Day - 05 | Elastic Load Balancing & Auto Scaling Groups | AWS Cloud Practitioner Certification CLF-C02

Created on 2024-07-21 08:08
Published on 2024-07-21 10:44

➤ Elastic Load Balancing & Auto Scaling Groups

→•← Scalability & High Availability

→•← Vertical Scalability

→•← Horizontal Scalability

→•← High Availability

→•← High Availability & Scalability For EC2

→•← Scalability vs Elasticity vs Agility

→•← What is load balancing?

→ Why use a load balancer?

→ Why use an Elastic Load Balancer?

→•← What's an Auto Scaling Group?

→ Auto Scaling Groups Scaling Strategies

→•← ELB & ASG Summary

Scalability & High Availability

Scalability refers to the ability of a system to handle increased load by adding resources.

It can be:

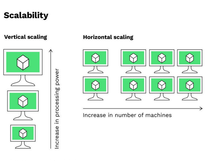
- **Vertical Scalability**: Increasing the capacity of existing resources (e.g., upgrading a server's CPU or memory).
- **Horizontal Scalability** *(= elasticity)*: Adding more instances of resources (e.g., adding more servers to a pool).

Vertical Scalability

Vertical scalability (or scaling up) involves increasing the capacity of a single instance or server. This can be done by adding more CPU, RAM, or storage. 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. 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. This implies distributed systems. It's easy to horizontally scale thanks the cloud offerings such as **Amazon EC2.**



Vertical & Horizontal Scalability

High Availability

High availability ensures that a system remains operational even when some components fail. The goal of high availability is to survive a data center loss (disaster).

This involves:

- **Redundancy**: Having multiple instances of critical components in at least 2 Availability Zones.
- **Failover Mechanisms**: Automatically switching to a standby component upon failure.
- **Load Balancing**: Distributing incoming traffic across multiple instances to prevent overload.

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

Scalability	Elasticity	Agility
ability to accommodate a larger load by making the hardware stronger (scale up), or by adding nodes (scale out)	once a system is scalable, elasticity means that there will be some "auto-scaling" so that the system can scale based on the load. This is "cloud-friendly": pay per-use, match demand, optimize costs	(not related to scalability - distraction) new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes.

Scalability vs Elasticity vs Agility

What is load balancing?

Load balancing is the process of distributing network or application traffic across multiple servers. It ensures no single server becomes a bottleneck, thus improving performance and reliability.

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
- High availability across zones

Why use an Elastic Load Balancer?

Elastic Load Balancing (ELB) is a managed load balancing service provided by AWS.

- **Automatic Scaling**: Automatically scales to handle incoming traffic.
- **Health Checks**: Monitors the health of registered instances and routes traffic only to healthy instances.
- **Integration with ASG**: Works seamlessly with Auto Scaling Groups to provide high availability and fault tolerance.
- **Security**: Supports integration with AWS Certificate Manager (ACM) for SSL termination, and can work with AWS Web Application Firewall (WAF) for enhanced security.
- 3 kinds of load balancers offered by AWS:

•• Application Load Balancer (HTTP / HTTPS only) – Layer 7

•• Network Load Balancer (ultra-high performance, allows for TCP) – Layer 4

•• Classic Load Balancer (slowly retiring) – Layer 4 & 7

What's an Auto Scaling Group?

An Auto Scaling Group (ASG) is a collection of EC2 instances that are treated as a logical group for the purposes of scaling and management.

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

Auto Scaling Groups Scaling Strategies

- **Manual Scaling**: Manually adjust the number of instances based on anticipated demand.
- **Scheduled Scaling**: Scale based on a predefined schedule.
- **Dynamic Scaling**: Automatically adjust based on real-time metrics such as CPU utilization or network traffic. Types include:

▫▫ Target Tracking Scaling: Maintain a target value for a specific metric (e.g., average CPU usage).

•• I want the average ASG CPU to stay at around 40%

▫▫ Scheduled Scaling: Scale in steps based on specified thresholds.

•• Anticipate a scaling based on known usage patterns

•• Example: increase the min. capacity to 10 at 5 pm on Fridays

▫▫ Simple / Step Scaling: Add or remove instances when a specific condition is met.

•• 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

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

Happy Learning !