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

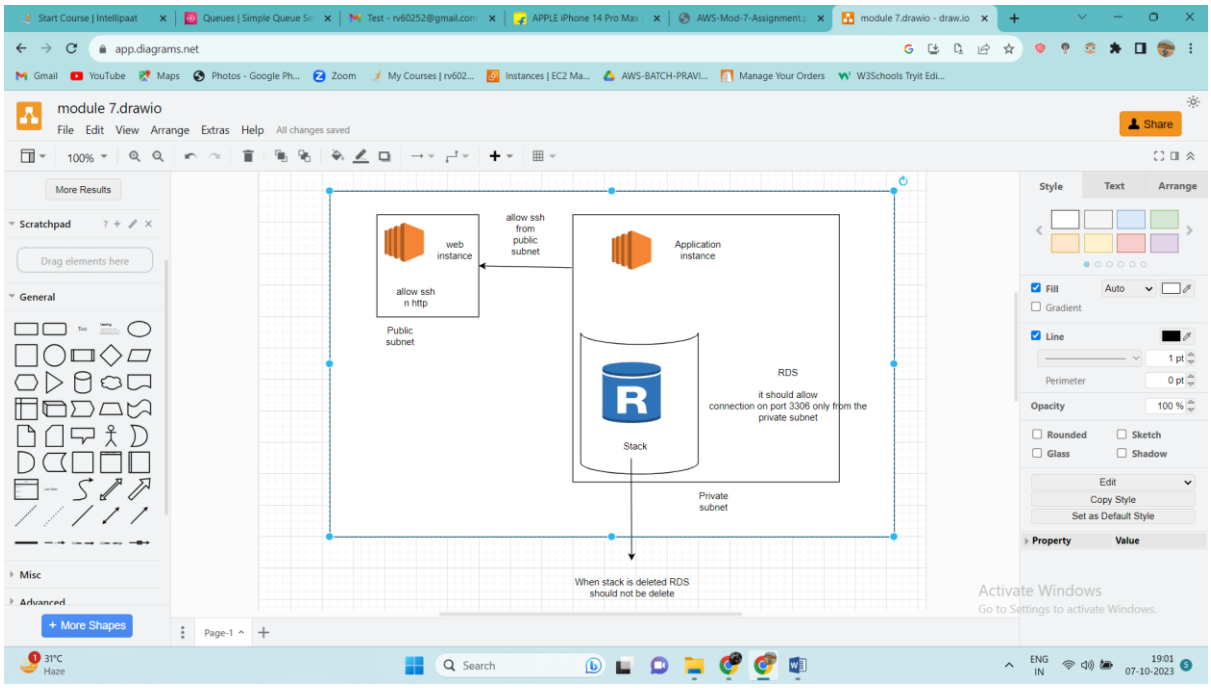
4. Setup a Route 53 hosted zone, and direct the traffic to the EC2 instance

You are also asked to propose a solution so that:

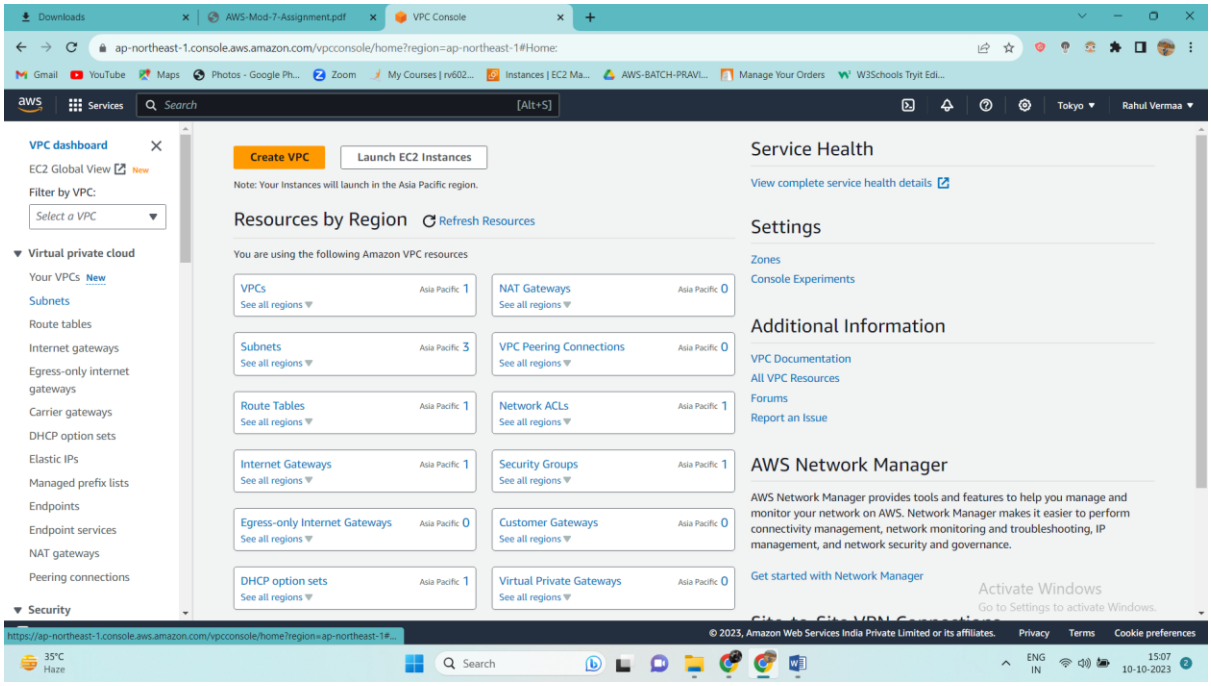
1. The development team can test the code without having to involve the System Admins and can invest their time in testing the code rather than provisioning, configuring, and updating the resources needed to test the code

2. When the development team deletes the stack,the RDS DB instance should not be deleted.

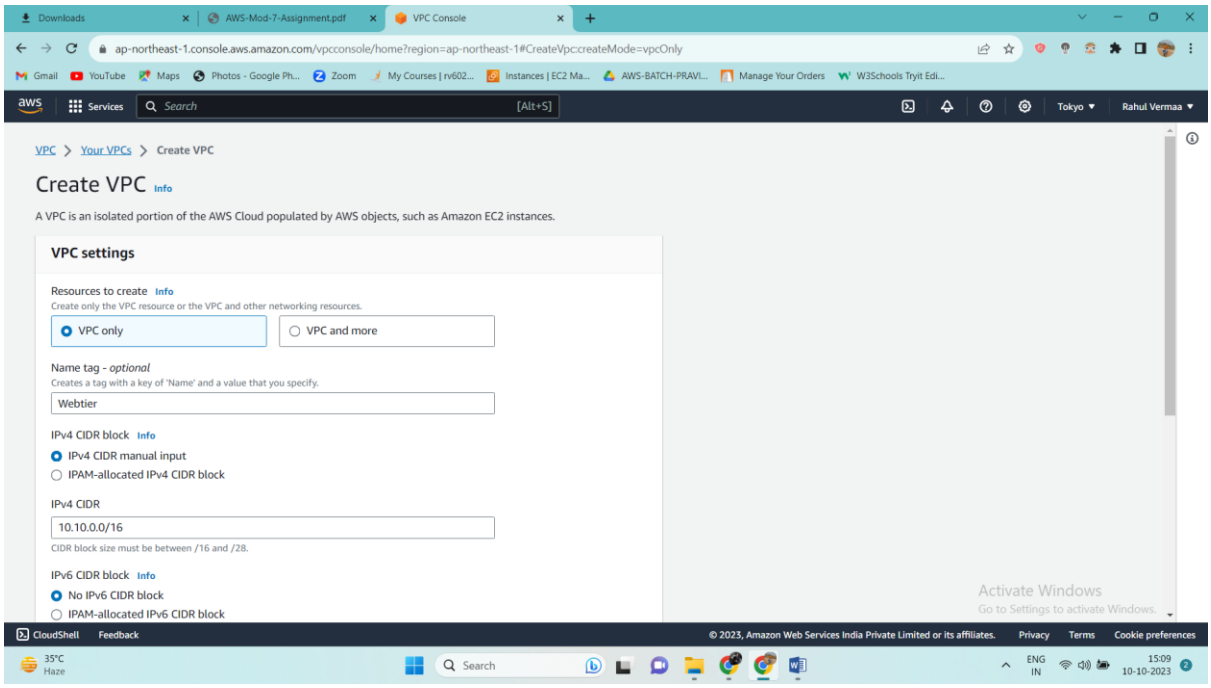
So this how our entire case study looks like



First we will create one VPC

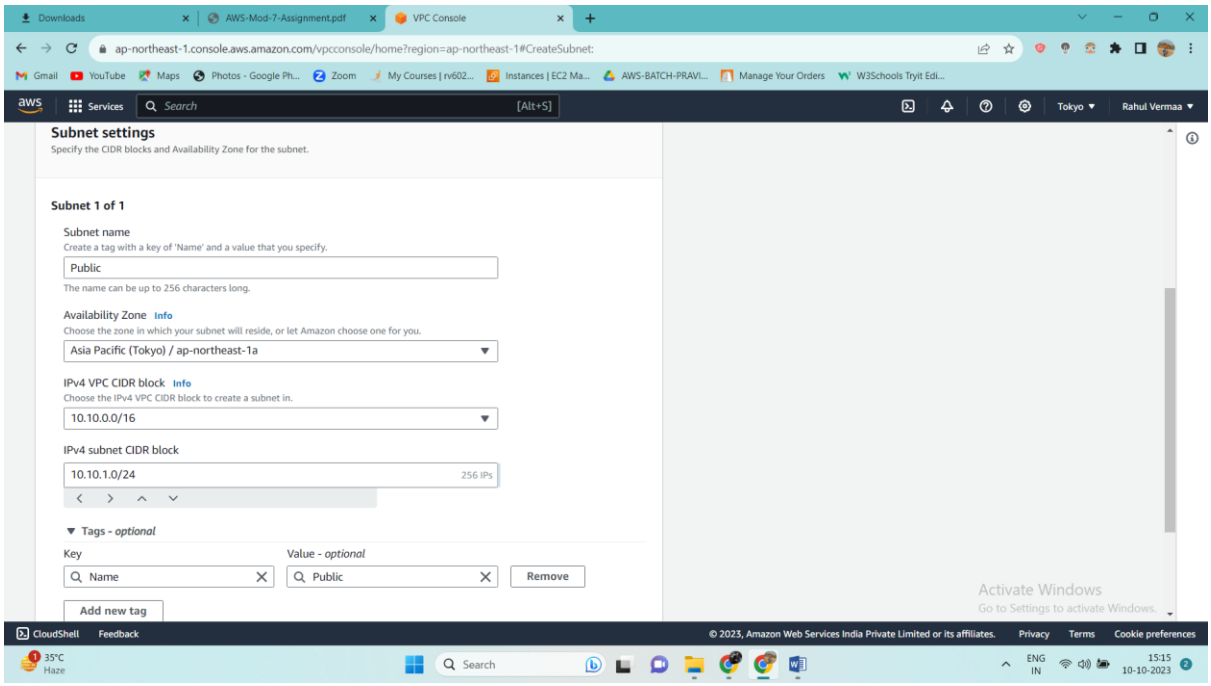


Select options and click it on create VPC button

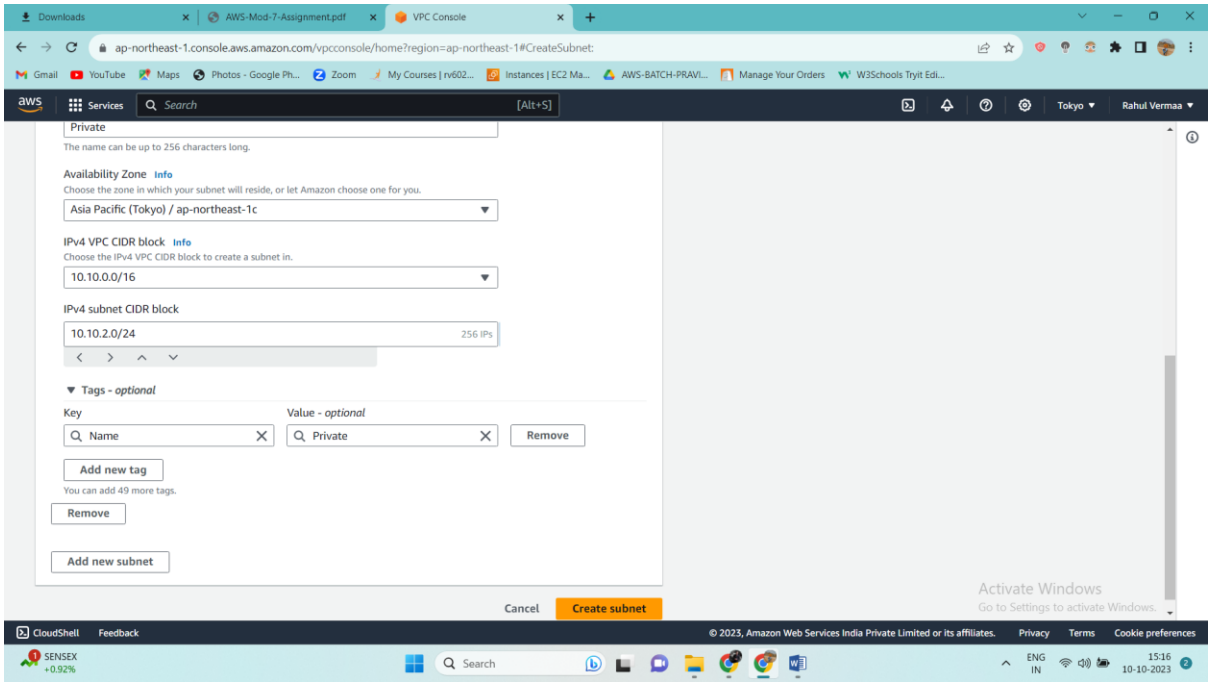


Now we will create 2 subnets public and private

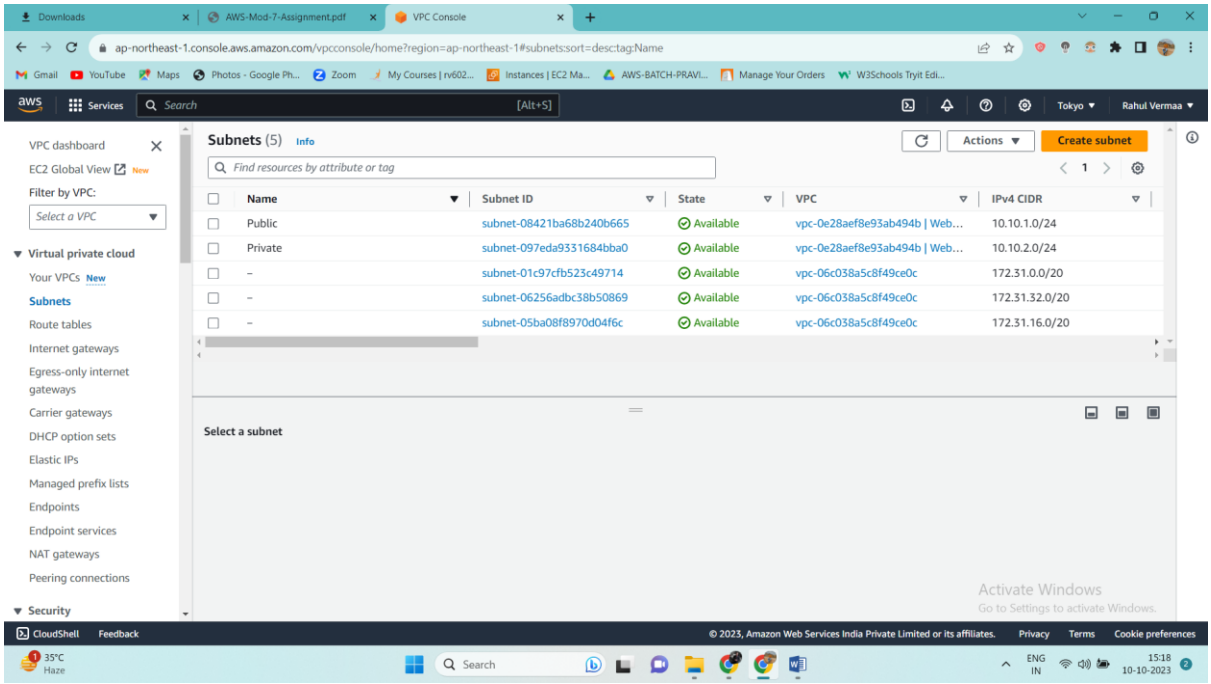
Select options and create your public subnet



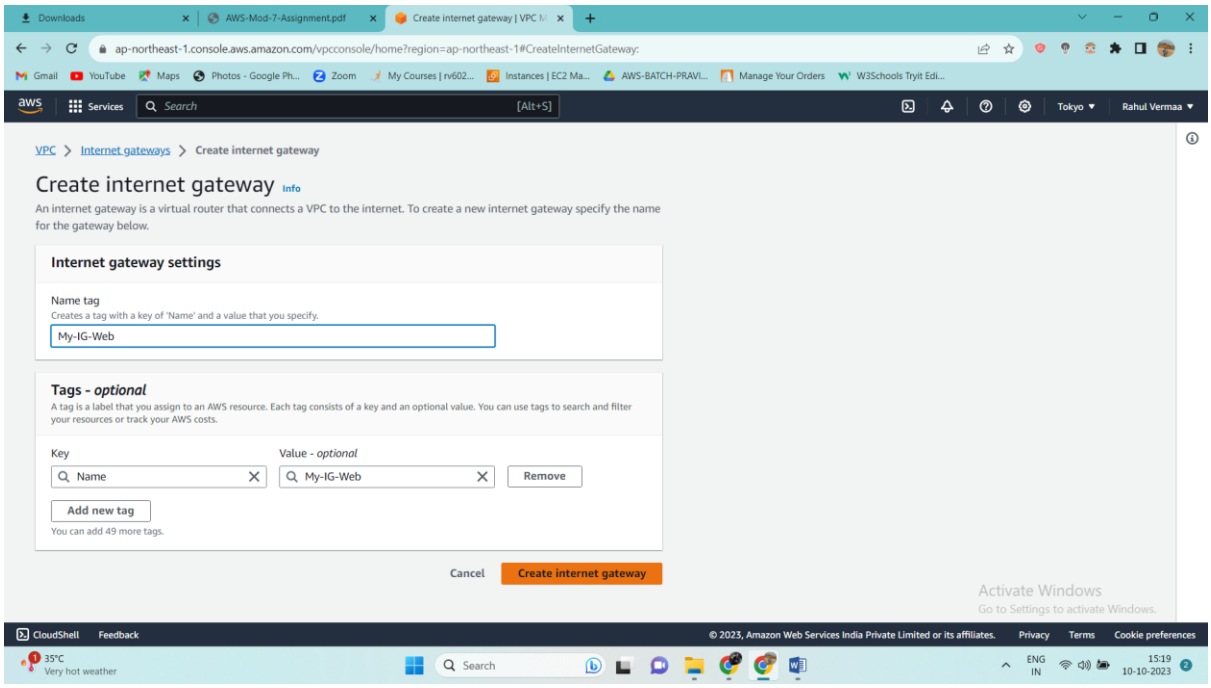
Now create second subnet you can click on add subnet button and create multiple subnets at once. But to take screen shots I have create them separately.



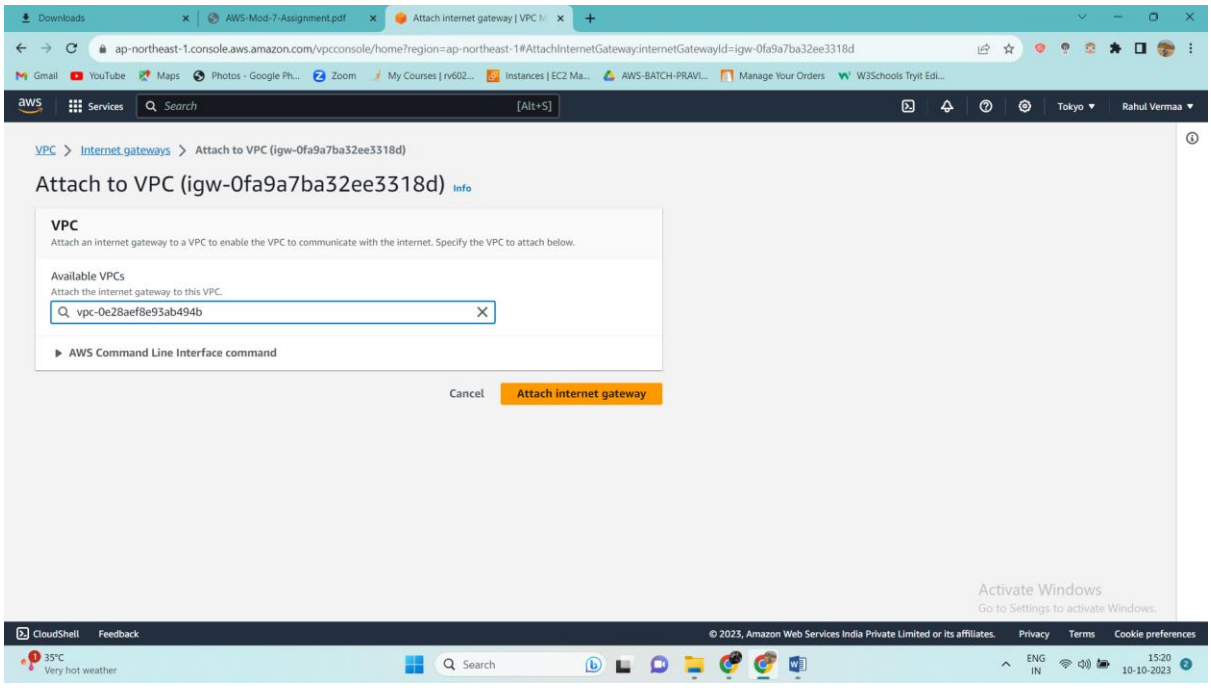
Both are created now



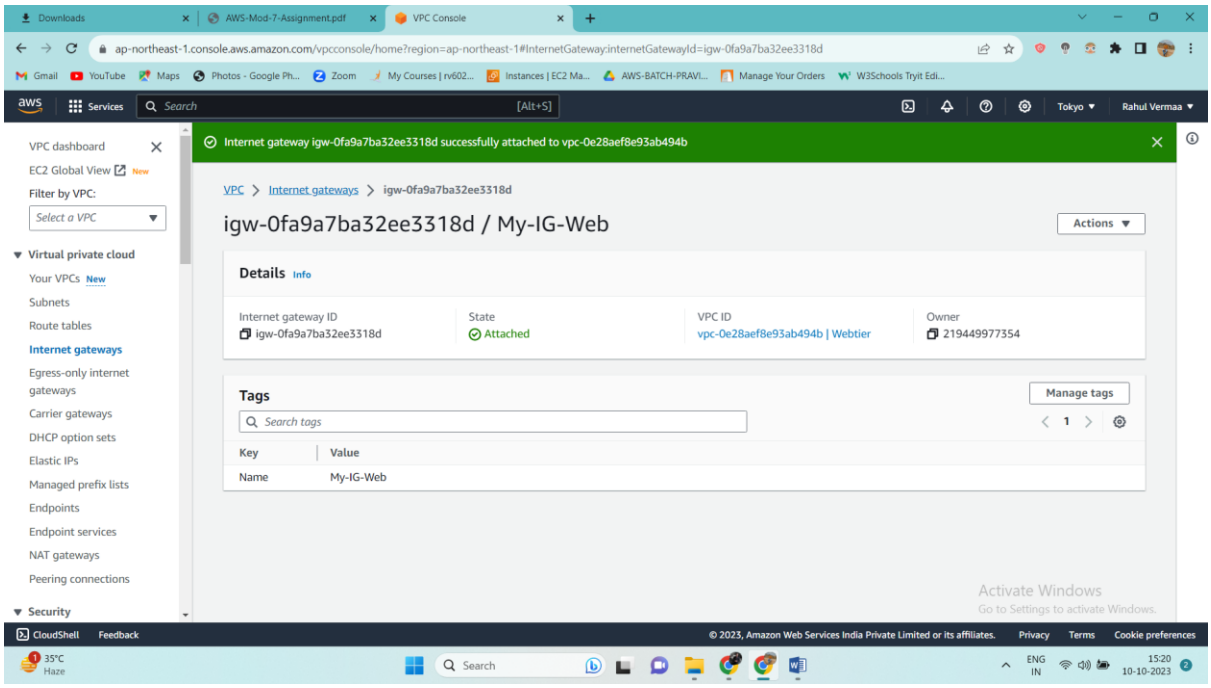
Now we need to install one internet gateway



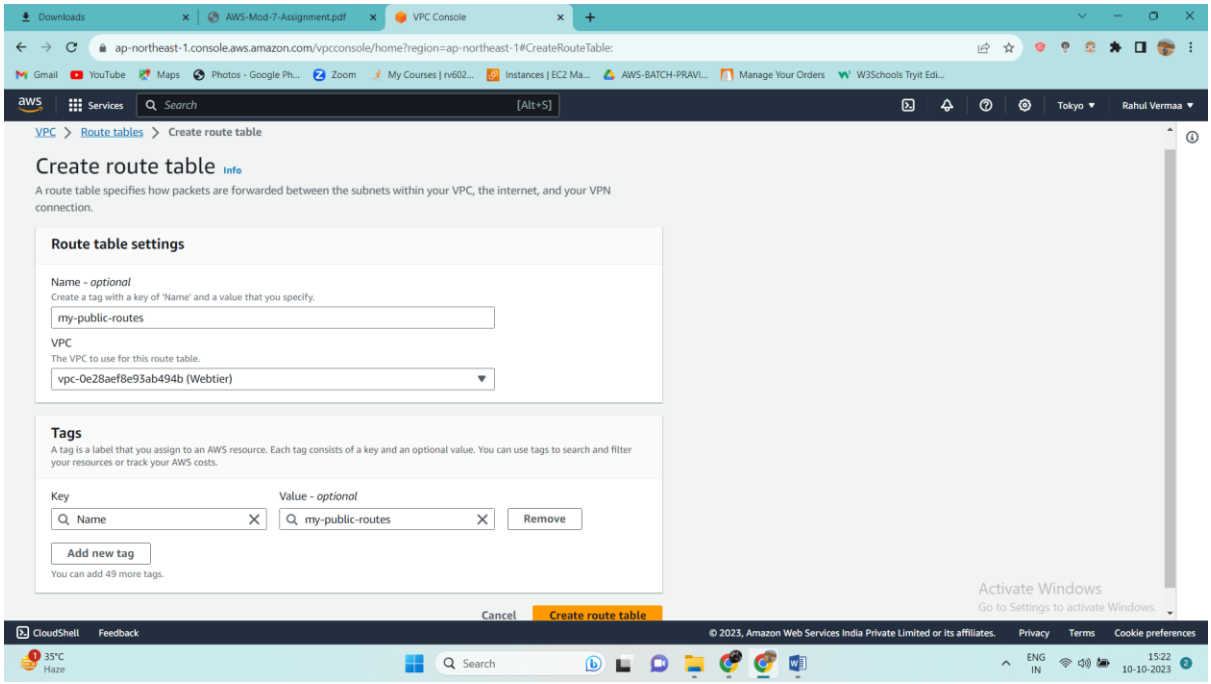
Created successfully and now attach it to our VPC



It’s attached to our VPC successfully

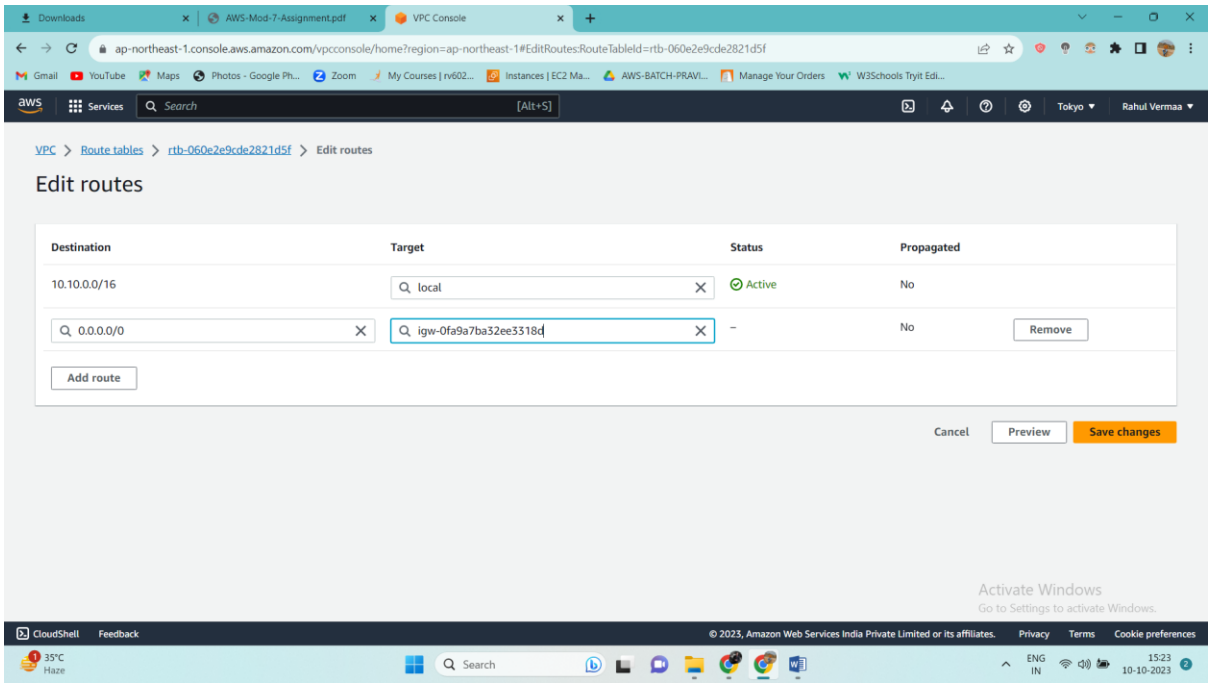


Now we will create one route table so that only public subnet is able to get the internet connection

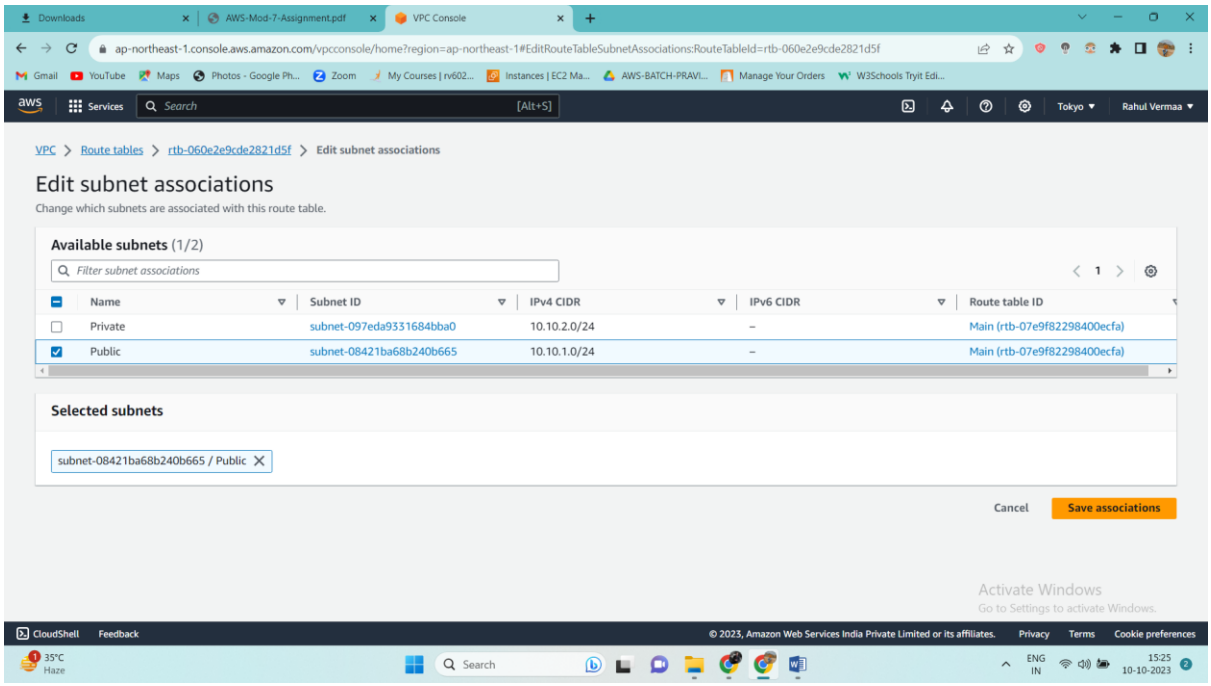


Now edit routes in route table and save changes

Route it to internet gateway

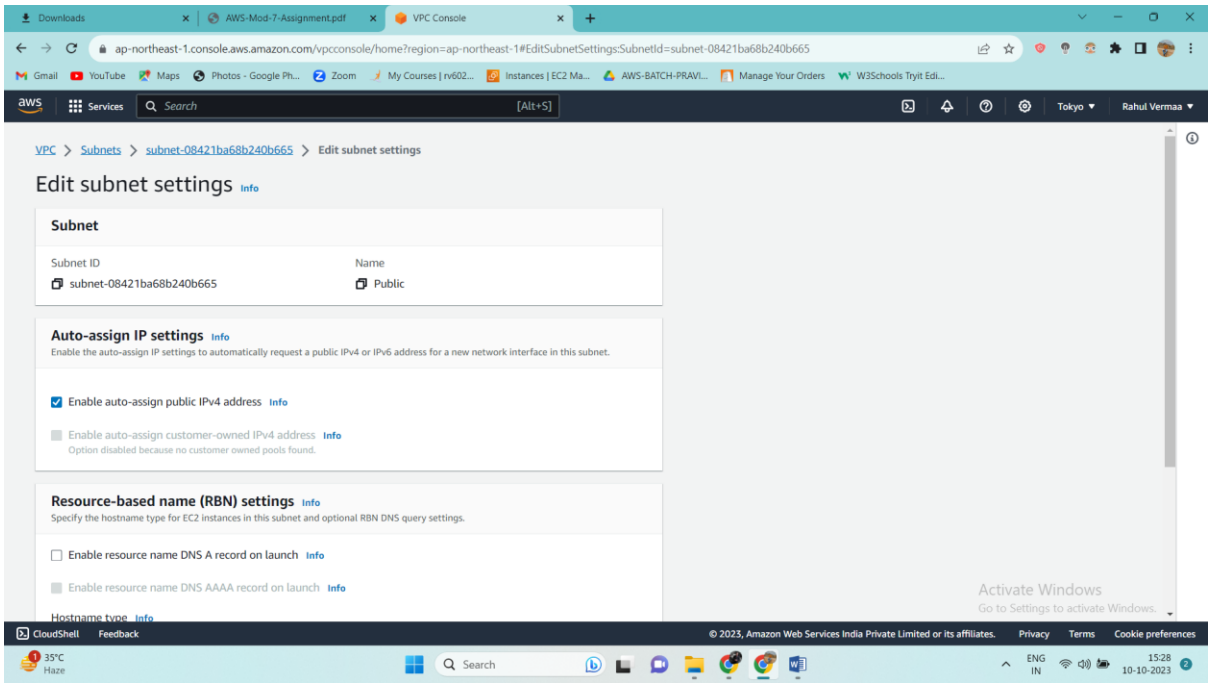


Now attach public subnet to our route table

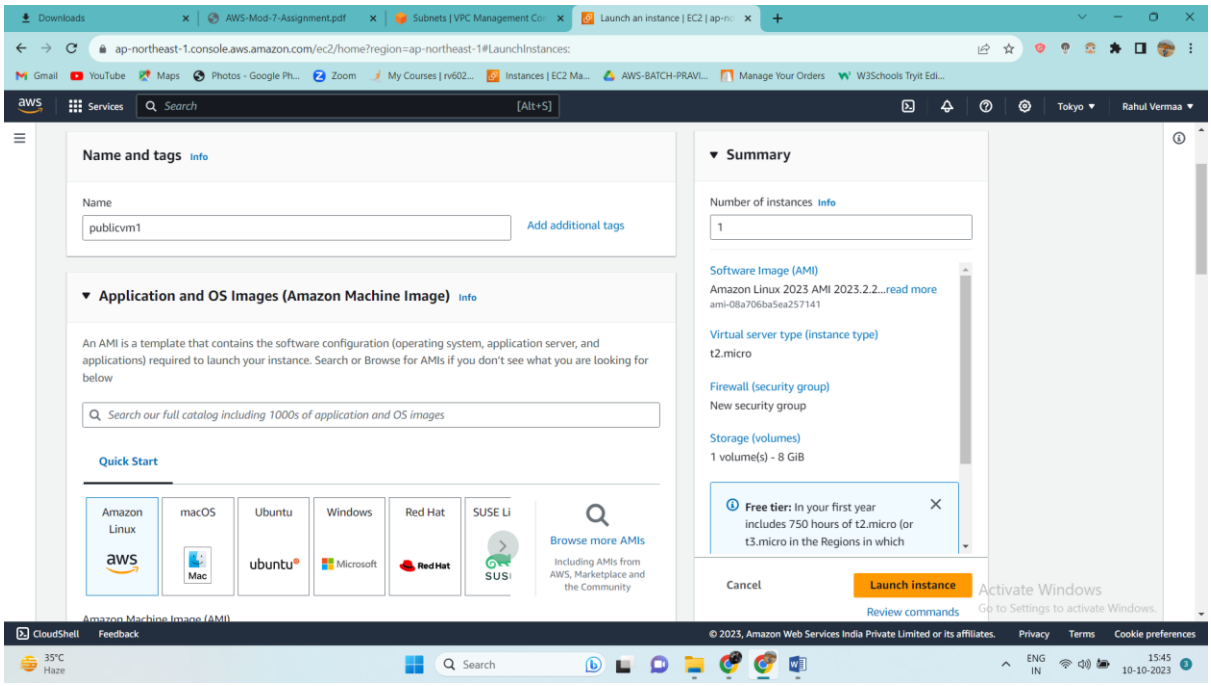


Now we have to enable ipv4 config in public subnet

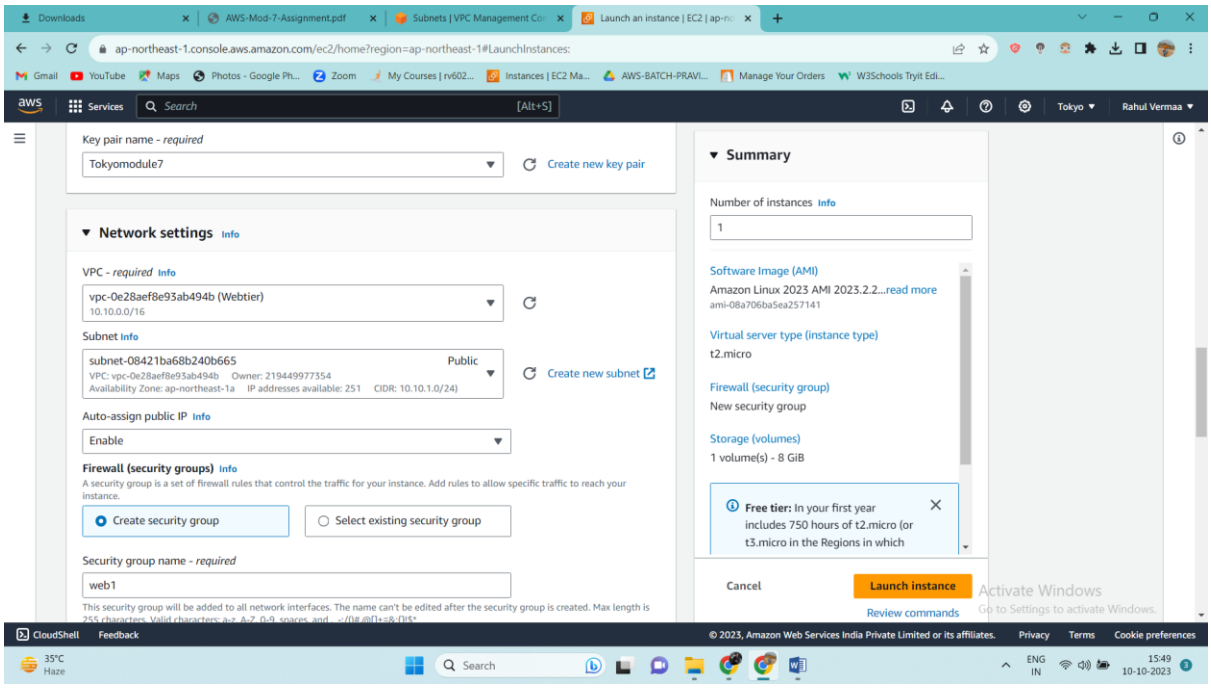
Go to public subnet settings and enable auto assign ip



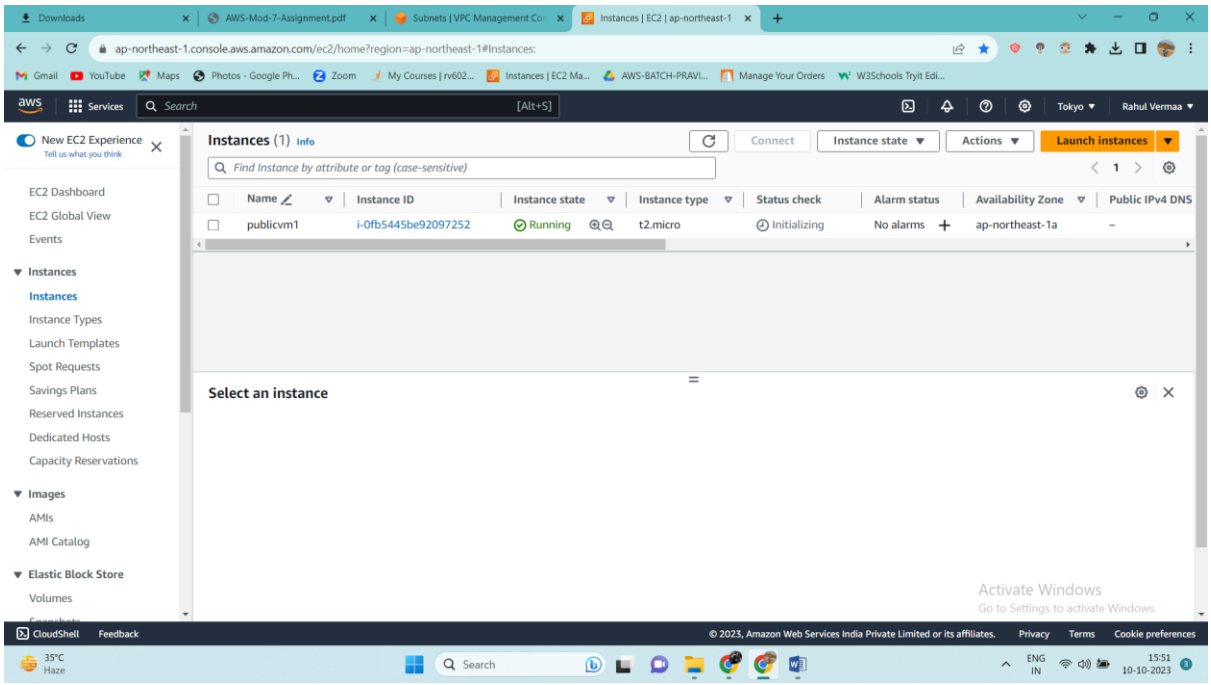
Now let’s create 2 instance publicvm1 and privatevm1



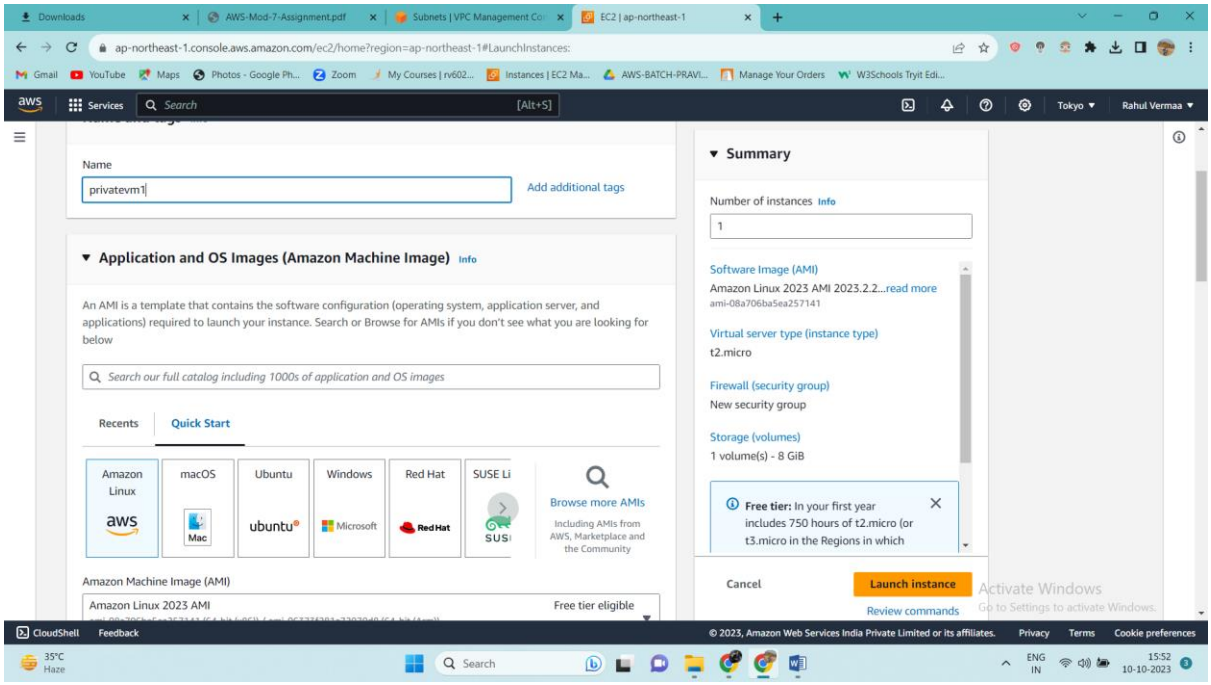
Select options like vpc, subnet etc...



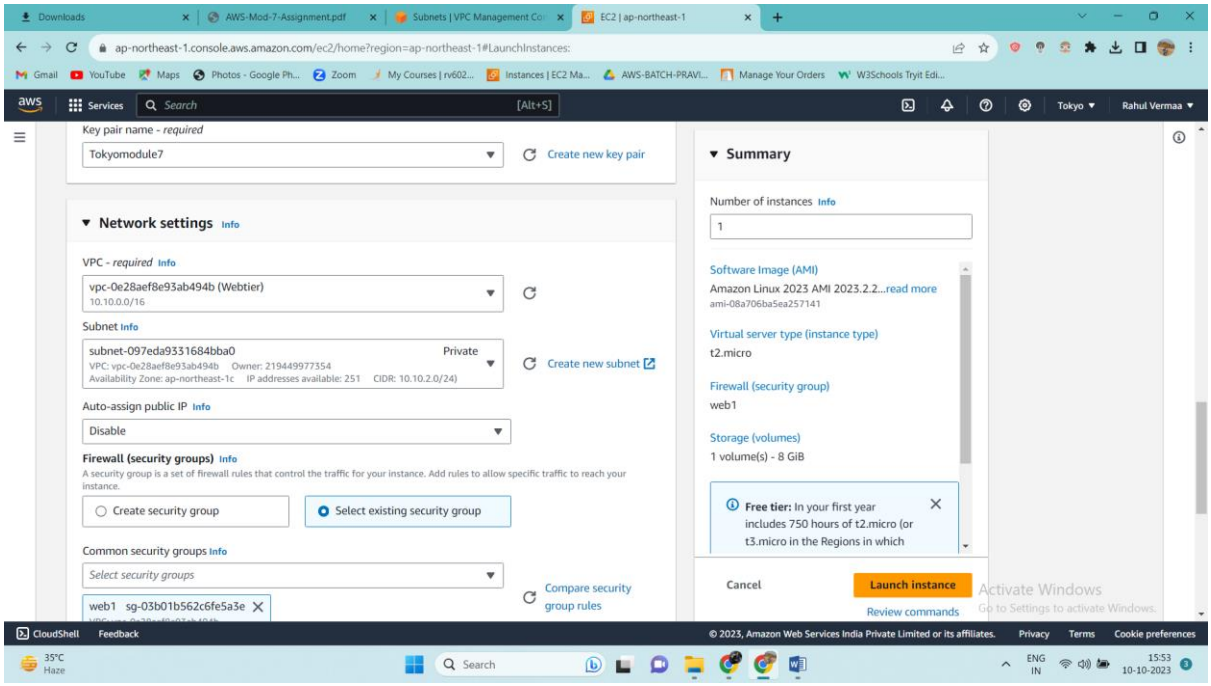
Public instance is created



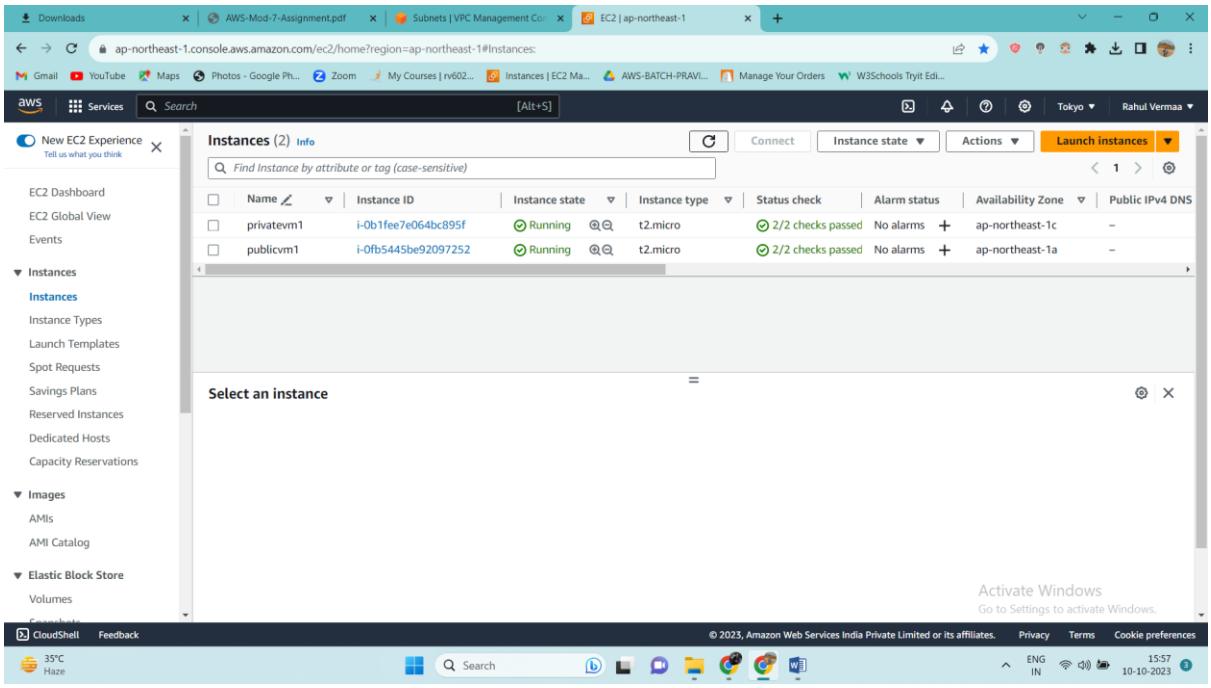
Similarly we will create one instance with private subnet



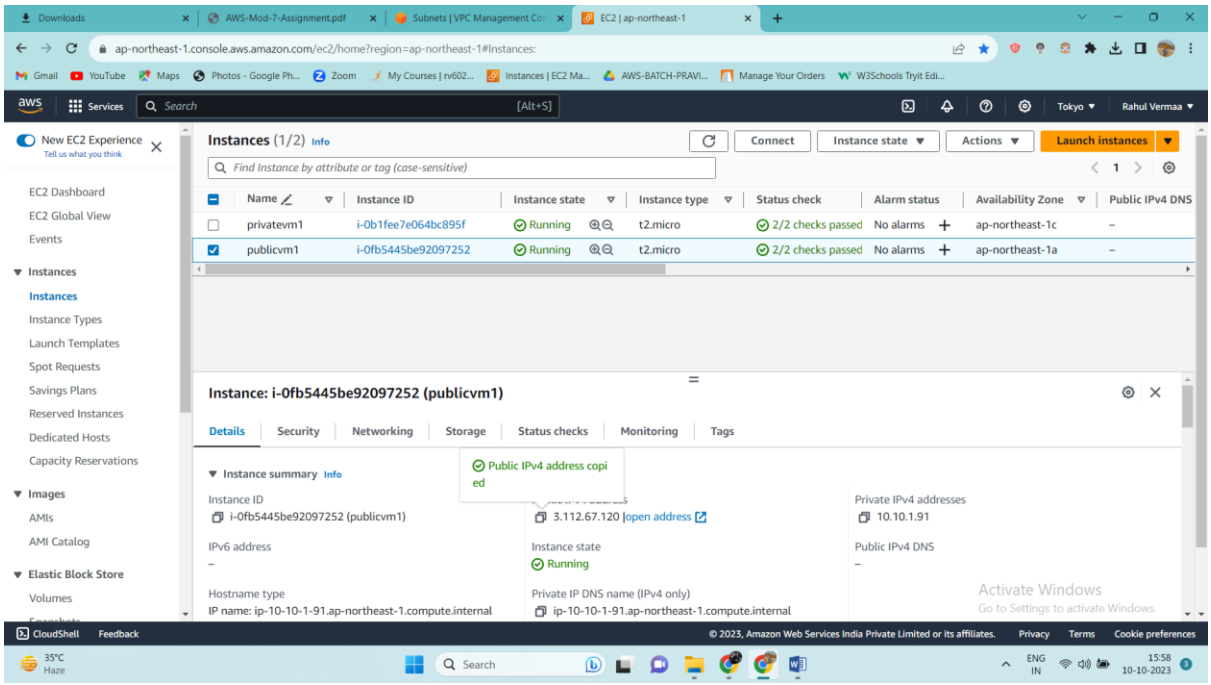
Select options like vpc, subnet, auto assign public ip-disabled etc…



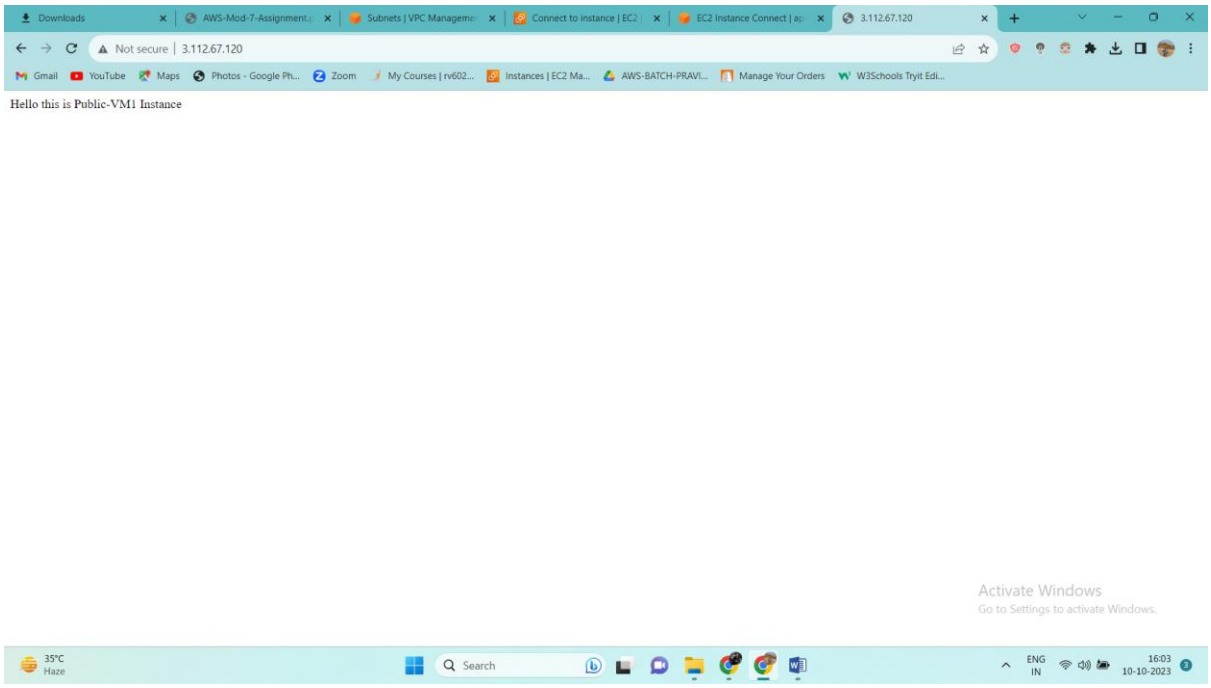
Both instance are running now



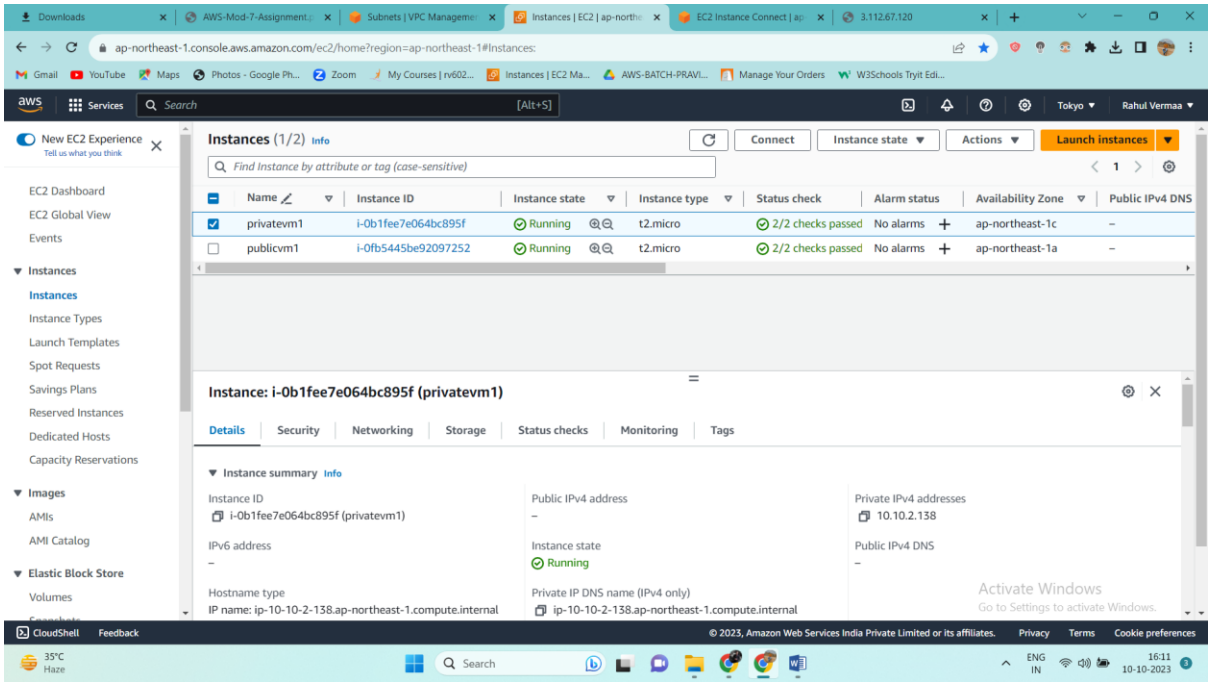
Now let’s connect to our instance and install some http file to display some data first will connect to publicvm1



Html file installed

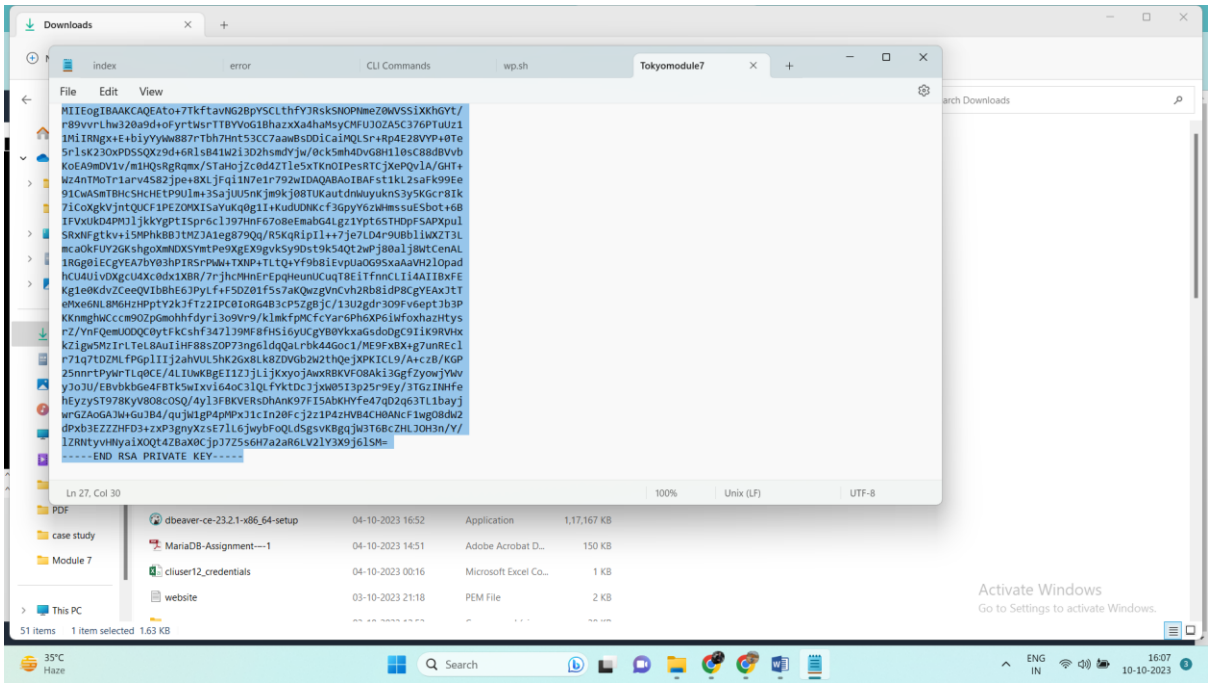


Now let’s try to connect with our privatevm1 instance where there is no ipv4 enabled

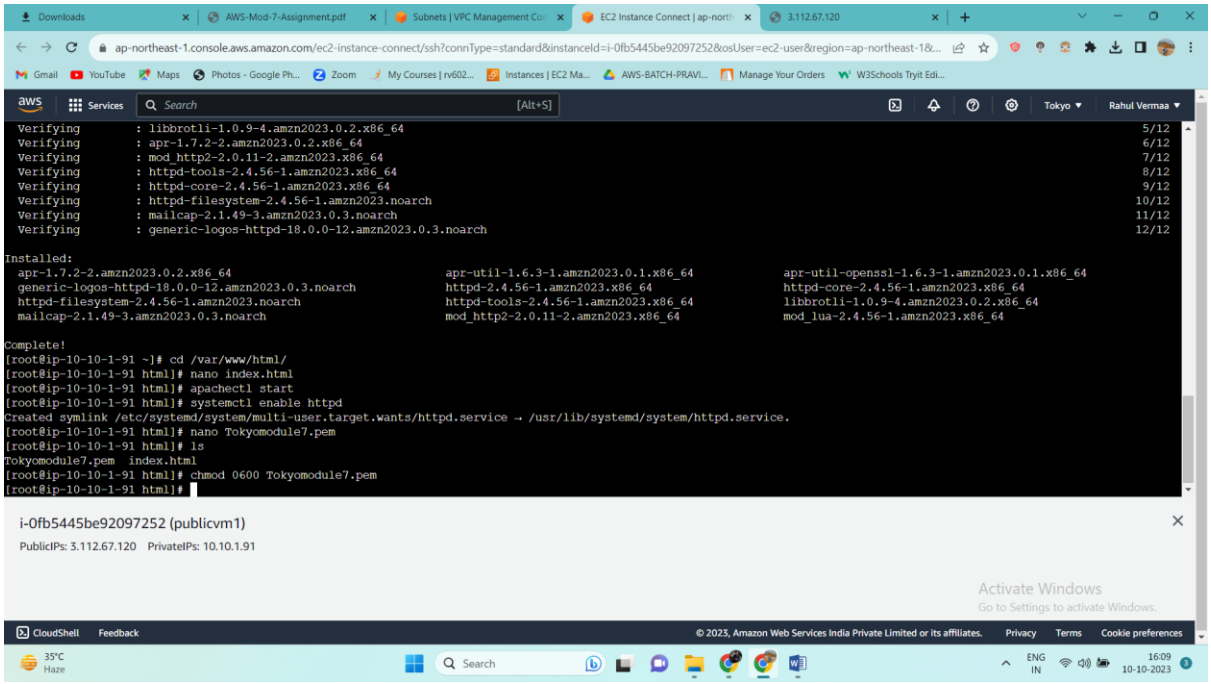


For this we have to create a path in our publicvm1

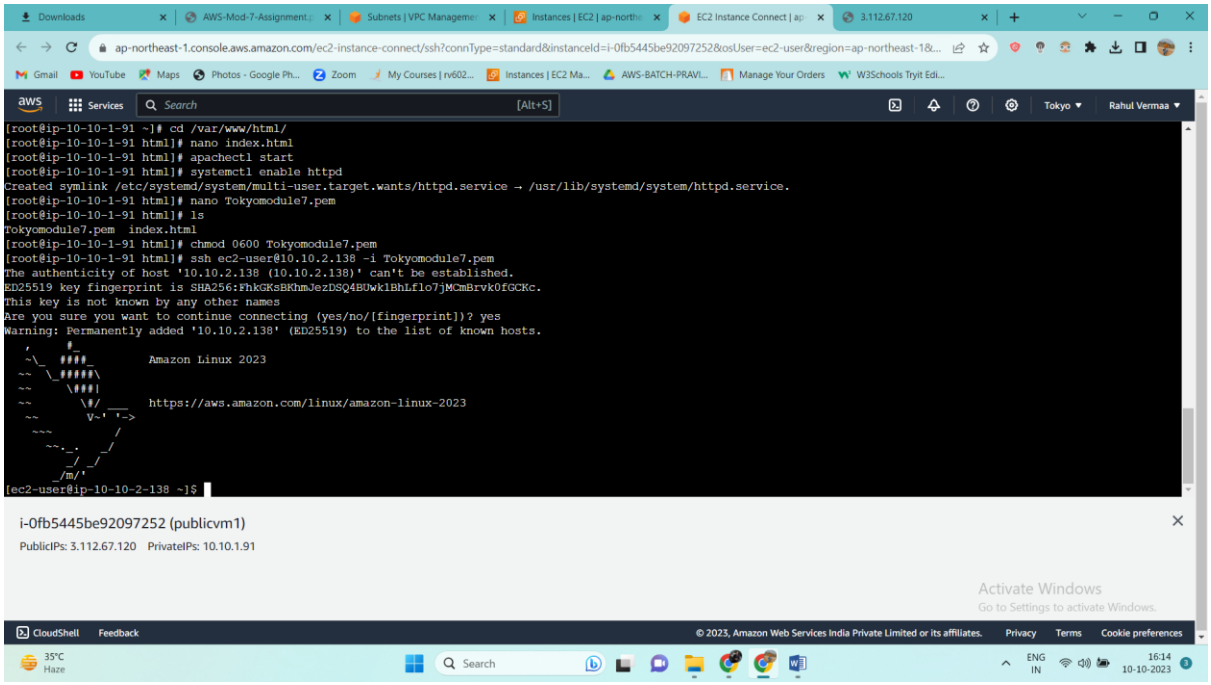
Create a file Tokyomodule7.pem and paste the key content in it



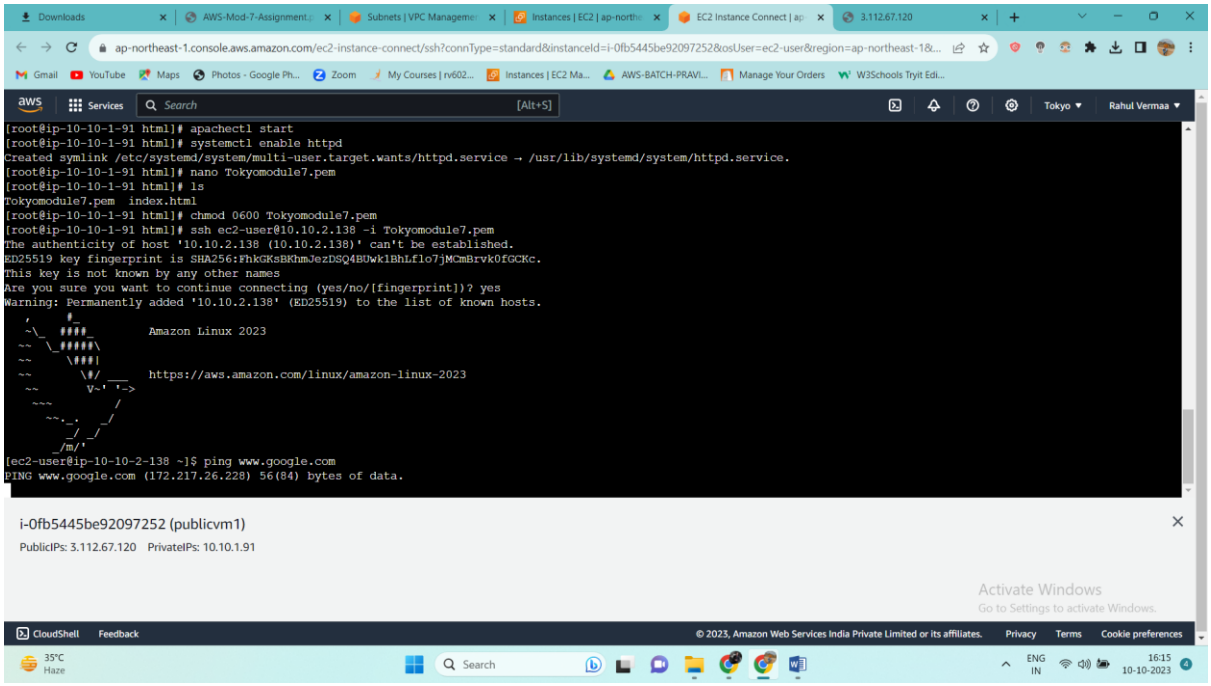
Allow that Tokyomodule7.pem file to read the content



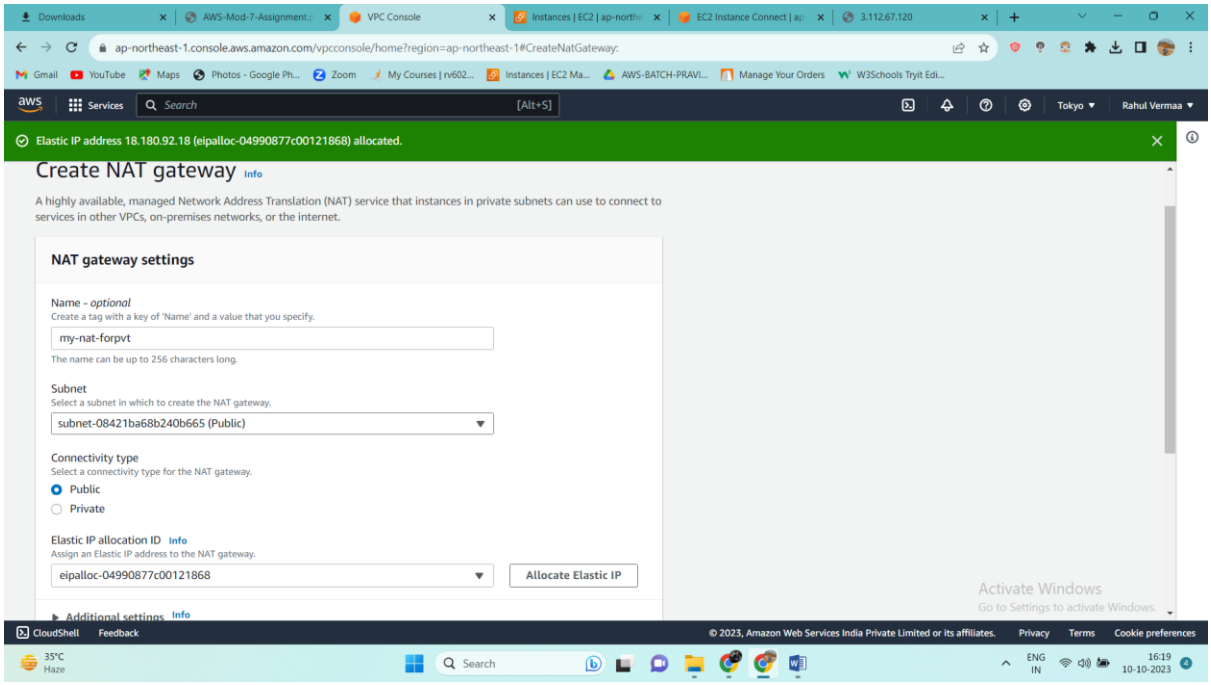
Now ssh from publicvm1 to connect privatevm1 and we are successfully connected to privatevm1



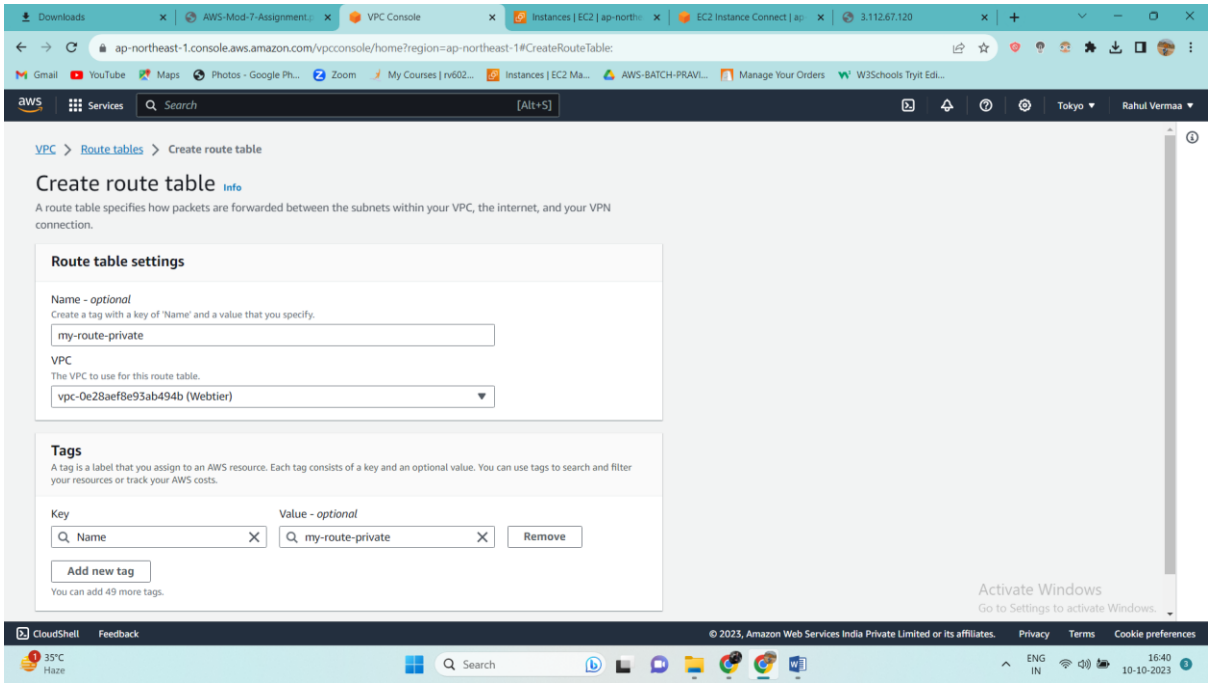
Internet is not there to give internet accces to our privatevm1 we will install one Nat gateway



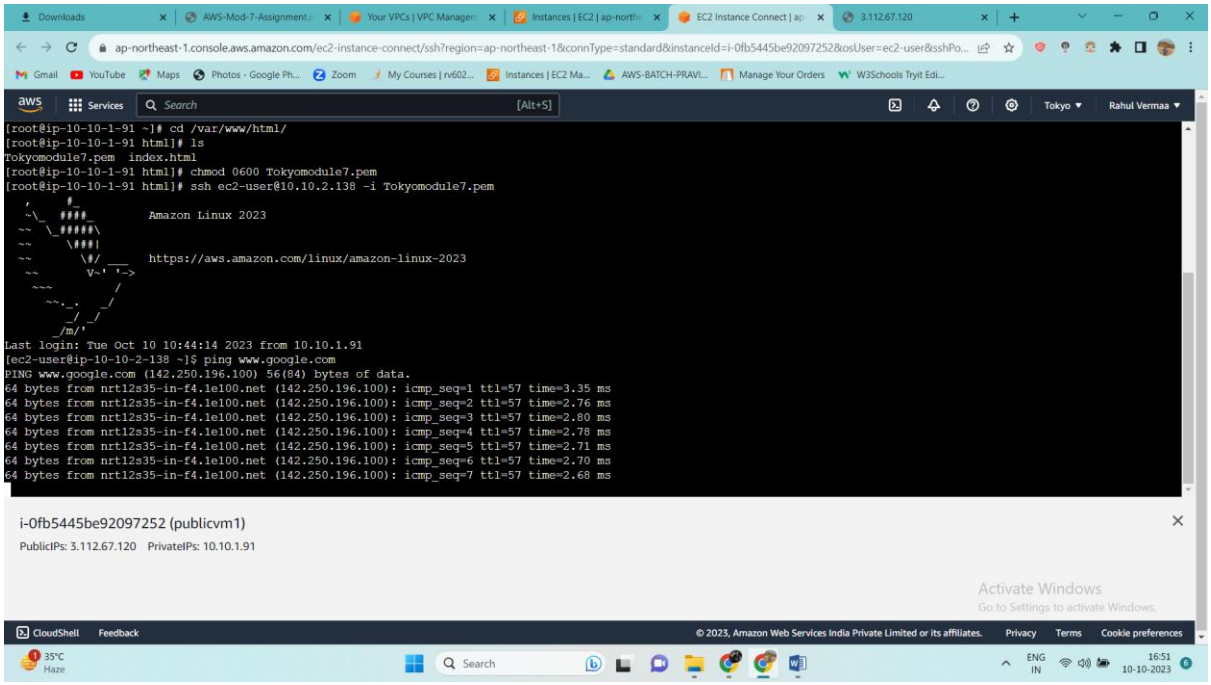
Let’s create one NAT gateway (we are allocating elastic ip)



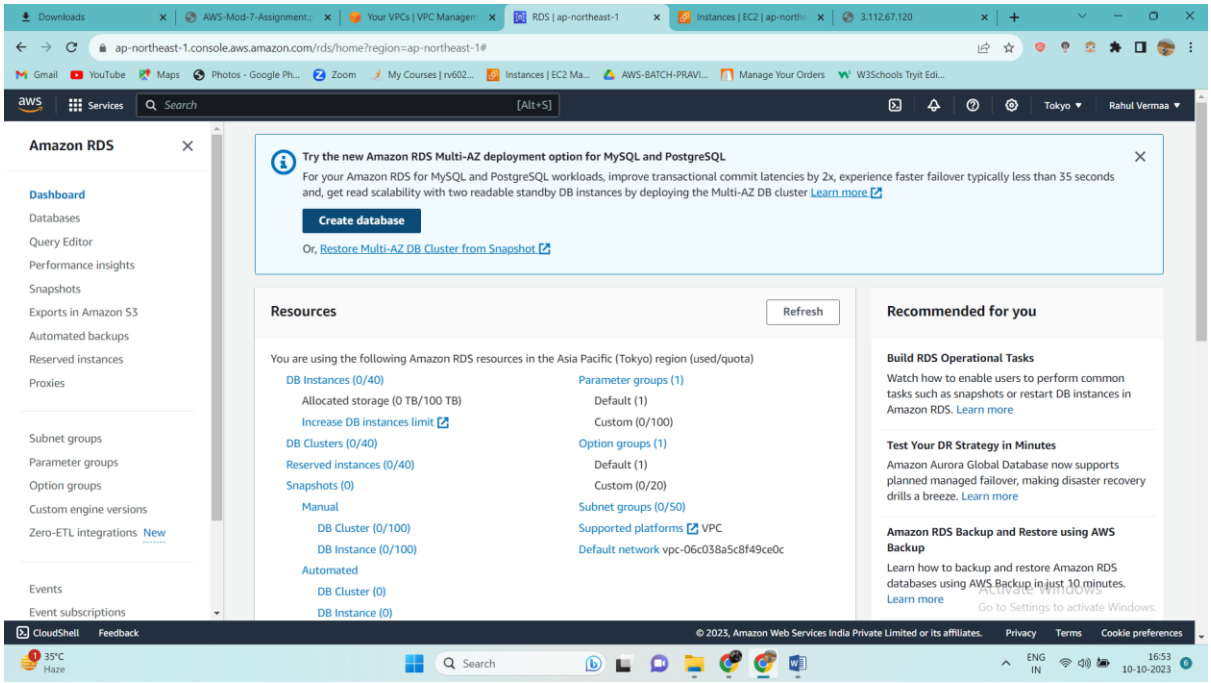
Now create a route table and route the traffic to Nat-gateway



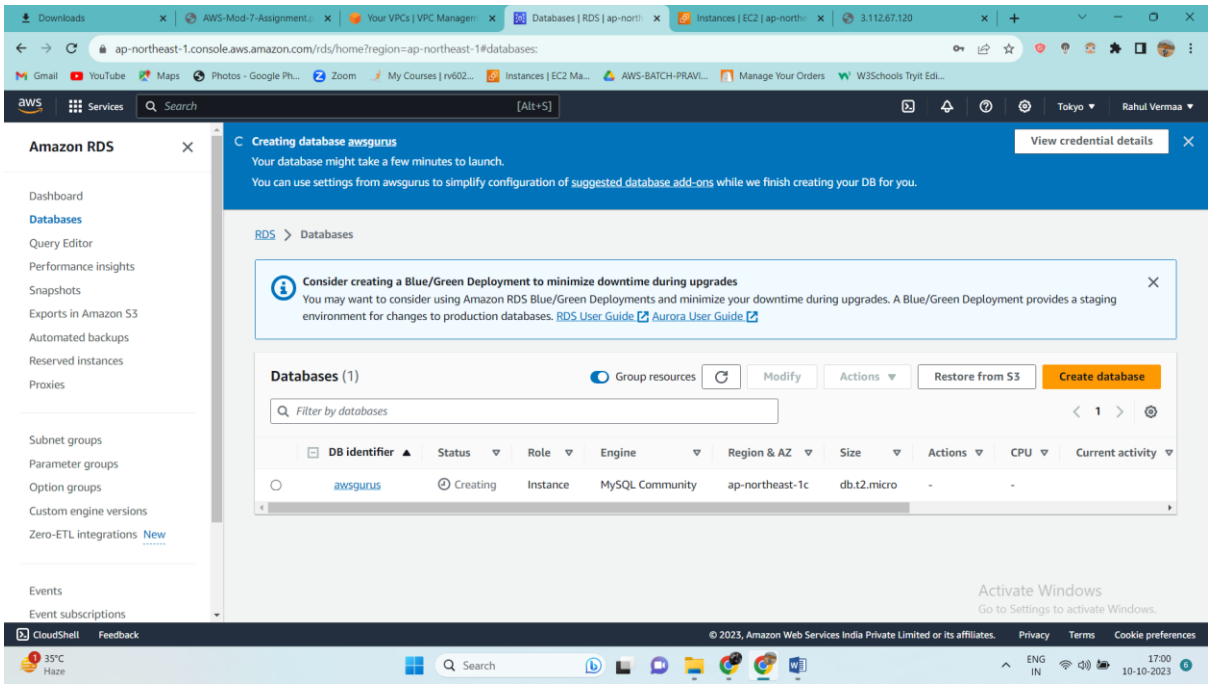
Now we are able to ping www.google.com that means internet is connected to our privatevm1



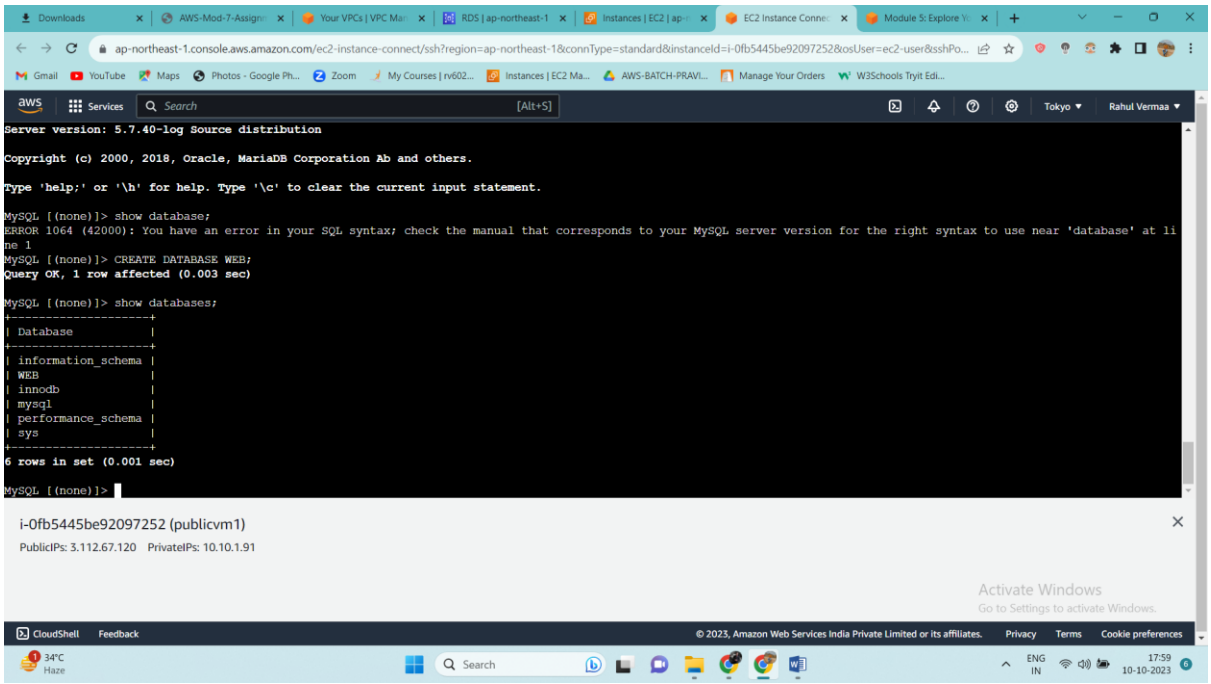
Now let’s create one RDS database



Our RDS is created in subnet private

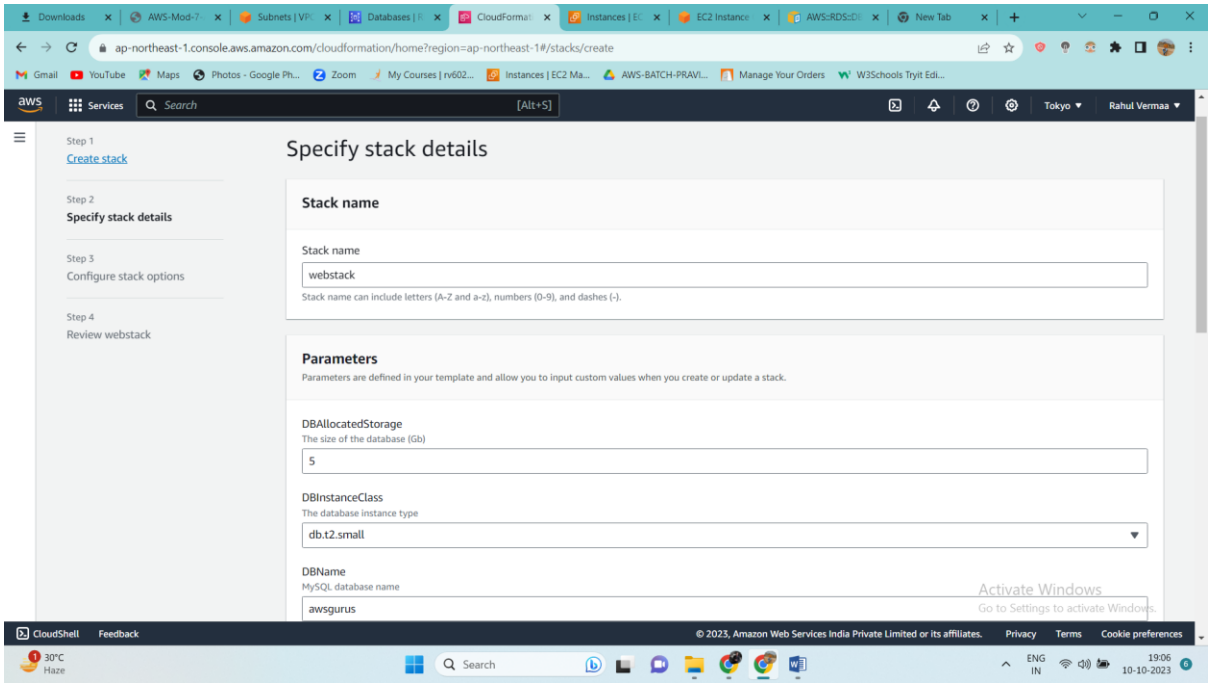


And we are connected to our RDS



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