Experiment – 4

❖ **Aim:** Learn the basic Elastic Load Balancing (ELB) services to load balance the infrastructure. To create an Amazon Machine Image (AMI) from a running instance. Create a load balancer. Automatically scale new instances. Create Amazon CloudWatch alarms and monitor the performance of your infrastructure.

➢ Task 1: Create an AMI for Auto Scaling

In this task, you will create an AMI from the existing Web Server 1. This will save the contents of the boot disk so that new instances can be launched with identical content.

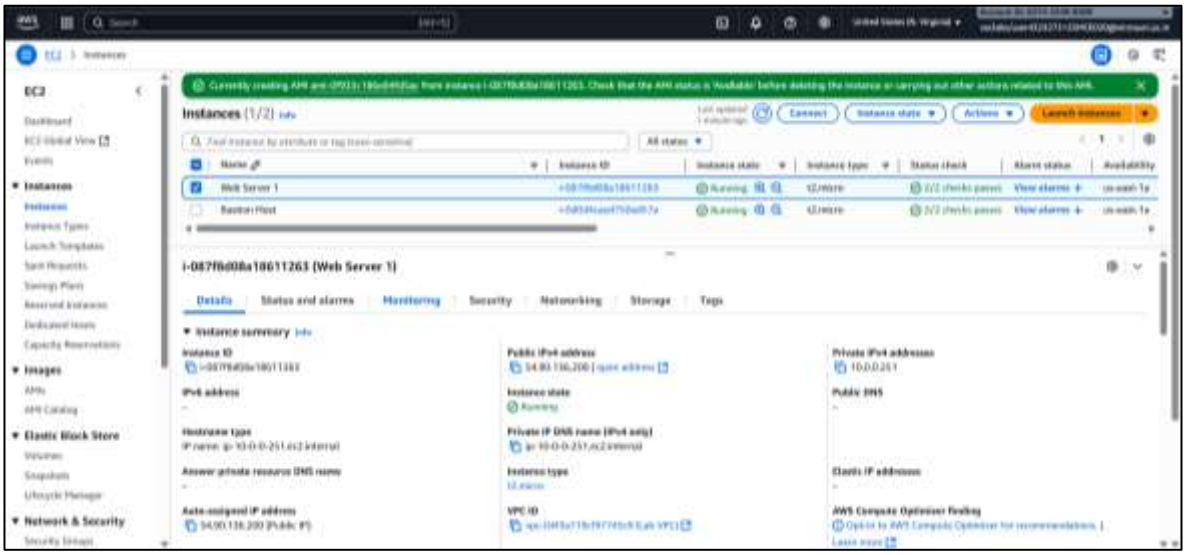
1. In the **AWS Management Console**, in the search box next to Services, search for and select **EC2**.
2. In the left navigation pane, choose **Instances**.

   First, you will confirm that the instance is running.

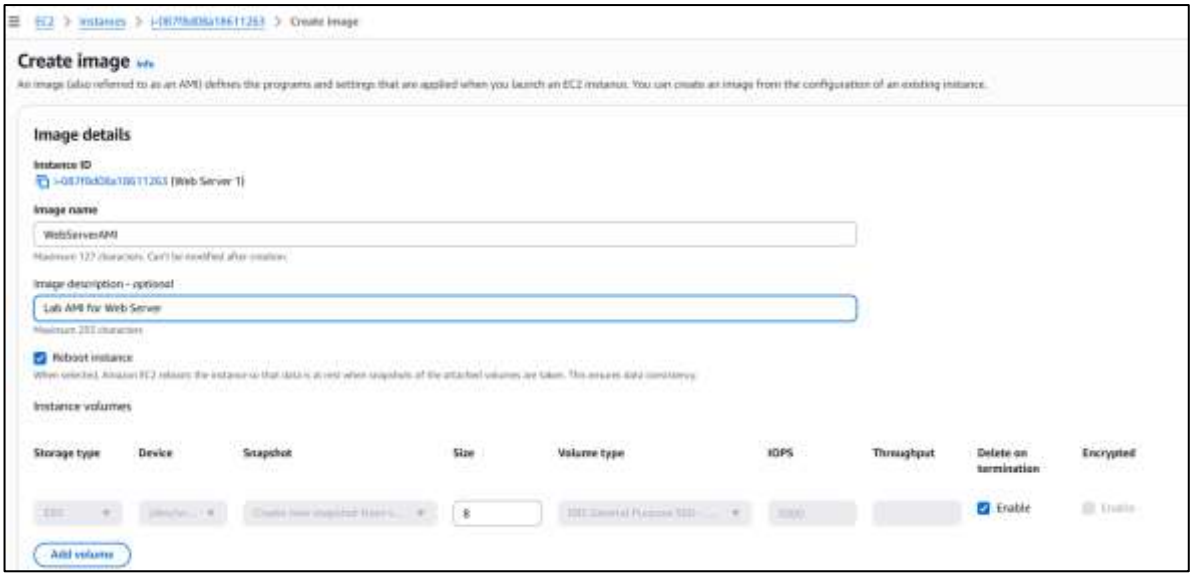
3. Wait until the **Status Checks** for **Web Server 1** displays 2/2 checks passed. If necessary, choose refresh to update the status.

   You will now create an AMI based upon this instance.

4. Select **Web Server 1**.



5. In the **Action** menu, choose **Image and templates** > **Create image**, then configure:
   - **Image name:** WebServerAMI
   - **Image description:** Lab AMI for Web Server



6. Choose **Create Image**

   A confirmation banner displays the **AMI ID** for your new AMI.

   You will use this AMI when launching the Auto Scaling group later in the lab.

➢ Task 2: Create a Load Balancer

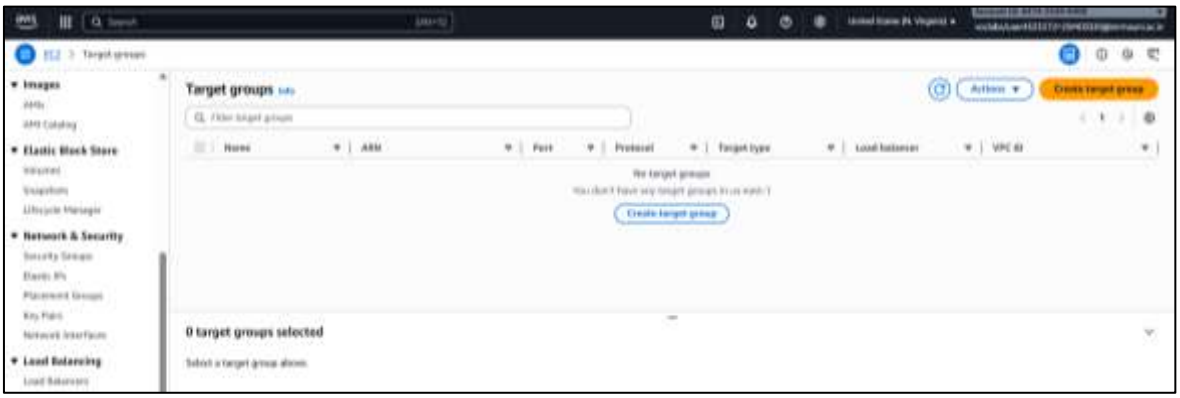
In this task, you will first create a target group and then you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

10. In the left navigation pane, choose **Target Groups**.

    Analysis: Target Groups define where to send traffic that comes into the Load Balancer. The Application Load Balancer can send traffic to multiple Target Groups based upon the URL of the incoming request, such as having requests from mobile

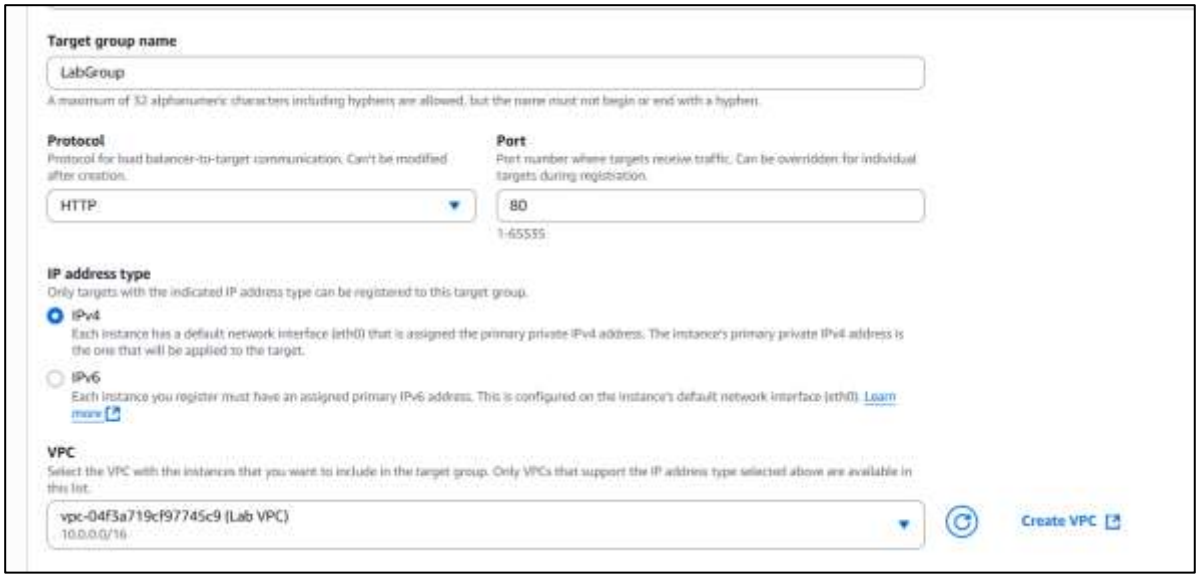
apps going to a different set of servers. Your web application will use only one Target Group.

