

Case Study

Problem Statement:

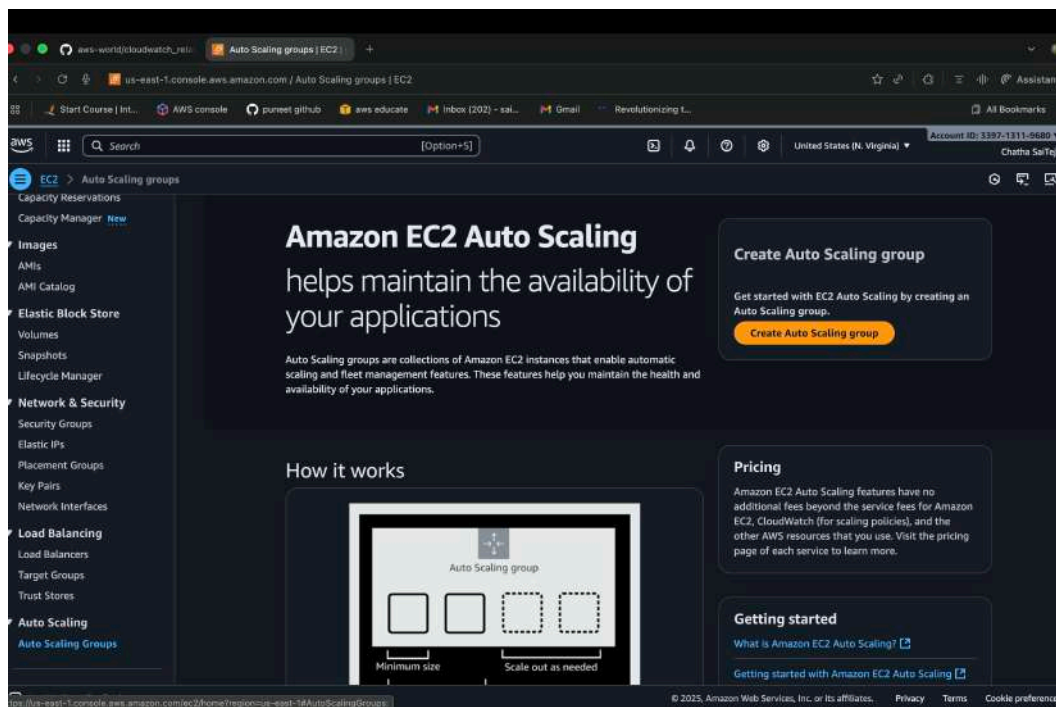
You work for XYZ Corporation that uses on premise solutions and a limited number of systems. With the increase in requests in their application, the load also increases. So, to handle the load the corporation has to buy more systems almost on a regular basis. realizing the need to cut down the expenses on systems, they decided to move their infrastructure to AWS.

Tasks To Be Performed:

1. Manage the scaling requirements of the company by:
 - a. Deploying multiple compute resources on the cloud as soon as the load increases and the CPU utilization exceeds 80%
 - b. Removing the resources when the CPU utilization goes under 60%
2. Create a load balancer to distribute the load between compute resources.
3. Route the traffic to the company's domain

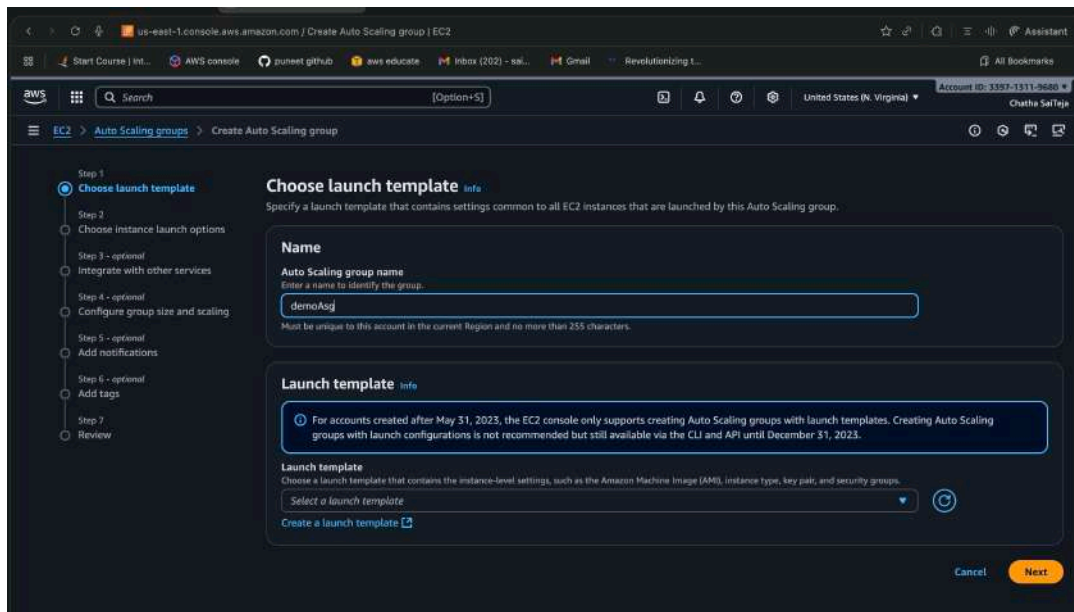
Step-by-Step Procedure:-

Step 1:- To scale the resources we have to use Auto Scaling Group (ASG) service in EC2 Dashboard ,then we can see ASG dashboard ,Click on Create Auto Scaling Group



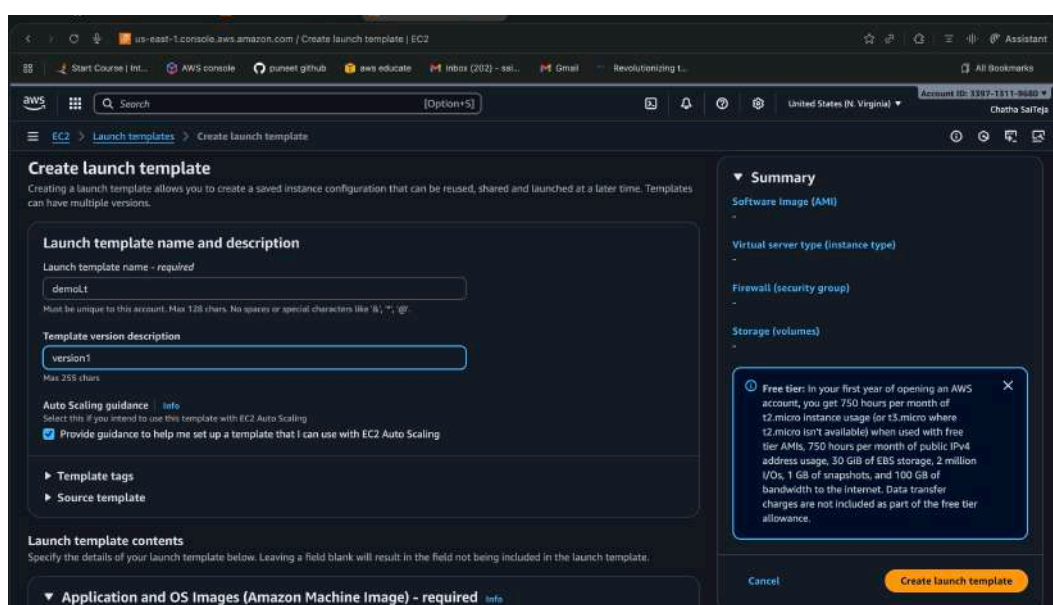
Click on Create Auto Scaling Group

Step 2:- The first step in creating an Auto Scaling Group is to specify a name for the group. Next, you need to create a Launch Template, which defines the configuration of the EC2 instances that the ASG will launch during scale-out or scale-in events. In the Launch Template, you provide settings similar to those used when creating an EC2 instance such as the AMI, instance type, key pair, security groups, and storage configurations. After entering these details, click on Create Launch Template. Once the template is created successfully, select it to associate with your Auto Scaling Group.



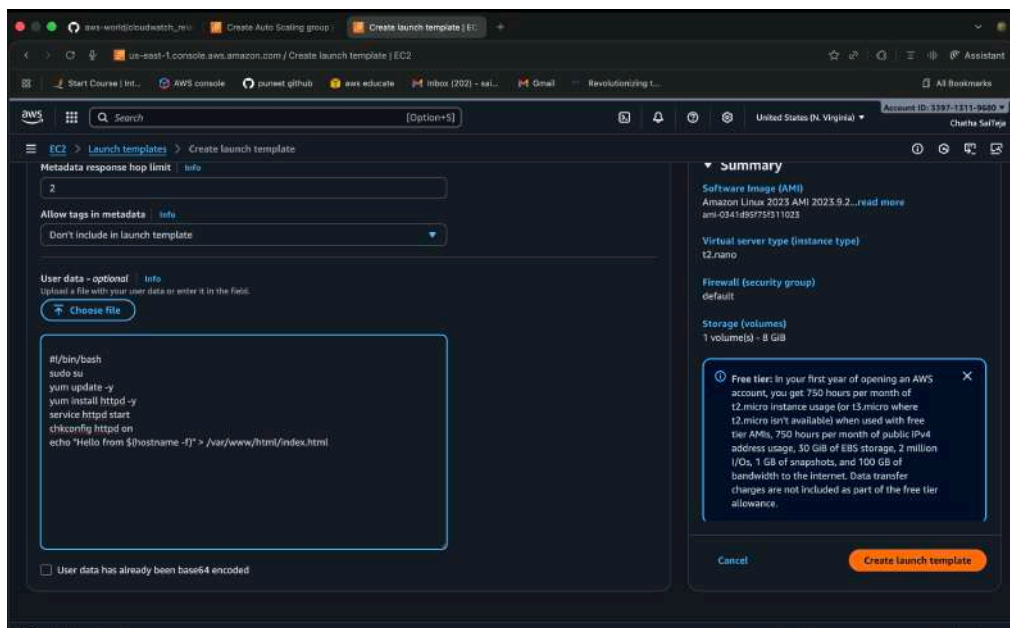
The screenshot shows the AWS Management Console interface for creating an Auto Scaling group. The page is titled 'Create Auto Scaling group | EC2'. On the left, a navigation pane shows a list of steps: Step 1: Choose launch template (selected), Step 2: Choose instance launch options, Step 3 - optional: Integrate with other services, Step 4 - optional: Configure group size and scaling, Step 5 - optional: Add notifications, Step 6 - optional: Add tags, and Step 7: Review. The main content area is titled 'Choose launch template' with a subtitle 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' There is a text input field for 'Name' with the value 'demoAsg' and a note 'Must be unique to this account in the current Region and no more than 255 characters.' Below this is a 'Launch template' section with a dropdown menu set to 'Select a launch template' and a 'Create a launch template' link. At the bottom right are 'Cancel' and 'Next' buttons.

Specify the name(demoAsg) and click on create launch template

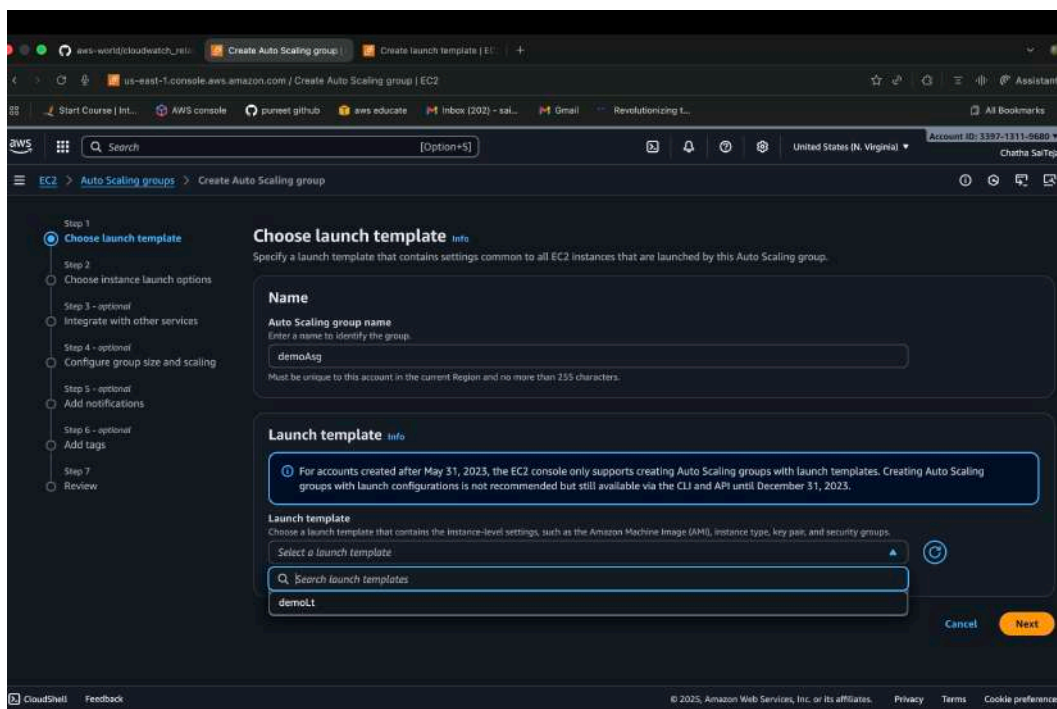


The screenshot shows the AWS Management Console interface for creating a launch template. The page is titled 'Create launch template'. On the left, a navigation pane shows a list of steps: Step 1: Launch template name and description (selected), Step 2: Template version description, Step 3: Auto Scaling guidance, Step 4: Template tags, and Step 5: Source template. The main content area is titled 'Create launch template' with a subtitle 'Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.' There is a text input field for 'Launch template name - required' with the value 'demoLt' and a note 'Must be unique to this account. Max 128 chars. No spaces or special characters like @, *, ?, !, &'. Below this is a 'Template version description' section with a text input field set to 'version1' and a note 'Max 255 chars'. There is also an 'Auto Scaling guidance' section with a checkbox 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling' which is checked. At the bottom are 'Template tags' and 'Source template' sections. On the right, a 'Summary' section lists the configuration details: Software image (AMI), Virtual server type (instance type), Firewall (security group), and Storage (volumes). At the bottom right are 'Cancel' and 'Create launch template' buttons.

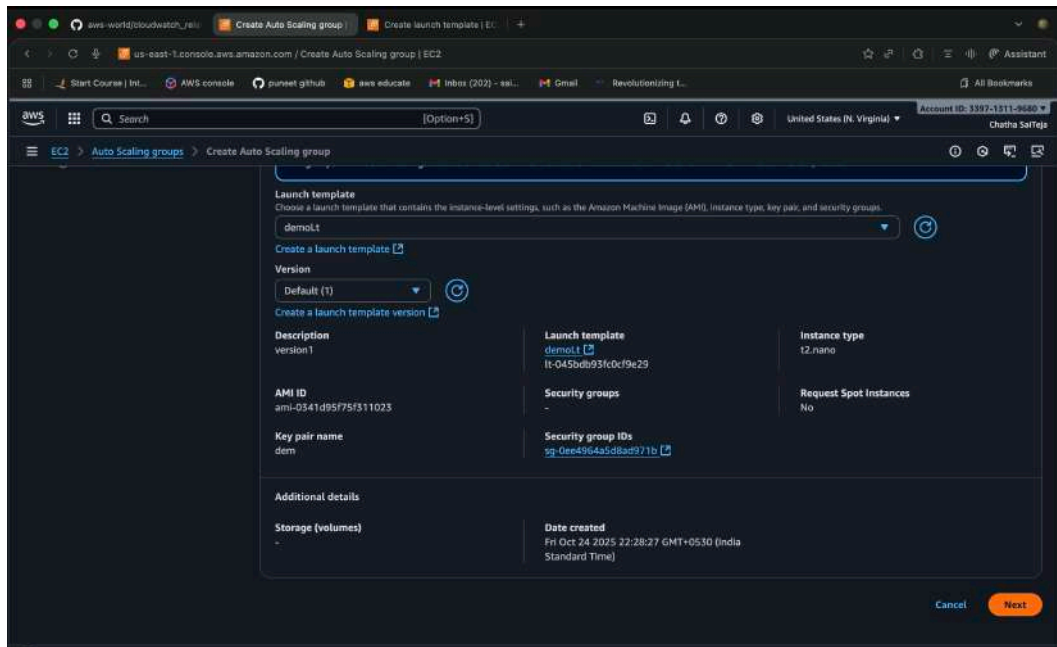
Specify the configurations and name(demoLt) of Launch Template



Click on create launch template

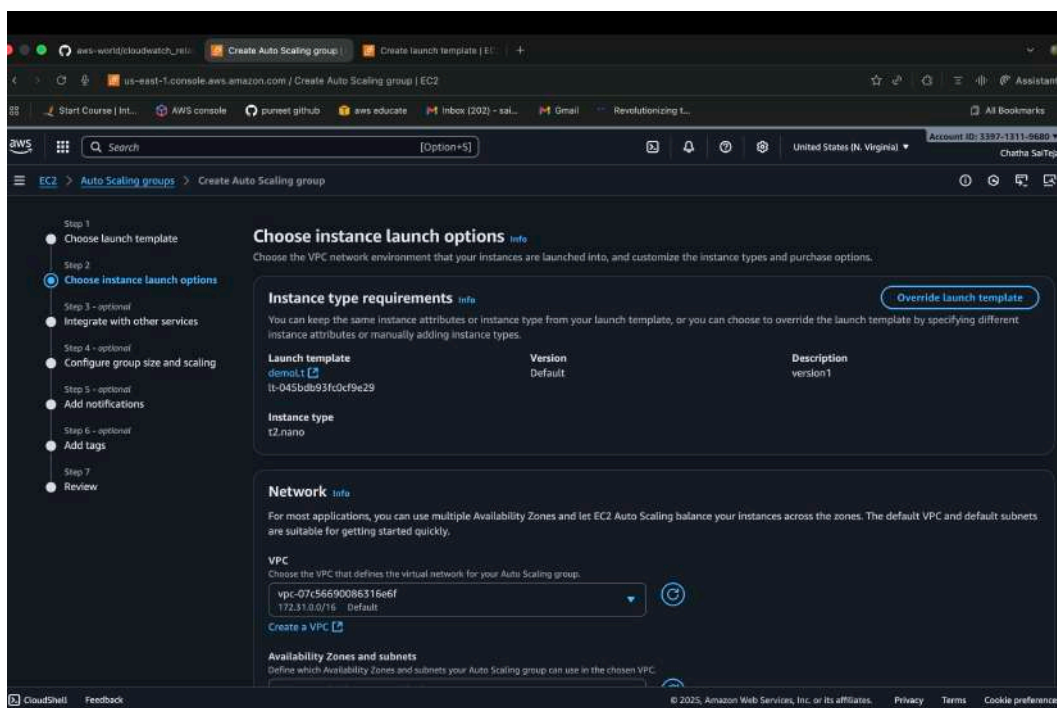


Select the Created Launch Template(demoLt)



Click on Next

Step 3:- we have another steps like choosing instance options, Integrate with other services and configure group and scaling but these are optional so click on next .



Click on next

Step 4:- In next step we have to mention the group size(desired-2, min-1, max-4) and we Can also mention the scaling policies but we don't give any policy.

The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 4: 'Configure group size and scaling - optional'. The left sidebar shows the progress: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional, Integrate with other services), Step 4 (selected), Step 5 (optional, Add notifications), Step 6 (optional, Add tags), and Step 7 (Review). The main content area is titled 'Configure group size and scaling - optional' and includes the following sections:

- Group size:** Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling. The 'Desired capacity type' is set to 'Units (number of instances)'. The 'Desired capacity' is set to '2'.
- Scaling:** You can resize your Auto Scaling group manually or automatically to meet changes in demand. The 'Scaling limits' section shows 'Min desired capacity' set to '1' and 'Max desired capacity' set to '4'. Below these, it says 'Equal or less than desired capacity' and 'Equal or greater than desired capacity' respectively.
- Automatic scaling - optional:** Choose whether to use a target tracking policy. The 'No scaling policies' option is selected, and the 'Target tracking scaling policy' option is unselected.

At the bottom, there are links for 'CloudShell', 'Feedback', and a copyright notice for Amazon Web Services, Inc. or its affiliates, along with links for 'Privacy', 'Terms', and 'Cookie preferences'.

Giving group size

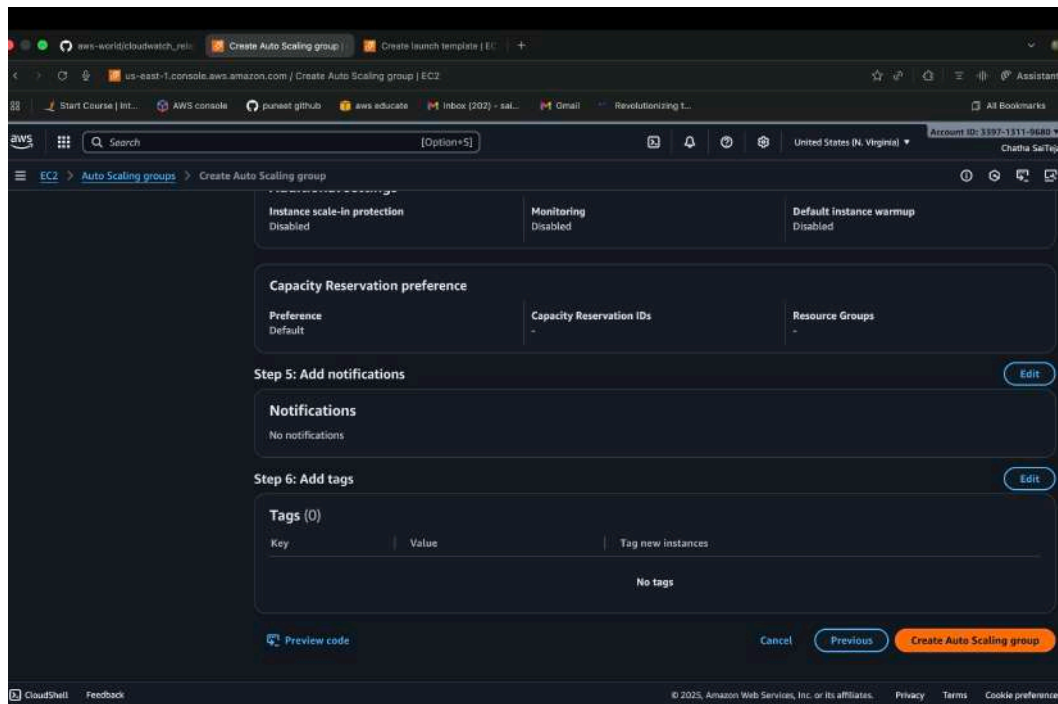
The screenshot shows the 'Create Auto Scaling group' wizard in the AWS Management Console, specifically Step 5: 'Additional capacity settings'. The left sidebar shows the progress: Step 1 (Choose launch template), Step 2 (Choose instance launch options), Step 3 (optional, Integrate with other services), Step 4 (optional, Configure group size and scaling), Step 5 (selected), Step 6 (optional, Add notifications), Step 7 (optional, Add tags), and Step 7 (Review). The main content area is titled 'Additional capacity settings' and includes the following sections:

- Capacity Reservation preference:** Select whether you want Auto Scaling to launch instances into an existing Capacity Reservation or Capacity Reservation resource group. The 'Default' option is selected, which states 'Auto Scaling uses the Capacity Reservation preference from your launch template.' Other options are 'None' (Instances will not be launched into a Capacity Reservation), 'Capacity Reservations only' (Instances will only be launched into a Capacity Reservation. If capacity isn't available, the instances fail to launch), and 'Capacity Reservations first' (Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run in On-Demand capacity).
- Additional settings:** This section includes 'Instance scale-in protection' (If protect from scale in is enabled, newly launched instances will be protected from scale in by default. The 'Enable instance scale-in protection' checkbox is unchecked), 'Monitoring' (The 'Enable group metrics collection within CloudWatch' checkbox is unchecked), and 'Default instance warmup' (The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet. The 'Enable default instance warmup' checkbox is unchecked).

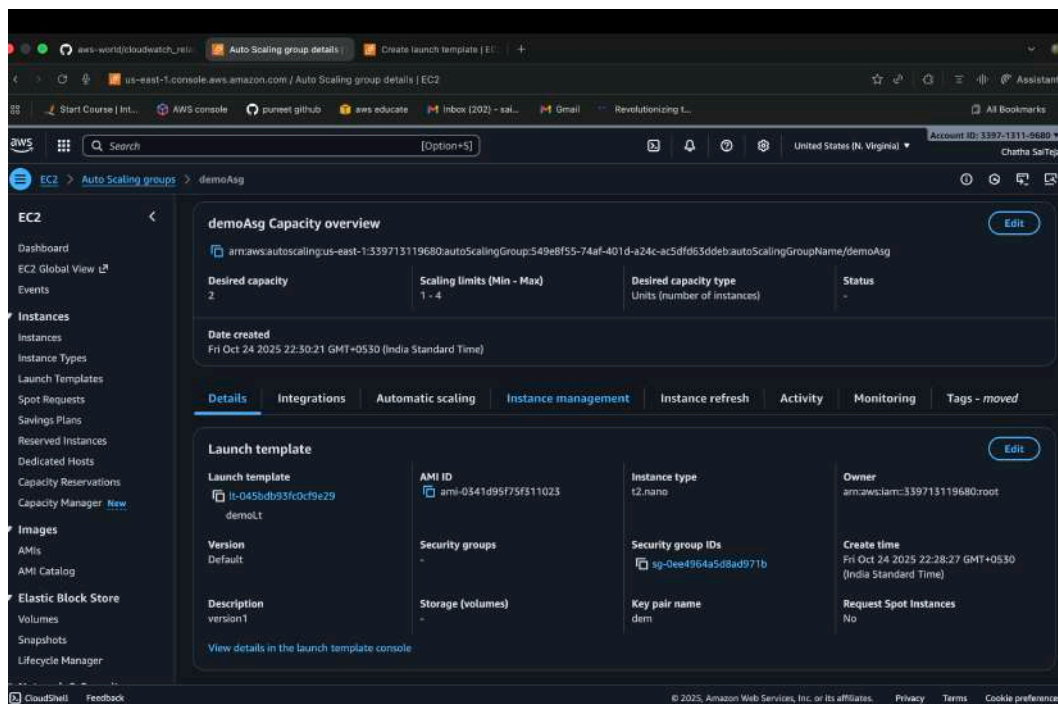
At the bottom, there are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

Click on next

Step 5:- In this Step we can review the total configuration of ASG and click on Create Auto Scaling Group.



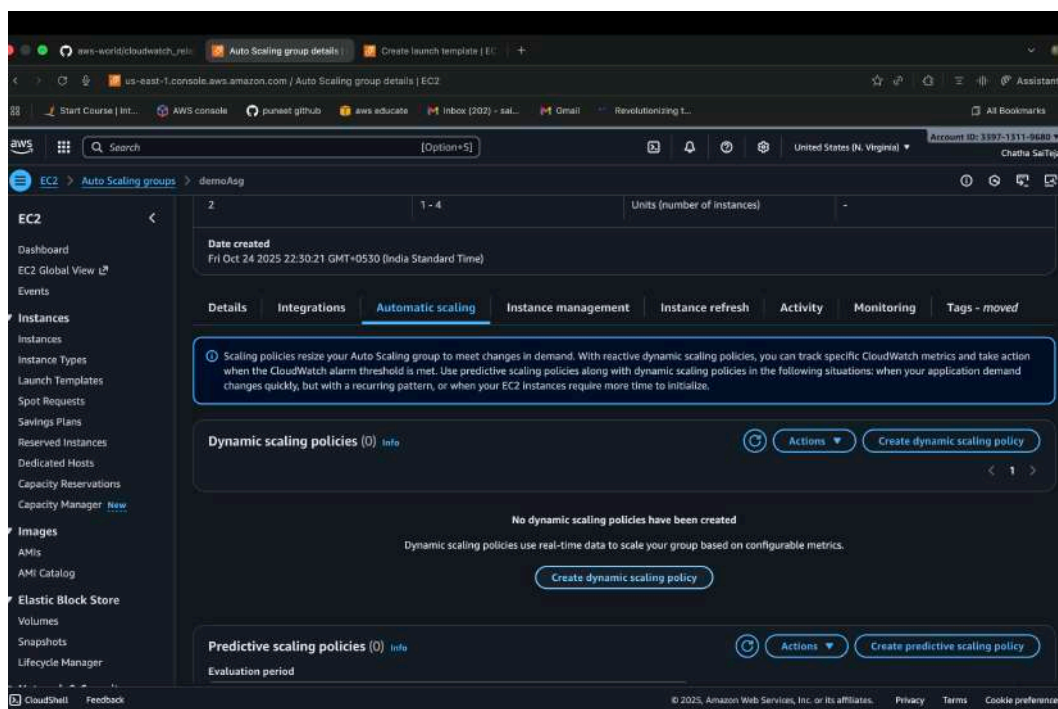
Click on Create Auto Scaling Group.



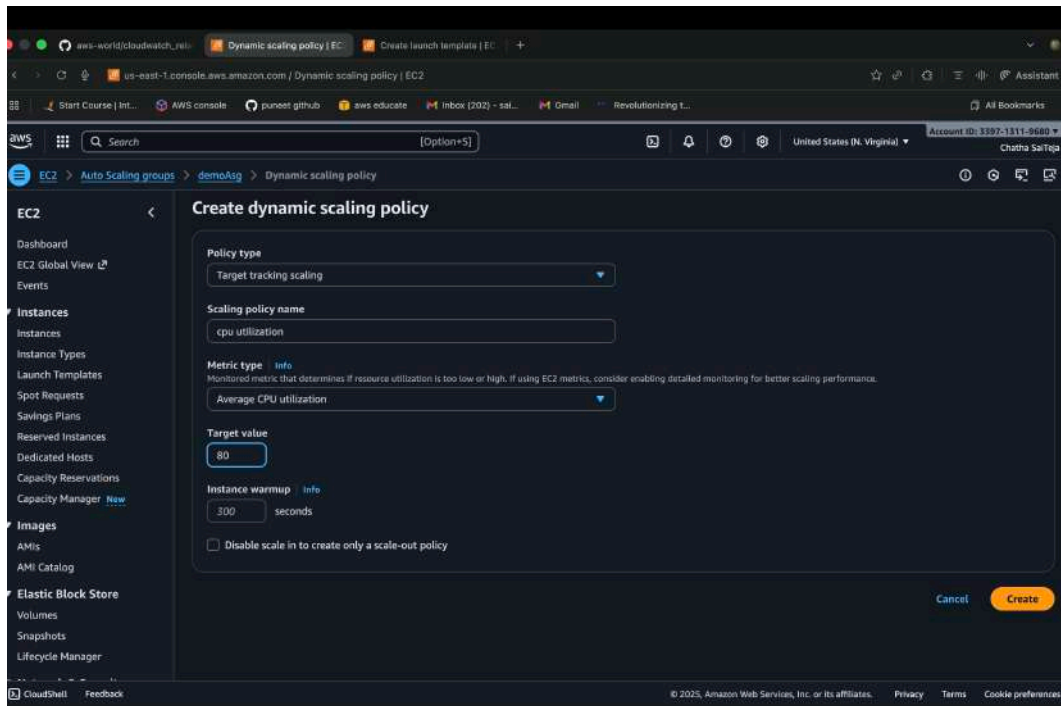
demoAsg Summary

Step 6:- To manage scaling requirements efficiently, Auto Scaling automatically adds or removes compute resources based on demand. You can configure it to add new instances when average CPU utilization reaches 80% and terminate instances when utilization drops below 60%. This behaviour is implemented through Automatic Scaling, which supports three types of dynamic scaling policies: Simple, Step, and Target Tracking.

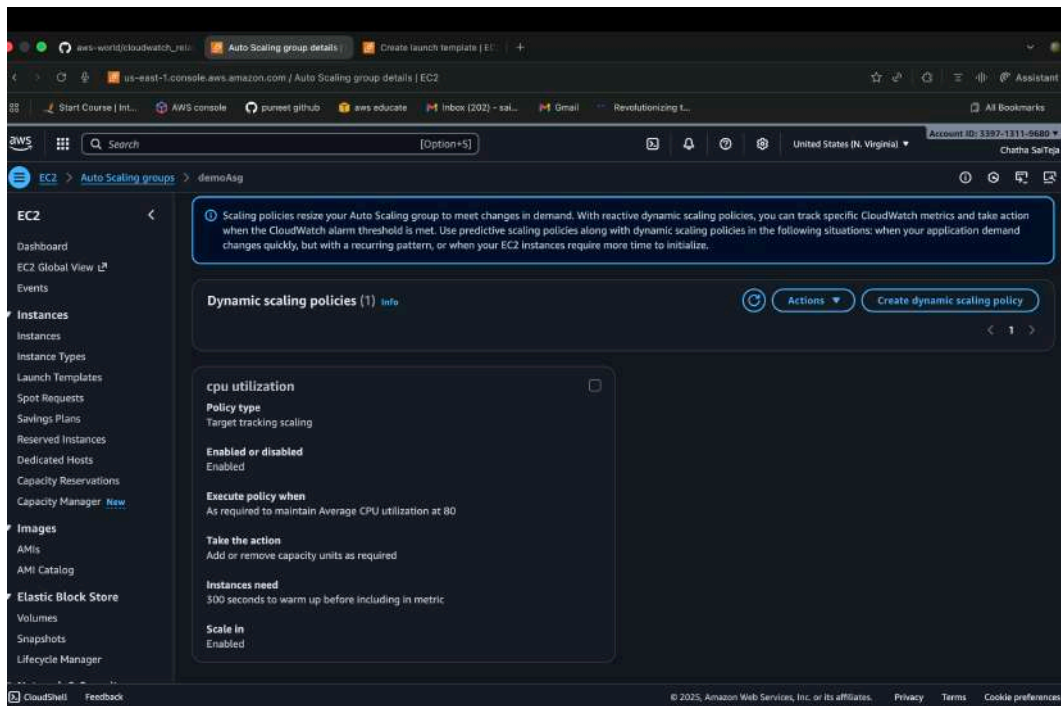
Using a Target Tracking policy, you simply specify a target metric for instance, an Average CPU utilization of 80%. Auto Scaling then automatically adjusts the capacity of your group by launching or terminating instances to maintain this target level.



Click on Automatic Scaling and Select dynamic Scaling policy

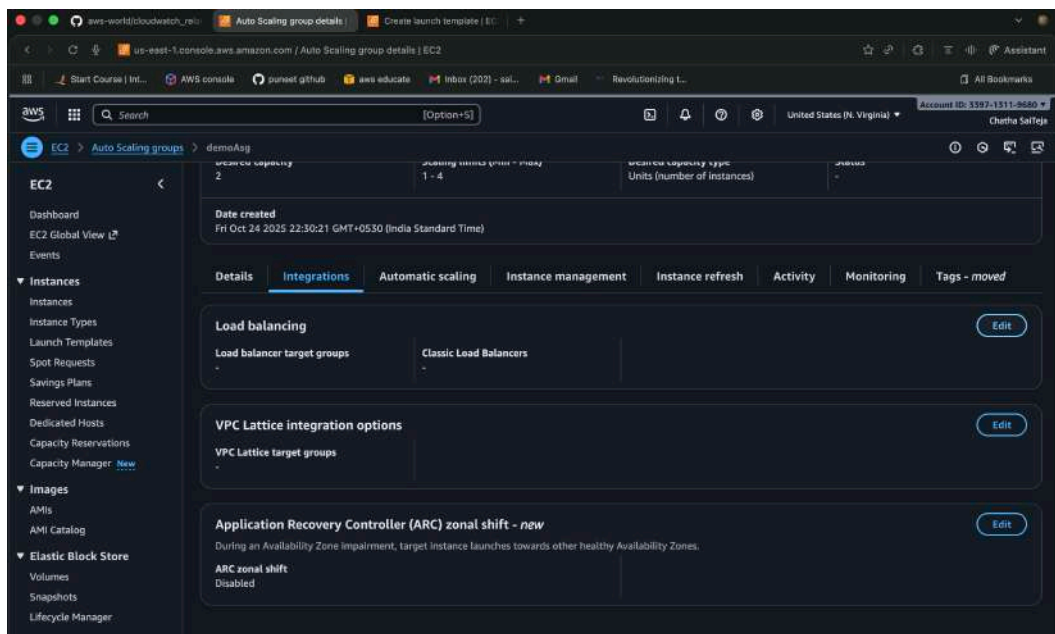


Specifying the Metric



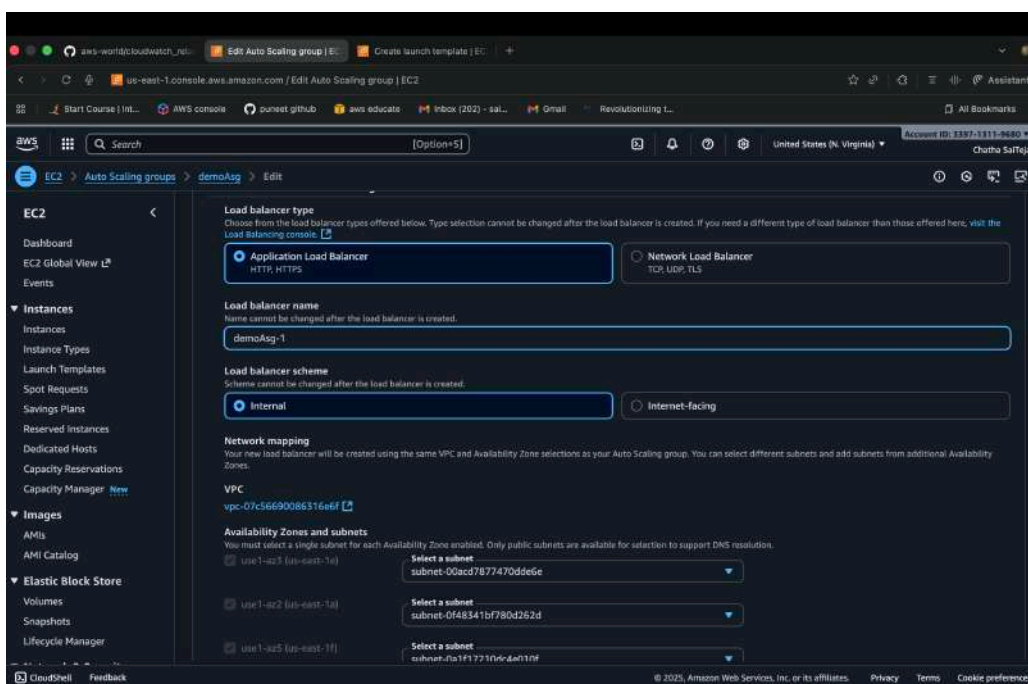
Target Tracking policy with 80% Avg cpu utilization metric

Step 7:- We Can Create a load balancer to distribute the load between the resources by selecting Integrations In the AutoScaling Group and Click on Edit.

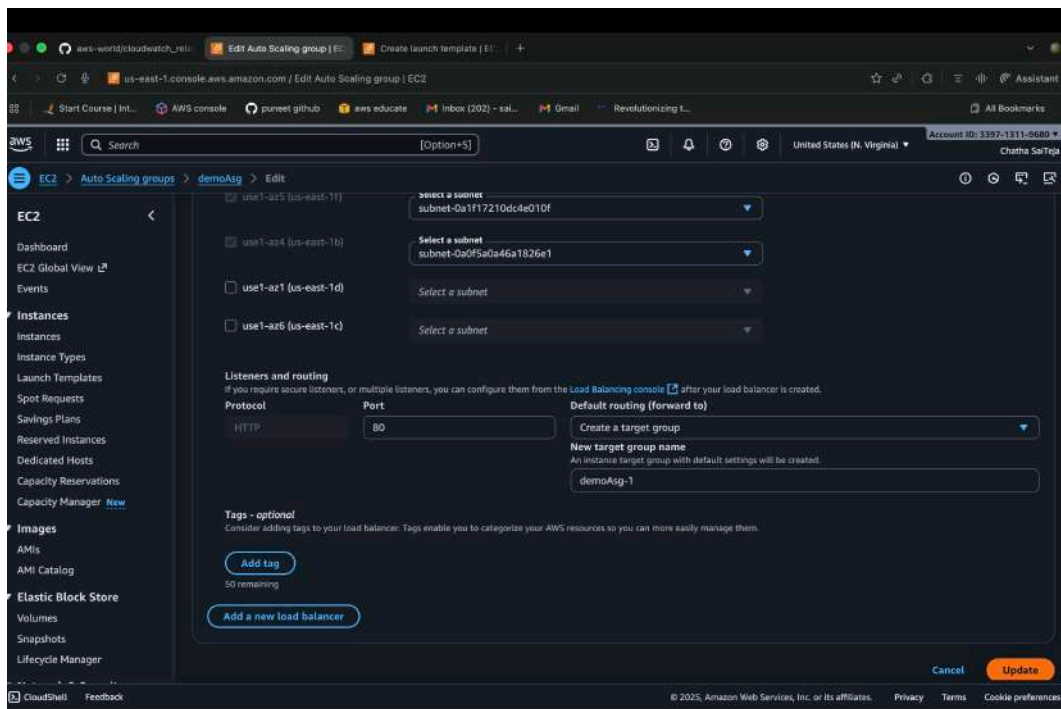


Click on Integrations

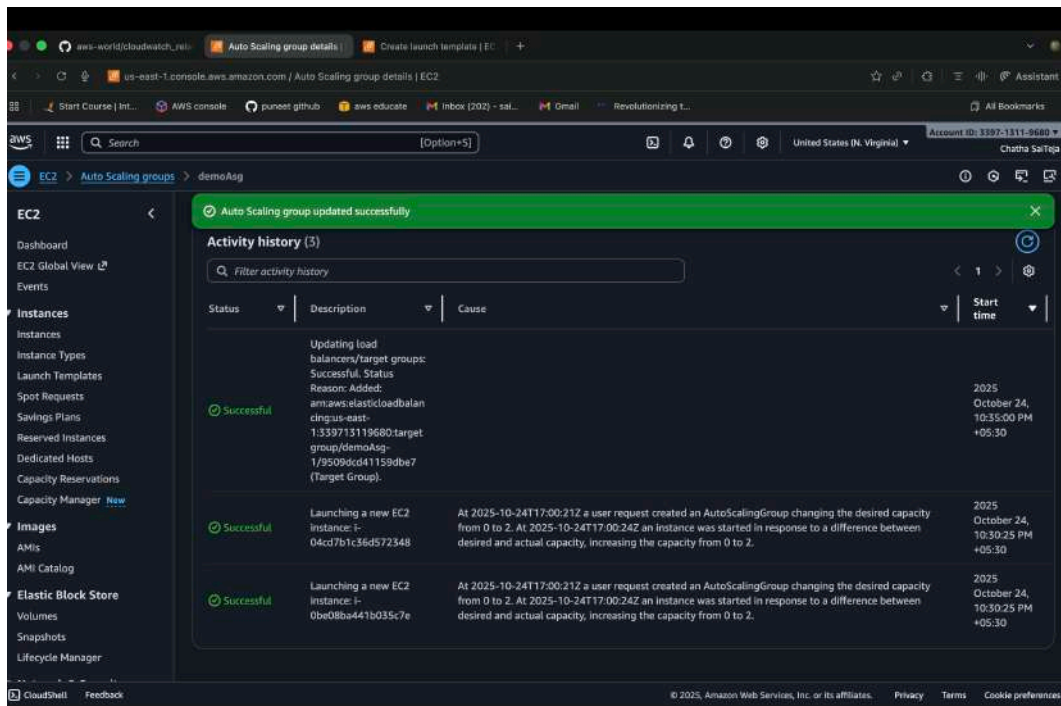
Step 8:- we select the load balancer type (ALB,NLB) and give the name to LB and create a target group it automatically creates a TG and then click on update then load balancer will be created.



Specify the name



Click on Update



Load balancer added successfully

Step 9:- The next one is we have to route the traffic to domain for that we have to buy a domain which is expensive so we ignoring the Route 53 task.