ROUTE53

* Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking
* Amazon Route 53 is an authoritative Domain Name System (DNS) service.
* DNS is the system that translates human-readable domain names (example.com) into IP addresses (192.0.2.0). With authoritative name servers in data centres all over the world, Route 53 is reliable, scalable, and fast.
* **DNS Keep track of all hostnames and IP address convert IP to Host and Host to IP**
* **DNS all about records only**
* **Name Server: Which identify the domain name is called "Name Server"**
* **Route53 is Global because**

So, it is responsible to convert IP address to name, name to IP address.

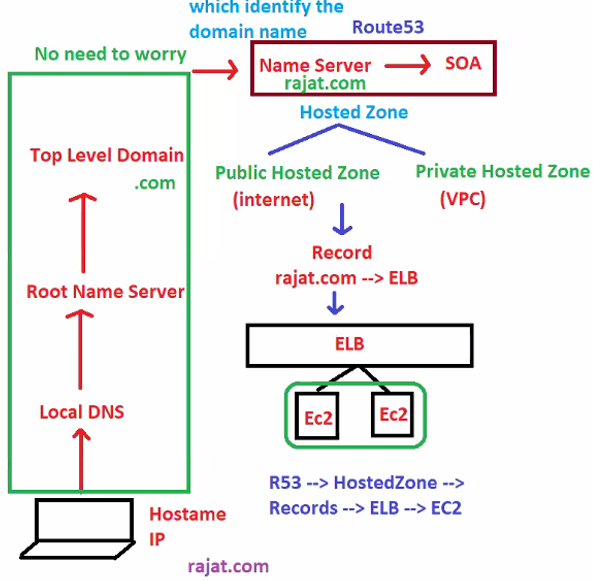
when we use **"google"** in browser address bar, it will be converted to IP Address.

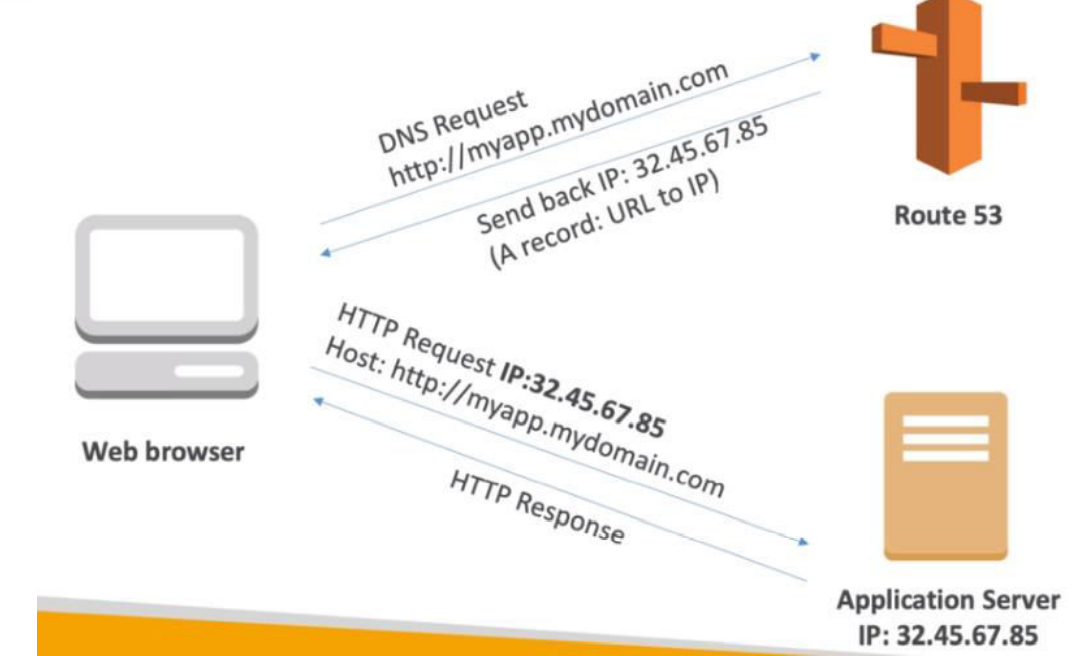
When we launch EC2 machine, we get public IP and private IP, we also get DNS name.

DNS Server is responsible to assign DNS name to the machine.

