a customer managed permission page in the console.

2. For **Customer managed permission details**, enter a customer managed permission name.
3. Choose the resource type to which this managed permission applies.
4. For **Policy template**, you define which operations are allowed to be performed on this resource type.

   - You can choose **Import managed permission** to use actions from an existing managed permission.
   - Select or deselect access level information to meet your requirements in the visual editor.
   - Add or modify conditions using the **JSON editor**.

5. (Optional) To attach tags to the managed permission, for **Tags**, enter a tag key and value. Add additional tags by choosing **Add new tag**. Repeat this step as needed.
6. When you're done, choose **Create customer managed permission**.

AWS CLI

To create a customer managed permission

- Run the command create-permission and specify a name, the resource type that the customer managed permission applies to, and the policy template body text.

  The following example command creates a managed permission for the `imagebuilder:Component` resource type.

```
$ aws ram create-permission \
    --name TestCMP \
    --resource-type imagebuilder:Component \
    --policy-template "{\"Effect\":\"Allow\",\"Action\":
[\"imagebuilder:ListComponents\"]}"
{
    "permission": {
        "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
        "version": "1",
        "defaultVersion": true,
        "isResourceTypeDefault": false,
        "name": "TestCMP",
        "resourceType": "imagebuilder:Component",
        "status": "ATTACHABLE",
        "creationTime": 1680033769.401,
        "lastUpdatedTime": 1680033769.401
    }
}
```

Create a new version of a customer managed permission

If the use case for your customer managed permission changes, you can create a new version of the managed permission. This doesn't affect your existing resource shares, only the new resource shares going forward that use this customer managed permission.

Each managed permission can have up to five versions, but you can associate only the default version.

Console

To create a new version of a customer managed permission

1. Navigate to the Managed permissions library.
2. Filter the list of managed permissions by **Customer managed**, or search for the name of the customer managed permission that you want to change.
3. From the managed permission details page, under the **Managed permission versions** section, choose **Create version**.
4. For **Policy template**, you can add or remove actions and conditions with the visual editor or JSON editor.

You also have the option to choose **Import managed permission** to use an existing policy template.

5. When you're finished, choose **Create version** at the bottom of the page.

AWS CLI

To create a new version of a customer managed permission

1. Find the Amazon Resource Name (ARN) of the managed permission for which you want create a new version. Do this by calling list-permissions with the `--permission-type CUSTOMER_MANAGED` parameter to include only customer managed permissions.

```
$ aws ram-cmp list-permissions --permission-type CUSTOMER_MANAGED
{
    "permissions": [
        {
            "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
            "version": "2",
            "defaultVersion": true,
            "isResourceTypeDefault": false,
            "name": "TestCMP",
            "permissionType": "CUSTOMER_MANAGED",
            "resourceType": "imagebuilder:Component",
            "status": "ATTACHABLE",
            "creationTime": 1680035597.346,
            "lastUpdatedTime": 1680035597.346
        }
    ]
}
```

2. After you have the ARN, you can call the create-permission-version operation and provide the updated policy template.

```
$ aws ram create-permission-version \
    --permission-arn arn:aws:ram:us-east-1:123456789012:permission/TestCMP \
    --policy-template {"Effect":"Allow","Action":
["imagebuilder:ListComponents"]}
{
    "permission": {
        "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
        "version": "2",
        "defaultVersion": true,
        "isResourceTypeDefault": false,
        "name": "TestCMP",
        "status": "ATTACHABLE",
        "resourceType": "imagebuilder:Component",
        "permission": "{\"Effect\":\"Allow\",\"Action\":
[\"imagebuilder:ListComponents\"]}",
        "creationTime": 1680038973.79,
        "lastUpdatedTime": 1680038973.79
    }
}
```

The output includes the version number of the new version.

Choose a different version to be the default for a customer managed permission

You can set another customer managed permission version as the new default version.

Console

To set a new default version for a customer managed permission

1. Navigate to the [Managed permissions library](https://console.aws.amazon.com/ram/home#Permissions:).
2. Filter the list of managed permissions by **Customer managed**, or search for the name of the customer managed permission that you want to change.
3. From the Customer managed permission details page, under the **Managed permission versions** section, use the dropdown list to choose the version that you want to set as the new default.
4. Choose **Set as default version**.
5. When the dialog box appears, confirm that you want this version to be the default for all new resource shares that use this customer managed permission. If you agree, choose **Set as default version**.

AWS CLI

To set a new default version for a customer managed permission

1. Find the version number that you want to set as the default version by calling list-permission-versions.

   The following example command retrieves the current versions for the specified managed permission.

```
$ aws ram list-permission-versions \
    --permission-arn arn:aws:ram:us-east-1:123456789012:permission/TestCMP
{
    "permissions": [
        {
            "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
            "version": "1",
            "defaultVersion": false,
            "isResourceTypeDefault": false,
            "name": "TestCMP",
            "permissionType": "CUSTOMER_MANAGED",
            "featureSet": "STANDARD",
            "resourceType": "imagebuilder:Component",
            "status": "UNATTACHABLE",
            "creationTime": 1680033769.401,
            "lastUpdatedTime": 1680035597.345
        },
        {
            "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
            "version": "2",
            "defaultVersion": true,
            "isResourceTypeDefault": false,
            "name": "TestCMP",
            "permissionType": "CUSTOMER_MANAGED",
            "featureSet": "STANDARD",
            "resourceType": "imagebuilder:Component",
            "status": "ATTACHABLE",
            "creationTime": 1680035597.346,
            "lastUpdatedTime": 1680035597.346
        }
    ]
}
```

2. After you have the version number to set as default, you can call the set-default-permission-version operation.

   ```
   $ aws ram-cmp set-default-permission-version \
       --permission-arn arn:aws:ram:us-east-1:123456789012:permission/TestCMP \
       --version 2
   ```

   This command returns no output if successful. You can run list-permission-versions again and verify that the `defaultVersion` field of the chosen version is now set to `true`.

Delete a customer managed permission version

You can have up to five versions of each customer managed permission. When a version is no longer needed, and not in use, you can delete it. You can't delete the default version of a customer managed permission. Deleted versions remain visible in the console for up to two hours with a deleted status before they are completely removed.

Console

To delete a customer managed permission version

1. Navigate to the **Managed permissions library**.
2. Filter the list of managed permissions by **Customer managed**, or search for the name of the customer managed permission with the version that you want to delete.
3. Make sure that the version you want to delete isn't currently the default.
4. For the **Versions** section of the page, choose the **Associated resource shares** tab to see if any shares use this version.

