Automating Infrastructure Deployment with AWS CloudFormation

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

This lab shows how to deploy multiple layers of infrastructure with CloudFormation, update a stack and delete a stack (while retaining some resources).

In this lab you will:

- Use AWS CloudFormation to deploy a 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.

Accessing the AWS Management Console

- Click Open Console
- Sign in to the AWS Management Console using your credentials

⚠ Please do not change the Region during this lab.

Task 1: Deploy a Networking Layer

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

- Network (Amazon VPC)
- Database
- Application

This way, templates can be re-used between systems, such as deploying a common network topology between Dev/Test/Production or deploying a standard database for multiple application.

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

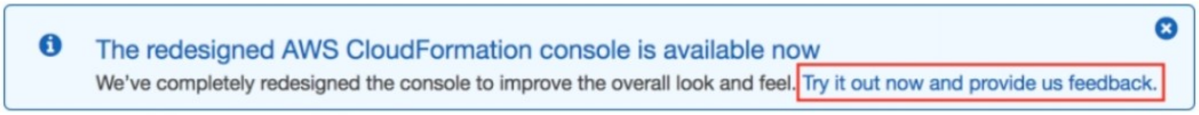
1. Download the template to your computer: lab-network.yaml (download from sharefolder)

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

Templates can be written in JSON or YAML. YAML is a markup language similar to JSON, but is easier to read and edit.

2. In the **AWS Management Console**, on the **Services** menu, click **CloudFormation**.

3. If you see this message, click Try it out now and provide us feedback:



The redesigned AWS CloudFormation console is available now

We've completely redesigned the console to improve the overall look and feel. Try it out now and provide us feedback.

4. Click **Create stack** and configure:

Step 1: Specify template

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

Step 2: Create Stack

- **Stack name**: *lab-network*
- Click **Next**

Step 3: Configure stack options

- Tags:
    - **Key**: application
    - **Value:** inventory
- Click **Next**

Step 4: Review lab-network

- Click Create stack

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

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

5. Click the **Stack info** tab.

6. Wait for the **Status** to change to CREATE_COMPLETE.

💬Click Refresh every 15 seconds to update the display if necessary. You can now examine the resources that were created.

7. Click the **Resources** tab.

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

💬If the list is empty, click Refresh to update the list.

8. Click the **Events** tab and scroll through the listing.

The listing shows (in reverse order) the activities performed by CloudFormation, such as starting to create a resource and then completing the resource creation. Any errors encountered during the creation of the stack will be listed in this tab.

9. Click the **Outputs** tab.

A CloudFormation stack can provide output information, such as the ID of specific resources and links to resources. You will see two outputs:

- **PublicSubnet:** The ID of the Public Subnet that was created (eg subnet-08aafd57f745035f1)
- **VPC:** The ID of the VPC that was created (eg vpc-08e2b7d1272ee9fb4)

Outputs can also provide values that will be used by other stacks. This is shown in the **Export name** column. In this case, the VPC and Subnet IDs are given an export name so that other stacks can retrieve the values and build resources inside the VPC and Subnet. You will be using these values in the next task.

10. Click the **Template** tab.

This tab shows the template that was used to create the stack. It shows the template that you uploaded while creating the stack. Feel free to examine the template and see the resources that were created, and the **Outputs** section at the end that defined which values to export.

Task 2: Deploy an Application Layer

Now that the network layer has been deployed, you will deploy an application layer that contains an Amazon EC2 instance and a Security Group.

The 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.

11. Download the template to your computer: **lab-application.yaml** (download it from share folder)

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

12. In the left navigation pane, click **Stacks**.

13. Click **Create stack** and configure:

Step 1: Specify template

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

Step 2: Create Stack

- **Stack name:** *lab-application*
- **NetworkStackName:** *lab-network*
- Click **Next**

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

Step 3: Configure stack options

- Tags:
  - **Key**: application
  - **Value**: inventory
- Click **Next**

Step 4: Review lab-network

- Click **Create stack**

While the stack is being created, examine the **Events** and **Resources** tab to view the resources that are being created.

14. Wait for the **Status** (in the Stack info tab) to change to CREATE_COMPLETE. Your application is now ready!

15. Click the **Outputs** tab.

16. Copy the **URL** that is displayed, then open a new web browser tab, paste the URL and press Enter.

A new browser tab will open, taking you to the application running on the web server.

A CloudFormation stack can also reference values from another CloudFormation stack. For example, here is a portion of the *lab-application* template that 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 to the Network Stack Name that you provided ("lab-network") when the stack was created. It then imports the value of lab-network-VPCID from the Outputs of the first stack and 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 created by the first stack.

In another example, here is the code that places the Amazon EC2 instance into the correct subnet:

```
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 that created by the first stack.

This demonstrates how multiple CloudFormation stacks can be used to deploy infrastructure in *multiple layers.*

Task 3: Update a Stack

CloudFormation can also update a stack that has been deployed. When updating a stack, 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. CloudFormation will leave all other resources as-is, without being modified by the update.

First, you will examine the current settings on the Security Group.

17. In the **AWS Management Console**, on the **Services** menu, click **EC2**.

18. In the left navigation pane, click **Security Groups**.

19. Select ☑ **Web Server Security Group.**

20. Click **the Inbound tab**.

You will see that there is currently only one rule in the Security Group, which permits HTTP traffic. You will now return to CloudFormation to update the stack.

21. On the **Services** menu, click **CloudFormation**.

22. Download the updated template to your computer: **lab-application2.yaml (**download from share folder)

This template has an additional configuration to permit inbound SSH traffic on port 22:

```
- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0
```

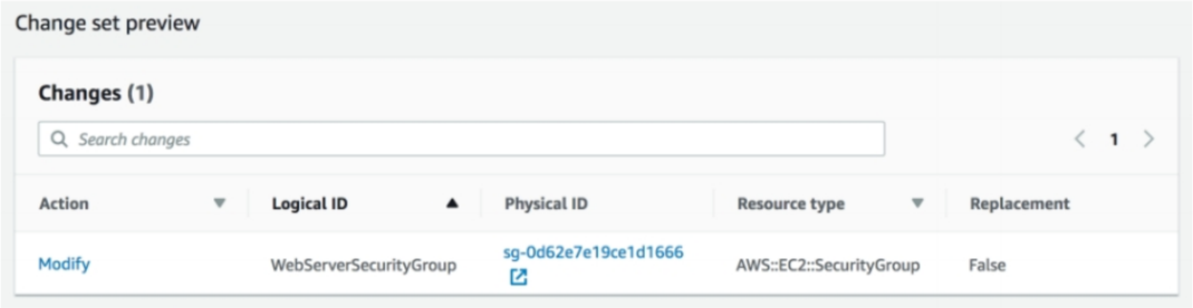
23. Click **lab-application.**

24. Click **Update** and configure:

- Click **Replace current template**
- **Template source: Upload a template file**
- **Upload a template file:** Click **Choose file** then select the **lab-application2.yaml** file you downloaded.

25. Click **Next three times** to advance to the Review page.

In the **Change set preview** section at the bottom of the page, CloudFormation will display what resources need to be updated:



Change set preview

Changes (1)

Action	Logical ID	Physical ID	Resource type	Replacement
Modify	WebServerSecurityGroup	sg-0d62e7e19ce1d1666	AWS::EC2::SecurityGroup	False

This is indicating that CloudFormation will **Modify** the Web Server security group without needing to replace it *(Replacement = False)*. This means there will be a minor change to the Security Group and no references to the security group will need to change.

26. Click **Update stack**

27. Wait for the **Status** (in the Stack info tab) to change to CREATE_COMPLETE.

💬Click Refresh every 15 seconds to update the status if necessary. You can now verify the change.

28. Return to the **EC2 console** and select the **Web Server security group.**

The **Inbound** tab should display an additional rule for *SSH traffic.*

This demonstrates how changes can be deployed in a repeatable, documented process. The CloudFormation template can be stored in a Source Code Repository (eg AWS CodeCommit) to maintain a history of the template and the infrastructure that has been deployed.

Task 4: Explore 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 using a drag-and-drop interface, and then edit their details using the integrated JSON and YAML editor. Whether you are a new or an experienced AWS CloudFormation user, AWS CloudFormation Designer can help you quickly see the interrelationship between a template's resources and easily modify templates.

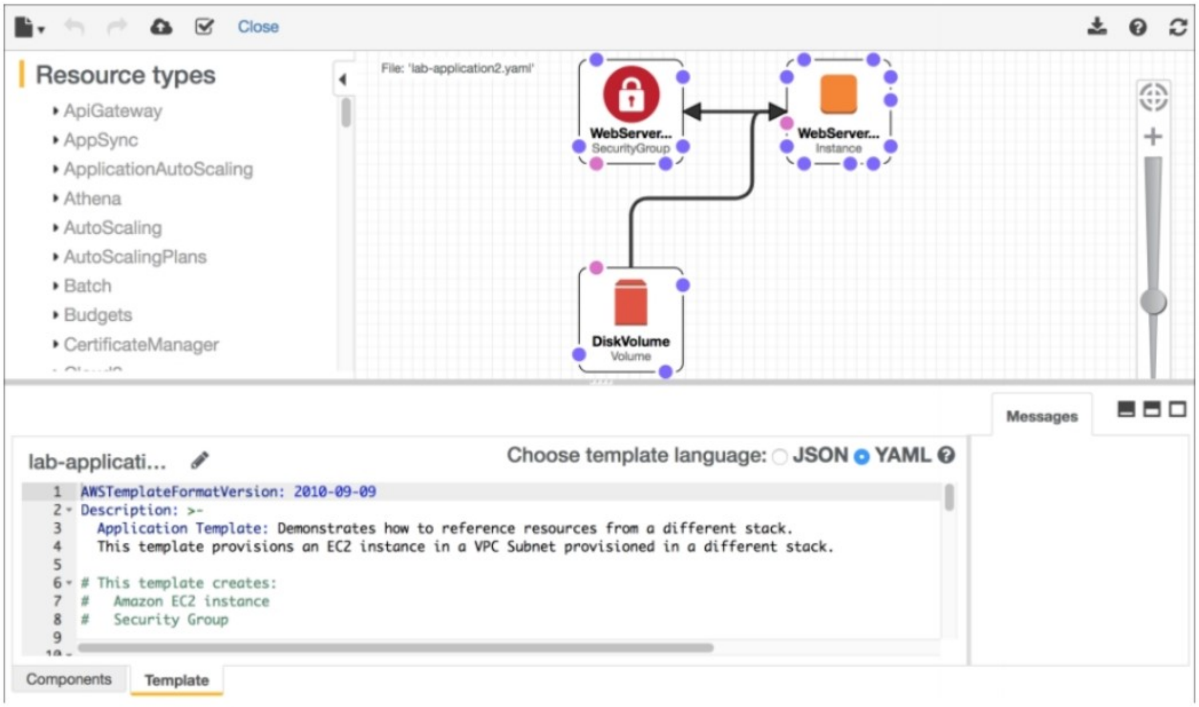
In this task, you will gain some hands-on experience with the Designer.

29. On the **Services** menu, click **CloudFormation**.

30. In the left navigation pane, click **Designer**.

31. Use the File menu to open a **Local file** and **select the lab-application2.yaml** template you downloaded previously.

Designer will display a graphical representation of the template:



Choose template language: ○ JSON ● YAML

```
AWSTemplateFormatVersion: 2010-09-09
Description: >-
  Application Template: Demonstrates how to reference resources from a different stack.
  This template provisions an EC2 instance in a VPC Subnet provisioned in a different stack.

# This template creates:
#   Amazon EC2 instance
#   Security Group

```

Rather than drawing a typical architecture diagram, Designer is a visual editor for CloudFormation templates so it draws the resources defined in a template and their relationships to each other.

32. Experiment with the features of the Designer. Some things to try are:

- Click on 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 you downloaded earlier in the lab and explore its resources too

Task 5: Delete the Stack

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

A **Deletion Policy** can also be specified against resources, which can preserve or (in some cases) backup a resource when its stack is deleted. This is useful for retaining databases, disk volumes or any resource that might be required after stack deletion.

The *lab-application stack* has been configured to take a snapshot of an Amazon EBS disk volume before it is deleted:

```
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 is directing 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.

33. Return to the main CloudFormation console by clicking the Close link at the top of the page.

34. Click **lab-application.**

35. Click **Delete**.

36. Click **Delete stack**

You can monitor the deletion process in the **Events** tab and update the screen by clicking Refresh occasionally. You might also see a reference to the EBS snapshot being created.

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

The application stack has been removed, but the network stack has remained untouched. This reinforces the idea that different teams (eg network team, application team) can manage their own stacks.

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

38. On the **Services** menu, click **EC2**.

39. In the left navigation pane, click **Snapshots**.

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

Lab Complete

💬Congratulations! You have completed the lab.

Delete all resources which created for this lab, if you are working with trail account. 😊