AWS Service Catalog
S3 Reference Blueprint

AWS Service Catalog S3 Reference Blueprint

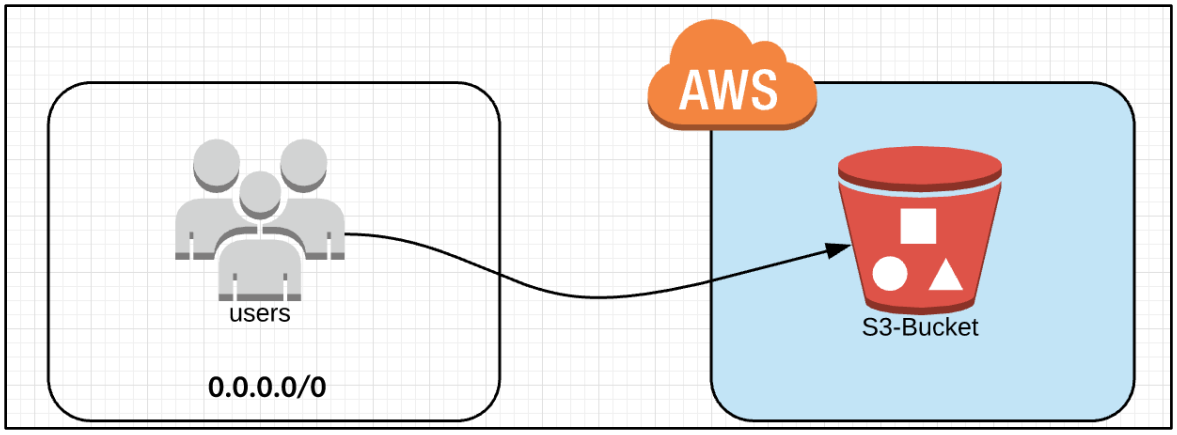
Table of Contents

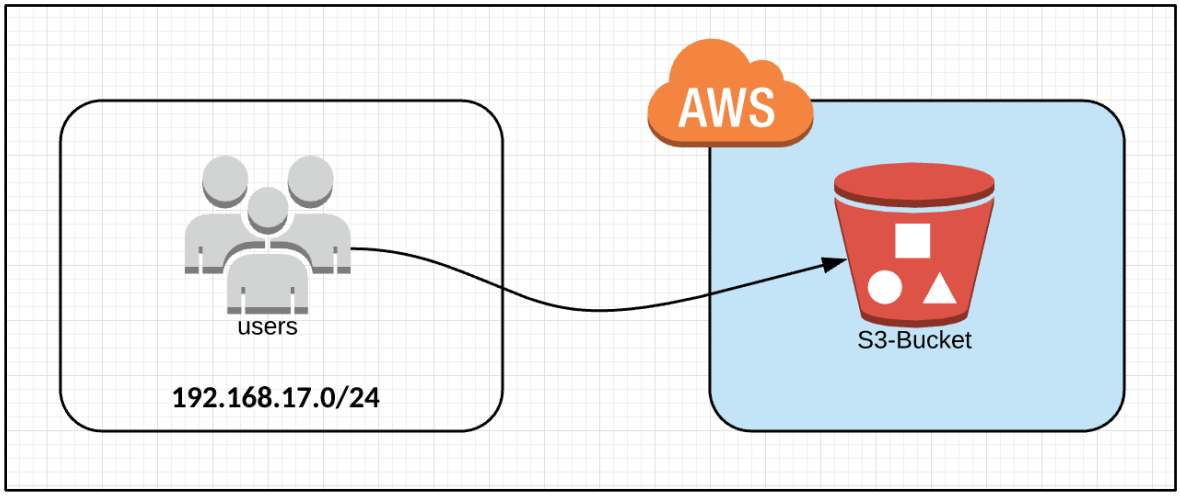
Service Catalog S3 Reference Blueprint

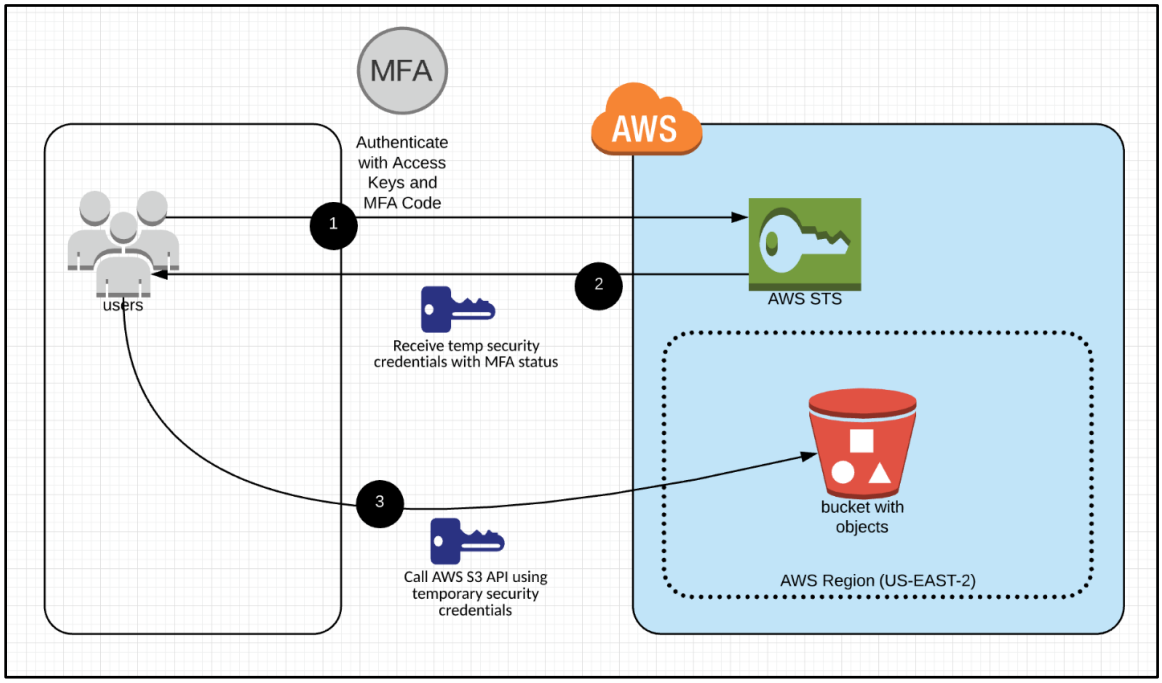
This reference blueprint demonstrates how an organization can leverage Service Catalog to provide Amazon Simple Storage Service (S3) buckets with various configurations for testing and integration.

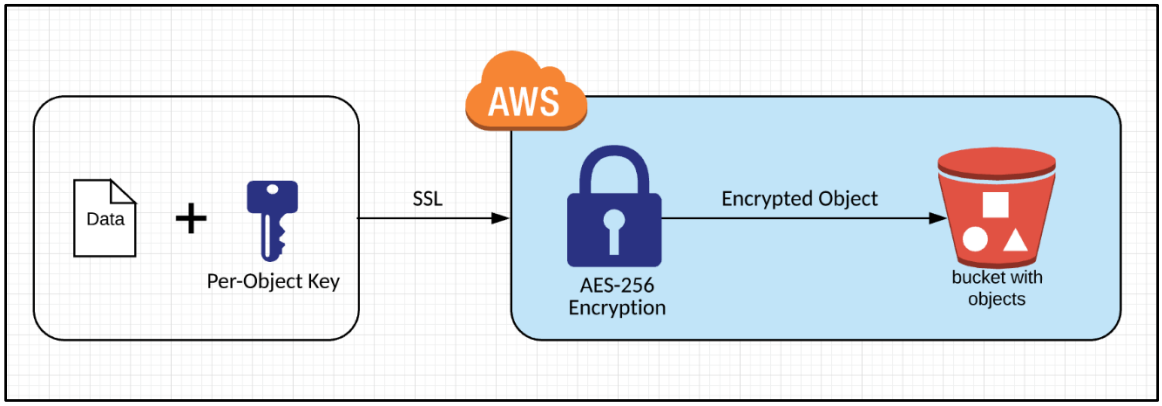
Once fully implemented, this reference blueprint creates a Service Catalog Portfolio called "Service Catalog S3 Reference Blueprint" with five associated products. The Service Catalog Product references CloudFormation templates for the Amazon S3 buckets which can be launched by end users through Service Catalog. The Service Catalog S3 products create S3 buckets with varying configurations: 1) Read-Only bucket with access from anywhere (public), 2) Private bucket with access restricted to a source CIDR block, 3) Private bucket with access requiring multi-factor authentication, 4) Private bucket with contents encrypted with S3 server side encryption, and 5) Private bucket with a transition rule set to migrate inactive objects to S3-Infrequent Access (IA) and Glacier.

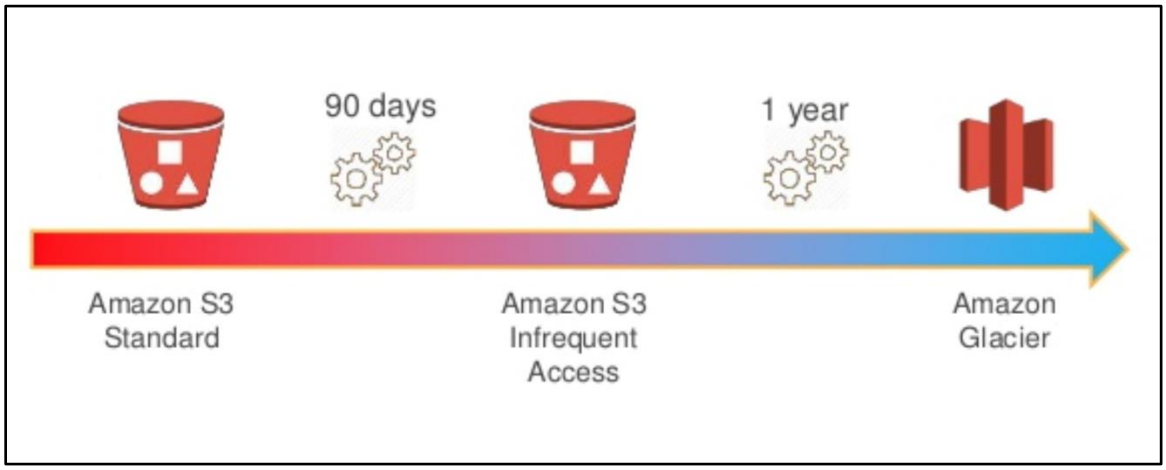
S3 Reference Blueprints











Getting Started

Note – Before you distribute the CloudFormation template to your organization, review the template and ensure that it is doing what you want it to do. Check IAM permissions, deletion policies, update stack behavior, and other aspects of the template and ensure that they are as per your expectations. These CloudFormation templates may need updates before you can use them in production.

There are two ways you can deploy this S3 reference blueprint:

1. If you are familiar with the AWS SDK and you are comfortable executing code or scripts (i.e. Python) to use AWS, we have included a simple Python script (**sc-s3-ra-setup.py**) you can use to deploy the reference blueprint. Follow the steps outlined in the "Script-based Installation Overview" below.

2. If you are not familiar with the AWS SDK, you can still perform the same steps manually while learning some of the hands-on steps to create AWS Service Catalog Portfolios, Products, assign constraints, apply Identity and Access Management (IAM) policies, etc. We have included step-by-step instructions in the "Manual Installation Overview" section in this document.

Regardless of the method you use, this reference blueprint creates a Service Catalog Portfolio with five associated S3 products. The Service Catalog products for the S3 reference blueprint can be launched by end users through Service Catalog.

Script-based Installation Overview

Prerequisites

The following prerequisites are required:

Python Development Environment

In order to deploy this reference blueprint utilizing the provided Python script, there is a requirement to have Python, Boto and the CLI installed. We do not provide all necessary documentation on how to get your development environment ready to use the AWS SDK. If you need some of these details please visit the Start Developing with Amazon Web Services (https://aws.amazon.com/developers/getting-started/) and the AWS SDK for Python (Boto3) (https://aws.amazon.com/sdk-for-python/). We also recommend you visit the official Python website for documentation and download options based on your preference of operating systems.

- Windows OS: https://www.python.org/downloads/windows/
- Mac OS X: https://www.python.org/downloads/mac-osx/
- Documentation: https://www.python.org/doc/

System with Permission to Execute Python Script

For automatic installation, the provided script utilizes modules for “boto3” and “random”. Instructions on installing and configuring the boto3 python module can be found in the Boto 3 Quick Start documentation at: http://boto3.readthedocs.io/en/latest/guide/quickstart.html

Install & Configure AWS Command Line Interface (CLI)

Installation and configuration of the AWS Command Line Interface (CLI). It is important to ensure that the AWS CLI configuration contains the correct target region as this region will be used to create the reference blueprint components within Service Catalog.
Instructions on installing and configuring the AWS Command Line Interface can be found on the AWS website at: https://aws.amazon.com/cli/

Using the AWS SDK to deploy this references blueprints

1. Download the reference blueprint zip file and expand its content into a folder.
    a. The location for the reference blueprint is: https://github.com/aws-samples/aws-service-catalog-reference-blueprints
2. Contents will include:
    * ./README.pdf (this file)
    * ./COPYING.pdf
    * ./LICENSE.pdf
    * ./NOTICE.pdf
    * ./sc-s3-ra-setup.py (python script used during setup process)
    * ./sc-s3-cidr-ra.yml (S3 Cloudformation Template in YAML)
    * ./sc-s3-cidr-ra.json (S3 Cloudformation Template in JSON)
    * ./sc-s3-encrypted-ra.json (S3 Cloudformation Template in JSON)
    * ./sc-s3-encrypted-ra.yml (S3 Cloudformation Template in YAML)
    * ./sc-s3-mfa-ra.json (S3 Cloudformation Template in JSON)
    * ./sc-s3-mfa-ra.yml (S3 Cloudformation Template in YAM)
    * ./sc-s3-public-ra.json (S3 Cloudformation Template in JSON)
    * ./sc-s3-public-ra.yml (S3 Cloudformation Template in YAML)
    * ./sc-s3-transition-ra.json (S3 Cloudformation Template in JSON)
    * ./sc-s3-transition-ra.yml (S3 Cloudformation Template in YAML)
    * ./sc-s3-ra-cidr-blueprint.png (image of reference blueprint)
    * ./sc-s3-ra-encrypted-blueprint.png (image of reference blueprint)
    * ./sc-s3-ra-mfa-blueprint.png (image of reference blueprint)
    * ./sc-s3-ra-public-blueprint.png (image of reference blueprint)
    * ./sc-s3-ra-transition-blueprint.png (image of reference blueprint)
3. Provide execute permissions to the python script.
4. Confirm AWS Region for deployment.
5. Execute the python setup script.

Service Catalog Portfolio Access

Once the setup script has completed there will be a new service catalog portfolio with new S3 products associated in the specified region. Before these products can be launched, access needs to be granted to the portfolio for the service catalog admin and end users. To grant access to the portfolio follow the steps outlined in the "Manual Installation Overview" section of this document.

Manual Installation Overview

Preparing your AWS Environment for Using Service Catalog

Before you get started with AWS Service Catalog you will need to be familiar with its components and the initial workflows for administrators and end users, which are the two primary user types in Service Catalog. You will also need to grant permissions to these users such that they can access the required functionality of Service Catalog.

AWS Service Catalog supports the following types of users:

- **Catalog administrators (administrators)** – Manage a catalog of products (applications and services), organizing them into portfolios and granting access to end users. Catalog administrators prepare AWS CloudFormation[1] templates, configure constraints, and manage IAM roles that are assigned to products to provide for advanced resource management.
- **End users** – Receive AWS credentials from their IT department or manager and use the AWS Management Console to launch products to which they have been granted access. Sometimes referred to as simply *users*, end users may be granted different permissions depending on your operational requirements. For example, a user may have the maximum permission level (to launch and manage all of the resources required by the products they use) or only permission to use particular service features.

Grant Permissions to Administrators and End Users

Catalog administrators and end users require different IAM permissions to use AWS Service Catalog. As a catalog administrator, you must have IAM permissions that allow you to access the AWS Service Catalog administrator console, create products, and manage products. Before your end users can use your products, you must grant them permissions that allow them to access the AWS Service Catalog end user console, launch products, and manage launched products as provisioned products.

AWS Service Catalog provides many of these permissions using managed policies. AWS maintains these policies and provides them in the AWS Identity and Access Management (IAM) service. You can use these policies by attaching them to the IAM users, groups, or roles that you and your end users use.

Create a Service Catalog Administrator and Grant Permissions

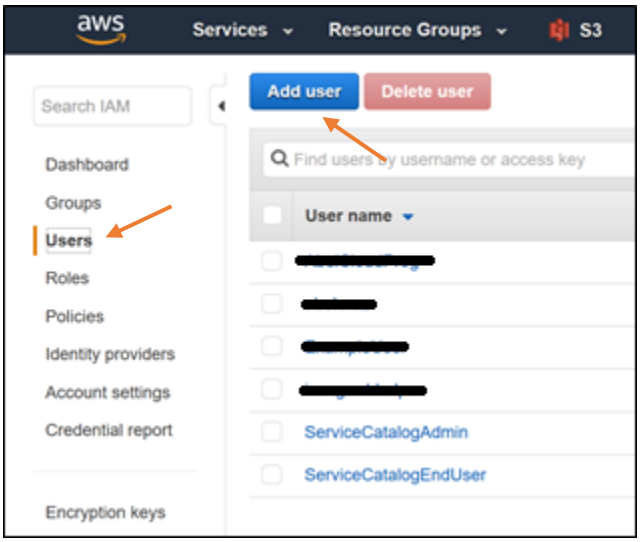
As a catalog administrator, you require access to the AWS Service Catalog administrator console view and IAM permissions that allow you to perform tasks such as the following:

[1] https://aws.amazon.com/documentation/cloudformation/

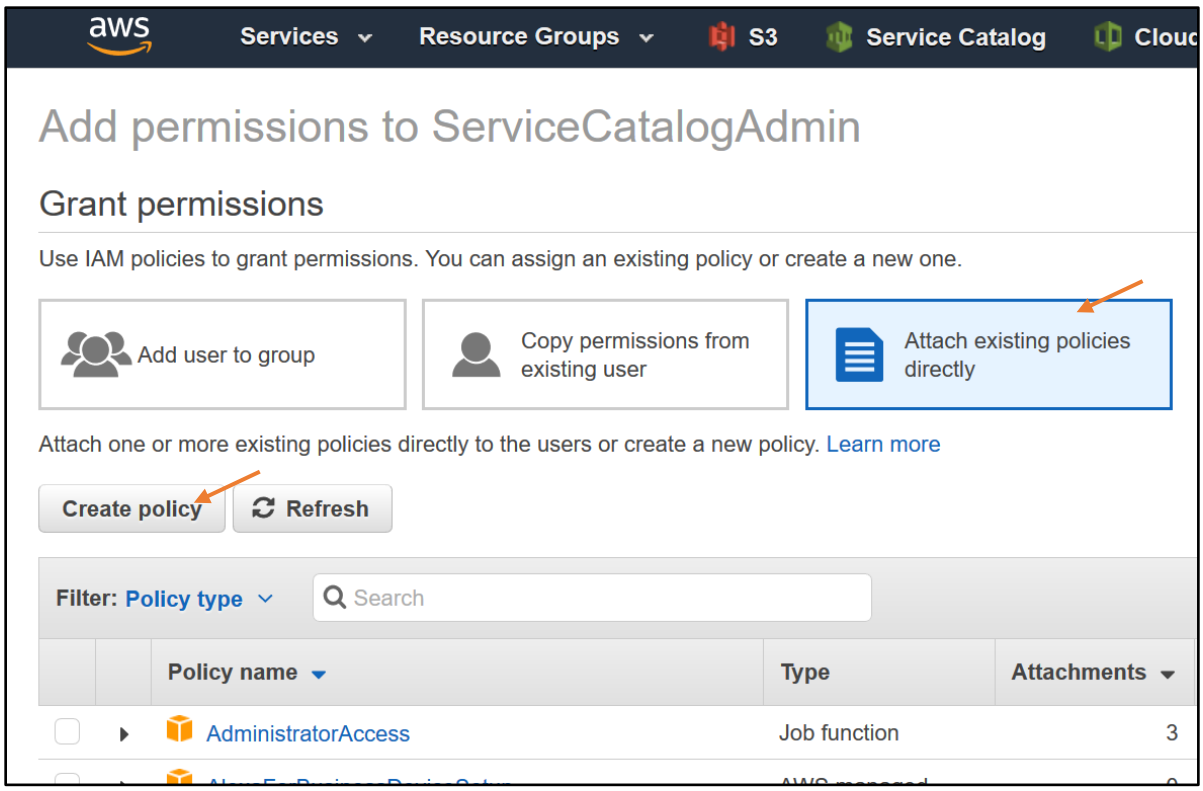
- Creating and managing portfolios
- Creating and managing products
- Adding template constraints to control the options that are available to end users when launching a product
- Adding launch constraints to define the IAM roles that AWS Service Catalog assumes when end users launch products
- Granting end users access to your products

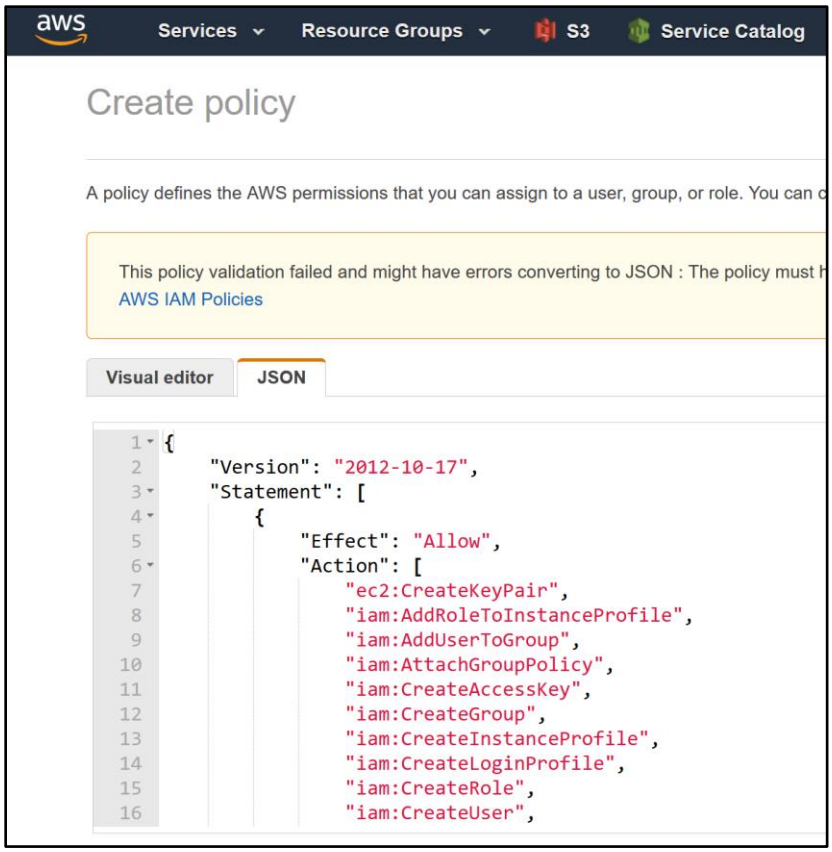
You, or an administrator who manages your IAM permissions, must attach policies to your IAM user, group, or role that are required to successfully deploy this product/solution.

1. Open the IAM console at https://console.aws.amazon.com/iam/.



2. In the navigation pane, choose **Users**. If you have already created an IAM user that you would like to use as the catalog administrator, choose the user name and choose **Add permissions**. Otherwise, create a user as follows:
    a. Choose **Add user**.
    b. For **User name**, type **ServiceCatalogAdmin**.
    c. Select **Programmatic access** and **AWS Management Console access**.
    d. Choose **Next: Permissions**.
3. Choose **Attach existing policies directly**.
4. Choose **Create policy** and do the following:
    a. For **Create Your Own Policy**, choose **Select**.
    b. For **Policy Name**, type **ServiceCatalogAdmin-AdditionalPermissions**.
    c. Copy the following example policy and paste it in **Policy Document**:

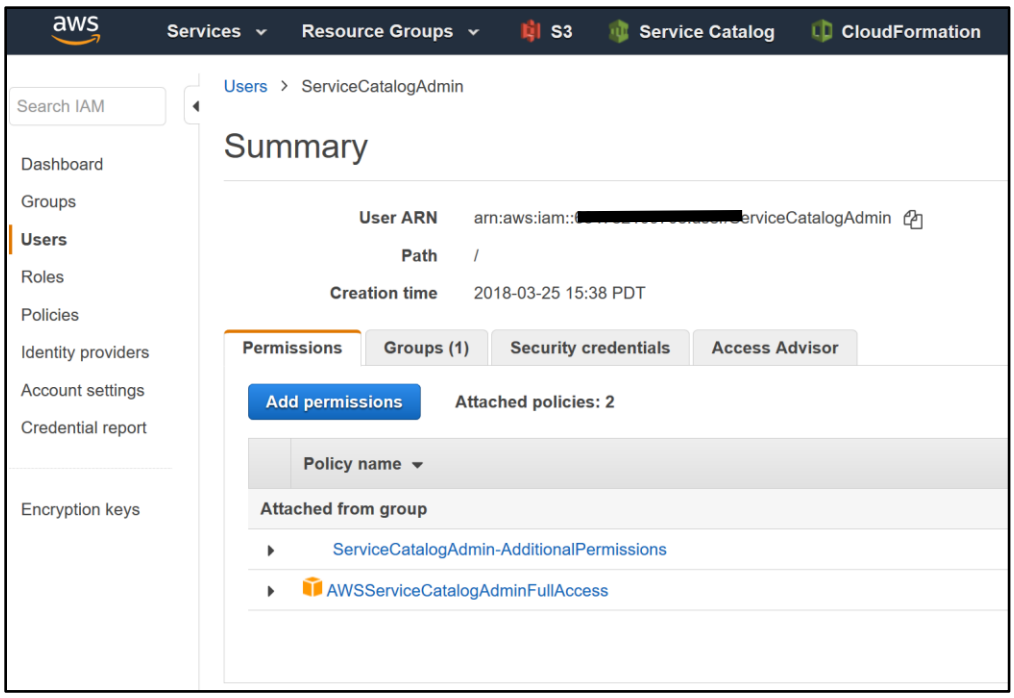




```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:CreateKeyPair",
                "iam:AddRoleToInstanceProfile",
                "iam:AddUserToGroup",
                "iam:AttachGroupPolicy",
                "iam:CreateAccessKey",
                "iam:CreateGroup",
                "iam:CreateInstanceProfile",
                "iam:CreateLoginProfile",
                "iam:CreateRole",
                "iam:CreateUser",
                "iam:Get*",
                "iam:List*",
                "iam:PutRolePolicy",
                "iam:UpdateAssumeRolePolicy"
            ],
            "Resource": [
                "*"
            ]
        }
    ]
}
```

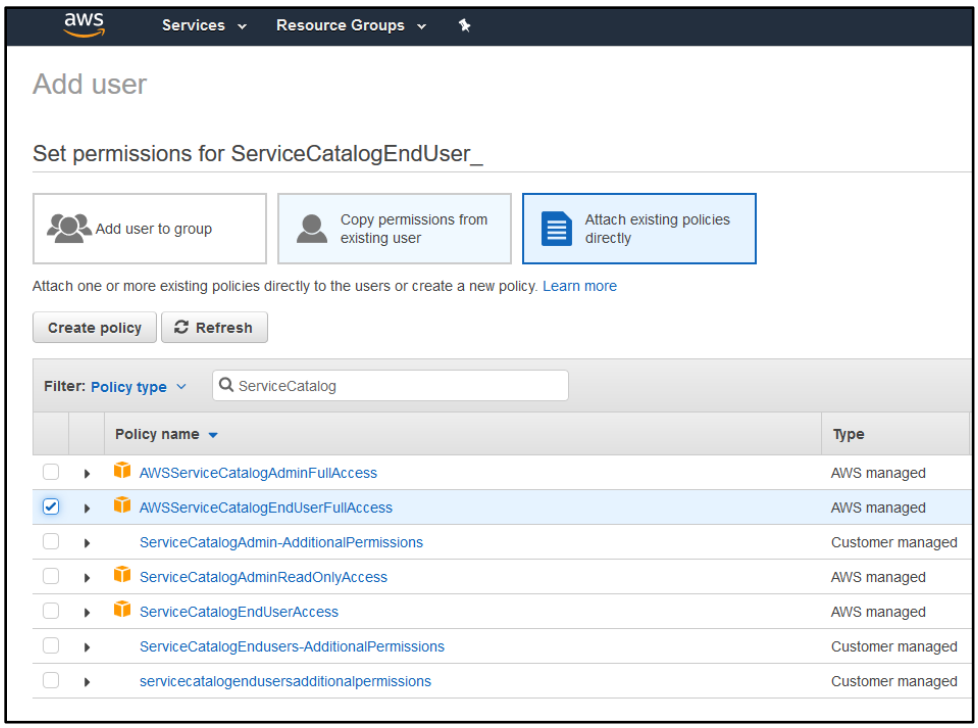
   d. (Optional) You must grant administrators additional permissions for Amazon S3 if they need to use a private CloudFormation template. For more information, see User Policy Examples[2] in the *Amazon Simple Storage Service Developer Guide*
   e. Choose **Create Policy**.
5. Return to the browser window with the permissions page and choose **Refresh**.
6. In the search field, type **ServiceCatalog** to filter the policy list.
7. Select the checkboxes for the **AWSServiceCatalogAdminFullAccess** and **ServiceCatalogAdmin-AdditionalPermissions** policies, and then choose **Next: Review**.
8. If you are updating a user, choose **Add permissions**.
9. If you are creating a user, choose **Create user**. You can download or copy the credentials and then choose **Close**.
10. To sign in as the catalog administrator, use your account-specific URL. To find this URL, choose **Dashboard** in the navigation pane and choose **Copy Link**. Paste the link in your browser, and use the name and password of the IAM user you created or updated in this procedure.

[2] https://docs.aws.amazon.com/AmazonS3/latest/dev/example-policies-s3.html

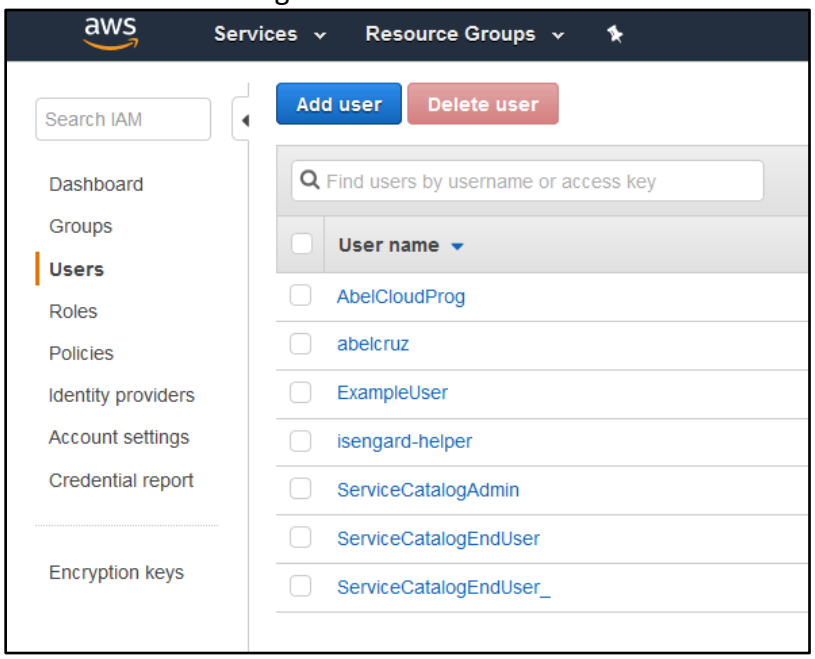


Create a Service Catalog End-User and Grant Permissions

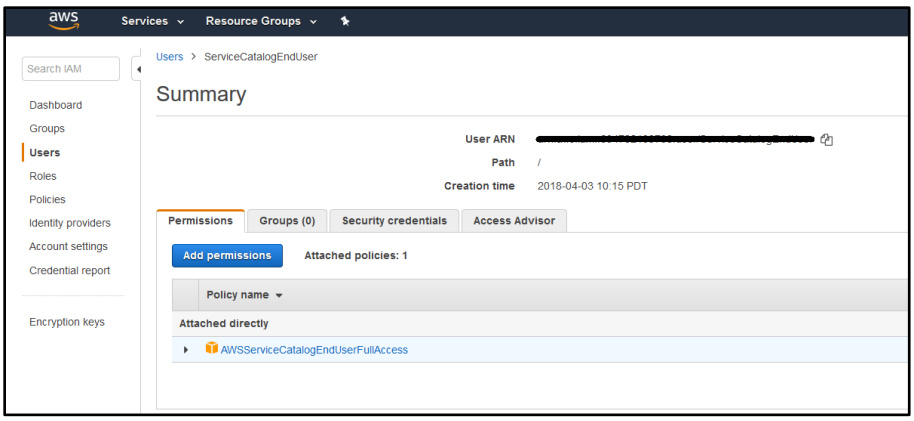
1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**. If you have already created an IAM user that you would like to use as the catalog end-user, choose the user name and choose **Add permissions** and skip to step 3. Otherwise, create a user as follows:
    e. Choose **Add user**.
    f. For **User name**, type **ServiceCatalogUser.**
    g. Select **Programmatic access** and **AWS Management Console access**.
    h. Choose **Next: Permissions**.
3. In the navigation pane, choose **Attach existing policies directly**
4. In the **Policy type** search bar type **AWSServiceCatalog** and select the **`AWSServiceCatalogEndUserFullAccess`** policy from the list.



5. Press **Next:Review**
6. Press **Create user**
7. Press **Close**
8. Once you are back on the window that shows the list of users select the ServiceCatalogEndUser from the list



9. In the summary page, copy and save the User ARN information located at the top of the window. You will need this information later when you try to launch the products in Service Catalog.



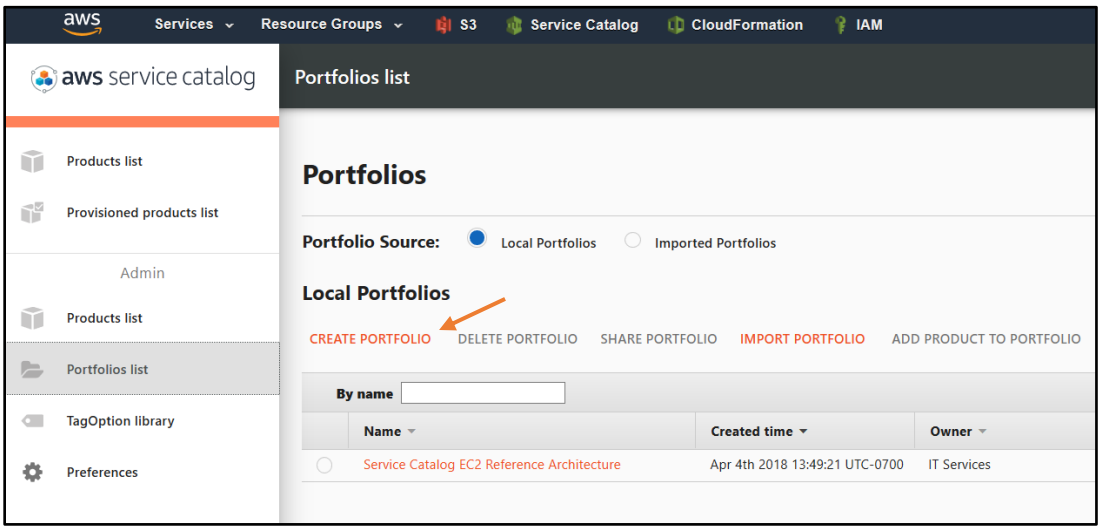
Required Files

To provision and configure portfolios and products, you use AWS CloudFormation templates, which are JSON— or YAML—formatted text files. For more information, see Template Formats in the *AWS CloudFormation User Guide*. These templates describe the resources that you want to provision. You can use the AWS CloudFormation editor or any text editor to create and save templates. For this Reference Blueprint, we've provided all required templates to get you started. The name of the templates are:

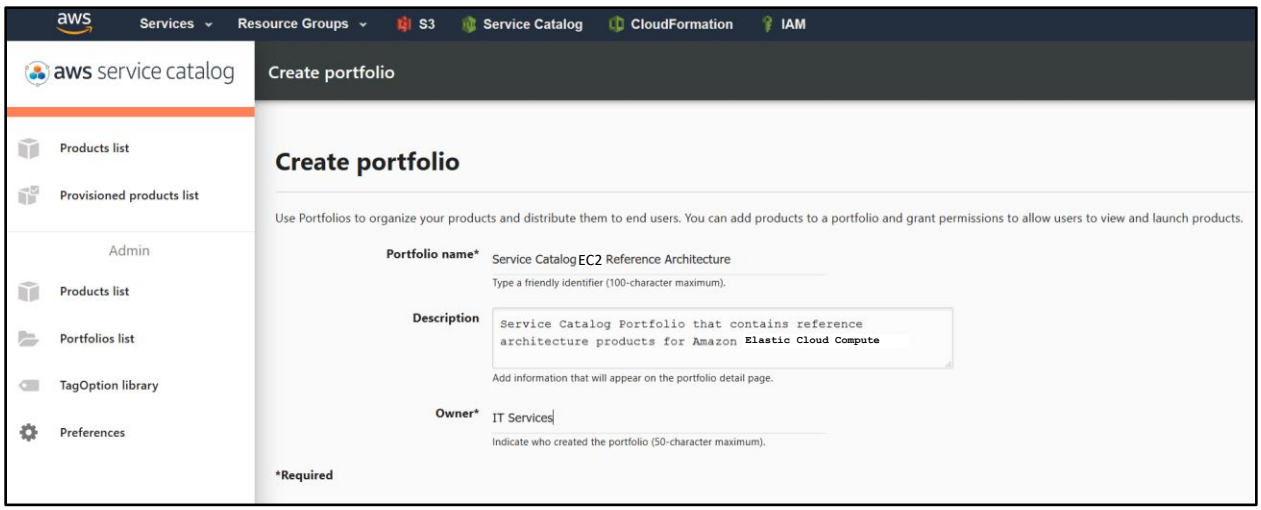
sc-s3-cidr-ra.yml or sc-s3-cidr-ra.json
sc-s3-encrypted-ra.json or sc-s3-encrypted-ra.yml
sc-s3-mfa-ra.json or sc-s3-mfa-ra.yml
sc-s3-public-ra.json or sc-s3-public-ra.yml
sc-s3-transition-ra.json or sc-s3-transition-ra.yml

Create an AWS Service Catalog Portfolio

To provide users with products, begin by creating a portfolio for those products.

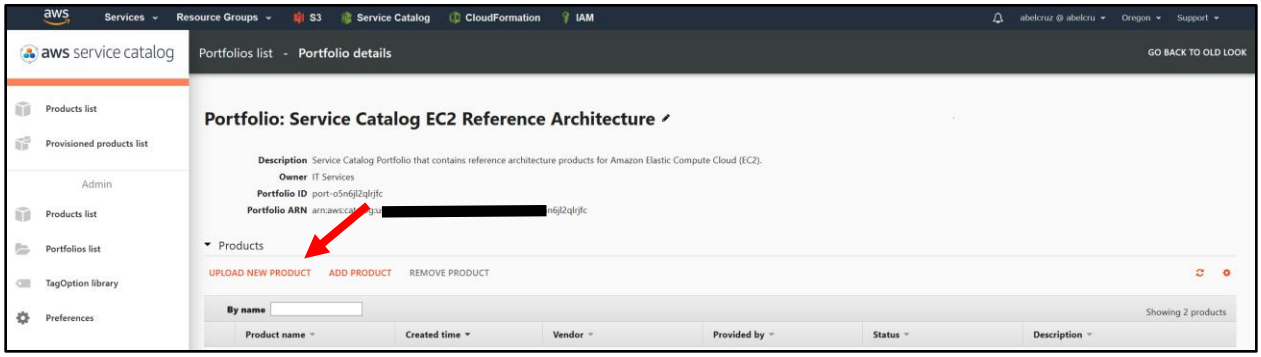


1. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/.
2. If you are using the AWS Service Catalog administrator console for the first time, choose **Get started** to start the wizard for configuring a portfolio. Otherwise, choose **Create portfolio**.
3. Type the following values:
    - **Portfolio name – Service Catalog S3 Reference Blueprint**
    - **Description – Service Catalog Portfolio that contains reference blueprint products for Amazon Elastic Compute Cloud.**
    - **Owner – IT Services**
4. Choose **Create**.



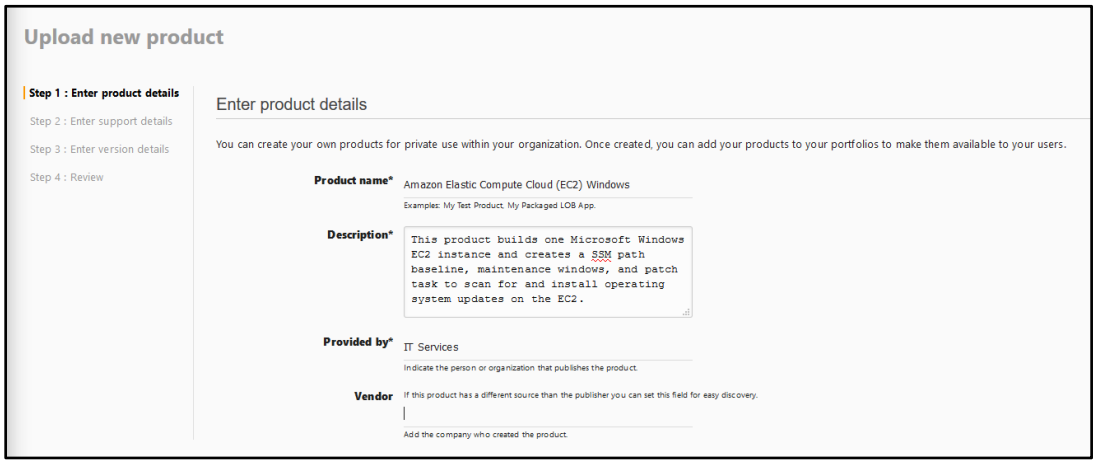
Create S3 Products in Service Catalog

1. In the Service Catalog console, click the Service Catalog drop-down at the top-left. Under *Admin*, click *Portfolios list*
2. Click the Portfolio which you created in the previous step (Service Catalog S3 Reference Blueprint)
3. Click *Upload New Product*

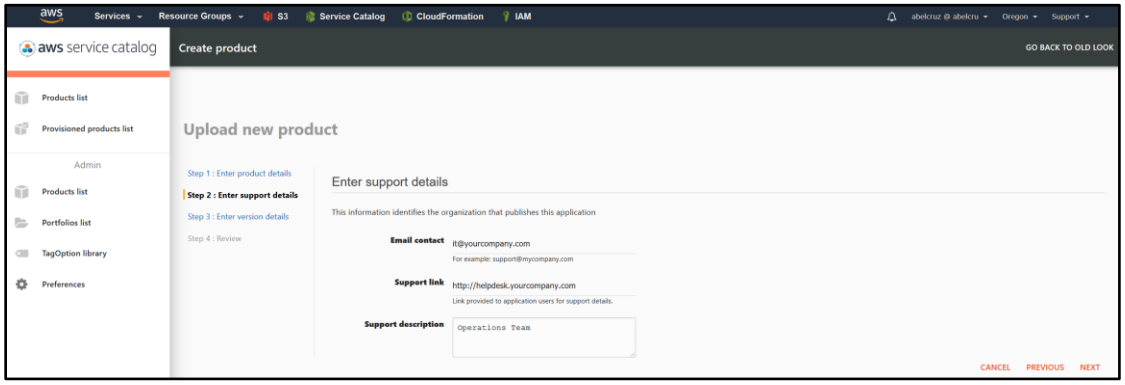


4. Enter in the requested product information and press **Next**:

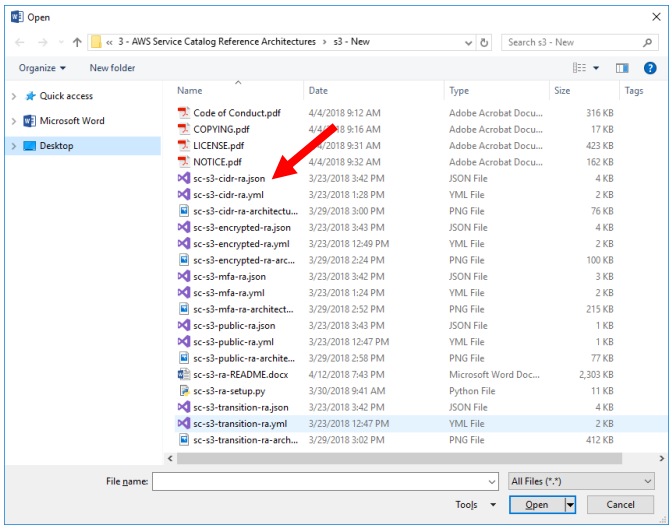
a. **Product Name:** Amazon Simple Storage Service (S3)
b. **Description:** This product builds five S3 reference products.
c. **Provided by:** IT Services
d. **Vendor:** <blank>



5. Enter in the requested support information, then click **Next**:
   a. **Email contact:** it@yourcompany.com
   b. **Support link:** http://helpdesk.yourcompany.com
   c. **Support description:** Operations Team



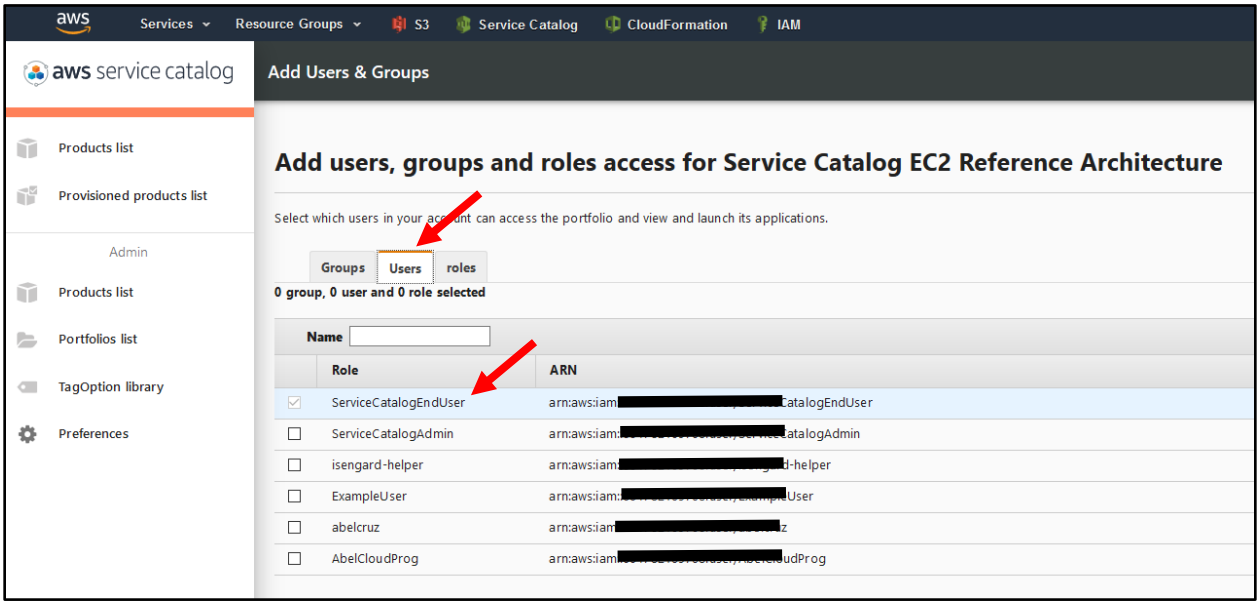
6. Click *Upload a template file*, then click *Browse*
   a. Navigate to the *sc-s3-cidr-ra.json* template on your system, then click *Open*



7. Fill in the remaining values on this page, then click *Next*
8. Verify the data on the review page, then click *Create*
9. Repeat steps 1 – 8 for each one of the other JSON CloudFormation templates for the other S3 products (i.e. sc-s3-encrypted-ra.json, sc-s3-mfa-ra.json, sc-s3-public-ra.json, sc-s3-transition-ra.json)

Share the Portfolio and S3 Product with end users and/or accounts

1. In the Service Catalog console, click *Portfolios List* under *Admin*
2. Click on the name of the portfolio you want to share (Service Catalog S3 Reference Blueprint)
3. Scroll down on this page and you will see two sections you can expand. Below is each section and the instructions on how to share with each:
    1. Users, groups and roles
        1. Click ***Add User, Group, or Role***
        2. Select each User, Group, and/or Role you wish to give access to this portfolio. In this example, we want to share with the **ServiceCatalogEndUser** user. The same procedure can be followed to share with a Group or with Roles.
        3. Click *Add Access*



Add a Launch Constraint to Assign an IAM Role

A launch constraint designates an IAM role that AWS Service Catalog assumes when an end user launches a product. For this step, you will add a launch constraint to the S3 product so that AWS Service Catalog can use the AWS resources that are part of the product's AWS CloudFormation template. This launch constraint will enable the end user to launch the product and, after it is launched, manage it as a provisioned product. For more information, see AWS Service Catalog Launch Constraints.

Without a launch constraint, you would need to grant additional IAM permissions to your end users before they could use S3/EC2. For example, the **`AWSServiceCatalogEndUserFullAccess`** policy grants the minimum IAM permissions required to access the AWS Service Catalog end user console view. By using a launch constraint, you can keep your end users' IAM permission to a minimum, which is an IAM best practice. For more information, see Grant least privilege in the *IAM User Guide*.

To add a launch constraint

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Policies**. Choose **Create policy** and do the following:

    a. For **Create Your Own Policy**, choose **Select**.
    b. For **Policy Name**, type **`s3-ra-LaunchPolicy`**.
    c. Copy the following example policy and paste it in **Policy Document**[3]:

[3] This policy has been created exclusively for demonstration purposes. As shown here, this policy demonstrates how to allow and/or restrict access to different AWS Services (i.e. S3, CloudFormation, etc.) resources. In its current form, the policy should never be used for production environments without consulting your security or

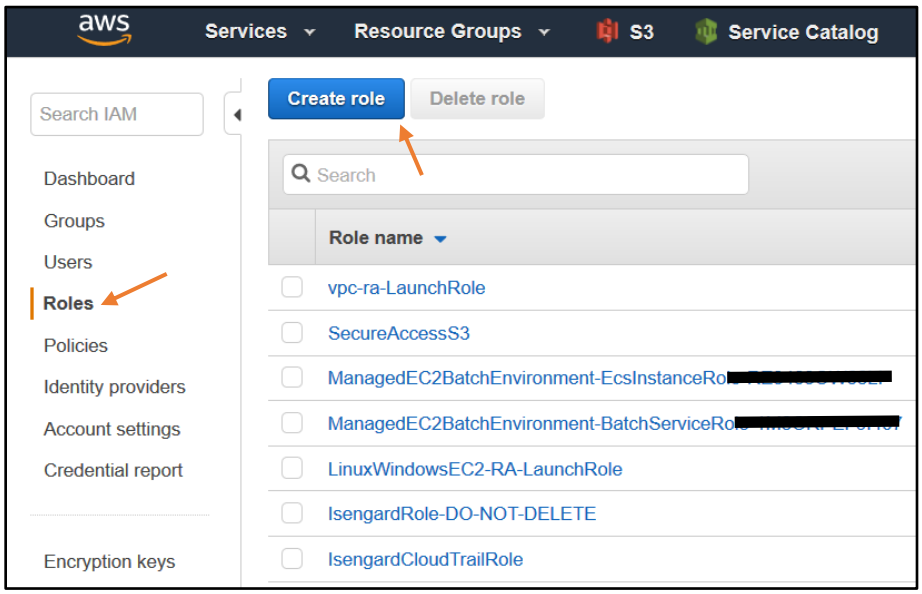
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "s3:GetObjectVersionTagging",
                "s3:CreateBucket",
                "s3:ListObjects",
                "s3:GetObjectAcl",
                "cloudformation:DescribeStackEvents",
                "s3:PutLifecycleConfiguration",
                "s3:GetObjectVersionAcl",
                "s3:PutBucketAcl",
                "cloudformation:UpdateStack",
                "s3:HeadBucket",
                "s3:GetIpConfiguration",
                "s3:GetBucketWebsite",
                "s3:GetBucketNotification",
                "s3:DeleteBucketPolicy",
                "s3:GetReplicationConfiguration",
                "s3:ListMultipartUploadParts",
                "cloudformation:DescribeStacks",
                "s3:GetObject",
                "cloudformation:DeleteStack",
                "s3:GetAnalyticsConfiguration",
                "s3:GetObjectVersionForReplication",
                "cloudformation:ValidateTemplate",
                "s3:ListBucketByTags",
                "s3:GetLifecycleConfiguration",
                "s3:GetBucketTagging",
                "s3:GetInventoryConfiguration",
                "s3:ListBucketVersions",
                "s3:GetBucketLogging",
                "s3:ListBucket",
                "s3:GetAccelerateConfiguration",
                "s3:GetBucketPolicy",
                "s3:PutEncryptionConfiguration",
                "s3:GetObjectVersionTorrent",
                "s3:PutBucketTagging",
                "s3:GetBucketRequestPayment",
                "s3:GetObjectTagging",
                "s3:GetMetricsConfiguration",
                "s3:DeleteBucket",
                "cloudformation:SetStackPolicy",
                "s3:ListBucketMultipartUploads",
                "s3:GetBucketVersioning",
                "s3:GetBucketAcl",
                "cloudformation:GetTemplateSummary",
                "s3:GetObjectTorrent",
                "s3:ListAllMyBuckets",
                "cloudformation:CreateStack",
                "s3:GetBucketCORS",
                "s3:PutBucketPolicy",
                "s3:GetBucketLocation",
```

compliance team. Keep in mind that an IAM policy should only grant the least amount of privileges required to perform a task and nothing more. For more information, see IAM Best Practices (IAM Best Practices).

```
                "s3:GetObjectVersion"
            ],
            "Resource": "*"
        }
    ]
}
```

d. Choose **Create Policy**.

3. In the navigation pane, choose **Roles**. Choose **Create role** and do the following:
   a. For **Select role type**, choose **AWS service** and then choose **Service Catalog**. Select **Service Catalog** under the **Select your use case** heading and then press the **Next: Permissions** button.
   b. Type **s3** in the Policy type search box and select the checkbox for the **`s3-ra-LaunchPolicy`** policy, and then choose **Next: Review**.
   c. For **Role name**, type **`s3-ra-LaunchRole`**.
   d. Choose **Create role**.



4. Open the AWS Service Catalog console at https://console.aws.amazon.com/servicecatalog/.
5. Choose the **Service Catalog S3 Reference Blueprint** portfolio.
6. On the portfolio details page, expand the **Constraints** section, and then choose **Add constraints**.
7. For **Product**, choose **Amazon Simple Storage Service (S3)**, and for **Constraint type**, choose **Launch**. Choose **Continue**.
8. On the **Launch constraint** page, for **IAM role**, choose **`s3-ra-LaunchRole`**, and then choose **Submit**.

Service Catalog Product Launch

Once access has been provided to one or more end users the S3 reference blueprint products can be launched. To launch an S3 reference blueprint product the user needs to log into Service Catalog, select the S3 Reference Blueprint Product and click launch. The launch process will ask the end user for various details about how the S3 product will be configured. For some of the products, you will need to provide the ServiceCatalogEndUser ARN information you saved above. After the form fields are filled out and the product is launched Service Catalog will execute a CloudFormation stack to build the product and provide the S3 details back to the end user.

Service Catalog S3 Reference Blueprint Cleanup

To remove the S3 Reference Blueprint from Service Catalog perform the following steps:

1. Terminate all Service Catalog S3 Reference Blueprint provisioned products.
2. Remove all products from the portfolio.
3. Remove all constraints from the portfolio.
4. Remove all access to users, groups and roles from the portfolio.
2. Remove all shares associated with the portfolio.
3. Remove all tags from the portfolio.
4. Remove all Tag Options from the portfolio.
5. Delete all products from Service Catalog.
6. Delete the portfolio from Service Catalog.

Disclaimer

AWS has made efforts to create safe and repeatable blueprints for users to use. However, as expressed in the AWS Service Terms, Amazon.com specifically disclaims all warranties in respect of the artifacts provided in the SCIB program. Furthermore, user acknowledges that SCIB content may in fact have bugs and/or may malfunction. User expressly acknowledges that there is inherent risks associated with using or testing non-production or sample products or services.

Note – Before you distribute the CloudFormation template to your organization, review the template and ensure that it is doing what you want it to do. Check IAM permissions, deletion policies, update stack behavior, and other aspects of the template and ensure that they are as per your expectations. These CloudFormation templates may need updates before you can use them in production.

License

Authors

Israel Lawson – AWS Sr. Solutions Architect
Kanchan Waikar – AWS Solutions Architect

Abel Cruz – AWS Sr. Business Development Manager

Acknowledgments

The following AWS team members have provided guidance, code review and other assistance throughout the design of this reference blueprint.

- David Aiken – AWS Solutions Architect Manager
- Mahdi Sajjadpour – AWS Principal Business Development Manager
- Phil Chen – AWS Sr. Solutions Architect
- Kenneth Walsh – AWS Solutions Architect