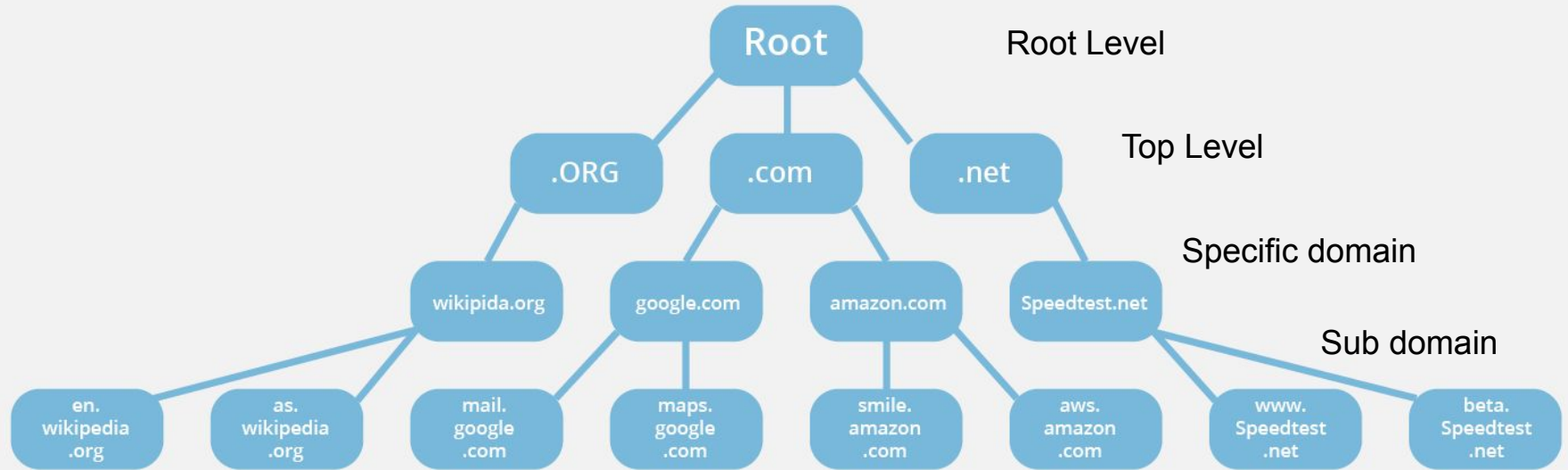


Introduction to DNS & AWS Route53 Services Overview

Agenda

- DNS Service Introduction
 - DNS Overview
 - How DNS Works?
- AWS Route53 Service Overview
- Unique Feature/Functionality Provided by Route53
- Route53 Service Demo
 - How to create Route53 Service?
 - Demo of some Route53 Features

Domain Name System Hierarchy



Source: <https://www.cloudflare.com/en-au/learning/dns/glossary/dns-root-server/>

Domain Name System Overview

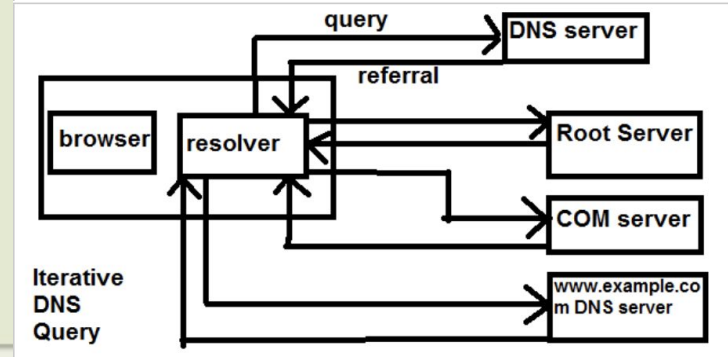
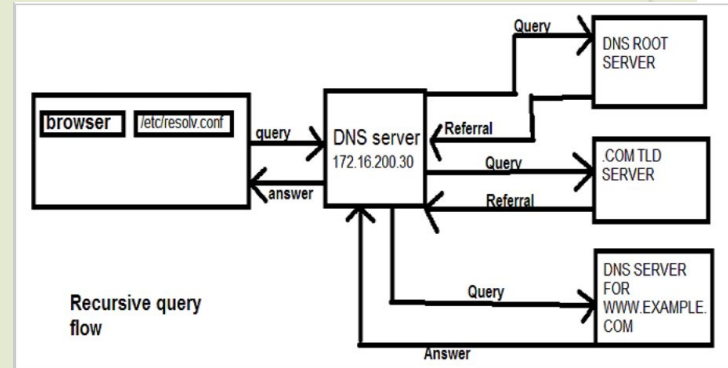
- DNS Management/administration is done in a distributed or in a decentralized manner
- DNS comprises of hierarchy of managed zone with “root” zone at the top of the hierarchy
- ROOT DNS servers are the name servers responsible for root zone/operate in root level
- ROOT DNS Servers answer queries for records within root zone (stored or cached)
- ROOT DNS Servers refer “other” requests to the appropriate TLD/Top Level Domain name servers
- TLD Name servers operate one level beneath the root DNS servers

Name Servers

Recursive Vs Iterative query diagram source:

