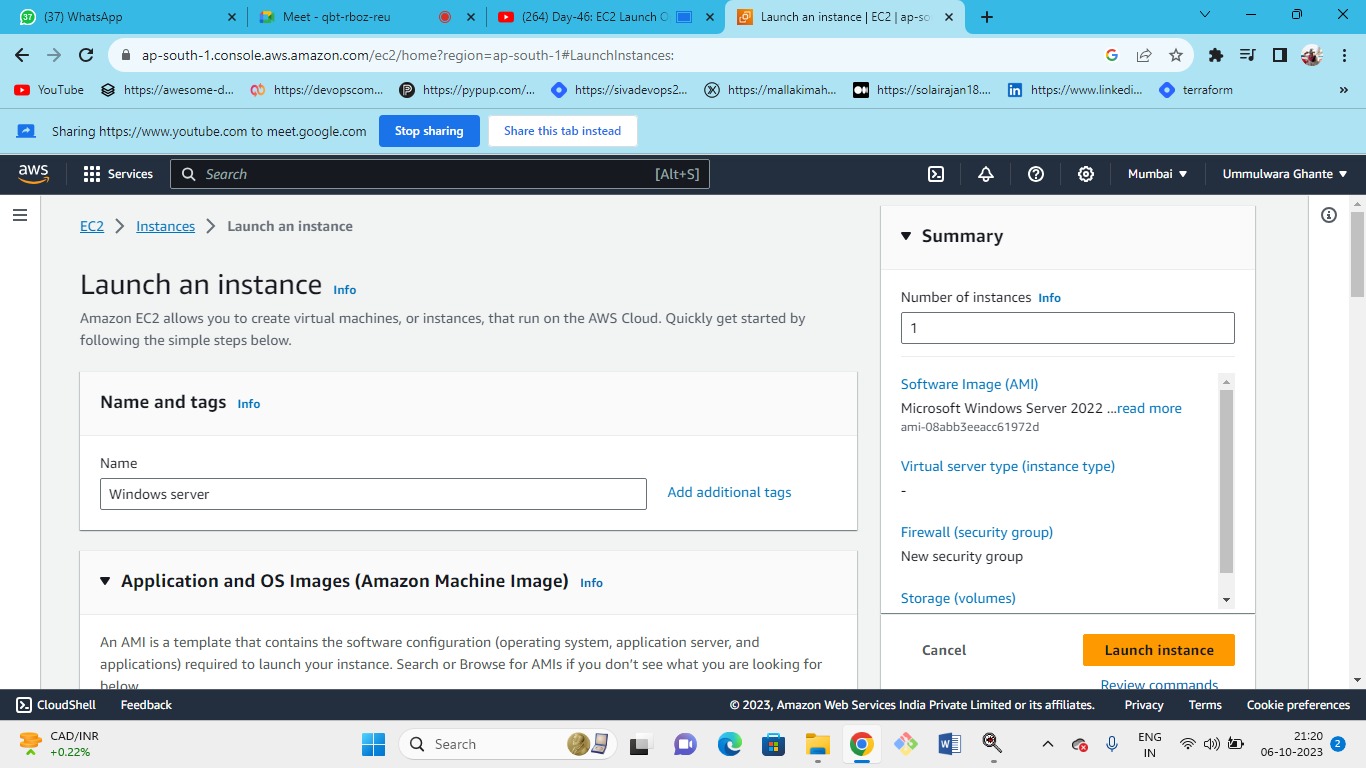
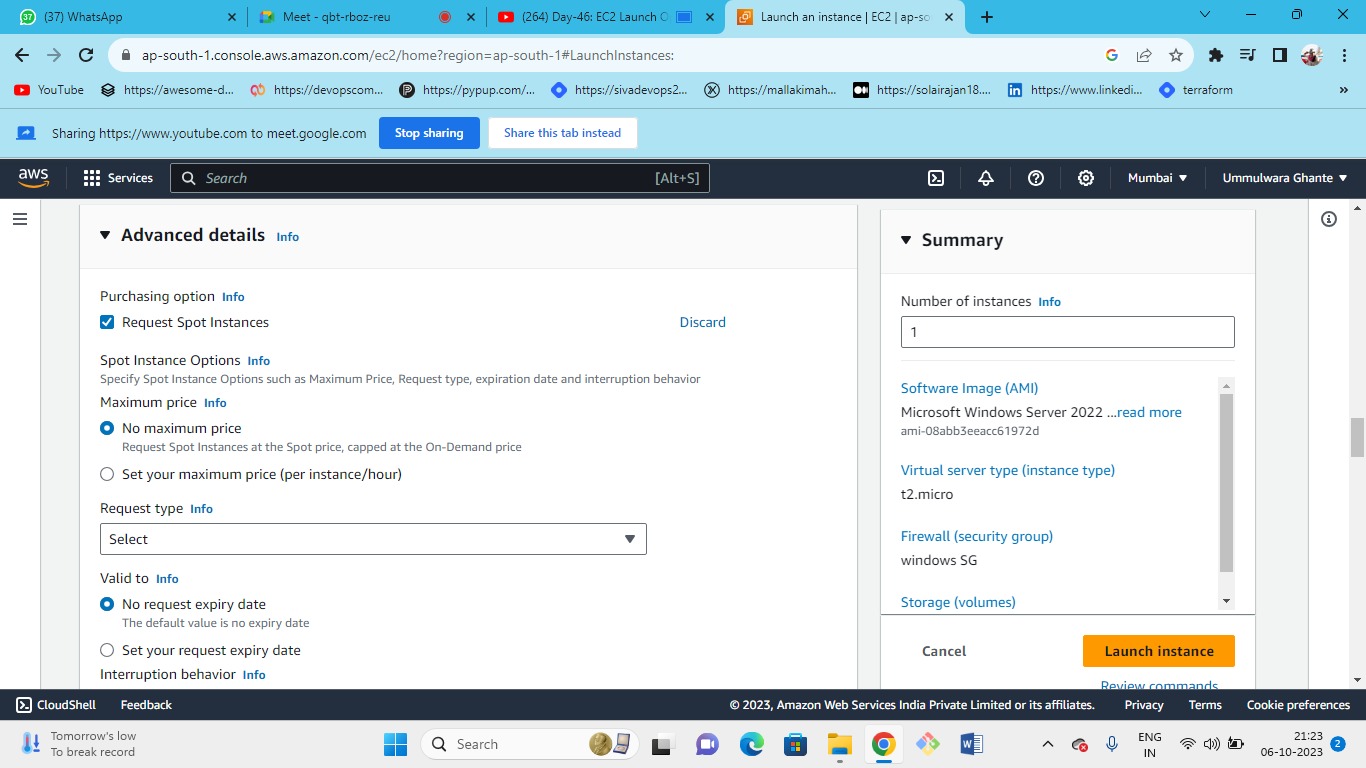
# Spot instance





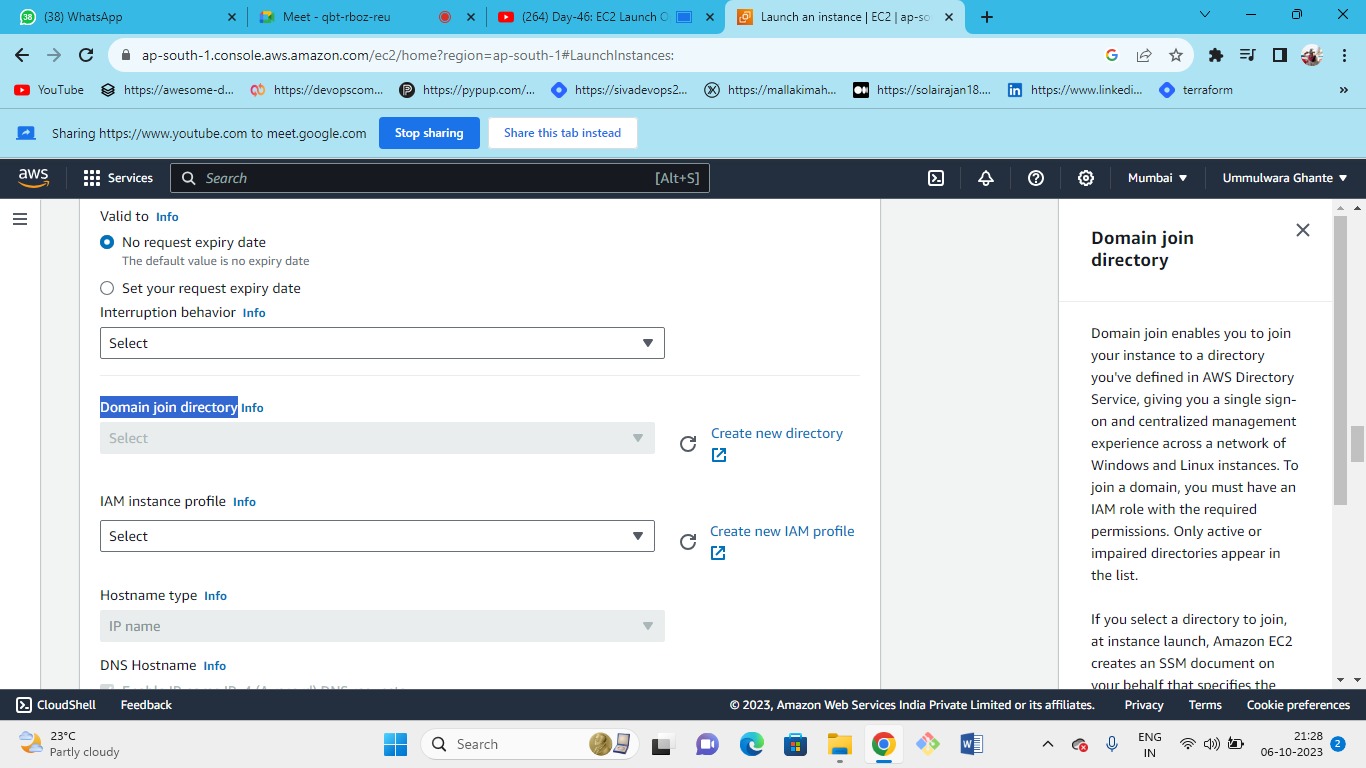
Select the windows server which has free tier-->instance type as t2.micro--->Select an existing SG ( in that select RDP as source my IP) ----> existing keypair or create new one





Request the spot instance at spot price or you can set the max price limit of your own so that if it crosses the spot price then you will not purchase the instance

ex: if spot price is 0.01 and you set limit till 0.45 later if the spot cross to 0.46 then you will not purchase the instance.



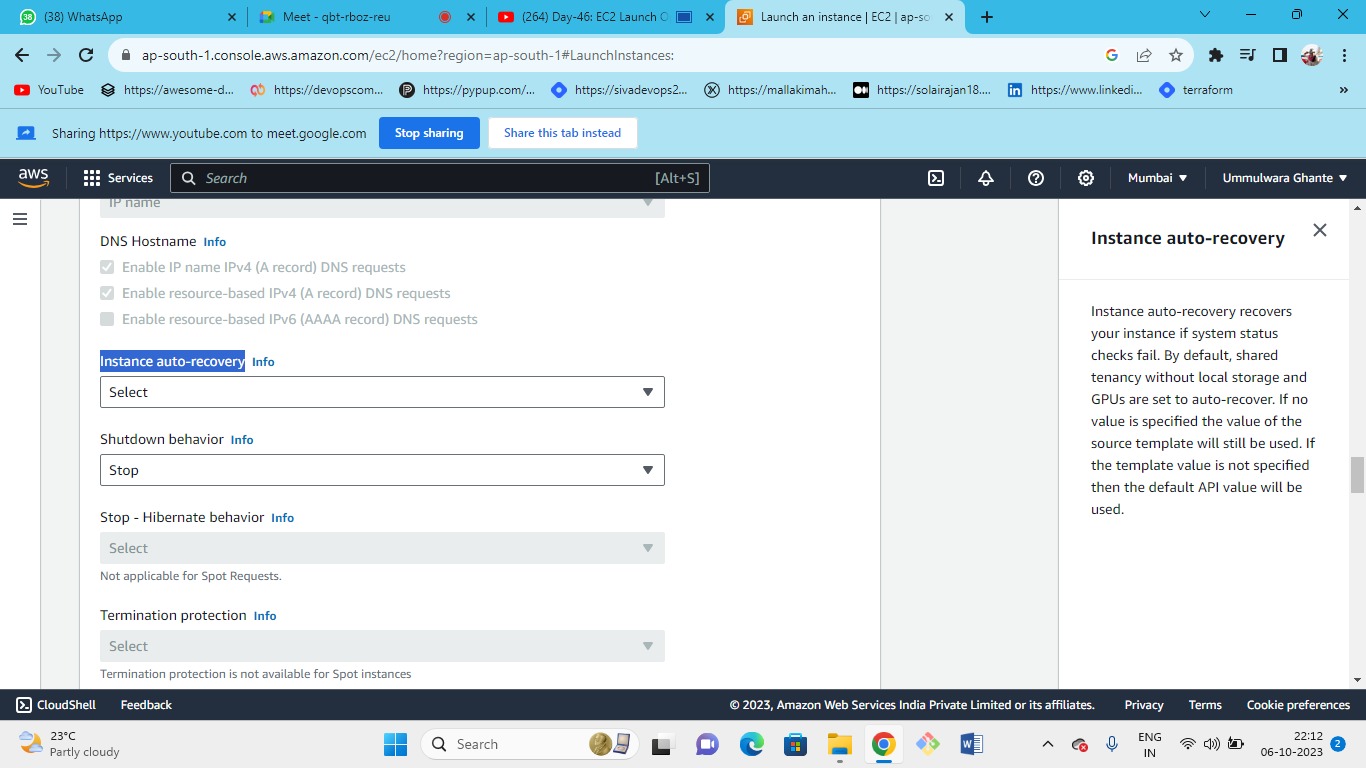


* If you created a domain then you can join it
* even you can create a new directory service
* if someone is using AD with the server then someone can login into the server by using their own user and password.by using a domain username and password.
* domain is the central location where you create a user, machines such as laptops, on-premises machines personal computers all these machines are given to the developer and administrator all will be there in the active directory and manage the permission that is active directory.
* You can manage the servers from active directory
* In an active directory, you can create a user very similar to the IAM and it is a Microsoft big tool.

**IAM Instance profile-**

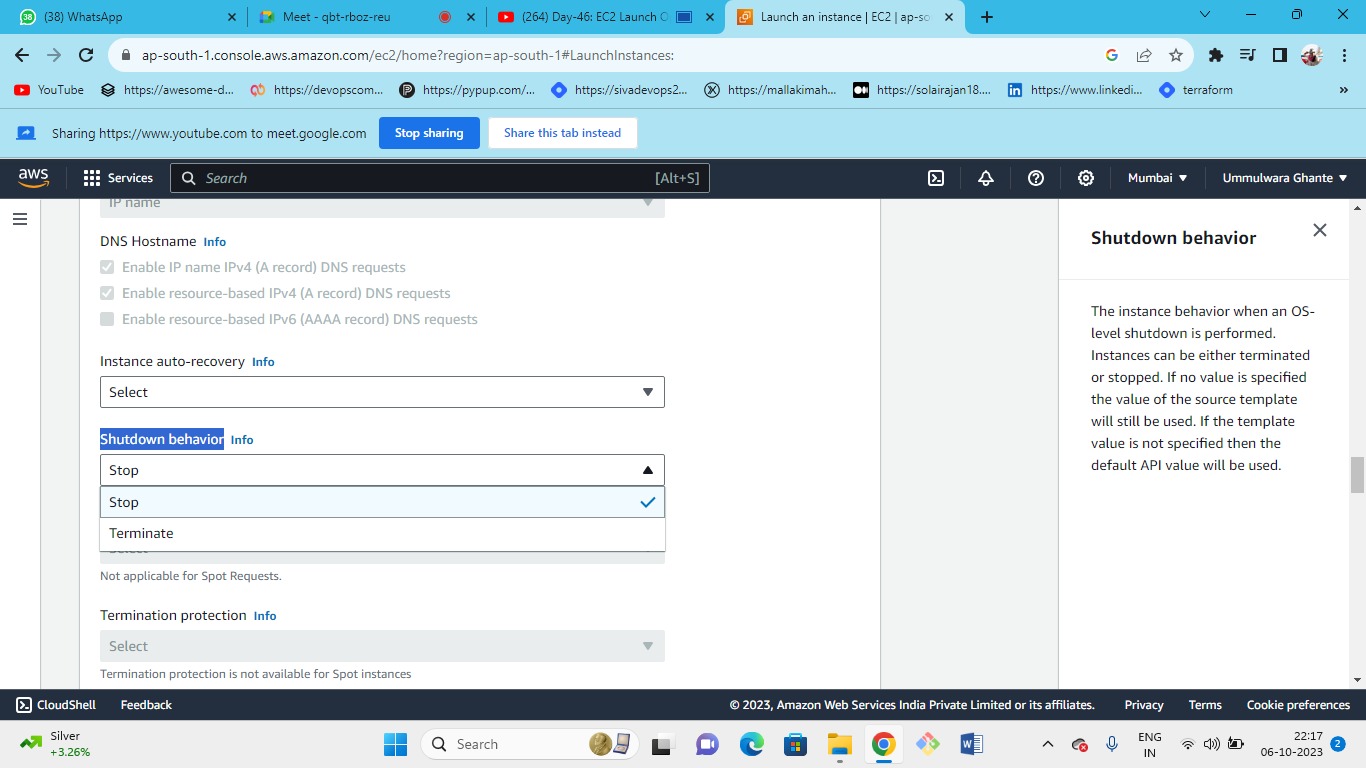
what is the profile of the ec2 instance what permissions can we give to ec2 through roles?

* Role Is nothing but a profile that profile becomes like a role
* If we need to attach permission to the ec2 instance then we need to attach the instance IAM Instance Profile.

`

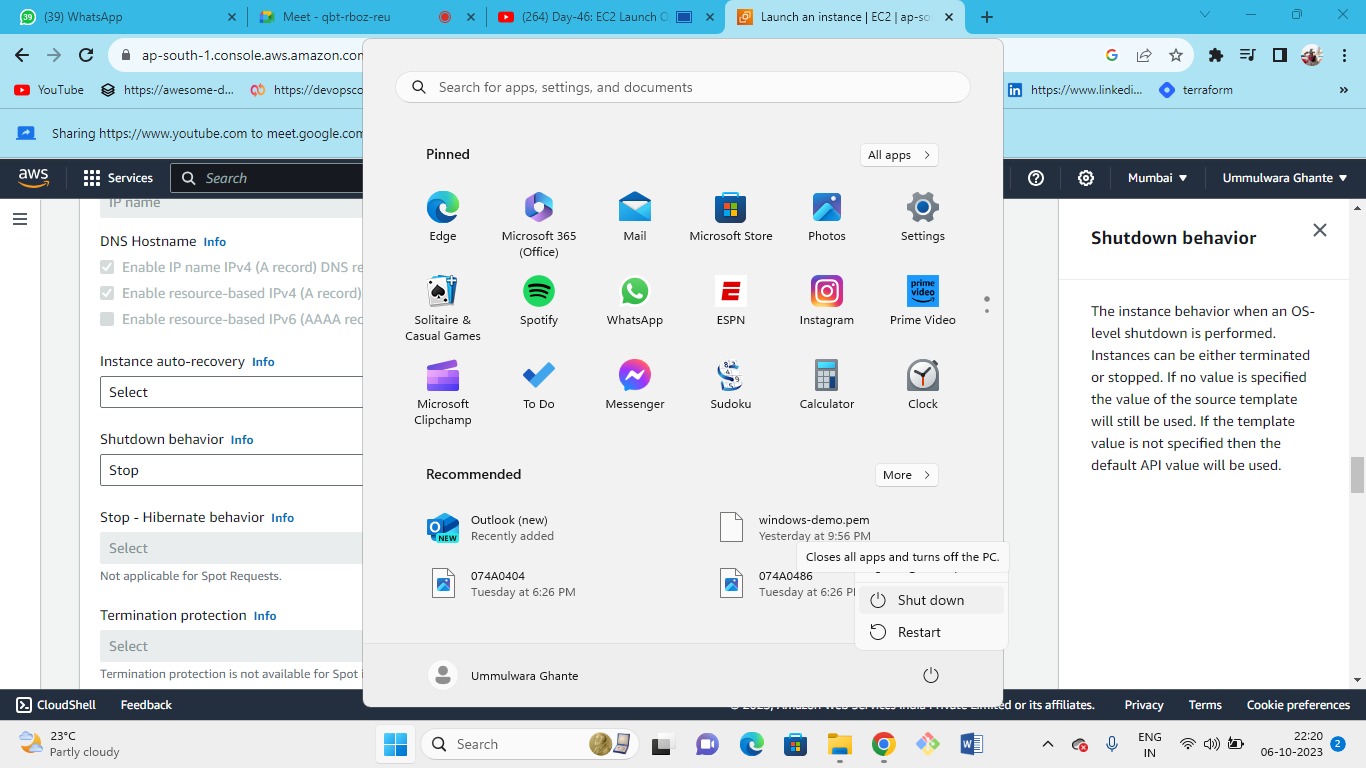


If the system status check fails then the automatically will be replaced by the aws.



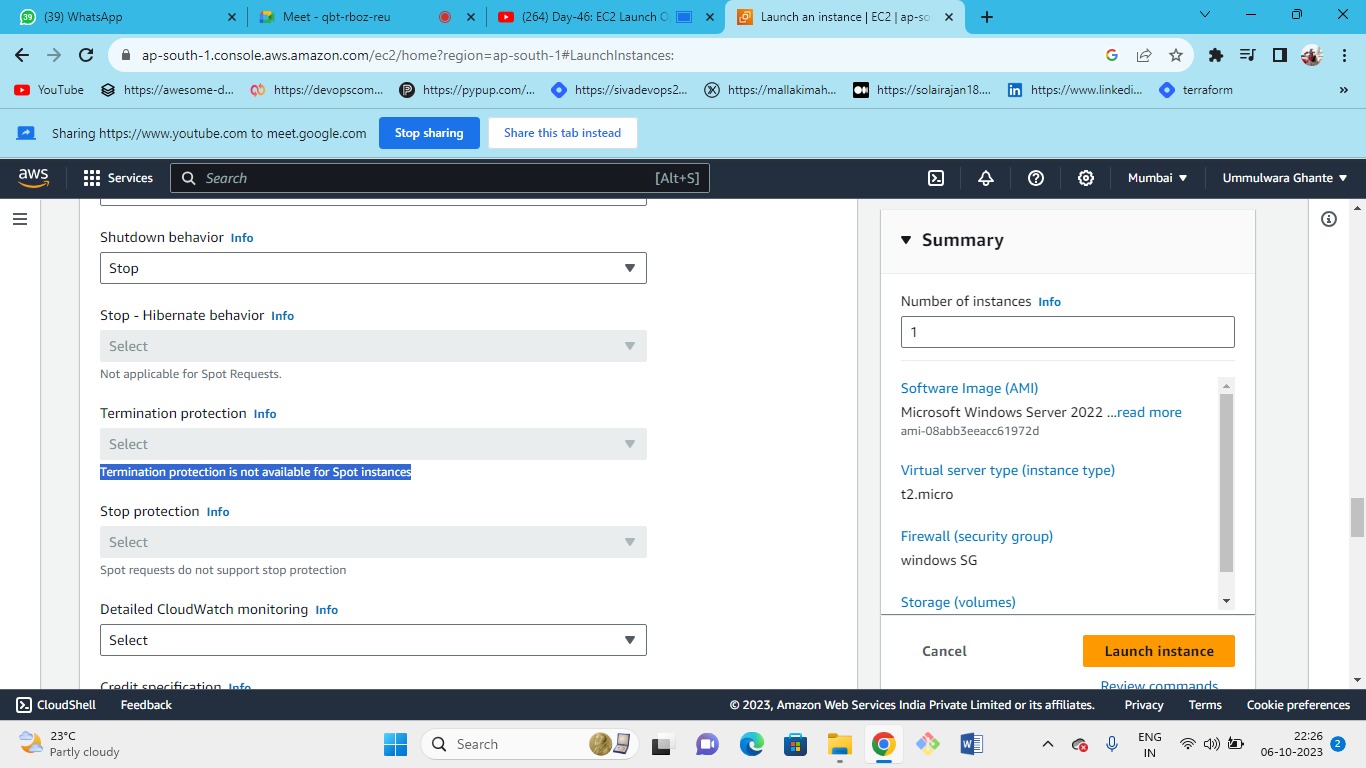


By setting the shutdown behaviour your server will either stop or will get terminated if someone shutdown your PC



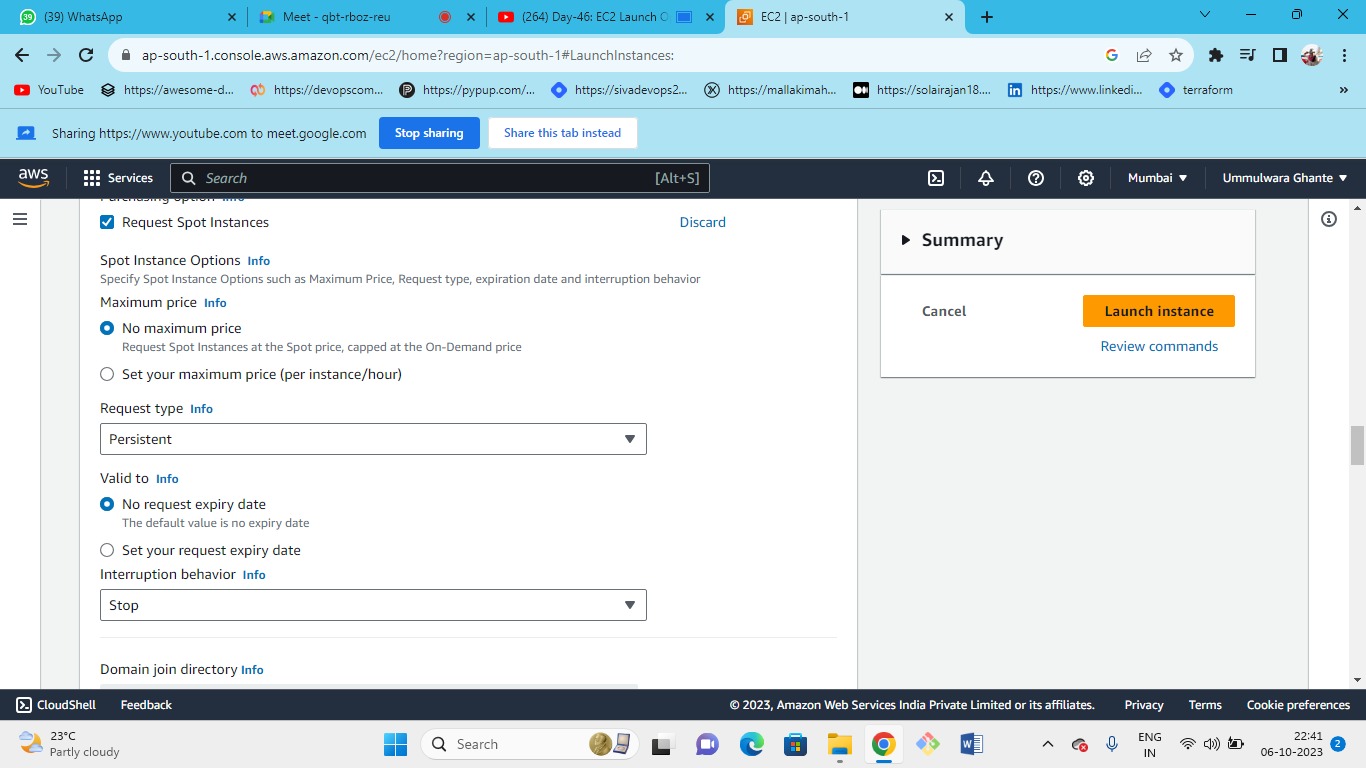


By default, shutdown behaviour is stopped.

* If someone shutdown with the OS level (if someone shutdowns your PC or laptop) server will shut down and if the shutdown behaviour is set to terminate then automatically the instance will get terminated. 



This is because aws can terminate the spot instance at any time depending on the fluctuating spot price so the termination protection is off.

Request Type:



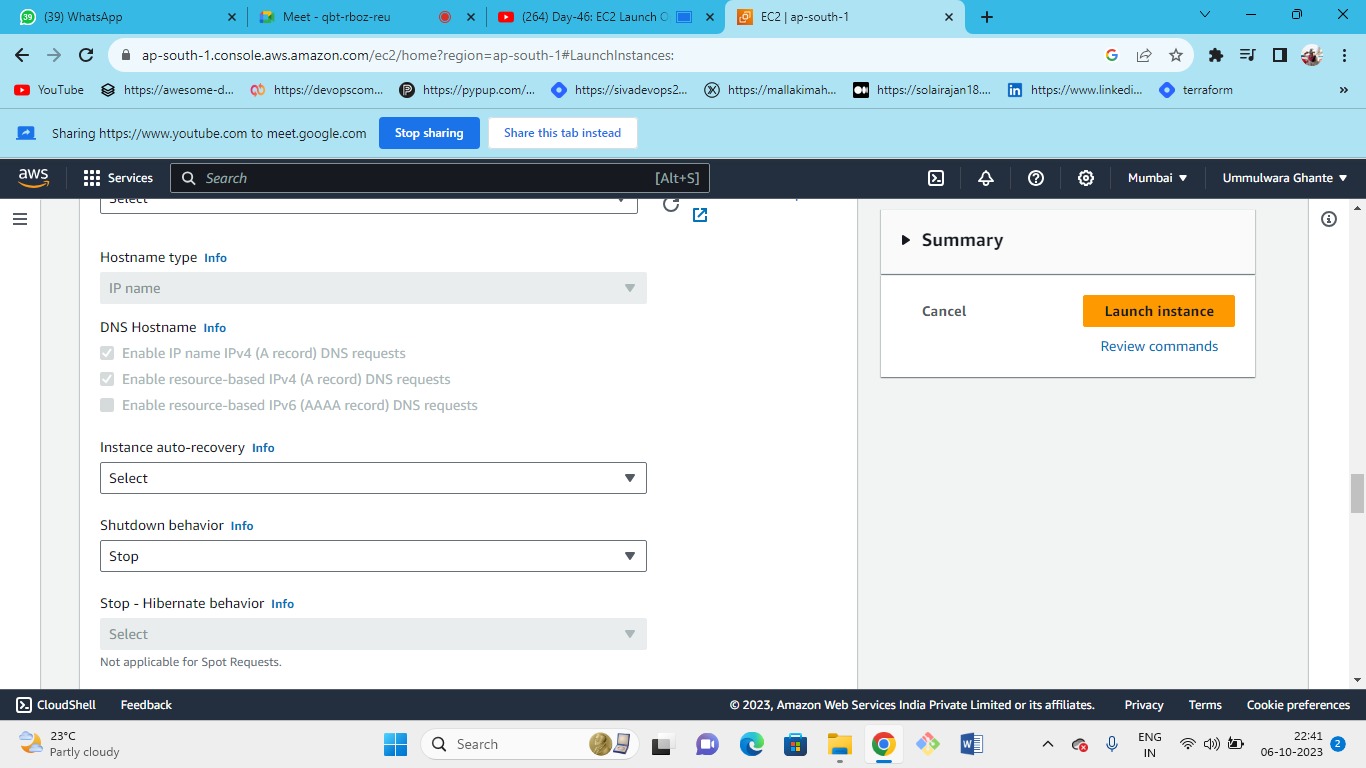
One Time 🡪 if capacity terminated it will not be relaunched again

Persistent 🡪 if capacity terminated it will be launched again

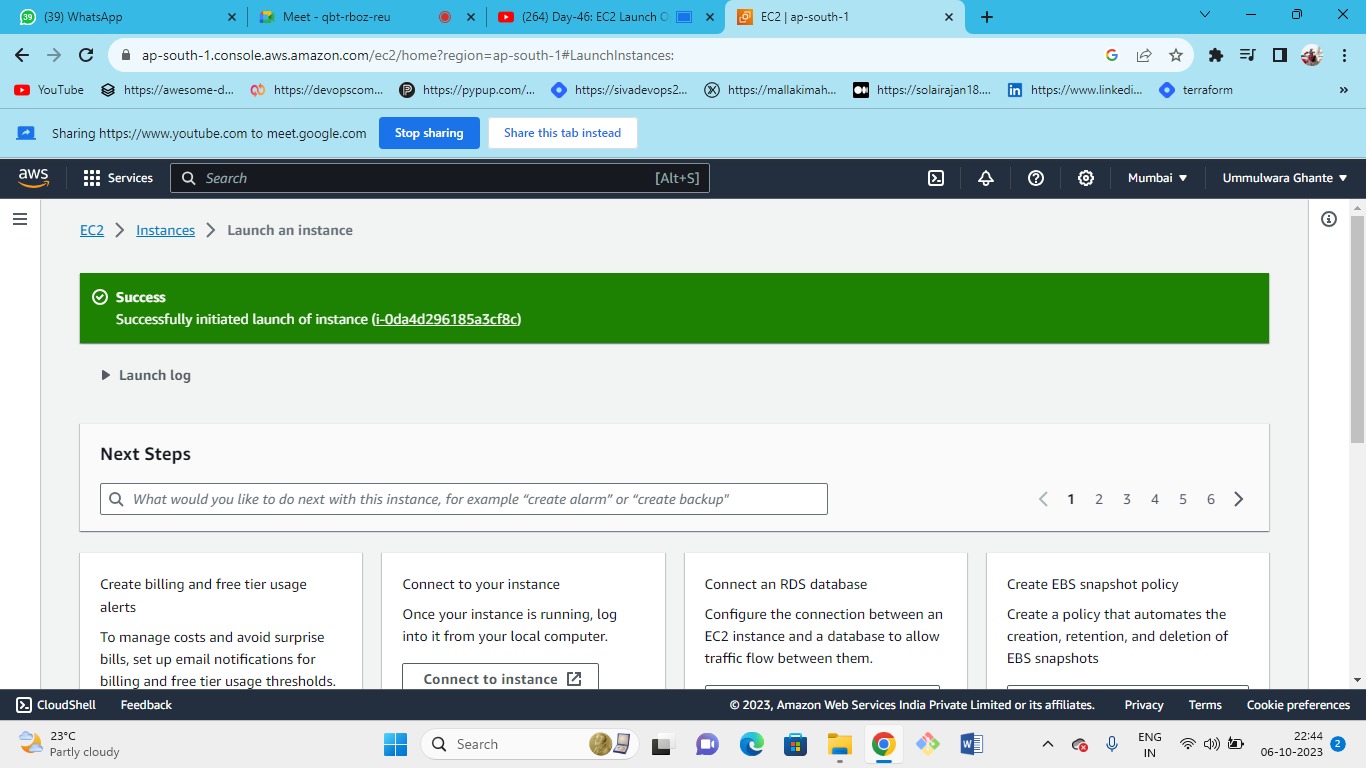
We can set our expiry date or we can keep it as default.

