

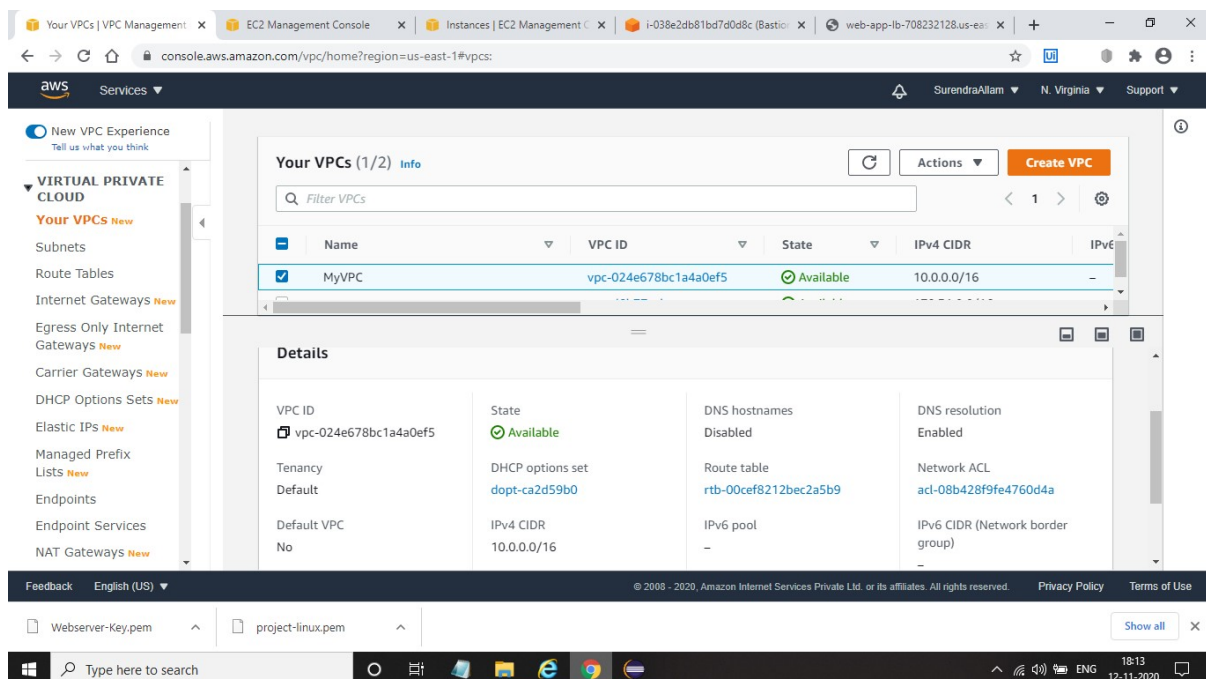
Advance AWS | Project-01

Deploying a Highly Available Web Application in VPC Using Bastion Host in AWS

Task-01: Creating a VPC without using VPC Wizard.

Name-Tag: MyVPC

IPv4 CIDR block: 10.0.0.0/16



Task-02: Create an Internet Gateway.

For the Public Subnet.

Name-Tag: MyPublicsubnet

IPv4 CIDR block: 10.0.1.0/24

The screenshot shows the AWS Management Console interface for the VPC service. The left sidebar lists various VPC services, and the main area displays the details for a specific subnet. The subnet is named 'MyPublicSubnet' and is in an 'available' state. It is associated with VPC 'vpc-024e678bc1a4a0ef5' and has an IPv4 CIDR of '10.0.1.0/24'. The available IPv4 addresses are 248. The subnet is located in the 'us-east-1c' availability zone and is associated with the 'rtb-0f9c95ed464f66263' route table. The network ACL is 'acl-08b428f9fe4760d4a'. The 'Auto-assign public IPv4 address' option is enabled.

Attribute	Value
Subnet ID	subnet-05efd87793c1c9965
VPC	vpc-024e678bc1a4a0ef5 MyVPC
State	available
IPv4 CIDR	10.0.1.0/24
Available IPv4 Addresses	248
IPv6 CIDR	-
Availability Zone	us-east-1c (use1-az4)
Network Border Group	us-east-1
Route Table	rtb-0f9c95ed464f66263 MyPublicRoute
Network ACL	acl-08b428f9fe4760d4a
Default subnet	No
Auto-assign public IPv4 address	Yes
Auto-assign customer-owned IPv4 address	No
Customer-owned IPv4 pool	-

For the Private Subnet.

Name-Tag: MyPrivatesubnet

IPv4 CIDR block: 10.0.2.0/24

The screenshot shows the AWS Management Console interface for the VPC service. The left sidebar lists various VPC services, and the main area displays the details for a specific subnet. The subnet is named 'MyPrivateSubnet' and is in an 'available' state. It is associated with VPC 'vpc-024e678bc1a4a0ef5' and has an IPv4 CIDR of '10.0.2.0/24'. The available IPv4 addresses are 249. The subnet is located in the 'us-east-1c' availability zone and is associated with the 'rtb-00ce8f212bec2a5b9' route table. The network ACL is 'acl-08b428f9fe4760d4a'. The 'Auto-assign public IPv4 address' option is disabled.

Attribute	Value
Subnet ID	subnet-06603590776d1dd00
VPC	vpc-024e678bc1a4a0ef5 MyVPC
State	available
IPv4 CIDR	10.0.2.0/24
Available IPv4 Addresses	249
IPv6 CIDR	-
Availability Zone	us-east-1c (use1-az4)
Network Border Group	us-east-1
Route Table	rtb-00ce8f212bec2a5b9
Network ACL	acl-08b428f9fe4760d4a
Default subnet	No
Auto-assign public IPv4 address	No
Auto-assign customer-owned IPv4 address	No
Customer-owned IPv4 pool	-

Task-03: Create private and public subnets for the VPC.

Name-Tag: My-IGW

The screenshot shows the AWS Management Console interface for Internet Gateways. The left sidebar contains navigation links for VPC services. The main content area displays a list of Internet Gateways with columns for Name, Internet gateway ID, State, and VPC ID. The 'My-IGW' gateway is highlighted, showing it is in an 'Attached' state and associated with the VPC 'vpc-024e678bc1a4a0ef5 | MyVPC'. Below the list, the 'Details' section provides further information about the selected gateway.

Name	Internet gateway ID	State	VPC ID
My-IGW	igw-096883a041a45baba	Attached	vpc-024e678bc1a4a0ef5 MyVPC
-	igw-4456683f	Attached	vpc-d6b77cab

Internet gateway ID	State	VPC ID	Owner
igw-096883a041a45baba	Attached	vpc-024e678bc1a4a0ef5 MyVPC	573310877256

Task-04: Create and Configure Route tables.

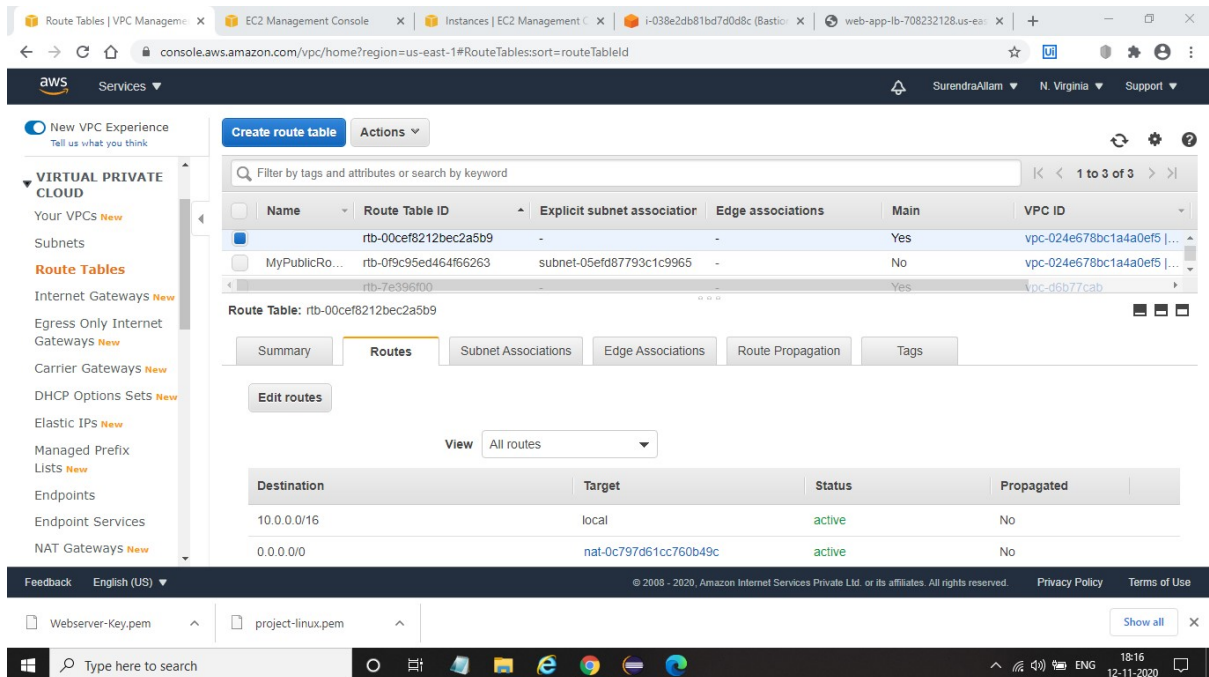
Name-Tag: MyPublicRoute

The screenshot shows the AWS Management Console interface for Route Tables. The left sidebar contains navigation links for VPC services. The main content area displays a list of Route Tables with columns for Name, Route Table ID, Explicit subnet association, Edge associations, Main, and VPC ID. The 'MyPublicRoute' table is highlighted, showing it is associated with the VPC 'vpc-024e678bc1a4a0ef5 | ...'. Below the list, the 'Routes' section shows the configuration for the selected route table, including a route to the Internet Gateway 'igw-096883a041a45baba'.

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID
MyPublicRoute	rtb-0f9c95ed464f66263	subnet-05efd87793c1c9965	-	No	vpc-024e678bc1a4a0ef5 ...

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	igw-096883a041a45baba	active	No

Name-Tag: MyPrivateRoute

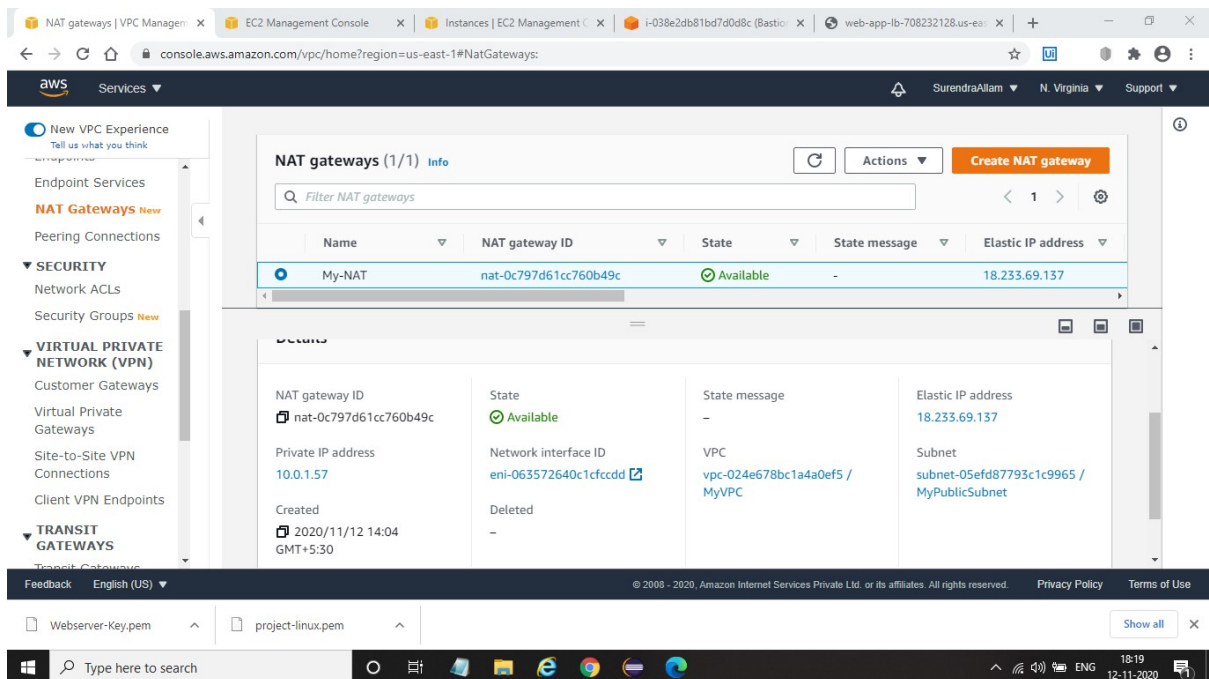


The screenshot shows the AWS Management Console for the 'Route Tables' page. The left sidebar lists various services under 'VIRTUAL PRIVATE CLOUD'. The main content area shows a list of route tables. The 'MyPrivateRoute' table is selected, and its details are shown below. The 'Routes' tab is active, displaying a table of routes.

Destination	Target	Status	Propagated
10.0.0.0/16	local	active	No
0.0.0.0/0	nat-0c797d61cc760b49c	active	No

Task-05: Create NAT- Gateway and attach the VPC

Task-06: Attach NAT Gateway to correct main Route table.



The screenshot shows the AWS Management Console for the 'NAT gateways' page. The left sidebar lists various services under 'VIRTUAL PRIVATE CLOUD'. The main content area shows a list of NAT gateways. The 'My-NAT' gateway is selected, and its details are shown below.

NAT gateway ID	State	State message	Elastic IP address
nat-0c797d61cc760b49c	Available	-	18.233.69.137

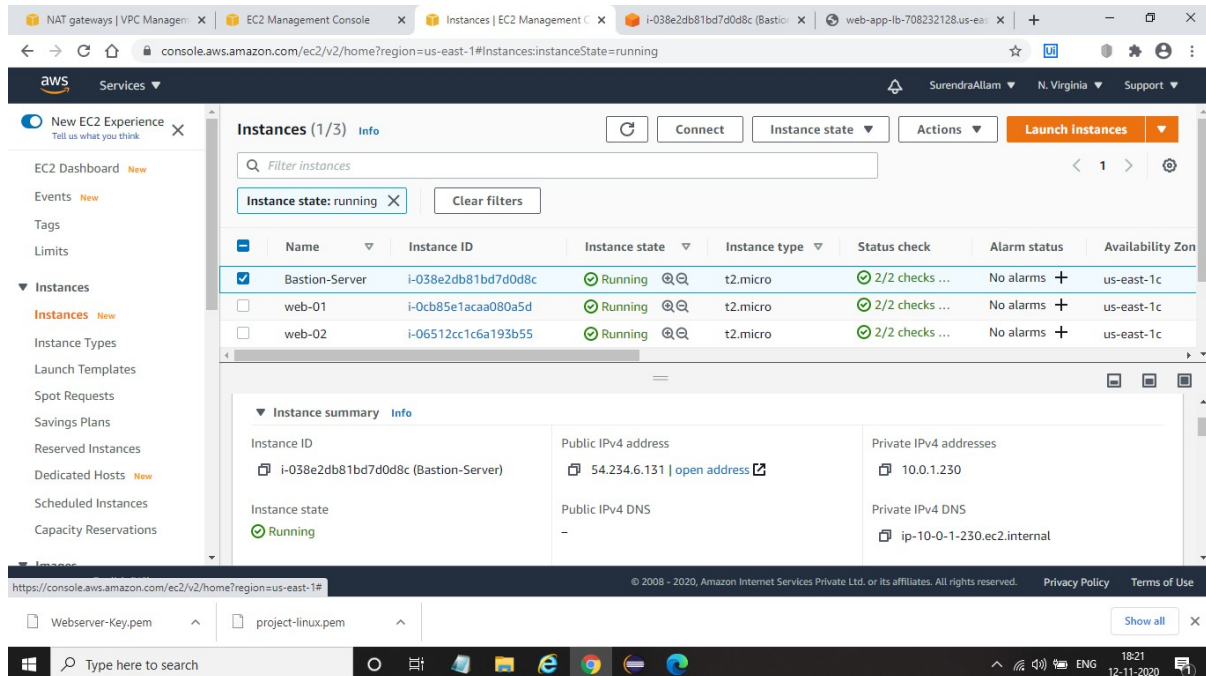
Additional details for the selected NAT gateway:

- Private IP address: 10.0.1.57
- Created: 2020/11/12 14:04 GMT+5:30
- Network interface ID: eni-063572640c1cfccdd
- VPC: vpc-024e678bc1a4a0ef5 / MyVPC
- Subnet: subnet-05efd87793c1c9965 / MyPublicSubnet

Task-07: Launch a Bastion Host instance in a Public Subnet.

Name-Tag: Bastion-Server

Bastion-Server Public IP: 54.234.6.131



The screenshot shows the AWS Management Console with the EC2 Instances page. The 'Bastion-Server' instance is selected, and its details are shown in the summary section. The instance is running and has a public IP address of 54.234.6.131.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Bastion-Server	i-038e2db81bd7d0d8c	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-01	i-0cb85e1acaa080a5d	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-02	i-06512cc1c6a193b55	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-038e2db81bd7d0d8c (Bastion-Server)	54.234.6.131 open address	10.0.1.230

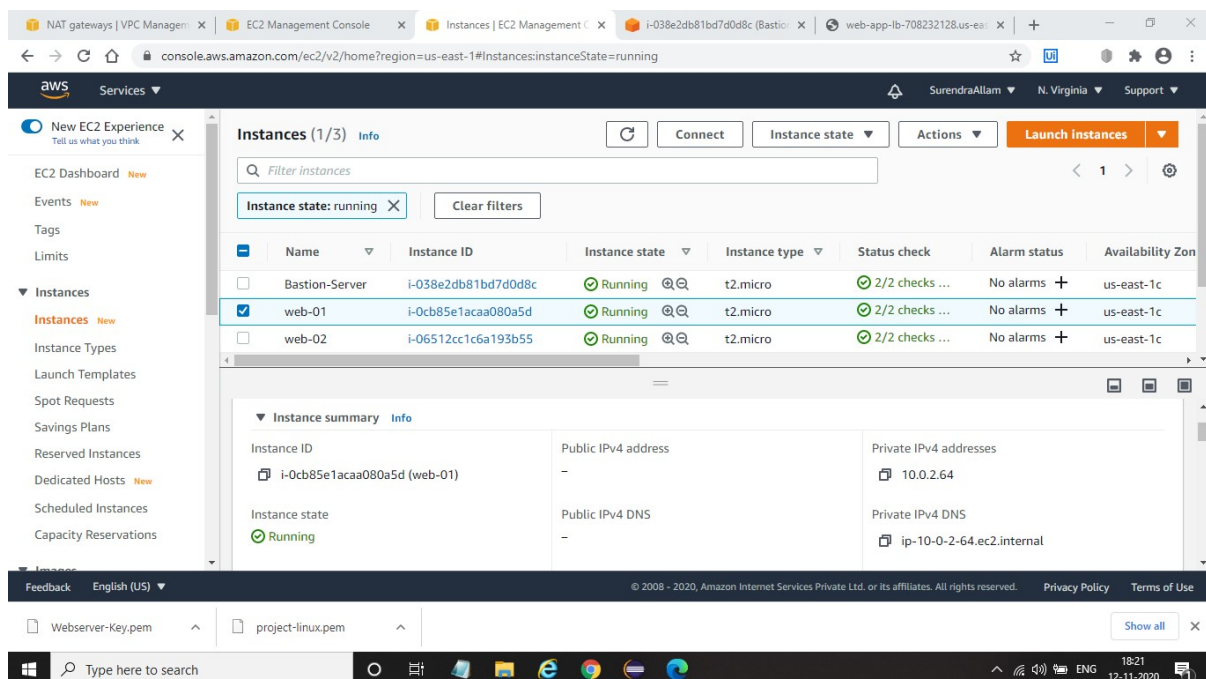
Instance state

Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	-	ip-10-0-1-230.ec2.internal

Task-08: Launch Web Server 01 in a Private Subnet.

Name-Tag: web-01

WebServer01 Private IP: 10.0.2.64



The screenshot shows the AWS Management Console with the EC2 Instances page. The 'web-01' instance is selected, and its details are shown in the summary section. The instance is running and has a private IP address of 10.0.2.64.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Bastion-Server	i-038e2db81bd7d0d8c	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-01	i-0cb85e1acaa080a5d	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-02	i-06512cc1c6a193b55	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0cb85e1acaa080a5d (web-01)	-	10.0.2.64

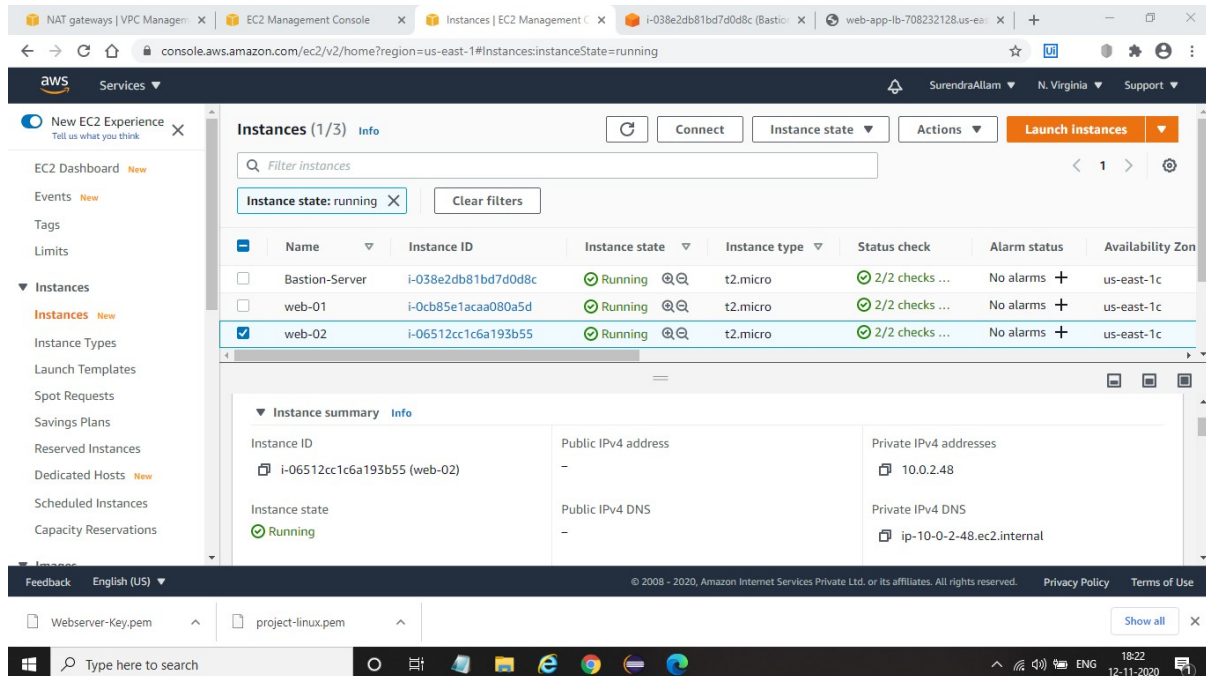
Instance state

Instance state	Public IPv4 DNS	Private IPv4 DNS
Running	-	ip-10-0-2-64.ec2.internal

Task-08: Launch Web Server 02 in a Private Subnet.

Name-Tag: web-02

WebServer02 Private IP: 10.0.2.48



The screenshot shows the AWS Management Console with the EC2 Instances page. The 'Instances (1/3)' list shows three instances: Bastion-Server, web-01, and web-02. web-02 is selected, and its details are shown in the summary pane below. The summary pane displays the Instance ID (i-06512cc1c6a193b55), Instance state (Running), Public IPv4 address (none), Private IPv4 addresses (10.0.2.48), Public IPv4 DNS (none), and Private IPv4 DNS (ip-10-0-2-48.ec2.internal).

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Bastion-Server	i-038e2db81bd7d0d8c	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-01	i-0cb85e1acaa080a5d	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-02	i-06512cc1c6a193b55	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c

Instance summary Info

Instance ID: i-06512cc1c6a193b55 (web-02)

Instance state: Running

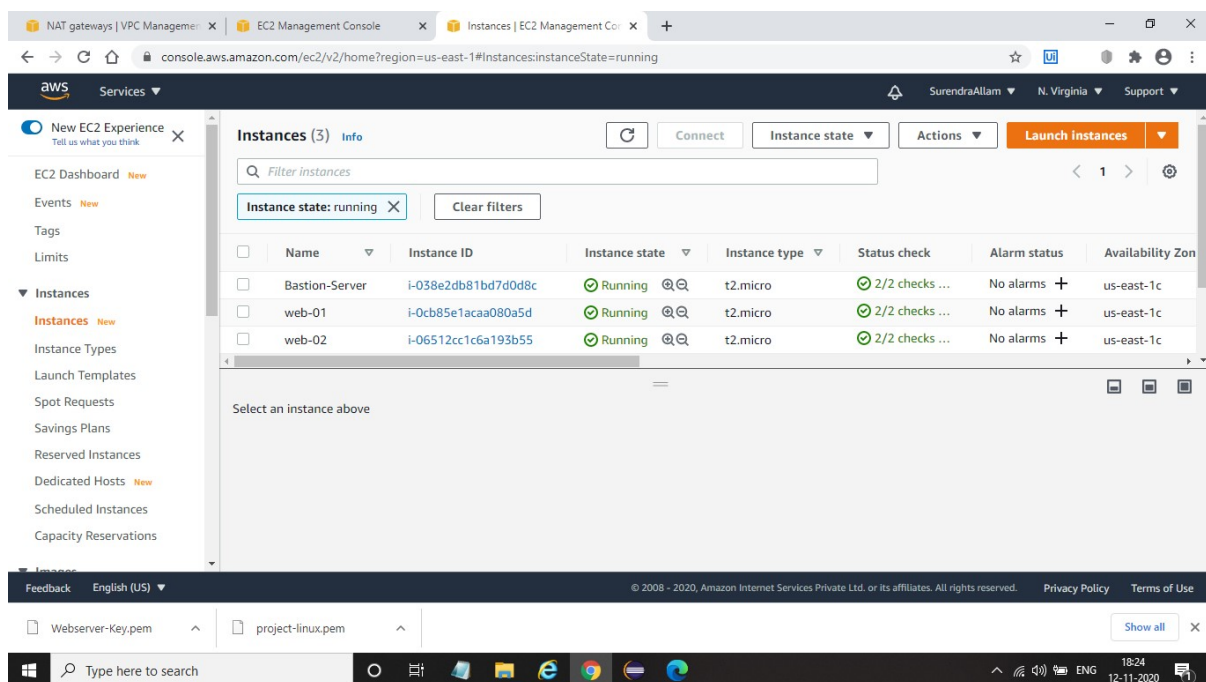
Public IPv4 address: -

Private IPv4 addresses: 10.0.2.48

Public IPv4 DNS: -

Private IPv4 DNS: ip-10-0-2-48.ec2.internal

Instance List:



The screenshot shows the AWS Management Console with the EC2 Instances page. The 'Instances (3)' list shows three instances: Bastion-Server, web-01, and web-02. All three instances are in the 'Running' state. The summary pane below the list shows the selected instance details.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
Bastion-Server	i-038e2db81bd7d0d8c	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-01	i-0cb85e1acaa080a5d	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c
web-02	i-06512cc1c6a193b55	Running	t2.micro	2/2 checks ...	No alarms	us-east-1c

Select an instance above

Task-09: Create a Load Balancer Security Group. Set up a Load Balancer and associate the two web instance to the Load Balancer.

The screenshot displays the AWS Management Console interface. The left-hand navigation pane shows the 'Instances' section expanded. The main content area is titled 'Load balancer: web-app-LB' and shows the 'Basic Configuration' tab. The configuration details are as follows:

Property	Value
Name	web-app-LB
ARN	arn:aws:elasticloadbalancing:us-east-1:573310877256:loadbalancer/app/web-app-LB/30690b217de1f344
DNS name	web-app-LB-708232128.us-east-1.elb.amazonaws.com (A Record)
State	active
Type	application

The top of the console shows a list of Load Balancers with the following details:

Name	DNS name	State	VPC ID	Availability Zones	Type
web-app-LB	web-app-LB-708232128.us-east-1.elb.amazonaws.com	active	vpc-024e678bc1a4a0ef5	us-east-1c, us-east-1e, ...	application

Checking Target Groups Health

The screenshot displays the AWS Management Console interface, specifically the 'Targets' tab for the 'web-app-LB' Target Group. The 'Registered targets (2)' section shows the following details:

Instance ID	Name	Port	Zone	Status	Status details
i-06512cc1c6a193b55	web-02	80	us-east-1c	healthy	
i-0cb85e1acaa080a5d	web-01	80	us-east-1c	healthy	

The top of the console shows the 'Group details' tab with the following information:

Property	Value
Target type	instance
Protocol : Port	HTTP : 80
Protocol version	HTTP1
VPC	vpc-024e678bc1a4a0ef5
Load balancer	web-app-LB

Task-10: SSH into the web servers via the Bastion Server.

Cmds:

1. vi web-serverkey.pem
2. i
3. paste the pem file text here
4. chmod 400web-serverkey.pem
5. ssh -i web-serverkey.pem ec2-user@<webserver private IP>

Task-11: Installing Apache and Publish a test index.html on both of the web servers.

ssh -i web-serverkey.pem ec2-user@<webserver-01 private IP>

1. Installing Apache in webserver-01
 - a. sudo su
 - b. yum update -y
 - c. yum install httpd -y
 - d. systemctl start httpd
 - e. systemctl enable httpd
 - f. cd /var/www/html
 - g. echo "REQUEST HANDLING BY SERVER 1" > index.html
2. exit
3. exit

ssh -i web-serverkey.pem ec2-user@<webserver-02 private IP>

1. Installing Apache in webserver-02
 - i. sudo su
 - ii. yum update -y
 - iii. yum install httpd -y
 - iv. systemctl start httpd
 - v. systemctl enable httpd
 - vi. cd /var/www/html
 - vii. echo "REQUEST HANDLING BY SERVER 2" > index.html
2. exit
3. exit

The screenshot shows the AWS Management Console with a terminal window open for an EC2 instance named i-038e2db81bd7d0d8c (Bastion-Server). The terminal output shows the installation of httpd and its dependencies, followed by starting the service and creating a symlink. The output is as follows:

```
Verifying : httpd-2.4.46-1.amzn2.x86_64
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-10-0-2-48 ec2-user]# systemctl start httpd
[root@ip-10-0-2-48 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-10-0-2-48 ec2-user]# cd /var/www/html
[root@ip-10-0-2-48 html]# echo "REQUEST HANDLING BY SERVER2" >index.html
[root@ip-10-0-2-48 html]# exit
exit
[ec2-user@ip-10-0-2-48 ~]$ exit
logout
Connection to 10.0.2.48 closed.
[ec2-user@ip-10-0-1-230 ~]$
```

Below the terminal output, the instance details for i-038e2db81bd7d0d8c (Bastion-Server) are shown, including Public IPs: 54.234.6.131 and Private IPs: 10.0.1.230. At the bottom, there are tabs for Webserver-Key.pem and project-linux.pem, and a Windows taskbar with the date 12-11-2020 and time 18:22.

Task-12: Access the webpage using the load balancer's DNS endpoint.

REQUEST HANDLING BY SERVER-1

The screenshot shows a web browser window with the address bar displaying 'web-app-lb-708232128.us-east-1.elb.amazonaws.com'. The page content is 'REQUEST HANDLING BY SERVER1'. The browser tabs show 'NAT gateways | VPC Management | EC2 Management Console | Instances | EC2 Management Console | i-038e2db81bd7d0d8c (Bastion-Server) | web-app-lb-708232128.us-east-1.elb.amazonaws.com'. The Windows taskbar at the bottom shows the date 12-11-2020 and time 18:23.

REQUEST HANDLING BY SERVER-2