<https://www.slashroot.in/difference-between-iterative-and-recursive-dns-query>

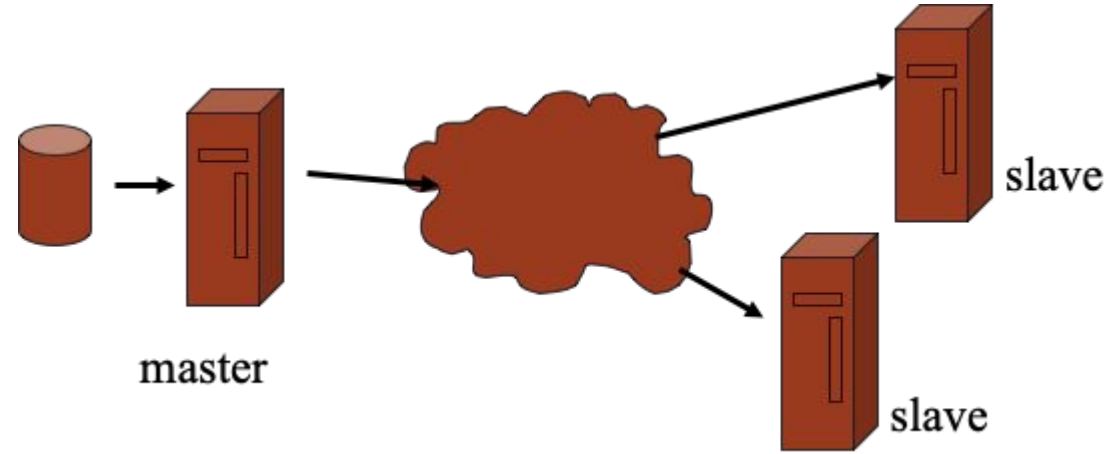
- Name servers answer 'DNS' questions/query
 - recursive vs iterative query
- Several types of name servers
 - Authoritative servers
 - master (primary)
 - slave (secondary)
 - Root servers
 - (Caching) recursive servers
 - also caching forwarders
 - Mixture of functionality



Recursive Vs Iterative Name Server

- Recursive servers do the actual name lookup on behalf of DNS clients
 - Will recursively query Root, TLD till it gets to the authoritative NS for a given domain
 - While it gets answers from Authoritative NS, it will mark them as non-authoritative when it forwards to the clients
 - Answers from a query are stored/cached for future lookups
- Iterative servers does not perform the actual name lookup on behalf of DNS clients
 - Will provide an answer if it has an entry in it's cache
 - If not it will provide only a referral service to the DNS client (Root, TLD)
 - Answers from a query are stored/cached for future lookups (possibly query originated from the NS)

Name Servers-Authoritative Name Server, Pri/Sec



- Provides Authoritative answer for one or more Zone
- Master / Primary server normally loads zone data from a file
- Secondary/Slave DNS servers can be configured to get update from master/primary using a process called zone replication
- Primary/Secondary DNS essentially provides redundancy/resilience and load balancing functionality to DNS request

How Does DNS Route Traffic To Your Web Application?

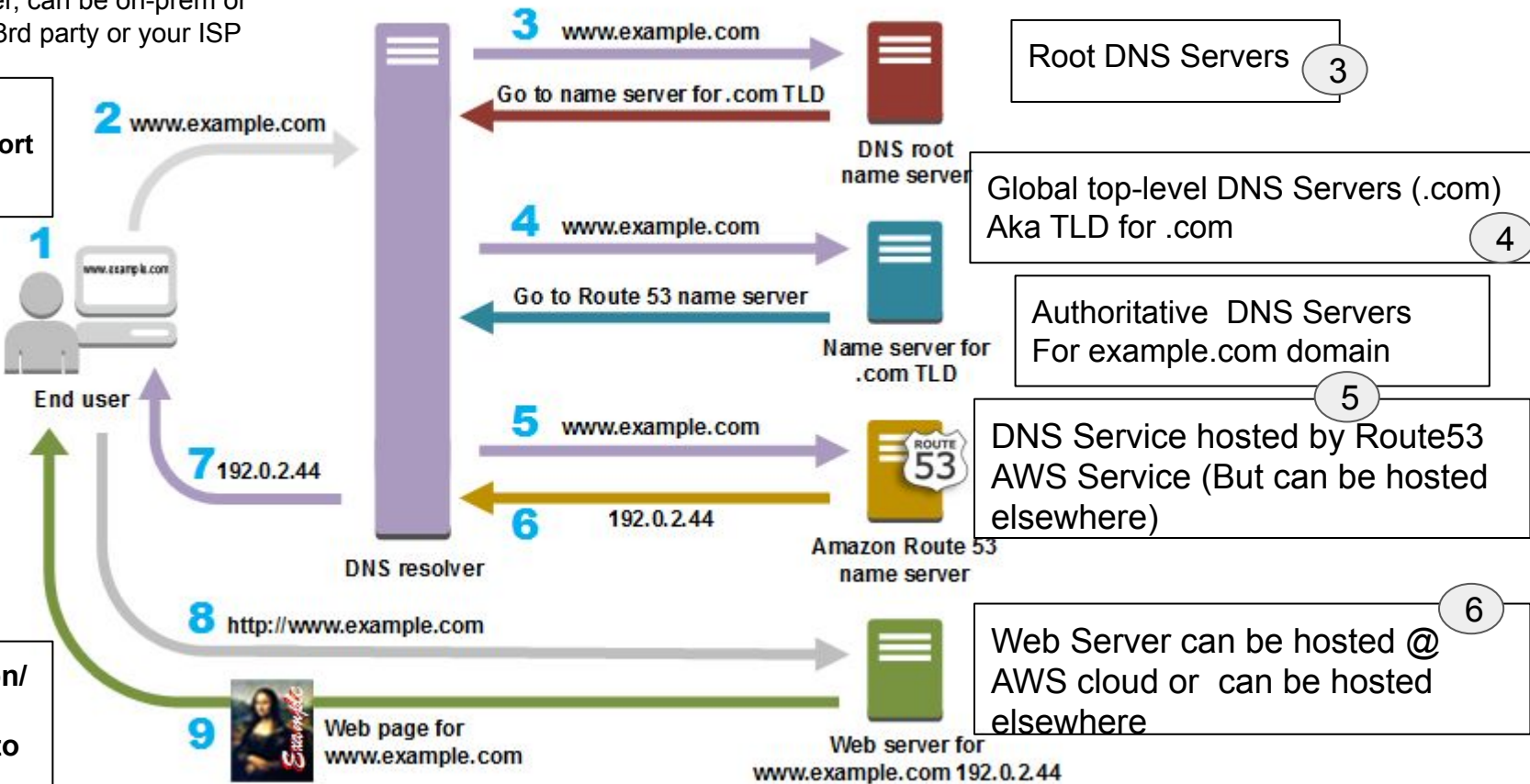
DNS Resolver, can be on-prem or
Provided by 3rd party or your ISP