   If there are any shares associated, you must change the customer managed permission version before you can delete this version.
5. Choose **Delete version** on the right side of the **Version** section.
6. In the confirmation dialog box, select **Delete** to confirm that you want to delete this version of your customer managed permission.

   Choose **Cancel** if you don't want to delete this version of your customer managed permission.

AWS CLI

To delete one version of a customer managed permission

1. Call the list-permission-versions operation to retrieve the available version numbers.
2. After you have the version number, provide it as a parameter to delete-permission-version.

   ```
   $ aws ram-cmp delete-permission-version \
       --permission-arn arn:aws:ram:us-east-1:123456789012:permission/TestCMP \
       --version 1
   ```

   This command returns no output if successful. You can run list-permission-versions again and verify that the version is no longer included in the output.

Delete a customer managed permission

If a customer managed permission is no longer needed, and not in use, you can delete it. You can't delete a customer managed permission that is associated with a resource share. The deleted customer managed permission disappears after two hours. Until then, it remains visible in the **Managed permission library** with a deleted status.

Console

To delete a customer managed permission

1. Navigate to the **Managed permissions library**.
2. Filter the list of managed permissions by **Customer managed**, or search for the name of the customer managed permission that you want to delete.
3. Confirm there are 0 associated shares from the managed permissions list before selecting the customer managed permission.

   If there are still resource shares associated with the managed permission, you must assign another managed permission to all resource shares before you can continue.
4. In the top right corner of the Customer managed permission details page, choose **Delete managed permission**.
5. When the confirmation dialog box appears, choose **Delete** to delete the managed permission.

AWS CLI

To delete a customer managed permission

1. Find the ARN of the managed permission you want to delete by calling list-permissions with the `--permission-type CUSTOMER_MANAGED` parameter to include only customer managed permissions.

   ```
   $ aws ram-cmp list-permissions --permission-type CUSTOMER_MANAGED
   {
       "permissions": [
           {
               "arn": "arn:aws:ram:us-east-1:123456789012:permission/TestCMP",
               "version": "2",
               "defaultVersion": true,
               "isResourceTypeDefault": false,
               "name": "TestCMP",
               "permissionType": "CUSTOMER_MANAGED",
               "resourceType": "imagebuilder:Component",
               "status": "ATTACHABLE",
               "creationTime": 1680035597.346,
               "lastUpdatedTime": 1680035597.346
           }
       ]
   }
   ```

2. After you have the ARN of the managed permission to delete, provide it as a parameter to delete-permission.

   ```
   $ aws ram delete-permission \
       --permission-arn arn:aws:ram:us-east-1:123456789012:permission/TestCMP
   {
       "returnValue": true,
       "permissionStatus": "DELETING"
   }
   ```

Updating AWS managed permissions to a newer version

Occasionally, AWS updates the AWS managed permissions available to attach to a resource share for a specific resource type. When AWS does this, it creates a new version of the AWS managed permission. Resource shares that include the specified resource type aren't automatically updated to use the latest version of the managed permission. You must explicitly update the managed permission for each resource share. This extra step is required so that you can evaluate the changes before you apply them to your resource shares.

Console

Whenever the console displays a page that lists the permissions associated with a resource share, and one or more of those permissions are using a version other than the default for the permission, the console displays a banner at the top of the console page. The banner indicates that your resource share is using a version other than the default.

In addition, individual permissions can display an **Update to default version** button next to the current version number when that version is not the default.

Choosing that button starts the **Update resource share** wizard. On Step 2 of the wizard you can update the version of any non-default permissions to use their default versions.

The changes are not saved until you complete the wizard by choosing **Submit** on the last page of the wizard.

> **Note**
>
> You can attach only the default version, and you can't revert to another version. For customer managed permissions, after you update the permissions to the default version, you can't apply another version to a resource share unless you first set that other version as the default. For example, if you updated a permission to the default version and then found an error that you wanted to roll back, you could designate the previous version as the default. Alternatively, you could create a different new version and then designate that as the default. After you performed one of those options, you would then update your resource shares to use what is now the default version.

