Building VPC, S3, EC2, and RDS Products with AWS Service Catalog

**SPL-176 - Version 1.2.21**

© 2024 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. All trademarks are the property of their owners.

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at [*AWS Training and Certification*](https://support.aws.amazon.com/#/contacts/aws-training).

**Overview**

In this lab, you will leverage AWS Service Catalog to build a development Amazon Virtual Private Cloud (Amazon VPC) product. You will add Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS), and Amazon Simple Storage Service (Amazon S3) products into that VPC.

You will start by creating an *AWS Service Catalog portfolio* that contains four *products*. Each AWS Service Catalog product is backed by an AWS CloudFormation template, which is supplied as part of the lab. The first product is a VPC environment for end user development purposes. That VPC will be the destination you build your other products into. The second product will be an EC2 Linux instance. The third product will be a RDS MySQL database deployment. Lastly, you will build a S3 product which is restricted by IAM user, and IP address range of your VPC.

As an AWS Service Catalog Administrator, you will create and assign *Template and Launch constraints* for the AWS Service Catalog products that you create. You will set tags for both the products and portfolio that you will create, and assign AWS Identity and Access Management (IAM) users to be able to leverage the newly created portfolio.

You will be assigned two IAM users. The first IAM user (referred to as *admin* user) will be assigned the role of an AWS Service Catalog admin (not an AWS admin) which will only have the ability to create portfolios, and products in AWS Service Catalog, but no direct access to other services. The second IAM user (referred to as *developer* user) will mimic an end-user experience and will only have the ability to launch AWS Service Catalog products.

You will be leveraging four IAM roles throughout the lab. The IAM roles have their *trusted entity* set to the AWS Service Catalog service which allows AWS Service Catalog to launch AWS services in the environment such as VPC, EC2, RDS and S3. This mechanism is used instead of granting direct access to AWS services to the end-user (*developer* in this lab), but still give them the ability to launch those services in a *defined, governed and pre-approved way*.

You will be using four IAM roles: the **SC-VPC-ROLE** for VPC, the **SC-EC2-ROLE** for EC2, the **SC-RDS-ROLE** for RDS, and the **SC-S3-ROLE** for S3.

In a typical deployment scenario, the *admin* user will be an automation or deployments team or a business unit AWS admin who does not have control of the entire AWS environment. The *developer* user will be either a developer, a business owner or an operations team member who is not concerned with the underlying AWS infrastructure and is more of a consumer of the final AWS services. We call this model the *Consumer, Creator process*.

TOPICS COVERED

By the end of this lab, you will be able to:

* Create an AWS Service Catalog Portfolio.
* Create an AWS Service Catalog VPC Product.
* Create an AWS Service Catalog EC2 Product.
* Create an AWS Service Catalog RDS Product.
* Create an AWS Service Catalog S3 Product.
* Create an AWS Service Catalog Launch Constraints.
* Create an AWS Service Catalog Template Constraints.
* Launch your newly created AWS Service Catalog VPC Product.
* Launch your newly created EC2, RDS, and S3 Products within your VPC Product.

ICON KEY

Various icons are used throughout this lab to call attention to certain aspects of the guide. The following list explains the purpose for each one:

* **Caution:** Information of special interest or importance (not so important to cause problems with the equipment or data if you miss it, but it could result in the need to repeat certain steps).
* **Note:** A hint, tip, or important guidance.
* **Task complete:** A conclusion or summary point in the lab.

DURATION

This lab requires **120** minutes to complete.

**Start lab**

1. To launch the lab, at the top of the page, choose **Start lab**.

**Caution:** You must wait for the provisioned AWS services to be ready before you can continue.

1. To open the lab, choose **Open Console**.

You are automatically signed in to the AWS Management Console in a new web browser tab.

**WARNING:** **Do not change the Region unless instructed.**

COMMON SIGN-IN ERRORS

**Error: You must first sign out**



If you see the message, **You must first log out before logging into a different AWS account:**

* Choose the **click here** link.
* Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
* Choose **Open Console** again.

**Error: Choosing Start Lab has no effect**

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

* Add the lab domain name to your pop-up or script blocker’s allow list or turn it off.
* Refresh the page and try again.

**Task 1: Create your Portfolio**

AWS Service Catalog portfolios contain products created by an AWS Service Catalog administrator, and are a way to allow specific end users access to these products with certain constraints determined by the admin. In this task, you will use the *admin* user credentials to create a new AWS Service Catalog Portfolio. The admin user will only have access to create portfolios and products in the AWS Service Catalog and will have Read-Only access to the AWS platform.

To login as *admin*, you will logout as the normal Lab user and then use a special Login URL.

1. Sign out of the **AWS Management Console** by doing the following:

* On the top-right of the screen choose the user or role that you see.
* Choose **Sign out**
* Close the browser tab.

1. To the left of the lab instructions you are reading, copy the value of **LoginURL**.

This is the **IAM user sign-in link** that points to the AWS Account used by this lab.

1. Open a new web browser tab, paste the **LoginURL**, then press **Enter**.

**Caution:** If you receive an error that the page *does not support access for federated users*, then choose **logout here** and try again.

1. At the **AWS** sign-in page, enter the following:

**Caution:** If this login step is missed by the user then template will not get uploaded and lab will not proceed.

* **Account ID** In case its not pre-filled. Paste the value of **AWSAccountID** located to the left of these instructions.
* **IAM user name**

user-admin

* **Password** Paste the value of **AdministratorPassword** located to the left of these instructions.
* Choose **Sign in**

1. At the top of the AWS Management Console, in the search bar, search for and choose

Service Catalog

.

You are redirected to the AWS Service Catalog web console.

**Caution:** Make sure that the region listed in the console matches the **LabRegion** listed to the left of these instructions.

1. In the left navigation pane, choose **Portfolios** from the **Administration** sub-menu.

**Note:** If the left navigation pane is not visible, choose the menu icon  in the top-left corner to reveal it.

1. Choose **Create portfolio**
2. On the **Create portfolio** screen, configure the following:

* For **Portfolio name**, enter

My Portfolio

.

* For **Owner**, enter

Lab

.

1. Choose **Create**.

The portfolio is created. A banner message with text like, “ **Success** You successfully created portfolio My Portfolio” and a listing of all the account portfolios are displayed on the page.

1. Choose the link for **My Portfolio**.

This will display the details of your portfolio.

1. Choose the **Access (0)** tab.

This is where you can grant users access to the portfolio and allow them to launch the associated resources.

1. Choose **Grant access**.
2. Choose the **Users** tab and configure the following:

* Select  **user-developer**
* Select  **user-admin**

1. Choose **Grant Access**

A banner message with text like, “ **Success** You added access for user(s): user-admin and user-developer” is displayed.

1. Choose the **Tags (0)** tab.

This is where you create portfolio tags which help identify information regarding the portfolio which you can access, and leverage programmatically.

1. On the **Add tags specific to this portfolio** screen, configure the following:

**Tag 1:**

* **Key** Enter

Portfolio

* **Value** Enter

My Portfolio

1. Choose **Add tag**

A banner message with text like, “ **Success** Portfolio was updated” is displayed.

**Tag 2:**

* **Key** Enter

Provider

* **Value** Enter

Infrastructure Team

1. Choose **Add tag**

A banner message with text like, “ **Success** Portfolio was updated” is displayed.

1. Choose the **TagOptions (0)** tab.
2. Choose **TagOptions library**.

This is where you create tags associated to products being launched.

1. On the **TagOptions library** page, under the **Create new TagOption** section, configure the following tags:

**Tag 1:**

* **Key** Enter

CostCenter

* **Value** Enter

Operations

1. Choose **Add**

A banner message with text like, “ **Success** Key value pair CostCenter: Operations has been added to your TagOptions library” is displayed.

**Tag 2:**

* **Key** Enter

CostCenter

* **Value** Enter

Development

1. Choose **Add**

A banner message with text like, “ **Success** Key value pair CostCenter: Development has been added to your TagOptions library” is displayed.

There should now be two tags in the **TagOptions** section of the page.

1. In the left navigation pane, choose **Portfolios** from the **Administration** sub-menu.
2. Choose the link for **My Portfolio**.
3. Choose the **TagOptions (0)** tab and configure the following:

* Select  *Key: CostCenter Value: Development*
* Select  *Key: CostCenter Value: Operations*

1. Choose **Actions**, and then choose **Associate TagOption**.

A pop-up window is displayed.

1. In the **Associate TagOptions** window, choose **Ok**.

A banner message with text like, “ **Success** You have associated 2 TagOptions: CostCenter: Operations and CostCenter: Development” is displayed.

**Task complete:** You have successfully created a portfolio, associated users to the portfolio, and added tagging for both the portfolio and products.

**Task 2: Create your products**

AWS Service Catalog products are a powerful way of grouping services into a full solution for your end users so that they do not need access to the AWS Management Console. AWS Service Catalog products are based on CloudFormation templates. They can be a single service, or a combination of services. In this task, you will create AWS Service Catalog products to associate to your portfolio.

TASK 2.1: CREATE AN AMAZON VPC PRODUCT

The first product you will create is an Amazon VPC which will be the foundational network that you will associate with your EC2 and RDS products.

1. Open the context (right-click) menu for [VPC.template](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-176/v1.2.21.prod-af58ff9b/scripts/VPC.template" \t "_blank), and download it to your computer by choosing *Save Link As…* option.
2. From the **My Portfolio** screen, choose the **Products (0)** tab.
3. Choose **Create product**.

The **Create product** page is displayed.

1. Under the **Product details** section, configure the following:

* For **Product type**, select **CloudFormation**.
* For **Product name**, enter

VPC

.

* For **Description**, enter

Development VPC Product

.

* For **Owner**, enter

Lab

.

1. Under the **Version details** section, configure the following:

* For **Version source**, select **Use a template file**.
* For **Upload a Cloudformation template file**, choose **Choose file** and browse to the **VPC.template** file that you downloaded earlier.
* For **Version name**, enter

1

.

1. Under the **Support details** section, configure the following:

* For **Email contact**, enter

support@example.com

.

1. Choose **Create product**.

A banner message with text like, “ **Success** Successfully added prod-xxxxxx to portfolio” is displayed.

**Note:** You can choose the refresh  button to confirm that the product was successfully added to portfolio.

TASK 2.2: CREATE A LINUX EC2 PRODUCT

The second product you will create is a Linux EC2 instance that allows your end user to choose a destination subnet and security group upon provisioning. The CloudFormation template this product is based on also ensures which specific AMI to use.

1. Open the context (right-click) menu for [LINUX.template](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-176/v1.2.21.prod-af58ff9b/scripts/LINUX.template" \t "_blank), and download it to your computer by choosing *Save Link As…* option.
2. From the **My Portfolio** screen, choose the **Products (1)** tab.
3. Choose **Create product**.

The **Create product** page is displayed.

1. Under the **Product details** section, configure the following:

* For **Product type**, select **CloudFormation**.
* For **Product name**, enter

EC2

.

* For **Description**, enter

Linux Instance Product

.

* For **Owner**, enter

Lab

.

1. Under the **Version details** section, configure the following:

* For **Version source**, select **Use a template file**.
* For **Upload a Cloudformation template file**, choose **Choose file** and browse to the **LINUX.template** file that you downloaded earlier.
* For **Version name**, enter

1

.

1. Under the **Support details** section, configure the following:

* For **Email contact**, enter

support@example.com

.

1. Choose **Create product**.

A banner message with text like, “ **Success** Successfully added prod-xxxxxx to portfolio” is displayed.

**Note:** You can choose the refresh  button to confirm that the product was successfully added to portfolio.

TASK 2.3: CREATE AN RDS NODE PRODUCT

The third product you will create is an Amazon RDS MySQL database instance. The product allows your developer user the ability to choose a DB Security Group and MySQL password upon provisioning. The CloudFormation template this product is based on also ensures the DB instance has 5GB of storage.

1. Open the context (right-click) menu for [RDS.template](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-176/v1.2.21.prod-af58ff9b/scripts/RDS.template" \t "_blank), and download it to your computer by choosing *Save Link As…* option.
2. From the **My Portfolio** screen, choose the **Products (2)** tab.
3. Choose **Create product**.

The **Create product** page is displayed.

1. Under the **Product details** section, configure the following:

* For **Product type**, select **CloudFormation**.
* For **Product name**, enter

RDS Node

.

* For **Description**, enter

RDS Node Product

.

* For **Owner**, enter

Lab

.

1. Under the **Version details** section, configure the following:

* For **Version source**, select **Use a template file**.
* For **Upload a Cloudformation template file**, choose **Choose file** and browse to the **RDS.template** file that you downloaded earlier.
* For **Version name**, enter

1

.

1. Under the **Support details** section, configure the following:

* For **Email contact**, enter

support@example.com

.

1. Choose **Create product**.

A banner message with text like, “ **Success** Successfully added prod-xxxxxx to portfolio” is displayed.

**Note:** You can choose the refresh  button to confirm that the product was successfully added to portfolio.

TASK 2.4: CREATE A S3 PRODUCT

The fourth product you will create is a S3 bucket which will allow access to a specific IAM user, and IP range.

1. Open the context (right-click) menu for [S3.template](https://us-west-2-tcprod.s3.us-west-2.amazonaws.com/courses/spl-176/v1.2.21.prod-af58ff9b/scripts/S3.template), and download it to your computer by choosing *Save Link As…* option.
2. From the **My Portfolio** screen, choose the **Products (3)** tab.
3. Choose **Create product**.

The **Create product** page is displayed.

1. Under the **Product details** section, configure the following:

* For **Product type**, select **CloudFormation**.
* For **Product name**, enter

S3

.

* For **Description**, enter

Product that creates a S3 Bucket

.

* For **Owner**, enter

Lab

.

1. Under the **Version details** section, configure the following:

* For **Version source**, select **Use a template file**.
* For **Upload a Cloudformation template file**, choose **Choose file** and browse to the **S3.template** file that you downloaded earlier.
* For **Version name**, enter

1

.

1. Under the **Support details** section, configure the following:

* For **Email contact**, enter

support@example.com

.

1. Choose **Create product**.

A banner message with text like, “ **Success** Successfully added prod-xxxxxx to portfolio” is displayed.

**Note:** You can choose the refresh  button to confirm that the product was successfully added to portfolio.

**Task complete:** You have successfully created VPC, S3, EC2, and RDS products for your end users to use.

**Task 3: Create Launch Constraints for your Products**

AWS Service Catalog acts on behalf of your end users as a proxy allowing for access to resources without giving specific user access to an entire service. AWS Service Catalog leverages IAM roles to allow for access to services associated by the admin user. In this task, you will create *launch constraints* for your products. Your *admin* user has access to create AWS Service Catalog portfolios and products and has read-only access to all other services. Your *admin* user is not able to launch VPC, S3, EC2, or RDS services. Therefore, you will need to assign an **IAM role** to each AWS Service Catalog product that allows the AWS Service Catalog to create the products and their underlying services (VPC, S3, EC2, RDS) on behalf of the end user – these are referred to as launch constraints.

TASK 3.1: ADD A LAUNCH CONSTRAINT FOR YOUR VPC PRODUCT

1. In the left navigation pane, choose **Portfolios** from the **Administration** sub-menu.
2. Choose the link for **My Portfolio**.
3. Choose the **Constraints (0)** tab.
4. Choose **Create constraint**.

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **VPC** from the drop-down menu.
* For **Constraint type**, select **Launch**.

1. Under the **Launch constraint** section, configure the following:

* For **Method**, select **Select IAM role**.
* For **IAM role**, select **SC-VPC-ROLE** from the drop-down menu.

1. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

TASK 3.2: ADD A LAUNCH CONSTRAINT FOR YOUR LINUX INSTANCE PRODUCT

1. Choose **Create constraint**

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **EC2** from the drop-down menu.
* For **Constraint type**, select **Launch**.

1. Under the **Launch constraint** section, configure the following:

* For **Method**, select **Select IAM role**.
* For **IAM role**, select **SC-EC2-ROLE** from the drop-down menu.

1. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

TASK 3.3: ADD A LAUNCH CONSTRAINT FOR YOUR RDS PRODUCT

1. Choose **Create constraint**

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **RDS Node** from the drop-down menu.
* For **Constraint type**, select **Launch**.

1. Under the **Launch constraint** section, configure the following:

* For **Method**, select **Select IAM role**.
* For **IAM role**, select **SC-RDS-ROLE** from the drop-down menu.

1. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

TASK 3.4: ADD A LAUNCH CONSTRAINT FOR YOUR S3 PRODUCT

1. Choose **Create constraint**.

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **S3** from the drop-down menu.
* For **Constraint type**, select **Launch**.

1. Under the **Launch constraint** section, configure the following:

* For **Method**, select **Select IAM role**.
* For **IAM role**, select **SC-S3-ROLE** from the drop-down menu.

1. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

**Task complete:** You have successfully created launch constraints so that your end users can launch the products you created via IAM roles without needing access to those services in the AWS management console.

**Task 4: Launch your VPC Product**

In this task, you will launch the development VPC product. This will create a VPC that has a Public and Private Subnet with associated internet gateway. This is a useful example on allowing your end consumers of products in your portfolio to dynamically create a VPC without allowing them access to make changes through the management console.

You will do this as the *developer* user, so you will sign out of the management console, and then sign in as *developer*.

1. Sign out of the **AWS Management Console** by doing the following:

* On the top-right of the screen choose the user name **user-admin @**.
* Choose **Sign out**.
* Close the browser tab.

1. To the left of these instructions, copy the value of **LoginURL**.
2. Open a new web browser tab, paste the **LoginURL**, then press **Enter**.
3. At the **AWS** sign-in page, enter:

* **Account ID** In case its not pre-filled. Paste the value of **AWSAccountID** located to the left of these instructions.
* **IAM user name**

user-developer

* **Password** Paste the value of **AdministratorPassword** located to the left of these instructions.
* Choose **Sign in**

1. At the top of the AWS Management Console, in the search bar, search for and choose

Service Catalog

.

You are now signed in as the *developer* user and directed to the AWS Service Catalog web console.

**Caution:** Make sure that the region listed in the console matches the **LabRegion** listed to the left of these instructions.

1. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
2. From the list of **Products**, select  **VPC** and choose **Launch product**.
3. For **Provisioned product name**, enter

myVPC

.

1. Under the **Parameters** section, configure the following:

* For **RegionAZ1Name**, elect **us-west-2a** from the drop-down menu.
* For **RegionAZ2Name**, select **us-west-2b** from the drop-down menu.
* Leave the rest of the values set to default for **VPC Configuration**, **Public Subnet Configuration** and **Private Subnet Configuration**.

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**
2. **Wait** until the Status of your VPC product changes from **Under Change** to **Available**.

**Note:** You can choose the refresh  button to update the status.

1. On the **Provisioned product details** screen, scroll down and choose the **Events (1)** tab.
2. Under **Events**, select the ARN of the launched CloudFormation stack under **Output value**, which would have a value like **arn:aws:cloudformation:us-west-2:xxxxxxxx:stack/SC-xxxxx**, to review the VPC CloudFormation template output values.

A new tab opens to the AWS CloudFormation service and you see the CloudFormation stack for your VPC provisioned product.

1. Choose the **Outputs** tab, and copy the **Value** of the following items to your text editor:

* **DevLinuxSecurityGroup**
* **Subnet1Public**
* **DevRDSSecurityGroup**

**Task complete:** You have successfully launched a development VPC environment as an end user.

**Task 5: Launch your S3 Product**

In this task, you will launch the development S3 product. This will create a S3 bucket that will give specific access to your developer user ARN, and to a specific IP range. This is a useful example on allowing your end consumers of products in your portfolio to dynamically create a S3 bucket without allowing them access to make changes through the management console.

1. Return to the AWS Service Catalog Management Console.
2. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
3. From the list of **Products**, select  **S3** and choose **Launch product**.

* For **Provisioned product name**, enter

mys3NUMBER

.

* Replace **NUMBER** with a long random number.

1. Under the **Parameters** section, configure the following:

* For **S3UserARN**, paste the value of **DeveloperUserARN** located to the left of the instructions you are reading
* For **S3AccessCIDR**, enter

10.3.0.0/16

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**
2. **Wait** until the Status of your S3 product changes from **Under Change** to **Available**.

**Note:** You can choose the refresh  button to update the status.

**Task complete:** You have successfully launched the S3 product which creates a bucket that only has access from your end user, and to a specific IP range.

**Task 6: Launch your EC2 Product**

In this task, you will launch the development EC2 product. This will create a Linux EC2 instance that will have a specific security group and subnet. This is a useful example on allowing your end consumers of products in your portfolio to dynamically create a EC2 instance without allowing them access to make changes through the management console.

1. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
2. From the list of **Products**, select  **EC2** and choose **Launch product**.
3. For **Provisioned product name**, type

myEC2

.

1. Under the **Parameters** section, configure the following:

* Verify **AWSAmiId** is set to its default value

/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86\_64-gp2

.

* Verify **ApplicationName** is set to

LinuxInstance

.

* Verify **InstanceType** is selected as

t2.small

.

* For **SecurityGroup**, select the value of **DevLinuxSecurityGroup** that you noted as part of step 92.
* For **Subnet**, select the value of **Subnet1Public** that you noted as part of step 92.

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**.
2. **Wait** until the Status of your EC2 product changes from **Under Change** to **Available**.

**Note:** You can choose the refresh  button to update the status.

**Task complete:** You have successfully launched an EC2 linux instance into your development VPC environment.

**Task 7: Launch your RDS Product**

In this task, you will launch the development RDS product. This will create a RDS node that will have a specific security group and subnet group. This is a useful example on allowing your end consumers of products in your portfolio to dynamically create a RDS node without allowing them access to make changes through the management console.

1. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
2. From the list of **Products**, select  **RDS Node** and choose **Launch product**.
3. For **Provisioned product name**, type

myRDS

.

1. Under the **Parameters** section, configure the following:

* Verify **ApplicationName** is set to

RDS

.

* For **DBSecurityGroup**, select the value of **DevRDSSecurityGroup** that you noted as part of step 92.
* For **DBSubnetGroupName**, select **devdbsubnet** from the drop-down menu.
* For **MySQLPassword**, paste the value of **AdministratorPassword** located to the left of these instructions.

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**.
2. **Wait** until the Status of your RDS product changes from **Under Change** to **Available**.

**Note:**

* You can choose the refresh  button to update the status.
* It may take up to 10 minutes for the RDS product to change its status to **Available**.

**Task complete:** You have successfully launched an RDS Node into your development VPC environment.

**Task 8: Add Template Constraints**

Template Constraints are what makes AWS Service Catalog flexible in applying guardrails for your end users, limiting their access allows for higher security and less mistakes. In this task, you will add template constraints. A template constraint can be applied to any CloudFormation template input parameter, limiting its input values to either one value or a list of values. In this example, you will limit the security group and subnet that can be selected for the Linux Instance and RDS products created earlier in the lab. Earlier you noticed that the end-user (*user-developer*) *can* select any security group and subnet in the VPC.

Once you configure the constraint, the *developer* user will *not* be able to select any security groups or subnets in the VPC and will be forced to select the single security group and subnet that you assigned in the Template Constraints.

You will sign in again as the *admin* user to perform this update.

1. Sign out of the **AWS Management Console** by doing the following:

* On the top-right of the screen choose the user name **user-developer @**
* Choose **Sign out**.
* Close the browser tab.

1. To the left of these instructions, copy the value of **LoginURL**.
2. Open a new web browser tab, paste the **LoginURL**, then press **Enter**.
3. At the **AWS** sign-in page, enter:

* **IAM user name**

user-admin

* **Password** Paste the value of **AdministratorPassword** located to the left of these instructions.
* Choose **Sign in**

1. At the top of the AWS Management Console, in the search bar, search for and choose

Service Catalog

.

You are now signed in as the *admin* user and directed to the AWS Service Catalog web console.

**Caution:** Make sure that the region listed in the console matches the **LabRegion** listed to the left of these instructions.

1. In the left navigation pane, choose **Portfolios** from the **Administration** sub-menu.

**Note:** If the left navigation pane is not visible, choose the menu icon  in the top-left corner to reveal it.

1. Choose the link for **My Portfolio**.
2. Choose the **Constraints (4)** tab.

TASK 8.1: CREATING EC2 PRODUCT CONSTRAINTS

In this task you will be adding two rules under one template constraint for the Linux EC2 Product, limiting the Security Group, and Subnet choices for end users.

1. Choose **Create constraint**.

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **EC2** from the drop-down menu.
* For **Constraint type**, select **Template**.

1. Under the **Template constraint** section, configure the following:

* For **Method**, select **Interactive editor**.
* For **Rule name**, enter

Linux-Security-Group-Constraint

.

* For **Rule description**, enter

Restricts by Security Group

.

* For **Product version**, select

1

 from the drop-down menu.

* For **Parameter**, select  Security Group.
* For **Parameter value**, select the checkbox  next to the Linux Security Group ID that you copied during step 92 earlier in the lab referencing to the value of **DevLinuxSecurityGroup**.

1. Choose **Add rule**.
2. Choose **Add additional rule** and configure the following:

* For **Rule name**, enter

Linux-Subnet-Constraint

.

* For **Rule description**, enter

Restricts to Public Subnet

.

* For **Product version**, select

1

 from the drop-down menu.

* For **Parameter**, select  Subnet.
* For **Parameter value**, select the checkbox  next to the Public Subnet ID that you copied during step 92 earlier in the lab referencing to the value of **Subnet1Public**.

1. Choose **Add rule**.
2. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

TASK 8.2: CREATING RDS PRODUCT CONSTRAINTS

In this step you will be adding two rules under one template constraint for the RDS Node Product limiting the Security Group and Subnet choices for end users.

1. Choose **Create constraint**.

The **Create constraint** page is displayed.

1. Under the **Create constraint** section, configure the following:

* For **Product**, choose **RDS Node** from the drop-down menu.
* For **Constraint type**, select **Template**.

1. Under the **Template constraint** section, configure the following:

* For **Method**, select **Interactive editor**.
* For **Rule name**, enter

RDS-Subnet-Group-Constraint

.

* For **Rule description**, enter

Restricts to Private DB Subnet Group

.

* For **Product version**, select

1

 from the drop-down menu.

* For **Parameter**, select  DBSubnetGroupName
* For **Parameter value**, select the checkbox  next to **devdbsubnet**.

1. Choose **Add rule**.
2. Choose **Add additional rule** and configure the following:

* For **Rule name**, enter

RDS-Security-Group-Constraint

.

* For **Rule description**, enter

Restricts by RDS Security Group

.

* For **Product version**, select

1

 from the drop-down menu.

* For **Parameter**, select  DBSecurityGroup.
* For **Parameter value**, select the checkbox  next to the RDS Security Group ID that you copied during step 92 earlier in the lab referencing to the value of **DevRDSSecurityGroup**.

1. Choose **Add rule**.
2. Choose **Create**.

A banner message with text like, “ **Success** Successfully created constraint” is displayed.

**Task complete:** You have successfully created template constraints to ensure your end users do not launch resources in any other environment but the development VPC environment with specific subnets and security groups.

**Task 9: Test your New Constraint**

In this task, you will verify that the template constraint you created in the previous task is working, and that the end-user (*developer* user) is constrained to the specific security group you added in the template constraint.

You will sign out of the console, and then sign in again as the *developer* user.

1. Sign out of the **AWS Management Console** by doing the following:

* On the top-right of the screen choose **user-admin @**.
* Choose **Sign out**.
* Close the browser tab.

1. To the left of these instructions, copy the value of **LoginURL**.
2. Paste the **LoginURL** into a new browser tab.
3. At the **AWS** sign-in page, enter:

* **Account ID** In case its not pre-filled. Paste the value of **AWSAccountID** located to the left of these instructions.
* **IAM user name**

user-developer

.

* **Password** Paste the value of **AdministratorPassword** located to the left of these instructions.

1. Choose **Sign in**.
2. At the top of the AWS Management Console, in the search bar, search for and choose

Service Catalog

.

You are now signed in as the *developer* user and directed to the AWS Service Catalog web console.

**Caution:** Make sure that the region listed in the console matches the **LabRegion** listed to the left of these instructions.

1. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
2. From the list of **Products**, select  **EC2** and choose **Launch product**.
3. For **Provisioned product name**, type

MyEC2-2

.

1. Under the **Parameters** section, configure the following:

* Verify **AWSAmiId** is set to its default value

/aws/service/ami-amazon-linux-latest/amzn-ami-hvm-x86\_64-gp2

* Verify **ApplicationName** is set to

LinuxInstance

* Verify **InstanceType** is selected as

t2.small

.

* **SecurityGroup** Here you see that the value is pre-selected to the one that was set in the template constraint as part of Task 8.1.
* For **Subnet**, here you see that the value is pre-selected to the one that was set in the template constraint as part of Task 8.1.

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**.
2. **Wait** until the Status of your EC2 product changes from **Under Change** to **Available**.

**Note:** You can choose the refresh  button to update the status.

1. From the left navigation pane, choose **Products** under the **Provisioning** sub-menu.
2. From the list of **Products**, select  **RDS Node** and choose **Launch product**.
3. For **Provisioned product name**, type

myRDS2

.

1. Under the **Parameters** section, configure the following:

* Verify **ApplicationName** is set to

RDS

.

* For **DBSecurityGroup**, here you see that the value is pre-selected to the one that was set in the template constraint as part of Task 8.2.
* For **DBSubnetGroupName**, here you see that the value is pre-selected to the one that was set in the template constraint as part of Task 8.2.
* For **MySQLPassword**, paste the value of **AdministratorPassword** located to the left of these instructions.

1. Under the **Manage tags** section, configure:

* Verify **Key** is set to *CostCenter*.
* Assign **Value** as *Development*.

1. Choose **Launch product**.
2. **Wait** until the Status of your RDS product changes from **Under Change** to **Available**.

**Note:**

* You can choose the refresh  button to update the status.
* It may take up to 10 minutes for the RDS product to change its status to **Available**.

**Task complete:** You have successfully verified that the template constraint you created in the previous task is working, and that the end-user (*developer* user) is constrained to the specific security group you added in the template constraint.

**Conclusion**

You have successfully completed the lab and have done the following:

* Created an AWS Service Catalog Portfolio.
* Created an AWS Service Catalog VPC Product.
* Created an AWS Service Catalog EC2 Product.
* Created an AWS Service Catalog RDS Product.
* Created an AWS Service Catalog S3 Product.
* Created an AWS Service Catalog Launch Constraints.
* Created an AWS Service Catalog Template Constraints.
* Launched your newly created AWS Service Catalog VPC Product.
* Launched your newly created EC2, RDS, and S3 Products within your VPC Product.

**End lab**

Follow these steps to close the console and end your lab.

1. Return to the **AWS Management Console**.
2. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.
3. Choose **End lab** and then confirm that you want to end your lab.

**Additional Resources**

* [AWS Service Catalog Documentation](https://docs.aws.amazon.com/servicecatalog/index.html).
* [AWS Service Catalog resources](https://aws.amazon.com/servicecatalog/resources/?service-catalog-blogs.sort-by=item.additionalFields.createdDate&service-catalog-blogs.sort-order=desc).
* [AWS Service Catalog Partners](https://aws.amazon.com/servicecatalog/partners/).

For more information about AWS Training and Certification, see [*https://aws.amazon.com/training/*](https://aws.amazon.com/training/).

*Your feedback is welcome and appreciated.*  
If you would like to share any feedback, suggestions, or corrections, please provide the details in our [*AWS Training and Certification Contact Form*](https://support.aws.amazon.com/#/contacts/aws-training).