

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:

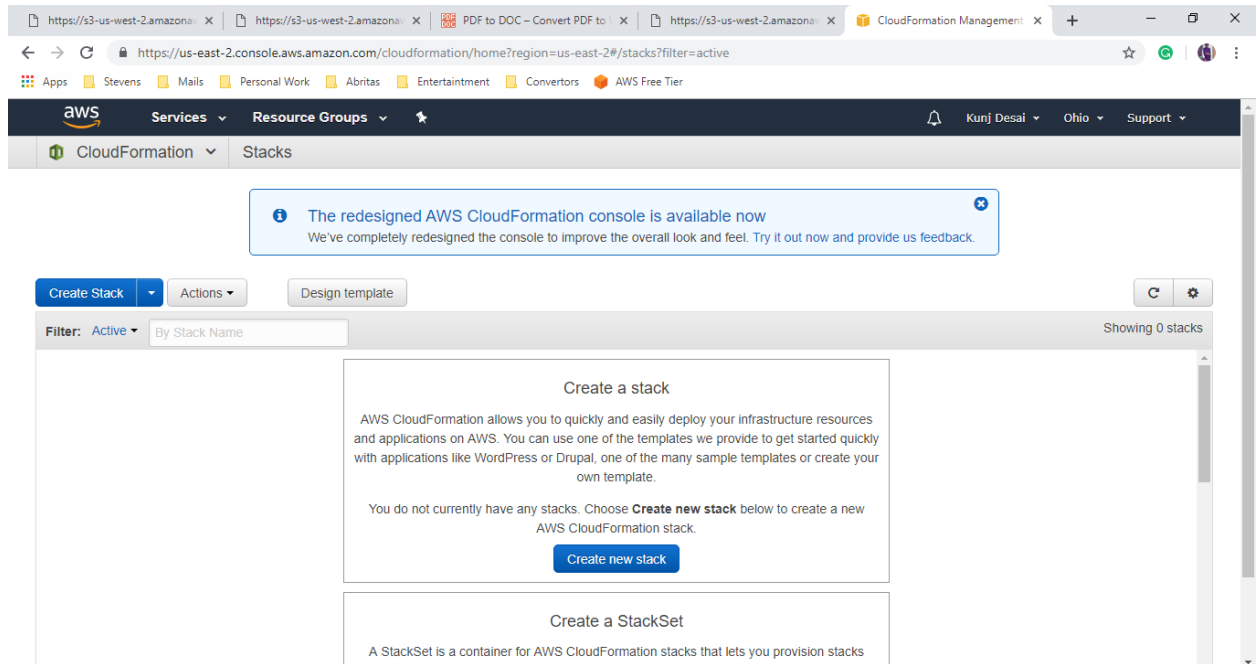
```

"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" }
    } ]
  }
},

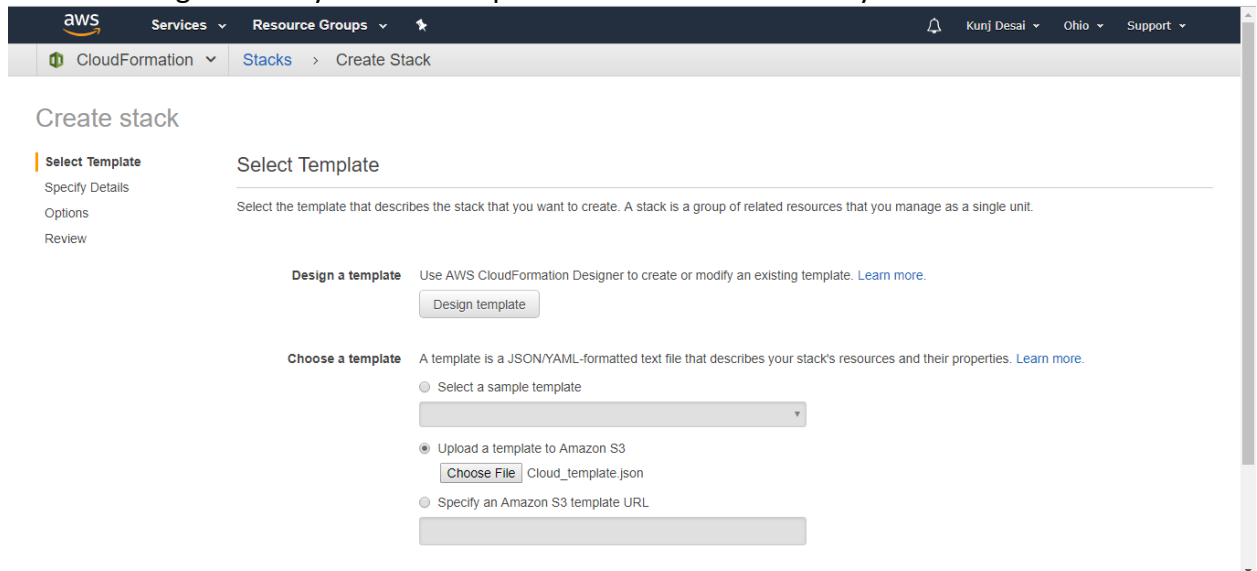
```

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.

[Select Template](#)
[Specify Details](#)
[Options](#)
[Review](#)

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Stack name

Parameters

InstanceType WebServer EC2 instance type

KeyName Name of an existing EC2 KeyPair to enable SSH access to the instance

SSHLocation The IP address range that can be used to SSH to the EC2 instances

[Cancel](#)
[Previous](#)
[Next](#)

- 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.

[aws](#)
[Services](#)
[Resource Groups](#)

[CloudFormation](#)
[Stacks](#)
[Create Stack](#)

Create stack

[Select Template](#)
[Specify Details](#)
[Options](#)
[Review](#)

Review

Template

Template URL https://s3.us-east-2.amazonaws.com/cf-templates-abu4jokmaamn-us-east-2/20191093bl-Cloud_template.json

Description My AWS CloudFormation Template: Creates Amazon Linux AMI.

Estimate cost Link is not available

Details

Stack name: Cloud-Stack

InstanceType t2.micro

KeyName Assignment

SSHLocation 0.0.0.0/0

Options

Taas

- 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.

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.

Create Stack Actions Design template

Filter: Active By Stack Name Showing 1 stack

	Stack Name	Created Time	Status	Drift Status	Description
<input checked="" type="checkbox"/>	Cloud-Stack	2019-04-19 00:27:29 UTC-0400	CREATE_COMPLETE	NOT_CHECKED	AWS CloudFormation Sample Template ...

Overview Outputs Resources **Events** Template Parameters Tags Stack Policy Change Sets Rollback Triggers

Filter by: Status Search events

2019-04-19	Status	Type	Logical ID	Status Reason
00:28:18 UTC-0400	CREATE_COMPLETE	AWS::CloudFormation::Stack	Cloud-Stack	
00:28:15 UTC-0400	CREATE_COMPLETE	AWS::EC2::Instance	EC2Instance	
00:27:42 UTC-0400	CREATE_IN_PROGRESS	AWS::EC2::Instance	EC2Instance	Resource creation Initiated
00:27:41 UTC-0400	CREATE_IN_PROGRESS	AWS::EC2::Instance	EC2Instance	
00:27:38 UTC-0400	CREATE_COMPLETE	AWS::EC2::SecurityGroup	InstanceSecurityGroup	

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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.

