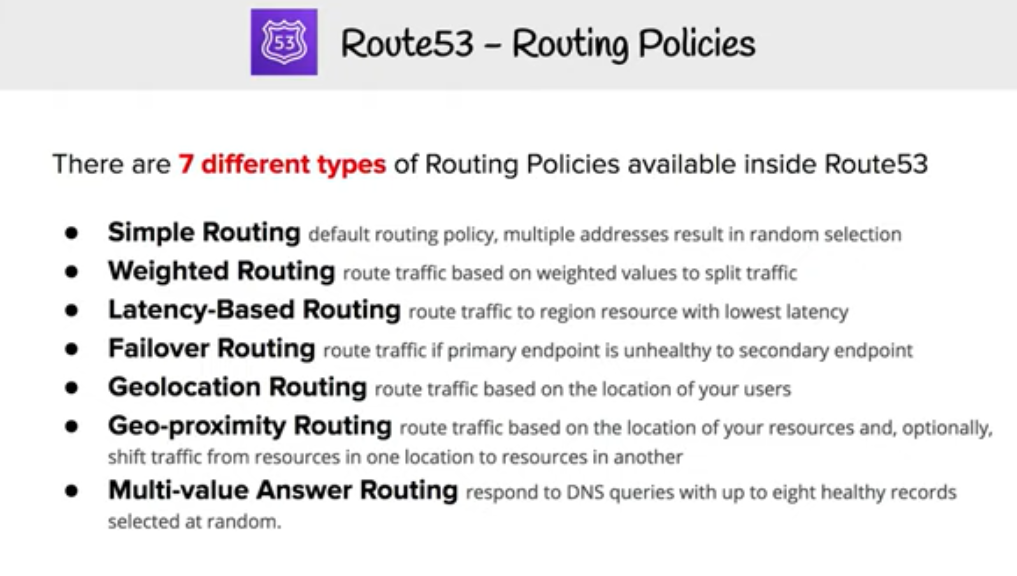
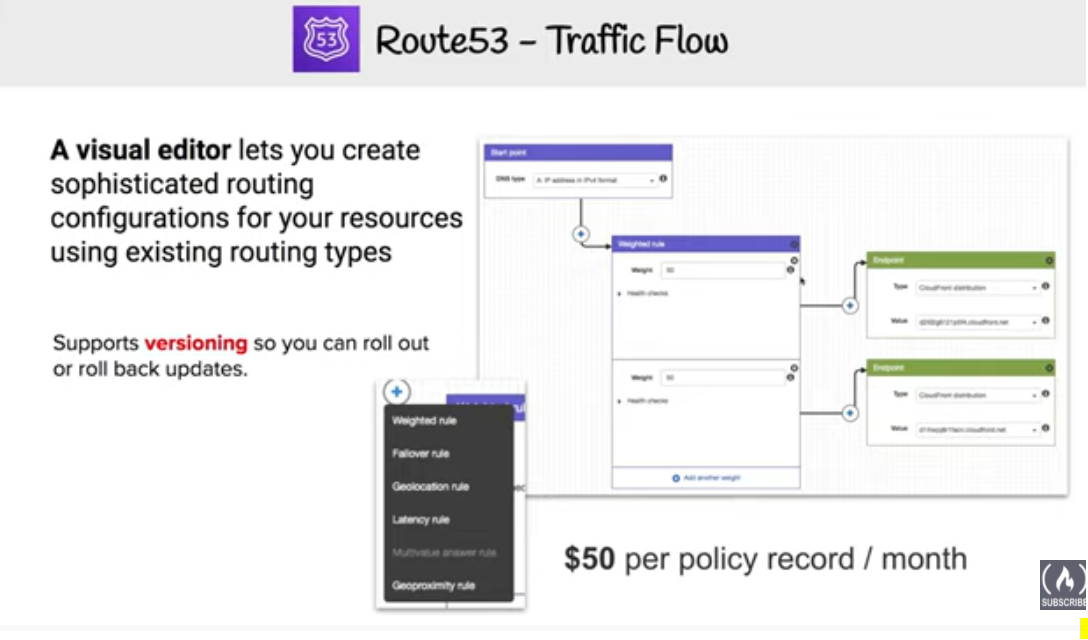
AWS Route 53 Routing Policy

