Elastic Load Balancing

Network Load Balancing

- Network Load Balancer is best suited for use-cases involving low latency and high throughput workloads that involve scaling to millions of requests per second.
- Network Load Balancer operates at the connection level (Layer 4), routing connections to targets - Amazon EC2 instances, microservices, and containers – within Amazon Virtual Private Cloud (Amazon VPC) based on IP protocol data.
- Network Load Balancers expose a fixed IP to the public web, therefore allowing your application to be predictably reached using these IPs,

Application Load Balancing

- Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers, IP addresses and Lambda functions based on the content of the request.
- Ideal for advanced load balancing of HTTP and HTTPS traffic
- It support dynamic port routing
- Application Load Balancer provides advanced request routing targeted at the delivery of modern application architectures, including microservices and container-based applications.
- Application and Classic Load Balancers expose a fixed DNS (=URL) rather than the IP address.
- An ALB cannot block or allow traffic based on geographic match conditions or IP based conditions.
- ALB supports authentication from OIDC compliant identity providers such as Google, Facebook and Amazon. It is implemented through an authentication action on a listener rule that integrates with Amazon Cognito to create user pools.

Gateway Load Balancer

- Gateway Load Balancers enable you to deploy, scale, and manage virtual appliances, such as firewalls, intrusion detection and prevention systems, and deep packet inspection systems.
- It combines a transparent network gateway (that is, a single entry and exit point for all traffic) and distributes traffic while scaling your virtual appliances with the demand.

- A Gateway Load Balancer operates at the third layer of the Open Systems Interconnection (OSI) model, the network layer.
- It listens for all IP packets across all ports and forwards traffic to the target group that's specified in the listener rule.
- It maintains stickiness of flows to a specific target appliance using 5-tuple (for TCP/UDP flows) or 3-tuple (for non-TCP/UDP flows).
- uses GENEVE Protocol on port 6081.
- Tareget group EC2 Instance and Private IP adresses

Sticky Session

- Sticky Sessions on your ALB is a distractor here. Sticky sessions are a mechanism to route requests from the same client to the same target.
- Application Load Balancer supports sticky sessions using load balancer generated cookies.
- If you enable sticky sessions, the same target receives the request and can use the cookie to recover the session context.

Connection Draining

- **To ensure that a Classic Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open**, use connection draining.
- This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.
- Between 0 to 3600 second

SNI

- SNI solve the problem of loading multiple SSL certificate onto one browser.
- You can now host multiple TLS secured applications, each with its own TLS certificate, behind a single load balancer. In order to use SNI, all you need to do is bind multiple certificates to the same secure listener on your load balancer.
- for ALB and NLB
- ALB will automatically choose the optimal TLS certificate for each client. These new features are provided at no additional charge.

Auto Scaling Group

- ASG does not have a dynamic Elastic IPs attachment feature.

- An Auto Scaling group also enables you to use Amazon EC2 Auto Scaling features such as health check replacements and scaling policies.
- Both maintaining the number of instances in an Auto Scaling group and automatic scaling are the core functionality of the Amazon EC2 Auto Scaling service
- You configure the size of your Auto Scaling group by setting the minimum, maximum, and desired capacity.
- If you do not define your desired capacity upfront, it defaults to your minimum capacity.

ASG Cooldown

- The cooldown period is a configurable setting for your Auto Scaling group that helps to ensure that it doesn't launch or terminate additional instances before the previous scaling activity takes effect so this would help.
- After the Auto Scaling group dynamically scales using a simple scaling policy, it waits for the cooldown period to complete before resuming scaling activities.
- After your Auto Scaling group launches or terminates instances, it waits for a cooldown period to end before any further scaling activities initiated by simple scaling policies can start.
- The intention of the cooldown period is to prevent your Auto Scaling group from launching or terminating additional instances before the effects of previous activities are visible.

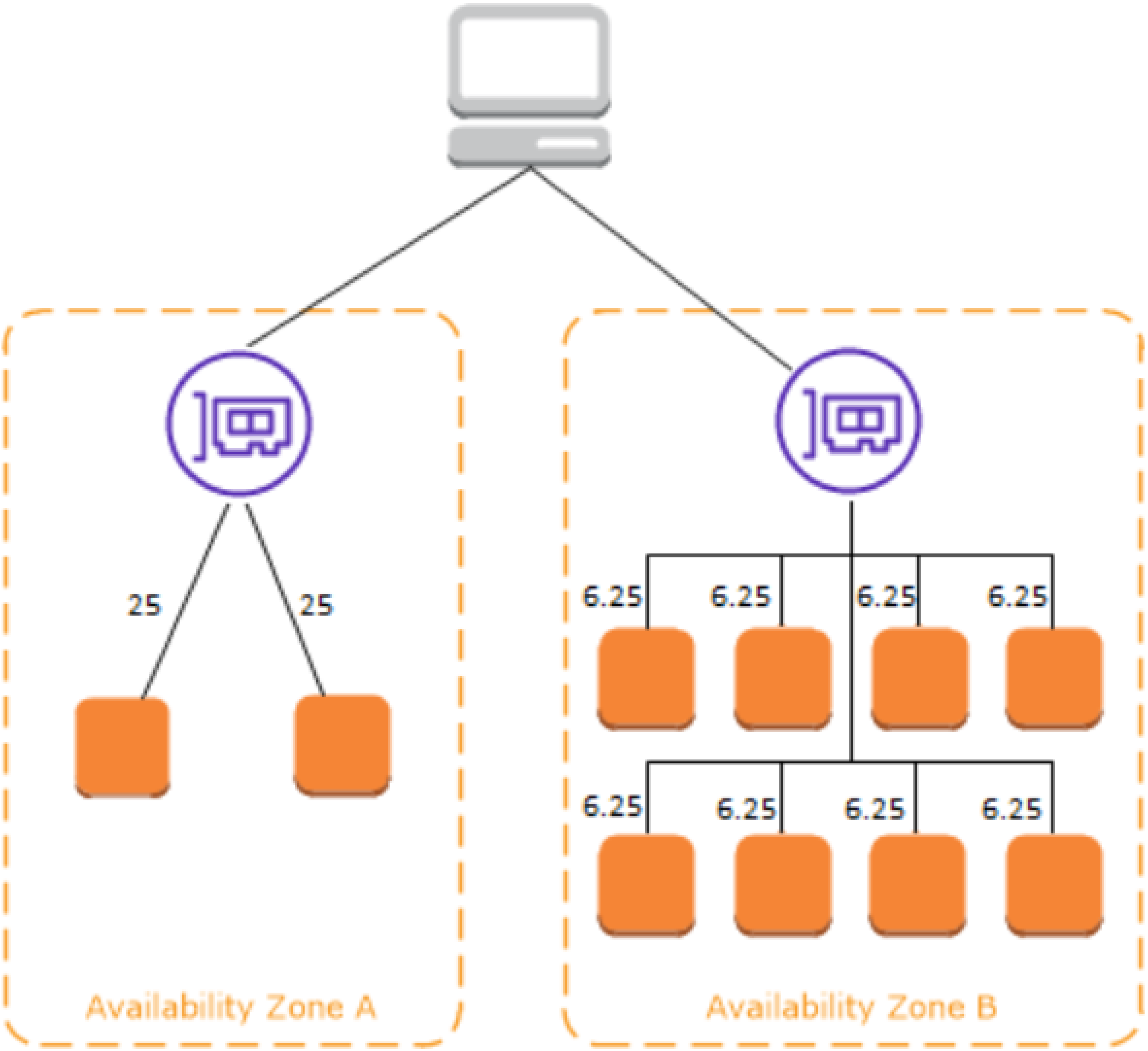
Policy for cloudwatch

- **Auto Scaling group scheduled action**
  - QUES ::::: The EC2 server fleet is behind an Application Load Balancer and the fleet strength is managed by an Auto Scaling group. Based on the historical data, the team is anticipating a huge traffic spike during the upcoming Thanksgiving sale.
    - The engineering team can create a scheduled action for the Auto Scaling group to pre-emptively provision additional instances for the sale duration.
    - This makes sure that adequate instances are ready before the sale goes live.
    - The scheduled action tells Amazon EC2 Auto Scaling to perform a scaling action at specified times.
    - To create a scheduled scaling action, you specify the start time when the scaling action should take effect, and the new minimum, maximum, and desired sizes for the scaling action.
    - At the specified time, Amazon EC2 Auto Scaling updates the group with the values for minimum, maximum, and desired size that are specified by the scaling action.
- Auto Scaling group target tracking scaling policy

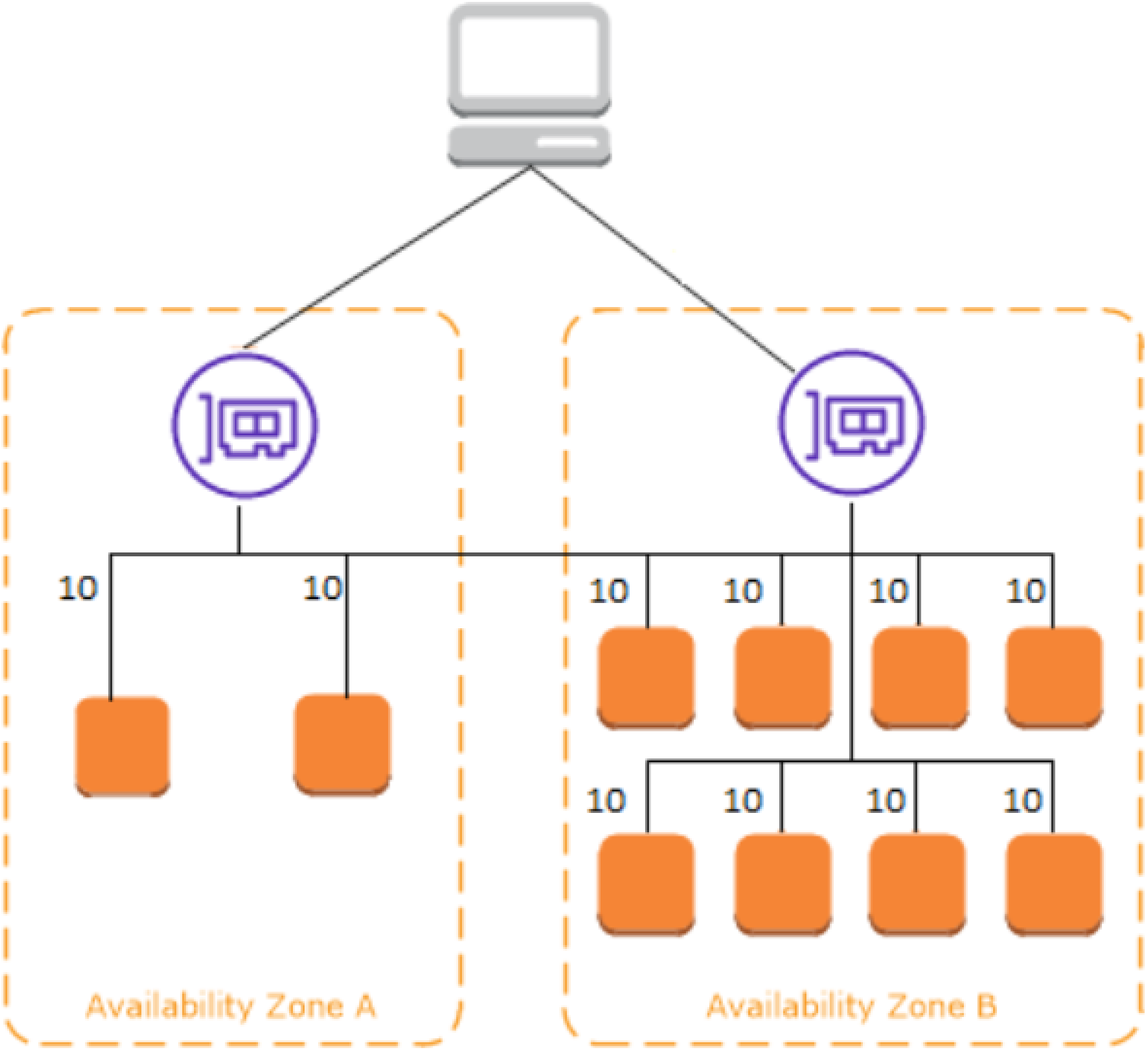
  - With target tracking scaling policies, you choose a scaling metric and set a target value.
  - Application Auto Scaling creates and manages the CloudWatch alarms that trigger the scaling policy and calculates the scaling adjustment based on the metric and the target value.
- Auto Scaling group step scaling policy
  - With step scaling, you choose scaling metrics and threshold values for the CloudWatch alarms that trigger the scaling process as well as
  - define how your scalable target should be scaled when a threshold is in breach for a specified number of evaluation periods.
  - The step scaling policy makes scaling adjustments based on a number of factors.
  - The PercentChangeInCapacity value increments or decrements the group size by a specified percentage. This does not relate to CPU utilization.
- Auto Scaling group Simple scaling policy
  - A simple scaling policy will add instances when 40% CPU utilization is reached, but it is not designed to maintain 40% CPU utilization across the group.
- Auto Scaling group lifecycle hook
  - Auto Scaling group lifecycle hooks enable you to perform custom actions as the Auto Scaling group launches or terminates instances.
  - For example, you could install or configure software on newly launched instances, or download log files from an instance before it terminates.
  - Lifecycle hooks cannot be used to pre-emptively provision additional instances for a specific period such as the sale duration.
- You can specify which subnets Auto Scaling will launch new instances into. Auto Scaling will try to distribute EC2 instances evenly across AZs. If only one subnet has EC2 instances running in it the first thing to check is that you have added all relevant subnets to the configuration

Cross Load Balancing

- ith *cross-zone load balancing*, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones.
- If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.
- for appliaction load balancer it is always enabled and does not have charges for inter AZ
- for network load balancer it disabled you pay charges for inter AZ if enables



with cross zone disable



with cross zone load balancing enabled

Health Check

- To better Understand heatlh check, check this example:
- QUES => A streaming solutions company is building a video streaming product by using an Application Load Balancer (ALB) that routes the requests to the underlying EC2 instances. The engineering team has noticed a peculiar pattern. The ALB removes an instance from its pool of healthy instances whenever it is detected as unhealthy but the Auto Scaling group fails to kick-in and provision the replacement instance.What could explain this anomaly?
  - **The Auto Scaling group is using EC2 based health check and the Application Load Balancer is using ALB based health check**
  - If the Auto Scaling group (ASG) is using EC2 as the health check type and the Application Load Balancer (ALB) is using its in-built health check, there may be a

situation where the ALB health check fails because the health check pings fail to receive a response from the instance.

  - At the same time, ASG health check can come back as successful because it is based on EC2 based health check.
  - Therefore, in this scenario, the ALB will remove the instance from its inventory, however, the ASG will fail to provide the replacement instance. This can lead to the scaling issues mentioned in the problem statement.
- NOTE -> ALB cannot use EC2 based health checks
- It is recommended to use ALB based health checks for both Auto Scaling group and Application Load Balancer. If both the Auto Scaling group and Application Load Balancer use ALB based health checks, then you will be able to avoid the scenario mentioned in the question.

Important Things

1.

- If any health check returns an unhealthy status the instance will be terminated.
- For the “impaired” status, the ASG will wait a few minutes to see if the instance recovers before taking action. If the “impaired” status persists, termination occurs.
- Unlike AZ rebalancing, termination of unhealthy instances happens first, then Auto Scaling attempts to launch new instances to replace terminated instances.

2.

- When rebalancing, Amazon EC2 Auto Scaling launches new instances before terminating the old ones, so that rebalancing does not compromise the performance or availability of your application.
- the scaling activity of Auto Scaling works in a different sequence compared to the rebalancing activity. Auto Scaling creates a new scaling activity for terminating the unhealthy instance and then terminates it. Later, another scaling activity launches a new instance to replace the terminated instance.