VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Fall Semester 2025-26

Lab Assignment – 4

Slot: L13+L14

Class: VL2025260105679

Branch: B.tech CSBS

Course code & title: CBS3005

Cloud, Microservices and Applications

Faculty name: Nithya K

DA by:

Kartikey Gupta

22BBS0105

1) A company wants to deploy a secure, scalable, and highly available web application on AWS for global users. Perform the following tasks in AWS and submit screenshots of each step as evidence:

(i) Launch multiple EC2 instances (web servers) and configure them in different Availability Zones.

(ii) Create an Application Load Balancer (ALB) to distribute traffic across these instances.

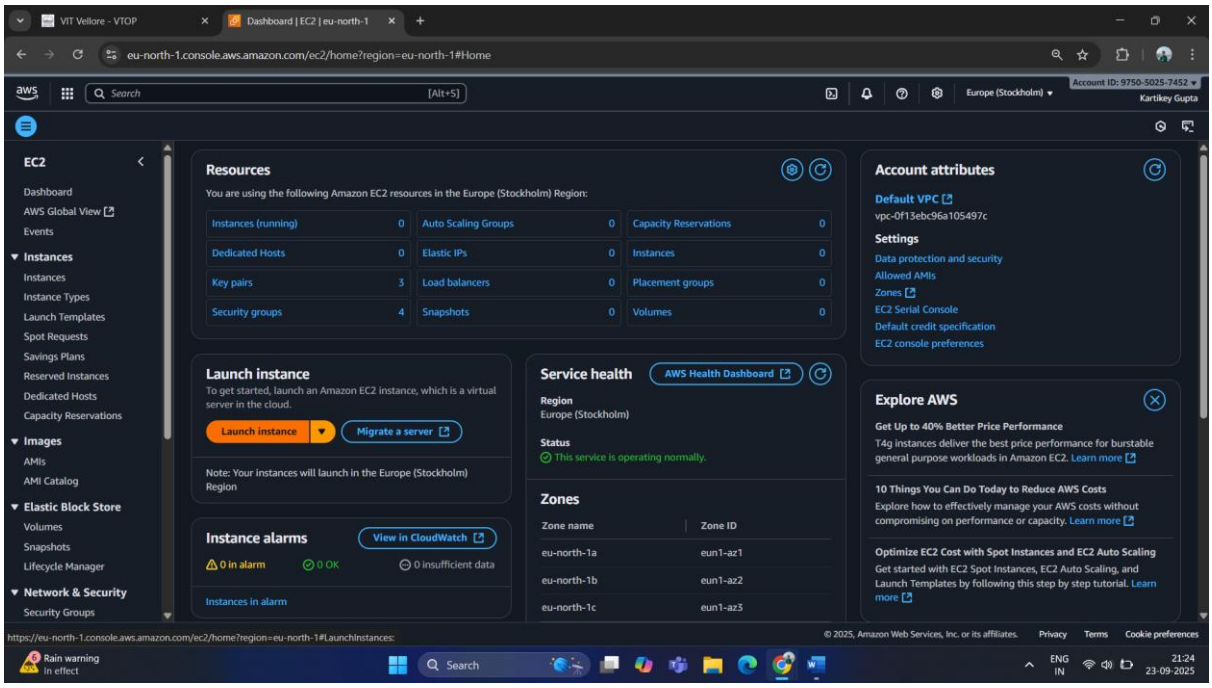
(iii) Configure health checks so that faulty instances are automatically removed from load balancing.

(iv) Enable Auto Scaling to add/remove instances based on traffic demand.

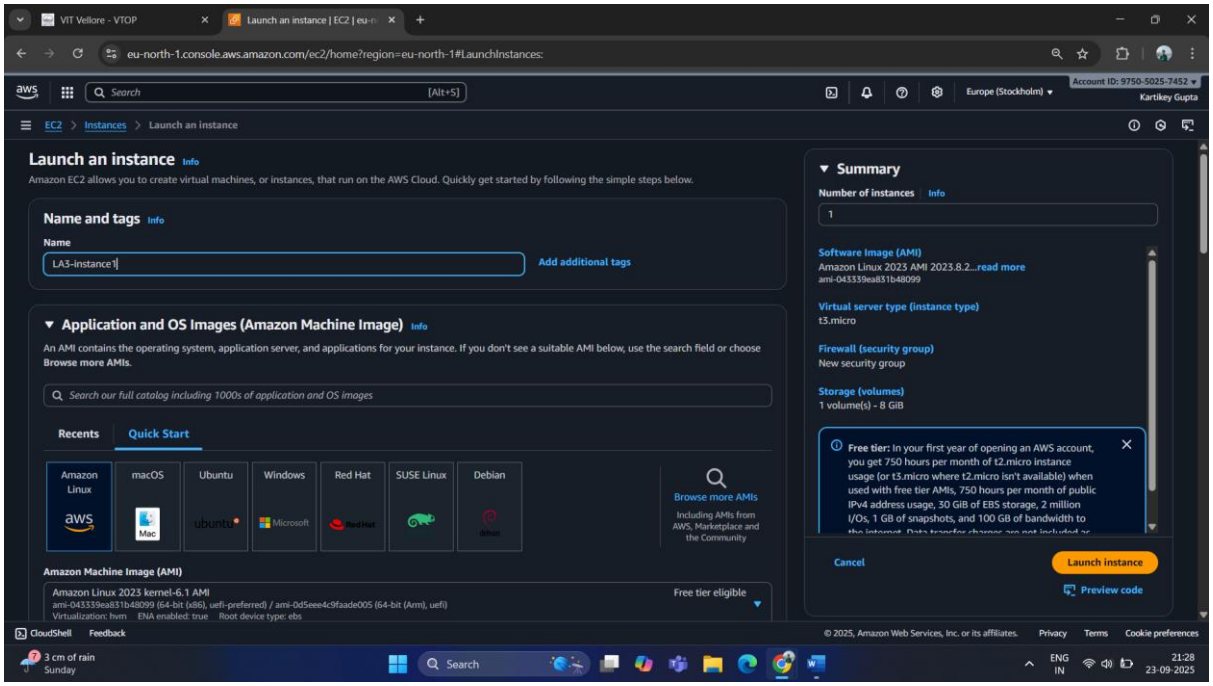
(v) Configure path-based routing: /auth requests go to the authentication service, /order requests go to the order processing service. Integrate the Load Balancer with Route 53 so that global users are routed to the nearest AWS region.

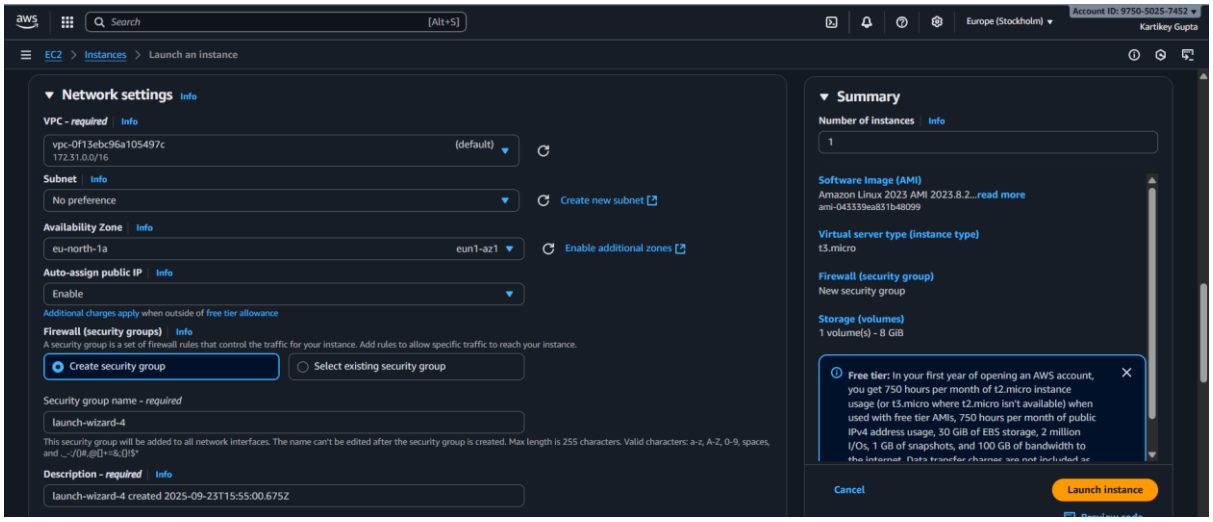
(i) Launch Multiple EC2 Instances in Different Availability Zones

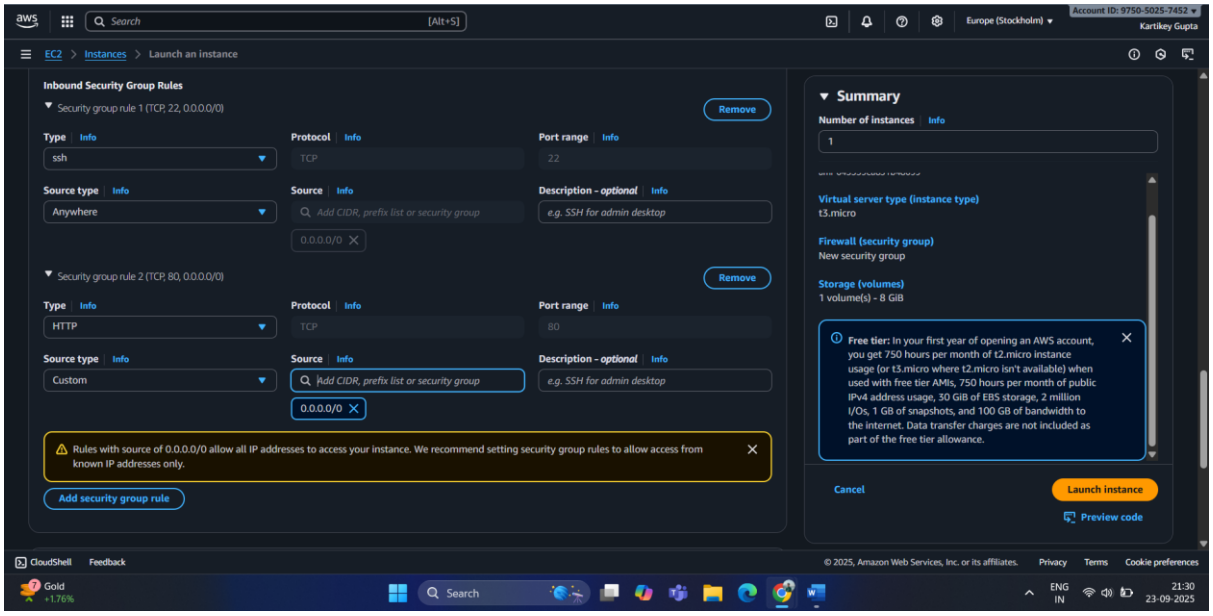
1. **Log in** to AWS Management Console → go to **EC2**.

