

# Amazon Route 53: Zones, Records, Policies, and Health Checks



**Andru Estes**

Principal Author

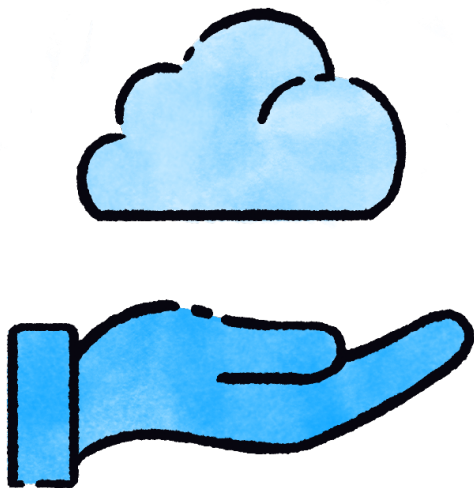
 andru-estes





# **Global DNS with Amazon Route 53**

# A DNS Review



If you've used the internet, you've used DNS!

DNS is used to convert human-friendly domain names into an Internet Protocol (IP) address

**Example:** `http://pluralsight.com` to `http://44.241.44.149`

Remember that IP addresses are used by computers to identify each other on networks

IP addresses come in two common forms:

- IPv4 (`1.1.1.1`)
- IPv6 (`2001:0db8:85a3:0000:0000:8a2e:0370:7334`)

# Top-level Domains (TLD) and Second-level Domains (SLD)

If we look at common domain names, you will notice a string of characters separated by dots (*periods*)

## Examples

google.com, www.pluralsight.com

# Top-level Domains (TLD) and Second-level Domains (SLD)



The last word in a domain name represents the top-level domain, or the TLD ([www.pluralsight.com](http://www.pluralsight.com))



The second to the last word in a domain name is known as a second-level domain, or the SLD ([www.pluralsight.com](http://www.pluralsight.com))



Anything in addition to those two is considered a subdomain ([www.pluralsight.com](http://www.pluralsight.com))

**Amazon Route 53 is a highly available, fully managed, and scalable Domain Name System (DNS) web service.**

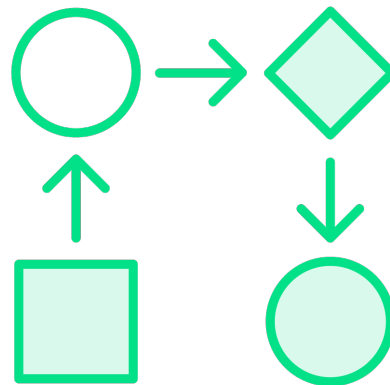
**Route 53 offers an SLA of  
100% availability!**

**Fun Fact: Port 53 is the port number for DNS traffic, thus the name Route 53.**

# What Route 53 Does



Domain Registrations



DNS Routing



Health Checks

**Route 53 is an Authoritative DNS service, meaning that you control all updates of the DNS records!**



## Domain Registrar

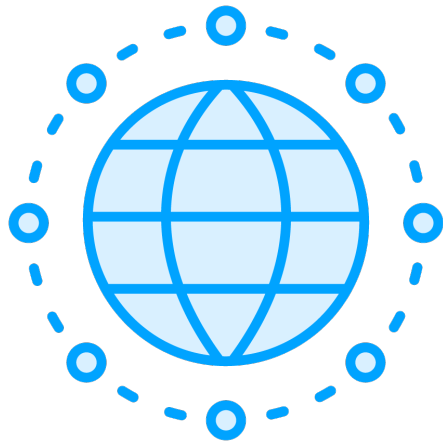
A registrar is an authority that can assign domain names directly under one or more top-level domains. These domains are registered with InterNIC, a service of ICANN, which enforces uniqueness of domain names across the internet.



# Managing Domains with Route 53 Hosted Zones

**Hosted Zones are containers for records to specify how you route traffic for your domains or subdomains.**

# The Two Types of Hosted Zones



## Public Hosted Zone

Contains records for routing traffic on the public internet



## Private Hosted Zone

Contains records for routing traffic within an Amazon VPC.

