K21Academy
Learn Cloud from Experts

amazon web services

Load Balancing, Auto Scaling & Route 53

[Edition 02]

[Last Update 210509]

Contents

1 Documentation Link 3

2 Elastic Load Balancing 4
2.1 ELB Security Groups 8
2.2 ELB Monitoring 11
2.3 Limits 11
2.4 Classic Load Balancer (CLB) 11
2.5 Security 14
2.6 Application Load Balancer (ALB) 15
2.7 Monitoring 17
2.8 Listeners and Rules 20
2.9 ALB and ECS 23
2.10 Network Load Balancer (NLB) 24
2.11 Sample Questions 26

3 AWS Auto Scaling 27
3.1 Amazon EC2 Auto Scaling 27
3.2 Scaling 29
3.3 Scaling Based on Amazon SQS 31
3.4 ASG Behavior & Configuration 31
3.5 Monitoring 35
3.6 Limits 35
3.7 Sample Questions 36

4 AMAZON ROUTE 53 38
4.1 Hosted Zones 39
4.2 Records 40
4.3 Routing Policies 42
4.3.1 Simple 42
4.3.2 Failover 43
4.3.3 Geo-location 43
4.3.4 Geo-proximity routing policy 44
4.3.5 Latency based routing 44
4.3.6 Multi-value answer routing policy 45
4.3.7 Weeighted 45
4.4 Traffic Flow 46
4.5 Route 53 Resolver 46
4.6 Charges 48
4.7 Sample Questions 49

5 AWS GLOBAL ACCELERATOR 50
5.1 Sample Questions 52

1 DOCUMENTATION LINK

1. What is Classis Load Balancer
   https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html
2. Elastic Load Balancing
   https://aws.amazon.com/elasticloadbalancing/
3. What is Classic Load Balancer?
   https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/introduction.html
4. Elastic Load Balancing FAQs
   https://aws.amazon.com/elasticloadbalancing/faqs/?nc=sn&loc=6
5. Elastic Load Balancing Features
   https://aws.amazon.com/elasticloadbalancing/features/
6. What is a Network Load Balancer?
   https://docs.aws.amazon.com/elasticloadbalancing/latest/network/introduction.html
7. Create a Network Load Balancer
   https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-networkload-balancer.html
8. What are Amazon EC2 instances?
   https://aws.amazon.com/ec2/
9. AWS Route53
   https://aws.amazon.com/route53/
10. AWS Route53 FAQ's
    https://aws.amazon.com/route53/faqs/
11. AWS Route53 Features
    https://aws.amazon.com/route53/features/
12. Introduction to Launch Template
    https://docs.aws.amazon.com/autoscaling/ec2/userguide/LaunchTemplates.html
13. Auto Scaling Groups
    https://docs.aws.amazon.com/autoscaling/ec2/userguide/AutoScalingGroup.html
14. Create Launch Template for Auto Scaling Group
    https://docs.aws.amazon.com/autoscaling/ec2/userguide/create-launchtemplate.html

2 ELASTIC LOAD BALANCING

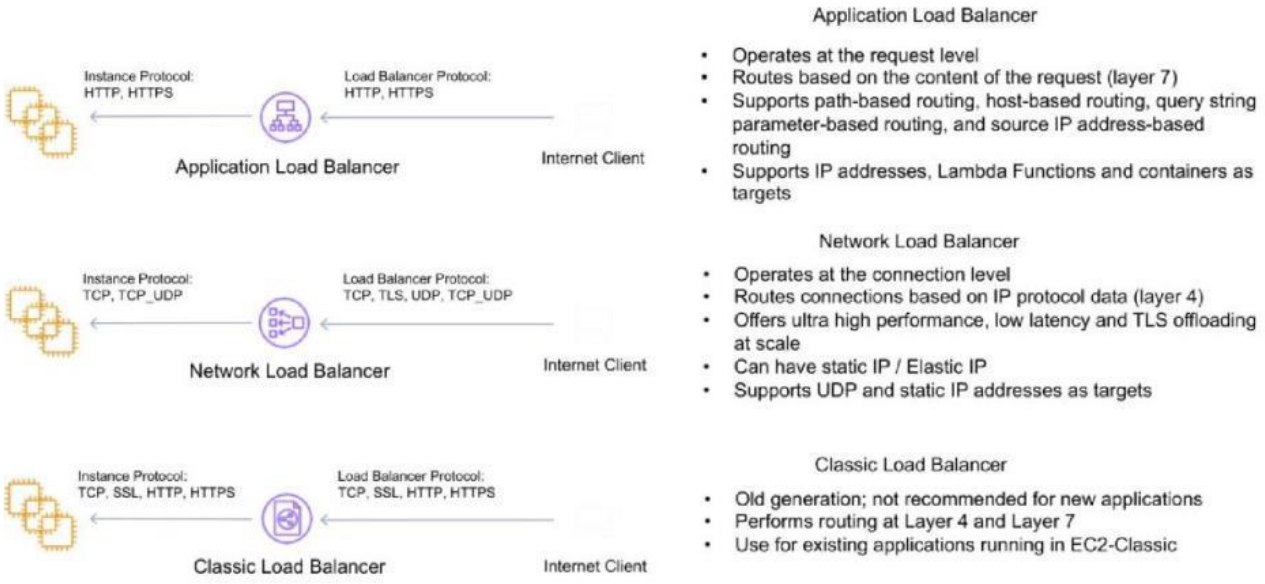
Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses.

There are three types of Elastic Load Balancer (ELB) on AWS:

1. Classic Load Balancer (CLB) – this is the oldest of the three and provides basic load balancing at both layer 4 and layer 7.
2. Application Load Balancer (ALB) – layer 7 load balancer that routes connections based on the content of the request.
3. Network Load Balancer (NLB) – layer 4 load balancer that routes connections based on IP protocol data.

Note: The Classic Load Balancer may be phased out over time and Amazon are promoting the ALB and NLB for most use cases within VPC.

The following image provides an overview of some of the key differences between the three types of ELB:



The following table provides a more detailed feature comparison:

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Protocols	HTTP, HTTPS	TCP	TCP, SSL, HTTP, HTTPS
Platforms	VPC	VPC	EC2-Classic, VPC
Health Checks	✓	✓	✓
CloudWatch Metrics	✓	✓	✓
Logging	✓	✓	✓
Zonal fail-over	✓	✓	✓
Connection draining	✓	✓	✓
Load balancing to multiple ports on an instance	✓	✓	
WebSockets	✓	✓	
IP addresses as targets	✓	✓	
Lambda functions as targets	✓		
Load balancer deletion protection	✓	✓	
Path-based routing	✓		
Host-based routing	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		
Source IP address CIDR-based routing	✓		
Native HTTP/2	✓		
Configurable idle connection timeout	✓		✓

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Cross-zone load balancing	✓	✓	✓
SSL offloading	✓	✓	✓
Server Name Indication (SNI)	✓		
Sticky sessions	✓		✓
Back-end server encryption	✓	✓	✓
Static IP		✓	
Elastic IP address		✓	
Preserve source IP address		✓	
Resource-based IAM permissions	✓	✓	✓
Tag-based IAM permissions	✓	✓	
Slow start	✓		
User authentication	✓		
Redirects	✓		
Fixed response	✓		
Custom security policies			✓

Elastic Load Balancing provides fault tolerance for applications by automatically balancing traffic across targets – Amazon EC2 instances, containers and IP addresses – and Availability Zones while ensuring only healthy targets receive traffic.

An ELB can distribute incoming traffic across your Amazon EC2 instances in a single Availability Zone or multiple Availability Zones.

Only 1 subnet per AZ can be enabled for each ELB.

Route 53 can be used for region load balancing with ELB instances configured in each region.

ELBs can be Internet facing or internal-only.

Internet facing ELB:

- ELB nodes have public IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- Need one public subnet in each AZ where the ELB is defined.
- ELB DNS name format: <name>-<id-number>.<region>.elb.amazonaws.com.

Internal only ELB:

- ELB nodes have private IPs.
- Routes traffic to the private IP addresses of the EC2 instances.
- ELB DNS name format: internal-<name>-<id-number>.<region>.elb.amazonaws.com.
- Internal-only load balancers do not need an Internet gateway.
- EC2 instances and containers can be registered against an ELB.
- ELB nodes use IP addresses within your subnets, ensure at least a /27 subnet and make sure there are at least 8 IP addresses available in order for the ELB to scale.
- An ELB forwards traffic to eth0 (primary IP address).
- An ELB listener is the process that checks for connection requests:
- Listeners for CLB provide options for TCP and HTTP/HTTPS.
- Listeners for ALB only provide options for HTTP and HTTPS.
- Listeners for NLB only provide TCP as an option.
- Deleting an ELB does not affect the instances registered against it (they won't be deleted; they just won't receive any more requests).
- For ALB at least 2 subnets must be specified.
- For NLB only one subnet must be specified (recommended to add at least 2).
- For CLB you don't need to specify any subnets unless you have "Enable advanced VPC configuration" enabled in which case you must specify two.
- ELB uses a DNS record TTL of 60 seconds to ensure new ELB node IP addresses are used to service clients.
- By default, the ELB has an idle connection timeout of 60 seconds, set the idle timeout for applications to at least 60 seconds.
- Perfect Forward Secrecy (PFS) provides additional safeguards against the eavesdropping of encrypted data, through the use of a unique random session key.
- Server Order Preference lets you configure the load balancer to enforce cipher ordering, providing more control over the level of security used by clients to connect with your load balancer.

- ELB does not support client certificate authentication (API Gateway does support this).

2.1 ELB Security Groups

Security groups control the ports and protocols that can reach the front-end listener.

In non-default VPCs you can choose which security group to assign.

You must assign a security group for the ports and protocols on the front-end listener.

You need to also allow the ports and protocols for the health check ports and back-end listeners.

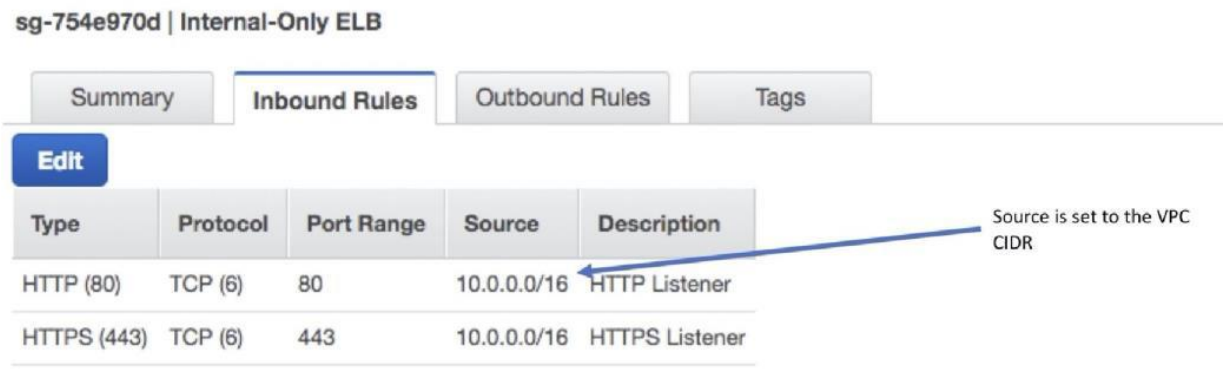
Security group configuration for ELB:

Inbound to ELB (allow)

- Internet-facing ELB:
- Source: 0.0.0.0/0.
- Protocol: TCP.
- Port: ELB listener ports.