AWS CLI

To update the version of an AWS 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 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 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. You don't need to specify the version number; the default version is automatically used.

```
$ 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 the previous step for each `ResourceShareArn` that you received in the results from the command in step 1.

Considerations for using customer managed permissions in AWS RAM

Customer managed permissions are only available in the AWS Region that you create them in. Not all resource types support customer managed permissions. For a list of supported resource types in AWS Resource Access Manager, see Shareable AWS resources.

Customer managed permissions with multiple statements aren't supported. You can only use single non-negating operators in customer managed permissions.

The following conditions aren't supported in customer managed permissions:

- Condition keys used to match properties of the principal:
  - `aws:PrincipalOrgId`
  - `aws:PrincipalOrgPaths`
  - `aws:PrincipalAccount`

- Condition keys used to restrict access for service principals:
  - `aws:SourceArn`
  - `aws:SourceAccount`
  - `aws:SourceOrgPaths`
  - `aws:SourceOrgID`
- System tags:
  - `aws:PrincipalTag/aws:`
  - `aws:ResourceTag/aws:`
  - `aws:RequestTag/aws:`

> **Note**
>
> The `aws:SourceAccount` value is automatically populated when sharing to service principals.

How managed permissions work

For a quick overview, watch the following video that demonstrates how managed permissions let you apply the best practice of least privilege access to your AWS resources.

This video demonstrates how to author and associate customer managed permissions following the best practice of least privilege. For more information see, ???.

When you create a resource share, you associate an AWS managed permission with each resource type that you want to share. If the managed permission has more than one version, the new resource share always uses the version designated as the default.

After you create the resource share, AWS RAM uses the managed permission to generate a resource-based policy that is attached to each shared resource.

The policy template in a managed permission specifies the following:

Effect

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

Action

The list of operations 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 permission. For more information, see Action in the *IAM User Guide*.

Condition

When and how a principal can interact with a resource in a resource share. Conditions add an extra layer of security to your shared resources. Use them to limit access for sensitive actions to your shared resources. For example, you can include conditions requiring the actions to originate from a specific corporate IP address range, or that the actions must be performed by users authenticated with multi-factor authentication. For more information about conditions, see AWS global condition context keys in the *IAM User Guide*. For more information about service-specific conditions, see Actions, resources, and condition keys for AWS services in the *Service Authorization Reference*.

> **Note**
>
> Conditions are available for customer managed permissions and supported resource types for AWS managed permissions.
> For information about conditions that are excluded from use with customer managed permissions, see Considerations for using customer managed permissions in AWS RAM.

Types of managed permissions

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

- **AWS managed permission** – There is one default managed permission available for every resource type that AWS RAM supports. The default managed permission is the one used for a resource type unless you explicitly choose one of the additional managed permissions. The default managed permission is intended to support the most common customer scenarios for sharing resources of the specified type. 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 AWS managed permissions use the following format: `AWSRAMDefaultPermission`*`ShareableResourceType`*. For example, for the `ec2:Subnet` resource type, the name of the default AWS managed permission is `AWSRAMDefaultPermissionSubnet`.

  > **Note**
  >
  > The default managed permission is separate from the default *version* of a managed permission. All managed permissions, whether default or one of the additional managed permissions supported by some resource types, are separate, complete permissions with different effects and actions that support different sharing scenarios, such as read-write versus read-only access. Any managed permission, whether AWS or customer managed can have multiple versions, one of which is the default version for that permission.

  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 create one resource share for the administrator with the full access managed permission. You can then create a separate resource share for other developers using the read-only managed permission to follow the practice of granting least privilege.

> **Note**
>
> All AWS services that work with AWS RAM support at least one default managed permission. You can view the available permissions for each AWS service on the **Managed 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 managed permissions.
>
> 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. If supported, you will also have the option to choose **Create customer managed permission** on the **Associate managed permissions** page.

- **Customer managed permission –** Customer managed permissions are managed permissions that you author and maintain by precisely specifying which actions can be performed under which conditions with resources shared using AWS RAM. For example, you want to limit read access for your Amazon VPC IP Address Manager (IPAM) pools, which help you manage your IP addresses at scale. You can create customer managed permissions for your developers to assign IP addresses, but not view the range of IP addresses other developer accounts assign. You can follow the best practice of least privilege, granting only the permissions required to perform tasks on shared resources.

Security in AWS Resource Access Manager

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 Resource Access Manager

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. You are also responsible for the security configuration and management tasks for the AWS services that you use. For more information about data privacy, see the Data Privacy FAQ. For information about data protection in Europe, see the AWS Shared Responsibility Model and GDPR blog post on the *AWS Security Blog*.

For data protection purposes, we recommend that you protect AWS account credentials and set up individual users with AWS IAM Identity Center or AWS Identity and Access Management (IAM). That way, each user is given only the permissions necessary to fulfill their job duties. We also recommend that you secure your data in the following ways:

- Use multi-factor authentication (MFA) with each account.
- Use SSL/TLS to communicate with AWS resources. We require TLS 1.2 and recommend TLS 1.3.
- Set up API and user activity logging with AWS CloudTrail. For information about using CloudTrail trails to capture AWS activities, see Working with CloudTrail trails in the *AWS CloudTrail User Guide*.
- Use AWS encryption solutions, along with all default security controls within AWS services.
- Use advanced managed security services such as Amazon Macie, which assists in discovering and securing sensitive data that is stored in Amazon S3.
- If you require FIPS 140-3 validated cryptographic modules when accessing AWS through a command line interface or an API, use a FIPS endpoint. For more information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-3.

We strongly recommend that you never put confidential or sensitive information, such as your customers' email addresses, into tags or free-form text fields such as a **Name** field. This includes when you work with 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 text fields used for names may be used for billing or diagnostic logs. If you provide a URL to an external server, we strongly recommend that you do not include credentials information in the URL to validate your request to that server.

Identity and access management for AWS Resource Access Manager

AWS Identity and Access Management (IAM) is an AWS service that helps an administrator securely control access to AWS resources. Administrators in IAM control who can be *authenticated* (signed in) and *authorized* (have permissions) to use AWS resources. By using IAM, you create principals, such as roles, users, and groups in your AWS account. You control the permissions that those principals 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 in the *IAM User Guide*.

Topics

How AWS RAM works with IAM

By default, IAM principals don't have permission to create or modify AWS RAM resources. To allow IAM principals to create or modify resources and perform tasks, you perform one of the following steps. These actions grant permission to use specific resources and API actions.

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

- Users and groups in AWS IAM Identity Center:

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

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

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

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 Resource Access Manager.

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 roles and users, 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

An IAM permission 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":{
            "<comparison-operator>":{
                "<key>":"<value>"
            }
        }
    }]
}
```

Effect

The *Effect* statement indicates whether the policy allows or denies a principal 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/${TagKey}` – Tests whether the service request includes a tag with the specified tag key exists and has the specified value.
- `aws:ResourceTag/${TagKey}` – Tests whether the resource acted on by the service request has an attached tag with a tag key that you specify in the policy.

  The following example condition checks that the resource referenced in the service request has an attached tag with the key name "Owner" and a value of "Dev Team".

  ```
  "Condition" : {
      "StringEquals" : {
          "aws:ResourceTag/Owner" : "Dev Team"
      }
  }
  ```

- `aws:TagKeys` – Specifies the tag keys that must be used to create or tag a resource share.
- `ram:AllowsExternalPrincipals` – Tests whether the resource share in the service request allows sharing with external principals. An external principal is an AWS account outside of your organization in AWS Organizations. If this evaluates to `False`, then you can share this resource share with accounts only in the same organization.
- `ram:PermissionArn` – Tests whether the permission ARN specified in the service request matches an ARN string that you specify in the policy.
- `ram:PermissionResourceType` – Tests whether the permission specified in the service request is valid for the resource type that you specify in the policy. Specify resource types using the format shown in the list of shareable resource types.
- `ram:Principal` – Tests whether the ARN of the principal specified in the service request matches an ARN string that you specify in the policy.
- `ram:RequestedAllowsExternalPrincipals` – Tests whether the service request includes the `allowExternalPrincipals` parameter and whether its argument matches the value you specify in the policy.
- `ram:RequestedResourceType` – Tests whether the resource type of the resource being acted on matches a resource type string that you specify in the policy. Specify resource types using the format shown in the list of shareable resource types.
- `ram:ResourceArn` – Tests whether the ARN of the resource being acted upon by the service request matches an ARN that you specify in the policy.
- `ram:ResourceShareName` – Tests whether the name of the resource share being acted upon by the service request matches a string that you specify in the policy.
- `ram:ShareOwnerAccountId` – Tests the account ID number of the resource share being acted upon by the service request matches a string that you specify in the policy.

AWS managed policies for AWS Resource Access Manager

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

AWS managed policies

- AWS managed policy: AWSResourceAccessManagerReadOnlyAccess
- AWS managed policy: AWSResourceAccessManagerFullAccess
- AWS managed policy: AWSResourceAccessManagerResourceShareParticipantAccess

- AWS managed policy: AWSResourceAccessManagerServiceRolePolicy
- AWS RAM updates to AWS managed policies

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

An AWS managed policy is a standalone policy that is created and administered by AWS. AWS managed policies are designed to provide permissions for many common use cases so that you can start assigning permissions to users, groups, and roles.

