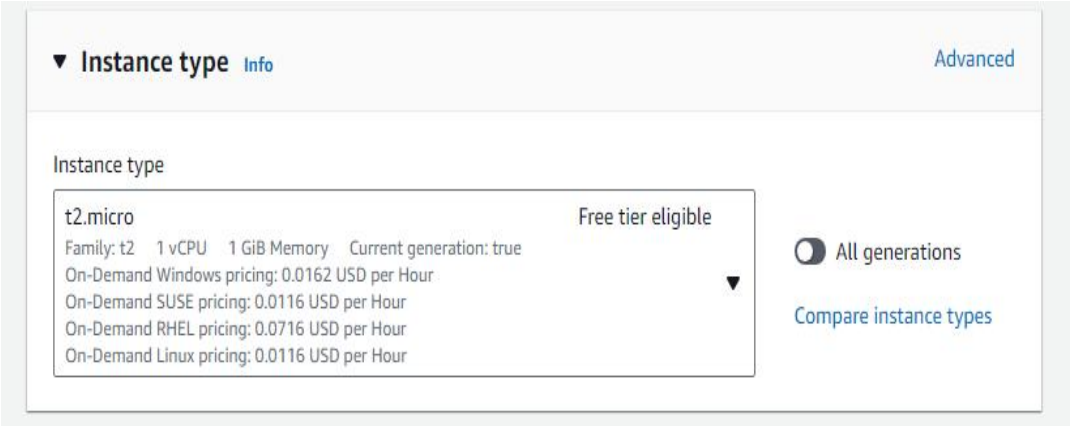
## Implementing High Availability with Auto Scaling and Load Balancer:

Description: Configure an application to automatically scale based on demand to ensure high availability and performance. Implementation: Set up an Auto Scaling group and configure scaling policies based on metrics like CPU utilization or request rate. Use Elastic Load Balancer to distribute traffic across instances.

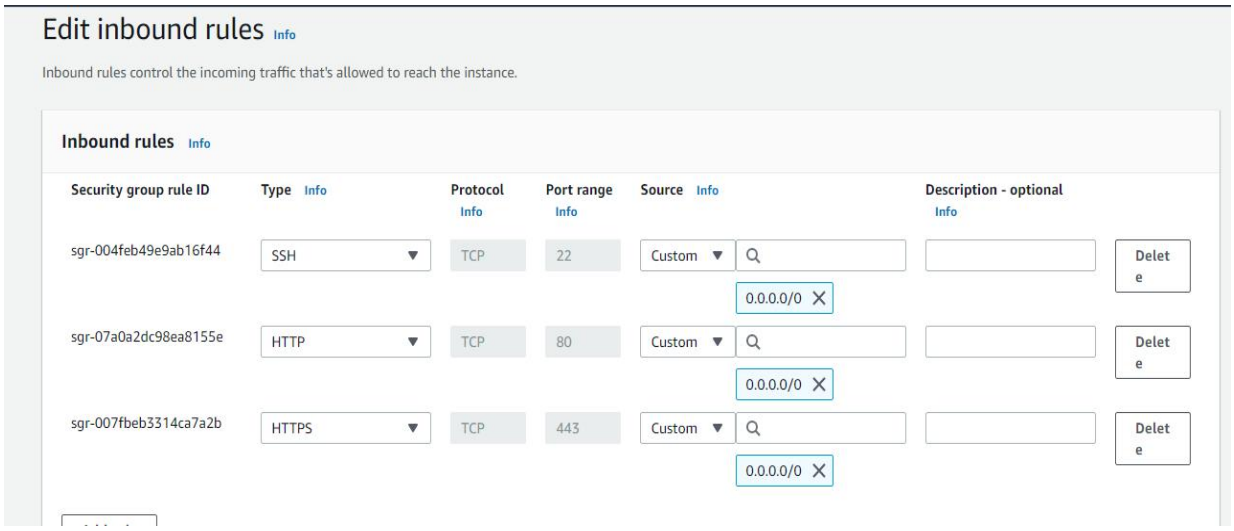
Setting up High Availability with Auto Scaling involves several steps. Below is a detailed explanation of each step:

