Elastic Load Balancing & Auto Scaling Groups

Scalability & High Availability

- **Scalability**: Ability of a system to handle an increase in load by adapting to the demand.
- **High Availability**: Ensures a system is operational and accessible for a high percentage of time, often achieved by reducing the impact of failures.
- There are two kinds of scalability:
  - Vertical Scalability
  - Horizontal Scalability (= elasticity)
- Scalability is linked but different to High Availability

Vertical Scalability

- Increasing the capacity of a single instance (e.g., moving from t3.medium to t3.large).
- Suitable for databases or applications where upgrading a single resource is more efficient.
- Limited by hardware constraints (can only scale up to a certain point).

Horizontal Scalability

- Adding more instances (servers) to distribute the load across multiple resources.
- Achieved through technologies like **Auto Scaling Groups (ASG)** and **Elastic Load Balancing (ELB)**.
- Preferred for applications needing resilience and distributed workloads.
- Horizontal scaling implies distributed systems.

High Availability

- Implemented by deploying resources across multiple **Availability Zones** (AZs).
- Ensures failover and redundancy in case of failures in one AZ.
- High Availability usually goes hand in hand with 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

Scalability vs Elasticity (vs Agility)

Term	Definition
Scalability	Ability to increase or decrease the capacity to handle varying levels of traffic or load.
Elasticity	Automatically adjusts resources up or down based on the load in real-time, preventing under or over-provisioning.

Term	Definition
Agility	The ability to deploy and manage resources quickly and efficiently in response to changing demands.

What is Load Balancing?

- Distributes incoming traffic across multiple targets (EC2 instances, containers, IP addresses) to ensure that no single resource is overwhelmed.

Why Use a Load Balancer?

- Ensures application fault tolerance and high availability by spreading the load across multiple servers.
- Protects against failures in a single resource by rerouting traffic automatically.
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites

Why Use an Elastic Load Balancer?

- **Elastic Load Balancer (ELB)** is a fully managed service that automatically distributes incoming application traffic across multiple targets in one or more Availability Zones.
- It improves fault tolerance, enhances performance, and scales according to demand.
- AWS guarantees that it will be working
- AWS takes care of upgrades, maintenance, high availability
- AWS provides only a few configuration knobs

Types of ELB

1. **Application Load Balancer (ALB)**: For HTTP and HTTPS traffic, operates at Layer 7 (application level).
2. **Network Load Balancer (NLB)**: Handles high-performance traffic at Layer 4 (transport level).
3. **Classic Load Balancer**: (slowly retiring) – Layer 4 & 7

What's an Auto Scaling Group?

- An **Auto Scaling Group (ASG)** ensures the right number of EC2 instances are running to handle the load.
- Automatically adjusts the number of instances based on metrics such as CPU utilization or custom-defined thresholds.
- Can span across multiple AZs to ensure high availability.
- 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
  - Replace unhealthy instances
- Cost Savings: only run at an optimal capacity (principle of the cloud)

Auto Scaling Group Scaling Strategies

- **Manual Scaling**: Adjusting the number of instances manually based on load prediction.
- **Dynamic Scaling**: Automatically adjusts the number of instances based on demand (e.g., CPU usage).
  - Simple / Step Scaling
    - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
    - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1
  - Target Tracking Scaling
    - Example: I want the average ASG CPU to stay at around 40%
  - Scheduled Scaling
    - Anticipate a scaling based on known usage patterns
    - Example: increase the min. capacity to 10 at 5 pm on Fridays
- **Predictive Scaling**: Uses machine learning to predict future traffic patterns and scales proactively.

ELB & ASG Summary

- High Availability vs Scalability (vertical and horizontal) vs Elasticity vs Agility in the Cloud
- Elastic Load Balancers (ELB)

  - Distribute traffic across backend EC2 instances, can be Multi-AZ
  - Supports health checks
  - 3 types: Application LB (HTTP – L7), Network LB (TCP – L4), Classic LB (old)
- Auto Scaling Groups (ASG)
  - Implement Elasticity for your application, across multiple AZ
  - Scale EC2 instances based on the demand on your system, replace unhealthy
  - Integrated with the ELB