



ASSIGNMENT-2

❖ Load balancer

Name: Bontha lakshmi prasanna
Gmail: prannaprantha36@gmail.com

❖ Create security group

EC2 Dashboard

EC2 Global View

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

Security Groups (1) Info

Actions

Export security groups to CSV

Create security group

Find resources by attribute or tag

< 1 >

<input type="checkbox"/>	Name	Security group ID	Security group name
<input type="checkbox"/>	-	sg-0a9d7f353334eeaa	default

EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name Info

web-sg

Name cannot be edited after creation.

Description Info

nothing

VPC Info

vpc-06a9afae9046333f5

Inbound rules Info

Inbound rule 1

Delete

Type Info

SSH

Protocol Info

TCP

Port range Info

22

Source type Info

Anywhere-IPv4

Source Info

0.0.0.0/0

0.0.0.0/0 X

Description - optional Info

Inbound rule 2

Delete

Type [Info](#)

Protocol [Info](#)

Port range [Info](#)

Custom TCP

TCP

80

Source type [Info](#)

Source [Info](#)

Description - optional [Info](#)

Anywhere-IPv4

0.0.0.0/0

0.0.0.0/0 X

⚠ Rules with destination of 0.0.0.0/0 or ::/0 allow all IP addresses to leave the instance. We recommend setting security group rules to leave the instance from known IP addresses only.

X

Tags - optional

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

No tags associated with the resource.

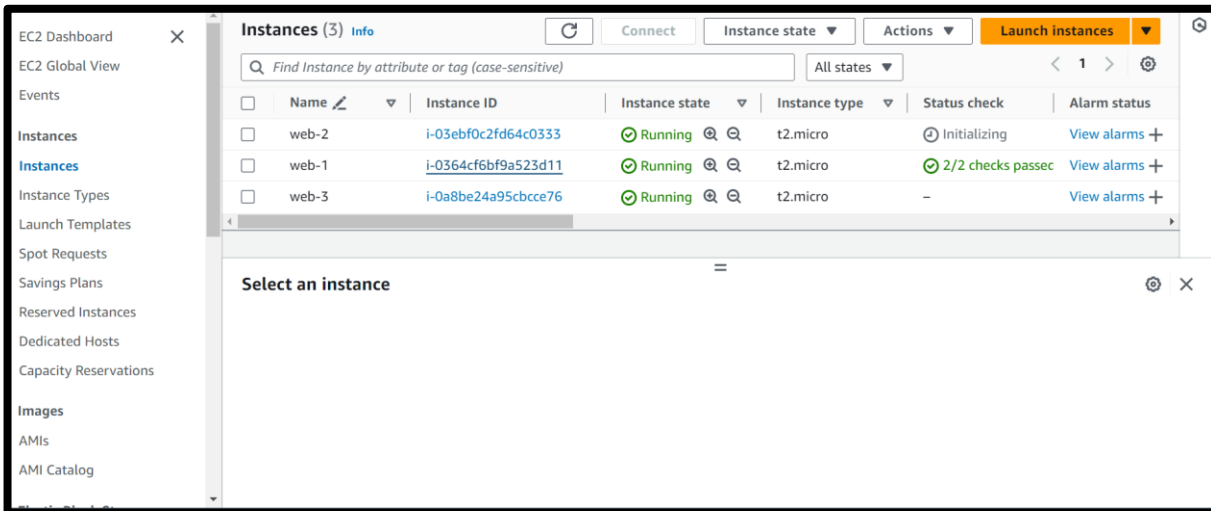
Add new tag

You can add up to 50 more tags

Cancel

Create security group

❖ Create three instances



The screenshot shows the AWS Management Console 'Instances' page. The left sidebar contains navigation links for EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, and AMI Catalog. The main content area is titled 'Instances (3)' and includes a search bar, a filter dropdown set to 'All states', and a table of three instances. Below the table is a 'Select an instance' section.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	web-2	i-03ebf0c2fd64c0333	Running	t2.micro	Initializing	View alarms
<input type="checkbox"/>	web-1	i-0364cf6bf9a523d11	Running	t2.micro	2/2 checks passed	View alarms
<input type="checkbox"/>	web-3	i-0a8be24a95cbcce76	Running	t2.micro	-	View alarms

Connect to instance

Connect to your instance [i-0364cf6bf9a523d11](#) (web-1) using any of these options

[EC2 Instance Connect](#) | [Session Manager](#) | [SSH client](#) | [EC2 serial console](#)

Instance ID

[i-0364cf6bf9a523d11](#) (web-1)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is `load-web1.pem`
3. Run this command, if necessary, to ensure your key is not publicly viewable.
`chmod 400 "load-web1.pem"`
4. Connect to your instance using its Public DNS:
`ec2-3-8-48-9.eu-west-2.compute.amazonaws.com`

Example:

```
ssh -i "load-web1.pem" ec2-user@ec2-3-8-48-9.eu-west-2.compute.amazonaws.com
```

Now we have to Install nginx in load-1

[illegible]

```

Transaction Summary
-----
stall 7 Packages

tal download size: 1.0 M
stalled size: 3.4 M
unloading Packages:
/?: generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                277 kB/s
/?: libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                          891 kB/s
/?: gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                    3.5 MB/s
/?: nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm                             1.4 MB/s
/?: nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm                       13 MB/s
/?: nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm                  271 kB/s
/?: nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                   626 kB/s
-----
tal                                                                    5.5 MB/s

nning transaction check
ansaction check succeeded.
nning transaction test
ansaction test succeeded.
nning transaction
Preparing
Running Scriptlet:
Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
Installing : libunwind-1.4.0-5.amzn2023.0.2.x86_64
Installing : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Installing : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64
Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
Installing : nginx-1:1.24.0-1.amzn2023.0.2.x86_64

```


➤ Now we have to Install nginx in load-3

```

vaminidurga@LAPTOP-H3TERLUG MINGW64 /d/AWS Keys
$ ssh -i "load-web1.pem" user@ec2-18-171-222-170.eu-west-2.compute.amazonaws.com
The authenticity of host 'ec2-18-171-222-170.eu-west-2.compute.amazonaws.com (18.171.222.170)' can't be established.
ED25519 key fingerprint is SHA256:FzWQUgTibtQ1JAFmZwtzRfSE8alOmOm+PVMKGcSH+6c.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-171-222-170.eu-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

```

[illegible]

```
[root@ip-172-31-1-165 ~]# yum update -y
Last metadata expiration check: 0:21:28 ago on Thu Jul 4 13:58:24 2024.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-1-165 ~]# yum install nginx -y
Last metadata expiration check: 0:21:41 ago on Thu Jul 4 13:58:24 2024.
Dependencies resolved.
```

Package	Architecture	Version	Repository	Size
Installing:				
nginx	x86_64	1:1.24.0-1.amzn2023.0.2	amazonlinux	32 k
Installing dependencies:				
generic-logos-httpd	noarch	18.0.0-12.amzn2023.0.3	amazonlinux	19 k
gperf-tools-libs	x86_64	2.9.1-1.amzn2023.0.3	amazonlinux	308 k
libunwind	x86_64	1.4.0-5.amzn2023.0.2	amazonlinux	66 k
nginx-core	x86_64	1:1.24.0-1.amzn2023.0.2	amazonlinux	586 k
nginx-filesystem	noarch	1:1.24.0-1.amzn2023.0.2	amazonlinux	9.1 k
nginx-mimetypes	noarch	2.1.49-3.amzn2023.0.3	amazonlinux	21 k

Transaction Summary

```
=====
Install 7 Packages
```

[illegible]Running
Transact

Transaction check succeeded.
Running transaction test

Transaction test succeeded
Running transaction

```
Preparing      :
```

```
Running scriptlet:
Installing      :
```

Installing	: nginx-1:1.24.0-1.amzn2023.0.2.x86_64	7/7
Running scriptlet:	: nginx-1:1.24.0-1.amzn2023.0.2.x86_64	7/7
Verifying	: generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch	1/7
Verifying	: gperf-tools-1:1.1.amzn2023.0.3.x86_64	2/7
Verifying	: libunwind-1.4.0-5.amzn2023.0.2.x86_64	3/7
Verifying	: nginx-1:1.24.0-1.amzn2023.0.2.x86_64	4/7
Verifying	: nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64	5/7
Verifying	: nlohmann-json-3.11.1-1.amzn2023.0.2.noarch	6/7
Verifying	: nginx-mmio-types-2.1.49-3.amzn2023.0.3.noarch	7/7

```
Installed:
generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch      gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64      libunwind-1.4.0-5.amzn2023.0.2.x86_64
nginx-1:1.24.0-1.amzn2023.0.2.x86_64                  nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64          nginx-filessystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
```

Complete!

```

Installing : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
Installing : nginx-1.1.24.0-1.amzn2023.0.2.x86_64
Running scriptlet: nginx-1.1.24.0-1.amzn2023.0.2.x86_64
Verifying : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch
Verifying : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64
Verifying : libunwind-1.4.0-5.amzn2023.0.2.x86_64
Verifying : nginx-1.1.24.0-1.amzn2023.0.2.x86_64
Verifying : nginx-core-1.1.24.0-1.amzn2023.0.2.x86_64
Verifying : nginx-filesystem-1.1.24.0-1.amzn2023.0.2.noarch
Verifying : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

```

Complete!

```
[root@ip-172-31-11-165 ~]# cd /usr/share/nginx/html
[root@ip-172-31-11-165 html]# mv index.html
```

```
[root@ip-172-31-11-165 html]# rm index.html
rm: remove regular file 'index.html'? yes
```

```
[root@ip-172-31-11-165 html]# vi index.html
```



```
this is webserver3
```

```
~  
~  
~  
~  
~
```



Not secure

18.171.222.170

this is webserver3

❖ Now create target group

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 (3)

< 1 > ⚙

<input type="checkbox"/>	Instance ID ▾	Name ▾	State ▾	Security groups ▾
<input type="checkbox"/>	i-0a8be24a95cbcce76	web-3	✔ Running	web-sg
<input type="checkbox"/>	i-03ebf0c2fd64c0333	web-2	✔ Running	web-sg
<input type="checkbox"/>	i-0364cf6bf9a523d11	web-1	✔ Running	web-sg

Review targets

Targets (3)

Remove all pending

☐ Show only pending

< 1 > ⚙

Instance ID ▾	Name ▾	Port ▾	State ▾	Security groups ▾	Zone ▾	Priority ▾
i-0a8be24a95cbcce76	web-3	80	✔ Running	web-sg	eu-west-2c	172
i-03ebf0c2fd64c0333	web-2	80	✔ Running	web-sg	eu-west-2b	172
i-0364cf6bf9a523d11	web-1	80	✔ Running	web-sg	eu-west-2a	172

3 pending

Cancel

Previous

Create target group

❖ Create load balancer

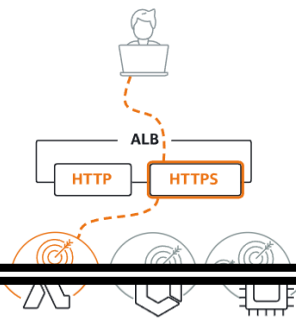
[EC2](#) > [Load balancers](#) > Compare and select load balancer type

Compare and select load balancer type

A complete feature-by-feature comparison along with detailed highlights is also available. [Learn more](#)

Load balancer types

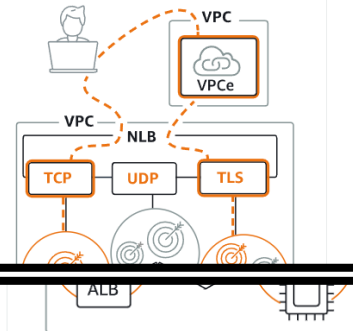
Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

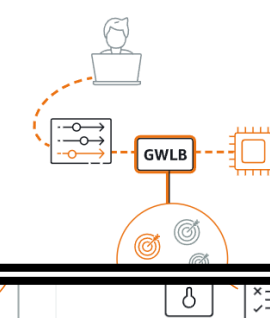
Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

Create

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

[EC2](#) > [Load balancers](#) > Create Application Load Balancer

Create Application Load Balancer [Info](#)

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances or Amazon ECS tasks on request attributes. When the load balancer receives a connection request, it evaluates the listener rules. If a rule is applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Create Application Load Balancer Info

The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.

► How Application Load Balancers work

Basic configuration

Load balancer name

Name must be unique within your AWS account and can't be changed after the load balancer is created.

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

Scheme Info

Scheme can't be changed after the load balancer is created.

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.

☒ eu-west-2a (euw2-az2)

Subnet

IPv4 address

Assigned by AWS

☒ eu-west-2b (euw2-az3)

Subnet

IPv4 address

Assigned by AWS

☒ eu-west-2c (euw2-az1)

Subnet

IPv4 address

Assigned by AWS

❖ Attach security group

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

web-sg
sg-00f69d8cd9262c45c VPC: vpc-06a9afae9046333f5

❖ Attach target group

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

Remove

Protocol HTTP Port 80
1-65535

Default action [Info](#)
Forward to prasanna HTTP
Target type: Instance, IPv4
[Create target group](#)

Listener tags - optional
Consider adding tags to your listener. Tags enable you to categorize your AWS resources so you can more easily manage them.

Add listener tag

You can add up to 50 more tags.

Add listener

Service integrations [Edit](#)
AWS WAF: None
AWS Global Accelerator: None

Tags [Edit](#)
None

Attributes

ⓘ

 Certain default attributes will be applied to your load balancer. You can view and edit them after creating the load balancer.

Creation workflow and status

► Server-side tasks and status

After completing and submitting the above steps, all server-side tasks and their statuses become available for monitoring.

Cancel

Create load balancer

❖ Copy DNS link and paste it on browser and the page gets update

Load balancer type Application	Status Active	VPC vpc-0f1ba19fa638a9748	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone Z368ELLRRE2KJO	Availability Zones subnet-0107b1683b8ff1fd9 us-west-1b (usw1-az1) subnet-0cedaa36fcb11eff8 us-west-1a (usw1-az3)	Date created July 9, 2024, 17:15 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:us-west-1:654654627808:loadbalancer/app/prasanna/f04b6d381ad1afbb		DNS name Info prasanna-1245736017.us-west-1.elb.amazonaws.com (A Record)	

