**Automating Infrastructure Deployment with AWS CloudFormation**

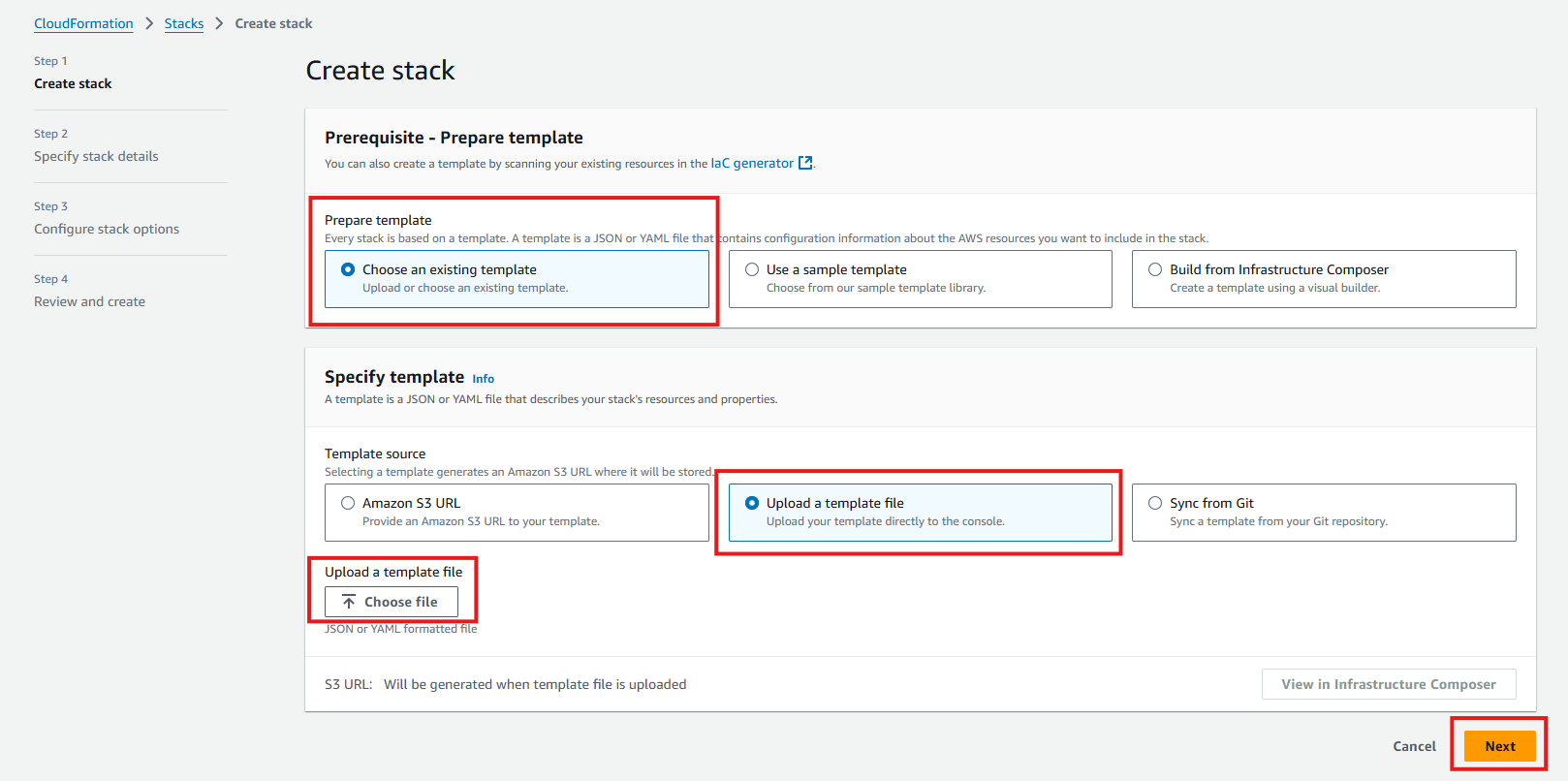
This project demonstrates my ability to deploy multiple layers of infrastructure with AWS CloudFormation, update a CloudFormation stack and delete a stack (while retaining some resources).

I leveraged AWS CloudFormation to:

1. deploy a virtual private cloud (VPC) networking layer
2. deploy an application layer that references the networking layer
3. explore templates with AWS CloudFormation Designer
4. delete a stack that has a deletion policy

Task1: Deploying A Networking Layer

1. Create stack in CloudFormation Service in the AWS management console

* Specified a template
* 
* I uploaded my YAML template named “lab-network.yaml”

A screenshot of a computer

Description automatically generated

1. Created a stack

* Stack name: lab-network
* A screenshot of a computer

  Description automatically generated

1. Configured Stack options

* In the Tags section, I entered these values:

Key: Application

Value: Inventory

A screenshot of a computer

Description automatically generated

1. I reviewed and created the stack

A screenshot of a computer

Description automatically generated

Task 2: Deploying an application layer

* Created an EC2 instance and a security group

1. Specified and uploaded a new template file

A screenshot of a computer

Description automatically generated

1. Created a stack

Stack name: lab-application

NetworkStackName: lab-network

A screenshot of a computer

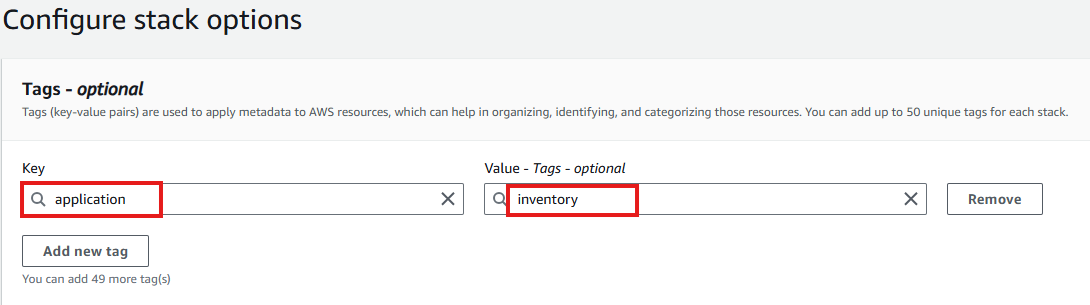
Description automatically generated

1. Configured Stack Options

* In the Tags section, I entered the following values

Key: application

Value: inventory



1. Reviewed and Created the new stack

A screenshot of a computer

Description automatically generated

1. I copied the URL form the output tab and pasted it a new browser tab.

A screenshot of a computer

Description automatically generated

1. The link opens the application which is running on the web server that this new CloudFormation stack created.

A screenshot of a computer

Description automatically generated

Task 3: Updating the lab-application stack to modify a setting in the security group

1. I navigated to EC2 services and selected security groups to check the current settings of the WebServerSecurityGroup.
2. I selected the inbound tap and this had only one rule

* The rule permitted HTTP traffic

A screenshot of a computer

Description automatically generated

1. I navigated back to the CloudFormation page to modify the lab-application template.
2. The new template had an additional configuration to permit inbound SSH traffic on port 22.

A screenshot of a computer program

Description automatically generated

1. In the stacks, I selected the lab-application and clicked on the update tab

A screenshot of a computer

Description automatically generated

1. I uploaded the new file

A screenshot of a computer

Description automatically generated

1. I left everything as it was, submitted the new file and waited for the update to complete.

A screenshot of a computer

Description automatically generated

1. I navigated back to the Webserver security group to verify that an additional inbound rule has been added.

Conclusion

* This demonstration shows 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).
* Versions and history of the templates and the infrastructure that was deployed can be maintained.

A screenshot of a computer

Description automatically generated

Task 4: Exploring templates with AWS CloudFormation Designer

1. I navigated to the designer section of CloudFormation and uploaded the second lab-application file to show the interrelationship between the templates resources.

A screenshot of a computer

Description automatically generated

Task 5 : Deleting the stack