# Automate Provisioning of AWS Resources using CloudFormation

Created by[Stephane Maarek | AWS Certified Cloud Practitioner,Solutions Architect,Developer](https://deloittedevelopment.udemy.com/user/stephane-maarek/)

**AWS Workspace**

**60-90 minutes**

In this lab, you will be a SysOps Administrator at a company that wants to automate the provisioning of AWS resources using CloudFormation. The company has seen rapid growth over the last year and now wants to overhaul its ad-hoc resource-provisioning processes. The company aims to implement a solution that facilitates automated provisioning of new resources and also supports the ability to preview the changes to existing resources, validate that they are in line with the expectations, and then roll-out the updates

Our SysOps Administration team has been tasked with building an automated resource provisioning system. We need you to implement a solution that helps the company to launch and configure the resources together as a stack. The solution should facilitate the ability to preview the changes to existing resources so any breaking changes can be preempted before the infrastructure is rolled out.

To build this end-to-end provisioning solution, you will leverage CloudWatch templates to automate the provisioning of resources. We want you to use CloudFormation change sets to monitor and approve the roll out of any updates to the existing infrastructure.

TASK

1-provision EC2 instance in a custom VPC using cloudformation

2-updae EC2 insatance by using the cloudformation change sets

3-Update the ecurity group by using the cloudformation change sets

4-Inspect the resources created by cloudformation

5-clean up

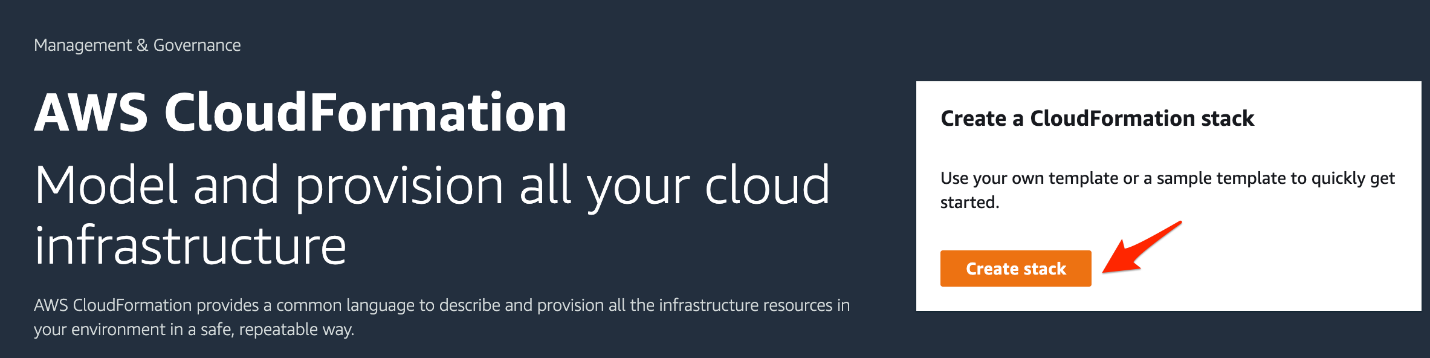
Provision EC2 instance in a custom VPC using cloudformation

To do this, you will need to set up a CloudFormation template to provision a t2.micro type of EC2 instance in a custom VPC. The template will also create a security group to allow HTTP traffic on port 80. The CloudFormation template will enforce the configuration standards by using parameters so that only t2.micro, t2.small or t2.medium instance types are allowed.

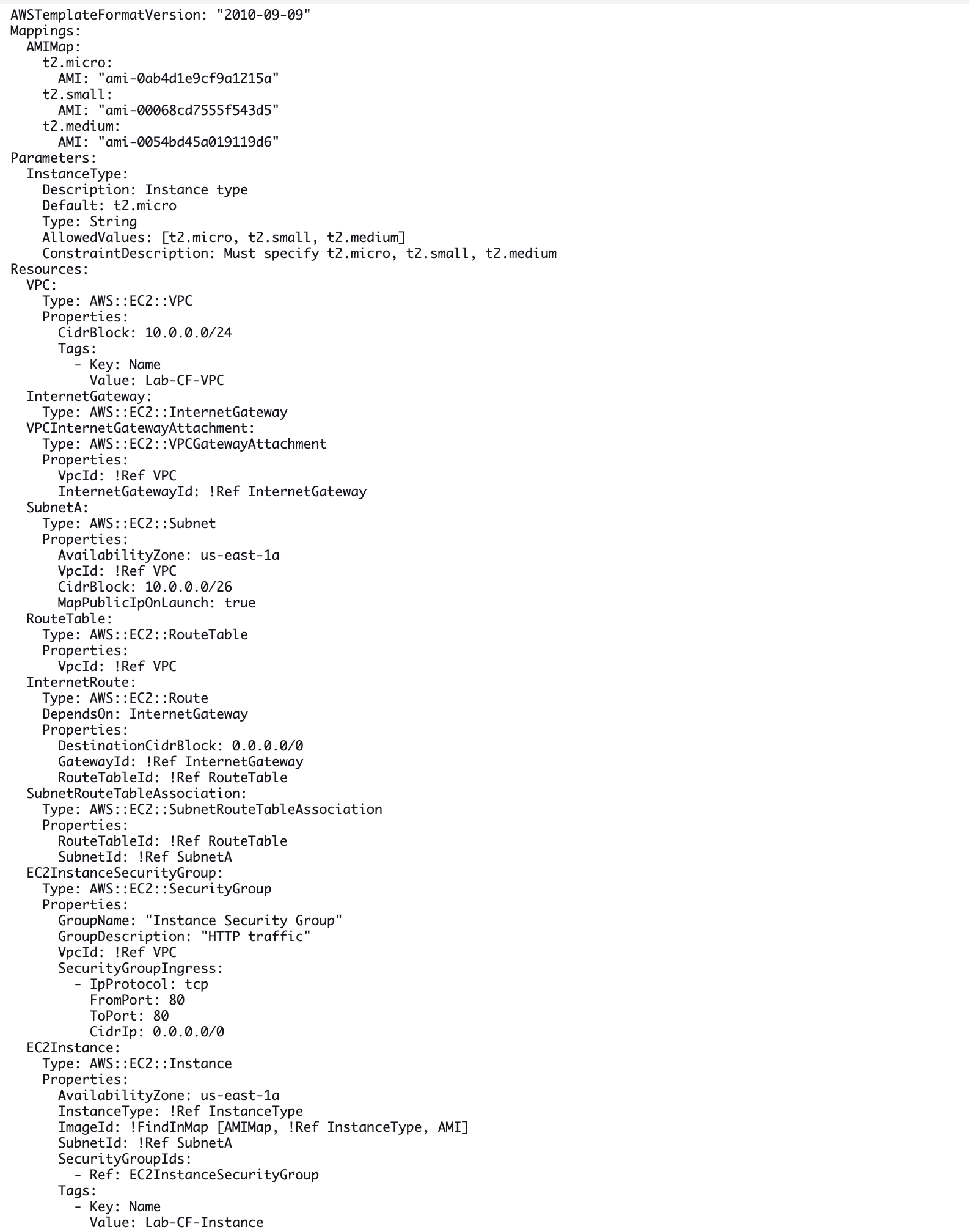
1. Make sure that you are in the N. Virginia AWS Region on the AWS Management Console. Enter **CloudFormation** in the search bar and select **CloudFormation**service.



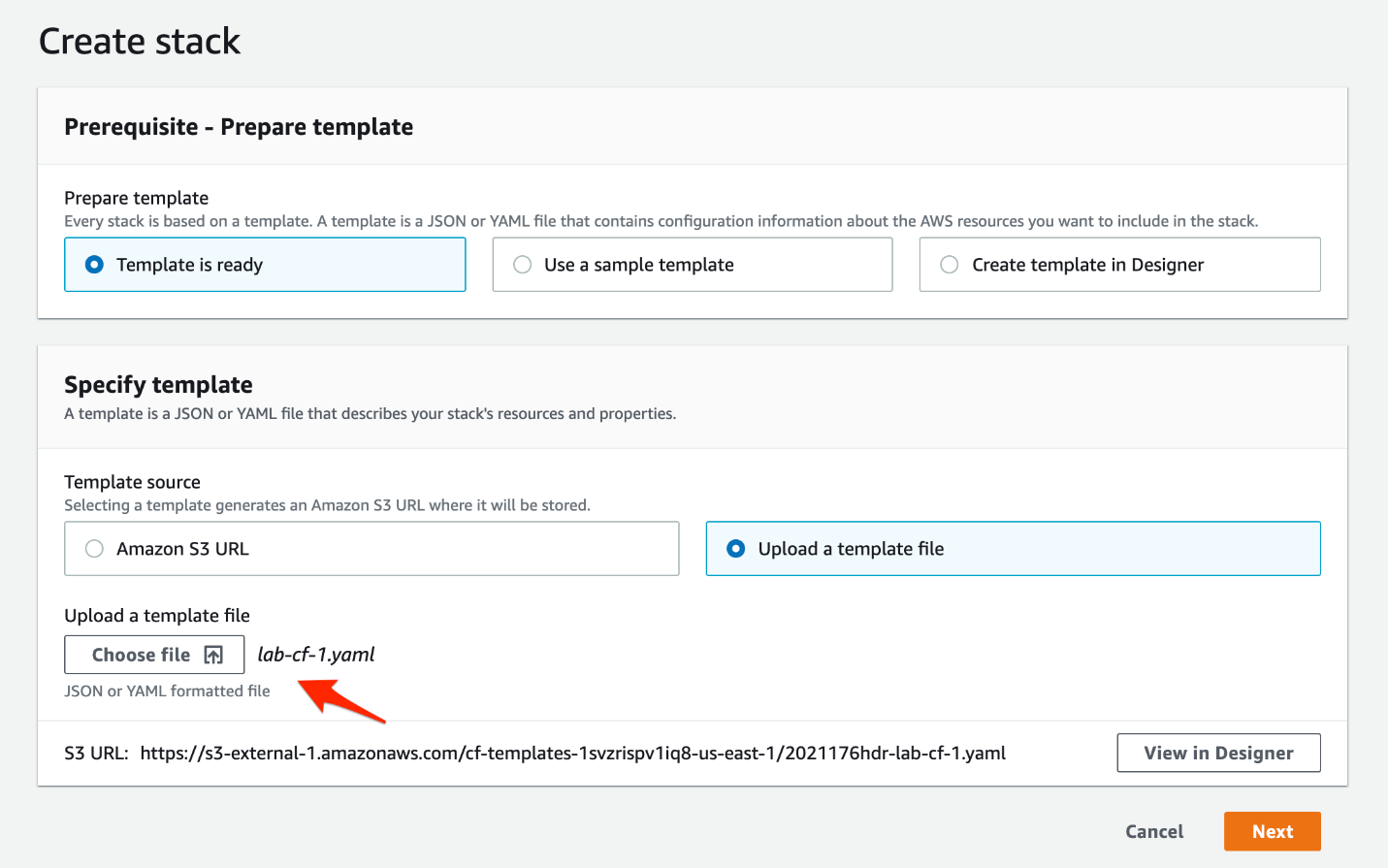
1. Click on the **Create stack** button.



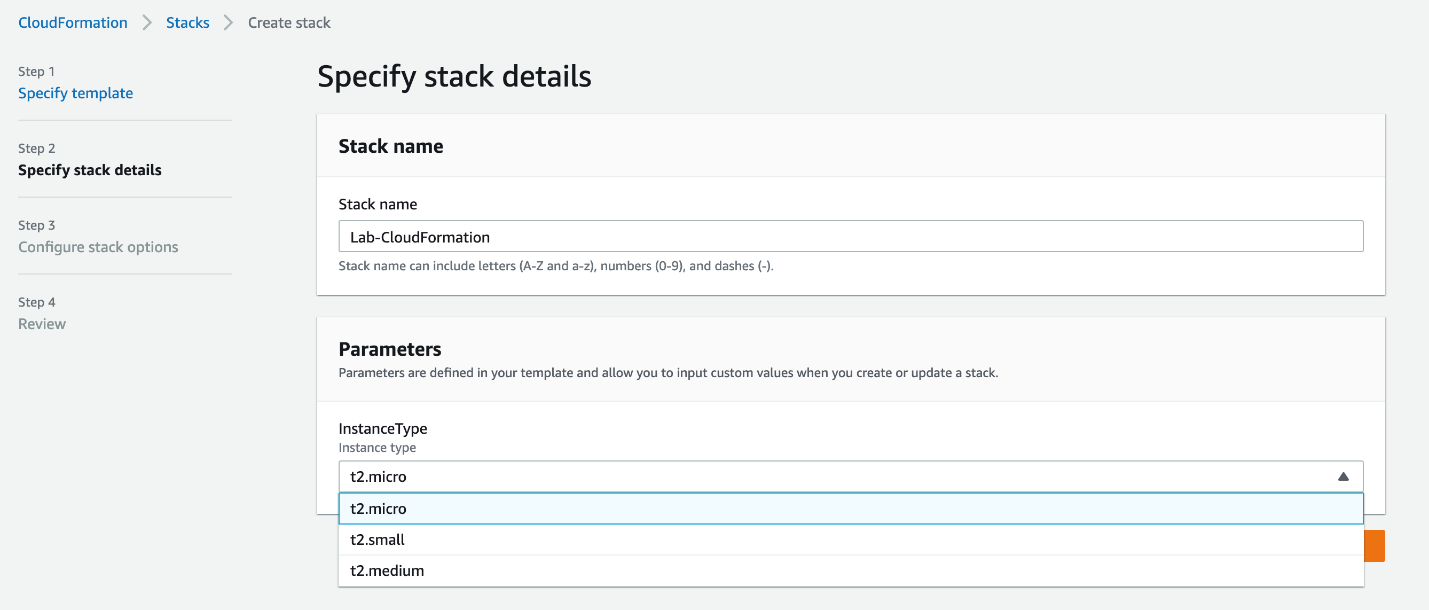
1. Select the following options to create the stack:
   1. Choose the option **Template is ready** for preparing the template. Specify the template by choosing the option **Upload a template file**and then click on the **Choose file** button. You should then upload the file **lab-cf-1.yaml**provided to you in the Resources for the project under Assets.



