**AWS Service Catalog Connector for ServiceNow**

v2.0.2 Installation Instructions

Author: MaSonya Scott; Co-Authors: Kenneth Walsh and Brian Terry

To help customers integrate provisioning secure, compliant, and pre-approved AWS products into their ServiceNow portal, AWS created the AWS Service Catalog Connector for ServiceNow.

AWS Service Catalog Connector for ServiceNow synchronizes AWS Service Catalog portfolios and products with the ServiceNow Service Catalog to enable ServiceNow users to request approved AWS products via ServiceNow.

**Topics**

* Background
* Getting Started
* Release Notes
* Baseline Permissions
* Configure AWS Service Catalog
* Configure ServiceNow
* Validate Configurations
* ServiceNow Additional Administrator Features
* Upgrade Instructions

**Background**

AWS Service Catalog allows you to centrally manage commonly deployed AWS services and provisioned software products. It helps your organization achieve consistent governance and compliance requirements, while enabling users to quickly deploy only the approved AWS services they need.

[ServiceNow](https://www.servicenow.com/) is an enterprise service management platform that places a service‑oriented lens on the activities, tasks, and processes that make up day‑to‑day work life to enable a modern work environment. [ServiceNow Service Catalog](https://www.servicenow.com/products/it-service-automation-applications/service-catalog.html) is a self-service application that end users can use to order IT services based on request fulfillment approvals and workflows.

**Getting Started**

Before installing the AWS Service Catalog Connector for ServiceNow, verify that you have the necessary permissions in your AWS account and ServiceNow instance.

**AWS prerequisites**

To get started you need an AWS account to configure your AWS portfolios and products. For details, see [Setting Up for AWS Service Catalog](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/setup.html).

For each AWS account, the Connector for ServiceNow also requires two AWS Identity and Access Management (IAM) users and two IAM roles:

* An IAM user to sync AWS Service Catalog portfolios and products to ServiceNow Service Catalog items.
* An IAM role configured as an AWS Service Catalog end user and assigned to each portfolio.
* An IAM end user to assume the previous end user role. This end user has a baseline of permissions to provision AWS services in the ServiceNow Service Catalog. This ServiceNow end user is linked to the end user role in IAM.
* An IAM launch role used to place baseline AWS service permissions into the AWS Service Catalog launch constraints. Configuring this role enables segregation of duty by provisioning product resources on behalf of the ServiceNow end user.

Baseline permissions enable an end user to provision the following AWS services: Amazon Simple Storage Service and Amazon Elastic Compute Cloud. To allow end users to provision AWS services beyond the baseline permissions, you must include the additional AWS service permissions to the launch role. For information about initial permissions setup actions, see Baseline Permissions.

**Note**

To use an AWS CloudFormation template to set up the AWS configurations of the Connector for ServiceNow, see [Connector for ServiceNow-AWS Configuration](https://s3.amazonaws.com/servicecatalogconnector/SC_ConnectorForServiceNowv2.0.2-AWS_Configurations.yml).

**ServiceNow Prerequisites**

In addition to the AWS account, you need a ServiceNow instance to install the ServiceNow Connector scoped application. The initial installation should occur in either an enterprise sandbox or a [ServiceNow Personal Developer Instance](https://developer.servicenow.com/app.do#!/training/article/app_store_learnv2_buildmyfirstapp_kingston_servicenow_basics/app_store_learnv2_buildmyfirstapp_kingston_personal_developer_instances?v=kingston) (PDI), depending on your organization’s technology governance requirements. The ServiceNow administrator needs the admin role to install the Connector for ServiceNow scoped application.

**Release Notes**

**Version 2.0.2** of the AWS Service Catalog Connector for ServiceNow includes:

* Support for AWS CloudFormation StackSets, enabling launch of AWS Service Catalog products across multiple regions and accounts.
* Support for AWS CloudFormation Change Sets, enabling a preview of resource changes from a launch or update.
* Display of AWS Service Catalog portfolios (including correlated products) as sub-categories in the ServiceNow Service Catalog.

This version also includes prior AWS Service Catalog Connector for ServiceNow features such as:

* Support AWS Service Catalog self-service actions.
* Enable ServiceNow administrators to delete AWS Service Catalog products in ServiceNow that do not have self-service actions associated.
* Render AWS Service Catalog products in the ServiceNow Portal page.
* Enable multi-account support.
* Request update against an existing AWS Service Catalog product provisioned in ServiceNow.
* Validate AWS Regions and identities associated with syncing AWS and ServiceNow.
* Sync product details in the My Asset/CMDB view.

**Baseline Permissions**

This section provides instructions on how to set up the baseline AWS users and permissions needed for the AWS Service Catalog Connector for ServiceNow. For each AWS account, the Connector for ServiceNow requires two IAM users and three roles:

* **AWS Service Catalog Sync User**: IAM user to sync AWS portfolios and products to ServiceNow catalog items (ServiceCatalogAdminReadOnly managed policy).
* **AWS Service Catalog End User role**: IAM role configured as an AWS Service Catalog end user and assigned to each AWS Service Catalog portfolio.
* **AWS Service Catalog End User**: Enables Connector for ServiceNow to provision AWS products by assuming a role that contains the trust relationship with the account and policies needed for the end user privileges in AWS Service Catalog.
* **SCConnect Launch role**: IAM role used to place baseline AWS service permissions into the AWS Service Catalog launch constraints. Configuring this role enables segregation of duty through provisioning product resources on behalf of the ServiceNow end user. The SCConnectLaunch role baseline contains permissions to Amazon EC2 and Amazon S3 services. If your products contain more AWS services, you must either include those services in the SCConnectLaunch role or create new launch roles.

**Creating AWS Service Catalog Sync User**

The following section describes how to create the AWS Service Catalog Sync user and associate the appropriate IAM permission. To perform this task, you need IAM permissions to create new users.

**To create AWS Service Catalog sync user**

1. Go to [Creating IAM Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html). Following the instructions there, create a policy called **SCConnectorAdmin** for ServiceNow administrators to delete AWS Service Catalog products in ServiceNow that do not have self-service actions associated. Copy the following policy and paste it into **Policy Document**:

{

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

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"servicecatalog:DisassociateProductFromPortfolio",

"servicecatalog:DeleteProduct",

"servicecatalog:DeleteConstraint",

"servicecatalog:DeleteProvisionedProductPlan",

"servicecatalog:DeleteProvisioningArtifact"

],

"Resource": "\*"

}

]

}

1. Go to [Creating an IAM User in Your AWS Account](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html). Following the instructions there, create a sync user (that is, SCSyncUser). The user needs programmatic and AWS Management Console access to follow the Connector for ServiceNow installation instructions.
2. Set permissions for your sync user (SCSyncUser). Choose **Attach existing policies directly** and select the **ServiceCatalogAdminReadOnlyAccess** and **SCConnectorAdmin** policies.
3. Review and choose **Create User**.
4. Note the Access and Secret Access information. Download the .csv file that contains the user credential information.

**Creating AWS Service Catalog End User**

The following section describes how to create the AWS Service Catalog end user and associate the appropriate IAM permission. To perform this task, you need IAM permissions to create new users.

If you are upgrading from an earlier version of the Connector, note that the **ServiceCatalogServiceNowAdditionalPermissions** AWS policy is no longer needed for the Connector for ServiceNow. Proceed to the **Create a SnowEndUser** role step.

**To create AWS Service Catalog end user**

* Go to [Create a role](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_create_for-user.html). Following the instructions there, create a role for the ServiceNow end user to assume (such as SnowEndUser).
* For products with StackSets, you need to create a StackSet inline policy. With AWS CloudFormation StackSets, you are able to create products that are deployed across multiple accounts and regions. Using an administrator account, you define and manage an AWS Service Catalog product, and use it as the basis for provisioning stacks into selected target accounts across specified regions. you need to have the necessary permissions defined in your AWS accounts.

To set up the necessary permissions go to [Granting Permissions for Stack Set Operations](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/stacksets-prereqs.html) to grant permissions for Stack Set Operations. Following the instructions there, create an **AWSCloudFormationStackSetAdministrationRole** and an **AWSCloudFormationStackSetExecutionRole.**

* + Create the Stackset inline policy to enable provisioning a product across multiple regions within one account.

{

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

"Statement": [

{

"Action": [

"sts:AssumeRole"

],

"Resource": [

"arn:aws:iam::123456789123:role/AWSCloudFormationStackSetExecutionRole"

],

"Effect": "Allow"

},

{

"Effect": "Allow",

"Action": [

"iam:GetRole",

"iam:PassRole"

],

"Resource": "arn:aws:iam::123456789123:role/AWSCloudFormationStackSetAdministrationRole"

}

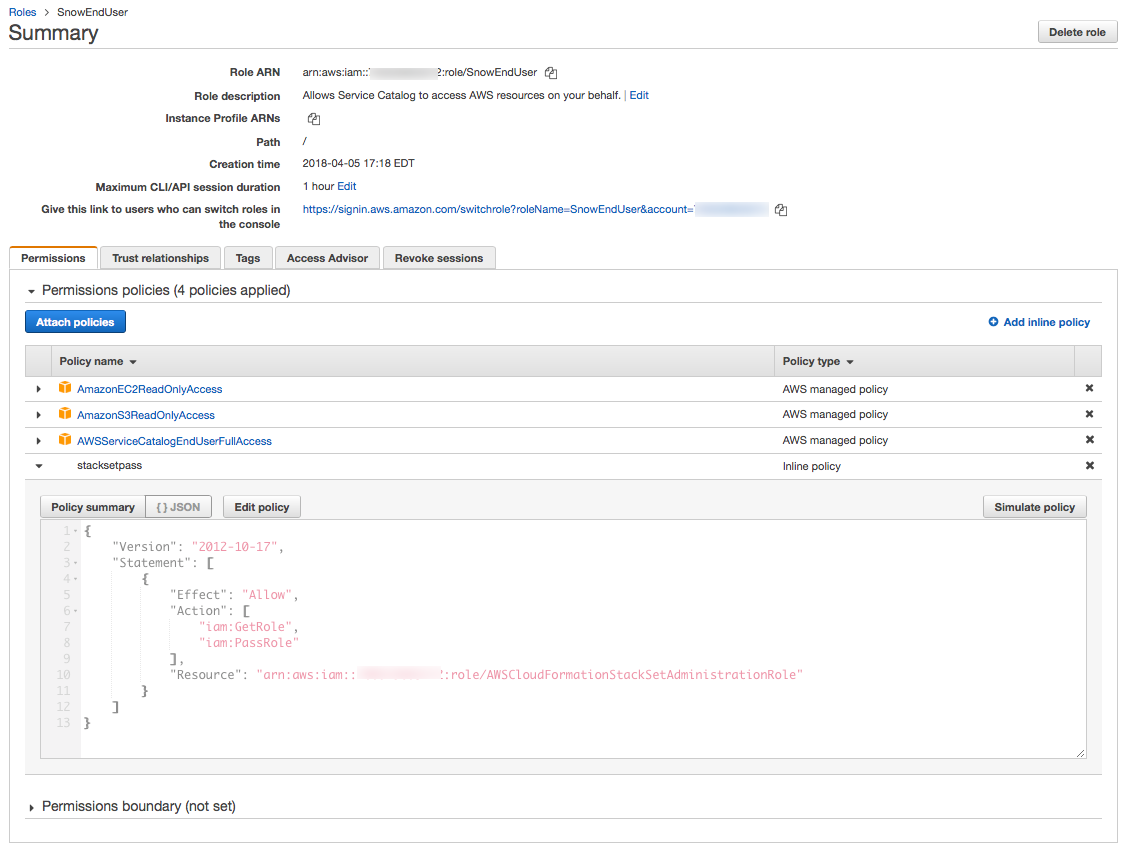
]

}

*Note: replace number string in blue text with your account information. The* [Connector for ServiceNow -AWS Configuration](https://s3.amazonaws.com/servicecatalogconnector/SC_ConnectorForServiceNowv2.0.2-AWS_Configurations.yml) file includes the Stackset permissions.

* + Add the following permissions (policies) to the role:
    - **AWSServiceCatalogEndUserFullAccess (AWS managed policy)**
    - StackSet (inline policy)
    - AmazonS3ReadOnlyAccess\*
    - AmazonEC2ReadOnlyAccess

\* Note: For AWS Service Catalog products with StackSets, you need to modify the SnowEndUser role to include the ReadOnly permissions for the service(s) you want to provision. For example, to provision an S3 bucket, include the AmazonS3ReadOnlyAccess policy to the SnowEndUser role.



* + Create a trust relationship on the SnowEndUser role to the account. Copy and paste the following text into the Trust Relationship (replacing the number string for ARN with your account information):

{

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

"Statement": [

{

"Effect": "Allow",

"Principal": {

"AWS": "arn:aws:iam::123456789123:root"

},

"Action": "sts:AssumeRole",

"Condition": {}

}

]

}

*Note: replace number string in blue text with your account information.*

* Go to [Create a policy](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html). Following the instructions there, create a policy called **StsAssume-SC**. Copy and paste the following text into the JSON editor (replacing the number string for ARN with your account information):

{

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

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": "sts:AssumeRole",

"Resource": "arn:aws:iam:: 123456789123:role/SnowEndUser"

}

*Note: replace number string in blue text with your account information.*

* Go to [Creating an IAM User in Your AWS Account](https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html). Following the instructions there, create a user (such as SCEndUser). The user needs programmatic and AWS Management Console access to follow the ServiceNow Connector installation instructions.
* Attach the assume policy (StsAssume-SC) to your end user (SCEndUser). Choose **Attach existing policies directly** and select **StsAssume-SC**.
* Review and choose **Create User**.
* Note the Access and Secret Access information. Download the .csv file that contains the user credential information.

**Creating SCConnectLaunch Role**

The following section describes how to create the **SCConnectLaunch** role. This role is used to place baseline AWS service permissions into the AWS Service Catalog launch constraints. For more information, see [AWS Service Catalog Launch Constraints](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/constraints-launch.html).

**To create SCConnectLaunch role**

1. Create the **AWSCloudFormationFullAccess** policy. Choose **create policy** and then paste the following in the JSON editor:

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [

"cloudformation:DescribeStackResource",

"cloudformation:DescribeStackResources",

"cloudformation:GetTemplate",

"cloudformation:List\*",

"cloudformation:DescribeStackEvents",

"cloudformation:DescribeStacks",

"cloudformation:CreateStack",

"cloudformation:DeleteStack",

"cloudformation:DescribeStackEvents",

"cloudformation:DescribeStacks",

"cloudformation:GetTemplateSummary",

"cloudformation:SetStackPolicy",

"cloudformation:ValidateTemplate",

"cloudformation:UpdateStack",

"cloudformation:CreateChangeSet",

"cloudformation:DescribeChangeSet",

"cloudformation:ExecuteChangeSet",

"cloudformation:DeleteChangeSet",

"s3:GetObject"

],

"Resource": "\*"

}

]

}

Note: **AWSCloudFormationFullAccess** now includes additional permissions for ChangeSets.

1. Create a policy called **ServiceCatalogSSMActionsBaseline**. Follow the instructions on [Creating IAM Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html), and paste the following into the JSON editor:

{

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

"Statement": [

{

"Sid": "Stmt1536341175150",

"Action": [

"servicecatalog:ListServiceActionsForProvisioningArtifact",

"servicecatalog:ExecuteprovisionedProductServiceAction",

"ssm:DescribeDocument",

"ssm:GetAutomationExecution",

"ssm:StartAutomationExecution",

"ssm:StopAutomationExecution",

"cloudformation:ListStackResources",

"ec2:DescribeInstanceStatus",

"ec2:StartInstances",

"ec2:StopInstances"

],

"Effect": "Allow",

"Resource": "\*"

}

]

}

1. Create the **SCConnectLaunch** role. Assign the trust relationship to AWS Service Catalog.

{

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

"Statement": [

{

"Sid": "",

"Effect": "Allow",

"Principal": {

"Service": "servicecatalog.amazonaws.com"

},

"Action": "sts:AssumeRole"

}

]

}

1. Attach the relevant policies to the **SCConnectLaunch** role. Attach the following baseline IAM policies:
   * **AmazonEC2FullAccess** (AWS managed policy)
   * **AmazonS3FullAccess** (AWS managed policy)
   * **AWSCloudFormationFullAccess** (custom managed policy)
   * **ServiceCatalogSSMActionsBaseline** (custom managed policy)

# Configure AWS Service Catalog

Now that you have created two IAM users with baseline permissions in each account, the next step is to configure AWS Service Catalog. This section describes how to configure AWS Service Catalog to have a portfolio that includes an Amazon S3 bucket product. Use the Amazon S3 template located at [Creating an Amazon S3 Bucket for Website Hosting for your preliminary product](https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/quickref-s3.html#scenario-s3-bucket-website). Copy and save the Amazon S3 template to your device.

**To configure AWS Service Catalog**

1. Create a portfolio by following the steps at [Create an AWS Service Catalog Portfolio](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/getstarted-portfolio.html).
2. To add the Amazon S3 bucket product to the portfolio you just created, in the AWS Service Catalog console, on the **Upload new product** page, enter product details.
3. For **Select template**, choose the Amazon S3 bucket AWS CloudFormation template you saved to your device.
4. Set **Constraint type** to **Launch** for the product that you just created with the SCConnectLaunch role in the baseline permissions. For additional launch constraint instructions, see [AWS Service Catalog Launch Constraints](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/constraints-launch.html).

**Note**

The AWS configuration design requires each AWS Service Catalog product to have either a launch or StackSet constraint. Failure to follow this step may result in an “Unable to Retrieve Parameter” message within ServiceNow Service Catalog.

**Creating Stack Set Constraint**

AWS CloudFormation Stack Sets enable users to create products that are deployed across multiple accounts and regions.

**To apply a stack set constraint to an AWS Service Catalog product**

1. Go to AWS Service Catalog (as a catalog admin)
2. Choose the portfolio that contains the product.
3. Expand **Constraints** and choose **Add constraints**.
4. Choose the product from **Product** and set **Constraint type** to **Stack Set**. Choose **Continue**.
5. On the **Stack Set constraint page**, enter a description.
6. Choose the account(s) in which you want to create products.
7. Choose the region(s) in which you want to deploy products. Products are deployed in these regions in the order that you specify.
8. Choose the **AWSCloudFormationStackSetAdministrationRole** Role that will be used to manage your target accounts.
9. Choose the **AWSCloudFormationStackSetExecutionRole** Role that the Administrator Role will assume.
10. Choose **Submit**.

Note: The [Connector for ServiceNow-AWS Configuration](https://s3.amazonaws.com/servicecatalogconnector/SC_ConnectorForServiceNowv2.0.2-AWS_Configurations.yml) template creates the permissions as well as outputs needed for StackSet constraints. Example StackSet outputs:

**SCStackSetAdministratorRoleARN**

arn:aws:iam::123456789123:role/AWSCloudFormationStackSetAdministrationRole **SCIAMStackSetExecutionRoleName**

AWSCloudFormationStackSetExecutionRole

**SCIAMAdminRoleARN**

arn:aws:iam::123456789123:role/AWSCloudFormationStackSetAdministrationRole

*Note: replace number string in blue text with your account information.*

# 

Note: Service Catalog products can only have either a StackSet or Launch constraint, but not both.

1. Add the SnowEndUser IAM role to the AWS Service Catalog portfolio. For additional user access instructions, see [Granting Access to Users](https://docs.aws.amazon.com/servicecatalog/latest/adminguide/catalogs_portfolios_users.html).

# Configure ServiceNow

After completing the IAM and AWS Service Catalog configurations, the next configuration area to set up is ServiceNow. Installation tasks within ServiceNow include:

* Clear the ServiceNow platform cache.
* Clear the Web Browser cache.
* Install the ServiceNow Connector scoped application, and upload and commit the ServiceNow Connector Update Set.
* Configure ServiceNow platform system admin components.
* Configure AWS Service Catalog Connector scoped application, including accounts, scheduled jobs sync, and permissions.

## Clear the ServiceNow Platform Cache

Before installing the AWS Service Catalog scoped app, we recommend that you clear the ServiceNow platform cache by typing in the following URL:https://[InsertServiceNowInstanceNameHere]/cache.do

**Note**

Ensure that you install the update set in a non-production/sandbox environment. Consult a ServiceNow system administrator if you need approval to clear the ServiceNow platform cache.

## Clear the Web Browser Cache

## Clear the Web Browser cache to clear previous rendered product forms.

## Installing ServiceNow Connector Scoped Application

The AWS Service Catalog Connector for ServiceNow is released as a conventional ServiceNow scoped application via a [ServiceNow Update Set](https://docs.servicenow.com/bundle/london-application-development/page/build/system-update-sets/reference/get-started-update-sets.html). ServiceNow update sets are code changes to the out-of-the-box platform and enable developers to move code across ServiceNow instance environments. The Connector for ServiceNow update set is available to download in the  [ServiceNow store](https://store.servicenow.com/sn_appstore_store.do#!/store/application/f0b117a3db32320093a7d7a0cf961912/). For users installing the update set on a ServiceNow Personal Developer Instance (PDI), download the code from [Connector for ServiceNow version 2.0.2 update set.](https://s3.amazonaws.com/servicecatalogconnector/AWS_SC_update_set_2.0.2.xml.gz)

The Connector for ServiceNow version 2.0.0 update set may be applied to a “Kingston,” or “London,” or "Madrid" platform release of ServiceNow.

If you do not already have a ServiceNow instance, begin with the first step below. If you already have a ServiceNow instance, proceed to **To download AWS Service Catalog Connector for ServiceNow**.

**To obtain a ServiceNow instance**

1. Go to [Obtaining a Personal Developer Instance](https://developer.servicenow.com/app.do#!/training/article/app_store_learnv2_buildmyfirstapp_jakarta_servicenow_basics/app_store_learnv2_buildmyfirstapp_jakarta_personal_developer_instances?v=jakarta).
2. Create ServiceNow developer program credentials.
3. Follow the instructions for requesting a ServiceNow instance.
4. Capture your instance details, including URL, administrative ID, and temporary password credentials.

**To download AWS Service Catalog Connector for ServiceNow**

1. From your ServiceNow dashboard, type **plugins** into the navigation panel in the upper left.
2. When the **System Plugins** page populates, next to the dropdown that says **Name**, search for **user criteria**.
3. Choose **User Criteria Scoped API** and then choose **Activate**.
4. From the [ServiceNow Store](https://store.servicenow.com/sn_appstore_store.do#!/store/application/f0b117a3db32320093a7d7a0cf961912/), download the AWS Service Catalog Connector. When prompted, log in with your administrator credentials.

**To install the update set**

1. From your ServiceNow dashboard, type **update sets** into the navigation panel in the upper left.
2. Choose **Retrieved Update Sets** from the results.
3. Select **Import Update Set from XML** and upload the release XML file.
4. Select the **AWS Service Catalog Connector for ServiceNow** update set.
5. Choose **Preview Update Set**, which makes ServiceNow validate the connector update set.
6. Choose **Update**.
7. Choose **Commit Update Set** to apply the update set and create the application. This procedure should complete 100%.

## Configuring ServiceNow Platform System Admin Components

To enable the AWS Service Catalog Connector for ServiceNow scoped application named **AWS Service Catalog**, the system admin must configure specific platform tables, forms, and views.