aws Services Resource Groups Kunj Desai Ohio Support

EC2 Dashboard Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

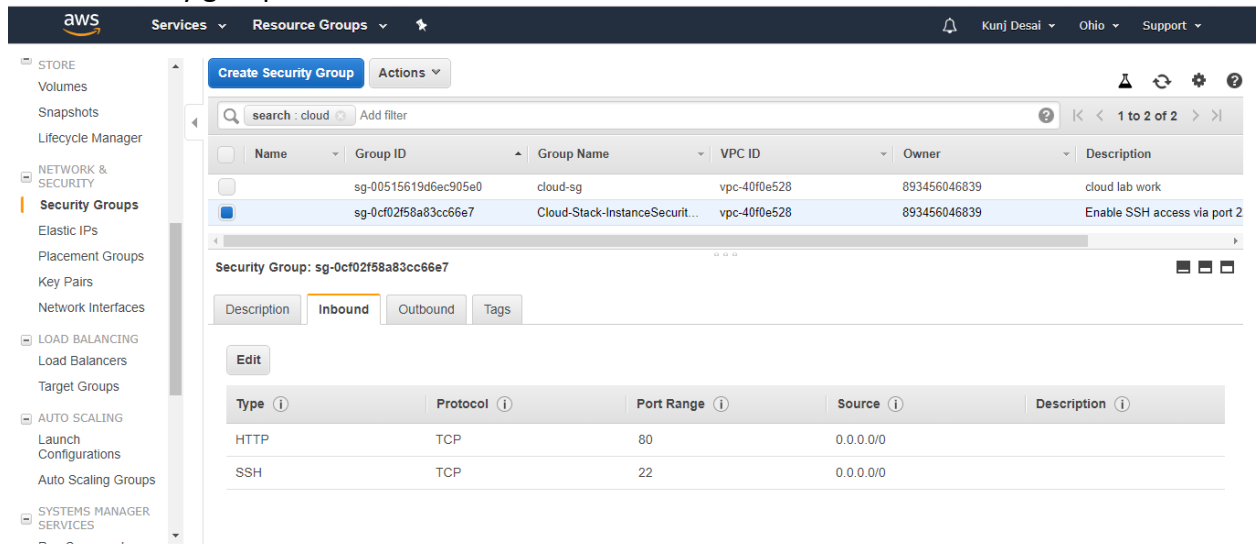
	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
<input type="checkbox"/>	Load Balancer	i-0cff5d0b015a82cea	t2.micro	us-east-2b	stopped		None		-
<input type="checkbox"/>	Server1	i-0343647f284f0c99a	t2.micro	us-east-2c	stopped		None		-
<input type="checkbox"/>	Server2	i-0435a441e8bc9d5ae	t2.micro	us-east-2c	stopped		None		-
<input type="checkbox"/>	Server3	i-05a21db16d748ba...	t2.micro	us-east-2c	stopped		None		-
<input type="checkbox"/>	Server4	i-0d2dfe86e73627ba8	t2.micro	us-east-2c	stopped		None		-
<input checked="" type="checkbox"/>		i-02b47b9a943d0d8c	t2.micro	us-east-2c	running	2/2 checks ...	None	ec2-18-216-80-44.us-e...	18.2

Description Status Checks Monitoring Tags

Instance ID	i-02b47b9a943d0d8c	Public DNS (IPv4)	ec2-18-216-80-44.us-east-2.compute.amazonaws.com
Instance state	running	IPv4 Public IP	18.216.80.44
Instance type	t2.micro	IPv6 IPs	-
Elastic IPs		Private DNS	ip-172-31-46-184.us-east-2.compute.internal
Availability zone	us-east-2c	Private IPs	172.31.46.184

Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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 CloudFormation logo\" data-bbox=\"100 100 200 200\"/>",
              "<h1>Congratulations, you have successfully launched the AWS CloudFormation sample.</h1>"
            ]]}},
            "mode" : "000644",
            "owner" : "root",
            "group" : "root"
          },
          "/etc/cfn/cfn-bundle.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:

```

"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.

Select Template

Select the template that describes the stack that you want to create. A stack is a group of related resources that you manage as a single unit.

Design a template Use AWS CloudFormation Designer to create or modify an existing template. [Learn more.](#)

Design template

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

☐ Select a sample template

☐ Upload a template to Amazon S3

Choose File No file chosen

☒ Specify an Amazon S3 template URL

[View/Edit template in Designer](#)

Cancel 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.

aws Services Resource Groups Kunj Desai Ohio Support

CloudFormation Stacks

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.](#)

Create Stack Actions Design template

Filter: Active By Stack Name Showing 2 stacks

Stack Name	Created Time	Status	Drift Status	Description
Load-Balancer	2019-04-22 17:14:41 UTC-0400	CREATE_COMPLETE	NOT_CHECKED	My AWS CloudFormation Template: Cr...
Cloud-Stack	2019-04-19 00:27:29 UTC-0400	CREATE_COMPLETE	NOT_CHECKED	AWS CloudFormation Sample Templat...

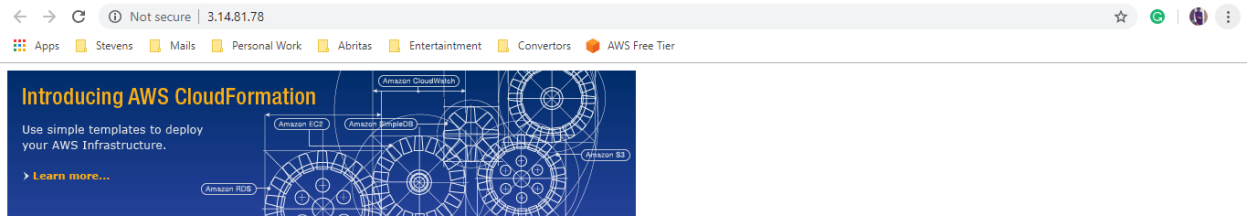
Overview Outputs Resources Events Template Parameters Tags Stack Policy Change Sets Rollback Triggers

Filter by: Status Search events

2019-04-22	Status	Type	Logical ID	Status Reason
17:16:34 UTC-0400	CREATE_COMPLETE	AWS::CloudFormation::Stack	Load-Balancer	
17:16:32 UTC-0400	CREATE_COMPLETE	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	
17:16:31 UTC-0400	CREATE_IN_PROGRESS	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	Received SUCCESS signal with UniqueId I-0dc73b a68e202d687
17:16:31 UTC-0400	CREATE_IN_PROGRESS	AWS::AutoScaling::AutoScalingGroup	WebServerGroup	Received SUCCESS signal with UniqueId i-0a626c 8822fb6fd8d

- 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**”.

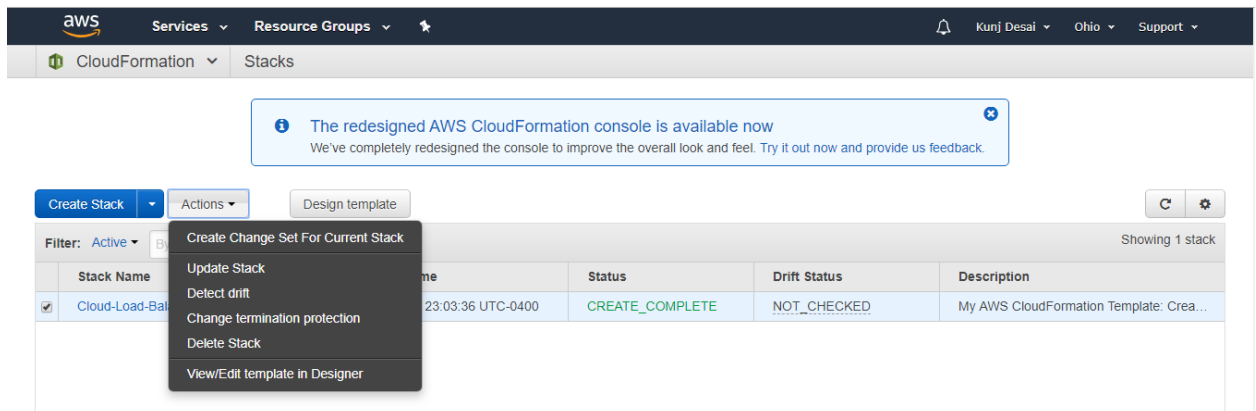


Congratulations, you have successfully launched the AWS CloudFormation sample.

- 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.

Select Template

To create a change set, provide a template that specifies the changes for the resources and properties that you want to update your stack with. [Learn more.](#)

Choose a template A template is a JSON/YAML-formatted text file that describes your stack's resources and their properties. [Learn more.](#)

- ☐ Use current template
- ☒ Upload a template to Amazon S3
 - Cloud_Load_Balancer.json
- ☐ Specify an Amazon S3 template URL

[Cancel](#) [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.

Create change set for Cloud-Load-Balancer stack

[Select Template](#)

Specify Details

[Options](#)

[Review](#)

Specify Details

Specify parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Specify a change set name, description, and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. [Learn more.](#)

Change set name

Description

Parameters

InstanceType WebServer EC2 instance type

KeyName Name of an existing EC2 KeyPair to enable SSH access to the instance

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

Overview

Change set ID: `arn:aws:cloudformation:us-east-2:893456046839:changeSet/Cloud-Lab/1d86ca79-b7a9-4411-869a-06693efbddd1`

Description:

Created time: 2019-04-23 00:19:10 UTC-0400

Status: **CREATE_COMPLETE**

Stack name: [Cloud-Stack](#)

► Change set input

▼ Changes

The changes CloudFormation will make if you execute this change set.

Filter Viewing 1 of 1

Action	Logical ID	Physical ID	Resource Type	Replacement
Modify	WebServerGroup	Cloud-Stack-WebServerGroup-1F9QTBOZ4IWDQ	AWS::AutoScaling::AutoScalingGroup	False

- 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.

aws Services Resource Groups

Launch Instance Connect Actions

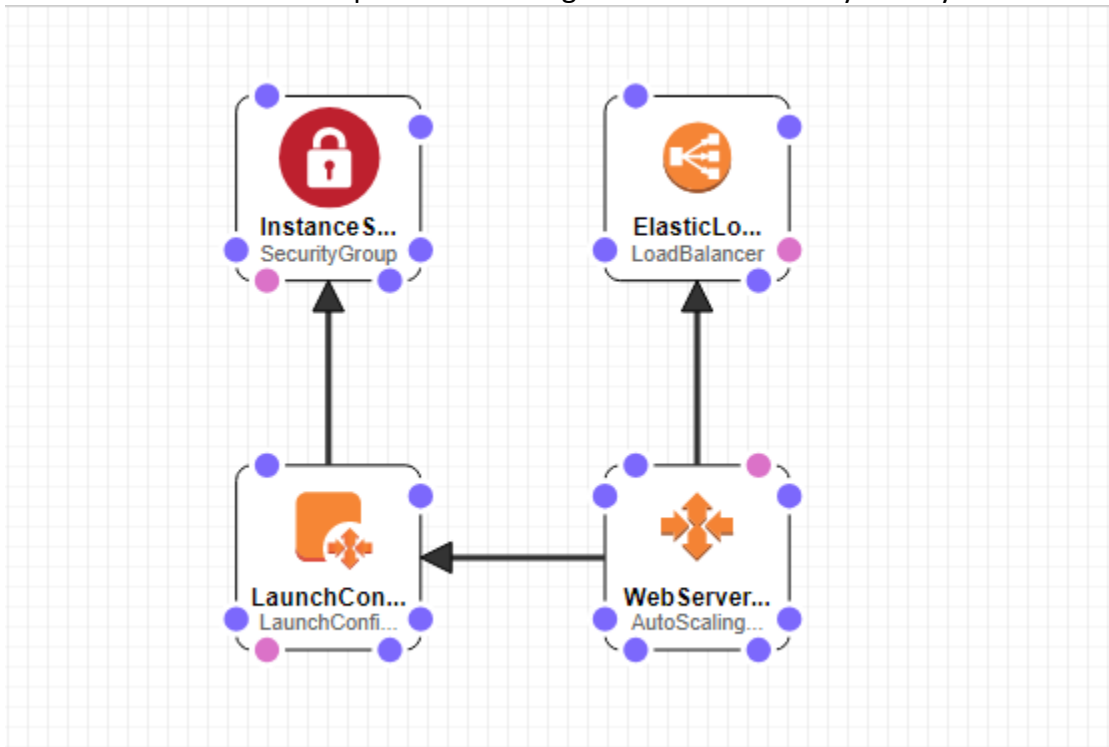
Filter by tags and attributes or search by keyword

	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4
<input type="checkbox"/>	Server 1	i-0097cc6974597d7b9	t2.micro	us-east-2b	running	2/2 checks ...	None	ec2-3-16-41-69.us-east...	3.15
<input type="checkbox"/>	Server 2	i-033b92284e98bacc1	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-13-58-151-65.us-e...	13.5
<input type="checkbox"/>	Server 3	i-0769dfecf59ccd0d8	t2.micro	us-east-2c	running	2/2 checks ...	None	ec2-18-225-10-235.us-...	18.2
<input type="checkbox"/>	Server 4	i-0e14d59b4c37073cf	t2.micro	us-east-2a	running	2/2 checks ...	None	ec2-18-224-73-244.us-...	18.2
<input type="checkbox"/>		i-031734be5d718fcdc	t2.micro	us-east-2a	terminated		None		

Select an instance above

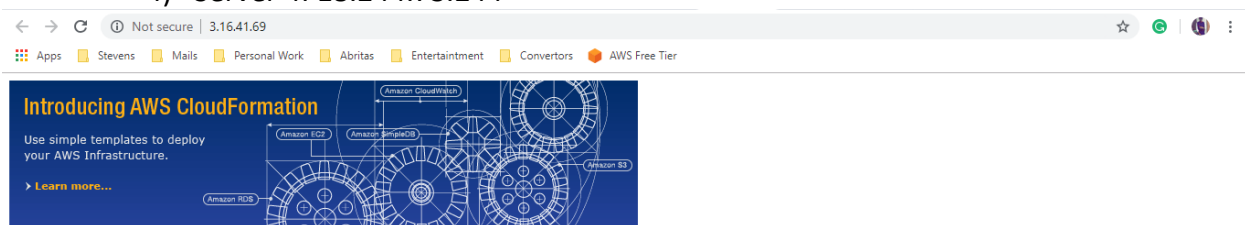
Feedback English (US) © 2008 - 2019, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- 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



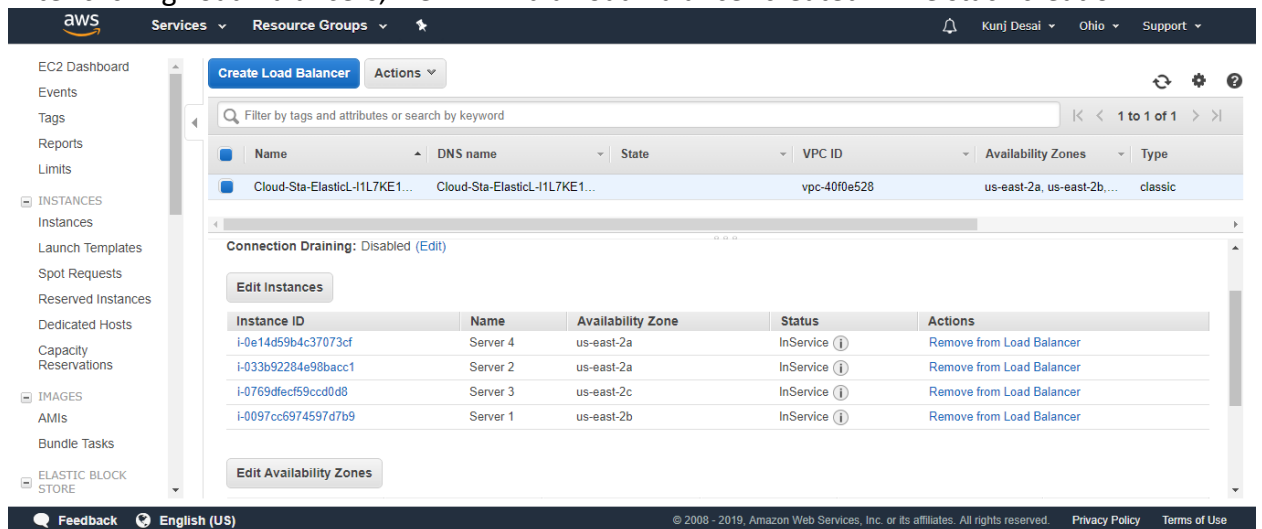
Congratulations, you have successfully launched the AWS CloudFormation sample.

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.