**Guided Lab: Automating Infrastructure Deployment with AWS CloudFormation**

**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 out-of-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 will 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:

* 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.

**Accessing the AWS Management Console**

1. At the top of these instructions, choose Start Lab to launch your lab.

A **Start Lab** panel opens, and it displays the lab status.

**Tip**: If you need more time to complete the lab, restart the timer for the environment by choosing the Start Lab button again.

1. Wait until the **Start Lab** panel displays the message *Lab status: ready*, then close the panel by choosing the **X**.
2. At the top of these instructions, choose AWS.

This action opens the AWS Management Console in a new browser tab. The system automatically logs you in.

**Tip**: If a new browser tab does not open, a banner or icon is usually at the top of your browser with the message that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and then choose **Allow pop-ups**.

1. Arrange the **AWS Management Console** tab so that it displays alongside these instructions. Ideally, you will have both browser tabs open at the same time so that you can follow the lab steps more easily.

**Do not change the Region unless specifically instructed to do so**.

**Task 1: Deploying a networking layer**

It is a best practice to deploy infrastructure in *layers*. Common layers are:

* Network (Amazon VPC)
* 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 will deploy an AWS CloudFormation template that creates a *networking layer* by using Amazon VPC.

1. Right-click the following link and download the template to your computer: [lab-network.yaml](https://labs.vocareum.com/web/2153179/1120187.0/ASNLIB/public/scripts/lab-network.yaml)

 If you want, you can open the template in a text editor to see how the AWS resources are defined.

Templates can be written in JavaScript Object Notation (JSON) or YAML Ain't Markup Language (YAML). YAML is a markup language that is similar to JSON, but it is easier to read and edit.

1. In the **AWS Management Console**, from the **Services** menu, choose **CloudFormation**.
2. If you see this message, click Try it out now and provide us feedback:
3. Choose **Create stack** and configure these settings.

**Step 1: Specify template**

* + **Template source:** **Upload a template file**
  + **Upload a template file:** Click **Choose file** then select the **lab-network.yaml** file that you downloaded.
  + Choose **Next**

**Step 2: Create Stack**

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

**Step 3: Configure stack options**

* + In the **Tags** section, enter these values.
    - **Key:** application
    - **Value:** inventory
  + Choose **Next**

**Step 4: Review lab-network**

* + Choose **Create stack**

The *template* will now be used by AWS CloudFormation to generate a *stack* of resources in the AWS account.

The specified *tags* are automatically propagated to the resources that are created, which makes it easier to identify resources that are used by particular applications.

1. Choose the **Stack info** tab.
2. Wait for the **Status** to change to CREATE\_COMPLETE.

 Choose **Refresh**  every 15 seconds to update the display, if necessary.

You can now examine the resources that were created.

1. Choose the **Resources** tab.

You will see a list of the resources that were created by the template.

 If the list is empty, update the list by choosing **Refresh** .

1. Choose the **Events** tab and scroll through the events log.

The events log shows (from more recent to less recent) the activities that were performed by AWS CloudFormation. Example events include starting to create a resource and then completing the resource creation. Any errors that were encountered during the creation of the stack will be listed in this tab.

1. Choose the **Outputs** tab.

A CloudFormation stack can provide *output information*, such as the ID of specific resources and links to resources.

Two outputs are listed.

* + **PublicSubnet:** The ID of the public subnet that was created (for example: \_subnet-08aafd57f745035f1\_\_
  + **VPC:** The ID of the VPC that was created (for example: *vpc-08e2b7d1272ee9fb4*)

Outputs can also be used to provide values to other stacks. This is shown in the **Export name** column. In this case, the VPC and subnet IDs are given export names so that other stacks can retrieve the values. These other stacks can then build resources inside the VPC and subnet that were just created. You will use these values in the next task.

1. Choose the **Template** tab.

This tab shows the template that was used to create the stack—that is, the template that you uploaded while you created the stack. Feel free to examine the template and see the resources that were created. Also feel free to explore the **Outputs** section at the end (this section defined which values to export).

**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 will *import* the VPC and subnet IDs from the *Outputs* of the existing CloudFormation stack. It will then use this information to create the security group in the VPC and the EC2 instance in the subnet.

1. Right-click the following link and download the template to your computer: [lab-application.yaml](https://labs.vocareum.com/web/2153179/1120187.0/ASNLIB/public/scripts/lab-application.yaml)

 If you want, you can open the template in a text editor to see how resources are defined.

1. In the left navigation pane, choose **Stacks**.
2. Select **Create stack > With new resources (standard)**, and then configure these settings.

**Step 1: Specify template**

* + **Template source:** **Upload a template file**
  + **Upload a template file:** Click **Choose file** then select the **lab-application.yaml** file that you downloaded.
  + Choose **Next**

**Step 2: Create Stack**

* + **Stack name:** lab-application
  + **NetworkStackName:** lab-network
  + Choose **Next**

 The *Network Stack Name* parameter tells the template the name of the first stack that you created (*lab-network*), so it can retrieve values from the *Outputs*.

**Step 3: Configure stack options**

* + In the **Tags** section, enter these values.
    - **Key:** application
    - **Value:** inventory
  + Choose **Next**

**Step 4: Review lab-application**

* + Choose **Create stack**

While the stack is being created, examine the details in the **Events** tab and the **Resources** tab. You can monitor the progress of the resource-creation process and the resource status.

1. In the **Stack info** tab, wait for the **Status** to change to CREATE\_COMPLETE.

Your application is now ready!

1. Choose the **Outputs** tab.
2. Copy the **URL** that is displayed, open a new web browser tab, paste the URL, and press ENTER.

The browser tab will open the application, which is running on the web server that this new CloudFormation stack created.

A CloudFormation stack can use reference values from another CloudFormation stack. For example, this portion of the *lab-application* template references the *lab-network* template:



  WebServerSecurityGroup:

    Type: AWS::EC2::SecurityGroup

    Properties:

      GroupDescription: Enable HTTP ingress

      VpcId:

        Fn::ImportValue:

          !Sub ${NetworkStackName}-VPCID

The last line uses the *network stack name* that you provided (*lab-network*) when the stack was created. It imports the value of *lab-network-VPCID* from the *Outputs* of the first stack. It then inserts the value into the VPC ID field of the security group definition. The result is that the security group is created in the VPC that was created by the first stack.

Here is another example, which is in the CloudFormation template that you just used to create the application stack. This template code places the EC2 instance into the subnet that was created by the network stack:



  SubnetId:

    Fn::ImportValue:

    !Sub ${NetworkStackName}-SubnetID

It takes the *subnet ID* from the *lab-network* stack and uses it in the *lab-application* stack to launch the instance into the public subnet, which was created by the first stack.

**Task 3: Updating a Stack**

AWS CloudFormation can also *update* a stack that has been 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 will update the *lab-application* stack to modify a setting in the security group.

First, you will examine the current settings for the security group.

1. In the **AWS Management Console**, from the **Services** menu, choose **EC2**.
2. In the left navigation pane, choose **Security Groups**.
3. Select the check box for  **lab-application-WebServerSecurityGroup...**.
4. Choose the **Inbound rules** tab.

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

You will now return to AWS CloudFormation to update the stack.

1. From the **Services** menu, choose **CloudFormation**.
2. Right-click the following link and download the updated template to your computer: [lab-application2.yaml](https://labs.vocareum.com/web/2153179/1120187.0/ASNLIB/public/scripts/lab-application2.yaml)

This template has an additional configuration to permit inbound Secure Shell (SSH) traffic on port 22:



  - IpProtocol: tcp

    FromPort: 22

    ToPort: 22

    CidrIp: 0.0.0.0/0

1. In the **Stacks** list of the **AWS CloudFormation console**, select **lab-application**.
2. Choose **Update** and configure these settings.
   * Select **Replace current template**
   * **Template source:** **Upload a template file**
   * **Upload a template file:** Click **Choose file** then select the **lab-application2.yaml** file that you downloaded.
3. Choose **Next** in each of the next *three* screens to advance to the **Review lab-application** page.

In the **Change set preview** section at the bottom of the page, AWS CloudFormation displays the resources that will be updated:

This change set preview indicates that AWS CloudFormation will *Modify* the *WebServerSecurityGroup* without needing to replace it (*Replacement = False*). This change set means that the security group will have a minor change applied to it, and no references to the security group will need to change.

1. Choose **Update stack**
2. In the **Stack info** tab, wait for the **Status** to change to UPDATE\_COMPLETE.

 Update the status by choosing **Refresh**  every 15 seconds, if necessary.

You can now verify the change.

1. Return to the **Amazon EC2 console** and from the left navigation pane, choose **Security Groups**.
2. In the **Security Groups** list, select **lab-application-WebServerSecurityGroup**.

The **Inbound rules** tab should display an additional rule that allows *SSH* traffic over *TCP port 22*.

This subtask demonstrates how changes can be deployed in a repeatable, documented process. The AWS CloudFormation templates can be stored in a source code repository (such as AWS CodeCommit). This way, you can maintain versions and a history of the templates and the infrastructure that was deployed.

**Task 4: Exploring templates with AWS CloudFormation Designer**

*AWS CloudFormation 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.

In this task, you will gain some hands-on experience with Designer.

1. From the **Services** menu, choose **CloudFormation**.
2. In the left navigation pane, choose **Designer**.

**Tip:** You might need to expand the left navigation pane by choosing the menu icon.

1. Choose the **File**  menu, select **Open > Local file**, and select the **lab-application2.yaml** template that you downloaded previously.

Designer will display a graphical representation of the template:

Instead of drawing a typical architecture diagram, Designer is a visual editor for AWS CloudFormation templates. It draws the resources that are defined in a template and their relationship to each other.

1. Experiment with the features of the Designer. Some things to try are:
   * Click the displayed resources. The lower pane will then display the portion of the template that defines the resources.
   * Try dragging a new resource—from the **Resource types** pane on the left—into the design area. The definition of the resource will be automatically inserted into the template.
   * Try dragging the resource connector circles to create relationships between resources.
   * Open the **lab-network.yaml** template that you downloaded earlier in the lab and also explore its resources in Designer.

**Task 5: Deleting the stack**

When resources are no longer required, AWS CloudFormation can delete the resources built for the stack.

A *deletion policy* can also be specified against resources. It can preserve or (in some cases) back up a resource when its stack is deleted. This feature is useful for retaining databases, disk volumes, or any resource that might be needed after the stack is deleted.

The *lab-application* stack was configured to take a snapshot of an Amazon Elastic Block Store (Amazon EBS) disk volume before it is deleted. The code in the template that accomplishes that configuration is:



DiskVolume:

  Type: AWS::EC2::Volume

  Properties:

    Size: 100

    AvailabilityZone: !GetAtt WebServerInstance.AvailabilityZone

    Tags:

      - Key: Name

        Value: Web Data

  DeletionPolicy: Snapshot

The *DeletionPolicy* in the final line directs AWS CloudFormation to create a snapshot of the disk volume before it is deleted.

You will now delete the *lab-application* stack and see the results of this deletion policy.

1. Return to the main **AWS CloudFormation console** by choosing the Close link at the top of the Designer page (choose **Leave page** if prompted).
2. In the list of stacks, choose the **lab-application** link.
3. Choose **Delete**
4. Choose **Delete stack**

You can monitor the deletion process in the **Events** tab and update the screen by choosing **Refresh**  occasionally. You might also see an events log entry that indicates that the EBS snapshot is being created.

1. Wait for the stack to be deleted. It will disappear from the stacks list.

The application stack \_\_ removed, but the network stack remained untouched. This scenario reinforces the idea that different teams (for example, the network team or the application team) could manage their own stacks.

You will now verify that a snapshot of the EBS volume was created before the EBS volume was deleted.

1. From the **Services** menu, choose **EC2**.
2. In the left navigation pane, choose **Snapshots**.

You should see a snapshot with a **Started** time in the last few minutes.

**Submitting your work**

1. At the top of these instructions, choose **Submit** to record your progress and when prompted, choose **Yes**.
2. If the results don't display after a couple of minutes, return to the top of these instructions and choose Grades

**Tip**: You can submit your work multiple times. After you change your work, choose **Submit** again. Your last submission is what will be recorded for this lab.

1. To find detailed feedback on your work, choose Details followed by  **View Submission Report**.

**Lab complete**

 Congratulations! You have completed the lab.

1. Choose End Lab at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated... You may close this message box now.*

1. Select the **X** in the top right corner to close the panel.