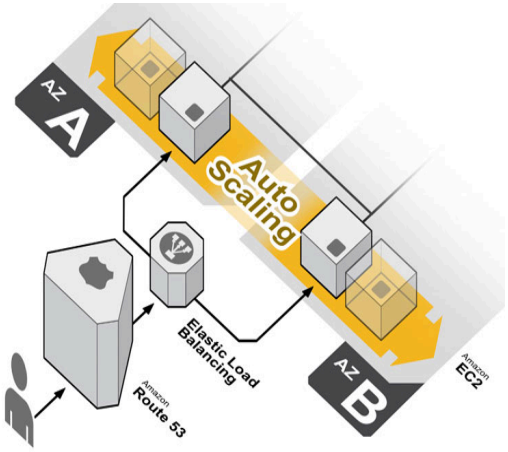
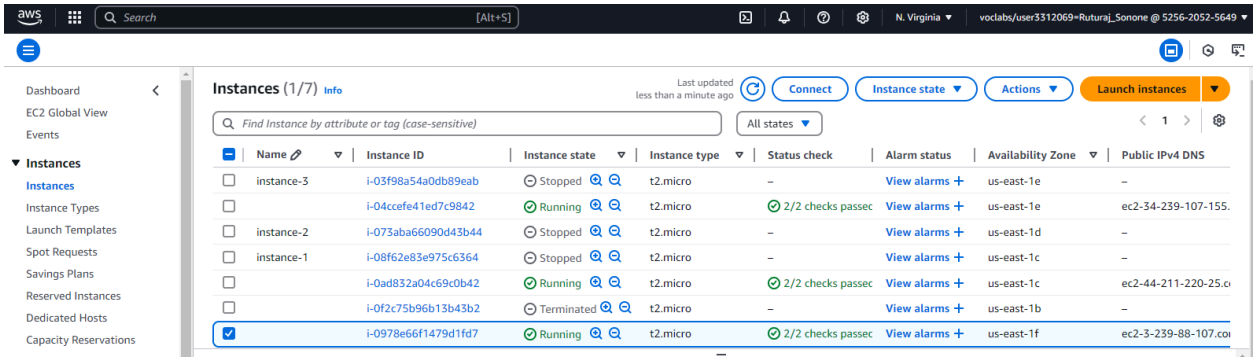


amazon web services
AUTO SCALING

Implementing Setting Up Auto Scaling in AWS

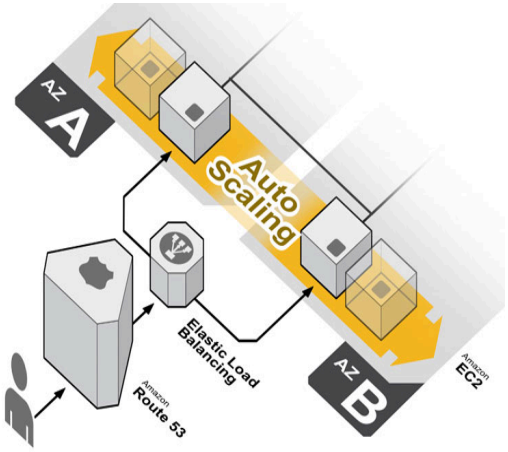
Step 1: Launch an EC2 Instance

1. **Log in to AWS Console**: Navigate to the AWS Management Console.
2. **Go to EC2**: From the services menu, select **EC2**.
3. **Launch an EC2 Instance**:
    - Choose an Amazon Machine Image (AMI) and instance type.
    - Configure network settings and security groups.
    - Install necessary applications or scripts for the instance.
4. **Create a Key Pair**: Download it for SSH access to the instance.
5. Once launched, ensure the instance is accessible and functioning.



Step 2: Create a Launch Template or Launch Configuration

1. **Go to Launch Templates**:
    - From the EC2 dashboard, select **Launch Templates**.
    - Click **Create Launch Template**.

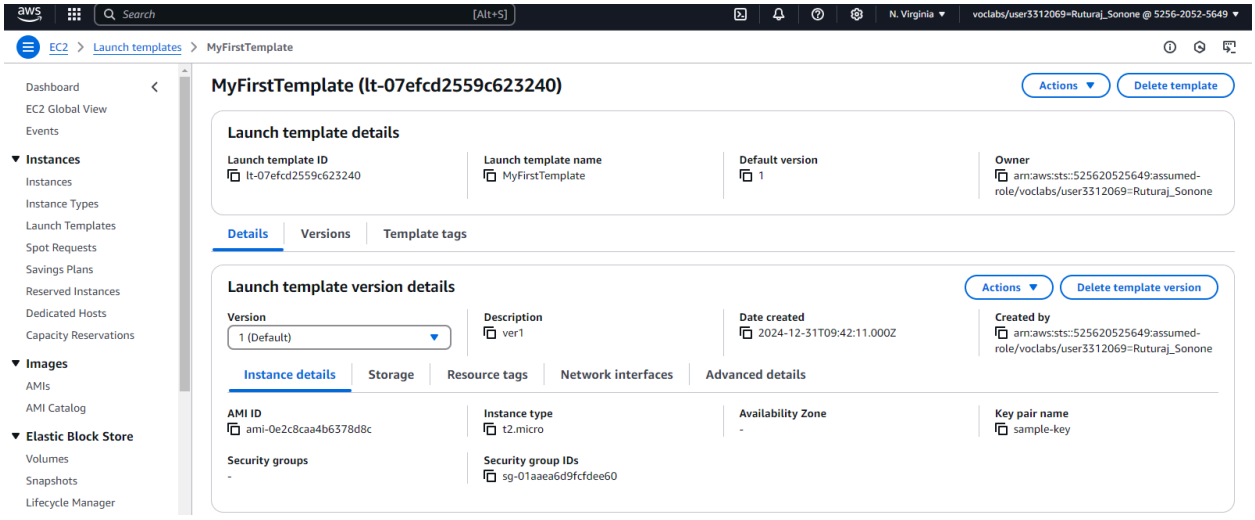


amazon web services
AUTO SCALING

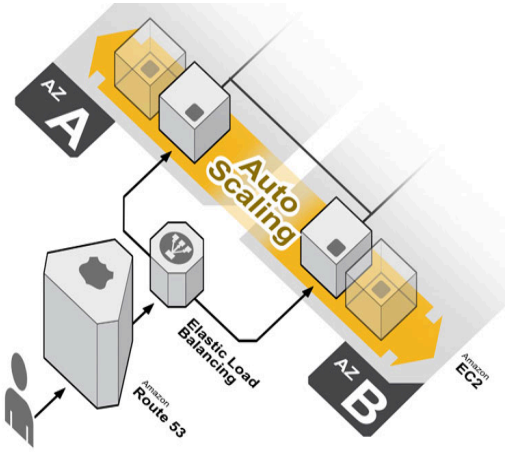
2. **Fill Template Details**:
    - Provide a name and description.
    - Specify the AMI, instance type, and key pair.
    - Configure storage, network, and security settings.
    - Add any startup scripts in the **User Data** section.
3. **Save the Launch Template**.

Alternatively, you can create a **Launch Configuration**:

- Navigate to the Auto Scaling section and select **Launch Configurations**.
- Follow similar steps to define instance settings.

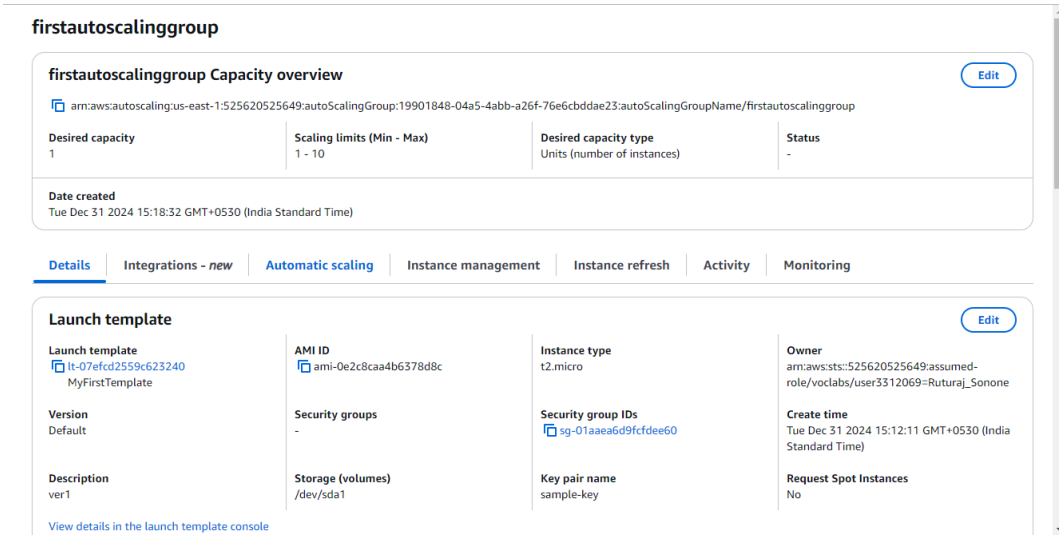


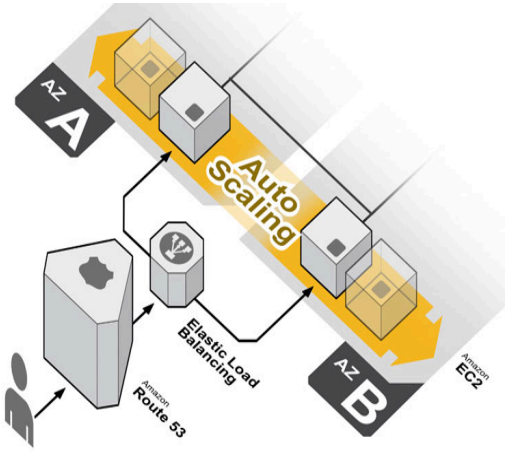
Step 3: Create an Auto Scaling Group



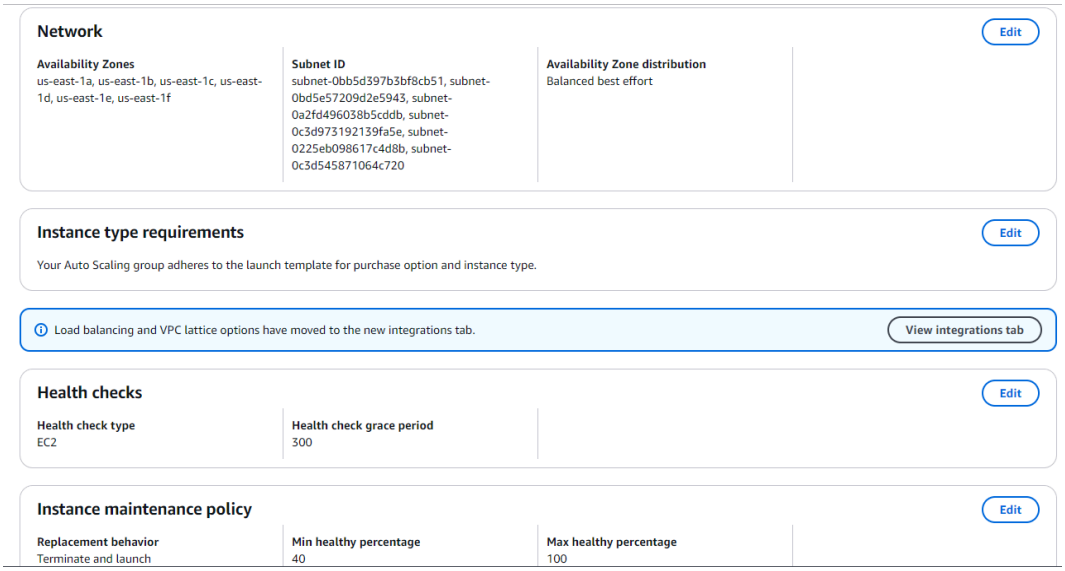
amazon web services
AUTO SCALING

1. **Navigate to Auto Scaling Groups**:
    - From the EC2 dashboard, select **Auto Scaling Groups**.
2. **Create a New Auto Scaling Group**:
    - Choose the **Launch Template** or **Launch Configuration** created earlier.
3. **Configure Group Details**:
    - Specify the VPC and subnets where instances will be launched.
    - Set the minimum, maximum, and desired number of instances.
4. **Attach Load Balancer (Optional)**:
    - Attach an existing Application Load Balancer or Classic Load Balancer to distribute traffic.
5. **Set Scaling Policies**:
    - Enable scaling based on metrics like **CPU Utilization** or custom CloudWatch alarms.
    - Example: Add instances if CPU > 70% and remove instances if CPU < 20%.
6. **Review and Create**:
    - Confirm the settings and create the Auto Scaling Group.



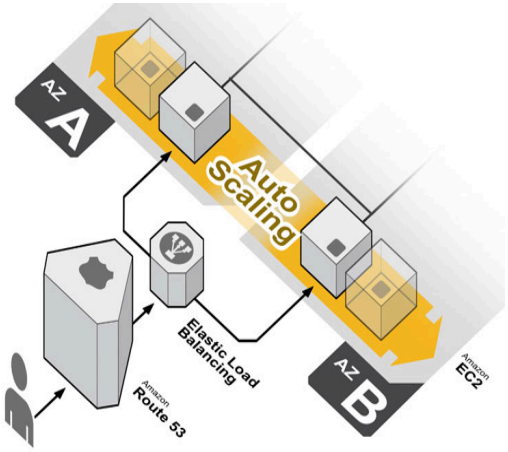


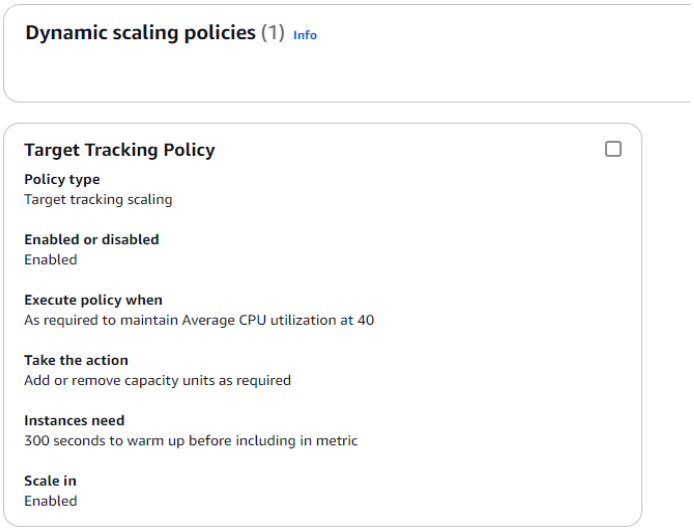
amazon web services
AUTO SCALING



Step 4: Configure Scaling Policies

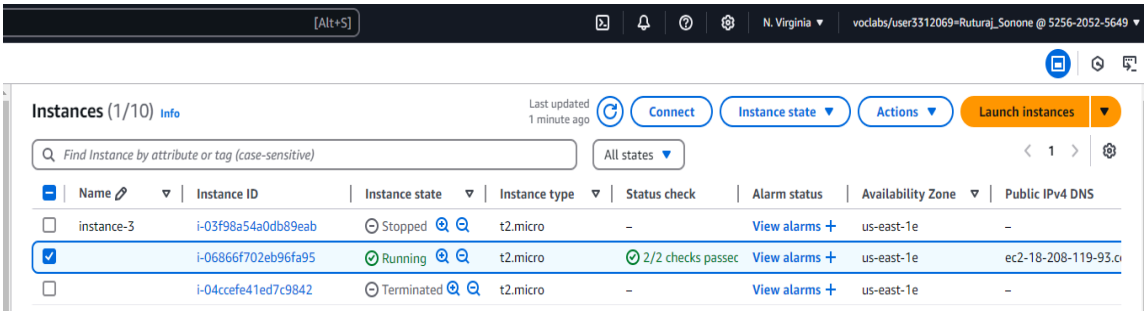
1. **Dynamic Scaling**:
    - Navigate to the Auto Scaling Group settings.
    - Add policies to scale in or out based on CloudWatch alarms.
    - Example:
        - **Scale Out**: Add instances when CPU exceeds 70%.
        - **Scale In**: Remove instances when CPU drops below 20%.
2. I have chosen **dynamic scaling** , but you can choose **Predictive scaling policies** or **Scheduled actions** as per your choice.

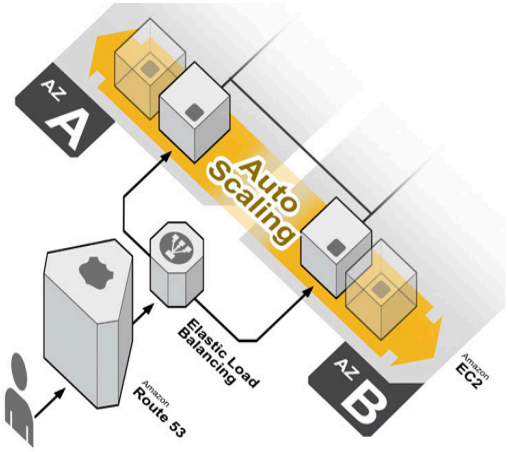




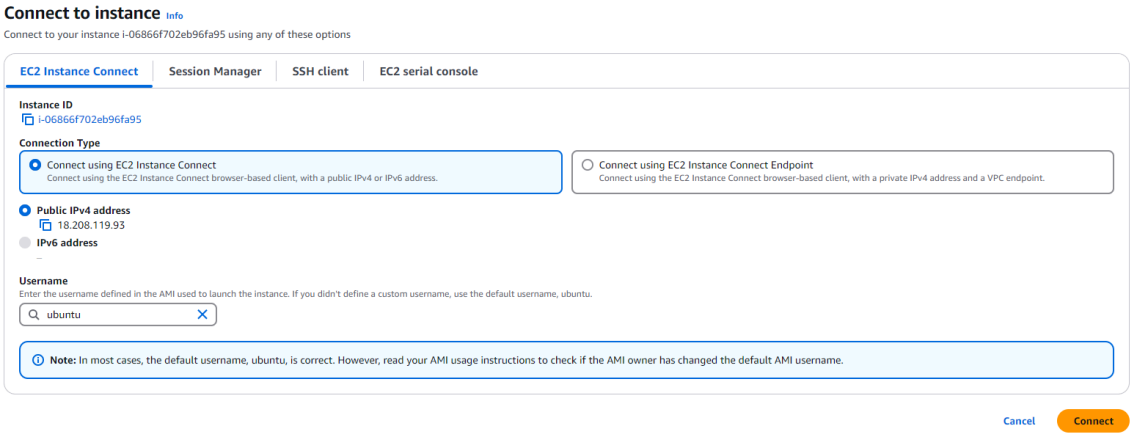
Step 5: Test the Auto Scaling

1. **Simulate Load**:
    - Use tools like **stress** to increase CPU usage or generate traffic.
      - To increase CPU stress for testing purposes on an AWS EC2 instance via SSH, you can use the **stress** tool. Here's how you can do it:





amazon web services
AUTO SCALING

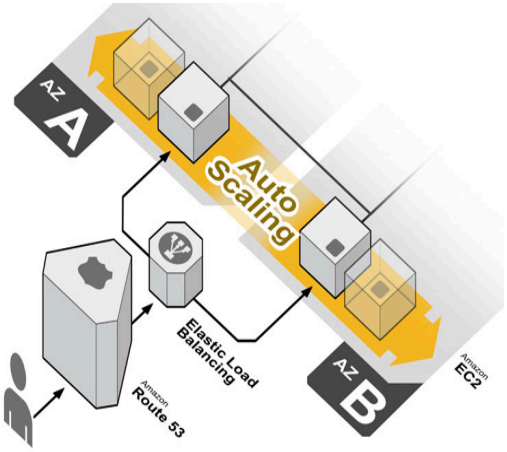


- **Steps to Stress Test the CPU:**
- **Install Stress Tool** (if not already installed):

```
ubuntu@ip-172-31-60-180:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Get:5 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [7196 B]
Get:6 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]
Get:7 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]
Get:8 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]
Fetched 186 kB in 1s (285 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-60-180:~$
```

2. Run Stress Command:

Example to stress the CPU with 5 workers for 60 seconds:



amazon web services
AUTO SCALING

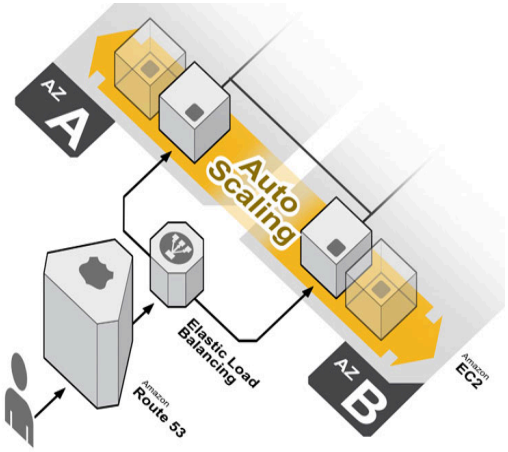
```
ubuntu@ip-172-31-60-180:~$ sudo apt-get install stress -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  stress
0 upgraded, 1 newly installed, 0 to remove and 58 not upgraded.
Need to get 18.1 kB of archives.
After this operation, 52.2 kB of additional disk space will be used.
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 stress amd64 1.0.7-1 [18.1 kB]
Fetched 18.1 kB in 0s (1195 kB/s)
Selecting previously unselected package stress.
(Reading database ... 70649 files and directories currently installed.)
Preparing to unpack .../stress_1.0.7-1_amd64.deb ...
Unpacking stress (1.0.7-1) ...
Setting up stress (1.0.7-1) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-60-180:~$
```

```
ubuntu@ip-172-31-60-180:~$ stress --cpu 5 --timeout 50
stress: info: [2583] dispatching hogs: 5 cpu, 0 io, 0 vm, 0 hdd
```

3. Check CPU Usage (Optional):

Use the top or htop command in another terminal session to monitor the CPU usage:

```
top
```



amazon web services
AUTO SCALING

4. Stop Stress Test:

If you want to stop the stress test before it completes, press `Ctrl+C` in the terminal running the stress command.

5. **Monitor Scaling Activity**:

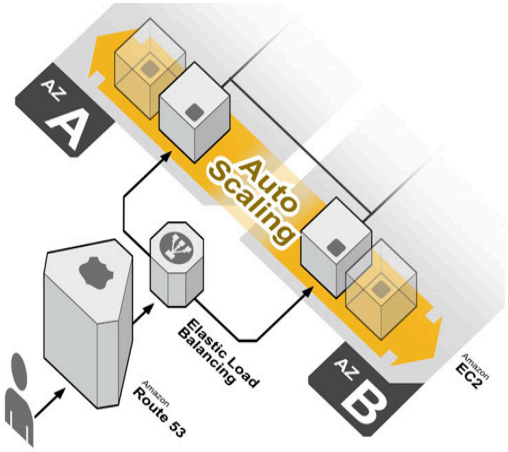
- Go to the **Activity** tab in the Auto Scaling Group to track instance launches and terminations.

Welcome RUTURAJ this is ip-172-31-52-214

=

Auto Scaling group: firstautoscalinggroup

Successful	Launching a new EC2 instance: i-087c1c92db238bdee	At 2024-12-31T12:02:55Z an instance was launched in response to an unhealthy instance needing to be replaced.	December 31, 05:32:57 PM +05:30	December 31, 05:33:59 PM +05:30
Successful	Terminating EC2 instance: i-06866f702eb96fa95	At 2024-12-31T12:02:55Z an instance was taken out of service in response to an EC2 health check indicating it has been terminated or stopped.	2024 December 31, 05:32:55 PM +05:30	2024 December 31, 05:33:17 PM +05:30
Successful	Terminating EC2 instance: i-0568a654eb39a12a0	At 2024-12-31T10:44:38Z a user request update of AutoScalingGroup constraints to min: 1, max: 10, desired: 1 changing the desired capacity from 4 to 1. At 2024-12-31T10:44:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 1. At 2024-12-31T10:44:49Z instance i-00afb7f1b42409be2 was selected for termination. At 2024-12-31T10:44:49Z instance i-0121302bf0e1964db was selected for termination. At 2024-12-31T10:44:49Z instance i-0568a654eb39a12a0 was selected for termination.	2024 December 31, 04:14:49 PM +05:30	2024 December 31, 04:15:51 PM +05:30



amazon web services
AUTO SCALING

Successful	Terminating EC2 instance: i-0121302bf0e1964db	At 2024-12-31T10:44:38Z a user request update of AutoScalingGroup constraints to min: 1, max: 10, desired: 1 changing the desired capacity from 4 to 1. At 2024-12-31T10:44:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 1. At 2024-12-31T10:44:49Z instance i-00afb7f1b42409be2 was selected for termination. At 2024-12-31T10:44:49Z instance i-0121302bf0e1964db was selected for termination. At 2024-12-31T10:44:49Z instance i-0568a654eb39a12a0 was selected for termination.	2024 December 31, 04:14:49 PM +05:30	2024 December 31, 04:16:12 PM +05:30
Successful	Terminating EC2 instance: i-00afb7f1b42409be2	At 2024-12-31T10:44:38Z a user request update of AutoScalingGroup constraints to min: 1, max: 10, desired: 1 changing the desired capacity from 4 to 1. At 2024-12-31T10:44:49Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 4 to 1. At 2024-12-31T10:44:49Z instance i-00afb7f1b42409be2 was selected for termination. At 2024-12-31T10:44:49Z instance i-0121302bf0e1964db was selected for termination. At 2024-12-31T10:44:49Z instance i-0568a654eb39a12a0 was selected for termination.	2024 December 31, 04:14:49 PM +05:30	2024 December 31, 04:16:11 PM +05:30

Auto Scaling group: firstautoscalinggroup

Successful	Terminating EC2 instance: i-04ccefe41ed7c9842	At 2024-12-31T10:42:18Z a monitor alarm TargetTracking-firstautoscalinggroup-AlarmLow-beb7f08f-409a-435c-9246-d9cd7213edc4 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 7 to 6. At 2024-12-31T10:42:29Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 7 to 6. At 2024-12-31T10:42:29Z instance i-04ccefe41ed7c9842 was selected for termination.	2024 December 31, 04:12:29 PM +05:30	2024 December 31, 04:15:11 PM +05:30
Successful	Launching a new EC2 instance: i-06866f702eb96fa95	At 2024-12-31T10:25:45Z a monitor alarm TargetTracking-firstautoscalinggroup-AlarmHigh-a04c62fb-1111-4026-8468-9439308b2e3f in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 7. At 2024-12-31T10:25:53Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 3 to 7.	2024 December 31, 03:55:55 PM +05:30	2024 December 31, 04:01:26 PM +05:30
Successful	Launching a new EC2 instance: i-0568a654eb39a12a0	At 2024-12-31T10:25:45Z a monitor alarm TargetTracking-firstautoscalinggroup-AlarmHigh-a04c62fb-1111-4026-8468-9439308b2e3f in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 3 to 7. At 2024-12-31T10:25:53Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 3 to 7.	2024 December 31, 03:55:55 PM +05:30	2024 December 31, 04:01:11 PM +05:30

Successful	Terminating EC2 instance: i-0978e66f1479d1fd7	At 2024-12-31T10:44:18Z a monitor alarm TargetTracking-firstautoscalinggroup-AlarmLow-beb7f08f-409a-435c-9246-d9cd7213edc4 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 5 to 4. At 2024-12-31T10:44:28Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 5 to 4. At 2024-12-31T10:44:28Z instance i-0978e66f1479d1fd7 was selected for termination.	2024 December 31, 04:14:28 PM +05:30	2024 December 31, 04:15:30 PM +05:30
Successful	Terminating EC2 instance: i-0ad832a04c69c0b42	At 2024-12-31T10:43:18Z a monitor alarm TargetTracking-firstautoscalinggroup-AlarmLow-beb7f08f-409a-435c-9246-d9cd7213edc4 in state ALARM triggered policy Target Tracking Policy changing the desired capacity from 6 to 5. At 2024-12-31T10:43:28Z an instance was taken out of service in response to a difference between desired and actual capacity, shrinking the capacity from 6 to 5. At 2024-12-31T10:43:28Z instance i-0ad832a04c69c0b42 was selected for termination.	2024 December 31, 04:13:28 PM +05:30	2024 December 31, 04:14:51 PM +05:30