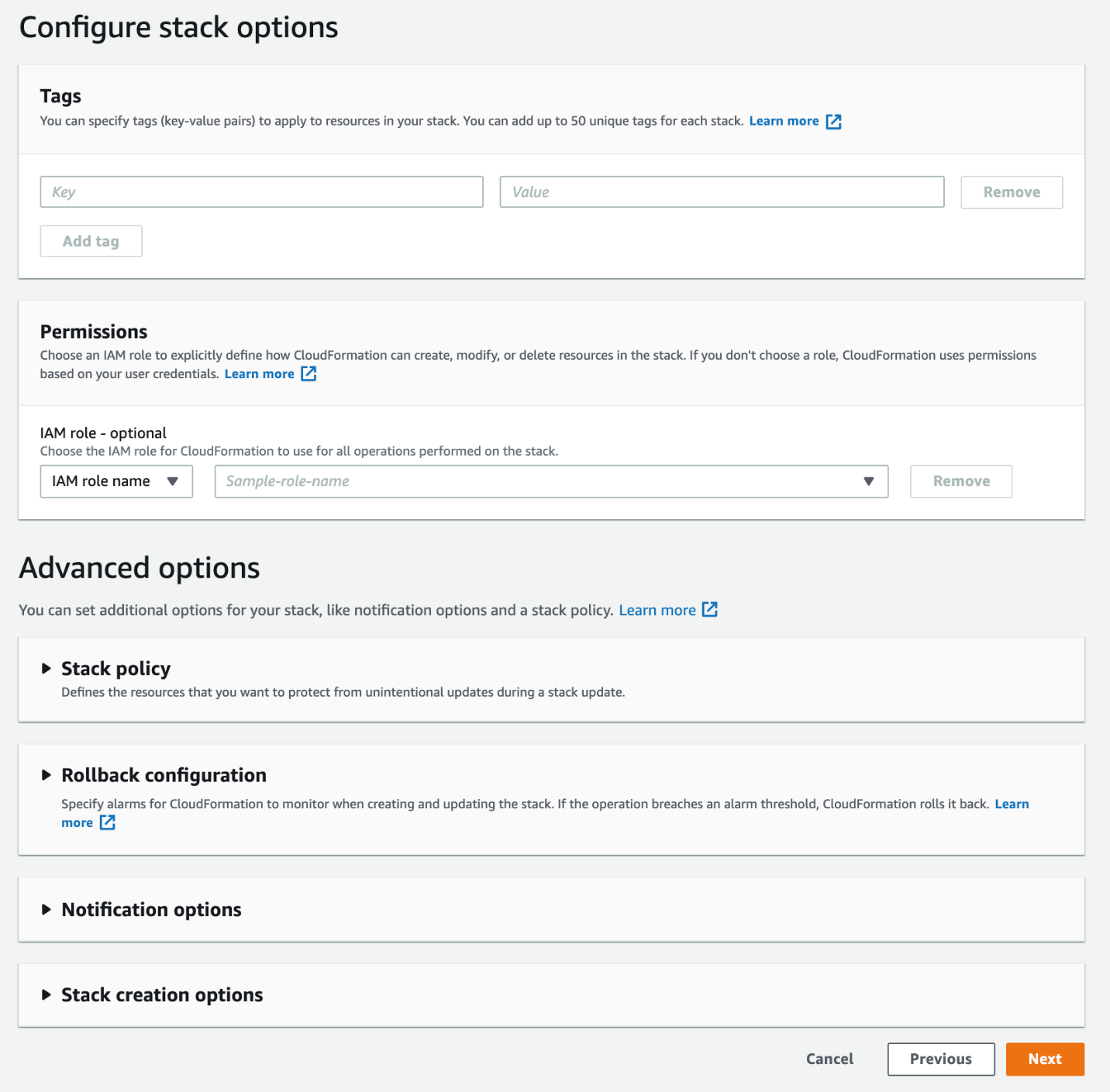
Then click on the **Next** button.



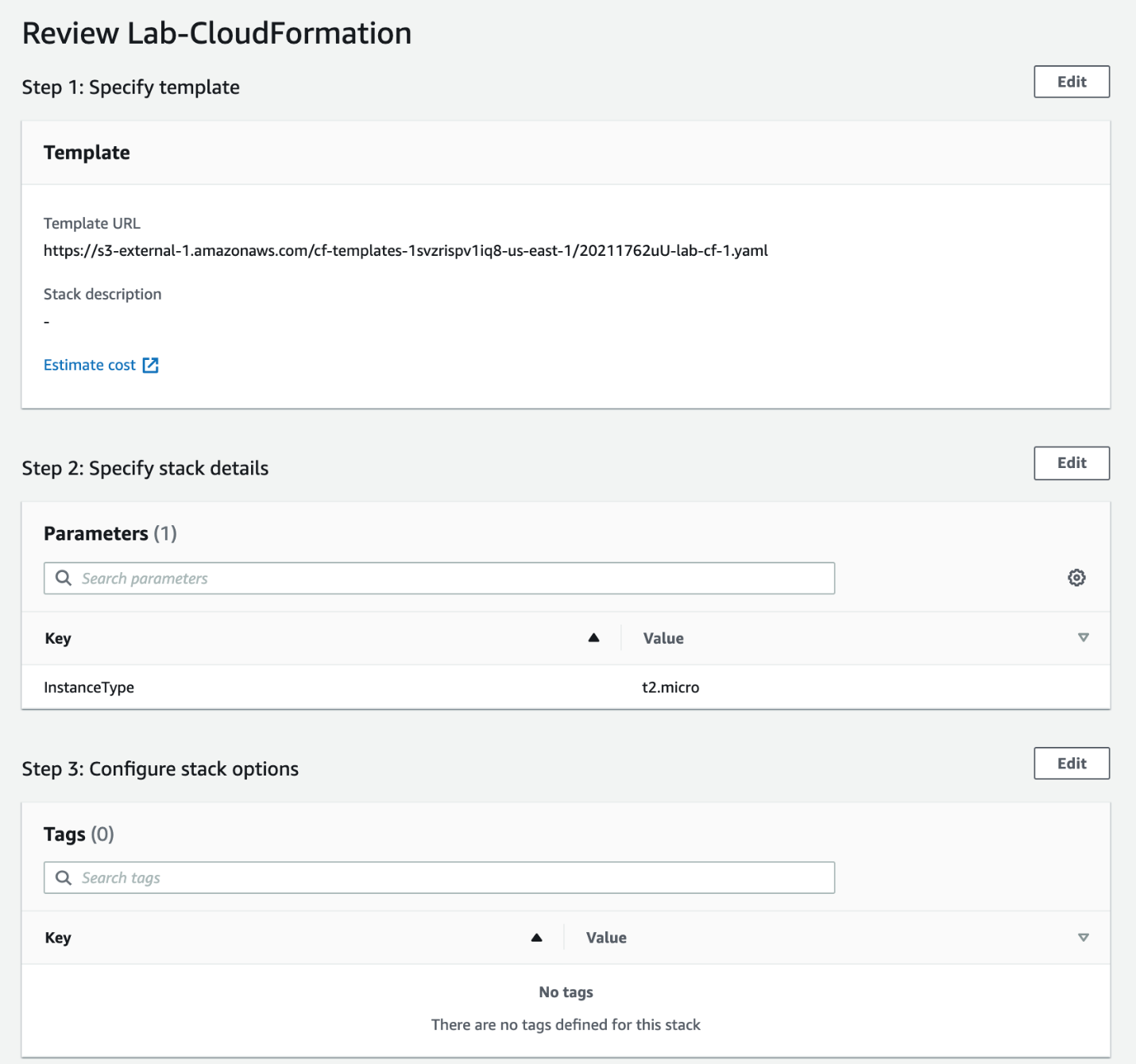
* 1. Specify the stack details by entering the **Stack name** as **Lab-CloudFormation**. Select the **InstanceType** parameter as t2.micro and click on the **Next** button.

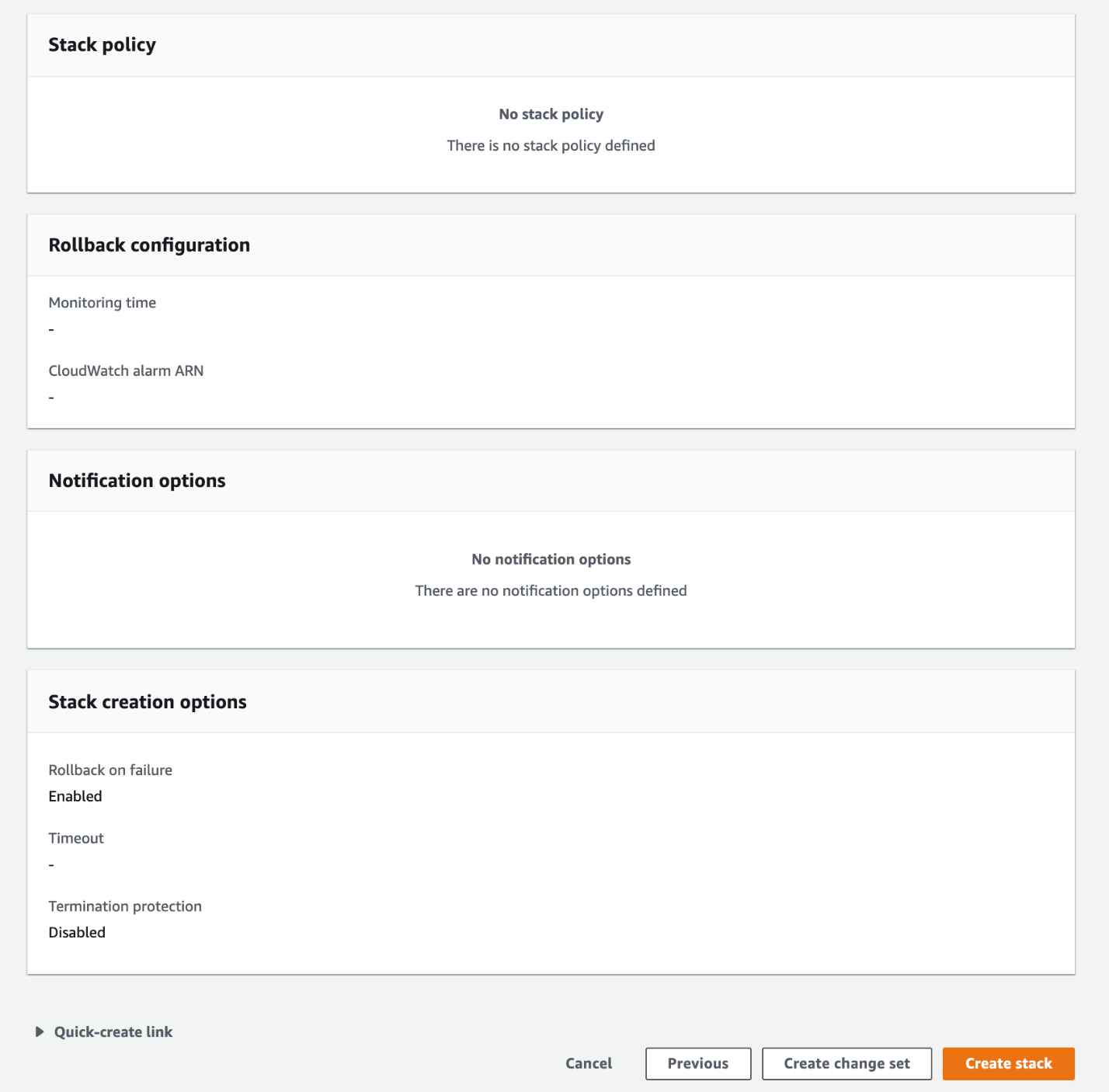


* 1. Leave all values unchanged for the**Configure stack options** section. Click on the **Next** button.

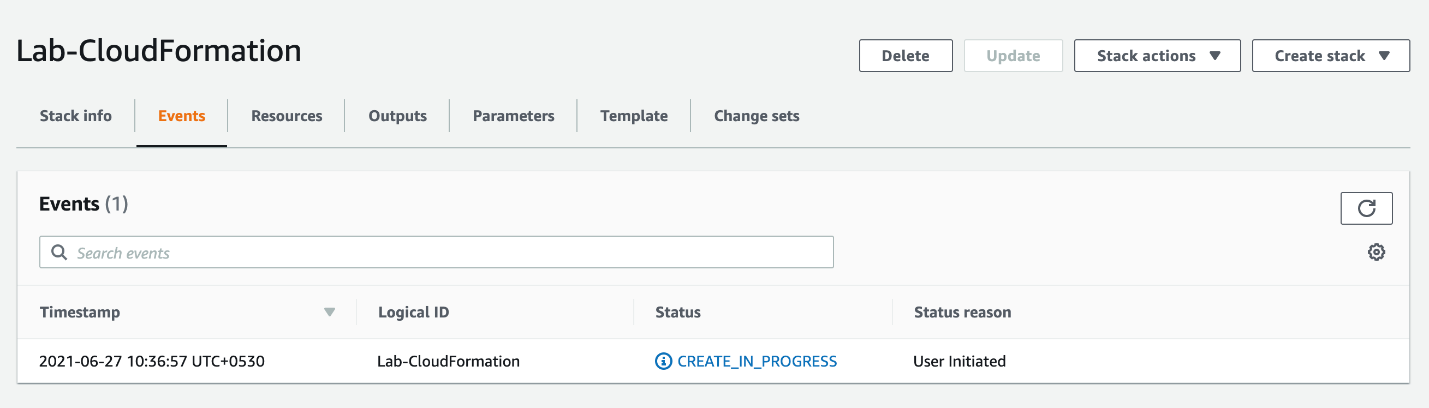


* 1. Review all sections for your stack **Lab-CloudFormation** and then click on the **Create stack** button.

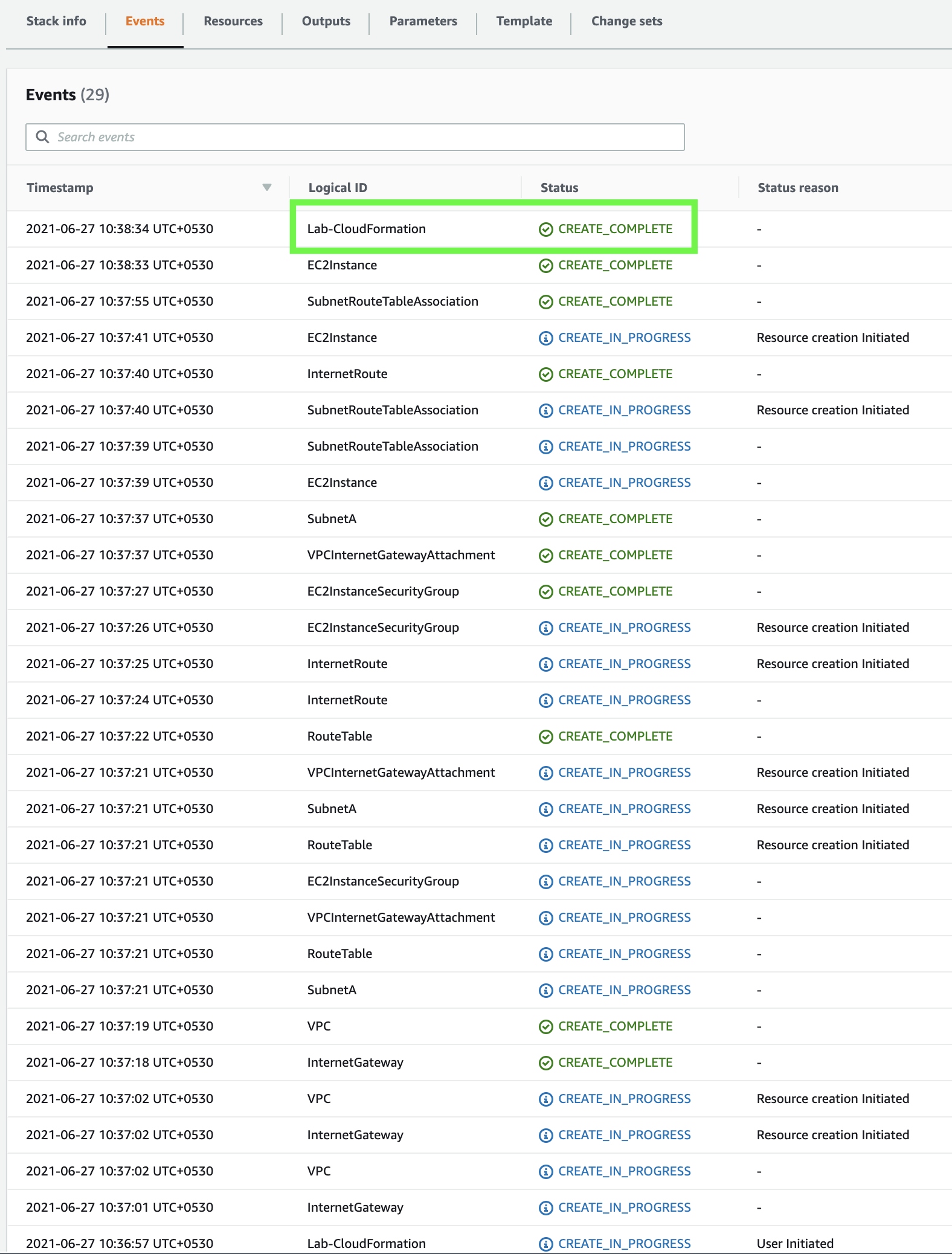


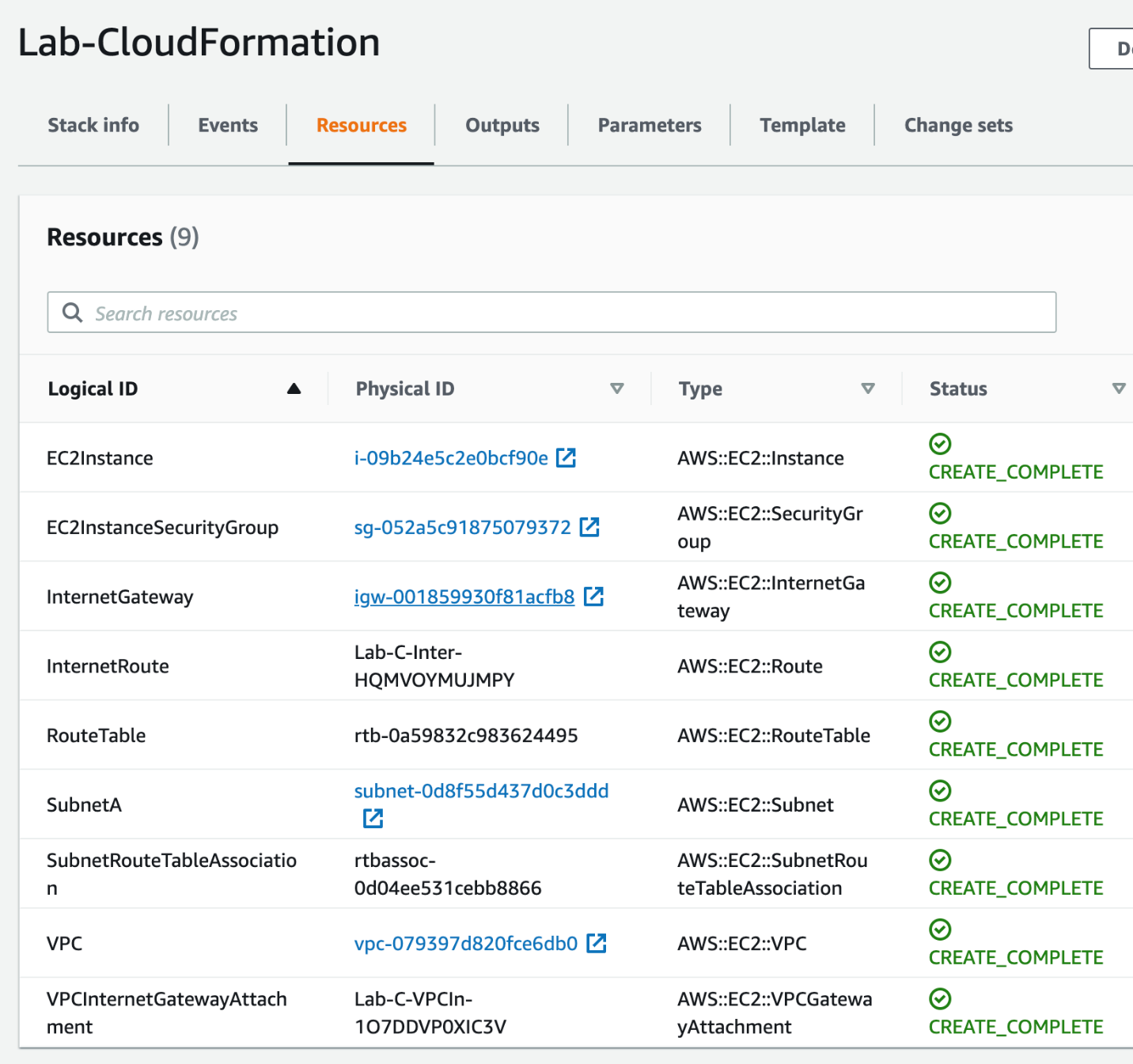


* 1. You should see an event with the **CREATE\_IN\_PROGRESS** status in the **Events** tab.



Within a few minutes, you should see that all resources have been provisioned successfully.





Yaml file

AWSTemplateFormatVersion: "2010-09-09"

Mappings:

AMIMap:

t2.micro:

AMI: "ami-0ab4d1e9cf9a1215a"

t2.small:

AMI: "ami-00068cd7555f543d5"

t2.medium:

AMI: "ami-0054bd45a019119d6"

Parameters:

InstanceType:

Description: Instance type

Default: t2.micro

Type: String

AllowedValues: [t2.micro, t2.small, t2.medium]

ConstraintDescription: Must specify t2.micro, t2.small, t2.medium

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/24

Tags:

- Key: Name

Value: Lab-CF-VPC

InternetGateway:

Type: AWS::EC2::InternetGateway

VPCInternetGatewayAttachment:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref VPC

InternetGatewayId: !Ref InternetGateway

SubnetA:

Type: AWS::EC2::Subnet

Properties:

AvailabilityZone: us-east-1a

VpcId: !Ref VPC

CidrBlock: 10.0.0.0/26

MapPublicIpOnLaunch: true

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

InternetRoute:

Type: AWS::EC2::Route

DependsOn: InternetGateway

Properties:

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref InternetGateway

RouteTableId: !Ref RouteTable

SubnetRouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref RouteTable

SubnetId: !Ref SubnetA

EC2InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupName: "Instance Security Group"

GroupDescription: "HTTP traffic"

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

EC2Instance:

Type: AWS::EC2::Instance

Properties:

AvailabilityZone: us-east-1a

InstanceType: !Ref InstanceType

ImageId: !FindInMap [AMIMap, !Ref InstanceType, AMI]

SubnetId: !Ref SubnetA

SecurityGroupIds:

- Ref: EC2InstanceSecurityGroup

Tags:

- Key: Name

Value: Lab-CF-Instance

This yam file is also in notepad doc

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>

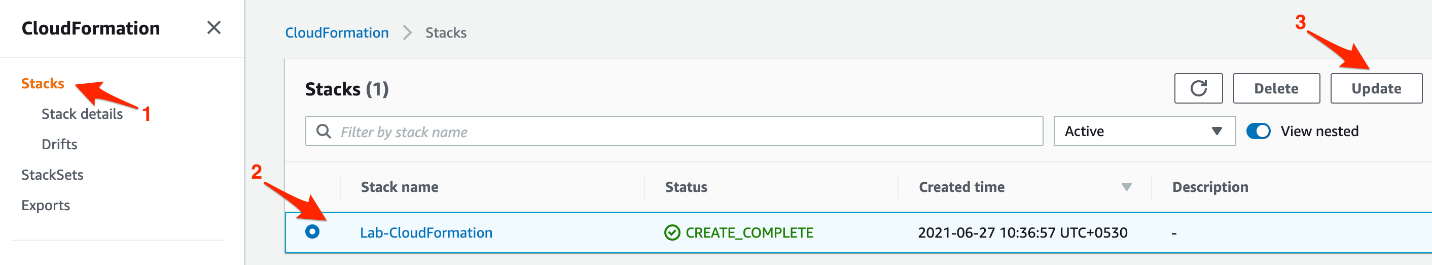
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/gettingstarted.templatebasics.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/parameters-section-structure.html>

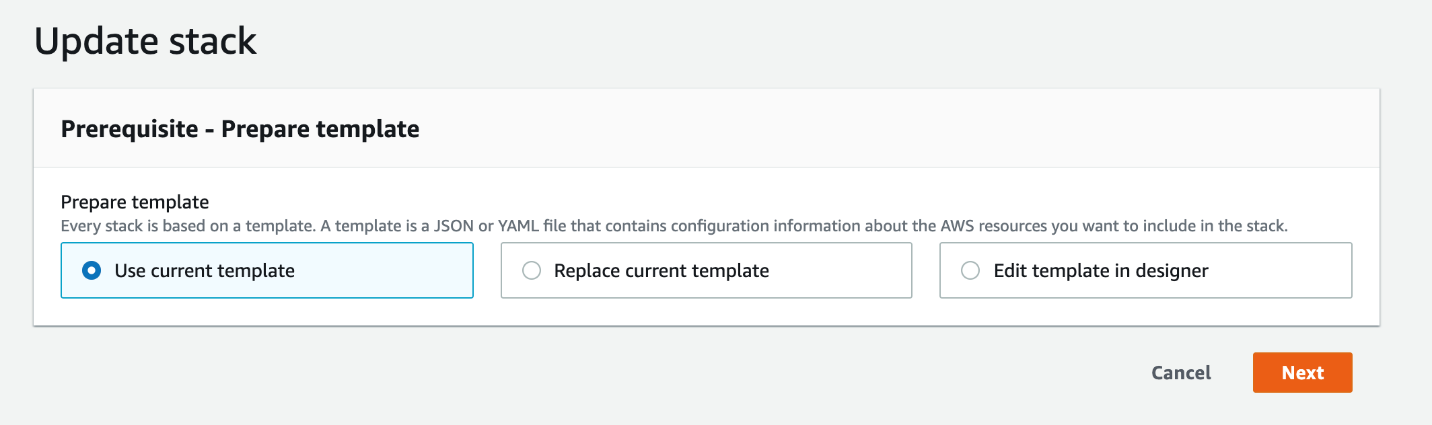
Update Ec2 by using the CF change sets

You will configure a CloudFormation change set by updating the CloudFormation template to set the type of the EC2 instance to t2.small. You then need to review the changes to your resources before executing the change set.

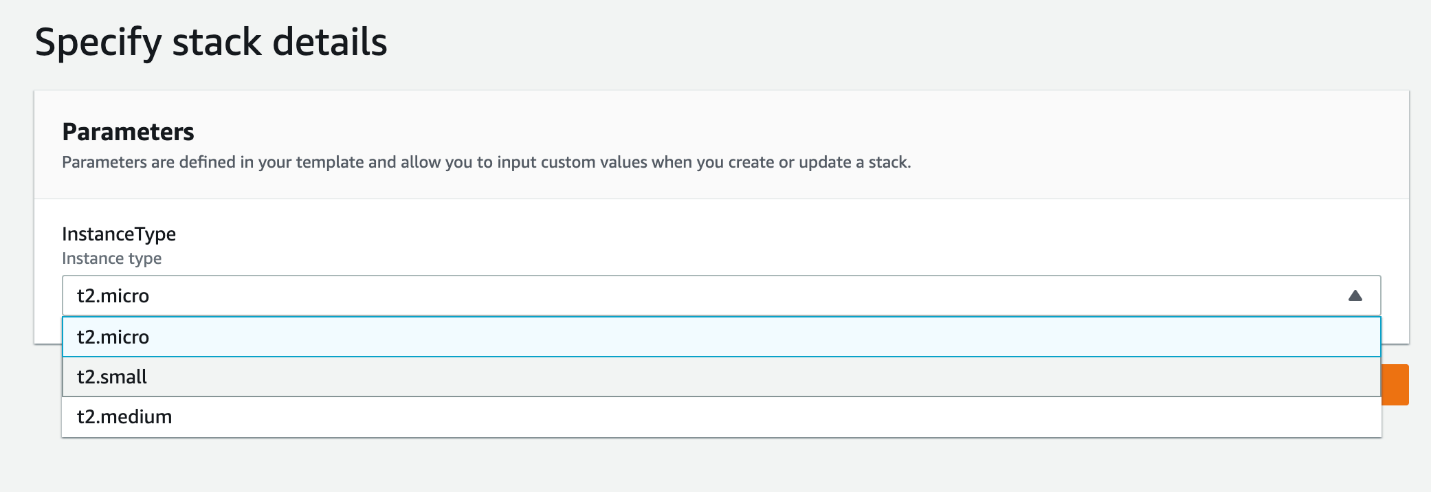
1. Navigate to the CloudFormation service. Click on the **Stacks** link in the left sidebar. Select the stack **Lab-CloudFormation** and then click on the **Update** button.



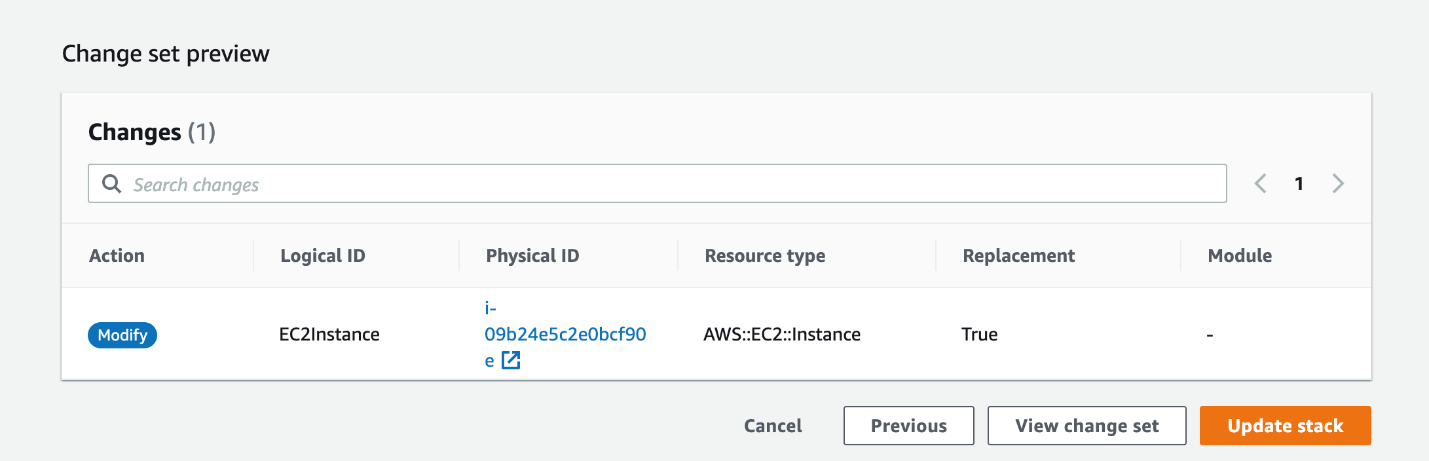
1. Select the option **Use current template** and click **Next**.



1. Specify stack details by selecting the parameter value for **InstanceType** as **t2.small**. Click on **Next**.



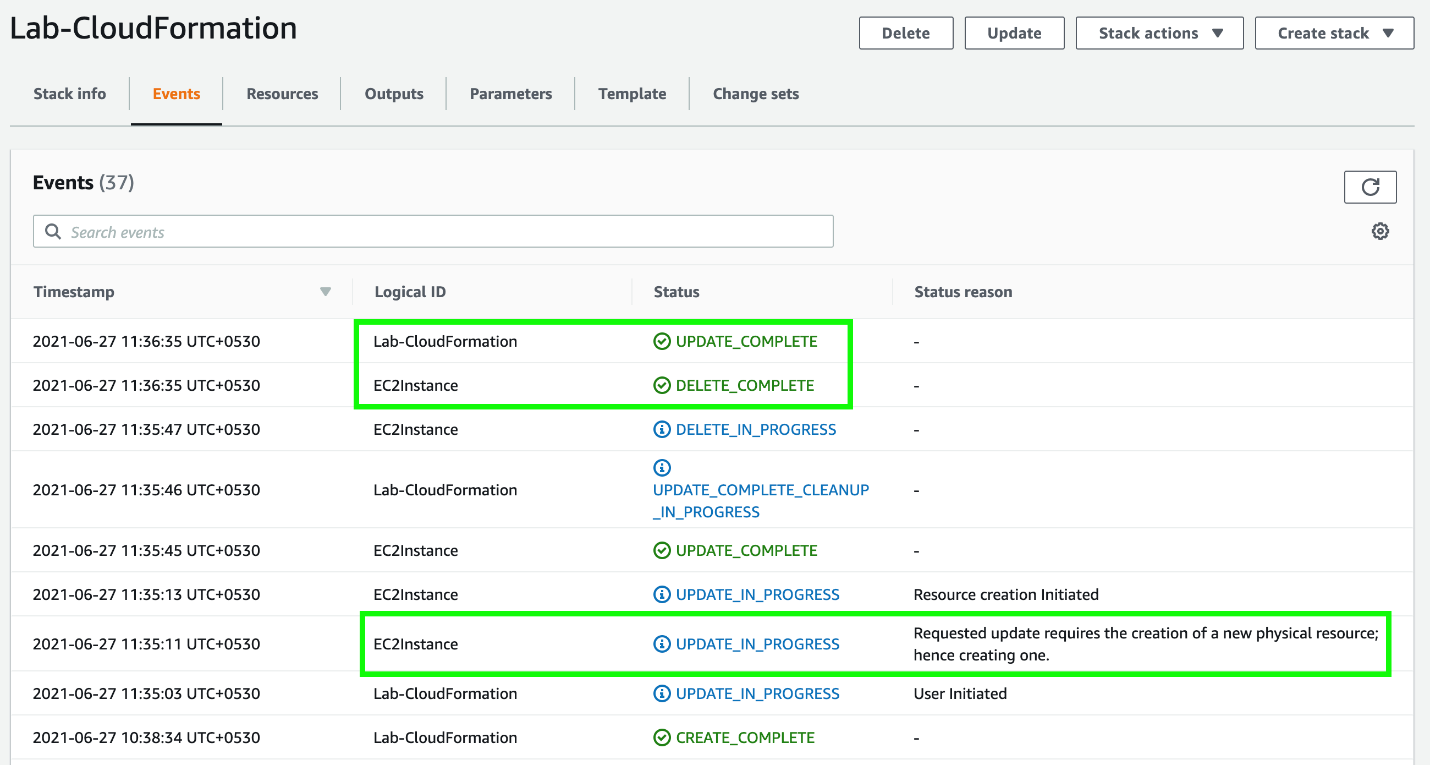
1. Leave all values unchanged for the**Configure stack options** section. Click on the **Next** button.
2. Scroll to the bottom of the Review section. Within a few moments you should see the **Change set preview** summary. This change set would execute a **Modify** action type for the EC2 instance type to be replaced from t2.micro to t2.small. Click on the **Update stack** button.



Within a few minutes, you should see that the new resources have been provisioned successfully. You should note that this new t2.small instance type is based on another AMI ID as shown below:



Therefore, you will see a new EC2 instance being created followed by the old EC2 instance getting deleted.

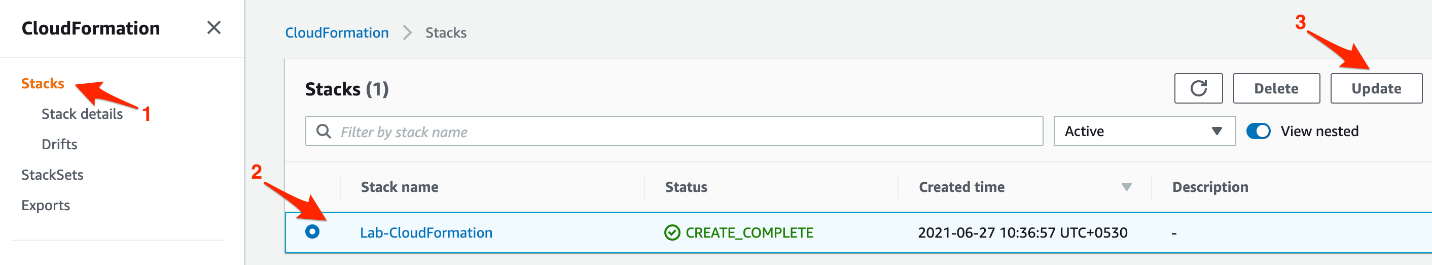


<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html>

Update SG using CF change sets

You will configure a CloudFormation change set by updating the template to have the security group also allow HTTPS traffic on port 443. You then need to review the changes to your resources before executing the change set.

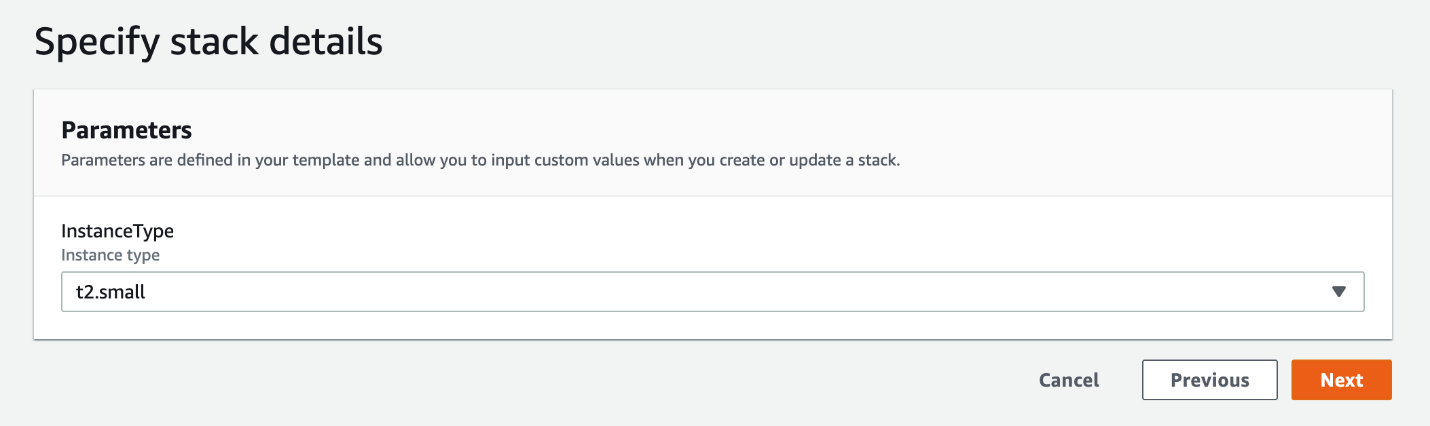
1. Navigate to the CloudFormation service. Click on the **Stacks** link in the left sidebar. Select the stack **Lab-CloudFormation** and then click on the **Update** button.



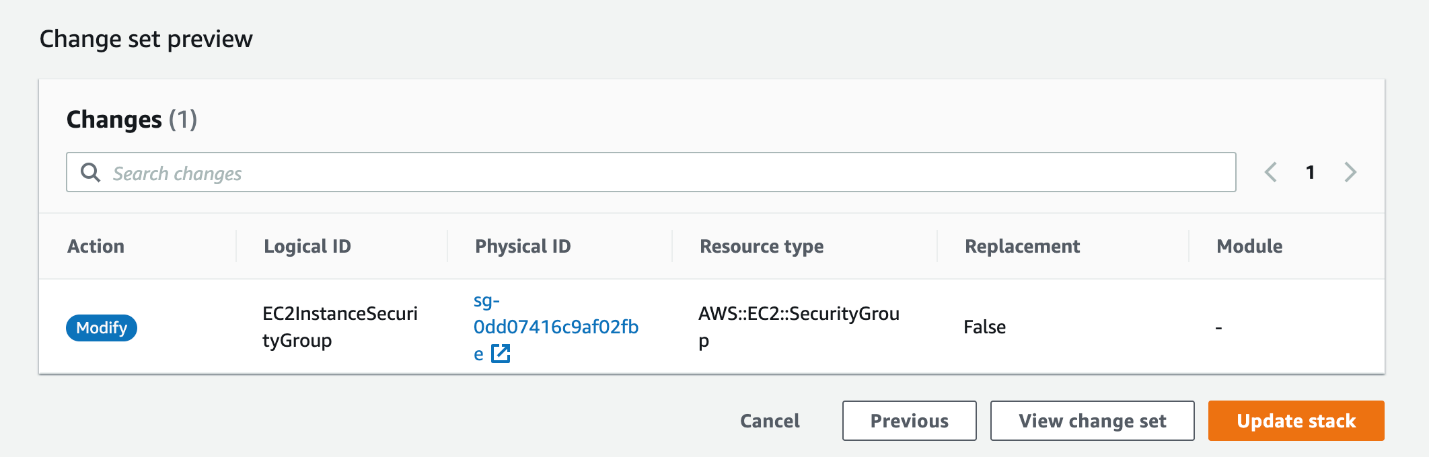
1. Click on the option **Replace current template**.Specify the template by choosing the option **Upload a template file**and then click on the **Choose file** button. You should then upload the file **lab-cf-2.yaml**provided for you in the Resources under Assets. This template has the additional Security Group ingress rule to allow HTTPS traffic on port 443. Click on the Next button.



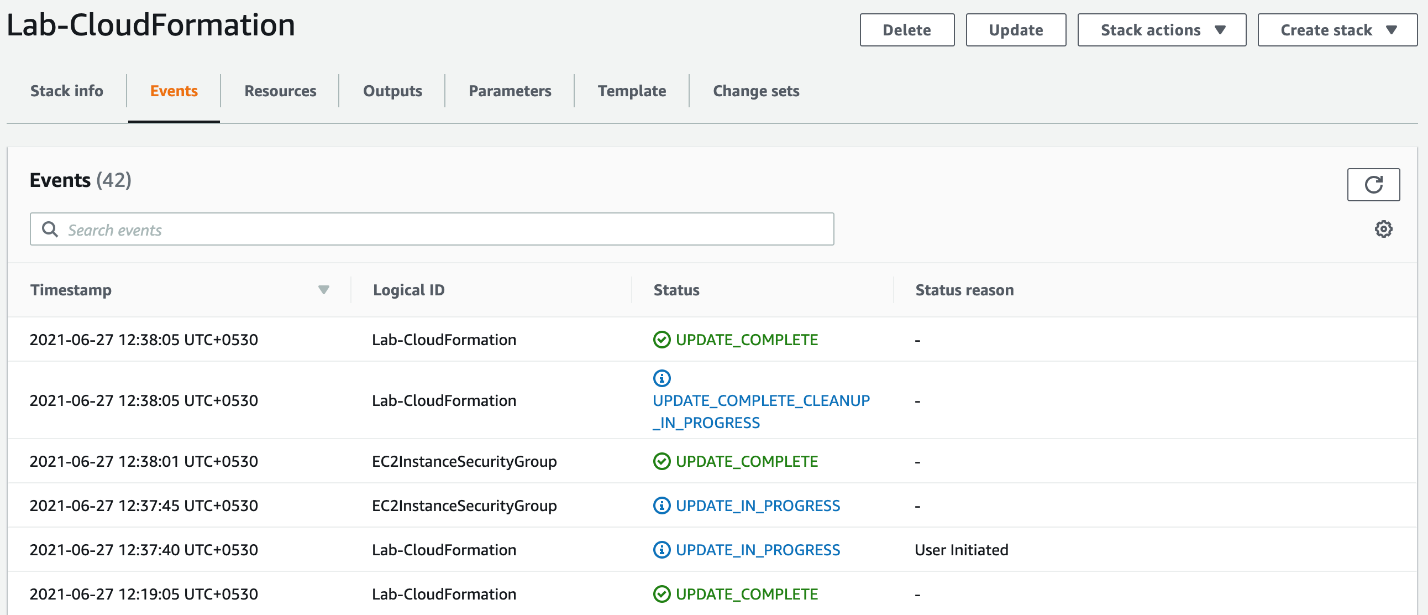
1. Leave the values for the **Specify stack details** section as unchanged. Click on **Next**.



1. Leave all values for the **Configure stack options** section as unchanged. Click on **Next**.
2. Scroll to the bottom of the Review section. Within a few moments you should see the **Change set preview** summary. This change set would execute a **Modify** action type for the EC2 instance security group. Click on the **Update stack** button.



You should see that the security group has been updated successfully.



Ref yaml file in note pad doc

AWSTemplateFormatVersion: "2010-09-09"

Mappings:

AMIMap:

t2.micro:

AMI: "ami-0ab4d1e9cf9a1215a"

t2.small:

AMI: "ami-00068cd7555f543d5"

t2.medium:

AMI: "ami-0054bd45a019119d6"

Parameters:

InstanceType:

Description: Instance type

Default: t2.small

Type: String

AllowedValues: [t2.micro, t2.small, t2.medium]

ConstraintDescription: Must specify t2.micro, t2.small, t2.medium

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: 10.0.0.0/24

Tags:

- Key: Name

Value: Lab-CF-VPC

InternetGateway:

Type: AWS::EC2::InternetGateway

VPCInternetGatewayAttachment:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

VpcId: !Ref VPC

InternetGatewayId: !Ref InternetGateway

SubnetA:

Type: AWS::EC2::Subnet

Properties:

AvailabilityZone: us-east-1a

VpcId: !Ref VPC

CidrBlock: 10.0.0.0/26

MapPublicIpOnLaunch: true

RouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

InternetRoute:

Type: AWS::EC2::Route

DependsOn: InternetGateway

Properties:

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref InternetGateway

RouteTableId: !Ref RouteTable

SubnetRouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref RouteTable

SubnetId: !Ref SubnetA

EC2InstanceSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupName: "Instance Security Group"

GroupDescription: "HTTP traffic"

VpcId: !Ref VPC

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 443

ToPort: 443

CidrIp: 0.0.0.0/0

EC2Instance:

Type: AWS::EC2::Instance

Properties:

AvailabilityZone: us-east-1a

InstanceType: !Ref InstanceType

ImageId: !FindInMap [AMIMap, !Ref InstanceType, AMI]

SubnetId: !Ref SubnetA

SecurityGroupIds:

- Ref: EC2InstanceSecurityGroup

Tags:

- Key: Name

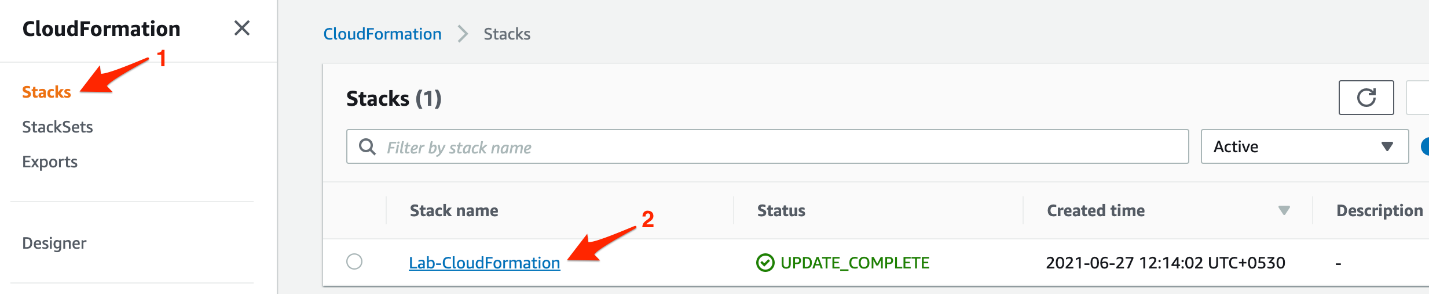
Value: Lab-CF-Instance

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/using-cfn-updating-stacks-changesets.html>

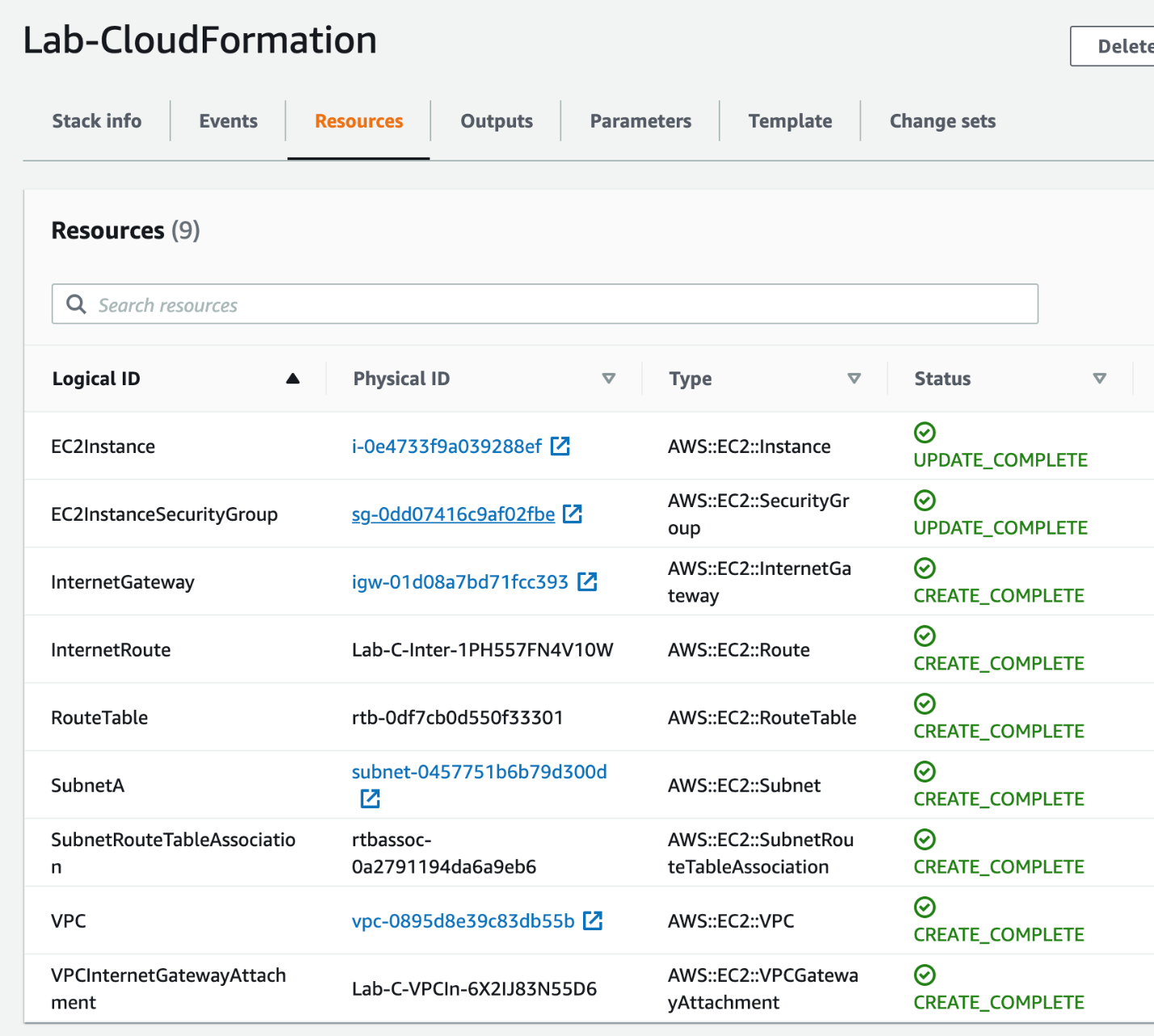
Inspect the resource created by CF

You will use the AWS management console to inspect the resources created by CloudFormation - VPC, EC2 instance, security group, internet gateway, and route table.

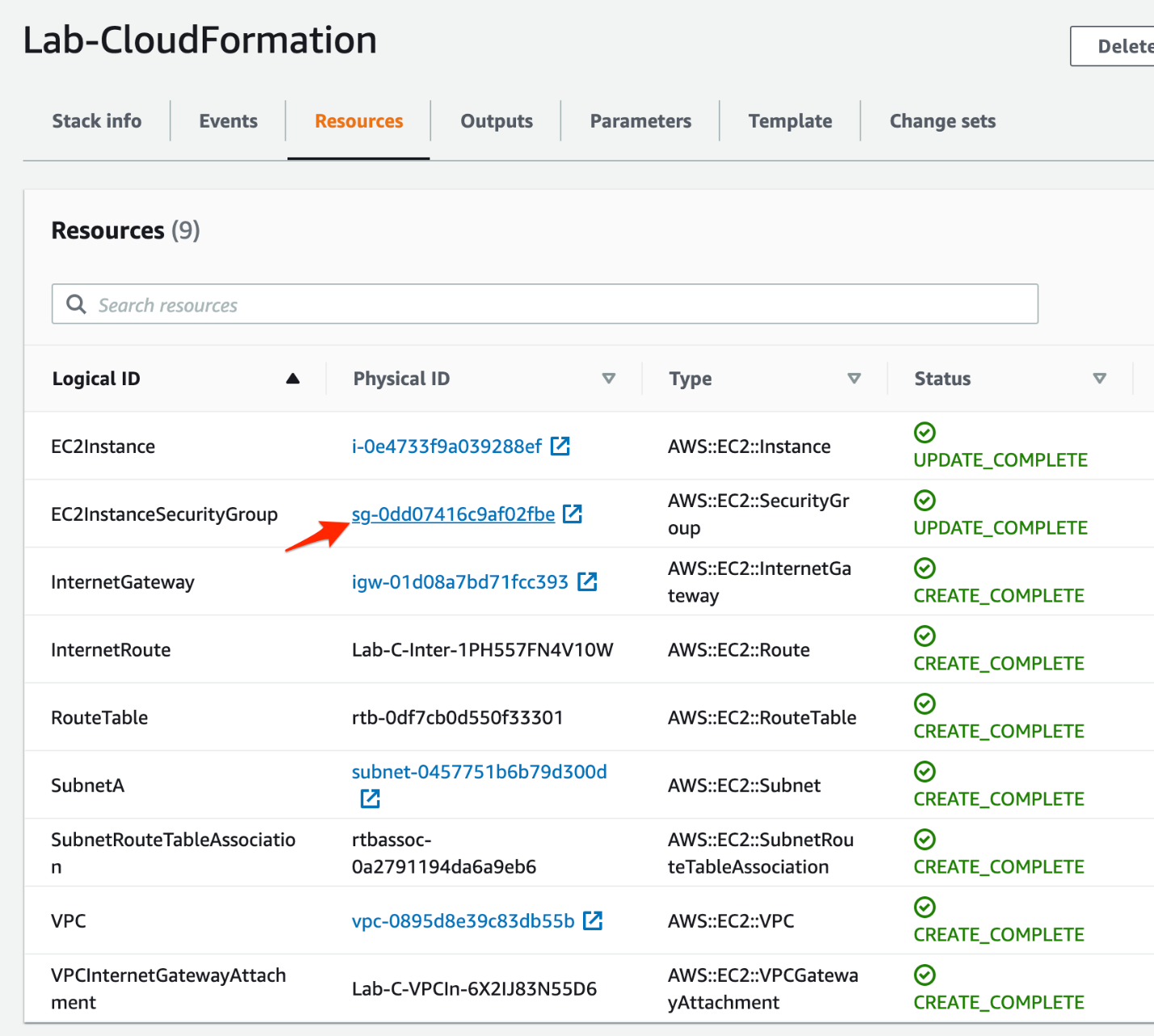
1. Navigate to the CloudFormation service. Click on the **Stacks** link in the left sidebar. Click on the stack **Lab-CloudFormation**.



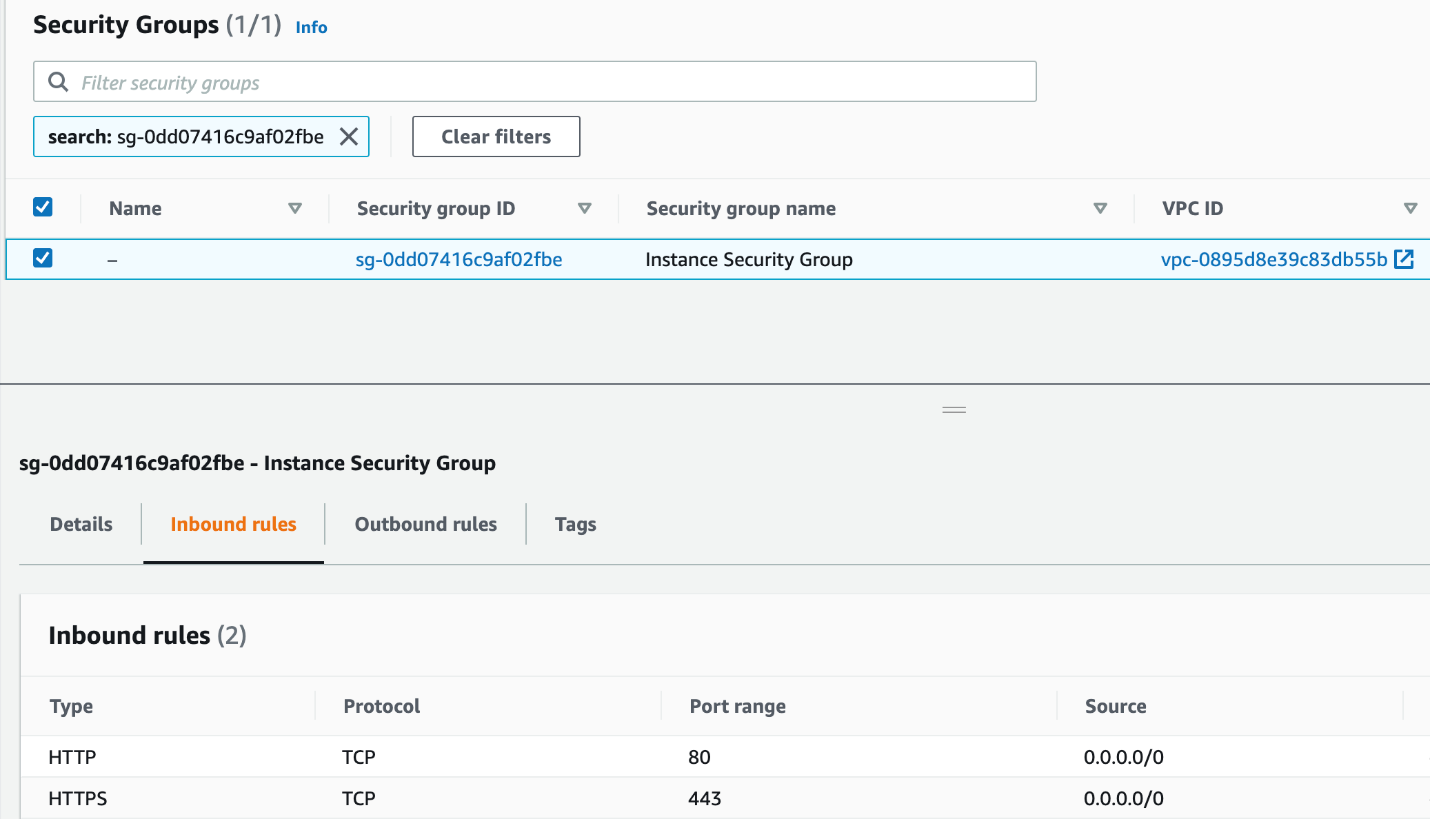
1. Select the resources tab. You should see all resources created by the stack **Lab-CloudFormation**.



1. Click on the link for the Security group - **EC2InstanceSecurityGroup**.



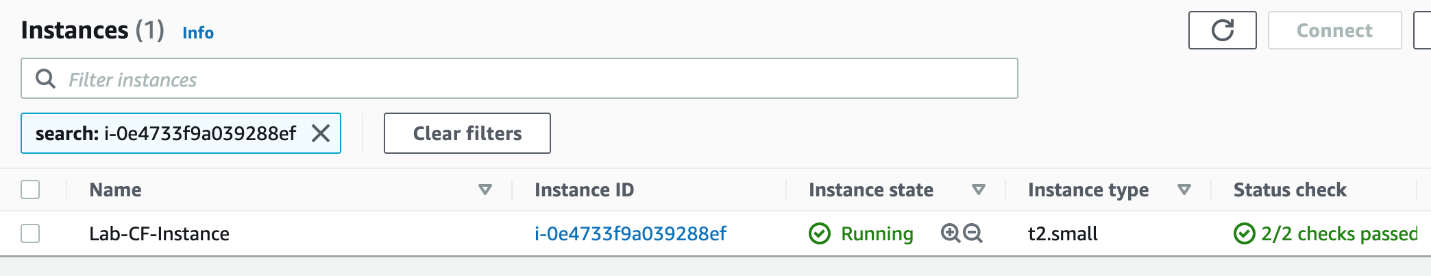
1. Click on the tab for the **Inbound rules**for the security group. You should see that the security group has inbound rules for HTTP traffic on port 80 as well as HTTPS traffic on port 443.



1. Navigate to the resources tab of the stack **Lab-CloudFormation**. Click on the link for the EC2 Instance.



You should see that the EC2 instance type is **t2.small**, which was changed from t2.micro to t2.small, as part of a CloudFormation change set executed earlier.



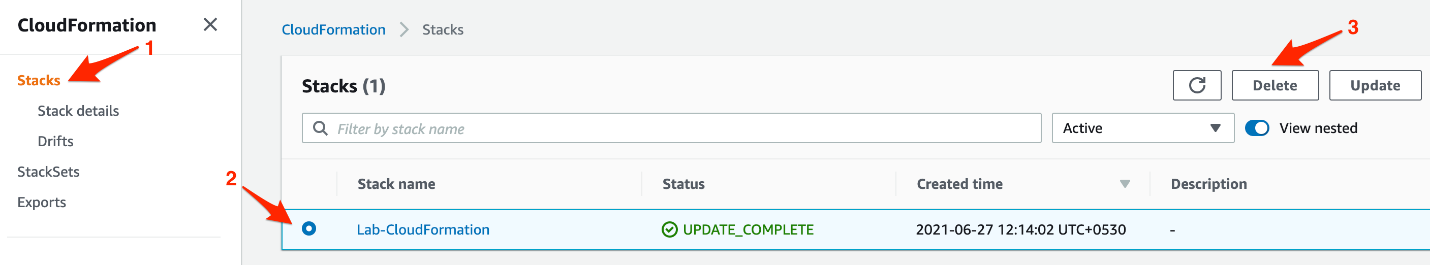
1. Navigate to the resources tab of the stack **Lab-CloudFormation**. Inspect the VPC, Internet Gateway and Subnet resources by following the respective links.

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/resources-section-structure.html>

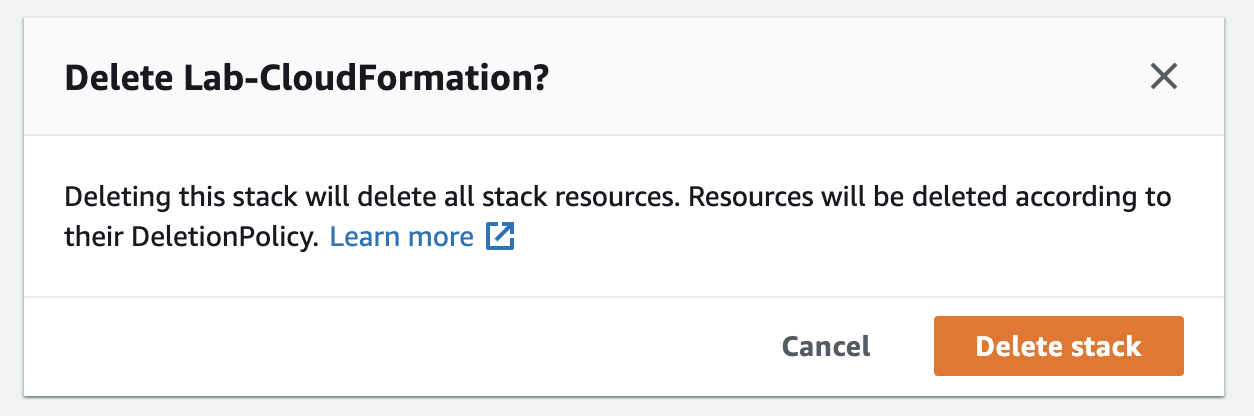
Clean Up

You created a custom VPC, an EC2 instance, and a security group via the CloudFormation template. You need to delete the CloudFormation stack to clean up the resources.

1. Navigate to the CloudFormation service. Click on the **Stacks** link in the left sidebar. Select the stack **Lab-CloudFormation** and then click on the **Delete** button.



Click on the **Delete stack** button to confirm that you want to delete the stack.



# Automate Load Balancer and VPC creation using CloudFormation

Created by[Rick Crisci](https://deloittedevelopment.udemy.com/user/rick-crisci/)

**AWS Workspace**

**90-180 minutes**

In this lab, you will be a solutions architect who must find efficient methods to create multiple EC2 deployments that are load balanced. You are inheriting a set of CloudFormation templates that do not work properly. You must resolve the problems with the CloudFormation templates and verify successful deployment of the necessary resources.

A new version of E-Commerce application is slated to be released next month. We need to maintain identical Test, Dev, and Production environments for this application. A consultant has attempted to create CloudFormation templates to automate the creation of these environments, but they are not working properly. I'd like you to review and correct the CloudFormation templates, and verify that they create the required resources.

The first template is designed to create the network environment for the application. It must create a VPC with two subnets in different AZs. Both subnets should be public and have a default route that allows traffic to the Internet.

The second template deploys an AutoScaling group of web servers behind a load balancer. Instances must be spread across AZs and be reachable from the Internet over HTTP port 80. The instances must self-launch and run a script to create a simple website.

TASK

1-Prepare AWS account for a new custom VPC

2-Review VPC Cloudformation Template

3-Troubleshoot Cloudformation template#1

4-Verify the results of the cloudformation template

5-Troubleshoot webApp cloudforation template

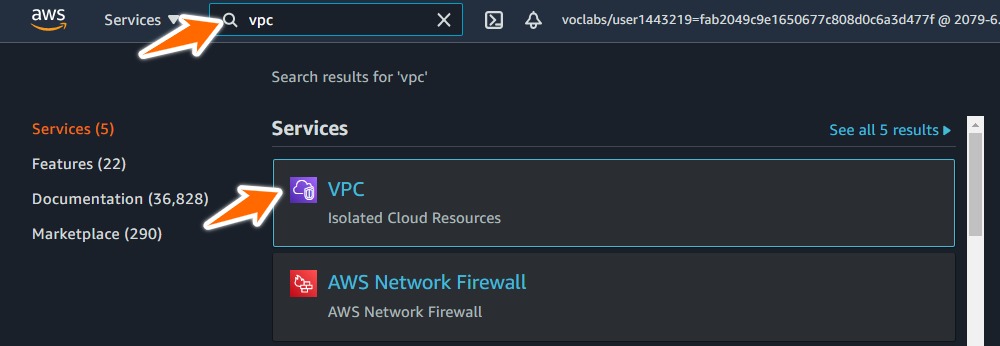
6-verify cloudformation template#2

7-Clean up

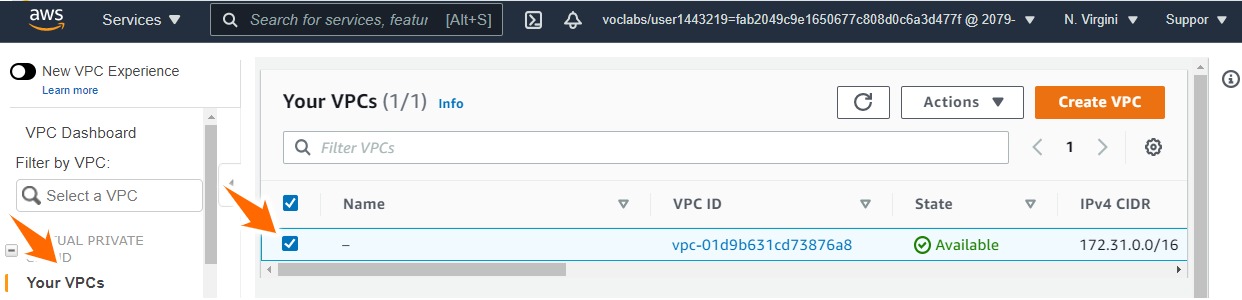
Prepare AWS account for a new custom VPC

You need to remove the default VPC from this AWS account in the current region. Eventually you will need to create firewall rules to allow your workstation to access EC2 instances that you will create, so you must find your public IP address.

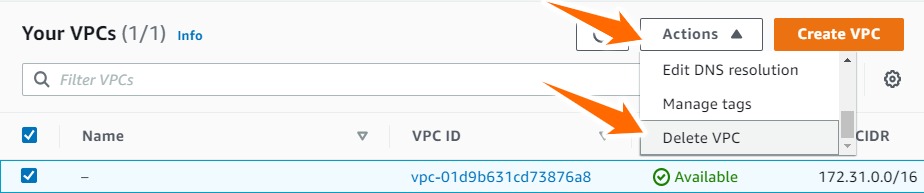
1. Launch the AWS Console and browse to the VPC console.



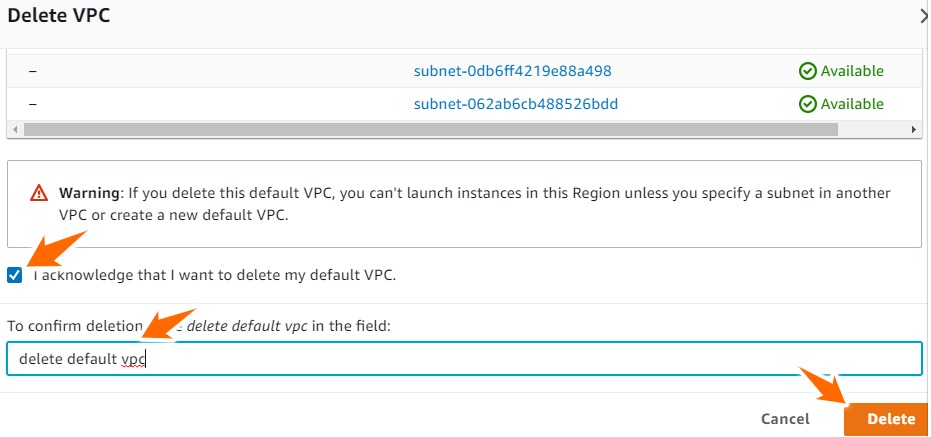
1. Click on **Your VPCs**and locate the default VPC. You can verify that it is the default VPC by scrolling to the right. Select the check box next to the default VPC.



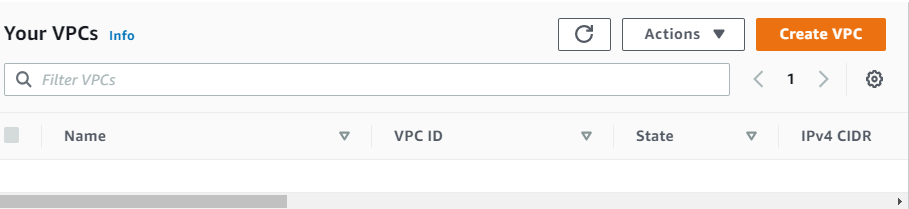
1. Click **Actions**and **Delete VPC**.



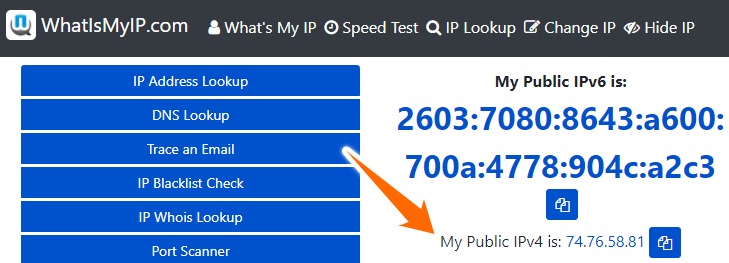
1. Acknowledge the warning and complete the deletion of the default VPC.



1. **Your VPCs** should now be empty.



1. Open a browser window from your workstation and go to [www.whatismyip.com](http://www.whatismyip.com/). Make a note of your public IPv4 address for future use.



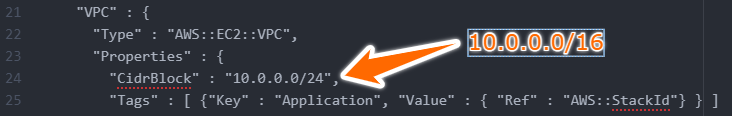
<https://docs.aws.amazon.com/vpc/latest/userguide/default-vpc.html>

Review VPC Cloudformation Template

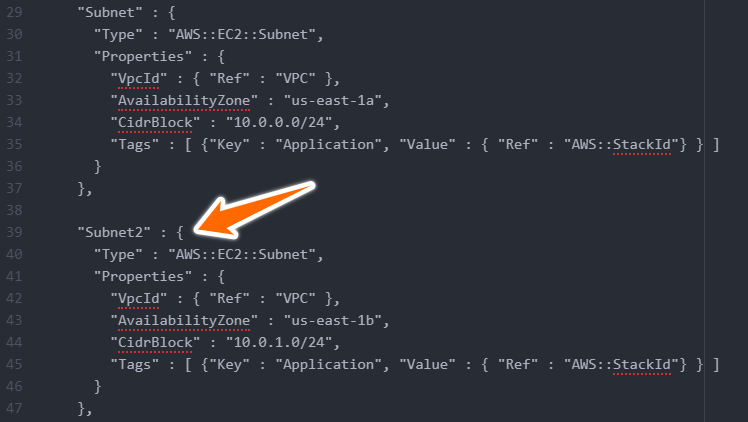
Validate that the template will create two subnets in different Availability Zones and add any necessary configuration to the CloudFormation Template.

* The VPC should be configured with the following CIDR block: 10.0.0.0/16.
* There should be two subnets in different AZs configured with the following CIDR blocks: 10.0.0.0/24 and 10.0.1.0/24.
* Interchangeable EC2 instances will be spread across these Availability Zones, and therefore the subnets should have the same routing and security configurations.
* Add any necessary configuration to the CloudFormation Template.

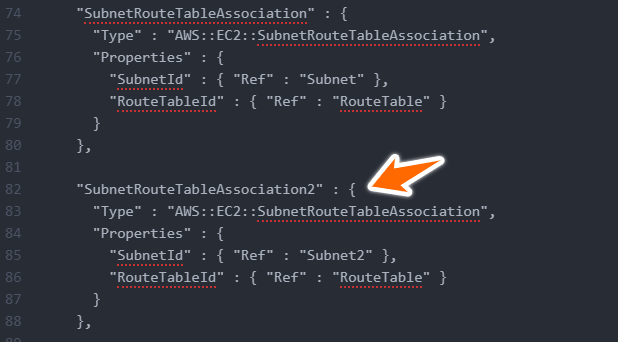
1. Choose a text editor. I recommend Atom. You can use the provided link to download it. <https://atom.io/>
2. Download the VPC CloudFormation Template provided and open it with your text editor.
3. Modify the VPC CIDR Range.



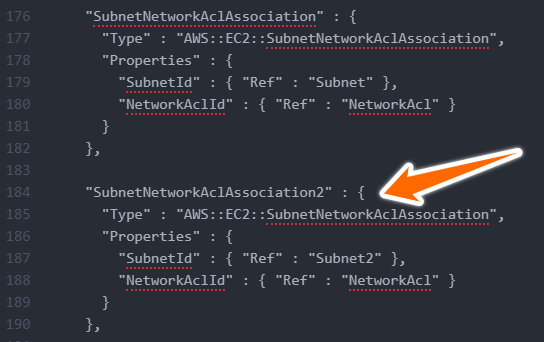
1. Add Subnet2 to the CF Template.



1. Add route table association for Subnet2



1. Add Network ACL Association for Subnet2



1. Save the completed template as **VPC CloudFormation Template - Corrected.txt** on your computer.

Ref VPC CF Template note pad

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-subnet.html>

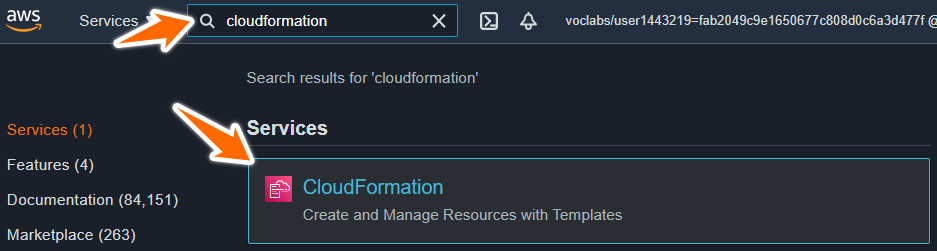
<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-subnet-network-acl-assoc.html>

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/Welcome.html>

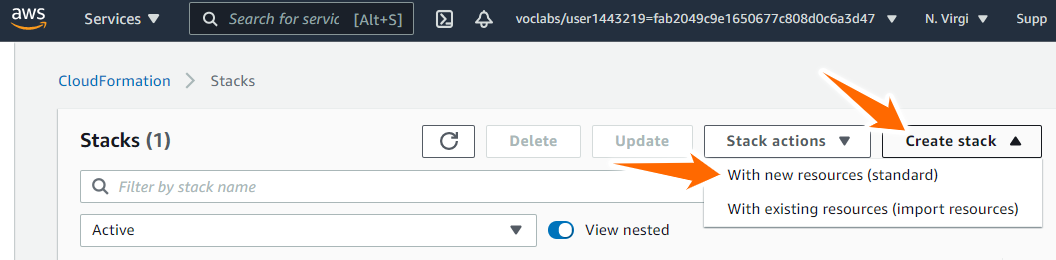
Troubleshoot Clodformation Template#1

Attempt to execute CloudFormation Template and resolve any errors until it successfully completes.

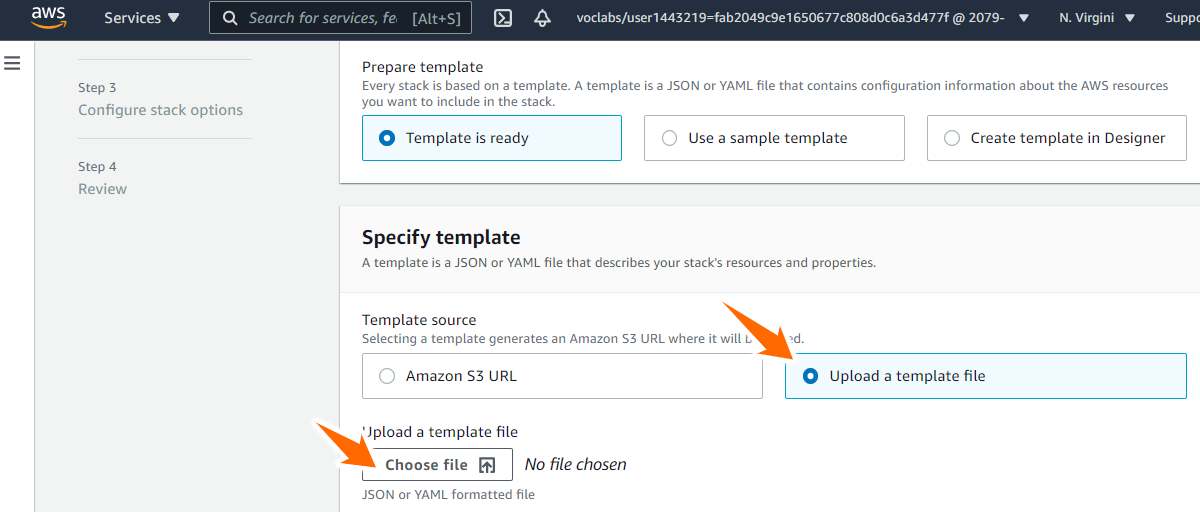
1. At this point your CloudFormation template should have the correct CIDR range and subnet configurations. If you wish to verify your template, you can use **VPC CloudFormation Template Solution 1.txt** from the resources.
2. From the AWS Console, browse to the CloudFormation console.



1. Create a stack of new resources.



1. Click on "Choose file" and browse to "CloudFormation Template for Task 2 Solution"



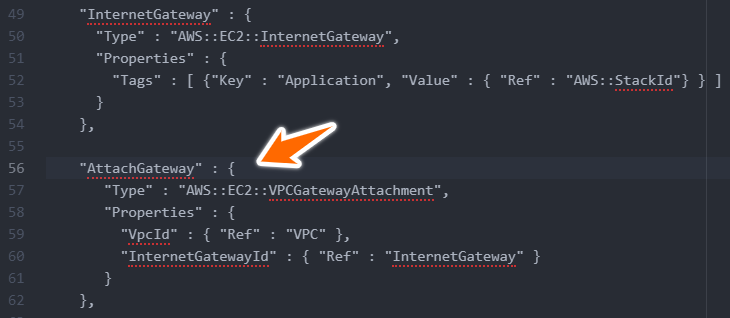
1. When you click next you will see an error like this:



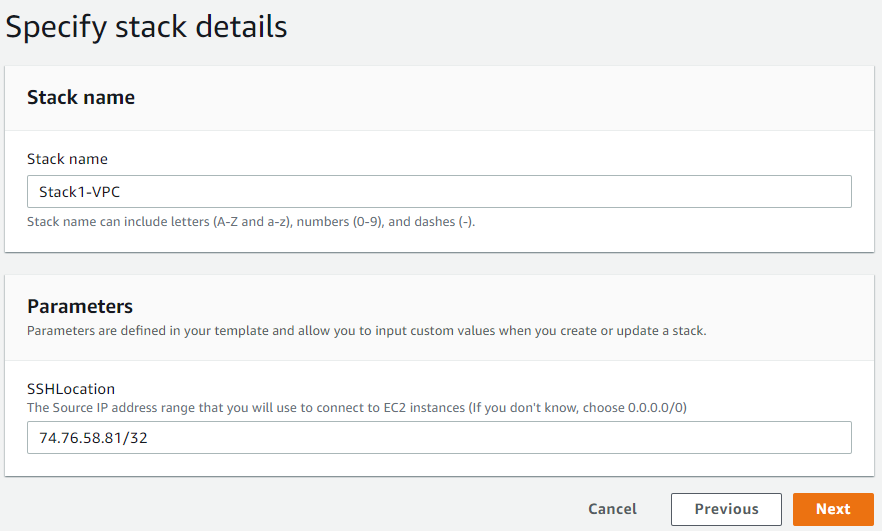
1. There is a comma missing from the line prior to the line mentioned in the error message. Resolve that, save the template, and then click on "Choose file" and try again.
2. When you click next you will see an error like this:



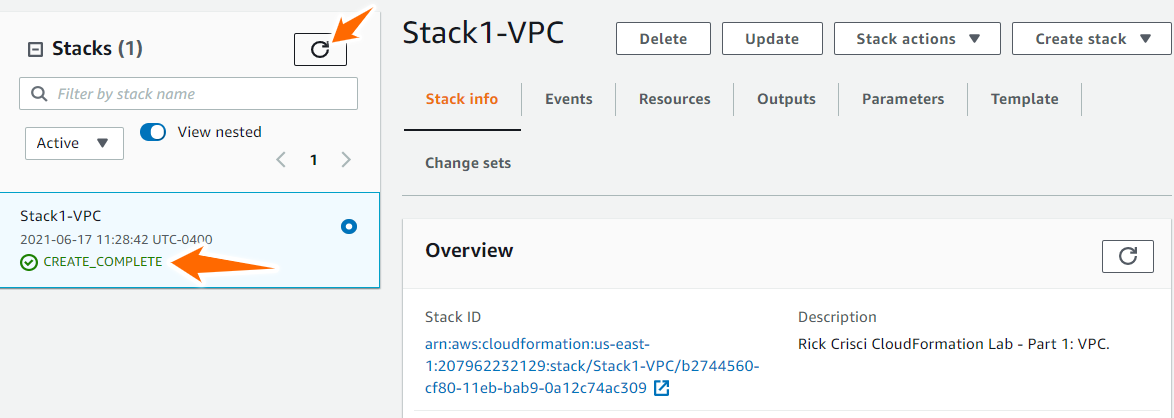
1. The Internet Gateway is not currently attached to the VPC. Create an Internet Gateway attachment in the template. Save the template, and then click on "Choose file" and try again.



1. Name the stack "Stack1-VPC" and input your public IP address as the SSHLocation, then click Next.



1. Accept the defaults and keep clicking "Next", then click "Create Stack"
2. Monitor the stack creation. Use the refresh button to update the status. When you see CREATE\_COMPLETE the stack has been successfully created.



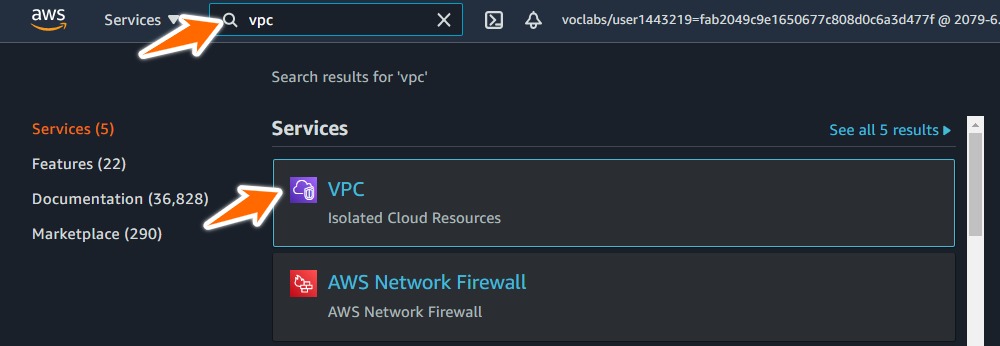
Ref cloudformation template1 note pad

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-resource-ec2-vpc-gateway-attachment.html>

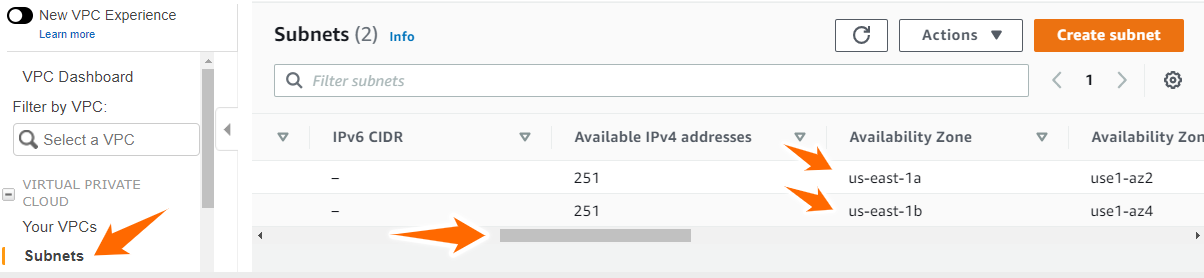
Verify The results of the CloudFormation Template

Validate that the base VPC has been created and that you have two public subnets in different Availability Zones. Create an EC2 key pair in the AWS Console. If there are any missing or incorrect elements, resolve the issues by deleting the stack, fixing the CF template, and creating a new stack.

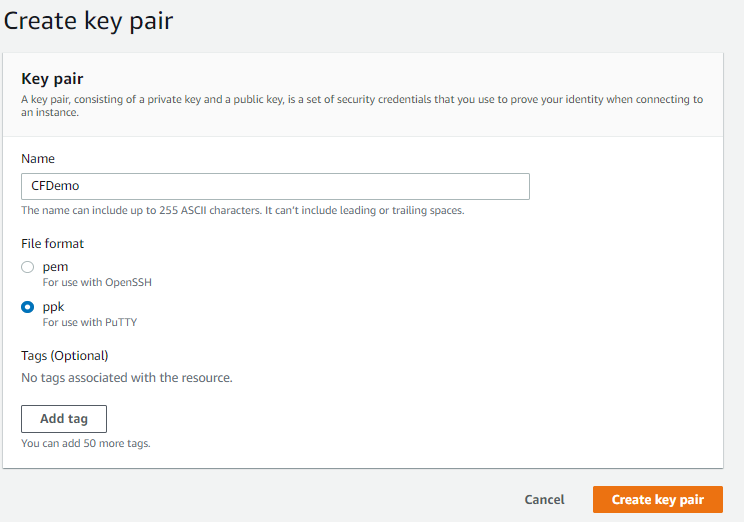
1. At this point your CloudFormation template should have successfully executed. If you wish to verify your template, you can use **VPC CloudFormation Template Solution 2.txt** from the resources
2. Go to the VPC console.



1. Click on **Your VPCs.**There should only be one VPC present.
2. Click on Subnets and verify that there are two subnets in different AZs.



1. Browse to the EC2 console and create a new KeyPair named "CFDemo"



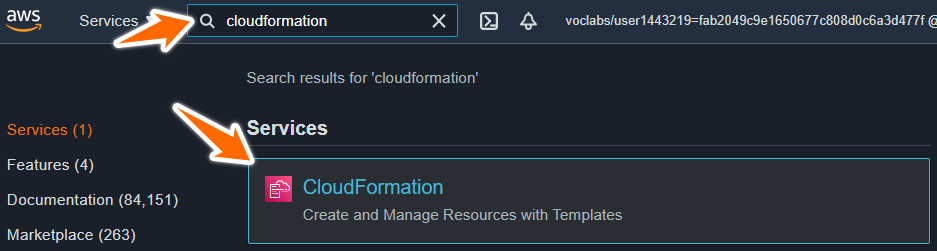
Ref cloudformation template2 note pad

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html>

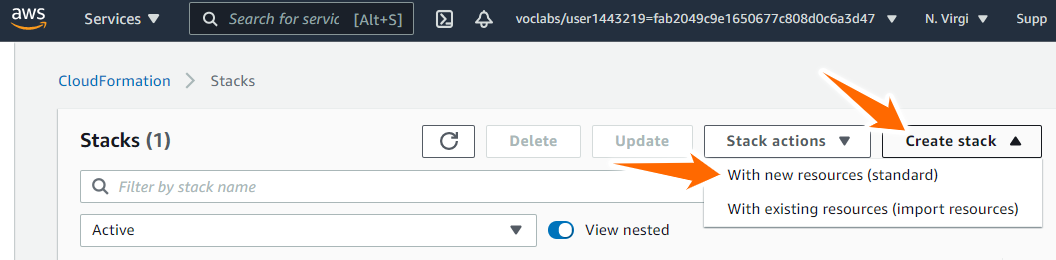
Troubleshoot WebApp cloudformatiom Template

The WebApp CloudFormation Template is used to create an auto scaling, load balanced group of web servers that run on EC2. Attempt to execute the WebApp CloudFormation Template provided and resolve any errors until it successfully completes.

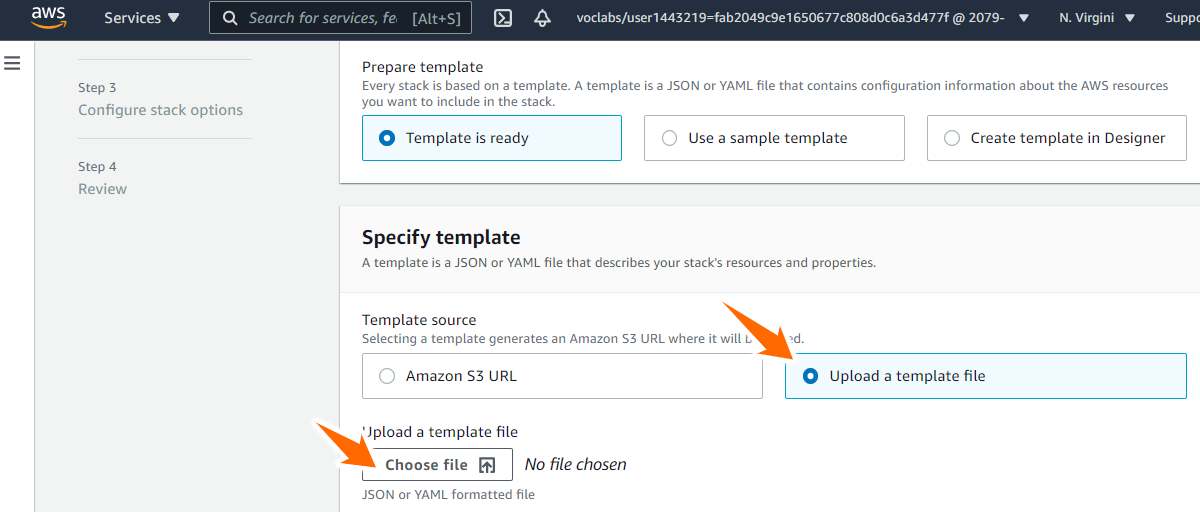
1. Download the WebApp CloudFormation Template provided in Resources and open it with your text editor.
2. From the AWS Console, browse to the CloudFormation console.



1. Create a stack of new resources.



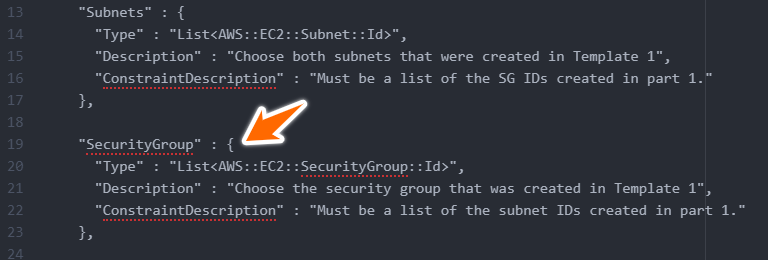
1. Click on "Choose file" and browse to "CloudFormation Template for Task 2 Solution"



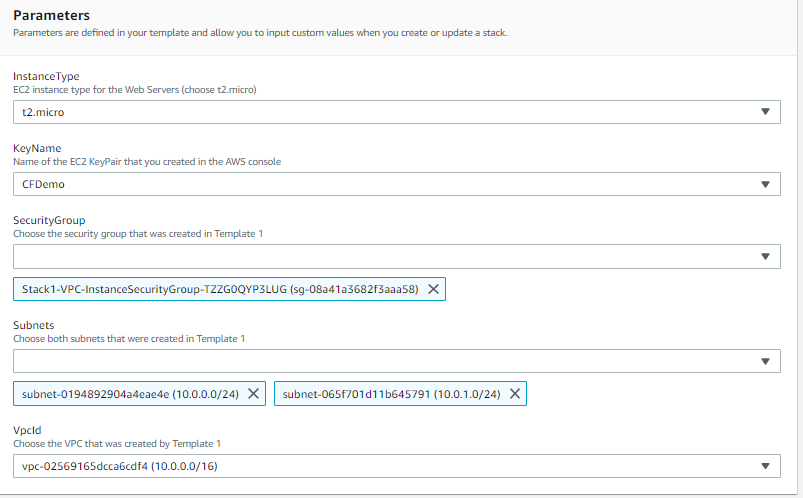
1. When you click next you will see an error like this:



1. The Security Group parameter is missing from the template. Resolve the issue, save the template, and then click on "Choose file" and try again.



1. For Stack Name, type Stack2-WebApp
2. Fill out the parameters. Notice the description for each parameter will guide you. Your screen will look slightly different from the image shown here because your VPC and Subnet identifiers will be different.



1. Click "Next" twice, then click "Create Stack"
2. Monitor the stack creation. Use the refresh button to update the status. When you see CREATE\_COMPLETE the stack has been successfully created.

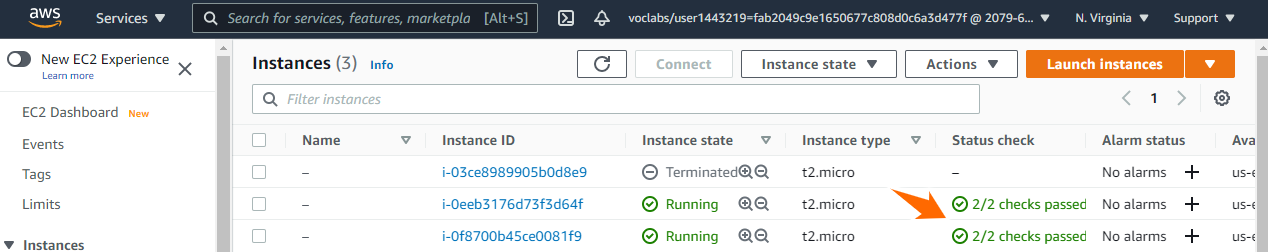
Ref WebApp cloudformation template note pad

<https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-ec2-security-group.html>

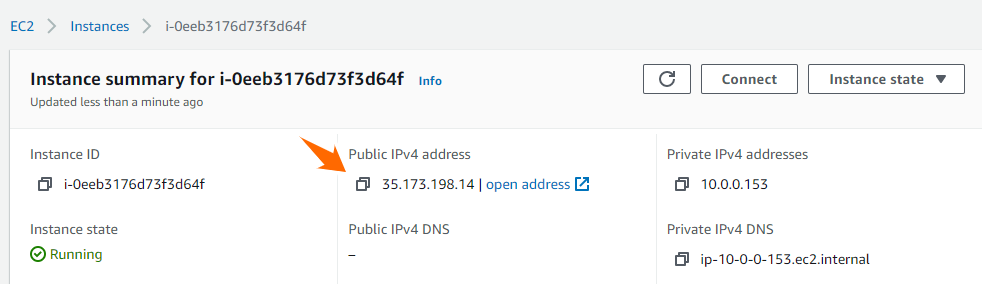
Verify CloudFormation Template#2

Verify that the web servers are working properly by connecting to their IP addresses in a web browser. Do not test the Load Balancer, as it may take too long for DNS to propagate.

1. At this point your WebApp CloudFormation template should have successfully executed. If you wish to verify your template, you can use **WebApp CloudFormation Template Solution.txt** from the resources.
2. Browse to the EC2 Console and confirm that you have two running EC2 instances.
3. Verify that the status check on both instances shows 2/2 checks passed



1. Find the public IP of each web server, and copy and paste it into a new browser window.



1. You should see a web page that says "Rick Crisci Lab Web Server"

Ref WebApp cloudformation template note pad

Clean Up

Our test of the CloudFormations template is now complete. Remove the stacks of AWS resources

1. Go to the CloudFormation Console and click on **Stacks**.
2. Select Stack2-WebApp and click on Delete. Wait for the deletion process to complete.
3. Select Stack1-VPC and click on Delete. Wait for the deletion process to complete.
4. Validate that the VPC and EC2 instances are gone.

Mark task as complete

Report issue