

AWS ASSESSMENT PROJECT

1) Create a VPC with a private subnet and a public subnet

The screenshot shows the AWS Management Console interface for the 'Subnets' page. The left sidebar contains navigation links for VPC Dashboard, Subnets, Route Tables, Internet Gateways, Egress Only Internet Gateways, Carrier Gateways, DHCP Options Sets, Elastic IPs, Managed Prefix, VPCs, Endpoints, Endpoint Services, NAT Gateways, and Peering Connections. The main content area displays a table of subnets with columns: Name, Subnet ID, State, VPC, IPv4 CIDR, Availability IPv4, IPv6 CIDR, Availability Zone, and Availability. Two subnets are listed: 'MyPublicSub' (subnet-03a18077273c527b) and 'MyPrivateSub' (subnet-0a0711477c752d78). Below the table, the details for 'MyPublicSub' are shown, including its description, flow logs, route table, network ACL, tags, and sharing information. The subnet is in the 'available' state, associated with VPC 'vpc-0232e04765796a88', and has an IPv4 CIDR of '10.0.0.0/24'. It is located in the 'us-east-1' availability zone.

Name	Subnet ID	State	VPC	IPv4 CIDR	Availability IPv4	IPv6 CIDR	Availability Zone	Availability
MyPublicSub	subnet-03a18077273c527b	available	vpc-0232e04765796a88	10.0.0.0/24	251	-	us-east-1f	us-east-1
MyPrivateSub	subnet-0a0711477c752d78	available	vpc-0232e04765796a88	10.0.1.0/24	251	-	us-east-1f	us-east-1

Subnet: subnet-03a18077273c527b

Description	Flow Logs	Route Table	Network ACL	Tags	Sharing
Subnet ID	subnet-03a18077273c527b				
VPC	vpc-0232e04765796a88 / MyVPC				
Available IPv4 Addresses	251				
Availability Zone	us-east-1f us-east-1c us-east-1d				
Route Table	rt-015a90710a0400				
Default subnet	No				
Auto-assign customer-owned IPv4 address	No				
Auto-assign IPv6 address	No				
Owner	30885234083				
Network Border Group	us-east-1				
Network ACL	acl-0582c313980110894				
Auto-assign public IPv4 address	Yes				
Customer-owned IPv4 pool	-				
Output ID	-				

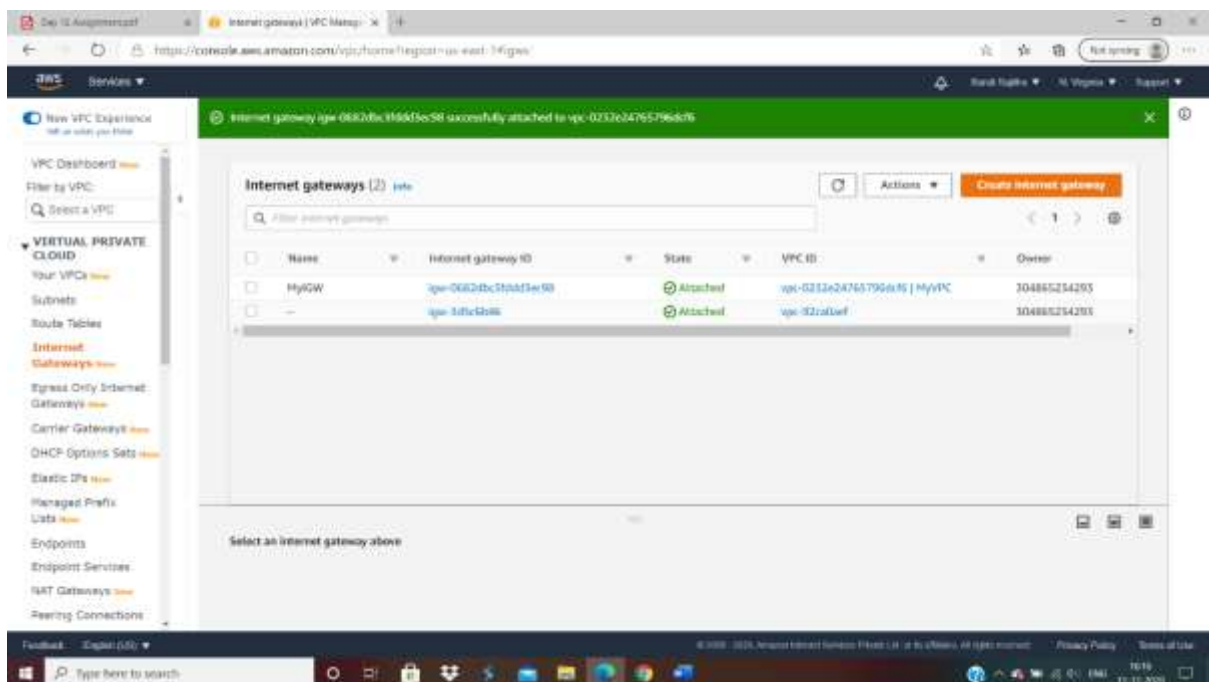
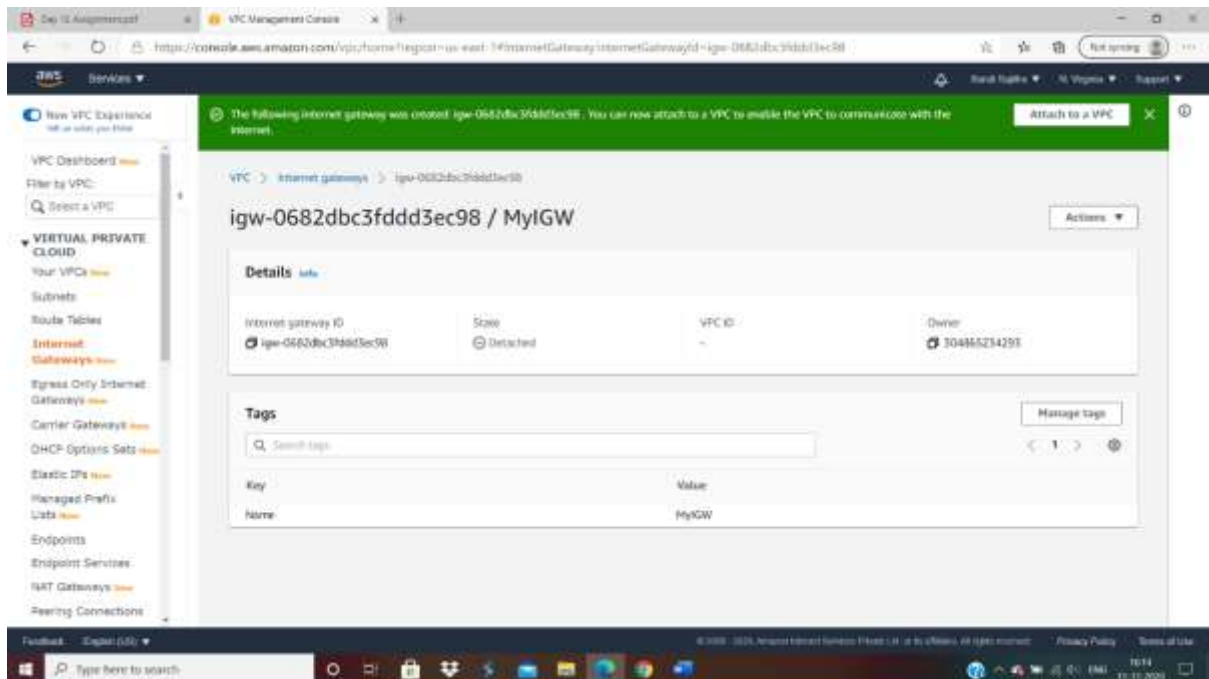
The screenshot shows the AWS Management Console interface for the 'Subnets' page, displaying the details for the 'MyPrivateSub' subnet. The left sidebar is identical to the previous screenshot. The main content area shows the details for 'MyPrivateSub' (subnet-0a0711477c752d78). The subnet is in the 'available' state, associated with VPC 'vpc-0232e04765796a88', and has an IPv4 CIDR of '10.0.1.0/24'. It is located in the 'us-east-1' availability zone.

Name	Subnet ID	State	VPC	IPv4 CIDR	Availability IPv4	IPv6 CIDR	Availability Zone	Availability
MyPublicSub	subnet-03a18077273c527b	available	vpc-0232e04765796a88	10.0.0.0/24	251	-	us-east-1f	us-east-1
MyPrivateSub	subnet-0a0711477c752d78	available	vpc-0232e04765796a88	10.0.1.0/24	251	-	us-east-1f	us-east-1

Subnet: subnet-0a0711477c752d78

Description	Flow Logs	Route Table	Network ACL	Tags	Sharing
Subnet ID	subnet-0a0711477c752d78				
VPC	vpc-0232e04765796a88 / MyVPC				
Available IPv4 Addresses	251				
Availability Zone	us-east-1f us-east-1c us-east-1d				
Route Table	rt-015a90710a0400				
Default subnet	No				
Auto-assign customer-owned IPv4 address	No				
Auto-assign IPv6 address	No				
Owner	30885234083				
Network Border Group	us-east-1				
Network ACL	acl-0582c313980110894				
Auto-assign public IPv4 address	Yes				
Customer-owned IPv4 pool	-				
Output ID	-				

2) Create a IGW and associate with the public subnet



3) Create a route table with VPC

The screenshot shows the 'Create route table' page in the AWS Management Console. The page title is 'Create route table'. Below the title, there is a description: 'A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.' The 'Name tag' field is set to 'PublicRouteTable'. The 'VPC' dropdown menu is set to 'vpc-0232e04765796a08'. There is a table with two columns: 'Key' and 'Value', both with a limit of 255 characters. Below the table, it says 'This resource currently has no tags'. There is an 'Add Tag' button with a note '(90 remaining) (up to 50 tags maximum)'. At the bottom right, there are 'Cancel' and 'Create' buttons.

Route Tables + Create route table

Create route table

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Name tag: PublicRouteTable

VPC: vpc-0232e04765796a08

Key	Value
This resource currently has no tags	

Add Tag (90 remaining) (up to 50 tags maximum)

* Required

Cancel Create



The screenshot shows the 'Route Tables' page in the AWS Management Console. The page title is 'Route Tables (VPC Management)'. There is a 'Create route table' button. Below the button, there is a table with columns: Name, Route Table ID, Explicit subnet association, Edge associations, Main, VPC ID, and Owner. The table contains three rows of data. Below the table, there is a section for 'Route Table: rt-0552120818e0c18f'. This section has tabs for 'Summary', 'Routes', 'Subnet Associations', 'Edge Associations', 'Route Propagation', and 'Tags'. The 'Summary' tab is selected, showing details for the route table, including its ID, VPC, and owner.

Route Tables (VPC Management)

Create route table

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
rt-0d15e80318cd480c	rt-0d15e80318cd480c	-	-	Yes	vpc-0232e04765796a08	304865234293
PublicRoute...	rt-0552120818e0c18f	-	-	No	vpc-0232e04765796a08	304865234293
rt-084735cd	rt-084735cd	-	-	Yes	vpc-0232e04765796a08	304865234293

Route Table: rt-0552120818e0c18f

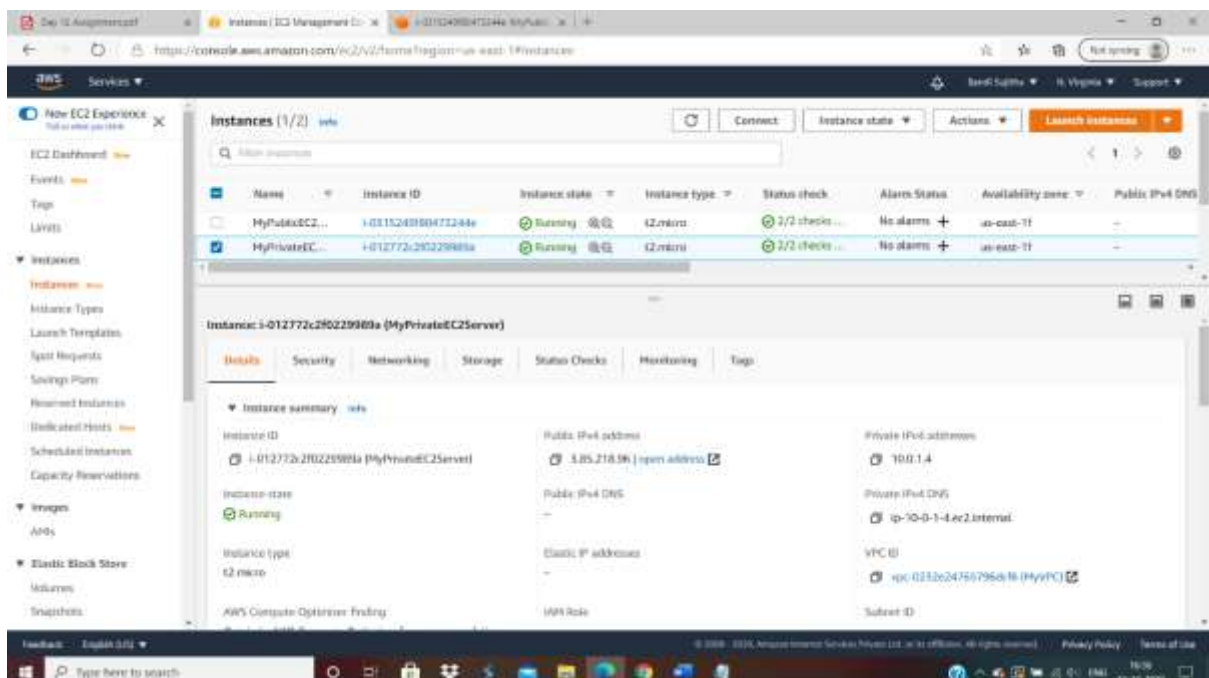
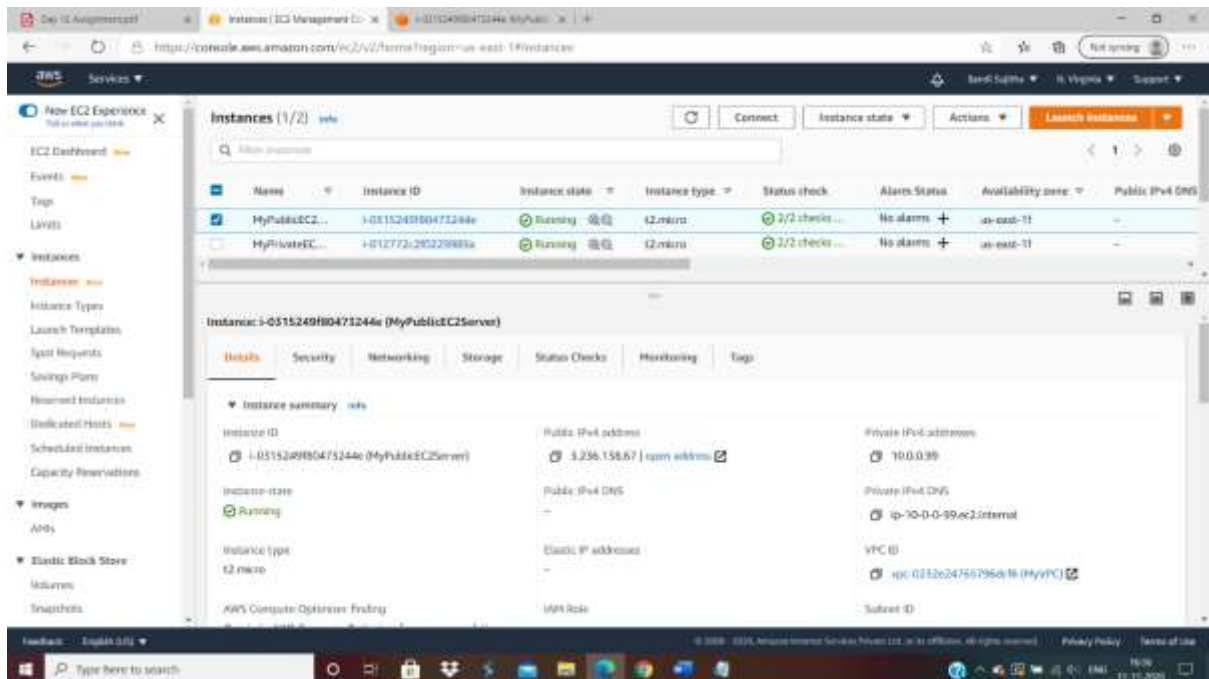
Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID: rt-0552120818e0c18f