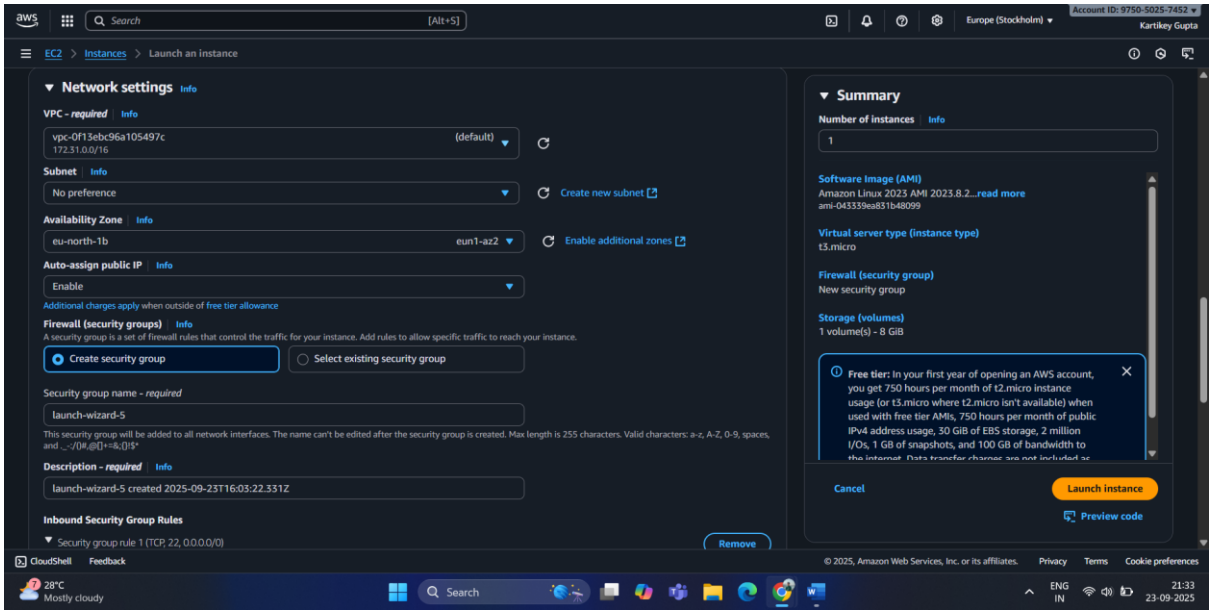


2. Click **Launch Instance** → choose Amazon Linux 2 / Ubuntu AMI.

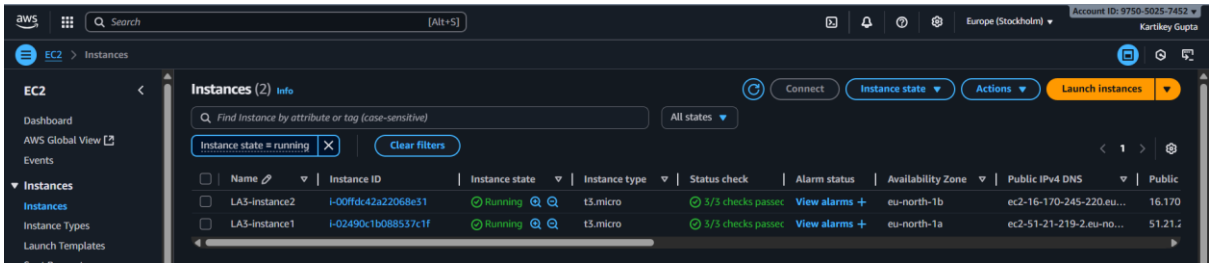

VIT Vellore - VTOP
Launch an instance | EC2 | eu-n
eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchInstances:
Search
[Alt+S]
Europe (Stockholm)
Account ID: 9750-5025-7452
Kartikey Gupta
EC2 > Instances > Launch an instance
Launch an instance Info
Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.
Name and tags Info
Name
LA3-instance1
Add additional tags
Application and OS Images (Amazon Machine Image) Info
An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.
Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Ubuntu
Windows
Red Hat
SUSE Linux
Debian
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 kernel-6.1 AMI
ami-043339ea831b48099 (64-bit (x86), uefi-preferred) / ami-0d5eee4c9faade005 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Free tier eligible
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-043339ea831b48099
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
3 cm of rain
Sunday
Search
ENG
IN
21:28
23-09-2025


Search
[Alt+S]
Europe (Stockholm)
Account ID: 9750-5025-7452
Kartikey Gupta
EC2 > Instances > Launch an instance
Network settings Info
VPC - required Info
vpc-0f13ebc96a105497c
172.31.0.0/16
(default)
Subnet Info
No preference
Create new subnet
Availability Zone Info
eu-north-1a
eun1-az1
Enable additional zones
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Security group name - required
launch-wizard-4
This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*
Description - required Info
launch-wizard-4 created 2025-09-23T15:55:00.675Z
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.8.2...read more
ami-043339ea831b48099
Virtual server type (instance type)
t3.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to
Cancel
Launch instance





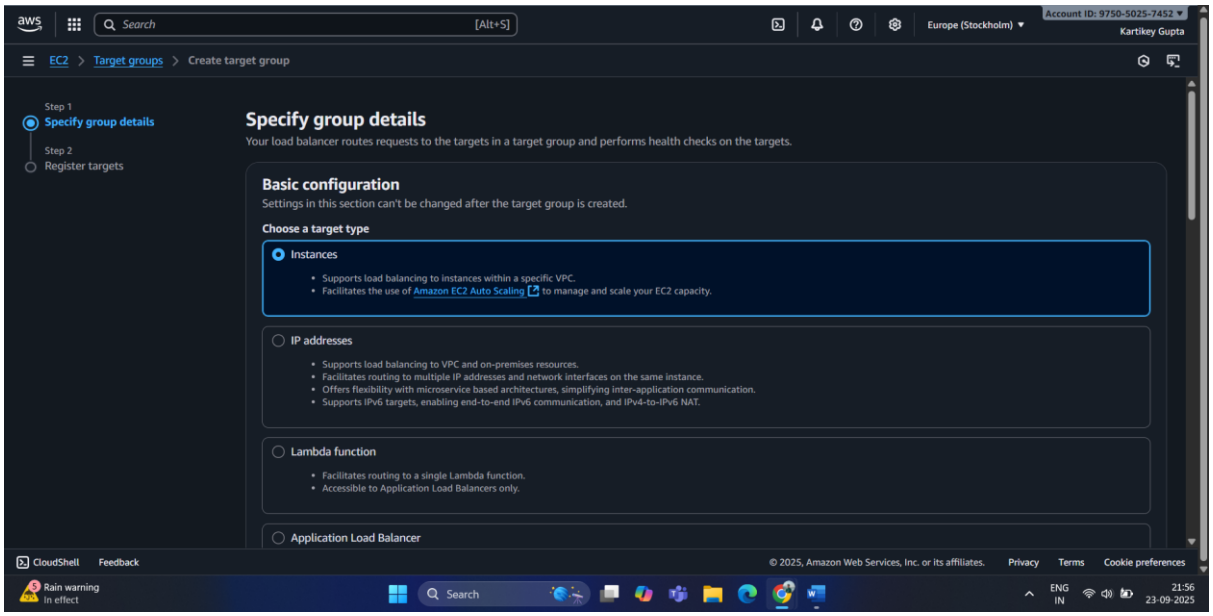
3. Select an **instance type** (e.g., t2.micro for free tier).
4. In **Network settings**:
    - Choose a **VPC** (default or custom).
    - Place **Instance 1** in Availability Zone A (e.g., ap-south-1a).
    - Place **Instance 2** in Availability Zone B (e.g., ap-south-1b).
5. Configure **Security Group**:
    - Allow HTTP (port 80) and SSH (port 22).
6. Launch instances and note **public IPs**.

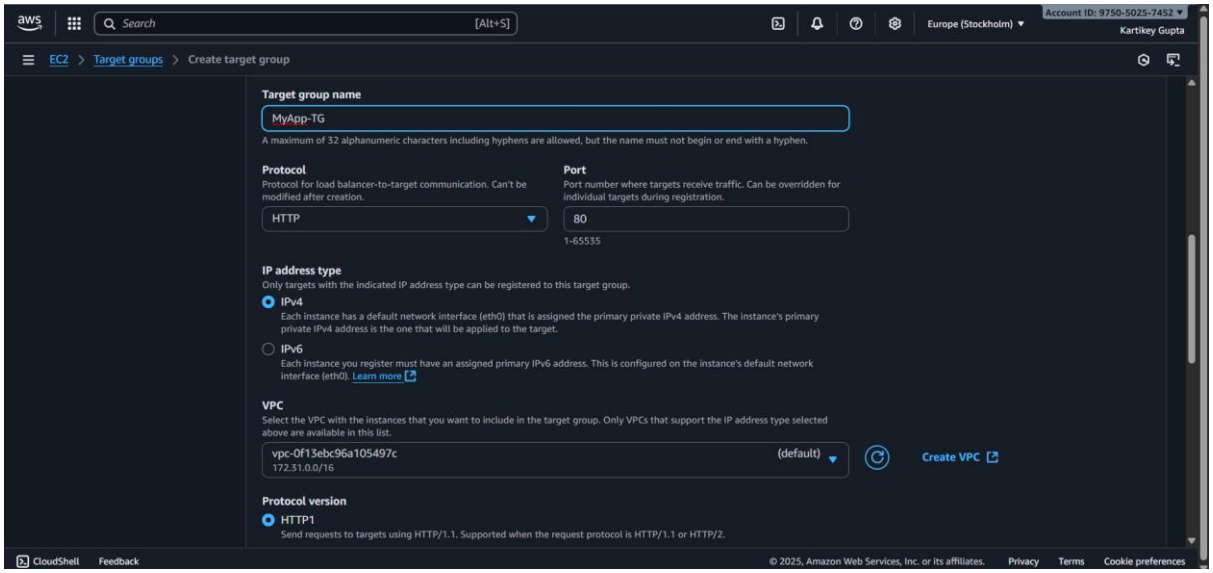


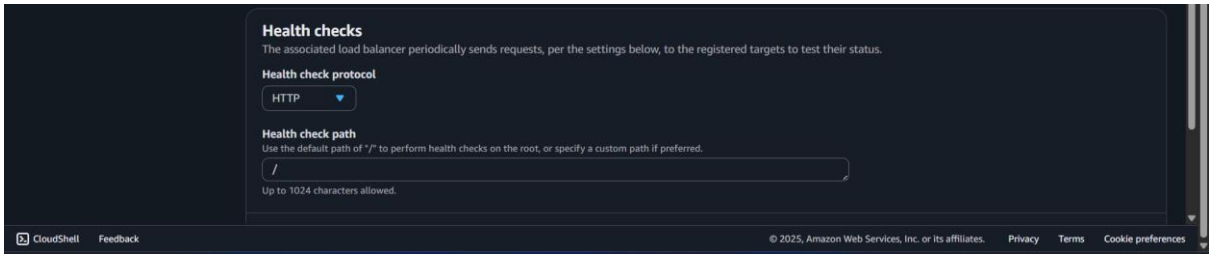
(ii) Create Application Load Balancer (ALB)

