
AWS Service Catalog

RDS Reference Blueprint



AWS Service Catalog RDS Reference Blueprint

Copyright © 2018 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

Service Catalog RDS Reference Blueprint	2
RDS Reference Blueprint	2
Getting Started.....	4
Script-based Installation Overview	5
Prerequisites.....	5
Python Development Environment	5
System with Permission to Execute Python Script	5
Install & Configure AWS Command Line Interface (CLI)	5
Using the AWS SDK to deploy this references blueprints	6
Modify Python code	Error! Bookmark not defined.
Service Catalog Portfolio Access	6
Manual Installation Overview	6
Preparing your AWS Environment for Using Service Catalog	6
Grant Permissions to Administrators and End Users	7
Create a Service Catalog Administrator and Grant Permissions.....	7
Create a Service Catalog End-User and Grant Permissions.....	11
Required Files	11
Create an AWS Service Catalog Portfolio.....	12
Create RDS Products in Service Catalog.....	13
Share the Portfolio and RDS Product with end users and/or accounts.....	15
Add a Launch Constraint to Assign an IAM Role	16
To add a launch constraint.....	16
Service Catalog Product Launch.....	19
Service Catalog RDS Reference Blueprint Cleanup.....	19
Disclaimer.....	20
License.....	20
Authors.....	20
Acknowledgments.....	20

Service Catalog RDS Reference Blueprint

This reference blueprint demonstrates how an organization can leverage Service Catalog to provide Amazon Relation Database Service (RDS) databases testing and integration.

When implemented this reference blueprint creates a Service Catalog Portfolio called “Service Catalog RDS Reference Blueprint” with four associated products. The Service Catalog Products reference CloudFormation templates can deploy PostgreSQL, MySQL, MariaDB and Microsoft SQL which can be launched by end users through Service Catalog as either single-instance databases or multi-availability zone databases.

RDS Reference Blueprint

To try to deploy either of the RDS blueprint options described in this document, you do not need to recreate the reference VPC shown in the following diagram. If, however, you are interested in deploying a duplicate of the reference VPC shown here, you can first follow the steps outlined in the “Service Catalog VPC Reference Blueprint” documentation that was included in the ZIP file under the */VPC directory.

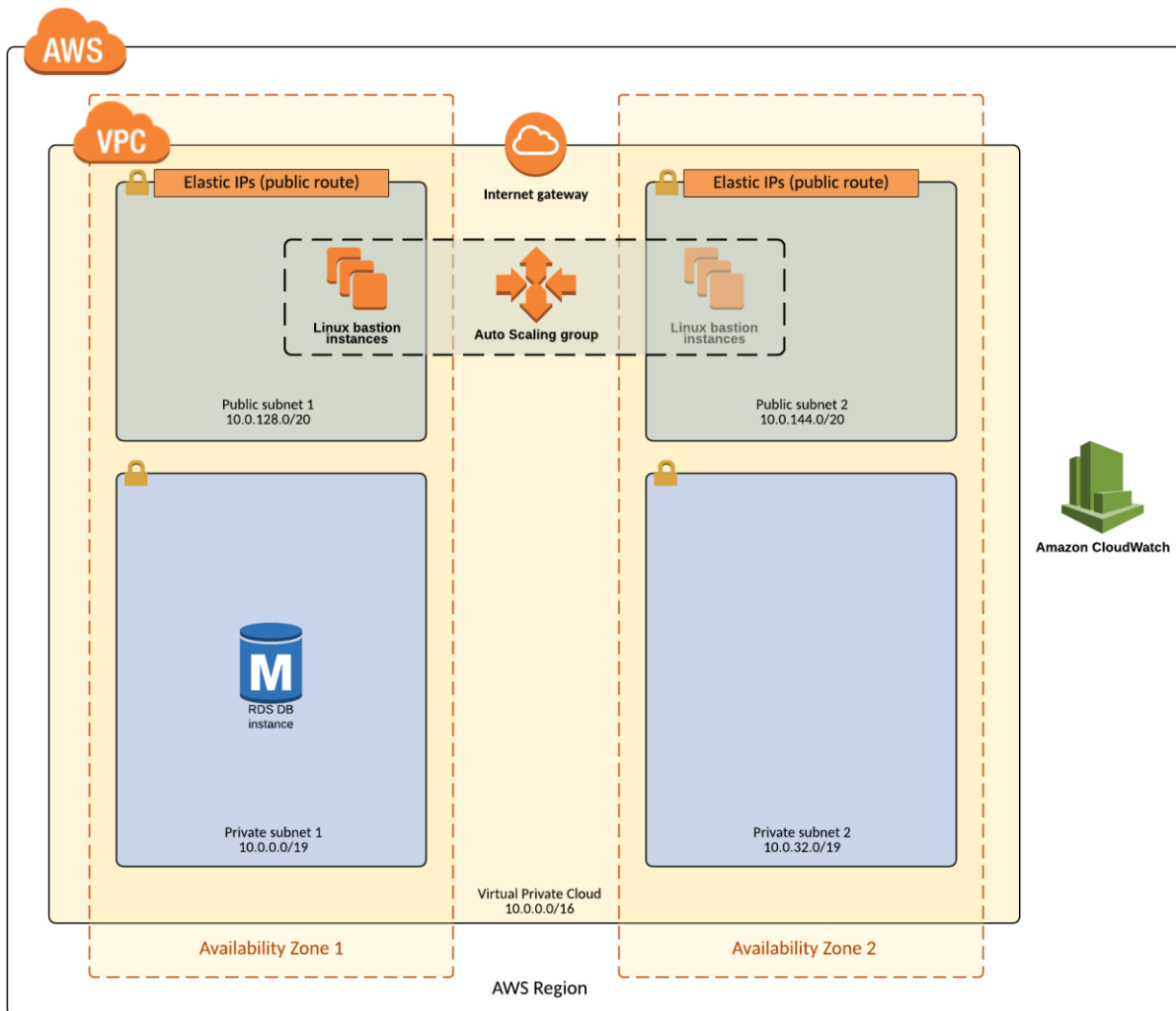


Figure 1 – Single-Instance Database Deployment

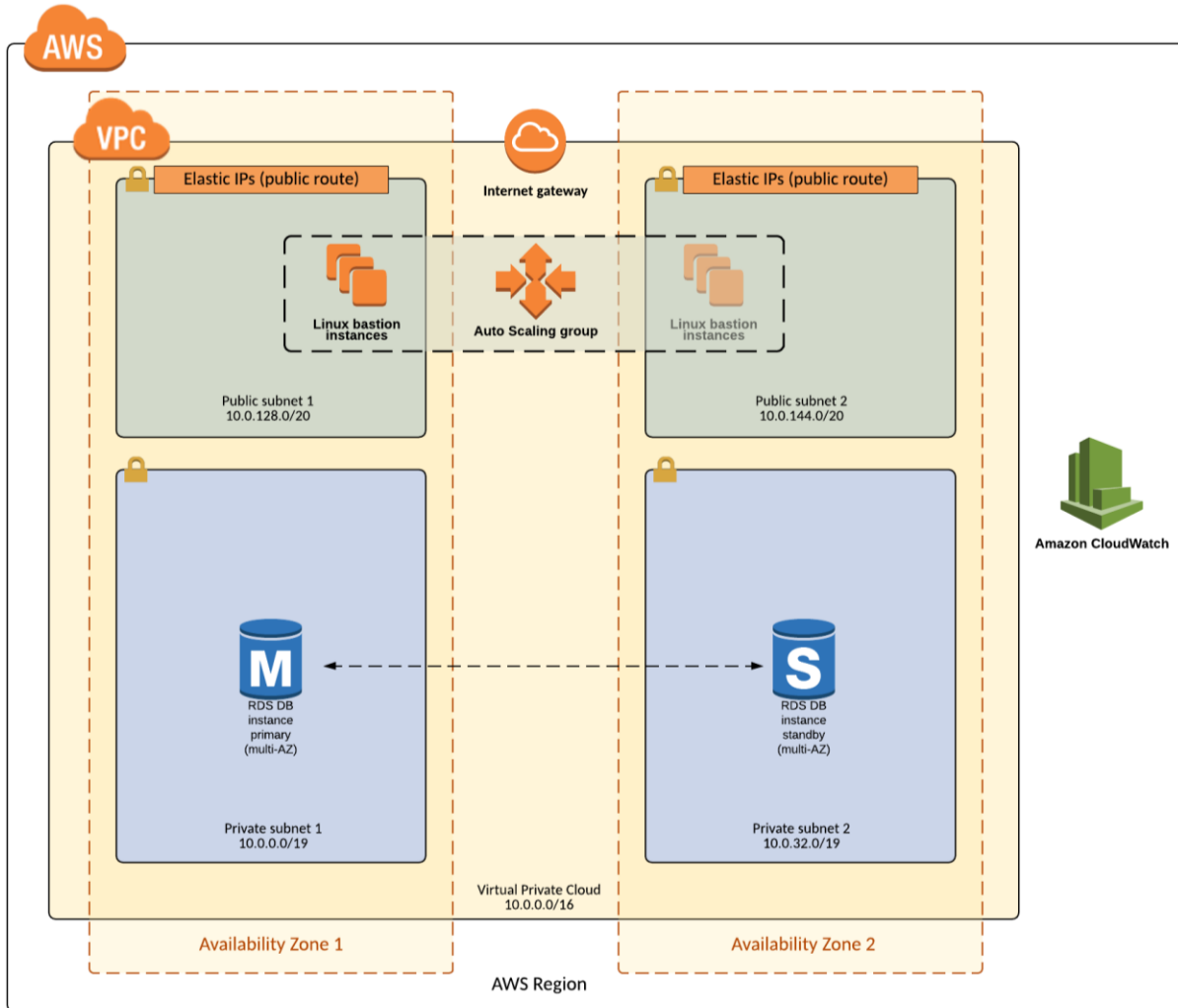


Figure 2 - Multi-Availability Zone Database Deployment

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 RDS 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-rds-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 four associated RDS products (MarisDB, MSSQL, Postgresql, MySQL). The Service Catalog CloudFormation template for each RDS product used in this 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/) (<https://aws.amazon.com/developers/getting-started/>) and the [AWS SDK for Python \(Boto3\)](https://aws.amazon.com/sdk-for-python/) (<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-mr-ra-setup.py (python script used during setup process)
 - * ./sc-rds-mariadb-ra.yml (RDS Cloudformation Template in YAML)
 - * ./sc-rds-mariadb-ra.json (RDS Cloudformation Template in JSON)
 - * ./sc-rds-mssql-ra.yml (RDS Cloudformation Template in YAML)
 - * ./sc-rds-mssql-ra.json (RDS Cloudformation Template in JSON)
 - * ./sc-rds-mysql-ra.yml (RDS Cloudformation Template in YAML)
 - * ./sc-rds-mysql-ra.json (RDS Cloudformation Template in JSON)
 - * ./sc-rds-postgresql-ra.yml (RDS Cloudformation Template in YAML)
 - * ./sc-rds-postgresql-ra.json (RDS Cloudformation Template in JSON)
 - * ./sc-rds-ra-blueprint-multi-az.png (image of reference multi-az blueprint)
 - * ./sc-rds-ra-blueprint-single-instance.png (image of reference single-instance 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 RDS 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 CloudFormation¹ 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

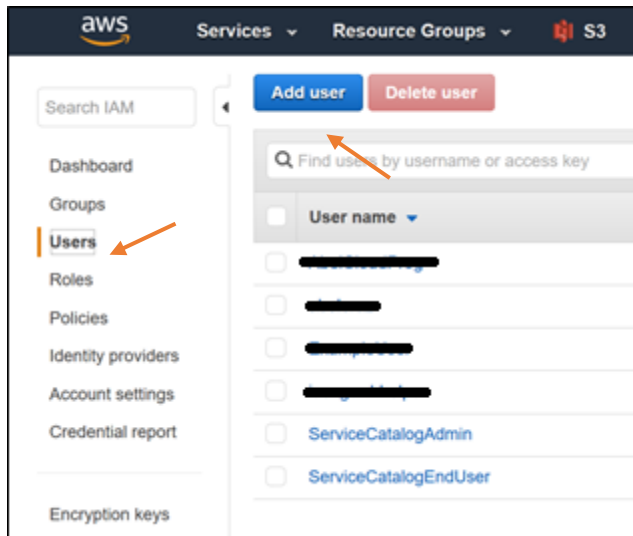
As a catalog administrator, you require access to the AWS Service Catalog administrator console view and IAM permissions that allow you to perform tasks such as the following:

- 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/>.

¹ <https://aws.amazon.com/documentation/cloudformation/>



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**:

aws Services Resource Groups S3 Service Catalog Cloud

Add permissions to ServiceCatalogAdmin

Grant permissions

Use IAM policies to grant permissions. You can assign an existing policy or create a new one.

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 Search

	Policy name	Type	Attachments
<input type="checkbox"/>	AdministratorAccess	Job function	3

aws Services Resource Groups S3 Service Catalog

Create policy

A policy defines the AWS permissions that you can assign to a user, group, or role. You can c

This policy validation failed and might have errors converting to JSON : The policy must h
[AWS IAM Policies](#)

Visual editor

JSON

```

1 {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "ec2:CreateKeyPair",
8         "iam:AddRoleToInstanceProfile",
9         "iam:AddUserToGroup",
10        "iam:AttachGroupPolicy",
11        "iam:CreateAccessKey",
12        "iam:CreateGroup",
13        "iam:CreateInstanceProfile",
14        "iam:CreateLoginProfile",
15        "iam:CreateRole",
16        "iam:CreateUser",

```

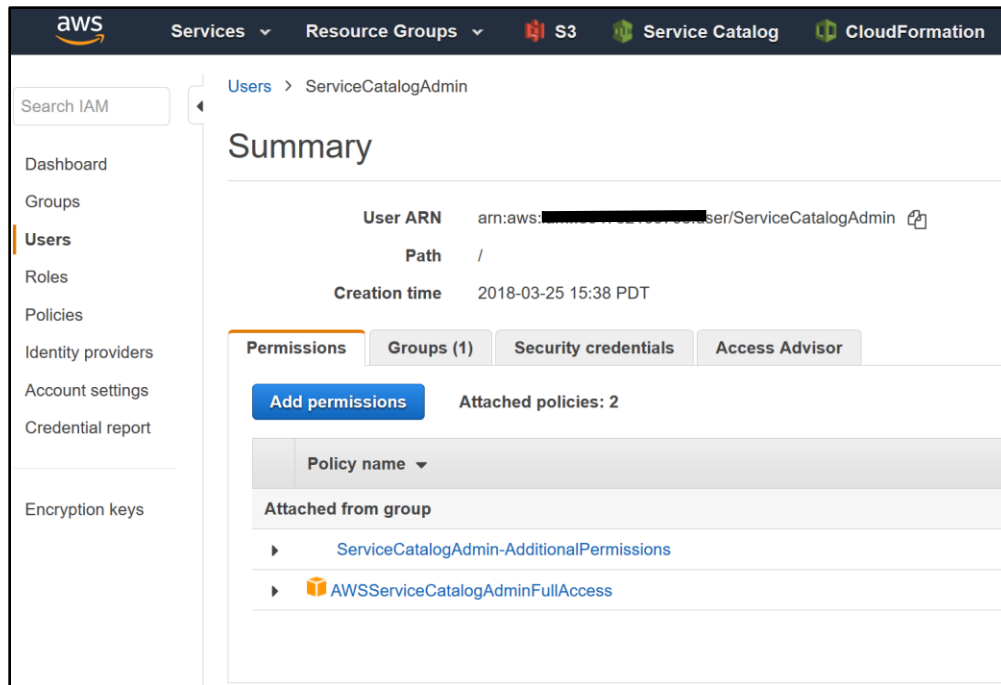
```

{
  "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 Examples²](#) 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**

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 four templates to get you started. The name of the templates are:

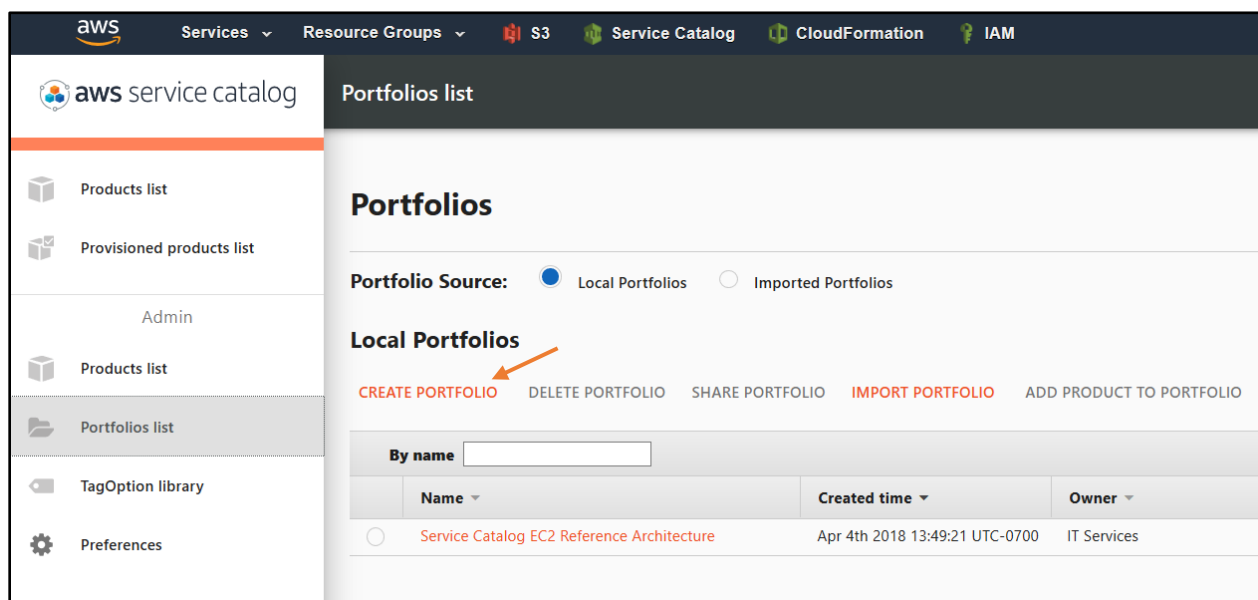
- sc-rds-mariadb-ra.yml (RDS Cloudformation Template in YAML)
- sc-rds-mariadb-ra.json (RDS Cloudformation Template in JSON)

- sc-rds-mssql-ra.yml (RDS Cloudformation Template in YAML)
- sc-rds-mssql-ra.json (RDS Cloudformation Template in JSON)
- sc-rds-mysql-ra.yml (RDS Cloudformation Template in YAML)
- sc-rds-mysql-ra.json (RDS Cloudformation Template in JSON)
- sc-rds-postgresql-ra.yml (RDS Cloudformation Template in YAML)
- sc-rds-postgresql-ra.json (RDS Cloudformation Template in JSON)

You only need either the JSON or YML version of the files for each RDS product, but not both.

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 RDS Reference Blueprint
 - **Description** – Service Catalog Portfolio that contains reference blueprint products for Amazon Relational Database Service.
 - **Owner** – IT Services
4. Choose **Create**.

Create portfolio

Use Portfolios to organize your products and distribute them to end users. You can add products to a portfolio and grant permissions to allow users to view and launch products.

Portfolio name* Service Catalog EC2 Reference Architecture
Type a friendly identifier (100-character maximum).

Description Service Catalog Portfolio that contains reference architecture products for Amazon Elastic Cloud Compute
Add information that will appear on the portfolio detail page.

Owner* IT Services
Indicate who created the portfolio (50-character maximum).

*Required

Create RDS 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 RDS Reference Blueprint)
3. Click *Upload New Product*

Portfolio: Service Catalog EC2 Reference Architecture

Description: Service Catalog Portfolio that contains reference architecture products for Amazon Elastic Compute Cloud (EC2).

Owner: IT Services

Portfolio ID: port-05nfg12qlyfc

Portfolio ARN: arn:aws:servicecatalog:us-east-1:123456789012:portfolio/port-05nfg12qlyfc

Products

UPLOAD NEW PRODUCT ADD PRODUCT REMOVE PRODUCT

By name

Product name Created time Vendor Provided by Status Description Showing 2 products

4. Enter in the requested product information and press **Next**:
 - a. **Product Name:** Amazon Relational Database Service (RDS) – MariaDB
 - b. **Description:** This product builds a MariaDB RDS reference products.
 - c. **Provided by:** IT Services
 - d. **Vendor:** <blank>

Upload new product

Step 1 : Enter product details
Step 2 : Enter support details
Step 3 : Enter version details
Step 4 : Review

Enter product details

You can create your own products for private use within your organization. Once created, you can add your products to your portfolios to make them available to your users.

Product name* Amazon Elastic Compute Cloud (EC2) Windows
Examples: My Test Product, My Packaged LOB App.

Description* This product builds one Microsoft Windows EC2 instance and creates a SSM path baseline, maintenance windows, and patch task to scan for and install operating system updates on the EC2.

Provided by* IT Services
Indicate the person or organization that publishes the product.

Vendor
If this product has a different source than the publisher you can set this field for easy discovery.
Add the company who created the product.

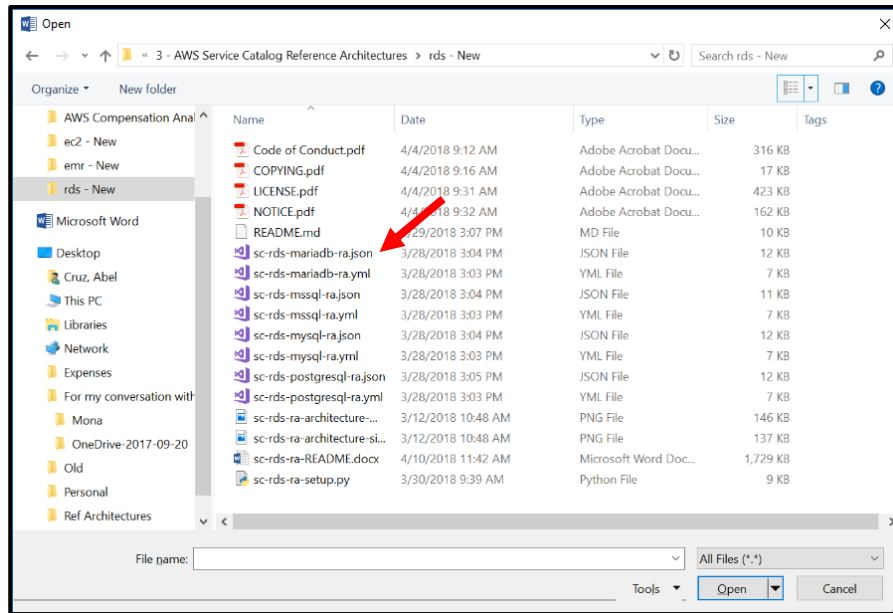
5. Enter in the requested support information, then click **Next**:
 - a. **Email contact:** it@yourcompany.com
 - b. **Support link:** <http://helpdesk.yourcompany.com>
 - c. **Support description:** Operations Team

The screenshot shows the AWS Service Catalog interface. The left sidebar contains navigation links: Products list, Provisioned products list, Admin, Products list, Portfolios list, TagOption library, and Preferences. The main content area is titled 'Create product' and 'Upload new product'. It shows 'Step 2 : Enter support details' as the active step. The form includes:

- Email contact:** it@yourcompany.com (with example text: support@mycompany.com)
- Support link:** http://helpdesk.yourcompany.com (with note: Link provided to application users for support details.)
- Support description:** Operations Team

At the bottom right, there are buttons for CANCEL, PREVIOUS, and NEXT.

6. Click *Upload a template file*, then click *Browse*
 - a. Navigate to the *sc-rds-mariadb-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 of the following Product Name, Product Description & CloudFormation template name combination:

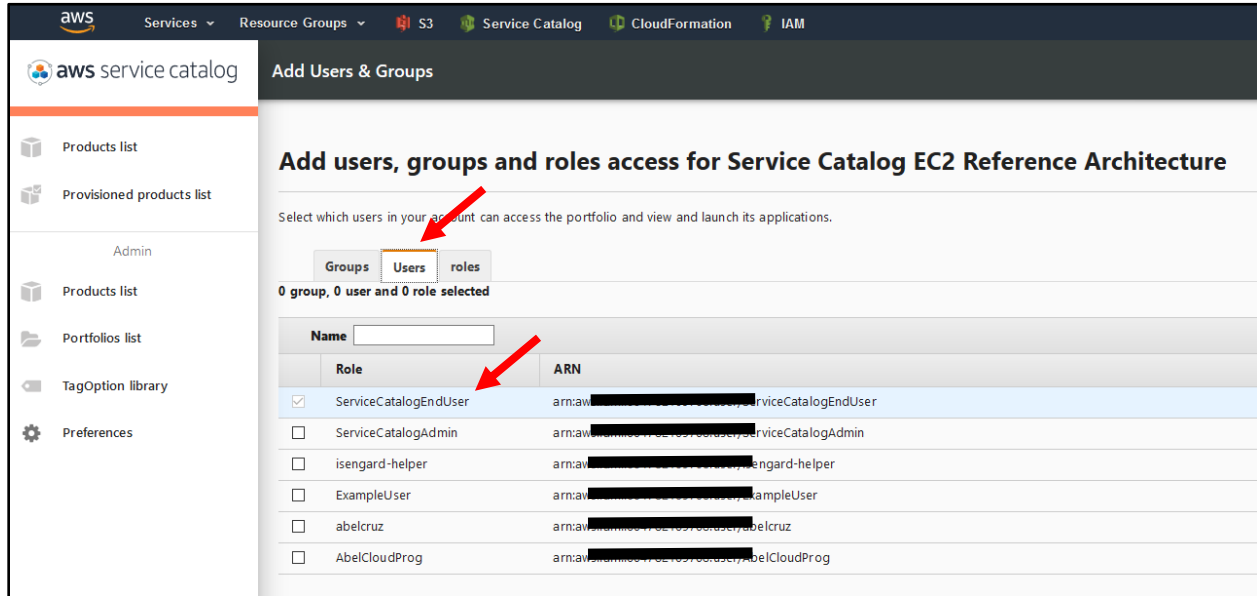
Product Name	Product Description	CloudFormation Template Name
Amazon Relational Database Service (RDS) – MSSQL	This product builds a MSSQL RDS reference products	sc-rds-mssql-ra.json
Amazon Relational Database Service (RDS) – MySQL	This product builds a MySQL RDS reference products	sc-rds-mysql-ra.json
Amazon Relational Database Service (RDS) – PostgreSQL	This product builds a PostgreSQL RDS reference products	sc-rds-postgresql-ra.json

Share the Portfolio and RDS 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 RDS 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 RDS 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 RDS/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 **rds-ra-LaunchPolicy**.

- c. Copy the following example policy and paste it in **Policy Document**³:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "ec2:AuthorizeSecurityGroupIngress",
        "rds:CreateDBSubnetGroup",
        "ec2:UpdateSecurityGroupRuleDescriptionsIngress",
        "cloudformation:DescribeStackEvents",
        "ec2:RevokeSecurityGroupEgress",
        "rds:ModifyDBInstance",
        "ec2:DescribeAccountAttributes",
        "cloudformation:UpdateStack",
        "rds:ModifyDBClusterSnapshotAttribute",
        "rds>DeleteDBInstance",
        "ec2:UpdateSecurityGroupRuleDescriptionsEgress",
        "rds:DescribeDBSnapshots",
        "rds>DeleteDBSnapshot",
        "rds:StartDBInstance",
        "cloudformation:DescribeStacks",
        "ec2:RevokeSecurityGroupIngress",
        "rds>DeleteDBSubnetGroup",
        "s3:GetObject",
        "rds>CreateDBSnapshot",
        "rds:RestoreDBInstanceFromDBSnapshot",
        "cloudformation>DeleteStack",
        "rds>CreateDBClusterSnapshot",
        "rds:ModifyDBSnapshotAttribute",
        "ec2:DescribeSubnets",
        "cloudformation:ValidateTemplate",
        "rds:AuthorizeDBSecurityGroupIngress",
        "rds:RestoreDBClusterFromSnapshot",
        "rds:DescribeDBSubnetGroups",
        "rds>CreateDBInstance",
        "ec2:CreateSecurityGroup",
```

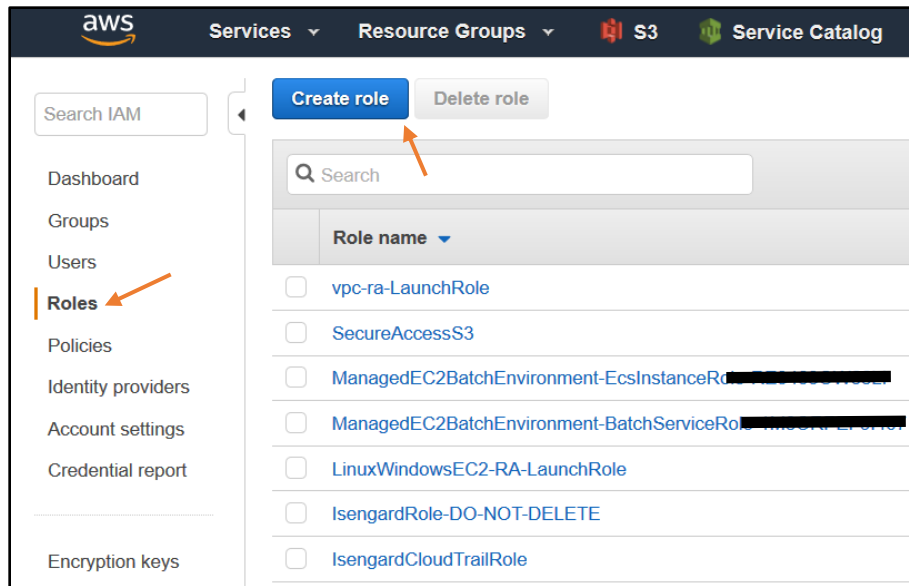
³ This policy has been created exclusively for demonstration purposes. As shown here, this policy demonstrates how to allow and/or restrict access to different AWS Services (i.e. RDS, CloudFormation, EC2, etc.) resources. In its current form, the policy should never be used for production environments without consulting your security or 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](#)).

```

        "rds:DescribeDBInstances",
    "rds:DescribeDBClusterSnapshotAttributes",
        "cloudformation:SetStackPolicy",
        "ec2:AuthorizeSecurityGroupEgress",
        "rds:CopyDBSnapshot",
        "rds:CopyDBClusterSnapshot",
        "rds:StopDBInstance",
        "rds:DescribeDBSnapshotAttributes",
        "ec2:DescribeSecurityGroups",
        "cloudformation:GetTemplateSummary",
        "rds>DeleteDBClusterSnapshot",
        "rds>CreateDBSecurityGroup",
        "rds:RebootDBInstance",
        "ec2:DescribeSecurityGroupReferences",
        "cloudformation:CreateStack",
        "ec2:DescribeVpcs",
        "rds:DescribeDBClusterSnapshots",
        "ec2>DeleteSecurityGroup",
        "rds:ModifyDBSubnetGroup",
        "ec2:DescribeStaleSecurityGroups"
    ],
    "Resource": "*"
},
{
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": "catalog-user:*",
    "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 **rds** in the Policy type search box and select the checkbox for the **rds-ra-LaunchPolicy** policy, and then choose **Next: Review**.
 - c. For **Role name**, type **rds-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 RDS Reference Blueprint** portfolio.
6. On the portfolio details page, expand the **Constraints** section, and then choose **Add constraints**.
7. For **Product**, choose **Amazon Relational Database Service (RDS)**, and for **Constraint type**, choose **Launch**. Choose **Continue**.
8. On the **Launch constraint** page, for **IAM role**, choose **rds-ra-LaunchRole**, and then choose **Submit**.

Service Catalog Product Launch

Once access has been provided to one or more end users the RDS reference blueprint products can be launched. To launch an RDS reference blueprint product the user (**ServiceCatalogEndUser**) needs to log into Service Catalog, select any one of the RDS Reference Blueprint Products (i.e. Amazon Relational Database Service (RDS) – MariaDB, MSSQL, MySQL or PostgreSQL) and click launch. The launch process will ask the end user for various details about how the RDS product will be configured. 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 RDS details back to the end user.

Service Catalog RDS Reference Blueprint Cleanup

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

1. Terminate all Service Catalog RDS 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

This project is licensed under the Apache 2.0 license – see the attached LICENSE.pdf file for details

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