Lab Assignment 2

- **There are six steps we need to follow to get the final output which are as follows:**

1. <u>The steps for accessing Amazon EC2 Instances are as follows:</u>

- We must have a valid account with AWS. If not then create one and then sign in for accessing the resources.

- After signing in, landed on the dashboard of AWS services. Now clicking on EC2 under All services > Compute > EC2.

Now there are 2 ways to create instances:

A. Step for Creating an Amazon EC2 Instances (Using AWS Panel):

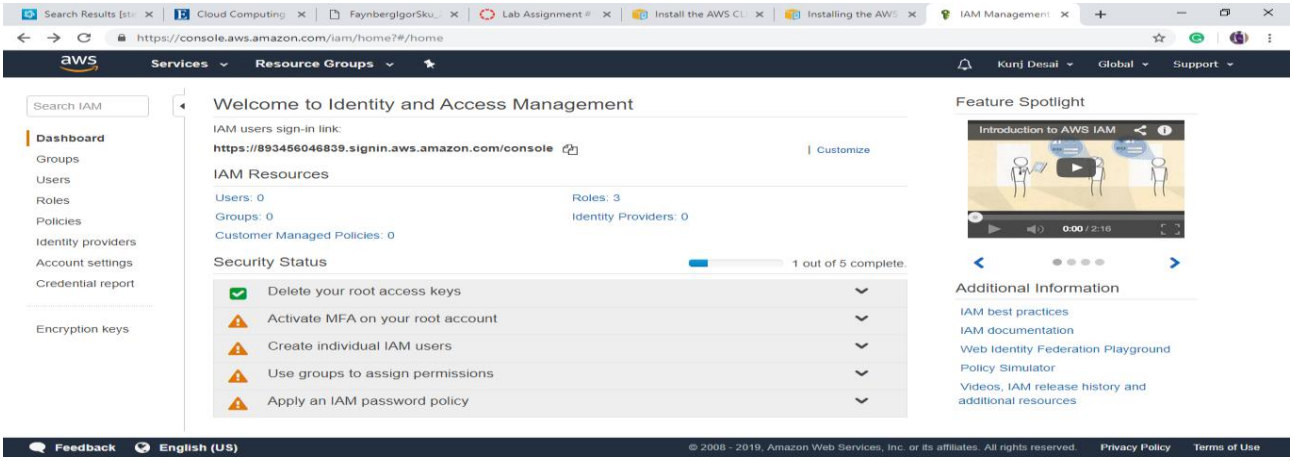
- Now in EC2 Dashboard, Launching Instance by clicking on "**Launch Instance**" button to create new instance and able to
- Check existing running instances by clicking on Running Instances under Resources category.
- Now, this is a first step to create an AWS Instance, here we have to choose an Amazon Machine Image (AMI). Selecting select Amazon Linux AMI (64-bit).
- The next step after selecting AMI is to choose an Instance Type. Selecting **t2.micro** for free tier eligibility.
- As I have selected free tier instance type, I have jumped to the final step in instance creation. In this step I have to review
- The instance information and launch it by clicking on Launch button at the bottom.

- After review, for instance security, I have to assign a key pair to the created instance. I can use existing key pair as I had already created one else I have to create one by choosing other option in dropdown list.
- After creating an instance, it is ready to launch and can view it by clicking View Instance.
- Now, we can view description about your instance and other modules. Now viewing Instances and its description.
- Now, repeat the same steps to create new EC2 instance every time. I have created five instances in total and named them as Load Balancer, Server 1, Server 2, Server 3, and Server 4.

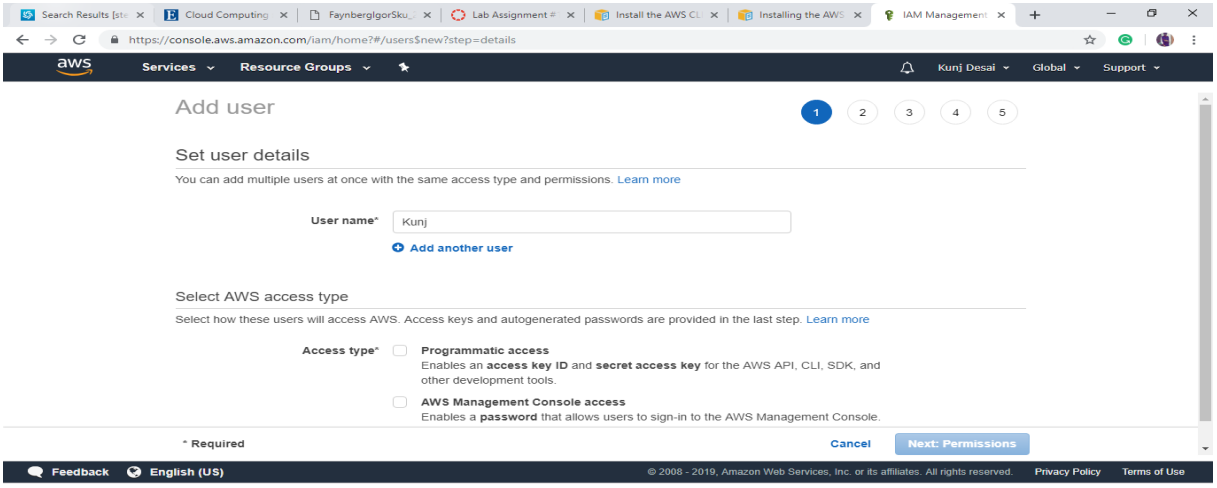
B. Step for Creating an Amazon EC2 Instances (Using AWS Command Line Interface):

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

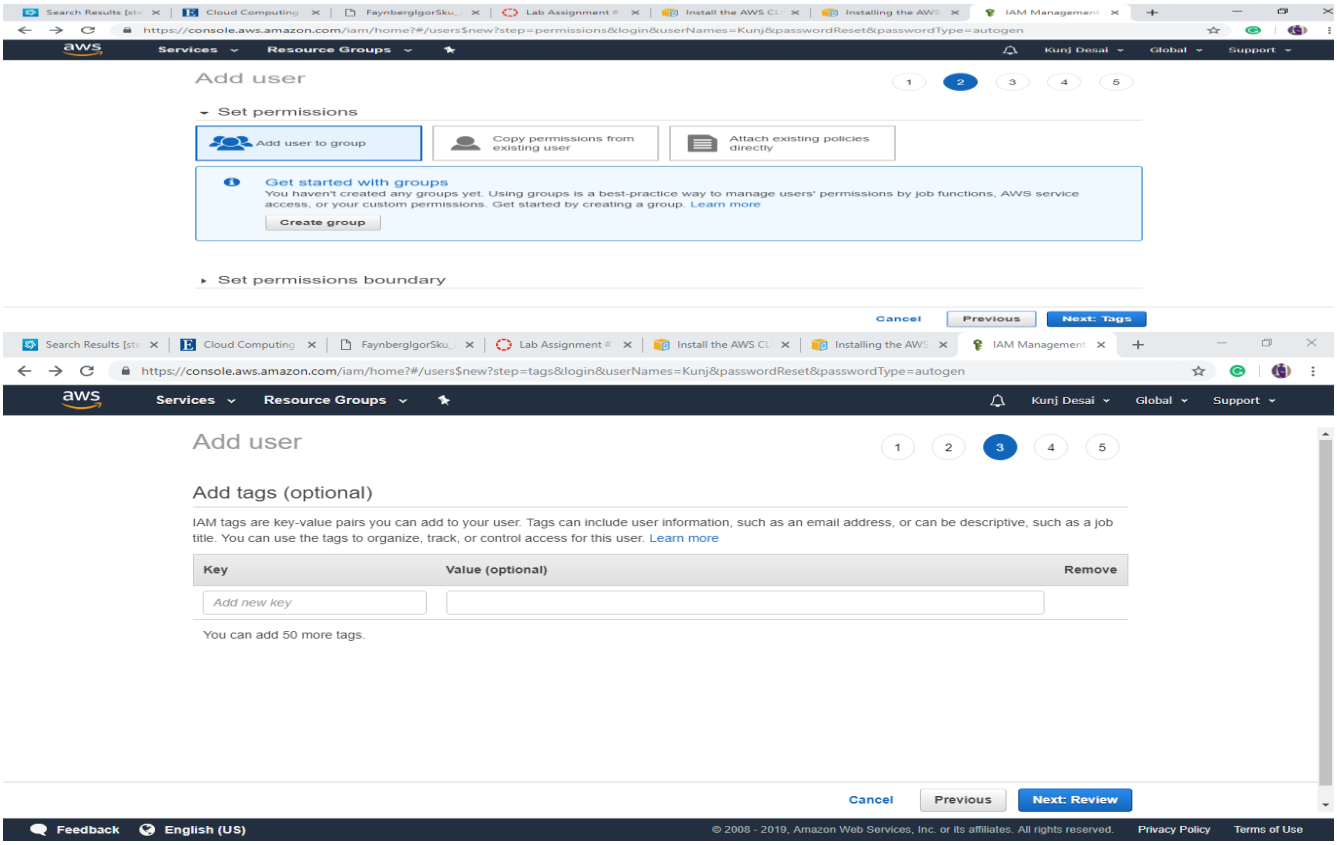
- Now to access the resources, we need **access key id** and **secret access key.** If we are using it first time, then we have to create one by login in to Amazon EC2 and then search for IAM in AWS service dashboard. The IAM dashboard would look like this, having everything 0 under IAM Resources tab.



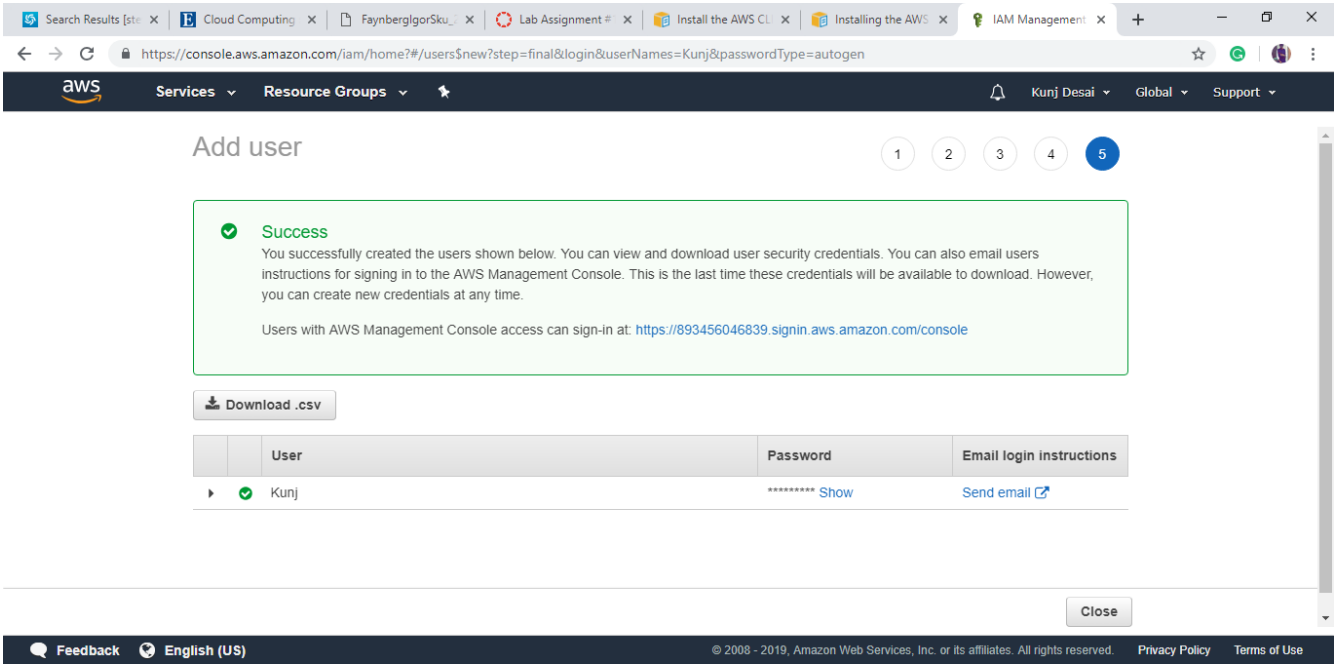
- Now in Users navigation bar and clicking Add user. Then entered a user name and, selected AWS Management Console access and we can either set custom password or auto generate it.