Step 1-7
DNS=UDP Port
53 Query

DNS
Resolvers
are
recursive

Root/TLD
NS are
iterative

TCP session/
connection
HTTP/TLS to
the server

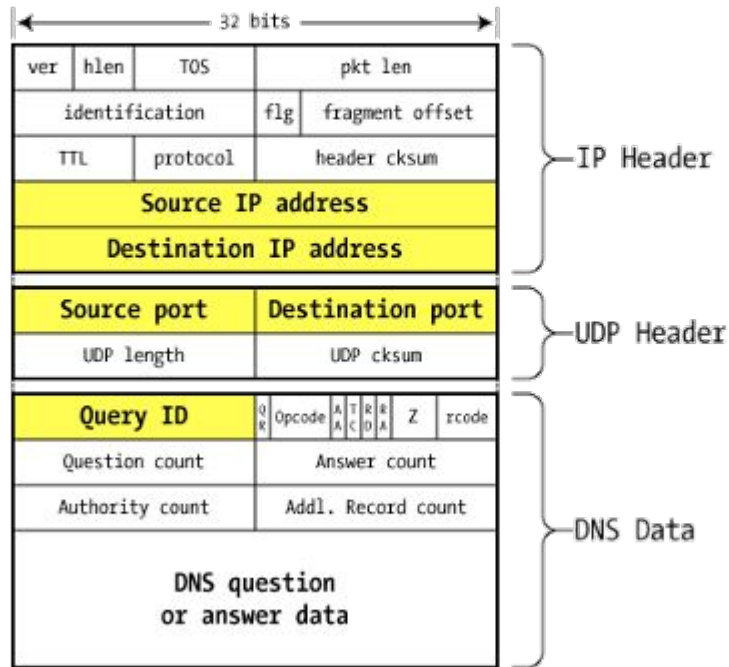


Source: <https://aws.amazon.com/route53/what-is-dns/>

Overview of DNS/Caching and Resolving Process

- A client/resolver asks for name to ip address mapping to a local DNS server
- If the server has the mapping already it provides, else it will find out the mapping using the distributed/hierarchical DNS system
 - Local DNS/caching servers will use "recursive query" to find out the mapping/answer by itself
- Reaches out to the "root" DNS servers (Top level domain), which are well known/pre-defined in DNS servers
- Root DNS servers "redirect" the caching DNS with "referrals" that will help caching DNS servers to "Authoritative" DNS for a given domain
 - This type of query/response is called as "Iterative query"
- Authoritative "Name Server(s)" are the one which can provide "Authentic/Authoritative" domain name to IP address mapping
- All these operations will result in local DNS/caching server getting the mapping of Domain name to IP address
- Local DNS/caching DNS server provides the mapping info to the requesting client

DNS Protocol packet layout (UDP)



DNS = UDP/Port53

DNS packet on the wire

Route53 DNS Service Overview

- Route53 is a Global AWS Services, spread across dozens of locations worldwide
- Uses Anycast Infrastructure, Promises 100 % Availability
- Offers unique routing feature functionality such as:
 - Simple routing, round-robin, failover, low latency based, Geo routing etc
- Integrates with other AWS Services
- Can be managed via AWS console, AWS CLI or SDK

How to register a DNS Domain?

- Register a Domain name with a DNS Registrar
- Could be in AWS/Route53 or 3rd party
- Create a Hosted zone*
- Create DNS “Records” **
- Delegate your DNS hosted zone “NS” to TLD*

* : If done in Route53, this is automatically taken care of

** : NS and SOA records will be automatically created with Route53

Route53 AWS Console/Dashboard

← → ↺

console.aws.amazon.com/route53/v2/home#GetStarted

★ Bookmarks

CRS-and-CRS-NG

CA-AS

Personal

CSR1kv-KVM

Music

🔍 > Plex It!

🔍 Image result for ru...

Imported

South Indian Bank

South Indian B

aws

Services ▾


🔍 Search for services, features, marketplace products, and docs [Option+S]


☰


Route 53 > Get started


Get started [Info](#)

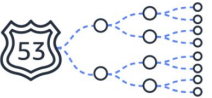
Choose your starting point


☒ **Register a domain**
Register the name, such as example.com, that your users use to access your application.


☐ **Transfer domain**
You can transfer domain names to Route 53 that you registered with another registrar.


☐ **Create hosted zones**
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.


☐ **Configure health checks**
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.


☐ **Configure traffic flow**
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.


☐ **Configure resolvers**
A regional service that lets you route DNS queries between your VPCs and your network.


Cancel

Get started

Route53 AWS Console/Dashboard



Services ▼

🔍 Search for services, features, marketplace products, and docs

[Option+S]

1: Domain Search

2: Contact Details

3: Verify & Purchase

Choose a domain name

.com - \$12.00 ▼

Check

Availability for 'muthuayyanar.com'

| Domain Name | | Status | Price /1 Year | Action |
|------------------|---|-----------|---------------|------------------------------|
| muthuayyanar.com | ✓ | Available | \$12.00 | <button>Add to cart</button> |

Related domain suggestions

| Domain Name | | Status | Price /1 Year | Action |
|-------------------|---|-----------|---------------|------------------------------|
| drmutuayyanar.com | ✓ | Available | \$12.00 | <button>Add to cart</button> |
| krupaayyanar.com | ✓ | Available | \$12.00 | <button>Add to cart</button> |

Use below URL to get additional details on information to be provided to complete “domain name registration” using Route53
<https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-register-values-specify.html>

What's unique about Route53 DNS Service?


- One stop shop to do Domain name registration , delegation and zones are created automatically
- Route53 performs health checks on your resources and is Integrated with other AWS Services
- Offers variety of advanced routing feature
- AWS guarantees 100% availability

Route53 DNS Service Routing Policy

- **Simple routing policy** – Use to route internet traffic to a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.
- **Failover routing policy** – Use when you want to configure active-passive failover.
- **Geolocation routing policy** – Use when you want to route internet traffic to your resources based on the location of your users.
- **Geoproximity routing policy** – Use when you want to route traffic based on the location of your resources and, optionally, shift traffic from resources in one location to resources in another.
- **Latency routing policy** – Use when you have resources in multiple locations and you want to route traffic to the resource that provides the best latency.
- **Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.
- **Weighted routing policy** – Use to route traffic to multiple resources in proportions that you specify.

Source: <https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/route-53-concepts.html>

Route53 Routing Policy

 Services ▾

Search for services, features, marketplace products, and docs [Option+S]

vocstartsoft/user934475=muthuraja.ayyanar@sjsu.edu @ 4079-222... ▾ Global ▾ Support ▾

Route 53

Dashboard

Hosted zones

Health checks

▼ Traffic flow

Traffic policies

Policy records

▼ Domains

Registered domains

Pending requests

▼ Resolver

VPCs

Inbound endpoints

Outbound endpoints

Rules

Query logging

Switch to old console

 **Route 53 couldn't update the page**
Route 53 encountered an unknown error and couldn't update your page. Try refreshing the page.

Refresh page

Route 53 Dashboard [Info](#)

DNS management

A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

Create hosted zone

Traffic management

A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

Create policy

Availability monitoring

Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

Create health check

Domain registration

Error

Domains

Register domain

Find and register an available domain, or [transfer your existing domains](#) to Route 53.

Enter a domain name

Check

Each label (each part between dots) can be up to 63 characters long and must start with a-z or 0-9. Maximum length: 255 characters, including dots. Valid characters: a-z, 0-9, and - (hyphen)

Notifications

Find notifications

< 1 >

| Resource | Status | Last update |
|-----------------------------|--------|-------------|
| No notifications to display | | |

More resources

Documentation

API reference

FAQs

Forum - DNS and health checks

Forum - Domain name registration

Request a limit increase

Service health

To view the current status of Route 53, see the [AWS Service Health Dashboard](#).

Route53 Create Hosted Zone

Services ▾

Route 53

×

Dashboard

Hosted zones

Health checks

▼ Traffic flow

Traffic policies

Policy records

▼ Domains

Registered domains

Pending requests

▼ Resolver

VPCs

Inbound endpoints

Outbound endpoints

Rules

Query logging

Switch to old console

Route 53 > Hosted zones > Create hosted zone

Create hosted zone

info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name

info

This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! * # \$ % & ' () ^ + , - / : ; < = > ? @ [\] ^ _ ` { | } . ~

Description - optional

info

This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 0/256

Type

info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☒ Public hosted zone

A public hosted zone determines how traffic is routed on the internet.

☐ Private hosted zone

A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags

info

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

Add tag

You can add up to 50 more tags.

Cancel

Create hosted zone

Route 53 > Hosted zones > Create hosted zone

Create hosted zone

info

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name

info

This is the name of the domain that you want to route traffic for.

Valid characters: a-z, 0-9, ! * # \$ % & ' () ^ + , - / : ; < = > ? @ [\] ^ _ ` { | } . ~

Description - optional

info

This value lets you distinguish hosted zones that have the same name.

The description can have up to 256 characters. 25/256

Type

info

The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

☒ Public hosted zone

A public hosted zone determines how traffic is routed on the internet.

☐ Private hosted zone

A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags

info

Apply tags to hosted zones to help organize and identify them.

Key

×

Value - optional

×

Remove tag

Add tag

Custom tag value

You can add up to 49 more tags.

