Security Governance at Scale

Student Lab Guide

Contents

Lab 0: AWS Control Tower Setup Instructions

As part of the training, the instructor walks through hands-on labs. To enhance the lecture portion of this course, you have a lab document that describes the steps. You can work through the labs after the course using your own account. Optionally, you can follow along with the instructor using your own environment. If you decide to follow along, keep the following points in mind:

- Instructors cannot support lab questions after the training.
- You assume responsibility for the resources that you create. To decommission resources, follow the steps in Lab 4.
- You must add a payment method.
- Costs might be incurred if you do not decommission resources properly.

For help, contact AWS Support at: https://aws.amazon.com/premiumsupport

Email accounts

To run the labs, you will need four email accounts. You must be the owner of the accounts and have full access to their inboxes.

Ensure that the email accounts are not associated with existing AWS accounts. We recommend that you use email aliases. Multiple email providers support that feature. The four email addresses are used for the following purposes:

- AWS Control Tower management account (example, your-email+ct-management@your-email-provider.com)
- AWS Control Tower audit account (example, your-email+ct-audit@your-email-provider.com)
- AWS Control Tower log archive account (example, your-email+ct-log-archive@your-email-provider.com)
- Labs example account: Martha Rivera (example, your-email+martha-rivera@your-email-provider.com)

  Note
  Ensure that you are the owner of the email addresses. You will create actual AWS accounts with them.

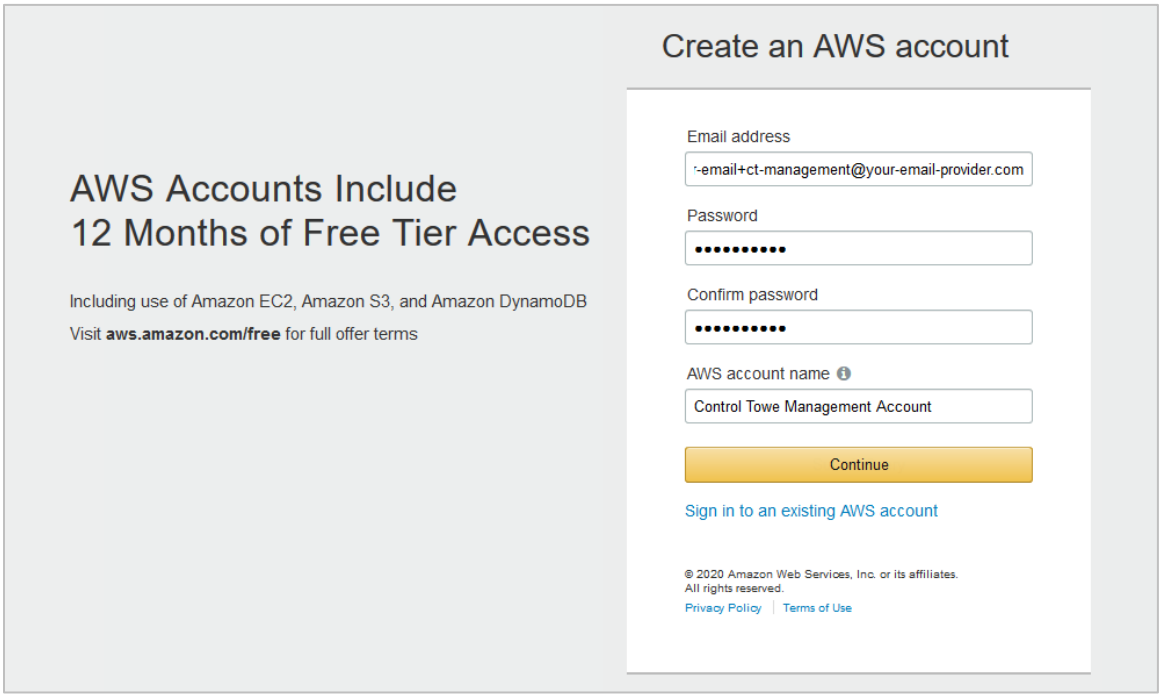
Use dedicated accounts that you create for this training, which you can decommission at the end of the training. We recommend that you don't use existing production, test, or development accounts that your organization might have provided. You should also not use accounts that might be in use for other purposes. Lab 4 provides instructions on how to decommission accounts and their provisioned resources.

AWS account

Sign up for an AWS account using the alias you chose for your AWS Control Tower management account. To sign up for an account, you need to provide a payment method. During the labs, you will provision resources, and some of the resources will incur costs. You will be provided with instructions on how to delete those resources to avoid additional costs. Ensure that you can log in and that your account is activated, so you are prepared to take the labs.

To sign up for an AWS account:

1. Go to https://portal.aws.amazon.com/billing/signup?type=enterprise#/start.
2. To sign up, use your management account email address, and give it a name. Remember the password you create, because this account will be your root AWS account.
3. Complete the Free Tier form, click ***Continue***, and follow the steps.



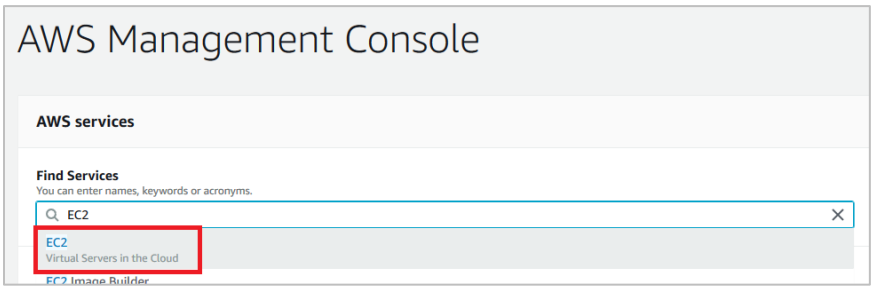
If you need assistance while setting up your AWS account, refer to the documentation at https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account. After you create your account and can log in, proceed to the next section to learn how to set up AWS Control Tower's landing zone.

AWS Control Tower landing zone

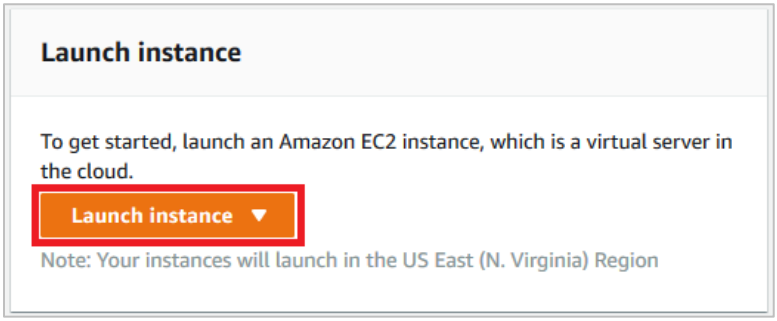
To set up AWS Control Tower, your account needs certain security scores. Depending on the domain you chose for your email addresses, those scores are lower than what is required for the AWS Control Tower setup.

We recommend that you spin up an Amazon Elastic Compute Cloud (Amazon EC2) Free Tier instance for 20 minutes and then terminate it. That should increase your account's security containment scores. Use the following steps to spin up a Free Tier Amazon EC2 instance:

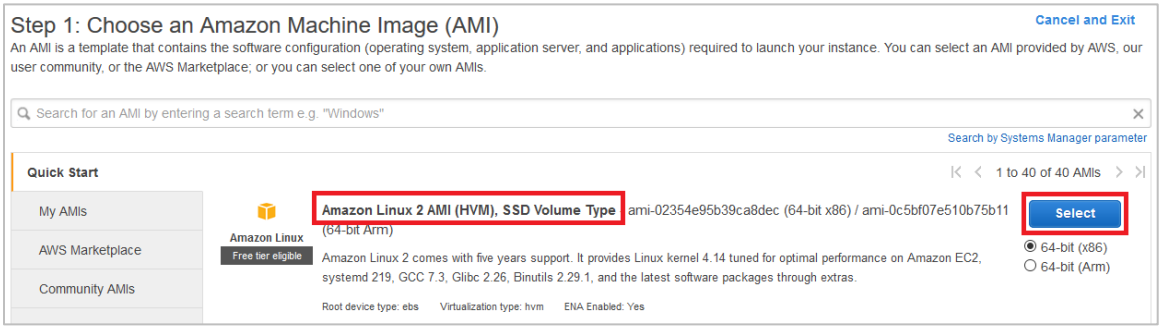
1. Open the AWS Management Console using your management account
2. Navigate to the Amazon EC2 service dashboard.



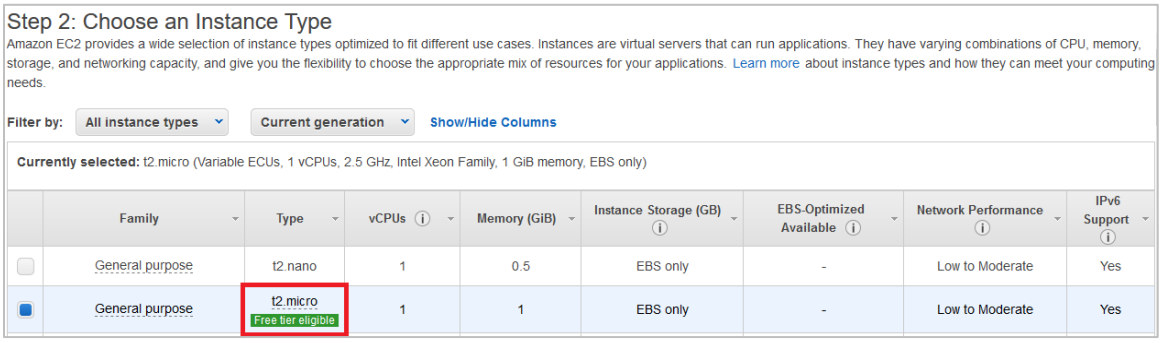
3. Choose ***Launch instance***.



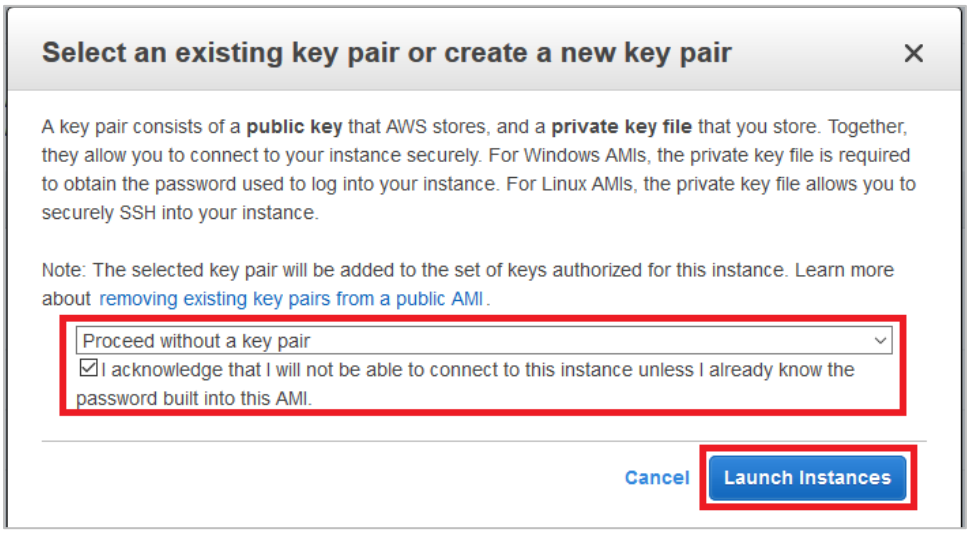
4. Choose ***Amazon Linux 2 AMI*** and then choose ***Select***.



5. Next, make sure your instance type is ***Free tier eligible***, as shown in the next image.

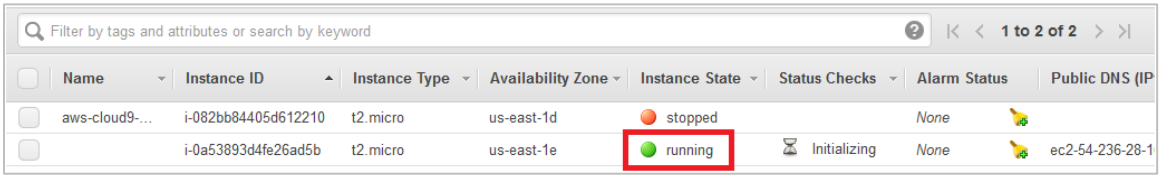


6. Choose ***Review*** and launch.
7. You need to decide your key pair. Select ***Proceed without a key pair***, select the ***I acknowledge that I will not be able to connect to this instance unless I already know the password build into this AMI*** box, and then choose ***Launch instances***.

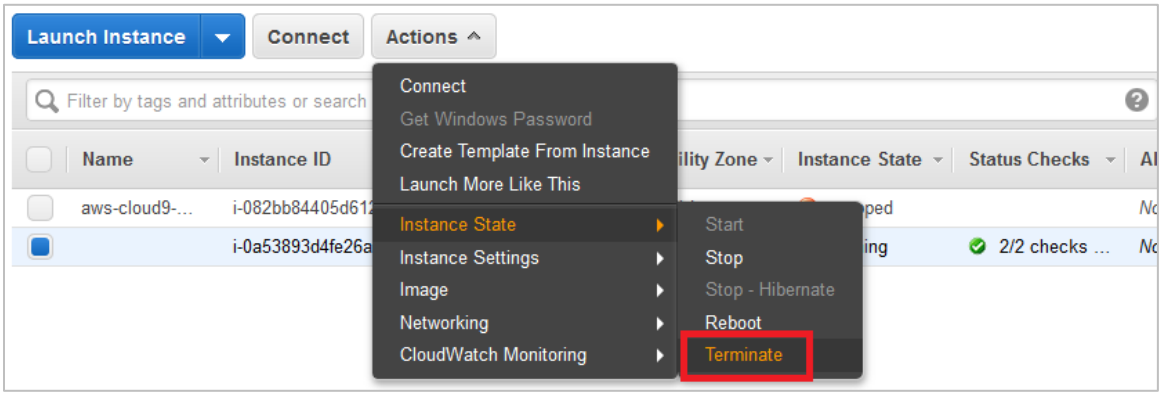


You will receive the message that states, *Your instances are now launching.*

8. Choose ***View instances***.
9. After your Instance State shows as ***running***, wait 20 minutes.



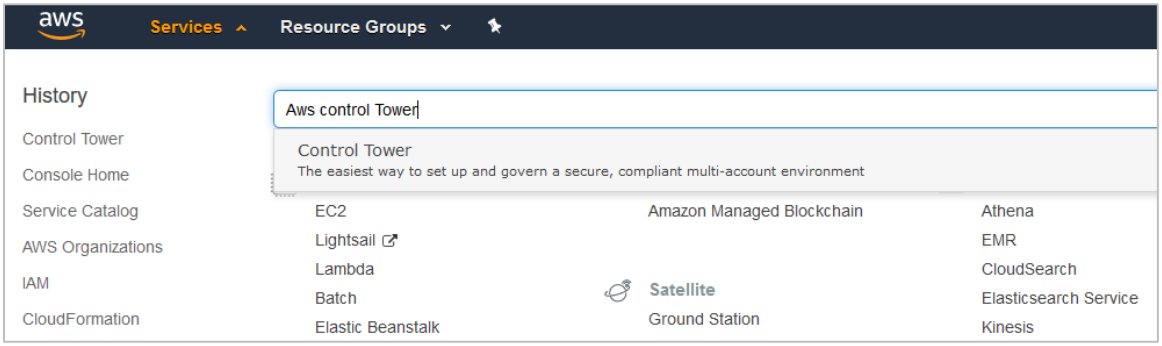
10. After 20 minutes, choose ***Instance State***, and then choose ***Terminate*** to terminate your instance.



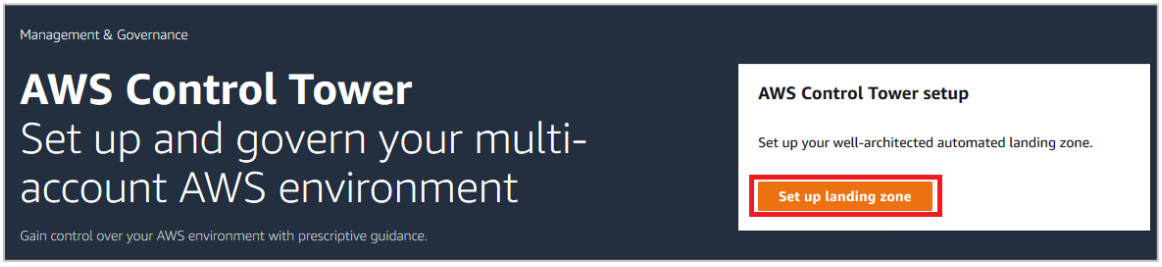
Next, you will set up an AWS Control Tower landing zone.

11. Use your management account to log in to the AWS Management Console.

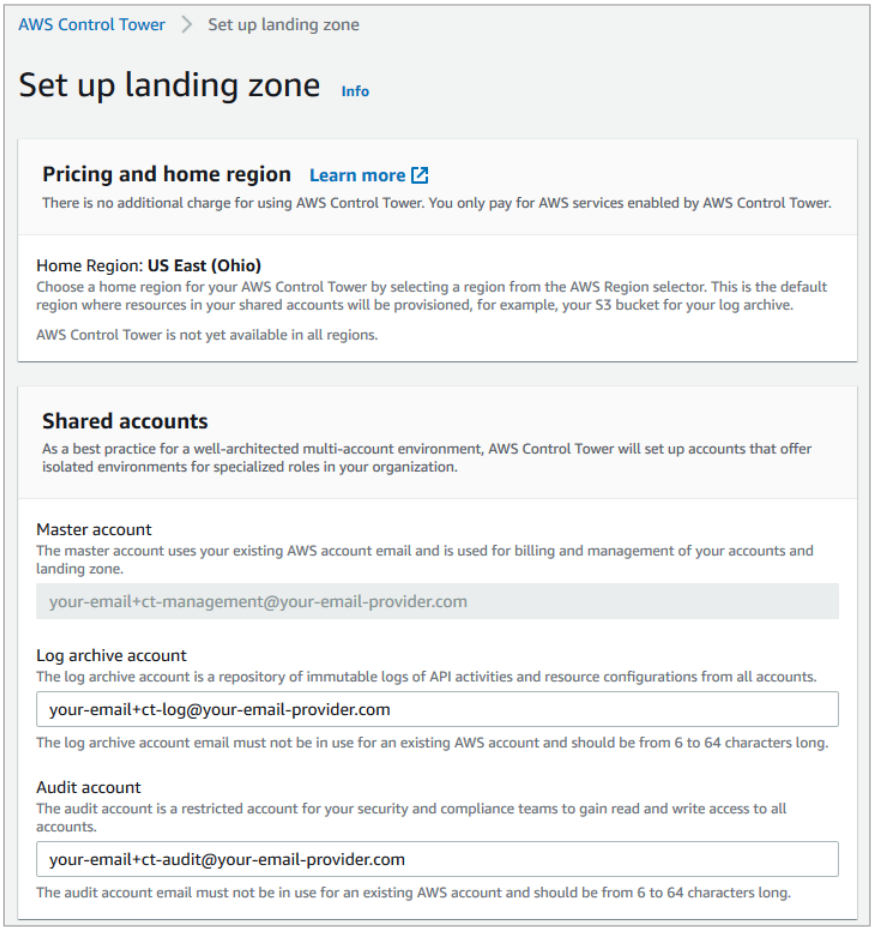
12. Navigate to the AWS Control Tower.



13. On the AWS Control Tower page, choose ***Set up landing zone***.

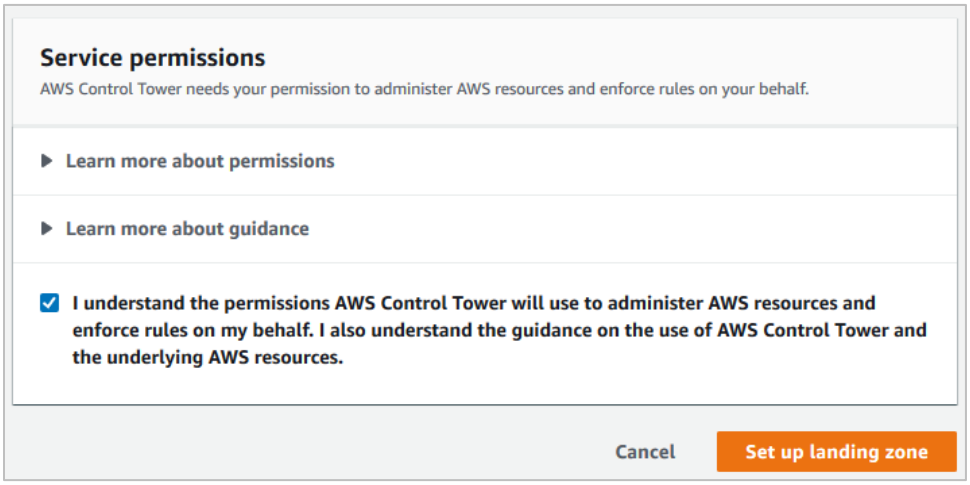


14. Fill in the form using the Audit and Log Archive email aliases:

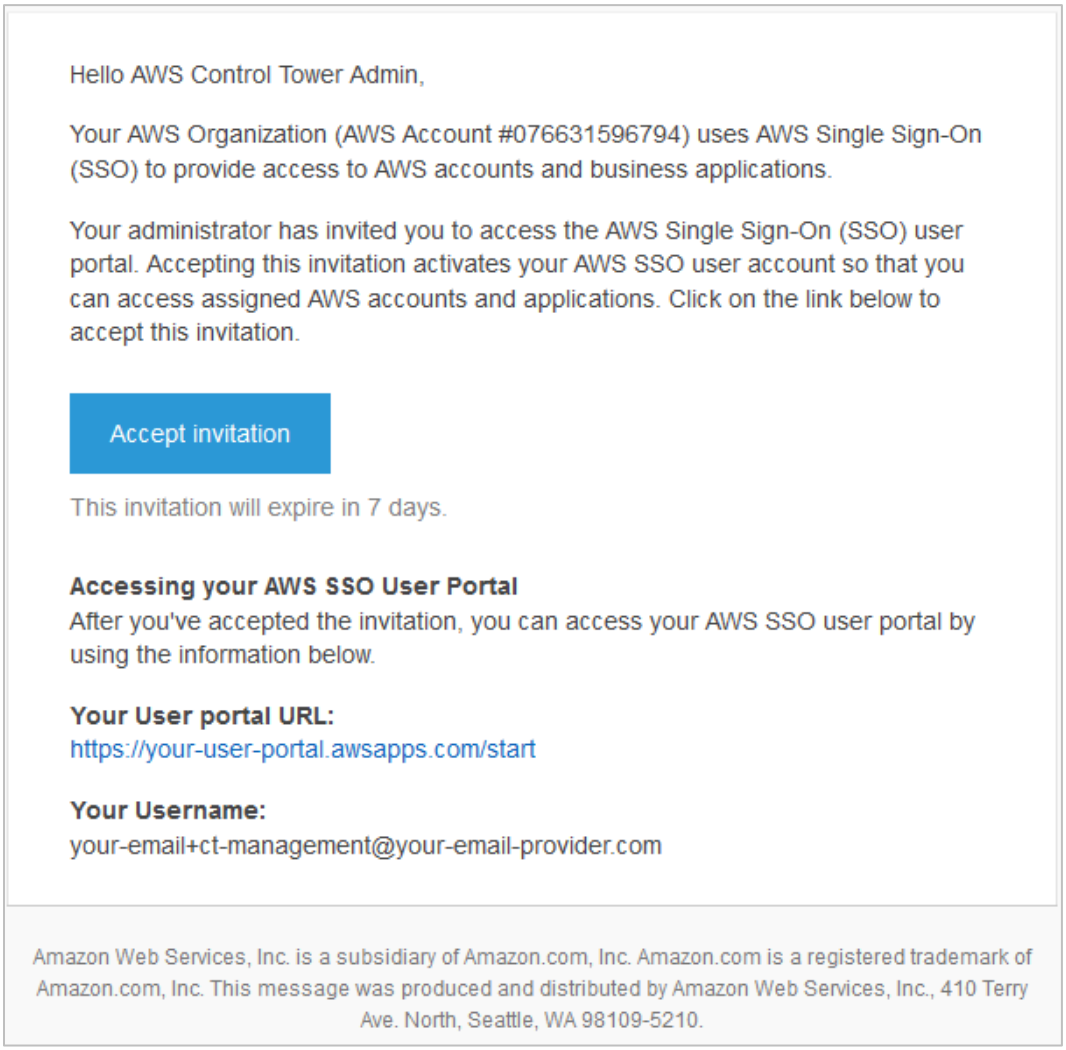


15. Select the ***I understand the permissions AWS Control Tower will use to administer AWS resources and enforce rules on my behalf*** box. Make sure you read and understand the Service permissions.

16. Choose ***Set up landing zone***.



17. You will receive a series of emails to the shared email aliases that you provided. Look for an email similar to the following example.



Hello AWS Control Tower Admin,

Your AWS Organization (AWS Account #076631596794) uses AWS Single Sign-On (SSO) to provide access to AWS accounts and business applications.

Your administrator has invited you to access the AWS Single Sign-On (SSO) user portal. Accepting this invitation activates your AWS SSO user account so that you can access assigned AWS accounts and applications. Click on the link below to accept this invitation.

Accept invitation

This invitation will expire in 7 days.

Accessing your AWS SSO User Portal
After you've accepted the invitation, you can access your AWS SSO user portal by using the information below.

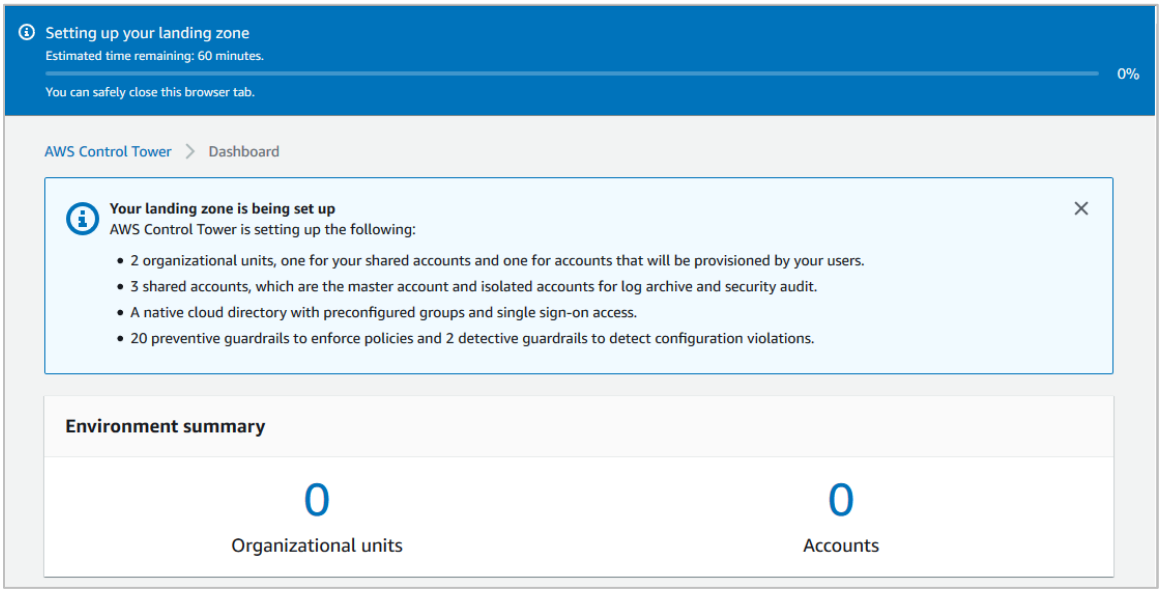
Your User portal URL:
https://your-user-portal.awsapps.com/start

Your Username:
your-email+ct-management@your-email-provider.com

Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web Services, Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.

18. Choose ***Accept invitation***. Provide a password for AWS Single Sign-On. Record it for later use.
19. You will also receive a separate email to verify your email account. Follow the instructions in the email to verify your email address.

Note
The landing zone setup process can take up to 90 minutes.



After the landing zone is ready, you should be able to complete the labs and follow the instructions provided by your instructor during the training.

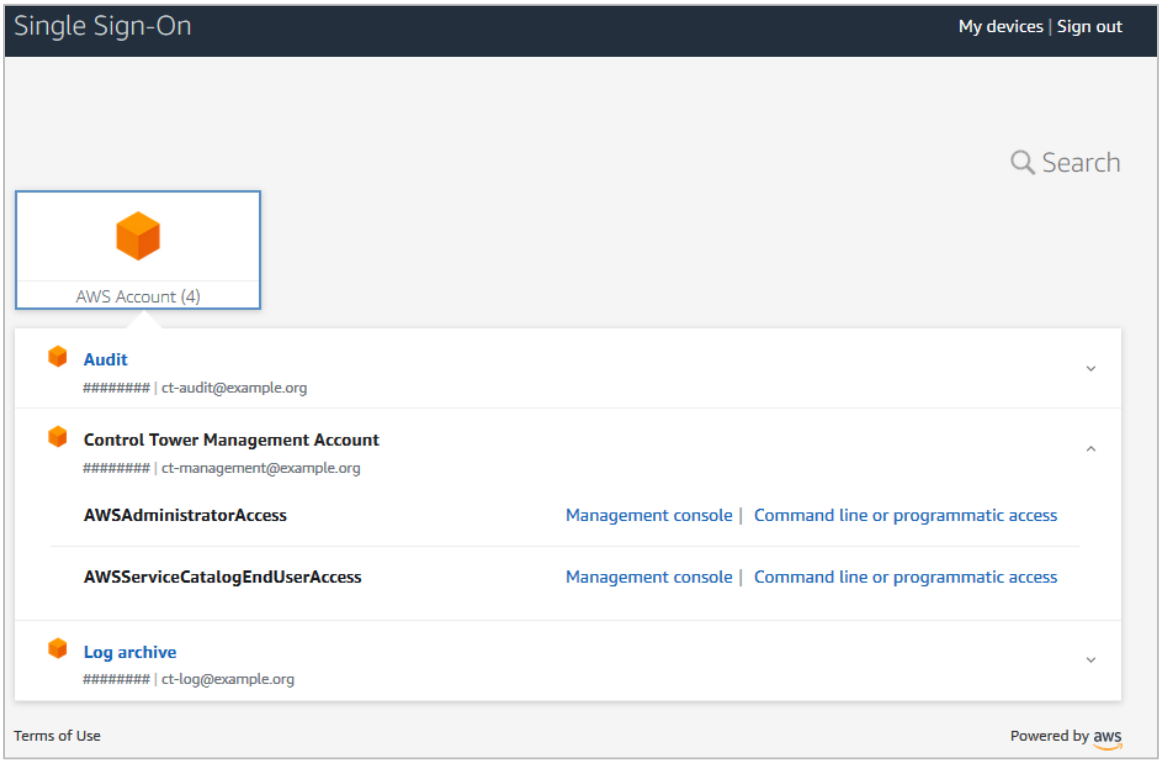
Note
We recommend that you follow the instructions in the next section to enable AWS Cost Explorer.

AWS Cost Explorer

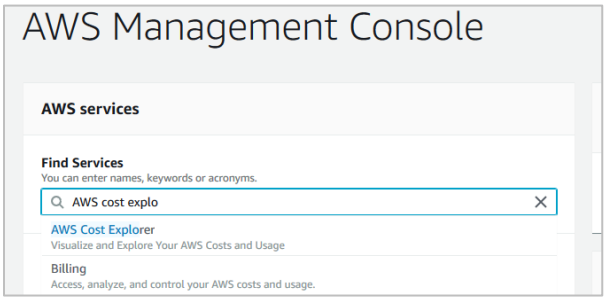
In this section, you will learn how to check account costs from the management account. Use this section after you set up AWS Control Tower landing zone. You will need your user portal URL for AWS SSO. You must also know the different roles that are available for each account when you log in to the AWS SSO portal.

You can access a multi-account cost overview with AWS Cost Explorer:

1. Using SSO, log in to the AWS Control Tower management account with the ***AWSAdministratorAccess*** role.

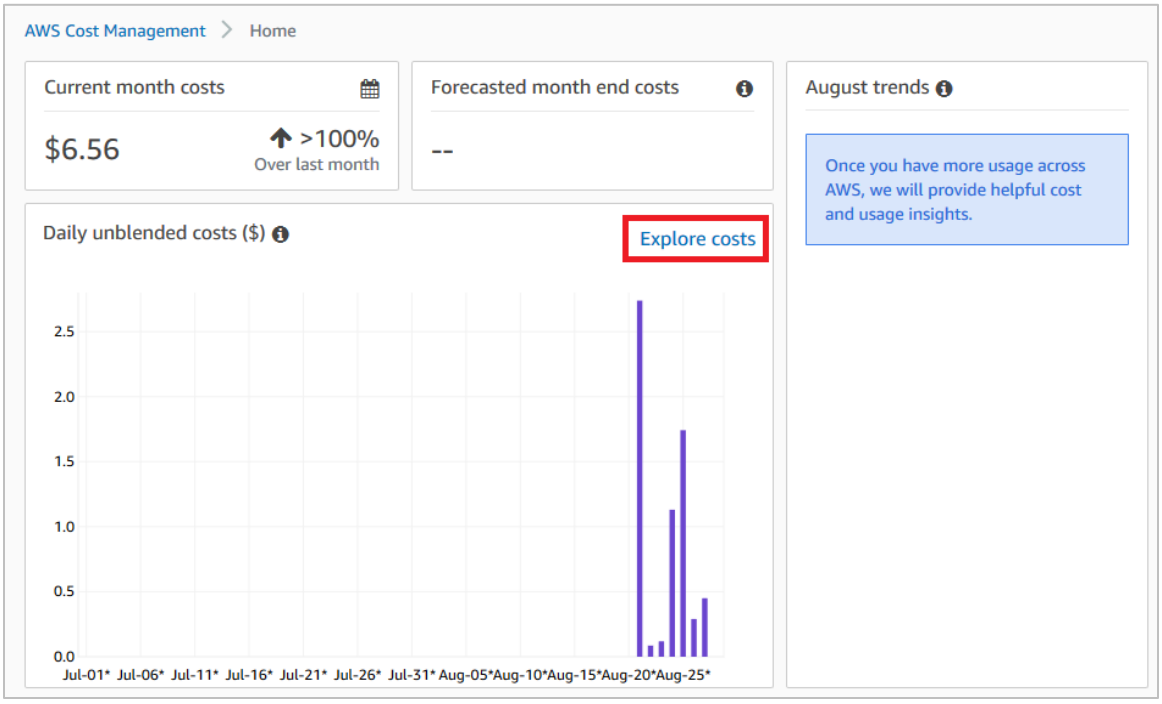


2. In the AWS Management Console, choose ***AWS Cost Management***, and then choose ***AWS Cost Explorer***.

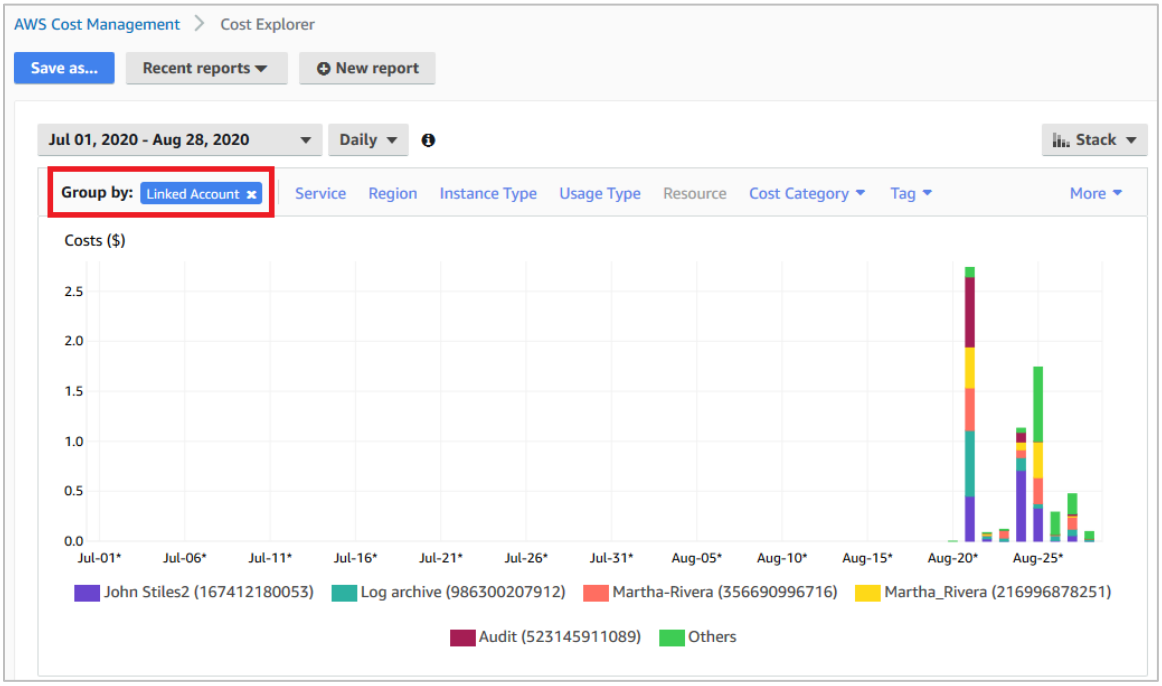


Note
AWS Cost Explorer is not enabled by default. After you enable the service, it takes some time to receive reports.

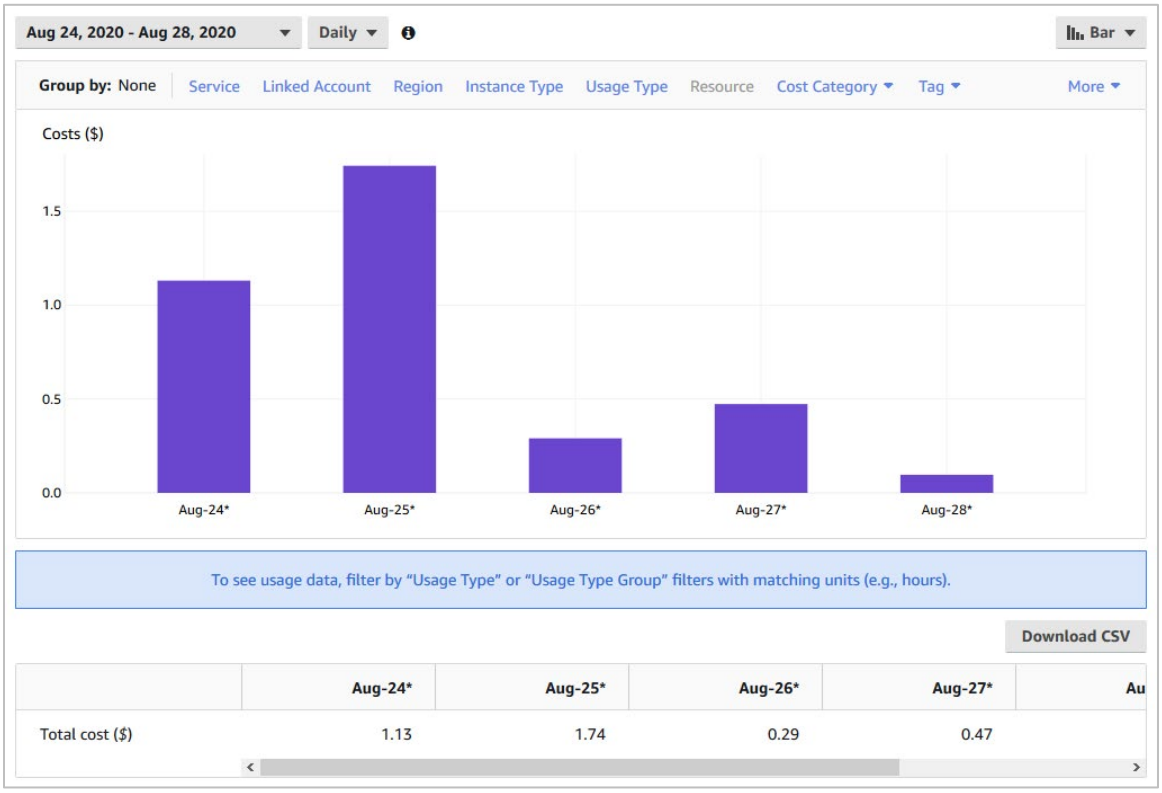
3. On the AWS Cost Explorer dashboard, find the ***Daily unblended costs*** section, and choose ***Explore costs***.



4. Choose the ***Linked Account*** filter in the ***Group by*** tab. This shows the breakdown charges for each account.



The following example shows incurred costs for the accounts used in these labs. You can see that the cost is below $1.00 daily.



Important

This is an average estimate. Your costs might vary. Make sure that you follow the decommission steps in Lab 4, to avoid additional costs.

Lab 1: AWS Control Tower Basic Tasks

You work for a biotechnology organization tasked with developing a vaccine for an existing virus. Multiple teams are involved, and they must be able to work in an independent and agile manner. While the main focus is to develop an effective vaccine, the organization needs to ensure that the teams operate securely, and protect personally identifiable information (PII) or Protected Health Information (PHI). In addition, processes and workloads must be compliant with different regulations, such as the Health Insurance Portability and Accountability Act of 1996 (HIPAA).

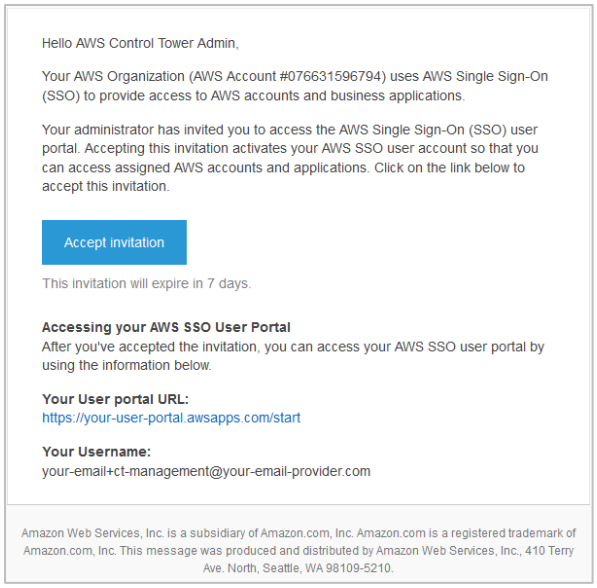
You need a solution that scales as the amount of data and number of processes grow exponentially. Your organization has decided to implement governance at scale as a framework. The goal is to adopt best practices while managing AWS resources, using them in an efficient, agile, secure, and compliant way. This lab starts after you create a landing zone. In this lab, you will:

- Log in and access the AWS console and AWS SSO
- Create an organizational unit (OU) and enable a guardrail
- Launch an AWS account using Account Factory

Ensure that you use the same Region when performing this lab and the previous lab. Regions must be the same when you set up your landing zone using AWS Control Tower.

Task 1: Log in and access the AWS console and AWS Single Sign-On

When you first set up AWS Control Tower, you must use a management account to establish your landing zone. After the landing zone is ready, a series of emails are sent to the default provisioned accounts for audit, log archive, and management account. You should receive two emails: one to confirm your account and a second message that provides your AWS SSO URL. The following image shows an example of the second email message.



Hello AWS Control Tower Admin,

Your AWS Organization (AWS Account #076631596794) uses AWS Single Sign-On (SSO) to provide access to AWS accounts and business applications.

Your administrator has invited you to access the AWS Single Sign-On (SSO) user portal. Accepting this invitation activates your AWS SSO user account so that you can access assigned AWS accounts and applications. Click on the link below to accept this invitation.

Accept invitation

This invitation will expire in 7 days.

Accessing your AWS SSO User Portal
After you've accepted the invitation, you can access your AWS SSO user portal by using the information below.

Your User portal URL:
https://your-user-portal.awsapps.com/start

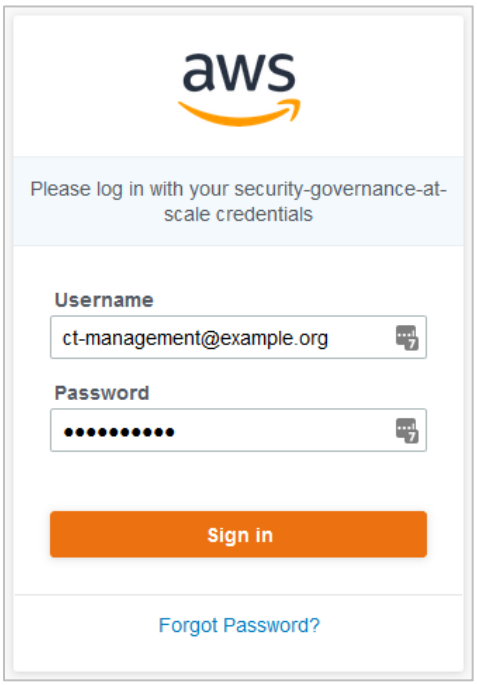
Your Username:
your-email+ct-management@your-email-provider.com

Amazon Web Services, Inc. is a subsidiary of Amazon.com, Inc. Amazon.com is a registered trademark of Amazon.com, Inc. This message was produced and distributed by Amazon Web Services, Inc., 410 Terry Ave. North, Seattle, WA 98109-5210.

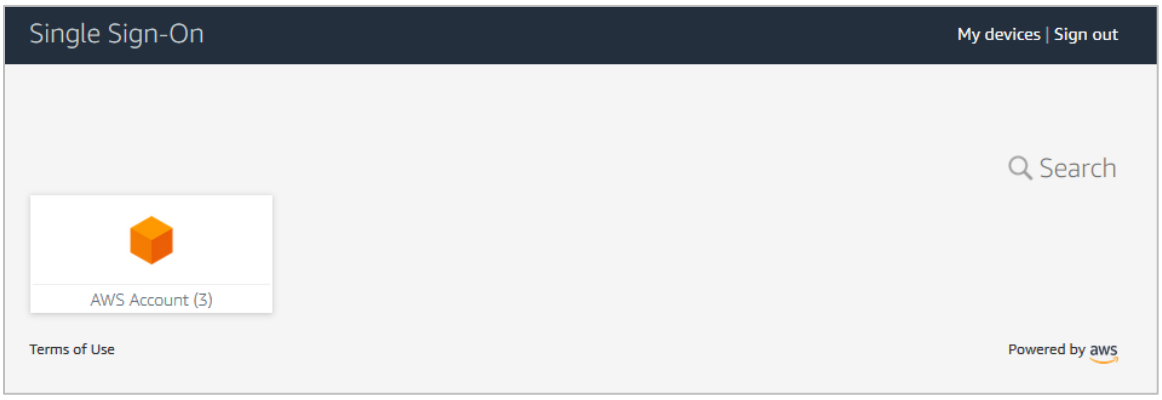
Note
Write down (and potentially bookmark) your user portal's URL. You will need the URL to log in AWS SSO for this lab and others.

To sign in:

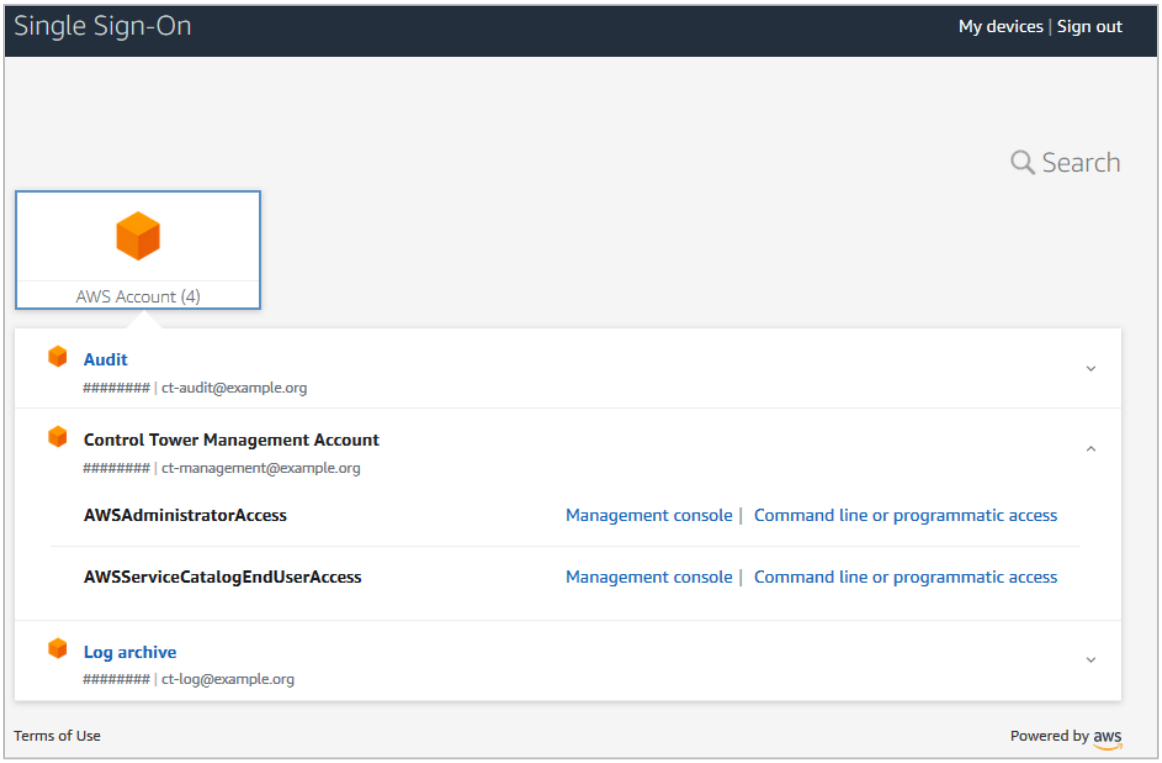
1. Accept the invitation.
2. Go to your user portal URL. The log-in screen should be similar to the following image:



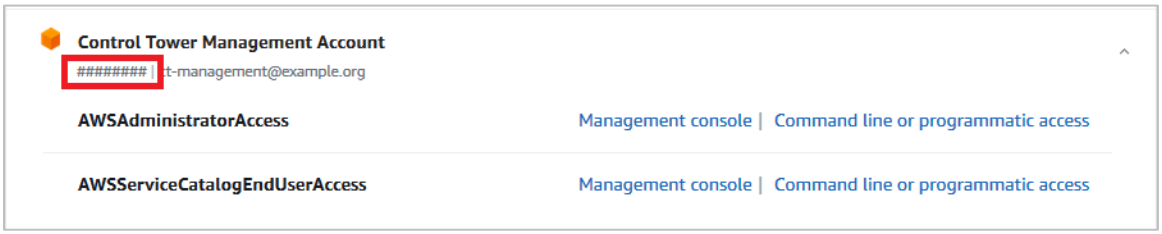
3. Enter your management account credentials, and choose ***Sign in***.
4. On see the ***Single Sign-On*** page, expand your ***AWS Accounts***, and choose your management account.



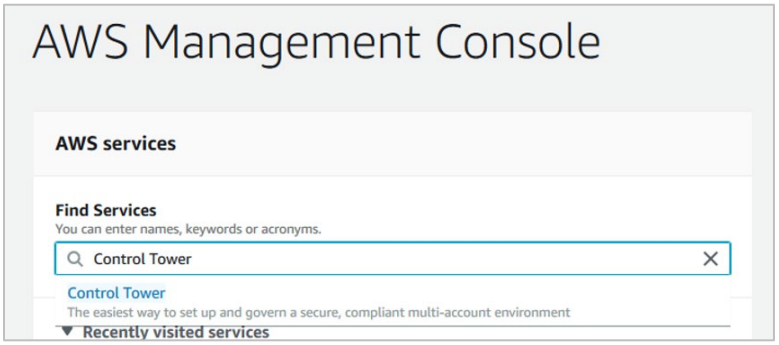
5. Choose the ***Management console***.



Note
Write down the management account ID number. You will use it later in the lab.



6. Enter ***AWS Control Tower***, and then choose the service to go to the landing page.



Task 2: Create an organizational unit (OU) and enable a guardrail

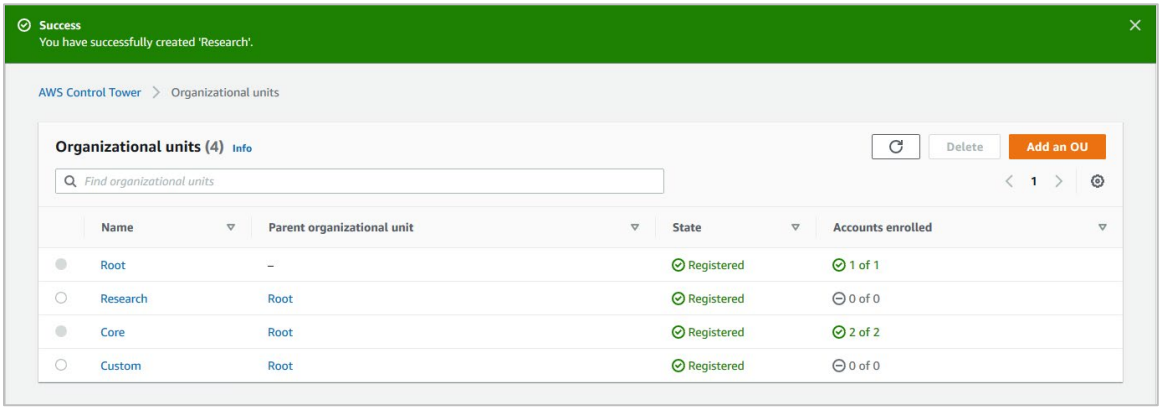
AWS Organizations is an account management service that lets you consolidate multiple AWS accounts into an organization that you create and centrally manage. In AWS Control Tower, organizations help you to centrally manage billing; access, compliance, security; and shared resources across your member AWS accounts. Accounts are grouped into logical groups, called organizational units (OUs).

Create an OU

To create an OU from the AWS Control Tower dashboard:

1. Log in to AWS Control Tower using your management account.
2. In the navigation pane, choose ***Organizational units***.
3. Choose ***Add an OU,*** enter the name *Research*, and then choose ***Add***.

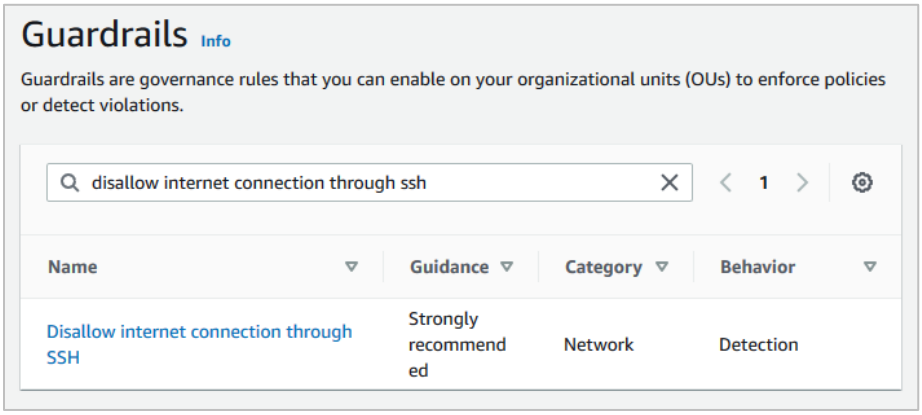
A success notification displays on the page when the account is ready.



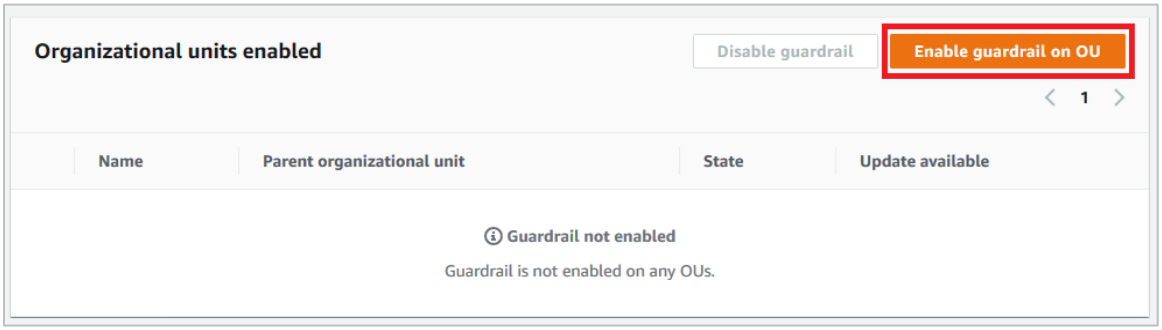
Enable a strongly recommended guardrail

A guardrail is a high-level rule that provides ongoing governance for your AWS environment. AWS Control Tower provides three categories of guardrails: *mandatory*, *strongly recommended*, and *elective*. To enable a strongly recommended guardrail on the new OU:

1. In the navigation pane, choose ***Guardrails***.
2. In the search field, enter ***Disallow internet connection through SSH***.

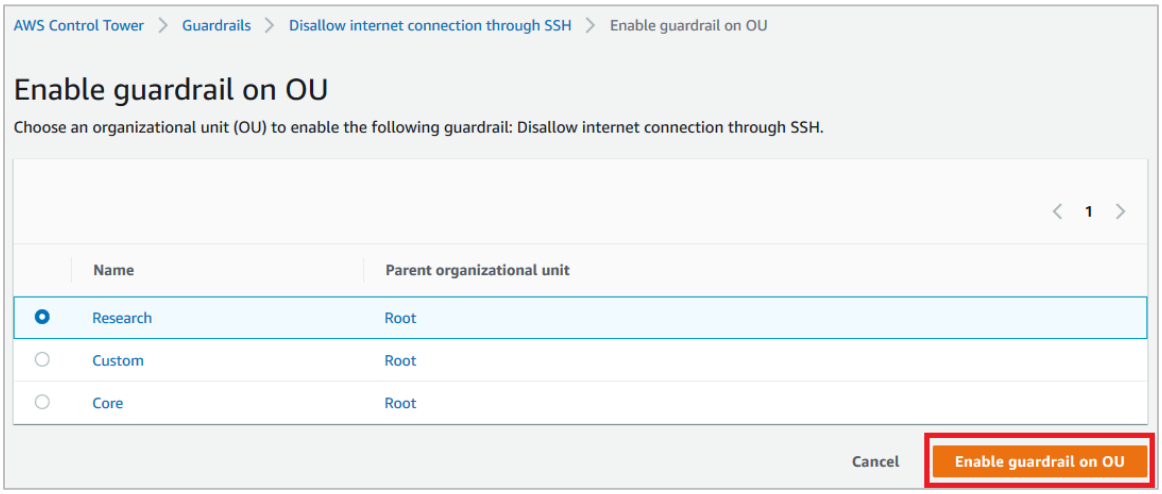


3. Choose the guardrail, and scroll to the Organizational units enabled section.
4. Choose ***Enable Guardrail on OU***.

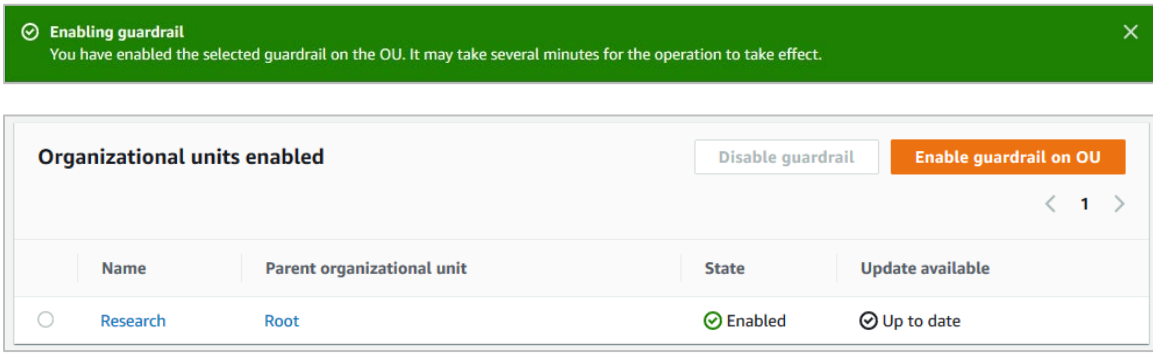


A new page lists the names of your OUs.

5. Select the ***Research*** OU, and chose ***Enable guardrail on OU***.



After the guardrail is enabled, you see a success notification at the top of the page.



A guardrail applies to an entire OU, and every AWS account in the OU is affected by the guardrail. When users perform work in any AWS account in your landing zone, they are subject to the guardrails that govern their account's OU.

Explore guardrail types

Through guardrails, AWS Control Tower implements *preventive* or *detective* controls that help you govern your resources and monitor compliance across groups of AWS accounts. Guardrails enable you to express your policy intentions.

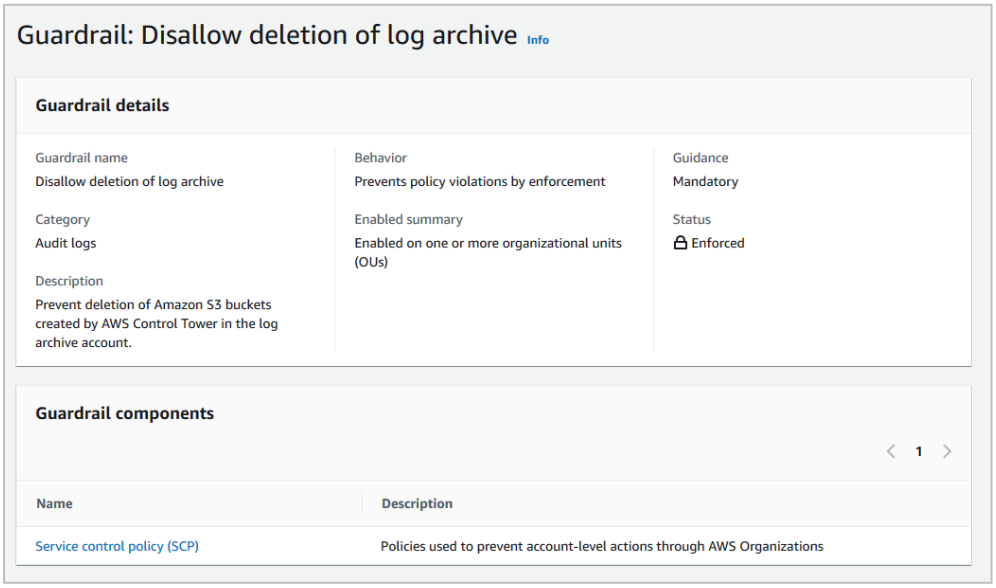
View a preventative guardrail

A preventive guardrail ensures that your accounts maintain compliance, because it disallows actions that lead to policy violations. To view a preventative guardrail:

1. In the navigation pane, choose ***Guardrails***.
2. In the search field, enter ***Disallow deletion of log archive***.
3. Select the guardrail.

   If you enable this preventative guardrail on an OU, you can determine whether a user has attempted to delete an Amazon Simple Storage Service (Amazon S3) bucket created by AWS Control Tower in the log archive account under that OU.

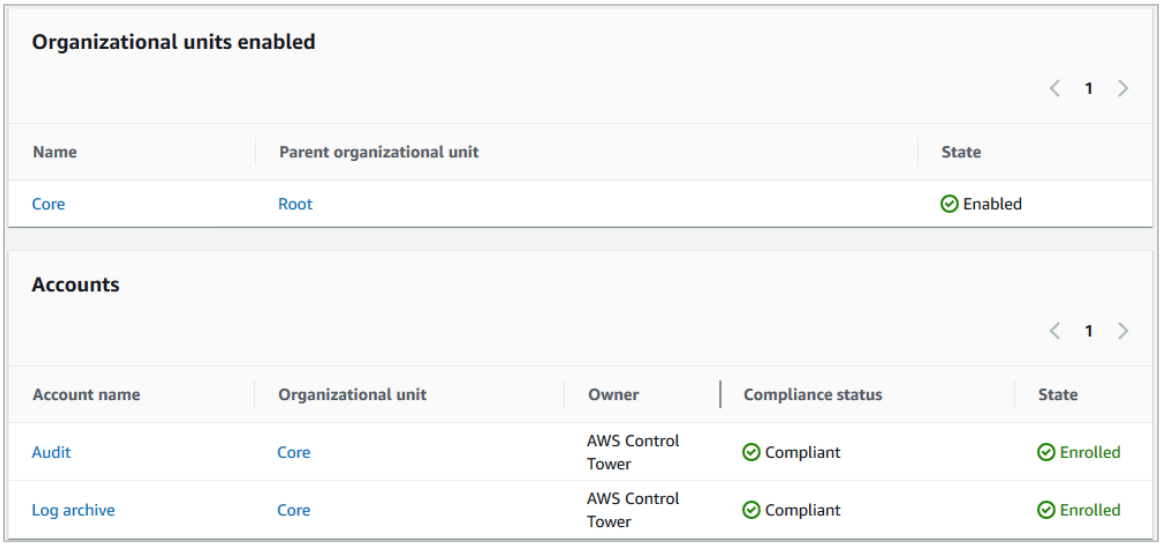
   The status of a preventive guardrail is either *enforced* or *not enabled*. Notice the enforced status in the following image.



Preventive guardrails are implemented using service control policies (SCPs), which are part of AWS Organizations.

4. In the Guardrail components section, choose ***Service control policy (SCP)*** to view the policy.

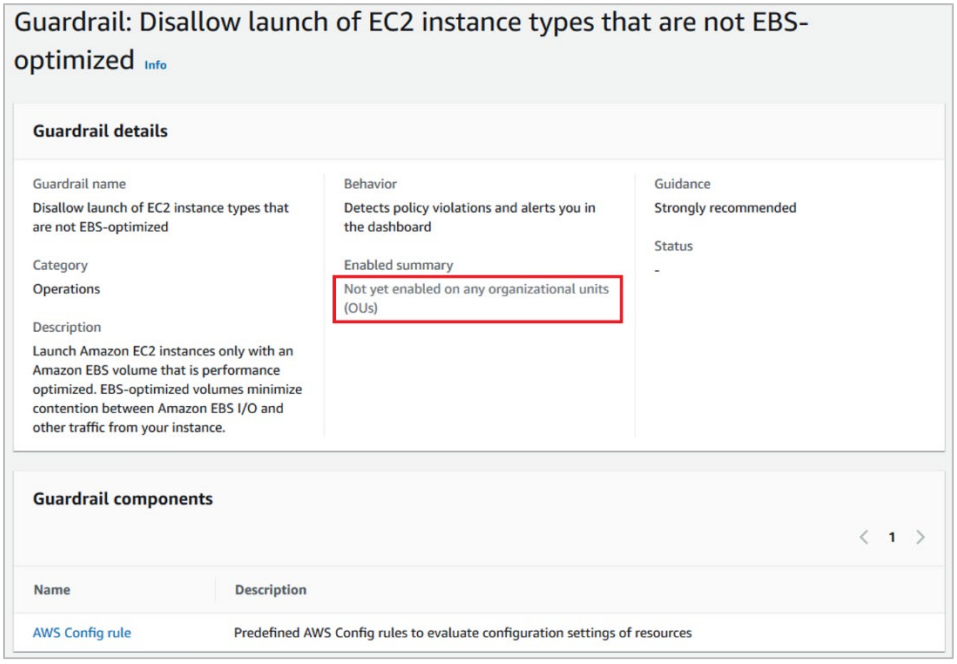
5. Close the Service control policy (SCP) window.
6. Scroll to the Accounts section. You can see that the Log archive and Audit OUs have the guardrail enabled.



View a detective guardrail

A detective guardrail detects noncompliance of resources in your accounts, such as policy violations, and provides alerts through the dashboard. To view a detective guardrail:

1. In the navigation pane, click ***Guardrails***.
2. In the search field, enter ***Disallow launch of EC2 instance types that are not EBS-optimized.***
3. Select the guardrail. The status of a detective guardrail is *clear*, *in violation*, or *not enabled*. Notice that the guardrail in the following image is not enabled.

