Elastic Load Balancer

Elastic Load Balancing distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of your applications. A load balancer accepts incoming traffic from clients and routes requests to its registered EC2 instances in one or more Availability Zones. The load balancer also monitors the health of its registered instances and ensures that it routes traffic only to healthy instances. When the load balancer detects an unhealthy instance, it stops routing traffic to that instance, and then resumes routing traffic to that instance when it detects that the instance is healthy again.

Elastic Load Balancing supports two types of load balancers:

Application Load Balancers and Classic Load Balancers. There is a key difference between the way you configure these load balancers. With a Classic Load Balancer, you register instances with the load balancer. With an Application Load Balancer, you register the instances as targets in a target group, and route traffic to a target group.

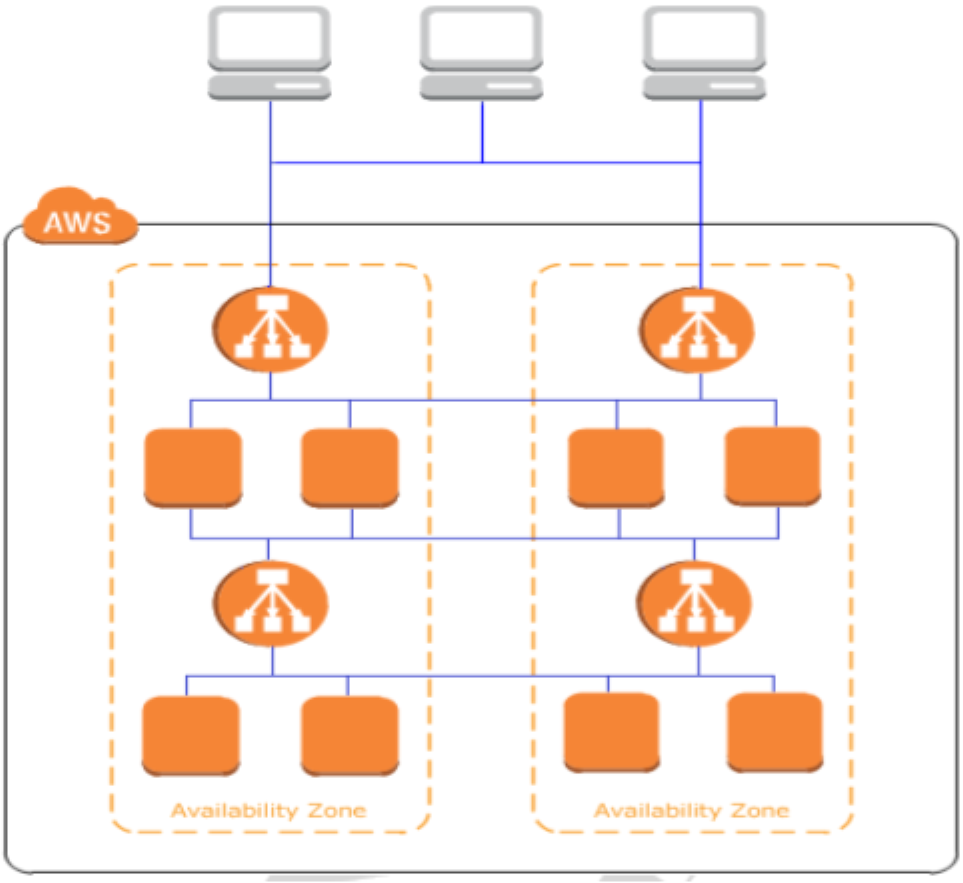
When you create a load balancer, you must choose whether to make it an internal load balancer or an Internet-facing load balancer. Note that when you create a Classic Load Balancer in EC2Classic, it must be an Internet-facing load balancer. The nodes of an Internet-facing load balancer have public IP addresses. The DNS name of an Internet facing load balancer is publicly resolvable to the public IP addresses of the nodes. Therefore, Internet facing load balancers can route requests from clients over the Internet.

The nodes of an internal load balancer have only private IP addresses. The DNS name of an internal load balancer is publicly resolvable to the private IP addresses of the nodes. Therefore, internal load balancers can only route requests from clients with access to the VPC for the load balancer.

Note: Both Internet-facing and internal load balancers route requests to your instances using private IP addresses. Therefore, your instances do not need public IP addresses to receive requests from an internal or an Internet-facing load balancer.

If your application has multiple tiers, for example web servers that must be connected to the Internet and database servers that are only connected to the web servers, you can design an architecture that uses both internal and Internet-facing load balancers. Create an Internet-facing load balancer and register the web servers with it. Create an internal load balancer and register the database servers with it.

The web servers receive requests from the Internet-facing load balancer and send requests for the database servers to the internal load balancer. The database servers receive requests from the internal load balancer.



Creating Elastic Load Balancer:

Prerequisites:

- Choose any two Availability Zones you will use for your EC2 instances. Verify that your virtual private cloud (VPC) has at least one public subnet in each of these Availability Zones.

- Launch at least one EC2 instance in each Availability Zone.

- Ensure that the security group for your EC2 instances allows HTTP access on port 80. To test the web server, copy the DNS name of the instance and verify whether browser displays the default page of the web server or not.

The below screenshot shows the instance created named elbtestproj-web1, it has a public IP address but this IP is dynamic and changes after every reboot of the instance. We need to assign an Elastic IP to this instance which is static and does not change.

Elastic IP:

An Elastic IP address is a static IPv4 address designed for dynamic cloud computing. An Elastic IP address is associated with your AWS account. With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.

Assigning Elastic IP:

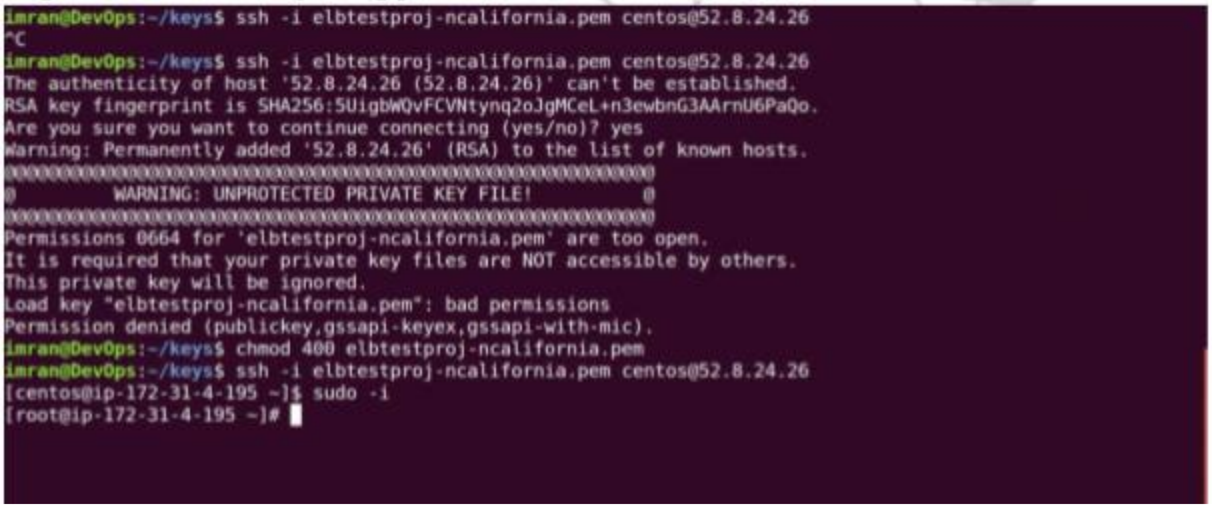
Navigate to the Elastic IPs tab on the EC2 Dashboard and click on allocate new address.

You will get a message ‘New address request succeeded’

Click on actions and select Associate address

Provide the instance id and click on associate. Then your instance will be allocated with this elastic IP

Login to the instance by using public IP



```
imran@DevOps:~/keys$ ssh -i elbtestproj-ncalifornia.pem centos@52.8.24.26
^C
imran@DevOps:~/keys$ ssh -i elbtestproj-ncalifornia.pem centos@52.8.24.26
The authenticity of host '52.8.24.26 (52.8.24.26)' can't be established.
RSA key fingerprint is SHA256:5UigbWQvFCVNtynq2oJgMCeL+n3ewbnG3AArnU6PaQo.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '52.8.24.26' (RSA) to the list of known hosts.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@         WARNING: UNPROTECTED PRIVATE KEY FILE!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
Permissions 0664 for 'elbtestproj-ncalifornia.pem' are too open.
It is required that your private key files are NOT accessible by others.
This private key will be ignored.
Load key "elbtestproj-ncalifornia.pem": bad permissions
Permission denied (publickey,gssapi-keyex,gssapi-with-mic).
imran@DevOps:~/keys$ chmod 400 elbtestproj-ncalifornia.pem
imran@DevOps:~/keys$ ssh -i elbtestproj-ncalifornia.pem centos@52.8.24.26
[centos@ip-172-31-4-195 ~]$ sudo -i
[root@ip-172-31-4-195 ~]#
```

Create centos/6 EC2 instance.

Before that just create a new key pair and new security group and assign those to new EC2 when creating.

Install apache service by using below command

# yum install httpd

Start apache service

# service httpd start

Enable apache service

# chkconfig httpd on

Stop & Disable firewall

# service iptables stop

# chkconfig iptables off

Create a test webpage for apache using html.

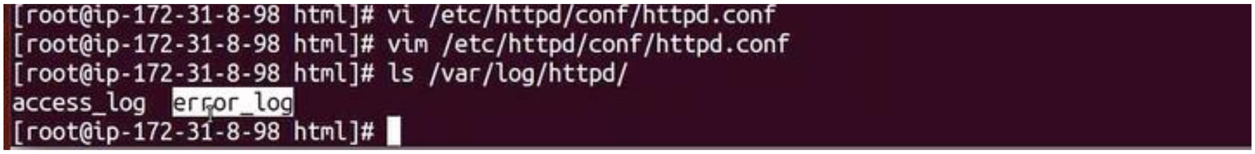
cd /var/www/html/ vi index.html

/var/www/html location is the default location to serve the web pages. Here we have an config file as well

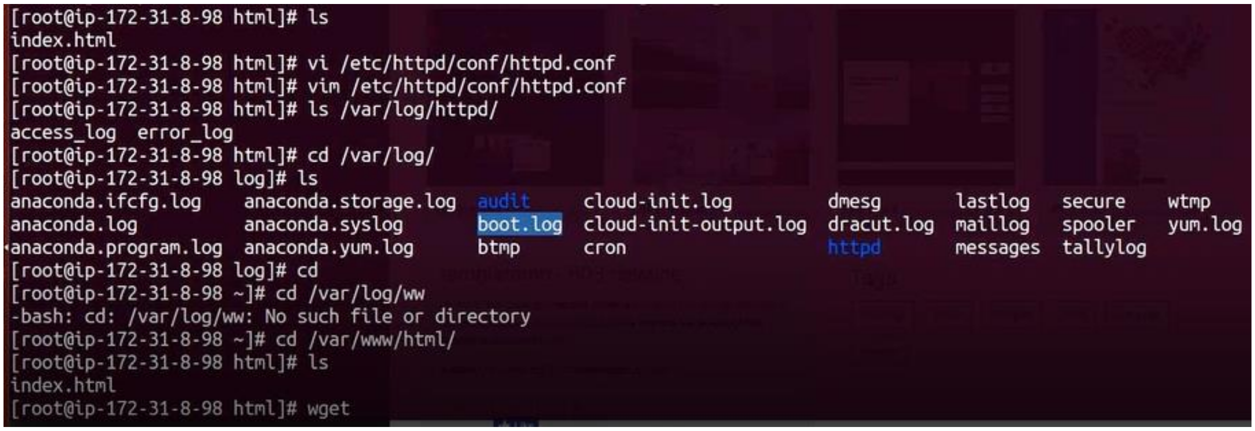
Vim /etc/httpd/conf/httpd.conf   -- all config about our apache server we have books to know this

We are going to have log files reg this apache and if we want to change any configuration we can change here like we can document root dir called /var/www/html location also.

Cd /var/log/httpd   -- we get starting point to get trouble shoot when we have an error to start service



```
[root@ip-172-31-8-98 html]# vi /etc/httpd/conf/httpd.conf
[root@ip-172-31-8-98 html]# vim /etc/httpd/conf/httpd.conf
[root@ip-172-31-8-98 html]# ls /var/log/httpd/
access_log  error_log
[root@ip-172-31-8-98 html]# 
```



```
[root@ip-172-31-8-98 html]# ls
index.html
[root@ip-172-31-8-98 html]# vi /etc/httpd/conf/httpd.conf
[root@ip-172-31-8-98 html]# vim /etc/httpd/conf/httpd.conf
[root@ip-172-31-8-98 html]# ls /var/log/httpd/
access_log  error_log
[root@ip-172-31-8-98 html]# cd /var/log/
[root@ip-172-31-8-98 log]# ls
anaconda.ifcfg.log    anaconda.storage.log  audit     cloud-init.log         dmesg       lastlog   secure    wtmp
anaconda.log          anaconda.syslog       boot.log  cloud-init-output.log  dracut.log  maillog   spooler   yum.log
anaconda.program.log  anaconda.yum.log      btmp      cron                   httpd       messages  tallylog
[root@ip-172-31-8-98 log]# cd
[root@ip-172-31-8-98 ~]# cd /var/log/ww
-bash: cd: /var/log/ww: No such file or directory
[root@ip-172-31-8-98 ~]# cd /var/www/html/
[root@ip-172-31-8-98 html]# ls
index.html
[root@ip-172-31-8-98 html]# wget
```

If we see any error we have to go to log folder and check for log file then only we can able to resolve the issue.

ls

Cd /var/www/html

<html>

<head>

<body>

<h1>This is a test webpage.</h1>

</body>

</head>

</html>

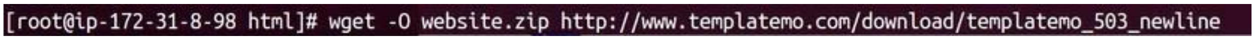
Or

Go to www.tooplate.com and download one static website and put into our server.

cd /var/www/html

sudo yum install wget unzip –y

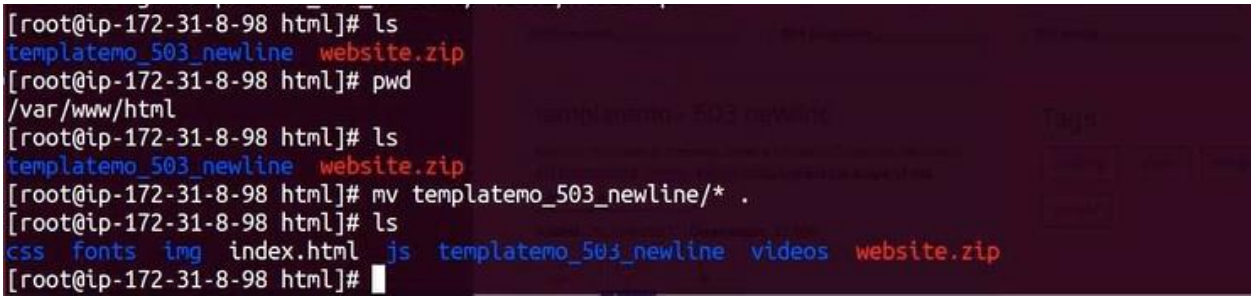
wget –O website.zip <url from this site>



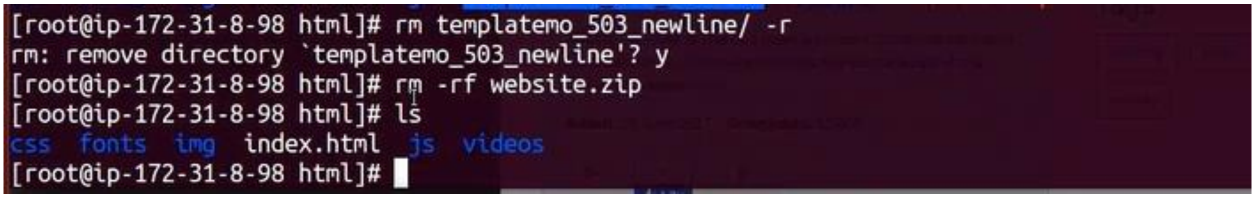
```
[root@ip-172-31-8-98 html]# wget -O website.zip http://www.templatemo.com/download/templatemo_503_newline
```

unzip website.zip

mv <dir name>/* .



```
[root@ip-172-31-8-98 html]# ls
templatemo_503_newline  website.zip
[root@ip-172-31-8-98 html]# pwd
/var/www/html
[root@ip-172-31-8-98 html]# ls
templatemo_503_newline  website.zip
[root@ip-172-31-8-98 html]# mv templatemo_503_newline/* .
[root@ip-172-31-8-98 html]# ls
css  fonts  img  index.html  js  templatemo_503_newline  videos  website.zip
[root@ip-172-31-8-98 html]# 
```



```
[root@ip-172-31-8-98 html]# rm templatemo_503_newline/ -r
rm: remove directory `templatemo_503_newline'? y
[root@ip-172-31-8-98 html]# rm -rf website.zip
[root@ip-172-31-8-98 html]# ls
css  fonts  img  index.html  js  videos
[root@ip-172-31-8-98 html]# 
```

Now just restart our web server.

sudo service apache2 stop

sudo service apache2 start

sudo service apache2 status

Test the webpage Enter the ec2 inst public IP in browser.

http://52.8.24.26:8080

We are able to see our static website into our server and it is better than a static text on our website.

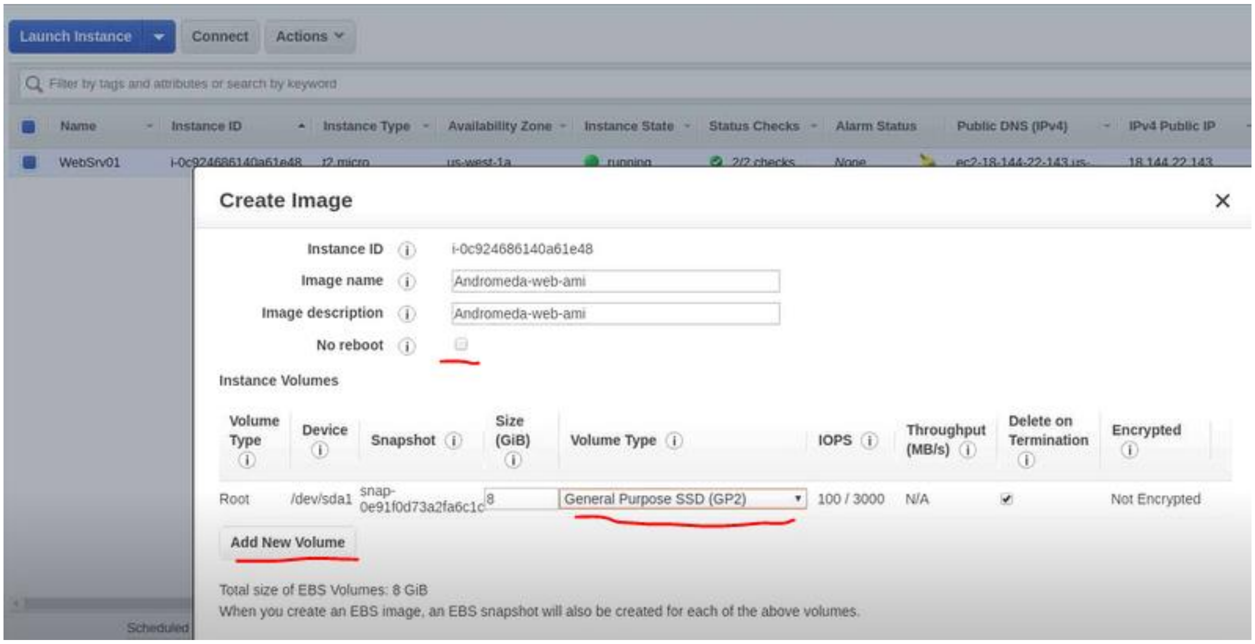
AWS AMI:

An Amazon Machine Image (AMI) is a special type of virtual appliance that is used to create a virtual machine within the Amazon Elastic Compute Cloud ("EC2"). It serves as the basic unit of deployment for services delivered using EC2.

AMI Creation:

Create AMI of the instance which we will use to spin web02 instance. Web02 instance is exactly similar to web01, so instead to creating new instance from scratch and setting up apache, we can create an AMI (image) of web01 instance and can spin as many as web instances we want. Select the instance in which we have to create an image. Click on actions, select image and click on create image.

- You will get a create image dialog box as shown below. Give proper name and description for image and click on create image
- It takes few minutes to create an image which you can see on AMIs navigation pane



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP
WebSrv01	i-0c924686140a61e48	t2.micro	us-west-1a	running	2/2 checks	None	ec2-18-144-22-143.us-	18.144.22.143

Create Image ✕

Instance ID: i-0c924686140a61e48
Image name: Andromeda-web-ami
Image description: Andromeda-web-ami
No reboot: □

Instance Volumes

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0e91f0d73a2fa6c1c	8	General Purpose SSD (GP2)	100 / 3000	N/A	☑	Not Encrypted

Total size of EBS Volumes: 8 GiB
When you create an EBS image, an EBS snapshot will also be created for each of the above volumes.

Create web02 instance from elbtestproj-web-ami.

Click on Launch instance --> My AMI --> Select your AMI --> Follow the wizard and create the instance similar to web01.

Tag Name: elbtestproj-web02 Security: Select existing security group --> elbtestproj-SG Select an existing key pair --> <same key used for web01> Assign Elastic IP

Test the webpage Enter the ec2 instance public IP in browser.

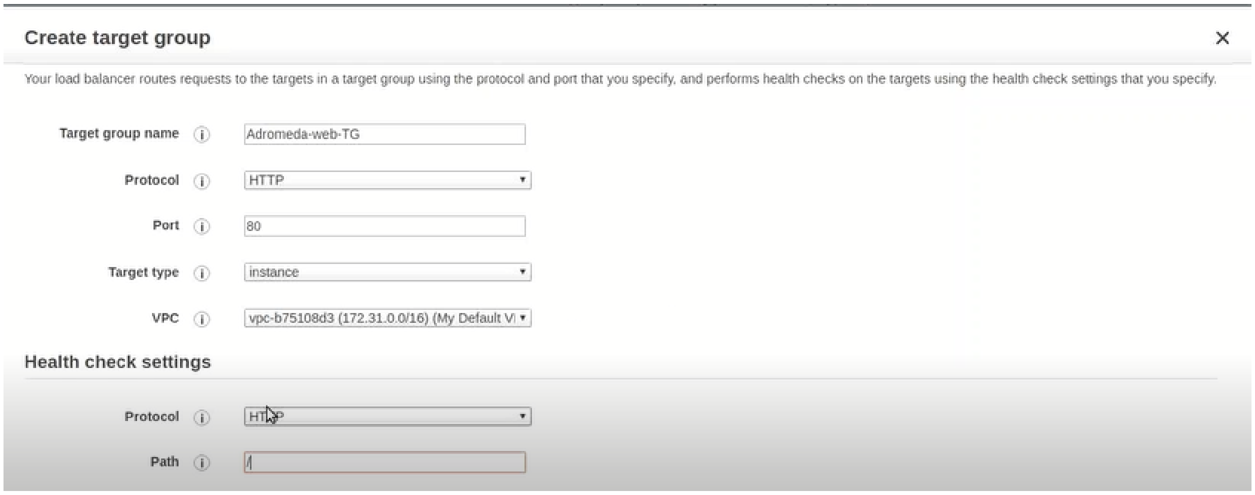
http://54.215.191.250//

It is just a snapshot, but snapshot is all about volume where as AMI is all about complete EC2 instance.

Load Balancer setup :

First we have to create target group for instances and then load balancer has to create.

- Click on Target Group on the left side of the navigation pane, click on create target group.
- Provide name for target group and click on create
- Once the Target group is created click on the targets tab, Edit and Select web01 & web02 instances. Add to registered and click on Save.
- Then the instances will be added to the Target Group.



Create target group

Your load balancer routes requests to the targets in a target group using the protocol and port that you specify, and performs health checks on the targets using the health check settings that you specify.

Target group name: Adromeda-web-TG

Protocol: HTTP

Port: 80

Target type: instance

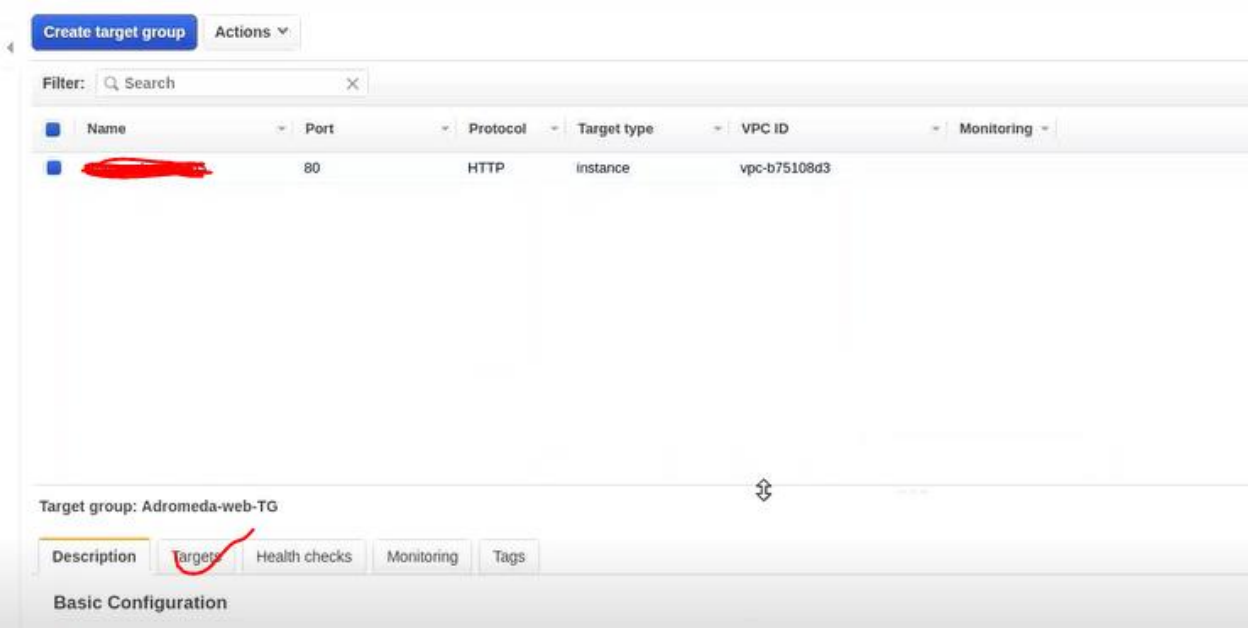
VPC: vpc-b75108d3 (172.31.0.0/16) (My Default V

Health check settings

Protocol: HTTP

Path: /

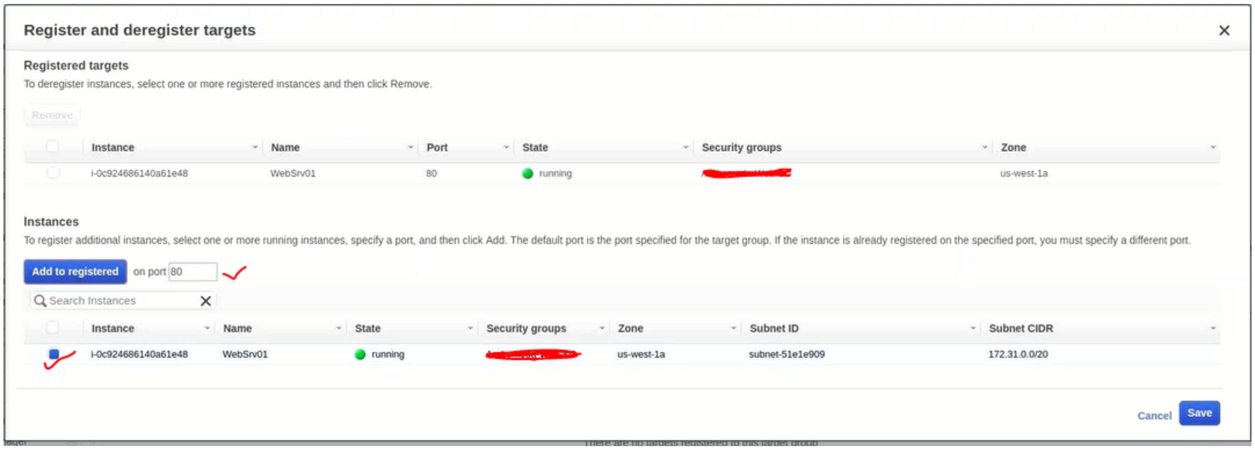
Now Target group is created and we have to add instances under target group here



Name	Port	Protocol	Target type	VPC ID	Monitoring
[redacted]	80	HTTP	instance	vpc-b75108d3	

Target group: Adromeda-web-TG

Basic Configuration



Register and deregister targets

Registered targets

To deregister instances, select one or more registered instances and then click Remove.

Instance	Name	Port	State	Security groups	Zone
i-0c924686140a61e48	WebSrv01	80	● running	[redacted]	us-west-1a

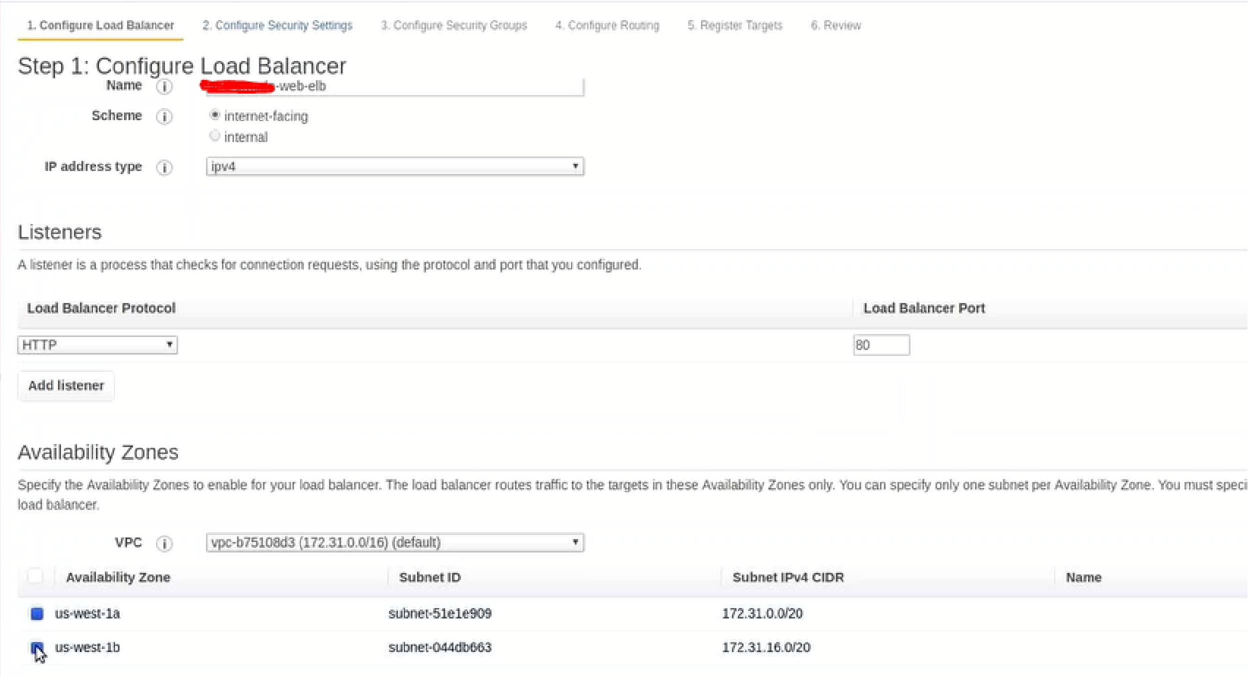
Instances

To register additional instances, select one or more running instances, specify a port, and then click Add. The default port is the port specified for the target group. If the instance is already registered on the specified port, you must specify a different port.

	Instance	Name	State	Security groups	Zone	Subnet ID	Subnet CIDR
☑	i-0c924686140a61e48	WebSrv01	● running	[redacted]	us-west-1a	subnet-51e1e909	172.31.0.0/20

Added one instance and one mre we have to create from AMI and need to add here.

Creating Load Balancer:



1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 1: Configure Load Balancer

Name [redacted]-web-elb

Scheme ● internet-facing ○ internal

IP address type ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

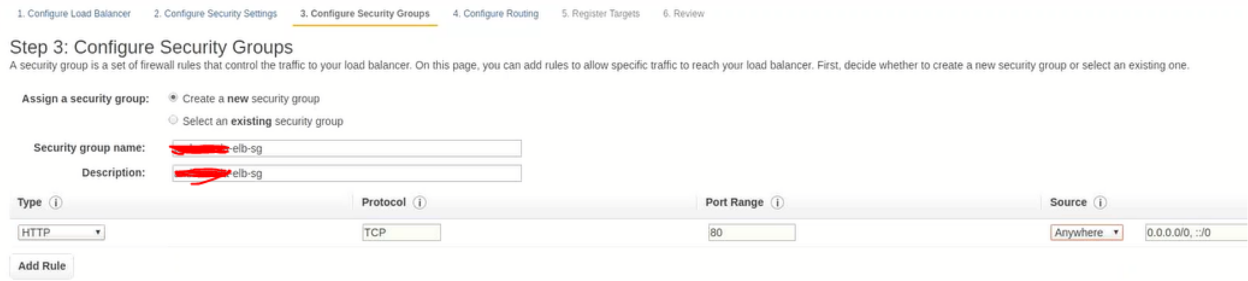
Load Balancer Protocol	Load Balancer Port
HTTP	80

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You must speci
load balancer.

VPC vpc-b75108d3 (172.31.0.0/16) (default)

	Availability Zone	Subnet ID	Subnet IPv4 CIDR	Name
☑	us-west-1a	subnet-51e1e909	172.31.0.0/20	
☑	us-west-1b	subnet-044db663	172.31.16.0/20	



1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 3: Configure Security Groups

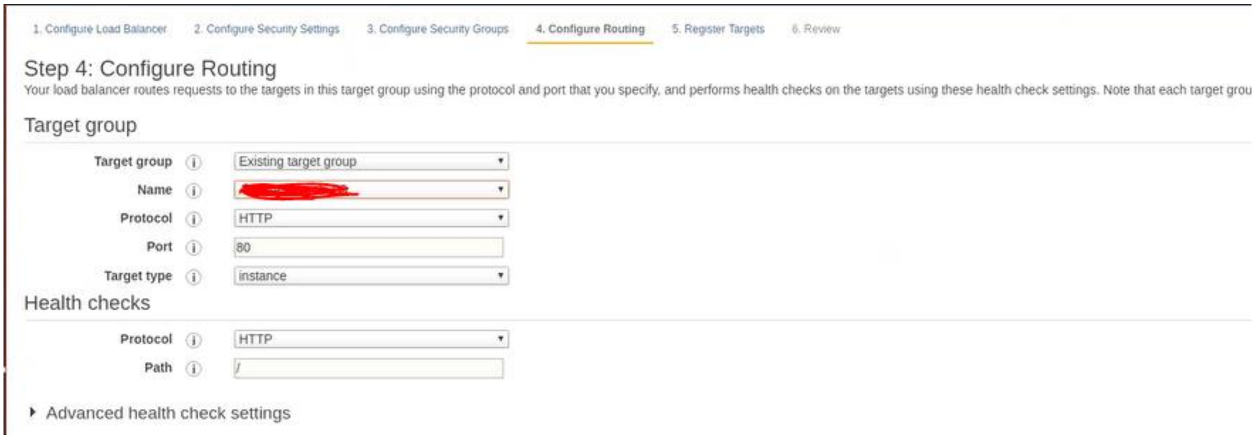
A security group is a set of firewall rules that control the traffic to your load balancer. On this page, you can add rules to allow specific traffic to reach your load balancer. First, decide whether to create a new security group or select an existing one.

Assign a security group: ● Create a **new** security group ○ Select an **existing** security group

Security group name: [redacted]-elb-sg

Description: [redacted]elb-sg

Type	Protocol	Port Range	Source	
HTTP	TCP	80	Anywhere	0.0.0.0/0, ::/0



1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review

Step 4: Configure Routing

Your load balancer routes requests to the targets in this target group using the protocol and port that you specify, and performs health checks on the targets using these health check settings. Note that each target grou

Target group

Target group Existing target group

Name [redacted]

Protocol HTTP

Port 80

Target type instance

Health checks

Protocol HTTP

Path /

And Register and create. Done

Here am going to create one more instance from AMI and that is going to be add into this target group and accessing from this load balancer.

Her we can Health check tab under target group reg health check.

Click on Load Balancers in Left Pane and select Create Load Balancer, you will get two types of load balancers, Application load balancer and Classic load balancer. Select Application Load Balancer and click on Continue.

Configure the details of the load balancer. Provide the name for load balancer, the name of your Application Load Balancer must be unique within your set of Application Load Balancers. For one region, it can have a maximum of 32 characters and contain only alphanumeric characters and hyphens, and must not begin or end with a hyphen. Click on next configure security group.

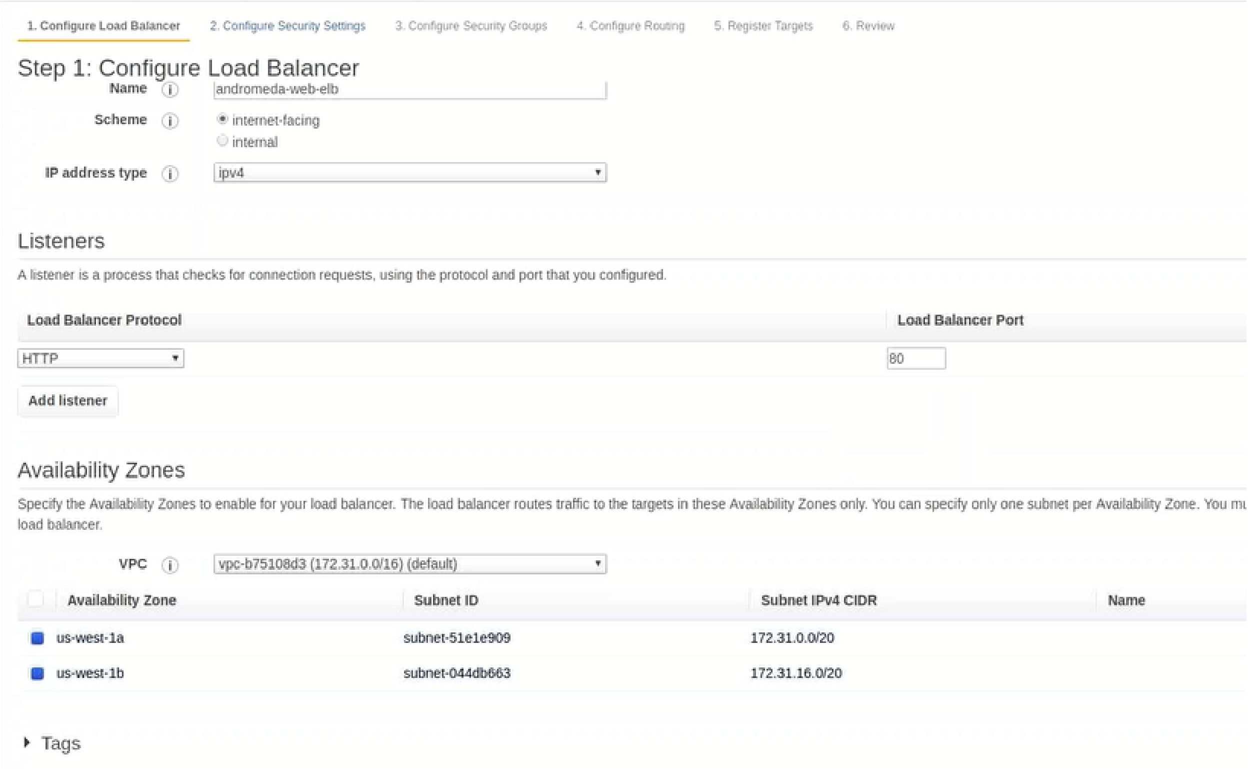
For Availability Zones, select the VPC that you used for your EC2 instances. For each of the two Availability Zones that contain your EC2 instances, select the Availability Zone and then select the public subnet for that Availability Zone. Add atleast two Availability Zones to increase the availability of the load balancer. Click on next

Configure the details of security group to improve the load balancer’s security.

Create a new security group for load balancer which accepts HTTP traffic on port 80.

Configure the target group. The default rule for your listener routes requests to the registered targets in this target group. The load balancer checks the health of targets in this target group using the health check settings defined for the target group.

Register targets with the Target Groups


1. Configure Load Balancer
2. Configure Security Settings
3. Configure Security Groups
4. Configure Routing
5. Register Targets
6. Review
Add listener
Tags

Step 1: Configure Load Balancer

Name: andromeda-web-elb

Scheme: ● internet-facing ○ internal

IP address type: ipv4

Listeners

A listener is a process that checks for connection requests, using the protocol and port that you configured.

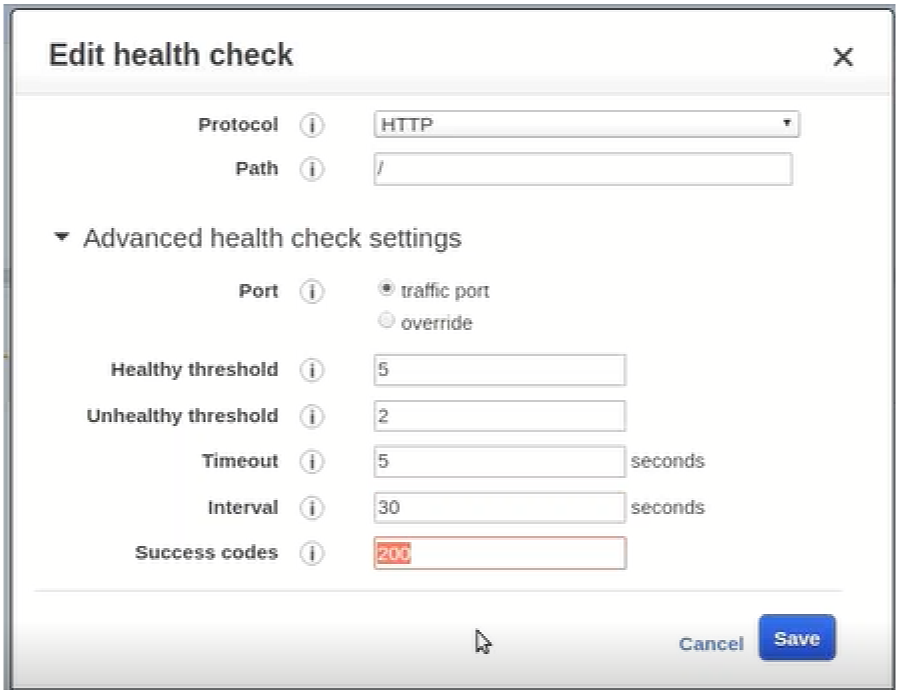
Load Balancer Protocol	Load Balancer Port
HTTP	80

Availability Zones

Specify the Availability Zones to enable for your load balancer. The load balancer routes traffic to the targets in these Availability Zones only. You can specify only one subnet per Availability Zone. You mu
load balancer.

VPC: vpc-b75108d3 (172.31.0.0/16) (default)

□	Availability Zone	Subnet ID	Subnet IPv4 CIDR	Name
■	us-west-1a	subnet-51e1e909	172.31.0.0/20	
■	us-west-1b	subnet-044db663	172.31.16.0/20	



Edit health check

Protocol: HTTP

Path: /

▾ Advanced health check settings

Port: ● traffic port ○ override

Healthy threshold: 5

Unhealthy threshold: 2

Timeout: 5 seconds

Interval: 30 seconds

Success codes: 200

This is all about Health check about target group instances.