

Security Governance at Scale

Student Lab Guide

Contents

Lab 0: AWS Control Tower Setup Instructions	3
Email accounts	3
AWS account	4
AWS Control Tower landing zone.....	4
AWS Cost Explorer.....	10
Lab 1: AWS Control Tower Basic Tasks	14
Task 1: Log in and access the AWS console and AWS Single Sign-On.....	14
Task 2: Create an organizational unit (OU) and enable a guardrail.....	17
Create an OU	17
Enable a strongly recommended guardrail	17
Explore guardrail types	19
Task 3: Enroll a new AWS account using Account Factory.....	23
Explore network baseline settings and modifications	23
Create an account using Account Factory	25
Lab 2: AWS Service Catalog Portfolio Management	27
Task 1: Share a portfolio with selective organizational units	27
Share a set of products across child accounts.....	27
Create launch constraint roles on all accounts in the organization	27
Deploy the stacks on the management account separately.....	30
Explore the new assets.....	32
Share a portfolio from management account with a specific OU	34
Create a portfolio on the management account (from the Getting Started Library)	35
Share the product portfolio	38
Add launch constraints to the portfolio.....	40
Task 2: Enable self-service in a child account.....	42
Lab 3: AWS Control Tower Customizations	46
Task 1: Set up the Customizations for Control Tower solution.....	46
Task 2: Deploy customizations.....	49
Prepare repository access with AWS Cloud9	49
Ensure that Git is installed and is a supported version	51

Configure an AWS CLI credential helper	52
Connect to the CodeCommit.....	52
Commit and deploy the customizations.....	56
Task 3: View deployed customizations	58
View preventive guardrails	58
View detective guardrails.....	59
View IAM role.....	62
Lab 4: Lab Decommission Instructions.....	63
Decommission and clean up Lab 3.....	63
Delete the AWS CloudFormation StackSets.....	63
Detach and delete the service control policies.....	67
Delete the customization solution stack from Lab 3	69
Delete AWS Cloud9 development environment	69
Decommission and clean up Lab 2.....	73
Clean up Amazon EC2 and Amazon S3 resources.....	73
Delete roles, users, and groups created by AWS CloudFormation	76
Decommission and clean up Lab 1 and Lab 0	79
Unmanage and delete AWS Control Tower accounts	79
AWS Control Tower managed resources cleanup walkthrough	81
Delete and close the member accounts	81
AWS account deletion additional considerations.....	82

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.

The screenshot shows the 'Create an AWS account' wizard. On the left, there's a promotional message: 'AWS Accounts Include 12 Months of Free Tier Access' and 'Including use of Amazon EC2, Amazon S3, and Amazon DynamoDB'. It also says 'Visit aws.amazon.com/free for full offer terms'. On the right, there are input fields for 'Email address' (containing a placeholder), 'Password' (redacted), 'Confirm password' (redacted), and 'AWS account name' (containing 'Control Tower Management Account'). A large yellow 'Continue' button is at the bottom, and a link 'Sign in to an existing AWS account' is just above it. At the very bottom, there's a small note about copyright and links to 'Privacy Policy' and 'Terms of Use'.

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>.

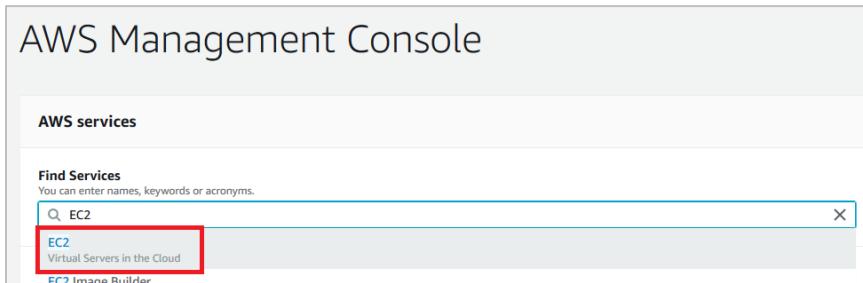
After you create your account and can log in, proceed to the next section to learn how to set up AWS Control Tower's landing zone.

AWS Control Tower landing zone

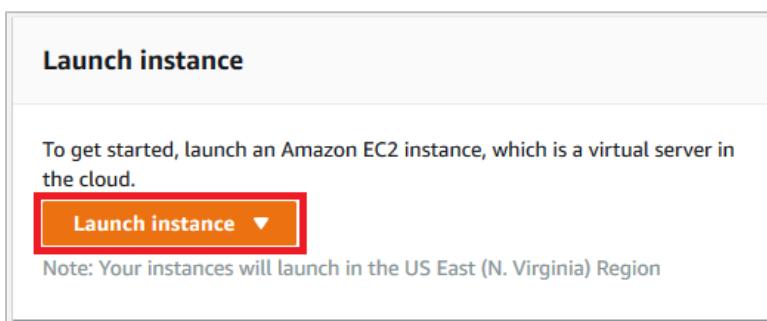
To set up AWS Control Tower, your account needs certain security scores. Depending on the domain you chose for your email addresses, those scores are lower than what is required for the AWS Control Tower setup.

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:

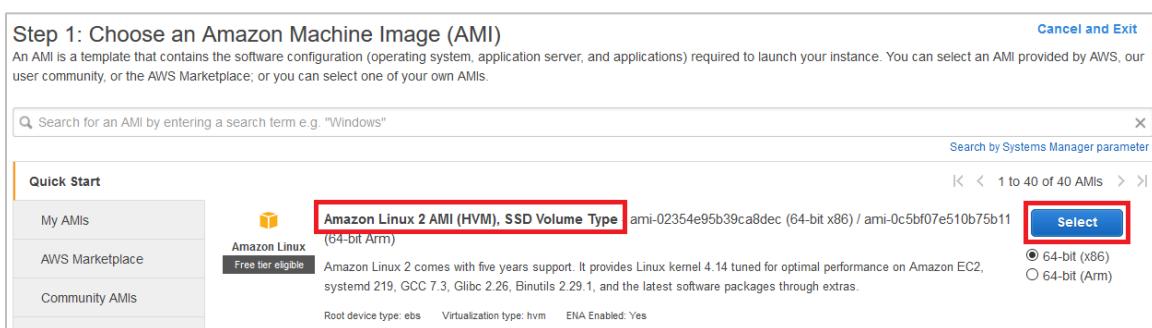
1. Open the AWS Management Console using your management account
2. Navigate to the Amazon EC2 service dashboard.



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.

Step 2: Choose an Instance Type

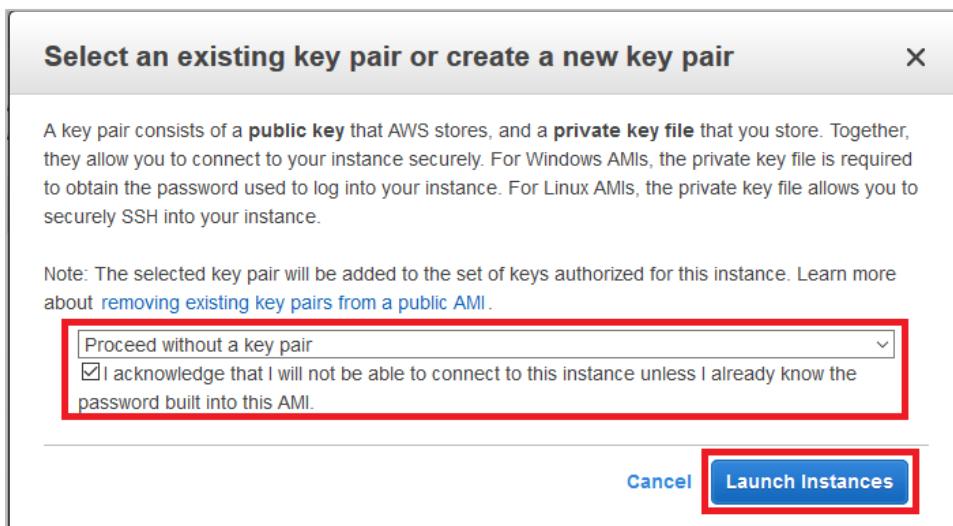
Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types ▾ Current generation ▾ Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs ⓘ	Memory (GiB) ⓘ	Instance Storage (GB) ⓘ	EBS-Optimized Available ⓘ	Network Performance ⓘ	IPv6 Support ⓘ
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes

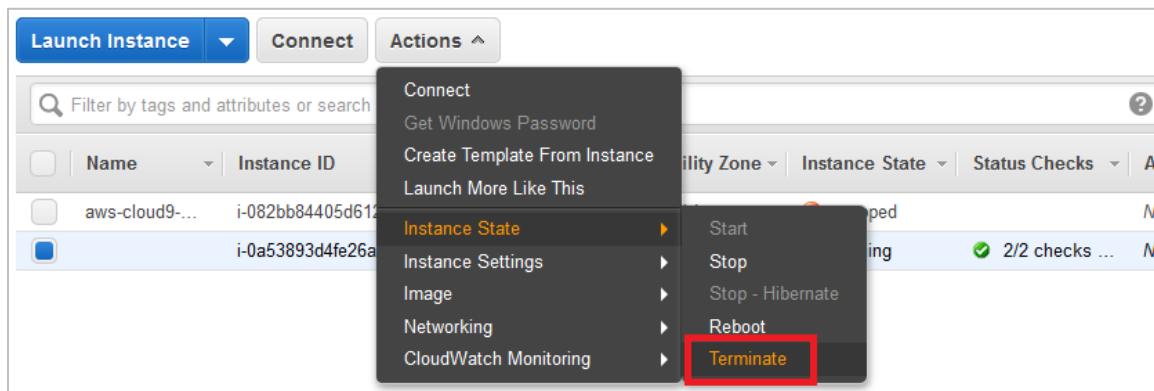
6. Choose **Review** and launch.
7. You need to decide your key pair. Select **Proceed without a key pair**, select the *I acknowledge that I will not be able to connect to this instance unless I already know the password built into this AMI* box, and then choose **Launch instances**.



- 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.

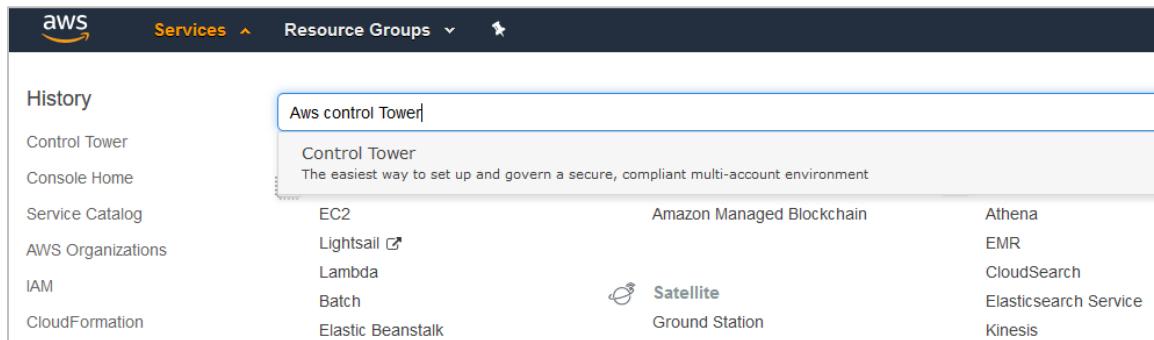
Filter by tags and attributes or search by keyword								
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IP)
<input type="checkbox"/>	aws-cloud9...	i-082bb84405d612210	t2.micro	us-east-1d	stopped		None	
<input type="checkbox"/>		i-0a53893d4fe26ad5b	t2.micro	us-east-1e	running	Initializing	None	ec2-54-236-28-1

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

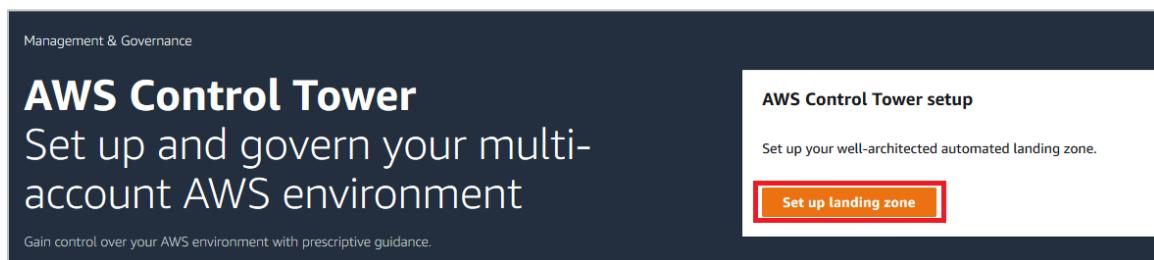


Next, you will set up an AWS Control Tower landing zone.

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:

AWS Control Tower > Set up landing zone

Set up landing zone Info

Pricing and home region [Learn more](#)

There is no additional charge for using AWS Control Tower. You only pay for AWS services enabled by AWS Control Tower.

Home Region: US East (Ohio)
 Choose a home region for your AWS Control Tower by selecting a region from the AWS Region selector. This is the default region where resources in your shared accounts will be provisioned, for example, your S3 bucket for your log archive.
 AWS Control Tower is not yet available in all regions.

Shared accounts
 As a best practice for a well-architected multi-account environment, AWS Control Tower will set up accounts that offer isolated environments for specialized roles in your organization.

Master account
 The master account uses your existing AWS account email and is used for billing and management of your accounts and landing zone.
 your-email+ct-management@your-email-provider.com

Log archive account
 The log archive account is a repository of immutable logs of API activities and resource configurations from all accounts.
 your-email+ct-log@your-email-provider.com
 The log archive account email must not be in use for an existing AWS account and should be from 6 to 64 characters long.

Audit account
 The audit account is a restricted account for your security and compliance teams to gain read and write access to all accounts.
 your-email+ct-audit@your-email-provider.com
 The audit account email must not be in use for an existing AWS account and should be from 6 to 64 characters long.

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*.

Service permissions
 AWS Control Tower needs your permission to administer AWS resources and enforce rules on your behalf.

▶ [Learn more about permissions](#)

▶ [Learn more about guidance](#)

I understand the permissions AWS Control Tower will use to administer AWS resources and enforce rules on my behalf. I also understand the guidance on the use of AWS Control Tower and the underlying AWS resources.

[Cancel](#) [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>

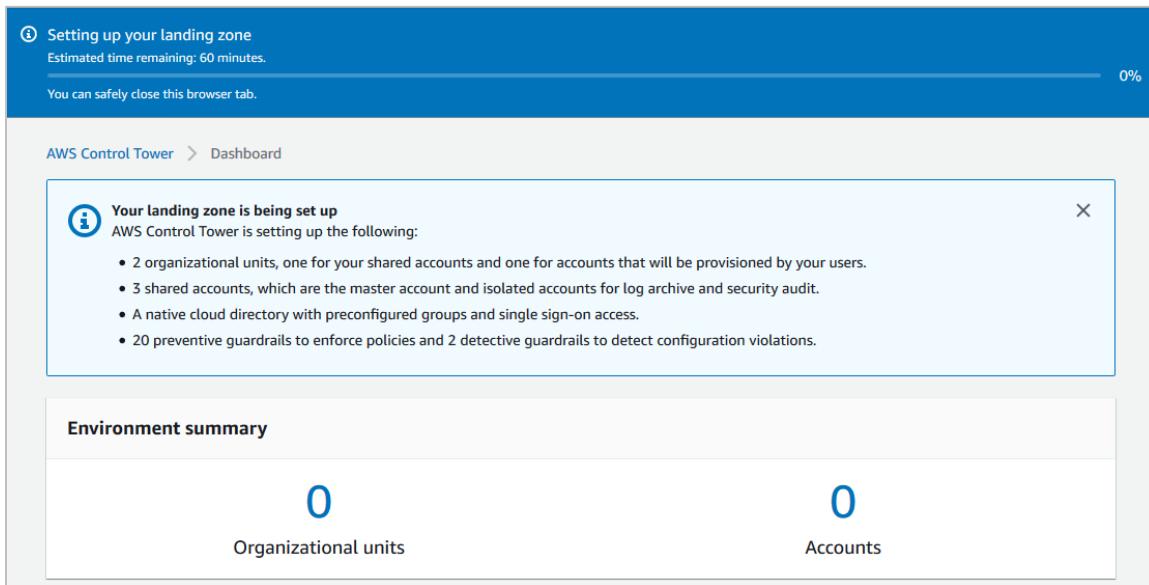
Your Username:
your-email+ct-management@your-email-provider.com

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

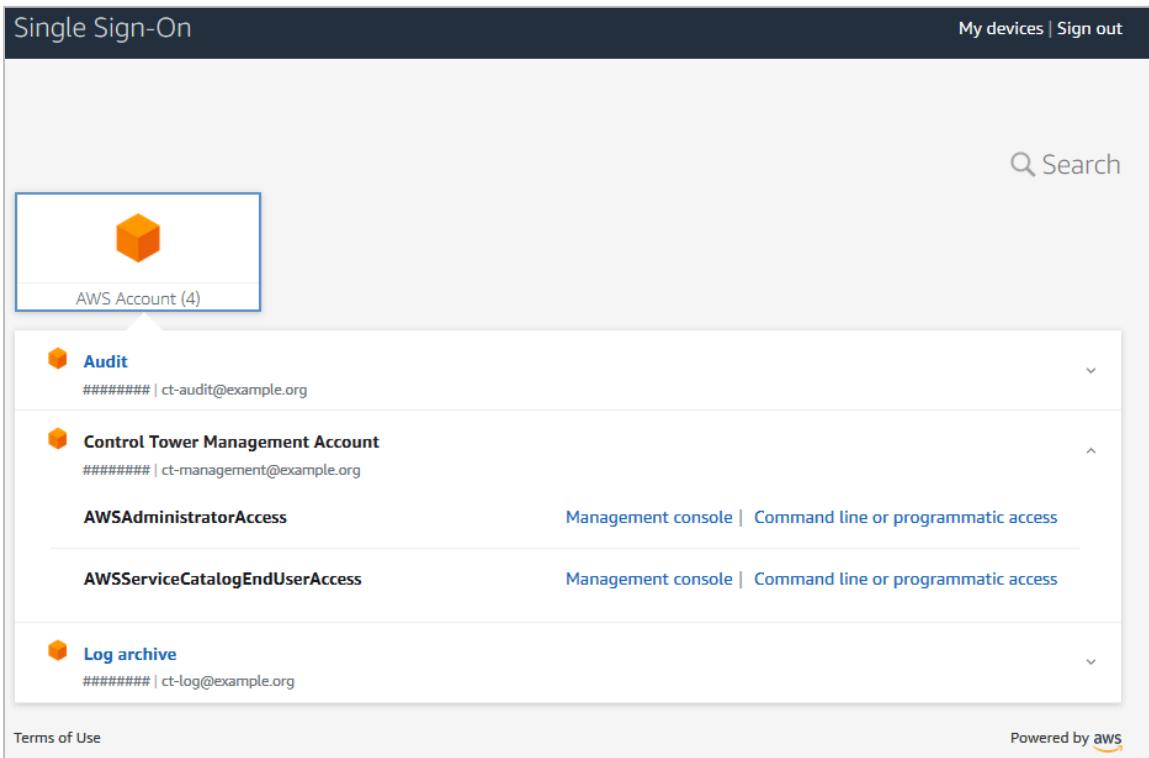
We recommend that you follow the instructions in the next section to enable AWS Cost Explorer.

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.

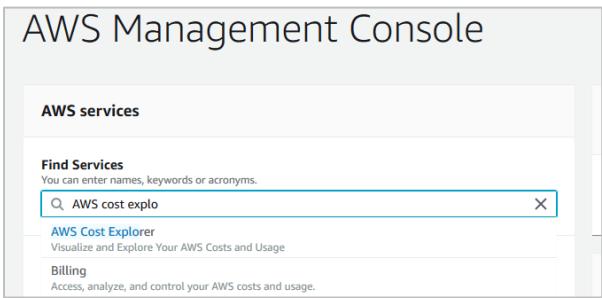


The screenshot shows the AWS Single Sign-On interface. At the top, there's a navigation bar with "Single Sign-On" on the left and "My devices | Sign out" on the right. A search bar is located in the top right corner. Below the navigation bar, there's a section titled "AWS Account (4)" with a blue cube icon. The main content area lists four accounts with their respective icons, names, and access methods:

- Audit**: Audit icon, ###### | ct-audit@example.org, Management console | Command line or programmatic access
- Control Tower Management Account**: Control Tower icon, ###### | ct-management@example.org, Management console | Command line or programmatic access
- AWSAdministratorAccess**: Administrator icon, Management console | Command line or programmatic access
- AWSServiceCatalogEndUserAccess**: Service Catalog icon, Management console | Command line or programmatic access
- Log archive**: Log icon, ###### | ct-log@example.org, Management console | Command line or programmatic access

At the bottom of the page, there are links for "Terms of Use" and "Powered by aws".

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



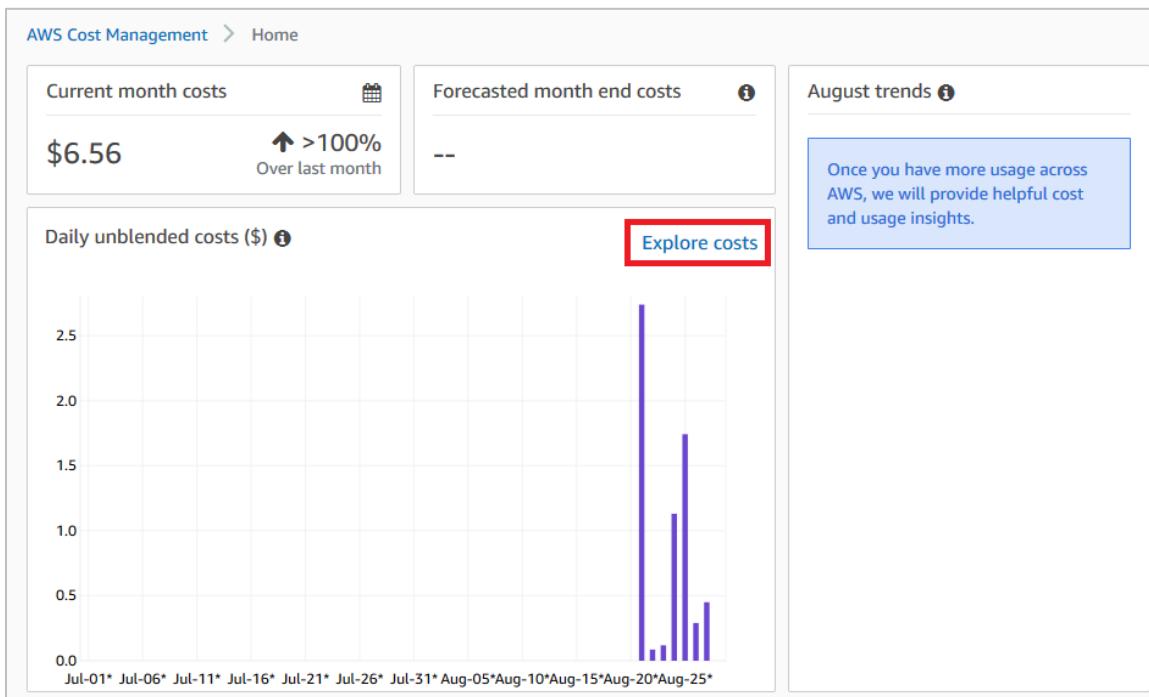
The screenshot shows the AWS Management Console search results for "AWS cost explo". The search bar contains "AWS cost explo". Below the search bar, the results show:

- AWS Cost Explorer**: Visualize and Explore Your AWS Costs and Usage
- Billing**: Access, analyze, and control your AWS costs and usage.

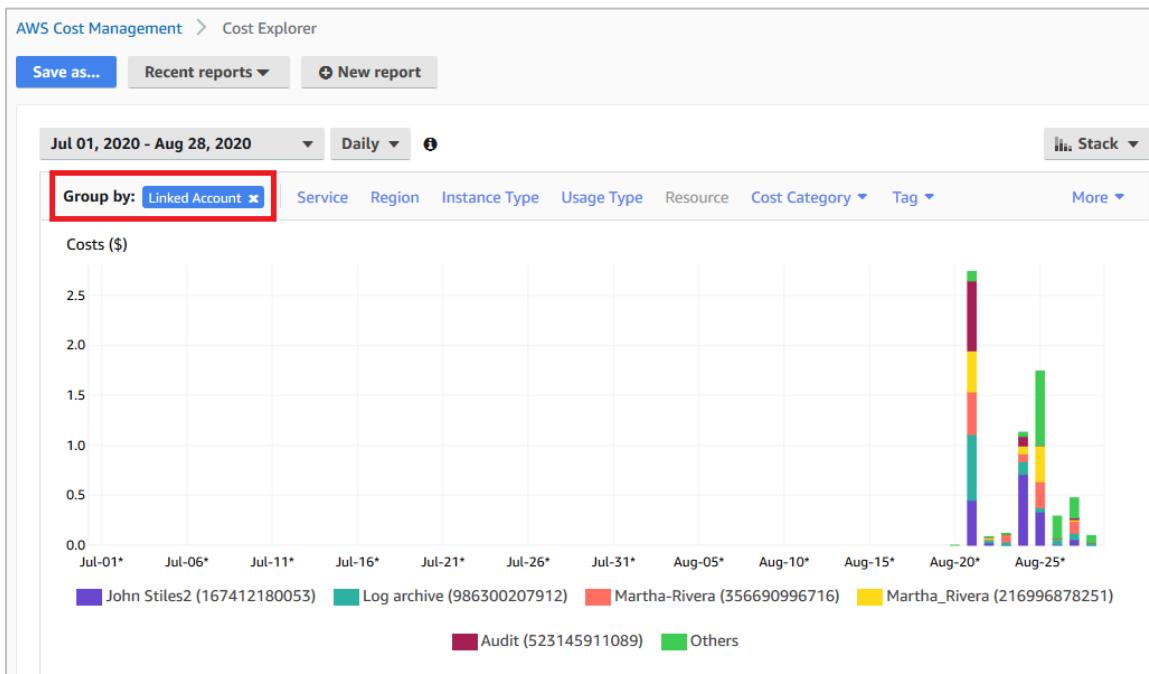
Note

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

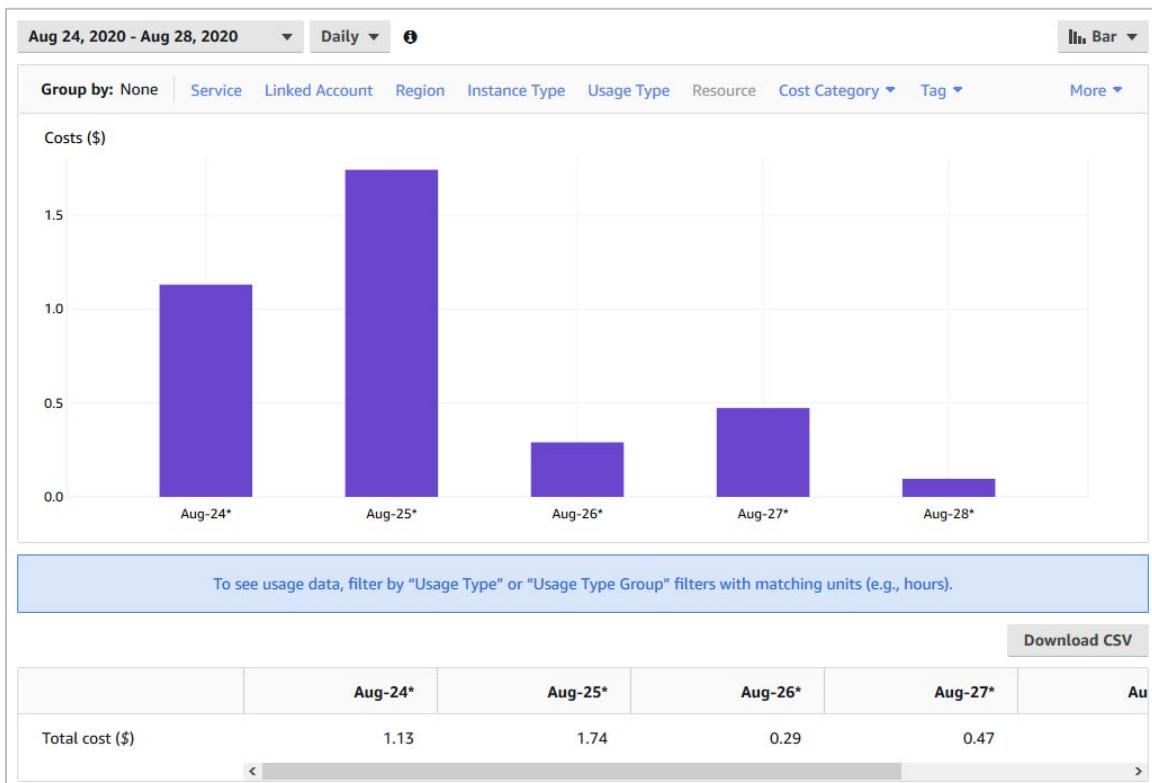
3. On the AWS Cost Explorer dashboard, find the **Daily unblended costs** section, and choose **Explore costs**.



