Scaling and Load Balancing Your Architecture

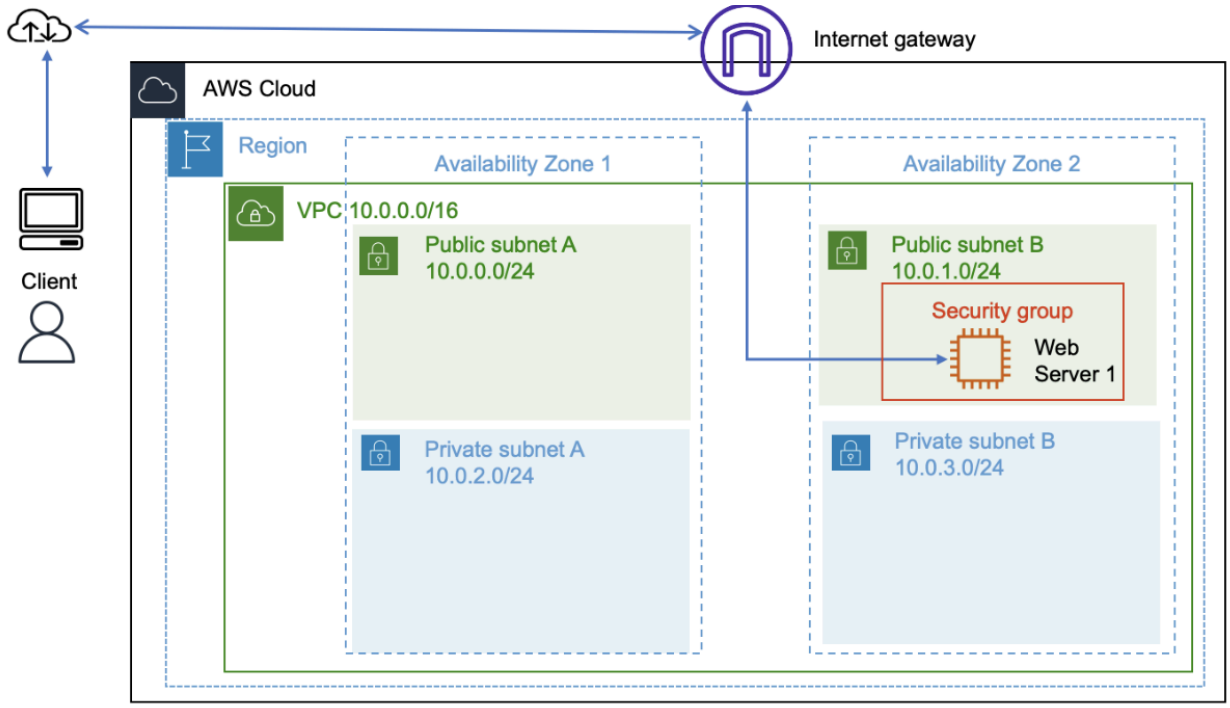
Lab overview

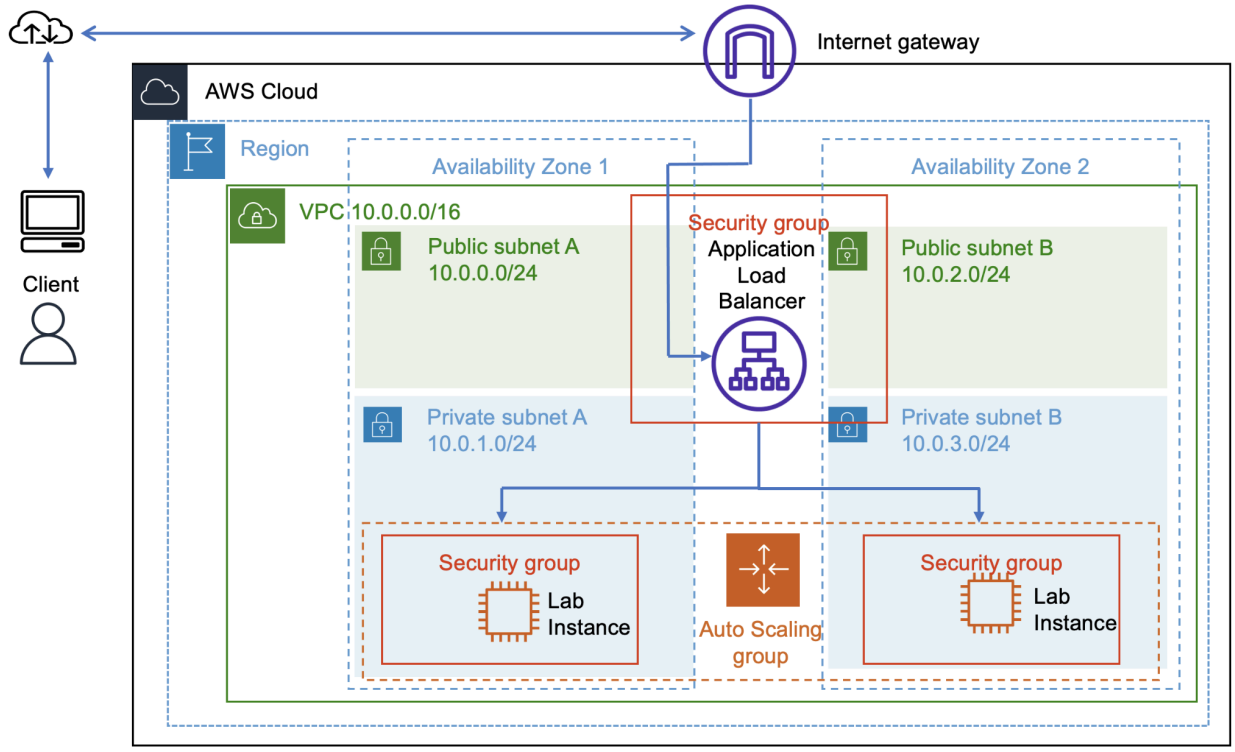
In this lab, you use the Elastic Load Balancing (ELB) and Amazon EC2 Auto Scaling to load balance and automatically scale your infrastructure.

ELB automatically distributes incoming application traffic across multiple Amazon Elastic Compute Cloud (Amazon EC2) instances. ELB provides the amount of load balancing capacity needed to route application traffic to help you achieve fault tolerance in your applications.

Auto Scaling helps you maintain application availability and gives you the ability to scale your Amazon EC2 capacity out or in automatically according to conditions that you define. You can use auto scaling to help ensure that you are running your desired number of EC2 instances. Auto scaling can also automatically increase the number of EC2 instances during spikes in demand to maintain performance and can decrease capacity during lulls to reduce costs. Auto scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

The following is the starting architecture:


Internet gateway
AWS Cloud
Region
Availability Zone 1
Availability Zone 2
VPC 10.0.0.0/16
Public subnet A
10.0.0.0/24
Public subnet B
10.0.1.0/24
Security group
Web
Server 1
Private subnet A
10.0.2.0/24
Private subnet B
10.0.3.0/24
Client


Internet gateway
AWS Cloud
Region
Availability Zone 1
Availability Zone 2
VPC 10.0.0.0/16
Security group
Application
Load
Balancer
Public subnet A
10.0.0.0/24
Public subnet B
10.0.2.0/24
Private subnet A
10.0.1.0/24
Private subnet B
10.0.3.0/24
Security group
Lab
Instance
Auto Scaling
group
Security group
Lab
Instance
Client

Objectives

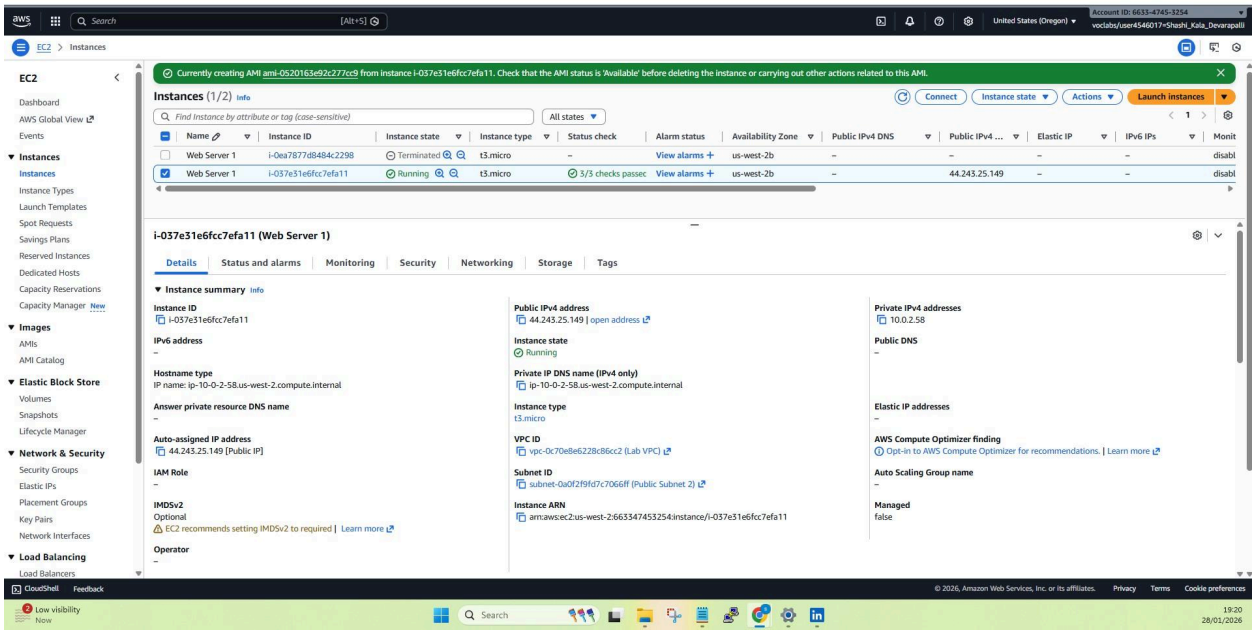
After completing this lab, you should be able to do the following:

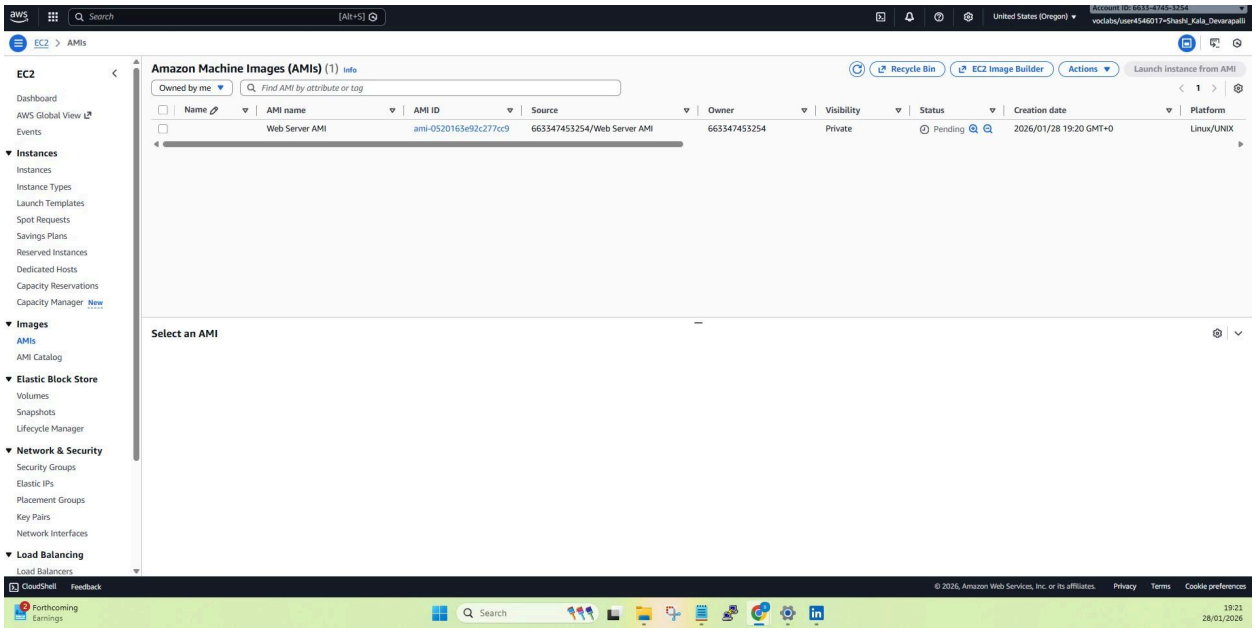
- Create an AMI from an EC2 instance.
- Create a load balancer.
- Create a launch template and an Auto Scaling group.
- Configure an Auto Scaling group to scale new instances within private subnets.
- Use Amazon CloudWatch alarms to monitor the performance of your infrastructure.

Task 1: Creating an AMI for auto scaling

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

5. On the **AWS Management Console**, in the **Search** bar, enter and choose `EC2` to open the **Amazon EC2 Management Console**.
6. In the left navigation pane, locate the **Instances** section, and choose **Instances**. The **Web Server 1** instance is listed. You now create an AMI based on this instance.
7. Choose the **Web Server 1** instance, which should appear in a Running state.
8. From the **Actions** dropdown list, choose **Image and templates** > **Create image**, and then configure the following options:
    - For **Image name**, enter `Web Server AMI`
    - For **Image description - *optional***, enter `Lab AMI for Web Server`
9. Choose **Create image**.
   The confirmation screen displays the AMI ID for your new AMI. You use this AMI when launching the Auto Scaling group later in the lab.





Task 2: Creating a load balancer

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

10. In the left navigation pane, locate the **Load Balancing** section, and choose **Load Balancers**.
11. Choose **Create load balancer**.

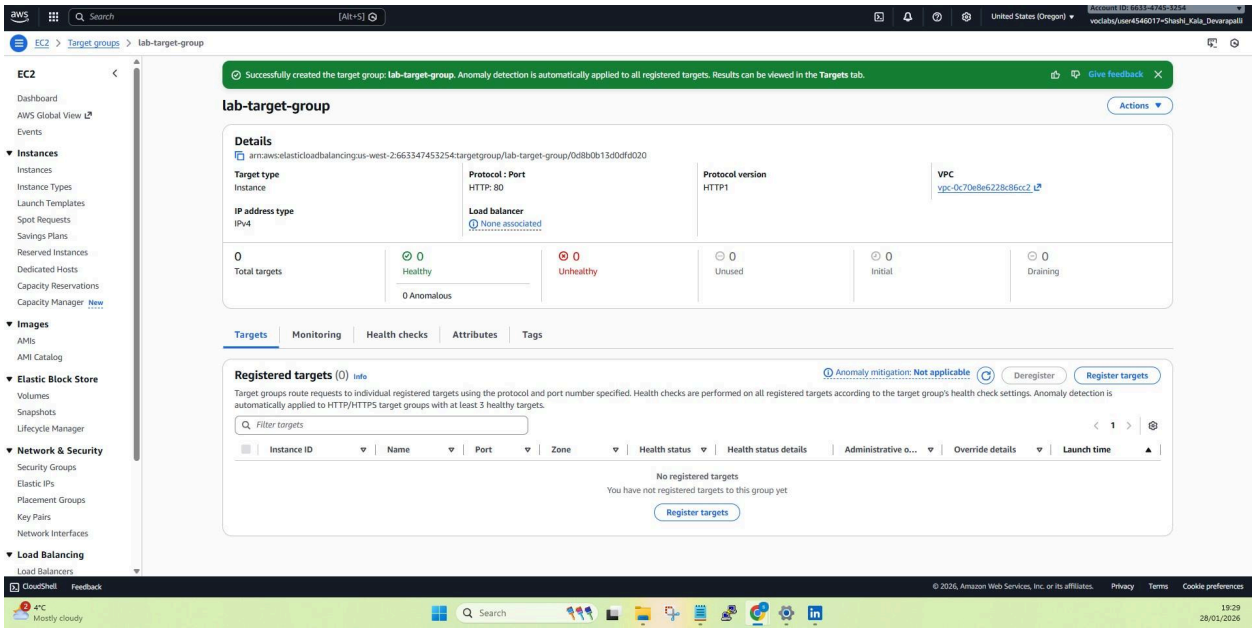
12. In the **Load balancer types** section, for **Application Load Balancer**, choose **Create**.
13. On the **Create Application Load Balancer** page, in the **Basic configuration** section, configure the following option:
    - For the **Load balancer name**, enter `LabELB`
14. In the **Network mapping** section, configure the following options:
    - For **VPC**, choose **Lab VPC**.
    - For **Mappings**, choose both Availability Zones listed.
    - For the first Availability Zone, choose **Public Subnet 1**.
    - For the second Availability Zone, choose **Public Subnet 2**.
15. These options configure the load balancer to operate across multiple Availability Zones.
16. In the **Security groups** section, choose the **X** for the **default** security group to remove it.
17. From the **Security groups** dropdown list, choose **Web Security Group**.
    The **Web Security Group** has already been created for you, which permits HTTP access.
18. In the **Listeners and routing** section, choose the **Create target group** link.
    Note: This link opens a new browser tab with the **Create target group** configuration options.
19. On the new **Target groups** browser tab, in the **Basic configuration** section, configure the following:
    - For **Choose a target type**, choose **Instances**.
    - For **Target group name**, enter `lab-target-group`
20. At the bottom of the page, choose **Next**.
21. On the **Register targets** page, choose **Create target group**.
    Once the target group has been created successfully, close the **Target groups** browser tab.
22. Return to the **Load balancers** browser tab. In the **Listeners and routing** section, choose **Refresh** to the right of the **Forward to** dropdown list for **Default action**.
23. From the **Forward to** dropdown list, choose **lab-target-group**.
24. At the bottom of the page, choose **Create load balancer**.

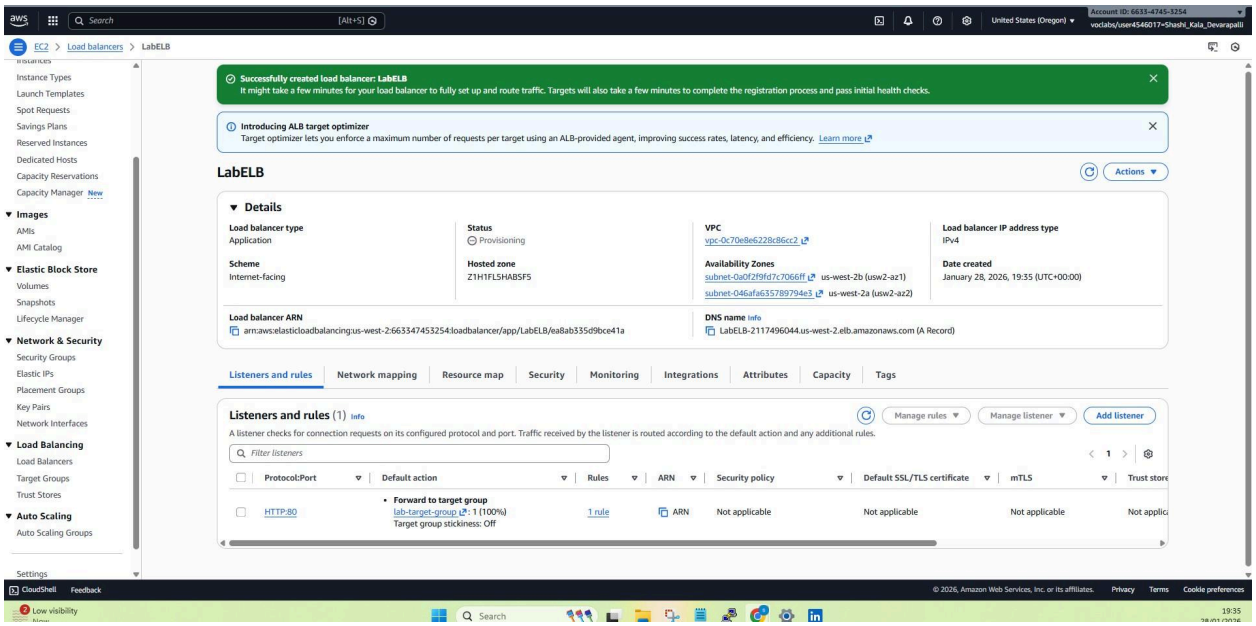
You should receive a message similar to the following:

Successfully created load balancer: LabELB

24. To view the **LabELB** load balancer that you created, choose **View load balancer**.

25. To copy the **DNS name** of the load balancer, use the copy option , and paste the DNS name into a text editor.





Task 3: Creating a launch template

In this task, you 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, instance type, key pair, security group, and disks.

26. At the top of the AWS Management Console, in the search bar, enter and choose `EC2`
27. In the left navigation pane, locate the **Instances** section, and choose **Launch Templates**.
28. Choose **Create launch template**.
29. On the **Create launch template** page, in the **Launch template name and description** section, configure the following options:
    - For **Launch template name - *required***, enter `lab-app-launch-template`
    - For **Template version description**, enter `A web server for the load test app`
    - For **Auto Scaling guidance**, choose **Provide guidance to help me set up a template that I can use with EC2 Auto Scaling**.
30. In the **Application and OS Images (Amazon Machine Image) - required** section, choose the **My AMIs** tab. Notice that **Web Server AMI** is already chosen.
31. In the **Instance type** section, choose the **Instance type** dropdown list, and choose **t3.micro**.
32. In the **Key pair (login)** section, confirm that the **Key pair name** dropdown list is set to **Don't include in launch template**.

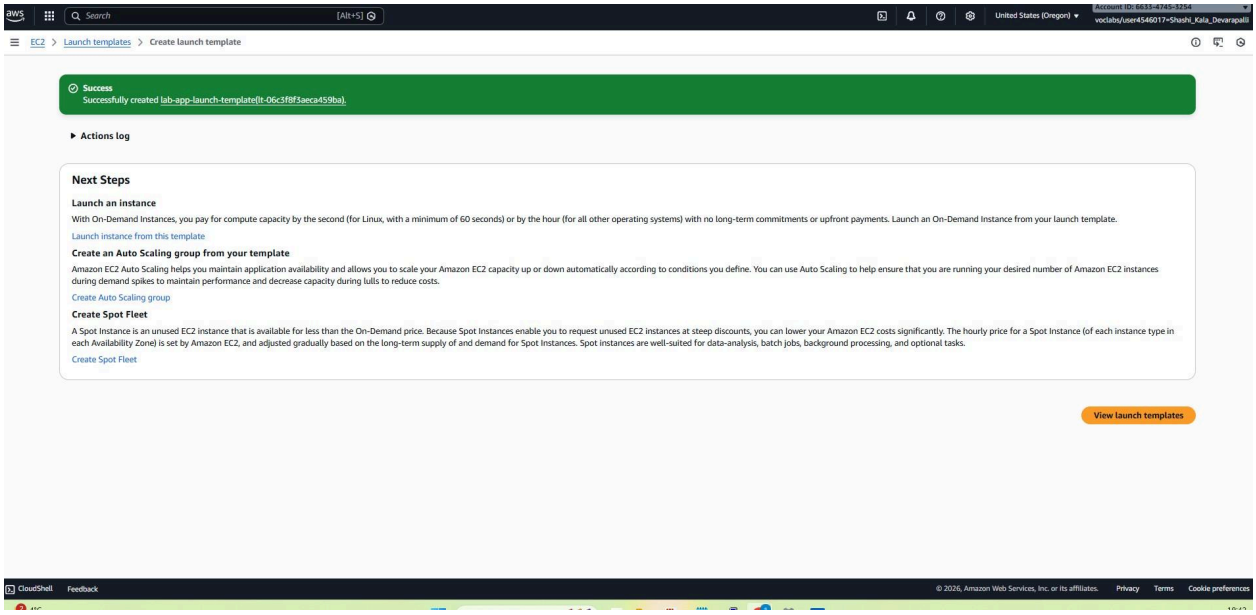
    Amazon EC2 uses public key cryptography to encrypt and decrypt login information. To log in to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

    Note: In this lab, you do not need to connect to the instance.
33. In the **Network settings** section, choose the **Security groups** dropdown list, and choose **Web Security Group**.

    When you launch an instance, you can pass user data to the instance. The data can be used to run configuration tasks and scripts.
34. Choose **Create launch template**.

    You should receive a message similar to the following:

    Successfully created lab-app-launch-template.
35. Choose **View launch templates**.



Task 4: Creating an Auto Scaling group

In this task, you use your launch template to create an Auto Scaling group.

36. Choose **lab-app-launch-template**, and then from the **Actions** dropdown list, choose **Create Auto Scaling group**
37. On the **Choose launch template or configuration** page, in the **Name** section, for **Auto Scaling group name**, enter `Lab Auto Scaling Group`
38. Choose **Next**.
39. On the **Choose instance launch options** page, in the **Network** section, configure the following options:
    - From the **VPC** dropdown list, choose **Lab VPC**.
    - From the **Availability Zones and subnets** dropdown list, choose **Private Subnet 1 (10.0.1.0/24)** and **Private Subnet 2 (10.0.3.0/24)**.
40. Choose **Next**.
41. On the **Configure advanced options – *optional*** page, configure the following options:
    - In the **Load balancing – *optional*** section, choose **Attach to an existing load balancer**.
    - In the **Attach to an existing load balancer** section, configure the following options:

        - Choose **Choose from your load balancer target groups**.
        - From the **Existing load balancer target groups** dropdown list, choose **lab-target-group | HTTP**.
    - In the **Health checks – *optional*** section, for **Health check type**, choose **ELB**.
42. Choose **Next**.
43. On the **Configure group size and scaling policies – *optional*** page, configure the following options:
    - In the **Group size – *optional*** section, enter the following values:
        - **Desired capacity**:`2`
        - **Minimum capacity**: `2`
        - **Maximum capacity**: `4`
    - In the **Scaling policies – *optional*** section, configure the following options:
        - Choose **Target tracking scaling policy**.
        - For **Metric type**, choose **Average CPU utilization**.
        - Change the **Target value** to `50`
    - This change tells Auto Scaling to maintain an average CPU utilization across all instances of 50 percent. Auto Scaling automatically adds or removes 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.
44. Choose **Next**.
45. On the **Add notifications – *optional*** page, choose **Next**.
46. On the **Add tags – *optional*** page, choose **Add tag** and configure the following options:
    - **Key:** Enter `Name`
    - **Value - optional:** Enter `Lab Instance`
47. Choose **Next**.
48. Choose **Create Auto Scaling group**.

    These options launch EC2 instances in private subnets across both Availability Zones.

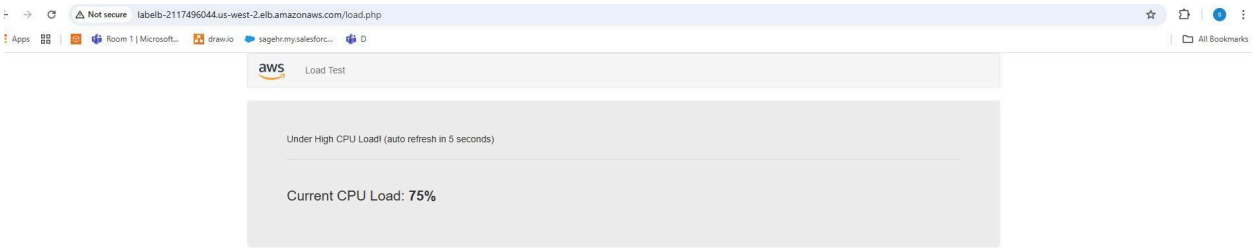
    Your Auto Scaling group initially shows an **Instances** count of zero, but new instances will be launched to reach the desired count of two instances.

    Note: If you experience an error related to the t3.micro instance type not being available, then rerun this task by choosing the t2.micro instance type instead.

Task 5: Verifying that load balancing is working

In this task, you verify that load balancing is working correctly.

49. In the left navigation pane, locate the **Instances** section, and choose **Instances**.
    You should see two new instances named **Lab Instance**. These instances were launched by auto scaling. If the instances or names are not displayed, wait 30 seconds, and then choose refresh .
    First, you confirm that the new instances have passed their health check.
50. In the left navigation pane, in the **Load Balancing** section, choose **Target Groups**.
51. Choose **lab-target-group**.
    In the **Registered targets** section, two **Lab Instance** targets should be listed for this target group.
52. Wait until the **Health status** of both instances changes to *healthy*. To check for updates, choose refresh .
    A *healthy* status indicates that an instance has passed the load balancer's health check. This check means that the load balancer will send traffic to the instance. You can now access the instances launched in the Auto Scaling group using the load balancer.
53. Open a new web browser tab, paste the DNS name that you copied before, and press Enter.
    The **Load Test** application should appear in your browser, which means that the load balancer received the request, sent it to one of the EC2 instances, and then passed back the result.



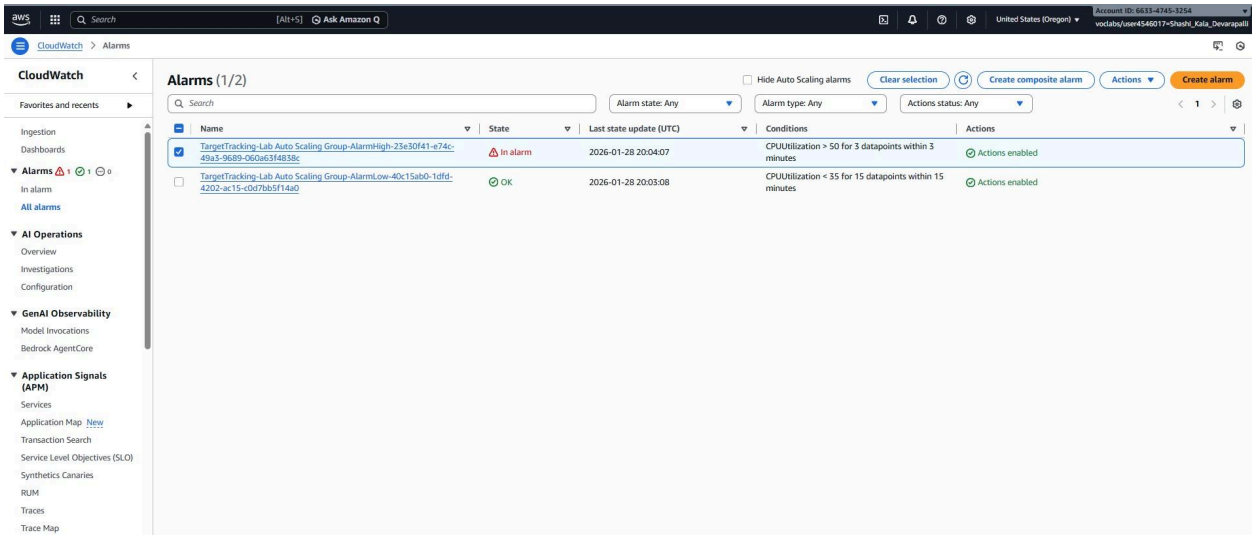
Task 6: Testing auto scaling

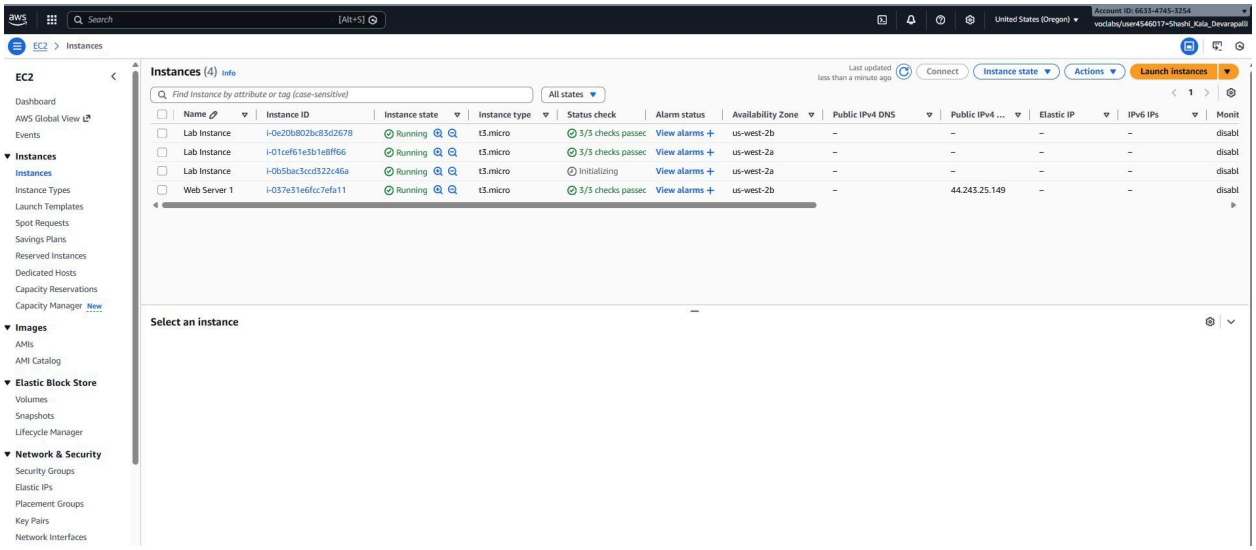
You created an Auto Scaling group with a minimum of two instances and a maximum of four instances. Currently, two instances are running because the minimum size is two and the group is currently not under any load. You now increase the load to cause auto scaling to add additional instances.

56. Return to the AWS Management Console, but keep the **Load Test** application tab open. You return to this tab soon.
57. In the AWS Management Console, in the search bar, enter and choose `CloudWatch`
58. In the left navigation pane, in the **Alarms** section, choose **All alarms**.
    Two alarms are displayed. The Auto Scaling group automatically created these two alarms. These alarms automatically keep the average CPU load close to 50 percent while also staying within the limitation of having 2–4 instances.
59. Choose the alarm that has **AlarmHigh** in its name. This alarm should have a **State** of *OK*.
    If the alarm is not showing *OK* for the **State**, wait a minute and then choose refresh until the **State** changes.
    The *OK* state indicates that the alarm has not been initiated. It is the alarm for **CPU Utilization > 50**, which adds instances when the average CPU utilization is high. The chart should show very low levels of CPU at the moment.
    You now tell the application to perform calculations that should raise the CPU level.
60. Return to the browser tab with the **Load Test** application.
61. Next to the AWS logo, choose **Load Test**.
    This step causes the application to generate high loads. The browser page

automatically refreshes so that all instances in the Auto Scaling group will generate loads. Do not close this tab.

62. Return to browser tab with the **CloudWatch Management Console**.
    In less than 5 minutes, the **AlarmLow** alarm status should change to *OK*, and the **AlarmHigh** alarm status should change to *In alarm*.
    To update the display, choose refresh every 60 seconds.
    You should see the **AlarmHigh** chart indicating an increasing CPU percentage. Once it crosses the 50 percent line for more than 3 minutes, it initiates auto scaling to add additional instances.
63. Wait until the **AlarmHigh** alarm enters the *In alarm* state.
    You can now view the additional instance or instances that were launched.
64. In the AWS Management Console, in the search bar, enter and choose `EC2`
65. In the left navigation pane, locate the **Instances** section, and choose **Instances**.
    More than two instances named **Lab Instance** should now be running. Auto scaling created the new instances in response to the alarm.

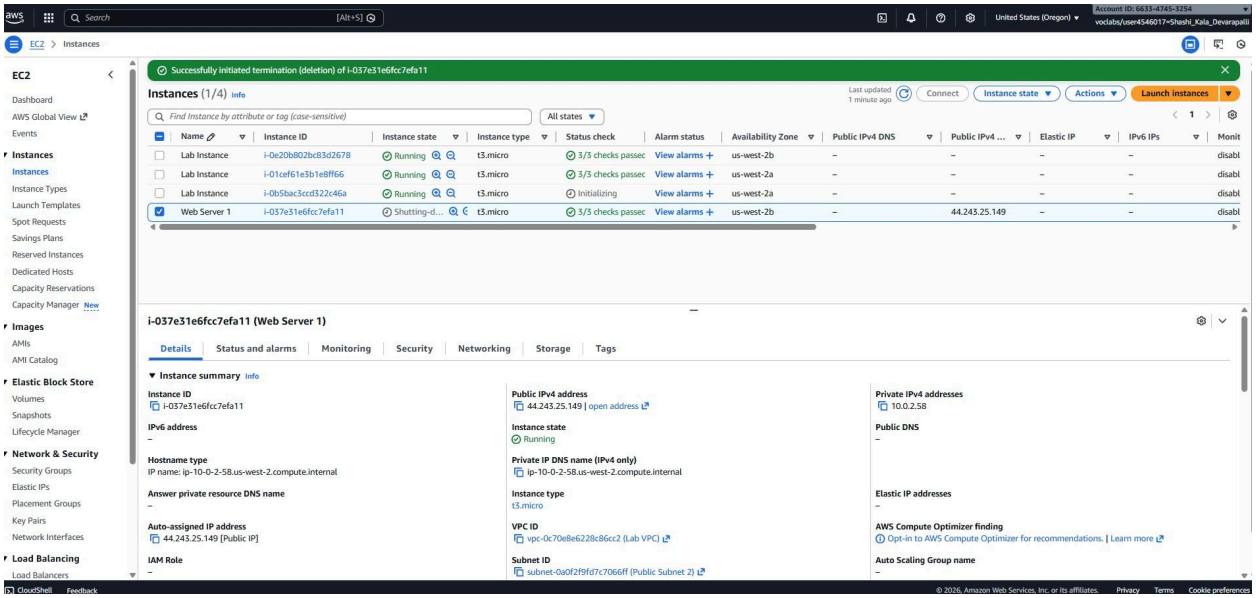




Task 7: Terminating the Web Server 1 instance

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

66. Choose **Web Server 1**, and ensure that it is the only instance selected.
67. From the **Instance state** dropdown menu, choose **Terminate instance**.
68. Choose **Terminate**.


Search
[Alt+S]
United States (Oregon)
Account ID: 6633-4745-3254
voclabs/user4546017=Shashi_Kala_Devarapalli
EC2 > Instances
Successfully initiated termination (deletion) of i-037e31e6fcc7efa11
Instances (1/4) Info
Last updated 1 minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IPv4 DNS
Public IPv4 ...
Elastic IP
IPv6 IPs
Monit
Lab Instance i-0e20b802bc83d2678 Running t3.micro 3/3 checks passed View alarms us-west-2b
Lab Instance i-01cef61e3b1e8ff66 Running t3.micro 3/3 checks passed View alarms us-west-2a
Lab Instance i-0b5bac3ccd322c46a Running t3.micro Initializing View alarms us-west-2a
Web Server 1 i-037e31e6fcc7efa11 Shutting-d... t3.micro 3/3 checks passed View alarms us-west-2b 44.243.25.149
EC2
Dashboard
AWS Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Capacity Manager New
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
i-037e31e6fcc7efa11 (Web Server 1)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-037e31e6fcc7efa11
IPv6 address
–
Hostname type
IP name: ip-10-0-2-58.us-west-2.compute.internal
Answer private resource DNS name
–
Auto-assigned IP address
44.243.25.149 [Public IP]
IAM Role
–
Public IPv4 address
44.243.25.149 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-10-0-2-58.us-west-2.compute.internal
Instance type
t3.micro
VPC ID
vpc-0c70e8e6228c86cc2 (Lab VPC)
Subnet ID
subnet-0a0f2f9fd7c7066ff (Public Subnet 2)
Private IPv4 addresses
10.0.2.58
Public DNS
–
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
CloudShell
Feedback
© 2026, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences