Lab Assignment 3

- **There are five steps which I followed to achieve the goal of the assignment which are as follows:**

1. Step for JSON Template setup for Amazon Cloud Formation:

- Amazon have some template samples available on its website, to check them we can visit on the link below:
  https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-west-2.html#w1ab2c21c45c15c15

- I am using the free template having name Amazon EC2 instance in a security group:
  https://s3-us-west-2.amazonaws.com/cloudformation-templates-us-west-2/EC2InstanceWithSecurityGroupSample.template

- In the template, I am making my key(which I used in previous lab) as default key:

```
"KeyName": {
  "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instance",
  "Type": "AWS::EC2::KeyPair::KeyName",
  "Default": "Assignment",
  "ConstraintDescription" : "must be the name of an existing EC2 KeyPair."
},
```

- Changing Instance type to t2.micro, as it is free:

```
"InstanceType" : {
  "Description" : "WebServer EC2 instance type",
  "Type" : "String",
  "Default" : "t2.micro",
  "AllowedValues" : [ "t1.micro", "t2.nano", "t2.micro", "t2.small", "t2.medium", "t2.large", "m1.small", "m1.medium", "m1.large", "m1.xlarge", "m2.xlarge",
,
  "ConstraintDescription" : "must be a valid EC2 instance type."
},
```

- Not changing the CIDR rule, so it can connect from any outside IP:

```
"SSHLocation" : {
  "Description" : "The IP address range that can be used to SSH to the EC2 instances",
  "Type": "String",
  "MinLength": "9",
  "MaxLength": "18",
  "Default": "0.0.0.0/0",
  "AllowedPattern": "(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})\\.(\\d{1,3})/(\\d{1,2})",
  "ConstraintDescription": "must be a valid IP CIDR range of the form x.x.x.x/x."
}
},
```

- Removing unnecessary mappings:

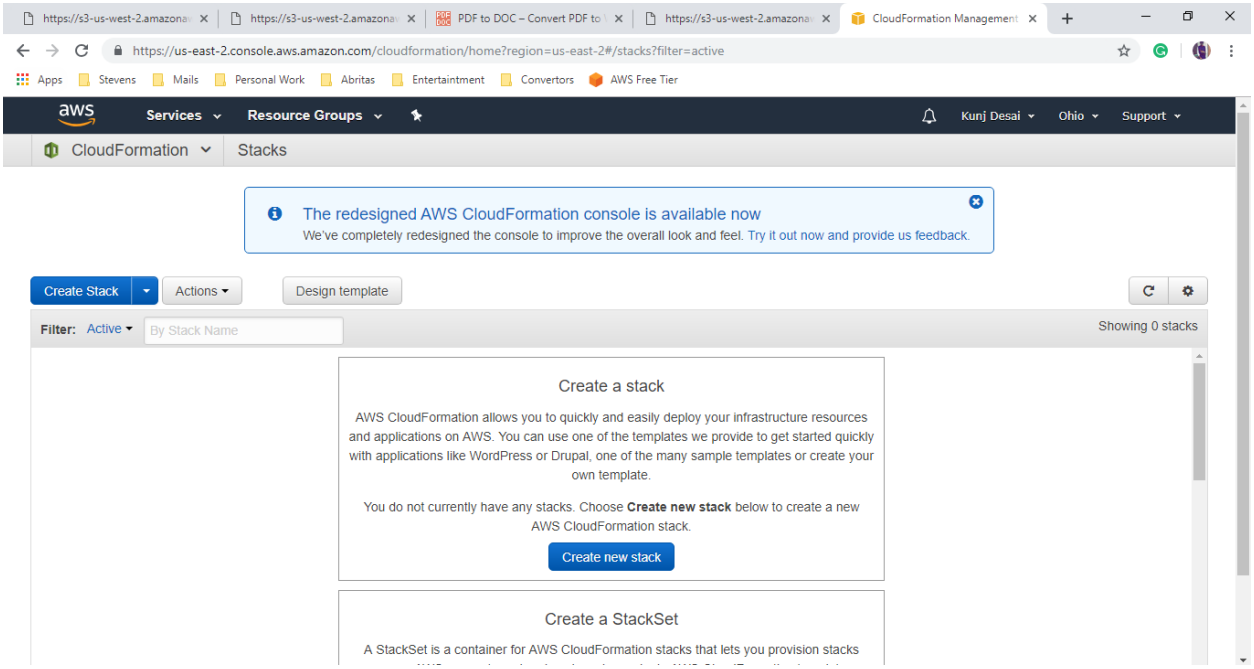
```
"Mappings" : {
  "AWSInstanceType2Arch" : {

    "t2.micro"    : { "Arch" : "HVM64"  }
  },

  "AWSInstanceType2NATArch" : {

    "t2.micro"    : { "Arch" : "NATHVM64"  }

  }
,
  "AWSRegionArch2AMI" : {

    "us-east-2"        : {"HVM64" : "ami-0cd3dfa4e37921605", "HVMG2" : "NOT_SUPPORTED"}

  }

},
```

- Changing Security group document, by adding setting for port 80 for http connection:

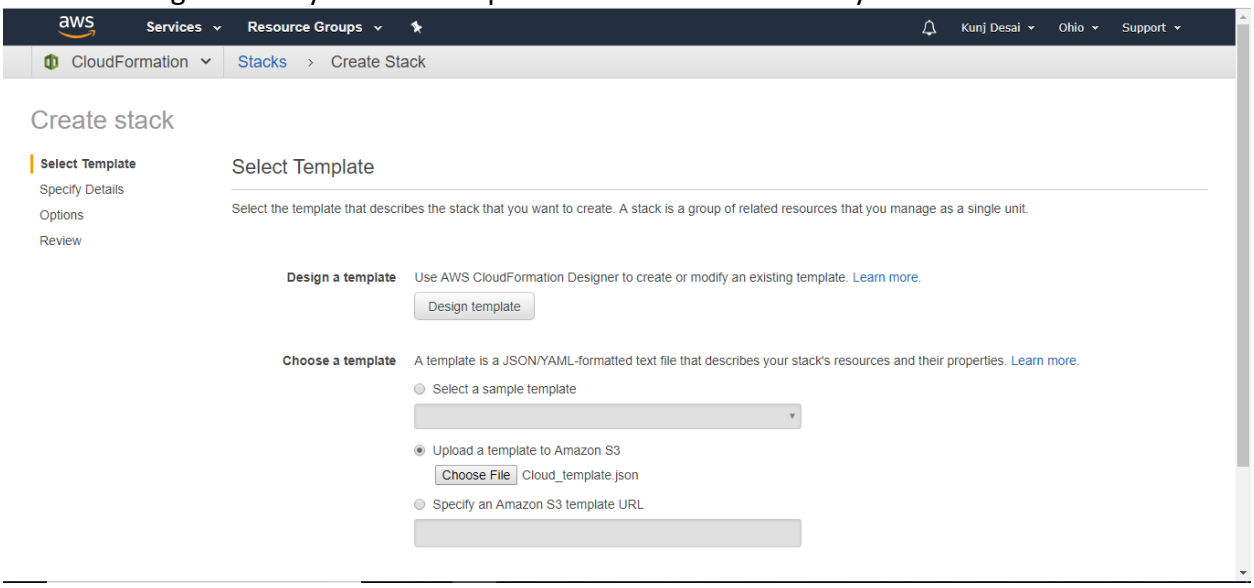
```
    "InstanceSecurityGroup" : {
      "Type" : "AWS::EC2::SecurityGroup",
      "Properties" : {
        "GroupDescription" : "Enable SSH access via port 22",
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "22",
          "ToPort" : "22",
          "CidrIp" : { "Ref" : "SSHLocation"}
        } ]
        "SecurityGroupIngress" : [ {
          "IpProtocol" : "tcp",
          "FromPort" : "80",
          "ToPort" : "80",
          "CidrIp" : { "Ref" : "SSHLocation"}
        } ]
      }
    }
  },
```

2. Step for Creating a Stack by Amazon Cloud Formation (Using AWS Panel):

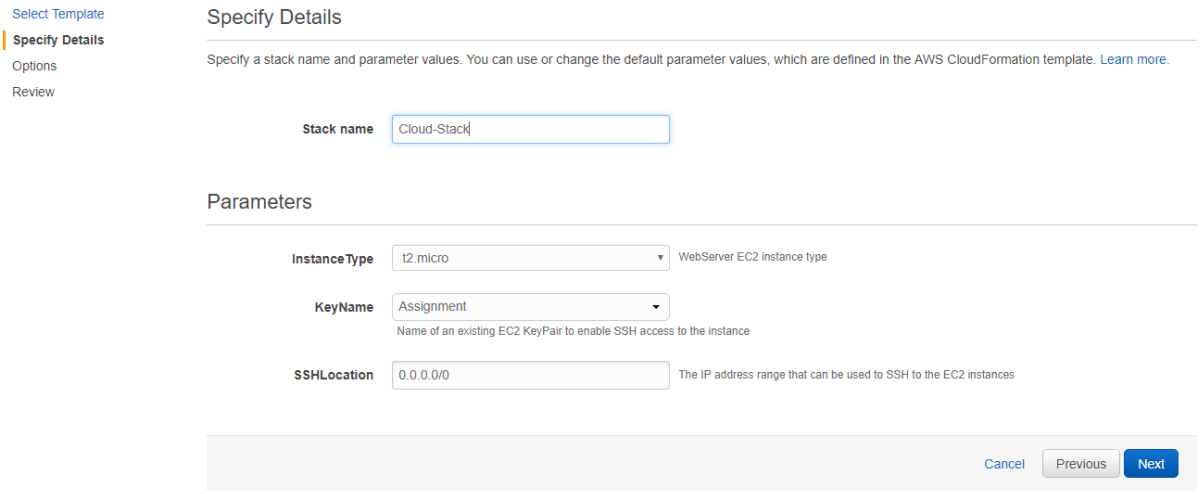
- Click on Create New Stack button after login to AWS account then select CloudFormation service in AWS dashboard.



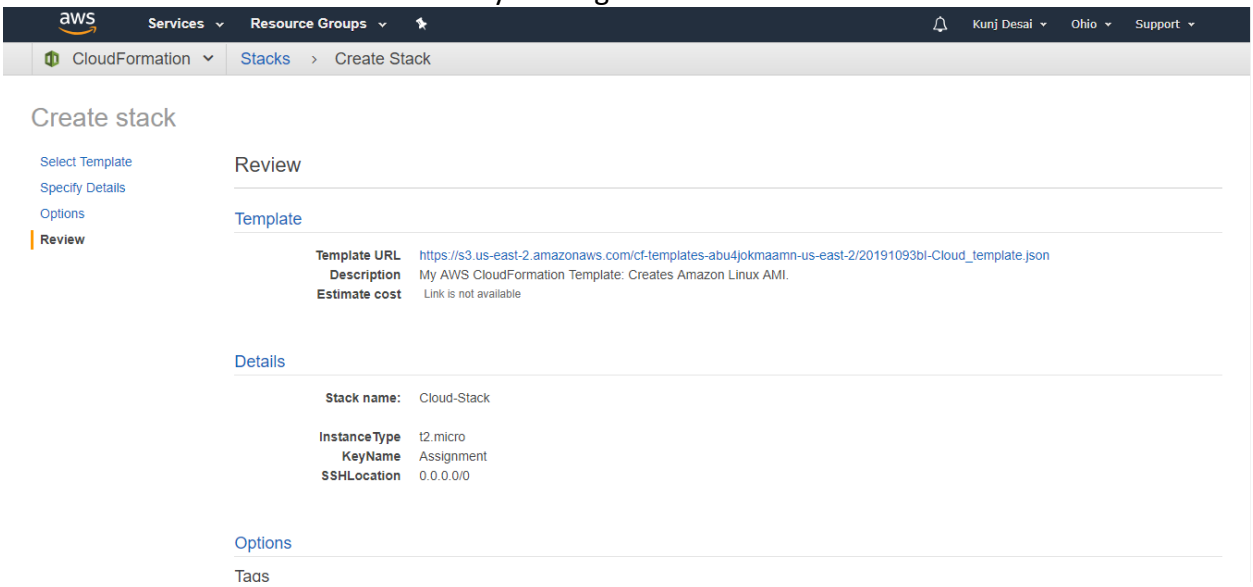
- Now selecting a recently created template from the local directory and click on next:



- In this step, we have to specify details after selecting template. Click next to proceed.

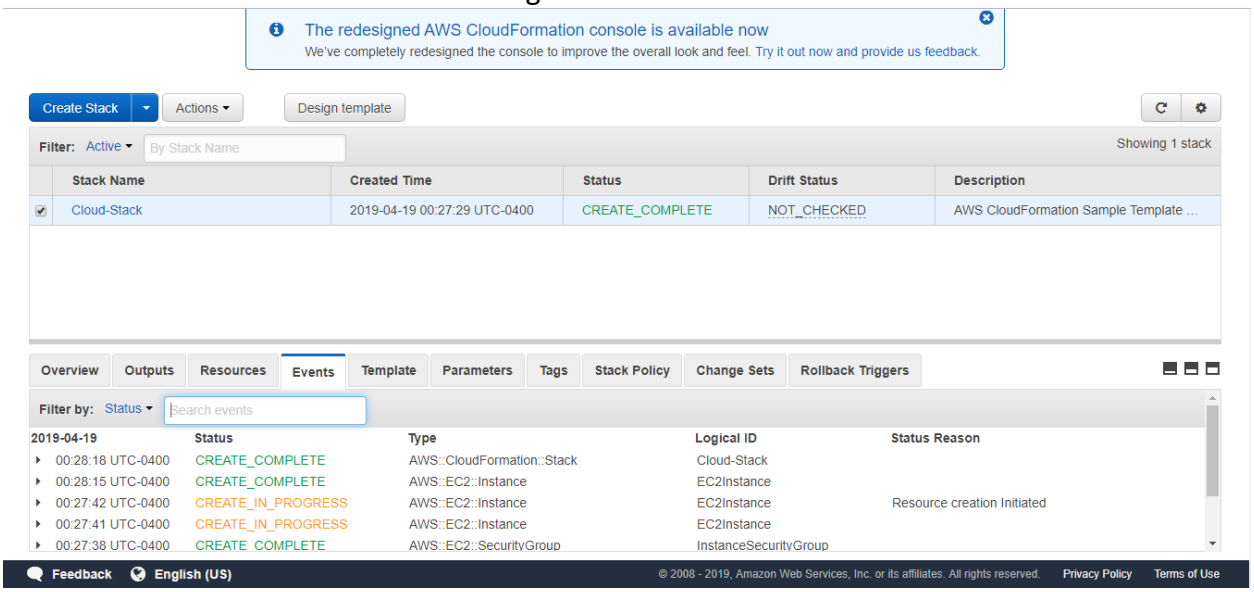


- After not making any changes in Option Menu, we came to Review menu to review the details of the created stack. And finally clicking to create Button to create a stack.



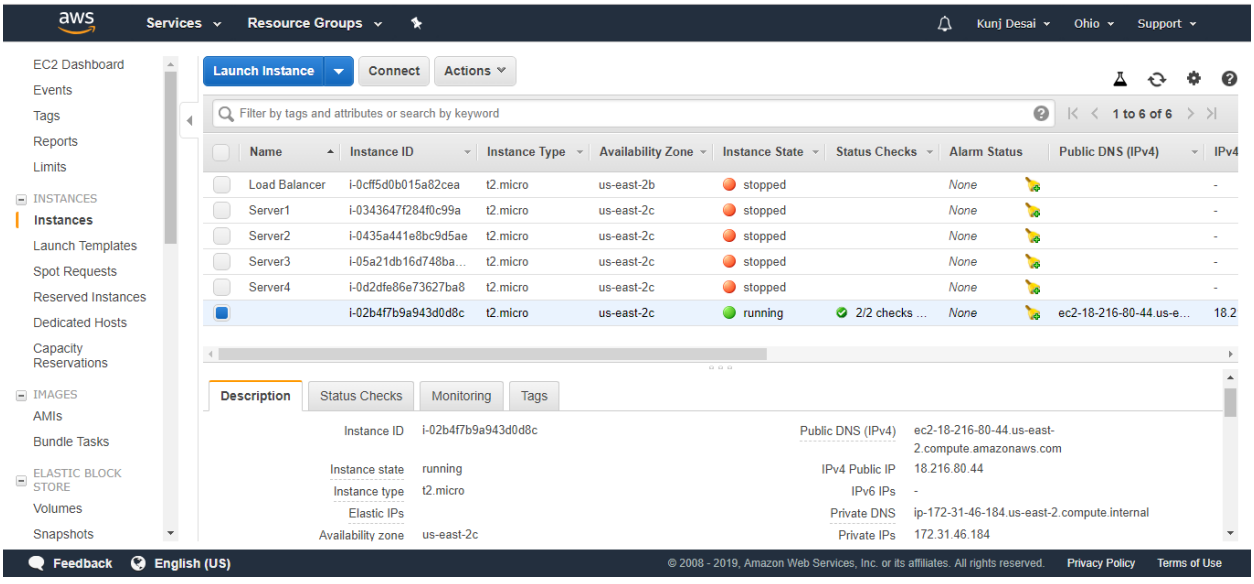
- Now here in this page, we can check that the stack creation is in process. And all stacks events are under process.

- After creation of stack successfully, the status changes to “CREATE_COMPLETE” . And we can check the instance created during the creation of stack.

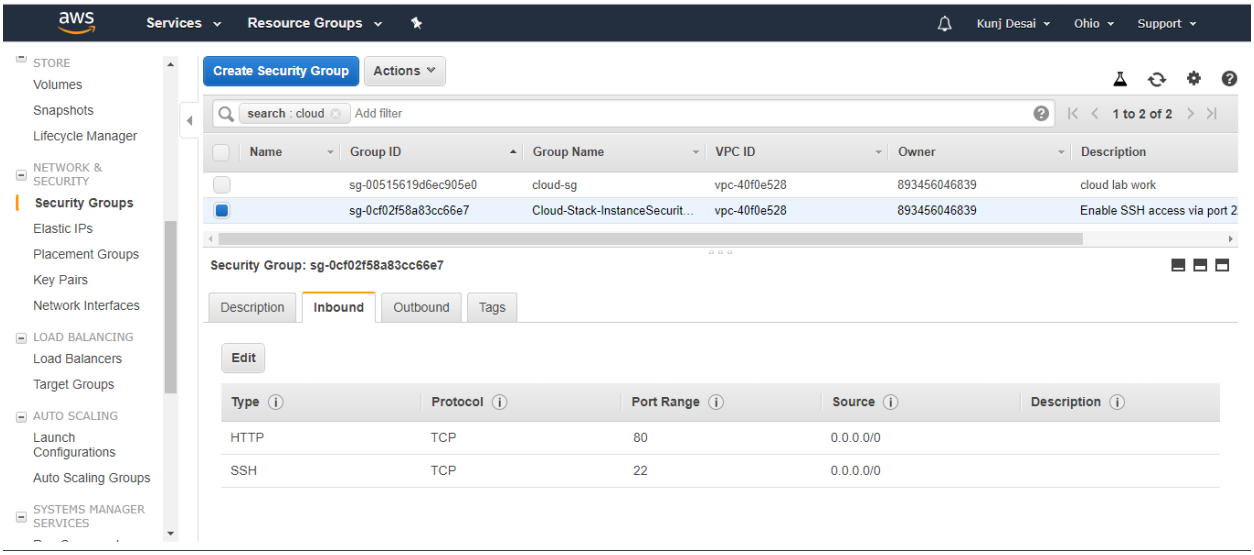


- We can also check all other events created and their status.

- To verify the instance, we logged in to EC2 service and found new running instance under instances menu.



- Now we can also verify the security group that has been created during stack creation under security group menu in EC2 service.



3. Step for JOSN template setup for a Load Balancer on Amazon Cloud Formation:

- Amazon have some templates sample available on its website, to check them we can visit the link below: https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/sample-templates-services-us-west-2.html#w1ab2c21c45c15c15

- I am using a free template having name Load Balanced Auto Scaling group: https://s3.amazonaws.com/cloudformation-templates-us-east-1/AutoScalingKeepAtNSample.template

- In this template, I am changing the key default value, instance type default value and CIDR rule, as I did before for this lab in previous steps.

```
        "KeyName": {
            "Description": "Name of an existing EC2 KeyPair to enable SSH access to the instance",
            "Type": "AWS::EC2::KeyPair::KeyName",
            "Default": "Assignment",
            "ConstraintDescription": "must be the name of an existing EC2 KeyPair."
        },
        "InstanceType": {
            "Description": "WebServer EC2 instance type",
            "Type": "String",
            "Default": "t2.micro",
            "AllowedValues": ["t2.micro"],
            "ConstraintDescription": "must be a valid EC2 instance type."
        },
```

- Now creating 2 instances by keeping minSize and maxSize value is 2 in Auto Scaling. After completing using value 2, I will change it to 4 for four instances and check version controlling.

```
"Resources" : {
  "WebServerGroup" : {
    "Type" : "AWS::AutoScaling::AutoScalingGroup",
    "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "2",
      "MaxSize" : "2",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
    },
```

- In LaunchConfig, we can check **httpd** for http server and format of index.html file in files document.

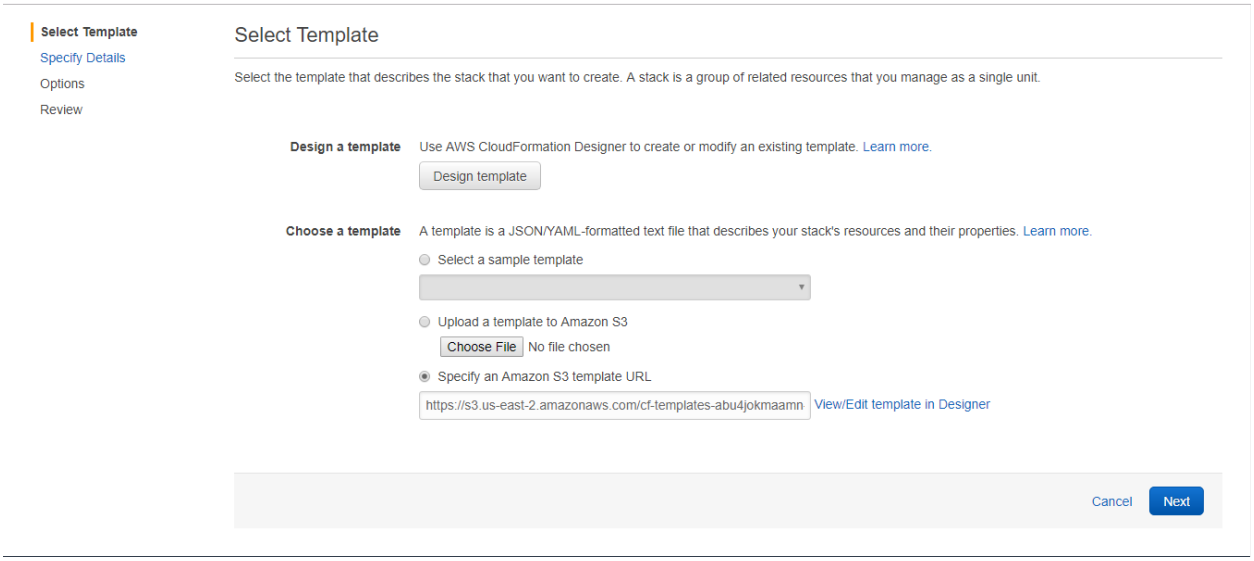
```
"LaunchConfig" : {
  "Type" : "AWS::AutoScaling::LaunchConfiguration",
  "Metadata" : {
    "Comment" : "Install a simple application",
    "AWS::CloudFormation::Init" : {
      "config" : {
        "packages" : {
          "yum" : {
            "httpd" : []
          }
        },

        "files" : {
          "/var/www/html/index.html" : {
            "content" : { "Fn::Join" : ["\n", [
              "<img src=\"", {"Fn::FindInMap" : ["Region2Examples", {"Ref" : "AWS::Region"}, "Examples"]}, "/cloudformation_graphic.png\" alt=\"AWS CloudFor
              "<h1>Congratulations, you have successfully launched the AWS CloudFormation sample.</h1>"
            ]]},
            "mode"    : "000644",
            "owner"   : "root",
            "group"   : "root"
          },

          "/etc/cfn/cfn-hup.conf" : {
            "content" : { "Fn::Join" : ["", [
              "[main]\n",
              "stack=", { "Ref" : "AWS::StackId" }, "\n",
              "region=", { "Ref" : "AWS::Region" }, "\n"
            ]]},
            "mode"    : "000400",
            "owner"   : "root",
            "group"   : "root"
          },
```

- Port setting for Load Balancer:

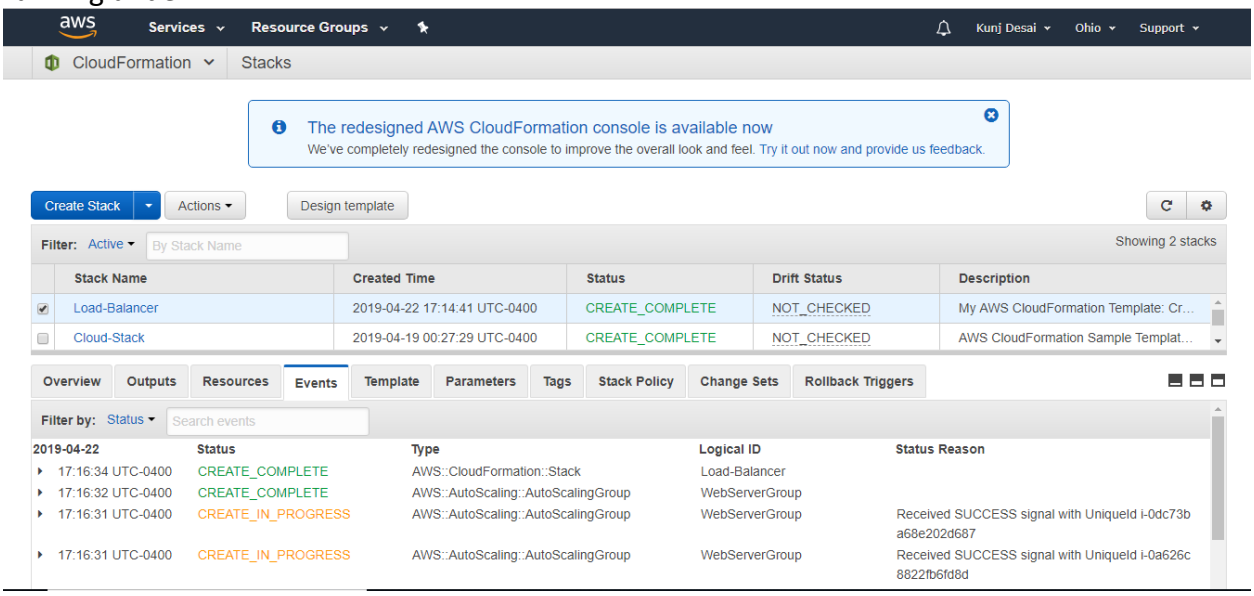
```
    "ElasticLoadBalancer" : {
      "Type" : "AWS::ElasticLoadBalancing::LoadBalancer",
      "Properties" : {
        "AvailabilityZones" : { "Fn::GetAZs" : "" },
        "CrossZone" : "true",
        "Listeners" : [ {
          "LoadBalancerPort" : "80",
          "InstancePort" : "80",
          "Protocol" : "HTTP"
        } ],
        "HealthCheck" : {
          "Target" : "HTTP:80/",
          "HealthyThreshold" : "3",
          "UnhealthyThreshold" : "5",
          "Interval" : "30",
          "Timeout" : "5"
        }
      },
```

4. Steps for updating a Stack by Amazon Cloud Formation:

- Select the created stack and click action then update stack for uploading Load Balancer. Then, select Load Balancer template to update. Click next.

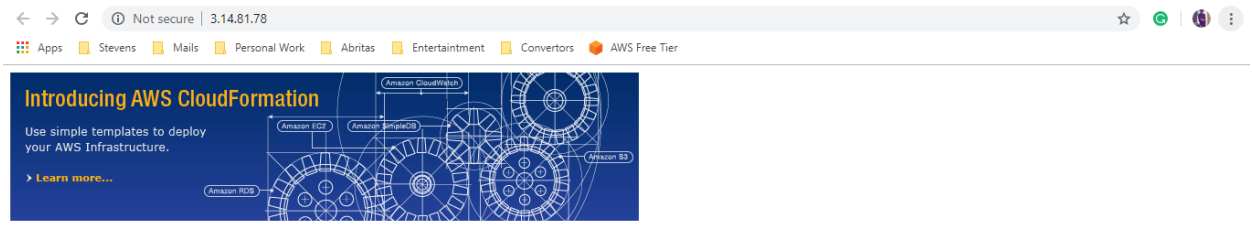


- After performing same steps, that we did while configuration template for stack creating. We will get “UPDATE_IN_PROGRESS” on current stack and can check all events running under.



- After status changes to “UPDATE_COMPLETE”. We can check the output of Load balancer. We can also verify that instances, security group and network interfaces have been created successfully.

- After successful update of stack, we can check that the instances are running on servers. This image is for "**Server 1".**

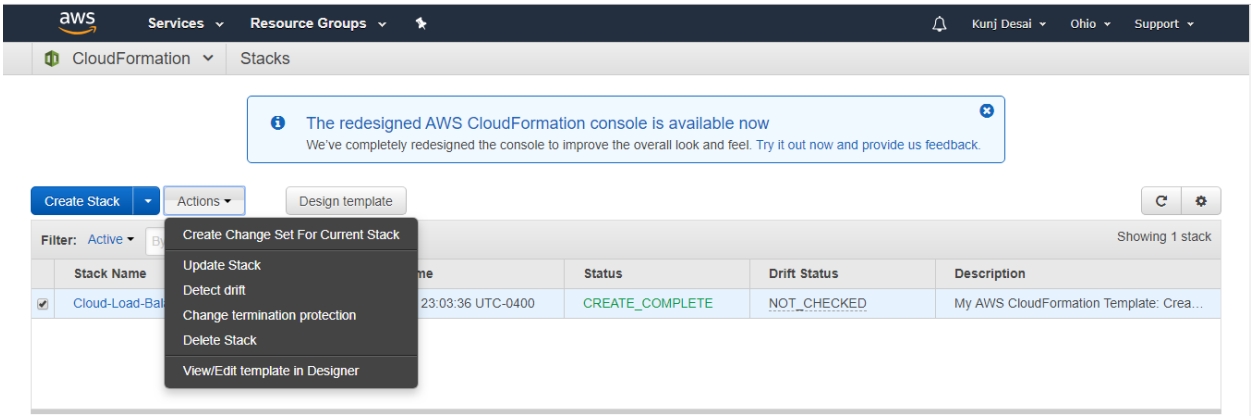


Congratulations, you have successfully launched the AWS CloudFormation sample.

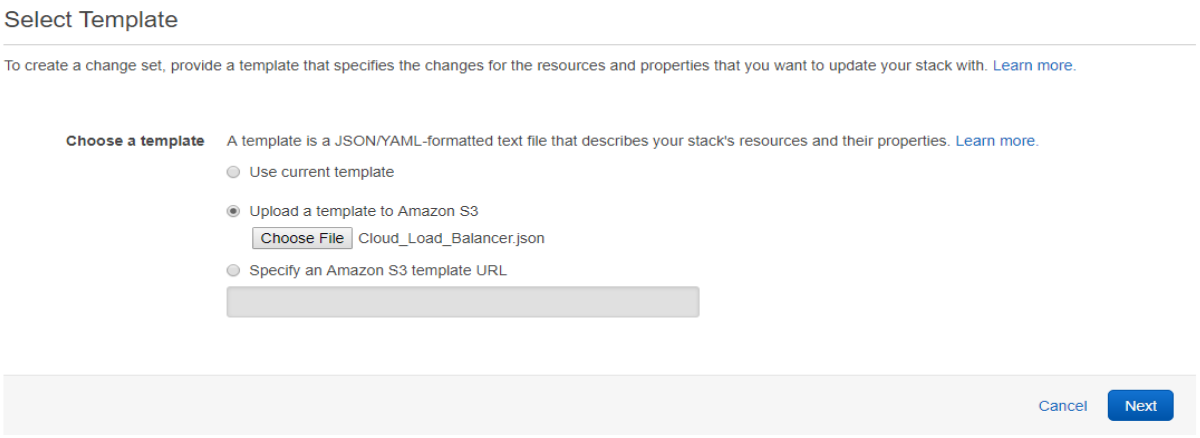
- Now after successful execution of servers on instances, I will change the minSize and maxSize value to 4 from 2 in Auto Scaling(as mentioned before).

```
"Resources" : {
  "WebServerGroup" : {
    "Type" : "AWS::AutoScaling::AutoScalingGroup",
    "Properties" : {
      "AvailabilityZones" : { "Fn::GetAZs" : "" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "4",
      "MaxSize" : "4",
      "LoadBalancerNames" : [ { "Ref" : "ElasticLoadBalancer" } ]
    },
    "CreationPolicy" : {
      "ResourceSignal" : {
        "Timeout" : "PT15M",
        "Count"   : "2"
      }
    },
```

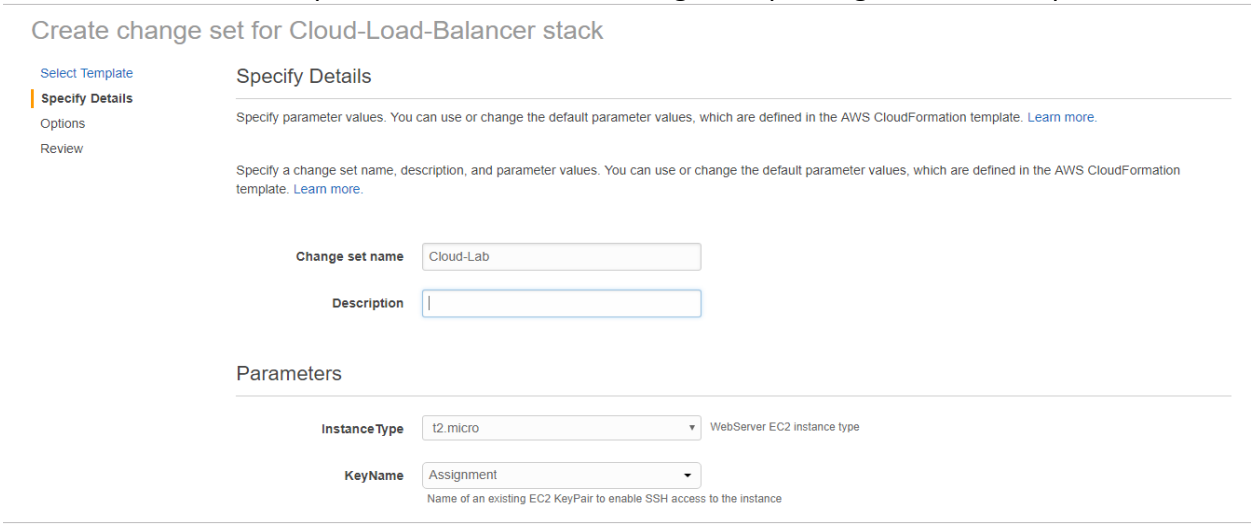
- After changing the Auto Scaling min and max values. We will create Change set for current stack.



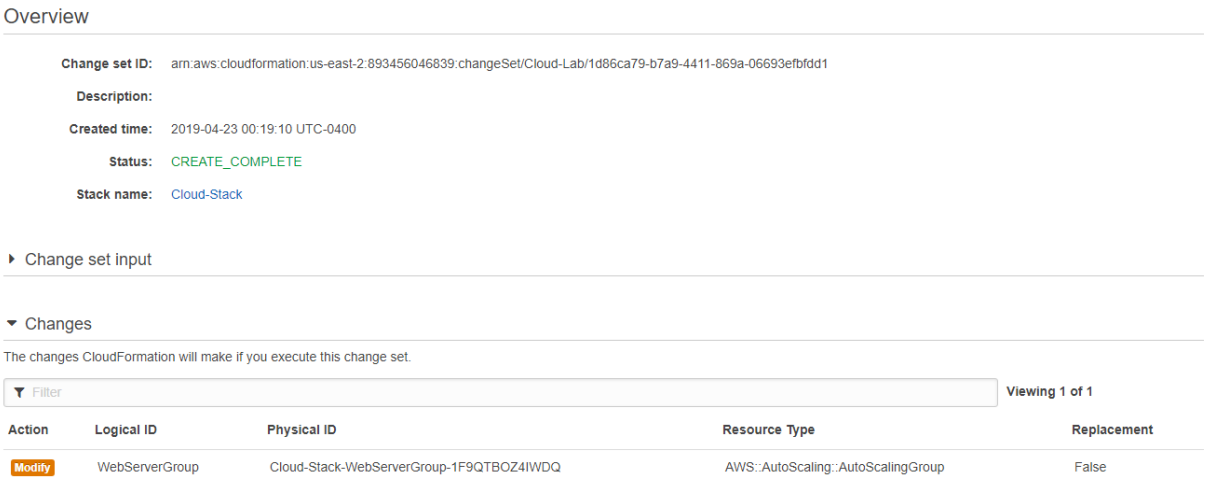
- While creating change set, we will provide the updated JSON template for Auto Scaling. Then, Click next.



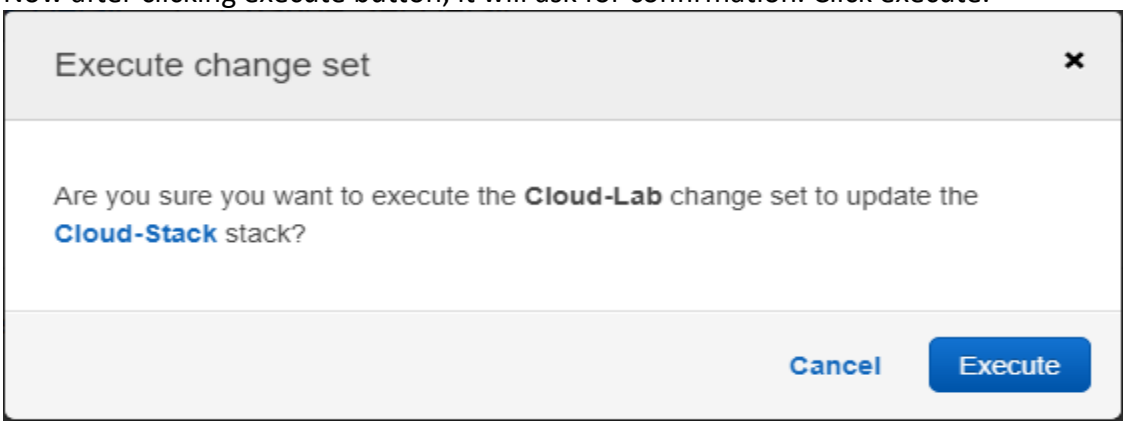
- Now changing the specification details for change set for current stack. Then, click next and follow the same step what we did while creating and updating the stack template.



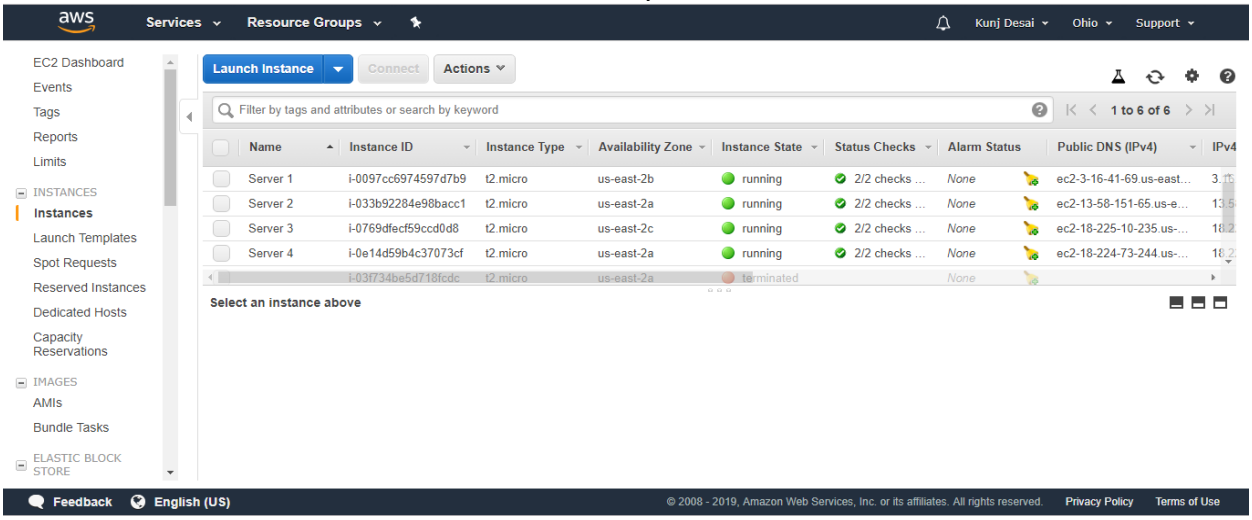
- Now check the overview of updated stack details, and notice the WebServerGroup has been –modified instead of replacing. Then click on Execute.



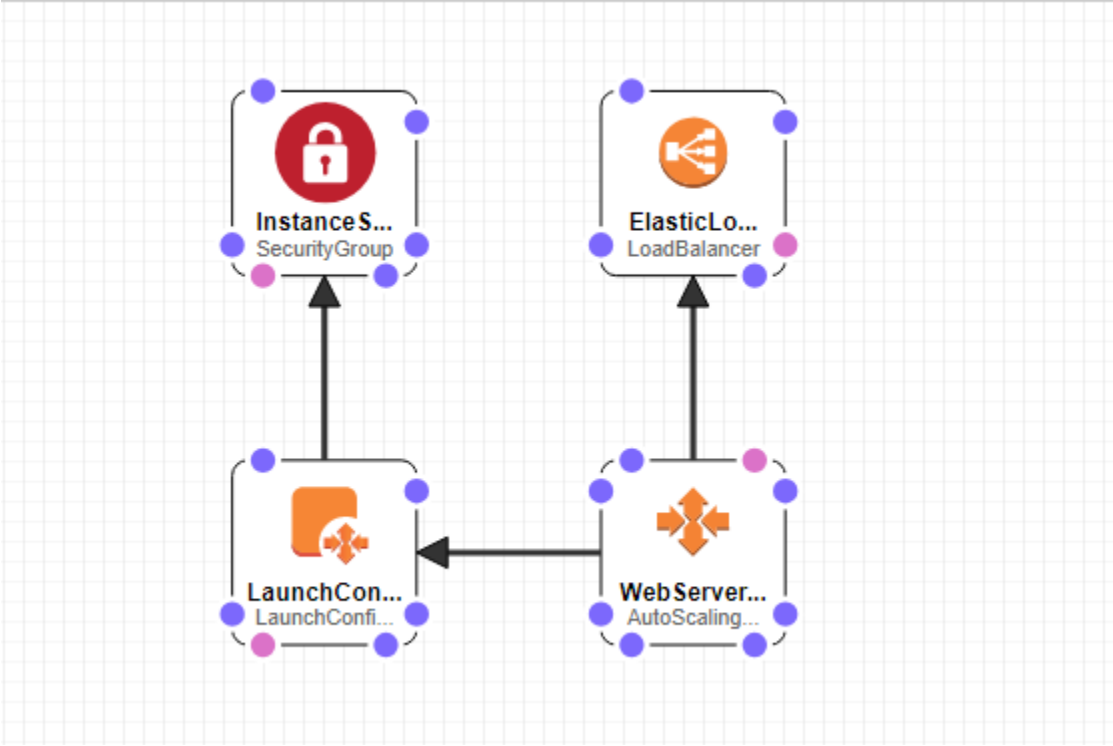
- Now after clicking execute button, it will ask for confirmation. Click execute.



- After it we can check all the events on the current stack has been updated and will get “Update_Complete” status. We can also verify that instances, security group and network interfaces have been created successfully.

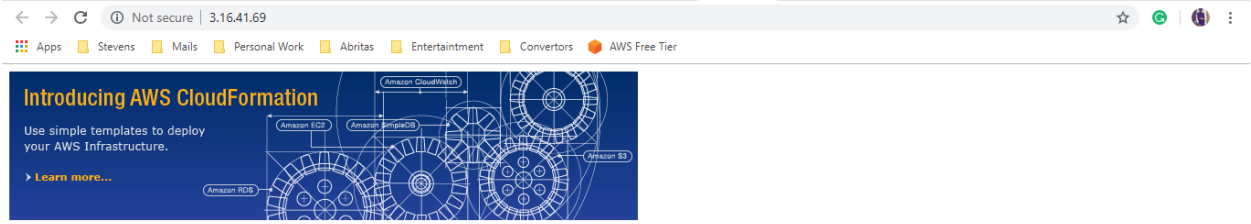


- We can also check the template in the designer and can manually modify.



- After successfully update the stack, we can check that the instances are running on the servers.

1) Server 1: 3.16.41.69
2) Server 2: 13.58151.65
3) Server 3: 18.255.10.235
4) Server 4: 18.244.73.244

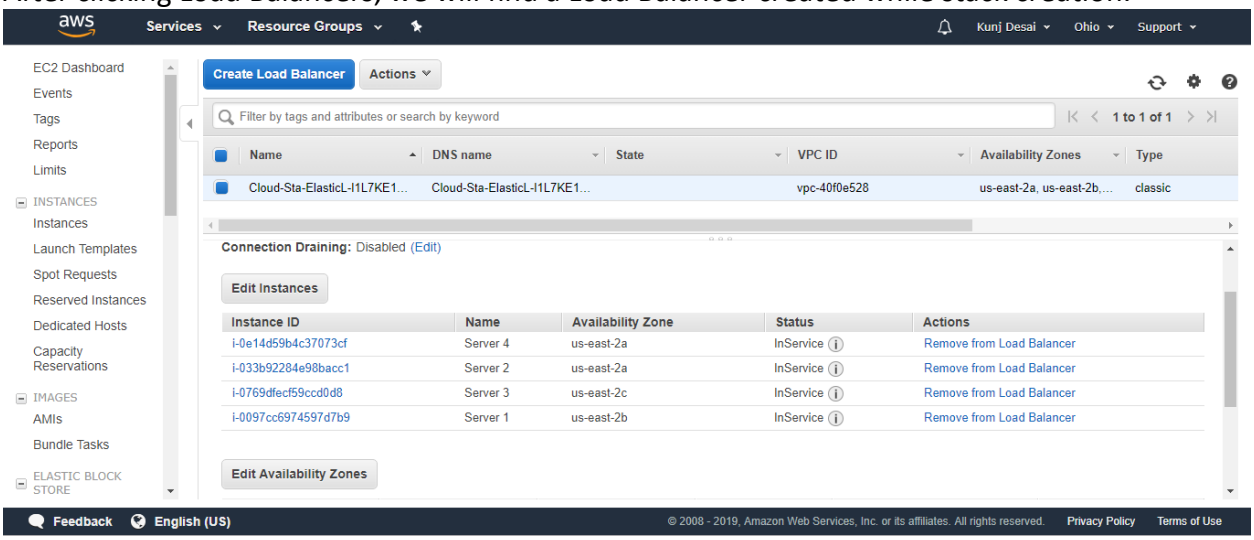


Congratulations, you have successfully launched the AWS CloudFormation sample.

5. Steps to check working of Load Balancer:

- Click on EC2 dashboard to find the Load Balancer and click on it.

- After clicking Load Balancers, we will find a Load Balancer created while stack creation.



- Now visit the load balancer link (http://cloud-sta-elasticl-i1l7ke1e9zwc-1581805070.us-east-2.elb.amazonaws.com/). And this load balancer automatically balances the load and redirect to any of the created servers by dividing the load at equal proportion.