Cancel

Create hosted zone

Route53 Created Hosted Zone

✔ muthurajaayyanar.com was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

[Route 53](#) > [Hosted zones](#) > muthurajaayyanar.com

muthurajaayyanar.com [Info](#)

Delete zone

Test record

Configure query logging

► Hosted zone details

Edit

Records (2)

DNSSEC signing

Hosted zone tags (1)

Records (2) [Info](#)

Automatic mode is the current search behavior optimized for best filter results. [To change modes go to settings.](#)



Edit

Delete record

Import zone file

Create record

🔍 Filter records by property or value

Type ▼

Routing policy ▼

Alias ▼

< 1 > ⚙️

| <input type="checkbox"/> | Record name ▼ | Type ▼ | Routing policy ▼ | Differe ntiator ▼ | Value/Route traffic to ▼ |
|--------------------------|----------------------|--------|------------------|----------------------|--|
| <input type="checkbox"/> | muthurajaayyanar.com | NS | Simple | - | ns-778.awsdns-33.net. ns-320.awsdns-40.com. ns-1909.awsdns-46.co.uk. ns-1364.awsdns-42.org. |
| <input type="checkbox"/> | muthurajaayyanar.com | SOA | Simple | - | ns-778.awsdns-33.net. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400 |

Route53 Create Record

Route 53 > Hosted zones > muthurajaayyanar.com > Create record

Quick create record [Info](#)

[Switch to wizard](#) [Add another record](#)

▼ Record 1

Routing policy [Info](#)

Simple routing

Record name [Info](#)

blog.muthurajaayyanar.com

Valid characters: a-z, 0-9, !, " # \$ % & ' () * +, - / : ; < = > ? @ [\] ^ _ { | } . - ~

Alias

☐

Record type [Info](#)

A – Routes traffic to an IPv4 address and so...

Value [Info](#)

192.0.2.235

TTL (seconds) [Info](#)

300 1m 1h 1d

Recommended values: 60 to 172800 (two days)

Enter multiple values on separate lines.

Cancel

Create records

► View existing records

The following table lists the existing records in muthurajaayyanar.com.

Choose routing policy [Info](#)

The routing policy determines how Amazon Route 53 responds to queries.

Routing policy

[Switch to quick create](#)

☒ Simple routing

Use if you're routing traffic to just one resource, such as a webserver.

☐ Weighted

Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

☐ Geolocation

Use when you want to route traffic based on the location of your users.

☐ Latency

Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

☐ Failover

Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

☐ Multivalue answer

Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

Simple Route53 DNS Routing Policy



DNS Simple and TTL based Routing

- Simple Routing is used when a resource record maps to a single resource
- Each record has a TTL and DNS clients caches the record for the duration of TTL
- Upon Expiry of TTL, DNS CLient will try to update it's cache by sending another DNS query
 - Create two EC2 instance in two different AZ (EC2-1a & EC2-1f)
 - Create a record set (ttl-test) in the Route53 hosted zone of a domain (say test.com)
 - Make sure to use public IP address of EC2-1a for ttl-test.test.com
 - Check using nslookup/dig to see the name to ip address mapping and via the browser (also check the TTL of the record in dig)
 - Now change the ip address for ttl-test.test.com record with EC2-1f's IP
 - Route53 will update the record after the TTL expires

Route53 Alias vs CNAME based Routing

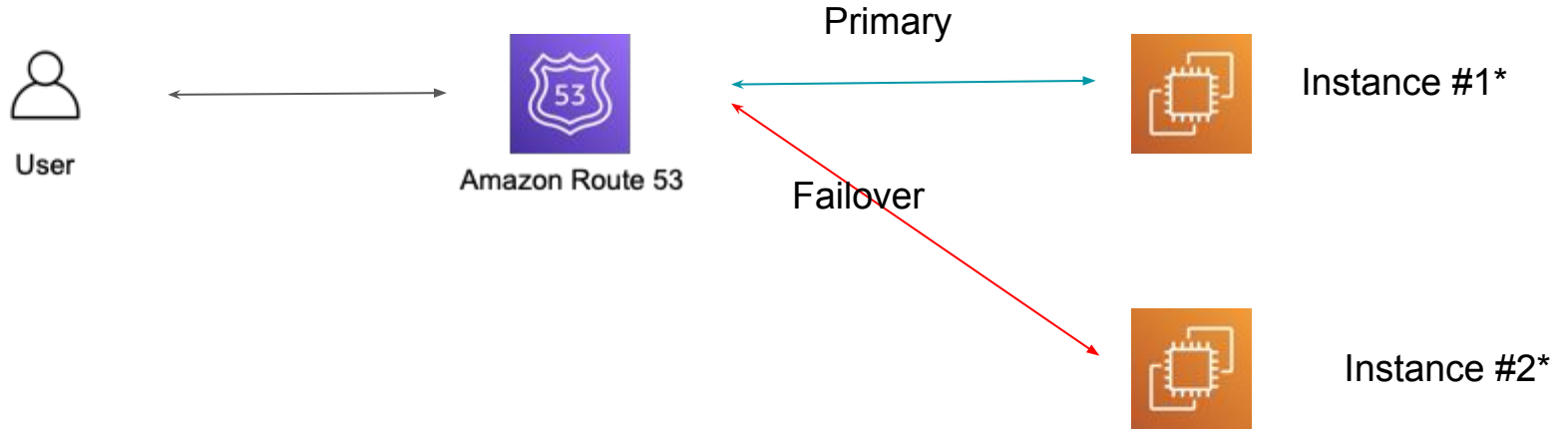
- “Alias” is AWS Route53 specific extension
- Maps record set to specific/select AWS service domain name such as ELB/S3/Cloudfront etc
- Alias allows mapping of recordset at the “zone apex”
- If you have a domain name “test.com”, Alias recordset can be used to map an AWS resources/service to “test.com”
- There are no charges whatsoever for using “Alias” recordset type
- Health checks can be configured for Alias recordset type
- All DNS implementation supports CNAME
- Map record set to a canonical name
- CNAME does not allow recordset to be mapped to “zone apex” using CNAME type of record
- That’s using CNAME recordset type only sub-level domain names can be mapped, say to “www-internal.test.com” to www.test.com
- As usual a very nominal charge will be incurred by customer for using CNAME (for usage over and beyond what AWS offers for free)
- CNAME recordset type does not support health checks (But the mapped resource can have health check)

Route53 Alias vs CNAME based Routing

- Run an EC2 instance and confirm it's running a web server
 - Check to see if EC2 instance can be accessed using it's public IP or AWS host/domain name
 - Create ALB and put the EC2 behind ALB
 - Create a recordset "cname-test" of type "CNAME" under "test.com" hosted zone
 - Use the ALB hostname as the "value" for this record (not the IP address"
 - Check and confirm that you can access the EC2 instance using CNAME alias record
- Use the same EC2 instance that is used for CNAME
 - We will now create a new recordset, give it a name "alias-test" ie.s alias-test.test.com
 - Leave the type as "IP address" for this recordset but we will select the "Yes" radio button next to the field "Alias" (by default this is "No")
 - Next choose the "Alias Target" using the pull down arrow to point to the ALB that was created
 - Confirm the EC2 target is accessible using the "Alias" record type
 - There is no charge for querying/using alias recordset type

Failover Route53 DNS Routing Policy

Amazon EC2 Instance or other
AWS Resource/Service or
Or Resource outside of AWS



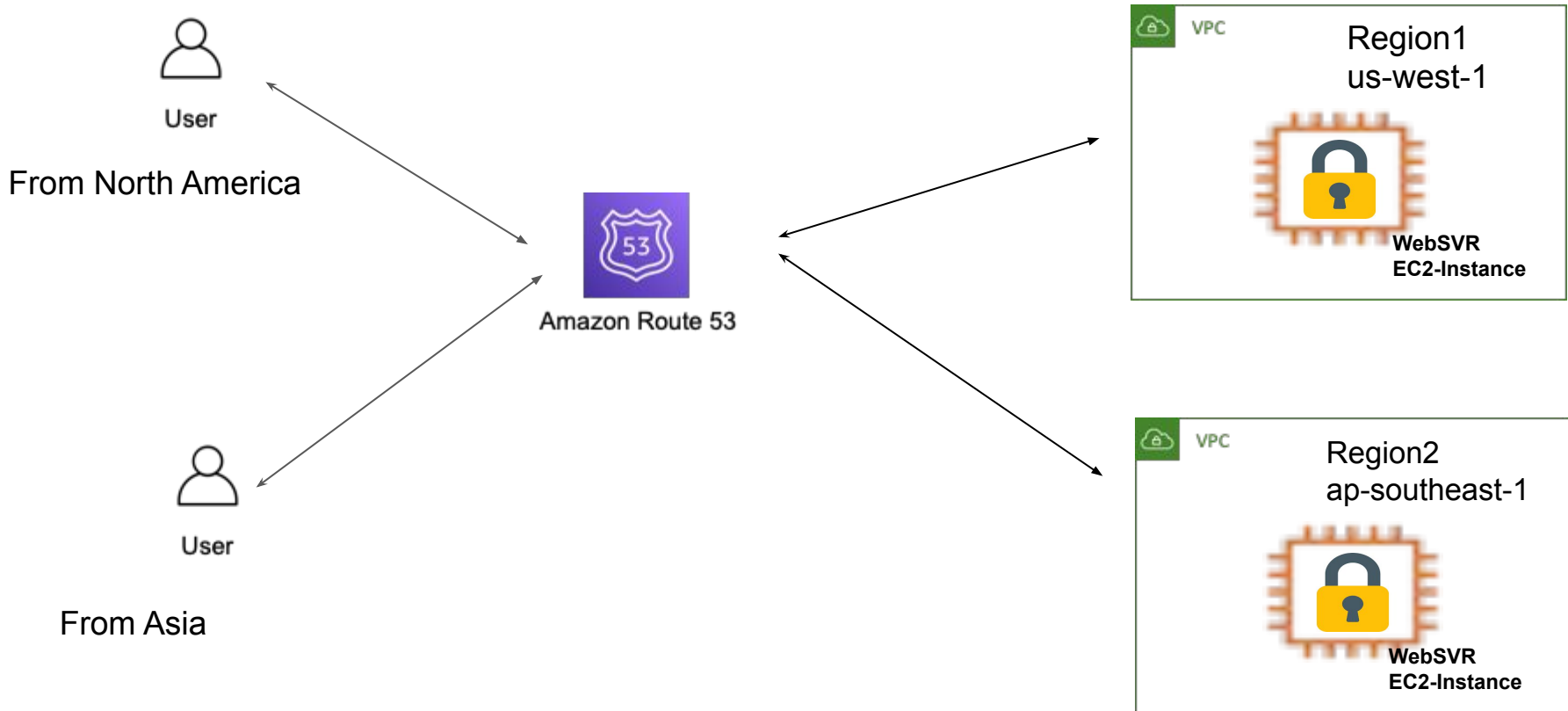
Route53 depends on Health checks for Failover based DNS routing , hence it's mandatory to set up health check to be able to use this feature

*: Instances could be in same or Different Regions

Route53 Failover Routing

- Setup a primary EC2 instance in region #1 and confirm it's running a web server
 - Check to see if EC2 instance can be accessed using it's public IP or AWS host/domain name
 - Idea is to setup Route53 routing such that primary instance will be used all the time and only use secondary if primary becomes unhealthy/unavailable
 - Create a health check @ Route53
 - Create a Pri recordset "failover-test" of type "IPv4" under "test.com" hosted zone
 - Associate health check created for this EC2 instance (Mandatory)
 - Check and confirm that you can access the EC2 instance using failover-test.test.com
- Setup a second EC2 instance in region #2 and confirm it's running a web server
 - Check to see if EC2 instance can be accessed using it's public IP or AWS host/domain name
 - Create a health check @ Route53 (not needed for secondary, but as a best practice create)
 - Create a Sec recordset "failover-test" of type "IPv4" under "test.com" hosted zone
 - Associate health check created for this EC2 instance (optional)
 - Create a failure condition for primary EC2 instance
 - Check and confirm that you can access the EC2 instance using failover-test.test.com

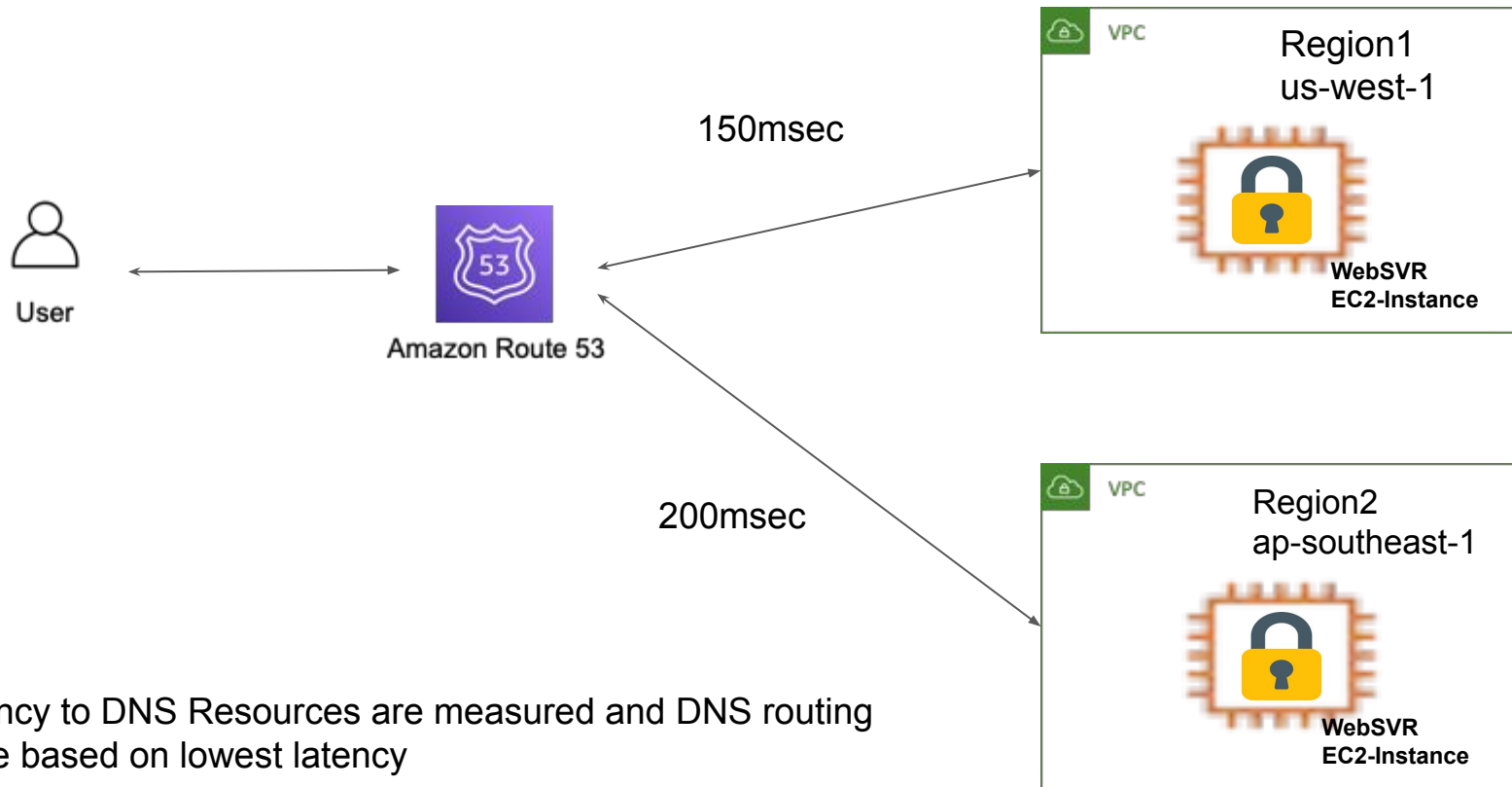
Geolocation Based Route53 DNS Routing Policy



Route53 Geolocation based Routing

- Route request based on where the DNS query is originating from
- That is, Route53 routes DNS query from users to AWS resources that are close to the user geographical locations
- **Used to provide localized content, content that are prohibited in some geography or for compliance reasons**
- When defining DNS recordset, choose routing type as “geolocation” based, record type is still “ipv4”
- Make sure to select the appropriate Geo location (can be selected down to state)
- Also create a record with “default” geo location for users who originated DNS query from other geo locations
- Setup three EC2 instance in various regions and confirm they are running a web server
- Check to see if those EC2 instances can be accessed using it's public IP or AWS host/domain name
- Create three recordset (with same name) “geolocation-test” of type “IPv4” under “test.com” hosted zone
- Make sure to select appropriate geolocation of the EC2 instance
- Ensure a default geolocation based record is there as a catchall record for other locations
- Check and confirm that you can access the EC2 instance using geolocation-test.test.com record from different geo location
- Use vpn to emulate users connecting from

