AWS re:Start LAB

Scaling and Load Balancing

WEEK 9

aws re/start

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.

Topics covered

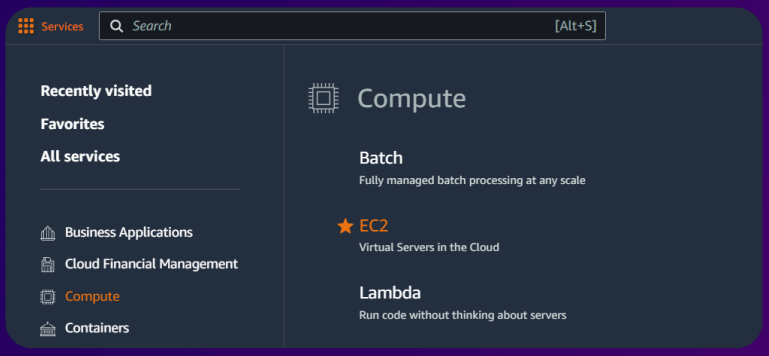
- Create an AMI from an EC2 instance.
- Create a load balancer.
- Create a launch template and an Auto Scaling group.
- Configure an Auto Scaling group to scale new instances within private subnets.
- Use Amazon CloudWatch alarms to monitor the performance of your infrastructure.

Task 1

Creating an AMI for auto scaling

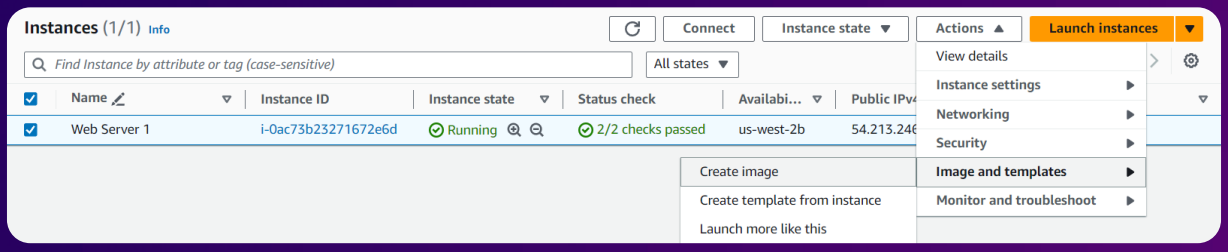
Step 1: Access the EC2 Management Console

Open the AWS Management Console, and select EC2.



Step 2: Create image

Navigate to the **Instances** section, select the **Web Server 1** instance, and choose Actions > Image and templates > Create image to create an AMI based on this instance.



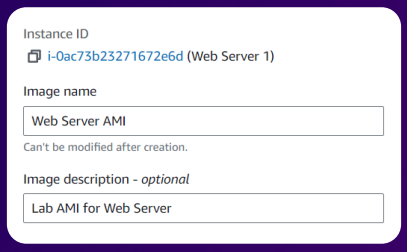
aws re/start

Task 1

Creating an AMI for auto scaling

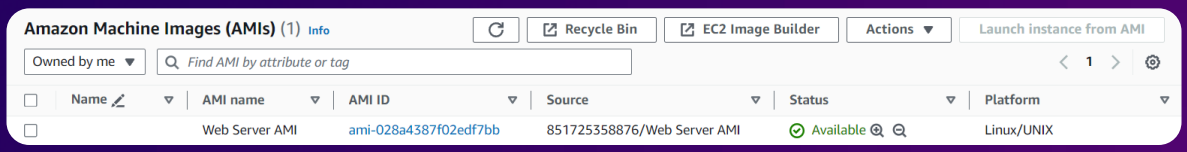
Step 3: Configure the image

In the **Create image** section, configure the following settings.



Step 4: Review image

Navigate to the **AMIs** section, and review the newly created Amazon Machine Image Web Server AMI.



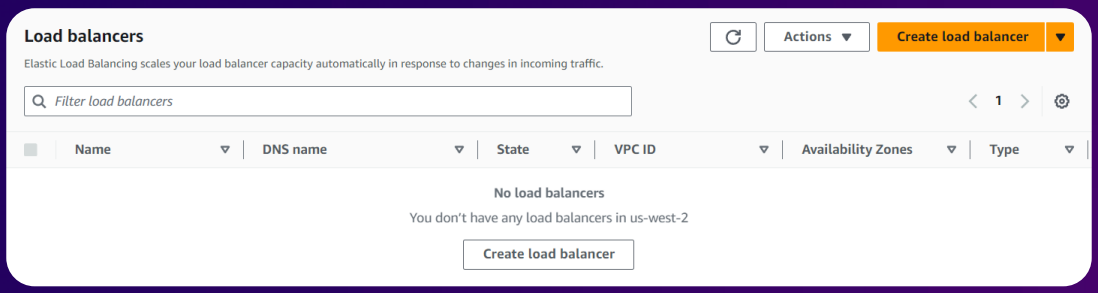
aws re/start

Task 2

Creating a load balancer

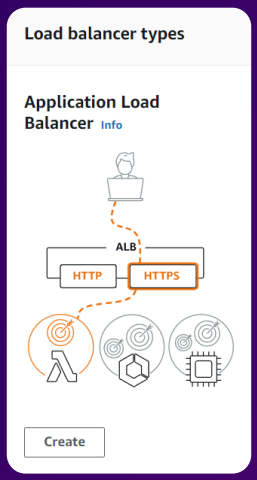
Step 1: Create load balancer

Navigate to the **Load balancers** section, and select Create load balancer.



Step 2: Load balancer types

In the **Load balancer types** section, select **Application Load Balancer**, and choose Create.



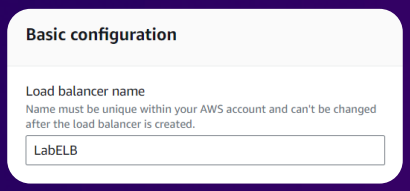
aws re/start

Task 2

Creating a load balancer

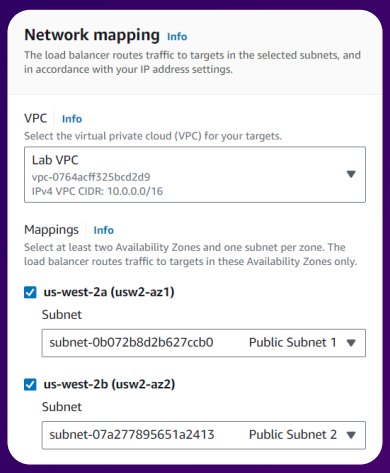
Step 3: Basic configuration

In the **Basic configuration** section, for **Load balancer name**, enter LabELB.



Step 4: Network mapping

In the **Network mapping** section, configure the following settings.



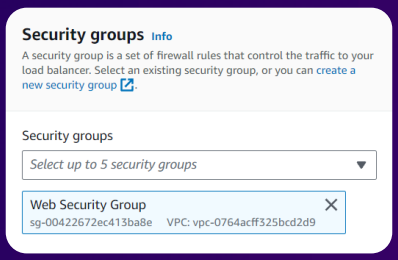
aws re/start

Task 2

Creating a load balancer

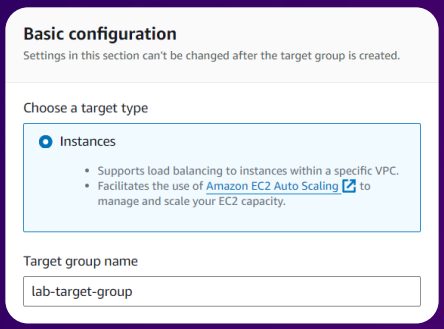
Step 5: Security groups

In the **Security groups** section, select the Web Security Group, which permits HTTP access.



Step 6: Create target group

In the **Listeners and routing** section, choose the Create target group link. On the new **Target groups** tab, configure the following settings, and choose Create target group.

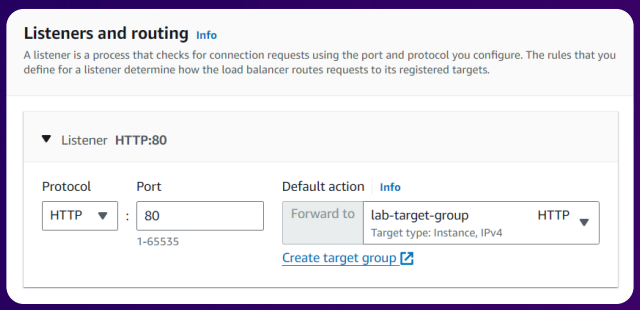


Task 2

Creating a load balancer

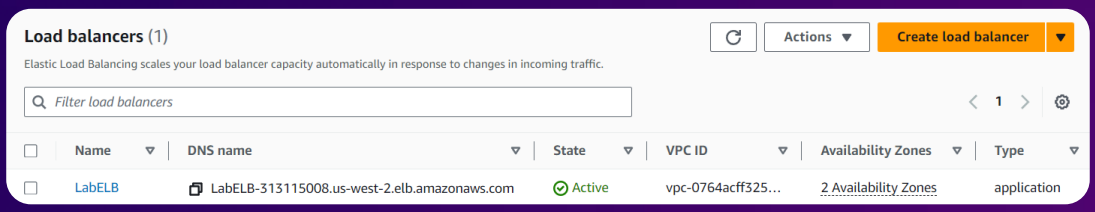
Step 7: Listeners and routing

In the **Listeners and routing** section, for the **Listener** tab, in the **Forward to** dropdown list, choose the lab-target-group.



Step 8: Review load balancer creation

Navigate to the **Load balancers** section, and review the newly created load balancer **LabELB**, and copy the DNS name.



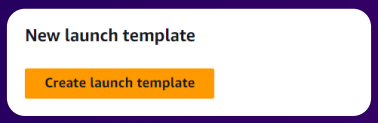
aws re/start

Task 3

Creating a launch template

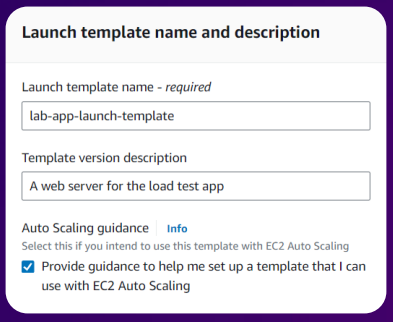
Step 1: Create launch template

Navigate to the **Launch Templates** section, and select Create launch template.



Step 2: Launch template name and description

In the **Launch template name and description** section, configure the following settings.

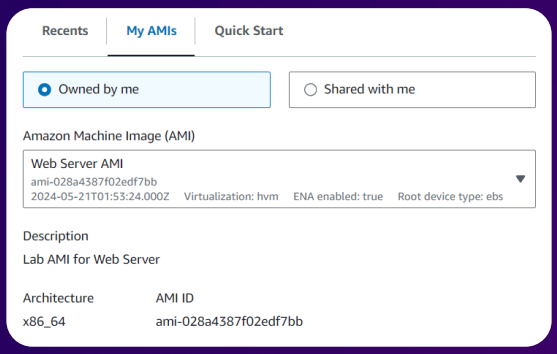


Task 3

Creating a launch template

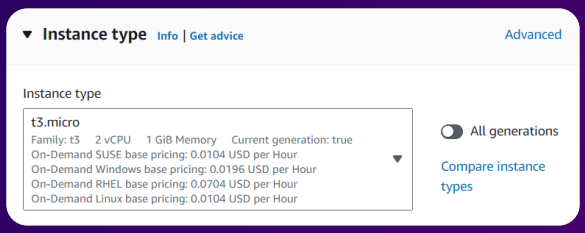
Step 3: Application and OS Images

In the **Application and OS Images (Amazon Machine Image)** section, choose the **My AMIs** tab, and select the Web Server AMI.



Step 4: Instance type

In the **Instance type** section, choose t3.micro.



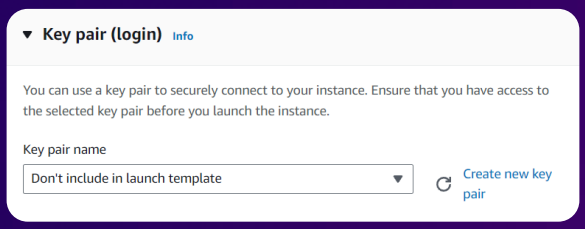
aws re/start

Task 3

Creating a launch template

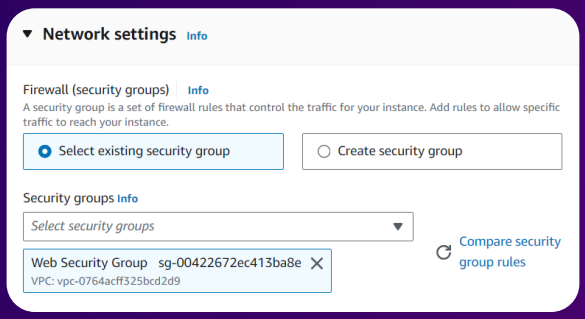
Step 5: Key pair (login)

In the **Key pair (login)** section, for **Key pair name**, select Don't include in launch template.



Step 6: Network settings

In the **Network settings** section, for **Security groups**, select the Web Security Group.



aws re/start

Task 4

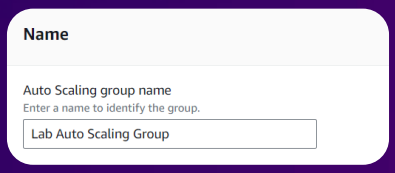
Creating an Auto Scaling group

Step 1: Create Auto Scaling group

In the **Launch Templates** section, choose the newly created **lab-app-launch-template** launch template, and from the Actions dropdown list, select Create Auto Scaling group.

Step 2: Name

In the **Name** section, for **Auto Scaling group name**, enter Lab Auto Scaling Group.

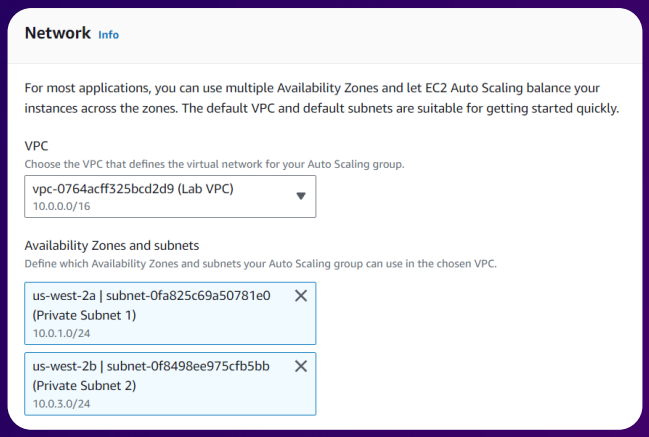


Task 4

Creating an Auto Scaling group

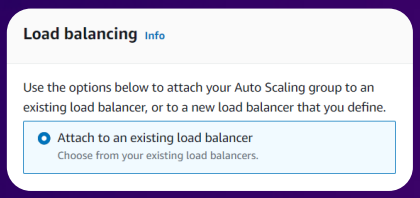
Step 3: Network

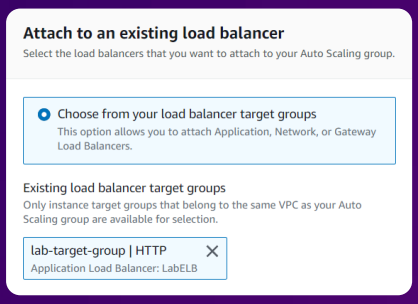
In the **Network** section, configure the following settings.



Step 4: Configure advanced options

For the **Load balancing** and **Attach to an existing load balancer** sections, configure the following settings.



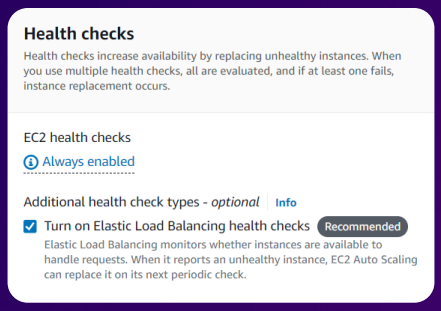


Task 4

Creating an Auto Scaling group

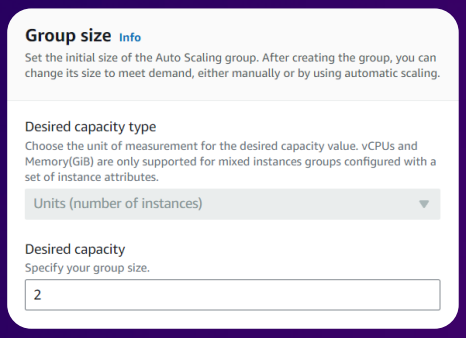
Step 5: Health checks

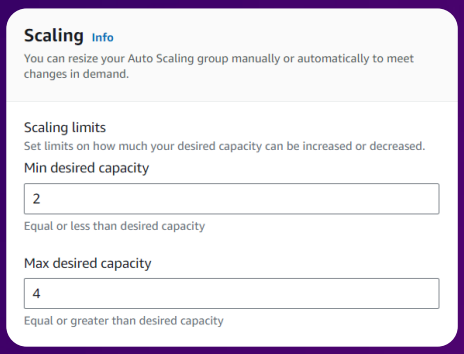
In the **Health checks** section, for **Additional health check types**, select Turn on Elastic Load Balancing health checks.



Step 6: Configure group size and scaling

For the **Group size** and **Scaling** sections, configure the following settings.





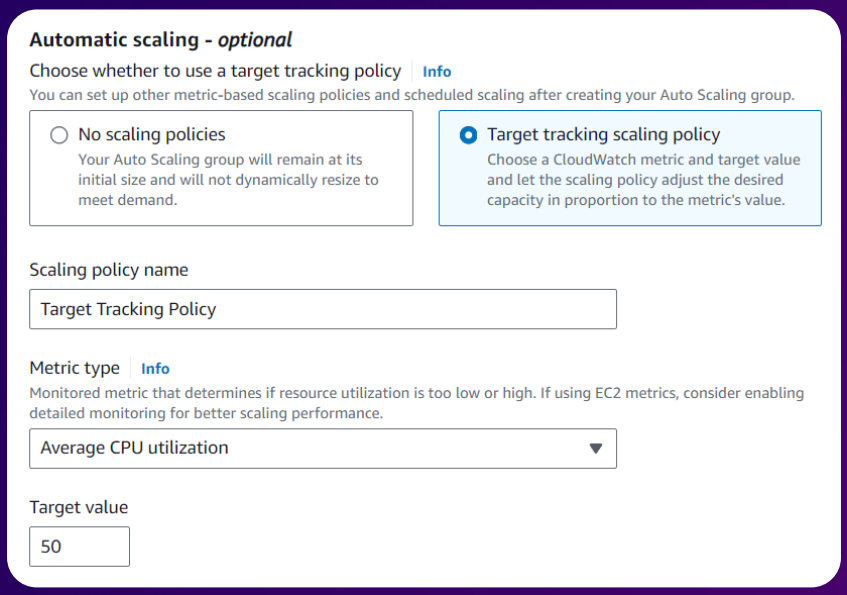
aws re/start

Task 4

Creating an Auto Scaling group

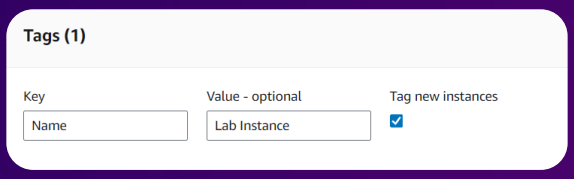
Step 7: Automatic scaling

In the **Automatic scaling** section, configure the following settings.



Step 8: Tags

In the **Add tags** section, configure the following Tags settings.



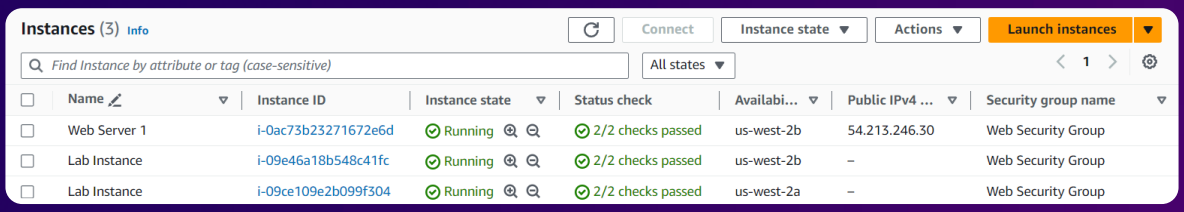
aws re/start

Task 5

Verifying that load balancing is working

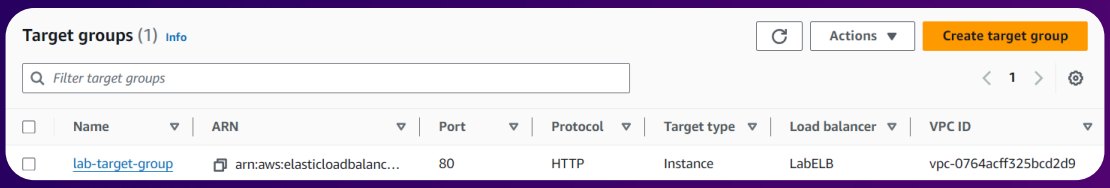
Step 1: Review Auto Scaling

Navigate to the **Instances** section, you should see two new instances named **Lab Instance**. These instances were launched by auto scaling.



Step 2: Review Target Groups

Navigate to the **Target Groups** section, and select the lab-target-group target group.



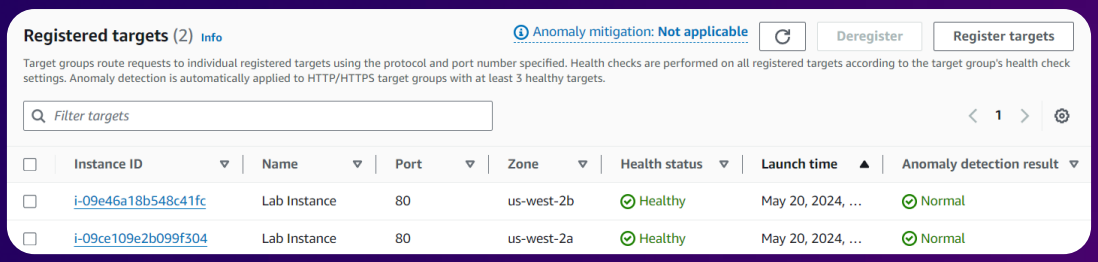
aws re/start

Task 5

Verifying that load balancing is working

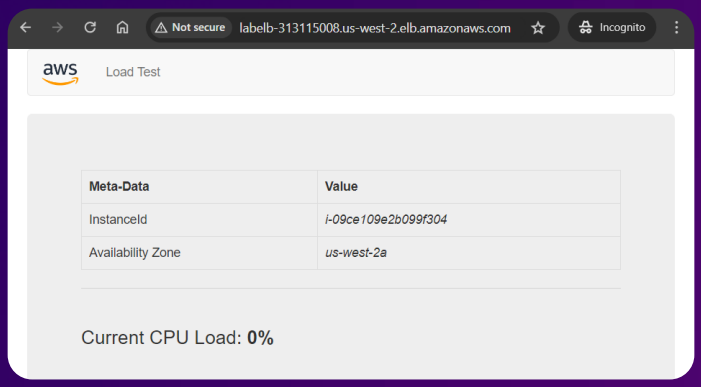
Step 3: Review Health status

In the **Registered targets** section, two Lab Instance targets should be listed for this target group. Review the Healthy **Health status** of both instances.



Step 4: Review Load Balancing

Access the instances launched in the Auto Scaling group using the load balancer DNS name. 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.



aws re/start

Task 6

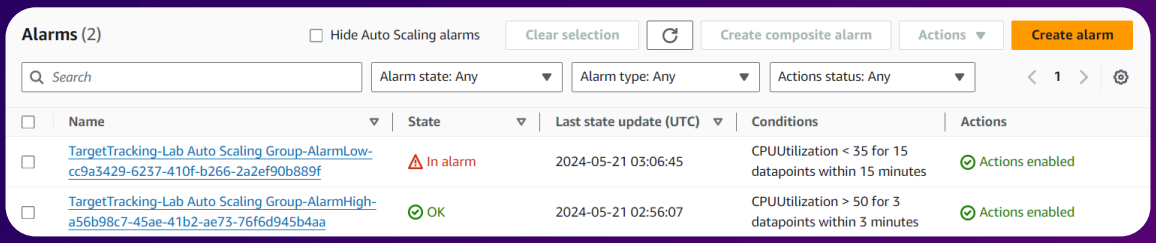
Testing auto scaling

Step 1: Access the CloudWatch Console

In the AWS Management Console, select CloudWatch.

Step 2: Review CloudWatch alarms

Navigate to the **All alarms** section. 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. The **AlarmHigh** alarm should have a **State** of OK.



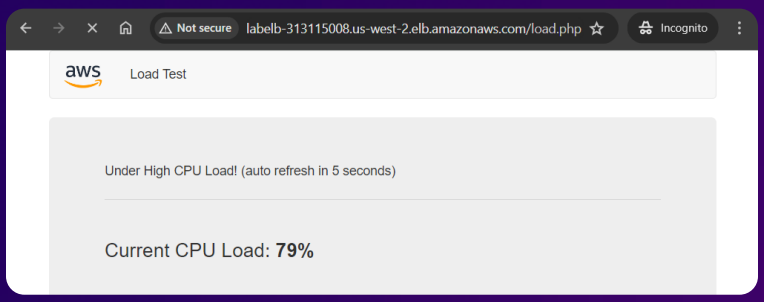
aws re/start

Task 6

Testing auto scaling

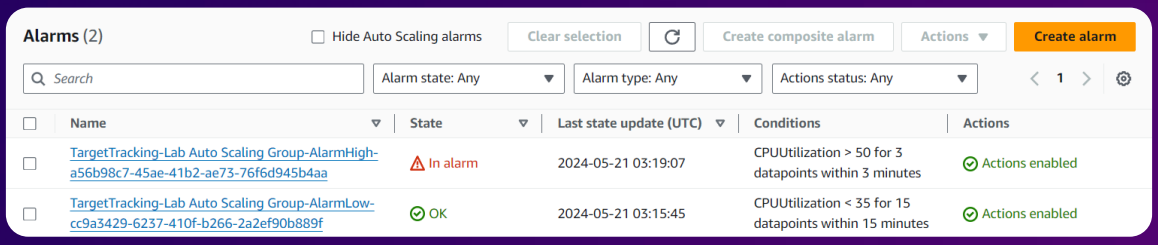
Step 3: Load Test

In the Load Test application, choose Load Test, this causes the application to generate high loads that raise the CPU level.



Step 4: Review the In alarm State

The **AlarmLow** alarm status should change to OK, and the **AlarmHigh** alarm status should change to In alarm.



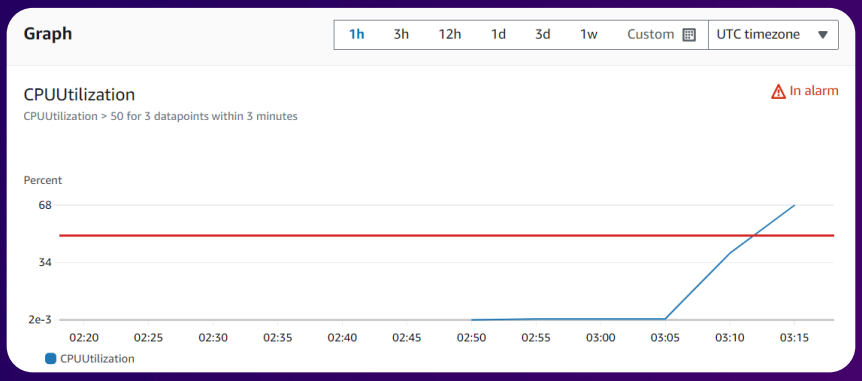
aws re/start

Task 6

Testing auto scaling

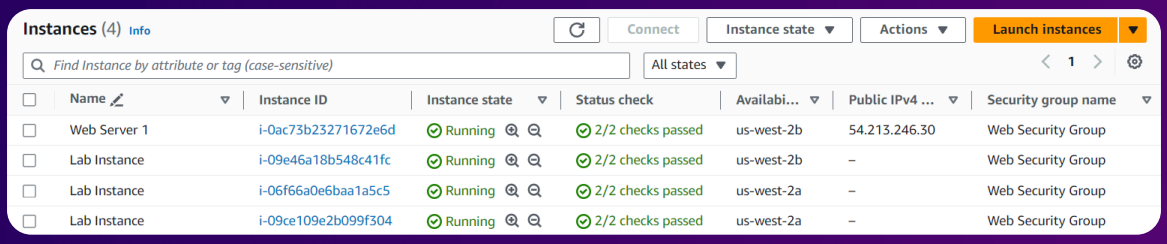
Step 5: Review the CPUUtilization Graph

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.



Step 6: Review alarm response

Navigate to the **Instances** section, more than two instances named **Lab Instance** should now be running. Auto scaling created the new instances in response to the alarm.



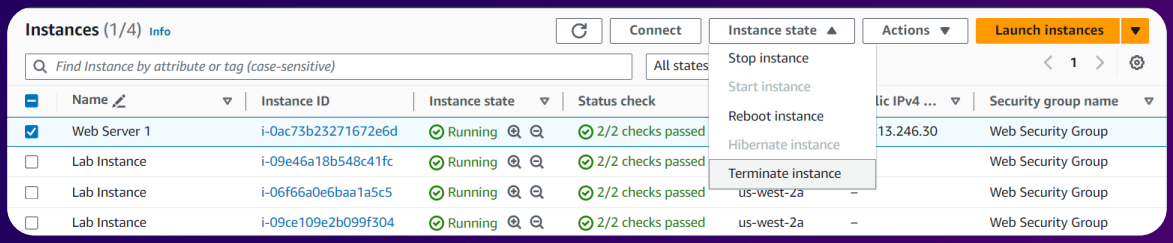
aws re/start

Task 7

Terminating the Web Server 1 instance

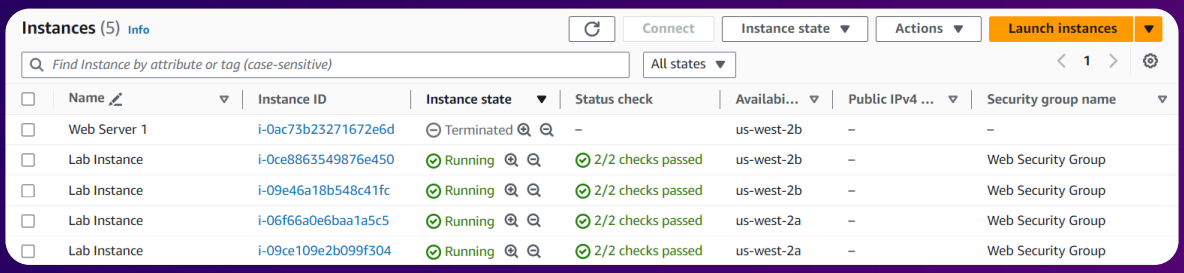
Step 1: Terminate instance

In the **Instances** section, terminate the **Web Server 1** instance.



Step 2: Review instance termination

Review the instance termination. Notice how, while the **Web Server 1** instance was being terminated, another **Lab Instance** was launched by the Auto Scaling Group in response to the alarm.



aws re/start

Conclusions

Application Load Balancers
Application Load Balancers operate at the application layer (HTTP/HTTPS), offering advanced routing and load balancing features.

Listeners
Listeners in load balancers check for connection requests using the configured protocol and port, directing traffic to the appropriate target group.

Target Groups
Target Groups define the destinations for traffic routed by load balancers, grouping EC2 instances or other resources for load distribution.

Launch templates
Launch templates standardize the configuration of EC2 instances, simplifying the process of launching and managing instances consistently.

Auto Scaling Groups
Auto Scaling Groups automatically adjust the number of EC2 instances based on demand, ensuring optimal performance and cost efficiency.

aws re/start

Cristhian Becerra

cristhian-becerra-espinoza

+51 951 634 354

cristhianbecerra99@gmail.com

Lima, Peru