Q) 1 . Describe VPC and Subnet. Mention its purpose and its significance in a cloud environment

ANS –

Virtual Private Cloud (VPC) networks are global resources. It works similarly like network resources but it gives us control over the network. Each VPC network consists of one or more IP address range called subnets. This means we can provide how many the number of systems can be attached to a particular network. Subnets are regional resources and have IP address ranges associated with them. Subnets work similarly like local IP addresses or we can say that each subnet is associated with a particular system.

In the cloud, it gives us full control over security, resources, and connectivity which is one of the best features in the cloud environment.

Q) 2 . Describe CIDR address

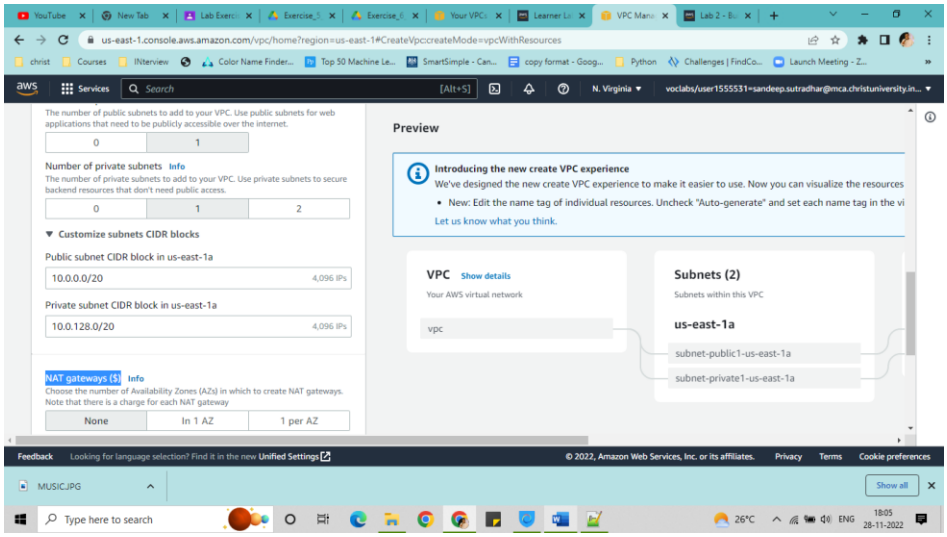
ANS –

CIDR (Classless Inter-Domain Routing or supernetting) addresses are made up of two sets of numbers: a prefix, which is the binary representation of the network address, and a suffix, which declares the total number of bits in the entire address ex 192.168.129.23/17 where 17 tells that we have only 17 bits to assign for assign to different systems and 32 – 17 = 15 bits (First 15 bits)in Ipv4 is fixed. The real notation of CIDR is (2^8.2^8.2^8.2^8) i.e, Highest ip address in IPV4 can be 255.255.255.255. So we can say that in this example 192.168 is fixed and 129.23 can be vary.

Q) 2 . Create a customized network as depicted below using AWS academy canvas environment

ANS –

Step 1 → Open VPC → Create new vpc → select vpc and more → Give (the name in Name tag auto-generation) sslabvpc-vpc → type 10.0.0.0/16 in the IPV4 CIDR block → Keep number of availability zone, number of public subnets and private subnets “1” → In the Customized subnets CIDR blocks -- Change Public subnet CIDR block in us-east-1a to 10.0.0.0/24 and Change Private subnet CIDR block in us-east-1a to 10.0.1.0/24 → Set NAT gateways($) as ln 1 AZ → Set VPC endpoints as none → keep everything as it is → Select Create VPC



Step 2 → It will give the following options by default

Route tables

lab-rtb-public

lab-rtb-private1-us-east-1a

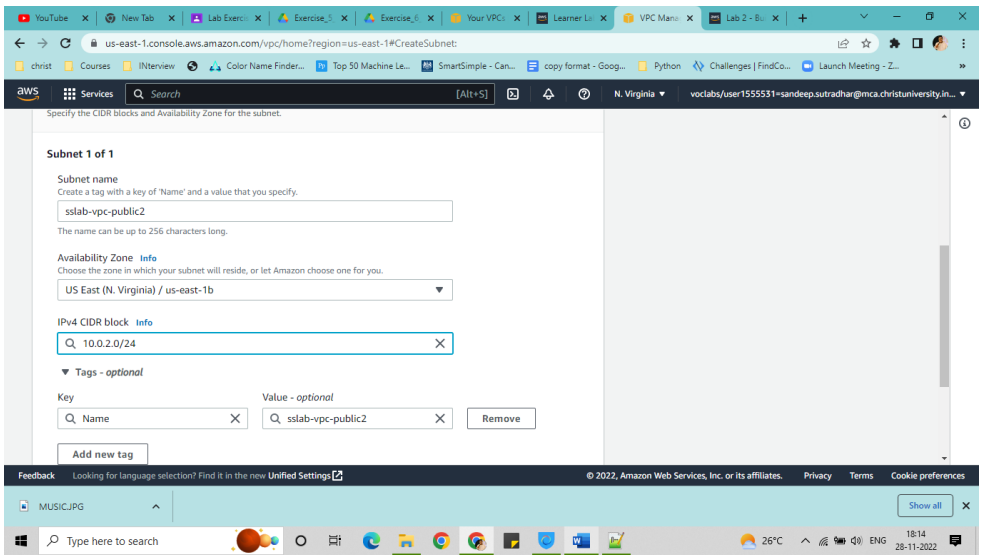
Network connections

lab-igw

lab-nat-public1-us-east-1a

Step 3 → In left side of vpc dashboard → Click Subnets → Click **Create subnet** → Select the VPC ID which we created sslabvpc -vpc → Give Subnet name as sslab-subnet-public2 →

Select the second Availability zone like us-east-1b → in the IPV4 CIDR block give address as 10.0.2.0/24 → Click create subnet



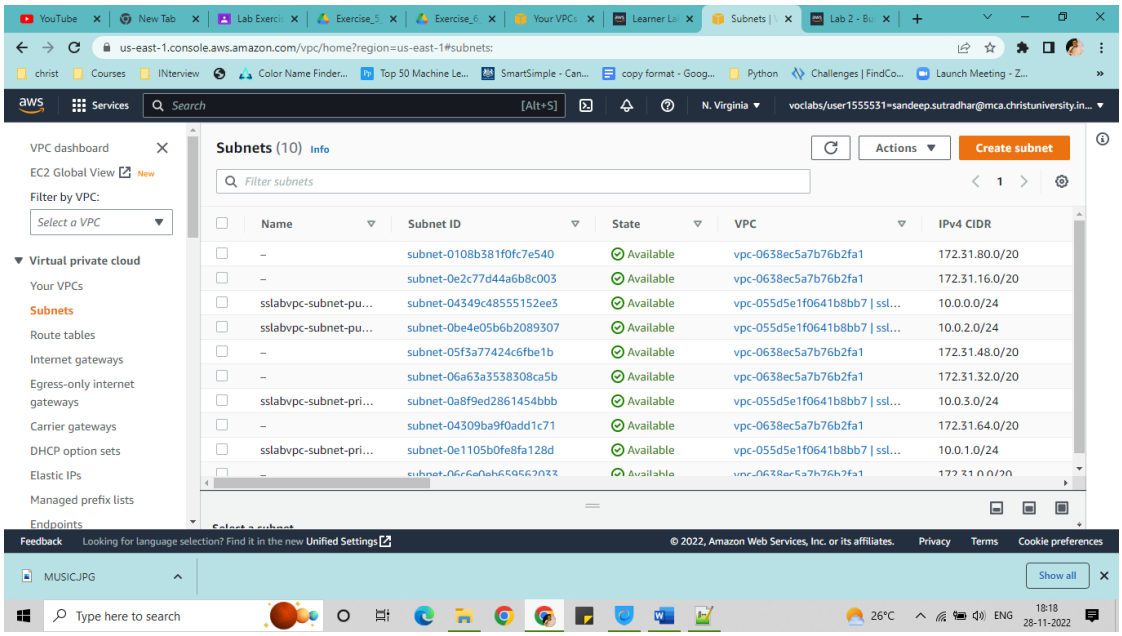
Step 4 → Create 2nd private subnet similar to step 3 with

VPC ID: sslab-vpc

Subnet name: sslab-subnet-private2

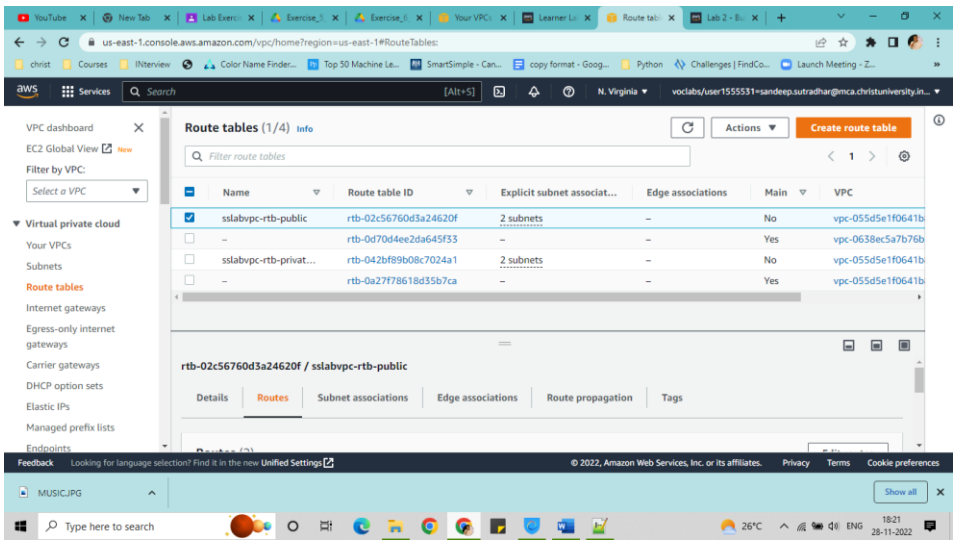
Availability Zone: us-east-1b

IPv4 CIDR block: 10.0.3.0/24 → Create subnet

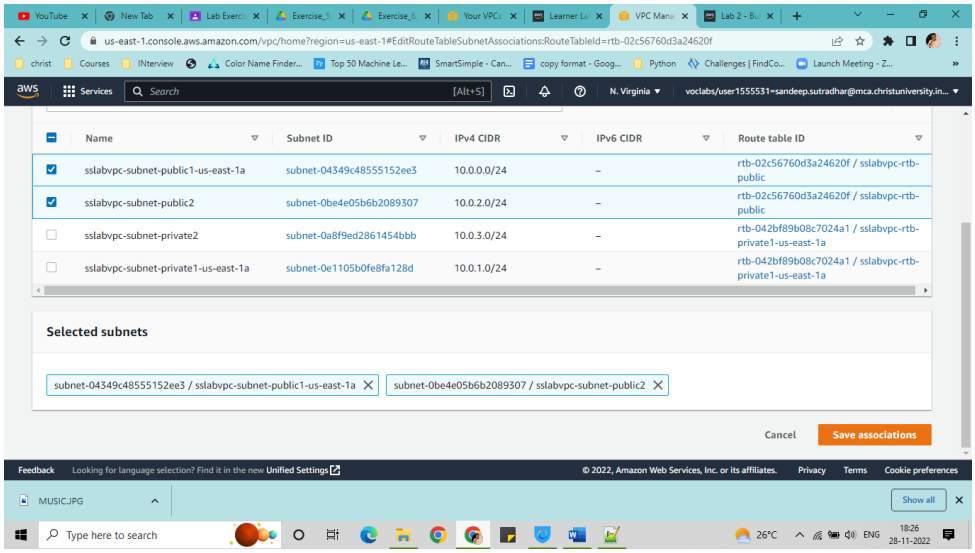


STEP 5 → ROUTE TABLES:

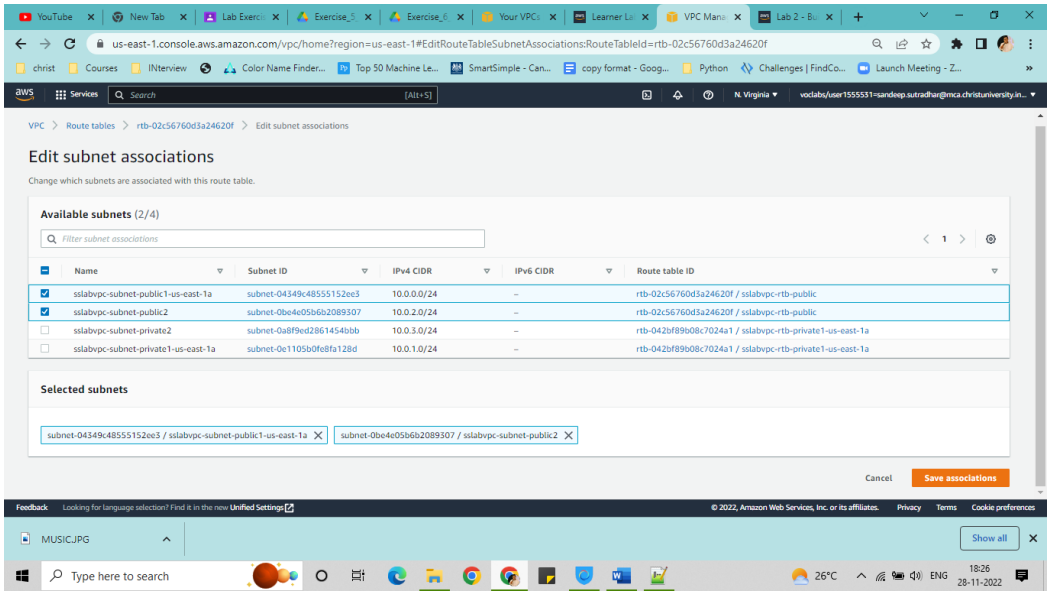
In the left side click Route tables → Select the sslab-rtb-private1-us-east-1a route table → down menu select Routes → this will show the network address which is used to forward any packet to forward to the internet



Choose subnet association → Choose **Edit subnet associations** → **select** both the private networks → Choose **Save associations**

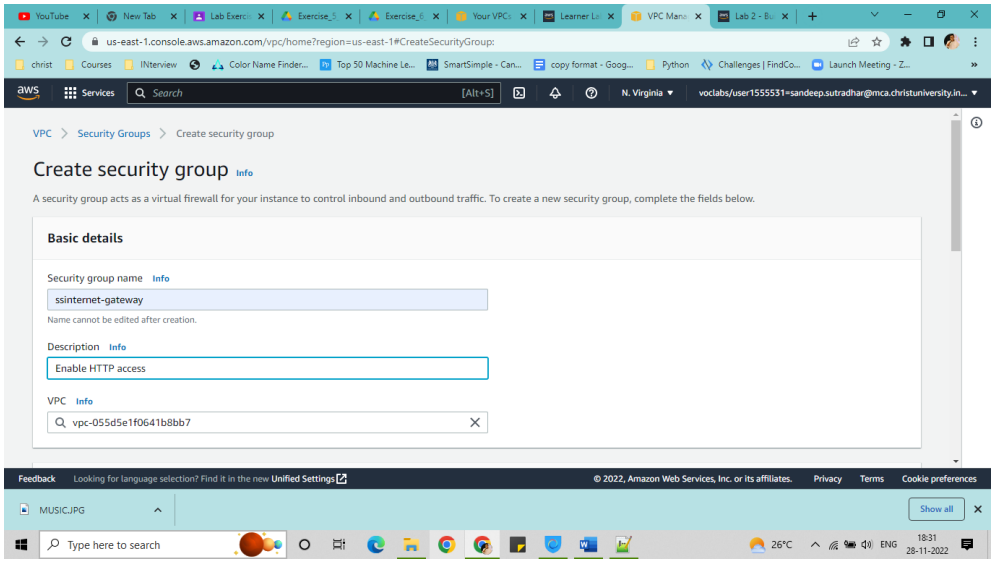


SIMILARLY do it for public subnet association



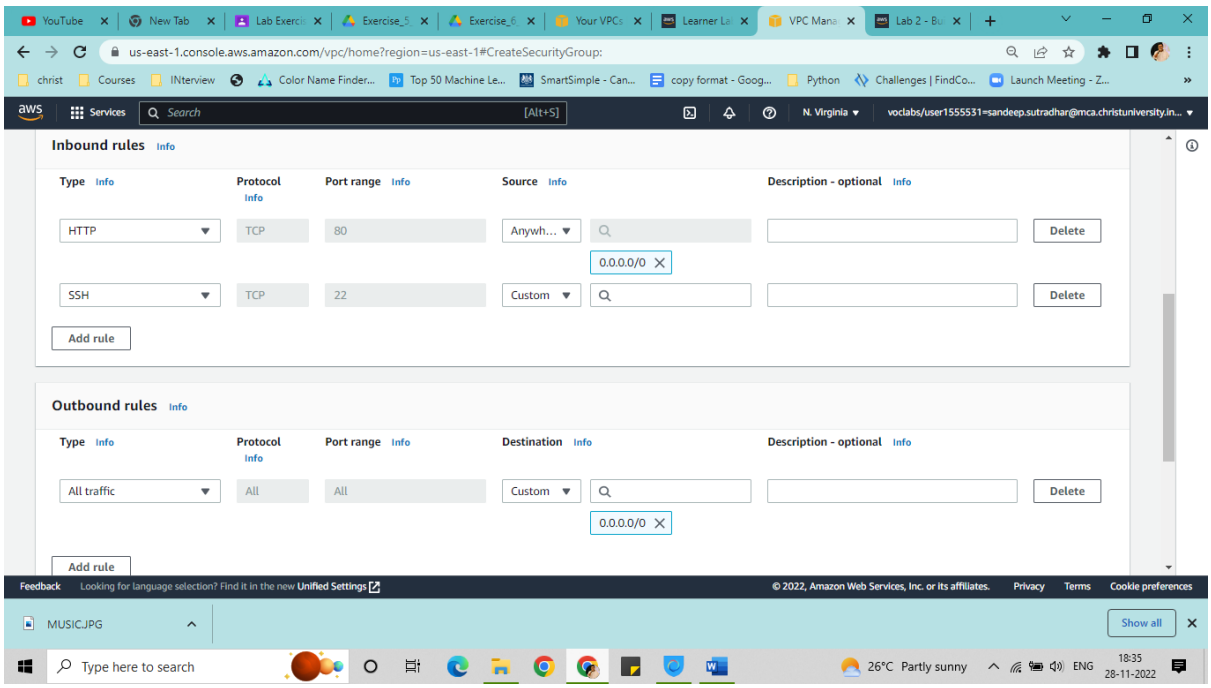
STEP 6 → Create a VPC Security Group:

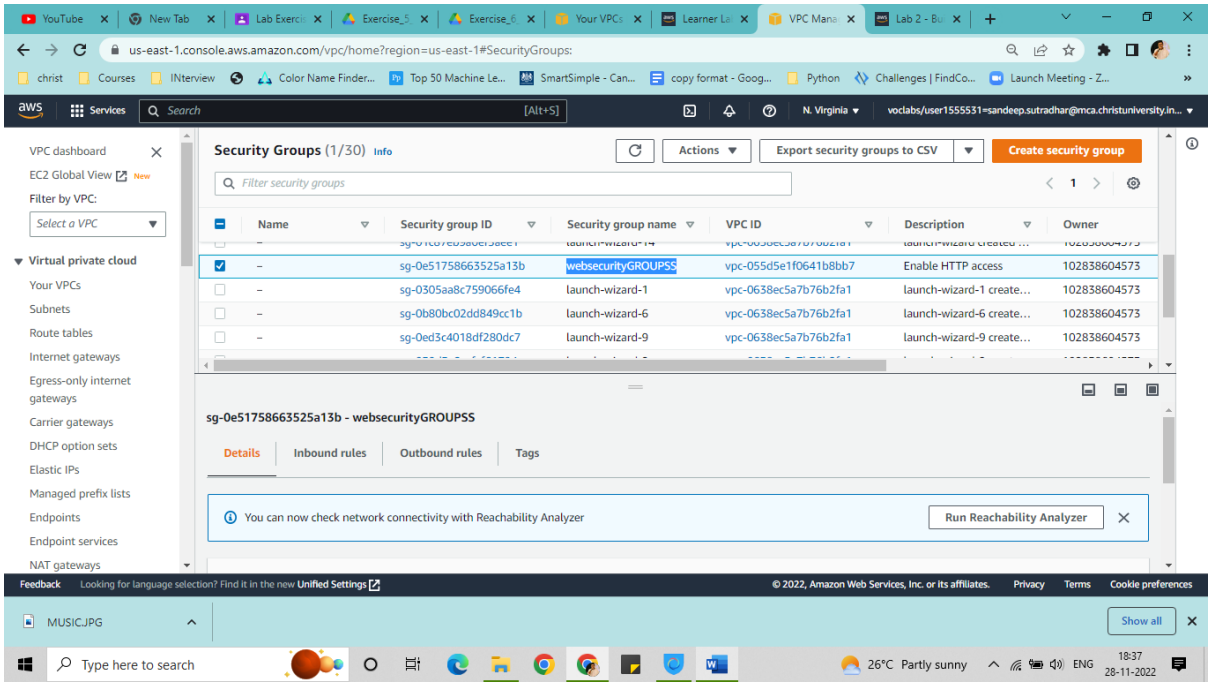
In the left pane → Under security select Security Groups → Choose **Create security group** and then configure → Security group name: WEBsecuritygroupss → Description: Enable HTTP access → in the VPC select the VPC we created and remove the selected one (continue..)



➔ In the inbound rule pane choose **Add rule** → type HTTP and in the Source select Allow from anywhere IPV4 → Description: Permit web requests

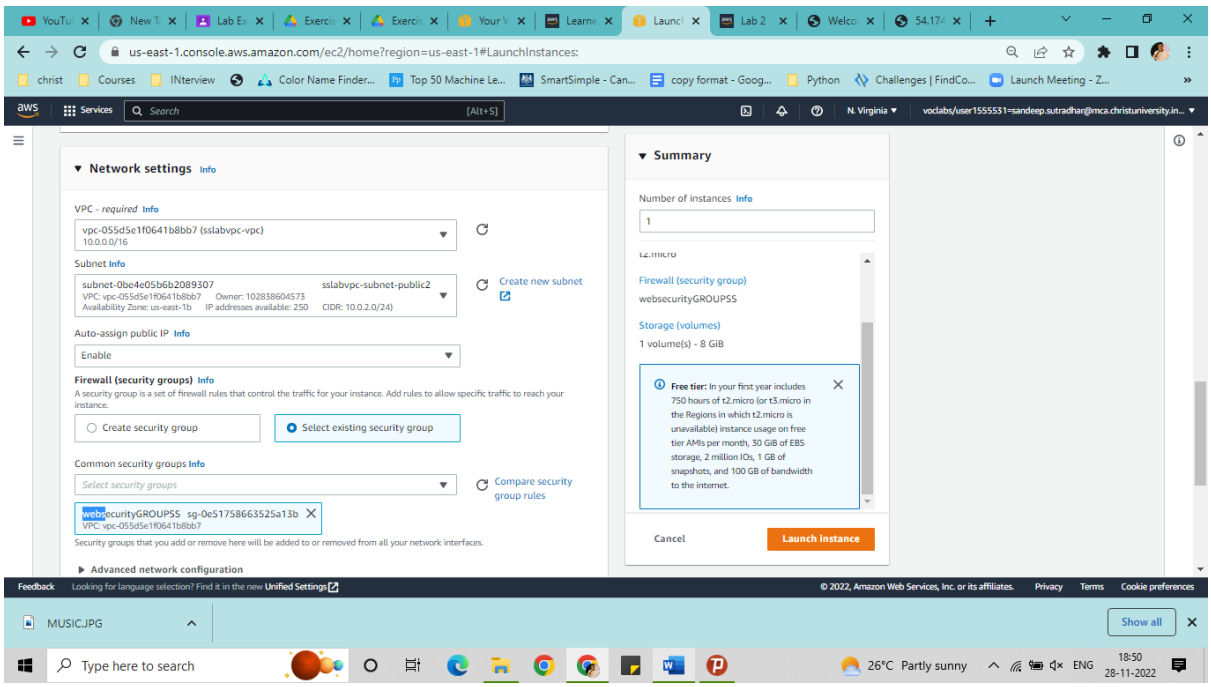
➔ ADD Rule and select ssh → Source select Allow from anywhere IPV4 → Leave everything as it is → select Create security group





STEP 7 → INSTANCE CREATE:

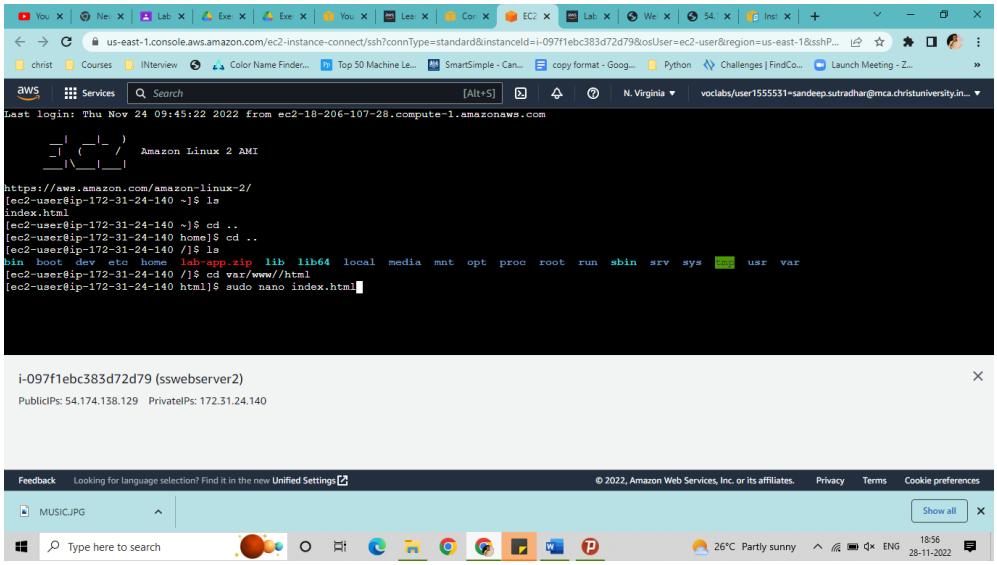
Go to EC2 INSTANCE → create instance with name webserverss2 → change accordingly → In the network setting click on edit → in the VPC select the vpc we created above (sslabvpc-vpc) → select subnet as public subnet2 → Enable the auto assign public IP → In the firewall security groups select Select existing security group → from the dropdown select the above created security group (websecirtygroupss) → Select Launch Instance →



STEP 8→ CREATING A WEB SERVER AND MAKE A WEB PAGE :

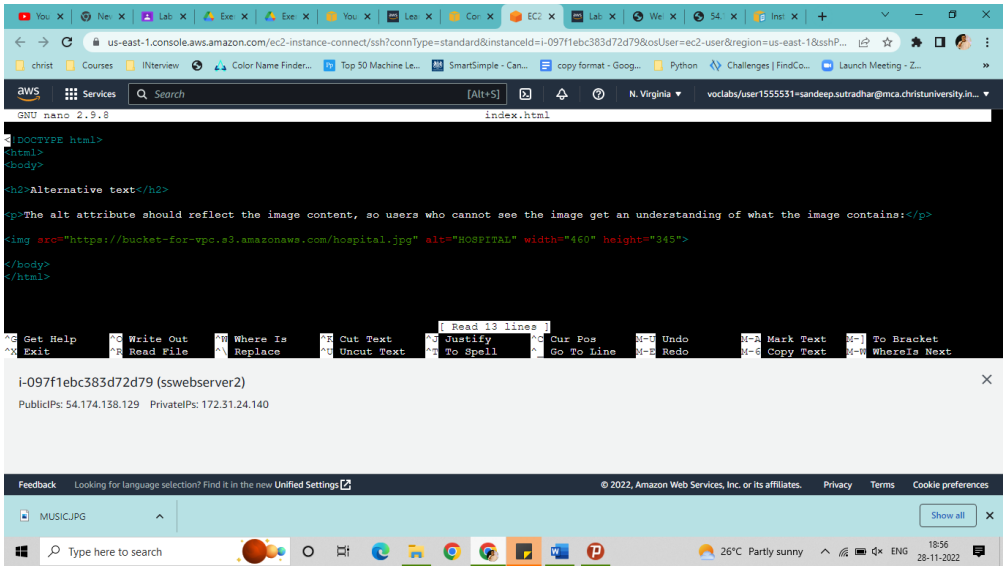
Connect the instance → Write the following code

- Sudo yum update -y
- Sudo amazon-linux-extras install php 8.0 mariadb10.5
- Sudo yum install -y httpd
- Sudo systemctl start httpd
- Sudo systemctl enable httpd
- Cd ..
- Cd ..
- Cd var/www/html
- Sudo nano index.html

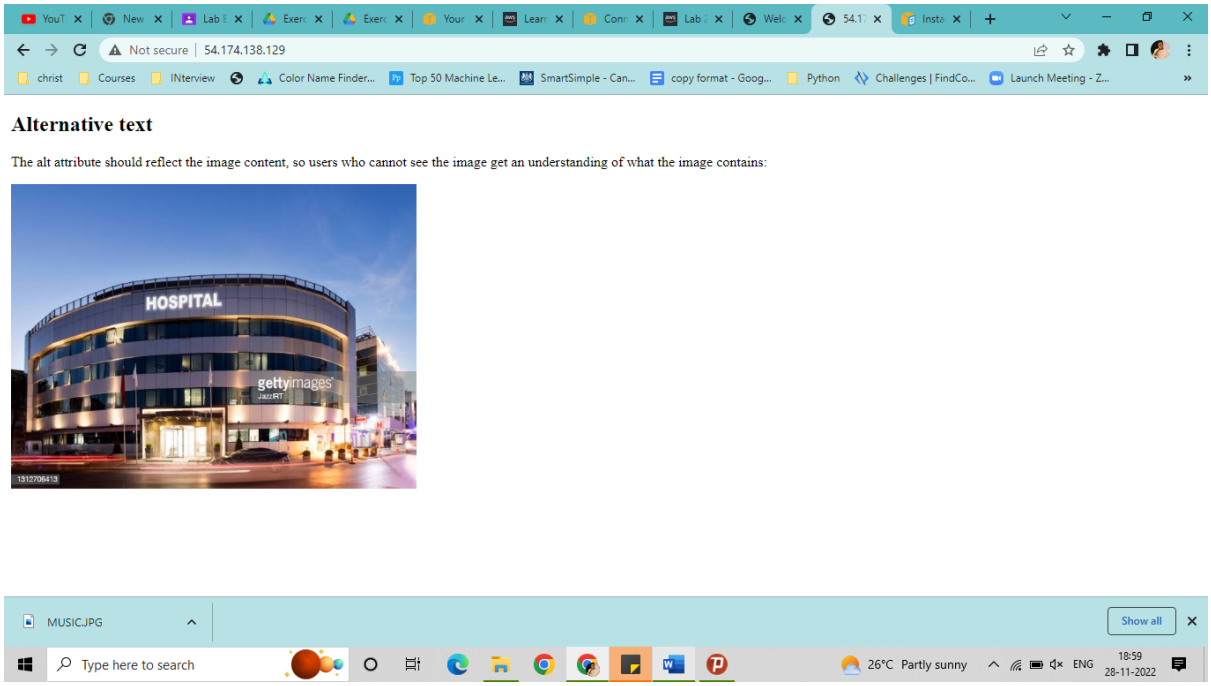



```
<!DOCTYPE html>

<html>

<body>


<h2>Alternative text</h2>


<p>The alt attribute should reflect the image content, so users who cannot see the image get an understanding of what the image contains:</p>


<img src="https://bucket-for-vpc.s3.amazonaws.com/hospital.jpg" alt="HOSPITAL" width="460" height="345">


</body>

</html>
```

Ctrl + o enter → ctrl +x enter


GNU nano 2.9.8
index.html
<!DOCTYPE html>
<html>
<body>
<h2>Alternative text</h2>
The alt attribute should reflect the image content, so users who cannot see the image get an understanding of what the image contains:
<img src="https://bucket-for-vpc.s3.amazonaws.com/hospital.jpg" alt="HOSPITAL" width="460" height="345">
</body>
</html>
[Read 13 lines]
i-097f1ebc383d72d79 (sswebserver2)
PublicIPs: 54.174.138.129 PrivateIPs: 172.31.24.140

OUTPUT :


Not secure | 54.174.138.129
Alternative text
The alt attribute should reflect the image content, so users who cannot see the image get an understanding of what the image contains:
HOSPITAL