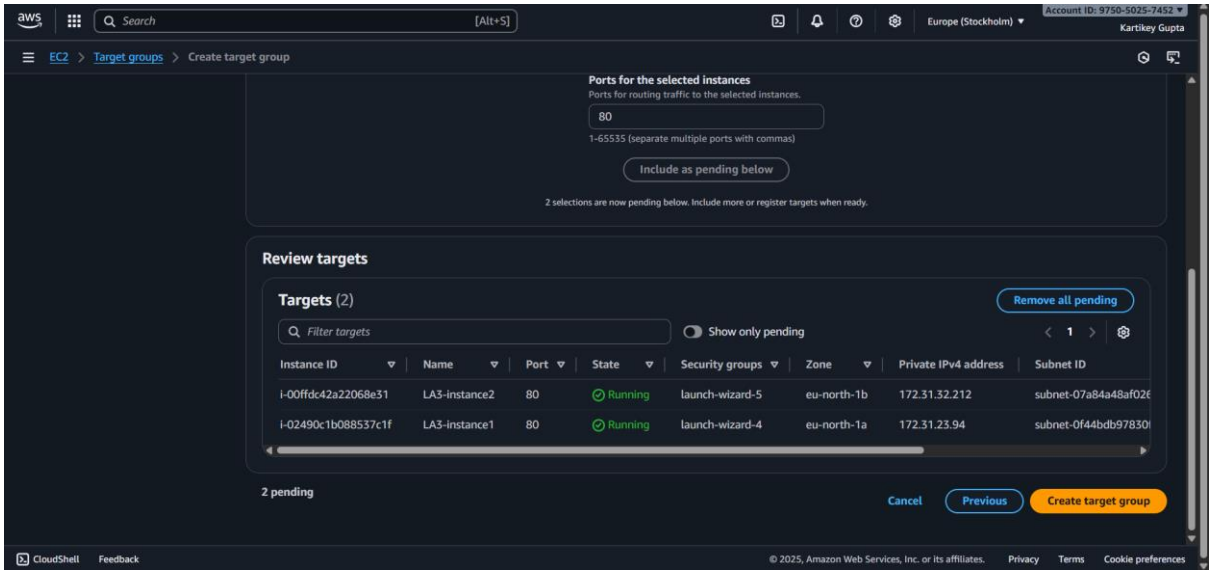
Step 1: Create Target Group(s) first

1. Go to **EC2 → Target Groups → Create target group**.
2. Configure:
    - **Target type**: Instances
    - **Name**: e.g., MyApp-TG
    - **Protocol**: HTTP
    - **Port**: 80
    - **VPC**: Select the same VPC where your EC2 instances are launched
    - **Health check**: HTTP, path /







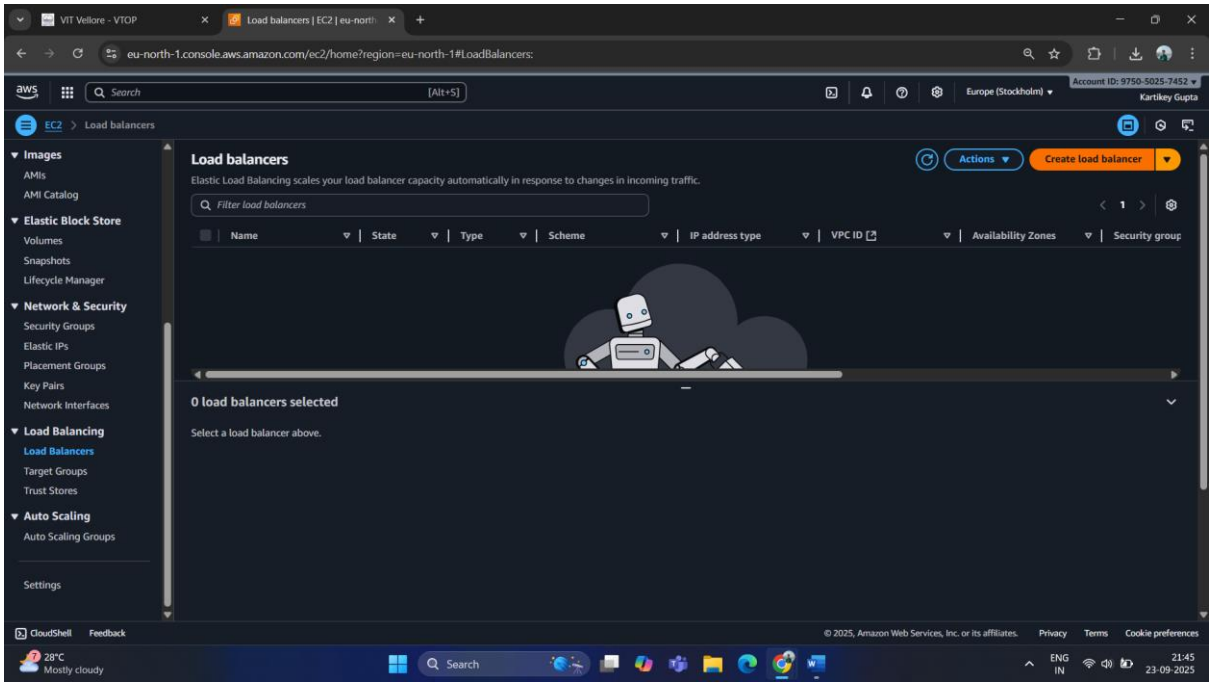


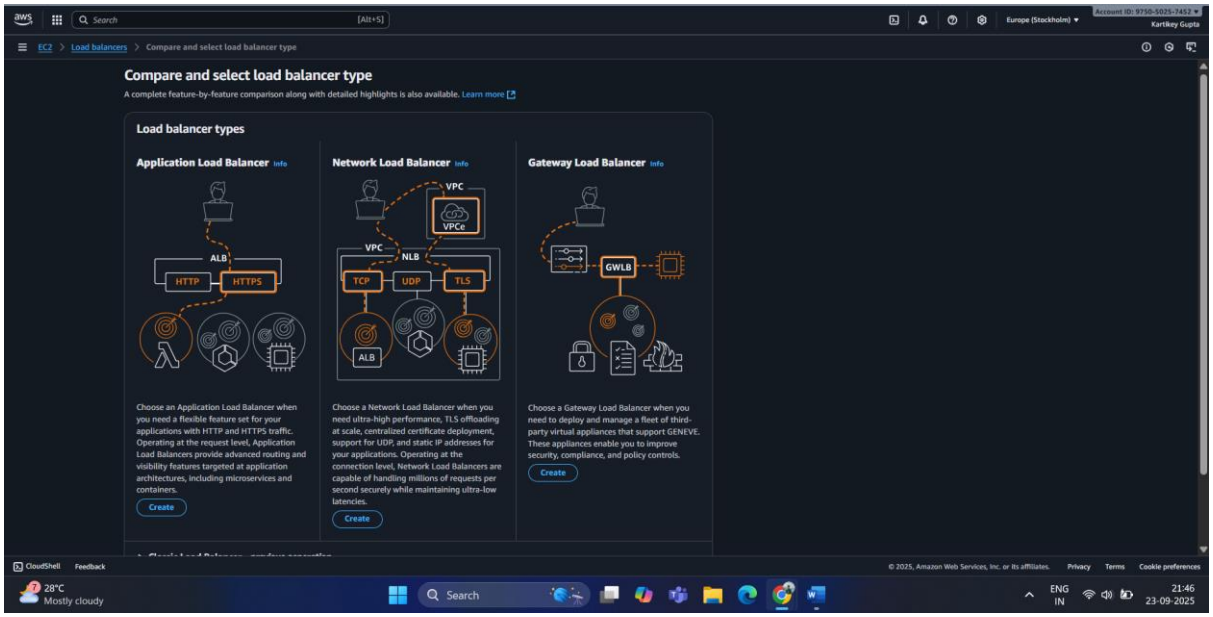
3. Click **Next** and **register your EC2 instances** in this Target Group.
4. Complete Target Group creation.
5. Install servers
    - sudo yum update -y
    - sudo yum install -y httpd
    - sudo systemctl start httpd

- sudo systemctl enable httpd
- echo "Hello from $(hostname -f)" > /var/www/html/index.html

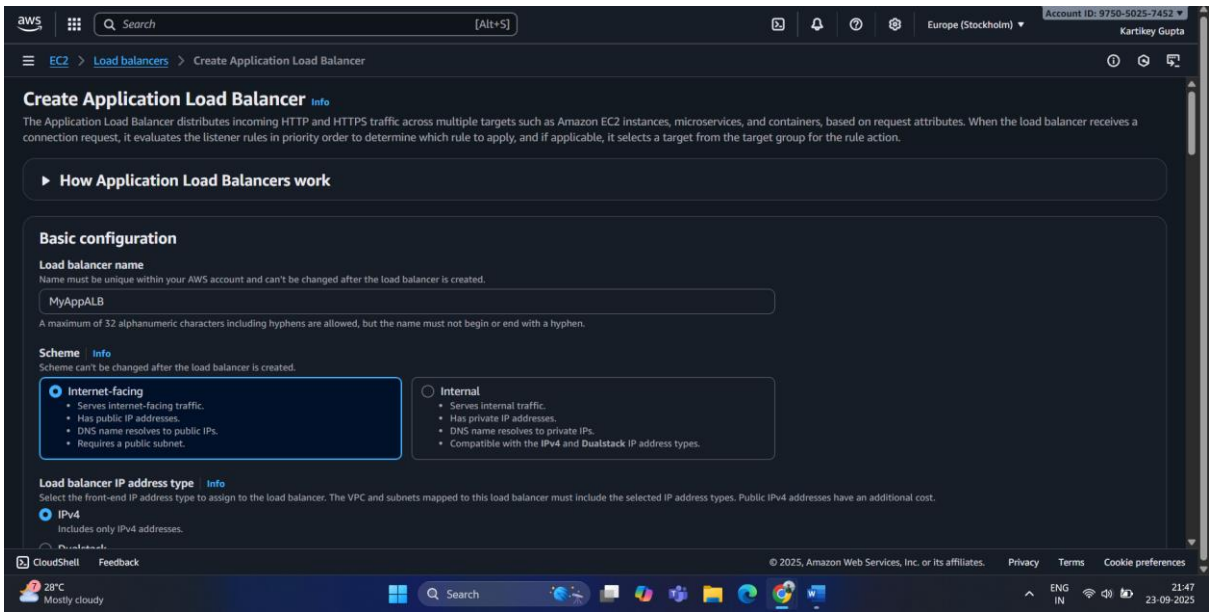
Step 2: Create ALB

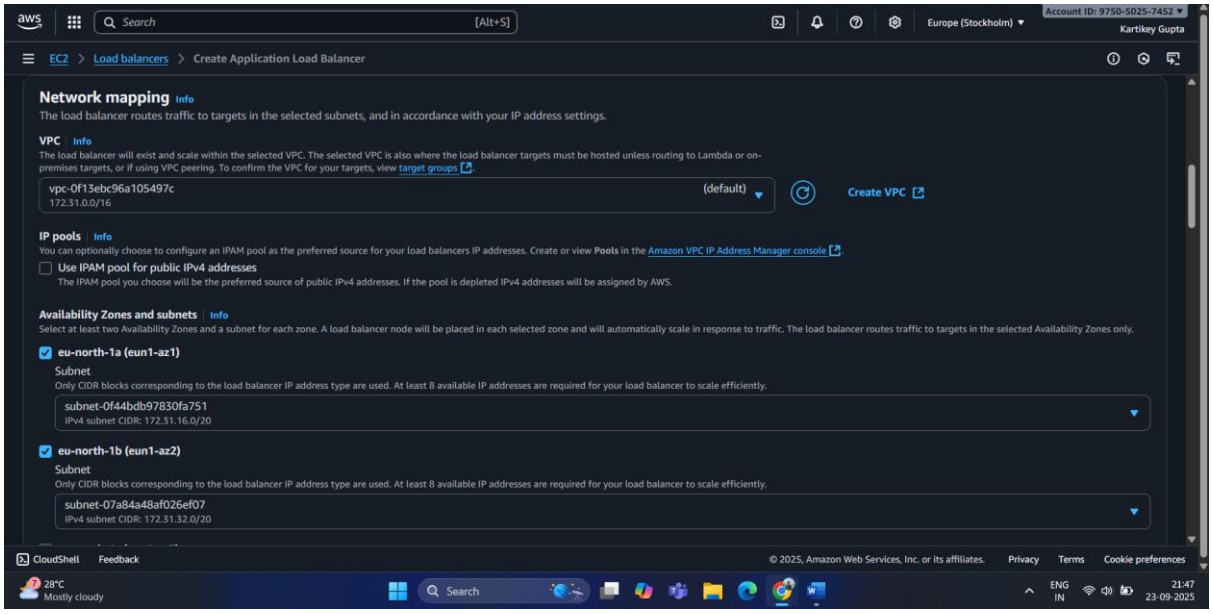
1. Go to **EC2 → Load Balancers → Create Load Balancer → Application Load Balancer**.
2. Configure:
   - **Name**: MyAppALB
   - **Scheme**: Internet-facing
   - **IP address type**: IPv4
   - **Listeners**: HTTP (80)
3. Select **Availability Zones**: choose at least 2 public subnets in different AZs.




Search [Alt+S]
Europe (Stockholm)
Kartikey Gupta
EC2 > Load balancers > Compare and select load balancer type
Compare and select load balancer type
A complete feature-by-feature comparison along with detailed highlights is also available. Learn more
Load balancer types
Application Load Balancer Info
Network Load Balancer Info
Gateway Load Balancer Info
ALB
HTTP
HTTPS
VPC
VPCe
NLB
TCP
UDP
TLS
GWLB
Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.
Create
Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.
Create
Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.
Create
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Mostly cloudy
Search
ENG
IN
21:46
23-09-2025

1. Configure:


Search [Alt+S]
Europe (Stockholm)
Kartikey Gupta
EC2 > Load balancers > Create Application Load Balancer
Create Application Load Balancer Info
The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.
How Application Load Balancers work
Basic configuration
Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
MyAppALB
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Scheme Info
Scheme can't be changed after the load balancer is created.
Internet-facing
Serves internet-facing traffic.
Has public IP addresses.
DNS name resolves to public IPs.
Requires a public subnet.
Internal
Serves internal traffic.
Has private IP addresses.
DNS name resolves to private IPs.
Compatible with the IPv4 and Dualstack IP address types.
Load balancer IP address type Info
Select the front-end IP address type to assign to the load balancer. The VPC and subnets mapped to this load balancer must include the selected IP address types. Public IPv4 addresses have an additional cost.
IPv4
Includes only IPv4 addresses.
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
28°C
Mostly cloudy
Search
ENG
IN
21:47
23-09-2025

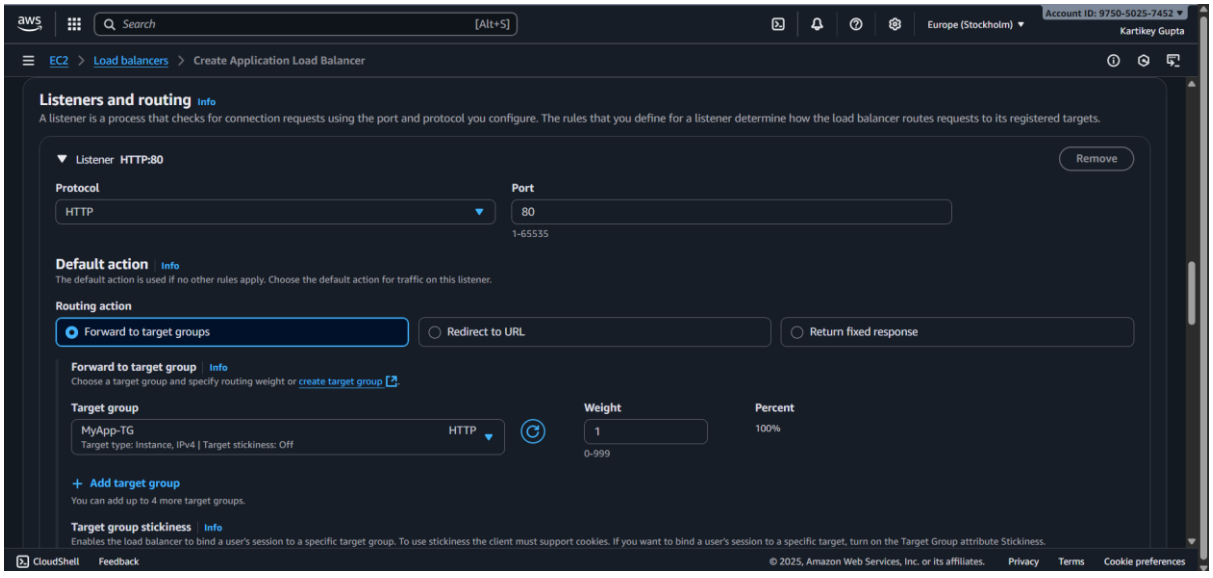


Step 3: Configure Security Group

1. Either select an existing SG or create a new one.
2. **Inbound rule**: HTTP, port 80, source 0.0.0.0/0
3. Outbound can remain default (All traffic).

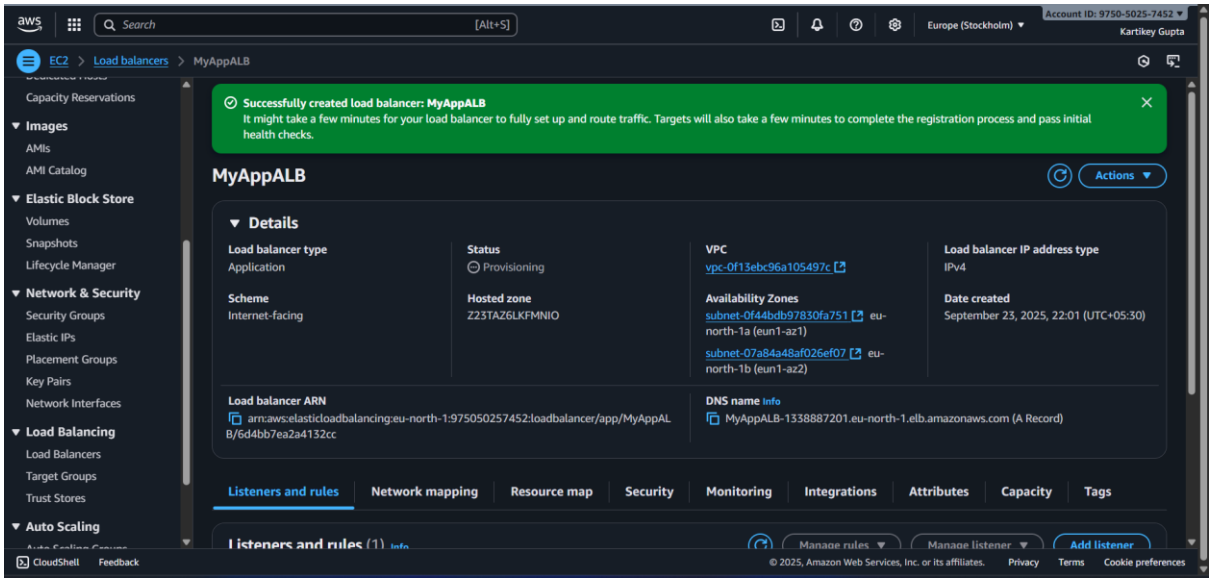
Step 4: Attach Target Group

1. Under **Listener → Default action**, select **Forward to → Your Target Group** (e.g., MyApp-TG).



Step 5: Review and Create

- Review all settings and click **Create Load Balancer**.
- Note the **ALB DNS name** — this is what users will access.



Step (iii): Configure Health Checks

Step 1: Open Target Groups

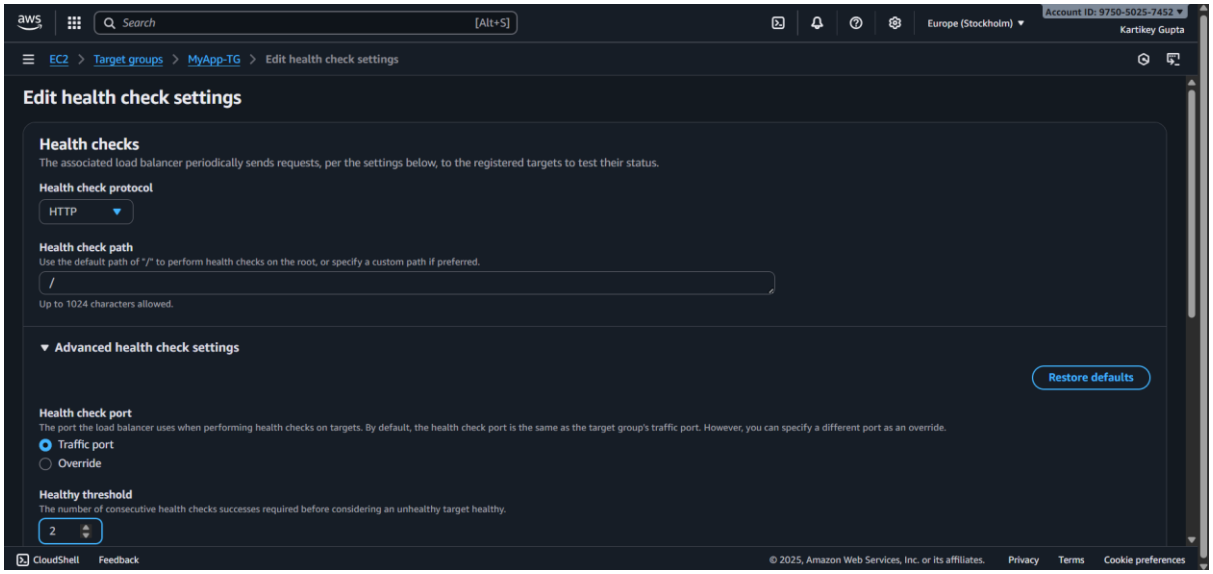
1. Go to **EC2** → **Target Groups** in the AWS console.
2. Select the **Target Group** you created (e.g., MyApp-TG).

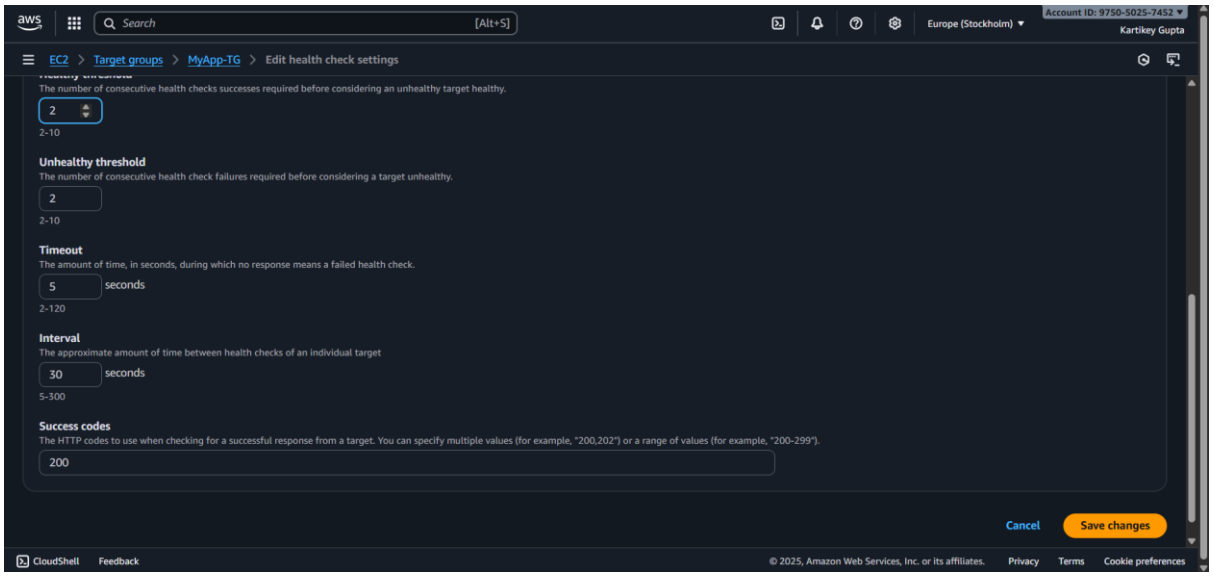
Step 2: Go to Health Checks

1. In the Target Group details, click the **Health checks** tab.

Step 3: Configure Health Check Settings

- **Protocol:** HTTP
- **Path:** / (or /health if you have a dedicated health endpoint)
- **Healthy threshold:** 2 → instance must pass 2 consecutive checks to be marked healthy
- **Unhealthy threshold:** 2 → instance must fail 2 consecutive checks to be marked unhealthy
- **Timeout:** 5 seconds → how long to wait for a response
- **Interval:** 30 seconds → time between checks



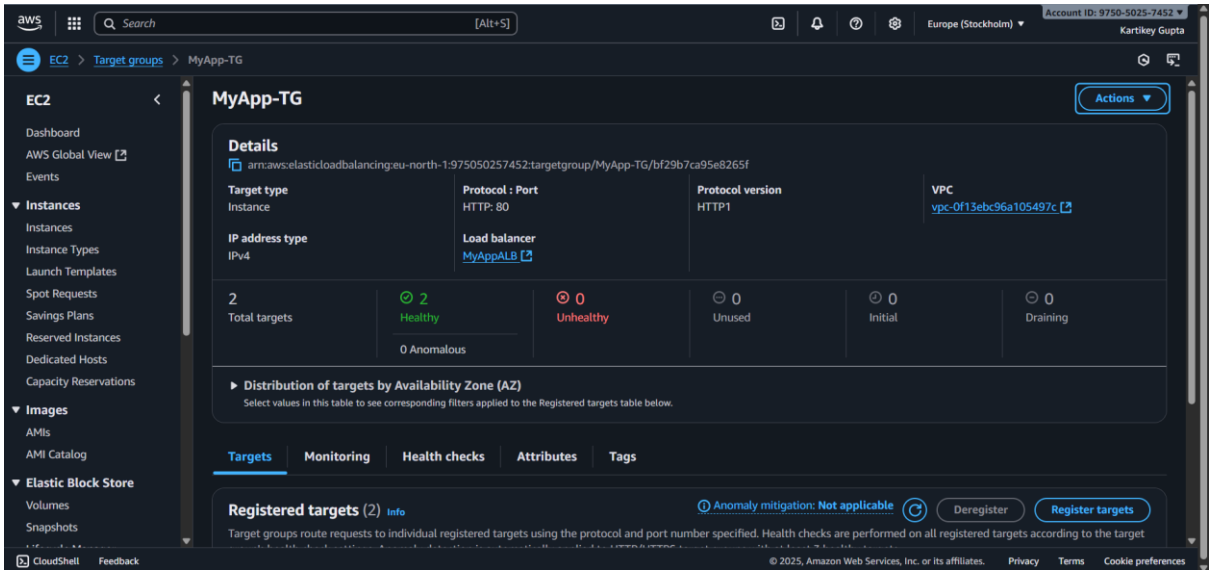


Step 4: Save

- Click **Save** to apply health check settings.

Step 5: Verify

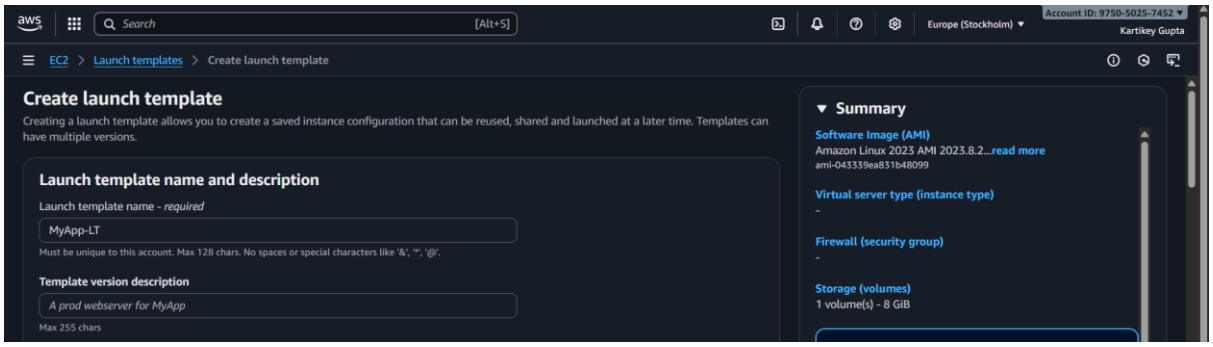
- Go back to **Targets** tab in the Target Group.
- After a few seconds/minutes, the **Status** column should show **healthy** for your EC2 instances.
- If an instance fails, check:
    - Security group allows inbound HTTP from ALB
    - Web server is running and responding at the path you configured

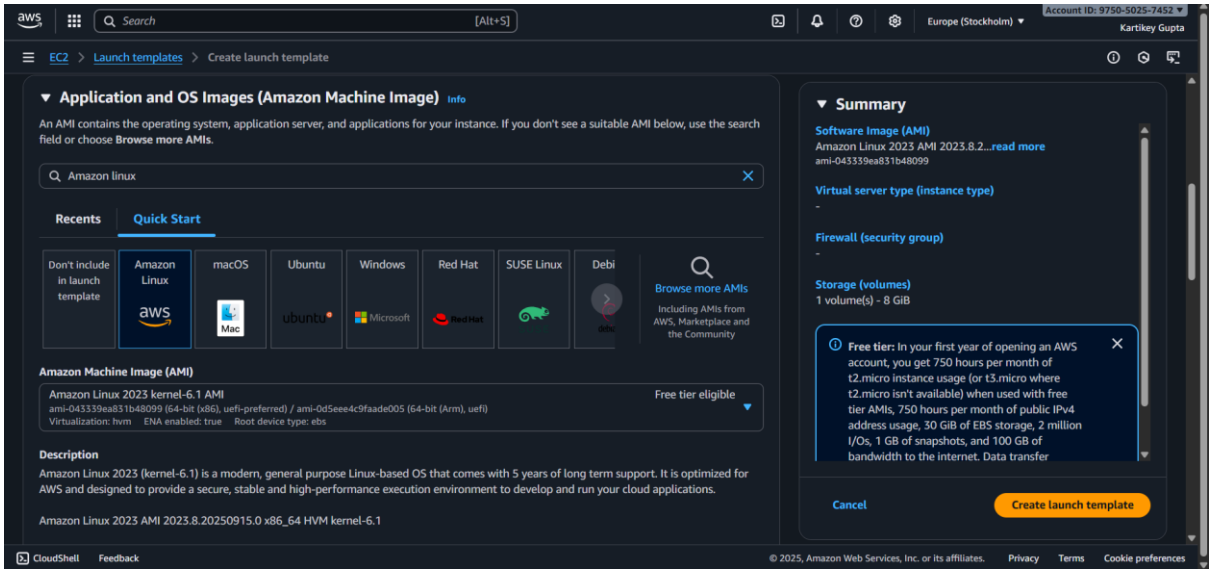


Step (iv): Enable Auto Scaling

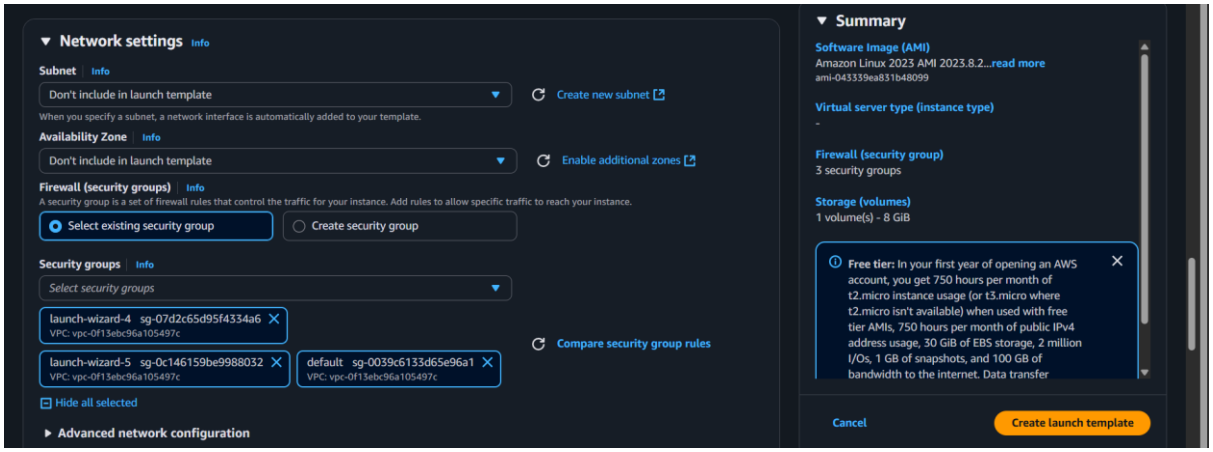
Step 1: Prepare a Launch Template

1. Go to **EC2 → Launch Templates → Create Launch Template**.
2. Configure template:
    - **Name:** MyApp-LT
    - **AMI:** Use the same AMI as your EC2 instances



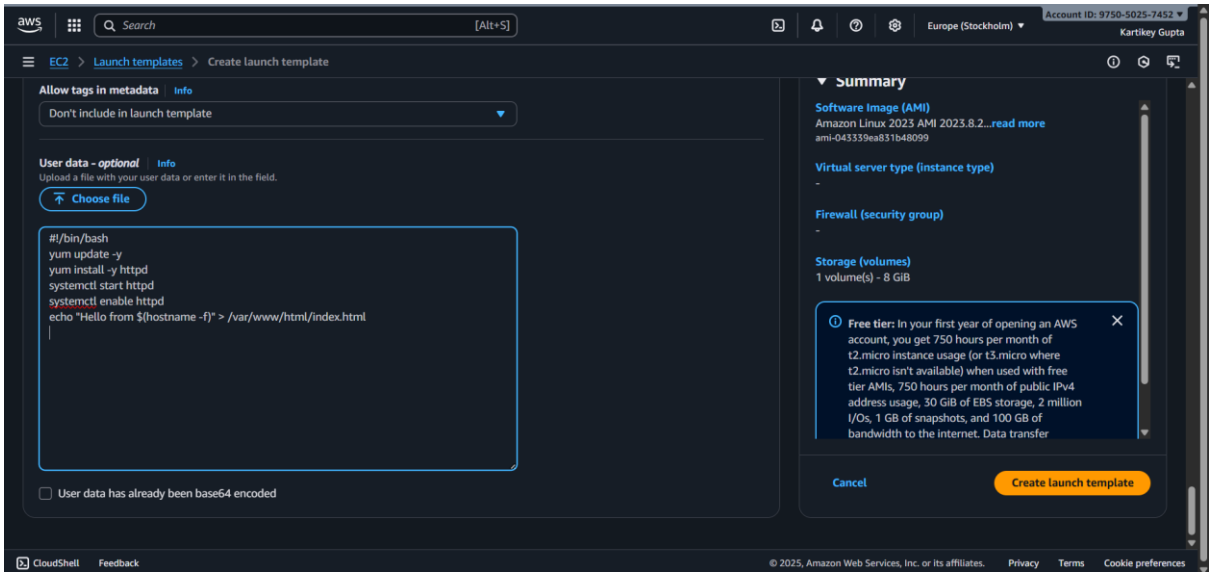


- **Instance type:** e.g., t2.micro
- **Key pair:** Your existing key
- **Security group:** Same as your running EC2 (allow HTTP 80 from ALB SG)

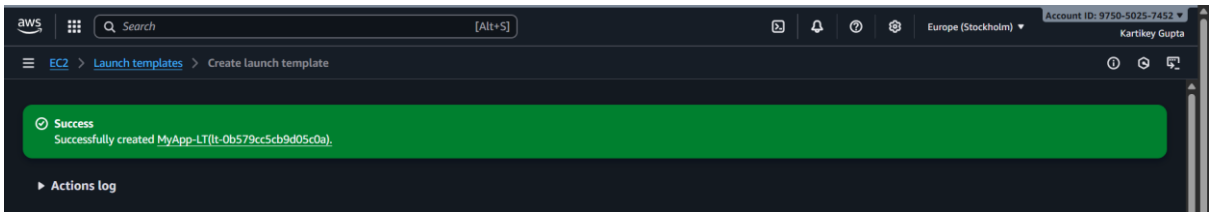


- **User data (optional):** Script to auto-install Apache/web server on launch:

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Hello from $(hostname -f)" > /var/www/html/index.html
```

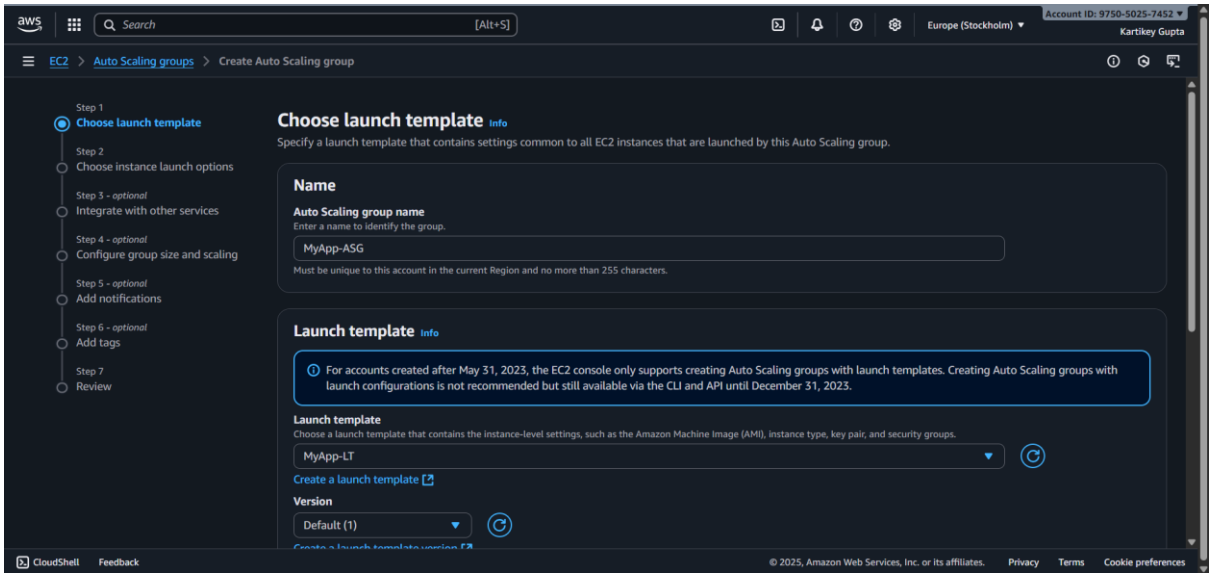


3. Save the Launch Template.



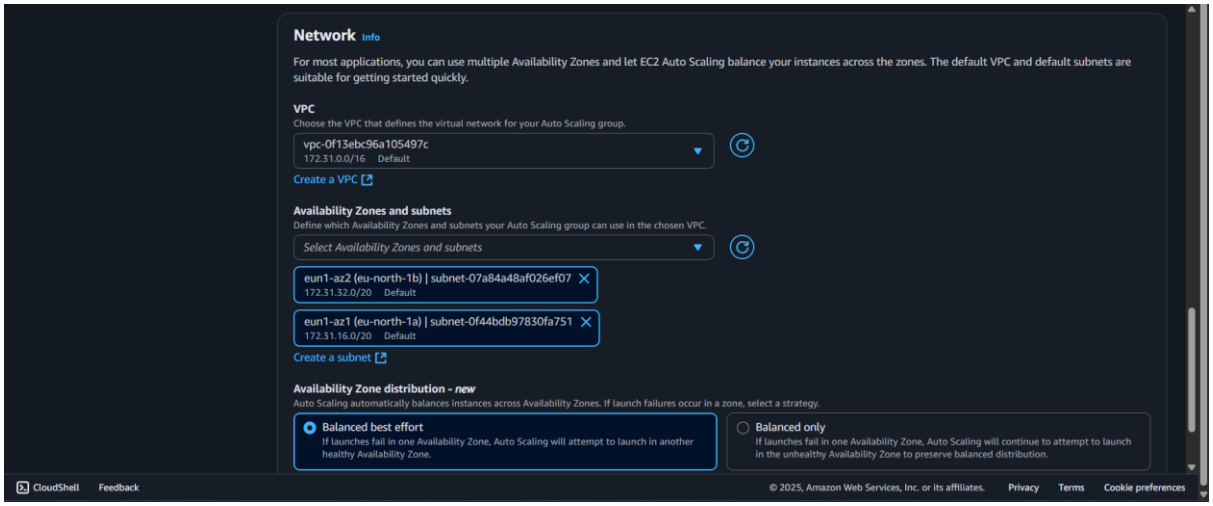
Step 2: Create Auto Scaling Group

1. Go to **EC2 → Auto Scaling Groups → Create Auto Scaling Group**.
2. **Select launch template:** MyApp-LT
3. **Name:** MyApp-ASG



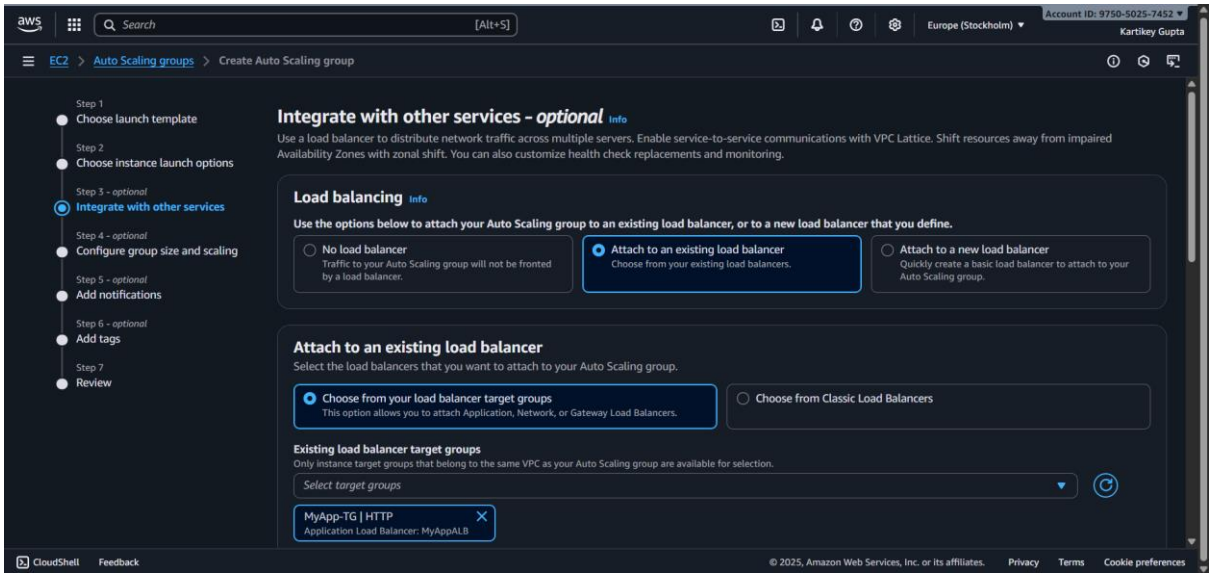
Step 3: Configure Network

1. **VPC:** Same as your instances/ALB
2. **Subnets:** Select at least **2 AZs** for high availability



Step 4: Attach to Target Group

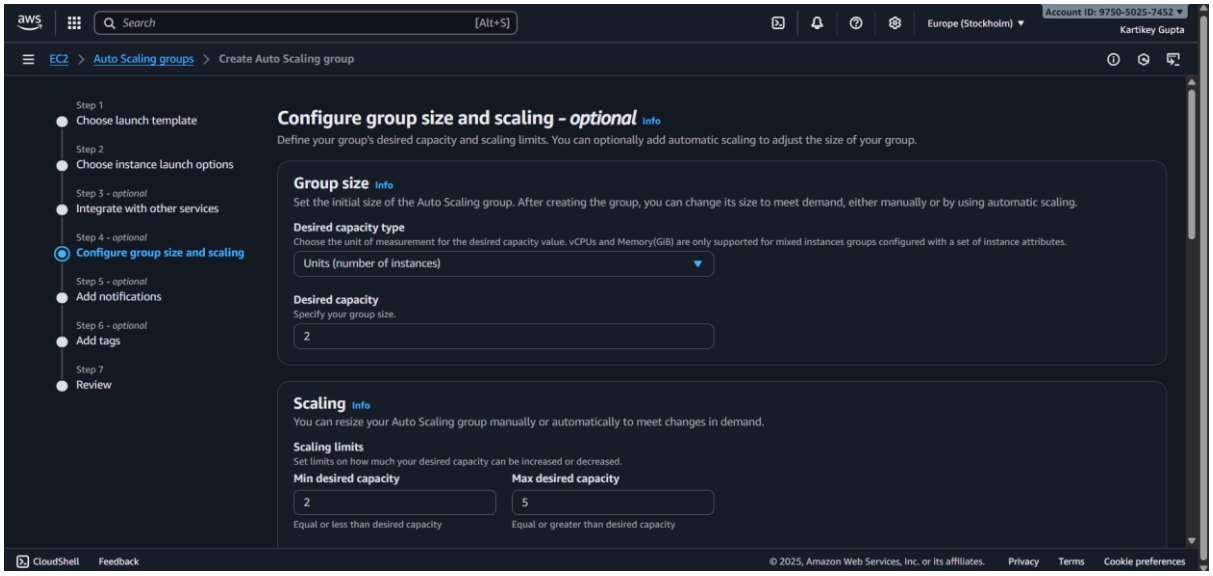
1. Select **"Attach to an existing load balancer"**
2. Choose your **ALB Target Group** (MyApp-TG)



Step 5: Configure Group Size

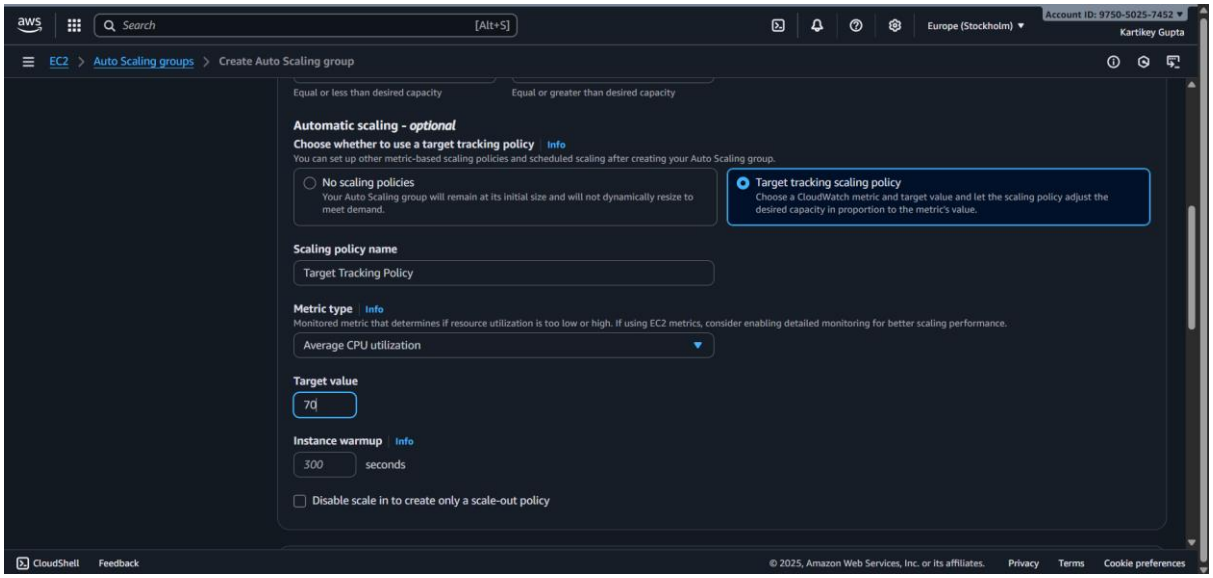
- **Desired capacity:** 2 → initially 2 instances
- **Minimum:** 2 → ensures at least 2 instances are always running

- **Maximum:** 5 → allows scaling up to 5 instances if needed



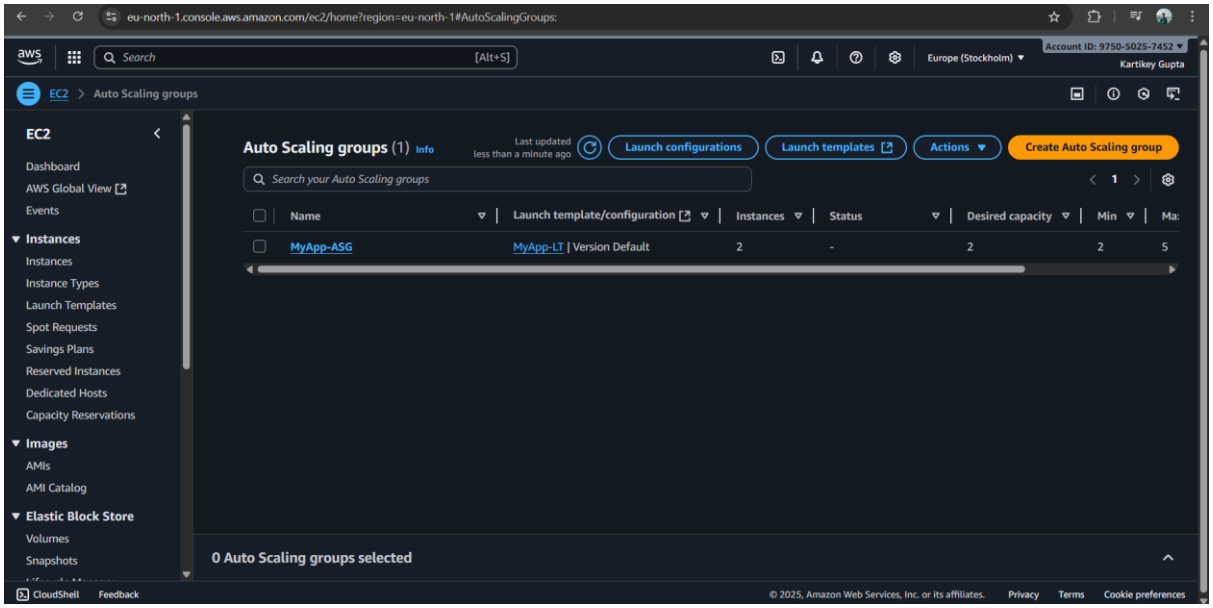
Step 6: Add Scaling Policies

1. **Target tracking / simple scaling**
2. Example policy:
    - **Scale out:** CPU > 70% → add instance(s)



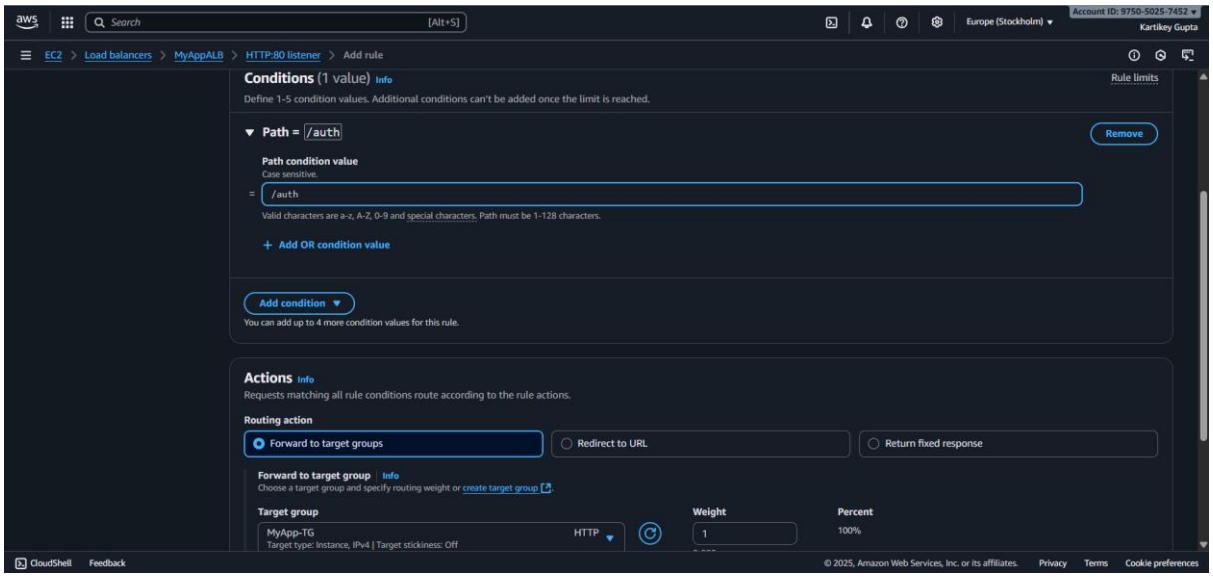
Step 7: Review and Create

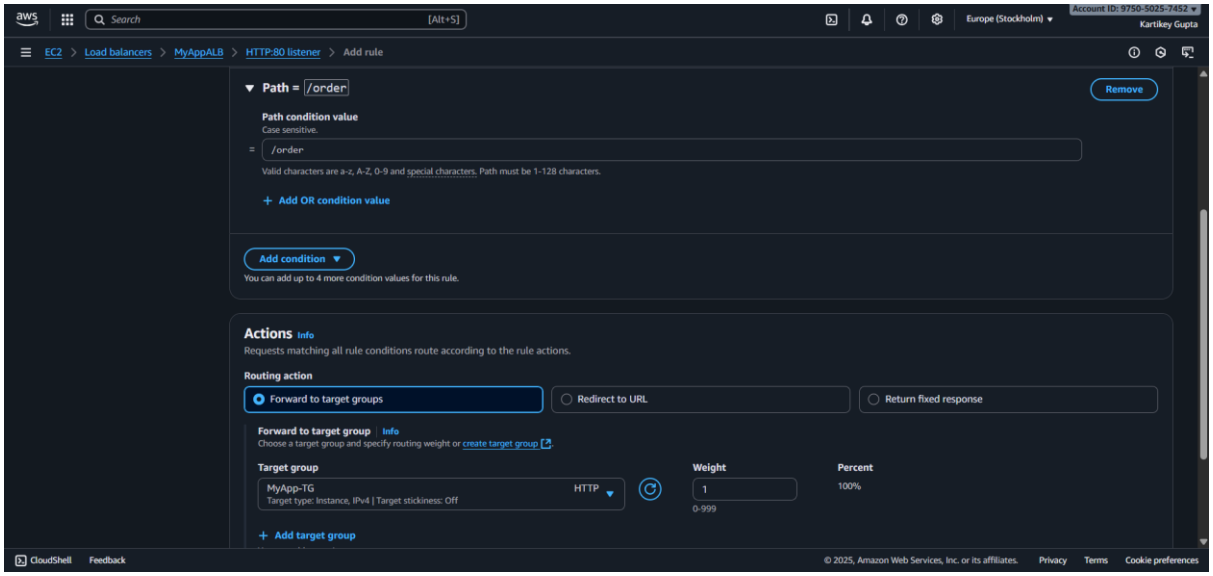
- Review all settings → click **Create Auto Scaling Group**
- Auto Scaling will now automatically launch 2 instances initially, register them with ALB, and scale based on traffic.

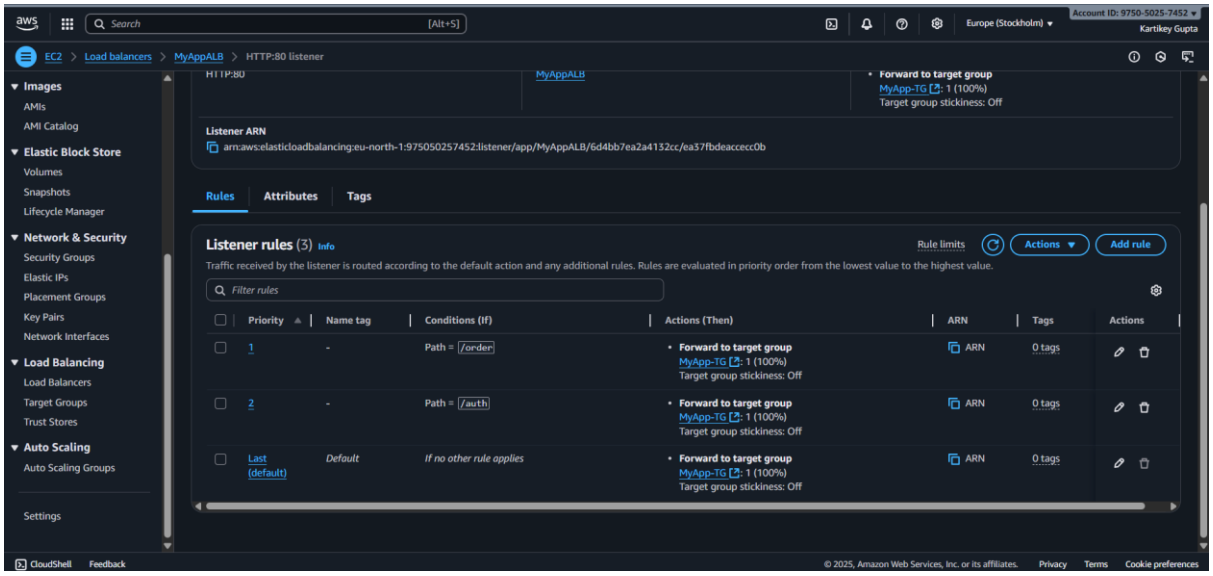


(v) Configure Path-Based Routing + Route 53

1. Go to **ALB → Listeners → View/Edit rules**.
2. Add rules:
    - If path is /auth → forward to **Target Group: AuthService**.
    - If path is /order → forward to **Target Group: OrderService**.
3. Ensure you created **separate EC2 groups** (Auth, Order) running respective services.
4. Save listener rules.




Search [Alt+S]
Europe (Stockholm)
Account ID: 9750-5025-7452
Kartikey Gupta
EC2 > Load balancers > MyAppALB > HTTP:80 listener > Add rule
Path = /order
Remove
Path condition value
Case sensitive.
/order
Valid characters are a-z, A-Z, 0-9 and special characters. Path must be 1-128 characters.
Add OR condition value
Add condition
You can add up to 4 more condition values for this rule.
Actions Info
Requests matching all rule conditions route according to the rule actions.
Routing action
Forward to target groups
Redirect to URL
Return fixed response
Forward to target group | Info
Choose a target group and specify routing weight or create target group
Target group
MyApp-TG
HTTP
Target type: Instance, IPv4 | Target stickiness: Off
Weight
1
0-999
Percent
100%
Add target group
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Search [Alt+S]
Europe (Stockholm)
Account ID: 9750-5025-7452
Kartikey Gupta
EC2 > Load balancers > MyAppALB > HTTP:80 listener
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores
Auto Scaling
Auto Scaling Groups
Settings
HTTP:80
MyAppALB
Forward to target group
MyApp-TG: 1 (100%)
Target group stickiness: Off
Listener ARN
arn:aws:elasticloadbalancing:eu-north-1:975050257452:listener/app/MyAppALB/6d4bb7ea2a4132cc/ea37fbdeaccecc0b
Rules
Attributes
Tags
Listener rules (3) Info
Rule limits
Actions
Add rule
Traffic received by the listener is routed according to the default action and any additional rules. Rules are evaluated in priority order from the lowest value to the highest value.
Filter rules
Priority
Name tag
Conditions (If)
Actions (Then)
ARN
Tags
Actions
1
-
Path = /order
Forward to target group
MyApp-TG: 1 (100%)
Target group stickiness: Off
ARN
0 tags
2
-
Path = /auth
Forward to target group
MyApp-TG: 1 (100%)
Target group stickiness: Off
ARN
0 tags
Last (default)
Default
If no other rule applies
Forward to target group
MyApp-TG: 1 (100%)
Target group stickiness: Off
ARN
0 tags
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences