Route 53 - Global DNS

R53 Public Hosted Zones

The VPC+2 address is the address of the Route 53 resolver that is associated with the VPC. The VPC+2 address is the address that you use to forward DNS queries from your VPC to the Route 53 resolver.

R53 Hosted Zones (HZs):

- An R53 Hosted Zone is a DNS DB for a domain
- R53 is GLOBALLY RESILIENT
- HZs are created automatically when you register a domain using R53. But HZs can be created separately
- HZs host DNS records for a domain (eg. A, CNAME, MX, etc)
- HZs are what the DNS system references; authoritative source to confirm a domain Eg. animals4life.org
- Two types of hosted zones in R53: public and private

Public Hosted Zones

A PUBLIC hosted zone is a container that holds information about how you want to route traffic on the internet for a specific domain which is accessible from the public internet

- Creating a Public HZ creates 4 R53 Nameservers that are all accessible from public internet and VPCs
- Monthly cost for hosting public HZ and small cost for queries against it

R53: Private Hosted Zones

Like public hosted zones but for VPC's within AWS. A private hosted zone is a container that holds information about how you want Route 53 to respond to DNS queries for a domain and its subdomains within one or more VPCs that you create with the Amazon VPC service

- Private Hosted Zones are ALWAYS associated with a VPC and are only accessible within those VPCs
- To gain access to a private hosted zone, you need to be querying from within a VPC, AND that VPC needs to be explicity associated with the private host (through VPC+2 address)
- Enabled split-view (overlap public / private) for PUBLIC and INTERNAL use with the same zone name (eg. use same address and only intranet is shown for internal traffic from within local network but user sess website when accessed from public internet). To implement: Public Hosted zone is subset of private hosted zone for the same domain, that contains the records that have to be public. Private hosted zone contains all records, including the public ones.

R53: CNAME vs R53 Alias

The issue with only using CNAMEs:

- A record maps a NAME to IP Address ie. catagram.io => 1.3.3.7.
- CNAME on the other hands, maps NAME to NAME (ie. www.catagram.io => catagram.io)

`Problem:` can't use a CNAME record for the naked/apex of a domain (without www.). AWS services like ELB don't give you an IP address, they only use DNS names, so a naked/apex domain like catagram.io using a CNAME record would be invalid within ELB. A normal DNS record like www.catagram.io would work fine with a CNAME record like www.catagram.io => catagram.io.

`Solution:` R53 Alias records let you use CNAME for domain apex EXAM: Question about when to use CNAME vs Alias. -> For anything that is not the apex/naked domain, CNAME is fine. For the apex/naked domain, you have to use Alias. But use Alias as default option.

R53 Alias

- ALIAS records map a NAME to an AWS Resource
- ALIAS records can be used for naked/apex and normal records. Alias functions like CNAME for non-apex/non-naked domains
- No charge for ALIAS requests pointing at AWS resources
- Alias record is a subtype: You can have an A type record alias and CNAME type record alias. You have to match the type of the record to the type of the record you're pointing at. Eg. An A record alias can only point to an A record, CNAME alias can only point to a CNAME record EXAM: For AWS Services, ALIAS should be your default choice

Alias record is outside of DNS standard so it only functions if Route 53 is hosting your domains.

R53: Simple Routing

Simple routing lets you configure standard DNS records, with no special Route 53 routing such as weighted or latency. With simple routing, you typically route traffic to a single resource, for example, to a web server for your website.

- Simple Routing supports 1 record per name (www)
- Each record can have multiple values
- Simple Routing does not support health checks EXAM: Use Simple Royuting when you want to route requests towards ONE SERVICE such as a web server

R53 Health Checks

Amazon Route 53 health checks monitor the health and performance of your web applications, web servers, and other resources. Health checks supports:

- TCP
- HTTP/HTTPS: must received 2xx or 3xx response within 2s
- HTTP/HTTPS with String Matching must receive a 2xx or 3xx response and contain a string within 2s that matches a string you specify. Most strict health check because you verify the content of the response.

Each health check that you create can monitor one of the following:

- The health of a specified resource, such as a web server
- The status of other health checks
- The status of an Amazon CloudWatch alarm

- Health checkers are located GLOBALLY (you can access AWS services or resources from anywhere in the world with an IP address)
- Health Checks occur every 30s (or every 10s if you pay more)
- States: Healthy or Unhealthy
- Types:
  - Endpoint: HTTP, HTTPS, TCP
  - CloudWatch Alarm: Monitor a CloudWatch Alarm
  - Calculated: Combine other health checks
- Globally, you have distributed health checkers. If 18%+ of health checkers report status as healthy, the health check returns healthy
- In most cases, an UNHEALTHY record is NOT returned in queries

R53 - Failover Routing

Failover routing lets you route traffic to a resource when the resource is healthy or to a different resource when the first resource is unhealthy

- Use when you want to configure "active passive failover"

DEMO - Using R53 and Failover Routing - NOTE: This demo requires an R53 registered domain

Create Failover routing and private hosted zones.
https://learn.cantrill.io/courses/1820301/lectures/41301585

R53 - Multi Value Routing

Multivalue value routing lets you configure Amazon Route 53 to return multiple values, such as IP addresses for your web servers, in response to DNS queries. You can specify multiple values for almost any record, but multivalue answer routing also lets you check the health of each resource, so Route 53 returns only values for healthy resources

- Improves availability, but NOT a replacement for load balancing
- Up to 8 healthy records are returned. If you have more than 8 records, 8 at random are returned

R53 - Weighted Routing

Weighted routing lets you associate multiple resources with a single domain name (catagram.io) and choose how much traffic is routed to each resource behind the record (www). This can be useful for a variety of purposes:

- simple load balancing. Simple because it's not taking current load into account, just distributing traffic evenly according weight value.
- or testing new versions of software: direct 10% of traffic to new version, 90% to old version

Each record is returned based on the weight assigned to it.

- Can assign weights to each hosted zone. Eg. 3 Hosted Zones, 40, 40, 20, totalling 100. First 2 each get 40% of traffic, last only gets 20% of traffic. This doesn't need to add to 100
- A 0 weight means record is never returned unless all are 0, then all are considered
- If a chosen record is unhealthy, the selection process based on weight is repeated until a healthy record is found.

R53 - Latency Routing

Should be used when trying to `optimize for performance and user experience` by serving their request to the AWS region with the lowest latency.

- Latency-based routing supports `one record` with the same name in `each region`
- In the background, AWS mantains a database that contains a latency table between users and regions and matched user to lowest latency
- Latency DB is NOT real time
- can be combined with health checks: if a region is unhealthy the next lowest latency region is used

R53 - Geolocation Routing

Geolocation routing lets you choose the resources that serve your traffic based on the geographic `location` of your users, meaning the location that DNS queries originate from.

- Geolocation records are tagged with a `location`
- Does not return the closest record, but the only the relevant record to the user location or the default (if present, because optional) or no answer. "Default" is backup answer
- Good for:
  - regional restrictions: serve content only if user is in a specific location (state, country, continent)
  - ser language specific content
  - or load balancing across regional endpoints IMPORTANT to understant: Geolocation routing is used to route traffic based on customer location. NOT on proximity, but based on tag and specificity. You can control the response based on the location tag on the record of the user. EXAM: Geolocation routing NOT about closest record, but returns served content in RELEVANT locations only or nothing (if used as restriction).

R53 - Geoproximity Routing

Geoproximity routing lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources. The distance between the location of your users and the AWS resources is calculated and used to define where to route traffic.

- You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a bias for the specific AWS region. --> The BIAS expands or shrinks the size of the geographic AWS region from which traffic is routed to a resource. You can define a plus or minus bias, to increase or decrease the size of the region.

R53 Interoperability

How Route53 provides Registrar and DNS Hosting features and steps through architectures where it is used for BOTH, or only one of those functions - and how it integrates with other registrars or DNS hosting.

R53 does two things:

1. Domain Registrar
2. Domain Hosting. It can do both or either, user choice

R53 - When you REGISTER domain [Domain Hosting / Domain Registrar]

1. Accepts your money (domain registration fee) [DR]
2. Allocates 4 name servers (NS) [DH]
3. Creates a Zone File (domain hosting) on the above NS's [DH]
4. R53 communicates with registry of TLD (domain registrar) [DR]

R53 - DNSSEC

DNSSEC strengthens authentication in DNS using digital signatures based on public key cryptography. With DNSSEC, it's not DNS queries and responses themselves that are cryptographically signed, but rather DNS data itself is signed by the owner of the data.