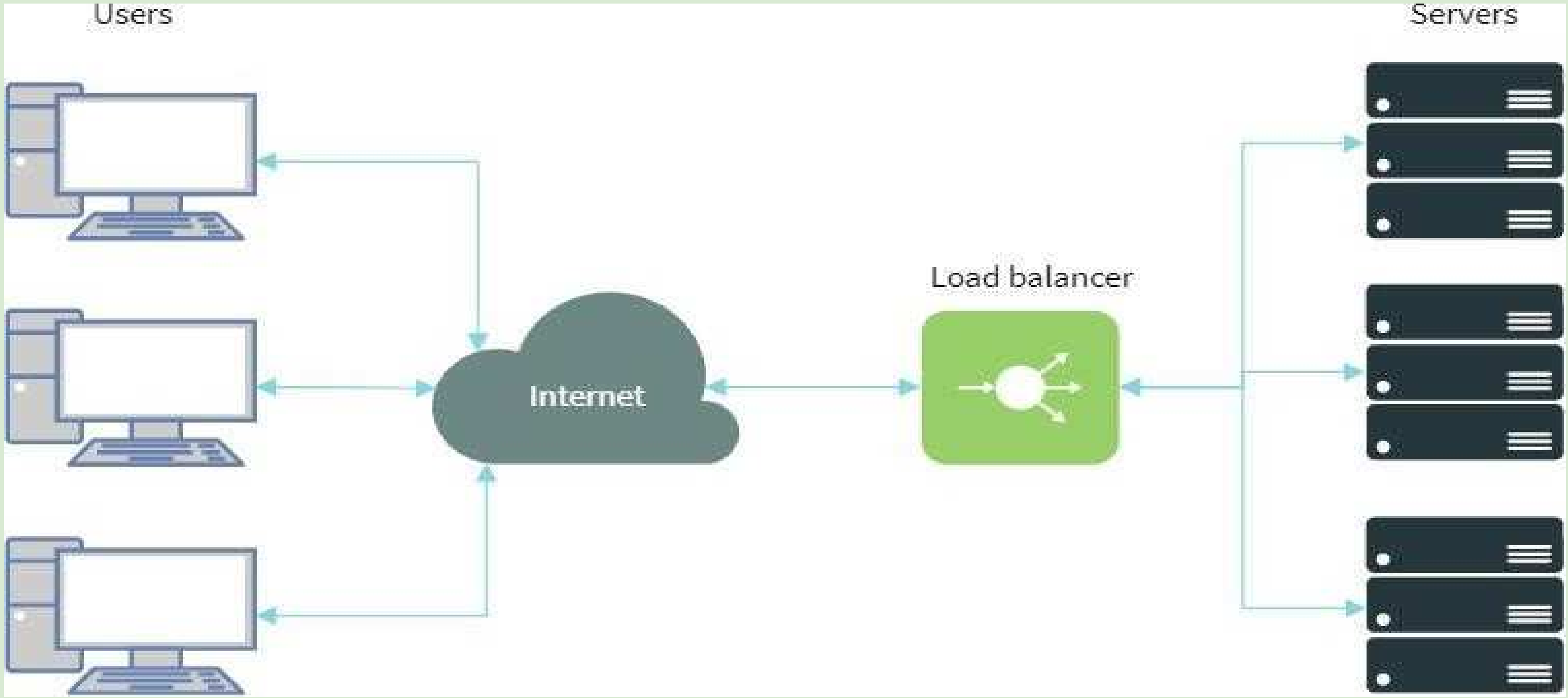
Amazon Web Services

Elastic Load Balancer

Elastic Load Balancer

- An Elastic Load Balancer is a service which uniformly distributes network traffic and workloads across multiple servers

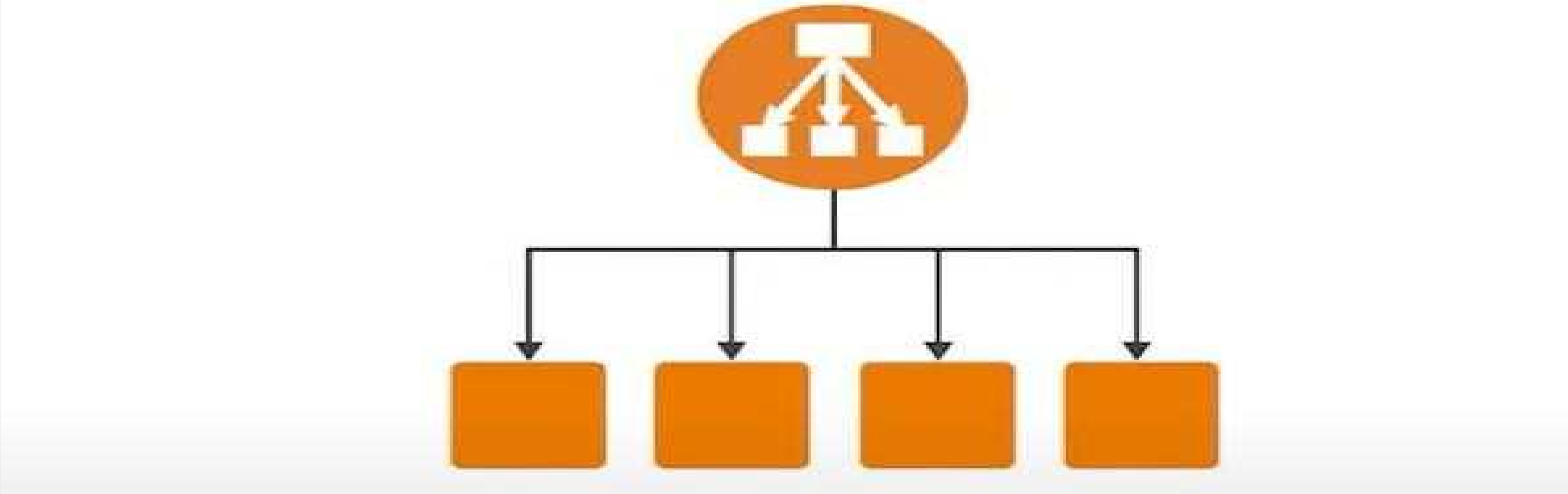


Types of Load Balancer

- Classic Load Balancer
- Application Load Balancer
- Network Load Balancer
- Gateway Load Balancer

Classic Load Balancer

- A Classic load balancer distributes equally incoming application traffic across multiple EC2 instances in multiple Availability Zones. This increases the fault tolerance of your applications. Elastic Load Balancing detects unhealthy instances and routes traffic only to healthy instances.

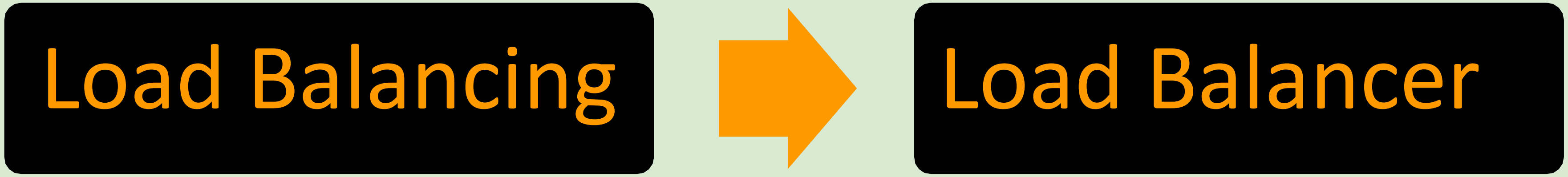


Classic Load Balancer

- Create Linux EC2 Machine
- Bootstrap Script Code: **Click Here**
- Enable HTTP Port
- Create Second Linux EC2 Machine
- Bootstrap Script Code: **Click Here**
- Enable HTTP Port

Classic Load Balancer

- Create Load balancer



Load Balancing

Load Balancer

- Click on Create for Classic Load Balancer
- Define Load Balancer
- Assign Security Group
- Configure Security Settings

Classic Load Balancer

- Configure Health Check (Important)

- Enter Response timeout as 2

- Enter Interval as 5

- Enter Unhealthy threshold as 2

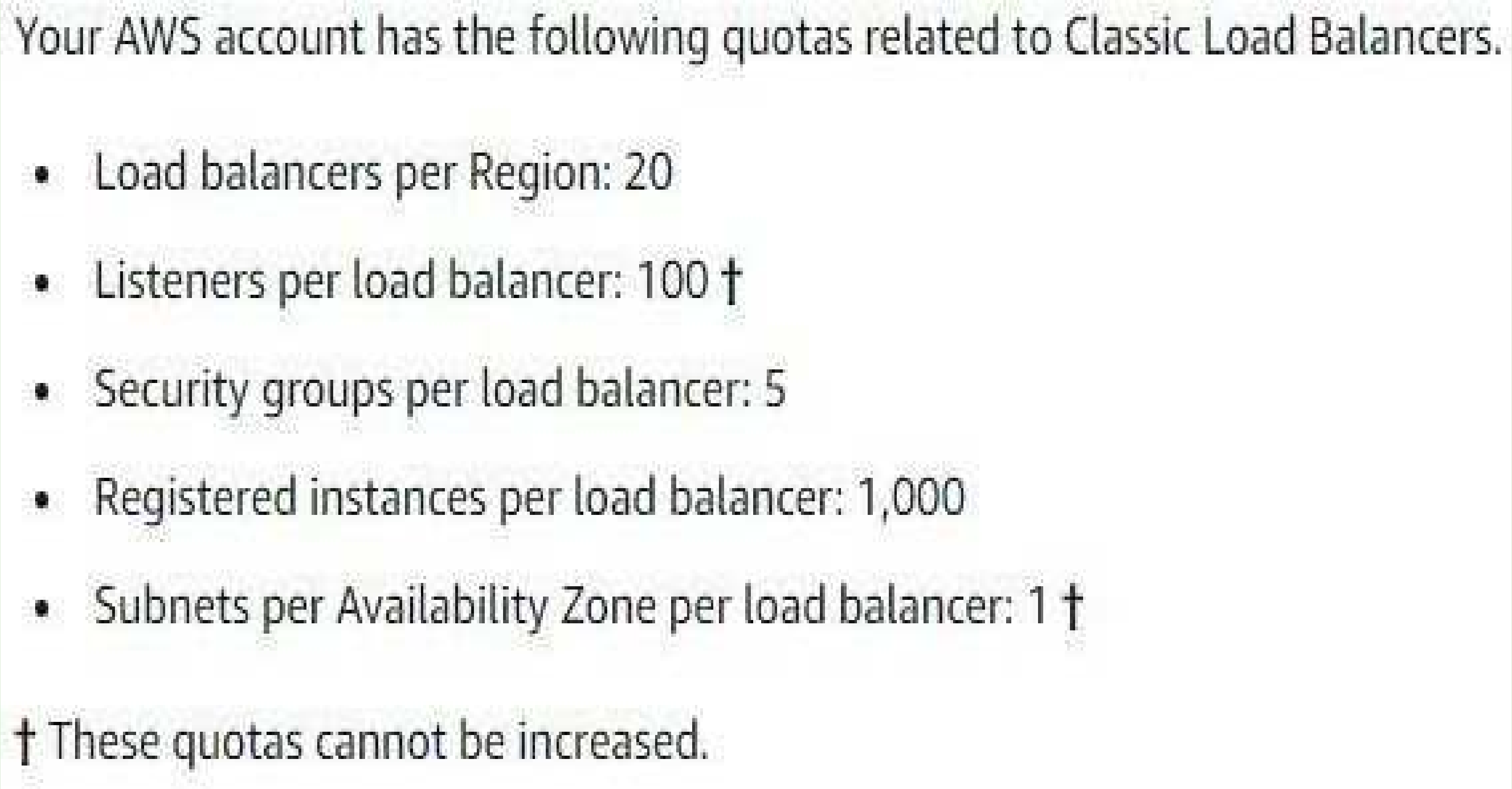
- Enter healthy threshold as 2

Classic Load Balancer

- Add EC2 Instances
- Add Tags
- Review
- Click on Create
- Copy DNS Name & Paste in the browser
- Stop First EC2 Machine

Quotas for your Classic Load Balancer:

- **Document:** **Click Here**



Your AWS account has the following quotas related to Classic Load Balancers.

- Load balancers per Region: 20
- Listeners per load balancer: 100 †
- Security groups per load balancer: 5
- Registered instances per load balancer: 1,000
- Subnets per Availability Zone per load balancer: 1 †

† These quotas cannot be increased.

Classic Load Balancer:

- The load balancer also monitors the health of its registered targets and ensures that it routes traffic only to healthy targets. When the load balancer detects an unhealthy target, it stops routing traffic to that target. It then resumes routing traffic to that target when it detects that the target is healthy again.

- Example. You enter a retail store, pick up things you need to buy and approach the billing counters. You see, there are 3 counters that are open and all 3 of them has a very long queue. The store manager sees you and other customers getting annoyed. He decides to open the other two counters. Now the load on those three counters gets reduced and eventually divided amongst the five counters. This makes the customers happy and reduces the cashier’s workload. This is the concept of load balancing.

Classic Load Balancer:

- **Response Timeout**: The amount of time to wait when receiving a response from the health check, in seconds. Valid values: 2 to 60

- **Interval**: The amount of time between health checks of an individual instance, in seconds. Valid values: 5 to 300

- **Unhealthy Threshold**: The number of consecutive failed health checks that must occur before declaring an EC2 instance unhealthy. Valid values: 2 to 10

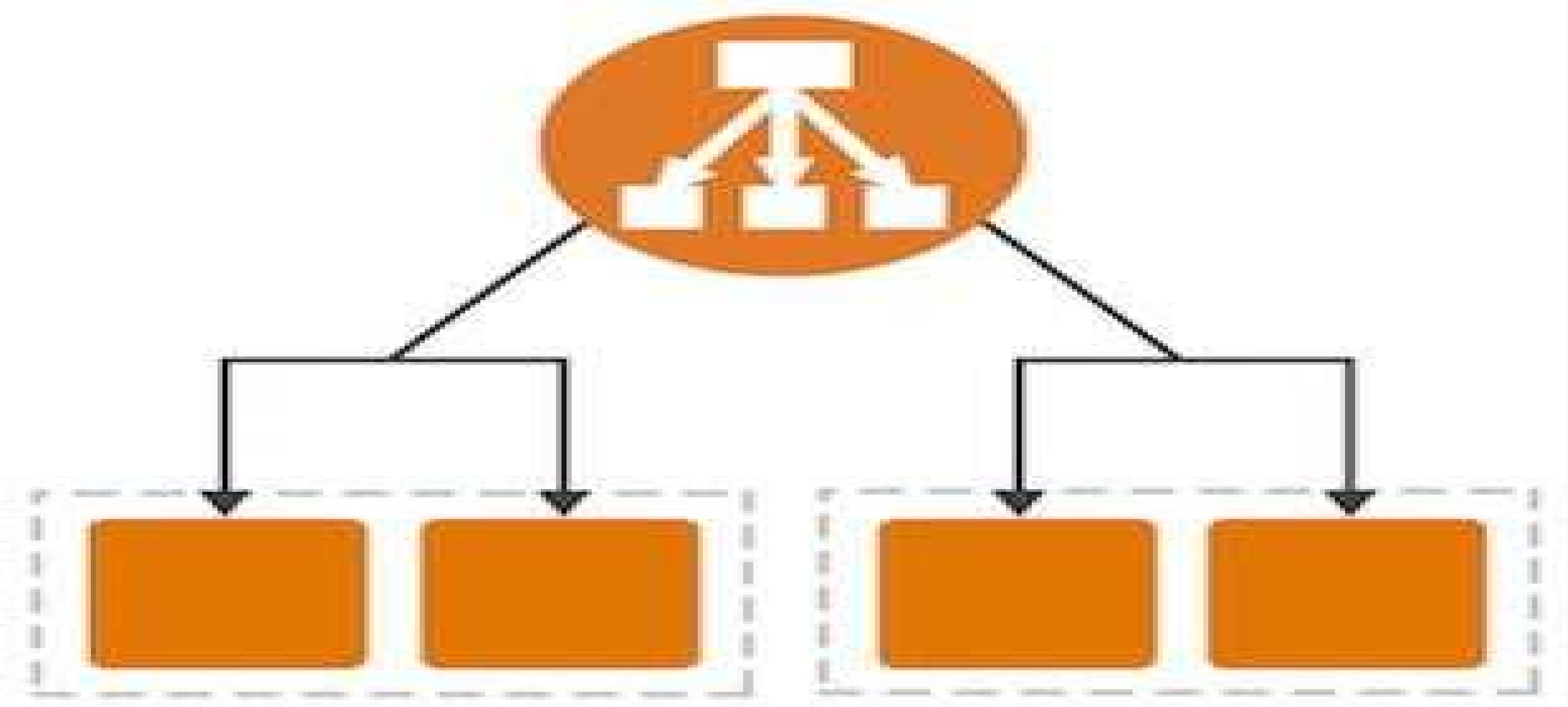
- **Healthy Threshold:** The number of consecutive successful health checks that must occur before declaring an EC2 instance healthy. Valid values: 2 to 10

Amazon Web Services

Application Load Balancer

Application Load Balancer

- Application Load Balancer routing traffic to targets based on the content of the request.



Application Load Balancer

- Create Windows EC2 Machine
- Enable HTTP Port
- Install Web server
- Create html file in wwwroot folder
- Create one more folder & create the html file

Application Load Balancer

- Create Second Windows EC2 Machine

- Enable HTTP Port

- Install Web server

- Create html file in wwwroot folder

- Create one more folder & create the html file

Application Load Balancer

- Create Target Group

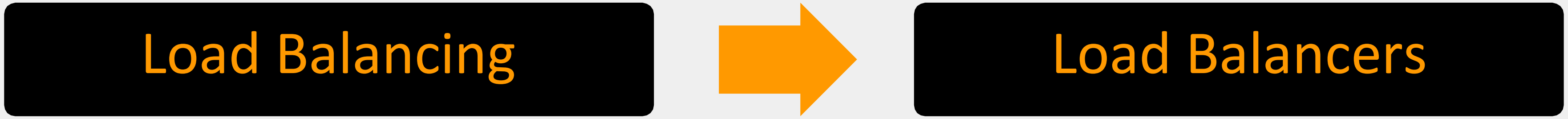
Load Balancing

Target Groups

- Click on Create Target Group
- Enter the Name of the Target Group Name
- Click on Advanced health Check Settings
- Click on Next
- Select the machine & Click on Include as pending below
- Click on Create target group
- Create one more target group for second machine

Application Load Balancer

- Create Application Load Balancer



Load Balancing

Load Balancers

- Click on Create Load Balancer
- Click on Create for application Load balancer
- Enter the Name
- Select all Availability zone
- Select our security group
- Select Default Target Group (First Target Group)
- Click on Create Load Balancer

Application Load Balancer

- Now check our target group
- Go to our Application load Balancer
- Go to Listeners tab
- Select the listener
- Click on edit
- Select the second target group
- Click on save changes
- Click on View/edit rules
- Click on Add Rules

Application Load Balancer

- Click on Insert rule
- Select the condition as Path
- Enter the path & forward to first machine
- Add another Rule
- For the second machine rule it’s mandatory to mention * after the path
- Save the Load Balancer
- Copy the DNS and Paste in Browser
- Change the path as per the folders we have created in the webservers

Application Load Balancer

- Listener: A listener is a process that checks for connection requests.

- **Response Timeout**: The amount of time to wait when receiving a response from the health check, in seconds. Valid values: 2 to 60

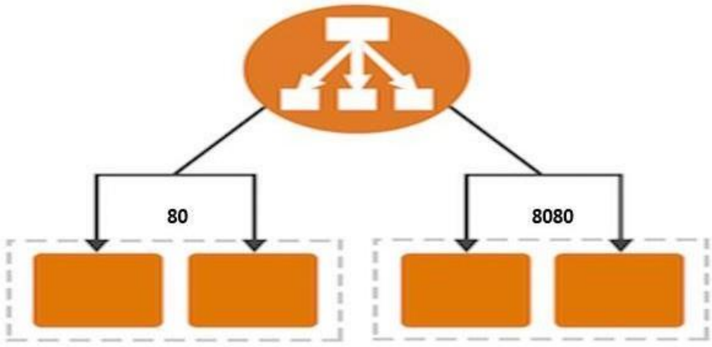
- **Interval**: The amount of time between health checks of an individual instance, in seconds. Valid values: 5 to 300

- **Unhealthy Threshold**: The number of consecutive failed health checks that must occur before declaring an EC2 instance unhealthy. Valid values: 2 to 10

- **Healthy Threshold:** The number of consecutive successful health checks that must occur before declaring an EC2 instance healthy. Valid values: 2 to 10

Network Load Balancer

- Network Load Balancer routing traffic to targets based on the port number. It has the capability to respond to millions of request every second.



- Create First Linux EC2 Machine
- Bootstrap Script Code: **Click Here**
- Enable the Type as HTTP
- Create Second Linux EC2 Machine
- Bootstrap Script Code: **Click Here**
- Enable the Type as HTTP
- Enable Custom TCP Type & Change the Port Range to 8080
- Now Check the Public IP Address for both the Machines
- Add :8080 Port number for Second EC2 Machine
- Connect Second EC2 Machine using Putty
- Change the User

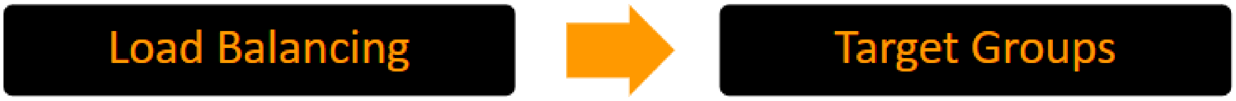
  sudo su
- Change folder

  cd /etc/httpd/conf
- Edit httpd.conf file

  vi httpd.conf
- Convert file into insert mode by pressing i
- Change Listen to 8080
- Save the file Press ESC

  :wq!

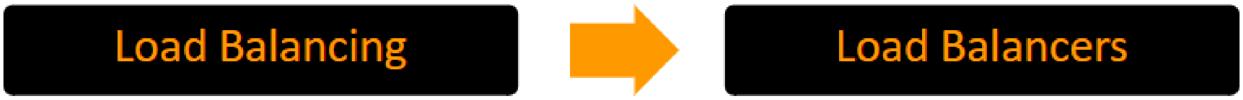
- Restart the server service httpd restart
- Create Target Group



Load Balancing

Target Groups

- Click on Create target group
- Enter the name of Target Group Name
- Change Protocol to TCP & Port number needs to be 80 for the first Machine
- Click on Next
- Add first Machine as Target
- Click on Create target group
- Create Target group for Second Machine Enter port number as 8080
- Create Network Load balancer



Load Balancing

Load Balancers

- Click on Create For Network Load balancer
- Enter the name of Load balancer
- Map all the subnet
- Select Listener & Target group
- Click on Create load balancer
- Now check our load balancer
- Go to Listeners tab
- Click on Add Listener
- Change port number to 8080
- Add Action to Forward to
- Select Second Machine
- Click on Add Listener
- Now Copy the DNS Name
- We can select the appropriate load balancer based on our application needs. If we need flexible application management, we recommend that we use an Application Load Balancer. If extreme performance and static IP is needed for our application, we

recommend that we use a Network Load Balancer. If we have an existing application that was built within the EC2- Classic network, then we should use a Classic Load Balancer.

- Click Here