- Choose **Create Target Group.**

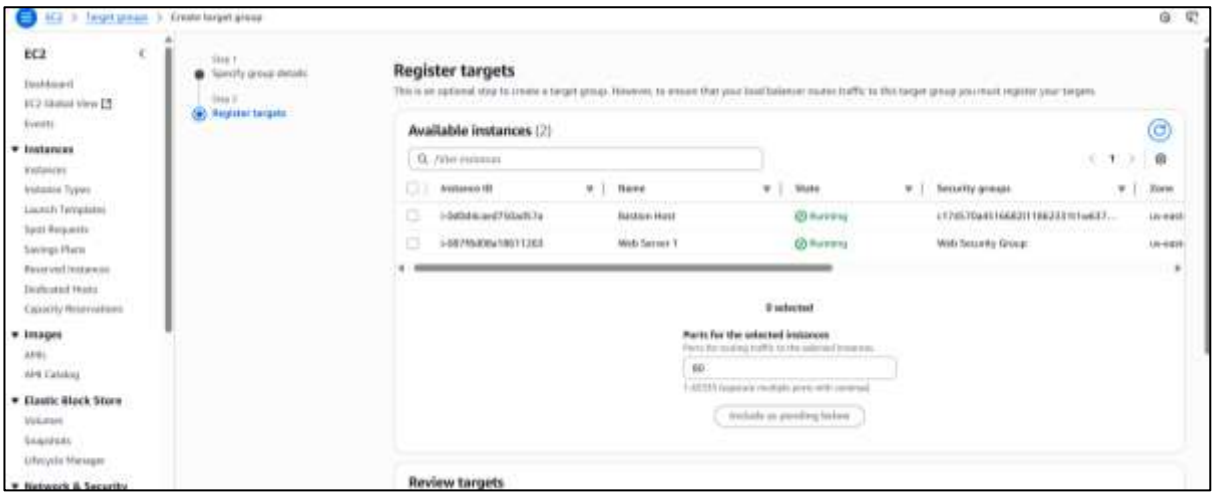


- Choose a target type: **Instances**

- **Target group name**, enter: LabGroup
- Select **Lab VPC** from the **VPC** drop-down menu.



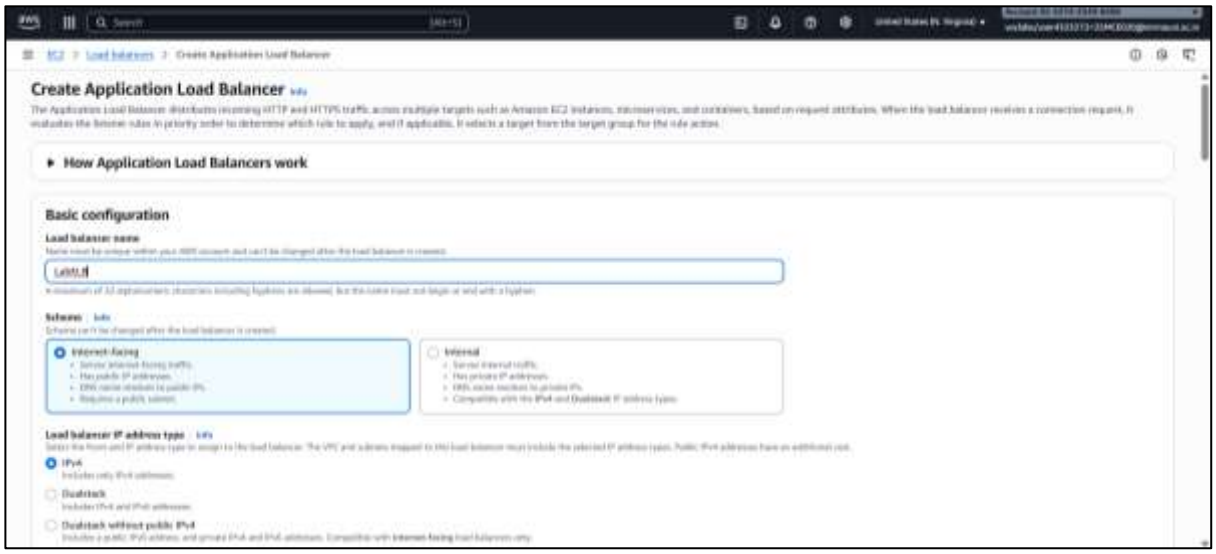
11. Choose **Next**. The **Register targets** screen appears.
12. Review the settings and choose **Create target group**



13. In the left navigation pane, choose **Load Balancers**.
14. At the top of the screen, choose **Create load balancer**.

    Several different types of load balancer are displayed. You will be using an Application Load Balancer that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: Comparison of Load Balancers.

15. Under **Application Load Balancer**, choose **Create.**
16. Under **Load balancer name**, enter: LabELB

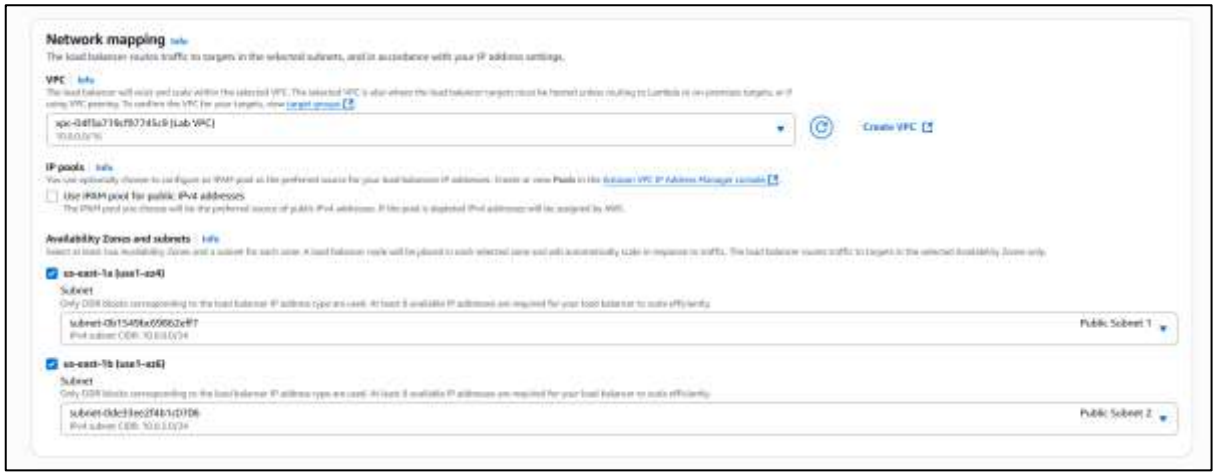


17. Scroll down to the **Network mapping** section, then:
- For **VPC**, choose **Lab VPC**

  You will now specify which subnets the Load Balancer should use. The load balancer will be internet facing, so you will select both Public Subnets.

- Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
- Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.

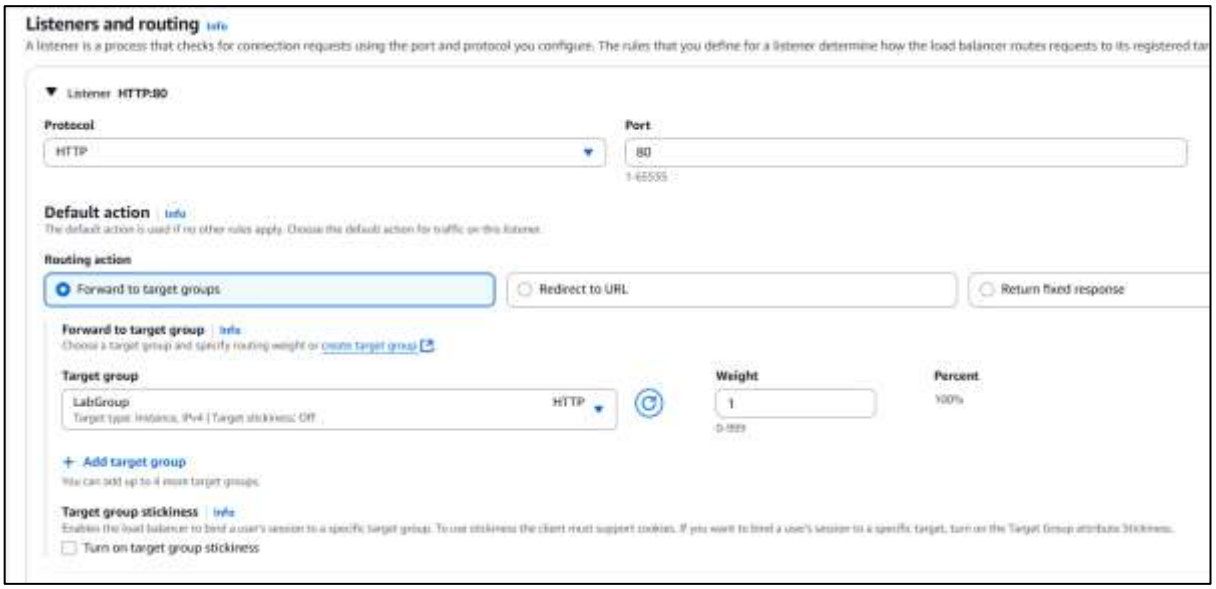
  You should now have two subnets selected: **Public Subnet 1** and **Public Subnet 2**.



18. In the **Security groups** section:
- Choose the Security groups drop down menu and select **Web Security Group**
- Below the drop down menu, choose the **X** next to the default security group to remove it.

  The **Web Security Group** security group should now be the only one that appears.

19. For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.

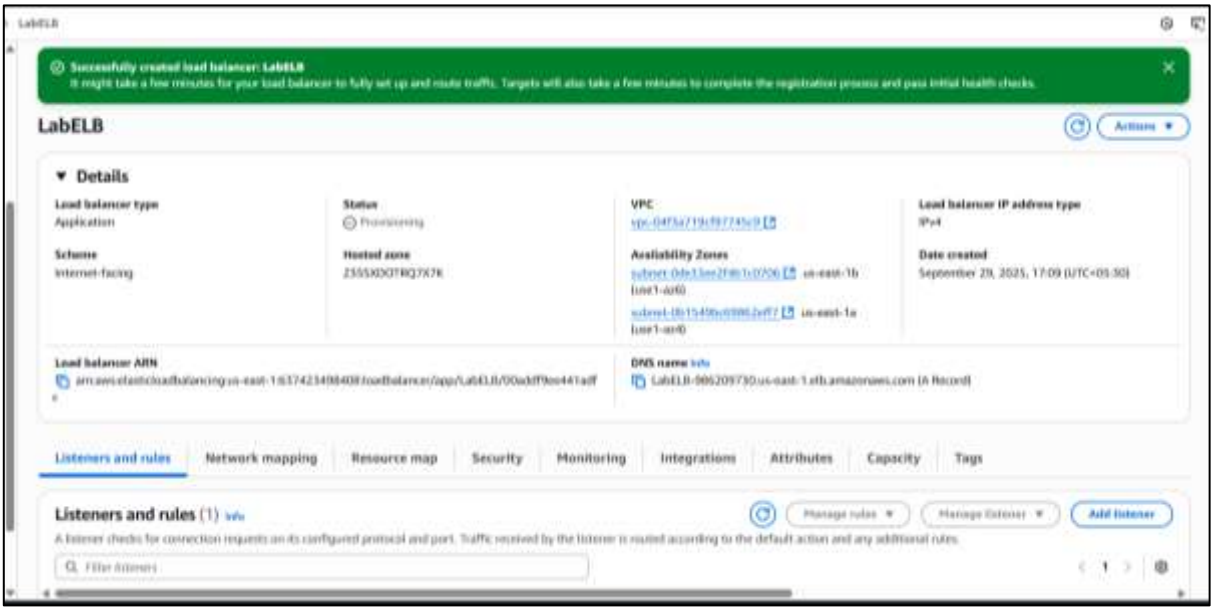


20. Scroll to the bottom and choose **Create load balancer**

    The load balancer is successfully created.

- Choose **View load balancer**

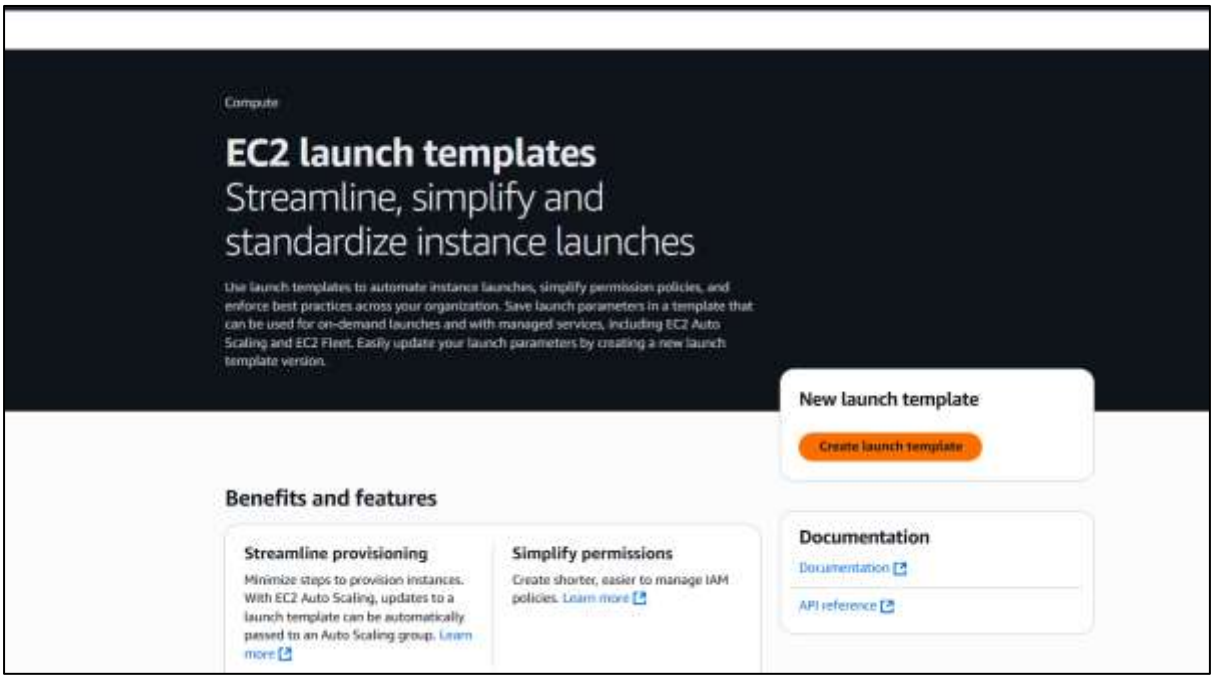
    The load balancer will show a state of provisioning. There is no need to wait until it is ready. Please continue with the next task.



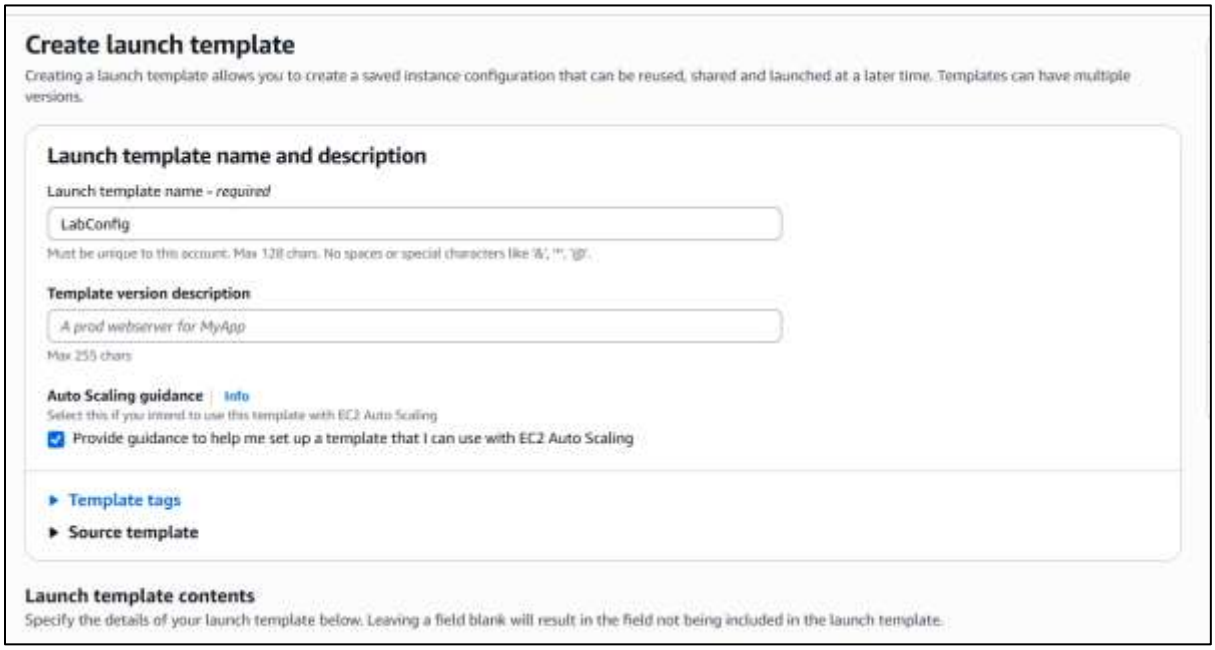
➢ Task 3: Create a Launch Template and an Auto Scaling Group

In this task, you will create a launch template for your Auto Scaling group. A launch template is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch template, you specify information for the instances such as the AMI, the instance type, a key pair, and security group.

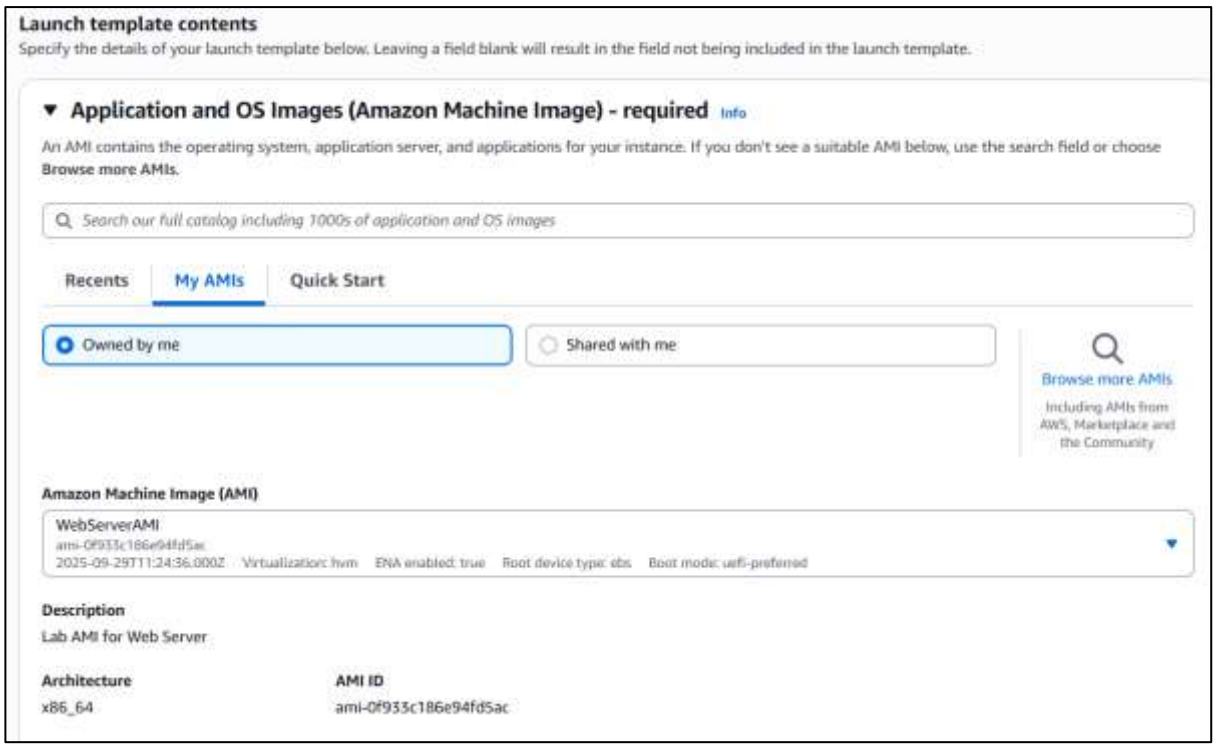
21. In the left navigation pane, choose **Launch Templates**.
22. Choose **Create Launch Template.**



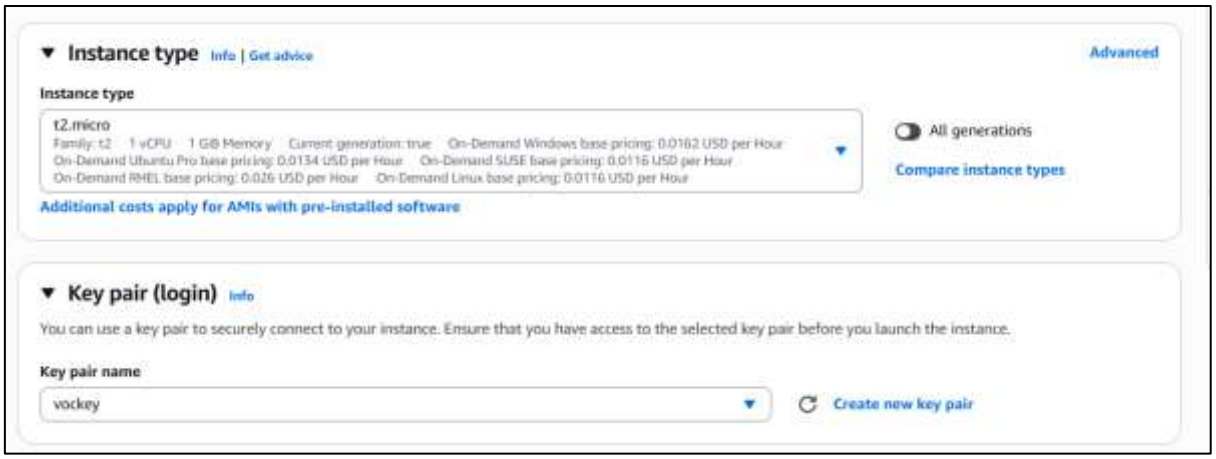
23. Configure the launch template settings and create it:
    - **Launch template name:** LabConfig



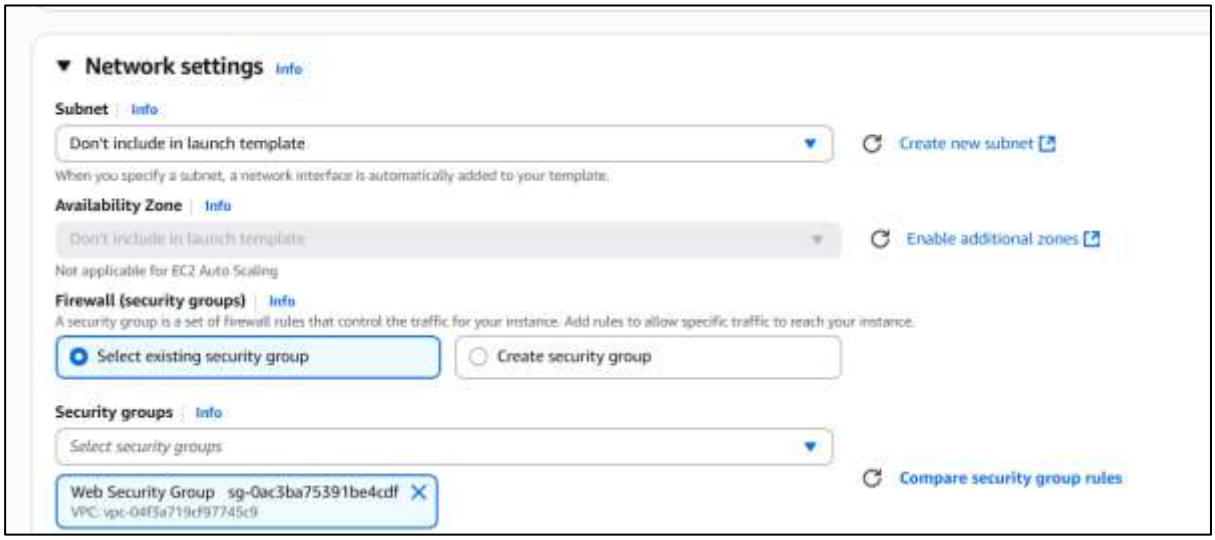
- Under **Auto Scaling guidance**, select Provide guidance to help me set up a template that I can use with EC2 Auto Scaling
- In the Application and OS Images (Amazon Machine Image) area, choose My AMIs.
- **Amazon Machine Image (AMI)**: choose Web Server AMI



- **Instance type:** choose t2.micro
- **Key pair name**: choose vockey

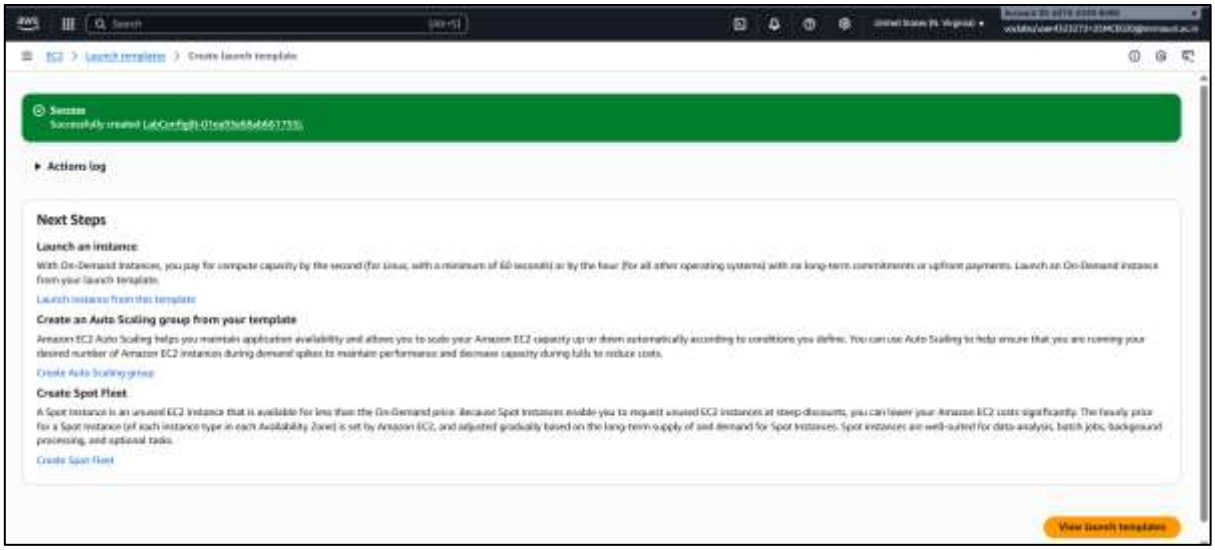


- **Firewall (security groups)**: choose Select existing security group
- **Security groups**: choose Web Security Group
- Scroll down to the **Advanced details** area and expand it.

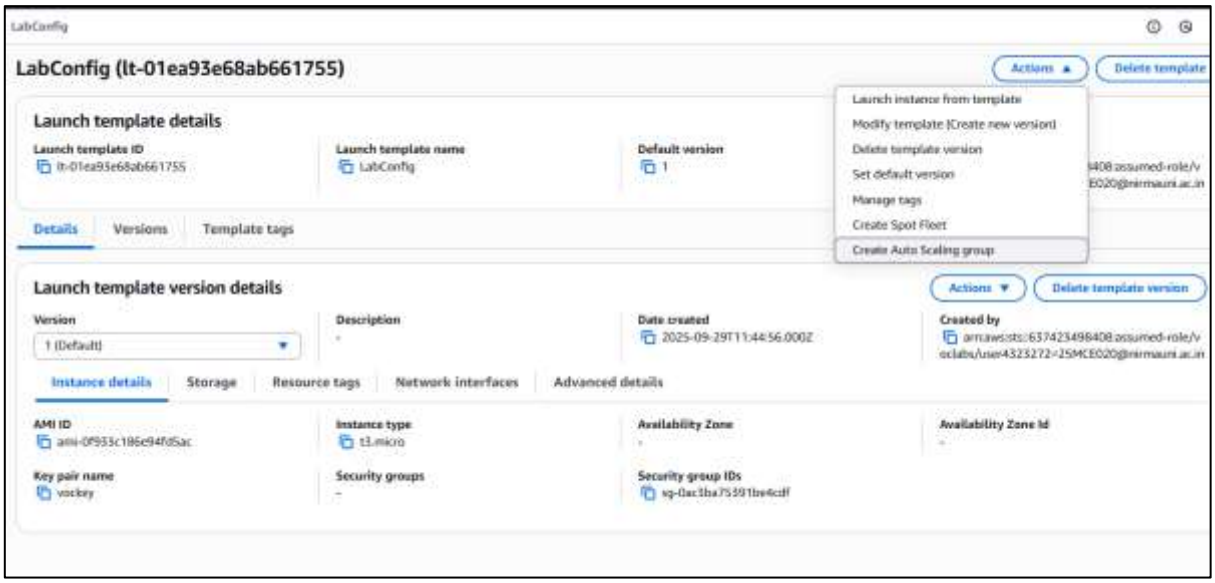


- Scroll down to the **Detailed CloudWatch monitoring** setting. Select Enable
- Choose **Create launch template**

Next, you will create an Auto Scaling group that uses this launch template.



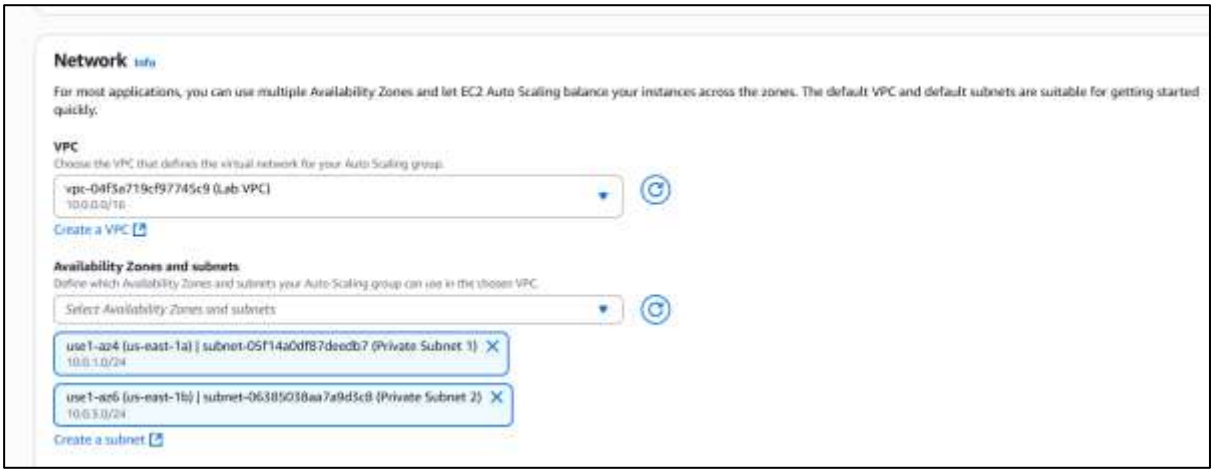
24. In the Success dialog, choose the **LabConfig** launch template.



25. From the **Actions** menu, choose Create Auto Scaling group.
26. Configure the details in Step 1 (Choose launch template or configuration):
- **Auto Scaling group name:** Lab Auto Scaling Group
- **Launch template**: confirm that the LabConfig template you just created is selected.

- Choose **Next.**

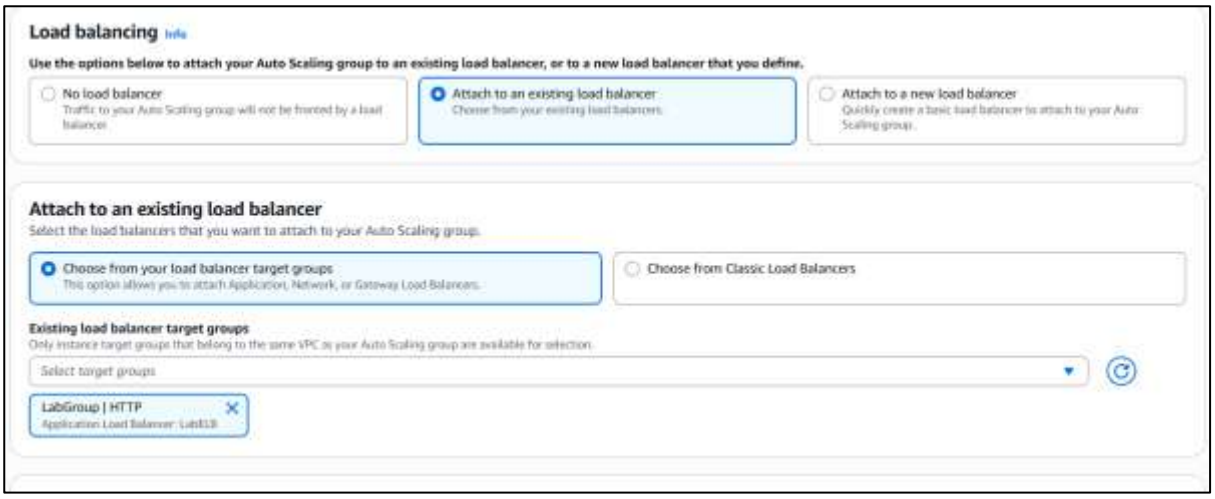
27. Configure the details in Step 2 (Choose instance launch options):
- **VPC**: choose Lab VPC
- **Availability Zones and subnets**: Choose Private Subnet 1 and then choose Private Subnet 2.
- Choose **Next.**



28. Configure the details in Step 3 (Configure advanced options):
- Choose **Attach to an existing load balancer**
- **Existing load balancer target groups**: select LabGroup.
- In the **Additional settings** pane:
- Select **Enable group metrics collection within CloudWatch**

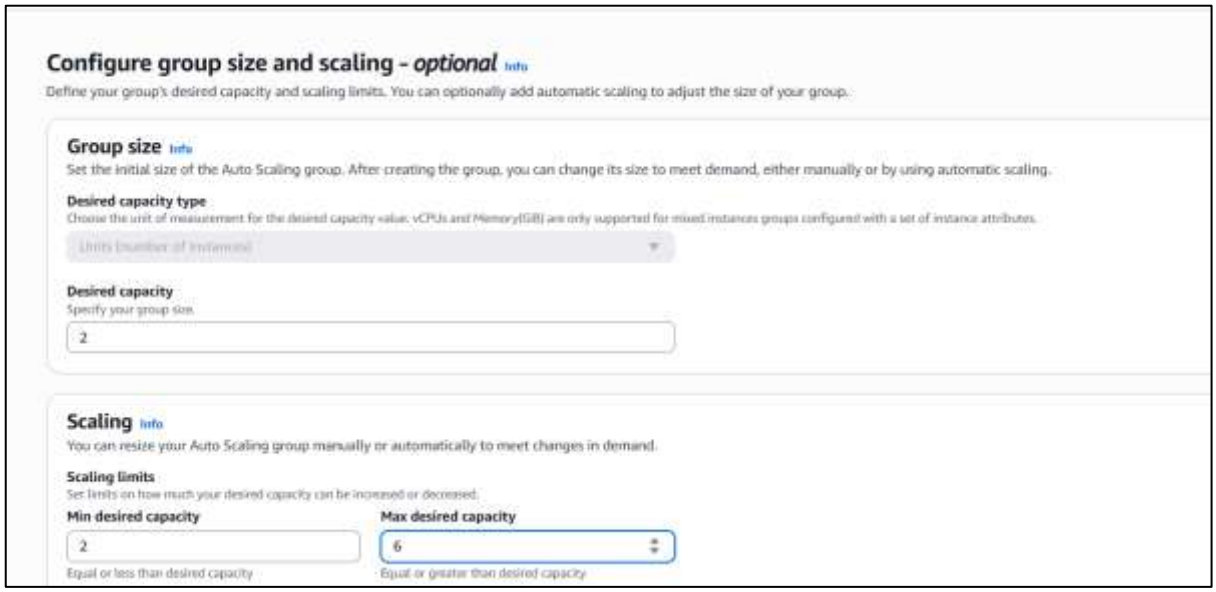
  This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.

- Choose **Next.**



29. Configure the details in Step 4 (Configure group size and scaling policies - optional):
   - Under **Group size**, configure:
   - **Desired capacity:** 2
   - **Minimum capacity:** 2
   - **Maximum capacity:** 6

   This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.

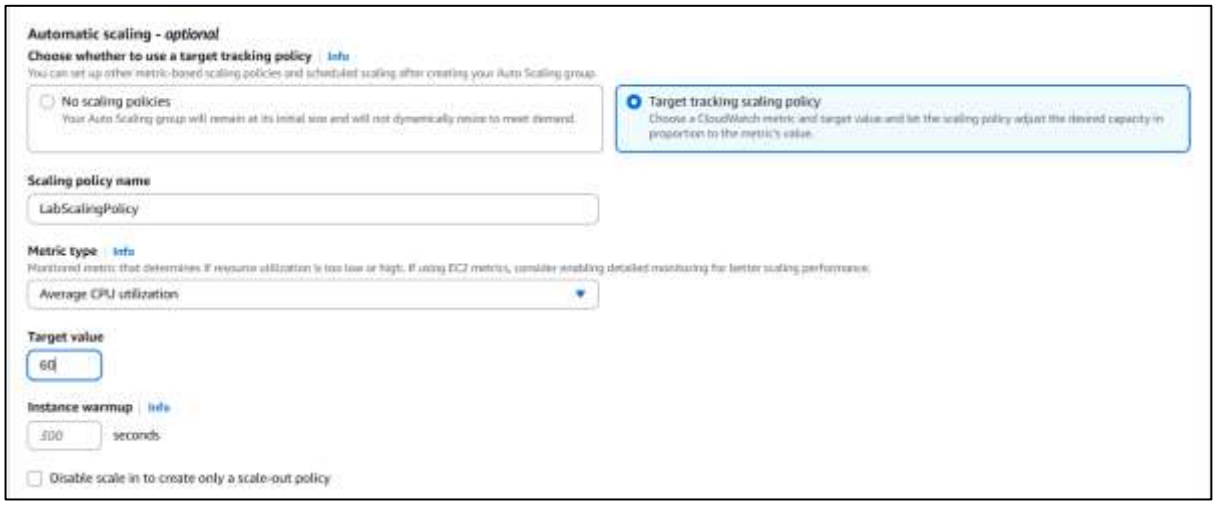


   - Under **Scaling policies**, choose Target tracking scaling policy and configure:
   - **Scaling policy name:** LabScalingPolicy
   - **Metric type:** Average CPU Utilization
   - **Target value:** 60

   This tells Auto Scaling to maintain an average CPU utilization across all instances at 60%. Auto Scaling will automatically add or remove capacity as required to keep the

metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.

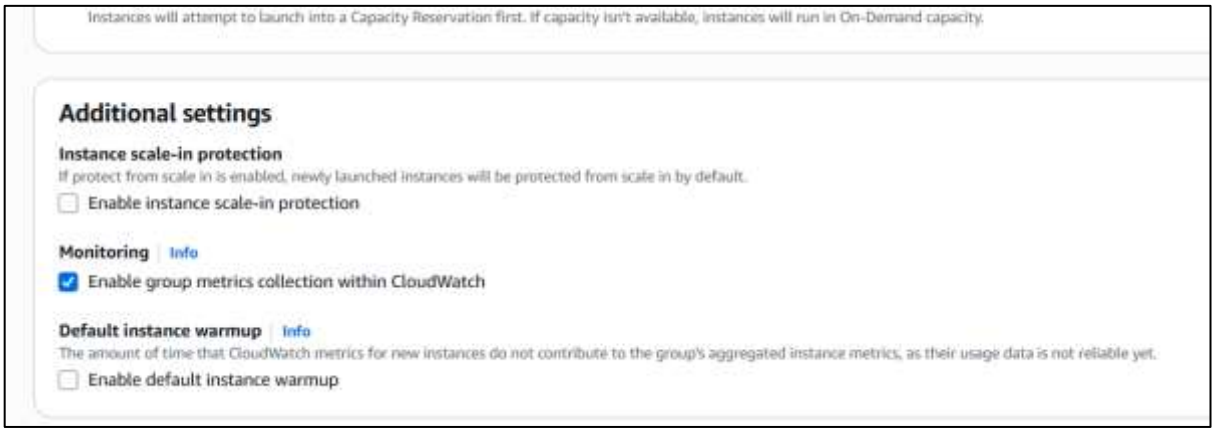
- Choose **Next.**



30. Configure the details in Step 5 (Add notifications - optional):

    Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.

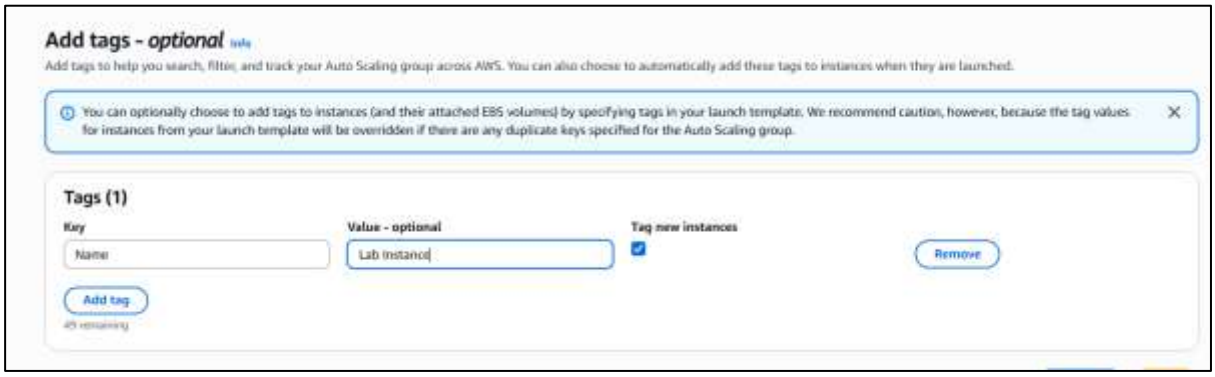
- Choose **Next.**



31. Configure the details in Step 6 (Add tags - optional):

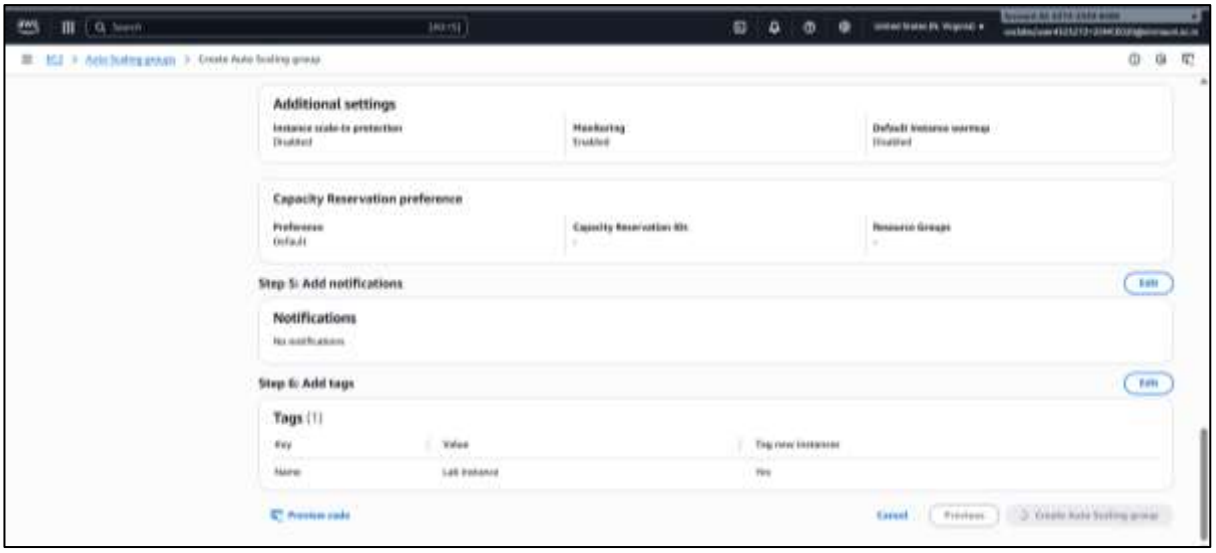
    Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.

- Choose **Add Tag** and Configure the following:
  - **Key:** Name
  - **Value:** Lab Instance
- Choose **Next.**



32. Configure the details in Step 6 (Review):
- Review the details of your Auto Scaling group
- Choose **Create Auto Scaling Group**

Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.



➢ Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

33. In the left navigation pane, choose **Instances**.

You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.

If the instances or names are not displayed, wait 30 seconds and choose refresh in the top-right.

Next, you will confirm that the new instances have passed their Health Check.

34. In the left navigation pane, choose **Target Groups**.
35. Select LabGroup.
36. Choose the **Targets** tab.

    Two target instances named Lab Instance should be listed in the target group.

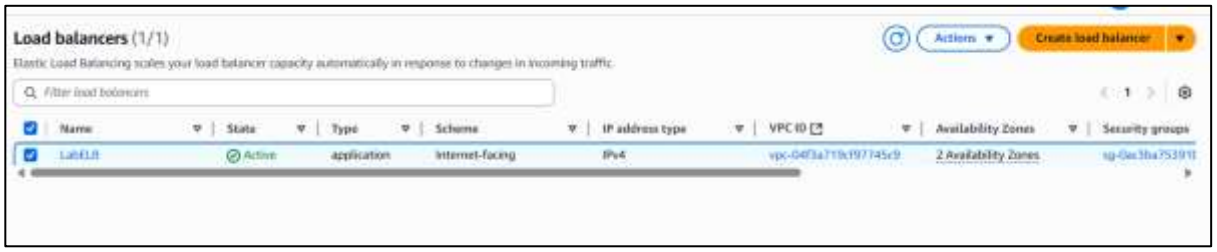
37. Wait until the **Status** of both instances transitions to healthy.

    Choose Refresh in the upper-right to check for updates if necessary.

    Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

    You can now access the Auto Scaling group via the Load Balancer.

38. In the left navigation pane, choose **Load Balancers**.
39. Select the LabELB load balancer.



40. In the Details pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".

    It should look similar to: LabELB-1998580470.us-west-2.elb.amazonaws.com

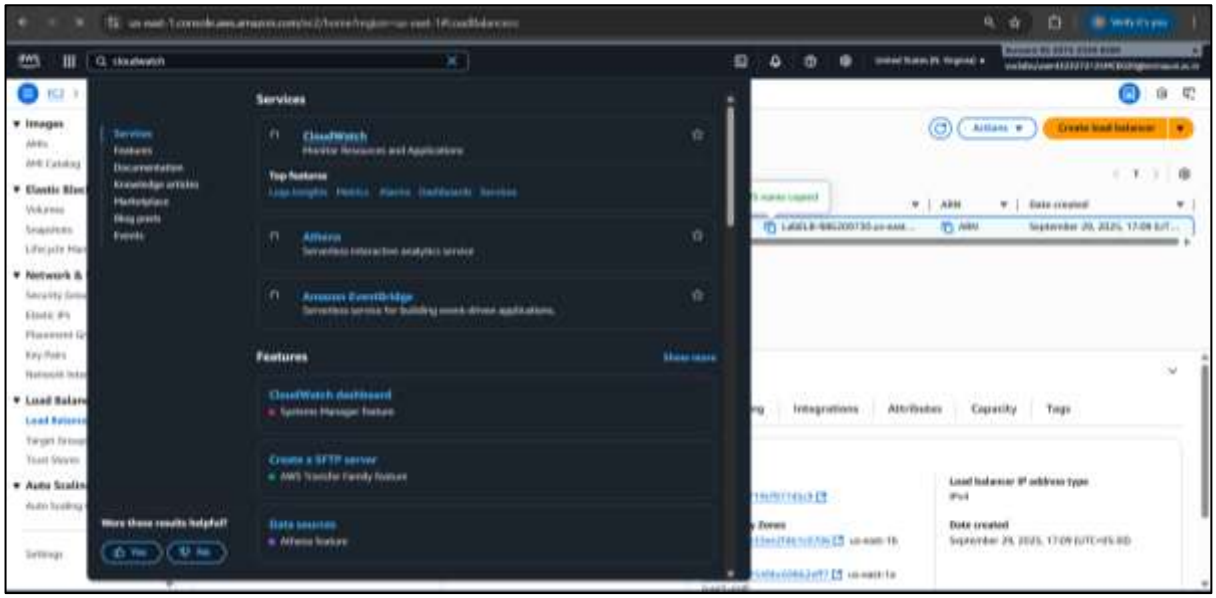
41. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.

    The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

➢ Task 5: Test Auto Scaling

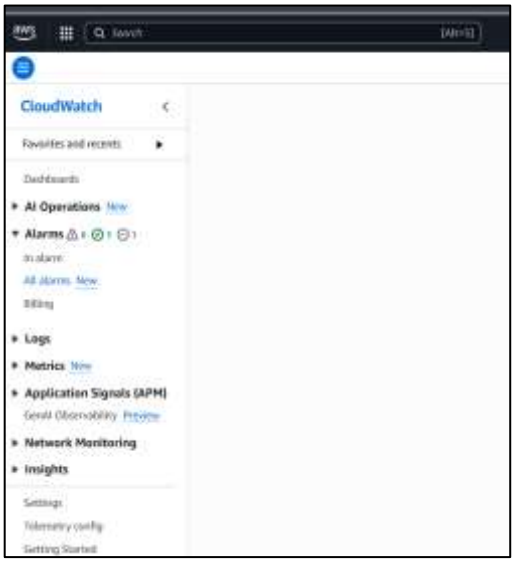
You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

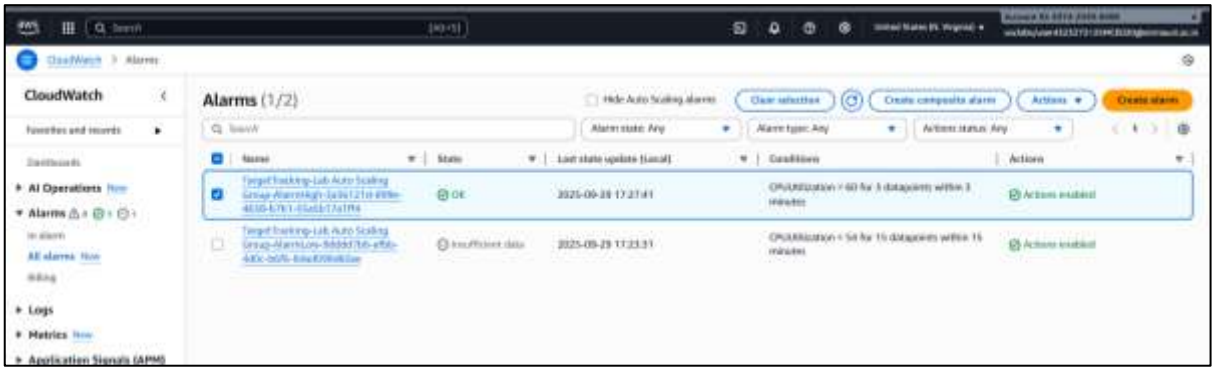
42. Return to the AWS Management Console, but do not close the application tab — you will return to it soon.
43. In the search box next to Services, search for and select **CloudWatch**.



44. In the left navigation pane, choose **All alarms**.

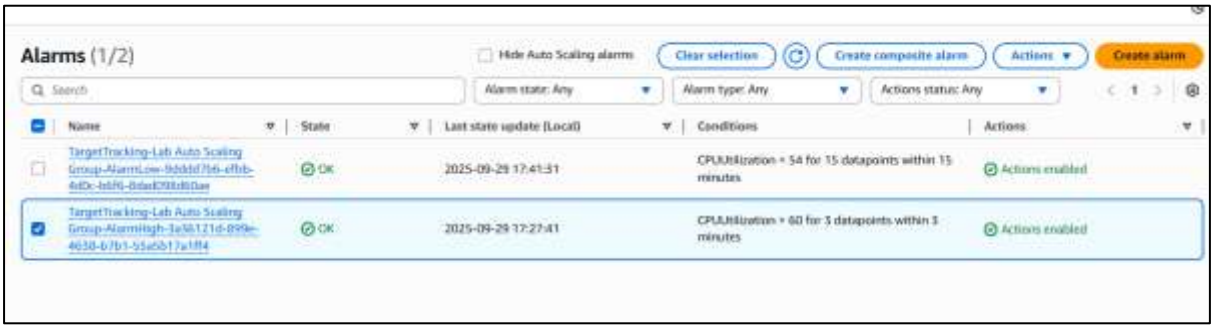
Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.





- On the **Services** menu, choose **EC2**.
- In the left navigation pane, choose **Auto Scaling Groups**.
- Select **Lab Auto Scaling Group**.
- In the bottom half of the page, choose the **Automatic Scaling** tab.
- Select **LabScalingPolicy**.
- Choose **Action** and **Edit**.
- Change the **Target Value** to 50.
- Choose **Update.**
- On the **Services** menu, choose **CloudWatch**.
- In the left navigation pane, choose **All alarms** and verify you see two alarms.

45. Choose the **OK** alarm, which has AlarmHigh in its name.



If no alarm is showing **OK**, wait a minute then choose refresh in the top-right until the alarm status changes.

The **OK** indicates that the alarm has not been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.

You will now tell the application to perform calculations that should raise the CPU level.

46. Return to the browser tab with the web application.
47. Choose **Load Test** beside the AWS logo.

This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.

48. Return to browser tab with the **CloudWatch** console.

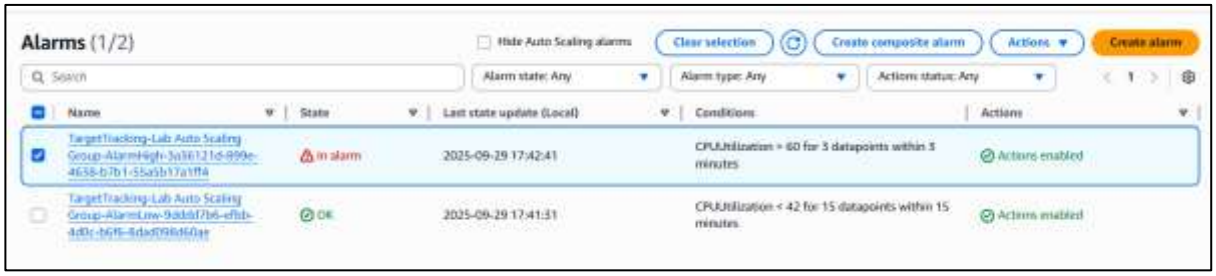
In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to In alarm.

You can choose Refresh in the top-right every 60 seconds to update the display.

You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.

49. Wait until the **AlarmHigh** alarm enters the In alarm state.

    You can now view the additional instance(s) that were launched.



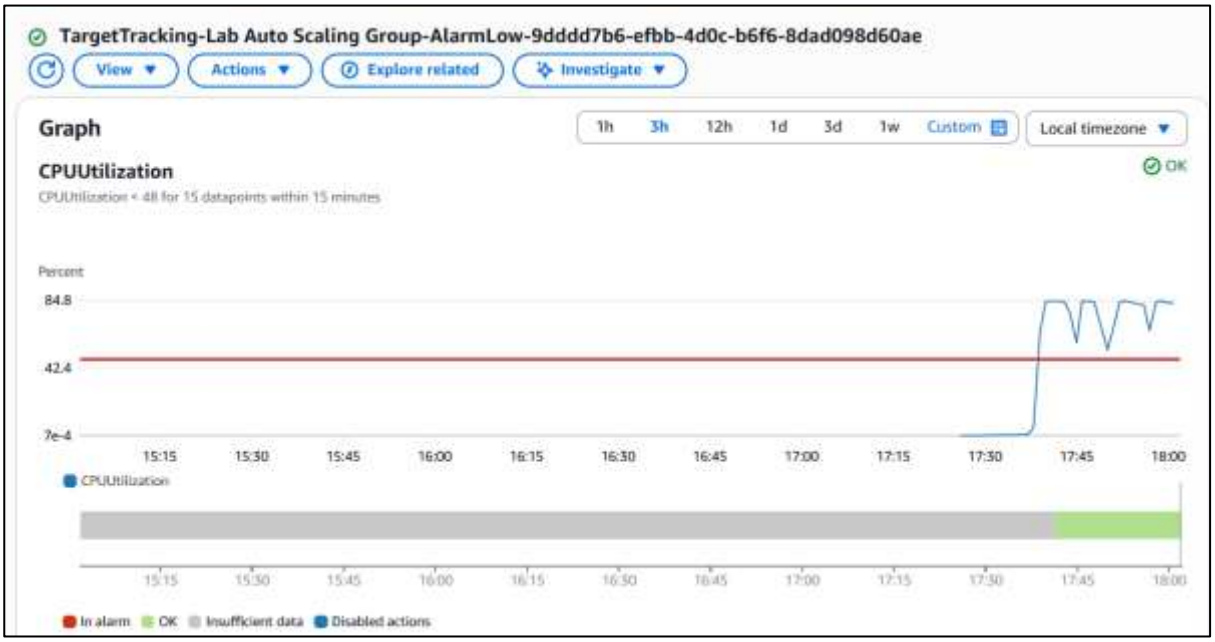
50. In the search box next to **Services,** search for and select **EC2**.
51. In the left navigation pane, choose **Instances**.

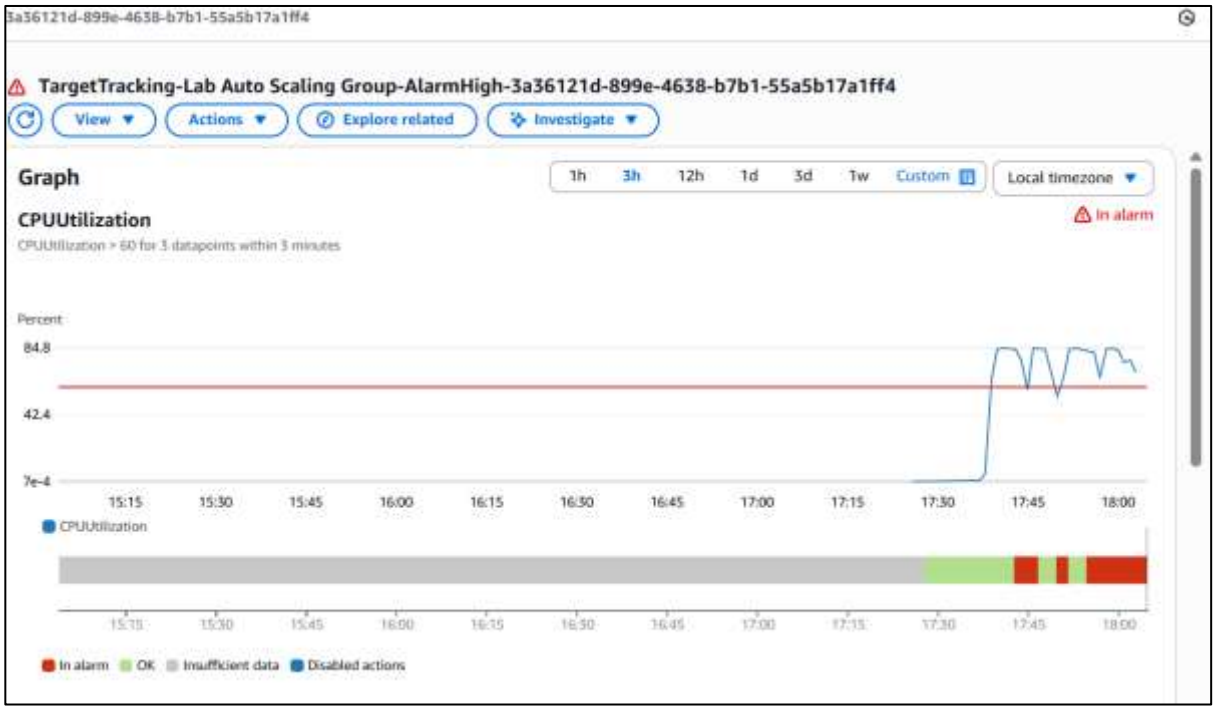
    More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the CloudWatch alarm.

➢ Task 6: Terminate Web Server 1

In this task, you will terminate Web Server 1. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

52. Select **Web Server 1** (and ensure it is the only instance selected).
53. In the **Instance State** menu, choose **Instance State** > **Terminate Instance**.
54. Choose Terminate.


TargetTracking-Lab Auto Scaling Group-AlarmLow-9dddd7b6-efbb-4d0c-b6f6-8dad098d60ae
View
Actions
Explore related
Investigate
Graph
1h
3h
12h
1d
3d
1w
Custom
Local timezone
CPUUtilization
OK
Percent
84.8
42.4
7e-4
15:15
15:30
15:45
16:00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
CPUUtilization
In alarm
OK
Insufficient data
Disabled actions


3a36121d-899e-4638-b7b1-55a5b17a1ff4
TargetTracking-Lab Auto Scaling Group-AlarmHigh-3a36121d-899e-4638-b7b1-55a5b17a1ff4
View
Actions
Explore related
Investigate
Graph
1h
3h
12h
1d
3d
1w
Custom
Local timezone
CPUUtilization
In alarm
Percent
84.8
42.4
7e-4
15:15
15:30
15:45
16:00
16:15
16:30
16:45
17:00
17:15
17:30
17:45
18:00
CPUUtilization
In alarm
OK
Insufficient data
Disabled actions

❖ Conclusion: