what is self signed certificate:-

----------------------------------------

**Web browsers do not recognize the self-signed certificates as valid. When using a self-signed certificate, the web browser shows a warning to the visitor that the web site certificate cannot be verified.**

**Typically, the self-signed certificates are used for testing purposes or internal usage. You should not use a self-signed certificate in production systems that are exposed to the Internet**

### Go to the private ip server and install jenkins, nginx:-

---------------------------------------------------------------------------------

Here, nginx will act as reverse proxy

**sudo apt-get update**

**sudo apt-get install nginx -y**

**sudo apt install openjdk-8-jre-headless -y**

**sudo wget -q -O - https://pkg.jenkins.io/debian-stable/jenkins.io.key | sudo apt-key add -**

**sudo apt-add-repository "deb http://pkg.jenkins-ci.org/debian binary/"**

**sudo apt update -y**

**sudo apt install jenkins -y**

**nginx -v (to check nginx version)**

Get a Certificate:-

-------------------------

**Next, you will need to purchase or create an SSL certificate. These commands are for a self-signed certificate, but you should get an officially signed certificate if you want to avoid browser warnings.**

**Move into the proper directory and generate a certificate.**

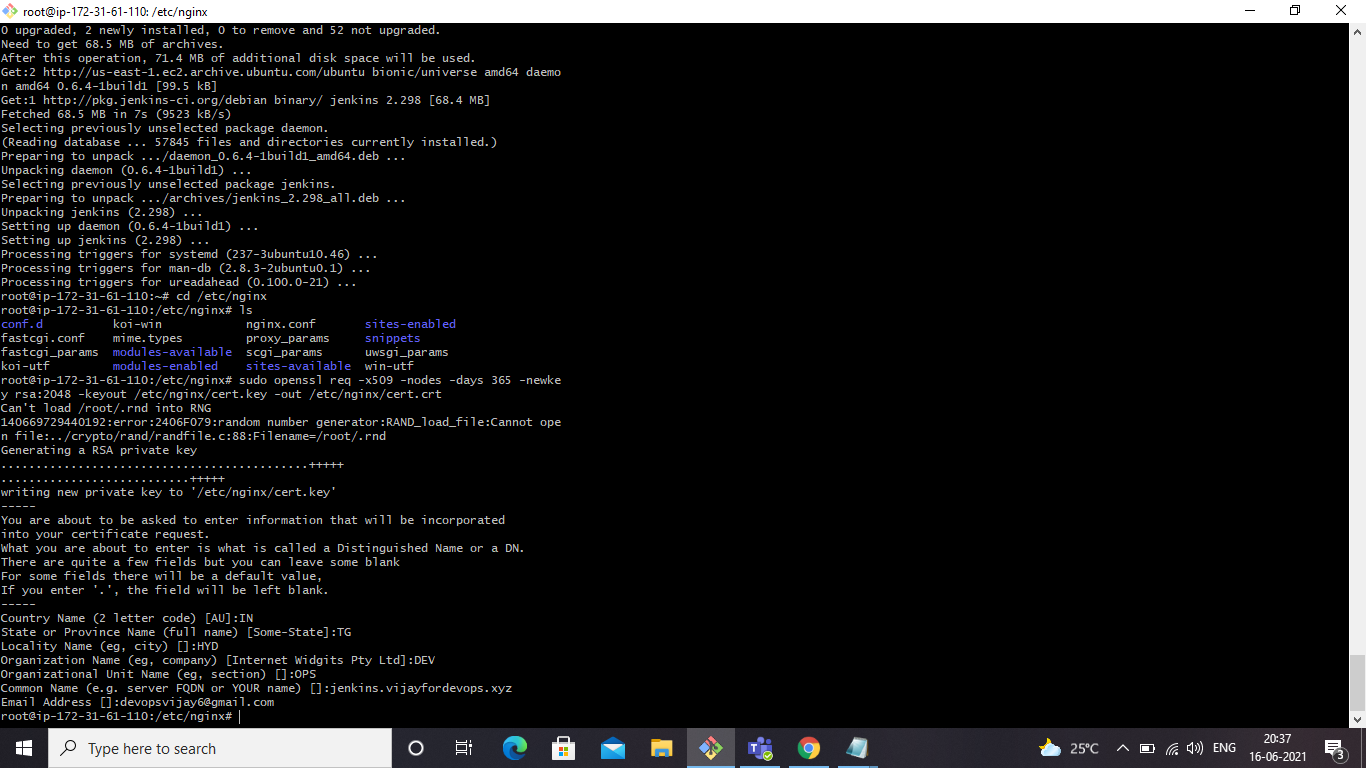
**Go to**

**cd /etc/nginx**

**sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/cert.key -out /etc/nginx/cert.crt**

**Enter the above command**

**You will be prompted to enter some information about the certificate. You can fill this out however you’d like; just be aware the information will be visible in the certificate properties. We’ve set the number of bits to 2048 since that’s the minimum needed to get it signed by a CA. If you want to get the certificate signed, you will need to create a CSR.**

****

**Next we need to edit the configuration file**

Edit the Configuration:-

-------------------------------

**To go**

**cd /etc/nginx/sites-enabled/**

**You will get default file, make changes in that file or create a new default file**

**mv default back.default**

**vi default**

**server {**

**listen 80;**

**return 301 https://$host$request\_uri;**

**}**

**server {**

**listen 443;**

**server\_name jenkins.vijayfordevops.xyz;**

**ssl\_certificate /etc/nginx/cert.crt;**

**ssl\_certificate\_key /etc/nginx/cert.key;**

**ssl on;**

**ssl\_session\_cache builtin:1000 shared:SSL:10m;**

**ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;**

**ssl\_ciphers HIGH:!aNULL:!eNULL:!EXPORT:!CAMELLIA:!DES:!MD5:!PSK:!RC4;**

**ssl\_prefer\_server\_ciphers on;**

**access\_log /var/log/nginx/jenkins.access.log;**

**location / {**

**proxy\_set\_header Host $host;**

**proxy\_set\_header X-Real-IP $remote\_addr;**

**proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;**

**proxy\_set\_header X-Forwarded-Proto $scheme;**

**# Fix the “It appears that your reverse proxy set up is broken" error.**

**proxy\_pass http://localhost:8080;**

**proxy\_read\_timeout 90;**

**proxy\_redirect http://localhost:8080 https://jenkins.vijayfordevops.xyz;**

**}**

**}**

**Paste the above configuration file and change your server name to required**

**Domain name.here my domain name is jenkins.vijayfordevops.xyz**

**In our configuration, the cert.crt and cert.key settings reflect the location where we created our SSL certificate. You will need to update the servername and `proxyredirect` lines with your own domain name. There is some additional Nginx magic going on as well that tells requests to be read by Nginx and rewritten on the response side to ensure the reverse proxy is working.**

**The first section tells the Nginx server to listen to any requests that come in on port 80 (default HTTP) and redirect them to HTTPS.**

Configure Jenkins:-

-------------------------

**For Jenkins to work with Nginx, we need to update the Jenkins config to listen only on the localhost interface instead of all (0.0.0.0), to ensure traffic gets handled properly. This is an important step because if Jenkins is still listening on all interfaces, then it will still potentially be accessible via its original port (8080). We will modify the /etc/default/jenkins configuration file to make these adjustments.**

**sudo vi /etc/default/jenkins**

**JENKINS\_ARGS="--webroot=/var/cache/jenkins/war --httpListenAddress=127.0.0.1 --httpPort=$HTTP\_PORT -ajp13Port=$AJP\_PORT"**

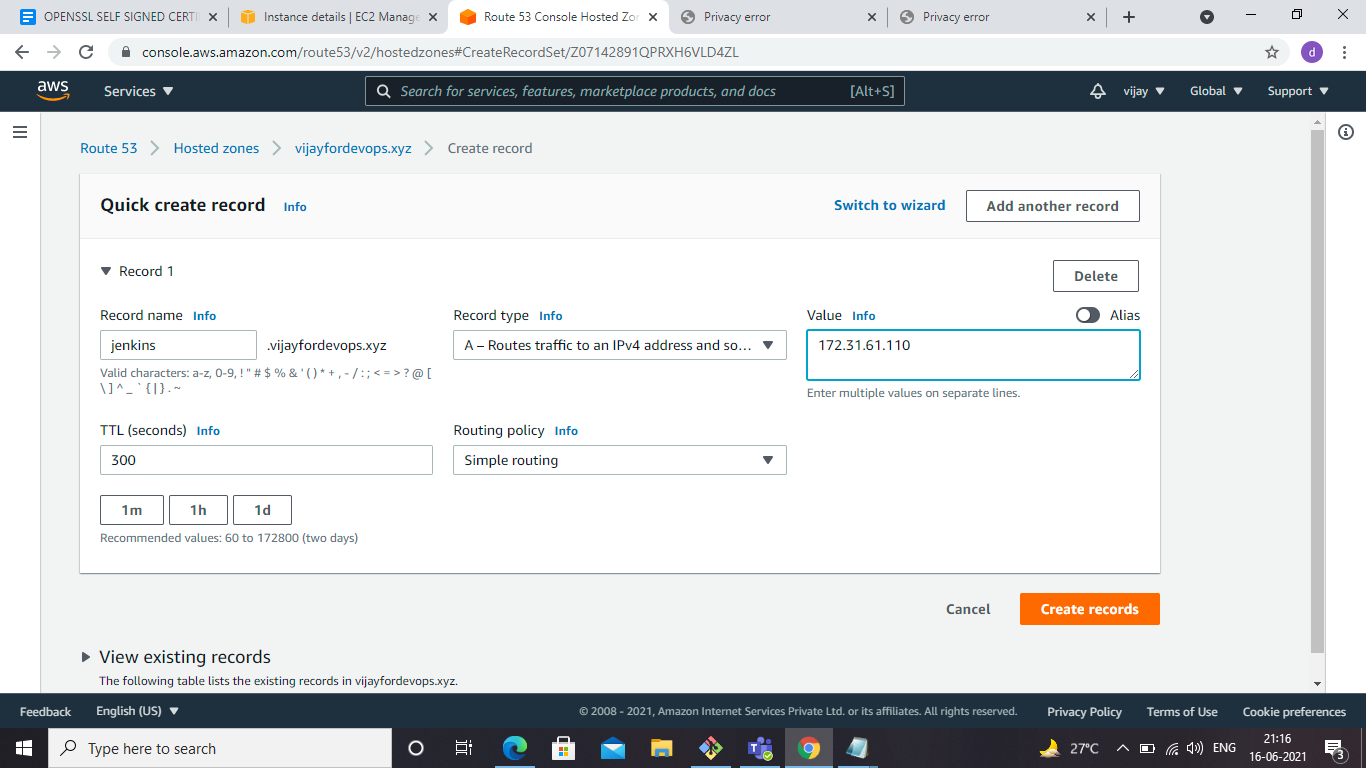
* **Next we need to create a record in route53**

ROUTE53:-

-----------------

**Go to route53 and create a hostedzone(public)**

* **Create a ‘A’ record in hosted zone on server private ip**

****

**Go back to server and type nslookup to check the ip address to domain name resolution after creating the record**

**nslookup jenkins.vijayfordevops.xyz**

**sudo service jenkins restart**

**sudo service nginx restart**

**Since we installed jenkins on private server, we need vpn to connect to private instances**

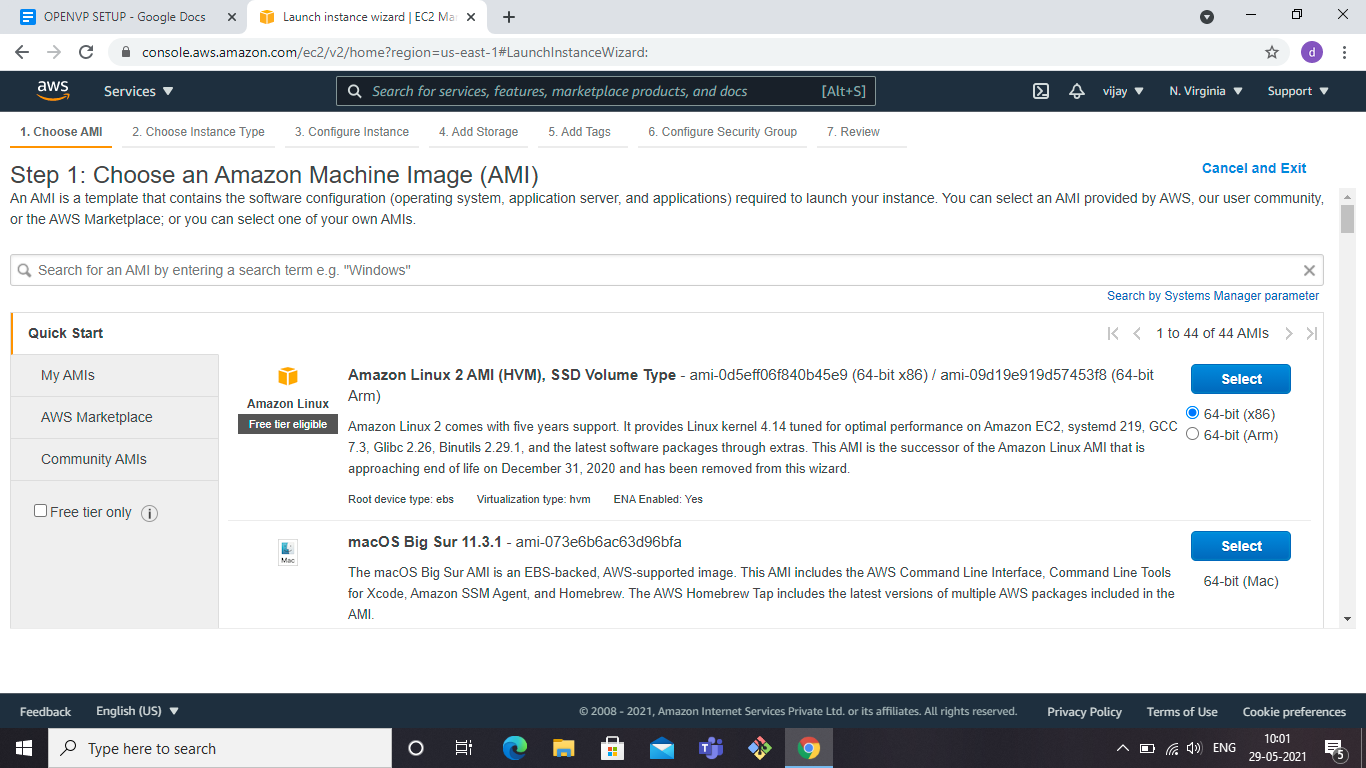
VPN:-

--------

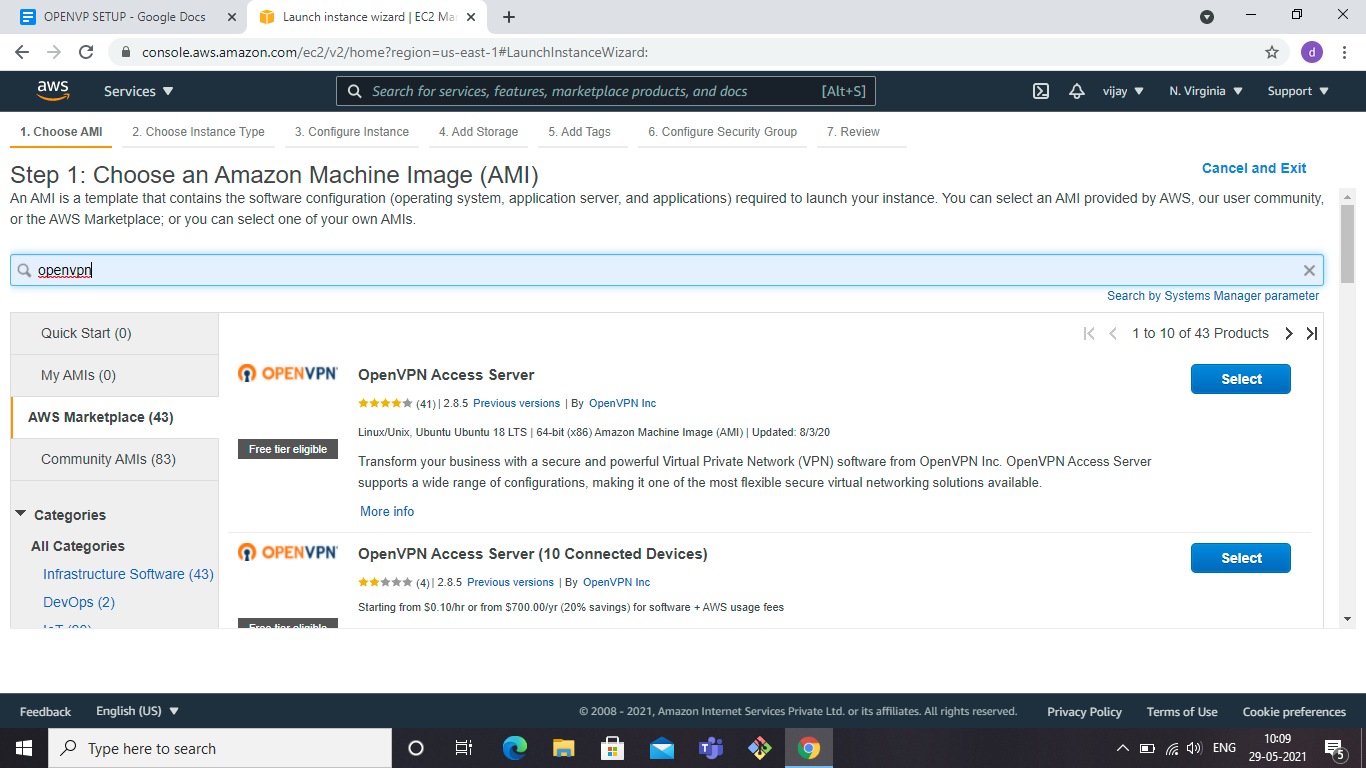
# Launch openvpn instance:-

**Go to services**

* **Ec2**
* **Launch instances**

****

* Select aws marketplace



* **Select free tier eligible openvpn**

**Step 1: Choose an Amazon Machine Image (AMI)**

* **continue**

**Step 2: Choose an Instance Type**

* **Choose t2.micro**

**Step 3: Configure Instance Details**

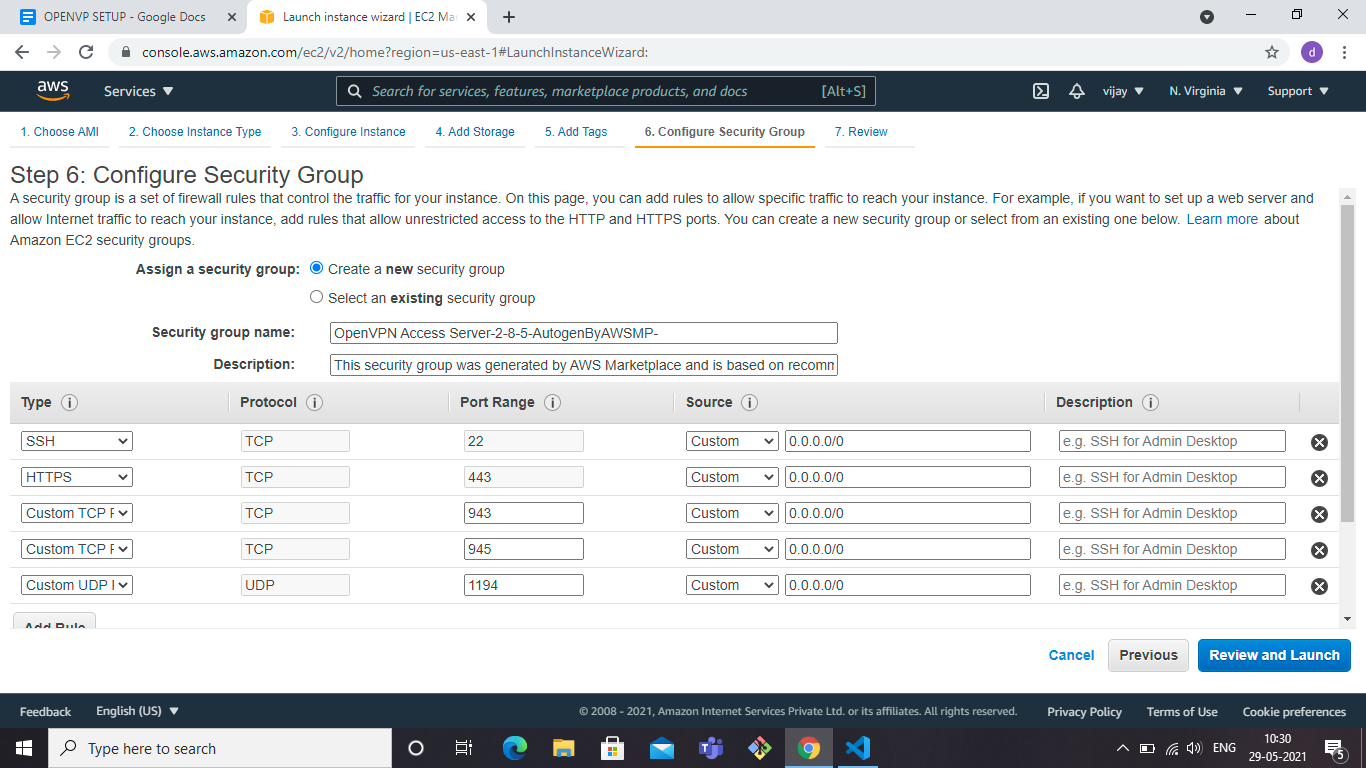
* **Select the desired vpc you want to keep openvpn, place in jenkins private instance vpc**
* **Select public subnet**
* **Enable auto-assign public ip**

**Step 4: Add Storage**

* **Default**

**Step 5: Add Tags**

**Step 6: Configure Security Group**

****

* **Keep the default security group got from openvpn**
* **Review and launch**

# Login to openvpn instance:-

**$ ssh -i "NV-key.pem"** [**root@ec2-3-239-19-139.compute-1.amazonaws.com**](mailto:root@ec2-3-239-19-139.compute-1.amazonaws.com)

**Press ENTER for the following:-**

**Please enter 'yes' to indicate your agreement [no]:**

**> Press ENTER for default [yes]:**

**> Press Enter for default [1]:**

**> Press ENTER for default [943]:**

**> Press ENTER for default [443]:**

**> Press ENTER for default [no]:**

**> Press ENTER for default [no]:**

**> Press ENTER for default [yes]:**

**> Press ENTER for EC2 default [yes]:**

**> Press ENTER for default [yes]:**

**> Please specify your Activation key (or leave blank to specify later):**

**During normal operation, OpenVPN AS can be accessed via these URLs:**

**Admin UI: https://3.239.19.139:943/admin**

**Client UI: https://3.239.19.139:943/**

* **Select client ui to download the openvpn client**
* **Before that you need to get the openvpn user password**
* **So login with openvpnas user**

**Please login as the user "openvpnas" rather than the user "root".**

**$ ssh -i "NV-key.pem"** [**openvpnas@ec2-3-239-19-139.compute-1.amazonaws.com**](mailto:openvpnas@ec2-3-239-19-139.compute-1.amazonaws.com)

**openvpnas@ip-10-0-1-192:~$ sudo passwd openvpn**

**Enter new UNIX password:**

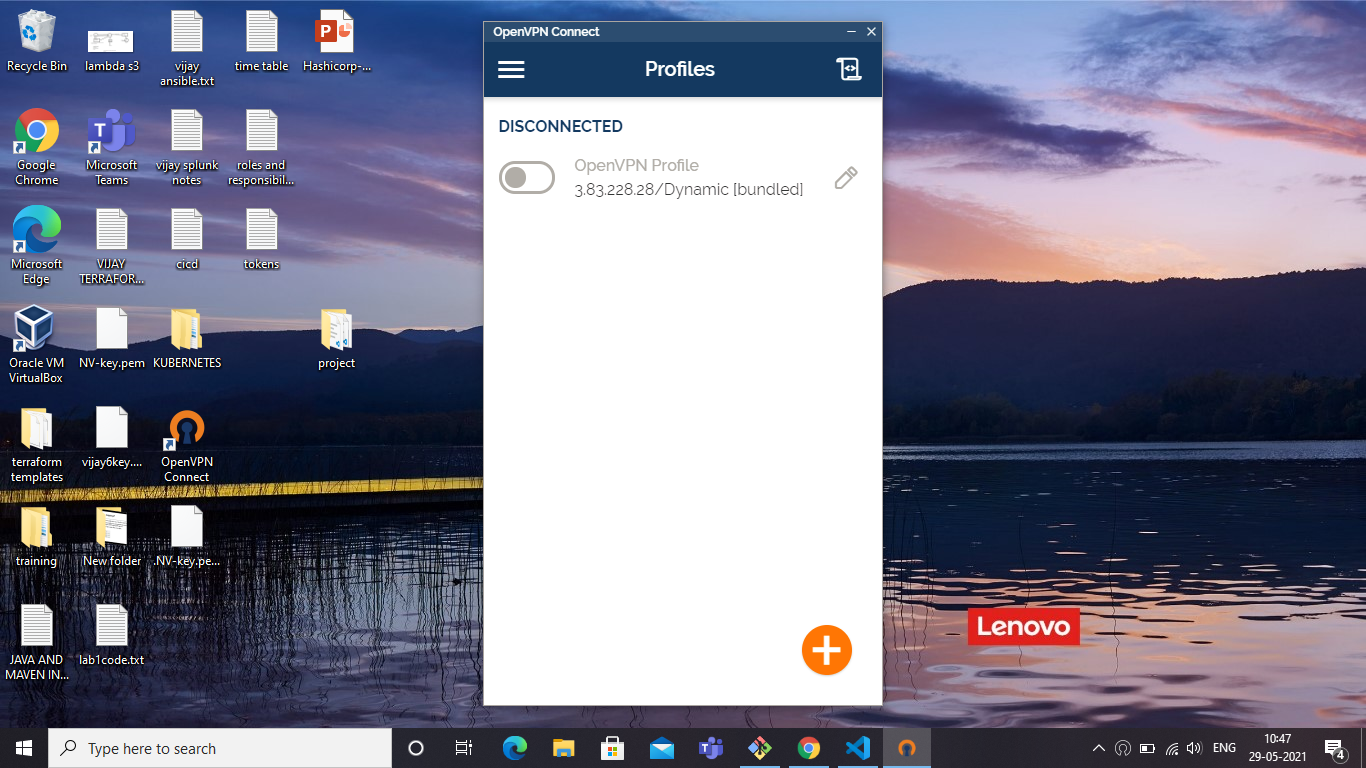
**Retype new UNIX password:**

**passwd: password updated successfully**

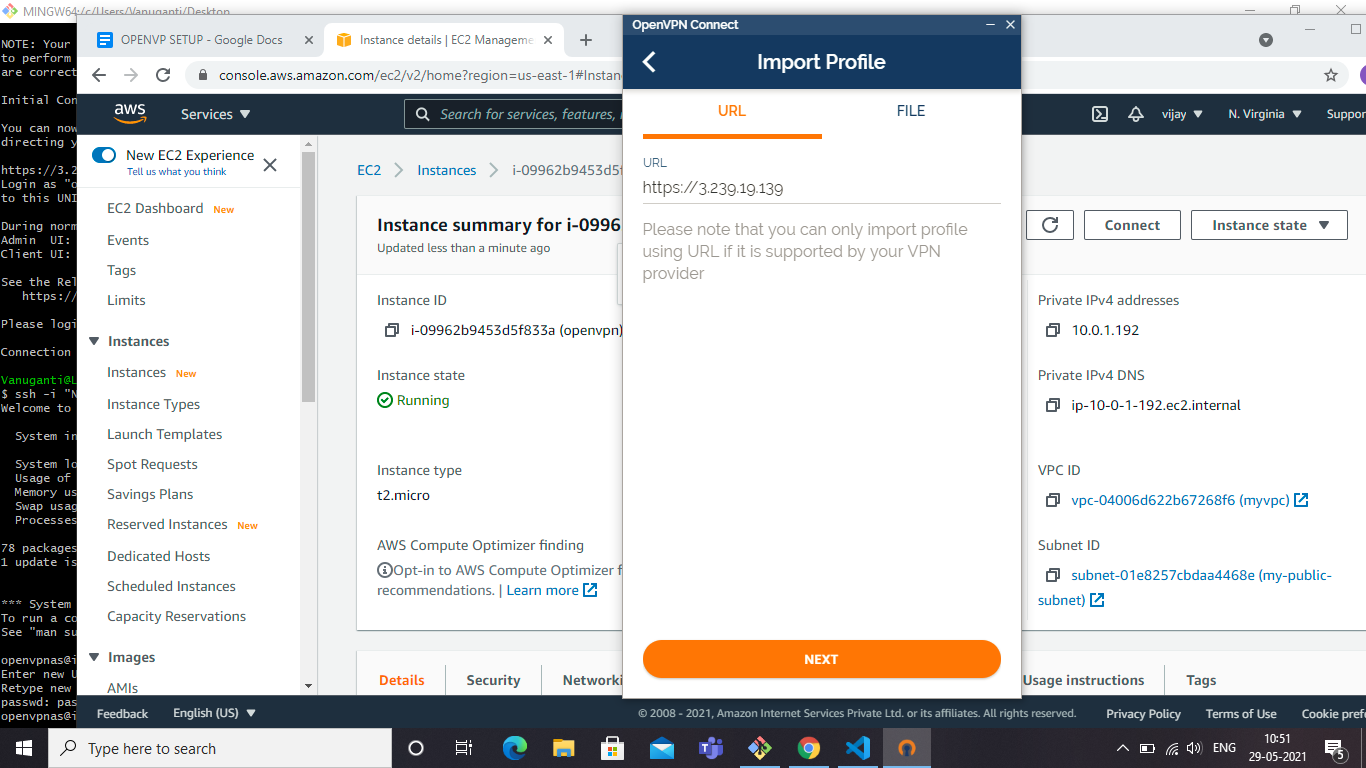
**Client UI:** [**https://3.239.19.139:943/**](https://3.239.19.139:943/)

* **Now login to url with**
* **Openvpn username**
* **Password**
* **Download the client depending upon the operating system**

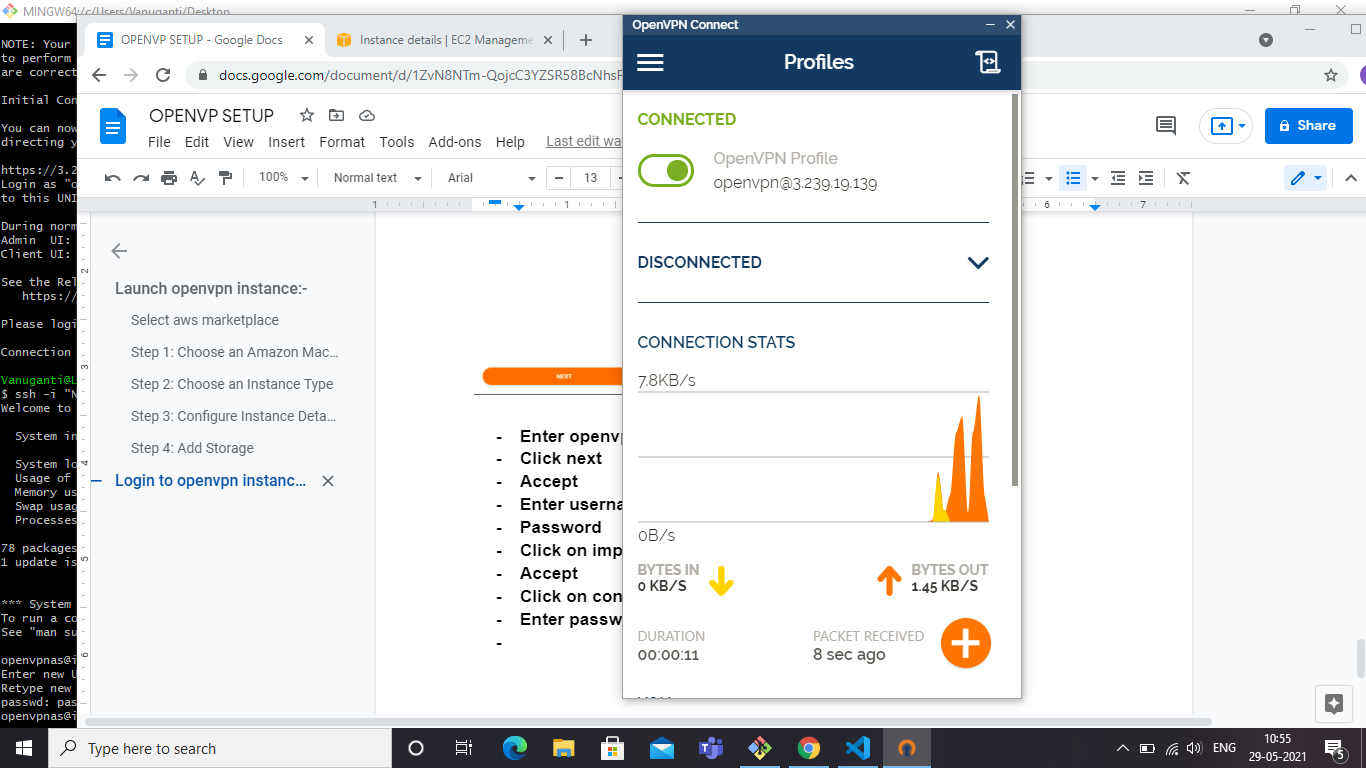
**Once you download, open the openvpn client from your local laptop**

****

* **Click on plus symbol to start new connection**

****

* **Enter openvpn server public ip url**
* **Click next**
* **Accept**
* **Enter username**
* **Password**
* **Click on import**
* **Accept**
* **Click on connect**
* **Enter password**

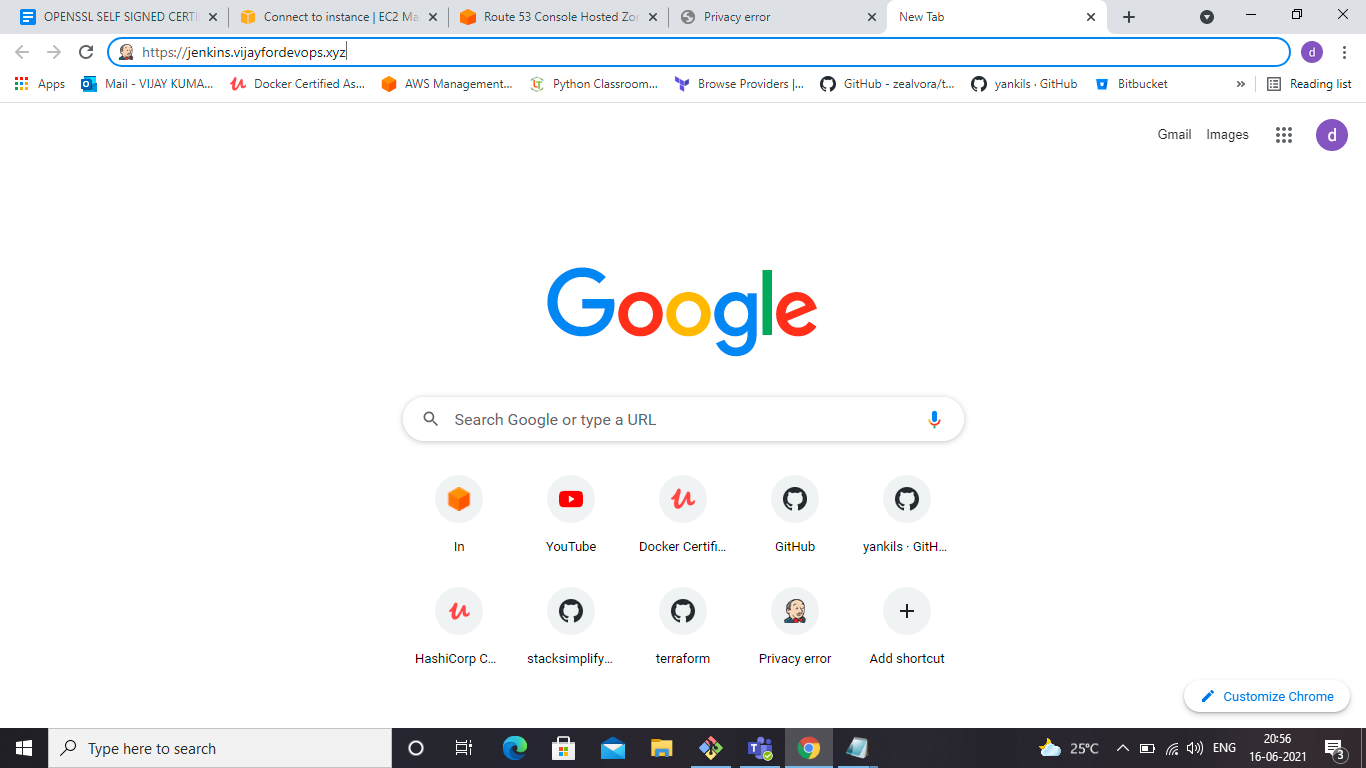
****

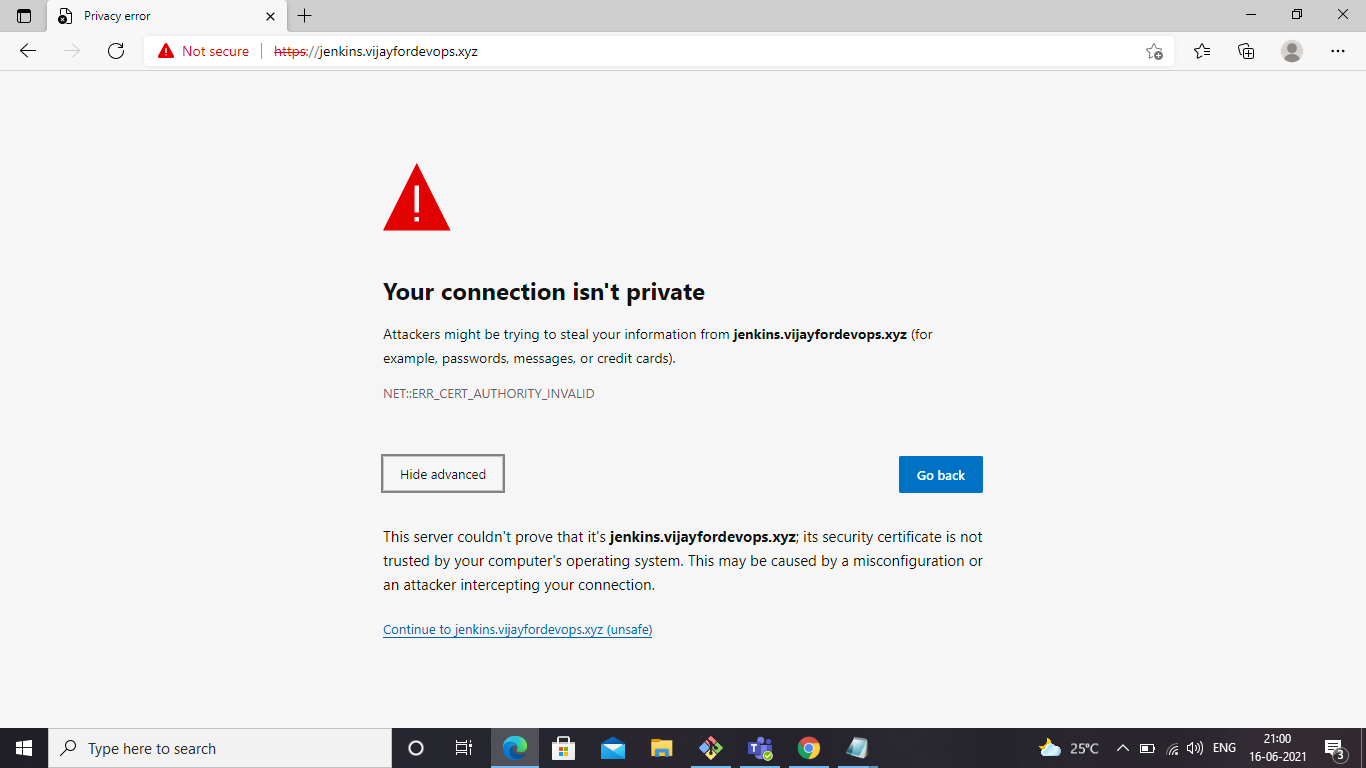
* **Vpn will be connected to vpn server**

**goto**

**internet explorer browser and type**

**https://jenkins.vijayfordevops.xyz**

****

****