Interruption Behaviour:

* It is associated with the spot instance
* Means 🡪 if the spot price increases than set limit then AWS will terminate/stop the instance by giving 2Min notification
* Using one time request 🡪 then interruption behaviour is always terminated.
* Using Persistent request 🡪 then interruption behaviour can be selected as stop or terminated.



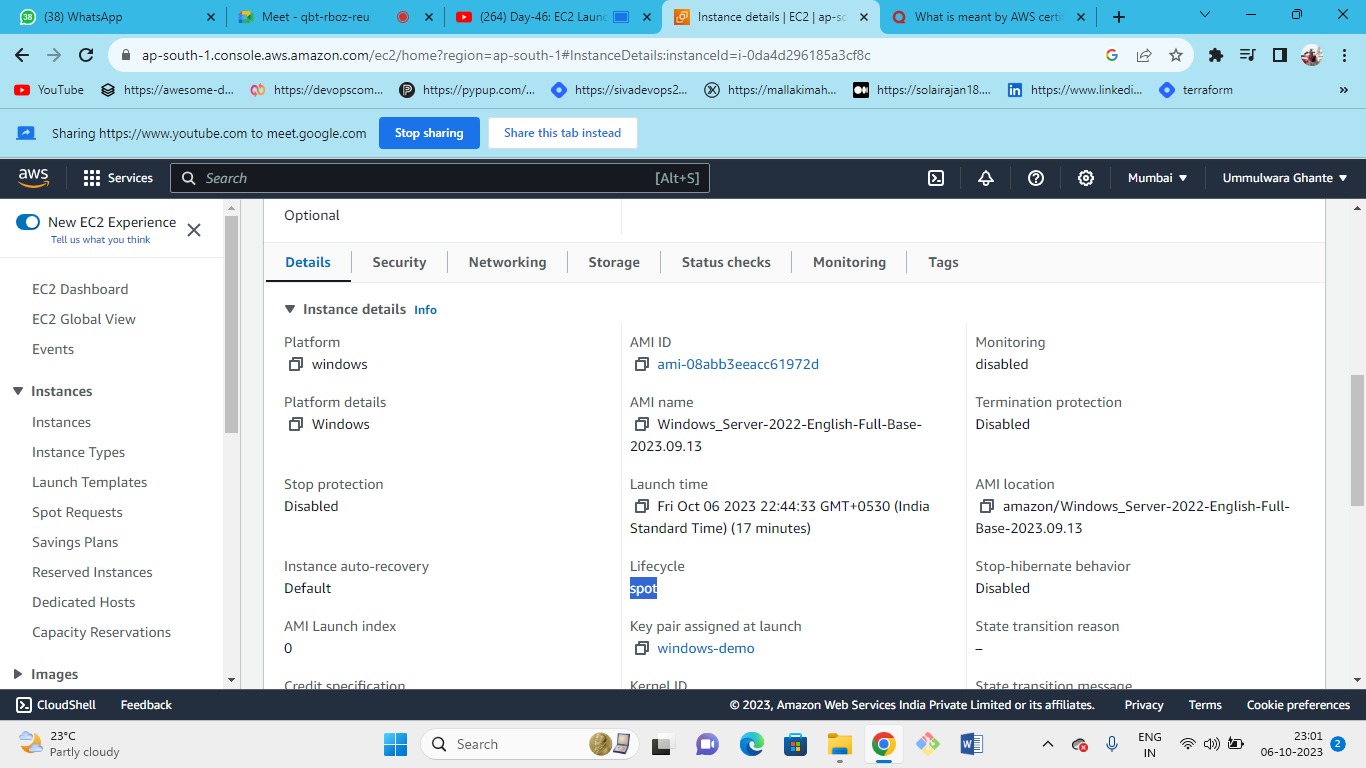






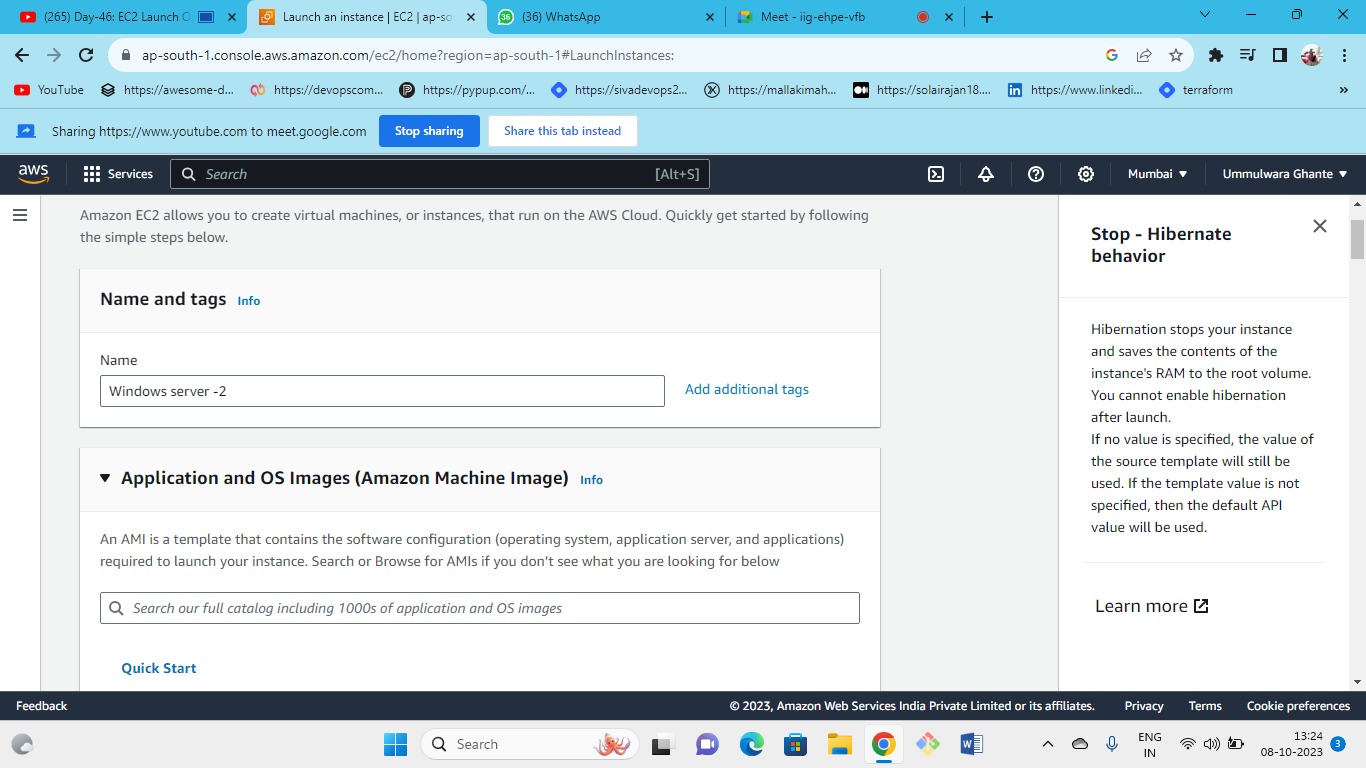
Note - Shutdown behaviour and interruption behaviour should be same.

When you select the spot request then stop-hibernate behaviour, termination protection, stop protection are all disabled.

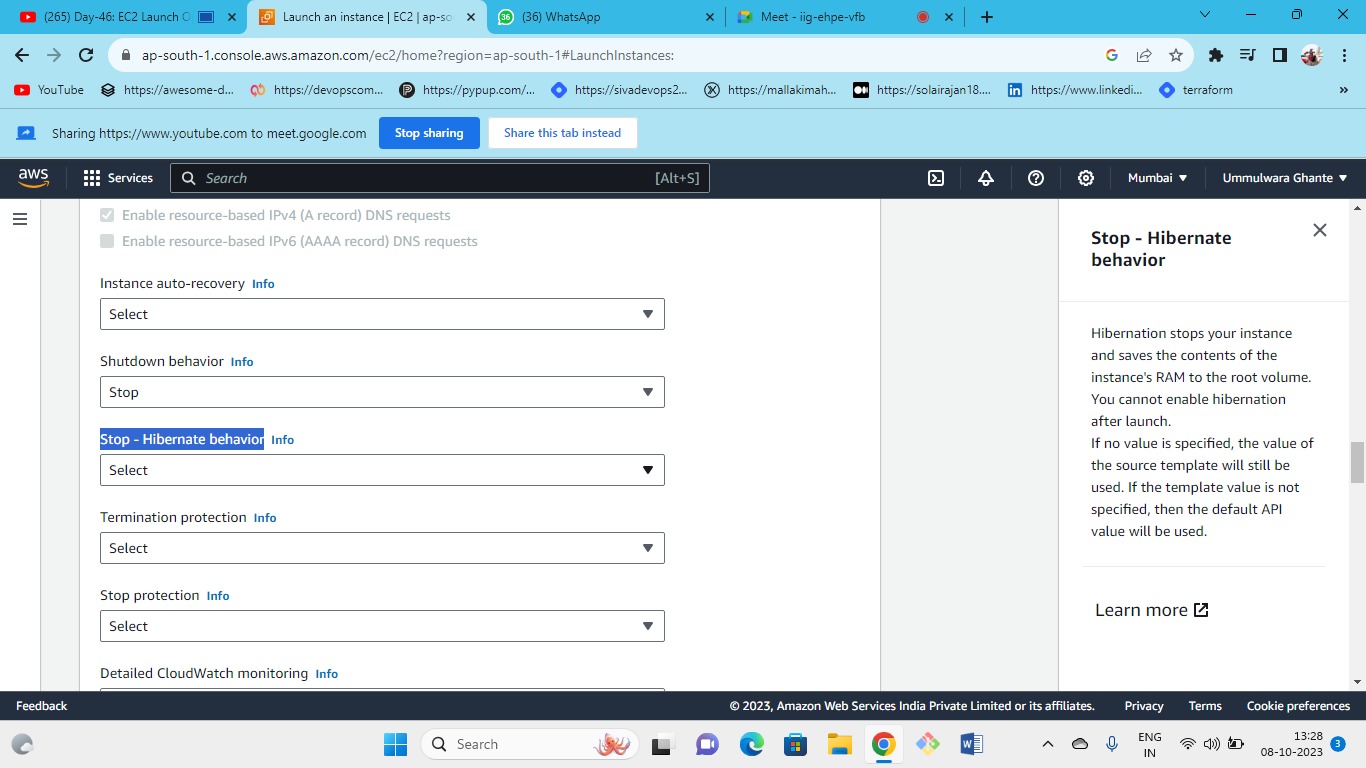




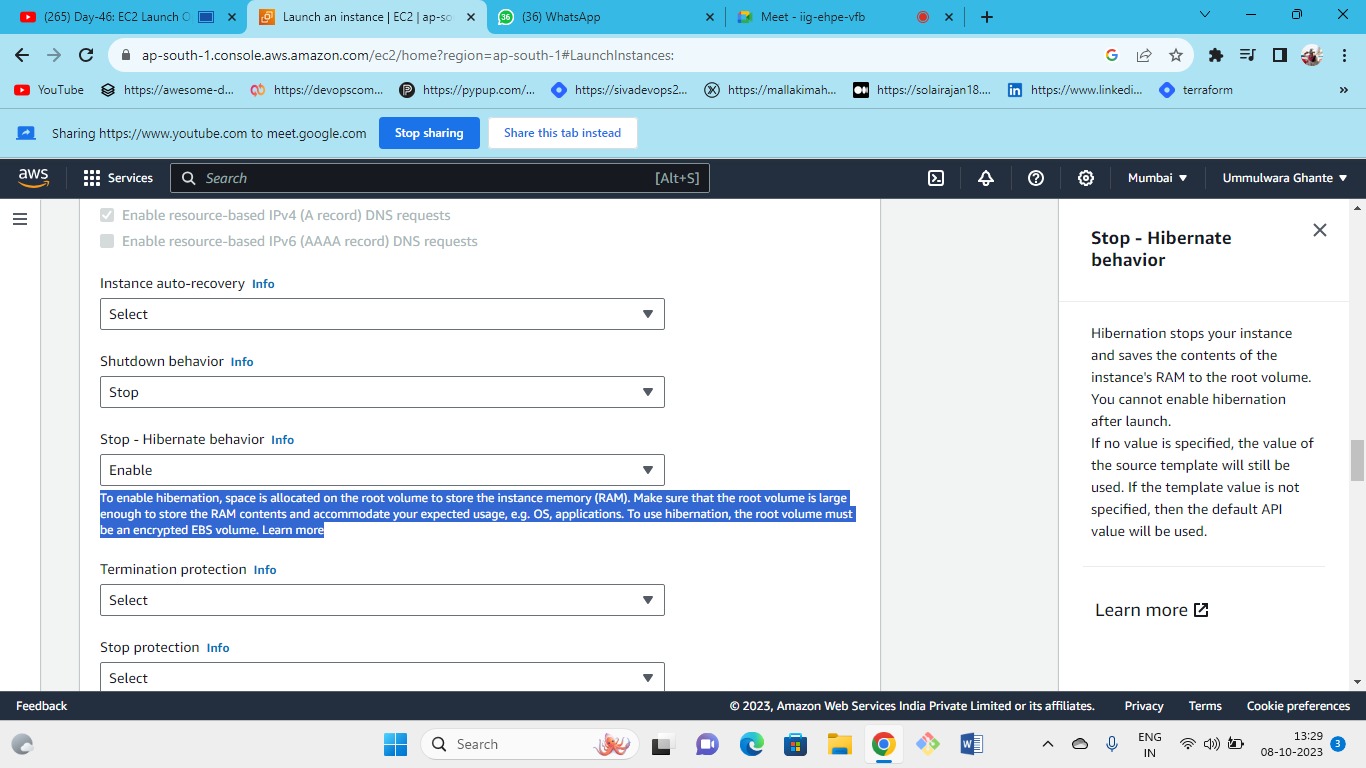
The instance that we have just launched for the Windows server is the spot instance.



OS (windows)--->key pair(old or can create new)---> instance type(t2.micro)--->security group(existing as RDP in that source my IP)



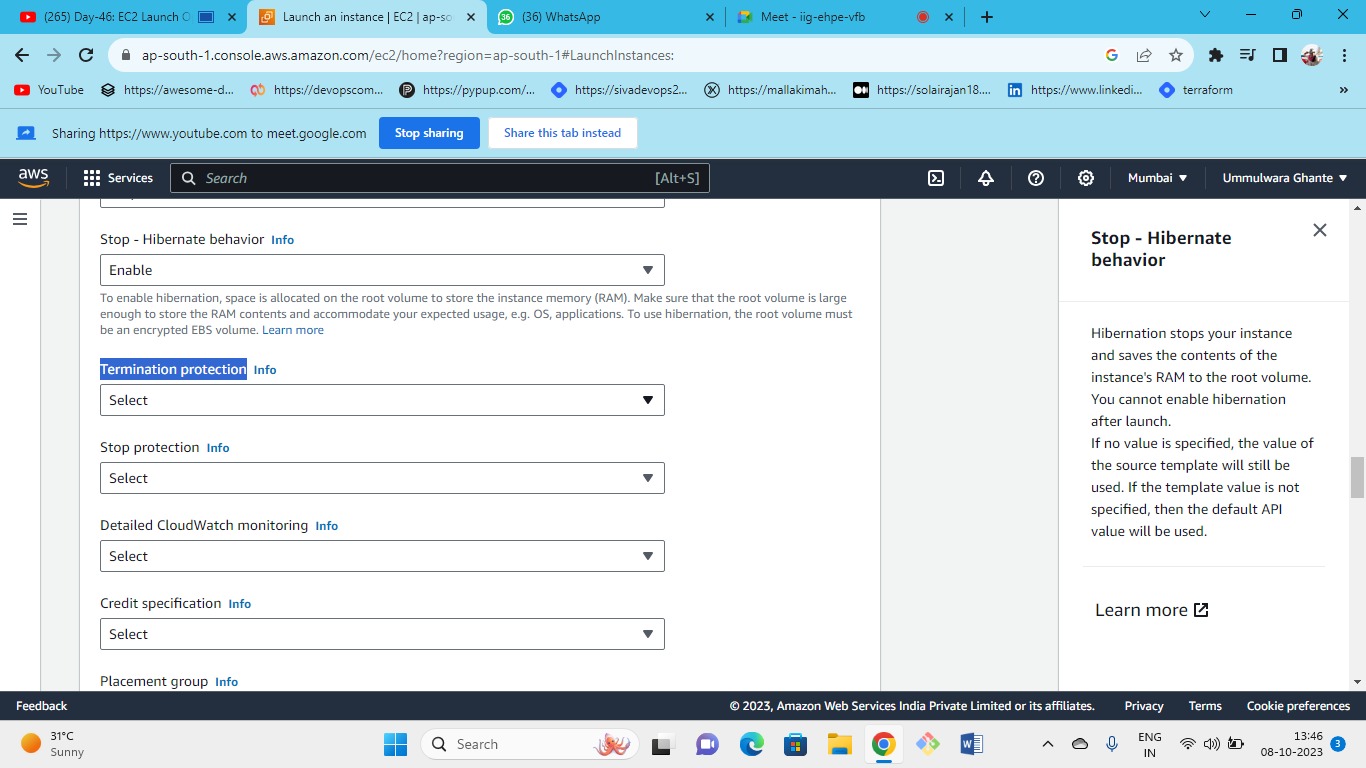




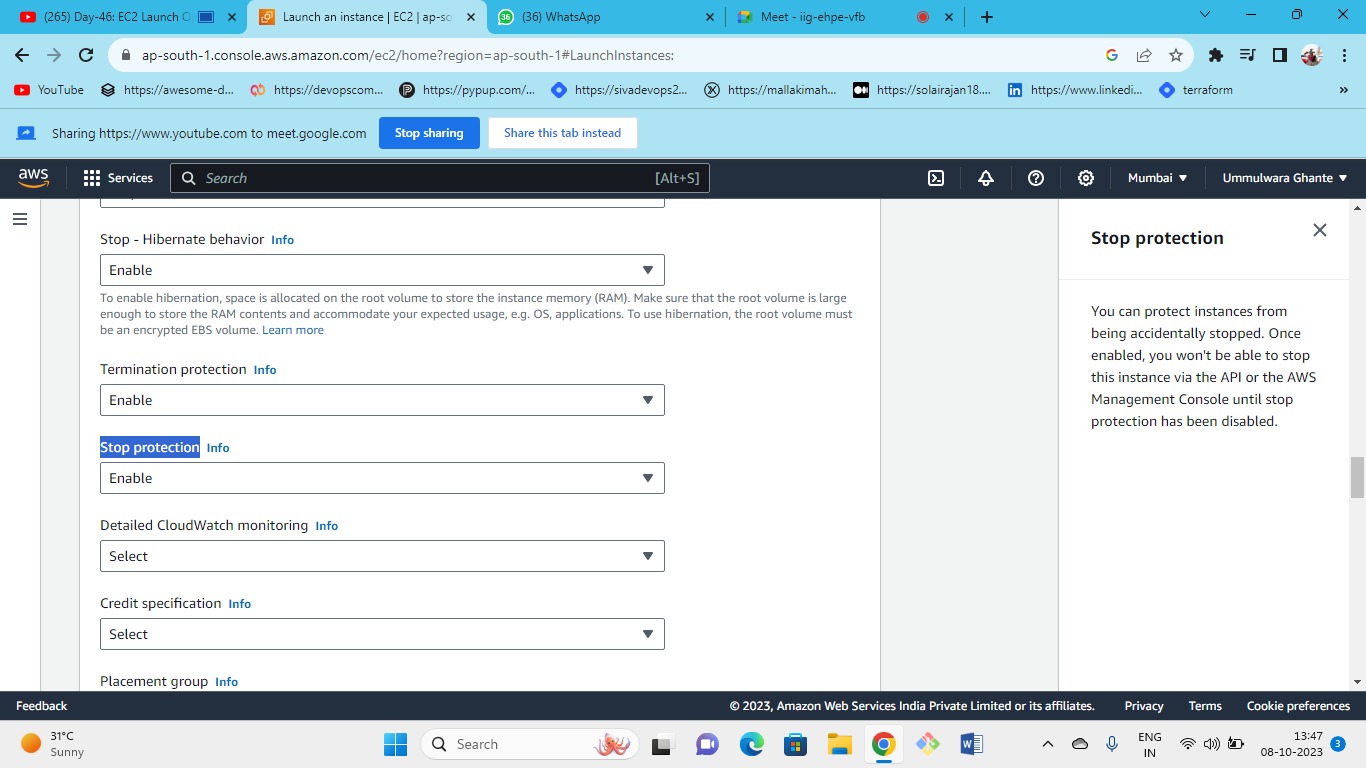


Hibernation stops your instance and saves the contents of the instance's RAM to the root volume. You cannot enable hibernation after launch.

Termination protection- by enabling this no one can terminate our instance

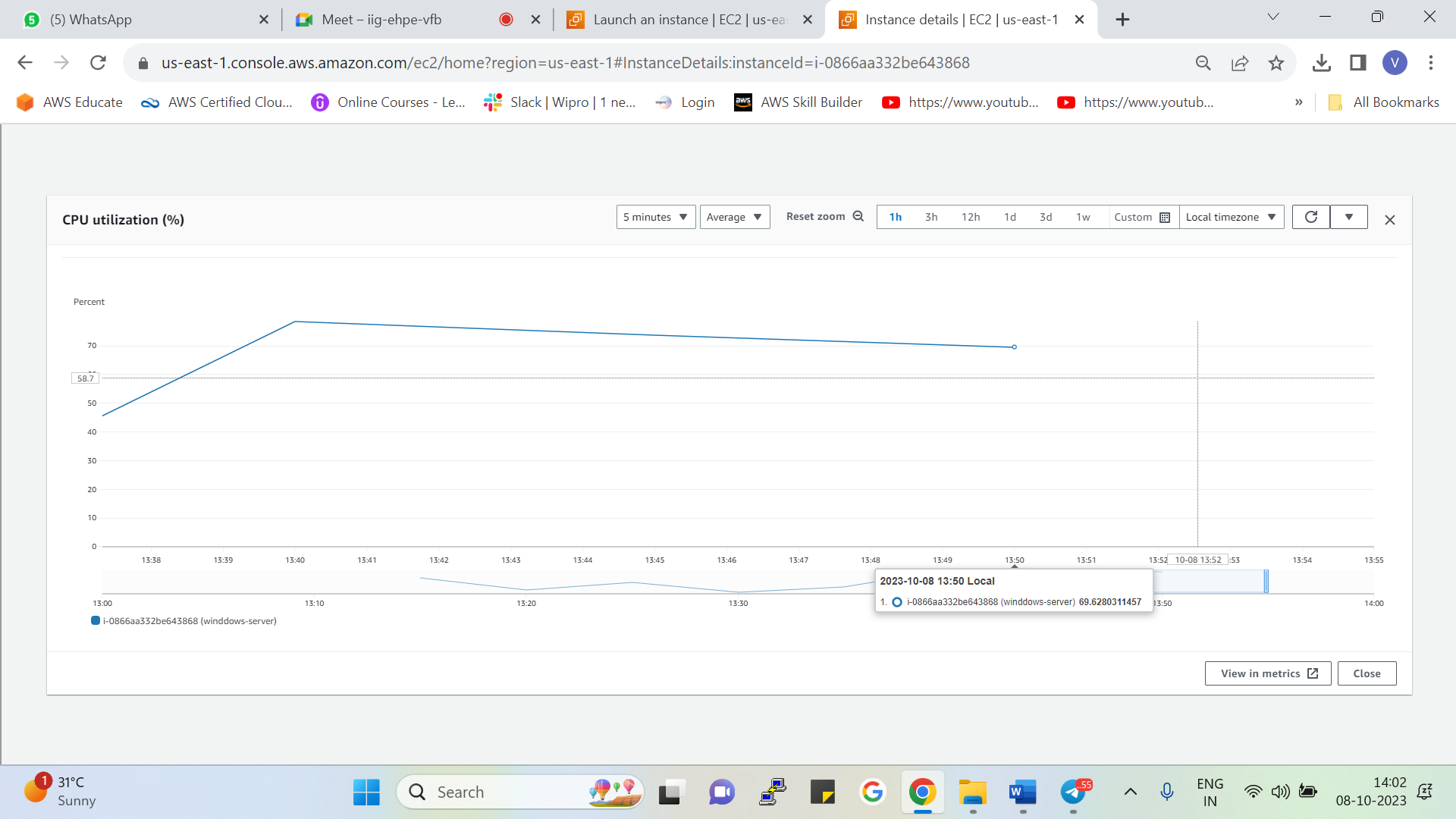




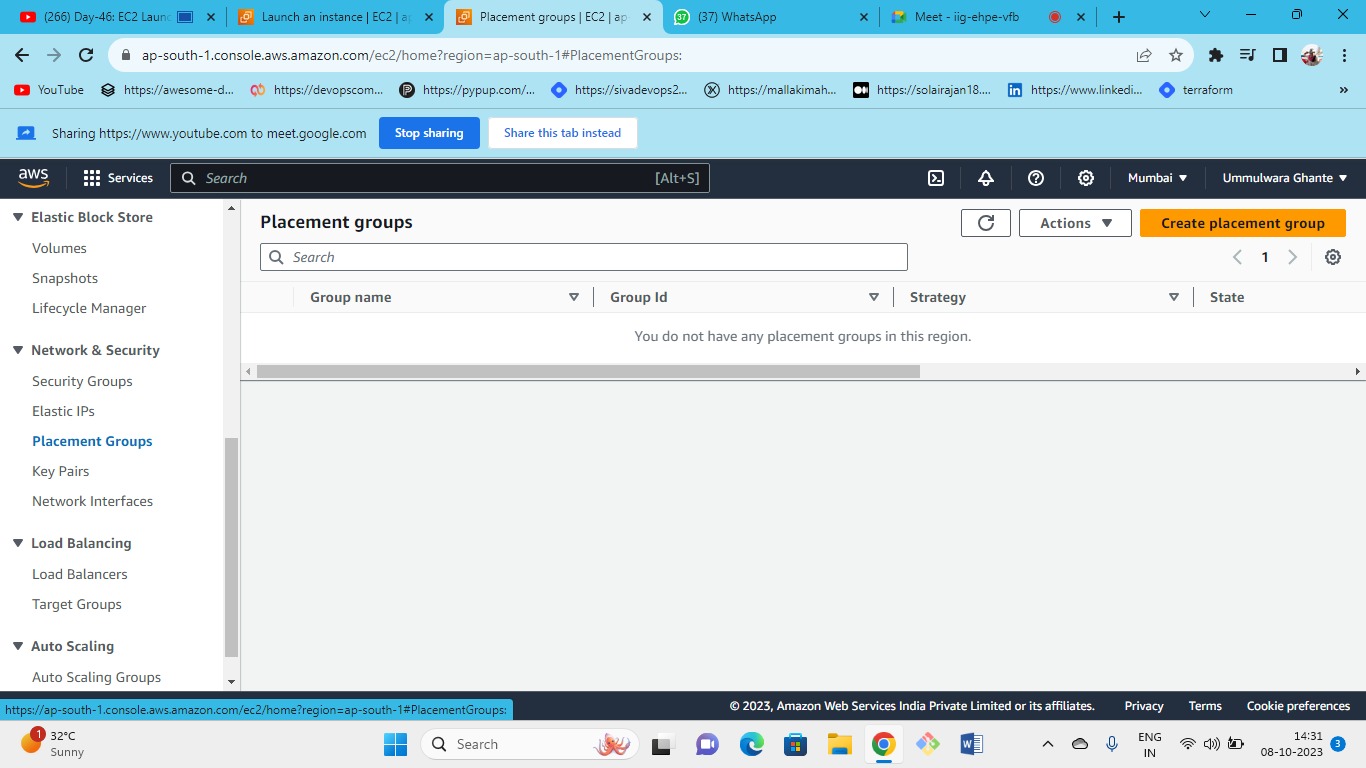




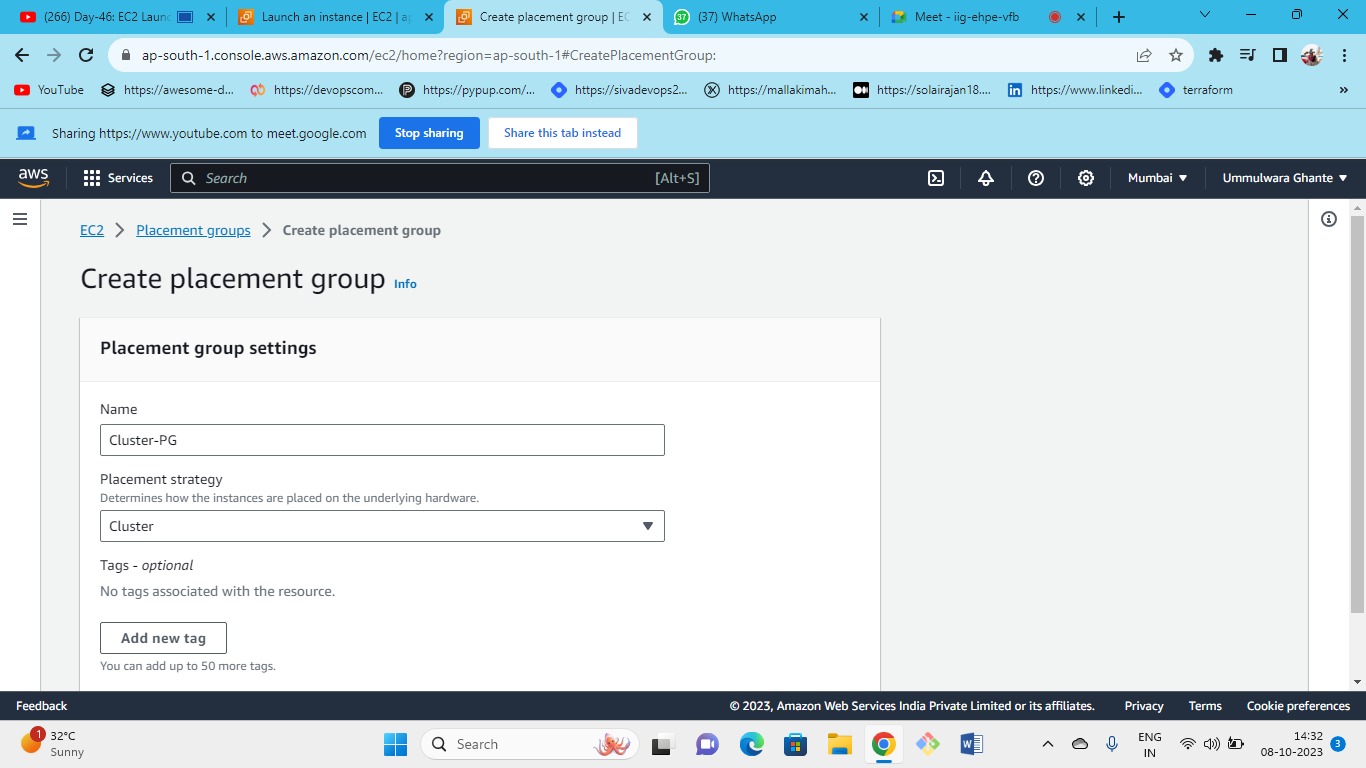
Stop Protection - no one can stop our instance



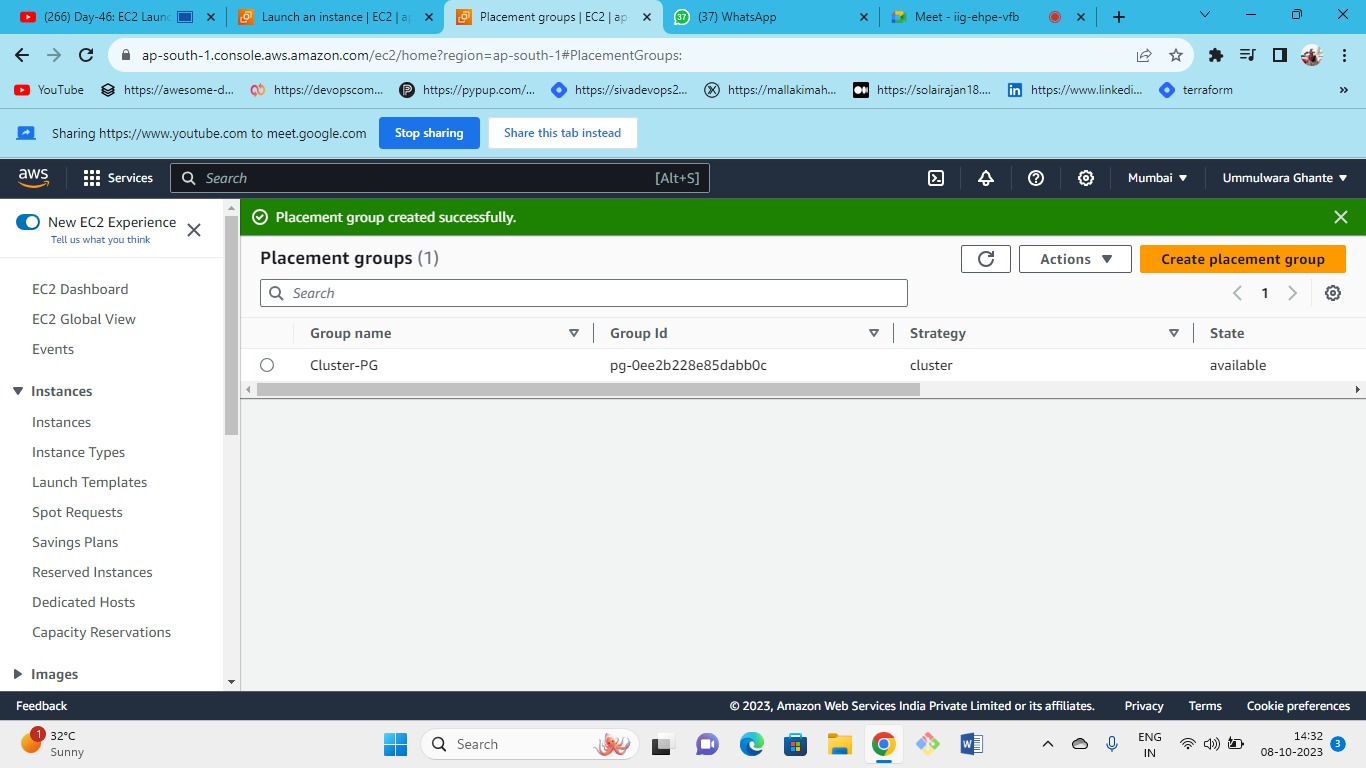
* As we can see this is the CPU utilization of the previous instance
* At every 5 min, we will get the data points this is called the basic monitoring
* Detailed monitoring has 1 min interval granularity and its chargeable













Placement group is nothing but how you want to place your EC2 instances.

Placement Groups consist of three types

1. **Cluster** – All instances in P1 share the same underlying hardware with instances or groups within same AZ

**Adv. :** High Performance, low latency, tightly coupled

**Drawbacks:** Failover (AZ failover), low availability, less redundancy, single performance failure.

AZ121111111111

P1

1. **Partition**

* All these partitions are in same AZ but do not share same underlying hardware with each other.
* Can launch multiple instances in one partition.
* But in partition one all instances will be sharing same underlying hardware.
* P1 and P2 don’t share same underlying hardware.
* Here we have different parts or partitions but all the partitions are in same AZ.

**Adv :** High Performance, Hardware failure problem solved, good redundancy

**Drawbacks :** if AZ fails then all will be down

AZ11

Part 3

Part 2

Part 1

1. **Spread** - Here every instance is placed on different hardware

at different rack spread (host level spread) each rack has 7 instances its own network and power source.

It works on multi-AZ

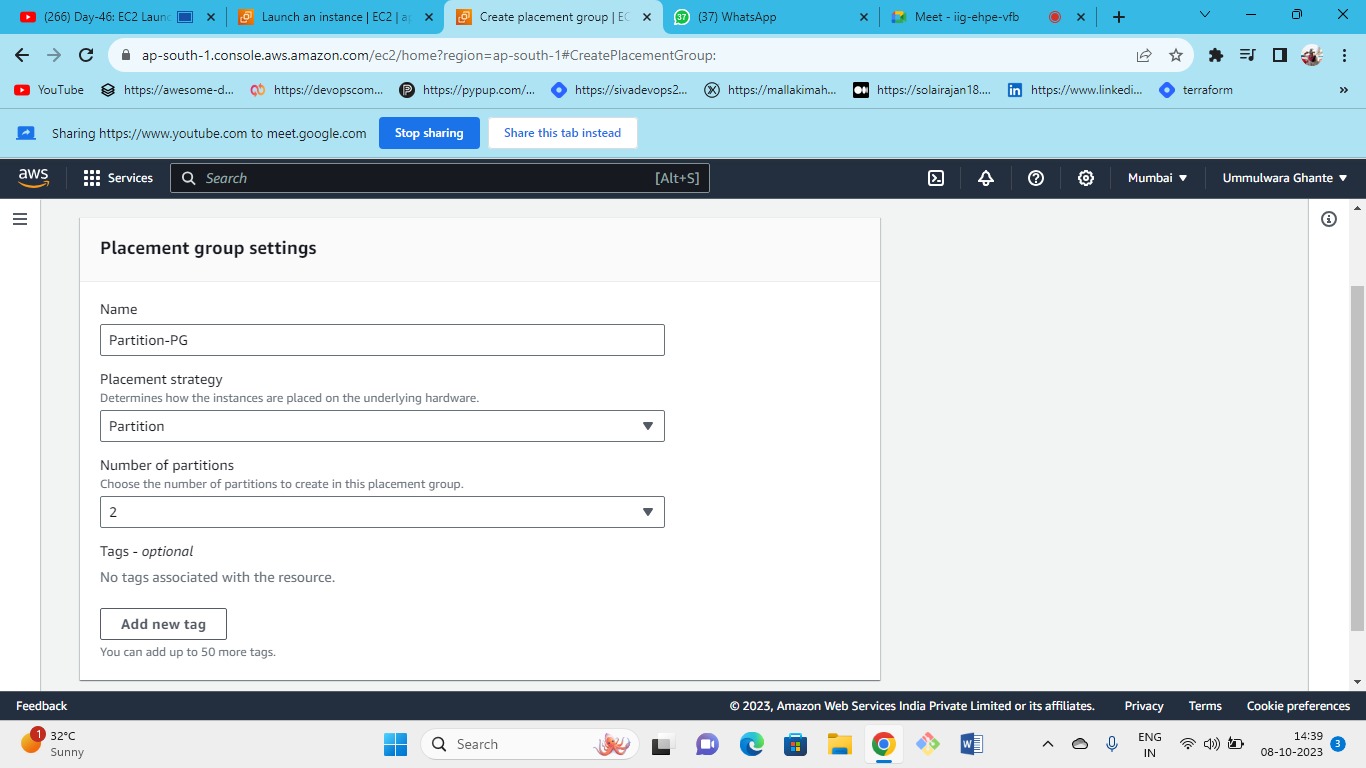
**Adv :** high latency, zero downtime

**Drawbacks :** low performance

AZ1

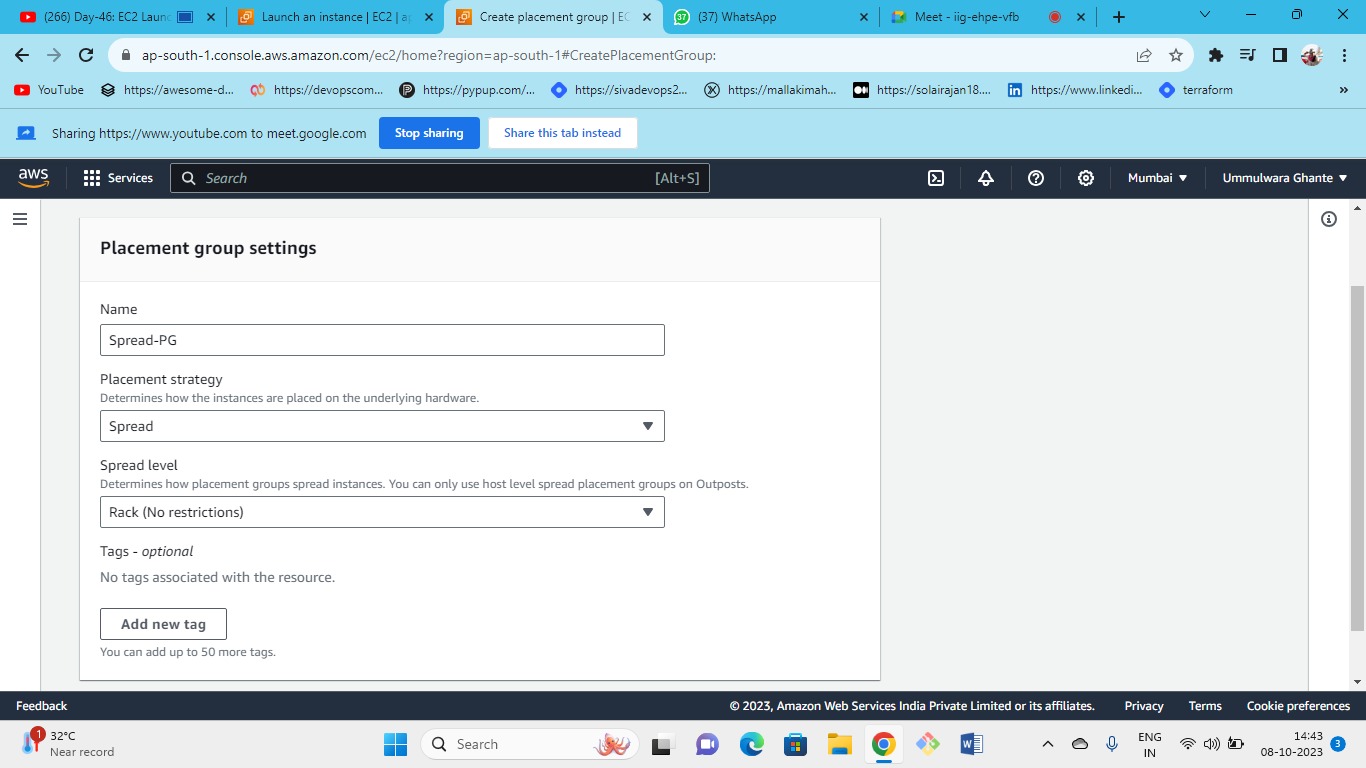
Instances

Part-1 Part-2 Part-3 Part-4

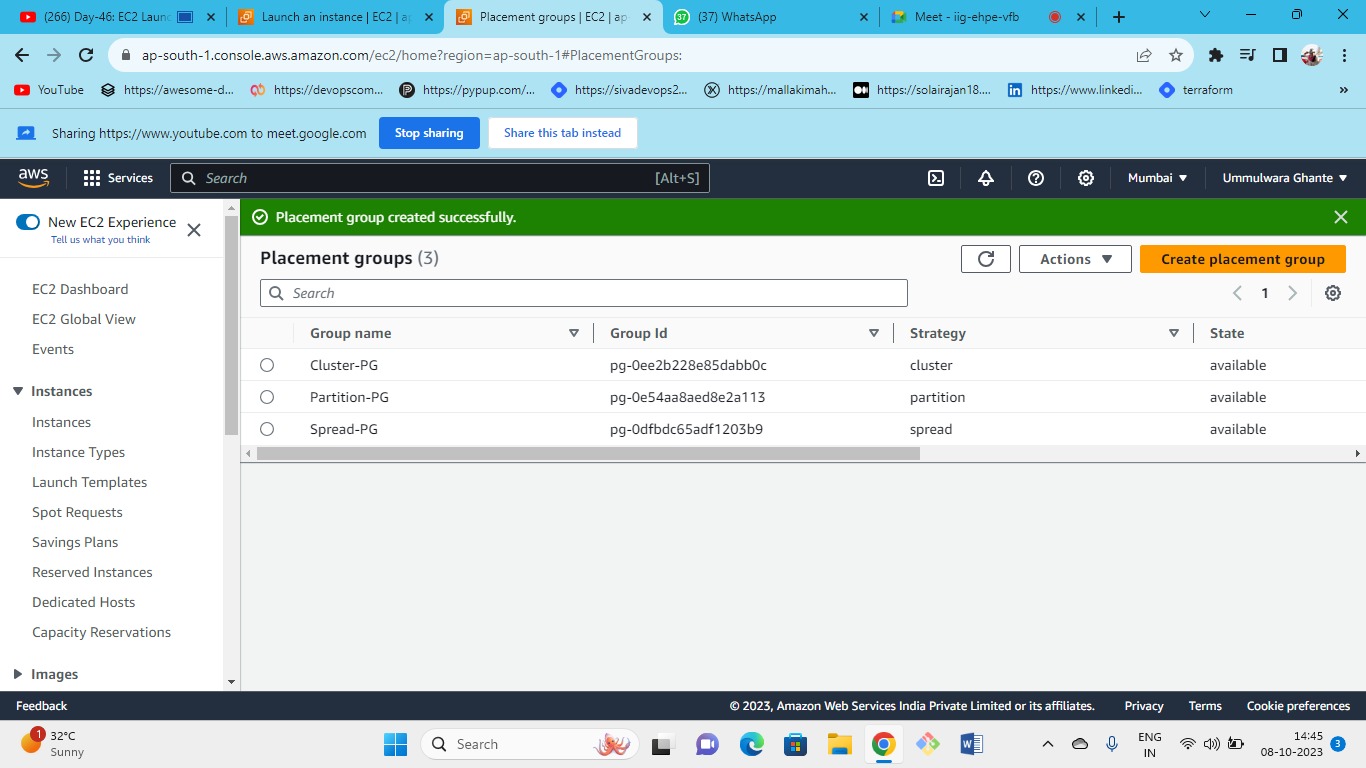




as you do the partition according to that the instance will divide

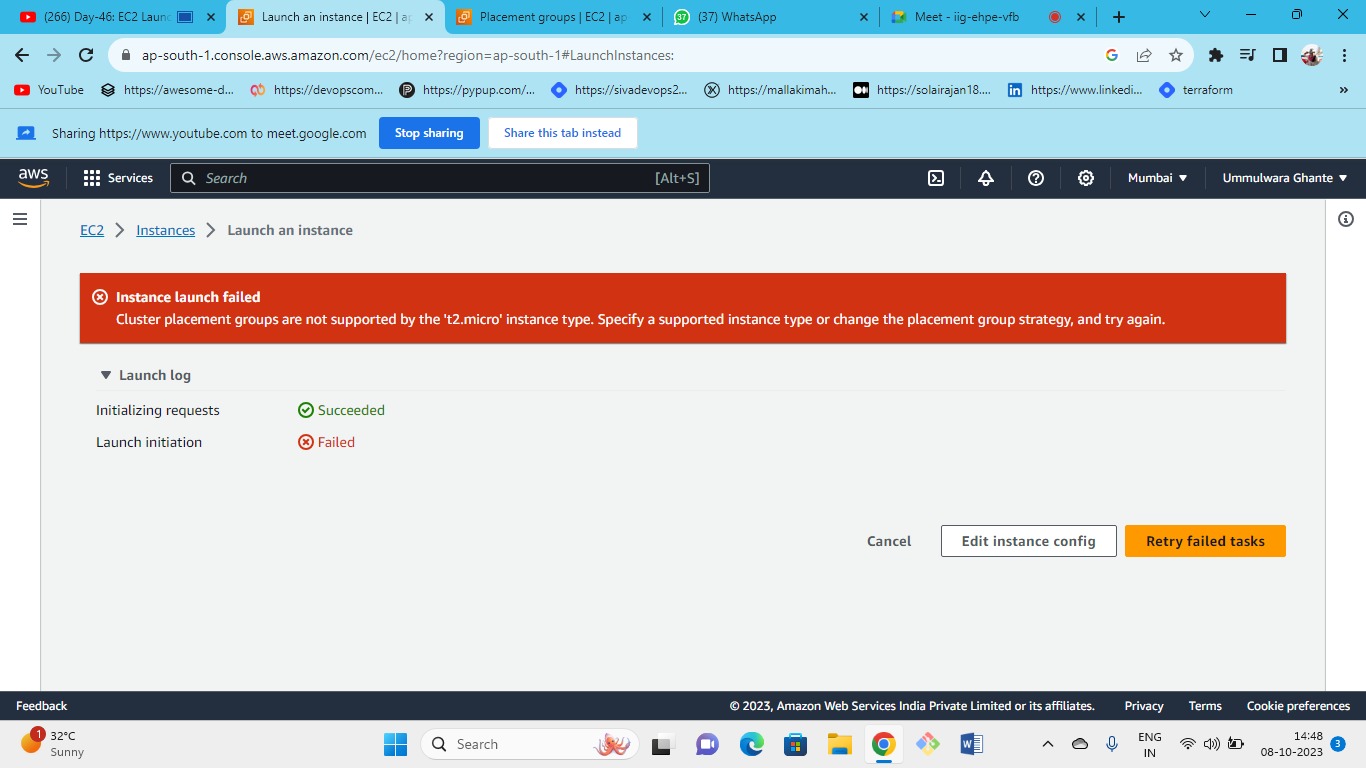






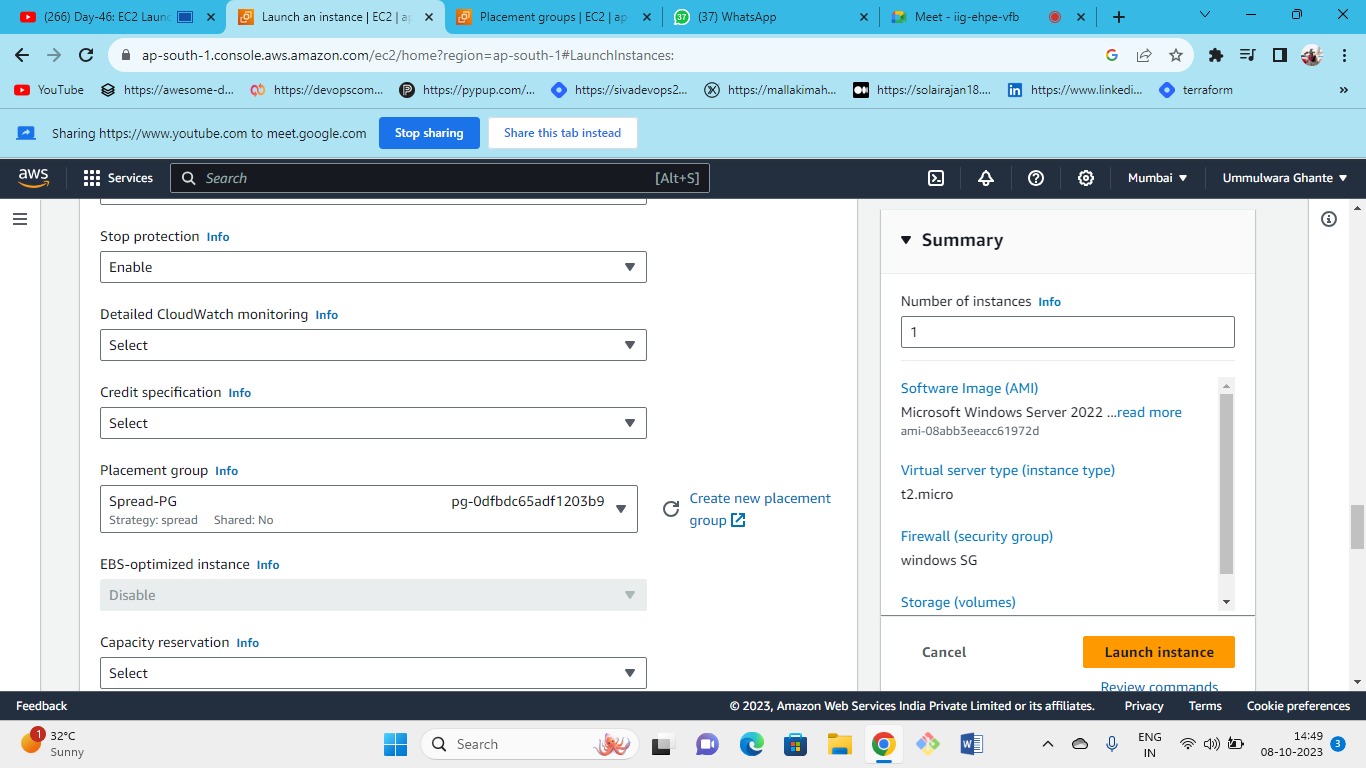


As we have first kept the Cluster PG so we got the following error.



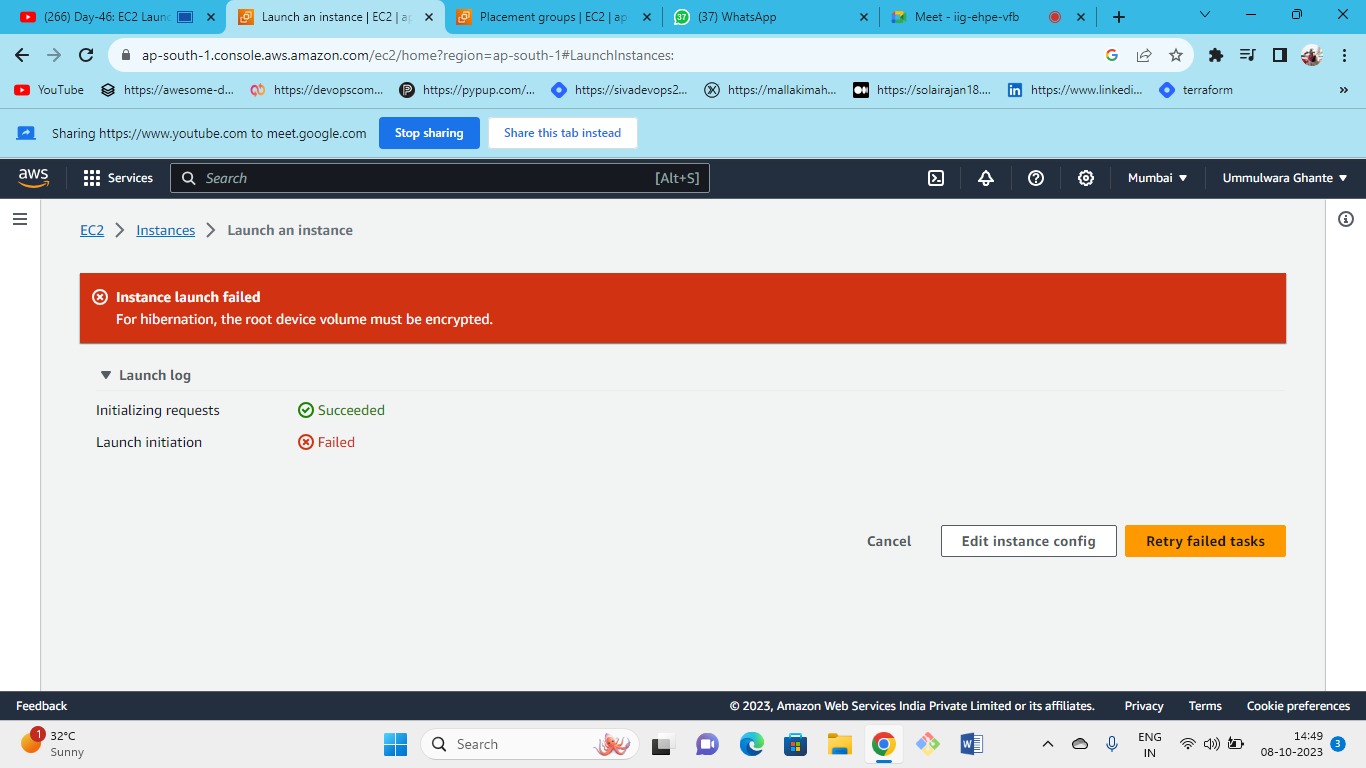


Instance launch failed because cluster PG do not support t2 micro instance type so we can change the instance type or change the placement group



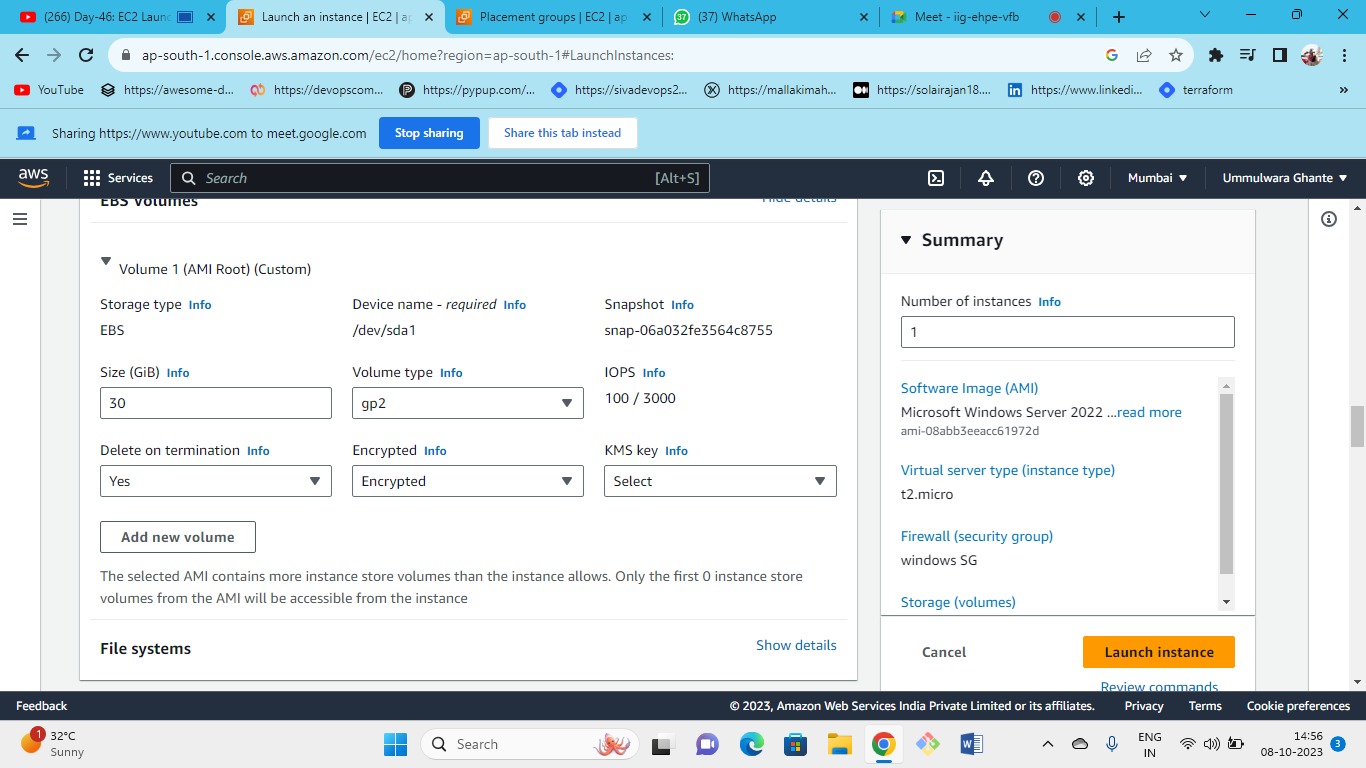


So here we have changed the placement group type to spread

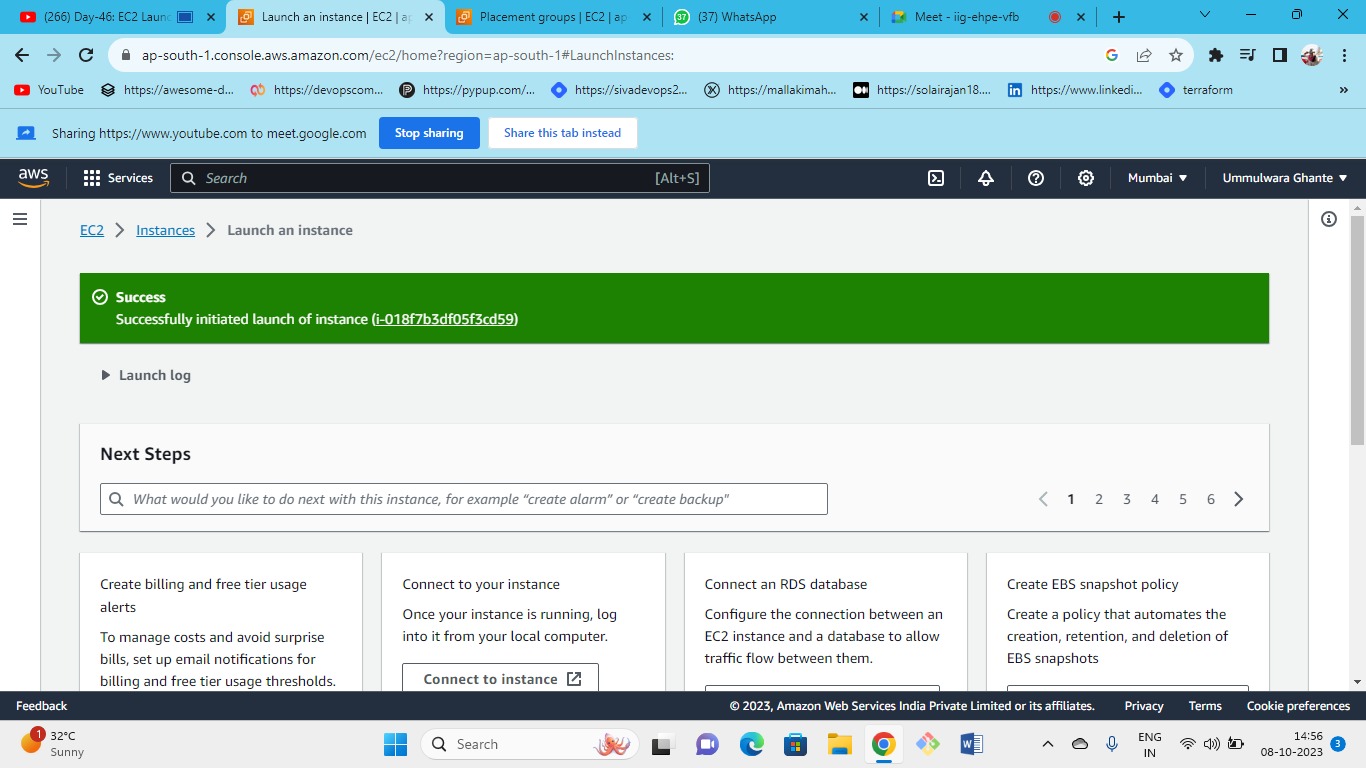




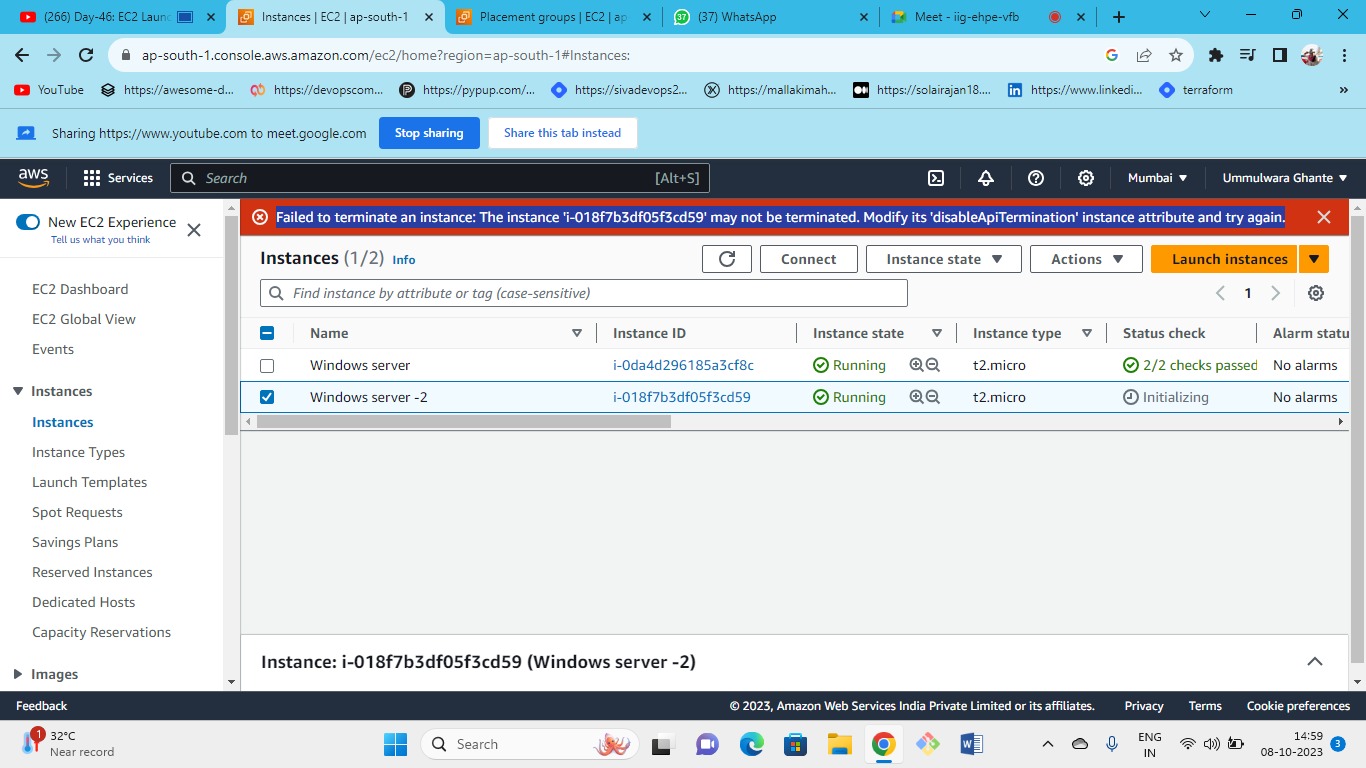
As we can see we need to encrypt our root volume





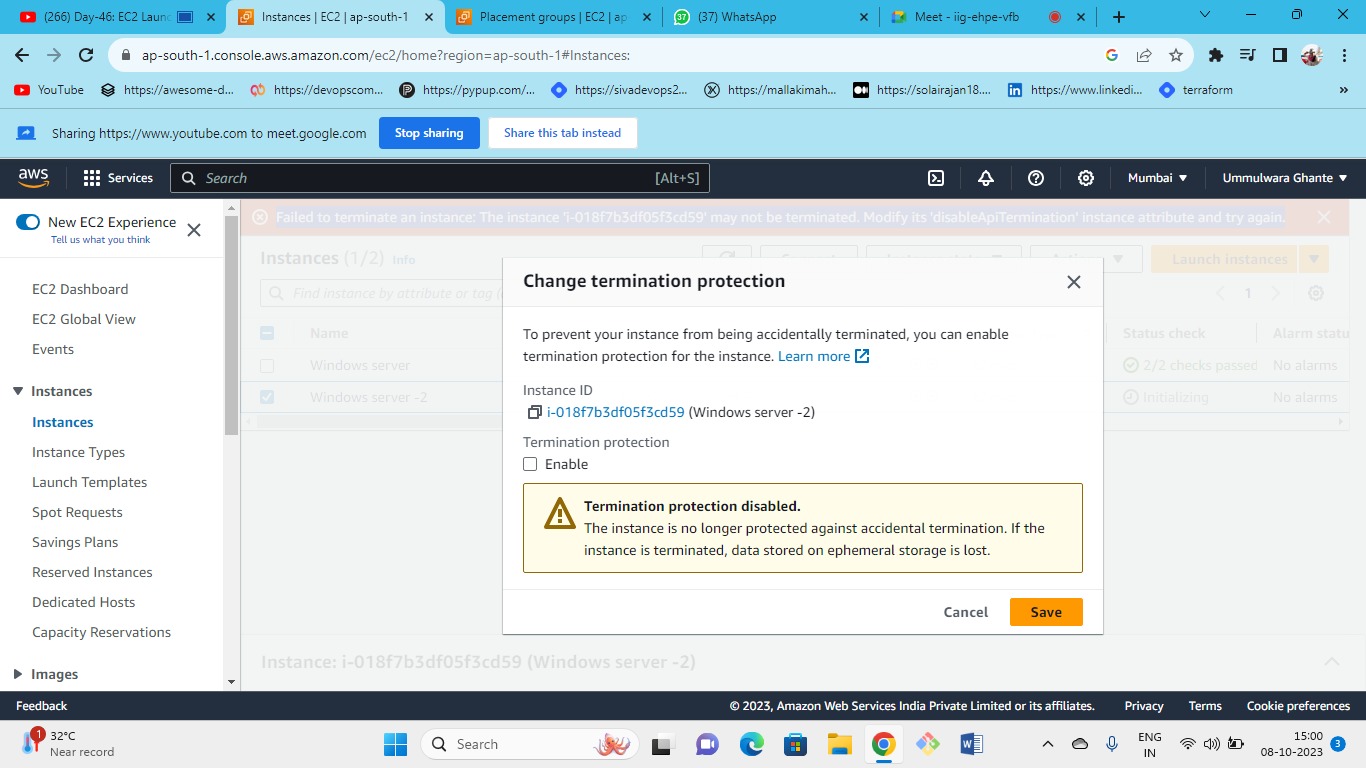






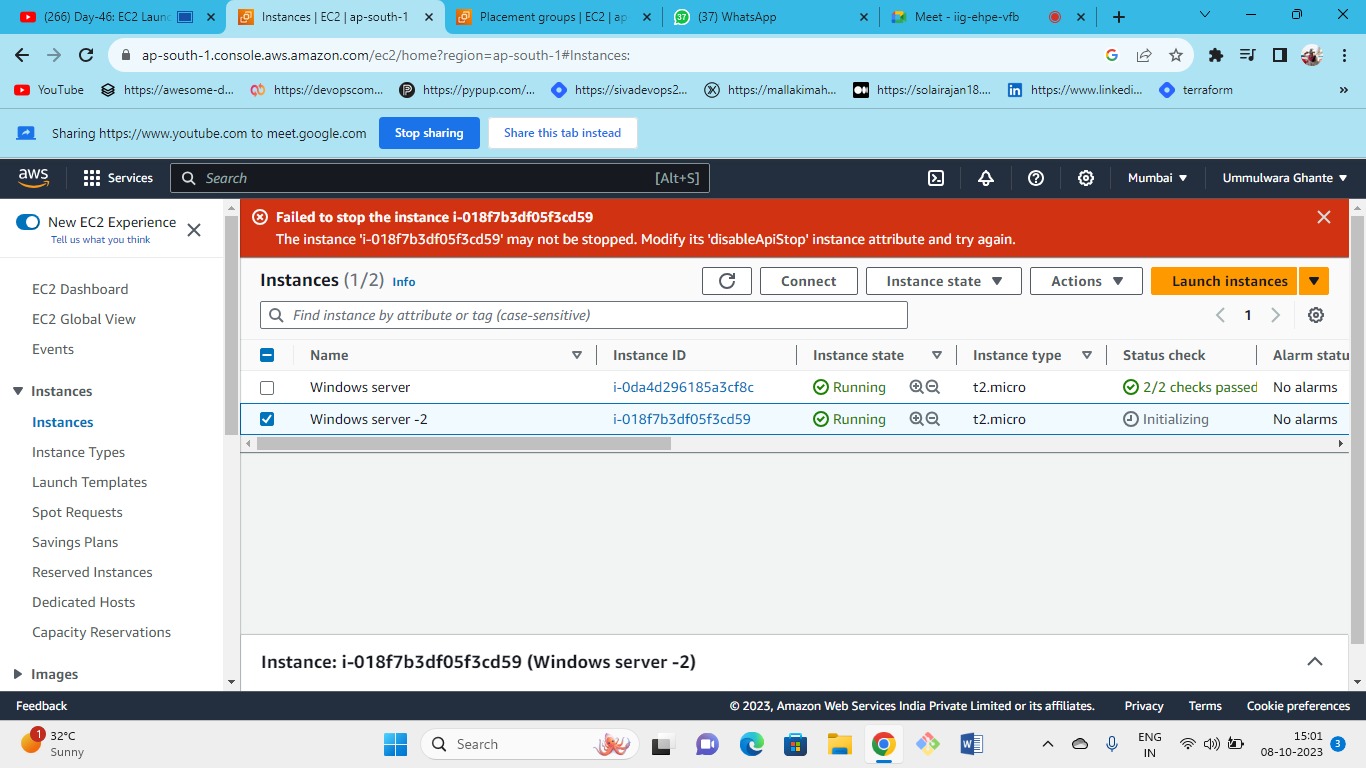


as we have enabled the terminate protection so our instance is not getting terminated so we need to modify the permissions



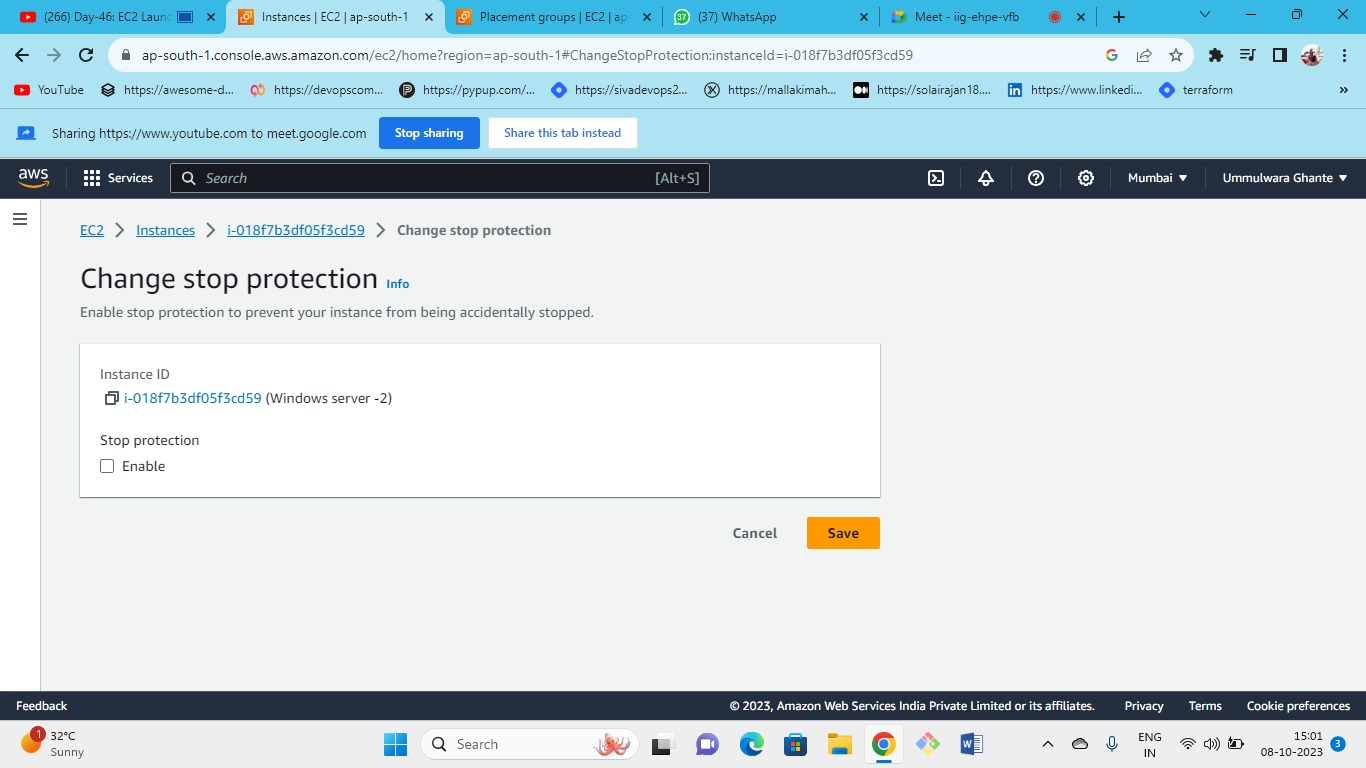


We can change the termination protection as disable



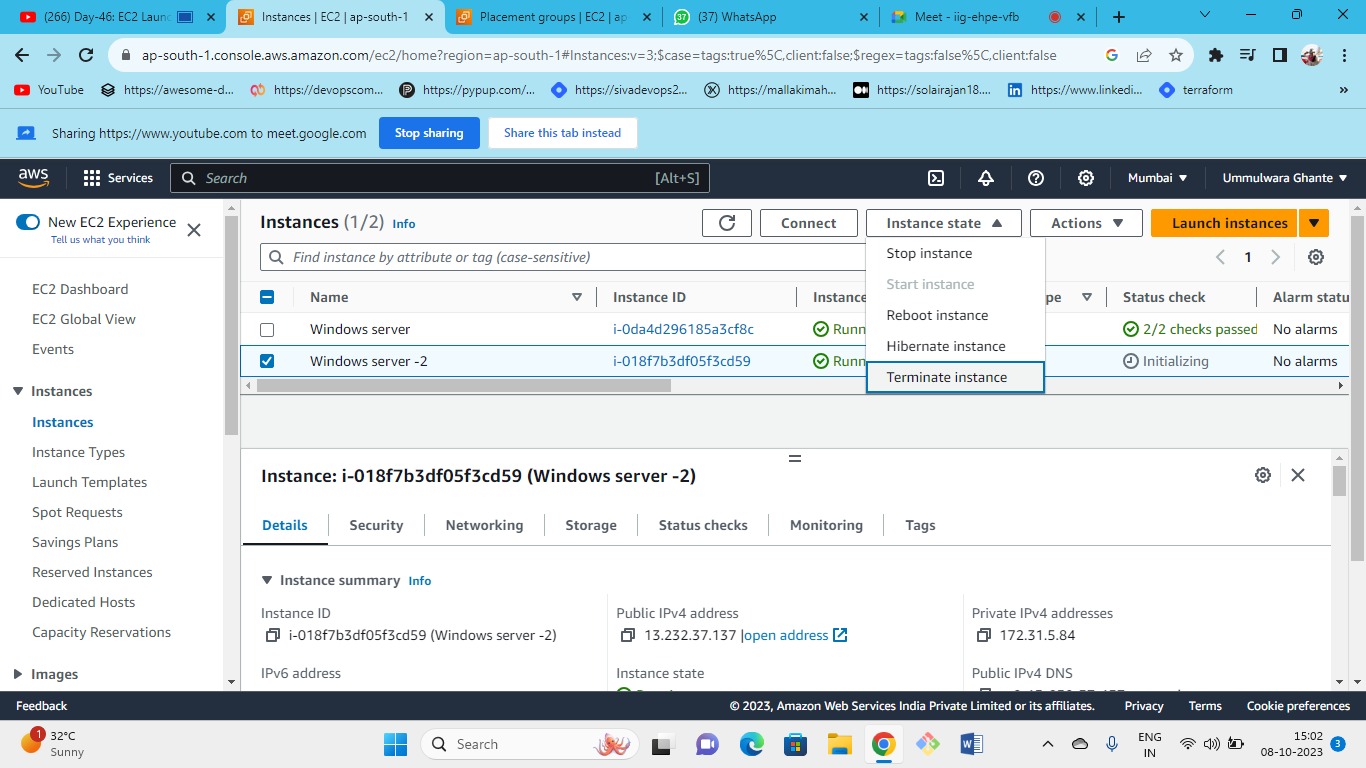


Even we can’t stop the instance as we have enable stop termination protection so we also need to modify the permissions of stop protection

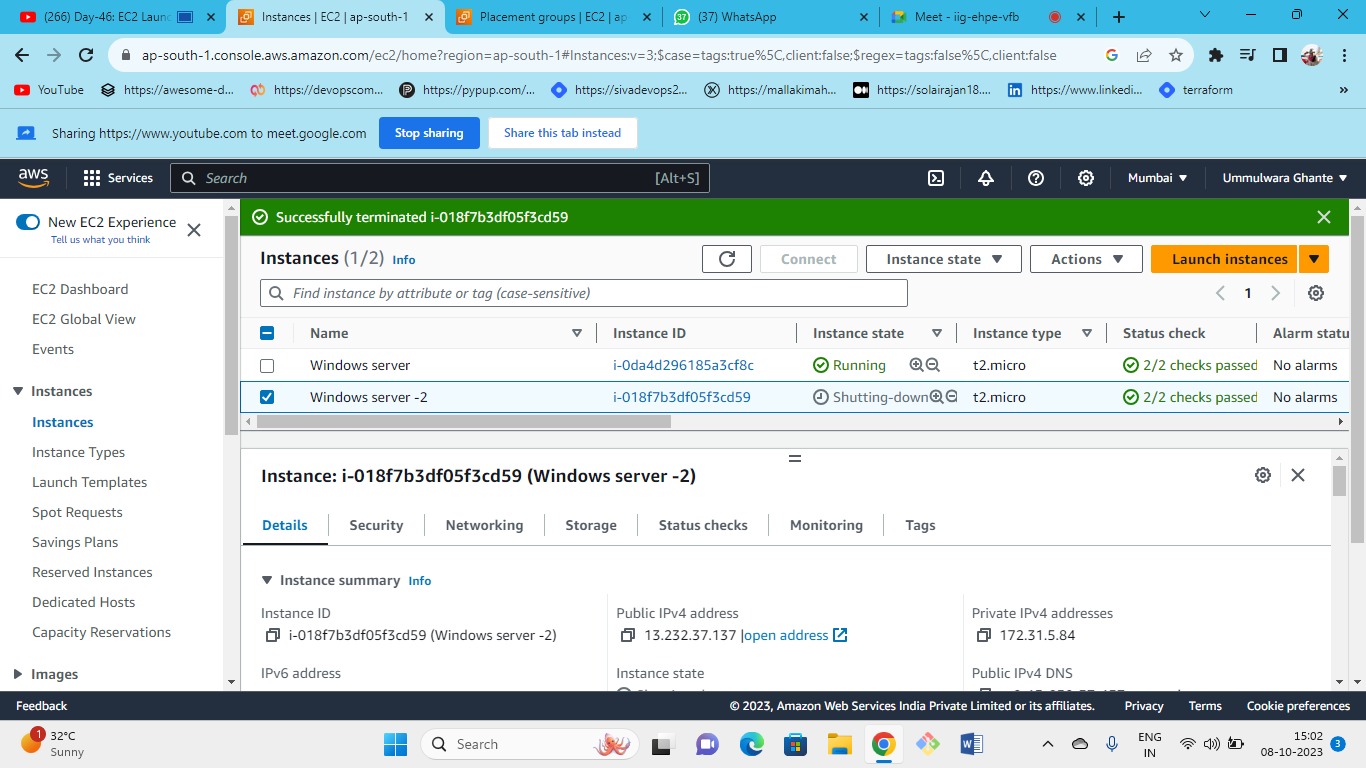




Untick and Save changes

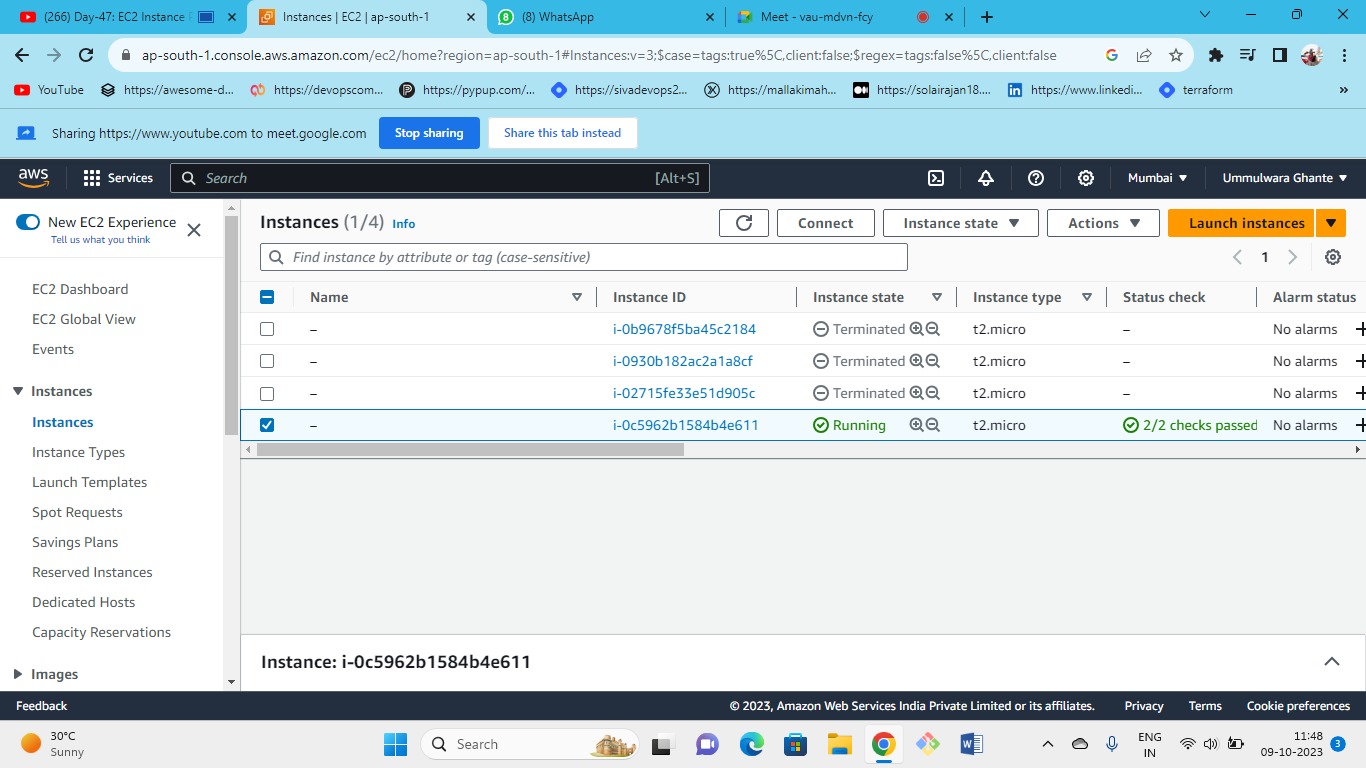




even we have hibernate option available here

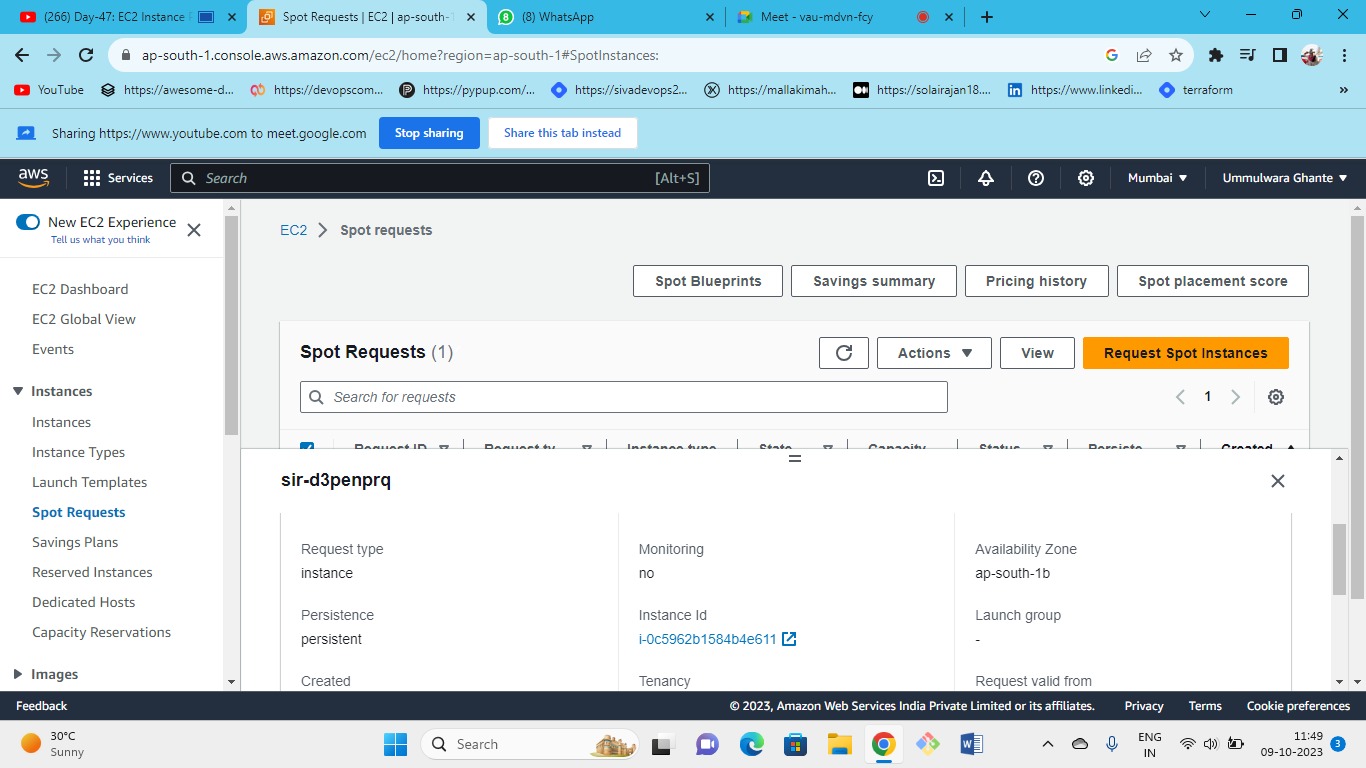


finally, we terminated the instance



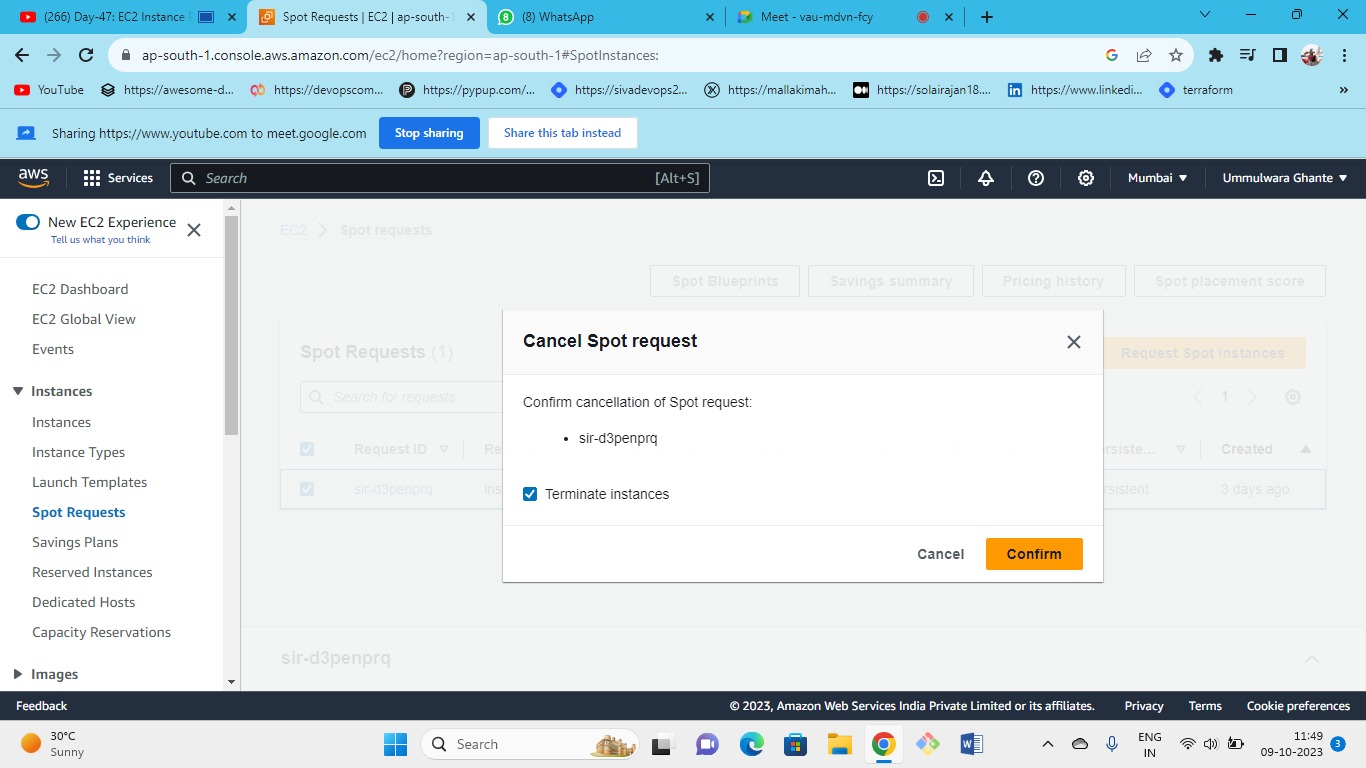


As we have seen that we terminated the instance but still the instance will be running even if we terminate the instance, it will create another instance with same configurations this is because we have created spot request and it’s persistent.



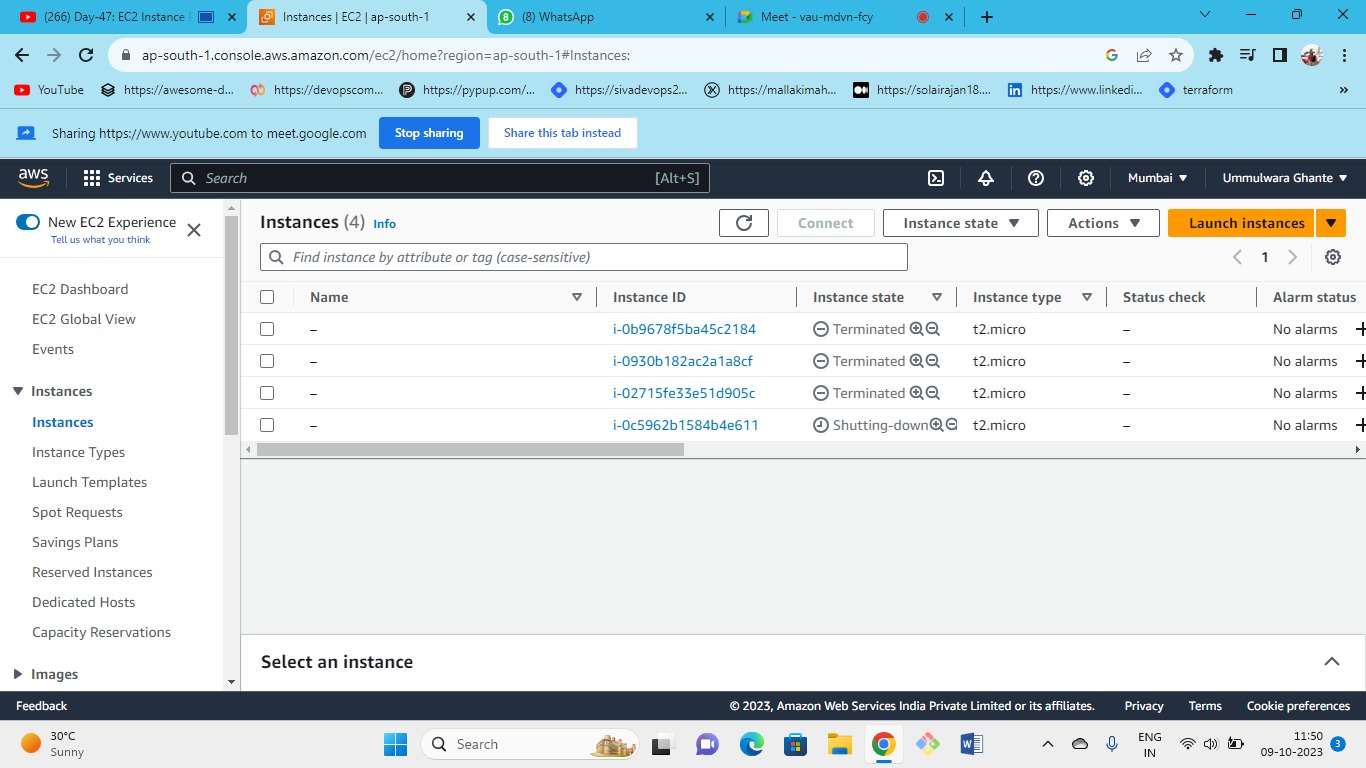


To cancel🡪spot request🡪actions🡪cancel request

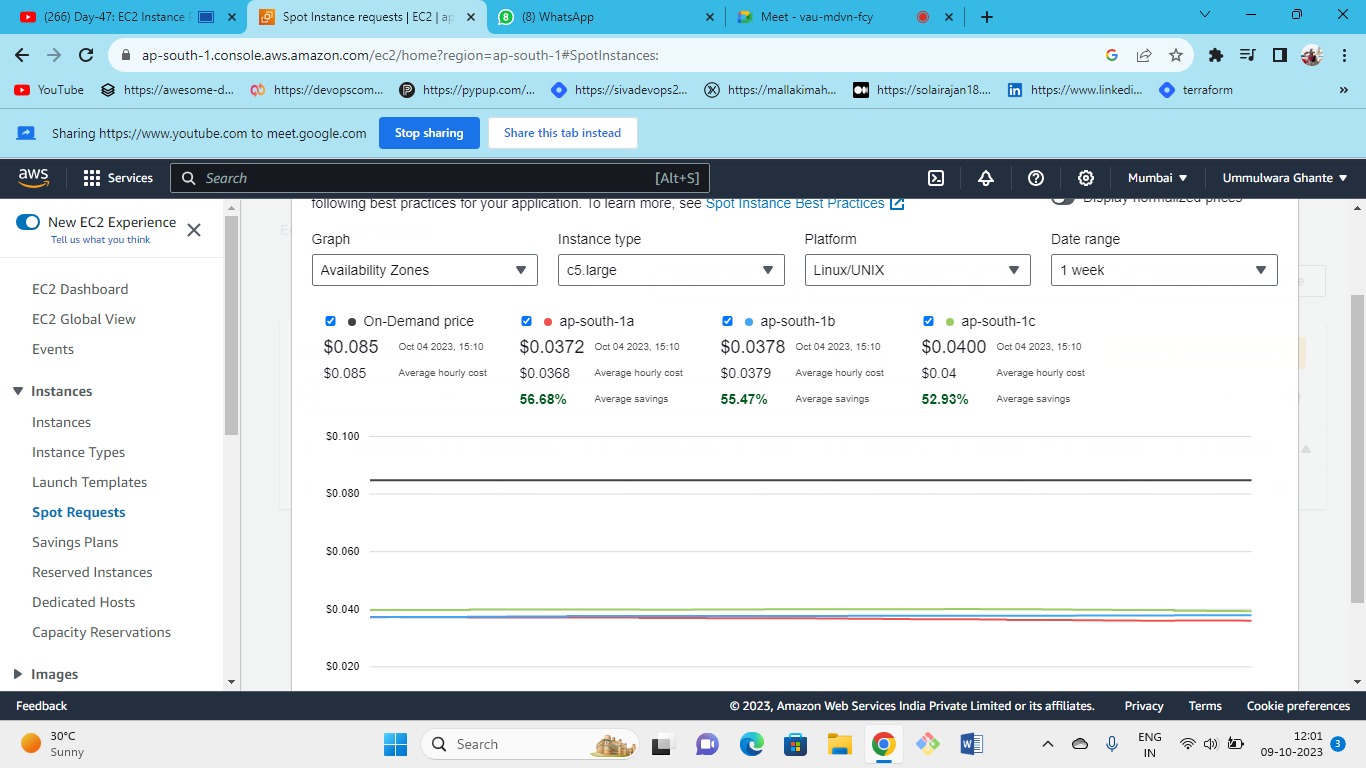




Tick the checkbox of terminate the instance and the request will be cancelled and instance will be terminated.









Spot instance size fluctuates and price vary because of the capacity but the on-demand price stays constant.

Discount on price also depending on the instance type you choose.