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).

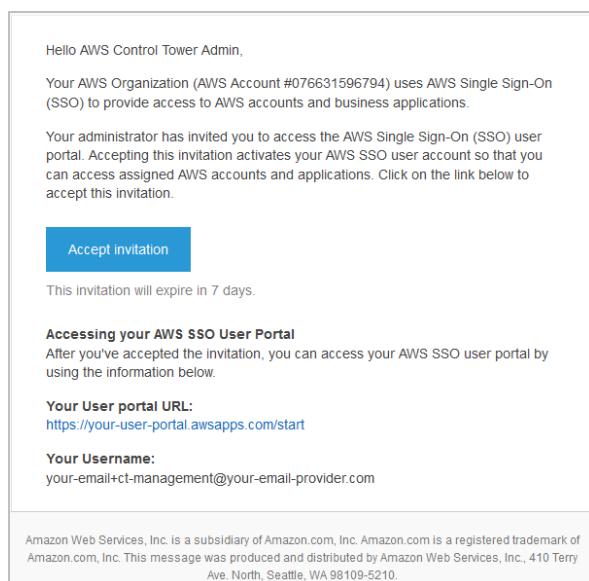
You need a solution that scales as the amount of data and number of processes grow exponentially. Your organization has decided to implement governance at scale as a framework. The goal is to adopt best practices while managing AWS resources, using them in an efficient, agile, secure, and compliant way. 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.

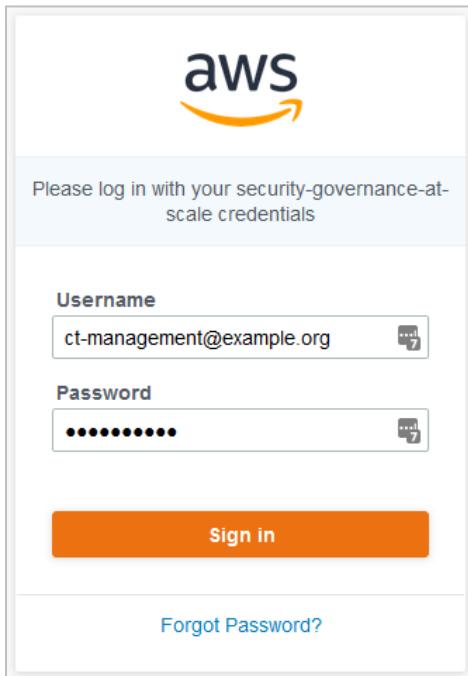


Note

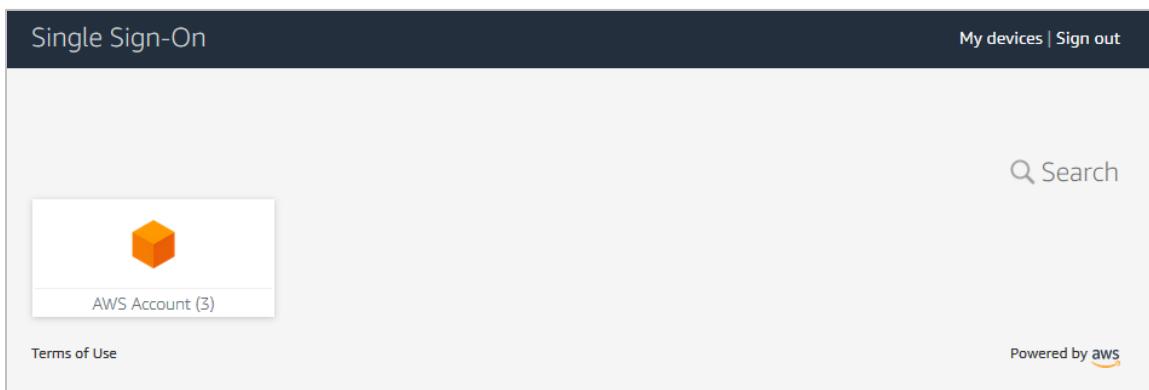
Write down (and potentially bookmark) your user portal's URL. You will need the URL to log in AWS SSO for this lab and others.

To sign in:

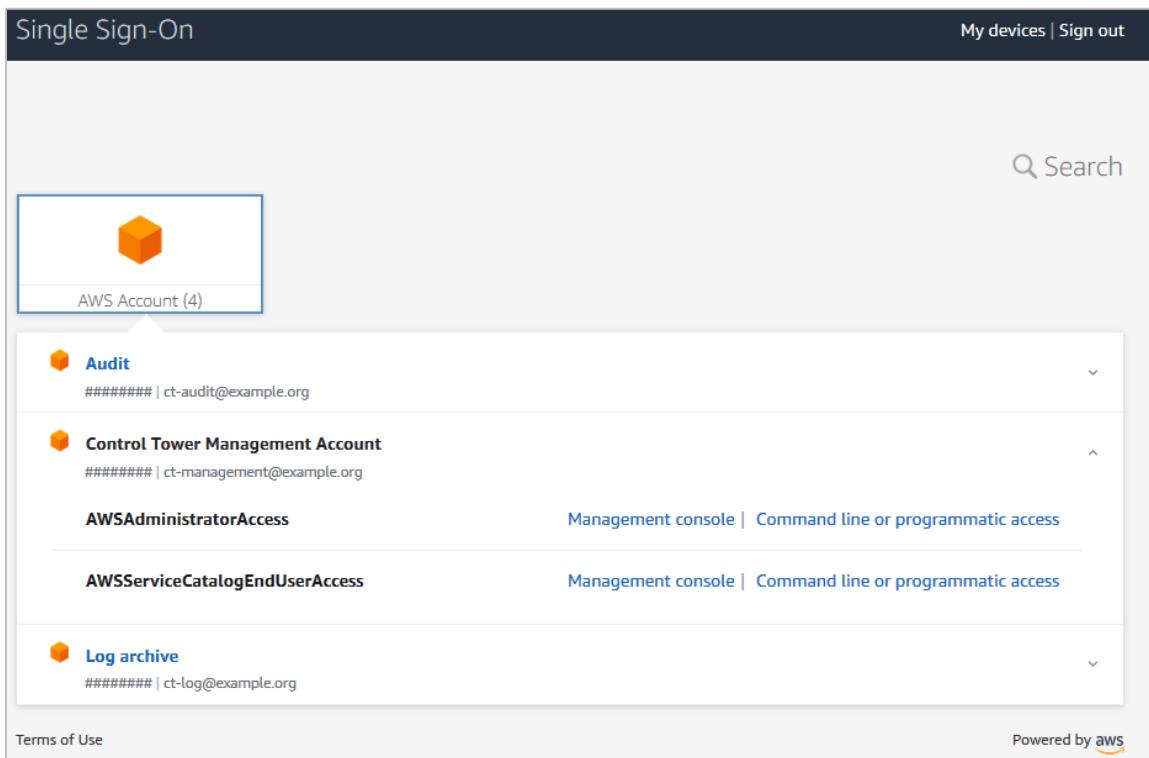
1. Accept the invitation.
2. Go to your user portal URL. The log-in screen should be similar to the following image:



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*.



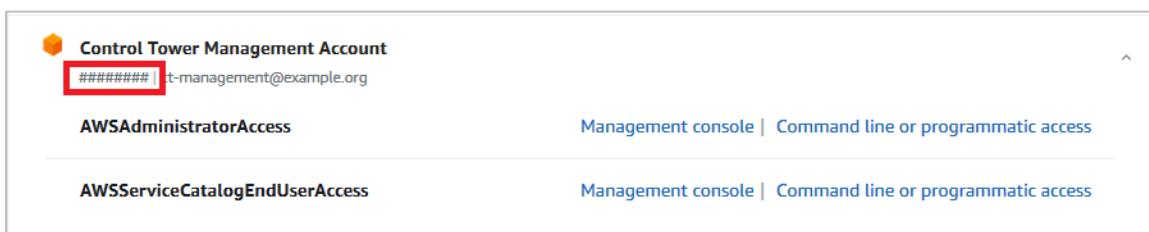
The screenshot shows the AWS Single Sign-On interface. At the top, there's a navigation bar with "Single Sign-On" on the left and "My devices | Sign out" on the right. A search bar is located in the top right corner. Below the navigation bar, there's a section titled "AWS Account (4)" with a blue cube icon. The account list includes:

- Audit**: [Manage](#) | [AWS IAM Identity Center](#)
- Control Tower Management Account**: [Manage](#) | [AWS IAM Identity Center](#)
- AWSAdministratorAccess**: [Management console](#) | [Command line or programmatic access](#)
- AWSServiceCatalogEndUserAccess**: [Management console](#) | [Command line or programmatic access](#)
- Log archive**: [Manage](#) | [AWS IAM Identity Center](#)

At the bottom of the page, there are links for "Terms of Use" and "Powered by aws".

Note

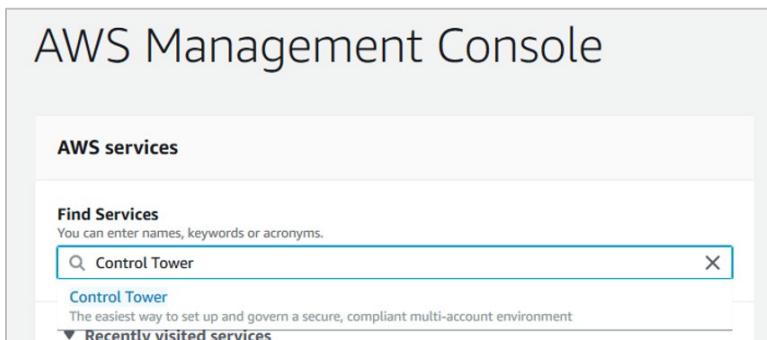
Write down the management account ID number. You will use it later in the lab.



The screenshot shows the AWS Single Sign-On interface. It displays the "Control Tower Management Account" details. The account ID "#####" and email "ct-management@example.org" are shown. The email address is highlighted with a red box. Below the account details, there are two access roles:

- AWSAdministratorAccess**: [Management console](#) | [Command line or programmatic access](#)
- AWSServiceCatalogEndUserAccess**: [Management console](#) | [Command line or programmatic access](#)

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



The screenshot shows the AWS Management Console search results for "Control Tower". The search bar at the top contains "Control Tower". Below the search bar, the results show:

- Control Tower**: The easiest way to set up and govern a secure, compliant multi-account environment
- Recently visited services**

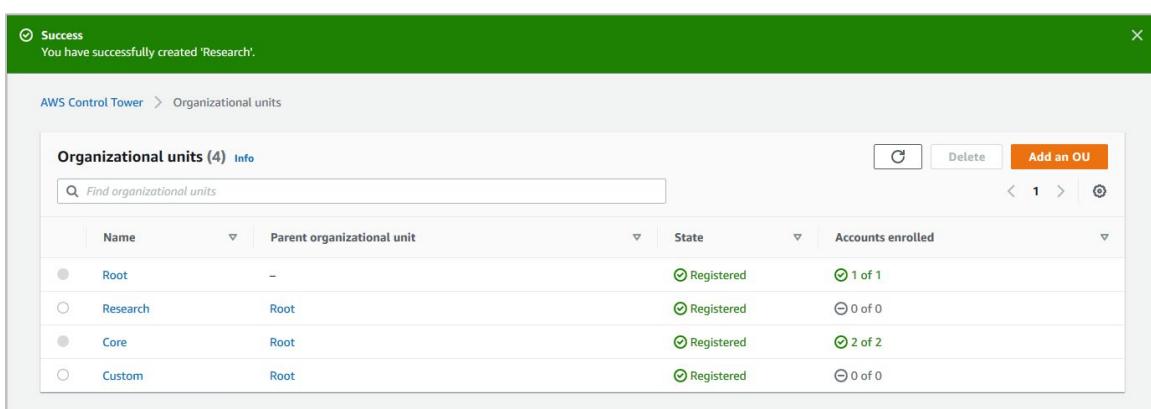
Task 2: Create an organizational unit (OU) and enable a guardrail

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.



The screenshot shows the AWS Control Tower interface for managing organizational units. At the top, a green success banner reads: "Success You have successfully created 'Research'." Below the banner, the navigation bar shows "AWS Control Tower > Organizational units". The main content area is titled "Organizational units (4) Info". A search bar with the placeholder "Find organizational units" is present. To the right are buttons for "Create" (disabled), "Delete", and "Add an OU". A table lists four organizational units:

Name	Parent organizational unit	State	Accounts enrolled
Root	-	Registered	1 of 1
Research	Root	Registered	0 of 0
Core	Root	Registered	2 of 2
Custom	Root	Registered	0 of 0

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**.

Guardrails Info

Guardrails are governance rules that you can enable on your organizational units (OUs) to enforce policies or detect violations.

Name	Guidance	Category	Behavior
Disallow internet connection through SSH	Strongly recommended	Network	Detection

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

Organizational units enabled

Name	Parent organizational unit	State	Update available
<small> ⓘ Guardrail not enabled</small> Guardrail is not enabled on any OUs.			

A new page lists the names of your OUs.

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

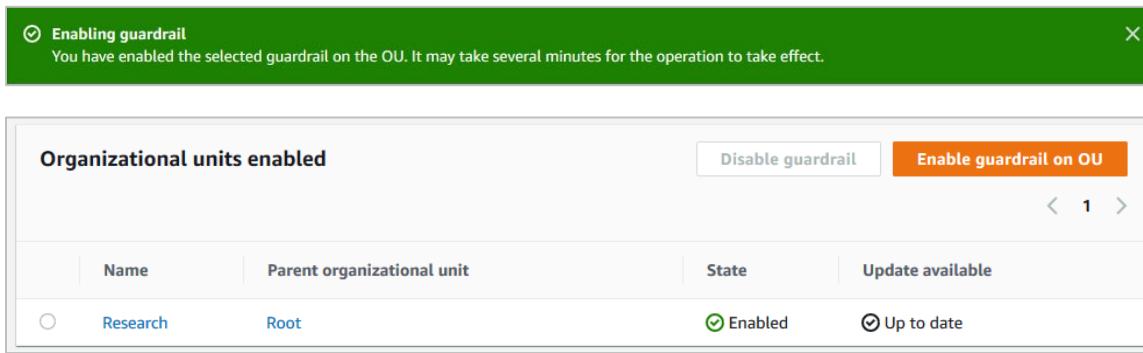
AWS Control Tower > Guardrails > Disallow internet connection through SSH > Enable guardrail on OU

Enable guardrail on OU

Choose an organizational unit (OU) to enable the following guardrail: Disallow internet connection through SSH.

Name	Parent organizational unit
<input checked="" type="radio"/> Research	Root
<input type="radio"/> Custom	Root
<input type="radio"/> Core	Root

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



Name	Parent organizational unit	State	Update available
Research	Root	Enabled	Up to date

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 preventative 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 preventative guardrail is either *enforced* or *not enabled*. Notice the enforced status in the following image.

Guardrail: Disallow deletion of log archive [Info](#)

Guardrail details		
Guardrail name Disallow deletion of log archive	Behavior Prevents policy violations by enforcement	Guidance Mandatory
Category Audit logs	Enabled summary Enabled on one or more organizational units (OUs)	Status Enforced
Description Prevent deletion of Amazon S3 buckets created by AWS Control Tower in the log archive account.		

Guardrail components	
< 1 >	
Name	Description
Service control policy (SCP)	Policies used to prevent account-level actions through AWS Organizations

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

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

Service control policy (SCP)

```

1  {
2    "Version": "2012-10-17",
3    "Statement": [
4      {
5        "Sid": "GRAUDITBUCKETDELETIONPROHIBITED",
6        "Effect": "Deny",
7        "Action": [
8          "s3>DeleteBucket"
9        ],
10       "Resource": [
11         "arn:aws:s3:::aws-controltower*"
12       ],
13       "Condition": {
14         "ArnNotLike": {
15           "aws:PrincipalARN": "arn:aws:iam::*:role/AWSControlTowerExecution"
16         }
17       }
18     }
19   ]
20 }
```

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

Organizational units enabled		
Name	Parent organizational unit	State
Core	Root	Enabled

Accounts				
Account name	Organizational unit	Owner	Compliance status	State
Audit	Core	AWS Control Tower	Compliant	Enrolled
Log archive	Core	AWS Control Tower	Compliant	Enrolled

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.

Guardrail: Disallow launch of EC2 instance types that are not EBS-optimized <small>Info</small>		
Guardrail details		
Guardrail name Disallow launch of EC2 instance types that are not EBS-optimized	Behavior Detects policy violations and alerts you in the dashboard	Guidance Strongly recommended
Category Operations	Enabled summary Not yet enabled on any organizational units (OUs)	Status -
Description Launch Amazon EC2 instances only with an Amazon EBS volume that is performance optimized. EBS-optimized volumes minimize contention between Amazon EBS I/O and other traffic from your instance.		
Guardrail components		
Name	Description	
AWS Config rule	Predefined AWS Config rules to evaluate configuration settings of resources	

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

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

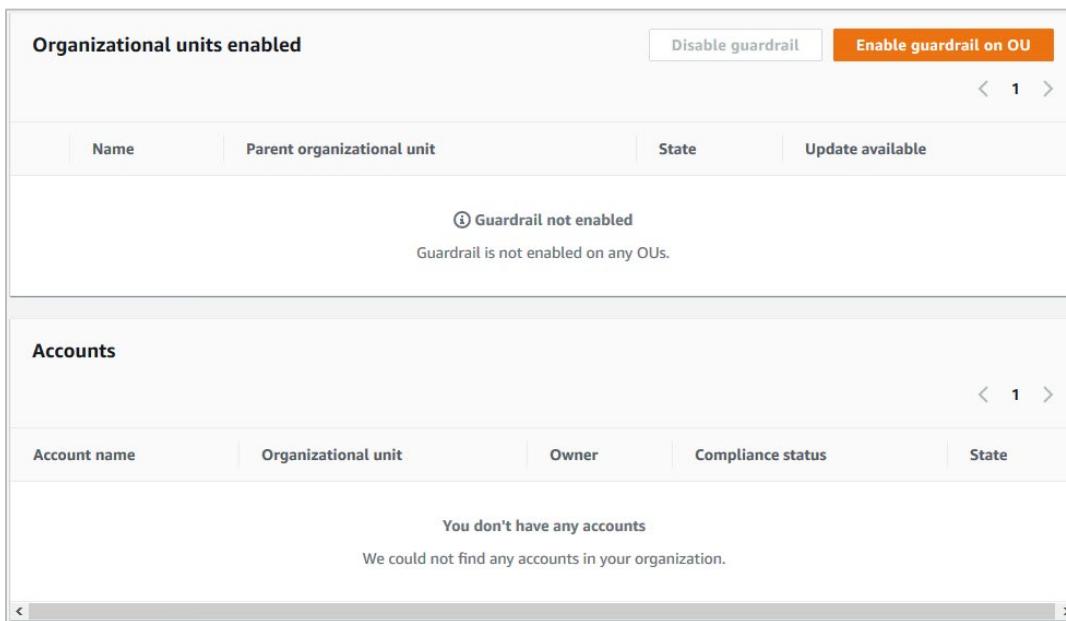


```

1 AWSTemplateFormatVersion: 2010-09-09
2 Description: Configure AWS Config rules to check whether EBS optimization is enabled for
3
4 Parameters:
5   ConfigRuleName:
6     Type: 'String'
7     Description: 'Name for the Config rule'
8
9 Resources:
10  CheckForEbsOptimizedInstance:
11    Type: AWS::Config::ConfigRule
12    Properties:
13      ConfigRuleName: !Sub ${ConfigRuleName}
14      Description: Checks whether EBS optimization is enabled for your EC2 instances tha
15      Source:
16        Owner: AWS
17        SourceIdentifier: EBS_OPTIMIZED_INSTANCE
18      Scope:
19        ComplianceResourceTypes:
20          - AWS::EC2::Instance
21

```

- Close the AWS Config rule window.
- Scroll to the Accounts section. The guardrail has not been enabled on any OUs.



Name	Parent organizational unit	State	Update available
<small> ⓘ Guardrail not enabled</small> Guardrail is not enabled on any OUs.			

Account name	Organizational unit	Owner	Compliance status	State
<small>You don't have any accounts</small> We could not find any accounts in your organization.				

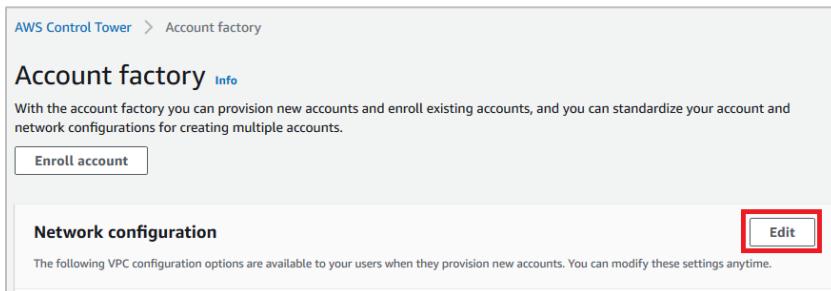
Task 3: Enroll a new AWS account using Account Factory

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**.

Edit account factory network configuration

VPC configuration options for new accounts

Internet-accessible subnet

Allow your users to create a public subnet in the VPC when provisioning a new account. If you edit the account factory configuration to enable public subnets when provisioning a new account, account factory configures Amazon VPC to create a [NAT Gateway](#). You will be billed for your usage by [Amazon VPC](#).

Maximum number of private subnets

Specify the maximum number of private subnets in the VPC.

2

Address range (CIDR) restriction for account VPCs

Range of addresses within which your account VPCs will be created.

10.0.0.0/24

Must be a valid 0.0.0.0/x format

Regions for VPC creation

Regions where VPCs are automatically created when an account is provisioned.

US East (N. Virginia)
 US East (Ohio)
 US West (Oregon)
 EU (Ireland)
 Asia Pacific (Sydney)

Availability Zones

Number of Availability Zones to configure subnets in each VPC.

3

Cancel Save

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

 **Success**
You have successfully updated your account factory settings.

AWS Control Tower > Account factory

Account factory Info

With the account factory you can provision new accounts and enroll existing accounts, and you can standardize your account and network configurations for creating multiple accounts.

Enroll account

Network configuration Edit

The following VPC configuration options are available to your users when they provision new accounts. You can modify these settings anytime.

Internet-accessible subnet Disallow	Address range (CIDR) for account VPCs 10.0.0.0/24	Regions for VPC creation US East (N. Virginia)
Maximum number of private subnets 2		
Availability Zone count 3		

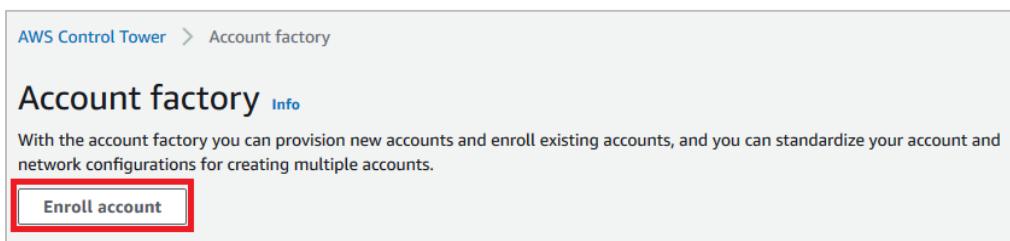
So far, you created an OU, enabled a strongly recommended guardrail, and modified the network baseline settings. Next, you will provision a new account in the OU with Account Factory.

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***.

Account details

Account enrollment provisions a new account or brings an existing account into AWS Control Tower governance.

Account email
 Specify a new email if you are creating a new account in your landing zone, or an existing email to extend governance to an existing AWS account.

Must be from 6 to 64 characters long.

Display name
 Name for account as it appears in AWS Control Tower

Must contain only letters, numbers, periods, dashes, underscores. Must begin with a letter or number. Do not use spaces.

AWS SSO email
 Designate an SSO user.

Must be from 6 to 64 characters long.

AWS SSO user name
 First and last name intended for creating an AWS SSO user

Organizational unit
 Defines governance for an account, and enables all guardrails on that OU

Cancel **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.

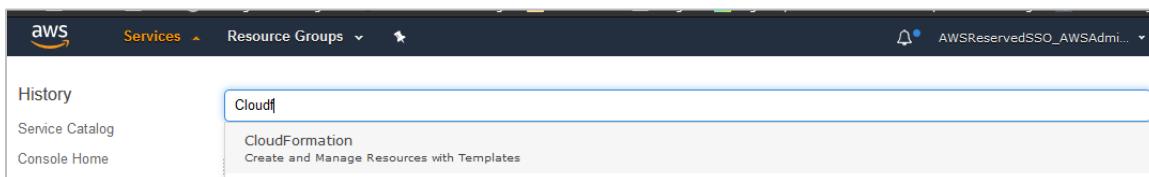
Share a set of products across child accounts

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:

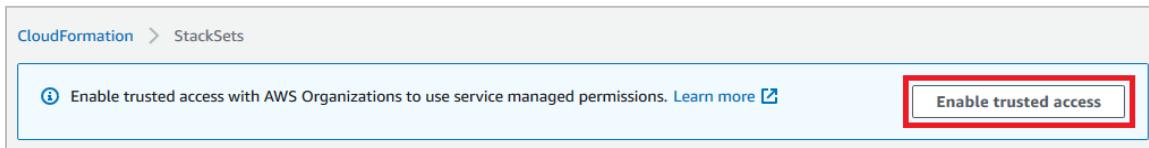
1. In your management account, navigate to the AWS CloudFormation service.



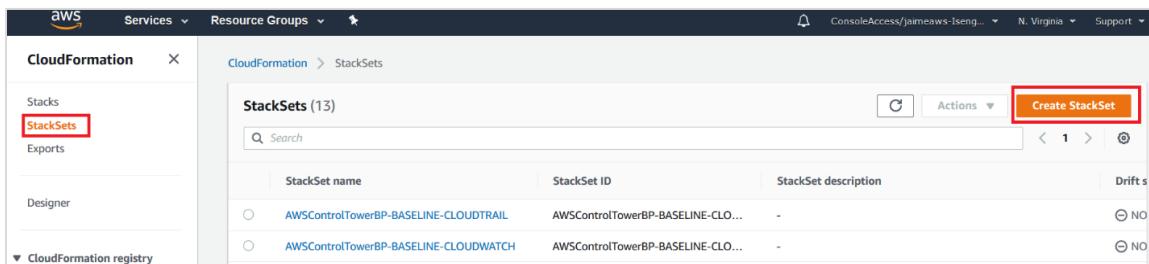
2. In the navigation pane, choose **StackSets**.

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-CONSTRAINT-ROLES**.
 - For StackSet description, enter **Create AWS Service Catalog Launch Constraint Roles across the organization**.

Specify StackSet details

StackSet name

StackSet name

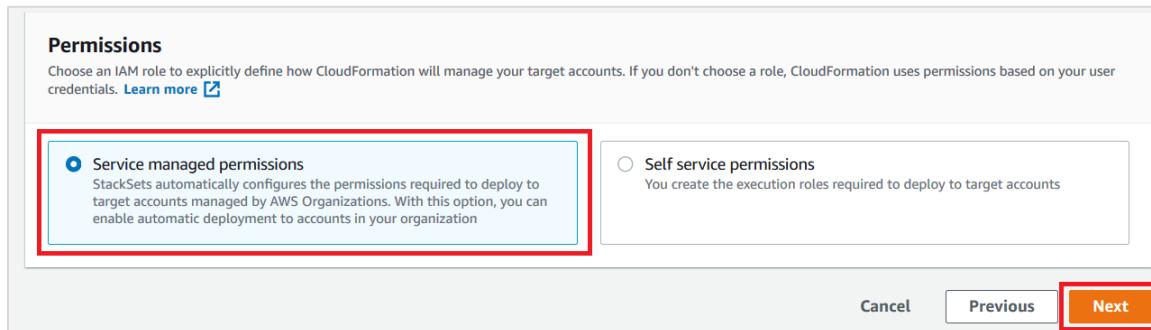
Must contain only letters, numbers, and dashes. Must start with a letter.

StackSet description
You can use the description to identify the stack set's purpose or other important information.

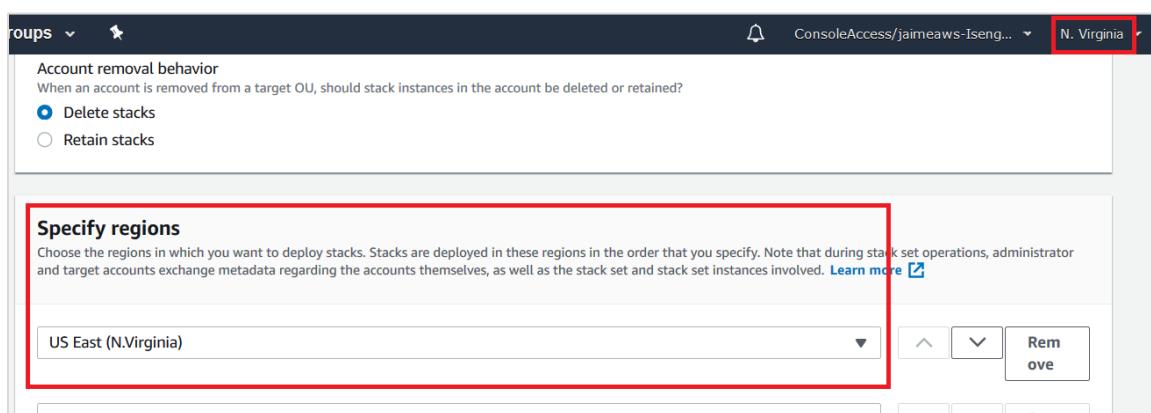
StackSet description

- 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.)

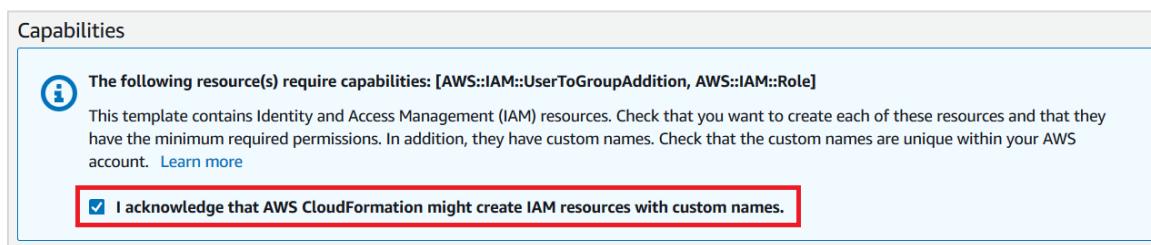
- o Note that the default password for SCEndUser is *Change@me*. You will need this later to log in and confirm that the constraints are in effect.
7. Choose **Next**.
 8. Select **Service managed permissions**, and choose **Next**.



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.

The screenshot shows the AWS CloudFormation console with the navigation path 'CloudFormation > StackSets'. A search bar at the top contains the text 'launch'. Below it, a table lists one StackSet entry:

StackSet name	StackSet ID	StackSet description	Drift status
SC-LAUNCH-CONSTRAINT-ROLES	SC-LAUNCH-CONSTRAINT-ROL...	Create AWS Service Catalog Launch Constraint Ro...	NOT_CHECKED

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**.

Specify template
 A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
 Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL Upload a template file

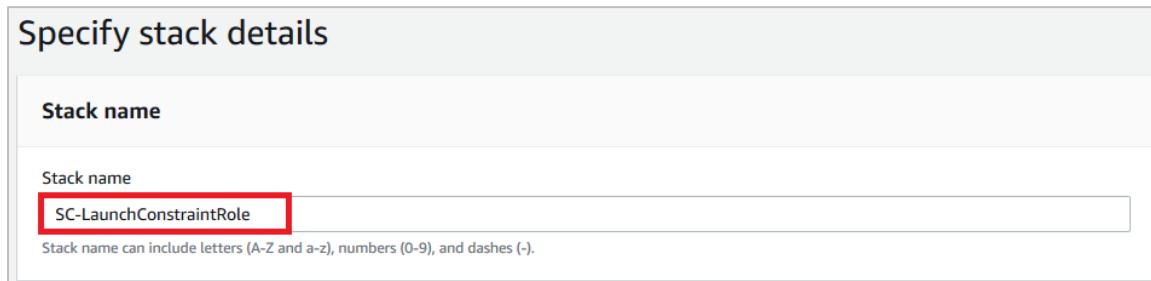
Amazon S3 URL

Amazon S3 template URL

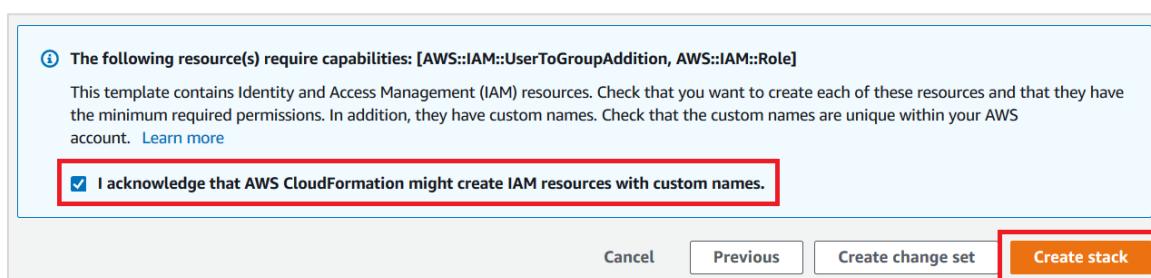
S3 URL: https://marketplace-sa-resources.s3.amazonaws.com/Create_scenduser_launchconstraint.yaml [View in Designer](#)

Cancel **Next**

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

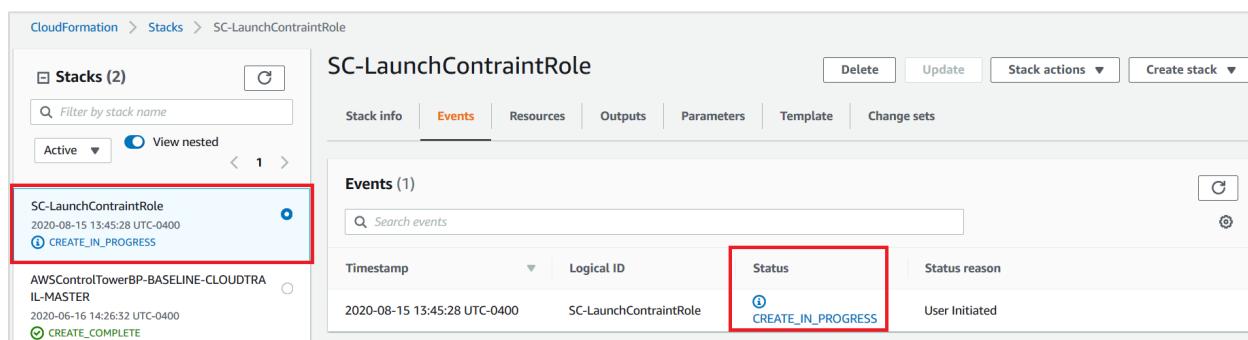


6. Accept the defaults on the Configure stack options page, and choose *Next*.
7. On the Review page, select *I acknowledge that AWS CloudFormation might create IAM resources with custom names*.
8. Choose *Create stack*.



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.

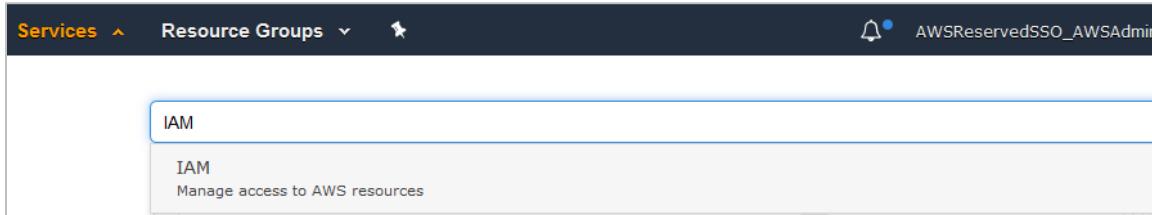


After the status changes to CREATE_COMPLETE, you can share the portfolio with a specific OU.

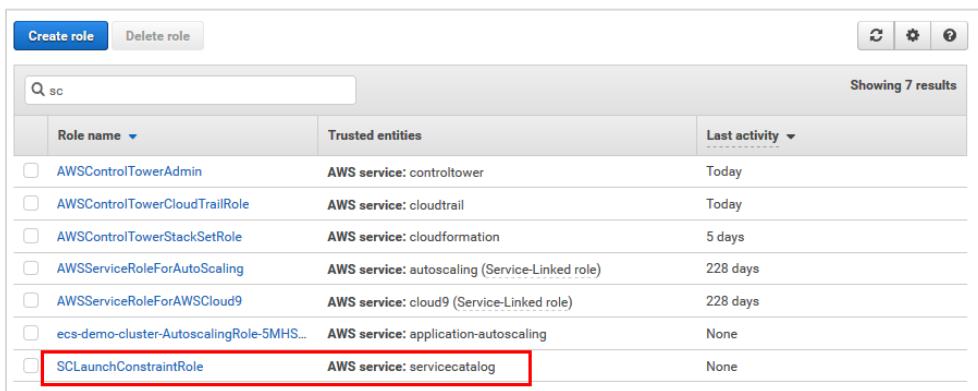
Explore the new assets

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

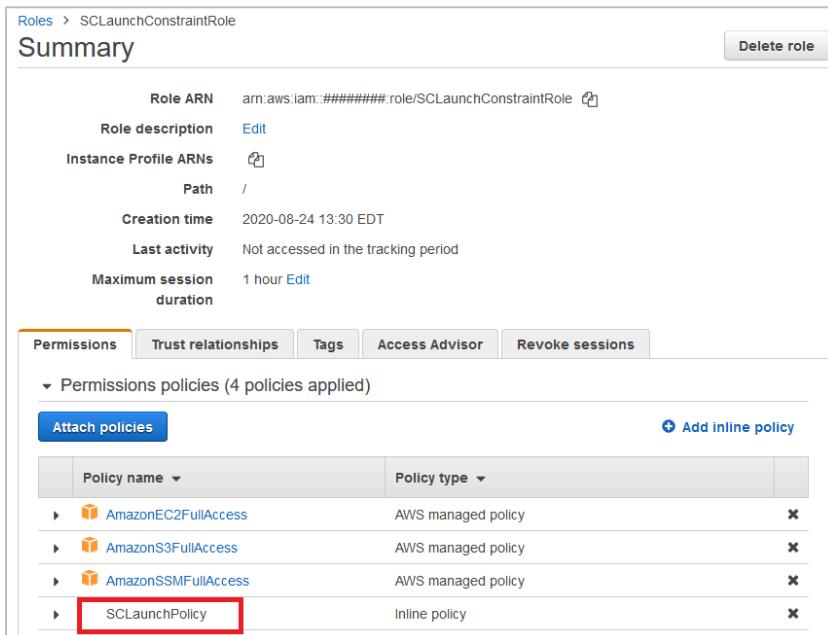
1. Navigate to the AWS Identity and Access Management (IAM) service.



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



Role name	Trusted entities	Last activity
AWSControlTowerAdmin	AWS service: controltower	Today
AWSControlTowerCloudTrailRole	AWS service: cloudtrail	Today
AWSControlTowerStackSetRole	AWS service: cloudformation	5 days
AWSServiceRoleForAutoScaling	AWS service: autoscaling (Service-Linked role)	228 days
AWSServiceRoleForAWSCloud9	AWS service: cloud9 (Service-Linked role)	228 days
ecs-demo-cluster-AutoscalingRole-5MHS...	AWS service: application-autoscaling	None
SCLaunchConstraintRole	AWS service: servicecatalog	None

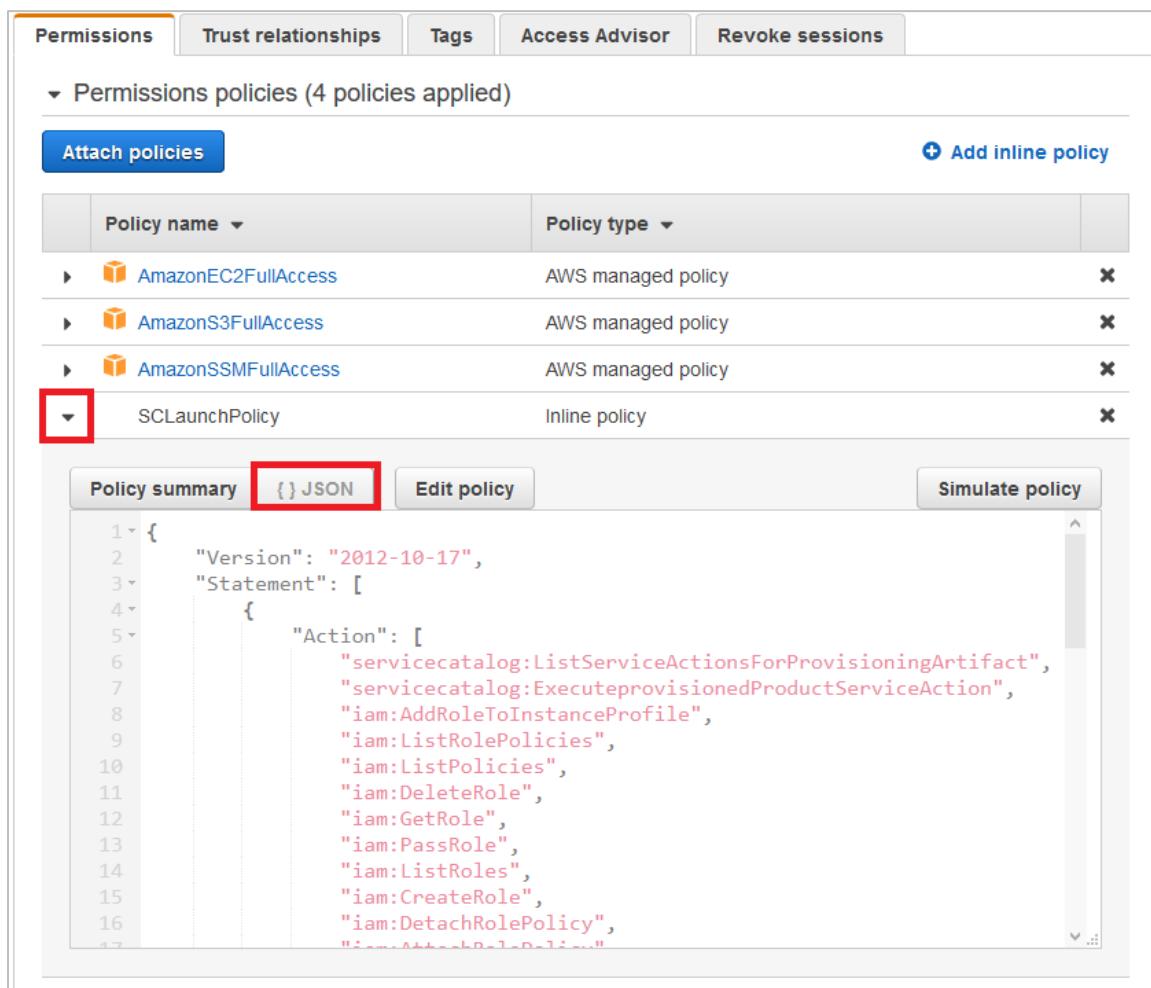


Role ARN	arn:aws:iam:#####.role/SCLaunchConstraintRole
Role description	Edit
Instance Profile ARNs	Edit
Path	/
Creation time	2020-08-24 13:30 EDT
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions [Attach policies](#) [Add inline policy](#)

Policy name	Policy type
AmazonEC2FullAccess	AWS managed policy
AmazonS3FullAccess	AWS managed policy
AmazonSSMFullAccess	AWS managed policy
SCLaunchPolicy	Inline policy

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



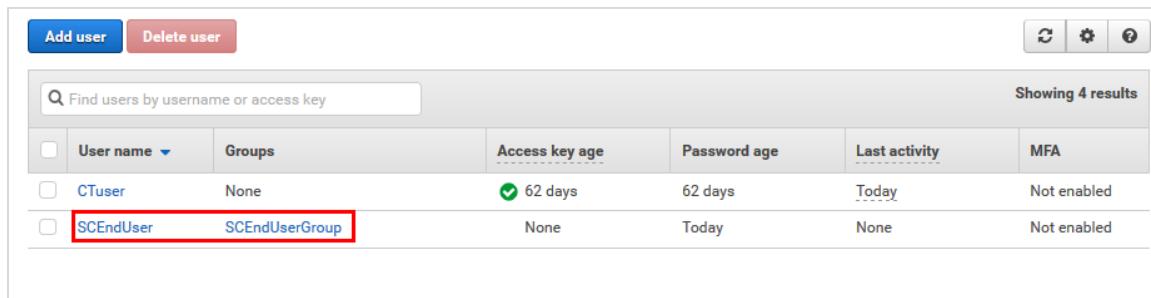
The screenshot shows the AWS IAM Policies page. At the top, there are tabs for Permissions, Trust relationships, Tags, Access Advisor, and Revoke sessions. Below these tabs, it says "Permissions policies (4 policies applied)". There is a blue "Attach policies" button and a blue "+ Add inline policy" button. A table lists four policies: AmazonEC2FullAccess, AmazonS3FullAccess, AmazonSSMFullAccess, and SCLaunchPolicy. The SCLaunchPolicy row has a red box around its "Policy name" column. Below the table is a panel for the SCLaunchPolicy. It has tabs for Policy summary, { } JSON (which is selected and highlighted with a red box), Edit policy, and Simulate policy. The JSON code in the { } JSON tab is as follows:

```

1  {
2      "Version": "2012-10-17",
3      "Statement": [
4          {
5              "Action": [
6                  "servicecatalog>ListServiceActionsForProvisioningArtifact",
7                  "servicecatalog>Execute provisioned ProductService Action",
8                  "iam>AddRoleToInstanceProfile",
9                  "iam>ListRolePolicies",
10                 "iam>ListPolicies",
11                 "iam>DeleteRole",
12                 "iam>GetRole",
13                 "iam>PassRole",
14                 "iam>ListRoles",
15                 "iam>CreateRole",
16                 "iam>DetachRolePolicy",
17                 "iam>AttachRolePolicy"
18             ]
19         }
20     ]
21 }

```

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



The screenshot shows the AWS IAM Users page. At the top, there are buttons for Add user and Delete user. There are also three small icons. Below these are search and filter fields. The main area shows a table of users. The table has columns: User name, Groups, Access key age, Password age, Last activity, and MFA. Two users are listed: CTUser and SCEndUser. The SCEndUser row has a red box around the "Groups" column, which contains "SCEndUserGroup".

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

Summary

User ARN	arn:aws:iam::932986852197:user/SCEndUser	
Path	/	
Creation time	2020-08-17 10:49 EDT	

Permissions **Groups (1)** **Tags** **Security credentials** **Access Advisor**

▼ Permissions policies (2 policies applied)

Add permissions **Add inline policy**

Policy name	Policy type
Attached from group	
▶ IAMUserChangePassword	AWS managed policy from group SCEndUserGroup
▶ AWSServiceCatalogEndUserFullAccess	AWS managed policy from group SCEndUserGroup

▶ Permissions boundary (not set)

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.

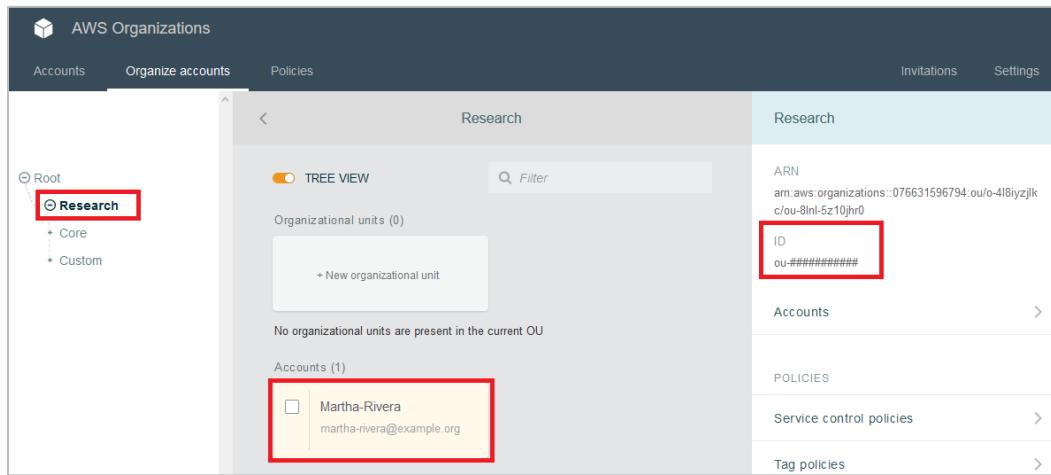
AWS services

Find Services
You can enter names, keywords or acronyms.

AWS Organizations
Central governance and management across AWS accounts.

CodeArtifact **AWS Organizations**

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.

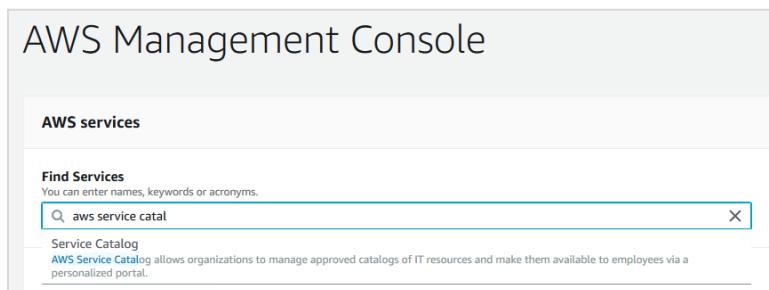


The screenshot shows the AWS Organizations console. In the left sidebar, under 'Root', the 'Research' button is highlighted with a red box. The main area displays 'Organizational units (0)' and 'Accounts (1)'. Under 'Accounts', there is one entry for 'Martha-Rivera' with the email 'martha-rivera@example.org', also highlighted with a red box. On the right, the 'Research' panel shows the ARN: 'arn:aws:organizations::076631596794:ou/o-4l8iyzjlk/c/ou-8lnl-5z10hr0'. Below it, the 'ID' field is highlighted with a red box, showing 'ou #####'. Other sections like 'Accounts', 'Policies', 'Service control policies', and 'Tag policies' are listed below.

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

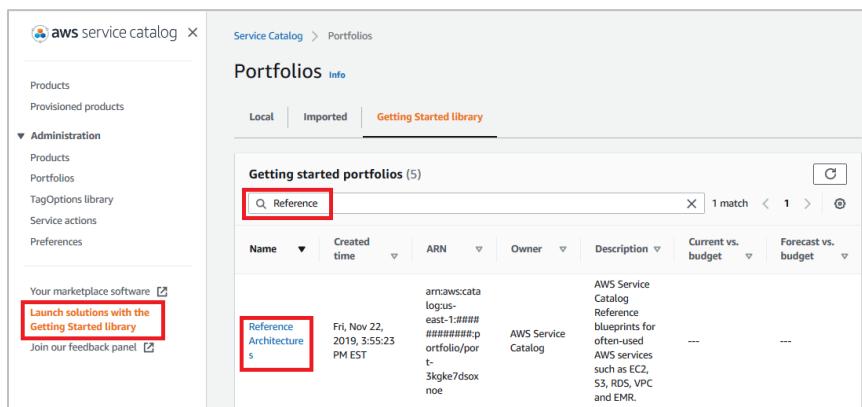
In this section, you will use existing products in the Getting Started Library to create the portfolio.

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



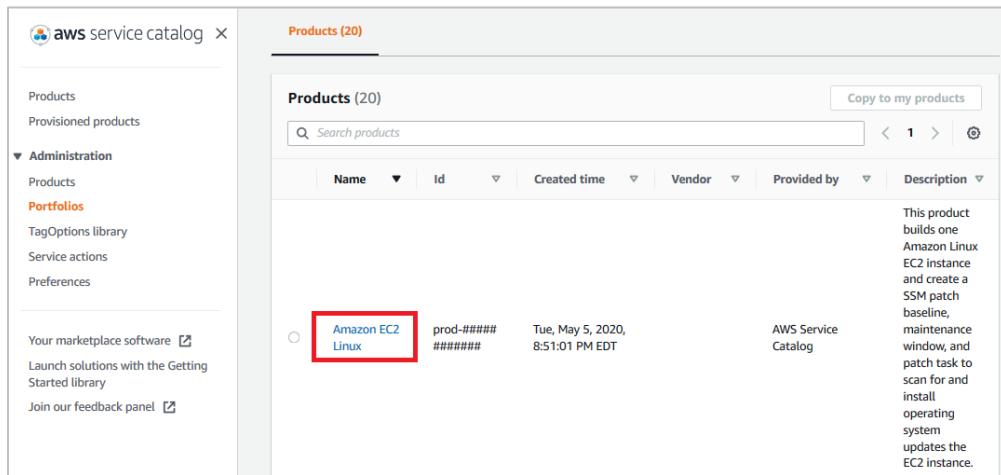
The screenshot shows the AWS Management Console with 'AWS services' selected. In the search bar, 'aws service catal' is typed, and a dropdown menu for 'Service Catalog' is shown. The description states: 'AWS Service Catalog allows organizations to manage approved catalogs of IT resources and make them available to employees via a personalized portal.'

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.



The screenshot shows the AWS Service Catalog 'Portfolios' page. The left sidebar has 'Administration' expanded, with 'Products' selected. A red box highlights the 'Launch solutions with the Getting Started Library' link. The main area shows 'Getting started portfolios (5)'. A search bar with 'Reference' is highlighted with a red box. One portfolio entry, 'Reference Architectures', is also highlighted with a red box. The table columns include Name, Created time, ARN, Owner, Description, Current vs. budget, and Forecast vs. budget. The 'Reference Architectures' row shows the ARN: 'arn:aws:cata-logus-east-1:####:portfolio/portfoliot-3kgke7dsoxnoe', Owner: 'AWS Service Catalog', and Description: 'AWS Service Catalog Reference blueprints for often-used AWS services such as EC2, S3, RDS, VPC and EMR.'

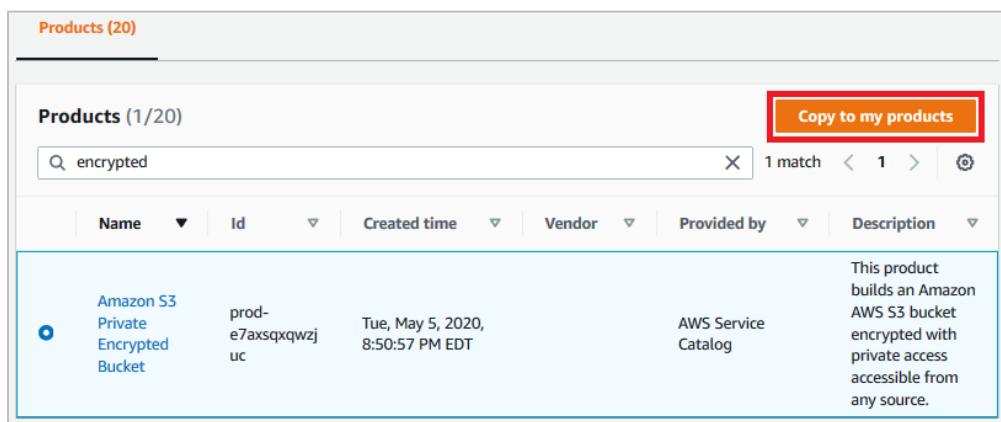
5. Select *Amazon EC2 Linux*, and choose *Copy to my products*.



The screenshot shows the AWS Service Catalog interface. In the left navigation pane, under 'Administration', 'Portfolios' is selected. The main area displays a list of products with 20 items. One product, 'Amazon EC2 Linux', is highlighted with a red box. To its right, there is a detailed description of the product.

Name	Id	Created time	Vendor	Provided by	Description
Amazon EC2 Linux	prod-#####	Tue, May 5, 2020, 8:51:01 PM EDT		AWS Service Catalog	This product builds one Amazon Linux EC2 instance and create a SSM patch baseline, maintenance window, and patch task to scan for and install operating system updates the EC2 instance.

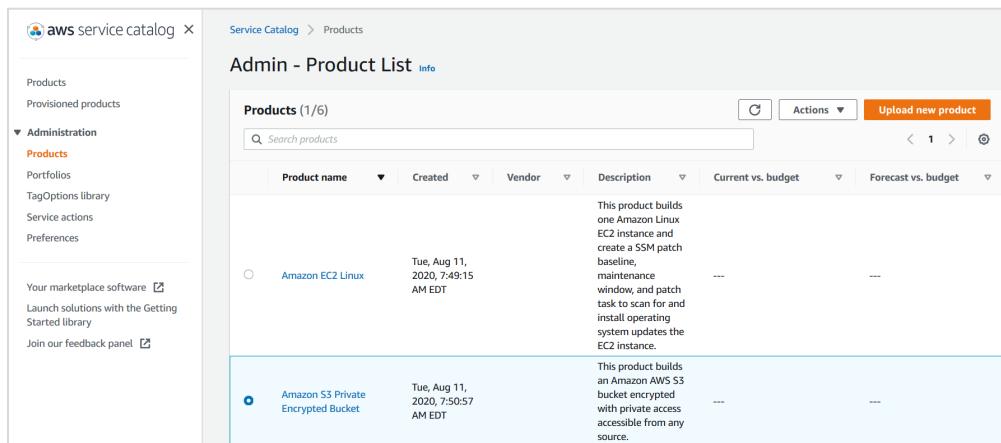
6. Select *Amazon S3 Private Encrypted Bucket*, and choose *Copy to my products*.



The screenshot shows the AWS Service Catalog interface. In the left navigation pane, under 'Administration', 'Products' is selected. The main area displays a list of products with 1/20 items. One product, 'Amazon S3 Private Encrypted Bucket', is highlighted with a blue box. To its right, there is a detailed description of the product.

Name	Id	Created time	Vendor	Provided by	Description
Amazon S3 Private Encrypted Bucket	prod-e7axsqxwzjuc	Tue, May 5, 2020, 8:50:57 PM EDT		AWS Service Catalog	This product builds an Amazon AWS S3 bucket encrypted with private access accessible from any source.

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



The screenshot shows the AWS Service Catalog interface. In the left navigation pane, under 'Administration', 'Products' is selected. The main area displays a list of products with 1/6 items. Two products are visible: 'Amazon EC2 Linux' and 'Amazon S3 Private Encrypted Bucket'. The 'Amazon S3 Private Encrypted Bucket' is highlighted with a blue box. To its right, there is a detailed description of the product.

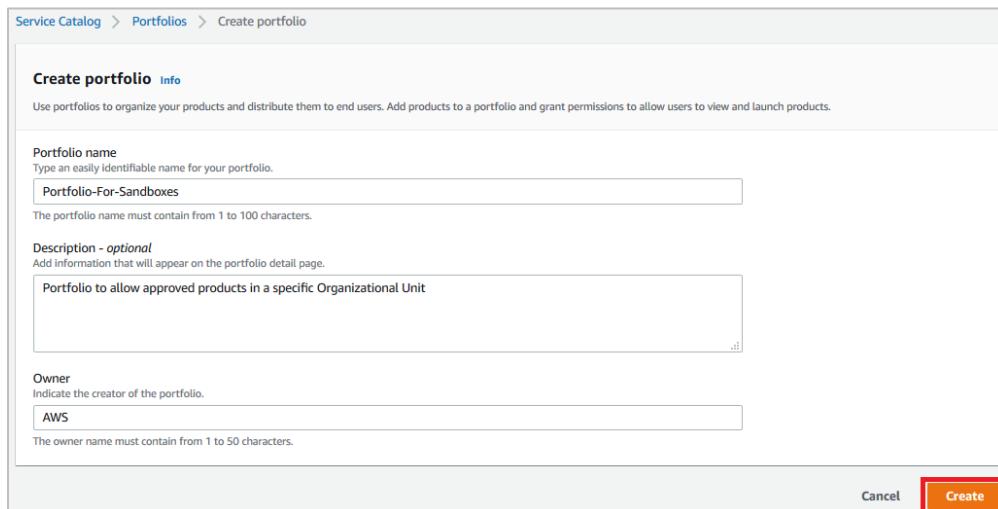
Product name	Created	Vendor	Description	Current vs. budget	Forecast vs. budget
Amazon EC2 Linux	Tue, Aug 11, 2020, 7:49:15 AM EDT		This product builds one Amazon Linux EC2 instance and create a SSM patch baseline, maintenance window, and patch task to scan for and install operating system updates the EC2 instance.	---	---
Amazon S3 Private Encrypted Bucket	Tue, Aug 11, 2020, 7:50:57 AM EDT		This product builds an Amazon AWS S3 bucket encrypted with private access accessible from any source.	---	---

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*:

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



Create portfolio Info

Use portfolios to organize your products and distribute them to end users. Add products to a portfolio and grant permissions to allow users to view and launch products.

Portfolio name
Type an easily identifiable name for your portfolio.

The portfolio name must contain from 1 to 100 characters.

Description - optional
Add information that will appear on the portfolio detail page.

Owner
Indicate the creator of the portfolio.

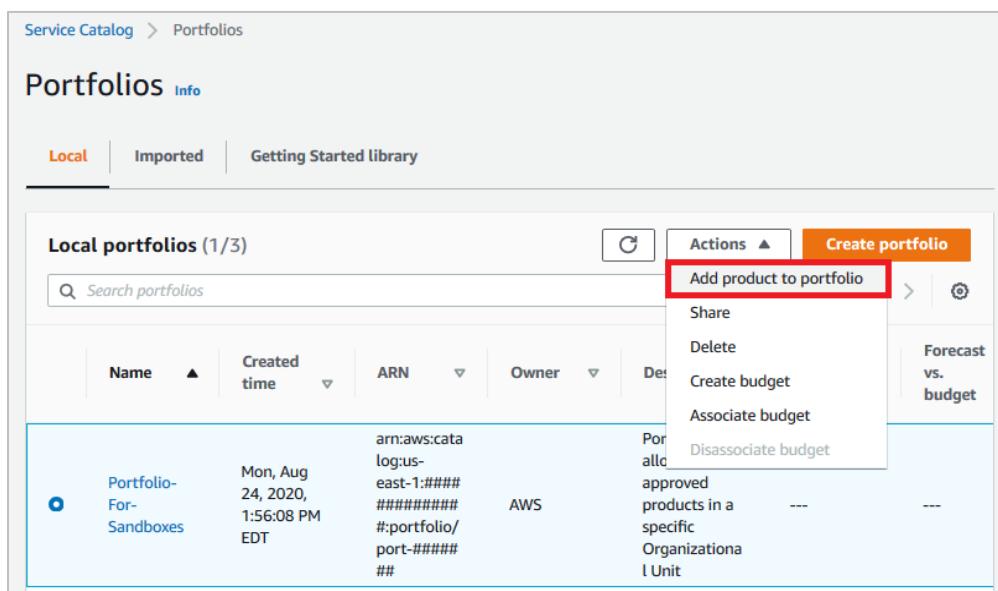
The owner name must contain from 1 to 50 characters.

Create

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

11. On the Portfolios page, choose **Portfolio-For-Sandboxes**.

12. For Actions, select *Add product to portfolio*.



Portfolios Info

Local | Imported | Getting Started library

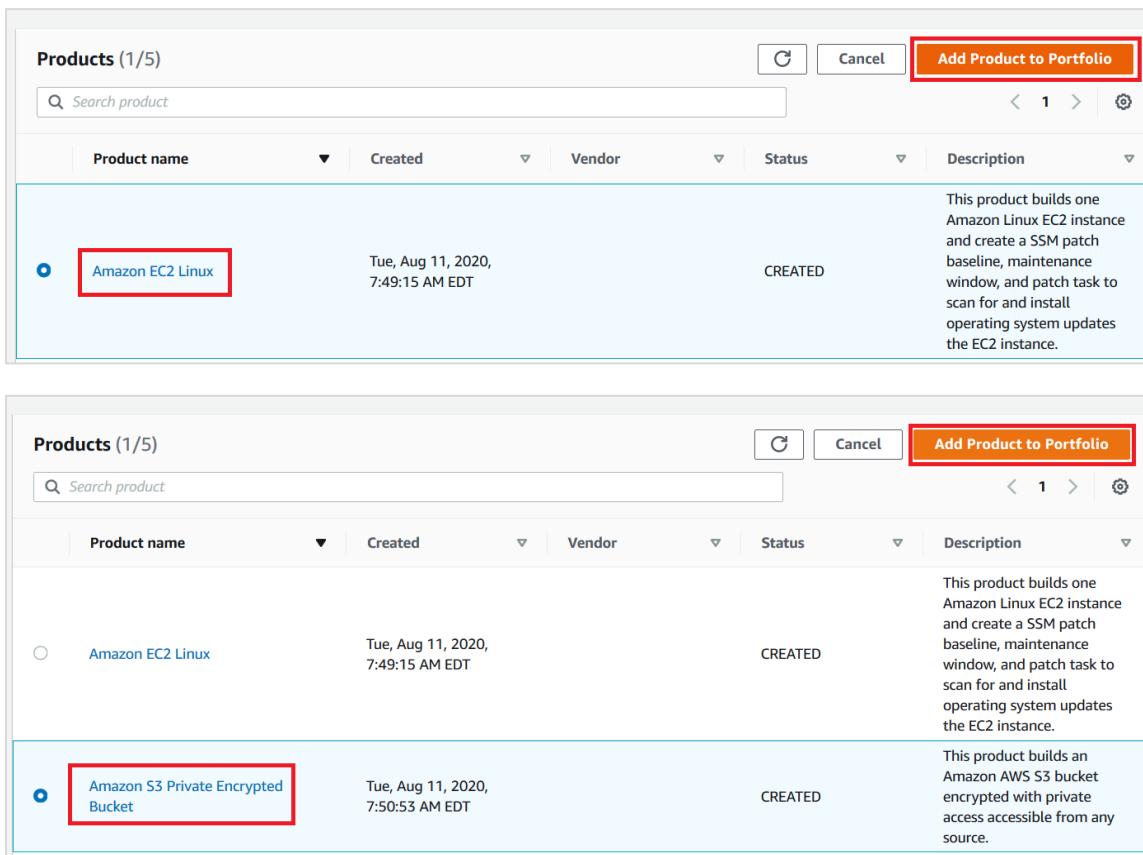
Local portfolios (1/3)

Name	Created time	ARN	Owner	Details
Portfolio-For-Sandboxes	Mon, Aug 24, 2020, 1:56:08 PM EDT	arn:aws:cata log:us-east-1:#### ##### #:portfolio/port-#### ##	AWS	Portfolio approved products in a specific Organization Unit

Actions ▾

- Create portfolio**
- Add product to portfolio**
- Share
- Delete
- Create budget
- Associate budget
- Disassociate budget
- Forecast vs. budget

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**.

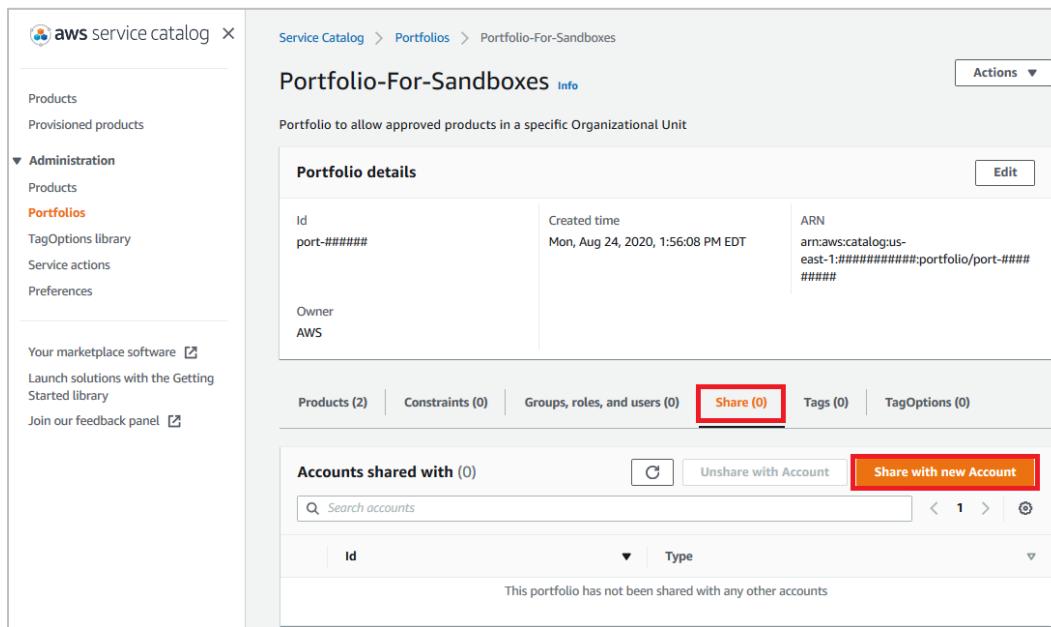


Products (1/5)					
<input type="button" value="Cancel"/> <input style="border: 2px solid red; background-color: #ff9900; color: white; padding: 2px 10px; margin-right: 10px;" type="button" value="Add Product to Portfolio"/> <input type="button" value="Search product"/> < 1 > 					
Product name	Created	Vendor	Status	Description	
<input checked="" type="radio"/> Amazon EC2 Linux	Tue, Aug 11, 2020, 7:49:15 AM EDT		CREATED	This product builds one Amazon Linux EC2 instance and create a SSM patch baseline, maintenance window, and patch task to scan for and install operating system updates the EC2 instance.	
<input checked="" type="radio"/> Amazon S3 Private Encrypted Bucket	Tue, Aug 11, 2020, 7:50:53 AM EDT		CREATED	This product builds an Amazon AWS S3 bucket encrypted with private access accessible from any source.	

Share the product portfolio

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

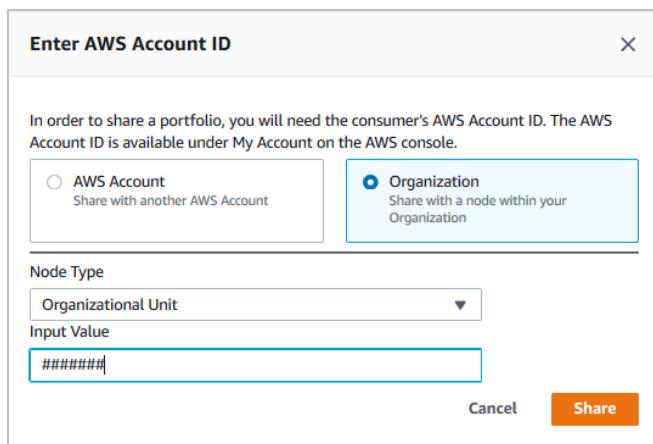
1. On the *Portfolios* page, choose *Portfolio-For-Sandboxes*.
2. Select *Share*, and then choose *Share with new Account*.



The screenshot shows the AWS Service Catalog interface. On the left, there's a sidebar with links like Products, Provisioned products, Administration (Products, Portfolios, TagOptions library, Service actions, Preferences), Your marketplace software, Launch solutions with the Getting Started library, and Join our feedback panel. The main content area shows the 'Portfolio-For-Sandboxes' details. It includes fields for Id (port-#####), Created time (Mon, Aug 24, 2020, 1:56:08 PM EDT), Owner (AWS), and ARN (arn:aws:catalog:us-east-1:#####:portfolio/port-#####). Below this, there are tabs for Products (2), Constraints (0), Groups, roles, and users (0), Tags (0), and TagOptions (0). The 'Share (0)' button is highlighted with a red box. Further down, there's a section for 'Accounts shared with (0)' with a search bar and a 'Share with new Account' button.

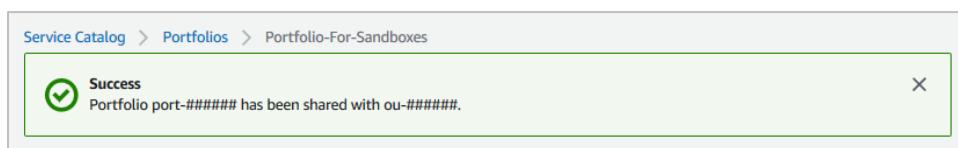
Now, you can share the portfolio with the Research OU.

3. In the Enter AWS Account ID dialogue box, do the following:
 - o Select *Organization*.
 - o For Node Type, select *Organizational Unit*.
 - o For Input Value, enter the organizational unit ID that you recorded at the beginning of this task.



The dialog box has a title 'Enter AWS Account ID' and a message: 'In order to share a portfolio, you will need the consumer's AWS Account ID. The AWS Account ID is available under My Account on the AWS console.' There are two radio buttons: 'AWS Account' (unchecked) and 'Organization' (checked and highlighted with a blue border). Below these are dropdowns for 'Node Type' (set to 'Organizational Unit') and 'Input Value' (containing '#####'). At the bottom are 'Cancel' and 'Share' buttons.

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

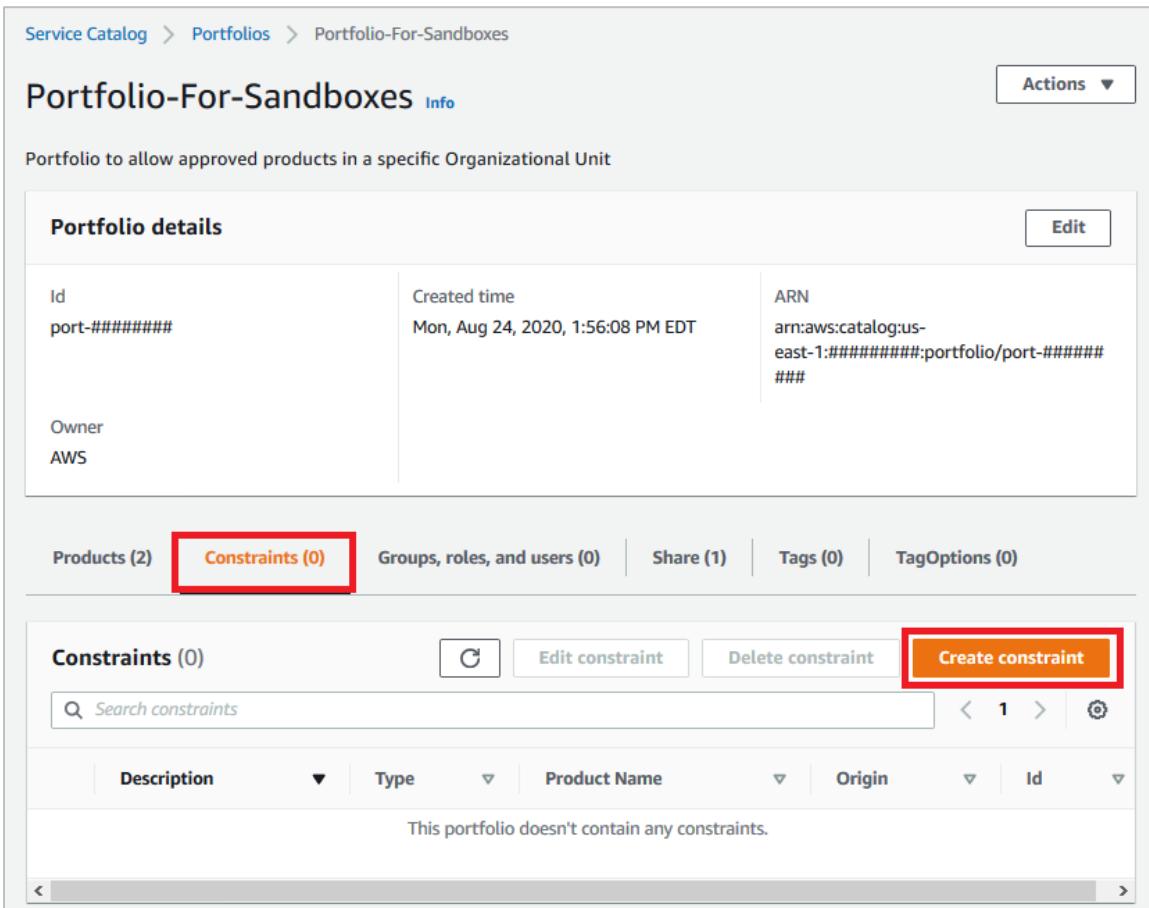


The screenshot shows a success message in a green-bordered box: 'Success: Portfolio port-##### has been shared with ou-#####'. There is also a close button 'X' in the top right corner of the message box.

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*.



The screenshot shows the AWS Service Catalog interface for a portfolio named "Portfolio-For-Sandboxes". The top navigation bar shows "Service Catalog > Portfolios > Portfolio-For-Sandboxes". Below the title, it says "Portfolio to allow approved products in a specific Organizational Unit". The main content area is titled "Portfolio details" and includes fields for Id (port-#####), Created time (Mon, Aug 24, 2020, 1:56:08 PM EDT), and ARN (arn:aws:catalog:us-east-1:#####:portfolio/port-#####). The Owner field shows "AWS". Below this, there are tabs for "Products (2)", "Constraints (0)" (which is highlighted with a red box), "Groups, roles, and users (0)", "Share (1)", "Tags (0)", and "TagOptions (0)". The "Constraints" section shows a table with columns: Description, Type, Product Name, Origin, and Id. A search bar and a "Create constraint" button are also present. A message at the bottom states "This portfolio doesn't contain any constraints."

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*.

Create constraint Info

Constraints are active as soon as you create them. When created, constraints are applied to all versions of a product that are not already launched.

Product
Select the product from the portfolio to apply the constraints.
 Amazon EC2 Linux

Constraint type
Choose the type of constraint to apply to the product you've selected.

Launch
Allows you to assign an IAM role to the product that is used to provision the AWS resources.

Notification
Allows you to stream product notifications to an Amazon SNS topic.

Template
Allows you to limit the options that are available to end users when they launch the product.

StackSet
Allows you to configure product deployment across accounts and regions using AWS CloudFormation StackSets.

Tag Update
Allows you to update tags after the product has been provisioned.

Launch constraint Info

A launch constraint specifies the AWS Identity and Access Management (IAM) role that AWS Service Catalog assumes when an end user launches a product. Only one launch constraint can be applied to each product.

Method
You can search for the IAM role you want to assign as a launch constraint, or if you know the ARN of the role you can enter it.

Select IAM role
Assign the role of the product using an IAM role in your account.

Enter ARN
Assign a role to the product using an ARN.

Enter role name
Assign a role to the product using the local role name.

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

Launch constraint Info

A launch constraint specifies the AWS Identity and Access Management (IAM) role that AWS Service Catalog assumes when an end user launches a product. Only one launch constraint can be applied to each product.

Method
You can search for the IAM role you want to assign as a launch constraint, or if you know the ARN of the role you can enter it.

Select IAM role
Assign the role of the product using an IAM role in your account.

Enter ARN
Assign a role to the product using an ARN.

Enter role name
Assign a role to the product using the local role name.

Role name
Assign an IAM role name to the launch constraint. When an account uses the launch constraint, the IAM role with that name in the account will be used.

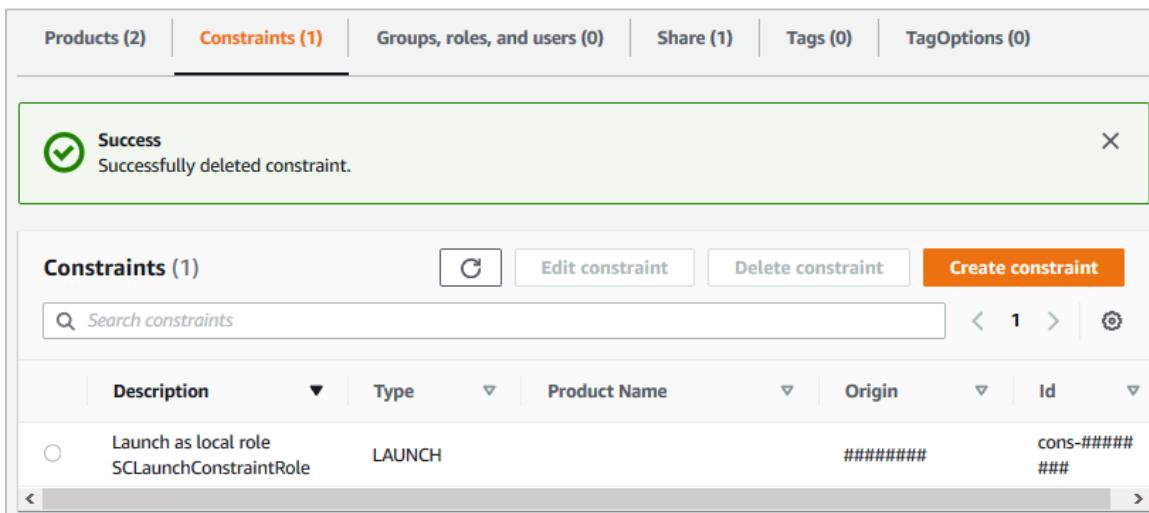
The role name can have up to 64 characters. Valid characters: a-z, A-Z, 0-9, and + , . @ ~ _

Description - optional
Describe the role assigned to the launch constraint. If left blank, this field will autopopulate with the default description. For example, "Launch as "arn:aws:iam::xxxxxxxxxxxx:abcde"

The description must contain 1-2000 characters.

Create

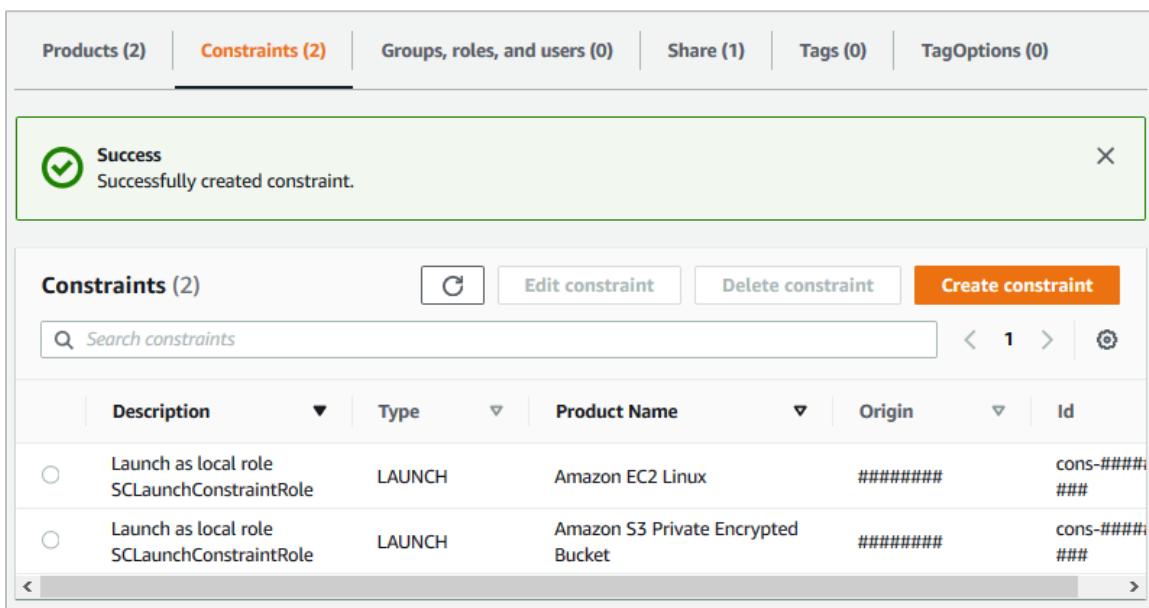
You should see a success message after the launch constraint is successfully created.



The screenshot shows the AWS Service Catalog interface with the 'Constraints' tab selected. A green success message box displays 'Successfully deleted constraint.' Below it, the 'Constraints (1)' table lists one item:

Description	Type	Product Name	Origin	Id
Launch as local role SCLaunchConstraintRole	LAUNCH	#####	#####	cons-##### ###

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.



The screenshot shows the AWS Service Catalog interface with the 'Constraints' tab selected. A green success message box displays 'Successfully created constraint.' Below it, the 'Constraints (2)' table lists two items:

Description	Type	Product Name	Origin	Id
Launch as local role SCLaunchConstraintRole	LAUNCH	Amazon EC2 Linux	#####	cons-##### ###
Launch as local role SCLaunchConstraintRole	LAUNCH	Amazon S3 Private Encrypted Bucket	#####	cons-##### ###

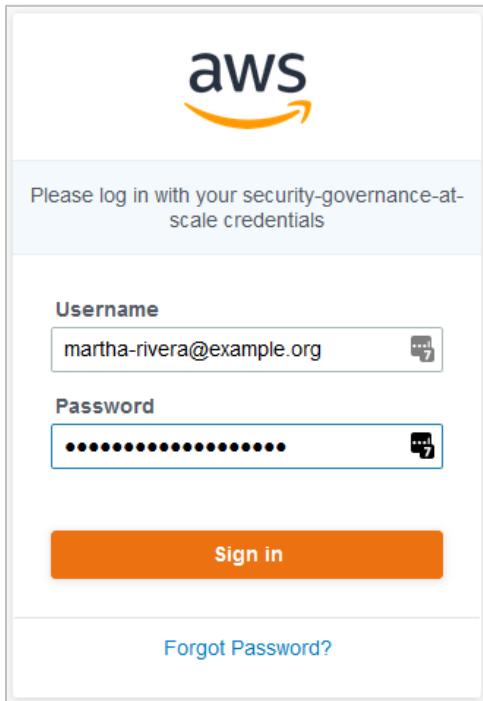
Task 2: Enable self-service in a child account

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.

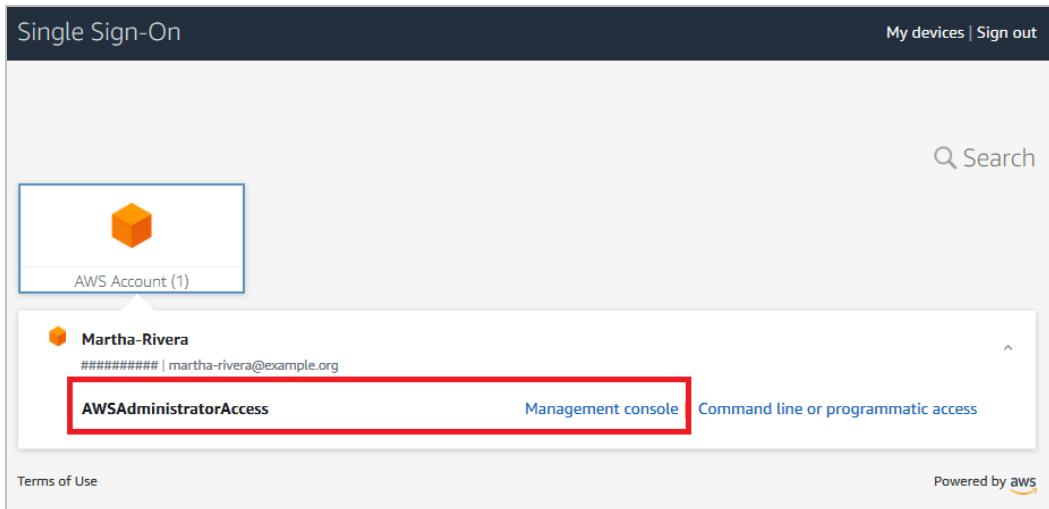
By using the AWS Service Catalog local launch constraint feature, you can add users directly to the local portfolios so they can launch AWS Service Catalog products. By using local launch constraints, there is no need to create local portfolios from imported portfolios.

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



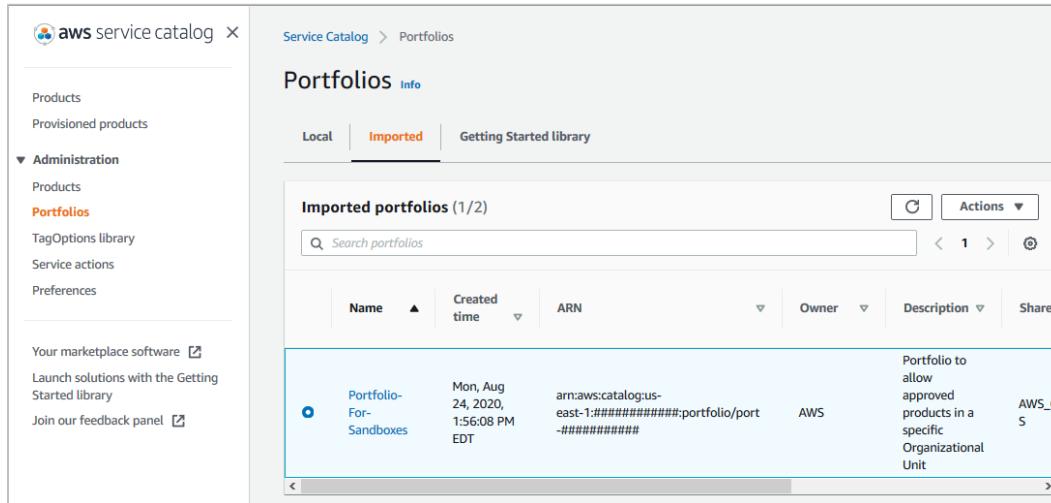
2. Choose the **AWSAdministratorAccess** role.



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.

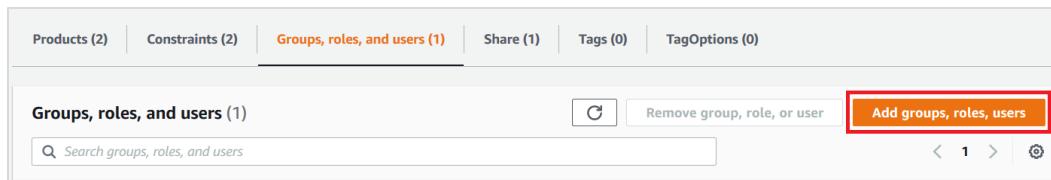
3. Open to the AWS Service Catalog console.
4. In the navigation pane, locate **Administration**, and choose **Portfolios**.
5. Select **Imported**, and then choose **Portfolio-For-Sandboxes** from the list.



The screenshot shows the AWS Service Catalog interface. The left sidebar has 'Administration' expanded, with 'Portfolios' selected. The main area shows 'Imported portfolios (1/2)' with one item listed:

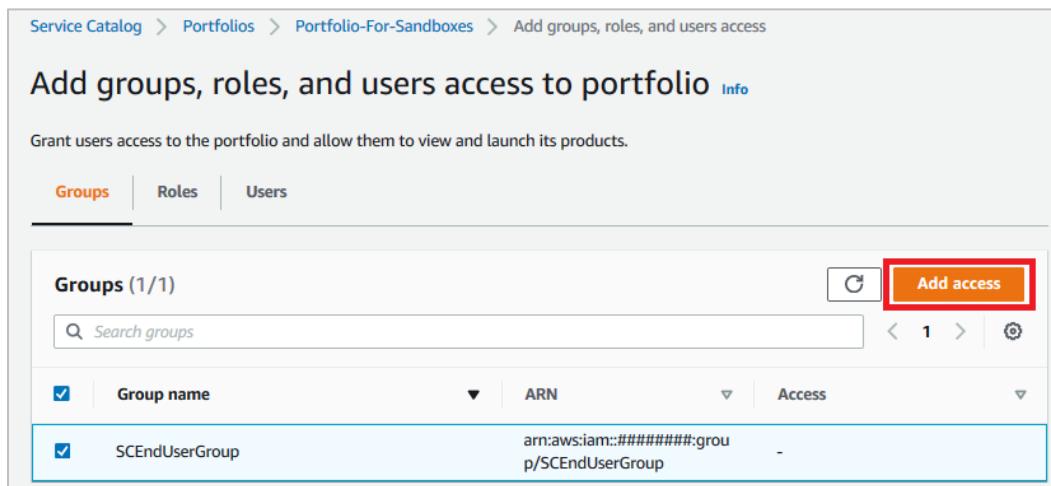
Name	Created time	ARN	Owner	Description	Share
Portfolio-For-Sandboxes	Mon, Aug 24, 2020, 1:56:08 PM EDT	arn:aws:catalog:us-east-1:#####:portfolio/port-#####	AWS	Portfolio to allow approved products in a specific Organizational Unit	AWS_C_S

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



The screenshot shows the 'Groups, roles, and users' tab selected. A red box highlights the 'Add groups, roles, users' button.

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.



The screenshot shows the 'Add groups, roles, and users access' screen. A red box highlights the 'Add access' button. The table shows the selected group:

Group name	ARN	Access
SCEndUserGroup	arn:aws:iam:#####:group/SCEndUserGroup	-

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



- 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.

The screenshot shows the AWS sign-in interface for an IAM user. It includes fields for Account ID (12 digits) or account alias (containing "your-account-id"), IAM user name (containing "SCEndUser"), and Password (containing "Change@me"). There is also a "Sign in" button and links for "Sign in using root user email" and "Forgot password?".

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.

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

The screenshot shows a table titled "Products list" with columns: Product name, Vendor, Owner, and Description. Two rows are visible: "Amazon EC2 Linux" and "Amazon S3 Private Encrypted Bucket". The first row is highlighted with a red border. The "Product name" column contains three dots before the product name. The "Owner" column for both rows is "AWS Service Catalog". The "Description" column for "Amazon EC2 Linux" states: "This product builds one Amazon Linux EC2 instance and creates a SSM patch baseline, maintenance window, and patch task to scan for and install operating system updates to the EC2 instance." The "Description" column for "Amazon S3 Private Encrypted Bucket" states: "This product builds an Amazon AWS S3 bucket encrypted with private access accessible from any source."

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

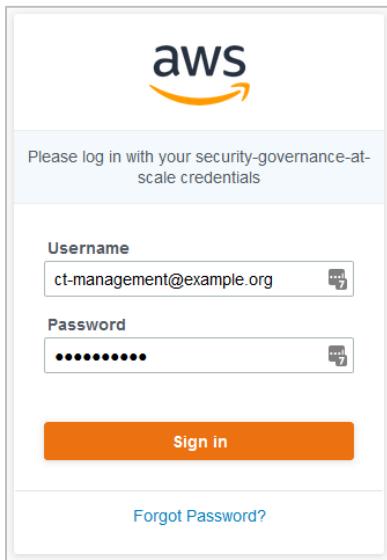
Ensure that you use the same Region for the labs. You must be in the same Region that you selected when you set up your landing zone using AWS Control Tower.

Task 1: Set up the Customizations for Control Tower solution

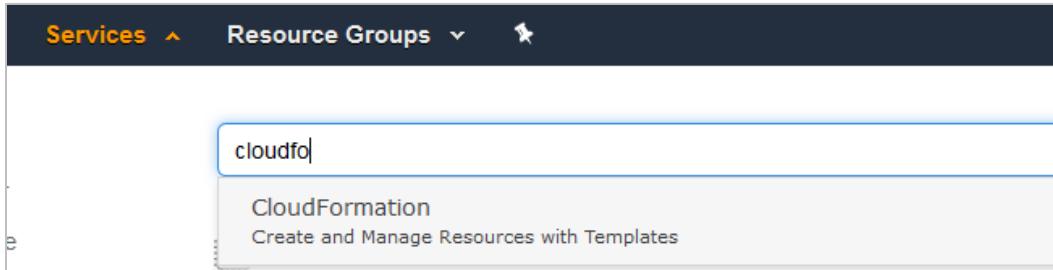
In this section of the lab, you will deploy the Customizations for Control Tower solution on your management account in your AWS Control Tower home Region. This is the AWS Region where you launched AWS Control Tower.

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

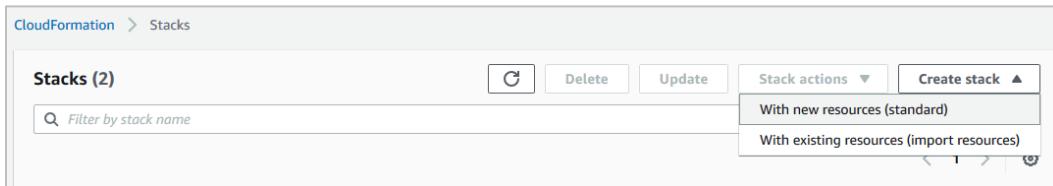
1. Using the AWS SSO user portal, log in to AWS Management Console with your management account.



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



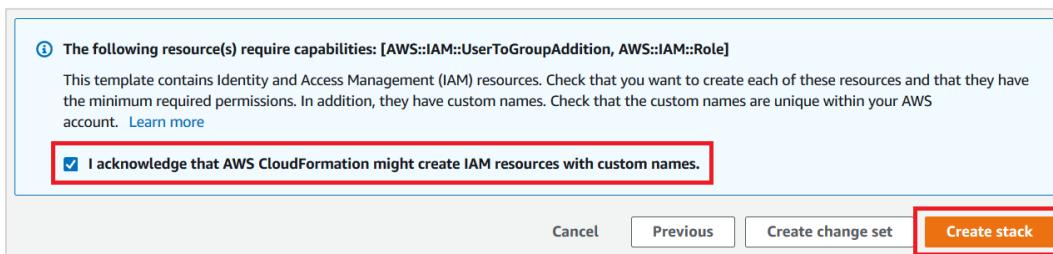
3. In the navigation pane of the AWS CloudFormation console, select **Stacks**, and then choose **Create Stack**.
4. Choose **With new resources (standard)**.



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**.

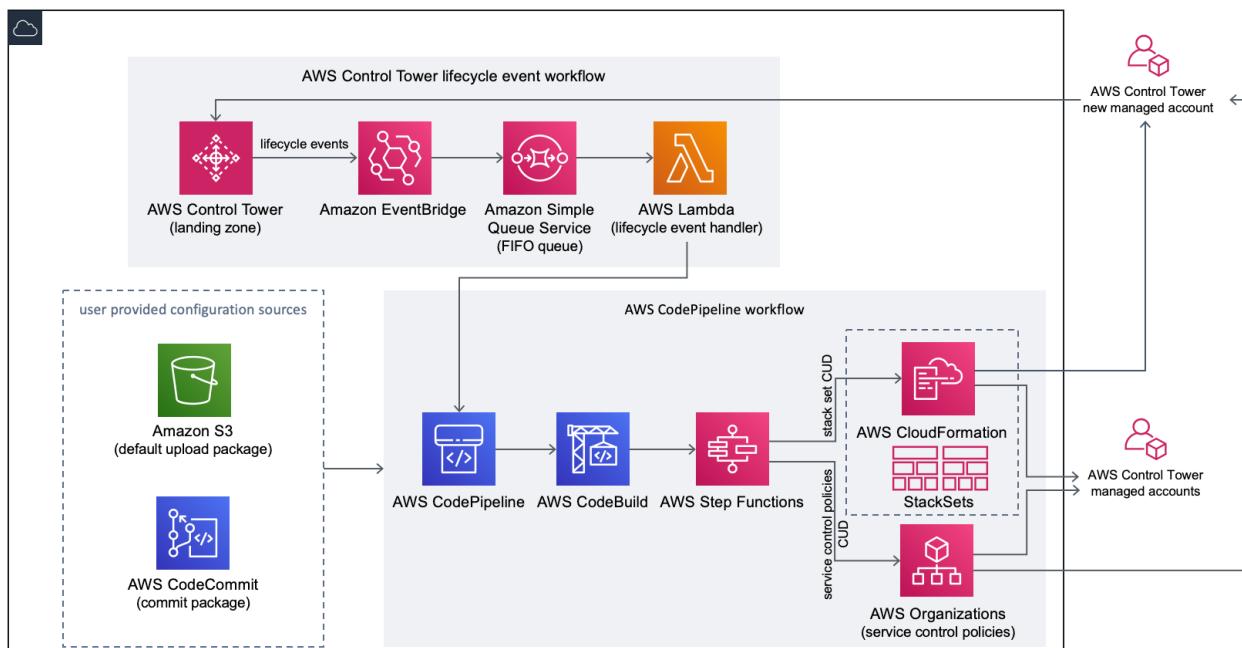


9. Accept the defaults on the Configure stack options page, and choose **Next**.
10. On the Review page, select **I acknowledge that AWS CloudFormation might create IAM resources with custom names.**
11. Choose **Create stack**.



The stack creation takes approximately 5 minutes.

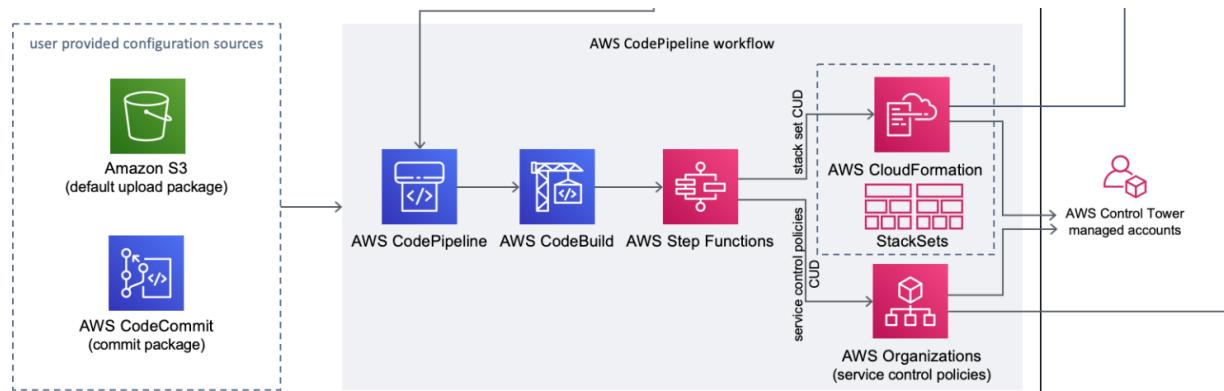
When it completes, you successfully deployed the Customizations for Control Tower solution. This solution follows best practices and is the recommended solution to customize AWS Control Tower. The following architecture diagram shows the resources provisioned by the stacks and how they connect.



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.

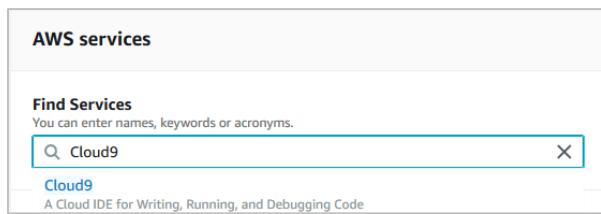
Prepare repository access with AWS Cloud9

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

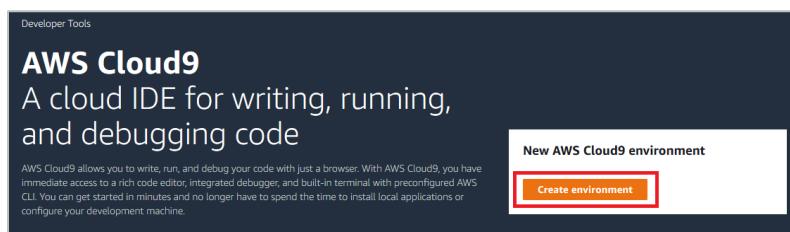
contains a collection of tools that you can use to write code and build, run, test, debug, and release software. From your development environment, you can clone existing repositories, create repositories, and commit and push code changes to a repository.

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:

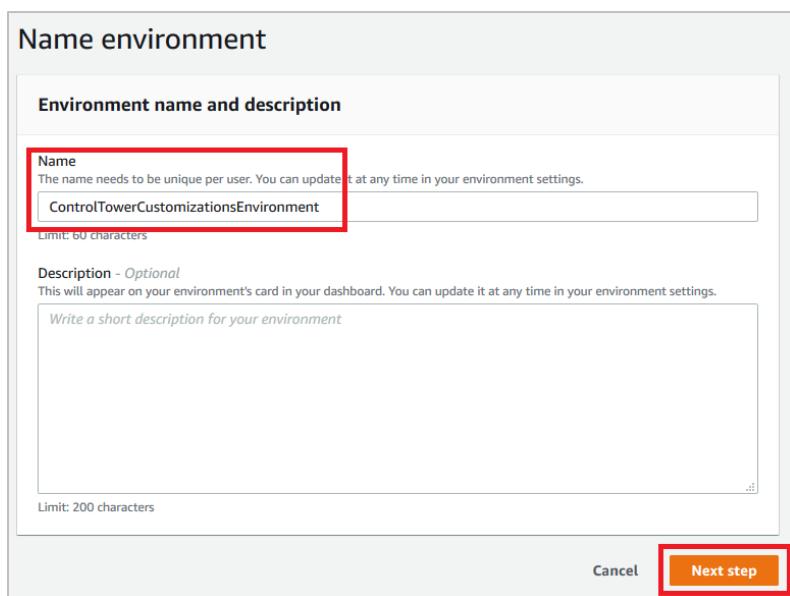
1. Use your management account to open the AWS Cloud9 console.



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*.



Name
The name needs to be unique per user. You can update it at any time in your environment settings.
<input type="text" value="ControlTowerCustomizationsEnvironment"/>
Limit: 60 characters
Description - Optional
This will appear on your environment's card in your dashboard. You can update it at any time in your environment settings.
<input type="text" value="Write a short description for your environment"/>
Limit: 200 characters
Cancel
Next step

4. In Step 2, Configure Settings, configure your environment as follows:
 - o In Environment type, choose *Create a new EC2 instance for environment (direct access)*.
 - o 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.

- o 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*.

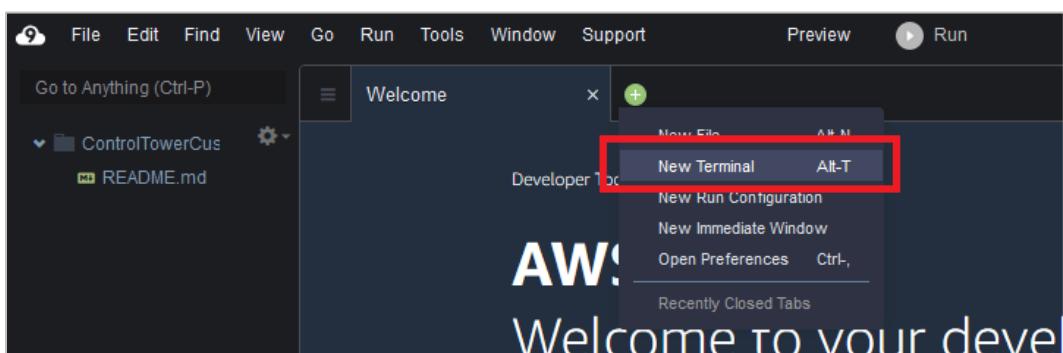
Note

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.

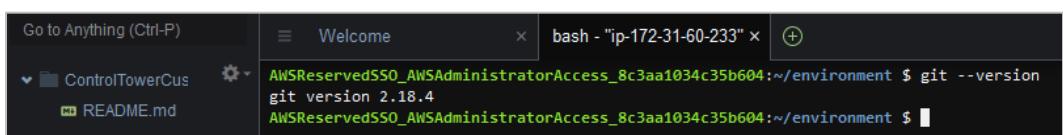
Ensure that Git is installed and is a supported version

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.


 A screenshot of a terminal window. The window title is "bash - "ip-172-31-60-233" x". The terminal prompt is "AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment \$". The user has entered the command "git --version" and pressed Return. The output of the command is displayed in green text: "git version 2.18.4". The rest of the screen is mostly black, matching the dark mode theme.

Configure an AWS CLI credential helper

After you create the AWS Cloud9 environment, you can configure an 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:

- o `git config --global credential.helper '!aws codecommit credential-helper $@'`
- o `git config --global credential.UseHttpPath true`

```
~/environment $ git config --global credential.helper '!aws codecommit credential-helper $@'  
~/environment $ git config --global credential.UseHttpPath true  
~/environment $ █
```

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

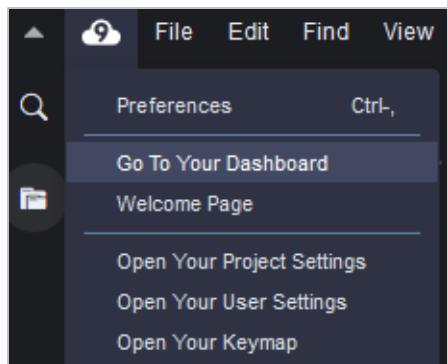
Connect to the CodeCommit

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

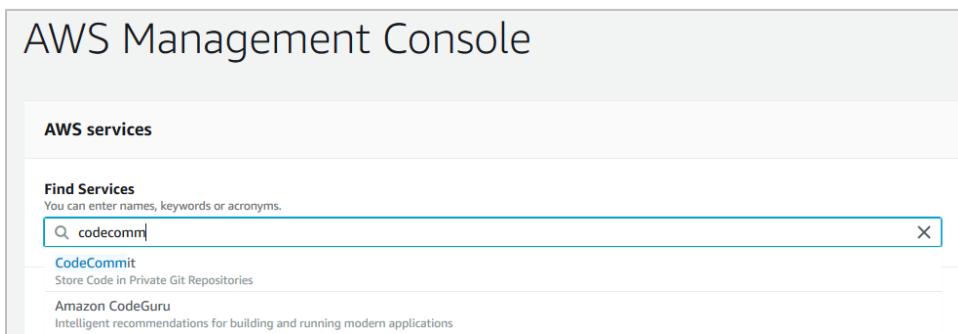
1. In your development environment, you must create a local folder to work with the code. For this example, use a folder named *lab3*. To create the folder, from the terminal, run `mkdir lab3`.

```
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment $ mkdir lab3  
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment $ ls  
lab3 README.md
```

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.



- Open the AWS CodeCommit console.



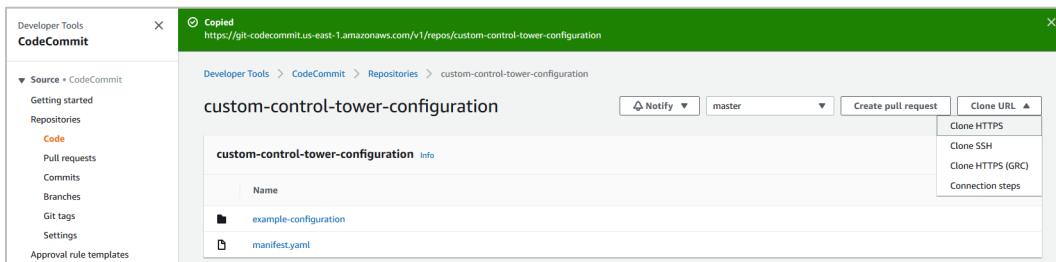
AWS Management Console

AWS services

Find Services
You can enter names, keywords or acronyms.
codecommit

- CodeCommit**
Store Code in Private Git Repositories
- Amazon CodeGuru**
Intelligent recommendations for building and running modern applications

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



Developer Tools **CodeCommit**

Source + CodeCommit

- Getting started
- Repositories
- Code**
- Pull requests
- Commits
- Branches
- Git tags
- Settings
- Approval rule templates

custom-control-tower-configuration

custom-control-tower-configuration [Info](#)

Name

- example-configuration
- manifest.yaml

Notify master Create pull request Clone URL ▲

- Clone HTTPS
- Clone SSH
- Clone HTTPS (GRC)
- Connection steps

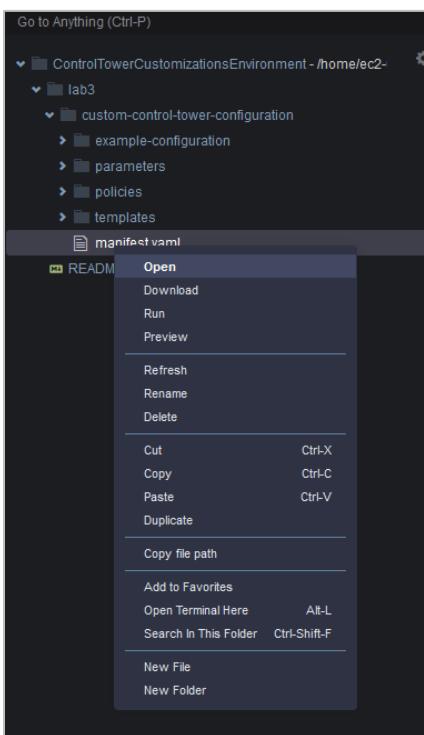
- Return to your development environment. From your terminal, in the lab3 local folder you created, clone the repository you just copied. For example, enter:

```
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.

```

1 ---
2 #!/bin/bash -e
3 # deploying Custom Control Tower: Code Pipeline, Step functions, Lambda, SSM parameters, and StackSets
4 region: us-east-1
5 Control Tower Home Region
6 version: 2020-01-01
7
  
```

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.

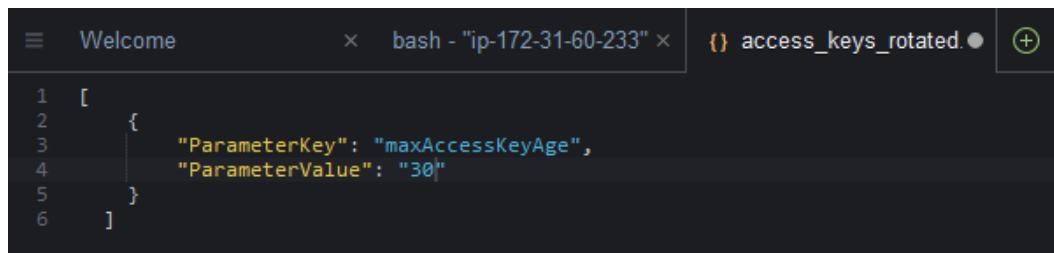
```
# Control Tower Custom CloudFormation Resources - Create Additional IAM Role
cloudformation_resources:
- name: stackset-1
  template_file: templates/describe-regions-iam-role.template
  deploy_method: stack_set
  deploy_to_ou: # :type: list
    - Research
  regions:
    - us-east-1
```

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.



```

 1  [
 2    {
 3      "ParameterKey": "maxAccessKeyAge",
 4      "ParameterValue": "30"
 5    }
 6  ]

```

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:

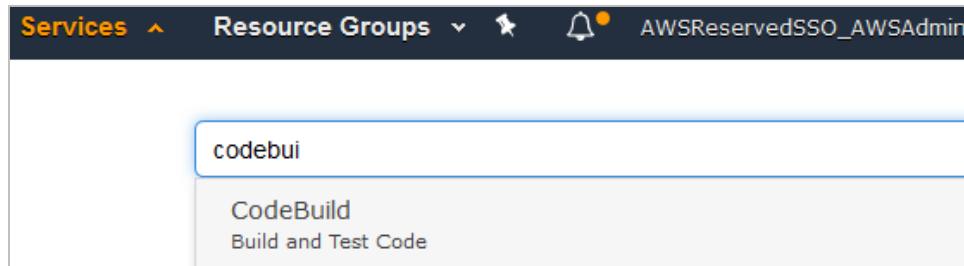
  git commit --amend --reset-author

  git commit --amend --reset-author

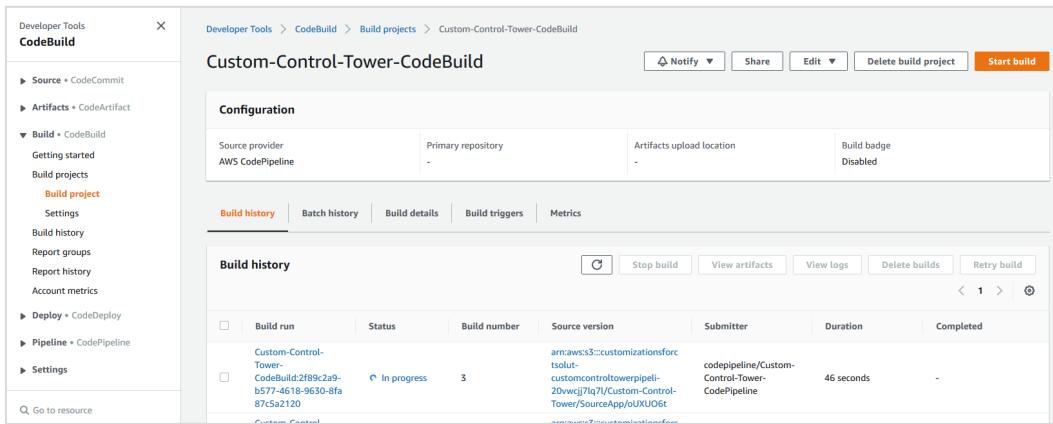
5 files changed, 119 insertions(+), 6 deletions(-)
create mode 100644 parameters/access_keys_rotated.json
create mode 100644 policies/preventive-guardrails.json
create mode 100644 templates/access_keys_rotated.template
create mode 100644 templates/describe-regions-iam-role.template
AWSReservedSSO_AWSAdministratorAccess_8c3aa1034c35b604:~/environment/lab3/custom-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.

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



- Watch the build in progress:



The screenshot shows the AWS CodeBuild console with the project 'Custom-Control-Tower-CodeBuild'. The left sidebar has a tree view with 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings' sections. The 'Build' section is expanded, showing 'Build project' (selected), 'Settings', 'Build history', 'Report groups', 'Report history', and 'Account metrics'. The main area shows the 'Configuration' tab with fields for 'Source provider' (AWS CodePipeline), 'Primary repository' (empty), 'Artifacts upload location' (empty), and 'Build badge' (Disabled). Below is the 'Build history' tab, which is active. It shows one build run: 'Custom-Control-Tower-CodeBuild:2f89:2a...b577-4618-9630-8fa87c5d2120' is in progress (Status: In progress, Build number: 3, Source version: arn:aws:s3:::customizationsfor...tsol..., Submitter: codepipeline/Custom-Control-Tower-CodePipeline, Duration: 46 seconds). Buttons at the top of the build history table include 'Stop build', 'View artifacts', 'View logs', 'Delete builds', and 'Retry build'.

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.

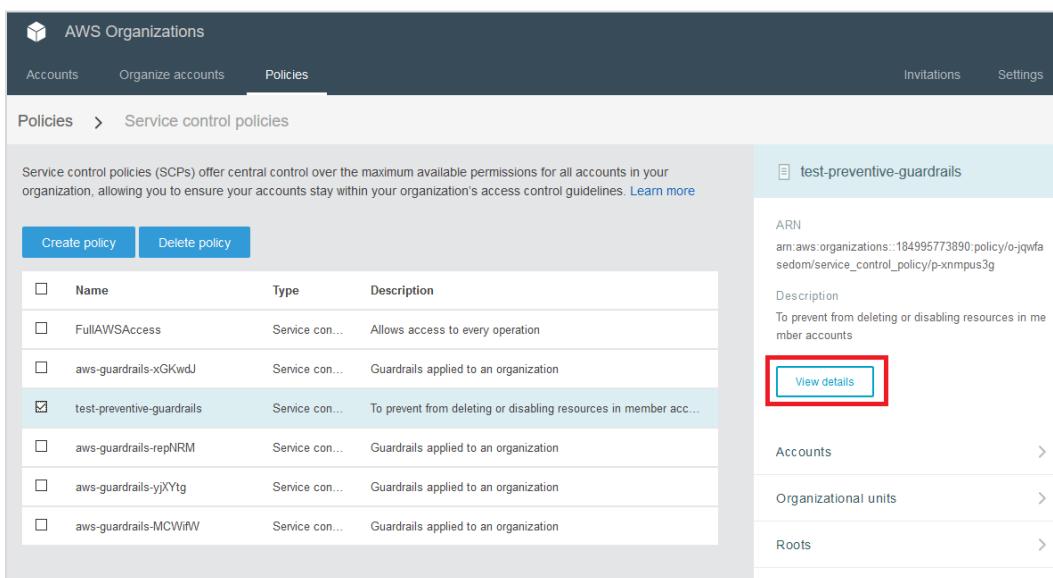
Note

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

View preventive guardrails

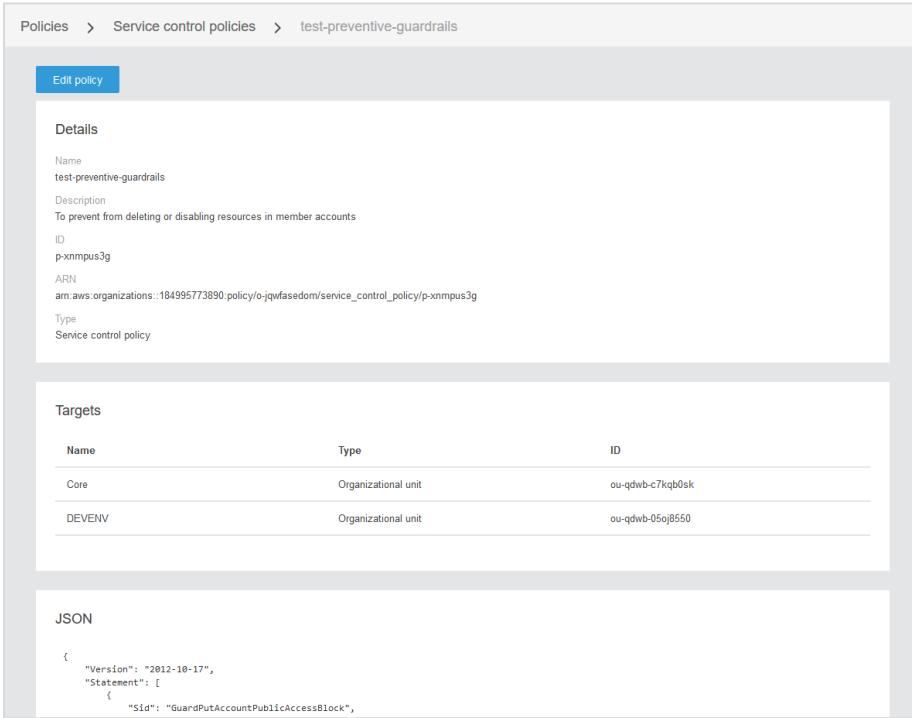
To access and review service control policies in AWS Organizations:

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



The screenshot shows the AWS Organizations console under the 'Policies' tab. The left sidebar has 'Accounts', 'Organize accounts', 'Policies' (selected), 'Invitations', and 'Settings'. The main area shows the 'Service control policies' list with policies like 'FullAWSAccess', 'aws-guardrails-xGKwdJ', 'test-preventive-guardrails' (selected), 'aws-guardrails-repNRM', 'aws-guardrails-yXtYtg', and 'aws-guardrails-MCWiW'. A red box highlights the 'View details' button for the selected policy. On the right, the details for 'test-preventive-guardrails' are shown: ARN (arn:aws:organizations::184995773890:policy/o-jqwfa...sedom/service_control_policy/p-xnmpus3g), Description (To prevent from deleting or disabling resources in member accounts), and three navigation links: 'Accounts', 'Organizational units', and 'Roots'.

3. Explore the details.



The screenshot shows the AWS Config console interface for viewing a service control policy. The policy name is 'test-preventive-guardrails'. The description states 'To prevent from deleting or disabling resources in member accounts'. The ARN is 'arn:aws:organizations::184995773890:policy/o-jqwfasedom/service_control_policy/p-xnmpus3g'. The type is 'Service control policy'. Below this, the 'Targets' section lists two organizational units: 'Core' and 'DEVENV'. In the JSON section, the policy document is displayed:

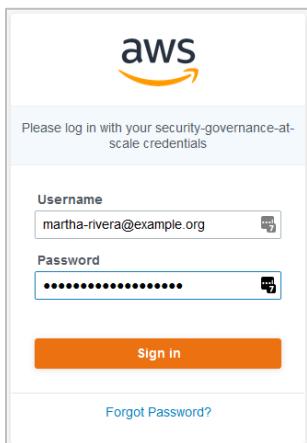
```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "GuardPutAccountPublicAccessBlock",
      ...
    }
  ]
}
```

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



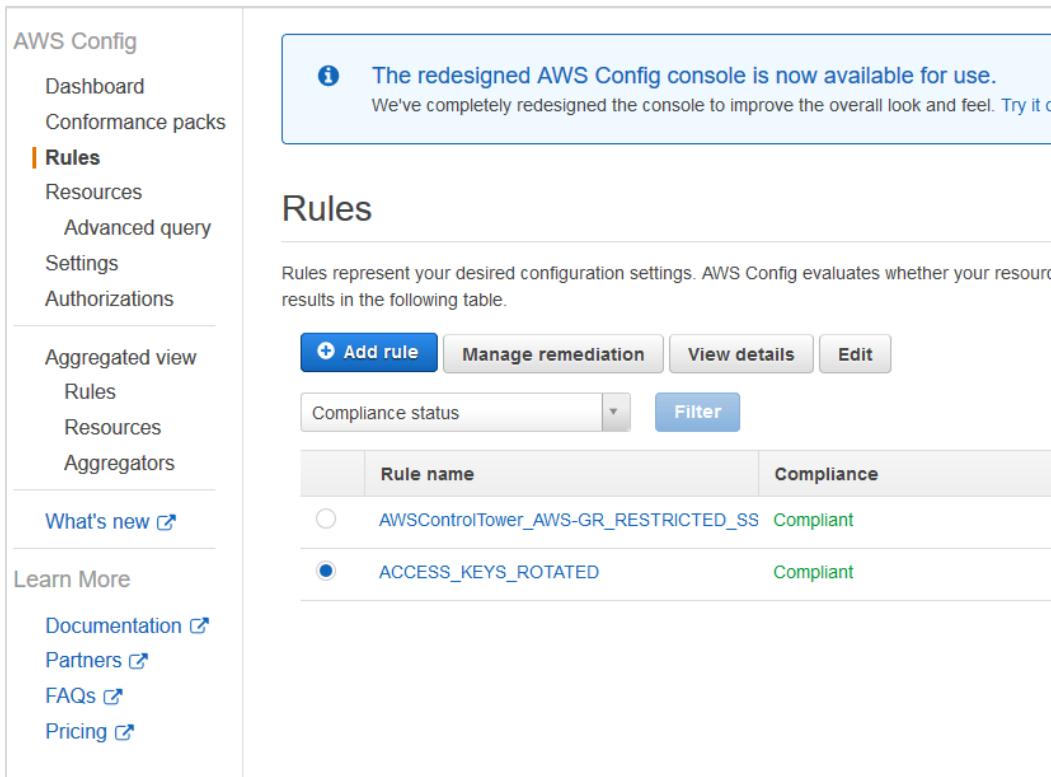
3. Choose the **AWSAdministratorAccess** role.

A screenshot of the AWS Single Sign-On dashboard. The top navigation bar includes "Single Sign-On", "My devices | Sign out", and a search bar. Below the navigation is a section titled "AWS Account (1)" with a cube icon. The main content area shows a user profile for "Martha-Rivera" with the email "martha-rivera@example.org". Underneath the profile, there are three buttons: "AWSAdministratorAccess" (which is highlighted with a red box), "Management console", and "Command line or programmatic access". At the bottom of the page are links for "Terms of Use" and "Powered by aws".

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

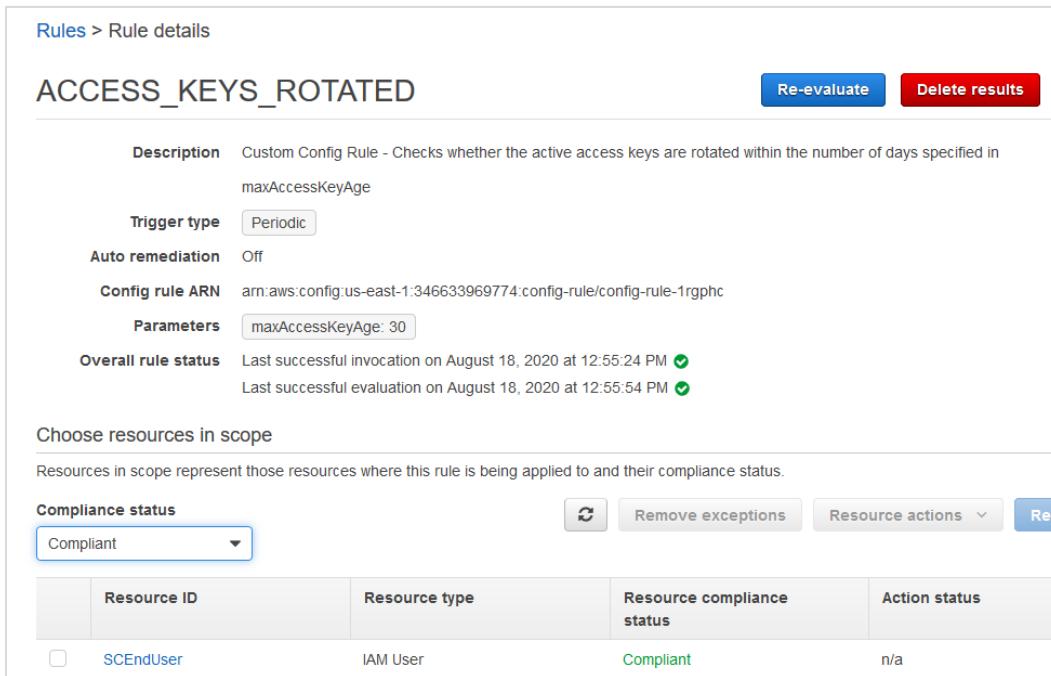
A screenshot of the AWS services search interface. The title is "AWS services". Below it is a "Find Services" section with a search bar containing "config". A result for "Config" is listed, described as "Track Resource Inventory and Changes".

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



The screenshot shows the AWS Config Rules page. On the left, there's a sidebar with links like Dashboard, Conformance packs, Rules (which is selected), Resources, Advanced query, Settings, Authorizations, Aggregated view, Rules, Resources, Aggregators, What's new, Learn More, Documentation, Partners, FAQs, and Pricing. A prominent message box says: "The redesigned AWS Config console is now available for use. We've completely redesigned the console to improve the overall look and feel. Try it out." The main content area is titled "Rules" and contains a table of rules. The table has columns for Rule name and Compliance. It shows two entries: "AWSControlTower_AWS-GR_RESTRICTED_SS" (Compliant) and "ACCESS_KEYS_ROTATED" (Compliant). There are buttons for Add rule, Manage remediation, View details, and Edit.

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



The screenshot shows the "Rule details" page for the "ACCESS_KEYS_ROTATED" rule. At the top, it says "Rules > Rule details". The rule name is "ACCESS_KEYS_ROTATED". There are buttons for Re-evaluate and Delete results. The rule details include:

- Description:** Custom Config Rule - Checks whether the active access keys are rotated within the number of days specified in maxAccessKeyAge.
- Trigger type:** Periodic
- Auto remediation:** Off
- Config rule ARN:** arn:aws:config:us-east-1:346633969774:config-rule/config-rule-1rgphc
- Parameters:** maxAccessKeyAge: 30
- Overall rule status:** Last successful invocation on August 18, 2020 at 12:55:24 PM (green checkmark)
Last successful evaluation on August 18, 2020 at 12:55:54 PM (green checkmark)

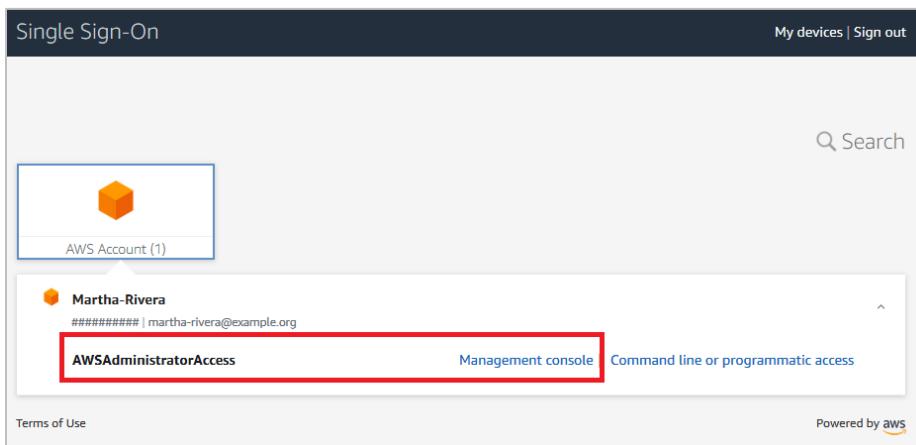
The "Choose resources in scope" section shows a table of resources. The table has columns for Resource ID, Resource type, Resource compliance status, and Action status. It lists one resource: "SEndUser" (IAM User, Compliant, n/a).

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.

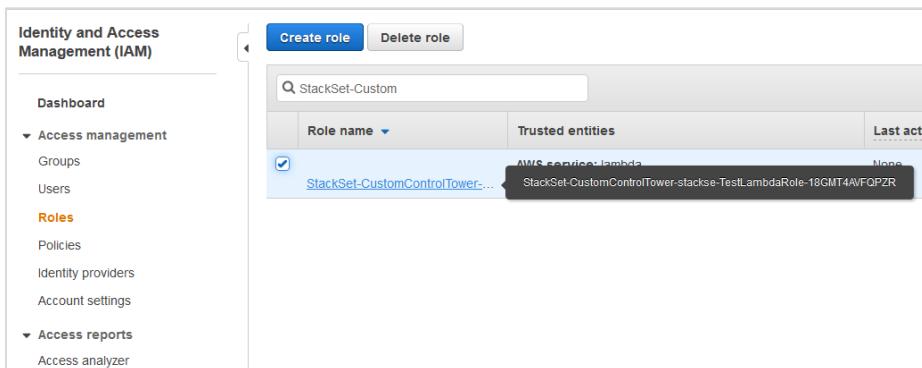
View IAM role

Access IAM console to review custom roles:

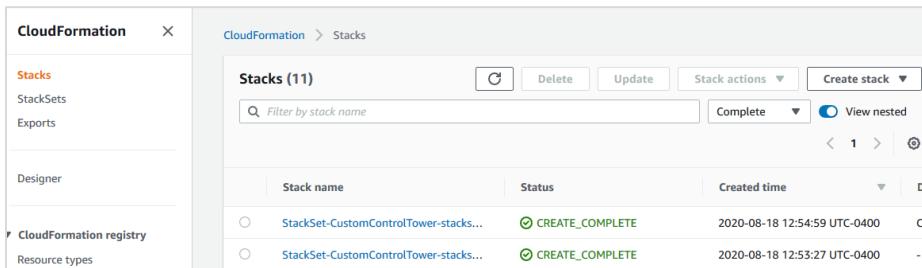
1. Log in to a provisioned account (an account such as Research or Custom OU) with AWSAdministratorAccess. You can use Martha's account for this step.



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.

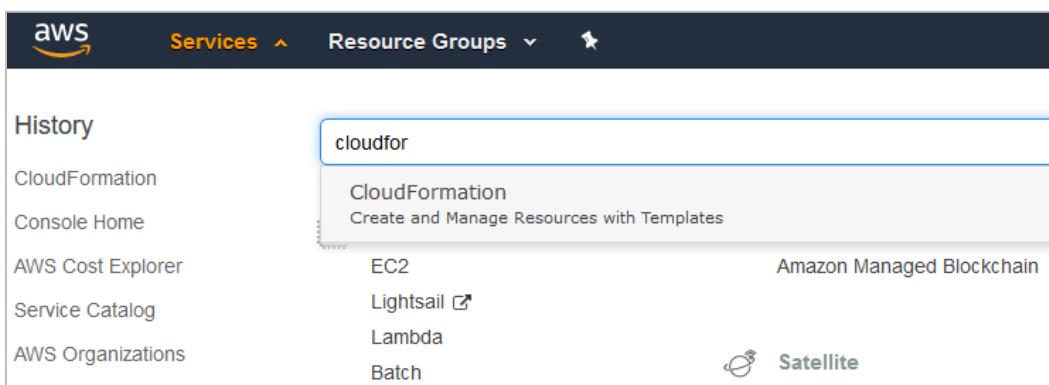
Decommission and clean up Lab 3

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

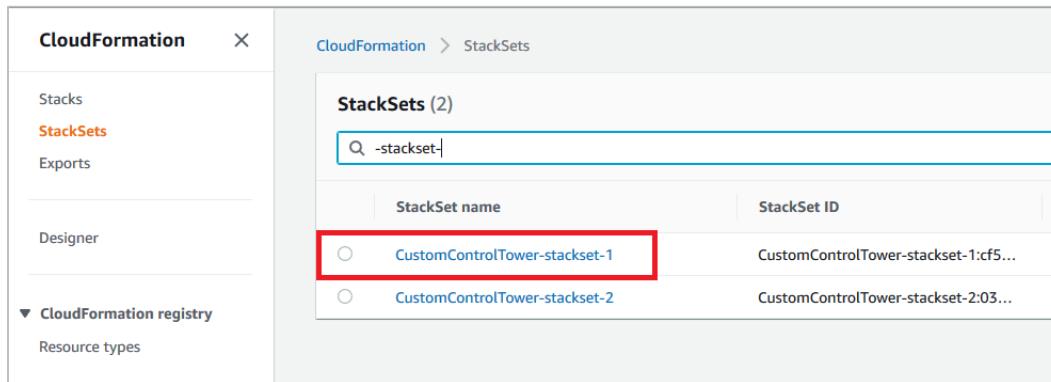
Delete the AWS CloudFormation StackSets

To delete the StackSets:

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

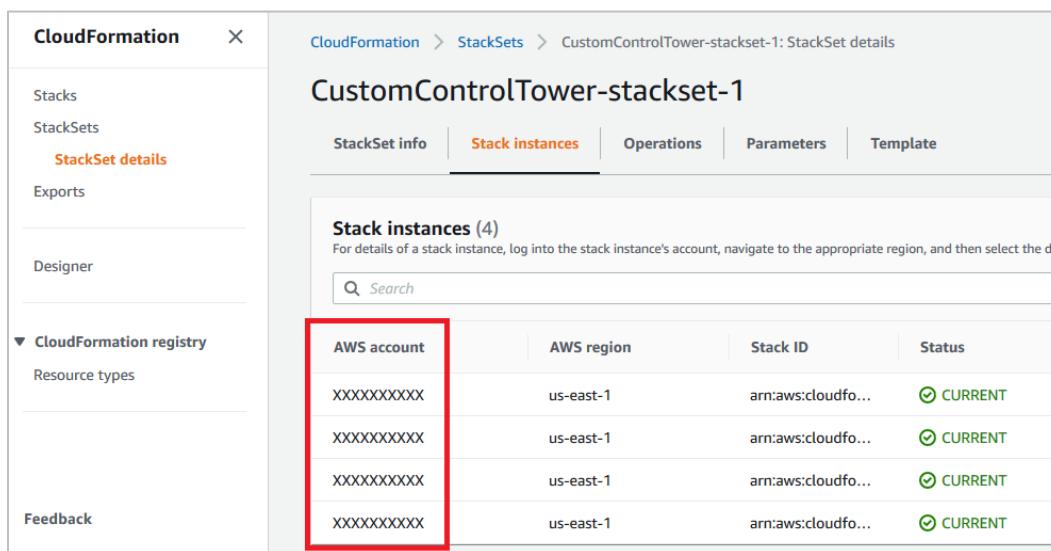


2. In the navigation pane, got to **StackSets**, select the **stackset-1** deployed by this solution.



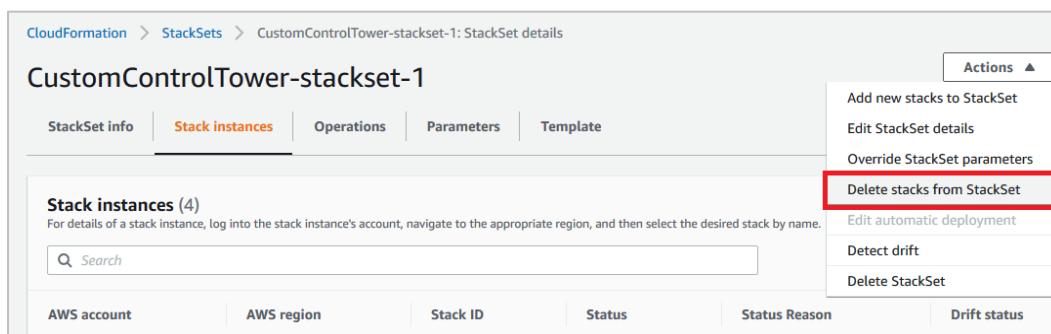
StackSet name	StackSet ID
CustomControlTower-stackset-1	CustomControlTower-stackset-1:cf5...
CustomControlTower-stackset-2	CustomControlTower-stackset-2:03...

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



AWS account	AWS region	Stack ID	Status
XXXXXXXXXX	us-east-1	arn:aws:cloudfo...	CURRENT
XXXXXXXXXX	us-east-1	arn:aws:cloudfo...	CURRENT
XXXXXXXXXX	us-east-1	arn:aws:cloudfo...	CURRENT
XXXXXXXXXX	us-east-1	arn:aws:cloudfo...	CURRENT

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



- Add new stacks to StackSet
- Edit StackSet details
- Override StackSet parameters
- Delete stacks from StackSet**
- Edit automatic deployment
- Detect drift
- Delete StackSet

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

CustomControlTower-stackset-1: Delete stacks from StackSet

Set deployment options

Accounts

Identify accounts or organizational units in which you want to modify stacks

Deployment locations

StackSets can be deployed into accounts or an organizational unit.

Deploy stacks in accounts Deploy stacks in organizational units

Account numbers

Enter account numbers or populate from a file.

1234567891,1234567892,1234567893

12-Digit account numbers separated by commas.

Upload .csv file No file chosen

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

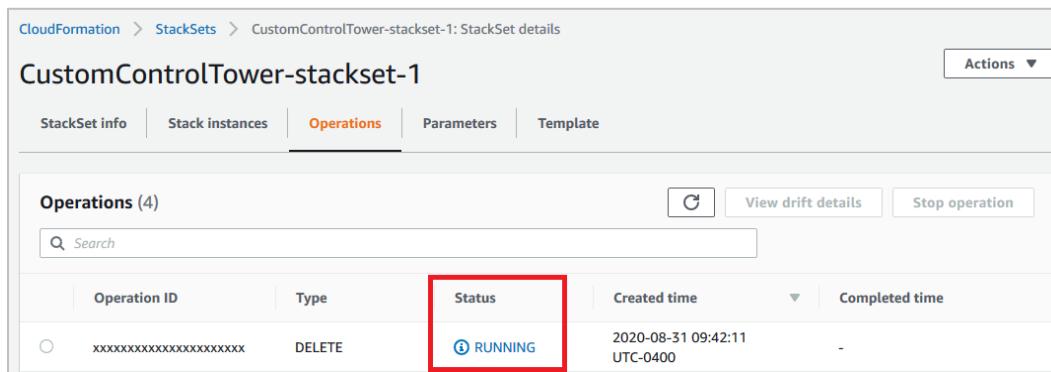
Specify regions

Choose the regions in which you want to deploy stacks. Stacks are deployed in these regions in the order that you specify. Note that during stack set operations, administrator and target accounts exchange metadata regarding the accounts themselves, as well as the stack set and stack set instances involved. [Learn more](#)

US East (N.Virginia)	▼	▲	▼	Remove
	▼	▲	▼	Remove

Add all regions **Remove all regions**

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

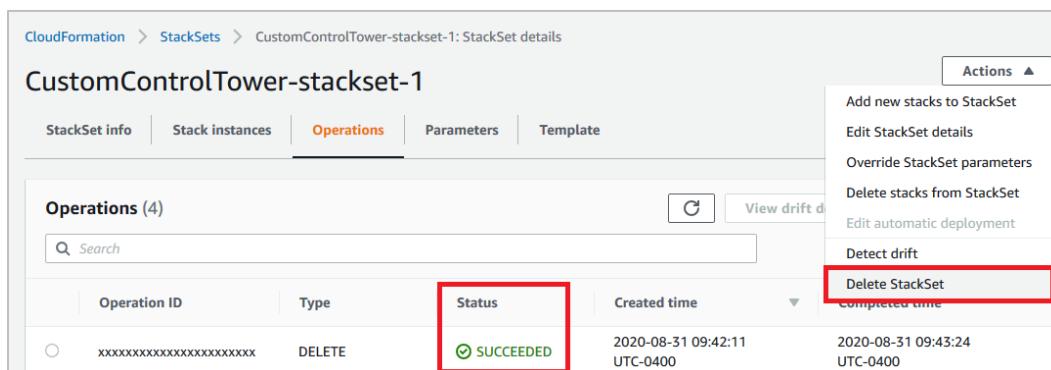


CustomControlTower-stackset-1

Operations (4)

Operation ID	Type	Status	Created time	Completed time
xxxxxxxxxxxxxxxxxxxxxx	DELETE	(i) RUNNING	2020-08-31 09:42:11 UTC-0400	-

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

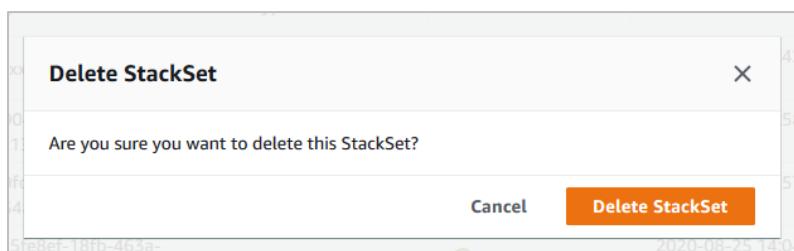


CustomControlTower-stackset-1

Operations (4)

Operation ID	Type	Status	Created time	Completed time
xxxxxxxxxxxxxxxxxxxxxx	DELETE	(i) SUCCEEDED	2020-08-31 09:42:11 UTC-0400	2020-08-31 09:43:24 UTC-0400

- Confirm the deletion when prompted.



- Repeat these steps for stackset-2.

Note

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

CustomControlTower-stackset-2: Delete stacks from StackSet

Set deployment options

Accounts
 Identify accounts or organizational units in which you want to modify stacks

Deployment locations
 StackSets can be deployed into accounts or an organizational unit.

Deploy stacks in accounts Deploy stacks in organizational units

Account numbers
 Enter account numbers or populate from a file.
 1XXXXXXXXXX3

12-Digit account numbers separated by commas.

Upload .csv file

Specify regions

Choose the regions in which you want to deploy stacks. Stacks are deployed in these regions in the order that you specify. Note that during stack set operations, administrator and target accounts exchange metadata regarding the accounts themselves, as well as the stack set and stack set instances involved. [Learn more](#)

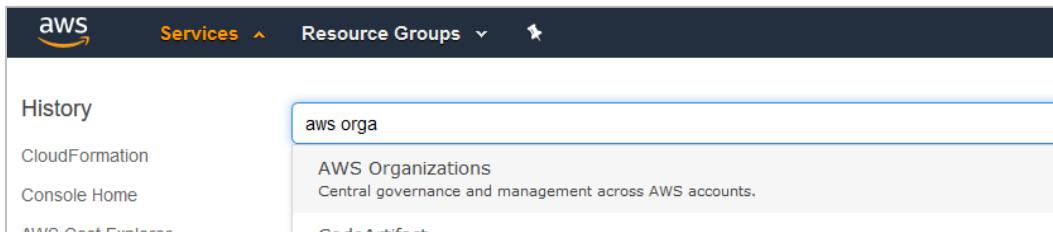
EU (Ireland)	▼	^	▼	Remove
US East (N.Virginia)	▼	^	▼	Remove
US East (Ohio)	▼	^	▼	Remove
US West (Oregon)	▼	^	▼	Remove
	▼	^	▼	Remove

Add all regions Remove all regions

Detach and delete the service control policies

Next, you will detach and delete the SCPs:

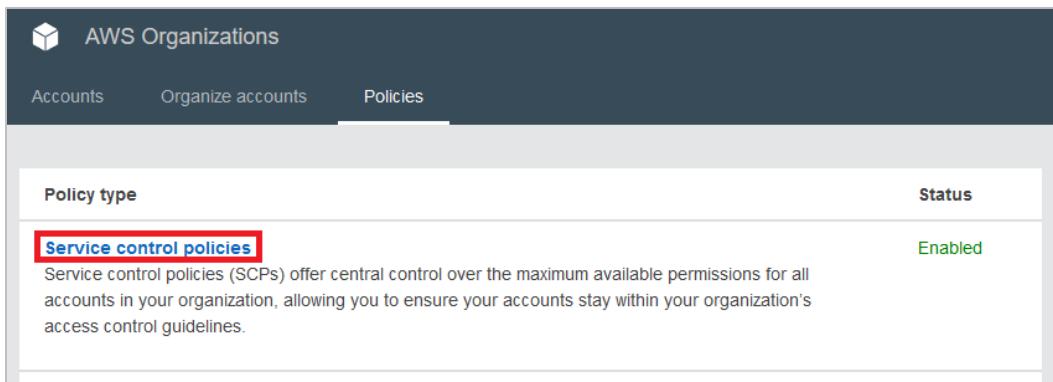
1. Open the AWS Organizations console.



AWS Organizations

Central governance and management across AWS accounts.

2. Select the **Policies** tab.

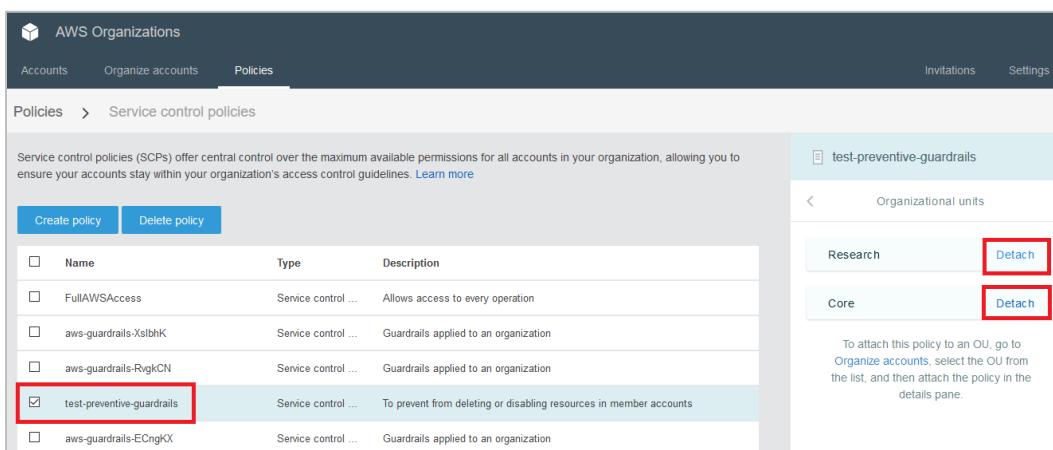


Policy type	Status
Service control policies	Enabled

Service control policies (SCPs) offer central control over the maximum available permissions for all accounts in your organization, allowing you to ensure your accounts stay within your organization's access control guidelines.

3. Select **test-preventative-guardrails**, and choose **Organizations units**.

4. Choose **Detach** for all OUs.



Name	Type	Description
FullAWSAccess	Service control ...	Allows access to every operation
aws-guardrails-XslbhK	Service control ...	Guardrails applied to an organization
aws-guardrails-RvgiCN	Service control ...	Guardrails applied to an organization
test-preventive-guardrails	Service control ...	To prevent from deleting or disabling resources in member accounts
aws-guardrails-ECngiKX	Service control ...	Guardrails applied to an organization

test-preventive-guardrails

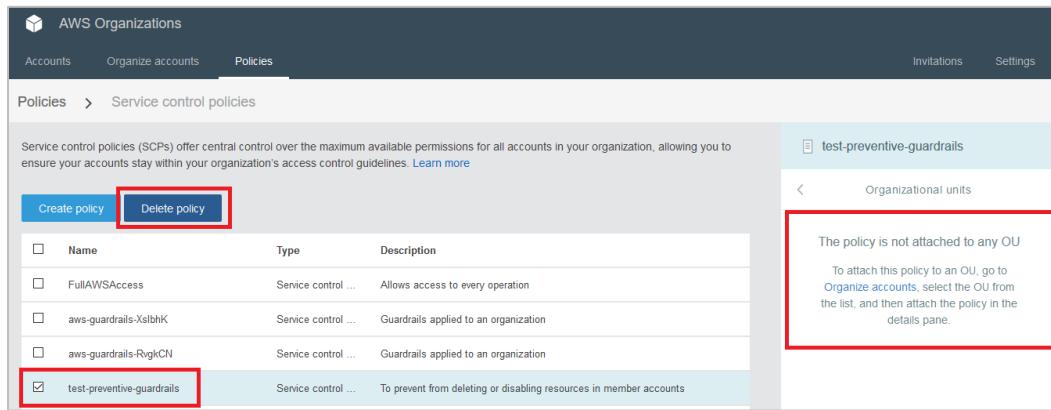
Organizational units

Research Detach

Core Detach

To attach this policy to an OU, go to **Organize accounts**, select the OU from the list, and then attach the policy in the details pane.

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



The screenshot shows the AWS Organizations console under the 'Policies' tab, specifically the 'Service control policies' section. At the top, there are buttons for 'Create policy' and 'Delete policy'. Below is a table listing policies:

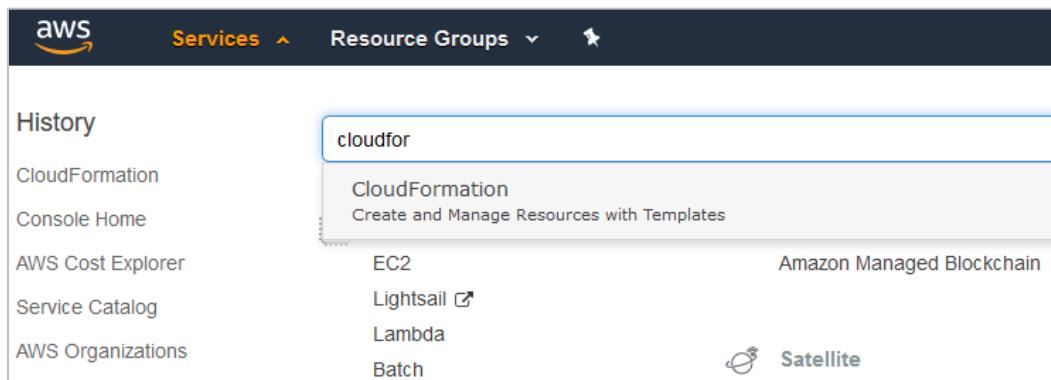
Name	Type	Description
FullAWSAccess	Service control ...	Allows access to every operation
aws-guardrails-XslbhK	Service control ...	Guardrails applied to an organization
aws-guardrails-RvgkCN	Service control ...	Guardrails applied to an organization
test-preventive-guardrails	Service control ...	To prevent from deleting or disabling resources in member accounts

A modal window for the 'test-preventive-guardrails' policy is open on the right, stating: 'The policy is not attached to any OU. To attach this policy to an OU, go to Organize accounts, select the OU from the list, and then attach the policy in the details pane.' This text is also highlighted with a red box.

Delete the customization solution stack from Lab 3

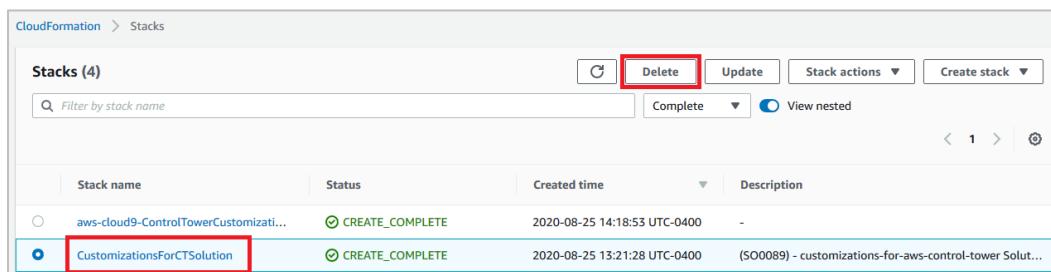
Now, you will delete the customization solution stack:

1. Open the AWS CloudFormation console.



The screenshot shows the AWS Services navigation bar. The 'CloudFormation' service is selected and highlighted in blue. Other services like EC2, Lightsail, Lambda, and Batch are also listed.

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



The screenshot shows the AWS CloudFormation 'Stacks' page. There are four stacks listed:

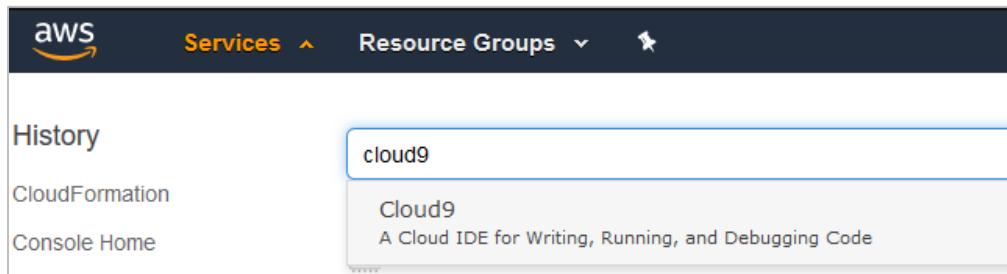
Stack name	Status	Created time	Description
aws-cloud9-ControlTowerCustomizati...	CREATE_COMPLETE	2020-08-25 14:18:53 UTC-0400	-
CustomizationsForCTSolution	CREATE_COMPLETE	2020-08-25 13:21:28 UTC-0400	(SO0089) - customizations-for-aws-control-tower Solut...

The 'CustomizationsForCTSolution' stack is highlighted with a red box. The 'Delete' button in the top toolbar is also highlighted with a red box.

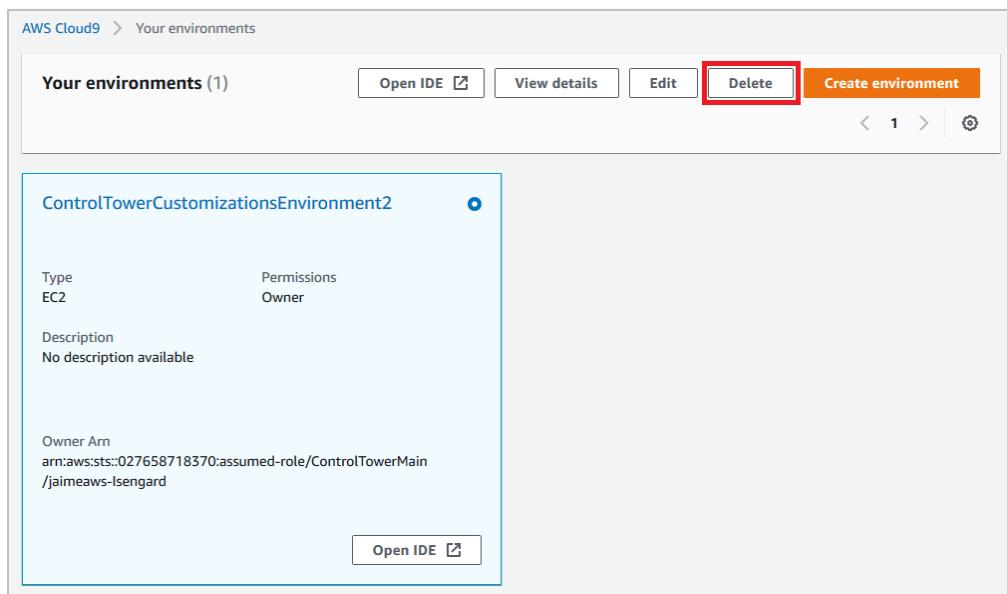
Delete AWS Cloud9 development environment

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

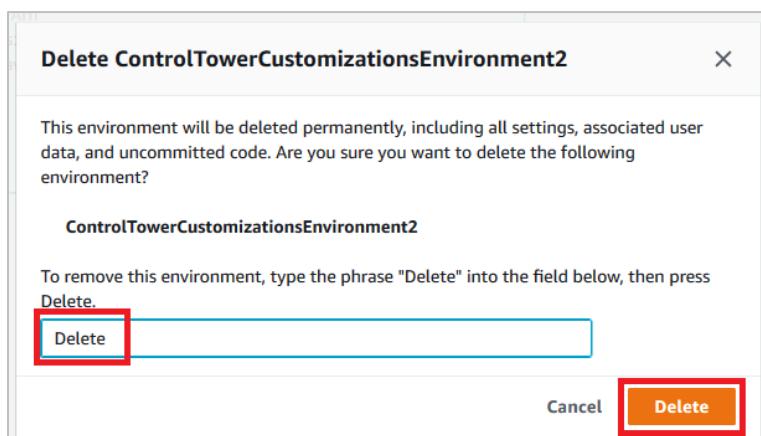
1. Open the AWS Cloud9 console.



2. Delete the environment you used for Lab 3.



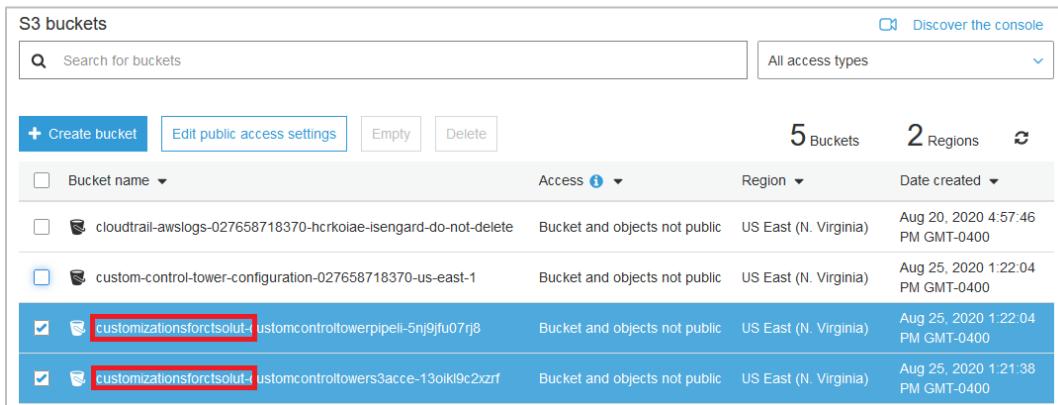
3. Enter **Delete** when prompted, and confirm the deletion.



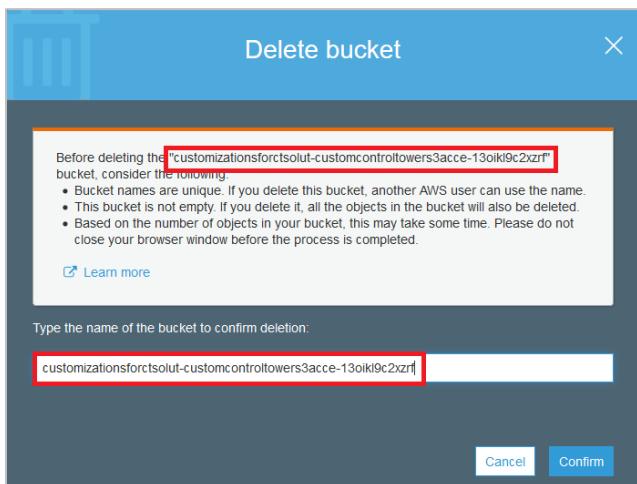
