

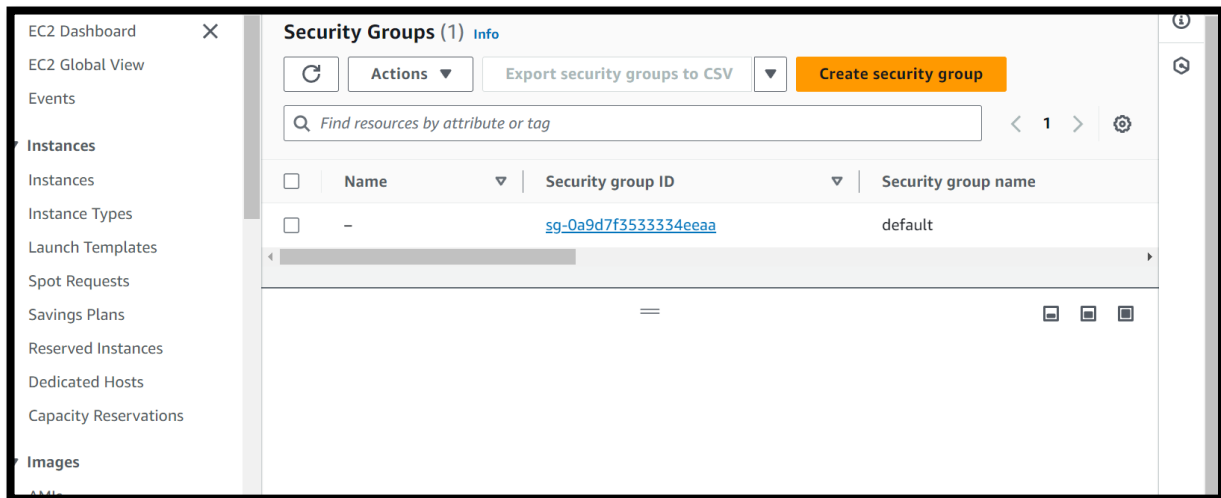


ASSIGNMENT-2

❖ Load balancer

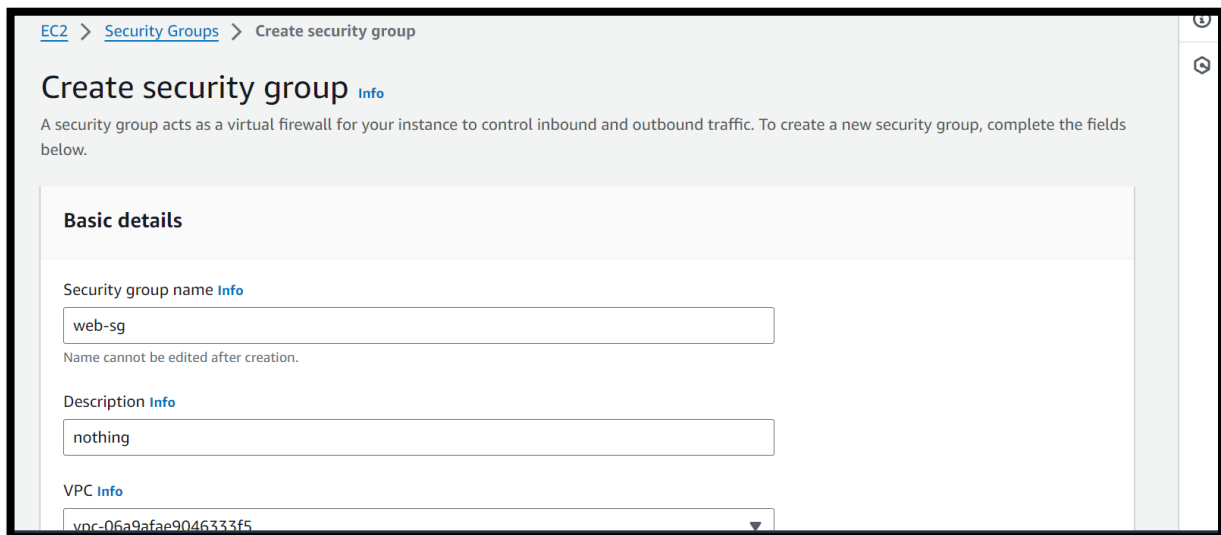
Name: Yamini Durga Padithapu
Gmail: yaminidurga190@gmail.com

