

CS 524 Introduction to Cloud Computing

Dharmit Viradia

Lab Assignment 4

Prof. Igor Faynberg

Step for Load Balancer Template setup for Amazon Cloud Formation

- I am using a free template having name Load Balanced Auto Scaling group

<https://s3.amazonaws.com/cloudformation-templates-us-east-1/AutoScalingKeepAtNSample.template>

- Create new Load Balancer Template in JSON as:

```
{  
    "AWSTemplateFormatVersion" : "2010-09-09",  
  
    "Description" : "AWS CloudFormation Sample Template AutoScalingKeepAtNSample: Create a  
load balanced, Auto Scaled sample website. This example creates an Auto Scaling group behind  
a load balancer with a simple health check. **WARNING** This template creates one or more  
Amazon EC2 instances and an Application Load Balancer. You will be billed for the AWS  
resources used if you create a stack from this template.",  
  
    "Parameters" : {  
        "VpcId" : {  
            "Type" : "AWS::EC2::VPC::Id",  
            "Description" : "VpcId of your existing Virtual Private Cloud (VPC)",  
            "ConstraintDescription" : "must be the VPC Id of an existing Virtual Private Cloud."  
        },  
  
        "Subnets" : {  
            "Type" : "List<AWS::EC2::Subnet::Id>",  
            "Description" : "The list of SubnetIds in your Virtual Private Cloud (VPC)",  
            "ConstraintDescription" : "must be a list of at least two existing subnets associated with at  
least two different availability zones. They should be residing in the selected Virtual Private  
Cloud."  
        },  
  
        "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", "m2.2xlarge", "m2.4xlarge",  
                "m3.medium", "m3.large", "m3.xlarge", "m3.2xlarge", "m4.large", "m4.xlarge", "m4.2xlarge",  
                "m4.4xlarge", "m4.10xlarge", "c1.medium", "c1.xlarge", "c3.large", "c3.xlarge", "c3.2xlarge",  
                "c3.4xlarge", "c3.8xlarge", "c4.large", "c4.xlarge", "c4.2xlarge", "c4.4xlarge", "c4.8xlarge",  
                "g2.2xlarge", "g2.8xlarge", "r3.large", "r3.xlarge", "r3.2xlarge", "r3.4xlarge", "r3.8xlarge",  
                "i2.xlarge", "i2.2xlarge", "i2.4xlarge", "i2.8xlarge", "d2.xlarge", "d2.2xlarge", "d2.4xlarge",  
                "d2.8xlarge", "hi1.4xlarge", "hs1.8xlarge", "cr1.8xlarge", "cc2.8xlarge", "cg1.4xlarge"],  
            "ConstraintDescription" : "must be a valid EC2 instance type."  
        }  
    }  
}
```

```

    },
    "KeyName" : {
        "Description" : "Name of an existing EC2 KeyPair to enable SSH access to the instances",
        "Type" : "AWS::EC2::KeyPair::KeyName",
        "Default": "ec2@dharmit",
        "MinLength": "1",
        "MaxLength": "255",
        "AllowedPattern" : "[\\x20-\\x7E]*",
        "ConstraintDescription" : "can contain only ASCII characters."
    },
    "Location" : {
        "Description" : "The IP address range that can be used to SSH to the EC2 instances",
        "Type": "String",
        "MinLength": "9",
        "MaxLength": "18",
        "Default": "71.255.89.254/32",
        "AllowedPattern": "(\\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."
    }
},
"Mappings" : {
    "Region2Examples" : {
        "us-east-1" : { "Examples" : "https://s3.amazonaws.com/cloudformation-examples-us-east-1" },
        "us-west-2" : { "Examples" : "https://s3-us-west-2.amazonaws.com/cloudformation-examples-us-west-2" },
        "us-west-1" : { "Examples" : "https://s3-us-west-1.amazonaws.com/cloudformation-examples-us-west-1" },
        "eu-west-1" : { "Examples" : "https://s3-eu-west-1.amazonaws.com/cloudformation-examples-eu-west-1" },
        "eu-west-2" : { "Examples" : "https://s3-eu-west-2.amazonaws.com/cloudformation-examples-eu-west-2" },
        "eu-west-3" : { "Examples" : "https://s3-eu-west-3.amazonaws.com/cloudformation-examples-eu-west-3" },
        "eu-north-1" : { "Examples" : "https://s3-eu-north-1.amazonaws.com/cloudformation-examples-eu-north-1" },
        "eu-central-1" : { "Examples" : "https://s3-eu-central-1.amazonaws.com/cloudformation-examples-eu-central-1" },
        "ap-southeast-1" : { "Examples" : "https://s3-ap-southeast-1.amazonaws.com/cloudformation-examples-ap-southeast-1" },
        "ap-northeast-1" : { "Examples" : "https://s3-ap-northeast-1.amazonaws.com/cloudformation-examples-ap-northeast-1" },
        "ap-northeast-2" : { "Examples" : "https://s3-ap-northeast-2.amazonaws.com/cloudformation-examples-ap-northeast-2" },
    }
}

```

```

    "ap-northeast-3" : { "Examples" : "https://s3-ap-northeast-
3.amazonaws.com/cloudformation-examples-ap-northeast-3" },
    "ap-southeast-2" : { "Examples" : "https://s3-ap-southeast-
2.amazonaws.com/cloudformation-examples-ap-southeast-2" },
    "ap-south-1" : { "Examples" : "https://s3-ap-south-1.amazonaws.com/cloudformation-
examples-ap-south-1" },
    "us-east-2" : { "Examples" : "https://s3-us-east-2.amazonaws.com/cloudformation-
examples-us-east-2" },
    "ca-central-1" : { "Examples" : "https://s3-ca-central-1.amazonaws.com/cloudformation-
examples-ca-central-1" },
    "sa-east-1" : { "Examples" : "https://s3-sa-east-1.amazonaws.com/cloudformation-
examples-sa-east-1" },
    "cn-north-1" : { "Examples" : "https://s3.cn-north-1.amazonaws.com.cn/cloudformation-
examples-cn-north-1" },
    "cn-northwest-1" : { "Examples" : "https://s3.cn-northwest-
1.amazonaws.com.cn/cloudformation-examples-cn-northwest-1" }
}

,
"AWSInstanceType2Arch" : {
    "t1.micro" : { "Arch" : "HVM64" },
    "t2.nano" : { "Arch" : "HVM64" },
    "t2.micro" : { "Arch" : "HVM64" },
    "t2.small" : { "Arch" : "HVM64" },
    "t2.medium" : { "Arch" : "HVM64" },
    "t2.large" : { "Arch" : "HVM64" },
    "m1.small" : { "Arch" : "HVM64" },
    "m1.medium" : { "Arch" : "HVM64" },
    "m1.large" : { "Arch" : "HVM64" },
    "m1.xlarge" : { "Arch" : "HVM64" },
    "m2.xlarge" : { "Arch" : "HVM64" },
    "m2.2xlarge" : { "Arch" : "HVM64" },
    "m2.4xlarge" : { "Arch" : "HVM64" },
    "m3.medium" : { "Arch" : "HVM64" },
    "m3.large" : { "Arch" : "HVM64" },
    "m3.xlarge" : { "Arch" : "HVM64" },
    "m3.2xlarge" : { "Arch" : "HVM64" },
    "m4.large" : { "Arch" : "HVM64" },
    "m4.xlarge" : { "Arch" : "HVM64" },
    "m4.2xlarge" : { "Arch" : "HVM64" },
    "m4.4xlarge" : { "Arch" : "HVM64" },
    "m4.10xlarge" : { "Arch" : "HVM64" },
    "c1.medium" : { "Arch" : "HVM64" },
    "c1.xlarge" : { "Arch" : "HVM64" },
    "c3.large" : { "Arch" : "HVM64" },
    "c3.xlarge" : { "Arch" : "HVM64" },
    "c3.2xlarge" : { "Arch" : "HVM64" },
    "c3.4xlarge" : { "Arch" : "HVM64" },
    "c3.8xlarge" : { "Arch" : "HVM64" },
}

```

```

"c4.large" : { "Arch" : "HVM64" },
"c4.xlarge" : { "Arch" : "HVM64" },
"c4.2xlarge" : { "Arch" : "HVM64" },
"c4.4xlarge" : { "Arch" : "HVM64" },
"c4.8xlarge" : { "Arch" : "HVM64" },
"g2.2xlarge" : { "Arch" : "HVMG2" },
"g2.8xlarge" : { "Arch" : "HVMG2" },
"r3.large" : { "Arch" : "HVM64" },
"r3.xlarge" : { "Arch" : "HVM64" },
"r3.2xlarge" : { "Arch" : "HVM64" },
"r3.4xlarge" : { "Arch" : "HVM64" },
"r3.8xlarge" : { "Arch" : "HVM64" },
"i2.xlarge" : { "Arch" : "HVM64" },
"i2.2xlarge" : { "Arch" : "HVM64" },
"i2.4xlarge" : { "Arch" : "HVM64" },
"i2.8xlarge" : { "Arch" : "HVM64" },
"d2.xlarge" : { "Arch" : "HVM64" },
"d2.2xlarge" : { "Arch" : "HVM64" },
"d2.4xlarge" : { "Arch" : "HVM64" },
"d2.8xlarge" : { "Arch" : "HVM64" },
"hi1.4xlarge" : { "Arch" : "HVM64" },
"hs1.8xlarge" : { "Arch" : "HVM64" },
"cr1.8xlarge" : { "Arch" : "HVM64" },
"cc2.8xlarge" : { "Arch" : "HVM64" }
},
"AWSInstanceType2NATArch" : {
  "t1.micro" : { "Arch" : "NATHVM64" },
  "t2.nano" : { "Arch" : "NATHVM64" },
  "t2.micro" : { "Arch" : "NATHVM64" },
  "t2.small" : { "Arch" : "NATHVM64" },
  "t2.medium" : { "Arch" : "NATHVM64" },
  "t2.large" : { "Arch" : "NATHVM64" },
  "m1.small" : { "Arch" : "NATHVM64" },
  "m1.medium" : { "Arch" : "NATHVM64" },
  "m1.large" : { "Arch" : "NATHVM64" },
  "m1.xlarge" : { "Arch" : "NATHVM64" },
  "m2.xlarge" : { "Arch" : "NATHVM64" },
  "m2.2xlarge" : { "Arch" : "NATHVM64" },
  "m2.4xlarge" : { "Arch" : "NATHVM64" },
  "m3.medium" : { "Arch" : "NATHVM64" },
  "m3.large" : { "Arch" : "NATHVM64" },
  "m3.xlarge" : { "Arch" : "NATHVM64" },
  "m3.2xlarge" : { "Arch" : "NATHVM64" },
  "m4.large" : { "Arch" : "NATHVM64" },
  "m4.xlarge" : { "Arch" : "NATHVM64" },
  "m4.2xlarge" : { "Arch" : "NATHVM64" },
  "m4.4xlarge" : { "Arch" : "NATHVM64" }
}

```

```

    "m4.10xlarge" : { "Arch" : "NATHVM64" },
    "c1.medium"   : { "Arch" : "NATHVM64" },
    "c1.xlarge"   : { "Arch" : "NATHVM64" },
    "c3.large"    : { "Arch" : "NATHVM64" },
    "c3.xlarge"   : { "Arch" : "NATHVM64" },
    "c3.2xlarge"  : { "Arch" : "NATHVM64" },
    "c3.4xlarge"  : { "Arch" : "NATHVM64" },
    "c3.8xlarge"  : { "Arch" : "NATHVM64" },
    "c4.large"    : { "Arch" : "NATHVM64" },
    "c4.xlarge"   : { "Arch" : "NATHVM64" },
    "c4.2xlarge"  : { "Arch" : "NATHVM64" },
    "c4.4xlarge"  : { "Arch" : "NATHVM64" },
    "c4.8xlarge"  : { "Arch" : "NATHVM64" },
    "g2.2xlarge"  : { "Arch" : "NATHVMG2" },
    "g2.8xlarge"  : { "Arch" : "NATHVMG2" },
    "r3.large"    : { "Arch" : "NATHVM64" },
    "r3.xlarge"   : { "Arch" : "NATHVM64" },
    "r3.2xlarge"  : { "Arch" : "NATHVM64" },
    "r3.4xlarge"  : { "Arch" : "NATHVM64" },
    "r3.8xlarge"  : { "Arch" : "NATHVM64" },
    "i2.xlarge"   : { "Arch" : "NATHVM64" },
    "i2.2xlarge"  : { "Arch" : "NATHVM64" },
    "i2.4xlarge"  : { "Arch" : "NATHVM64" },
    "i2.8xlarge"  : { "Arch" : "NATHVM64" },
    "d2.xlarge"   : { "Arch" : "NATHVM64" },
    "d2.2xlarge"  : { "Arch" : "NATHVM64" },
    "d2.4xlarge"  : { "Arch" : "NATHVM64" },
    "d2.8xlarge"  : { "Arch" : "NATHVM64" },
    "hi1.4xlarge" : { "Arch" : "NATHVM64" },
    "hs1.8xlarge" : { "Arch" : "NATHVM64" },
    "cr1.8xlarge" : { "Arch" : "NATHVM64" },
    "cc2.8xlarge" : { "Arch" : "NATHVM64" }
  }
}

,
  "AWSRegionArch2AMI" : {
    "us-east-1"      : {"HVM64" : "ami-0080e4c5bc078760e", "HVMG2" : "ami-0aeb704d503081ea6"},
    "us-west-2"      : {"HVM64" : "ami-01e24be29428c15b2", "HVMG2" : "ami-0fe84a5b4563d8f27"},
    "us-west-1"      : {"HVM64" : "ami-0ec6517f6edbf8044", "HVMG2" : "ami-0a7fc72dc0e51aa77"},
    "eu-west-1"      : {"HVM64" : "ami-08935252a36e25f85", "HVMG2" : "ami-0d5299b1c6112c3c7"},
    "eu-west-2"      : {"HVM64" : "ami-01419b804382064e4", "HVMG2" : "NOT_SUPPORTED"},
    "eu-west-3"      : {"HVM64" : "ami-0dd7e7ed60da8fb83", "HVMG2" : "NOT_SUPPORTED"},
    "eu-central-1"   : {"HVM64" : "ami-0cfbf4f6db41068ac", "HVMG2" : "ami-0aa1822e3eb913a11"},
    "eu-north-1"     : {"HVM64" : "ami-86fe70f8", "HVMG2" : "ami-32d55b4c"}
  }
}

```

```

    "ap-northeast-1" : {"HVM64" : "ami-00a5245b4816c38e6", "HVMG2" : "ami-09d0e0e099ecabba2"},
    "ap-northeast-2" : {"HVM64" : "ami-00dc207f8ba6dc919", "HVMG2" : "NOT_SUPPORTED"},
    "ap-northeast-3" : {"HVM64" : "ami-0b65f69a5c11f3522", "HVMG2" : "NOT_SUPPORTED"},
    "ap-southeast-1" : {"HVM64" : "ami-05b3bcf7f311194b3", "HVMG2" : "ami-0e46ce0d6a87dc979"},
    "ap-southeast-2" : {"HVM64" : "ami-02fd0b06f06d93dfc", "HVMG2" : "ami-0c0ab057a101d8ff2"},
    "ap-south-1" : {"HVM64" : "ami-0ad42f4f66f6c1cc9", "HVMG2" : "ami-0244c1d42815af84a"},
    "us-east-2" : {"HVM64" : "ami-0cd3dfa4e37921605", "HVMG2" : "NOT_SUPPORTED"},
    "ca-central-1" : {"HVM64" : "ami-07423fb63ea0a0930", "HVMG2" : "NOT_SUPPORTED"},
    "sa-east-1" : {"HVM64" : "ami-05145e0b28ad8e0b2", "HVMG2" : "NOT_SUPPORTED"},
    "cn-north-1" : {"HVM64" : "ami-053617c9d818c1189", "HVMG2" : "NOT_SUPPORTED"},
    "cn-northwest-1" : {"HVM64" : "ami-0f7937761741dc640", "HVMG2" : "NOT_SUPPORTED"}
  }
}

"Resources" : {
  "WebServerGroup" : {
    "Type" : "AWS::AutoScaling::AutoScalingGroup",
    "Properties" : {
      "VPCZoneIdentifier" : { "Ref" : "Subnets" },
      "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
      "MinSize" : "2",
      "MaxSize" : "2",
      "TargetGroupARNs" : [ { "Ref" : "ALBTargetGroup" } ]
    },
    "CreationPolicy" : {
      "ResourceSignal" : {
        "Timeout" : "PT15M",
        "Count" : "2"
      }
    },
    "UpdatePolicy": {
      "AutoScalingRollingUpdate": {
        "MinInstancesInService": "1",
        "MaxBatchSize": "1",
        "PauseTime" : "PT15M",
        "WaitOnResourceSignals": "true"
      }
    }
  },
  "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\"/>",
        "<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"
    },
    "/etc/cfn/hooks.d/cfn-auto-reloader.conf" : {
      "content": { "Fn::Join" : ["", [
        "[cfn-auto-reloader-hook]\n",
        "triggers=post.update\n",
        "path=Resources.LaunchConfig.Metadata.AWS::CloudFormation::Init\n",
        "action=/opt/aws/bin/cfn-init -v ",
        "  --stack ", { "Ref" : "AWS::StackName" },
        "  --resource LaunchConfig ",
        "  --region ", { "Ref" : "AWS::Region" }, "\n",
        "runas=root\n"
      ]]},
      "mode" : "000400",
      "owner" : "root",
      "group" : "root"
    }
  }
}

```

```

        },
        "services" : {
            "sysvinit" : {
                "httpd" : { "enabled" : "true", "ensureRunning" : "true" },
                "cfn-hup" : { "enabled" : "true", "ensureRunning" : "true",
                               "files" : ["/etc/cfn/cfn-hup.conf", "/etc/cfn/hooks.d/cfn-auto-reloader.conf"]}
            }
        }
    }
},
"Properties" : {
    "KeyName" : { "Ref" : "KeyName" },
    "ImageId" : { "Fn::FindInMap" : [ "AWSRegionArch2AMI", { "Ref" : "AWS::Region" },
                                      { "Fn::FindInMap" : [ "AWSInstanceType2Arch", { "Ref" : "InstanceType" }, "Arch" ]
] },
    "SecurityGroups" : [ { "Ref" : "InstanceSecurityGroup" } ],
    "InstanceType" : { "Ref" : "InstanceType" },
    "UserData" : { "Fn::Base64" : { "Fn::Join" : [ "", [
        "#!/bin/bash -xe\n",
        "yum update -y aws-cfn-bootstrap\n",
        "/opt/aws/bin/cfn-init -v ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource LaunchConfig ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n",
        "/opt/aws/bin/cfn-signal -e $? ",
        "    --stack ", { "Ref" : "AWS::StackName" },
        "    --resource WebServerGroup ",
        "    --region ", { "Ref" : "AWS::Region" }, "\n"
    ]]} }
},
"ApplicationLoadBalancer" : {
    "Type" : "AWS::ElasticLoadBalancingV2::LoadBalancer",
    "Properties" : {
        "Subnets" : { "Ref" : "Subnets" }
    }
},
"ALBListener" : {
    "Type" : "AWS::ElasticLoadBalancingV2::Listener",
    "Properties" : {
        "DefaultActions" : [
            {
                "Type" : "forward",

```

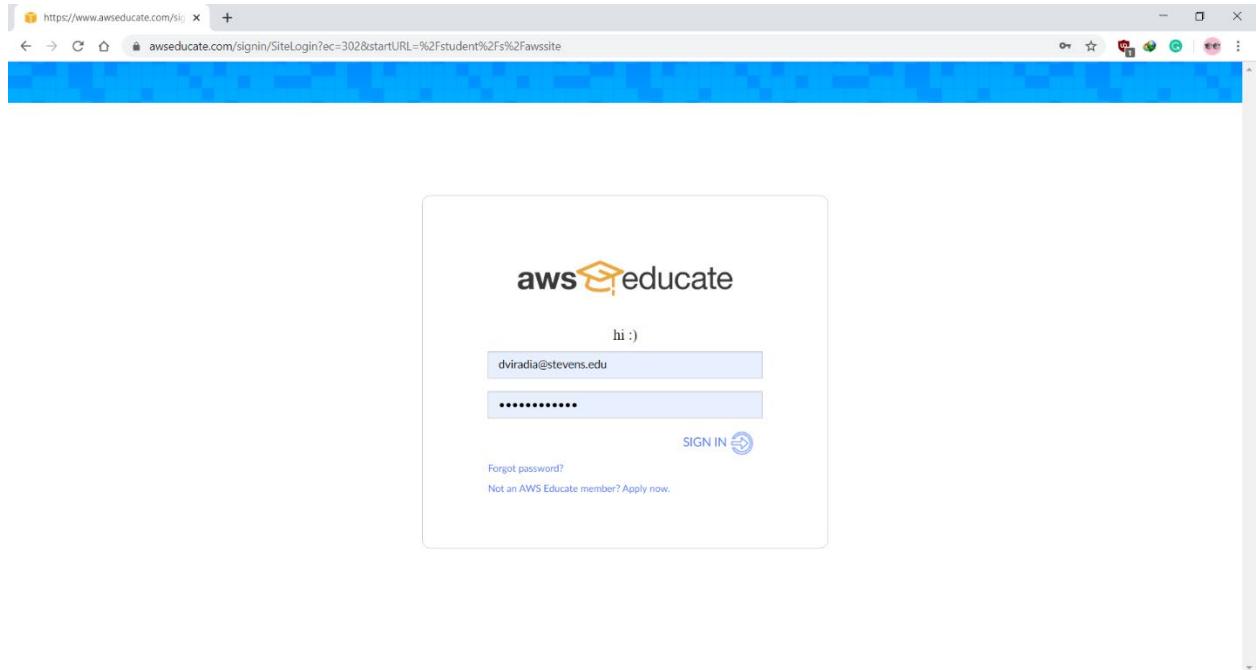
```

        "TargetGroupArn" : { "Ref" : "ALBTargetGroup" }
    }],
    "LoadBalancerArn" : { "Ref" : "ApplicationLoadBalancer" },
    "Port" : "80",
    "Protocol" : "HTTP"
}
},
"ALBTargetGroup" : {
    "Type" : "AWS::ElasticLoadBalancingV2::TargetGroup",
    "Properties" : {
        "HealthCheckIntervalSeconds" : 30,
        "HealthCheckTimeoutSeconds" : 5,
        "HealthyThresholdCount" : 3,
        "Port" : 80,
        "Protocol" : "HTTP",
        "UnhealthyThresholdCount" : 5,
        "VpcId" : {"Ref" : "VpcId"}
    }
},
"InstanceSecurityGroup" : {
    "Type" : "AWS::EC2::SecurityGroup",
    "Properties" : {
        "GroupDescription" : "Enable SSH access and HTTP access on the configured port",
        "GroupName" : "Load Balancer Security Group",
        "SecurityGroupIngress" :
            [ { "IpProtocol" : "tcp", "FromPort" : "22", "ToPort" : "22", "CidrIp" : { "Ref" : "Location" } },
              { "IpProtocol" : "tcp", "FromPort" : "80", "ToPort" : "80", "CidrIp" : "0.0.0.0/0" } ],
        "VpcId" : { "Ref" : "VpcId" }
    }
},
"Outputs" : {
    "URL" : {
        "Description" : "URL of the website",
        "Value" : { "Fn::Join" : [ "", [ "http://", { "Fn::GetAtt" : [
            "ApplicationLoadBalancer", "DNSName" ] } ] ] }
    }
}
}

```

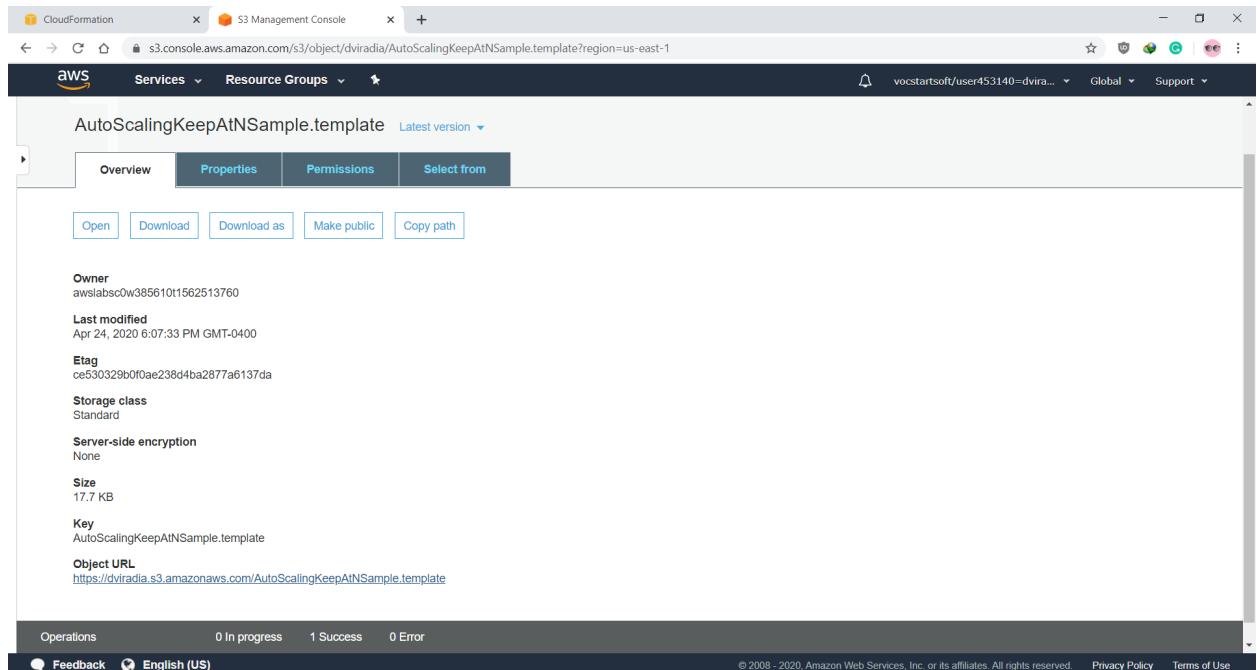
Step for Creating a CloudFormation

- First login into your AWS account

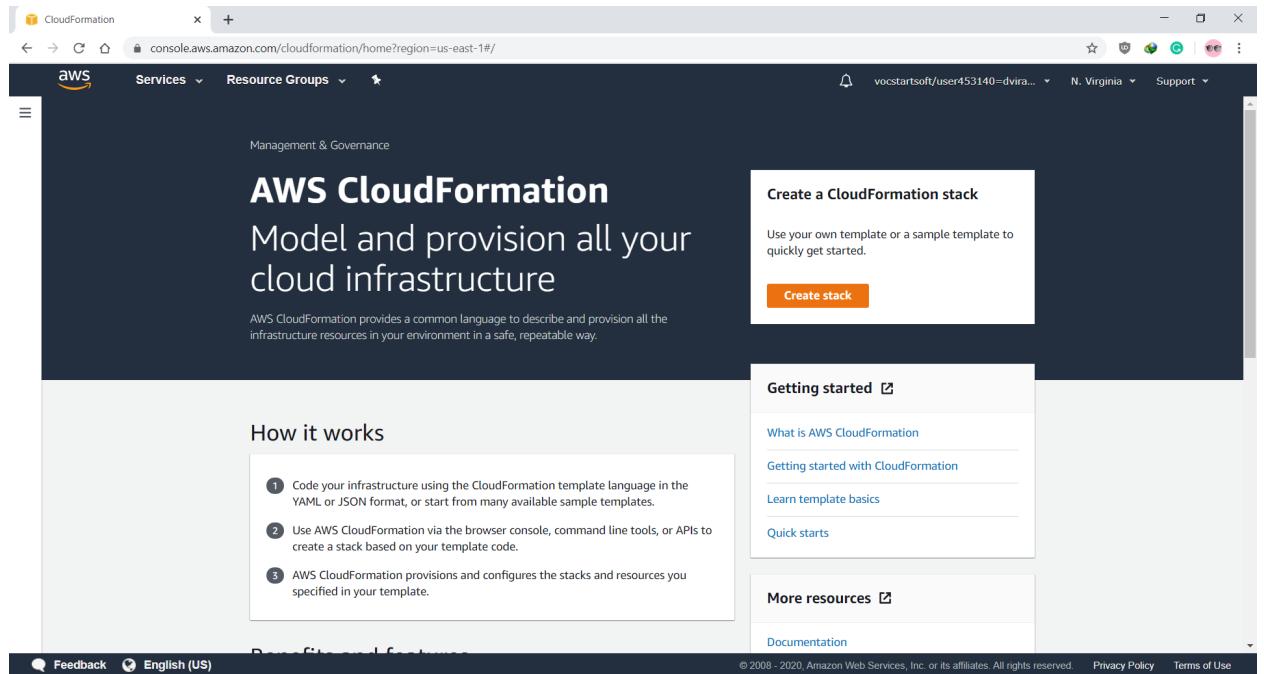


- Upload Template to S3 Bucket and give Public Access

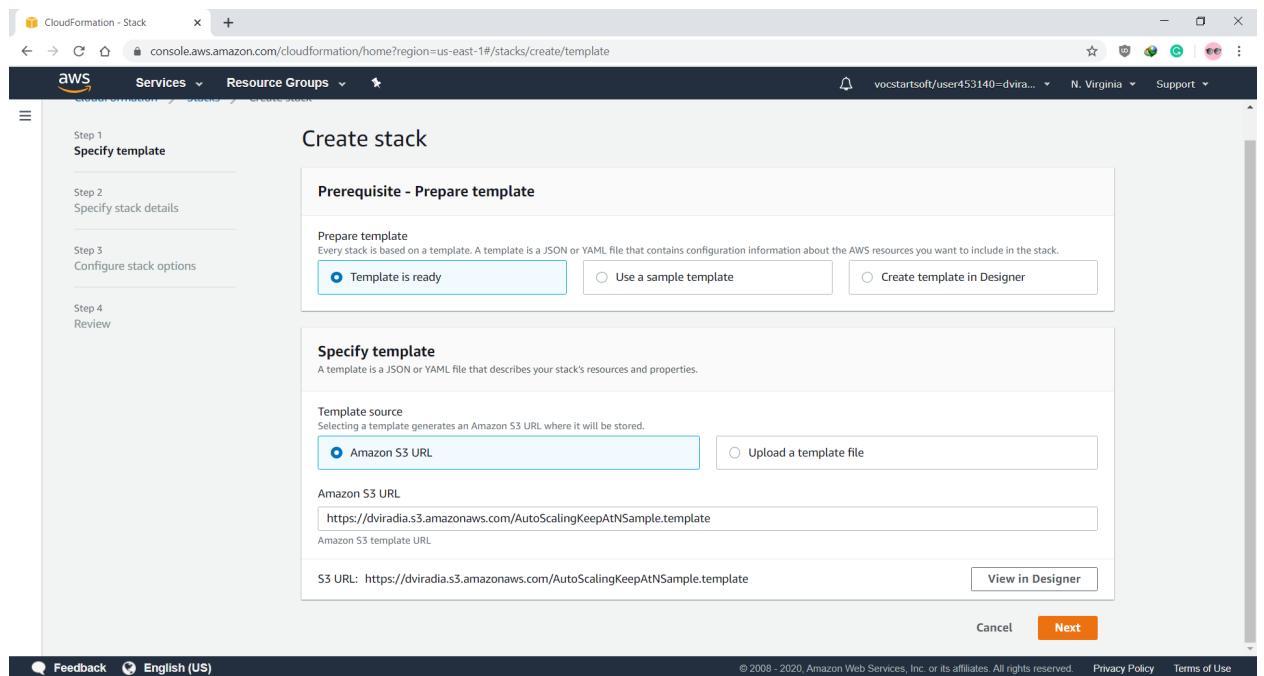
I have used already created bucket from previous Lab



- On AWS Dashboard go to CloudFormation Services and Click on Create Stack button



- Now, selecting a recently created template from the S3 directory and click on next



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

CloudFormation - Stack

Services Resource Groups

Step 3 Configure stack options Step 4 Review

Parameters

Stack name: Load-Balancer

InstanceType: WebServer EC2 instance type t2.micro

KeyName: ec2@dharmit

Location: 71.255.89.254/32

Subnets: subnet-a96f4dff (172.31.32.0/20), subnet-9c6793 (172.31.64.0/20), subnet-08473236 (172.31.48.0/20)

VpcId: vpc-1993e163 (172.31.0.0/16)

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

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

CloudFormation - Stack

Services Resource Groups

Notification options

No notification options

There are no notification options defined

Stack creation options

Rollback on failure: Enabled

Timeout: -

Termination protection: Disabled

Quick-create link Cancel Previous Create change set **Create stack**

Feedback English (US) © 2008 - 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved. Privacy Policy Terms of Use

- After the creation of stack successfully, the status changes to “CREATE_COMPLETE”. And we can check the Load Balancer URL in the output tab.

The screenshot shows the AWS CloudFormation console with the "Events" tab selected for the "Load-Balancer" stack. The table lists 24 events, all of which are in the "CREATE_COMPLETE" state. The first event is from 2020-04-24 22:41:06 UTC-0400, and the last event is from 2020-04-24 22:41:03 UTC-0400. Other resources listed include ALBListener, ApplicationLoadBalancer, and WebServerGroup.

Timestamp	Logical ID	Status	Status reason
2020-04-24 22:41:06 UTC-0400	Load-Balancer	CREATE_COMPLETE	-
2020-04-24 22:41:04 UTC-0400	ALBListener	CREATE_COMPLETE	-
2020-04-24 22:41:03 UTC-0400	ALBListener	CREATE_IN_PROGRESS	Resource creation Initiated
2020-04-24 22:41:03 UTC-0400	ALBListener	CREATE_IN_PROGRESS	-
2020-04-24 22:41:00 UTC-0400	ApplicationLoadBalancer	CREATE_COMPLETE	-
2020-04-24 22:39:33 UTC-0400	WebServerGroup	CREATE_COMPLETE	-
2020-04-24 22:39:32 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	Received SUCCESS signal with UniqueId i-0821fb8796d23e03
2020-04-24 22:39:32 UTC-0400	WebServerGroup	CREATE_IN_PROGRESS	Received SUCCESS signal with UniqueId i-07c10a15410e8005

The screenshot shows the AWS CloudFormation console with the "Outputs" tab selected for the "Load-Balancer" stack. There is one output entry named "URL" with the value "http://Load-Appli-1ASK4TEF4GYAB-338121307.us-east-1.elb.amazonaws.com".

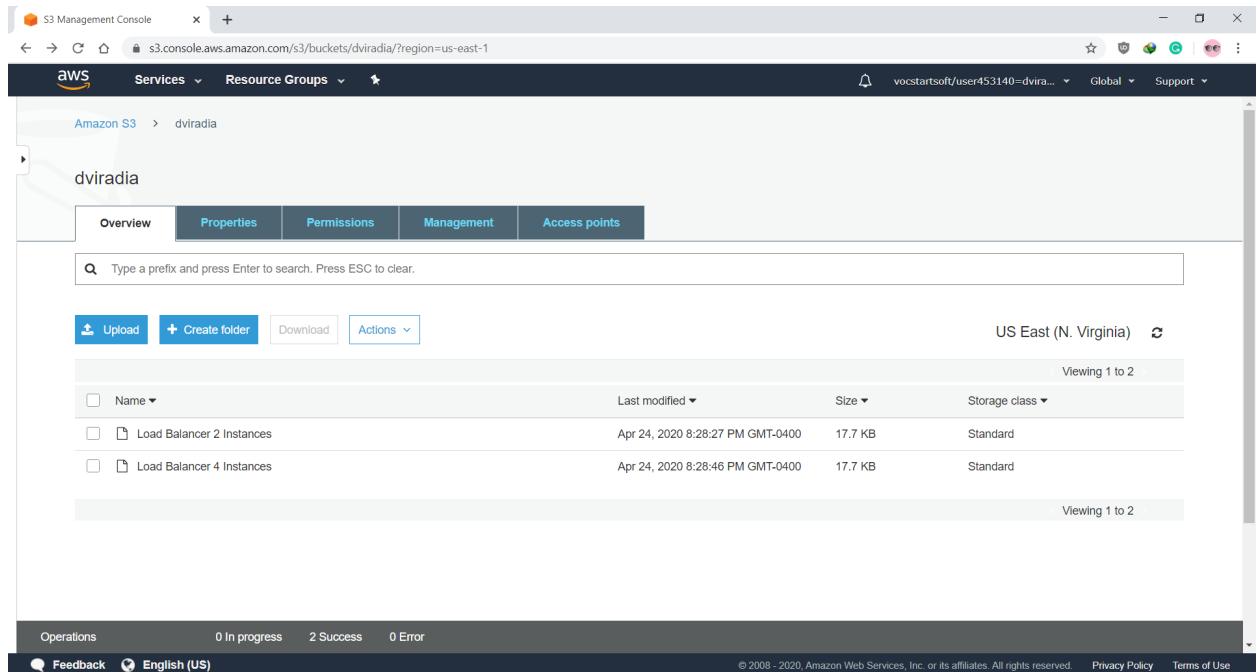
Key	Value	Description	Export name
URL	http://Load-Appli-1ASK4TEF4GYAB-338121307.us-east-1.elb.amazonaws.com	URL of the website	-

Step for Updating a Stack by Amazon Cloud Formation

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

```
"Resources" : {
    "WebServerGroup" : {
        "Type" : "AWS::AutoScaling::AutoScalingGroup",
        "Properties" : {
            "VPCZoneIdentifier" : { "Ref" : "Subnets" },
            "LaunchConfigurationName" : { "Ref" : "LaunchConfig" },
            "MinSize" : "4",
            "MaxSize" : "4",
            "TargetGroupARNs" : [ { "Ref" : "ALBTargetGroup" } ]
        }
    }
},
```

- Now Upload the Updated Template to S3 Bucket



- Now on CloudFormation Dashboard Click on Update tab

The screenshot shows the AWS CloudFormation dashboard with the 'Load-Balancer' stack selected. The 'Stack info' tab is active, displaying the following details:

- Stack ID:** arnaws:cloudformation:us-east-1:498314746917:stack/Load-Balancer/68c92940-8679-11ea-932e-120fb0f0187
- Description:** AWS CloudFormation Sample Template AutoScalingKeepAtNSample: Create a load balanced, Auto Scaled sample website. This example creates an Auto Scaling group behind a load balancer with a simple health check. **WARNING** This template creates one or more Amazon EC2 instances and an Application Load Balancer. You will be billed for the AWS resources used if you create a stack from this template.
- Status:** UPDATE_COMPLETE
- Root stack:** -
- Created time:** 2020-04-24 18:17:39 UTC-0400
- Updated time:** 2020-04-24 18:34:40 UTC-0400

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

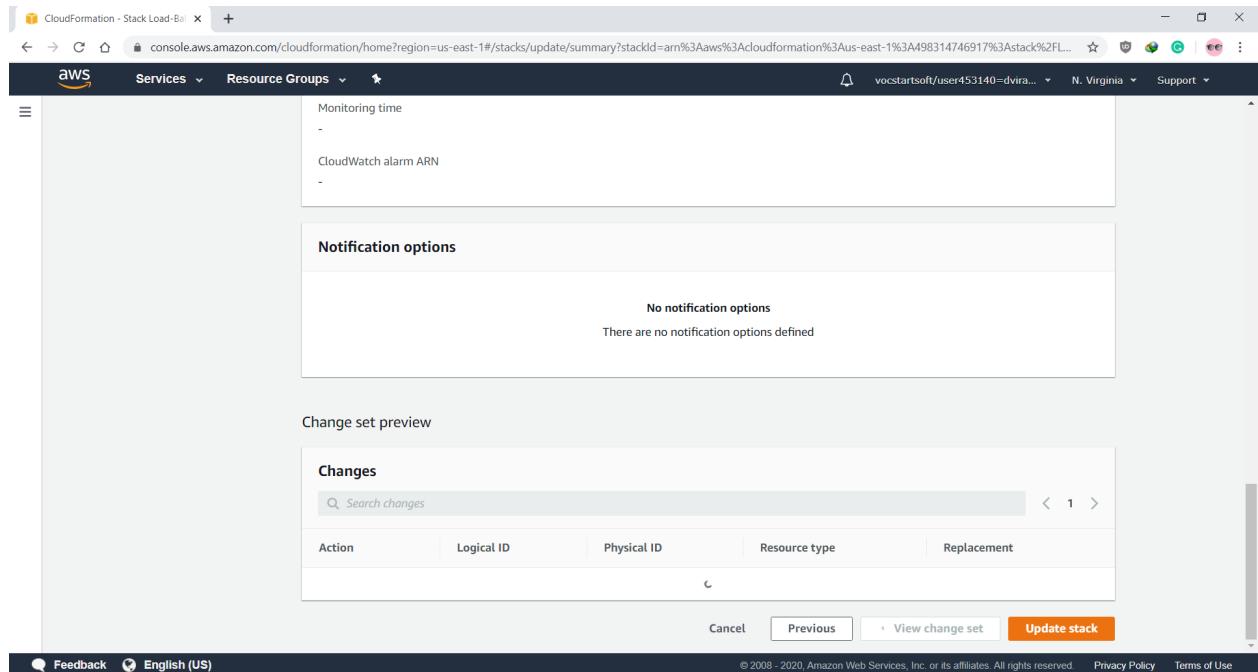
The screenshot shows the 'Update stack' wizard in progress, specifically Step 1: Specify template. The 'Prerequisite - Prepare template' section contains the following options:

- Use current template
- Replace current template
- Edit template in designer

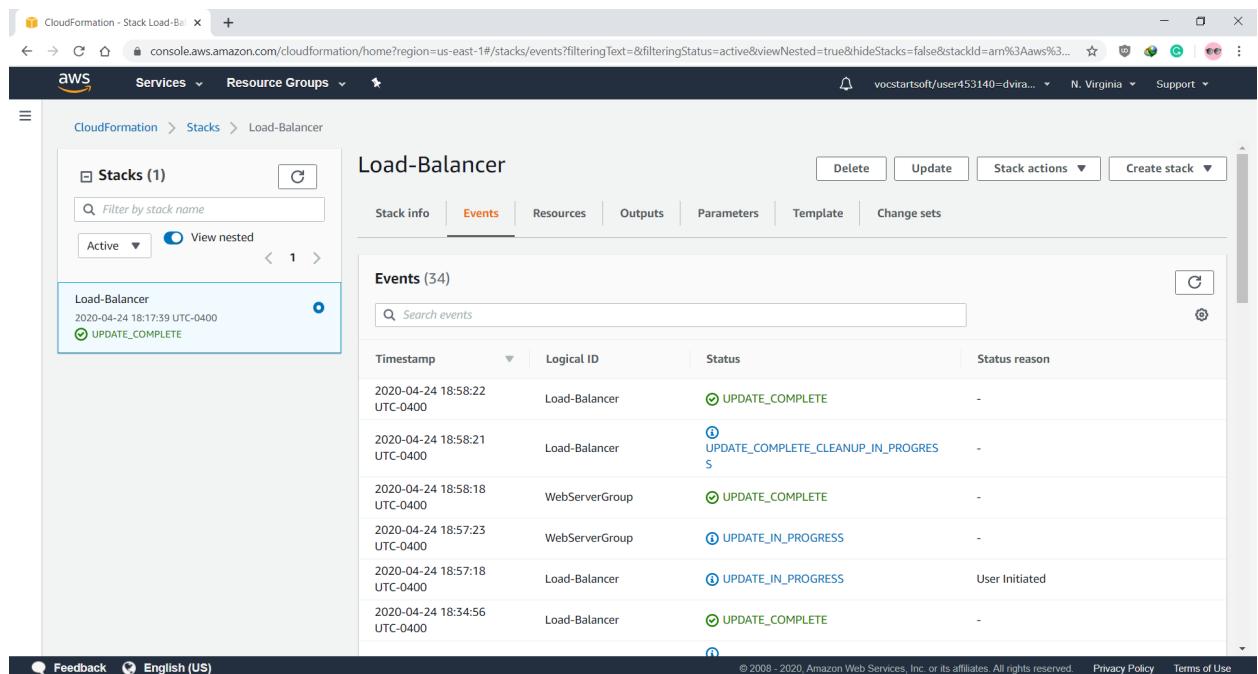
The 'Specify template' section includes the following details:

- Template source:** A note stating "Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack."
- Amazon S3 URL:** The URL is set to <https://s3-external-1.amazonaws.com/cf-templates-pdo9i62b7v49-us-east-1/2020115Ndx-template1uiwo87fbvom>.
- S3 URL:** The URL is also displayed as <https://s3-external-1.amazonaws.com/cf-templates-pdo9i62b7v49-us-east-1/2020115Ndx-template1uiwo87fbvom>.

- After performing same steps, that we did while configuring template for stack creating



- After status change to "UPDATE_COMPLETE". We can check the output of the Load Balancer.



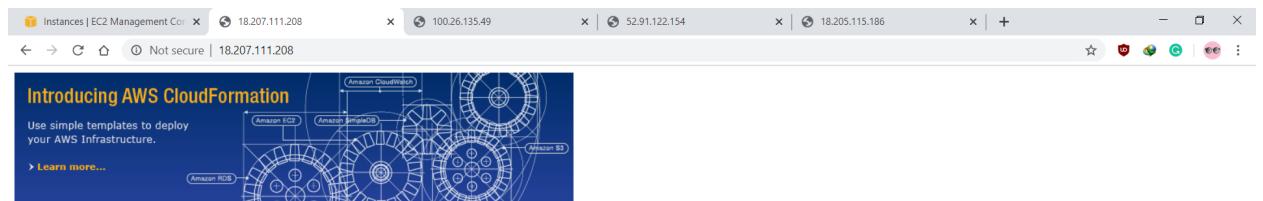
- Click on link from output tab to view Server page



Congratulations, you have successfully launched the AWS CloudFormation sample.

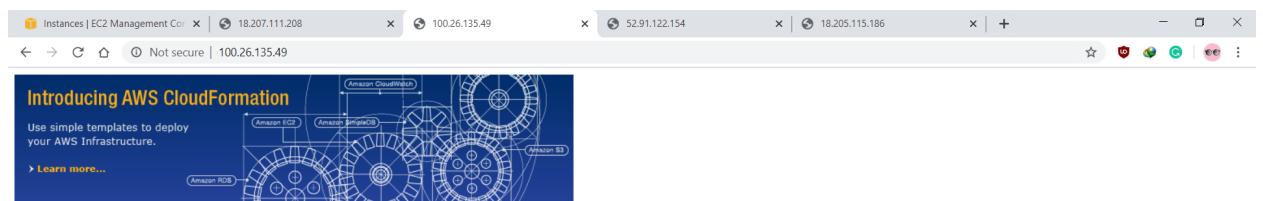
- After successful update of stack, we can check that the instances are running on the servers

Server 1



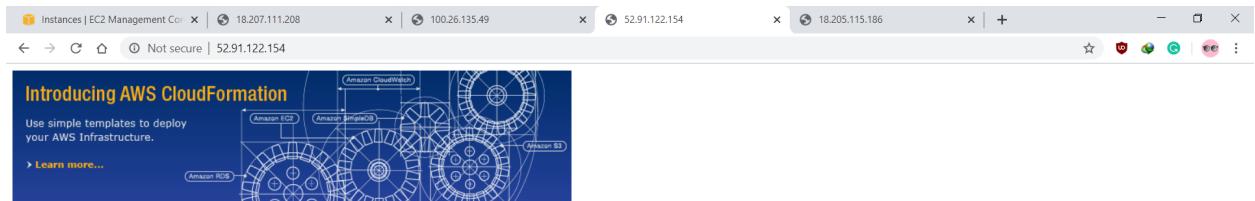
Congratulations, you have successfully launched the AWS CloudFormation sample.

Server 2



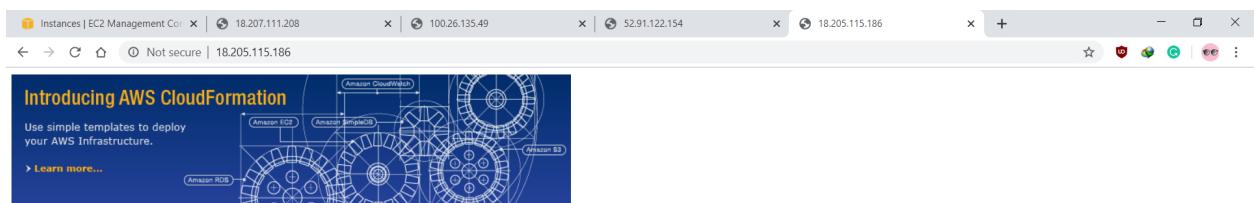
Congratulations, you have successfully launched the AWS CloudFormation sample.

Server 3



Congratulations, you have successfully launched the AWS CloudFormation sample.

Server 4



Congratulations, you have successfully launched the AWS CloudFormation sample.

Steps to change Servers index.html file on Amazon EC2 instance

- Access the AWS Instance using command line by executing following command

```
$ ssh -i "ec2@dharmit.pem" ec2-user@ec2-3-227-19-192.compute-1.amazonaws.com
```

```
cat Select ec2-user@ip-172-31-2-242:~  
C:\Users\dharm\Desktop\Cloud Computing>ssh -i "ec2@dharmit.pem" ec2-user@ec2-3-227-19-192.compute-1.amazonaws.com  
The authenticity of host 'ec2-3-227-19-192.compute-1.amazonaws.com (3.227.19.192)' can't be established.  
ECDSA key fingerprint is SHA256:RxWcEHVLr6T7MpXgG5oC/0wtXujxZC4/EsoLe6xoxrA.  
Are you sure you want to continue connecting (yes/no)? yes  
Warning: Permanently added 'ec2-3-227-19-192.compute-1.amazonaws.com,3.227.19.192' (ECDSA) to the list of known hosts.  
[| ( ) / Amazon Linux AMI  
__| \__|__|  
  
[| ( ) / Amazon Linux AMI  
__| \__|__|  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
26 package(s) needed for security, out of 47 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-2-242 ~]$
```

- Visit the index.html file location by executing following command

```
$ cd /var/www/html
```

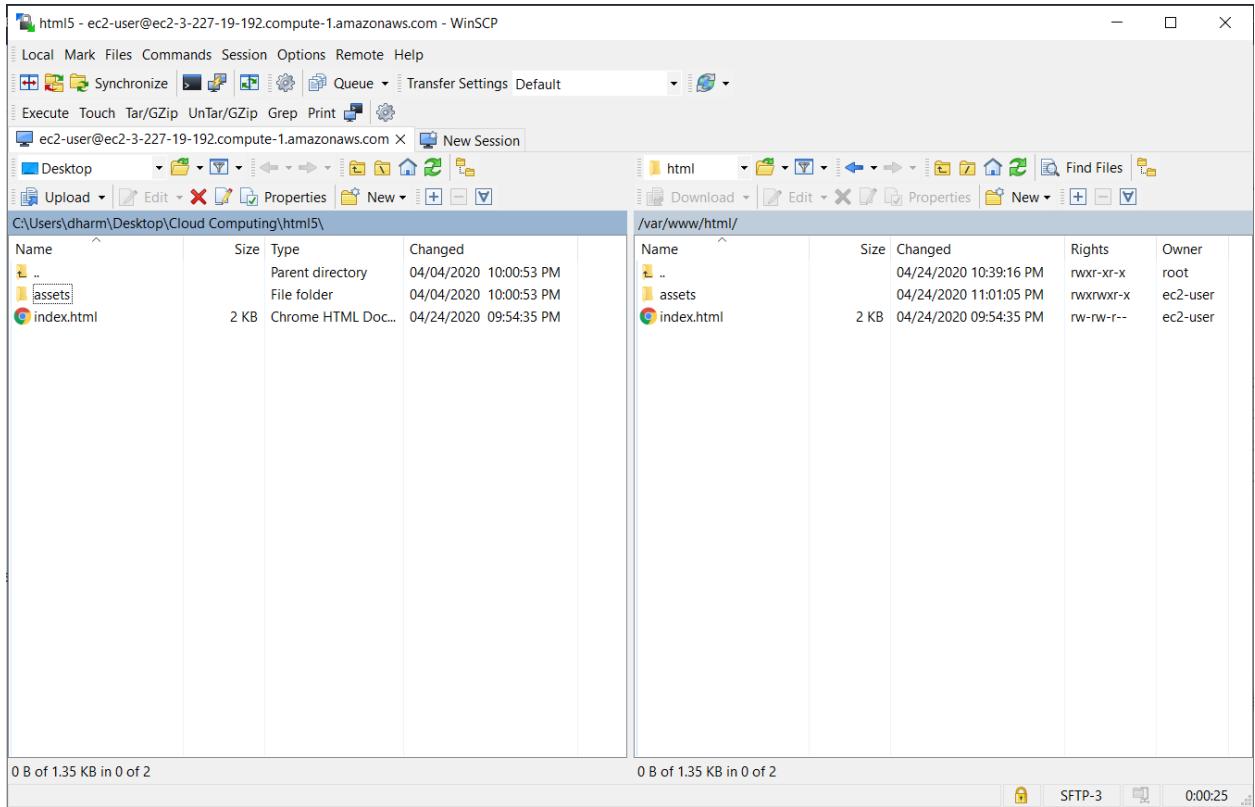
```
cat ec2-user@ip-172-31-2-242:/var/www  
[ec2-user@ip-172-31-2-242 ~]$ cd /var/www/  
[ec2-user@ip-172-31-2-242 www]$
```

- Change the directory permission to allow ftp upload at the location

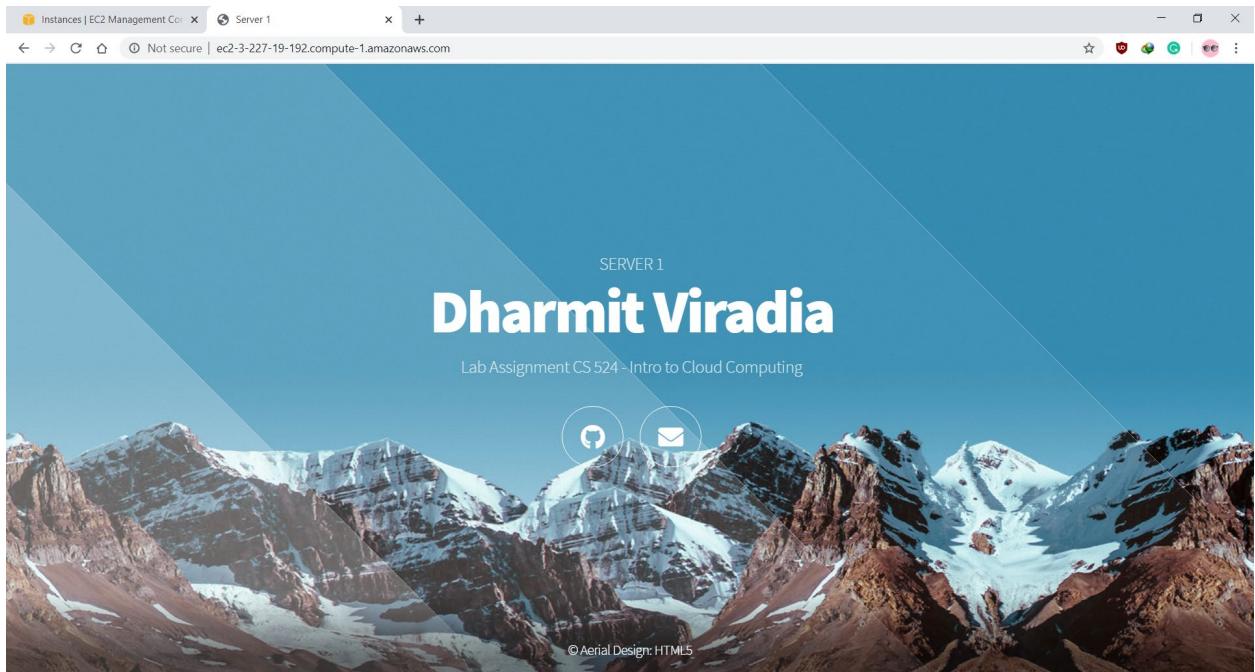
```
$ sudo chmod -R 777 html  
$ cd html
```

```
cat ec2-user@ip-172-31-2-242:/var/www/html  
[ec2-user@ip-172-31-2-242 www]$ sudo chmod -R 777 html  
[ec2-user@ip-172-31-2-242 www]$ cd html  
[ec2-user@ip-172-31-2-242 html]$ ls  
index.html  
[ec2-user@ip-172-31-2-242 html]$
```

- Now access the AWS directory using WinSCP a famous FTP uploader

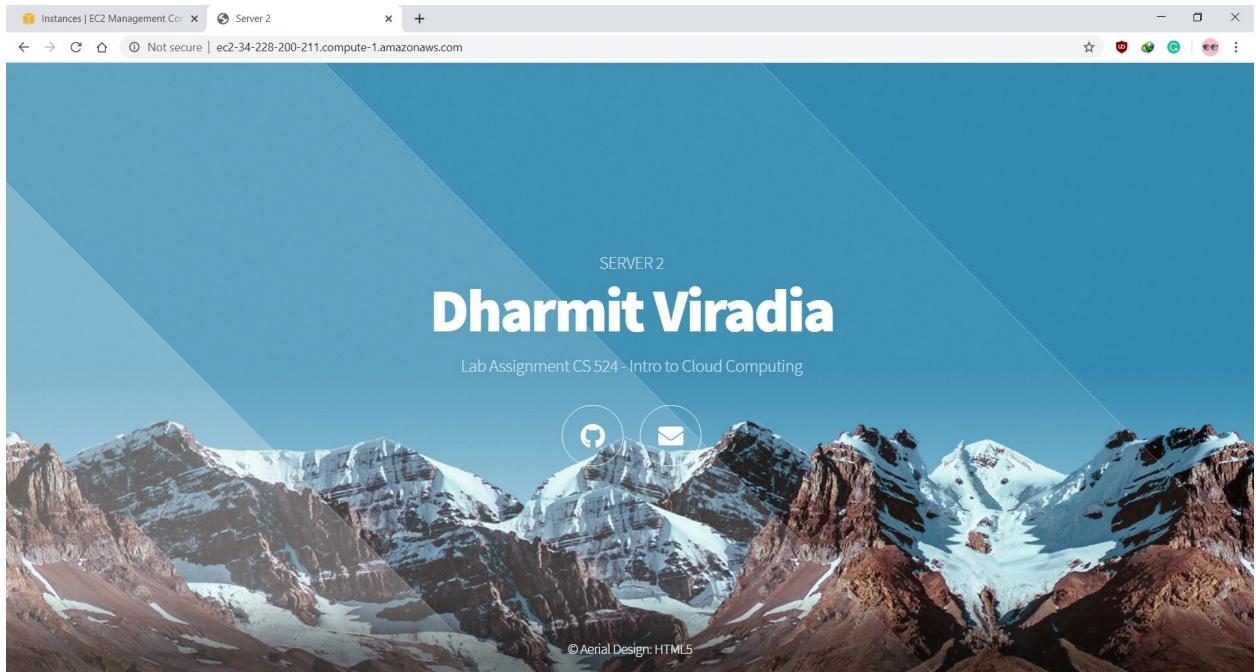


- Actual page on browser after editing index.html file over AWS EC2 for “Server 1” or Instance number 1.

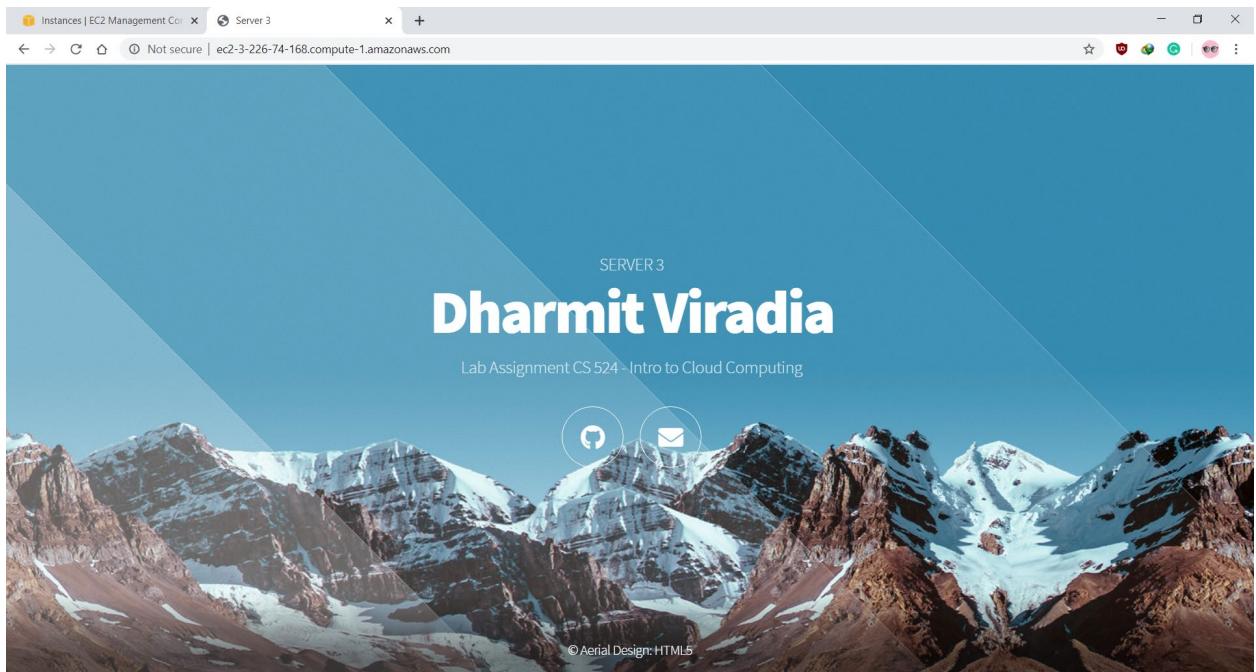


- Now, repeat the same steps to edit the index.html file on each instance you want. I have edited the index.html file for all my four instances i.e. Server 1, Server 2, Server 3, and Server 4.

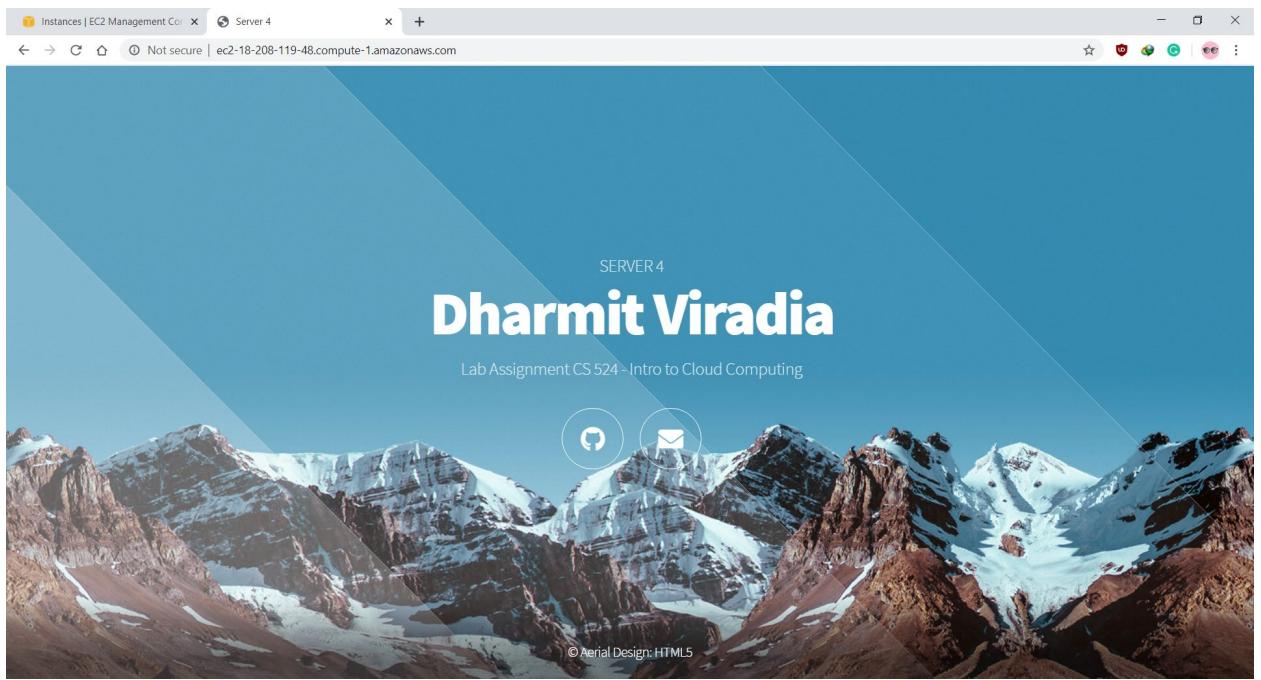
Server 2



Server 3

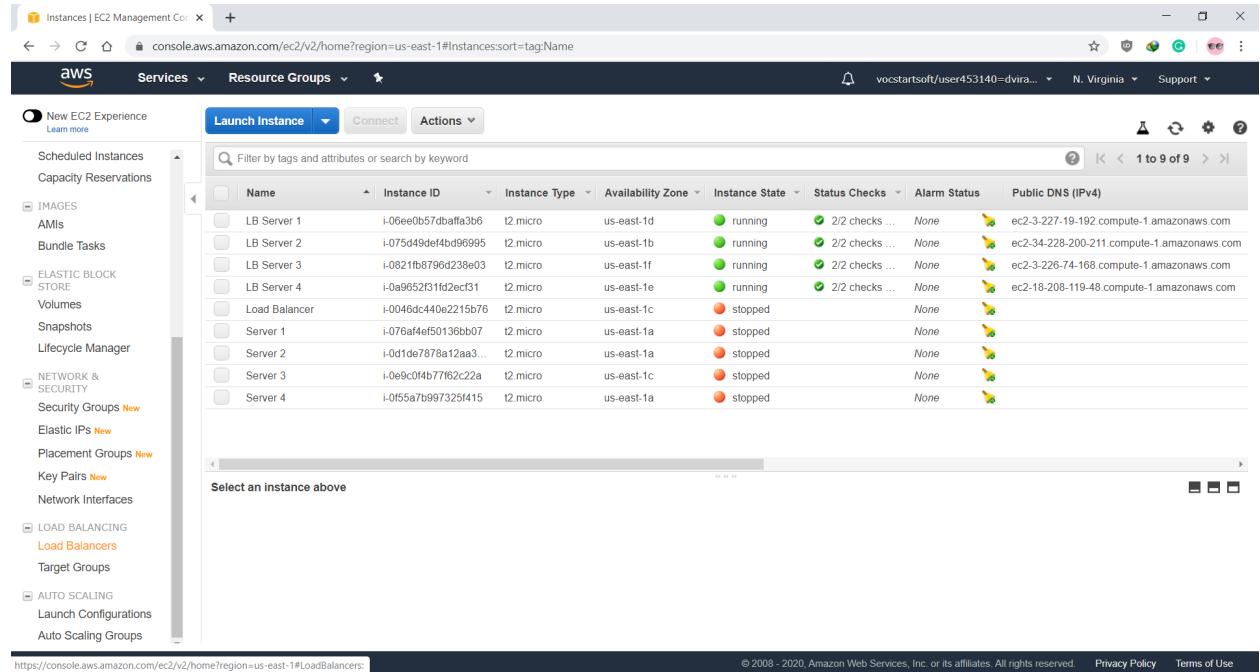


Server 4



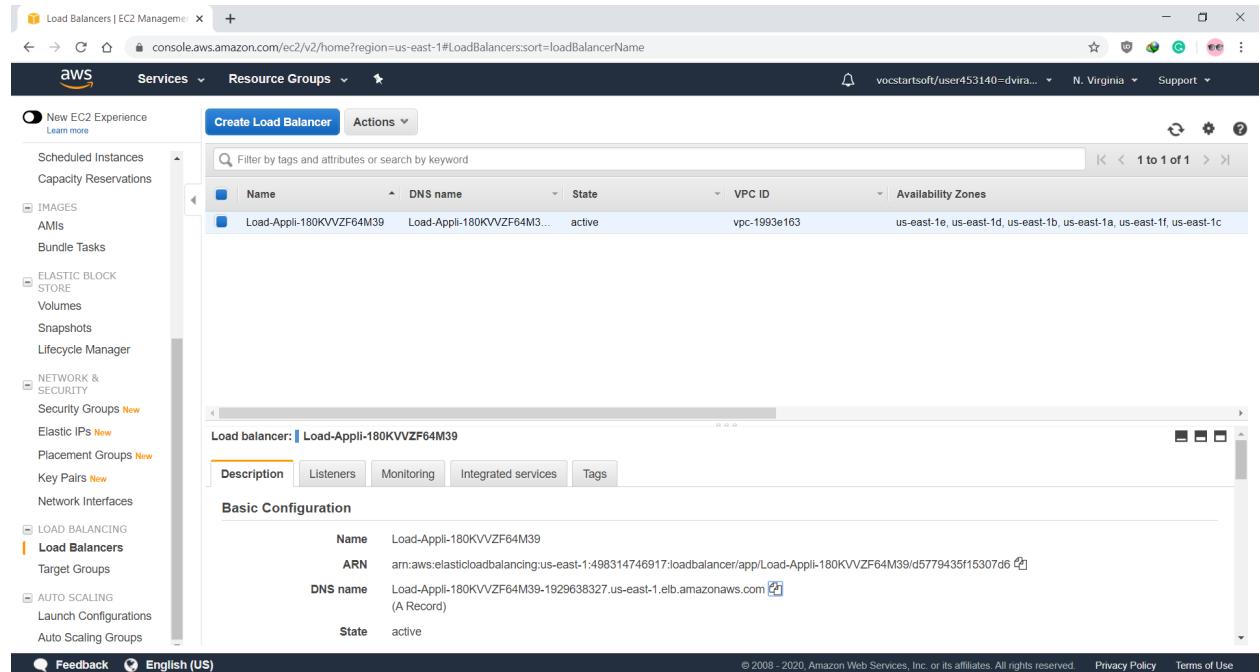
Steps to check working of Load Balancer

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



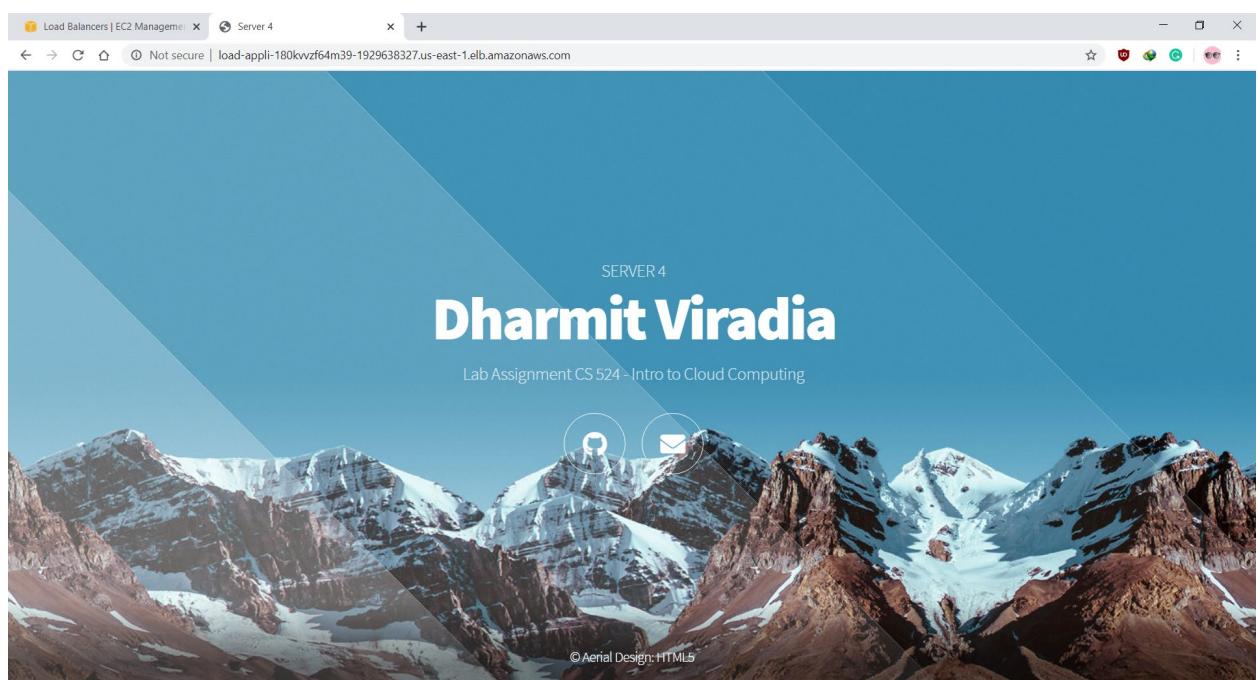
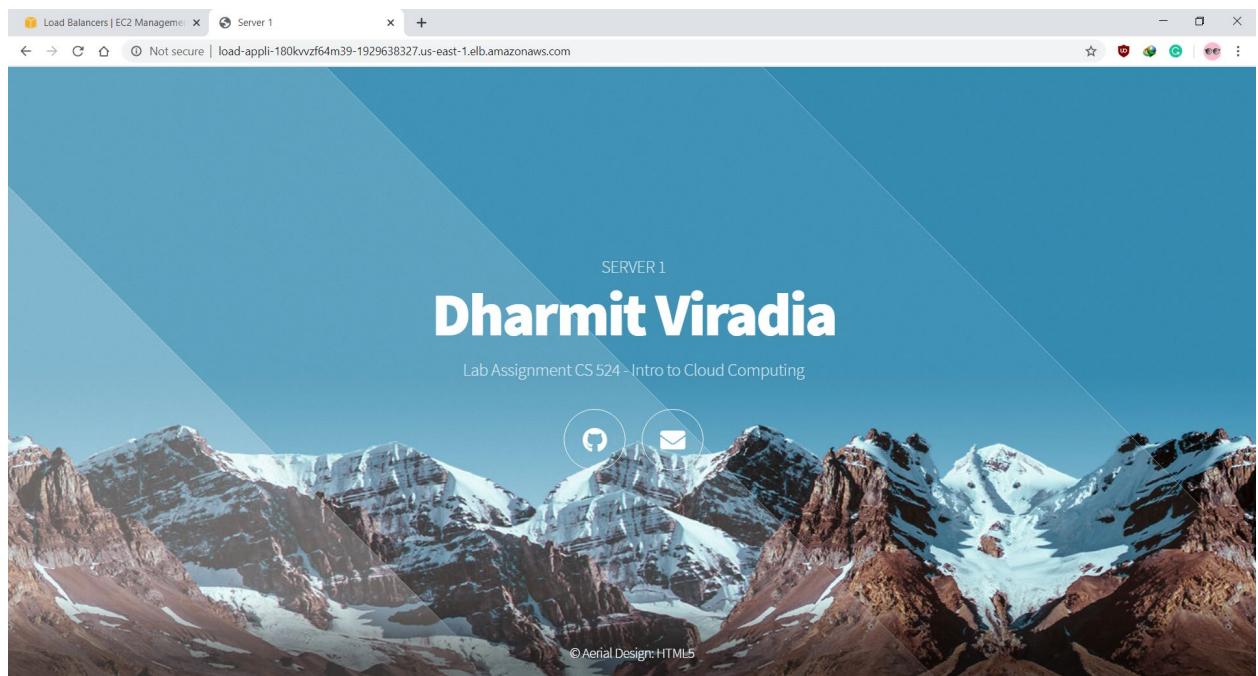
The screenshot shows the AWS EC2 Instances dashboard. On the left, there's a sidebar with various service links like Scheduled Instances, Capacity Reservations, Images, AMIs, etc. The main area has a table titled "Instances" with columns: Name, Instance ID, Instance Type, Availability Zone, Instance State, Status Checks, Alarm Status, and Public DNS (IPv4). There are 9 instances listed, including four LB Servers and five Server instances. One instance, "Load Balancer", is highlighted in blue. Below the table, a message says "Select an instance above". At the bottom, there's a URL bar with the address https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers: and some footer links.

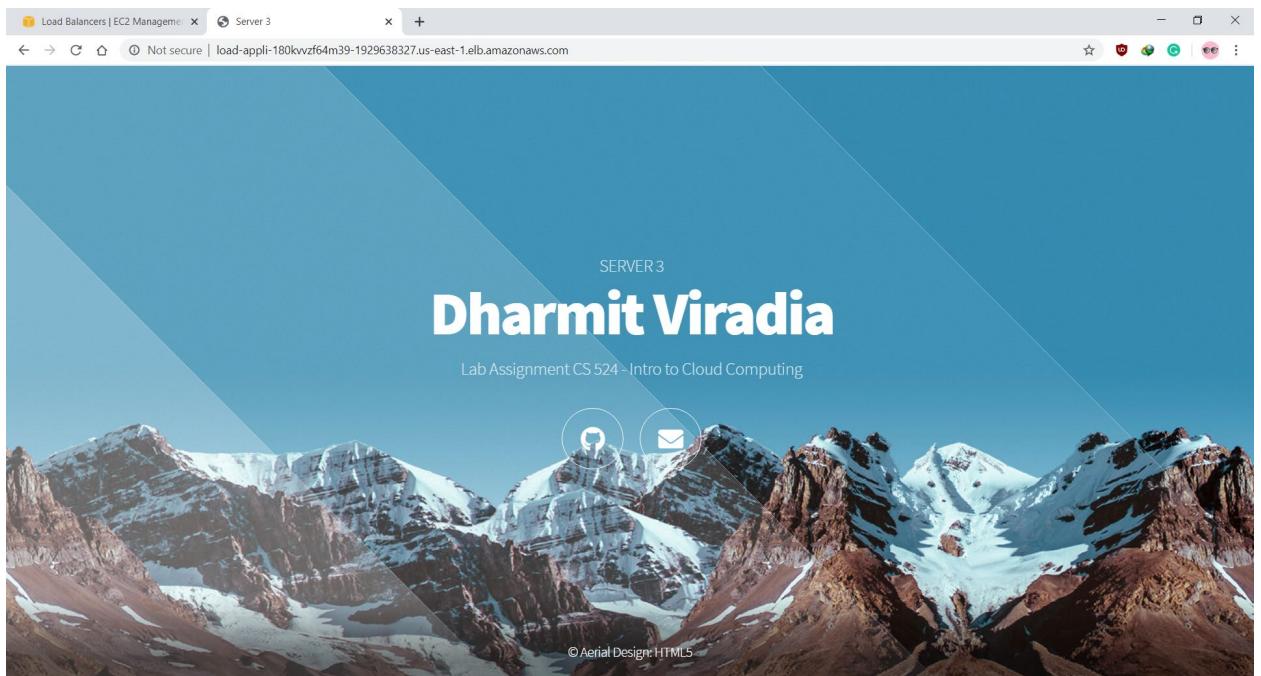
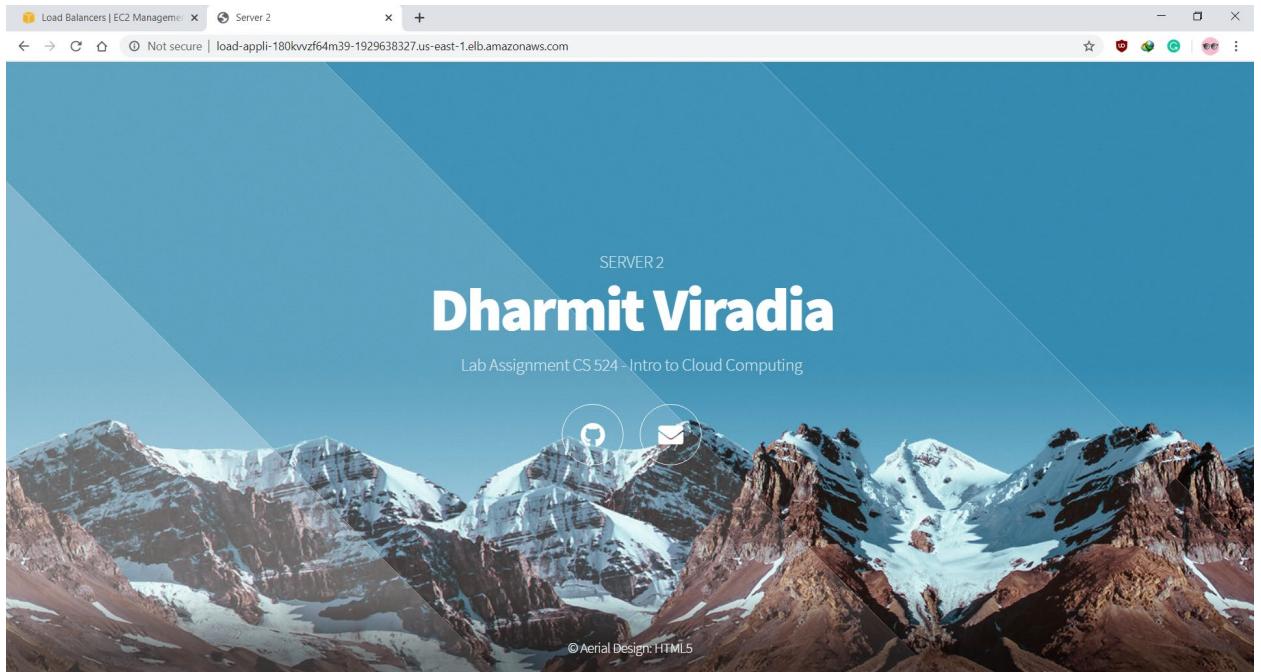
- After clicking Load Balancers, we will find a load balancer created while stack creation



The screenshot shows the AWS Load Balancers dashboard. The sidebar is identical to the previous EC2 dashboard. The main area shows a table with columns: Name, DNS name, State, VPC ID, and Availability Zones. One load balancer, "Load-Appi-180KVVF64M39", is selected and its details are shown below the table. The "Basic Configuration" section displays the Name (Load-Appi-180KVVF64M39), ARN (arn:aws:elasticloadbalancing:us-east-1:498314746917:loadbalancer/app/Load-Appi-180KVVF64M39/d5779435f15307d6), DNS name (Load-Appi-180KVVF64M39-1929638327.us-east-1.elb.amazonaws.com), and State (active). Below this, tabs for Description, Listeners, Monitoring, Integrated services, and Tags are visible. At the bottom, there's a URL bar with the address https://console.aws.amazon.com/ec2/v2/home?region=us-east-1#LoadBalancers: and some footer links.

- Now visit the load balancer DNS Name and this load balancer automatically balances the load and redirect to any of the created servers by dividing the load at equal proportion





(In my case Server 1 ran First, then Server 4, then Server 2 and in Last Server 3)