AUTO SCALING

Auto Scaling in AWS is a service that automatically adjusts the number of EC2 instances or other AWS resources based on demand, ensuring high availability and cost optimization. It helps maintain application performance by dynamically adding or removing instances according to real-time traffic, workload, or other defined policies.

Auto Scaling is a cloud computing feature that allows an application to automatically adjust its resources (like compute instances) based on real-time demand. This ensures that the application maintains performance and availability while optimizing cost-efficiency.

Types of Auto Scaling:

1. **Vertical Auto Scaling (Scaling Up/Down)**:
    - This involves adding more resources (CPU, RAM, storage) to an existing instance to meet demand, or reducing resources when demand decreases.
    - It's limited to scaling within the capabilities of a single instance (e.g., upgrading an instance type to a larger one).
2. **Horizontal Auto Scaling (Scaling In/Out)**:
    - This is the more common form of auto scaling and involves adding or removing instances from a pool based on demand.
    - **Scaling Out**: Adding more instances when the demand increases (e.g., handling more traffic).
    - **Scaling In**: Removing instances when the demand decreases (to save cost).
3. **Scheduled Auto Scaling**:
    - Allows you to set up scaling actions based on a predefined schedule (e.g., scaling out at certain hours of the day when traffic is expected to be high).
    - Useful for predictable workloads, like when traffic increases during specific hours or events.

4. **Dynamic Auto Scaling**:
    - Adjusts resources based on real-time metrics such as CPU utilization, request latency, or memory usage.
    - It reacts to changes in traffic or load automatically and continuously, without predefined schedules.
5. **Predictive Auto Scaling**:
    - Uses machine learning algorithms to predict future traffic patterns and adjust resources ahead of time to ensure performance.
    - Ideal for workloads with complex or cyclical traffic, as it anticipates the needs based on historical data.

Key Components of AWS Auto Scaling

1. **Launch Template/Configuration** – Defines the instance type, AMI, security group, and other settings.
2. **Auto Scaling Group (ASG)** – A collection of EC2 instances that are managed together.
3. **Scaling Policies** – Rules that determine when to add or remove instances:
    - **Target Tracking** – Keeps a metric (e.g., CPU utilization) at a target level.
    - **Step Scaling** – Adjusts capacity in steps (e.g., add 2 instances if CPU > 70%).
    - **Simple Scaling** – Adds/removes instances based on predefined conditions.
    - **Scheduled Scaling** – Increases/decreases instances at specific times.

Benefits of AWS Auto Scaling

✔ **High Availability** – Ensures the application remains available by maintaining the right number of instances.
✔ **Cost Optimization** – Reduces costs by removing unnecessary instances when demand is low.
✔ **Performance Efficiency** – Scales up resources to handle traffic spikes, preventing slowdowns.
✔ **Flexibility** – Works across multiple AWS services beyond EC2.

Configure Load Balancer with Auto Scaling and Creating Launch Template

Step 1: Sign in to AWS Console

Step 2: Navigate to **EC2 Dashboard** > scroll down and click **"Auto Scaling Groups"** under the **"Auto Scaling"** section.

Step 3: Create an Auto Scaling Group (ASG): Click the "Create Auto Scaling group" button > Enter a name in Auto Scaling Group

Step 4: Choose a Launch Template or Configuration

1. **Select a Launch Template** (recommended) or create a new one:
    - If you don't have a launch template, click **"Create a launch template"** and define:
        - Amazon Machine Image (AMI)
        - Instance type (t2.micro)
        - Key pair (for SSH access)
        - Security group (firewall rules) - HTTP & ssh > Click **Next**.

Step 5: Configure Auto Scaling Options

1. **Choose a VPC and Subnets** (for instance placement).
2. **Enable Load Balancer :**
    - **Application Load Balancer (ALB)**, can attach existing load balancer.

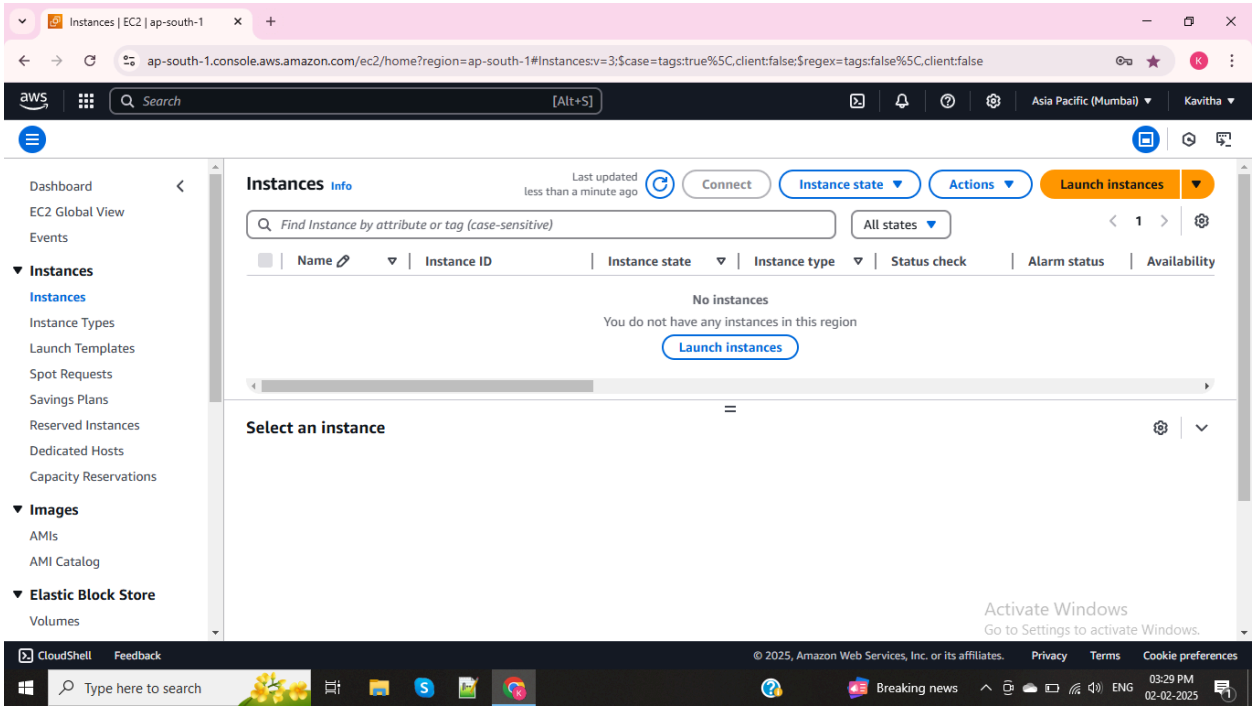
Step 6: Define Scaling Policies

1. Select **Scaling policy type**:
2. Define the minimum and maximum number of instances.
3. Click **Next**.

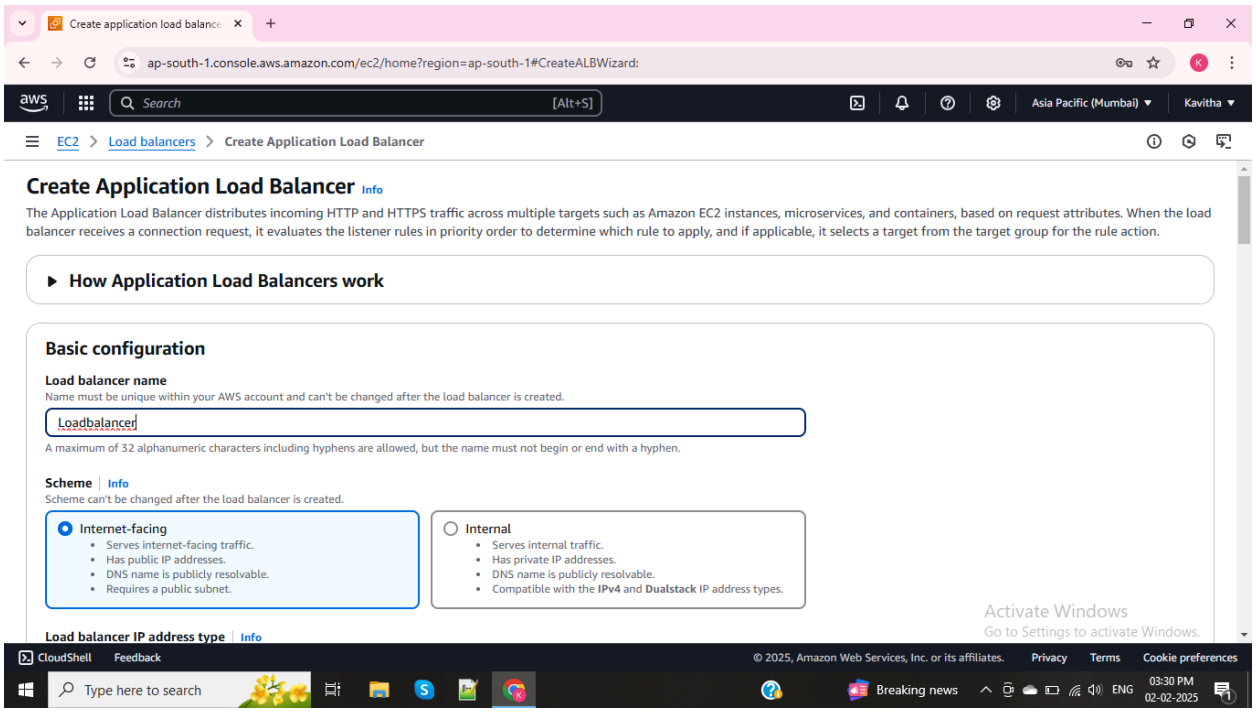
Step 7: Review and Create the Auto Scaling Group

1. Review all configurations.
2. Click **"Create Auto Scaling group"**.

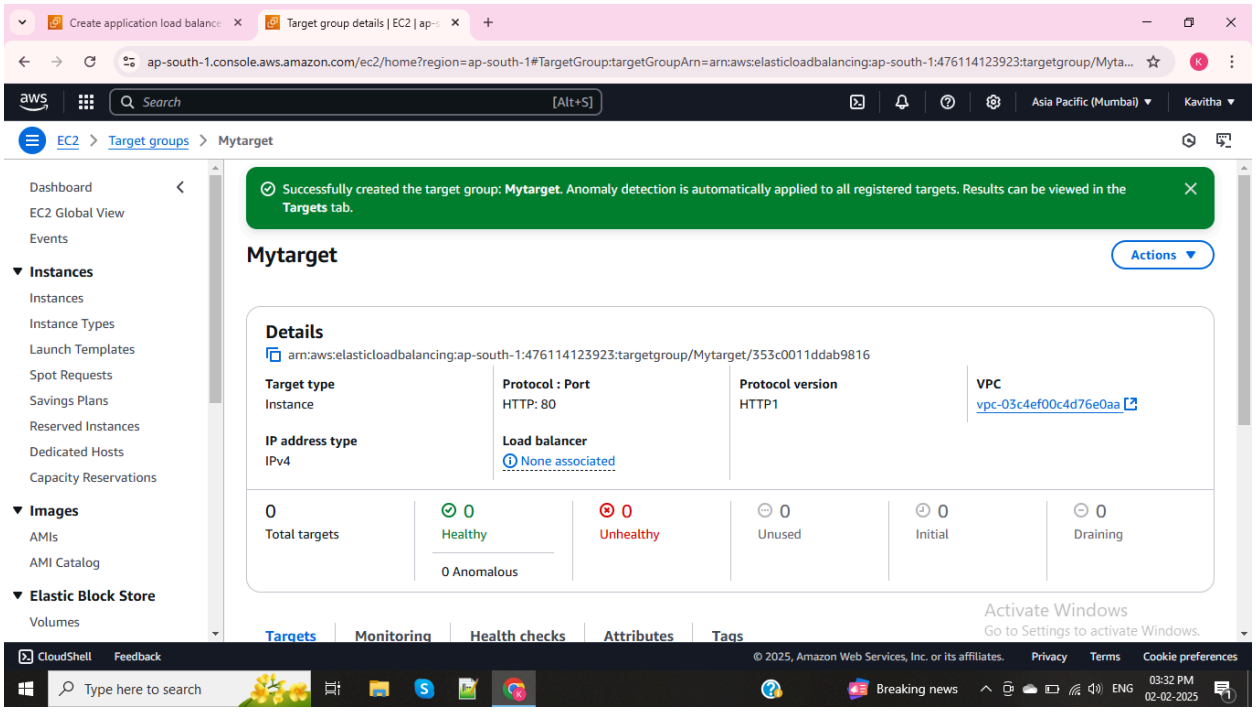
No Instance in EC2 Instance:



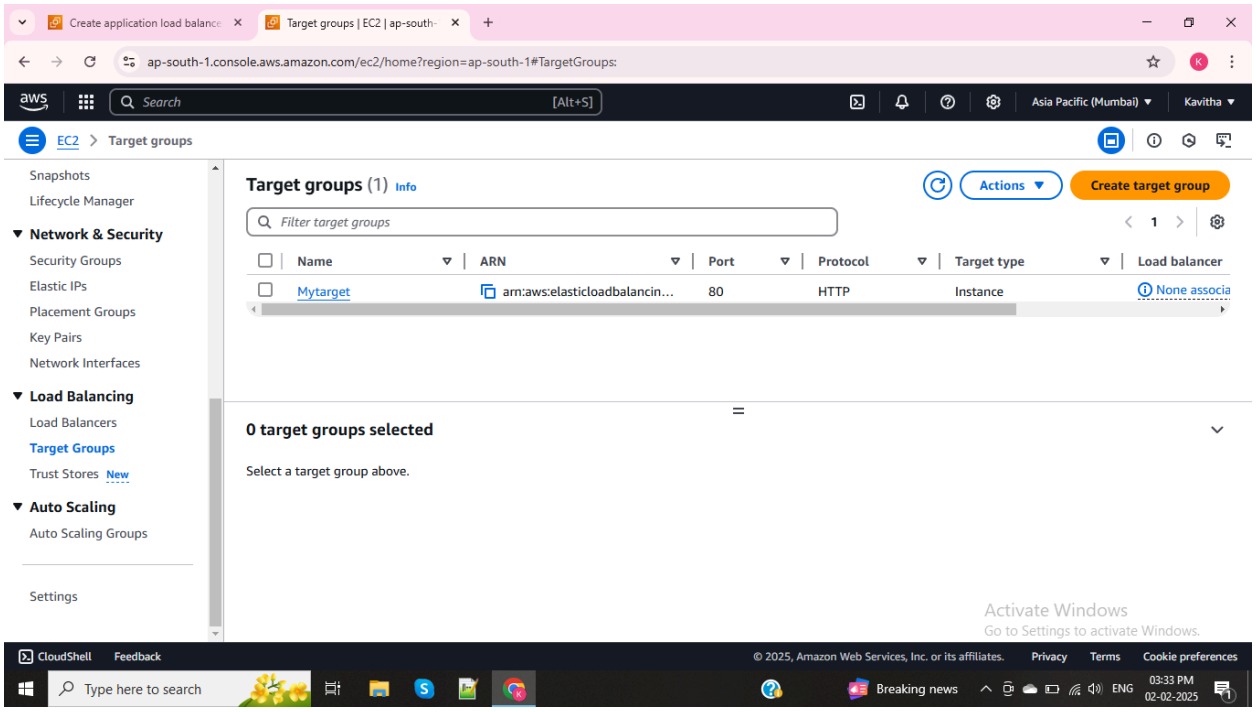
Creating Application LoadBalancer to attach with Auto scaling group



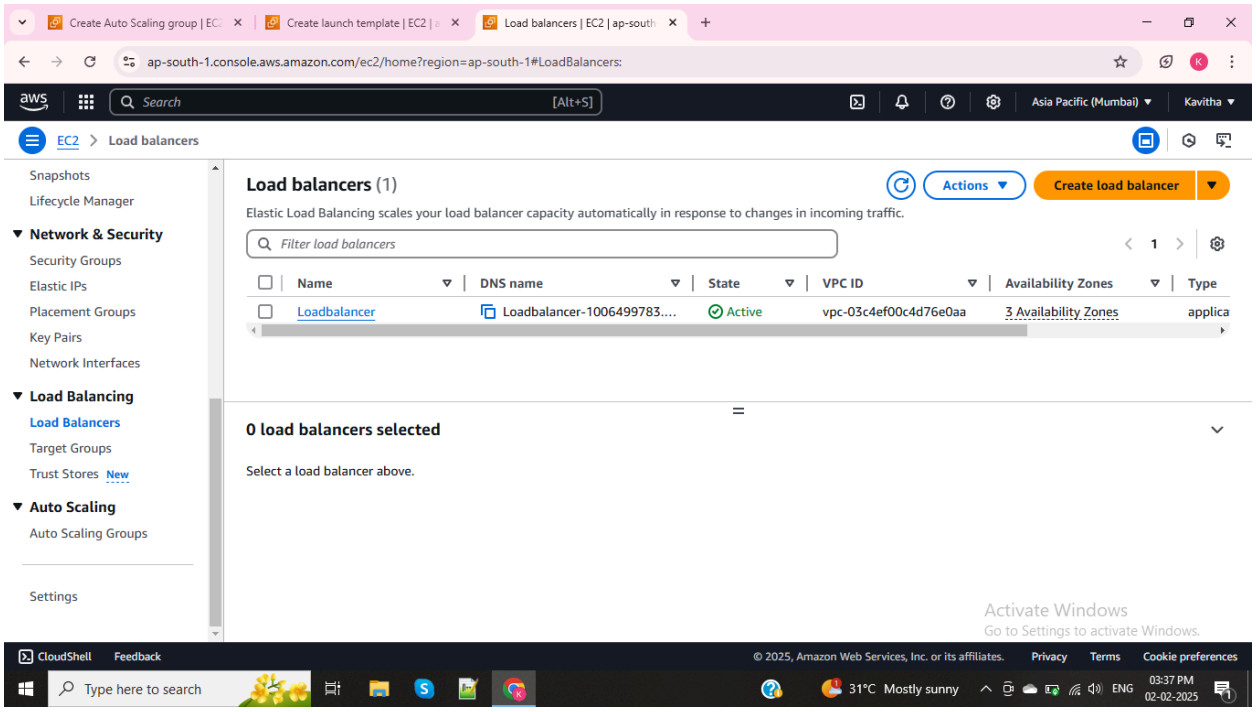
Creating Target group



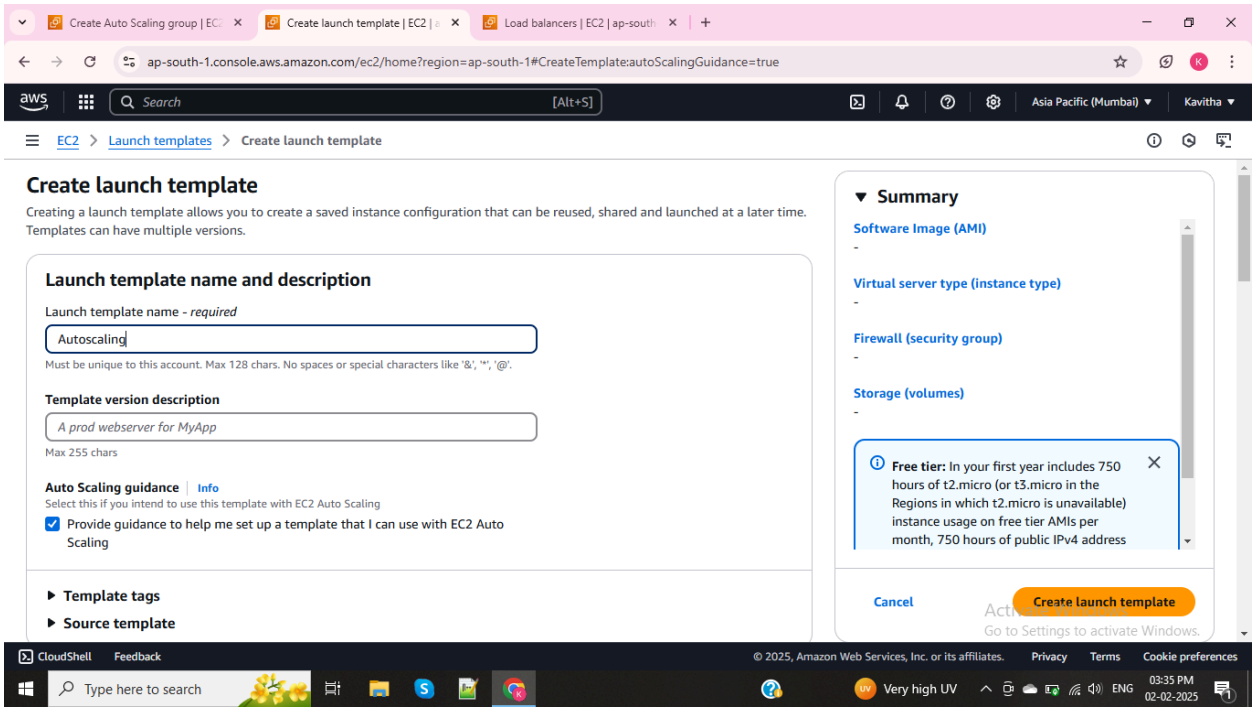
OUTPUT



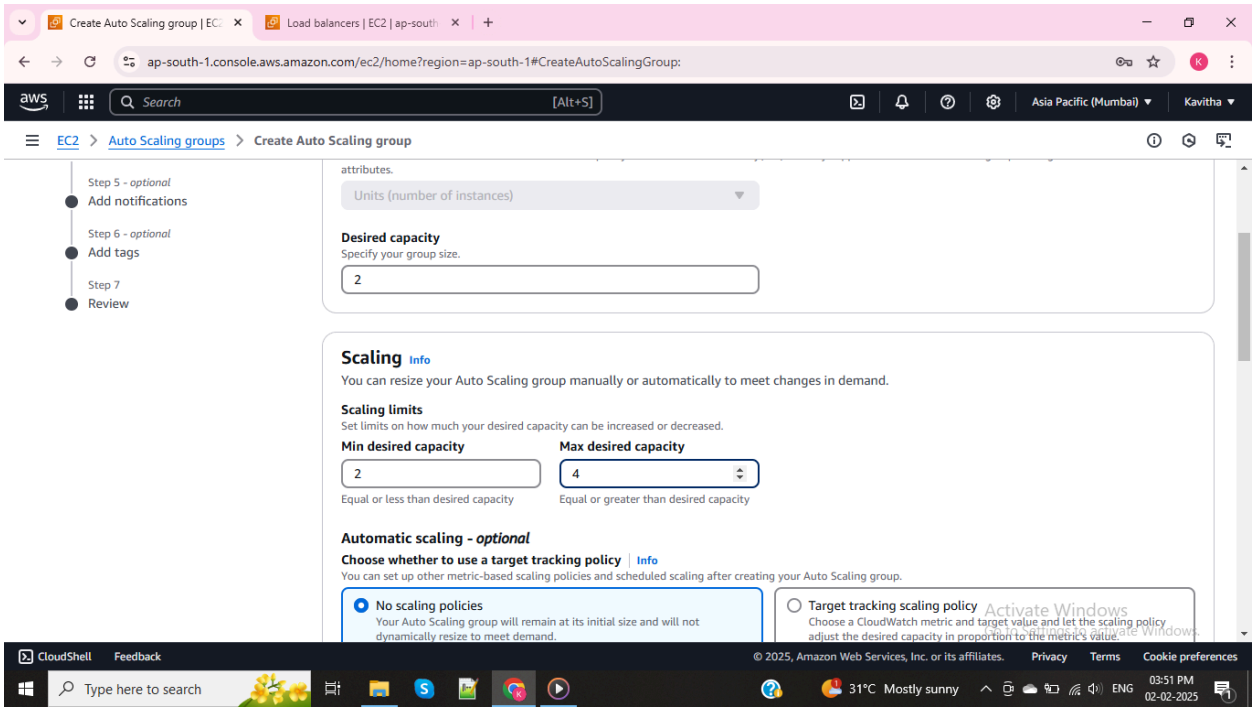
Load Balancer is created successfully



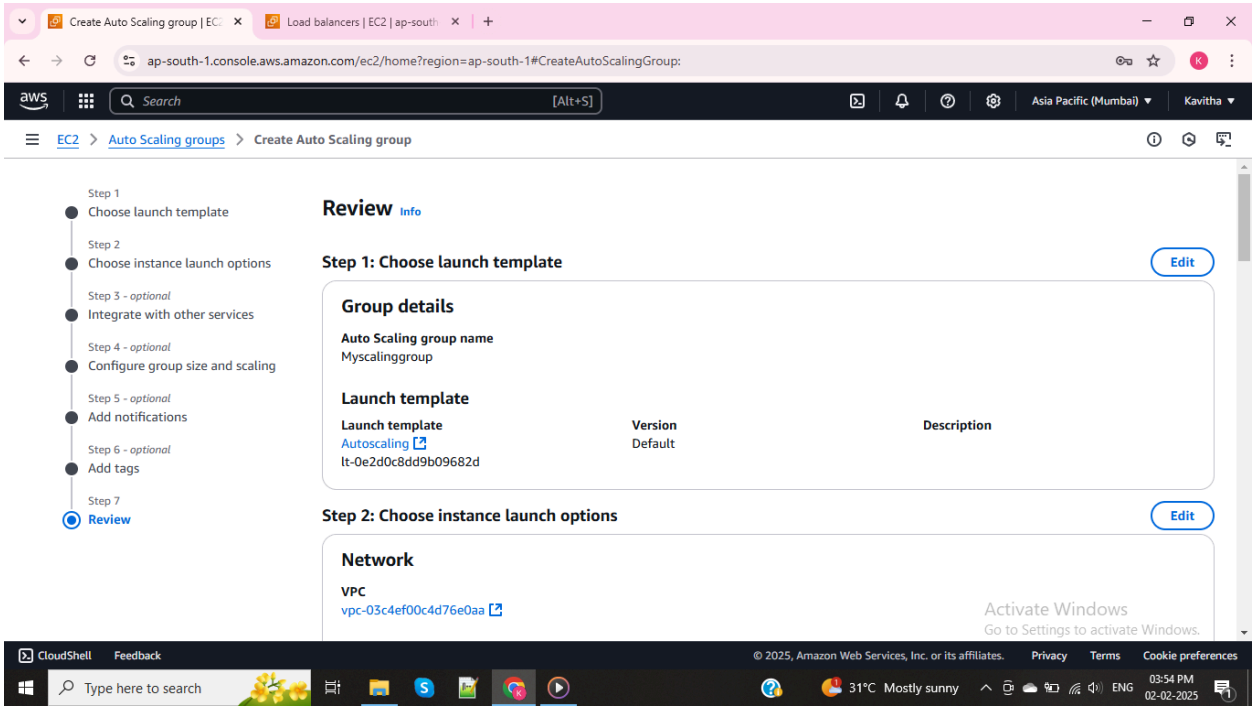
Creating Launch template

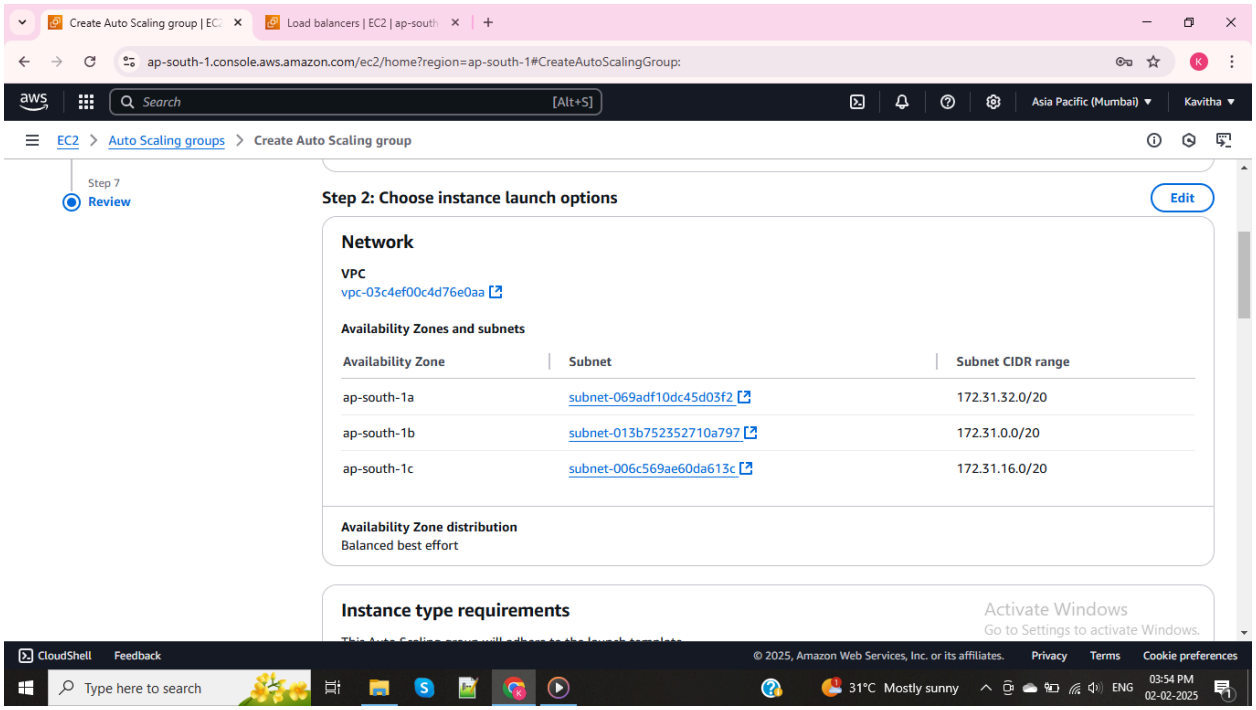


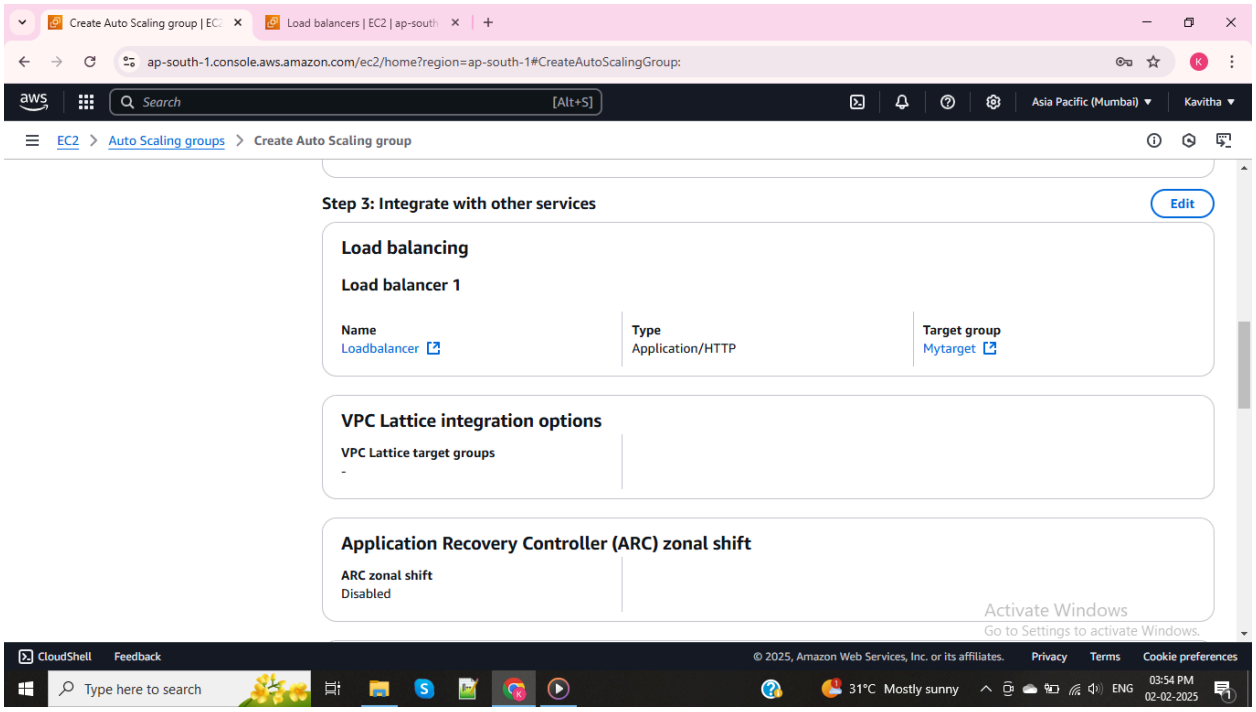
Configure group size and scaling policies

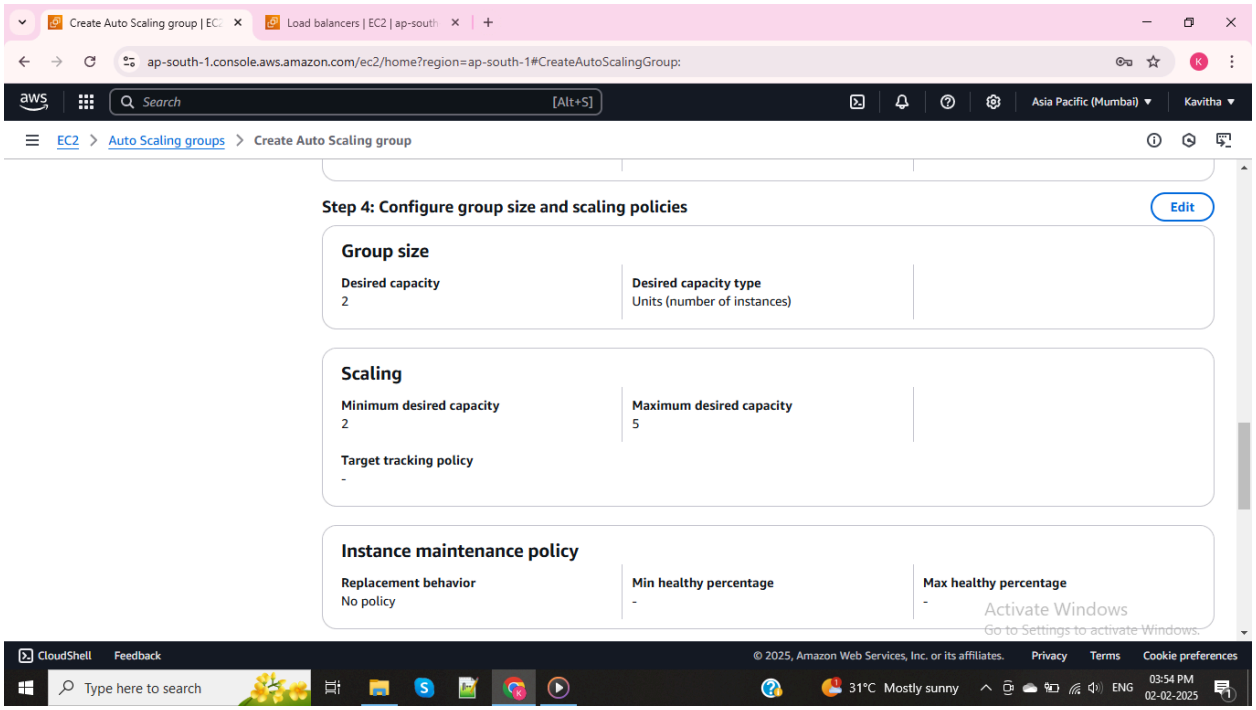


Reviewing all groups before created Auto Scaling

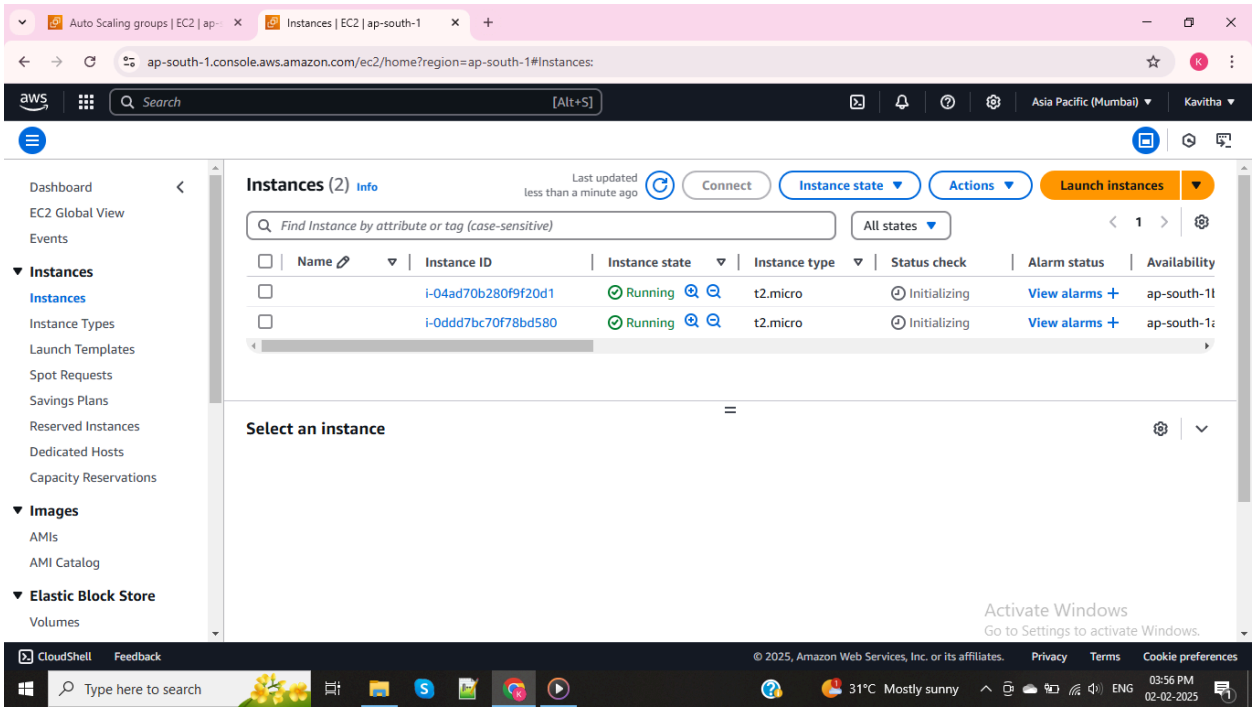



Create Auto Scaling group | EC2
Load balancers | EC2 | ap-south-1
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:
Search [Alt+S]
Asia Pacific (Mumbai)
Kavitha
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 7
Review
Step 2: Choose instance launch options
Edit
Network
VPC
vpc-03c4ef00c4d76e0aa
Availability Zones and subnets
Availability Zone | Subnet | Subnet CIDR range
ap-south-1a | subnet-069adf10dc45d03f2 | 172.31.32.0/20
ap-south-1b | subnet-013b752352710a797 | 172.31.0.0/20
ap-south-1c | subnet-006c569ae60da613c | 172.31.16.0/20
Availability Zone distribution
Balanced best effort
Instance type requirements
Activate Windows
Go to Settings to activate Windows.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly sunny
ENG
03:54 PM
02-02-2025

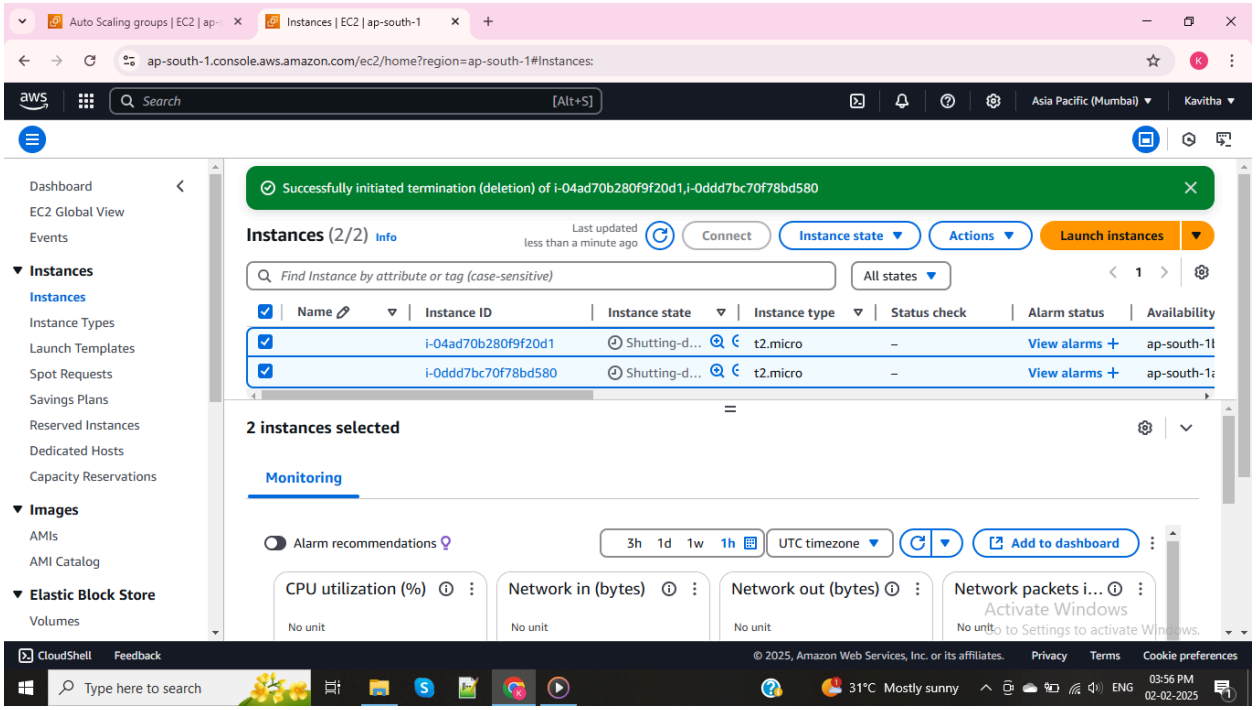

Create Auto Scaling group | EC2
Load balancers | EC2 | ap-south-1
ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateAutoScalingGroup:
Search [Alt+S]
Asia Pacific (Mumbai)
Kavitha
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 3: Integrate with other services
Edit
Load balancing
Load balancer 1
Name
Loadbalancer
Type
Application/HTTP
Target group
Mytarget
VPC Lattice integration options
VPC Lattice target groups
-
Application Recovery Controller (ARC) zonal shift
ARC zonal shift
Disabled
Activate Windows
Go to Settings to activate Windows.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
Type here to search
31°C Mostly sunny
ENG
03:54 PM
02-02-2025



EC2 Instance launch successfully with the help of Launch template



Now, EC2 Instance terminating whether instance launch again with minimum desired count or not



After Instance terminated again automatically its created new two EC2 Instance (In scaling I give minimum desired capacity is 2)