



# Andhra Pradesh State Skill Development Corporation



# AWS CLOUD COMPUTING

## NATGATEWAYS



## **Configuration of NAT Gateways**





## NAT Gateways

NAT is a networking technique commonly used to give an entire private network access to the internet without assigning each host a public IPv4 address. The hosts can initiate connections to the internet and receive responses, but not receive inbound connections initiated from the internet.

When a host in the private network initiates an internet-bound connection, the NAT device's public IP address becomes the source IP address for the outbound traffic. The response traffic from the internet therefore uses that public IP address as the destination IP address. The NAT device then routes the response to the host in the private network that initiated the connection.

The Networking service offers a reliable and highly available NAT solution for your VPC in the form of a NAT gateway.

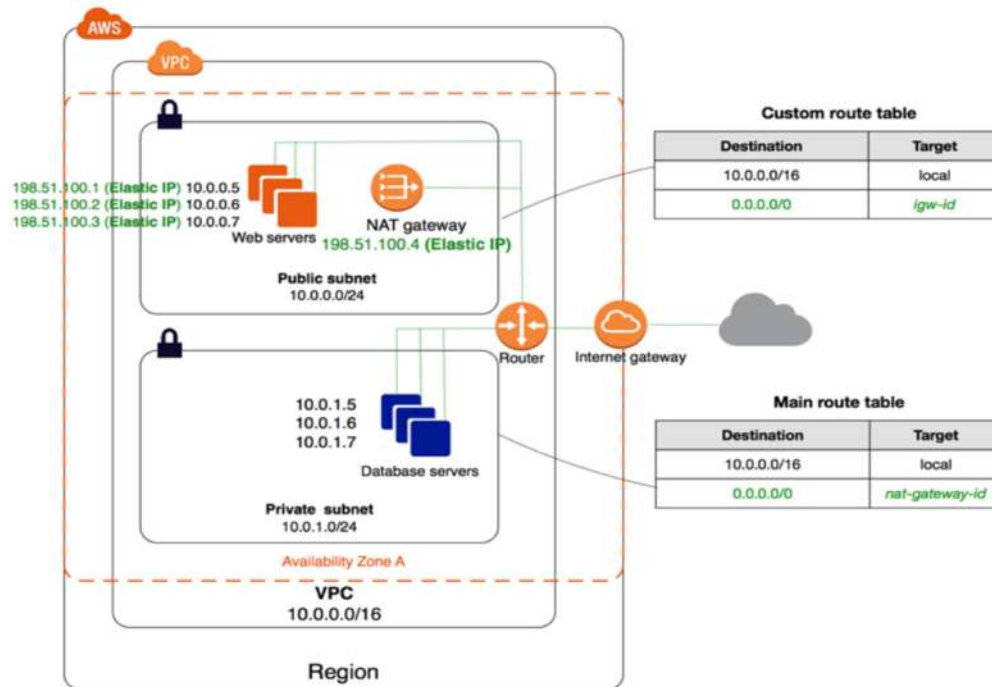
Example scenario: Imagine you have resources that need to receive inbound traffic from the internet (for example, web servers). You also have private resources that need to be protected from inbound traffic from the internet. All of these resources need to initiate connections to the internet to request software updates from sites on the internet.

You set up a VPC and add a public subnet to hold the web servers. When launching the instances, you assign public IP addresses to them so they can receive inbound internet traffic. You also add a private subnet to hold the private instances. They cannot have public IP addresses because they are in a private subnet.

You add an internet gateway to the VPC. You also add a route rule in the public subnet's route table that directs internet-bound traffic to the internet gateway. The public subnet's instances can now initiate connections to the internet and also receive inbound connections initiated from the internet. Remember that you can use security rules to control the types of traffic that are allowed in and out of the instances at the packet level.

You add a NAT gateway to the VPC. You also add a route rule in the private subnet's route table that directs internet-bound traffic to the NAT gateway. The private subnet's instances can now initiate connections to the internet. The NAT gateway allows responses, but it does not allow connections that are initiated from the internet. Without that NAT gateway, the private instances would instead need to be in the public subnet and have public IP addresses to get their software updates.

The following diagram represents the NAT Gateway connection in AWS.

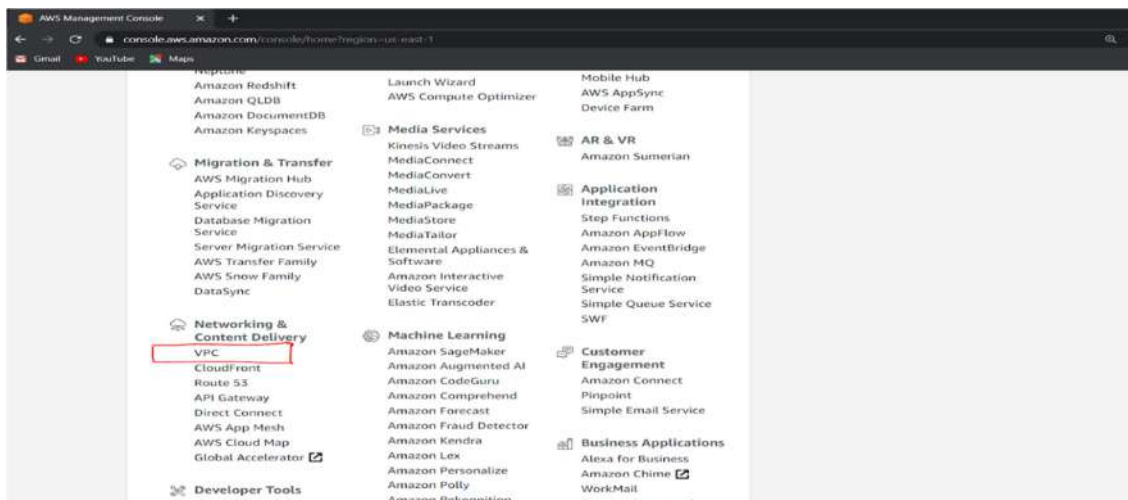


## Practical Steps:

### To Create your own VPC

Amazon VPC enables you to launch AWS resources into a virtual network that you've defined. If you have a default VPC, you can skip this section and move to the next task,

Open AWS console, Click on Services. Select Networking and Content Delivery and click on VPC



On VPC Dashboard panel

Click on Your VPC, Click on Create VPC button





New VPC Experience  
Tell us what you think

Create VPC Actions

Filter by tags and attributes or search by keyword

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR (Network Border Group)	DHCP options set	Main Route
custom_vpc	vpc-024ce986c1b44d0a	available	192.168.0...	-	-	dopt-0a3ee470	rtb-0d5c34d
vpc-29be8753	vpc-29be8753	available	172.31.0...	-	-	dopt-0a3ee470	rtb-3479f4a

Filter by VPC:  
Select a VPC

VIRTUAL PRIVATE CLOUD

Your VPCs

Subnets

Route Tables

Internet Gateways

Egress Only Internet Gateways

Carrier Gateways

DHCP Options Sets

Elastic IPs

Managed Prefix Lists

Endpoints

Endpoint Services

NAT Gateways

Peering Connections

On “Create VPC”, page  
For Name tag → Example\_VPC  
For IPv4 CIDR block → 192.168.0.0/16  
Click on “Create” button

VPCs > Create VPC

### Create VPC

A VPC is an isolated portion of the AWS cloud populated by AWS objects, such as Amazon EC2 instances. You must specify an IPv4 address range for your VPC. Specify the IPv4 address range as a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. You cannot specify an IPv4 CIDR block larger than /16. You can optionally associate an IPv6 CIDR block with the VPC.

Name tag

IPv4 CIDR block\*

IPv6 CIDR block ☒ No IPv6 CIDR Block ☐ Amazon provided IPv6 CIDR block ☐ IPv6 CIDR owned by me

Tenancy Default

\* Required

Cancel Create

Verify Example\_VPC was created

aws Services Resource Groups

New VPC Experience  
Tell us what you think

Create VPC Actions

Filter by tags and attributes or search by keyword

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	IPv6 CIDR (Network Border Group)	DHCP options set
custom_vpc	vpc-024ce986c1b44d0a	available	192.168.0...	-	-	dopt-0a3ee470
Example_V...	vpc-07340cf7e97a5edc7	available	192.168.0...	-	-	dopt-0a3ee470
vpc-29be8753	vpc-29be8753	available	172.31.0...	-	-	dopt-0a3ee470

VPC: vpc-07340cf7e97a5edc7

Description CIDR Blocks Flow Logs Tags

VPC ID	State	Tenancy
vpc-07340cf7e97a5edc7	available	default
IPv4 CIDR	192.168.0.0/16	Default VPC No
		Classic link Disabled



## To create public subnet

Click on Subnet. Click on Create Subnet button



Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
private_sub...	subnet-007205657302a155	available	vpc-024ce9686c1b44d0a...	192.168.2.0/24	251	-	us-east-
public_subnet	subnet-087748441c995e036	available	vpc-024ce9686c1b44d0a...	192.168.1.0/24	250	-	us-east-
subnet-107d9076	subnet-107d9076	available	vpc-29be8753	172.31.0.0/20	4091	-	us-east-
subnet-3d8331e	subnet-3d8331e	available	vpc-29be8753	172.31.80.0/20	4091	-	us-east-
subnet-5395f5d	subnet-5395f5d	available	vpc-29be8753	172.31.64.0/20	4091	-	us-east-
subnet-6baa1626	subnet-6baa1626	available	vpc-29be8753	172.31.16.0/20	4091	-	us-east-
subnet-890a3ab7	subnet-890a3ab7	available	vpc-29be8753	172.31.48.0/20	4090	-	us-east-
subnet-f7e20ba8	subnet-f7e20ba8	available	vpc-29be8753	172.31.32.0/20	4090	-	us-east-

On Create Subnet, page

For Name tag → Example\_pub\_sub

For VPC → Example\_VPC

For IPv4 CIDR block → 192.168.10.0/24

Click on Create button

Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag:

VPC:

Availability Zone:

VPC CIDRs	CIDR	Status	Status Reason
	192.168.0.0/16	associated	

IPv4 CIDR block:

\* Required

[Cancel](#) [Create](#)

Verify Example\_pub\_subnet got created

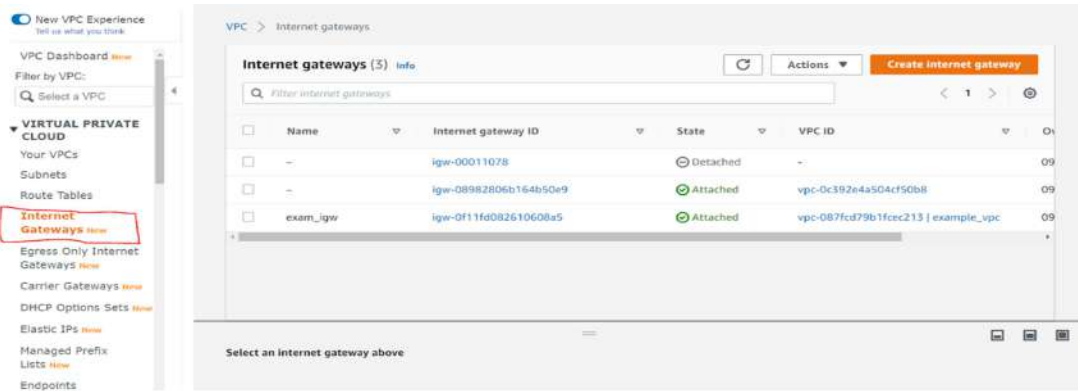
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 CIDR	Availability Zone
Example_p...	subnet-09392dc8f3064d6dc	available	vpc-07340cf7e97a5edc7	192.168.10.0/24	251	-	us-east-

Subnet: subnet-09392dc8f3064d6dc

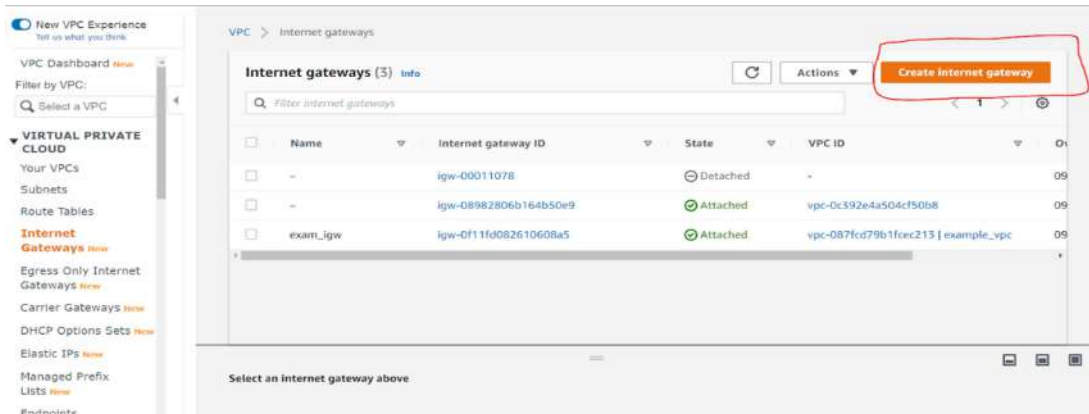
Description: Subnet ID: subnet-09392dc8f3064d6dc, VPC: vpc-07340cf7e97a5edc7 | Example\_VPC, Available IPv4 Addresses: 251, State: available, IPv4 CIDR: 192.168.10.0/24, IPv6 CIDR: -

## Create Internet gateway and attach to your VPC

In VPC Dashboard panel, Click on Internet Gateways

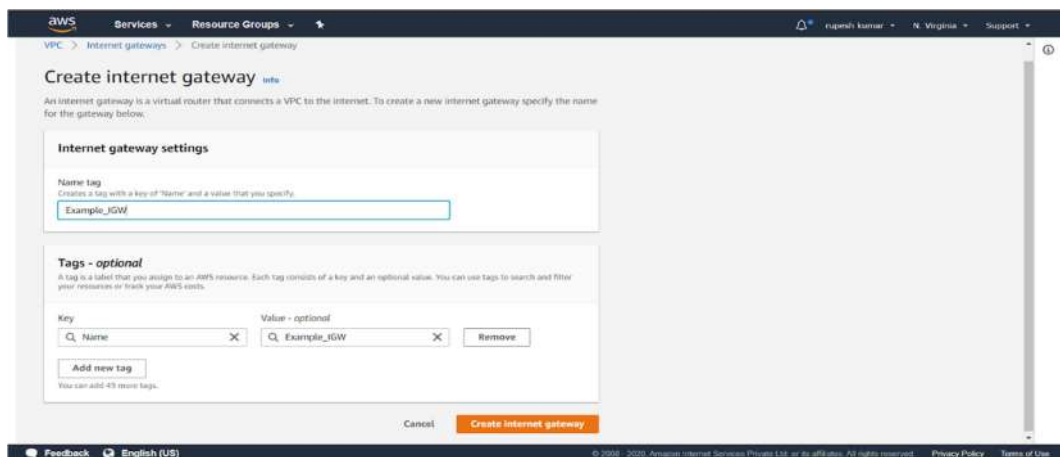


Click on Create Internet Gateway button

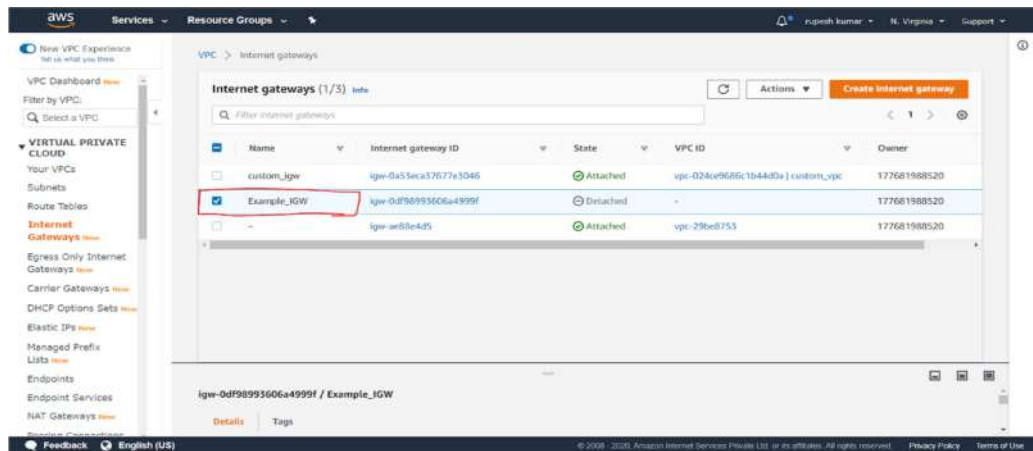


In Create Internet Gateway, box

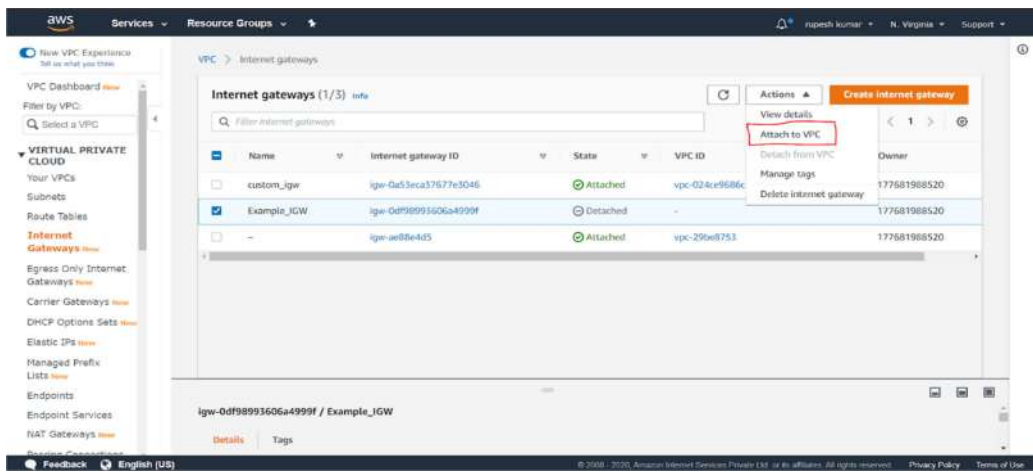
For Name tag → Example\_IGW, Click on “Create internet gateway” button



Verify Internet gateway is created

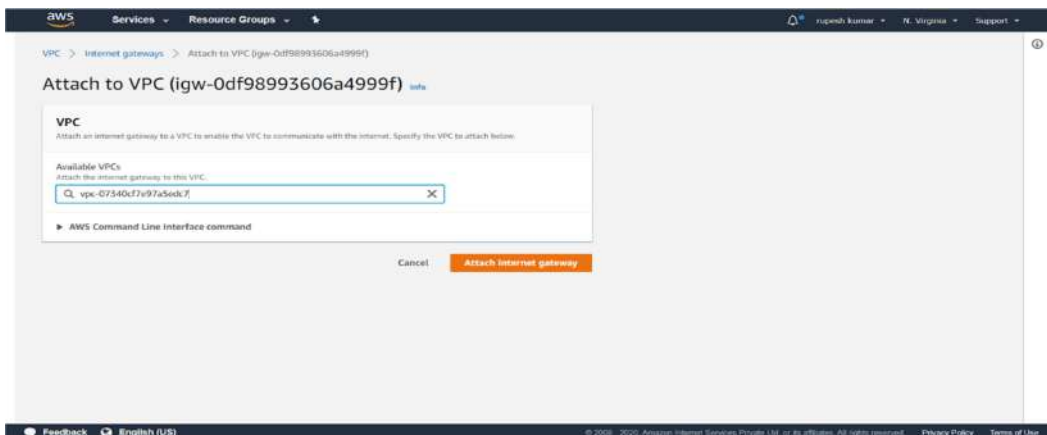


Select Example\_IGW  
Click on actions and select option “Attach to VPC”



In “Attach to VPC” box

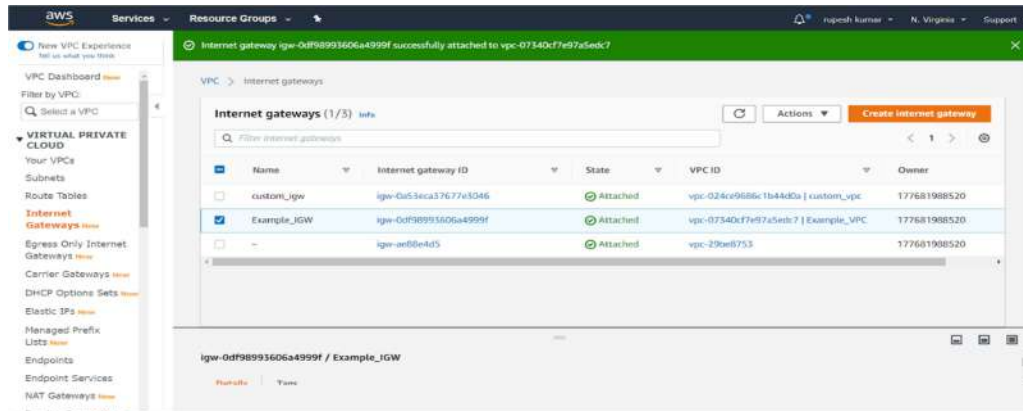
For VPC ☐ Example\_VPC  
Click on “Attach internet gateway” button





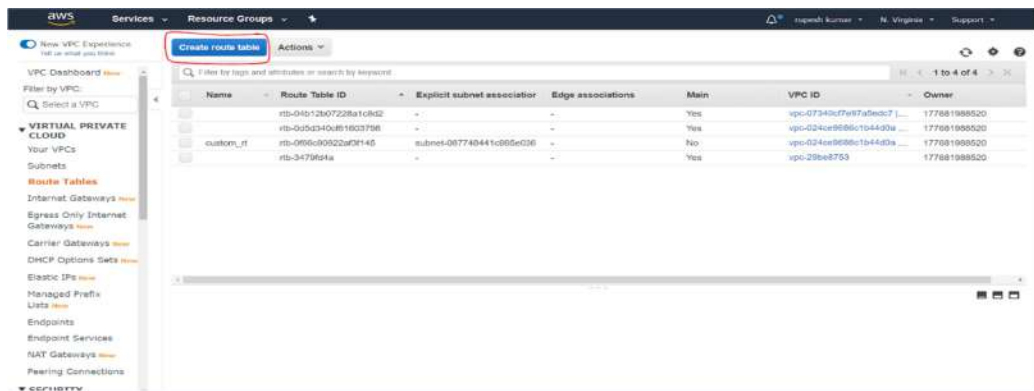


Verify Internet gateway is connected to your VPC

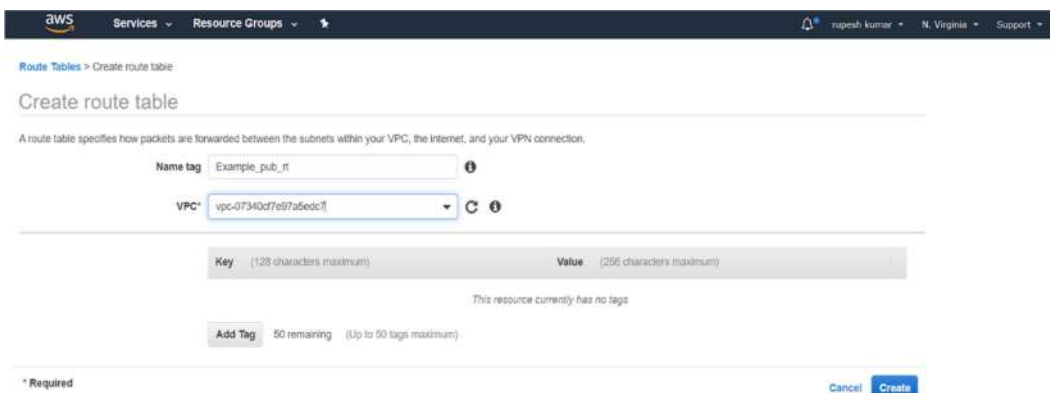


Create Public Routing Table, associate subnet and add routing rules

On VPC Dashboard panel, Click on Route Table then Click on “Create Route Table” button



On “Create Route Table” box  
For Name tag → Example\_pub\_rt  
For VPC → Example\_VPC  
Click on “Create” button





Verify Example\_pub\_rt table is created

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
Example_pub_rt	rtb-01a2e79e45c259516	-	-	No	vpc-07340cd7e97a5edc7	1776819885
rtb-04b12b07228a1c8d2	rtb-04b12b07228a1c8d2	-	-	Yes	vpc-07340cd7e97a5edc7	1776819885
rtb-0d5d340d61603798	rtb-0d5d340d61603798	-	-	Yes	vpc-024ce9686c1b44d3a	1776819885
custom_rt	rtb-086c90822af3f145	subnet-087748441c995e036	-	No	vpc-024ce9686c1b44d3a	1776819885
rtb-3479894a	rtb-3479894a	-	-	Yes	vpc-29be8753	1776819885

Click on “Subnet Association” button

Route Table: rtb-01a2e79e45c259516

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID: rtb-01a2e79e45c259516

Explicitly Associated with: -

Owner: 177681988520

Main: No

VPC: vpc-07340cd7e97a5edc7 (Example\_VPC)

Click on Edit subnet association button

Route Table: rtb-01a2e79e45c259516

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
You do not have any subnet associations.		

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:



Select checkbox of Example\_pub\_sub → 192.168.10.0/24  
Click on save button



**Route Tables** > Edit subnet associations

Edit subnet associations

Route table: rtb-01a2e79e45c259516 (Example\_pub\_rt)

Associated subnets: subnet-09392dc8f3064d5dc

Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-0514b442038ab700f (Example_priv_sub)	192.168.20.0/24	-	Main
<input checked="" type="checkbox"/> subnet-09392dc8f3064d5dc (Example_pub_subnet)	192.168.10.0/24	-	Main

\* Required

Cancel Save

Verify Example\_pub\_subnet is associated with routing table

**Route Tables**

Name	Route Table ID	Explicit subnet associations	Edge associations	Main	VPC ID
<input checked="" type="checkbox"/> Example_pub_rt	rtb-01a2e79e45c259516	subnet-09392dc8f3064d5dc	-	No	vpc-07340cf7e97a5edc7
<input type="checkbox"/> rtb-04b12b07228a1cb2	rtb-04b12b07228a1cb2	-	-	Yes	vpc-07340cf7e97a5edc7
<input type="checkbox"/> rtb-0d93340c81603798	rtb-0d93340c81603798	-	-	Yes	vpc-024ce9686c1b44d3a
<input type="checkbox"/> custom_rt	rtb-096c90622a3f145	subnet-087748441c995e036	-	No	vpc-024ce9686c1b44d3a
<input type="checkbox"/> rtb-347983a	rtb-347983a	-	-	Yes	vpc-295e8f753

Route Table: rtb-01a2e79e45c259516

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit subnet associations

Subnet ID	IPv4 CIDR	IPv6 CIDR
subnet-09392dc8f3064d5dc	192.168.10.0/24	-

The following subnets have not been explicitly associated with any route tables and are therefore associated with the main route table:

Click on Route button and Click on Edit button

**Route Tables**

Name	Route Table ID	Explicit subnet associations	Edge associations	Main	VPC ID
<input checked="" type="checkbox"/> Example_pub_rt	rtb-01a2e79e45c259516	subnet-09392dc8f3064d5dc	-	No	vpc-07340cf7e97a5edc7
<input type="checkbox"/> rtb-04b12b07228a1cb2	rtb-04b12b07228a1cb2	-	-	Yes	vpc-07340cf7e97a5edc7
<input type="checkbox"/> rtb-0d93340c81603798	rtb-0d93340c81603798	-	-	Yes	vpc-024ce9686c1b44d3a
<input type="checkbox"/> custom_rt	rtb-096c90622a3f145	subnet-087748441c995e036	-	No	vpc-024ce9686c1b44d3a
<input type="checkbox"/> rtb-347983a	rtb-347983a	-	-	Yes	vpc-295e8f753

Route Table: rtb-01a2e79e45c259516

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Edit routes

View: All routes

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No





Click on “Add route” button

The screenshot shows the AWS 'Edit routes' interface. A table lists existing routes with columns: Destination, Target, Status, and Propagated. The first row shows a route to 192.168.0.0/16 via a local target, which is active. Below the table, the 'Add route' button is highlighted with a red rectangle. At the bottom right, there are 'Cancel' and 'Save routes' buttons.

For Destination → 0.0.0.0/0

For Target → select Example\_IGW

Click on Save button

This screenshot shows the same 'Edit routes' page after a new route has been added. The table now has two rows: the original route to 192.168.0.0/16 and a new route to 0.0.0.0/0 via the target 'igw-0d98863006a4939f'. The new route is highlighted with a blue selection bar. The 'Add route' button is no longer visible. The 'Save routes' button is at the bottom right.

Verification

Public route is added through internet gateway

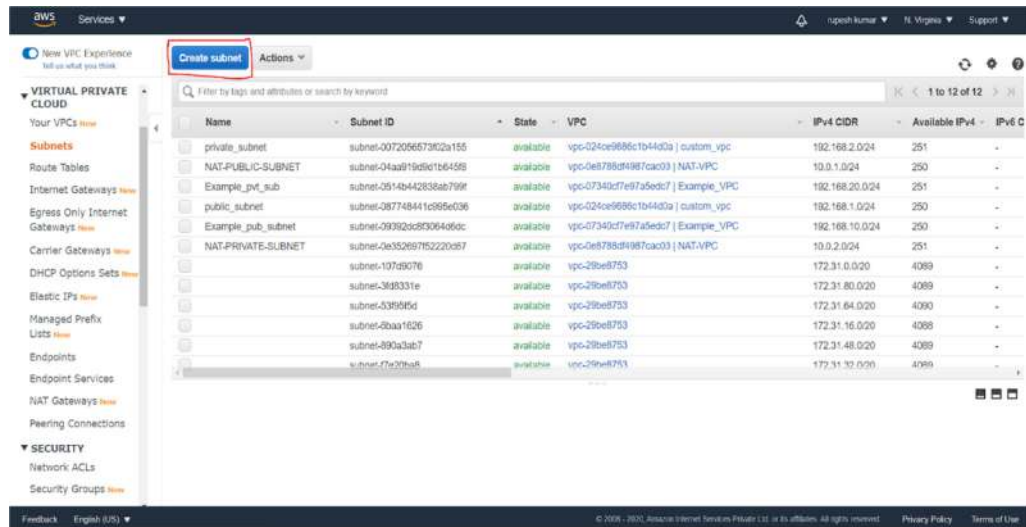
Verify Status column show Active

This block contains two screenshots from the AWS VPC console. The top screenshot shows the 'Route Tables' list in the left-hand navigation pane. The bottom screenshot shows the 'Routes' tab for a specific route table (rft-01a2e79e45c259516). It displays a table of routes with columns: Destination, Target, Status, and Propagated. The route to 0.0.0.0/0 via 'igw-0d98863006a4939f' is highlighted with a red box around the 'active' status.



## Create NAT Subnet

Click on create subnet



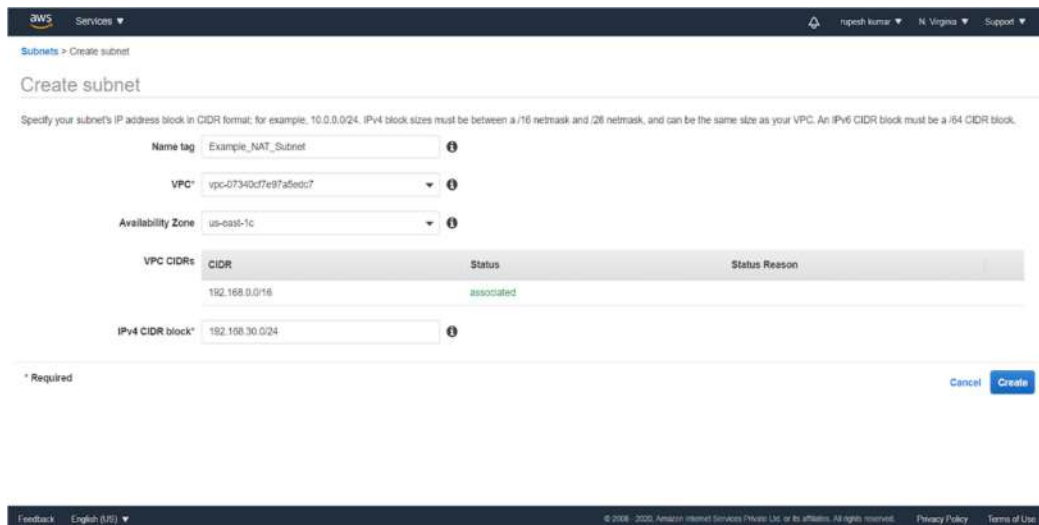
Name	Subnet ID	State	VPC	IPv4 CIDR	Available IPv4	IPv6 C
private_subnet	subnet-00720667302a155	available	vpc-024ce9686c1b4403a   custom_vpc	192.168.2.0/24	251	-
NAT-PUBLIC-SUBNET	subnet-04aa919d9d1b64598	available	vpc-0e8788d4987cac03   NAT-VPC	10.0.1.0/24	250	-
Example_pvt_sub	subnet-0514b442638ab799f	available	vpc-07340cd7e97a5edc7   Example_VPC	192.168.20.0/24	251	-
public_subnet	subnet-087748441c995e036	available	vpc-024ce9686c1b4403a   custom_vpc	192.168.1.0/24	250	-
Example_pub_subnet	subnet-09362cd803064a6dc	available	vpc-07340cd7e97a5edc7   Example_VPC	192.168.10.0/24	250	-
NAT-PRIVATE-SUBNET	subnet-0a352697852220367	available	vpc-0e8788d4987cac03   NAT-VPC	10.0.2.0/24	251	-
	subnet-107d9076	available	vpc-29be8753	172.31.0.0/20	4089	-
	subnet-368331e	available	vpc-29be8753	172.31.80.0/20	4089	-
	subnet-335695d	available	vpc-29be8753	172.31.64.0/20	4090	-
	subnet-6baa1626	available	vpc-29be8753	172.31.16.0/20	4088	-
	subnet-890a3ab7	available	vpc-29be8753	172.31.48.0/20	4089	-
	subnet-f7a20ba8	available	vpc-29be8753	172.31.32.0/20	4089	-

For the subnet details:

Name tag → Example\_NAT\_subnet

VPC → Example\_VPC

IPv4 CIDR Block → 192.168.30/24



Create subnet

Specify your subnet's IP address block in CIDR format; for example, 10.0.0.0/24. IPv4 block sizes must be between a /16 netmask and /28 netmask, and can be the same size as your VPC. An IPv6 CIDR block must be a /64 CIDR block.

Name tag: Example\_NAT\_Subnet

VPC: vpc-07340cd7e97a5edc7

Availability Zone: us-east-1c

VPC CIDR	Status	Status Reason
192.168.0.0/16	associated	

IPv4 CIDR block: 192.168.30.0/24

\* Required

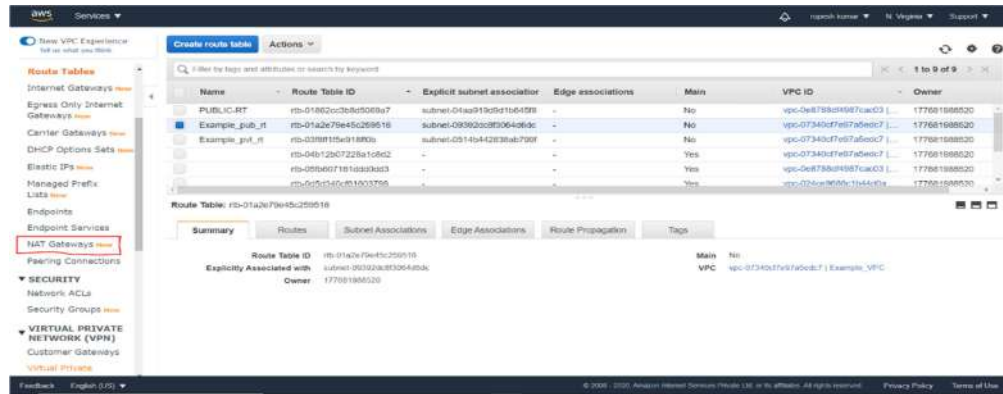
Cancel Create



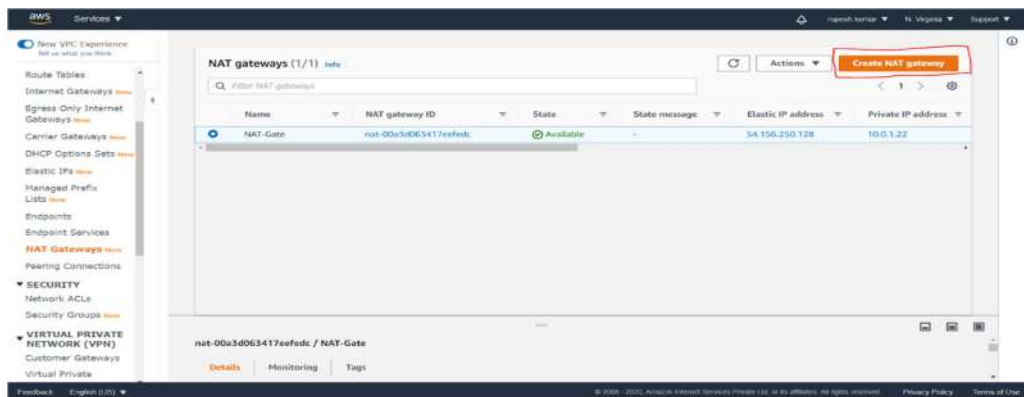
Now click on create and then NAT subnet was created

## Create NAT Gateway

Click on NAT Gateway in the VPC dashboard

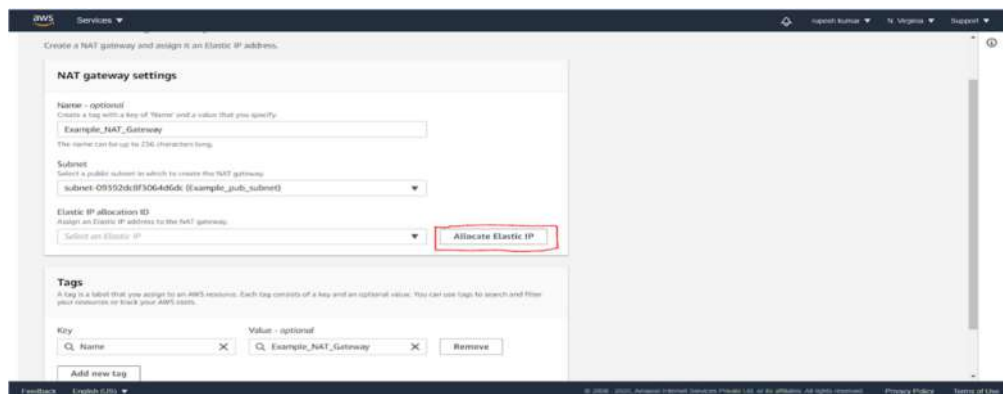


Now Click on create NAT Gateway



In the NAT Gateway settings

- Name → Example\_NAT\_Gateway
- Subnet → select the public subnet from the Example\_VPC
- For the Elastic IP → Click on Allocate Elastic IP



Now elastic ip got created and then click on the create NAT Gateway

Now NAT Gateway was created

Name	NAT gateway ID	State	State message	Elastic IP address	Private IP address
Example_NAT_Gateway	nat-0bc5e183faffbee54	Pending	-	-	192.168.10.191
NAT-Gate	nat-00a3063477ee6dc	Available	-	54.156.250.128	10.0.1.32

## Create NAT Route Table

From the VPC dashboard console click on the route table

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
PUBLIC-RT	rtb-01862cc3b8d509ba7	subnet-01a2e71e41c256516	-	No	vpc-6e8789d4987ca033	1776810985020
Example_pub_rt	rtb-01a2e71e41c256516	subnet-09592ab3f06405dc	-	No	vpc-67340cd7e97a5edc7	1776810985020
Example_priv_rt	rtb-0388115e918806	subnet-0514b42833ab799f	-	No	vpc-67340cd7e97a5edc7	1776810985020
	rtb-04b1260722ba1a8d2	-	-	Yes	vpc-6e8789d4987ca033	1776810985020
	rtb-05b607161a830a33	-	-	Yes	vpc-6e8789d4987ca033	1776810985020
	rtb-0a9f34d581a03798	-	-	Yes	vpc-67340cd7e97a5edc7	1776810985020

Now click on the create route table

For the route table details:

Name tag → NAT\_RT

VPC → Example\_VPC



**AWS** Services ▾

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 NAT\_RT ⓘ

VPC vpc-07340cfe97a5edc7f ⓘ

Key (128 characters maximum) Value (256 characters maximum)

This resource currently has no tags

Add Tag 50 remaining (Up to 50 tags maximum)

\* Required

Cancel Create

Feedback English (US) © 2018 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on create, route table was created

**AWS** Services ▾

New VPC Experience Tell us what you think

**VIRTUAL PRIVATE CLOUD**

Your VPCs **Route Tables** Subnets Internet Gateways Egress Only Internet Gateways Carrier Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists Endpoints Endpoint Services NAT Gateways Peering Connections

**SECURITY**

Network ACLs Security Groups

Create route table Actions ▾

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
NAT_RT	rtb-013d2efcb67862414	-	-	No	vpc-07340cfe97a5edc7f	177681988520
PUBLIC_RT	rtb-01682cc3b0d5089a7	subnet-04aa01bd9d1b6409b	-	No	vpc-0e6789d94987cac03	177681988520
Example_pub_rt	rtb-01a2e79e45c256516	subnet-09392dd0f306499dc	-	No	vpc-07340cfe97a5edc7f	177681988520
Example_priv_rt	rtb-0388f15e91880b	subnet-0514b44283fab799f	-	No	vpc-07340cfe97a5edc7f	177681988520
	rtb-04b12b07229a1cd82	-	-	Yes	vpc-07340cfe97a5edc7f	177681988520
	rtb-05b607161a5d5933	-	-	Yes	vpc-0e6789d94987cac03	177681988520

Route Table: rtb-013d2efcb67862414

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID: rtb-013d2efcb67862414

Explicitly Associated with: Owner: 177681988520

Main: No

VPC: vpc-07340cfe97a5edc7f | Example\_VPC

Feedback English (US) © 2018 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on subnet association

**AWS** Services ▾

New VPC Experience Tell us what you think

**VIRTUAL PRIVATE CLOUD**

Your VPCs **Route Tables** Subnets Internet Gateways Egress Only Internet Gateways Carrier Gateways DHCP Options Sets Elastic IPs Managed Prefix Lists Endpoints Endpoint Services NAT Gateways Peering Connections

**SECURITY**

Network ACLs Security Groups

Create route table Actions ▾

Filter by tags and attributes or search by keyword

Name	Route Table ID	Explicit subnet association	Edge associations	Main	VPC ID	Owner
NAT_RT	rtb-013d2efcb67862414	-	-	No	vpc-07340cfe97a5edc7f	177681988520
PUBLIC_RT	rtb-01682cc3b0d5089a7	subnet-04aa01bd9d1b6409b	-	No	vpc-0e6789d94987cac03	177681988520
Example_pub_rt	rtb-01a2e79e45c256516	subnet-09392dd0f306499dc	-	No	vpc-07340cfe97a5edc7f	177681988520
Example_priv_rt	rtb-0388f15e91880b	subnet-0514b44283fab799f	-	No	vpc-07340cfe97a5edc7f	177681988520
	rtb-04b12b07229a1cd82	-	-	Yes	vpc-07340cfe97a5edc7f	177681988520
	rtb-05b607161a5d5933	-	-	Yes	vpc-0e6789d94987cac03	177681988520

Route Table: rtb-013d2efcb67862414

Summary Routes Subnet Associations Edge Associations Route Propagation Tags

Route Table ID: rtb-013d2efcb67862414

Explicitly Associated with: Owner: 177681988520

Main: No

VPC: vpc-07340cfe97a5edc7f | Example\_VPC

Feedback English (US) © 2018 Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

Click on edit subnet association

Select the Example\_NAT\_subnet and click on Save



Subnet ID	IPv4 CIDR	IPv6 CIDR	Current Route Table
subnet-093920a2806405dc   Example_pub_subnet	192.168.10.0/24	-	rtb-01a2e79e45c259516
subnet-0514b4283ab799f   Example_priv_subnet	192.168.20.0/24	-	rtb-038f15e918f5b
subnet-04aa1a0a2f1aa70e   Example_NAT_Subnet	192.168.30.0/24	-	Main

Now click on routes and add the NAT Gateway

Click on edit routes and add the route

For the routes details:

Destination → 0.0.0.0/0

Target → NAT Gateway and select the Example\_NAT\_Gateway

Now the routes were added successfully.

Destination	Target	Status	Propagated
192.168.0.0/16	local	active	No
0.0.0.0/0	nw-0bc3e183fa8f5e54	active	No

## Verify NAT Gateway Configuration

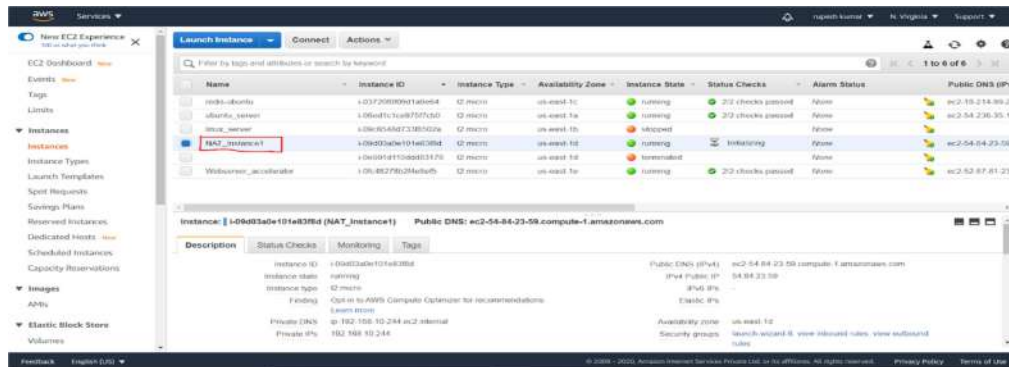
To check the NAT Gateway connection, create two instances.

Create the first instance (NAT\_Instance1):

VPC → Example\_VPC

Subnet → Example\_pub\_subnet

Name Tag → NAT\_Instance1

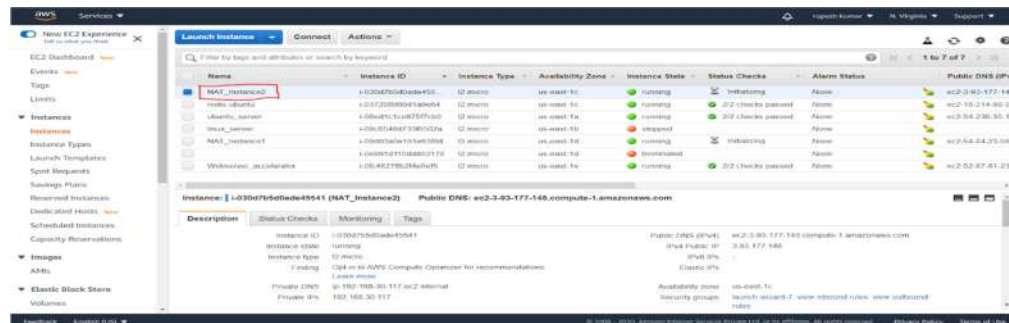


Create the second instance (NAT\_Instance2):

VPC → Example\_VPC

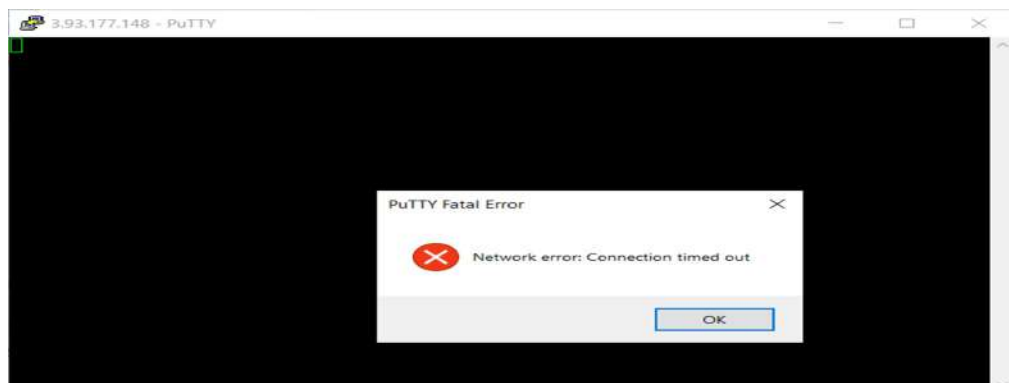
Subnet → Example\_NAT\_subnet

Name Tag → NAT\_Instance2



Now connect through NAT\_Instance2 using putty, it should be failed.

Here the NAT Gateway shouldn't allow the direct connection to the instance.



If you want to connect to NAT\_Instance2, first connect to the public instance and then connect to the instance which is in NAT Gateway.

Now connect to the NAT\_Instance1 using putty.  
Instance1 was connected successfully

```
ec2-user@ip-192-168-10-244:~$ ssh -i /home/ec2-user/.ssh/instance1-key.pem ec2-user@192.168.30.117
login as: ec2-user
Authenticating with public key "imported-openssh-key"

      _ _ _ _ _
     /   /   /   \
    /___/___/___\  Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-10-244 ~]$
```

Now check if the instance is working or not using the private ip address of the instance2.

```
root@ip-192-168-10-244:/home/ec2-user$ ssh -i /home/ec2-user/.ssh/instance1-key.pem ec2-user@192.168.30.117
      _ _ _ _ _
     /   /   /   \
    /___/___/___\  Amazon Linux AMI

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-10-244 ~]$ sudo su
[root@ip-192-168-10-244 ec2-user]# ping 192.168.30.117
PING 192.168.30.117 (192.168.30.117) 56(84) bytes of data.
 64 bytes from 192.168.30.117: icmp_seq=1 ttl=255 time=1.51 ms
 64 bytes from 192.168.30.117: icmp_seq=2 ttl=255 time=1.55 ms
 64 bytes from 192.168.30.117: icmp_seq=3 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=4 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=5 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=6 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=7 ttl=255 time=1.47 ms
 64 bytes from 192.168.30.117: icmp_seq=8 ttl=255 time=1.52 ms
 64 bytes from 192.168.30.117: icmp_seq=9 ttl=255 time=1.53 ms
 64 bytes from 192.168.30.117: icmp_seq=10 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=11 ttl=255 time=1.50 ms
 64 bytes from 192.168.30.117: icmp_seq=12 ttl=255 time=1.54 ms
 64 bytes from 192.168.30.117: icmp_seq=13 ttl=255 time=1.50 ms
```

The results show that the instance2 is working properly.

Now connect to the instance2 in NAT subnet using the private ip, the command is  
Ssh ec2-user@192.168.30.117.

```
root@ip-192-168-10-244:/home/ec2-user$ ssh -i /home/ec2-user/.ssh/instance1-key.pem ec2-user@192.168.30.117
[ec2-user@ip-192-168-10-244 ~]$ ssh ec2-user@192.168.30.117
The authenticity of host '192.168.30.117 (192.168.30.117)' can't be established.
ECDSA key fingerprint is SHA256:XLXSkORrs2nSKwckc3GjaWtIoICopGXX5GSgUTN1c/Y.
ECDSA key fingerprint is MD5:30:d9:ab:ff:5b:77:e5:0c:ab:ef:9f:f6:80:c3:8e:25.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.30.117' (ECDSA) to the list of known hosts.
Permission denied (publickey).
[ec2-user@ip-192-168-10-244 ~]$
```

It gives the error message, now add the key pair related to the NAT subnet instance and add the content of the keypair.  
nano example.pem

```
root@ip-192-168-10-244:/home/ec2-user
GNU nano 2.5.3 File: example.pem Modified
DrqDz3PJejLo9qm8u0CQnab0LqkCgYEA6NppDG2vYoL++uDjEObwFA1G0qIOEv7QdEOJ/MMv+vdm
1kjd3Dh9kdYkl7wjAvWlnznVp2/VeIFUREzDGX1ke4tLyjkz8TLL901UUQ2KuFssuxDRABtPpkYq
CQKs4UhdRYjtlwxHFGy+cvmy8J+Ergptuwz7hIIkTAnbNH7jgUCgYEAAnvYTjiVQJwgGVoh+a+ly
j8YZoQBbX7vUn4p6YSpwXGRKefSgQVVCQjXpb8NtRTH1By6gYonzw2sn+gpBNP83P7RLm+clMn
+81M1YpmijZxc2633NUiegsegmeStZgk9/zRs2+j/Cv95Cd80Y2NM5KmsjMN/KSuPk1V9ozPfsMC
gYAFta1Tv7DIQpWl/M20kWUbqMOu0Ih10Me9whY1G3gmuEBMuO3iQ4RYuh6F1fhpzyozgFCL3YMn
hExtTrGowSri7Cxd9g8e//beZm7p2eIn6RxsrdniJ/ypABlxxR4GHCaCivxocsWfa5cz7ImFuvOe
7OSJ7JfUo56QKNUepvt7HQKBgE+m0F+utL4KGFrcqoljiTiRayGbTbfXHgaA73FypzknkTo/n+Lge
SNmxVgPEtjanJ8sNlrcLRPmdS28f4pe0kaBU4ngqdpKETolG8vQPuJfS9ga4X2sJZVawGCaRzYKu
7P9SMKWv8ESgqoTtgIjkw/eWqTqO4JhS0HineKq1HifPAoGAE4GjkOhkrml6Ajtoq8yFeifPWA3S
Ry2ITBnPyKCz7bn9DcsFvj8YfD1xb4dQmBsdU1lwUU4OG7zoR269bp1VYBuuJi8LZaj8hkHmeA03
h/zvnab6SfmbZ3SNnFYPHD3uhVi8yQ7aFZBjaVd2aSR2KYwgGqoa2t5Oc94T0h91Nak=
-----END RSA PRIVATE KEY-----

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Now run the command to connect to the instance in the NAT subnet

```
root@ip-192-168-10-244:/home/ec2-user
[root@ip-192-168-10-244 ec2-user]# nano example.pem
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117
Warning: UNPROTECTED PRIVATE KEY FILE!
Permissions 0644 for 'example.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "example.pem": bad permissions
Permission denied (publickey).
[root@ip-192-168-10-244 ec2-user]#
```

Now change the permissions using the command

Chmod 400 example.pem

And again, try to connect the NAT instance

```
ec2-user@ip-192-168-30-117:~
[root@ip-192-168-10-244 ec2-user]# chmod 400 example.pem
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117
Amazon Linux AMI
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
No packages needed for security; 2 packages available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-192-168-30-117 ~]$
```



After running the command immediately, the prompt was changed from instance1 to instance2.

```
ec2-user@ip-192-168-30-117:~$  
[root@ip-192-168-10-244 ec2-user]# chmod 400 example.pem  
[root@ip-192-168-10-244 ec2-user]# ssh -i example.pem ec2-user@192.168.30.117  
  
  _I_ ( _I_ )  
 _I_ \ _I_ _I_   Amazon Linux AMI  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
No packages needed for security; 2 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-192-168-30-117 ~]$
```

Check if the instance2 is working properly.

```
ec2-user@ip-192-168-30-117:~$  
  _I_ \ _I_ _I_   Amazon Linux AMI  
  
https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/  
No packages needed for security; 2 packages available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-192-168-30-117 ~]$ ping google.com  
PING google.com (172.217.164.142) 56(84) bytes of data:  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=1 ttl=111 ti  
me=2.62 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=2 ttl=111 ti  
me=2.18 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=3 ttl=111 ti  
me=2.24 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=4 ttl=111 ti  
me=2.15 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=5 ttl=111 ti  
me=2.14 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=6 ttl=111 ti  
me=6.01 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=7 ttl=111 ti  
me=2.20 ms  
64 bytes from iad30s24-in-f14.1e100.net (172.217.164.142): icmp_seq=8 ttl=111 ti  
me=2.26 ms
```

The results show that the NAT Gateway connections were working properly.