Placement Empowerment Program

Cloud Computing and DevOps Centre

Classic Load Balancer & Auto Scaling in AWS

Name: Janani E Department : CSE

Introduction:

In today's PoC, I worked on implementing Classic Load Balancer (CLB) and Auto Scaling in AWS to ensure efficient traffic distribution and automatic resource management.

What is Load Balancer?

A **Load Balancer** in AWS is a service that automatically distributes incoming traffic across multiple targets (such as EC2 instances, containers, and IP addresses) in different **Availability Zones** to ensure high availability, fault tolerance, and scalability.

What is Auto Scaling?

Auto Scaling in AWS is a feature that **automatically adjusts the number of EC2 instances** in response to traffic demand. It ensures that the right number of instances are running to handle the load efficiently, reducing costs and improving availability.

Benefits:

✅1. Hands-on Experience with Load Balancing

- Understands how **Classic Load Balancer (CLB)** distributes traffic across multiple EC2 instances.
- Ensures high **availability** and **fault tolerance**.

✅2. Improved Understanding of Auto Scaling

- Learns how to **dynamically scale** EC2 instances based on real-time demand.
- Ensures applications run efficiently without manual intervention.

✅3. Cost Optimization

- Prevents over-provisioning by **scaling down** instances when demand is low.
- Ensures **cost-efficient** resource management.

✅6. Real-World Cloud Deployment Skills

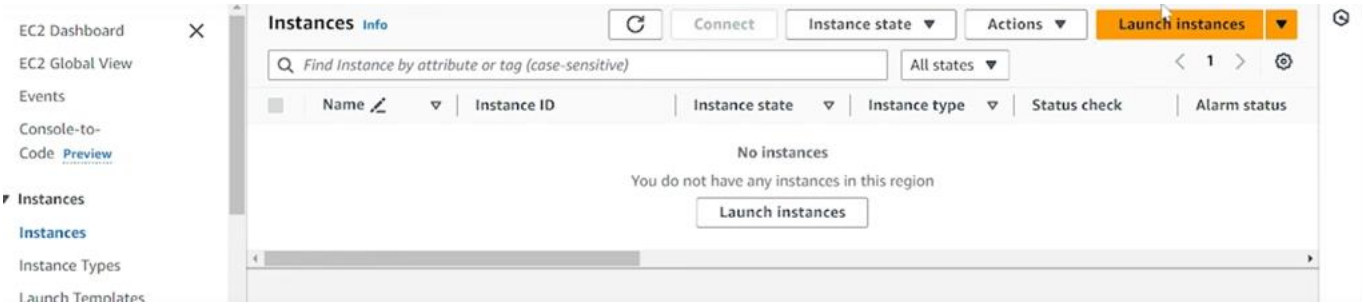
- Builds practical skills useful for **cloud-based applications and DevOps**.
- Prepares for **AWS certification exams** and real-world projects.

Step-by-Step Overview

Step1: Log in to AWS Console

1.Go to AWS Management Console.

2. Navigate to EC2.

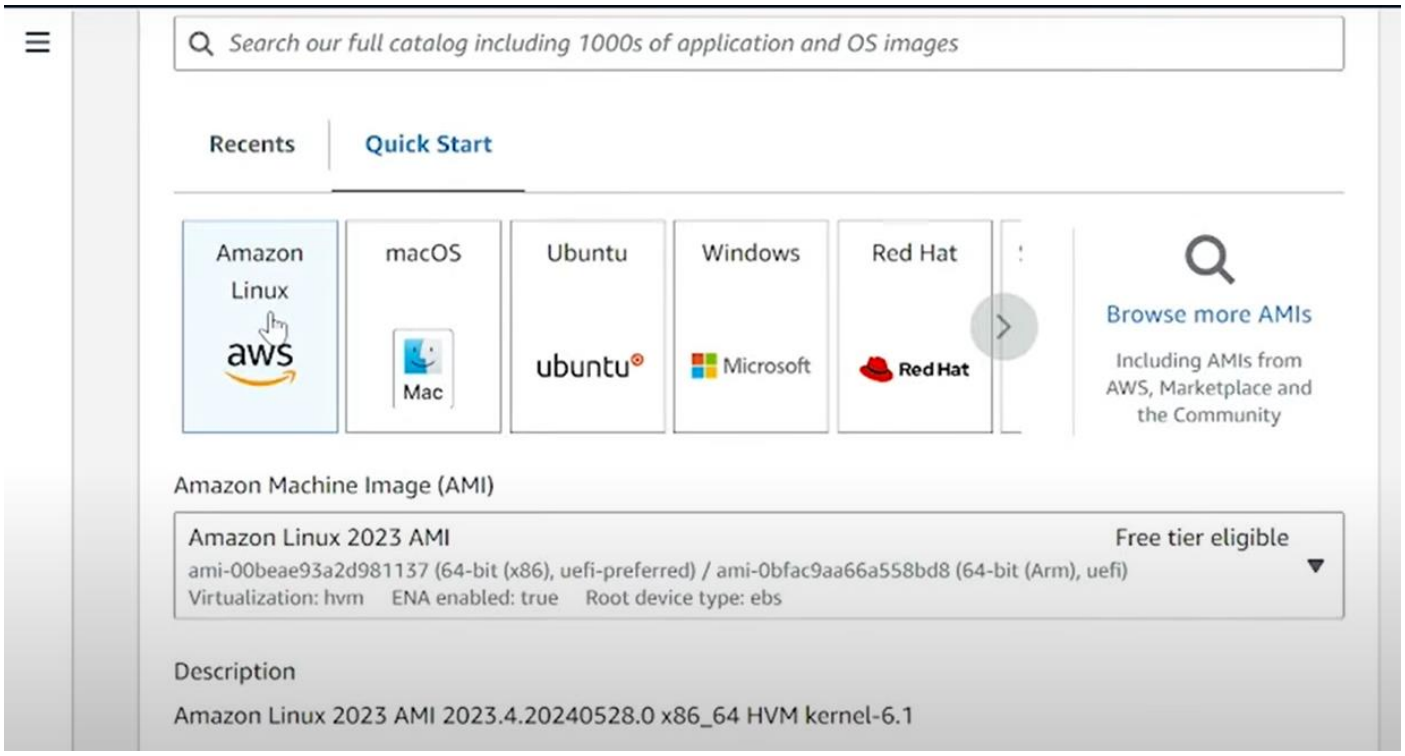


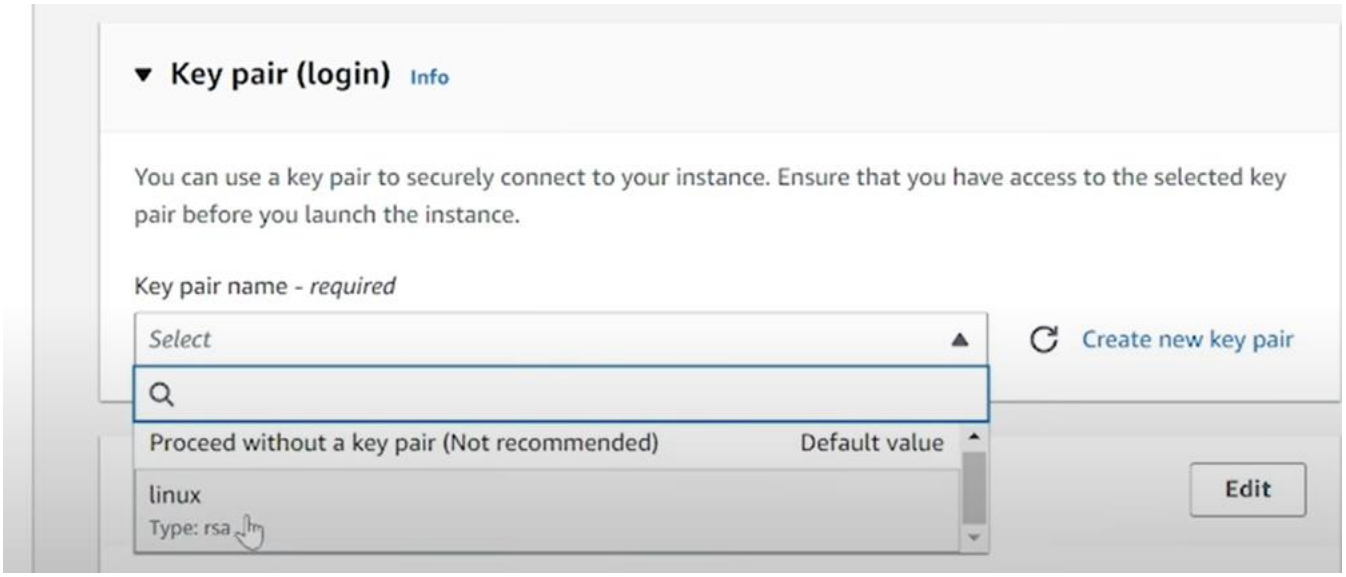
Step 2 :Create Two EC2 Instances

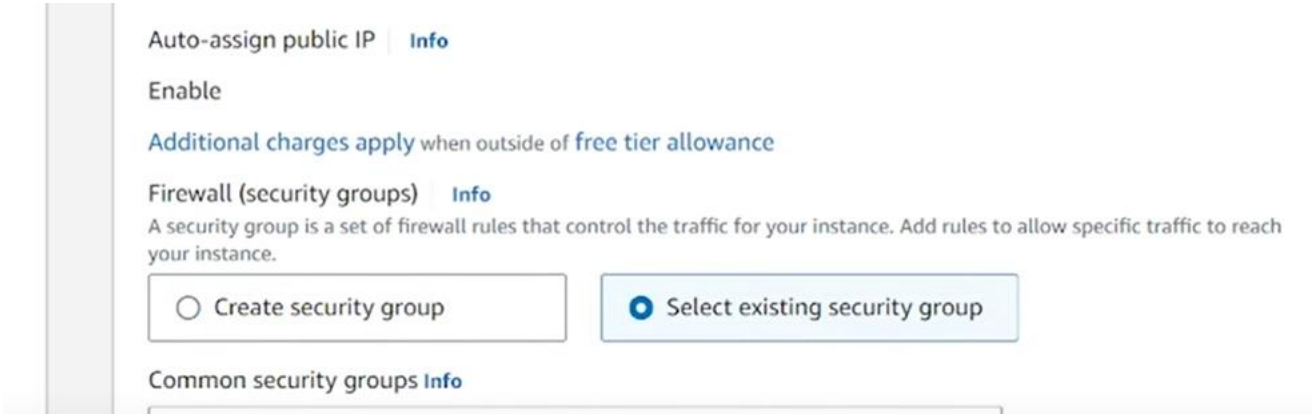
1. In the EC2 Dashboard, click Launch Instances.
2. 2. Choose an Amazon Linux 2 or Ubuntu AMI.
3. 3. Select t2.micro (free tier eligible).

