Lab - Automation with 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 provides experience in deploying and editing CloudFormation stacks. It is an interactive experience, requiring you to consult documentation to discover how to define resources within a CloudFormation template.

The lab will demonstrate how to:

- **Deploy** an AWS CloudFormation stack with a defined Virtual Private Cloud (VPC), and Security Group.
- **Configure** an AWS CloudFormation stack with resources, such as an Amazon Simple Storage Solution (S3) bucket and Amazon Elastic Compute Cloud (EC2).
- **Terminate** an AWS CloudFormation and its respective resources.

Duration

This lab will require approximately **45 minutes** to complete.

Accessing the AWS Management Console

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

   A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.
3. At the top of these instructions, click AWS

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

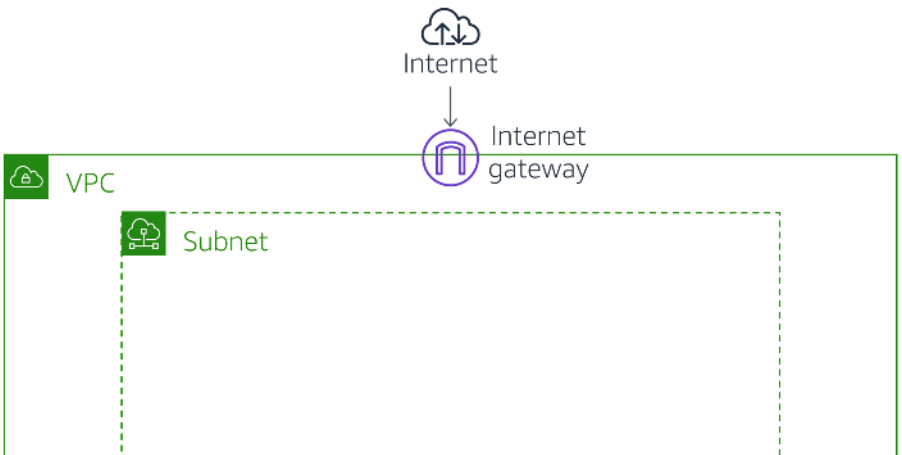
   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

   ⚠ Please do not change the Region during this lab.

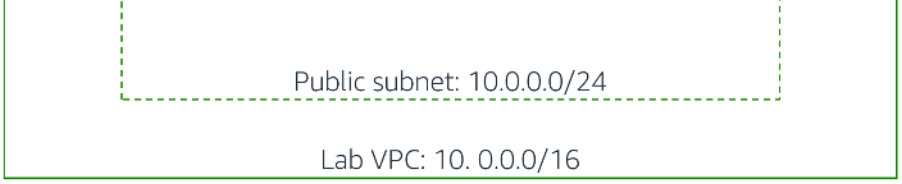
Task 1: Deploy a CloudFormation Stack

You will begin by deploying a CloudFormation stack that creates a VPC as shown in this diagram:



Public Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	Internet gateway



5. Right-click this link and download the CloudFormation template: task1.yaml
6. Open this file in a Text Editor (not a Word Processor).

   Look through the file. You will notice several sections:

   - The Parameters section is used to prompt for inputs that can be used elsewhere in the template. The template is asking for two IP address (CIDR) ranges for defining the VPC.
   - The Resources section is used to define the infrastructure to be deployed. The template is defining the VPC, and a Security Group.
   - The Outputs section is used to provide selective information about resources in the stack. The template is providing the Default Security Group for the VPC that is created.

   The template is written in a format called YAML, which is commonly used for configuration files. The format of the file is important, including the indents and hyphens. CloudFormation templates can also be written in JSON.

   You will now use this **template** to launch a **CloudFormation stack**.
7. In the **AWS Management Console**, on the **Services** menu, click **CloudFormation**.
8. Click **Create stack** then:

   - Click ⦿ **Upload a template file**
   - Click **Browse** or **Choose file** and upload the template file you downloaded earlier
   - Click **Next**
9. On the **Specify Details** page, configure:

   - **Stack name:** `Lab`

     In the **Parameters** section, you will see that CloudFormation is prompting for the IP address ('CIDR') range for the VPC and Subnet. A default value has been specified by the template, so there is no need to modify these values.
10. Click **Next**

    The **Options** page can be used to specify additional parameters. You can browse the page, but leave settings at their default values.
11. Click **Next**

    The **Review** page displays a summary of all settings. Some of the resources are defined with *custom names*, which can lead to naming conflicts. Therefore, CloudFormation prompts for an acknowledgement that custom names are being used.
12. Click **Create stack**

    The stack will now enter the *CREATE_IN_PROGRESS* status.
13. 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.
14. Click the **Resources** tab.

    The listing shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.
15. Wait until the status changes to *CREATE_COMPLETE*. You can click **Refresh** occasionally to update the display.

    Optional: Go to the VPC console to see the *Lab VPC* that was created. Then, return to the CloudFormation console.

Task 2: Add an Amazon S3 Bucket to the Stack

In this task, you will gain experience in editing a CloudFormation template.

Your objective is:

- Add an Amazon S3 bucket to the template
- Then **update the stack** with the revised template

This will result in a new bucket being deployed.

Rather than following pre-defined steps, you will need to discover how to **update the template yourself**!

Here are some tips:

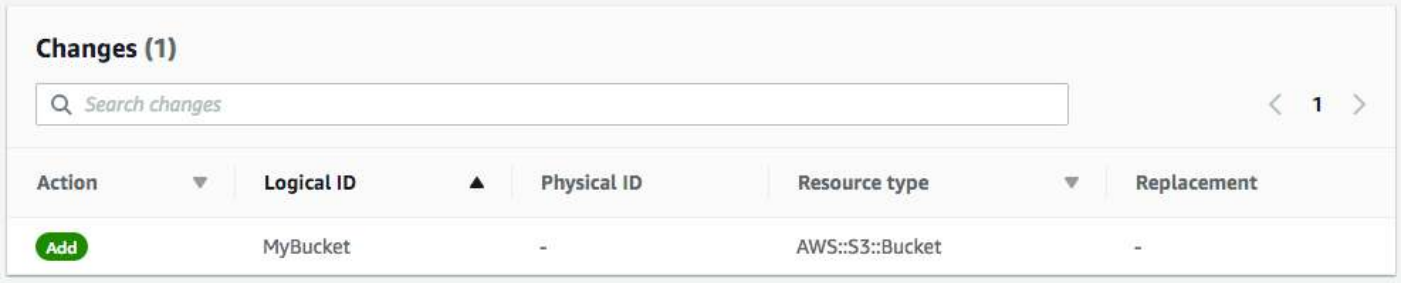
- You should edit the **task1.yaml** file you downloaded earlier to include an Amazon S3 bucket
- Use this documentation page for assistance: Amazon S3 Template Snippets
- Look at the **YAML example**
- Your code should go under the **Resources:** header in the template file
- **You do not require any *Properties* for this bucket resource**
- Indents are important in YAML — use two spaces for each indent
- The correct solution is actually **only needs two lines** — one for the identifier and one for the Type

Once you have edited the template, continue with the following steps to update the stack.

16. In the CloudFormation console, select ☑ **Lab**.
17. Click **Update**.
18. Choose **Replace current template**, then choose **Upload a template file**. Click **Choose file**, then browse to and select the task1.yaml file that you modified.
19. Click **Next**

    ⚠ If you receive an error message, ask your instructor for assistance in debugging the problem.
20. On the **Specify stack details** page, click **Next**
21. On the **Configure stack options** page, click **Next**

    Wait for CloudFormation to calculate the changes. Towards the bottom of the page, you should see something similar to this:



    This indicates that CloudFormation will **Add** an Amazon S3 bucket. All other resources defined in the template will be **unchanged**. This demonstrates that it is fast and easy to add additional resources to an existing stack, since those resources do not need to be redeployed.
22. Click **Update stack**

    After a minute, the stack status will change from *UPDATE_IN_PROGRESS* to *UPDATE_COMPLETE*.
23. Click the **Resources** tab.

    The bucket will now be displayed in the list of resources. CloudFormation will have assigned it a random name so that it does not conflict with any existing buckets.

    ⚠ If the bucket was not correctly created, please ask your instructor for assistance.

To download a sample solution, right-click and download this link: task2.yaml

Optional: Go to the S3 console to see the bucket that was created. Then, return to the CloudFormation console.

Task 3: Add an Amazon EC2 Instance to the Stack

In this task, your objective is to **add an Amazon EC2 instance to the template**, then update the stack with the revised template.

Whereas the bucket definition was rather simple (just two lines), defining an Amazon EC2 instance is more complex because it needs to use associated resources, such as an AMI, security group and subnet.

First, however, you will add a special parameter that is used to provide a value for the Amazon Machine Image (AMI).

24. Update the template by adding these lines in the **Parameters** section:

```
AmazonLinuxAMIID:
  Type: AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>
  Default: /aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2
```

This parameter uses the **AWS Systems Manager Parameter Store** to retrieve the latest AMI (specified in the *Default* parameter, which in this case is *Amazon Linux 2*) for the stack's region. This makes it easy to deploy stacks in different regions without having to manually specify an AMI ID for every region.

For more details of this method, see: AWS Compute Blog: Query for the latest Amazon Linux AMI IDs using AWS Systems Manager Parameter Store.

When writing CloudFormation templates, you can refer to other resources in the template by using the `!Ref` keyword. For example, here is a portion of the *task1.yaml* template that defines a VPC, then references the VPC within the Route Table definition:

```
VPC:
  Type: AWS::EC2::VPC
  Properties:
    CidrBlock: 10.0.0.0/16

PublicRouteTable:
  Type: AWS::EC2::RouteTable
  Properties:
    VpcId: !Ref VPC
```

Note that it uses `!Ref VPC` to refer to the VPC resource. You will use this technique when defining the EC2 instance.

25. Use the tips below to update the template to **add an Amazon EC2 instance** with the following **Properties**:
    - **ImageId:** Refer to `AmazonLinuxAMIID`, which is the parameter added in the previous step
    - **InstanceType:** `t3.micro`
    - **SecurityGroupIds:** Refer to `AppSecurityGroup`, which is defined elsewhere in the template
    - **SubnetId:** Refer to `PublicSubnet`, which is defined elsewhere in the template
    - **Tags:** Use this YAML block:

```
    Tags:
      - Key: Name
        Value: App Server
```

Here are some tips:

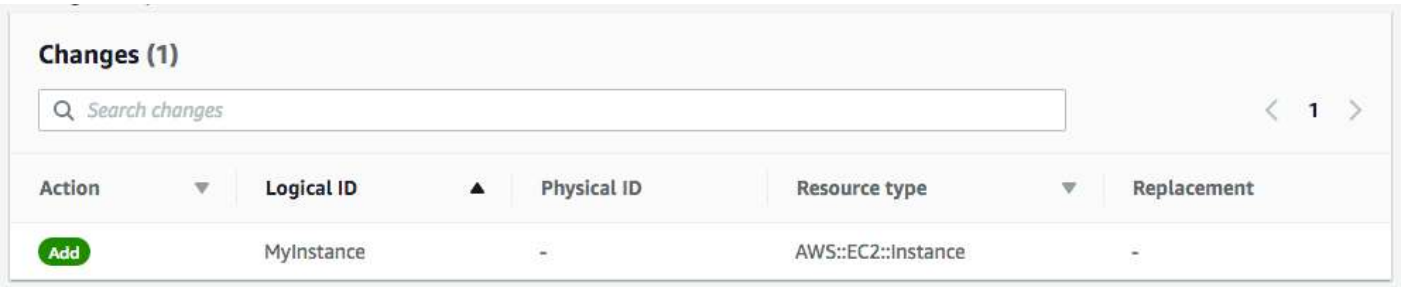
- Use this documentation page for assistance: AWS::EC2::Instance

- Use the **YAML version**
- Your code should go under the **Resources:** header in the template file
- **Only add the five Properties listed above**, there is no need to include any other properties
- When referring to other resources in the same template, use `!Ref` — see the example at the beginning of this task
- When referring to **SecurityGroupIds**, the template is actually expecting a *list* of security groups. You therefore need to list the security group like this:

```
    SecurityGroupIds:
      - !Ref AppSecurityGroup
```

26. Once you have edited the template, **update the stack** with your revised template file.

    You should see this before deploying the update:



⚠ If you are experiencing difficulties in editing the template, please ask your instructor for assistance.

To download a sample solution, right-click and download this link: task3.yaml

The instance will now be displayed in the **Resources** tab.

Optional: Go to the EC2 console to see the *App Server* that was created. Then, return to the CloudFormation console.

Task 4: Delete the Stack

When a CloudFormation stack is deleted, CloudFormation will automatically delete the resources that it created.

You will now delete the stack.

27. In the CloudFormation console, select ☑ **Lab**.
28. Click **Delete**, then at the prompt, click **Delete stack** .

    The stack will show *DELETE_IN_PROGRESS*. After a few minutes, the stack will disappear.

    Optional: Verify that the Amazon S3 bucket, Amazon EC2 instance and the VPC have been deleted.

Lab Complete

Congratulations! You have completed the lab.

29. Click End Lab at the top of this page and then click **Yes** to confirm that you want to end the lab.

    A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."
30. Click the **X** in the top right corner to close the panel.