Note

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*.

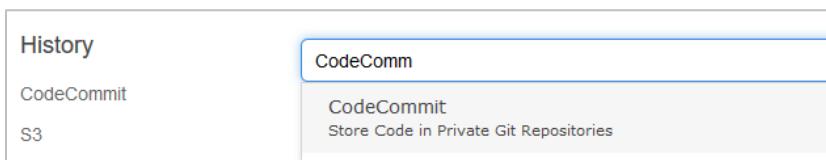


The screenshot shows the AWS S3 buckets console. There are 5 Buckets and 2 Regions listed. Two buckets have been selected for deletion: 'customizationforctsolut-' and 'customizationforctsolut-'. Both buckets are located in US East (N. Virginia) and were created on Aug 25, 2020.



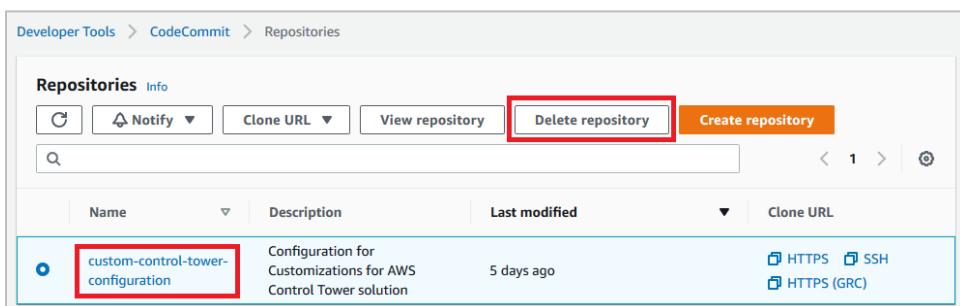
The dialog box is titled 'Delete bucket'. It contains a warning message: 'Before deleting the "customizationforctsolut-customcontrollowers3acce-13oiik9c2xzf" bucket, consider the following.' The message lists three points: Bucket names are unique, This bucket is not empty, and Based on the number of objects in your bucket, this may take some time. A 'Learn more' link is provided. Below the message, a text input field contains the bucket name 'customizationforctsolut-customcontrollowers3acce-13oiik9c2xzf'. At the bottom are 'Cancel' and 'Confirm' buttons.

5. Navigate to the CodeCommit repository.



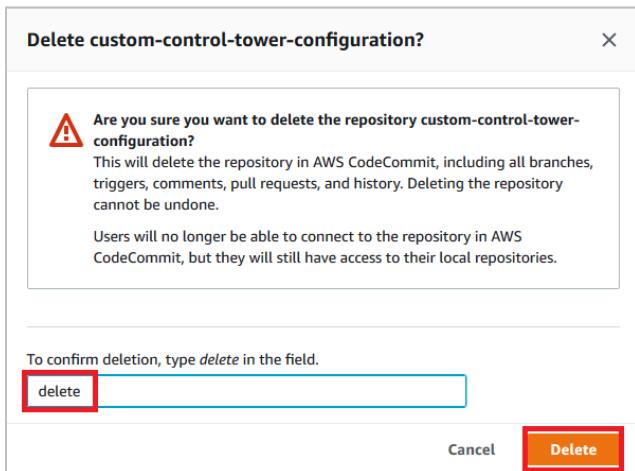
The screenshot shows the AWS CodeCommit History page. The sidebar has links for History, CodeCommit, and S3. The main area shows a search bar with 'CodeComm' typed in, and a description below it: 'CodeCommit Store Code in Private Git Repositories'.

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

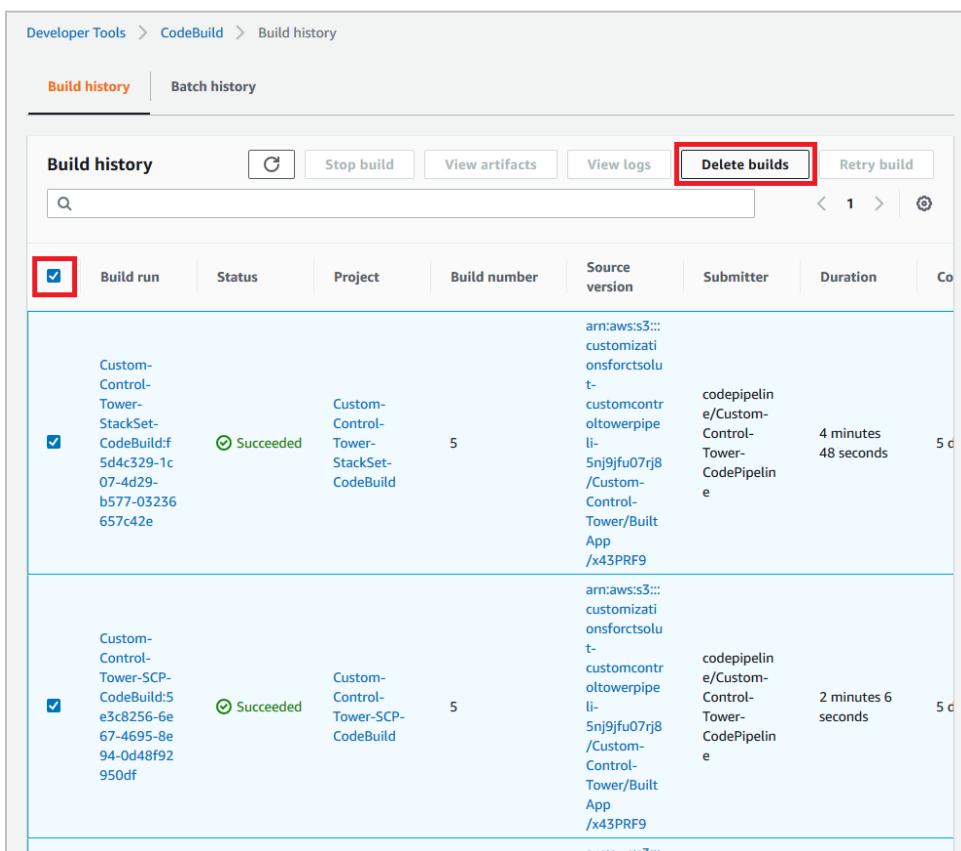


The screenshot shows the AWS CodeCommit Repositories page. The top navigation bar includes 'Developer Tools > CodeCommit > Repositories'. The 'Repositories' tab is selected. A red box highlights the 'Delete repository' button in the top right of the toolbar. The table below lists one repository: 'custom-control-tower-configuration'. A red box highlights the repository name. The table columns are Name, Description, Last modified, and Clone URL.

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



8. From AWS CodeBuild, delete the builds:



Build run	Status	Project	Build number	Source version	Submitter	Duration	Co
Custom-Control-Tower-StackSet-CodeBuild: 5d4c329-1c07-4d29-b577-03236657c42e	Succeeded	Custom-Control-Tower-StackSet-CodeBuild	5	arn:aws:s3:::customizatonsforstsolt-customcontrltoverpipe <li-5nj9jfu07rj8 app="" built="" custon-control-tower="" td="" x43prf9<=""> <td>codepipeline/e/Custom-Control-Tower-CodePipeline</td> <td>4 minutes 48 seconds</td> <td>5 d</td> </li-5nj9jfu07rj8>	codepipeline/e/Custom-Control-Tower-CodePipeline	4 minutes 48 seconds	5 d
Custom-Control-Tower-SCP-CodeBuild: 5e3c8256-6e67-4695-8e94-0d48f92950df	Succeeded	Custom-Control-Tower-SCP-CodeBuild	5	arn:aws:s3:::customizatonsforstsolt-customcontrltoverpipe <li-5nj9jfu07rj8 app="" built="" custon-control-tower="" td="" x43prf9<=""> <td>codepipeline/e/Custom-Control-Tower-CodePipeline</td> <td>2 minutes 6 seconds</td> <td>5 d</td> </li-5nj9jfu07rj8>	codepipeline/e/Custom-Control-Tower-CodePipeline	2 minutes 6 seconds	5 d

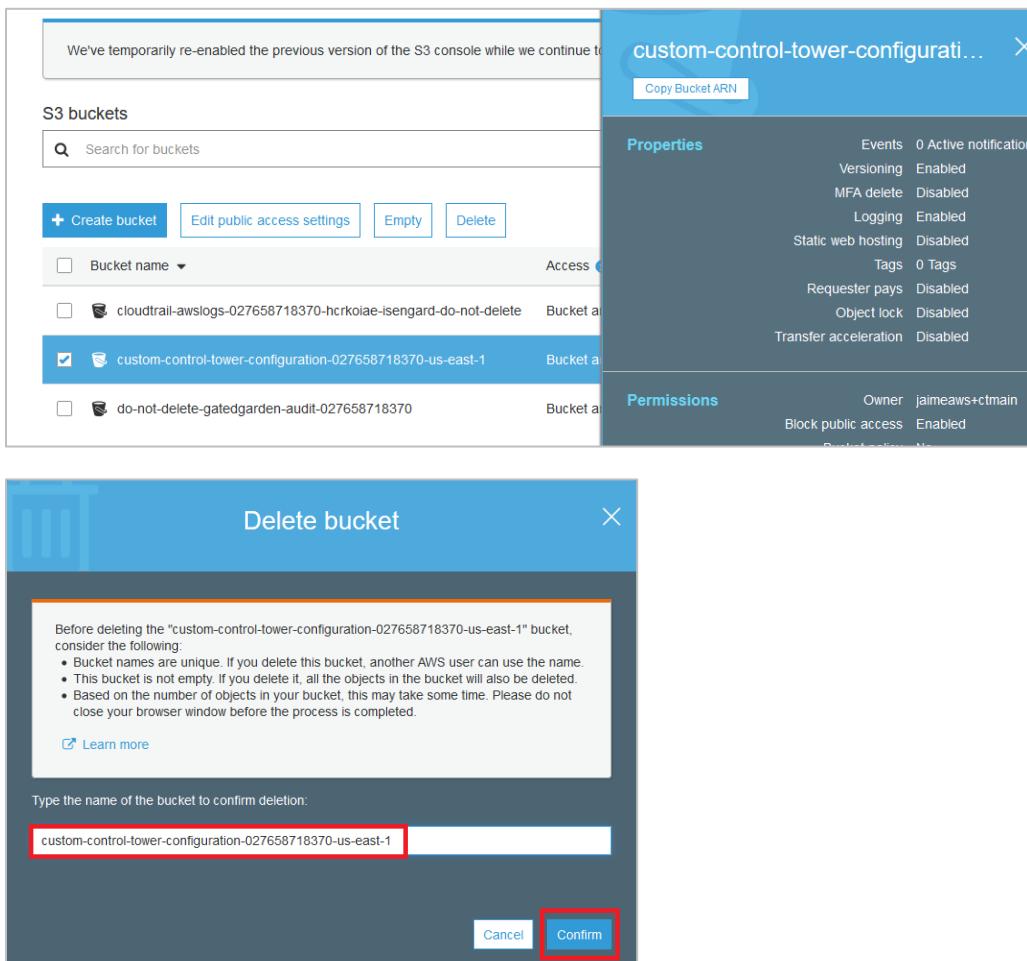
Build history | Batch history

Build history | Stop build | View artifacts | View logs | Delete builds | Retry build

Custom-Control-Tower-StackSet-CodeBuild: 5d4c329-1c07-4d29-b577-03236657c42e

Custom-Control-Tower-SCP-CodeBuild: 5e3c8256-6e67-4695-8e94-0d48f92950df

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



The screenshot shows the AWS S3 console interface. On the left, there is a message: "We've temporarily re-enabled the previous version of the S3 console while we continue to work on the new one." Below this is a search bar labeled "Search for buckets". There are four buttons: "+ Create bucket", "Edit public access settings", "Empty", and "Delete". A dropdown menu for "Bucket name" is open, showing three options: "cloudtrail-awslogs-027658718370-hcrkoiae-isengard-do-not-delete", "custom-control-tower-configuration-027658718370-us-east-1" (which is selected), and "do-not-delete-gatedgarden-audit-027658718370". To the right, a modal window titled "custom-control-tower-configuration-027658718370-us-east-1" displays bucket properties and permissions.

Properties	
Events	0 Active notification
Versioning	Enabled
MFA delete	Disabled
Logging	Enabled
Static web hosting	Disabled
Tags	0 Tags
Requester pays	Disabled
Object lock	Disabled
Transfer acceleration	Disabled

Permissions	
Owner	jaimeaws+ctmain
Block public access	Enabled

In the center, a "Delete bucket" dialog box is open. It contains a warning message: "Before deleting the 'custom-control-tower-configuration-027658718370-us-east-1' bucket, consider the following:

- Bucket names are unique. If you delete this bucket, another AWS user can use the name.
- This bucket is not empty. If you delete it, all the objects in the bucket will also be deleted.
- Based on the number of objects in your bucket, this may take some time. Please do not close your browser window before the process is completed.

". Below this is a link "Learn more". A text input field contains the bucket name "custom-control-tower-configuration-027658718370-us-east-1". At the bottom are two buttons: "Cancel" and "Confirm", with "Confirm" being highlighted with a red box.

Decommission and clean up Lab 2

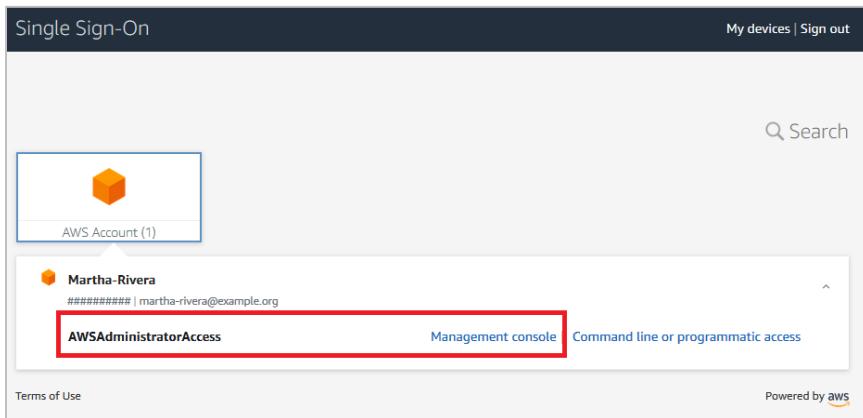
In Lab 2, you created AWS Service Catalog portfolios and launch constraints. You also used an AWS CloudFormation stack to provision IAM roles, user groups, and users to implement the constraints.

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.

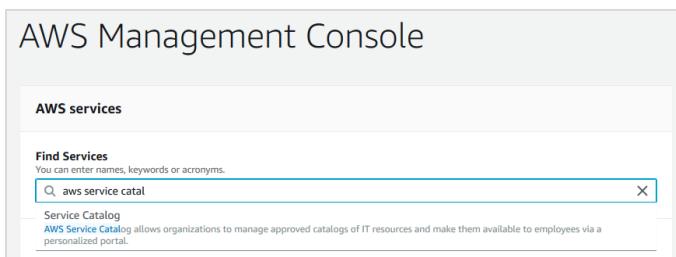
Clean up Amazon EC2 and Amazon S3 resources

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.

Portfolio-For-Sandboxes [Info](#)

Portfolio to allow approved products in a specific Organizational Unit

Portfolio details

Id port-vuvbdczg463po	Created time Mon, Aug 24, 2020, 1:56:08 PM EDT	ARN arn:aws:catalog:us-east-1
Owner AWS		

[Products \(2\)](#) | [Constraints \(2\)](#) | [Groups, roles, and users \(1\)](#) | [Share \(0\)](#) | [TagOptions \(0\)](#)

Groups, roles, and users (1/1)

Name	Type	ARN
SCEndUserGroup	IAM	arn:aws:iam::167412180053:group/SCEndUserGroup

[Remove group, role, or user](#) | [Add groups, roles, users](#)

10. Remove the portfolio.

[Service Catalog](#) > [Portfolios](#)

Portfolios [Info](#)

[Local](#) | [Imported](#) | [Getting Started library](#)

Imported portfolios (1/2)

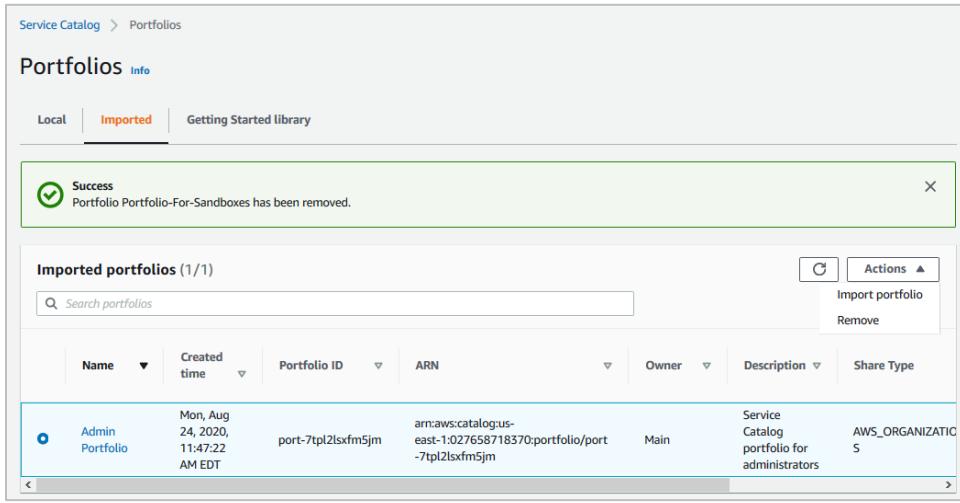
Name	Created time	Portfolio ID	ARN	Owner	Description	Share Type
Admin Portfolio	Mon, Aug 24, 2020, 11:47:22 AM EDT	port-7tpl2lsxfm5jm	arn:aws:catalog:us-east-1:027658718370:portfolio/port-7tpl2lsxfm5jm	Main	Service Catalog portfolio for administrators	AWS_ORGANIZATIONAL
Portfolio-For-Sandboxes	Mon, Aug 24, 2020, 1:56:08 PM EDT	port-vuvbdczg463po	arn:aws:catalog:us-east-1:027658718370:portfolio/port-vuvbdczg463po	AWS	Portfolio to allow approved products in a specific Organizational Unit	AWS_ORGANIZATIONAL

[Actions](#) | [Import portfolio](#) | [Remove](#)

Warning

Are you sure you want to remove portfolio Portfolio-For-Sandboxes?

[Cancel](#) | [Continue](#)



The screenshot shows the AWS Service Catalog interface under the 'Portfolios' section. The 'Imported' tab is selected. A success message box displays: 'Success: Portfolio Portfolio-For-Sandboxes has been removed.' Below this, a table lists 'Imported portfolios (1/1)'. The single entry is 'Admin Portfolio', which was created on Mon, Aug 24, 2020, 11:47:22 AM EDT. It has an ARN of arn:aws:catalouge-east-1:027658718370:portfolio/port-7tpl2lsxfm5jm, is owned by Main, and is a Service Catalog portfolio for administrators, shared with AWS_ORGANIZATIONS.

Delete roles, users, and groups created by AWS CloudFormation

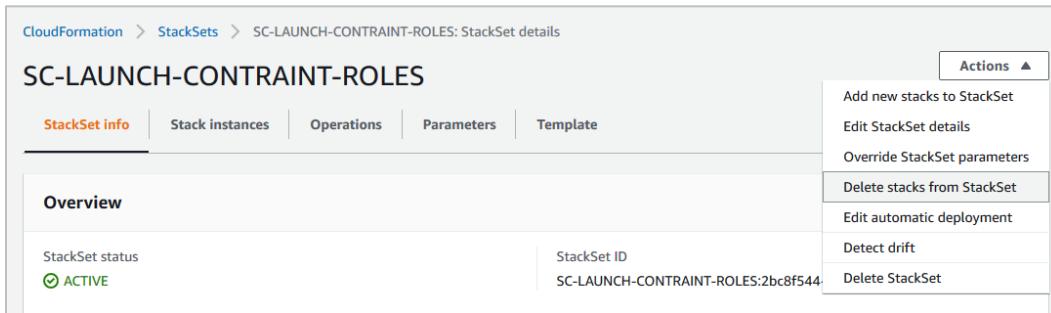
The final decommission task for Lab 3 is to delete the roles, users, and groups created by AWS CloudFormation:

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-CONSTRAINT-ROLES StackSet**, choose **Actions**, and then choose **Delete stacks from StackSet**.



The screenshot shows the AWS CloudFormation StackSets interface. The 'SC-LAUNCH-CONSTRAINT-ROLES' StackSet is selected. In the 'Actions' dropdown menu, the option 'Delete stacks from StackSet' is highlighted. The 'StackSet info' tab is active, showing the 'Overview' section. The 'StackSet status' is listed as 'ACTIVE' and the 'StackSet ID' is 'SC-LAUNCH-CONSTRAINT-ROLES:2bc8f544'.

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

Note

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

SC-LAUNCH-CONSTRAINT-ROLES: Delete stacks from StackSet

Review

Step 1: Set deployment options

Deployment configuration

Automatic deployment <input checked="" type="checkbox"/> Enabled	Retain stacks on account removal
	<input type="checkbox"/> Delete stacks
Deployment targets	
<input type="checkbox"/> ou-your-id-for-Custom	

Regions

<input type="text" value="Q"/>	<	1	>
Region			
us-east-1			

Deployment options

Maximum concurrent accounts 1	Failure tolerance 0
Retain stacks No	

[Cancel](#) [Previous](#) [Submit](#)

- Delete the StackSet.

CloudFormation > StackSets > SC-LAUNCH-CONSTRAINT-ROLES: StackSet details

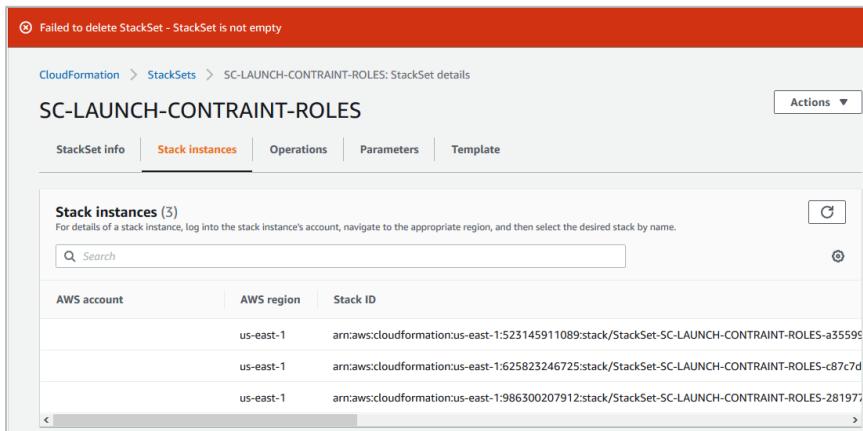
SC-LAUNCH-CONSTRAINT-ROLES

[StackSet info](#) | [Stack instances](#) | [Operations](#) | [Parameters](#) | [Template](#)

Overview

StackSet status <input checked="" type="checkbox"/> ACTIVE	StackSet ID SC-LAUNCH-CONSTRAINT-ROLES:2bc8f544	Actions ▾
		Delete StackSet

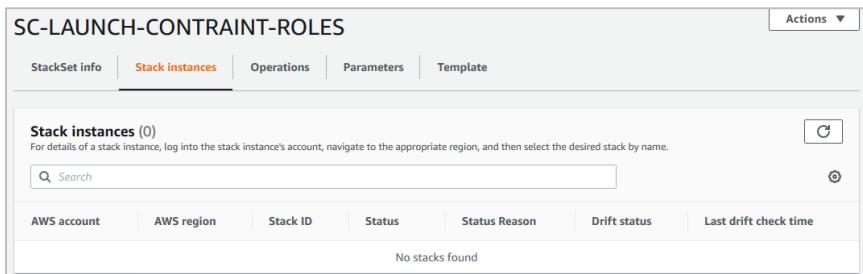
- You might get an error like the one shown.



The screenshot shows the AWS CloudFormation StackSets console. At the top, a red banner displays the error message: "Failed to delete StackSet - StackSet is not empty". Below the banner, the navigation path is "CloudFormation > StackSets > SC-LAUNCH-CONSTRAINT-ROLES: StackSet details". The main title is "SC-LAUNCH-CONSTRAINT-ROLES". The "Stack instances" tab is selected, showing a table with three entries. The table columns are "AWS account", "AWS region", and "Stack ID". The data rows are:

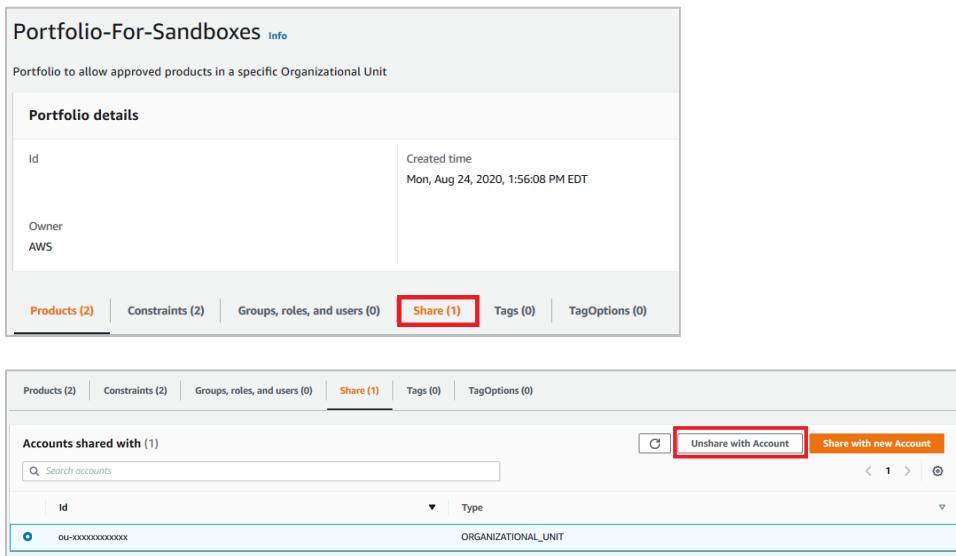
AWS account	AWS region	Stack ID
us-east-1		arn:aws:cloudformation:us-east-1:523145911089:stack/StackSet-SC-LAUNCH-CONSTRAINT-ROLES-a35595
us-east-1		arn:aws:cloudformation:us-east-1:625823246725:stack/StackSet-SC-LAUNCH-CONSTRAINT-ROLES-c87c7d
us-east-1		arn:aws:cloudformation:us-east-1:986300207912:stack/StackSet-SC-LAUNCH-CONSTRAINT-ROLES-281977

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:



The screenshot shows the AWS CloudFormation StackSets console. The title is "SC-LAUNCH-CONSTRAINT-ROLES". The "Stack instances" tab is selected, showing a table with zero entries. The table columns are "AWS account", "AWS region", "Stack ID", "Status", "Status Reason", "Drift status", and "Last drift check time". A message at the bottom of the table area says "No stacks found".

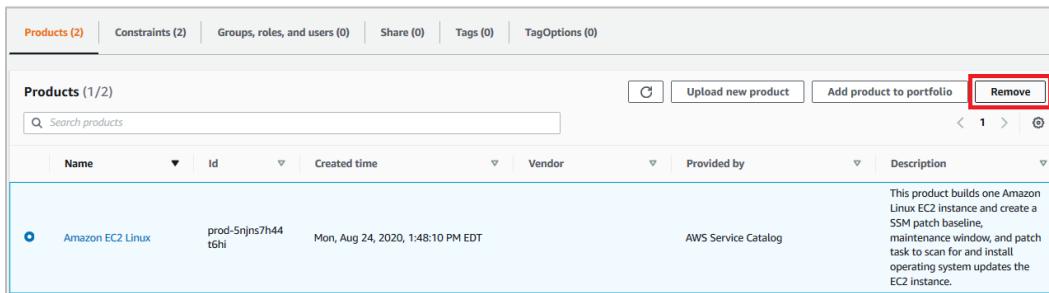
9. Delete the **SC-LaunchConstraintRole Stack**.
10. Delete the portfolios that were created as part of this lab. You will need to unshare them and remove any constraints or user, group, or role access associated to them



The first screenshot shows the "Portfolio-For-Sandboxes" page. It displays "Products (2)", "Constraints (2)", "Groups, roles, and users (0)", a red box around "Share (1)", "Tags (0)", and "TagOptions (0)".

The second screenshot shows the "Accounts shared with (1)" section. It lists "ou-xxxxxxxxxx" under "Id" and "ORGANIZATIONAL_UNIT" under "Type". It includes "Search accounts" and "Unshare with Account" and "Share with new Account" buttons.

11. Delete the products in the portfolio.



Name	Id	Created time	Vendor	Description
Amazon EC2 Linux	prod-5njns7h44t6hi	Mon, Aug 24, 2020, 1:48:10 PM EDT	AWS Service Catalog	This product builds one Amazon Linux EC2 instance and create a SSM patch baseline, maintenance window, and patch task to scan for and install operating system updates the EC2 instance.

12. Delete the portfolio.

13. Do this on your member account (Martha's account) and on your management account.

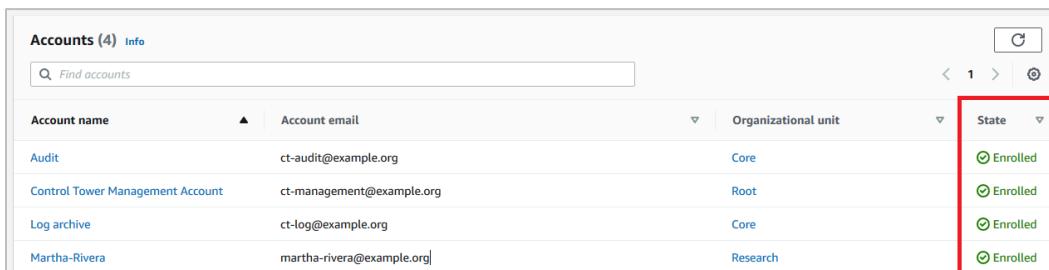
Decommission and clean up Lab 1 and Lab 0

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

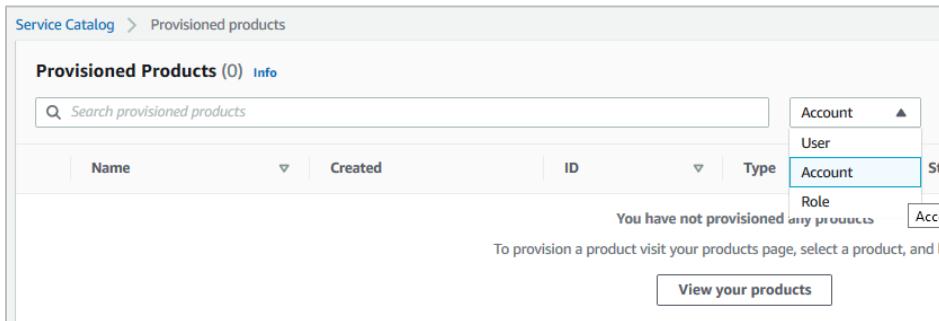
The detailed steps to unmanage and delete accounts enrolled with AWS Control Tower are available here in the documentation portal. The logical flow is:

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.



Account name	Account email	Organizational unit	State
Audit	ct-audit@example.org	Core	Enrolled
Control Tower Management Account	ct-management@example.org	Root	Enrolled
Log archive	ct-log@example.org	Core	Enrolled
Martha-Rivera	martha-rivera@example.org	Research	Enrolled

- 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.



The screenshot shows the AWS Service Catalog interface under 'Provisioned products'. At the top, there's a search bar labeled 'Search provisioned products'. Below it is a table header with columns: Name, Created, ID, Type, Status, and Actions. The 'Type' column is currently set to 'Account'. A message in the center of the page states 'You have not provisioned any products' and provides a link 'To provision a product visit your products page, select a product, and la...'. At the bottom right is a button labeled 'View your products'.

- Remove the accounts from your organization.
- For more information about closing an account, see <https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/close-account.html#closing-member-account>.
- Remove the accounts from the organization prior to their deletion.
- Set up a password for the accounts and a payment method to remove them from the organization so you can access them separately.
- 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.
- Use your management account to go to AWS Organizations, and remove all member accounts.



The screenshot shows the AWS Organizations console. The top navigation bar includes 'Accounts', 'Organize accounts', and 'Policies'. Below the navigation is a table with columns: 'Account name', 'Email', 'Account ID', and 'Status'. The table lists four accounts: 'Control Tower Management Account' (selected), 'Martha-Rivera' (status: 'Created on 8/31/20'), 'Audit' (status: 'Created on 8/24/20'), and 'Log archive' (selected, status: 'Created on 8/24/20').

- 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 Organizations			
Accounts	Organize accounts	Policies	
		Add account	Remove account
<input type="checkbox"/>	Account name	Email	Account ID
<input type="checkbox"/>	★ Control Tower Management Account	ct-management@example.org	Joined on 8/24/20
<input type="checkbox"/>	Audit	ct-audit@example.org	Created on 8/24/20
<input checked="" type="checkbox"/>	Log archive	ct-log@example.org	Created on 8/24/20

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/controlltower/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 Policies](#)

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/controlltower/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](#).