**Note**: 53 is DNS port number.

* Route53 is a Managed DNS (Domain Name System)
* DNS is a collection of rules and records which helps clients understand how to reach a server through URLs. In AWS, the most common records are:
* **A** : URL to IPv4
* **AAAA** : URL to IPv6
* **CNAME** : URL to URL
* **Alias** : URL to AWS resource

****



* **Container of records is called "Hosted Zone"**
* **Name of the Hosted Zone nothing domain name**
* **i.e rajat.com (in our example)**
* **Hosted Zones contains "RECORDS"**
* When we type rajat.com definitely a record might have created

**Flow will be like this**

**Rajat.com--> Route53 --> Hosted Zone --> Records --> ELB --> EC2**

* whenever we create public hosted zone they will be 2 records will be created automatically

**1.NS records**

* Pool of servers

**2.SOA**

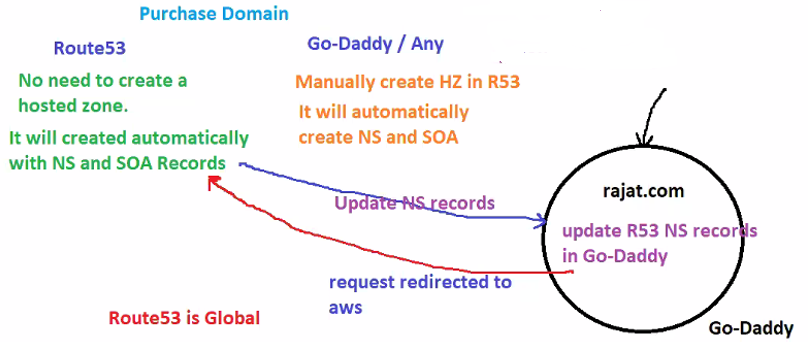
* Admin for hosted zone

NS and SOA (default records) records are automatically created and handled by AWS

**We should not delete this default records**

**Two ways we can purchase the domain**

* we can purchase **ROUTE53** or **Go-Daddy/Any were**
* **Note**: if we purchase domain name in ROUTE53, **no need to create hosted zone it will create automatically**
* if we purchase domain name in **Go-Daddy/any other website**, in this case we have to create hosted zone manually **in ROUTE53**
* **if we purchase domain name in Go-Daddy in this case, if any one hit rajat.com website.**
* **request will go to Go-Daddy/3rd party site, in order to redirect the request to ROUTE53 we need to update ROUT53 NS records in Go-Daddy if we update then request will redirect to aws**

****

**Route53 can use**

* public domain names own (or buy)
* private domain names that can be resolved by your instances in your VPCs

**Route53 has advanced features such as**

* Load balancing (through DNS-also called client load balancing)
* Health checks
* Routing policy

You pay $0.50 per month per hosted zone

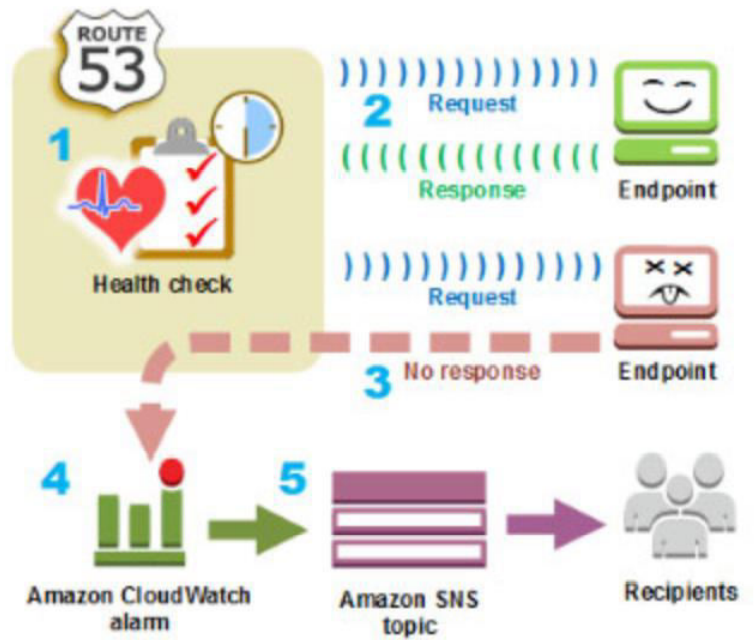
A hosted zone is a container for records, and records contain information about how you want to route traffic for a specific domain, such as example.com, and its subdomains (apex.example.com, acme.example.com). A hosted zone and the corresponding domain have the same name.

There are two types of hosted zones:

* **Public hosted zones** contain records that specify how you want to route traffic on the internet
* **Private hosted zones** contain records that specify how you want to route traffic in an Amazon VPC

**Health Checks**

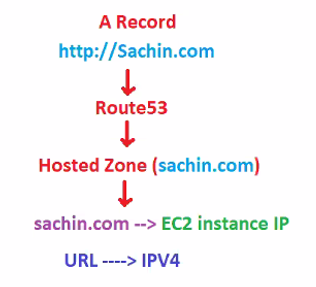
* You can set health checks for individual records sets.
* **Can have HTTP, TCP and HTTPS health checks (no SSL verification)**
* If a record set fails a health check it will be removed from route53 until it passes the health check
* You can set SNS notifications to alert you if a health check is failed



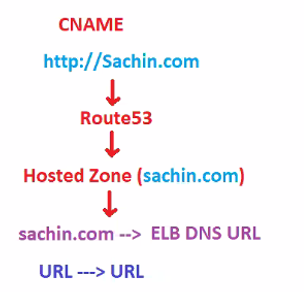
**Supported DNS Records Types**

* **A Record Type**: The value for an **A record is an IP4** address in dotted decimal notation

if someone type **sachin.com** flow will be like this

****

* **AAAA Record Type**: The value for an AAAA **record is an IPV6**
* CAA Record Type: lets you specify which certificate Authority are allowed to issue certis for a domain
* **CNAME Record Type:** The value element is the same as the domain name

****

* **CNAME records are chargeable and alias records are free**
* Sachin.com 🡪 is also called Main / Naked domain / Zone apex records
* **a.sachin.com 🡪 Sub-Domains**
* For Naked Domain you cannot use CNAME records instead of use **ALIAS records**
* Due to this reason, always choose **ALIAS over CNAME**
* In Real time we will combination of A record and ALIAS records
* **MX Record Type**
* **NAPTR Record Type**
* **NS Record Type**: An NS record identifies the name servers for the hosted zone. The value for an NS record is the domain name of a name server.
* **PTR Record Type**: is same format as domain name.
* **SOA Record Type:** A start of authority (SOA) record provides information about a domain and the corresponding Amazon Route 53 hosted zone.
* SPF Record Type
* SRV Record Type
* TXT Record Type

**Start Of Authority (SOA) record**

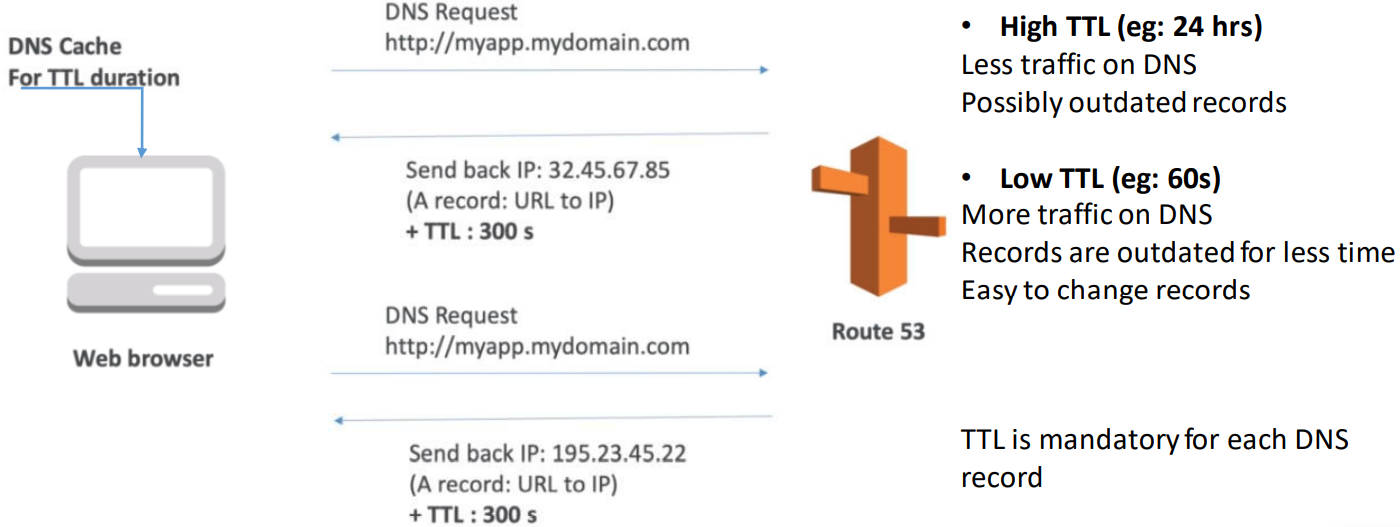
* The name of the server that supplied the data for the zone.
* The administrator of the zone
* The current version of the data file

**NS stands for Name Server Record**

* They are used by Top level Domain servers to direct traffic to the content DNS server which contains the authoritative DNS record

**What is TTL ?**

* The length that a DNS record is cached on either the resolving server or the users own local PC is equal to the value of the TTL in seconds.
* The lower the TTL, the faster changes to DNS records take to propagate throughout the internet.
* The web browser will cache the DNS record for 300s, once expired it will check with DNS

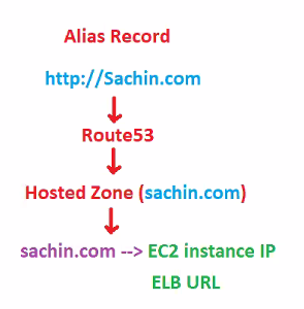


**What is CNAME ?**

* The CNAME can be used to resolve one domain name to another.
* ELB to domain name

**Alias Record**

* Alias records are used to map resources record sets in your hosted zone to Elastic Load Balancers, CloudFront distributions, or S3 buckets that are configured as websites.
* Alias records work like a CNAME record in that you can map one DNS NAME(example.com) to another target DNS name(myelb12.elb.amazonaws.com)



**Key Difference:** A CNAME cant be used for naked domain names (zone apex record).

You can’t have CNAME for http://cloudrsh.com, it must be either an A record or Alias.

Given choice, always choose an Alias record over CNAME.CNAME are charged, Alias are not

**Scenario**

Multiple webservers are connected to Load Balancer. Load balancer DNS name is not user friendly.

* By using route53, we buy the domain name.
* We connect the domain name to load balancer.
* We provide route 53 user friendly name domain name to public.

**2nd Advantage:**

**Generally, webserver and Load balancer will be in same availability zone.**

**If Availability zone is lost?**

* We can create load balancer and attach webserver present multiple availability zones. If one availability zone is lost, LB will direct to another availability zone.

So, load balancer handles webserver failure and availability zone failure.

**If Region is lost?**

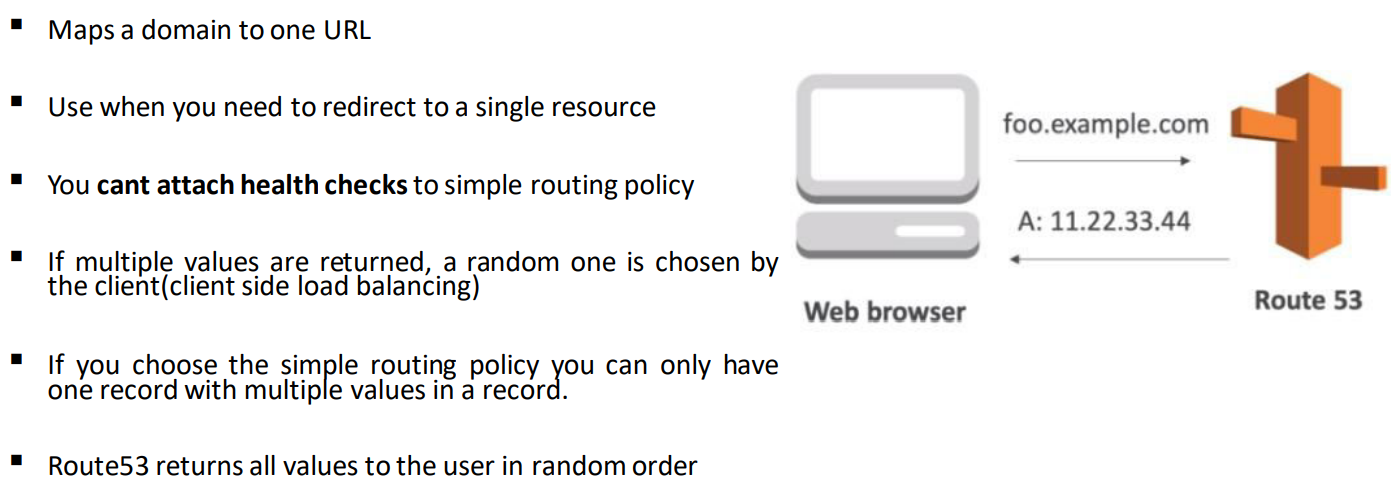
* We cannot create load balancer outside region. In this create we use Route53.
* **We connect multiple regions to Route 53**
* **Note: Route 53 helps from regional failures.**
* How Route53 will distribute the traffic?
* 50% - 50%
* 20% - 80%
* Main - failover
* It depends on routing policy.

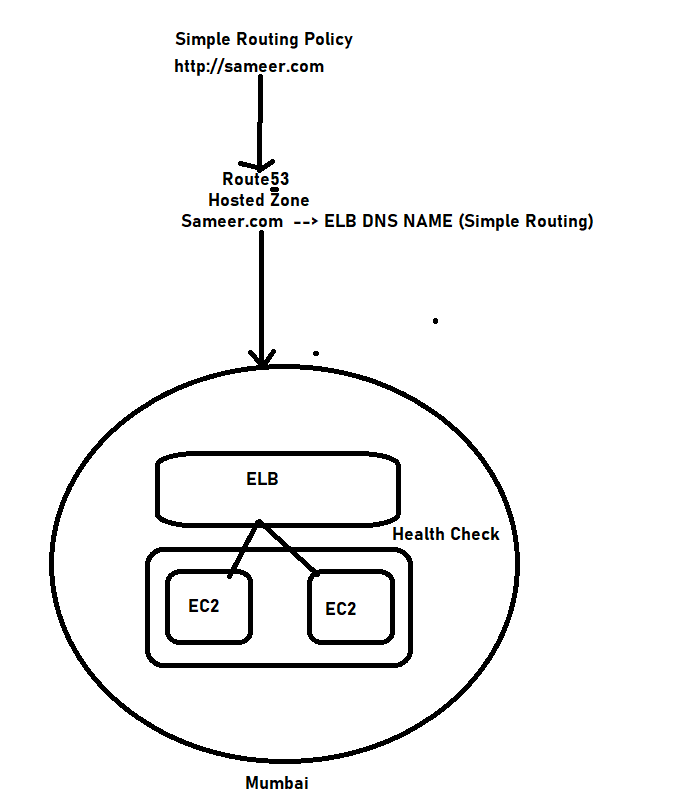
**We have 5 routing policies**

* Simple
* Weighted
* Latency
* Failover
* Geolocation

**1. simple**

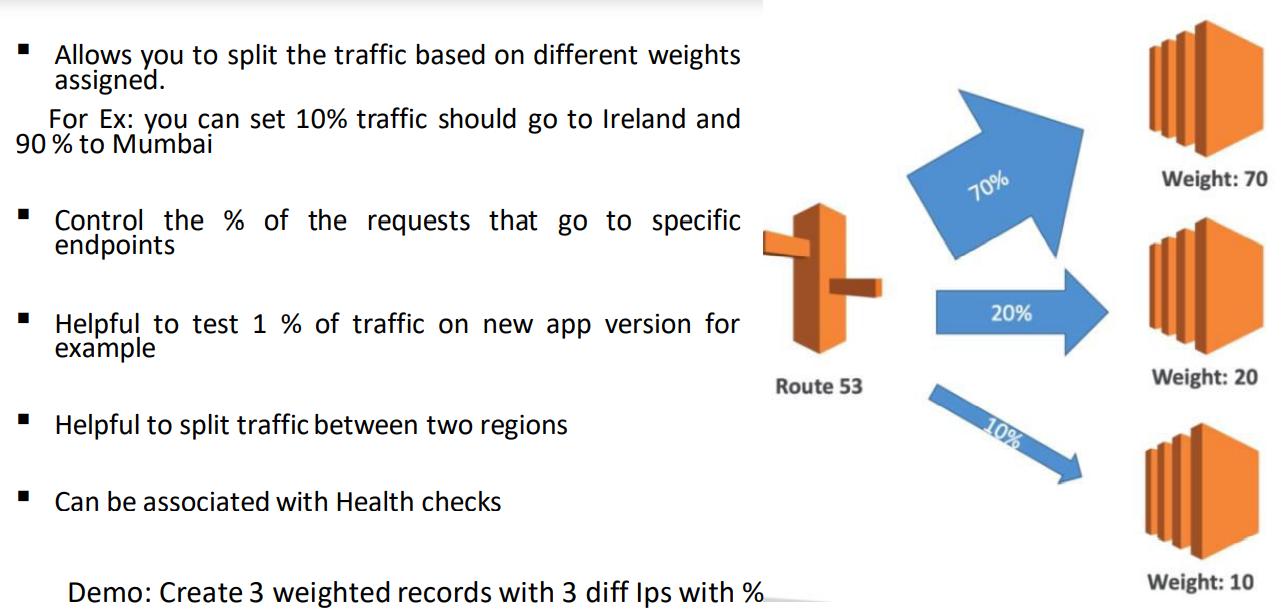
* This is default routing policy.
* Using simple routing policy, we get the **1st advantage**.
* This is all about providing Route 53 user friendly domain name to public.
* We cannot get the 2nd advantage
* We can connect only one region.
* Use for a single resource that performs a given function for your domain, for example, a web server that serves content for the example.com website.



****

**Weighted**

* In this policy, we connect two regions.
* Based on the weights we assign; load will be distributed.
* Weight can be
* 80 - 20 (80% traffic to region 1, 20% to region 2)
* 60 - 40
* Use to route traffic to multiple resources in proportions that you specify



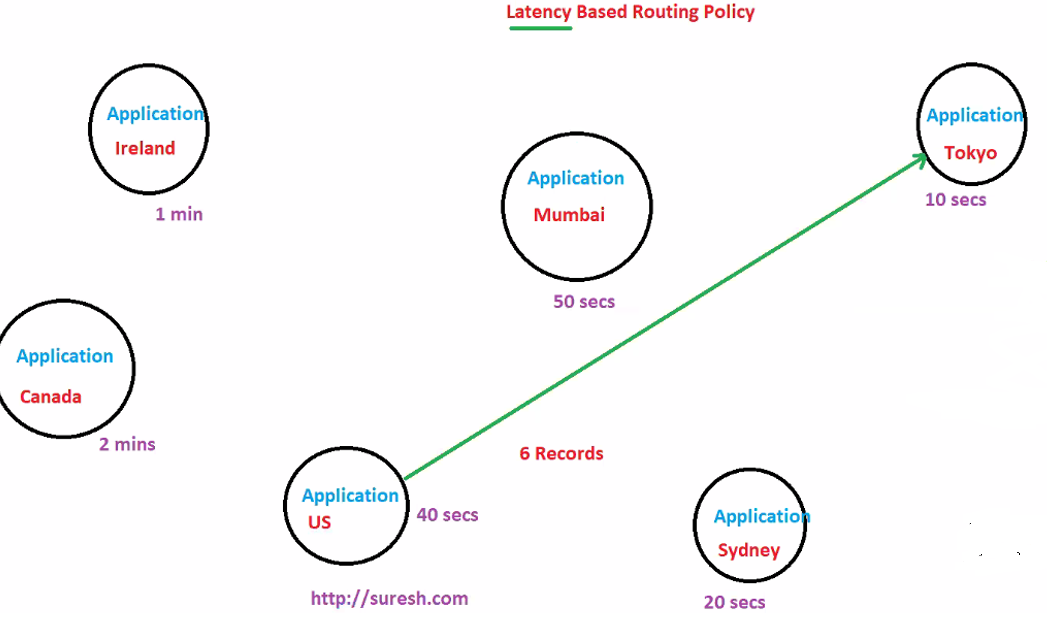
**Latency**

The term "Latency" refers to delay due to network traffic.

* The policy continuously monitor network traffic ( latency) in multiple regions by sending ping requests.
* Diverts the traffic to the region which has less latency.
* Use when you have resources in multiple AWS Regions and you want to route traffic to the region that provides the best latency.
* Allows you to route your traffic based on the lowest network latency for your end user. (ie, which region will give them the fastest response time).
* To use latency-based routing, you create a latency resource record set for the EC2 or ELB resource in each region that hosts your website.
* When Amazon Route53 receives a query for your site, it selects the latency resource record set for the region that gives the user the lowest latency.
* Route 53 then responds with the value associated with that resource record set.
* Demo: Create 3 diff records with latency policy with 3 diff Ips and use VPN
* I have my application in each and every region,if user accessing suresh.com from US which location get connected

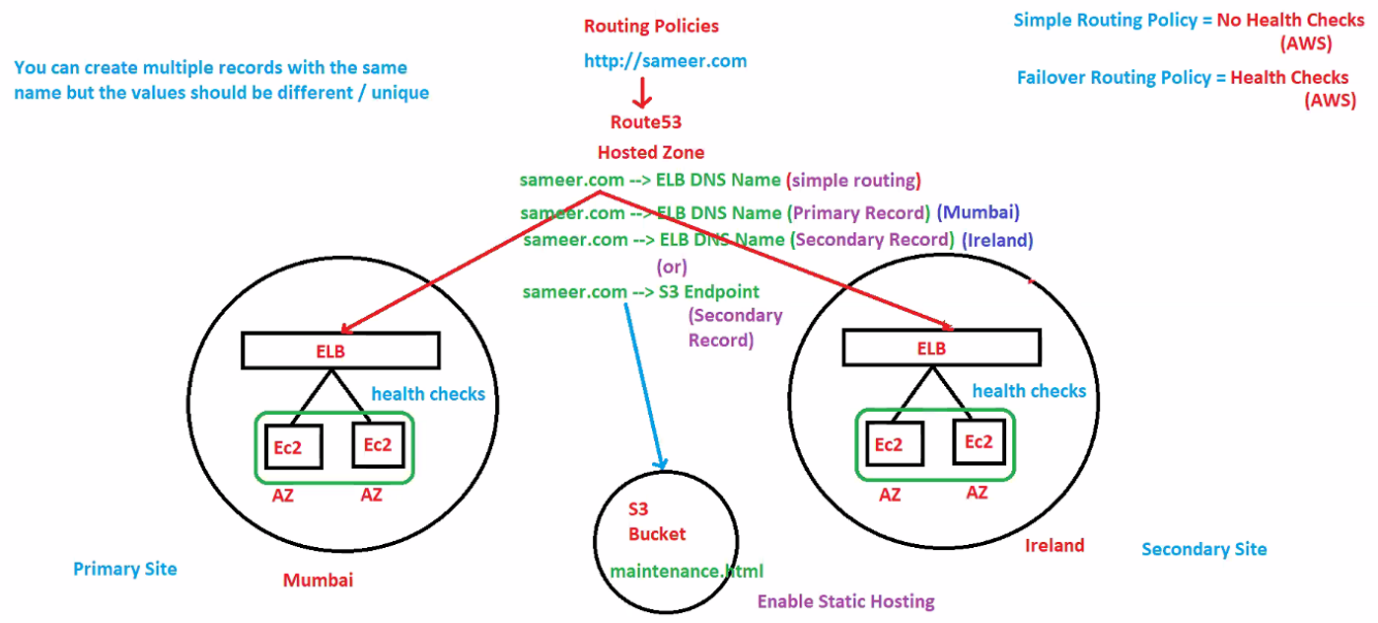
**right Ans is Tokyo**

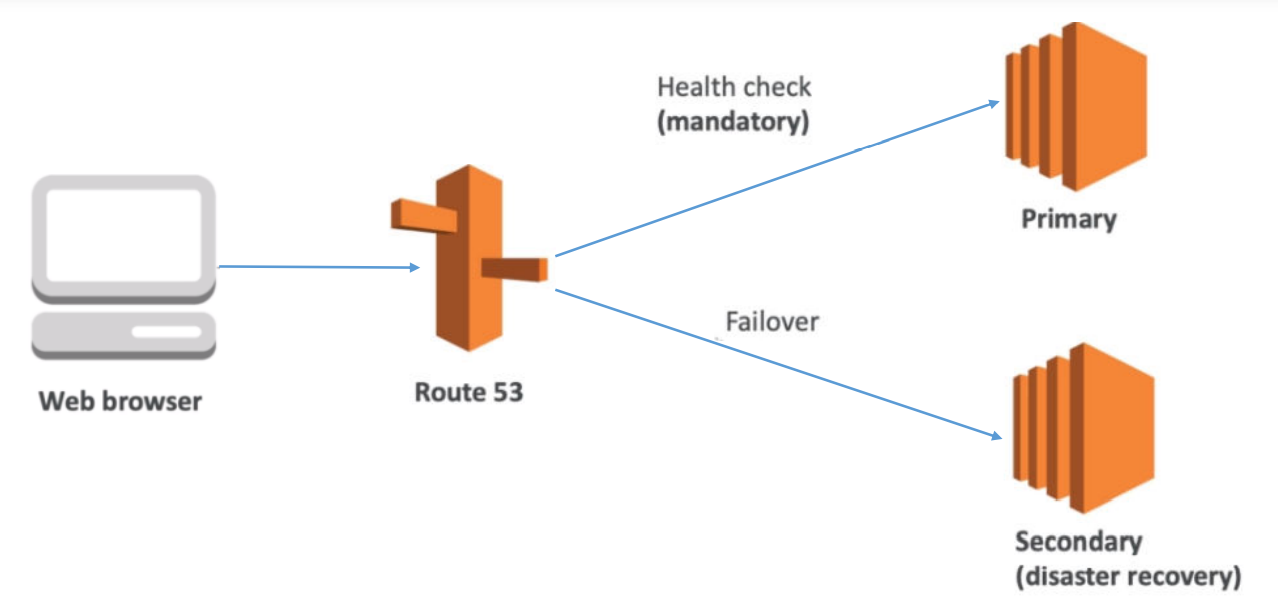
* whichever region has low latency then request will redirect to particular region.



**Failover**

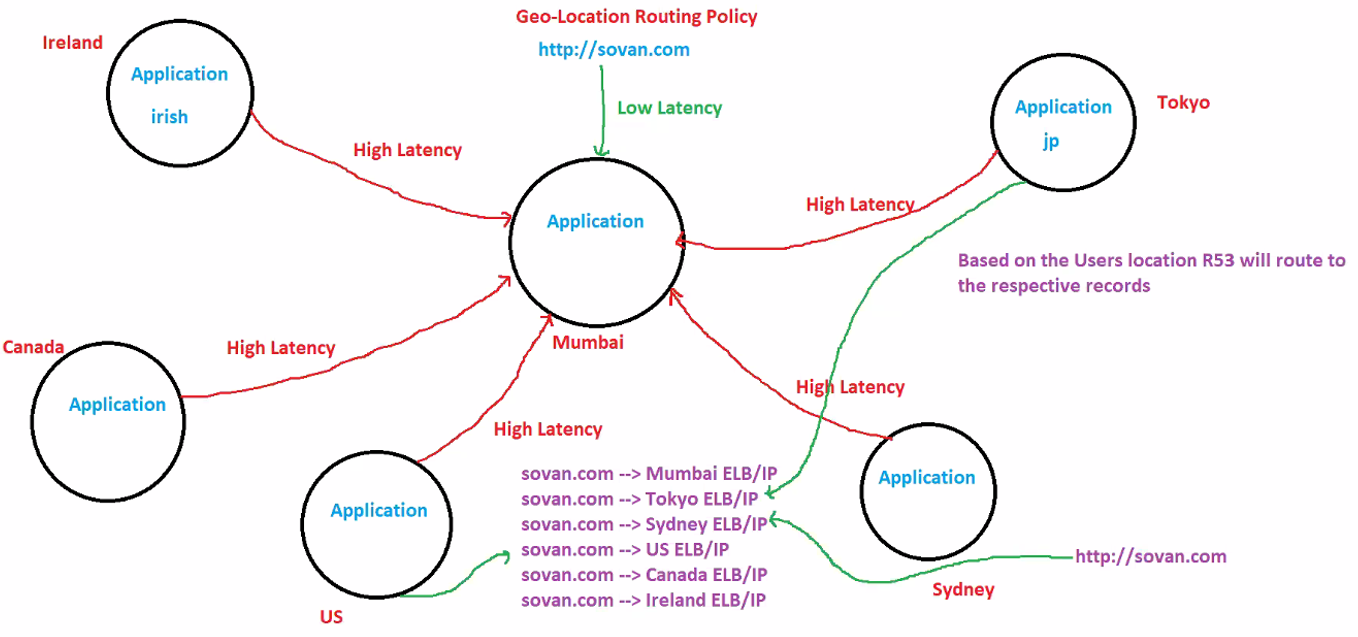
* We will define one region as "Main" (Active)
* another region as “Standby" (Passive)
* Use when you want to configure active-passive failover.
* Route 53 will divert all traffic to Main region.
* In case "Main" region is lost, the Route 53 will divert the traffic to "Standby"
* Route 53 will monitor the health of load balancer continuously.
* Failover routing policies are used when you want to create an active/passive setup.
* For Ex: you may want your primary site to be in Mumbai and DR site in Ireland
* Route53 will monitor the health of your primary site using a health check.
* A health check monitors the health of your end points

****



**Geolocation**

* If a user request website from London, he will be directed to London region.
* If a user request website from India, he will be directed to India region.
* Use when you want to route traffic based on the location of your users.
* Geolocation routing lets you choose where your traffic will be sent based on the geographical location
* of your users(ie, the location from which DNS queries originate ).
* For Ex: You might want all queries from Europe to be routed to a fleet of EC2 instances that are
* specifically configured for your European customers.
* These servers have local language and displays Euros instead dollars.
* Demo: Create 3 records with 3 diff Ips with no health checks and keep 1 default for all default traffic.



* Application hosted in Mumbai, if any one accessing same application from **INDIA will get low latency**
* Same application if we are accessing from US will get high latency will improve this using edge location
* based on the user’s location **ROUTE53** will route to the respective records
* We need to create 6 records based on our example
* request will go to the ROUTE53 and ROUTE53 understand like request came from US and will give the response based on user location
* **Except simple routing policy, remaining four policies are related to multiple regions.**

**Multivalue answer routing policy** – Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

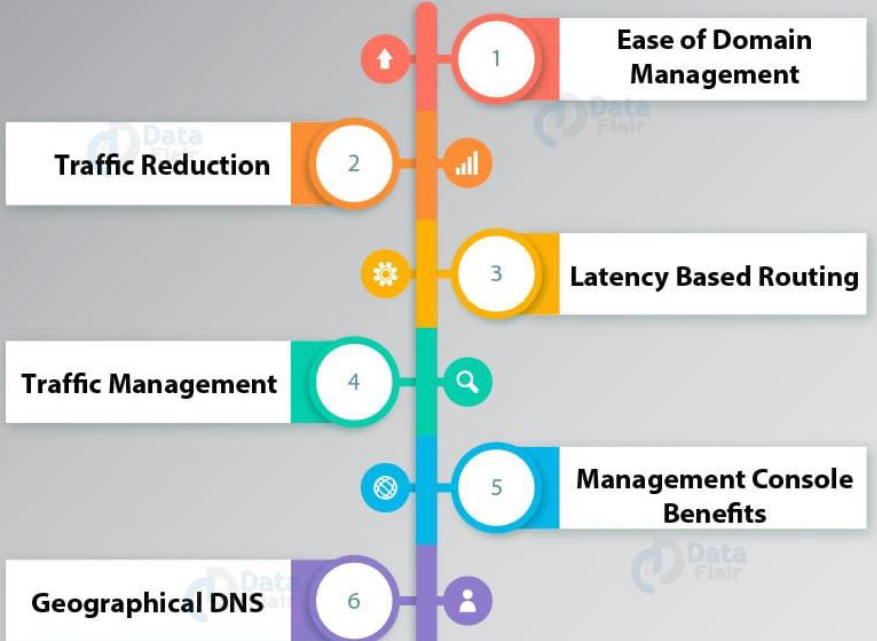
**Same as simple routing policy but MV has health checks**

* MultiValue answer routing lets you configure Amazon Route53 to return multiple values, such as IPaddresses 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 to check the health of the each resource, so Route3 will return only values for healthy resources.
* This is very similar to simple routing however it allows you to put health check on each record set



**Features of Route53**

* Easy to register your domain − We can purchase all level of domains like .com, .net, .org, etc. directly from Route 53.
* **Highly reliable** − Route 53 is built using AWS infrastructure. Its distributed nature towards DNS servers help to ensure a consistent ability to route applications of end users.
* **Scalable** − Route 53 is designed in such a way that it automatically handles large volume queries without the user’s interaction.
* **Can be used with other AWS Services** − Route 53 also works with other AWS services. It can be used to map domain names to our Amazon EC2 instances,
* Amazon S3 buckets, Amazon and other AWS resources.
* **Easy to use** − It is easy to sign-up, easy to configure DNS settings, and provides quick response to DNS queries.
* **Health Check**: Route 53 monitors the health of the application. If an outage is
* detected, then it automatically redirects the users to a healthy resource.
* **Cost-Effective** − Pay only for the domain service and the number of queries that the
* service answers for each domain.
* **Secure** − By integrating Route 53 with AWS (IAM), there is complete control over every user within the AWS account, such as deciding which user can access which part of Route 53.



* ELB’s don’t have an IP address. You resolve them using a DNS name. For naked

domain names like (bbc.co.uk) a IPv4 address is needed. Amazon resolves this by

using Alias Records.

* Alias Records allow you to resolve naked domain names (Zone Apex record) to an ELB DNS record not CNAMES.
* Alias records are not charged, CNAME are charged
* 50 Domains per account and 500 public hosted zones per account, no limit on

private hosted zones.