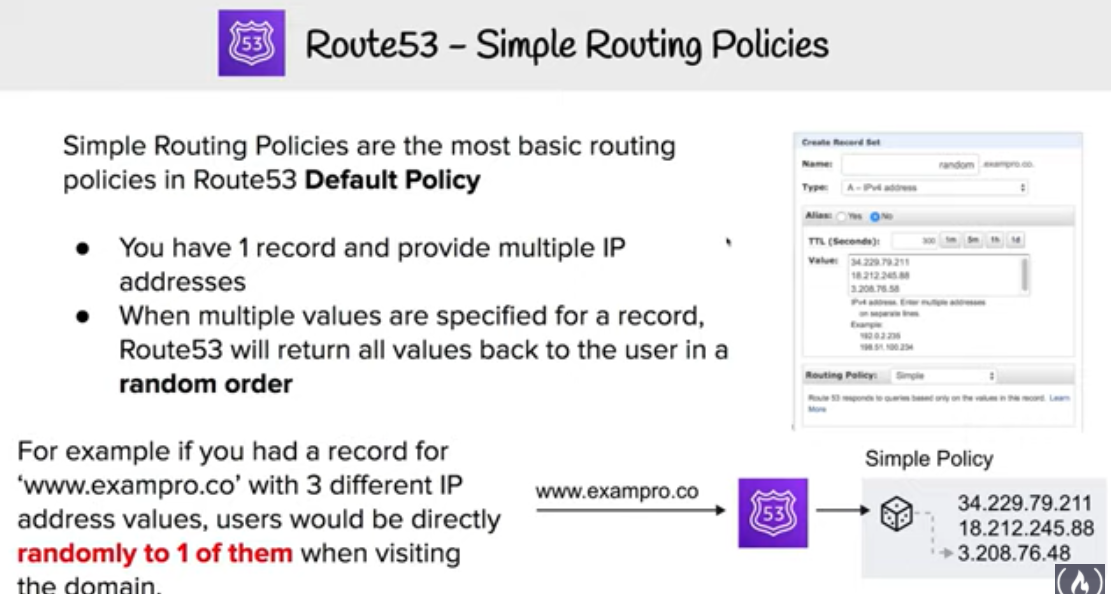
# AWS Route 53 Routing Policy





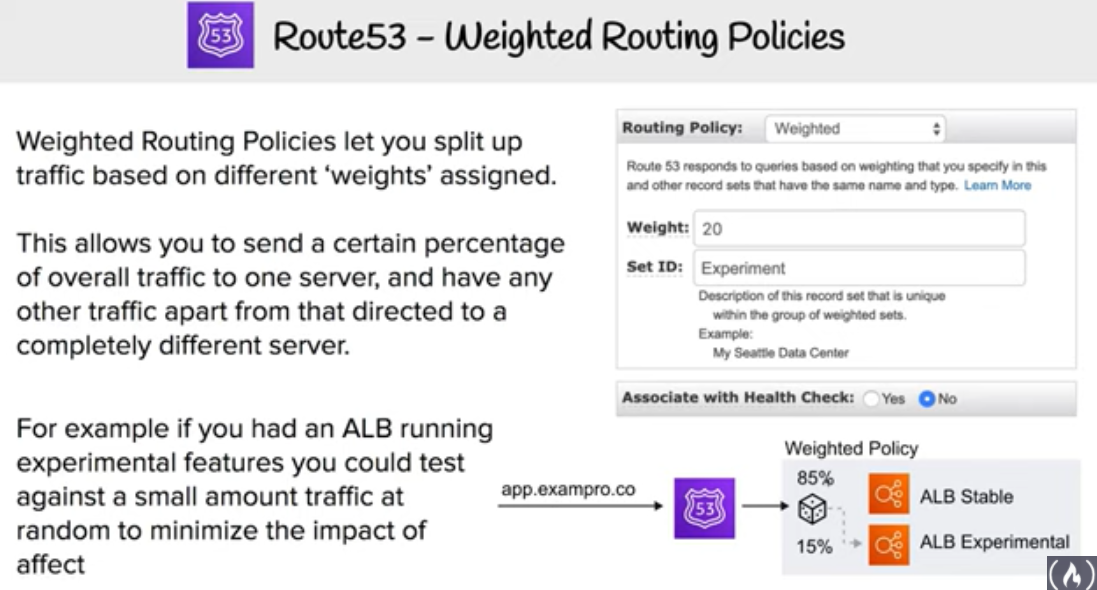
AWS Route 53 routing policy determines how AWS would respond to the DNS queries and provides multiple Routing policy options

## Simple Routing Policy



* Simple routing policy is a ***simple round robin policy*** and can be applied when there is a single resource doing the function for the domain for e.g. web server that serves content for the website
* AWS Route 53 responds to the DNS queries based on the values in the resource record set for e.g. ip address in an A record

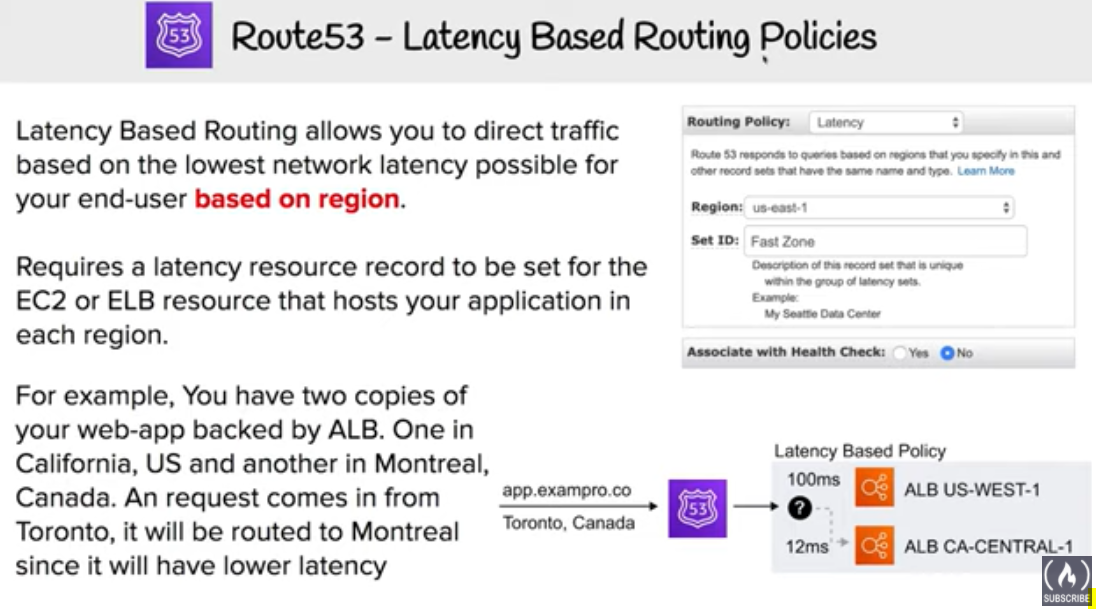
## Weighted Routing Policy



* Weighted routing policy enables Route 53 to route traffic to different resources in specified proportions (weights) for e.g., 75% one server and 25% to the other during a pilot release
* **Weights can be assigned any number from 0 to 255**
* Weighted routing policy can be applied when there are multiple resources that perform the same function for e.g., webservers serving the same site
* Weighted resource record sets let you associate multiple resources with a single DNS name
* Common use cases include
  + load balancing
  + A/B testing and piloting new versions of software
* **To create a group of weighted resource record sets, two or more resource record sets can be created that have the same combination of DNS name and type, and each resource record set is assigned a unique identifier and a relative weight.**
* When processing a DNS query, Route 53 searches for a resource record set or a group of resource record sets that have the specified name and type.
* Route 53 selects one from the group. Probability of any one resource record set being selected depends on its weight as a proportion of the total weight for all resource record sets in the group for e.g., suppose for www.example.com has three resource record sets with weights of 1 (20%), 1 (20%), and 3 (60%)(sum = 5). On average, Route 53 selects each of the first two resource record sets one-fifth of the time, and returns the third resource record set three-fifths of the time.

## 

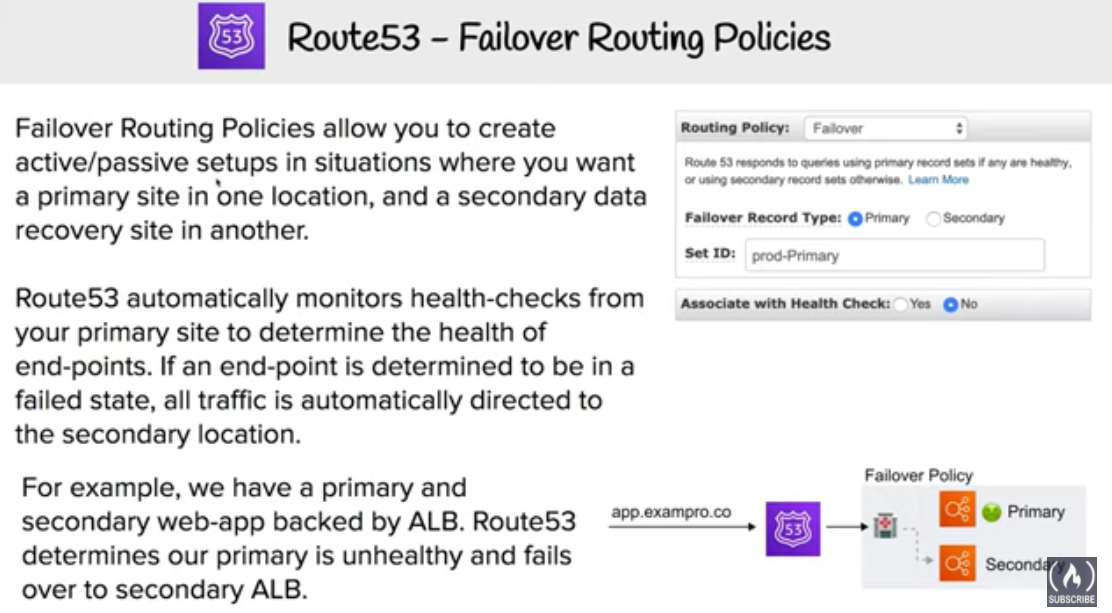
## Latency-based Routing (LBR) Policy



* Latency-based Routing Policy enables Route 53 to respond to the DNS query based on which data center gives the user the lowest network latency
* Latency-based routing policy can be used when there are multiple resources performing the same function and Route 53 needs to be configured to respond to the DNS queries with the resources that provide the fastest response with lowest latency
* Latency resource record set can be created for the EC2 resource in each region that hosts the application. When Route 53 receives a query for the corresponding domain, it selects the latency resource record set for the EC2 region that gives the user the lowest latency. Route 53 then responds with the value associated with that resource record set for e.g., you might have web servers for example.com in the EC2 data centers in Ireland and in Tokyo. When a user browses to example.com from Singapore, Route 53 will pick up the data center (Tokyo) which has the lowest latency from the users location
* **Latency between hosts on the Internet can change over time as a result of changes in network connectivity and routing. Latency-based routing is based on latency measurements performed over a period of time, and the measurements reflect these changes**for e.g. if the latency from the user in Singapore to Ireland improves, the user can be routed to Ireland
* Latency based routing cannot guarantee users from the same geographic will be served from the same location for any compliance reason
* Latency resource record sets can be created using any record type that Route 53 supports except NS or SOA



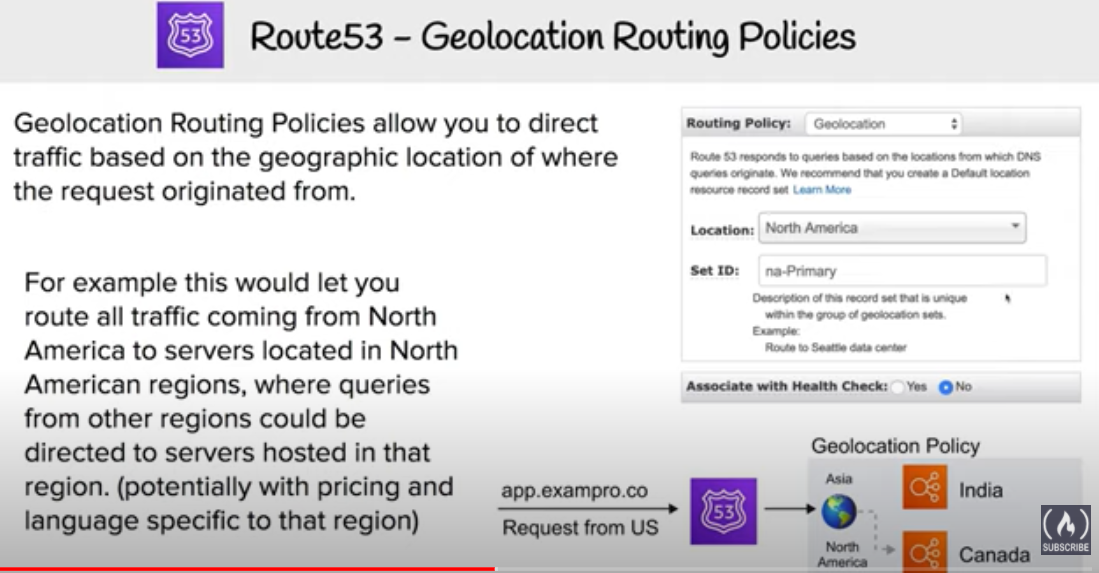
## Failover Routing Policy



* **Failover routing policy allows active-passive failover configuration, in which one resource takes all traffic when it’s available and the other resource takes all traffic when the first resource isn’t available.**
* **Route 53 health checking agents will monitor each location/endpoint of the application to determine the availability**.
* **Failover routing policy is applicable for Public hosted zones only**



