Assignments 22nd Sep2024 SavitaNalawade

Difference Between Below topics:

1. NACL vs SH

S.NO	Security Group	NACL(Network Access Control List)
1	Acts as a firewall for associated Amazon EC2 instances.	Acts as a firewall for associated subnets.
2	Controls both inbound and outbound traffic at the instance level.	Controls both inbound and outbound traffic at the subnet level.
3	You can secure your VPC instances using only security groups.	Network ACLs are an additional layer of defense.
4	Supports allow rules only.	Supports allow rules and deny rules.
5	Stateful (Return traffic is automatically allowed, regardless of any rules).	Stateless (Return traffic must be explicitly allowed by rules).
6	Has separate rules for inbound and outbound traffic.	Has separate rules for inbound and outbound traffic.
7	A newly created security group denies all inbound traffic by default.	A default NACL allows inbound and outbound traffic by default.
8	A newly created security group has an outbound rule that allows all outbound traffic by default	A custom NACLs deny inbound and outbound traffic by default unless you explicitly add rules to allow traffic.
9	Instances associated with a security group can’t talk to each other unless you add rules allowing it.	Each subnet in your VPC must be associated with a network ACL. If none is associated, the default nACL is selected.
10	Security groups are associated with network interfaces.	You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time.

2.Apache vs Ngnix Web Server

S.NO	Apache Web Server	Ngnix Web Server
1	Apache is an open-source web server.	Nginx is a web server. It is also used as a reverse proxy server which receives the request from client and send the request to proxy server.
2	It is mostly used for Unix, Linux, Windows and Solaris platforms.	It is mostly used for Unix like systems, and does not completely support Windows.
3	It was developed by Apache group and initially released on 25 March 1999.	It was developed by Nginx.inc and initially released on 4 october 2004.
4	Apache is written in C and XML.	Nginx is written in C language.
5	It is designed for web server.	It is designed for proxy server as well as web server.
6	In heavy web traffic, it cannot support multiple requests.	It can support multiple client requests with limited hardware resources.
7	In Apache, modules are dynamically fixed that make it more complex.	In Nginx, modules cannot be loaded dynamically as there is a core software in which they are complied .
8	It follows Multi-Threaded approach to process client requests.	It follows Event-Driven approach to process client requests.
9	In Apache, there is a dynamic content in web server itself.	It does not support provide dynamic content.
10	Apache's performance for static content is lower than that of Nginx.	Nginx's performance of static content is two times faster than that of Apache as it can simultaneously run thousands of connections and it uses less memory comparatively.
11	Less security provided as compared to Nginx and also the codebase is very high.	It provides better security with a smaller codebase.
12	File system location are passed to interpret the client requests.	It passes Uniform Resource Identifier (URI) to interpret the client requests.
13	Complex configuration system as compared to Nginx.	It has relatively simpler configuration system.

3.ALB vs NLB

Features	ALB(Application Load Balancer)	NLB(Network Load Balancer)
OSI layer	Works on layer 7, the application layer.	Works on layer 4, the transport layer
Target types	Works with IP, instance, and lambda target types.	Works with IP, instance, and ALB target types.
Protocols	Ends connection.	Ends connection.
Algorithms	Round-robin.	Flow-Hash
Routing Capabilities	Advanced! Supports routing based on URL path or domain.	Straightforward, no path or host-based routing.
Performance	Flexible and optimized for volatile web traffic.	Built for high throughput and low latency.
Path & host based Routing	Supported	Not Supported
Healthchecks	Target group and per target	Pre-Target basis
Integration with AWS services	AWS WAF, ECS,Lambda etc..	Limited Integration
TLS Termination	Yes	NO
Session Stickiness	Supported using Cookies	Not Supported
Use cases	Web applications, APIs	Low latency, High throughput apps
Scalibility	Excellent	Excellent
Protocol Supports	Http,Https, Websocket	TCP,TLS, UDP

4.Route53 and Routing Policy

❖ Route53

- Amazon Route 53 is a highly available and scalable cloud DNS web service.

- It is designed for developers and corporations to route the end users to Internet applications by translating human-readable names like www.geeksforgeeks.org into the numeric IP Address like 192.0.1.1 that computers use to connect.
- If a web application requires a domain name, Route53 service helps to register the name for the website (i.e domain name).
- Whenever a user enters the domain name, Route53 helps to connect the user to the website.
- If any failure is detected at any level, it automatically routes the user to a healthy resource.
- Amazon Route 53 is cost effective, secure and scalable.
- Amazon Route 53 is flexible, highly available and reliable.

❖ Routing Policy

There are different type of Routing Policy

1.Simple routing policy – Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website. You can use simple routing to create records in a private hosted zone.

2.Failover routing policy – Use when you want to configure active-passive failover. You can use failover routing to create records in a private hosted zone.

3.Geolocation routing policy – Use when you want to route traffic based on the location of your users. You can use geolocation routing to create records in a private hosted zone.

4.Latency routing policy – Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency. You can use latency routing to create records in a private hosted zone.

5.Weighted routing policy – Use to route traffic to multiple resources in proportions that you specify. You can use weighted routing to create records in a private hosted zone.