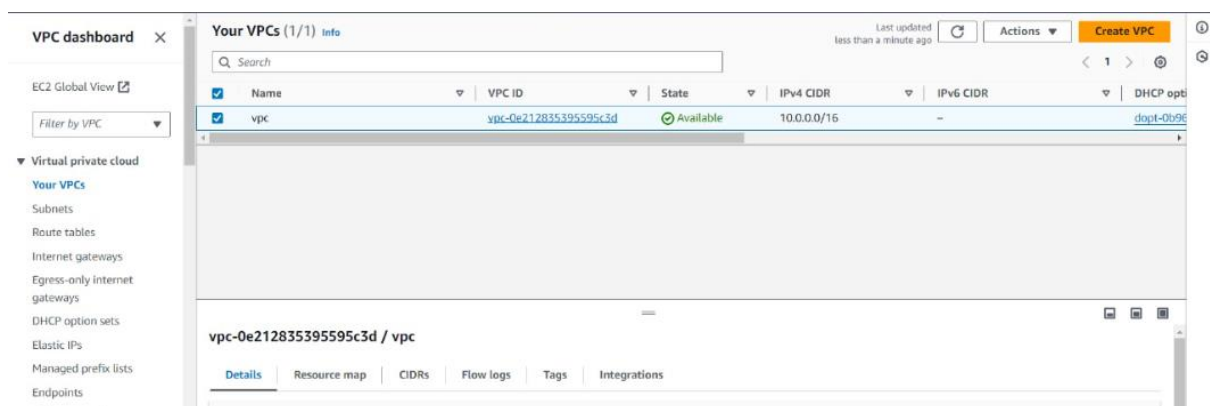
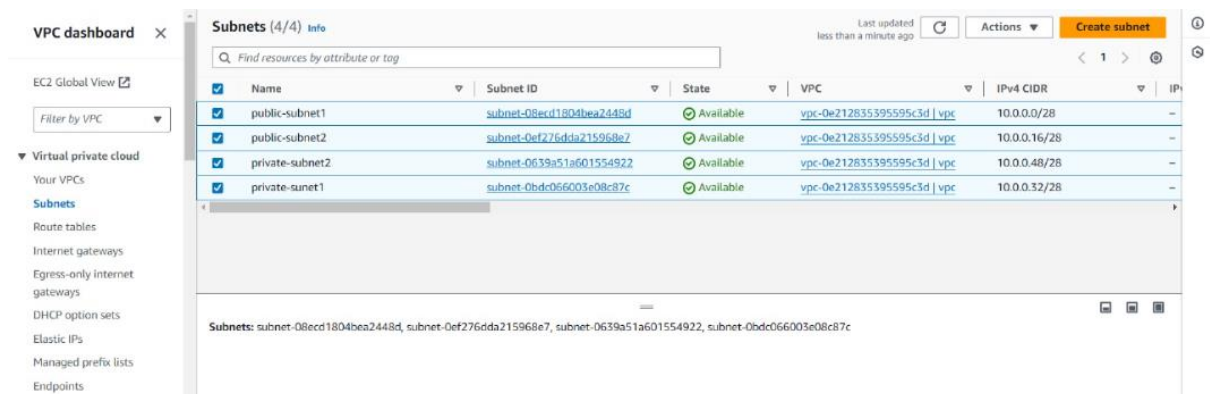


DESIGNING A SECURE NETWORK ARCHITECTURE USING VPC

As the project is mostly done in the AWS console I'm providing the screenshots of the steps we followed and here and there we used some commands so I'll mention them at that particular step.



Creating a VPC after configuration



Creating the subnets and connecting them to VPC and configuring them

VPC dashboard ×

EC2 Global View [↗](#)

Filter by VPC ▾

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways**
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs

Internet gateways (1/1) Info

Search

Actions ▾ Create internet gateway

Name	Internet gateway ID	State	VPC ID	Owner
igw	igw-0ed319fbc57b9049e	Attached	vpc-0e212835395595c3d vpc	211125521097

igw-0ed319fbc57b9049e / igw

VPC dashboard ×

EC2 Global View [↗](#)

Filter by VPC ▾

Virtual private cloud

- Your VPCs
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways**
- DHCP option sets
- Elastic IPs

NAT gateways (1/1) Info

Find resources by attribute or tag

Actions ▾ Create NAT gateway

Name	NAT gateway ID	Connectivity...	State	State message	Primary public I...	Primary private
nat gateway	nat-08617381f07a00150	Public	Available	-	13.53.64.90	10.0.0.8

nat-08617381f07a00150 / nat gateway

We used a IGW for public route table and NAT gateway for public route table

VPC dashboard ×

EC2 Global View [↗](#)

Filter by VPC ▾

Virtual private cloud

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists

Route tables (2/3) Info

Find resources by attribute or tag

Last updated less than a minute ago

Actions ▾ Create route table

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0c89500db2f8e7de1	-	-	Yes	vpc-0e212835395595c3d v
public route	rtb-0c9d61c1bd9b5ea4a	2 subnets	-	No	vpc-0e212835395595c3d v
private route	rtb-0e8e783f2ee204623	2 subnets	-	No	vpc-0e212835395595c3d v

Route tables: rtb-0e8e783f2ee204623, rtb-0c9d61c1bd9b5ea4a

EC2 Dashboard ×

EC2 Global View [↗](#)

Events

Instances

- Instances**
- Instance Types
- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running X Clear filters

Connect Instance state ▾ Actions ▾ Launch instances ▾

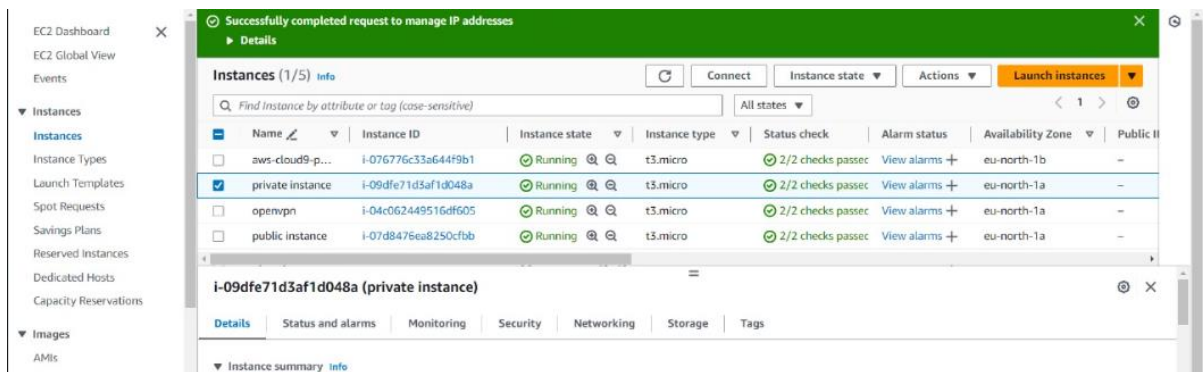
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
aws-cloud9-p...	i-076776c33a644f9b1	Running	t3.micro	2/2 checks passed	View alarms +	eu-north-1b	-

i-076776c33a644f9b1 (aws-cloud9-public-af2d22a913e148859d69edb555291a97)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary Info

Launched a AWS cloud9 environment



Launched a public ec2 ubuntu instance and a private ec2 ubuntu instance and finally an instance named OpenVPN by using IAM.

```
Preparing to unpack .../nginx_1.24.0-2ubuntu7_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7) ...
Setting up nginx (1.24.0-2ubuntu7) ...
Setting up nginx-common (1.24.0-2ubuntu7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

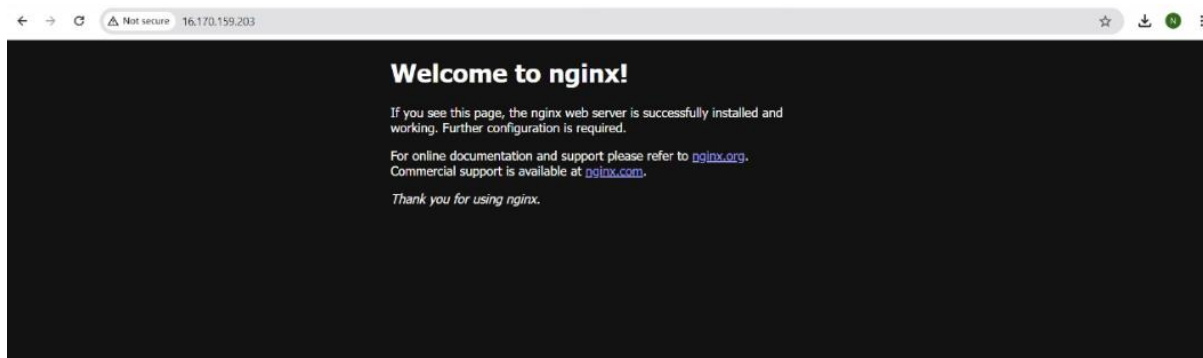
No user sessions are running outdated binaries.

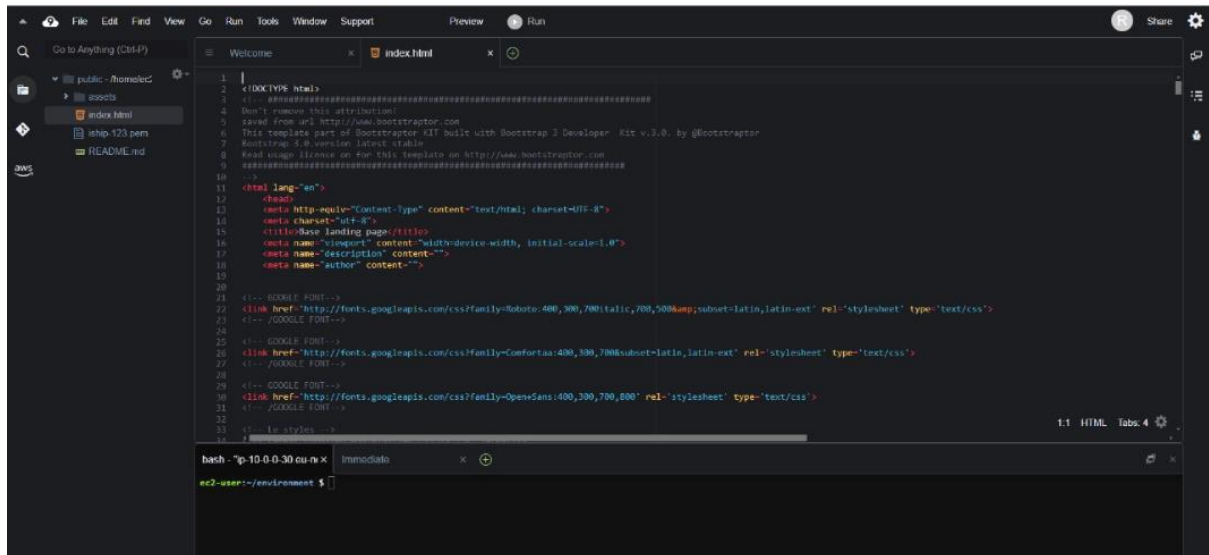
No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-10-0-0-4:/home/ubuntu# chmod 777 /var/www/html
root@ip-10-0-0-4:/home/ubuntu# ls
root@ip-10-0-0-4:/home/ubuntu# cd ..
root@ip-10-0-0-4:/home# cd ..
root@ip-10-0-0-4:/# # cd /var/www/html
root@ip-10-0-0-4:/var/www/html# ls
index.nginx-debian.html
root@ip-10-0-0-4:/var/www/html#
```

i-07d8476ea8250cfbb (public instance)
PublicIPs: 16.170.159.203 PrivateIPs: 10.0.0.4

Connecting the public instance to ec2 and given the following commands

1. sudo su
2. apt-get update
3. apt-get install nginx -y
4. chmod 777 /var/www/html
5. cd /var/www/html
6. ls





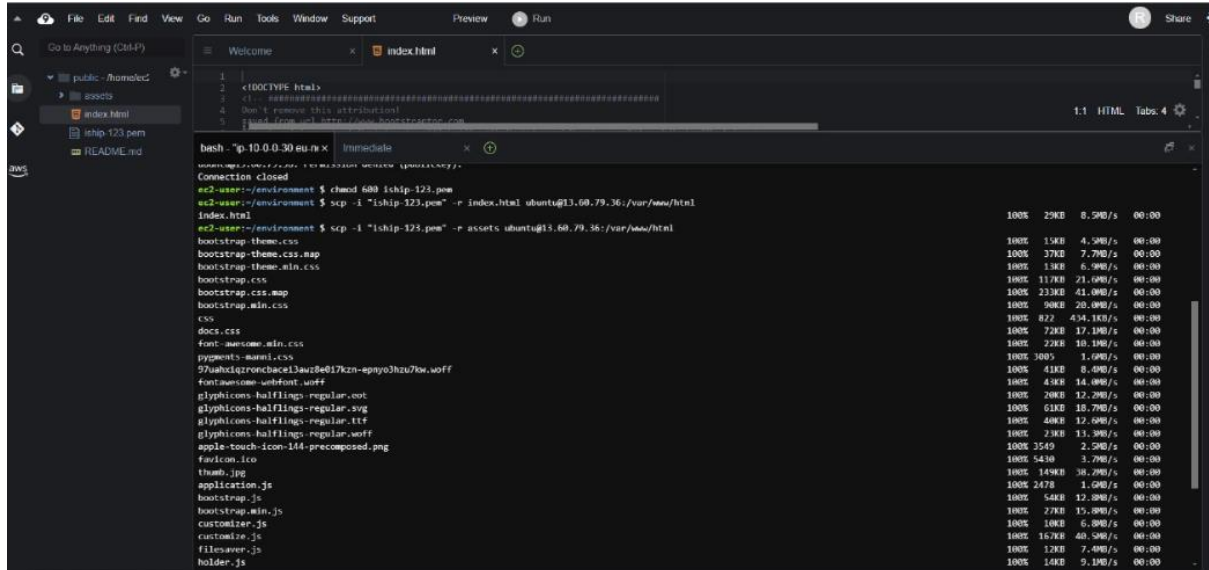
Uploading the sample template and the pem file to the cloud9 environment.

Then given the required commands

1. `chmod 600 name.pem`

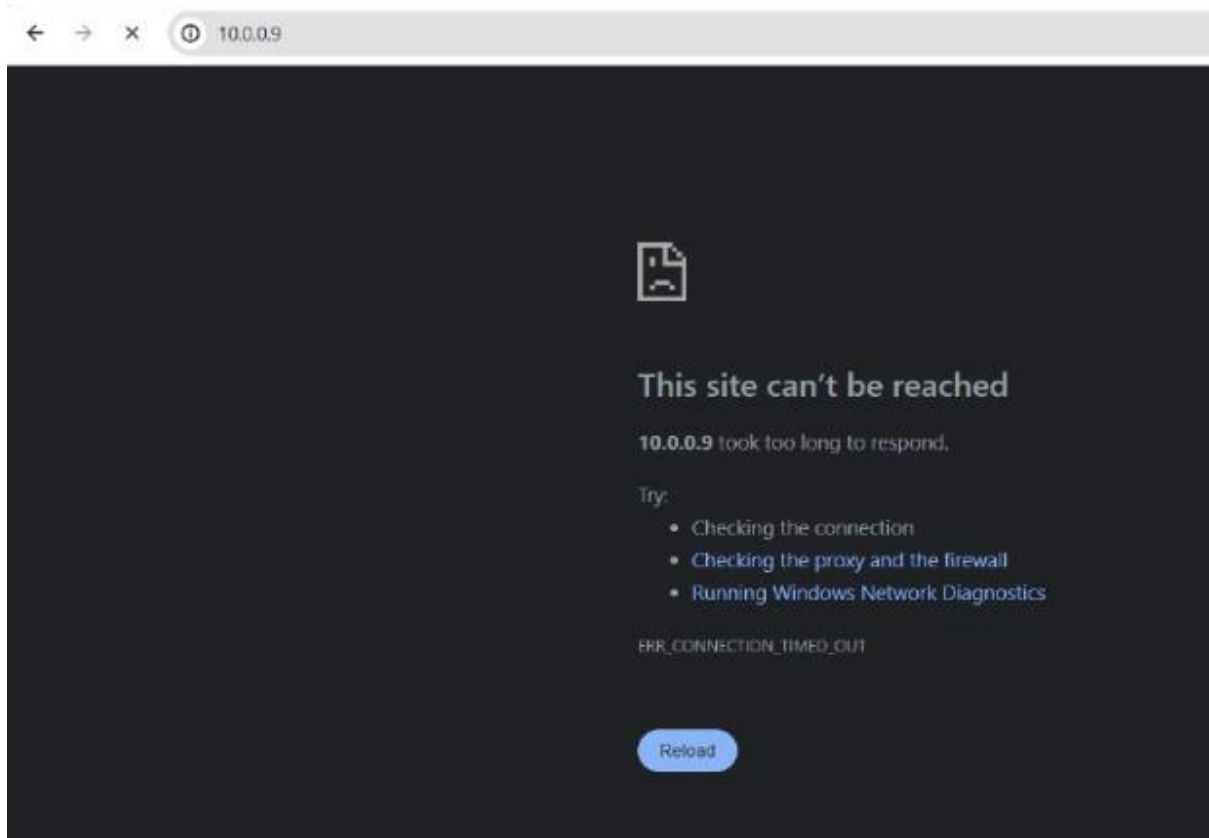
2. `scp -i "name.pem" -r index.html source-path ec2-user@IP:/destination-path`

3. `scp -i "name.pem" -r assets source-path ec2-user@IP:/destination-path`

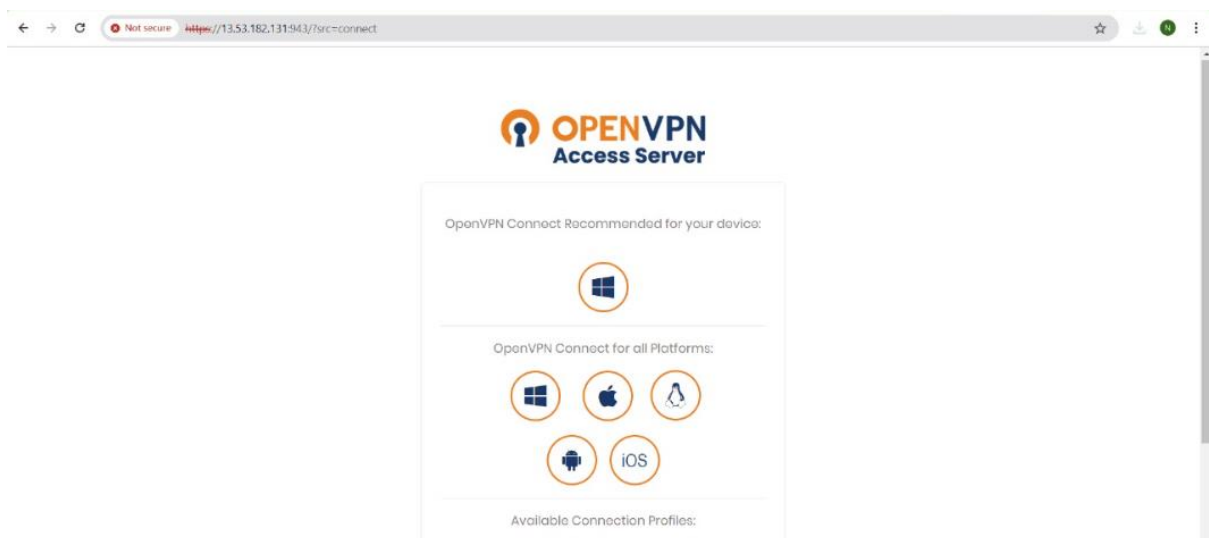


Now the private instance is connected to ec2 and removed the public IP.

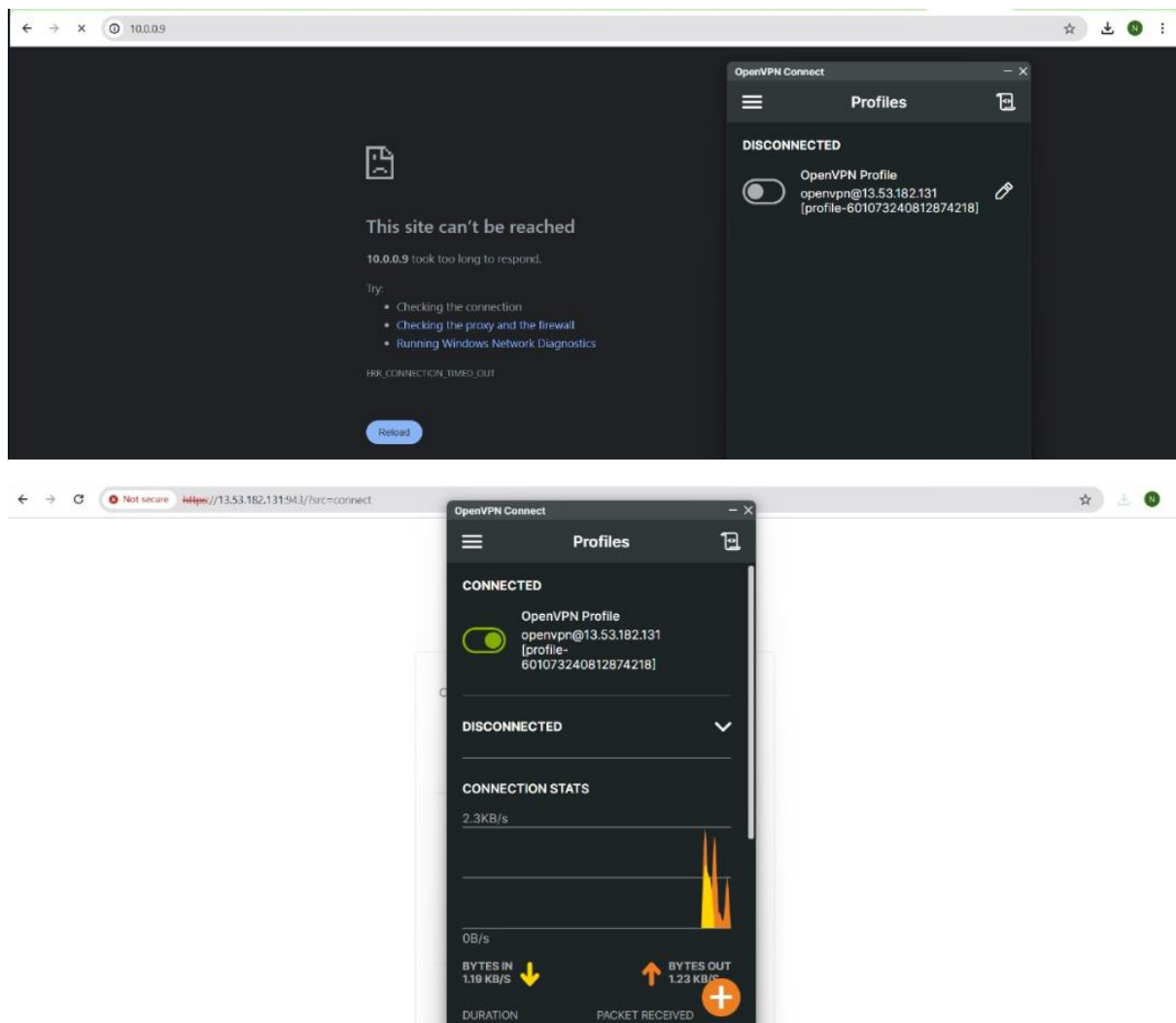
Now taken the private IP and passed it in the new tab



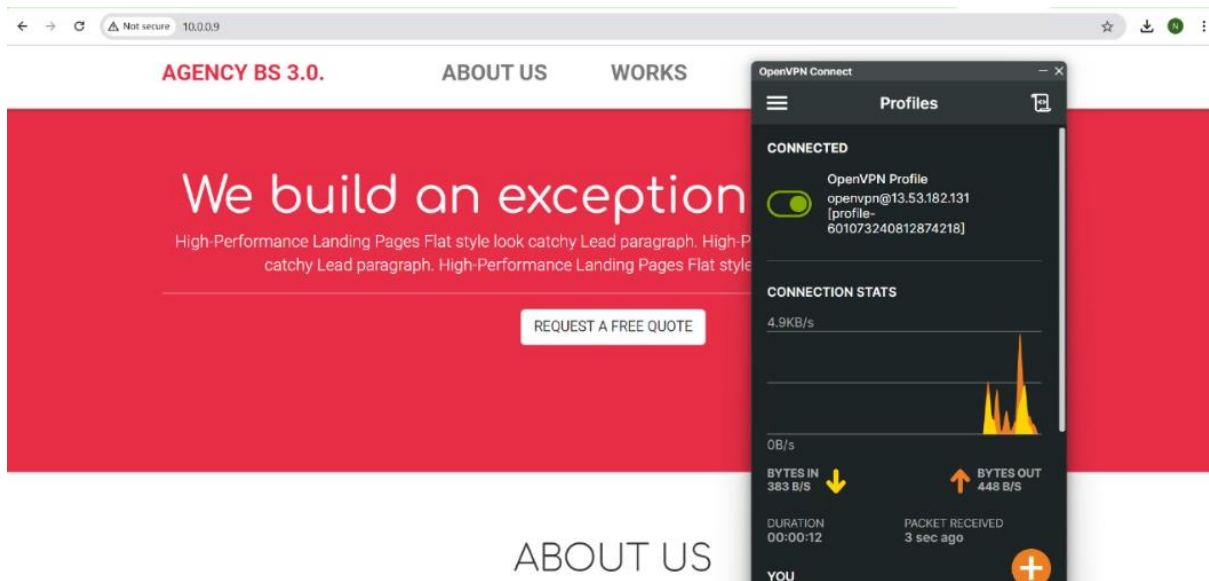
Now download and configure the OpenVPN which is a third party access control.



We can't still access so let's connect with the OpenVPN



Now we can access the sample website using private IP and pasting it in the new tab



ABOUT US

Here with this we designed a secure network architecture using VPC and third party access control OpenVPN.

The basic architecture design is the below one

