

Case Study on cloudformation

Problem Statement:

You work for XYZ Corporation. Your company wants to launch a new web-based application. The development team has prepared the code, but it is not tested yet. The development team needs System Admins to build a web server to test the code, but the System Admins are not available.

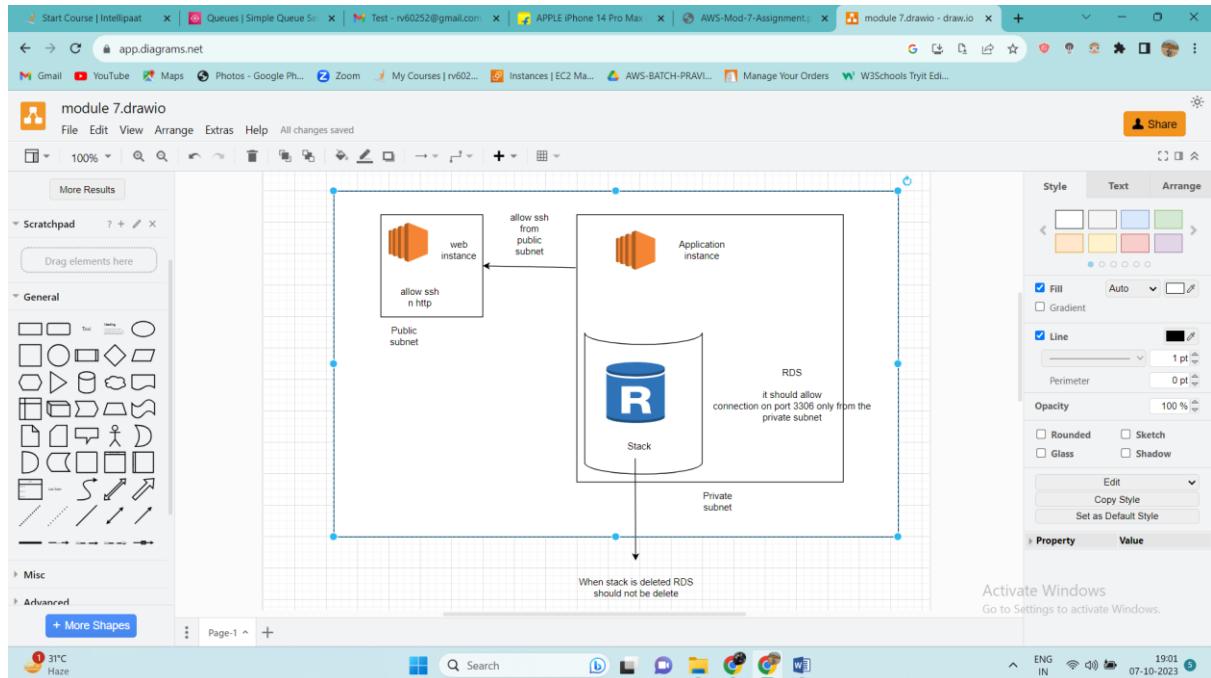
You are asked to create three-tier architecture and perform following tasks in each tier:

1. Web tier: Launch an instance in the public subnet so that the instance should allow HTTP and SSH from the Internet
2. Application tier: Launch an instance in the private subnet of the web tier, and it should allow only SSH from the public subnet of the web tier
3. DB tier: Launch an RDS MySQL instance in the private subnet, and it should allow connection on port 3306 only from the private subnet of the application tier
4. Setup a Route 53 hosted zone, and direct the traffic to the EC2 instance

You are also asked to propose a solution so that:

1. The development team can test the code without having to involve the System Admins and can invest their time in testing the code rather than provisioning, configuring, and updating the resources needed to test the code
2. When the development team deletes the stack, the RDS DB instance should not be deleted.

So this is how our entire case study looks like



First we will create one VPC

VPC dashboard

Create VPC

Launch EC2 Instances

Resources by Region

Category	Count	Region
VPCs	1	Asia Pacific
NAT Gateways	0	Asia Pacific
Subnets	3	Asia Pacific
VPC Peering Connections	0	Asia Pacific
Route Tables	1	Asia Pacific
Network ACLs	1	Asia Pacific
Internet Gateways	1	Asia Pacific
Security Groups	1	Asia Pacific
Egress-only Internet Gateways	0	Asia Pacific
Customer Gateways	0	Asia Pacific
DHCP option sets	1	Asia Pacific
Virtual Private Gateways	0	Asia Pacific

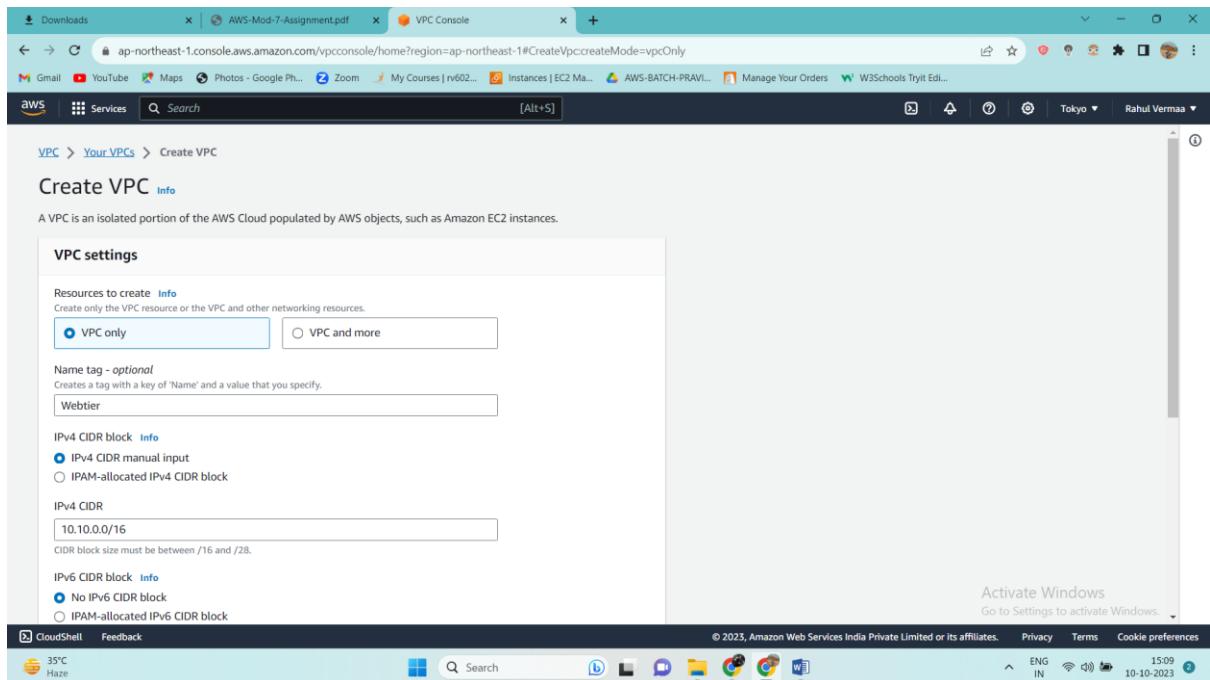
Service Health

Settings

Additional Information

AWS Network Manager

Select options and click it on create VPC button



VPC settings

Resources to create [Info](#)
Create only the VPC resource or the VPC and other networking resources.

VPC only VPC and more

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

Webtier

IPv4 CIDR block [Info](#)
 IPv4 CIDR manual input IPAM-allocated IPv4 CIDR block

10.10.0.0/16

IPv6 CIDR block [Info](#)
 No IPv6 CIDR block IPAM-allocated IPv6 CIDR block

CloudShell Feedback

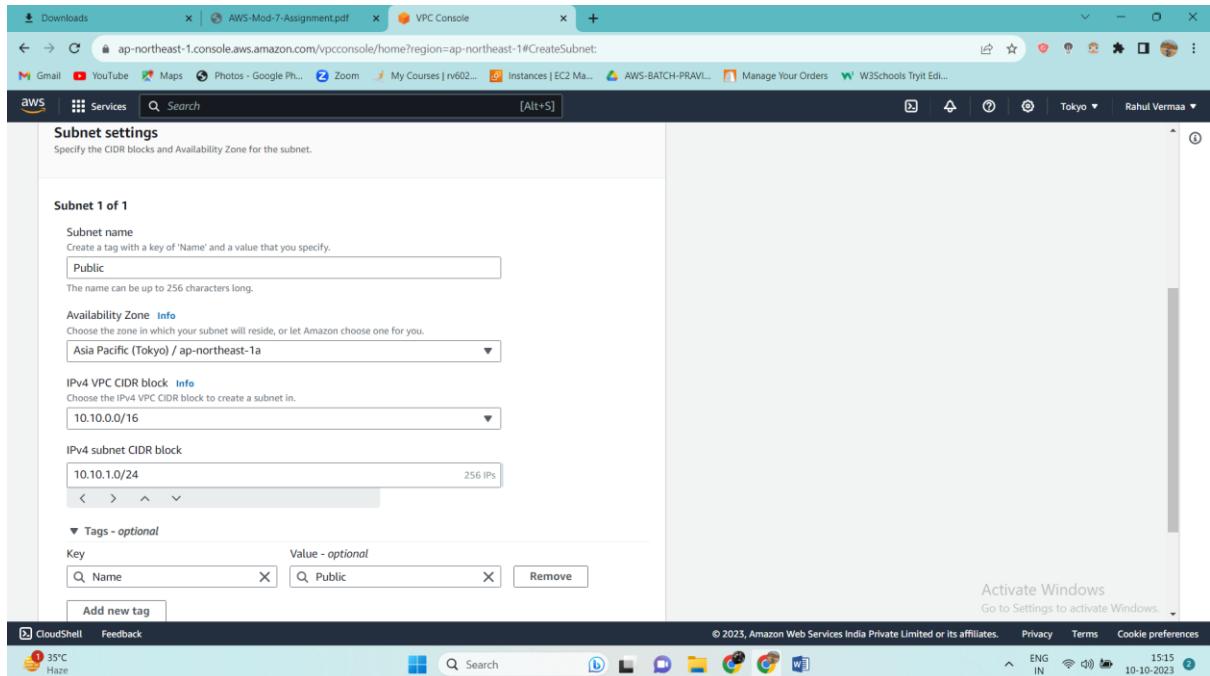
Activate Windows
Go to Settings to activate Windows.

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

35°C Haze 15:09 10-10-2023

Now we will create 2 subnets public and private

Select options and create your public subnet



Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

Public

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

Asia Pacific (Tokyo) / ap-northeast-1a

IPv4 VPC CIDR block [Info](#)
Choose the IPv4 VPC CIDR block to create a subnet in.

10.10.0.0/16

IPv4 subnet CIDR block
10.10.1.0/24 256 IPs

Tags - *optional*

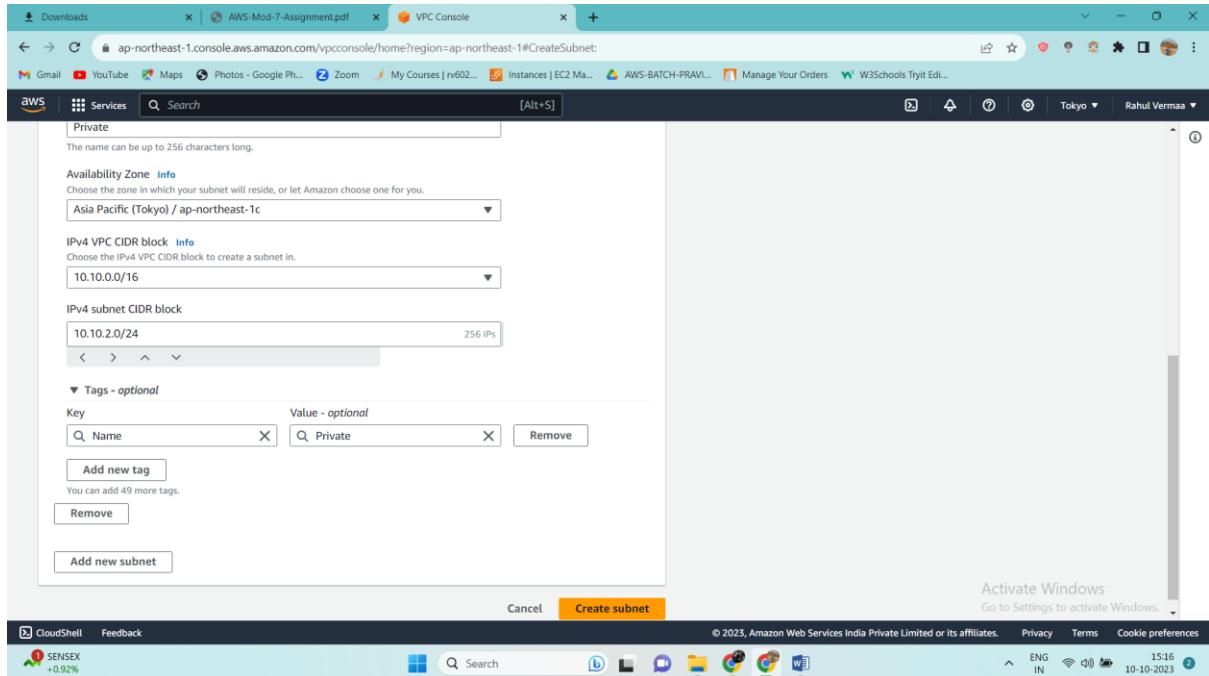
Key: Name Value: Public Remove Add new tag

Activate Windows
Go to Settings to activate Windows.

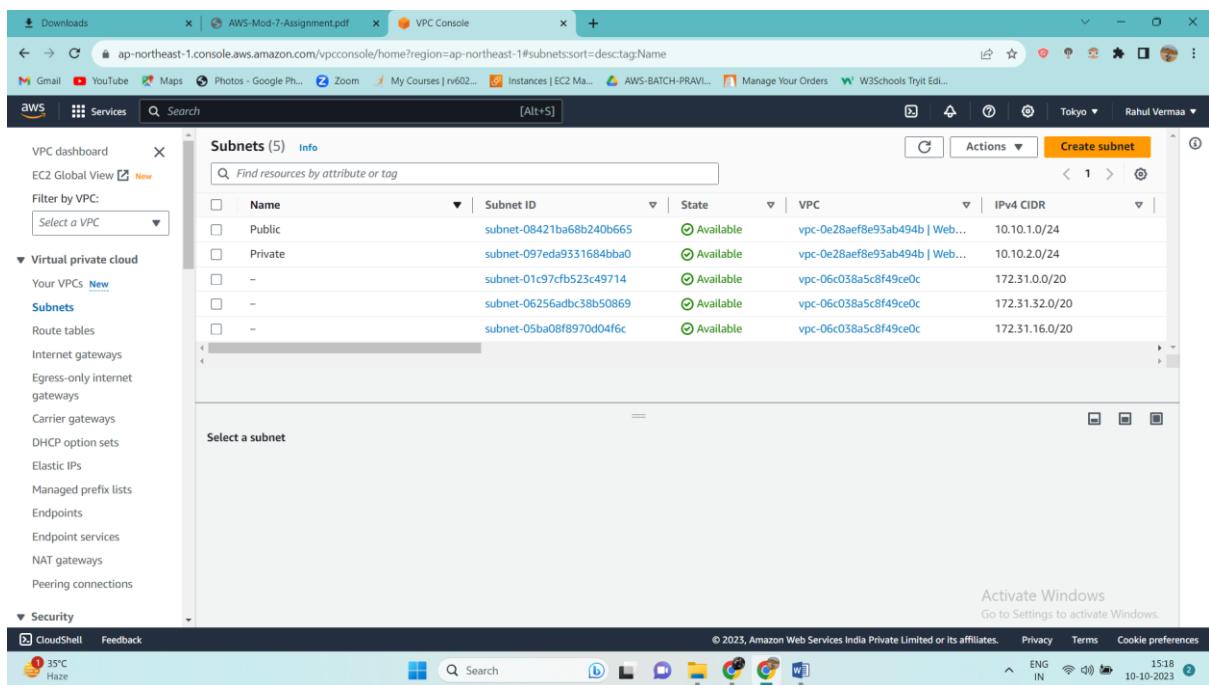
© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

35°C Haze 15:15 10-10-2023

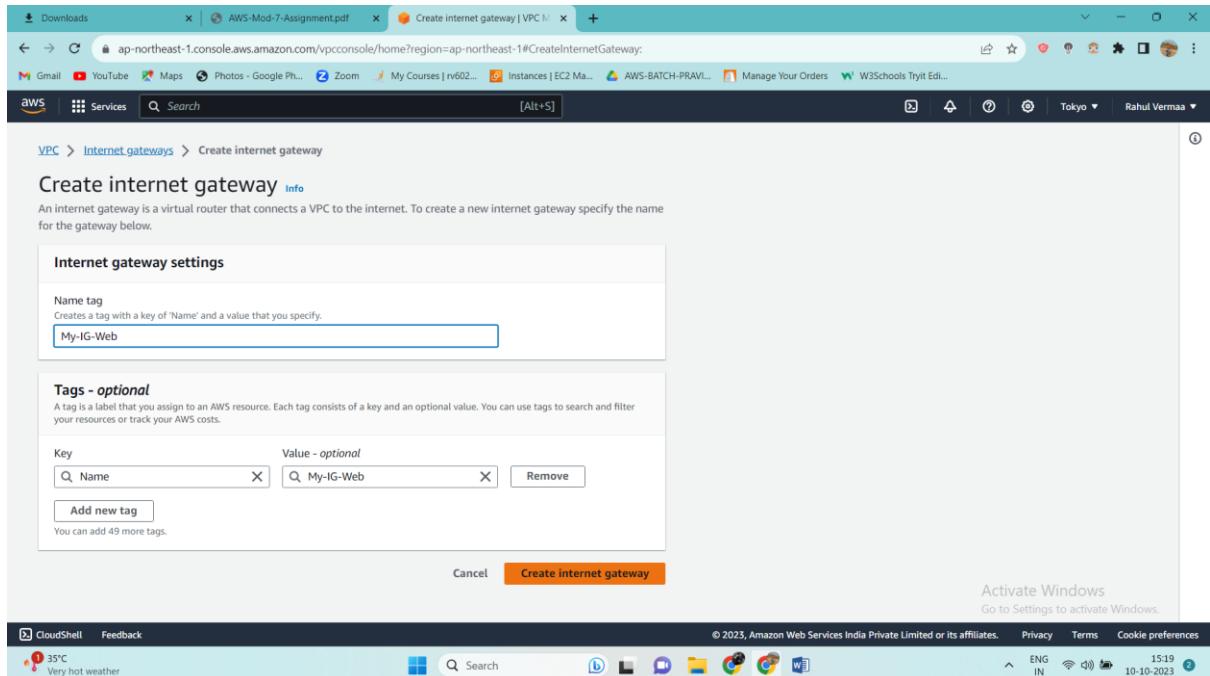
Now create second subnet you can click on add subnet button and create multiple subnets at once. But to take screen shots I have create them separately.



Both are created now



Now we need to install one internet gateway



Internet gateway settings

Name tag

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

My-IG-Web

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 Value - optional

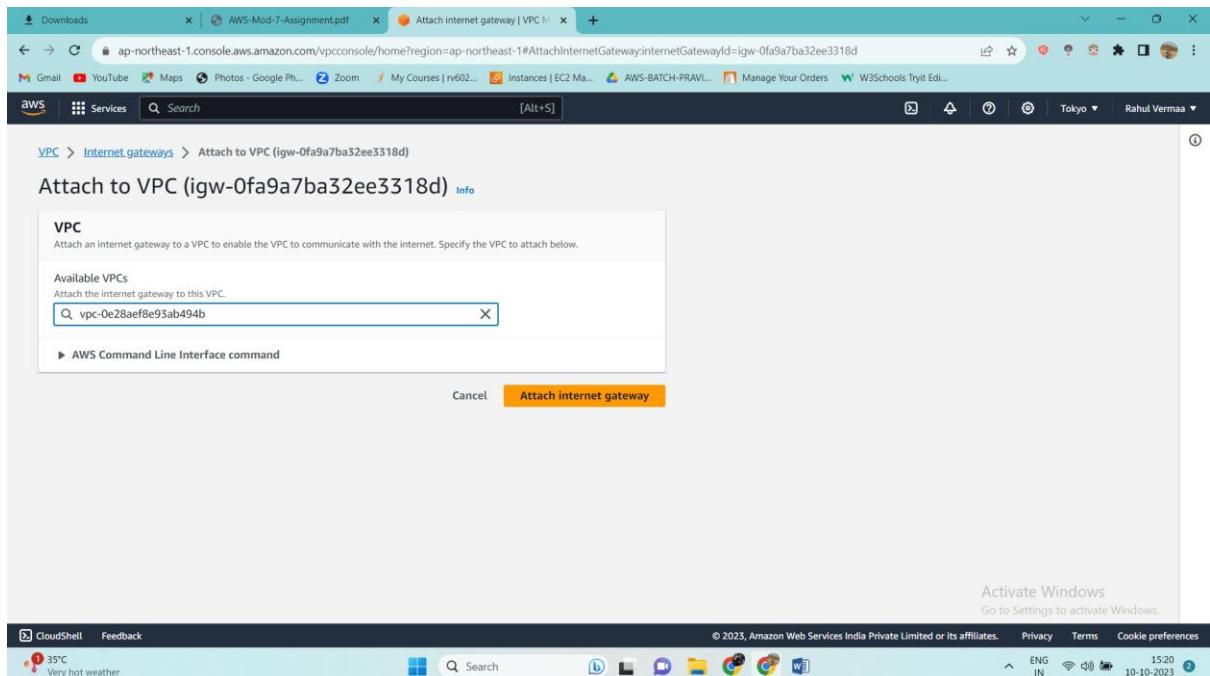
Name My-IG-Web Remove

Add new tag

You can add 49 more tags.

Create internet gateway

Created successfully and now attach it to our VPC



VPC

Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs

Attach the internet gateway to this VPC.

vpc-0e28aef8e95ab494b

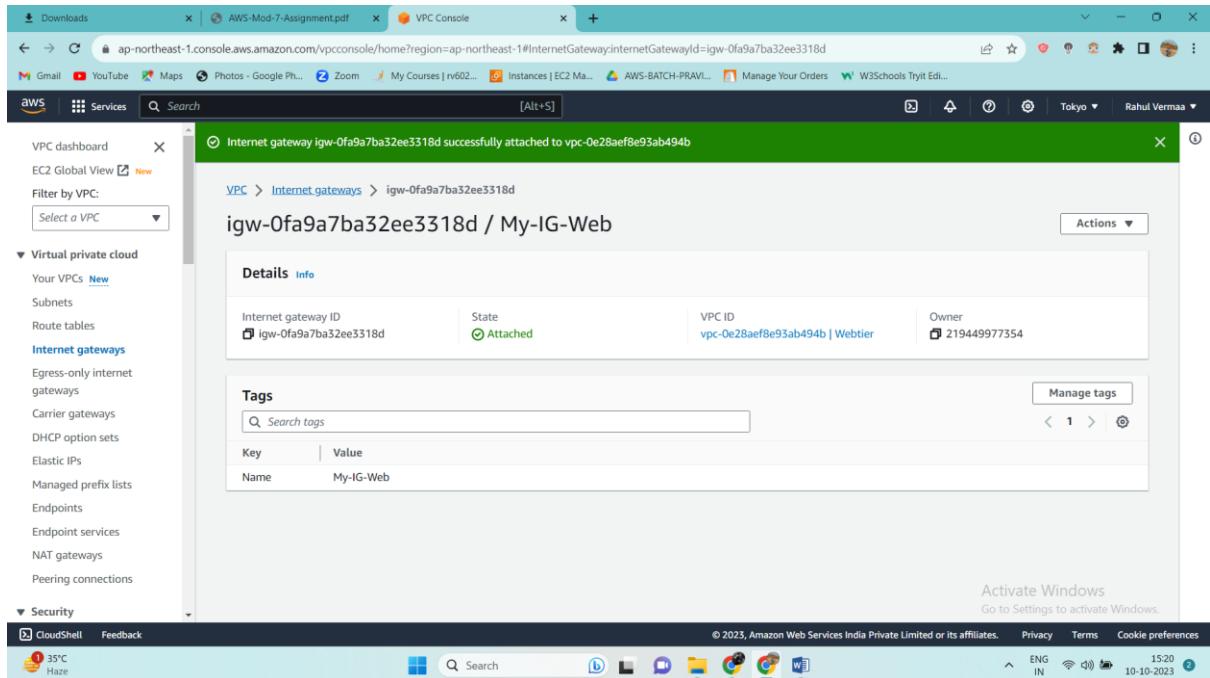
AWS Command Line Interface command

Attach internet gateway

Activate Windows

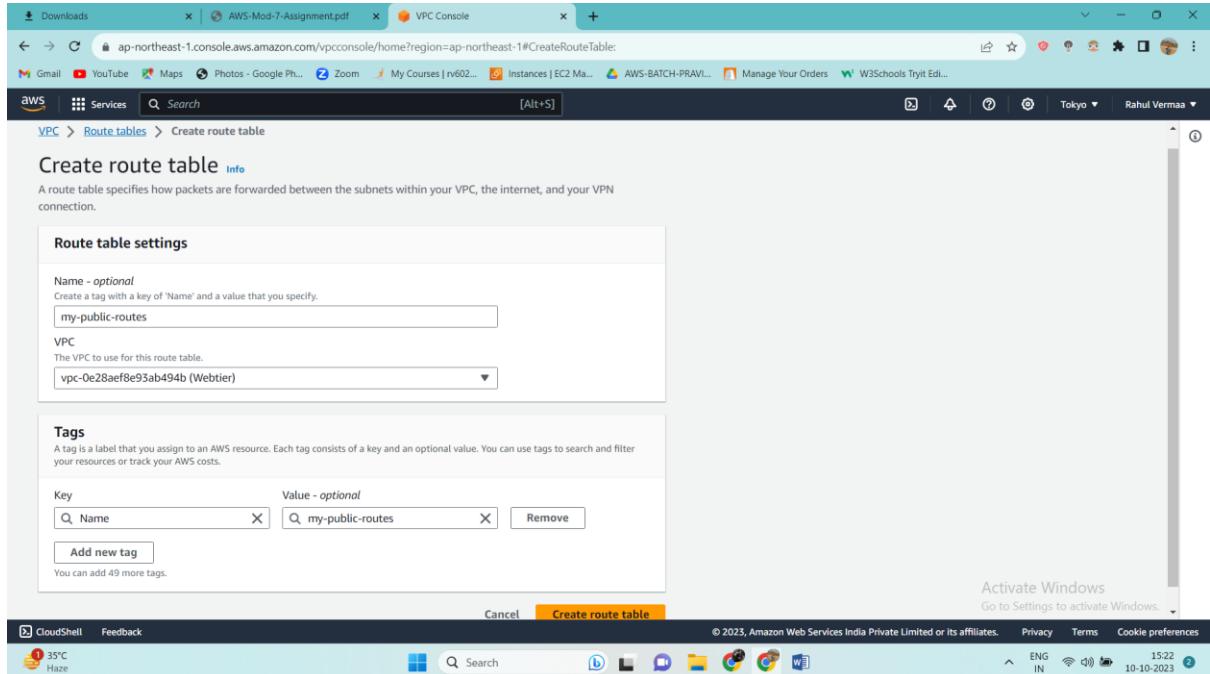
Go to Settings to activate Windows.

It's attached to our VPC successfully



The screenshot shows the AWS VPC Console interface. On the left, a sidebar lists various VPC components: Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections). The Internet gateways section is currently selected. The main content area displays a success message: "Internet gateway igw-0fa9a7ba32ee3318d successfully attached to vpc-0e28aef8e93ab494b". Below this, the "igw-0fa9a7ba32ee3318d / My-IG-Web" Internet gateway is shown in detail. The "Details" tab is selected, showing the Internet gateway ID (igw-0fa9a7ba32ee3318d), State (Attached), VPC ID (vpc-0e28aef8e93ab494b | Webtier), and Owner (219449977354). The "Tags" tab shows a single tag: Name (My-IG-Web). The bottom of the screen shows the Windows taskbar with the date and time (10-10-2023, 15:20), system icons, and the AWS CloudShell icon.

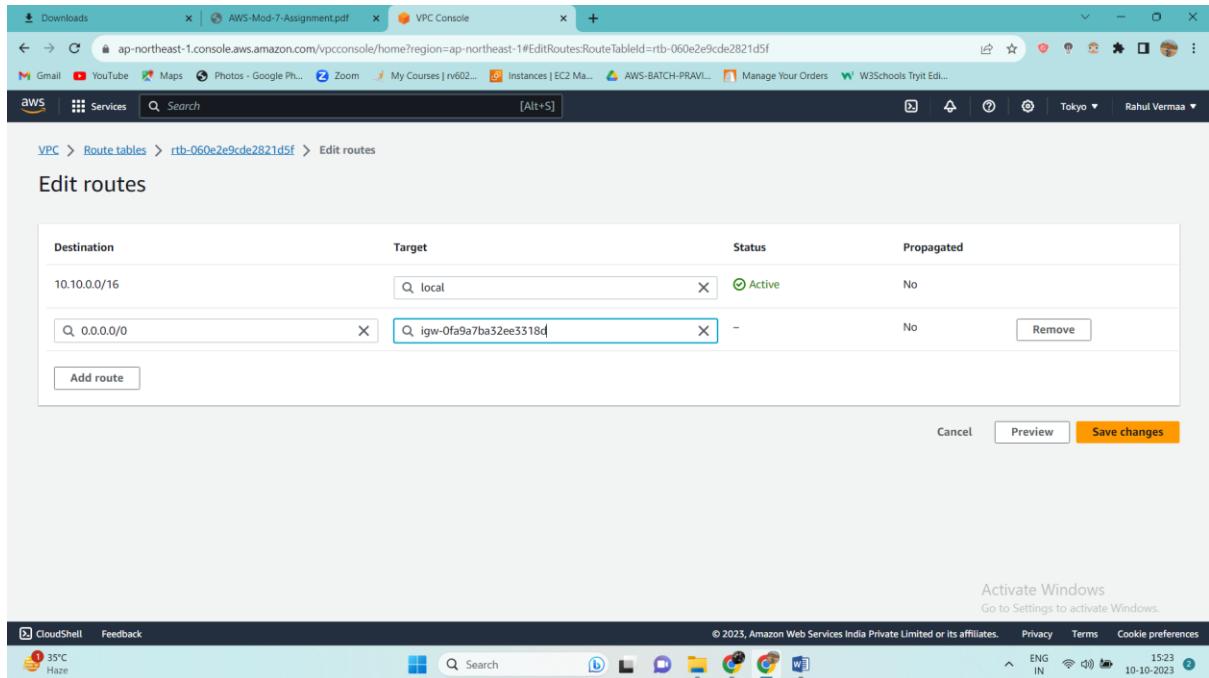
Now we will create one route table so that only public subnet is able to get the internet connection



The screenshot shows the AWS VPC Console interface. The left sidebar shows the "Route tables" section. The main content area is titled "Create route table" with an "Info" link. It explains that a route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection. The "Route table settings" section includes a "Name - optional" field with "my-public-routes" entered, and a "VPC" dropdown set to "vpc-0e28aef8e93ab494b (Webtier)". The "Tags" section allows adding tags, with one tag "Name" set to "my-public-routes". At the bottom, there are "Cancel" and "Create route table" buttons. The bottom of the screen shows the Windows taskbar with the date and time (10-10-2023, 15:22), system icons, and the AWS CloudShell icon.

Now edit routes in route table and save changes

Route it to internet gateway



aws Services Search [Alt+S] Downloads AWS-Mod-7-Assessment.pdf VPC Console

ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#EditRoutes:RouteTableId=rtb-060e2e9cde2821d5f

Gmail YouTube Maps Photos - Google Ph... Zoom My Courses | rv602... Instances | EC2 Ma... AWS-BATCH-PRAVI... Manage Your Orders W3Schools Tryit Edi...

VPC > Route tables > rtb-060e2e9cde2821d5f > Edit routes

Edit routes

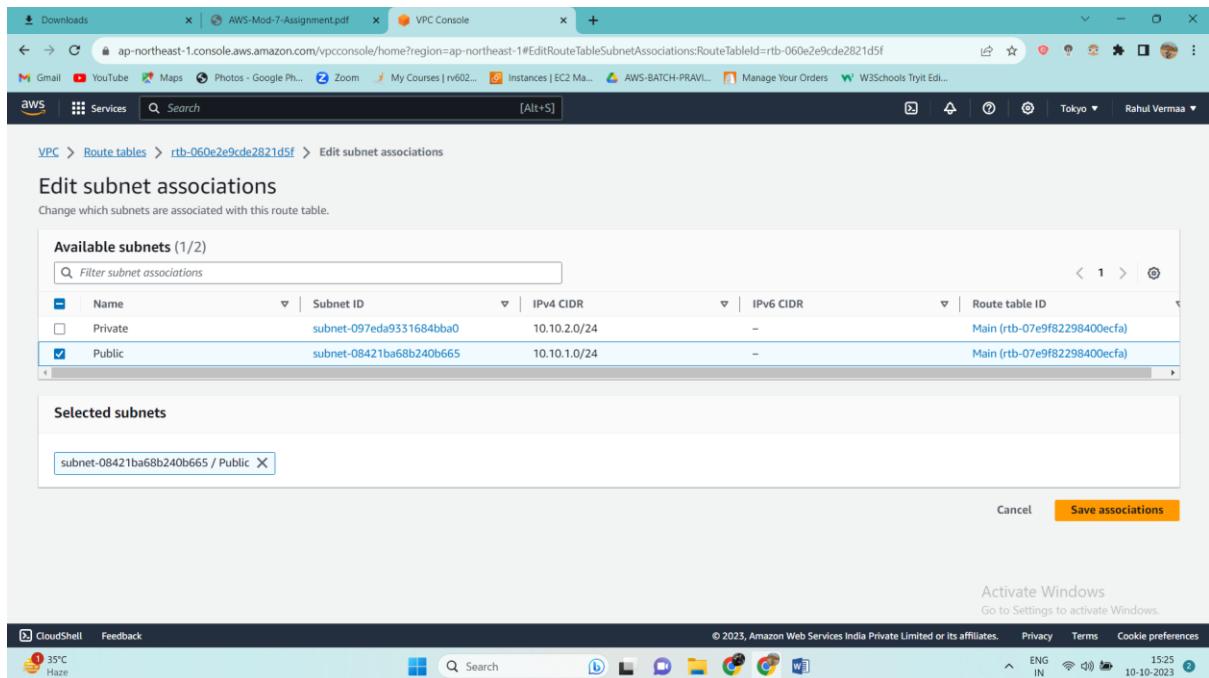
Destination	Target	Status	Propagated
10.10.0.0/16	local	Active	No
0.0.0.0/0	igw-0fa9a7ba32ee3318d	-	No

Add route Cancel Preview Save changes

Activate Windows Go to Settings to activate Windows.

