



COS20019

Cloud Computing Architecture

Week 8 – ACA Module 11:
Guided Lab: Automating Infrastructure with AWS CloudFormation

Truong Ngoc Gia Hieu
105565520

Guided Lab: Automating Infrastructure Deployment with AWS CloudFormation

A. Lab Overview and objectives

Deploying infrastructure in a consistent, reliable manner is difficult. It requires people to follow documented procedures without taking any undocumented shortcuts. It can also be difficult to deploy infrastructure after hours when fewer staff are available. AWS CloudFormation changes this situation by defining infrastructure in a template that can be automatically deployed—even on an automated schedule.

In this lab, you learn how to deploy multiple layers of infrastructure with AWS CloudFormation, update a CloudFormation stack, and delete a stack (while retaining some resources).

After completing this lab, you should be able to do the following:

- Use AWS CloudFormation to deploy a virtual private cloud (VPC) networking layer.
- Use AWS CloudFormation to deploy an application layer that references the networking layer.
- Explore templates with AWS CloudFormation Designer.
- Delete a stack that has a deletion policy.

Duration

This lab will require approximately **20 minutes** to complete.

AWS service restrictions

In this lab environment, access to AWS services and service actions might be restricted to the ones that are needed to complete the lab instructions. You might encounter errors if you attempt to access other services or perform actions beyond the ones that are described in this lab.

B. Accessing the AWS Management Console

To access the AWS Management Console, click on the button **Start Lab** and wait the circle next to the AWS text **turns from yellow to green**



Figure 1: AWS Management activated

C. Task 1: Deploying a networking layer

It's a best practice to deploy infrastructure in *layers*. Common layers include the following:

- Network (Amazon Virtual Private Cloud)
- Database
- Application

This way, templates can be reused between systems. For example, you can deploy a common network topology between development, test, and production environments, or deploy a standard database for multiple applications.

In this task, you deploy an AWS CloudFormation template that creates a networking layer by using Amazon VPC.

Step 1.1: Download the material named [lab-network.yaml](#)

Step 1.2: Search and select **CloudFormation** service. Then, Choose **Create stack > With new resources (standard)**

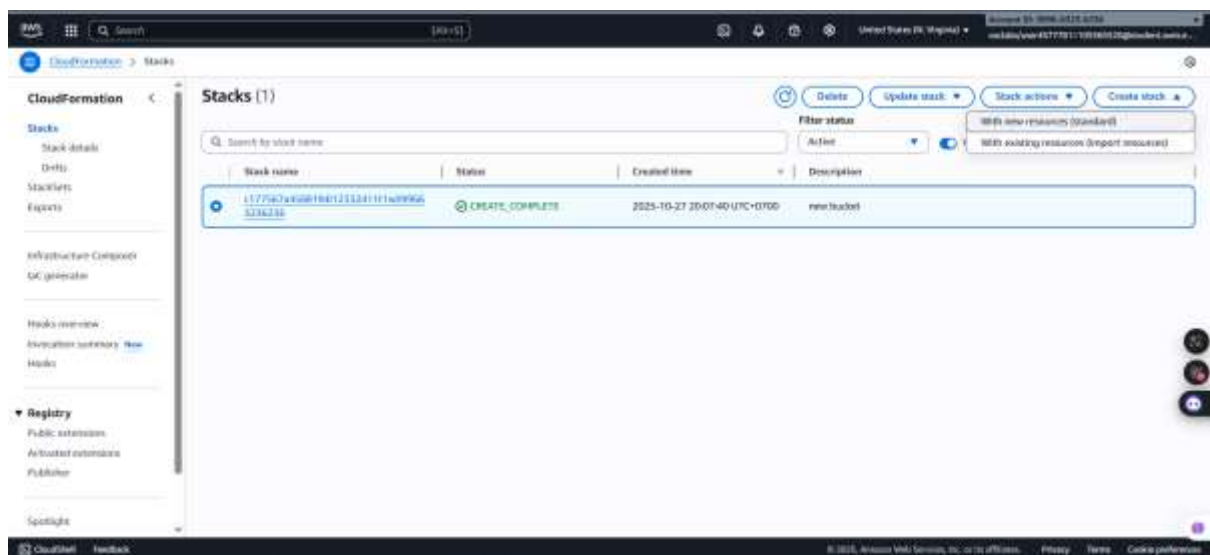


Figure 2: CloudFormation homepage

Step 1.3: Configure following settings in **Step 1: Create stack**

- **Prepare template:** Choose Template is ready.
- **Template source:** Choose Upload a template file > Choose file, and then choose the lab-network.yaml file that you downloaded.
- Choose **Next**.

Prerequisite - Prepare template
You can also create a template by scanning your existing resources in the [IaC generator](#).

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Choose an existing template
Upload or choose an existing template.

☐ Build from Infrastructure Composer
Create a template using a visual builder.

Specify template [info](#)
This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

Template source
Selecting a template generates an Amazon S3 URL, where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

☐ Amazon S3 URL
Provide an Amazon S3 URL to your template.

☒ Upload a template file
Upload your template directly to the console.

☐ Sync from Git
Sync a template from your Git repository.

Upload a template file
[Choose file](#)

lab-network.yaml

JSON or YAML formatted file

S3 URL: <https://s3-us-east-1.amazonaws.com/uf-templates-3p5gut7j5267-us-east-1/2025-10-27T13:15:15.755Z9wo-lab-network.yaml>

[View in Infrastructure Composer](#)

[Cancel](#) [Next](#)

Figure 3: Create stack confirmation

Step 1.4: Configure Step 2: Specify stack details

- Stack name: lab-network
- Choose **Next**.

Specify stack details

Provide a stack name

Stack name

lab-network

Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 11/128.

Figure 4: Specify task details confirmation

Step 1.5: Configure Step 3: Configure stack options

- In the **Tags** section, choose **Add new tag** and configure the following:
 - Key:** application
 - Value:** inventory
- Choose **Next**.

Configure stack options

Tags - optional
Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organizing, identifying, and categorizing those resources. You can add up to 50 unique tags for each stack.

Key	Value - Tags - optional	
application	inventory	Remove

[Add new tag](#)

You can add 45 more tag(s)

Figure 5: Tag confirmation

Step 1.6: Review and click **Submit button**

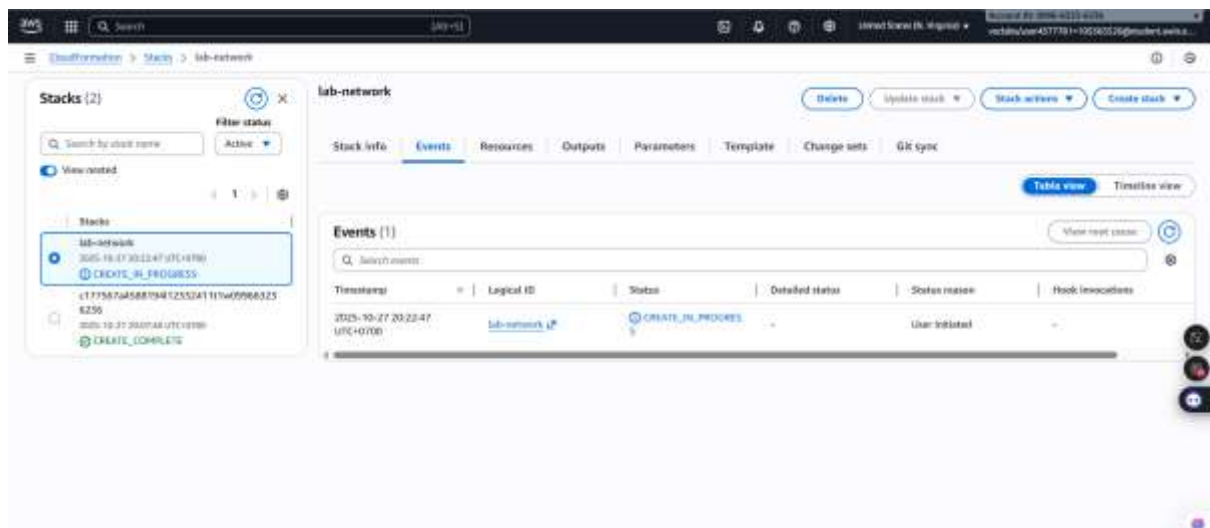


Figure 6: Submit successfully

Step 1.7: Choose the **Stack info** tab and check the Status to change to **CREATE_COMPLETE**.

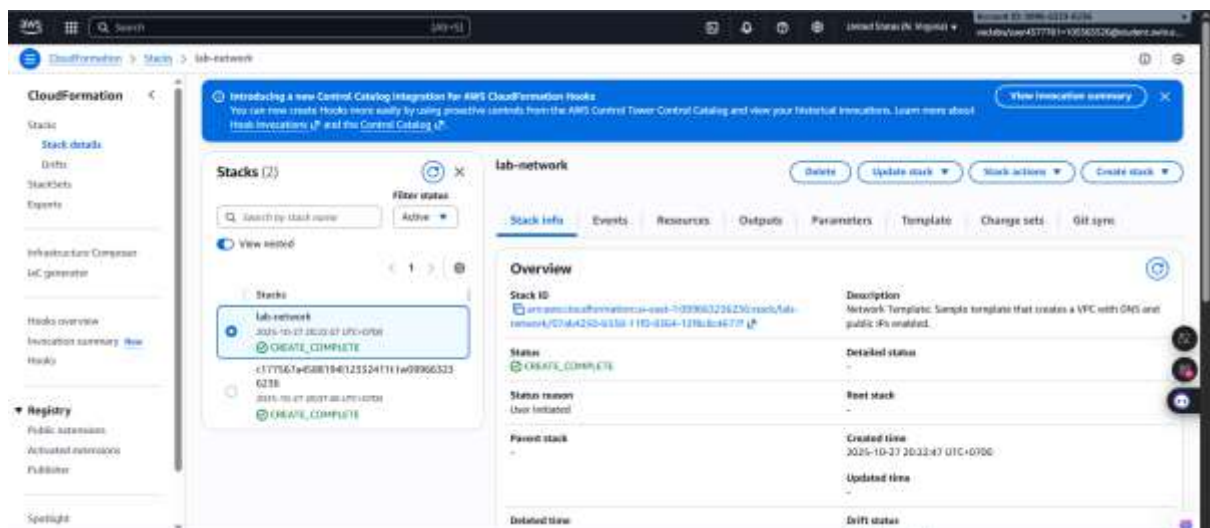


Figure 7: Stack info tab

Step 1.8: Choose the **Resources** tab.

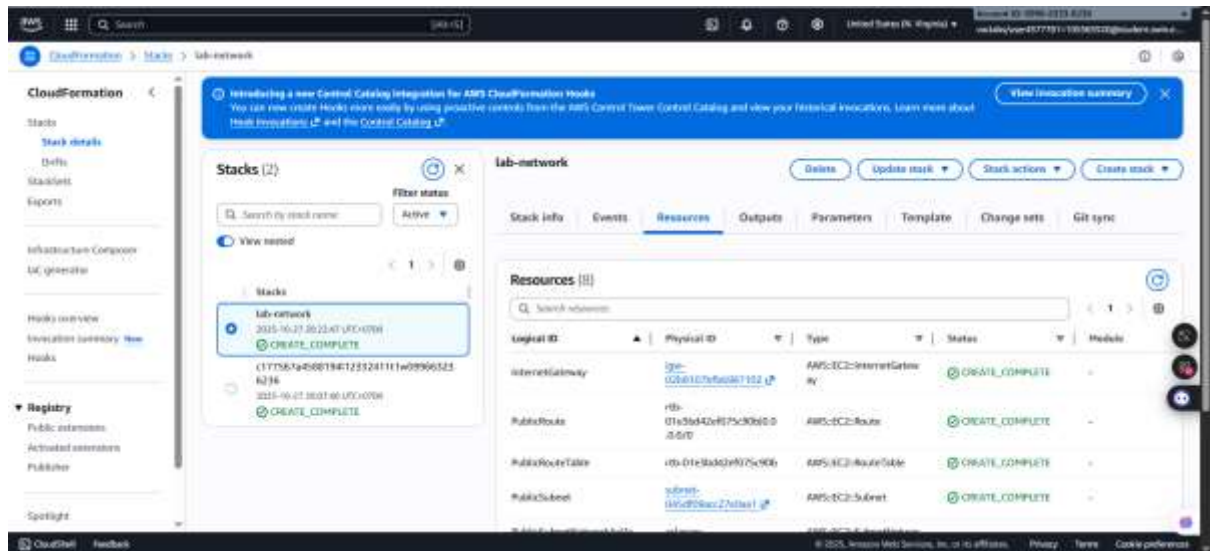


Figure 8: Resources tab

Step 1.9: Choose the **Outputs** tab.

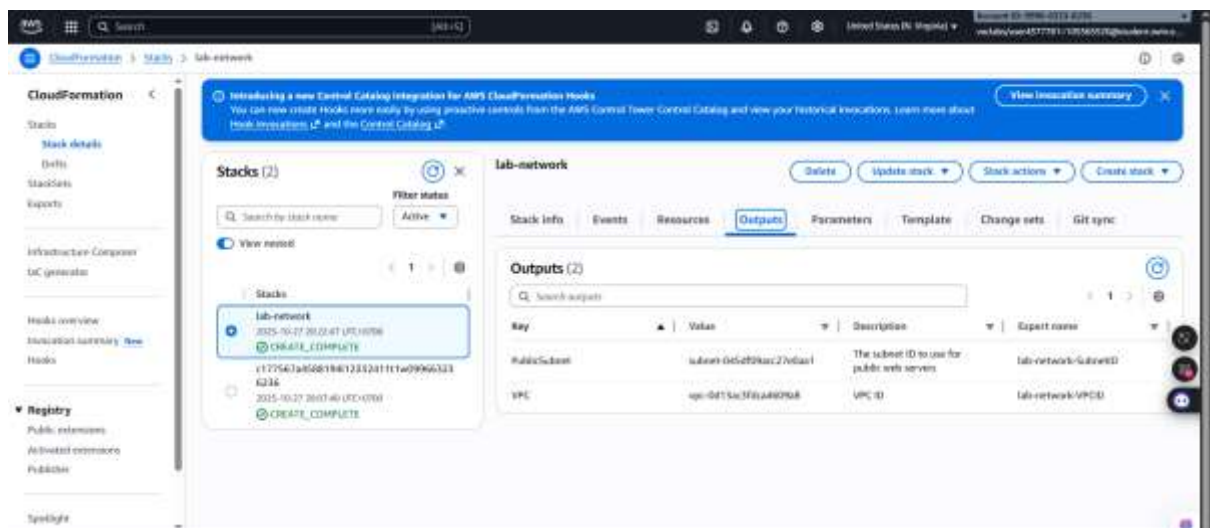


Figure 9: Outputs tab

Step 1.10: Choose the **Template** tab.

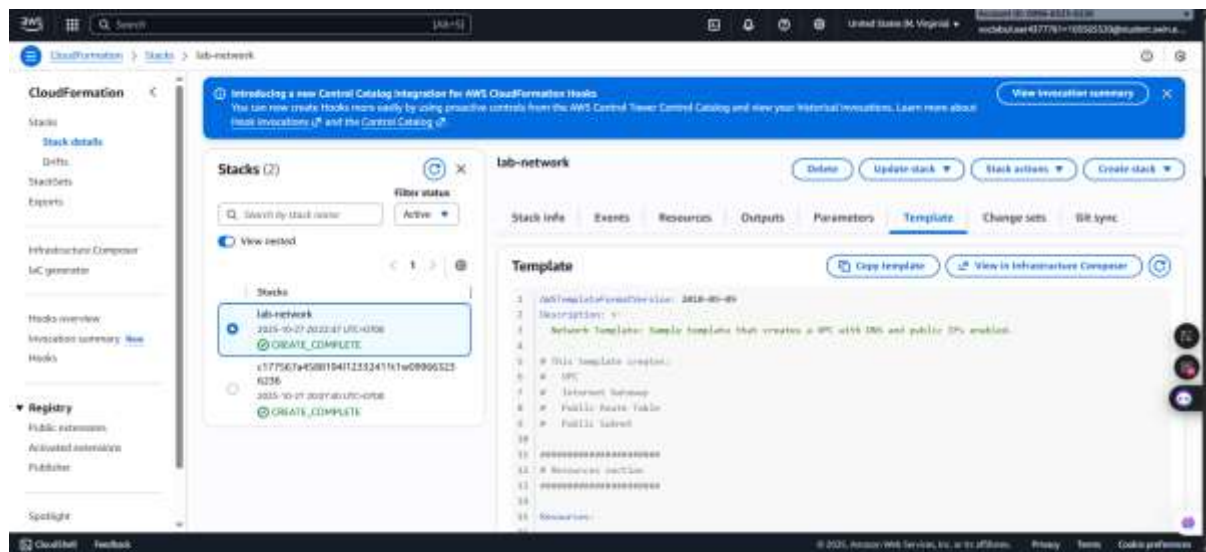


Figure 10: Template tab

D. Task 2: Deploying an application layer

Now that you deployed the *network layer*, you will deploy an *application layer* that contains an Amazon Elastic Compute Cloud (Amazon EC2) instance and a security group. The AWS CloudFormation template *imports* the VPC and subnet IDs from the *outputs* of the existing CloudFormation stack. Then, it uses this information to create the security group in the VPC and the EC2 instance in the subnet.

Step 2.1: Download the material [lab-application.yaml](#)

Step 2.2: In the CloudFormation homepage, choose **Stacks**, select **Create stack** button, and choose **Create stack > With new resources (standard)**

Step 2.3: In **Step 1: Create Stack**, configure:

- **Prepare template:** Choose **Template is ready**.
- **Template source:** Choose **Upload a template file > Choose file**, and then choose the **lab-application.yaml** file that you downloaded.
- Choose **Next**.

Create stack

Prerequisite - Prepare template
You can also create a template by scanning your existing resources in the [IAC generator](#).

Prepare template
Every stack is based on a template, a template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Choose an existing template
Upload or choose an existing template.

☐ Build from Infrastructure Composer
Create a template using a visual builder.

Specify template [info](#)
This [GitHub repository](#) contains sample CloudFormation templates that can help you get started on new infrastructure projects. [Learn more](#)

Template source
Selecting a template generates an Amazon S3 URL, where it will be stored. A template is a JSON or YAML file that describes your stack's resources and properties.

☐ Amazon S3 URL
Provide an Amazon S3 URL to your template.

☒ Upload a template file
Upload your template directly to the console.

☐ Sync from Git
Sync a template from your Git repository.

Upload a template file
[Choose file](#)

lab-application.yaml

JSON or YAML formatted file

Figure 11: Create stack configuration

Step 2.4: Configure following settings in Step 2: Specify stack details

- **Stack name:** lab-application
- Notice the NetworkStackName: lab-network
- Choose **Next**.

Specify stack details

Provide a stack name
Stack name

lab-application

Stack name must contain only letters (a-z, A-Z), numbers (0-9), and hyphens (-) and start with a letter. Max 128 characters. Character count: 15/128.

Parameters
Parameters are defined in your template and allow you to input custom values when you create or update a stack.

AmazonLinuxAMIID
/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2

NetworkStackName
Name of an active CloudFormation stack that contains the networking resources, such as the VPC and subnet that will be used in this stack.

lab-network

Figure 12: Specify task configuration

Step 2.5: Step 3: Configure stack options

- In the **Tags** section, choose **Add new tag** and configure the following:
 - **Key:** application
 - **Value:** inventory
- Choose **Next**.

Configure stack options

Tags - optional
Tags (key-value pairs) are used to apply metadata to AWS resources, which can help in organizing, identifying, and categorizing those resources. You can add up to 50 unique tags for each stack.

Key

Value - Tags - optional

application

inventory

[Add new tag](#)

You can add 49 more tags

Figure 13: Tags configuration

Step 2.6: Check all configurations are correct and then press **Submit button**

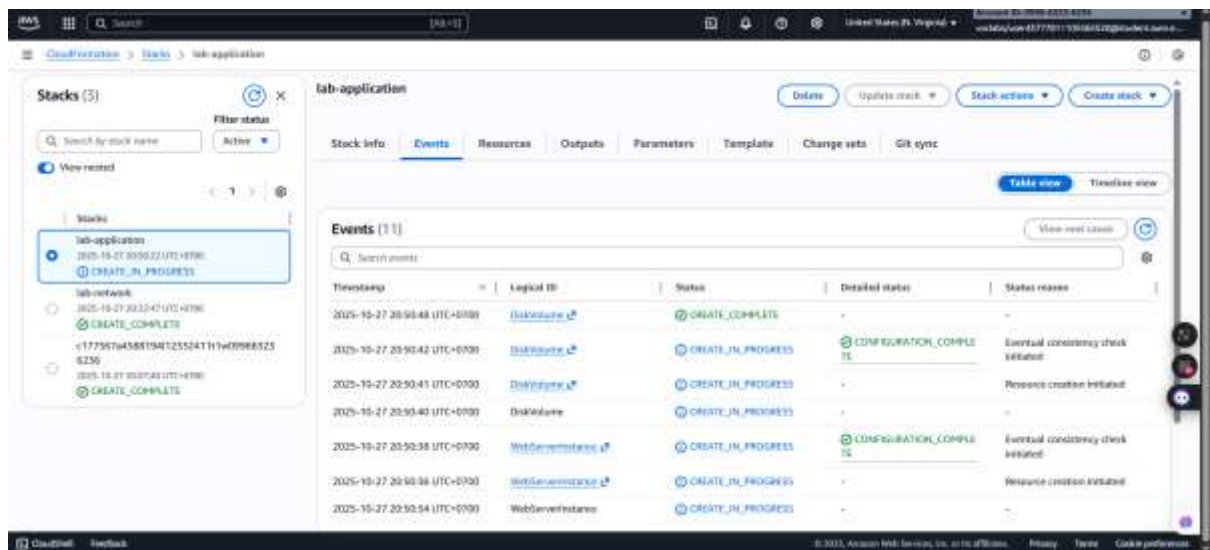


Figure 14: Submit successfully

Step 2.7: In the Stack info tab, wait for the **Status** to change to **CREATE_COMPLETE**.

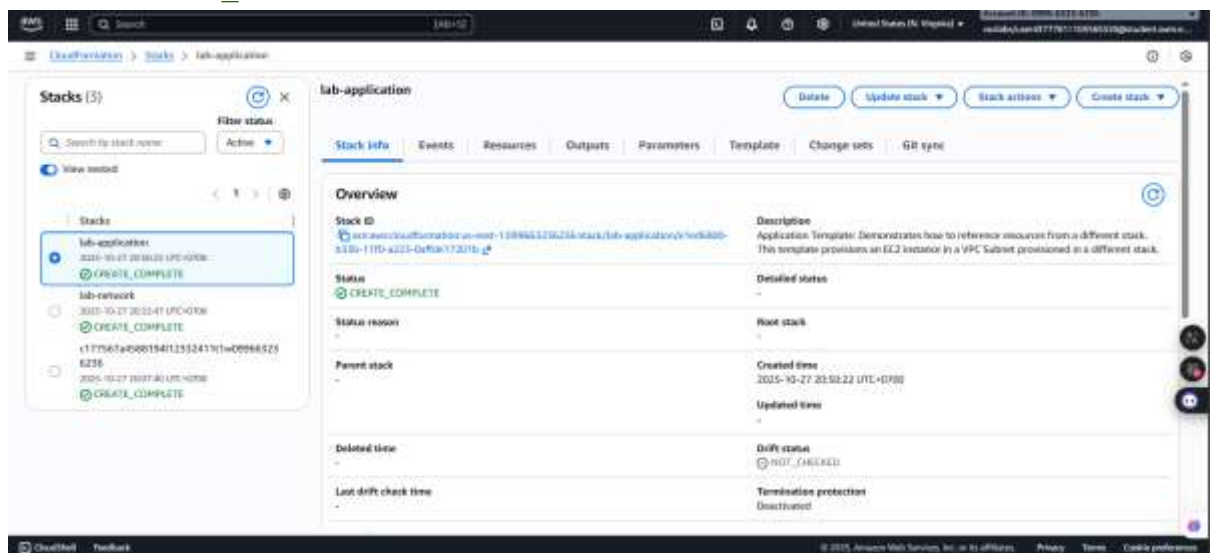


Figure 15: Status checks

Step 2.8: Choose the **Outputs** tab.

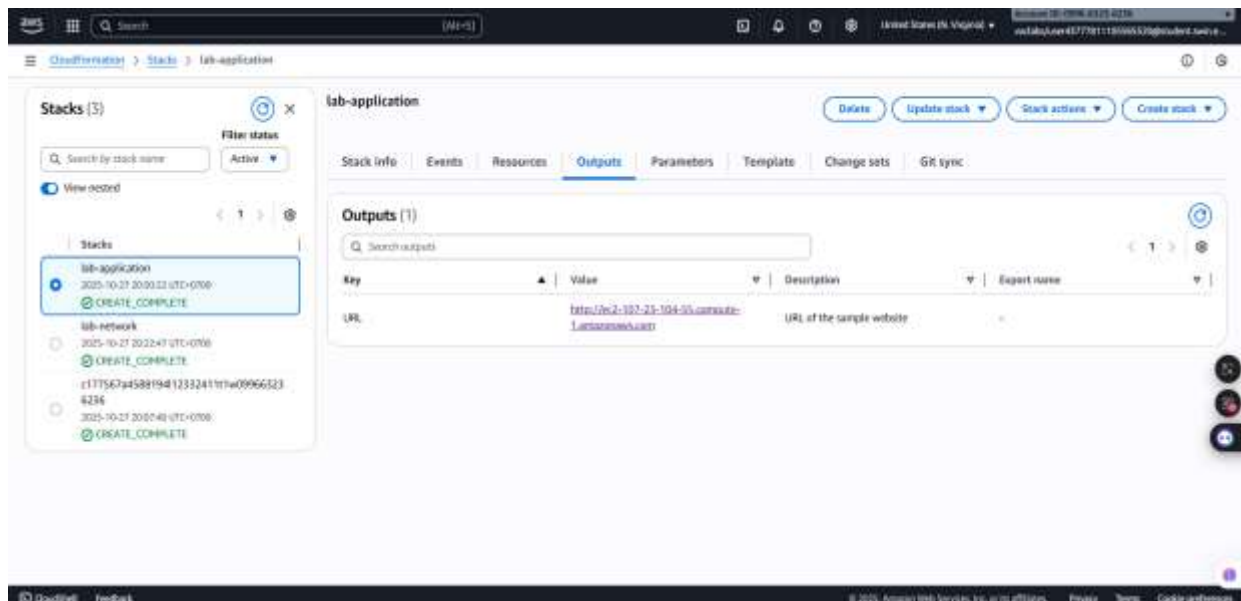


Figure 16: Outputs tab

Step 2.9: Copy the **URL** that is displayed, open a new web browser tab, paste the URL, and press ENTER



Congratulations, you have successfully launched the AWS CloudFormation sample.

Figure 17: Link result

E. Task 3: Updating a Stack

AWS CloudFormation can also *update* a stack that was deployed. When you update a stack, AWS CloudFormation will only modify or replace the resources that are being changed. Any resources that are not being changed will be left as-is.

In this task, you update the lab-application stack to modify a setting in the security group.

Step 3.1: At the top of the AWS Management Console, in the search box, search for and choose **EC2**. In the left navigation pane, in the Network & Security section, choose **Security Groups**.

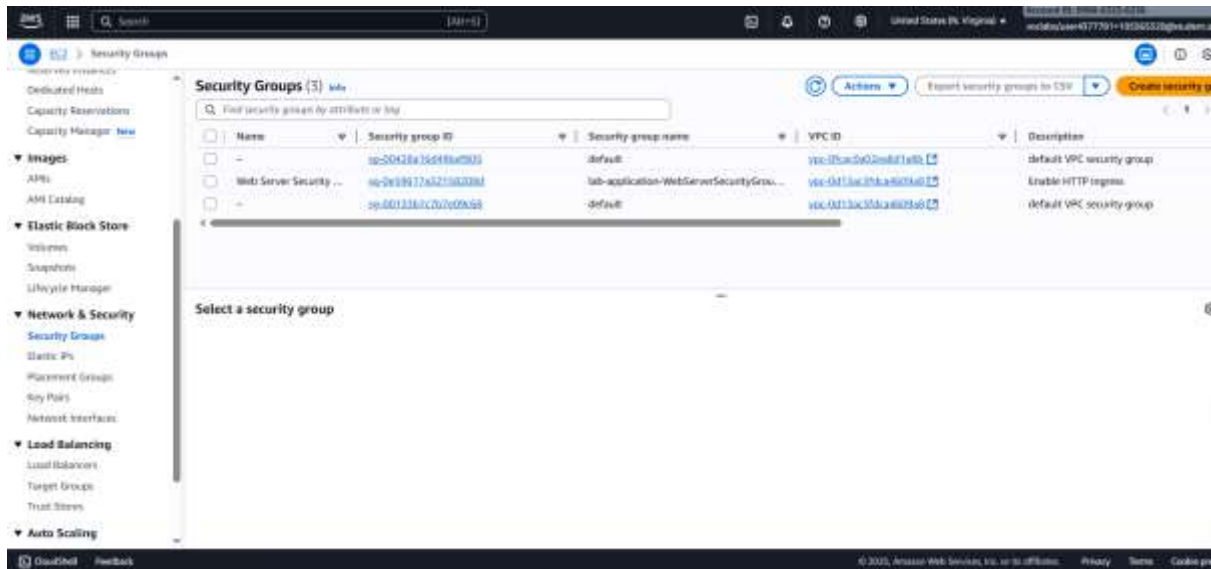


Figure 18: Security Groups page

Step 3.2: Select the check box for **lab-application-WebServerSecurityGroup**. Next, Choose the **Inbound rules** tab.

Currently, only one rule is in the security group. The rule permits *HTTP* traffic.

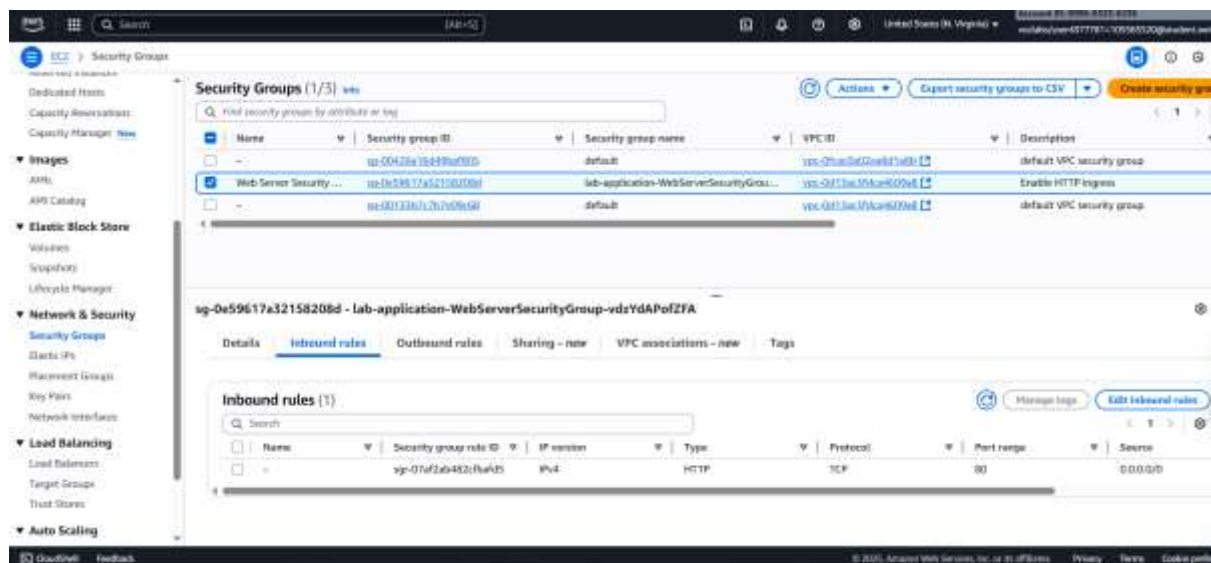


Figure 20: Inbound rules tab

Step 3.3: From the **Services** menu at the top, choose **CloudFormation**. From the **Stacks** list of the AWS CloudFormation console, choose **lab-application**.

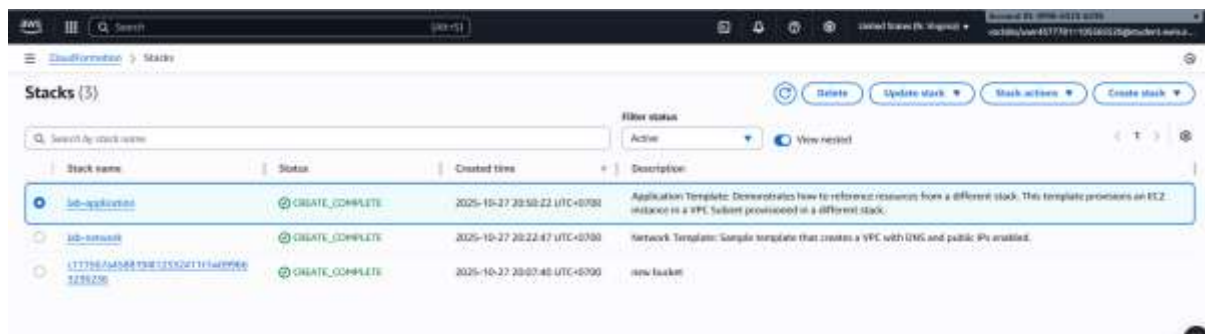


Figure 21: lab-application selected

Step 3.4: Choose **Update stack** => **Create change set** and configure:

- Prepare template: Choose Replace current template.
- Template source: Choose Upload a template file.
- Upload a template file: **Choose file**, and then choose the **lab-application2.yaml** file that you downloaded.

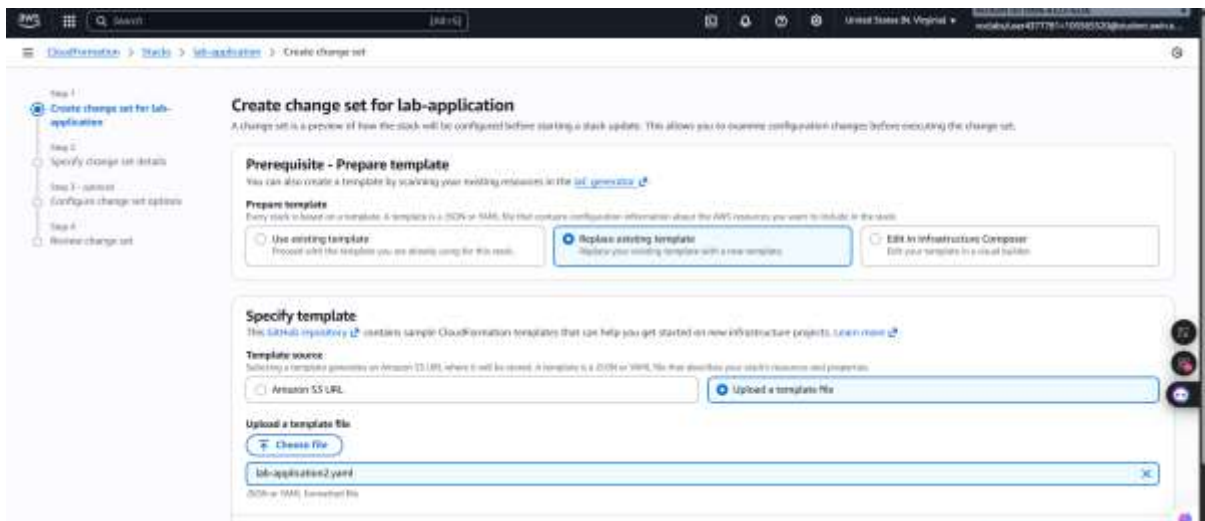


Figure 22: Step 1 configuration

Step 3.5: Choose **Next** on each of the next three screens to go to the Review lab-application page. In the next step, Choose **Submit**.

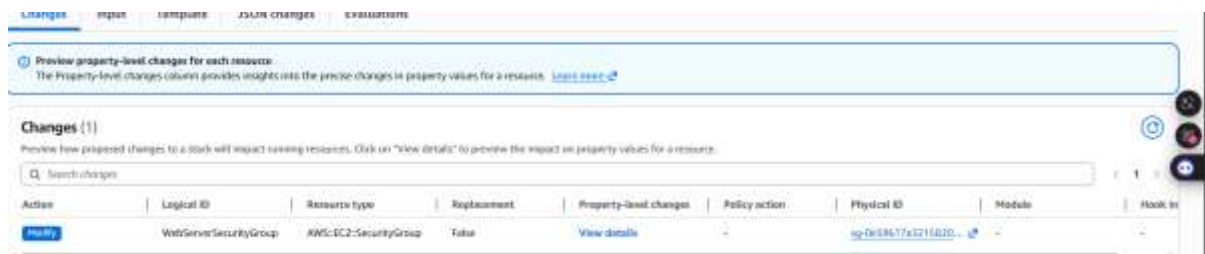


Figure 23: Changed successfully

Step 3.6: In the **Stack info** tab, wait for the Status to change to **UPDATE_COMPLETE**.

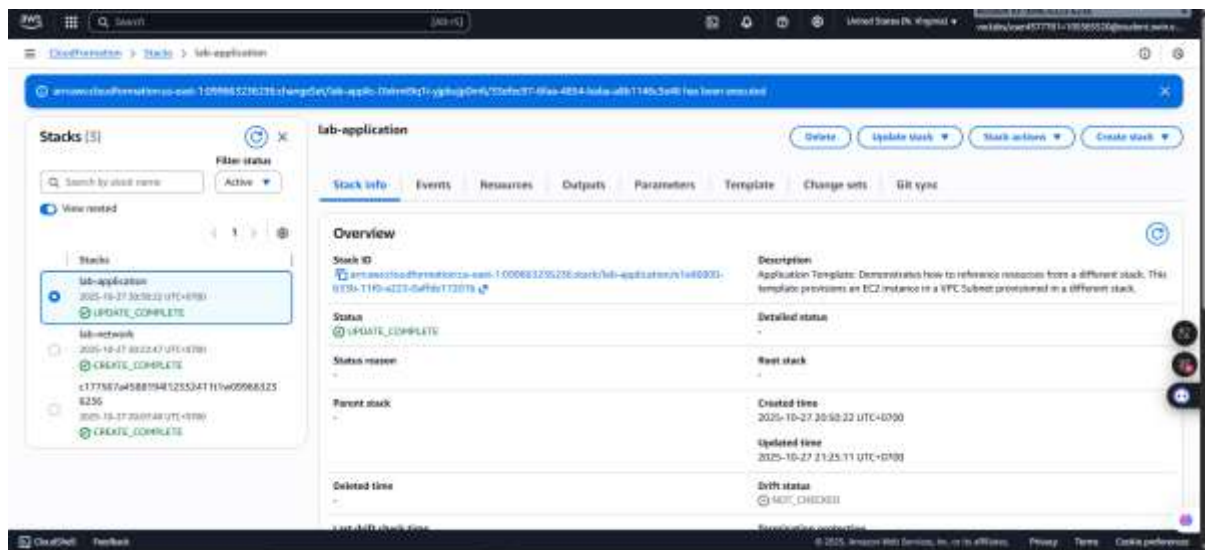


Figure 24: Status check

Step 3.7: Return to the EC2 and choose **Security Groups** to check the inbound rules of the **lab-application-WebServerSecurityGroup**.

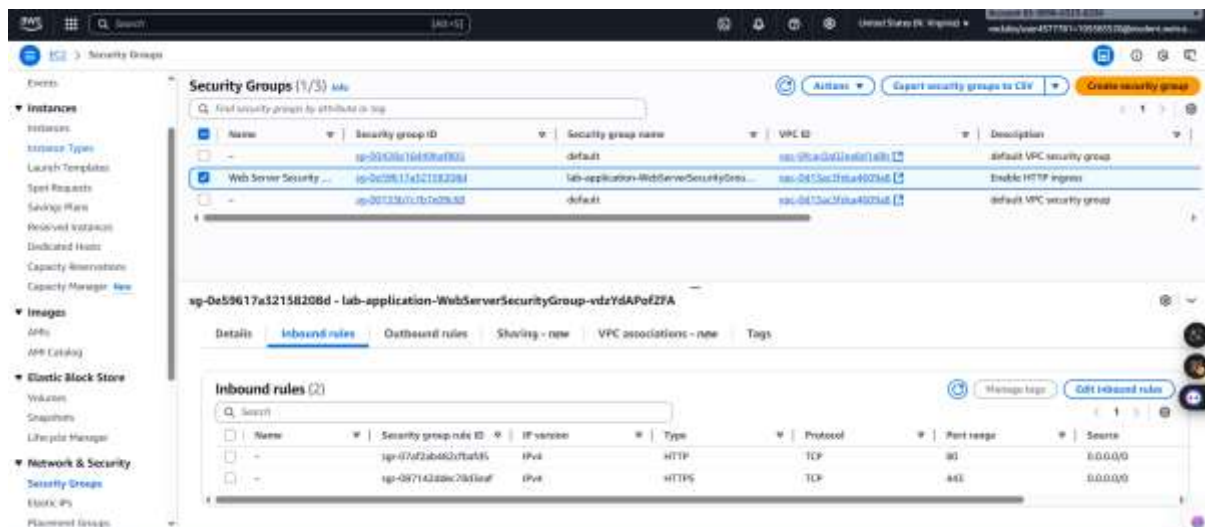


Figure 25: Inbound rules check

F. Task 4: Exploring templates with AWS CloudFormation Designer

AWS CloudFormation Designer (Designer) is a graphic tool for creating, viewing, and modifying AWS CloudFormation templates. With Designer, you can diagram your template resources by using a drag-and-drop interface, and then edit their details through the integrated JSON and YAML editor.

Whether you are a new to AWS CloudFormation or an experienced AWS CloudFormation user, Designer can help you quickly see the interrelationship between a template's resources. It also enables you to easily modify templates

Step 4.1: From the Services menu, choose **CloudFormation**.

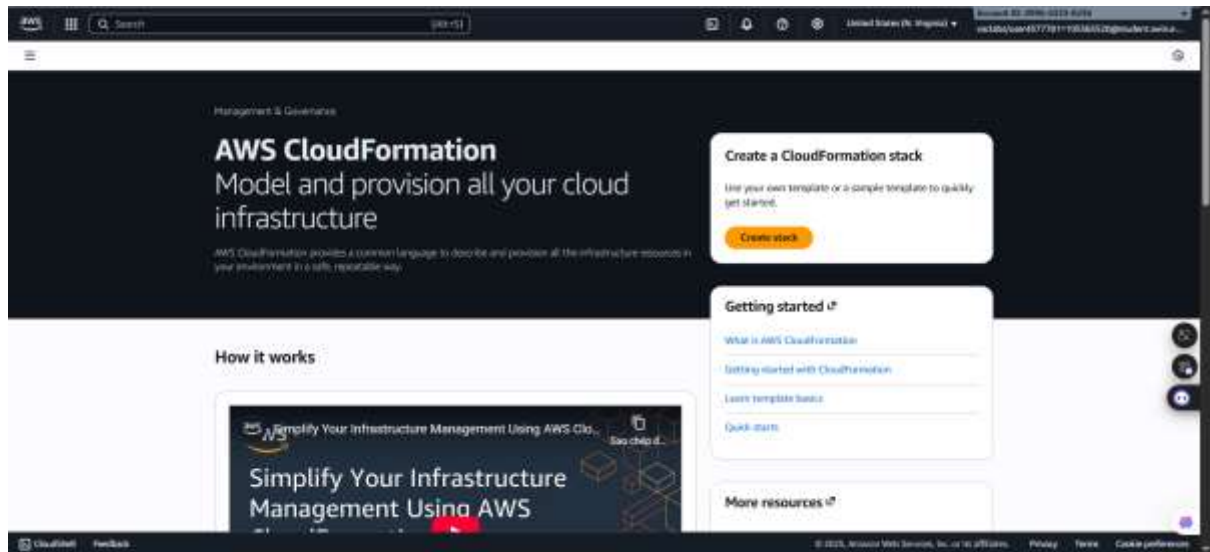


Figure 26: CloudFormation homepage

G. Task 5: Deleting the stack

Step 5.1: Return to the main AWS **CloudFormation** console, In the list of **stacks**, choose the **lab-application** link and choose **Delete**.

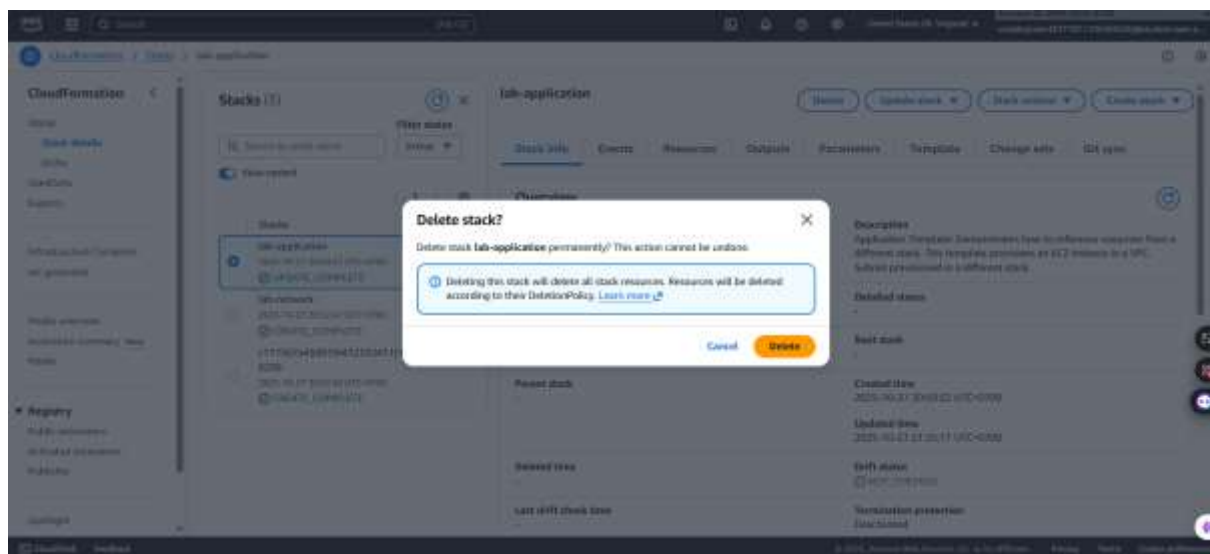


Figure 27: Delete announcement

Step 5.2: Choose **Delete**.

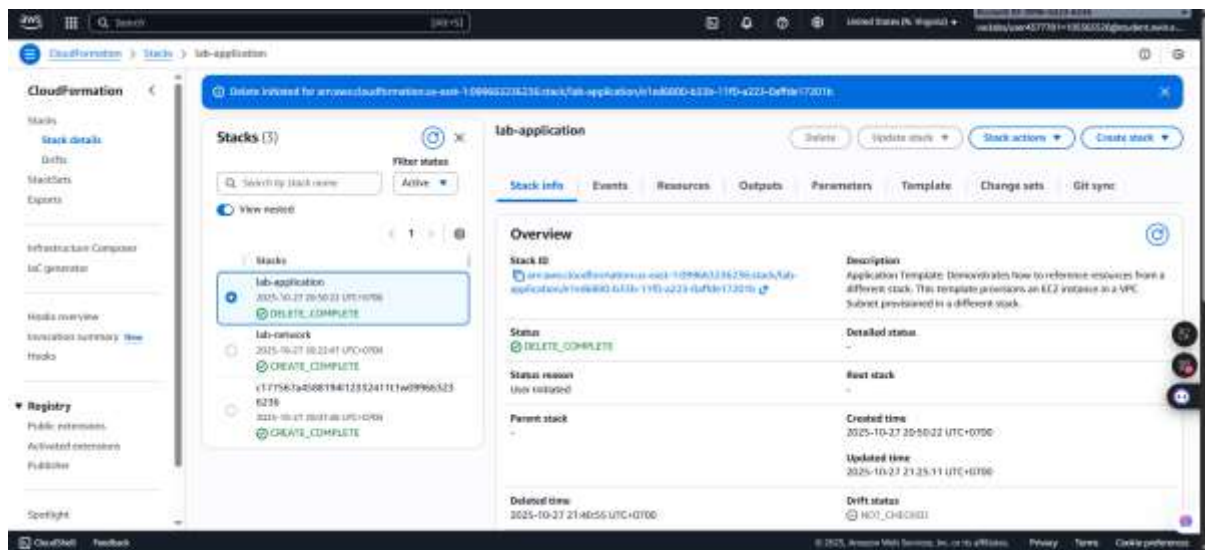


Figure 28: Delete successfully

Step 5.3: From the Services menu, choose EC2. In the left navigation pane, in the Elastic Block Store section, choose **Snapshots**.

You see a snapshot **Web Data** with a **Started** time in the last few minutes, and it changes to **Completed** soon.

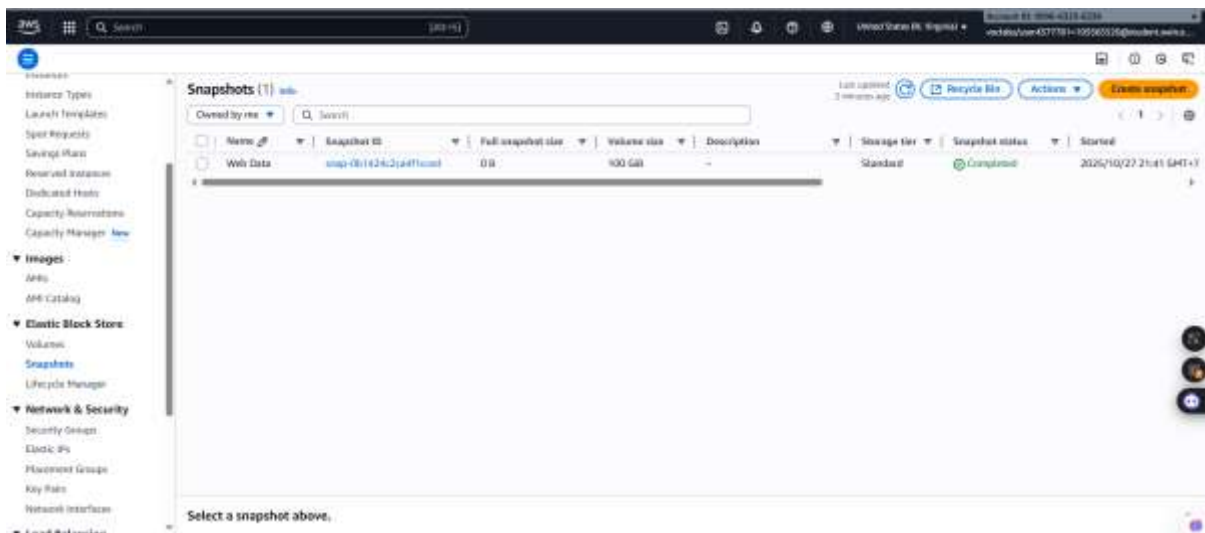
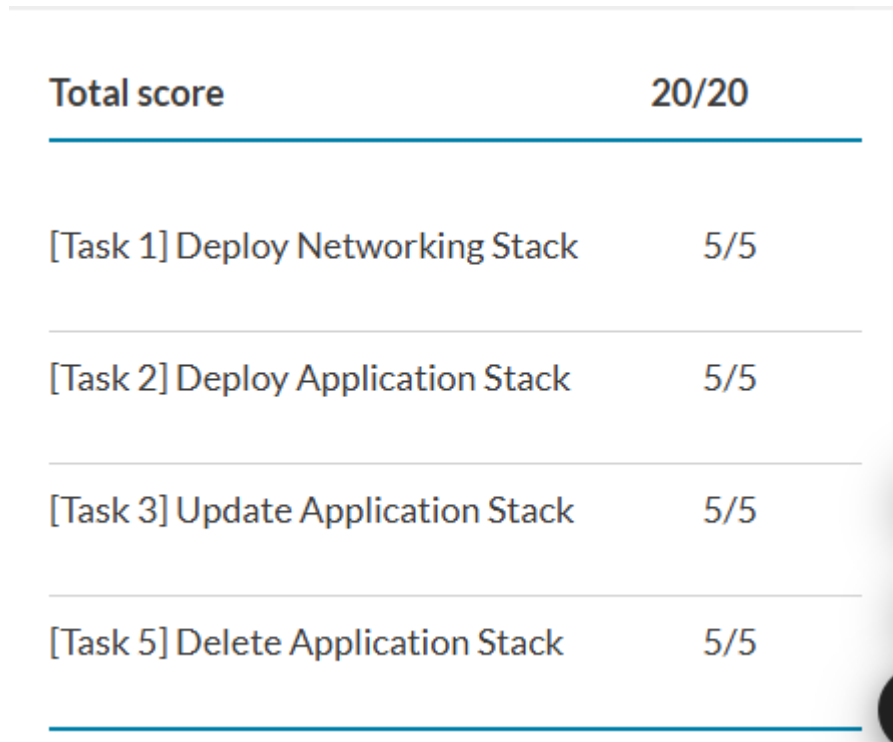


Figure 29: Snapshots page

G. Submitting work

A screenshot of a table with a white background and light gray horizontal dividers. The table has two columns: the first column contains task descriptions in a dark gray font, and the second column contains scores in a bold dark gray font. The first row is a summary row with a blue horizontal line above it and a blue horizontal line below it. The subsequent four rows are task-specific, each with a blue horizontal line below it. The table is partially cut off on the right side.

Total score	20/20
[Task 1] Deploy Networking Stack	5/5
[Task 2] Deploy Application Stack	5/5
[Task 3] Update Application Stack	5/5
[Task 5] Delete Application Stack	5/5

Figure 30: Completed work

H. Conclusions

I successfully completed this guided lab, demonstrating proficiency in automating infrastructure using AWS CloudFormation. I deployed the environment in layers, first creating the networking stack (lab-network) , and then the application stack (lab-application) which referenced the network outputs. I then practiced updating the application stack to modify the Security Group and add an HTTPS ingress rule. Finally, I verified the concept of deletion policies by observing that a snapshot of the application data was retained after I deleted the lab-application stack. This hands-on exercise confirmed my ability to use Infrastructure as Code for reliable deployment, updating, and layered resource management