WHIZLABS S

Home / AWS / Guided Lab / Using AWS S3 to Store ELB Access Logs

Using AWS S3 to Store ELB Access Logs

Level: **Advanced**

Amazon EC2 Amazon S3 Amazon Web Services Elastic Load Balancing

Lab Overview Lab Steps

Cloud Developer, Cloud Administrator

Storage, Compute

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the Open Console button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,
    - Leave the Account ID as default. Never edit/remove the 12 digit Account ID

0h 56m 9s left

End Lab

Open Console

Validation

Lab Credentials —

User Name

Whiz_User_75551.91152521

Password

79426887-444f-4678-9c6e

Access Key

AKIARVNPBAEXNSFK2DGK

Confidentialité - Conditions

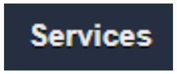
present in the AWS Console. otherwise, you cannot proceed with the lab.

- Now copy your **User Name** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign in** button.

3. Once Signed In to the AWS Management Console, Make the default AWS Region as **US East (N. Virginia) us-east-1.**

Note : If you face any issues, please go through FAQs and Troubleshooting for Labs.

Task 2: Launching two web servers with apache service installed

1. Make sure you are in the **US East (N. Virginia)** Region.
2. Navigate to  menu in the top, then click on **EC2** in the **Compute** section.
3. Click on **Instances** from the left side bar and then click on
4. Name : Enter ***webserver-A***
5. **For Amazon Machine Image (AMI): Search for Amazon Linux 2 AMI in the search box and click on the select button.**

Secret Key ⓘ

cYrpIVKcb/Rw4PLi4gg6v22

Support Documents +

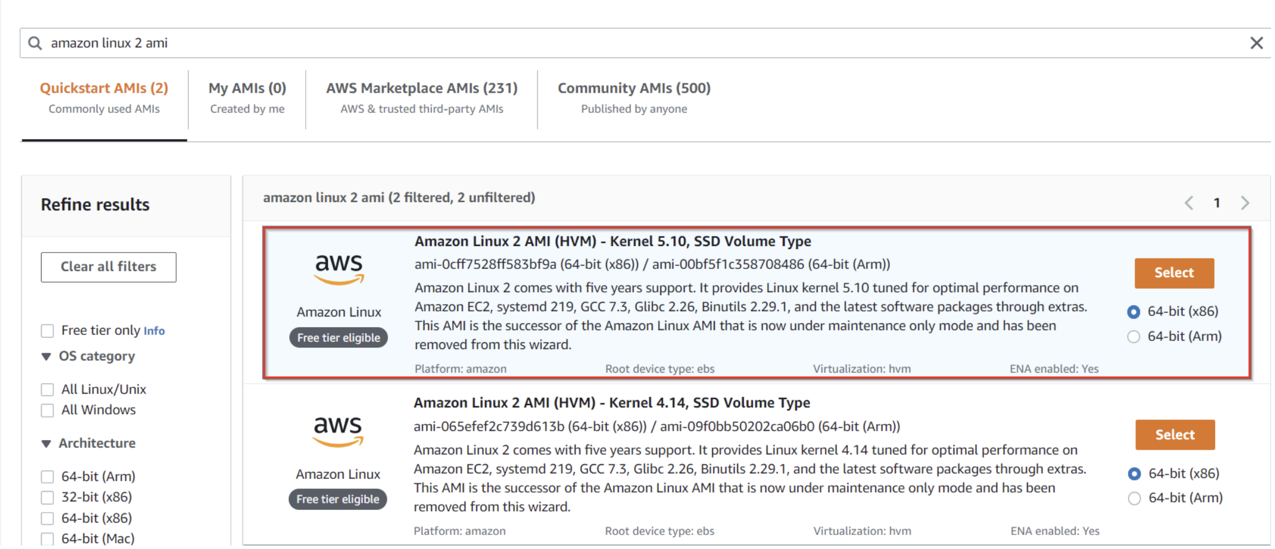
Need help?

How to use Hands on Lab

Troubleshooting Lab

FAQs

Submit Feedback | Share



amazon linux 2 ami

Quickstart AMIs (2) Commonly used AMIs | My AMIs (0) Created by me | AWS Marketplace AMIs (231) AWS & trusted third-party AMIs | Community AMIs (500) Published by anyone

Refine results

Clear all filters

□ Free tier only Info

▼ OS category

□ All Linux/Unix
□ All Windows

▼ Architecture

□ 64-bit (Arm)
□ 32-bit (x86)
□ 64-bit (x86)
□ 64-bit (Mac)

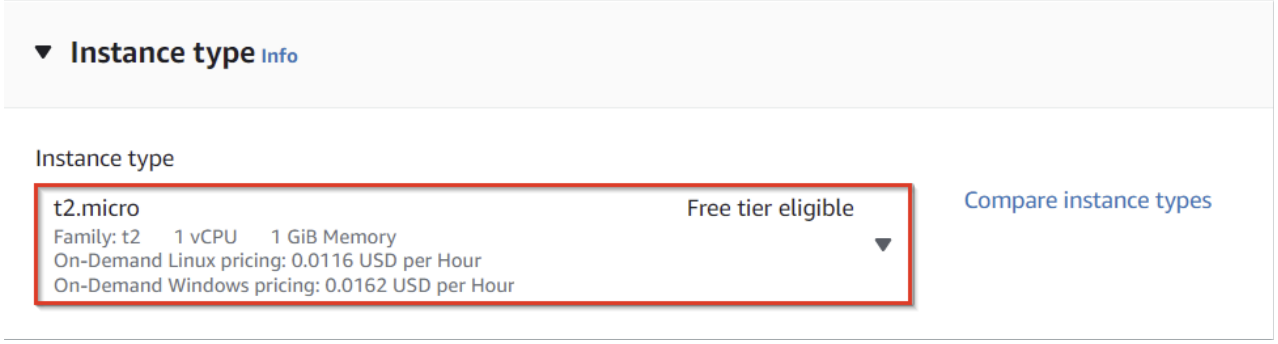
amazon linux 2 ami (2 filtered, 2 unfiltered)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes
Amazon Linux — Free tier eligible
Select
● 64-bit (x86)
○ 64-bit (Arm)

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type
ami-065efef2c739d613b (64-bit (x86)) / ami-09f0bb50202ca06b0 (64-bit (Arm))
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes
Amazon Linux — Free tier eligible
Select
● 64-bit (x86)
○ 64-bit (Arm)

Note: if there are two AMI's present for Amazon Linux 2 AMI, choose any of them.

6. **For Instance Type:** select ***t2.micro***



▼ Instance type Info

Instance type

t2.micro — Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

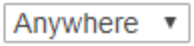
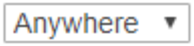
Compare instance types

7. **For Key pair:** Select **Create a new key pair** Button

    1. Key pair name: **WhizKey**
    2. Key pair type: **RSA**

   3. Private key file format: **.pem**

8. Select **Create key pair** Button.
9. In Network Settings Click on **Edit**:
   1. Auto-assign public IP: **Enable**
   2. Select **Create new Security group**
   3. Security group name : Enter ***webserver-SG***
   4. Check Allow SSH from and Select Anywhere from dropdown

- To add **SSH**,
  - Choose Type: SSH ▾
  - Source: Select  Anywhere ▾
- For **HTTP,** Click on **Add security group rule** button
  - Choose Type: **HTTP**
  - Source: Select  Anywhere ▾

10. Click on ▾ Advanced Details and under the **User data:** section, enter the following script

```
#!/bin/bash
```

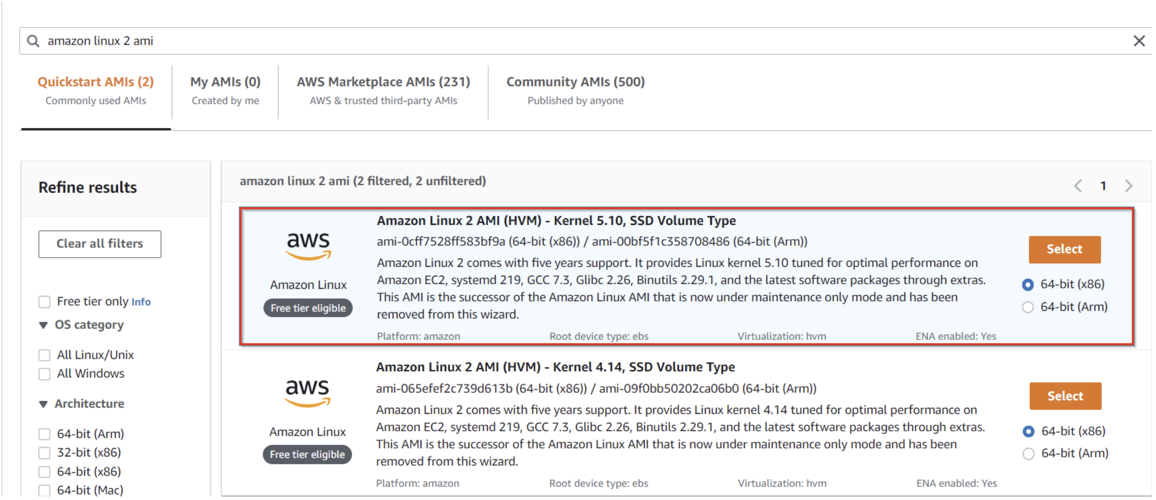
Copy

```
sudo su
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Response coming from server A" > /var/www/html/index.html
```

Note : The above script creates an HTML page served by Apache HTTP Server

11. Keep Rest thing Default and Click on **Launch Instance** Button.
12. Select **View all Instances** to View Instance you Created
13. **Launch Status:** Your instances are now launching, Navigate to **Instances** page from left menu and wait the status of the EC2 Instance changes to running
14. After a few minutes, you will see a new instance named **webserver-A** running.
15. Repeat the above steps for creating **webserver-B**.
16. Click on

17. Name : Enter ***webserver-B***
18. **For Amazon Machine Image (AMI):** Search for **Amazon Linux 2 AMI** in the search box and click on the **select** button.

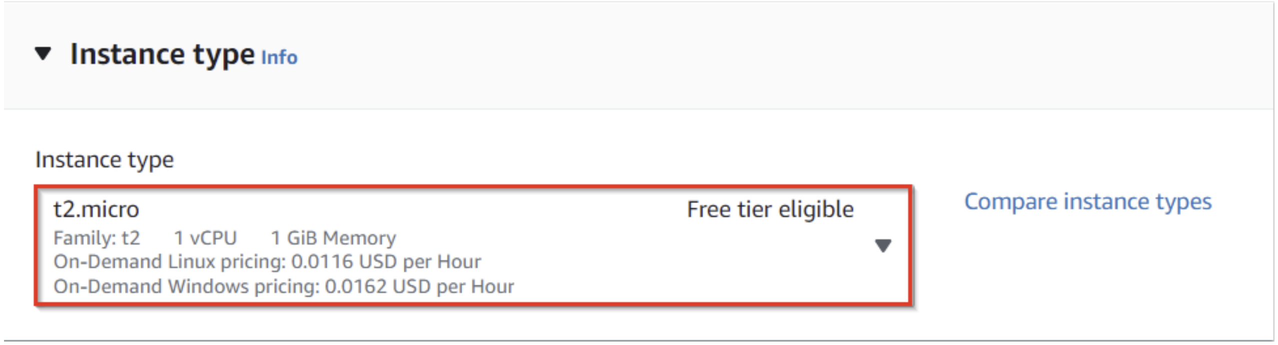


amazon linux 2 ami (2 filtered, 2 unfiltered) 1

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Amazon Linux 2 comes with five years support. It provides Linux kernel 5.10 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes
● 64-bit (x86)
○ 64-bit (Arm)

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type
ami-065efef2c739d613b (64-bit (x86)) / ami-09f0bb50202ca06b0 (64-bit (Arm))
Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras. This AMI is the successor of the Amazon Linux AMI that is now under maintenance only mode and has been removed from this wizard.
Platform: amazon Root device type: ebs Virtualization: hvm ENA enabled: Yes
● 64-bit (x86)
○ 64-bit (Arm)

19. **For Instance Type:** select ***t2.micro***



Instance type
t2.micro Free tier eligible
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

20. **For Key pair: Select an existing key pair**

21. In Network Settings Click on **Edit:**

    1. Auto-assign public IP: **Enable**

    2. **Select Existing Security group and Select webserver-SG**

22. Click on ▼ Advanced Details and under the User data: section, enter the following script

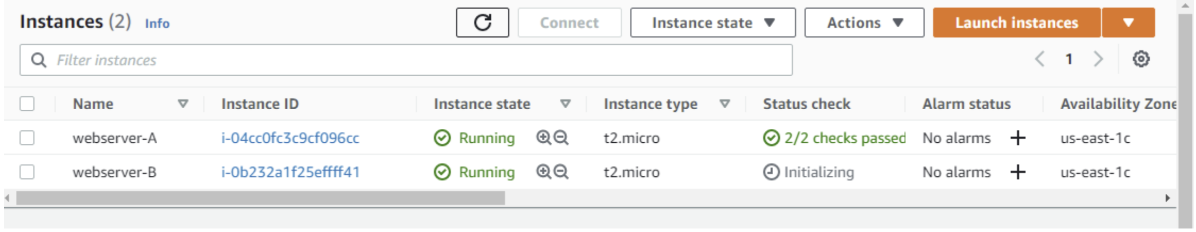
```
#!/bin/bash
sudo su
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "Response coming from server B" > /var/www/html/index.html
```

Copy

Note : The above script creates an HTML page served by Apache HTTPD Server

23. Click on **Launch Instances**.

24. Navigate to the EC2 dashboard to see webserver-A and webserver-B running as shown below:



Instances (2) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
□	webserver-A	i-04cc0fc3c9cf096cc	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1c
□	webserver-B	i-0b232a1f25effff41	✓ Running	t2.micro	Initializing	No alarms +	us-east-1c

Task 3: Creating a Target Group

1. In the EC2 console, navigate to **Target Groups** in the left-side panel under **Load**

Balancer in the **Load Balancing** section.

2. Click on button on the top right corner.

3. Basic configuration:

   - Choose a target type : Select **Instances**
   - Target group name : Enter ***web-server-TG***
   - Protocol : Select **HTTP**
   - Port : Enter ***80***

4. Health Checks:

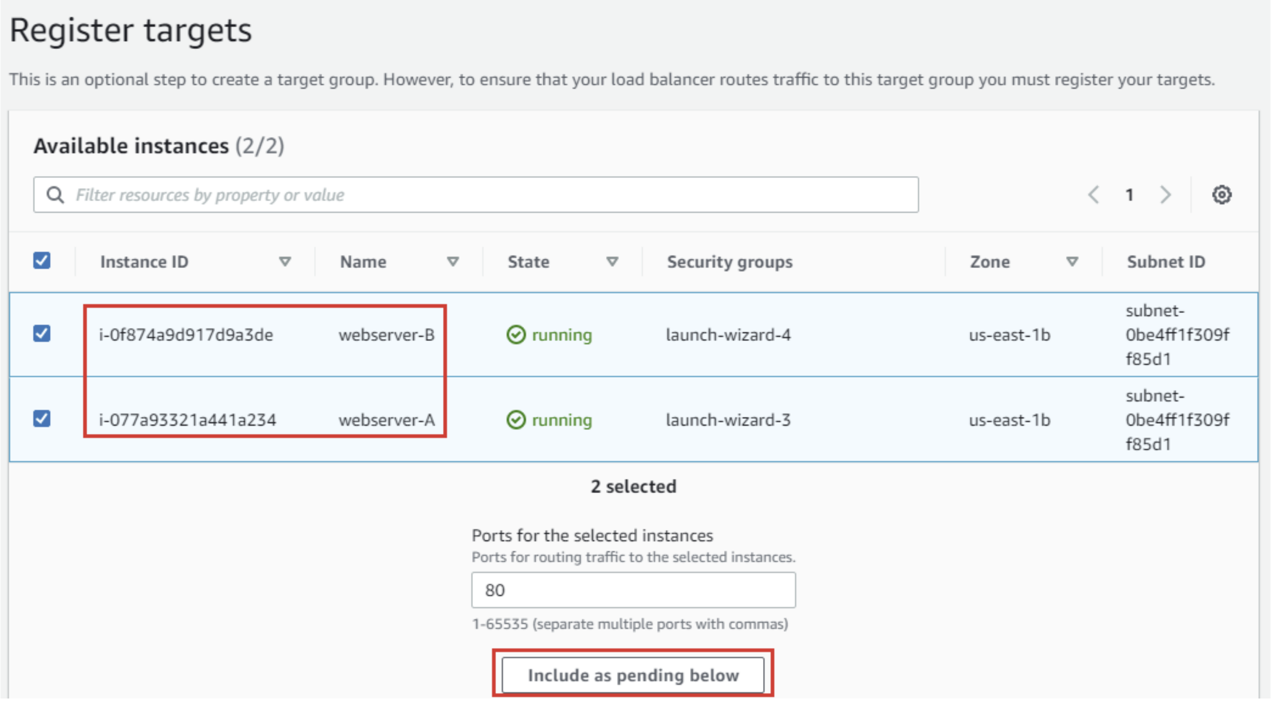
   - Health check protocol : Select HTTP
   - Health check path : Enter ***/index.html***
   - Click and expand **Advanced health check settings**
   - Healthy threshold : Enter ***3***
   - Unhealthy threshold : **2 (Default)**
   - Timeout : **5 seconds (Default)**
   - Interval : Enter ***6*** seconds
   - Success code : **200 (Default)**

5. Leave everything as default and click on **Next** button.

6. Register targets:

   - Select the two instances we have created i.e **webserver-A** and **webserver-B**

- Click on **Include as pending below** and scroll down



Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (2/2)

Filter resources by property or value

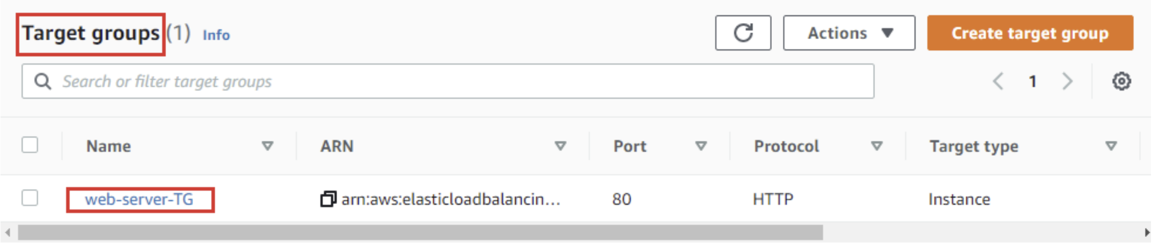
☑	Instance ID	Name	State	Security groups	Zone	Subnet ID
☑	i-0f874a9d917d9a3de	webserver-B	✓ running	launch-wizard-4	us-east-1b	subnet-0be4ff1f309ff85d1
☑	i-077a93321a441a234	webserver-A	✓ running	launch-wizard-3	us-east-1b	subnet-0be4ff1f309ff85d1

2 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

7. click on **Create target group**

8. Your Target group has been successfully created.



Target groups (1) Info

□	Name	ARN	Port	Protocol	Target type
□	web-server-TG	arn:aws:elasticloadbalancin...	80	HTTP	Instance

Task 4: Creating an Application Load Balancer

1. In the EC2 console, navigate to **Load Balancers** in the left-side panel under **Load Balancing**.

2. Click on

3. On the next screen, choose **Application Load Balancer** since we are testing the high availability of the web application and click on **Create** button.

4. Basic configuration:

   - Load balancer name : Enter ***Web-server-LB***
   - Scheme : Select **Internet-facing**
   - Ip address type : Choose **ipv4**

5. Network mapping:

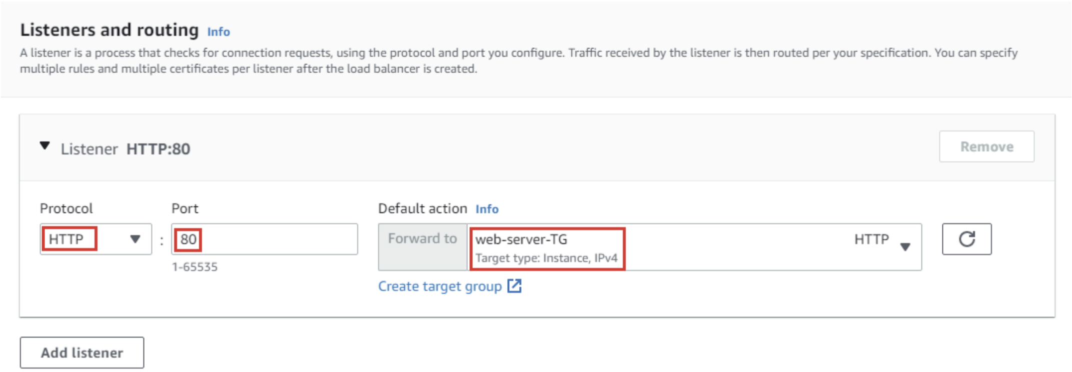
   - VPC : Select **Default**
   - Mappings : Check **All Availability Zones**

6. Security groups:

   - Security groups : Select **an existing security group** i.e **webserver-SG** from the drop down menu

7. Listeners and routing:

   - Protocol : Select **HTTP**
   - Port : Enter ***80***
   - Default action : Select **web-server-TG** from the drop down menu



Listeners and routing Info

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener **HTTP:80**

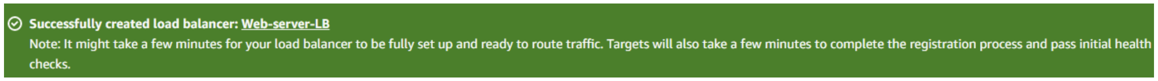
Protocol: HTTP

Port: 80 (1-65535)

Default action Info: Forward to web-server-TG (Target type: Instance, IPv4) HTTP

8. Leave everything as default and click on

9. You have successfully created Application Load Balancer.



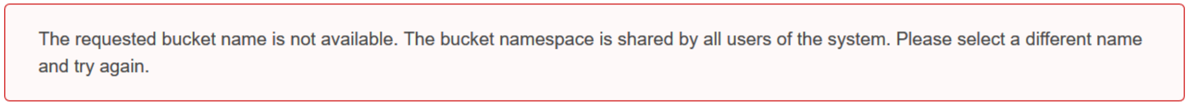
Successfully created load balancer: Web-server-LB

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

Task 5: Configuring the Load Balancer to store Access logs in S3 bucket

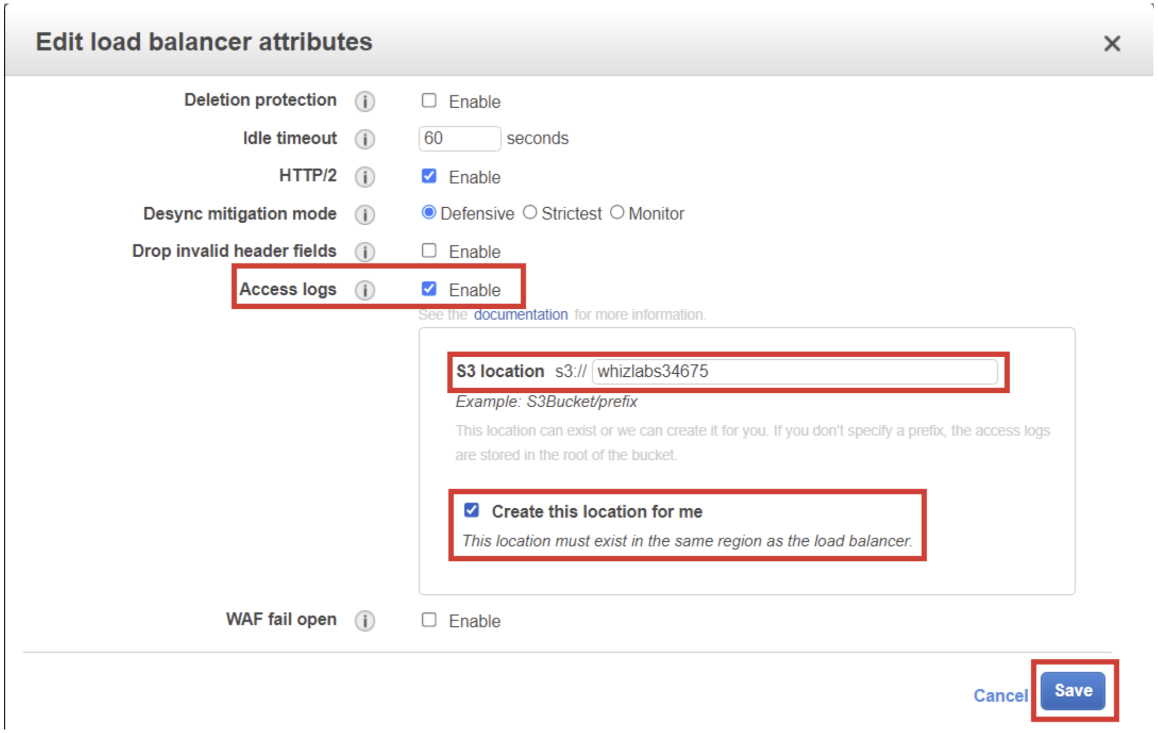
1. Navigate to **Load Balancers** and then select the load balancer that you have created in the above step.
2. Click on **Actions** and then click on **Edit attributes** to enable the access log feature.
3. Check the box next to the **Access log** and enter the **name of the bucket**(your choice) where you need to store the ELB access logs. For example, the **bucket name** in the below screenshot is **whizlabs34675**.
4. Check the box **Create this location for me** to create the S3 bucket in the same region as your ELB.
5. If you receive an error about the bucket name not being available, use a different,

unique name.



The requested bucket name is not available. The bucket namespace is shared by all users of the system. Please select a different name and try again.

6. Finally, click on **Save**.



Edit load balancer attributes ✕

Deletion protection ⓘ □ Enable

Idle timeout ⓘ 60 seconds

HTTP/2 ⓘ ☑ Enable

Desync mitigation mode ⓘ ● Defensive ○ Strictest ○ Monitor

Drop invalid header fields ⓘ □ Enable

Access logs ⓘ ☑ Enable

See the documentation for more information.

S3 location s3:// whizlabs34675

Example: S3Bucket/prefix

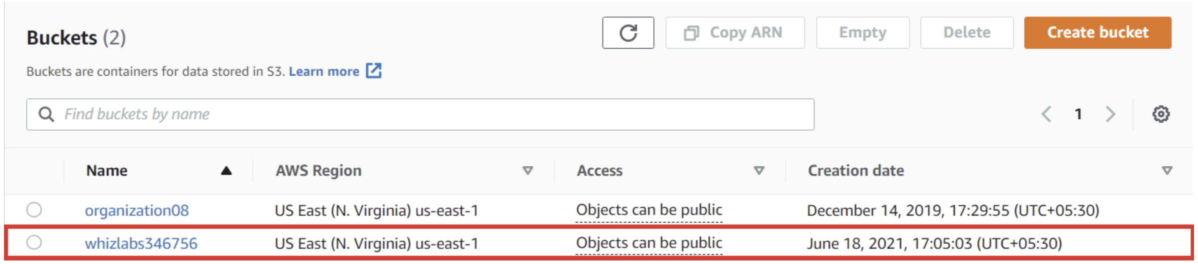
This location can exist or we can create it for you. If you don't specify a prefix, the access logs are stored in the root of the bucket.

☑ Create this location for me

This location must exist in the same region as the load balancer.

WAF fail open ⓘ □ Enable

7. Navigate to the **S3 console.** There you will be able to see the new bucket created.



Buckets (2)

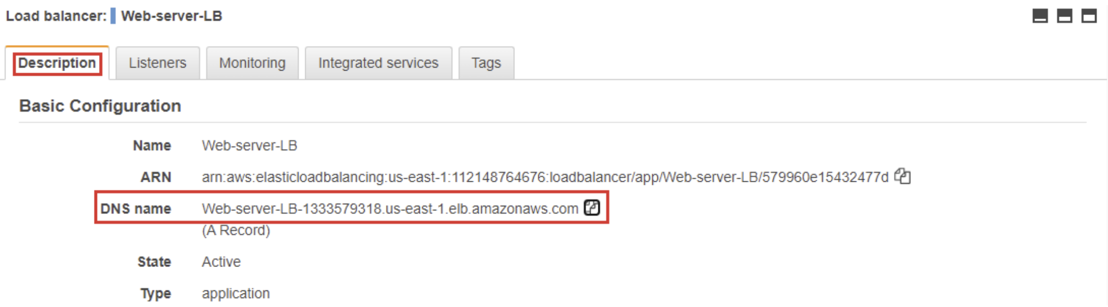
Buckets are containers for data stored in S3. Learn more

	Name	AWS Region	Access	Creation date
○	organization08	US East (N. Virginia) us-east-1	Objects can be public	December 14, 2019, 17:29:55 (UTC+05:30)
○	whizlabs346756	US East (N. Virginia) us-east-1	Objects can be public	June 18, 2021, 17:05:03 (UTC+05:30)

Task 6: Testing the Load Balancer and Stored Access Logs

1. Navigate to Load Balancers and select our **load balancer.** Click on **Description** , **copy the DNS name** and paste it in the browser.

   Example DNS URL: **Web-application-LB-1853289169.us-east-1.elb.amazonaws.com**



Load balancer: Web-server-LB

Basic Configuration

Name Web-server-LB

ARN arn:aws:elasticloadbalancing:us-east-1:112148764676:loadbalancer/app/Web-server-LB/579960e15432477d

DNS name Web-server-LB-1333579318.us-east-1.elb.amazonaws.com (A Record)

State Active

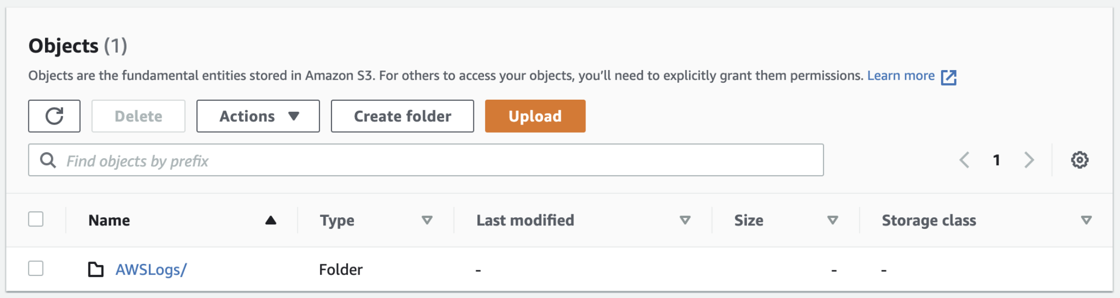
Type application

2. Refresh the browser couple of times and you will see the request is serving from both servers .i.e you will see the response either of the following two:

   - **RESPONSE COMING FROM SERVER A**
   - **RESPONSE COMING FROM SERVER B.**

   Note : This implies that load is shared between the two web servers via Application Load Balancer.

3. **Navigate to the S3 console** and enter into the bucket that you created to store ELB access logs. You will find the access logs under **AWSLogs** folder.



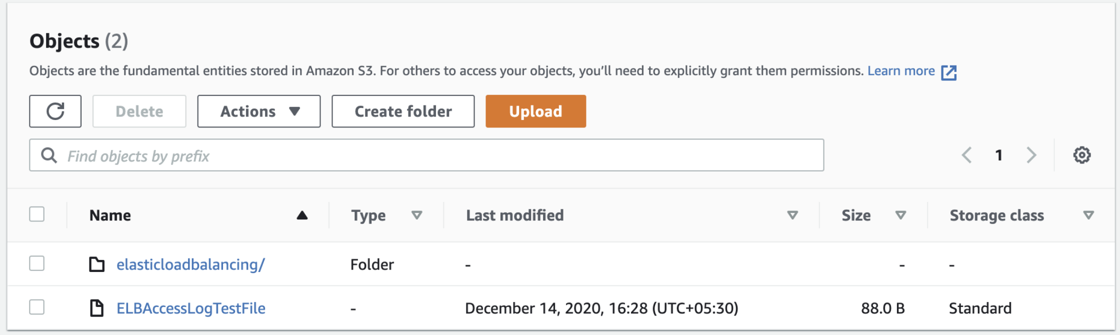
Objects (1)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	AWSLogs/	Folder	-	-	-

4. Click on the directory containing the load balancer URL to see whether the access logs are in the bucket. You should see a new folder as shown below:

Note: It can take up to 5 minutes for the **elasticloadbalancing** folder to be created.



Objects (2)

Objects are the fundamental entities stored in Amazon S3. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	elasticloadbalancing/	Folder	-	-	-
□	ELBAccessLogTestFile	-	December 14, 2020, 16:28 (UTC+05:30)	88.0 B	Standard

5. You can download the generated access log files (.zip file) to your local machine for review.

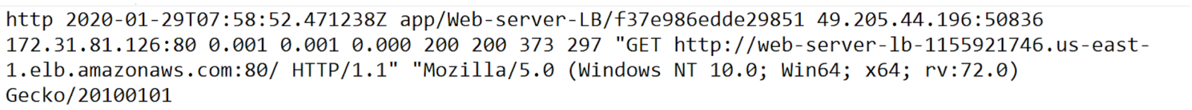
6. The **log file** will be present in a **hierarchy**, which goes like this:

- **(Bucket_name) / AWSLogs / (Account_number) / elasticloadbalancing / us-east-1 / (Year) / (Month) / (Day) / (LogFile)**

7. Select the file and click on the **Actions** button as above and choose **Download.** (Incase, you are unable to download the log file, click on the **Object actions** button above and choose the option to **Make public**, then try **downloading** again.)

8. You can extract the download file using **Winzip**.

9. Your log file entry will look like something like the snippet below:
   Note: Only 1 file will be created, and it will be updated as you access the ELB DNS more.



```
http 2020-01-29T07:58:52.471238Z app/Web-server-LB/f37e986edde29851 49.205.44.196:50836
172.31.81.126:80 0.001 0.001 0.000 200 200 373 297 "GET http://web-server-lb-1155921746.us-east-
1.elb.amazonaws.com:80/ HTTP/1.1" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:72.0)
Gecko/20100101
```

Note : The generated log file contains the following below:

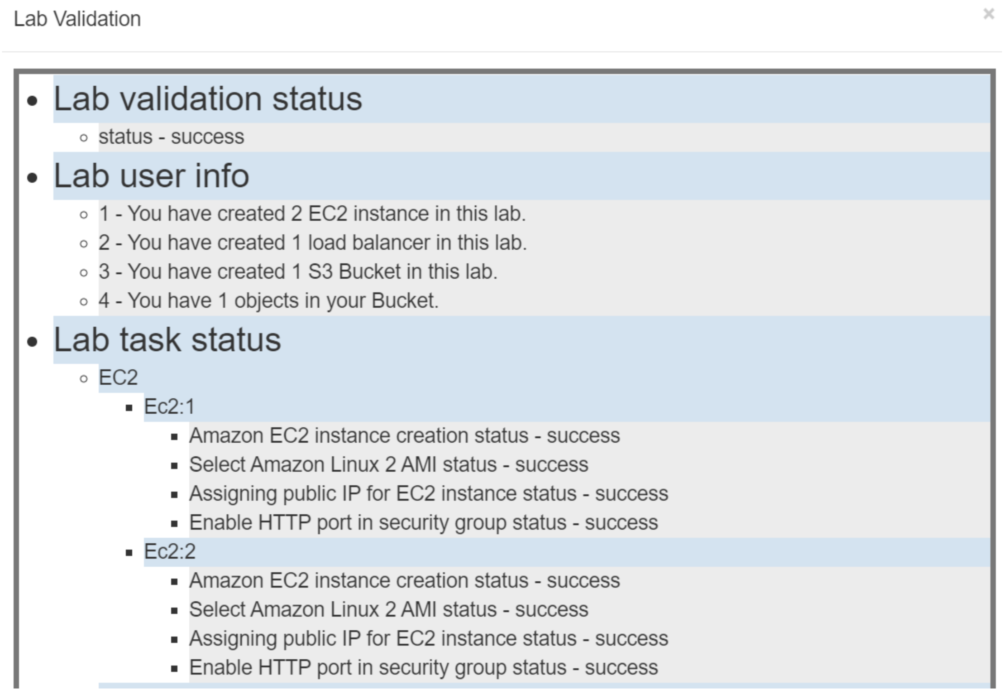
- **Time stamp** at which the load balancer accessed (2020-01-29T07:58:52.471238Z)
- **Name of the Load balancer** (Web-server-LB)
- **Client IP address**(49.205.44.196)
- **DNS name of Load balancer** (web-server-lb-1155921746.us-east-1.elb.amazonaws.com)
- The browser name (Mozilla)

Task 7: Validation Test

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.

2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.

3. Sample output :



Lab Validation

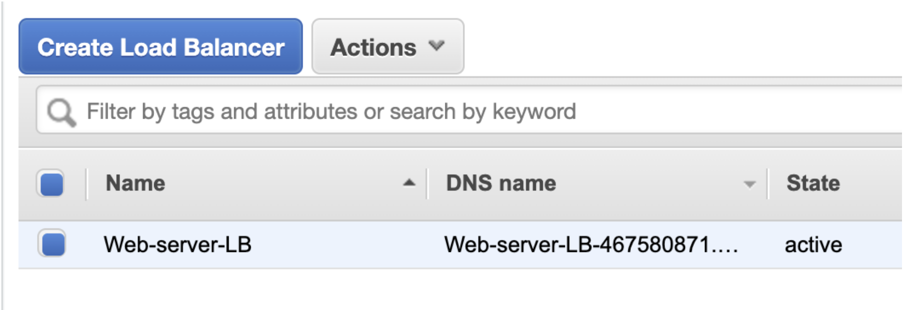
- Lab validation status
  - status - success
- Lab user info
  - 1 - You have created 2 EC2 instance in this lab.
  - 2 - You have created 1 load balancer in this lab.
  - 3 - You have created 1 S3 Bucket in this lab.
  - 4 - You have 1 objects in your Bucket.
- Lab task status
  - EC2
    - Ec2:1
      - Amazon EC2 instance creation status - success
      - Select Amazon Linux 2 AMI status - success
      - Assigning public IP for EC2 instance status - success
      - Enable HTTP port in security group status - success
    - Ec2:2
      - Amazon EC2 instance creation status - success
      - Select Amazon Linux 2 AMI status - success
      - Assigning public IP for EC2 instance status - success
      - Enable HTTP port in security group status - success

Task 8: Delete AWS Resources

Deleting Load balancer

1. In the EC2 console, navigate to **Load Balancers** in the left-side panel.

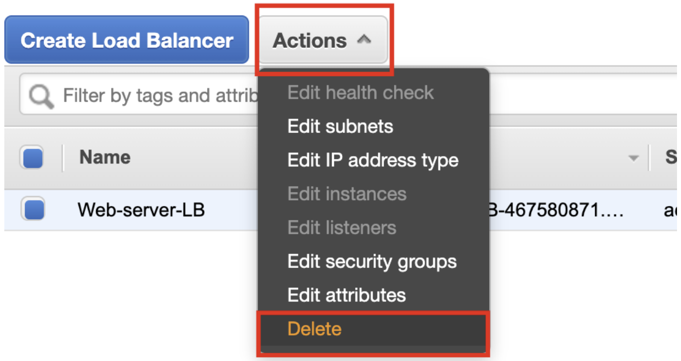
2. **Web-server-LB** will be listed here.



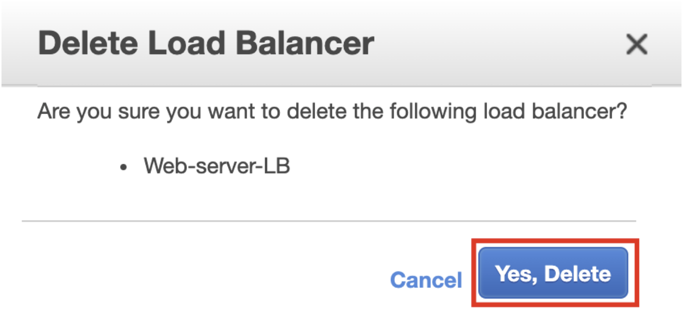
Name	DNS name	State
Web-server-LB	Web-server-LB-467580871....	active

3. To **delete** the load balancer, need to perform the following actions:

- **Select** the load balancer,
- Click on the **Actions** button,
- select the **Delete** option.



4. Confirm by clicking on the **Yes, Delete** button when a pop-up is shown.



Are you sure you want to delete the following load balancer?

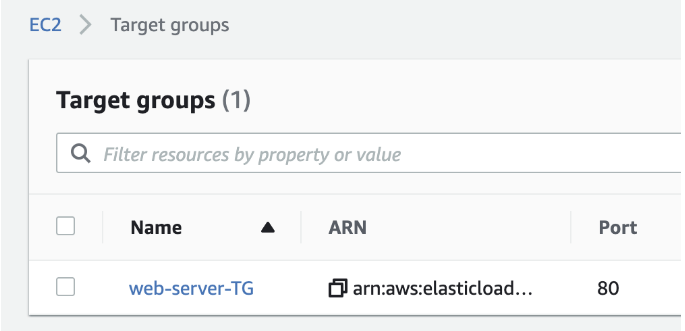
- Web-server-LB

5. Web-server-LG will be deleted immediately.

Deleting Target groups

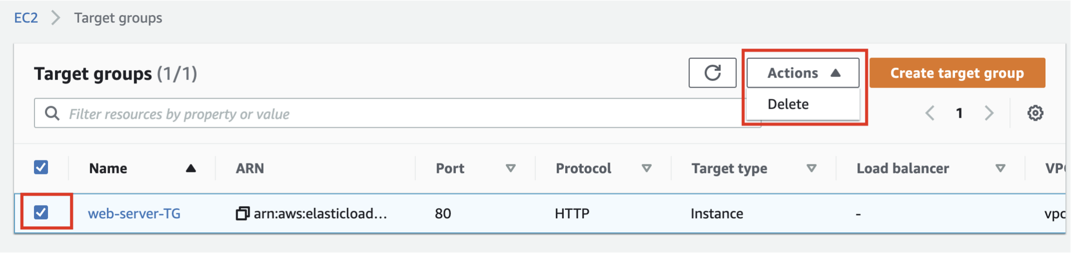
1. In the EC2 console, navigate to **Target groups** in the left-side panel.

2. **Web-server-TG** will be listed here.



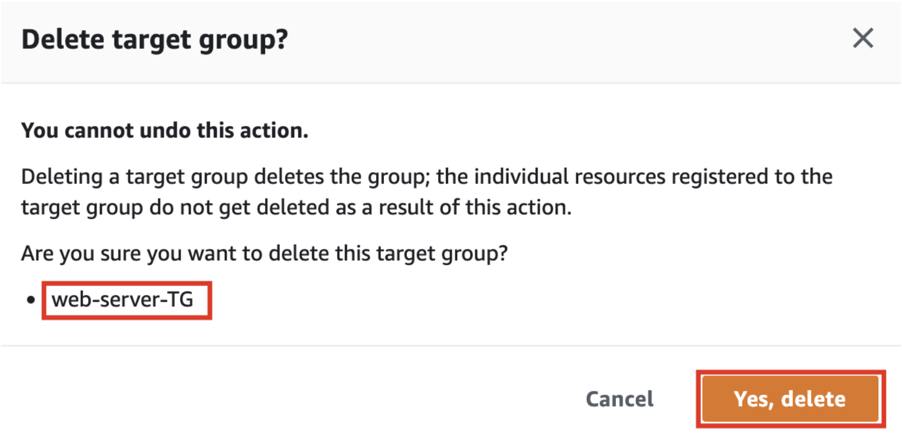
□	Name ▲	ARN	Port
□	web-server-TG	arn:aws:elasticload...	80

3. To delete the **target group**, need to perform the following actions:
    - **Select** the load balancer,
    - Click on the **Actions** button,
    - select the **Delete** option



☑	Name ▲	ARN	Port	Protocol	Target type	Load balancer	VP
☑	web-server-TG	arn:aws:elasticload...	80	HTTP	Instance	-	vpc

- Confirm by clicking on the **Yes, delete** button when a pop-up is shown.



Delete target group?

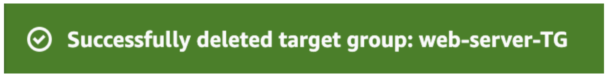
You cannot undo this action.

Deleting a target group deletes the group; the individual resources registered to the target group do not get deleted as a result of this action.

Are you sure you want to delete this target group?

- web-server-TG

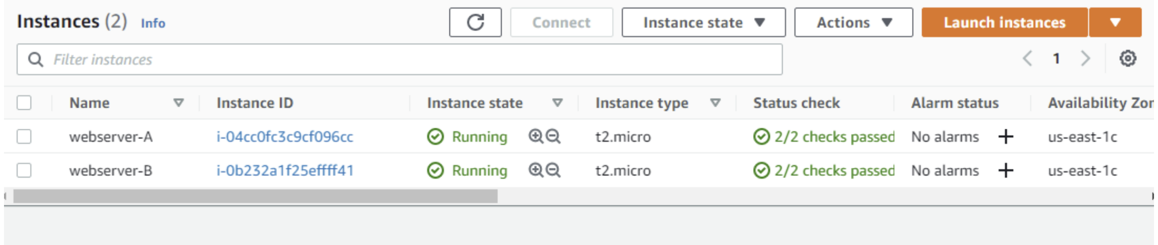
- Web-server-TG will be deleted immediately.



Successfully deleted target group: web-server-TG

Terminating EC2 Instances

1. In the EC2 console, navigate to ▼ INSTANCES in the left-side panel.
2. Two EC2 Instance **Webserver-A** and **Webserver-B** will be listed here.



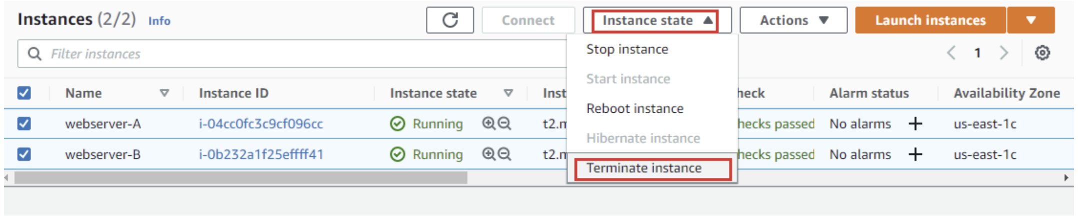
Instances (2) Info

□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zor
□	webserver-A	i-04cc0fc3c9cf096cc	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1c
□	webserver-B	i-0b232a1f25effff41	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1c

3. To terminate the **EC2 Instances**, need to perform the following actions:

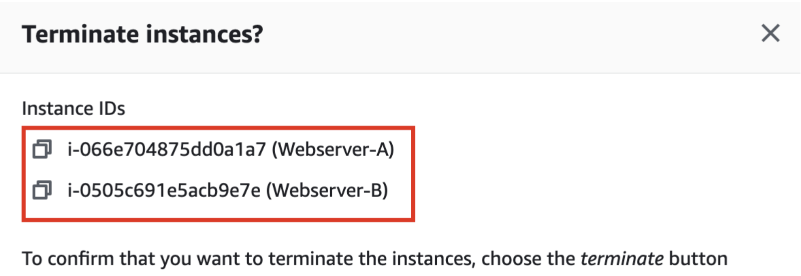
- **Select** the EC2 instances,
- Click on the **Instance state** button,

- select the **Terminate instance** option



☑	Name	Instance ID	Instance state	Inst	heck	Alarm status	Availability Zone
☑	webserver-A	i-04cc0fc3c9cf096cc	Running	t2.m	hecks passed	No alarms +	us-east-1c
☑	webserver-B	i-0b232a1f25effff41	Running	t2.m	hecks passed	No alarms +	us-east-1c

4. Confirm by clicking on the **Terminate** button when a pop-up is shown.



Terminate instances?

Instance IDs

i-066e704875dd0a1a7 (Webserver-A)

i-0505c691e5acb9e7e (Webserver-B)

To confirm that you want to terminate the instances, choose the *terminate* button

About Us Subscription Instructions and Guidelines FAQ’s Contact Us