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)
- Scalability is linked but different to High Availability

Vertical Scalability

- Vertically scalability means increasing the size of the instance
- For example, your application runs on a t2.micro
- Scaling that application vertically means running it on a t2.large
- Vertical scalability is very common for non distributed systems, such as a database.
- RDS, ElastiCache are services that can scale vertically.
- There's usually a limit to how much you can vertically scale (hardware limit)

Horizontal Scalability

- Horizontal Scalability means increasing the number of instances / systems for your application
- Horizontal scaling implies distributed systems.
- This is very common for web applications / modern applications
- It's easy to horizontally scale such as Amazon EC2

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
- The high availability can be passive (for RDS Multi AZ for example)
- The high availability can be active (for horizontal scaling)

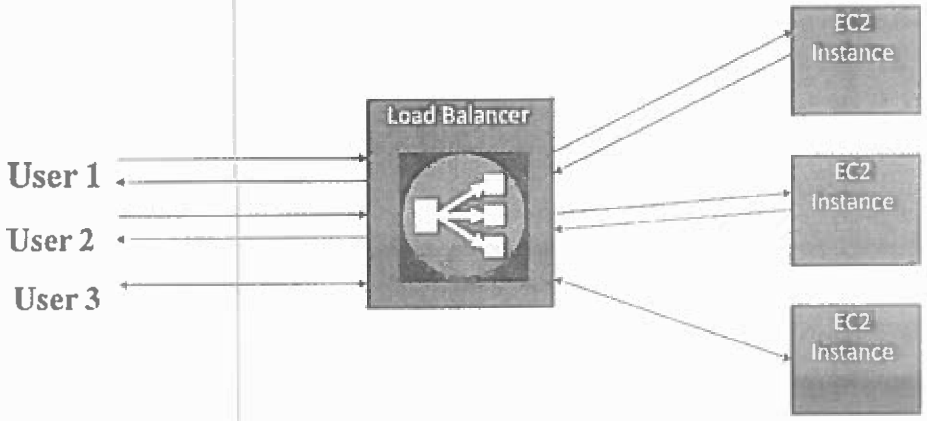
High Availability & Scalability For EC2

- 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, 448 vCPUs
- 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

15

What is load balancing?

- Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream.



16

Why use a load balancer?

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

17

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

18

Types of load balancer on AWS

- AWS has 3 kinds of Load Balancers

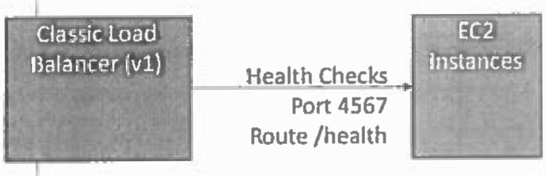
- Classic Load Balancer (v1 - old generation) - 2009
- Application Load Balancer (v2 - new generation) - 2016
- Network Load Balancer (v2 - new generation) - 2017
- Overall, it is recommended to use the newer / v2 generation load balancers as they provide more features

- You can setup internal (private) or external (public) ELBs

19

Health Checks

- 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



20

Application Load Balancer (v2)

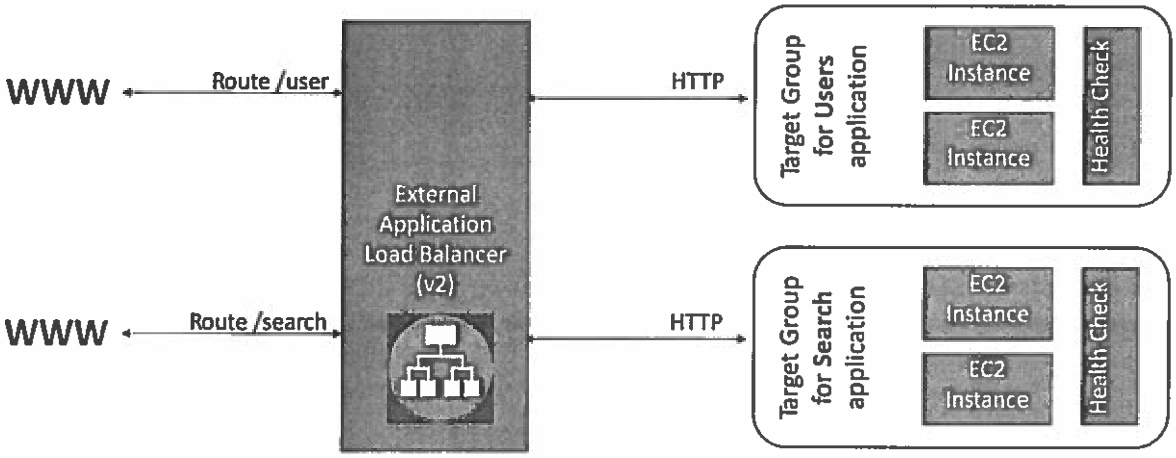
- Application load balancers (Layer 7) allow to do:
  - Load balancing to multiple HTTP applications across machines (target groups)
  - Load balancing to multiple applications on the same machine (ex: containers)
  - Load balancing based on route in URL
  - Load balancing based on hostname in URL

- Basically, they're awesome for micro services & container-based application (example: Docker & Amazon ECS)
- Has a port mapping feature to redirect to a dynamic port

- In comparison, we would need to create one Classic Load Balancer per application before.That was very expensive and inefficient!

21

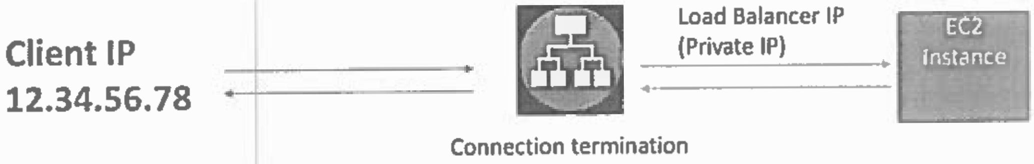
Application Load Balancer (v2) HTTP Based Traffic



22

Application Load Balancer v2 Good to Know

- 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 & Websockets protocols
- The application servers don't see the IP of the client directly
  - The true IP of the client is inserted in the header X-Forwarded-For
  - We can also get Port (X-Forwarded-Port) and proto (X-Forwarded-Proto)



Client IP
12.34.56.78

Load Balancer IP
(Private IP)

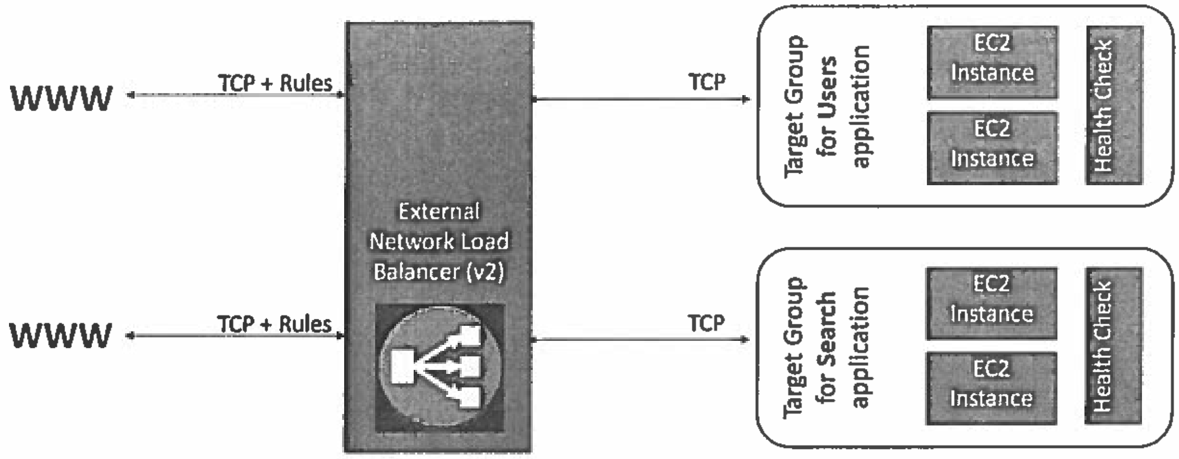
EC2
Instance

Connection termination

Network Load Balancer (v2)

- Network load balancers (Layer 4) allow to do:
  - Forward TCP traffic to your instances
  - Handle millions of request per seconds
  - Support for static IP or elastic IP
  - Less latency ~100 ms (vs 400 ms for ALB)
- Network Load Balancers are mostly used for extreme performance and should not be the default load balancer you choose
- Overall, the creation process is the same as Application Load Balancers

Network Load Balancer (v2) TCP Based Traffic



25

Load Balancer Good to Know

- Classic Load Balancers are Deprecated
  - Application Load Balancers for HTTP / HTTPs & Websocket
  - Network Load Balancer for TCP
- CLB and ALB support SSL certificates and provide SSL termination
- All Load Balancers have health check capability
- ALB can route on based on hostname / path
- ALB is a great fit with ECS (Docker)

26

Load Balancer Good to Know

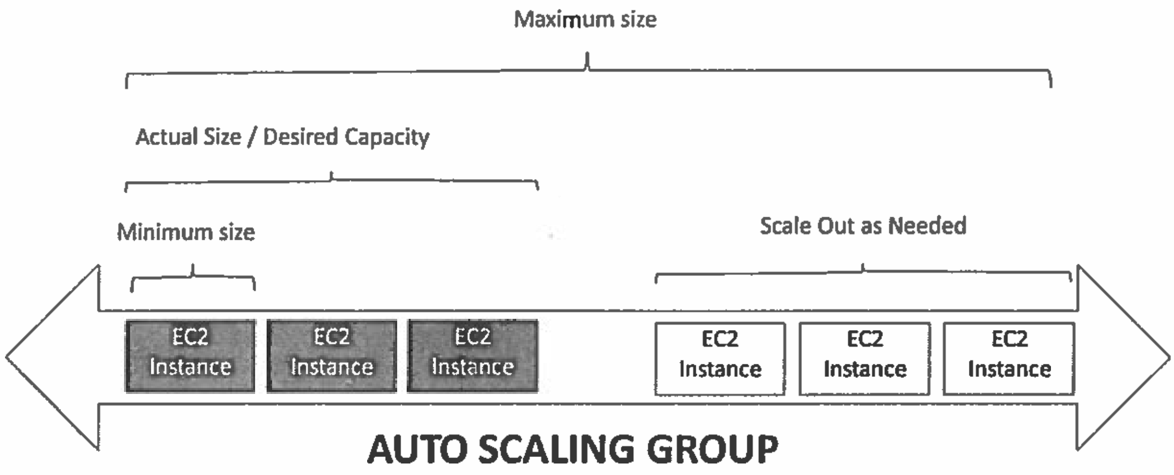
- Any Load Balancer (CLB, ALB, NLB) has a static host name. Do not resolve and use underlying IP
- LBs can scale but not instantaneously – contact AWS for a “warm-up”
- NLB directly see the client IP
- 4xx errors are client induced errors
- 5xx errors are application induced errors
  - Load Balancer Errors 503 means at capacity or no registered target
- If the LB can’t connect to your application, check your security groups!

What’s an Auto Scaling Group?

- In real-life, the load on your websites and application can change
- In the cloud, you can create and get rid of servers very quickly

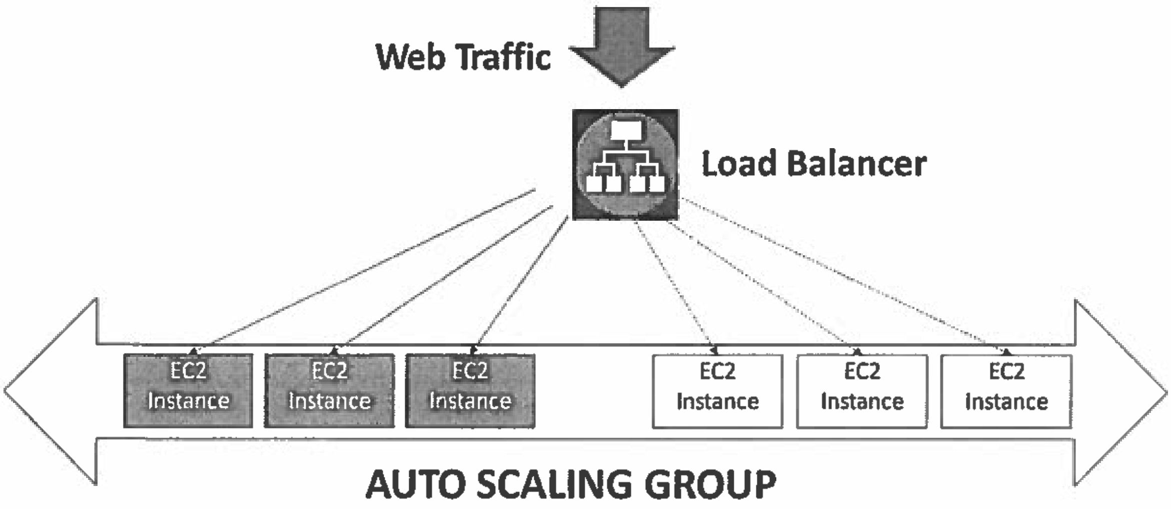
- The goal of an Auto Scaling Group (ASG) is to:
  - Scale out (add EC2 instances) to match an increased load
  - Scale in (remove EC2 instances) to match a decreased load
  - Ensure we have a minimum and a maximum number of machines running
  - Automatically Register new instances to a load balancer

Auto Scaling Group in AWS



29

Auto Scaling Group in AWS With Load Balancer



30

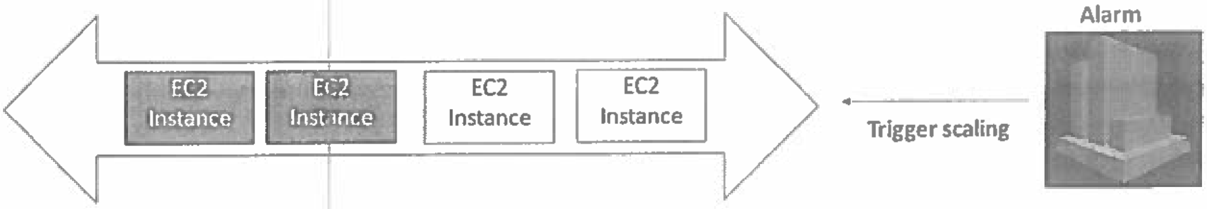
ASGs have the following attributes

- A launch configuration
  - AMI + Instance Type
  - EC2 User Data
  - EBS Volumes
  - Security Groups
  - SSH Key Pair
- Min Size / Max Size / Initial Capacity
- Network + Subnets Information
- Load Balancer Information
- Scaling Policies

31

Auto Scaling Alarms

- It is possible to scale an ASG based on CloudWatch alarms
- An Alarm monitors a metric (such as Average CPU)
- Metrics are computed for the overall ASG instances
- Based on the alarm:
  - We can create scale-out policies (increase the number of instances)
  - We can create scale-in policies (decrease the number of instances)



32

Auto Scaling New Rules

- It is now possible to define ”better” auto scaling rules that are directly managed by EC2
  - Target Average CPU Usage
  - Number of requests on the ELB per instance
  - Average Network In
  - Average Network Out
- These rules are easier to set up and can make more sense

33

ASG Brain dump

- Scaling policies can be on CPU, Network… and can even be on custom metrics or based on a schedule (if you know your visitors patterns)
- ASGs use Launch configurations and you update an ASG by providing a new launch configuration
- IAM roles attached to an ASG will get assigned to EC2 instances
- ASG are free.You pay for the underlying resources being launched
- Having instances under an ASG means that if they get terminated for whatever reason, the ASG will restart them. Extra safety!
- ASG can terminate instances marked as unhealthy by an LB (and hence replace them)

34