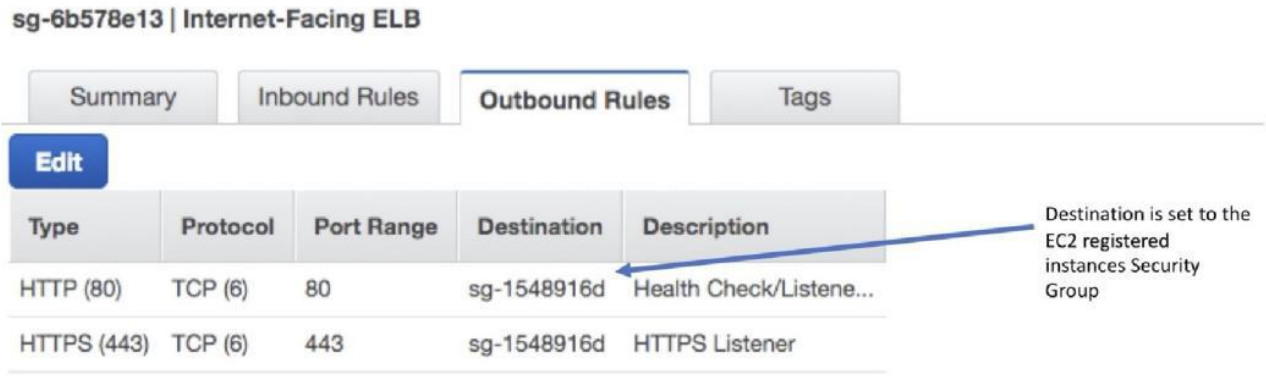
Internal-only ELB:

- Source: VPC CIDR.
- Protocol: TCP.
- Port: ELB Listener ports.



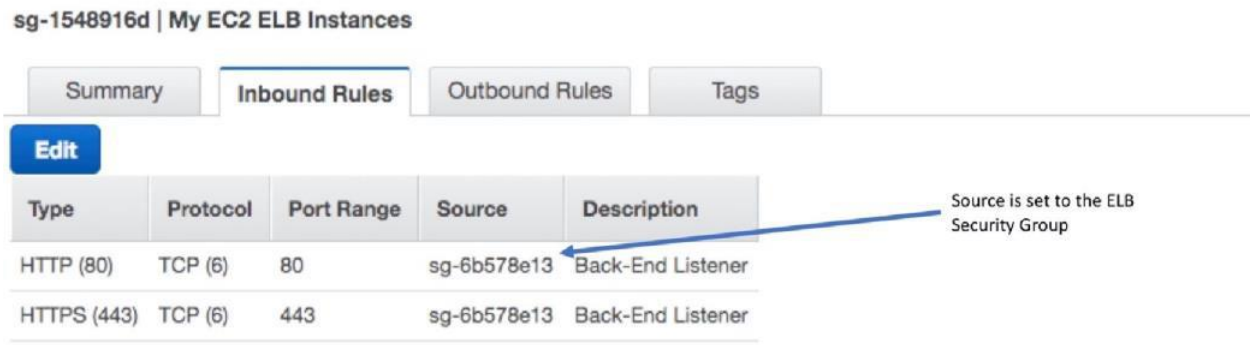
Outbound (allow, either type of ELB):

- Destination: EC2 registered instances security group.
- Protocol: TCP.
- Port: Health Check/Listener.



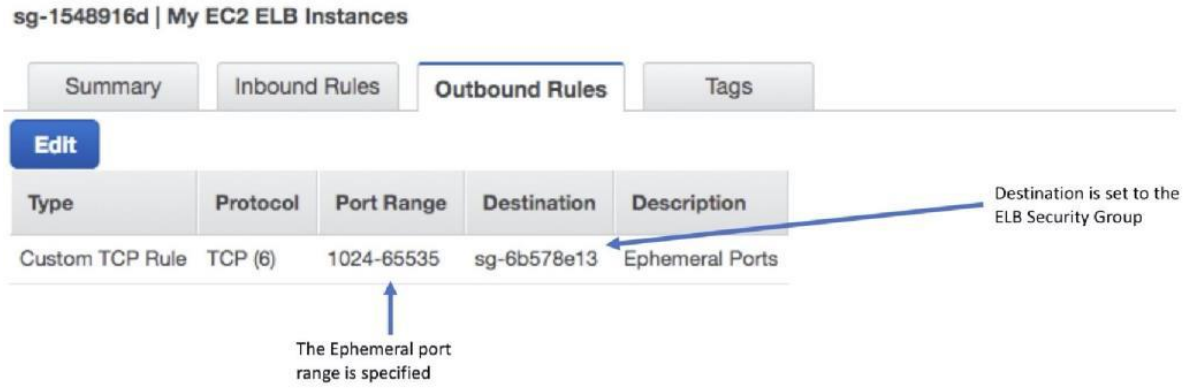
Security group configuration for registered instances:

- Inbound to registered instances (Allow, either type of ELB).
- Source: ELB Security Group.
- Protocol: TCP.
- Port: Health Check/Listener.



Outbound (Allow, for both types of ELB).

- Destination: ELB Security Group.
- Protocol: TCP.
- Port: Ephemeral.



It is also important to ensure NACL settings are set correctly.

Distributed Denial of Service (DDoS) protection:

- ELB automatically distributes incoming application traffic across multiple targets, such as Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, and IP addresses, and multiple Availability Zones, which minimizes the risk of overloading a single resource.
- ELB, like CloudFront, only supports valid TCP requests, so DDoS attacks such as UDP and SYN floods are not able to reach EC2 instances.
- ELB also offers a single point of management and can serve as a line of defence between the internet and your backend, private EC2 instances.

2.2 ELB Monitoring

Monitoring takes place using:

1. **CloudWatch – Every 1 minute**
   - ELB service only sends information when requests are active.
   - Can be used to trigger SNS notifications.
2. **Access Logs o Disabled by default.**
   - Includes information about the clients (not included in CloudWatch metrics).
   - Can identify requester, IP, request type etc.
   - Can be optionally stored and retained in S3.
3. **CloudTrail**
   - Can be used to capture API calls to the ELB.
   - Can be stored in an S3 bucket.

2.3 Limits

The following table details the default limits for your account on a per-region basis:

Name	Default Limit
Application Load Balancers	20
Network Load Balancers	20
Target Groups	3000
Classic Load Balancers	20

2.4 Classic Load Balancer (CLB)

- The Classic Load Balancer provides basic load balancing across multiple Amazon EC2 instances and operates at both the request level and connection level.
- Operates at layer 4 and layer 7.
- Supported protocols: TCP, SSL, HTTP, HTTPS.
- CLB does not support HTTP/2.

Load balancers can listen on the following ports:

- [EC2-VPC] 1-65535.
- [EC2-Classic] 25, 80, 443, 465, 587, 1024-65535.
- CLB's do not have pre-defined IPv4 addresses but are resolved using a DNS name.
- Does not support Elastic IPs.
- Supports IPv4 and IPv6.
- Within a VPC only IPv4 is supported.
- Provides SSL termination and processing.

Sticky Sessions:

- Cookie-based sticky sessions are supported.
- Session stickiness uses cookies and ensures a client is bound to an individual back-end instance for the duration of the cookie lifetime.
- Cookies can be inserted by the application or by the load balancer when configured.
- After cookies expire new requests will be routed by the load balancer normally and a new cookie will be inserted and bind subsequent sessions to the same back-end instance.
- With application-inserted cookies if the back-end instance becomes unhealthy, new requests will be routed by the load balancer normally and a new cookie will be inserted and bind subsequent sessions to the same back-end instance.
- With CLB-inserted cookies if the back-end instance becomes unhealthy, new requests will be routed by the load balancer normally BUT the session will no longer be sticky.
- Must have multiple CLBs for multiple SSL certs.
- Integrates with Auto Scaling, CloudWatch, CloudTrail and Route 53.
- Instances monitored by CLB are reported as InService or OutofService.
- Supports domain zone apex records, e.g., example.com.
- Wildcard certificates are supported.

Health checks:

- Can be configured for HTTP, TCP, HTTPS, SSL.
- Ping port specifies the port for the health check.
- Ping path specifies the path to check, e.g., /index.html.

- Can define timeout, interval, unhealthy threshold, healthy threshold.
- For fault tolerance it is recommended to distribute registered instances across multiple AZs (ideally evenly).

Cross-zone load balancing:

- Cross-zone load balancing is enabled by default for CLB and ALB but not for NLB (when created through the console).
- Cross-zone load balancing is NOT enabled by default if the CLB is created from the CLI or API.
- You can enable or disable cross-zone load balancing on the CLB and NLB at any time.
- For the ALB, cross-zone load balancing is always on and cannot be disabled.
- When cross-zone load balancing is enabled, each load balancer node distributes traffic across the registered targets in all enabled Availability Zones.
- When cross-zone load balancing is disabled, each load balancer node distributes traffic across the registered targets in its Availability Zone only.
- Connection draining is enabled by default and provides a period of time for existing connections to close cleanly.
- When connection draining is in action a CLB will be in the status “InService: Instance deregistration currently in progress”.
- CLB can take 1 to 7 minutes to detect an increase in load and scale.
- If you’re anticipating a fast increase in load, you can contact AWS and instruct them to pre-warm (provision) additional CLB nodes.

Listeners:

- A CLB listener is the process that checks for connection requests.
- You can configure the protocol/port on which your CLB listener listens.
- Front-end listeners check for traffic from clients to the CLB.
- Back-end listeners are configured with the protocol/port to check for traffic from the CLB to the EC2 instances.
- Front-end and back-end listeners can listen on ports 1-65535.
- Front-end and back-end listeners must be at the same layer (e.g., layer 4 or layer 7).
- There is a 1:1 mapping between front-end and back-end listeners.
- Up to 100 listeners can be configured.

- Supports L4 (TCP, SSL) and L7 (HTTP, HTTPS) listeners.
- With packet interception the source IP/port will be from the ELB.
- Proxy protocol for TCP/SSL carries the source (client) IP/port information.
- The Proxy Protocol header helps you identify the IP address of a client when you have a load balancer that uses TCP for back-end connections.
- Ensure the client doesn't go through a proxy or there will be multiple proxy headers.
- Also need to ensure the EC2 instance's TCP stack can process the extra information.
- X-forwarded-for for HTTP/HTTPS carries the source IP/port information.
- To use an HTTPS listener the CLB must have an X.509 SSL/TLS server certificate – this will allow the CLB to terminate the secure session from the client to the CLB.
- The session between the CLB and the EC2 instance can be re-encrypted.
- You can use a certificate generated by AWS Certificate Manager (ACM) or your own certificate.
- If you don't want interception/offloading you can use TCP listeners with certificates on the EC2 instances (traffic is secured end-to-end).
- Proxy protocol only applies to L4.
- X-forwarded-for only applies to L7.
- To filter by source IP use NACLs for proxy protocol (L4) / X-forwarded-for (L7) headers with the EC2 instance's application performing the filtering.

2.5 Security

- CLB supports a single X.509 certificate.
- Two-way authentication with client certificates is not supported on the CLB – you would need to pass through the session using the proxy protocol and have an application that supports client-side certificates.
- When using end-to-end encryption use TCP not SSL/HTTPS on the CLB (does not support Session Stickiness).
- AWS ACM certificates include an RSA public key – ensure you include a set of ciphers that support RSA in the security policy.
- The latest predefined security policy does not include support for SSLv3.
- When choosing a custom security policy, you can select the ciphers and protocols (only for CLB).

SSL Security Policy includes:

- Protocol Versions (SSL/TLS)
- Supports TLS 1.0, 1.1, 1.2, SSL 3.0
- SSL Ciphers o Encryption algorithms
- SSL can use different ciphers to encrypt data

Server Order Preference

- When enabled the first match in the cipher list with the Client list is used
- If disabled (default) the first match in the client cipher list with the CLB is used

2.6 Application Load Balancer (ALB)

- The Application Load Balancer operates at the request level (layer 7), routing traffic to targets – EC2 instances, containers and IP addresses based on the content of the request.
- You can load balance HTTP/HTTPS applications and use layer 7-specific features, such as X-Forwarded-For headers.
- Supports HTTPS termination between the clients and the load balancer.
- Supports management of SSL certificates through AWS IAM and AWS Certificate Manager for pre-defined security policies.
- Server Name Indication (SNI) supports multiple secure websites using a single secure listener With Server Name Indication a client indicates the hostname to connect to.
- IP addresses as targets allows load balancing any application hosted in AWS or on-premises using IP addresses of the application back-ends as targets Need at least 2 availability zones and you can distribute incoming traffic across your targets in multiple Availability Zones.
- Automatically scales its request handling capacity in response to incoming application traffic.
- Can configure an Application Load Balancer to be Internet facing or create a load balancer without public IP addresses to serve as an internal (non-Internet-facing) load balancer.
- Native IPv6 support.
- Internal only ALB only supports IPv4.

Content-Based Routing allows the routing of requests to a service based on the content of the request:

- Host-based routing – route client requests based on the Host field of the HTTP header allowing you to route to multiple domains from the same load balancer.
- Path-based routing – route a client request based on the URL path of the HTTP header (e.g., /images or /orders).
- Provides support for micro-services and containers with load balancing across multiple ports on a single EC2 instance.
- Better performance for real-time streaming.
- Deletion protection can be enabled.
- Request tracing (allows you to track a request by its unique ID). © 2021 Digital Cloud Training 50
- Better health checks and CloudWatch metrics.
- Integration with Amazon Cognito for user authentication.
- Uses a round-robin load balancing algorithm.
- Slow start mode allows targets to “warm up” with a ramp-up period.

Health Checks:

- Can have custom response codes in health checks (200-399).
- There are more details provided in the API and management console for health check failures.
- Reason codes are returned with failed health checks.
- Health checks do not support WebSocket’s.
- Fail open means if no AZ contains a healthy target, the load balancer nodes route requests to all targets.
- Detailed access log information is provided and saved to an S3 bucket every 5 or 6 minutes.
- ALB does not support back-end server authentication (CLB does).
- ALB does not support EC2-Classic (CLB does).
- Deletion protection is possible.
- Deregistration delay is similar to connection draining.

Sticky Sessions:

- Session stickiness uses cookies and ensures a client is bound to an individual back-end instance for the duration of the cookie lifetime.
- ALB supports load balancer-generated cookies only.
- The name of the cookie is AWSALB.
- The contents of these cookies are encrypted using a rotating key.
- You cannot decrypt or modify load balancer-generated cookies.
- Sticky sessions are enabled at the target group level.
- You can also set the duration for the stickiness of the load balancer-generated cookie, in seconds.
- WebSocket's connections are inherently sticky (following the upgrade process).

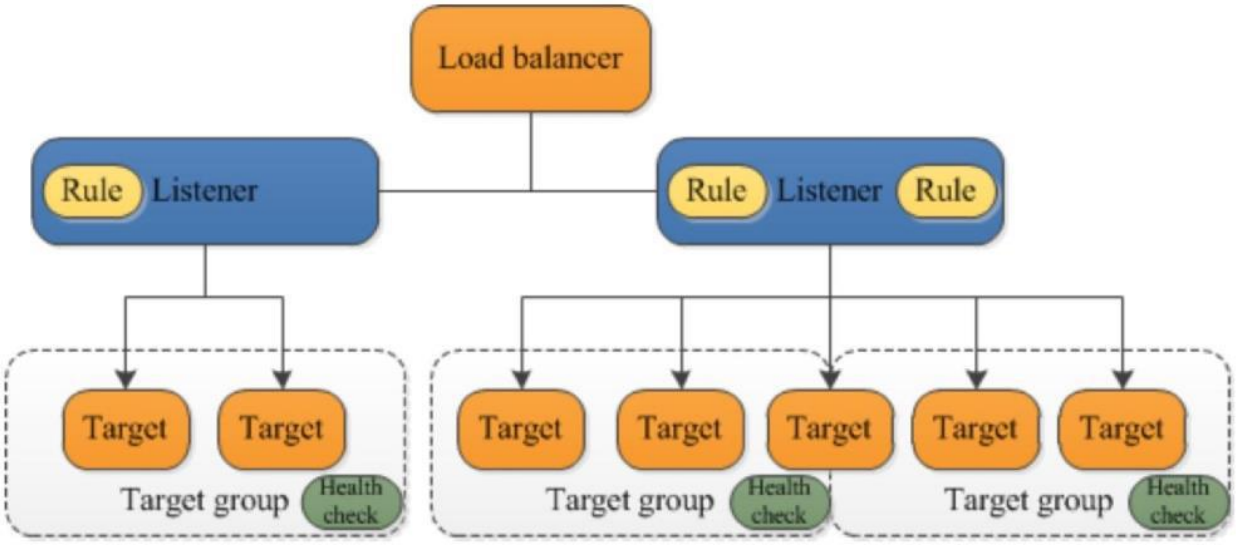
2.7 Monitoring

- CloudTrail can be used to capture API calls. Only pay for the S3 storage charges.
- CloudTrail records information on API calls only.
- To monitor other actions such as time the request was received, the client's IP address, request paths etc. use access logs.
- Access logging is optional and disabled by default.
- You are only charged for the S3 storage.
- ALB logs requests sent to the load balancer including requests that never made it to targets.
- ALB does not log health check requests.

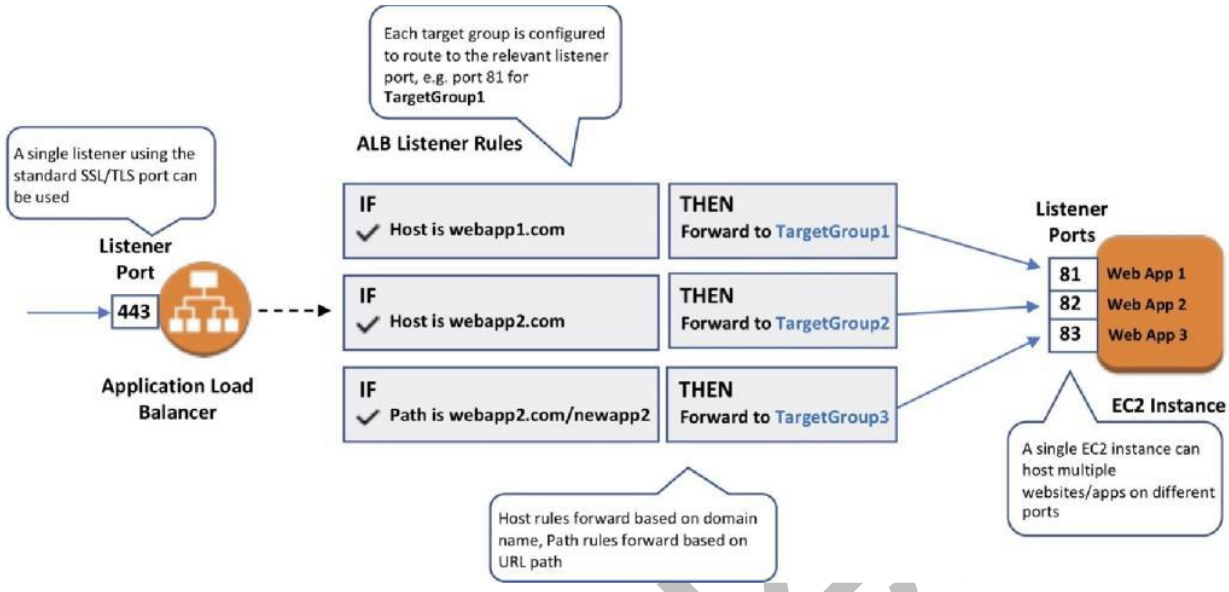
Logging of requests is best effort so shouldn't be relied on for auditing, Target groups

- Target groups are a logical grouping of targets (EC2 instances or ECS).
- Targets are the endpoints and can be EC2 instances, ECS containers, or IP addresses.
- Target groups can exist independently from the ALB.
- Target groups can have up to 1000 targets.
- A single target can be in multiple target groups.
- Only one protocol and one port can be defined per target group.

- The target type in a target group can be an EC2 instance ID, IP address (must be a valid private IP from an existing subnet) or AWS Lambda Function (ALB only).
- You cannot use public IP addresses as targets.
- You cannot use instance IDs and IP address targets within the same target group.
- A target group can only be associated with one load balancer.
- The following diagram illustrates the basic components. Notice that each listener contains a default rule, and one listener contains another rule that routes requests to a different target group. One target is registered with two target groups.



- Target groups are used for registering instances against an ALB or NLB.
- Target groups are a regional construct.
- The following diagram shows how target groups can be used with host-based and target-based routing to route traffic to multiple websites, running on multiple ports, on a single EC2 instance:



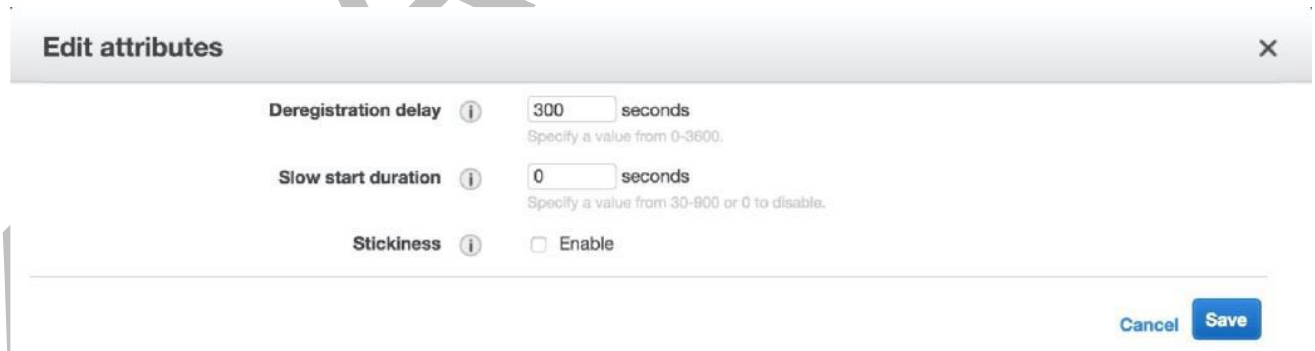
The following attributes can be defined:

Deregistration delay – the amount of time for Elastic Load Balancing to wait before deregistering a target.

Slow start duration – the time period, in seconds, during which the load balancer sends a newly registered target a linearly increasing share of the traffic to the target group.

Stickiness – indicates whether sticky sessions are enabled.

The default settings for attributes are shown below:



- Auto Scaling groups can scale each target group individually.
- You can only use Auto Scaling with the load balancer if using instance IDs in your target group.
- Health checks are defined per target group.
- ALB can route to multiple target groups.

- You can register the same EC2 instance or IP address with the same target group multiple times using different ports (used for routing requests to micro-services).
- If you register by instance ID the traffic is routed using the primary private IP address of the primary network interface.
- If you register by IP address you can route traffic to an instance using any private address from one or more network interfaces.
- You cannot mix different types within a target group (EC2, ECS, IP).
- An EC2 instance can be registered with the same target group multiple times using multiple ports.

IP addresses can be used to register:

- Instances in a peered VPC.
- AWS resources that are addressable by IP address and port.
- On-premises resources linked to AWS through Direct Connect or a VPN connection.

2.8 Listeners and Rules

Listeners:

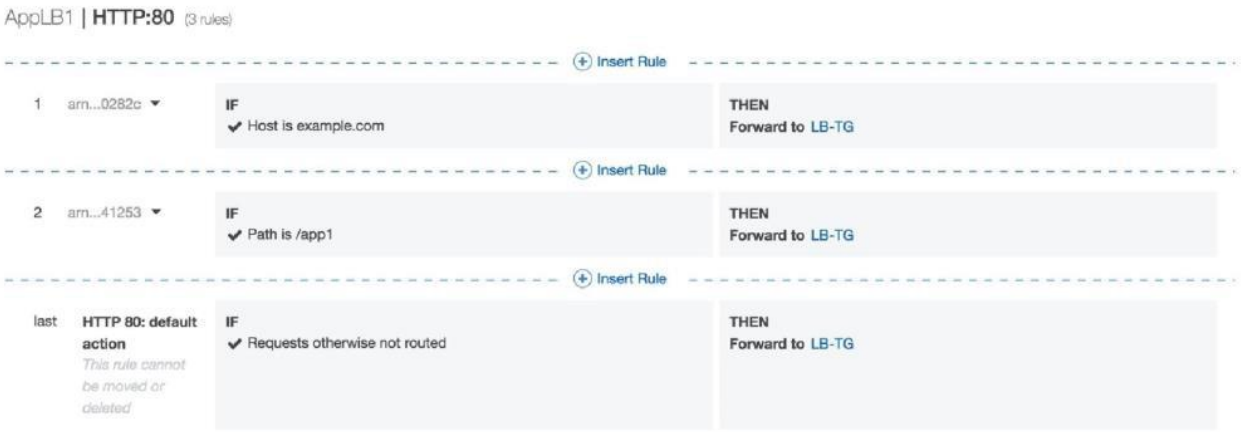
- Each ALB needs at least one listener and can have up to 10.
- Listeners define the port and protocol to listen on.
- Can add one or more listeners.
- Cannot have the same port in multiple listeners.

Listener rules:

- Rules determine how the load balancer routes requests to the targets in one or more target groups.
- Each rule consists of a priority, one or more actions, an optional host condition, and an optional path condition.
- Only one action can be configured per rule.
- One or more rules are required.
- Each listener has a default rule and you can optionally define additional rules.
- Up to 100 rules per ALB.
- Rules determine what action is taken when the rule matches the client request.
- Rules are defined on listeners.

- You can add rules that specify different target groups based on the content of the request (content-based routing).
- If no rules are found the default rule will be followed which directs traffic to the default target groups.

The image below shows a ruleset with a host-based and path-based entry and a default rule at the end:



Default rules:

- When you create a listener, you define an action for the default rule.
- Default rules cannot have conditions.
- You can delete the non-default rules for a listener at any time.
- You cannot delete the default rule for a listener.
- When you delete a listener all of its rules are deleted.
- If no conditions for any of a listener's rules are met, the action for the default rule is taken.

Rule priority:

- Each rule has a priority and they are evaluated in order of lowest to highest.
- The default rule is evaluated last.
- You can change the value of a non-default rule at any time.
- You cannot change the value of the default rule.

Rule action:

- Only one target group per action.
- Each rule has a type and a target group.

- The only supported action type is forward, which forwards requests to the target group.
- You can change the target group for a rule at any time.

Rule conditions:

- There are two types of rule condition: host and path.
- When the conditions for a rule are met the action is taken.
- Each rule can have up to 2 conditions, 1 path condition and 1 host condition.
- Optional condition is the path pattern you want the ALB to evaluate in order for it to route requests.

Request routing:

- After the load balancer receives a request, it evaluates the listener rules in priority order to determine which rule to apply, and then selects a target from the target group for the rule action using the round robin routing algorithm.
- Routing is performed independently for each target group even when a target is registered with multiple target groups.
- You can configure listener rules to route requests to different target groups based on the content of the application traffic.

Content-based routing:

- ALB can route requests based on the content of the request in the host field: host-based or path-based.
- Host-based is domain name-based routing e.g., example.com or app1.example.com.
- The host field contains the domain name and optionally the port number.
- Path-based is URL based routing e.g., example.com/images, example.com/app1.
- You can also create rules that combine host-based and path-based routing.
- Anything that doesn't match content routing rules will be sent to a default target group.

2.9 ALB and ECS

- ECS service maintains the “desired count” of instances.
- Optionally a load balancer can distribute traffic across tasks.
- All containers in a single task definition are placed on a single EC2 container instance.
- You can put multiple containers in the same task definition behind a CLB.
- Define multiple host ports in the service definition.
- Define these listener ports as listeners on the CLB.
- ECS service can only use a single load balancer.
- If your task definition requires multiple ports per container, you must use a CLB with multiple listeners.
- ALB cannot do multiple listeners on a single task definition.
- AWS does not recommend connecting multiple services to the same CLB.
- ALB allows containers to use dynamic host port mapping so that multiple tasks from the same service are allowed on the same container host.
- ALB supports path-based routing and priority rules.
- ALB integrates with EC2 container service using service load balancing.
- If a service uses multiple ports, then multiple task definitions will need to be created with multiple target groups.

Federated authentication:

- ALB now supports authentication from OIDC compliant identity providers such as Google, Facebook and Amazon.
- Implemented through an authentication action on a listener rule that integrates with Amazon Cognito to create user pools.
- AWS SAM can also be used with Amazon Cognito.

2.10 Network Load Balancer (NLB)

Network Load Balancer operates at the connection level (Layer 4), routing connections to targets – Amazon EC2 instances, containers and IP addresses based on IP protocol data.

It is architected to handle millions of requests/sec, sudden volatile traffic patterns and provides extremely low latencies.

Network Load Balancer supports features including:

- WebSocket's
- TLS termination
- Preserves the source IP of the clients
- Provides stable IP support and Zonal isolation
- Long-running connections that are very useful for WebSocket type applications
- High throughput – designed to handle traffic as it grows and can load balance millions of requests/seconds.
- Extremely low latencies for latency-sensitive applications.
- Uses static IP addresses – each NLB provides a single IP address for each AZ.
- Can also assign an Elastic IP to the load balancer per AZ.
- The IP-per-AZ feature reduces latency with improved performance, improves availability through isolation and fault tolerance and makes the use of NLBs transparent to your client applications.
- Preserves the source IP of clients and provides stable IP support and Zonal isolation.
- Can load balance any application hosted in AWS or on-premises using IP addresses of the application back-ends as targets.
- NLB supports connections from clients to IP-based targets in peered VPCs across different AWS Regions.
- Supports both network and application target health checks.
- Supports long-running/lived connections (ideal for WebSocket applications).
- Supports failover between IP addresses within and across regions (uses Route 53 health checks).
- Integration with Route 53 enables the removal of a failed load balancer IP address from service and subsequent redirection of traffic to an alternate Network Load Balancer in another region.
- Supports cross-zone load balancing (not enabled by default when created through the console, unlike ALB and CLB).

- Uses the same API as the Application Load Balancer.
- Also uses Target Groups (see section above).

Target groups for Network Load Balancers support the following protocols and ports:

- Protocols: TCP, TLS, UDP, TCP_UDP.
- Ports: 1-65535.

The following table summarizes the supported combinations of listener protocol and target group settings:

Listener Protocol	Target Group Protocol	Target Group Type	Health Check Protocol
TCP	TCP | TCP_UDP	instance | ip	HTTP | HTTPS | TCP
TLS	TCP | TLS	instance | ip	HTTP | HTTPS | TCP
UDP	UDP | TCP_UDP	instance	HTTP | HTTPS | TCP
TCP_UDP	TCP_UDP	instance	HTTP | HTTPS | TCP

- CloudWatch reports Network Load Balancer metrics.
- Enhanced logging – can use the Flow Logs feature to record all requests sent to your load balancer.

2.11 Sample Questions

Q1 : Which type of load balancer operates at the Application layer?

A. Classic load balancer

B. Application load balancer

C. Network load balancer

D. Both classic and application load balancers

Answer : D

Expalaination: Both classic and application load balancers operate at the Application layer. Classic load balancers also operate at layer 4, the Transport layer. Network load balancers operate at the Transport layer, which is layer 4 of the OSI model.

Q2 : Which of the following protocols are supported by an application load balancer? (Choose two.)

A. SSH

B. HTTP

C. HTTPS

D. FTP

Answer: B, C

Explanation : This should be an easy answer: Application load balancers, as well as classic load balancers, only support HTTP and HTTPS.

Q3 : At what OSI layer does a network load balancer operate?

A. 4
B. 7
C. 4 and 7
D. 6

Answer:A

Explanation : Application load balancers operate at the Application layer, which is layer 7 of the OSI model. ELBs (classic load balancers) operate at the Transport layer, layer 4, as well as layer 7, and network load balancers operate at layer 4 as well.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm06

3 AWS AUTO SCALING

3.1 Amazon EC2 Auto Scaling

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost.

AWS Auto Scaling refers to a collection of Auto Scaling capabilities across several AWS services.

The services within the AWS Auto Scaling family include:

Amazon EC2 (known as Amazon EC2 Auto Scaling)

Amazon ECS

Amazon DynamoDB

Amazon Aurora

GENERAL AUTO SCALING CONCEPTS

- Amazon EC2 Auto Scaling helps you ensure that you have the correct number of Amazon EC2 instances available to handle the load for your application.
- You create collections of EC2 instances, called Auto Scaling groups.
- Automatically provides horizontal scaling (scale-out) for your instances.
- Triggered by an event of scaling action to either launch or terminate instances.
- Availability, cost, and system metrics can all factor into scaling.
- Auto Scaling is a region-specific service.
- Auto Scaling can span multiple AZs within the same AWS region.
- Auto Scaling can be configured from the Console, CLI, SDKs and APIs.
- There is no additional cost for Auto Scaling, you just pay for the resources (EC2 instances) provisioned.
- Auto Scaling works with ELB, CloudWatch and CloudTrail.
- You can determine which subnets Auto Scaling will launch new instances into.
- Auto Scaling will try to distribute EC2 instances evenly across AZs.
- Launch configuration is the template used to create new EC2 instances and includes parameters such as instance family, instance type, AMI, key pair and security groups.
- You cannot edit a launch configuration once defined.

A launch configuration:

- Can be created from the AWS console or CLI.
- You can create a new launch configuration, or.
- You can use an existing running EC2 instance to create the launch configuration. o The AMI must exist on EC2.
- EC2 instance tags and any additional block store volumes created after the instance launch will not be taken into account.
- If you want to change your launch configurations you have to create a new one, make the required changes, and use that with your auto scaling groups.
- You can use a launch configuration with multiple Auto Scaling Groups (ASG).
- An ASG is a logical grouping of EC2 instances managed by an Auto Scaling Policy.
- An ASG can be edited once defined.
- You can attach one or more classic ELBs to your existing ASG.
- You can attach one or more Target Groups to your ASG to include instances behind an ALB.
- The ELBs must be in the same region.
- Once you do this any EC2 instance existing or added by the ASG will be automatically registered with the ASG defined ELBs.
- If adding an instance to an ASG would result in exceeding the maximum capacity of the ASG the request will fail.

You can add a running instance to an ASG if the following conditions are met:

- The instance is in a running state.
- The AMI used to launch the instance still exists.
- The instance is not part of another ASG.
- The instance is in the same AZs for the ASG.

3.2 Scaling

The scaling options define the triggers and when instances should be provisioned/de-provisioned.

There are four scaling options:

1. Maintain – keep a specific or minimum number of instances running.
2. Manual – use maximum, minimum, or a specific number of instances.
3. Scheduled – increase or decrease the number of instances based on a schedule.
4. Dynamic – scale based on real-time system metrics (e.g., CloudWatch metrics).

The following table describes the scaling options available and when to use them:

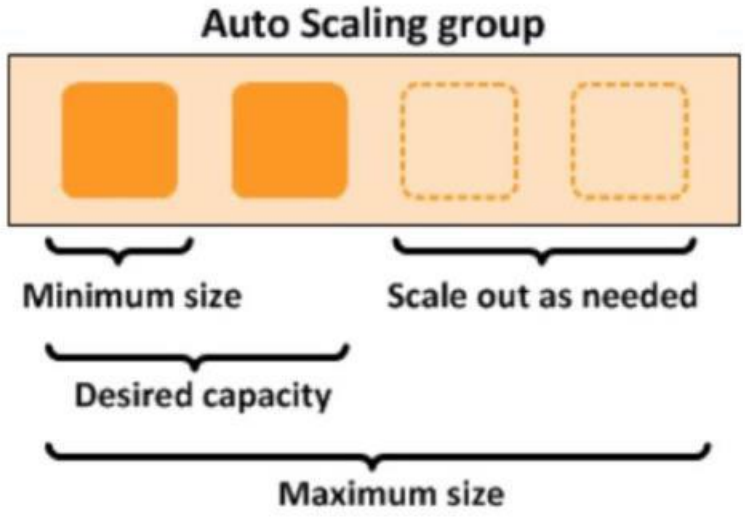
Scaling Type	What it is	When to use
Maintain	Ensures the required number of instances are running	Use when you always need a known number of instances running at all times
Manual	Manually change desired capacity via the console or CLI	Use when your needs change rarely enough that you're OK to make manual changes
Scheduled	Adjust min/max instances on specific dates/times or recurring time periods	Use when you know when your busy and quiet times are. Useful for ensuring enough instances are available *before* very busy times
Dynamic	Scale in response to system load or other triggers using metrics	Useful for changing capacity based on system utilization, e.g. CPU hits 80%

The scaling options are configured through Scaling Policies which determine when, if, and how the ASG scales and shrinks.

The following table describes the scaling policy types available for dynamic scaling policies and when to use them (more detail further down the page):

Scaling	What it is	When to use
Target Tracking Policy	The scaling policy adds or removes capacity as required to keep the metric at, or close to, the specified target value	A use case is that you want to keep the aggregate CPU usage of your ASG at 70%
Simple Scaling Policy	Waits until health check and cool down period expires before re-evaluating	This is a more conservative way to add/remove instances. Useful when load is erratic. AWS recommend step scaling instead of simple in most cases
Step Scaling Policy	Increase or decrease the current capacity of your Auto Scaling group based on a set of scaling adjustments, known as step adjustments	Useful when you want to vary adjustments based on the size of the alarm breach

The diagram below depicts an Auto Scaling group with a Scaling policy set to a minimum size of 1 instance, a desired capacity of 2 instances, and a maximum size of 4 instances:



3.3 Scaling Based on Amazon SQS

- Can also scale based on an Amazon Simple Queue Service (SQS) queue.
- This comes up as an exam question for SAA-C02.
- Uses a custom metric that's sent to Amazon CloudWatch that measures the number of messages in the queue per EC2 instance in the Auto Scaling group.
- Then use a target tracking policy that configures your Auto Scaling group to scale based on the custom metric and a set target value. CloudWatch alarms invoke the scaling policy.
- Use a custom "backlog per instance" metric to track not just the number of messages in the queue but the number available for retrieval.
- Can base off the SQS Metric "ApproximateNumberOfMessages".

3.4 ASG Behavior & Configuration

EC2 Auto Scaling – Termination Policy:

- Termination policies control which instances are terminated first when a scale-in event occurs.
- There is a default termination policy and options for configuring your own customized termination policies.
- The default termination policy is designed to help ensure that your instances span Availability Zones evenly for high availability.
- The default policy is kept generic and flexible to cover a range of scenarios.
- You can define Instance Protection which stops Auto Scaling from scaling in and terminating the instances.
- If Auto Scaling fails to launch instances in an AZ it will try other AZs until successful.
- The default health check grace period is 300 seconds.
- Scale-out is the process in which EC2 instances are launched by the scaling policy.
- Scale-in is the process in which EC2 instances are terminated by the scaling policy.
- It is recommended to create a scale-in event for each scale-out event created.
- Auto Scaling can perform rebalancing when it finds that the number of instances across AZs is not balanced.
- Auto Scaling rebalances by launching new EC2 instances in the AZs that have fewer instances first, only then will it start terminating instances in AZs that had more instances.

- Auto Scaling may go over the maximum number of instances by 10% temporarily for the purposes of rebalancing.

An imbalance may occur due to:

- Manually removing AZs/subnets from the configuration.
- Manually terminating EC2 instances.
- EC2 capacity issues.
- Spot price is reached.

Health checks:

- By default, uses EC2 status checks.
- Can also use ELB health checks and custom health checks.
- ELB health checks are in addition to the EC2 status checks.
- If any health check returns an unhealthy status the instance will be terminated.
- With ELB an instance is marked as unhealthy if ELB reports it as OutOfService.
- A healthy instance enters the InService state.
- If an instance is marked as unhealthy it will be scheduled for replacement.
- If connection draining is enabled, Auto Scaling waits for in-flight requests to complete or timeout before terminating instances.
- The health check grace period allows a period of time for a new instance to warm up before performing a health check (300 seconds by default).
- If using an ELB it is best to enable ELB health checks as otherwise EC2 status checks may show an instance as being healthy that the ELB has determined is unhealthy. In this case the instance will be removed from service by the ELB but will not be terminated by Auto Scaling.
- Elastic IPs and EBS volumes are detached from terminated instances and will need to be manually reattached.
- Using custom health checks a CLI command can be issued to set the instance's status to unhealthy, e.g.:
- aws autoscaling set–instance-health –instance-id i-123abc45d –health-status Unhealthy
- Once in a terminating state an EC2 instance cannot be put back into service again.
- However, there is a short time period in which a CLI command can be run to change an instance to healthy.

- Unlike AZ rebalancing, termination of unhealthy instances happens first, then Auto Scaling attempts to launch new instances to replace terminated instances.
- You can manually remove (detach) instances from an ASG using the AWS Console or CLI.
- When detaching an instance, you can optionally decrement the ASG's desired capacity (so it doesn't launch another instance).
- An instance can be attached to one ASG at a time.
- You can suspend and then resume one or more of the scaling processes for your Auto Scaling group.
- Suspending scaling processes can be useful when you want to investigate a configuration problem or other issue with your web application and then make changes to your application, without invoking the scaling processes.
- You can manually move an instance from an ASG and put it in the standby state.
- Instances in standby state are still managed by Auto Scaling, are charged as normal, and do not count towards available EC2 instance for workload/application use.
- Auto scaling does not perform health checks on instances in the standby state.
- Standby state can be used for performing updates/changes/troubleshooting etc. without health checks being performed or replacement instances being launched.
- When you delete an ASG the instances will be terminated.
- You can choose to use Spot instances in launch configurations and specify a bid price.
- Auto Scaling treats spot instances the same as on-demand instances.
- You cannot mix Spot instances with on-demand.
- If you want to change the bid price you need to create a new launch configuration.

Auto Scaling can be configured to send an SNS email when:

- An instance is launched.
- An instance is terminated.
- An instance fails to launch.
- An instance fails to terminate.

Merging ASGs

- You can merge multiple single AZ Auto Scaling Groups into a single multi-AZ ASG.
- Merging can only be performed by using the CLI.
- Process is to rezone one of the groups to cover/span the other AZs for the other ASGs.
- Then delete the other ASGs.
- Can be performed on ASGs with or without ELBs attached to them.
- The resulting ASG must be one of the pre-existing ASGs.

Cooldown Period

- 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.
- The default cooldown period is applied when you create your Auto Scaling group.
- The default value is 300 seconds.
- You can configure the default cooldown period when you create the Auto Scaling group, using the AWS Management Console, the create-auto-scaling-group command (AWS CLI), or the CreateAutoScalingGroup API operation.
- Automatically applies to dynamic scaling and optionally to manual scaling but not supported for scheduled scaling.
- Can override the default cooldown via scaling-specific cooldown.

Scheduled:

- You cannot configure two scheduled activities at the same date/time.
- Scheduled actions can be edited from the AWS Console or CLI.
- Cooldown timer is not supported for scheduled or step on-demand scaling.

Dynamic:

- An alarm is an object that watches over a single metric, e.g., CPU/memory/network utilization.
- You need to have a scale-out and a scale-in policy configured.

Step scaling:

- Configure multiple steps/adjustments.
- Does not support cool down timers.
- Can respond to multiple alarms and initiate multiple scaling activities.
- Supports a warm-up timer which is the time it will take a newly launched instance to be ready.
- The warm-up period is the period of time in which a newly created EC2 instance launched by ASG using step scaling is not considered toward the ASG metrics.

3.5 Monitoring

- Basic monitoring sends EC2 metrics to CloudWatch about ASG instances every 5 minutes.
- Detailed can be enabled and sends metrics every 1 minute (chargeable).
- When the launch configuration is created from the console basic monitoring of EC2 instances is enabled by default.
- When the launch configuration is created from the CLI detailed monitoring of EC2 instances is enabled by default.
- When you enable Auto Scaling group metrics, Auto Scaling sends sampled data to CloudWatch every minute.
- Configure ASG and EC2 monitoring options so they use the same time period, e.g., detailed monitoring (EC2) and 60 seconds (ASG), or basic monitoring (EC2) and 300 seconds (ASG).

3.6 Limits

Name	Default Limit
Auto Scaling Groups	200
Launch Configurations	200

3.7 Sample Questions

Q1 : Which of the following items are included in an Auto Scaling Launch Configuration? (Choose two.)

A. The AMI to use for creating new instances
B. The EBS storage volume for the instances to create
C. The polling time for monitoring network latency
D. The IAM role to associate with created instances

Answer : A, D

Explanation: Launch configurations are concerned primarily with creating new instances while staying abstract from the details of what is on those instances. So the AMI and IAM role for an instance is a general configuration, applies to all created instances, and is correct (A and D). The polling time for latency isn't connected to launching new instances (although it might be a trigger configured elsewhere). Each instance is associated with a different EBS volume, so selecting an EBS volume for multiple instances doesn't actually make sense.

Q2 : Which of the following items are included in an Auto Scaling Launch Configuration? (Choose two.)

A. The AMI to use for creating new instances
B. The EBS storage volume for the instances to create
C. The polling time for monitoring network latency
D. The IAM role to associate with created instances

Answer:A, D

Explanation: Launch configurations are concerned primarily with creating new instances while staying abstract from the details of what is on those instances. So the AMI and IAM role for an instance is a general configuration, applies to all created instances, and is correct (A and D). The polling time for latency isn't connected to launching new instances (although it might be a trigger configured elsewhere). Each instance is associated with a different EBS volume, so selecting an EBS volume for multiple instances doesn't actually make sense.

Q3. What technology enables compute capacity to adjust as loads change?

A. Load balancing

B. Automatic failover

C. Round robin

D. Auto Scaling

Answer: D

Explanation: AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, it's easy to setup application scaling for multiple resources across multiple services in minutes. The service provides a simple, powerful user interface that lets you build scaling plans for resources including Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables and indexes, and Amazon Aurora Replicas. AWS Auto Scaling makes scaling simple with recommendations that allow you to optimize performance, costs, or balance between them. If you're already using Amazon EC2 Auto Scaling to dynamically scale your Amazon EC2 instances, you can now combine it with AWS Auto Scaling to scale additional resources for other AWS services. With AWS Auto Scaling, your applications always have the right resources at the right time.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm06

4 AMAZON ROUTE 53

- Amazon Route 53 is a highly available and scalable Domain Name System (DNS) service.
- Route 53 offers the following functions:
- Domain name registry.
- DNS resolution.
- Health checking of resources.
- Route 53 can perform any combination of these functions.
- Route 53 provides a worldwide distributed DNS service.
- Route 53 is located alongside all edge locations.
- Health checks verify Internet connected resources are reachable, available and functional.
- Route 53 can be used to route Internet traffic for domains registered with another domain registrar (any domain).
- When you register a domain with Route 53 it becomes the authoritative DNS server for that domain and creates a public hosted zone.
- To make Route 53 the authoritative DNS for an existing domain without transferring the domain create a Route 53 public hosted zone and change the DNS Name Servers on the existing provider to the Route 53 Name Servers.
- Changes to Name Servers may not take effect for up to 48 hours due to the DNS record Time To Live (TTL) values.
- You can transfer domains to Route 53 only if the Top Level Domain (TLD) is supported.
- You can transfer a domain from Route 53 to another registrar by contacting AWS support.
- You can transfer a domain to another account in AWS however it does not migrate the hosted zone by default (optional).
- It is possible to have the domain registered in one AWS account and the hosted zone in another AWS account.
- Primarily uses UDP port 53 (can use TCP).
- AWS offer a 100% uptime SLA for Route 53.
- You can control management access to your Amazon Route 53 hosted zone by using IAM.
- There is a default limit of 50 domain names, but this can be increased by contacting support.

- Private DNS is a Route 53 feature that lets you have authoritative DNS within your VPCs without exposing your DNS records (including the name of the resource and its IP address(es) to the Internet.
- You can use the AWS Management Console or API to register new domain names with Route 53.

4.1 Hosted Zones

- A hosted zone is a collection of records for a specified domain.
- A hosted zone is analogous to a traditional DNS zone file; it represents a collection of records that can be managed together.
- There are two types of zones:
  1. Public host zone – determines how traffic is routed on the Internet.
  2. Private hosted zone for VPC – determines how traffic is routed within VPC (resources are not accessible outside the VPC).
- Amazon Route 53 automatically creates the Name Server (NS) and Start of Authority (SOA) records for the hosted zones.
- Amazon Route 53 creates a set of 4 unique name servers (a delegation set) within each hosted zone.
- You can create multiple hosted zones with the same name and different records.
- NS servers are specified by Fully Qualified Domain Name (FQDN) but you can get the IP addresses from the command line (e.g. dig or nslookup).
- For private hosted zones you can see a list of VPCs in each region and must select one.
- **For private hosted zones you must set the following VPC settings to "true":**
  - enableDnsHostname
  - enableDnsSupport
- You also need to create a DHCP options set.
- You can extend an on-premises DNS to VPC.
- You cannot extend Route 53 to on-premises instances.
- You cannot automatically register EC2 instances with private hosted zones (would need to be scripted).
- Health checks check the instance health by connecting to it.

Health checks can be pointed at:

- Endpoints
- Status of other health checks
- Status of a CloudWatch alarm
- Endpoints can be IP addresses or domain names.

4.2 Records

- Amazon Route 53 currently supports the following DNS record types:
  - A (address record)
  - AAAA (IPv6 address record)
  - CNAME (canonical name record)
  - CAA (certification authority authorization)
  - MX (mail exchange record)
  - NAPTR (name authority pointer record
  - NS (name server record)
  - PTR (pointer record)
  - SOA (start of authority record)
  - SPF (sender policy framework)
  - SRV (service locator)
  - TXT (text record)
  - Alias (an Amazon Route 53-specific virtual record)
- The Alias record is a Route 53 specific record type.
- Alias records are used to map resource record sets in your hosted zone to Amazon Elastic Load Balancing load balancers, Amazon CloudFront distributions, AWS Elastic Beanstalk environments, or Amazon S3 buckets that are configured as websites.
- You can use Alias records to map custom domain names (such as api.example.com) both to API Gateway custom regional APIs and edge-optimized APIs and to Amazon VPC interface endpoints.
- The Alias is pointed to the DNS name of the service.
- You cannot set the TTL for Alias records for ELB, S3, or Elastic Beanstalk environment (uses the service's default).
- Alias records work like a CNAME record in that you can map one DNS name (e.g. example.com) to another 'target' DNS name (e.g. elb1234.elb.amazonaws.com).

- An Alias record can be used for resolving apex / naked domain names (e.g. example.com rather than sub.example.com).
- A CNAME record can't be used for resolving apex / naked domain names.

Generally, use an Alias record where possible. The following table details the differences between Alias and CNAME records:

CNAME Records	Alias Records
Route 53 charges for CNAME queries	Route 53 doesn't charge for alias queries to AWS resources
You can't create a CNAME record at the top node of a DNS namespace (zone apex)	You can create an alias record at the zone apex (however you can't route to a CNAME at the zone apex)
A CNAME record redirects queries for a domain name regardless of record type	Route 53 follows the pointer in an alias record only when the record type also matches
A CNAME can point to any DNS record that is hosted anywhere	An alias record can only point to a CloudFront distribution, Elastic Beanstalk environment, ELB, S3 bucket as a static website, or to another record in the same hosted zone that you're creating the alias record in
A CNAME record is visible in the answer section of a reply from a Route 53 DNS server	An alias record is only visible in the Route 53 console or the Route 53 API
A CNAME record is followed by a recursive resolver	An alias record is only followed inside Route 53. This means that both the alias record and its target must exist in Route 53

Route 53 supports wildcard entries for all record types, except NS records.

4.3 Routing Policies

Routing policies determine how Route 53 responds to queries.

The following table highlights the key function of each type of routing policy:

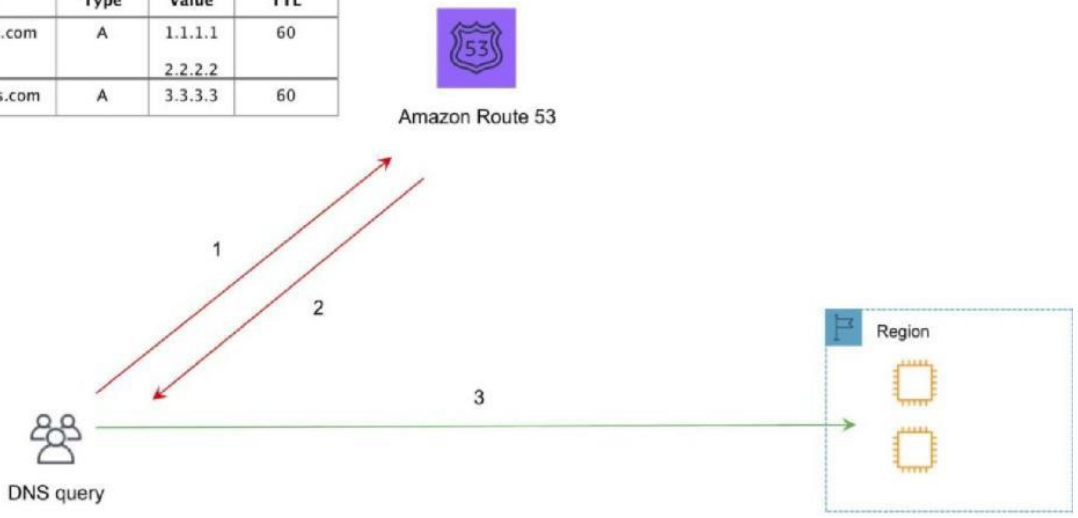
Policy	What it Does
Simple	Simple DNS response providing the IP address associated with a name
Failover	If primary is down (based on health checks), routes to secondary destination
Geolocation	Uses geographic location you're in (e.g. Europe) to route you to the closest region
Geoproximity	Routes you to the closest region within a geographic area
Latency	Directs you based on the lowest latency route to resources
Multivalue answer	Returns several IP addresses and functions as a basic load balancer
Weighted	Uses the relative weights assigned to resources to determine which to route to

4.3.1 SIMPLE

- An A record is associated with one or more IP addresses
- Uses round robin
- Does not support health checks

The following diagram depicts an Amazon Route 53 Simple routing policy configuration:

Name	Type	Value	TTL
simple.dctlabs.com	A	1.1.1.1 2.2.2.2	60
simpler.dctlabs.com	A	3.3.3.3	60



4.3.2 FAILOVER

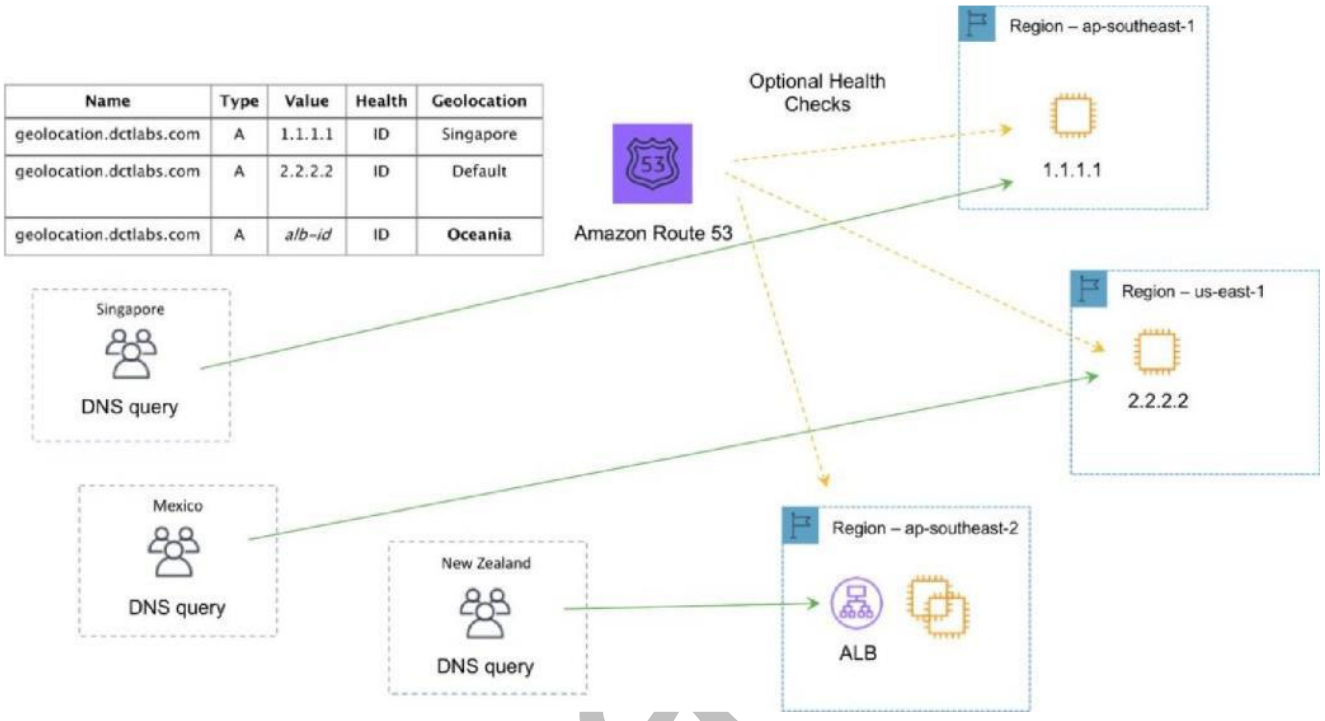
- Failover to a secondary IP address.
- Associated with a health check.
- Used for active-passive.
- Routes only when the resource is healthy.
- Can be used with ELB.
- When used with Alias records set Evaluate Target Health to “Yes” and do not use health checks.

The following diagram depicts an Amazon Route 53 Failover routing policy configuration:

4.3.3 GEO-LOCATION

- Caters to different users in different countries and different languages.
- Contains users within a particular geography and offers them a customized version of the workload based on their specific needs.
- Geolocation can be used for localizing content and presenting some or all of your website in the language of your users.
- Can also protect distribution rights.
- Can be used for spreading load evenly between regions.
- If you have multiple records for overlapping regions, Route 53 will route to the smallest geographic region.
- You can create a default record for IP addresses that do not map to a geographic location.

The following diagram depicts an Amazon Route 53 Geolocation routing policy configuration:



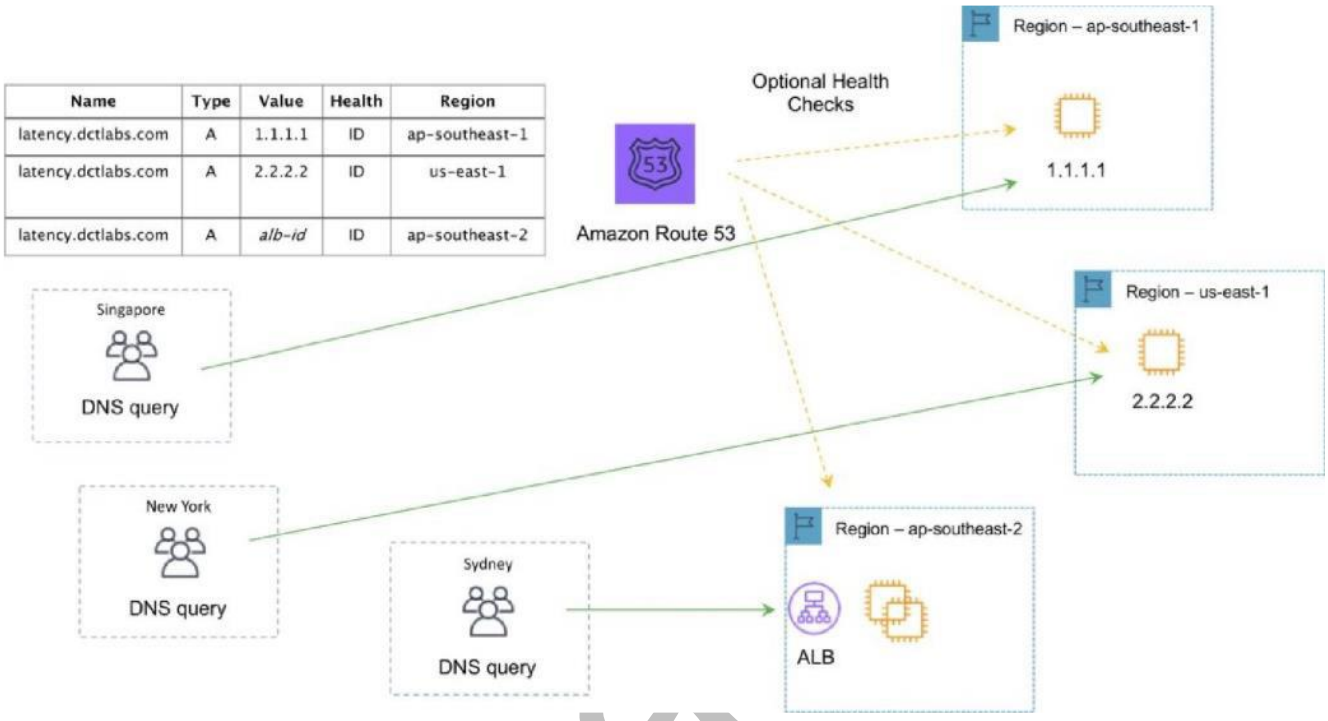
4.3.4 GEO-PROXIMITY ROUTING POLICY

- Use for routing traffic based on the location of resources and, optionally, shift traffic from resources in one location to resources in another.

4.3.5 LATENCY BASED ROUTING

- AWS maintains a database of latency from different parts of the world.
- Focused on improving performance by routing to the region with the lowest latency.
- You create latency records for your resources in multiple EC2 locations.

The following diagram depicts an Amazon Route 53 Latency based routing policy configuration:



4.3.6 MULTI-VALUE ANSWER ROUTING POLICY

- Use for responding to DNS queries with up to eight healthy records selected at random.

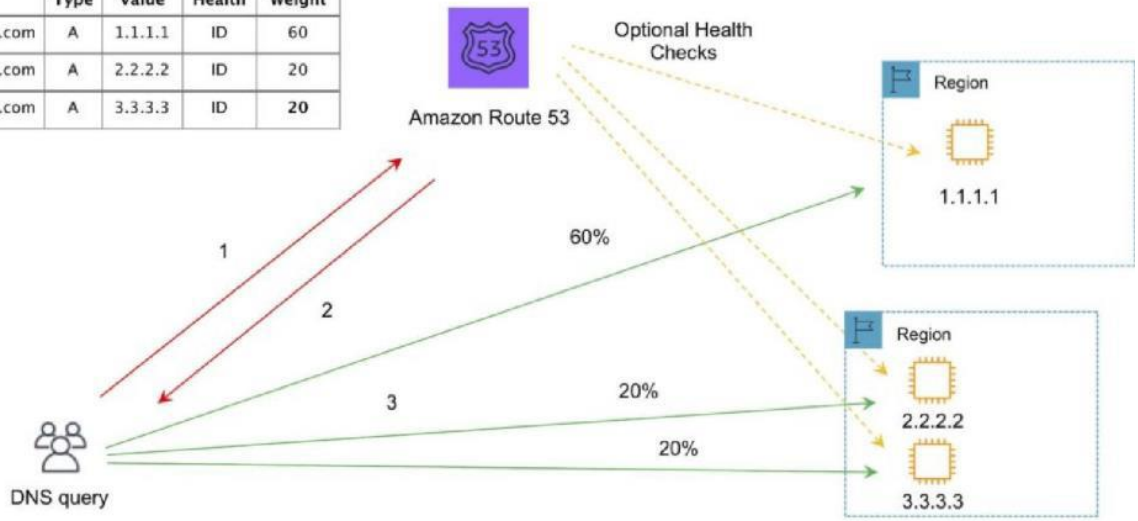
The following diagram depicts an Amazon Route 53 Multivalue routing policy configuration:

4.3.7 WEEIGHTED

- Similar to simple but you can specify a weight per IP address.
- You create records that have the same name and type and assign each record a relative weight.
- Numerical value that favors one IP over another.
- To stop sending traffic to a resource you can change the weight of the record to 0.

The following diagram depicts an Amazon Route 53 Weighted routing policy configuration:

Name	Type	Value	Health	Weight
weighted.dctlabs.com	A	1.1.1.1	ID	60
weighted.dctlabs.com	A	2.2.2.2	ID	20
weighted.dctlabs.com	A	3.3.3.3	ID	20



4.4 Traffic Flow

- Route 53 Traffic Flow provides Global Traffic Management (GTM) services.
- Traffic flow policies allow you to create routing configurations for resources using routing types such as failover and geolocation.
- Create policies that route traffic based on specific constraints, including latency, endpoint health, load, geo-proximity and geography.

Scenarios include:

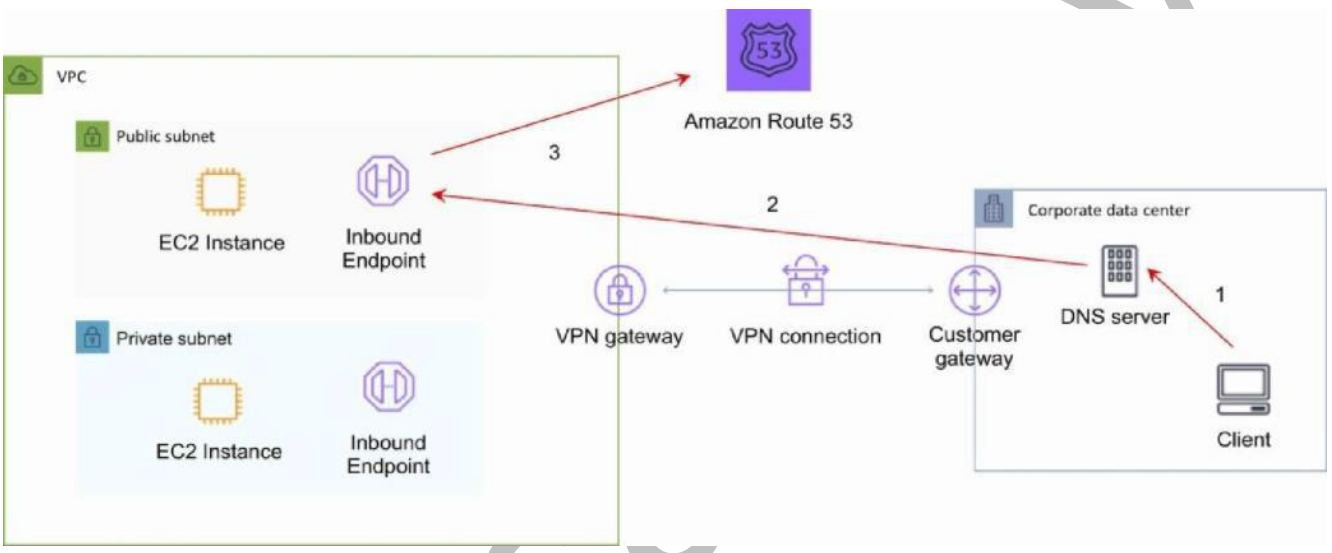
- Adding a simple backup page in Amazon S3 for a website.
- Building sophisticated routing policies that consider an end user's geographic location, proximity to an AWS region, and the health of each of your endpoints.
- Amazon Route 53 Traffic Flow also includes a versioning feature that allows you to maintain a history of changes to your routing policies, and easily roll back to a previous policy version using the console or API.

4.5 Route 53 Resolver

- Route 53 Resolver is a set of features that enable bi-directional querying between on-premises and AWS over private connections.
- Used for enabling DNS resolution for hybrid clouds.

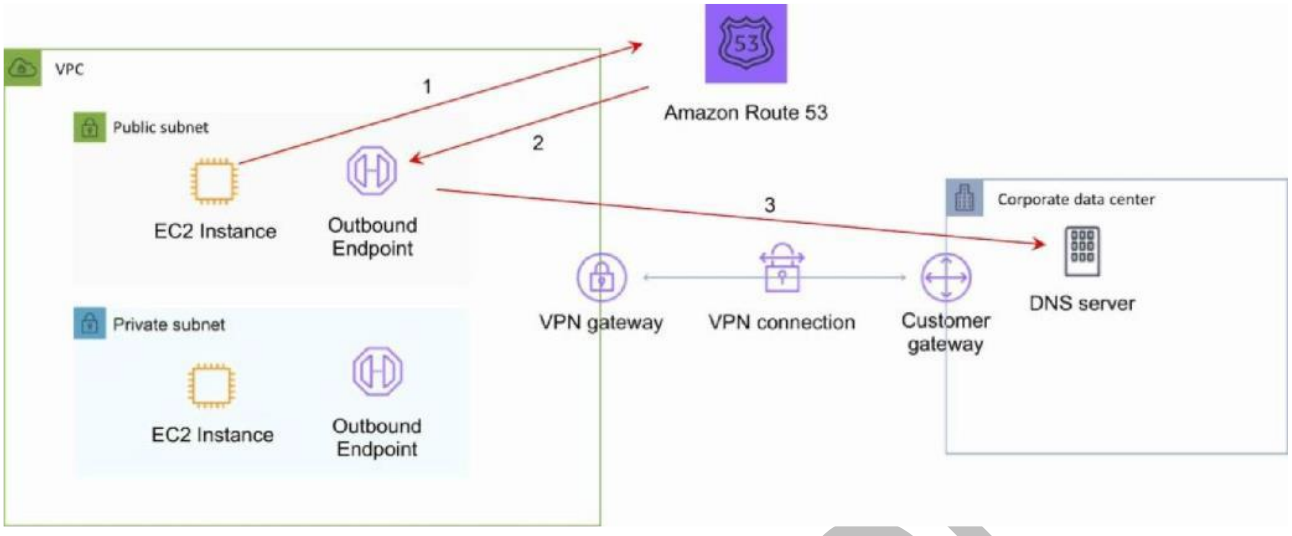
Route 53 Resolver Endpoints:

- Inbound query capability is provided by Route 53 Resolver Endpoints, allowing DNS queries that originate on-premises to resolve AWS hosted domains.
- Connectivity needs to be established between your on-premises DNS infrastructure and AWS through a Direct Connect (DX) or a Virtual Private Network (VPN).
- Endpoints are configured through IP address assignment in each subnet for which you would like to provide a resolver.



Conditional forwarding rules:

- Outbound DNS queries are enabled through the use of Conditional Forwarding Rules. .
- Domains hosted within your on-premises DNS infrastructure can be configured as forwarding rules in Route 53 Resolver.
- Rules will trigger when a query is made to one of those domains and will attempt to forward DNS requests to your DNS servers that were configured along with the rules.
- Like the inbound queries, this requires a private connection over DX or VPN.



4.6 Charges

- You pay per hosted zone per month (no partial months).
- A hosted zone deleted within 12 hours of creation is not charged (queries are charges).

Additional charges for:

- Queries
- Traffic Flow
- Health Checks
- Route 53 Resolver ENIs + queries
- Domain names

Alias records are free of charge when the records are mapped to one of the following:

- Elastic Load Balancers
- Amazon CloudFront distributions
- AWS Elastic Beanstalk environments
- Amazon S3 buckets that are configured as website endpoints
- Health checks are charged with different prices for AWS vs non-AWS endpoints.
- You do not pay for the records that you add to your hosted zones.
- Latency-based routing queries are more expensive.
- Geo DNS and geo-proximity also have higher prices.

4.7 Sample Questions

Q1 : What type of services are associated with Route 53?

A. Storage services
B. Database services
C. Compute services
D. Networking services

Answer : D
Explanation: Route 53 is the DNS service managed by AWS. It provides domain management and registration.

Q2 : How many domains can you register and manage with Route 53?

A. 50
B. 100
C. There is no limit.
D. There is a limit of 50, but it can be raised upon request

Answer : D

Explanation: Route 53 supports up to 50 domain names by default, but this limit can be raised if requested.

Q5 : Which of the following record sets are supported by Route 53?

A. A records
B. MX records
C. Alias records
D. All of the above

Answer : D

Route 53 supports all of the records mentioned, including alias records.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm06

5 AWS GLOBAL ACCELERATOR

- AWS Global Accelerator is a service that improves the availability and performance of applications with local or global users.
- It provides static IP addresses that act as a fixed entry point to application endpoints in a single or multiple AWS Regions, such as Application Load Balancers, Network Load Balancers or EC2 instances.
- Uses the AWS global network to optimize the path from users to applications, improving the performance of TCP and UDP traffic.
- AWS Global Accelerator continually monitors the health of application endpoints and will detect an unhealthy endpoint and redirect traffic to healthy endpoints in less than 1 minute.

Details and Benefits

- Uses redundant (two) static anycast IP addresses in different network zones (A and B).
- The redundant pair are globally advertised.
- Uses AWS Edge Locations – addresses are announced from multiple edge locations at the same time.
- Addresses are associated to regional AWS resources or endpoints.
- AWS Global Accelerator's IP addresses serve as the frontend interface of applications.
- Intelligent traffic distribution: Routes connections to the closest point of presence for applications.
- Targets can be Amazon EC2 instances or Elastic Load Balancers (ALB and NLB).
- By using the static IP addresses, you don't need to make any client-facing changes or update DNS records as you modify or replace endpoints.
- The addresses are assigned to your accelerator for as long as it exists, even if you disable the accelerator and it no longer accepts or routes traffic.
- Does health checks for TCP only – not UDP.
- Can assign target weight within a region to control routing and also "dial" up or down traffic to a region.

Fault tolerance:

- Has a fault-isolating design that increases the availability of your applications.
- AWS Global Accelerator allocates two IPv4 static addresses that are serviced by independent network zones.

- Similar to Availability Zones, these network zones are isolated units with their own set of physical infrastructure and service IP addresses from a unique IP subnet.
- If one IP address from a network zone becomes unavailable, due to network disruptions or IP address blocking by certain client networks, client applications can retry using the healthy static IP address from the other isolated network zone.

Global performance-based routing:

- AWS Global Accelerator uses the vast, congestion-free AWS global network to route TCP and UDP traffic to a healthy application endpoint in the closest AWS Region to the user.
- If there's an application failure, AWS Global Accelerator provides instant failover to the next best endpoint.

Fine-grained traffic control:

- AWS Global Accelerator gives you the option to dial up or dial down traffic to a specific AWS Region by using traffic dials.
- The traffic dial lets you easily do performance testing or blue/green deployment testing for new releases across different AWS Regions, for example.
- If an endpoint fails, AWS Global Accelerator assigns user traffic to the other endpoints, to maintain high availability.
- By default, traffic dials are set to 100% across all endpoint groups so that AWS Global Accelerator can select the best endpoint for applications.

Continuous availability monitoring:

- AWS Global Accelerator continuously monitors the health of application endpoints by using TCP, HTTP, and HTTPS health checks.
- It instantly reacts to changes in the health or configuration of application endpoints, and redirects user traffic to healthy endpoints that deliver the best performance and availability to end users.

Client affinity:

- AWS Global Accelerator enables you to build applications that require maintaining state.
- For stateful applications where you need to consistently route users to the same endpoint, you can choose to direct all requests from a user to the same endpoint, regardless of the port and protocol.

Distributed denial of service (DDoS) resiliency at the edge:

- By default, AWS Global Accelerator is protected by AWS Shield Standard, which minimizes application downtime and latency from denial of service attacks by using always-on network flow monitoring and automated in-line mitigation.
- You can also enable AWS Shield Advanced for automated resource-specific enhanced detection and mitigation, as well as 24x7 access to the AWS DDoS Response Team (DRT) for manual mitigations of sophisticated DDoS attacks.

5.1 Sample Questions

Q1 : A start-up company has a web application based in the us-east-1 Region with multiple Amazon EC2 instances running behind an Application Load Balancer across multiple Availability Zones. As the companyâ€™s user base grows in the us-west-1 Region, it needs a solution with low latency and high availability.

What should a solutions architect do to accomplish this?

A. Provision EC2 instances in us-west-1. Switch the Application Load Balancer to a Network Load Balancer to achieve cross-Region load balancing.

B. Provision EC2 instances and an Application Load Balancer in us-west-1. Make the load balancer distribute the traffic based on the location of the request.

C. Provision EC2 instances and configure an Application Load Balancer in us-west-1. Create an accelerator in AWS Global Accelerator that uses an endpoint group that includes the load balancer endpoints in both Regions.

D. Provision EC2 instances and configure an Application Load Balancer in us-west-1. Configure Amazon Route 53 with a weighted routing policy. Create alias records in Route 53 that point to the Application Load Balancer.

Answer: C

Explanation: Register endpoints for endpoint groups: You register one or more regional resources, such as Application Load Balancers, Network Load Balancers, EC2

Instances, or Elastic IP addresses, in each endpoint group. Then you can set weights to choose how much traffic is routed to each endpoint.

Endpoints in AWS Global Accelerator

Endpoints in AWS Global Accelerator can be Network Load Balancers, Application Load Balancers, Amazon EC2 instances, or Elastic IP addresses. A static IP address serves as a single point of contact for clients, and Global Accelerator then distributes incoming traffic across healthy endpoints. Global Accelerator directs traffic to endpoints by using the port (or port range) that you specify for the listener that the endpoint group for the endpoint belongs to.

Each endpoint group can have multiple endpoints. You can add each endpoint to multiple endpoint groups, but the endpoint groups must be associated with different listeners.

Q2 : Which of the following are benefits of AWS Global Accelerator? (Choose two.)

A. Reduced cost to run services on AWS
B. Improved availability of applications deployed on AWS
C. Higher durability of data stored on AWS
D. Decreased latency to reach applications deployed on AWS
E. Higher security of data stored on AWS

Answer: BD

Explanation: AWS Global Accelerator is a networking service that helps you improve the availability and performance of the applications that you offer to your global users. AWS Global Accelerator is easy to set up, configure, and manage. It provides static IP addresses that provide a fixed entry point to your applications and eliminate the complexity of managing specific IP addresses for different AWS Regions and Availability Zones.

For more Questions Please check Certification Sample Quiz under each module

Link: https://k21academy.com/awssaquizm06