## 1. Launch and Ec2 instance.

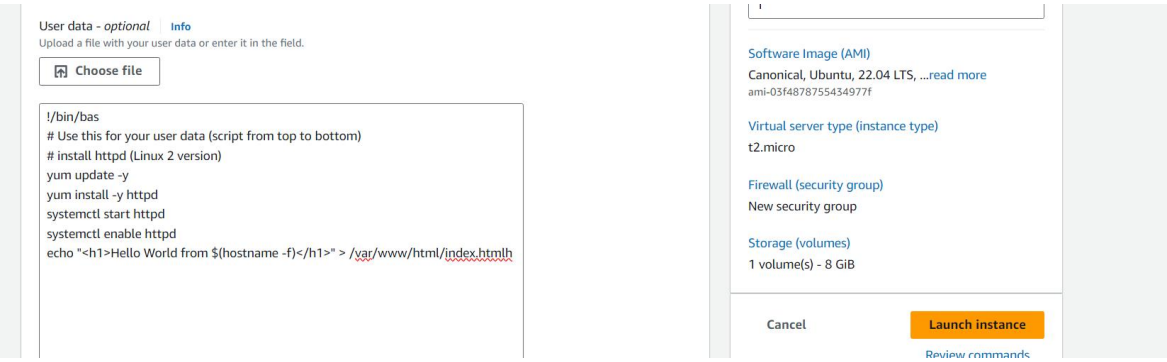
1. Navigate to the EC2 dashboard.
2. Click "Launch Instance."
3. Choose the Ubuntu Server AMI for both instances.
4. Select an appropriate instance type, like t2.micro.




5. Configure instance details like the number of instances, network settings, etc.
6. Add storage if needed, and configure tags (optional)
7. We'll be using the Amazon Linux 2 AMI and launching instances with a t2.micro instance type.
8. If you already have a key pair, you can proceed with the existing key pair. If not, create a new key pair.
9. For the Security Group, we'll create a new one with rules to allow HTTP/HTTPS and SSH access. This will ensure that our web application can be accessed securely, and we have the necessary SSH access to manage our instances.



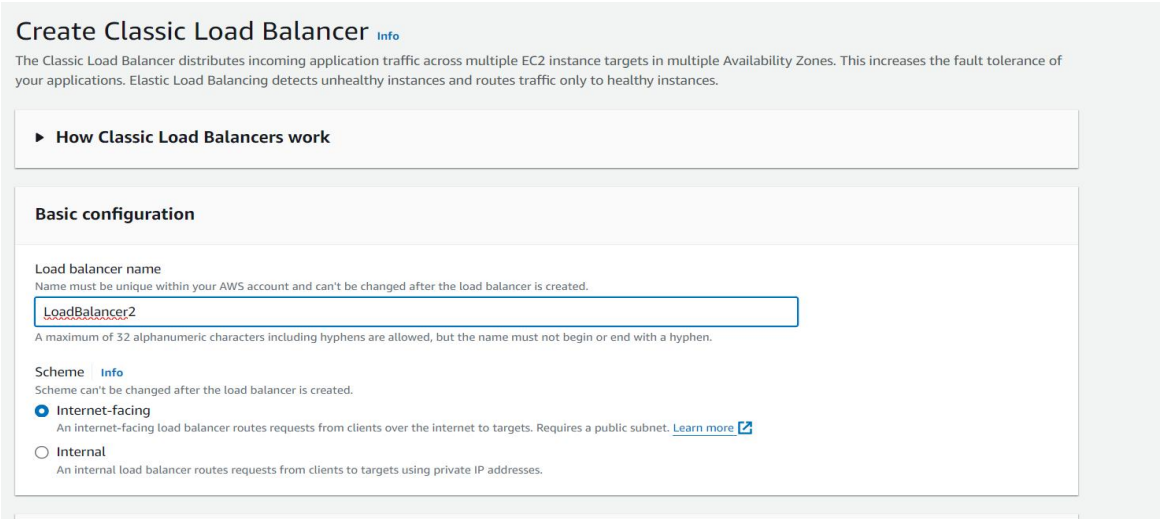

10. Under the **Advanced Details** section paste this script into the **user data**



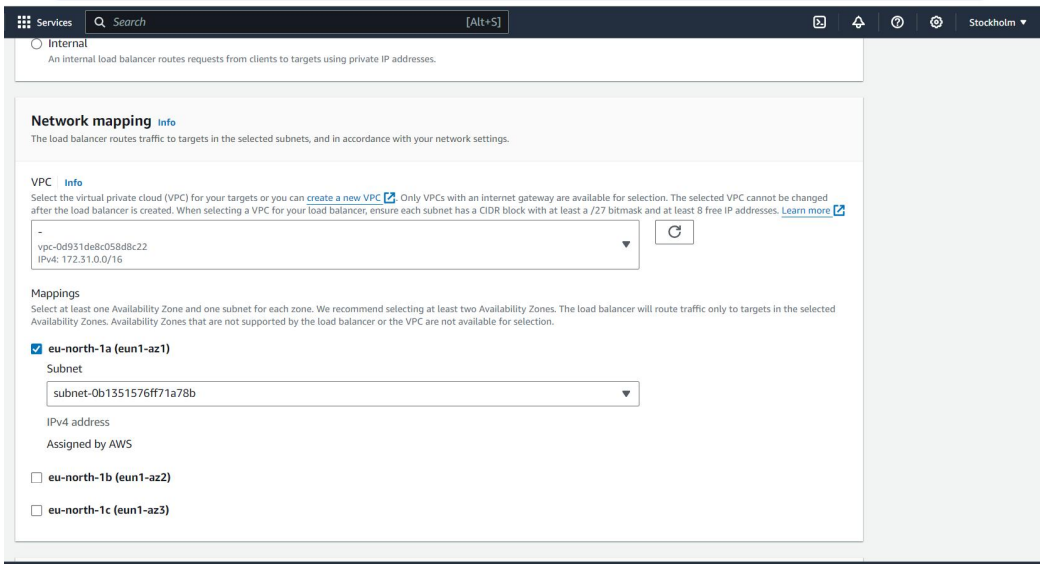

11. This user data script will be executed when each instance is launched, ensuring that the necessary packages are updated, the Apache HTTP server is installed, and a basic HTML file is created to serve as the content for our web server.

## 2. Create an elastic Load Balancer:

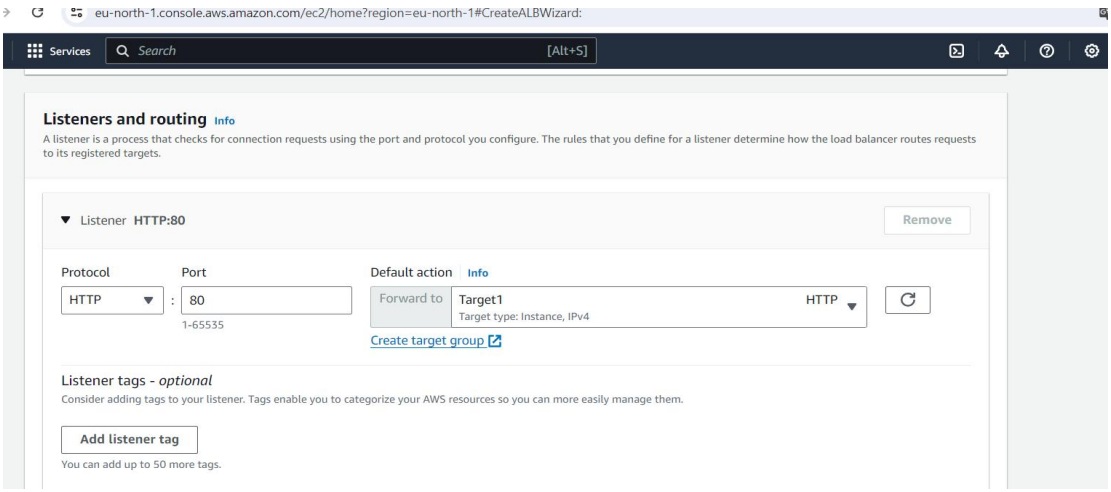
1. In the AWS Management Console, go to the EC2 service.
2. Under "Load Balancing," select "Load Balancers."
3. Click "Create Load Balancer."



4. Choose the appropriate load balancer type.
5. Configure the listener settings, such as port and protocol.
6. Assign the Auto Scaling group instances to the target group.
7. For the 'Target groups,' select the target group that you created earlier. This will specify which instances should receive traffic from the ALB.
8. Ensure that the 'Scheme' is set to 'Internet-facing.' This allows the ALB to receive traffic from the internet.



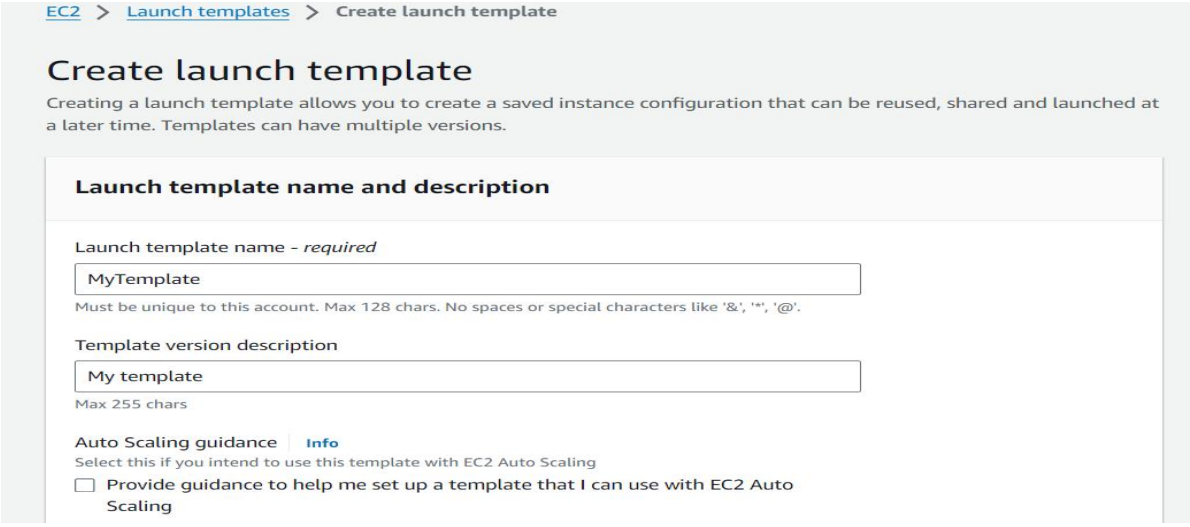

9. Under 'Network mapping,' select the appropriate Virtual Private Cloud (VPC) for your application. Choose the VPC where your instances are located.



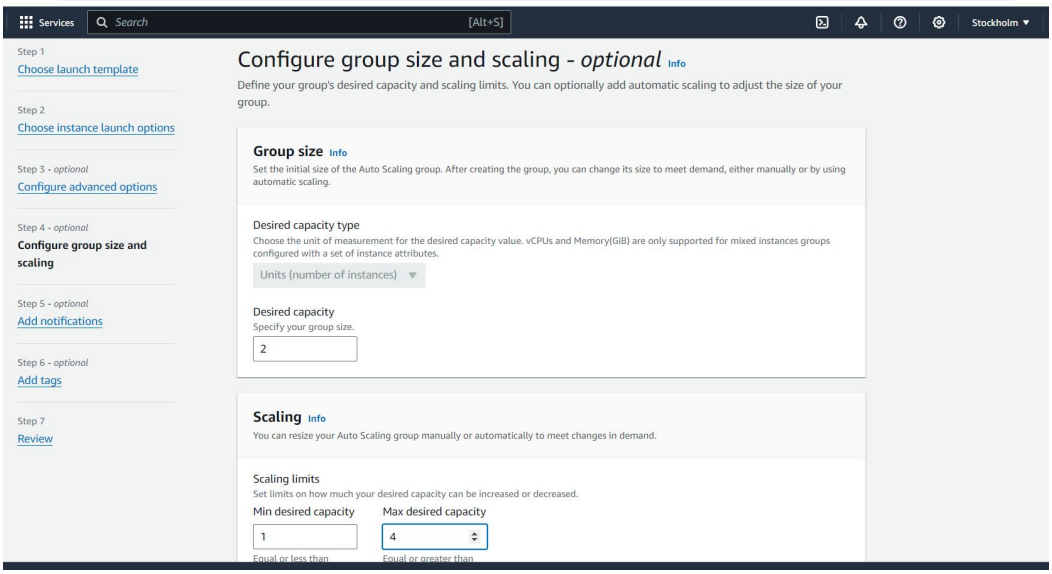

10. Select three subnets within the VPC for the ALB to distribute traffic across Availability Zones. This helps improve availability and fault tolerance.
11. For the 'Security groups,' choose the previously created security group that allows HTTP/HTTPS and SSH access. This ensures that the ALB can communicate with the instances securely.

## 3. Create an Auto Saling Group:

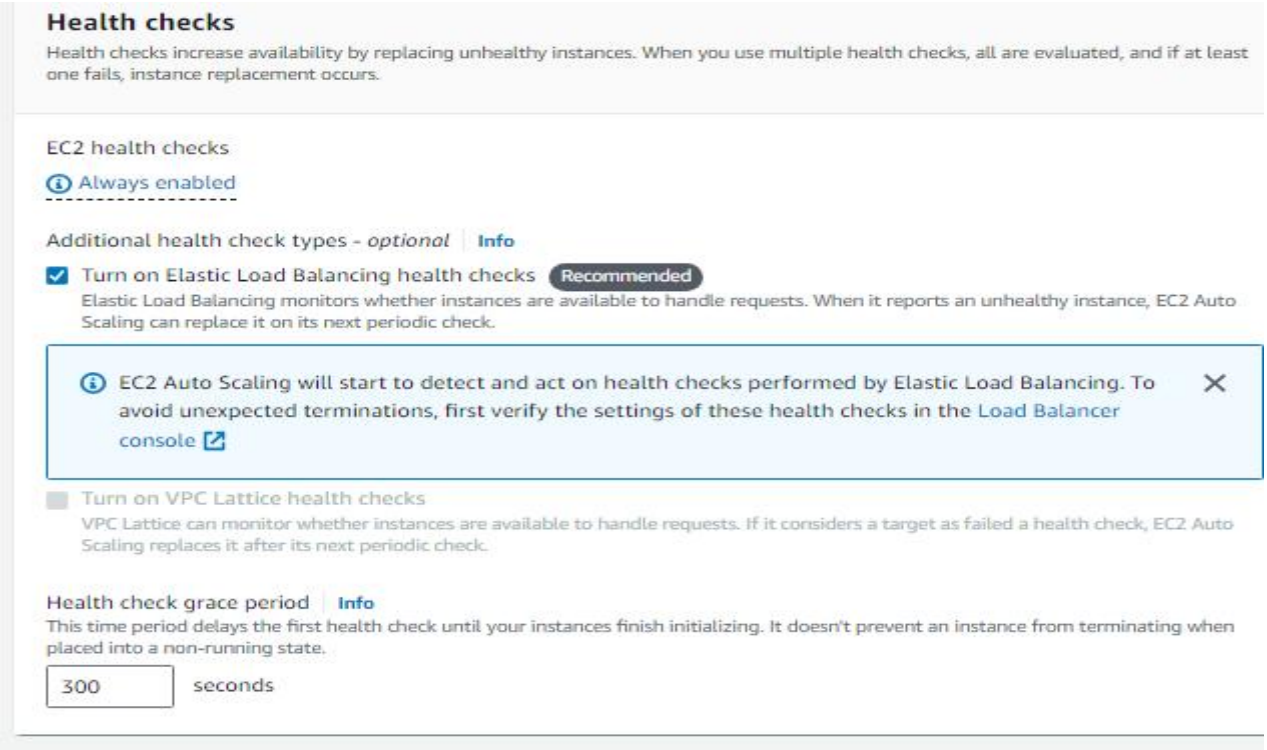
1. Navigate to the AWS Management Console.
2. Open the EC2 service.
3. In the left navigation pane, click on "Auto Scaling Groups."




4. Click the "Create Auto Scaling group" button.
5. Select Name and Instance Template: Choose a name for your Auto Scaling Group and select the launch template we created earlier.
6. This template includes all the configurations needed for the instances.
7. Select the desired launch configuration or create a new one.



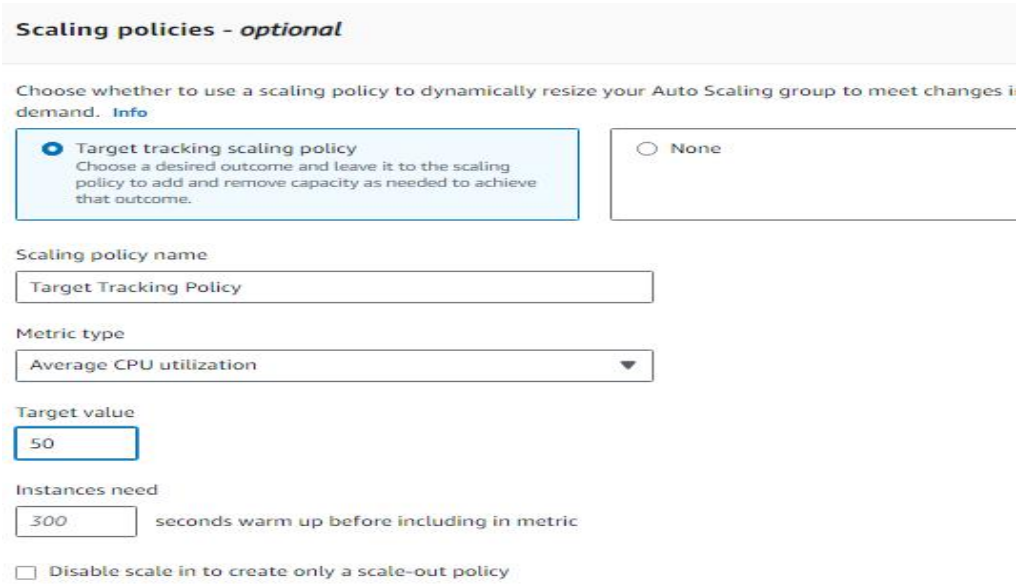

8. Set up health checks to ensure instances are healthy.




9. Choose Instance Scaling Limits: Specify the minimum, desired, and maximum number of instances you want to maintain within the Auto Scaling Group. This allows the ASG to automatically adjust the number of instances based on demand.

## 4. Configure Scaling Policies:

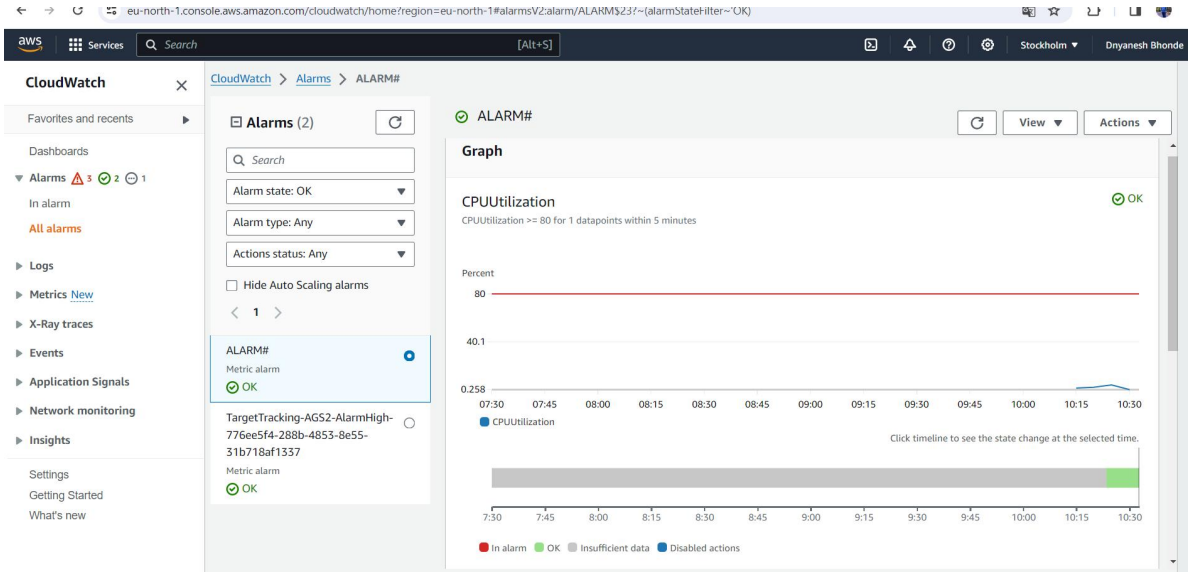
1. In the Auto Scaling group, under the "Automatic scaling" tab, click on "Add policy."
2. Choose a scaling policy type (e.g., Target Tracking Scaling or Step Scaling).
3. Define the scaling policy with appropriate metrics (e.g., CPU utilization, request rate).




4. Set target values or thresholds for the chosen metric.
5. Configure cooldown periods to prevent rapid scaling.

## 5. Test and monitor

1. Deploy your application to the instances launched by the Auto Scaling group.
2. Monitor the Auto Scaling group and Elastic Load Balancer for proper functioning.
3. Simulate load and verify that instances scale in and out based on the defined scaling policies.
4. Ensure traffic is evenly distributed across instances by checking the load balancer.




5. You can observe and analyze the performance changes by checking the following AWS monitoring services: CloudWatch, Auto Scaling Group Activity History, and EC2 Instance Monitoring.
6. Finally, cross-check the Application Load Balancer (ALB) DNS to verify whether it successfully distributed the traffic across the instances.


**Hello World from ip-172-31-41-27.eu-north-1.compute.internal**

7. Here's a **sample email notification** issued by **CloudWatch** via SNS, however all CloudWatch email alerts look alike.

> You are receiving this email because your Amazon CloudWatch Alarm "ALARM4" in the EU (Stockholm) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [19.21956536319807 (25/01/24 13:32:00)] was less than or equal to the threshold (30.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Thursday 25 January, 2024 13:37:35 UTC".
>
> View this alarm in the AWS Management Console:
> https://eu-north-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=eu-north-1#alarmsV2:alarm/ALARM4
>
> Alarm Details:
> - Name: ALARM4
> - Description: ALARM4
> - State Change: OK -> ALARM
> - Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [19.21956536319807 (25/01/24 13:32:00)] was less than or equal to the threshold (30.0) (minimum 1 datapoint for OK -> ALARM transition).
> - Timestamp: Thursday 25 January, 2024 13:37:35 UTC

8. Looking at this email, you can see the **little data available**: name of the alarm triggered, timestamp, region, and state.

**Reference:**


https://docs.aws.amazon.com/whitepapers/latest/real-time-communication-on-aws/high-availability-and-scalability-on-aws.html

https://medium.com/@bstuurmanct/achieving-high-availability-with-an-auto-scaling-group-and-an-application-load-balancer-on-aws-8caae425f6c7

https://docs.aws.amazon.com/autoscaling/ec2/userguide/auto-scaling-benefits.html

https://docs.aws.amazon.com/autoscaling/ec2/userguide/tutorial-ec2-auto-scaling-load-balancer.html