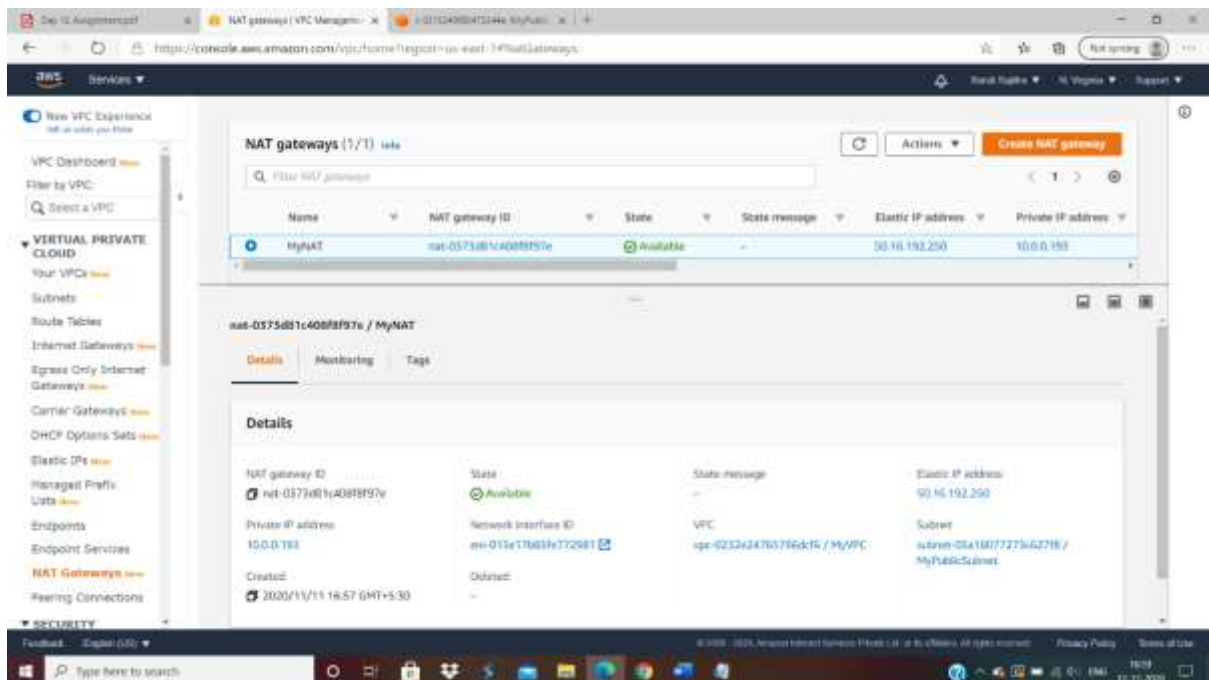
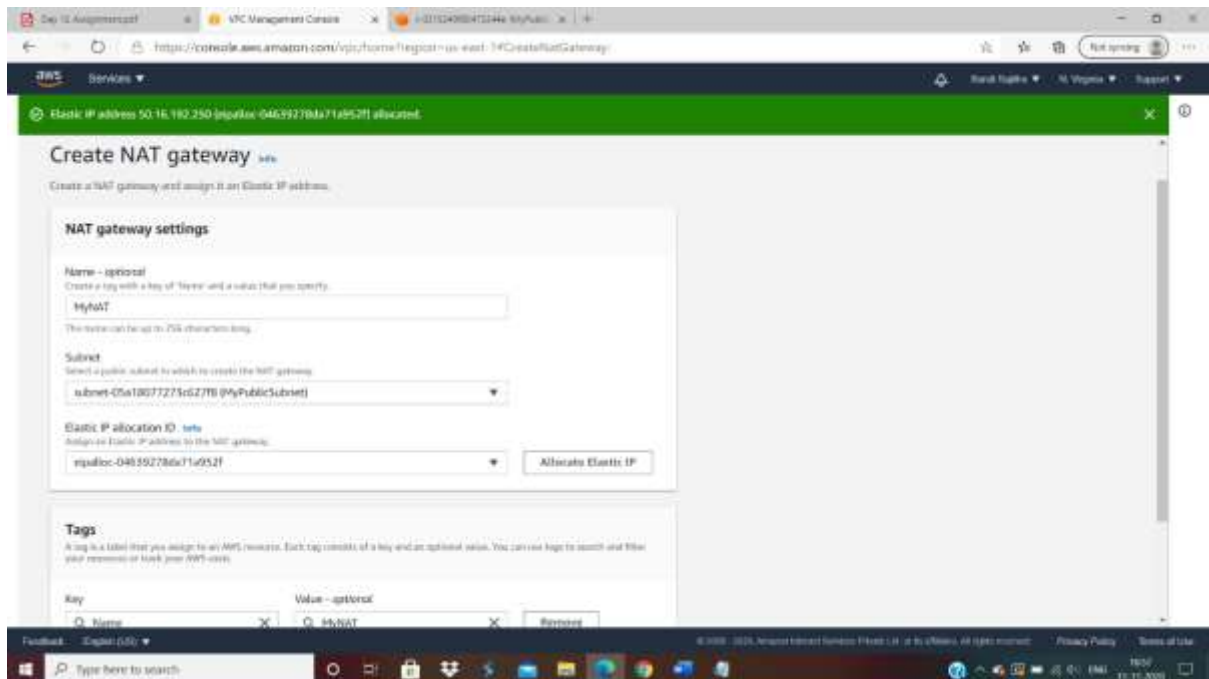
Explicitly Associated with VPC: vpc-0232e04765796a08 (My VPC)

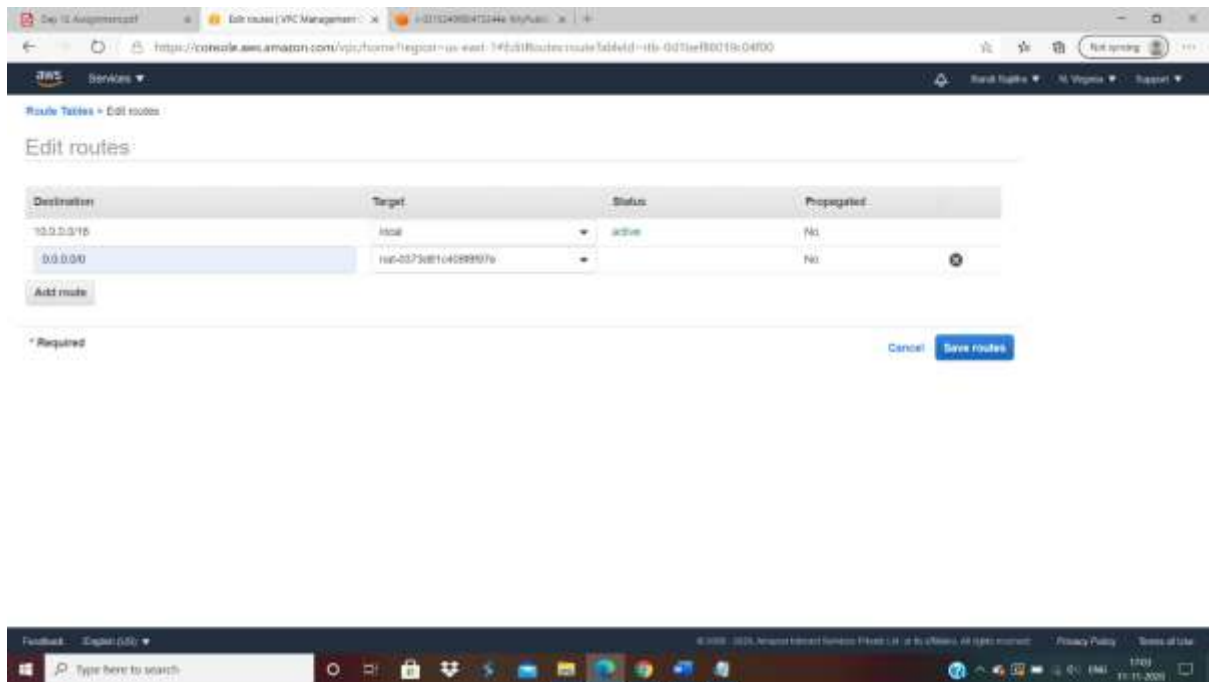
Owner: 304865234293

4. Creating two instances using linux

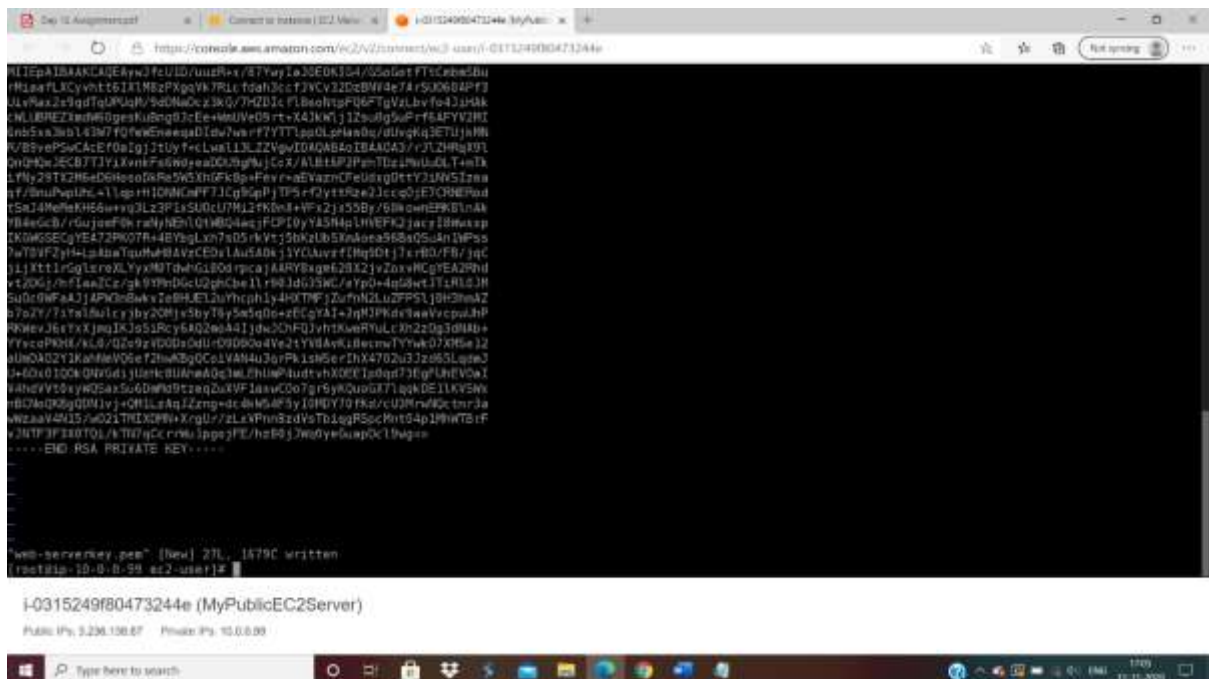


5) Create a NAT gateway and associate with public subnet

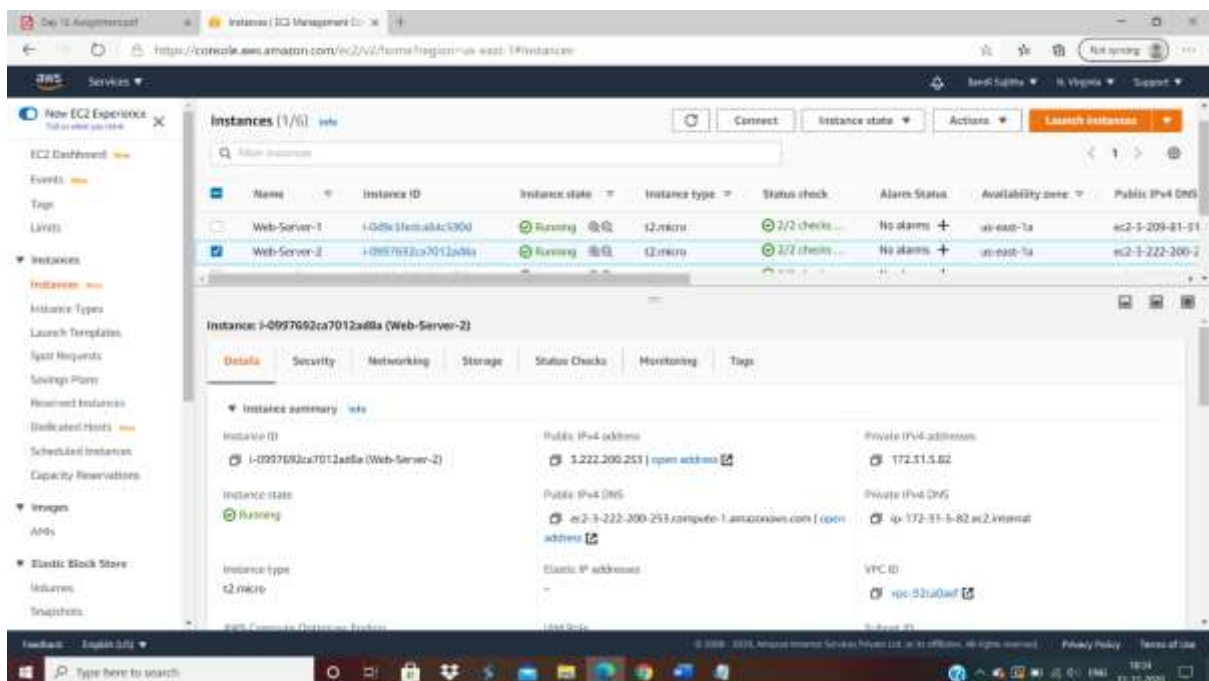
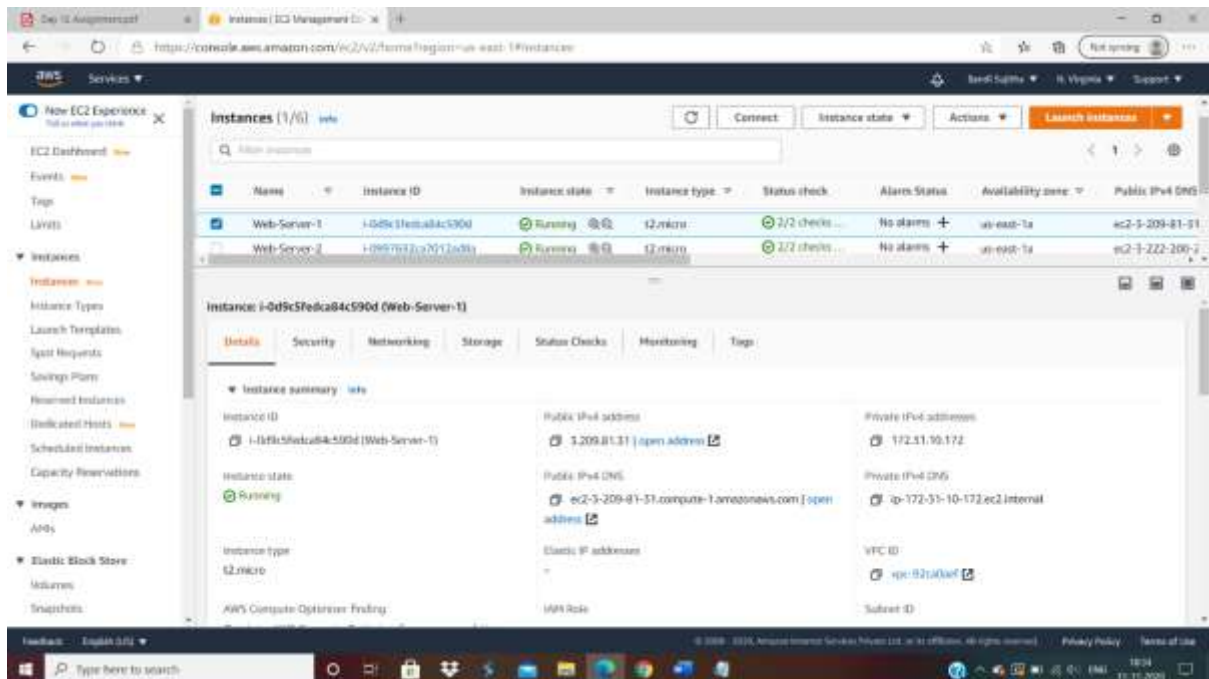




6) Connect the public instance and copy the private key and login to private ip



8) launch two webserver in the private subnet



9) Create a load balancer in the public subnet range

Step 3: Configure Security Groups
A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Filter: VPC security groups

Security Group ID	Name	Description	Actions
sg-09552b47a7c25e9	Bastion-SG	Security group for Bastion Server	Copy to new
sg-07210b6f	default	default VPC security group	Copy to new
sg-0669ee33ac2b1b41	launch-wizard-1	launch-wizard-1 created 2025-11-18T12:34:40.266+05:30	Copy to new
sg-0a6c321867ab0d057	LoadBalancer-SG	Security group for the load balancer	Copy to new
sg-079150ba23230e844	WebServer-SG	Security group for webserver	Copy to new
sg-0083e60e0a1e3e79	WebServer-SG	Security group for webserver	Copy to new

[Cancel](#) [Previous](#) [Next: Configure Routing](#)

Step 5: Register Targets
Register targets with your target group. If you register a target in an enabled Availability Zone, the load balancer starts routing requests to the targets as soon as the registration process completes and the target passes the initial health checks.

Registered targets
To deregister instances, select one or more registered instances and then click Remove.

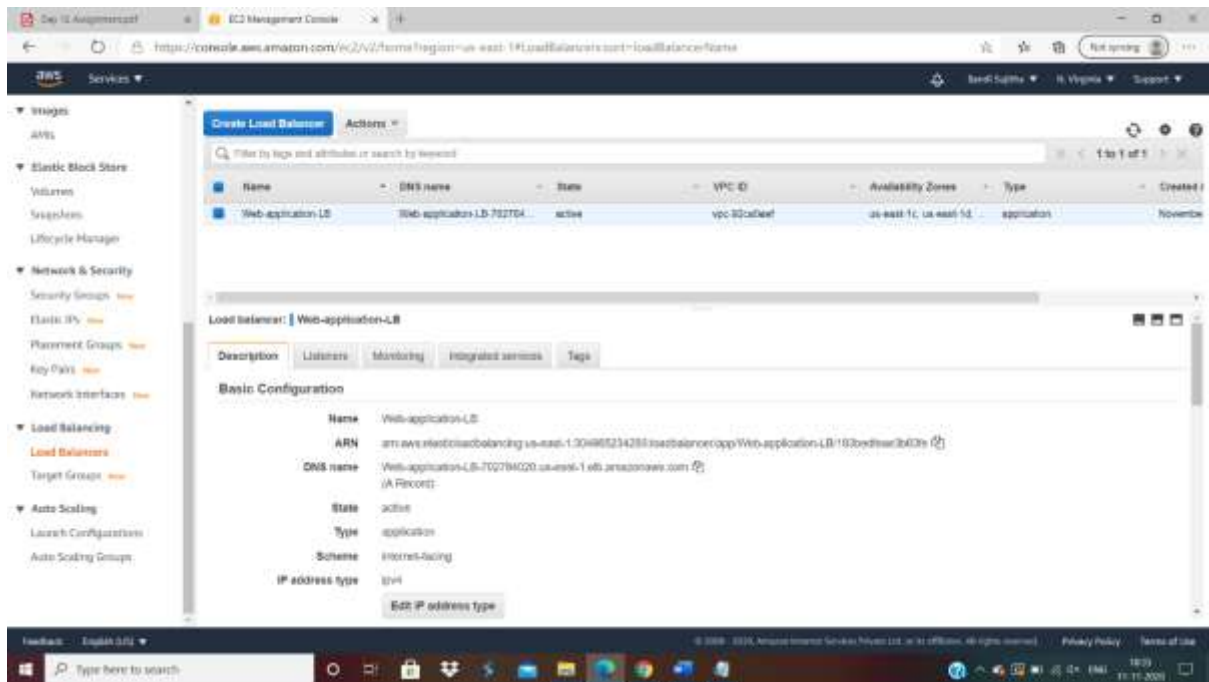
Instance	Name	Port	State	Security groups	Zone
i-00b36dca8c9903	WebServer-1	80	running	WebServer-SG	us-east-1a
i-08a7992ca7013a894	WebServer-2	80	running	WebServer-SG	us-east-1a

Instances
To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

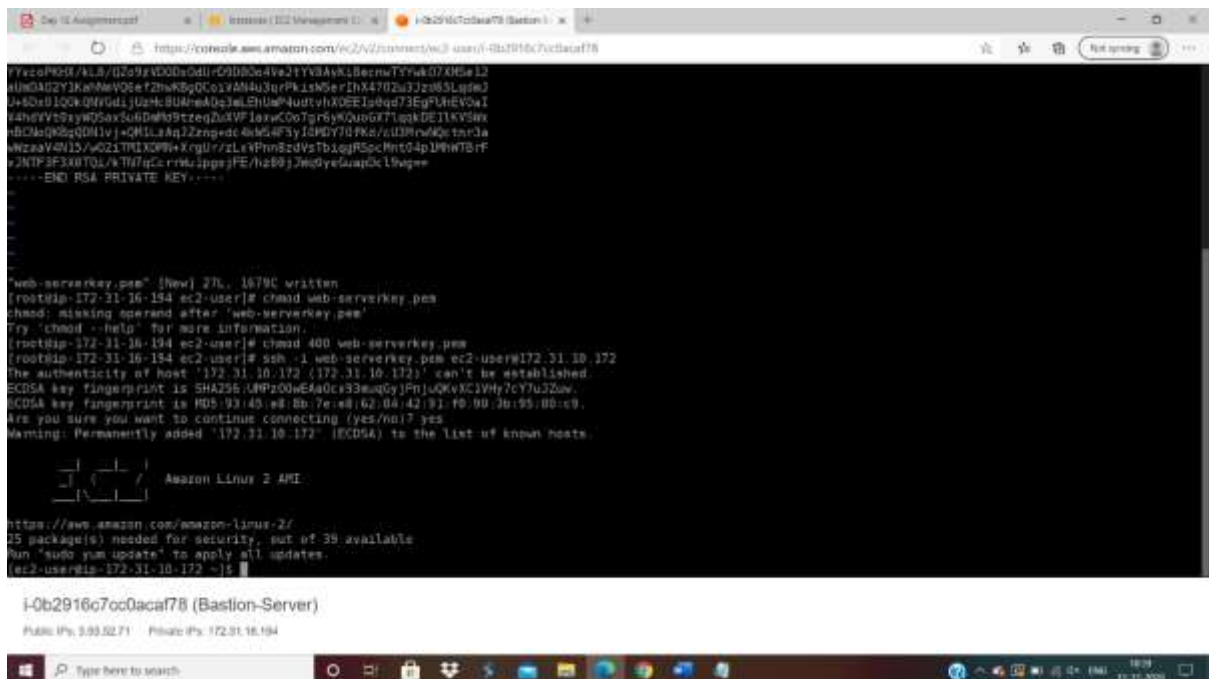
[Add to registered](#) on port: 80

Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
i-00b36dca8c9903	WebServer-1	running	WebServer-SG	us-east-1a	subnet-7a516b1	172.31.0.0/28
i-08a7992ca7013a894	WebServer-2	running	WebServer-SG	us-east-1a	subnet-7a516b1	172.31.0.0/28
i-0b31916c7a0a0a77f	HealthCheck-Target	monitoring	HealthCheck-SG	us-east-1a	subnet-7a516b1	172.31.0.0/28

[Cancel](#) [Previous](#) [Next: Review](#)



10) Connect the Bastion server with the web server 1 & 2 private ip addresses



Step 12: Augmenting ec2

https://console.aws.amazon.com/ec2/v2/console/ec2-user/i-0b2916c7oc0aca7f8

```
Installing mailcap-2.1.41-2.amzn2.noarch 5/9
Installing httpdfilesystem-2.4.46-1.amzn2.noarch 7/9
Installing mod_http2-1.15.14-2.amzn2.x86_64 8/9
Installing httpd-2.4.46-1.amzn2.x86_64 9/9
Verifying apr-util-1.6.1-5.amzn2.0.2.x86_64 1/9
Verifying httpdfilesystem-2.4.46-1.amzn2.noarch 2/9
Verifying apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/9
Verifying httpd-tools-2.4.46-1.amzn2.x86_64 4/9
Verifying mod_http2-1.15.14-2.amzn2.x86_64 5/9
Verifying apr-1.6.3-5.amzn2.0.2.x86_64 6/9
Verifying mailcap-2.1.41-2.amzn2.noarch 7/9
Verifying generic-logos-httpd-18.0.0-4.amzn2.noarch 8/9
Verifying httpd-2.4.46-1.amzn2.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.46-1.amzn2

Dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2          apr-util.x86_64 0:1.6.1-5.amzn2.0.2          apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
generic-logos-httpd.noarch 0:18.0.0-4.amzn2  httpdfilesystem.noarch 0:2.4.46-1.amzn2          httpd-tools.x86_64 0:2.4.46-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2          mod_http2.x86_64 0:1.15.14-2.amzn2

Complete!
[root@ip-172-31-10-172 ec2-user]# systemctl start httpd
[root@ip-172-31-10-172 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-10-172 ec2-user]# echo "REQUEST HANDLING BY SERVER1">index.html
[root@ip-172-31-10-172 ec2-user]# echo "REQUEST HANDLING BY SERVER1">index.html
[root@ip-172-31-10-172 ec2-user]# exit
exit
[ec2-user@ip-172-31-10-172 ~]$ exit
logout
Connection to 172.31.10.172 closed.
root@ip-172-31-10-194 ec2-user]#
```

i-0b2916c7oc0aca7f8 (Bastion-Server)

Public IP: 3.93.52.71 Private IP: 172.31.16.194

Type here to search

Step 12: Augmenting ec2

https://console.aws.amazon.com/ec2/v2/console/ec2-user/i-0b2916c7oc0aca7f8

```
Installing generic-logos-httpd-18.0.0-4.amzn2.noarch 5/9
Installing mailcap-2.1.41-2.amzn2.noarch 6/9
Installing httpdfilesystem-2.4.46-1.amzn2.noarch 7/9
Installing mod_http2-1.15.14-2.amzn2.x86_64 8/9
Installing httpd-2.4.46-1.amzn2.x86_64 9/9
Verifying apr-util-1.6.1-5.amzn2.0.2.x86_64 1/9
Verifying httpdfilesystem-2.4.46-1.amzn2.noarch 2/9
Verifying apr-util-bdb-1.6.1-5.amzn2.0.2.x86_64 3/9
Verifying httpd-tools-2.4.46-1.amzn2.x86_64 4/9
Verifying mod_http2-1.15.14-2.amzn2.x86_64 5/9
Verifying apr-1.6.3-5.amzn2.0.2.x86_64 6/9
Verifying mailcap-2.1.41-2.amzn2.noarch 7/9
Verifying generic-logos-httpd-18.0.0-4.amzn2.noarch 8/9
Verifying httpd-2.4.46-1.amzn2.x86_64 9/9

Installed:
httpd.x86_64 0:2.4.46-1.amzn2

Dependency Installed:
apr.x86_64 0:1.6.3-5.amzn2.0.2          apr-util.x86_64 0:1.6.1-5.amzn2.0.2          apr-util-bdb.x86_64 0:1.6.1-5.amzn2.0.2
generic-logos-httpd.noarch 0:18.0.0-4.amzn2  httpdfilesystem.noarch 0:2.4.46-1.amzn2          httpd-tools.x86_64 0:2.4.46-1.amzn2
mailcap.noarch 0:2.1.41-2.amzn2          mod_http2.x86_64 0:1.15.14-2.amzn2

Complete!
[root@ip-172-31-5-82 ec2-user]# systemctl start httpd
[root@ip-172-31-5-82 ec2-user]# systemctl enable httpd
Created symlink from /etc/systemd/system/multi-user.target.wants/httpd.service to /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-5-82 ec2-user]# echo "REQUEST HANDLING BY SERVER2">index.html
[root@ip-172-31-5-82 ec2-user]# exit
exit
[ec2-user@ip-172-31-5-82 ~]$ exit
logout
Connection to 172.31.5.82 closed.
root@ip-172-31-16-194 ec2-user]#
```

i-0b2916c7oc0aca7f8 (Bastion-Server)

Public IP: 3.93.52.71 Private IP: 172.31.16.194

Type here to search