- Now inside permission step, if we wish to create a group then can create here else we will create it later. Next step is Review, to review your details and then, next is to finish the process by clicking complete button.

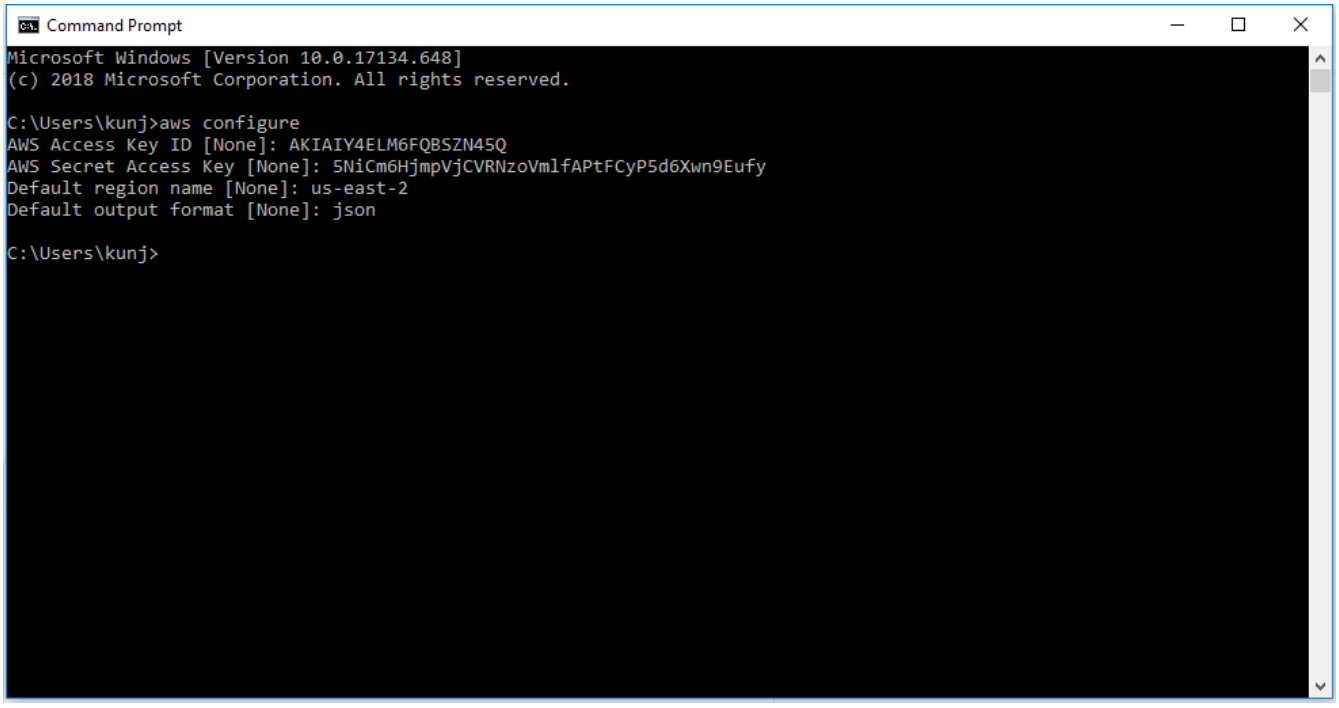


- Now going back to Users tab in IAM dashboard. Selected the user we just created. We will see the details of the user. Clicked on **Security credentials,** under it clicked **Create access key.** From here we will create our Access key id and secret access key.



- 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. Make sure that the policy *AdministratorAccess* is attached with the user. (IAM Dashboard > Policies > Check *AdministratorAccess* > Click Policies actions > Attach and select Username).

  $ aws ec2 create-security-group -–group-name <ENTER NAME> --description "<ENTER DESCRIPTION>"

- **$ aws ec2 authorize-security-group-ingress –-group-name <ENTER_GROUP_NAME> --protocol tcp –-port 22 –-cidr <YOUR_PUBLIC_IP>/32**

```
C:\Users\kunj>aws ec2 create-security-group --group-name cloud-sg --description "cloud lab work"
{
    "GroupId": "sg-00515619d6ec905e0"
}
```

- 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.

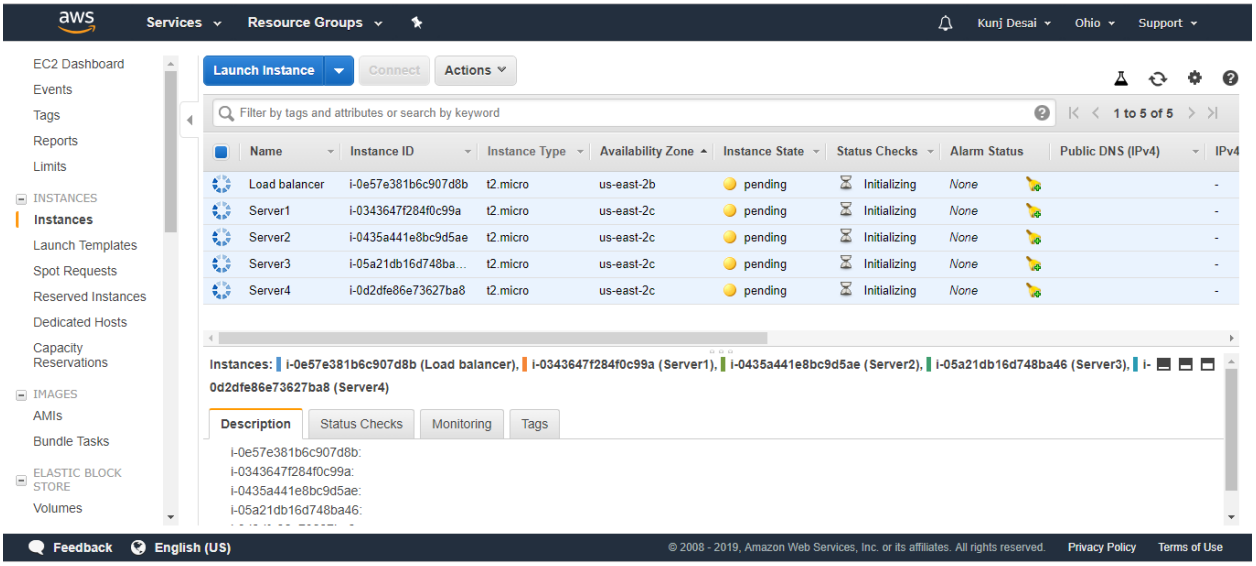
  $ aws ec2 create-key-pair --key-name Kunj_AWS -–query “KeyMaterial” –-ouput text

- 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_id> --security-group-ids <group-id-created-above> --count <number-of-instances> --instance-type <instance-type> --key-name <your-key-pair> --query ‘Instance[0].InstanceId’

- Find AMI and Instance Type
- In my case:
  1. **AMI : ami-8ca83fec**
  2. **Group Id : sg-55cb0f2e**
  3. **Instance Count : 5**
  4. **Instance Type : t2.micro**
  5. **Key Name : Assignment**

- 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 add HTTP port i.e. 80 to the security group for inbound access.

  $ aws ec2 authorize-security-group-ingress –group-name cloud-sg –protocol tcp –port 80 –cidr 192.168.112.1/32

2. <u>The steps for Accessing AWS instance:</u>

- Ensuring read write permission on instance by executing below command

  $ chmod 400 Assignment.pem

- Adding the .pem file to ec2-user in putty use the following command:

  $ pscp -i C:\path\my-key-pair.ppk C:\path\Sample_file.txt ec2-user@public_dns:/home/ec2-user/Sample_file.txt

- Establishing connection with EC2 Instance by executing below command

  $ ssh –I “Assignment.pem” ec2-user@ec2-52-27-159-54.us-west-2.compute.amazonaws.com

3. 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

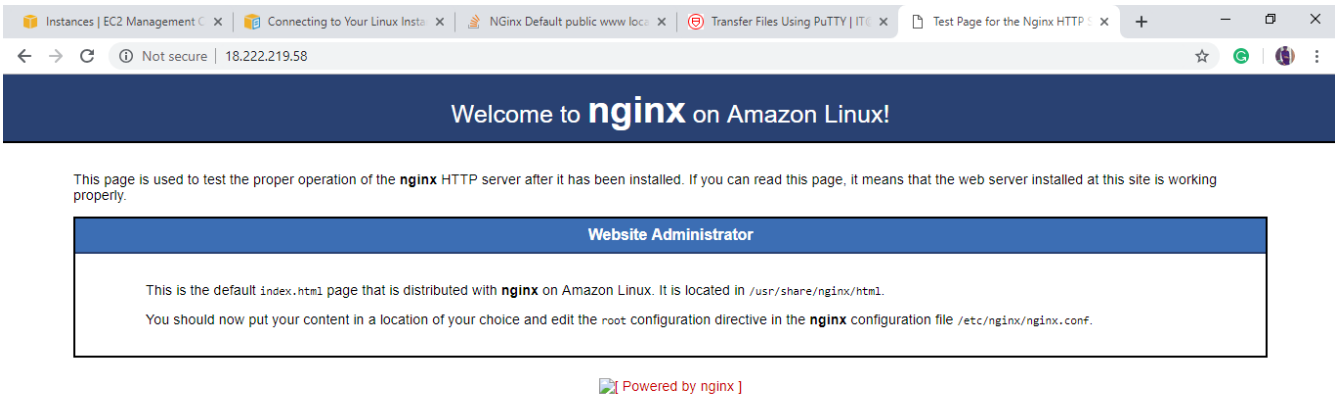
- *(Click **y** for to download and install packages)*

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

  $ sudo service nginx start

- Checking Security Group and adding inbound rule for HTTP for running server on instance.

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



- 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.

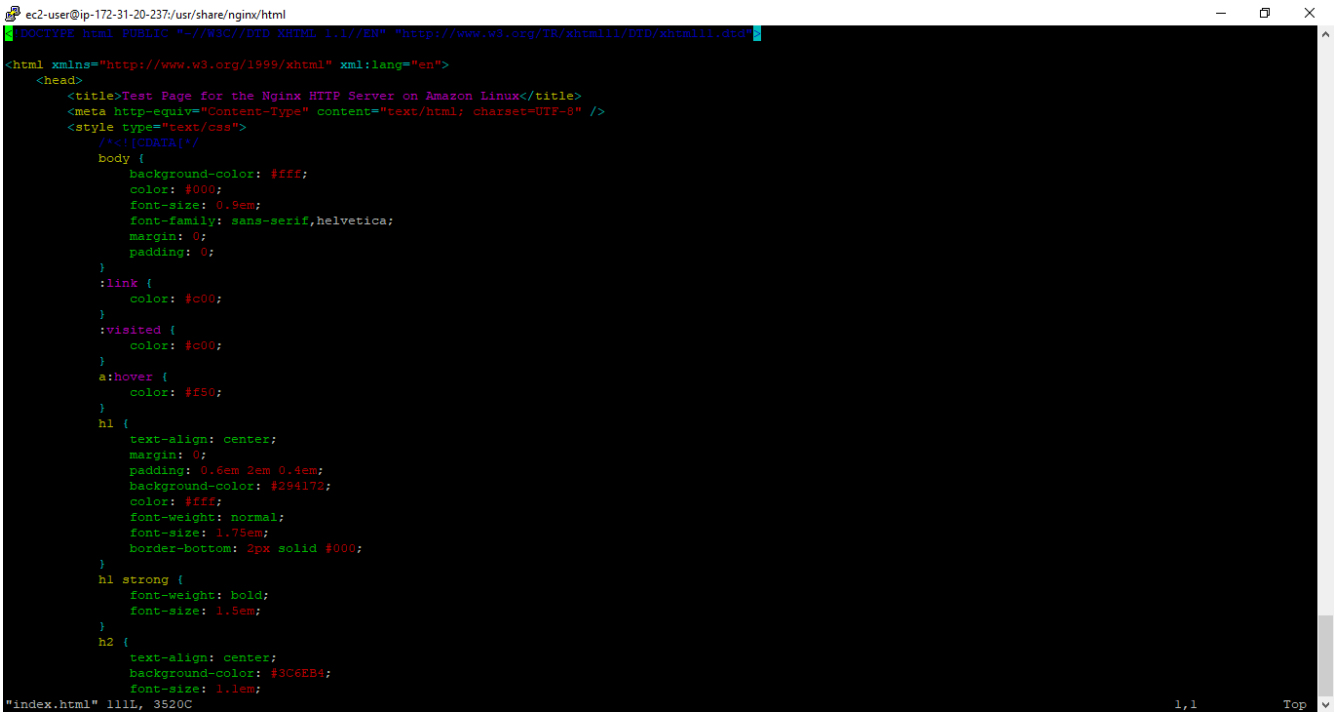
4. Steps to change nginx server index.html file on Amazon EC2 instance:

- 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

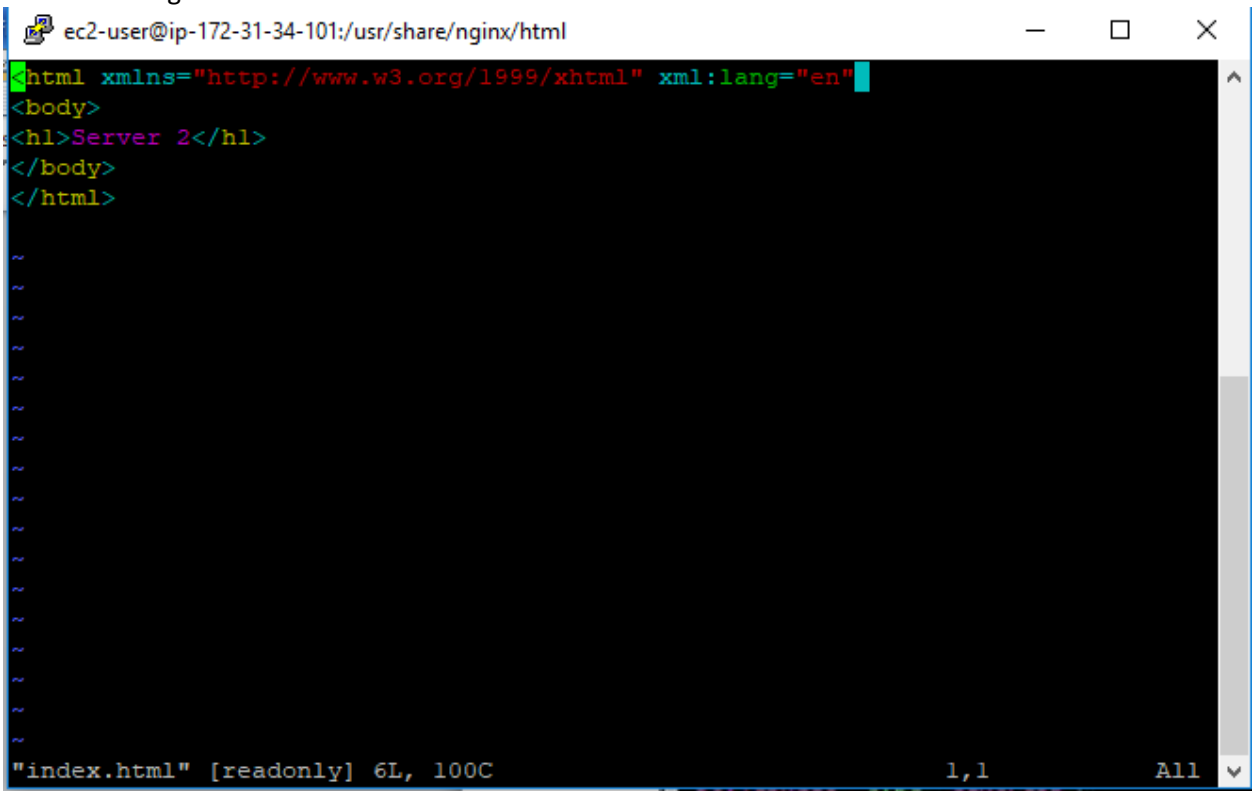
- Then open the index.html file in **vim**. To open type following command in the terminal window

  $ sudo vim index.html

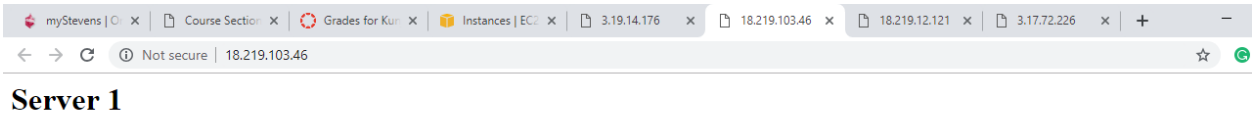


- (***sudo*** command allows you to write a read-only else you can write **:w !sudo tee % > /dev/null** after pressing escape and colon before qutting)

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



- 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. Please find the links below:

- Server 1 : 18.219.103.46
- Server 2 : 18.219.12.121
- Server 3 : 3.17.72.226
- Server 4 : 3.19.14.176

- (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.

5. 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 "Assignment.pem" ec2-18-218-160-216.us-east-2.compute.amazonaws.com

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

$ sudo vim nginx.conf

- Now editing file by adding and replacing code by following codes

```
events{
worker_conncetions 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)

root@ip-172-31-21-241:/etc/nginx

```
events {
worker_connections 768;
}
http {
upstream myapp {
#ip_hash;
server ec2-18-219-103-46.us-east-2.compute.amazonaws.com  weight=1;
server ec2-18-219-12-121.us-east-2.compute.amazonaws.com  weight=1;
server ec2-3-17-72-226.us-east-2.compute.amazonaws.com  weight=1;
server ec2-3-19-14-176.us-east-2.compute.amazonaws.com  weight=1;
}
server {
listen 80;
server_name myapp.com;
location / {
proxy_pass http://myapp;
}
}
}
~
```

- 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 [LOAD_BALANCER_DNS_NAME]

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

6. 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

  $ vim visit_server

- Now, we will trace the load balancing server load distribution, but before that we need to give permission as **visit_server** is executable file by running the following command:

  $ chmod 777 visit_server

➢ **NOTE:** Also to note here, the "visit_server" file should be in the root folder. For getting to the root folder using the following command:

$ cd ~

➢ Also, get the sudo permission using the following command:

$ sudo su

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

```
upstream myapp {
#ip_hash;
server ec2-18-219-103-46.us-east-2.compute.amazonaws.com  weight=1;
server ec2-18-219-12-121.us-east-2.compute.amazonaws.com  weight=1;
server ec2-3-17-72-226.us-east-2.compute.amazonaws.com  weight=1;
server ec2-3-19-14-176.us-east-2.compute.amazonaws.com  weight=1;
}
server {
```

- So the final output is as follows by running the visit_server tool using the following command:

$ ./visit_server –d [LOAD_BALANCER_DNS_NAME]