**Public Hosted Zones: Used for public domains that you own or have delegated control of.**

**Private Hosted Zones: Used for private domain names, and get associated with VPCs for query resolution.**

**\$0.50** per hosted zone per month for first 25 zones

**\$0.10** per hosted zone per month for additional zones

**Additionally charged for DNS queries...**



# **Amazon Route 53 Records**

<https://t.me/learningnets>



## Records

DNS records dictate how and where you want your DNS traffic routed

# What Makes up a Record?



**Domain or Subdomain Name:** pluralsight.com



**Record Type:** A, CNAME, MX, NS, SOA



**Value:** 52.217.202.128 or AWS Resource Alias



**Routing Policy:** Policy that sets how to respond to a DNS query



**Time to Live (TTL):** Number of seconds DNS resolvers cache a record value

# Important Route 53 Record Types



**A**

Routes traffic to an IPv4 address



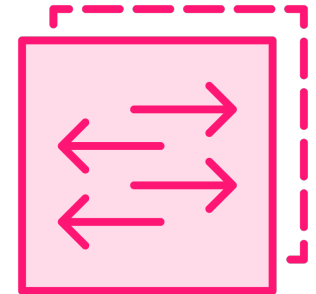
**AAAA**

Routes traffic to an IPv6 address



**NS**

Identifies the name servers for the hosted zone



**Alias**

Special record that routes traffic to AWS resources

## Important Route 53 Record Type Value Examples

### **A:**

- Public IPv4 address of an EC2 instance
- 33.4.5.6

### **AAAA:**

- Public IPv6 address of an EC2 instance
- 2001:0db8:85a3:0000:0000:8a2e:0370:7334

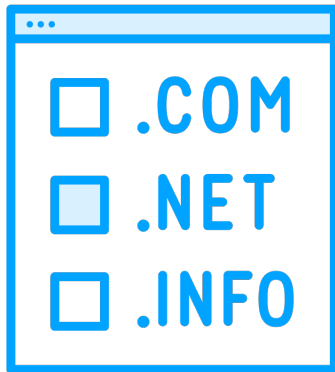
### **NS:**

- ns-1234.awsdns-45.org.

### **Alias:**

- Public DNS name of a CloudFront distribution
- d111111abcdef8.cloudfront.net

# Route 53 CNAME Record Type



## CNAME

Maps one DNS name to another domain.

www.acloud.guru →  
www.pluralsight.com



## Important Information

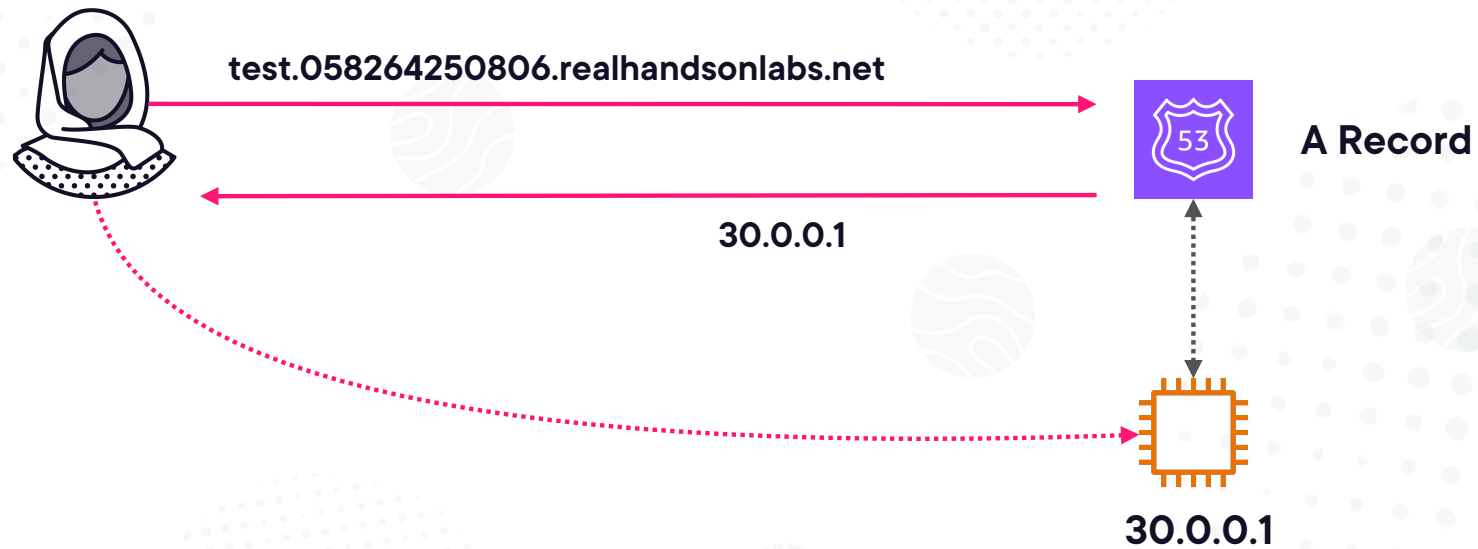
CANNOT be created for the Zone Apex.

