

*Route*53



**Highly available and scalable cloud Domain Name System (DNS).
Register and manage domains, create DNS routing rules eg. failovers.**



Route53 – Introduction

Route53 is a DNS is a Domain Name Service ***think*** Godaddy or NameCheap but with more synergies with AWS Services.



You can:

- register and manage domains
- create various records sets on a domain
- Implement complex traffic flows eg. Blue/green deploy, failovers
- Continuously monitor records via health checks
- resolve VPC's outside of AWS

Choose a domain name

gaga4president2028

.com - \$12.00

Check

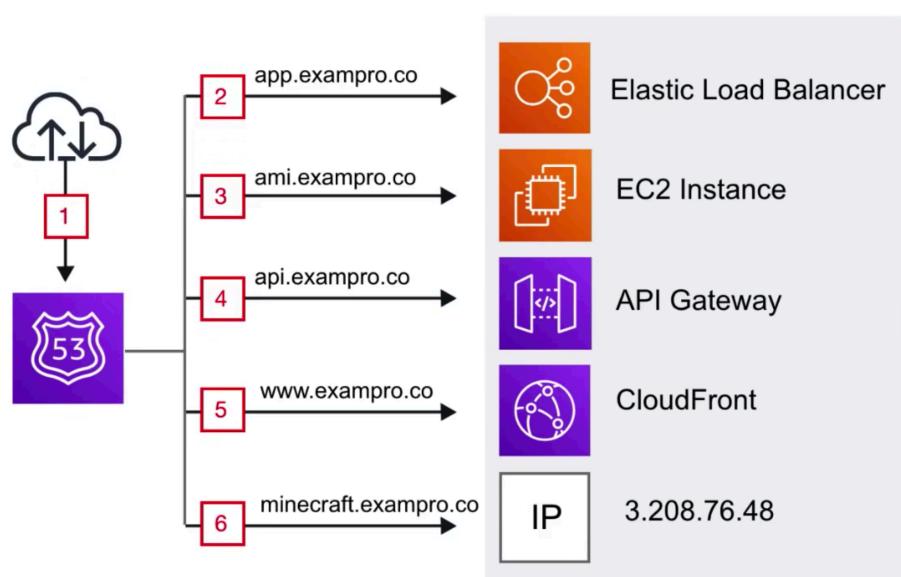
(A)
SUBSCRIBE



Route53 - Use Case

Use Route53 to get your custom domains to point to your AWS Resources

1. Incoming internet traffic
2. Route traffic to our web-app backed by ALB
3. Route traffic to an instance we use to tweak our AMI
4. Route traffic to API gateway which powers our API
5. Route traffic to CloudFront which serves our S3 static hosted website
6. Route traffic to an Elastic IP (EIP) which is a static IP that hosts our company Minecraft server





Route53 – Record Sets

We create record sets which allows us to point our naked domain (exampro.co) and subdomains via Domain records.

For example we can send our www subdomain using an A record to point a specific IP address.

Create Record Set

Name: www.exampro.co.

Type: A – IPv4 address
CNAME – Canonical name
MX – Mail exchange
AAAA – IPv6 address
TXT – Text
PTR – Pointer
SRV – Service
SPF – Sender Policy Framework
NAPTR – Network Address and Port Translation
CAA – Certificate Authority Authorization
NS – Name Server
SOA – Start of Authority
TTL (Seconds): 300, 1m, 5m, 1h, 1d
Value: 192.0.2.235

Alias: Yes No

IPv4 address. Enter multiple addresses on separate lines.
Example:
192.0.2.235
198.51.100.234





Route53 – Routing Policies

There are **7 different types** of Routing Policies available inside Route53

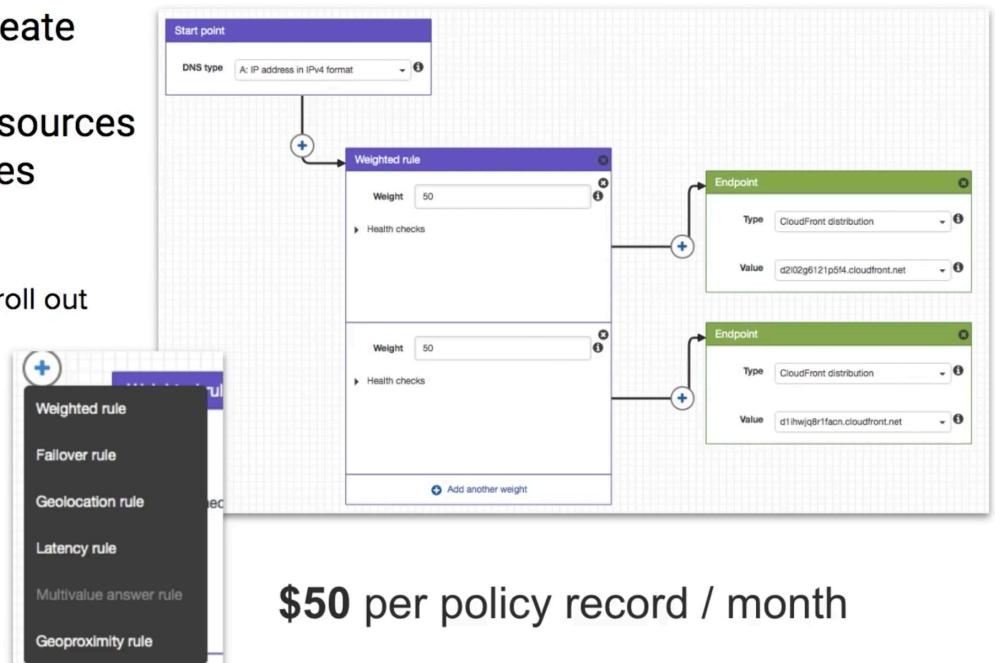
- **Simple Routing** default routing policy, multiple addresses result in random selection
- **Weighted Routing** route traffic based on weighted values to split traffic
- **Latency-Based Routing** route traffic to region resource with lowest latency
- **Failover Routing** route traffic if primary endpoint is unhealthy to secondary endpoint
- **Geolocation Routing** route traffic based on the location of your users
- **Geo-proximity Routing** route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another
- **Multi-value Answer Routing** respond to DNS queries with up to eight healthy records selected at random.



Route53 – Traffic Flow

A visual editor lets you create sophisticated routing configurations for your resources using existing routing types

Supports **versioning** so you can roll out or roll back updates.



SUBSCRIBE



Route53 – Simple Routing Policies

Simple Routing Policies are the most basic routing policies in Route53 **Default Policy**

- You have 1 record and provide multiple IP addresses
- When multiple values are specified for a record, Route53 will return all values back to the user in a **random order**

For example if you had a record for ‘www.exampro.co’ with 3 different IP address values, users would be directly **randomly to 1 of them** when visiting the domain.

The screenshot shows the 'Create Record Set' dialog box. The 'Name' field contains 'random.exampro.co.' and the 'Type' field is set to 'A - IPv4 address'. Under 'Alias', 'No' is selected. The 'TTL (Seconds)' dropdown is set to 300. The 'Value' field lists three IP addresses: 34.229.79.211, 18.212.245.88, and 3.208.76.58. A note below says 'IPv4 address. Enter multiple addresses on separate lines.' An 'Example' section shows the resolved IP addresses: 192.0.2.235, 198.51.100.234, and 198.51.100.234. The 'Routing Policy' is set to 'Simple'. A note at the bottom states: 'Route 53 responds to queries based only on the values in this record. Learn More'.





Route53 – Weighted Routing Policies

Weighted Routing Policies let you split up traffic based on different ‘weights’ assigned.

This allows you to send a certain percentage of overall traffic to one server, and have any other traffic apart from that directed to a completely different server.

For example if you had an ALB running experimental features you could test against a small amount traffic at random to minimize the impact of affect

Routing Policy: Weighted

Route 53 responds to queries based on weighting that you specify in this and other record sets that have the same name and type. [Learn More](#)

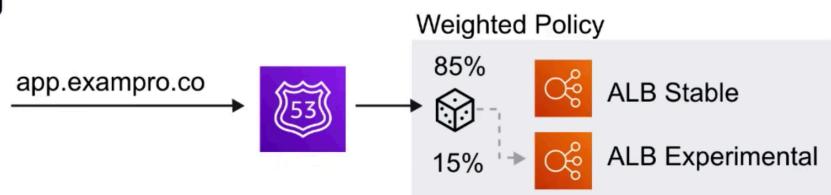
Weight: 20

Set ID: Experiment

Description of this record set that is unique within the group of weighted sets.

Example:
My Seattle Data Center

Associate with Health Check: Yes No





Route53 - Latency Based Routing Policies

Latency Based Routing allows you to direct traffic based on the lowest network latency possible for your end-user **based on region**.

Requires a latency resource record to be set for the EC2 or ELB resource that hosts your application in each region.

For example, You have two copies of your web-app backed by ALB. One in California, US and another in Montreal, Canada. A request comes in from Toronto, Canada

app.exampro.co
Toronto, Canada

Latency Based Policy

100ms → ALB US-WEST-1

12ms → ALB CA-CENTRAL-1

Routing Policy: Latency

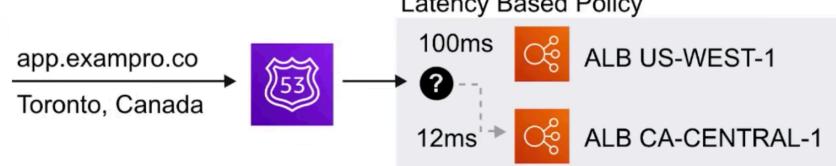
Route 53 responds to queries based on regions that you specify in this and other record sets that have the same name and type. [Learn More](#)

Region: us-east-1

Set ID: Fast Zone

Description of this record set that is unique within the group of latency sets.
Example:
My Seattle Data Center

Associate with Health Check: Yes No





Route53 – Failover Routing Policies

Failover Routing Policies allow you to create active/passive setups in situations where you want a primary site in one location, and a secondary data recovery site in another.

Route53 automatically monitors health-checks from your primary site to determine the health of end-points. If an end-point is determined to be in a failed state, all traffic is automatically directed to the secondary location.

For example, we have a primary and secondary web-app backed by ALB. Route53 determines our primary is unhealthy and fails over to secondary ALB.

Routing Policy: Failover

Route 53 responds to queries using primary record sets if any are healthy, or using secondary record sets otherwise. [Learn More](#)

Failover Record Type: Primary Secondary

Set ID: prod-Primary

Associate with Health Check: Yes No





Route53 – Geolocation Routing Policies

Geolocation Routing Policies allow you to direct traffic based on the geographic location of where the request originated from.

For example this would let you route all traffic coming from North America to servers located in North American regions, where queries from other regions could be directed to servers hosted in that region. (potentially with pricing and language specific to that region)

Routing Policy: **Geolocation**

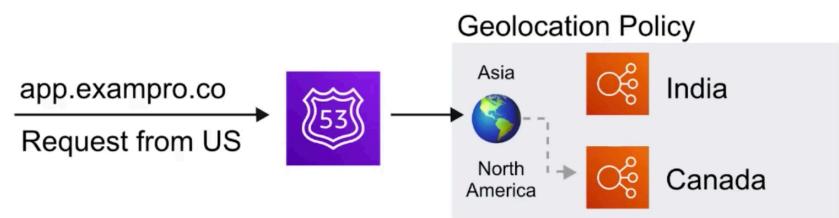
Route 53 responds to queries based on the locations from which DNS queries originate. We recommend that you create a Default location resource record set [Learn More](#)

Location: North America

Set ID: na-Primary

Description of this record set that is unique within the group of geolocation sets.
Example:
Route to Seattle data center

Associate with Health Check: Yes No





Route53 – Geoproximity Routing Policies

Geoproximity Routing Policies allow you to direct traffic based on the geographic location of your users, and your AWS resources.

You can route more or less traffic to a specific resource by specifying a ‘Bias’ value.

Bias values expand or shrink the size of the geographic region from which traffic is routed to.

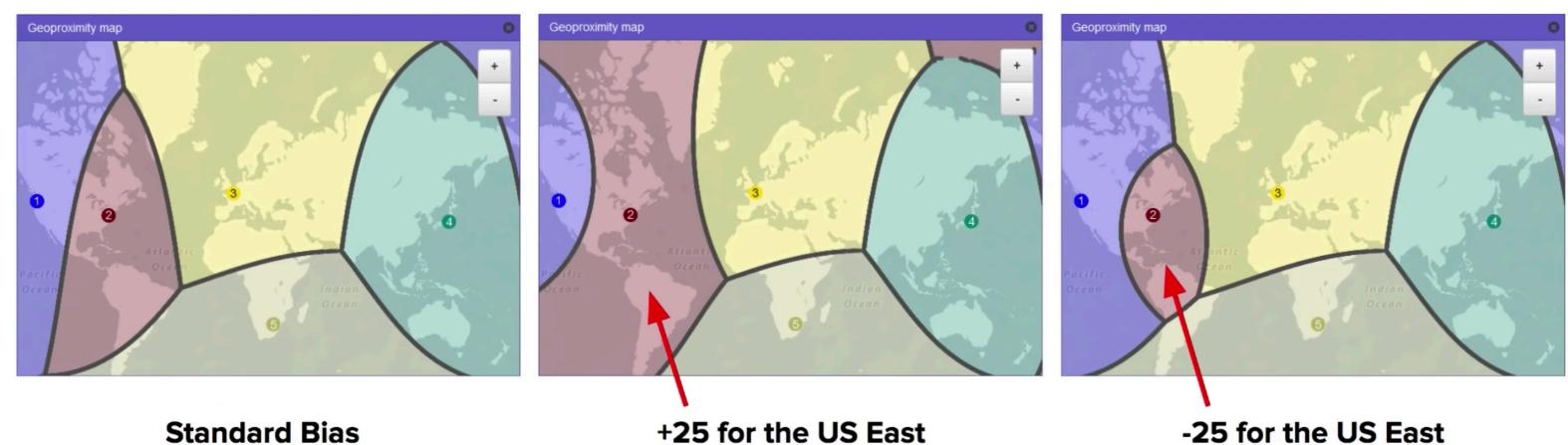
You must use Route53 Traffic Flow in order to use geoproximity routing policies

The screenshot shows the configuration of a Geoproximity rule. It includes fields for Region (set to 1), Endpoint Location (set to US East (N. Virginia)), Coordinates (Using US East (N. Virginia) Coordinates), and Bias (-2). A slider bar indicates the bias range from -99 to 99, with the current value at -2. There is also a 'Health checks' section.



Route53 – Geoproximity Routing Policies

How Bias Works for Geoproximity





Route53 – Geoproximity Routing Policies

In the Route53 Traffic Flow you can select any regions and visualize the bias

Geoproximity rule

Region: ①

Endpoint Location: China (Beijing)

Coordinates: EU (Frankfurt), EU (London), EU (Paris), EU (Stockholm)

Bias: Asia Pacific (Singapore), Asia Pacific (Sydney), Asia Pacific (Seoul), Asia Pacific (Tokyo), Asia Pacific (Hong Kong), Asia Pacific (Mumbai), China (Beijing), China (Ningxia), South America (São Paulo)

Health check

Hide geoproximity map

Geoproximity map

②

①

Pacific Ocean

Atlantic Ocean

Indian Ocean

Southern Ocean

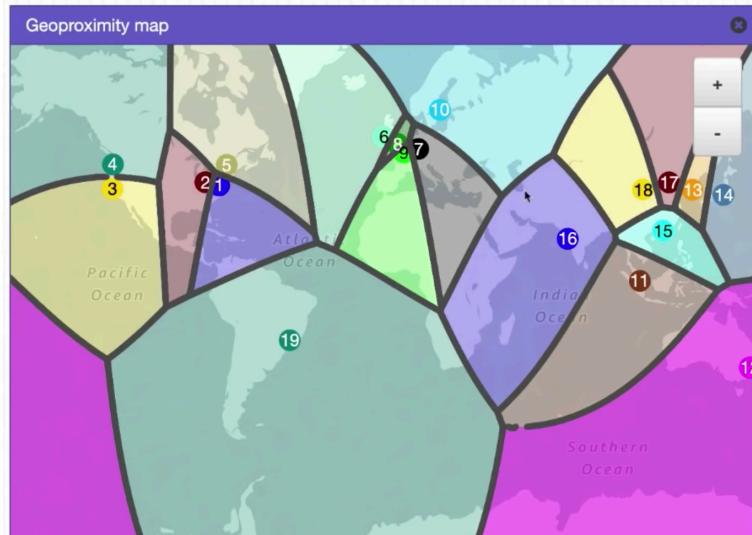
+

-

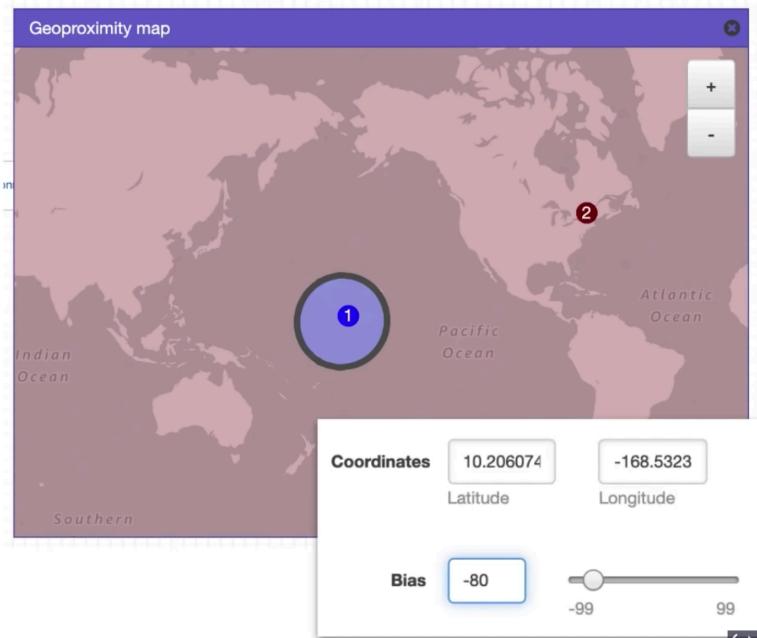


Route53 – Geoproximity Routing Policies

All Current Regions Selected



Can provide **custom coordinates** over region





Route53 – Multi-Value Answer Policies

Multi-Value Answer Policies let you configure Route53 to return multiple values such as IP addresses for your web-servers, in response to DNS queries.

Multiple values can be specified for almost any record. Route53 automatically performs health-checks on resources and only returns values of ones deemed healthy.

Similar to Simple Routing, however with an added health check for your record set resources.

The screenshot shows the AWS Route 53 console configuration for a multi-value answer policy. It includes fields for 'Set ID' (multivalue), a description (Route to Seattle data center), and an 'Associate with Health Check' section with a 'No' radio button selected. The policy itself is shown as a shield icon with a '53' and a dashed arrow pointing to a box containing three IP addresses: 34.229.79.211, 18.212.245.88, and 3.208.76.48.





Route53 – Health Checks

- Checks health every **30s** by default. Can be reduce to every **10s**
- A health check can **initial a failover** if status is returned unhealthy
- A CloudWatch Alarm can be created to alert you of status unhealthy
- A health check can **monitor other health checks** to create a chain of reactions.

can create up to **50 health checks** for AWS endpoints that are within or linked to the same AWS account.

AWS Endpoints	Non-AWS Endpoints
\$0.50* per health check / month	\$0.75 per health check / month
Optional health check features:	
<ul style="list-style-type: none">• HTTPS• String Matching• Fast Interval• Latency Measurement	

[Create health check](#) [Delete health check](#) [Edit health check](#)

Filter by keyword

1 to 1 of 1 health check

Name	Status	Description	Alarms
app.exampro.co	15 minutes ago now Healthy https://app.exampro.co:443/health-check	1 of 1 in OK	



Route53 – Resolver

Formerly known as **.2 resolver**

A regional service that lets you route DNS queries between your VPCs and your network

DNS Resolution for **Hybrid Environments** (On-Premise and Cloud)

Basic configuration

Direction of DNS queries [Info](#)

You can configure endpoints for inbound DNS queries (to your VPC), outbound DNS queries (from your VPC), or both.

Inbound and outbound

Configure endpoints that allow DNS queries both to and from your VPC.



Inbound only

Configure an endpoint that allows DNS queries to your VPC from your network or another VPC.



Outbound only

Configure an endpoint that allows DNS queries from your VPC to your network or another VPC.





Route53 CheatSheet

- Route53 is a DNS provider, register and manage domains, create record sets. Think Godaddy or NameCheap
- Simple Routing - Default routing policy, multiple addresses result in a random endpoint selection
- Weighted Routing - Split up traffic based on different 'weights' assigned (percentages)
- Latency-Based Routing - Directs traffic based on region, for lowest possible latency for users.
- Failover Routing - Primary site in one location, secondary data recovery site in another. (change on health check)
- Geolocation Routing - Route traffic based on the geographic location of a requests origin.
- Geo-proximity Routing - Route traffic based on geographic location using 'Bias' values (needs Route53 Traffic Flow)
- Multi-value Answer Routing - Return multiple values in response to DNS queries. (using health checks)
- Traffic Flow - visual editor, for chaining routing policies, can version policy records for easy rollback
- AWS Alias Record - AWS' smart DNS record, detects changed IPs for AWS resources and adjusts automatically.
- Route53 Resolver - Lets you regionally route DNS queries between your VPCs and your network **Hybrid Environments**
- Health checks can be created to monitor and automatically over endpoints. You can have health checks monitor other health checks