Project - 1: Deploying a Multi-Tier Website Using AWS EC2

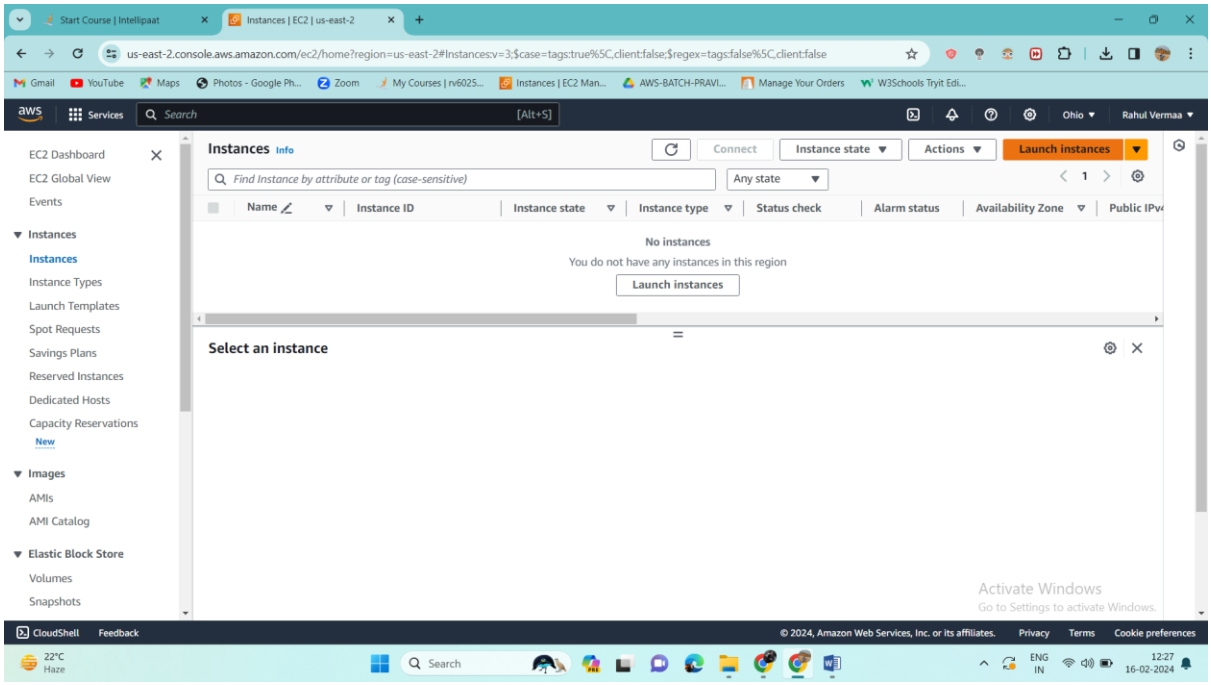
Description: Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) cloud. Using Amazon EC2 eliminates your need to invest in hardware up front so you can develop and deploy applications faster. You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. Amazon EC2 enables you to scale up or down to handle changes in requirements or spikes in popularity, reducing your need to forecast traffic.

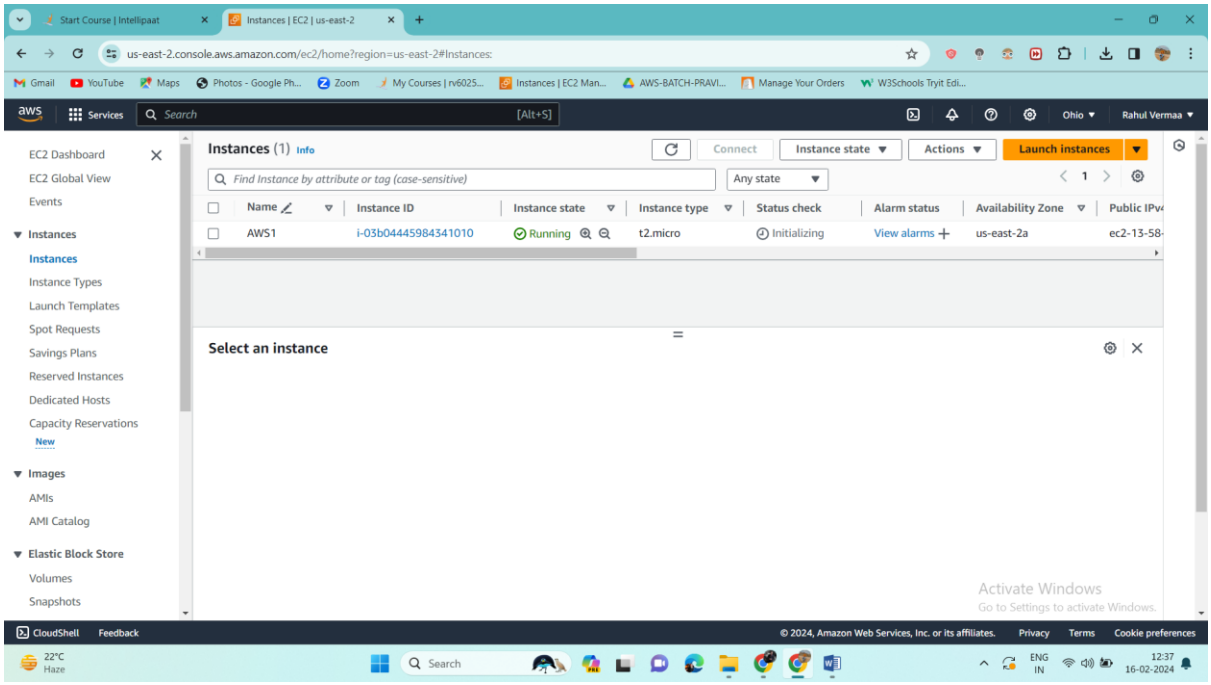
Problem Statement: Company ABC wants to move their product to AWS. They have the following things set up right now: 1. MySQL DB 2. Website (PHP) The company wants high availability on this product, therefore wants Auto Scaling to be enabled on this website.

Steps To Solve:

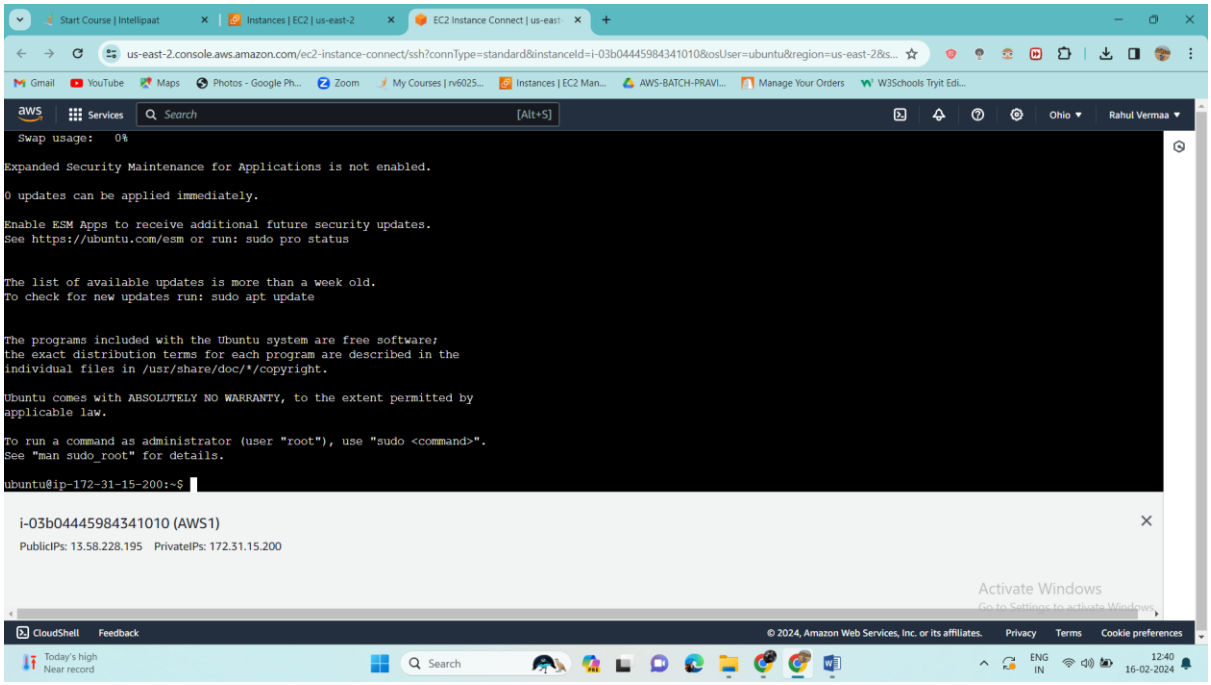
1. Launch an EC2 Instance
2. Enable Auto Scaling on these instances (minimum 2)
3. Create an RDS instance.
4. Create database & table in RDS instance.
    a. Database name: intel
    b. Table name: data
    c. Database password: intel123
5. Change host name in website
6. Allow traffic from EC2 to RDS instance
7. Allow all traffic to EC2 instance.

Open AWS console and create an instance





Now login to your ec2 instance



>First Update your system using the command-

sudo apt-get update

>Then use this command in Putty to install Apache2-

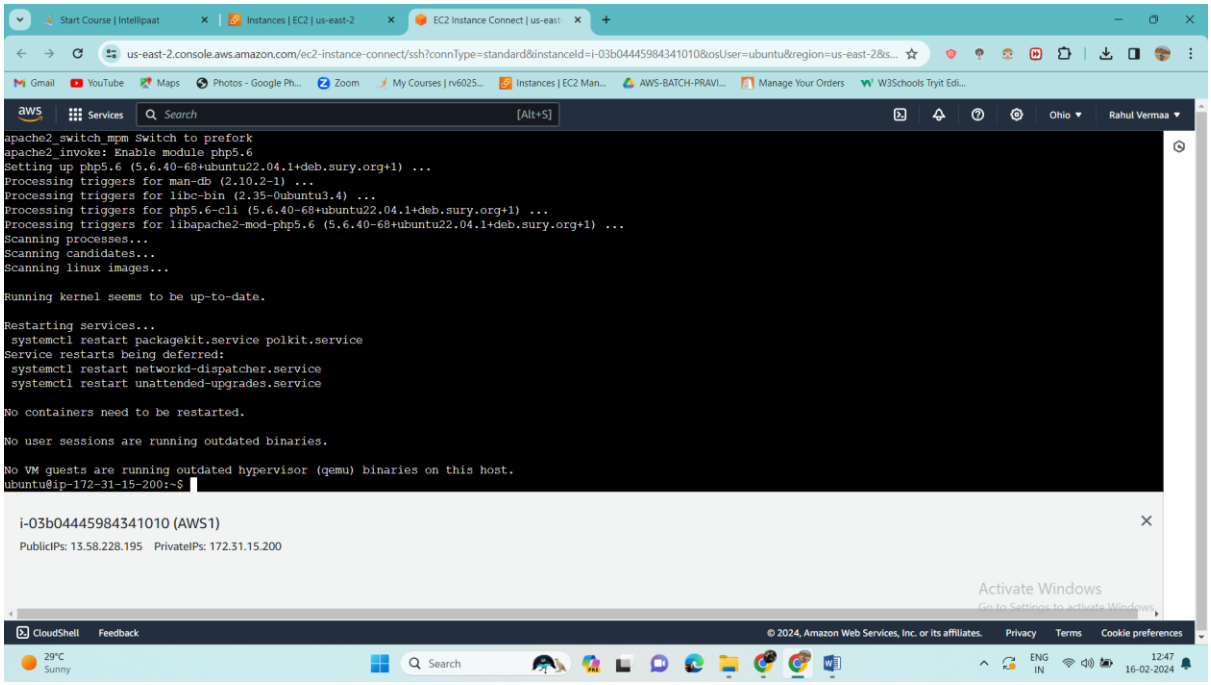
sudo apt-get install apache2 -y

> Then install php-mysql using the following command-

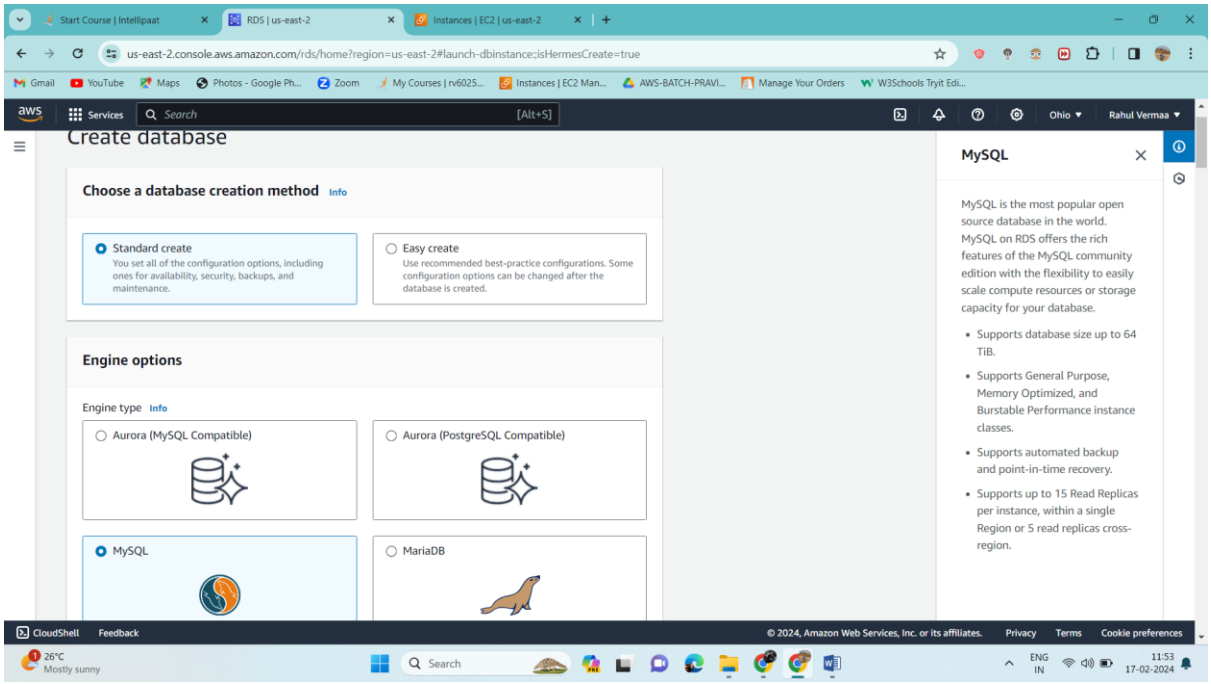
sudo add-apt-repository -y ppa:ondrej/php

sudo apt install php5.6 mysql-client php5.6-mysqli –y

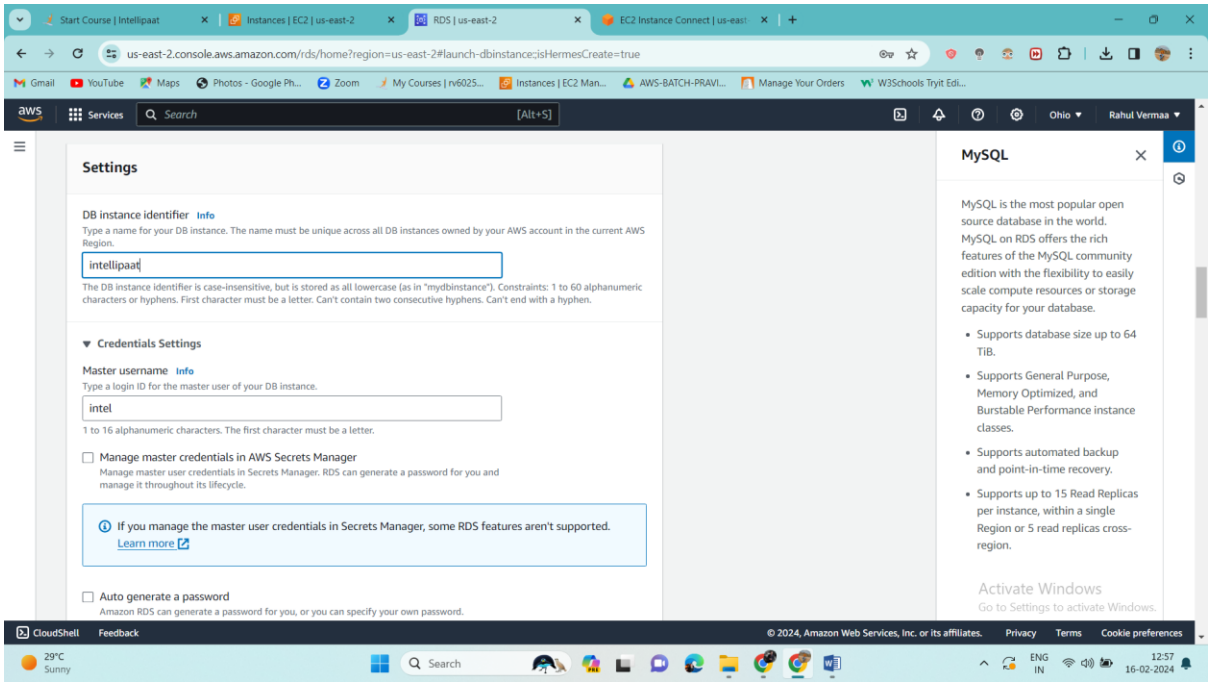
Now everything is updated in your system