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Haze 15:23 10-10-2023

Now attach public subnet to our route table



aws Services Search [Alt+S] Downloads AWS-Mod-7-Assessment.pdf VPC Console

ap-northeast-1.console.aws.amazon.com/vpcconsole/home?region=ap-northeast-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-060e2e9cde2821d5f

Gmail YouTube Maps Photos - Google Ph... Zoom My Courses | rv602... Instances | EC2 Ma... AWS-BATCH-PRAVI... Manage Your Orders W3Schools Tryit Edi...

VPC > Route tables > rtb-060e2e9cde2821d5f > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)				
Filter subnet associations				
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
Private	subnet-097eda9331684bba0	10.10.2.0/24	-	Main (rtb-07e9fb2298400ecfa)
Public	subnet-08421ba68b240b665	10.10.1.0/24	-	Main (rtb-07e9fb2298400ecfa)

Selected subnets

subnet-08421ba68b240b665 / Public

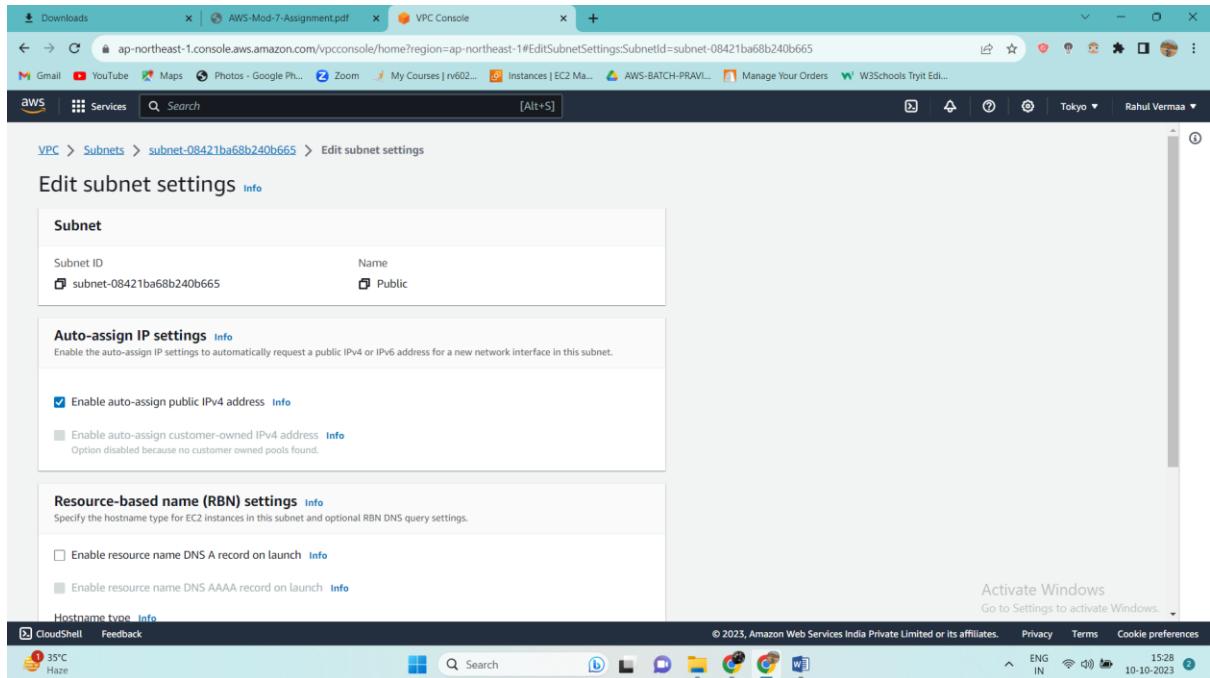
Cancel Save associations

Activate Windows Go to Settings to activate Windows.

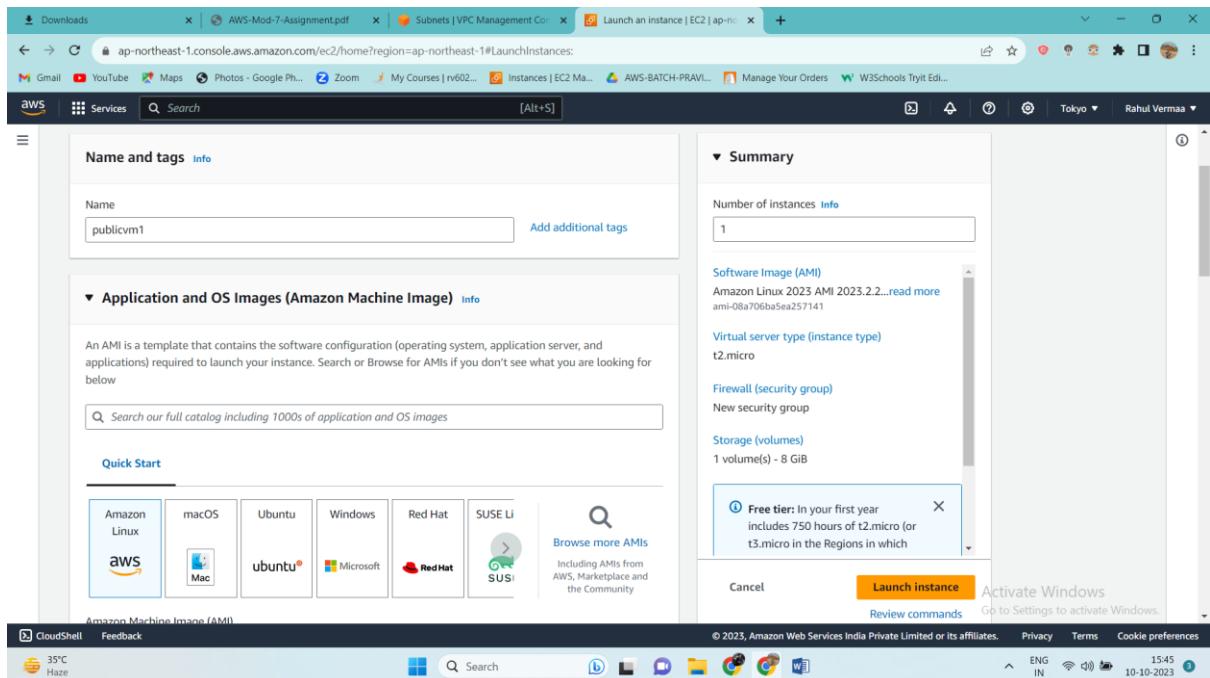
CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Haze 15:25 10-10-2023

Now we have to enable ipv4 config in public subnet

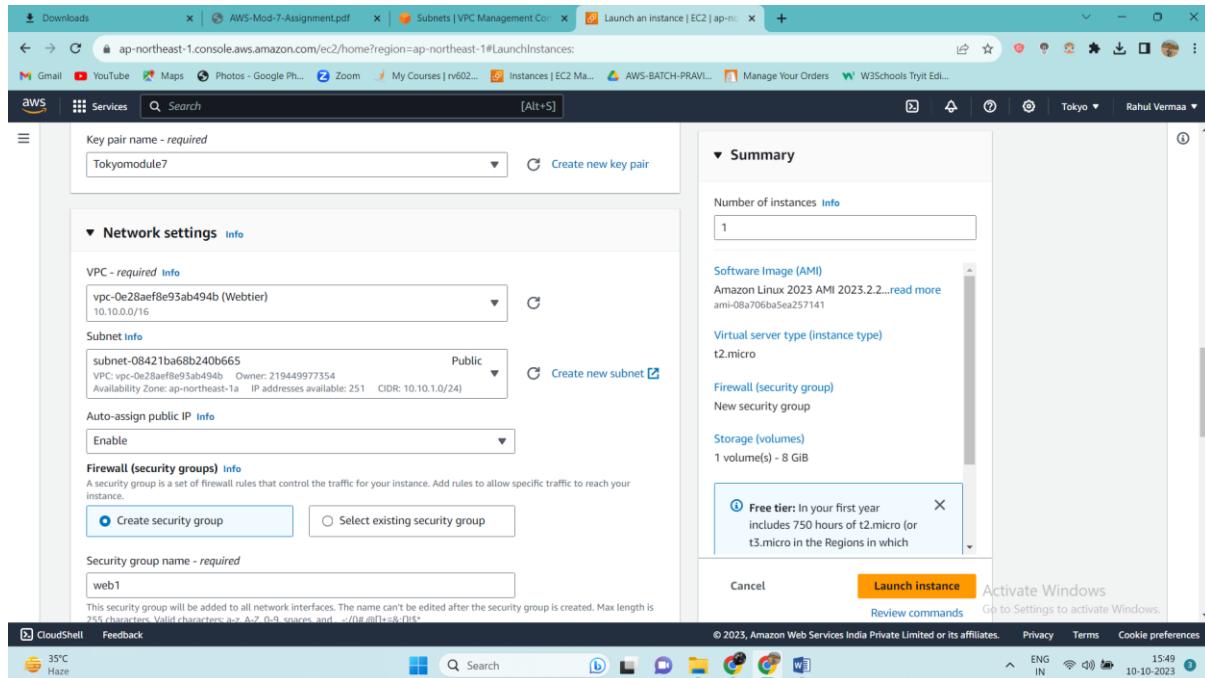
Go to public subnet settings and enable auto assign ip



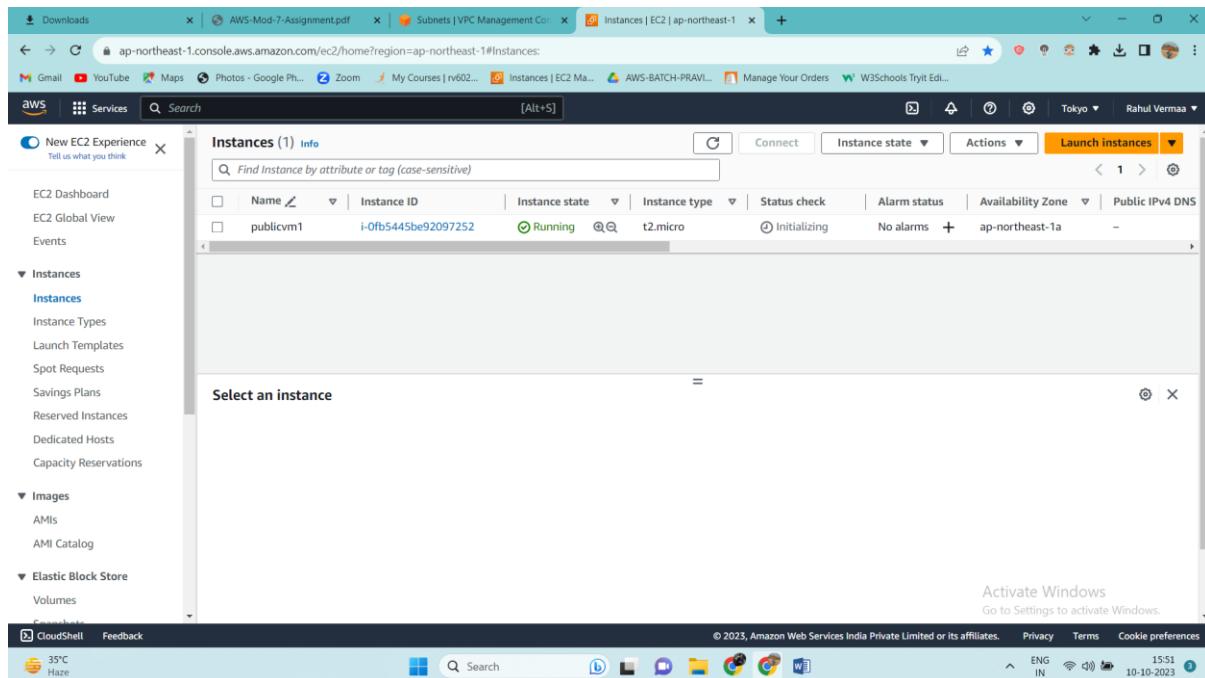
Now let's create 2 instance publicvm1 and privatevm1



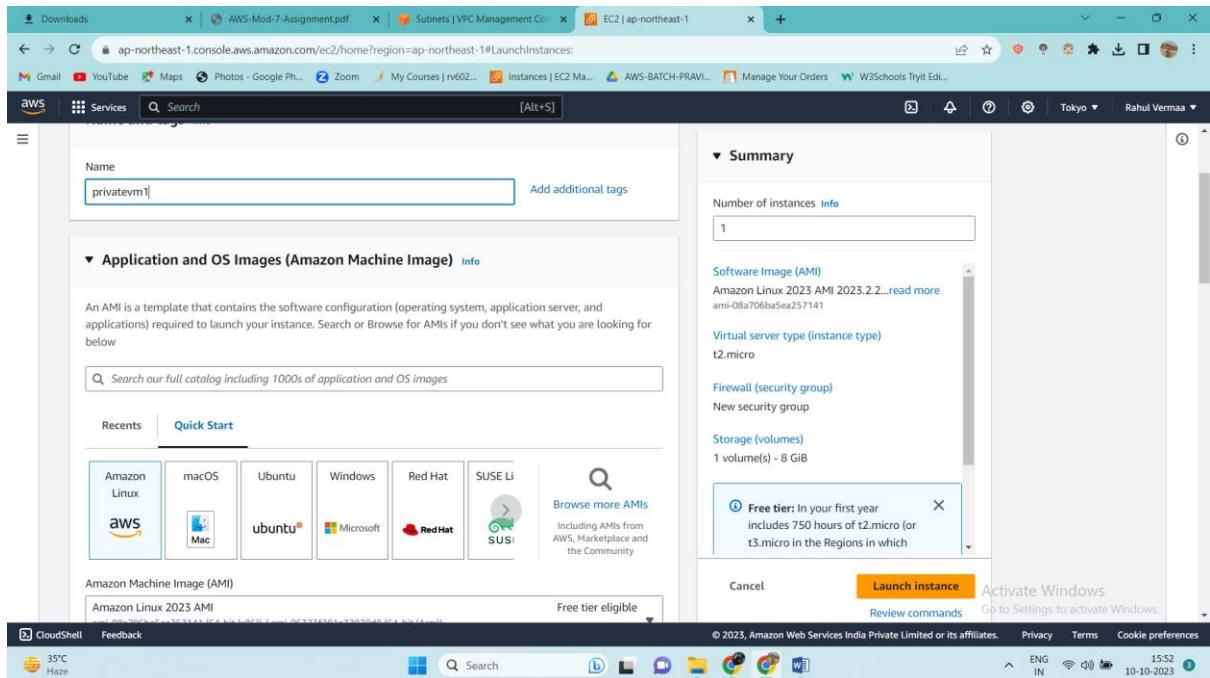
Select options like vpc, subnet etc...



Public instance is created

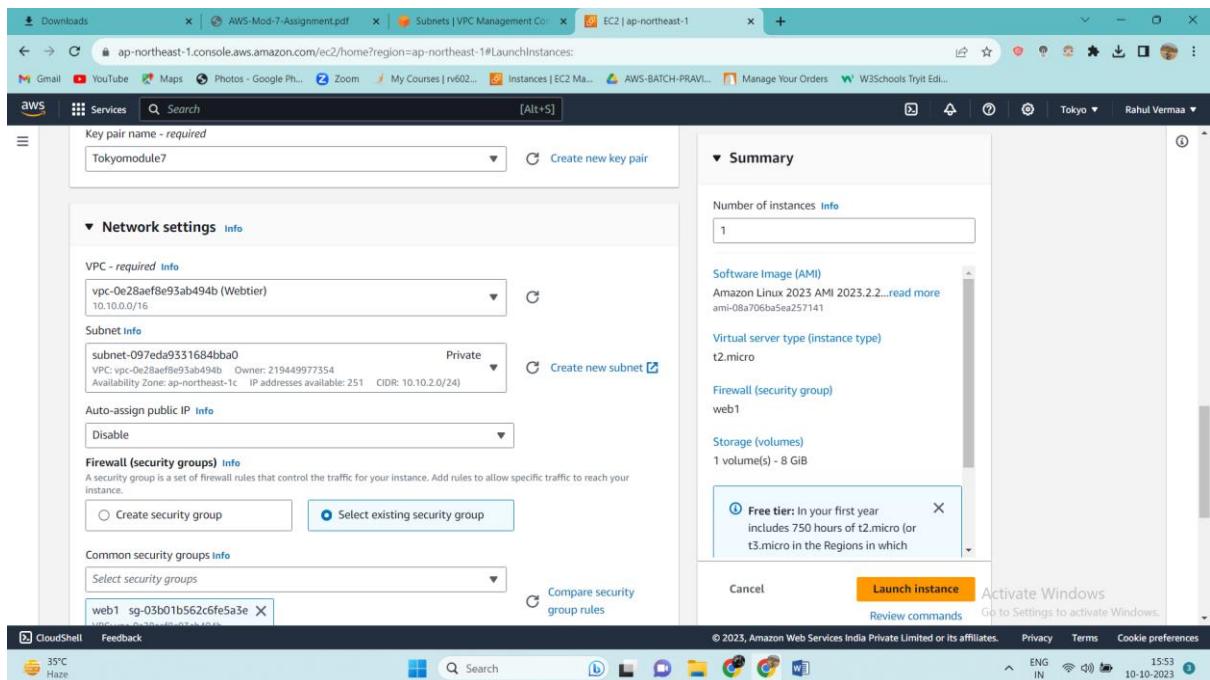


Similarly we will create one instance with private subnet



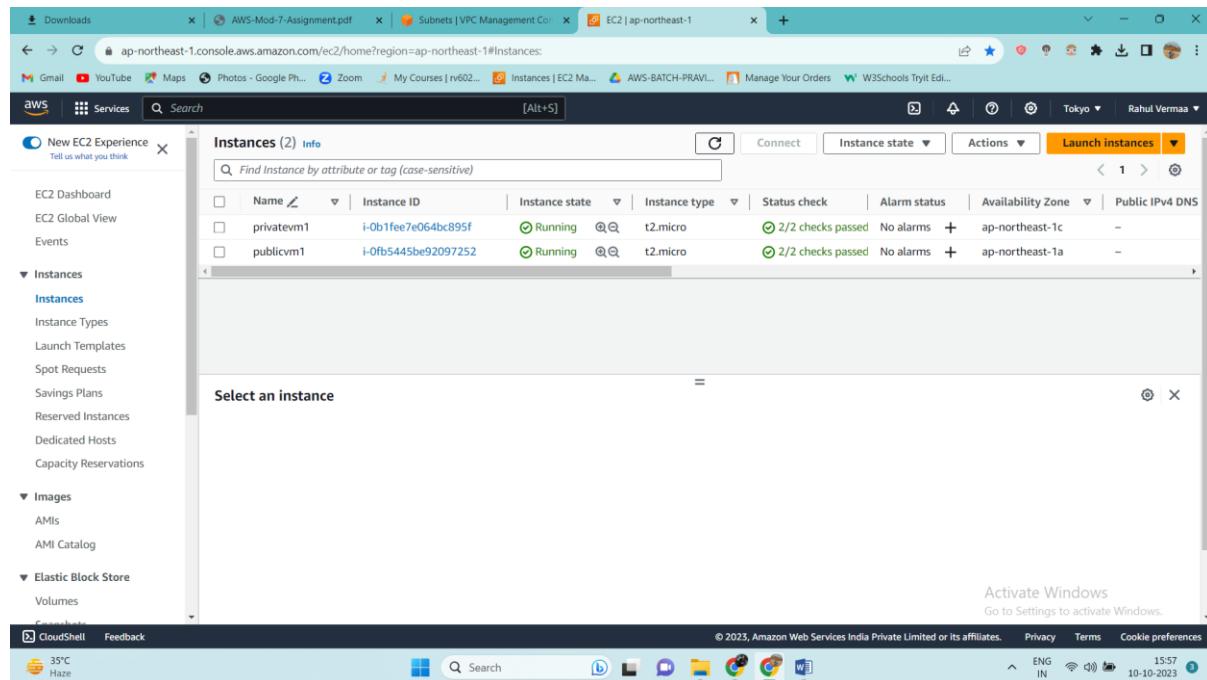
The screenshot shows the AWS EC2 Launch Instances page. The instance is named 'privatevm1'. The software image is 'Amazon Linux 2023 AMI 2023.2.2'. The virtual server type is 't2.micro'. A new security group is selected. Storage is 1 volume(s) - 8 GiB. A 'Free tier' modal is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch instances)'. The 'Launch instance' button is highlighted.

Select options like vpc, subnet, auto assign public ip- disabled etc...



The screenshot shows the AWS EC2 Launch Instances page. The instance is named 'Tokymodule7'. The VPC is 'vpc-0e28aef8e93ab494b (Webtier)'. The subnet is 'subnet-097eda9331684bba0'. The 'Auto-assign public IP' option is set to 'Disable'. The virtual server type is 't2.micro'. A 'Free tier' modal is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which you launch instances)'. The 'Launch instance' button is highlighted.

Both instance are running now

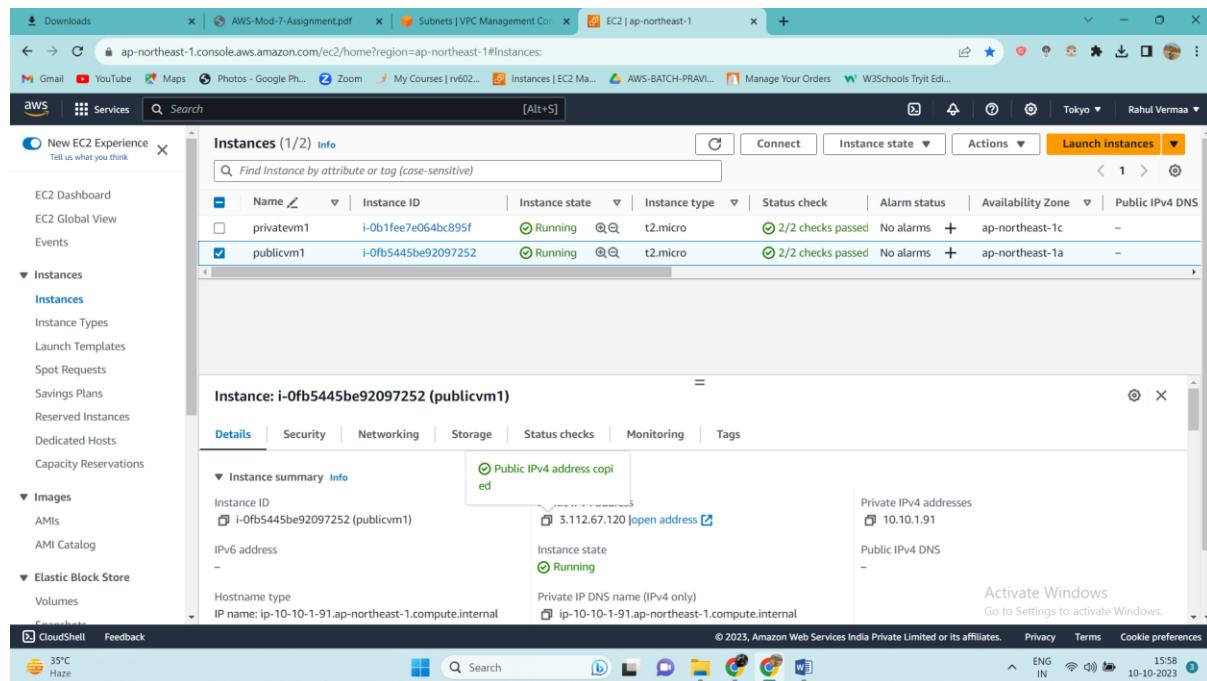


The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main table lists two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
privatevm1	i-0b1fee7e064bc895f	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-1c	-
publicvm1	i-0fb5445be92097252	Running	t2.micro	2/2 checks passed	No alarms	ap-northeast-1a	-

Below the table, a modal window titled "Select an instance" is open, showing the same two instances. The status bar at the bottom right shows "15:57 10-10-2023".

