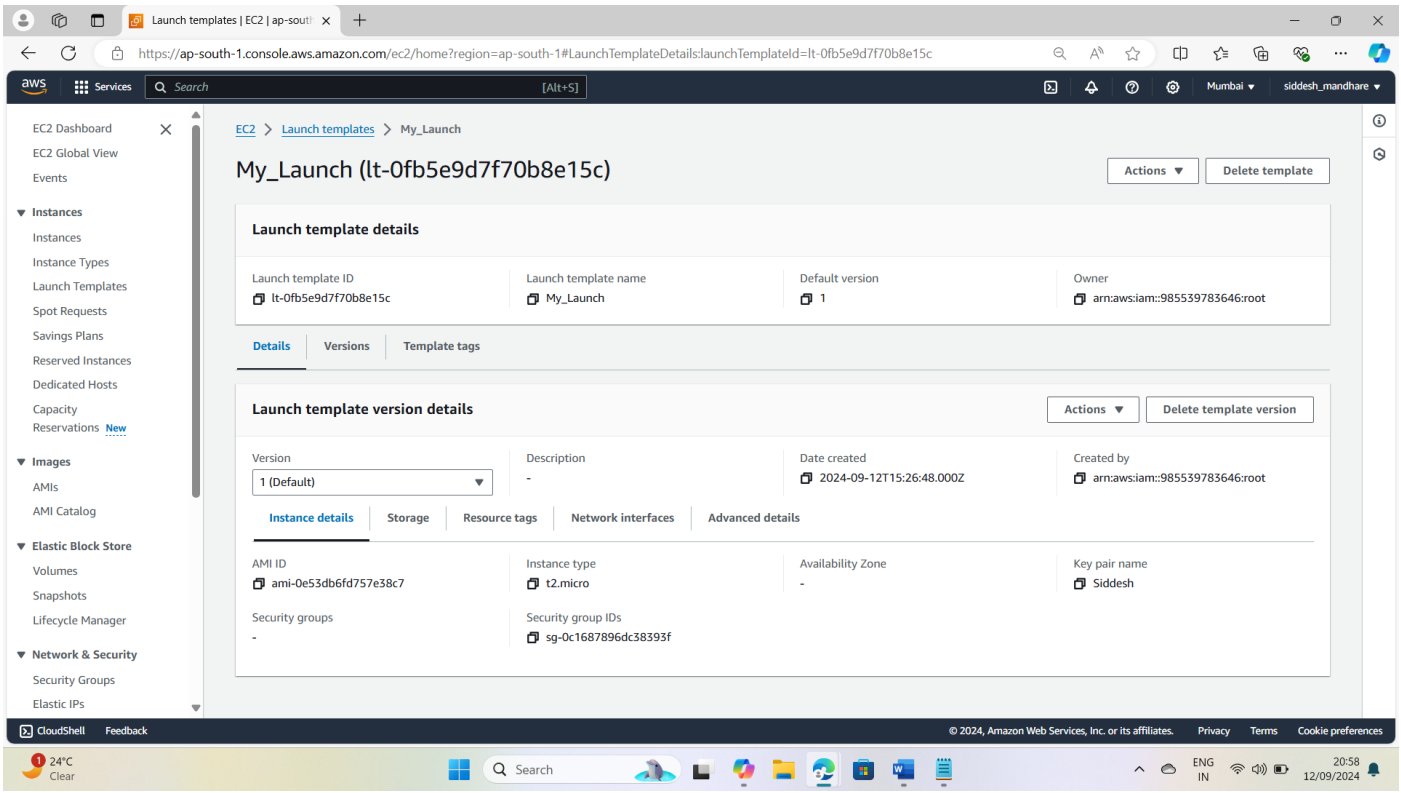
AWS- Auto Scaling Practical

Siddesh Mandhare || 10/09/24

Name: - Autoscaling ensures your application stays available by adjusting resources as needed.

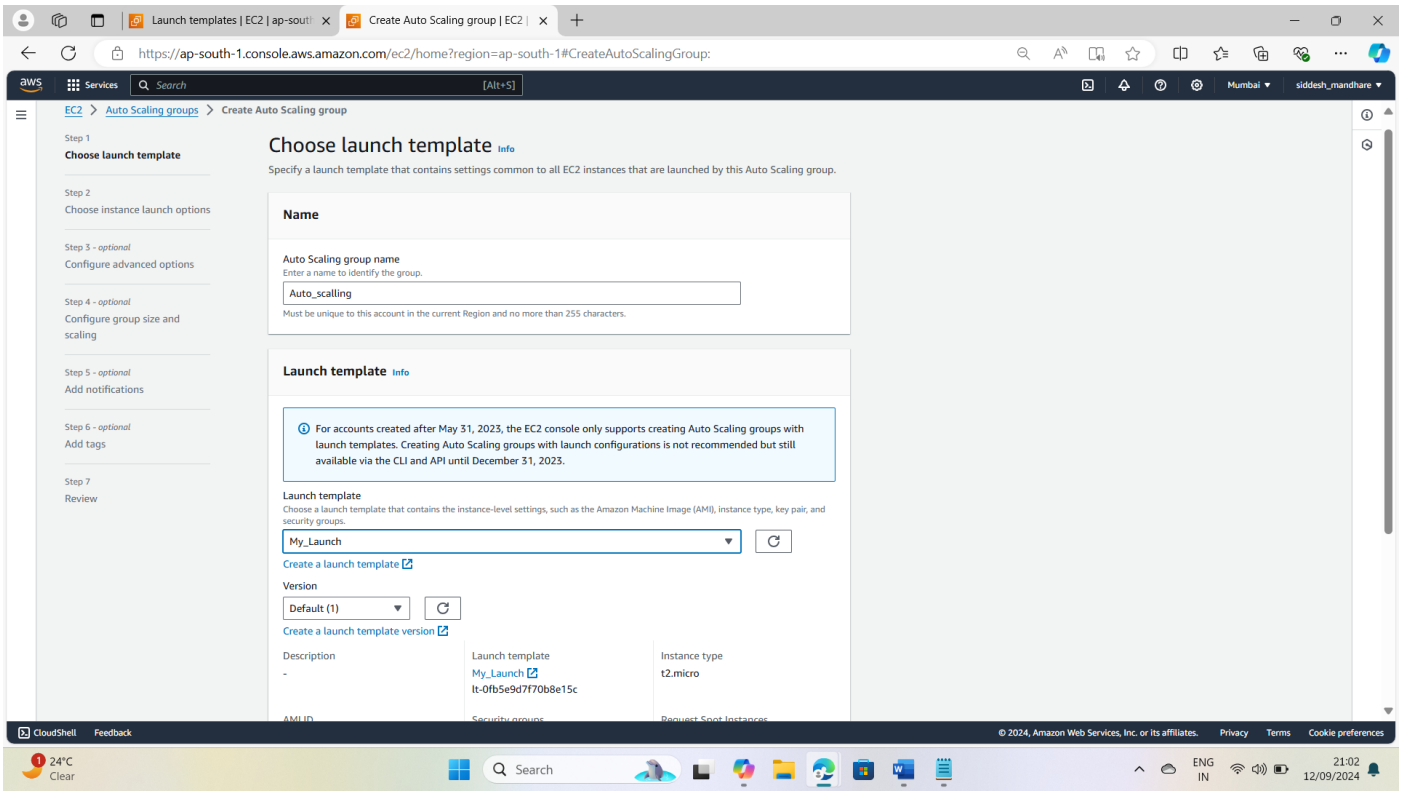
Step 1: - Create launch template configuration.

1. Choose Amazon Linux as the AMI.
2. Select the t2.micro instance type.
3. Choose a key pair.
4. Configure the security group. (efs_demo)
5. Assign a IAM role with access of S3FullAccess.
6. In advanced settings, add a bootstrap script. (Copy S3 file)
7. Create the launch template.

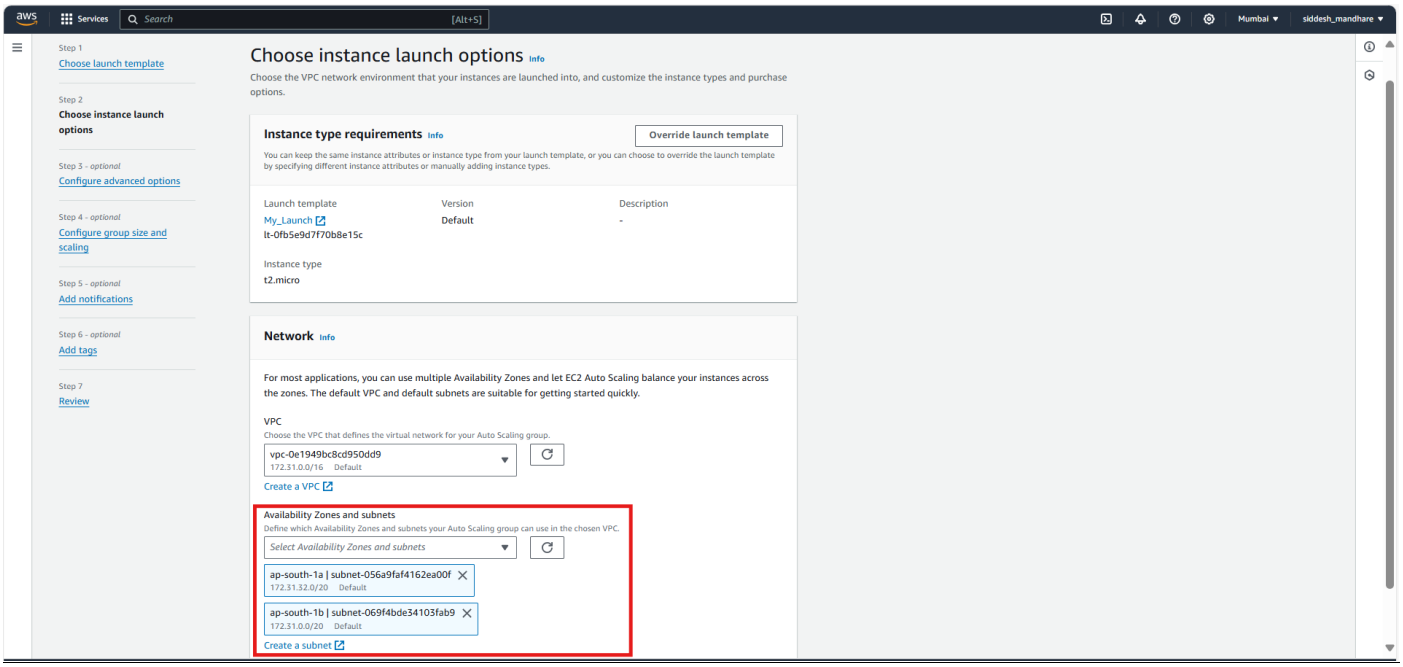


Step 2: - Create Autoscaling Group

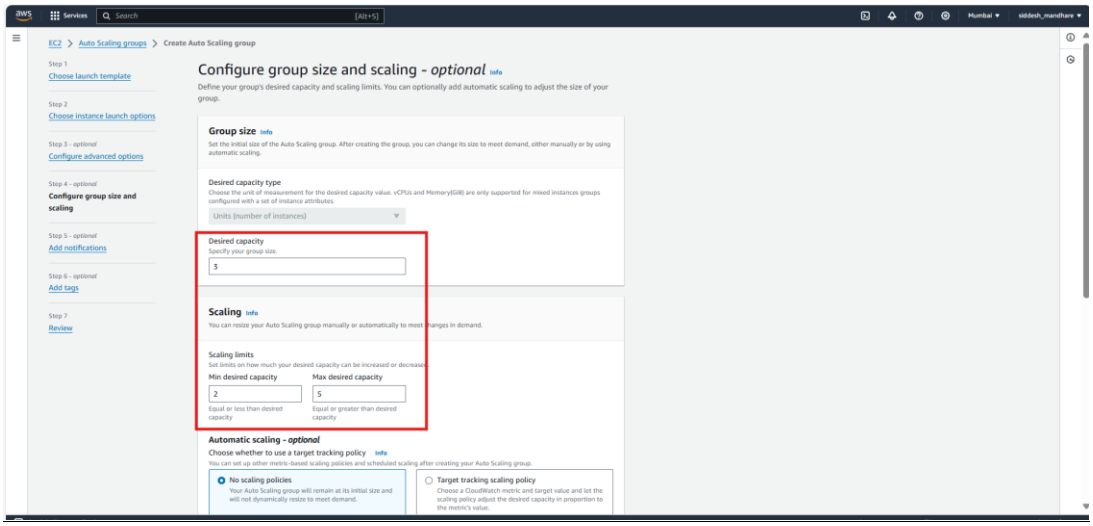
1) Provide group name and select template which is created