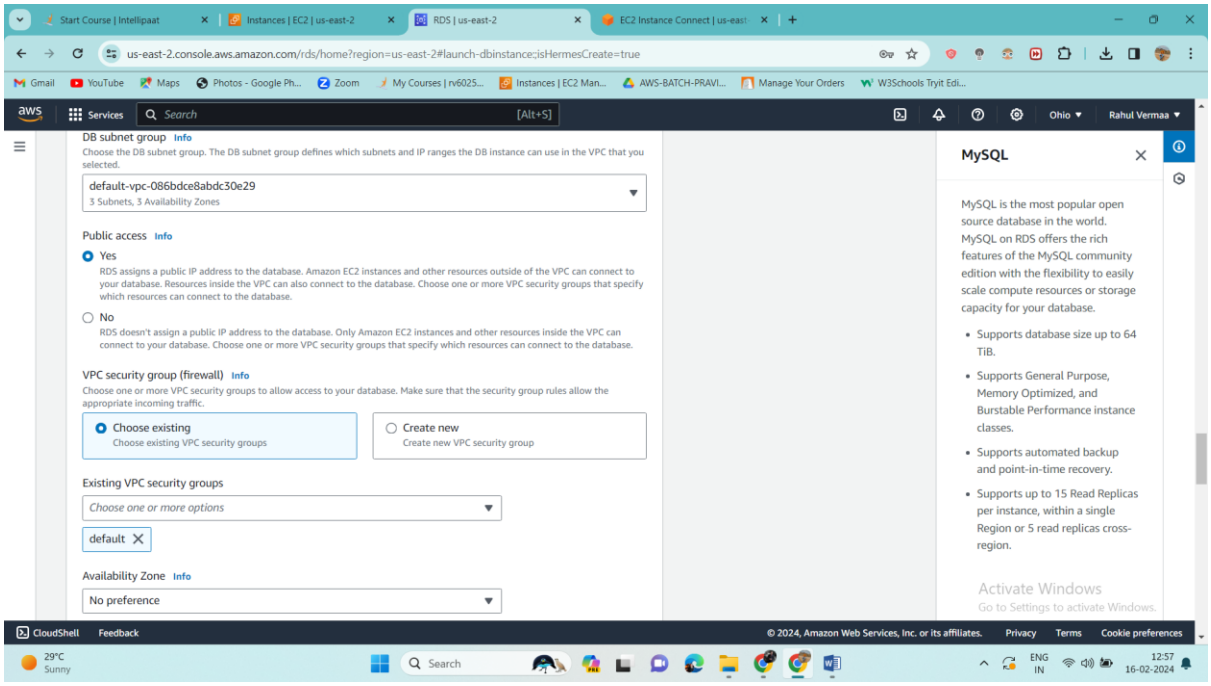
Now create RDS



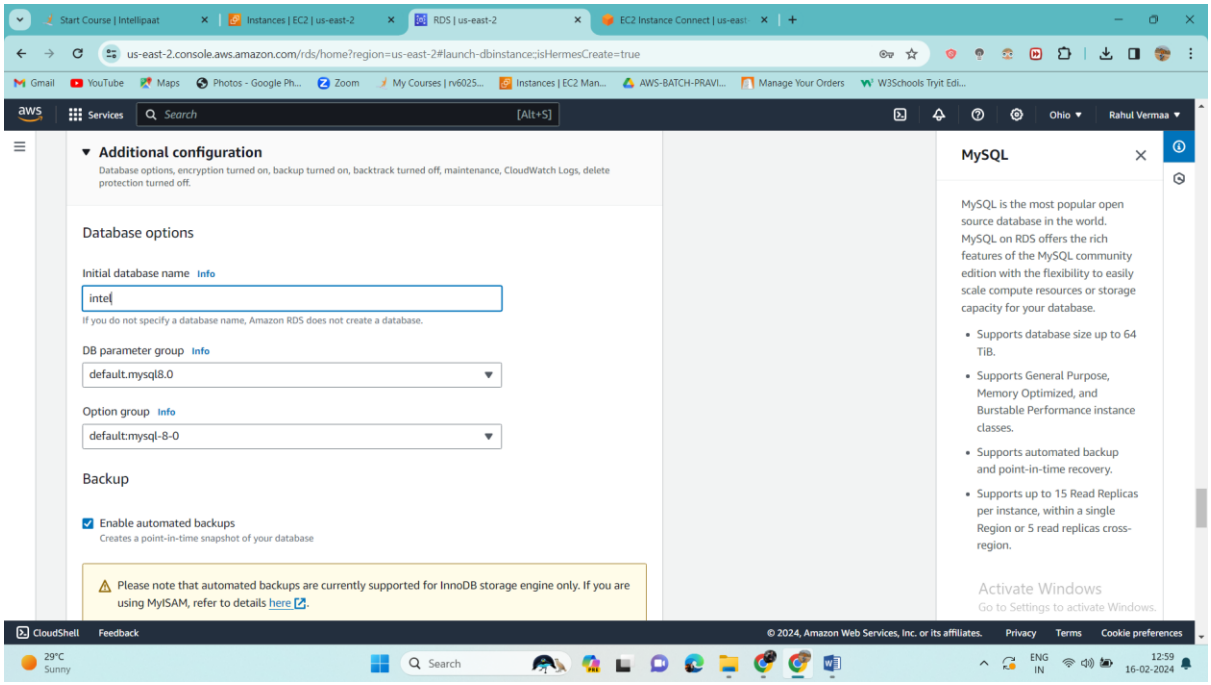
Set password as- intel123



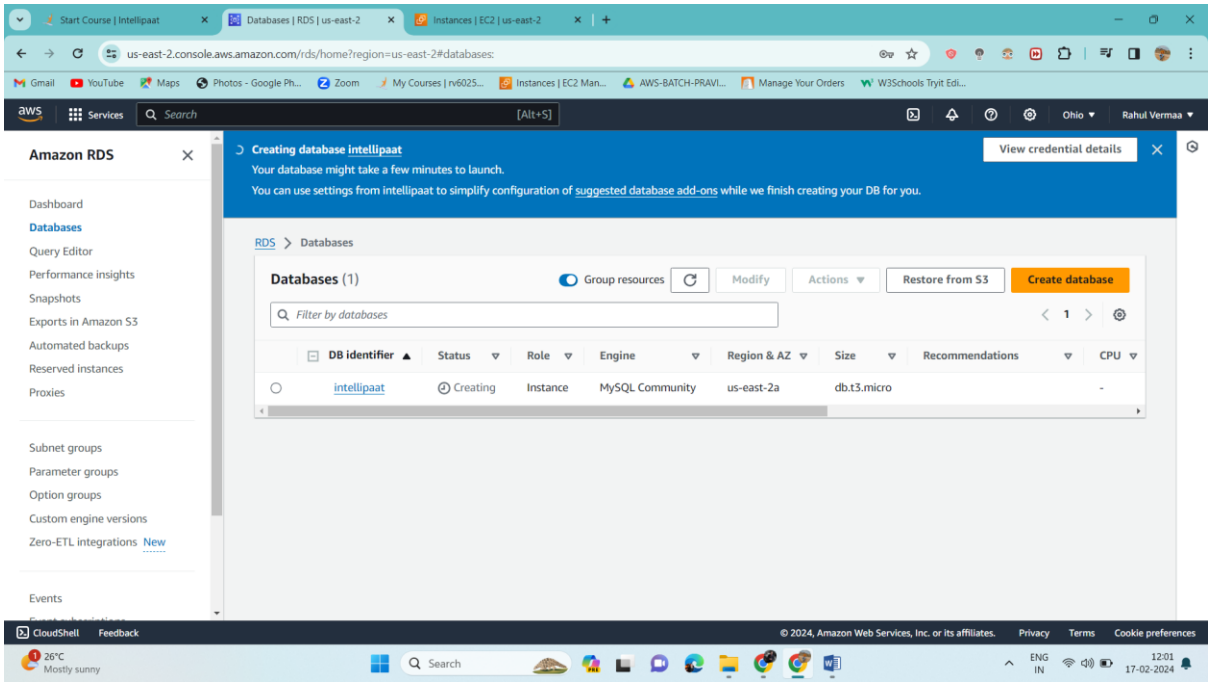
Default VPC

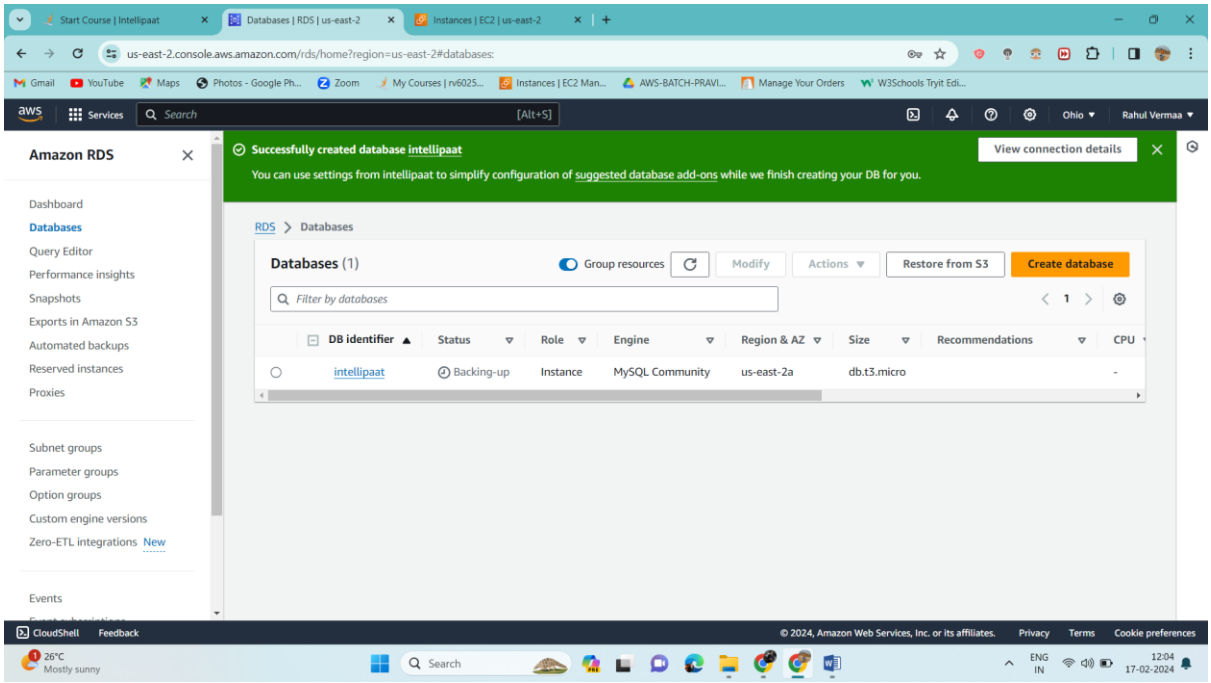


Database name- intel



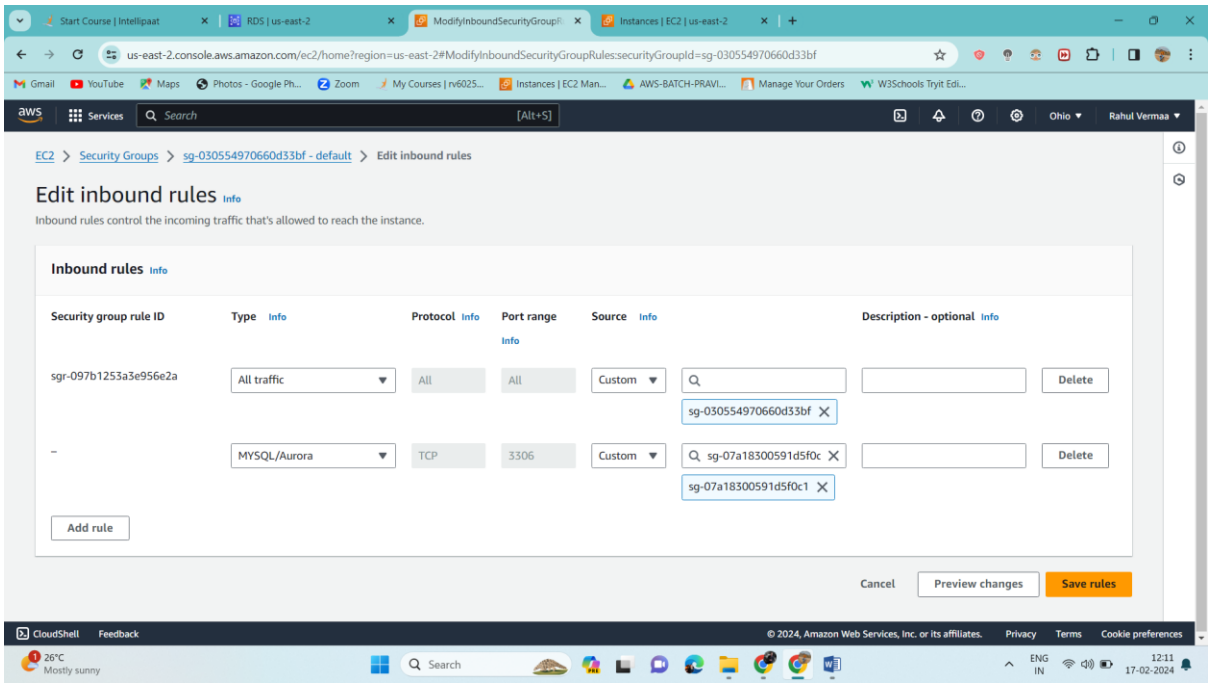
RDS is created now





Now we have to configure RDS security group settings

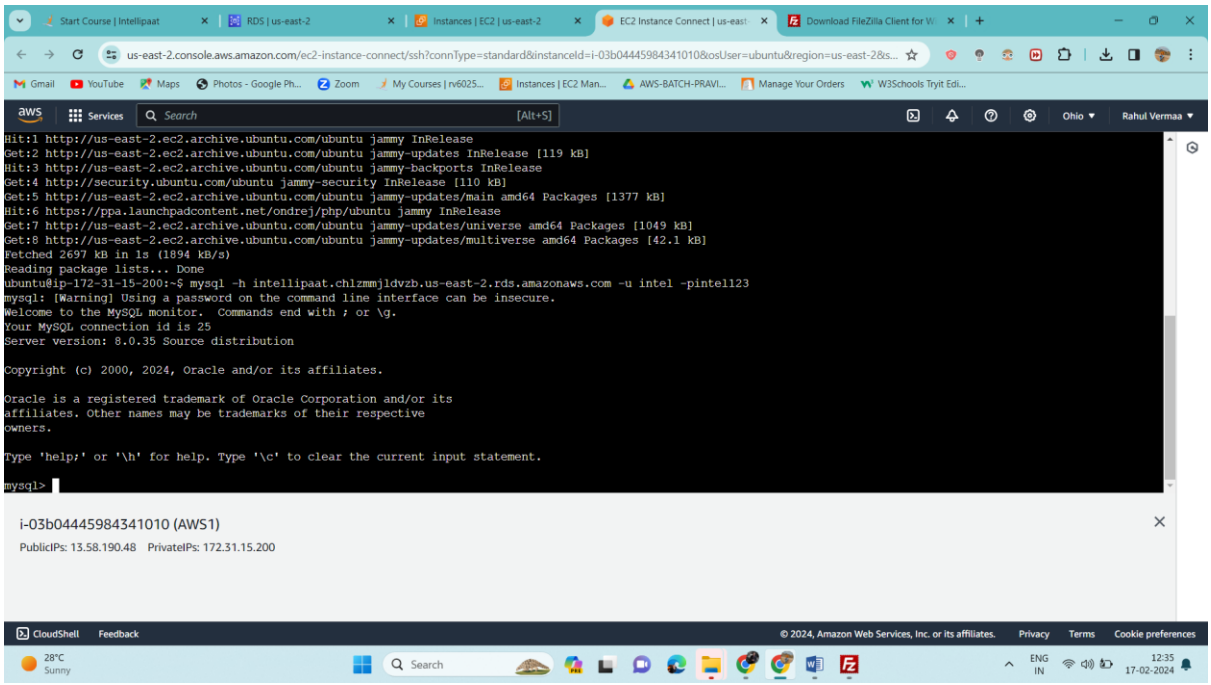
Then paste the EC2 Security ID in Source> Custom > Security Group by keeping the Type as MYSQL/Aurora



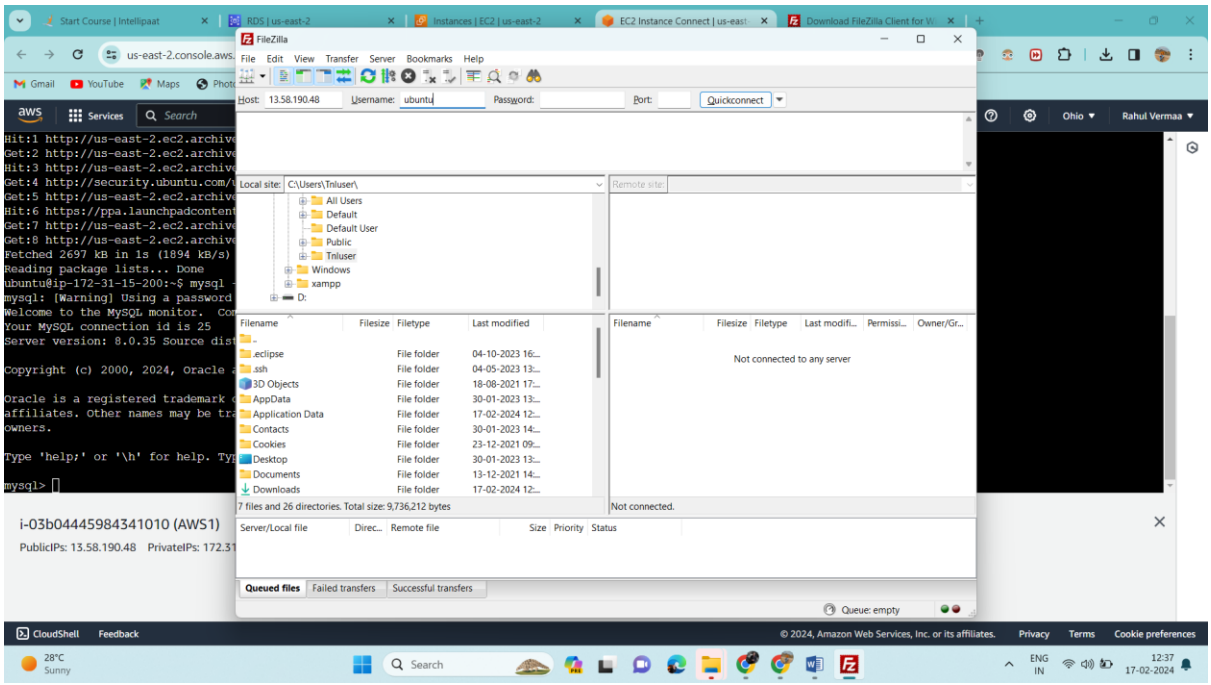
Now we will connect to mysql using the below command

mysql –h (End point of RDS) –u (username- intel) –p(password without space- intel123)

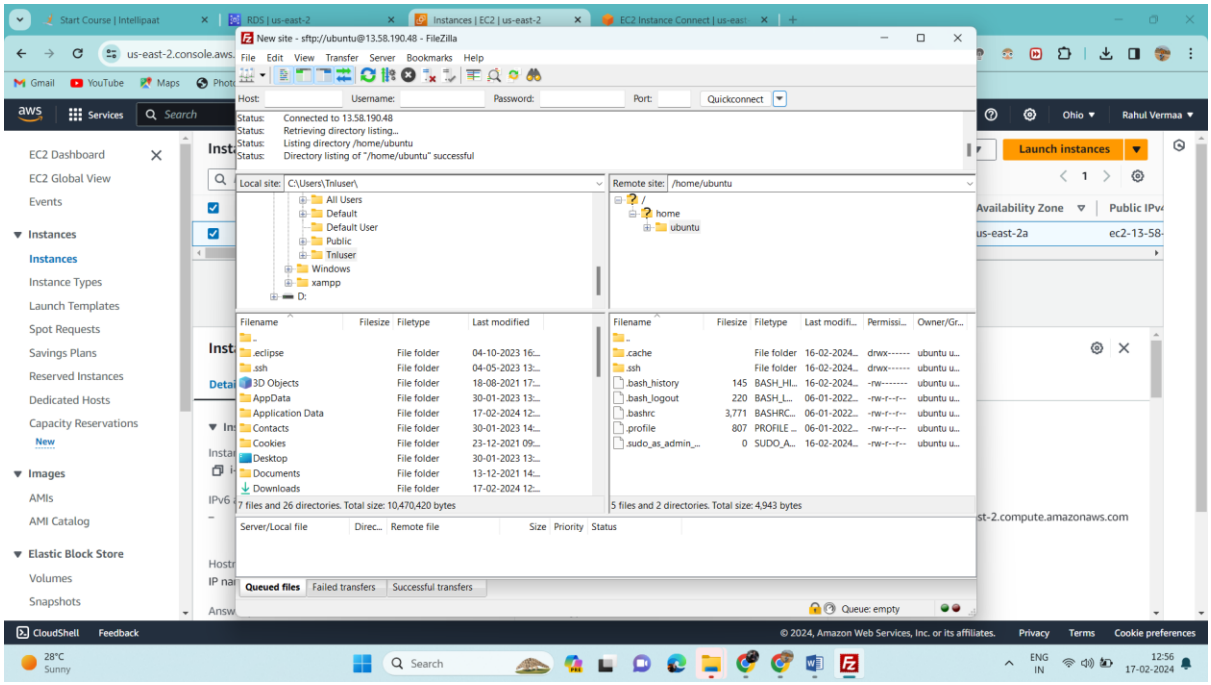
-pintel123



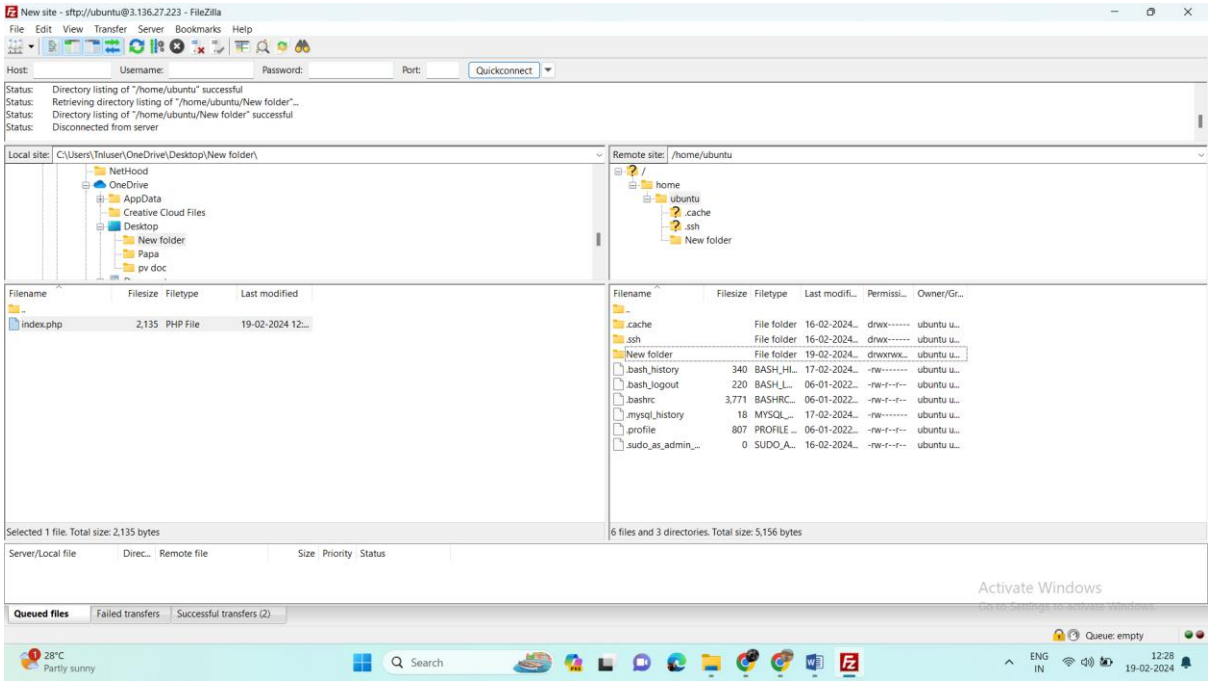
Now we'll use filezilla Host- End point of ec2 and username- Ubuntu and click on connect



We are connect our ec2 with filezilla



Now we have created one ‘New folder’ and pasted our website in it

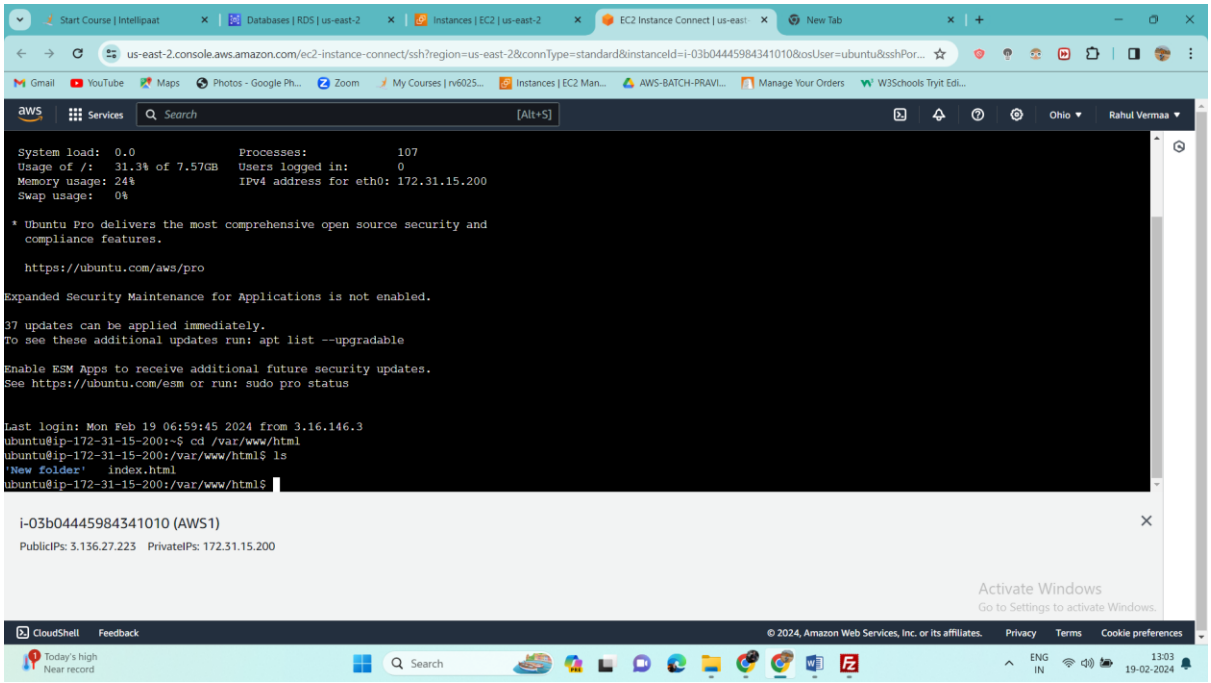


Now by using some commands will check our file

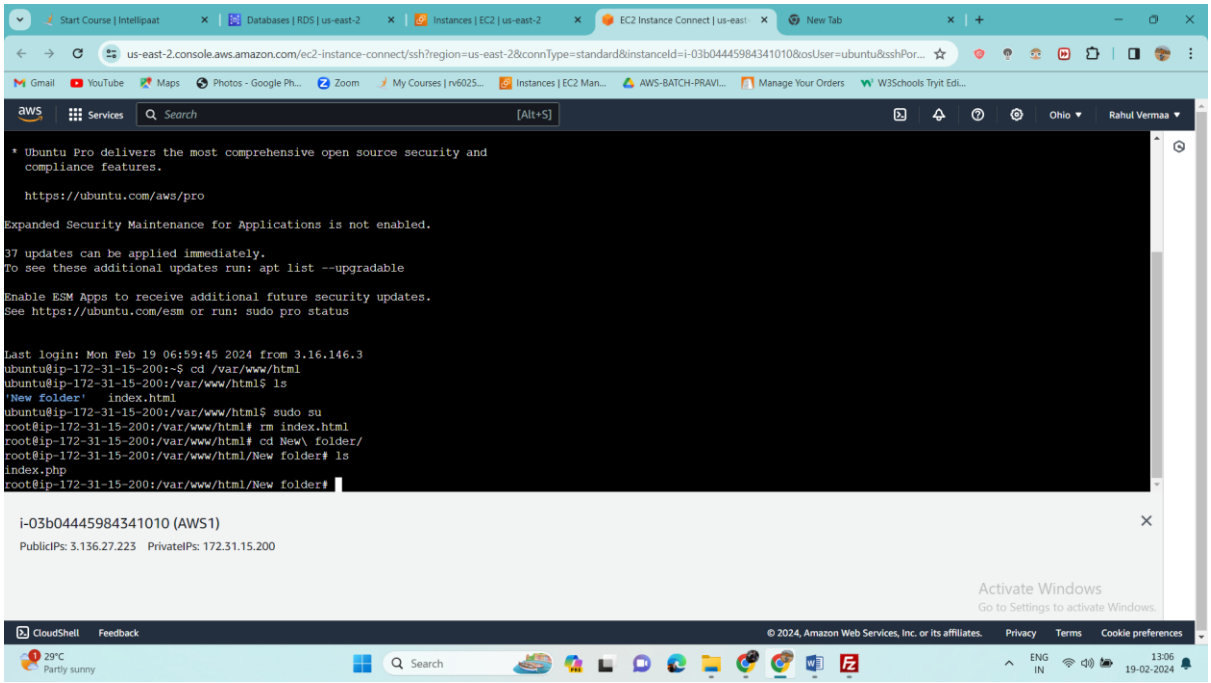
sudo cp -r New\ folder/ /var/www/html

cd /var/www/html

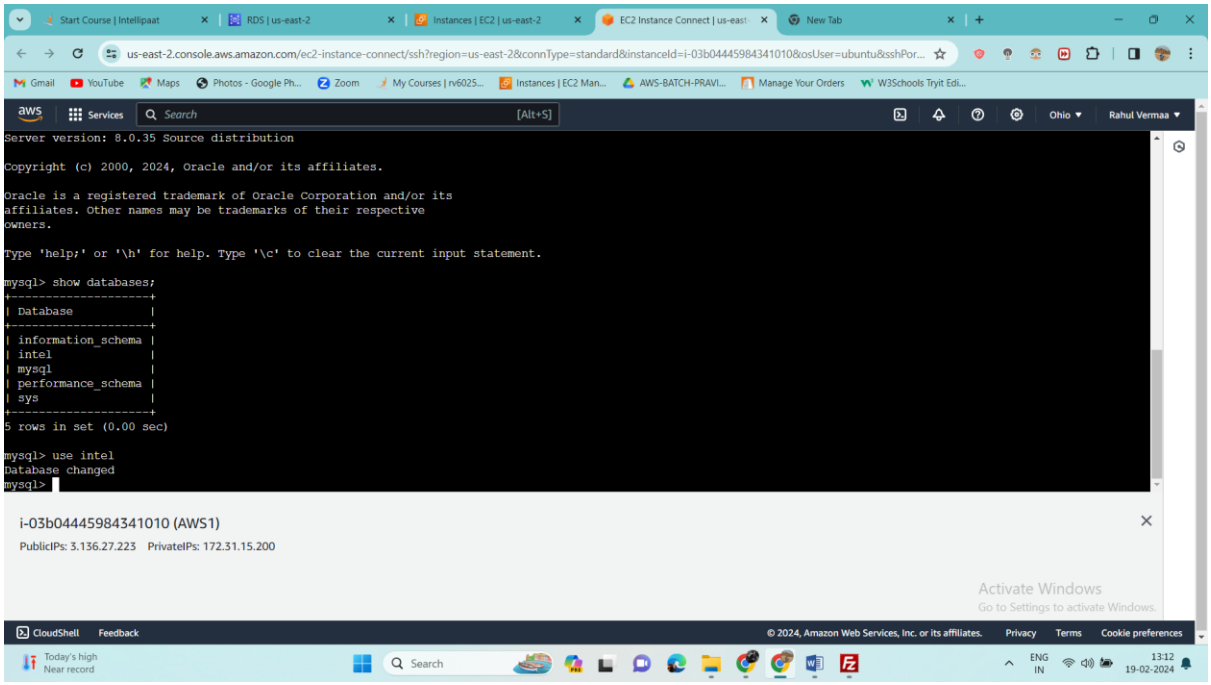
ls



Now we will remove index.html file and will add index.php

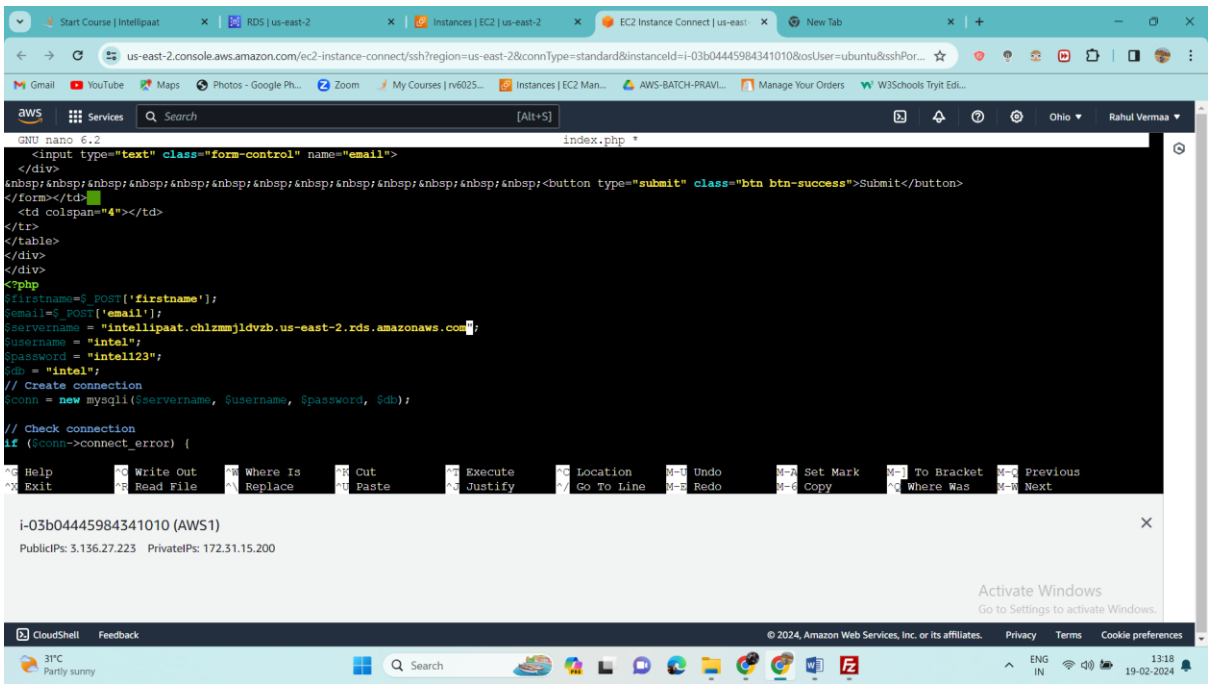


Now connect to mysql

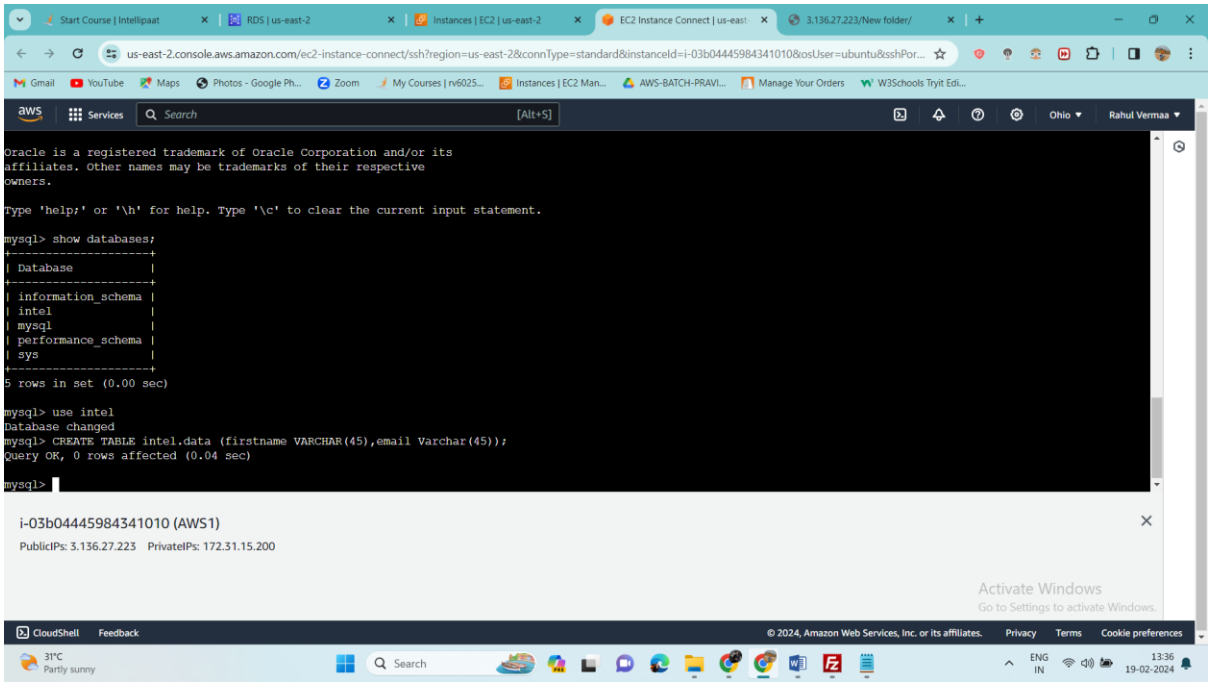


Now go to the path where website files are kept and run the index.php file by using sudo nano index.php

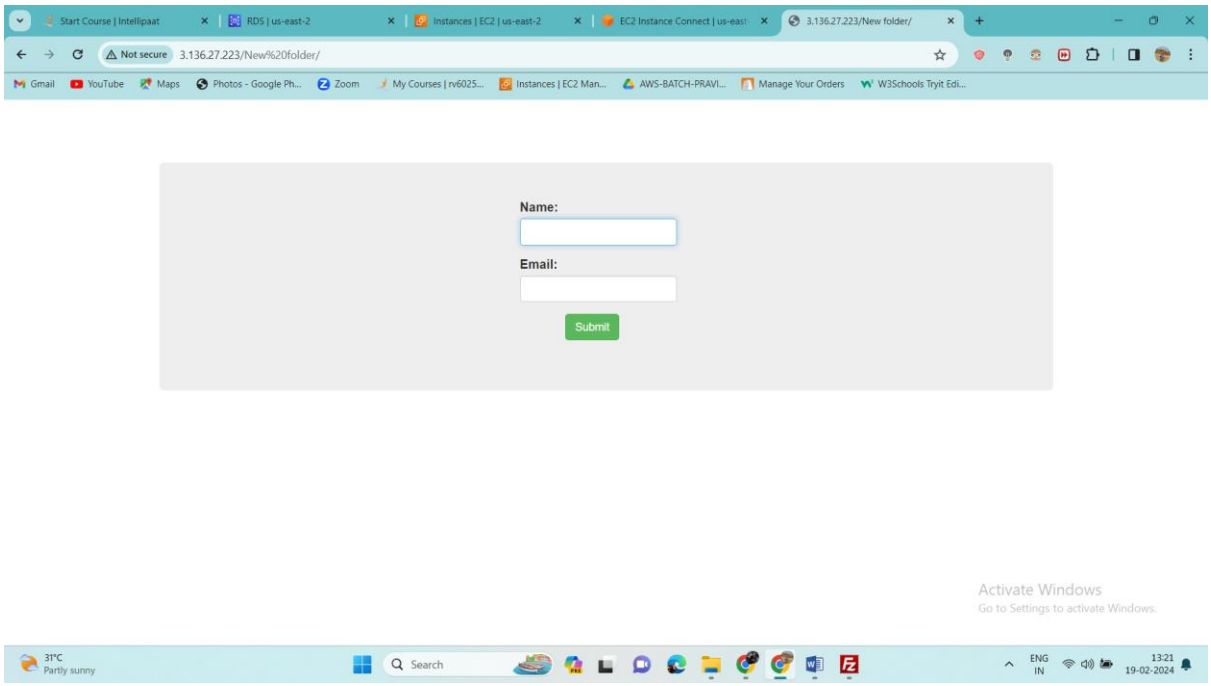
Now after this, GNU nano will pop up where you have to make changes in your code, you have to check if in your server name, the endpoint of your RDS is there along with username, password and db name



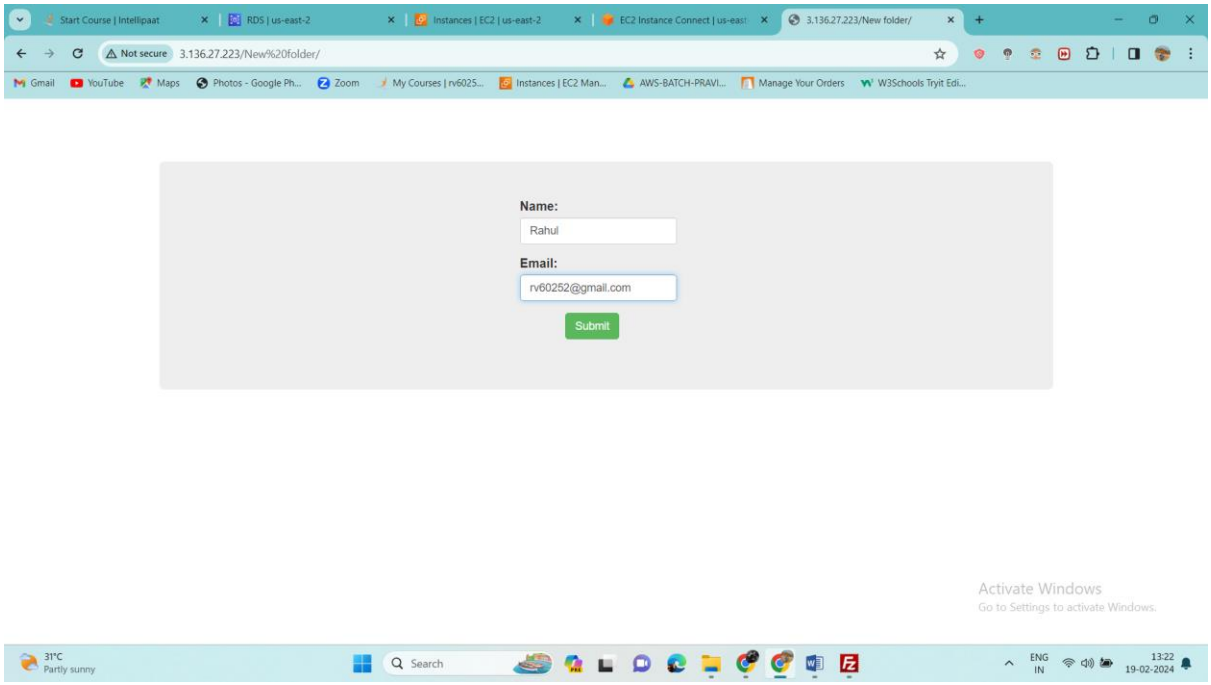
We have created one table in our intel databases



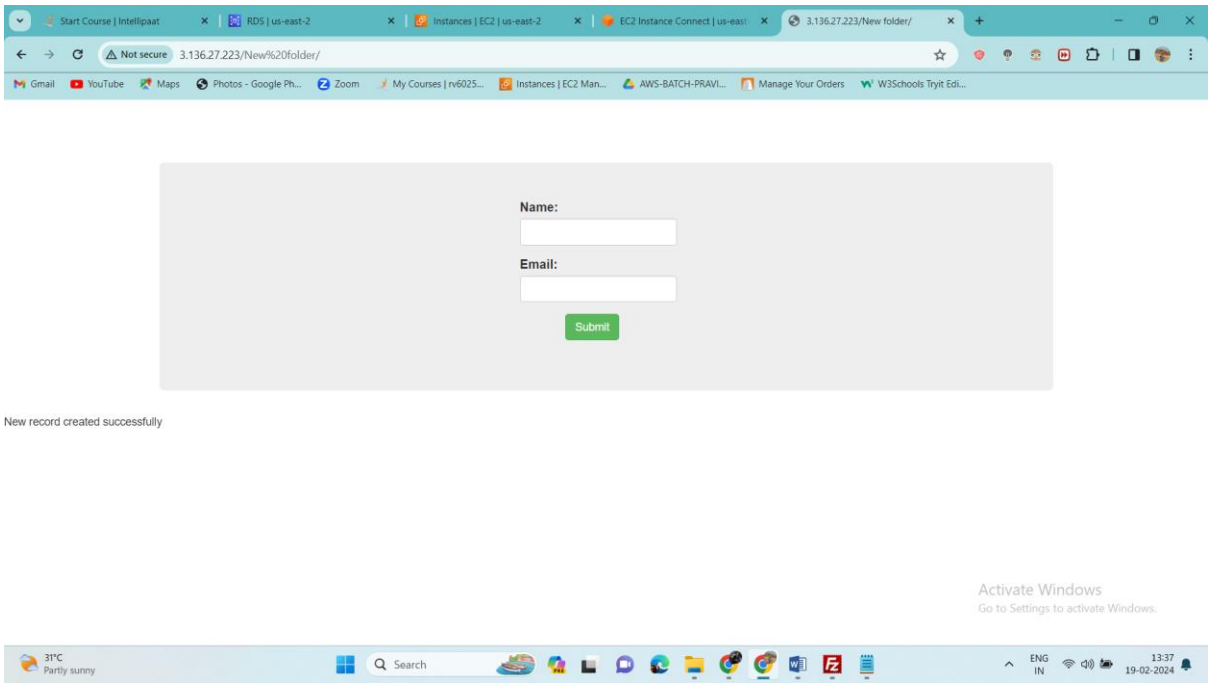
Now when you will try, and copy paste the Public IP of your EC2 Instance



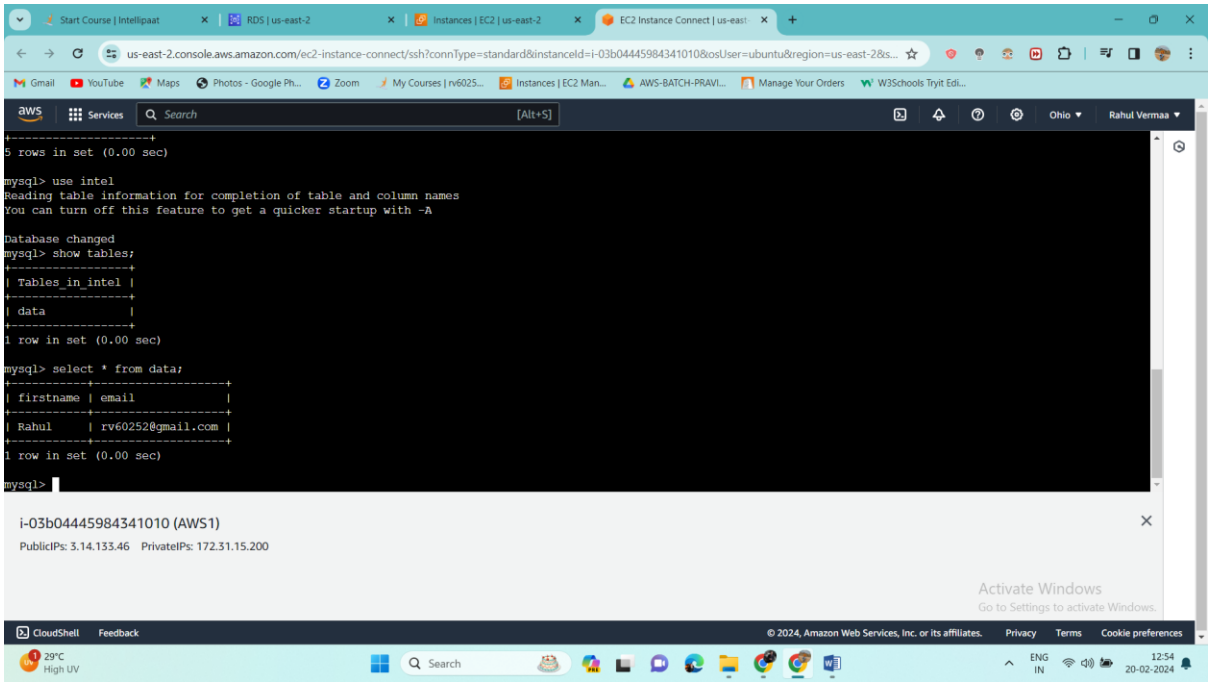
Now let’s add some data in it



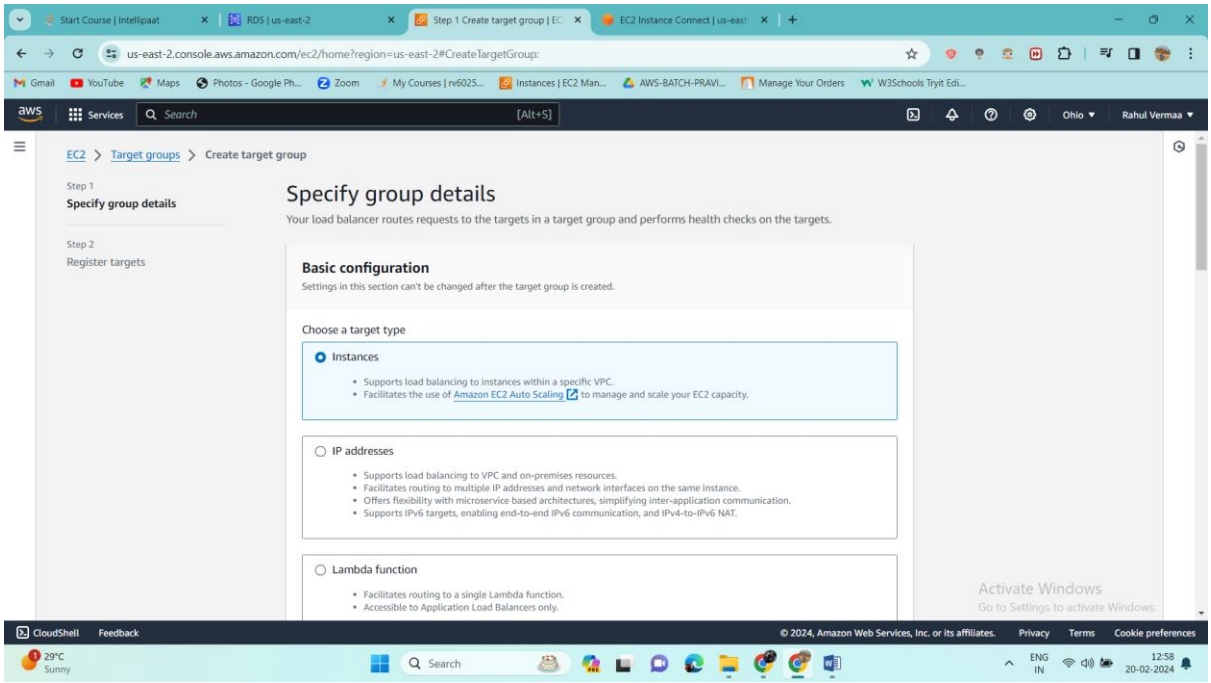
And our data is added successfully



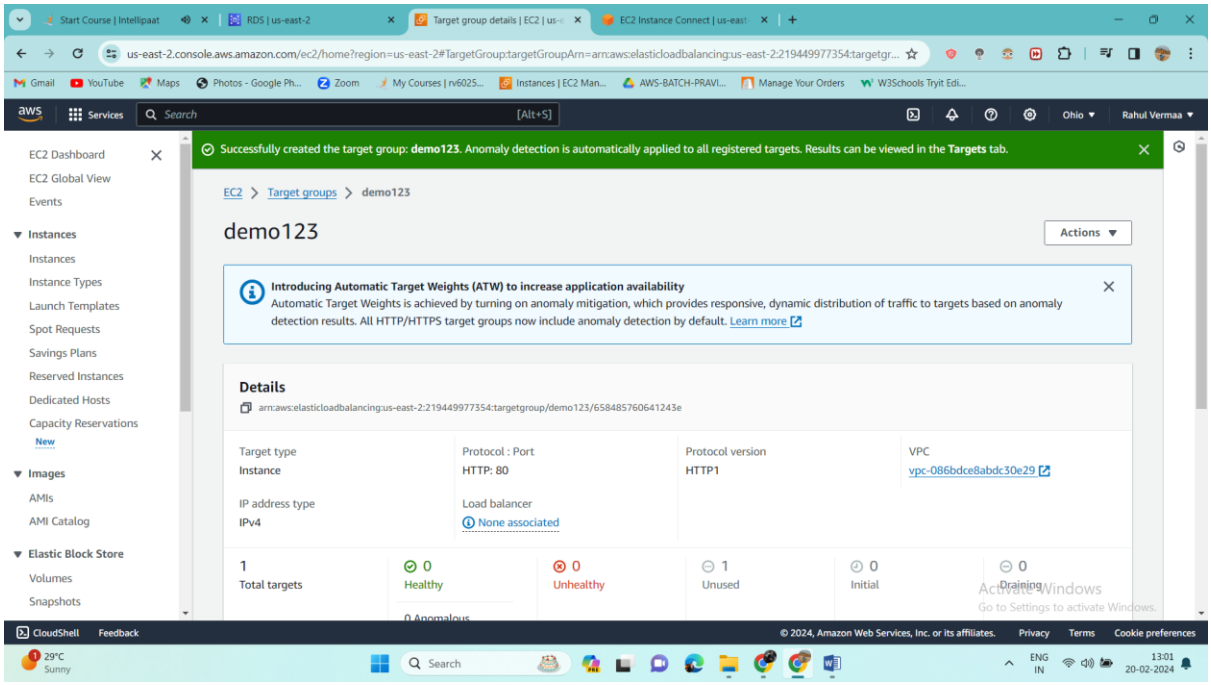
Let’s check it in our my sql



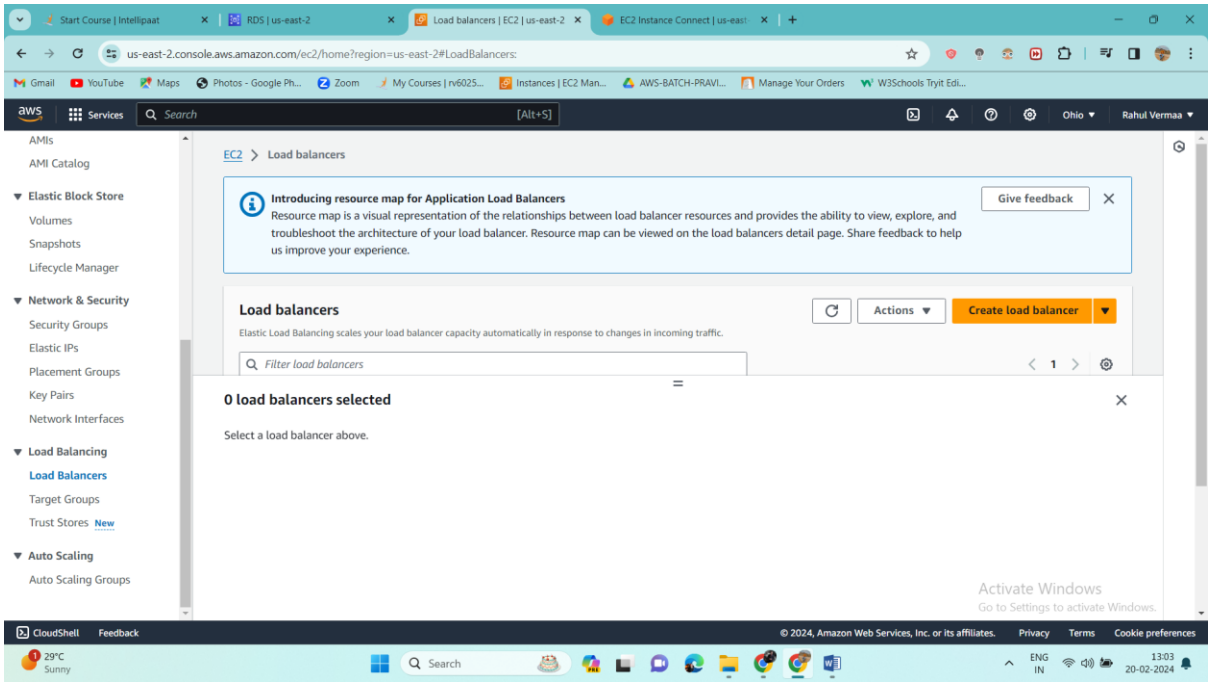
Now we will create load balancer, before that we will create Target group



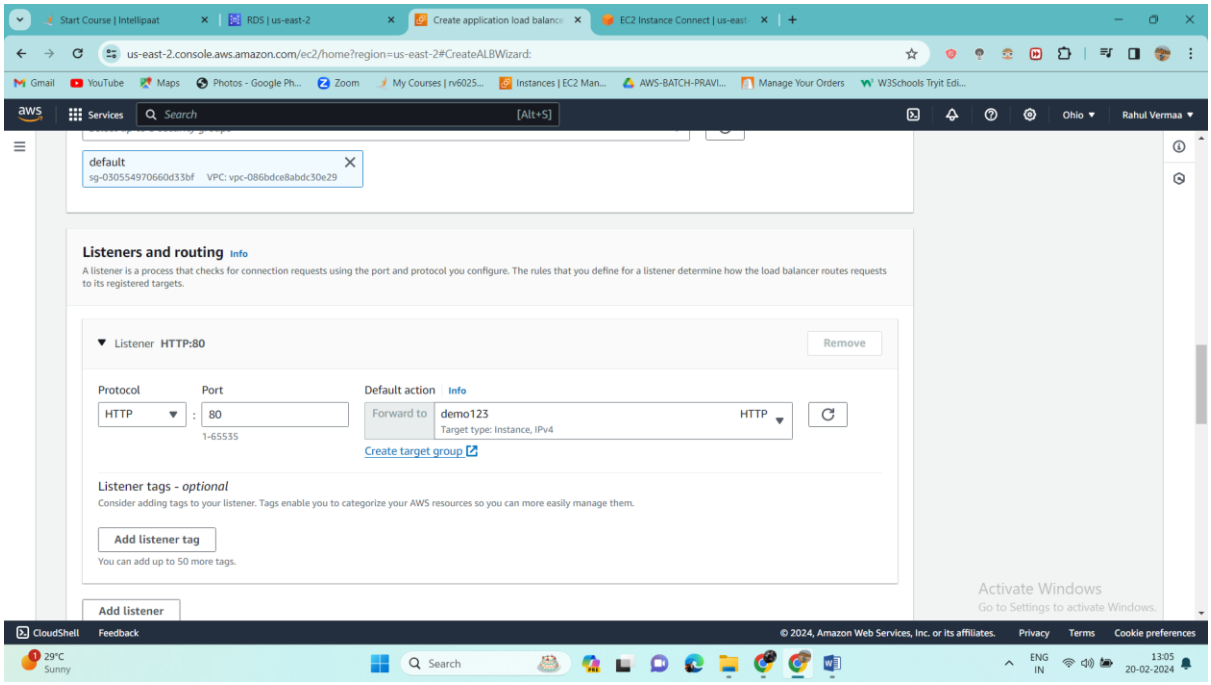
We have added our instance and our target group is created



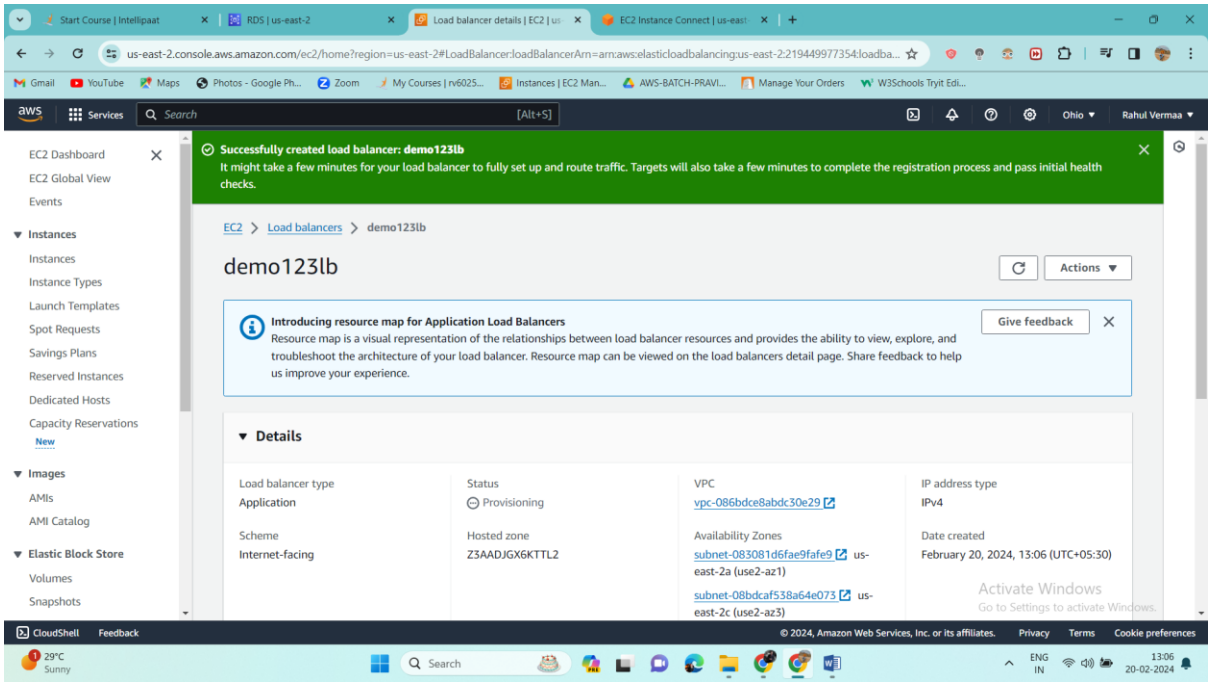
Now we can create our load balancer



Now everything default we just have to select our target group

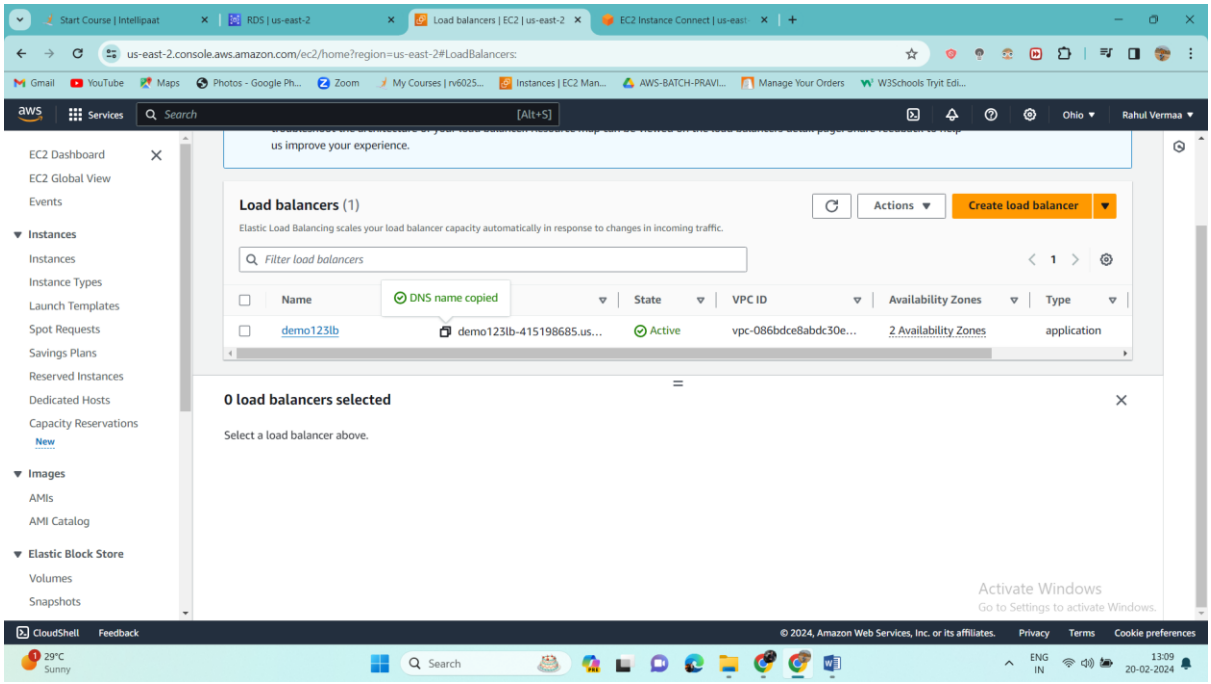


And our load balancer is also created

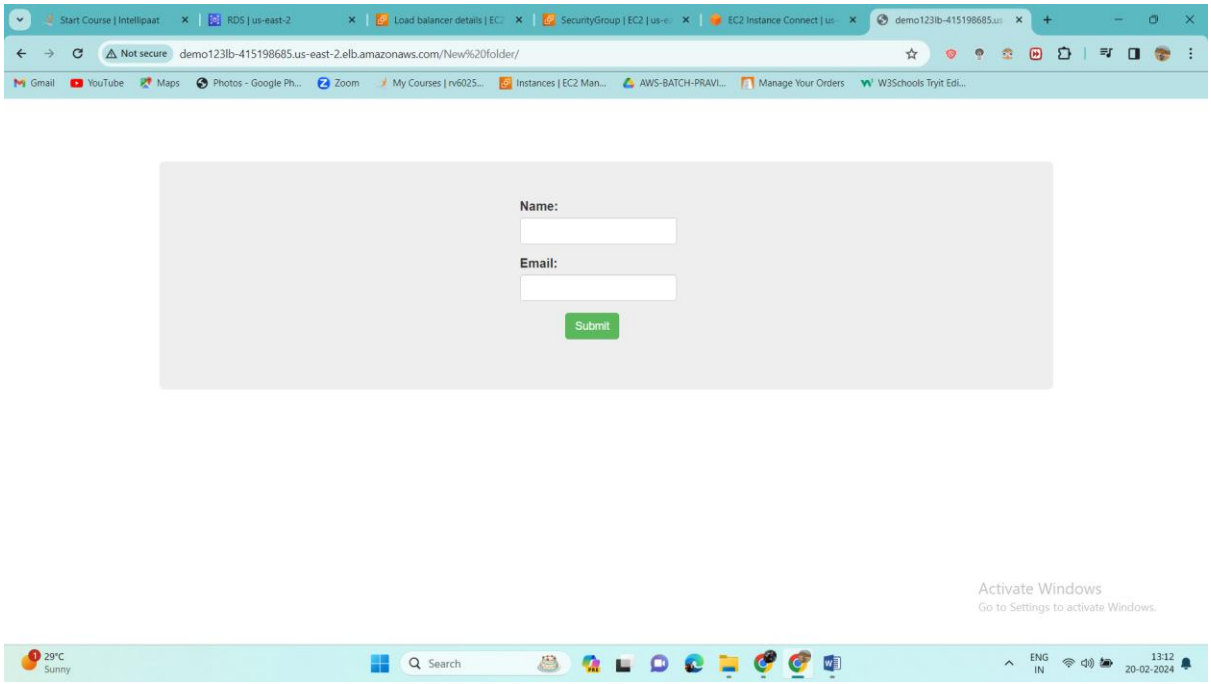


Now let’s check if it’s working fine let’s copy it’s DNS and paste it in browser

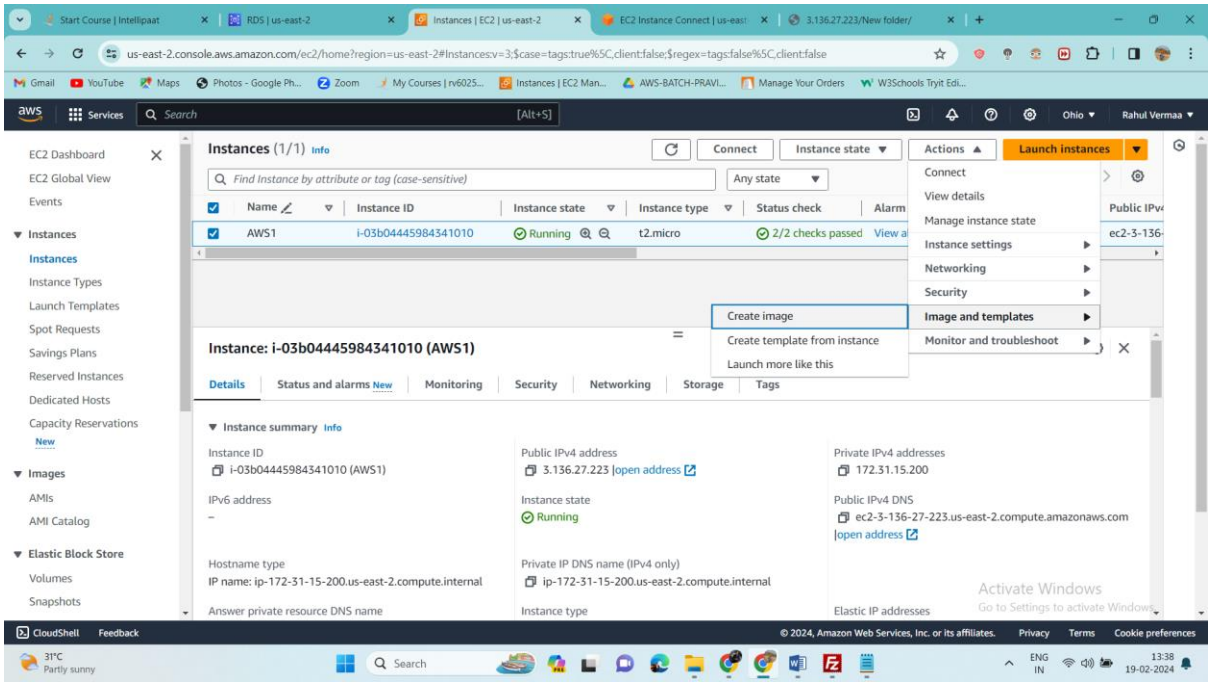
It should show our website



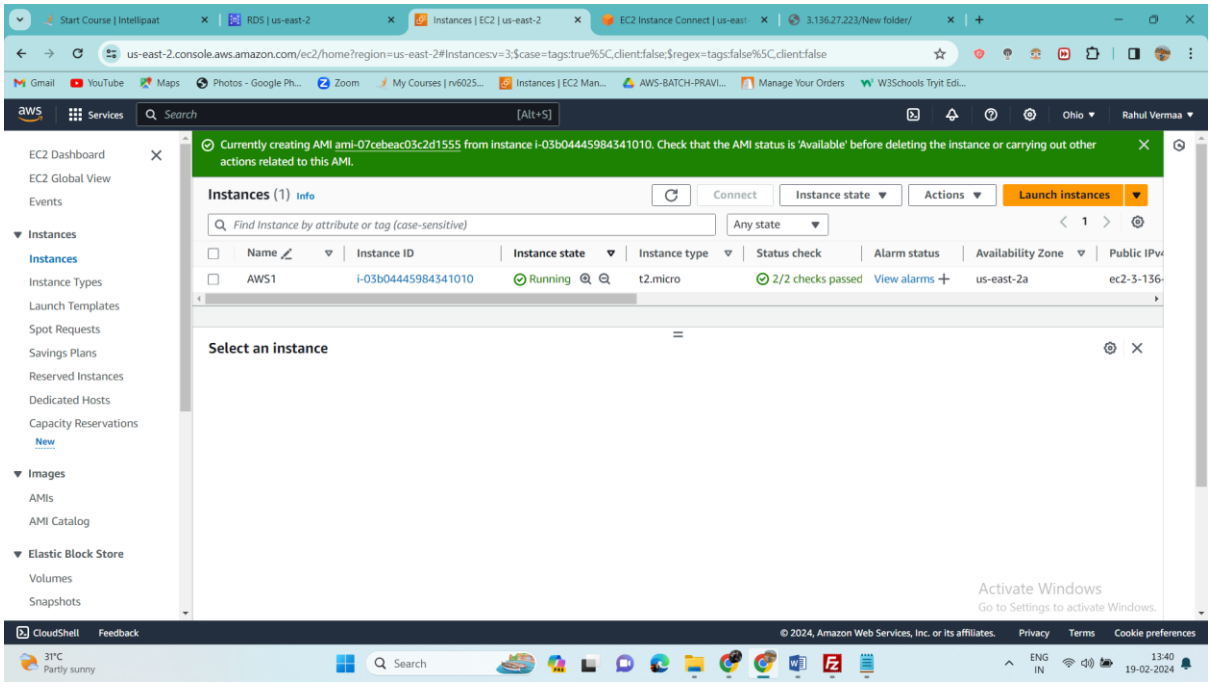
And it’s working properly



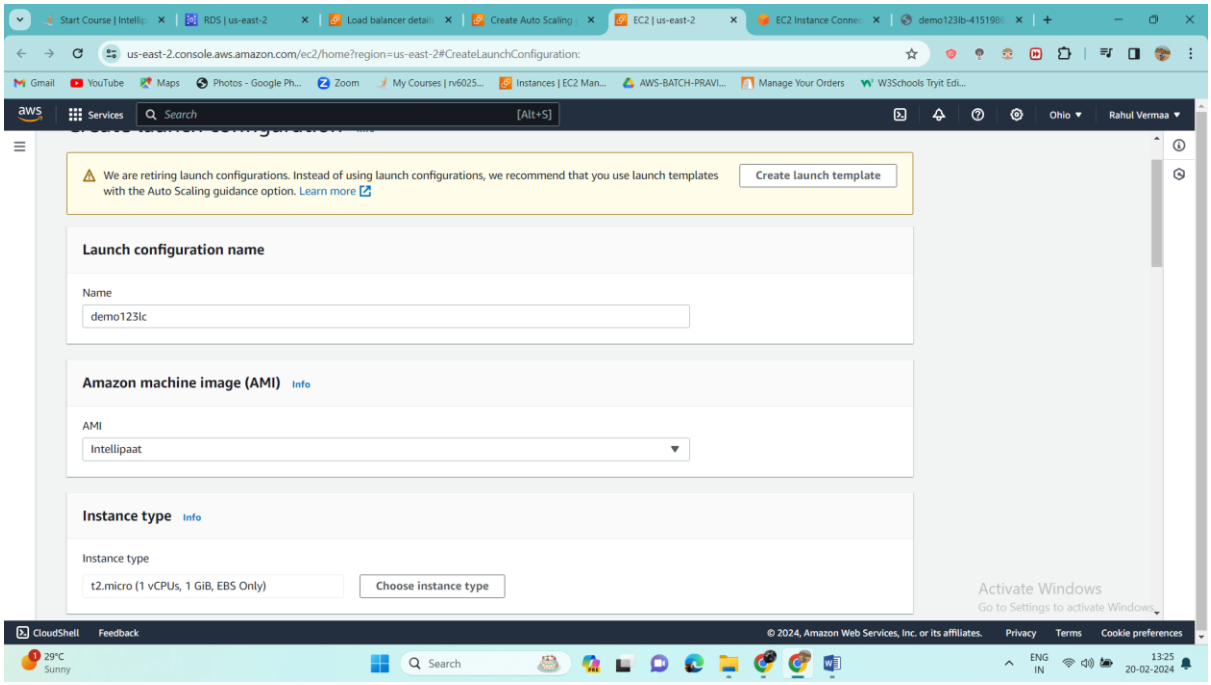
Now, we'll do the autoscaling of our website by going to our EC2 Instance and then click on Actions and Create Image



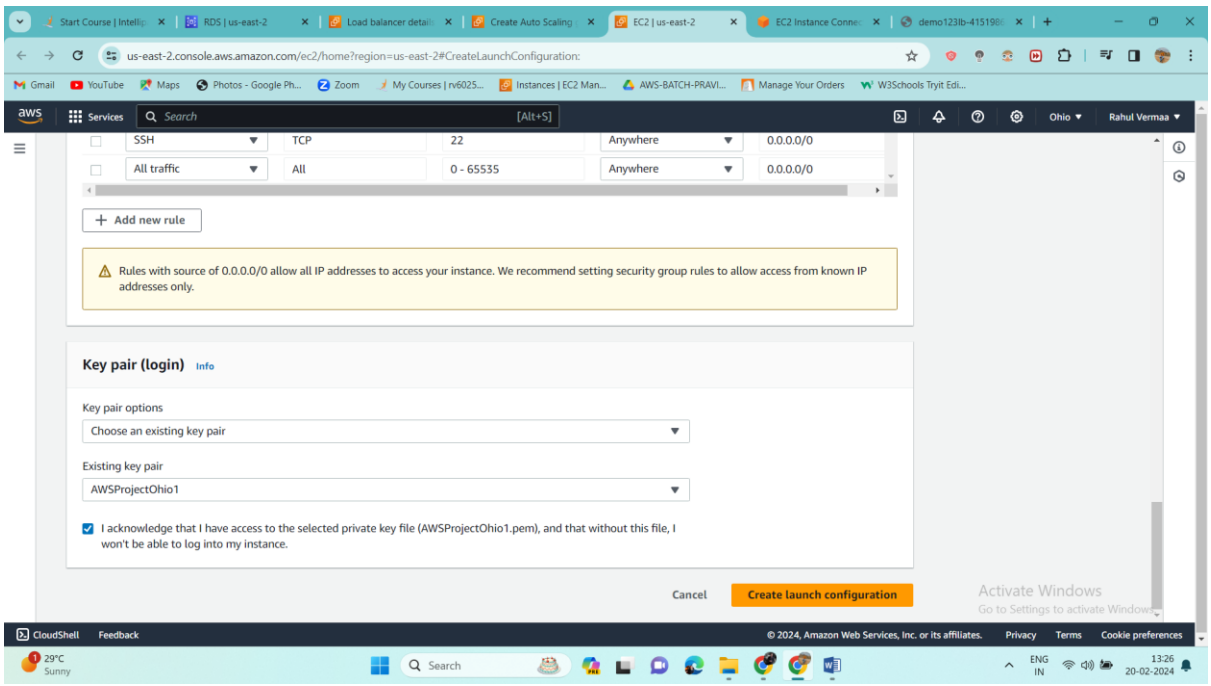
Our image is created



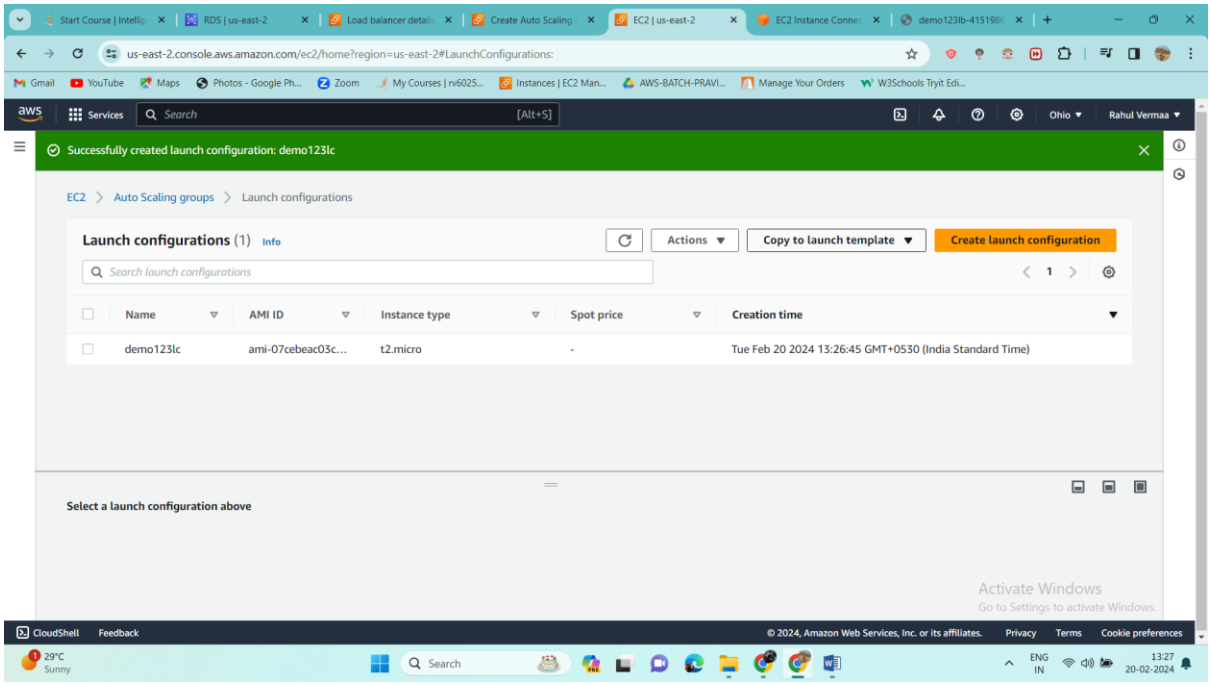
Now we will create launch configuration and will attach our AMI and we have selected instance type as t2. micro



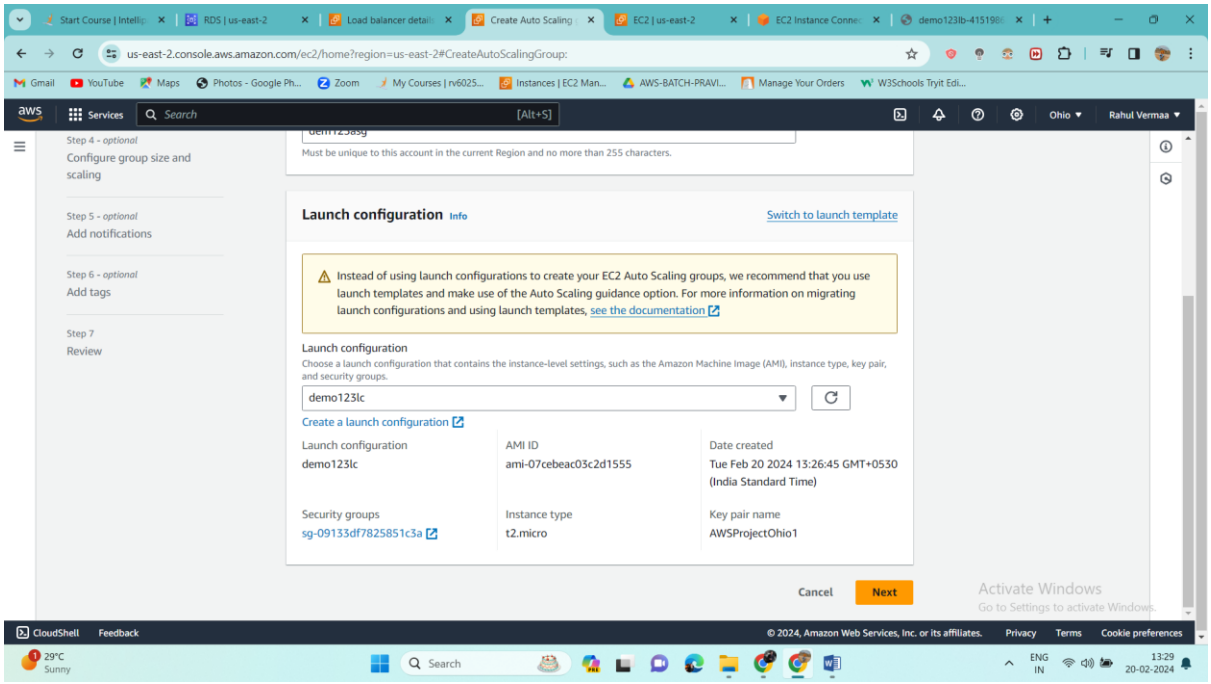
Will select key pair and create our launch configuration



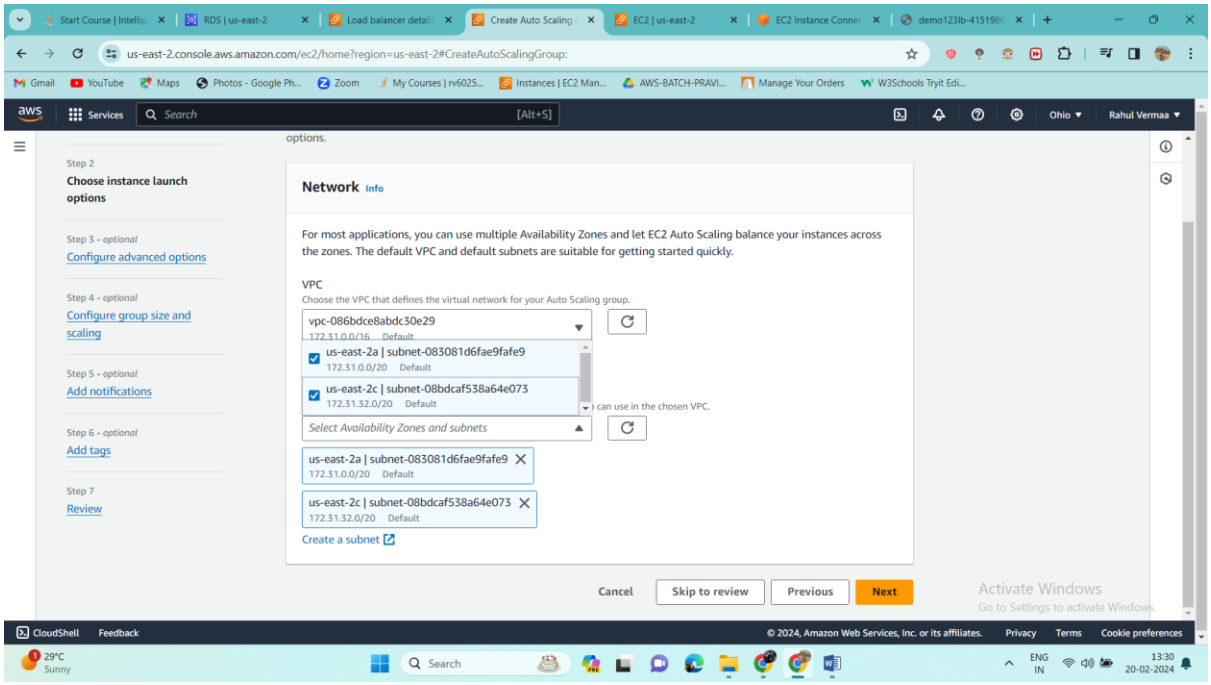
It’s created successfully



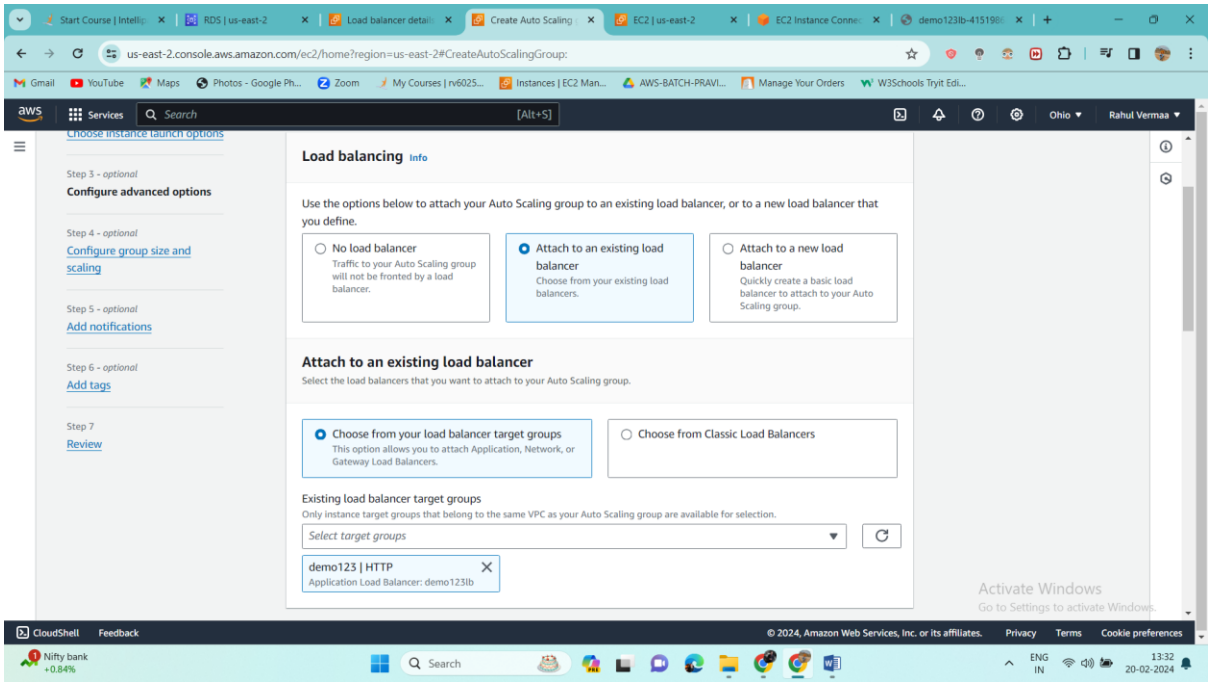
Now we create our autoscaling group



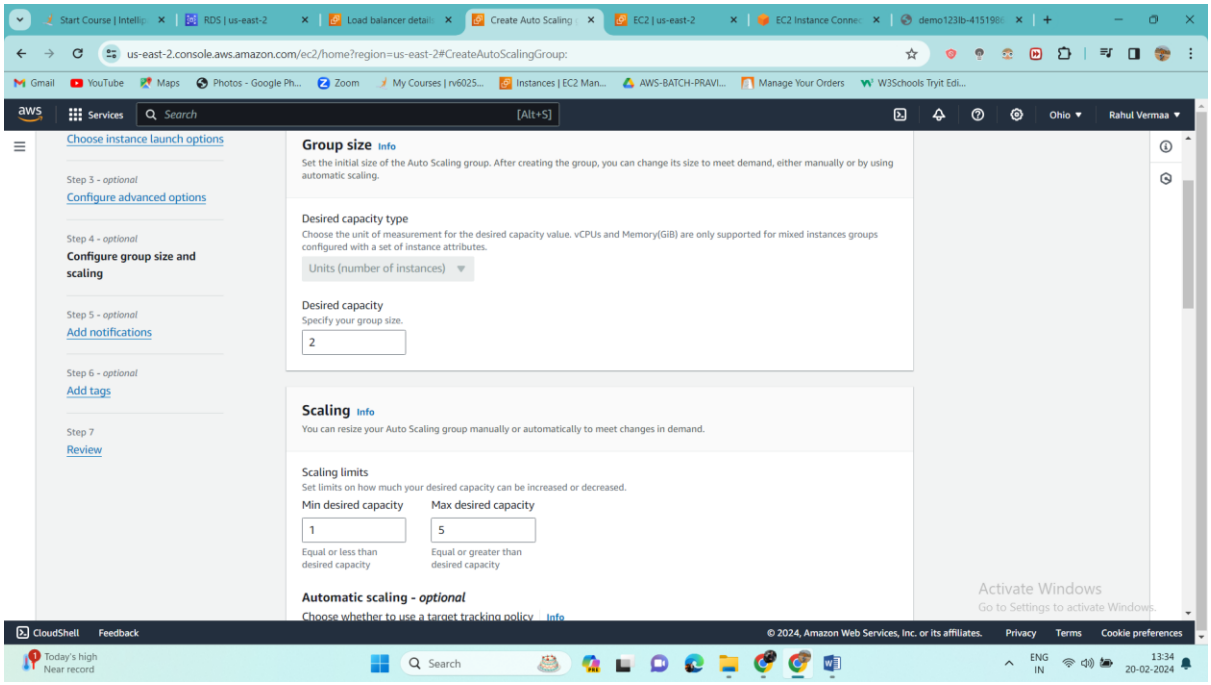
Select vpc and subnets



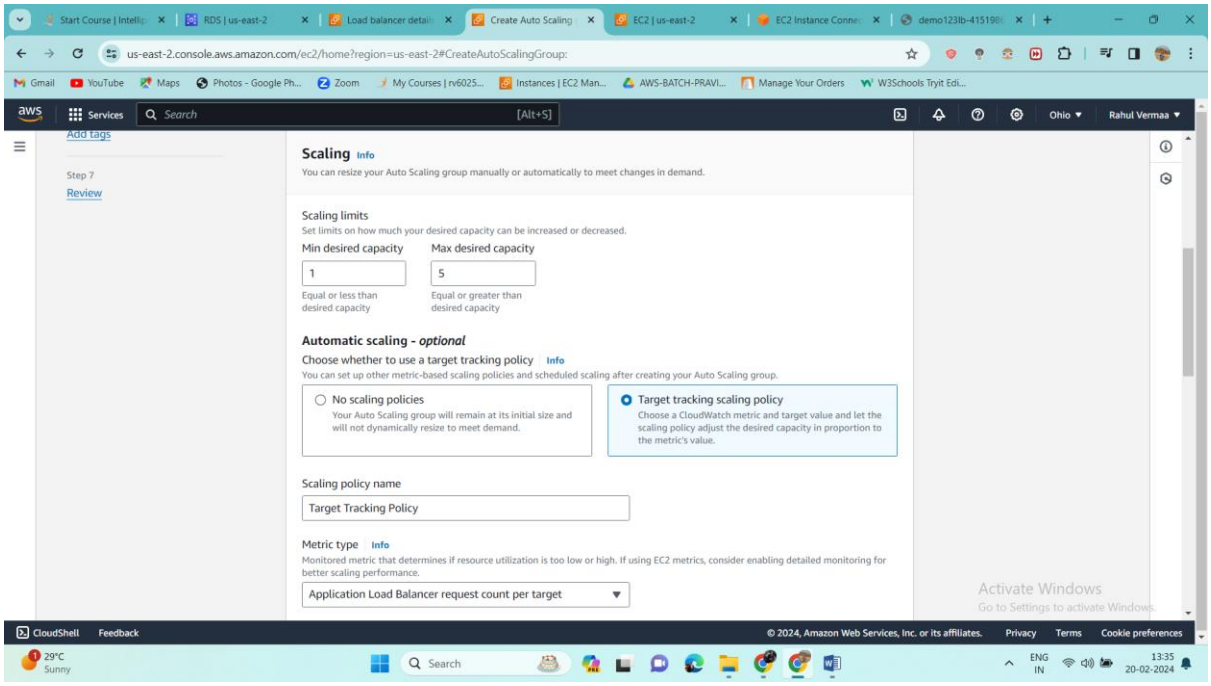
Now attach load balancer



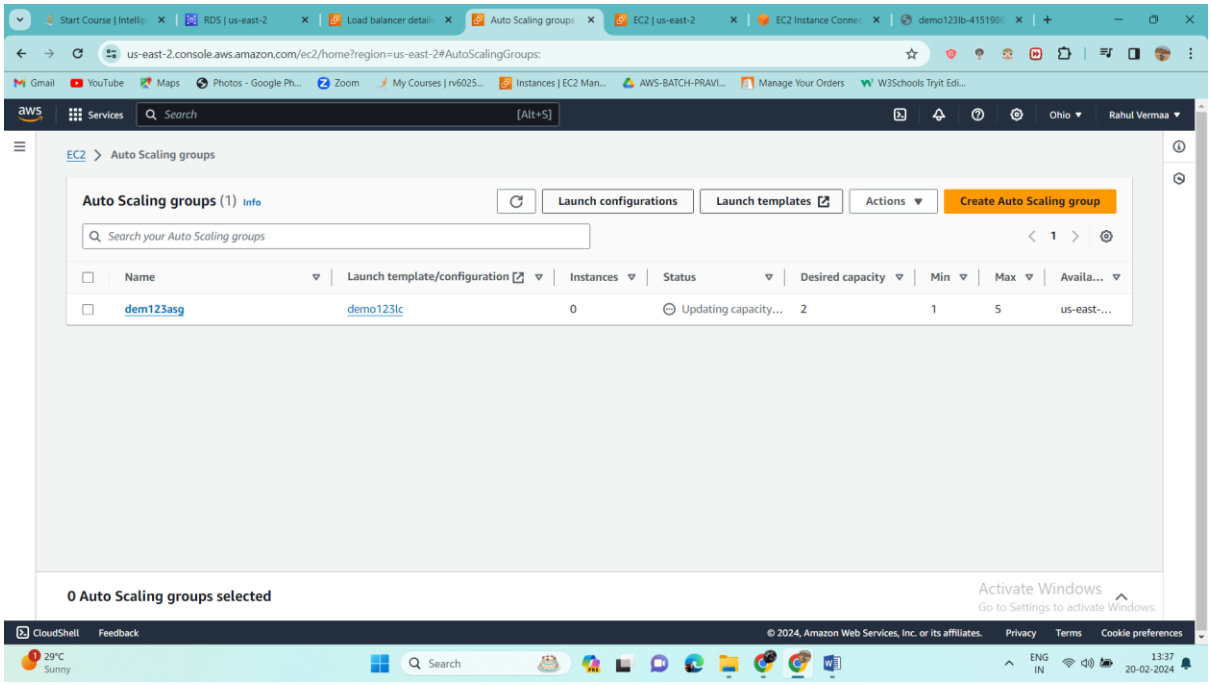
We have to give desired capacity requirements



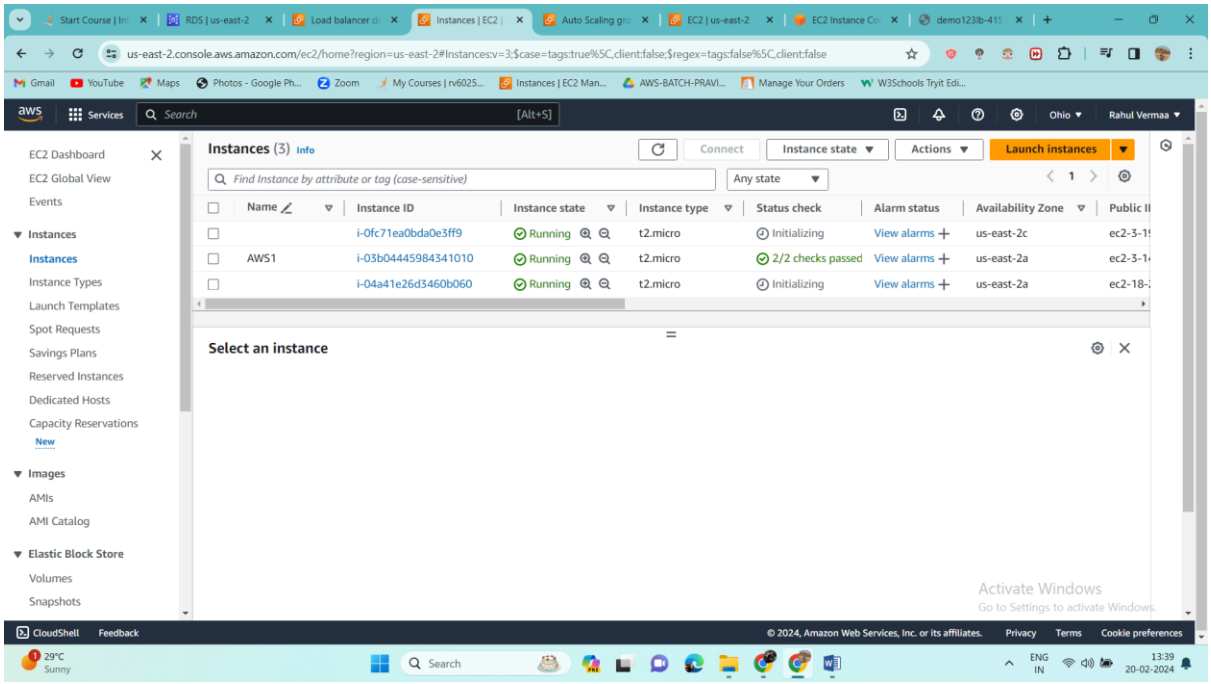
Scalling

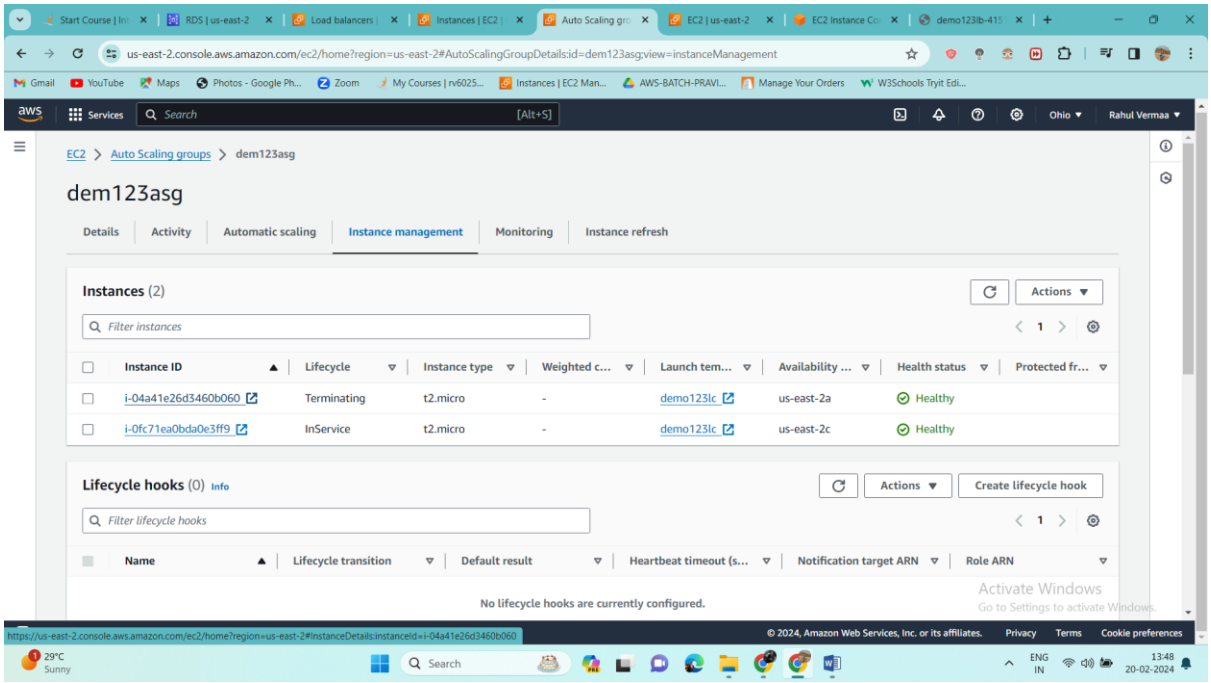


And it’s done



And our instances is created




us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#AutoScalingGroupDetails:id=dem123asg;view=instanceManagement
Services
Search
[Alt+S]
Ohio
Rahul Vermaa
EC2 > Auto Scaling groups > dem123asg
dem123asg
Details
Activity
Automatic scaling
Instance management
Monitoring
Instance refresh
Instances (2)
Actions
Filter instances
Instance ID
Lifecycle
Instance type
Weighted c...
Launch tem...
Availability ...
Health status
Protected fr...
i-04a41e26d3460b060
Terminating
t2.micro
demo123lc
us-east-2a
Healthy
i-0fc71ea0bda0e3ff9
InService
t2.micro
demo123lc
us-east-2c
Healthy
Lifecycle hooks (0) Info
Actions
Create lifecycle hook
Filter lifecycle hooks
Name
Lifecycle transition
Default result
Heartbeat timeout (s...
Notification target ARN
Role ARN
No lifecycle hooks are currently configured.
Activate Windows
Go to Settings to activate Windows.
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
29°C
Sunny
Search
ENG
IN
13:48
20-02-2024