Now let's connect to our instance and install some http file to display some data first will connect to publicvm1

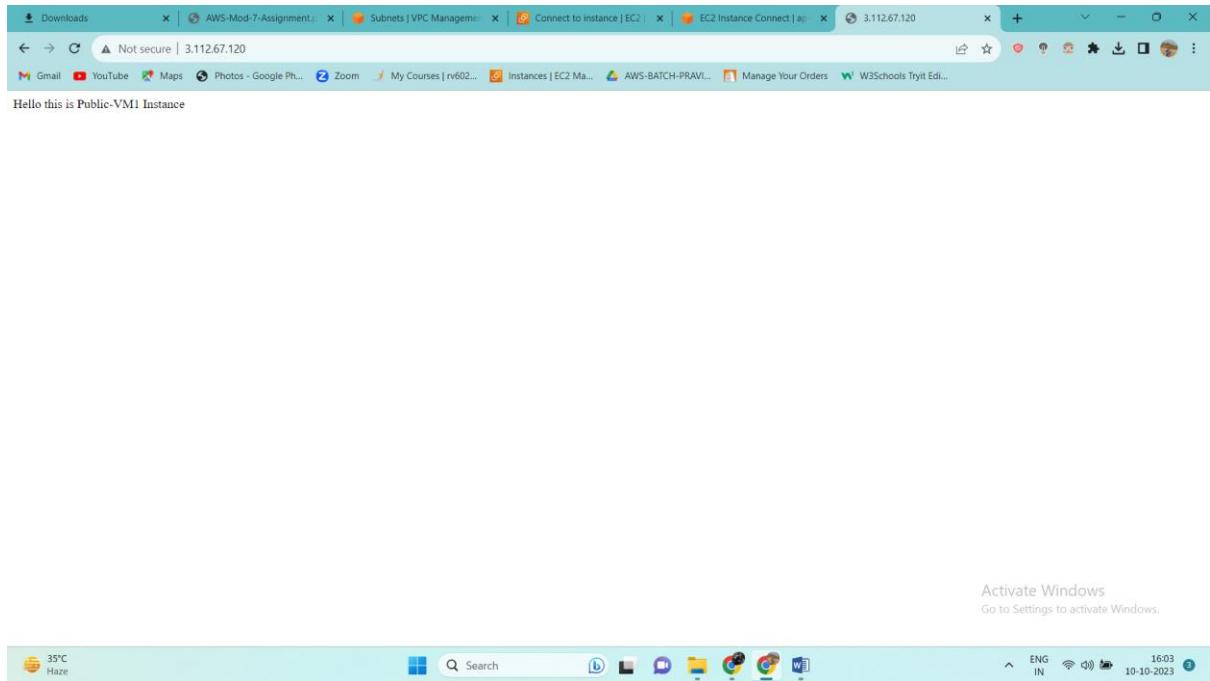


The screenshot shows the AWS EC2 Instances page with the "publicvm1" instance selected. The main table shows the instance is running. The "Details" tab is selected in the instance details panel, which displays the following information:

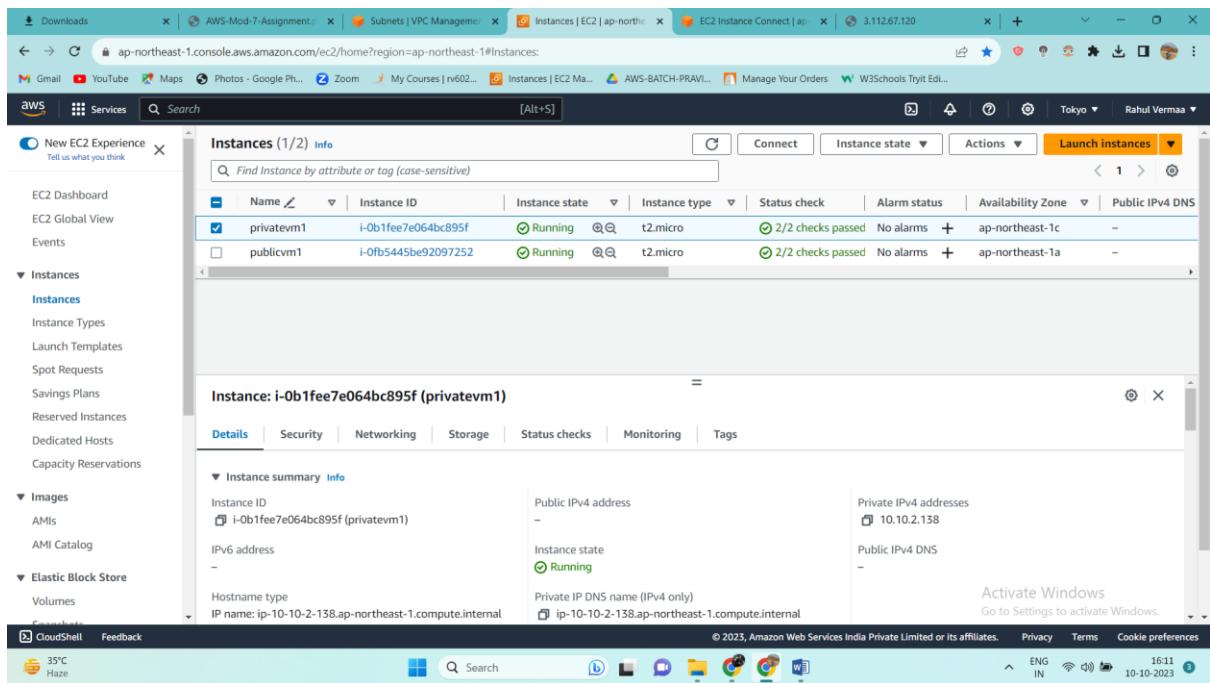
Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance ID: i-0fb5445be92097252 (publicvm1)						
Public IPv4 address copied						
3.112.67.120 [open address]						
Instance ID: i-0fb5445be92097252 (publicvm1)						
IPv6 address: -						
Instance state: Running						
Hostname type: IP name: ip-10-10-1-91.ap-northeast-1.compute.internal						
Private IP DNS name (IPv4 only): ip-10-10-1-91.ap-northeast-1.compute.internal						

The status bar at the bottom right shows "15:58 10-10-2023".

Html file installed

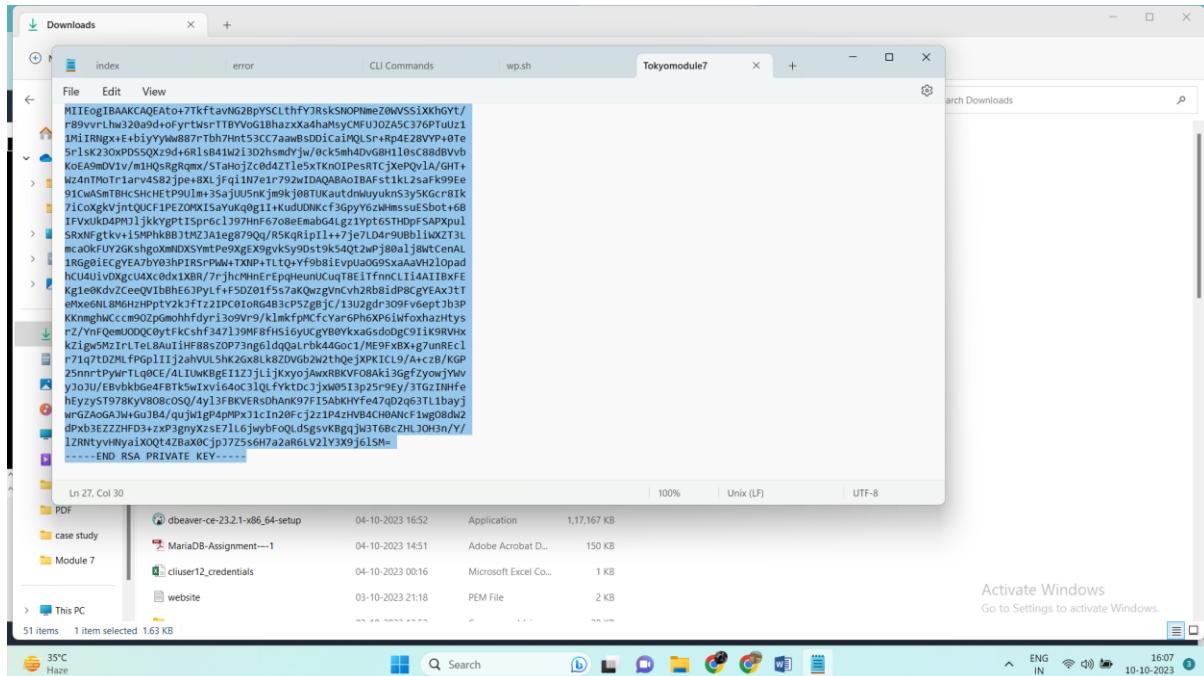


Now let's try to connect with our privatevm1 instance where there is no ipv4 enabled

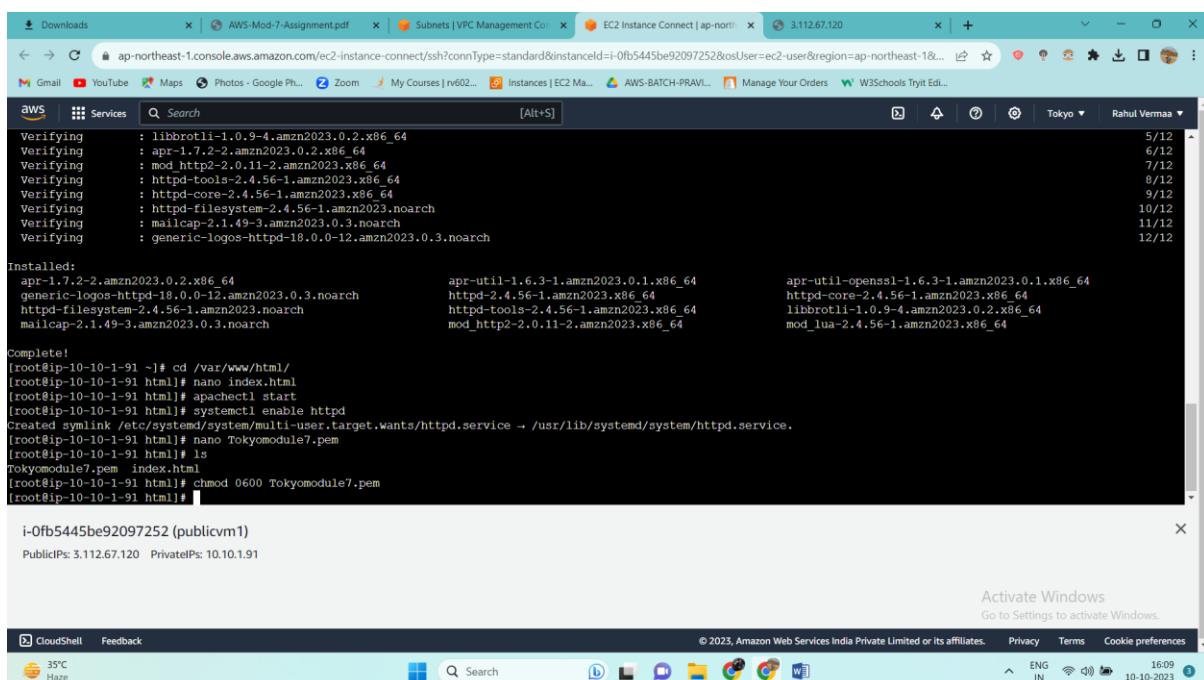


For this we have to create a path in our publicvm1

Create a file Tokyomodule7.pem and paste the key content in it



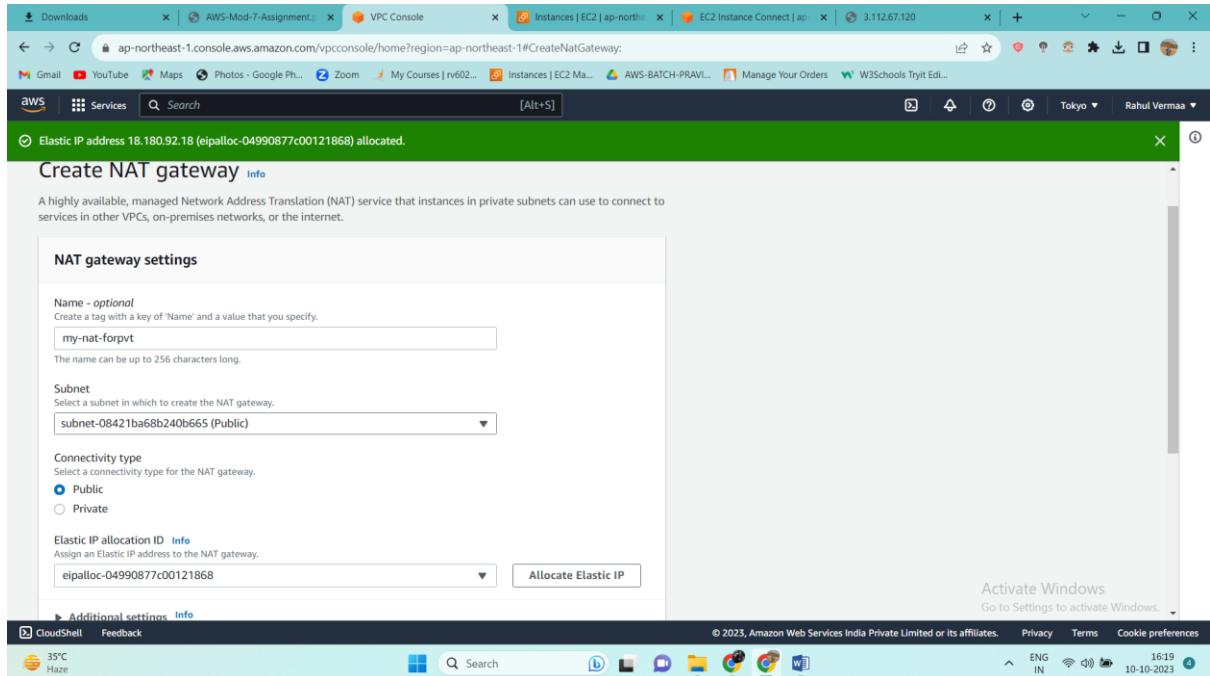
Allow that Tokyomodule7.pem file to read the content



Now ssh from publicvm1 to connect privatevm1 and we are successfully connected to privatevm1

Internet is not there to give internet accces to our privatevm1 we will install one Nat gateway

Let's create one NAT gateway (we are allocating elastic ip)



Elastic IP address 18.180.92.18 (eipalloc-04990877c00121868) allocated.

Create NAT gateway [Info](#)

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

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.

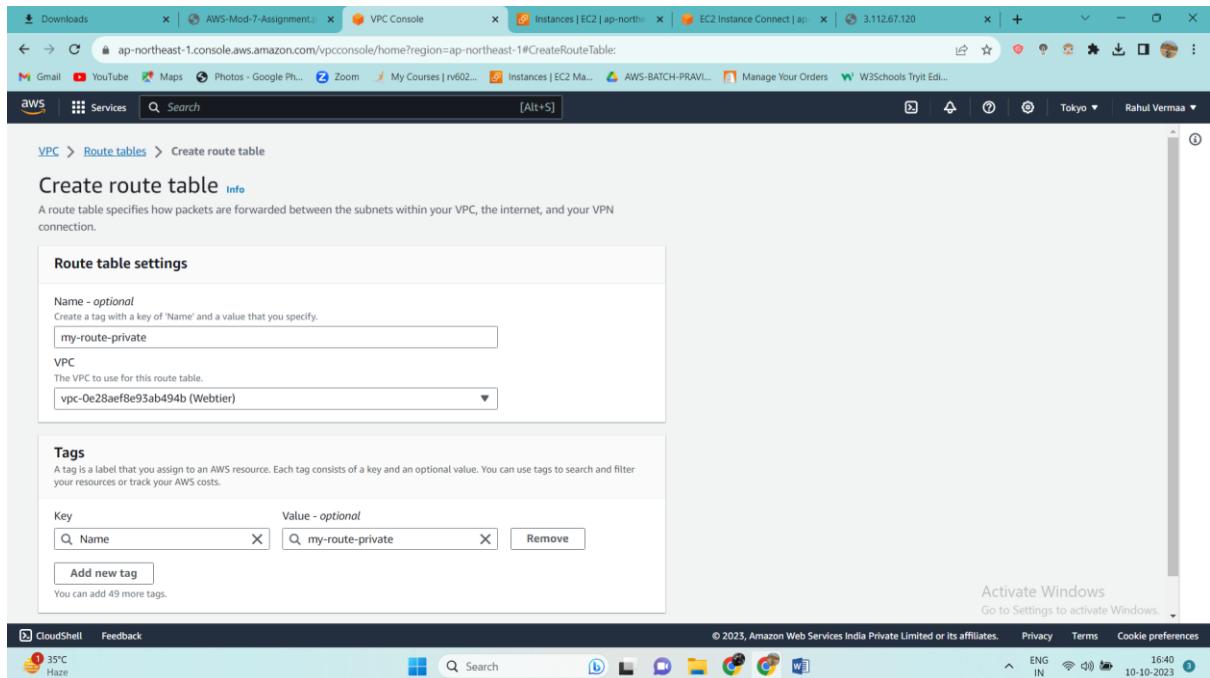
Connectivity type
Select a connectivity type for the NAT gateway.
 Public
 Private

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

[Additional settings](#) [Info](#)

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Haze 16:19 10-10-2023

Now create a route table and route the traffic to Nat-gateway



VPC > Route tables > Create route table

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.

VPC
The VPC to use for this route table.

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.

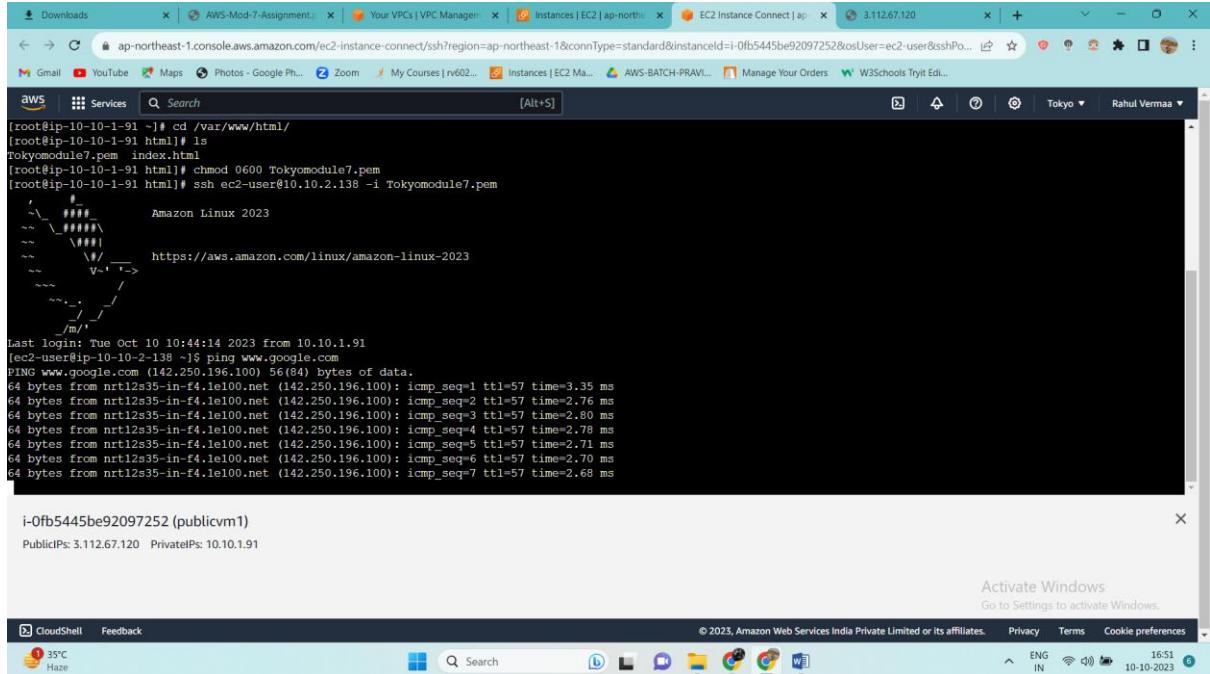
Key Value - *optional*

You can add 49 more tags.

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 35°C Haze 16:40 10-10-2023

Now we are able to ping www.google.com that means internet is connected to our privatevm1

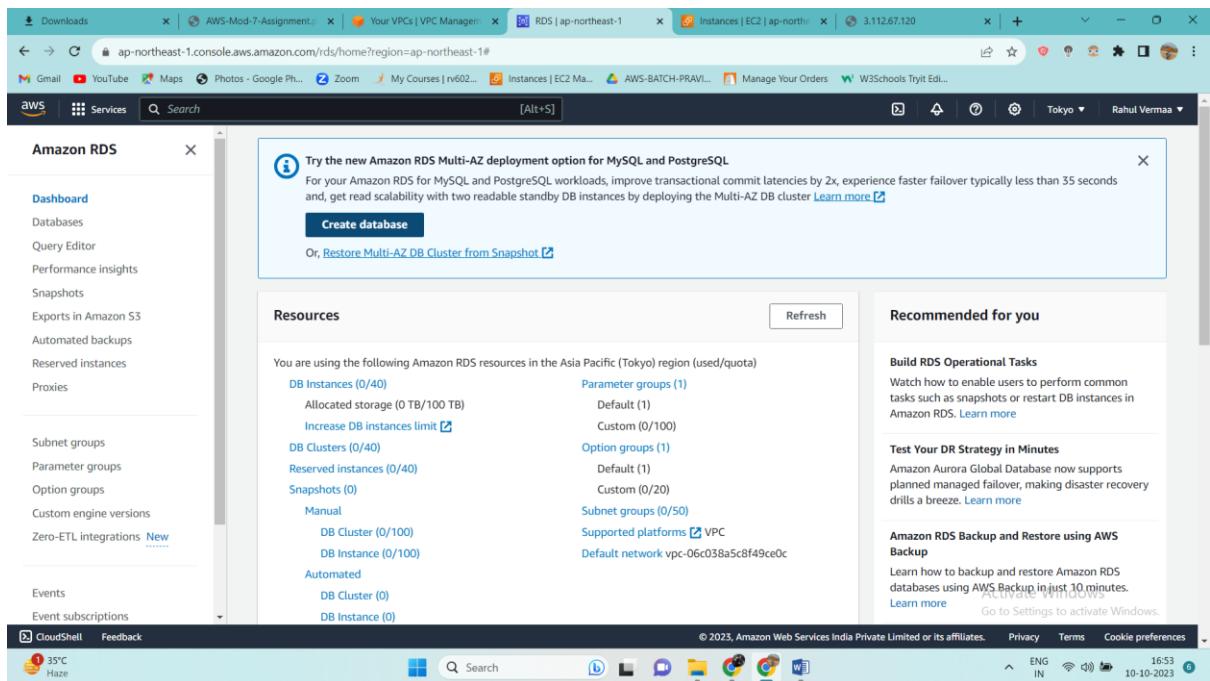


```
[root@ip-10-10-1-91 ~]# cd /var/www/html/
[root@ip-10-10-1-91 html]# ls
tokyomodule7.pem index.html
[root@ip-10-10-1-91 html]# chmod 0600 Tokyomodule7.pem
[root@ip-10-10-1-91 html]# ssh ec2-user@10.10.2.138 -i Tokyomodule7.pem
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Tue Oct 10 10:44:14 2023 from 10.10.1.91
[ec2-user@ip-10-10-2-138 ~]$ ping www.google.com
PING www.google.com (142.250.196.100) 56(84) bytes of data.
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=1 ttl=57 time=3.35 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=2 ttl=57 time=2.76 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=3 ttl=57 time=2.80 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=4 ttl=57 time=2.78 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=5 ttl=57 time=2.71 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=6 ttl=57 time=2.70 ms
64 bytes from rtr12s35-in-f4.le100.net (142.250.196.100): icmp seq=7 ttl=57 time=2.68 ms

i-Ofb5445be92097252 (publicvm1)
PublicIPs: 3.112.67.120 PrivateIPs: 10.10.1.91
```

Now let's create one RDS database



Amazon RDS

Dashboard

Databases

Query Editor

Performance insights

Snapshots

Exports in Amazon S3

Automated backups

Reserved instances

Proxies

Subnet groups

Parameter groups

Option groups

Custom engine versions

Zero-ETL integrations

Events

Event subscriptions

Resources

Refresh

You are using the following Amazon RDS resources in the Asia Pacific (Tokyo) region (used/quota)

DB Instances (0/40)	Parameter groups (1)
Allocated storage (0 TB/100 TB)	Default (1)
Increase DB instances limit	Custom (0/100)

DB Clusters (0/40)	Option groups (1)
Reserved instances (0/40)	Default (1)
Snapshots (0)	Custom (0/20)
Manual	Subnet groups (0/50)

DB Cluster (0/100)	Supported platform
DB Instance (0/100)	Default network vpc-06c038a5c8f49ce0c

Automated

DB Cluster (0)

DB Instance (0)

Recommended for you

Build RDS Operational Tasks

Watch how to enable users to perform common tasks such as snapshots or restart DB instances in Amazon RDS. [Learn more](#)

Test Your DR Strategy in Minutes

Amazon Aurora Global Database now supports planned managed failover, making disaster recovery drills a breeze. [Learn more](#)

Amazon RDS Backup and Restore using AWS Backup

Learn how to backup and restore Amazon RDS databases using AWS Backup in just 10 minutes. [Learn more](#)

Activate Windows

CloudShell Feedback

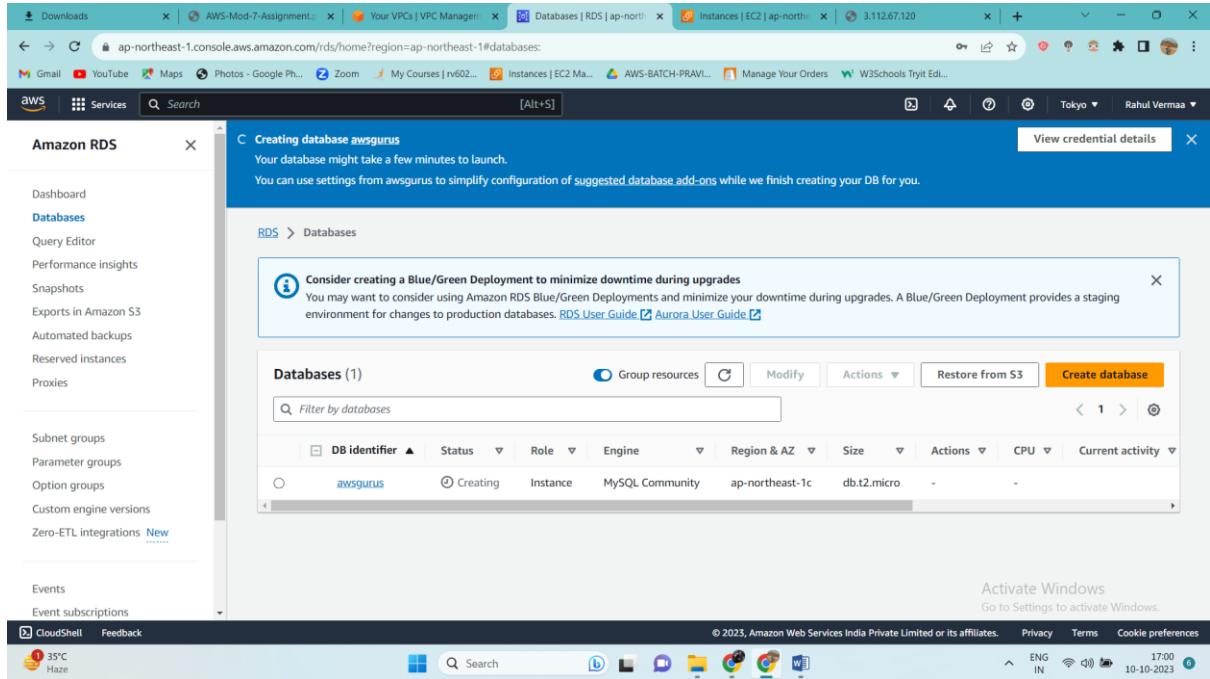
CloudShell Feedback

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

35°C Haze

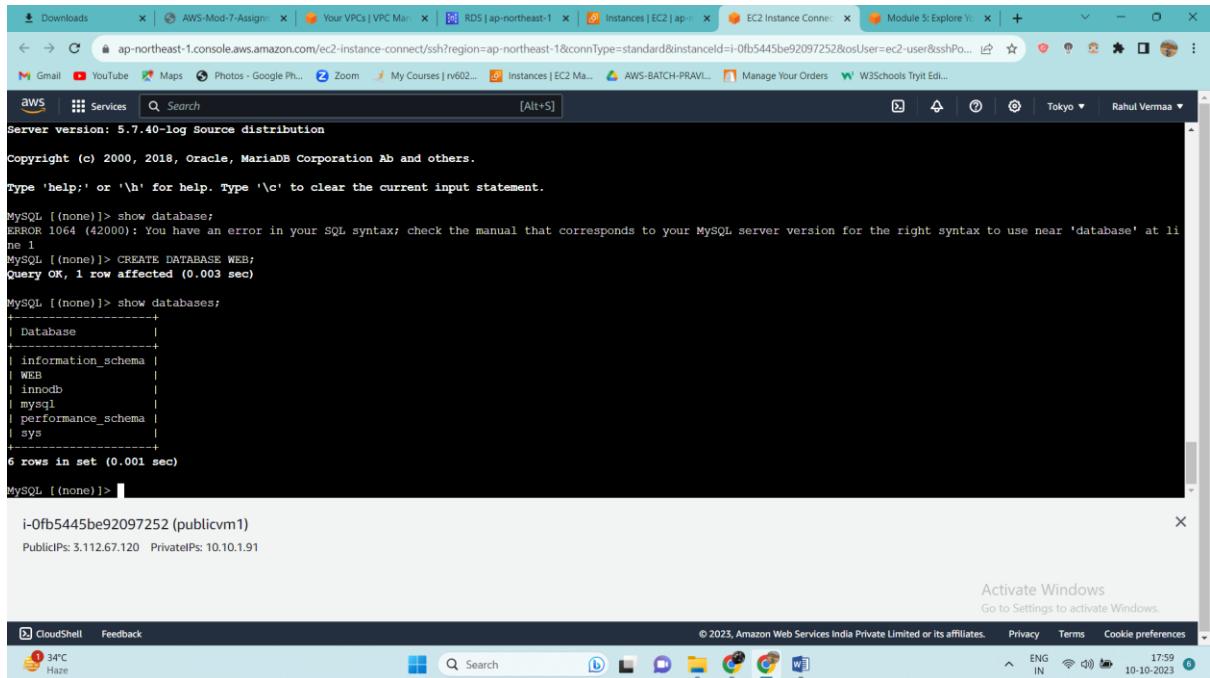
16:53 IN 10-10-2023

Our RDS is created in subnet private



The screenshot shows the AWS RDS console with a database named 'awsgurus' in the 'Creating' status. The database is a MySQL Community instance with the engine type 'MySQL Community', region 'ap-northeast-1c', and size 'db.t2.micro'. The 'Create database' button is visible at the top right of the table.

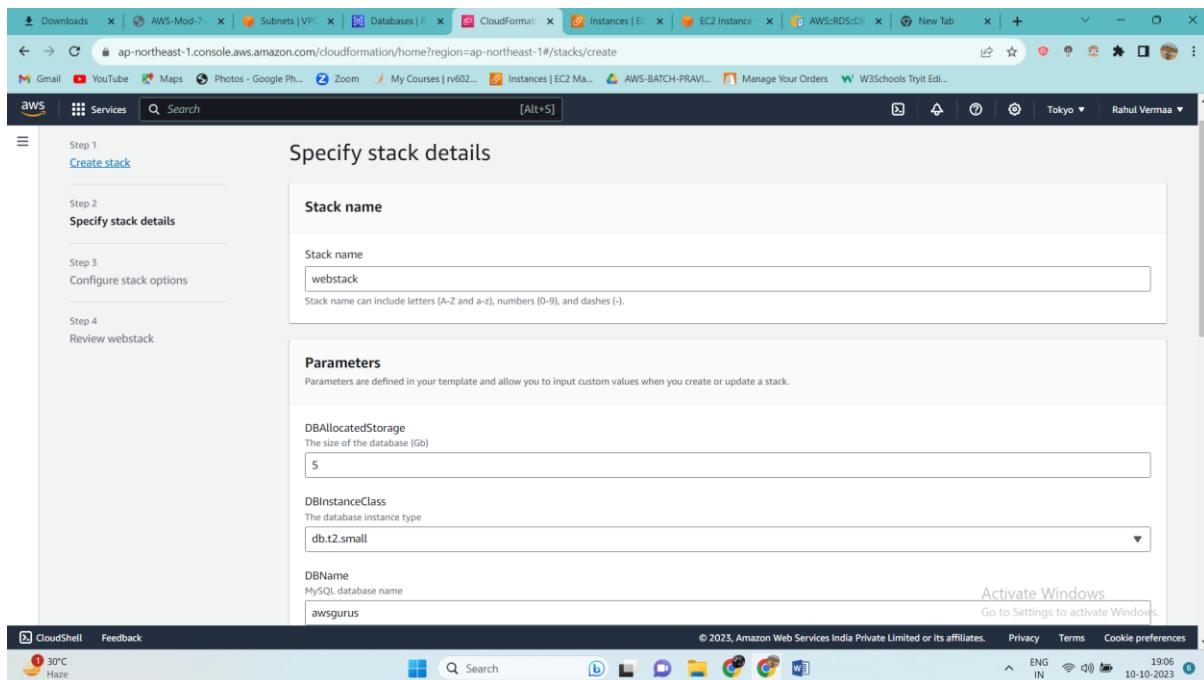
And we are connected to our RDS



The screenshot shows a CloudShell terminal with a MySQL session connected to the 'awsgurus' database. The session shows the creation of a database named 'WEB' and a successful 'CREATE DATABASE' command. The terminal also displays the MySQL prompt and the results of a 'show databases' command, which lists the databases: 'information_schema', 'WEB', 'innodb', 'mysql', 'performance_schema', and 'sys'. The session ends with a 'Query OK, 1 row affected (0.003 sec)' message and a '6 rows in set (0.001 sec)' message.

Now let's create one stack to fulfil this statement :

The development team can test the code without having to involve the System Admins and can invest their time in testing the code rather than provisioning, configuring, and updating the resources needed to test the code



Our stack is created