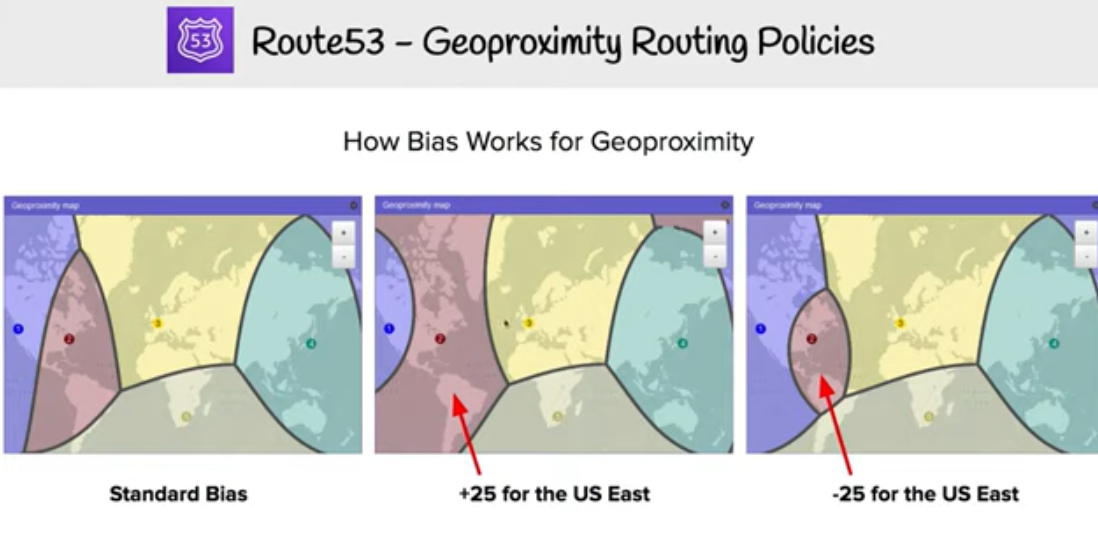
## Geolocation Routing Policy



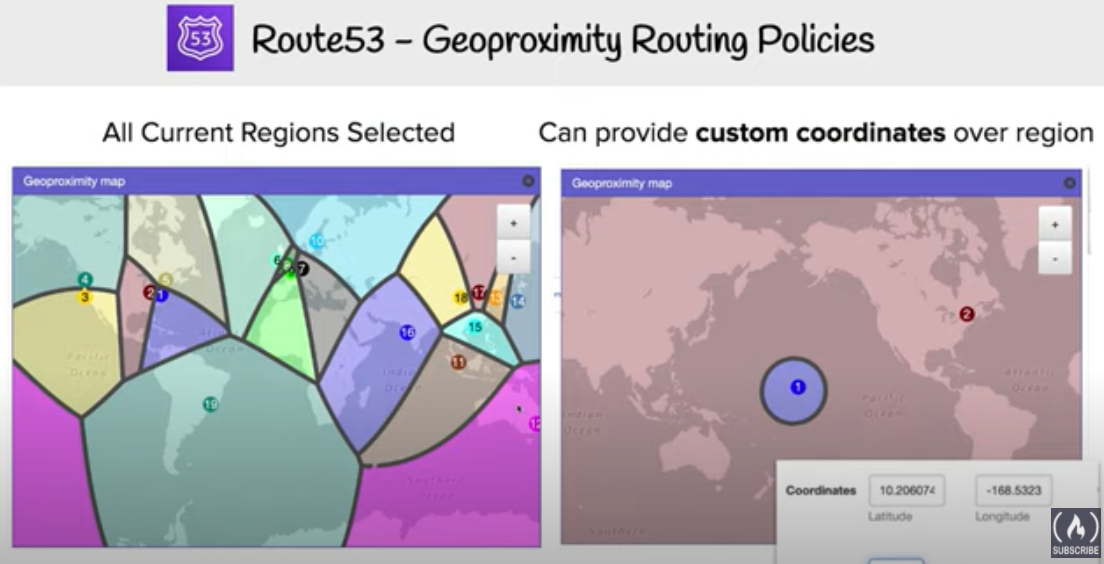
* **Geolocation routing policy enables Route 53 to respond to DNS queries based on the geographic location of the users i.e. location from which the DNS queries originate**
* Common use cases include
  + **localization of content and presenting some or all of the website in the users language**
  + **restrict distribution of content to only the locations in which you have distribution rights**.
  + **balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint**.
* **Geolocation routing policy allows geographic locations to be specified by continent, country, or by state (only in US)**
* Geolocation record sets, if created, for overlapping geographic regions for e.g. continent and then for the country within the same continent, priority goes to the smallest geographic region, which allows some queries for a continent to be routed to one resource and queries for selected countries on that continent to a different resource
* Geolocation works by mapping IP addresses to locations, which might not mapped to a exact geographic location
* **A default resource record set can be created to handle these queries and also the ones which do not have an explicit record set created**
* Route 53 returns a “no answer”response for queries from those locations, if a default resource record set if not created
* Two geolocation resource record sets that specify the same geographic location **cannot** be created
* **Route 53 supports the edns-client-subnet extension of EDNS0 (EDNS0 adds several optional extensions to the DNS protocol.) to improve the accuracy of geolocation routing**

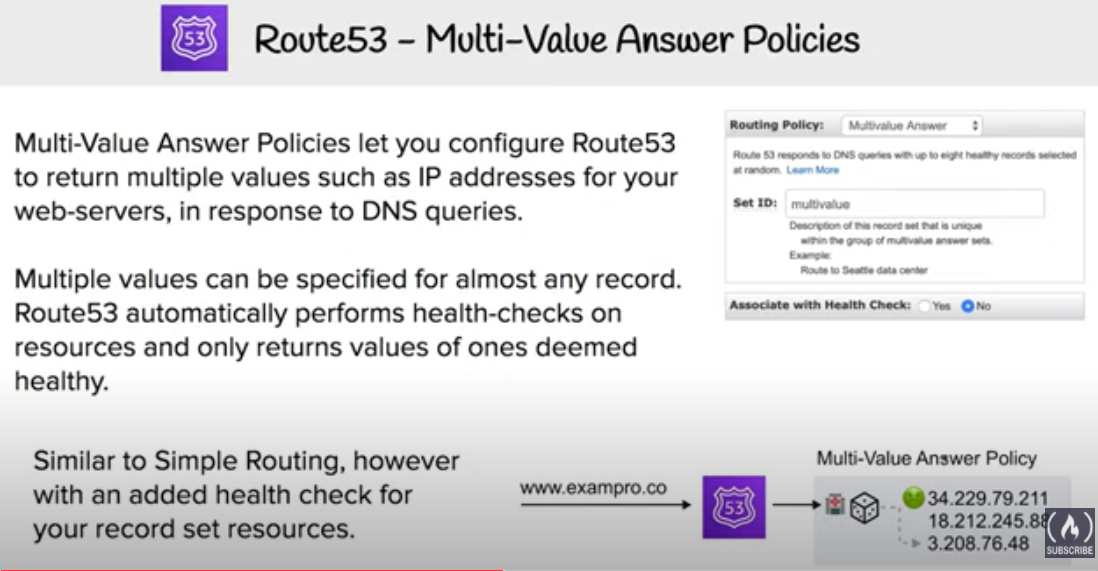
## 

## 



## 





## AWS Certification Exam Practice Questions

1. You have deployed a web application targeting a global audience across multiple AWS Regions under the domain name example.com. You decide to use Route 53 Latency-Based Routing to serve web requests to users from the region closest to the user. To provide business continuity in the event of server downtime you configure weighted record sets associated with two web servers in separate Availability Zones per region. During a DR test you notice that when you disable all web servers in one of the regions Route 53 does not automatically direct all users to the other region. What could be happening? (Choose 2 answers)
   1. Latency resource record sets cannot be used in combination with weighted resource record sets.
   2. **You did not setup an http health check for one or more of the weighted resource record sets associated with the disabled web servers**
   3. The value of the weight associated with the latency alias resource record set in the region with the disabled servers is higher than the weight for the other region.
   4. One of the two working web servers in the other region did not pass its HTTP health check
   5. **You did not set “Evaluate Target Health” to “Yes” on the latency alias resource record set associated with example.com in the region where you disabled the servers.**
2. The compliance department within your multi-national organization requires that all data for your customers that reside in the European Union (EU) must not leave the EU and also data for customers that reside in the US must not leave the US without explicit authorization. What must you do to comply with this requirement for a web based profile management application running on EC2?
   1. Run EC2 instances in multiple AWS Availability Zones in single Region and leverage an Elastic Load Balancer with session stickiness to route traffic to the appropriate zone to create their profile (should be in 2 different regions – US and Europe)
   2. Run EC2 instances in multiple Regions and leverage Route 53’s Latency Based Routing capabilities to route traffic to the appropriate region to create their profile (Latency based routing policy would not guarantee the compliance requirement)
   3. **Run EC2 instances in multiple Regions and leverage a third party data provider to determine if a user needs to be redirect to the appropriate region to create their profile**
   4. Run EC2 instances in multiple AWS Availability Zones in a single Region and leverage a third party data provider to determine if a user needs to be redirect to the appropriate zone to create their profile(should be in 2 different regions – US and Europe)
3. A US-based company is expanding their web presence into Europe. The company wants to extend their AWS infrastructure from Northern Virginia (us-east-1) into the Dublin (eu-west-1) region. Which of the following options would enable an equivalent experience for users on both continents?
   1. Use a public-facing load balancer per region to load-balance web traffic, and enable HTTP health checks.
   2. Use a public-facing load balancer per region to load-balance web traffic, and enable sticky sessions.
   3. **Use Amazon Route 53, and apply a geolocation routing policy to distribute traffic across both regions**
   4. Use Amazon Route 53, and apply a weighted routing policy to distribute traffic across both regions.
4. You have been asked to propose a multi-region deployment of a web-facing application where a controlled portion of your traffic is being processed by an alternate region. Which configuration would achieve that goal?
   1. **Route 53 record sets with weighted routing policy**
   2. Route 53 record sets with latency based routing policy
   3. Auto Scaling with scheduled scaling actions set
   4. Elastic Load Balancing with health checks enabled
5. Your company is moving towards tracking web page users with a small tracking image loaded on each page. Currently you are serving this image out of us-east, but are starting to get concerned about the time it takes to load the image for users on the west coast. What are the two best ways to speed up serving this image? Choose 2 answers
   1. **Use Route 53’s Latency Based Routing and serve the image out of us-west-2 as well as us-east-1**
   2. **Serve the image out through CloudFront**
   3. Serve the image out of S3 so that it isn’t being served of your web application tier
   4. Use EBS PIOPs to serve the image faster out of your EC2 instances
6. Your API requires the ability to stay online during AWS regional failures. Your API does not store any state, it only aggregates data from other sources – you do not have a database. What is a simple but effective way to achieve this uptime goal?
   1. Use a CloudFront distribution to serve up your API. Even if the region your API is in goes down, the edge locations CloudFront uses will be fine.
   2. Use an ELB and a cross-zone ELB deployment to create redundancy across datacenters. Even if a region fails, the other AZ will stay online.
   3. Create a Route53 Weighted Round Robin record, and if one region goes down, have that region redirect to the other region.
   4. **Create a Route53 Latency Based Routing Record with Failover and point it to two identical deployments of your stateless API in two different regions. Make sure both regions use Auto Scaling Groups behind ELBs.** (Refer [link](http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/dns-failover.html))