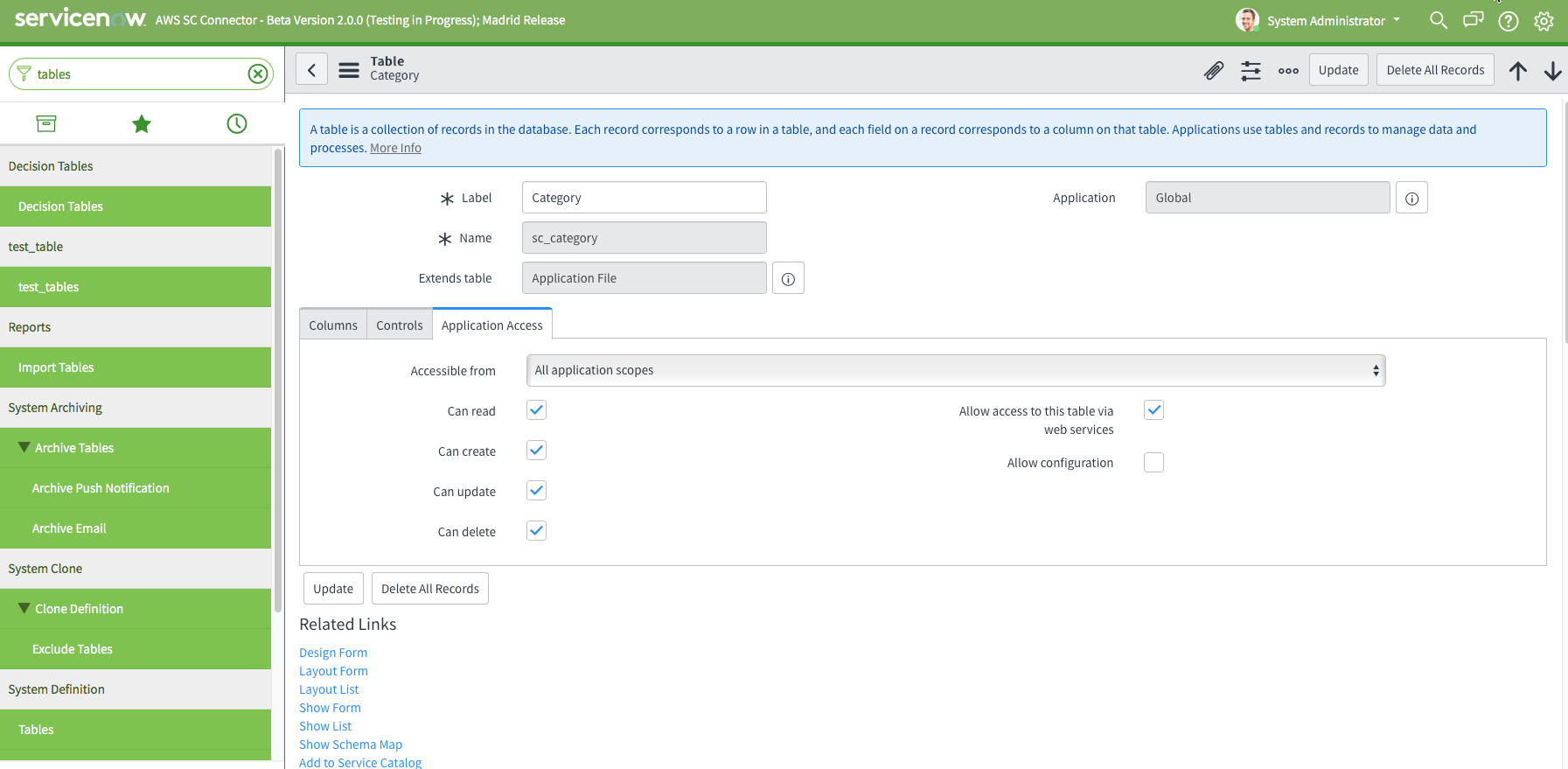
**Note**

If you are upgrading from an earlier version, the permissions on ServiceNow Platform tables (User Criteria and Catalog Variable Set) are no longer needed for the Connector for ServiceNow.

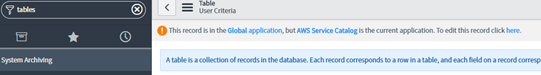
### Enable permissions on ServiceNow Platform tables (Category and Catalog Item Category).

### For AWS products to display under AWS portfolios as sub-categories in the ServiceNow Service Catalog, you need to modify the Application Access form for Category and Catalog Item Category tables.

1. Enter "Tables" in the Navigator and select System Definition -> Tables
2. In the list of tables search for a table with Label "Category" (or Name "sc\_category"). The list of tables will be displayed. Choose Category to view the form defining the table.
3. Choose the "Application Access" tab on the form and choose the "Can Create", “Can Update, and "Can delete" checkboxes on the form. Select the "Update" button.



Note: You may need to click on the link at the top to edit the record.



1. Repeat the steps used on the Category table above for the "Catalog Item Category" table (type **sc\_cat\_item\_category** in the “Go to Name Search” field).

## ServiceNow Permissions for Administrators of the Connector Scoped App.

The AWS Service Catalog scoped app comes with two ServiceNow roles that enable access to configure the application. This enables system admins to grant one or more users’ privileges to administer the application without having to open up full sysadmin access to them. These roles can be assigned either to individual users or to one administrator user.

**To set up application administrator privileges**

1. Type **Users** in the navigator and select **System Security – Users**.
2. Select a user to grant one or both previous roles (such as admin) to. You can also [Create a User](https://docs.servicenow.com/bundle/jakarta-platform-administration/page/administer/users-and-groups/task/t_CreateAUser.html).
3. Choose **Edit** on the **Roles** tab of the form.
4. Filter the collection of roles by the prefix “x\_”.
5. Choose one or both of the following and add them to the user: **x\_126749\_aws\_sc\_account\_admin**,**x\_126749\_aws\_sc\_portfolio\_manager**
6. Choose **Save**.

**To add AWS Service Catalog to ServiceNow Service Catalog categories**

1. Navigate to **Self Service | Service Catalog** and select the **Add content** icon in the upper right.
2. Select the **AWS Service Catalog Product** entry. Add it to your catalog home page by choosing the first **Add Here** link on the second row of the selection panel at the bottom of the page.

**To add a change request type**

1. If you are upgrading from a previous version of the AWS Service Catalog scoped app, you must remove the **AWS Product Termination** change request type before creating a new change request type.
2. You must add a new change request type called **AWS Provisioned Product Event** for the scoped application to trigger an automated change request in Change Management. For instructions, see [Add a new change request type](https://docs.servicenow.com/bundle/istanbul-it-service-management/page/product/change-management/task/t_AddNewChangeType.html).
3. Open an existing change request.
4. Open the context (right-click) menu for **Type** and then choose **Show Choice List**.
5. Choose **New** and fill in the following fields:
   * **Table**: **Change Request**
   * **Label**: **AWS Provisioned Product Event**
   * **Value**: **AWSProvisionedProductEvent**
   * **Sequence**: pick the next unused value
6. Submit the form.

## Configuring AWS Service Catalog Connector Scoped Application

Having installed and configured the AWS Service Catalog Connector for ServiceNow in the previous procedure, you must configure the AWS Service Catalog scoped application and applicable roles.

**To configure the AWS Service Catalog scoped application and applicable roles**

1. Got to User Administration on the ServiceNow>Roles>New, create a role called **order\_aws\_sc\_products**. This role is granted to any users with permission to order AWS Service Catalog products. For instructions, see [Create a role](https://docs.servicenow.com/bundle/london-platform-administration/page/administer/roles/task/t_CreateARole.html).
2. Grant roles to the following users:
   * **System Administrator (admin)**: For simplicity in this example, user **admin** is designated as the administrator of the AWS Service Catalog scoped application. Grant this user both of the administrative permissions from the adapter, **x\_126749\_aws\_sc\_portfolio\_manager** and**x\_126749\_aws\_sc\_account\_admin**. In a real scenario, these roles would likely be granted to two different users.
   * **Abel Tuter**: The user **abel.tuter** is chosen as an illustrative end user. Grant Abel the new role **order\_aws\_sc\_products**. This allows him to order products from AWS.

## Configuring Accounts

1. Log in as the system administrator in ServiceNow.
2. In the AWS Service Catalog scoped app **Accounts** menu, create two accounts, one for sync and one for provisioning: **snow-stsuser-account** and **snow-sync-account.** Note that the names here are chosen for convenience to make it easy to see which IAM user they correspond to (these are the users created in the AWS setup).
3. The **snow-stsuser-account** account has no regions configured. The **snow-sync-account** user has one region configured, matching the region where AWS Service Catalog is defined. You validate this in the next section.
4. Note that you need to use the keys and secret keys from the users you created in AWS.

## Validating Connectivity to AWS Regions

You can now validate connectivity to AWS regions between the ServiceNow **snow-sync-account** and the AWS IAM SyncUser.

**To validate connectivity to AWS regions**

1. In the AWS Service Catalog scoped app, choose **Accounts**.
2. Select **snow-sync-account** and choose **Validate Regions**.
3. A successful connection result in the message, “Successfully performed AWS Service Catalog SearchProductsAsAdmin action in each referenced Region.”

If the AWS IAM access key or secret access key are incorrect, you will receive the message similar to the following: Error performing AWS Service Catalog SearchProductsAsAdmin action in one or more Regions: us-east-1: The security token included in the request is invalid. Check that the access key and secret access key are correct.

## Manually Syncing Scheduled Jobs

During the initial setup, manually execute the sync instead of waiting for Scheduled Jobs to run.

**To sync the accounts manually**

1. Log in as system administrator.
2. Find **Scheduled Jobs** in the navigator panel.
3. Search for job **Sync all Accounts**, select it, and choose **Execute Now**.

**Note**

If you do not see **Execute Now** in the upper left corner, choose **Configure Job Definition**. **Execute Now** will be visible.

## Granting Access to Portfolios

Data is visible in the AWS Service Catalog scoped app menus after the adapter’s scheduled synchronization job has run.

To grant access to AWS Service Catalog products in ServiceNow, you must establish a link between the AWS SnowEndUser role discovered from the **Sync All Scheduled Job** and snow-stsuser-account entry created in the ServiceNow AWS Service Catalog scoped app.

**To grant access to AWS Service Catalog products in ServiceNow**

1. In the AWS Service Catalog scoped app, choose the **Identities** module.
2. Select the ARN address for the AWS SnowEndUser role and assign it to account **snow-stsuser-account**. You can double-click the cell in the account column, or click the SCEndUser user name and edit the form presented.

**Role Grants** is available within the **Identities** modules to conveniently associate the ServiceNow role **order\_aws\_sc\_products** to the AWS SnowEndUser role identity.

1. Choose **New** and enter the **Role** of **order\_aws\_sc\_products** and the SnowEndUser identity.
2. Choose **Submit**.

The **Identities** module now has a view of the associated role. You can test the AWS identity to determine if the ServiceNow end user with the **order\_aws\_sc\_products** role can order an AWS Service Catalog product.

**To test access to portfolios**

1. Choose the **Test Authorization** button shown in the AWS identity module.
2. If the test is successful, the message Successfully performed SearchProducts action as arn:aws:iam::AWS Account:role/SnowEndUser is returned.
3. An unsuccessful test returns the message Error using account…
4. Given the preceding setup, Abel Tuter can now order products from AWS Service Catalog in ServiceNow.

# Validate Configurations

You are now ready to validate the AWS Service Catalog Connector for ServiceNow installation procedures.

**To validate the configuration of the Connector**

1. Log into your ServiceNow instance as the end user (for example, Abel Tuter).
2. Type **Service Catalog** in the navigation filter and choose **Service Catalog**.
3. The user interface view displays the AWS Service Catalog category.

**To order a product**

1. Select the **AWS Service Catalog S3 Storage** product to provision.
2. Fill in the product request details including product name, parameters, and tags.
3. Choose **Order Now** to submit the ServiceNow request and provision the AWS Service Catalog product.
4. After approximately one minute, you receive an order status indicating that your request was submitted.

**To view provisioned products**

1. Go to **My Assets** in the ServiceNow standard user interface.
2. In **My Asset Requests**, view the requests that have been made.
3. To view the product, personalize the list view to show the associated configuration item by choosing the Settings icon in the header row of the table of asset requests.
4. Select **Configuration item (configuration\_item)** and add it to the view  with the **>** button. Move it to below **Stage** in the list.
5. The configuration item (the product that was ordered) shows in the list of assets.
6. To view the product, choose the configuration item name.
7. View the **Outputs** for the provisioned product in the **Outputs** tab of the form.
8. View the provisioning history of the product in the **Product Events** tab of the form.

# ServiceNow Additional Administrator Features

This section provides information about additional administrator features for the AWS Service Catalog Connector for ServiceNow.

## Deleting AWS Service Catalog Products

The Connector for ServiceNow version 1.6.7 enables ServiceNow administrators with**x\_126749\_aws\_sc\_account\_admin** permission the ability to delete AWS Service Catalog products that do not have self-service actions associated.

**Note**

For the Connector for ServiceNow version 1.6.7, you must disassociate self-service actions from AWS Service Catalog products within the AWS Management Console before managing products with the ServiceNow platform.

**To delete AWS Service Catalog products**

1. In the Connector, go to **AWS Service Catalog - Products**. Choose the AWS Service Catalog product to delete.
2. Choose **Manage Product**.
3. Choose **Delete Product**.
4. A warning appears. Choose **OK**.
5. After the deletion is complete, a message appears telling you the product has been deleted.

## Ordering AWS Service Catalog Products Through the ServiceNow Service Portal

The Connector for ServiceNow version 1.6.7 supports ordering AWS Service Catalog products through Service Portal by using the Service Catalog and Order Something views. The release also includes pages and widgets that you can add to Service Portal that enable users to view their provisioned products.

**Note**

The audience for the Service Portal Features section is a ServiceNow administrator or equivalent. The ServiceNow user requires permissions to modify the Service Portal.

### Service Portal Widgets

The Connector for ServiceNow version 1.6.7 includes new widgets that you can add to your Service Portal. This version also includes two alternative view Portal Pages for the following:

* **My AWS Products** – Provides an overview of all provisioned products owned by the user.
* **AWS Product Details** – Provides details of a single provisioned product.

To access the new widgets, you need to update the Service Portal Designer.

**To update the Service Portal Designer**

1. Go to [Create and edit a page using the Service Portal Designer](https://docs.servicenow.com/bundle/kingston-servicenow-platform/page/build/service-portal/task/t_ConfigureAPage.html).
2. Following the instructions, choose the **Service Portal Index** page.
3. Under the **Order Something** container, add the **My AWS Products** widget.
4. The new widget appears on your main Service Portal view.

### Service Portal Pages

The following section describes the two new pages available in the Service Portal Beta release of the AWS Service Catalog Connector, **My AWS Products** and **AWS Product Details**. You can add links to these pages on the Service Portal home page or other pages by using the usual page configuration mechanism in Service Portal.

**My AWS Products**

Provides an overview of all provisioned products owned by the user. Terminated products are displayed separately from current products in a panel that is collapsed on initial page load.

The **My AWS Products** page is available using the following format:

http://<insertinstancename>.service-now.com/sp?id=aws\_sc\_pp

**AWS Product Details**

Provides details of a single provisioned product.

The **AWS Product Details** page is available using the following format:

http://<insertinstancename>.service-now.com/sp?id=aws\_sc\_pp\_details&sys\_id=<provisioned product id>

# Upgrade Instructions

This section provides steps for upgrading from an earlier version of the AWS Service Catalog Connector for ServiceNow.

**To upgrade to the latest version of the Connector**

1. Clear the ServiceNow platform cache by typing in the following URL:https://[InsertServiceNowInstanceNameHere]/cache.do

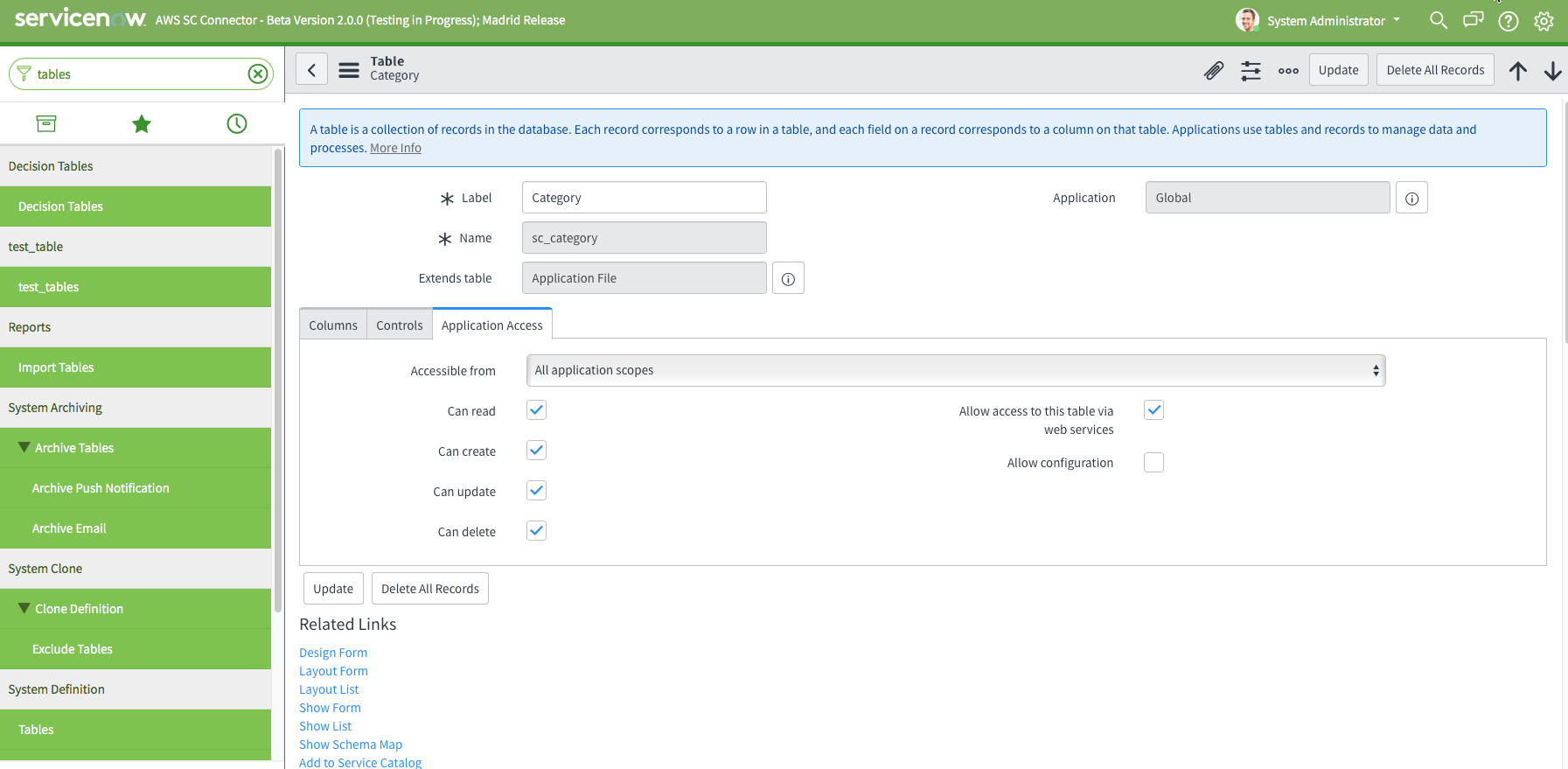
**Note**

Make sure you are installing the update set in a non-production/sandbox environment. Consult a ServiceNow system administrator if you need approval to clear the ServiceNow platform cache.

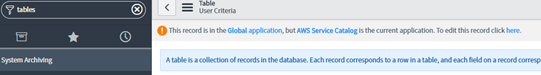
1. Clear the Web Browser cache
2. If you are upgrading from an earlier version, the permissions on ServiceNow Platform tables (User Criteria and Catalog Variable Set) are no longer needed for the Connector for ServiceNow.
3. Enable permissions on ServiceNow Platform tables (Category and Catalog Item Category).

### For AWS products to display under AWS portfolios as sub-categories in the ServiceNow Service Catalog, you need to modify the Application Access form for Category and Catalog Item Category tables.

1. Enter "Tables" in the Navigator and select System Definition -> Tables
2. In the list of tables search for a table with Label "Category" (or Name "sc\_category"). The list of tables will be displayed. Choose Category to view the form defining the table.
3. Choose the "Application Access" tab on the form and choose the "Can Create", “Can Update, and "Can delete" checkboxes on the form. Select the "Update" button.



Note: You may need to click on the link at the top to edit the record.



1. Repeat the steps used on the Category table above for the "Catalog Item Category" table (type **sc\_cat\_item\_category** in the “Go to Name Search” field).
2. From your ServiceNow dashboard, type plugins in the navigation panel in the upper left.
3. When the **System Plugins** page populates, next to the dropdown that says **Name**, search for **user criteria**.
4. Choose **User Criteria Scoped API** and then choose **Activate**.
5. Download the **Connector for ServiceNow update set** from the [ServiceNow store](https://store.servicenow.com/sn_appstore_store.do#!/store/application/f0b117a3db32320093a7d7a0cf961912/). For users installing the update set on a ServiceNow Personal Developer Instance (PDI), download the code from [Connector for ServiceNow version 2.0.2 update set.](https://s3.amazonaws.com/servicecatalogconnector/AWS_SC_update_set_2.0.2.xml.gz)

The Connector for ServiceNow version 2.0.0 update set may be applied to a “Kingston,” “London” or “Madrid” platform release of ServiceNow.

1. From your ServiceNow dashboard, type update sets in the navigation panel in the upper left.
2. Choose **Retrieved Update Sets** from the results.
3. Select **Import Update Set from XML** and upload the release XML file.
4. Select the **AWS Service Catalog Connector for ServiceNow** update set.
5. Choose **Preview Update Set**, which makes ServiceNow validate the connector update set.
6. Choose **Update**.
7. Choose **Commit Update Set** to apply the update set and create the application. This procedure should complete 100%.

**To update permissions**

1. Go to [Creating IAM Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html). Following the instructions there, create a policy called **SCConnectorAdmin** for ServiceNow administrators to delete AWS Service Catalog products in ServiceNow that do not have self-service actions associated. Copy the following policy and paste it into **Policy Document**:

{

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

"Statement": [

{

"Sid": "VisualEditor0",

"Effect": "Allow",

"Action": [

"servicecatalog:DisassociateProductFromPortfolio",

"servicecatalog:DeleteProduct",

"servicecatalog:DeleteConstraint",

"servicecatalog:DeleteProvisionedProductPlan",

"servicecatalog:DeleteProvisioningArtifact"

],

"Resource": "\*"

}

]

}

**Note**

The **ServiceCatalogServiceNowAdditionalPermissions** AWS policy is no longer needed for the Connector for ServiceNow.

1. Create a policy called **ServiceCatalogSSMActionsBaseline**. Follow the instructions at [Creating IAM Policies](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_create.html), and paste the following into the JSON editor:

{

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

"Statement": [

{

"Sid": "Stmt1536341175150",

"Action": [

"servicecatalog:ListServiceActionsForProvisioningArtifact",

"servicecatalog:ExecuteprovisionedProductServiceAction",

"ssm:DescribeDocument",

"ssm:GetAutomationExecution",

"ssm:StartAutomationExecution",

"ssm:StopAutomationExecution",

"cloudformation:ListStackResources",

"ec2:DescribeInstanceStatus",

"ec2:StartInstances",

"ec2:StopInstances"

],

"Effect": "Allow",

"Resource": "\*"

}

]

}

1. Update the **AWSCloudFormationFullAccess** policy. Choose **create policy** and then paste the following in the JSON editor:

{

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

"Statement": [

{

"Effect": "Allow",

"Action": [

"cloudformation:DescribeStackResource",

"cloudformation:DescribeStackResources",

"cloudformation:GetTemplate",

"cloudformation:List\*",

"cloudformation:DescribeStackEvents",

"cloudformation:DescribeStacks",

"cloudformation:CreateStack",

"cloudformation:DeleteStack",

"cloudformation:DescribeStackEvents",

"cloudformation:DescribeStacks",

"cloudformation:GetTemplateSummary",

"cloudformation:SetStackPolicy",

"cloudformation:ValidateTemplate",

"cloudformation:UpdateStack",

"cloudformation:CreateChangeSet",

"cloudformation:DescribeChangeSet",

"cloudformation:ExecuteChangeSet",

"cloudformation:DeleteChangeSet",

"s3:GetObject"

],

"Resource": "\*"

}

]

}

Note: **AWSCloudFormationFullAccess** now includes additional permissions for ChangeSets.

1. Attach the **ServiceCatalogSSMActionsBaseline and AWSCloudFormationFullAccess** IAM policies to the **SCConnectLaunch** role, which were created during the Baseline Permissions setup.

**To add a change request type**

1. When upgrading from a previous version of the AWS Service Catalog scoped app, you must remove the **AWS Product Termination** change request type before creating a new change request type.
2. You also must add a new change request type called **AWS Provisioned Product Event** for the scoped application to trigger an automated change request in Change Management. For instructions, see [Add a new change request type](https://docs.servicenow.com/bundle/madrid-it-service-management/page/product/change-management/task/t_AddNewChangeType.html).
   1. Open an existing change request.
   2. Open the context (right-click) menu for **Type** and then choose **Show Choice List**.
   3. Choose **New** and fill in the following fields:
      1. **Table**: **Change Request**
      2. **Label**: **AWS Provisioned Product Event**
      3. **Value**: **AWSProvisionedProductEvent**
      4. **Sequence**: pick the next unused value
   4. Submit the form.