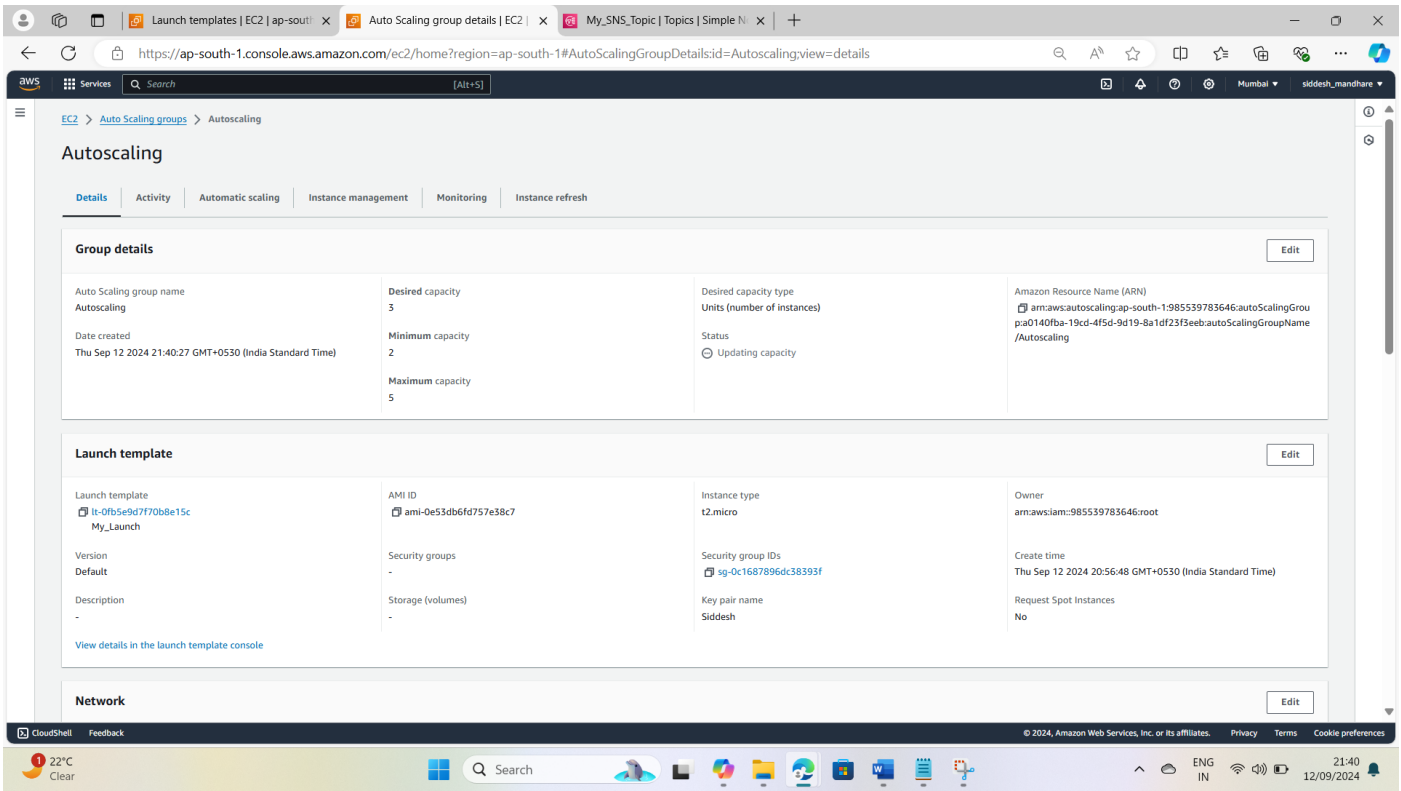
2) Select Availability Zone which you want to create instance



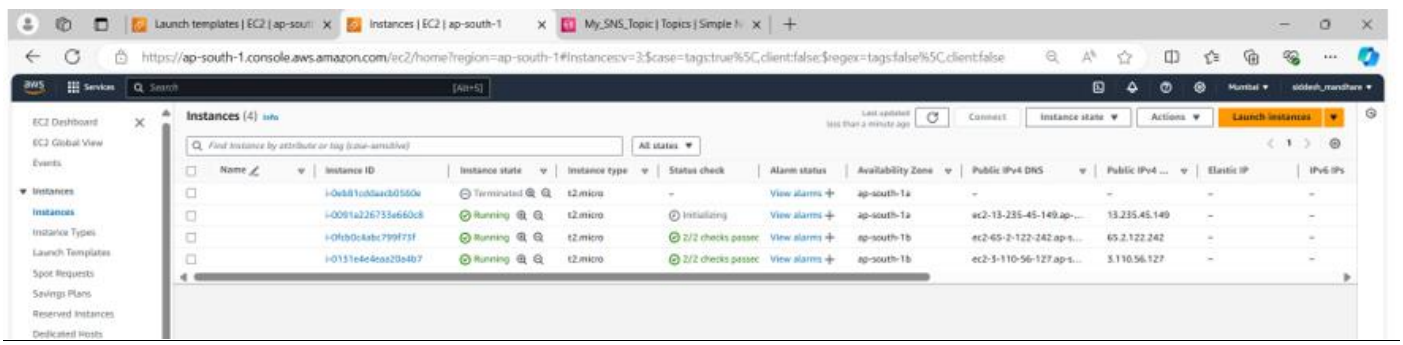
3) Configured group size and scaling as per requirement



4) Autoscaling group have been created

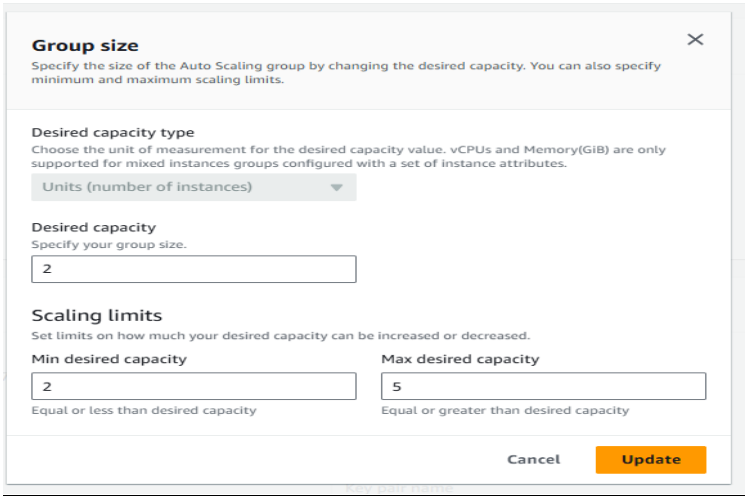


5) Autoscaling group have been created 3 instances as we mentioned in desire “3” count

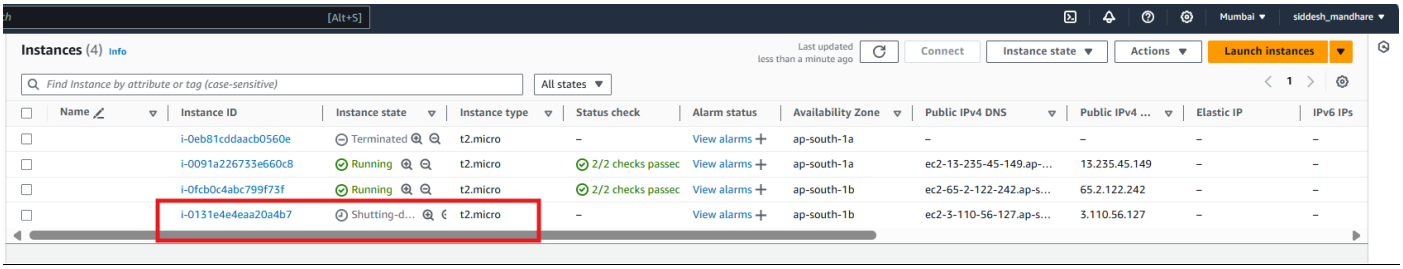


Manual scaling

Step 1: - If I manually change the desired value to 2 from 3, the Autoscaling group will remove 1 instance



Step 2: - We can see one instance is going to terminate

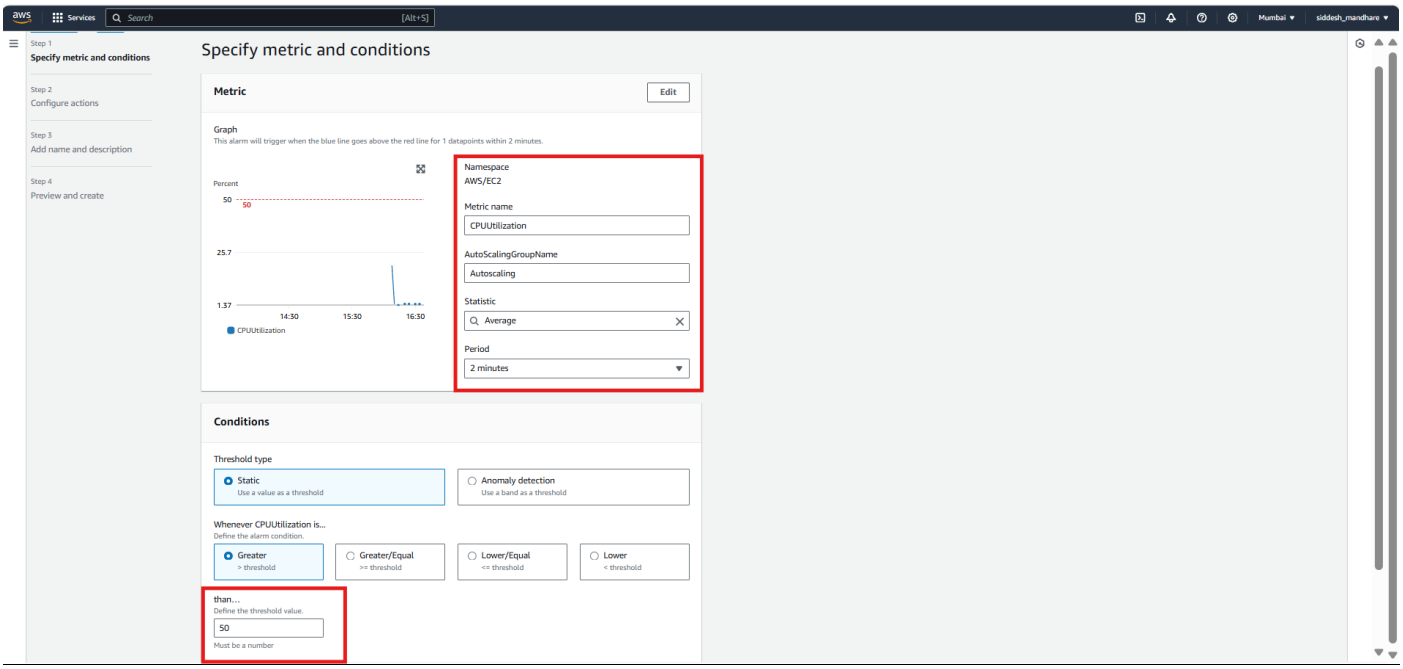


Automatic scaling

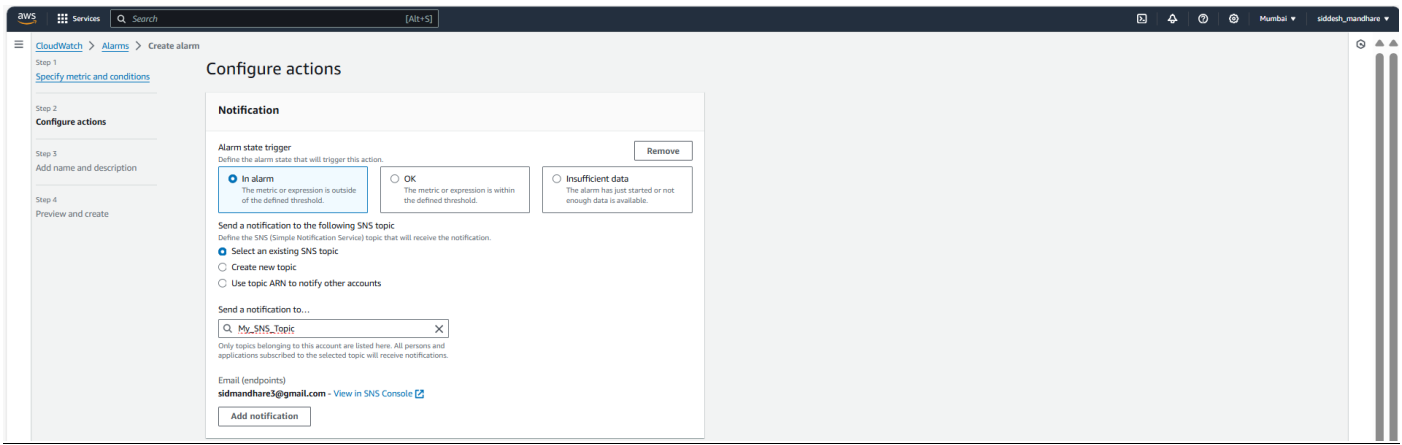
We required alarm to set create “Dynamic scaling policies”

Step 1: - Create alarm for Autoscaling group (Increase)

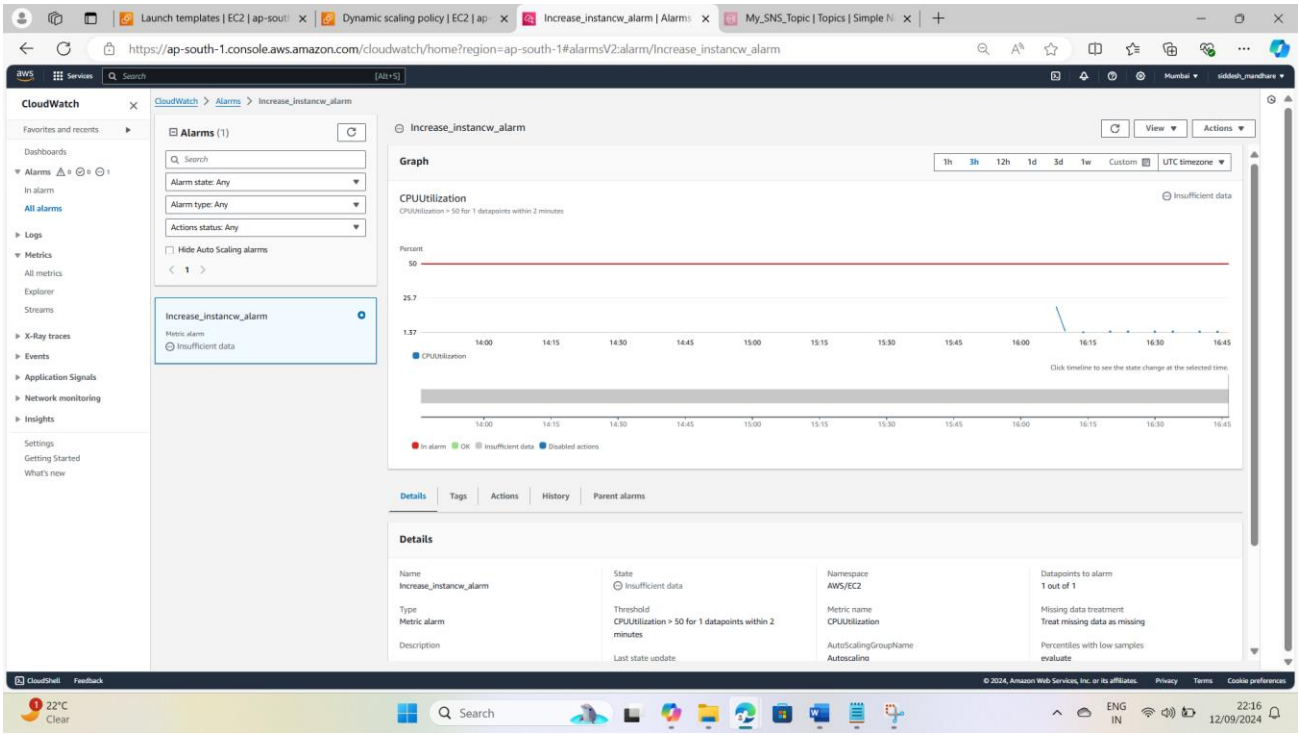
1) Select metrics (CPU Utilization) and time period “1 min” and threshold value (utilization>50)



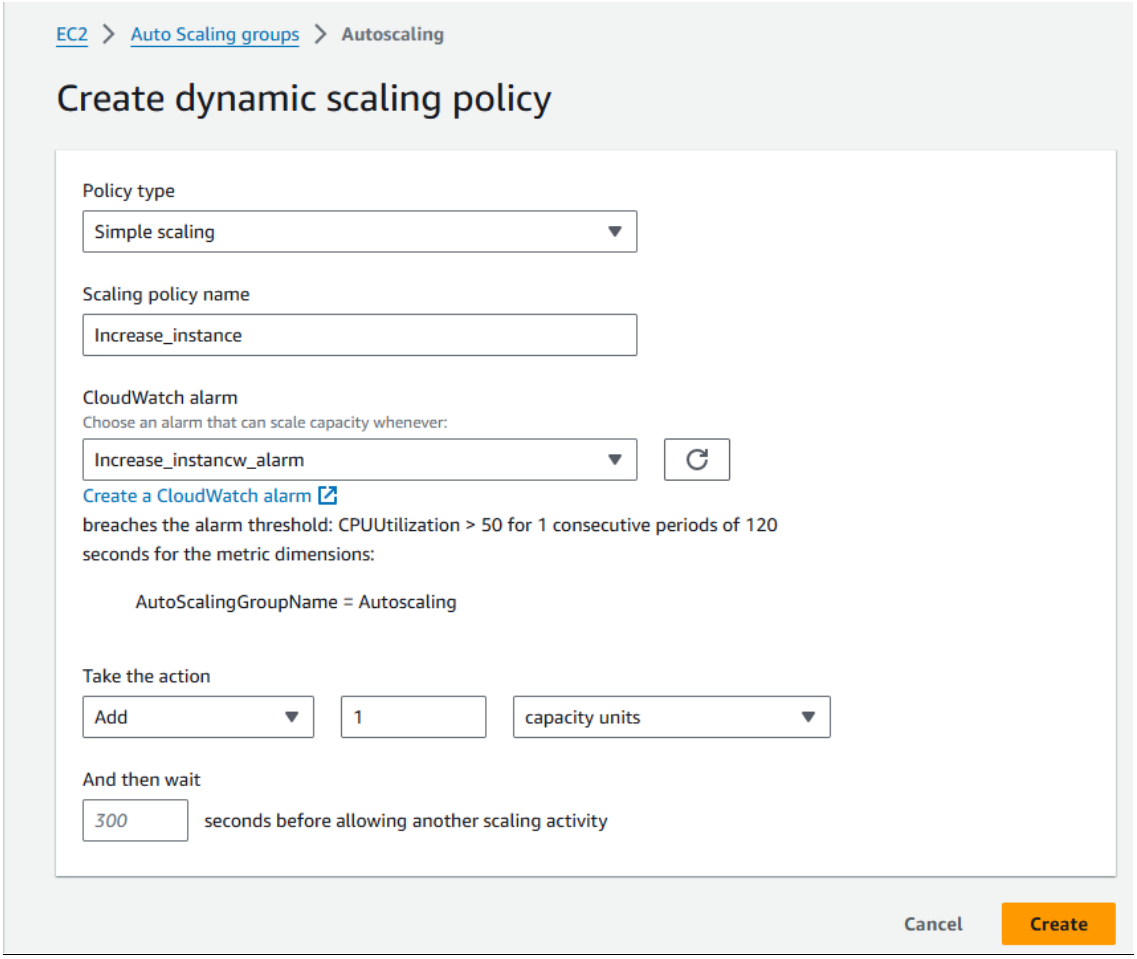
2) Select in which state you want to trigger “In alarm” and add SNS topic



