Elastic Load Balancing and Auto Scaling Group

- Hight Availability and Scalability
- Overview on Elastic Load Balancer(ELB)
- Types of Elastic Load Balancer
- Sticky session
- Cross Zone load balancing
- Overview on Auto Scaling group(ASG)
- ASG scaling policy
- Demos

 @TeamShiksha

 @teamShiksha

 @teamshiksha

Hight Availability and Scalability

- **Scalability** mean that an application or system can hanlde great loads by adapting. There are two kind of scalability: Vertical and Horizontal(also called elasticity)(Scale up/down)
- **Vertical** scalability means, you will increase the size of your instance(server). Example: changing server from t2.micro to m5.2xlarge. (common for non-distributed system)
- **Horizontal** scalability means you will increase the number of instances. Exmaple: You had two t2.micro and now you have moved to 10 t2.micro instances. (common for web applications, easy). (Scale in/out)
- **High availability** mean running your application in more than 1 data centers(Availabiity Zone)
- The main purpose of high availability is to survivie data center loss.

Overview on Elastic Load Balancer(ELB)

- Load balancers are servers that forward traffic to multiple servers downstream
- Spread load across multiple downstream instances(servers)
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances(check the route for health of an instance, then tansfers traffic)
- Enforce stickiness with cookies
- High availability across zones
- Integrated with many AWS offerings / services
  - EC2, ASG, ECS
  - AWS Certificate Manager, CloudWatch
  - Route 53, AWS WAF, AWS Global Accelerator
- An Elastic load balancer is a managed load balancer:
  - AWS guarantee that it will be working
  - AWS takes care of upgrades, maintenance, high availability

Types of Elastic Load Balancer

Application Load Balancer

- Layer 7(Application layer of OSI model)(HTTP and HTTPS only)
- Load balancing to multiple HTTP application on different machines(target group)
- Load balacing to mutiple application on the same machine (container)
- Support to HTTP/2, websocket and redirect from http to https
- Routing based path, hostname and query string on different target group
- Target groups: EC2 instances, ECS, Ip addresses and Lambda
- ALB can also route to multiple target groups
- Good to know:
  - Fixed host name (XXX.region.elb.amazons.com)
  - The application servers don’t see the IP of the client directly, use header X-Forwarded-For for true IP address

Types of Elastic Load Balancer

Network Load Balancer:

- Layer 4 allows to TCP, TLS and UDP traffic to your instances.
- Low latency
- Handles millions of requests
- Has one static IP per AZ, and support assigning Elastic IP
- Used for extreme performance, TCP or UDP traffic
- Not included in the AWS free tier
- Target groups: EC2, IP and ALB


@TeamShiksha


@teamShiksha


@teamshiksha

Types of Elastic Load Balancer

Gateway Load Balancer

- Deploy, scale, and managed a fleet of 3rd party network virtual appliances in AWS. Example: Firewalls, Intrusion Detection and Prevention Systems, Deep Packet Inspection Systems, Payload manipulation
- Operates at Layer 4 (Network Layer) - IP Packets
- Uses the GENEVE protocol on port 6081
- Target group EC2 and IP address
- Combine the following functions:
  - Transparent Network Gateway - Single entry/exit for all traffic
  - Load Balancer - distributes traffic to your virtual appliances

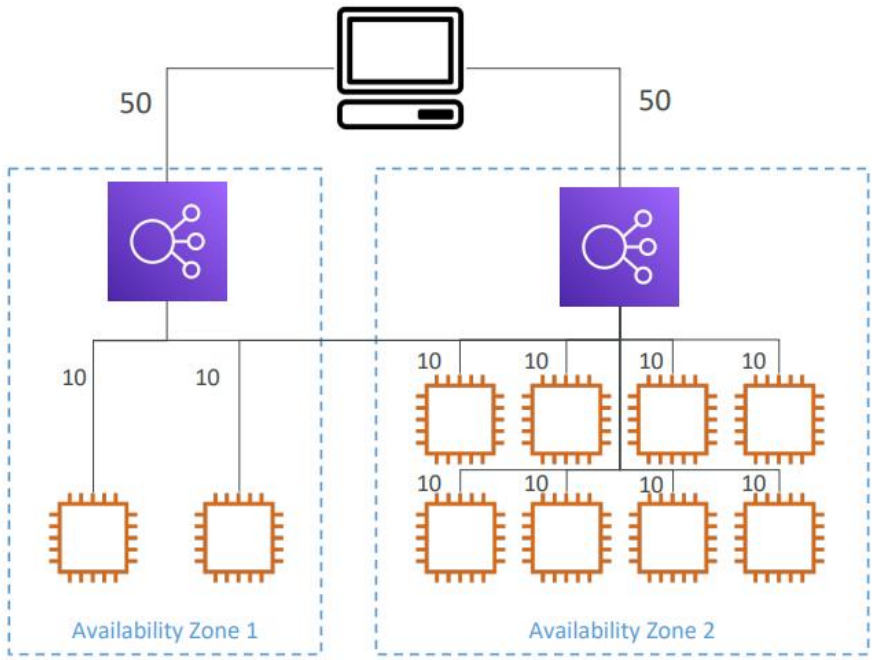
Sticky session

- Implement stickiness so that the same client is always redirected to the same instance behind a load balancer.
- Works for ALB and NLB
- The “cookie” used for stickiness has an expiration date you control
- Enabling stickiness may bring imbalance to the load over the backend EC2 instances
- Use case: make sure the user doesn’t lose his session data
- Two types of cookies:
  - **Application based cookies**
    - Generated by the load balancer
  - **Duration based cookies**:
    - Cookie generated by the load balancer
    - Do not use AWSALB, AWSALBAPP, or AWSALBTG

Cross Zone load balancing Diagram

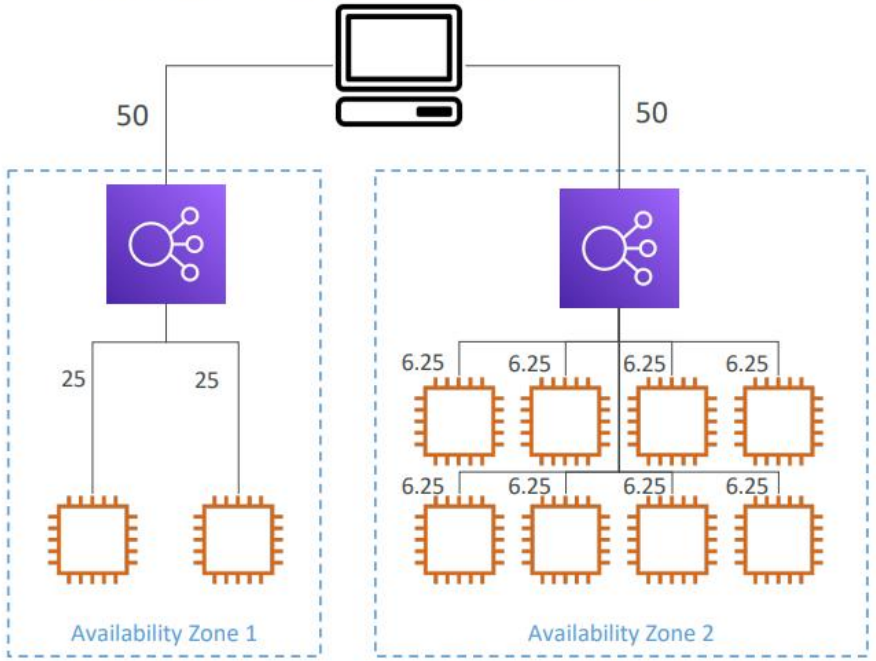
With Cross Zone Load Balancing:

each load balancer instance distributes evenly across all registered instances in all AZ



Without Cross Zone Load Balancing:

Requests are distributed in the instances of the node of the Elastic Load Balancer



@TeamShiksha @teamShiksha @teamshiksha

Cross Zone load balancing

- Each load balancer instances distributes evenly across all registered instance in all AZ.
- Application load balancer:
  - Enabled by default(can be disabled at the Target Group Level)
  - No charges for inter AZ data
- Network and Gateway Load Balancer
  - Disabled by default
  - You pay charges($) for inter AZ data if enabled
- Use case: make sure the user doesn't lose his session data
- **Connection Draining OR Deregistration Delay**
  - Time or complete "in-flight requests" while the instance is re-registering unhealthy

Overview on Auto Scaling group(ASG)

- In real-life, the load on your website and application can change and 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 instance) to match an increased load
  - **Scale in**(Remove EC2 instance) to match a decrease load
  - Ensure we have minimum and a maximum number of EC2 instance running
  - Automatically register new instances to a load balancer
  - Re-create an EC2 instance in case a previous one is terminated
- ASG are free, you only pay for underlying resources.
- Attributes: These attributes need to be set for ASG to work properly


@TeamShiksha


@teamShiksha


@teamshiksha

ASG scaling policy

- **Predictive scaling**: continuously forecast load and schedule scaling ahead
- **Dynamic Scaling**
  - Target Tracking Scaling
    - Simple to set-up
    - Example: I want the average ASG CPU to stay at around 40%
  - 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 I
- **Scheduled Scaling**
  - Anticipate a scaling based on known usage patterns
  - Example: increase the min capacity to 10 at 5 pm on Fridays
- **Good metrics to scale on**: CPUtilization, RequestCountPerTarget, Network In/Out, Custom metrics
- **Cool Down period**: After every scaling activity you are in a cool down period of 5 minutes


@TeamShiksha


@teamShiksha

@teamshiksha

MCQ

Scaling an EC2 instance from r4.large to r4.4xlarge is called .....

A. Horizontal scaling
B. Vertical scaling
C. This is not scaling
D. None of the above


@TeamShiksha


@teamShiksha


@teamshiksha

MCQ

You are running a website on 10 EC2 instances fronted by an Elastic Load Balancer. Your users are complaining about the fact that the website always asks them to re-authenticate when they are moving between website pages. You are puzzled because it's working just fine on your machine and in the Dev environment with 1 EC2 instance. What could be the reason?

A. Your website must have an issue when hosted on multiple EC2 instances
B. The EC2 instances log out users as they can't see their IP addresses, instead, they receive
C. ELB IP addresses.
D. The Elastic Load Balancer does not have Sticky Sessions enabled
E. None of the above


@TeamShiksha


@teamShiksha


@teamshiksha

MCQ

You are working as a developer for a company and you are required to design an architecture for a high-performance, low-latency application that will receive millions of requests per second. Which type of Elastic Load Balancer should you choose?

A. Network Load Balancer
B. Application Load Balancer
C. Gateway or Classic Load Balancer

MCQ

You have 10 instance running in a region(ap-south-1), 2 of them are running in ap-south-1a AZ and 8 of them are running in ap-south-1b AZ. If you are using an Application Load Balancer and receiving a constant traffic. What would be the percentage load on each instance in ap-south-1a AZ.

A. 10 %
B. 25 %
C. 20 %
D. Application load balancer only works in single AZ