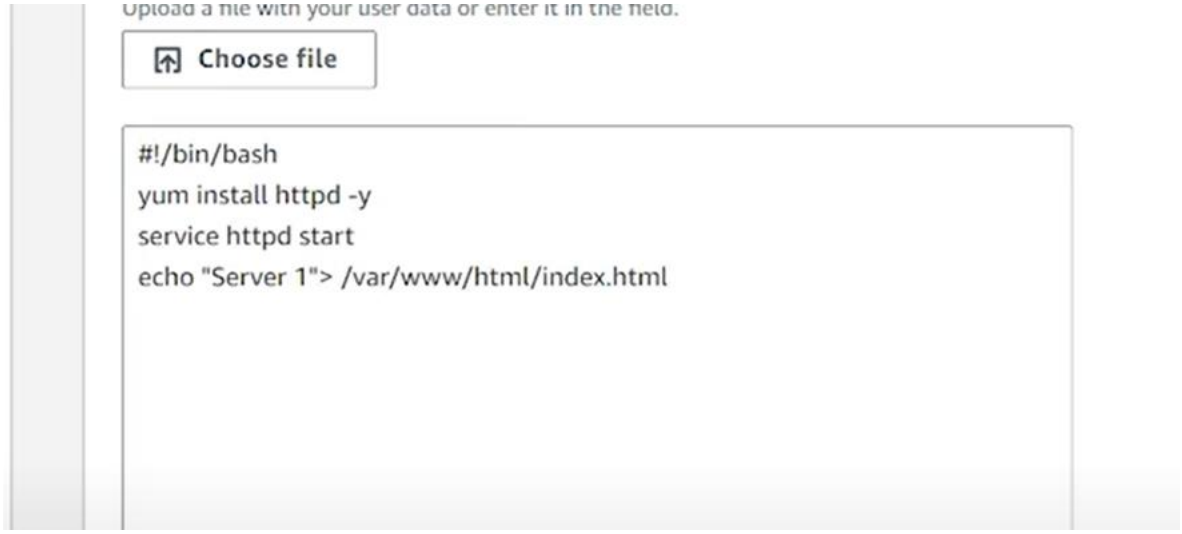
4. Configure the instance: • Number of instances: 2 • Keep default settings. 5. Add User Data (optional) to install a web server: • In Advanced Details, under User Data, enter:


Name and tags Info
Name
Server 1
Add additional tags
Application and OS Images (Amazon Machine Image) Info


Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
macOS
Mac
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community
Amazon Machine Image (AMI)
Amazon Linux 2023 AMI
Free tier eligible
ami-00beae93a2d981137 (64-bit (x86), uefi-preferred) / ami-0bfac9aa66a558bd8 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs
Description
Amazon Linux 2023 AMI 2023.4.20240528.0 x86_64 HVM kernel-6.1

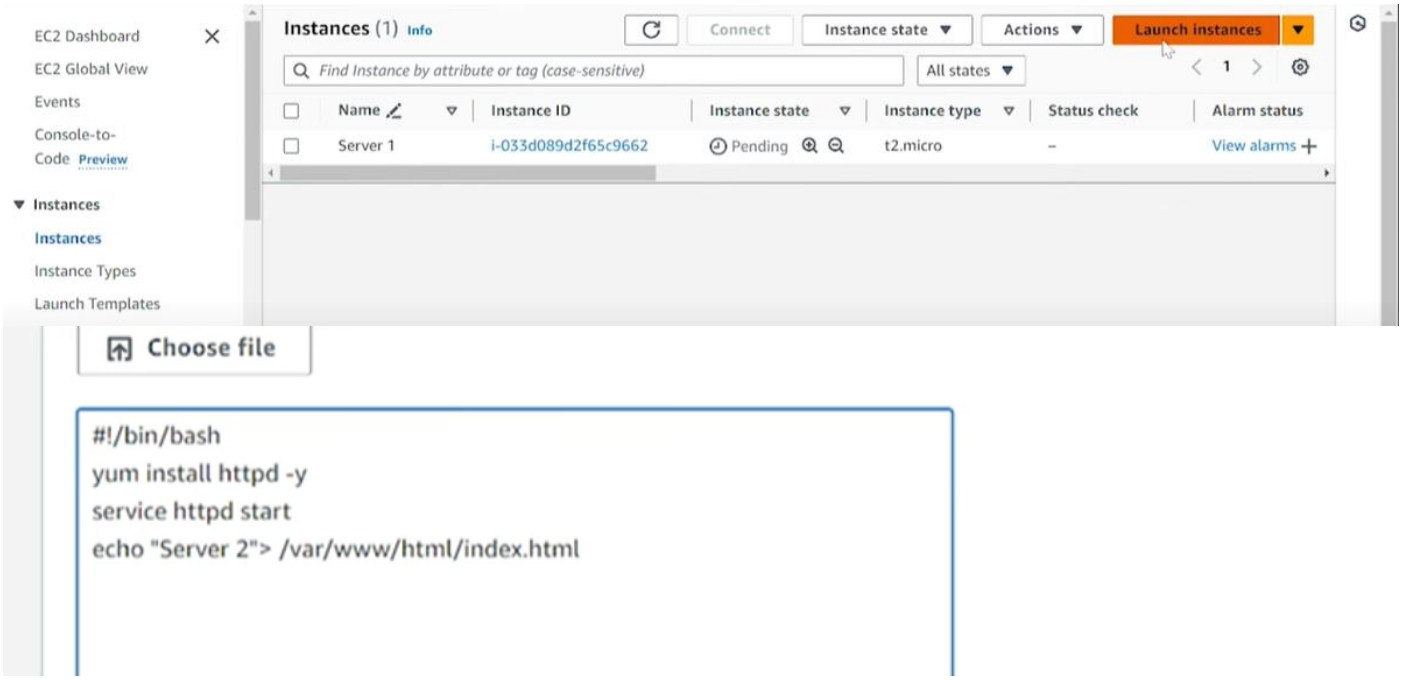

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.
Key pair name - required
Select
Create new key pair
Proceed without a key pair (Not recommended)
Default value
linux
Type: rsa
Edit

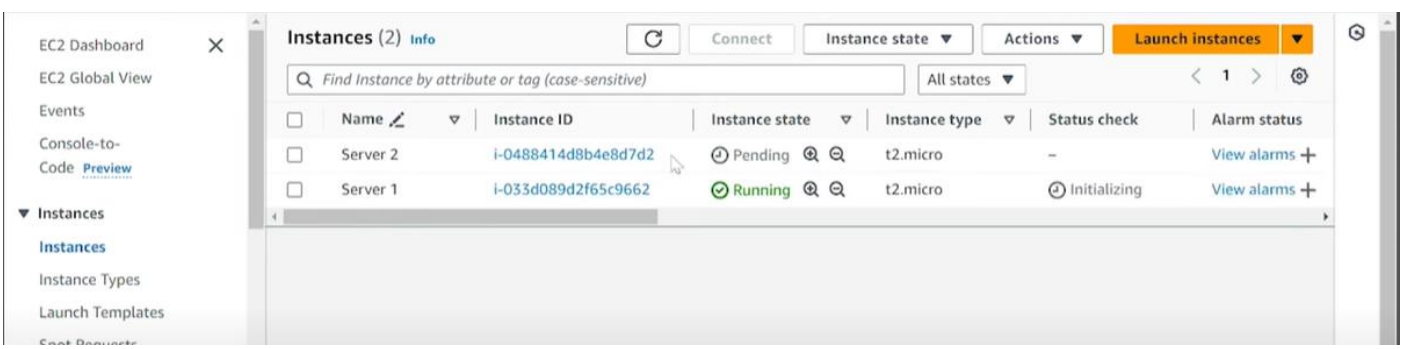

Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
Common security groups Info



Step 3 :

For **Server 2**, follow the same steps, but modify the **User Data script** to display a different message.





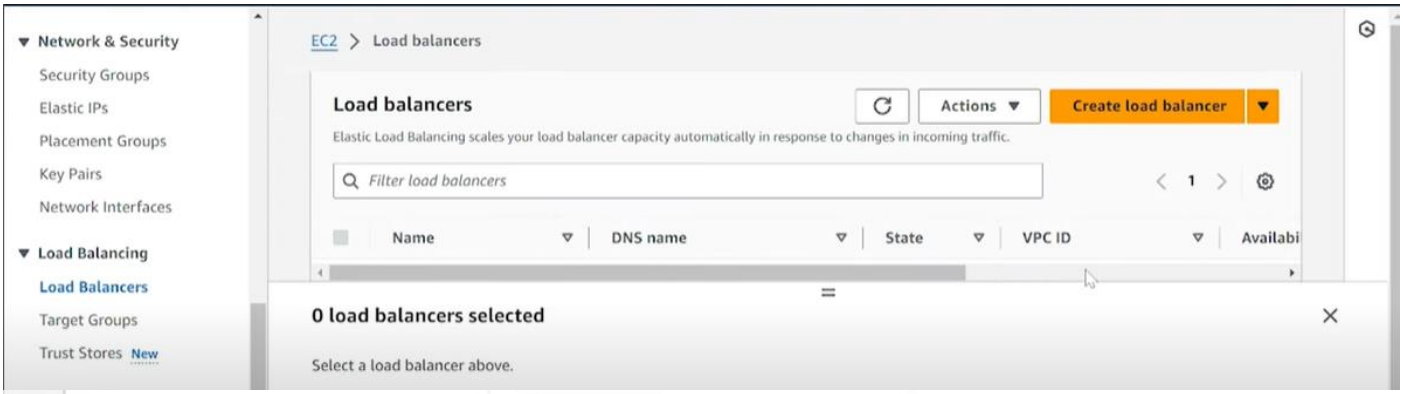
Step 4 :

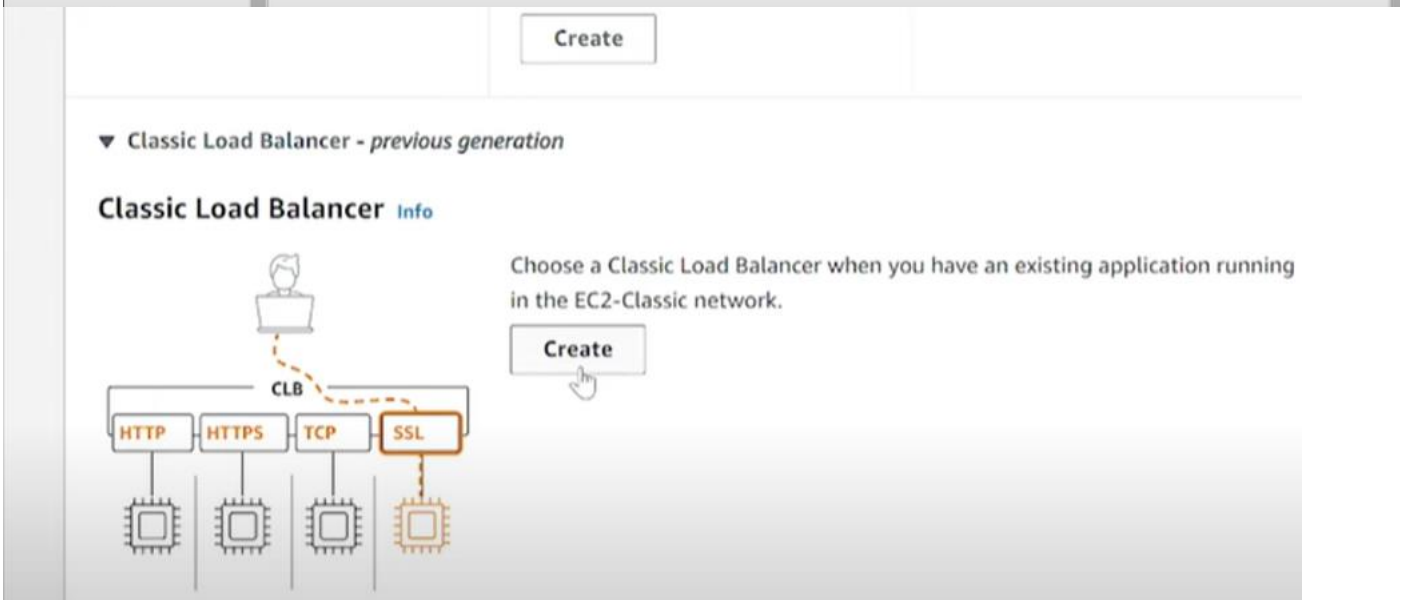
Select Load Balancer Type

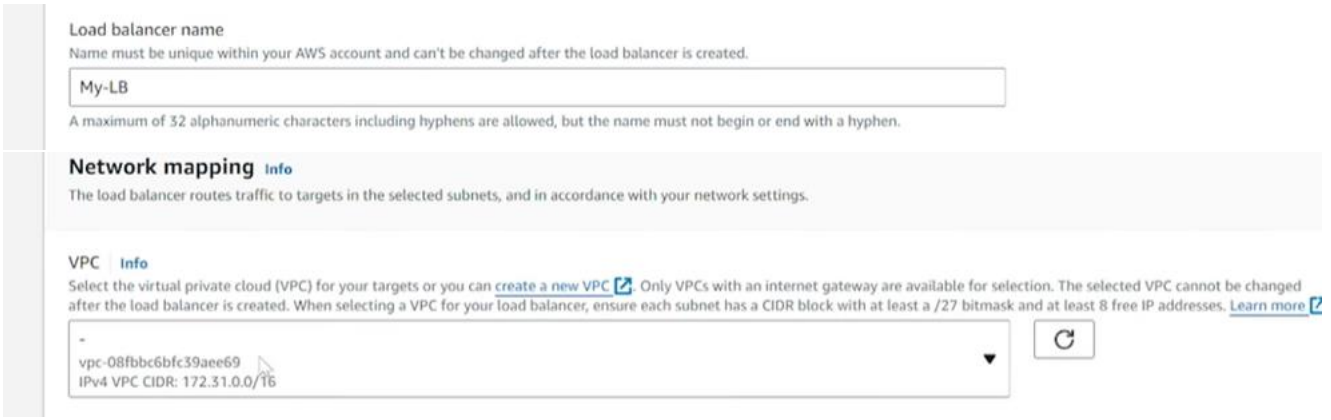
1. Choose **"Classic Load Balancer."**
2. Click **"Create."**

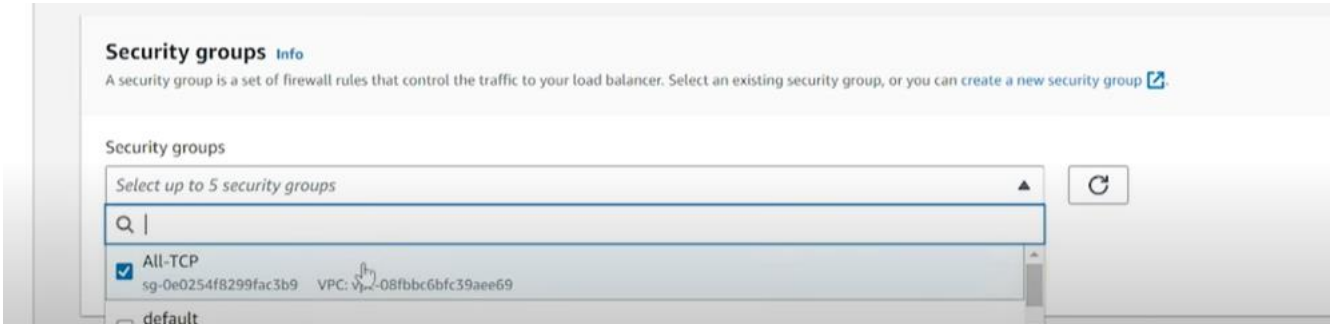
Configure Load Balancer Settings

1. **Name:** Enter a unique name (e.g., MyClassicLB).
2. **Scheme:** Select **Internet-facing** (for public access) or **Internal** (for private use).
3. **Listener Configuration:**
   - **Protocol:** HTTP
   - **Port:** 80 (default for web traffic)
4. **VPC & Availability Zones:**
   - Choose a **VPC** where your EC2 instances are running.
   - Select at least **two Availability Zones** for high availability.









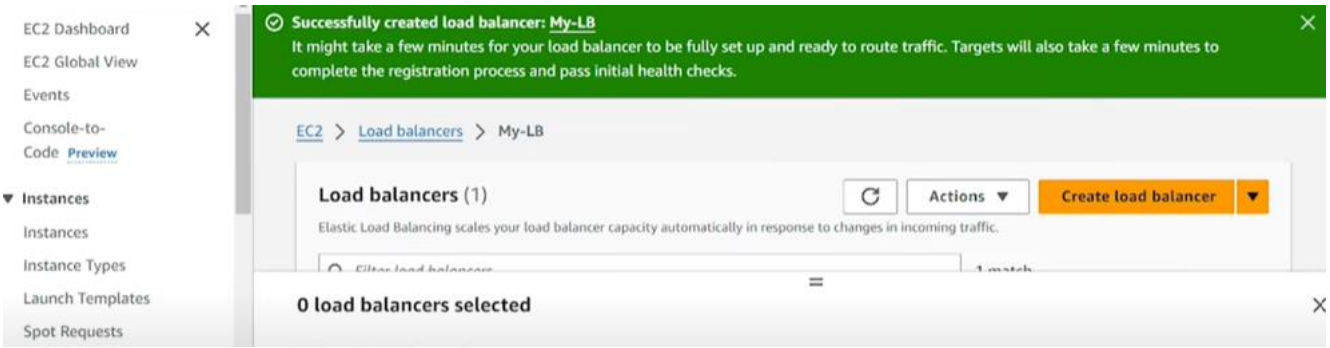
Step 5 :

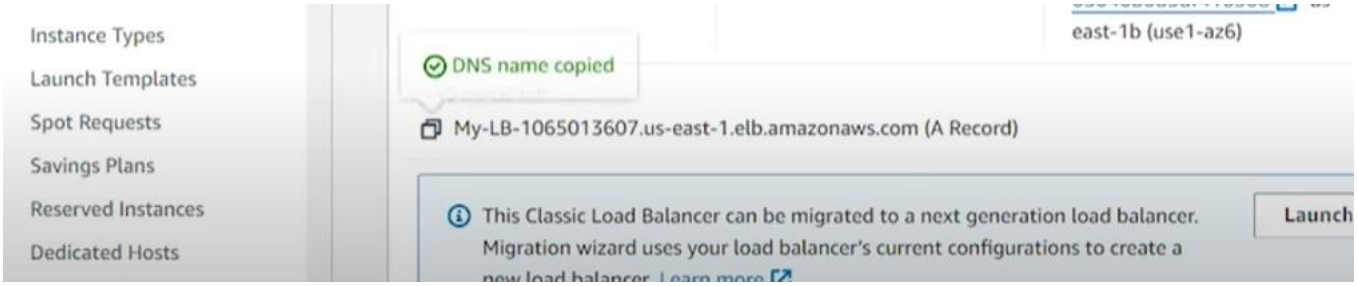
Copy the **DNS Name** of your Load Balancer.

- Open a web browser and paste the DNS link in the address bar.
- For example, http://myclassiclb-1234567890.us-west-2.elb.amazonaws.com.

Verify the Web Server Output

- **If the Load Balancer is correctly configured**, the browser should show the **web page** from one of the EC2 instances (e.g., **Server 1** or **Server 2**).
- **Refresh the page** multiple times. You should see the page alternate between **Server 1** and **Server 2** (depending on the load balancing).

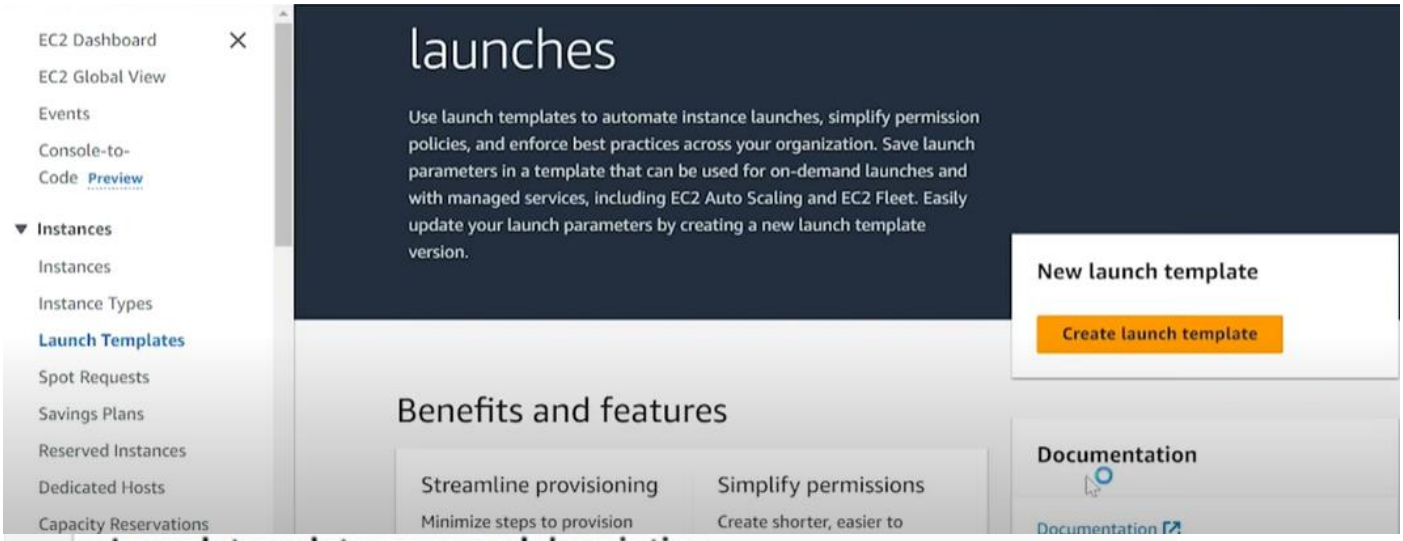


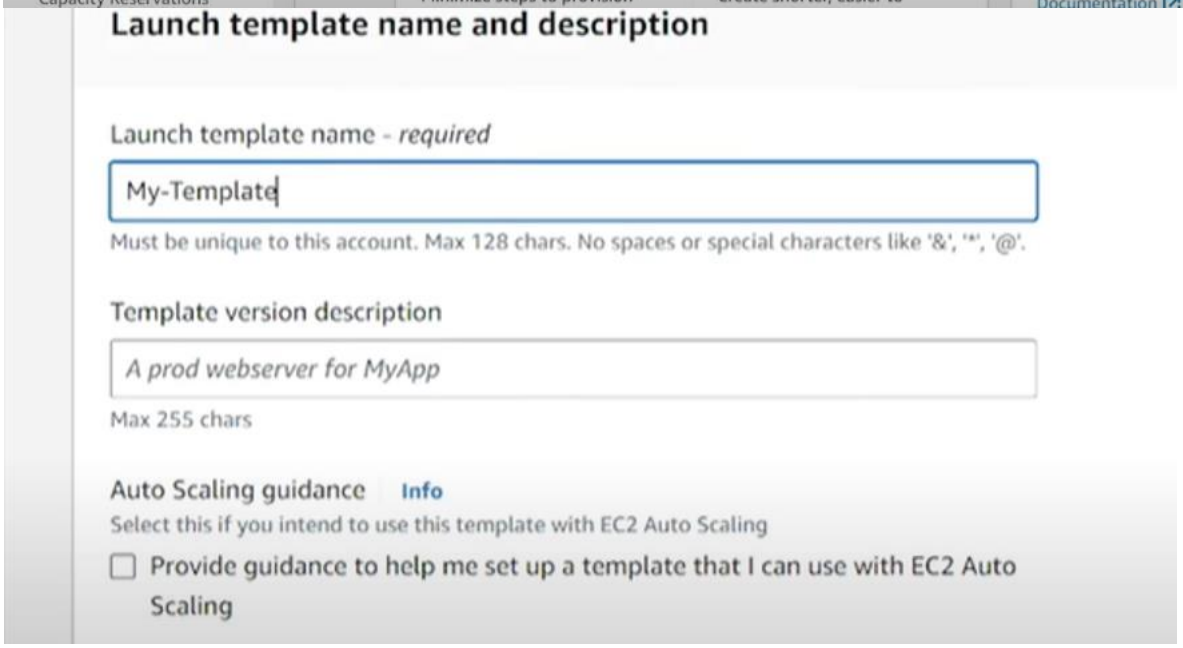


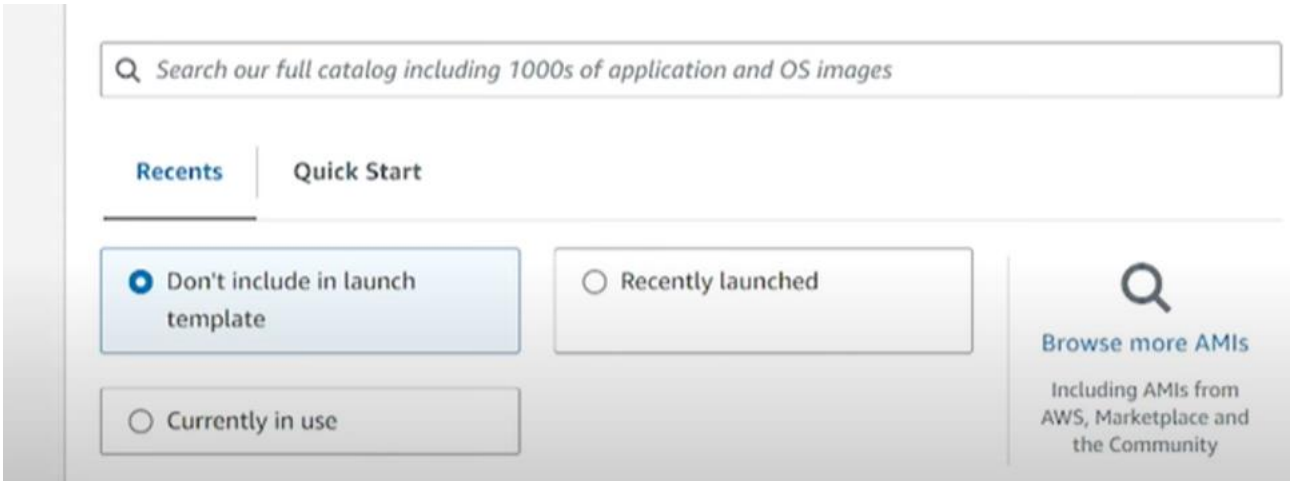
Step 6 :

Create Launch Template

1. In the **Launch Templates** section, click **Create launch template**.
2. Fill in the following information:
    - **Template Name:** My-Template.
    - **Template Version Description:** My-Template.
    - **AMI ID:** Choose the **Amazon Linux 2** or **Ubuntu** AMI you selected for Server 1 and Server 2.
    - **Instance Type:** t2.micro (or any other type that fits your needs).
    - **Key Pair:** Select an existing key pair or create a new one for SSH access.
    - **Network:** Select the **VPC** you are using (ensure both EC2 instances are in the same VPC).
    - **Subnet:** Choose a subnet or use the default.
    - **Security Group:** Select the security group that allows HTTP (port 80) and HTTPS (port 443).
    - **IAM Role (Optional):** Select if needed.







Step 7 :

Creating AMI for Server 1

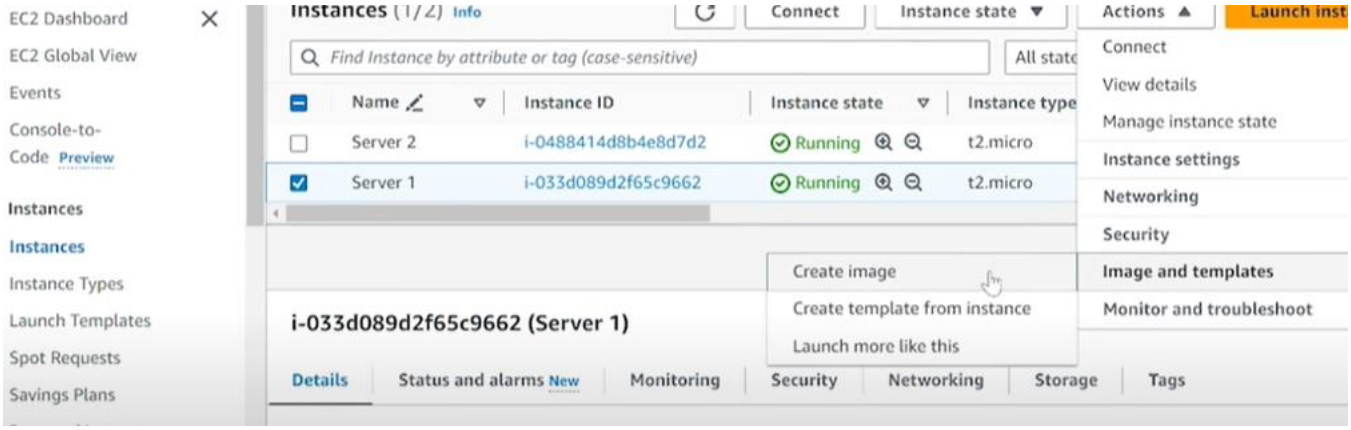
1. **Log in to AWS Console** and navigate to **EC2**.
2. In the **Instances** section, select **Server 1**.
3. Click on **Actions → Image and templates → Create image**.
    - Provide a **name** for the image (e.g., Server1-AMI).

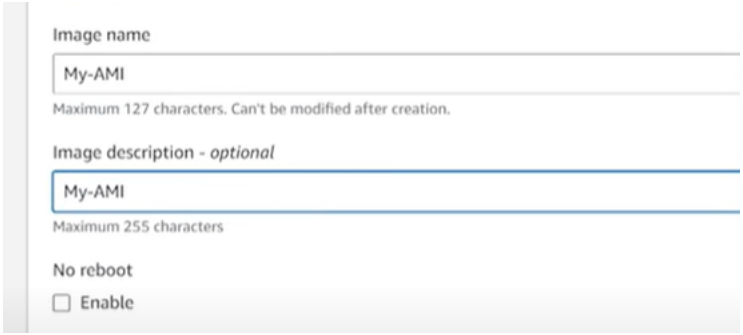
- Optionally, add a description.
- Select **No Reboot** if you do not want to reboot the instance.

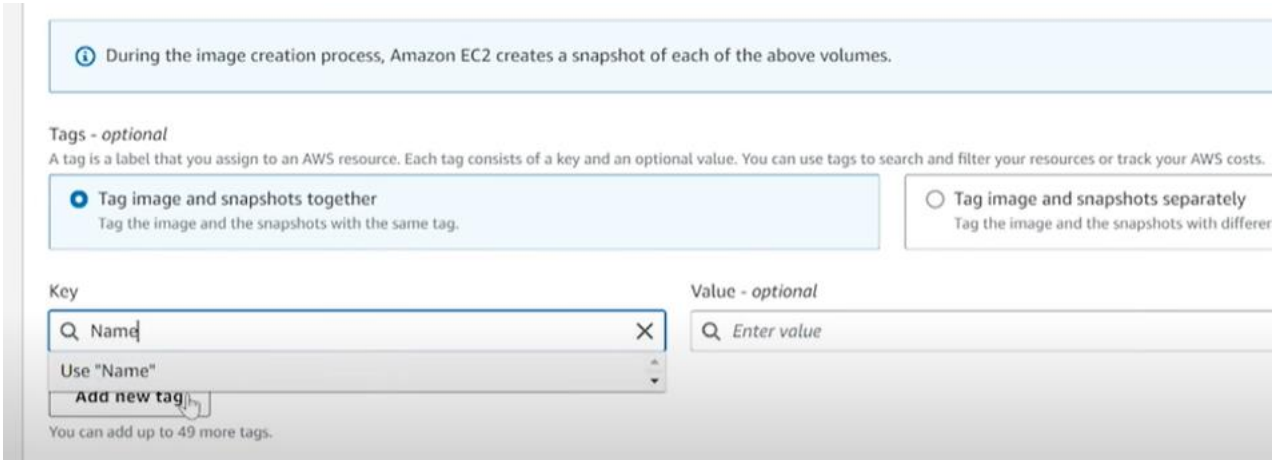
4. Click **Create Image** and wait for the image creation to complete.
5. Once ready, the **AMI** will be available in the **AMIs** section under **Images**.

Or Creating AMI for Server 2

1. **Select Server 2** in the **Instances** section.
2. Repeat the same steps:
   - Click **Actions → Image and templates → Create image**.
   - Name the image (e.g., Server2-AMI).
3. Click **Create Image** and wait for the process to complete.
4. **Server 2's AMI** will appear in the **AMIs** section once ready.

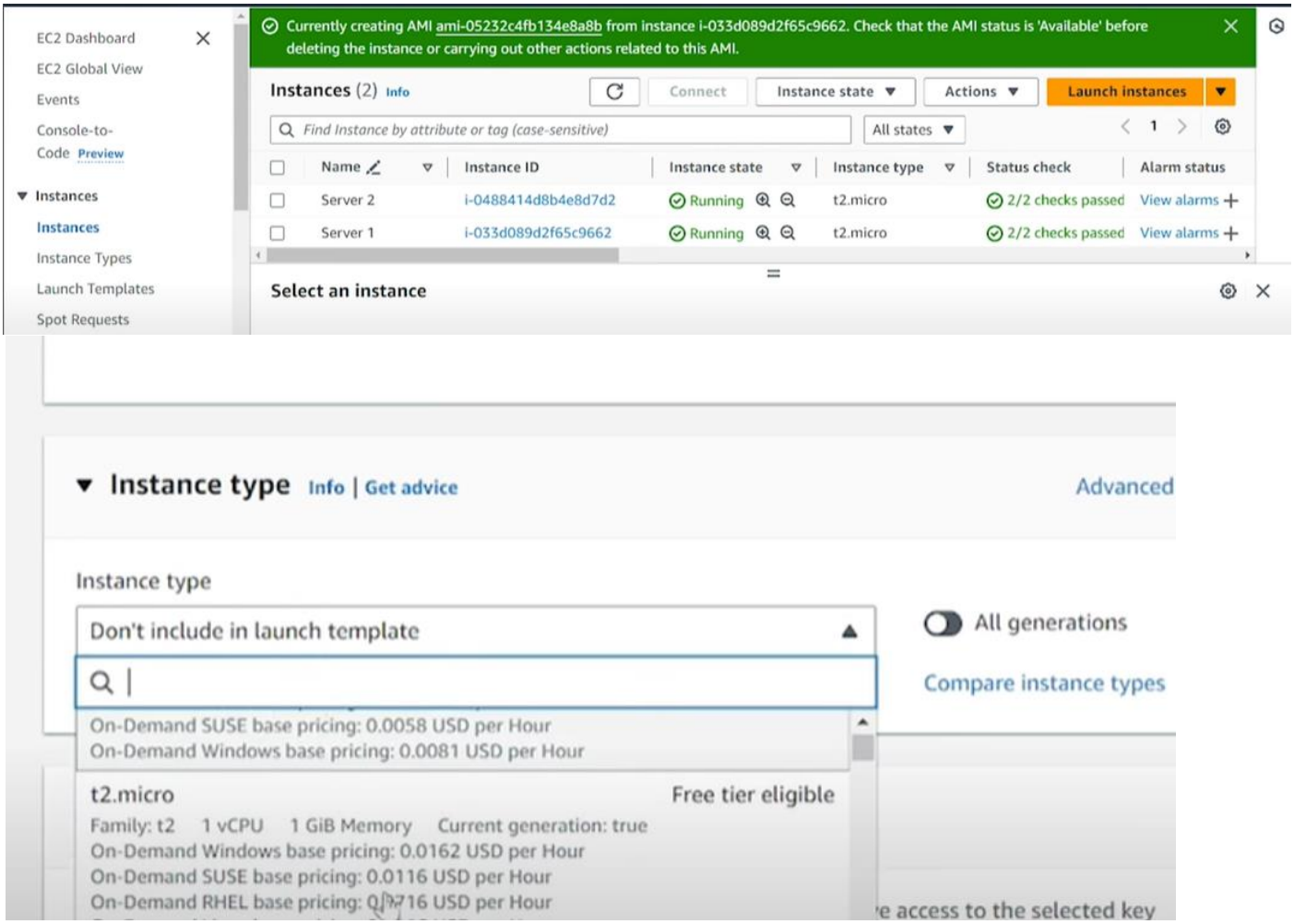


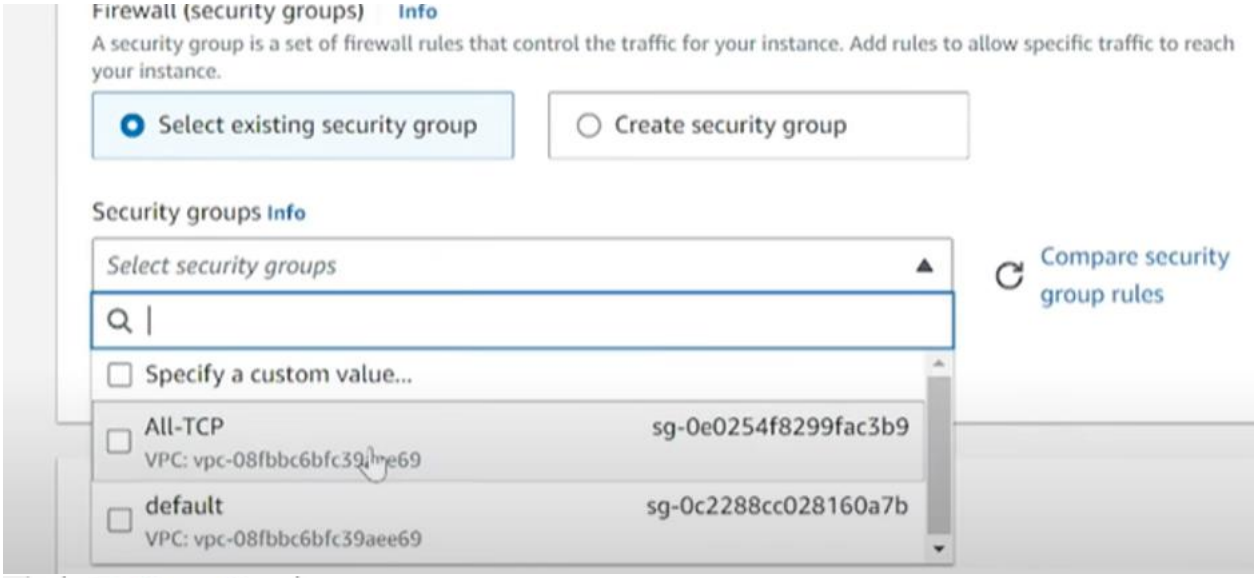


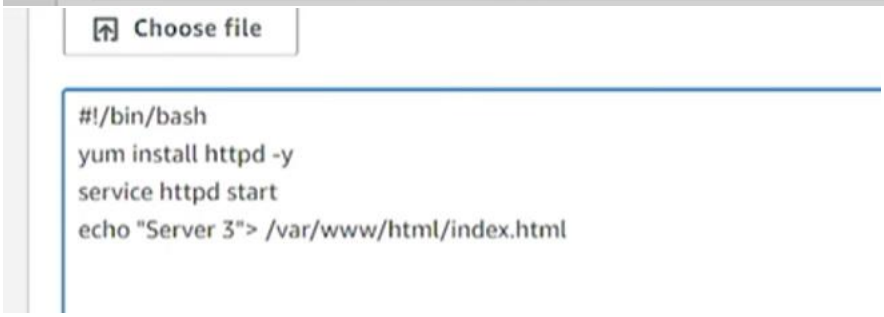


Step 9 :

Create Launch Template follows Step 6 as a continue. Name : Server 3

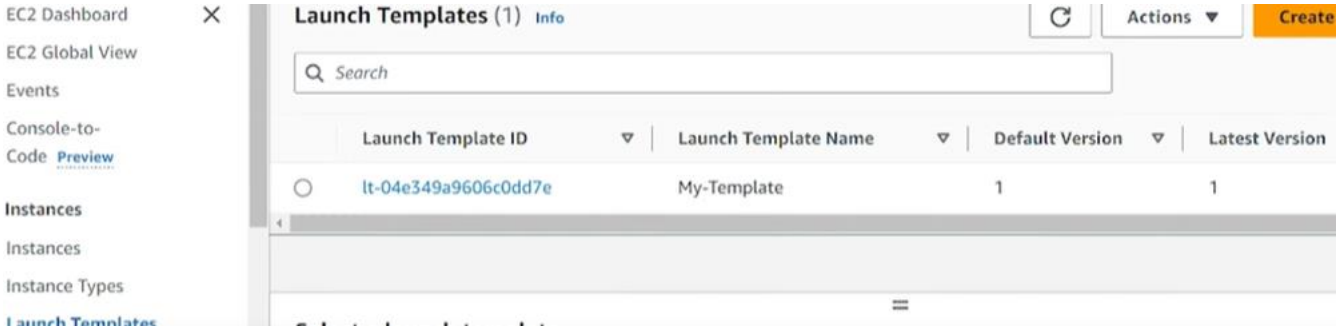


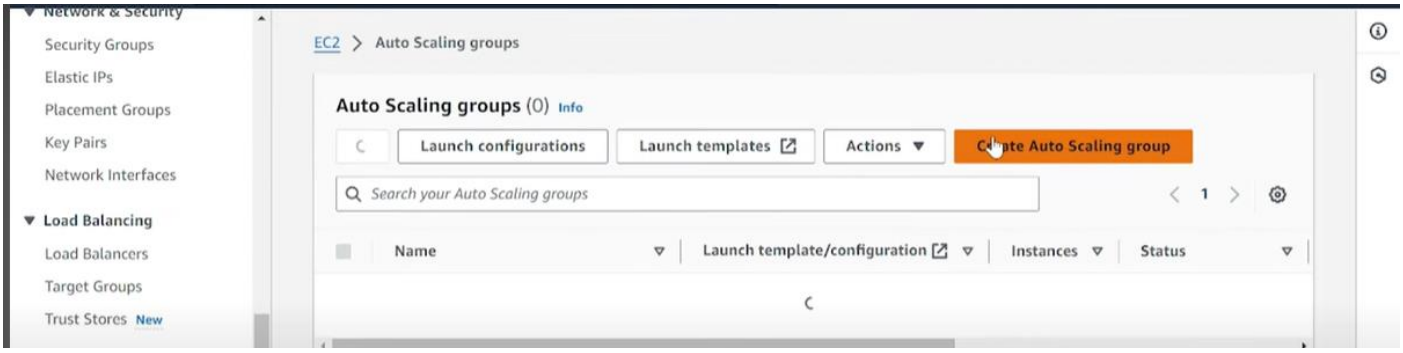


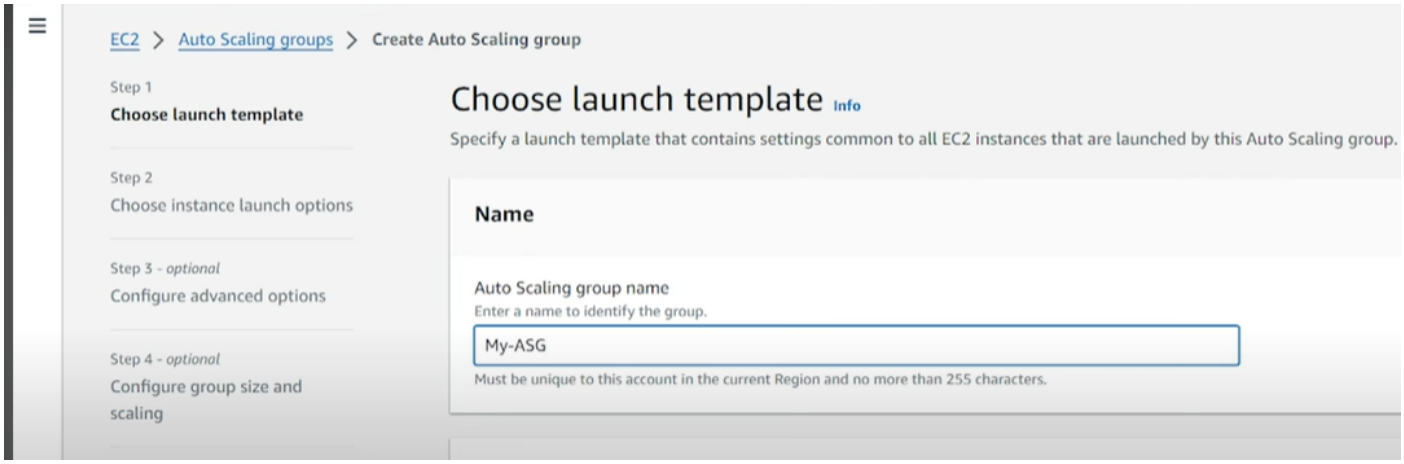


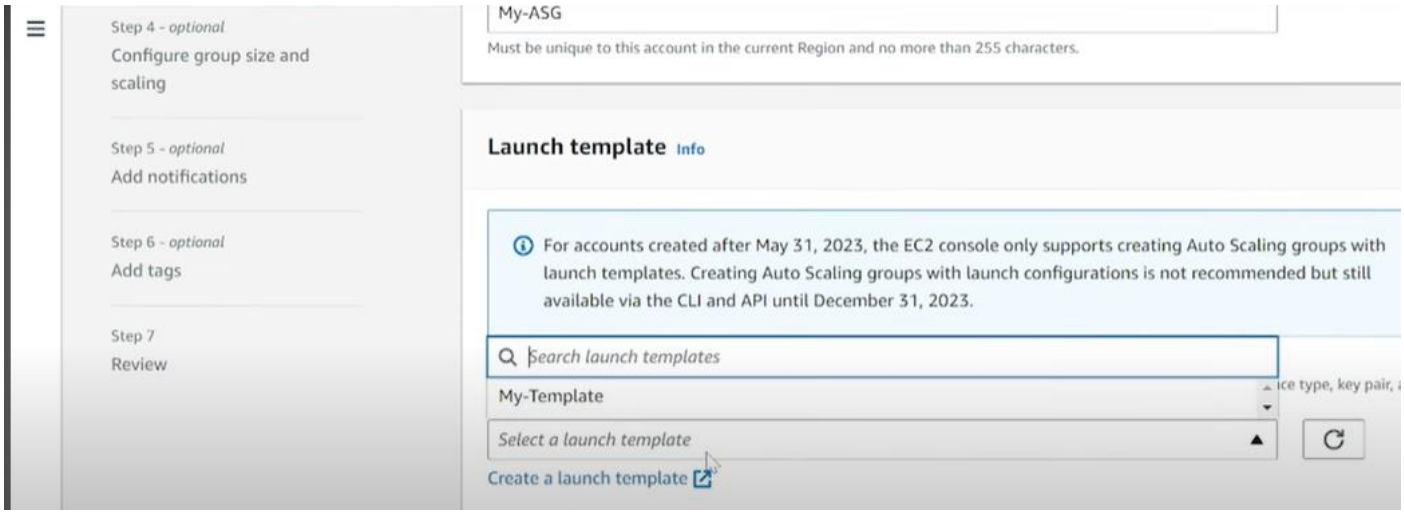
Step 9 :

Create AutoScaling Groups.

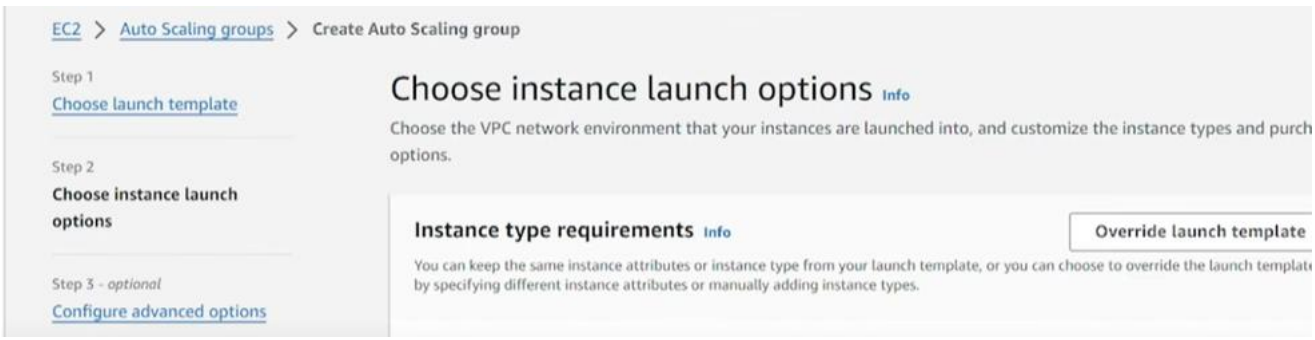


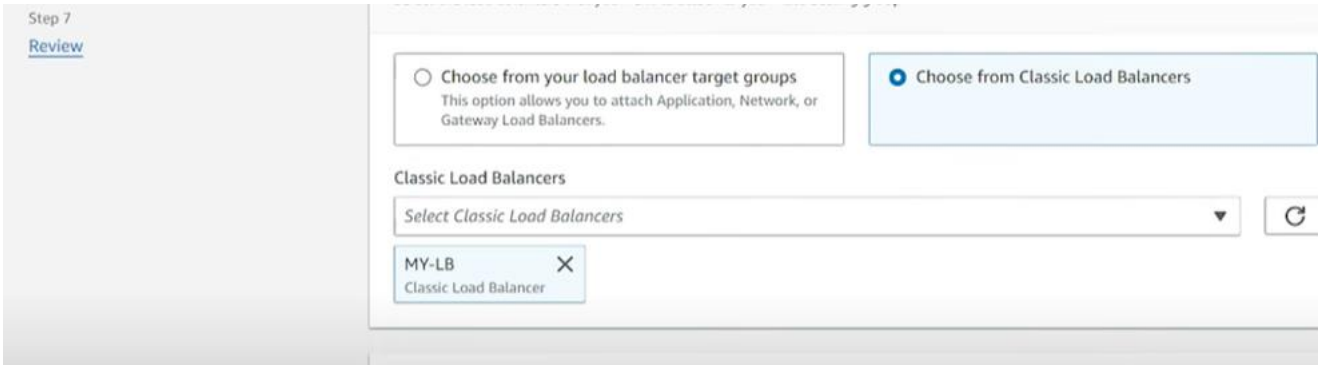

Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups
Trust Stores New
EC2 > Auto Scaling groups
Auto Scaling groups (0) Info
Launch configurations
Launch templates
Actions
Create Auto Scaling group
Search your Auto Scaling groups
1
Name
Launch template/configuration
Instances
Status

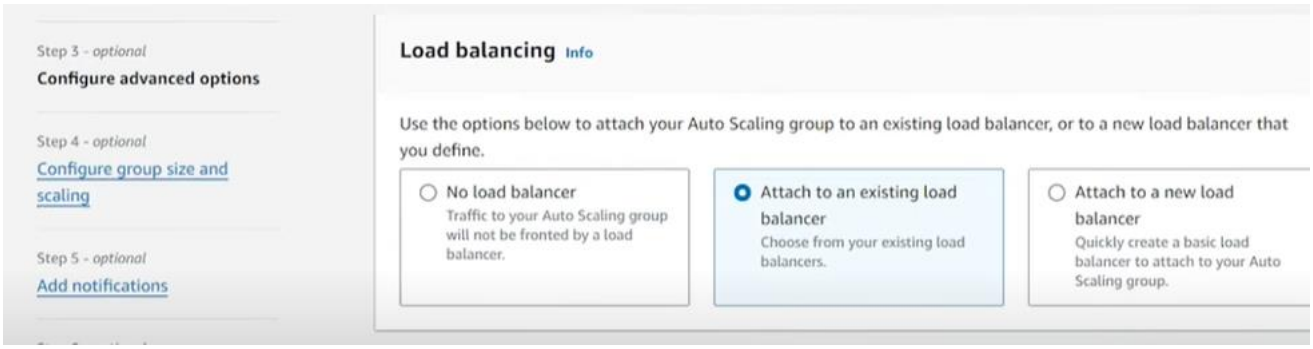

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Configure advanced options
Step 4 - optional
Configure group size and scaling
Choose launch template Info
Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.
Name
Auto Scaling group name
Enter a name to identify the group.
My-ASG
Must be unique to this account in the current Region and no more than 255 characters.

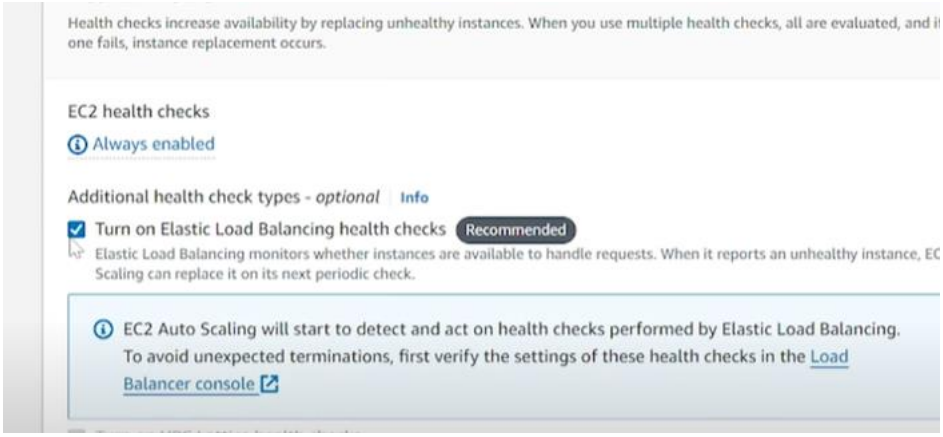

Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Step 6 - optional
Add tags
Step 7
Review
My-ASG
Must be unique to this account in the current Region and no more than 255 characters.
Launch template Info
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.
Search launch templates
My-Template
Select a launch template
Create a launch template

Click next..

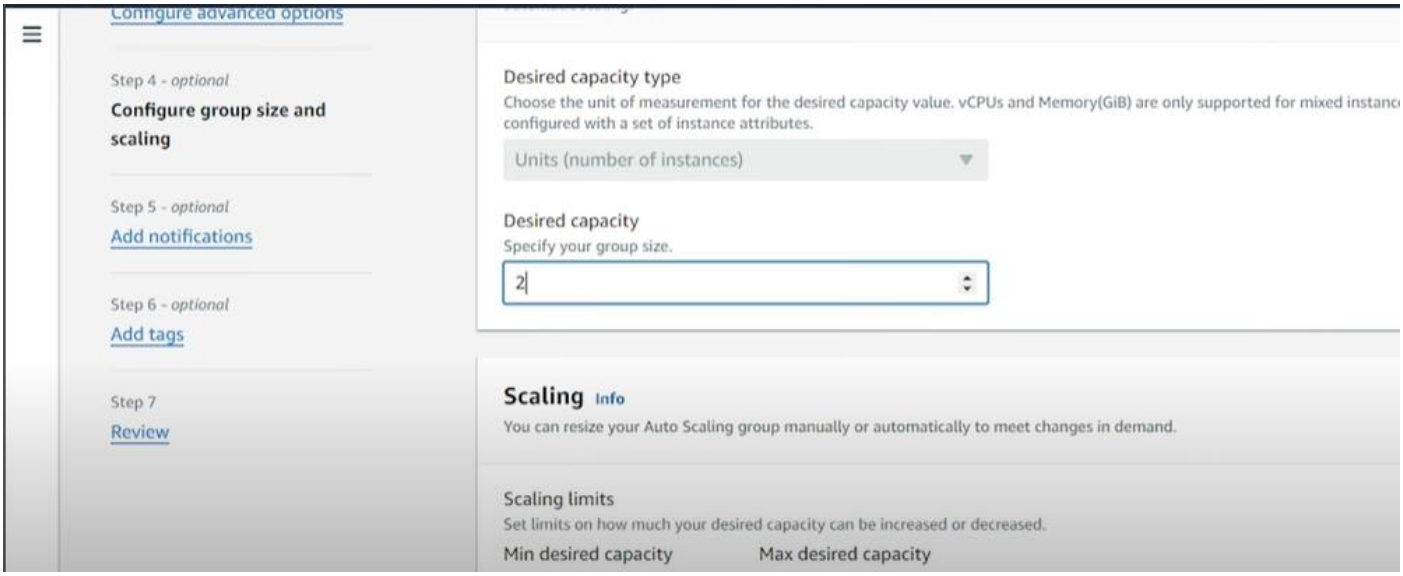

EC2 > Auto Scaling groups > Create Auto Scaling group
Step 1
Choose launch template
Step 2
Choose instance launch options
Step 3 - optional
Configure advanced options
Choose instance launch options Info
Choose the VPC network environment that your instances are launched into, and customize the instance types and purch
options.
Instance type requirements Info
Override launch template
You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch templat
by specifying different instance attributes or manually adding instance types.


Step 7
Review
Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.
Choose from Classic Load Balancers
Classic Load Balancers
Select Classic Load Balancers
MY-LB
Classic Load Balancer

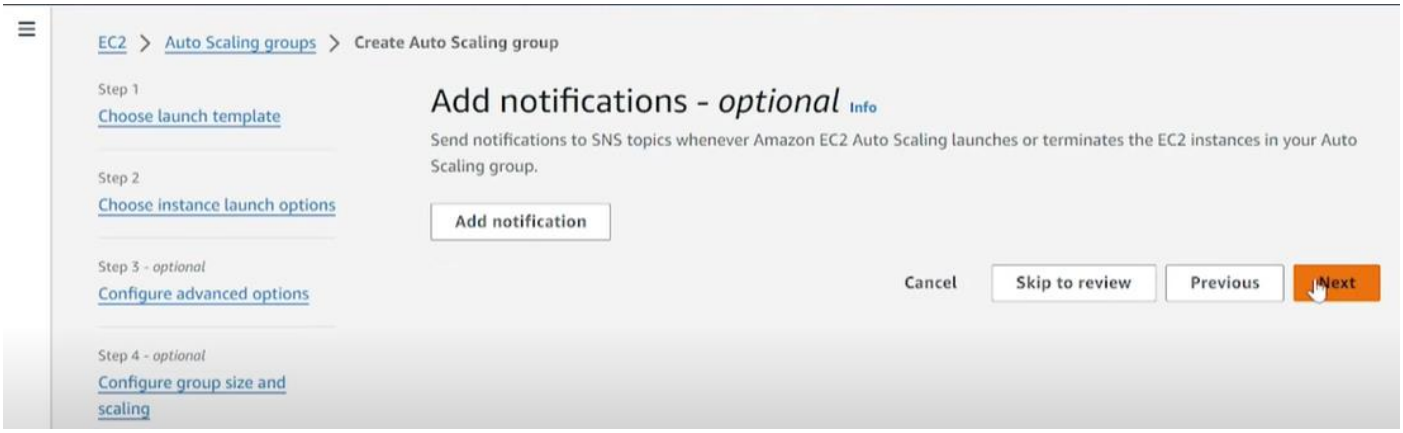

Step 3 - optional
Configure advanced options
Step 4 - optional
Configure group size and scaling
Step 5 - optional
Add notifications
Load balancing Info
Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.
No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.
Attach to an existing load balancer
Choose from your existing load balancers.
Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.


Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and i
one fails, instance replacement occurs.
EC2 health checks
Always enabled
Additional health check types - optional Info
Turn on Elastic Load Balancing health checks Recommended
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC
Scaling can replace it on its next periodic check.
EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing.
To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console

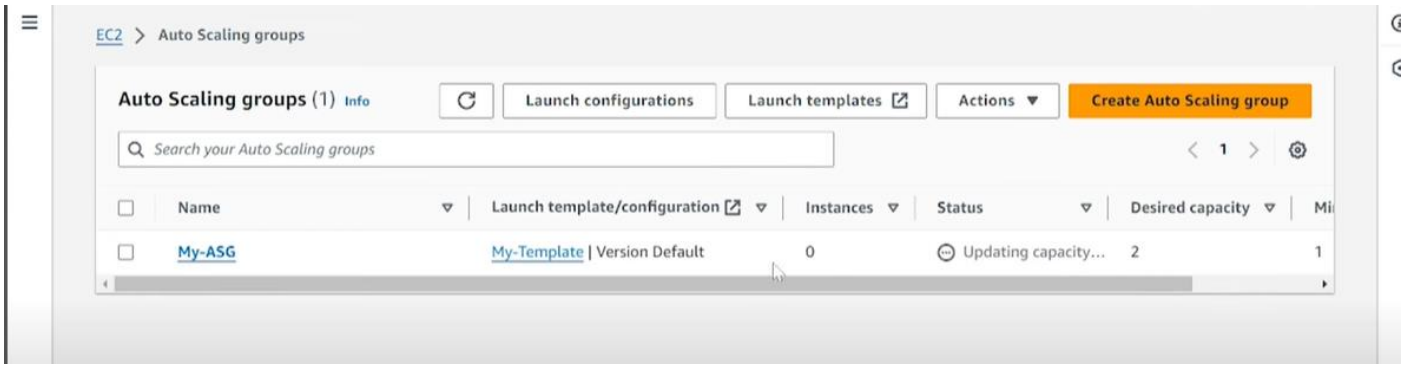
Click Next...



After this step all are default and chlick next..

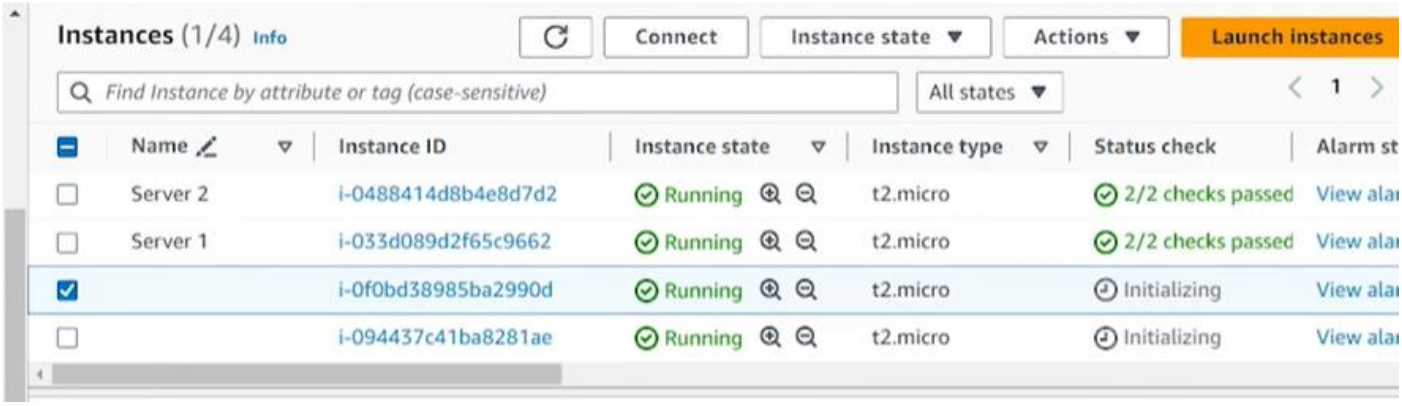


click Next and click Creat auto scaling Groups.



Step 10:

Go to EC2 instance there will running another 2 servers with the name of SERVER 3 which we given in the Launch Template.



Run the public IP address in the web browser, and it will display Server 3. Now, go to the Load Balancer, copy the DNS URL, and paste it into the web browser.

In this PoC, how do we route the servers using a load balancer .

If the instances are reloaded repeatedly, the servers may get terminated.

Outcome:

Accessing EC2 Public IP

- When you enter the **public IP address** of any EC2 instance (e.g., Server 1, Server 2, or Server 3) in a web browser, it should display the corresponding server name.

Testing Load Balancer DNS

- When you enter the **Load Balancer DNS URL** in a web browser, it should route requests to different servers **based on round-robin or health check rules**.

*The output should change depending on which server the Load Balancer directs traffic to.

Verifying Traffic Distribution

- When checking the **Load Balancer Target Group**, the status of instances should show **Healthy**.
- If any instance is unhealthy, it should be replaced based on the **Auto Scaling Group** policy.