❖ Create security group



The screenshot shows the AWS Management Console's 'Security Groups' page. On the left is a navigation menu with options like 'EC2 Dashboard', 'Instances', 'Images', etc. The main panel is titled 'Security Groups (1) Info'. It features a search bar, a table with one entry, and a 'Create security group' button.

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



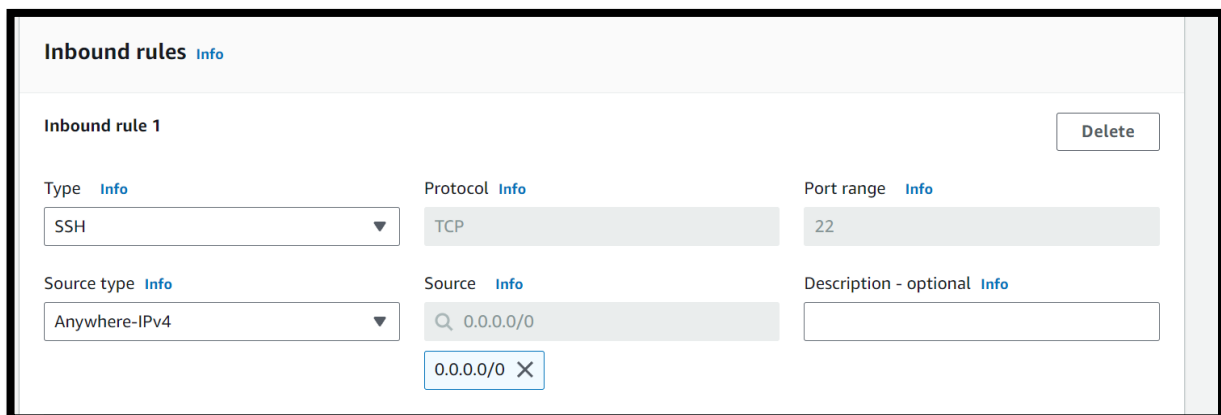
This screenshot shows the 'Create security group' form. It includes a breadcrumb trail 'EC2 > Security Groups > Create security group' and a title 'Create security group Info'. A descriptive paragraph explains the purpose of a security group. Below is a 'Basic details' section with three fields: 'Security group name' (filled with 'web-sg'), 'Description' (filled with 'nothing'), and 'VPC' (filled with 'vpc-06a9afae9046333f5').

Basic details

Security group name [Info](#)
web-sg
Name cannot be edited after creation.

Description [Info](#)
nothing

VPC [Info](#)
vpc-06a9afae9046333f5



This screenshot shows the 'Inbound rules' section. It has a title 'Inbound rules Info' and a 'Delete' button. Below is a form for 'Inbound rule 1' with six fields: 'Type' (SSH), 'Protocol' (TCP), 'Port range' (22), 'Source type' (Anywhere-IPv4), 'Source' (0.0.0.0/0), and 'Description - optional'.

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
Description - optional [Info](#):

Inbound rule 2

Delete

Type [Info](#)

Custom TCP

Protocol [Info](#)

TCP

Port range [Info](#)

80

Source type [Info](#)

Anywhere-IPv4

Source [Info](#)

Q 0.0.0.0/0

0.0.0.0/0 X

Description - optional [Info](#)

⚠ 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.

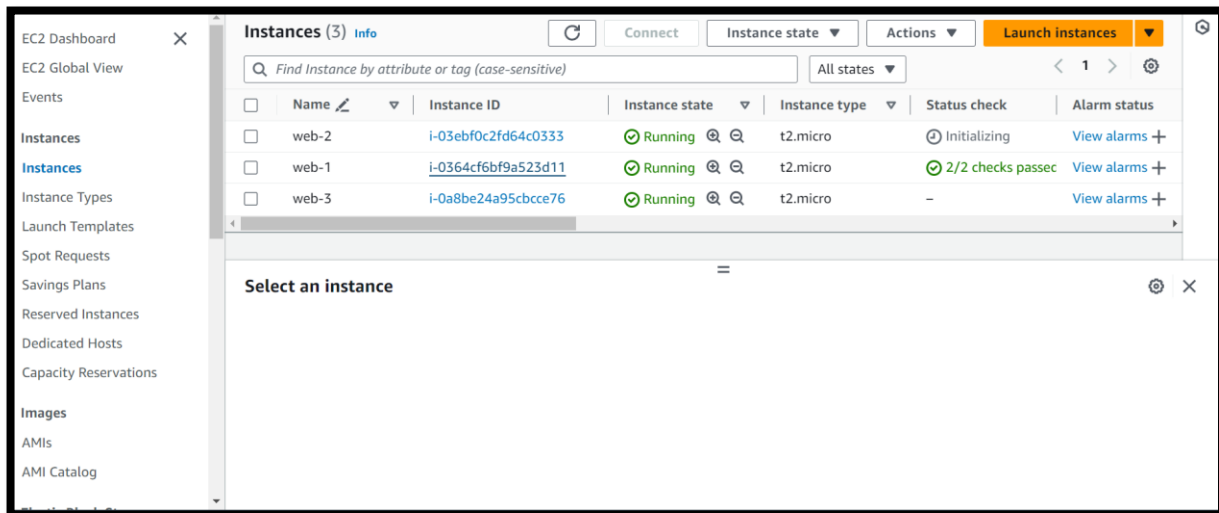
Add new tag

You can add up to 50 more tags

Cancel

Create security group

❖ Create three instances



The screenshot shows the AWS Management Console 'Instances' page. On the left is a navigation sidebar with links to 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 panel is titled 'Instances (3) Info' and includes a search bar, a 'Find Instance by attribute or tag (case-sensitive)' input, and a 'All states' dropdown. Below this is a table listing three instances:

	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-0a8be24a95cbce76	Running	t2.micro	-	View alarms

Below the table is a 'Select an instance' section with a search bar and a list of instance IDs.



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`

❖ Connect load-1

Now we have to Install nginx in load-1

```

admin@ubuntuLAPTOP-H3IERLUG:~$ ssh -i "/root/.ssh/dec2-3-8-48-9.eu-west-2.compute.amazonaws.com" ec2-user@ec2-3-8-48-9.eu-west-2.compute.amazonaws.com
Warning: Permanently added 'ec2-3-8-48-9.eu-west-2.compute.amazonaws.com' (EC25519) to the list of known hosts.
ec2-user@ip-172-31-19-120 ~$ sudo -i
root@ip-172-31-19-120 ~# yum update -y
Last metadata expiration check: 0:05:21 ago on Thu Jul 4 13:56:21 2024.
Dependencies resolved.
Nothing to do.
Complete!
root@ip-172-31-19-120 ~# yum install nginx -y
Last metadata expiration check: 0:05:38 ago on Thu Jul 4 13:56:21 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
gperftools-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-mmtyypes                      noarch            2.1.49-3.amzn2023.0.3                amazonlinux          21 k
=====

Transaction Summary
-----
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                277 kB/s | 19 kB    00:00

```

```

-----
Transaction Summary
-----
install 7 Packages

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

nning transaction check
transaction check succeeded.
nning transaction test
transaction test succeeded.
nning transaction
Preparing
Running scriptlet: nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch
Installing : nginxfilesystem-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

```

```

Transaction Summary
-----
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                277 kB/s | 19 kB    00:00
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                          891 kB/s | 66 kB    00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                    3.5 MB/s | 308 kB   00:00
(4/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm                             1.4 MB/s | 32 kB     00:00
(5/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm                       13 MB/s | 586 kB    00:00
(6/7): nginxfilesystem-1.24.0-1.amzn2023.0.2.noarch.rpm                   271 kB/s | 9.1 kB   00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                   626 kB/s | 21 kB     00:00
-----
Total: 5.5 MB/s | 1.0 MB    00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : 1/1
  Running scriptlet: nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch      1/7
  Installing          : nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch      1/7
  Installing          : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch        2/7
  Installing          : libunwind-1.4.0-5.amzn2023.0.2.x86_64              3/7
  Installing          : gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64        4/7
  Installing          : nginx-core-1:1.24.0-1.amzn2023.0.2.x86_64          5/7
  Installing          : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch    6/7
  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           : gperftools-libs-2.9.1-1.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           : nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch      6/7
  Verifying           : nginx-mimetypes-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        nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch
  nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-19-120 ~]#

```

```

Complete!
[root@ip-172-31-19-120 ~]# cd /usr/share/nginx/html
[root@ip-172-31-19-120 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-172-31-19-120 html]# vi index.html

```

```
this is webserver1
```

```

Complete!
[root@ip-172-31-19-120 ~]# cd /usr/share/nginx/html
[root@ip-172-31-19-120 html]# rm index.html
rm: remove regular file 'index.html'? yes
[root@ip-172-31-19-120 html]# vi index.html
[root@ip-172-31-19-120 html]# systemctl restart nginx
[root@ip-172-31-19-120 html]#

```

← → ↻ ⚠ Not secure 3.8.48.9

this is webserver1

➤ Now we have to Install nginx in load-2

```
[vaminidurga@LAPTOP-H3IERLUG MINGW64 /d/AWS Keys]
$ ssh -i "load-web1.pem" ec2-user@ec2-13-40-108-21.eu-west-2.compute.amazonaws.com
The authenticity of host 'ec2-13-40-108-21.eu-west-2.compute.amazonaws.com (13.40.108.21)' can't be established.
ED25519 key fingerprint is SHA256:VOXBmRq/WpbUtXnQBZE/lVlgOrFQnnfwMu7XA+clzY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-40-108-21.eu-west-2.compute.amazonaws.com' (ED25519) to the list of known hosts.
```

```
y~  
##  
##### Amazon Linux 2023  
~~~~\#####  
~~~\###|  
~~~/#/  
~~~~V~'|_> https://aws.amazon.com/linux/amazon-linux-2023  
~~~~_  
~~~~_|_/_<->  
~~~~|_/_/_
```

```
[ec2-user@ip-172-31-42-144 ~]$
```

```

root@ip-172-31-11-103 ~# 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-172-31-11-165 ~]# 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
gperftools-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

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm
(4/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm
(5/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm
(6/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm

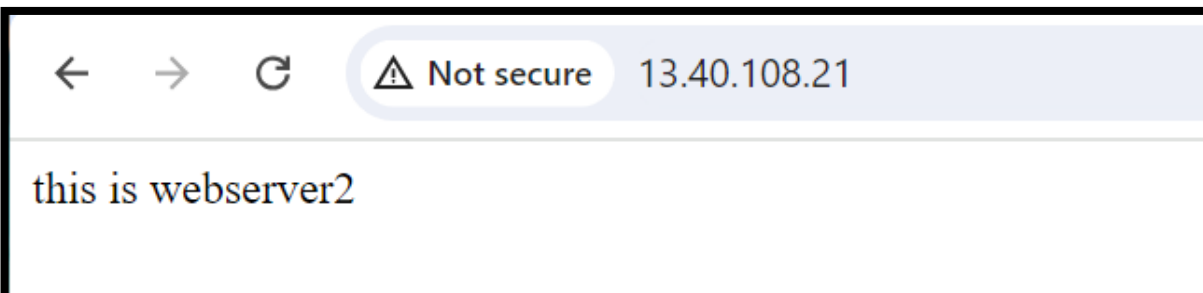
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing
Running scriptlet: nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
Installing : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
Installing : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
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-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-logs-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
Installed:
generic-logs-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-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch
Complete!
72-21-11-165 16:04 /usr/share/nginx/html

```



➤ Now we have to Install nginx in load-3

[illegible]

