

In this Hands-on we will learn how to communicate from one vpc to another vpc. (Peering connection)

Problem Statement:

You work for XYZ Corporation and based on the expansion requirements of your corporation you have been asked to create and setup distinct Amazon VPC for production and development team.

You are expected to perform the following tasks for the respective VPCs

Production Network:

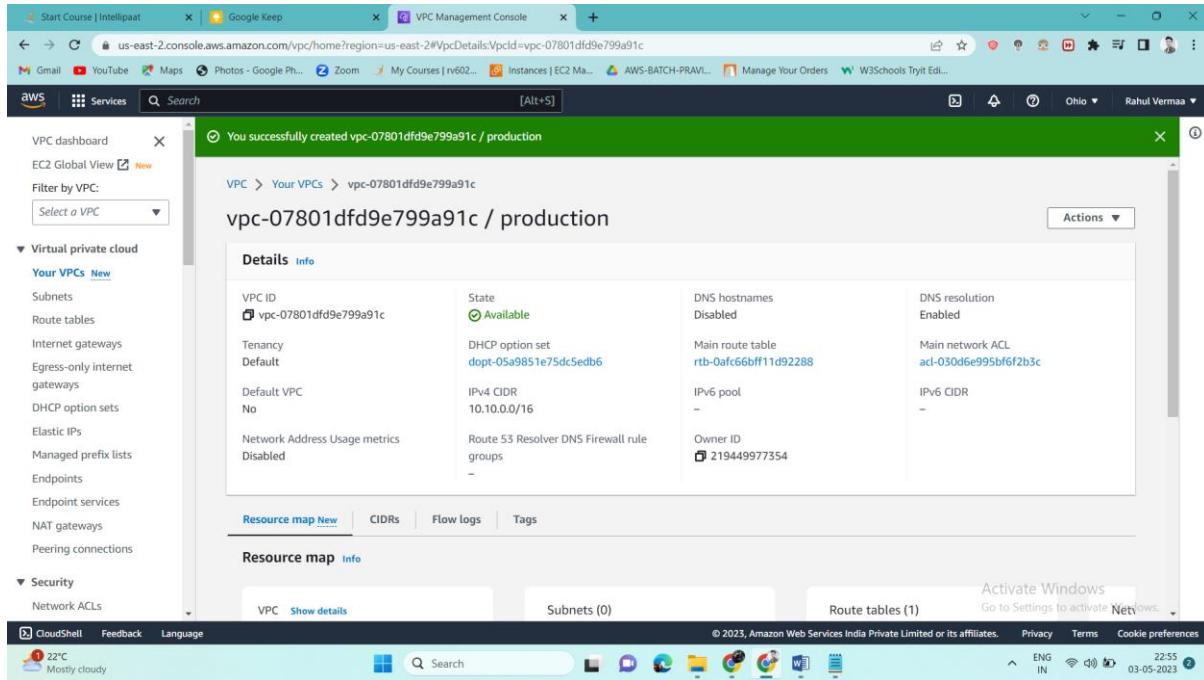
1. Design and build a 4 tier architecture
2. Create 5 subnets out of which 4 should be private with names app1, app2, dbcache and db and one should be public named web.
3. Launch instances in all subnets and name them as per the subnet that they have been launched in.
4. Allow dbcache instance and app1 subnet to send internet requests
5. Manage security groups and NACLs

Development Network:

1. Design and build 2 tier architecture with two subnets named web and db and launch instances in both subnets and name them as per the subnet names.
2. Make sure only web subnet can send internet requests
3. Create peering connection between production network and development network
4. Setup connection between db subnets of both production network and development network respectively.

Step 1: At first we'll create our VPC and will define it as Production, for that we'll search VPC in search box of AWS and just click on Create VPC

We are using 10.10.0.0/16 this CIDR range for our VPC



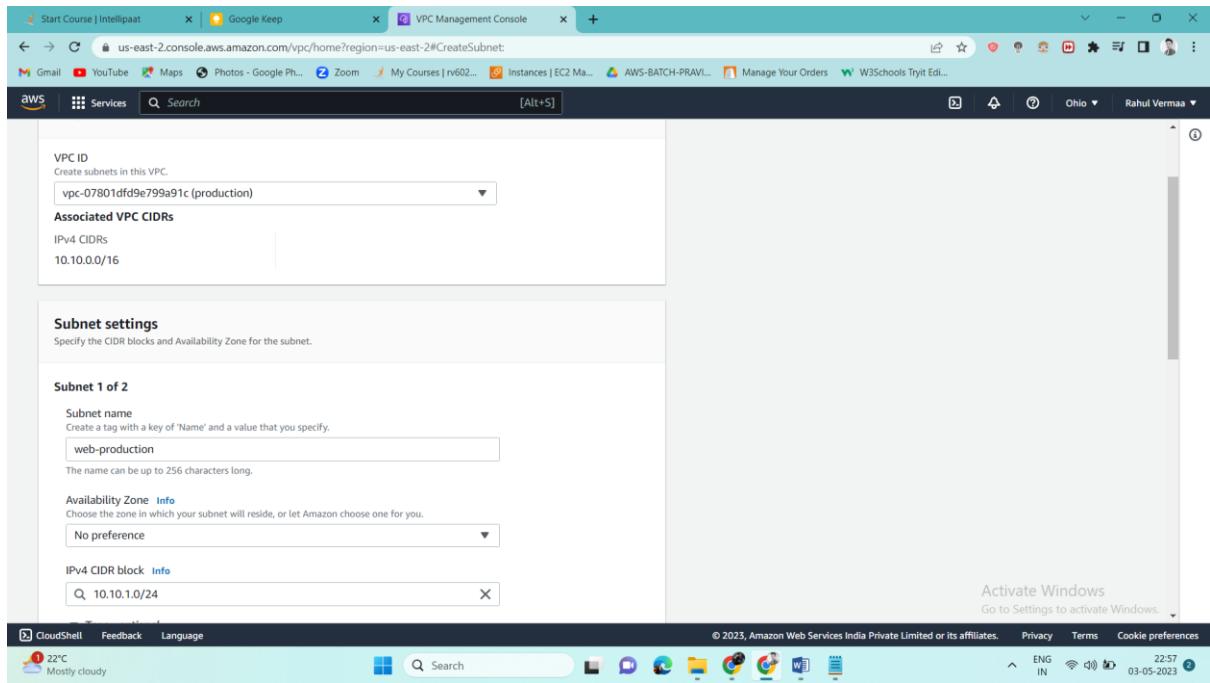
The screenshot shows the AWS VPC Management Console. A success message at the top says "You successfully created vpc-07801dfd9e799a91c / production". The main table displays the following details for the VPC:

Details			
VPC ID vpc-07801dfd9e799a91c	State Available	DNS hostnames Disabled	DNS resolution Enabled
Tenancy Default	DHCP option set dopt-05a9851e75dc5edb6	Main route table rtb-0afc66bfff11d92288	Main network ACL acl-030d6e995bf6f2b3c
Default VPC No	IPv4 CIDR 10.10.0.0/16	IPv6 pool -	IPv6 CIDR -
Network Address Usage metrics Disabled	Route 53 Resolver DNS Firewall rule groups -	Owner ID 219449977354	

Below the table, there are tabs for "Resource map" and "CIDRs". The "Resource map" tab is active. At the bottom, there are buttons for "VPC", "Show details", "Subnets (0)", and "Route tables (1)".

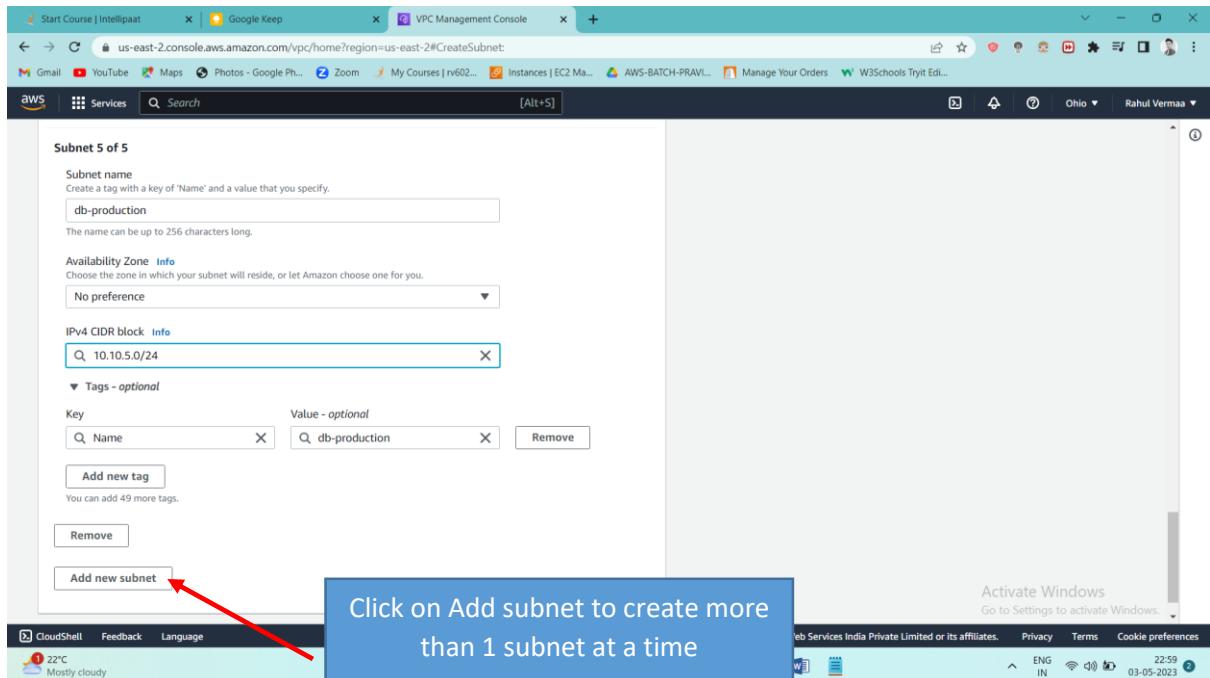
Step 2: Now go to subnets and create Subnets

We'll select Production VPC under VPC ID



We have to create 5 Different subnets we can do it all at once, we just have to click on add subnet button and enter new subnet details and repeat that step.

So we're creating the following subnets **Web-production, app1, dbcashe, app2 and DB-production**.



Subnets are successfully created

You have successfully created 5 subnets: subnet-0b00ad163886aefb9, subnet-0939fcce42a1da866, subnet-02aa3f6b17cdfd18a, subnet-0fa35f3a82ebf18d9, subnet-010fc4d7da3153b6e

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 CIDR
db-production	subnet-010fc4d7da3153b6e	Available	vpc-07801dfd9e799a91c pro...	10.10.5.0/24	-
dbcache	subnet-02aa3f6b17cdfd18a	Available	vpc-07801dfd9e799a91c pro...	10.10.3.0/24	-
app1	subnet-0939fcce42a1da866	Available	vpc-07801dfd9e799a91c pro...	10.10.2.0/24	-
web-production	subnet-0b00ad163886aefb9	Available	vpc-07801dfd9e799a91c pro...	10.10.1.0/24	-
app2	subnet-0fa35f3a82ebf18d9	Available	vpc-07801dfd9e799a91c pro...	10.10.4.0/24	-

Step 3: Now under VPC go to internet gateways and create one named as (Production-IGW)

The following internet gateway was created: igw-0190ec03e566a18fc - production-igw. You can now attach to a VPC to enable the VPC to communicate with the internet.

Internet gateway ID	State	VPC ID	Owner
igw-0190ec03e566a18fc	Detached	-	219449977354

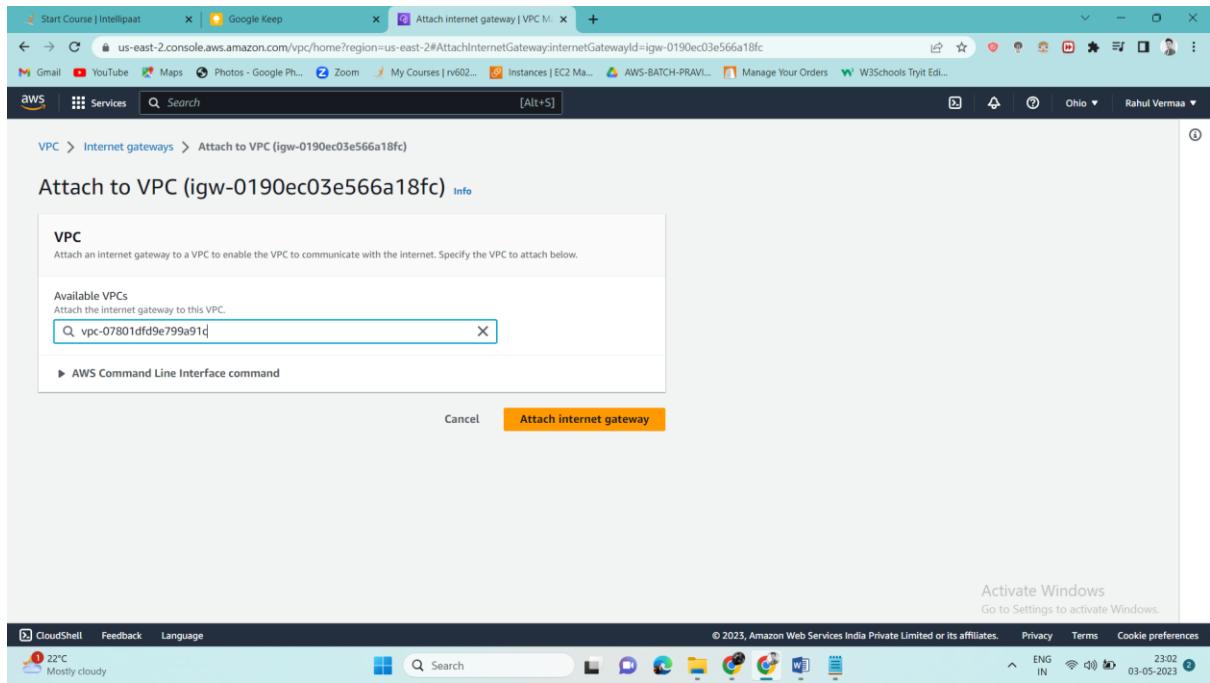
Tags

Key	Value
Name	production-igw

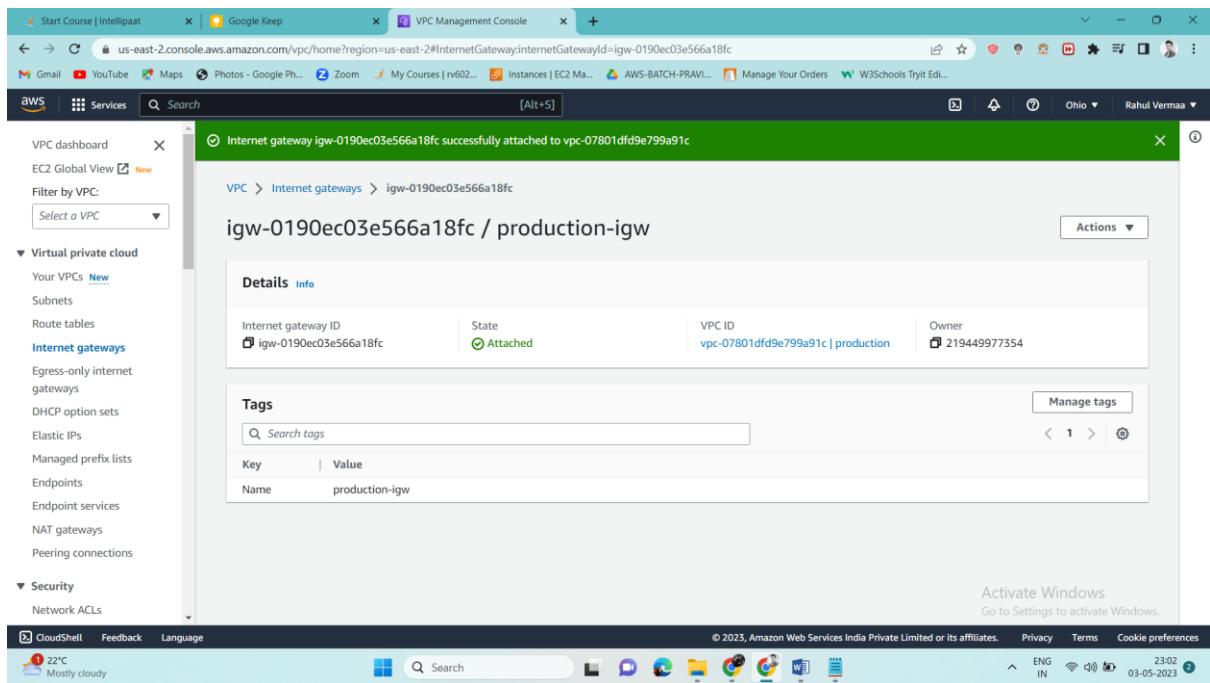
We have attached our internet gateway to our VPC

Select your internet gateway go and click on attach to vpc

Now select your VPC



Internet gateway is Successfully attached to our VPC now



Step 4: Now we have to create route one public route table

(Public route table means it is connected to internet) **productionPublicRoute**

Now we have to route this table (**productionPublicRoute**) to internet gateway

Just go to routes → click on edit routes

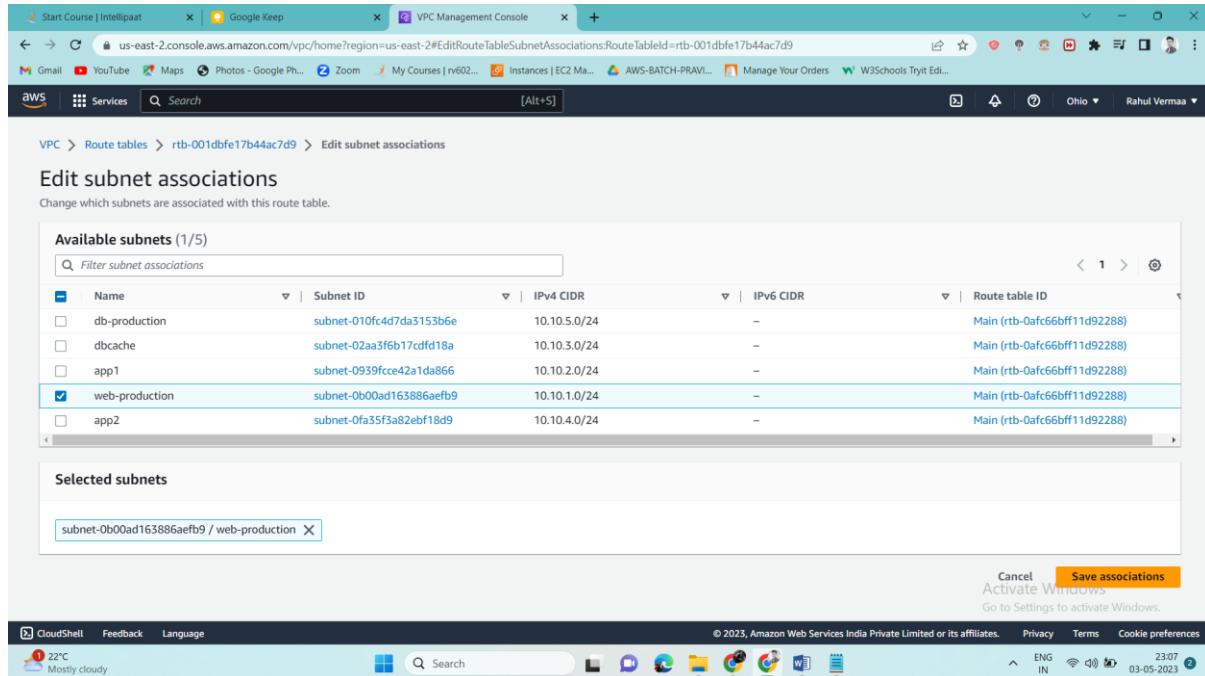
Destination should be 0.0.0.0/0 and Target would be- igw-06bc538194b0dbf57 i.e our **production-IGW (Internet gateways)** and just click on save changes

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0190ec03e566a18fc	Active	No

Step 5: Now to make our **web subnet** as public

We have to associate it in our public route table

For that go to subnet association → click on edit subnet association → select Web-production and click on save associations



VPC > Route tables > rtb-001dbe17b44ac7d9 > Edit subnet associations

Available subnets (1/5)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
db-production	subnet-010fc4d7da315b6e	10.10.5.0/24	-	Main (rtb-0afc66bfff11d92288)
dbcache	subnet-02aa3f6b17cdf18a	10.10.3.0/24	-	Main (rtb-0afc66bfff11d92288)
app1	subnet-0939fcc42a1da866	10.10.2.0/24	-	Main (rtb-0afc66bfff11d92288)
<input checked="" type="checkbox"/> web-production	subnet-0b00ad163886aefb9	10.10.4.0/24	-	Main (rtb-0afc66bfff11d92288)
app2	subnet-0fa35f3a2ebf18d9	10.10.4.0/24	-	Main (rtb-0afc66bfff11d92288)

Selected subnets

subnet-0b00ad163886aefb9 / web-production

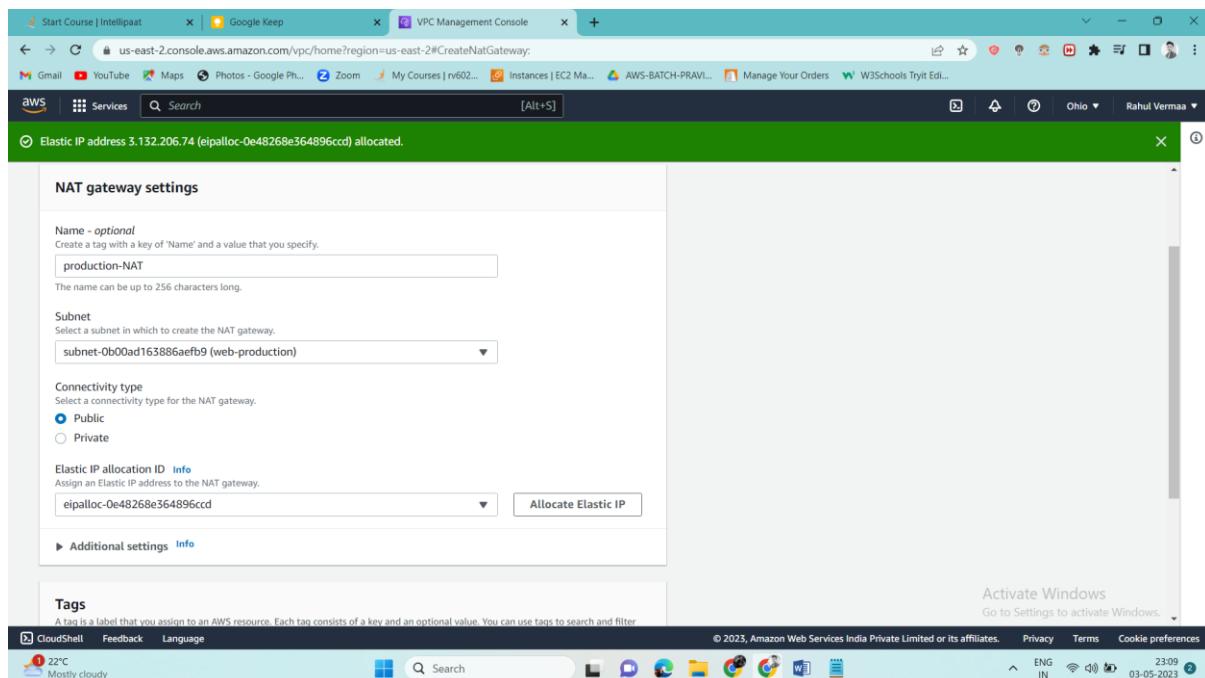
Cancel **Save associations**

Step 6: Now we'll create one NAT gateways to give internet access to our private subnets.

For that go to Nat gateways and create one name it as - (**production NAT**)

In that select subnet web-production (which have internet access)

And allocate one elastic IP and click on create button



Elastic IP address 3.132.206.74 (eipalloc-0e48268e364896cc) allocated.

NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Subnet
Select a subnet in which to create the NAT gateway.

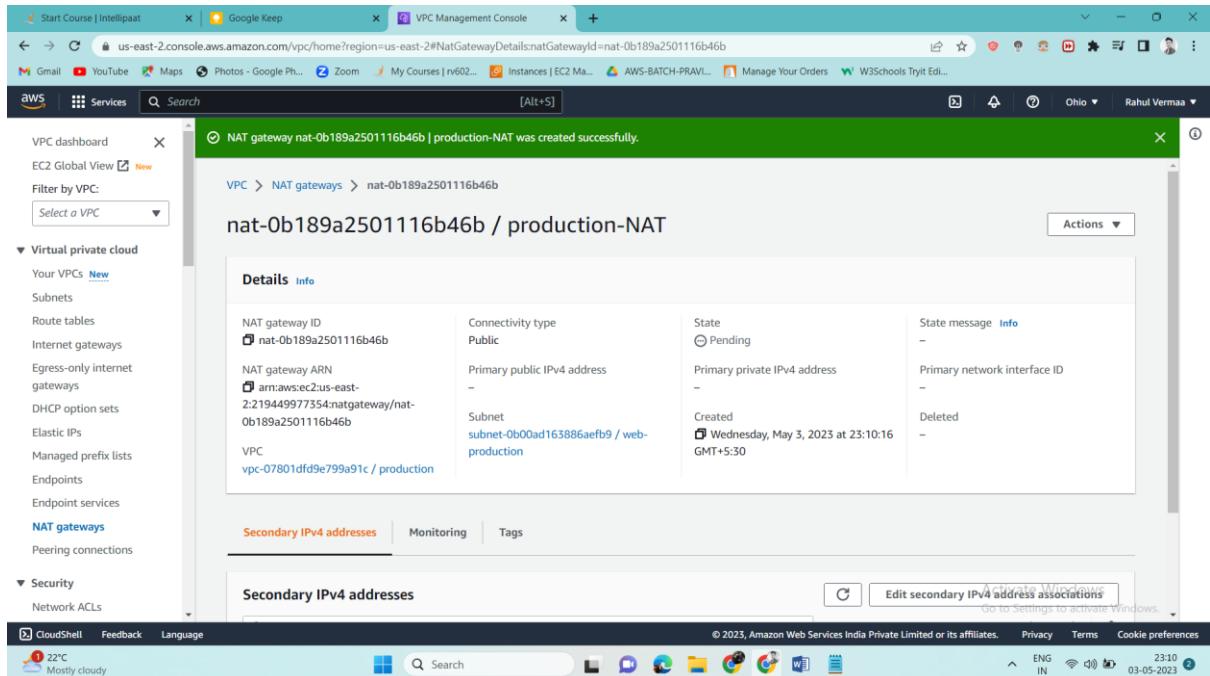
Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

Elastic IP allocation ID [Info](#)
Assign an Elastic IP address to the NAT gateway.

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter.

Activate Windows
Go to Settings to activate Windows.

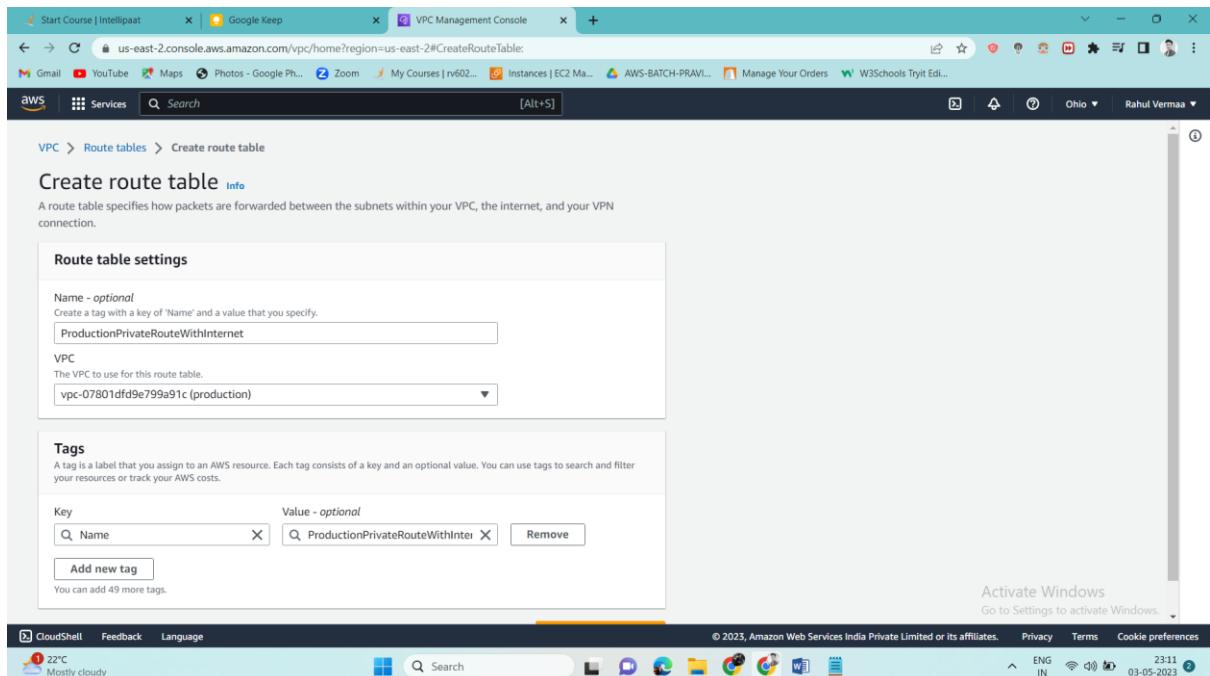
Nat gateway is successfully created



The screenshot shows the AWS VPC Management Console with a success message: "NAT gateway nat-0b189a2501116b46b | production-NAT was created successfully." The details table for the NAT gateway includes:

NAT gateway ID	Connectivity type	State	State message
nat-0b189a2501116b46b	Public	Pending	-
NAT gateway ARN	Primary public IPv4 address	Primary private IPv4 address	Primary network interface ID
arn:aws:ec2:us-east-2:219449977354:natgateway/nat-0b189a2501116b46b	-	-	-
VPC	Subnet	Created	Deleted
vpc-07801fdf9e799a91c / production	subnet-0b00ad163886aefb9 / web-production	Wednesday, May 3, 2023 at 23:10:16 GMT+5:30	-

Step 7: Now we have to create private route table with internet access that means whatever subnets are associated under these route tables can access internet. Name it as **(ProductionPrivateRouteWithInternet)**.



The screenshot shows the "Create route table" page. The "Route table settings" section includes:

- Name - optional:** ProductionPrivateRouteWithInternet
- VPC:** vpc-07801fdf9e799a91c (production)

The "Tags" section shows a single tag: "Name" with value "ProductionPrivateRouteWithInternet".

Now go to routes → click on edit routes → Destination should be 0.0.0.0/0 and Target would be- igw-06bc538194b0dbf57 i.e our production-IGW (internet gateway)

Now go to subnet association → edit subnet association and select **app1 and dbcache**

And click on save

Available subnets (2/5)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
db-production	subnet-010fc4d7da3153b6e	10.10.5.0/24	-	Main (rtb-0afc66bfff11d92288)
<input checked="" type="checkbox"/> dbcache	subnet-02aa3f6b17cdfd18a	10.10.3.0/24	-	Main (rtb-0afc66bfff11d92288)
<input checked="" type="checkbox"/> app1	subnet-0939fcce42a1da866	10.10.2.0/24	-	Main (rtb-0afc66bfff11d92288)
web-production	subnet-0b00ad163886aefb9	10.10.1.0/24	-	rtb-001dbfe17b44ac7d9 / productionP...
app2	subnet-0fa35f3a82ebf18d9	10.10.4.0/24	-	Main (rtb-0afc66bfff11d92288)

Selected subnets

subnet-0939fcce42a1da866 / app1 subnet-02aa3f6b17cdfd18a / dbcache

Cancel **Save associations**

Step 8: Now we have to create one more route table for private subnet without internet

Just got to route table and create one. Name it as
(ProductionPrivateRouteWithoutInternet)

Create route table [Info](#)

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

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
ProductionPrivateRouteWithoutInternet

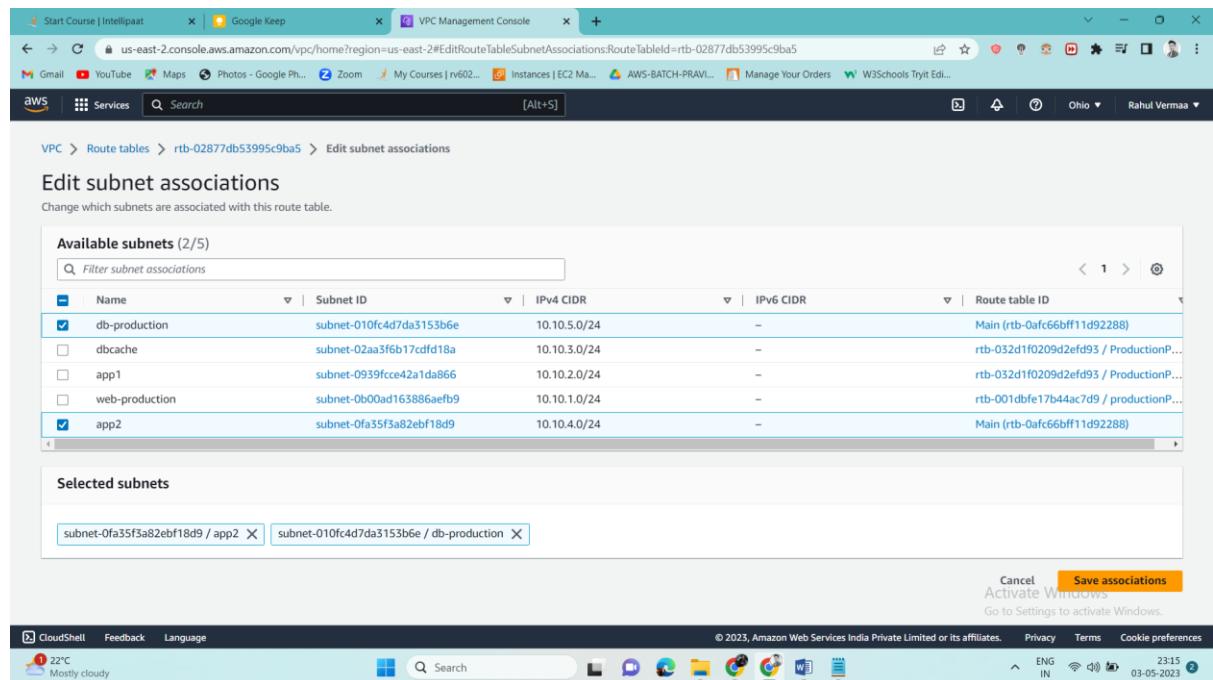
VPC
The VPC to use for this route table.
vpc-07801df9e799a91c (production)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - *optional*
You can add 49 more tags.

Activate Windows
Go to Settings to activate Windows.

Now just associate these subnets to this route table **app2** and **Db-production**

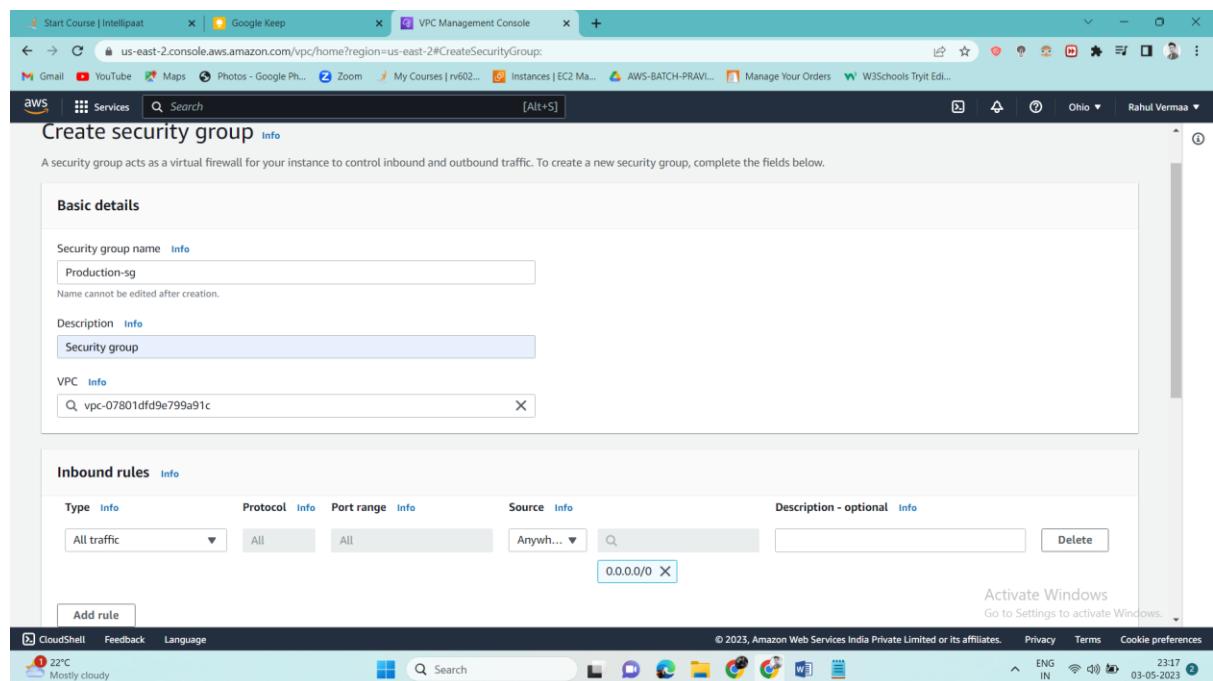


The screenshot shows the AWS VPC Management Console with the URL <https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#EditRouteTableSubnetAssociations:RouteTableId=rtb-02877db53995c9ba5>. The page is titled 'Edit subnet associations' and shows a table of available subnets. The 'Selected subnets' section contains 'subnet-0fa35f3a82ebf18d9 / app2' and 'subnet-010fc4d7da3153b6e / db-production'. The 'Save associations' button is highlighted in orange.

Step 9: Now go to security groups and create one

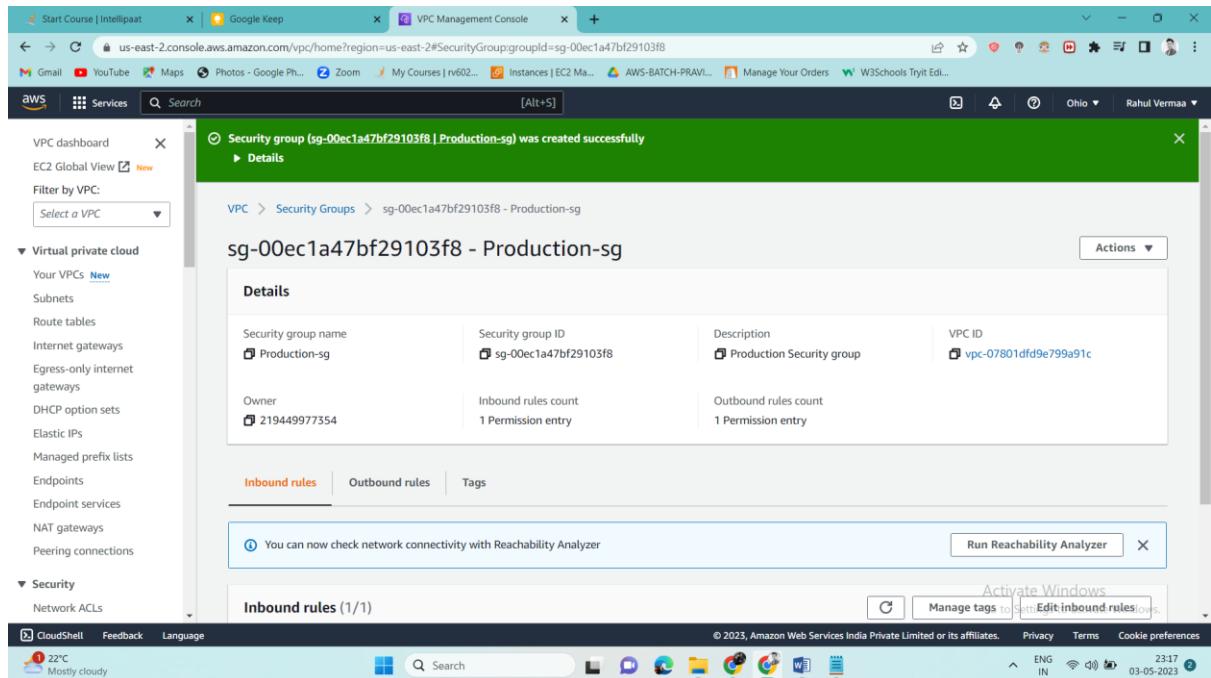
Select production VPC instead of default one

And in edit inbound rules type- all traffic and destination – anywhere



The screenshot shows the AWS VPC Management Console with the URL <https://us-east-2.console.aws.amazon.com/vpc/home?region=us-east-2#CreateSecurityGroupGroup:VpcId=vpc-07801dfd9e799a91c>. The page is titled 'Create security group' and shows the 'Basic details' and 'Inbound rules' sections. The 'Add rule' button is highlighted in orange.

Security group is successfully created



The screenshot shows the AWS VPC Management Console with a success message: "Security group (sg-00ec1a47bf29103f8 | Production-sg) was created successfully". The main details table for the security group "sg-00ec1a47bf29103f8 - Production-sg" is displayed, showing the following information:

Security group name	Security group ID	Description	VPC ID
Production-sg	sg-00ec1a47bf29103f8	Production Security group	vpc-07801dfd9e799a91c
Owner	219449977354	Inbound rules count 1 Permission entry	Outbound rules count 1 Permission entry

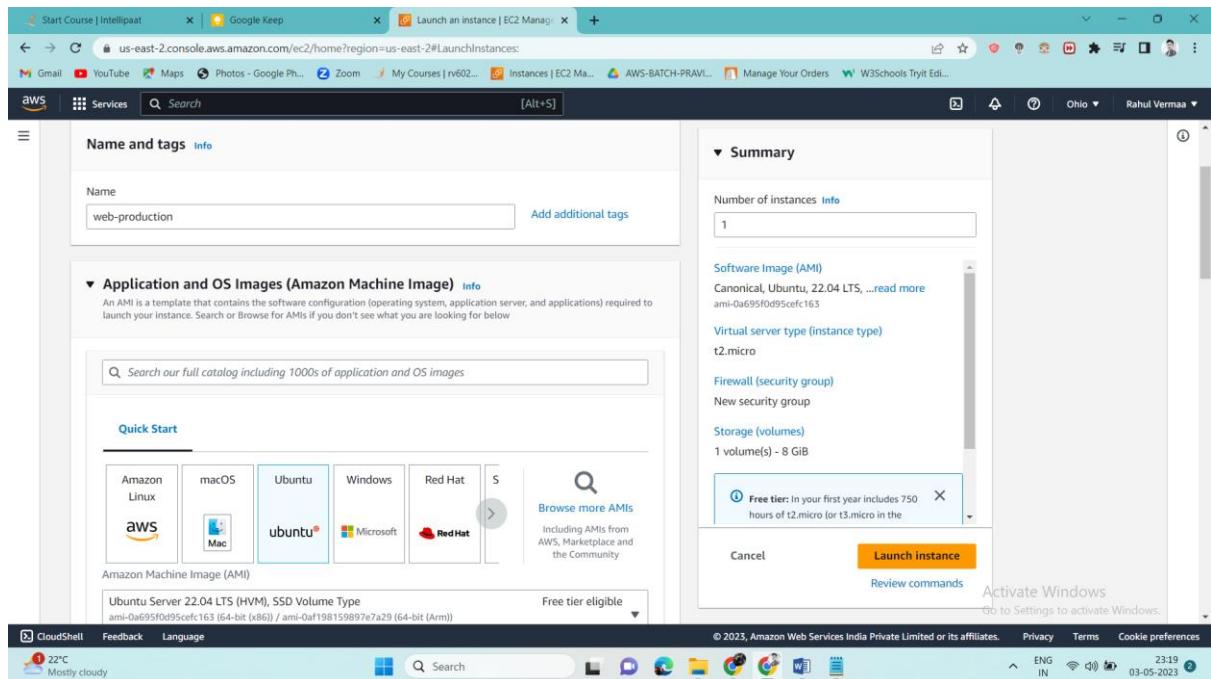
The "Inbound rules" tab is selected, showing one rule: "You can now check network connectivity with Reachability Analyzer". A "Run Reachability Analyzer" button is available. The "Actions" dropdown menu is open, showing options like "Edit inbound rules", "Delete", and "Copy".

Step 10: Now we have to create instances one by one for our subnets

First instance web-production → Os- Ubuntu

Under network settings select your VPC (production) and subnet- web-production

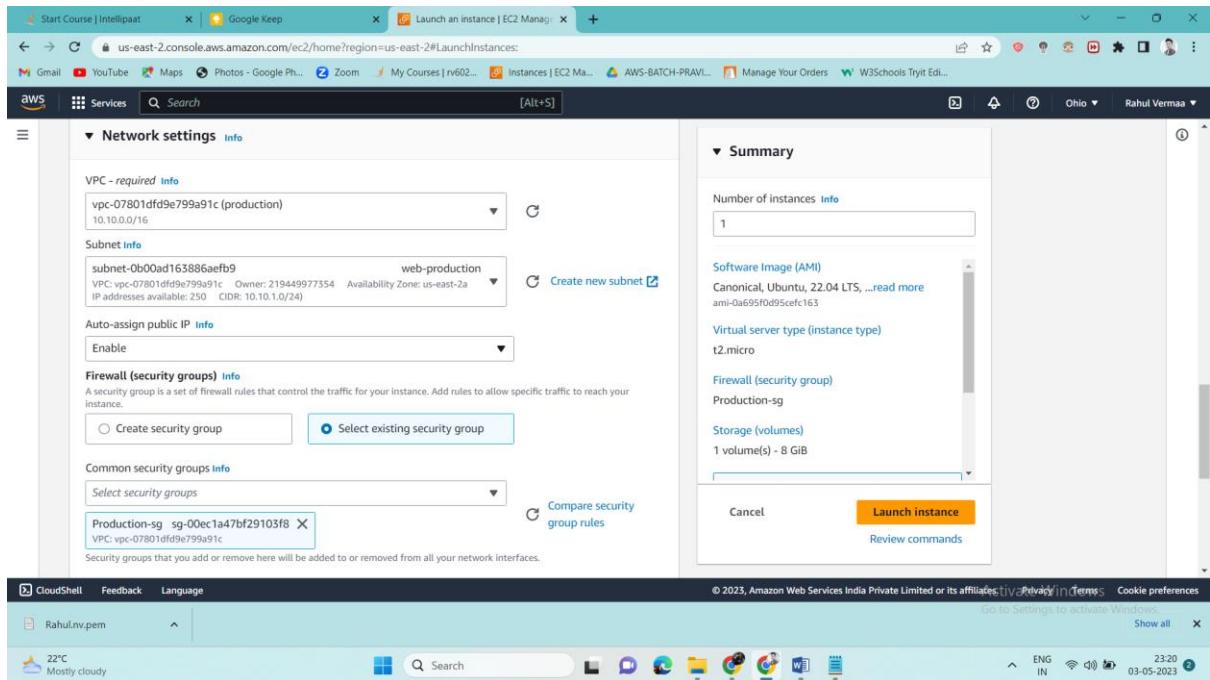
Enable auto assign public ip and select existing security group i.e production-SG



The screenshot shows the "Launch an instance" wizard in the AWS Management Console. The "Name and tags" step is selected, with the name "web-production" entered. The "Summary" step is shown on the right, with the following configuration:

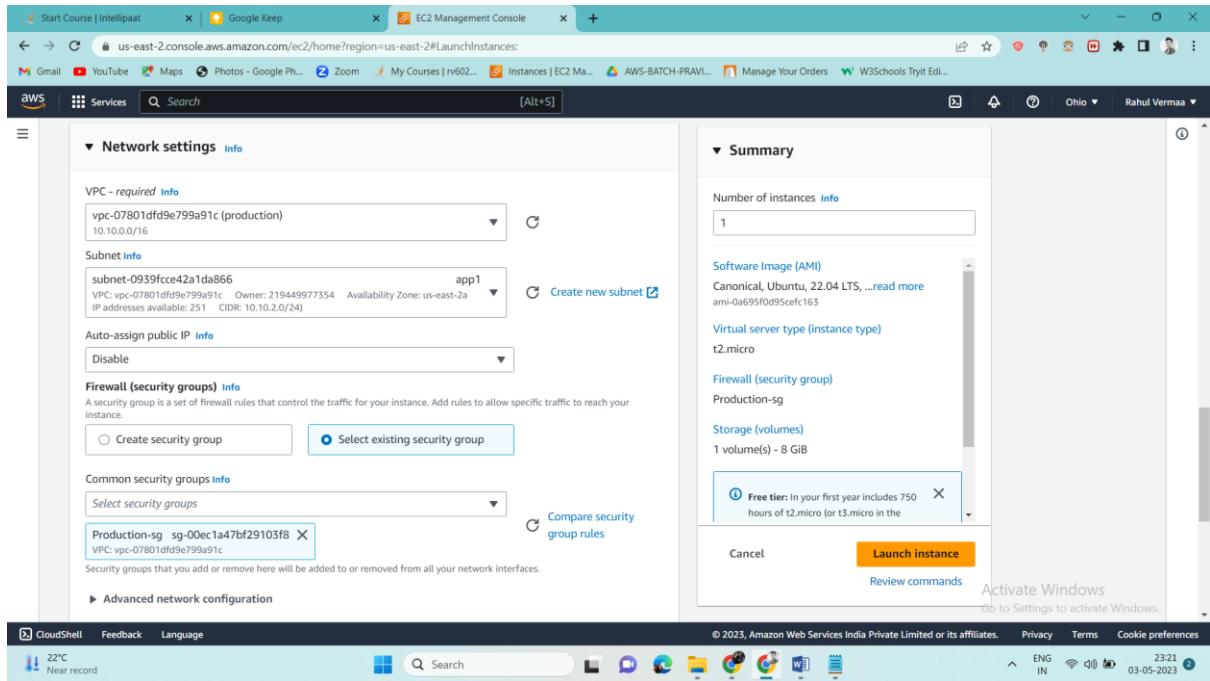
- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 22.04 LTS (ami-0a695f0d95cefc163)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A "Free tier: In your first year includes 750 hours of t2.micro (or t3.micro) in the Free tier" message is displayed. A "Launch instance" button is visible. The "Review commands" and "Activate Windows" sections are also present at the bottom.



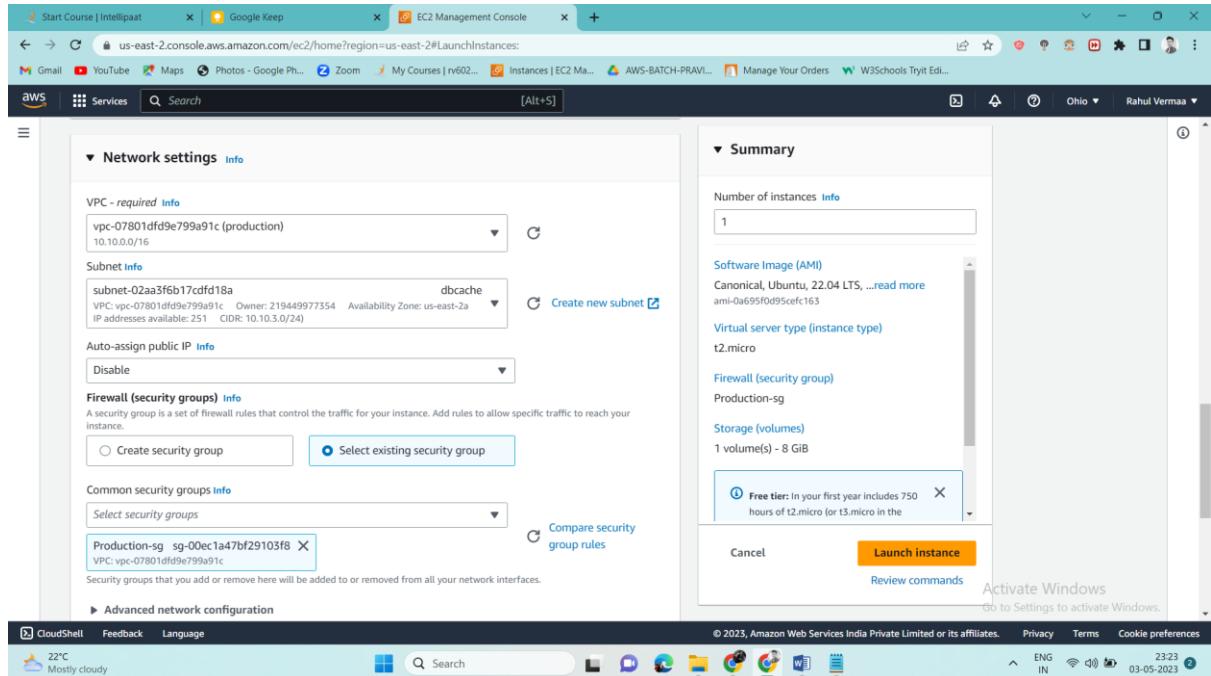
Step 11: Now create second instance app1 everything will be same.

We just have to change some steps like in this we have to select subnet- app1 and auto assign public ip should be disable. Rest all the steps are similar OR same



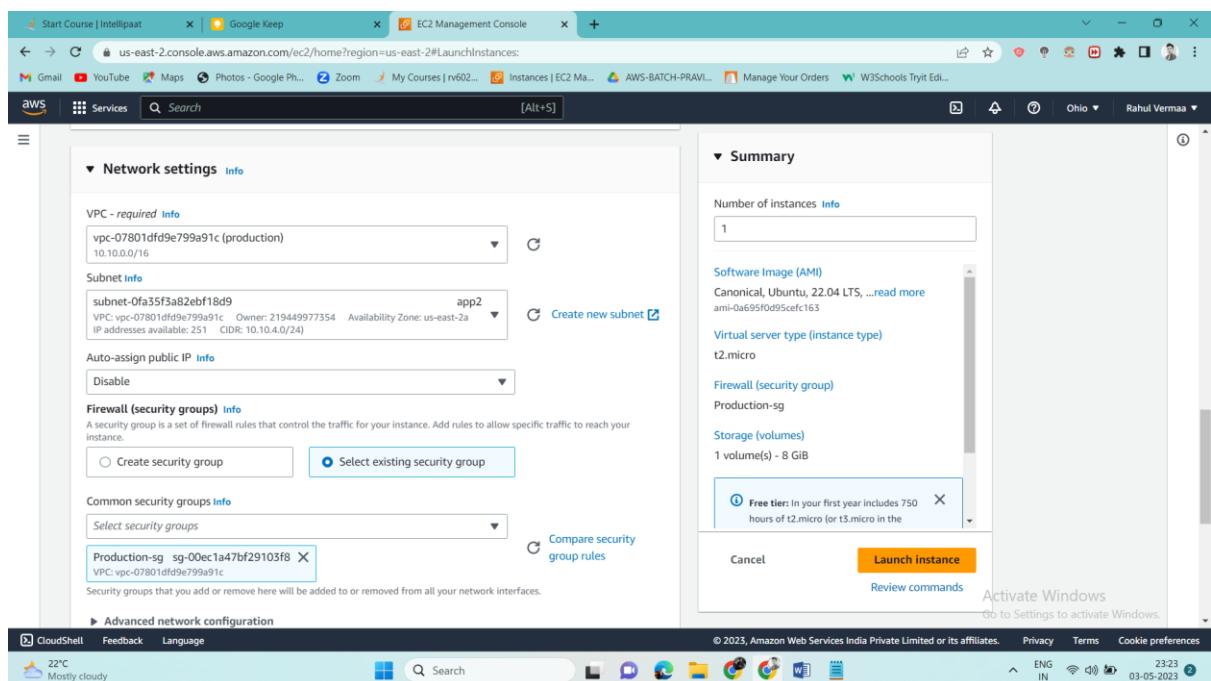
Step 12: Now create third instance dbcashe everything will be same.

We just have to change some steps like in this we have to select subnet- dbcashe and auto assign public ip should be disable. Rest all the steps are similar OR same



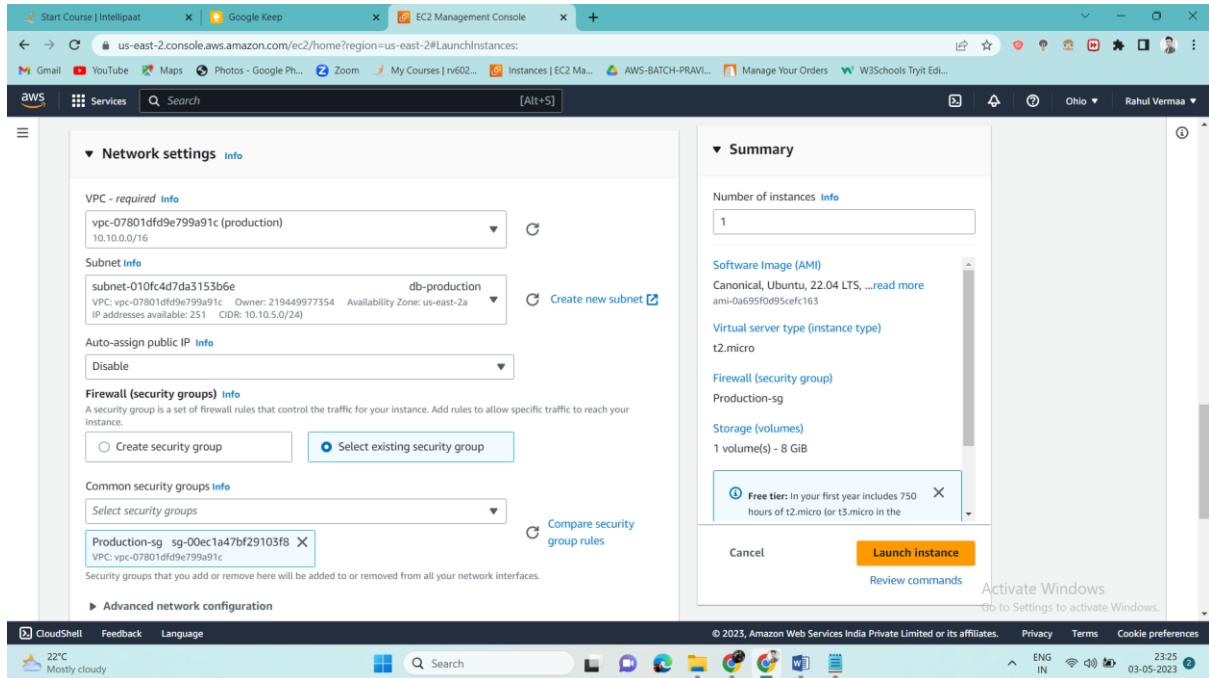
Step 13: Now create fourth instance app2 everything will be same.

We just have to change some steps like in this we have to select subnet- app2 and auto assign public ip should be disable. Rest all the steps are similar OR same

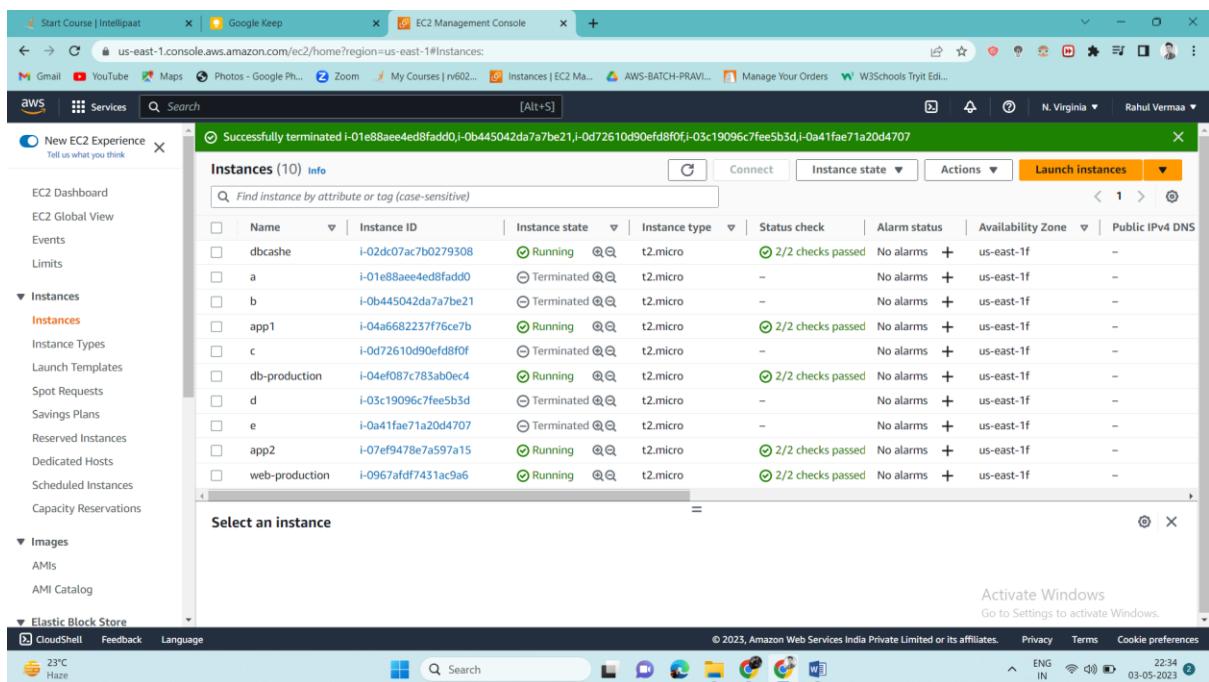


Step 14: Now create our last instance db-production everything will be same.

We just have to change some steps like in this we have to select subnet- db-production and auto assign public ip should be disable. Rest all the steps are similar OR same



All instances are created and ready



Step 15: Now will test our instances are working properly or not

-We'll connect to **Web-production instance** to check if it has internet access or not

-After connecting, Write some commands-

sudo su

apt-get update

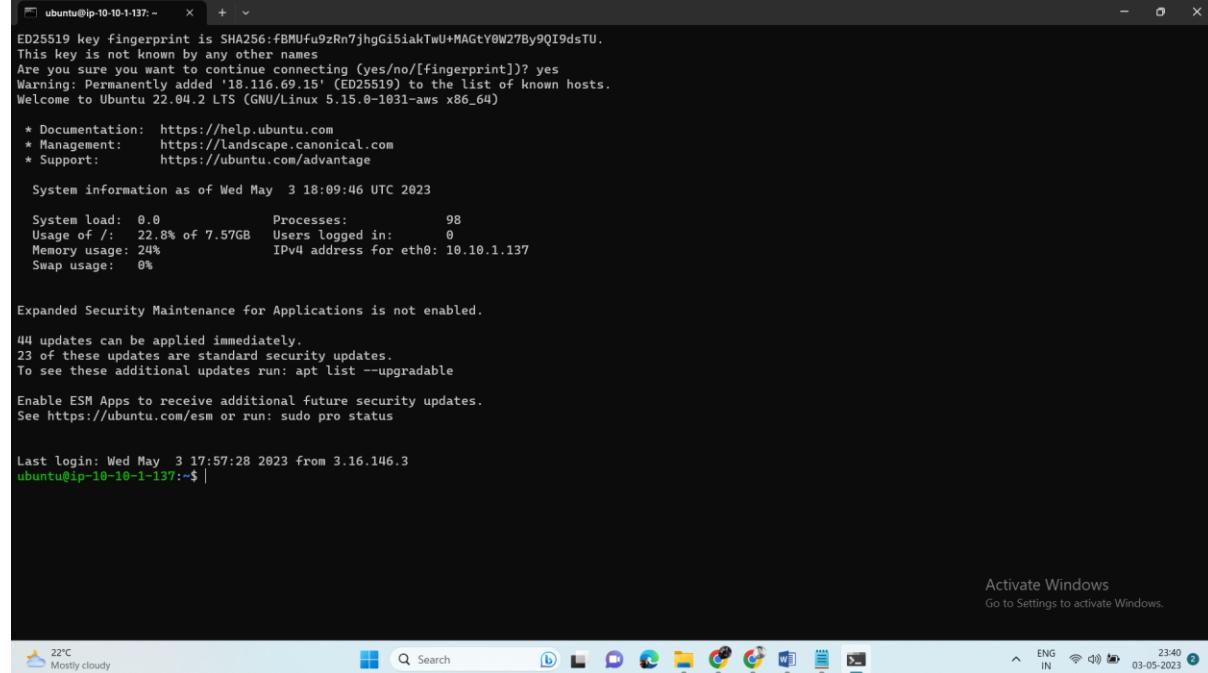
nano Rahul-nvv.pem

-Open your key pair in notepad (mine is Rahul-nvv.pem) Copy Rahul-nvv.pem key content and paste it in the nano editor. Now press **ctrl+s** **ctrl+x** to save and exit

chmod 400 Rahul-nvv.pem (read permission)

Now copy ssh client code of **db-production** instance

And paste it in our instance and type yes



```
ubuntu@ip-10-10-1-137:~ + 
ED25519 key fingerprint is SHA256:fBMUfu9zRn7jhgGi5iakTwU+MAGtY0W27By9QI9dsTU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '18.116.69.15' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed May  3 18:09:46 UTC 2023

System load: 0.0          Processes:      98
Usage of /: 22.8% of 7.57GB  Users logged in:    0
Memory usage: 24%          IPv4 address for eth0: 10.10.1.137
Swap usage:  0%

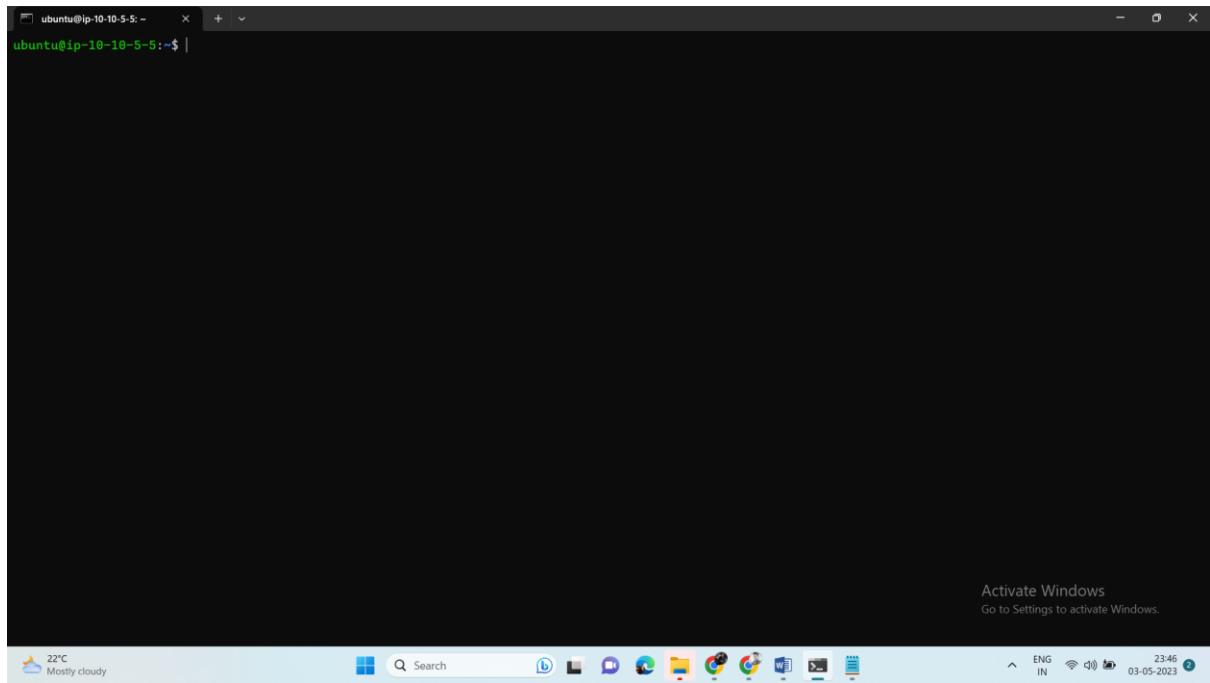
Expanded Security Maintenance for Applications is not enabled.

44 updates can be applied immediately.
23 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

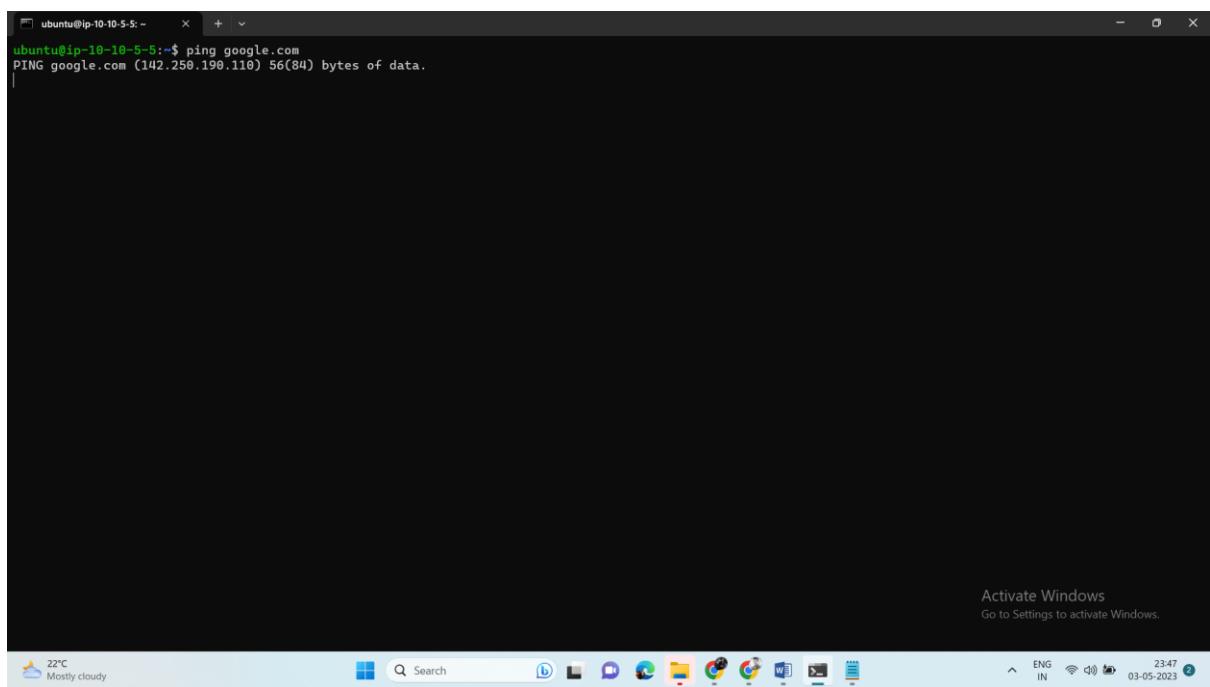
Last login: Wed May  3 17:57:28 2023 from 3.16.146.3
ubuntu@ip-10-10-1-137:~$ |
```

Now we are connected to **dbproduction10.10.5.5**



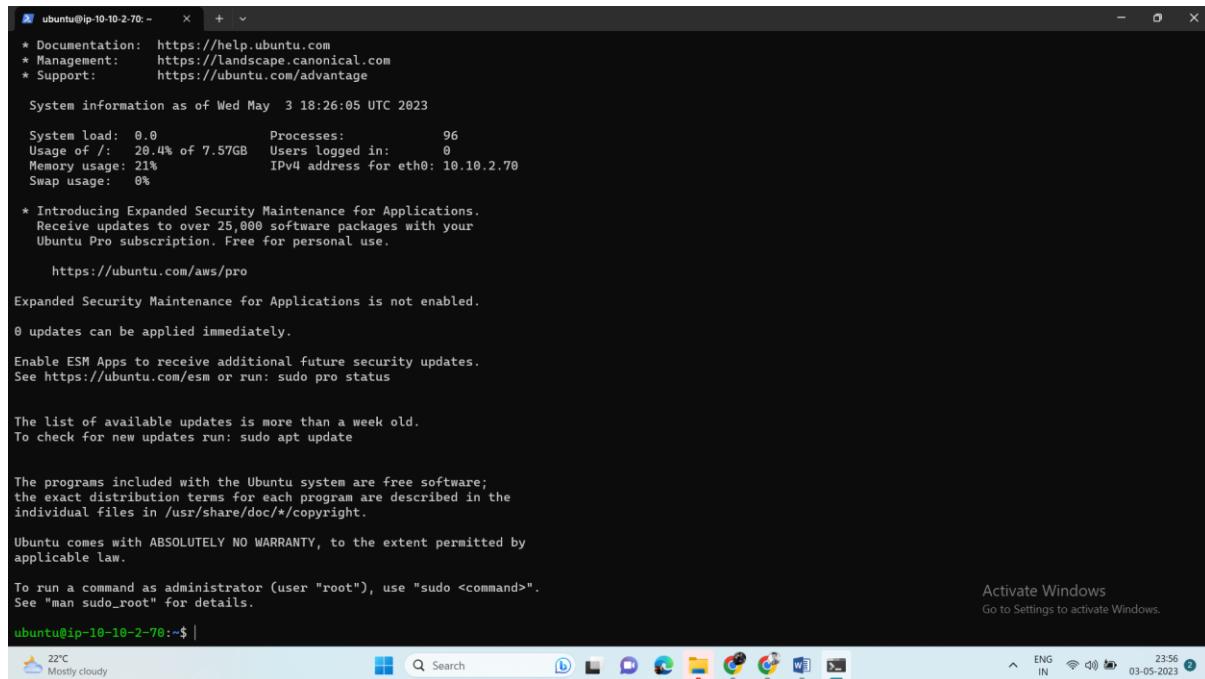
Now will try to ping google, write- **ping www.google.com**

As you can see we are not able to ping because it don't have internet access



Step 16: Now will exit from db-production instance and connect it **app1** as it is private instance but it is allowed for internet access.

We are connected to app1



```
ubuntu@ip-10-10-2-70:~ + ~
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Wed May 3 18:26:05 UTC 2023

System load: 0.0 Processes: 96
Usage of /: 20.4% of 7.57GB Users logged in: 0
Memory usage: 21% IPv4 address for eth0: 10.10.2.70
Swap usage: 0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

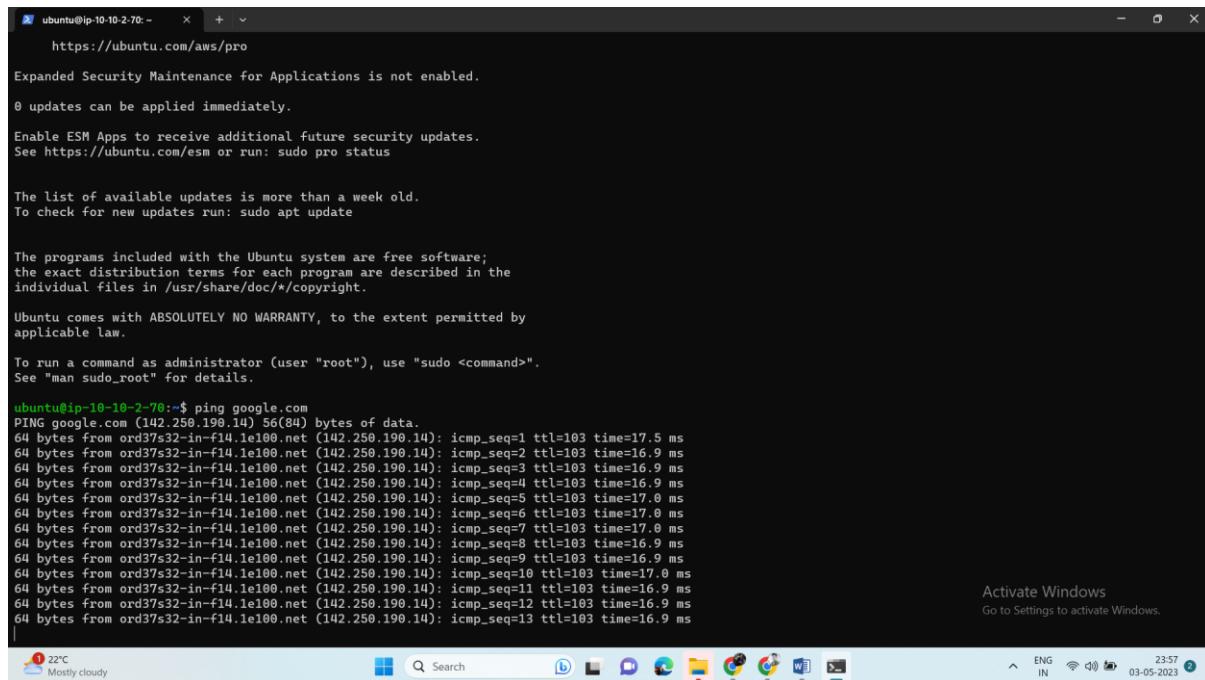
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-2-70:~$ |
```

Now will check if it is connected to internet or not for that will write- **ping www.google.com**

And here we can see it is able to ping google.



```
ubuntu@ip-10-10-2-70:~ + ~
https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-10-10-2-70:~$ ping google.com
PING google.com (142.250.190.14) 56(84) bytes of data.
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=1 ttl=103 time=17.5 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=2 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=3 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=4 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=5 ttl=103 time=17.0 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=6 ttl=103 time=17.0 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=7 ttl=103 time=17.0 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=8 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=9 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=10 ttl=103 time=17.0 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=11 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=12 ttl=103 time=16.9 ms
64 bytes from ord37s32-in-f14.le100.net (142.250.190.14): icmp_seq=13 ttl=103 time=16.9 ms

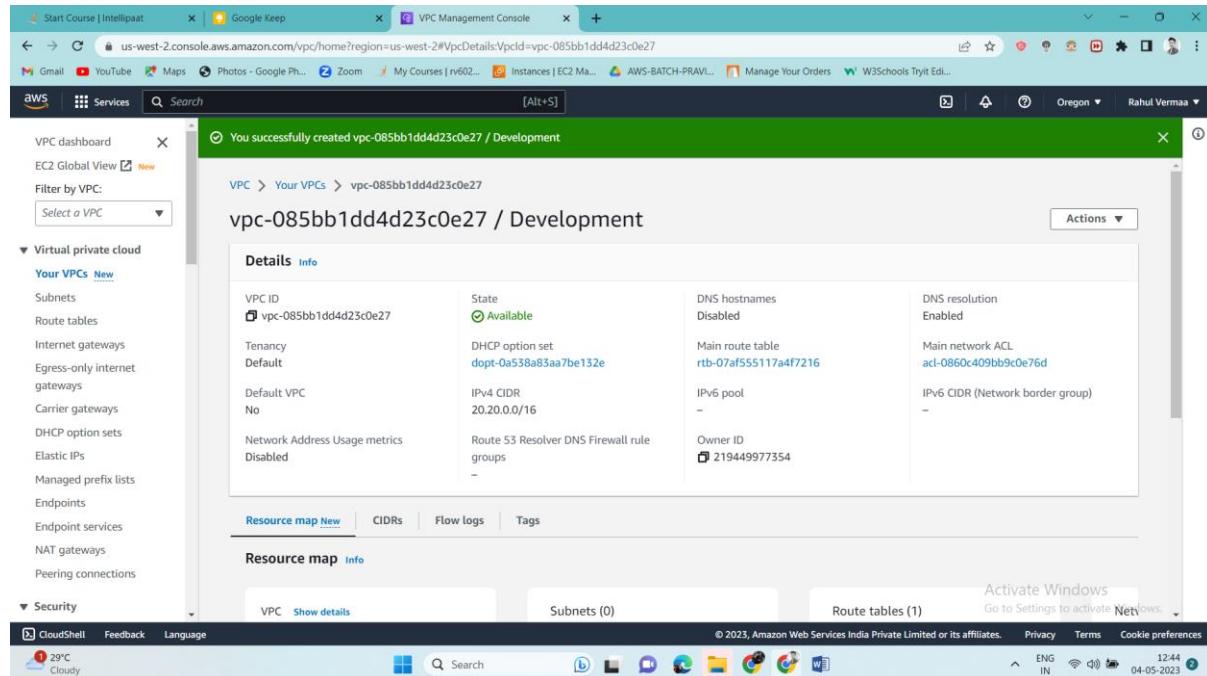
Activate Windows
Go to Settings to activate Windows.

ubuntu@ip-10-10-2-70:~$ |
```

Now will work on Development Network

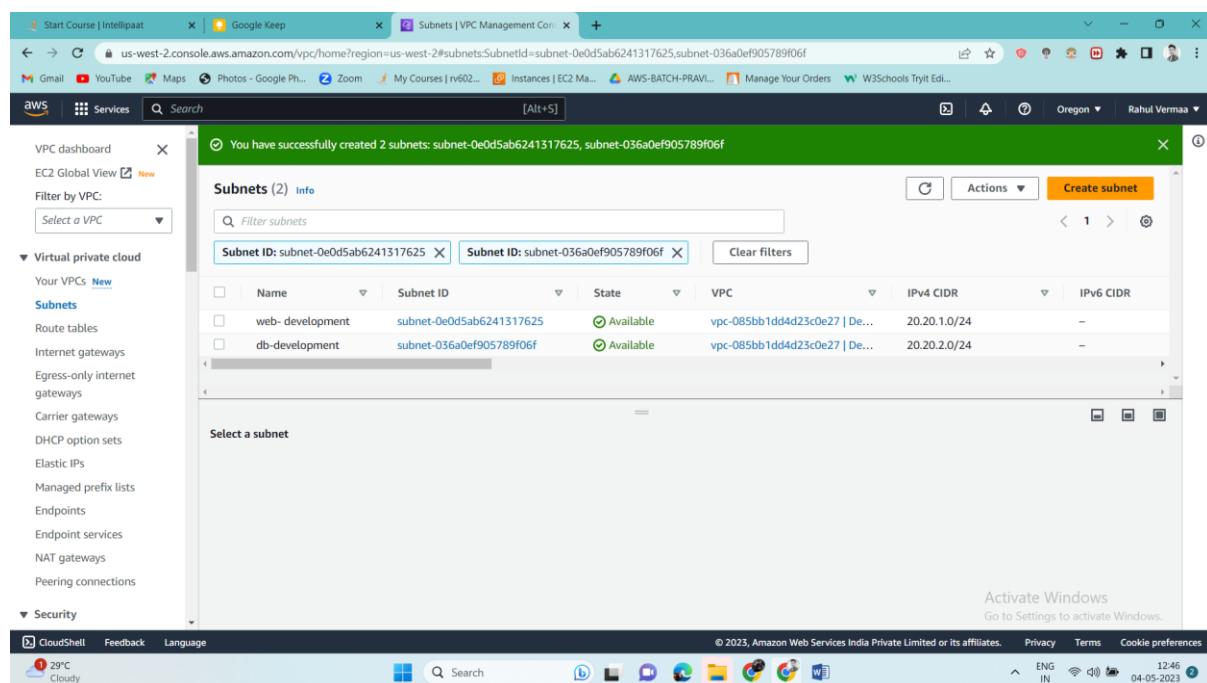
Step 17: Will do that in Oregon region

First will create VPC named as- **Development**



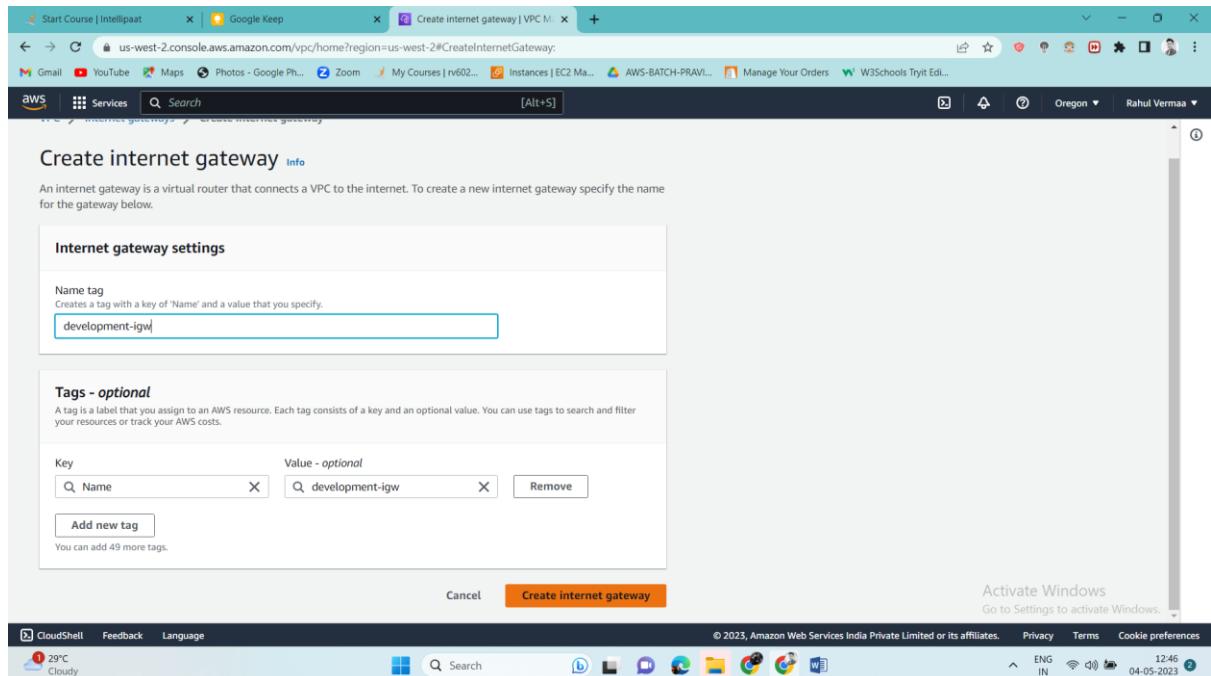
The screenshot shows the AWS VPC Management Console. A green success message at the top says "You successfully created vpc-085bb1dd4d23c0e27 / Development". The main table displays the details of the newly created VPC, including its ID, state (Available), and various network settings. The VPC is associated with a main route table (rtb-07af55517a4f7216) and a main network ACL (acl-0860c409bb90e76d). The table also shows the IPv4 CIDR (20.20.0.0/16) and the owner ID (219449977354). Below the table, there are tabs for "Resource map" and "CIDRs". The bottom of the screen shows the AWS navigation bar and the status bar indicating the region as "Oregon" and the date and time as "04-05-2023".

Step 18: Now we'll create 2 subnets- **webdevelopment** and **db-development**



The screenshot shows the AWS Subnets page. A green success message at the top says "You have successfully created 2 subnets: subnet-0e0d5ab6241317625, subnet-036a0ef905789f06f". The main table lists the two subnets: "web- development" and "db- development". Both subnets are in an "Available" state, belong to the "Development" VPC, and have IPv4 CIDRs of 20.20.1.0/24 and 20.20.2.0/24 respectively. The table includes columns for Name, Subnet ID, State, VPC, IPv4 CIDR, and IPv6 CIDR. Below the table, there is a "Select a subnet" dropdown. The bottom of the screen shows the AWS navigation bar and the status bar indicating the region as "Oregon" and the date and time as "04-05-2023".

Step 19: Now we need one internet gateway



Internet gateway settings

Name tag

Creates a tag with a key of 'Name' and a value that you specify.

development-igw

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

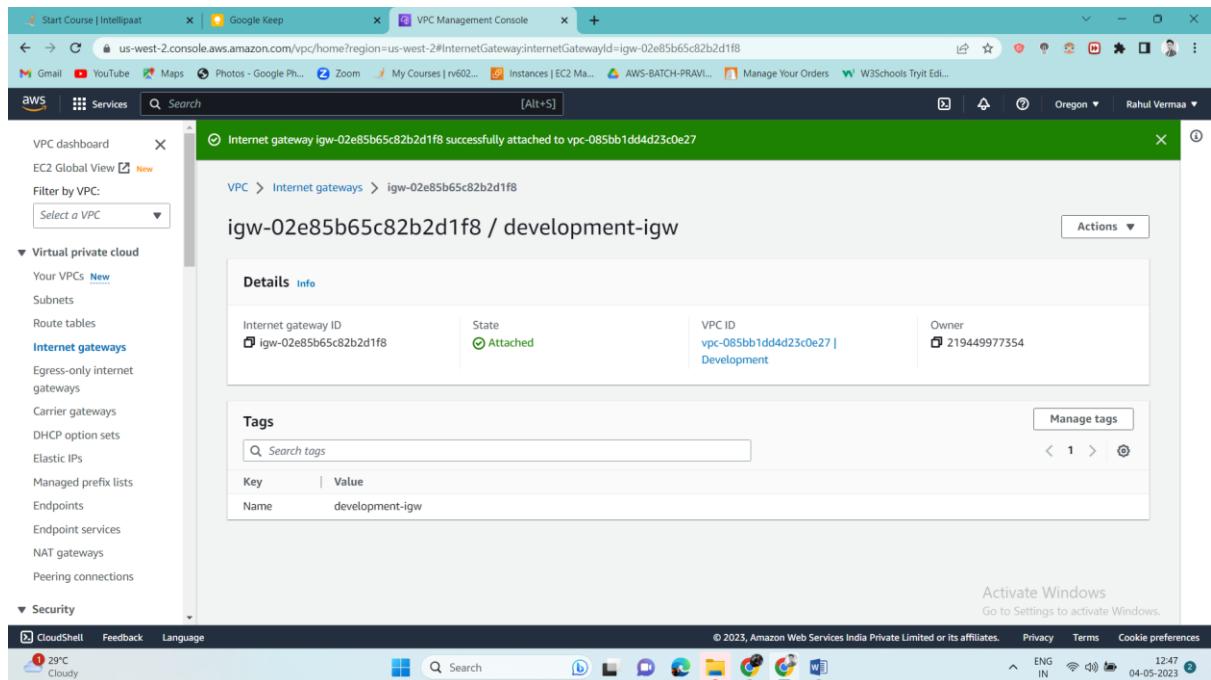
Key: Name, Value - optional: development-igw

Add new tag

You can add 49 more tags.

Create internet gateway

Now we have to attach it to our VPC (Development)



Internet gateway igw-02e85b65c82b2d1f8 successfully attached to vpc-085bb1dd4d23c0e27

VPC > Internet gateways > igw-02e85b65c82b2d1f8 / development-igw

Actions

Details

Internet gateway ID: igw-02e85b65c82b2d1f8, State: Attached, VPC ID: vpc-085bb1dd4d23c0e27, Owner: Development

Tags

Name: development-igw

Activate Windows

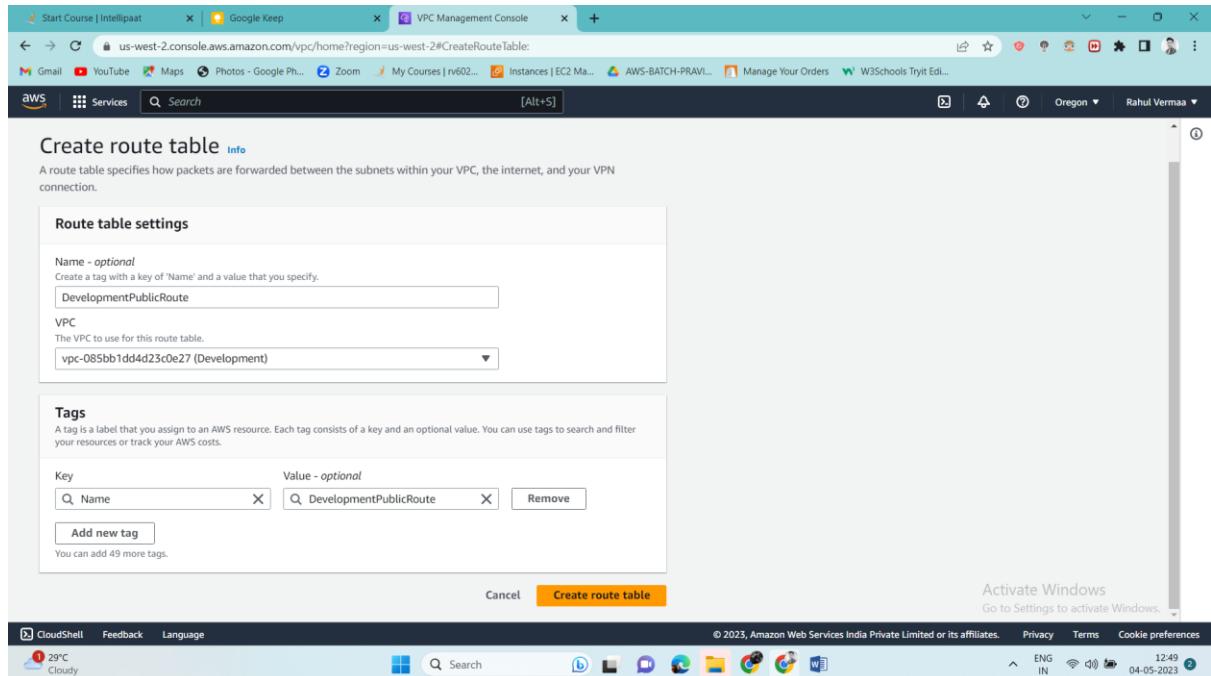
CloudShell Feedback Language

CloudShell Feedback Language

Step 20: Now we need two route tables

DevelopmentPublicRoute and DevelopmentPrivateRoute

First will create public route- **DevelopmentPublicRoute**



Create route table [Info](#)

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

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

DevelopmentPublicRoute

VPC
The VPC to use for this route table.

vpc-085bb1dd4d23c0e27 (Development)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - *optional* Remove

Add new tag

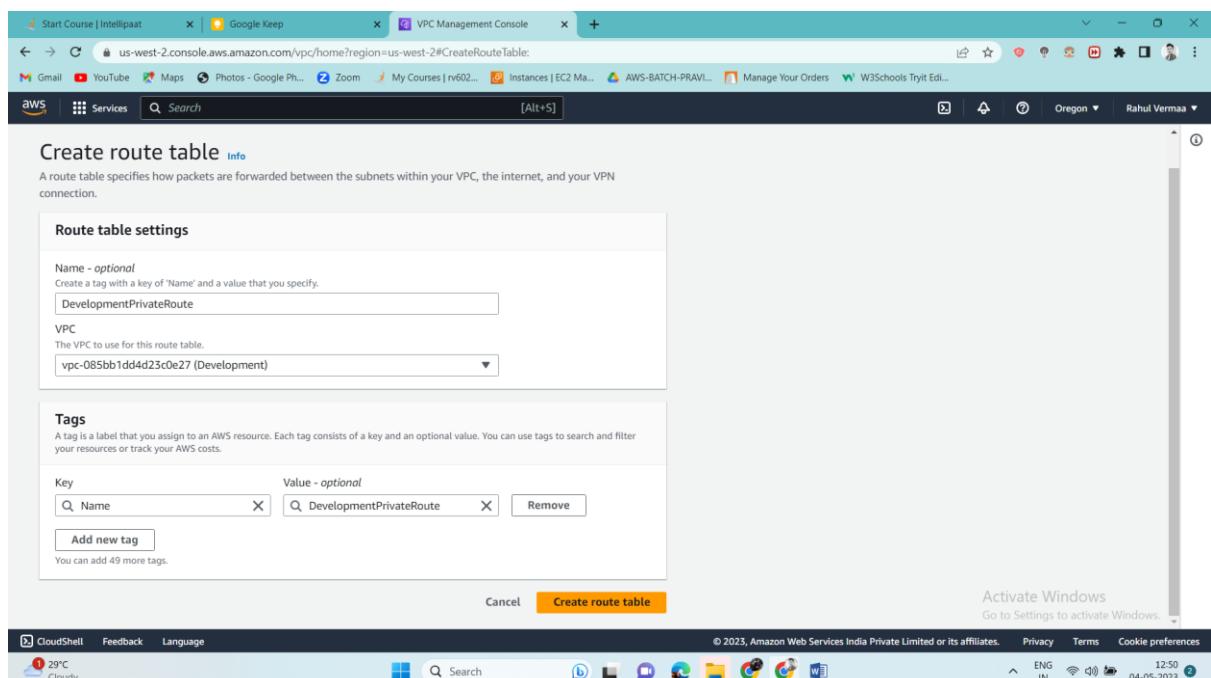
You can add 49 more tags.

Cancel **Create route table**

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Cloudy 12:49 04-05-2023 ENG IN

Now will create our second route table i.e public route table- **DevelopmentPrivateRoute**



Create route table [Info](#)

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

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

DevelopmentPrivateRoute

VPC
The VPC to use for this route table.

vpc-085bb1dd4d23c0e27 (Development)

Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key Value - *optional* Remove

Add new tag

You can add 49 more tags.

Cancel **Create route table**

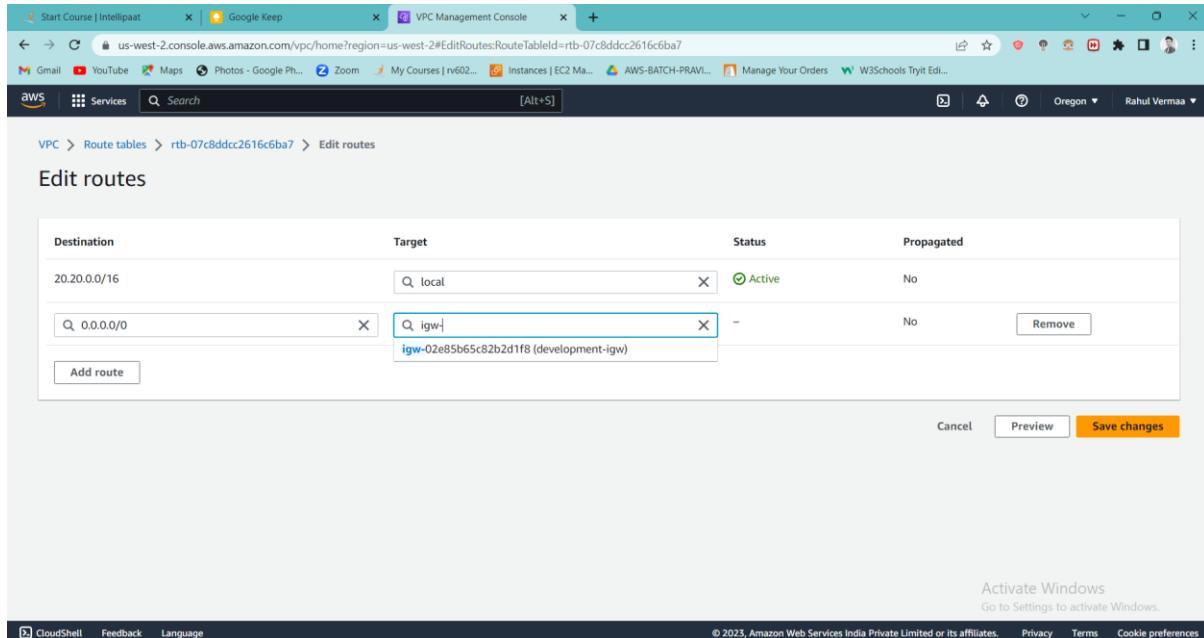
Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 29°C Cloudy 12:50 04-05-2023 ENG IN

Step 21: Now we're going to route internet gateway to our public route table

For that just select your public route (**DevelopmentPublicRoute**) → go to routes → click on edit route

Type- 0.0.0.0/0 and destination would be internet gateway and select your internet gateway.



Destination	Target	Status	Propagated
20.20.0.0/16	local	Active	No
0.0.0.0/0	igw-[igw-02e85b65c82b2d1fb (development-igw)]	-	No

Now we have to associate subnet in that route table

For that just go to subnet association → click on edit subnet association

Select web-development and click on save.

VPC > Route tables > rtb-07c8ddcc2616c6ba7 > Edit subnet associations

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> web- development	subnet-0e0d5ab6241317625	20.20.1.0/24	-	Main (rtb-07af555117a4f7216)
<input type="checkbox"/> db- development	subnet-036a0ef905789f06f	20.20.2.0/24	-	Main (rtb-07af555117a4f7216)

Selected subnets

subnet-0e0d5ab6241317625 / web- development

Cancel **Save associations**

Activate Windows
Go to Settings to activate Windows.

Step 22: Now we associate subnet in our private route table (**DevelopmentPrivateRoute**). Same steps as step 21, just select db-development instead of web-development.

VPC > Route tables > rtb-0fea09ba7cf47d94 > Edit subnet associations

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input type="checkbox"/> web- development	subnet-0e0d5ab6241317625	20.20.1.0/24	-	rtb-07c8ddcc2616c6ba7 / DevelopmentPrivateRoute
<input checked="" type="checkbox"/> db- development	subnet-036a0ef905789f06f	20.20.2.0/24	-	Main (rtb-07af555117a4f7216)

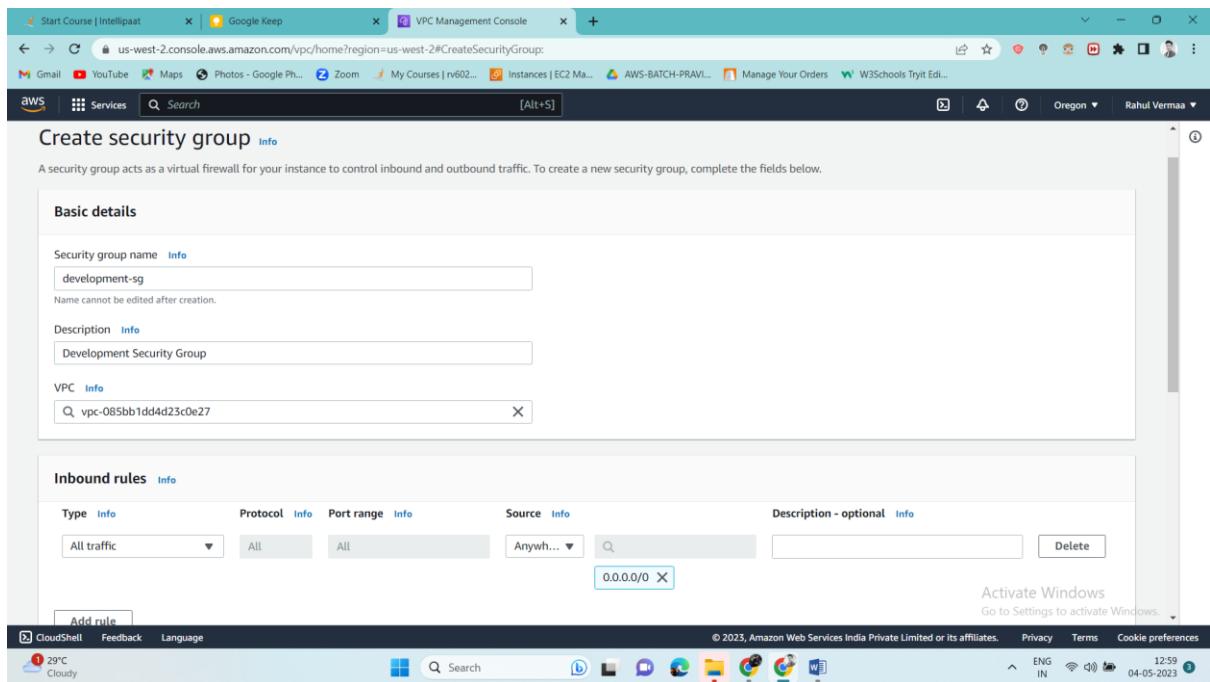
Selected subnets

subnet-036a0ef905789f06f / db-development

Cancel **Save associations**

Activate Windows
Go to Settings to activate Windows.

Step 23: After that we'll create security group



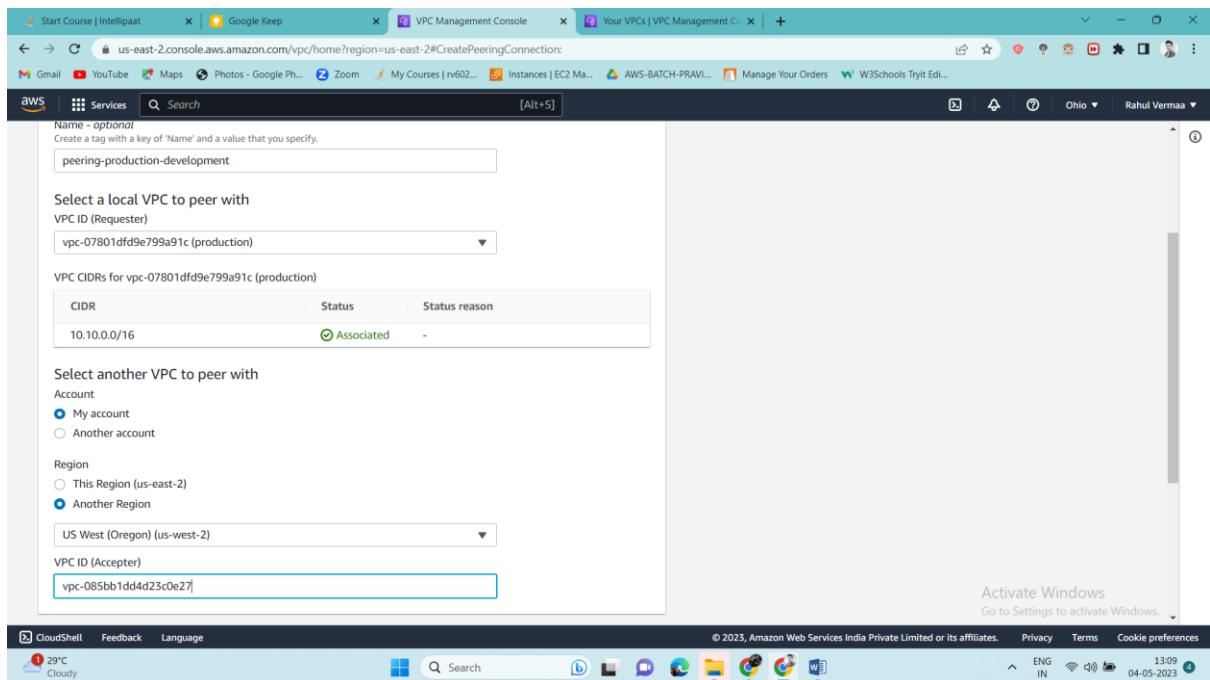
Step 24: Now will establish our peering connection

For that we'll go back to our Ohio region

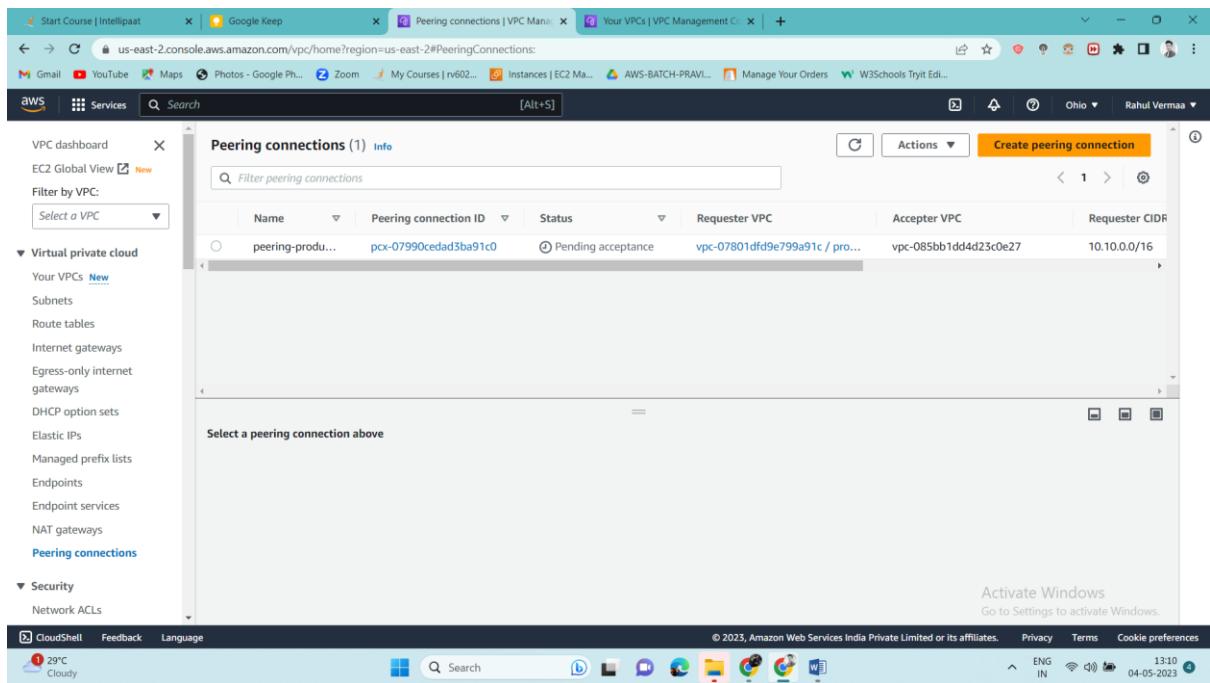
Just go to peering connection click on create button

Define it's name, here our requester will be Production and accepter is Development in Oregon region.

For accepter, we have to select another region option, as we have established our development network in Oregon region. After that we have to paste VPC id of our development VPC Oregon.

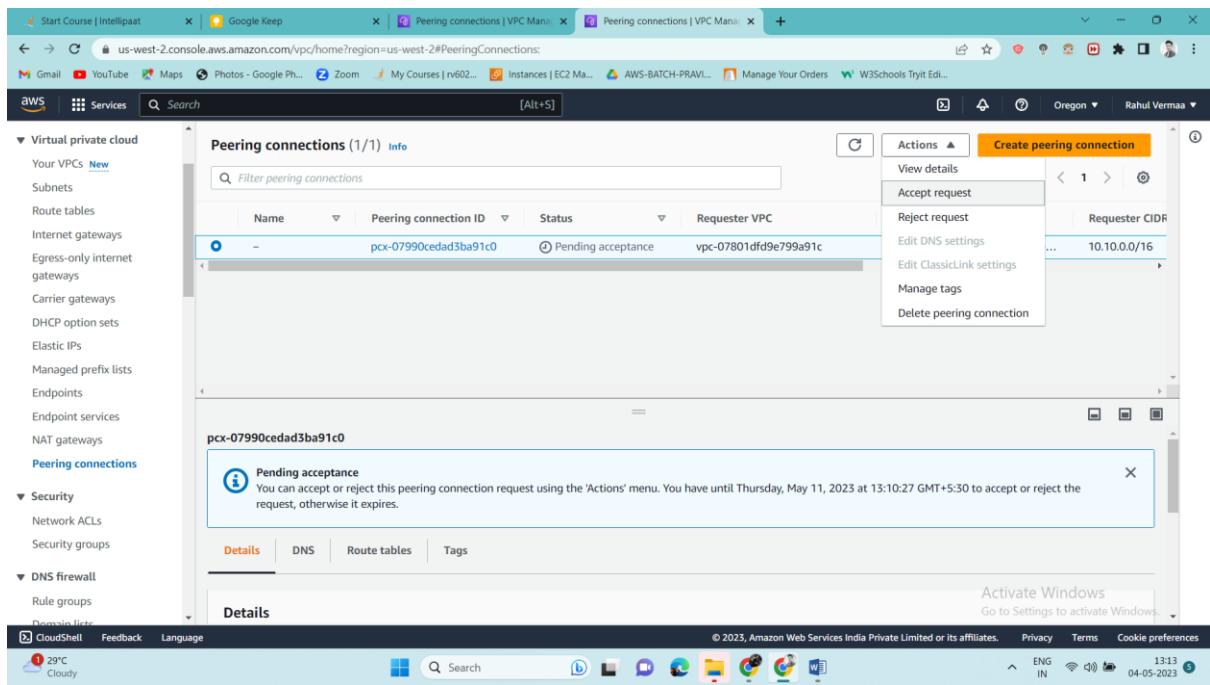


After creation it's status is showing pending acceptance

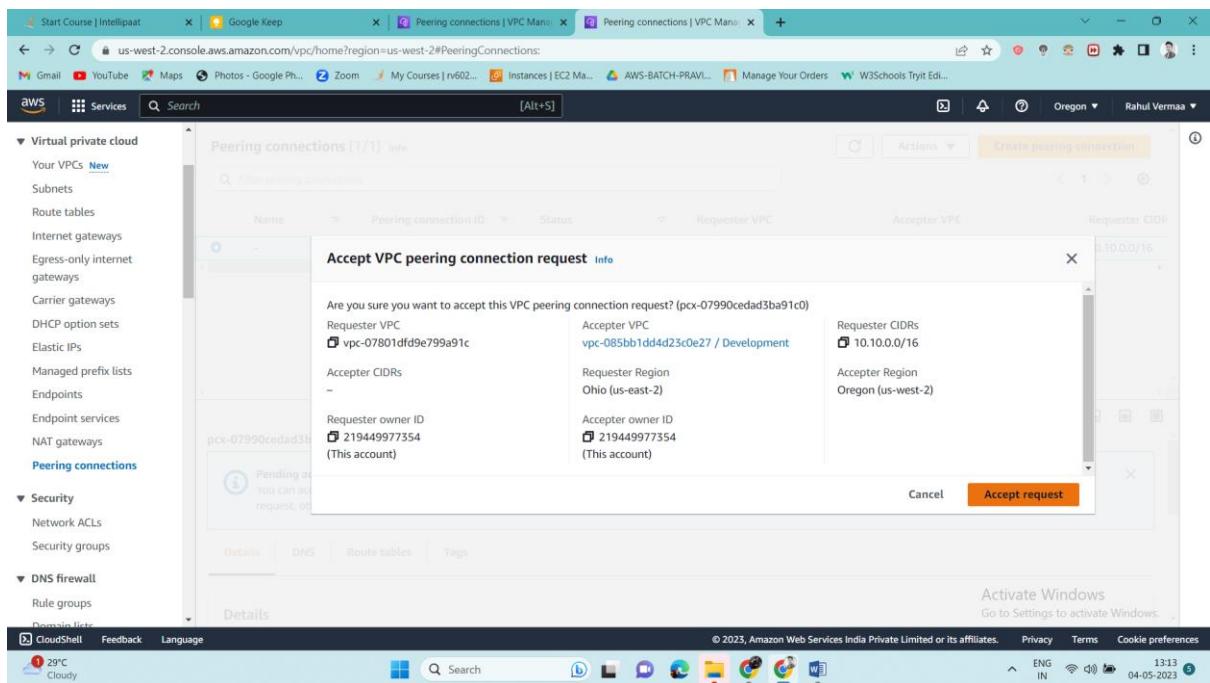


Step 25: For that we just have to go to our Oregon region

Then go to peering connection and select the peering connection click on action and accept the request.



The screenshot shows the AWS Management Console with the 'Peering connections' page open. A context menu is displayed for a peering connection named 'pcx-07990cedad3ba91c0' with the status 'Pending acceptance'. The 'Accept request' option is highlighted in the menu. The AWS navigation bar and various service links are visible at the top.



The screenshot shows the 'Accept VPC peering connection request' dialog box. It displays information about the request, including the Requester VPC (vpc-07801dfd9e799a91c) and Acceptor VPC (vpc-085bb1dd4d23c0e27 / Development). The 'Accept request' button is highlighted in orange. The AWS navigation bar and various service links are visible at the top.

Now it's status is showing Active

Your VPC peering connection (pcx-07990cedad3ba91c0) has been established.
To send and receive traffic across this VPC peering connection, you must add a route to the peered VPC in one or more of your VPC route tables. [Info](#)

Peering connections (1/1) [Info](#)

Name	Peering connection ID	Status	Requester VPC	Accepter VPC	Requester CIDR
pcx-07990cedad3ba91c0	pcx-07990cedad3ba91c0	Active	vpc-07801dfd9e799a91c	vpc-085bb1dd4d23c0e27 / De...	10.10.0.0/16

pcx-07990cedad3ba91c0

[Details](#) [DNS](#) [Route tables](#) [Tags](#)

Details

Step 26: Now we have to modify our route table

Just go to ohio region subnet and select dp-production copy it's CIDR

Now go to route table of Oregon region → select developmentPrivateRoute → go to route → Click on edit route paste it in destination and in target select peering connection.

Edit routes

Destination	Target	Status	Propagated
20.20.0.0/16	local	Active	No
10.10.5.0/24	pcx-07990cedad3ba91c0	-	No

[Add route](#) [Cancel](#) [Preview](#) [Save changes](#)

Updated routes for rtb-0fea09baf7cf47d94 / DevelopmentPrivateRoute successfully

rtb-0fea09baf7cf47d94 / DevelopmentPrivateRoute

Details

Route table ID: rtb-0fea09baf7cf47d94

Main: No

Owner ID: 219449977354

Explicit subnet associations: subnet-036a0ef905789f06f / db-development

Edge associations: -

Routes (2)

Both

Edit routes

Step 27: Similarly we have to do the same steps for requester also i.e Production, Just go to ohio region click on route table → select ProductionPrivateRouteWithoutInternet → go to routes click on edit route

In Destination paste db-development subnet CIDR and in target select peering connection.

Edit routes

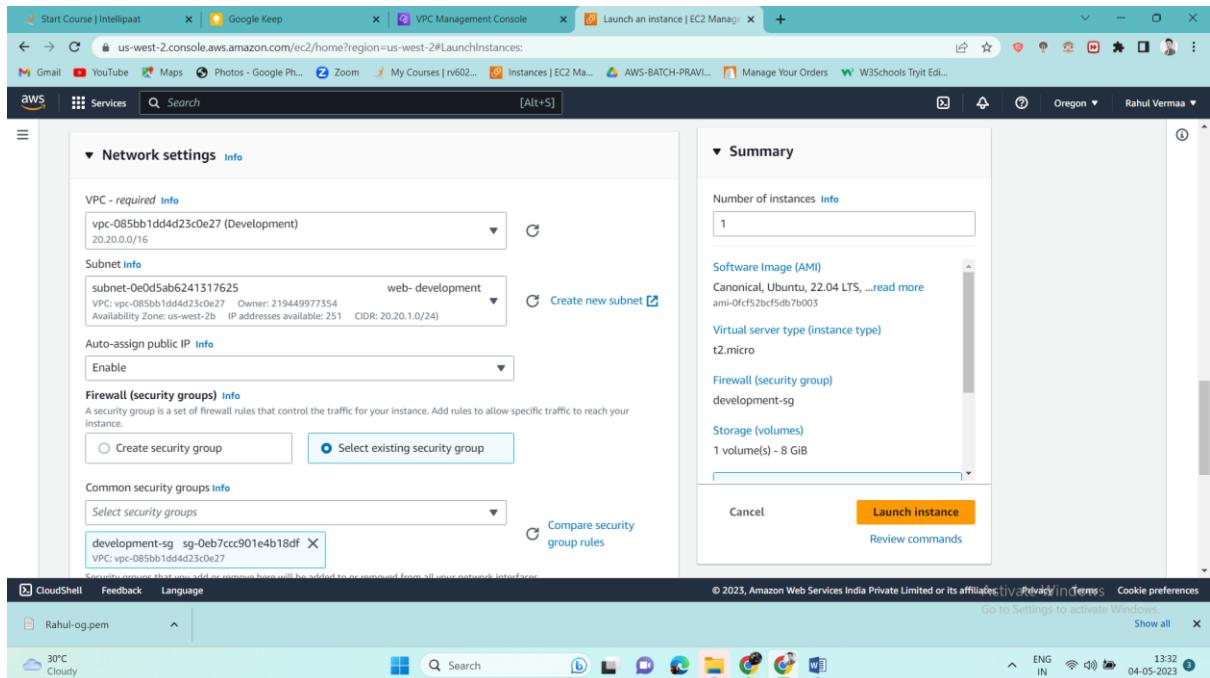
Destination	Target	Status	Propagated
10.10.0.0/16	local	Active	No
20.20.2.0/24	pcx-07990cedad3ba91c0 (peering-production-development)	-	No

Add route

Cancel Preview Save changes

Activate Windows
Go to Settings to activate Windows.

Step 28: Now we left with instances only, so will go to Oregon go ec2 and launch instance. Name it as **Web-development**

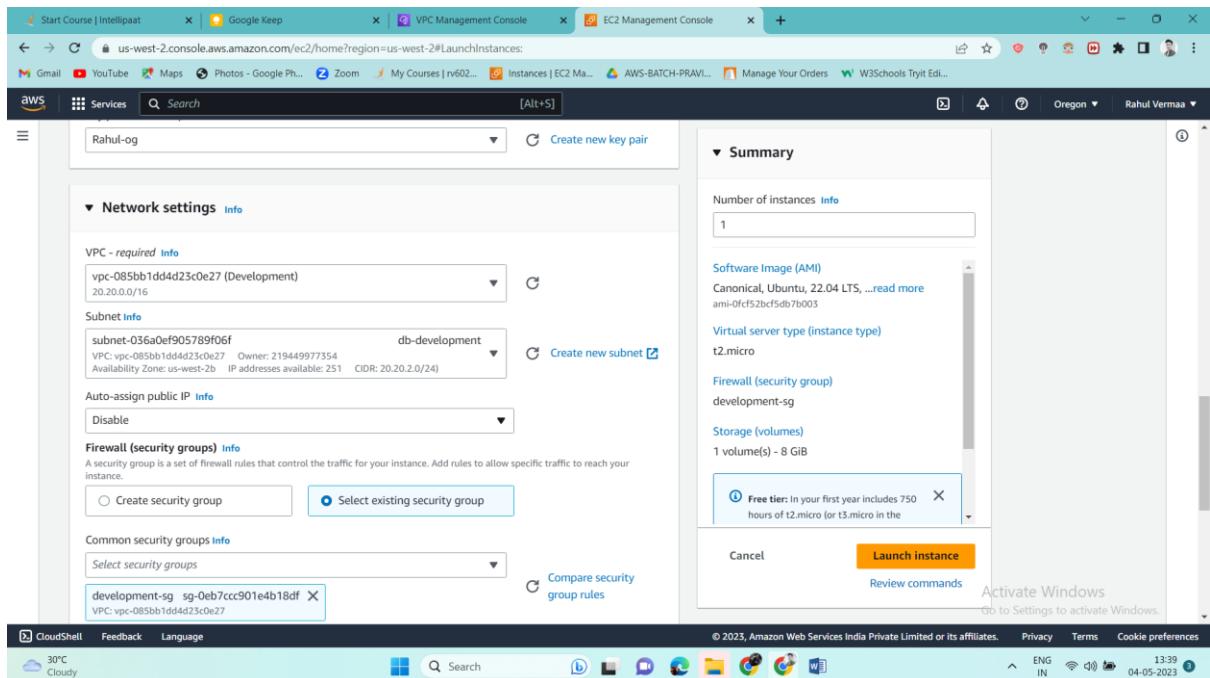


The screenshot shows the AWS EC2 Management Console with the 'Launch an Instance' wizard open. The instance configuration is as follows:

- VPC - required**: Development VPC (vpc-085bb1dd4d23c0e27)
- Subnet Info**: Subnet-0e5ab6241317625 (subnet-0e5ab6241317625) assigned to the 'web- development' security group. Availability Zone: us-west-2b, IP addresses available: 251, CIDR: 20.20.1.0/16.
- Auto-assign public IP**: Enabled.
- Firewall (security groups)**: Select existing security group (development-sg sg-0eb7cc901e4b18df).
- Common security groups**: development-sg sg-0eb7cc901e4b18df.

The 'Summary' section shows 1 instance, AMI: Canonical, Ubuntu, 22.04 LTS, Virtual server type: t2.micro, Firewall: development-sg, Storage: 1 volume(s) - 8 GiB. The 'Launch instance' button is highlighted in orange.

Same with second instance **db-development**

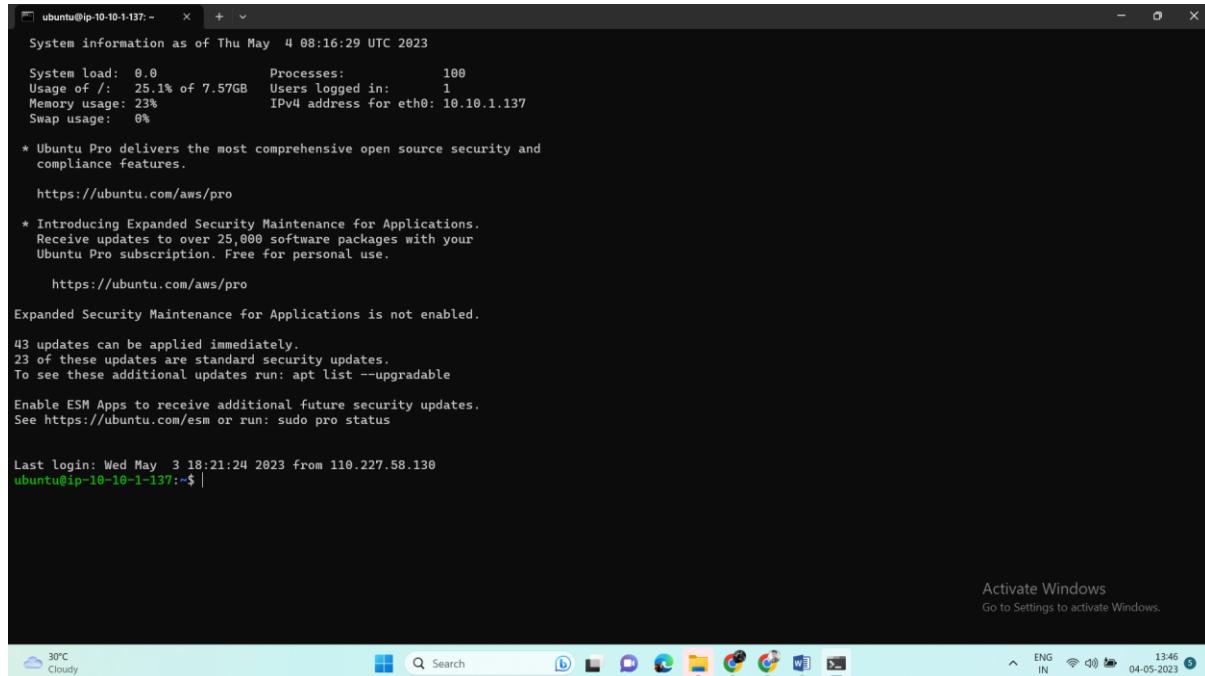


The screenshot shows the AWS EC2 Management Console with the 'Launch an Instance' wizard open. The instance configuration is as follows:

- VPC - required**: Development VPC (vpc-085bb1dd4d23c0e27)
- Subnet Info**: Subnet-036a0ef905789f06f (subnet-036a0ef905789f06f) assigned to the 'db-development' security group. Availability Zone: us-west-2b, IP addresses available: 251, CIDR: 20.20.2.0/24.
- Auto-assign public IP**: Disabled.
- Firewall (security groups)**: Select existing security group (development-sg sg-0eb7cc901e4b18df).
- Common security groups**: development-sg sg-0eb7cc901e4b18df.

The 'Summary' section shows 1 instance, AMI: Canonical, Ubuntu, 22.04 LTS, Virtual server type: t2.micro, Firewall: development-sg, Storage: 1 volume(s) - 8 GiB. A tooltip for the 'Free tier' is visible, stating 'In your first year includes 750 hours of t2.micro (or t3.micro in the'. The 'Launch instance' button is highlighted in orange.

Step 29: Now we need to verify our peering connection is successful or not for that we have to login through our instance. We are logged in to web-production Ohio

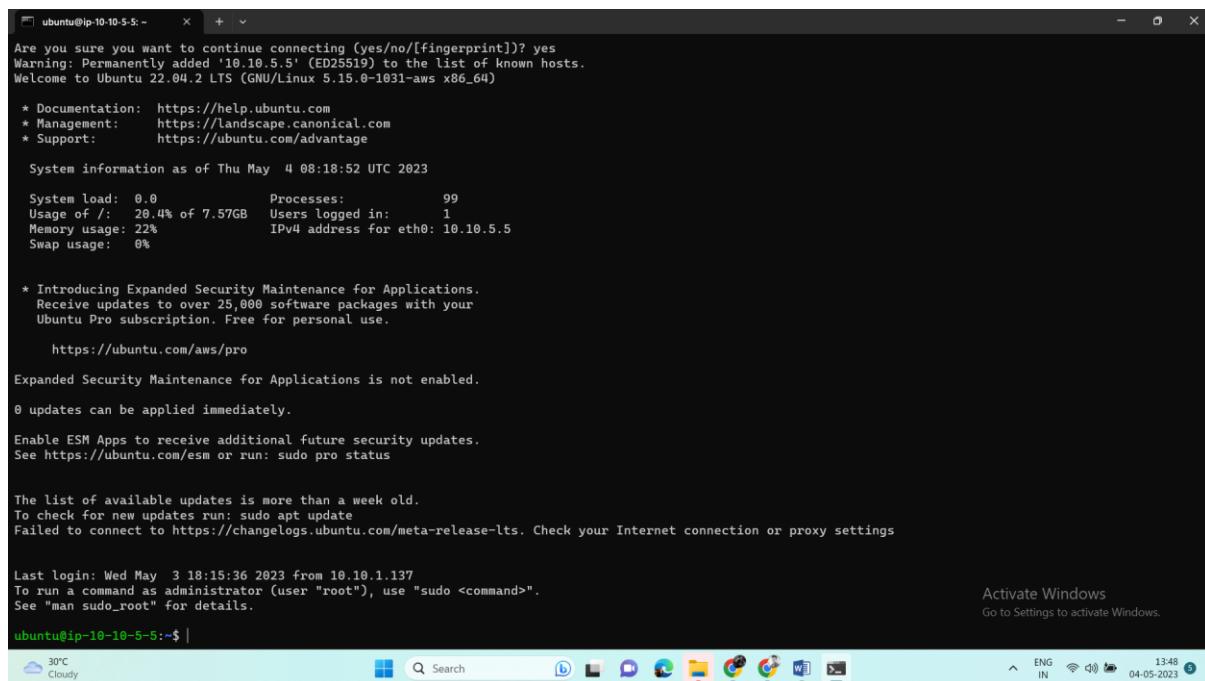


```
ubuntu@ip-10-1-137:~ % + ~
System information as of Thu May 4 08:16:29 UTC 2023
System load: 0.0 Processes: 100
Usage of /: 25.1% of 7.57GB Users logged in: 1
Memory usage: 23% IPv4 address for eth0: 10.10.1.137
Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.
  https://ubuntu.com/aws/pro
* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.
  https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
43 updates can be applied immediately.
23 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
Last login: Wed May 3 18:21:24 2023 from 110.227.58.130
ubuntu@ip-10-1-137:~$ |
```

Activate Windows
Go to Settings to activate Windows.

Cloudy 30°C Search ENG IN 13:46 04-05-2023

We are logged into our db-production



```
ubuntu@ip-10-10-5-5:~ % + ~
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.10.5.5' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

System information as of Thu May 4 08:18:52 UTC 2023
System load: 0.0 Processes: 99
Usage of /: 20.4% of 7.57GB Users logged in: 1
Memory usage: 22% IPv4 address for eth0: 10.10.5.5
Swap usage: 0%
* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.
  https://ubuntu.com/aws/pro
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status
The list of available updates is more than a week old.
To check for new updates run: sudo apt update
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

Last login: Wed May 3 18:15:36 2023 from 10.10.1.137
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
ubuntu@ip-10-10-5-5:~$ |
```

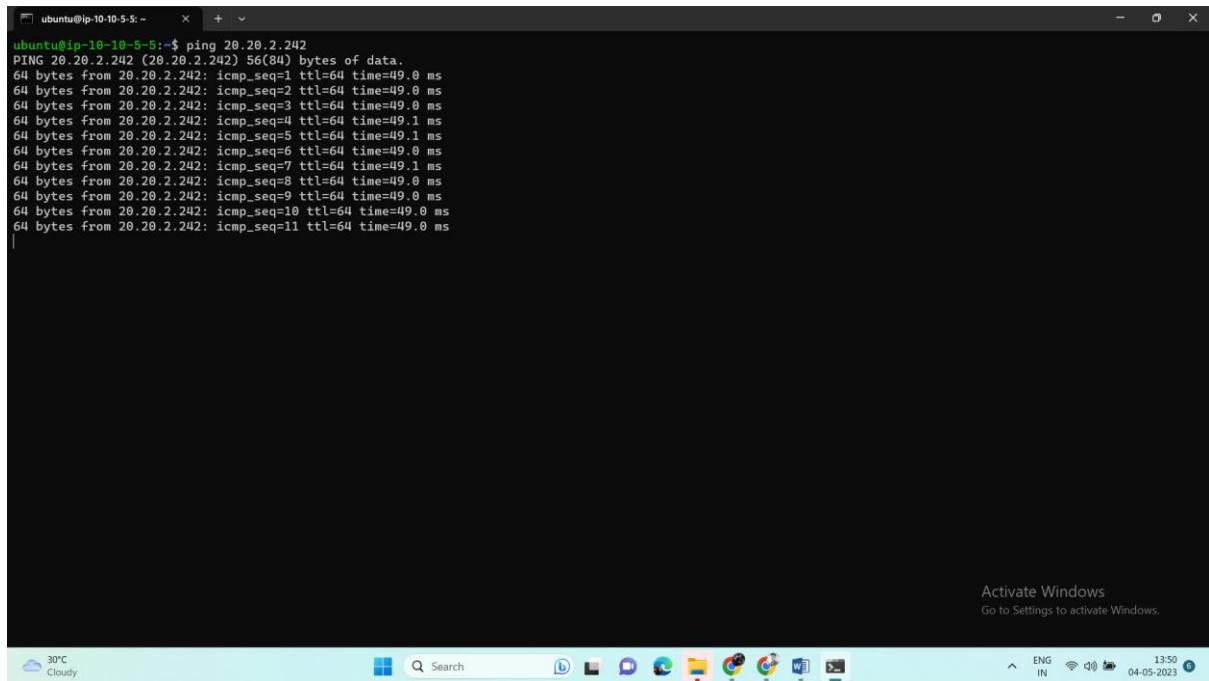
Activate Windows
Go to Settings to activate Windows.

Cloudy 30°C Search ENG IN 13:48 04-05-2023

Now will try to ping db-development oregon

And as you can see it's successfully done. That means we are able to communicate between two VPC.

From db-production Ohio 10.10.5.5 to db-development Oregon 20.20.2.242

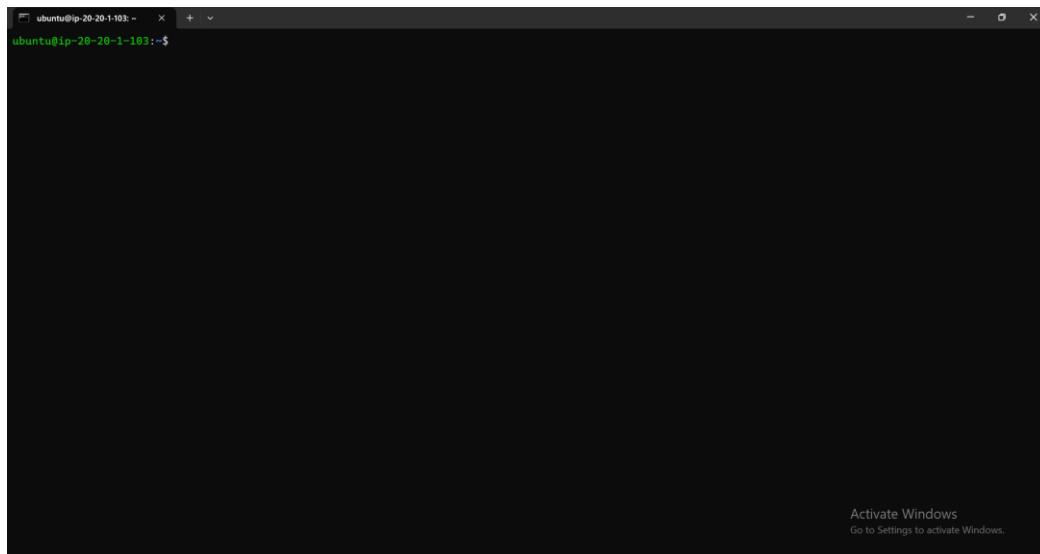


```
ubuntu@ip-10-10-5-5:~$ ping 20.20.2.242
PING 20.20.2.242 (20.20.2.242) 56(84) bytes of data.
64 bytes from 20.20.2.242: icmp_seq=1 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=2 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=3 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=4 ttl=64 time=49.1 ms
64 bytes from 20.20.2.242: icmp_seq=5 ttl=64 time=49.1 ms
64 bytes from 20.20.2.242: icmp_seq=6 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=7 ttl=64 time=49.1 ms
64 bytes from 20.20.2.242: icmp_seq=8 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=9 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=10 ttl=64 time=49.0 ms
64 bytes from 20.20.2.242: icmp_seq=11 ttl=64 time=49.0 ms
```

Activate Windows
Go to Settings to activate Windows.

30°C Cloudy Search ENG IN 13:50 04-05-2023

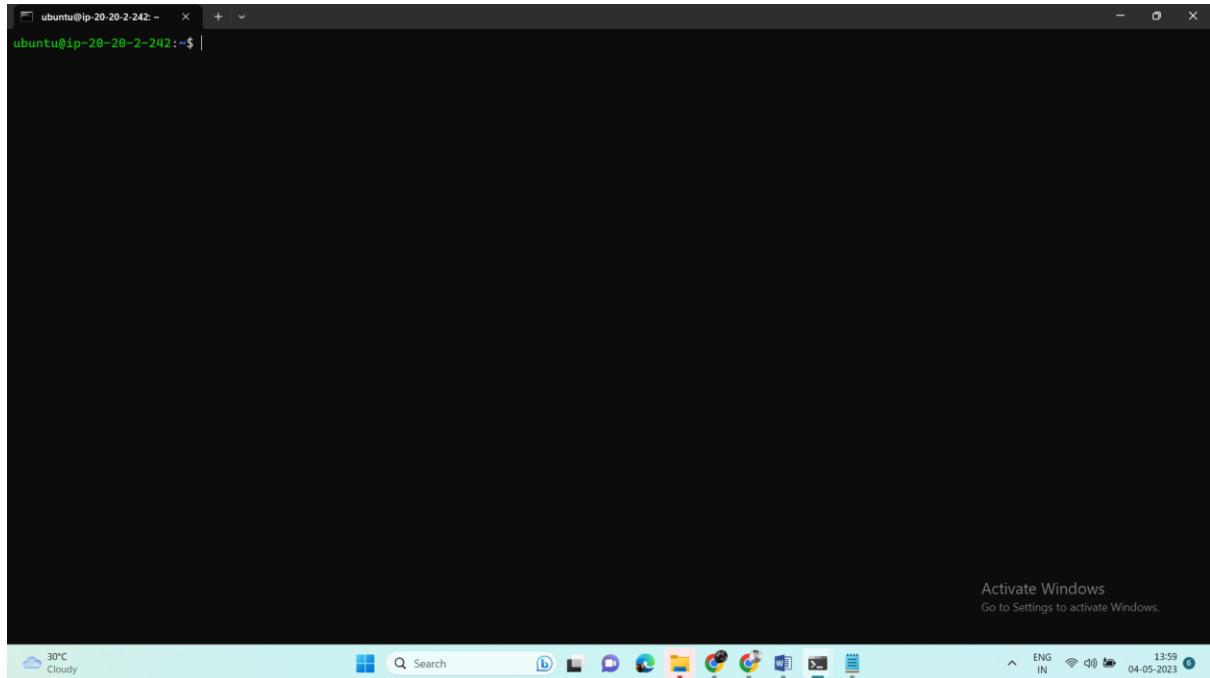
Step 30: Now we will try the same thing from our web-development Oregon instance.



```
ubuntu@ip-20-20-1-103:~$
```

Activate Windows
Go to Settings to activate Windows.

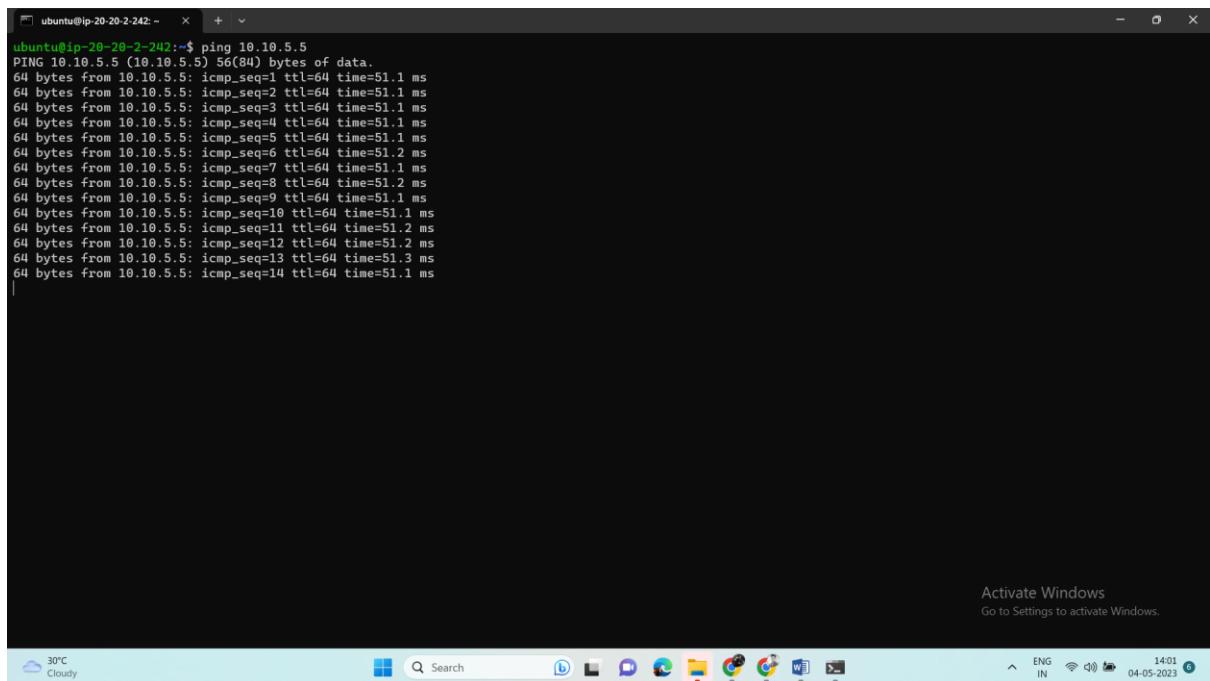
We are logged into db-development



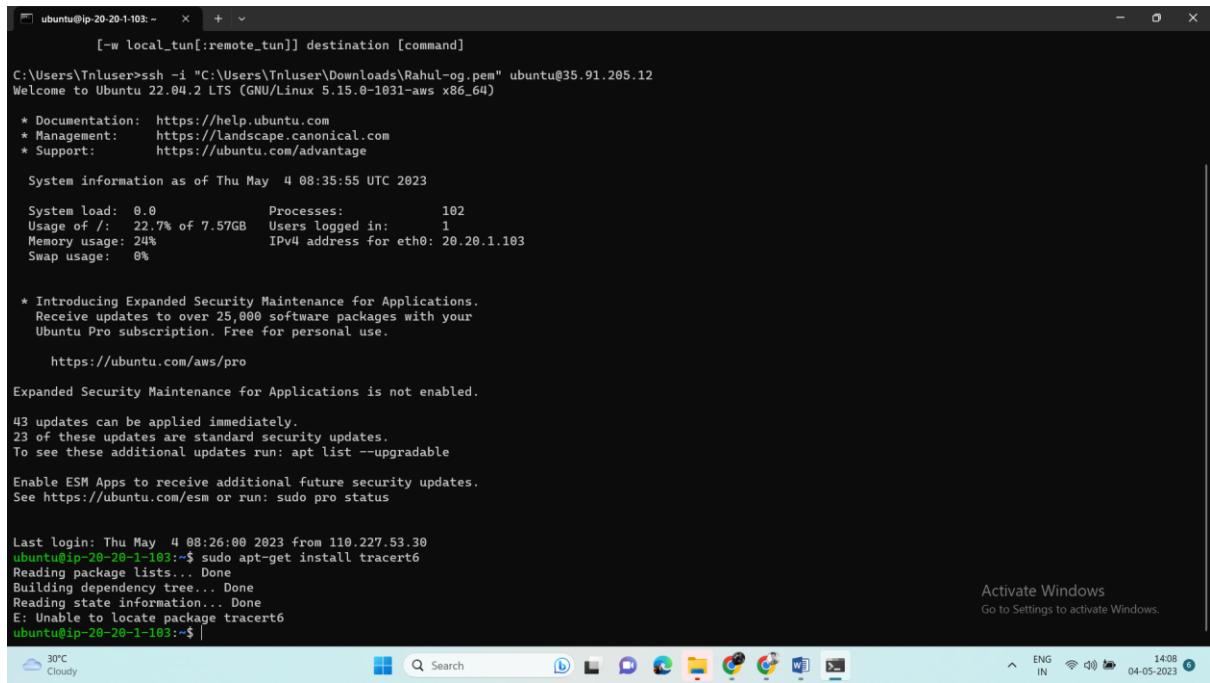
```
ubuntu@ip-20-20-2-242:~$
```

Now from deb-development oregon will ping db-production ohio

i.e from 20.20.2.242 to 10.10.5.5 we are able to communicate



```
ubuntu@ip-20-20-2-242:~$ ping 10.10.5.5
PING 10.10.5.5 (10.10.5.5) 56(84) bytes of data.
64 bytes from 10.10.5.5: icmp_seq=1 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=2 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=3 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=4 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=5 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=6 ttl=64 time=51.2 ms
64 bytes from 10.10.5.5: icmp_seq=7 ttl=64 time=51.2 ms
64 bytes from 10.10.5.5: icmp_seq=8 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=9 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=10 ttl=64 time=51.1 ms
64 bytes from 10.10.5.5: icmp_seq=11 ttl=64 time=51.2 ms
64 bytes from 10.10.5.5: icmp_seq=12 ttl=64 time=51.2 ms
64 bytes from 10.10.5.5: icmp_seq=13 ttl=64 time=51.3 ms
64 bytes from 10.10.5.5: icmp_seq=14 ttl=64 time=51.1 ms
```



ubuntu@ip-20-20-1-103:~ + -

```
[-w local_tun[:remote_tun]] destination [command]

C:\Users\Tnuser>ssh -i "C:\Users\Tnuser\Downloads\Rahul-og.pem" ubuntu@35.91.205.12
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Thu May  4 08:35:55 UTC 2023

System load: 0.0      Processes:          102
Usage of /: 22.7% of 7.57GB  Users logged in: 1
Memory usage: 24%          IPv4 address for eth0: 20.20.1.103
Swap usage:  0%

* Introducing Expanded Security Maintenance for Applications.
  Receive updates to over 25,000 software packages with your
  Ubuntu Pro subscription. Free for personal use.

  https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

43 updates can be applied immediately.
23 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Thu May  4 08:26:00 2023 from 110.227.53.30
ubuntu@ip-20-20-1-103:~$ sudo apt-get install traceroute
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
E: Unable to locate package traceroute
ubuntu@ip-20-20-1-103:~$ |
```

Activate Windows
Go to Settings to activate Windows.

Cloudy 30°C

Search

14:08 04-05-2023 ENG IN

Note: To avoid charges delete everything after use.

Thank You