Keep in mind that AWS managed policies might not grant least-privilege permissions for your specific use cases because they're available for all AWS customers to use. We recommend that you reduce permissions further by defining customer managed policies that are specific to your use cases.

You cannot change the permissions defined in AWS managed policies. If AWS updates the permissions defined in an AWS managed policy, the update affects all principal identities (users, groups, and roles) that the policy is attached to. AWS is most likely to update an AWS managed policy when a new AWS service is launched or new API operations become available for existing services.

For more information, see AWS managed policies in the *IAM User Guide*.

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.

JSON

```
{
    "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.

JSON

```
{
    "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.

JSON

```
{
    "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.

JSON

```
{
    "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.

Call `enable-sharing-with-aws-organizations` to create the service linked role in your account.

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
- Example 2: Allow sharing of specific resource types
- Example 3: Restrict sharing with external AWS accounts

Example 1: Allow sharing of specific resources

You can use an IAM permission 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.

For more information about when and why to use `...IfExists` operators, see ...IfExists condition operators in the *IAM User Guide*.

JSON

```
{
    "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.

The actions, `AssociateResourceShare` and `CreateResourceShare`, can accept principals and `resourceArns` as independent input parameters. Therefore, AWS RAM authorizes each principal and resource independently, so there could be multiple request contexts. This means when a principal is being associated to a AWS RAM resource share, the `ram:RequestedResourceType` condition key is not present in the request context. Similarly, when a resource is being associated to a AWS RAM resource share, the `ram:Principal` condition key is not present in the request context. Therefore, to allow `AssociateResourceShare` and `CreateResourceShare` when associating principals to the AWS RAM resource share, you can use the `Null` condition operator.

For example, the following policy limits principals to sharing only Amazon Route 53 resolver rules and allows them to associate any principal to that share.

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [{
        "Sid": "AllowOnlySpecificResourceType",
        "Effect": "Allow",
        "Action": ["ram:CreateResourceShare", "ram:AssociateResourceShare"],
        "Resource": "*",
        "Condition": {
            "StringEquals": {
                "ram:RequestedResourceType": "route53resolver:ResolverRule"
            }
        }
    },
    {
    "Sid": "AllowAssociatingPrincipals",
     "Effect": "Allow",
        "Action": ["ram:CreateResourceShare", "ram:AssociateResourceShare"],
        "Resource": "*",
        "Condition": {
            "Null": {
                "ram:Principal": "false"
             }
        }
    }
  ]
}
```

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.

JSON

```
{
    "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

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.

AWS RAM authorizes APIs separately for each principal and resource listed in the call.

JSON

```
{
    "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.

JSON

```
{
    "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.

AWS RAM authorizes APIs separately for each principal and resource listed in the call.

The operator `StringEqualsIfExists` allows a request if either the request doesn't include a resource type parameter, or if it does include that parameter, that its value exactly matches the specified resource type. If you're including a principal you must have `...IfExists`.

For more information about when and why to use `...IfExists` operators, see [...IfExists condition operators]() in the *IAM User Guide*.

JSON

```
{
    "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.

AWS RAM authorizes APIs separately for each principal and resource listed in the call.

JSON

```
{
    "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`.

If you're using an `"Effect": "Deny"` element with a negated condition operator, like `StringNotEqualsIfExists`, the request is still denied even if the condition key is not present. Use a `Null` condition operator to check if a condition key is absent at the time of authorization.

AWS RAM authorizes APIs separately for each principal and resource listed in the call.

JSON

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

Example 6: Prevent resource shares with RetainSharingOnAccountLeaveOrganization enabled

The following SCP prevents users from creating or modifying resource shares when the `ram:RetainSharingOnAccountLeaveOrganization` condition key is set to `true`.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Deny",
            "Action": [
                "ram:CreateResourceShare",
                "ram:AssociateResourceShare",
                "ram:DisassociateResourceShare"
            ],
            "Resource": "*",
            "Condition": {
                "Bool": {
                    "ram:RetainSharingOnAccountLeaveOrganization": "true"
                }
            }
        }
    ]
}
```

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. External accounts (accounts added to the resource share via invitation) will not be impacted, and will continue to be associated with the resource share.

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 EventBridge

Delivers a near-real-time stream of system events that describe changes in AWS resources. EventBridge 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 EventBridge.

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.

Monitoring AWS RAM using EventBridge

Using Amazon EventBridge, you can set up automatic notifications for specific events in AWS RAM. Events from AWS RAM are delivered to EventBridge in near-real time. You can configure EventBridge 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 EventBridge 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 EventBridge 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 that helps you determine who made the request.

- AWS account root credentials
- Temporary security credentials from an AWS Identity and Access Management (IAM) role or federated user.
- Long-term security credentials from an IAM user.
- Another AWS service.

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"
}
```

Compliance validation for AWS Resource Access Manager

To learn whether an AWS service is within the scope of specific compliance programs, see AWS services in Scope by Compliance Program and choose the compliance program that you are interested in. For general information, see AWS Compliance Programs.

You can download third-party audit reports using AWS Artifact. For more information, see Downloading Reports in AWS Artifact.

Your compliance responsibility when using AWS services is determined by the sensitivity of your data, your company's compliance objectives, and applicable laws and regulations. For more information about your compliance responsibility when using AWS services, see AWS Security Documentation.

Resilience in AWS Resource Access Manager

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 Resource Access Manager

As a managed service, AWS Resource Access Manager is protected by AWS global network security. For information about AWS security services and how AWS protects infrastructure, see AWS Cloud Security. To design your AWS environment using the best practices for infrastructure security, see Infrastructure Protection in *Security Pillar AWS Well‐Architected Framework*.

You use AWS published API calls to access AWS RAM through the network. Clients must support the following:

- Transport Layer Security (TLS). We require TLS 1.2 and recommend TLS 1.3.
- Cipher suites with perfect forward secrecy (PFS) such as DHE (Ephemeral Diffie-Hellman) or ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Most modern systems such as Java 7 and later support these modes.

Access AWS Resource Access Manager using an interface endpoint (AWS PrivateLink)

You can use AWS PrivateLink to create a private connection between your VPC and AWS Resource Access Manager. You can access AWS RAM as if it were in your VPC, without the use of an internet gateway, NAT device, VPN connection, or Direct Connect connection. Instances in your VPC don't need public IP addresses to access AWS RAM.

You establish this private connection by creating an *interface endpoint*, powered by AWS PrivateLink. We create an endpoint network interface in each subnet that you enable for the interface endpoint. These are requester-managed network interfaces that serve as the entry point for traffic destined for AWS RAM.

For more information, see Access AWS services through AWS PrivateLink in the *AWS PrivateLink Guide*.

Considerations for AWS RAM

Before you set up an interface endpoint for AWS RAM, review Considerations in the *AWS PrivateLink Guide*.

AWS RAM supports making calls to all of its API actions through the interface endpoint.

VPC endpoint policies are supported for AWS RAM. By default, full access to AWS RAM is allowed through the interface endpoint.

Create an interface endpoint for AWS RAM

You can create an interface endpoint for AWS RAM using either the Amazon VPC console or the AWS Command Line Interface (AWS CLI). For more information, see Create an interface endpoint in the *AWS PrivateLink Guide*.

Create an interface endpoint for AWS RAM using the following service name:

```
com.amazonaws.region.ram
```

If you enable private DNS for the interface endpoint, you can make API requests to AWS RAM using its default Regional DNS name. For example, `ram.us-east-1.amazonaws.com`.

Create an endpoint policy for your interface endpoint

An endpoint policy is an IAM resource that you can attach to an interface endpoint. The default endpoint policy allows full access to AWS RAM through the interface endpoint. To control the access allowed to AWS RAM from your VPC, attach a custom endpoint policy to the interface endpoint.

An endpoint policy specifies the following information:

- The principals that can perform actions (AWS accounts, IAM users, and IAM roles).
- The actions that can be performed.
- The resources on which the actions can be performed.

For more information, see Control access to services using endpoint policies in the *AWS PrivateLink Guide*.

Example: VPC endpoint policy for AWS RAM actions

The following is an example of a custom endpoint policy. When you attach this policy to your interface endpoint, it grants access to the listed AWS RAM actions for all principals on all resources.

JSON

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

Troubleshooting issues with AWS RAM

Use the information in this section of the guide to help you diagnose and fix common issues when you work with AWS Resource Access Manager (AWS RAM).

Topics

Error: "Your account ID does not exist in an AWS organization"

Scenario

You get the error "Your account ID does not exist in an AWS organization" when trying to share a resource with accounts or organizational units (OUs) in your organization.

Cause

This error can happen if the service-linked role AWSServiceRoleForResourceAccessManager isn't successfully created when you turn on integration between AWS Resource Access Manager and AWS Organizations.

Solution

To re-create the required service-linked role, perform the following steps to turn off integration and then turn it on again.

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

1. Sign in to your the management account of your organization using an IAM role or user with administrative permissions.
2. Navigate to the Services page in the AWS Organizations console.
3. Choose **RAM**.
4. Choose **Disable trusted access**.
5. Navigate to the Settings page in the AWS RAM console.
6. Select the box **Enable sharing with AWS Organizations**, and then choose **Save settings**.

You should now be able to use AWS RAM to share your resources with accounts and OUs in the organization.

Error: "AccessDeniedException"

Scenario

You get an Access Denied exception when trying to share a resource or view a resource share.

Cause

You can receive this error if you attempt to create a resource share when you don't have the required permissions. This can be caused by insufficient permissions in policies attached to your AWS Identity and Access Management (IAM) principal. It can also happen because of restrictions in place from an AWS Organizations service control policy (SCP) that affects your AWS account.

Solution

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

- Users and groups in AWS IAM Identity Center:

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

- Users managed in IAM through an identity provider:

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

To resolve the error, you need to ensure the permissions are granted by `Allow` statements in the permission policy used by the principal that makes the request. In addition, the permissions must not be blocked by your organization's SCPs.

To **create** a resource share, you need the following two permissions:

- `ram:CreateResourceShare`
- `ram:AssociateResourceShare`

To **view** a resource share, you need the following permission:

- `ram:GetResourceShares`

To **attach permissions** to a resource share, you need the following permission:

- *`resourceOwningService`*`:`*`PutPolicyAction`*

  This is a placeholder. You must replace it with the "PutPolicy" permission (or equivalent) for the service that owns the resource that you want to share. For example, if you are sharing a Route 53 resolver rule, then the required permission would be: `route53resolver:PutResolverRulePolicy`. If you want to allow the creation of a resource share that contains multiple resource types, then you must include the relevant permission for each resource type that you want to permit.

The following example shows what such an IAM permission policy might look like.

JSON

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

Error: "UnknownResourceException"

Scenario

You get one of the following errors:

- "CannotCreateResourceShare: UnknownResourceException: OrganizationalUnit ou-*xxxx* could not be found"
- "CannotUpdateResourceShare: UnknownResourceException: OrganizationalUnit ou-*xxxx* could not be found".

Cause

These errors can occur if you enable integration between AWS RAM and AWS Organizations by using either the Organizations console or the Organizations EnableAWSServiceAccess API instead of by using the AWS RAM console. When you enable integration by using the Organizations console or API, the service doesn’t create the `AWSServiceRoleForResourceAccessManager` role in your account. That role is needed to access information about your organization. Because the role wasn't created, AWS RAM can’t access details about the accounts or organizational units (OUs) in your organization.

Solution

To resolve the issue, turn off integration between AWS RAM and AWS Organizations. Then turn it on again by calling the AWS RAM EnableSharingWithAwsOrganization API operation, or by using the AWS Management Console to perform the following steps.

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

1. Sign in to your the management account of your organization using an IAM role or user with administrative permissions.
2. Navigate to the Services page in the AWS Organizations console.
3. Choose **RAM**.
4. Choose **Disable trusted access**.
5. Navigate to the Settings page in the AWS RAM console.
6. Select the box **Enable sharing with AWS Organizations**, and then choose **Save settings**.

You should now be able to use AWS RAM to share your resources with accounts and OUs in the organization.

Errors when trying to share with accounts outside of my organization

Scenario

You get one of the following errors when you try to share resources with accounts that are outside of your organization:

- "You cannot share the resource outside your organization."
- "The resource you are attempting to share can only be shared within your AWS Organization."

- "InvalidParameterException: Principal Account-ID is not in your AWS organization. You do not have permission to add external AWS accounts to a resource share."
- "OperationNotPermittedException: The resource you are attempting to share can only be shared within your AWS Organization."

Possible causes and solutions

Some resource types can be shared only with accounts in the same organization

Some resource types can’t be shared with any account that isn't a member of that organization. An example resource type with this restriction is virtual private connections (VPCs) that are part of Amazon Elastic Compute Cloud (Amazon EC2).

To verify if you can share a particular resource type with accounts and principals outside of your organization, see Shareable AWS resources.

The service-linked role wasn't successfully created

This issue can occur if the service-linked role `AWSServiceRoleForResourceAccessManager` wasn't successfully created when you turned on integration between AWS RAM and AWS Organizations.

If you receive one of these errors when attempting to share a resource with an account that ***is*** part of your organization, perform the following steps to delete and re-create the service-linked role.

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

1. Sign in to your the management account of your organization using an IAM role or user with administrative permissions.
2. Navigate to the Services page in the AWS Organizations console.
3. Choose **RAM**.
4. Choose **Disable trusted access**.
5. Navigate to the Settings page in the AWS RAM console.
6. Select the box **Enable sharing with AWS Organizations**, and then choose **Save settings**.

Can't see shared resources in the destination account

Scenario

Users can't see the resources that they believe are shared with them from other AWS accounts.

Possible causes and solutions

Sharing with AWS Organizations was turned on by using Organizations instead of AWS RAM

If AWS Organizations was turned on by using Organizations instead of AWS RAM, then sharing within the organization fails. To check if this is the cause of the problem, navigate to the Settings page in the AWS RAM console and verify that the **Enable sharing with AWS Organizations** checkbox is selected.

- If the checkbox is selected, then this is not the cause.
- If the checkbox is not selected, then this might be the cause. *Don't select the checkbox yet*. Perform the following steps to correct the situation.

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

1. Sign in to your the management account of your organization using an IAM role or user with administrative permissions.
2. Navigate to the Services page in the AWS Organizations console.
3. Choose **RAM**.
4. Choose **Disable trusted access**.
5. Navigate to the Settings page in the AWS RAM console.
6. Select the box **Enable sharing with AWS Organizations**, and then choose **Save settings**.

You might need to update the share and specify the accounts or organizational units within the organization to share with.

The resource share doesn't specify this account as a principal

In the AWS account that created the resource share, view the resource share in the AWS RAM console. Verify that the account that can't access the resources is listed as a **Principal**. If it isn't, then update the share to add the account as a principal.

The role or user in the account doesn't have required minimum permissions

When you share a resource in account A to another account B, roles and users in account B don't automatically get access to the resources in the share. The administrator of account B must first grant permission to the IAM roles and users in account B who need to access the resource. As an example, the following policy shows how you might grant read-only access to roles and users in account B for a resource from account A. The policy specifies the resource by its Amazon Resource Name (ARN).

JSON

```
{
    "Version":"2012-10-17",
    "Statement": [
        {
            "Action": [
                "ram:Get*",
                "ram:List*"
            ],
            "Effect": "Allow",
            "Resource": "arn:aws:<service>:us-east-1:<Account-A-ID>:<resource-id>"
        }
    ]
}
```

The resource is in a different AWS Region than the current console setting

AWS RAM is a Regional service. Resources exist in a specific AWS Region, and to see them, the AWS Management Console must be configured to view the resources in that Region.

The AWS Region that the console is currently accessing is displayed in the upper-right corner of the console. To change it, choose the current Region name and from the dropdown menu, choose the Region whose resources you want to see.

Error: Limit exceeded

Scenario

You receive "You have reached the limit on the number of resources you can share" or "ResourceShareLimitExceededException" when trying to share resources.

Cause

These errors occur when you reach the maximum number of resources you can share using either the AWS RAM service or the AWS service that created the resource you're trying to share. This quota (formerly referred to as a limit) can affect both the sharing account or the account you're sharing the resource with.

Solution

1. To view your quotas, in the AWS account where you are seeing the error, navigate to one of the following pages, depending on the type of quota you're reaching:
   - The AWS RAM page in the Service Quotas console
   - The page for the AWS service whose resources are impacted by the quota
2. Scroll down and choose the relevant quota.
3. If it's available for this quota, choose **Request quota increase**.
4. Enter a new value for the quota, and then choose **Request**.
5. The request appears on the Quota request history page, where you can check on the status of the request until it's finalized.

The other account in my organization never receives an invitation

Scenario

When you share resources with another account in the same organization managed by AWS Organizations, they don't receive invitations.

Cause

This is **expected behavior** if your account has sharing within the AWS organization turned on.

When this option is turned on and you share with another account in your organization, no invitations are sent and no acceptance is required. All organization accounts that you reference as principals in the resource share can immediately begin accessing the resources in the share.

If your account has *not* turned on sharing within the AWS organization, then when you share with other accounts, even if they are in the same AWS organization, they are treated as standalone accounts. Invitations are sent and must be accepted before users can access the resources in the shares.

You can't share a VPC subnet

Scenario

When you attempt to use AWS RAM to share a VPC subnet with another account, the sharing operation succeeds. However, the consuming account shows `LIMIT EXCEEDED` for that resource in the AWS RAM console.

Cause

Some individual resource types have service-specific restrictions separate from the restrictions enforced by AWS RAM. Some of those restrictions can effectively prevent sharing even if you haven't reached one of the restrictions in AWS RAM. Limits are an example of these restrictions. Amazon Virtual Private Cloud (Amazon VPC) limits the number of subnets that you can share with another individual account. If you try to share a subnet with a consuming account that already contains the maximum number of subnets, then that consuming accounts displays `LIMIT EXCEEDED` in the console for that resource. For more information about this limit, see Amazon VPC Quotas – VPC sharing in the *Amazon Virtual Private Cloud User Guide*.

To address this, first check for other resource shares that might be sharing the specified resource with the affected account, and remove those shares that you might no longer require. You can also request an increase for a limit that supports adjusting. Use the Service Quotas console to request a limit increase.

Note

AWS RAM doesn't automatically detect limit increase changes. You must re-associate the resource or principal to the resource share for RAM to detect the change.

Service quotas for AWS RAM

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

Note

The following definitions apply to the description in the quotas below:

- **Resource** – An individual AWS service-created element that you want to share, such as an Amazon S3 bucket or an Amazon EC2 instance. Each resource referenced in a resource share counts as one against this quota. If you share the same resource in three different resource shares, it increases your count for this quota by three.
- **Resource share** – An AWS RAM created container that you can use to share resources. Each resource share, regardless of how many resources it contains, counts as one against your quota.
- **Shared principal** – An identifier that you've associated with a resource share. This can be an AWS Identity and Access Management (IAM) role or user, an AWS account identifier, an organizational unit, or an entire organization. Each shared principal that you reference in a resource share adds one to your quota use. If you share with an entire organization by referencing its ID, it counts as only one against this quota.
- **Customer managed permission** – Managed permissions that you create to address specific use cases using least privilege access to manage how your shared resources are used.

Resource	Default limit
Maximum number of resource shares per AWS Region	25,000
Maximum number of resource associations per resource share	5,000
Maximum number of principal associations per resource share	5,000

Resource	Default limit
Maximum number of customer managed permissions	1,500
Maximum number of customer managed permissions per resource type	10
Maximum number of versions per customer managed permission	5
Maximum number of resource associations across all resource shares in an AWS Region **Note** Each resource included in a resource share counts against this limit. If a resource is included in 10 different resource shares, that counts 10 against the limit.	25,000
Maximum number of principal associations across all resource shares in an AWS Region **Note** Each principal included in a resource share counts against this limit. If a principal is included in 10 different resource shares, that counts 10 against the limit.	25,000

Resource	Default limit
Maximum number of pending invitations per sharing account • This quota applies to only **sending** accounts who are sharing with accounts that are not part of the same AWS Organizations. • There is no quota to limit how many pending invitations a receiving account can have. • Invitations are *not* used when sharing between accounts that are part of the same AWS Organizations and you've turned on resource sharing within the AWS Organizat ions.	250

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
AWS SDK for C++	AWS SDK for C++ code examples
AWS SDK for Go	AWS SDK for Go code examples
AWS SDK for Java	AWS SDK for Java code examples
AWS SDK for JavaScript	AWS SDK for JavaScript code examples
AWS SDK for .NET	AWS SDK for .NET code examples
AWS SDK for PHP	AWS SDK for PHP code examples
AWS SDK for Python (Boto3)	AWS SDK for Python (Boto3) code examples
AWS SDK for Ruby	AWS SDK for Ruby code examples

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.

Change	Description	Date
Added support for sharing Amazon CloudFront resources	You can now share Amazon CloudFront VPC Origins with other AWS accounts within your organization.	October 6, 2025
Added support to share Billing and Cost Management resources	You can now share Billing and Cost Management dashboard s with other AWS accounts or your organization with AWS RAM.	August 19, 2025
Added support for sharing AWS Cloud Map resources	You can now share AWS Cloud Map namespaces with other AWS accounts within your organization.	August 14, 2025
Added support to share Amazon Application Recovery Controller (ARC) resources	You can now share Amazon Application Recovery Controller (ARC) plans with other AWS accounts or your organization with AWS RAM.	July 31, 2025
Added support for sharing Oracle Database@AWS resources	You can now share Oracle Database@AWS Exadata infrastructure and ODB networks with other AWS accounts within your organization.	June 30, 2025

Added support for sharing Multi-party approval resources	You can now share Multi-party approval approval teams with other AWS accounts or within your organization.	June 17, 2025
Added support for sharing Amazon SageMaker AI resources	You can now use AWS RAM to share Amazon SageMaker AI Partner Apps with other AWS accounts and with your organization.	June 6, 2025
Added support for sharing AWS Network Firewall resources	You can now use AWS RAM to share AWS Network Firewall firewalls with other AWS accounts and with your organization.	May 28, 2025
Added support for sharing AWS Systems Manager resources	You can share a AWS Systems Manager deny-access policy with other AWS accounts or your organizations with AWS RAM.	April 30, 2025
Added support for sharing AWS CodeConnections resources	You can now share AWS CodeConnections code connections with other AWS accounts or within in your organization.	March 5, 2025
Added support for sharing AWS Billing resources	You can now share AWS Billing views with other AWS accounts in your organization.	December 20, 2024
Added support for sharing Amazon VPC Lattice resource configurations	You can now share Amazon VPC Lattice resource configurations with other AWS accounts.	December 1, 2024

Added support for sharing Amazon API Gateway resources	You can now share API Gateway domain names with other AWS accounts or within your organization.	November 21, 2024
Added support for sharing Amazon VPC resources	You can now share Amazon VPC Security groups with other AWS accounts or within your organization.	October 30, 2024
Added support for sharing AWS End User Messaging SMS resources	You can share AWS End User Messaging SMS resources with other AWS accounts or your organizations with AWS RAM.	September 24, 2024
AWS PrivateLink	With AWS PrivateLink for AWS RAM, you can connect directly to RAM by using an interface endpoint in your virtual private cloud (VPC).	September 9, 2024
Added support for sharing AWS Backup	You can share logically air-gapped vaults across AWS accounts or within your organization.	August 7, 2024
Added support for sharing Elastic Load Balancing resources	You can share Elastic Load Balancing trust stores across AWS accounts or within your organization.	August 5, 2024
Added support for sharing Amazon Bedrock Custom Models	You can now use AWS RAM to share Amazon Bedrock custom models with other AWS accounts and with your organization.	August 1, 2024

Added support for sharing AWS CloudHSM Backups	You can share AWS CloudHSM Backups with other AWS accounts or your organizat ions with AWS RAM.	June 28, 2024
Added support to share Amazon SageMaker AI Model Registry resources.	You can now share advanced parameters securely and efficiently across AWS accounts or within your organization.	June 27, 2024
Added support to share Amazon SageMaker AI JumpStart	You can now share Amazon SageMaker AI JumpStart Hubs with AWS accounts or within your organization.	June 27, 2024
Added support for sharing Amazon Route 53 Resolver Profiles	You can now use AWS RAM to share Amazon Route 53 Resolver Profiles with other AWS accounts within your organization.	April 22, 2024
Added support to share AWS Systems Manager Parameter Store resources	You can now share advanced parameters securely and efficiently across AWS accounts or within your organization.	February 21, 2024
Added support to share Amazon FSx for OpenZFS Snapshots	You can now share Amazon FSx for OpenZFS Snapshots with other AWS accounts within your organization.	December 19, 2023
Added support to share Amazon Simple Storage Service resources	You can now share Amazon Simple Storage Service Access Grants Instance with other AWS accounts or your organization with AWS RAM.	November 27, 2023

Added support to share AWS Resource Explorer views	You can now share AWS Resource Explorer views with other AWS accounts within your organization.	November 14, 2023
Added support to share Amazon Application Recovery Controller (ARC) resources	You can now share Amazon Application Recovery Controller (ARC) clusters with other AWS accounts or your organization with AWS RAM.	October 18, 2023
Added support to share Amazon DataZone resources	You can now share Amazon DataZone resources with other AWS accounts or your organization.	October 4, 2023
Added support for service principal sharing	You can now associate service principals to resource shares. This allows specified services to manage necessary actions for customer resources on your behalf.	August 29, 2023
Added support to share SageMaker Model Card resources	You can now share SageMaker Model Card resources with other AWS accounts or your organization.	August 18, 2023
Added support for Amazon SageMaker AI Feature Store feature groups and SageMaker AI Catalog as shareable resources	You can now share Amazon SageMaker AI Feature Store feature groups and SageMaker AI Catalog resources with other AWS accounts or your organization.	July 20, 2023

Service quota limit increase for pending invitations	The maximum number of pending invitations per sharing account has been increased from 20 to 250.	June 8, 2023
Added support for AWS AppSync GraphQL APIs as shareable resources	You can now share AWS AppSync GraphQL APIs with other AWS accounts with AWS RAM.	May 24, 2023
Added support for AWS Verified Access groups as shareable resources	You can now create and manage AWS Verified Access groups centrally, and then share them with other AWS accounts or your organization.	April 27, 2023
Added support for customer managed permission in the AWS RAM console	You can now securely author and maintain fine-grained resource access controls for supported resource types.	April 19, 2023
