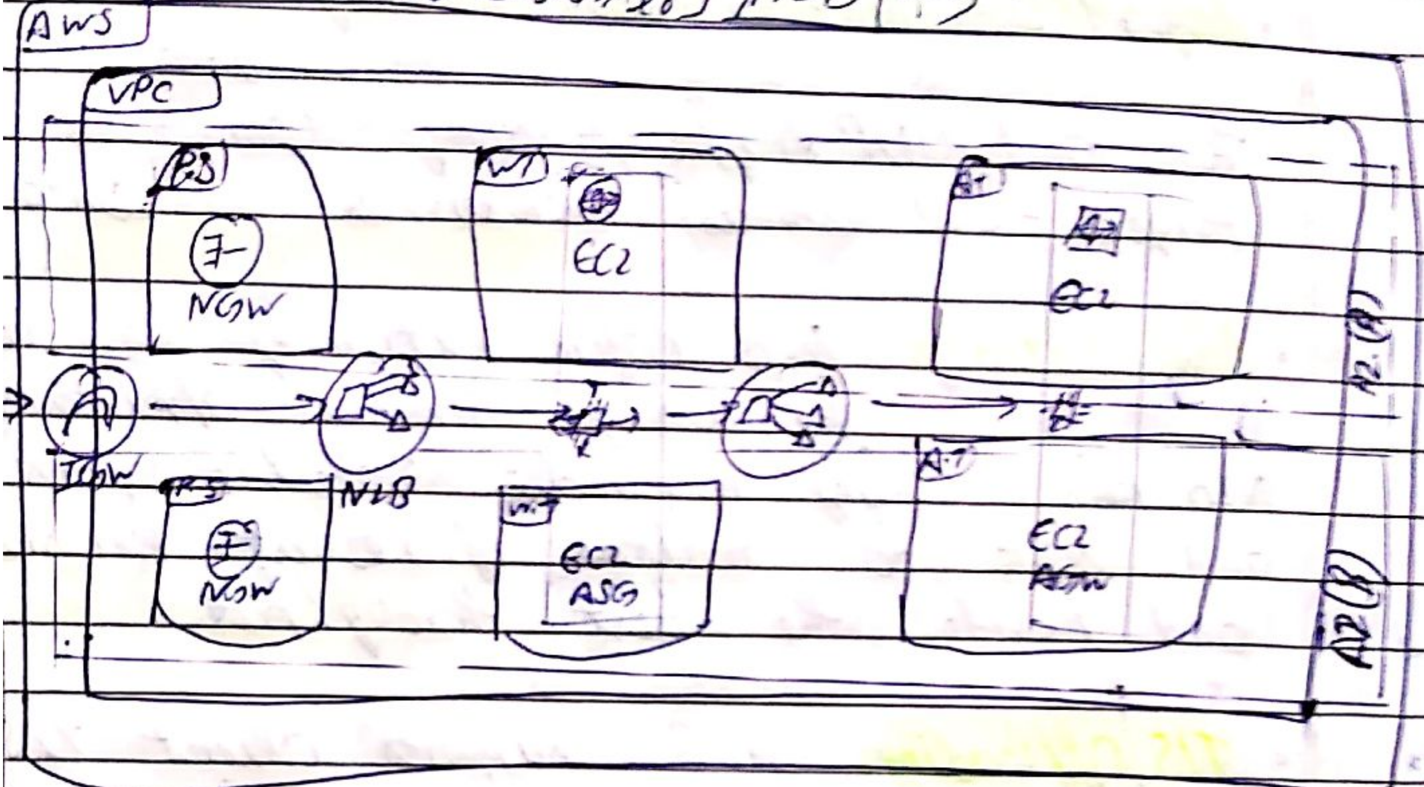
Day 10/100 AWS

AWS - [Network Load Balancer] NLB →



AWS - NLB

NLB works at the layer4 [Transport layer Connection level) of the OSI model. NLB supports load balancing of application using TCP, UDP and TCP_UDP listeners, as well as TLS listeners

NLB is specifically designed for high performance traffic that is not Conventional web traffic. NLB is capable of handling millions of req. per second while maintaining ultra-low latencies

Features:

- Layer 4 (Connection-based) Load Balancing: You can load balance both TCP and UDP traffic, routing connections to targets - EC2 instances, microservices and containers.

- Low Latency: NLB offers extremely low latencies for latency - sensitive application. NLB has ability to handle volatile workloads and scale to millions of request per second and provide the best throughput.

- TLS Offloading: NLB supports client TLS session termination (encryption & decryption). This enables you to offload TLS termination task to the load balancer, while preserving the source IP address for your back-end application.

- Preserve Source IP address: NLB preserves that client side source IP allowing back-end to see the IP address of the client. This can then be used by application for further processing.

- Static IP Support ⇒ NLB automatically provides a static IP per Availability Zone ~~8s~~ (subnet) that can be used by application as the front end IP of the load balancer.

- Elastic IP support - NLB also allows you the option to assign as Elastic IP per Availability Zone (subnet) there by providing your own fixed IP.

- Sticky Sessions - Sticky sessions (source IP affinity) are mechanism to route requests from the same client to the same target. Stickiness is defined at the target group level.

- Long-lived TCP Connections - NLB supports long-lived TCP connection that are ideal for webSocket type of application

- Zonal Isolation - NLB can be enabled in a single Availability Zone to support architectures that require zonal isolation. NLB is designed for ~~[illegible]~~ application architectures in a single zone However, It is recommend to configure the load balancer and target in multiple AZs for achieving high availability

- DNS fail-over - If there are no healthy targets registered with the NLB or if NLB nodes in a given zone are unhealthy, then Amazon Route 53 will direct traffic to load balancer nodes in other AZs.

- Integration with Route53 → In the event that your NLB is unresponsive, integration with Route53 will remove the unavailable load balance IP address from service and direct traffic to the alternate NLB in another region.

- Integration with AWS services - NLB is integrated with other ~~ALB~~ AWS services such as Auto scaling, Elastic Containers Service (ECS), Elastic BeanStalk, AWS Certificate Manager (ACM), CloudWatch Config, CloudTrail, CodeDeploy and Cloudformation

- Central API support - NLB uses the same API as Application load Balancer [ALB]. This will enable you to work with target groups health checks, and load balance across multiple ports on the same EC2 instance to support Containerized application

Key Points:

1. NLB operates at Connection level.

2. You Cannot associate Security Group with NLB.

3. You can select only one Subnet for Availability Zone.

4. NLB is more expensive as Compare to other AWS LB.

5. Both Classic LB and ALB use Connection multiplexing, but NLB dont.

6. NLB support for registering target by IP address, including targets outside the VPC for the load balancer

7. NLB supports for routing requests to multiple applications on a single EC2 instance User can register each instance or IP address with the same target group using multiple ports.

8. NLB with TCP & TLS listeners can be used to setup Privatelink. You cannot setup Privatelink with UDP listener on NLB.

9. You can use Route 53 health checking and DNS failover features to enhance the availability of the application running behind NLB's.

10 You can monitor and analyze traffic patterns & troubleshoot issues with cloud watch metrics , VPC flow logs & cloud Trail logs.

11. For TCP traffic, the NLB selects a target using a flow hash algorithm based on the protocal source IP address, source port, destination IP-address , destination port & TCP sequence number.

12. To support both TCP and UDP on the same port , create a TCP_UDP listener. The target groups for TCP_UDP listener must use the TCP_UDP protocol.

Use Cases

- when You need to seamlessly support spiky or high-volume inbound TCP requests without pre-warming.

- when you need support of static or elastic IP address.

- when You want to support for routing request to multiple application on a single EC2 instance. NLB is well suited to ECS.

- when You need to support an IP address or an IP target outside of the VPC.