aws

Administrator Guide

AWS Service Catalog AppRegistry

AWS Service Catalog AppRegistry: Administrator Guide

Table of Contents

What is AppRegistry?

With AppRegistry, you create applications to store associated resources. You can create attribute groups that describe the context of your applications based on metadata you provide. You can create tags that assign metadata to applications and attribute groups and associate resources with applications. You can also share applications and attribute groups to accounts, organizations, and organizational units. The following video (06:57) shows how you can create application in AppRegistry.

Greater application visibility and governance with :AppRegistry.

> **Note**
>
> The console shown in the video might differ from the console available today.

AWS service integrations

The following AWS services integrate with AppRegistry:

- **AWS Billing**

  With Billing, you can use cost allocation tags to analyze cost trends for your applications and resources.
- **Amazon CloudWatch**

  With CloudWatch, you can use a feature called Application Insights to monitor application resources and detect issues with applications.
- **AWS Management Console**

  With the AWS Management Console, you can access and view your applications and associated resources.
- **AWS Resource Access Manager (AWS RAM)**

  With AWS RAM, you can share applications and attribute groups through AWS Organizations.
- **AWS Resilience Hub**

  With Resilience Hub, you can collect application resources to create a resiliency policy.

- **AWS Resource Groups**

  With Resource Groups, you can use tags to share applications and attribute groups, as well as organize application resources.

- **AWS Service Catalog**

  With Service Catalog, you can create provisioned products, which are AWS CloudFormation stacks, that you can associate with applications.

- **AWS Service Management Connector**

  With Service Management Connector, you can view applications in a ServiceNow Configuration Management Database.

- **AWS Systems Manager**

  With Systems Manager, you can use a feature called Application Manager to detect and fix issues with resources in the context of their applications and clusters.

- **AWS Well-Architected Tool**

  With AWS WA Tool, you can extend AWS Well-Architected functionality, best practices, measurements, and learnings into your existing architecture governance processes, applications, and workflows.

Pricing

For information about pricing, see AWS Service Catalog pricing.

Key concepts of AWS Service Catalog AppRegistry

This topic describes the key components of AppRegistry.

Topics

- Application sharing

Applications

An application is a group of resources and metadata. When you create an application, you provide the application with a name and description. After you create an application, you can add a tag-based resource group or an AWS CloudFormation stack resource group to it. You can also associate attribute groups and tags with the application.

> **Note**
>
> When you create an application, AppRegistry vends a user tag called the `awsApplication` tag. This tag identifies resources associated with an application. For more information, see The `awsApplication` tag.

You can create applications in myApplications in the AWS Management Console, in the AppRegistry console, and with the AWS CLI using the AppRegistry API. You can also create applications with the AWS CDK or an AWS SDK of your choice.

You can view and manage applications in myApplications in the AWS Management Console, in the AppRegistry console, and with the AWS CLI, as well as in a set of AWS services.

The `awsApplication` tag

The `awsApplication` tag is a tag that AppRegistry vends when you create an AppRegistry application. This tag marks resources as belonging to an application. This tag consists of a key-value pair, where the key is `awsApplication` and value is the Amazon Resource Name (ARN) of an application.

To help you manage and monitor your applications, AWS highly recommends you ensure all of your application resources are tagged with the `awsApplication` tag. Review Tutorial: Existing AppRegistry application resources and the `awsApplication` tag for instructions.

The `awsApplication` tag provides additional features and capabilities for your applications and resources, including:

- Access to application monitoring and management in myApplications dashboard in the AWS Management Console

- Viewing details about your application's costs, security findings, alarms, metrics and usage
- Filtering by application in integrated AWS services, such as AWS Cost Explorer and AWS Security Hub

For AppRegistry applications created before November 8th, 2023, AppRegistry creates the `awsApplication` tag after you perform your first resource association. You can then apply the `awsApplication` tag to any other resources you want to add to the application. For AppRegistry applications created after November 8th, 2023, AppRegistry creates the `awsApplication` tag when you create the application.

As of September 18th, 2024 for **existing** AppRegistry users, the following behaviors apply:

- For existing AppRegistry applications which include resources without the `awsApplication` tag applied, AWS does not retroactively apply the tag.
- For existing AppRegistry applications, if you use the `AssociateResource` AppRegistry API, AWS does not automatically apply the tag unless you define the `options` parameter value as `APPLY_APPLICATION_TAG`.
- For existing AppRegistry applications, if you use the AppRegistry console to add a new resource to the application, AWS automatically applies the `awsApplication` tag to that new resource.
- For new applications created in the AppRegistry console, AWS automatically applies the `awsApplication` tag to all resources added to the application.
- For new applications created with myApplications in the AWS Management Console, AWS automatically applies the `awsApplication` tag to all resources added to the application.

Tutorial: Existing AppRegistry application resources and the `awsApplication` tag

If you have existing AppRegistry application, AWS recommends that you retroactively apply the `awsApplication` tag to all of the resources in the application, and also ensure any future resources added to the application have the `awsApplication` tag applied. This tutorial provides instructions for both recommendations.

> **Note**
>
> Managing an application's resources by adding or removing the `awsApplication` tag requires specific permissions. Review the minimum permissions for the AppRegistry APIs in

the *AWS Service Catalog Developer Guide* and the Resource Groups APIs in the *AWS Resource Groups API Reference*.

Apply the `awsApplication` tag to the resources in an existing application

In this situation, you have an existing AppRegistry application which includes resources that are *not* tagged with the `awsApplication` tag. The following procedure provides instructions to apply the `awsApplication` tag to those resources.

To tag an existing application's resources with the `awsApplication` tag:

1. Identify the **application's tag value** (`awsApplication` tag value), which is expressed as an Amazon Resource Name (ARN).
   - Call the AppRegistry `GetApplication` API and find the value in the `applicationTag` response parameter.
   - Alternatively, you can navigate to the myApplications dashboard for the application and copy the `awsApplication` tag value from the Application summary widget.
2. After identifying the Application tag value, call the AppRegistry `ListAssociatedResources` API to view a list of resources that are already in the application.
3. Call the AppRegistry `AssociateResource` API with the `options` parameter value as `APPLY_APPLICATION_TAG`.

   Example CLI command:

   ```
   aws servicecatalog-appregistry associate-resource --application application_ARN
                          --resource-type type --resource name --option
    "APPLY_APPLICATION_TAG"
   ```

Add more resources to an existing application by applying the `awsApplication` tag

In this situation, you have an existing AppRegistry application to which you want to add new resources, and you want those resources to have the `awsApplication` tag applied.

Use myApplications in the AWS Management Console to add resources to an application. Resources added to the application using myApplications are automatically tagged with the `awsApplication` tag.

You can also use the AWS Resource Groups `GroupResources API` to add resources to a specified group. The group is an application group, where the ARN is the `awsApplication` tag value.

To manually add resources to an application:

Call the Resource Groups `GroupResources` API and define the following parameters:

- `Group` — The name or the Amazon resource name (ARN) of the application group to add resources to.
- `ResourceARNs` — The list of Amazon resource names (ARNs) of the resources to be added to the group.

Example CLI command:

```
aws resourcegroups group-resources –-group "ApplicationGroup-ARN" –-resourceArns
 "Resource-ARNs"
```

Resource tag-sync tasks

An automatic tag-synchronization of application resources (a *tag-sync task*) is an application resource management strategy that works by automatically adding and removing The `awsApplication tag` from resources to manage their inclusion in an application. When you create a tag-sync task in the application, you specify a tag key-value pair to sync to the application, such as `Project:Blue`. The task then adds any resources tagged with `Project:Blue` to the application by adding the `awsApplication` tag to those resources.

When you perform the following actions, AWS adds all resources tagged with the `Project:Blue` tag to the application by applying the `awsApplication` tag to those resources:

- Create an application using the existing tag key-value pair `Project:Blue`. For more information about bulk-onboarding application resources by specifying an existing tag key-value pair at application creation, review Creating your first application in myApplications in the AWS Management Console *Getting started guide*.
- Create a tag-sync task in an existing application using the `Project:Blue` tag.

After you configure the tag-sync task, it continuously manages the application's resources, adding or removing resources as they are tagged or untagged with the specified key-value pair.

When the task is active, if you tag a resource with the `Project:Blue` tag, the tag-sync adds that resource to the application by applying the `awsApplication` tag to it.

When you remove the `Project:Blue` tag from a resource, the tag-sync removes the resource from the application by removing the `awsApplication` tag.

Topics

Tag-sync task required permissions

Creating and managing application tag-sync tasks requires you to specify or create an IAM role that allows the tag-sync task to manage the application resources.

When configuring a tag-sync task, AWS recommends creating and using a new role to ensure the role includes the correct trust permissions. With this option, AWS creates a role named *tag-sync-role-**`region-uniqueID`***. This role is comprised of the following permissions:

- The `ResourceGroupsTaggingAPITagUntagSupportedResources` AWS managed policy — Allows the tag-sync task to tag and untag resources. You can modify the role's resource permissions based on your application needs by adding or removing a specific resource's `TagResource` and `UntagResource` permissions. For example, add `amplify:TagResource` and `amplify:UntagResource` to allow the tag-sync task to manage tags for AWS Amplify resources.
- A role trust policy— Allows AWS Resource Groups to assume the role and perform related tasks on your behalf.
- An inline policy— Allows AWS Resource Groups to group and ungroup resources.

> **Important**
>
> To avoid disrupting the tag-sync task, do not delete this role or edit its trust or inline policies.

If you choose to use an existing IAM role, ensure it includes the following permissions:

- Permissions to tag and untag application resources.

  Option 1: Use AWS managed policies

  Use both the AWS Resource Groups `ResourceGroupsTaggingAPITagUntagSupportedResources` and `ResourceGroupsandTagEditorFullAccess` AWS managed policies to grant the permissions required to tag and untag all of the resource types supported by Resource Groups Tagging API, with some exceptions. The `ResourceGroupsandTagEditorFullAccess` policy also grants the permissions required to retrieve all tagged, or previously tagged, resources through the Resource Groups Tagging API.

  Option 2: Manually add permissions to an existing policy

  If you choose not to use AWS managed policies, you must manually configure your policy to include all of the permissions required to tag and untag your specific resources. For example, add the `sqs:TagQueue` permission if you have an Amazon SQS queue resource in your application. In addition to the resource-specific permissions, your policy must include the following Resource Groups Tagging API permissions:

  - `resource-groups:GroupResources`
  - `resource-groups:UngroupResources`
  - `tag:GetResources`
  - `tag:TagResources`
  - `tag:UntagResources`

- A trust policy attached that allows AWS Resource Groups to assume the role and perform these tasks on your behalf. The following is an example trust policy.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Statement1",
      "Effect": "Allow",
      "Principal": {
        "Service": "resource-groups.amazonaws.com"
      },
      "Action": "sts:AssumeRole"
    }
```

```
  ]
}
```

Create a tag-sync task

This section provides instructions to create a tag-sync task for resources in an existing application using either myApplications in the AWS Management Console or with the AWS API.

myApplications in AWS Management Console

For instructions to bulk-onboard application resources by specifying an existing tag key-value pair at application creation, review Creating your first application in myApplications in the AWS Management Console *Getting started guide*.

AWS Resource Groups API

To create a tag-sync task

1. Create a new tag-sync task by calling the Resource Groups `StartTagSyncTask` API and specifying the following parameters:

   - **Group** — The Amazon resource name (ARN) or name of the application group for which you want to create a tag-sync task.
   - **TagKey** — The tag key.
   - **TagValue** — The tag value.
   - **RoleARN** — The ARN of the role that AWS Resource Groups assumes when performing the tag-sync task to apply the `awsApplication` tag to resources. This role must have the tagging permissions to all resources you want to include in the application. For more information, review Tag-sync task required permissions.

   ```
   aws resourcegroups start-tag-sync-task --group appgroup-ARN-or-name --
   tagkey tag-key --tagvalue tag-value --roleArn role-ARN
   ```

   Example output:

   The output includes a `TaskArn` that you can later use to query the status of the tag-sync task.

```
{
    "GroupArn" : "string",
    "GroupName" : "string",
    "TaskArn": "string",
    "TagKey" : "string",
    "TagValue" : "string",
    "RoleArn" : "string",
                        }
```

> **Note**
>
> For some accounts, an attempted `StartTagSyncTask` call may result in a `GroupNotFound` error. To resolve this error, call the AppRegistry `UpdateApplication` API. This call enables the application to perform tag-sync tasks.

2. Verify the status of the tag-sync task by calling the `GetTagSyncTask` API. Enter the `TaskArn` from the output of the previous `StartTagSyncTask` call.

```
aws resourcegroups get-tag-sync-task --taskArn task-ARN
```

Example output:

```
{
    "GroupArn" : "string",
    "GroupName" : "string",
    "TaskArn": "string",
    "TagKey" : "string",
    "TagValue" : "string",
    "RoleArn" : "string",
    "Status" : "string",
    "ErrorMessage" : "string",
    "ManagedGroupArn": "string",
    "CreatedAt": "timestamp"
}
```

(optional) To list all tag-sync tasks

To list all active tag-sync tasks, call the `ListTagSyncTasks API`. You can optionally include the `Filters` parameter with a `GroupArn` or `GroupName` to narrow your results to tasks in a specific application.

```
aws resourcegroups list-tag-sync-task --filters group-name
```

Example output:

```
{
    "TagSyncTasks": [
        {
            "GroupArn" : "string",
            "GroupName" : "string",
            "TaskArn": "string",
            "TagKey" : "string",
            "TagValue" : "string",
            "RoleArn" : "string",
            "Status" : "string",
            "ErrorMessage" : "string",
            "CreatedAt": "timestamp"
        }
    ],
    "NextToken": "string"
 }
```

To cancel a tag-sync task

To cancel a tag-sync task, call the `CancelTagSyncTask API` and enter the `TaskArn` of the tag-sync task you want to delete.

```
aws resourcegroups cancel-tag-sync-task --task-arn task-ARN
```

Attribute groups

Attribute groups are JSON objects that store application metadata. You associate attribute groups with applications to understand applications in the context of their associated metadata.

Example: Attribute group with metadata

The following snippet shows an attribute group with metadata that includes a team name, department number, department name, and email address.

```
{
 "Team" : "WebTeam",
 "Department": "10006",
 "ParentDept": "Research",
 "ContactAlias": "research@team.com"
}
```

You can use the AppRegistry `AssociateAttributeGroup` API to apply metadata to an application. You can use the AppRegistry `DisassociateAttributeGroup` API to remove metadata from an application.

You can associate attribute groups with applications in the console and with the AWS CLI using the AppRegistry API, with AWS CloudFormation stack resources, or with CDK constructs.

You can update an attribute group definition at any time with the AppRegistry `UpdateAttributeGroup` API. When you update an attribute group definition, the update applies to every application the attribute group is associated with.

You can share attribute groups to accounts, organizations, and organizational units with the following permissions:

- **Allow associations**

  Allows IAM principals in shared accounts to associate and disassociate attribute groups.
- **Read only associations**

  Allows IAM principals in shared accounts to view attribute groups

You can automate stack updates and metadata changes in a continuous delivery and continuous integration pipeline. This allows stakeholders to query and receive information about attribute groups

You can view and manage attribute groups in the console and with the AWS CLI.

Tags

Tags are key-value pairs that act as metadata. You create tags using key-value pairs. You can add tags to applications and attribute groups, so you can group them by environment, owner, purpose, or other criteria.

> **Note**
>
> This tag is not the same as the [the awsApplication tag]. The `awsApplication` tag tag is an AWS user tag that AppRegistry vends when you create an application. You can add the `awsApplication` tag tag to resources, so you can identify which resources are associated with an application.

Example: AWS CLI output with `tags` parameter

The following is an example of the output for an application created in the AWS CLI, which includes the `tags` and `applicationTag` parameters.

```
{
                    "application": {
                    "arn": "string",
                    "creationTime": "string",
                    "description": "string",
                    "id": "string",
                    "lastUpdatedTime": "string",
                    "name": "string",
                    "applicationTag": {"awsApplication":"arn:aws:resource-groups:us-
east-1:234567891011:group/myExampleApp/012345example6789101112131"},
                    "tags": {
                        "myKey":"myValue"
                    }
                }
            }
```

You can [view and manage tags] in the console and with the AWS CLI.

Application sharing

Deploying applications across multiple AWS accounts is common and considered a best practice that can help isolate and manage business applications and data. With AppRegistry and AWS

Resource Access Manager (AWS RAM), you can share applications and attribute groups with one or more accounts, organizations, and organizaional units. You can share applications and attributes in the console and AWS CLI using the AWS Resource Access Manager API and infrastructure as code. Resources can be associated with shared applications. For more information, see Sharing resources with accounts in your organization.

AWS Service Catalog AppRegistry default service quotas

This topic describes the default service quotas for AWS Service Catalog AppRegistry. For more information, seeWhat is Service Quotas? in the *Service Quotas*.

AppRegistry service quotas

- Applications per AWS Region: 2,000
- Attribute groups per AWS Region: 2,000
- Resources per application: 1,000
- Attribute groups per application: 1,000
- Applications per attribute group: 1,000
- Attribute group size: 8,000 characters

> **Note**
>
> You can request a quota increase. For more information, see Requesting a quota increase in the *Service Quotas*.

Getting started with AppRegistry

When you create a repository for all of your AWS applications and associated resources, you increase the visibility and governance of these applications, which helps you define and manage application metadata and better understand the AWS applications and resources in your organization.

Key tasks in AppRegistry

The following topics help you get started with AppRegistry.

- Create an application to group resources and metadata. For more information, see Creating applications.
- Add a user tag called `awsApplication` tag to resources, so you can identify which resources are associated with an application. For more information, see the `awsApplication` tag.
- Associate resources with your application. For more information, see Associating and disassociating application resources.
- Share your application to other accounts in your organization. For more information, see Sharing resources with accounts in your organization.
- Create and associate an attribute group with your application. For more information, see Managing attribute groups.
- Create tags to organize your application resources. For more information, see Managing tags.

> **Note**
>
> This section includes a tutorial that describes how to create applications and attribute groups in the console and programatically with the AWS CLI using the AppRegistry API.

Topic

- Tutorial: Create your first application and attribute group in AppRegistry

Tutorial: Create your first application and attribute group in AppRegistry

The procedures in this topic describe how to get started with AppRegistry.

In the first section, you learn how to create an application and attribute group, as well as how to add an attribute group and application resource to an application.

In the second section, you learn how to discover information about applications and attribute groups.

> **Note**
>
> The procedures in topic show you how to complete AppRegistry tasks in the AWS CLI. You can use the information from the following procedures to complete the AppRegistry tasks in the AWS Management Console by following the appropriate links.

Topics

Create your first application and attribute group

The first two procedures in this section describe how to create an application and attribute group. The following procedures in this section describe how to add an attribute group and application resource to an application.

Create an application

The procedure in this section shows how to create an application in the AWS CLI. For information about creating an application in the console, see Creating applications.

The example describes how to format a command from the perspective of an application builder.

- To create an application in the AWS CLI, run the following command:

```
aws servicecatalog-appregistry create-application --name "CC_Payments_App" --
description "Real-time payments service for processing customer orders."
```

Example: Output

The following shows the output you might encounter.

```
{
        "application": {
            "id": "your-resource-id",
            "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/
applications/your-resource-id",
            "name": "CC_Payments_App",
            "description": "Real-time payments service for processing customer
 orders.",
            "creationTime": "2023-12-12T20:17:20.017000+00:00",
            "lastUpdateTime": "2023-12-12T20:17:20.017000+00:00",
            "applicationTag": {
                "awsApplication": "arn:aws:resource-groups:your-Region:your-account-
id:group/CC_Payments_App/your-resource-id"
            }
        }
}
```

> **Note**
>
> When you create an application, AppRegistry vends a user tag called the `awsApplication` tag on your behalf. You can add this tag to resources, so you can identify which resources are associated with an application. For more information, see the `awsApplication` tag.

Create an attribute group

The procedures in this section show how to create two attribute groups. For information about creating an attribute group in the console, see Creating attribute groups.

The examples describe how to format commands from the perspective of an administrator and application builder.

> **Note**
>
> The following command and output is for an attribute group an administrator might create to share in multiple accounts.

- To create an attribute group in the AWS CLI from the administrator's perspective, run the following command:

```
aws servicecatalog-appregistry create-attribute-group --name
 "Corp_Application_Classification_High" --description "Applications classified as
 high." --attributes
 '{"ApplicationResilience":"high","DataSecurity":"high","DataSensitivity":"high"}'
```

Example: Output

The following shows the output you might encounter.

```
{
  "attributeGroup": {
    "id": "your-resource-id",
    "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/attribute-groups/your-
resource-id",
    "name": "Corp_Application_Classification_High",
    "description": "Applications classified as high.",
    "creationTime": "2023-12-12T20:14:27.413000+00:00",
    "lastUpdateTime": "2023-12-12T20:14:27.413000+00:00",
    "tags": {}
  }
}
```

> **Note**
>
> The following command and output is for an attribute group an application builder might create to track information about an application.

- To create an attribute group in the AWS CLI from the application builder's perspective, run the following command:

```
aws servicecatalog-appregistry create-attribute-group --name "Commerce_Payments"
 --description "24X7 real-time payments processing." --attributes
 '{"Team":"payments","app-
type":"processing","SLA":"0.1h","Runtime":"Java-12","Support":{"Phone":"XXX-XXX-
XXXX","Email":"support@app.com"},"Compliance":["SOC-1","PCI"]}}'
```

Example: Output

The following shows the output you might encounter.

```
{
    "attributeGroup": {
        "id": "your-resource-id",
        "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/attribute-
groups/your-resource-id",
        "name": "Commerce_Payments",
        "description": "24X7 real-time payments processing.",
        "creationTime": "2023-12-12T20:15:49.658000+00:00",
        "lastUpdateTime": "2023-12-12T20:15:49.658000+00:00",
        "tags": {}
    }
}
```

Add an attribute group to an application

The procedure in this section shows how to add an attribute group with an application. For information about associating an attribute group with an application in the console, see Associating and disassociating attribute groups.

The example includes information from the application and attribute group you created from the application builder's perspective.

- To add an attribute group to an application, run the following command:

```
aws servicecatalog-appregistry associate-attribute-group --application
 "CC_Payments_App" --attribute-group "Commerce_Payments"
```

Example: Output

The following shows the output you might encounter.

```
{
"applicationArn": "arn:aws:servicecatalog:your-Region:your-account-id:/
applications/your-resource-id",
"attributeGroupArn": "arn:aws:servicecatalog:your-Region:your-account-id:/attribute-
groups/your-resource-id"
}
```

Add AWS CloudFormation stack resource to an application

The procedure in this section shows how to add an AWS CloudFormation stack resource to an application in the AWS CLI. For information about associating an AWS CloudFormation stack resource with an application in the console, see Associating and disassociating application resources.

The example includes information from the application you created from the application builder's perspective and a AWS CloudFormation stack resource you must create separately.

> **Important**
>
> To complete this tutorial, you must create a AWS CloudFormation stack resource named **cc_payment_app_cfn_stackCODE**. For information about how to create a AWS CloudFormation stack resource, see Creating a stack in the *AWS CloudFormation User Guide*

- To add an AWS CloudFormation stack resource to an application, run the following command:

```
aws servicecatalog-appregistry associate-resource --application CC_Payments_App
 --resource-type CFN_STACK --resource cc_payment_app_cfn_stackCODE --apply-tag
 cc_payment_application
```

Example: Output

The following shows the output you might encounter.

```
{
  "applicationArn": "arn:aws:servicecatalog:your-Region: XXXXXXXXXX:/applications/your-
resource-id",
  "resourceArn": "arn:aws:cloudformation:you-Region: XXXXXXXXXX:stack/
cc_payment_app_cfn_stack/your-resource-id"
```

```
}
```

Discover information about your applications

The procedures in this section describe how you can use AppRegistry to find information about applications and attribute groups.

View your applications

The following shows how to view a list of your applications using the AWS CLI. For information about viewing your applications in the console, see Managing applications.

If you created an application other than the application described in the previous section, the output in this procedure will look different.

> **Note**
>
> The following command will only show the applications in your current AWS Region.

- To view a your applications in your application registry, run the following command:

```
aws servicecatalog-appregistry list-applications
```

Example: Output

The following shows the output you might encounter.

```
{
    "applications": [
        {
            "id": "your-resource-id",
            "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/
applications/your-resource-id",
            "name": "CC_Payments_App",
            "description": "Real-time payments service for processing customer
 orders.",
            "creationTime": "2023-12-12T20:17:20.017000+00:00",
            "lastUpdateTime": "2023-12-12T20:17:20.017000+00:00"
        }
```

```
    ]
}
```

View your attribute groups

The following shows how to view a list of your attribute groups using the AWS CLI. For information about viewing your attribute groups in the console, see Managing attribute groups.

If you created an attribute group other than the attribute groups described in the previous section, the output in this procedure will look different.

> **Note**
>
> The following command will only show the attribute groups in your current AWS Region.

- To view your attribute groups, run the following command:

```
aws servicecatalog-appregistry list-attribute-groups
```

Example: Output

The following shows the output you might encounter.

```
{
    "attributeGroups": [
        {
            "id": "your-resource-id",
            "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/attribute-
groups/your-resource-id",
            "name": "Corp_Application_Classification_High",
            "description": "Applications classified as high.",
            "creationTime": "2023-12-12T20:14:27.413000+00:00",
            "lastUpdateTime": "2023-12-12T20:14:27.413000+00:00"
        },
        {
            "id": "your-resource-id",
            "arn": "arn:aws:servicecatalog:your-Region:your-account-id:/attribute-
groups/your-resource-id",
            "name": "Commerce_Payments",
```

```
            "description": "24X7 real-time payments processing.",
            "creationTime": "2023-12-12T20:15:49.658000+00:00",
            "lastUpdateTime": "2023-12-12T20:15:49.658000+00:00"
        }
    ]
}
```

View an attribute group that's been added to an application

The following shows how to view an attribute group that's been added to an application in the AWS CLI. For information about how to do this in the console, see Using application details.

The output in this procedure shows the attribute group you created from the application builder's perspective in the previous section.

- To view an attribute group that's been added to an application, run the following command:

```
aws servicecatalog-appregistry list-associated-attribute-groups --application
 "CC_Payments_App"
```

Example: Output

The following shows the output you might encounter.

```
{
    "attributeGroups": [
        "your-resource-id"
    ]
}
```

Using AppRegistry

This topic describes how to create and manage applications in the AppRegistry.

Topics

Managing applications

This section describes how to create and manage applications in AppRegistry. After you define an application by specifying its name, description, and share configuration, you can perform the following actions:

- Associate resources with the application. For more information, see Managing application definitions.
- Associate attribute groups with the application. For more information, see Managing attribute groups.
- Associate tags with the application. For more information, see Managing tags.
- Share the application with accounts, organizations, and organizational units. For more information, see Sharing resources with accounts in your organization.

> **Note**
>
> When you create an application, AppRegistry vends an AWS user tag called the `awsApplication` tag. The `awsApplication` tag identifies resources associated with an application. For more information, see the `awsApplication` tag.

Topics

- Editing applications
- Deleting applications
- Managing application resources

Creating applications

You create applications to group resources and metadata. After you enter a name and description for your application, you can associate resources, attribute groups, and tags with it. You can also share your application with other accounts in your organization.

When you create an application, AppRegistry vends a user tag called the `awsApplication` tag on your behalf. You can add this tag to resources to help identify which resources are associated with an application.

myApplications in AWS Management Console

Use myApplications in the AWS Management Console to create a new application and organize its resources.

AWS recommends creating all of your new applications using myApplications in the AWS Management Console. This method ensures all of the resources added to the application are tagged with the `awsApplication` tag and provides you with the additional features and benefits of myApplications.

AppRegistry console

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose **Create application**.
4. Under **Application name and description**, enter a name for your application. You can optionally enter a description for your application.
5. (Optional) Under **Application share configuration**, choose **Turn on cross-account sharing** to share the application's visibility with accounts, organizational units, and organizations. For more information, see Sharing resources with accounts in your organization.
6. (Optional) Under **Resource collections**, select resources to associate to your application. For more information, see Managing application definitions.

7. (Optional) Under **Attribute groups**, select one or more attribute groups to associate to your application. For more information, see Managing attribute groups.
8. (Optional) Under **Application tags**, create tags using key-value pairs to assign metadata to your application. For more information, see Managing tags.
9. Confirm your application configuration, and then choose **Create application**.

Using Application details

The **Application details** screen shows the following information:

- The application's name and description
- When the application was created and who created the application
- The application's ID, ARN, and resource group ARN
- The application's share configuration
- The resources and attribute groups associated with the application, as well as the application's resource shares

You can also view the tags you create to organize application resources and the `awsApplication` tag, which is an AWS user tag that you can use to add resources to an application.

> **Note**
>
> For AppRegistry applications created before November 8th, 2023, AppRegistry creates the `awsApplication` tag after you perform your first resource association. This tag's value is a unique identifier for the application. You can then apply the `awsApplication` tag to any other resources you want to add to the application. For AppRegistry applications created after November 8th, 2023, AppRegistry creates the `awsApplication` tag when you create the application.

You can perform the following actions from the **Application details** screen:

- View applications in AWS Systems Manager Application Manager. For more information, see Viewing applications in AWS Systems Manager Application Manager.
- Delete and edit applications. For more information, see Deleting applications and Editing applications.

- View and manage resources associated with applications. For more information, see Associating and disassociating application resources.
- View and manage attribute groups associated with applications. For more information, see Associating and disassociating attribute groups.
- View and manage resource shares associated with applications. For more information, see Sharing application resources with accounts in your organization.
- View and manage tags you create to organize application resources and identify resources associated with an application. For more information, see Managing tags and The `awsApplication` tag.

Topics

Viewing Application details

This topic describes how to view the **Application details** screen.

To view Application details

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to view. Or select the application that you want to view, and then choose **View**. You're directed to the **Application details** screen.

Viewing applications in AWS Systems Manager Application Manager

You can view applications in AWS Systems Manager Application Manager to gain operational information and detect issues with AWS resources. For more information, see AWS Systems Manager Application Manager.

To view an application in AWS Systems Manager Application Manager

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/

2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to view. Or select the application that you want to view, and then choose **View**. You're directed to the **Application details** screen.
4. Choose **View in Application Manager**. You're directed to the AWS Systems Manager Application Manager console.

Editing applications

> **Note**
>
> You can also use myApplications in the AWS Management Console to manage, edit, and delete your applications. Review Managing applications in the AWS Management Console *Getting started guide* for instructions.

With AWS Service Catalog AppRegistry, you can update application descriptions from the **Applications** screen or **Application details** screen.

To edit applications from Application screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, select the application that you want to edit, and then choose **Edit**.
4. On **Edit application description**, update the description, and then choose **Save changes**.

To edit applications from the Applications details screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to edit. Or select the application that you want to edit, and then choose **View**. You're directed to the **Application details** screen.

4. On **Application details**, choose **Edit**.
5. On **Edit application description**, update the description, and then choose **Save changes**.

Deleting applications

> **Note**
>
> You can also use myApplications in the AWS Management Console to manage, edit, and delete your applications. Review Managing applications in the AWS Management Console *Getting started guide* for instructions.

With AppRegistry, you can delete applications from the **Applications** screen or **Application details** screen. Before deleting an application, you must complete the following prerequisites:

Prerequisites

- Remove all resources associated with your application
- Remove the application tag from all associated resources

Delete an application from the Applications screen

The following procedure describes how to delete an application from the **Applications** screen.

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. Select the application that you want to delete, and then choose **Delete**.
4. Confirm your deletion, and then choose **Delete application**.

Delete an application from the Application details screen

The following procedure describes how to delete an application from the **Application details** screen.

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/

2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, Choose the name of the application that you want to view. Or select the application that you want to view, and then choose **View**. You're directed to the **Application details** screen.
4. On **Application details**, choose **Delete**.
5. Confirm your deletion, and choose **Delete application**.

Managing application resources

> **Note**
>
> You can also use myApplications in the AWS Management Console to add and remove resources from your applications. Review Managing resources in the AWS Management Console *Getting started guide* for instructions.

An application resource is an object within an AWS service that you can tag with the `awsApplication` tag. AWS customers and services use the `awsApplication` tag to add and remove resources from applications and identify which resources are associated with an application.

You add resources to your application after you define your application. You can add and remove application resources with any of the existing methods for tagging resources, infrastructure as code, and the AppRegistry API.

To add and remove application resources with the AppRegistry API, use the console procedures or the AppRegistry `AssociateResource` and `DisassociateResource` APIs. You can can add the `awsApplication` tag to a resource using the AppRegistry `AssociateResource` API with the `APPLY_APPLICATION_TAG` option.

> **Note**
>
> Adding and removing resources requires certain permissions. For more information, see AssociateResource and DisassociateResource in the *AWS Service Catalog AppRegistry Developer Guide*.

AppRegistry integrates with AWS Resource Groups. When you create an application, AWS Resource Groups creates an application resource group and a resource group for every AWS CloudFormation stack or tag-based resource you associate with your application. You can list the resources in your application by calling the Resource Groups `ListGroupResources` API on the application resource group. Any resource tagged with the `awsApplication` tag for this application will be a member of this group.

For information about resource types and related functionalities you can use with AppRegistry applications, see Supported resource types for AppRegistry applications.

This section decribes how to manage application definitions as you create and associate deployed resources to applications in your local account and AWS Region.

Topics

Associating and disassociating application resources

An application resource is an object within an AWS service that you can tag with the `awsApplication` tag, which is an AWS user tag that AppRegistry vends on your behalf. The following procedures describe how to associate and disassociated application resources.

> **Note**
>
> For AppRegistry applications created before November 8th, 2023, AppRegistry creates the `awsApplication` tag after you perform your first resource association. This tag's value is a unique identifier for the application. You can then apply the `awsApplication` tag to any other resources you want to add to the application. For AppRegistry applications created after November 8th, 2023, AppRegistry creates the `awsApplication` tag when you create the application.

Topics

- Disassociate application resources from an application

Associate application resources in a new application

The following procedure describes how to associate application resources in a new application.

To associate application resources in a new application.

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose **Create application**.
4. Under **Application name and description**, enter a name and optional description for your application.
5. Under **Resource collections**, choose one or more provisioned products or CloudFormation stacks to associate to your application.
6. Choose **Create application**.

Associate application resources in an existing application

The following procedure describes how to associate application resources in an existing application.

To associate application resources in an existing application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the left navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to associate resources to. Or select the name of application that you want to associate resources to, and choose **View**. You're directed to the **Application details** screen.
4. Choose **Resource collections**, and then choose **Associate resource collection**.
5. Under **Resource collections**, choose one or more provisioned products or CloudFormation stacks to associate to your application.
6. Choose **Save changes**.

> **Note**
>
> If you share an application with this account, and the application has read-only permissions, associate and disassociate actions are disabled for resource collections.

Disassociate application resources from an application

The following procedure describes how to disassociate application resources from an existing application.

To disassociate application resources from an existing application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to disassociate resources from. Or select the name of the application that you want to disassociate resources from, and choose **View**. You're directed to the **Application details** screen.
4. Choose **Resource collections**, select the resource that you want to disassociate from the application, and then choose **Disassociate**.
5. Confirm your disassociation, and then choose **Ok**.

> **Note**
>
> If you share an application with this account, and the application has read-only permissions, associate and disassociate actions are disabled for resource collections.

Controlling the resources associated to applications

This topic includes policy templates that you can use to control how tag key-value pairs are associated to applications.

The following policy templates are organized by scenario and include values that can be replaced with your information.

Sample policy: Stack only association

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": "servicecatalog:AssociateResource",
            "Resource": "arn:aws:servicecatalog:*:*:*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceType": "CFN_STACK"
                }
            }
        }
    ]
}
```

Sample policy: Stack association that allows a specific stack name

```
{
    "Version": "2012-10-17",
    "Statement": [
       {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
```

```
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceType": "CFN_STACK",
                    "servicecatalog:ResourceIdentifier": "StackName2"
                }
            }
        }
    ]
}
```

Sample policy: Stack association that allows multiple specific stack names

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceType": "CFN_STACK",
                    "servicecatalog:ResourceIdentifier": ["StackName1", "StackName2"]
                }
            }
        }
    ]
}
```

Sample policy: Tag value association that denies a specific tag query value while allowing other tag queries

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:ResourceType": "TAG_QUERY",
                    "servicecatalog:ResourceIdentifier": ["StackName1", "StackName2"]
                }
            }
        }
    ]
}
```

Sample policy: Allow tag query association only

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
```

```
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceType": "TAG_QUERY"
                }
            }
        }
    ]
}
```

Sample policy: Allow tag query association/deny specific tag query values

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:ResourceType": "CFN_STACK"
                }
            }
        },
```

```
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "servicecatalog:ResourceType": ["TAG_QUERY"],
                    "servicecatalog:ResourceIdentifier": ["StackName1", "StackName2"]
                }
            }
        }
    ]
}
```

Sample policy: Allow specific tag query value and specific stack

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "servicecatalog:*",
                "cloudformation:DescribeStacks",
                "resource-groups:*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceIdentifier": ["StackName1", "StackName2",
 "EmptyStack", "EmptyStack2"]
                }
            }
        },
```

```
        {
            "Effect": "Deny",
            "Action": [
                "servicecatalog:AssociateResource"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "servicecatalog:ResourceType": "CFN_STACK",
                    "servicecatalog:ResourceIdentifier": ["StackName1", "StackName2"]
                }
            }
        }
    ]
}
```

Supported resource types for AppRegistry applications

This topic includes a list of supported resource types by service for AppRegistry applications.

> **Note**
>
> If you don't see a resource type for your application, you can submit feedback to suggest a resource type to be included in a future update.

Resource type	Sevice
aws::cloudfront::distribution	cloudfront
aws::cloudwatch::alarm	cloudwatch
aws::docdb::cluster	docdb
aws::docdb::clustersnapshot	docdb
aws::docdb::dbclusterparametergroup	docdb
aws::docdb::dbinstance	docdb
aws::docdb::dbsubnetgroup	docdb

Resource type	Sevice
aws::docdb::es	docdb
aws::dynamodb::table	dynamodb
aws::ec2::capacityreservation	ec2
aws::ec2::customergateway	ec2
aws::ec2::dhcpoptions	ec2
aws::ec2::eip	ec2
aws::ec2::host	ec2
aws::ec2::image	ec2
aws::ec2::instance	ec2
aws::ec2::internetgateway	ec2
aws::ec2::launchtemplate	ec2
aws::ec2::natgateway	ec2
aws::ec2::networkacl	ec2
aws::ec2::networkinterface	ec2
aws::ec2::reservedinstances	ec2
aws::ec2::routetable	ec2
aws::ec2::securitygroup	ec2
aws::ec2::snapshot	ec2
aws::ec2::spotinstancesrequest	ec2
aws::ec2::subnet	ec2

Resource type	Sevice
aws::ec2::transitgateway	ec2
aws::ec2::transitgatewayroutetable	ec2
aws::ec2::volume	ec2
aws::ec2::vpc	ec2
aws::ec2::vpcpeeringconnection	ec2
aws::ec2::vpnconnection	ec2
aws::ec2::vpngateway	ec2
aws::ecs::cluster	ecs
aws::ecs::containerinstance	ecs
aws::ecs::service	ecs
aws::ecs::task	ecs
aws::ecs::taskdefinition	ecs
aws::elasticache::cachecluster	elasticache
aws::elasticache::snapshot	elasticache
aws::elasticloadbalancing::loadbalancer	elasticloadbalancing
aws::elasticloadbalancingv2::loadbalancer	elasticloadbalancingv2
aws::elasticloadbalancingv2::targetgroup	elasticloadbalancingv2
aws::iam::instanceprofile	iam
aws::iam::oidcprovider	iam
aws::iam::policy	iam

Resource type	Sevice
aws::iam::samlprovider	iam
aws::iam::servercertificate	iam
aws::kinesis::stream	kinesis
aws::lambda::function	lambda
aws::logs::loggroup	logs
aws::neptune::dbcluster	neptune
aws::neptune::dbclusterparametergroup	neptune
aws::neptune::dbclustersnapshot	neptune
aws::neptune::dbparametergroup	neptune
aws::neptune::dbsubnetgroup	neptune
aws::neptune::eventsubscription	neptune
aws::opensearchservice::domain	opensearchservice
aws::rds::clustersnapshot	rds
aws::rds::dbcluster	rds
aws::rds::dbclusterparametergroup	rds
aws::rds::dbinstance	rds
aws::rds::dbparametergroup	rds
aws::rds::dbsecuritygroup	rds
aws::rds::dbsubnetgroup	rds
aws::rds::eventsubscription	rds

Resource type	Sevice
aws::rds::optiongroup	rds
aws::rds::ri	rds
aws::rds::snapshot	rds
aws::redshift::cluster	redshift
aws::redshift::clusterparametergroup	redshift
aws::redshift::clustersubnetgroup	redshift
aws::s3::bucket	s3
aws::sns::topic	sns
aws::sqs::queue	sqs
aws::ssm::document	ssm
aws::ssm::maintenancewindow	ssm
aws::ssm::managedinstance	ssm
aws::ssm::parameter	ssm
aws::ssm::patchbaseline	ssm

Managing attribute groups

This section describes how to create and manage AWS attribute groups in AppRegistry. When you create an attribute group, you can do the following:

- Share the attribute group's visibility with your organizational structure. For more information, see Sharing resources with accounts in your organization.
- Associate applications to the attribute group. For more information, see Managing applications.
- Assign metadata to the attribute group by creating tags using key/value pairs. For more information, see Managing tags.

After you create an attribute group, you can view its details on the **Attribute group details** screen. For more information, see Using attribute group details.

Topics

Creating attribute groups

With AppRegistry, you create attribute groups to store AWS application metadata.

To create an attribute group

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose **Create attribute group**.
4. Under **Create attribute group**, enter a name and description for your attribute group, and provide the JSON schema that captures your metadata taxonomy.
5. (Optional) Under **Attribute group share configuration**, choose **Turn on cross-account sharing** to share the attribute groups's visibility with your organizational structure. For more information, see Sharing resources with accounts in your organization.
6. (Optional) Under **Assign attribute group to an application**, select one or more applications to associate to the attribute group. For more information, see Managing applications.
7. (Optional) Under **Add tags**, create tags using key/value pairs to assign metadata to the attribute group. For more information, see Managing tags.
8. Choose **Create attribute group**.

Using Attribute group details

You can view the following details for an attribute group from the **Attribute group details** screen:

- Name
- Description
- Date created
- Attribute group ID
- Attribute group ARN
- Created by
- Share configuration

For more information, see Viewing Attribute group details.

You can also perform the following actions from this screen:

- Edit attribute groups. For more information, see Editing attribute groups.
- Delete attribute groups. For more information, see Deleting attribute groups.
- View and edit attribute group metadata. For more information, see Viewing attribute group metadata.
- View resource shares associated to attribute groups. You can also create and delete resource shares. For more information, see Sharing resources with accounts in your organization.
- View attribute group tags. You can also add and delete attribute group tags. For more information, see Managing tags.

Topics

Viewing Attribute group details

This topic describes how to view the **Attribute group details** screen.

To view Attribute group details

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then **Attribute groups**. You're directed to the **Attribute groups** screen.

3. On **Attribute groups**, choose the name of the attribute group that you want to view. Or select the attribute group that you want to edit, and then choose **View**. You're directed to the **Attribute group details** screen.

Viewing attribute group metadata

An attribute group is an open JSON object where you define the metadata for a resource. Metadata is data that describes other data. For more information, see Attribute groups on Overview of AWS Service Catalog AppRegistry.

To view attribute group metadata

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen where you can view all of your attribute groups.
3. On **Attribute groups**, choose the name of the attribute groups that you want to view. Or select the attribute group that you want to edit, and then choose **View**. You're directed to the **Attribute group details** screen.
4. On **Attribute group details**, choose **Metadata**.

Example: Attribute group metadata

```
{
 "Team" : "WebTeam",
 "Department": "10006",
 "ParentDept": "Research",
 "ContactAlias": "research@team.com"
}
```

Editing attribute groups

With AWS Service Catalog AppRegistry, you can update an attribute group description and JSON schema from the **Attribute groups** screen or **Attribute group details** screen.

> **Note**
>
> When you update an attribute group definition, the update applies to every application that associated to the attribute group.

To edit attribute groups from the Attribute groups screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and the choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, select the attribute group that you want to edit, group and then choose **Edit**.
4. On **Edit attribute group**, update the description or JSON schema, then choose **Save changes**.

To edit attribute groups from the Attribute group details screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose the name of the attribute groups that you want to edit. Or select the attribute group that you want to edit, and then choose **View**. You're directed to the **Attribute group details** screen.
4. On **Attribute group details**, and then choose **Edit**.
5. On **Edit attribute group**, update the description or JSON schema, then choose **Save changes**.

Deleting attribute groups

With AWS Service Catalog AppRegistry, you can delete an attribute group from the **Attribute groups** screen or **Attribute group details** screen.

To delete attribute groups from the Attribute groups screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.

3. On **Attribute groups**, select the name of the attribute group that you want to delete, and then choose **Delete**.
4. Confirm your deletion, and then choose **Delete attribute group**.

To delete attribute groups from the Attribute group details screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose the name of the attribute groups that you want to edit. Or select the attribute group that you want to edit, and then choose **View**. You're directed to the **Attribute group details** screen.
4. On **Attribute group details**, choose **Delete**.
5. Confirm your deletion, and then choose **Delete attribute group**.

Associating and disassociating attribute groups

This topic describes how to associate and disassociate attribute groups in AppRegistry.

Topic

Associate attribute groups to a new application

The following procedure describes how to associate attribute groups to a new application.

To associate attribute groups to a new application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.

3. On **Applications**, choose **Create application**.
4. Under **Application name and description**, enter a name for your application. You can optionally enter a description for your application.
5. Under **Attribute groups**, select one or more attribute groups from the dropdown menu to associate to your application.
6. Choose **Create application**.

Associate attribute groups to an existing application from the Applications screen

The following procedure describes how to associate attribute groups to an existing application from the **Applications** screen.

To associate attribute groups to an existing application from the Applications screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to associate an attribute group to. Or select the application that you want to associate an attribute group to, and then choose **View**. You're directed to the **Application details** screen.
4. Choose **Attribute groups**, and then choose **Associate attribute group**.
5. Under **Attribute groups**, select an attribute group from the dropdown menu to associate to your application, and then choose **Save changes**.

Associate attribute groups to an existing application from the Attribute groups screen

The following procedure describes how to associate an attribute group to an existing application from the **Attribute groups** screen.

To associate attribute groups to an existing application from the Attribute groups screen

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.

3. On **Attribute groups**, choose **Create Attribute group**.
4. Under **New attribute group**, enter a name and description for your attribute group, and provide the JSON schema that captures your metadata taxonomy.

   Example: attribute group metadata

   ```
   {
    "Team" : "WebTeam",
    "Department": "10006",
    "ParentDept": "Research",
    "ContactAlias": "research@team.com"
   }
   ```

5. Under **Assign attribute group to an application**, select one or more applications to associate to your attribute group.
6. Choose **Create attribute group**.

Disassociate attribute groups from an existing application

The following procedure describes how to disassociate an attribute group from an existing application.

To disassociate attribute groups from an existing application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to disassociate an attribute group from. Or select the application that you want to disassociate an attribute group from, and then choose **View**. You're directed to the **Application details** screen.
4. Choose **Attribute groups**, and then select the attribute group that you want to disassociate from your application.
5. Choose **Disassociate**, confirm your disassociation, and then choose **Save changes**.

Sharing resources with accounts in your organization

You can share applications and attribute groups to an account, organizational unit, or organization.

AppRegistry integrates with AWS Resource Access Manager (AWS RAM), so you can view a list of resource shares associated with applications and attribute groups. For more information, see What is AWS Resource Access Manager? in the *AWS Resource Access Manager User Guide*.

When you create a resource share for an account, organization, or organizational unit, you can access the application or attribute group with the permission type that you select. For more information, see Sharing your AWS resources in the *AWS Resource Access Manager User Guide*.

This section describes how to create and manage resource shares for applications and attribute groups.

> **Note**
>
> When you create an application, AppRegistry vends a user tag called *the `awsApplication` tag*. You can add this tag to resources to identify which resources are associated with an application. The `awsApplication` tag is included in all shared applications. For more information, see The `awsApplication` tag.

Topics

- Creating and managing resource shares in applications
- Creating and managing resource shares in attribute groups
- Using AWS Resource Access Manager to share resources

Creating and managing resource shares in applications

This topic describes how to create and manage resource shares for AppRegistry applications. For information about creating applications, see Creating applications.

> **Note**
>
> Before a member account can enable cross-account sharing, the management account in the organization must enable sharing. For more information, see Sharing your AWS resources in the *AWS Resource Access Manager User Guide*.

To create a resource shares for a new application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose **Create application**.
4. Under **Application name and description**, enter a name for your application. You can optionally enter a description for your application.
5. To enable sharing for a management account, under **Application share configuration**, choose **Enable**.
   - On **Settings**, select **Enable sharing with AWS Organizations**, and then choose **Save settings**.
6. To enable sharing for a member account, under **Application share configuration**, choose **Turn on cross-account sharing**.
   a. For **Select Organization entity**, select your preferred organization entity (**AWS Organization Account**, **AWS Organization Unit**, or **AWS Organization**).
   b. For **ID**, enter the ID for your preferred organization entity.
   c. For **Share permission**, select **Allow associations** or **Read only**.
      - **Allow associations** when the selected account can associate resource collections and attribute groups to the application.
      - **Read only** when the selected account can view the application only.

   > **Note**
   >
   > When you select **Turn on cross-account sharing**, you can display the organizational structure in a heirarchy or list view by choosing **Display organizational structure**. You can add an organization entity by choosing **Add new**. You can delete an organization entity by choosing **Remove** next to the organization entity that you're deleting.

7. Complete your application configuration, and then choose **Create application**.

To create a resource share for an existing application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen.
3. On **Applications**, choose the name of the application that you want to create a resource share for. Or select the application that you want to create a resource share for, and then choose **View**. You're directed to the **Application details** screen.
4. On **Application details**, choose **Share**, and then choose **Create new share**.

> **Tip**
>
> The **Share** tab displays resource shares associated to the application. You can manage these resource shares by choosing **Manage in RAM console**. For more information, see What is AWS Resource Access Manager? in the *AWS Resource Access Manager User Guide*.

5. To enable sharing for a management account, under **Application share configuration**, choose **Enable**.
   - On **Settings**, select **Enable sharing with AWS Organizations**, and then choose **Save settings**.
6. To enable sharing for a member account, under **Application share configuration**, choose **Turn on cross-account sharing**.
   a. For **Select Organization entity**, select your preferred organization entity (**AWS Organization Account**, **AWS Organization Unit**, or **AWS Organization**).
   b. For **ID**, enter the ID for your preferred organization entity.
   c. For **Share permission**, select **Allow associations** or **Read only**.
      - **Allow associations** when the selected account can associate resource collections and attribute groups to the application.
      - **Read only** when the selected account can view the application only.

> **Note**
>
> When you select **Turn on cross-account sharing**, you can display the organizational structure in a heirarchy or list view by choosing **Display organizational structure**. You can add an organization entity by choosing **Add new**. You can delete an organization entity by choosing **Remove** next to the organization entity that you're deleting.

7. Confirm your resource share configuration, and then choose **Create share**.

Creating and managing resource shares in attribute groups

This topic describes how to create and manage resource shares for new and existing AppRegistry attribute groups. For information about creating attribute groups, see Creating attribute groups.

> **Note**
>
> Before a member account can enable cross-account sharing, the management account in the organization must enable sharing. For more information, see Sharing your AWS resources in the *AWS Resource Access Manager User Guide*.

To create a resource shares in new attribute group

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose **Create attribute groups**.
4. Under **Create attribute group**, enter a name and description for your attribute group, and provide the JSON schema that captures your metadata taxonomy.
5. To enable sharing for a management account, under **Attribute group share configuration**, choose **Enable**.
   - On **Settings**, select **Enable sharing with AWS Organizations**, and then choose **Save settings**.

6. To enable sharing for a member account, under **Attribute group share configuration**, choose **Turn on cross-account sharing**.

   a. For **Select Organization entity**, select your preferred organization entity (**AWS Organization Account**, **AWS Organization Unit**, or **AWS Organization**).

   b. For **ID**, enter the ID for your preferred organization entity.

   c. For **Share permission**, select **Allow associations** or **Read only**.

   - **Allow associations** when the selected account can associate resource collections and attribute groups to the application.
   - **Read only** when the selected account can view the application only.

   > **Note**
   >
   > When you select **Turn on cross-account sharing**, you can display the organizational structure in a heirarchy or list view by choosing **Display organizational structure**. You can add an organization entity by choosing **Add new**. You can delete an organization entity by choosing **Remove** next to the organization entity that you're deleting.

7. Complete your attribute group configuration, and then choose **Create attribute group**.

To create a resource share in an existing attribute group

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose the name of the attribute group that you want to create a resource share for. Or select the attribute group that you want to create a resource share for, and then choose **View**. You're directed to the **Attribute group details** screen.
4. On **Attribute group details**, choose **Share**, and then choose **Create new share**.

   > **Tip**
   >
   > The **Share** tab displays resource shares associated to the application. You can manage these resource shares by choosing **Manage in RAM console**. For more information,

see What is AWS Resource Access Manager? in the *AWS Resource Access Manager User Guide*.

5. To enable sharing for a management account, under **Attribute group share configuration**, choose **Enable**.

   - On **Settings**, select **Enable sharing with AWS Organizations**, and then choose **Save settings**.

6. To enable sharing for a member account, under **Attribute group share configuration**, choose **Turn on cross-account sharing**.

   a. For **Select Organization entity**, select your preferred organization entity (**AWS Organization Account**, **AWS Organization Unit**, or **AWS Organization**).

   b. For **ID**, enter the ID for your preferred organization entity.

   c. For **Share permission**, select **Allow associations** or **Read only**.

      - **Allow associations** when the selected account can associate resource collections and applications to the application.
      - **Read only** when the selected account can view the attribute group only.

   Note

   When you select **Turn on cross-account sharing**, you can display the organizational structure in a heirarchy or list view by choosing **Display organizational structure**. You can add an organization entity by choosing **Add new**. You can delete an organization entity by choosing **Remove** next to the organization entity that you're deleting.

7. Confirm your resource share configuration, and then choose **Create share**.

Using AWS Resource Access Manager to share resources

AppRegistry integrates with AWS Resource Access Manager (AWS RAM) to enable resource sharing. AWS RAM is a service that enables you to share AppRegistry applications and attribute groups with other AWS accounts or through AWS Organizations.

With AWS RAM you share resources that you own by creating a resource share. A resource share specifies the resources to share, and the consumers with whom to share them. Consumers can include:

- Specific AWS accounts inside or outside of its organization in AWS Organizations
- An organizational unit inside its organization in AWS Organizations
- Its entire organization in AWS Organizations

For more information about AWS RAM, see the AWS RAM User Guide.

Topics

Prerequisites for sharing applications and attributes

These are the prerequisites to share applications and attributes:

- You must own the application or attribute group in your AWS account. This means that the resource must be provisioned in your account. You cannot share an application or attribute group that has been shared with you.
- You must have access to AWS Organizations and AWS RAM.
- You must enable sharing with AWS Organizations. For more information, see Enable Sharing with AWS Organizations in the *AWS RAM User Guide*.

Sharing and unsharing applications or attribute groups

This section describes how to share or unshare an AppRegistry application or attribute group with AWS RAM.

When you share an application or attribute group using the AppRegistry console, you create a resource share. A resource share is an AWS RAM resource that lets you share your resources across AWS accounts. It specifies the resources to share, and the consumers with whom they are shared.

You can share an application or attribute group that you own using the AppRegistry console, AWS RAM console, or the AWS CLI.

- To share an application or attribute group that you own using the AppRegistry console, you can either share an application or attribute group when you create it in the AppRegistry console, or you can access **Shares** for the specific application or attribute group you want to share.

- To share an application or attribute group that you own using the AWS RAM console, see Creating a Resource Share in the *AWS RAM User Guide*.
- To share an application or attribute group that you own using the AWS CLI, use the create-resource-share command. For more information, see AWS Resource Access Manager API Reference.

To unshare a shared application or attribute group that you own, you must remove it from the resource share. You can do unshare using the AppRegistry console, AWS RAM console, or AWS CLI.

- To unshare a shared application or attribute group using the AppRegistry console, choose the application or attribute group from **Applications** or **Attribute Groups**. Then select **Shares**, and choose **Delete** for that application or attribute group.
- To unshare a shared an application or attribute group that you own using the AWS RAM console, see Updating a Resource Share in the *AWS RAM User Guide*.
- To unshare a shared an application or attribute group that you own using the AWS CLI, use the disassociate-resource-share command. For more information, see AWS Resource Access Manager API Reference.

Managing tags

Tags act as metadata to organize application resources. You create tags using key-value pairs. You add tags to applications and attribute groups, so you can group them by environment, owner, purpose, or other criteria.

> **Note**
>
> The tags discussed in this section are **not** the same as the the `awsApplication` tag. The `awsApplication` tag is a tag AppRegistry vends on your behalf when you create an application, and AWS automatically applies it to all resources in the application.

Topics

Adding and deleting tags in a new application

The following procedure describes how to add and delete tags in a new application. For information about creating a new application, see Creating applications.

To add and delete tags in a new application

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen where you can view all of your applications.
3. On **Applications**, choose **Create application**.
4. Under **Application name and description**, enter a name for your application. You can optionally enter a description for your application.
5. Under **Application tags**, choose **Add tag**, and then enter a Key/Value pair.
    a. To add another tag, choose **Add another**, and then enter a new key/value pair. You can create up to 50 tags for an application.
    b. To delete a tag, choose **Remove** next to the tag that you want to delete.
6. Complete your configuration, and then choose **Create application**.

> **Note**
>
> AppRegistry creates and adds tags that begin with `aws`, such as `aws:servicecatalog:applicationName`. These are considered internal tags and can't be removed.

Adding and deleting tags from the Application details screen

The following procedure describes how to add and delete tags from the **Application details** screen. For more information about using application details, see Using application details.

To add and delete tags from Application details

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Applications**. You're directed to the **Applications** screen where you can view all of your applications.
3. On **Applications**, choose the name of the application that you want to create a tag for. Or select the application that you want to create a tag for, and choose **View**. You're directed to the **Application details** screen.
4. On **Application details**, choose **Tags**.
5. Under **Add tags specific to this application**, enter a key/value pair, and then choose **Add tag**.
    a. To add another tag, enter a new key/value pair, and then choose **Add tag** again. You can create up to 50 tags for an application.
    b. To delete a tag, under **Application specific tags**, select the key/value pair that you want to remove, and then choose **Delete tag**.

Adding and deleting tags in a new attribute group

The following procedure describes how to add and delete in a new attribute group. For information about creating a new attribute group, see Creating attribute groups.

To add and delete tags in a new attribute group

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen.
3. On **Attribute groups**, choose **Create attribute group**.
4. Under **New attribute group**, enter a name and description for your attribute group, and and provide the JSON schema that captures your metadata taxonomy.
5. Under **Add tags**, enter a key/value pair to assign metadata to your attribute group.

a. To add another tag, choose **Add new item**, and then enter new key/value pair. You can create up to 50 tags for an attribute group.

b. To delete a tag, choose **Remove** next to the tag that you want to delete.

6. Complete your configuration, and then choose **Create attribute group**.

> **Note**
>
> AppRegistry creates and adds tags that begin with `aws`, such as `aws:servicecatalog:attributeGroupName` . These are considered internal tags and can't be removed.

Adding and deleting tags from Attribute group details

The following procedure describes how to add and delete tags from the **Attribute group details** screen. For more information about using attribute group details, see Using attribute group details.

To add and delete tags from Attribute group details

1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/
2. From the navigation pane, choose **AppRegistry**, and then choose **Attribute groups**. You're directed to the **Attribute groups** screen where you can view all of your attribute groups.
3. On **Attribute groups**, choose the name of the attribute group that you want to create a tag for. Or select the attribute group that you want to create a tag for, and choose **View**. You're directed to the **Attribute groups details** screen.
4. On **Attribute group details**, choose **Tags**.
5. Under **Add tags specific to this attribute group**, enter a key/value pair, and then choose **Add tag**.

   a. To add another tag, enter a new key/value pair, and then choose **Add tag** again. You can create up to 50 tags for an attribute group.

   b. To delete a tag, under **Attribute group specific tags**, select the key/value pair that you want to remove, and then choose **Delete tag**.

Security in AppRegistry

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. For more information about the compliance programs that apply to AppRegistry, 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 section helps you understand how to apply the shared responsibility model when using AppRegistry.

Topics

Data protection in AppRegistry

In the AWS shared responsibility model, you're responsible for ***security in the cloud*** while AWS is responsible for ***security of the cloud***. AWS protects the cloud infrastructure, and you protect

the content that's hosted in the cloud infrastructure. For information about data privacy and information about data protection in Europe, see the following:

- Data Privacy FAQ
- AWS Shared Responsibility Model and GDPR

You can use AWS Identity and Access Management (IAM) to set up user accounts and protect AWS account credentials. This grants users the required permissions to perfom work-related duties. As a best practice, we recommend that users create roles to access resources in AWS. For information about creating a role, see Creating a role to delegate permissions to an AWS service in the *IAM User Guide*. Other ways to secure data inlclude the following:

- Using multi-factor authentication (MFA) with each account.
- Using SSL/TLS to communicate with AWS resources. (TLS 1.2 or later recommended)
- Setting up API and user activity logging with AWS CloudTrail.
- Using AWS encryption solutions, including all default security controls within AWS services.
- Using an FIPS endpoint when accessing AWS through the command line interface or an API and if you need FIPS 140-2 validated cryptographic modules. For information about the available FIPS endpoints, see Federal Information Processing Standard (FIPS) 140-2.

> **Note**
>
> Data that you enter into AppRegistry and other AWS services can get picked up for inclusion in diagnostic logs.
> We recommend that you don't put sensitive or identifying information, such as customer account numbers, into free-form fields like `Name`. The same is true when using AppRegistry and other AWS services from the AWS Management Console, through the AWS CLI, by using an API, or by using one of the AWS SDKs.
> As a best practice, when you provide a URL to an external server, don't include information about credentials in the URL to validate your request.

Protecting Data with Encryption

Encryption at rest

AppRegistry uses Amazon DynamoDB databases that are encrypted at rest using Amazon-managed keys. For more information, refer to information about encryption at rest provided by Amazon DynamoDB.

Encryption in transit

AppRegistry uses Transport Layer Security (TLS) and client-side encryption of information in transit between the caller and AWS.

You can privately access AppRegistry APIs from Amazon Virtual Private Cloud (Amazon VPC) by creating VPC endpoints. With VPC endpoints, the routing between the VPC and AppRegistry is handled by the AWS network without the need for an internet gateway, NAT gateway, or VPN connection.

AWS PrivateLink powers the latest generation of VPC endpoints that AppRegistry uses. AWS PrivateLink is an AWS technology that enables the private connectivity between AWS services using Elastic Network Interfaces (ENIs) with private IPs in your VPCs.

AWS Identity and Access Management in AppRegistry

You must have credentials to access AWS Service Catalog AppRegistry. These credentials grant permission to access AWS resources, such as AWS Service Catalog portfolios or products. AppRegistry integrates with AWS Identity and Access Management (IAM). You can grant administrators the required permissions to create and manage products. You can grant end users the required permissions to launch products and manage provisioned products. Administrators and end users create and manage these polcies. Alternatively, AWS can create and manage them. To control access, you attach these policies to the roles and groups that you use with AppRegistry. For more information, see see IAM identities (users, user groups, and roles) in the *IAM User Guide*.

Topics

- Audience
- Troubleshooting AppRegistry identity and access
- Using service-linked roles for AWS Service Catalog AppRegistry

Audience

The permissions that you have through AWS Identity and Access Management (IAM) might depend on you AppRegistry role.

Administrator – If you're an AppRegistry administrator, you must have full access to the administrator console and IAM permissions that allow you to perform tasks, such as creating and managing portfolios and products, managing constraints, and granting access to end users.

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

Troubleshooting AppRegistry identity and access

The following information might help you diagnose and fix common issues that you can encounter when working with AppRegistry and AWS Identity and Access Management (IAM).

I'm unauthorized to perform an action in AppRegistry

If the AWS Management Console warns you that you're not authorized to perform an action, contact your administrator for assistance. Your administrator is the person who created your sign-in credentials.

Example: warning message

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
 awes:GetWidget on resource: my-example-widget
```

In the example, an error occurs when user `mateojackson` attempts to view details about the resource `my-example-widget`, but is unauthorized to perform the action `awes:GetPermission`.

I'm getting an access denied message when associating application resources

When you associate application resources with values for stacks or query tags that aren't supported, you receive a default error message:

Example: default error message

```
An error occurred (AccessDeniedException) when calling the AssociateResource
 operation: User: arn:aws:sts::[account number]:assumed-role/PringleTestRole/yingdon-
Isengard is not authorized to perform: servicecatalog:AssociateResource on resource:
 arn:aws:servicecatalog:us-west-2:[account number]:/applications/[application id] with
 an explicit deny
```

For more information, see the following:

- AssociateResource in the *AWS Service Catalog Developer Guide*
- DisassociateResource in the *AWS Service Catalog Developer Guide*
- Controlling the resource tag values associated to applications in the *AppRegistry Administrator Guide*

Using service-linked roles for AWS Service Catalog AppRegistry

This section describes how AWS Service Catalog AppRegistry uses the service-linked role `AWSServiceCatalogAppRegistryServiceRolePolicy` to create, update, and delete resource groups in your accounts. AWS Resource Groups allows you to manage your resources in groups instead individually. You can create resource groups that contain all of the resources in AWS CloudFormation stacks. For more information, see What are resource groups? in the *AWS Resource Groups User Guide*.

AppRegistry uses service-linked roles. A service-linked role is a type of IAM identity that links directly to an AWS service. For more information, see IAM identities (users, user groups, and roles) in the *IAM User Guide*. AppRegistry uses the service-linked role `AWSServiceRoleForAWSServiceCatalogAppRegistry`, which includes all of the permissions that are required to call other AWS services on your behalf.

Using service-linked roles make setting up AWS services more efficient because you don't have to add required permissions manually. AppRegistry defines its service-linked roles with the necessary permissions, The defined permissions include the trust policy and permissions policy. The permissions policy cannot be attached to any other entity (user, group, or role). For more information, see IAM identities (users, user groups, and roles) in the *IAM User Guide*.

You can delete a service-linked role only after deleting the related resources. This action protects your AppRegistry resources because you cannot inadvertently remove permission to access the resources.

> **Note**
>
> AppRegistry creates new tags on the resource groups `EnableAWSServiceCatalogAppRegistry` and `true`. If you modify these tags, AppRegistry loses permissions to manage service-linked resource groups that are created for applications and associated stacks.

Service-linked role permissions for AppRegistry

AppRegistry can call APIs on your behalf using the service-linked role `AWSServiceRoleForAWSServiceCatalogAppRegistry`. This role trusts the service principal ***servicecatalog-appregistry.amazonaws.com*** to assume the role.

The following role permissions policy allows AppRegistry to complete the following actions on the specified resources:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "cloudformation:DescribeStacks",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:CreateGroup",
                "resource-groups:Tag"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:RequestTag/EnableAWSServiceCatalogAppRegistry": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:DeleteGroup",
                "resource-groups:UpdateGroup",
                "resource-groups:GetTags",
                "resource-groups:Tag",
                "resource-groups:Untag"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
```

```
                    "aws:ResourceTag/EnableAWSServiceCatalogAppRegistry": "true"
                }
            }
        },
        {
            "Effect": "Allow",
            "Action": [
                "resource-groups:GetGroup",
                "resource-groups:GetGroupConfiguration"
            ],
            "Resource": [
                "arn:*:resource-groups:*:*:group/AWS_AppRegistry*",
                "arn:*:resource-groups:*:*:group/AWS_Cloudformation_Stack*"
            ]
        }
    ]
}
```

To allow an entity to create, edit, or delete a service-linked role, you must configure permissions. For more information, see Service-linked role permissions in the *IAM User Guide.*

You can allow an entity to create the service-linked role `AWSServiceRoleForAWSServiceCatalogAppRegistry` by adding this statement to the permissions policy for the IAM entity that creates the service-linked role.

```
{
    "Effect": "Allow",
    "Action": [
        "iam:CreateServiceLinkedRole"
    ],
    "Resource": "arn:aws:iam::*:role/aws-service-role/servicecatalog-
appregistry.amazonaws.com/AWSServiceRoleForAWSServiceCatalogAppRegistry*",
    "Condition": {"StringLike": {"iam:AWSServiceName": "servicecatalog-
appregistry.amazonaws.com"}}
}
```

Creating a service-linked role for AppRegistry

AppRegistry automatically creates your service-linked role when you create an application or update an existing application in the AWS Management Console, AWS CLI, or AWS API.

When customers request specific operations, AppRegistry automatically creates roles for them.

> **Important**
>
> If you completed an action with another AWS service that uses features that your service-linked role supports, the role can appear in your AWS account.

You can use the AWS Management Console to create a service-linked role with the use case `AWSServiceRoleForAWSServiceCatalogAppRegistry`.

You can use the AWS CLI or AWS API to create a service-linked role with the service name ***servicecatalog-appregistry.amazonaws.com***.

If you delete your service-linked role, you can create the role again in your account using the same process as before. For more information about creating and deleting service-linked roles, see Creating a service-linked role in the *IAM User Guide.*

Editing a Service-Linked Role for AppRegistry

After you create a service-linked role, you cannot change the name of the role because various entities might reference it. However, you can use the IAM console, AWS CLI, or AWS API to edit the role description. For more information, see Editing a service-linked role in the *IAM User Guide*.

Deleting a Service-Linked Role for AppRegistry

If you no longer need to use a feature or service that requires a service-linked role, we recommend that you delete the role. This way, you don't have an unused entity that's not actively monitored or maintained.

You must clean your service-linked role's resources before you can delete the role. You can use AppRegistry to clean the resources and then use the IAM console, AWS CLI, or AWS API to delete the role. For more information, see Deleting roles or instance profiles in the *IAM User Guide*.

To clean the resources that are associated with your service-linked role resources before you delete them, you must disassociate all resources from your applications. Then, you can disassociate all attribute groups from your applications. Finally, you can delete your applications.

Supported AWS Regions for AppRegistry service-linked roles

AppRegistry supports using service-linked roles in all AWS Regions where AppRegistry is available. For more information, see AWS service endpoints in the *AWS General Reference guide*.

Logging and Monitoring in AppRegistry

AppRegistry is a feature of AWS Service Catalog and uses the same logging and monitoring service. Since AWS Service Catalog integrates with AWS CloudTrail, so does AppRegistry. AWS CloudTrail is a service that captures all of the AppRegistry API calls and delivers the log files to an Amazon S3 bucket that you specify. For more information, see Logging AWS Service Catalog API Calls with AWS CloudTrail in the *AWS Service Catalog Administrator Guide*.

Compliance validation in AppRegistry

AppRegistry is a feature of AWS Service Catalog and uses the same compliance validation. For more information, see Compliance validation in AWS Service Catalog in the *AWS Service Catalog Administrator Guide*.

Resilience in AppRegistry

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.

For more information about AWS Regions and Availability Zones, and how AWS achieves resilience goals, see AWS Global Infrastructure.

Infrastructure Security in AppRegistry

As a managed service, AWS Service Catalog AppRegistry 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 AppRegistry through the network. Clients must support the following:

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

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

AWS managed polices

We recommend that you use AWS Service Catalog AppRegistry managed policies to add permissions to identies. For more information see IAM identities (users, user groups, and roles) in the *IAM User Guide*.

You could create customer managed policies. However, creating these types of polcies requires product expertise and time. Managed policies are designed to help you get started quickly because they provide permissions for common use cases. For more information, see Creating IAM policies and AWS managed policies in the *IAM User Guide*.

AWS services maintain and update managed policies. The permissions in these policies cannot be changed. To support new features, services periodically add permissions to managed policies. These updates effect all identities where you can find managed policies. Services typically update these policies during feature launches or when new operations become available. Services don't remove permissions from managed policies, so updates don't break existing permissions.

In addition, AWS supports managed policies for job functions that extend multiple services. For example, the `ReadOnlyAccess` policy provides read-only access to all services and resources. When services launch new features, AWS adds read-only permissions for new operations and resources. For a list of job functions and their descriptions, see AWS managed policies for job functions in the *IAM User Guide*.

AWSServiceCatalogAppRegistryFullAccess

AppRegistry provides you with `AWSServiceCatalogAppRegistryFullAccess`, an AWS managed policy that grants you full access to AppRegistry capabilities.

In this version of the policy, AppRegistry adds the resource group permissions `resource-groups:AssociateResource` and `resource-groups:DisassociateResource`, which allow you to call the resource groups for the AppRegistry `AssociateResource` and `DisassociateResource` APIs.

Note

You can use the AppRegistry `AssociateResource` and `DisassociateResource` APIs to add and remove resources associated with the `awsApplication` tag. For more information, see AssociateResource and DisassociateResource in the *AWS Service Catalog AppRegistry Developer Guide*.

AppRegistry also adds the permission `tag:GetResources`, which allows you to return all tagged resources. All tagged resources with defined tag keys and values can be included as resources for applications.

Permissions details

- **AWS CloudFormation** – Allows AppRegistry to update a stack in AWS CloudFormation.
- **Resource Groups** – Allows AppRegistry to create resource groups, return information about resource groups, delete resource groups, tag resource groups, return lists of tags associated with resource groups, remove tags from resource groups, retrieve resource tag information, and retrieve service configurations associated with resource groups.
- **IAM** – Allows AppRegistry to create an IAM role that's linked to a specific AWS service.

You can link to the following JSON policy in the IAM console or include it in your documentation.

```
{
     "Version": "2012-10-17",
     "Statement": [
      {
       "Sid": "AppRegistryUpdateStackAndResourceGroupTagging",
       "Effect": "Allow",
       "Action": [
        "cloudformation:UpdateStack",
        "tag:GetResources"
       ],
       "Resource": "*",
       "Condition": {
        "ForAnyValue:StringEquals": {
         "aws:CalledVia": "servicecatalog-appregistry.amazonaws.com"
        }
       }
      },
```

```
      {
       "Sid": "AppRegistryResourceGroupsIntegration",
       "Effect": "Allow",
       "Action": [
        "resource-groups:CreateGroup",
        "resource-groups:DeleteGroup",
        "resource-groups:GetGroup",
        "resource-groups:GetTags",
        "resource-groups:Tag",
        "resource-groups:Untag",
        "resource-groups:GetGroupConfiguration",
        "resource-groups:AssociateResource",
        "resource-groups:DisassociateResource"
       ],
       "Resource": "arn:aws:resource-groups:*:*:group/AWS_*",
       "Condition": {
        "ForAnyValue:StringEquals": {
         "aws:CalledVia": "servicecatalog-appregistry.amazonaws.com"
        }
       }
      },
      {
       "Sid": "AppRegistryServiceLinkedRole",
       "Effect": "Allow",
       "Action": "iam:CreateServiceLinkedRole",
       "Resource": "arn:aws:iam::*:role/aws-service-role/servicecatalog-
appregistry.amazonaws.com/AWSServiceRoleForAWSServiceCatalogAppRegistry*",
       "Condition": {
        "StringEquals": {
         "iam:AWSServiceName": "servicecatalog-appregistry.amazonaws.com"
        }
       }
      },
      {
       "Sid": "AppRegistryOperations",
       "Effect": "Allow",
       "Action": [
        "cloudformation:DescribeStacks",
        "servicecatalog:CreateApplication",
        "servicecatalog:GetApplication",
        "servicecatalog:UpdateApplication",
        "servicecatalog:DeleteApplication",
        "servicecatalog:ListApplications",
        "servicecatalog:AssociateResource",
```

```
        "servicecatalog:DisassociateResource",
        "servicecatalog:GetAssociatedResource",
        "servicecatalog:ListAssociatedResources",
        "servicecatalog:AssociateAttributeGroup",
        "servicecatalog:DisassociateAttributeGroup",
        "servicecatalog:ListAssociatedAttributeGroups",
        "servicecatalog:CreateAttributeGroup",
        "servicecatalog:UpdateAttributeGroup",
        "servicecatalog:DeleteAttributeGroup",
        "servicecatalog:GetAttributeGroup",
        "servicecatalog:ListAttributeGroups",
        "servicecatalog:SyncResource",
        "servicecatalog:ListAttributeGroupsForApplication",
        "servicecatalog:GetConfiguration",
        "servicecatalog:PutConfiguration"
       ],
       "Resource": "*"
      },
      {
       "Sid": "AppRegistryResourceTagging",
       "Effect": "Allow",
       "Action": [
        "servicecatalog:ListTagsForResource",
        "servicecatalog:UntagResource",
        "servicecatalog:TagResource"
       ],
       "Resource": "arn:aws:servicecatalog:*:*:*"
      }
     ]
}
```

AWSServiceCatalogAppRegistryReadOnlyAccess

`AWSServiceCatalogAppRegistryReadOnlyAccess` is an AWS managed policy that provides read-only access to AppRegistry capabilites. You can use this policy to associate tag keys and values with applications.

> **Note**
>
> All tagged resouces with defined tag keys and values can be included as resources for applications.

You can link to this JSON policy in the IAM console or include it in your documentation.

```
{
 "Version": "2012-10-17",
 "Statement": [
  {
   "Effect": "Allow",
   "Action": [
    "servicecatalog:GetApplication",
    "servicecatalog:ListApplications",
    "servicecatalog:GetAssociatedResource",
    "servicecatalog:ListAssociatedResources",
    "servicecatalog:ListAssociatedAttributeGroups",
    "servicecatalog:GetAttributeGroup",
    "servicecatalog:ListAttributeGroups",
    "servicecatalog:ListTagsForResource",
    "servicecatalog:ListAttributeGroupsForApplication",
    "servicecatalog:GetConfiguration"
   ],
   "Resource": "*"
  }
 ]
}
```

AWS managed policy updates

The following table includes information about the updates to the `AWSServiceCatalogAppRegistryFullAccess` and `AWSServiceCatalogAppRegistryReadOnlyAccess` policies.

Policy	Description	Date
AWSServiceCatalogAppRegistryFullAccess – Update to an existing policy	Added the resource group permission `tag:GetResources` , which allows you to retrieve resource tag information.	December 07, 2023
AWSServiceCatalogAppRegistryFullAccess – Update to an existing policy	Added the resource group permissions `resource-groups:AssociateResource` and `resource-groups:DisassociateResource` , which allow you to call the	November 13, 2023

Policy	Description	Date
	resource groups for `AssociateResource` and `DisassociateResource` .	
AWSServiceCatalogAppRegistryFullAccess – Update to an existing policy	Added the following: • `GetConfiguration` to retrieve a `TagKey` configuration from an account. • `PutConfiguration` to associate a `TagKey` configuration with an account. • The resource group actions `AssociateResource` and `DisassociateResource` , which are required to perform `AssociateResource` and `DisassociateResource` on a tag value.	November 17, 2022
AWSServiceCatalogAppRegistryReadOnlyAccess – Update to an existing policy	Added `GetConfiguration` to retrieve a `TagKey` configuration from an account.	November 17, 2022
AWSServiceCatalogAppRegistryServiceRolePolicy – Update to an existing policy	Updated `GetGroup` and `GetGroupConfiguration` permissions, which are required for AppRegistry to verify which service-linked resource groups exist in customer accounts.	October 24, 2022
AWSServiceCatalogAppRegistryFullAccess – Update to an existing policy	Added `ListAttributeGroupsForApplication` to list the details of all attribute groups associated with an application.	June 15, 2022
AWSServiceCatalogAppRegistryReadOnlyAccess – Update to an existing policy	Added `ListAttributeGroupsForApplication` to list the details of all attribute groups associated with an application.	June 15, 2022

Policy	Description	Date
AWSServiceCatalogAppRegistryServiceRolePolicy – Update to an existing policy	Added permissions to tag AWS Resource Groups when AWS Resource Groups are created.	August 24, 2021
AWSServiceCatalogAppRegistryFullAccess – Update to an existing policy	Added the following: • `UpdateStack` permissions to perform `SyncResource`, which updates the tags on the AWS Service Catalog stack. • `TagResource`, `ListTagForResources`, and `UntagResource` to perform tagging operations on resources. • `GetAssociatedResource`, as part of the integration with AWS Resource Groups.	August 24, 2021
AWSServiceCatalogAppRegistryReadOnlyAccess – Update to an existing policy	Added the following: • `ListTagForResources` to list all of the tags on a resource. • `GetAssociatedResource`, as part of the integration with AWS Resource Groups.	August 24, 2021

Troubleshooting in AppRegistry

If you encounter issues when working with AppRegistry, consult the topics in this section.

Topics

How to I resolve a resource tagging error for my application resources?

When a resource can't be successfully tagged or untagged with the `awsApplication` tag, the resource appears in the **Resource tagging error status** list. This list displays any resources that encountered tagging errors in the last 85 days, with a **Tag status** of **Error**.

Resource tagging errors can include any valid error code returned by the AWS service that hosts the resource that you want to tag. Common errors include the following:

- **You do not have permissions to tag or untag this resource** — Tagging and untagging resources requires specific permissions. Review Required permissions for Resource Groups and Toolkit for Eclipse for more information about using AWS managed policies or manually adding the necessary permissions to tag and untag resources.
- **You can't add a global resource to an application** — Not all global resources can be tagged or untagged from any AWS Region. Some global resources, such as Global Networks, must be tagged from a specific region only, usually the Home Region. You can Learn more about the differences between Regional and global resources in the *AWS Resource Access Manager User Guide*.

Document history

This table describes important additions to the AppRegistry documentation.

Feature	Description	Release date
Tag-sync task required permissions update	Updates to the tag-sync task required permissio ns to recommend the option of creating a new role when configuring a tag-sync task. This option allows AWS to automatic ally create a role with the correct trust permissio ns, ensuring the task can successfully manage the application's resources.	March 12, 2025
New `awsApplication` tag and application resource tag-sync	Updates to the `awsApplication` tag behavior and a new tag-sync feature for applicati on resources.	September 24, 2024
AWS managed policy updates – Updates to existing policies	Updates to the AWS Service Catalog AppRegist ry managed policy AWSServiceCatalogA ppRegistryFullAccess.	December 07, 2023
The `awsApplication` tag	The `awsApplication` tag is a user tag you can use to add and remove resources from applicati ons.	November 13, 2023

Feature	Description	Release date
AWS managed policy updates – Updates to existing policies	Updates to the AWS Service Catalog AppRegistry managed policy AWSServiceCatalogAppRegistryFullAccess.	November 13, 2023
AWS managed policy updates – Updates to existing policies	Updates to the following AWS Service Catalog AppRegistry managed policies: • AWSServiceCatalogAppRegistryFullAccess • AWSServiceCatalogAppRegistryReadOnlyAccess	November 17, 2022
AppRegistry Administrator Guide	The release of the AppRegistry Administrator Guide.	June 15, 2022