Added support for Amazon VPC Lattice service and service network shareable resources	You can now share Amazon VPC Lattice service and service network resources with other AWS accounts.	March 31, 2023
Added support for AWS Marketplace Catalog entities as shareable resources	You can now share your entities with other AWS accounts in the Marketplace.	March 27, 2023
Added support for managing permission versions in the AWS RAM console	You can now use the AWS RAM console to view version details and to update permissions to whichever version is designated as the default.	January 16, 2023

IAM best practices update	Updated guide to align with the IAM best practices. For more information, see Security best practices in IAM.	January 3, 2023
Added support for Amazon EC2 placement groups as shareable resources	You can now share Amazon EC2 placement groups with other AWS accounts to launch their instances in.	November 8, 2022
Added links to two introductory videos about AWS RAM	Added overview videos that describe AWS RAM and provide a walk-through of sharing a resource with other AWS accounts.	August 29, 2022
Added support for Amazon SageMaker AI pipelines	You can now share SageMaker AI pipelines with other AWS accounts.	August 2, 2022
Added support for AWS Service Catalog AppRegistry applications and attribute groups as shareable resource types	You can now share AppRegistry applications and attribute groups with other AWS accounts.	June 17, 2022
AWS Resource Access Manager receives SOC and ISO certification	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

AWS Resource Access Manager receives FedRAMP certification	AWS RAM has been validated as being compliant with the Federal Risk and Authoriza tion Management Program (FedRAMP).	April 8, 2022
AWS Resource Access Manager receives PCI DSS certification	AWS RAM has been validated as being compliant with Payment Card Industry (PCI) Data Security Standard (DSS).	February 27, 2022
Added support for Amazon VPC IPAM resource discoveries as shareable resources. Also, you can now share IPAM Pools with accounts outside of an organization	You can now share IPAM resource discoveries with other AWS accounts.	January 25, 2022
Added support for sharing global resources	You can now share global resources with other AWS accounts.	December 2, 2021
Added support for AWS Cloud WAN core networks as shareable global resources	You can now share Cloud WAN core networks with other AWS accounts.	December 2, 2021
Support for sharing Amazon VPC IP Address Manager (IPAM) pools	You can use AWS RAM to share Amazon VPC IPAM pools. For more information, see Sharable AWS resources in the *AWS RAM User Guide*.	December 1, 2021
Support for sharing Amazon SageMaker AI resources	You can use AWS RAM to share SageMaker AI lineage groups. For more information, see Sharable AWS resources in the *AWS RAM User Guide*.	November 30, 2021

Support for sharing AWS Migration Hub Refactor Spaces resources	You can use AWS RAM to share Migration Hub environments. For more information, see Sharable AWS resources in the *AWS RAM User Guide*.	November 29, 2021
Added information about AWS RAM AWS-managed IAM permission policies	Published details about the available AWS-managed permission policies that you can access in the IAM console and attach to the IAM principals in your AWS account.	September 16, 2021
Added support for sharing S3 on Outposts resources	You can now use AWS RAM to share S3 on Outposts with other AWS accounts.	August 5, 2021
Added support for additional managed permissions and sharing resources with IAM principals	For supported resource types, you can choose from additional AWS RAM managed permissions and share resources with individual IAM roles and users.	June 10, 2021
Added support for sharing AWS Systems Manager Incident Manager resources	You can now use AWS RAM to share AWS Systems Manager Incident Manager contacts and response plans with other AWS accounts.	May 10, 2021
Added support for sharing Amazon Route 53 resources	You can now use AWS RAM to share Amazon Route 53 Resolver DNS Firewall rule groups with other AWS accounts.	March 31, 2021

Added support for sharing AWS Transit Gateway resources	You can now use AWS RAM to share transit gateway multicast domains with other AWS accounts.	December 10, 2020
Added support for sharing AWS Network Firewall resources	You can now use AWS RAM to share AWS Network Firewall firewall policies and rule groups with other AWS accounts.	November 17, 2020
Added support for sharing for Outposts and local gateway route tables	You can now use AWS RAM to share Outposts and local gateway route tables with other AWS accounts.	October 15, 2020
Added support for sharing Route 53 query logs	You can now use AWS RAM to share Route 53 query logs with other AWS accounts.	September 7, 2020
Added support for sharing AWS Private Certificate Authority resources	You can now use AWS RAM to share AWS Private CA private certificate authorities (CAs) with other AWS accounts.	August 17, 2020
Added support for sharing AWS Glue data catalogs, databases, and tables	You can now use AWS RAM to share AWS Glue data catalogs, databases, and tables with other AWS accounts.	July 7, 2020
Added support for sharing Amazon VPC prefix lists	You can now use AWS RAM to share prefix lists.	June 29, 2020
Added support for sharing AWS Outposts customer-owned IPv4 addresses	You can now use AWS RAM to share AWS Outposts customer-owned IPv4 addresses with other AWS accounts.	April 22, 2020

Added support for sharing AWS App Mesh meshes	You can now use AWS RAM to share meshes with other AWS accounts.	January 17, 2020
Added support for sharing AWS CodeBuild projects and report groups	You can now use AWS RAM to share AWS CodeBuild projects and report groups with other AWS accounts.	December 13, 2019
Added support for sharing additional resources	You can now use AWS RAM to share Amazon EC2 Dedicated Hosts, AWS Resource Groups resource groups, and Amazon EC2 Image Builder component s, images, and image recipes with other AWS accounts.	December 2, 2019
Added support for sharing On-Demand Capacity Reservations	You can now use AWS RAM to share On-Demand Capacity Reservations with other AWS accounts.	July 29, 2019
Added support for sharing Aurora DB clusters	You can now use AWS RAM to share Aurora DB clusters with other AWS accounts.	July 2, 2019
Added support for sharing Traffic Mirroring targets	You can now use AWS RAM to share Traffic Mirroring targets with other AWS accounts.	June 25, 2019
Added support for sharing license configurations	You can now use AWS RAM to share AWS License Manager license configurations with other AWS accounts.	December 5, 2018
Added support for sharing subnets	You can now use AWS RAM to share Amazon VPC subnets with other AWS accounts.	November 27, 2018

Added support for sharing transit gateways	You can now use AWS RAM to share Amazon VPC transit gateways with other AWS accounts.	November 26, 2018
Added support for sharing Resolver rules	You can now use AWS RAM to share Route 53 Resolver rules with other AWS accounts.	November 20, 2018