

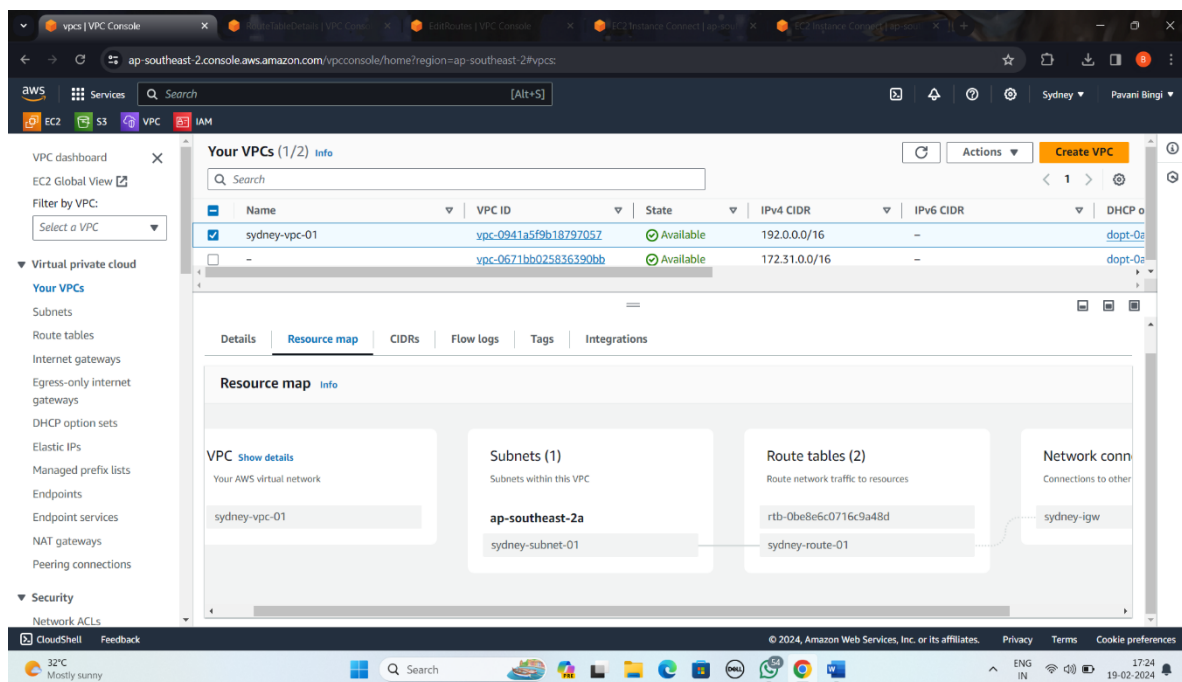
NAME:B.Pavani

MBL :7569217017

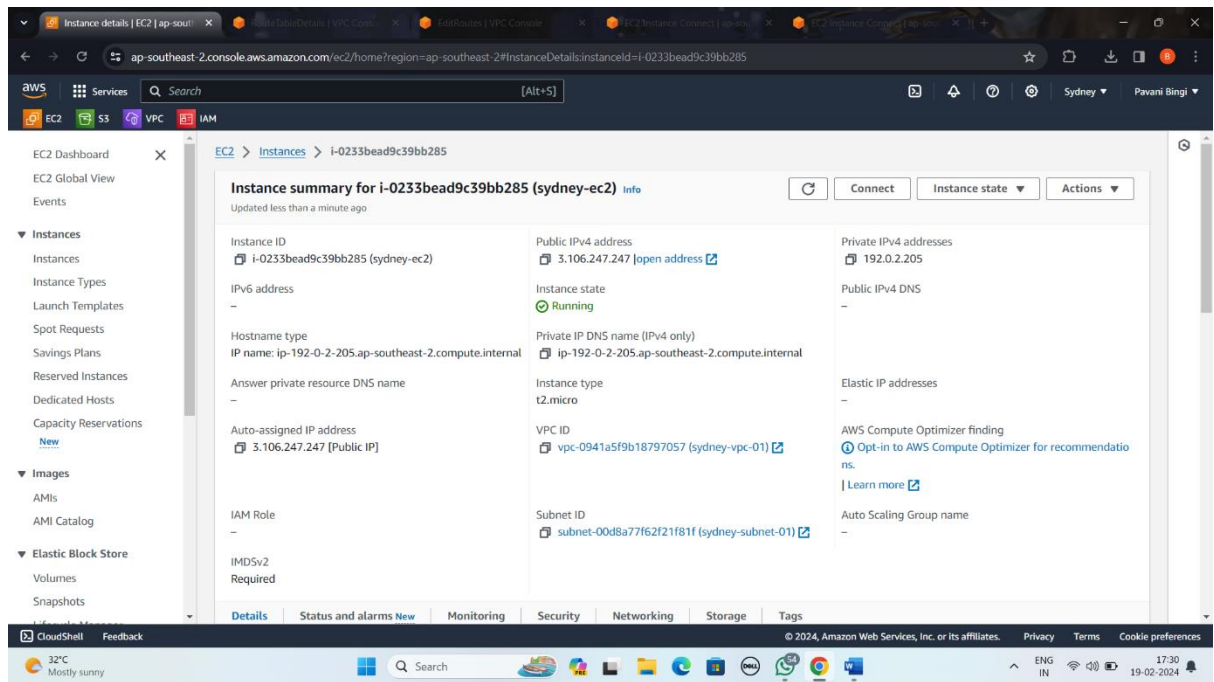
E-MAIL:bingipavani3432gmail.com

PEERING ASSIGNMENT

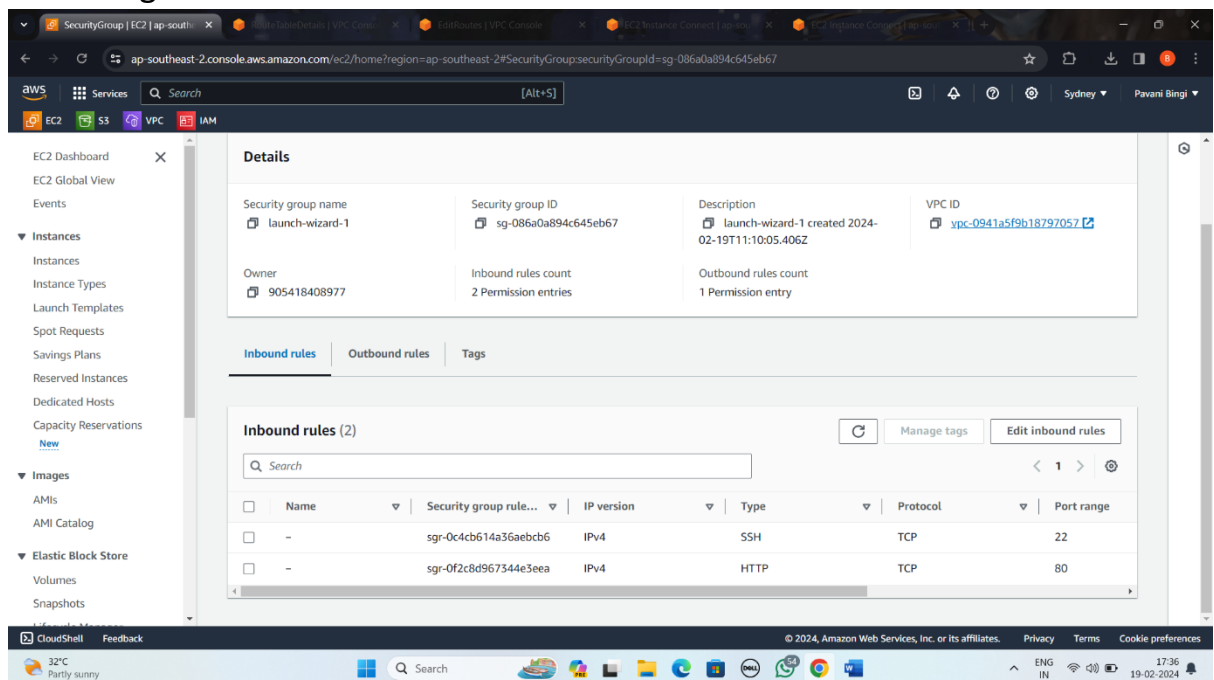
- Create a VPC in Sydney region naming Sydney-vpc and attach subnet,internet gateway and route table



- Create Ec2 instance in the same region by name Sydney-ec2

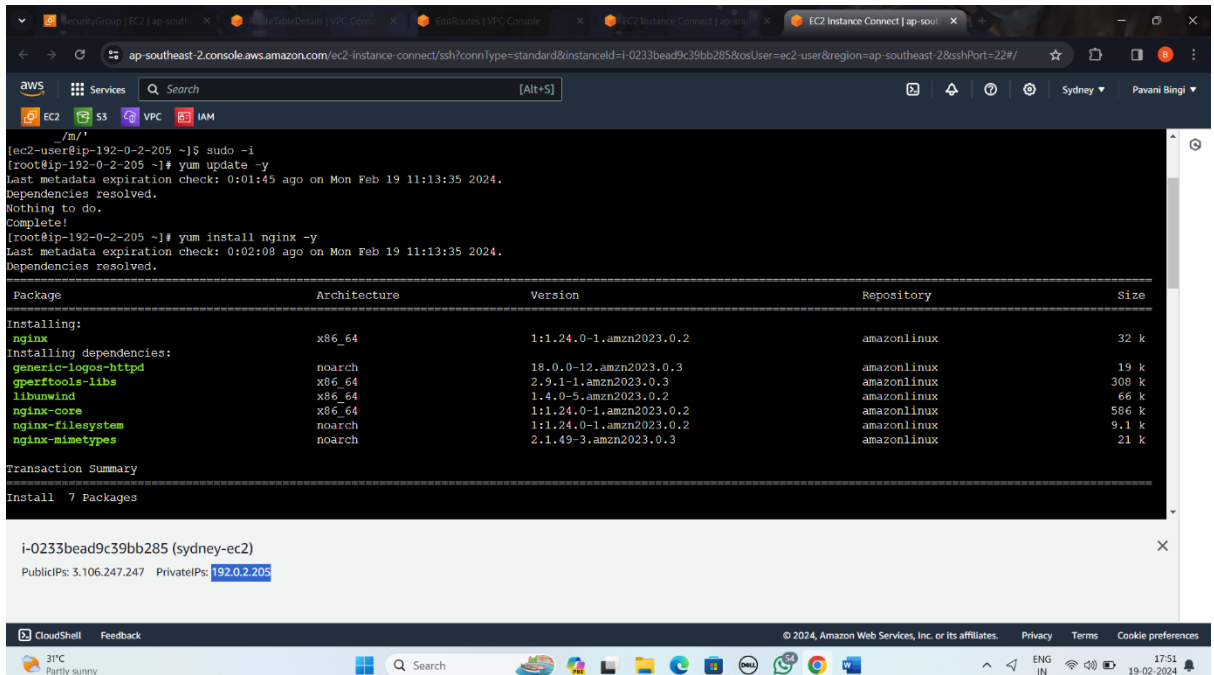


- enter into security groups edit inbound rules >>http and source save the changes



- connect to the CLI of the instance and connect to the root user, update the application packages. Install nginx for checking peering connection between vpc. use the command cd to change directory to default address of nginx and use ls to list all files and directory then remove the index.html file. create new file with name

index.html by adding the wanted data . check weather the server is active or dead by using the basic commands



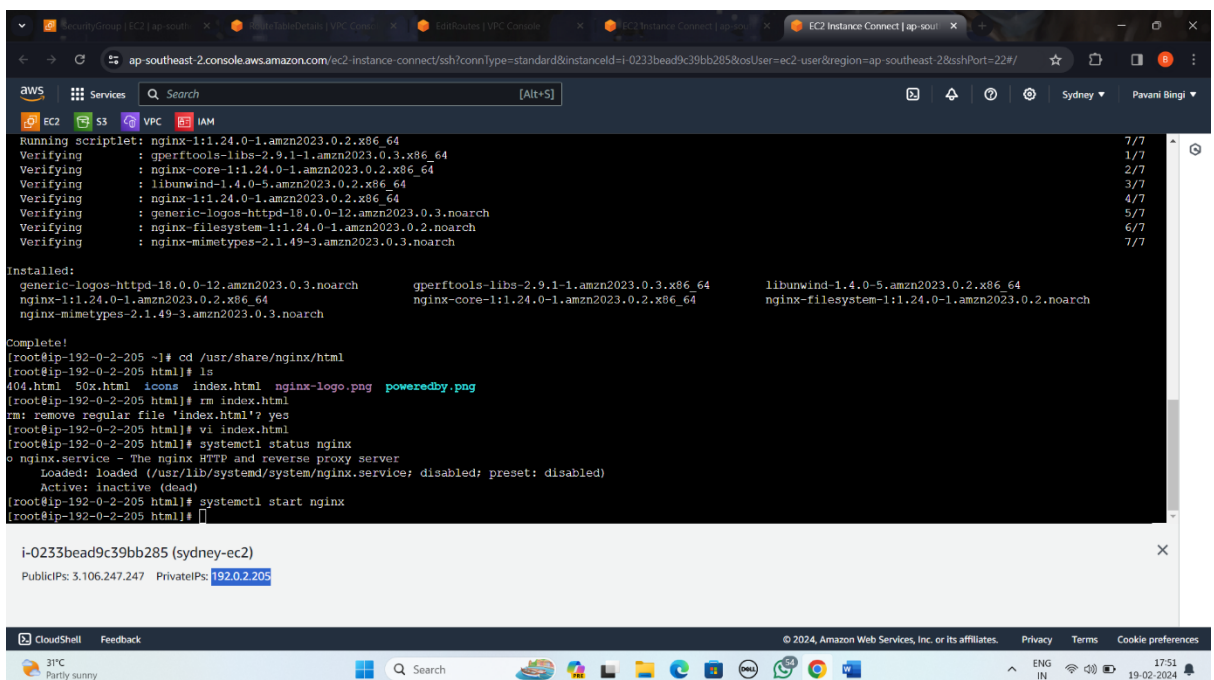
The screenshot shows a terminal window within the AWS CloudShell interface. The user is logged in as 'ec2-user' on an EC2 instance with IP '192.0.2.205'. The terminal shows the following commands and output:

```
[ec2-user@ip-192-0-2-205 ~]$ sudo -i
[root@ip-192-0-2-205 ~]# yum update -y
Last metadata expiration check: 0:01:45 ago on Mon Feb 19 11:13:35 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-192-0-2-205 ~]# yum install nginx -y
Last metadata expiration check: 0:02:09 ago on Mon Feb 19 11:13:35 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing: nginx	x86_64	1:1.24.0-1.amzn2023.0.2	amazonlinux	32 k
Installing dependencies:				
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
gperftools-libs	x86_64	2.9.1-1.amzn2023.0.3	amazonlinux	308 k
libunwind	x86_64	1.4.0-5.amzn2023.0.2	amazonlinux	66 k
nginx-core	x86_64	1:1.24.0-1.amzn2023.0.2	amazonlinux	586 k
nginxfilesystem	noarch	1:1.24.0-1.amzn2023.0.2	amazonlinux	9.1 k
nginx-mimetypes	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	21 k

```
Transaction Summary
Install 7 Packages
```

The terminal also shows the instance details for 'i-0233bead9c39bb285 (sydney-ec2)' with Public IP '3.106.247.247' and Private IP '192.0.2.205'.



The screenshot shows the continuation of the terminal session. The user has navigated to the nginx configuration directory and performed the following actions:

```
[root@ip-192-0-2-205 ~]# cd /usr/share/nginx/html
[root@ip-192-0-2-205 html]# ls
404.html  50x.html  icons  index.html  nginx-logo.png  poweredby.png
[root@ip-192-0-2-205 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-192-0-2-205 html]# vi index.html
[root@ip-192-0-2-205 html]# systemctl status nginx
nginx.service - The nginx HTTP and reverse proxy server
Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
Active: inactive (dead)
[root@ip-192-0-2-205 html]# systemctl start nginx
[root@ip-192-0-2-205 html]#
```

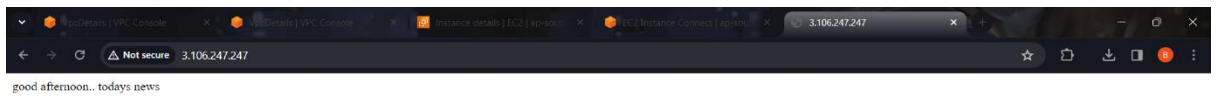
The terminal also shows the installation progress for the packages:

```
Running scriptlet: nginx-1:1.24.0-1.amzn2023.0.2.x86_64 7/7
Verifying      : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64 1/7
Verifying      : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64 2/7
Verifying      : libunwind-1.4.0-5.amzn2023.0.2.x86_64 3/7
Verifying      : nginx-1:1.24.0-1.amzn2023.0.2.x86_64 4/7
Verifying      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch 5/7
Verifying      : nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch 6/7
Verifying      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch 7/7
```

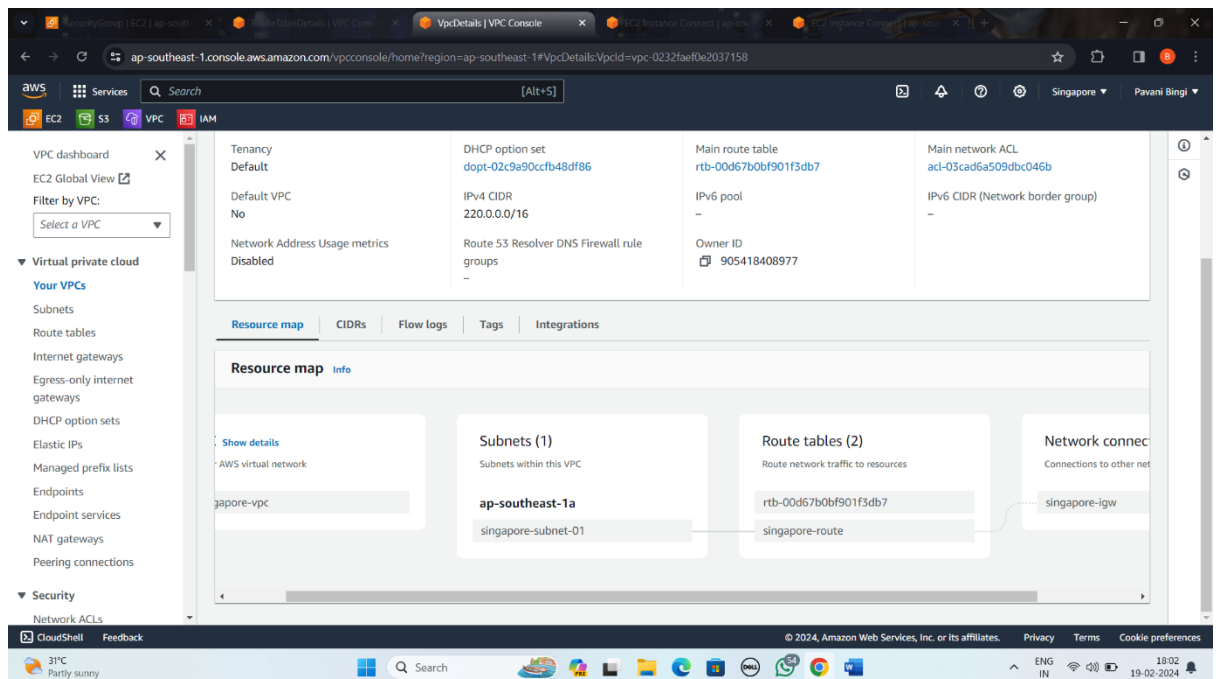
The terminal also shows the installed packages:

```
Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch  gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64  libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.24.0-1.amzn2023.0.2.x86_64               nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64       nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
```

- if the server is active copy the public address and paste it and add default port number of nginx which is 80 using colon : after the public ip



- create a new vpc in another region attach subnet,internet gateway and routetable



- enter into peering connections and new peering connection for two vpc's in diff regions by passing vpc id

pcx-0a4a9f08d5a29c492 / sydney-pc

Details

Requester owner ID	Acceptor owner ID	VPC Peering connection ARN
905418408977	905418408977	arn:aws:ec2:ap-southeast-2:905418408977:vpc-peering-connection/pcx-0a4a9f08d5a29c492
Peering connection ID	Requester VPC	Acceptor VPC
pcx-0a4a9f08d5a29c492	vpc-0941a5f9b18797057 / sydney-vpc-01	vpc-0232faef0e2037158
Status	Requester CIDRs	Acceptor CIDRs
Active	192.0.0.0/16	220.0.0.0/16
Expiration time	Requester Region	Acceptor Region
-	Sydney (ap-southeast-2)	Singapore (ap-southeast-1)

DNS settings

Requester VPC (vpc-0941a5f9b18797057 / sydney-vpc-01) info

Allow acceptor VPC to resolve DNS of hosts in requester VPC to private IP addresses

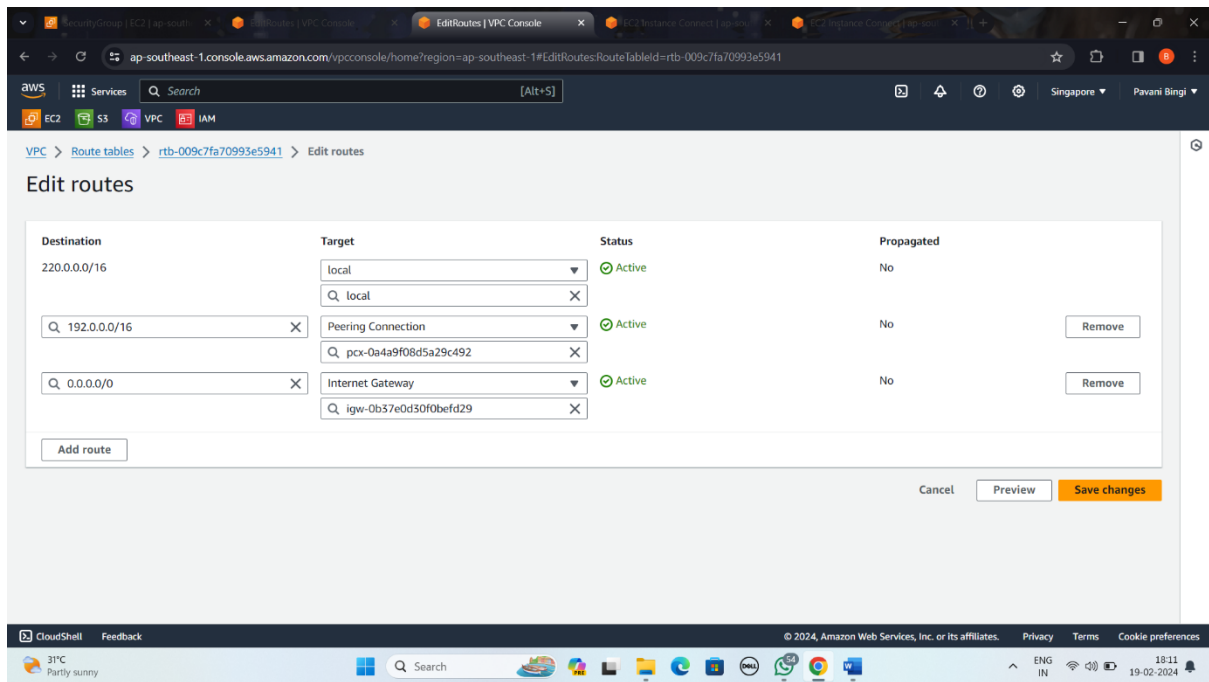
Disabled

- accept the peering connection in another region for connection establishment and then goto routetable give destination address to allow the traffic to flow in the both vpc

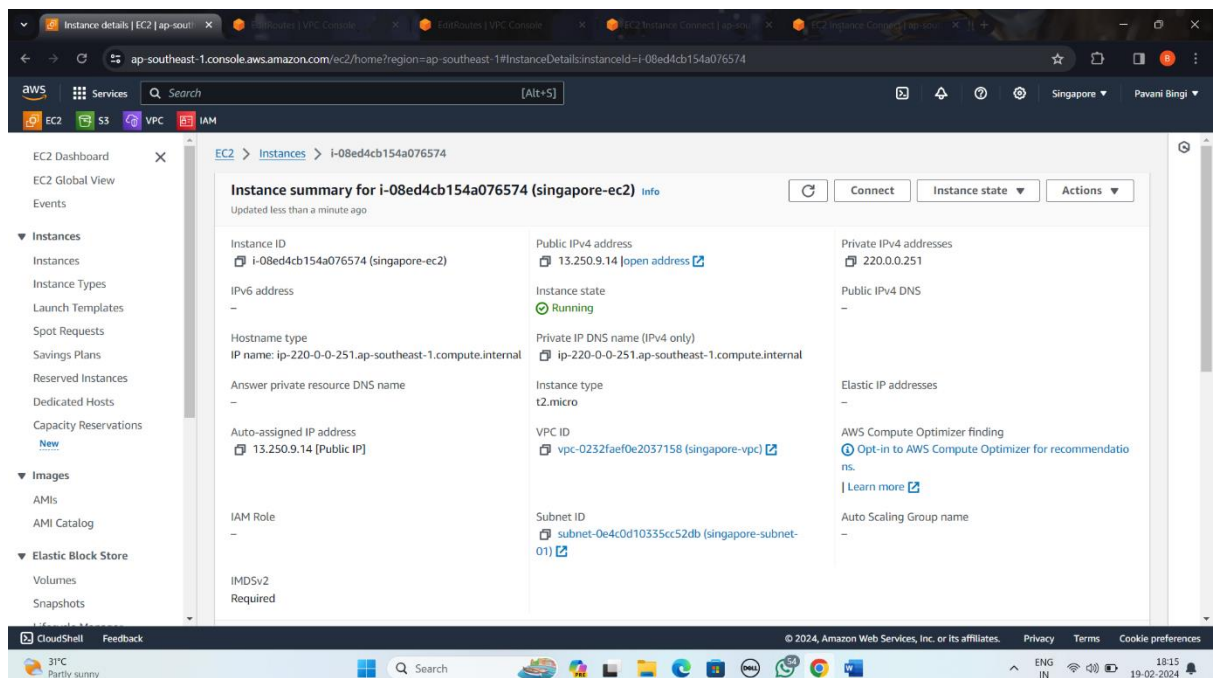
Edit routes

Destination	Target	Status	Propagated
192.0.0.0/16	local	Active	No
220.0.0.0/16	Peering Connection	Active	No
0.0.0.0/0	Internet Gateway	Active	No

Buttons: Add route, Cancel, Preview, Save changes



- after connecting to two vpc's in two diff regions by using peering goto the instance in the new region and create a new instance with following vpc settings



- now connect to CLI and connect to the root user for checking the peering connection establishment in the new instance which is on a different region

