CS 524 Introduction to Cloud Computing

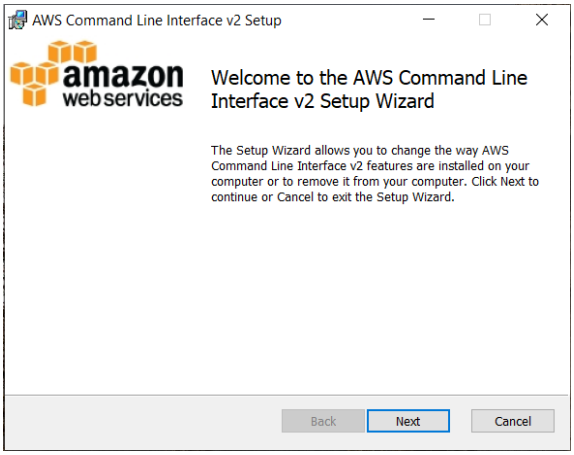
Dharmit Viradia

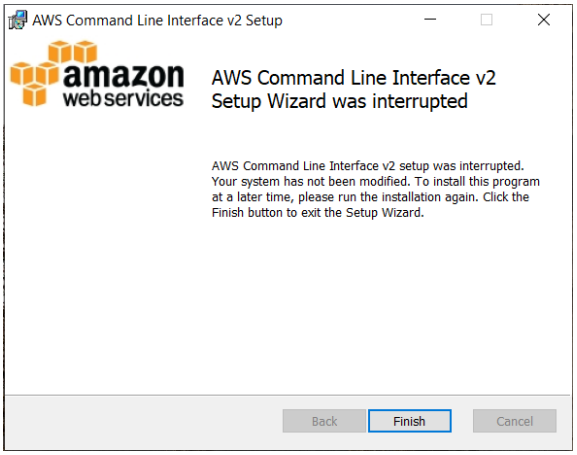
Lab Assignment 2

Prof. Igor Faynberg

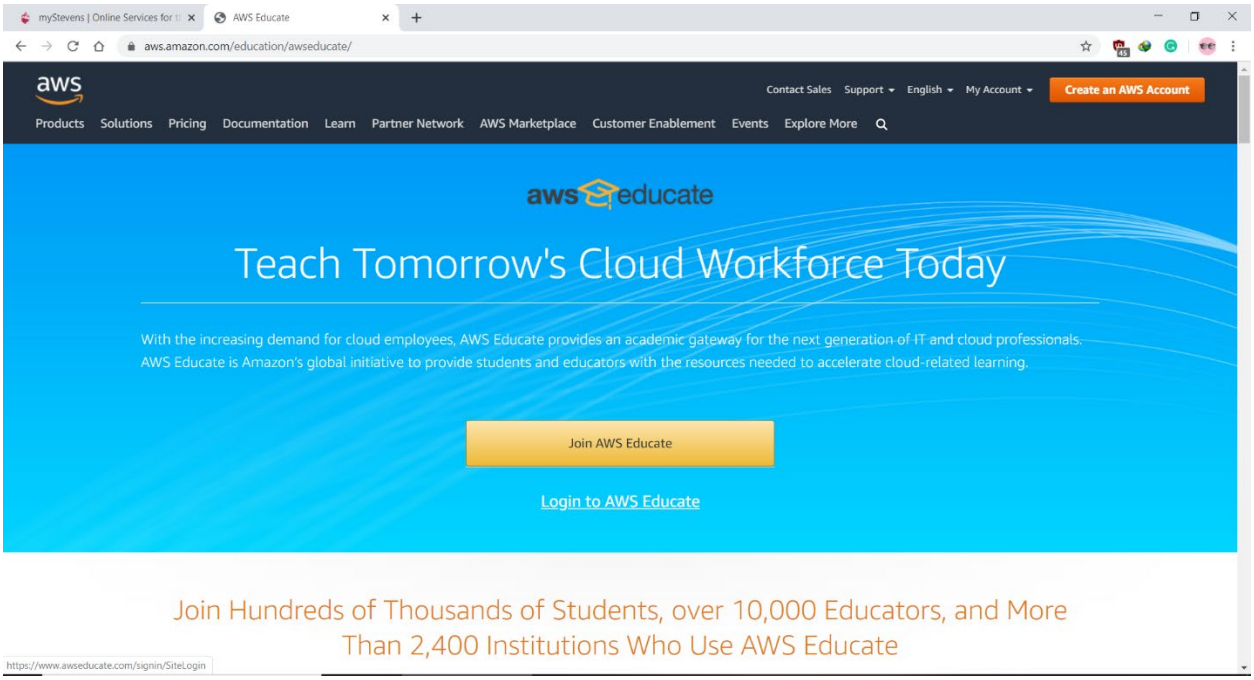
Step for Creating an Amazon EC2 Instances (Using AWS Command Line Interface)

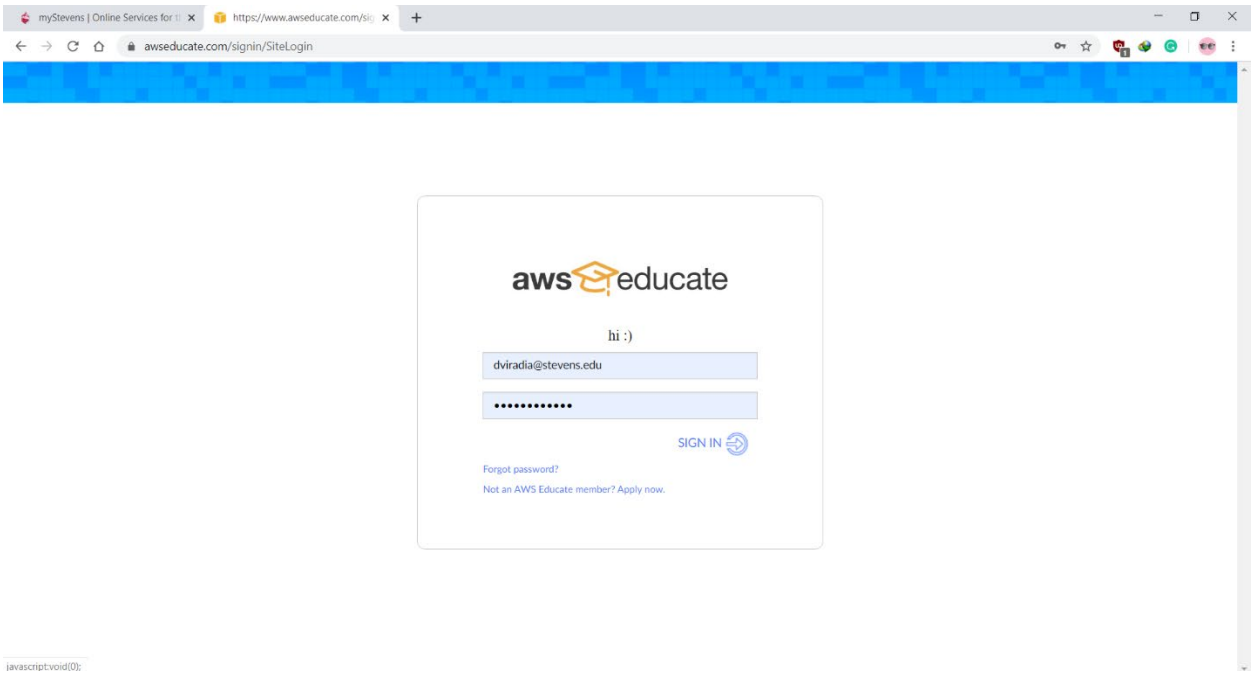
- First step in creating instances using command line interface is to download and install it.



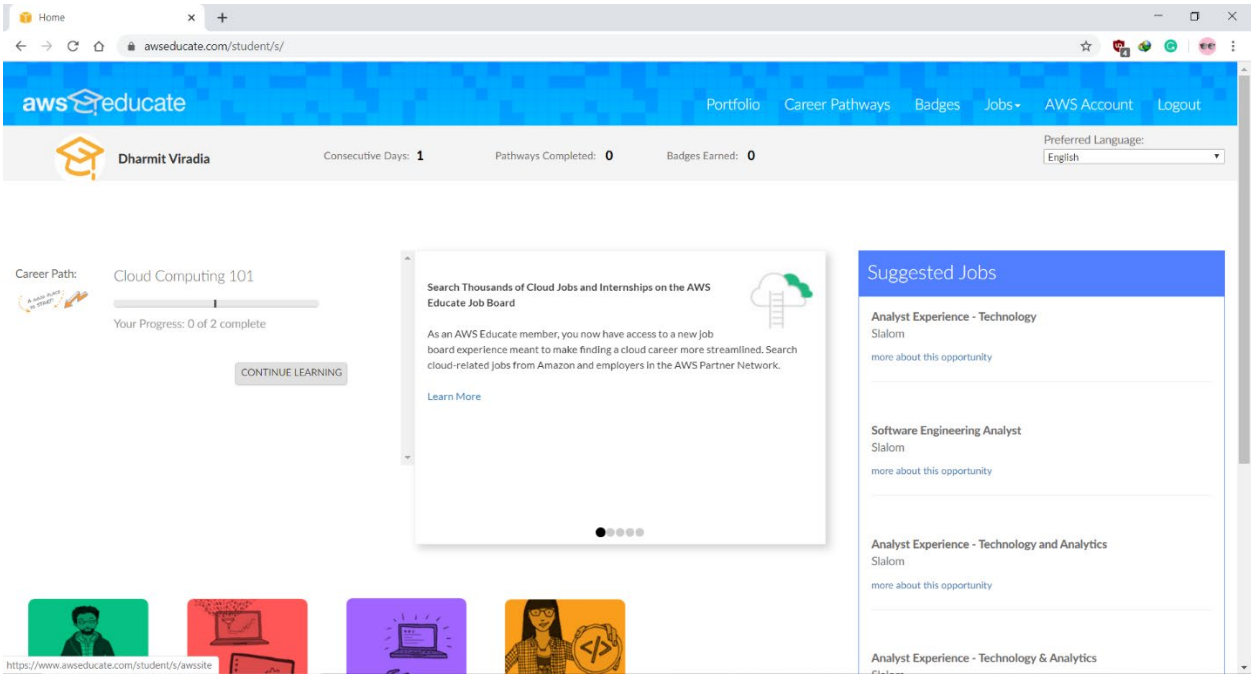


- Login to AWS Educate Account

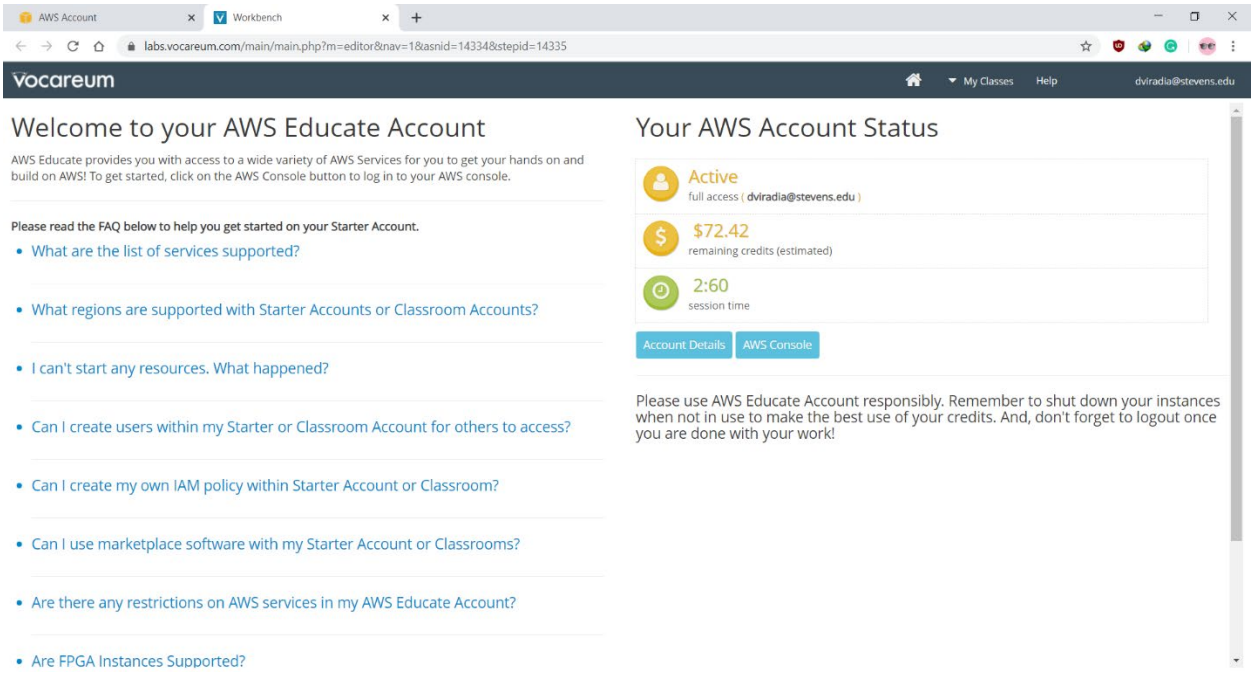




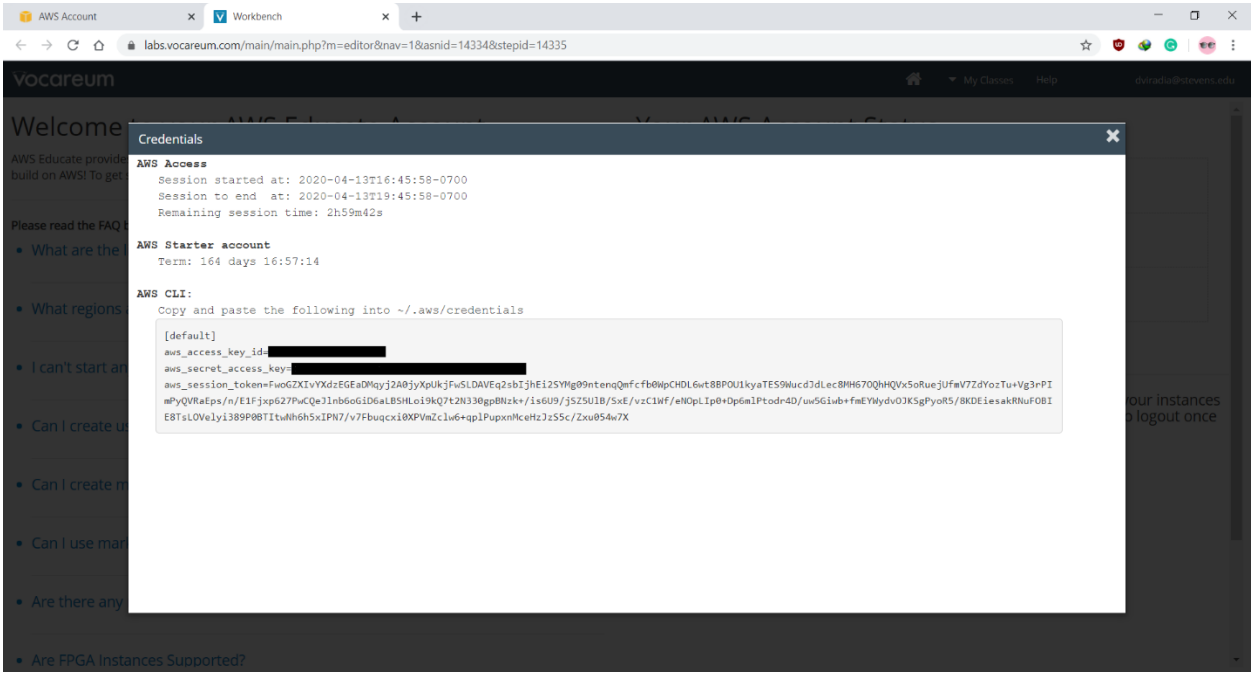
- Go to AWS Account



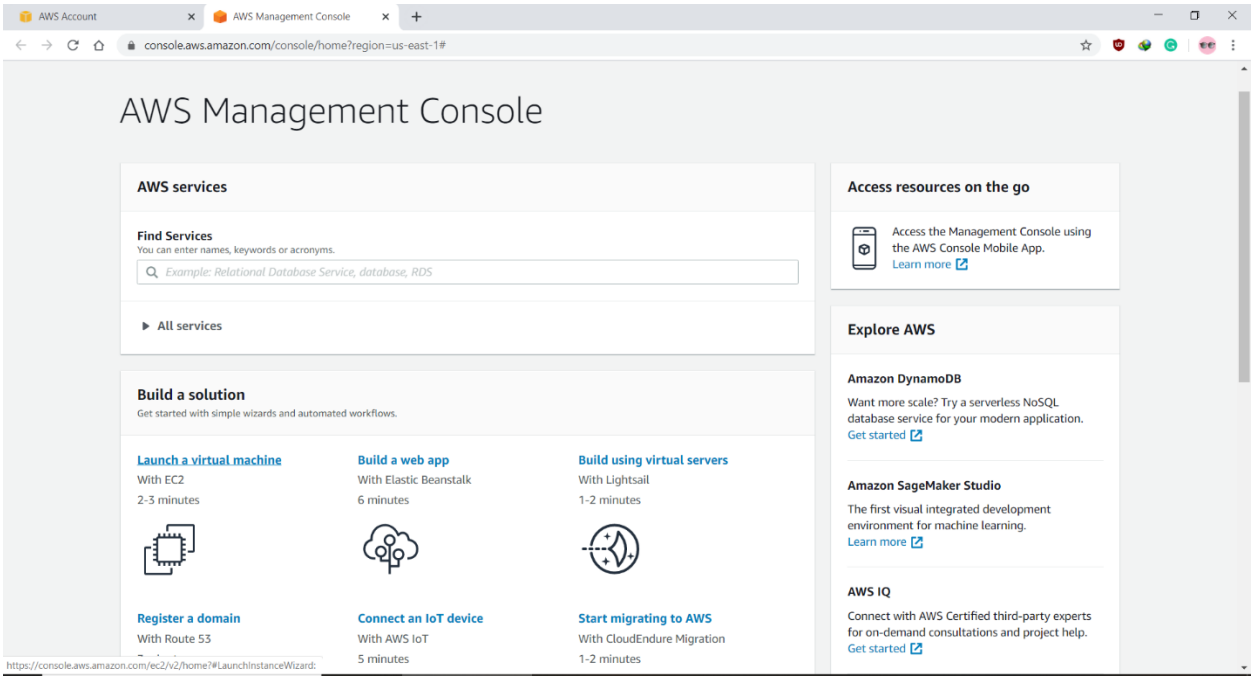
- Copy Account CLI Details to Use AWS Command Line Interface



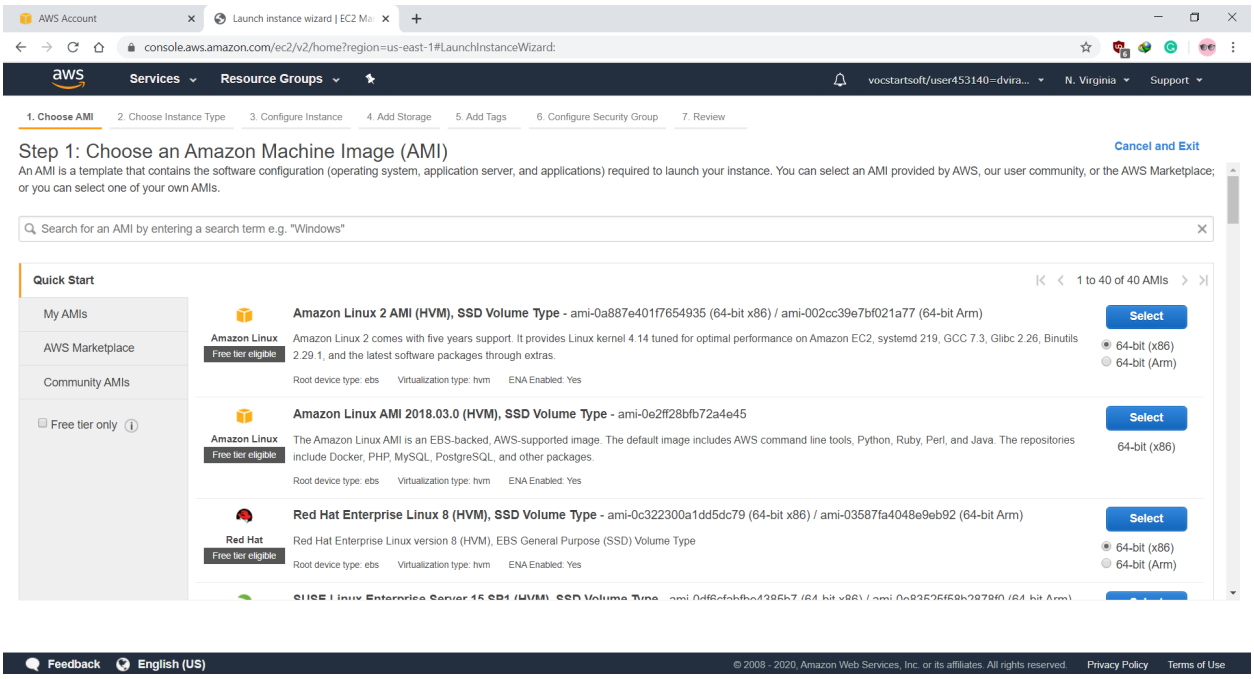
- Copy and Paste the following into ~/.aws/credentials



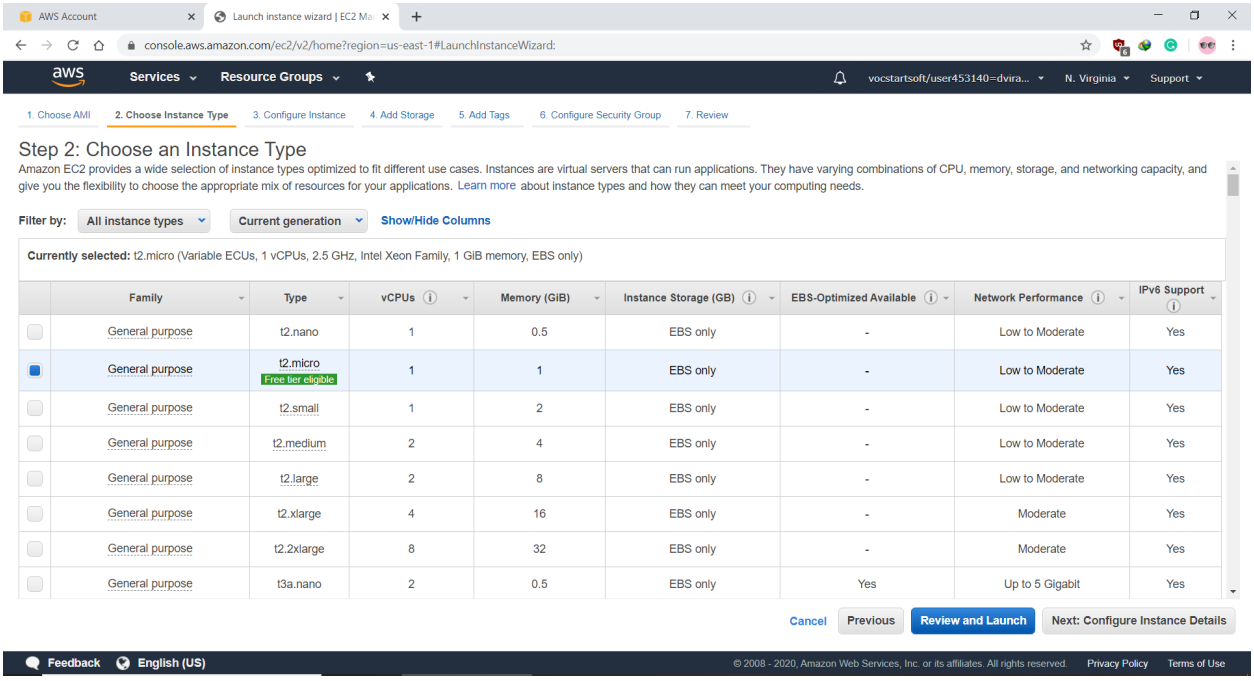
- Launch Virtual Machine



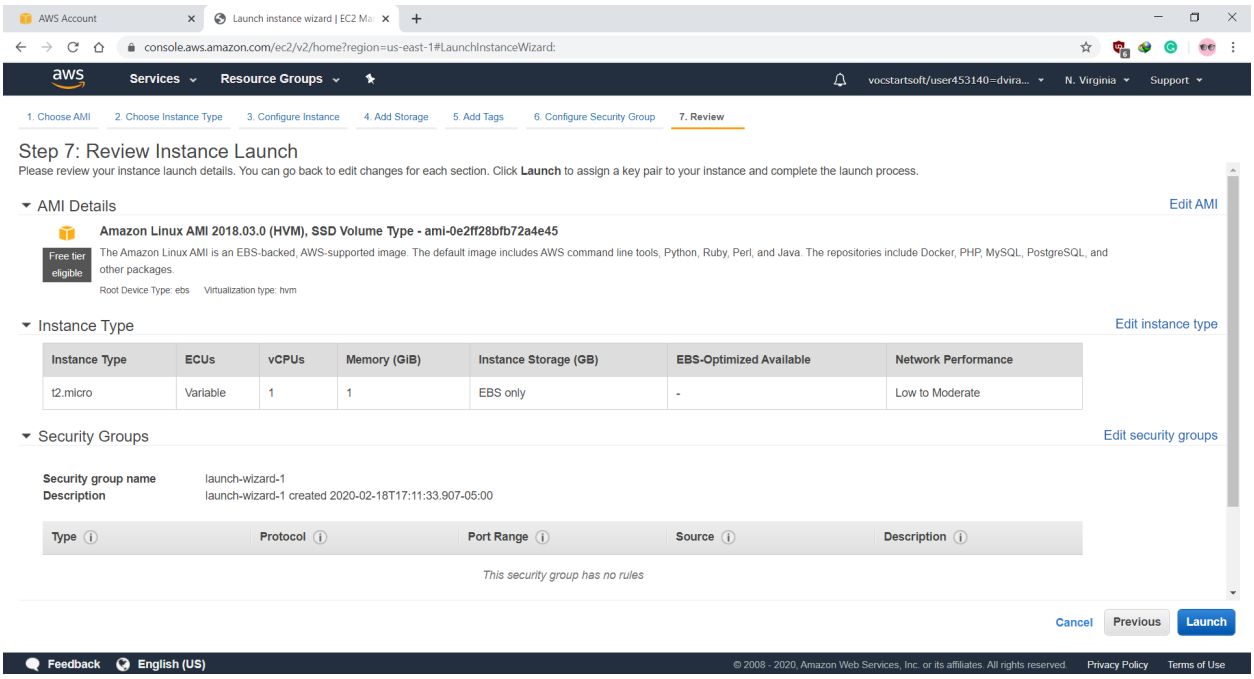
- Select Amazon Linux AMI



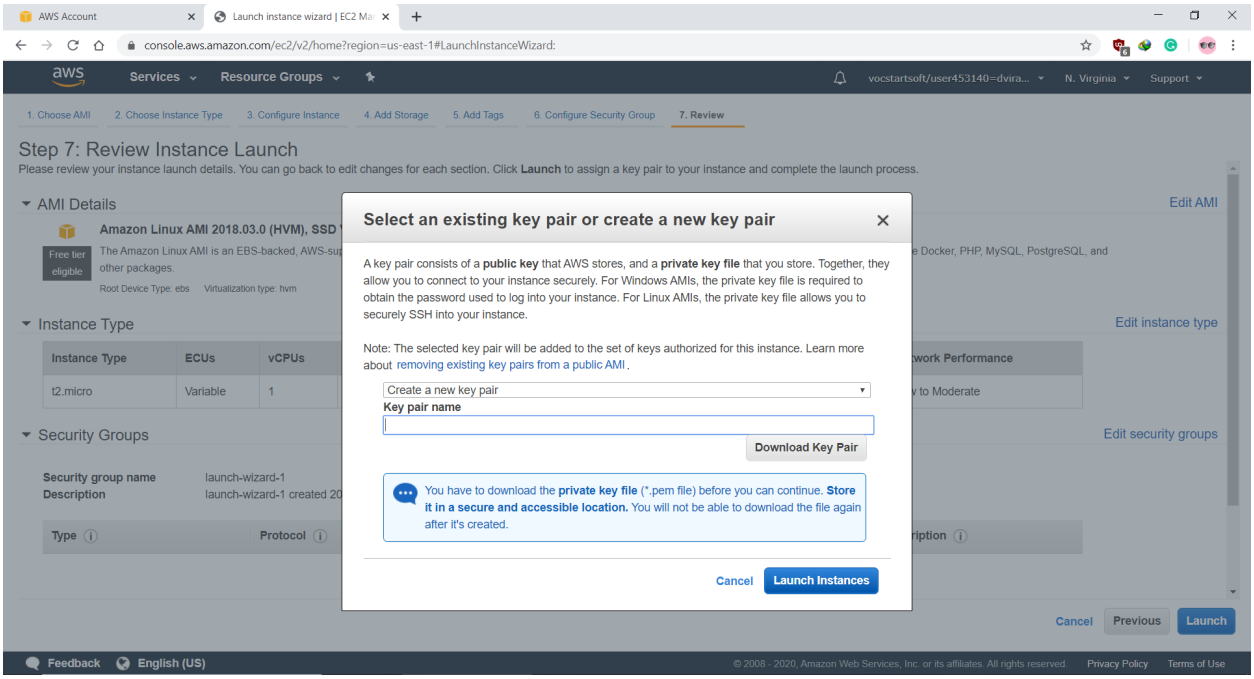
- Select Type t2.micro and review and launch



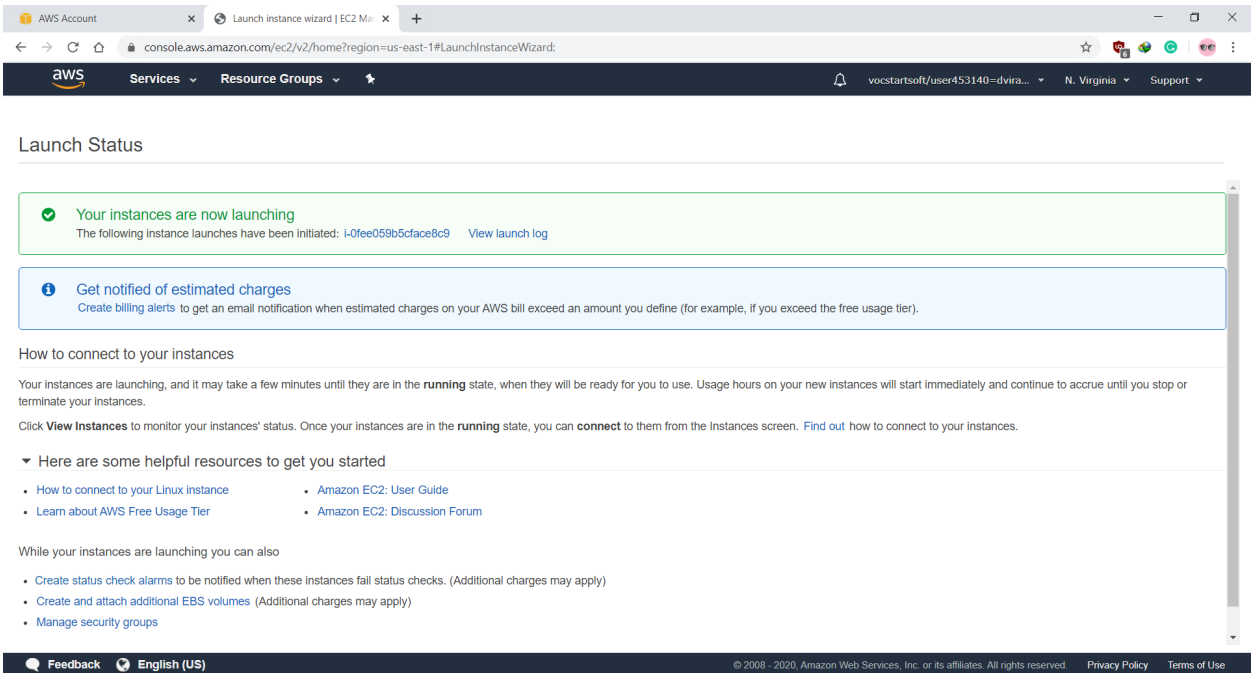
- Launch EC2 Instance



- Create Key Pair



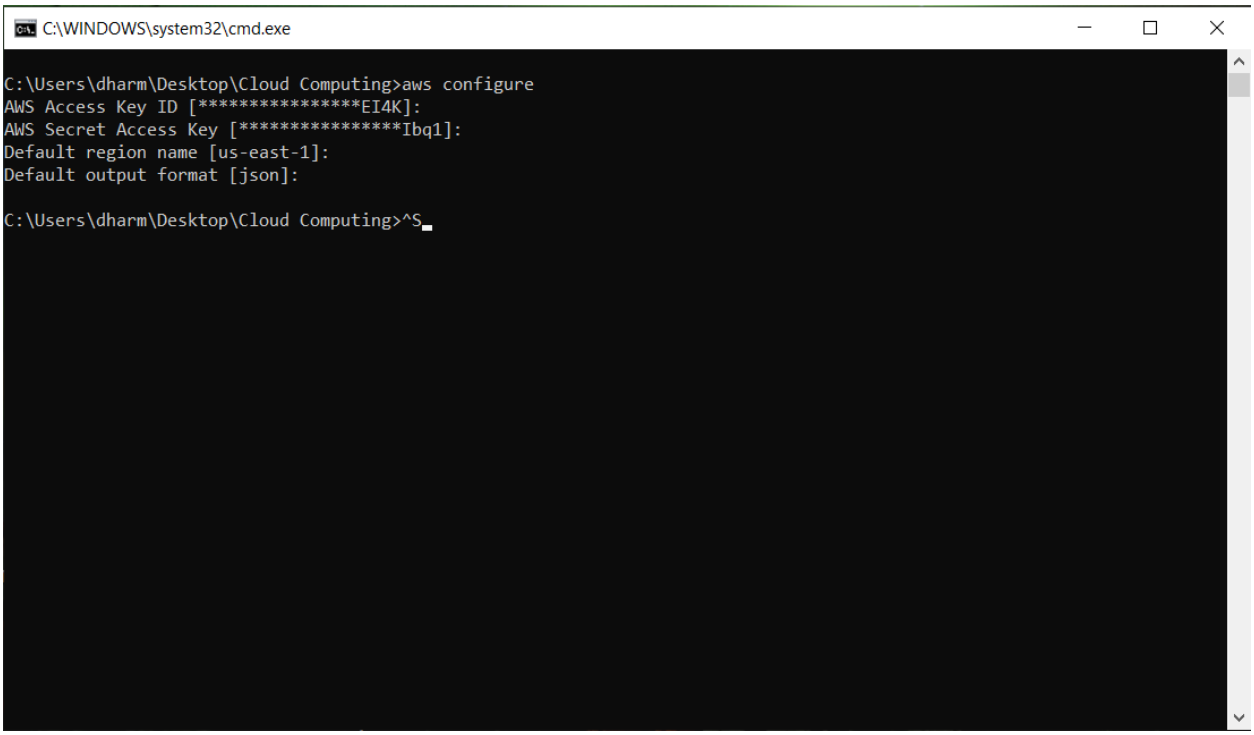
- EC2 Instance is Ready



Step for Accessing Amazon EC2 Instances (Using AWS Command Line Interface)

- Now, configuring AWS Access key id and Secret access key by executing following command and filling the prompted details.

  $ aws configure



- Now, creating a security group by executing the following command

  $ aws ec2 create-security-group --group-name default --description " default VPC security group "

- In this step, you can create a key pair to access aws resources by executing following command. If you already have one then no need to create new key pair. I already have a key-pair, so, I am using that.

  $ aws ec2 create-key-pair --key-name ec2@dharmit

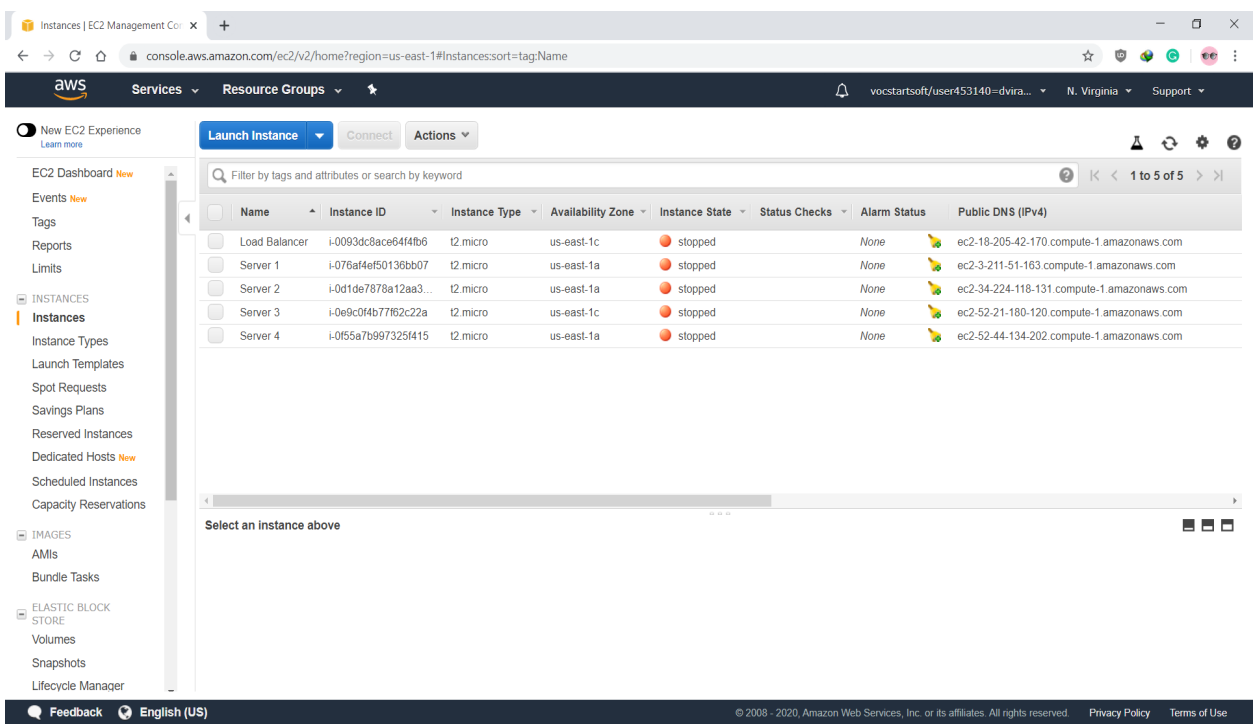
- Finally, to create instance using command line interface, we need **Amazon Linux AMI, Security Group Id, Instance Type** and **Key Pair Name**. And then execute the following command to create any number of instances.

  $ aws ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids default --count 5 --instance-type t2.micro --key-name ec2@dharmit

C:\WINDOWS\system32\cmd.exe - aws ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids default --count 5...

```
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids defau
lt --count 5 --instance-type t2.micro --key-name ec2@dharmit
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0915e09cc7ceee3ab",
            "InstanceId": "i-0f49c7431245b1b19",
            "InstanceType": "t2.micro",
            "KeyName": "ec2@dharmit",
            "LaunchTime": "2020-04-14T00:17:15+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-92-114.ec2.internal",
            "PrivateIpAddress": "172.31.92.114",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
```

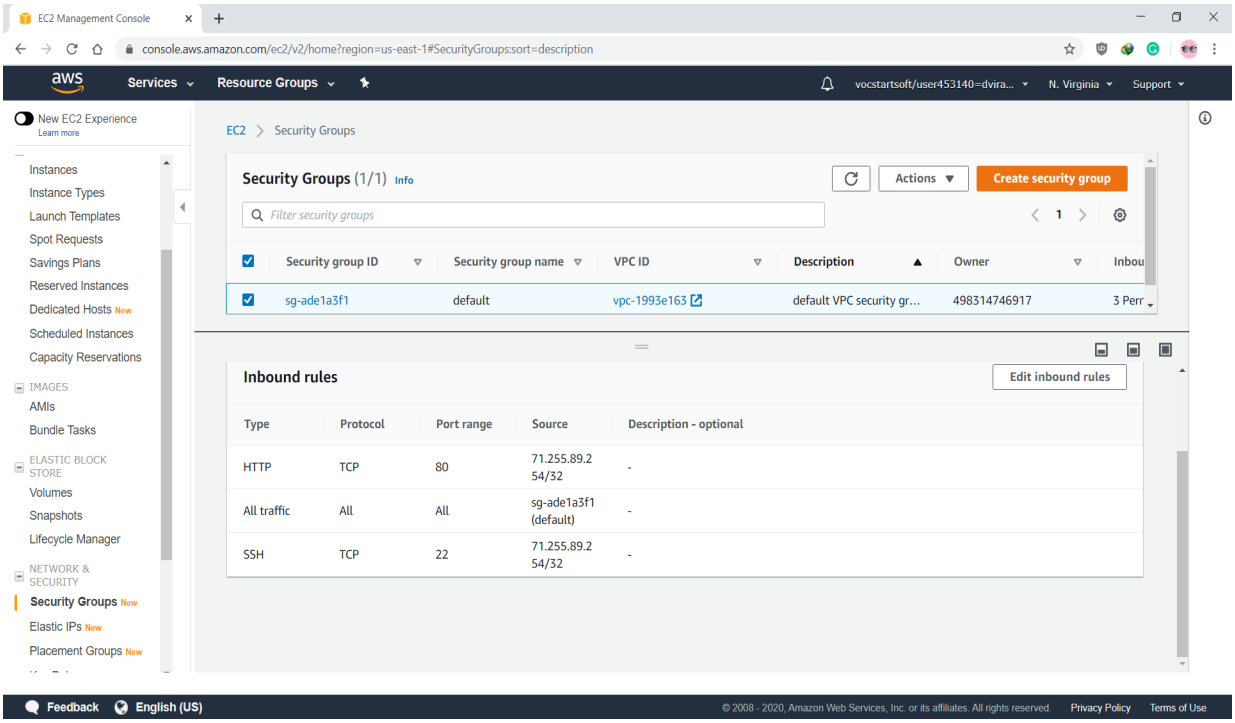
- Now, our 5 instances have been created. I have created five instances in total and named them as Load Balancer, Server 1, Server 2, Server 3, and Server 4.



- The last step is to configure the security group for inbound access.

  $ aws ec2 authorize-security-group-ingress –-group-name default --protocol tcp –-port 22 –-cidr 71.255.89.254/32

  $ aws ec2 authorize-security-group-ingress –-group-name default --protocol tcp –-port 80 –-cidr 71.255.89.254/32



Step for Accessing AWS instance

- Now get instances ID by executing following command

  $ aws ec2 describe-instances

C:\WINDOWS\system32\cmd.exe - aws ec2 describe-instances

```
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 describe-instances
{
    "Reservations": [
        {
            "Groups": [],
            "Instances": [
                {
                    "AmiLaunchIndex": 0,
                    "ImageId": "ami-09a5b0b7edf08843d",
                    "InstanceId": "i-0f55a7b997325f415",
                    "InstanceType": "t2.micro",
                    "KeyName": "ec2@dharmit",
                    "LaunchTime": "2020-04-06T04:10:46+00:00",
                    "Monitoring": {
                        "State": "disabled"
                    },
                    "Placement": {
                        "AvailabilityZone": "us-east-1a",
                        "GroupName": "",
                        "Tenancy": "default"
                    },
                    "PrivateDnsName": "ip-172-31-91-250.ec2.internal",
                    "PrivateIpAddress": "172.31.91.250",
                    "ProductCodes": [],
                    "PublicDnsName": "",
                    "State": {
                        "Code": 80,
                        "Name": "stopped"
                    },
                    "StateTransitionReason": "User initiated (2020-04-06 04:28:29 GMT)",
```

- Start the instances by executing the following command for all the instances by replacing instances id

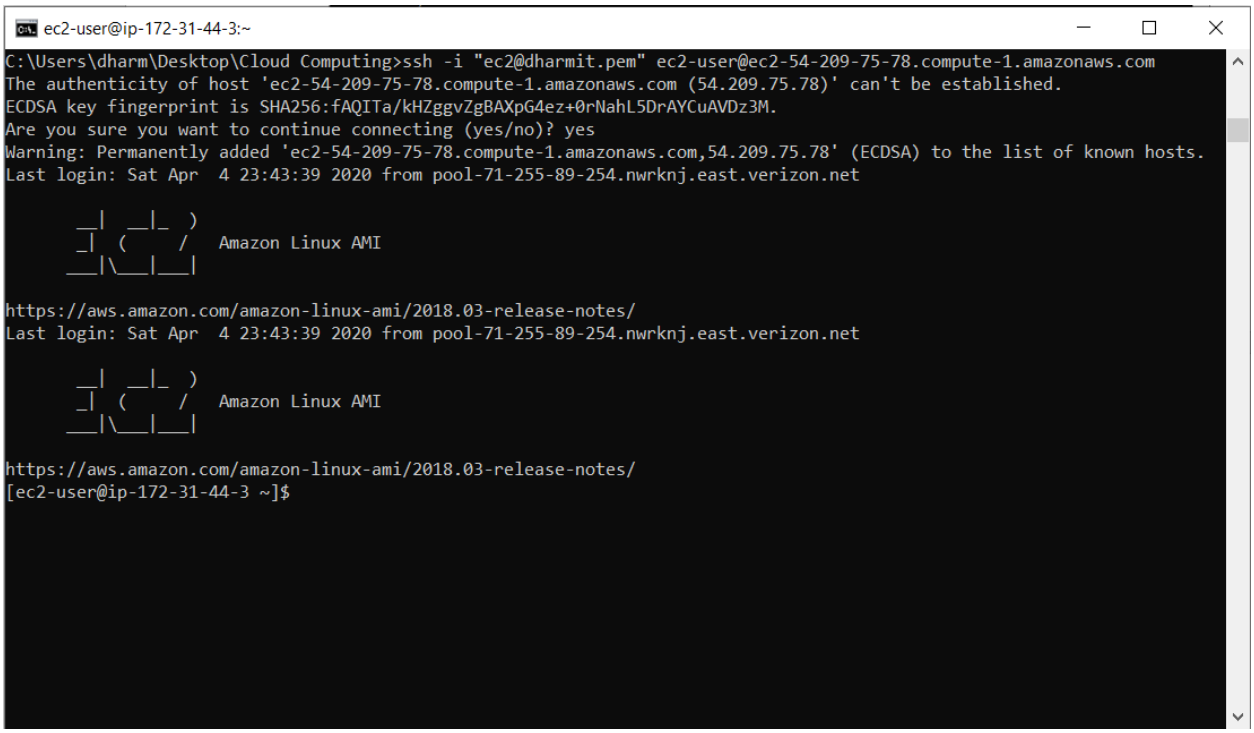
  $ aws ec2 start-instances --instance-ids i-0093dc8ace64f4fb6

C:\WINDOWS\system32\cmd.exe

```
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 start-instances --instance-ids i-0093dc8ace64f4fb6
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-0093dc8ace64f4fb6",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}


C:\Users\dharm\Desktop\Cloud Computing>
```

- Establishing connection with EC2 Instance by executing below command

  $ ssh -i "ec2@dharmit.pem" ec2-user@ec2-54-209-75-78.compute-1.amazonaws.com



Steps to install Nginx Server on Amazon EC2 instance

- After establish a connection with the EC2 instance, installing nginx server on it by executing the below command.

  $ sudo yum install nginx

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ sudo yum install nginx
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.16.1-1.37.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package              Arch                 Version                            Repository                  Size
================================================================================================================
Installing:
 nginx                x86_64               1:1.16.1-1.37.amzn1                amzn-updates               598 k

Transaction Summary
================================================================================================================
Install  1 Package

Total download size: 598 k
Installed size: 1.5 M
Is this ok [y/d/N]: y
Downloading packages:
nginx-1.16.1-1.37.amzn1.x86_64.rpm                                                        | 598 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : 1:nginx-1.16.1-1.37.amzn1.x86_64                                                            1/1
  Verifying  : 1:nginx-1.16.1-1.37.amzn1.x86_64                                                            1/1

Installed:
  nginx.x86_64 1:1.16.1-1.37.amzn1

Complete!
[ec2-user@ip-172-31-44-3 ~]$
```

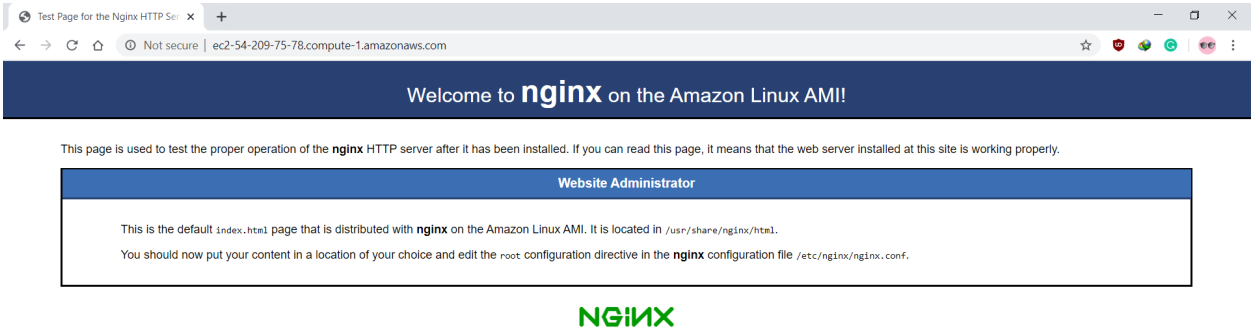
- After installing nginx, starting the services of the nginx by executing

  $ sudo service nginx start

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ sudo service nginx start
Starting nginx:                                            [  OK  ]
[ec2-user@ip-172-31-44-3 ~]$
```

- Testing server by running Public DNS (IPv4) on the browser.

  http://ec2-54-209-75-78.compute-1.amazonaws.com/

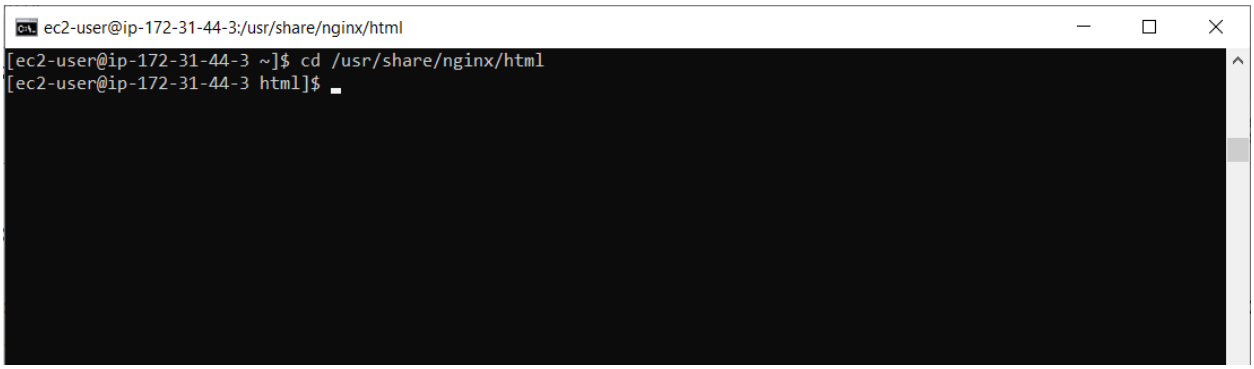


- Now, repeat the same steps to install Nginx Server on each instance you want to install. I have installed the nginx server on each instance i.e. Server 1, Server 2, Server 3, Server 4, and Load Balancer.

Steps to change nginx server index.html file on Amazon EC2 instance

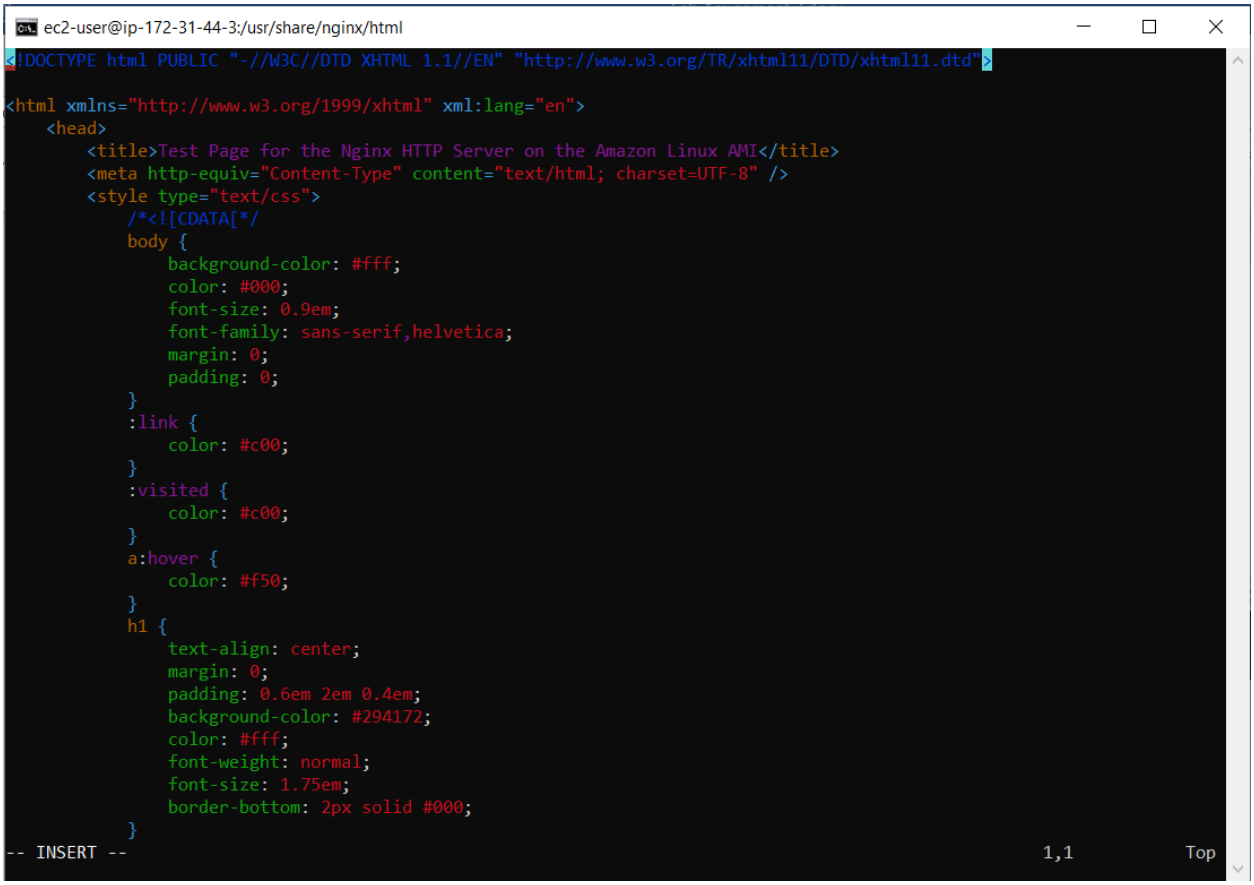
- After successfully installing nginx server, navigate to /usr/share/nginx/html directory. To navigate type following command in the terminal window

  $ cd /usr/share/nginx/html



- Then open the index.html file in vim. To open type following command in the terminal window

  $ sudo vim index.html

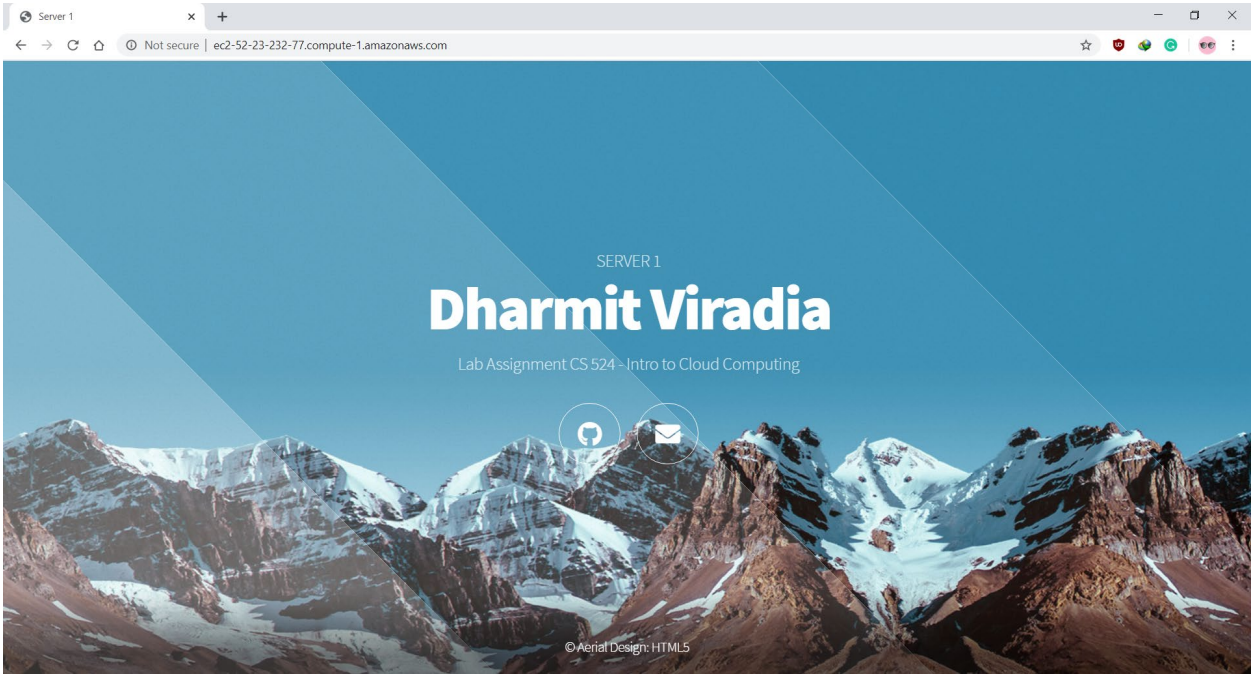


- Editing the index.html file for Server 1 or instance number 1.

ec2-user@ip-172-31-83-107:/usr/share/nginx/html

```
<!DOCTYPE HTML>

<html>
        <head>
                <title>Server 1</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <link rel="stylesheet" href="assets/css/main.css" />
                <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
        </head>
        <body class="is-preload">
                <div id="wrapper">
                        <div id="bg"></div>
                        <div id="overlay"></div>
                        <div id="main">

                                <!-- Header -->
                                        <header id="header">
                                        <p>SERVER 1</p>
                                                <h1>Dharmit Viradia</h1>
                                                <p>Lab Assignment CS 524 - Intro to Cloud Computing</p>
                                                <nav>
                                                        <ul>
                                                                <li><a href="https://github.com/dharmitviradia" class="i
con brands fa-github"><span class="label">Github</span></a></li>
                                                                <li><a href="mailto:dviradia@stevens.edu" class="icon so
lid fa-envelope"><span class="label">Email</span></a></li>
                                                        </ul>
                                                </nav>
                                        </header>
```

- 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 1
      http://ec2-3-211-51-163.compute-1.amazonaws.com/
      http://3.211.51.163/

    - Server 2
      http://ec2-34-224-118-131.compute-1.amazonaws.com/
      http://34.224.118.131/

    - Server 3
      http://ec2-52-21-180-120.compute-1.amazonaws.com/
      http://52.21.180.120/

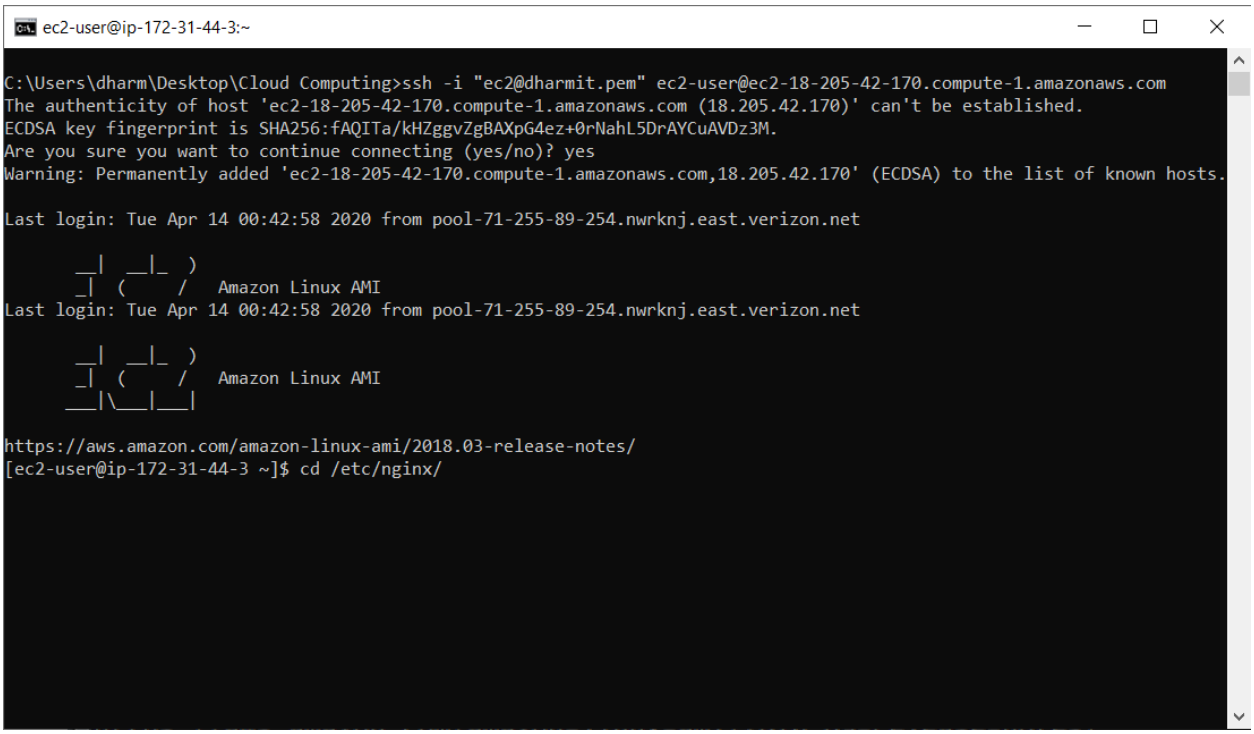
    - Server 4
      http://ec2-52-44-134-202.compute-1.amazonaws.com/
      http://52.44.134.202/

(Note, we can also edit index.html file of the Load Balancer instance. But there would be no use of modifying it, because whenever you try to hit/visit a public IP or DNS of Load Balancer, it would redirect you on one of the servers it is dealing with in my case it would redirect to one of the above servers. In next step, we will be handling this amazing feature of Load Balancer, and observer how it works to balance the load by redirecting to one of its servers.)

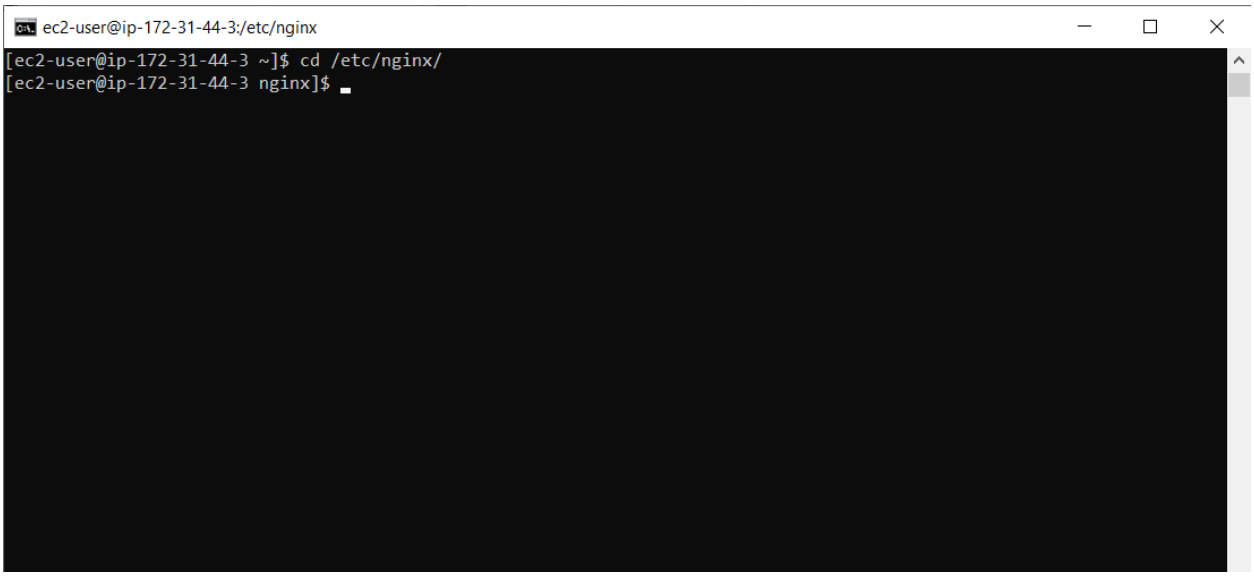
Steps to configure Load Balancer on Amazon EC2 instance

- First to configure the Load Balance, connect to Load Balance instance. Then navigate to /etc/nginx/ by executing the following commands in the terminal.

  $ ssh -i "ec2@dharmit.pem" ec2-user@ec2-18-205-42-170.compute-1.amazonaws.com



  $ cd /etc/nginx/



- Now open nginx.conf using vim by executing following command

  $ sudo vim nginx.conf

Select ec2-user@ip-172-31-44-3:/etc/nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events { worker_connections 768;}

http {
        upstream myapp {
        #ip_hash;
        server ec2-3-211-51-163.compute-1.amazonaws.com weight=1;
        server ec2-34-224-118-131.compute-1.amazonaws.com weight=1;
        server ec2-52-21-180-120.compute-1.amazonaws.com weight=1;
        server ec2-52-44-134-202.compute-1.amazonaws.com weight=1;
         }

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    keepalive_timeout   65;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    # Load modular configuration files from the /etc/nginx/conf.d directory.
    # See http://nginx.org/en/docs/ngx_core_module.html#include
    # for more information.
    include /etc/nginx/conf.d/*.conf;

    index   index.html index.htm;

    server {
        listen       80 default_server;
        listen       [::]:80 default_server;
-- INSERT --                                                          1,32          Top
```

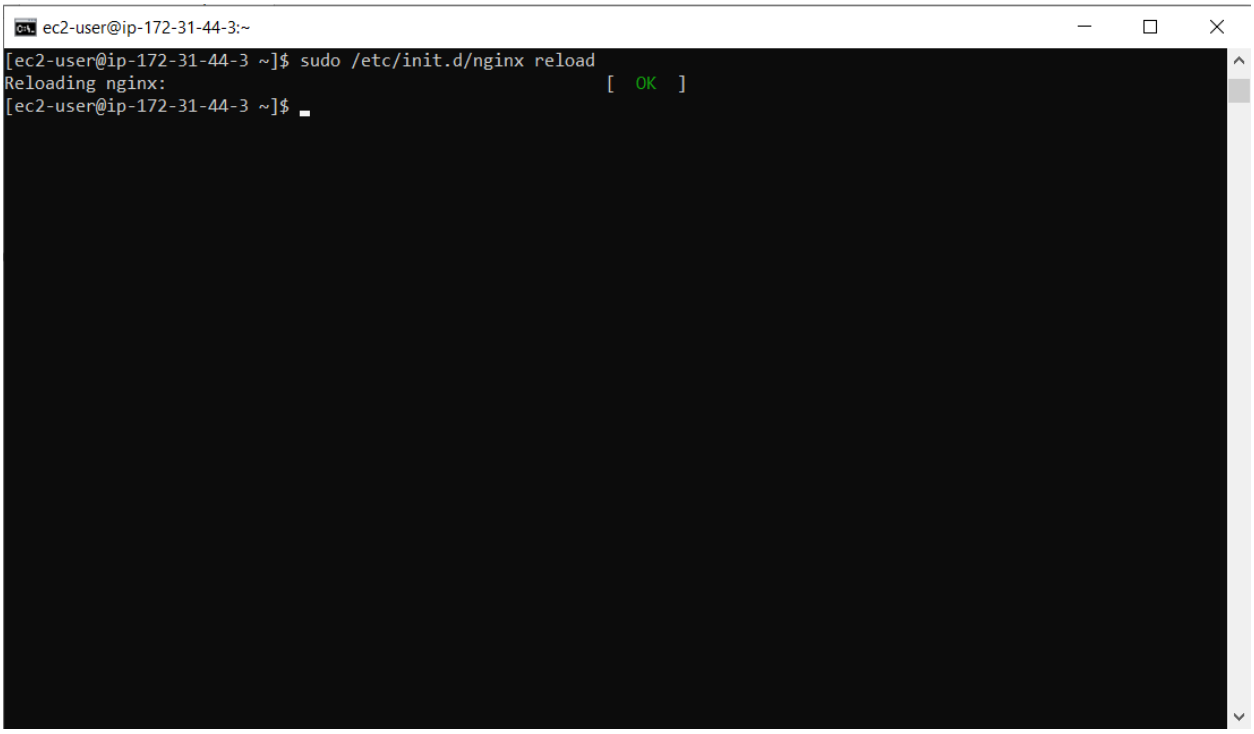
Now editing file by adding and replacing code by following codes events
{
worker_connections 768;
}
http {
upstream myapp {
#ip_hash;
server [SERVER_PUBLIC_DNS_NAME] weight=1;
server [SERVER_PUBLIC_DNS_NAME] weight=1;

```
                server [SERVER_PUBLIC_DNS_NAME] weight=1;
                server [SERVER_PUBLIC_DNS_NAME] weight=1; }

server {
        listen 80;
        server_name myapp.com;
        location / {
                proxy_pass http://myapp;
                }
        }
}
```

(**Note**: SERVER_PUBLIC_DNS_NAME would be replaced by your instance Public DNS)

- Now, run the following command in the shell (this will cause the new configuration to take effect):

  $ /etc/init.d/nginx reload



- Now, we have to use the curl command in the shell to visit the load balancer, which will distribute traffic among their servers.

  $ curl ec2-18-205-42-170.compute-1.amazonaws.com

Server 1

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ curl ec2-18-205-42-170.compute-1.amazonaws.com
<!DOCTYPE HTML>

<html>
        <head>
                <title>Server 1</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <link rel="stylesheet" href="assets/css/main.css" />
                <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
        </head>
        <body class="is-preload">
                <div id="wrapper">
                        <div id="bg"></div>
                        <div id="overlay"></div>
                        <div id="main">

                                <!-- Header -->
                                        <header id="header">
                                        <p>SERVER 1</p>
                                                <h1>Dharmit Viradia</h1>
                                                <p>Lab Assignment CS 524 - Intro to Cloud Computing</p>
                                                <nav>
                                                        <ul>
                                                                <li><a href="https://github.com/dharmitviradia" class="i
con brands fa-github"><span class="label">Github</span></a></li>
                                                                <li><a href="mailto:dviradia@stevens.edu" class="icon so
lid fa-envelope"><span class="label">Email</span></a></li>
                                                        </ul>
                                                </nav>
```

Server 2

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ curl ec2-18-205-42-170.compute-1.amazonaws.com
<!DOCTYPE HTML>

<html>
        <head>
                <title>Server 2</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <link rel="stylesheet" href="assets/css/main.css" />
                <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
        </head>
        <body class="is-preload">
                <div id="wrapper">
                        <div id="bg"></div>
                        <div id="overlay"></div>
                        <div id="main">

                                <!-- Header -->
                                        <header id="header">
                                        <p>SERVER 2</p>
                                                <h1>Dharmit Viradia</h1>
                                                <p>Lab Assignment CS 524 - Intro to Cloud Computing</p>
                                                <nav>
                                                        <ul>
                                                                <li><a href="https://github.com/dharmitviradia" class="i
con brands fa-github"><span class="label">Github</span></a></li>
                                                                <li><a href="mailto:dviradia@stevens.edu" class="icon so
lid fa-envelope"><span class="label">Email</span></a></li>
                                                        </ul>
                                                </nav>
```

Server 3

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ curl ec2-18-205-42-170.compute-1.amazonaws.com
<!DOCTYPE HTML>

<html>
        <head>
                <title>Server 3</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <link rel="stylesheet" href="assets/css/main.css" />
                <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
        </head>
        <body class="is-preload">
                <div id="wrapper">
                        <div id="bg"></div>
                        <div id="overlay"></div>
                        <div id="main">

                                <!-- Header -->
                                        <header id="header">
                                        <p>SERVER 3</p>
                                                <h1>Dharmit Viradia</h1>
                                                <p>Lab Assignment CS 524 - Intro to Cloud Computing</p>
                                                <nav>
                                                        <ul>
                                                                <li><a href="https://github.com/dharmitviradia" class="i
con brands fa-github"><span class="label">Github</span></a></li>
                                                                <li><a href="mailto:dviradia@stevens.edu" class="icon so
lid fa-envelope"><span class="label">Email</span></a></li>
                                                        </ul>
                                                </nav>
```

Server 4

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ curl ec2-18-205-42-170.compute-1.amazonaws.com
<!DOCTYPE HTML>

<html>
        <head>
                <title>Server 4</title>
                <meta charset="utf-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1, user-scalable=no" />
                <link rel="stylesheet" href="assets/css/main.css" />
                <noscript><link rel="stylesheet" href="assets/css/noscript.css" /></noscript>
        </head>
        <body class="is-preload">
                <div id="wrapper">
                        <div id="bg">
                        <div id="overlay">
                        <div id="main">

                                <!-- Header -->
                                        <header id="header">
                                        <p>SERVER 4</p>
                                                <h1>Dharmit Viradia</h1>
                                                <p>Lab Assignment CS 524 - Intro to Cloud Computing</p>
                                                <nav>
                                                        <ul>
                                                                <li><a href="https://github.com/dharmitviradia" class="i
con brands fa-github"><span class="label">Github</span></a></li>
                                                                <li><a href="mailto:dviradia@stevens.edu" class="icon so
lid fa-envelope"><span class="label">Email</span></a></li>
                                                        </ul>
                                                </nav>
```

(**Notice** that on each curl command, the load balancer is distributing traffic to each server sequentially)

Steps to collect information on visits to your website using Amazon EC2 instance

- Setting up Visit Server tool to track the distribution of the load. This tool visits the cluster 2000 times and returns the visit count on each server

  $ cd /usr/bin

  $ vim visit_server

```
ec2-user@ip-172-31-44-3:~
#!/usr/bin/env ruby
#
# This program is used for collecting web server visit information.
#
# Author: A. Genius
#

require 'optparse'

def print_usage
    puts "USAGE: visit_server -d DNS_NAME"
    exit
end

# add option switch and handler
options = {}

option_parser = OptionParser.new do |opts|

    # DNS_NAME argument
    options[:dns_name] = nil
    opts.on('-d', '--dns-name DNS_NAME', 'Specify a DNS NAME') { |dns_name| options[:dns_name] = dns_name }

    # HELP argument
    options[:help] = nil
    opts.on('-h', '--help', 'Display usage') { |help| options[:help] = help }

end

-- INSERT --                                                                    1,1           Top
```

- Now, we will trace the load balancing server load distribution for 3 scenarios by changing the server weight in the nginx.conf file

  - Scenario #1 – Server1 weight=1, Server2 weight=1, Server3 weight=1, Server4 weight=1

  - Scenario #2 – Server1 weight=1, Server2 weight=2, Server3 weight=3, Server4 weight=1

  - Scenario #3 – Server1 weight=1, Server2 weight=2, Server3 weight=1, Server4 weight=2

- Tracing load balancer for
  Scenario #1 – Server1 weight=1, Server2 weight=1, Server3 weight=1, Server4 weight=1

ec2-user@ip-172-31-44-3:/etc/nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events { worker_connections 768;}

http {
        upstream myapp {
        #ip_hash;
        server ec2-3-211-51-163.compute-1.amazonaws.com weight=1;
        server ec2-34-224-118-131.compute-1.amazonaws.com weight=1;
        server ec2-52-21-180-120.compute-1.amazonaws.com weight=1;
        server ec2-52-44-134-202.compute-1.amazonaws.com weight=1;
         }

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

-- INSERT --                                                          1,46          Top
```

ec2-user@ip-172-31-44-3:~

```
[ec2-user@ip-172-31-44-3 ~]$ visit_server -d ec2-18-205-42-170.compute-1.amazonaws.com
Starting to visit load balancing server
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................................
.....................................................................
------------------------
 Summary
------------------------
Server1 visit counts : 500
Server2 visit counts : 500
Server3 visit counts : 500
Server4 visit counts : 500
Total visit counts : 2000
[ec2-user@ip-172-31-44-3 ~]$
```

- Tracing load balancer for
  Scenario #2 – Server1 weight=1, Server2 weight=2, Server3 weight=3, Server4 weight=4

ec2-user@ip-172-31-44-3:/etc/nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events { worker_connections 768;}

http {
        upstream myapp {
        #ip_hash;
        server ec2-3-211-51-163.compute-1.amazonaws.com weight=1;
        server ec2-34-224-118-131.compute-1.amazonaws.com weight=2;
        server ec2-52-21-180-120.compute-1.amazonaws.com weight=3;
        server ec2-52-44-134-202.compute-1.amazonaws.com weight=4;
         }

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

-- INSERT --                                                          1,1           Top
```

ec2-user@ip-172-31-44-3:/etc/nginx

```
[ec2-user@ip-172-31-44-3 nginx]$ visit_server -d ec2-18-205-42-170.compute-1.amazonaws.com
Starting to visit load balancing server
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................
------------------------
 Summary
------------------------
Server1 visit counts : 200
Server2 visit counts : 400
Server3 visit counts : 600
Server4 visit counts : 800
Total visit counts : 2000
[ec2-user@ip-172-31-44-3 nginx]$
```

- Tracing load balancer for
  Scenario #3 – Server1 weight=1, Server2 weight=2, Server3 weight=1, Server4 weight=2

ec2-user@ip-172-31-44-3:/etc/nginx

```
# For more information on configuration, see:
#   * Official English Documentation: http://nginx.org/en/docs/
#   * Official Russian Documentation: http://nginx.org/ru/docs/

user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log;
pid /var/run/nginx.pid;

# Load dynamic modules. See /usr/share/doc/nginx/README.dynamic.
include /usr/share/nginx/modules/*.conf;

events { worker_connections 768;}

http {
        upstream myapp {
        #ip_hash;
        server ec2-3-211-51-163.compute-1.amazonaws.com weight=1;
        server ec2-34-224-118-131.compute-1.amazonaws.com weight=2;
        server ec2-52-21-180-120.compute-1.amazonaws.com weight=1;
        server ec2-52-44-134-202.compute-1.amazonaws.com weight=2;
         }

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';

    access_log  /var/log/nginx/access.log  main;

-- INSERT --                                                          1,1           Top
```

ec2-user@ip-172-31-44-3:/etc/nginx

```
[ec2-user@ip-172-31-44-3 nginx]$ sudo /etc/init.d/nginx reload
Reloading nginx:                                           [  OK  ]
[ec2-user@ip-172-31-44-3 nginx]$ visit_server -d ec2-18-205-42-170.compute-1.amazonaws.com
Starting to visit load balancing server
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................................
.....................................................................................
------------------------
 Summary
------------------------
Server1 visit counts : 333
Server2 visit counts : 667
Server3 visit counts : 333
Server4 visit counts : 667
Total visit counts : 2000
[ec2-user@ip-172-31-44-3 nginx]$
```

Steps to tcpdump Analysis using Amazon EC2 instance

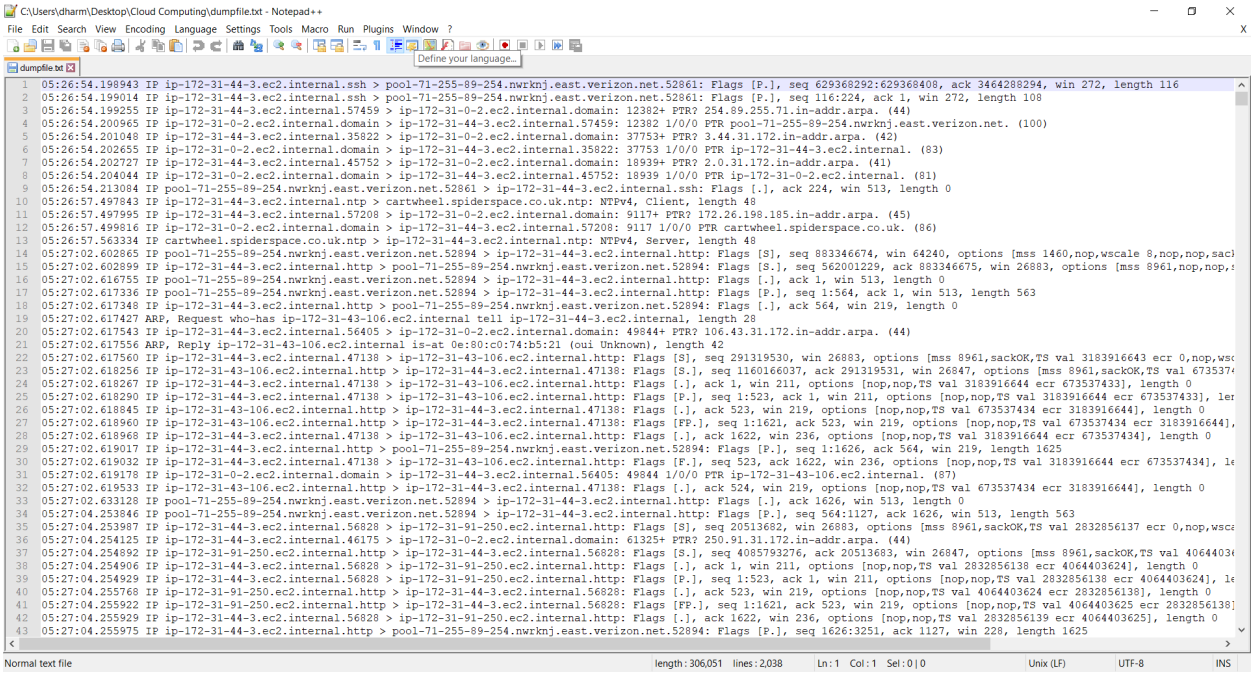
- Installing tcpdump packages

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ sudo yum install libpcap tcpdump ethereal
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main                                                                | 2.1 kB  00:00:00
amzn-updates                                                             | 2.5 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package libpcap.x86_64 14:1.4.0-1.20130826git2dbcaa1.10.amzn1 will be installed
---> Package tcpdump.x86_64 14:4.0.0-3.20090921gitdf3cb4.2.10.amzn1 will be installed
---> Package wireshark.x86_64 0:1.8.10-25.22.amzn1 will be installed
--> Processing Dependency: libgnutls.so.26(GNUTLS_1_4)(64bit) for package: wireshark-1.8.10-25.22.amzn1.x86_64
--> Processing Dependency: libsmi.so.2()(64bit) for package: wireshark-1.8.10-25.22.amzn1.x86_64
--> Processing Dependency: libgnutls.so.26()(64bit) for package: wireshark-1.8.10-25.22.amzn1.x86_64
--> Running transaction check
---> Package gnutls.x86_64 0:2.12.23-21.18.amzn1 will be installed
---> Package libsmi.x86_64 0:0.4.8-4.6.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package              Arch             Version                                    Repository            Size
================================================================================================================
Installing:
 libpcap              x86_64           14:1.4.0-1.20130826git2dbcaa1.10.amzn1     amzn-main            144 k
 tcpdump              x86_64           14:4.0.0-3.20090921gitdf3cb4.2.10.amzn1    amzn-main            372 k
 wireshark            x86_64           1.8.10-25.22.amzn1                         amzn-main             15 M
Installing for dependencies:
 gnutls               x86_64           2.12.23-21.18.amzn1                        amzn-main            450 k
 libsmi               x86_64           0.4.8-4.6.amzn1                            amzn-main            2.8 M
```

- Running tcpdump command first time and creating report in dumpfile.txt file

```
ec2-user@ip-172-31-44-3:~
[ec2-user@ip-172-31-44-3 ~]$ sudo tcpdump -i eth0 >> dumpfile.txt
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes

^C63 packets captured
65 packets received by filter
0 packets dropped by kernel
[ec2-user@ip-172-31-44-3 ~]$
```

- tcpdump data is logged into dumpfile.txt




```
05:26:54.198943 IP ip-172-31-44-3.ec2.internal.ssh > pool-71-255-89-254.nwrknj.east.verizon.net.52861: Flags [P.], seq 629368292:629368408, ack 3464288294, win 272, length 116
05:26:54.199014 IP ip-172-31-44-3.ec2.internal.ssh > pool-71-255-89-254.nwrknj.east.verizon.net.52861: Flags [P.], seq 116:224, ack 1, win 272, length 108
05:26:54.199255 IP ip-172-31-44-3.ec2.internal.57459 > ip-172-31-0-2.ec2.internal.domain: 12382+ PTR? 254.89.255.71.in-addr.arpa. (44)
05:26:54.200965 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-44-3.ec2.internal.57459: 12382 1/0/0 PTR pool-71-255-89-254.nwrknj.east.verizon.net. (100)
05:26:54.201048 IP ip-172-31-44-3.ec2.internal.35822 > ip-172-31-0-2.ec2.internal.domain: 37753+ PTR? 3.44.31.172.in-addr.arpa. (42)
05:26:54.202655 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-44-3.ec2.internal.35822: 37753 1/0/0 PTR ip-172-31-44-3.ec2.internal. (83)
05:26:54.202727 IP ip-172-31-44-3.ec2.internal.45752 > ip-172-31-0-2.ec2.internal.domain: 18939+ PTR? 2.0.31.172.in-addr.arpa. (41)
05:26:54.204044 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-44-3.ec2.internal.45752: 18939 1/0/0 PTR ip-172-31-0-2.ec2.internal. (81)
05:26:54.213084 IP pool-71-255-89-254.nwrknj.east.verizon.net.52861 > ip-172-31-44-3.ec2.internal.ssh: Flags [.], ack 224, win 513, length 0
05:26:57.497843 IP ip-172-31-44-3.ec2.internal.ntp > cartwheel.spiderspace.co.uk.ntp: NTPv4, Client, length 48
05:26:57.497995 IP ip-172-31-44-3.ec2.internal.57208 > ip-172-31-0-2.ec2.internal.domain: 9117+ PTR? 172.26.198.185.in-addr.arpa. (45)
05:26:57.499816 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-44-3.ec2.internal.57208: 9117 1/0/0 PTR cartwheel.spiderspace.co.uk. (86)
05:26:57.563334 IP cartwheel.spiderspace.co.uk.ntp > ip-172-31-44-3.ec2.internal.ntp: NTPv4, Server, length 48
05:27:02.602865 IP pool-71-255-89-254.nwrknj.east.verizon.net.52894 > ip-172-31-44-3.ec2.internal.http: Flags [S], seq 883346674, win 64240, options [mss 1460,nop,wscale 8,nop,nop,sack
05:27:02.602899 IP ip-172-31-44-3.ec2.internal.http > pool-71-255-89-254.nwrknj.east.verizon.net.52894: Flags [S.], seq 562001229, ack 883346675, win 26883, options [mss 8961,nop,nop,s
05:27:02.616755 IP pool-71-255-89-254.nwrknj.east.verizon.net.52894 > ip-172-31-44-3.ec2.internal.http: Flags [.], ack 1, win 513, length 0
05:27:02.617336 IP pool-71-255-89-254.nwrknj.east.verizon.net.52894 > ip-172-31-44-3.ec2.internal.http: Flags [P.], seq 1:564, ack 1, win 513, length 563
05:27:02.617348 IP ip-172-31-44-3.ec2.internal.http > pool-71-255-89-254.nwrknj.east.verizon.net.52894: Flags [.], ack 564, win 219, length 0
05:27:02.617427 ARP, Request who-has ip-172-31-43-106.ec2.internal tell ip-172-31-44-3.ec2.internal, length 28
05:27:02.617543 IP ip-172-31-44-3.ec2.internal.56405 > ip-172-31-0-2.ec2.internal.domain: 49844+ PTR? 106.43.31.172.in-addr.arpa. (44)
05:27:02.617556 ARP, Reply ip-172-31-43-106.ec2.internal is-at 0e:80:c0:74:b5:21 (oui Unknown), length 42
05:27:02.617560 IP ip-172-31-44-3.ec2.internal.47138 > ip-172-31-43-106.ec2.internal.http: Flags [S], seq 291319530, win 26883, options [mss 8961,sackOK,TS val 3183916643 ecr 0,nop,wsc
05:27:02.618256 IP ip-172-31-43-106.ec2.internal.http > ip-172-31-44-3.ec2.internal.47138: Flags [S.], seq 1160166037, ack 291319531, win 26847, options [mss 8961,sackOK,TS val 6735374
05:27:02.618267 IP ip-172-31-44-3.ec2.internal.47138 > ip-172-31-43-106.ec2.internal.http: Flags [.], ack 1, win 211, options [nop,nop,TS val 3183916644 ecr 673537433], length 0
05:27:02.618290 IP ip-172-31-44-3.ec2.internal.47138 > ip-172-31-43-106.ec2.internal.http: Flags [P.], seq 1:523, ack 1, win 211, options [nop,nop,TS val 3183916644 ecr 673537433], len
05:27:02.618845 IP ip-172-31-43-106.ec2.internal.http > ip-172-31-44-3.ec2.internal.47138: Flags [.], ack 523, win 219, options [nop,nop,TS val 673537434 ecr 3183916644], length 0
05:27:02.618960 IP ip-172-31-43-106.ec2.internal.http > ip-172-31-44-3.ec2.internal.47138: Flags [FP.], seq 1:1621, ack 523, win 219, options [nop,nop,TS val 673537434 ecr 3183916644],
05:27:02.618968 IP ip-172-31-44-3.ec2.internal.47138 > ip-172-31-43-106.ec2.internal.http: Flags [.], ack 1622, win 236, options [nop,nop,TS val 3183916644 ecr 673537434], length 0
05:27:02.619017 IP ip-172-31-44-3.ec2.internal.http > pool-71-255-89-254.nwrknj.east.verizon.net.52894: Flags [P.], seq 1:1626, ack 564, win 219, length 1625
05:27:02.619032 IP ip-172-31-44-3.ec2.internal.47138 > ip-172-31-43-106.ec2.internal.http: Flags [P.], seq 523, ack 1622, win 236, options [nop,nop,TS val 3183916644 ecr 673537434], le
05:27:02.619178 IP ip-172-31-0-2.ec2.internal.domain > ip-172-31-44-3.ec2.internal.56405: 49844 1/0/0 PTR ip-172-31-43-106.ec2.internal. (87)
05:27:02.619533 IP ip-172-31-43-106.ec2.internal.http > ip-172-31-44-3.ec2.internal.47138: Flags [.], ack 524, win 219, options [nop,nop,TS val 673537434 ecr 3183916644], length 0
05:27:02.633128 IP pool-71-255-89-254.nwrknj.east.verizon.net.52894 > ip-172-31-44-3.ec2.internal.http: Flags [.], ack 1626, win 513, length 0
05:27:04.253846 IP pool-71-255-89-254.nwrknj.east.verizon.net.52894 > ip-172-31-44-3.ec2.internal.http: Flags [P.], seq 564:1127, ack 1626, win 513, length 563
05:27:04.253987 IP ip-172-31-44-3.ec2.internal.56828 > ip-172-31-91-250.ec2.internal.http: Flags [S], seq 20513682, win 26883, options [mss 8961,sackOK,TS val 2832856137 ecr 0,nop,wsca
05:27:04.254125 IP ip-172-31-44-3.ec2.internal.46175 > ip-172-31-0-2.ec2.internal.domain: 61325+ PTR? 250.91.31.172.in-addr.arpa. (44)
05:27:04.254892 IP ip-172-31-91-250.ec2.internal.http > ip-172-31-44-3.ec2.internal.56828: Flags [S.], seq 4085793276, ack 20513683, win 26847, options [mss 8961,sackOK,TS val 4064403
05:27:04.254906 IP ip-172-31-44-3.ec2.internal.56828 > ip-172-31-91-250.ec2.internal.http: Flags [.], ack 1, win 211, options [nop,nop,TS val 2832856138 ecr 4064403624], length 0
05:27:04.254929 IP ip-172-31-44-3.ec2.internal.56828 > ip-172-31-91-250.ec2.internal.http: Flags [P.], seq 1:523, ack 1, win 211, options [nop,nop,TS val 2832856138 ecr 4064403624], le
05:27:04.255768 IP ip-172-31-91-250.ec2.internal.http > ip-172-31-44-3.ec2.internal.56828: Flags [.], ack 523, win 219, options [nop,nop,TS val 4064403624 ecr 2832856138], length 0
05:27:04.255922 IP ip-172-31-91-250.ec2.internal.http > ip-172-31-44-3.ec2.internal.56828: Flags [FP.], seq 1:1621, ack 523, win 219, options [nop,nop,TS val 4064403625 ecr 2832856138]
05:27:04.255929 IP ip-172-31-44-3.ec2.internal.56828 > ip-172-31-91-250.ec2.internal.http: Flags [.], ack 1622, win 236, options [nop,nop,TS val 2832856139 ecr 4064403625], length 0
05:27:04.255975 IP ip-172-31-44-3.ec2.internal.http > pool-71-255-89-254.nwrknj.east.verizon.net.52894: Flags [P.], seq 1626:3251, ack 1127, win 228, length 1625
```

- When we analyze the file contents, we see the first few lines being sent by the remote host to my desktop. Then the remote host issues an ARP request to get its own mac address. Since I had made a http request to the load balancer while tcpdump was running (on the load balancer), there are packet information from my local desktop to the load balancer, then from the load balancer to one of the servers, and finally back all the way to my local desktop.

Steps to Backup and Restore Volume on Amazon EC2 instance

- Creating snapshot of existing volume

  $ aws ec2 create-snapshot --volume-id vol-01bbd0a68eef4638c --description “Load Balancer Snapshot”

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 create-snapshot --volume-id vol-01bbd0a68eef4638c --description "Load Bal
ancer Snapshot"
{
    "Description": "Load Balancer Snapshot",
    "Encrypted": false,
    "OwnerId": "498314746917",
    "Progress": "",
    "SnapshotId": "snap-0b522323f7840707b",
    "StartTime": "2020-04-14T05:53:52+00:00",
    "State": "pending",
    "VolumeId": "vol-01bbd0a68eef4638c",
    "VolumeSize": 8,
    "Tags": []
}


C:\Users\dharm\Desktop\Cloud Computing>
```

- Stopping the instance to detach the volume

  $ aws ec2 stop-instances --instance-ids i-0093dc8ace64f4fb6

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 stop-instances --instance-ids i-0093dc8ace64f4fb6
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-0093dc8ace64f4fb6",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}


C:\Users\dharm\Desktop\Cloud Computing>
```

- Detaching the volume

  $ aws ec2 detach-volume --volume-id vol-01bbd0a68eef4638c

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 detach-volume --volume-id vol-01bbd0a68eef4638c
{
    "AttachTime": "2020-02-18T23:24:20+00:00",
    "Device": "/dev/xvda",
    "InstanceId": "i-0093dc8ace64f4fb6",
    "State": "detaching",
    "VolumeId": "vol-01bbd0a68eef4638c"
}

C:\Users\dharm\Desktop\Cloud Computing>
```

- Creating new EC2 instance

  $ aws ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids default --count 1 --instance-type t2.micro --key-name ec2@dharmit

```
C:\WINDOWS\system32\cmd.exe - aws  ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids default --count 1...
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 run-instances --image-id ami-0915e09cc7ceee3ab --security-group-ids defau
lt --count 1 --instance-type t2.micro --key-name ec2@dharmit
{
    "Groups": [],
    "Instances": [
        {
            "AmiLaunchIndex": 0,
            "ImageId": "ami-0915e09cc7ceee3ab",
            "InstanceId": "i-04cb05bdbbc9a3f64",
            "InstanceType": "t2.micro",
            "KeyName": "ec2@dharmit",
            "LaunchTime": "2020-04-14T20:44:36+00:00",
            "Monitoring": {
                "State": "disabled"
            },
            "Placement": {
                "AvailabilityZone": "us-east-1a",
                "GroupName": "",
                "Tenancy": "default"
            },
            "PrivateDnsName": "ip-172-31-85-75.ec2.internal",
            "PrivateIpAddress": "172.31.85.75",
            "ProductCodes": [],
            "PublicDnsName": "",
            "State": {
                "Code": 0,
                "Name": "pending"
            },
            "StateTransitionReason": "",
            "SubnetId": "subnet-727f535c",
            "VpcId": "vpc-1993e163",
```

- Stopping the new created instance to detach the volume

  $ aws ec2 stop-instances --instance-ids i-04cb05bdbbc9a3f64

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 stop-instances --instance-ids i-04cb05bdbbc9a3f64
{
    "StoppingInstances": [
        {
            "CurrentState": {
                "Code": 64,
                "Name": "stopping"
            },
            "InstanceId": "i-04cb05bdbbc9a3f64",
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}

C:\Users\dharm\Desktop\Cloud Computing>
```

- Detaching the volume on new instance

  $ aws ec2 detach-volume --volume-id vol-0e746efbf89345e44

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 detach-volume --volume-id vol-0e746efbf89345e44
{
    "AttachTime": "2020-04-14T20:44:37+00:00",
    "Device": "/dev/xvda",
    "InstanceId": "i-04cb05bdbbc9a3f64",
    "State": "detaching",
    "VolumeId": "vol-0e746efbf89345e44"
}

C:\Users\dharm\Desktop\Cloud Computing>
```

- Attaching the previously detached volume to the new instance

  $ aws ec2 attach-volume --volume-id vol-01bbd0a68eef4638c --instance-id i-0046dc440e2215b76 --device /dev/xvda

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 attach-volume --volume-id vol-01bbd0a68eef4638c --instance-id i-0046dc440
e2215b76 --device /dev/xvda
{
    "AttachTime": "2020-04-14T21:07:25.945000+00:00",
    "Device": "/dev/xvda",
    "InstanceId": "i-0046dc440e2215b76",
    "State": "attaching",
    "VolumeId": "vol-01bbd0a68eef4638c"
}

C:\Users\dharm\Desktop\Cloud Computing>
```

- Starting the new instance

  $ aws ec2 start-instances --instance-ids i-0046dc440e2215b76

```
C:\WINDOWS\system32\cmd.exe
C:\Users\dharm\Desktop\Cloud Computing>aws ec2 start-instances --instance-ids i-0046dc440e2215b76
{
    "StartingInstances": [
        {
            "CurrentState": {
                "Code": 0,
                "Name": "pending"
            },
            "InstanceId": "i-0046dc440e2215b76",
            "PreviousState": {
                "Code": 80,
                "Name": "stopped"
            }
        }
    ]
}

C:\Users\dharm\Desktop\Cloud Computing>
```

- Connecting to the new instance

  $ ssh -i "ec2@dharmit.pem" ec2-user@ec2-54-89-120-136.compute-1.amazonaws.com

```
Select ec2-user@ip-172-31-43-212:~
C:\Users\dharm\Desktop\Cloud Computing>ssh -i "ec2@dharmit.pem" ec2-user@ec2-54-89-120-136.compute-1.amazonaws.com
The authenticity of host 'ec2-54-89-120-136.compute-1.amazonaws.com (54.89.120.136)' can't be established.
ECDSA key fingerprint is SHA256:YmzJWXluBlQcBlvO2PssaVTboI+zYfsYm56Pyt4hR38.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'ec2-54-89-120-136.compute-1.amazonaws.com,54.89.120.136' (ECDSA) to the list of known hosts.

Last login: Tue Apr 14 05:46:41 2020 from pool-71-255-89-254.nwrknj.east.verizon.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
Last login: Tue Apr 14 05:46:41 2020 from pool-71-255-89-254.nwrknj.east.verizon.net

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-172-31-43-212 ~]$
```

- Starting the nginx on the new instance

  $ sudo service nginx start

```
Select ec2-user@ip-172-31-43-212:~
[ec2-user@ip-172-31-43-212 ~]$ sudo service nginx start
Starting nginx:                                            [  OK  ]
[ec2-user@ip-172-31-43-212 ~]$
```

- Server running on new instances