```

[root@ip-172-31-11-103 ~]# 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-172-31-11-165 ~]# 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
gperftools-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      91 k
nginx-mimetypes                      noarch            2.1.49-3.amzn2023.0.3    amazonlinux      21 k

Transaction Summary
-----
Install 7 Packages

Total download size: 1.0 M
Installed size: 3.4 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm                294 kB/s | 19 kB    00:00
(2/7): libunwind-1.4.0-5.amzn2023.0.2.x86_64.rpm                          978 kB/s | 66 kB    00:00
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm                     3.7 MB/s | 308 kB    00:00
(4/7): nginx-1.24.0-1.amzn2023.0.2.x86_64.rpm                             1.6 MB/s | 32 kB    00:00
(5/7): nginx-filesystem-1.24.0-1.amzn2023.0.2.noarch.rpm                   569 kB/s | 91 kB    00:00
(6/7): nginx-core-1.24.0-1.amzn2023.0.2.x86_64.rpm                         14 MB/s | 586 kB    00:00
(7/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm                    768 kB/s | 21 kB    00:00
-----
Total                                                                    5.7 MB/s | 1.0 MB    00:00

Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing...
Running scriptlet: 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-1:1.24.0-1.amzn2023.0.2.x86_64
Running scriptlet:
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      : nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch
Verifying      : nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Installing:
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      nginxfilesystem-1:1.24.0-1.amzn2023.0.2.noarch
nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch

Complete!
[root@ip-172-31-11-165 ~]# cd /usr/share/nginx/html

```

```

Installing      : generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch          6/7
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       : gperftools-libs-2.9.1-1.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       : nginx-filesystem-1:1.24.0-1.amzn2023.0.2.noarch            6/7
Verifying       : nginx-mimetypes-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-filesystem-1:1.24.0-1.amzn2023.0.2.noarch
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]# 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

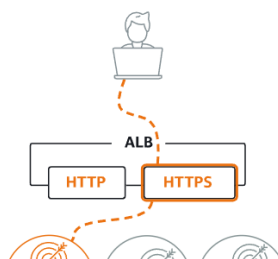
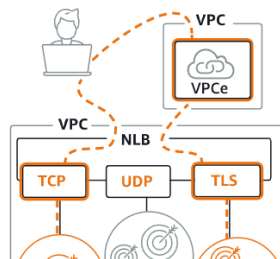

❖ Create load balancer

[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	Network Load Balancer Info	Gateway Load Balancer Info
		

<p>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.</p> <p>Create</p>	<p>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.</p> <p>Create</p>	<p>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.</p> <p>Create</p>
---	--	---

[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 that are 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.

[Schema](#) [Info](#)

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

yamini

Target type: Instance, IPv4

HTTP

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.

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

▼ Details			
Load balancer type Application	Status ✔ Active	VPC vpc-06a9afae9046333f5	Load balancer IP address type IPv4
Scheme Internet-facing	Hosted zone ZHURV8PSTC4K8	Availability Zones subnet-096769ccde0020bf5 eu-west-2a (euw2-az2) subnet-06d4f9d0fa446c276 eu-west-2c (euw2-az1) subnet-0e2e43e26d442930b eu-west-2b (euw2-az3)	Date created July 4, 2024, 20:01 (UTC+05:30)
Load balancer ARN arn:aws:elasticloadbalancing:eu-west-2:905418107262:loadbalancer/app/yaminild/0fa9c946461fbb36		DNS name Info yaminild-497449396.eu-west-2.elb.amazonaws.com (A Record)	