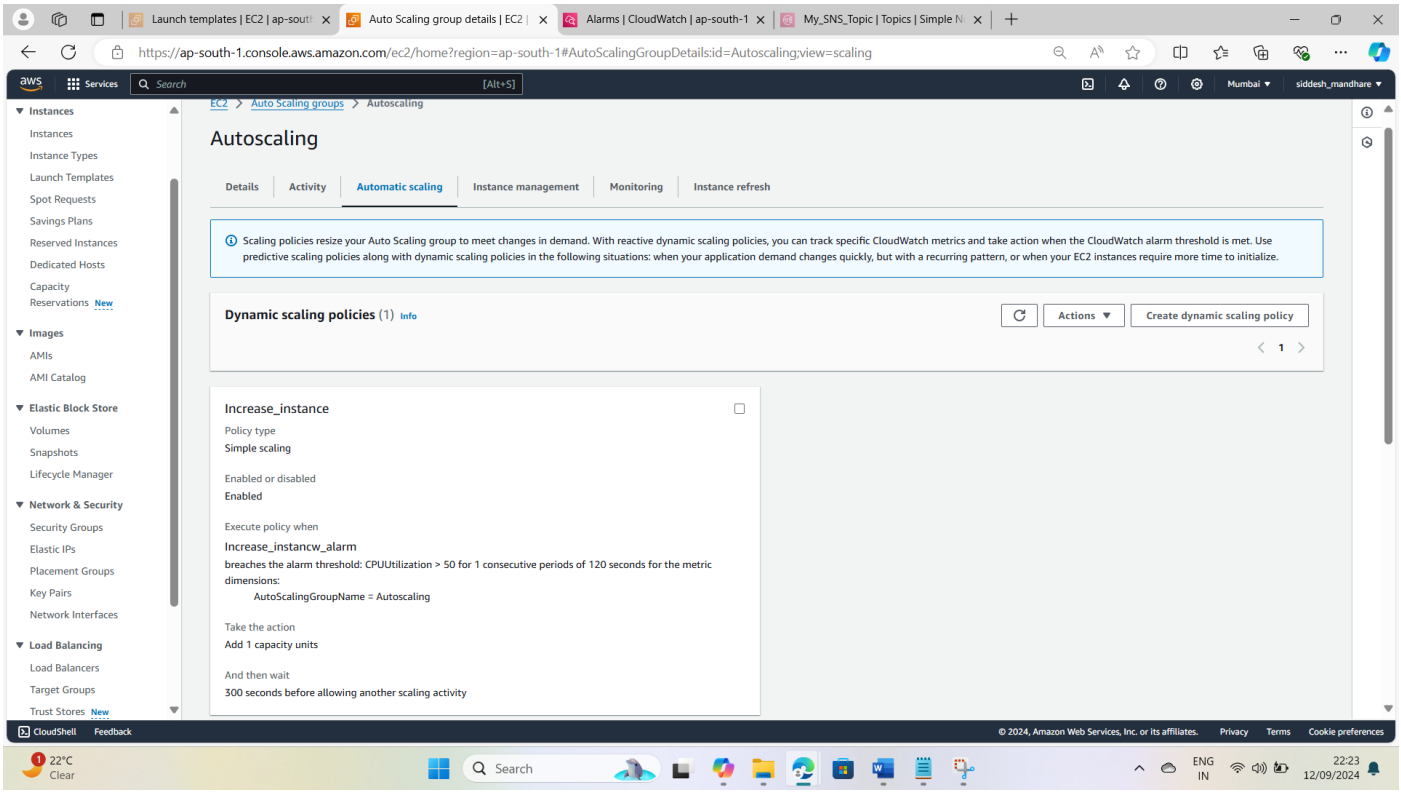
3) Alarm is created successfully



4) Create Dynamic Policy with alarm in take action added count as “1”. It means if utilization is goes above threshold i.e. 50% it should add one more instance.

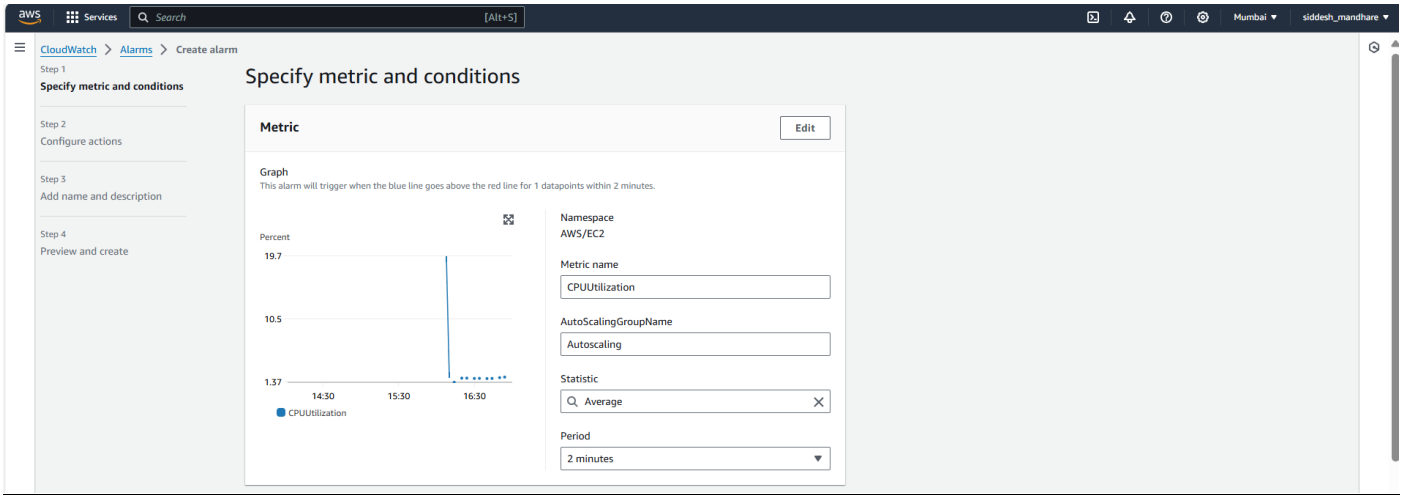


5) Created Dynamic policy as “Increase_instance”



Step 2: - Create Alarm for one more dynamic policy we need to create one more alarm as “Decrease_instance” Now condition is (if CPU utilization is below 20% it should trigger the alarm)

1) Create alarm



Conditions

Threshold type

- (●) **Static** — Use a value as a threshold
- ( ) **Anomaly detection** — Use a band as a threshold

Whenever CPUUtilization is...

Define the alarm condition.

- ( ) **Greater** — > threshold
- ( ) **Greater/Equal** — >= threshold
- ( ) **Lower/Equal** — <= threshold
- (●) **Lower** — < threshold

than...

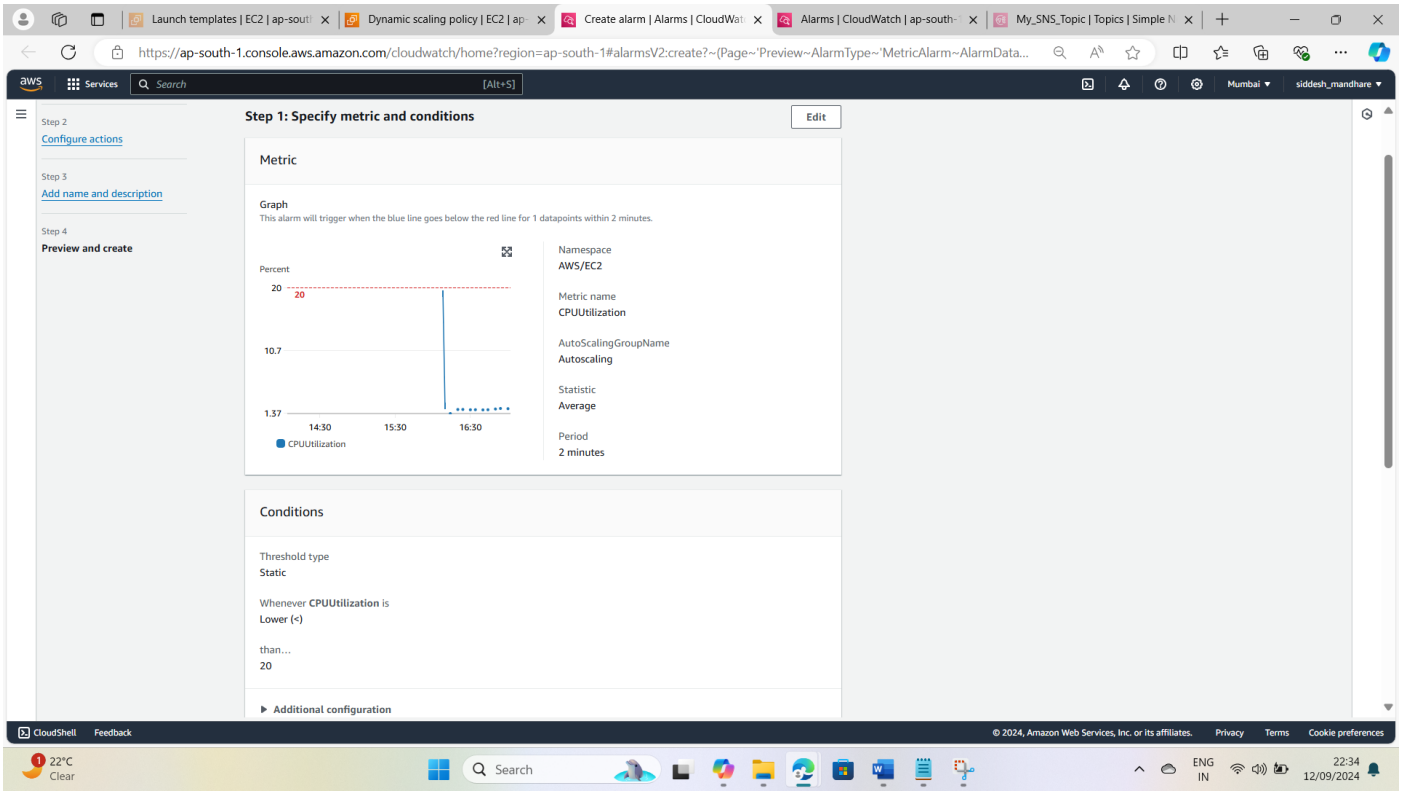
Define the threshold value.

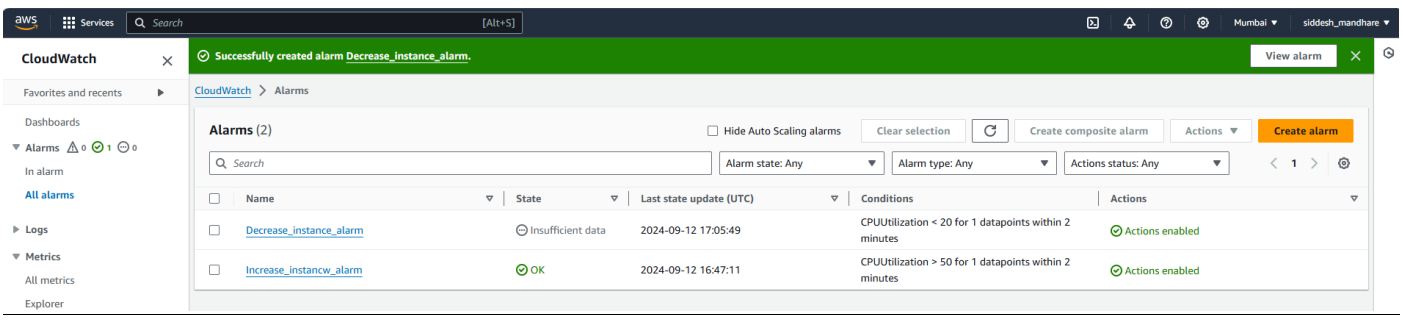
20

Must be a number

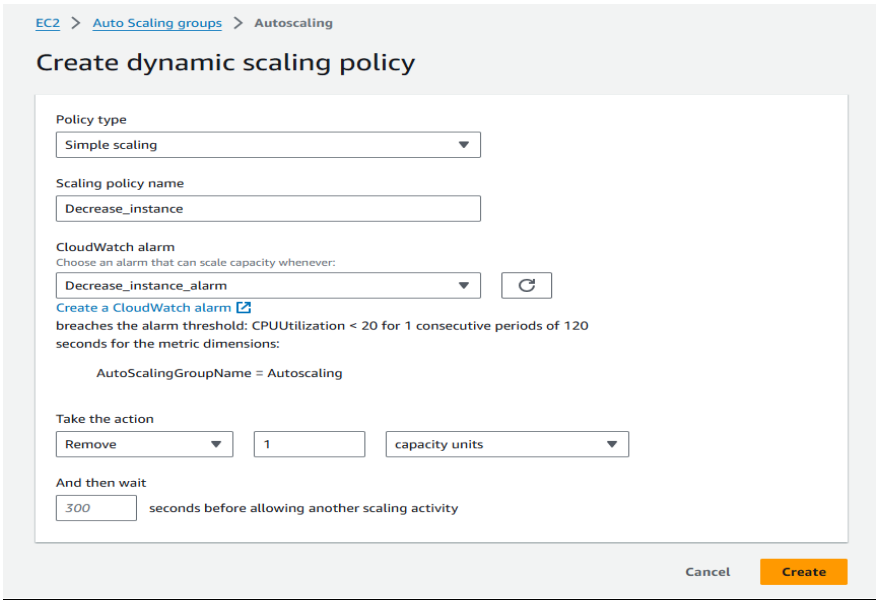
▶ **Additional configuration**

2) Alarm is created

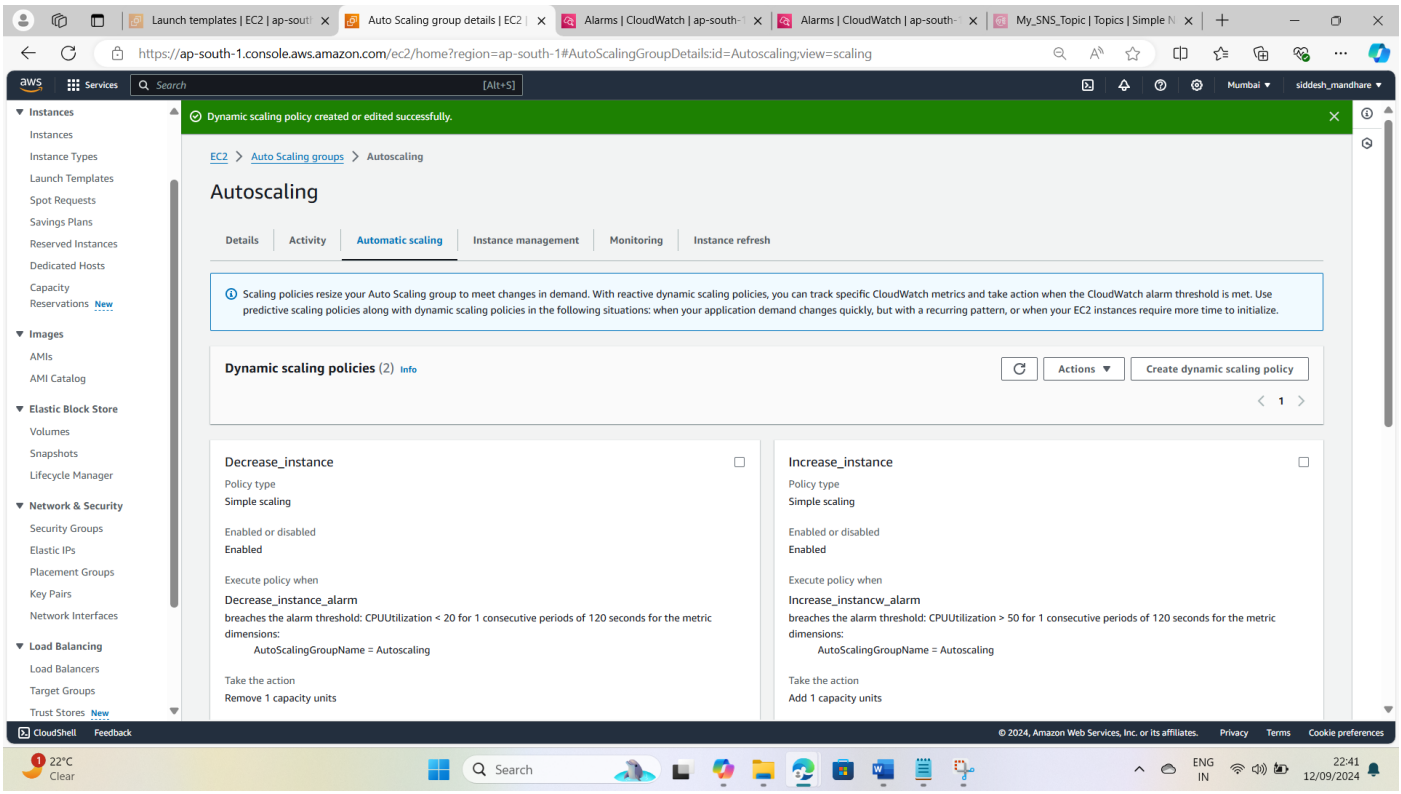




Step 3: - Create policy as “decrease_instance” in take action select “Remove” and count as “1” It means if utilization is below its threshold i.e. 20% it should remove one instance

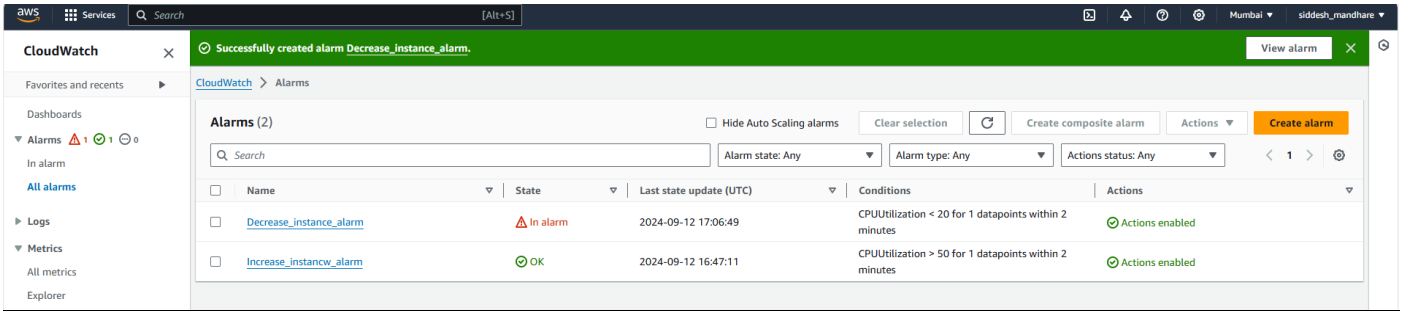


- We have added two policies successfully and both are enabled.

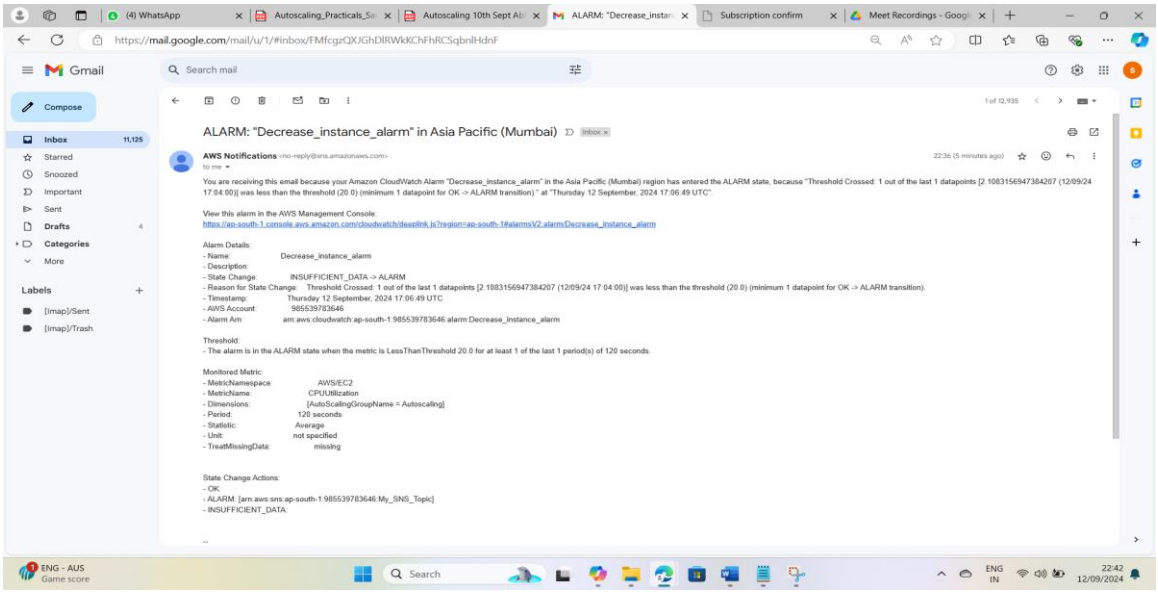


Step 4: -Output

We could see one alarm has been triggered “decrease_instance_alarm”. As checked average CPU utilization of instance is below 20% only.



❖ We got notification mail



Summary

- **Automatic Resource Management:** AWS Auto Scaling dynamically adjusts the number of EC2 instances or other resources in your application based on current demand. This ensures that you always have the right amount of resources to handle the load.
- **Improved Availability:** By automatically scaling up during high demand and scaling down during low demand, AWS Auto Scaling helps maintain the availability and performance of your application. This minimizes the risk of downtime and ensures a smooth user experience.
- **Cost Efficiency:** AWS Auto Scaling optimizes costs by automatically adjusting resources to match the current demand. This means you only pay for the resources you actually need, avoiding over-provisioning and reducing unnecessary expenses.
- **Customizable Policies:** You can define scaling policies based on specific metrics such as CPU utilization, memory usage, or custom CloudWatch metrics. This allows you to tailor the scaling behavior to the unique needs of your application.
- **Integration:** AWS Auto Scaling integrates seamlessly with other AWS services like EC2, ECS, and RDS. This makes it easy to manage and scale a wide range of resources within your AWS environment.
- **Health Monitoring:** AWS Auto Scaling continuously monitors the health of your instances. If an instance becomes unhealthy, it is automatically replaced with a new one, ensuring that your application remains reliable and resilient.
- **Scalability:** AWS Auto Scaling supports both horizontal scaling (adding more instances) and vertical scaling (increasing the size of instances). This flexibility allows you to efficiently manage resources as your application grows.
- **Predictive Scaling:** AWS Auto Scaling can use machine learning to predict future traffic patterns and scale your resources proactively. This helps ensure that your application is prepared for upcoming demand spikes.
- **Easy Setup and Management:** Setting up AWS Auto Scaling is straightforward, and the service provides a user-friendly interface for managing scaling policies and monitoring resource usage.