CLOUD CONCEPTS

(EC2 Launch Types)

- On Demand Instances: short workload, predictable pricing
- Reserved Instances: long workloads (>= 1 year)
- Convertible Reserved Instances: long workloads with flexible instances
- Scheduled Reserved Instances: launch within time window you reserve
- Spot Instances: short workloads, for cheap, can lose instances
- Dedicated Instances: no other customers will share your hardware
- Dedicated Hosts: book an entire physical server, control instance placement

- Whenever server is launched the default instance is **On Demand Instance**. It is the most expensive as it is available immediately. **Reserved instances** are recommended for database usage whenever usage is 1 – 3 years. Types of reserved instances are **convertible** and **scheduled**. In **convertible** type of server can be changed e.g., T2 micro (1cpu 1gb) can be changed to any server with higher configuration if more traffic is to be entertained while in **scheduled** the time span or time slot is defined i.e., on Monday from 8am – 10am**. Spot instances** are used for bidding. To get the requested instance the bid must be higher if any bid is higher than yours even the instances in your use will be taken and granted to the highest bidder. In all the discussed instance types of portions in big machine is allocated and are said **dedicated Instances**. **Dedicated host** means you own entire physical server even if you are running only one application.

EC2 On Demand

- Pay for what you use (billing per second, after the first minute)
- Has the highest cost but no upfront payment
- No long term commitment

- Recommended for short-term and un-interrupted workloads, where you can't predict how the application will behave

EC2 Reserved Instances

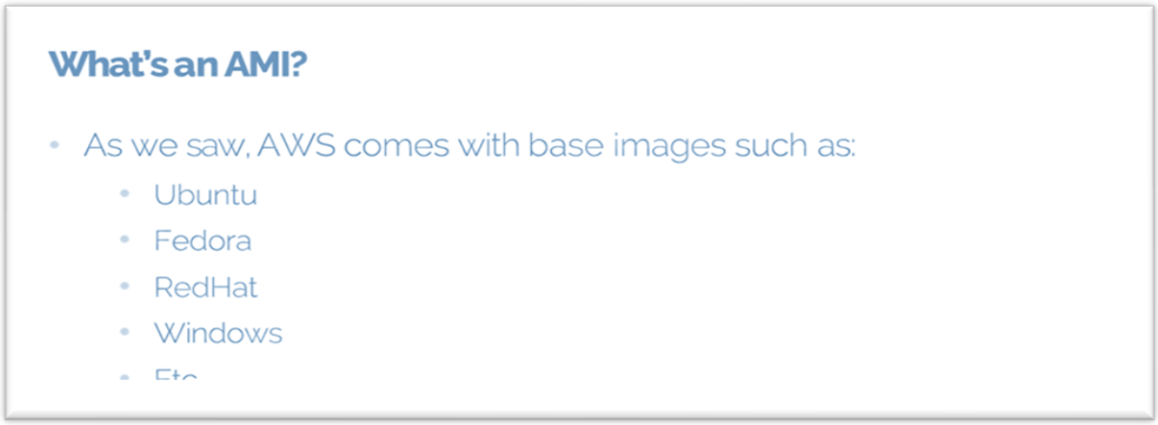
- Up to 75% discount compared to On-demand
- Pay upfront for what you use with long term commitment
- Reservation period can be 1 or 3 years
- Reserve a specific instance type
- Recommended for steady state usage applications (think database)
- **Convertible Reserved Instance**
  - can change the EC2 instance type
  - Up to 54% discount
- **Scheduled Reserved Instances**
  - launch within time window you reserve
  - When you require a fraction of day / week / month

EC2 Spot Instances

- Can get a discount of up to 90% compared to On-demand
- You bid a price and get the instance as long as its under the price
- Price varies based on offer and demand
- Spot instances are reclaimed with a 2 minute notification warning when the spot price goes above your bid
- Used for batch jobs, Big Data analysis, or workloads that are resilient to failures. Not great for critical jobs or databases

Todo: Take the overview of all the other instance types from on amazon but do not launch the server as all are paid.

Lab (Creatin AMI copy from one region to other and launching AMI copy in same region)



- Amazon machine image is base image or base AMI provided by Amazon. We simply select it and launch it with all the default services and configurations. We can create your own AMI. In this existing SG and key pair should be used. AMIs can be built for Windows and Linux machines.
- AMI are built in specific regions and have specific IDs in specific regions. AMI created in one region has ID A and same AMI created in other region will have ID B.
- AMI ID is unique.
- You can also use or rent your own created AMI with others. By default, the AMI is private it must be made public and then accessed publicly.
- Basic monitoring is free in AMAZON.

Why use AMI
Pre installed Packages
Faster boot time
Machine have basic monitoring configured
Security is managed
Maintainence, control and updates of AMI
Installing app ahead of time (auto scaling)
Using cutom made AMI otimized for running apps

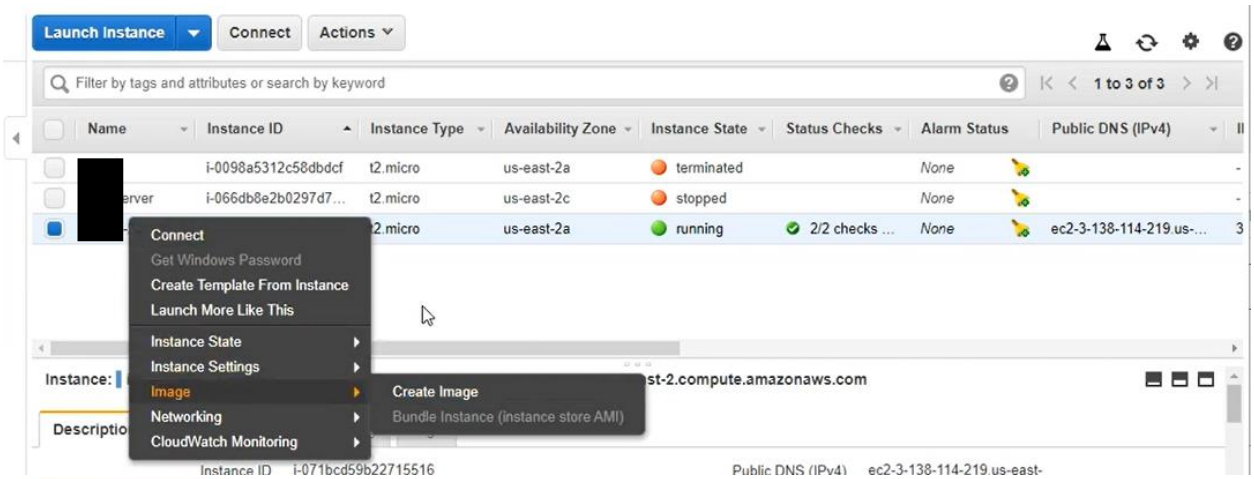
- AMIs are stored on S3 as S3 is durable, cheap, and resilient storage where most of the back-ups reside. As it is in S3 you will be charged for the space it occupies in s3. Although it is inexpensive but it is better to delete them when not required.
- By default, AMIs are private and locked for account or region. These can be made public and shared with others. They can also be placed on Amazon marketplace.

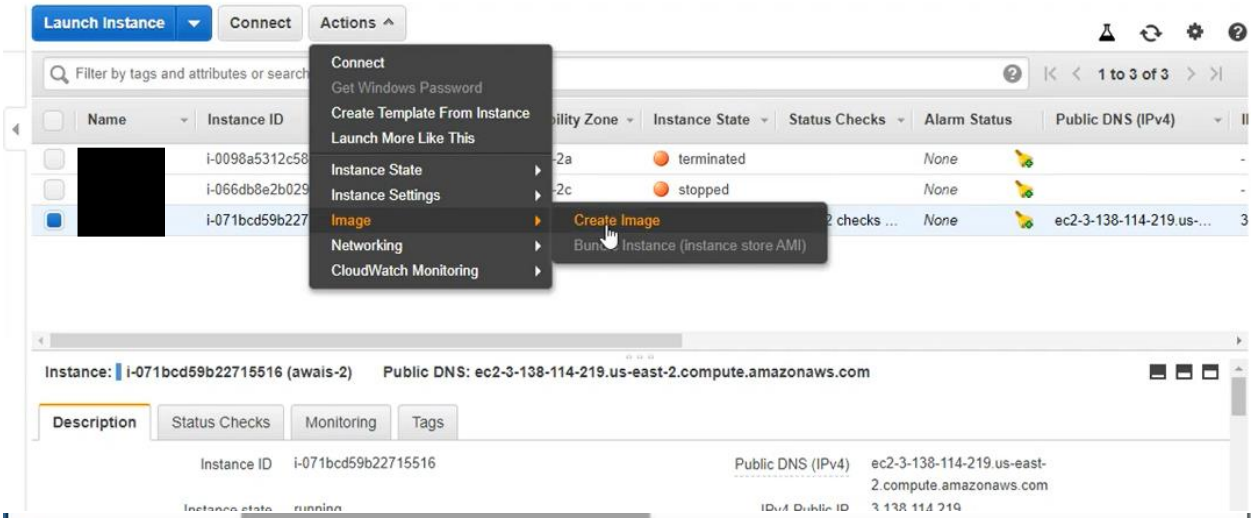
- Check the latest price on AWS marketplace.

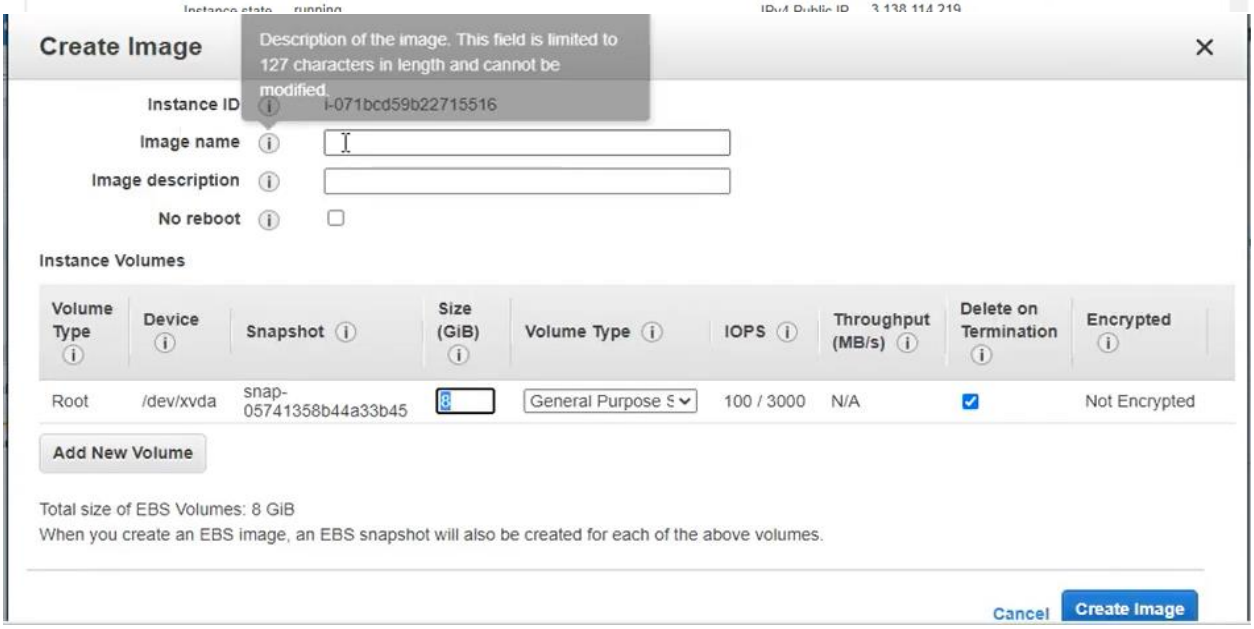
TODO: Launch the Apache server using Linux AMI with the boot script given below in user data. Make sure HTTP and SSH (Port 80 and 22) are enabled. It is recommended to access the server when the server status is checked 2/2. Save the Public IP of server and open in browser.

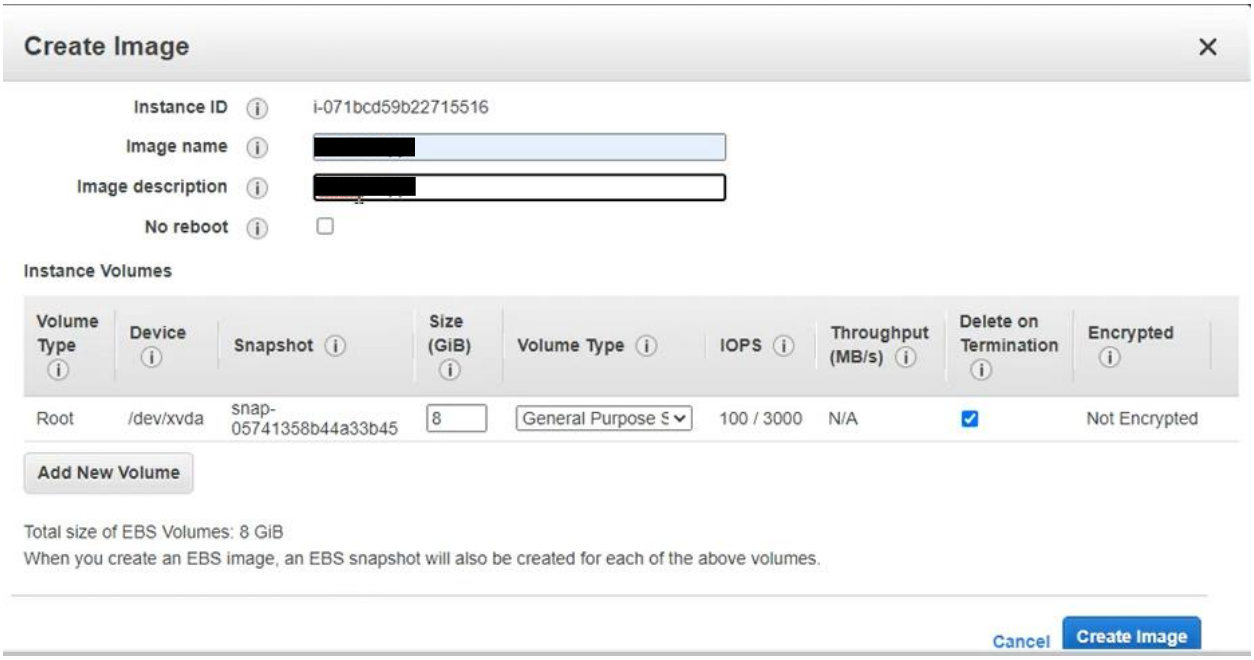
```
#!/bin/bash

#########################################################
##### USE THIS FILE IF YOU LAUNCHED AMAZON LINUX 2 #####
#########################################################

# get admin privileges
sudo su

# install httpd (Linux 2 version)
yum update -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
```

- To create the copy, go to actions and click create image. Set the image name and description and click on Create Image.

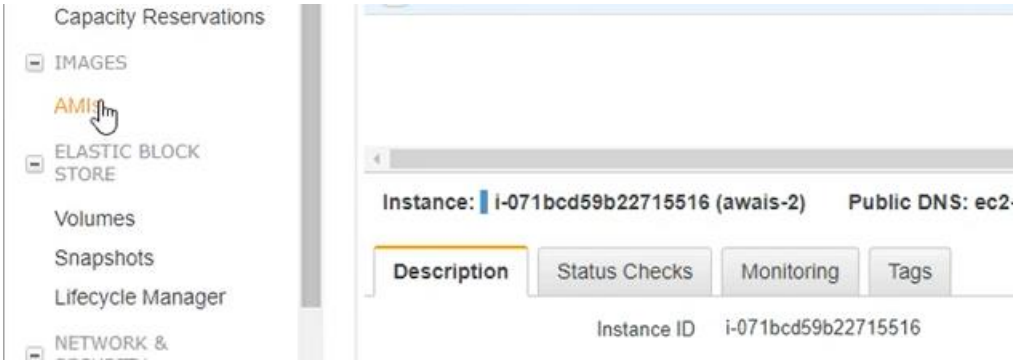

Launch Instance
Connect
Actions
Filter by tags and attributes or search by keyword
1 to 3 of 3
Name
Instance ID
Instance Type
Availability Zone
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
i-0098a5312c58dbdcf
t2.micro
us-east-2a
terminated
None
i-066db8e2b0297d7...
t2.micro
us-east-2c
stopped
None
running
2/2 checks ...
None
ec2-3-138-114-219.us-...
Connect
Get Windows Password
Create Template From Instance
Launch More Like This
Instance State
Instance Settings
Image
Networking
CloudWatch Monitoring
Create Image
Bundle Instance (instance store AMI)
Instance:
Description
Instance ID
i-071bcd59b22715516
Public DNS (IPv4)
ec2-3-138-114-219.us-east-

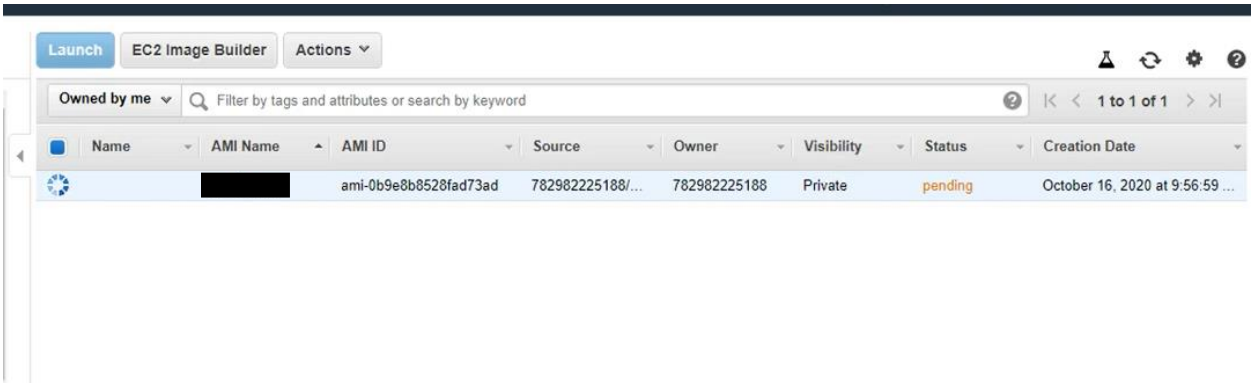

Launch Instance
Connect
Actions
1 to 3 of 3
Name
Instance ID
Instance State
Status Checks
Alarm Status
Public DNS (IPv4)
terminated
stopped
None
ec2-3-138-114-219.us-...
Connect
Get Windows Password
Create Template From Instance
Launch More Like This
Instance State
Instance Settings
Image
Networking
CloudWatch Monitoring
Create Image
Instance: i-071bcd59b22715516 (awais-2)
Public DNS: ec2-3-138-114-219.us-east-2.compute.amazonaws.com
Description
Status Checks
Monitoring
Tags
Instance ID
i-071bcd59b22715516
Public DNS (IPv4)
ec2-3-138-114-219.us-east-2.compute.amazonaws.com

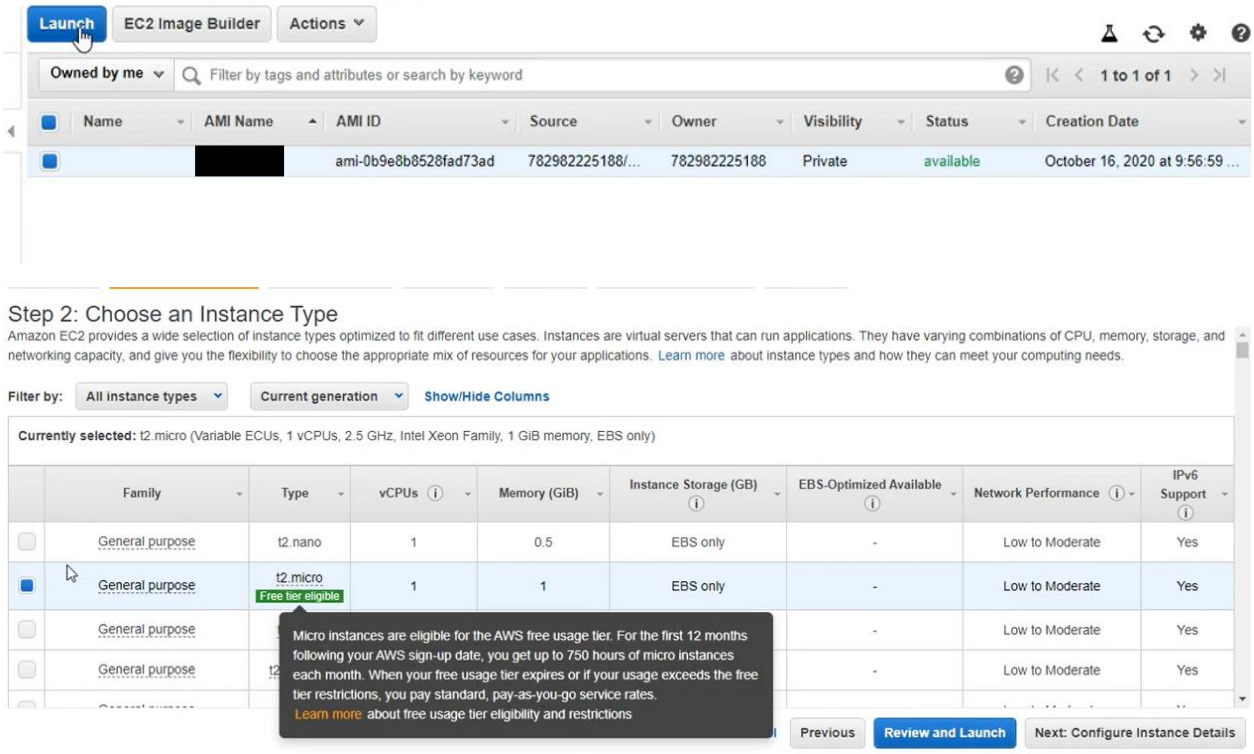

Create Image
Description of the image. This field is limited to 127 characters in length and cannot be modified.
Instance ID
i-071bcd59b22715516
Image name
Image description
No reboot
Instance Volumes
Volume Type
Device
Snapshot
Size (GiB)
Volume Type
IOPS
Throughput (MB/s)
Delete on Termination
Encrypted
Root
/dev/xvda
snap-05741358b44a33b45
8
General Purpose S
100 / 3000
N/A
Not Encrypted
Add New Volume
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.
Cancel
Create Image



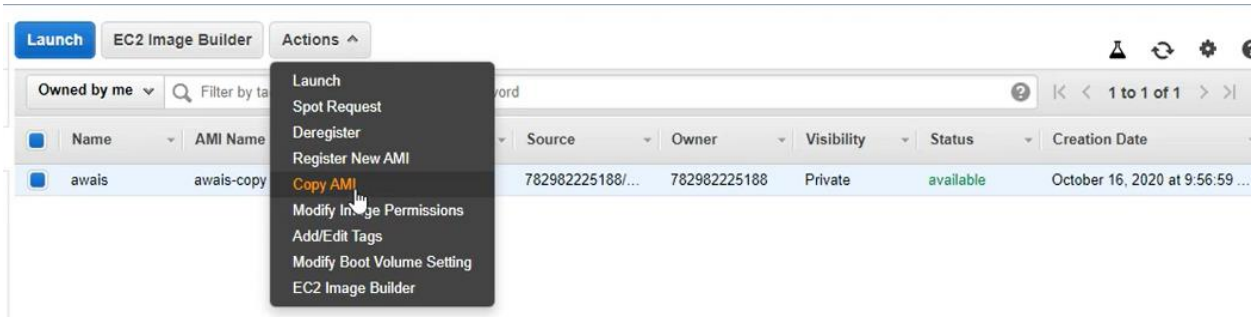
- Go to AMI on the left side and launch the server from there. Follow the steps and launch the server. Now the server is launched in the same region. In this case you will not be asked to select the AMI as the AMI is already copied. The new server will have Apache installed on it.
- You can make any changes if you want to. Same SG and Key pair are recommended to use.

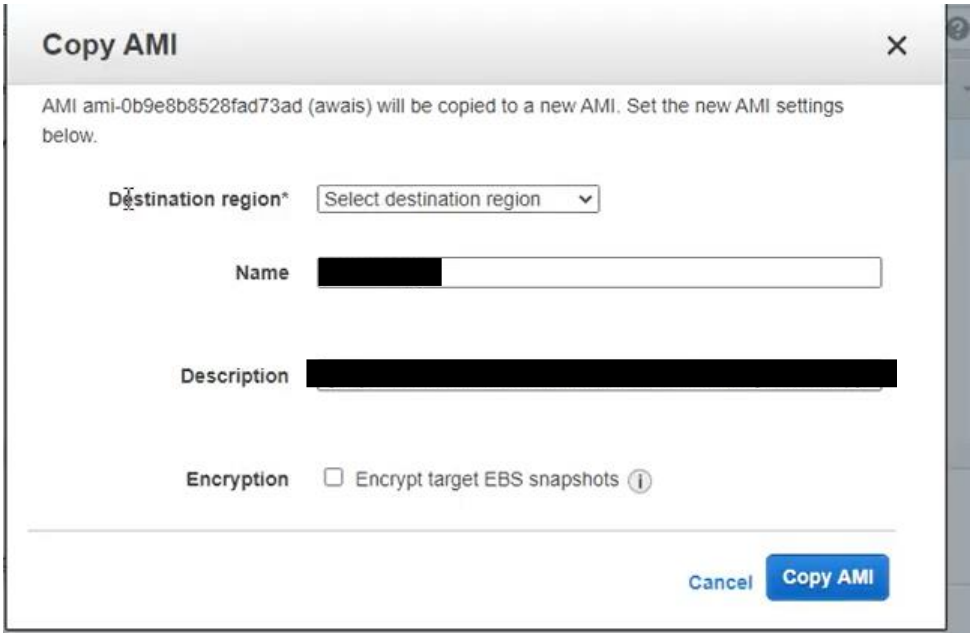




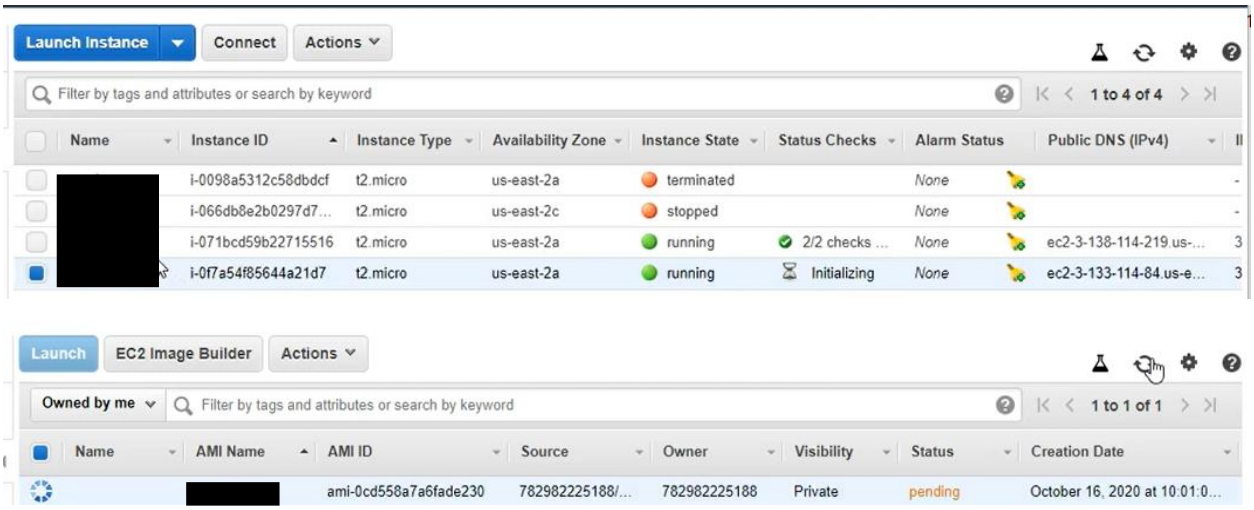


- To launch or copy the server to any other region select copy AMI and specify the region. Set the destination and click Copy AMI. Encryption option can be selected if required.

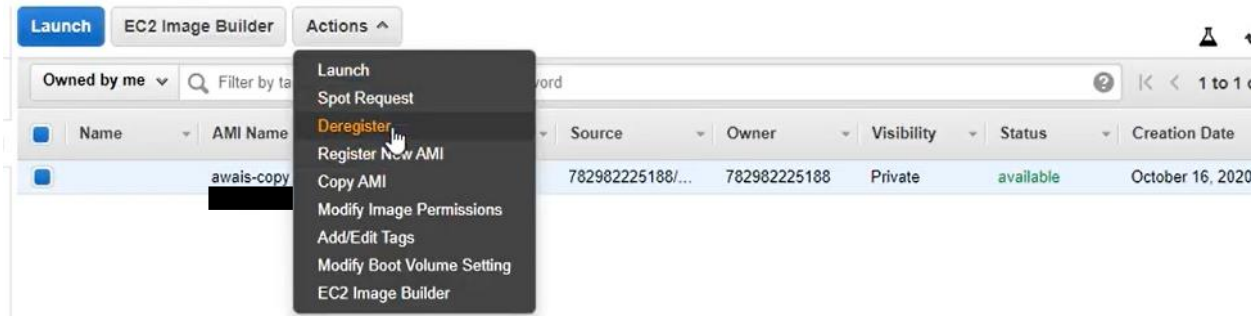




- Visit the region to see if the server is copied and launch the server there using similar steps. Here SG and Key pair is to be set as in different region SG and key pair cannot be used.



DONOT FORGET to delete the AMI and terminate the instance using deregister and terminate instances.



Lab (Using Elastic Load Balancer for scalability)

Scalability & High Availability

- Scalability means that an application / system can handle greater loads by adapting.
- There are two kinds of scalability:
  - Vertical Scalability
  - Horizontal Scalability (= elasticity)

- **Vertical Scaling** means to increase the size of the instance i.e., shifting from t2.micro to t2.large. It is very common for non-distributed systems like databases. RDS is an example of elastic cache that scales vertically. This scaling has a bound due to hardware limitation.
- **Horizontal scaling** means to increase the number of instances. It is very common for applications like web sites. EC2 is an example of horizontal scaling.

High Availability

- High Availability usually goes hand in hand with horizontal scaling
- High availability means running your application / system in at least 2 data centers (== Availability Zones)
- The goal of high availability is to survive a data center loss

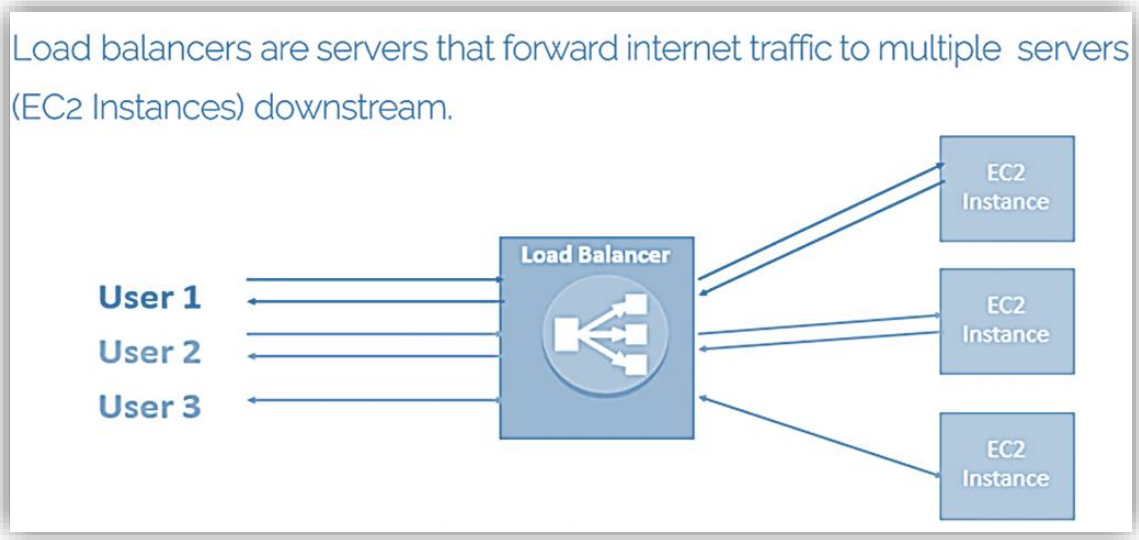
- Vertical Scaling: Increase instance size (= scale up / down)
  - From: t2.nano - 0.5G of RAM, 1 vCPU
  - To: u-12tb1.metal – 12.3 TB of RAM, 448vCPUs

- Horizontal Scaling: Increase number of instances (= scale out / in)
  - Auto Scaling Group
  - Load Balancer

High Availability: Run instances for the same application across multi AZ

- Auto Scaling Group multi AZ
- Load Balancer multi AZ

- **Load balancers** distribute the traffic amongst the instances and help in providing elasticity. Instances are attached with load balancer and the instances are then accessed through the load balancer via DNS providing single point of access to your application.

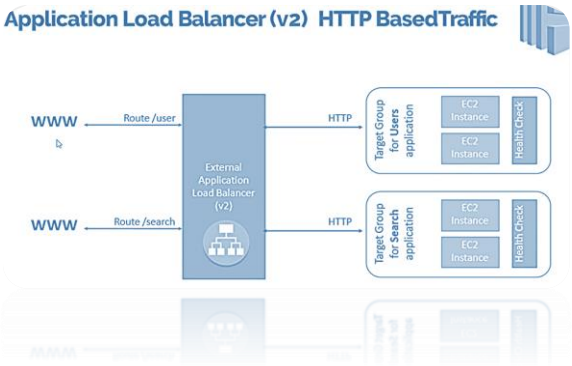


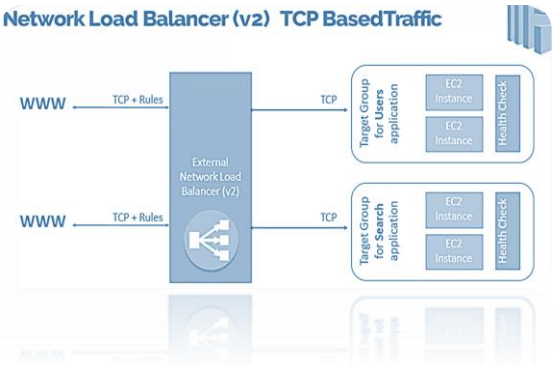
Why use load balancer?	Seamlessly handle failures of downstream instances
	Do regular helath checks of instances
	Provide SSL termination (HTTPS) for your website
	Enforce stickiness with cookies
	High availabilty across zones
	Separate public and private traffic

Why use an EC2 Load Balancer ?

- An ELB (EC2 Load Balancer) is a managed load balancer
  - AWS guarantees that it will be working
  - AWS takes care of upgrades, maintenance, high availability
  - AWS provides only a few configuration knobs
- It costs less to setup your own load balancer but it will be a lot more effort on your end.
- It is integrated with many AWS offerings / services

Application Load Balancer	Network Load Balancer and Classic load balancer
• Used to route HTTP / HTTPs (layer 7) traffic • Load balance across machine (target group) • Load balance on same machine (containers) • Best for micro services and container based applications e.g., Dockers and Amazon ECS • Has port mapping feature to redirect to dynamic port	• Used to route TCP (layer 5) traffic • Handle millions of requests per second • Support static and elastic IP • Used for extreme performance • Should not be the default load balancer





- Health Checks are crucial for Load Balancers
- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests
- The health check is done on a port and a route (/health is common)
- If the response is not 200 (OK), then the instance is unhealthy

Load Balancer Security Groups

Load Balancer Security Group:

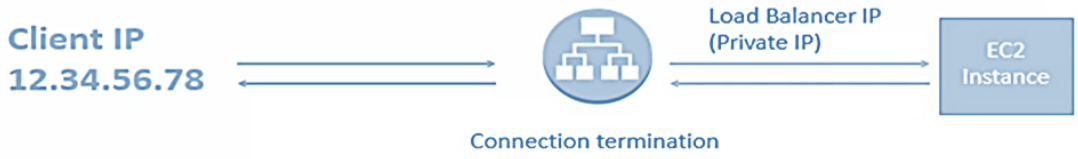
Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow HTTP from an...
HTTPS	TCP	443	0.0.0.0/0	Allow HTTPS from a...

Application Security Group: Allow traffic only from Load Balancer

Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	sg-054b5ff5ea02f2b6e (load-b	Allow Traffic only...

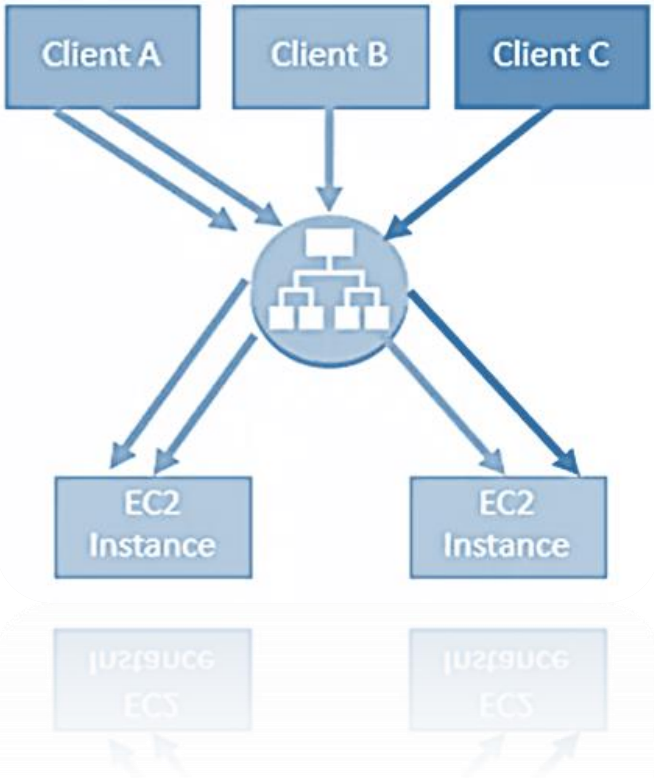
Application Load Balancer

- Stickiness can be enabled at the target group level
  - Same request goes to the same instance
  - Stickiness is directly generated by the ALB (not the application)
- ALB support HTTP/HTTPS & Web sockets protocols



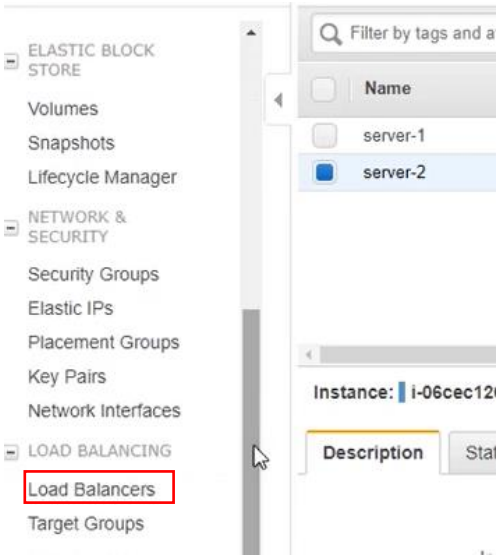
Load Balancer Stickiness

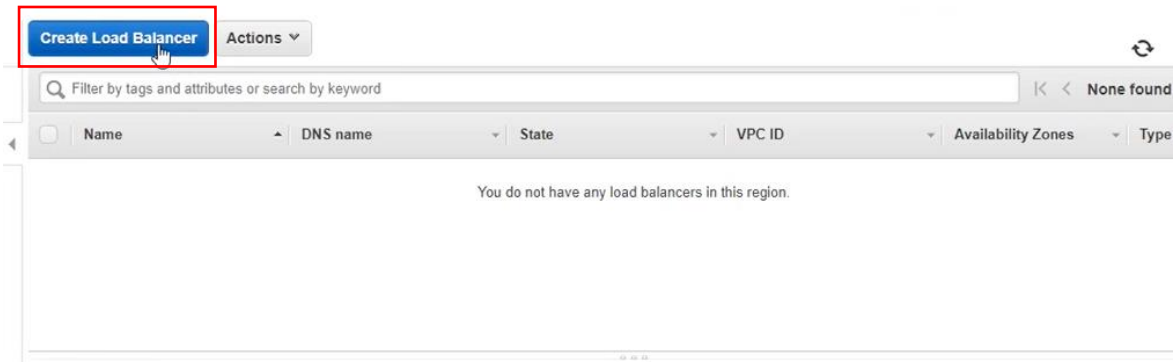
- Load balancer can be made sticky
- stickiness means that same client is always redirected to same instance
- Cookie is used for stickiness and has expiration date
- Enabling stickiness may bring imbalance to load balancer over backend EC2 instances

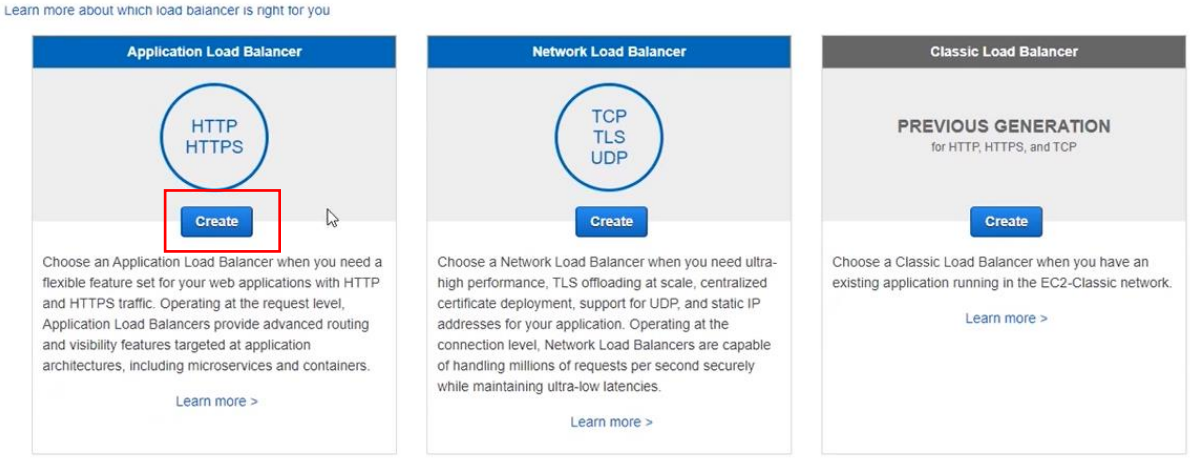


- Launch the Linux Server with boot strap code given below. You must launch two servers, just to make the difference change the echo statement of bash file for both servers. Your SG must have HTTP and SSH port open. Use public IPs of server to open in browser.

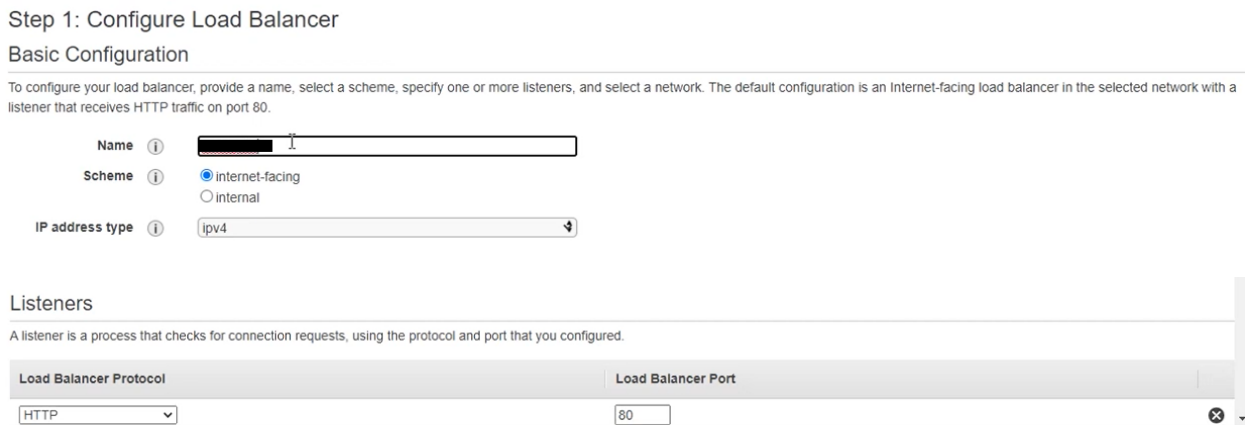
- On left side scroll to Load balancer and click create load balancer and select Application load balancer.

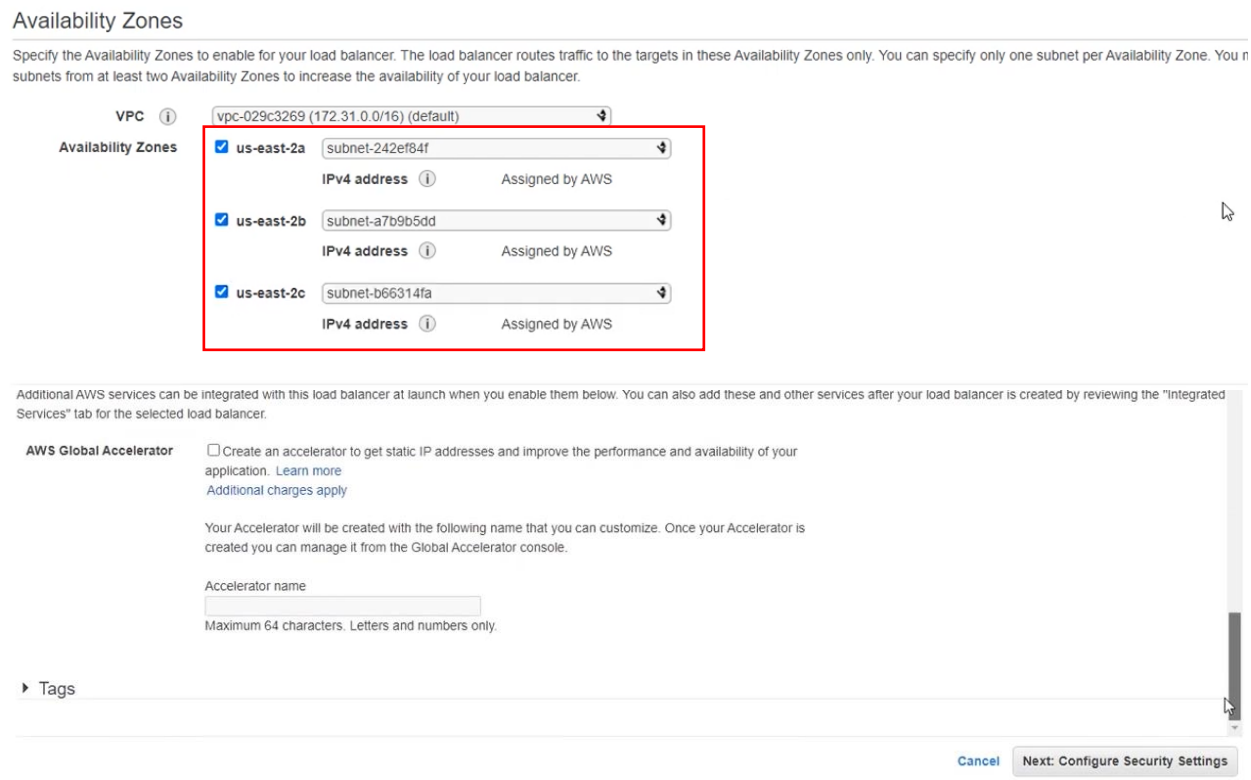




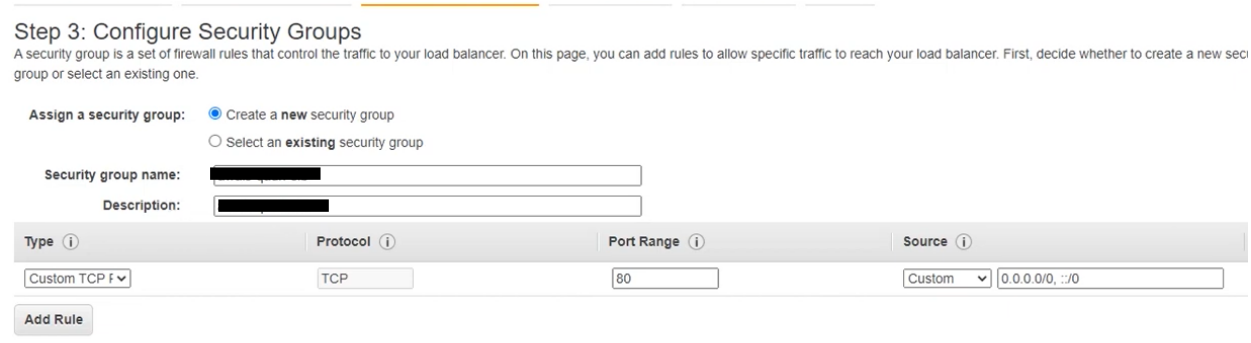


- Configure the load balancer set the name and select internet facing to manage the traffic from public internet. To make high availability select all the regions. The listener must be port 80 for HTTP traffic.

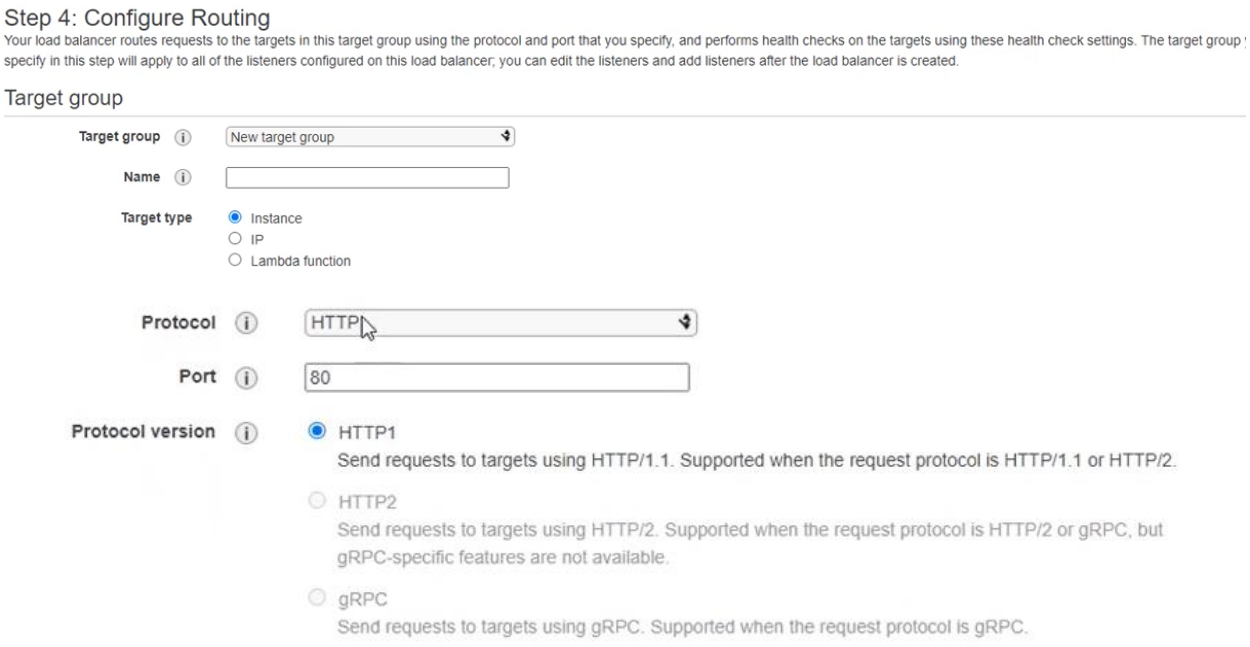




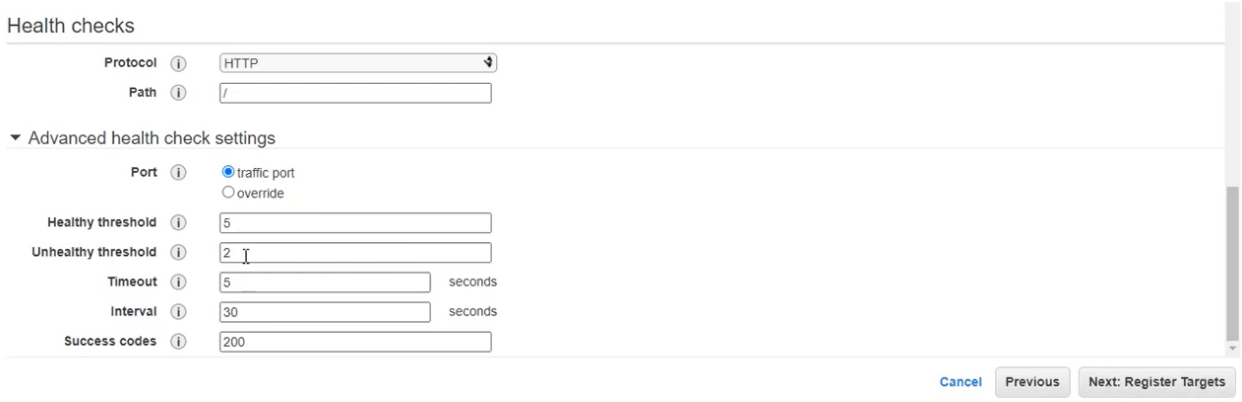
- Click Next and configure the security group.

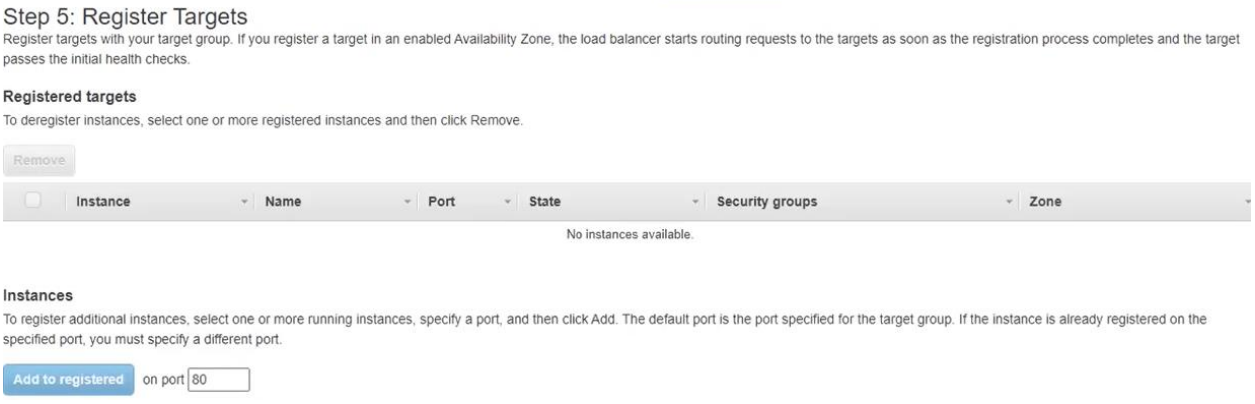


- Now configure target groups.

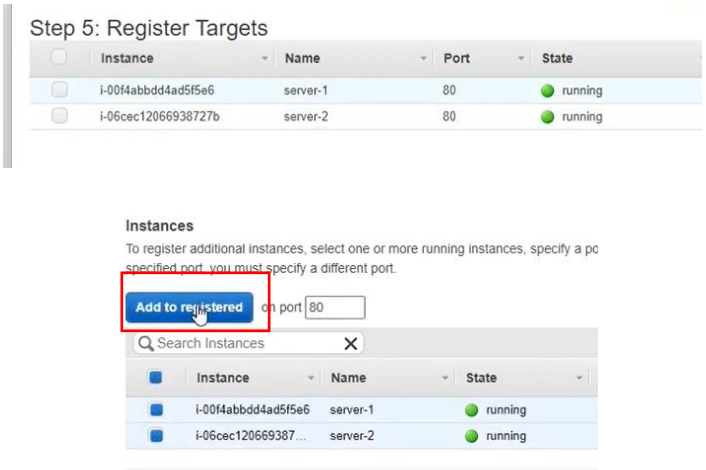


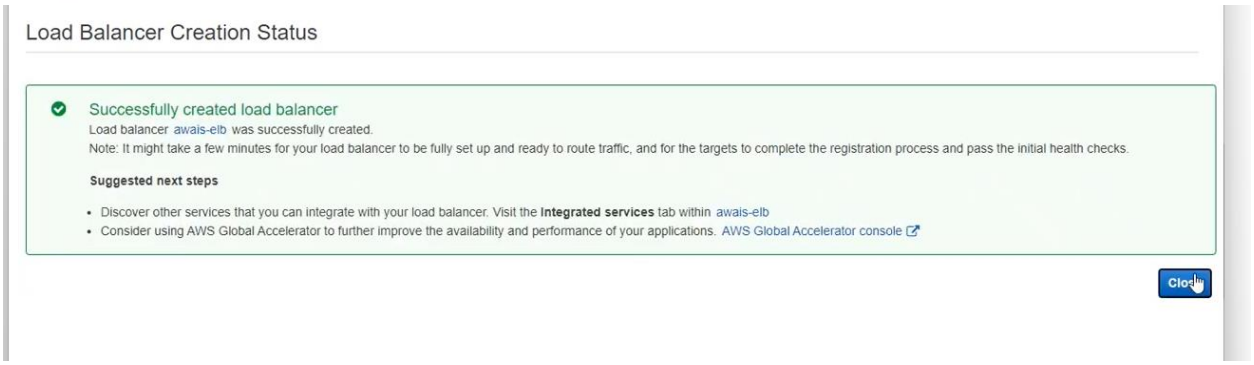
- You can set the path with any error.html file prompting any error say server down etc. The value for unhealthy threshold is 2 and for healthy its kept 5 i.e., no response is given by the server after 2 requests.

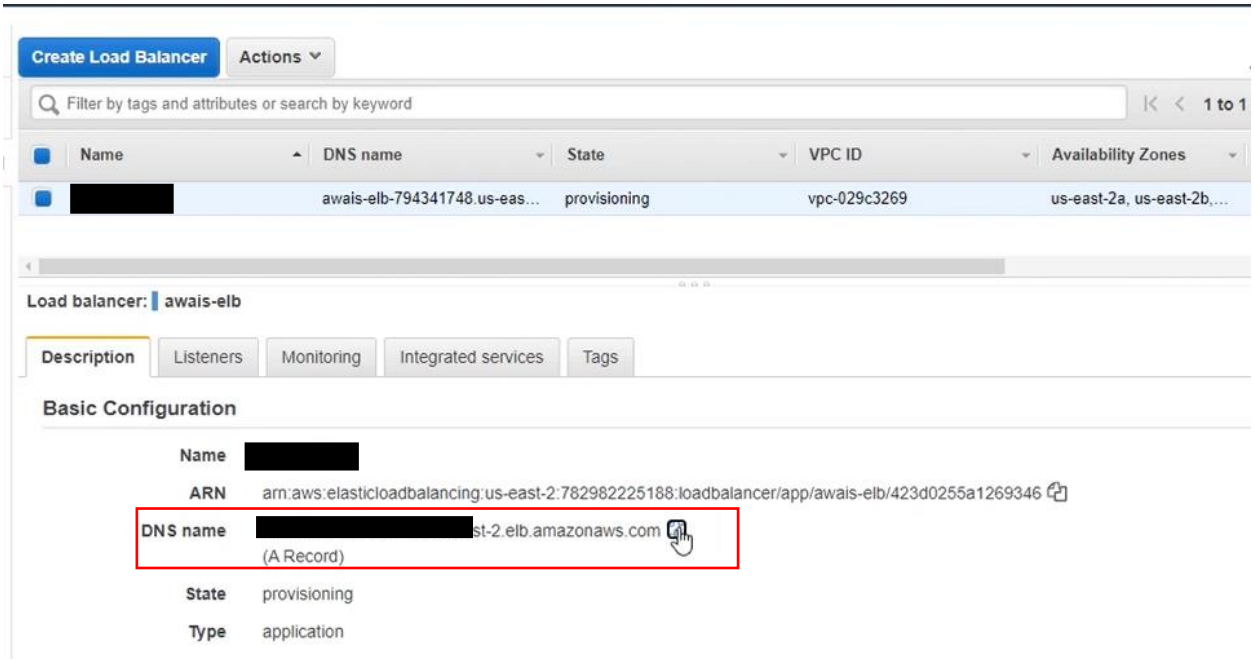




- After registering the target attach the instances (running) and click review and create.







- Now use the DNS address of Load balancer and browse it. Refresh the page to see different servers responding to the request.

  How to avoid usage of public IP of servers if DNS of load balancer is used? *Hint in Security group timeout can be used.*