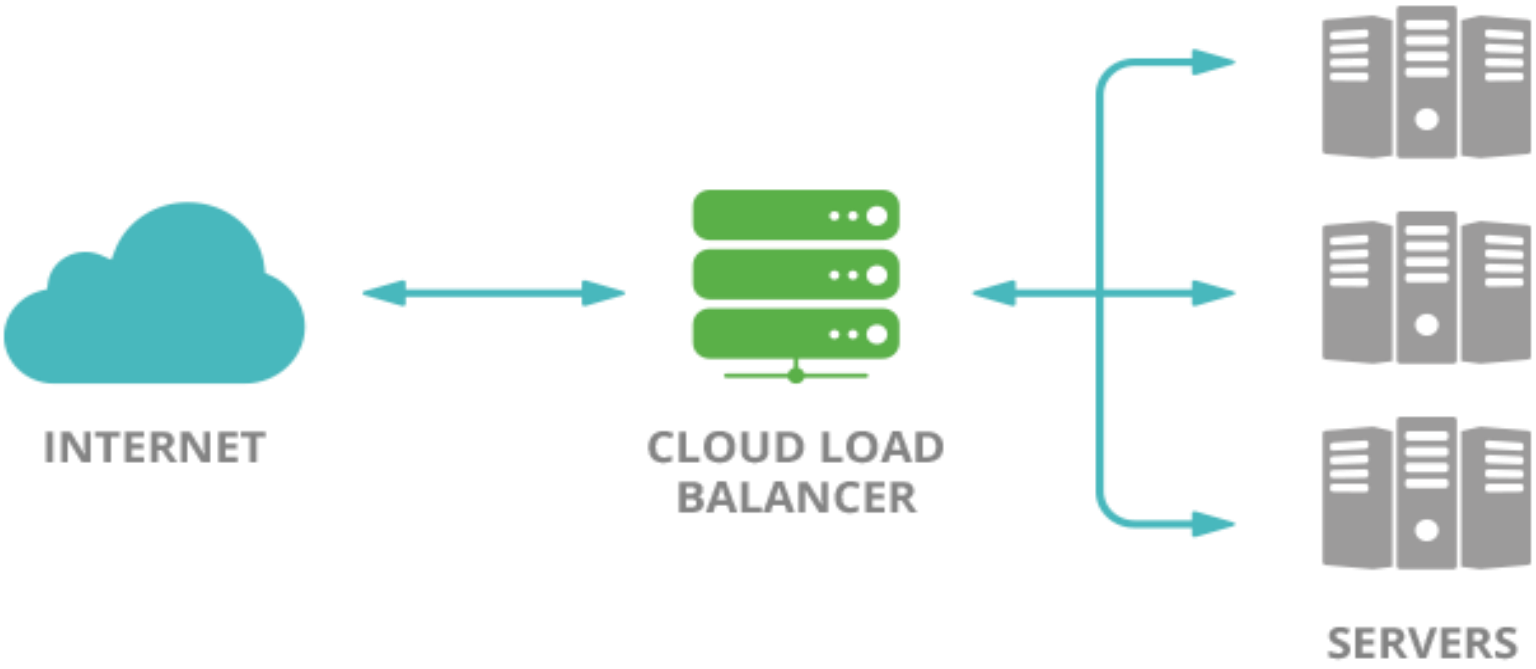
Elastic
Load Balancer

ELB Advantages

1) Load balancer

2) Failover

3) Any time any number of instances can be added or removed



Topics to be covered--ELB

1) ELB Introduction

2) CLB -- with linux and windows

3) ALB -- with linux and windows –same target group

4) ALB -- with linux and windows – different target group

5) AWS Global Accelerator

6) Sticky session

7) Configure log files in S3 bucket

Elastic Load Balancer

- ✓ A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the **availability** and **fault tolerance** of your applications.

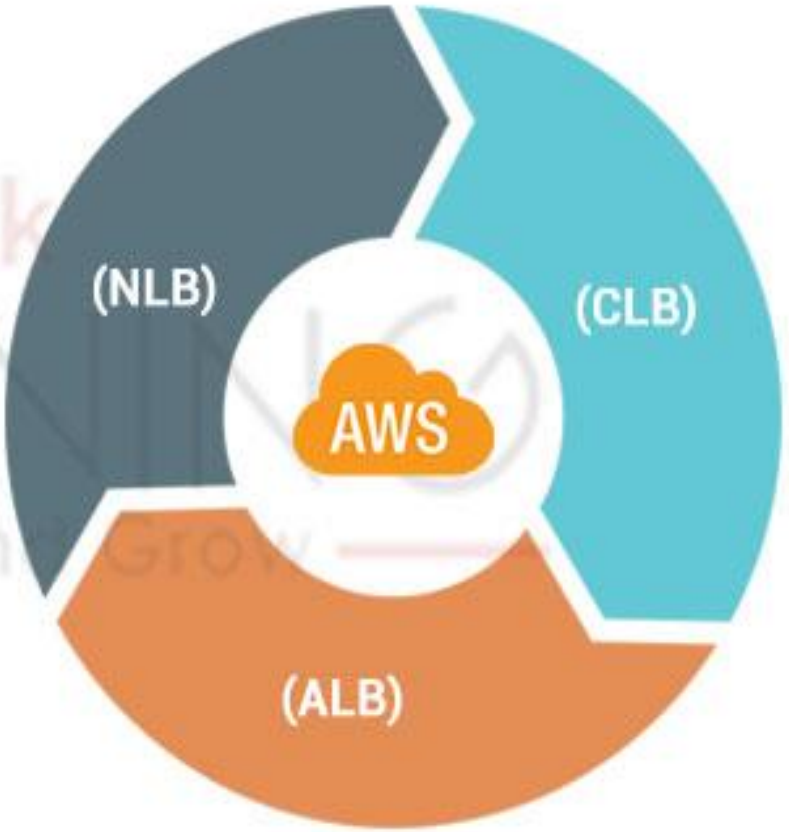
- ✓ You can add and remove compute resources from your load balancer as your needs change, without disrupting the overall flow of requests to your applications.

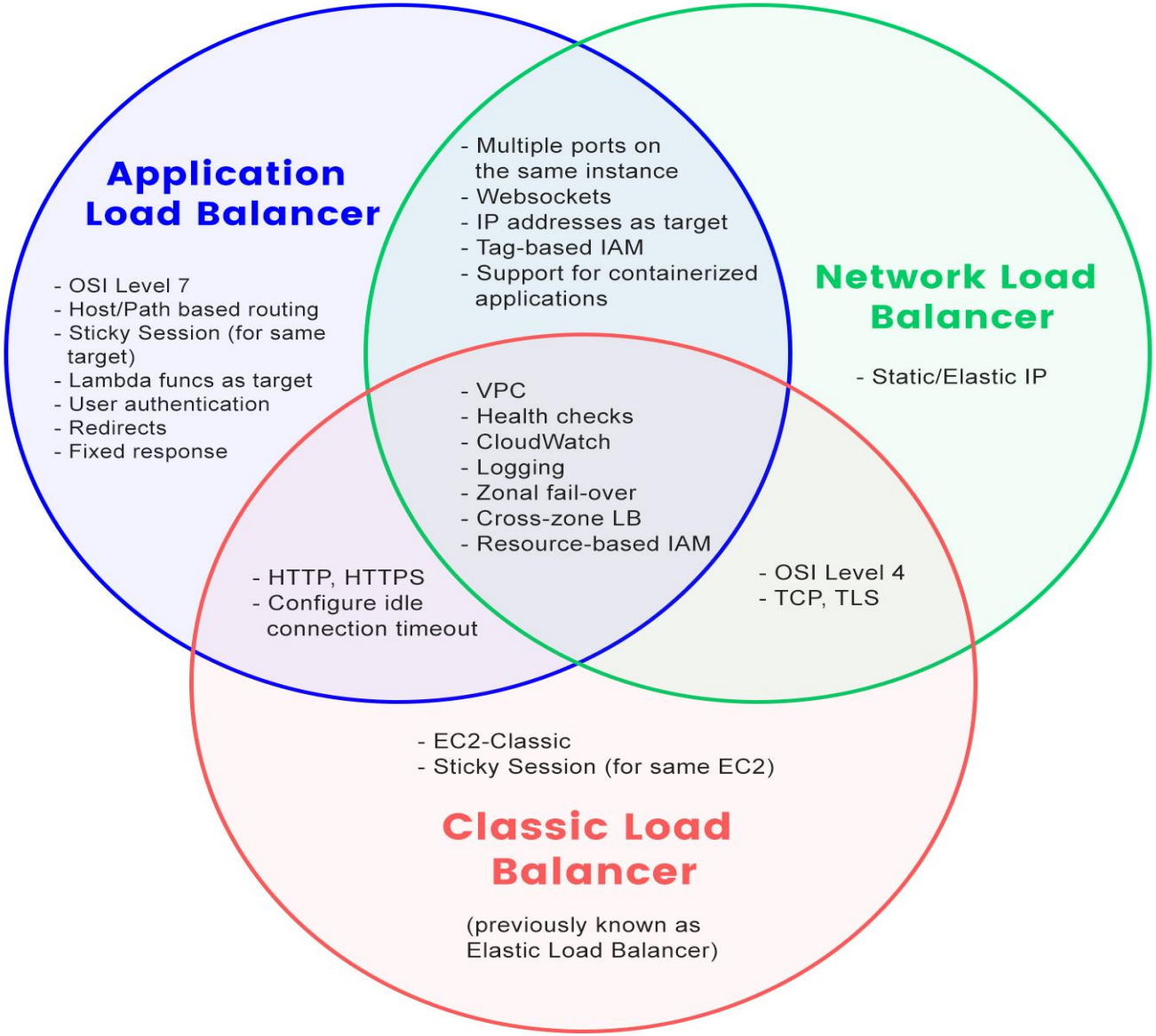
- ✓ You can configure health checks, which are used to monitor the health of the compute resources so that the load balancer can send requests only to the healthy ones. You can also offload the work of encryption and decryption to your load balancer so that your compute resources can focus on their main work.

Elastic Load Balancer Types

Amazon Load Balancer

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




Application Load Balancer
- OSI Level 7
- Host/Path based routing
- Sticky Session (for same target)
- Lambda funcs as target
- User authentication
- Redirects
- Fixed response
- Multiple ports on the same instance
- Websockets
- IP addresses as target
- Tag-based IAM
- Support for containerized applications
Network Load Balancer
- Static/Elastic IP
- VPC
- Health checks
- CloudWatch
- Logging
- Zonal fail-over
- Cross-zone LB
- Resource-based IAM
- HTTP, HTTPS
- Configure idle connection timeout
- OSI Level 4
- TCP, TLS
- EC2-Classic
- Sticky Session (for same EC2)
Classic Load Balancer
(previously known as Elastic Load Balancer)

Elastic Load Balancer

1. Classic Load Balancers.

Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level. Classic Load Balancer is intended for applications that were built within the EC2-Classic network.

2. Network Load Balancers

Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, containers and IP addresses based on IP protocol data. Ideal for load balancing of TCP traffic, Network Load Balancer is capable of handling millions of requests per second while maintaining ultra-low latencies. Network Load Balancer is optimized to handle sudden and volatile traffic patterns while using a single static IP address per Availability Zone. It is integrated with other popular AWS services such as Auto Scaling, Amazon EC2 Container Service (ECS), and Amazon CloudFormation

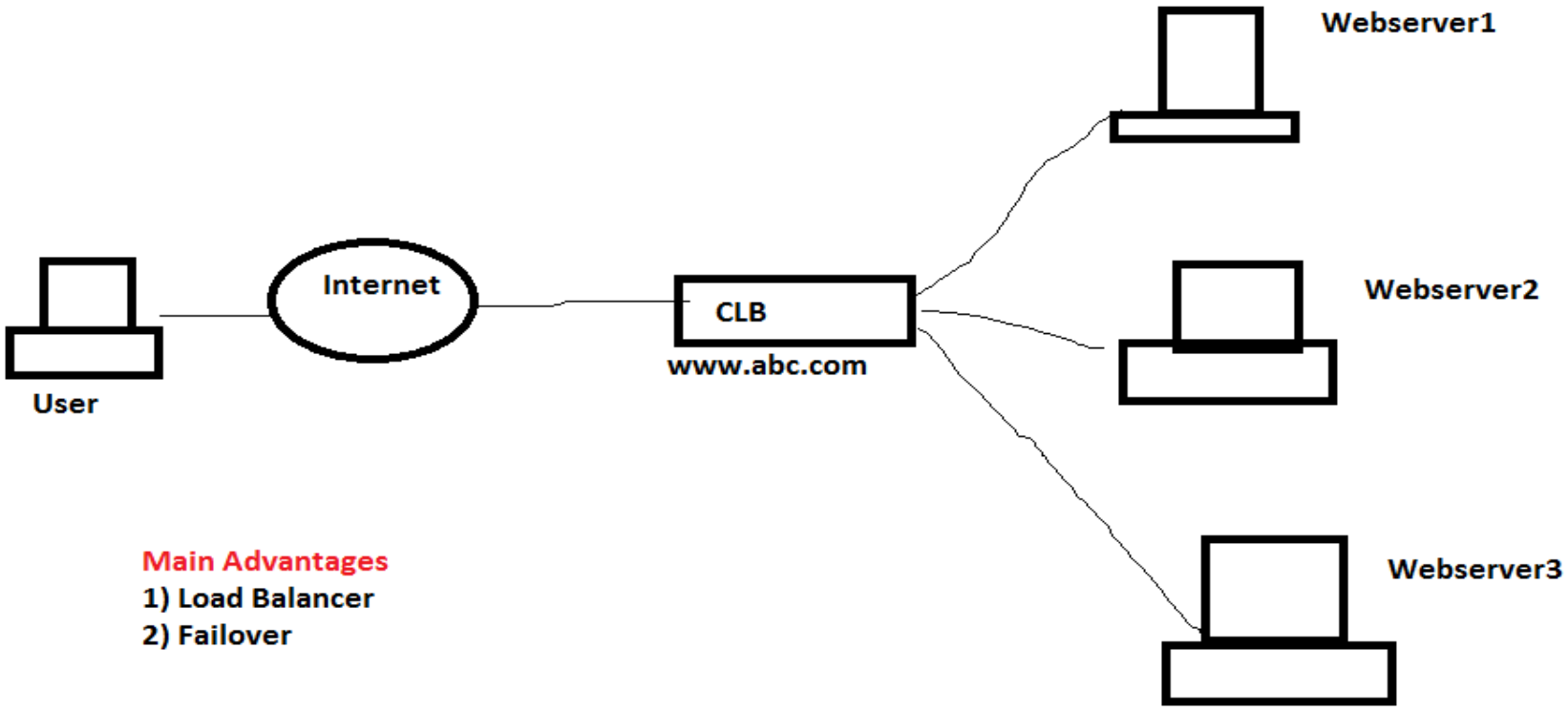
Elastic Load Balancer

1.Application Load Balancers

Application Load Balancer operates at the request level (layer 7), routing traffic to targets - EC2 instances, containers and IP addresses based on the content of the request. Ideal for advanced load balancing of HTTP and HTTPS traffic, Application Load Balancer provides advanced request routing targeted at delivery of modern application architectures, including microservices and container-based applications. Application Load Balancer simplifies and improves the security of your application, by ensuring that the latest SSL/TLS ciphers and protocols are used at all times.

Classic Load Balancers

ELB: Classic Load Balancer



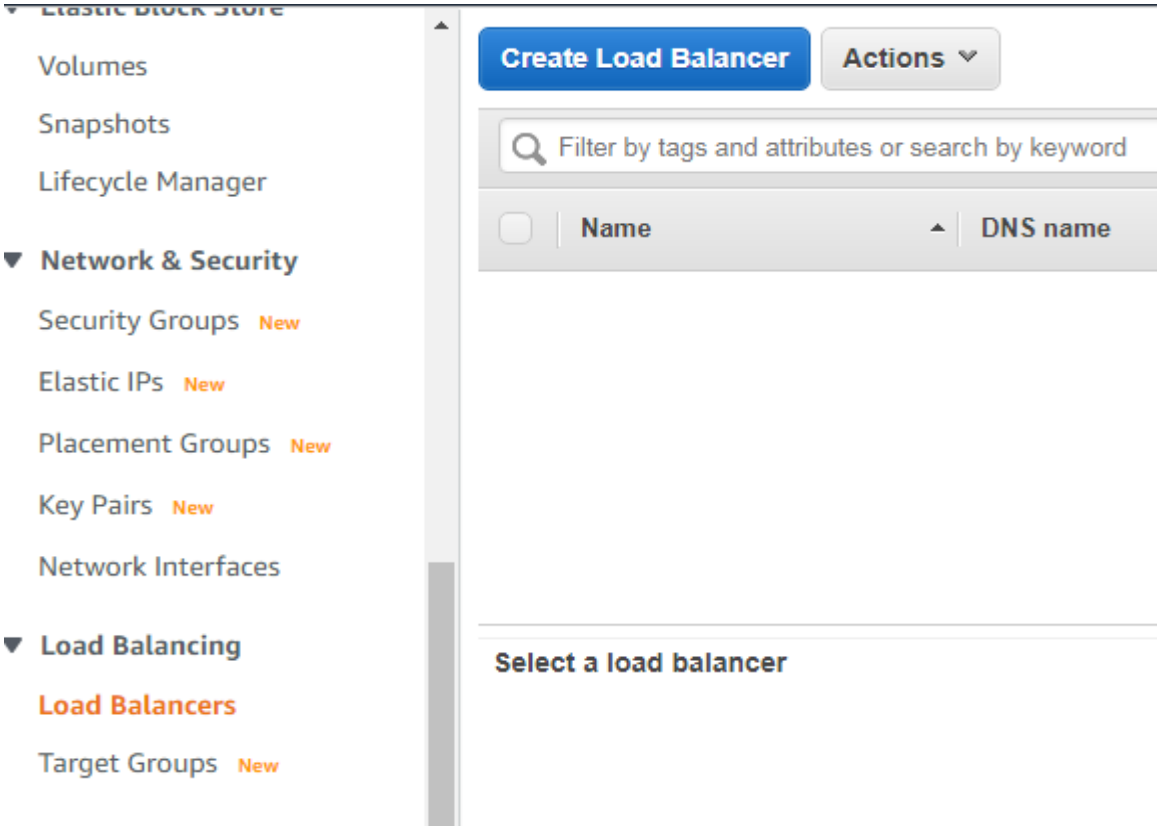
Main Advantages

1) Load Balancer
2) Failover

Classic Load Balancers Steps

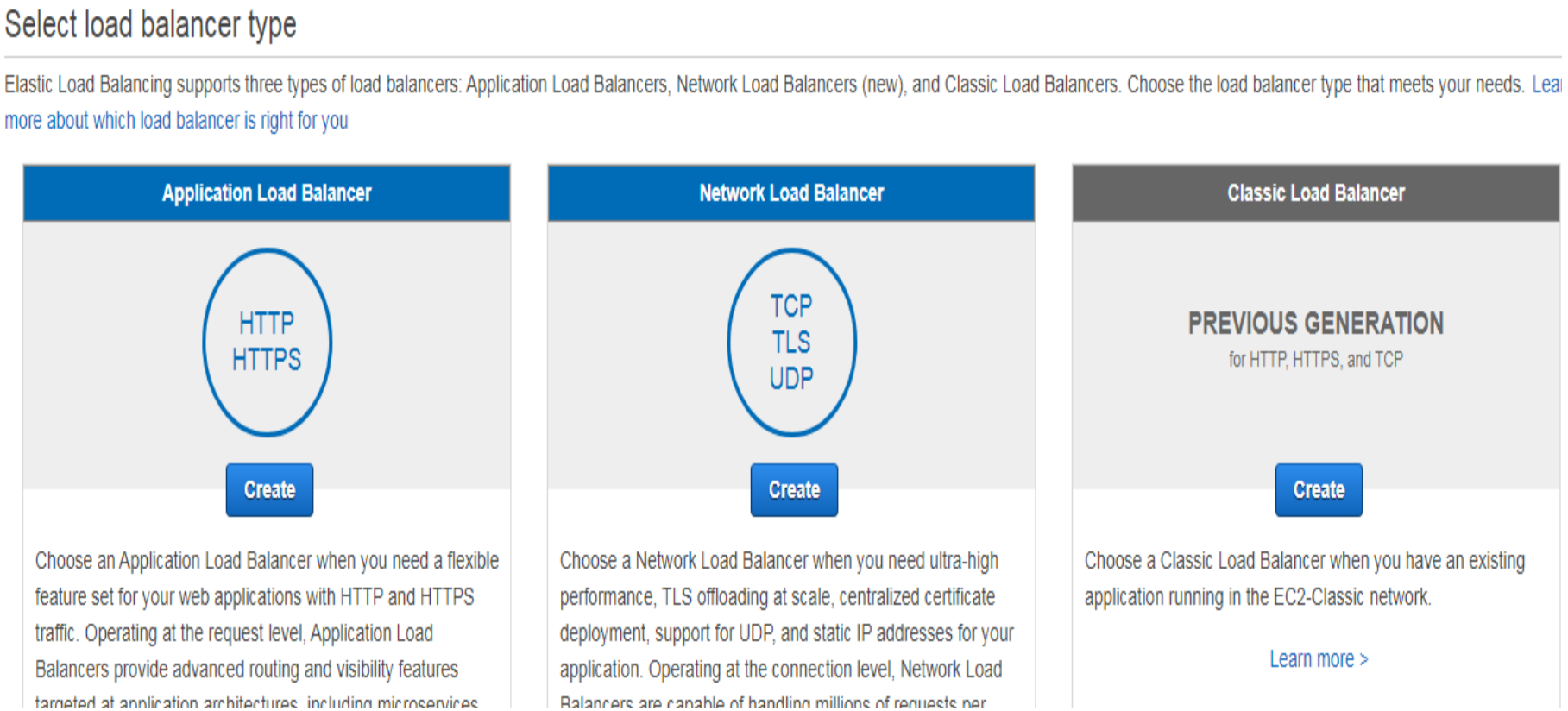
1) Create 3 Instances and Configure different web page

2) Click on load balancer –Create load balancer --



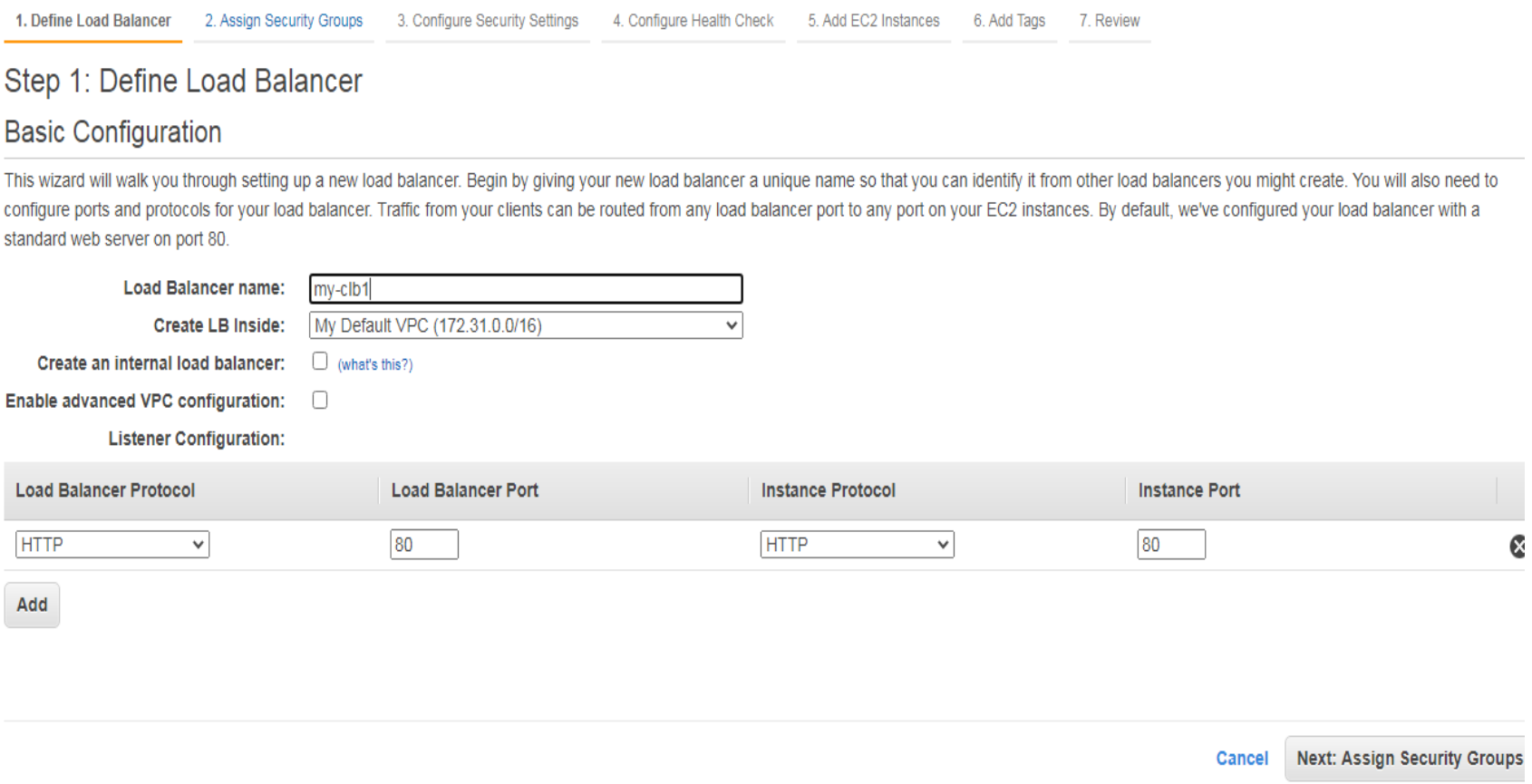
Classic Load Balancers Steps

Select Classic load balancer



Classic Load Balancers Steps

Give load balancer name



Classic Load Balancers Steps

Select Security group(SSH or RDP and HTTP allowed)

Step 2: Assign Security Groups

You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load balance
This can be changed at any time.

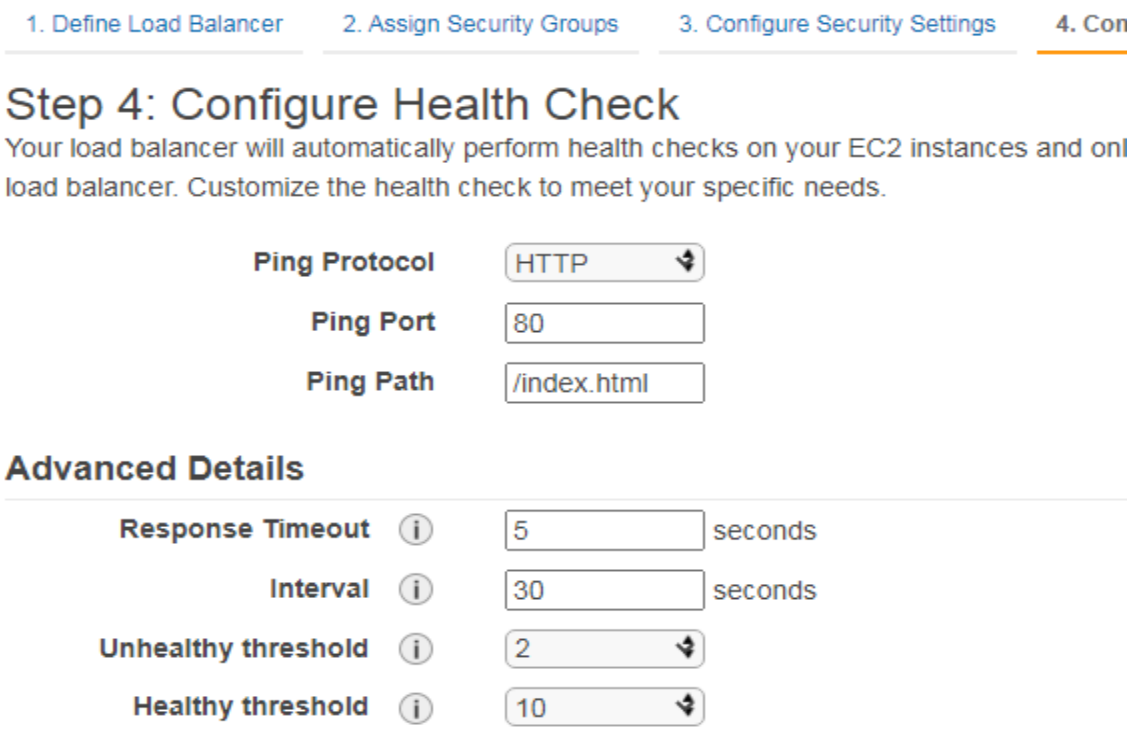
Assign a security group: ○ Create a **new** security group

◉ Select an **existing** security group

	Security Group ID	Name	Description
☐	sg-08c1caa50b77360f1	all traffic	launch-wizard-2 created 2020-08-13T18:34:27.224+05:30
☐	sg-9e8b8bfc	default	default VPC security group
☐	sg-0b82a58710c602670	EFS-SG	abc
☑	sg-06b9f816d6a387a04	Linux-SG	for project1
☐	sg-0c597cc5f9b72976b	linux-Sg-aws	launch-wizard-2 created 2020-08-18T21:33:14.152+05:30
☐	sg-0bd13efa05570edf1	windows -sg	launch-wizard-1 created 2020-08-06T08:48:12.876+05:30

Classic Load Balancers Steps

Keep default value



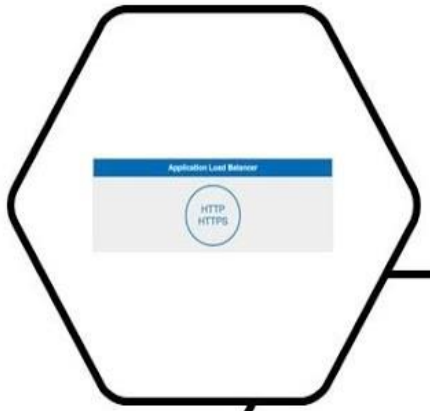
Classic Load Balancers Steps

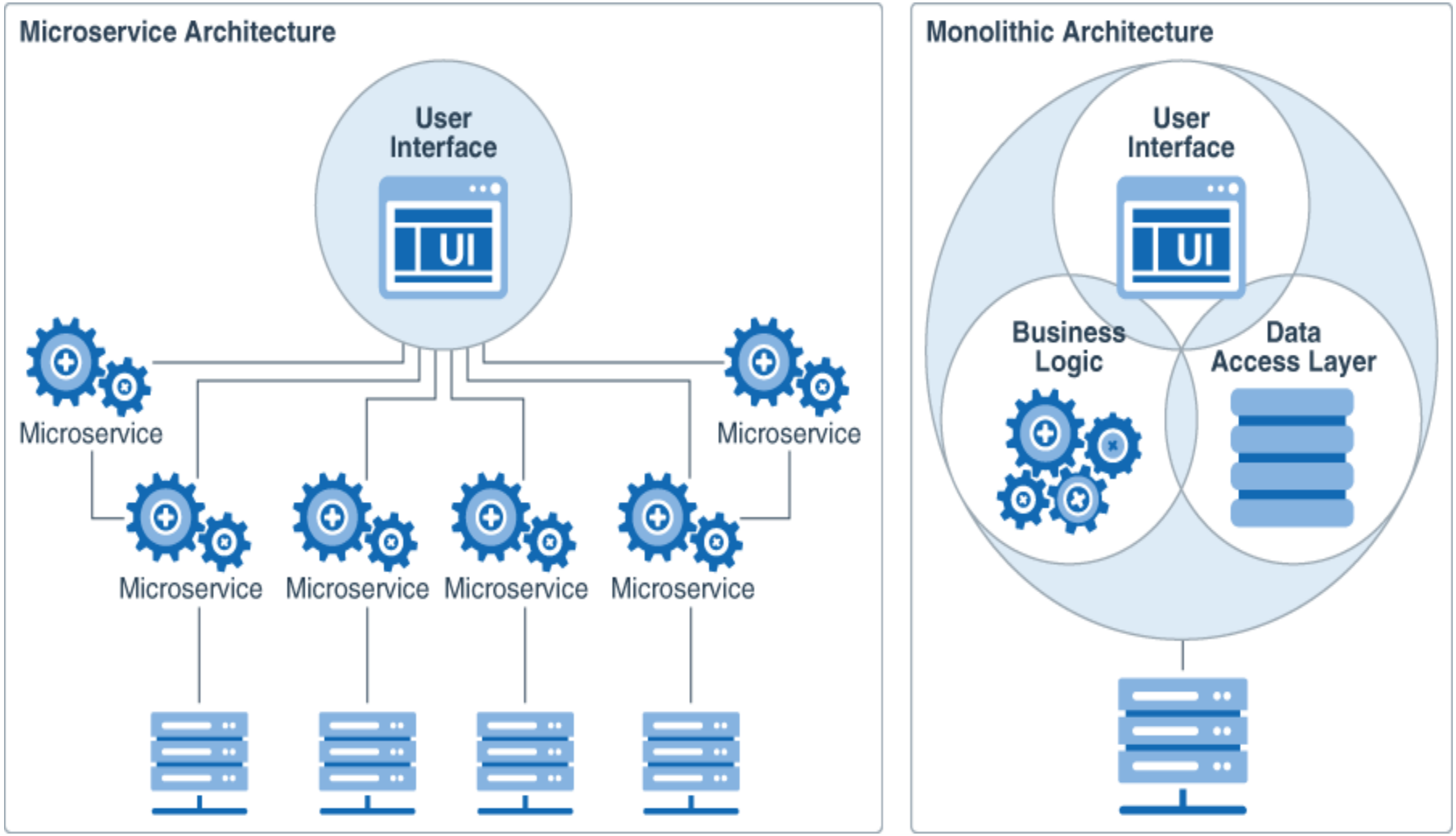
Next –add required number of instances –next-next--create

3) After Creating---scroll down –click on target—wait and refresh to check target status –changed from outservice to inservice

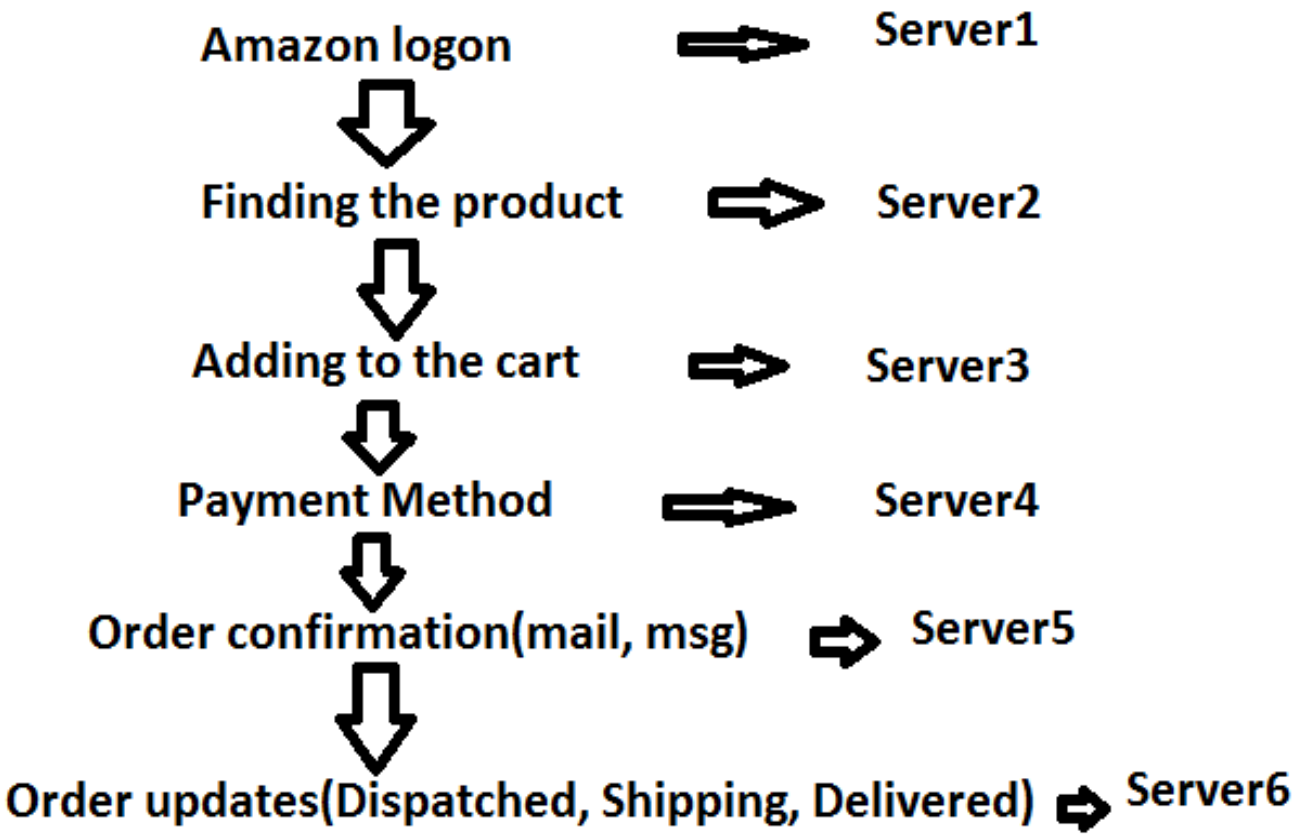
4)Description –copy dns name and paste in browser tab—keep refreshing

AWS Application Load Balancer



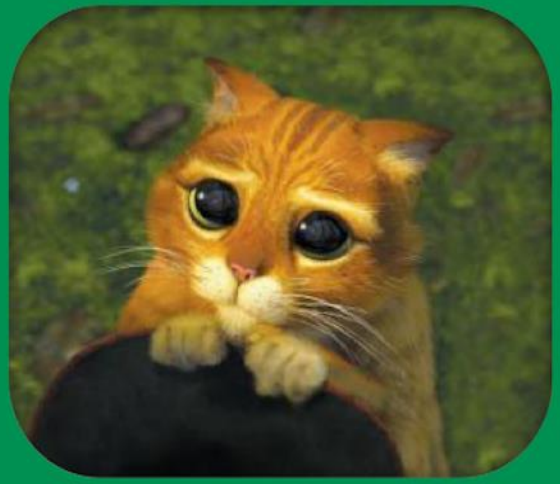

Microservice Architecture
User Interface
UI
Microservice
Microservice
Microservice
Microservice
Microservice
Microservice
Monolithic Architecture
User Interface
UI
Business Logic
Data Access Layer

Amazon.in Shopping Work Flow



DevOps Concepts: Pets vs. Cattle

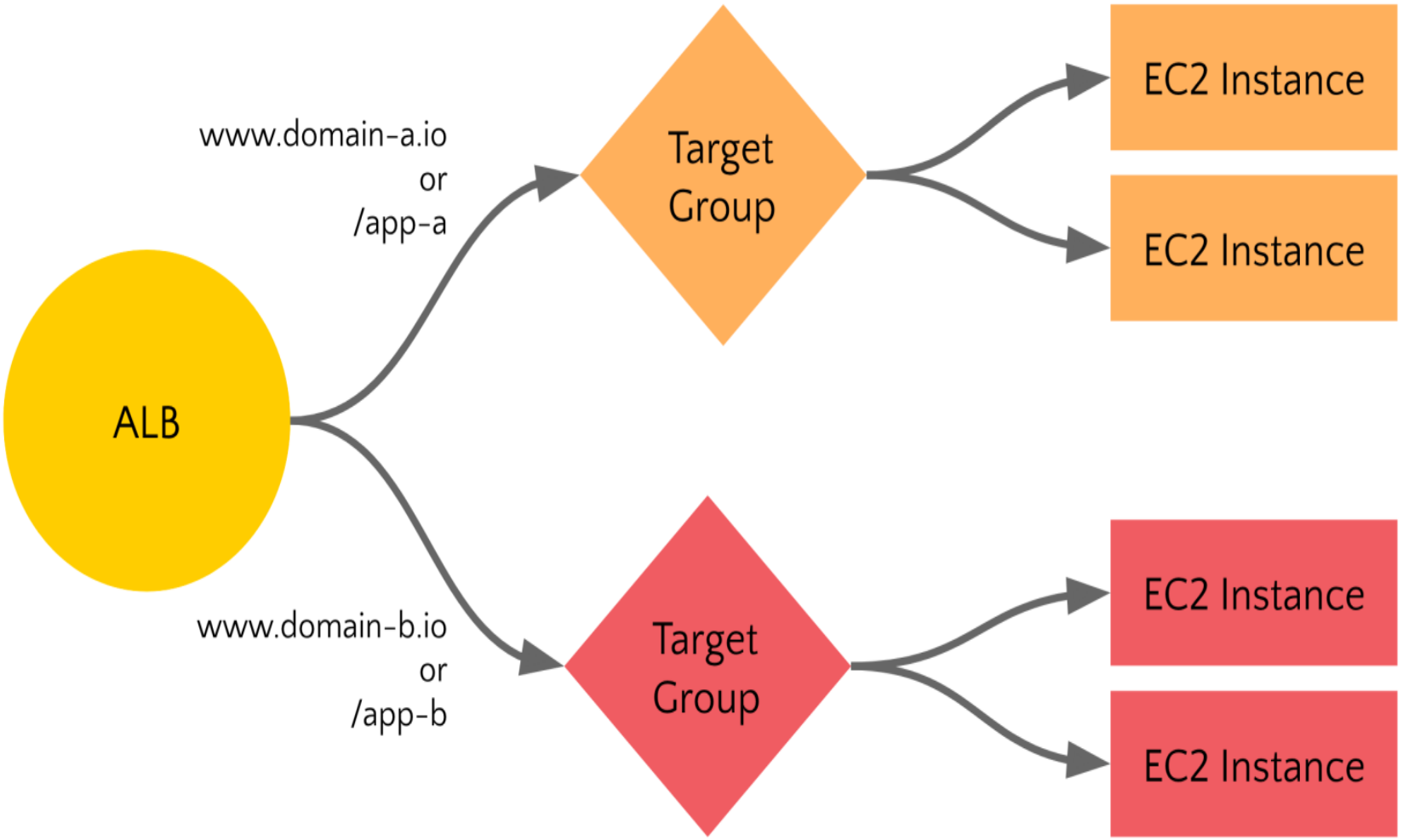
@Joachim8675309

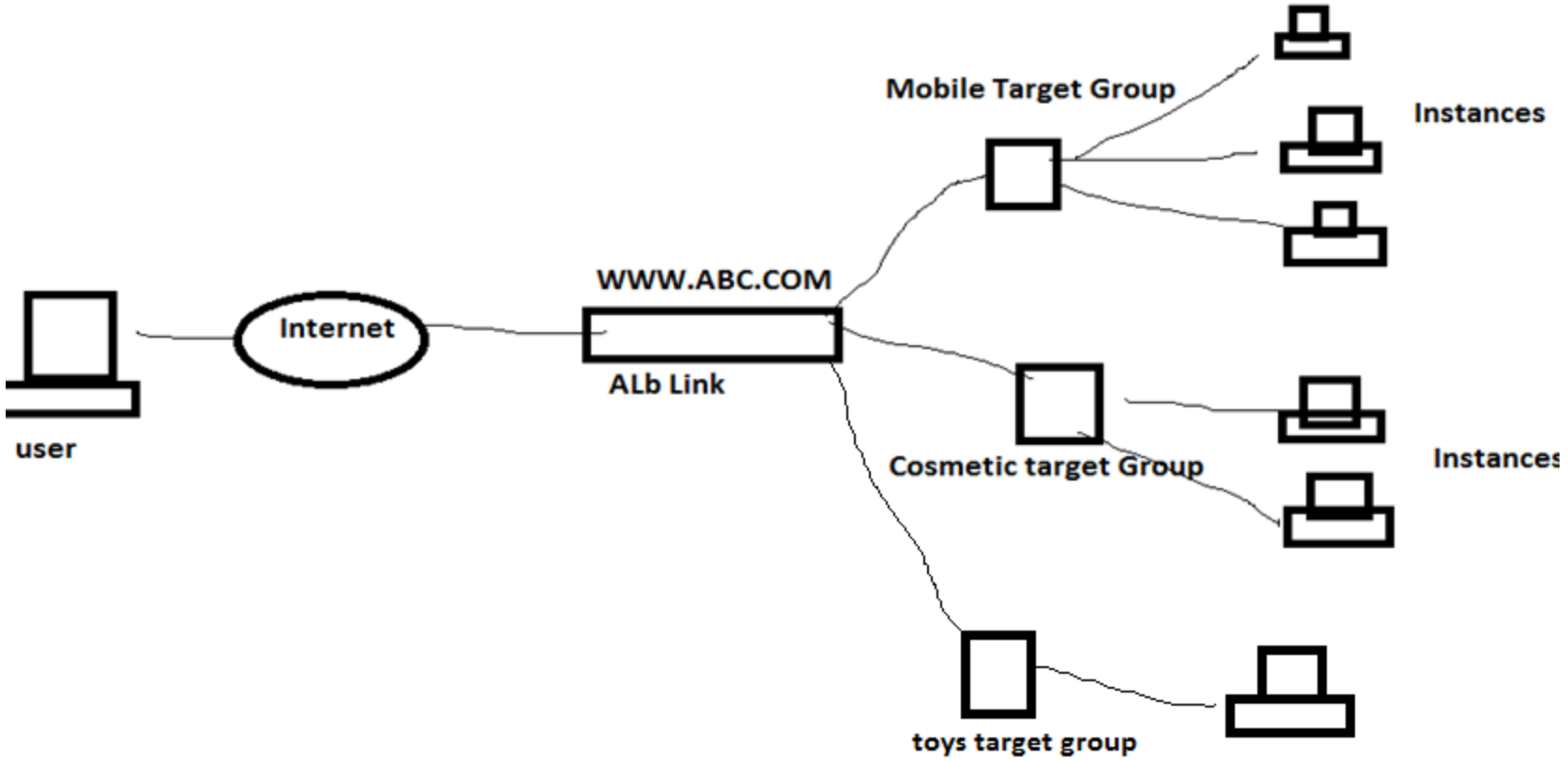


pets

vs

cattle


www.domain-a.io
or
/app-a
ALB
Target
Group
EC2 Instance
EC2 Instance
www.domain-b.io
or
/app-b
Target
Group
EC2 Instance
EC2 Instance


Mobile Target Group
Instances
WWW.ABC.COM
Internet
ALb Link
user
Cosmetic target Group
Instances
toys target group

Steps to configure ALB

1) Launch 3 instance and configure different web pages

2) Create Target groups for each application and add the instances

3) Create ALB

4) Configure condition

5) Copy ALB DNS and check

Steps to configure ALB

1) Launch 3 instance and configure different web pages

Mobile server --- mobile.html ---nokia, Samsung, iphone, htc

Cloth server ---cloth.html ---Jeans, T shirt, tie

Cosmetic server ---- cosmetic.html ---- powder, deo, oil

2) Create Target groups for each application and add the instances

EC2-Load balancer –Target group –Create Target group – mobile-target—create –close

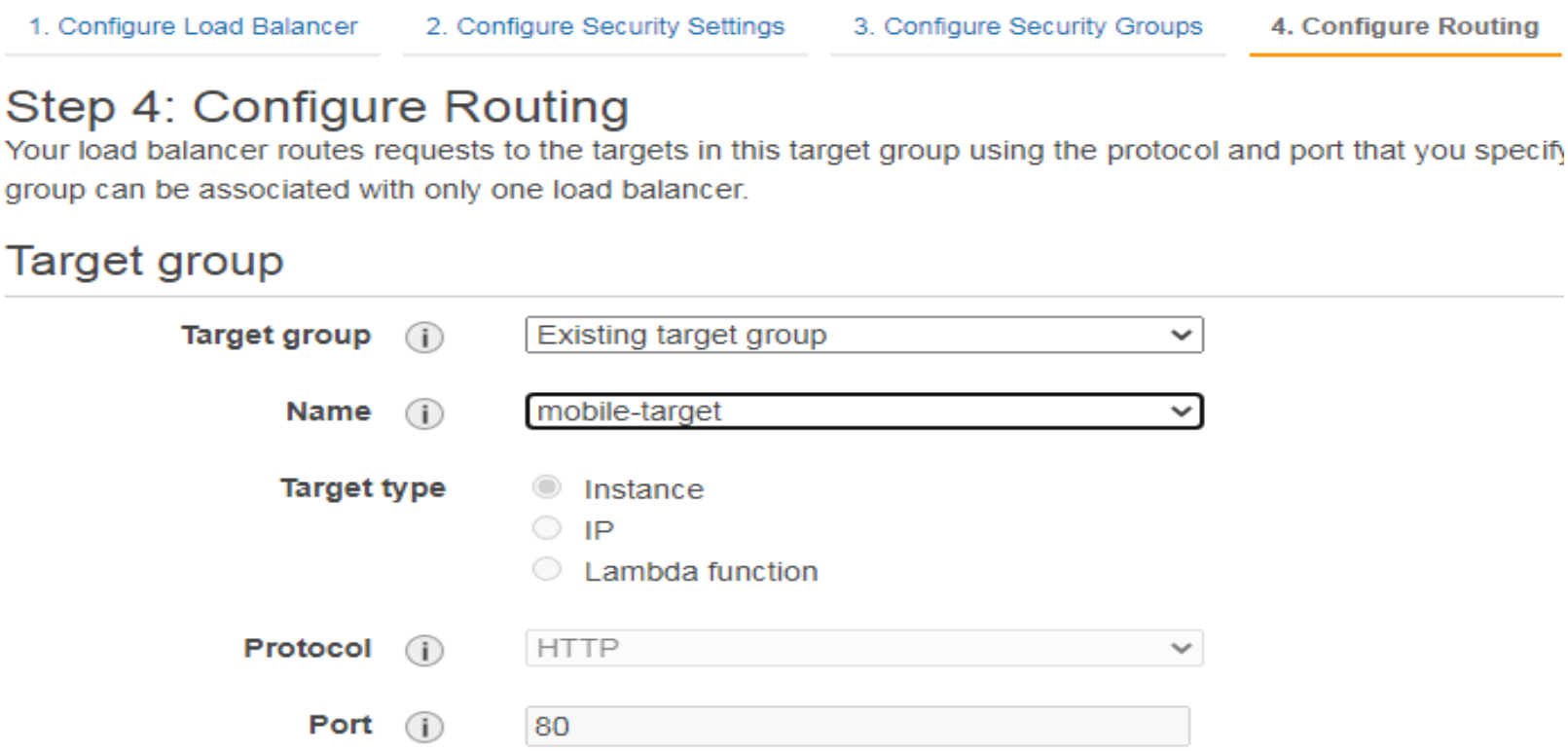
----Scroll down –Target –edit—select mobile server---add to registered –save

Do same for cloth and cosmetic server also

Steps to configure ALB

3) Create ALB

Load balancer –create load balancer—ALB – give name: my-alb ---scroll down and select all subnet –next---next –select same SG(Instance SG) ---- Select Target group: Existing , Name: Mobile target --Next—Next—Create Close



Steps to configure ALB

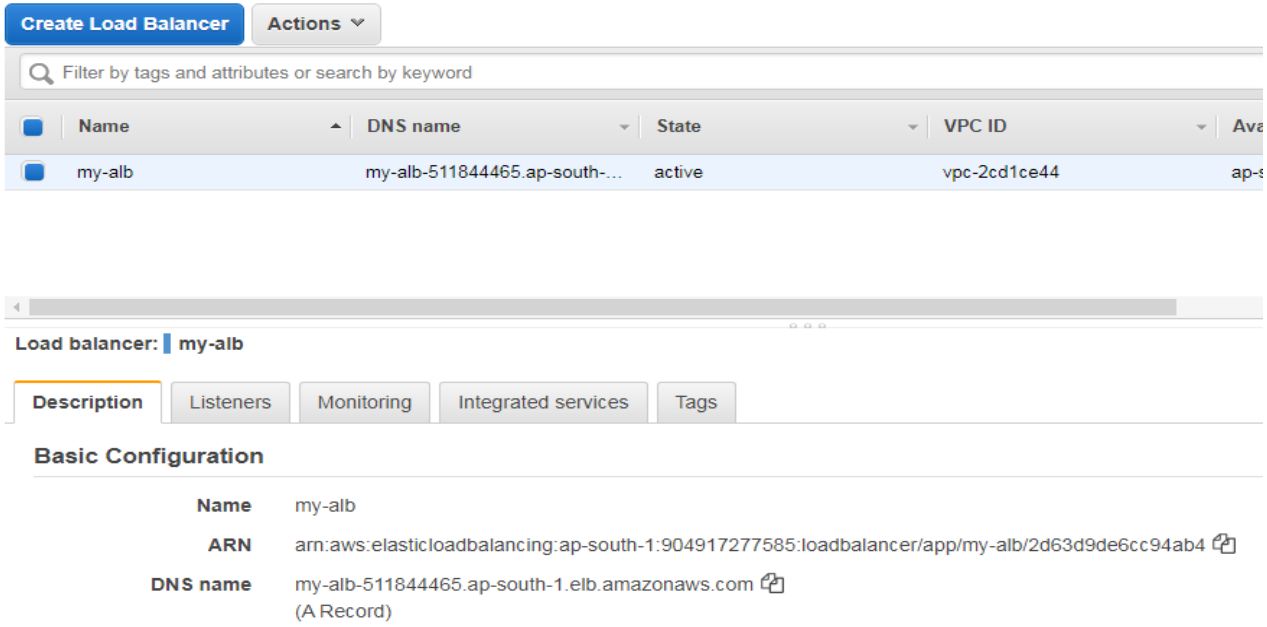
4) Configure condition: After ALB creation ---click on listeners---view/edit rules --- Add rule –Insert rule—

Add condition –Path --*mobile* -- click on tick mark

Add Action—Forward to –Mobile target -- click on tick mark

Save

Do same for Cloth and cosmetic server also



Steps to configure ALB

5) Copy ALB DNS and check

Come back to ALB –Click on Description –copy DNS name

And Paste in New Tab

- ALB/mobile.html
- ALB/cloth.html
- ALB/cosmetic.html

AWS Global Accelerator

- AWS Global Accelerator is a networking service that helps you improve the availability and performance of the applications that you offer to your global users.
- AWS Global Accelerator is easy to set up, configure, and manage. It provides static IP addresses that provide a fixed entry point to your applications and eliminate the complexity of managing specific IP addresses for different AWS Regions and Availability Zones.
- AWS Global Accelerator always routes user traffic to the optimal endpoint based on performance, reacting instantly to changes in application health, your user's location, and policies that you configure.
- You can test the performance benefits from your location with a speed comparison tool.

What does the term sticky session mean

- ✓ Sticky session refers to the feature of many commercial load balancing solutions for web-farms to route the requests for a particular session to the same physical machine that serviced the first request for that session.

- ✓ This is mainly used to ensure that a in-proc session is not lost as a result of requests for a session being routed to different servers.

- ✓ Since requests for a user are always routed to the same machine that first served the request for that session, sticky sessions can cause uneven load distribution across servers.

CLB vs ALB

1) CLB is used for small database size

2) ALB is used for huge database size

3) CLB use monolithic architecture (all services in same system)

4) ALB use micro services architecture (different services in different system)

5) CLB can be configure with all instance in same region

6) ALB can be configure with all instance in same or different region