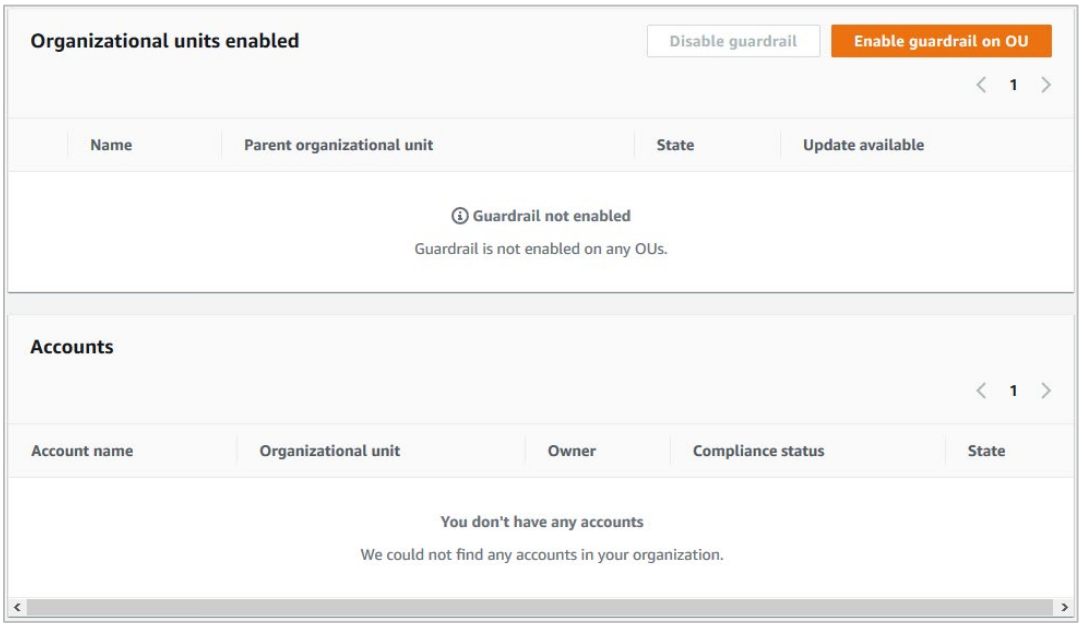


The detective guardrails are implemented using AWS Config rules and AWS Lambda functions.

4. In the Guardrail components section, choose ***AWS Config rule*** to view the rule.

5. Close the AWS Config rule window.

6. Scroll to the Accounts section. The guardrail has not been enabled on any OUs.



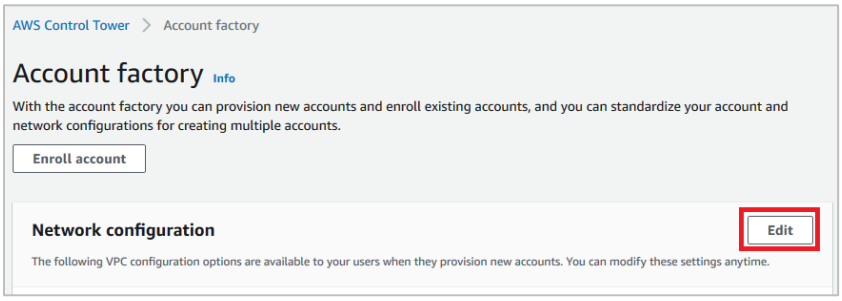
Task 3: Enroll a new AWS account using Account Factory

With Account Factory, you can provision new accounts and enroll existing accounts. You can also standardize your account and network configurations for creating multiple accounts. This gives you control over the virtual private cloud (VPC) configuration that is used for new accounts. This includes the Regions where VPCs are created when you provision an account.

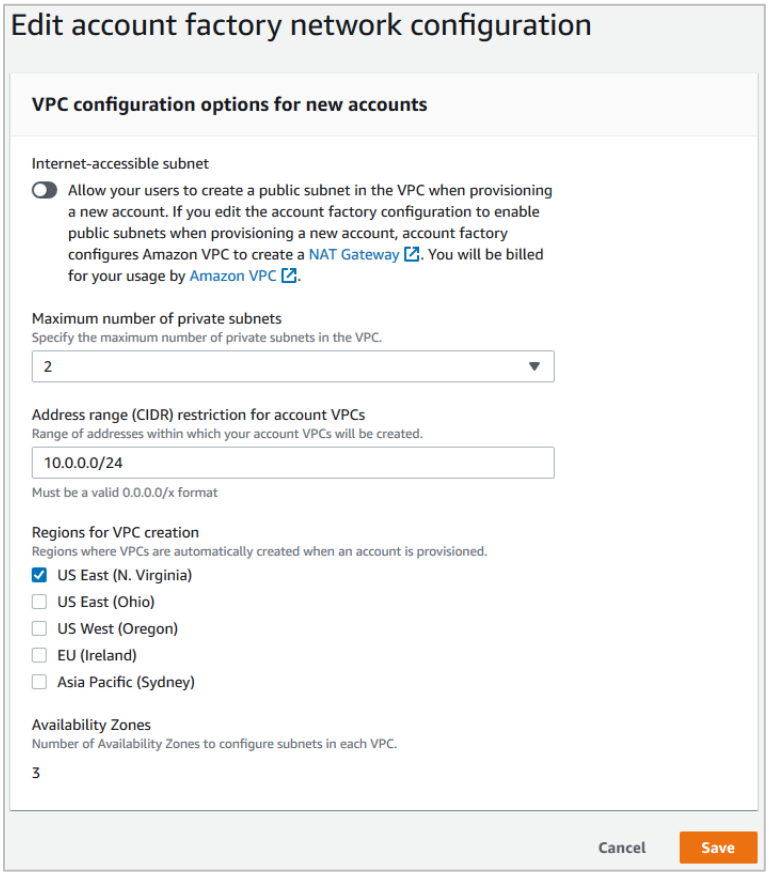
Explore network baseline settings and modifications

In this section, you will explore the available Account Factory options.

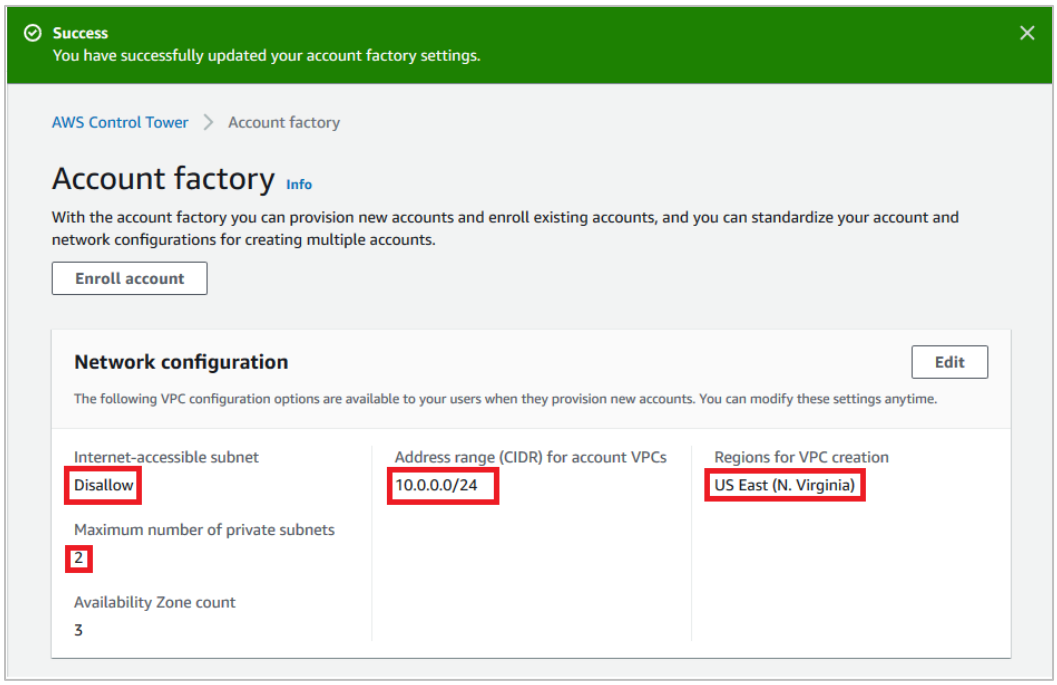
1. In the AWS Control Tower dashboard, in the navigation pane, choose ***Account factory***.
2. In the Network configuration section, choose ***Edit***.



3. In VPC configuration options for new accounts, find the option to enable the internet-accessible subnet. For this lab, keep this option deselected.
4. For Maximum number of private subnets, select ***2***.
5. For the CIDR address range, enter ***10.0.0.0/24***.
6. Select the ***US East (N. Virginia)*** Region.
7. Choose ***Save***.



After the settings update, a success notification displays at the top of the page.



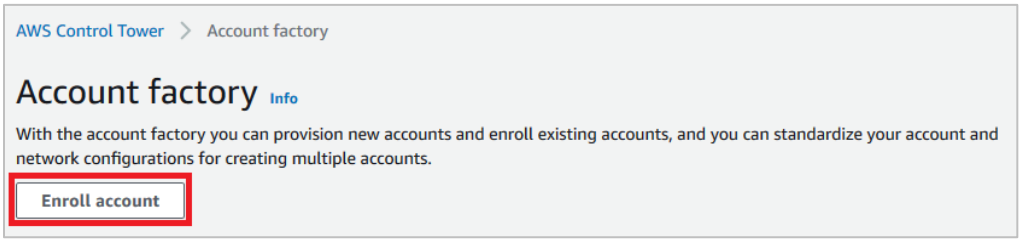
So far, you created an OU, enabled a strongly recommended guardrail, and modified the network baseline settings. Next, you will provision a new account in the OU with Account Factory.

Create an account using Account Factory

Now that your baseline environment is set up, you can use Account Factory to provision new accounts in your landing zone and enroll existing AWS accounts so that they are governed by AWS Control Tower.

An Account Factory account is an AWS account that is provisioned using Account Factory in AWS Control Tower. By default, an AWS Control Tower administrative user has permission to launch Account Factory. In this section, you will provision a new AWS account as an AWS Control Tower administrative user.

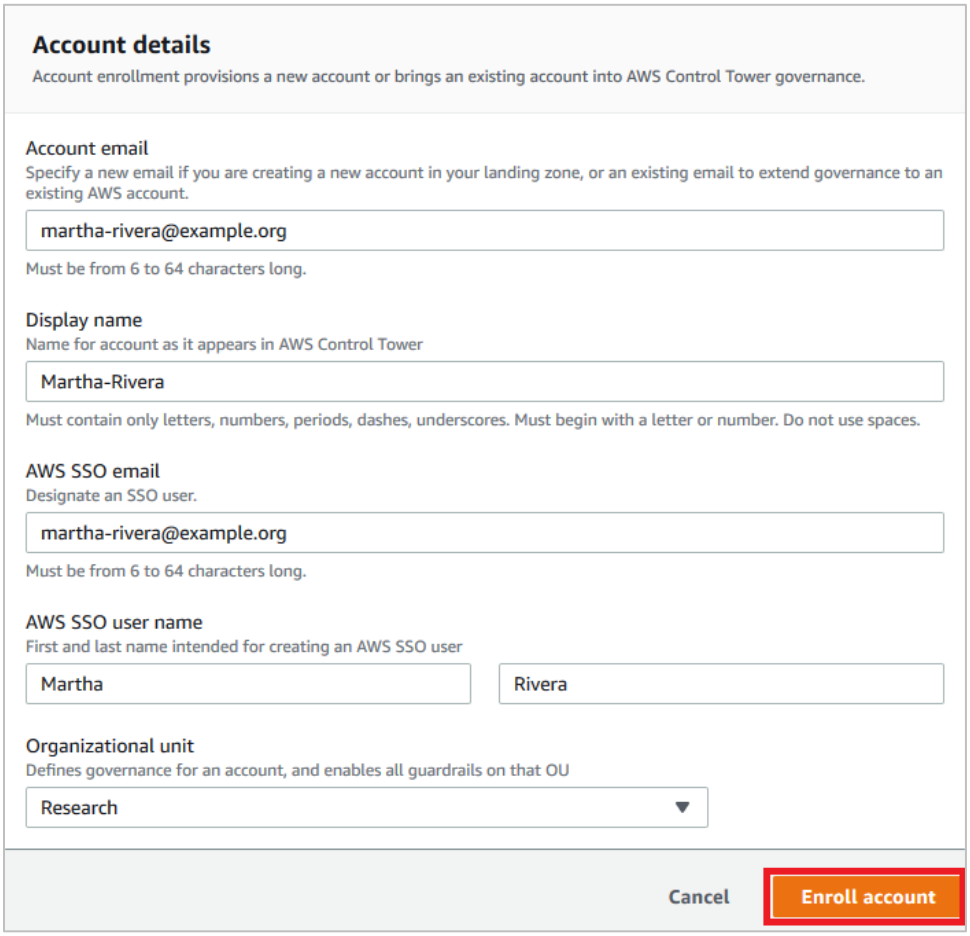
1. In the AWS Control Tower console, go to the navigation pane, and choose ***Account factory***.
2. Choose ***Enroll account***.



3. Enter the account email, display name, and the AWS Single Sign-On user name and email. Use a valid email address. You will receive an invitation for single sign-on. Use *Martha Rivera* as the user name.

Note
This lab uses a fictitious name and email address for illustration purposes. Use your own email account or alias. After the account enrollment process is successfully submitted, you will see a success notification at the top of the page.

4. For the organizational unit, choose ***Research***.
5. Choose ***Enroll account***.



6. To complete the process, look for an email, accept the invitation, and verify the account.

Important

The enrollment process can take up to 20 minutes to complete.

Lab 2: AWS Service Catalog Portfolio Management

As part of the daily workloads for your company, you need a set of services to perform daily tasks, and meet security and compliance requirements. AWS Service Catalog enables organizations to create and manage catalogs of IT services that are approved for use on AWS. These IT services can include everything from virtual machine (VM) images, servers, software, and databases to complete multi-tier application architectures.

You can centrally manage commonly deployed IT services to achieve consistent governance and meet your compliance requirements, while enabling users to deploy only the approved IT services they need. The IT services are grouped as products, and the products are organized in portfolios.

In this lab, you will learn how to:

- Share AWS Service Catalog portfolios with organizational units
- Enable self-service in a child account

Important
Ensure that you use the same Region when performing the labs.

Task 1: Share a portfolio with selective organizational units

You created an organizational unit and provisioned an AWS account in the first lab. Now, you will use those same resources in this lab to share the portfolio from the management account.

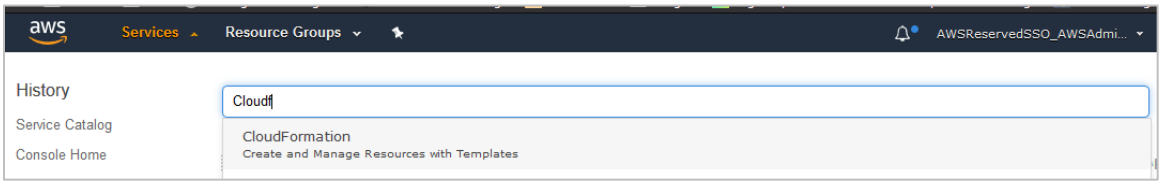
Share a set of products across child accounts

In this section, you will use the AWS CloudFormation service-managed permissions to create AWS Identity and Access Management (IAM) roles across multiple accounts in the organization. The AWS CloudFormation template creates the roles, users, and groups to define launch constraints in a portfolio. Only users belonging to the group can launch the products.

Create launch constraint roles on all accounts in the organization

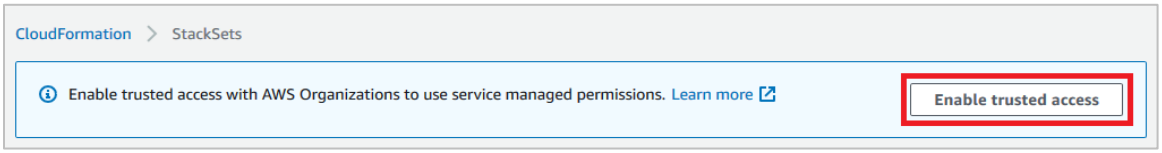
To create constraint roles:

1. In your management account, navigate to the AWS CloudFormation service.

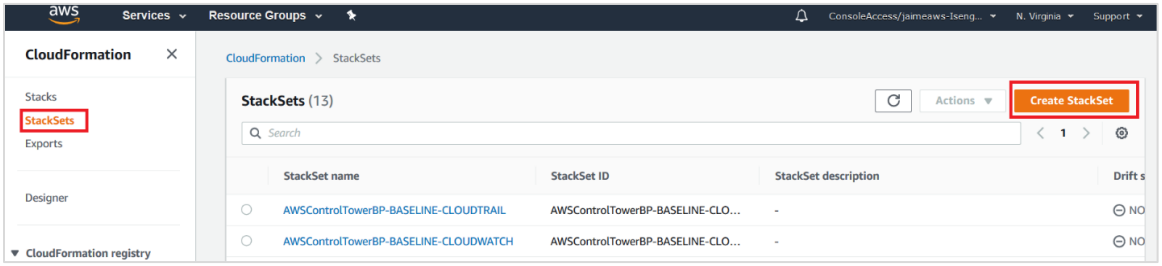


2. In the navigation pane, choose ***StackSets***.

Note
If you see a blue banner to enable trusted access, choose ***Enable trusted access***. This is shown if the trusted access is already enabled for your account.

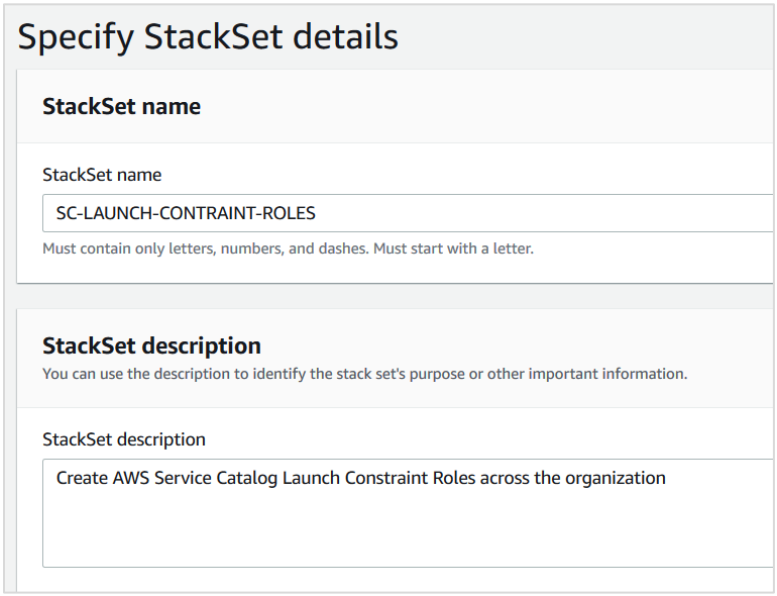


3. Choose ***Create StackSet***.



4. On the Choose a template page, in the Specify template area, choose the ***Amazon S3 URL*** box, and enter https://marketplace-sa-resources.s3.amazonaws.com/Create_scenduser_launchconstraint.yaml
5. Choose ***Next***.
6. On the Specify StackSet details page, enter the following values:

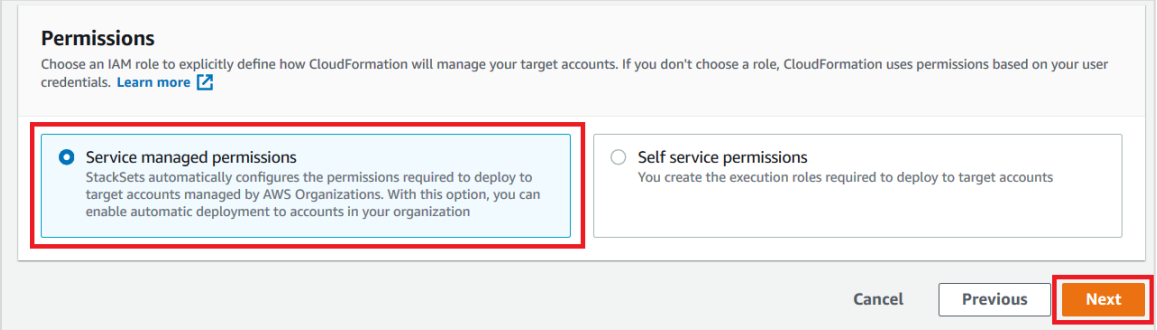
- For StackSet name, enter ***SC-LAUNCH-CONTRAINT-ROLES***.
- For StackSet description, enter ***Create AWS Service Catalog Launch Constraint Roles across the organization***.



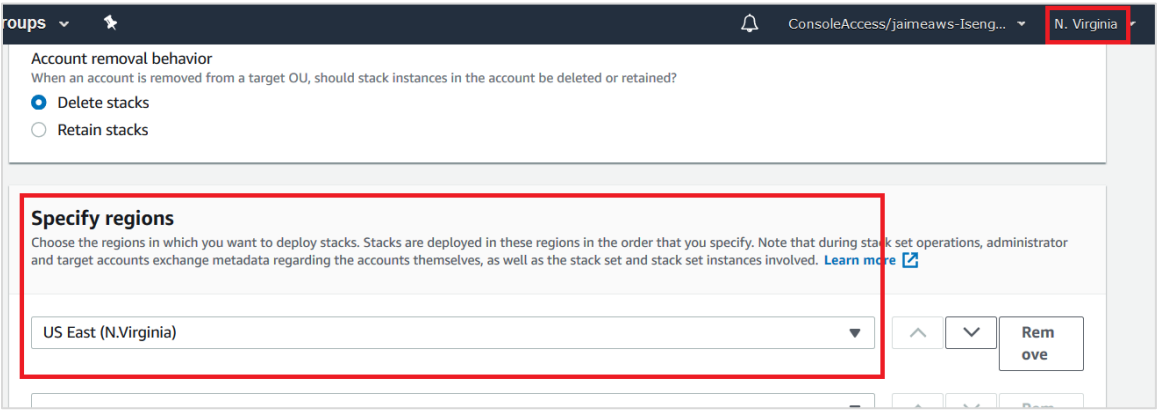
- Under Parameters, leave the default values. (Parameters are defined in your template and allow you to input custom values when you create or update a stack.)

- Note that the default password for SCEndUser is *Change@me*. You will need this later to log in and confirm that the constraints are in effect.

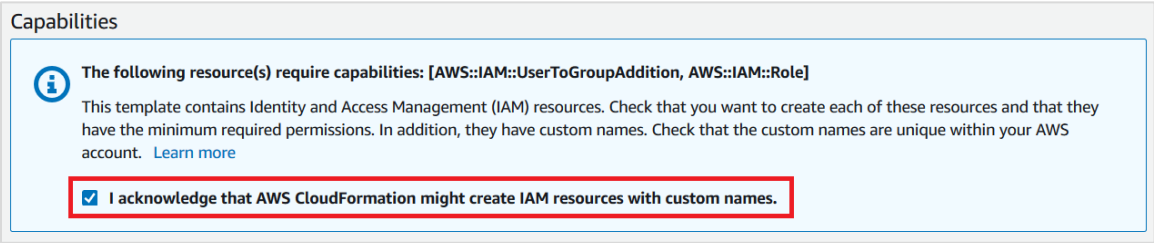
7. Choose ***Next***.
8. Select ***Service managed permissions***, and choose ***Next***.



9. On the Set deployment options page, in the Specify regions area, select the Region you are currently using, and choose ***Next***.

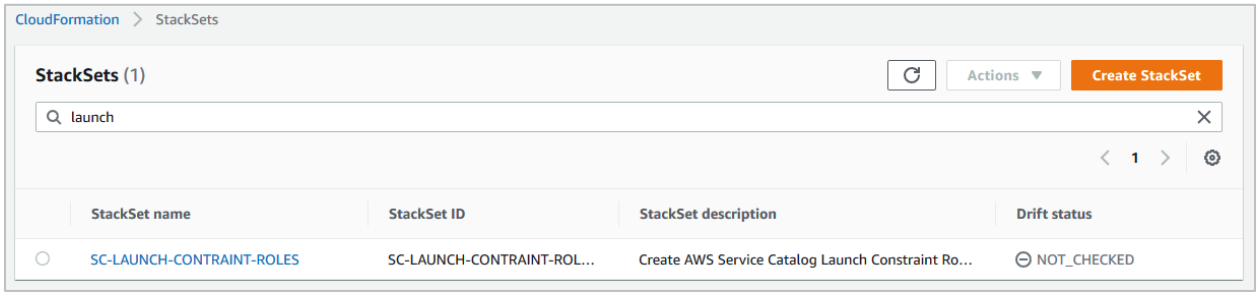


10. On the Review page, select ***I acknowledge that AWS CloudFormation might create IAM resources with custom names***.



11. Review the rest of options, and choose ***Submit***.

You successfully created the StackSet.



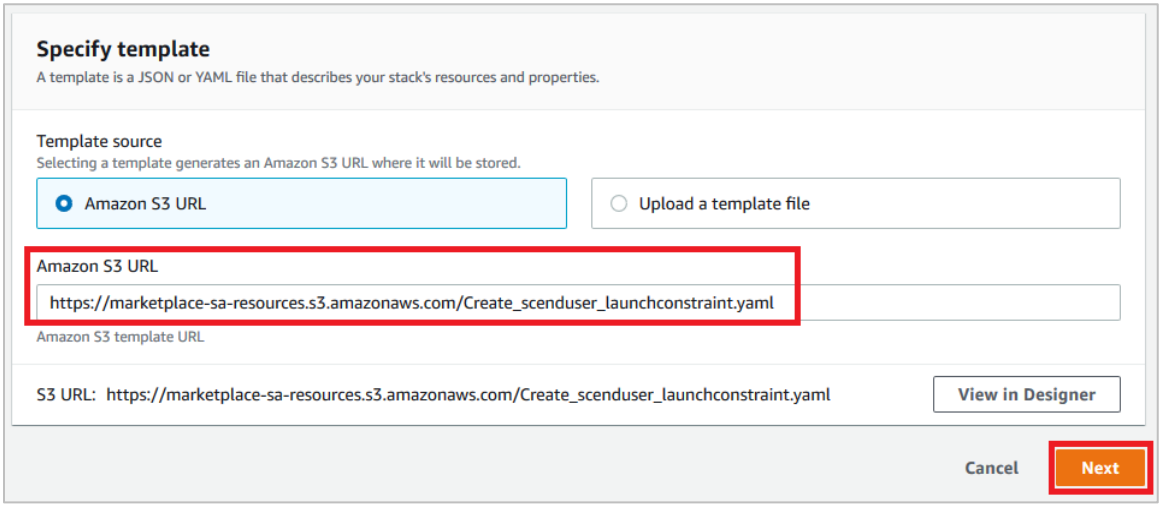
With service-managed permissions, you can deploy stack instances to accounts managed by AWS Organizations in specific Regions. With this model, you don't need to create the necessary IAM roles; StackSets creates the IAM roles for you.

The stack instances are not deployed on the management account when the stack is deployed using the service-managed permissions option, as chosen in this example. You must create the required roles on the management account.

Deploy the stacks on the management account separately

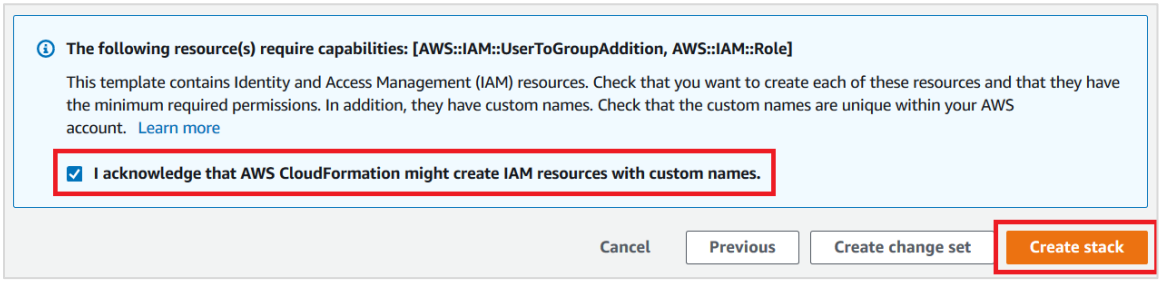
To deploy the stacks:

1. In the AWS CloudFormation console navigation pane, choose ***Stacks***, and then chose ***Create Stack***.
2. Choose ***With new resources (standard)***.
3. On the Create stack page, in the Specify template area, choose the ***Amazon S3 URL*** box, and enter https://marketplace-sa-resources.s3.amazonaws.com/Create_scenduser_launchconstraint.yaml.
4. Choose ***Next***.



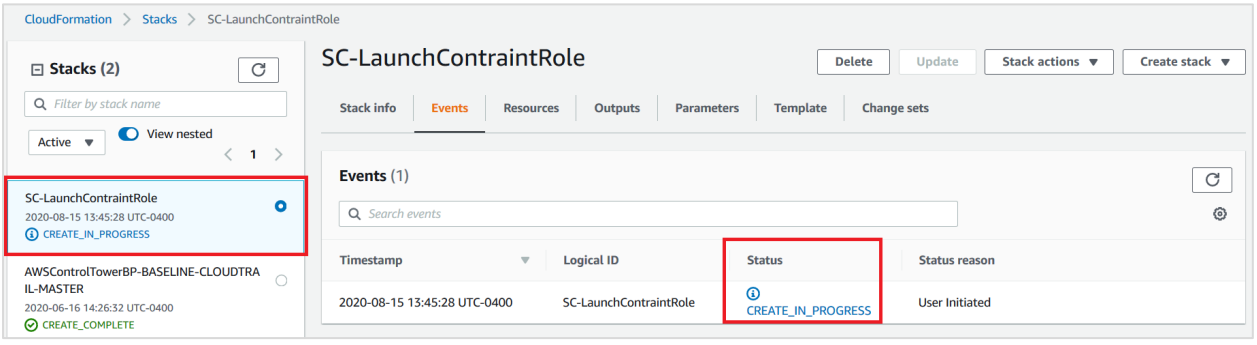
5. On the Specify stack details page, choose the ***Stack name*** box, enter ***SC-LaunchConstraintRole***, accept the defaults, and choose ***Next***.

6. Accept the defaults on the Configure stack options page, and choose ***Next***.
7. On the Review page, select ***I acknowledge that AWS CloudFormation might create IAM resources with custom names***.
8. Choose ***Create stack***.



You successfully created a launch constraint for the management account and for all remaining accounts in the organization. You can now use the AWS Service Catalog Local launch constraints feature.

You should see a window similar to the following image.

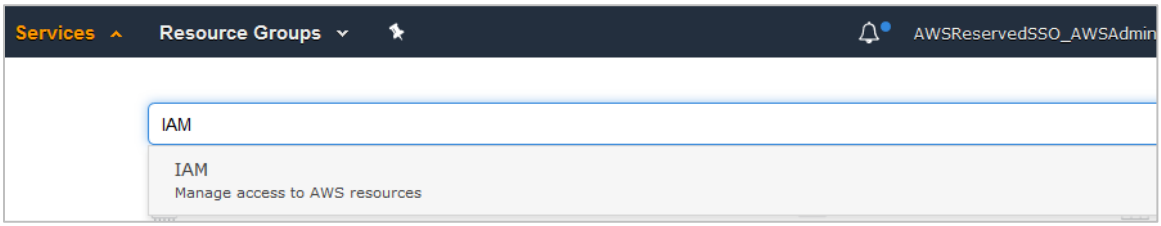


After the status changes to CREATE_COMPLETE, you can share the portfolio with a specific OU.

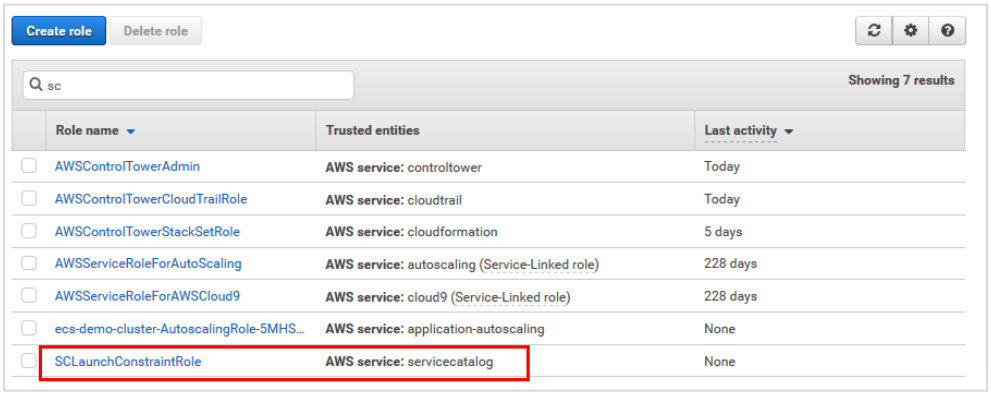
Explore the new assets

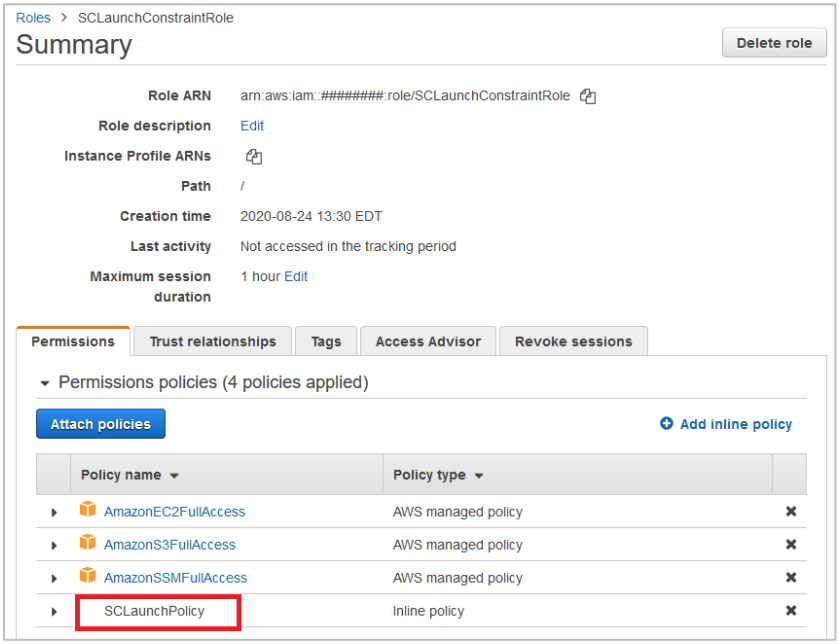
In this section, you will review some of the assets that you created in the previous task.

1. Navigate to the AWS Identity and Access Management (IAM) service.

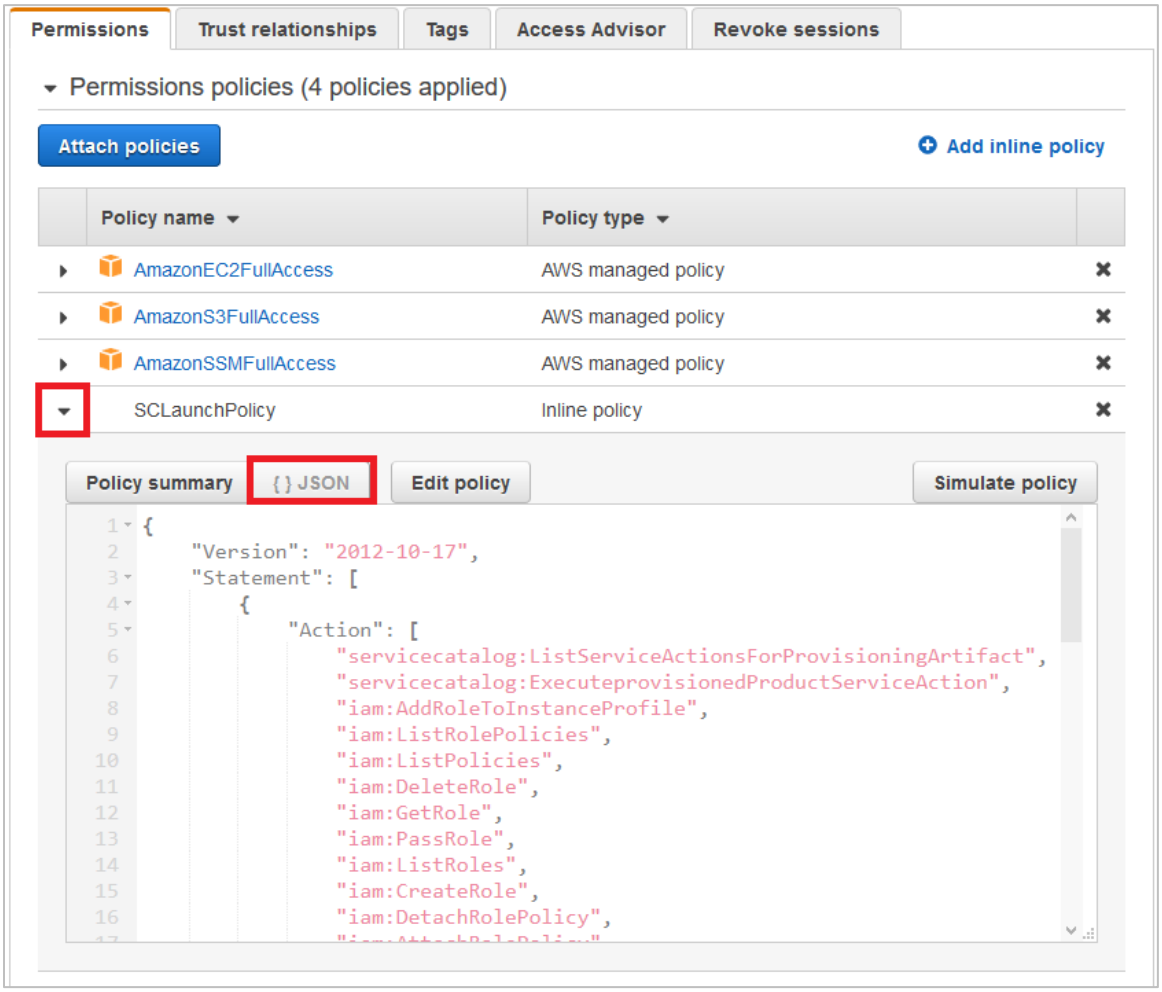


2. In the navigation pane, locate ***Access management***, and choose ***Roles***.
3. Choose ***SCLaunchConstraintRole***. The ***SCLaunchPolicy*** is attached to the new role.

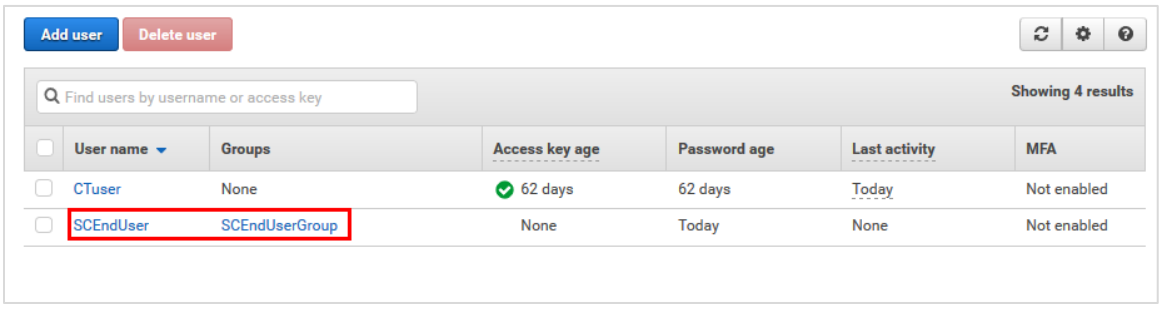




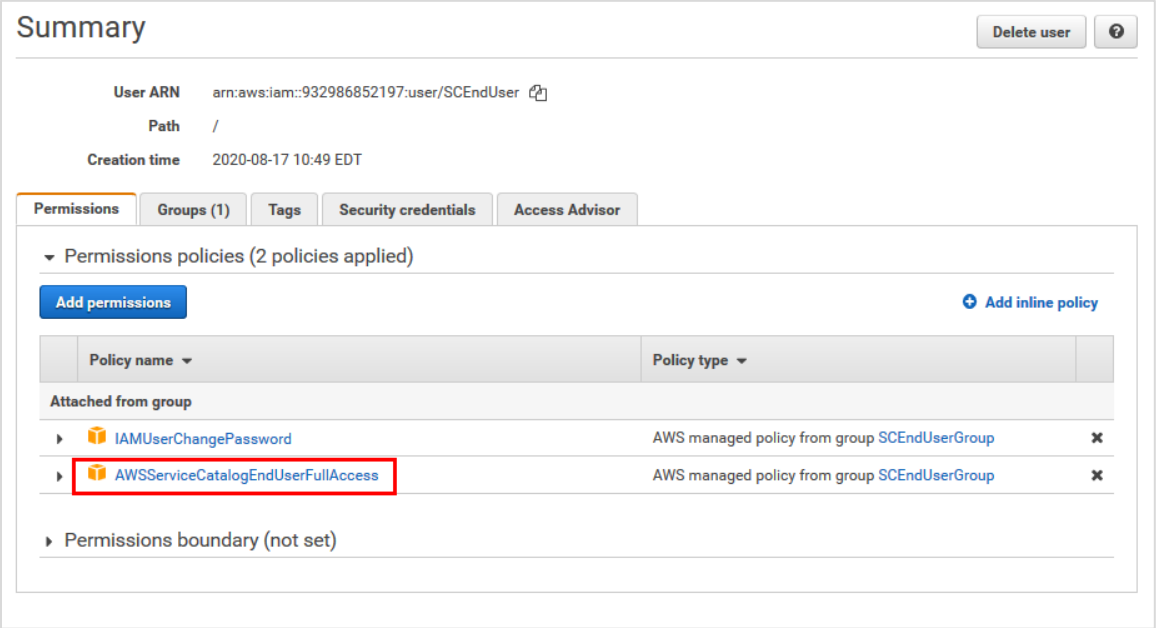
4. Expand the policy and choose ***{} JSON***. You can see that the Service Catalog is part of the allowed services.



5. In the navigation pane, choose ***Users***. Notice that the new ***SCEndUser*** and ***SCEndUserGroup*** are added.



6. Choose ***SCEndUser***. Notice that this user has permission to access the Service Catalog.



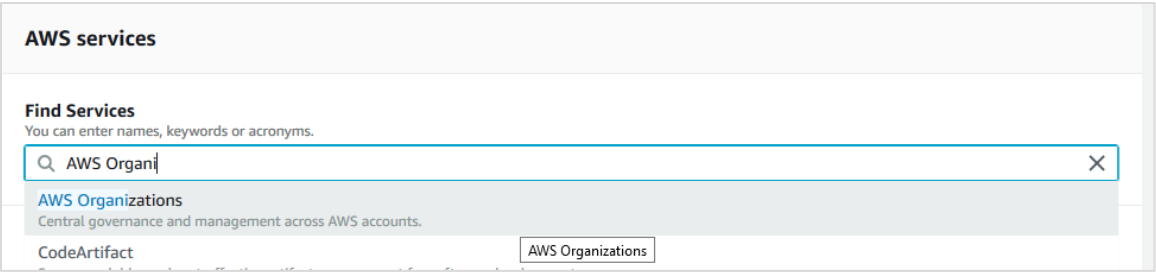
Share a portfolio from management account with a specific OU

In Lab 1, you enrolled an account and added it to an organizational unit called *Research*. You will use that account, Martha Rivera's account, and that OU in this section.

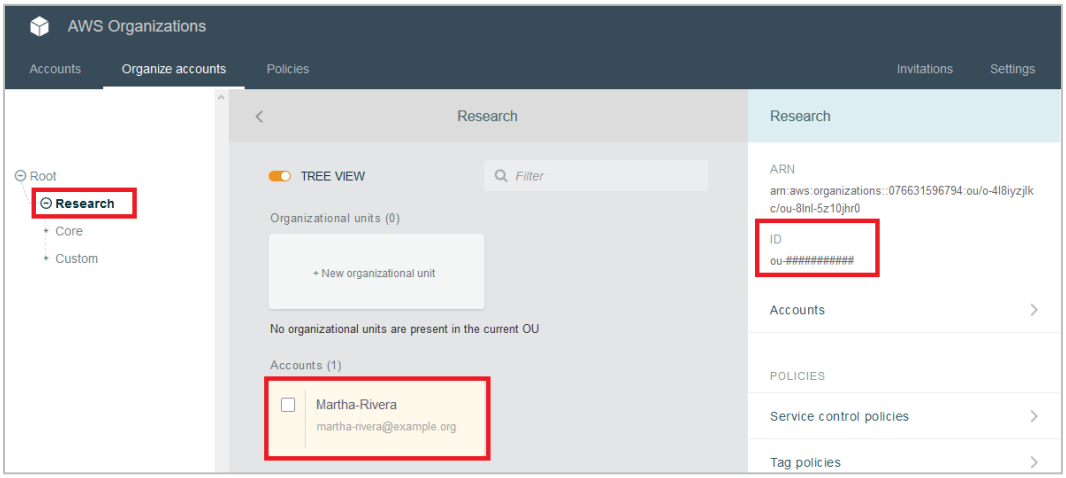
In this section, you will share a portfolio with that OU, so users in the OU can launch the portfolio products.

Collect the organizational unit ID

1. Using your management account, log in to AWS Management Console, and open the AWS Organizations console.



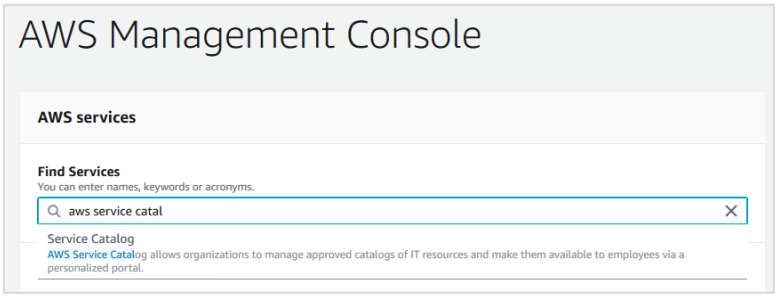
2. Choose the ***Organize accounts*** tab.
3. Select ***Research***.
4. In the ***details*** section, find the ***Organization Unit ID***, and record it for later use.



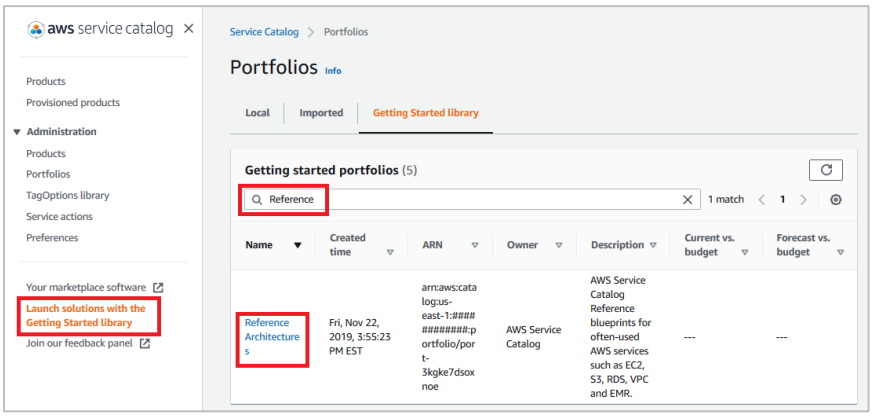
Create a portfolio on the management account (from the Getting Started Library)

In this section, you will use existing products in the Getting Started Library to create the portfolio.

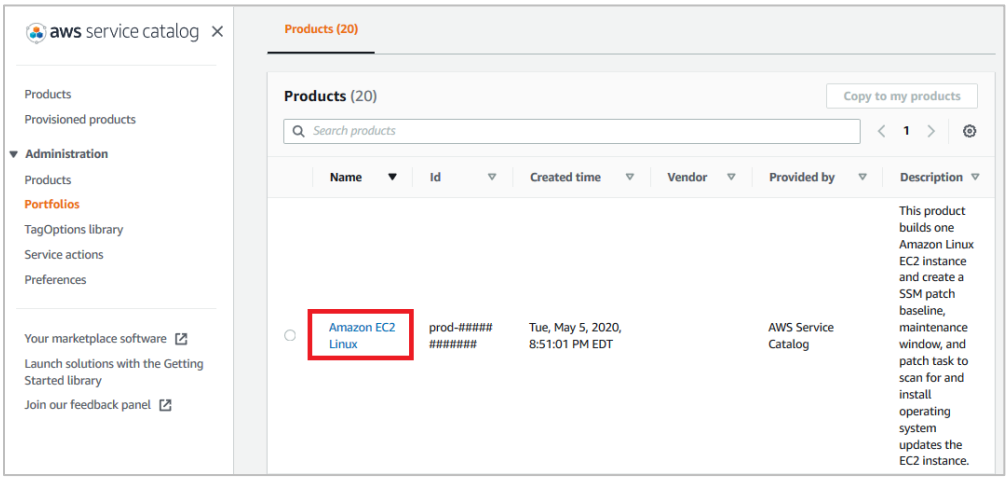
1. Use your management account to open the AWS Service Catalog.



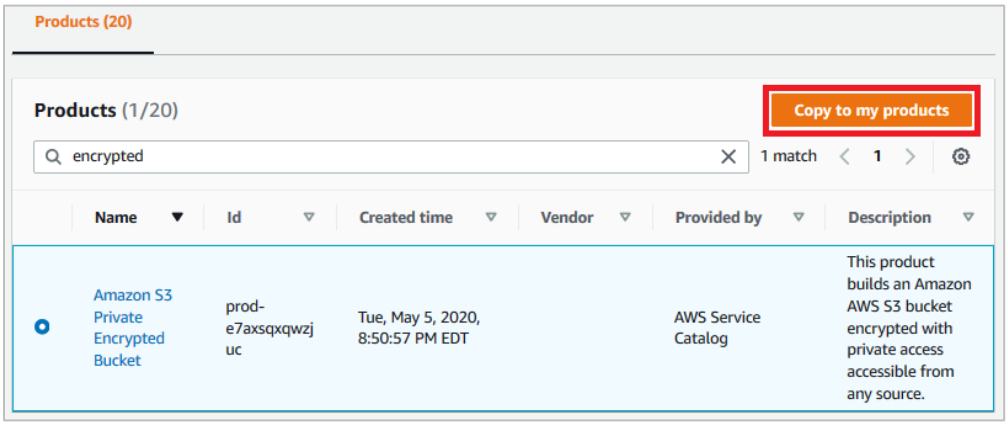
2. In the navigation pane, locate ***Administration***, and choose ***Products***.
3. In the navigation pane, choose ***Launch solutions with the Getting Started Library***.
4. In the search box, enter ***Reference Architectures***, and then select it.



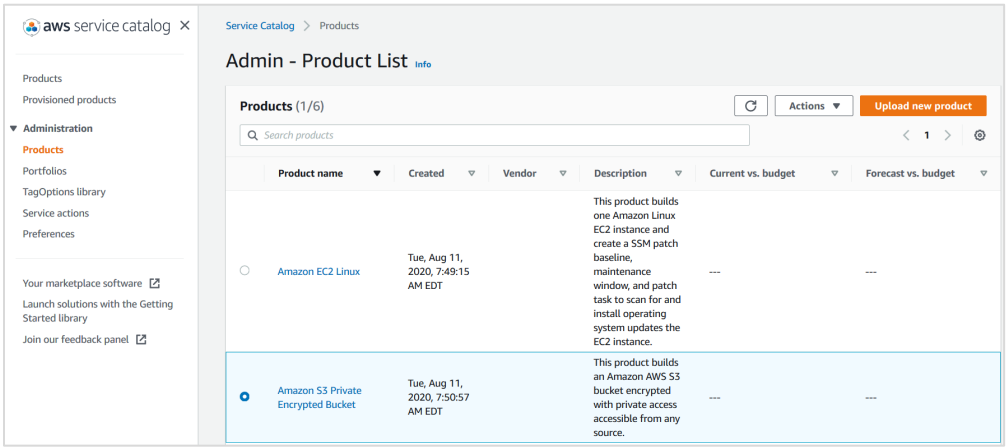
5. Select ***Amazon EC2 Linux***, and choose ***Copy to my products***.



6. Select ***Amazon S3 Private Encrypted Bucket***, and choose ***Copy to my products***.



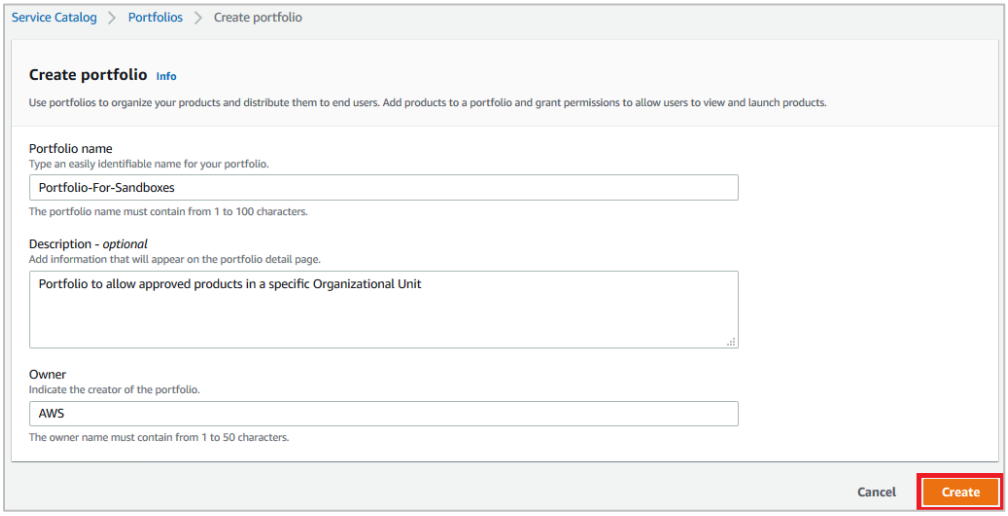
7. In the navigation pane, locate ***Administration***, and choose ***Products***.



8. In the navigation pane, locate ***Administration***, and choose ***Portfolios***.

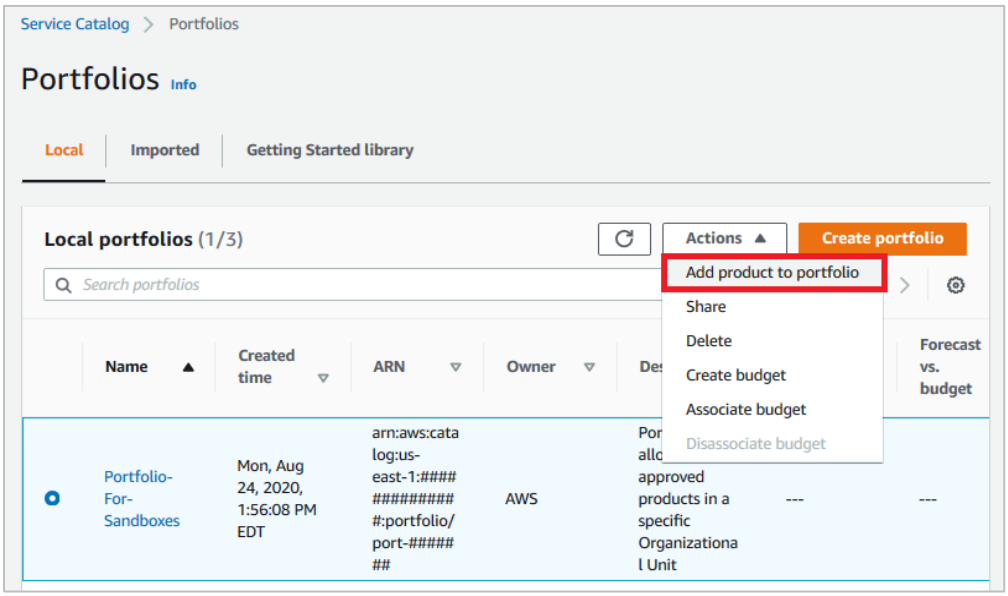
9. In the ***Local portfolios*** table, chose ***Create portfolio***.
10. Fill in the fields as follows, and then choose ***Create***:

- For Name, enter ***Portfolio-For-Sandboxes***.
- For Description, enter ***Portfolio to allow approved products in a specific organizational unit***.
- For Owner, enter ***AWS***.

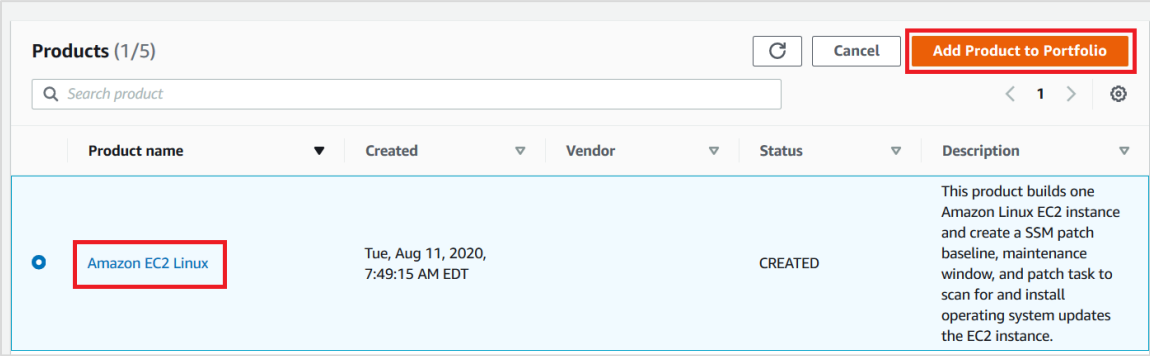


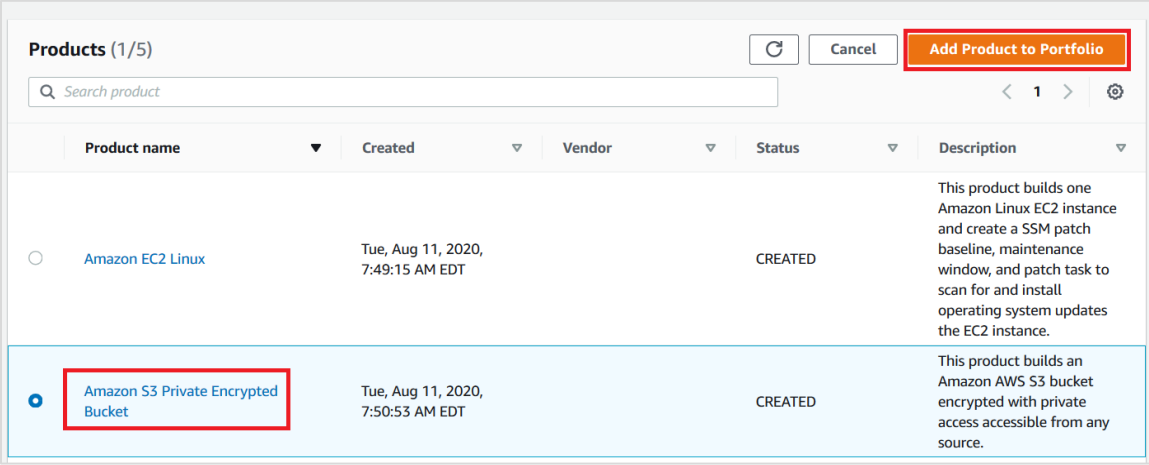
You are directed back to ***Portfolios*** page.

11. On the Portfolios page, choose ***Portfolio-For-Sandboxes***.
12. For Actions, select ***Add product to portfolio***.



13. One at a time, select the products you copied earlier, ***Amazon EC2 Linux*** and ***Amazon S3 Private Encrypted Bucket***, and choose ***Add Product to Portfolio***.

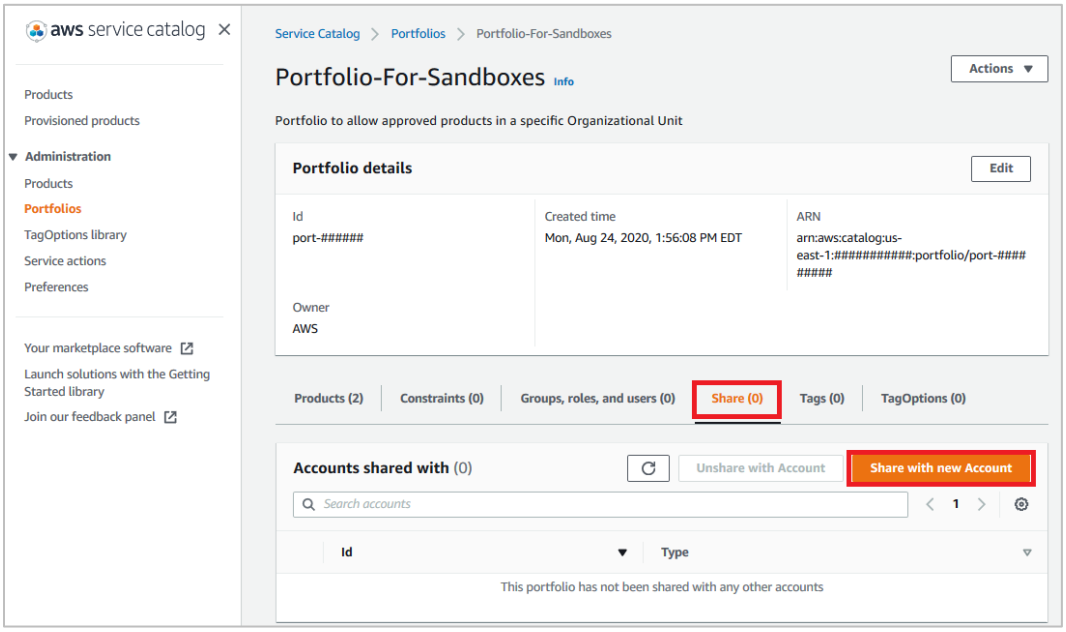




Share the product portfolio

At this stage, your product portfolio is ready for sharing. To share the portfolio:

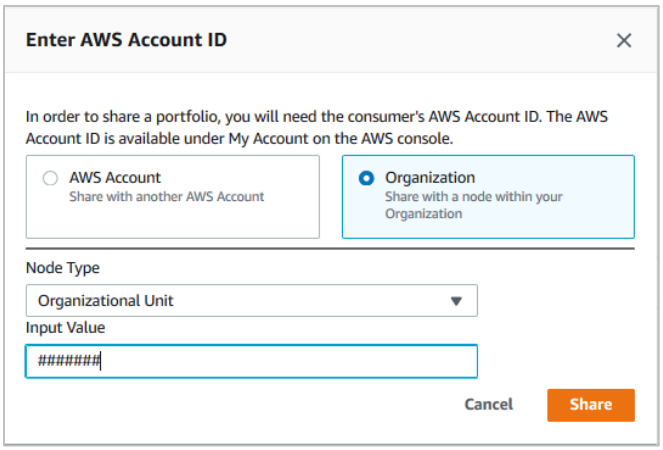
1. On the ***Portfolios*** page, choose ***Portfolio-For-Sandboxes***.
2. Select ***Share***, and then choose ***Share with new Account***.



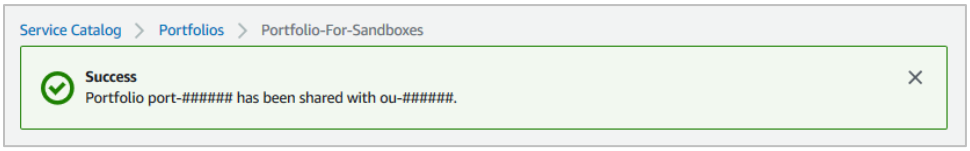
Now, you can share the portfolio with the Research OU.

3. In the Enter AWS Account ID dialogue box, do the following:

- Select ***Organization***.
- For Node Type, select ***Organizational Unit***.
- For Input Value, enter the organizational unit ID that you recorded at the beginning of this task.



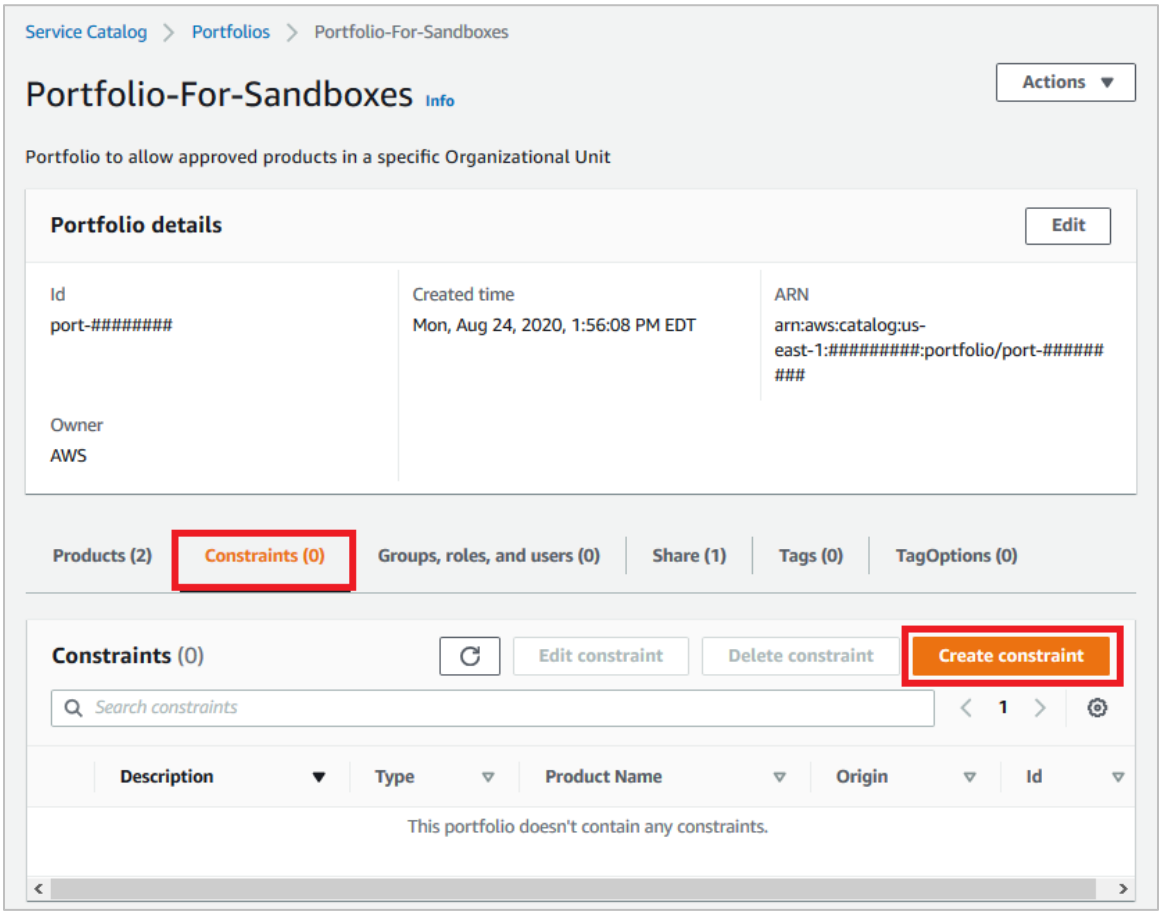
4. Choose ***Share***. You should see a success message.



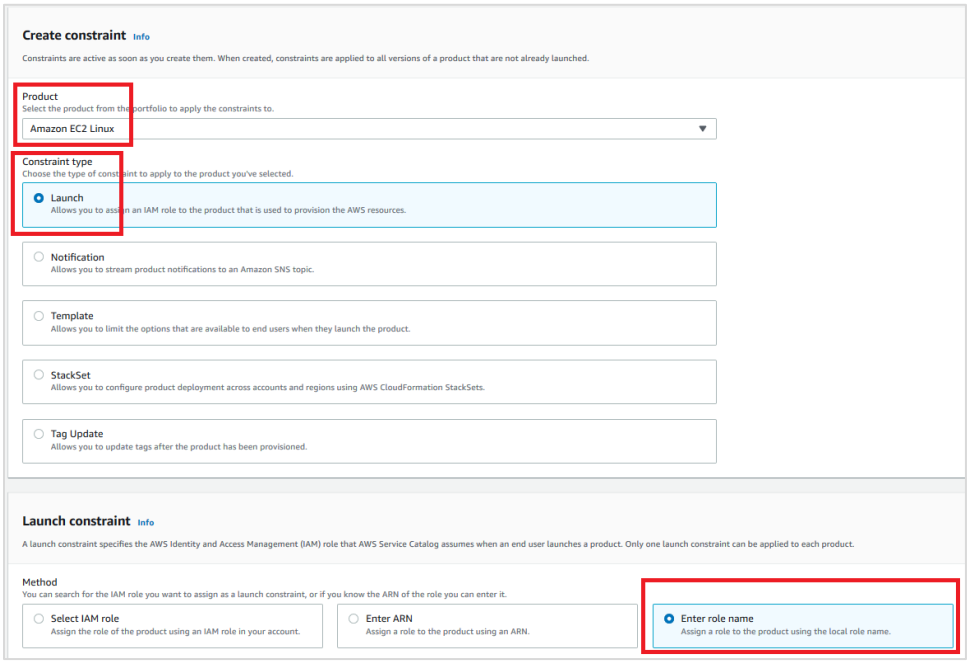
Add launch constraints to the portfolio

So far, you created a portfolio in the management account. You added products from the Service Catalog Getting Started library to the portfolio and shared it with an OU in your environment. Next, you will add launch constraints to the portfolio.

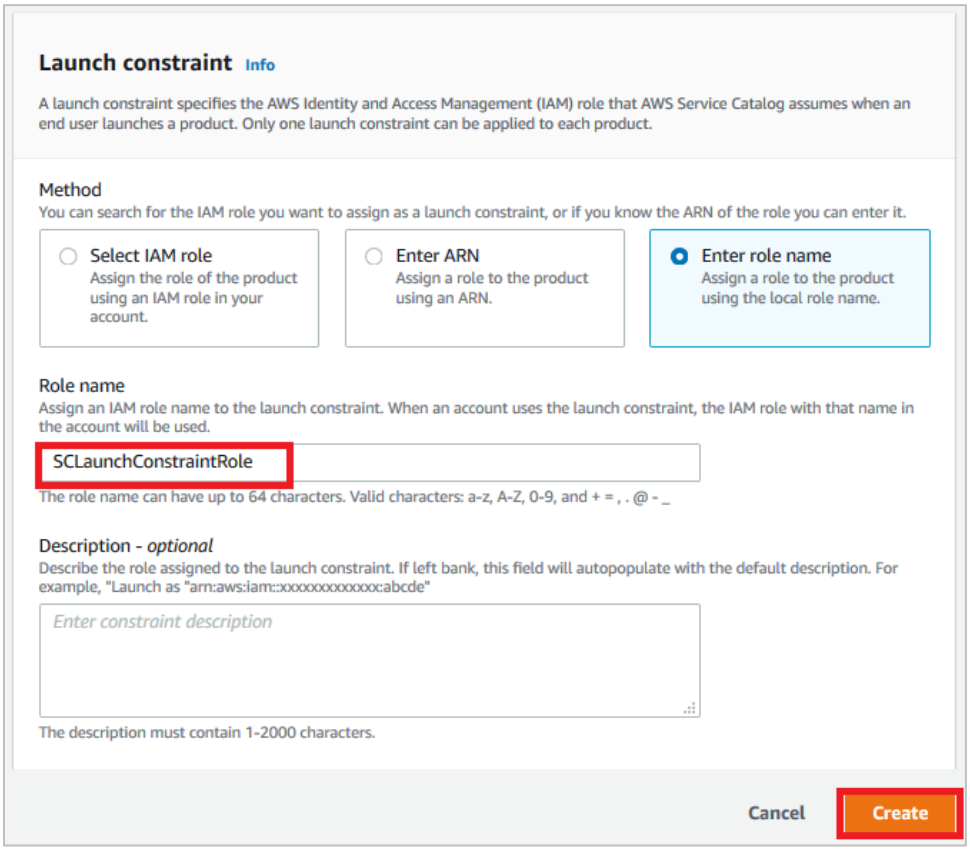
1. On the ***Portfolios*** page, choose ***Portfolio-For-Sandboxes***.
2. Select ***Constraints***, and choose ***Create constraint***.



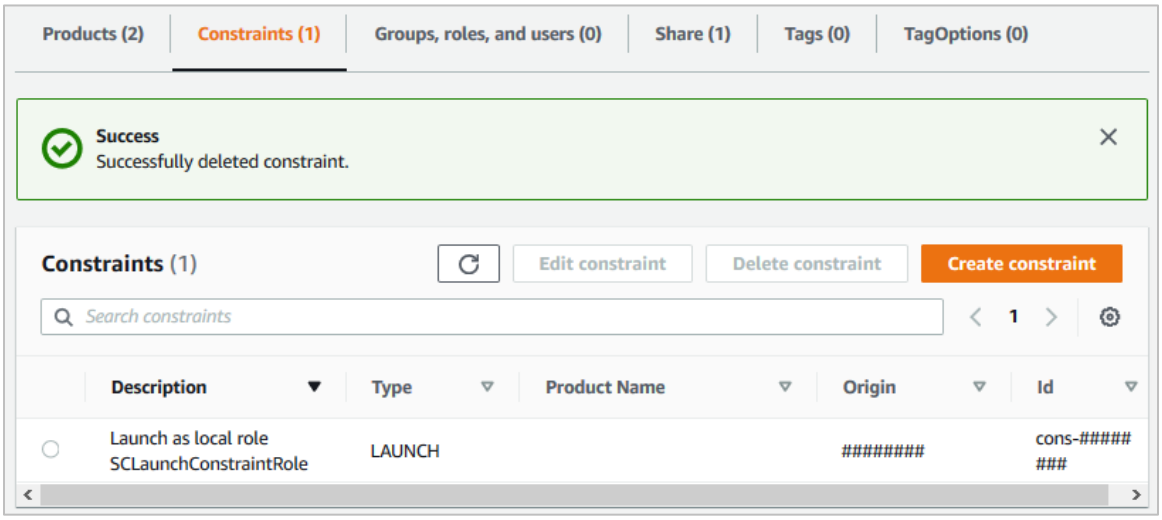
3. On the Create constraint page, locate ***Product***, and select ***Amazon EC2 Linux***.
4. For the constraint type, select ***Launch***.
5. Locate ***Launch constraint***, and select ***Enter role name for the method***.



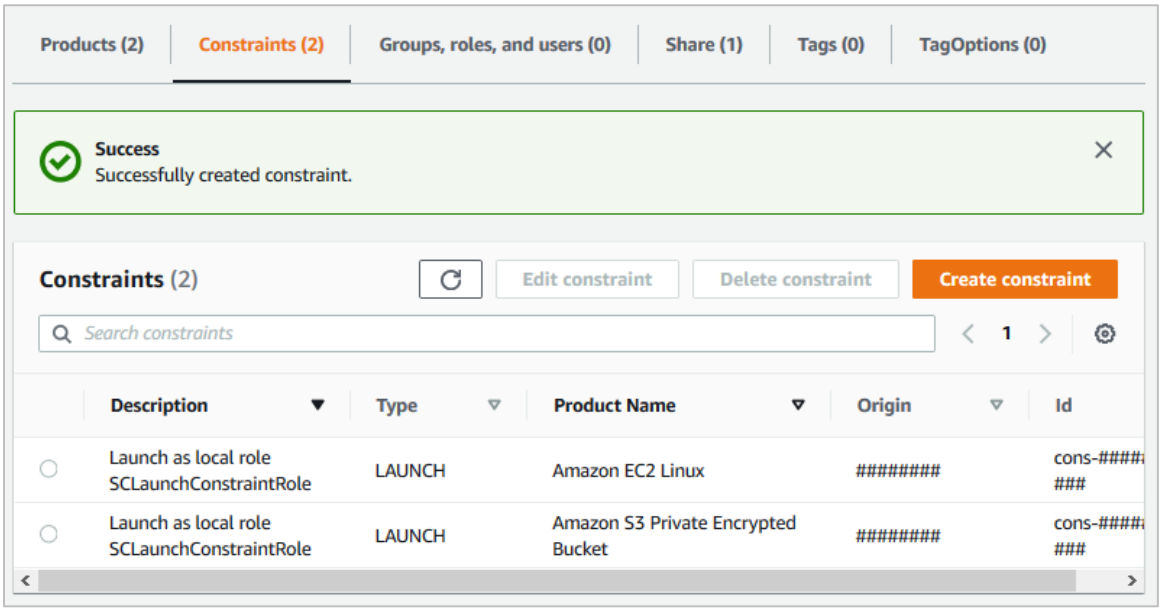
6. For Role name, enter ***SCLaunchConstraintRole*** (you created this role with AWS CloudFormation in the previous task), and choose ***Create***.



You should see a success message after the launch constraint is successfully created.



7. Repeat this process for the Amazon S3 Private Encrypted Bucket product. After the constraints are successfully completed, they display in the portfolio, as shown here.



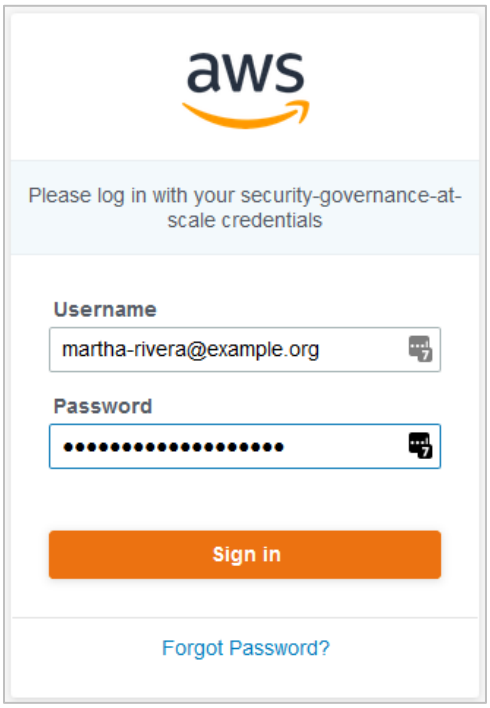
Task 2: Enable self-service in a child account

So far, you created a portfolio in the management account, added products from the Getting Started Library to the portfolio, shared it with an OU in your environment, and enabled local launch constraints. Now, you will enable the portfolio to local users in the AWS account.

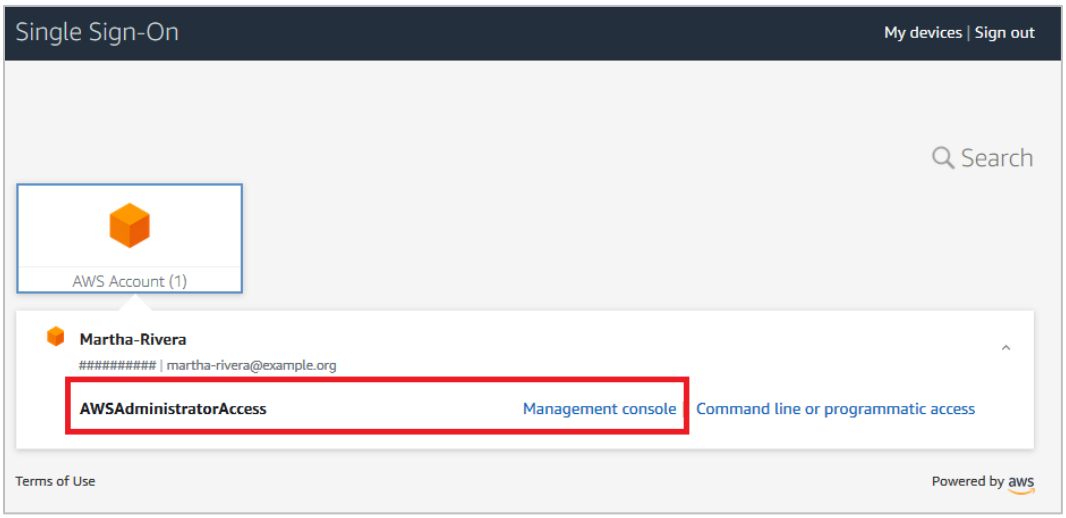
By using the AWS Service Catalog local launch constraint feature, you can add users directly to the local portfolios so they can launch AWS Service Catalog products. By using local launch constraints, there is no need to create local portfolios from imported portfolios.

Enable the portfolio to local users

1. Log in to the child account you created using the Account Factory enrollment account process. Proceed to the user portal screen, and enter Martha's credentials



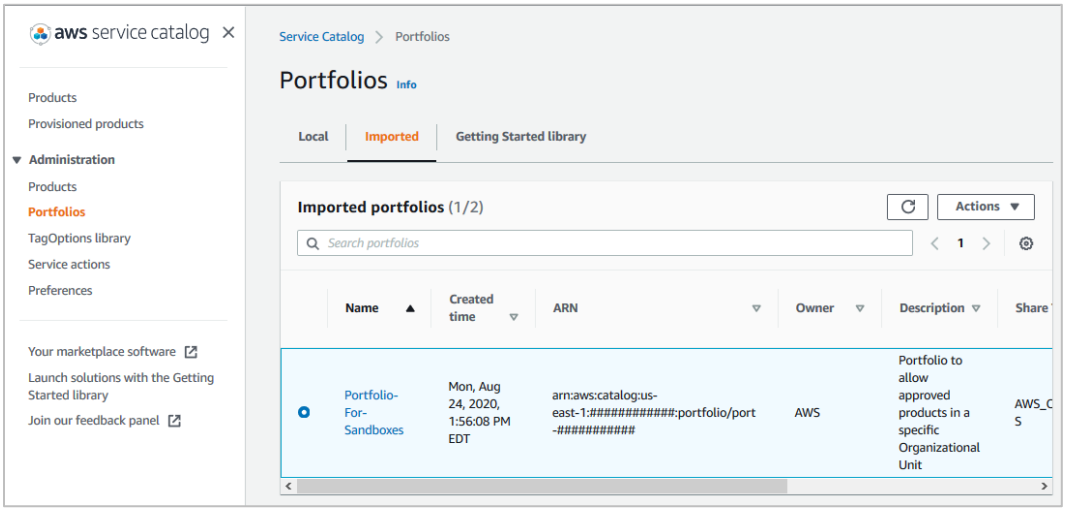
2. Choose the ***AWSAdministratorAccess*** role.



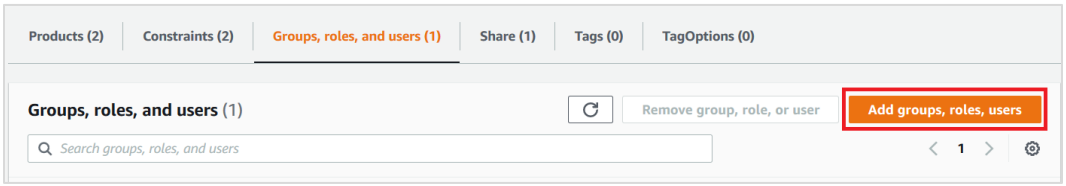
Note

This is not your management account. The example secondary account in this lab is called *Martha Rivera*. This account must be part of the *Research* organizational unit, because it is the one shared with the portfolio.

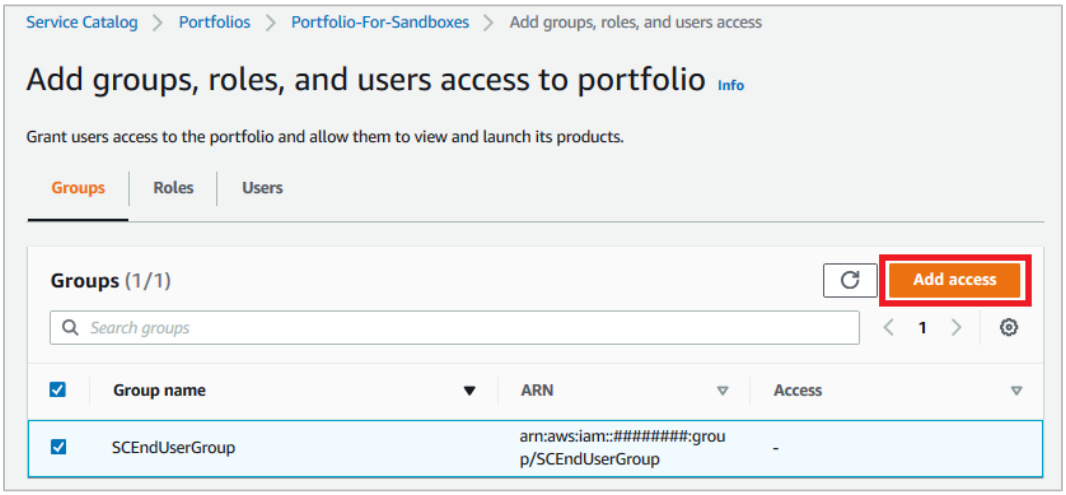
3. Open to the AWS Service Catalog console.
4. In the navigation pane, locate ***Administration***, and choose ***Portfolios***.
5. Select ***Imported***, and then choose ***Portfolio-For-Sandboxes*** from the list.



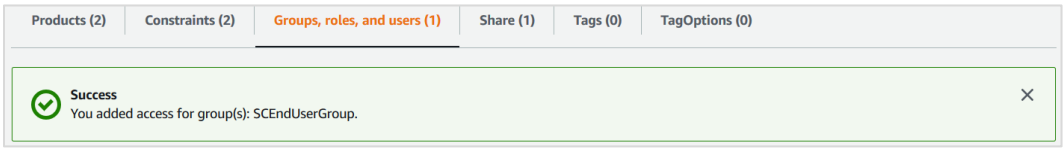
6. Select ***Groups, role, and users***, and choose ***Add groups, roles, users***.



7. For this part of the lab, select ***Groups***, enter ***SCEndUserGroup***, and select the group.
8. Choose ***Add access*** to grant permission to the ***SCEndUserGroup***, which was created as part of the AWS CloudFormation stack creation step.



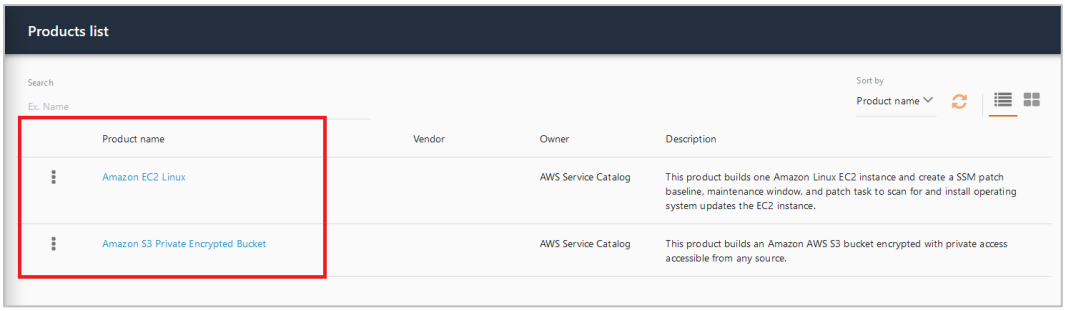
You should get a success message after the permissions are granted.



9. Now, allowed users in the secondary account will have a catalog of services ready to consume. You can verify this by logging in to the secondary account as a SCEndUser and try accessing the service catalog products. The default password for SCEndUser is Change@me. Make sure that the secondary account you choose is a member of the *Research* OU – the organizational unit with which you shared the portfolio.

Note
Confirm that the value for ***Account ID*** matches the account you enrolled using Account Factory on Lab 1 (Martha Rivera's account ID). You might be asked to change your password for your SCEndUser the first time you log in.

10. Navigate to Service Catalog, and explore your product list.



Lab 3: AWS Control Tower Customizations

As your organization works on new projects, chances are that new compliance regulations might arise. You might have new legal requirements, such as a new FDA protocol where you need to meet certain security and compliance criteria. For that reason, you must have a way to customize your guardrails to accommodate new requirements and adopt them quickly. The customization framework for AWS Control Tower lets you extend and customize your existing setup to remain agile and compliant.

The Customizations for AWS Control Tower solution combines AWS Control Tower and other highly available, trusted AWS services. This solution helps customers set up a secure, multi-account AWS environment efficiently, using AWS best practices. This solution enables customers to add customizations to their AWS Control Tower landing zone using AWS Config and service control policies (SCPs). You can deploy the custom template and policies to the different organizational units (OUs) in your organization.

This solution integrates with AWS Control Tower lifecycle events to ensure that resource deployments stay synced with the customer's landing zone. For example, when a new account is created using the AWS Control Tower, guardrails are automatically deployed.

In this lab, you will perform these tasks:

- Set up the Customizations for Control Tower solution
- Deploy customizations
- View deployed customizations

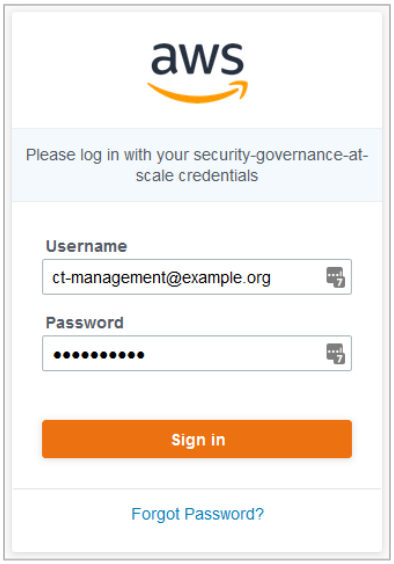
Note
Ensure that you use the same Region for the labs. You must be in the same Region that you selected when you set up your landing zone using AWS Control Tower.

Task 1: Set up the Customizations for Control Tower solution

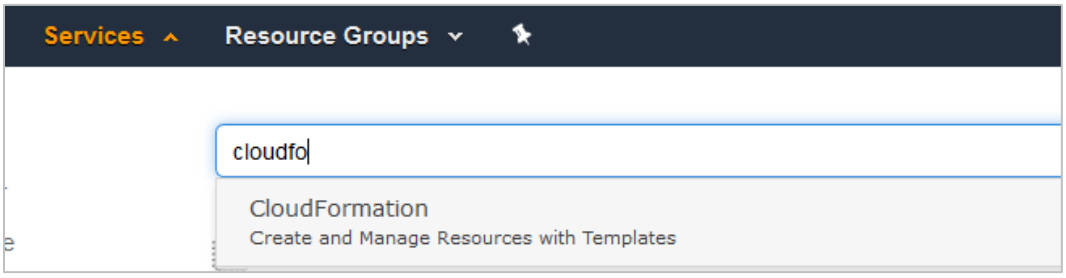
In this section of the lab, you will deploy the Customizations for Control Tower solution on your management account in your AWS Control Tower home Region. This is the AWS Region where you launched AWS Control Tower.

To deploy the Customizations for Control Tower solution on your management account:

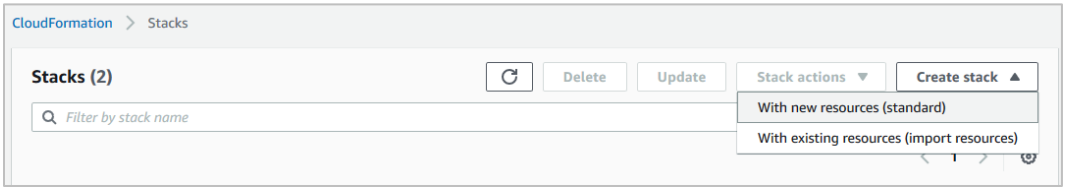
1. Using the AWS SSO user portal, log in to AWS Management Console with your management account.



2. Navigate to the ***AWS CloudFormation*** console.

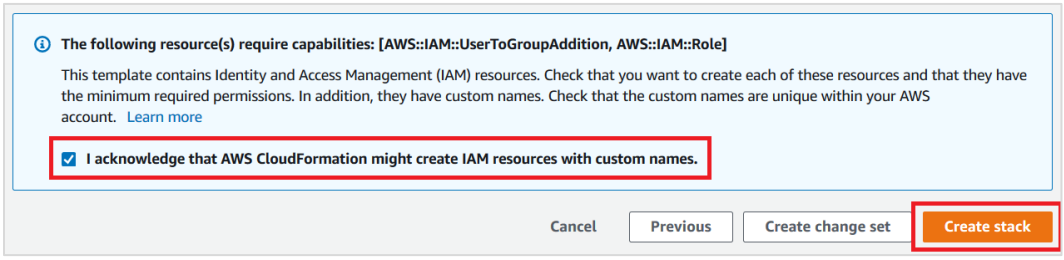


3. In the navigation pane of the AWS CloudFormation console, select ***Stacks***, and then choose ***Create Stack***.
4. Choose ***With new resources (standard)***.



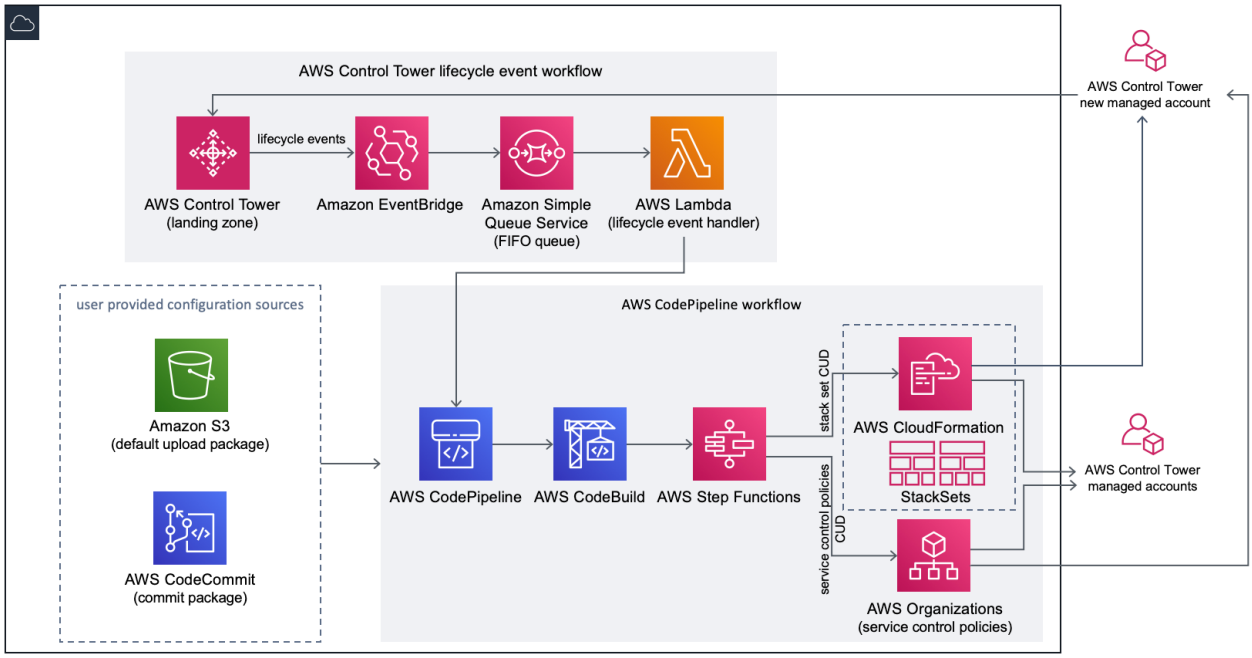
5. On the Create stack page, in the ***Specify template*** area, choose the ***Amazon S3 URL*** box, and enter https://s3.amazonaws.com/solutions-reference/customizations-for-aws-control-tower/latest/custom-control-tower-initiation.template.
6. Choose ***Next***.
7. On the Specify stack details page, for Stack name, enter ***CustomizationsForCTSolution***, and accept the other defaults.
8. On the same page, in the ***AWS CodePipeline Source*** area, select ***AWS Code Commit***.

9. Accept the defaults on the Configure stack options page, and choose ***Next***.
10. On the Review page, select ***I acknowledge that AWS CloudFormation might create IAM resources with custom names***.
11. Choose ***Create stack***.



The stack creation takes approximately 5 minutes.

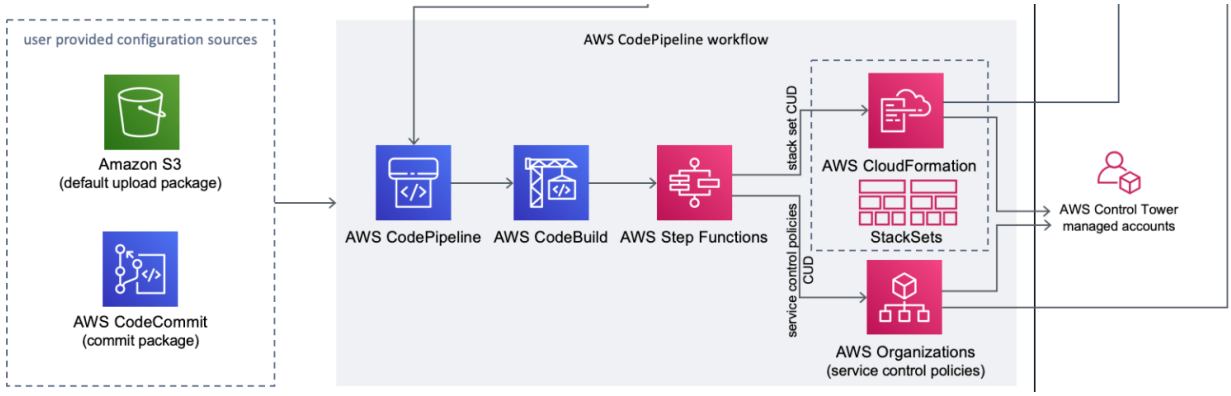
When it completes, you successfully deployed the Customizations for Control Tower solution. This solution follows best practices and is the recommended solution to customize AWS Control Tower. The following architecture diagram shows the resources provisioned by the stacks and how they connect.



The solution uses the AWS Control Tower lifecycle event workflow, such as a set of events that take place when different operations occur in AWS Control Tower. For example, when an account is created, an event triggers the solution workflow. The solution uses Amazon EventBridge to

process the events and then triggers actions. The actions are queued in a first-in, first-out (FIFO) queue in Amazon Simple Queue Service (Amazon SQS), which then is consumed by a lifecycle event handler using AWS Lambda.

In the following diagram example, you are not necessarily using the events lifecycle.



You use AWS CodeCommit to commit the package into the repository. AWS CodePipeline triggers the workflow. This builds the code, runs the necessary step functions, and deploys the customizations using AWS CloudFormation StackSets. The managed accounts are deployed with the configurations in the package (for instance, the manifest.yaml file that you will edit in the next task).

The next task covers the deployment of these customizations:

- Preventive guardrail (service control policy)
- Detective guardrail (AWS Config rule)
- IAM role

The code for the customizations is in a .zip file. You will use Git to commit the package to the AWS CodeCommit repository that the stack was provisioned on in the previous step.

Reference
For more information on lifecycle events in AWS Control Tower, see:
https://docs.aws.amazon.com/controltower/latest/userguide/lifecycle-events.html.

Task 2: Deploy customizations

You have successfully deployed customizations for AWS Control Tower framework. In this lab, you will deploy the customizations in the package described earlier – preventive guardrail, detective guardrail, and IAM role. All three customizations for this lab are part of the same package. You will deploy all three with a single Git commit.

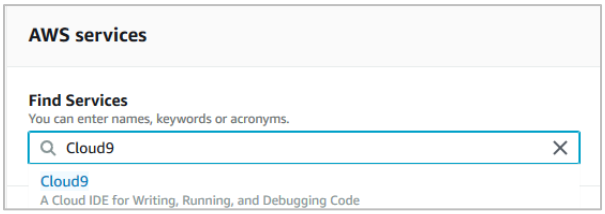
Prepare repository access with AWS Cloud9

To commit your customizations, you must connect to your Git repository in AWS CodeCommit. You can use AWS Cloud9 to make code changes in a CodeCommit repository. AWS Cloud9

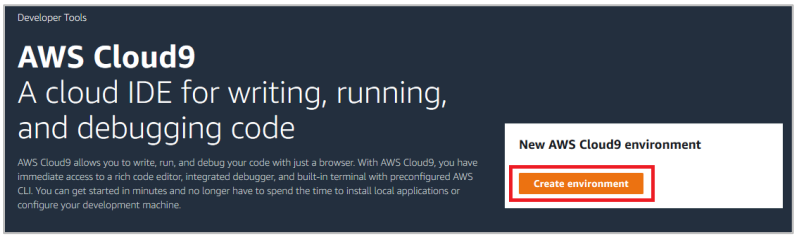
contains a collection of tools that you can use to write code and build, run, test, debug, and release software. From your development environment, you can clone existing repositories, create repositories, and commit and push code changes to a repository.

The AWS Cloud9 Amazon Elastic Compute Cloud (Amazon EC2) development environment is generally preconfigured with the AWS Command Line Interface (AWS CLI), an Amazon EC2 role, and Git. In most cases, you can run a few simple commands and start interacting with your repository. To set up a Cloud9 environment and access your code repository:

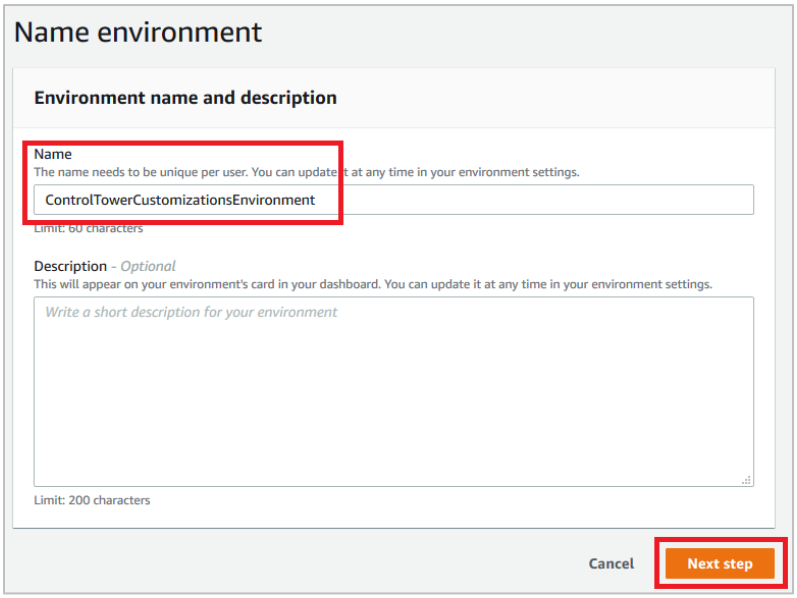
1. Use your management account to open the AWS Cloud9 console.



2. In the AWS Cloud9 console, choose ***Create environment***.



3. In Step 1, Name environment, enter ***ControlTowerCustomizationsEnvironment*** and an optional description for the environment, and then, choose ***Next step***.



4. In Step 2, Configure Settings, configure your environment as follows:
   - In Environment type, choose ***Create a new EC2 instance for environment (direct access***).
   - In Instance type, choose the appropriate instance type for your development environment. For this lab, choose the default of t2.micro.

     Note
     If you are working on a Region where t2.micro instances are not available, use t3.micro – those should be a Free Tier in Regions where t2s are not available.

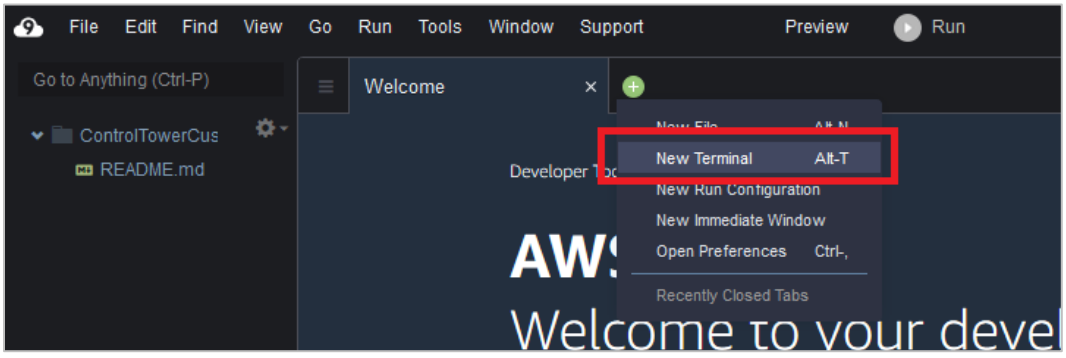
   - Accept the other default settings, unless you have reasons to choose otherwise. (For example, your organization uses a specific VPC, or your AWS account does not have any VPCs configured.) Then, choose ***Next step***.
5. In Step 3, Review, review your settings. Choose ***Previous step*** if you want to make any changes. If not, choose ***Create environment***.

   Note
   Creating an environment and connecting to it for the first time takes a few minutes. Wait until you are connected to your environment before moving to the next steps.

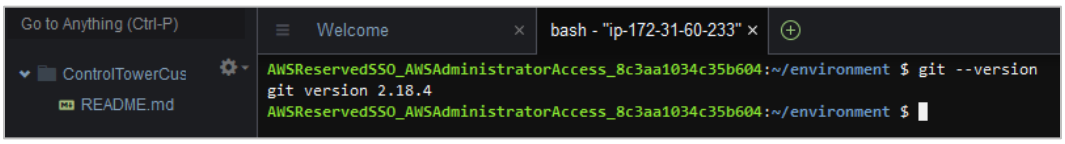
Ensure that Git is installed and is a supported version

In this section, you will check your Git setup:

1. Select the plus (+) on the tab to open a new terminal window, and select ***New Terminal***.



2. Enter `git --version`, and press ***Return***. You should see the following screen.



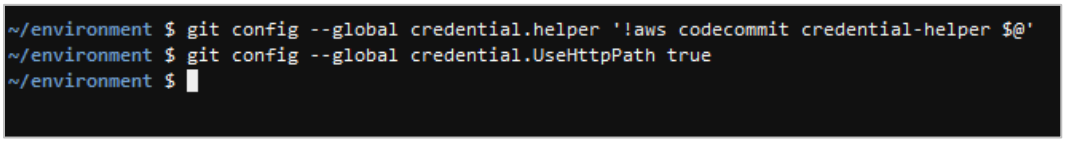
Configure an AWS CLI credential helper

After you create the AWS Cloud9 environment, you can configure am AWS CLI credential helper to manage the credentials for connections to your CodeCommit repository. The AWS Cloud9 development environment comes with AWS managed temporary credentials that are associated with your IAM user. You use these credentials with the AWS CLI credential helper.

1. Open the terminal window, and run the following command to verify that the AWS CLI is installed: `aws --version`.

   If successful, this command returns the currently installed version of the AWS CLI.

2. At the terminal, run the following commands to configure the AWS CLI credential helper for HTTPS connections:
   - `git config --global credential.helper '!aws codecommit credential-helper $@'`
   - `git config --global credential.UseHttpPath true`

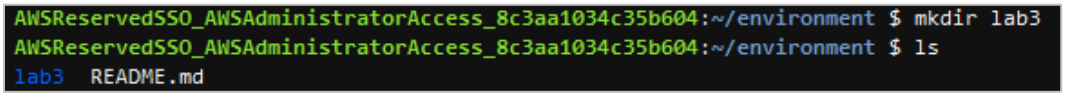


This allows you to clone the CodeCommit repository to this environment.

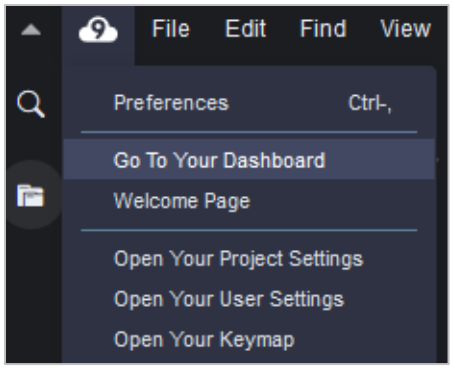
Connect to the CodeCommit

To connect to the repository created as part of the solution, complete the following steps:

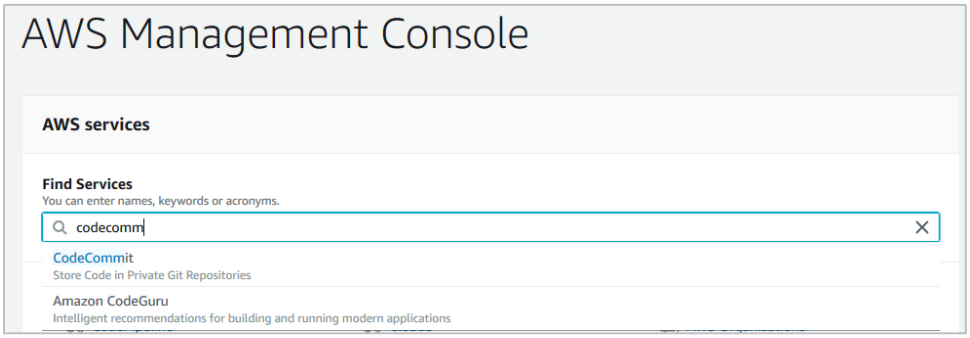
1. In your development environment, you must create a local folder to work with the code. For this example, use a folder named *lab3*. To create the folder, from the terminal, run `mkdir lab3`.



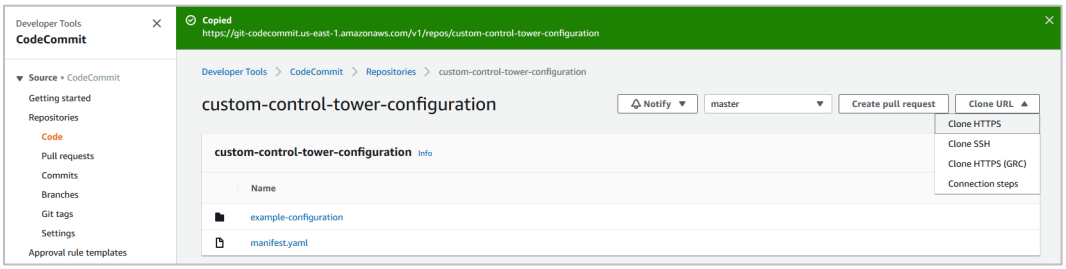
2. Navigate to the folder by running `cd lab3`.
3. Now, you will clone the repository. From the AWS Cloud9 window, select ***Cloud9*** to go to your dashboard. Use a new tab or window to keep your development environment available in the current tab or window.



4. Open the AWS CodeCommit console.



5. In the navigation pane, go to ***Repositories***, and choose ***Code***.
6. For Clone URL, select ***Clone HTTPS***.



7. Return to your development environment. From your terminal, in the lab3 local folder you created, clone the repository you just copied. For example, enter:

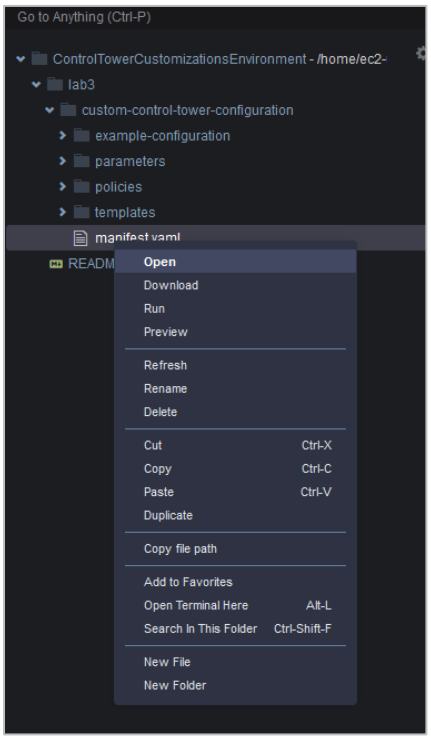
   `git clone https://git-codecommit.us-east-1.amazonaws.com/v1/repos/custom-control-tower-configuration`

```
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:
Cloning into 'custom-control-tower-configuration'...
remote: Counting objects: 13, done.
Unpacking objects: 100% (13/13), done.
```

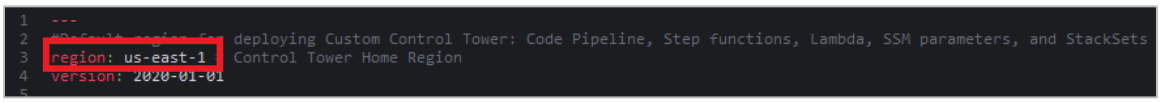
   Be sure to use the URL you copied from CodeCommit.

8. Navigate to custom-control-tower-configuration folder by entering `cd custom-control-tower-configuration`.
9. Next, download the .zip file with the customizations code. You can download the package by running: `wget https://marketplace-sa-resources.s3.amazonaws.com/ctlabs/custom-control-tower-configuration.zip`.
10. Unzip the package into the current folder by running: `unzip custom-control-tower-configuration.zip`.

    If you are prompted to replace the existing manifest.yaml file, enter `y`.
11. After the contents of the .zip file are extracted, delete the file by running: `rm custom-control-tower-configuration.zip`.
12. Locate the manifest.yaml file and open it from the last panel in AWS Cloud9.



    The file is documented with a description about the different sections and parameters available.
13. Update the Region where your AWS Control Tower landing zone was set up.



14. For this lab, ensure that the OUs listed in the file match the *Research* name you gave your organizational unit on Lab1.

```
---
#Default region for deploying Custom Control Tower: Code Pipeline, Step functions, Lambda, SSM parameters, and StackSets
region: us-east-1 # Control Tower Home Region
version: 2020-01-01

# Control Tower Custom Service Control Policies - Additional Preventive Guardrails
organization_policies:
  - name: test-preventive-guardrails
    description: To prevent from deleting or disabling resources in member accounts
    policy_file: policies/preventive-guardrails.json
    #Apply to the following OU(s)
    apply_to_accounts_in_ou: # :type: list
      - Core
      - Research

# Control Tower Custom CloudFormation Resources - Create Additional IAM Role
cloudformation_resources:
  - name: stackset-1
    template_file: templates/describe-regions-iam-role.template
    deploy_method: stack_set
    deploy_to_ou: # :type: list
      - Research
      - Custom
    regions:
      - us-east-1

# Control Tower Config Rule - Additional Detective Guardrails
  - name: stackset-2
    template_file: templates/access_keys_rotated.template
    parameter_file: parameters/access_keys_rotated.json
    deploy_method: stack_set
    deploy_to_ou: # :type: list
      - Research
    regions:
      - us-east-1
      - us-east-2
      - us-west-2
      - eu-west-1
      - ap-southeast-2
```

15. Modify the Region property further down again. Enter the home Region on which you deployed your AWS Control Tower landing zone. In this case, it's ***us-east-1***. Update the Region on line 25, too.

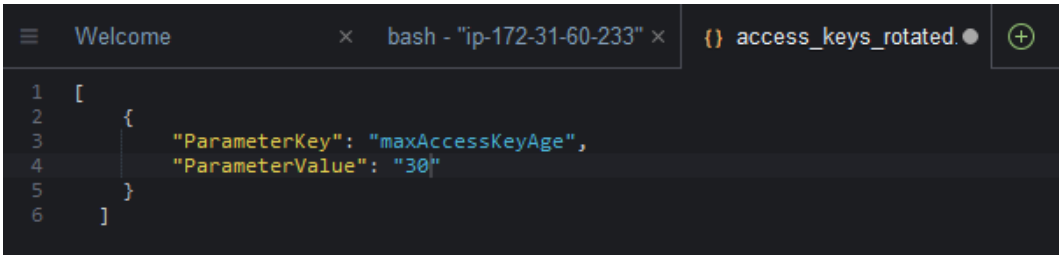
```
# Control Tower Custom CloudFormation Resources - Create Additional IAM Role
cloudformation_resources:
  - name: stackset-1
    template_file: templates/describe-regions-iam-role.template
    deploy_method: stack_set
    deploy_to_ou: # :type: list
      - Research
    regions:
      - us-east-1
```

16. Make sure that the ***deploy_to_ou*** property in ***# Control Tower Config Rule*** includes "Research," as shown in the example:

```
# Control Tower Config Rule - Additional Detective Guardrails
  - name: stackset-2
    template_file: templates/access_keys_rotated.template
    parameter_file: parameters/access_keys_rotated.json
    deploy_method: stack_set
    deploy_to_ou: # :type: list
      - Research
    regions:
      - us-east-1
      - us-east-2
      - us-west-2
      - eu-west-1
      - ap-southeast-2
```

17. Save your changes.
18. Look for the ***access_keys_rotated.json*** file in the ***parameters*** folder.

19. Replace the ***ParameterValue*** of ***24*** with ***30***. This increases the maximum key rotation limit from 24 to 30.



20. Save your changes.

Now, you are ready to commit your changes and deploy the customizations.

Commit and deploy the customizations

Before you begin, make sure you are in the correct folder in your AWS Cloud9 terminal. Type ***pwd*** to see the current directory:

```
AWSReservedSSO_AWSAdministratorAccess_eb3f02a162a86cdd:~/environment/lab3/custom-control-tower-configuration (master) $ pwd
/home/ec2-user/environment/lab3/custom-control-tower-configuration
```

You should be in ***custom-control-tower-configuration***. Check-in the customizations to your CodeCommit Repository by running the following commands:

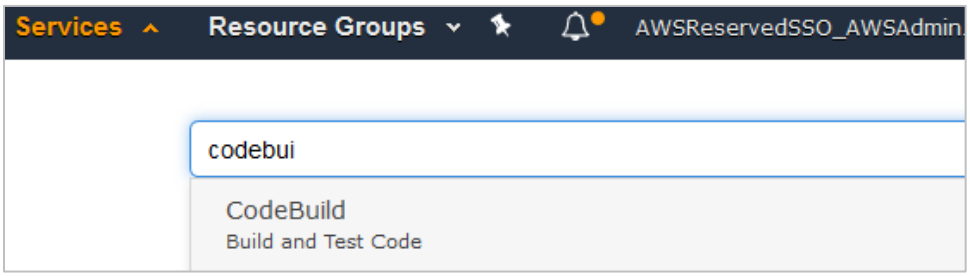
- `git status`
- `git add -A`
- `git commit -m 'Initial check in'`
- `git push`

You should see something similar to this on your terminal:

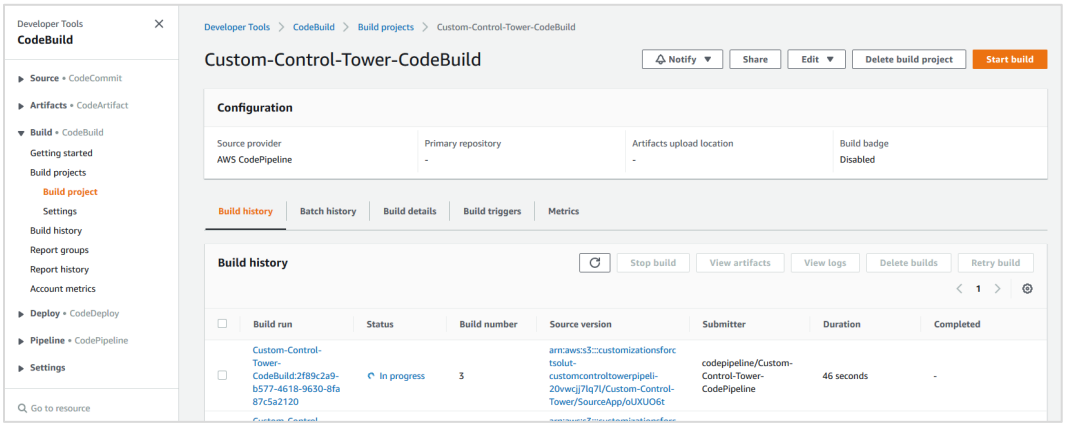
```
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-control-tower-configuration (master) $ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   manifest.yaml

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        parameters/
        policies/
        templates/

no changes added to commit (use "git add" and/or "git commit -a")
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-control-tower-configuration (master) $ git add -A
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-control-tower-configuration (master) $ git commit -m 'Initial Check-In'
[master 2b2c751] Initial Check-In
 Committer: EC2 Default User <ec2-user@ip-172-31-60-233.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 5 files changed, 119 insertions(+), 6 deletions(-)
 create mode 100644 parameters/access_keys_rotated.json
 create mode 100644 policies/preventive-guardrails.json
 create mode 100644 templates/access_keys_rotated.template
 create mode 100644 templates/describe-regions-iam-role.template
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-control-tower-configuration (master) $ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 1.77 KiB | 604.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0)
To https://git-codecommit.us-east-1.amazonaws.com/v1/repos/custom-control-tower-configuration
   5585c9a..2b2c751  master -> master
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-control-tower-configuration (master) $
```

Congratulations! You successfully deployed customizations for AWS Control Tower, added your customizations, and deployed them to your AWS Control Tower environment. AWS CodeBuild should be running.

21. To see a build in progress, navigate to AWS CodeBuild using your other tab.



22. Watch the build in progress:



23. When the build completes, proceed to the next task.

Task 3: View deployed customizations

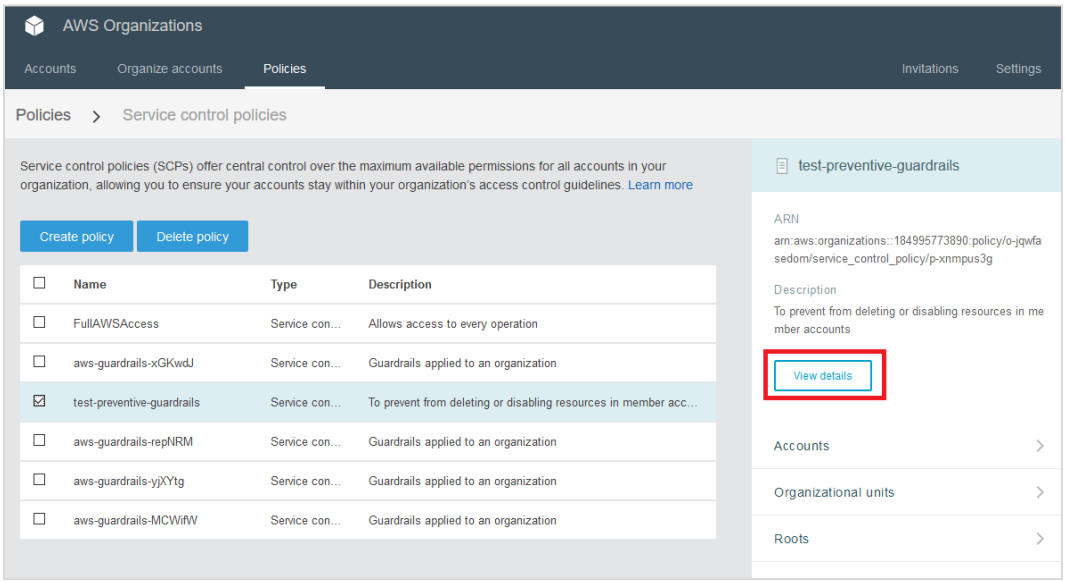
In this task, you will verify the customizations that you just deployed.

Note
The previous build must be completed before you can see the different created customizations.

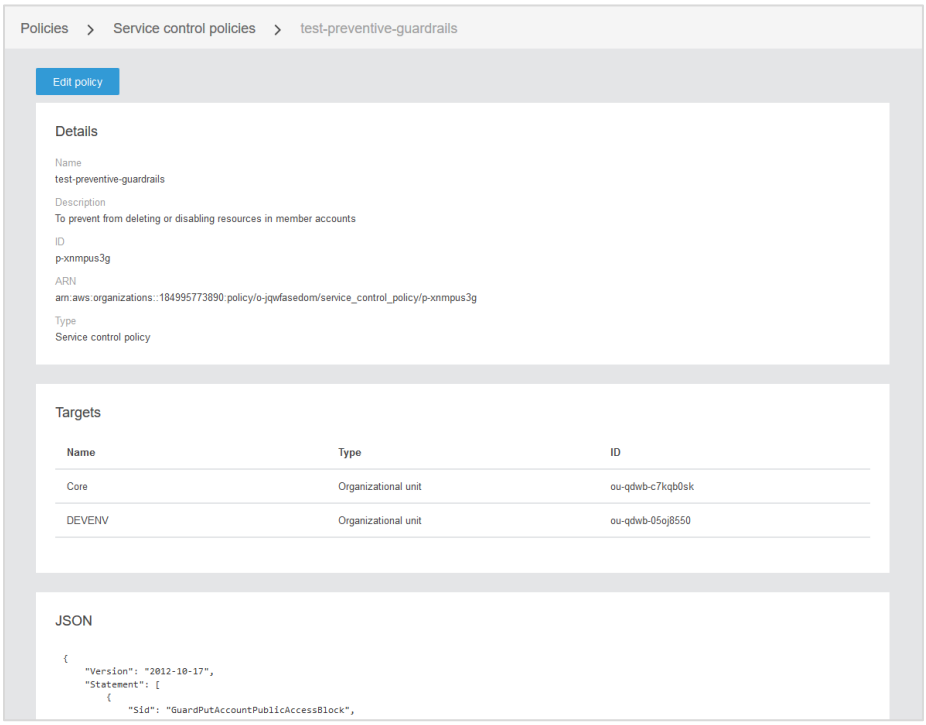
View preventive guardrails

To access and review service control policies in AWS Organizations:

1. In your management account, open ***AWS Organizations***, and select ***Policies***.
2. Select ***Service Control Policies***, choose ***test-preventive-guardrails***, and choose ***View details***.



3. Explore the details.

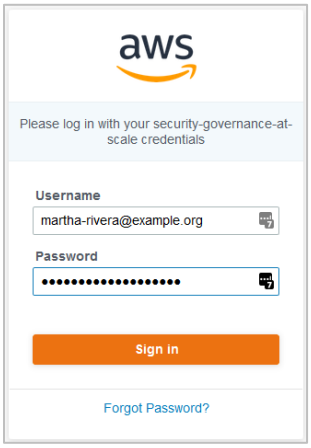


This example shows the *policies/preventive-guardrails.json* policy that you checked in to the CodeCommit repository.

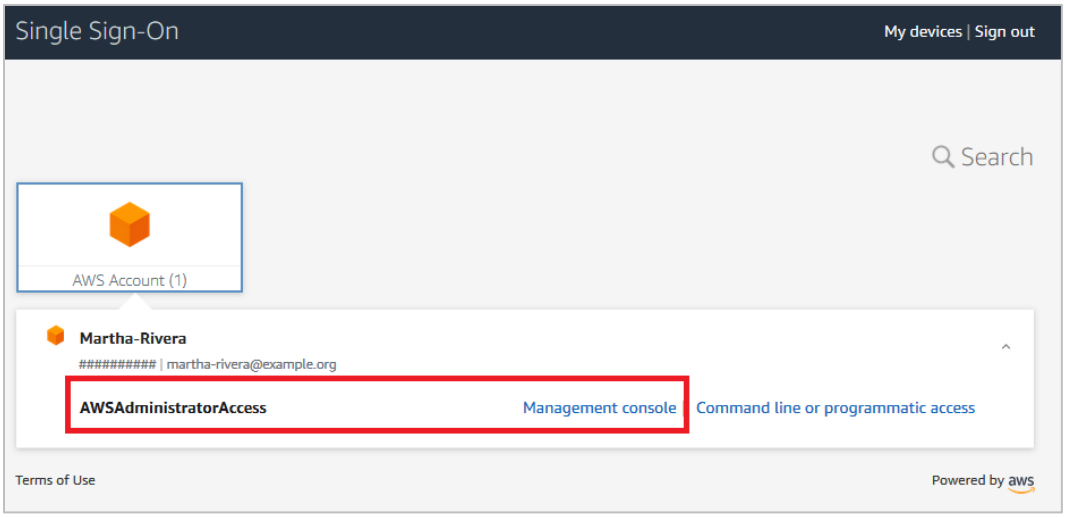
View detective guardrails

To access the AWS Config console and explore rules:

1. Log in with your Martha Rivera account. This account is a member of the organizational unit deployed with the Research customization. You provisioned this account using Account Factory on Lab1.
2. Proceed to the user portal screen, and enter Martha's credentials

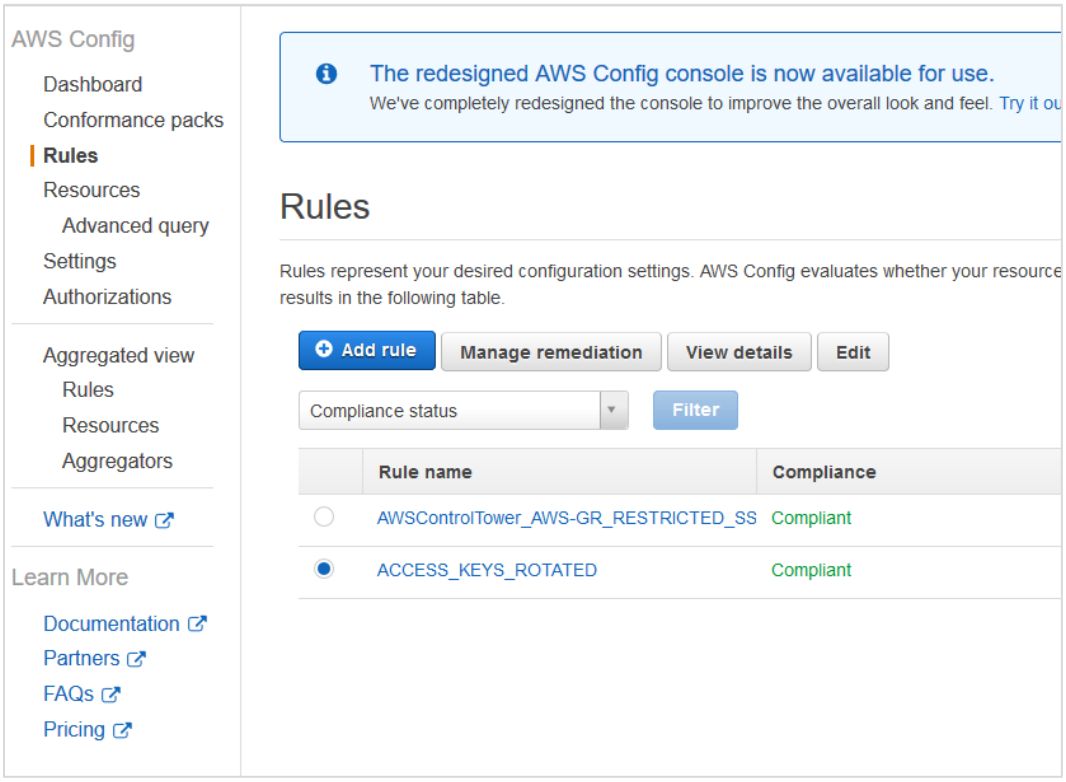


3. Choose the ***AWSAdministratorAccess*** role.

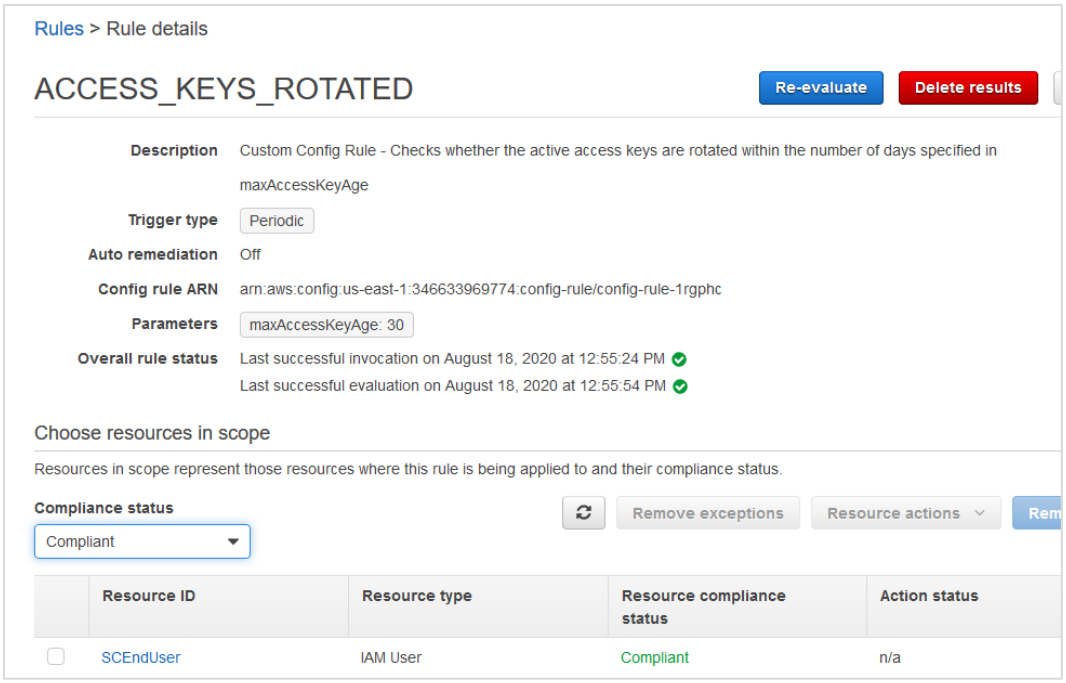


4. Open the ***AWS Config*** console.

5. In the navigation pane, choose ***Rules*** to list the Config Rules (detective guardrails) deployed.



6. Choose ***ACCESS_KEYS_ROTATED*** to view the details of the AWS Config rule.

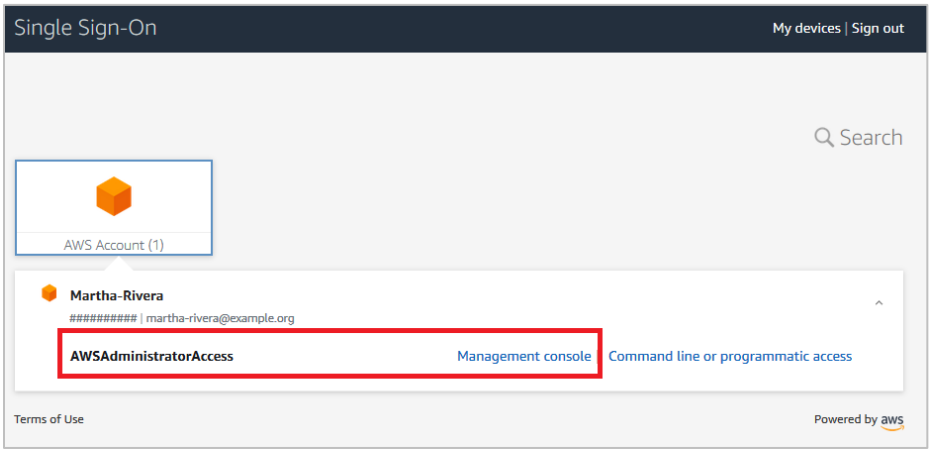


This AWS Config rule is deployed by the CodePipeline. You can verify this using the AWS CloudFormation console. The user created in Lab 2 is listed as Compliant.

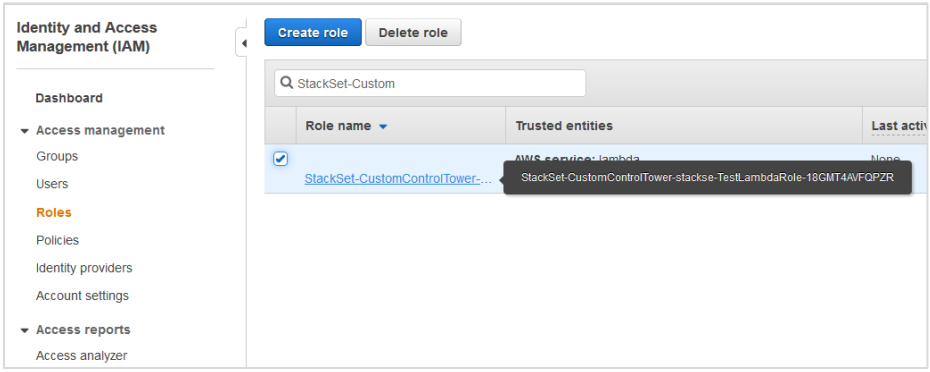
View IAM role

Access IAM console to review custom roles:

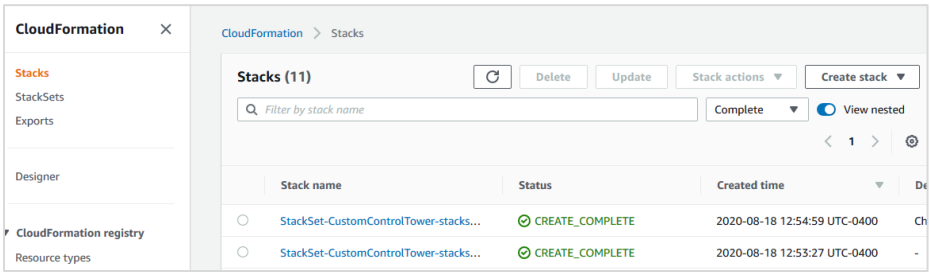
1. Log in to a provisioned account (an account such as Research or Custom OU) with AWSAdministratorAccess. You can use Martha's account for this step.



2. Open the IAM console.
3. In the navigation pane, go to ***Access management***, and choose ***Roles***.
4. Search for a role that starts with *StackSet-CustomControlTower-**.



5. This role is deployed by the CodePipeline. While you are on logged in to this account, you can also verify the AWS CloudFormation resources that created this role.



Lab 4: Lab Decommission Instructions

In this section, you will learn where resources are provisioned as part of each lab and how to decommission them to avoid additional costs. The following sections provide you with the necessary steps to clean up the resources and services used in this training.

Important

- Do not perform the cleanup activities after each individual lab, because the resources are reused across the labs.
- After you have completed the three labs or if you no longer need the accounts and resources used in them, follow the instructions.
- Decommission resources in reverse order: starting with Lab 3, then Lab 2, and then Lab 1. Finally, use the clean-up instructions to delete the AWS Control Tower landing zone.
- Decommission in reverse order because different resources depend on each other. Not proceeding in reverse order will result in errors while trying to delete some of the resources.
- Follow the dedicated sections on how to close the four AWS accounts used in the labs.

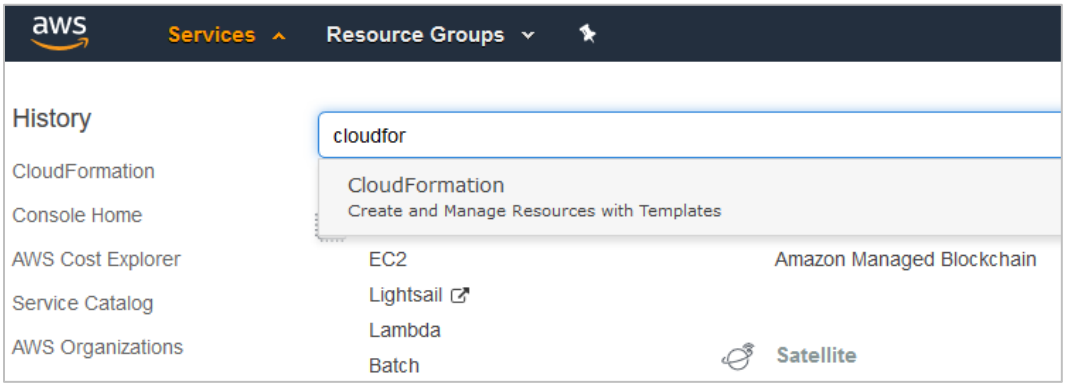
Decommission and clean up Lab 3

In Lab 3, you deployed the AWS Control Tower customization framework solution and deployed three customizations. Follow these steps to remove the resources:

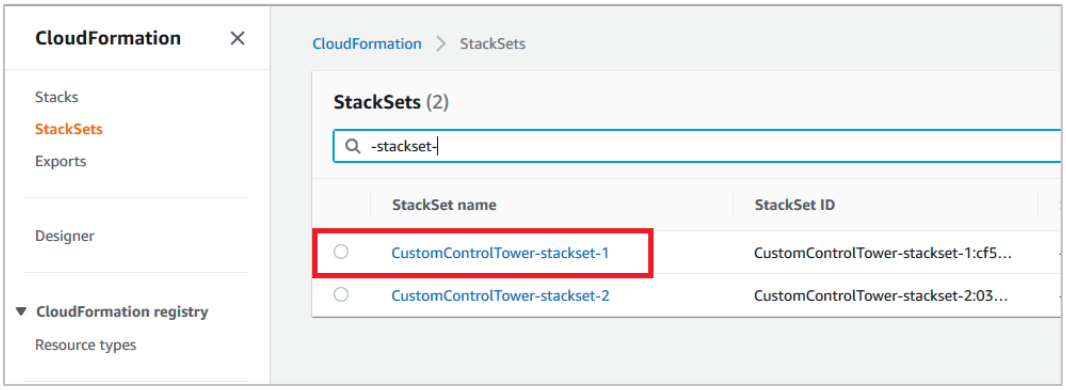
Delete the AWS CloudFormation StackSets

To delete the StackSets:

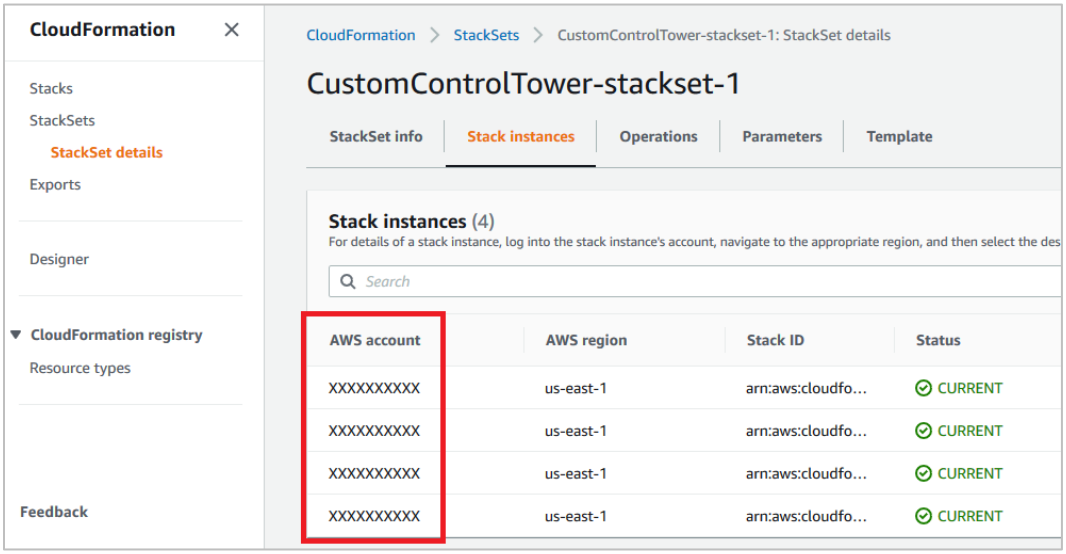
1. Open the ***AWS CloudFormation*** console.



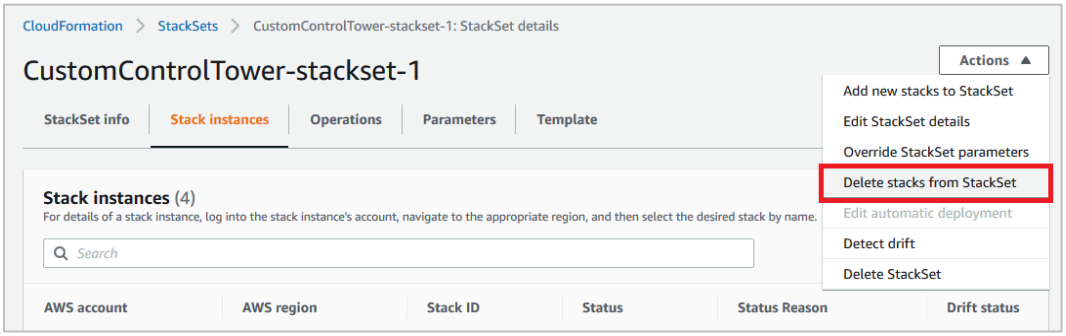
2. In the navigation pane, got to ***StackSets***, select the ***stackset-1*** deployed by this solution.



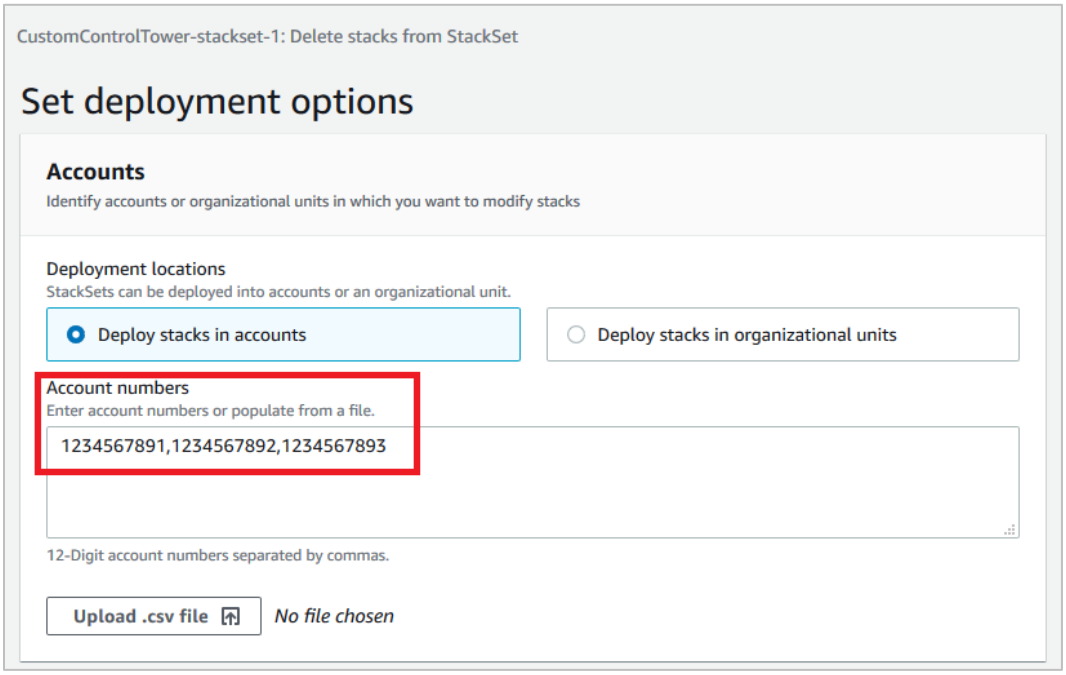
3. Copy the AWS account number. You will use it in next step.



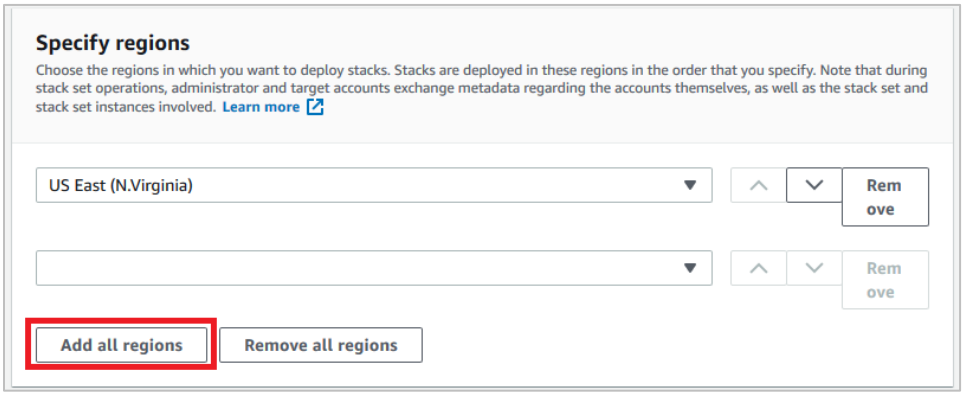
4. Choose ***Actions***, and select ***Delete stacks from StackSet***.



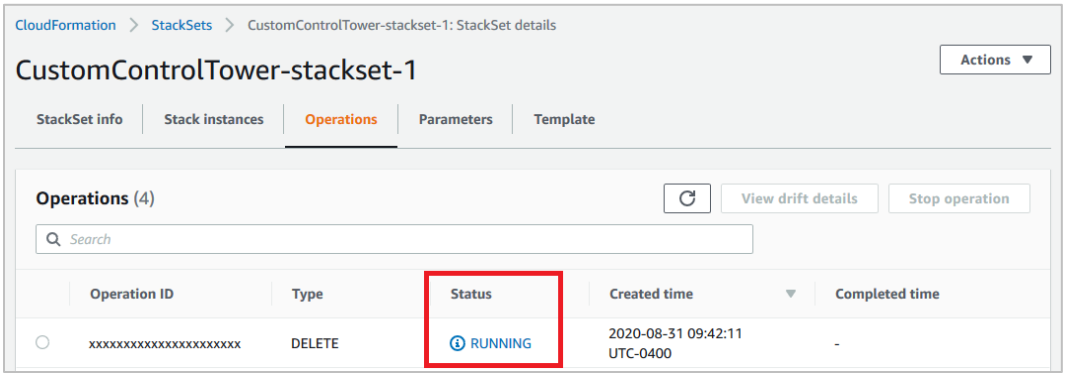
5. In Accounts, choose the ***Account numbers*** box, and enter in the AWS account number you previously copied.



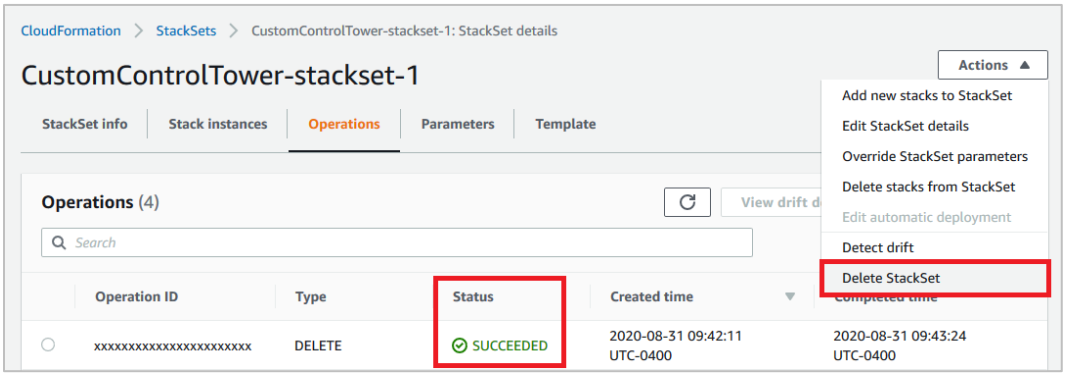
6. In Specify regions, choose ***Add all regions***.



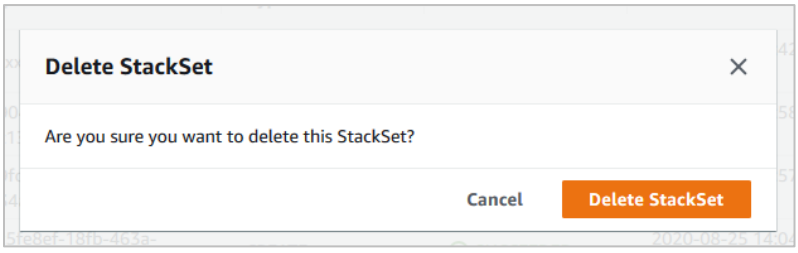
7. Choose ***Next***, and then choose ***Submit***.
8. Refresh the screen and wait for the Operation ID status to change to ***SUCCEEDED***.



9. After the status shows ***SUCCEEDED***, choose ***Actions***, and select ***Delete StackSet***.

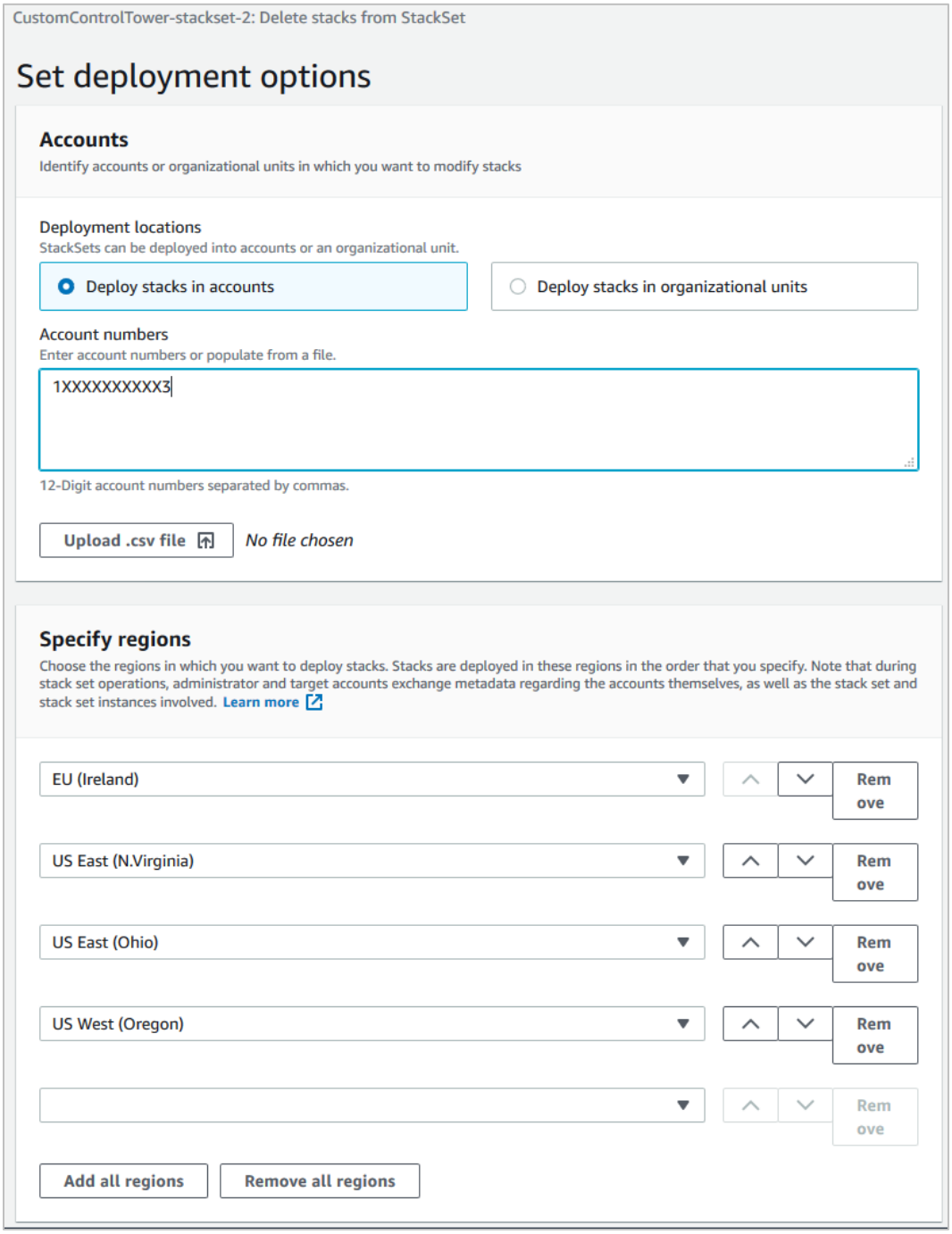


10. Confirm the deletion when prompted.



11. Repeat these steps for stackset-2.

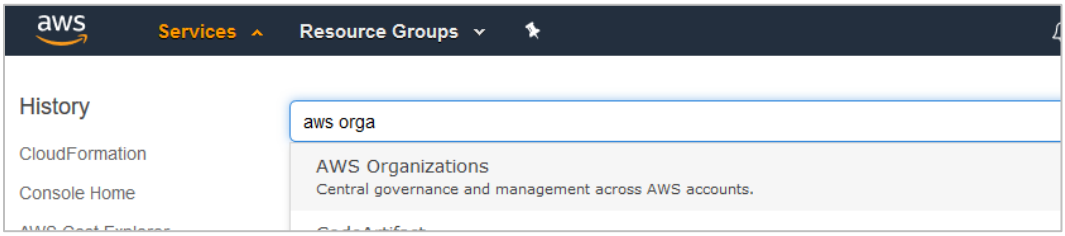
    Note
    For this StackSet, you might see only one account and more Regions.



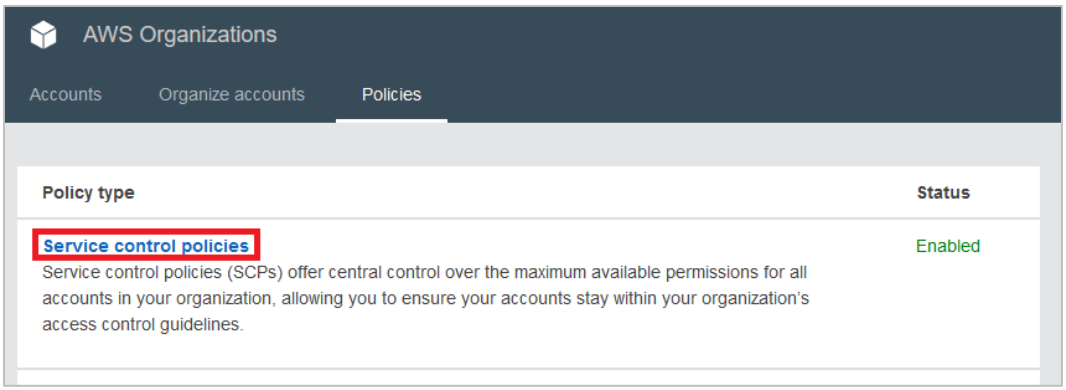
Detach and delete the service control policies

Next, you will detach and delete the SCPs:

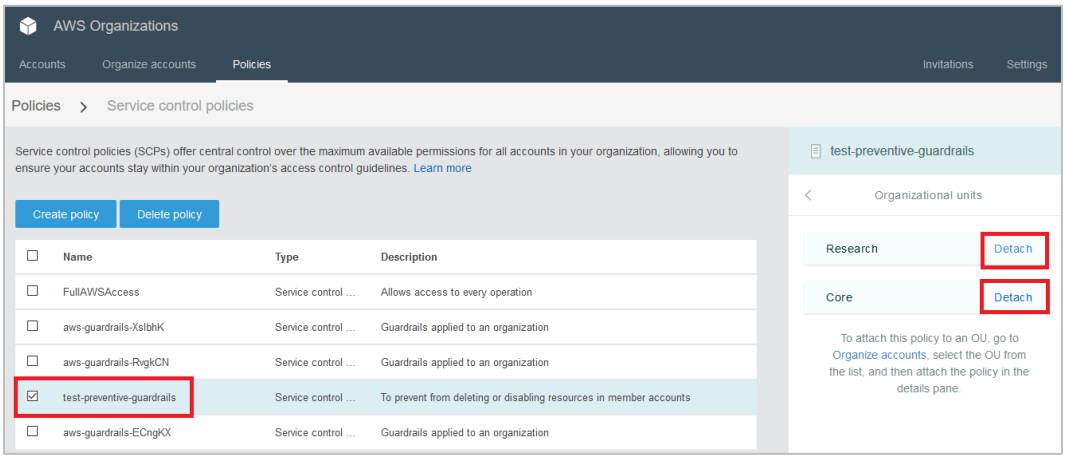
1. Open the AWS Organizations console.



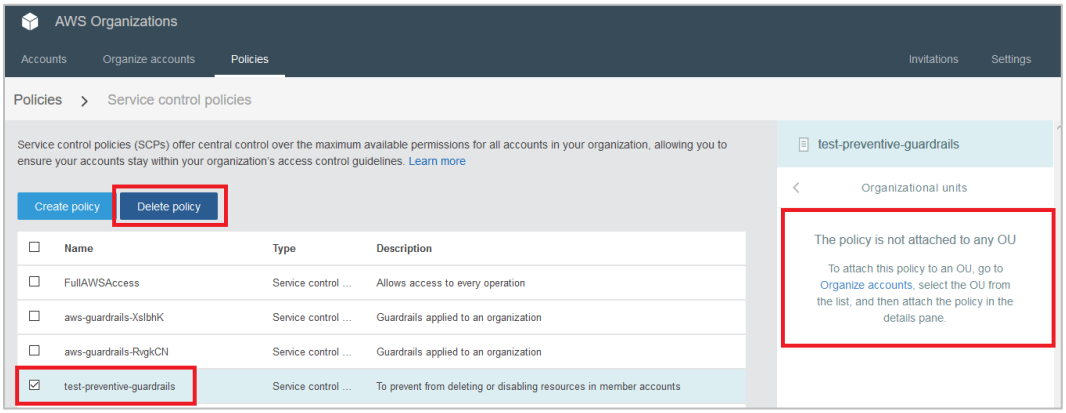
2. Select the ***Policies*** tab.



3. Select ***test-preventative-guardrails***, and choose ***Organizations units***.
4. Choose ***Detach*** for all OUs.



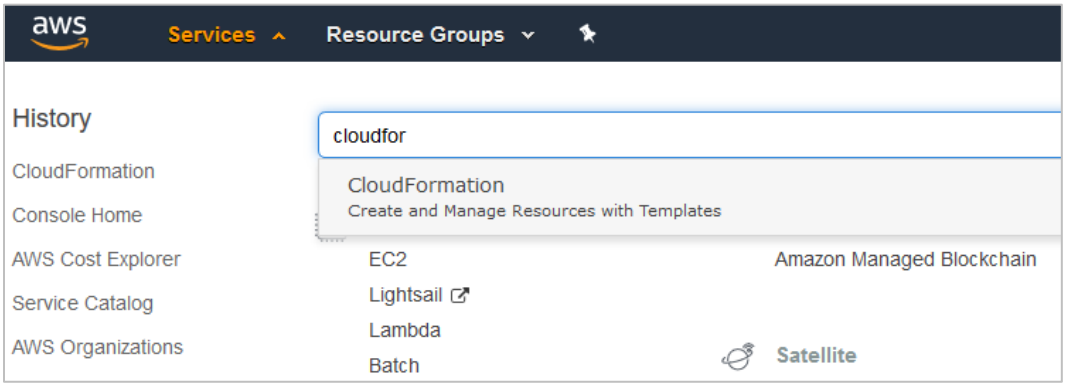
5. Select ***Delete Policy*** to delete the service control policy.



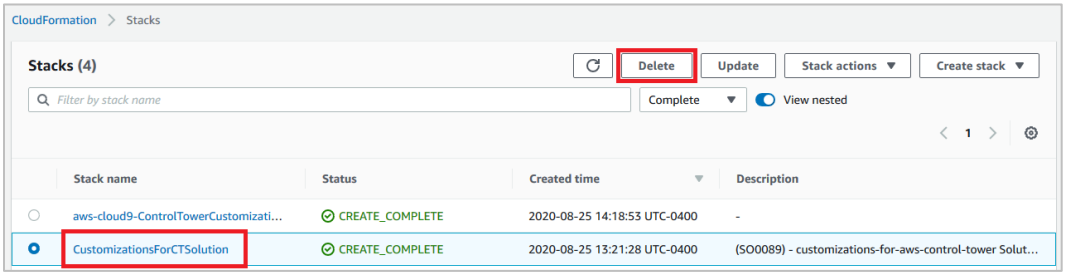
Delete the customization solution stack from Lab 3

Now, you will delete the customization solution stack:

1. Open the AWS CloudFormation console.



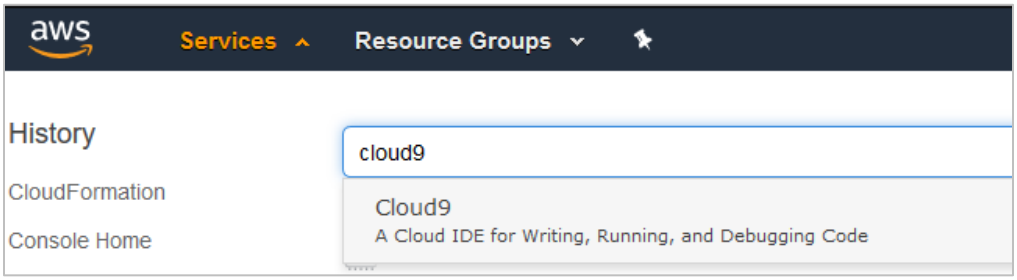
2. Select the Customization framework stack you deployed earlier.
3. Choose ***Delete*** to delete the solution.



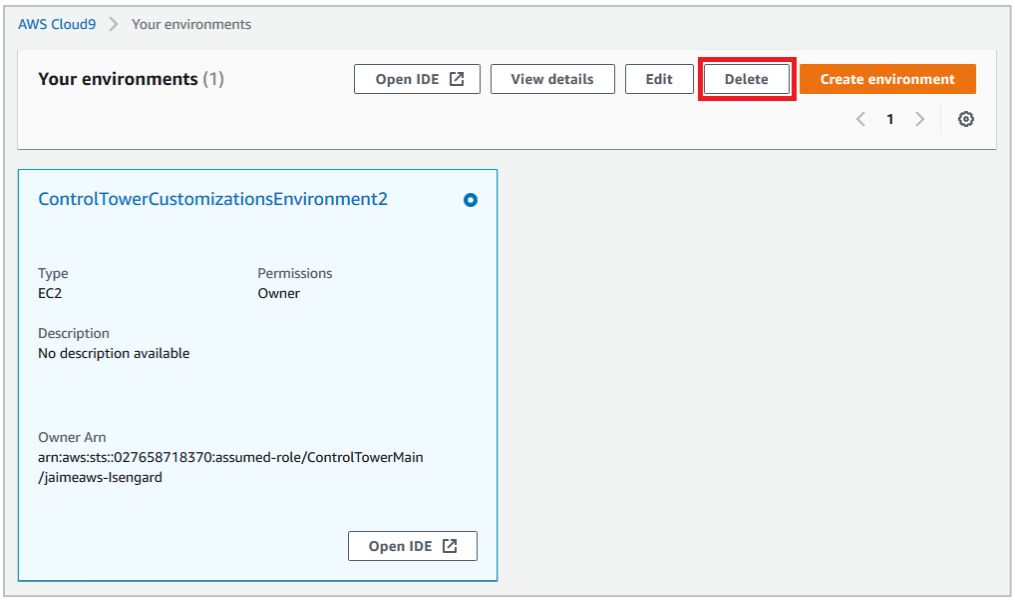
Delete AWS Cloud9 development environment

The last step in decommissioning Lab 3 is to delete the AWS Cloud9 development environment:

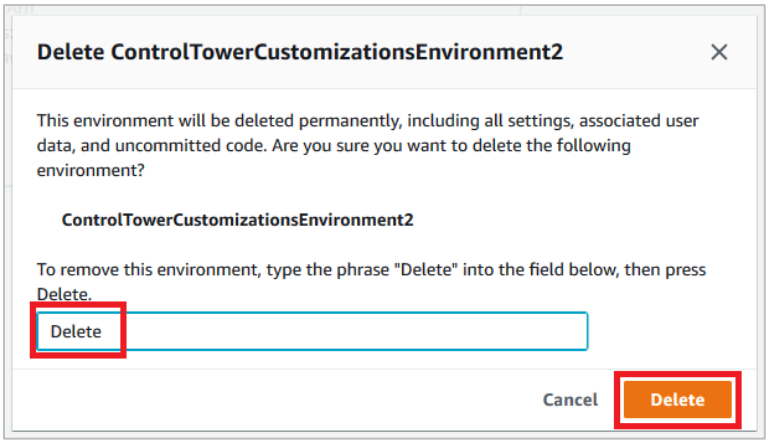
1. Open the AWS Cloud9 console.



2. Delete the environment you used for Lab 3.

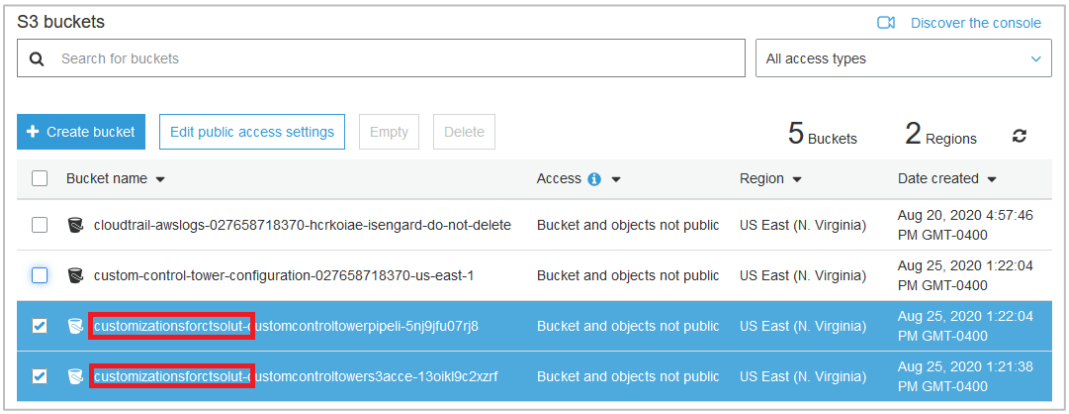


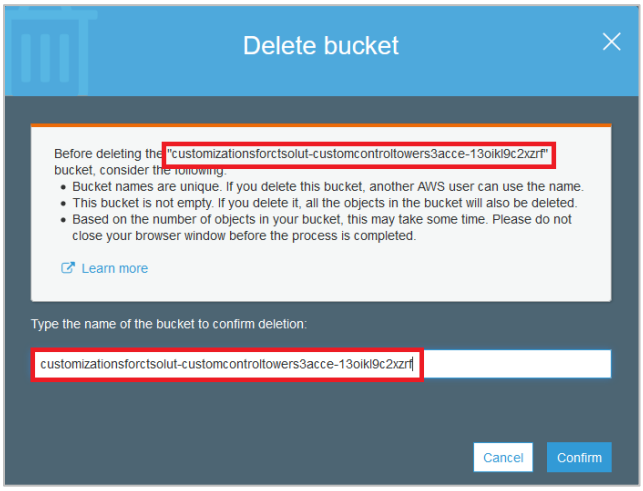
3. Enter ***Delete*** when prompted, and confirm the deletion.



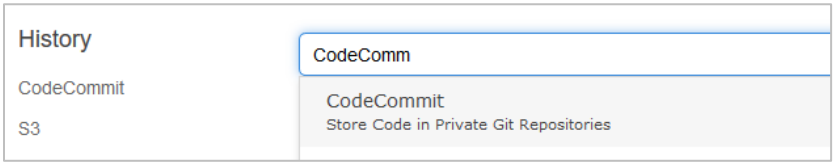
Note
The S3 bucket and AWS CodeCommit repository that are created as part of this solution are not deleted when the stack is deleted. You must delete them manually.

4. Delete the S3 buckets with names that start with *customizationforctsolut*.

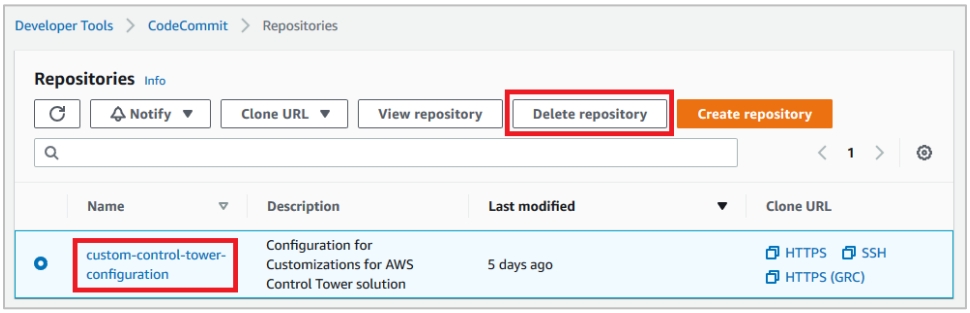




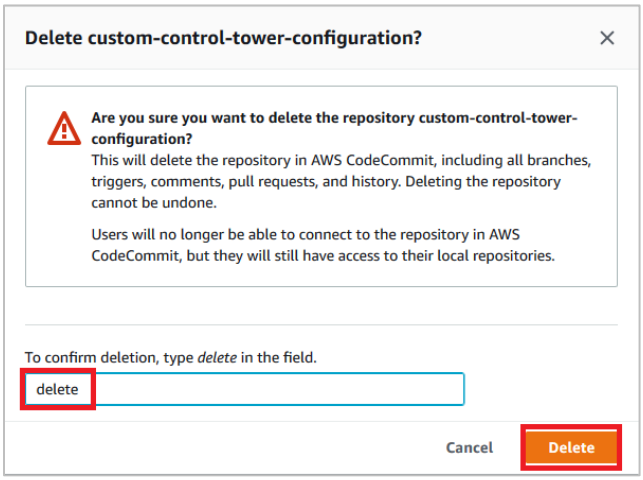
5. Navigate to the CodeCommit repository.



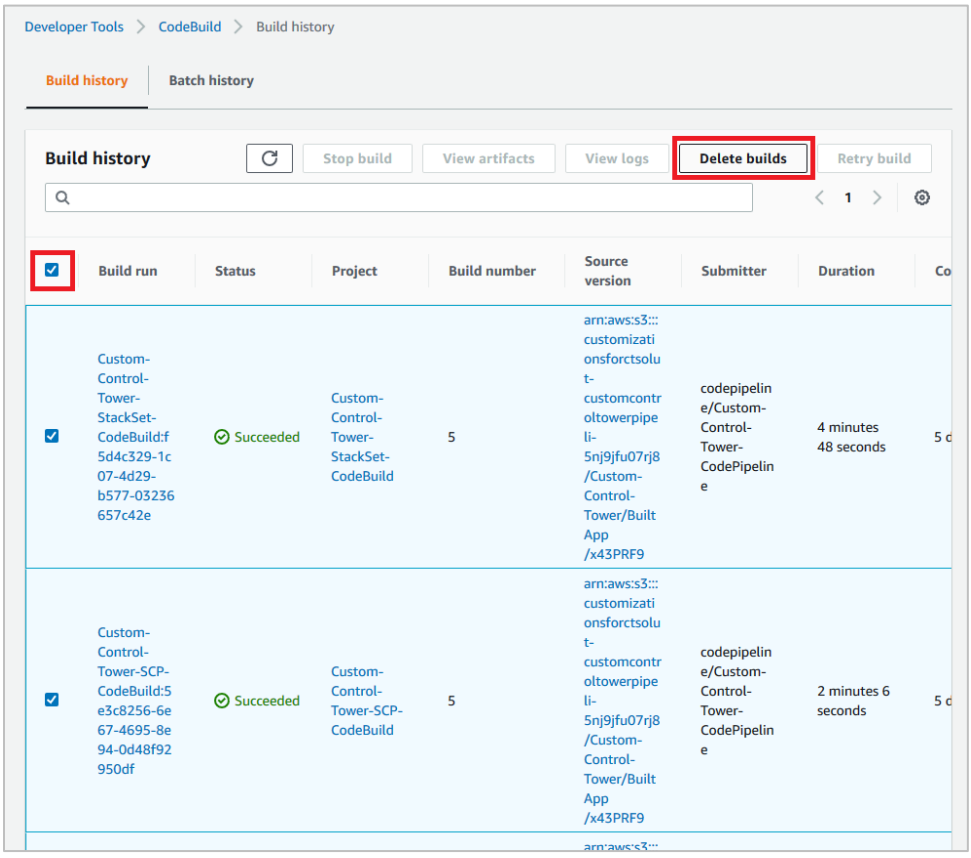
6. Select ***custom-control-tower-configuration*** and choose ***Delete repository***.



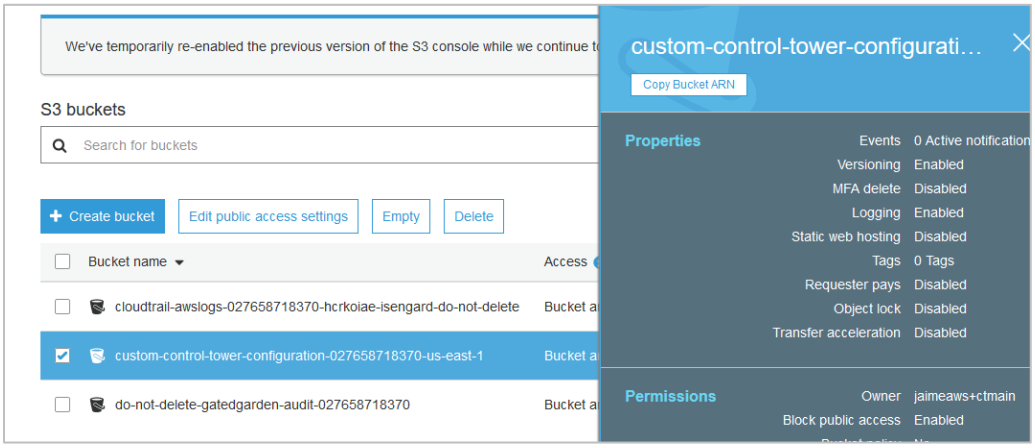
7. Enter ***delete***, and confirm the deletion.

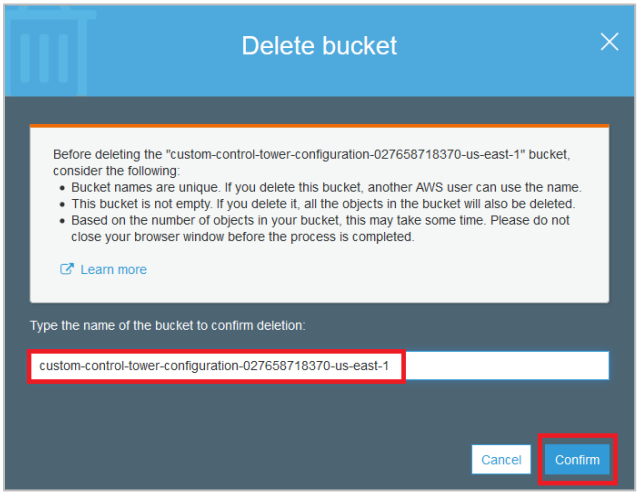


8. From AWS CodeBuild, delete the builds:



9. Now, return to Amazon S3, and delete the bucket that CodeCommit uses for configuration.





Decommission and clean up Lab 2

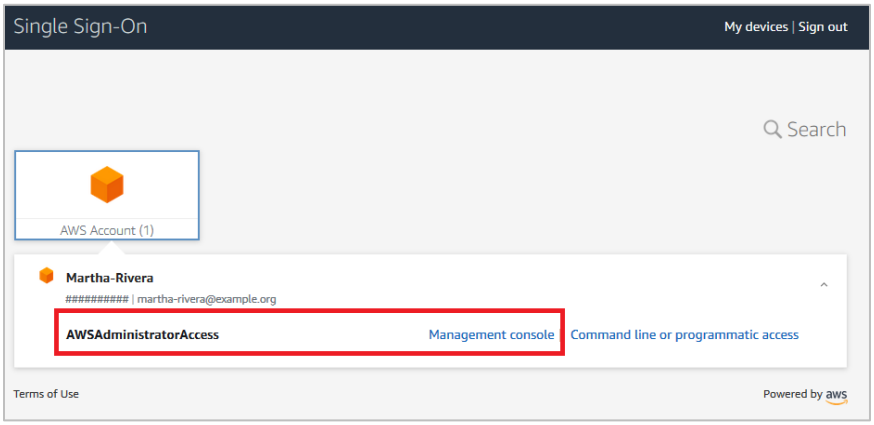
In Lab 2, you created AWS Service Catalog portfolios and launch constraints. You also used an AWS CloudFormation stack to provision IAM roles, user groups, and users to implement the constraints.

Resources that are deployed as part of this lab, such as a custom VPC and Amazon EC2 instances, can add costs to your account if left undeleted. We highly recommend that you clean up these resources after you are done with this lab, unless you plan on continue to use them.

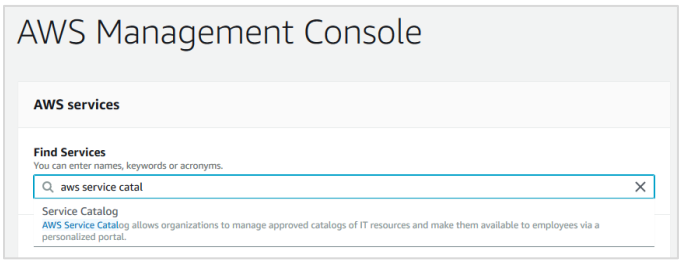
Clean up Amazon EC2 and Amazon S3 resources

If you launched an Amazon EC2 instance or Amazon S3 resources with the secondary account, SCEndUser, delete the resources using the following instructions. You can skip this step if you did not provision the resources.

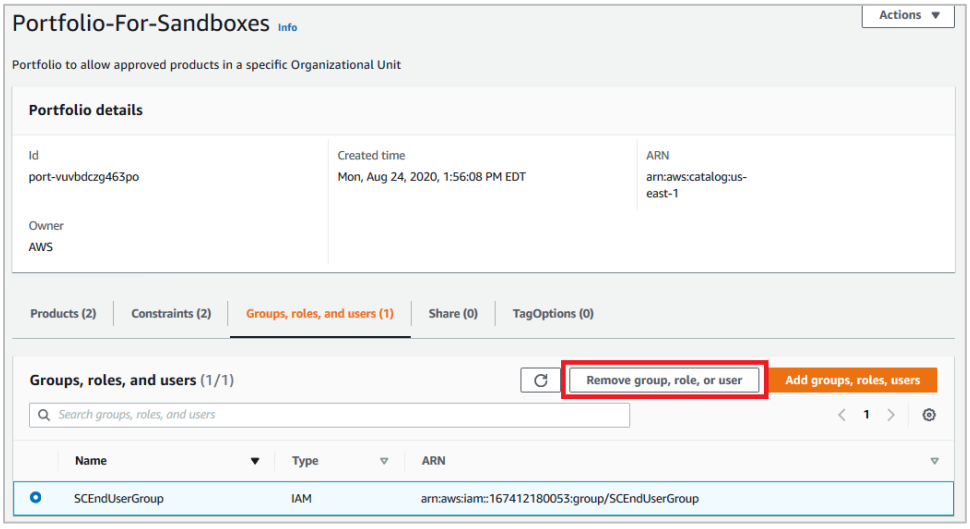
1. Log in to the Martha Rivera account as administrator.



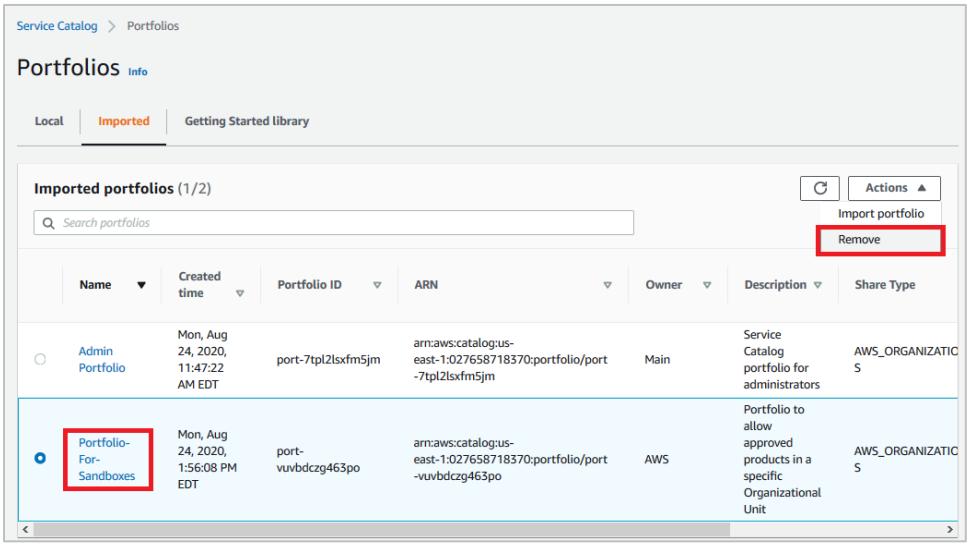
2. Open the AWS Service Catalog console.

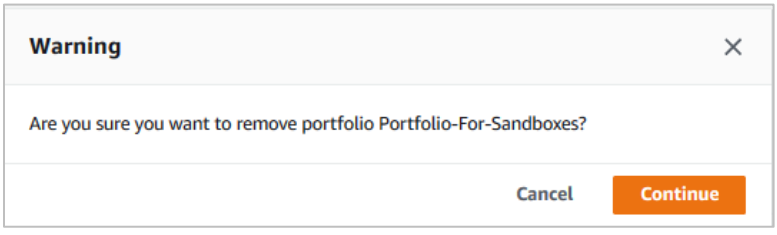


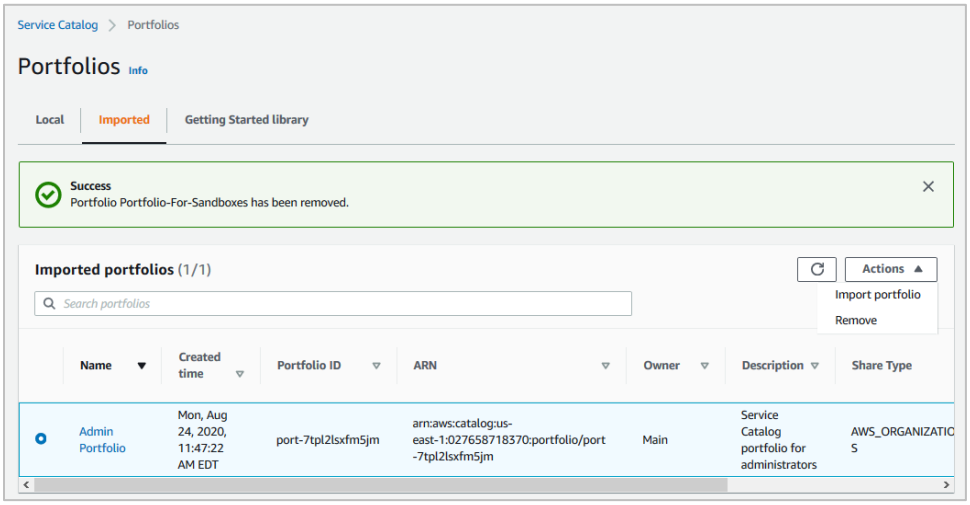
3. Select the ***Provision*** product list.
4. Choose ***Filter by***, and select ***Account***.
5. Choose the provisioned product that you created as part of this lab. (If the list is empty, skip the next two steps).
6. To delete the provisioned product, choose ***Actions***, and then choose ***Terminate***.
7. Wait for the ***TERMINATE_PROVISIONED_PRODUCT*** status change to **Succeeded**.
8. Clean up the AWS Service Catalog product and portfolios. *These resources do not add any expenses, if not used.*
9. Remove all associated groups, users, and roles to the portfolio; otherwise, you won't be able to delete the portfolio.



10. Remove the portfolio.





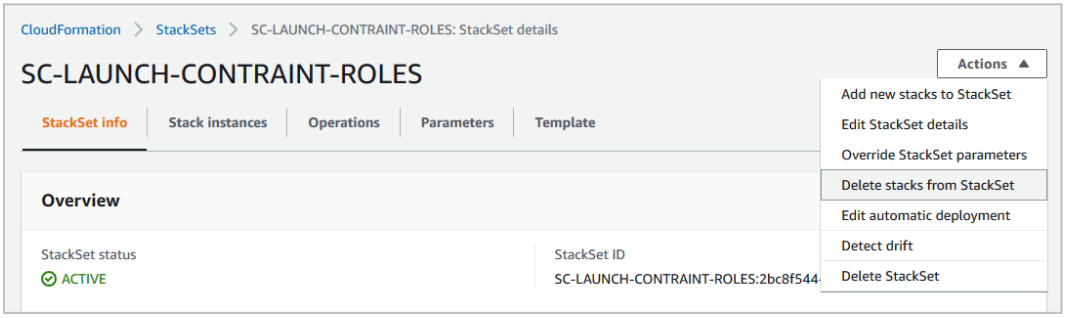


Delete roles, users, and groups created by AWS CloudFormation

The final decommission task for Lab 3 is to delete the roles, users, and groups created by AWS CloudFormation:

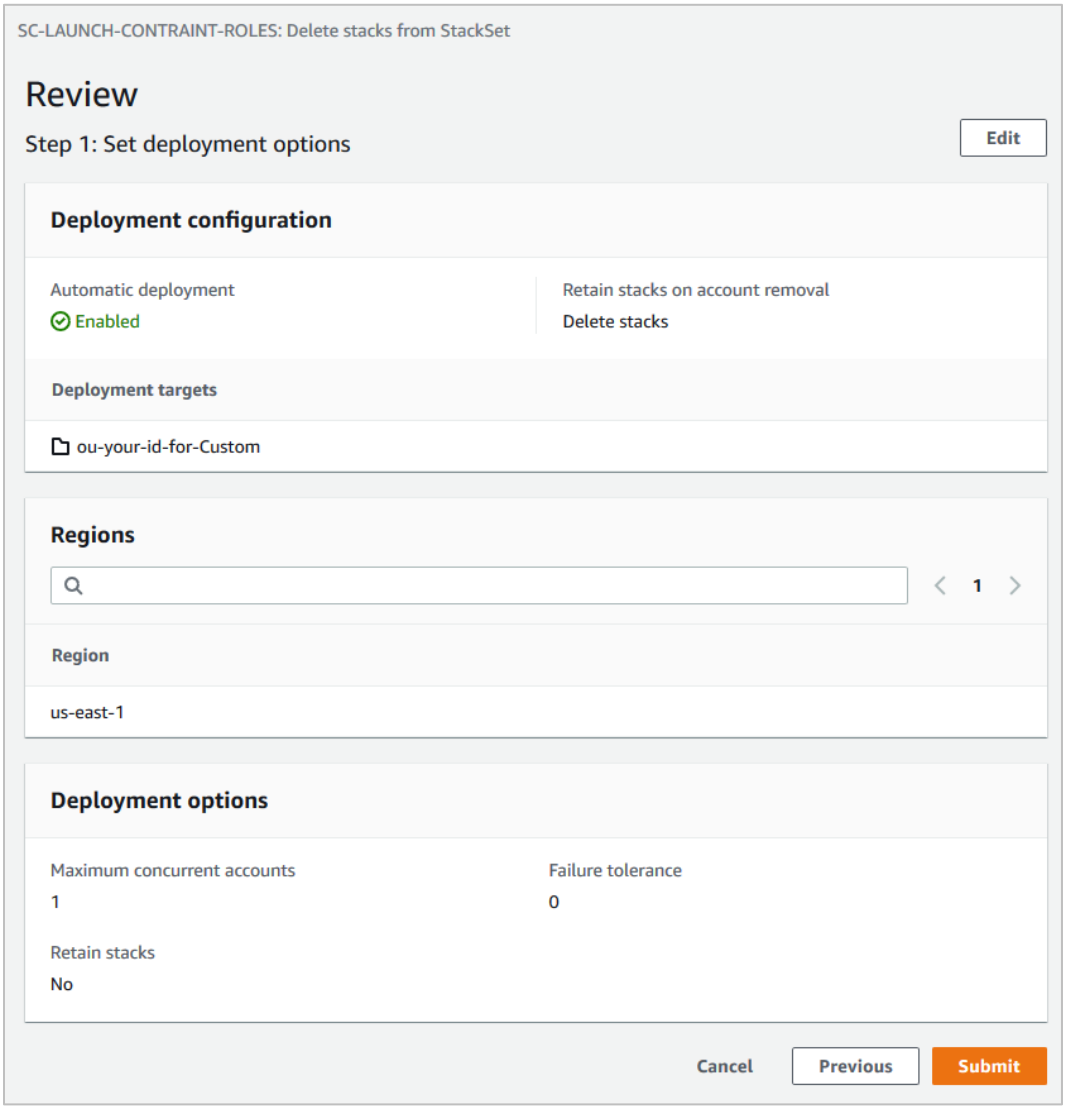
1. Use your management account to open the AWS CloudFormation console.
2. Select ***StackSets***, and choose ***SC-LAUNCH-CONSTRAINT-ROLES***.

   Note
   You will need the organizational unit that you used in Lab 2, when you shared the portfolio. It should be the ID of the Research OU.
3. After you select the ***SC-LAUNCH-CONSTAINT-ROLES StackSet***, choose ***Actions***, and then choose ***Delete stacks from StackSet***.

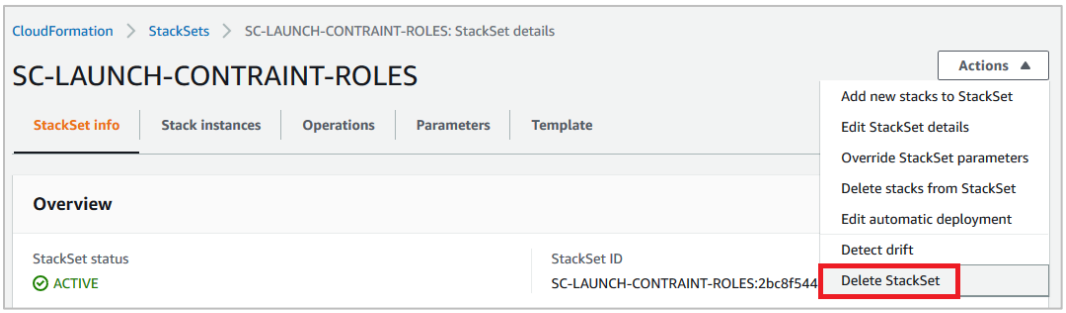


4. Enter the ***Research*** OU ID, and then choose ***Add all regions***.
5. Choose ***Next***, and then choose ***Submit***.

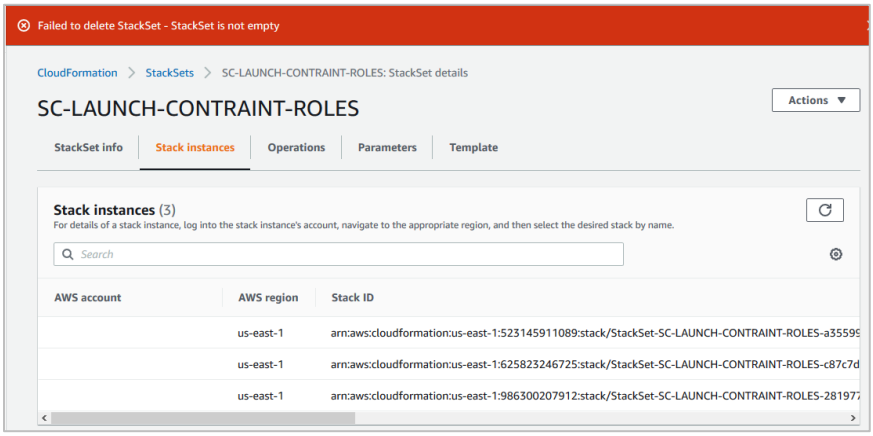
   Note
   You might need delete more OU IDs, depending on how many OUs you shared the portfolio with.



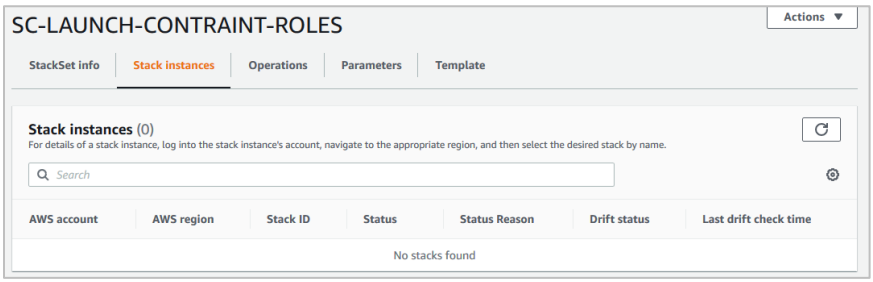
6. Delete the StackSet.



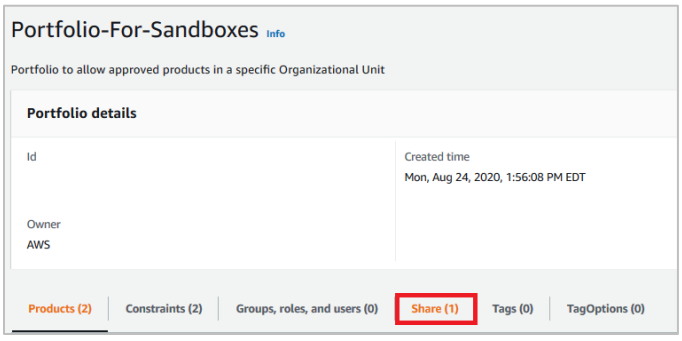
7. You might get an error like the one shown.

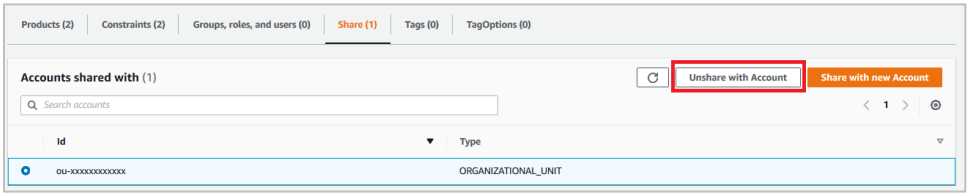


8. Make sure that there are no instances listed, so you can delete the StackSet. You can check the AWS Account numbers, see the OU that they belong to, and then click ***Delete stacks from StackSet*** until the list is empty:

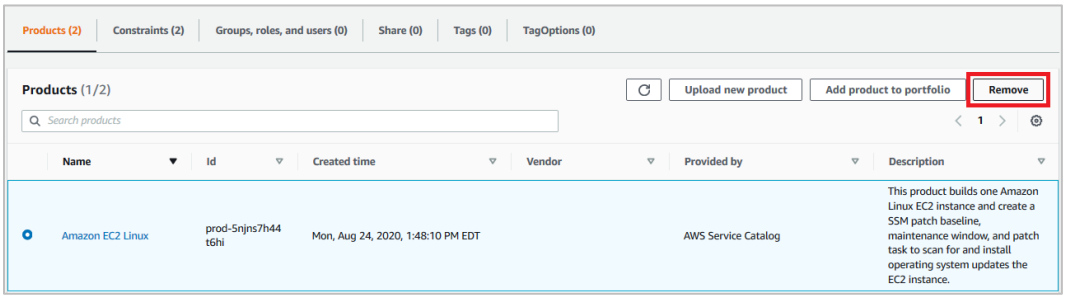


9. Delete the ***SC-LaunchConstraintRole Stack***.
10. Delete the portfolios that were created as part of this lab. You will need to unshare them and remove any constraints or user, group, or role access associated to them





11. Delete the products in the portfolio.



12. Delete the portfolio.
13. Do this on your member account (Martha's account) and on your management account.

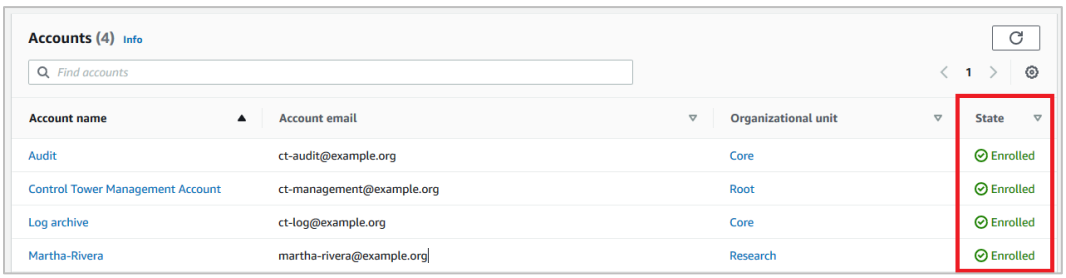
Decommission and clean up Lab 1 and Lab 0

In Lab 0, you set up a landing zone using AWS Control Tower. In Lab 1, you created and enrolled accounts into AWS Control Tower. Here are the steps and references that you can use to decommission all the resources that were created as part of Labs 0 and 1.

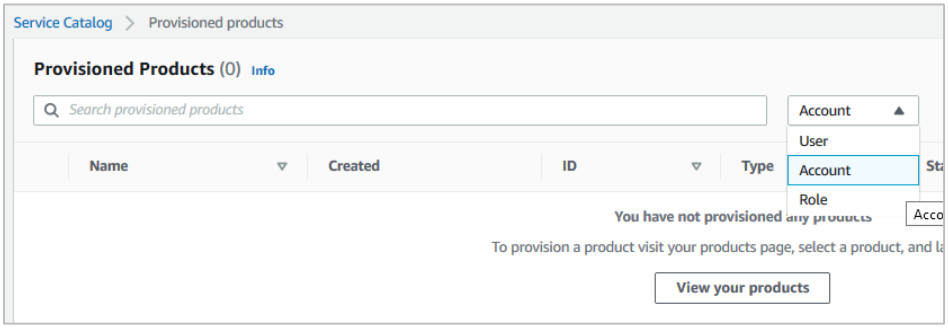
Unmanage and delete AWS Control Tower accounts

The detailed steps to unmanage and delete accounts enrolled with AWS Control Tower are available here in the documentation portal. The logical flow is:

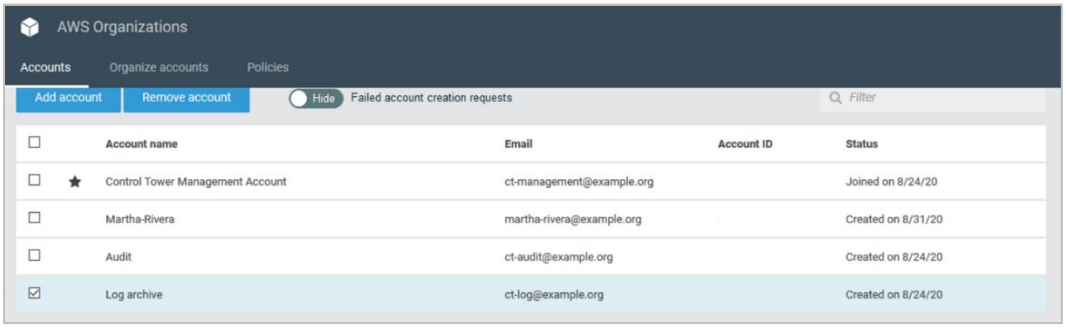
1. To unmanage an account, see https://docs.aws.amazon.com/controltower/latest/userguide/account-factory.html#unmanage-account.
2. From the AWS Service Catalog, terminate your provisioned (products) accounts. You will need to do one at a time and wait for each individual termination to finish before moving to the next one. Each termination process can take several minutes.
3. Return to the AWS Control Tower accounts console and ensure that your accounts show unregistered as soon as the termination process finishes.
4. Do this for the Martha Rivera account. Make sure that only the shared accounts remain enrolled.



5. If you see outstanding accounts, return to the AWS Service Catalog using your management account. Select ***Provisioned Products***. Then, choose ***Filter by Account*** to see additional (accounts) products. Delete the accounts until only the shared accounts (audit, log archive, and management) remain re-enrolled.

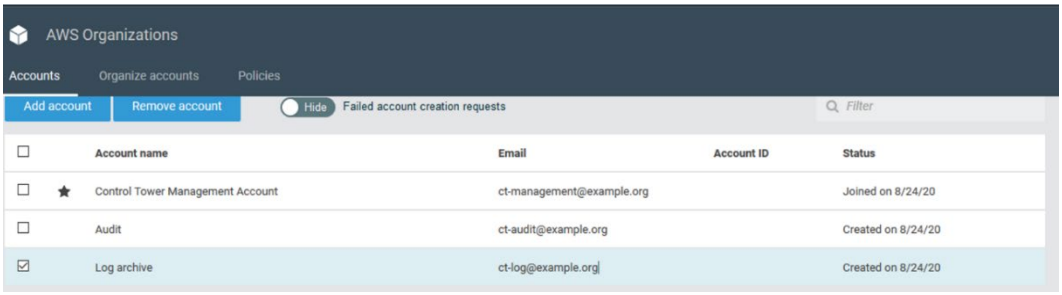


6. Remove the accounts from your organization.
7. For more information about closing an account, see https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/close-account.html#closing-member-account.
8. Remove the accounts from the organization prior to their deletion.
9. Set up a password for the accounts and a payment method to remove them from the organization so you can access them separately.
10. When you can access the accounts using the password you set up, remove them from your organization. There might be a waiting period until you can remove the accounts. See the next section about removing resources while the accounts remain in the organization so you don't incur additional costs.
11. Use your management account to go to AWS Organizations, and remove all member accounts.



12. Before you remove the audit and log archive accounts (and eventually remove the organization), review the next section. Remove all resources that were provisioned when setting up the landing zone from AWS Control Tower.

    Your AWS Organizations console should look similar to the example below:



AWS Control Tower managed resources cleanup walkthrough

In this section, you will decommission and delete all resources created by setting up the landing zone from AWS Control Tower.

Some of the shared accounts, including Martha's account, have a waiting period of around 7 days before they can be fully removed. You will receive a message when you try to remove them from the organization. By removing all the resources, you will ensure that there no additional costs are incurred during the waiting period.

A detailed set of instructions on how to clean up the resources used by AWS Control Tower is available. To decommission the resources created by AWS Control Tower, see https://docs.aws.amazon.com/controltower/latest/userguide/walkthrough-delete.html.

Make sure you follow those instructions, so no outstanding items remain active. Here is a summary of the items that need to be deleted:

- Delete SCPs
- Delete StackSets and Stacks
- Delete Amazon S3 Buckets in the Log Archive Account
- Clean Up Account Factory
- Clean Up Roles and Polices

Delete and close the member accounts

All resources in all member accounts should be deleted, including the decommissioning of all resources that were automatically created when setting up the landing zone using AWS Control Tower.

Here are the remaining steps required to delete the accounts used in this training:

1. Ensure that you can log in to all four of your accounts in AWS separately. You need a password and payment method for the accounts to do so.
2. With all member accounts (including log archive and audit accounts), go to AWS Organizations, and leave the organization, so only the management account remains active in the organization. Remember that you have to wait a grace period for all accounts before you can leave.

3. To delete and close your shared accounts, see https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/close-account.html.
4. To close your AWS accounts, see https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/close-account.html#closing-the-account.

AWS account deletion additional considerations

All the accounts provisioned with Account Factory and those used to set up a landing zone from AWS Control Tower are actual AWS accounts. Even though you unmanaged the accounts, an unmanaged (terminated) account is not closed or deleted.

When an account is unmanaged, the AWS SSO user that you selected when you created the account in Account Factory still has administrative access to the account.

If you do not want the user to have administrative access, you must change this setting in AWS SSO. To do this, update the account in Account Factory, and change the AWS SSO user email address for the account. For information about updating and moving your Account Factory account, see https://docs.aws.amazon.com/controltower/latest/userguide/account-factory.html#updating-account-factory-accounts.

Remove all accounts except your management account from AWS Organizations. After only the management account remains in AWS Organizations, then you can delete the organization, and delete and close your management account.

Accounts created in Account Factory are also regular AWS accounts. To delete the accounts associated with the four email addresses that you used during the training, follow the instructions for closing an account at https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/close-account.html. See the AWS Billing and Cost Management User Guide for further details.

If you encounter any issues that you cannot resolve during this clean-up process, contact AWS Support.