Example: pluralsight.com

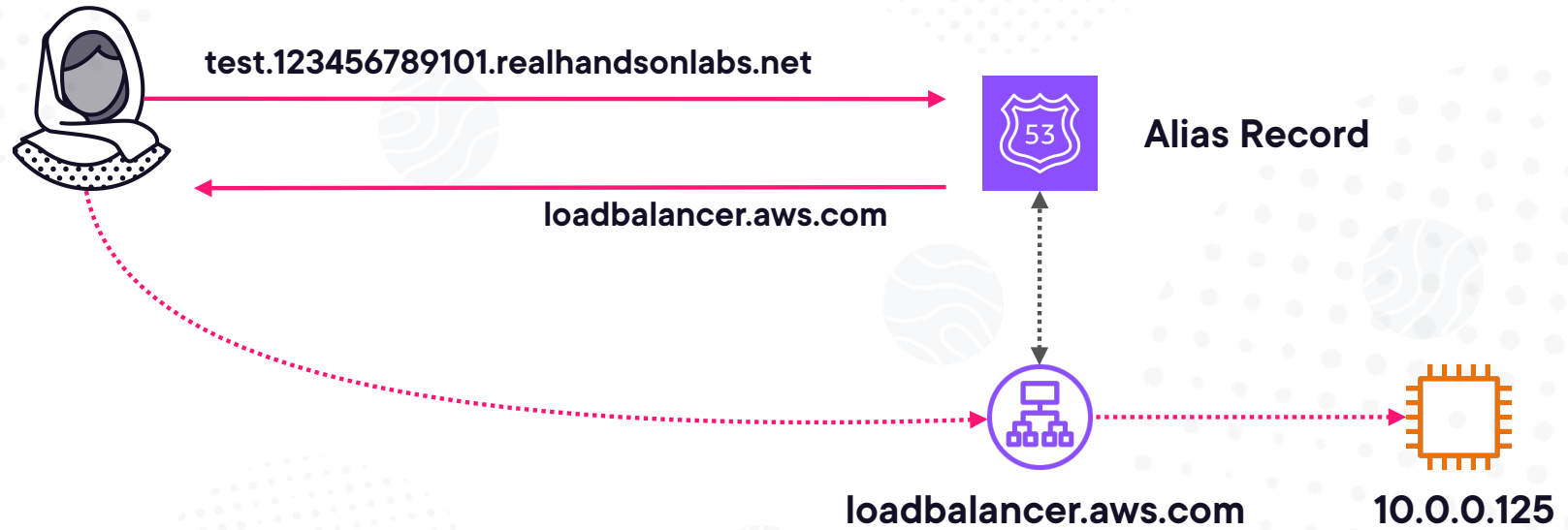
**Understand the difference  
between an alias record and  
a CNAME record.**

**Given the choice, you should likely choose an alias record over a CNAME.**

# Demo: Creating an “A” Route 53 Record



# Demo: Creating an “Alias” Route 53 Record





# **Amazon Route 53 Routing Policies**

# Routing Policy

A setting for records that determines how Route 53 responds to DNS queries.

---

Citation: AWS Documentation, <https://go.aws/3VXVx2H>

# Simple Routing Policy

Usually used for single resources and IP addresses

Alias records point to a single AWS resource, like an elastic load balancer

Useable within Private Hosted Zones

Supports multiple values for the same record\*

*\* Route 53 returns all values in a random order, meaning a random value will be used!*

# Weighted Routing Policy

**Set multiple resources under a single domain name or subdomain name**

**Control percentage of traffic that is routed to each resource**

**You must use records that have the same name and type**

**Assign each record a relative weight for controlling traffic**

**Set weight to zero to stop routing traffic to a resource**

# Calculating Weights

Weight for one record

---

Sum of weights for all records

# Weighted Routing Policy Calculation Example

Let us assume we have **4 resources** that we want to use

**1** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** 25  
**Value:** 192.0.2.10

**2** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** 10  
**Value:** 192.0.2.20

**3** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** 15  
**Value:** 192.0.2.30

**4** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** 50  
**Value:** 192.0.2.40

# Weighted Routing Policy Calculation Example

Let us assume we have **4 resources** that we want to use

**1** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** **25**  
**Value:** 192.0.2.10

**2** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** **10**  
**Value:** 192.0.2.20

**3** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** **15**  
**Value:** 192.0.2.30

**4** **Record Name:** www.pluralsight.com  
**Type:** A  
**TTL:** 600  
**Weight:** **50**  
**Value:** 192.0.2.40

# Weighted Routing Policy Calculation Example

Let us assume we have **4 resources** that we want to use

**1** Record Name: www.pluralsight.com  
Type: A  
TTL: 600  
Weight: **25**  
Value: 192.0.2.10

**2** Record Name: www.pluralsight.com  
Type: A  
TTL: 600  
Weight: **10**  
Value: 192.0.2.20

**3** Record Name: www.pluralsight.com  
Type: A  
TTL: 600  
Weight: **15**  
Value: 192.0.2.30

**4** Record Name: www.pluralsight.com  
Type: A  
TTL: 600  
Weight: **50**  
Value: 192.0.2.40

The weight for record number 4 would be **50/100**

# Failover Routing Policy



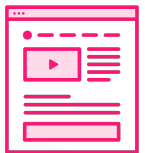
Routes traffic to a resource that is healthy and once that resource is marked unhealthy, it only responds with the healthy secondary resources



This is also commonly referred to as **Active-passive** routing



Leverages and requires health checks!

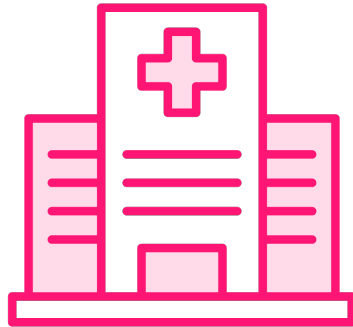


**Summary:** Primary resources are to be available the majority of the time and you want a secondary resource or group of resources to be on standby

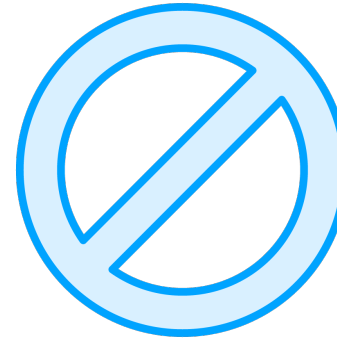
# Multivalued Answer Routing Policy



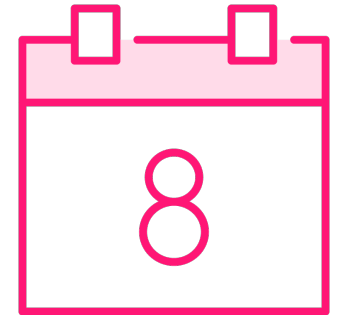
**Route 53 returns multiple values for DNS queries**



**Use Health Checks to return only healthy resources**



**Not meant to replace proper load balancing**



**Eight (or less) healthy records returned at once**



# **Additional Amazon Route 53 Routing Policies**

# Geolocation Routing Policy

**Choose where your traffic is sent based on the geographic location of your users**

**Geographic location, in this case, means where the DNS queries originate from**

**Perfect for localization (languages displayed) or restricting access to content based on location**

**Use Case: You want all European queries to go to a web server hosted in eu-west-2**

**Ensure you have a default record in place just in case there is no match to the location set in the policy!**

**Exam Pro Tip: These are not based on latency!**

# Latency-based Routing Policy

Allows you to route your traffic based on the lowest network latency for your end user

This will be the Region that provides the fastest response time

Create a latency resource record set for the resource in each Region that hosts your website

# Latency-based Routing Policy Exam Pro Tips



Consider this routing policy anytime that latency is the primary concern for your architecture



Users are always directed to the fastest lowest latency resource, even if it is across the globe in another AWS Region



Do not confuse these with Geolocation-based routing policies



# Geoproximity Routing Policy

Policy that lets Amazon Route 53 route traffic to your resources based on the geographic location of your users and your resources

You can also optionally choose to route more traffic or less to a given resource by specifying a value, known as a **bias**

A **bias** expands or shrinks the size of the geographic region from which traffic is routed to a resource

Expanding/growing sizes of biases:

- Specific positive integers from 1 to 99

Shrinking sizes of biases:

- Specific negative numbers from -1 to -99

# Geoproximity Map Example

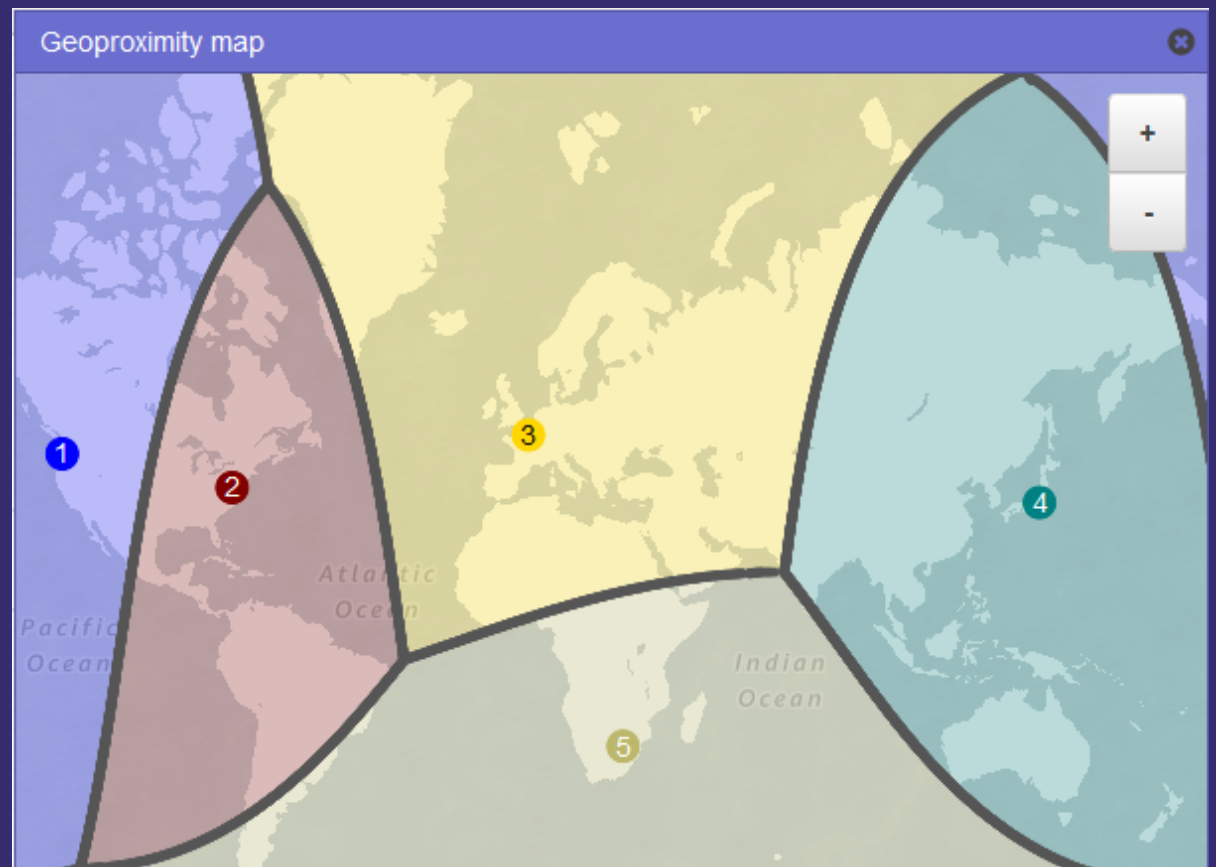


Image source: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy-geoproximity.html>

# Route 53 Traffic Flow for Geoproximity Routing

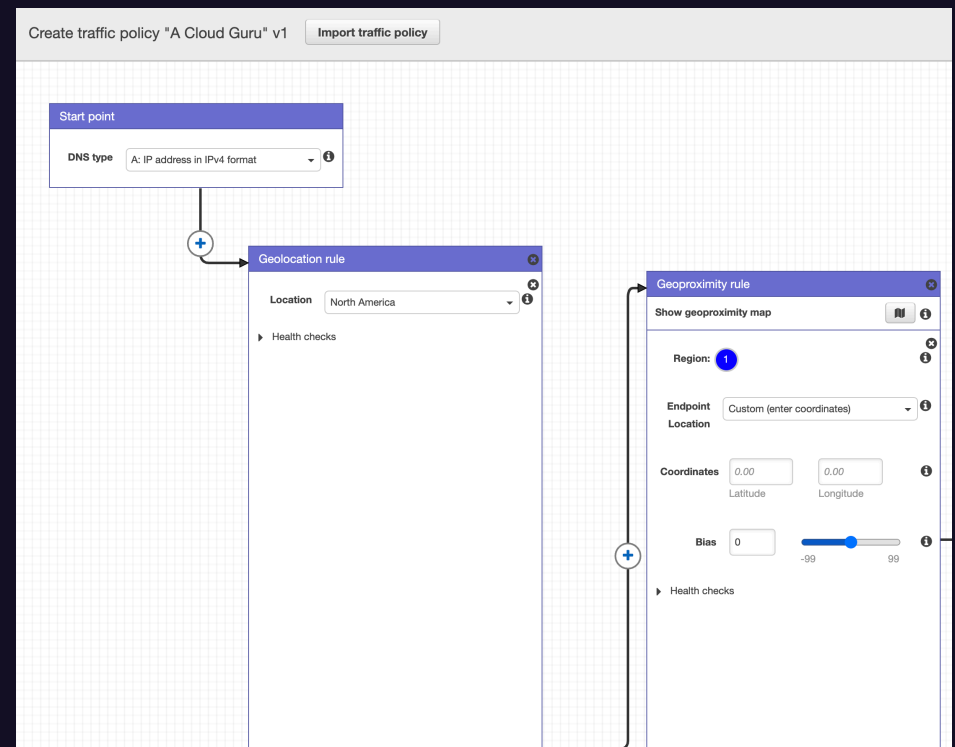
You **must** use this feature to access the maps used in the previous image!

You can use Route 53 traffic flow to build a routing system that uses a combination of:

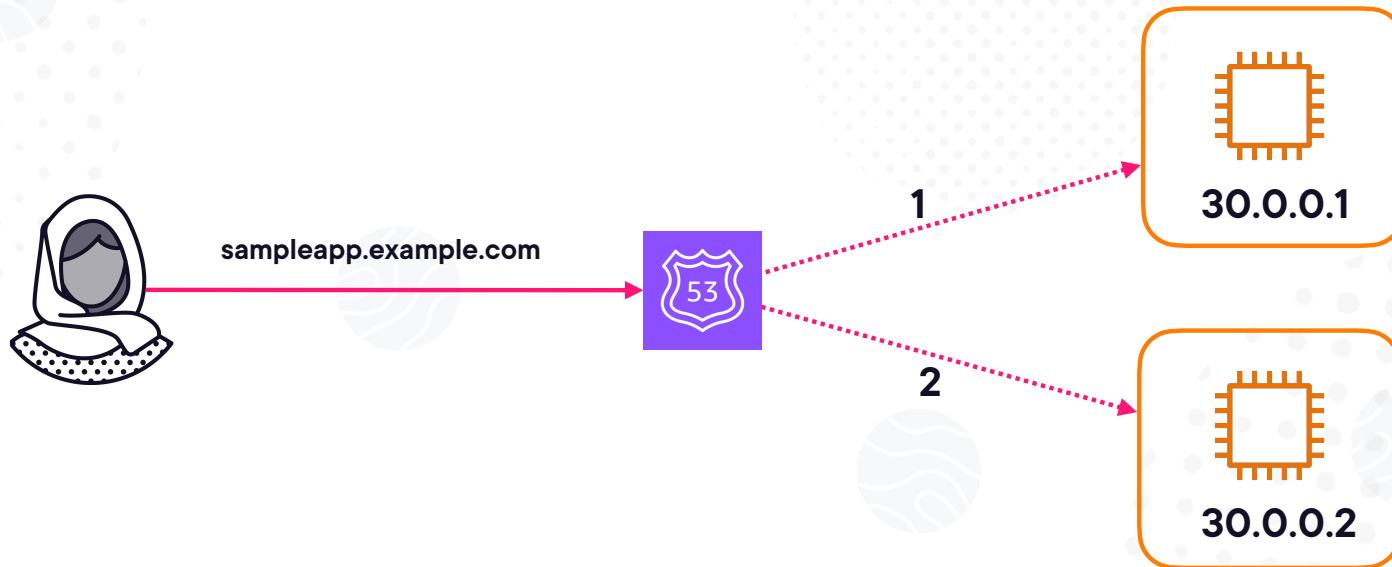
- **Geographic location**
- **Latency**
- **Availability to route traffic**

You can build your traffic routing policies:

- **From scratch**
- **You can pick a template from a library and then customize it**

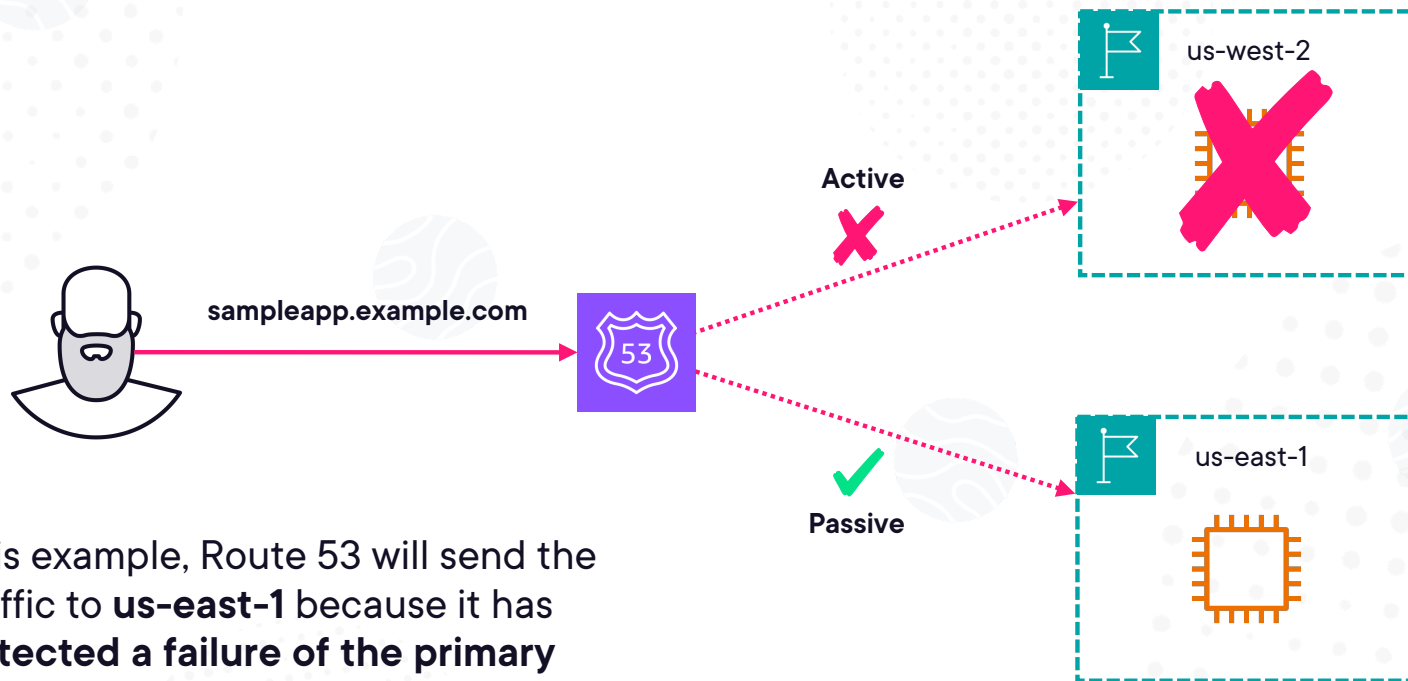


# Demo: Creating a Simple Routing Policy



If you choose the simple routing policy, you can only have **one record with multiple IP addresses**. If you specify multiple values in a record, **Route 53 returns all values to the user in a random order**.

# Demo: Creating a Failover Routing Policy



In this example, Route 53 will send the traffic to **us-east-1** because it has **detected a failure of the primary record resource in us-west-2**

# Demo: Creating a Multivalued Routing Policy

We will query our DNS entry for our 4 web servers that we set up our multivalued answer routing policy for



Each time we query the DNS name for our Route 53 record, the record values will be returned in a random order

**Important:** This is not a suitable replacement for load balancing!



# **Health Checking Route 53 Resources**

<https://t.me/learningnets>



## Route 53 Health Checks

You can set health checks on individual record sets in Route 53

If a record set fails a health check, it will be “removed” from Route 53 until it passes the health check

You can set SNS notifications to alert you about failed health checks

Image source: <https://unsplash.com>

# Three Health Check Types to Know

**Monitoring an  
endpoint**

**Monitoring other  
health checks**

**Monitoring  
CloudWatch Alarms**

# Health Check Types - Monitoring an Endpoint

**Route 53 uses ~15 global health checkers to monitor**

**Intervals can be set to 30 seconds or 10 seconds**

**You can choose the Regions where you want to check from**

**Health checks result in a HTTP 2XX or 3XX status code**

**Pro Tip: Ensure your firewalls allow the health checkers**

# Health Check Types - Monitoring Other Health Checks

These are commonly referred to as Calculated Health Checks!

Use these by combining multiple health checks into a single one

The monitored health checks are called Child Health Checks (*up to 256 at a time*)

Configure the checks with AND, OR, or NOT logic

Useful with multi-layered applications where the app should stay up as long as one endpoint is healthy

# Health Check Types - Monitoring CloudWatch Alarms



Create a CloudWatch Metric and associate a CloudWatch Alarm to use as a health check endpoint



The health check fails whenever the CloudWatch Alarm goes into an **ALARM** state



**Exam Pro Tip:** These are perfect for using within a Private Hosted Zone where there is no public access to the records!

**Exam Pro Tip: HTTP Health Checks (monitoring endpoints) are only available for publicly resolvable records!**



# **Module Summary and Exam Tips**

**Amazon Route 53 is the  
service to use for DNS  
management within AWS!**

**The service offers domain registration, DNS routing configuration, and health checks of resources.**

# Hosted Zone Exam Tips



**Public Hosted Zones:** Contain records for the public internet to use



An example is a public facing website, like **pluralsight.com!**



**Private Hosted Zones:** Contain records for internal, private networks to use



An example is an internal domain, like **yourcorporation.internal**

**Remember: You must  
associate Private Hosted  
Zones with VPCs!**

# Record Types You Need to Know

**A**

**AAAA**

**Alias**

**NS**

**CNAME\***

**SOA**

\* Alias records are typically going to be preferred over a CNAME in scenarios!

# Know These Routing Policies

Simple Routing

Failover Routing

Weighted Routing

Geolocation Routing

Geoproximity Routing (*Traffic Flow with biases*)

Multivalued Answer Routing

Latency-based Routing

**Important: Do not confuse  
latency-based routing and  
geolocation routing!**

**Remember you can use  
Traffic Flow to set up  
complex, advanced routing  
policies for Route 53.**

# Three Health Check Types to Remember

Monitoring an  
endpoint\*

Monitoring other  
health checks

Monitoring  
CloudWatch Alarms\*\*

\* Requires publicly resolvable IP address

\*\* Perfect for health checking Route 53 Private Hosted Zones!