

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.

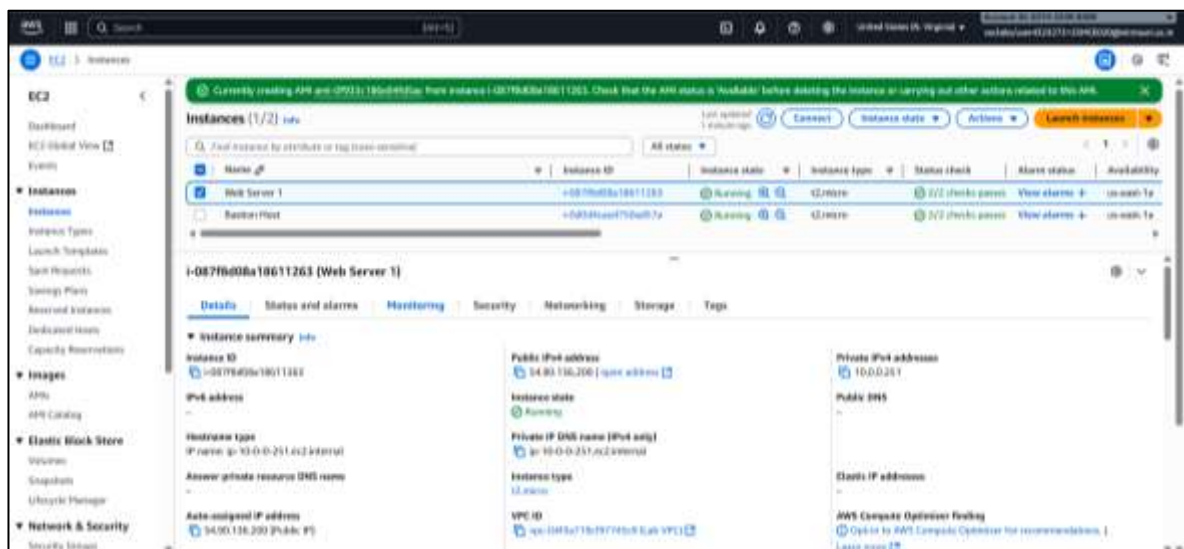
1. In the **AWS Management Console**, in the search box next to Services, search for and select **EC2**.
2. In the left navigation pane, choose **Instances**.

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**.

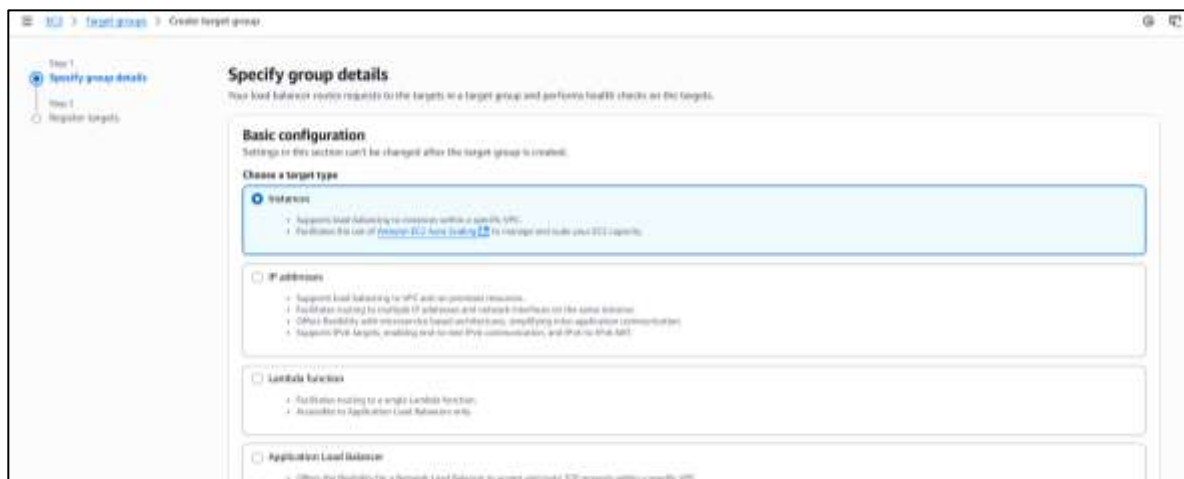
Analysis: Target Groups define where to send traffic that comes into the Load Balancer. The Application Load Balancer can send traffic to multiple Target Groups based upon the URL of the incoming request, such as having requests from mobile

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.

Target group name

LabGroup

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Protocol for load balancer-to-target communication. Can't be modified after creation.

HTTP

Port

Port number where targets receive traffic. Can be overridden for individual targets during registration.

80

1-65535

IP address type

Only targets with the indicated IP address type can be registered to this target group.

☒ IPv4

Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

☐ IPv6

Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

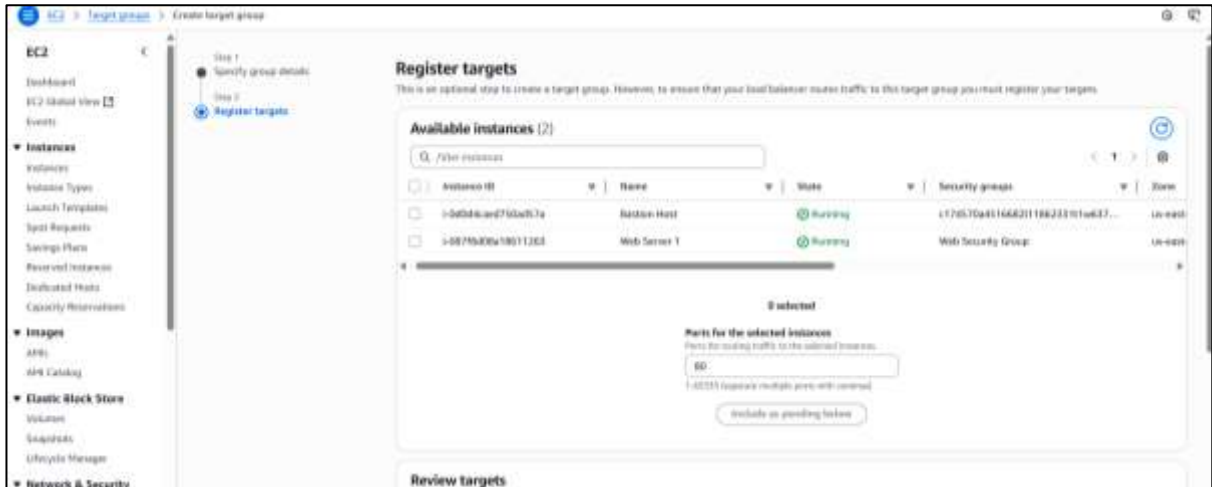
Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-04f3a719cf97745c9 (Lab VPC)

10.0.0.0/16

Create VPC

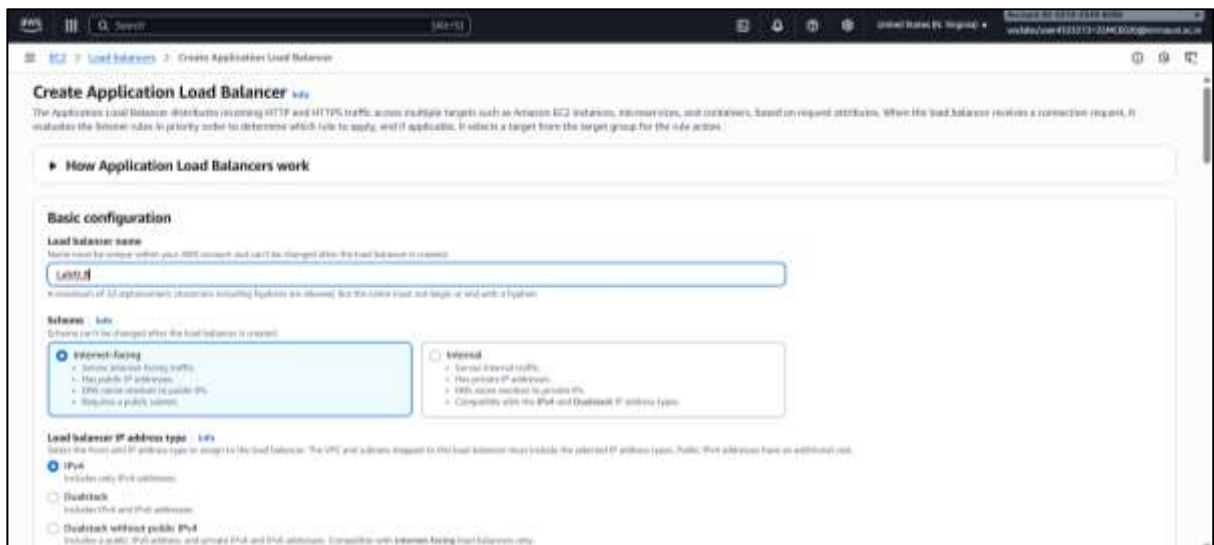
11. Choose **Next**. The **Register targets** screen appears.
12. Review the settings and choose **Create target group**



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**.

Network mapping Info

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC Info

The load balancer will route and route within the selected VPC. This selected VPC is also where the load balancer targets must be present unless routing to Lambda or on-premise targets, in which case VPC peering is required. To configure the VPC for your targets, view [target groups](#).

vpc-04f3a719cf97745c9 (Lab VPC)

IP pools Info

You can optionally choose to configure an IPAM pool as the preferred source for your load balancer IP addresses. Create or view [Pools in the Amazon VPC IP Address Manager console](#).

☐ Use IPAM pool for public IPv4 addresses

The IPAM pool addresses will be the preferred source of public IPv4 addresses. If the pool is depleted IPv4 addresses will be assigned by AWS.

Availability Zones and subnets Info

Select at least two Availability Zones and a subnet for each zone. A load balancer route will be placed in each selected zone and will automatically scale in response to traffic. The load balancer routes traffic to targets in the selected Availability Zones only.

☒ us-east-1a (us-east-1)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are valid. At least 8 available IP addresses are required for your load balancer to route efficiently.

subnet-0b1549ba0596b2ef7

Public Subnet 1

☒ us-east-1b (us-east-1)

Subnet

Only CIDR blocks corresponding to the load balancer IP address type are valid. At least 8 available IP addresses are required for your load balancer to route efficiently.

subnet-0d3c93ee374b1d1d6

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.

Security groups Info

A security group is a set of firewall rules that control the traffic to your load balancer. Select an existing security group, or you can [create a new security group](#).

Security groups

Select up to 5 security groups

Search

☐ default
sg-06e9d5261fc2545e VPC: vpc-04f3a719cf97745c9

☒ Web Security Group
sg-0a3ba75391be4c1ff VPC: vpc-04f3a719cf97745c9

☐ DB Security Group
sg-0cb61041083da7afa VPC: vpc-04f3a719cf97745c9

☐ c174570a4516682f11862331t1w637423498408-BastionSecurityGroup-QePSy8SHIPDL
sg-0e5c908b4c24b91ba VPC: vpc-04f3a719cf97745c9

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

Listeners and routing [info](#)

A listener is a process that checks for connection requests using the port and protocol you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets.

Listener HTTP:80

Protocol: HTTP Port: 80

Default action [info](#)

The default action is used if no other rules apply. Choose the default action for traffic on this listener.

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 LabGroup HTTP Weight: 1 Percent: 100%

Target group stickiness [info](#)

Enables the load balancer to bind a user's session to a specific target group. To use stickiness the client must support cookies. If you want to bind a user's session to a specific target, turn on the Target Group attribute Stickiness.

☐ Turn on target group stickiness

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.

LabELB

Successfully created load balancer: LabELB

It might take a few minutes for your load balancer to fully set up and route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.

Details

Load balancer type: Application

Status: Provisioning

Scheme: Internet-facing

Hosted zone: Z35SXDD7RQ7X78

VPC: vpc-04f3a719:97745c9

Availability Zones: us-east-1b, us-east-1a

Load balancer IP address type: IPv4

Date created: September 20, 2025, 17:09 (UTC+05:30)

Load balancer ARN: arn:aws:elasticloadbalancing:us-east-1:617423498408:loadbalancer/app/LabELB/00ackff9ee441aef

DNS name: LabELB-996209730.us-east-1.elb.amazonaws.com

Listeners and rules (1)

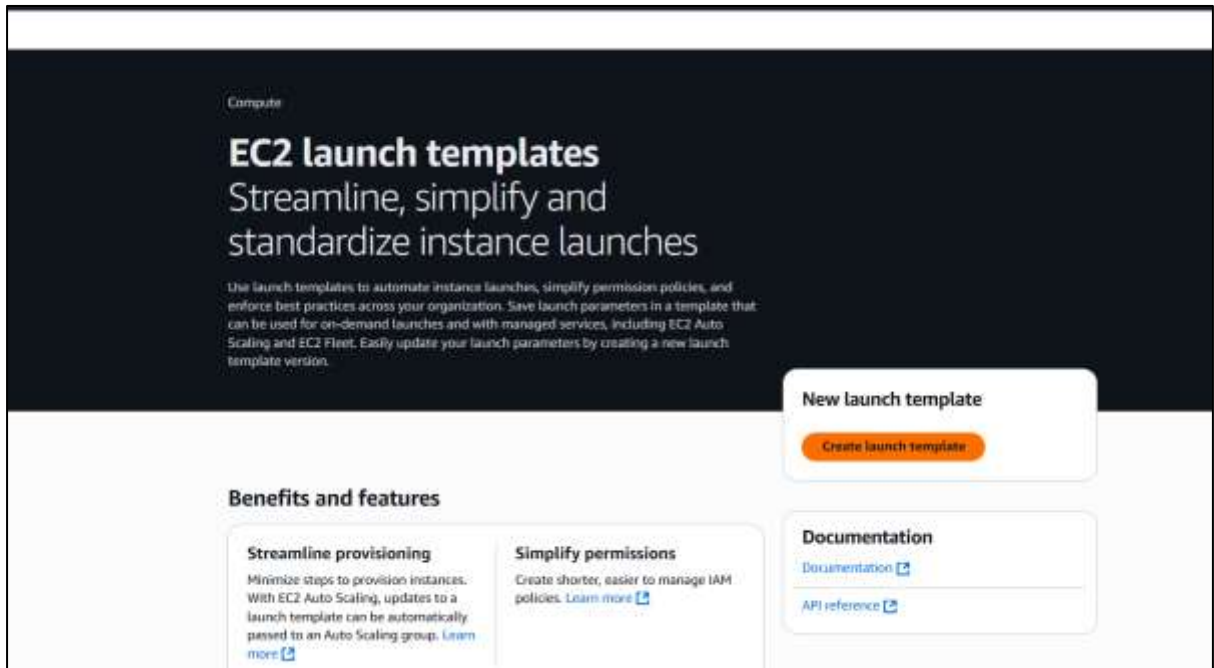
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

➤ 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

The screenshot shows the 'Create launch template' form. The title is 'Create launch template' with a subtitle explaining that creating a launch template allows you to create a saved instance configuration that can be reused, shared, and launched at a later time. The form has two main sections: 'Launch template name and description' and 'Template version description'. In the first section, the 'Launch template name' field is filled with 'LabConfig'. Below it, a note states 'Must be unique to this account. Max 128 chars. No spaces or special characters like ", !, or @'. In the second section, the 'Template version description' field is filled with 'A prod webserver for MyApp'. Below it, a note states 'Max 255 chars'. There is an 'Auto Scaling guidance' section with a link to 'Info' and a checkbox labeled 'Provide guidance to help me set up a template that I can use with EC2 Auto Scaling', which is checked. At the bottom, there are two expandable sections: 'Template tags' and 'Source template'. The 'Launch template contents' section at the very bottom states 'Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.'

- 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

Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

▼ **Application and OS Images (Amazon Machine Image) - required** [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 **My AMIs** Quick Start

☒ Owned by me ☐ Shared with me

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

WebServerAMI
ami-0f933c186e94fd5ac
2025-09-29T11:24:36.800Z · Virtualization: hvm · ENA enabled: true · Root device type: ebs · Boot mode: uefi-preferred

Description
Lab AMI for Web Server

| Architecture | AMI ID |
|--------------|-----------------------|
| x86_64 | ami-0f933c186e94fd5ac |

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

▼ **Instance type** [Info](#) | [Get advice](#) [Advanced](#)

Instance type

t2.micro
Family: t2 · 1 vCPU · 1 GB Memory · Current generation: true · On-Demand Windows base pricing: 0.0162 USD per Hour · On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour · On-Demand SUSE base pricing: 0.0116 USD per Hour · On-Demand RHEL base pricing: 0.026 USD per Hour · On-Demand Linux base pricing: 0.0116 USD per Hour

☐ All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ **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

vockey [Create new key pair](#)

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

Network settings [Info](#)

Subnet [Info](#)

Don't include in launch template

When you specify a subnet, a network interface is automatically added to your template.

[Create new subnet](#)

Availability Zone [Info](#)

Don't include in launch template

Not applicable for EC2 Auto Scaling

[Enable additional zones](#)

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.

☒ Select existing security group ☐ Create security group

Security groups [Info](#)

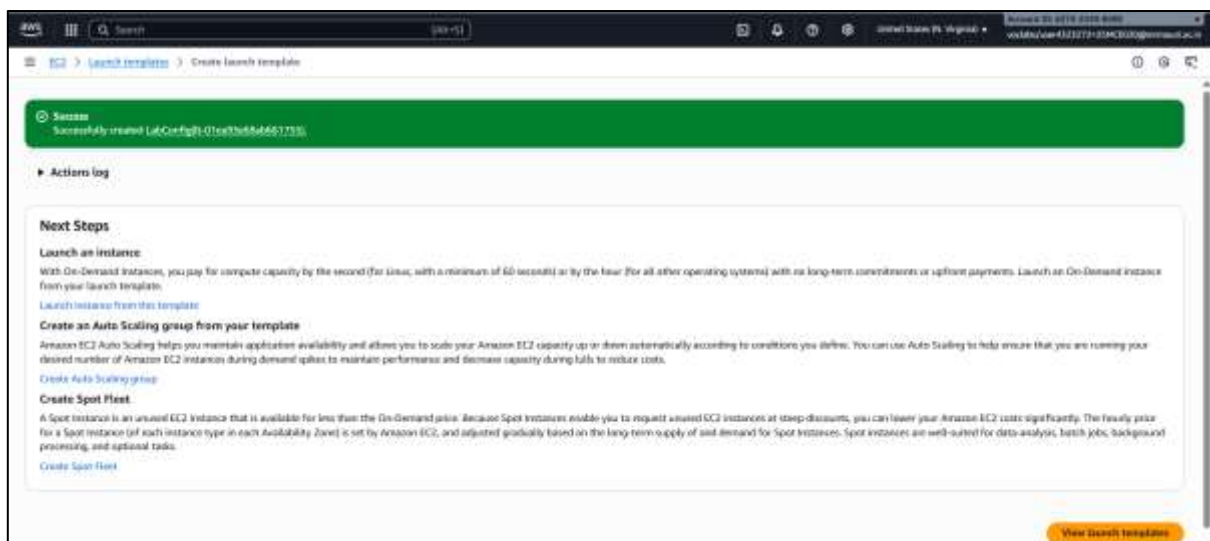
Select security groups

Web Security Group sg-0ac3ba75391be4cdf
VPC: vpc-04f5a719cf97745c9

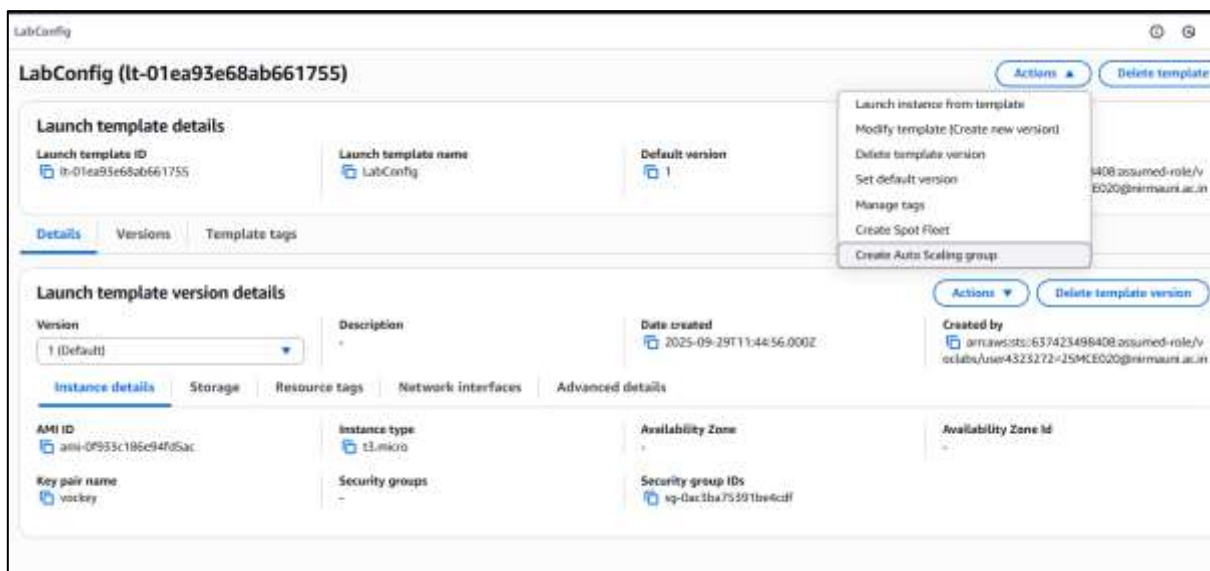
[Compare security group rules](#)

- 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.

The screenshot shows the 'Choose launch template' step in the AWS Management Console. On the left, a progress bar indicates the steps: Step 1 (selected), Step 2, Step 3 (optional), Step 4 (optional), Step 5 (optional), Step 6 (optional), and Step 7 (optional). The main content area is titled 'Choose launch template' with a subtitle 'Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.' There are two input fields: 'Auto Scaling group name' with the value 'Lab Auto Scaling Group' and 'Launch template' with the value 'LabConfig'. A note states: 'For accounts created after May 13, 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.' At the bottom, there is a 'Create a launch template' link.

- 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**.

The screenshot shows the 'Network' step in the AWS Management Console. It includes a 'VPC' section with a dropdown menu showing 'vpc-04f5e719cf97745c9 (Lab VPC)' and a 'Create a VPC' link. Below that is an 'Availability Zones and subnets' section with a dropdown menu showing 'use1-az4 (us-east-1a) | subnet-05f14a0df87deedb7 (Private Subnet 1)' and 'use1-az6 (us-east-1b) | subnet-06385038aa7a9d3c8 (Private Subnet 2)'. There are 'X' icons to remove each selection and a 'Create a subnet' link at the bottom.

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**.

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.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

☒ Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

☐ Choose from Classic Load Balancers

Existing load balancer target groups

Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

LabGroup | HTTP
Application Load Balancer | LabE3B

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.

Configure group size and scaling - optional [Info](#)

Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size [Info](#)

Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type

Choose the unit of measurement for the desired capacity value. vCPUs and Memory (GB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity

Specify your group size.

2

Scaling [Info](#)

You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits

Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity

2

Equal or less than desired capacity

Max desired capacity

6

Equal or greater than desired capacity

- 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**.

The screenshot shows the 'Automatic scaling - optional' section of the AWS Auto Scaling console. It prompts the user to 'Choose whether to use a target tracking policy'. Two options are available: 'No scaling policies' (selected with a radio button) and 'Target tracking scaling policy' (highlighted with a blue border and selected with a blue dot). The 'Target tracking scaling policy' option includes a description: 'Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.' Below this, the 'Scaling policy name' field contains 'LabScalingPolicy'. The 'Metric type' dropdown is set to 'Average CPU utilization'. The 'Target value' input field contains '60'. The 'Instance warmup' field is set to '300 seconds'. At the bottom, there is a checkbox for 'Disable scale in to create only a scale-out policy' which is currently unchecked.

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**.

The screenshot shows the 'Additional settings' section of the AWS Auto Scaling console. At the top, a note states: 'Instances will attempt to launch into a Capacity Reservation first. If capacity isn't available, instances will run in On-Demand capacity.' Below this, the 'Instance scale-in protection' section has a description: 'If protect from scale in is enabled, newly launched instances will be protected from scale in by default.' and an unchecked checkbox 'Enable instance scale-in protection'. The 'Monitoring' section has a description: 'Enable group metrics collection within CloudWatch' and a checked checkbox. The 'Default instance warmup' section has a description: 'The amount of time that CloudWatch metrics for new instances do not contribute to the group's aggregated instance metrics, as their usage data is not reliable yet.' and an unchecked checkbox.

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**.

Add tags - optional [info](#)

Add tags to help you search, filter, and track your Auto Scaling group across AWS. You can also choose to automatically add these tags to instances when they are launched.

Tags (1)

| Key | Value - optional | Tag new instances | |
|------|------------------|-------------------------------------|------------------------|
| Name | Lab Instance | <input checked="" type="checkbox"/> | Remove |

[Add tag](#)

49 remaining

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.

Additional settings

Instance scale-in protection: Disabled

Monitoring: Enabled

Default instance weights: Disabled

Capacity Reservation preference

Preference: Default

Capacity Reservation ID:

Resource Groups:

Step 5: Add notifications [Edit](#)

Notifications

No notifications

Step 6: Add tags [Edit](#)

Tags (1)

| Key | Value | Tag new instances |
|------|--------------|-------------------|
| Name | Lab Instance | Yes |

[Previous code](#) [Cancel](#) [Previous](#) [Create Auto Scaling group](#)

➤ 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.

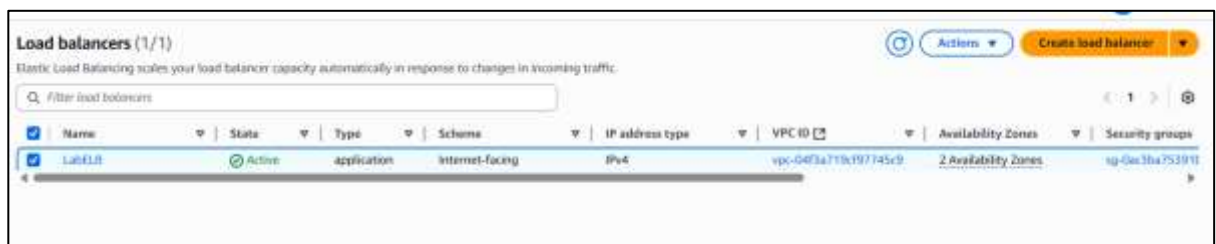
Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.

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.

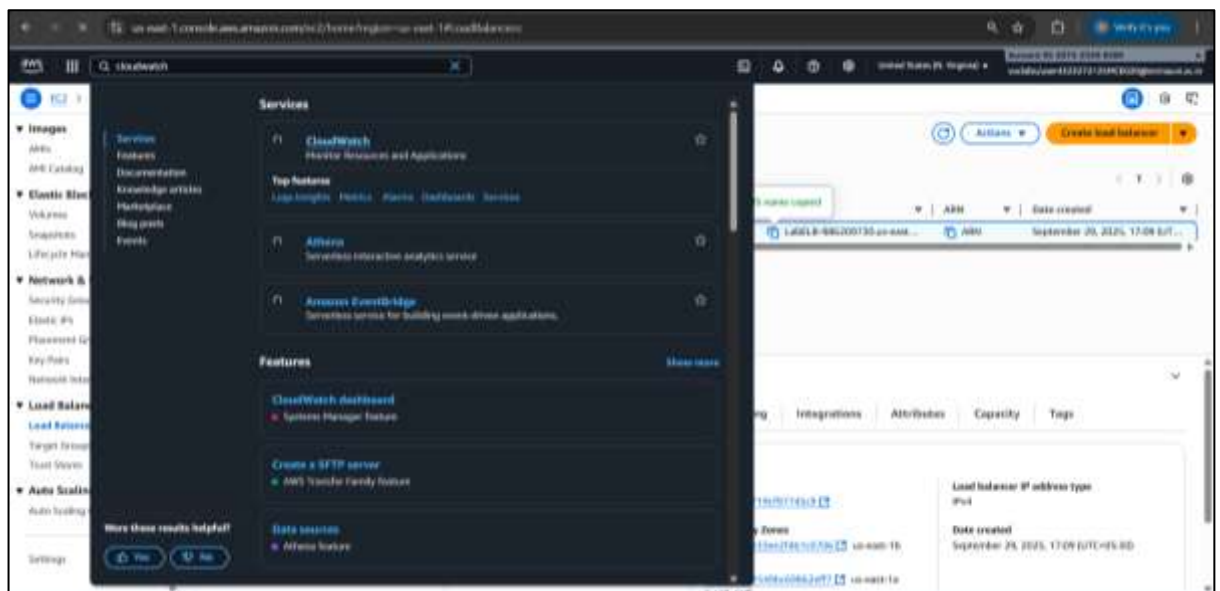
The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.



➤ Task 5: Test Auto Scaling

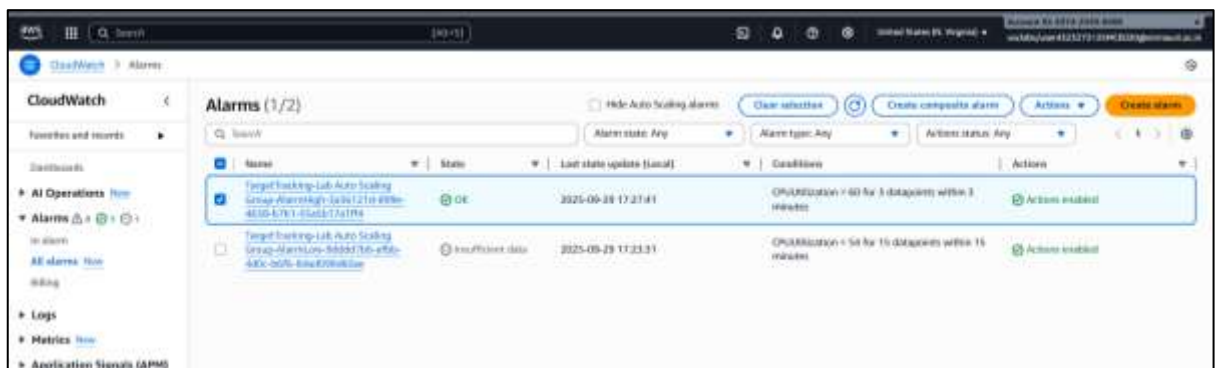
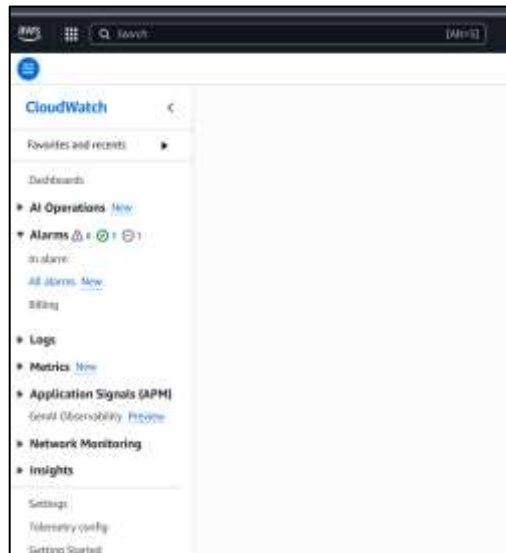
You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

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**.



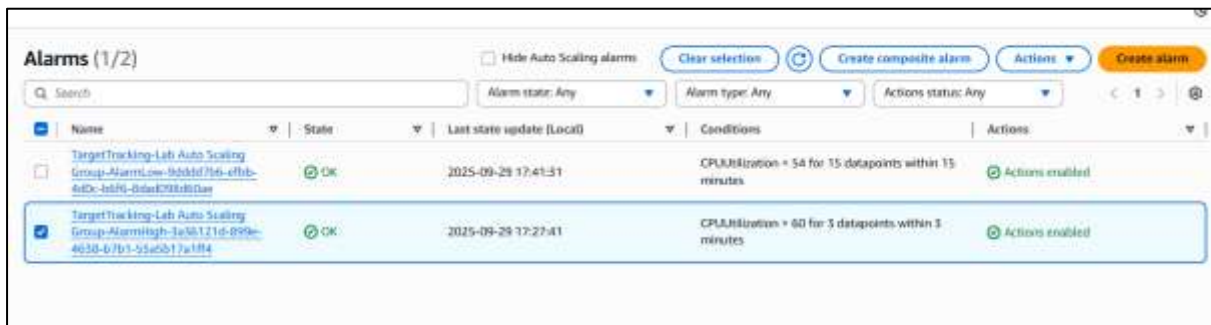
44. In the left navigation pane, choose **All alarms**.

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.

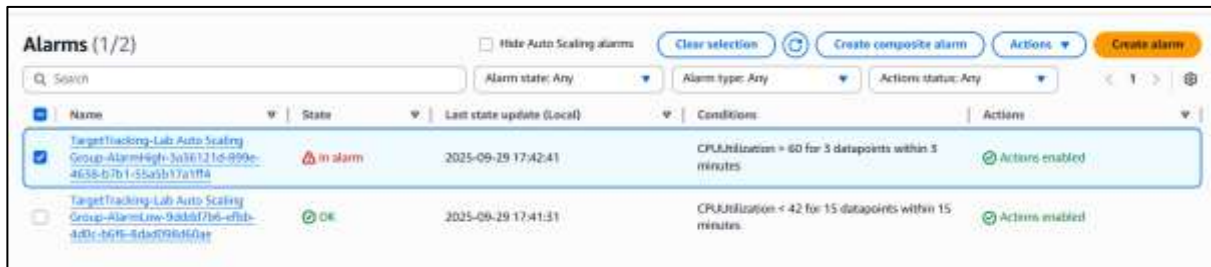
In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to In alarm.

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

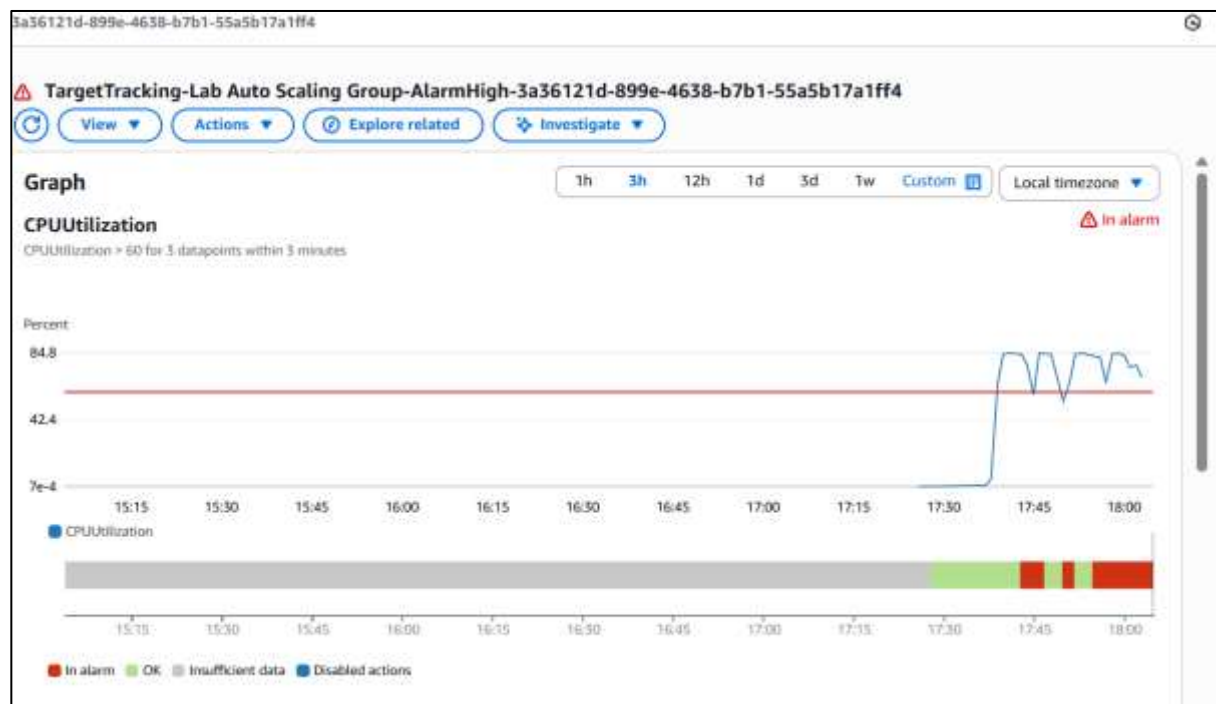
In this task, you will terminate Web Server 1. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

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**.





❖ Conclusion: