

AWS Service Catalog Connector for ServiceNow

(Installation Instructions)

This document provides instructions to integrating the AWS Service Catalog (“SC”) Connector for ServiceNow. The Connector for ServiceNow installation has three components:

1. Configure AWS (IAM and AWS Service Catalog)
2. Install & Configure ServiceNow (Scoped Application and Service Catalog)
3. Validate Connector for ServiceNow (AWS Product request, Provisioned Resources, Product Termination)

1. Configure AWS

The AWS Service Catalog (“SC”) Connector for ServiceNow requires baseline AWS users and permissions. For each AWS account, the ServiceNow Connector requires two IAM users and roles:

- **SC Sync User** – IAM user to Sync AWS portfolios and products to ServiceNow catalog items ([ServiceCatalogAdminReadOnly managed policy](#))
- **SC End User role** - IAM role configured as an AWS Service Catalog end user and assigned to each Service Catalog portfolio
- **SC End User** - Enables ServiceNow connector 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 Service Catalog.
- **SCConnect Launch role** - IAM role used to place baseline AWS service permissions into the 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 ec2 and S3 services. If your products contain more AWS services, you will need to either include those services to the SCConnectLaunch role or create new launch role(s).

I. Create SC Sync User

The following section details the steps to create the SC Sync User and associate the appropriate IAM permission. To perform this task, you need AWS IAM permissions to create new users.

- a. Create a sync user (for example, SCSyncUser) using [Creating an IAM User in Your AWS Account](#) instructions. The user needs programmatic and AWS Management console access to follow the ServiceNow Connector installation instructions.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* SCSyncUser [Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type*

- ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.
- ☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password*

- ☐ Autogenerated password
- ☒ Custom password

- b. Set permissions for your sync user (SCSyncUser). Click “Attach existing policies directly” and select the **ServiceCatalogAdminReadOnlyAccess** policy

Add user

Set permissions for SCSyncUser

[Add user to group](#) [Copy permissions from existing user](#) [Attach existing policies directly](#)

Attach one or more existing policies directly to the users or create a new policy. [Learn more](#)

[Create policy](#) [Refresh](#)

Filter: Policy type Showing 1 result

	Policy name	Type	Attachments	Description
<input type="checkbox"/>	ServiceCatalogAdminReadOnly...	AWS managed	2	Provides read only access to the service catalog admin console.

- c. Review and Create User
- d. Note the Access and Secret Access information. Download the csv file that contains the user credential information.

II. Create SC End User

The following section details the steps to create the SC End User and associate the appropriate IAM permission. To perform this task, you need AWS IAM permissions to create new users.

- a. [Create a Policy](#) called ServiceCatalogServiceNowAdditionalPermissions. Enter the following code into the JSON editor:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1507210800000",
      "Effect": "Allow",
      "Action": [
        "servicecatalog:ListProvisioningArtifacts"
      ]
    }
  ]
}
```

```
    ],  
    "Resource": [  
        "*"   
    ]  
  }  
]  
}
```

- b. [Create a role](#) for the ServiceNow associate end user to assume (for example, SnowEndUser).
1. Add the following permissions (policies) to the role:
 - ServiceCatalogServiceNowAdditionalPermissions (created in the previous step)
 - AWSServiceCatalogEndUserFullAccess
 2. Create a trust relationship on the SnowEndUser role to the account. Place the following text into the Trust Relationship:

```
{  
  "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.

Roles > SnowEndUser

Summary

Delete role

Role ARN	arn:aws:iam::[redacted]:role/SnowEndUser
Role description	Allows Service Catalog to access AWS resources on your behalf. Edit
Instance Profile ARNs	[redacted]
Path	/
Creation time	2018-04-05 17:18 EDT
Maximum CLI/API session duration	1 hour Edit
Give this link to users who can switch roles in the console	https://signin.aws.amazon.com/switchrole?roleName=SnowEndUser&account=[redacted]

Permissions

Trust relationships

Access Advisor

Revoke sessions

Attach policy

Attached policies: 2

Policy name	Policy type	
ServiceCatalogServiceNowAdditionalPermissions	Managed policy	✕
AWSServiceCatalogEndUserFullAccess	AWS managed policy	✕

- c. [Create a Policy](#) called **StsAssume-SC**. Place the following text in the JSON

```
{
  "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.

Policies > StsAssume-SC

Summary

Delete policy

Policy ARN arn:aws:iam::[redacted]:policy/StsAssume-SC

Description

Permissions

Attached entities (1)

Policy versions

Access Advisor

Policy summary

{ } JSON

Edit policy

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Sid": "VisualEditor0",
6-       "Effect": "Allow",
7-       "Action": "sts:AssumeRole",
8-       "Resource": "arn:aws:iam::[redacted]:role/SnowEndUser"
9-     }
10-   ]
11- }
```

- d. Create a user (for example, SCEndUser) using [Creating an IAM User in Your AWS Account](#) instructions. The user needs programmatic and AWS Management console access to follow the Connector for ServiceNow installation instructions.

Add user

Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

User name* SCEndUser

[Add another user](#)

Select AWS access type

Select how these users will access AWS. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

Access type* ☒ **Programmatic access**
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

☒ **AWS Management Console access**
Enables a **password** that allows users to sign-in to the AWS Management Console.

Console password* ☐ Autogenerated password ☒ Custom password

Attach the assume policy (StsAssume-SC) to your end user (SCEndUser). Click “Attach existing policies directly” and select the **StsAssume-SC**:

[Users](#) > SCEndUser

Summary

User ARN arn:aws:iam:::user/SCEndUser

Path /

Creation time 2018-03-15 03:02 EDT

Permissions **Groups (0)** **Security credentials** **Access Advisor**

[Add permissions](#) **Attached policies: 1**

Policy name	Policy type
Attached directly	
StsAssume-SC	Managed policy

[Add inline policy](#)

- e. Review and Create User
- f. Note the Access and Secret Access information. Download the csv file that contains the user credential information.

III. Create SCConnectLaunch role

The following section details the steps to create the SCConnectLaunch role. This role is used to place baseline AWS service permissions into the Service Catalog launch constraints. Click [Launch Constraints](#) for more information.

a. Create the **AWSCloudFormationFullAccess** Policy

Click create policy-> and paste the below 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",
        "s3:GetObject"
      ],
      "Resource": "*"
    }
  ]
}
```

Policies > AWSCloudFormationFullAccess

Summary

Delete policy

Policy ARN `arn:aws:iam::[redacted]:policy/AWSCloudFormationFullAccess`

Description

Permissions Attached entities (5) Policy versions Access Advisor

Policy summary {} JSON Edit policy ?

```

"cloudformation:DescribeStacks",
"cloudformation:DescribeStackEvents",
"cloudformation:DescribeStackResource",
"cloudformation:DescribeStackResources",
"cloudformation:GetTemplate",
"cloudformation:List*",
"cloudformation:CreateStack",
"cloudformation>DeleteStack",
"cloudformation:DescribeStackEvents",
"cloudformation:DescribeStacks",
"cloudformation:GetTemplateSummary",
"cloudformation:SetStackPolicy",
"cloudformation:ValidateTemplate",
"cloudformation:UpdateStack"

```

- b. Create the **SCConnectLaunch** role. Assign the trust relationship to AWS Service Catalog. 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)

The **SCConnectLaunch** role step is completed when you have the baseline policies as shown below:

Roles > SCConnectLaunch

Summary

Delete role

Role ARN `arn:aws:iam::[redacted]:role/SCConnectLaunch`

Role description Allows Service Catalog to access AWS resources on your behalf. | Edit

Instance Profile ARNs

Path /

Creation time 2018-04-05 17:04 EDT

Maximum CLI/API session duration 1 hour Edit

Permissions Trust relationships Access Advisor Revoke sessions

Attach policy Attached policies: 3

Policy name	Policy type	
AmazonEC2FullAccess	AWS managed policy	✕
AmazonS3FullAccess	AWS managed policy	✕
AWSCloudFormationFullAccess	Managed policy	✕

Add inline policy

Roles > SCConnectLaunch

Summary

[Delete role](#)

Role ARN	arn:aws:iam::[redacted]:role/SCConnectLaunch Copy
Role description	Allows Service Catalog to access AWS resources on your behalf. Edit
Instance Profile ARNs	Copy
Path	/
Creation time	2018-04-05 17:04 EDT
Maximum CLI/API session duration	1 hour Edit

[Permissions](#)
[Trust relationships](#)
[Access Advisor](#)
[Revoke sessions](#)

You can view the trusted entities that can assume the role and the access conditions for the role. [Show policy document](#)

[Edit trust relationship](#)

Trusted entities

The following trusted entities can assume this role.

Trusted entities
The identity provider(s) servicecatalog.amazonaws.com

Conditions

The following conditions define how and when trusted entities can assume the role.

There are no conditions associated with this role.

IV. Create AWS Service Catalog Products

In this section you will configure Service Catalog to have a portfolio that includes an Amazon S3 bucket product. Please use the following Amazon S3 template link here: [Creating an Amazon S3 Bucket for Website Hosting](#) for your preliminary product.

1. Creating a Service Catalog Portfolio

- AWS console->Service Catalog->Create Portfolio. Once a portfolio is created, add products to that portfolio

2. Creating an Service Catalog Product

- Upload new product-> Enter details ->Under Select template, choose S3 bucket CloudFormation template
 - Add a Launch Constraint for the product that you just created with the “**SCConnectLaunch**” Role assigned. See here for additional instructions: <https://docs.aws.amazon.com/servicecatalog/latest/adminguide/constraints-launch.html#constraints-launch-constraint>
 - Ensure you add the “SnowEndUser” IAM role to the portfolio. Click here for additional instructions: https://docs.aws.amazon.com/servicecatalog/latest/adminguide/catalogs_portfolios_users.html

After setting up AWS IAM, and AWS Service Catalog configurations, the the AWS setup for the integration is complete. Please validate all instructions above are completed prior to moving forward.

S3Storage01

Created from

AWS Service Catalog

Created

Oct 10th 2017 10:47:21 UTC-0400

Vendor

aws

Provided by

aws

Description

S3 Storage

▼ Versions (1)

CREATE NEW VERSION

ACTIONS

By name

Showing 1 version

Versions	Created time	Type	Description
<input type="radio"/> storage1	Oct 10th 2017 10:47:21 UTC-...	CLOUD_FORMATION_TEMPLATE	S3 Storage

▶ Portfolios

▶ Tags (1)

2. Install and Configure ServiceNow

Now that you completed the AWS IAM and AWS Service Catalog configurations, the next component area to setup is ServiceNow. High-level installation tasks within ServiceNow include:

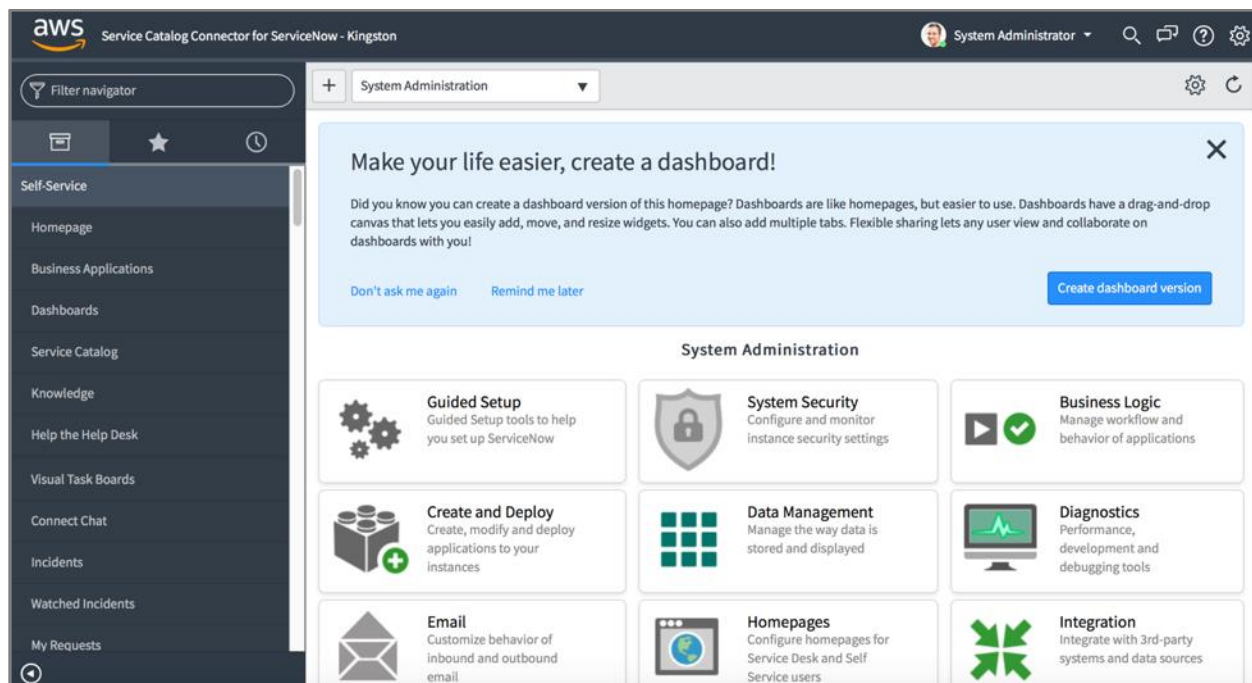
- Upload and Commit ServiceNow Connector “Update Set”
- Configure ServiceNow Platform System Admin Components
- Configure ServiceNow Connector Scoped App – AWS Service Catalog
 - Accounts (Based on two AWS users/account in phase I)
 - Sync via Scheduled Jobs (AWS and ServiceNow)
 - Identities (link the AWS End User to a ServiceNow role permission)
 - Associate ServiceNow role to ServiceNow End User/Group

I. Installing ServiceNow Connector Scoped Application

The ServiceNow Connector product is released as a conventional ServiceNow scoped application via an [Update Set](#). ServiceNow update sets are code changes to out of the box platform and enable developers to move code across ServiceNow instance environments. The AWS Service Catalog Connector for ServiceNow update set is available to download in the [ServiceNow store](#).

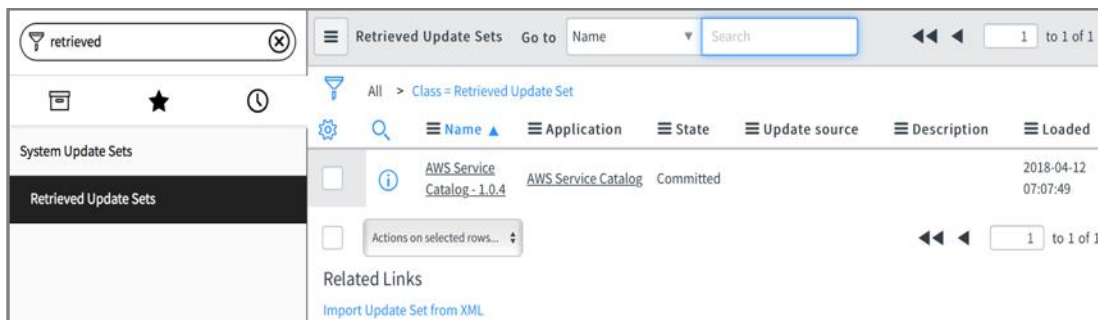
The update set may be applied to a "Helsinki", "Istanbul", "Jakarta" or "Kingston" platform release of ServiceNow.

1. Download the AWS SC Connector for ServiceNow scoped app from the ServiceNow store.
2. Log into your ServiceNow instance specified in the ServiceNow store. You need to login as the system administrator (user with the ServiceNow **admin** role associated). You will see the ServiceNow standard user interface view as shown below



a. Install the Update Set

1. In the navigator panel type "Update Sets" and select "Retrieved Update Sets" from the results.
2. Select "Import Update Set from XML" on the page and upload the release XML file.



3. Select the "AWS Service Catalog" update set.
4. Click Preview Update Set, which will make ServiceNow validate the set.
5. Click Update and the ServiceNow form will update to the following view:

Retrieved Update Set
AWS Service Catalog

Name: AWS Service Catalog

Application: AWS Service Catalog

Update source:

Parent:

State: Previewed

Loaded: 2018-02-12 09:58:52

Description:

Application name: AWS Service Catalog

Committed:

Inserted: 432

Updated: 0

Deleted: 0

Collisions: 0

Total: 432

Update Delete Run Preview Again Commit Update Set

- Click Commit Update Set to apply the update set and create the application.
The Update Set Commit procedure should complete 100%.

II. Configure ServiceNow Platform System Admin Components

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

a. Enable permissions on ServiceNow Platform tables

- Enter "Tables" in the Navigator and select System Definition -> Tables
- In the list of tables search for a table with Label "User Criteria" (or name "user_criteria"). The list of tables will be displayed, with the User Criteria table at the top. Select it by clicking on its label, to view the form defining the table.
- Select the "Application Access" tab on the form and select the "Can Create" and "Can delete" checkboxes on the form. Click the "Update" button.

aws service catalog ServiceNow Connector - Kingston

System Administrator

tables

Table
User Criteria

A table is a collection of records in the database. Each record corresponds to a row in a table, and each field on a record corresponds to a column on that table. Applications use tables and records to manage data and processes. [More Info](#)

* Label: User Criteria Application: Global

* Name: user_criteria

Extends table: Application File

Columns Controls Application Access

Accessible from: All application scopes

Can read ☒ Allow access to this table via web services ☒

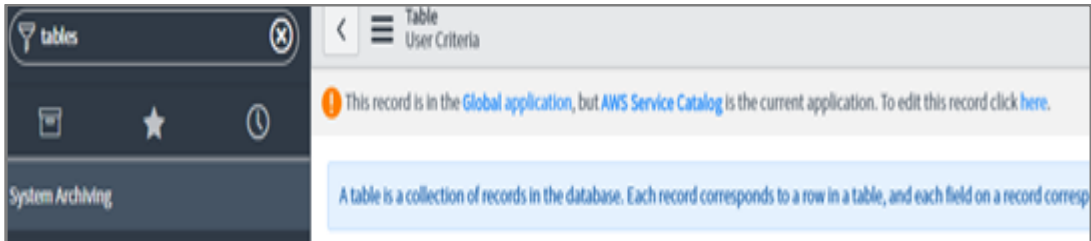
Can create ☒

Can update ☐ Allow configuration ☐

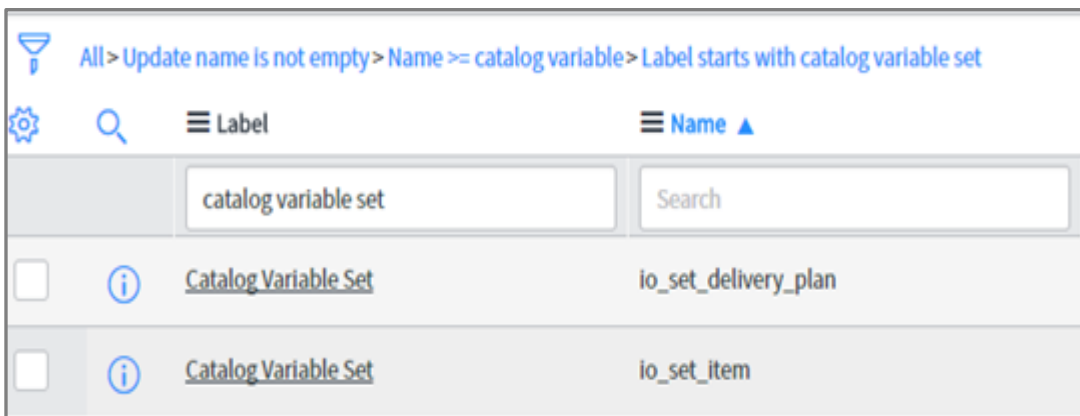
Can delete ☒

Update Delete All Records

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



4. Repeat the steps used on the User_Criteria table above for the "Catalog Variable Set" table (type **io_set_item** in the "Go to Name Search" field).
 - a. Note: you may see two tables in the search result. Choose the io_set_item table



b. Set up application administrator privileges

The AWS Service Catalog scoped application comes with two ServiceNow roles that enable access to configure the application, so that system admins can grant one or more users privileges to administer the application without having to open up full sysadmin access to them. The two roles are:

Role	What does it enable
x_126749_aws_sc_account_admin	Permits configuring the AWS accounts used to synchronize with AWS Service Catalog and to provision Products on behalf of end users
x_126749_aws_sc_portfolio_manager	Permits configuring which end users may view and order what Products from AWS Service Catalog

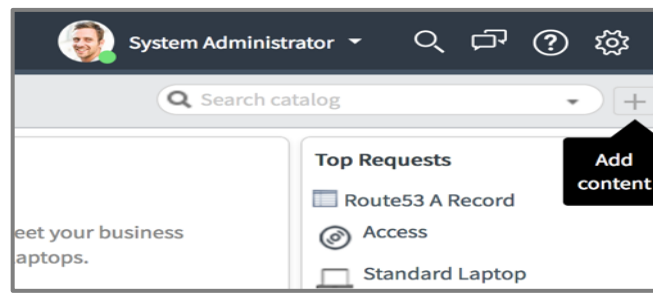
These roles can be assigned either to individual users or both to one administrator user. As the sysadmin user:

1. Type "Users" in the navigator and select "System Security - Users"

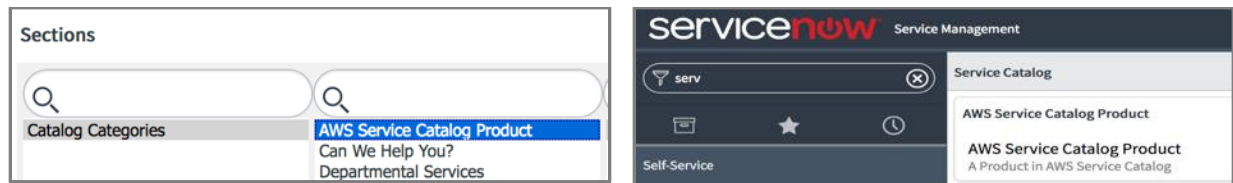
2. Select a user to grant one or both roles above to (for example, admin). You can also [Create a User](#)
3. Click "Edit" on the Roles tab of the form
4. Filter the Collection of roles by the prefix "x_"
5. Choose from x_126749_aws_sc_account_admin and/or x_126749_aws_sc_portfolio_manager and add them to the user.
6. Click Save

c. Add the AWS Service Catalog to the Service Catalog categories

1. Navigate to “Self Service” | Service Catalog” and select the “Add content” icon (top right):



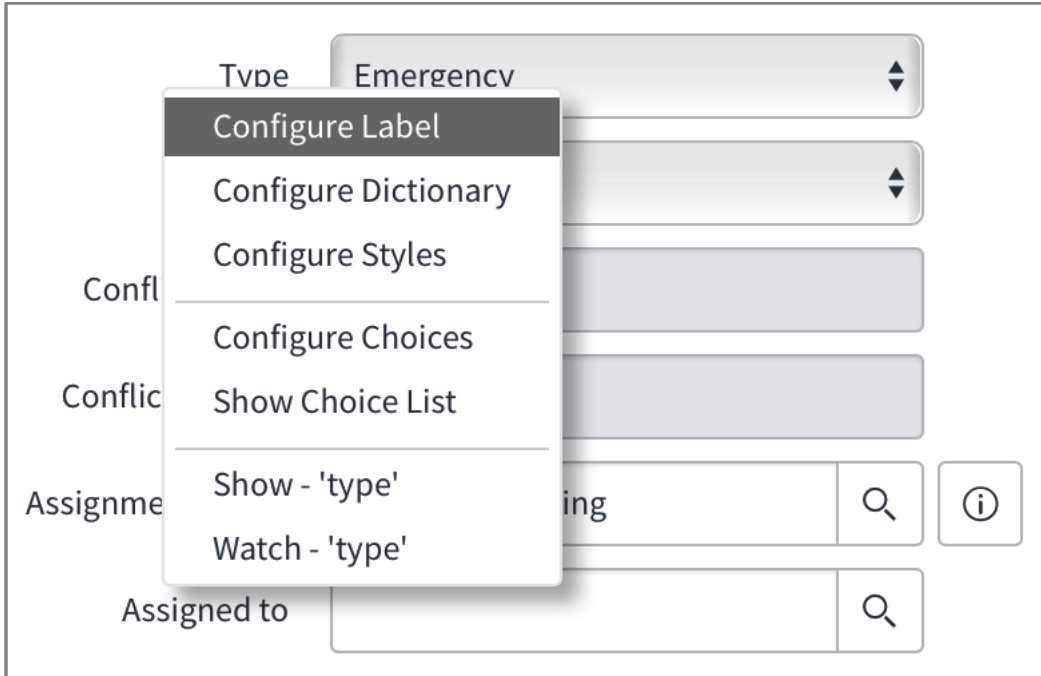
2. Select the “AWS Service Catalog Product” entry; add it to your catalog home page by clicking the first "Add Here" link on the second row of the selection panel at the bottom of the dialog.



d. Add a Change Request Type

You need to [add a new change request type](#) called **AWS Product Termination** for the scoped application to trigger an automated change request in Change Management.

1. Open an existing change request.
2. Right-click on Type and select Show Choice List.



Note: Some Mac users may experience trouble accessing Choice List without a mouse device.

3. Click New and fill in the following fields.
 - a. Table - Change Request
 - b. Label - AWS Product Termination
 - c. Value - AwsProductTermination
 - d. Sequence - pick the next unused value
4. Submit the form.

The screenshot shows the 'Choice' configuration form in the AWS Service Catalog console. The form is titled 'Choice New record'. It contains the following fields:

- Table**: -- None -- (dropdown)
- Element**: type (text input)
- Language**: en (text input)
- Label**: AWS Product Termination (text input)
- Value**: AwsProductTermination (text input, highlighted with a blue border)
- Sequence**: 4 (text input)
- Inactive**: ☐ (checkbox)
- Dependent value**: (text input)
- Hint**: (text input)

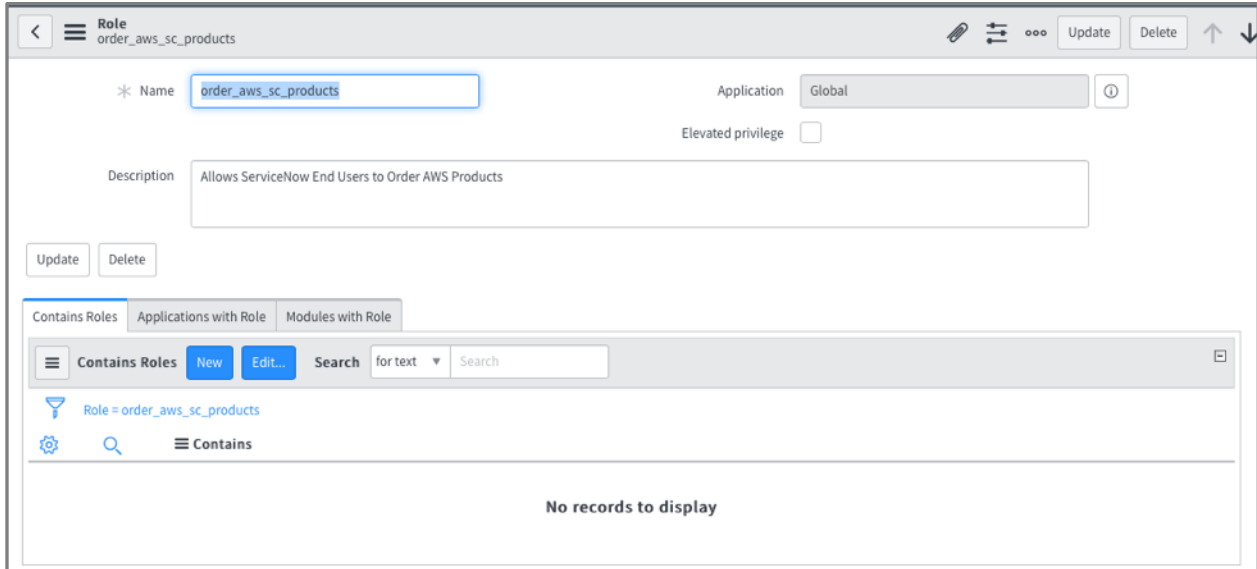
 There are 'Submit' buttons at the bottom left and top right of the form.

III. Configuring ServiceNow Connector Scoped Application – AWS Service Catalog

Having installed and configured the ServiceNow Connector in the previous lab instructions, you need to configure the AWS Scoped Application and applicable roles:

- a. [Create a role](#) called “order_aws_sc_products”.

This role is granted to any user(s) with permission to order AWS Service Catalog products:



The screenshot shows the ServiceNow 'Role' configuration page for 'order_aws_sc_products'. The 'Name' field is 'order_aws_sc_products', the 'Application' is 'Global', and the 'Description' is 'Allows ServiceNow End Users to Order AWS Products'. The 'Elevated privilege' checkbox is unchecked. Below the form, there are tabs for 'Contains Roles', 'Applications with Role', and 'Modules with Role'. The 'Contains Roles' tab is active, showing a search bar and a list of roles. The list is currently empty, displaying 'No records to display'.

- b. Grant roles to the following users:

- i. **System Administrator (admin):** For simplicity in this worked example, user **admin** is designated as the administrator of the AWS Service Catalog – ServiceNow Connector. He is granted (Roles->Edit) both of the administrative permissions from the adapter, *x_126749_aws_sc_portfolio_manager* and *x_126749_aws_sc_account_admin* (rather than assigning these to separate users).

User: System Administrator

Last name: Administrator
Title: System Administrator
Department: Finance
Password:
Password needs reset: ☐
Locked out: ☐
Active: ☒
Web service access only: ☐
Internal Integration User: ☐
Update

Calendar integration: Outlook
Time zone: System (America/Los_Angeles)
Date format: System (yyyy-MM-dd)
Business phone:
Mobile phone:
Photo: Click to add...

Related Links
[View Subscriptions](#)
[Reset a password](#)
[Change password](#)

Roles (6) Groups (1) Delegates Subscriptions Manage Subscriptions

Roles Edit... Go to Role Search

User = System Administrator

Role	State	Inherited	Inheritance Count
x_126749_aws_sc_portfolio_manager	Active	false	
admin	Active	false	
x_126749_aws_sc_account_admin	Active	false	

- ii. **Abel Tuter:** The user **abel.tuter** is chosen as an illustrative end user. Abel requires the new role `order_aws_sc_products` to order products from AWS.

User: Abel Tuter

User ID: abel.tuter
First name: Abel
Last name: Tuter
Title:
Department: Product Management
Password:
Password needs reset: ☐
Locked out: ☐
Active: ☒
Web service access only: ☐
Internal Integration User: ☐
Update Delete

Email: abel.tuter@example.com
Language: -- None --
Calendar integration: Outlook
Time zone: System (America/Los_Angeles)
Date format: System (yyyy-MM-dd)
Business phone:
Mobile phone:
Photo: Click to add...

Related Links
[View Subscriptions](#)
[Reset a password](#)

Roles (1) Groups Delegates Subscriptions Manage Subscriptions

Roles Edit... Go to Role Search

User = Abel Tuter

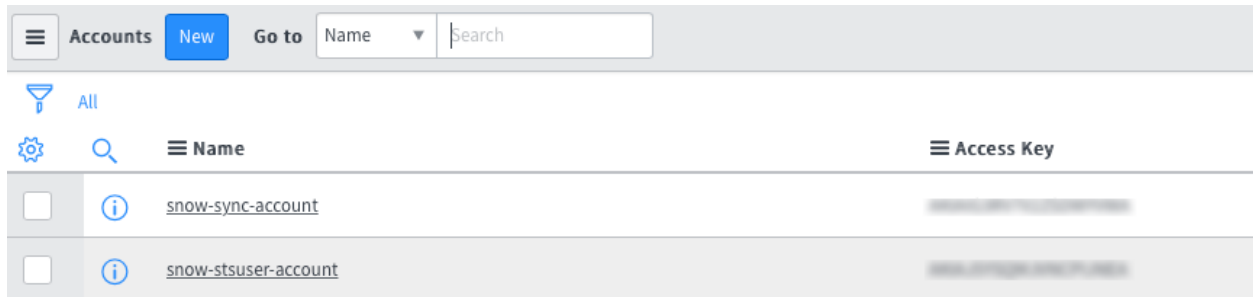
Role	State	Inherited	Inheritance Count
order_aws_sc_products	Active	false	

Actions on selected rows...

IV. Configure Accounts

Log in as the System Administrator (admin). In the AWS Service Catalog 'Accounts' menu, create two accounts, one for sync and one for provisioning. Note, the names 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)

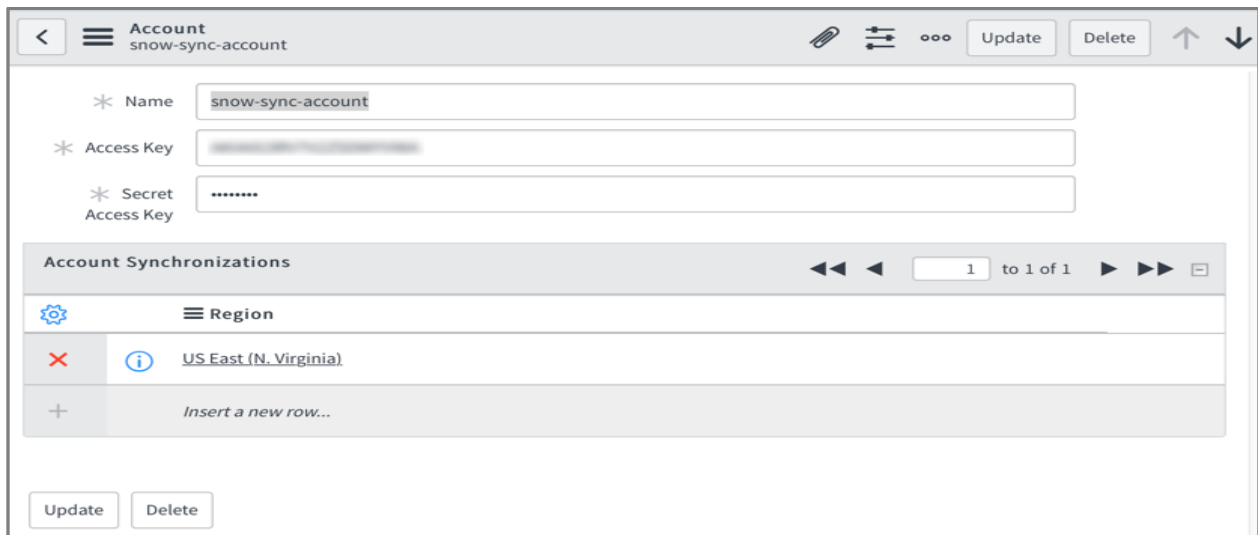


	Name	Access Key
<input type="checkbox"/>	snow-sync-account	AKIAIOSFODNN7EXAMPLE
<input type="checkbox"/>	snow-stsuser-account	AKIAIOSFODNN7EXAMPLE

The **snow-stsuser-account** account has no Regions configured. The **snow-sync-account** user has one Region configured, matching the Region where the AWS Service Catalog is defined (double click the text " Insert a new row " to add):

You will need to use the keys and secret keys from the users you created in AWS.

ServiceNow-AWS Correlations	
ServiceNow Account	AWS User
snow-sync-account	SCSyncUser
snow-stsuser-account	SCEndUser



Account snow-sync-account

Name: snow-sync-account

Access Key: [Redacted]

Secret Access Key: [Redacted]

Account Synchronizations

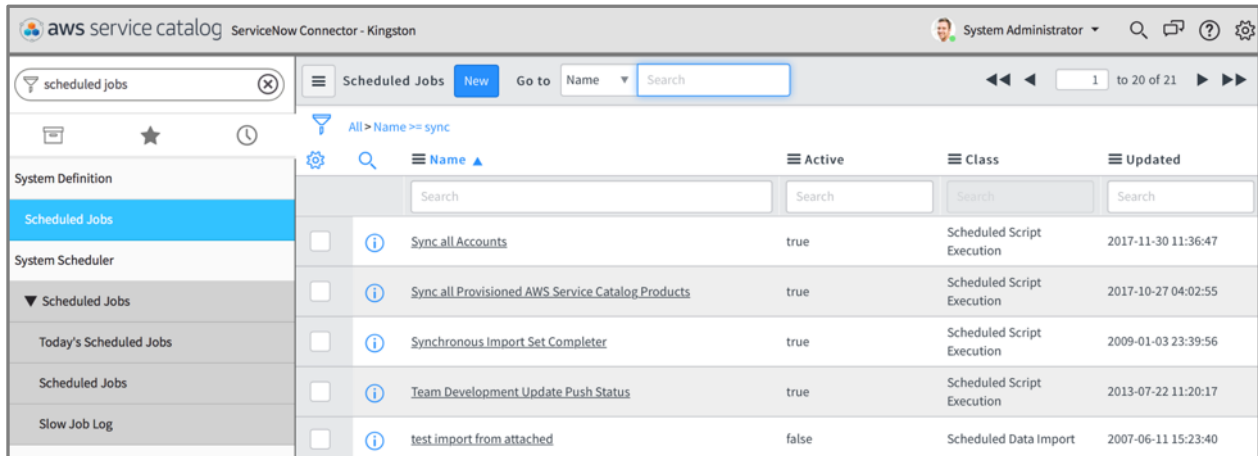
Region
US East (N. Virginia)
Insert a new row...

Update Delete

V. Scheduled Jobs (Initial Manual Sync)

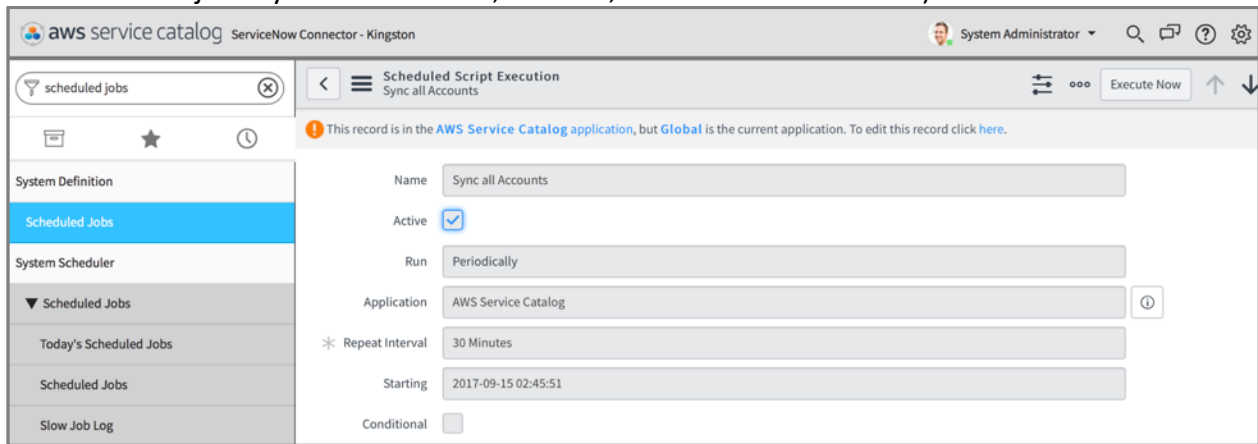
During the initial setup, manually execute the sync as oppose to waiting for the Scheduled Jobs to occur. To sync the accounts manually

- Log in as System Administrator
- Find "Scheduled Jobs" in the filter navigator panel



Name	Active	Class	Updated
Sync all Accounts	true	Scheduled Script Execution	2017-11-30 11:36:47
Sync all Provisioned AWS Service Catalog Products	true	Scheduled Script Execution	2017-10-27 04:02:55
Synchronous Import Set Completer	true	Scheduled Script Execution	2009-01-03 23:39:56
Team Development Update Push Status	true	Scheduled Script Execution	2013-07-22 11:20:17
test import from attached	false	Scheduled Data Import	2007-06-11 15:23:40

- Search for job "Sync all Accounts", select it, and click Execute Now.)



Scheduled Script Execution
Sync all Accounts

This record is in the [AWS Service Catalog](#) application, but [Global](#) is the current application. To edit this record click [here](#).

Name:

Active: ☒

Run:

Application:

* Repeat Interval:

Starting:

Conditional: ☐

Execute Now

Note: If you do not see Execute Now in the upper left-hand corner. Please click on Configure Job Definition. Execute Now will be visible.

VI. Grant Access to Portfolios

Data will be visible in the AWS Service Catalog menus once the adapter's scheduled synchronization job has run. In the Identities screen, select the **SnowEndUser** role and assign it to Account snow-stsuser-account (double click the cell in the Account column, or click the SCEndUser user name and edit the form presented):

The screenshot shows the AWS Identity console for the role `arn:aws:iam:::role/SnowEndUser`. The ARN field is populated with `arn:aws:iam:::role/SnowEndUser`. The Account dropdown is set to `snow-stsuser-account`. There are 'Update' and 'Delete' buttons at the bottom.

In the Role Grants menu, click New and enter the Role of 'order_aws_sc_products' and the **SnowEndUser** identity. The Role Grants table will now look like:

Role Grants		New	Go to	Role	Search	1 to 1 of 1
All						
Role	Identity					
<input type="checkbox"/>	order_aws_sc_products			arn:aws:iam:::role/SnowEndUser		

Given the set up above, Abel Tuter can now order Products from the AWS Service Catalog.

When Abel orders products he can view them by logging into ServiceNow and use the validation steps below.

The screenshot shows the AWS Service Catalog console. The breadcrumb is `Service Catalog > AWS Service Catalog`. The page title is **AWS Service Catalog** with the subtitle `Order products from the AWS Service Catalog.` Under the **Items** section, there is a card for `S3Storage01` under the category `S3 Storage`. A 'preview' button is visible, and a search bar contains the text `S3 Storage`. A '20 per page' dropdown and a 'Search catalog' search bar are at the top right.

3. Validate ServiceNow Connector

I. Ordering a Product

1. Browse ServiceNow's Service Catalog for AWS Service Catalog products.
2. Select a product to order.
3. On the form for the product, if necessary, select a role with which you are ordering this product ("I am ordering this in my capacity as ..."), and a launch option (Portfolio) and product version.
4. Enter the provisioning parameters for the Product. These will vary depending on the Product being ordered.
5. Optionally add tags to be assigned to the Product.
6. Click "Order now"
7. A workflow will be created to order the product. Approval may be required.
8. Once the product is provisioned by AWS Service Catalog, a short time will be required for a periodic synchronization job to update the status of the product on the form (up to one minute).

Service Catalog > AWS Service Catalog > WebServer

webserver

webserver

Product Name

Name

Choose a Role

Your role affects which launch options you may use.

☒ order_aws_sc_products

Launch option

☒ awsSnow

Rules:

- Launch as arn:aws:iam::700529603272:role/SCConnectLaunch

Product Version

☒ lampstack3

description

Order this Item

Delivery time 1 Day

[Order Now](#)

[Add to Cart](#)

Shopping Cart

Empty

[Order Status](#) [Back to Catalog](#) [Continue Shopping](#) [Home](#)


Thank you, your request has been submitted

Order Placed: 2018-04-20 18:45:45
Request Number: [REQ0010006](#) ☆
Estimated Delivery Date of Complete Order: 2018-04-21

Description	Delivery Date	Stage	Price (ea.)	Quantity	Total
S3 Storage	2018-04-21	▶		-	
Total					-

[Back to Catalog](#) [Continue Shopping](#) [Home](#)

II. Viewing Provisioned Products

1. Navigate to "My assets"
2. In the "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:
 - a. click the "Settings" cogwheel  in the header row of the table of asset requests
 - b. select " Configuration item (configuration_item) " and add it to the view by pressing the > button. Move it to below "Item":

Personalize List Columns

Available

Additional comments
Approval
Approval history
Assigned to
Assignment group
Backordered
Billable
Business duration
Business service
Catalog
Close notes
Closed
Closed by
Company
Contact type

>
<

Selected

Number
Item
Stage
Request
Request.Requested for
Request.Opened by
Due date
Quantity
Configuration item(configuration_item)

^
v

☒ Wrap column text ☐ Compact rows ☐ Active row highlighting
☒ Modern cell coloring
☒ Enable list edit ☒ Double click to edit

[Cancel](#) [OK](#)

This will mean that the configuration item (that is, the product that was ordered) shows in the

list of assets. See below example of Storage ordered:

The screenshot shows the AWS Service Catalog console interface. At the top, there's a navigation bar with a back arrow, a menu icon, the product name 'OrderS34Blog [Mycompanyassets view]', and action buttons: 'Dashboard', 'Form' (highlighted), 'Update', and 'Delete'. Below this, the 'Region' is set to 'us-east-1' with 'Update' and 'Delete' buttons. A 'Related Links' section includes a 'Subscribe' link. The main content area has two tabs: 'Events (1)' and 'Outputs (3)'. The 'Events' tab is active, showing a table with columns: Record Type, Status, Record Id, Provisioned Product Id, and Product. A single event is listed with a status of 'Succeeded'. Below the table, there's an 'Actions on selected rows...' dropdown and pagination controls showing '1 to 1 of 1'.

- Click on the configuration item name to view the Product
- View the Outputs for the provisioned Product in the "Outputs" tab of the form.
- View the history of the provisioning of the product in the "Events" tab of the form.

Log into the AWS Management Console, navigate to the Service Catalog console, and choose the provisioned product list. The product details give the status of the product as follows:

The screenshot shows the product details for 'OrderS34Blog'. On the left, there's a blue icon with the text 'Internal App'. To the right, the status is 'Available', and other details like 'Product', 'Version', and 'Provided by' are marked as 'Not Available'. Below this, the 'Events (1)' tab is active, showing a table with columns: Date, Status, Type, and Event message. A single event is listed for 'Apr 20th 2018' with a status of 'Succeeded' and a type of 'PROVISION_PRODUCT'. The event message includes the 'Record ID' and 'Provisioned product ID'.

You can also go to the AWS resources provisioned (in this example, an Amazon S3 bucket) to validate.

Bucket name ↑	S3 Provisioned Product ID	Access ⓘ ↑	Region ↑	Date created ↓
sc-pp-sbmvcbhk5pie-thatbucket-1fwi2clpvkrux		Not public *	US East (N. Virginia)	Apr 20, 2018 9:45:55 PM GMT-0400

III. Terminating Provisioned Products

Click the "Terminate" button on the form when viewing the provisioned product. A Change Request will be made to terminate the product.

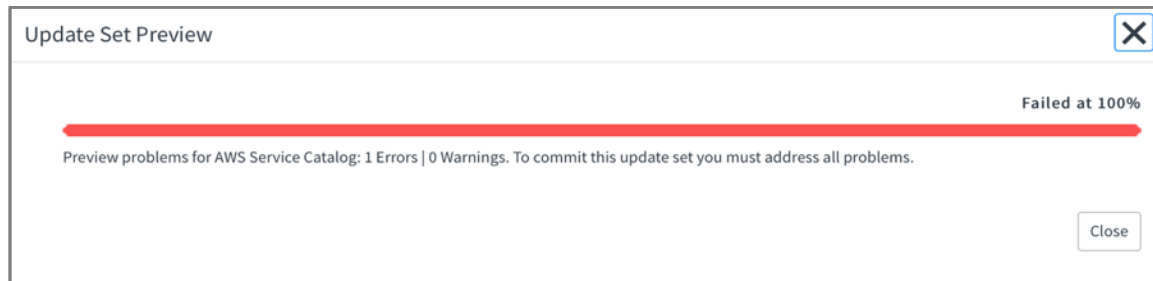
IV. Product Status Updates

The application automatically monitors the status of each provisioned product in AWS and will update the record of the Product in the ServiceNow CMDB within a short time (default 30s) of its status changing in AWS.

V. Troubleshooting

a. Update Set Error Alert

You may receive an error similar to the image below at the end of the preview:



This update set error does not affect the product functionality. Click Close and Scroll down to see the following Error information:

Retrieved Update Set
AWS Service Catalog

Application name: AWS Service Catalog

Update Set Preview Problems (1) | Customer Updates (432) | Child Update Sets

Type	Remote update	Description	Available Actions
Error	sc_category_e7ec31440f1103004dae09bce105...	Could not find a record in sc_homepage_renderer for column homepage_renderer referenced in this update	Find missing record Find missing update Accept remote update Skip remote update

Click Error under the “Type” column. You will then receive the following Update Set Preview Problem screen:

Update Set Preview Problem
Created 2018-03-15 01:48:48

Missing Item: Default

Missing Item Table: Homepage Category Renderer...

Missing Item Update Record:

Available Actions:
 [Find missing record](#)
[Find missing update](#)
[Accept remote update](#)
[Skip remote update](#)

Remote update: sc_category_e7ec31440f1103004dae09bce105...

Update Set: AWS Service Catalog

Type: Error

Status: -- None --

Description: Could not find a record in sc_homepage_renderer for column homepage_renderer referenced in this update

Click “Accept remote update”

The Status of the Update Set Preview Problem changes to “Ignored”

Update Set Preview Problem
Created 2018-03-15 01:48:48

Problem has been ignored

Missing Item:

Missing Item Table:

Missing Item Update Record:

Available Actions: [Find missing record](#), [Find missing update](#), [Skip remote update](#)

Remote update:

Update Set:

Type:

Status:

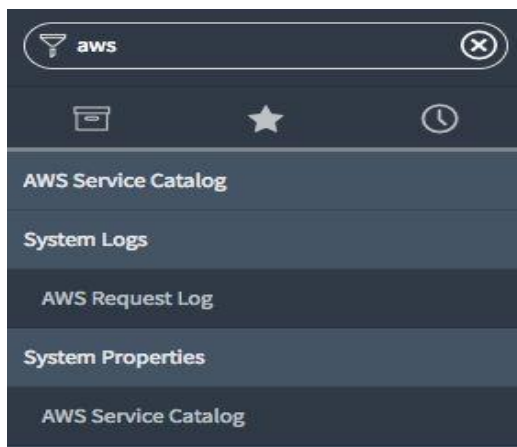
Description: Could not find a record in sc_homepage_renderer for column homepage_renderer referenced in this update

Click Update and proceed to commit the update set.

b. Diagnosing Problems Logging Requests to AWS

All requests the application makes to AWS are logged to `x_126749_aws_sc_req_log`. It is restricted to the `x_126749_aws_sc_account_admin` role. It captures various important facts about each request, in particular the service that was used, the action that was performed, the access key that was used and the role that was assumed, and the status code of the response and the request and response bodies.

Since the application is expected to make a large number of requests to AWS a business rule is configured to drop all records whose response status code is 200.



c. User Assigned a Role but Cannot See Associated Products

The application creates a User Criteria for each product to determine whether it may be viewed by any given user. When a user views a catalog ServiceNow tests each User Criteria and caches the results before displaying the set of items the user is entitled to view. This means that when a user is given a role that allows them to view AWS Service Catalog products that they were not previously entitled to view, they may not be able to view them immediately.

To solve this issue, either wait for the User Criteria cache to expire or empty the database cache immediately by visiting /cache.do as a system administrator. The latter option should be timed to minimize disruption to other users of the ServiceNow platform.

4. Questions on this guide:

For questions on the ServiceNow Connector Installation email aws-sc-servicenow-issues@amazon.com .