AAW

AWS ADVANCED PROJECT-1

**Deploying a Highly Available Web Application and Bastion Host in AWS**

ARAVIND G KUMAR

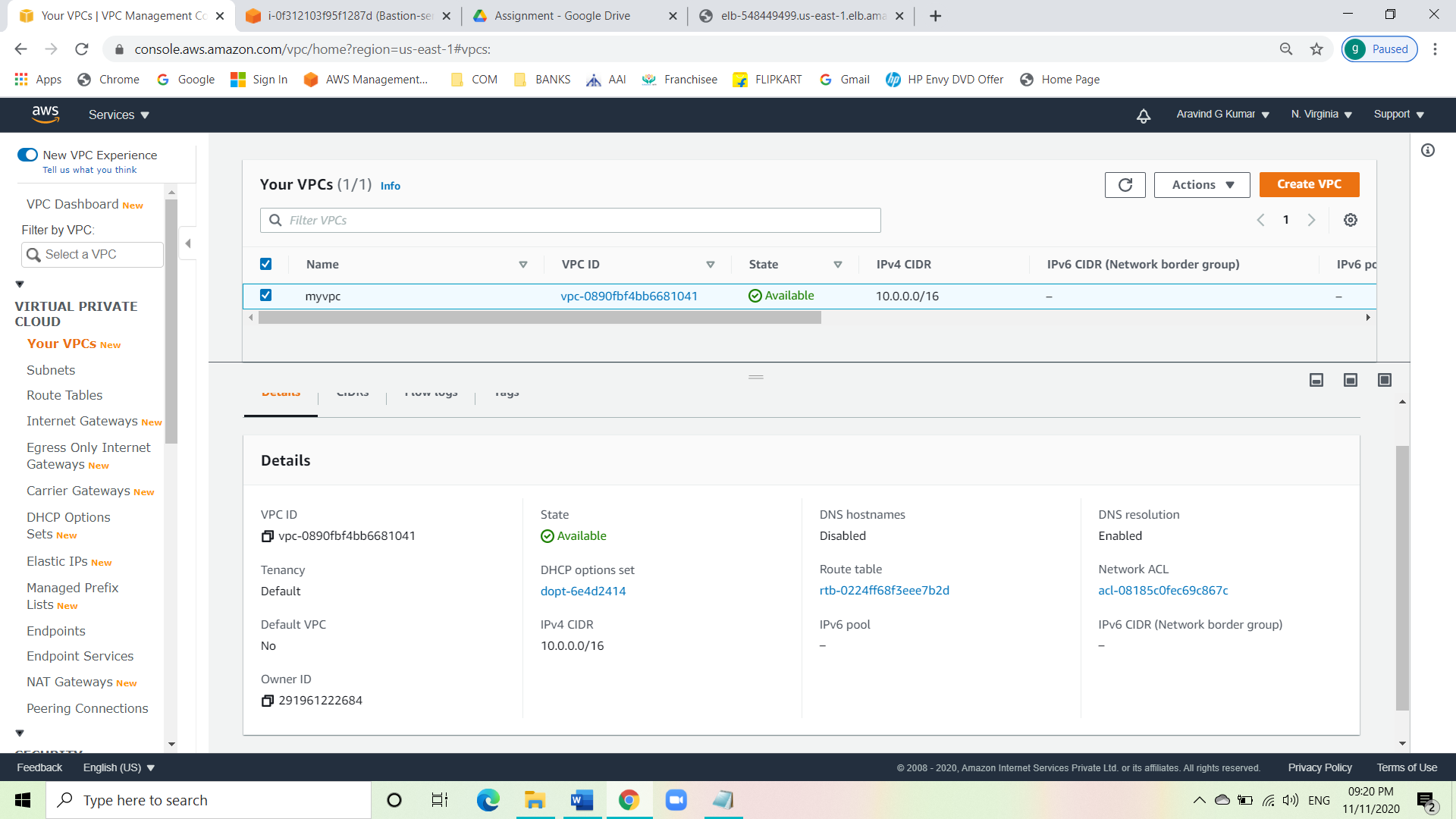
**Introduction:**

**A bastion host** is a system that is exposed to the internet. In terms of security, Bastion is the only server that is exposed to the internet and should be highly protective to malicious attacks.  A Bastion host is also known as a Jump Box. It is a computer that acts like a proxy server and that allows the client machine to connect to the remote server. It usually resides outside the firewall. The Bastion server filters the incoming traffic and prevents unwanted connections entering the network thus acting as a gateway to maintain the security of bastion hosts, all unnecessary software, daemons.

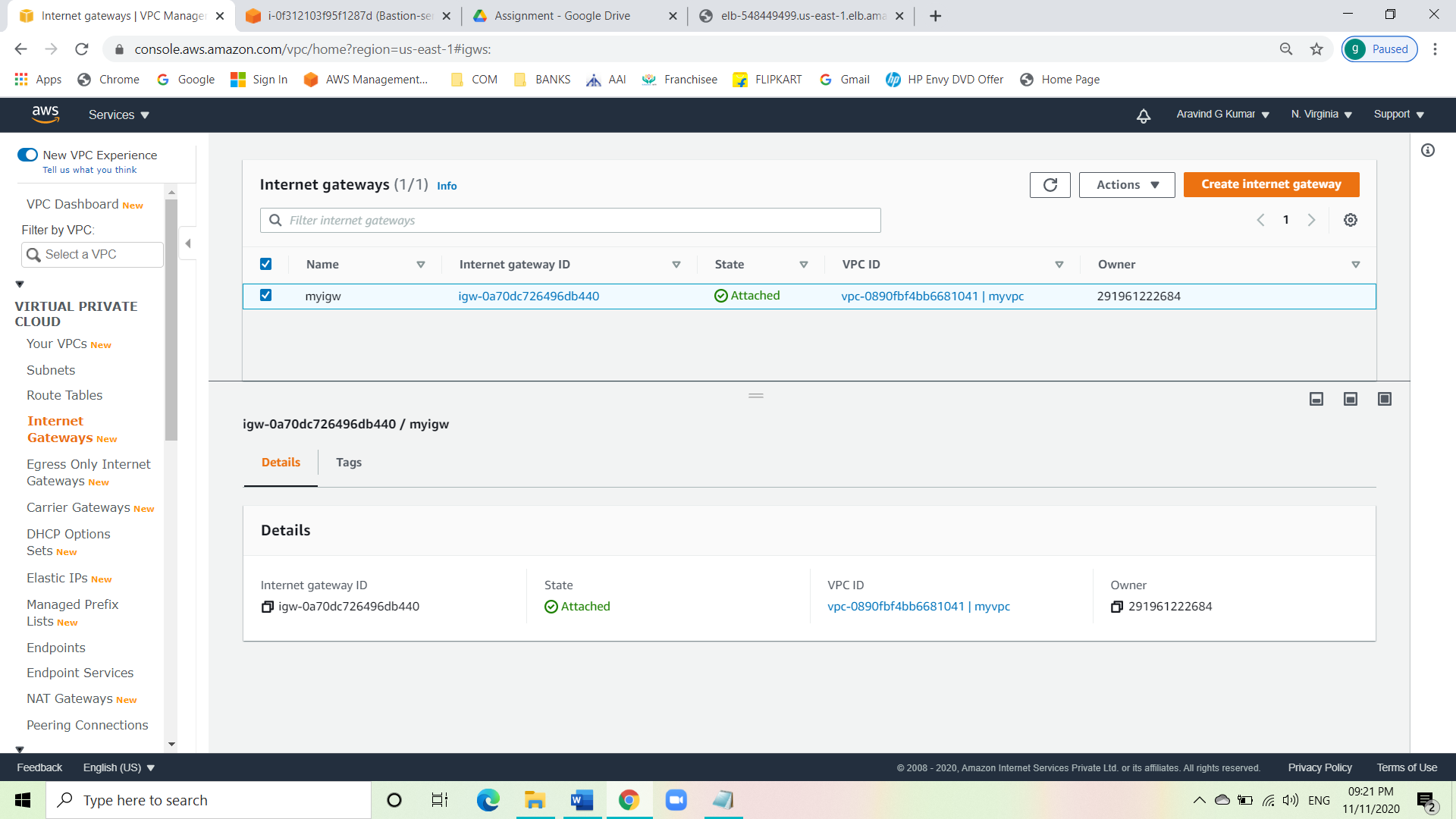
**High Availability:** Consider your application is running on a single EC2 instance. If the traffic to your application increases and you need further resources, we can launch multiple EC2 instances from an already running server and then use Elastic Load Balancing to distribute the traffic to your application among the newly-created servers. We can also eliminate the Fault tolerance in your application by placing the servers (EC2 instances) across different availability zones. In the event of Failure of one Availability zone, your application will serve or handle the traffic from another availability zone. High Availability and fault tolerance can be achieved using Elastic Load balancers.

In order to create a highly available web application and a bastion host follow the steps which is given below.

* Create a VPC for the servers and the ELB to be launched.

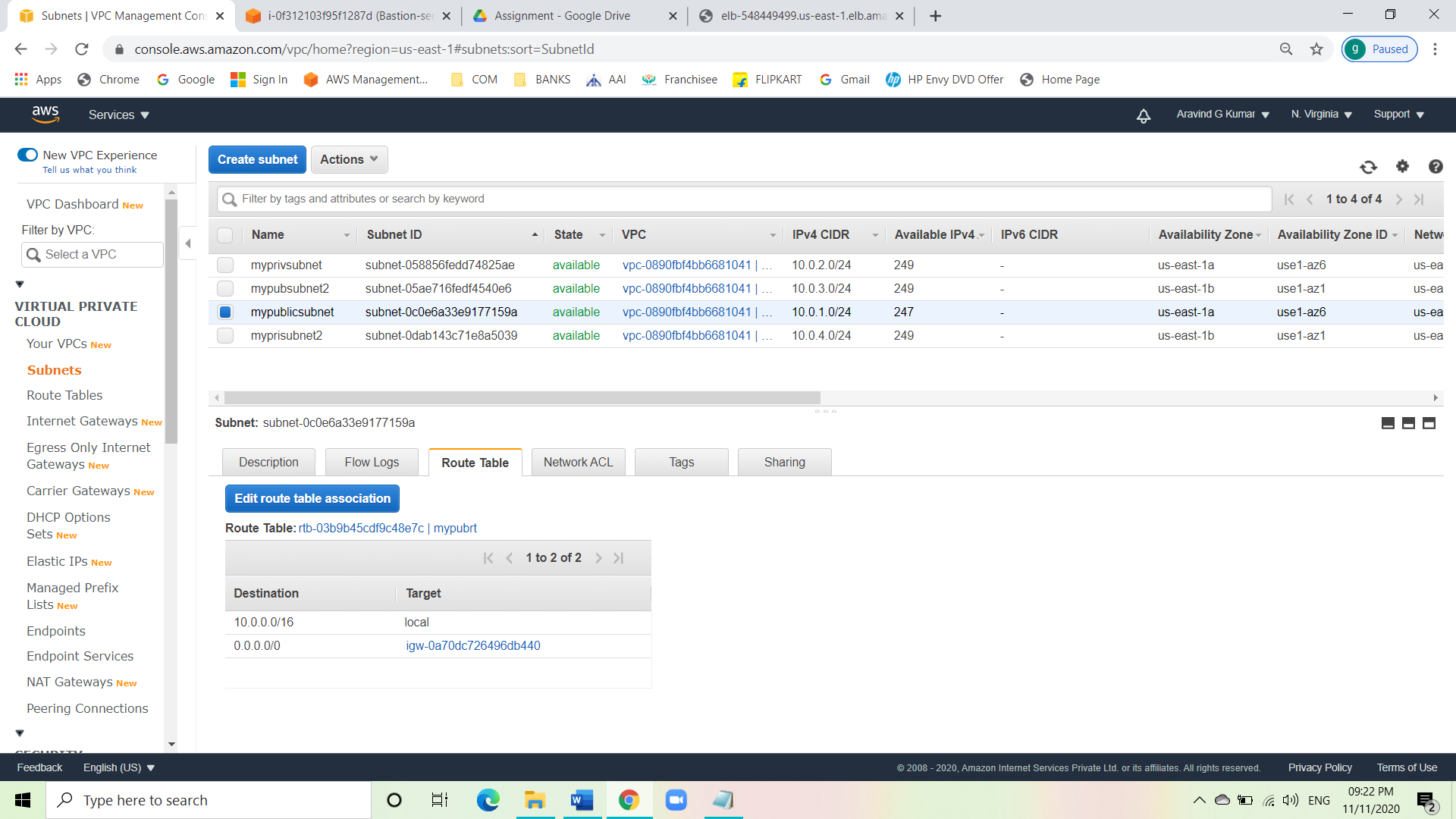


* Create an IGW and attach it to the VPC.

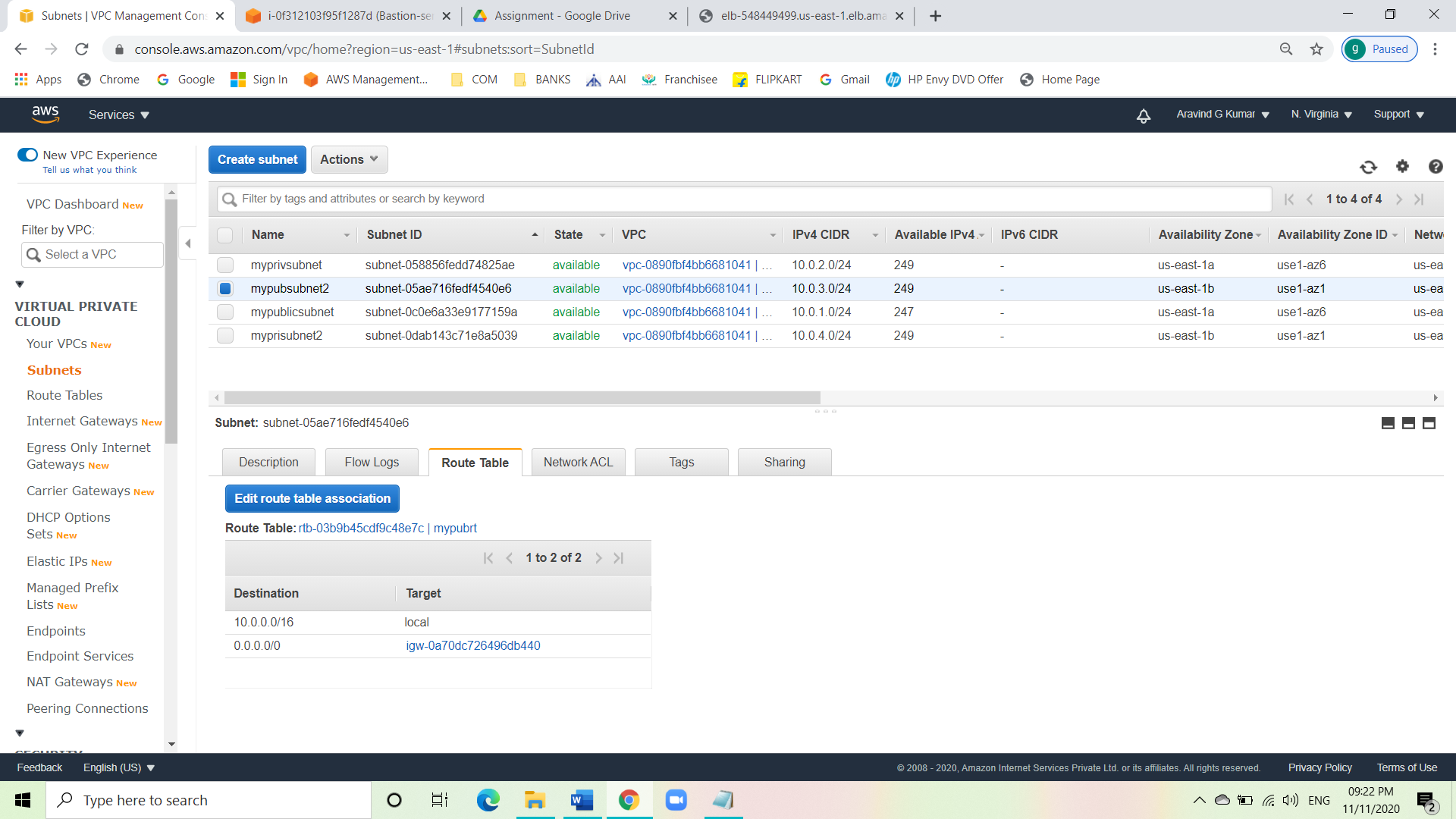


* Create 4 subnets.
* Create public and private subnets in the same availability zone, because we need the subnets to have public IPv4 for the load balancer to connect to the internet.

Public subnet 1



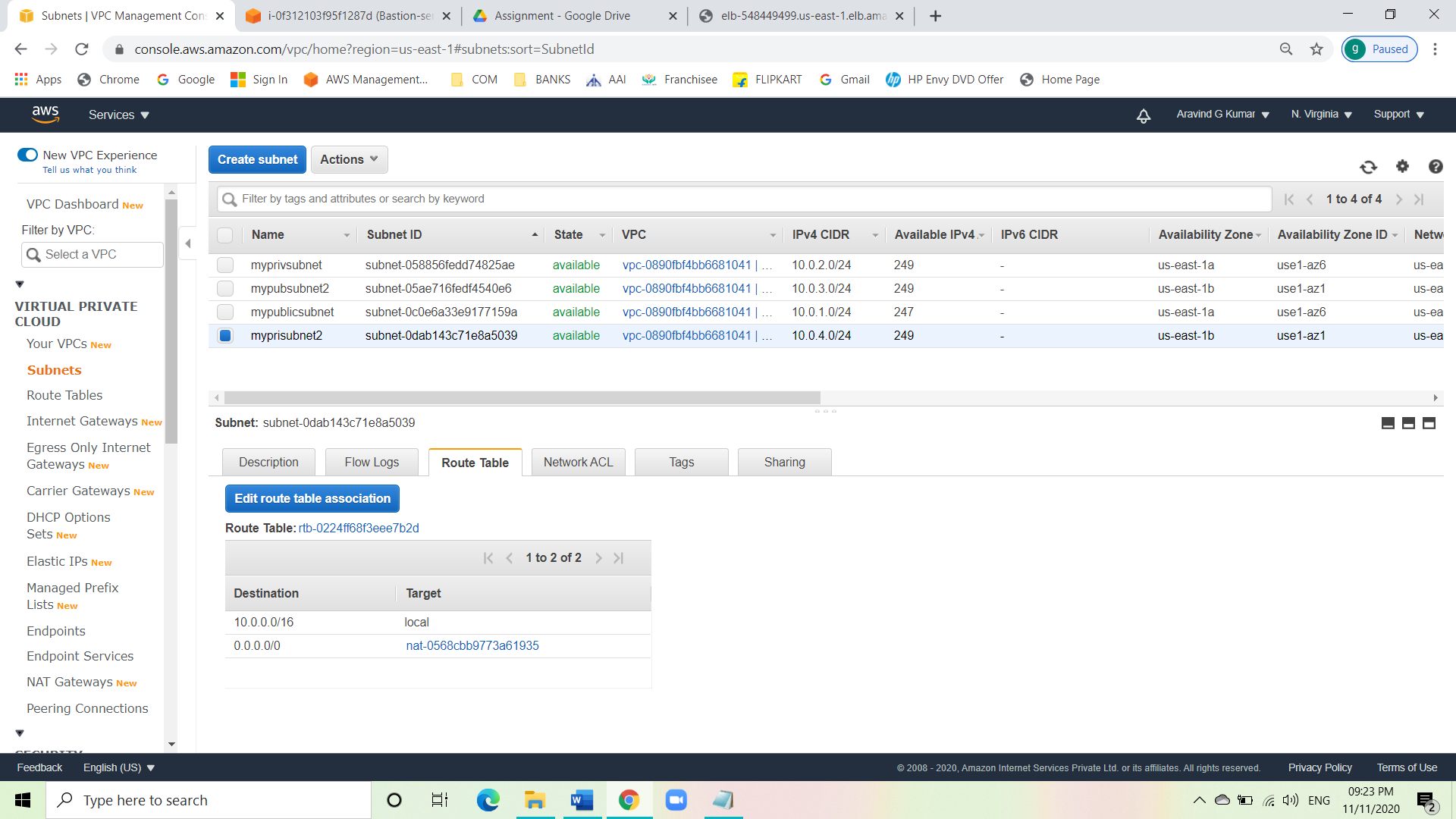
Public subnet 2



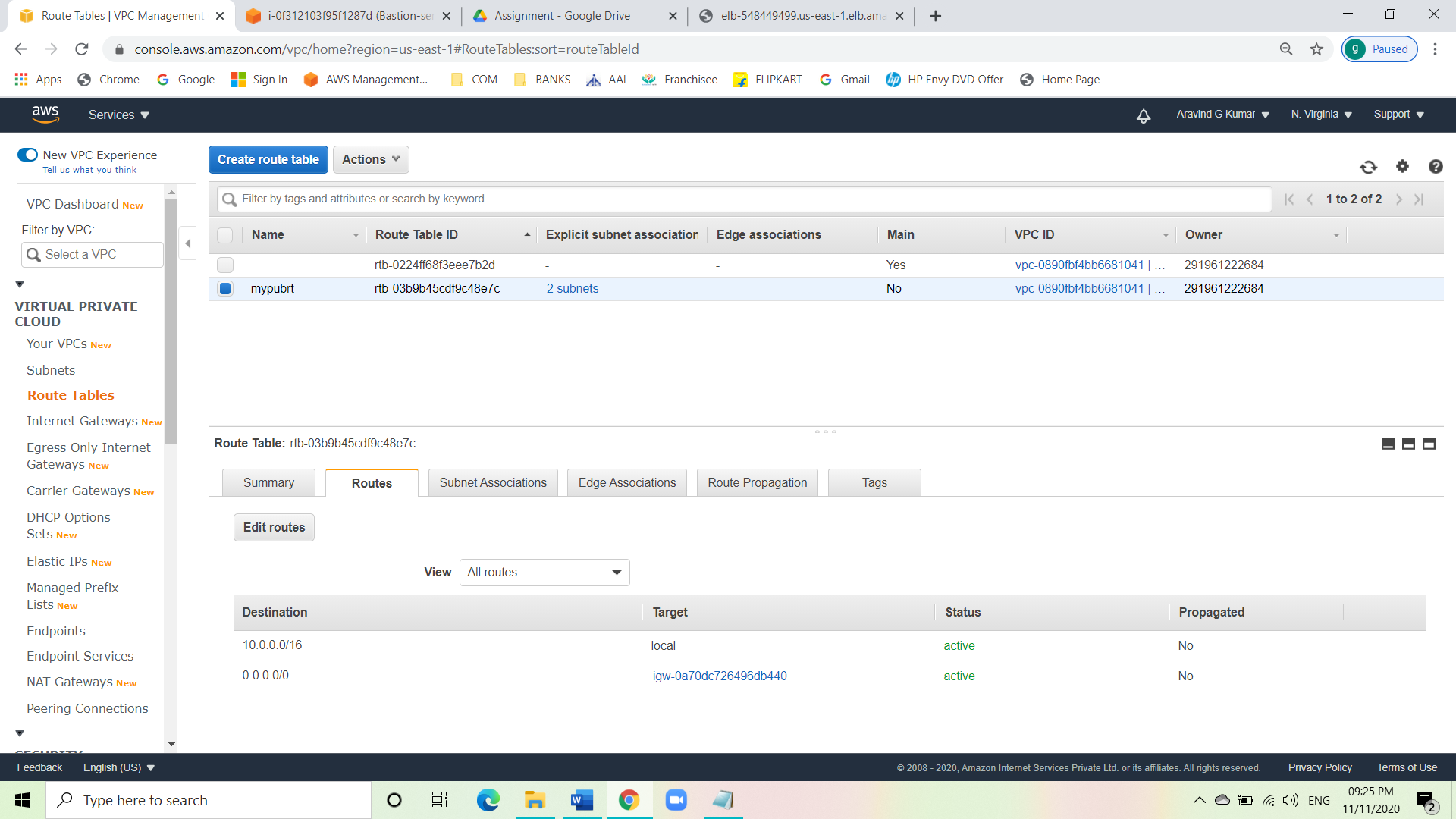
Private subnet 1



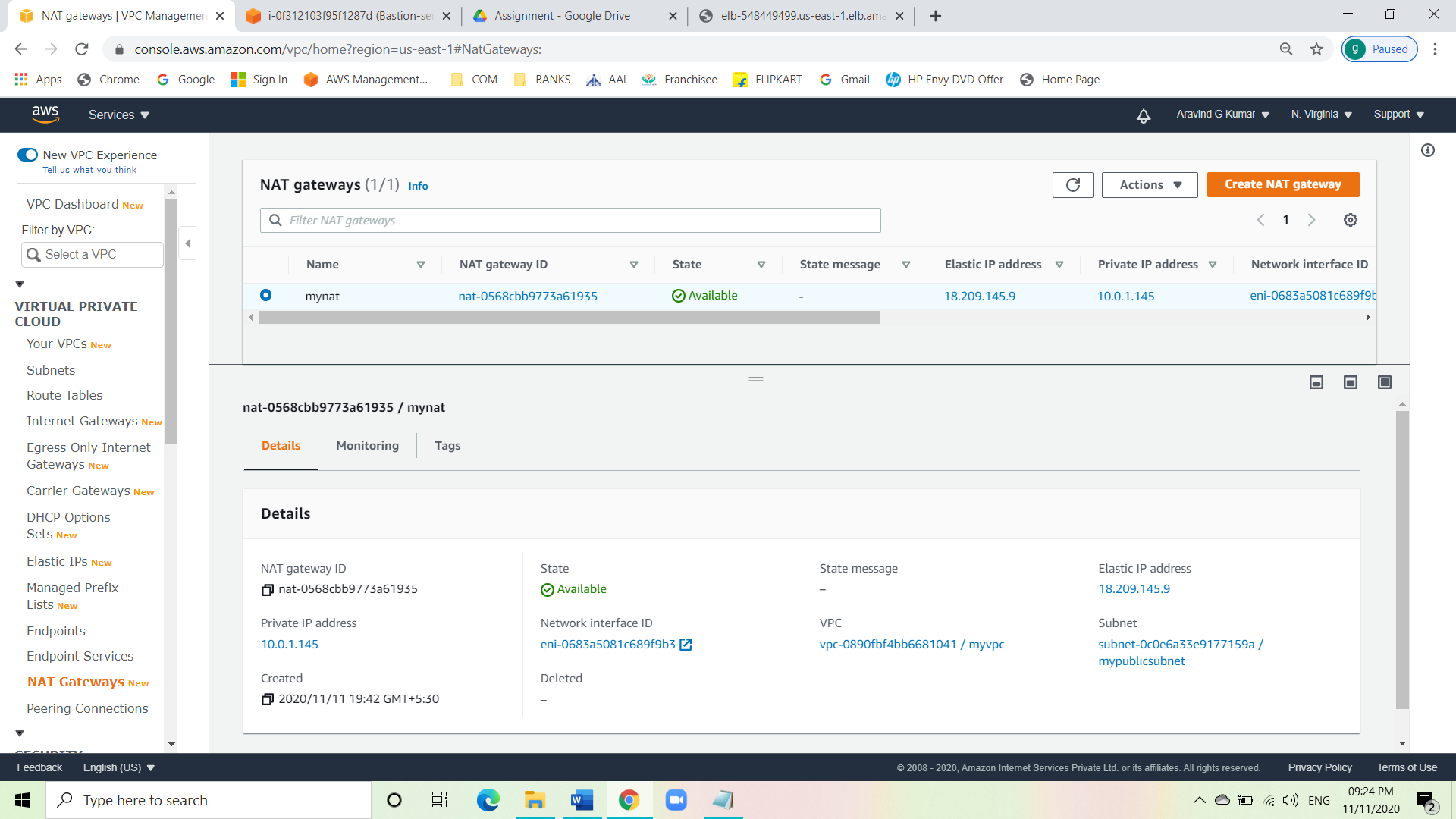
Private subnet 2

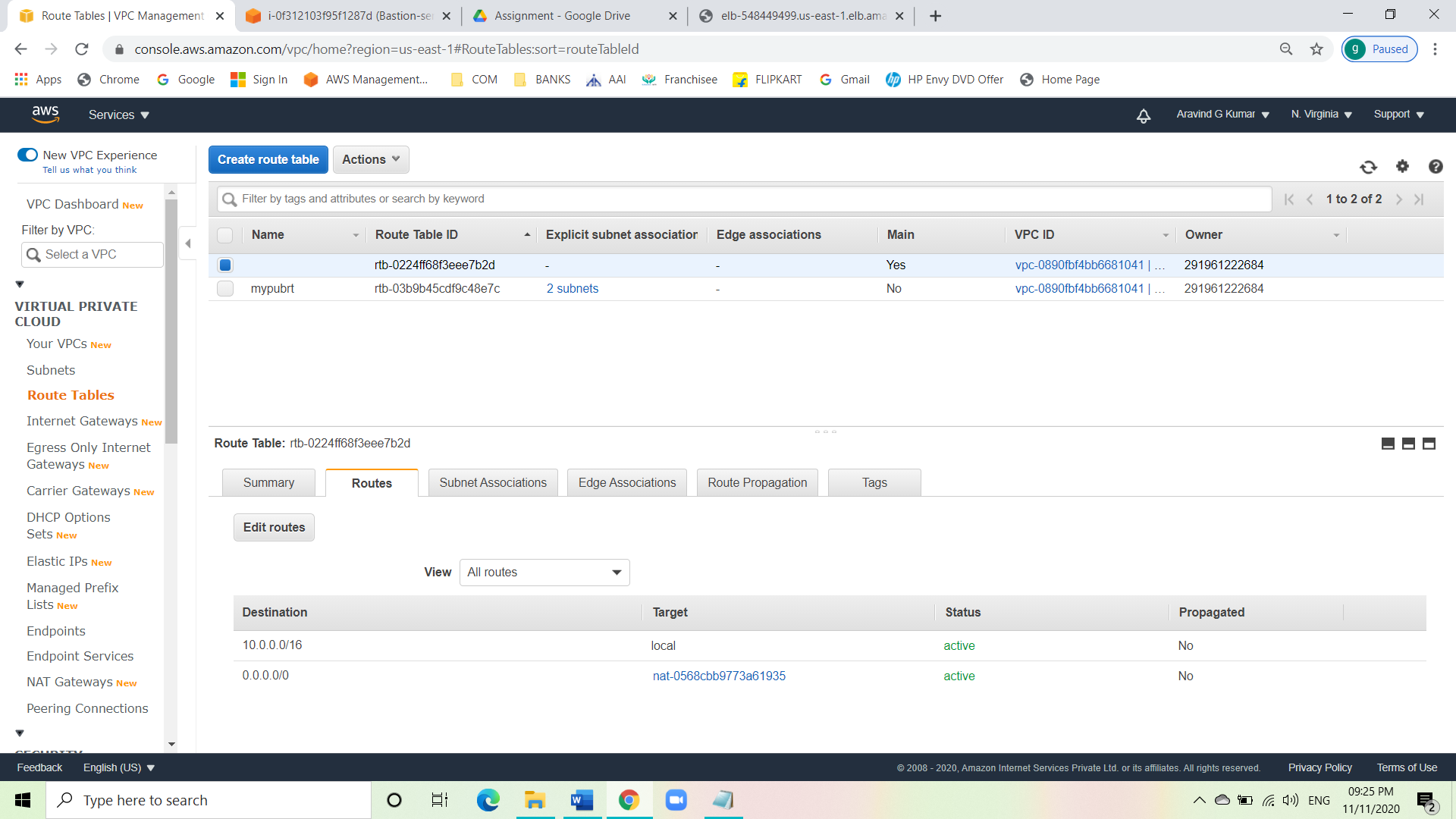


* Click on auto assign public IPv4 on the two public subnets created so that it will have a public IP address.
* Now create a public route table.
* Edit the routes and add a new route with CIDR 0.0.0.0/0 and attach it with the created IGW.

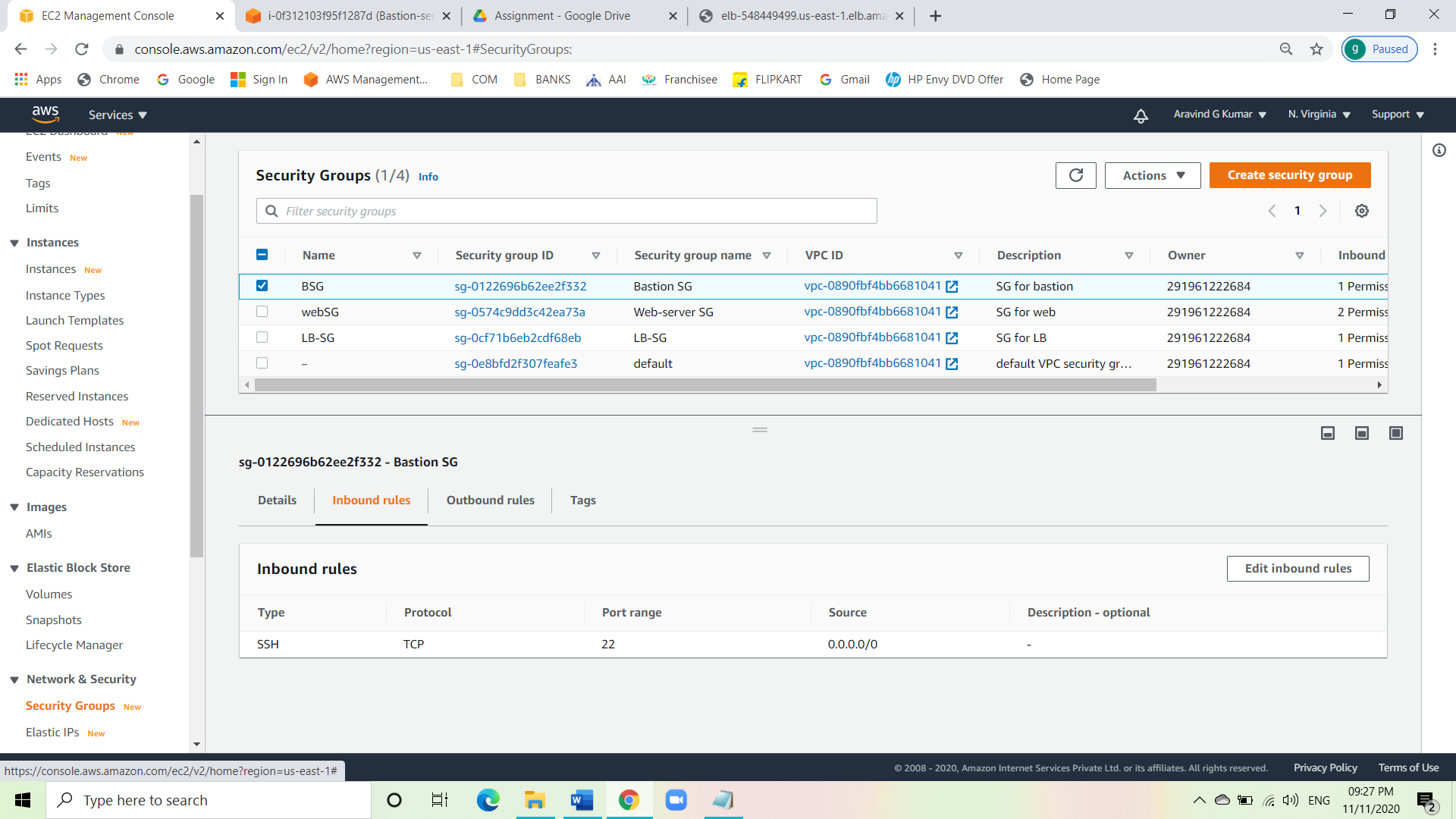


* Create a NAT gateway and attach it the default main route table.

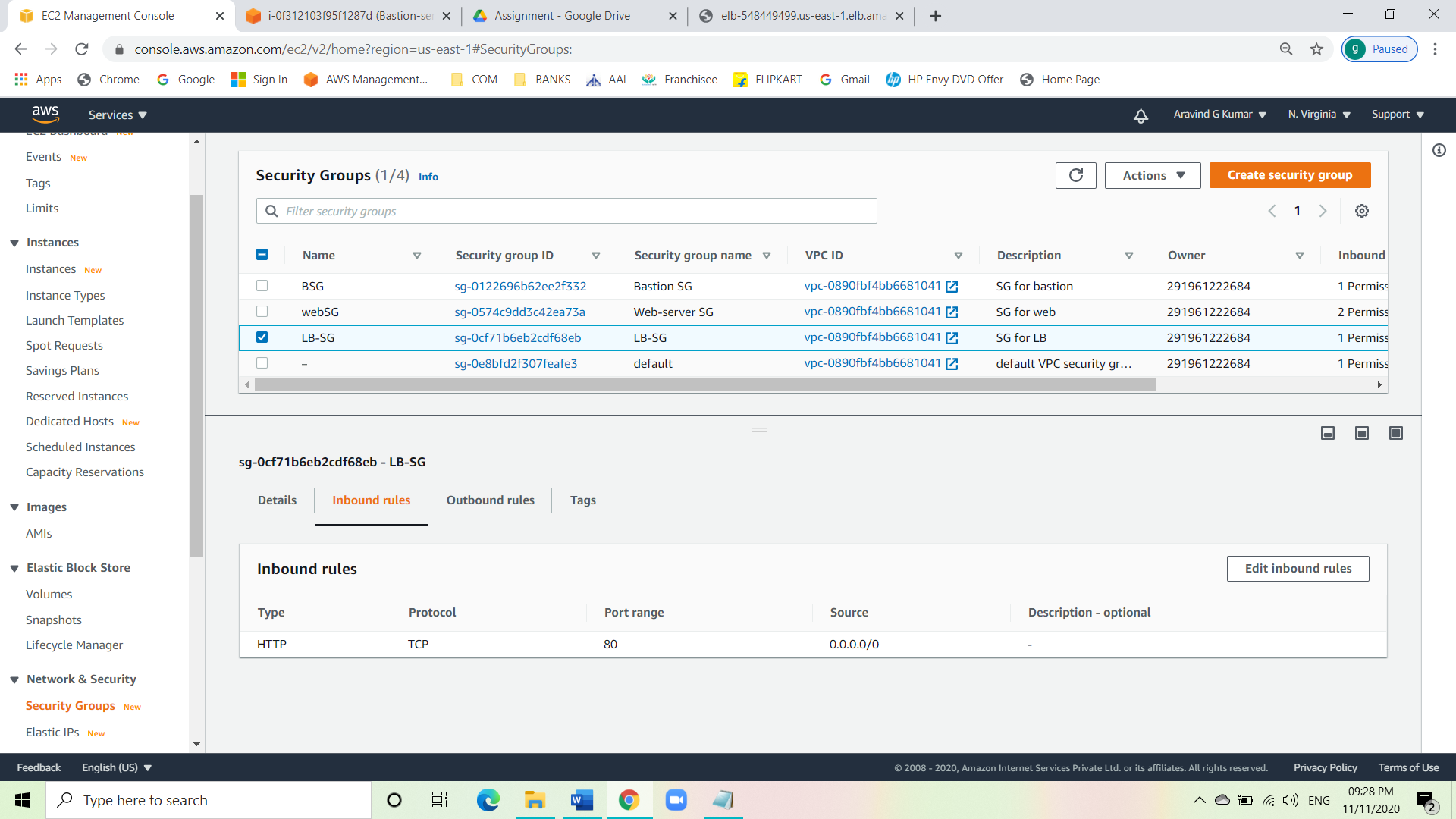




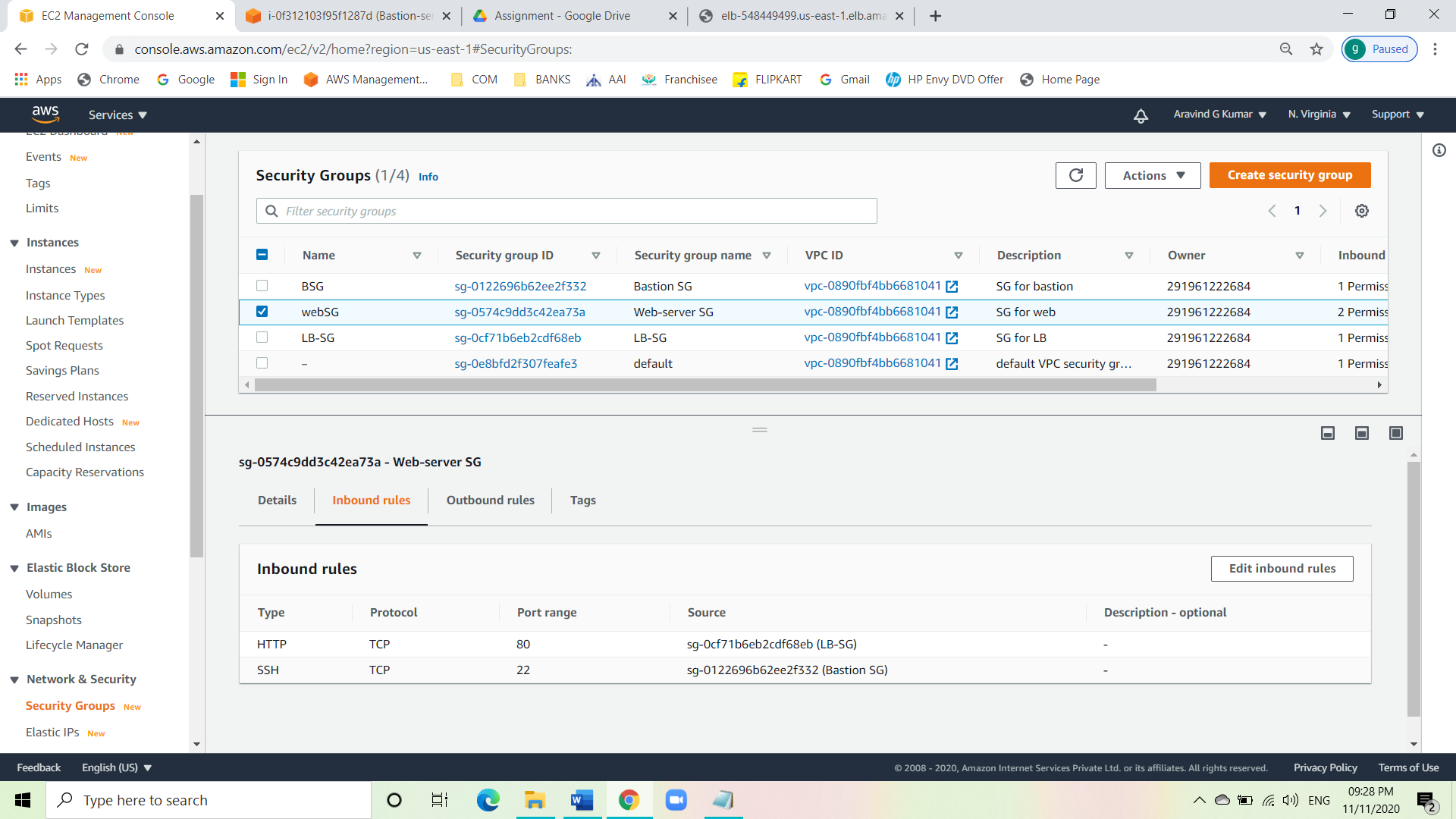
* Now go to EC2 dashboard.
* Go to security group.
* Create a security group for the Bastion server.
* Choose the type as SSH and source to custom (0.0.0.0/0).



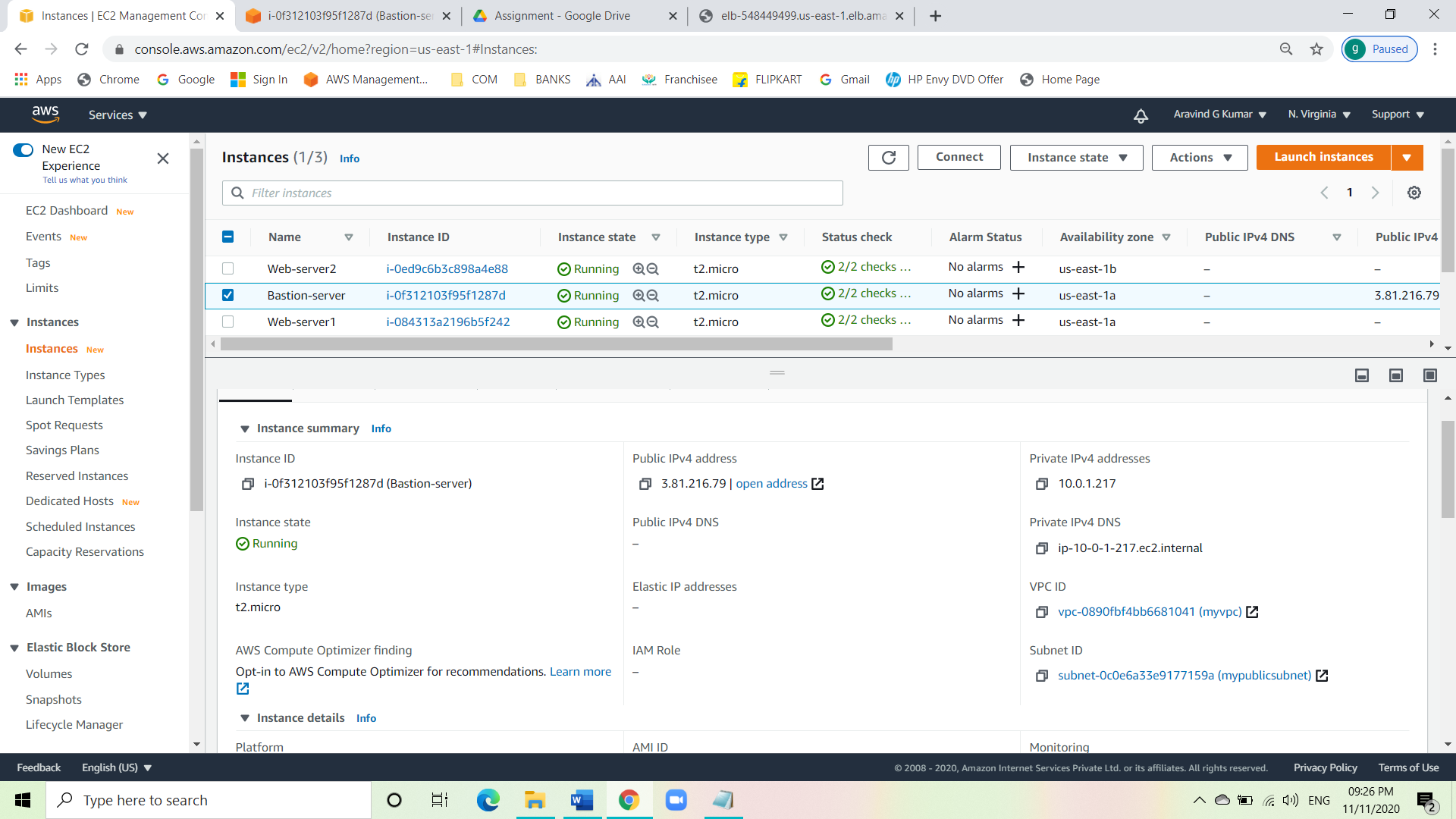
* Create a SG for the load balancer.
* Choose the type as HTTP and source to custom (0.0.0.0/0).



* Create a SG for the web servers.
* Choose the type as SSH and source to custom and choose the bastion SG.
* Add rule and choose the type HTTP and source to custom and choose load balancer SG.

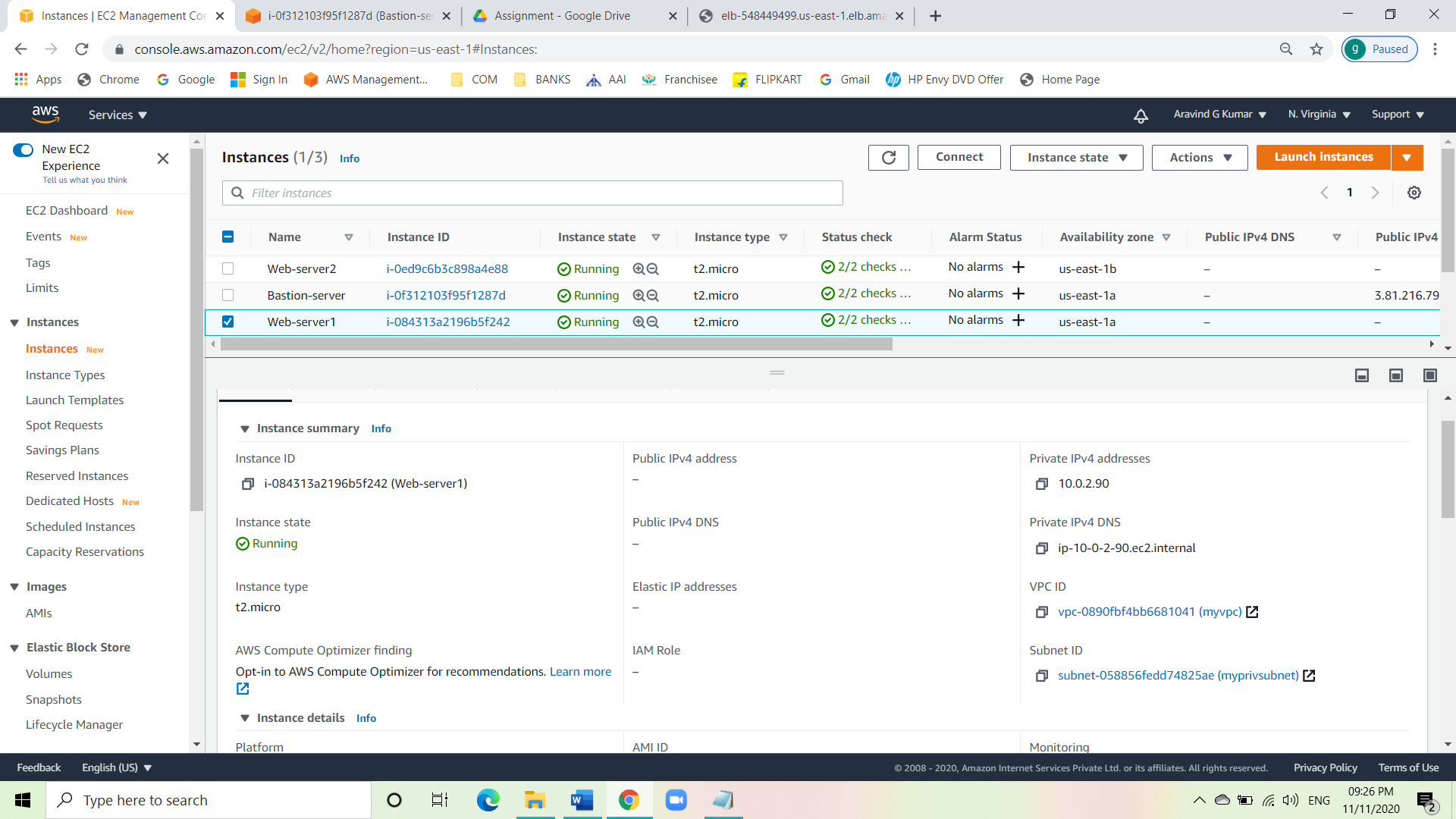


* Now start creating instances.
* First create a bastion instance with Linux AMI with the created VPC and public subnet attached to it.
* Select the created SG of bastion server and launch the instance.

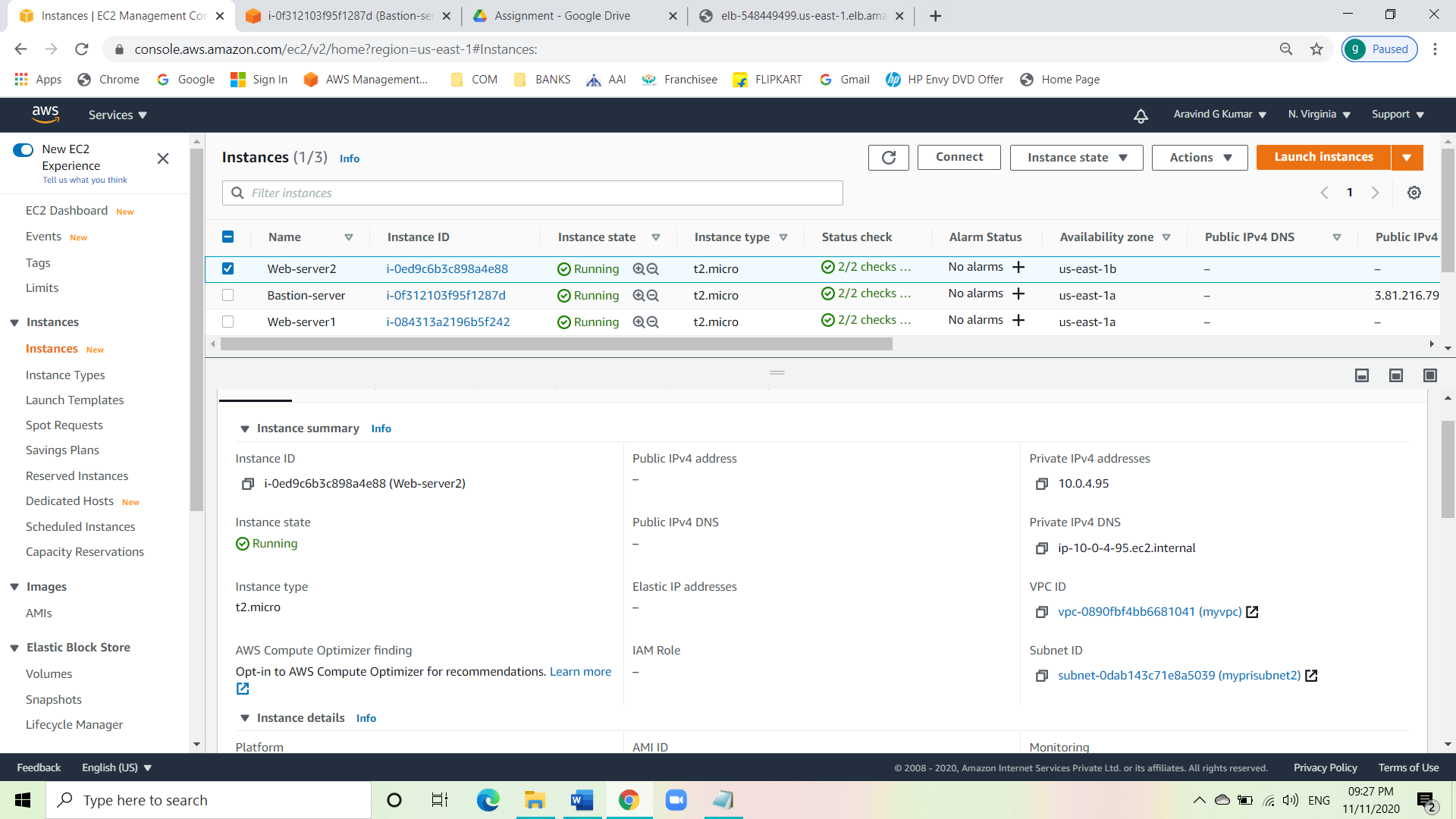


* Now create two web-servers with Linux AMI with the created VPC and attach the two private subnets to them.
* Choose the web-server SG for the two web-servers.

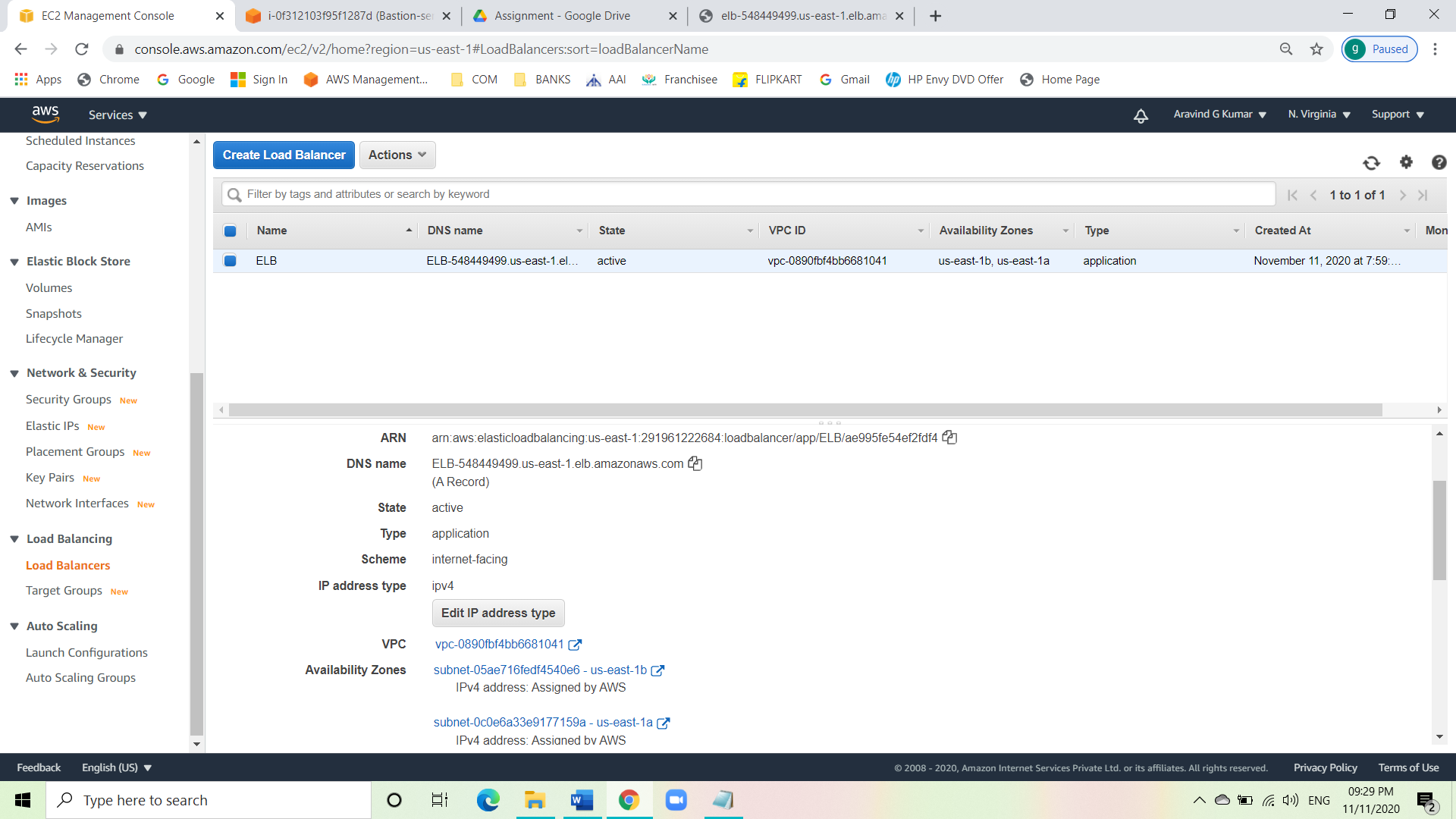
Web-server1



Web-server2



* Now create an ELB.
* Choose the VPC created.
* Select the created SG for load balancer.
* Give the target a name and under health give the path as **/index.html**.
* Register the targets. That is register the two web-servers with the ELB.
* Launch the ELB.



* Now connect the Bastion server using EC2 connect.
* Follow the codes below.

sudo su

vi web-serverkey.pem

i

copy the pem file

Esc:wq

Chmod 400 web-serverkey.pem

Ssh -I web-serverkey.pem ec2-user@private ip of web servers

Yes

Sudo su  
 yum update -y

Yum install httpd -y

Systemctl install httpd

Systemctl enable httpd

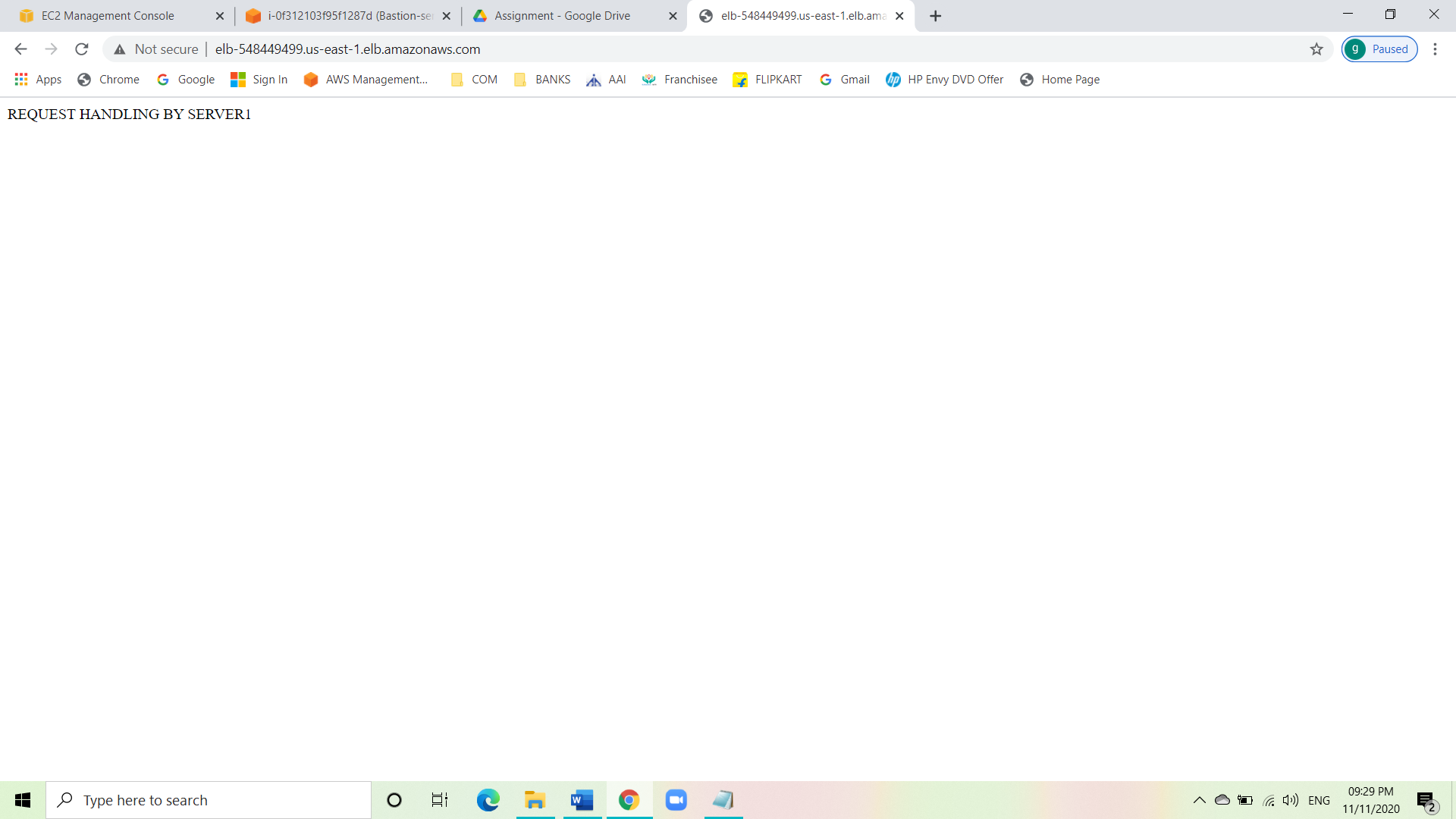
Echo “REQUEST HANDLING BY SERVER1/2”

Exit

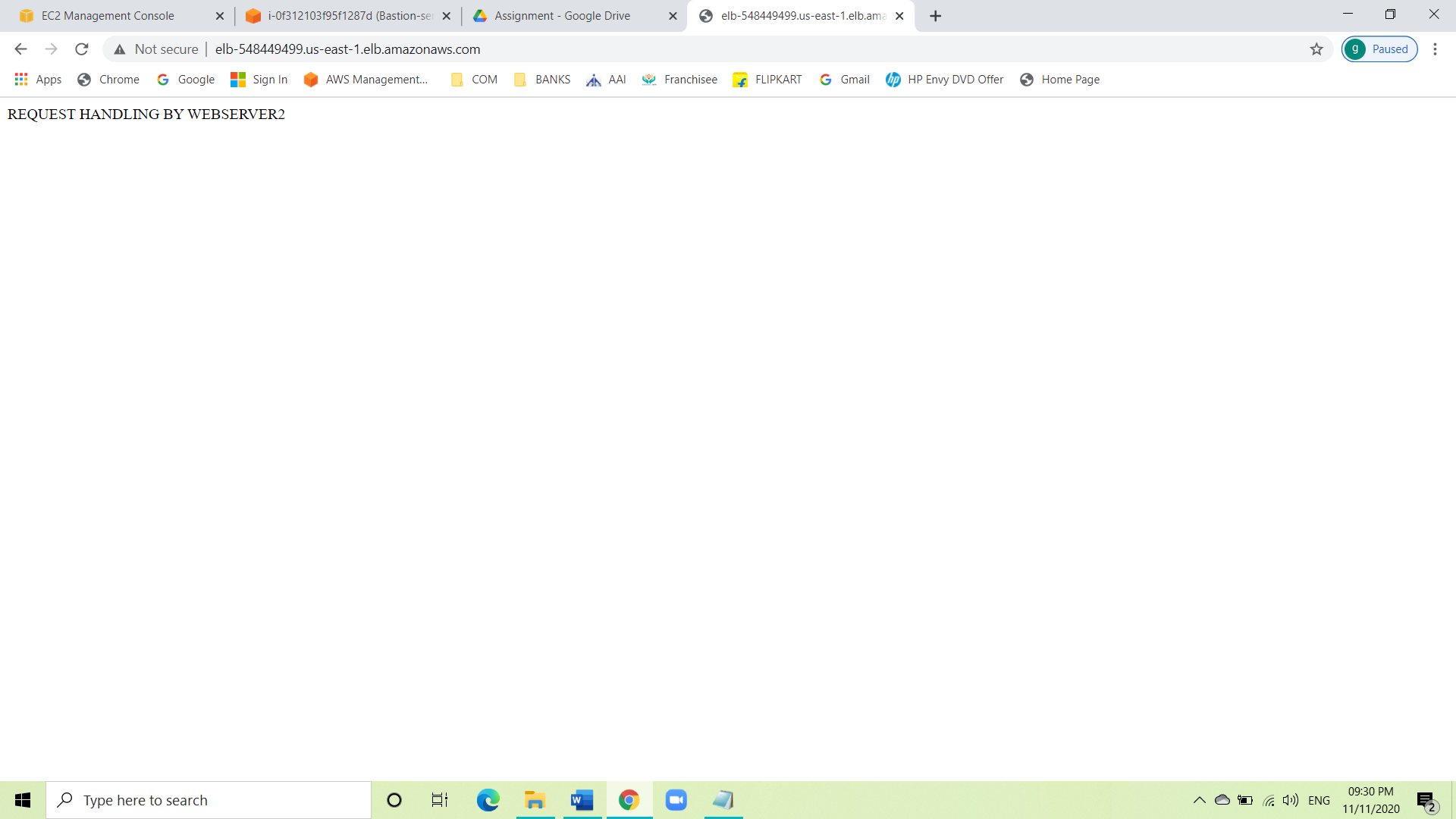
Exit

* Now use the ELB DNS link to see the message we entered for both the servers are reflecting.

Web-server1 message:



Web-server2 message:



**Conclusion:**

We have launched a Bastion server and two web-servers. We were able to SSH into the servers via Bastion Server successfully. We launched an Application Load Balancer and associated our web servers with the load balancer. We tested the load sharing between web servers. We successfully tested the high availability of the web application by making one of the web servers unhealthy.