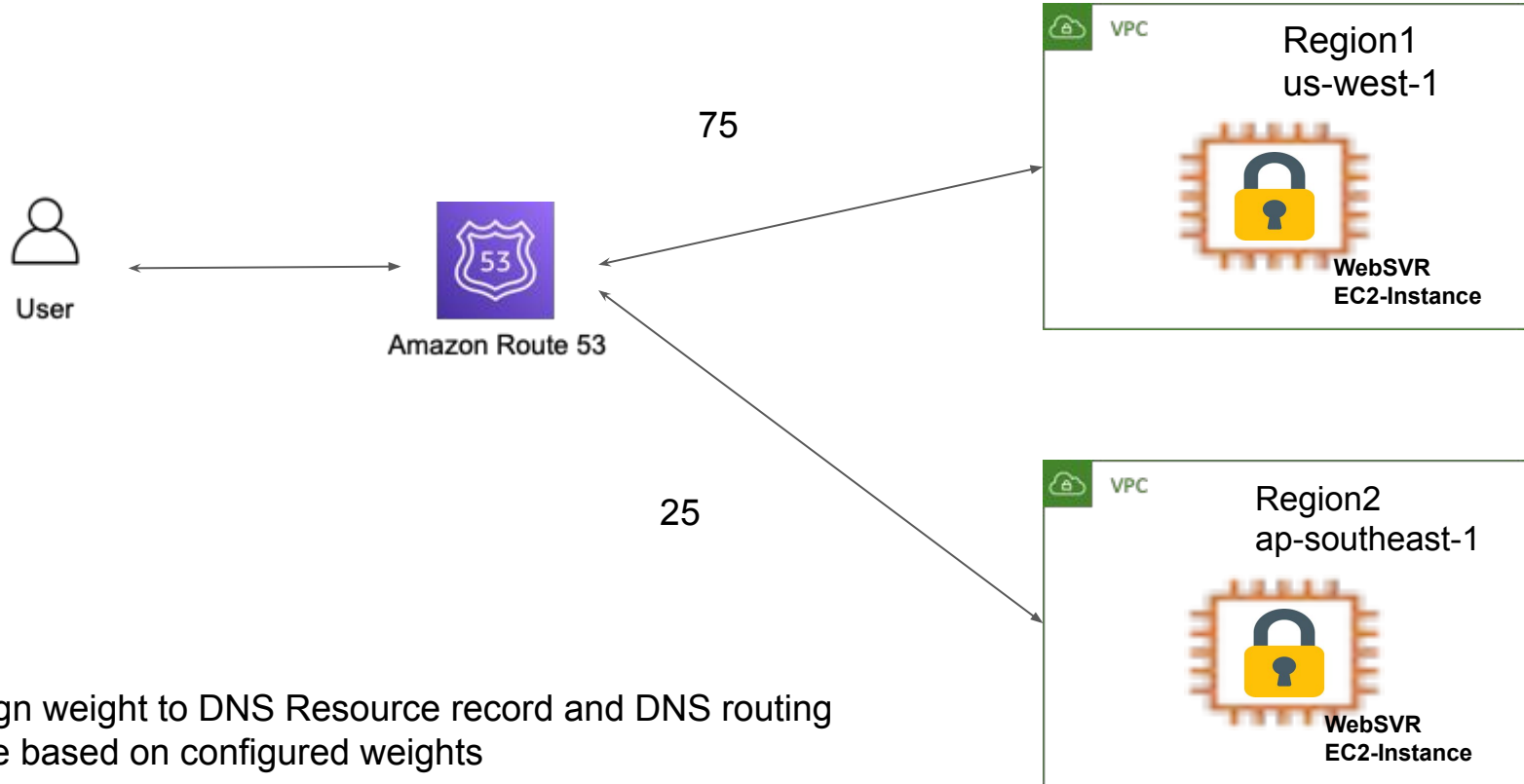
Latency Based Route53 DNS Routing Policy



Route53 Latency based Routing

- Scenario/Use case: Your application/web servers are deployed in various AWS Regions
 - For obvious reasons, you want to serve your users with application from a region that provides the best performance (Network latency based)
- Presume you have setup multiple recordset for your various resources across geo location/regions
- AWS will start to test/tabulate the latency across all AWS global infra for those resources
- Based on a given user DNS query a Route53 server fields, it will then route the request to a resource that has lowest latency
- How does this work? Assume you have an ELB/ALB ins US-West Region along with a target and another ELB/ALB in Singapore region with it's target
- Say, your users are in UK/London region, when they try to access your ELB/ALB (using the domain name recordset, say a common name Mapped to different ELB/ALB's ip address)
- Choose "latency" based routing type while creating recordset, AWS automatically selects the region based on the ip address (ipv4 type)
- Route53 will check to see the network latency from UK/London to US west region and to UK/London to Singapore region
- Then picks the site that shows the lowest network latency

Weighted Round Robin Route53 DNS Routing Policy



AWS AMI User Data

```
#!/bin/bash
#Install Apache/httpd Web Server
yum update -y
sudo yum install jq -y
yum install -y httpd.x86_64
systemctl start httpd.service
systemctl enable httpd.service
echo "Web Server is running on $(hostname -f)" > /var/www/html/index.html
echo " in region " ; curl -s http://169.254.169.254/latest/dynamic/instance-identity/document | jq -r .region >> /var/www/html/index.html
echo " on AZ" ; curl -s http://169.254.169.254/latest/dynamic/instance-identity/document | jq -r .availabilityZone >> /var/www/html/index.html
echo "Web Server is running using Public IP" ; curl -s http://169.254.169.254/latest/meta-data/public-ipv4 >> /var/www/html/index.html
```