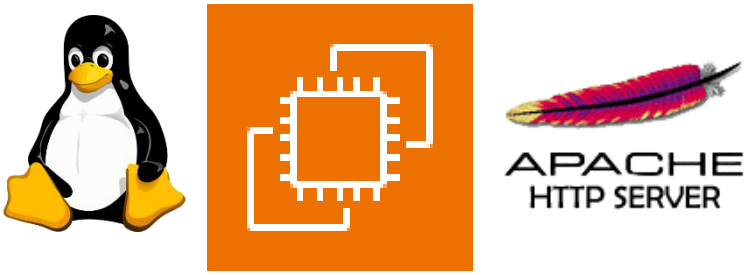
AWS Solution Architect Training with
AWS Cloud Practitioner Global Certification Training

Trainer: Aravindraj.G- Nminds Academy

Configure Auto Scaling with Fault tolerance in AWS



Pradeep Kumar R

B.E - Computer Science and Engineering

Mepco Schlenk Engineering College

Sivakasi

Objective

Auto Scaling in AWS is a service that allows you to automatically adjust the number of EC2 instances or other resources based on the demand for your application. This ensures that you have the right amount of compute capacity at all times, optimizing both performance and cost.

Key Concepts of AWS Auto Scaling:

1. Auto Scaling Groups (ASG):
    - An Auto Scaling Group is a collection of EC2 instances that can automatically scale in or out based on conditions that you specify.
    - You define a launch configuration or template (which specifies the AMI, instance type, security groups, etc.) for the instances in the ASG.
    - Auto Scaling Groups manage the minimum, maximum, and desired number of instances within your group.
2. Scaling Policies:
    - Scaling policies define when and how to scale your resources. AWS supports dynamic scaling, which adjusts capacity in response to real-time demand, and scheduled scaling, where you specify scaling actions based on predicted changes in demand.
    - Simple Scaling: Adds or removes instances based on a set threshold (e.g., if CPU utilization exceeds 80% for 5 minutes, add an instance).
    - Step Scaling: Allows you to add or remove instances in steps. For example, if the CPU usage exceeds 80%, scale by 2 instances, and if it exceeds 90%, scale by 5 instances.
    - Target Tracking Scaling: This adjusts the number of instances to keep a specific metric (like CPU utilization or request count) at a desired target value, such as 50% CPU utilization.
3. Health Checks:
    - Auto Scaling checks the health of your instances periodically. If an instance is deemed unhealthy (based on your defined health check), it is replaced automatically to maintain capacity and performance.

- Elastic Load Balancer (ELB) health checks: If your Auto Scaling group is integrated with an ELB, it can use the ELB health checks to determine the health of EC2 instances.
- EC2 Status Checks: EC2 instances have two built-in status checks — system status and instance status — to determine whether they are functioning correctly.

4. Launch Configurations/Launch Templates:
    - Launch Configuration is a template used by Auto Scaling to launch new EC2 instances. It contains settings like the AMI ID, instance type, security groups, and key pairs.
    - Launch Template is a more flexible and feature-rich option that provides more control over instance creation compared to launch configurations, allowing you to use multiple configurations (e.g., specifying the instance type, AMI, and EBS volume size).
5. Scheduled Scaling:
    - You can schedule scaling actions in advance based on predictable demand patterns. For example, if you know that traffic will increase during certain hours of the day, you can schedule an increase in the number of instances at those times.
6. Target Tracking:
    - Auto Scaling allows you to define target tracking policies, such as maintaining the average CPU utilization at a target value (e.g., 50%). AWS automatically adjusts the number of instances in the group to keep the metric at the desired value.

How AWS Auto Scaling Works:

1. Create an Auto Scaling Group: Define the number of EC2 instances, their type, and the scaling policies you want to apply.
2. Configure Scaling Policies: Define when and how Auto Scaling should increase or decrease the number of instances. This can be based on metrics like CPU utilization, network traffic, or custom metrics.
3. Health Checks: Set up health checks to automatically replace unhealthy instances in your group to ensure your application stays available.

4. Monitor and Adjust: Use Amazon CloudWatch to monitor your Auto Scaling groups, review performance metrics, and adjust your scaling policies as needed.

Types of Auto Scaling:

1. EC2 Auto Scaling: The most common type, which automatically adjusts the number of EC2 instances based on metrics or schedules.
2. Application Auto Scaling: This is used for other AWS services like Amazon ECS (Elastic Container Service), Amazon DynamoDB, Amazon Aurora, and AWS Lambda. You can scale resources based on demand in these services as well.
3. Elastic Load Balancer (ELB) Integration: You can combine Auto Scaling with ELB to ensure that incoming traffic is evenly distributed across healthy instances, and Auto Scaling will ensure there are enough instances to handle the traffic.

Benefits of Auto Scaling:

1. Cost Optimization: Auto Scaling ensures you only use the resources you need, which can reduce costs during periods of low demand. It can scale up when demand increases, ensuring that you don't experience service interruptions.
2. High Availability: Auto Scaling ensures that there are always enough healthy instances to meet traffic demands. If an instance fails or becomes unhealthy, Auto Scaling replaces it automatically.
3. Improved Performance: Auto Scaling adjusts the number of instances based on real-time demand, ensuring that your application performs well, even during traffic spikes.
4. Flexibility: You can set scaling policies based on metrics that matter to you (e.g., CPU, memory, network I/O, or custom CloudWatch metrics). This gives you flexibility to respond to different conditions and needs.
5. Automatic Adjustment: Scaling up or down happens automatically without needing manual intervention, reducing the risk of human error and ensuring the right resources are always in place.

Example Use Case:

Consider an e-commerce application that experiences spikes in traffic during a sale season or certain times of the day. You can use Auto Scaling to:

- Scale Out: Automatically add more EC2 instances when traffic increases, ensuring that users have a fast and responsive experience.
- Scale In: Automatically remove EC2 instances when traffic decreases, so you're not paying for idle resources.

Example: Setting up an EC2 Auto Scaling Group

1. Create a Launch Template: Define the configuration of EC2 instances, including the AMI, instance type, and other settings.
2. Create an Auto Scaling Group:
    - Specify the VPC, availability zones, and subnet(s).
    - Set the Desired Capacity, Minimum Capacity, and Maximum Capacity for your group.
3. Define Scaling Policies:
    - Set a policy to scale out if the average CPU utilization exceeds 80% for 5 minutes (for example).
    - Set a policy to scale in if the average CPU utilization is below 30% for 5 minutes.
4. Attach an Elastic Load Balancer (Optional): If you want to distribute incoming traffic across your instances, integrate an Elastic Load Balancer with your Auto Scaling group.

Auto Scaling and CloudWatch:

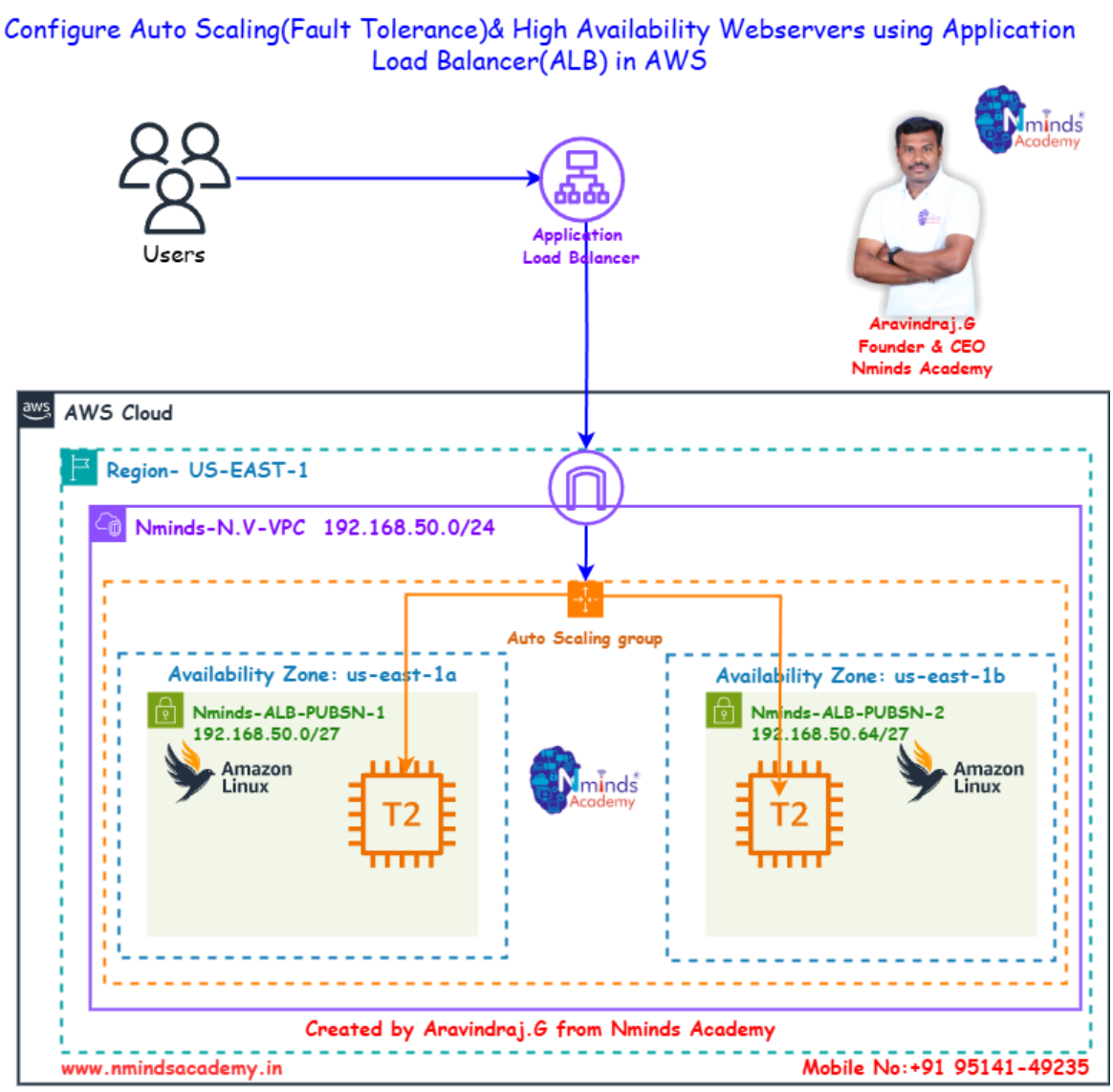
- CloudWatch Alarms: CloudWatch allows you to set up alarms based on EC2 instance metrics (like CPU utilization, disk reads, etc.). These alarms can trigger Auto Scaling actions to increase or decrease the number of instances.
- Custom Metrics: You can also use custom CloudWatch metrics to define more specific Auto Scaling conditions (e.g., application-specific metrics like request count or queue length).

Best Practices:

1. Use multiple availability zones to increase fault tolerance and reduce the risk of downtime.
2. Set up proper health checks to ensure that unhealthy instances are replaced automatically.

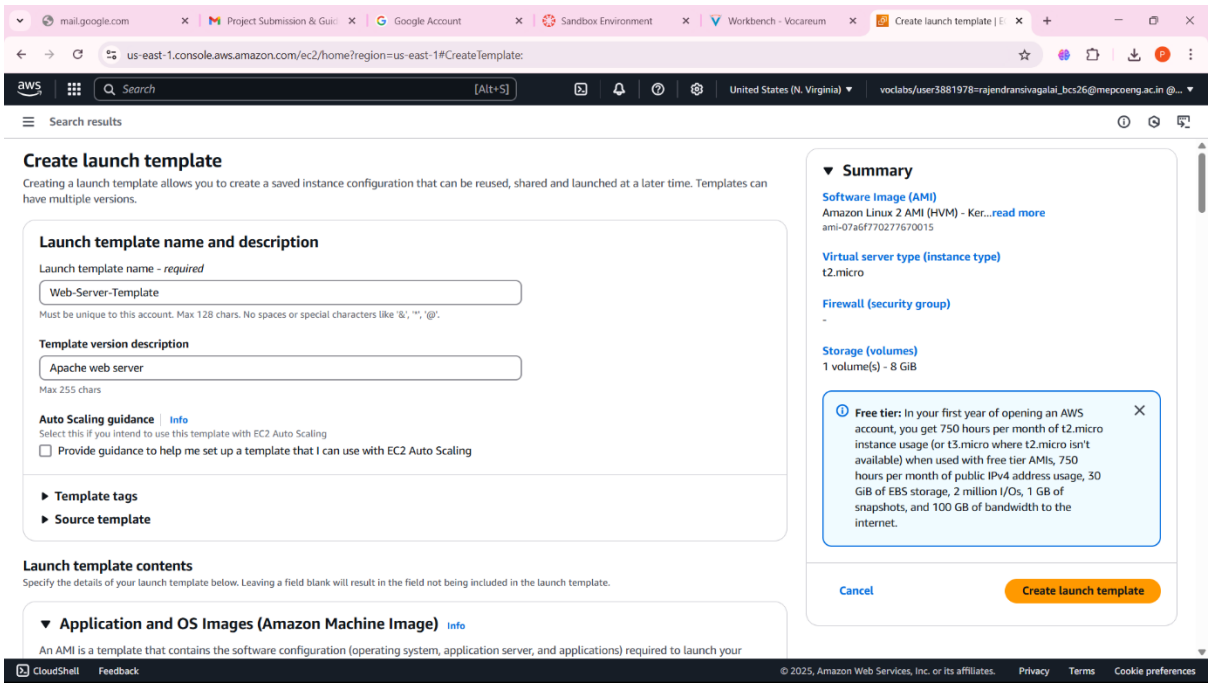
3. Avoid over-provisioning by setting the desired capacity close to the minimum capacity required for your baseline traffic.
4. Use target tracking scaling to automatically adjust your instances to a target metric, like CPU utilization or response time, ensuring better application performance.

Topology

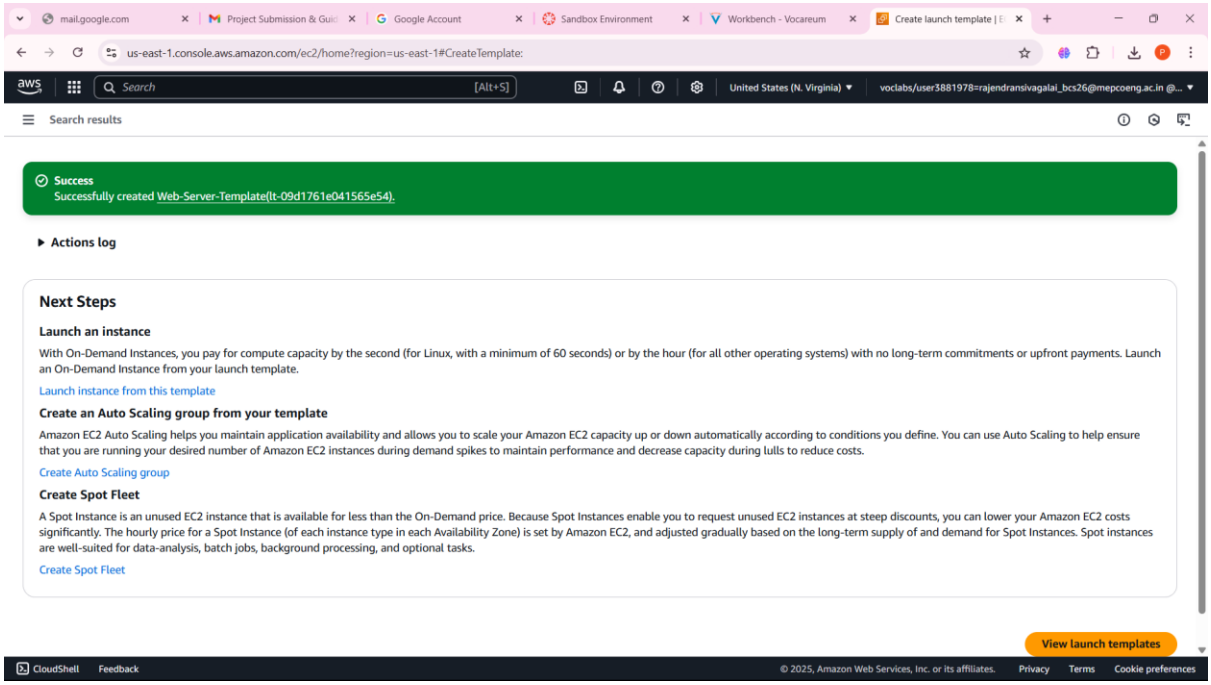


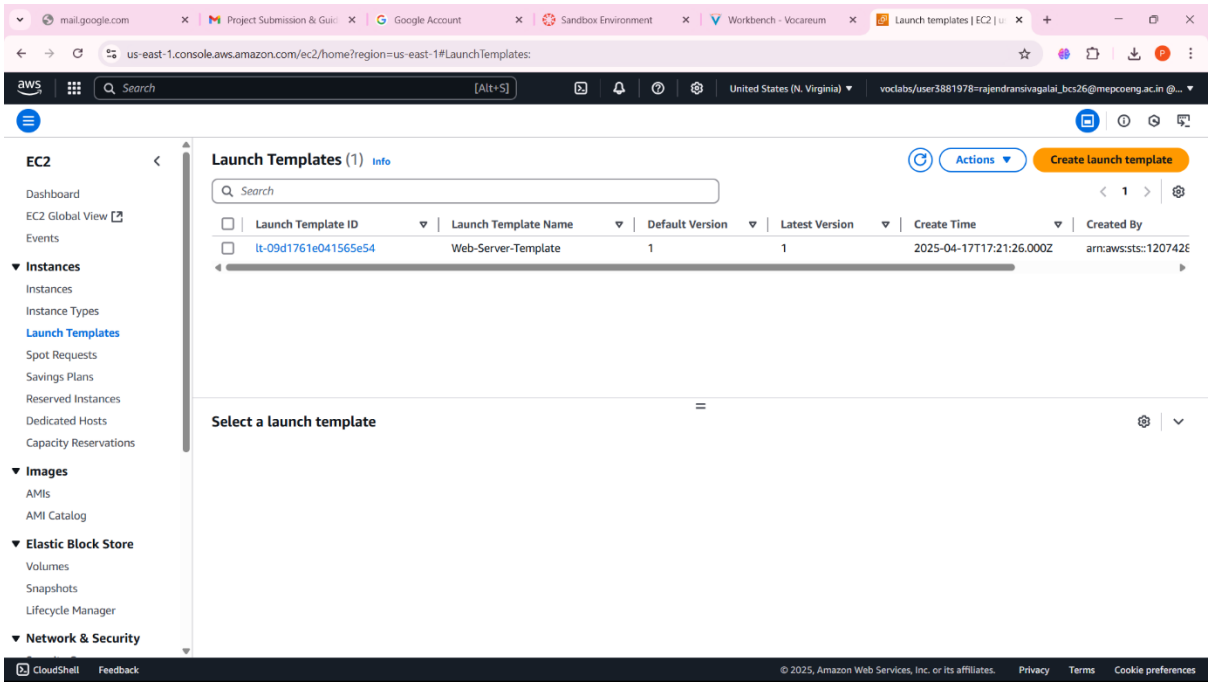
Execution Tasks:

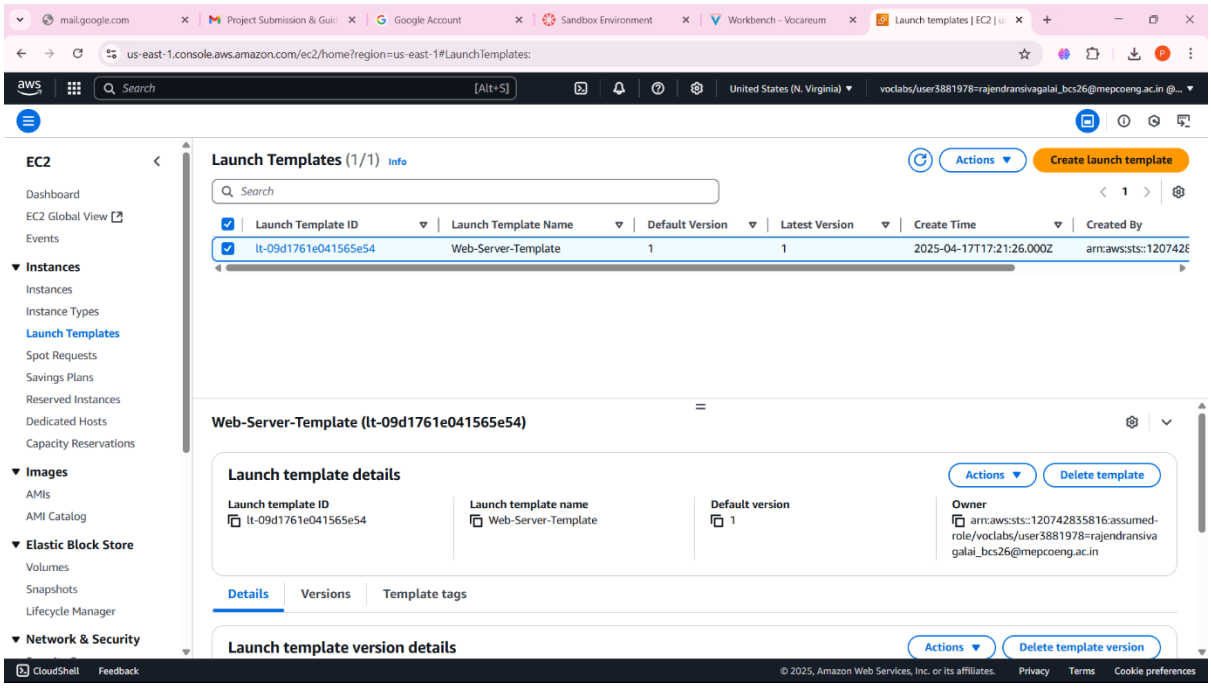
Step1: Create a Launch Template

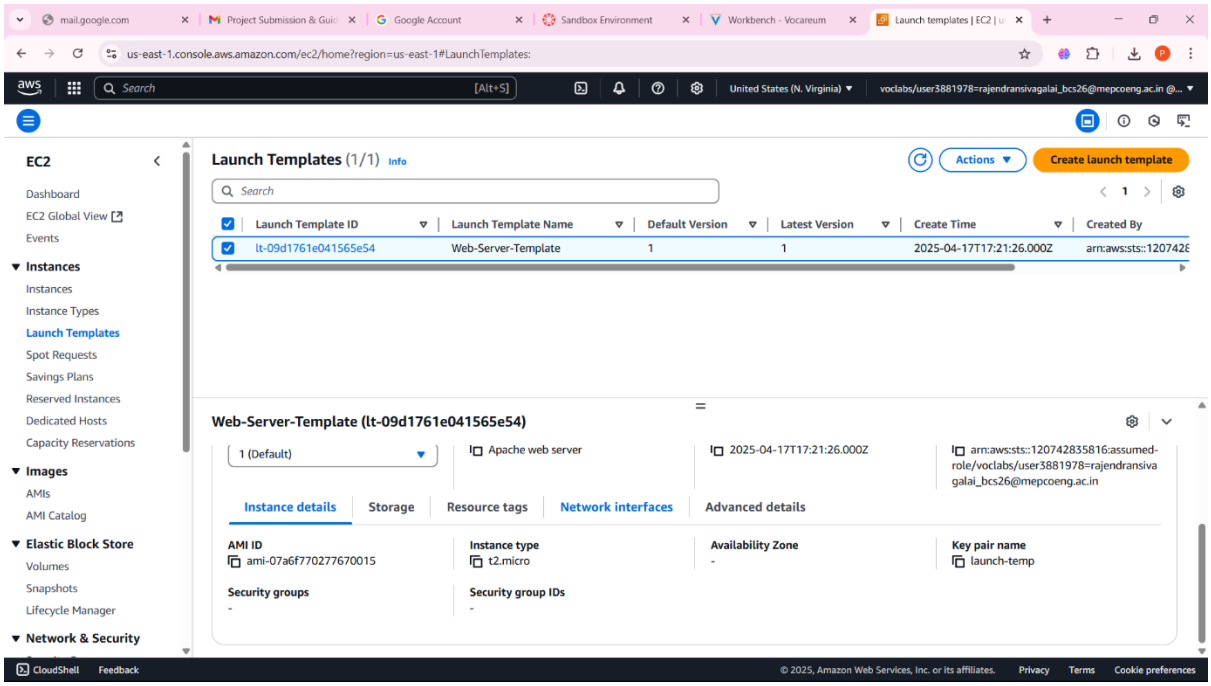


Step 2: Configure Template:

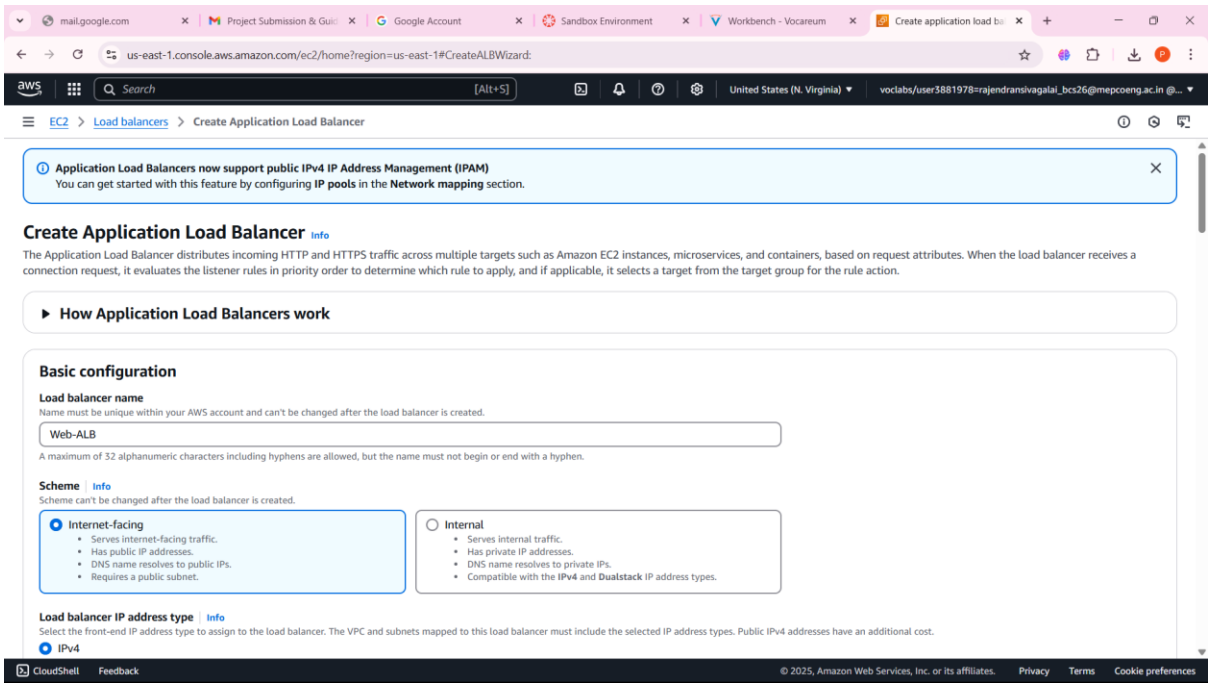


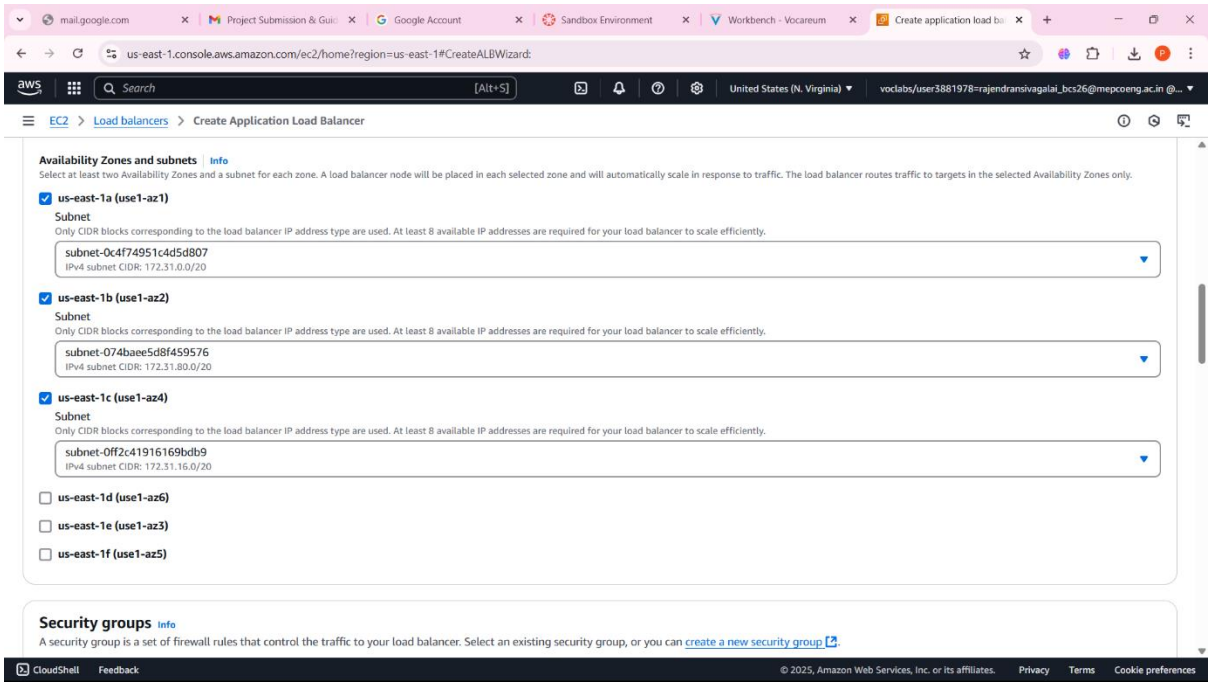

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates:
EC2
Launch Templates (1) Info
Actions
Create launch template
Launch Template ID
Launch Template Name
Default Version
Latest Version
Create Time
Created By
lt-09d1761e041565e54
Web-Server-Template
1
1
2025-04-17T17:21:26.000Z
Select a launch template
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

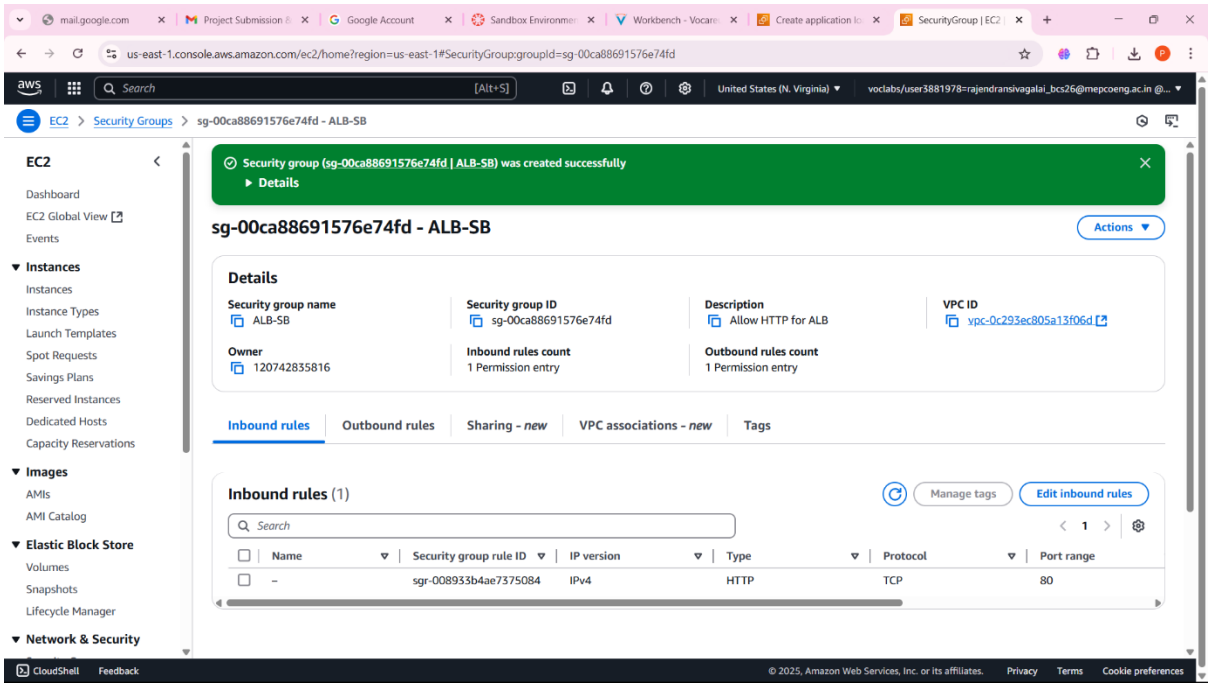

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates:
EC2
Launch Templates (1/1) Info
Actions
Create launch template
Launch Template ID
Launch Template Name
Default Version
Latest Version
Create Time
Created By
lt-09d1761e041565e54
Web-Server-Template
1
1
2025-04-17T17:21:26.000Z
Web-Server-Template (lt-09d1761e041565e54)
Launch template details
Actions
Delete template
Launch template ID
lt-09d1761e041565e54
Launch template name
Web-Server-Template
Default version
1
Owner
arn:aws:sts::120742835816:assumed-role/voclabs/user3881978=rajendransivagalai_bcs26@mepcoeng.ac.in
Details
Versions
Template tags
Launch template version details
Actions
Delete template version
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

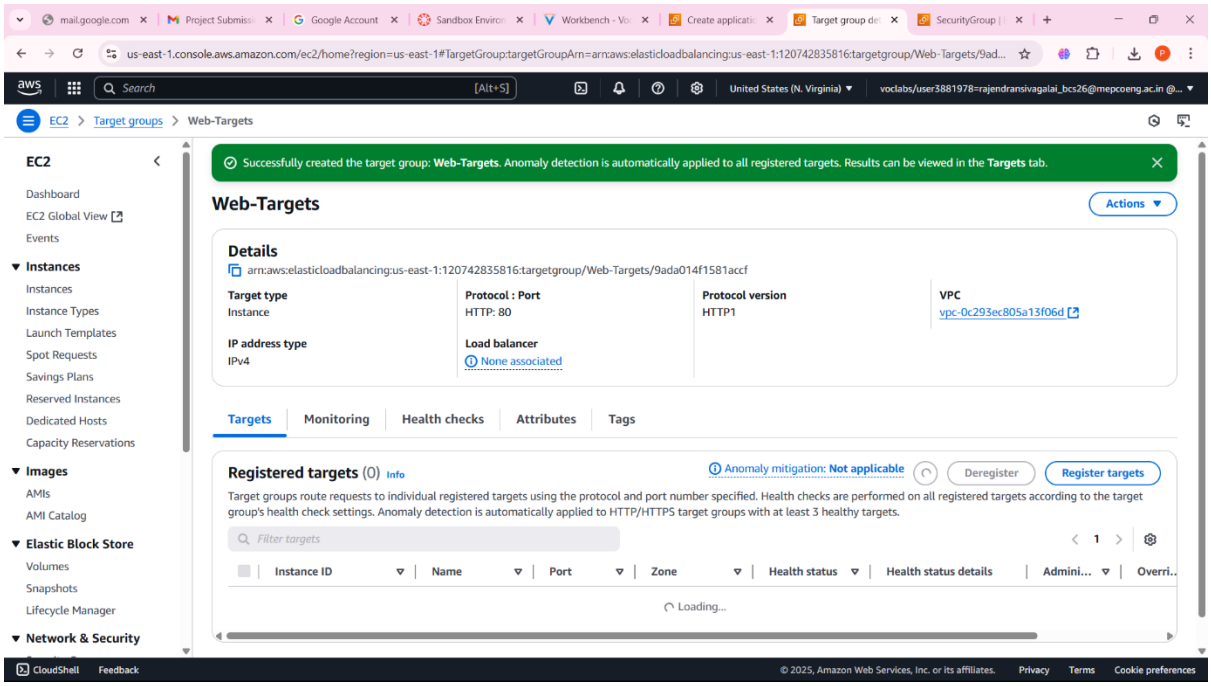


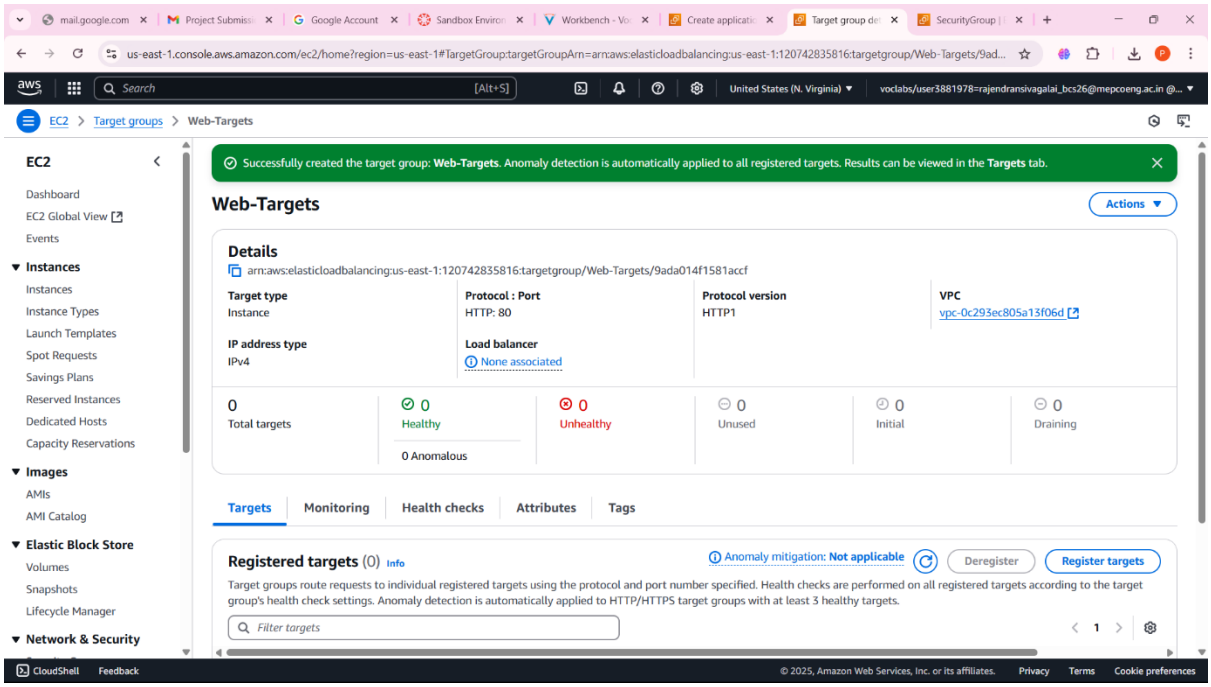
Step 3: Configure Application Load Balancer

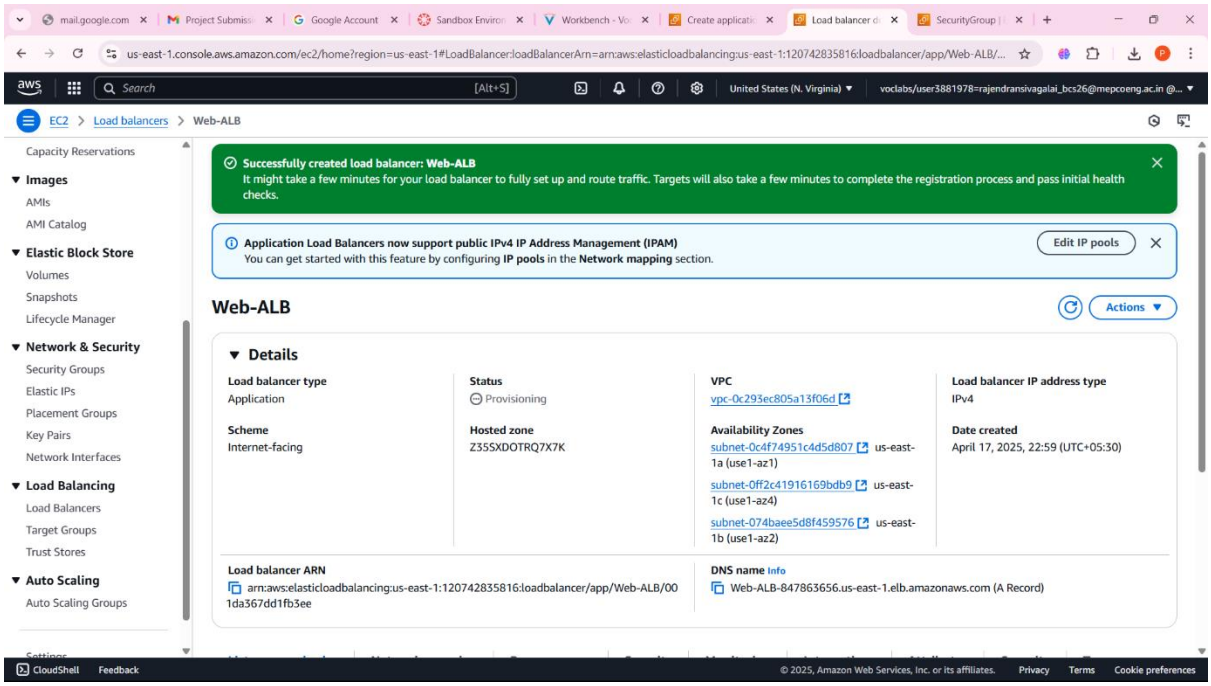



us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard:
EC2 > Load balancers > Create Application Load Balancer
Availability Zones and subnets Info
Select at least two Availability Zones and a subnet for each zone. A load balancer node will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.
us-east-1a (use1-az1)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0c4f74951c4d5d807
IPv4 subnet CIDR: 172.31.0.0/20
us-east-1b (use1-az2)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-074baee5d8f459576
IPv4 subnet CIDR: 172.31.80.0/20
us-east-1c (use1-az4)
Subnet
Only CIDR blocks corresponding to the load balancer IP address type are used. At least 8 available IP addresses are required for your load balancer to scale efficiently.
subnet-0ff2c41916169bdb9
IPv4 subnet CIDR: 172.31.16.0/20
us-east-1d (use1-az6)
us-east-1e (use1-az3)
us-east-1f (use1-az5)
Security groups Info
A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can create a new security group.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

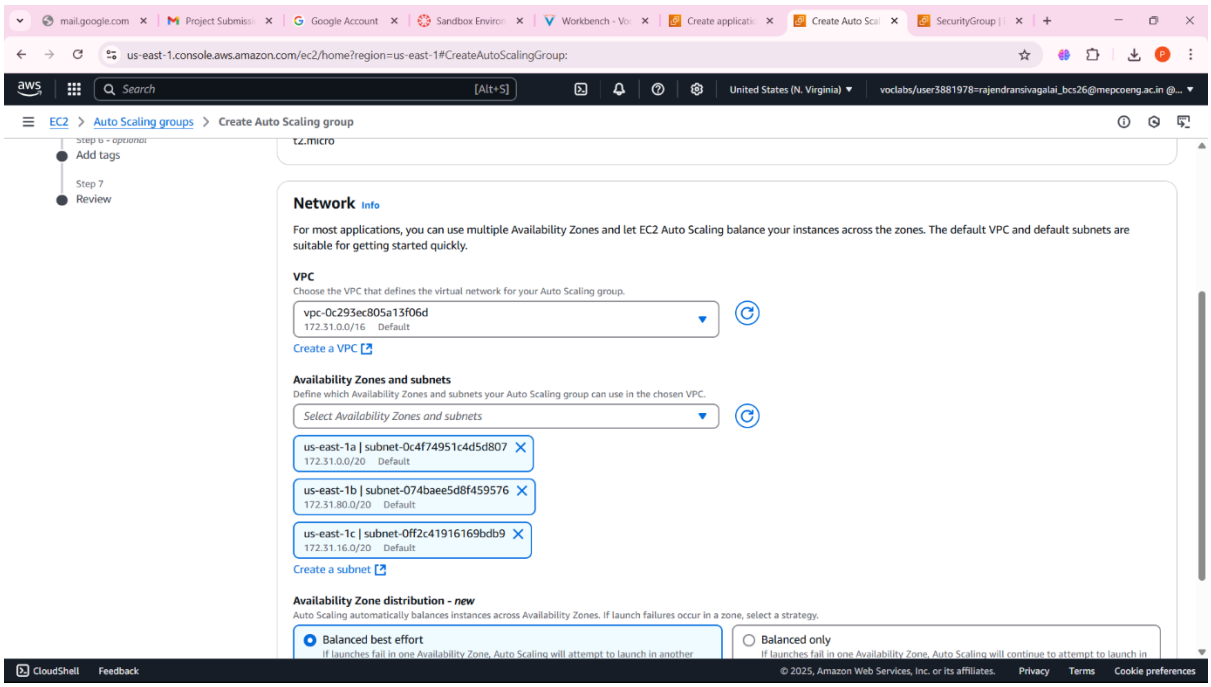

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SecurityGroup:groupId=sg-00ca88691576e74fd
EC2 > Security Groups > sg-00ca88691576e74fd - ALB-SB
Security group (sg-00ca88691576e74fd | ALB-SB) was created successfully
Details
sg-00ca88691576e74fd - ALB-SB
Actions
Details
Security group name
ALB-SB
Security group ID
sg-00ca88691576e74fd
Description
Allow HTTP for ALB
VPC ID
vpc-0c293ec805a13f06d
Owner
120742835816
Inbound rules count
1 Permission entry
Outbound rules count
1 Permission entry
Inbound rules
Outbound rules
Sharing - new
VPC associations - new
Tags
Inbound rules (1)
Manage tags
Edit inbound rules
Name
Security group rule ID
IP version
Type
Protocol
Port range
sgr-008933b4ae7375084
IPv4
HTTP
TCP
80
EC2
Dashboard
EC2 Global View
Events
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

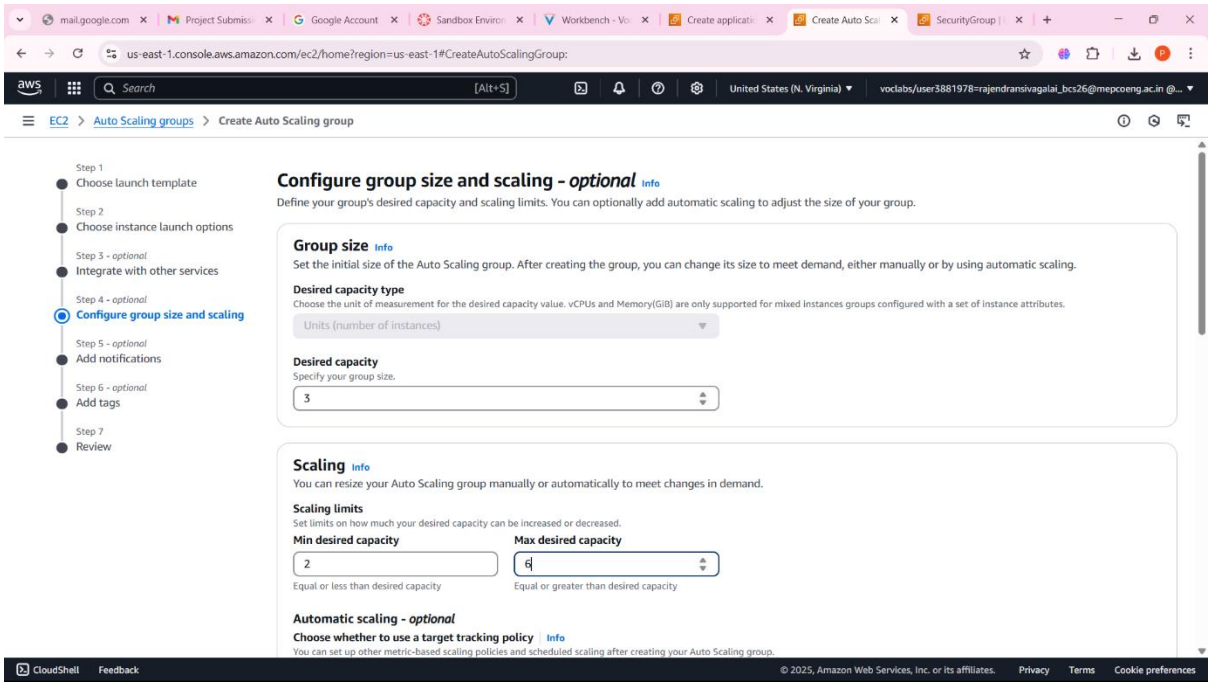

EC2 > Target groups > Web-Targets
Successfully created the target group: Web-Targets. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.
Web-Targets
Actions
Details
arn:aws:elasticloadbalancing:us-east-1:120742835816:targetgroup/Web-Targets/9ada014f1581accf
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-0c293ec805a13f06d
IP address type
IPv4
Load balancer
None associated
Targets
Monitoring
Health checks
Attributes
Tags
Registered targets (0) Info
Anomaly mitigation: Not applicable
Deregister
Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.
Filter targets
Instance ID
Name
Port
Zone
Health status
Health status details
Loading...
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

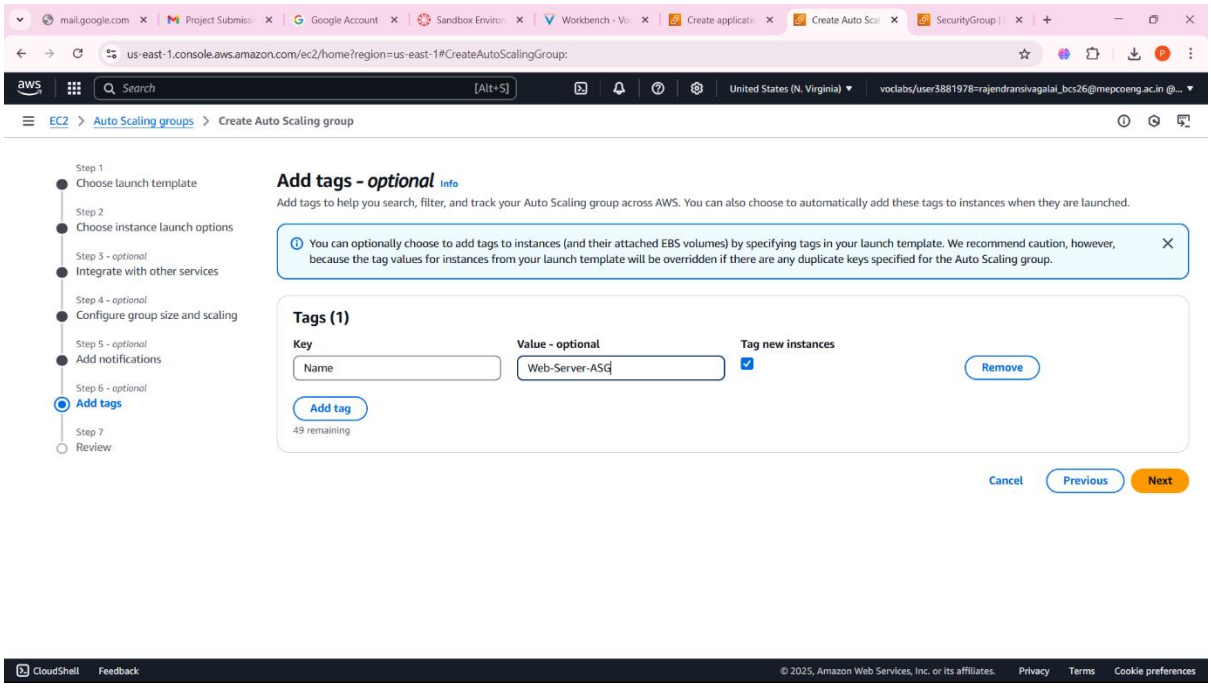

EC2 > Target groups > Web-Targets
Successfully created the target group: Web-Targets. Anomaly detection is automatically applied to all registered targets. Results can be viewed in the Targets tab.
Web-Targets
Actions
Details
arn:aws:elasticloadbalancing:us-east-1:120742835816:targetgroup/Web-Targets/9ada014f1581accf
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-0c293ec805a13f06d
IP address type
IPv4
Load balancer
None associated
0
Total targets
0
Healthy
0 Anomalous
0
Unhealthy
0
Unused
0
Initial
0
Draining
Targets
Monitoring
Health checks
Attributes
Tags
Registered targets (0) Info
Anomaly mitigation: Not applicable
Deregister
Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.
Filter targets
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

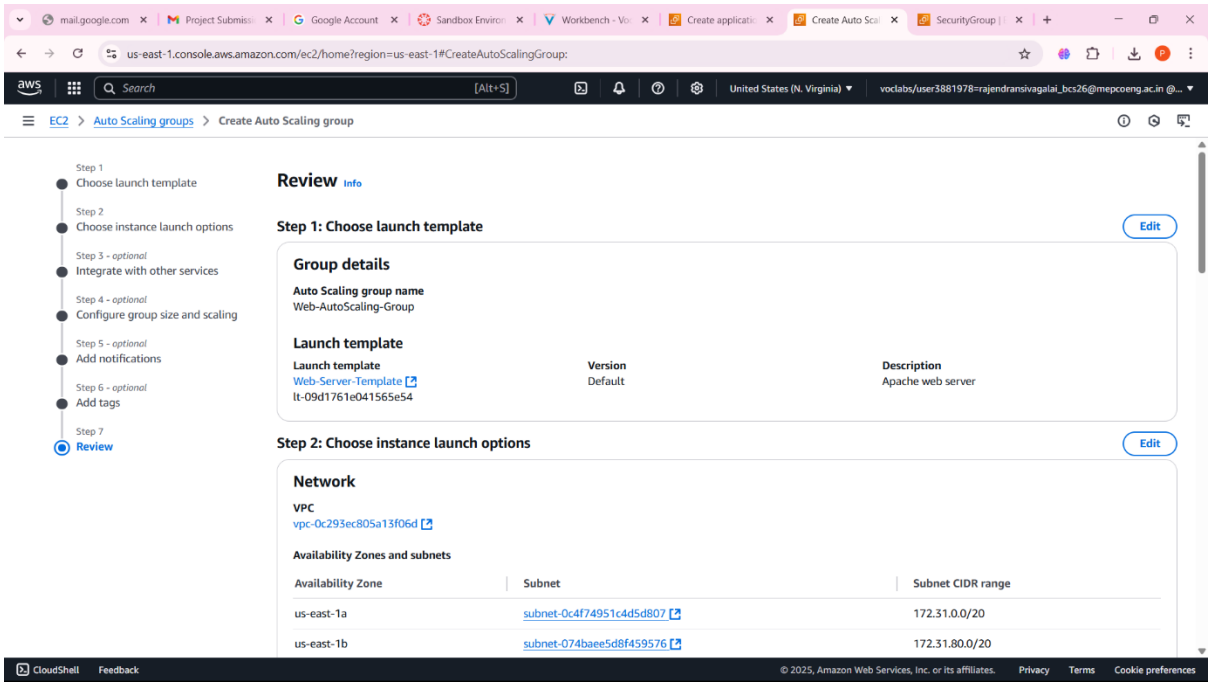


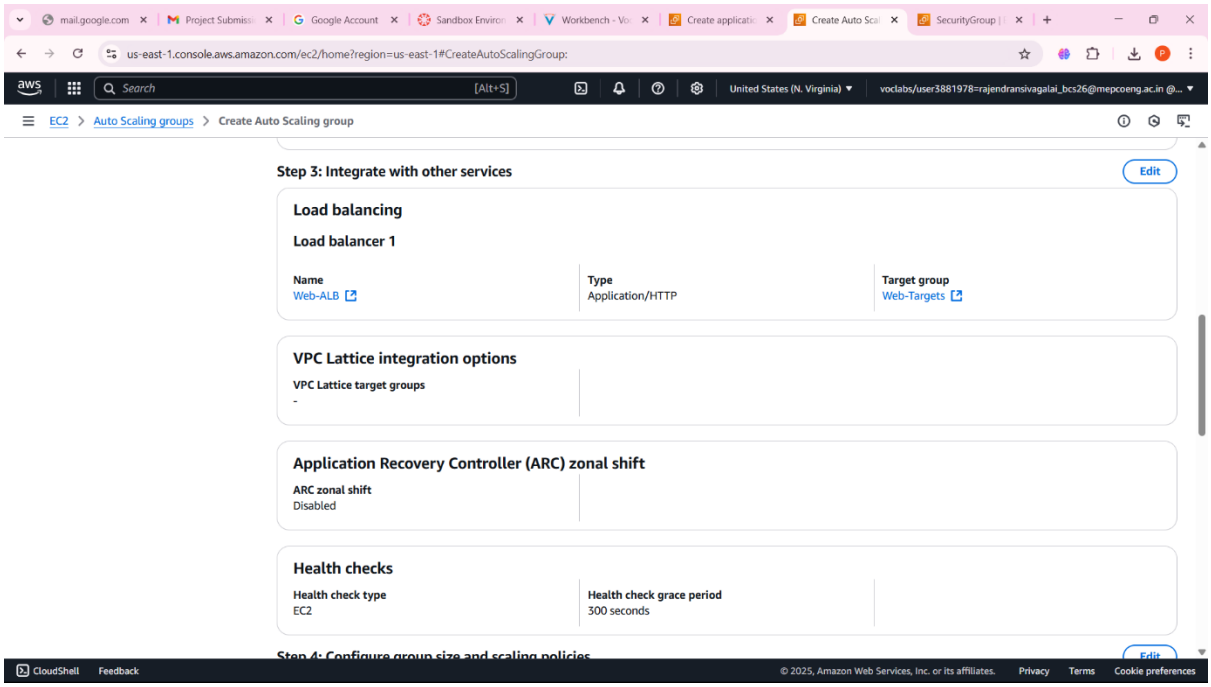
Step 4: Configure Auto Scaling Group

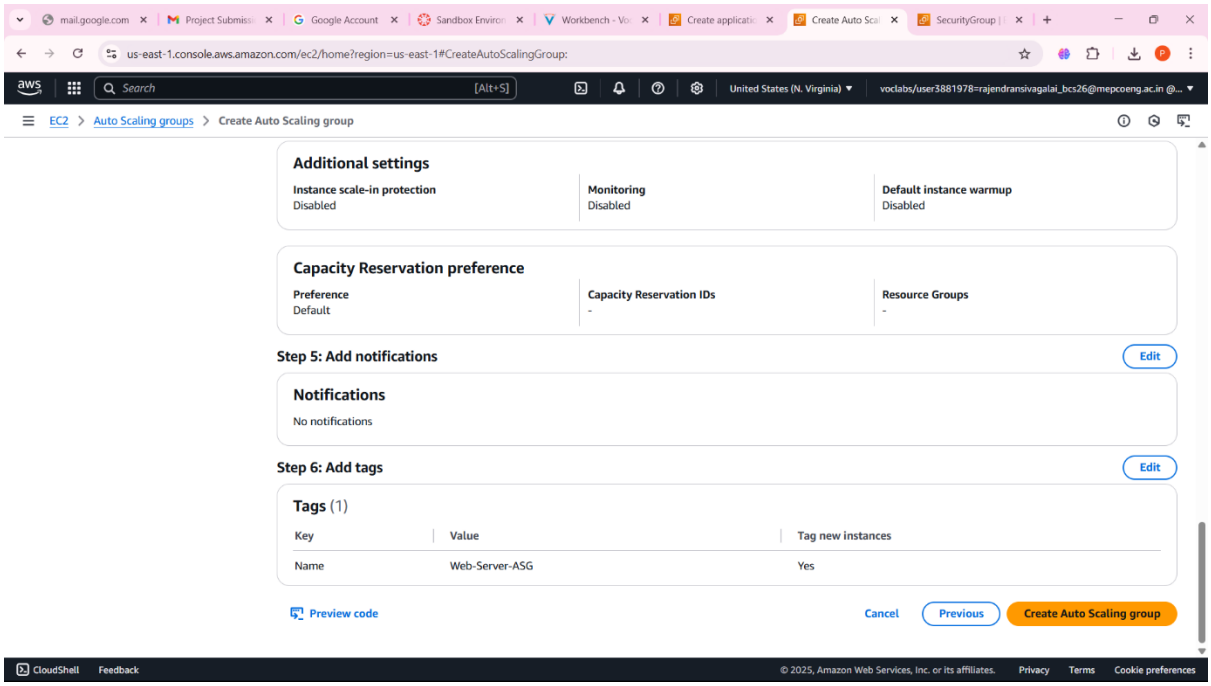


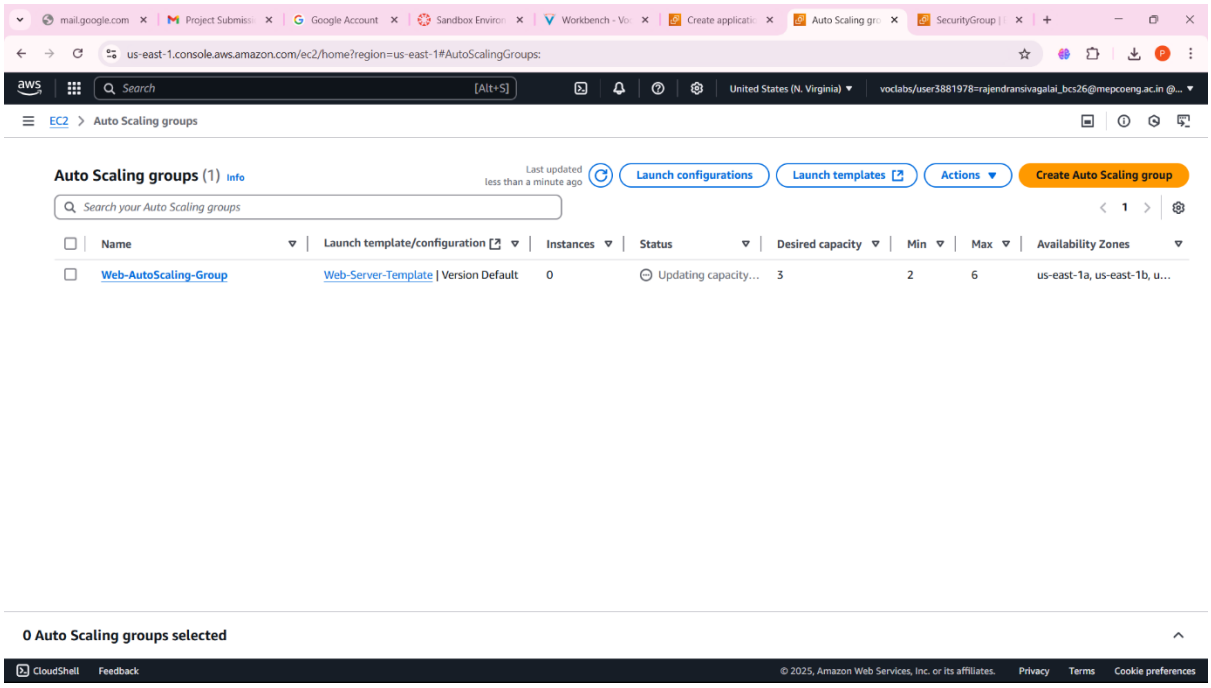

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Integrate with other services
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.
Group size Info
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.
Desired capacity type
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.
Units (number of instances)
Desired capacity
Specify your group size.
3
Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.
Scaling limits
Set limits on how much your desired capacity can be increased or decreased.
Min desired capacity
2
Equal or less than desired capacity
Max desired capacity
6
Equal or greater than desired capacity
Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:
EC2 > Auto Scaling groups > Create Auto Scaling group
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 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
Add tags - optional Info
Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.
You can optionally choose to add tags to instances (and their attached EBS volumes) by specifying tags in your launch template. We recommend caution, however, because the tag values for instances from your launch template will be overridden if there are any duplicate keys specified for the Auto Scaling group.
Tags (1)
Key
Name
Value - optional
Web-Server-ASG
Tag new instances
Remove
Add tag
49 remaining
Cancel
Previous
Next
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

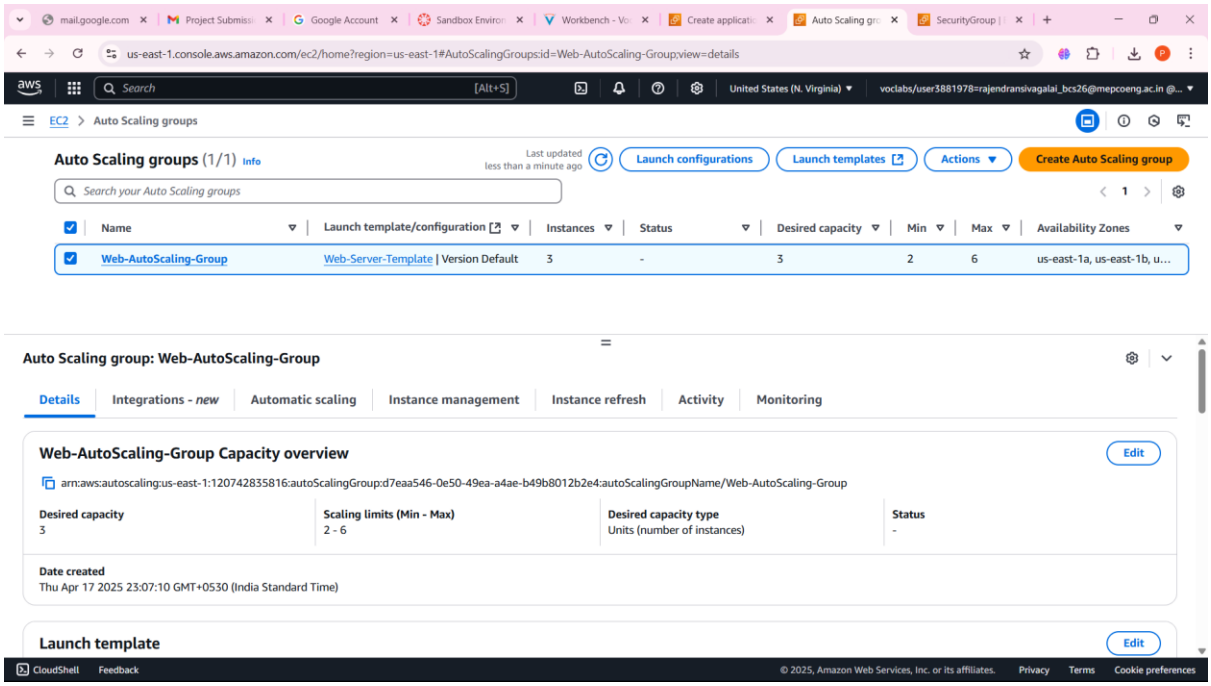

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1 Choose launch template
Step 2 Choose instance launch options
Step 3 - optional Integrate with other services
Step 4 - optional Configure group size and scaling
Step 5 - optional Add notifications
Step 6 - optional Add tags
Step 7 Review
Review Info
Step 1: Choose launch template
Edit
Group details
Auto Scaling group name
Web-AutoScaling-Group
Launch template
Launch template
Web-Server-Template
lt-09d1761e041565e54
Version
Default
Description
Apache web server
Step 2: Choose instance launch options
Edit
Network
VPC
vpc-0c293ec805a13f06d
Availability Zones and subnets
Availability Zone
Subnet
Subnet CIDR range
us-east-1a
subnet-0c4f74951c4d5d807
172.31.0.0/20
us-east-1b
subnet-074baee5d8f459576
172.31.80.0/20
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:
EC2 > Auto Scaling groups > Create Auto Scaling group
Step 3: Integrate with other services
Edit
Load balancing
Load balancer 1
Name
Web-ALB
Type
Application/HTTP
Target group
Web-Targets
VPC Lattice integration options
VPC Lattice target groups
-
Application Recovery Controller (ARC) zonal shift
ARC zonal shift
Disabled
Health checks
Health check type
EC2
Health check grace period
300 seconds
Edit
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences





Step 5: Test Auto Scaling and Fault Tolerance


us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:id=Web-AutoScaling-Group;view=details
Search
[Alt+S]
United States (N. Virginia)
EC2 > Auto Scaling groups
Auto Scaling groups (1/1) Info
Last updated less than a minute ago
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
Name
Launch template/configuration
Instances
Status
Desired capacity
Min
Max
Availability Zones
Web-AutoScaling-Group
Web-Server-Template | Version Default
3
-
3
2
6
us-east-1a, us-east-1b, u...
Auto Scaling group: Web-AutoScaling-Group
Details
Integrations - new
Automatic scaling
Instance management
Instance refresh
Activity
Monitoring
Web-AutoScaling-Group Capacity overview
Edit
arn:aws:autoscaling:us-east-1:120742835816:autoScalingGroup:d7eaa546-0e50-49ea-a4ae-b49b8012b2e4:autoScalingGroupName/Web-AutoScaling-Group
Desired capacity
3
Scaling limits (Min - Max)
2 - 6
Desired capacity type
Units (number of instances)
Status
-
Date created
Thu Apr 17 2025 23:07:10 GMT+0530 (India Standard Time)
Launch template
Edit
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences