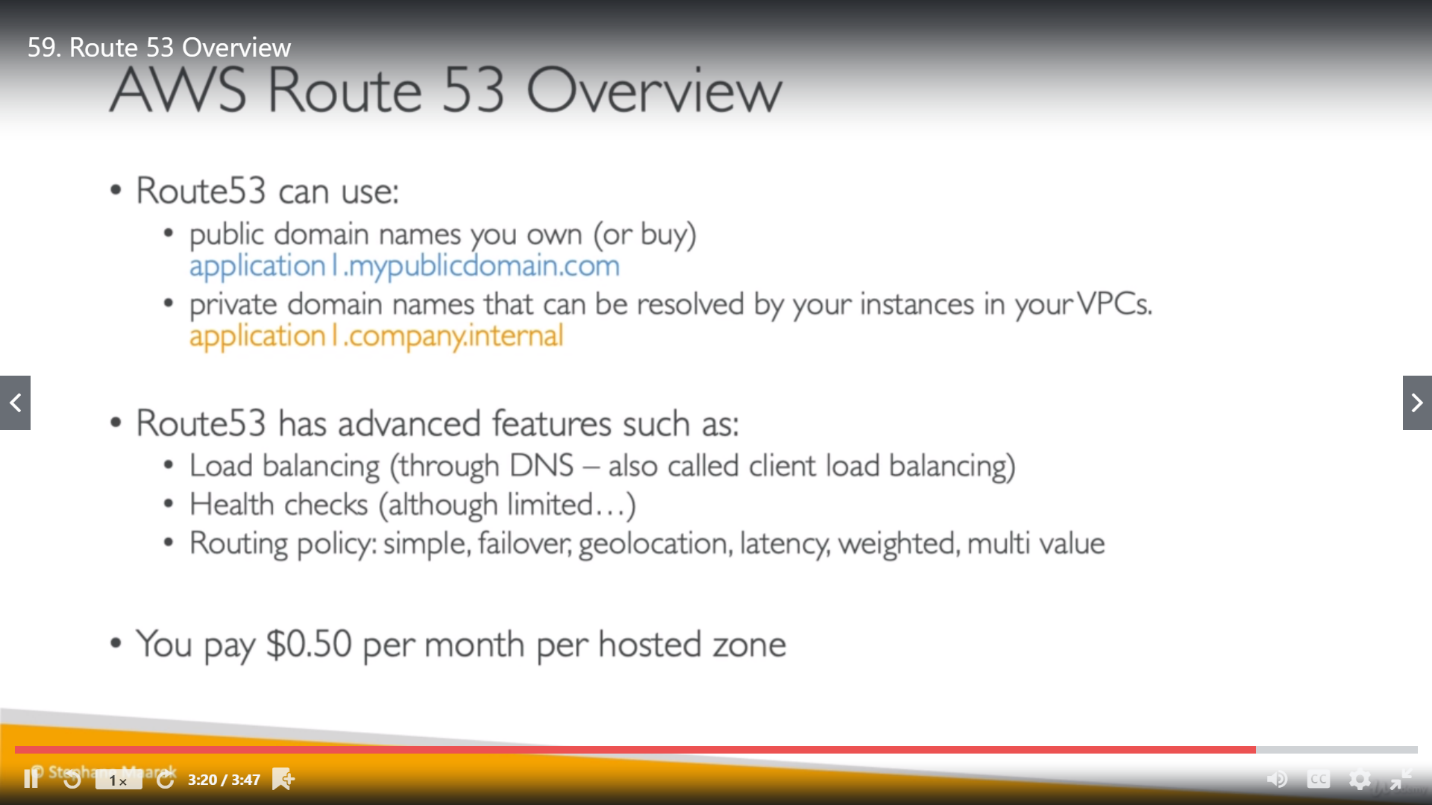
**Route-53**

**Topic covered: TTL, Record, Policy, root and non-root domain,**

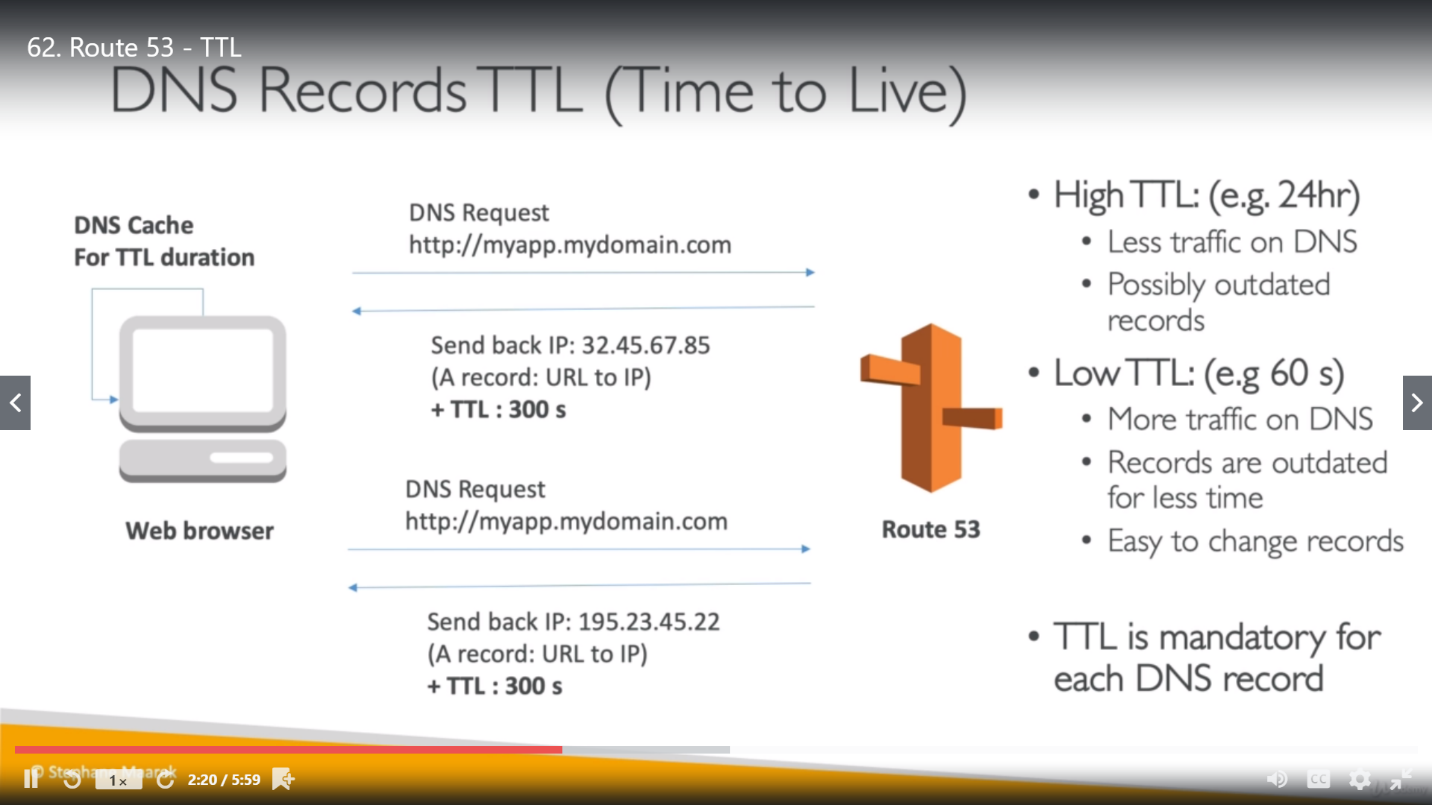
Amazon Route 53 is a highly available and scalable cloud [Domain Name System (DNS)](https://aws.amazon.com/route53/what-is-dns/) web service.





It is effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

**TTL (Time to Live)**: TTL (Time to Live) is a setting for each DNS record that specifies how long a resolver is supposed to cache (or remember) the DNS query before the query expires and a new one needs to be done.



<https://dyn.com/blog/dyn-tech-everything-you-ever-wanted-to-know-about-ttls/>

Once the request come from Browser to Route-53, Route-53 will send response with IP along with TTL (ex. 300sec). So, web browser will cache that DNS request and response for TTL duration. So, if we hit DNS name within 300 sec again, browser won’t ask Route-53 for IP.

**url = protocol + subdomain + second-level domain + top-level domain/ subdirectory**

<https://blog.hubspot.com/marketing/parts-url>

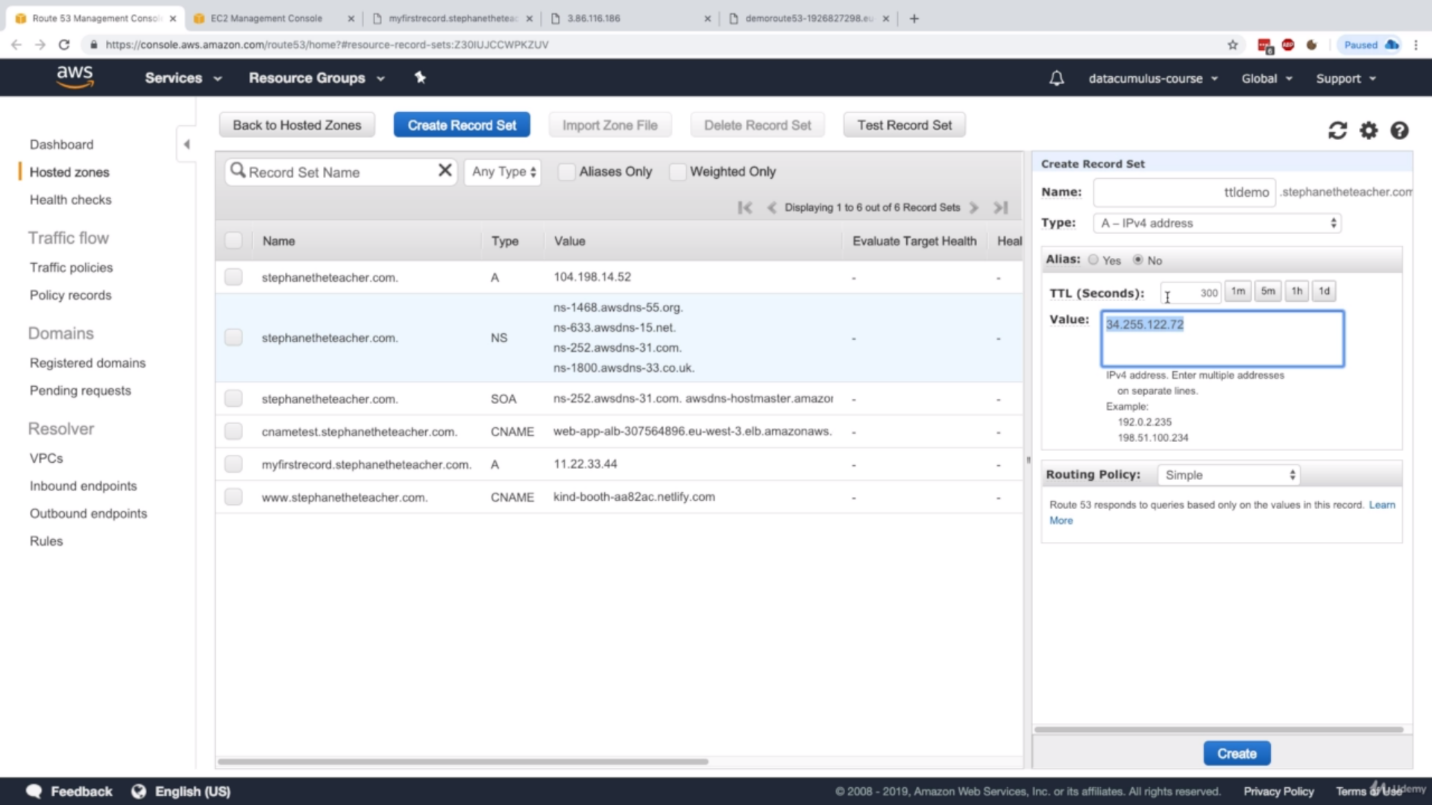
**DNS: D***omain***N***ame***S***ystem* (or **S***ervice* or **S***erver*), an [Internet](https://www.webopedia.com/TERM/I/Internet.html) service that translates [*domain names*](https://www.webopedia.com/TERM/D/domain_name.html) into IP addresses.

**Domain name:** blog.hubspot.com, here while buying AWS domain, we bought only hubspot.com means only (second level domain and top level domain), subdomain we can change as anything.

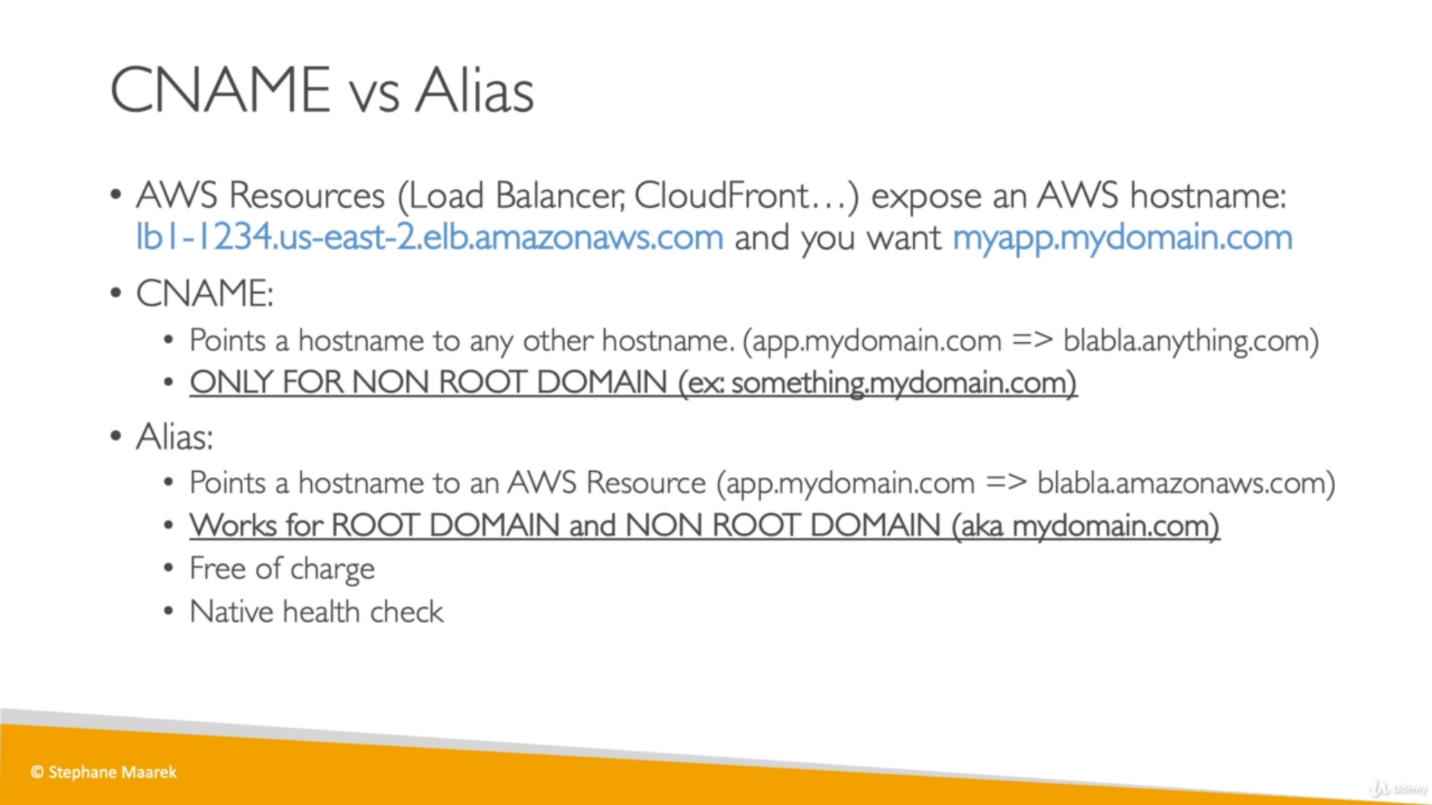
DNS has no concept of ports for older protocols such as HTTP, HTTPS, and SSL. DNS only points to the IP address.

The port to connect to for a particular service is [determined by convention](https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers). For example, the default port for HTTP is 80, the default port for HTTPS is 443, and the default port for SSH is 22.( <https://webmasters.stackexchange.com/questions/55123/can-dns-point-to-specific-port>)

**Create a record for Route-53**



**Difference between CNAME and Alias**

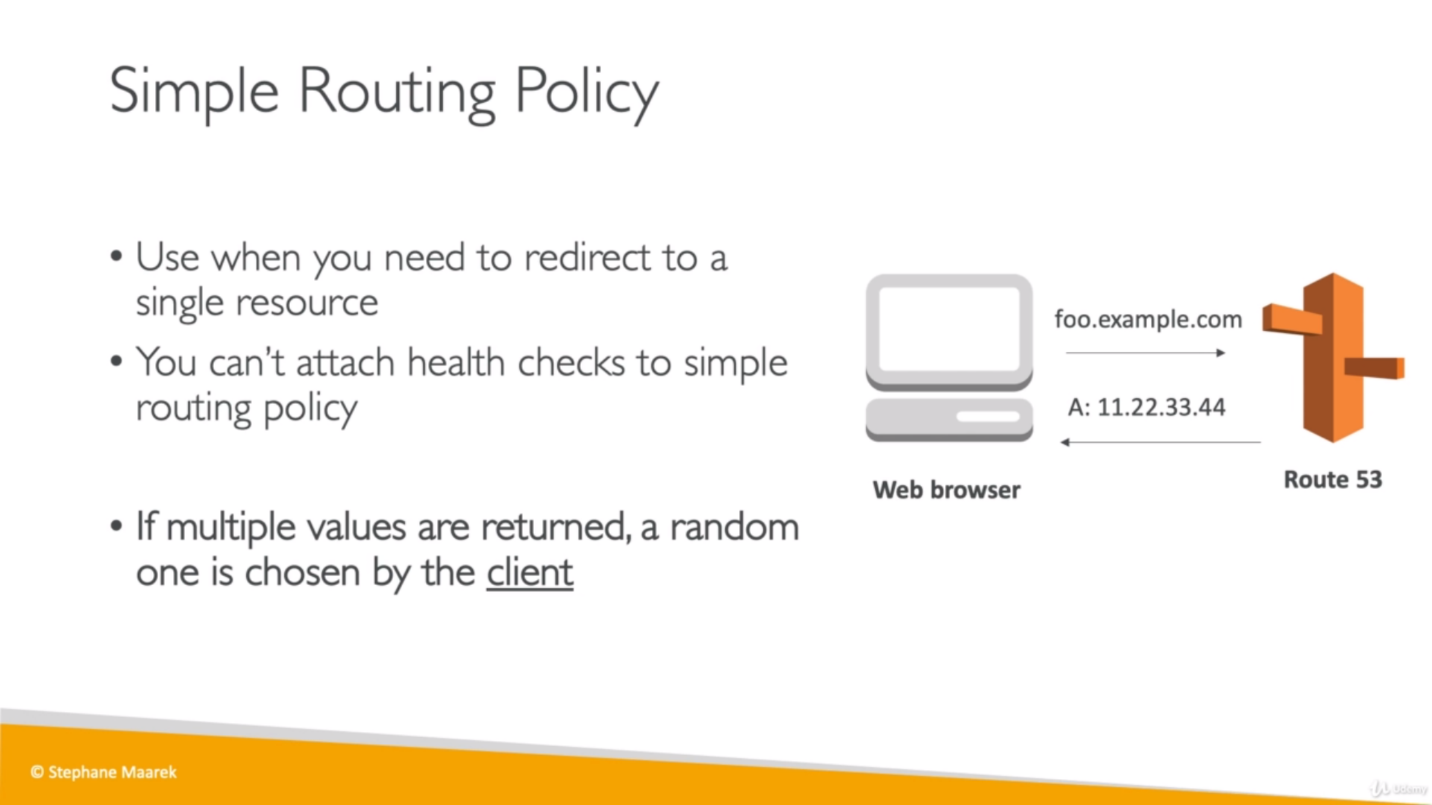


* CNAME can’t be like mydomain.com, it should be in the form of something.mydomain.com
* CNAME could point to anything but Alias can point only to amazon services like, Load balancer record with a route-53.
* Alias can work for both root and non-root domain like mydomain.com will work here.
* So, for root records (ex. ashvani.com) we must use Alias only.
* As alias is free for charge, so if we already paying for domain name through AWS then we should create an Alias for a request coming on url (which have DNS that we brought using AWS) to load balancer.

**Routing policies:**

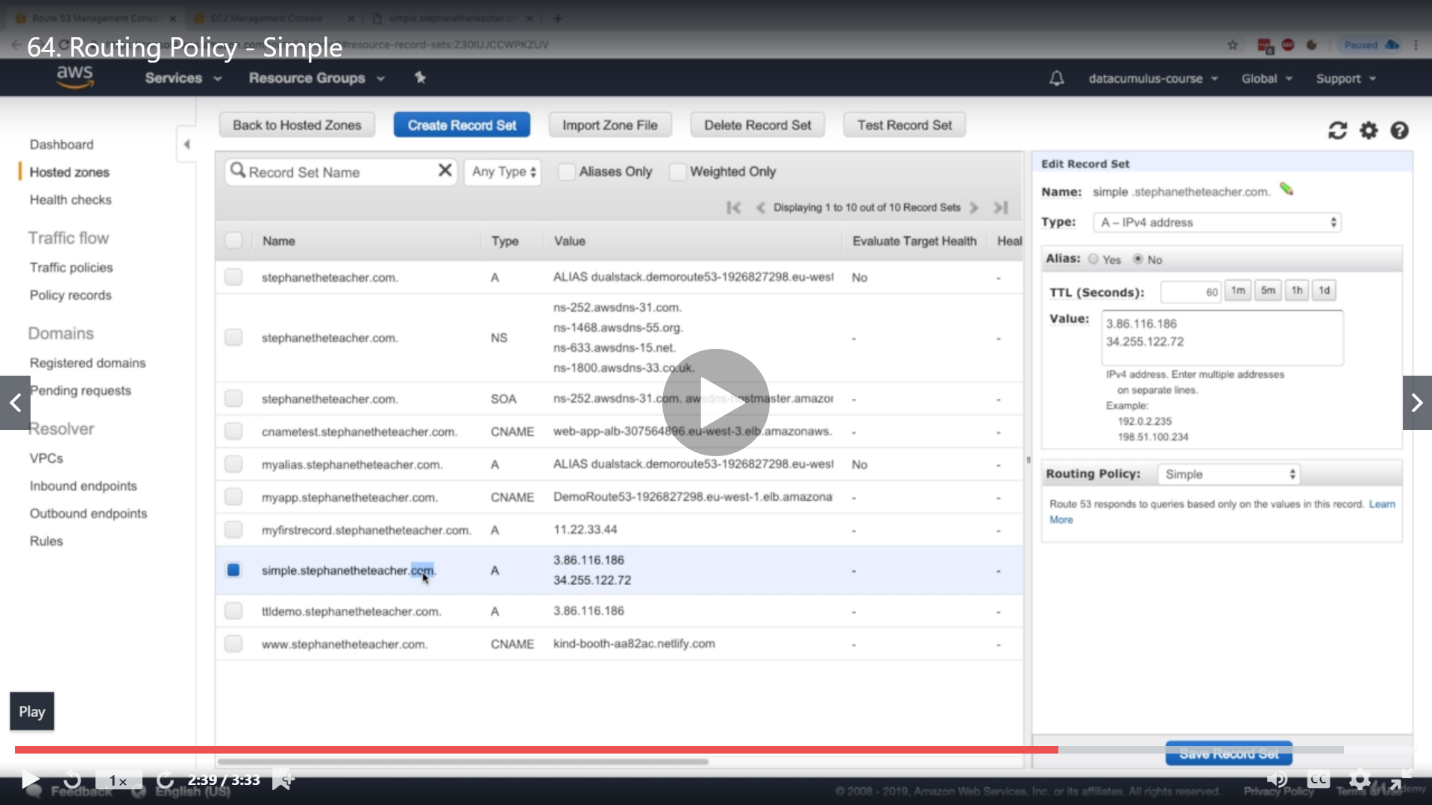
* **Simple Routing Policy**

We create single record set for multiple instances and provide all these instances as value for record set.



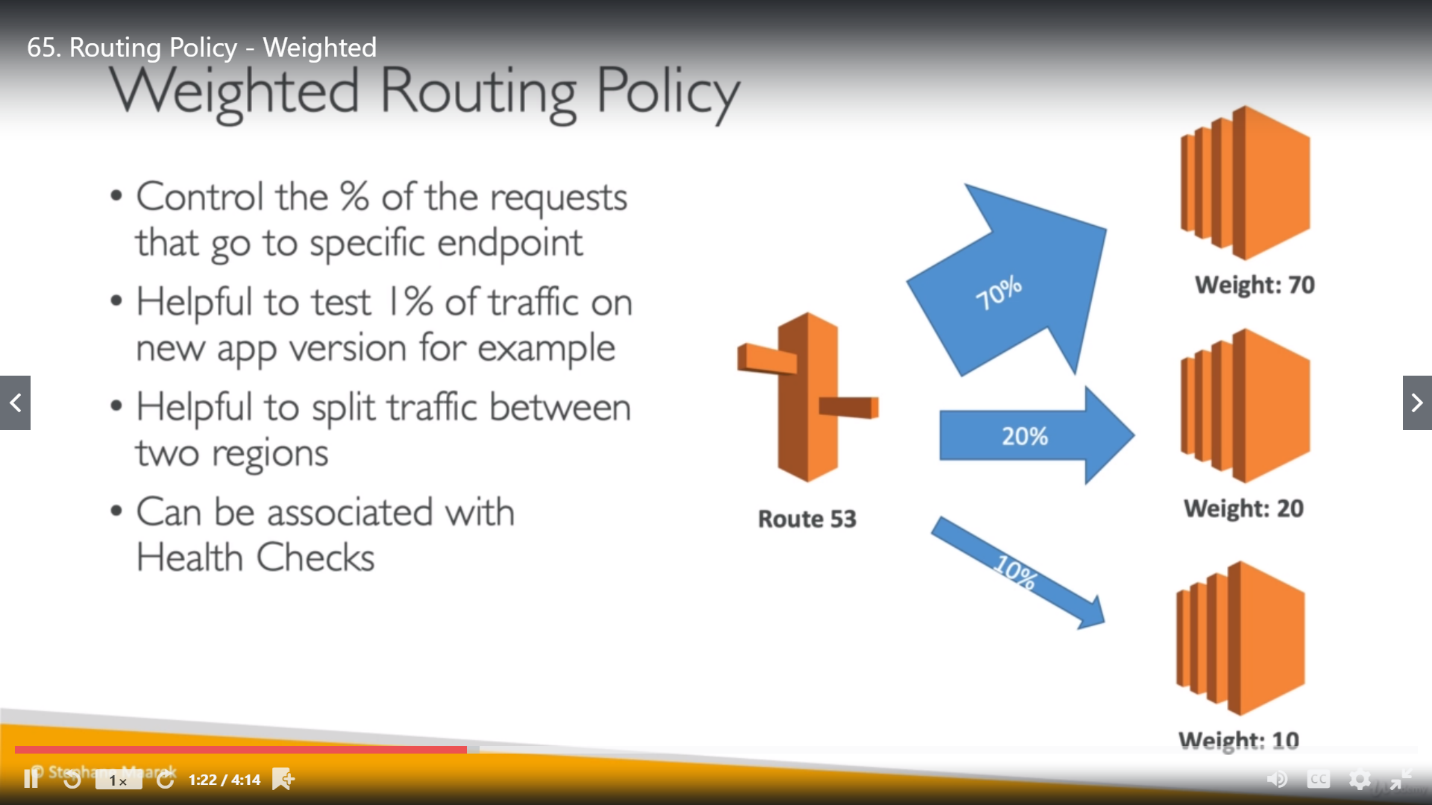
This achieves client-side load balancing, as we can give more than one IP for a single DNS record.

So, once we hit the URL for given TTL, it will show the same response but once the TTL complete there is a chance that we get another instance response.



* **Weighted Routing Policy:**

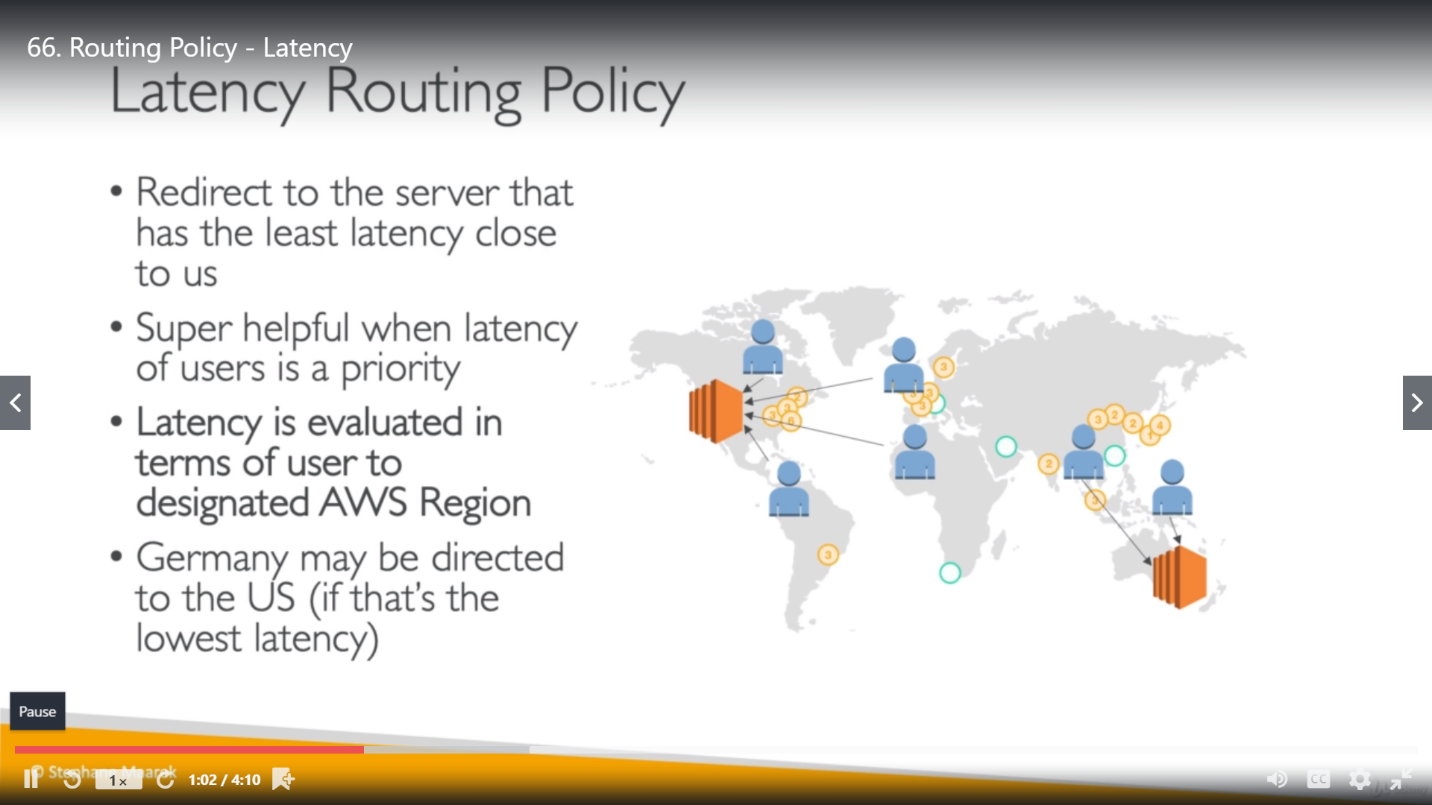
We must create multiple record set for multiple instances. Heath check is additional here.



It’s not must that sum of all weight of load has to be 100.

* **Latency Routing Policy:** (Latency means time interval between stimulation and response).We use it, when you have resources in multiple AWS Regions, and you want to route traffic to the region that provides the best latency.
* To use latency-based routing, you create latency records for your resources in multiple AWS Regions. When Route 53 receives a DNS query for your domain or subdomain (example.com or acme.example.com), it determines which AWS Regions you've created latency records for, determines which region gives the user the lowest latency, and then selects a latency record for that region. Route 53 responds with the value from the selected record, such as the IP address for a web server.

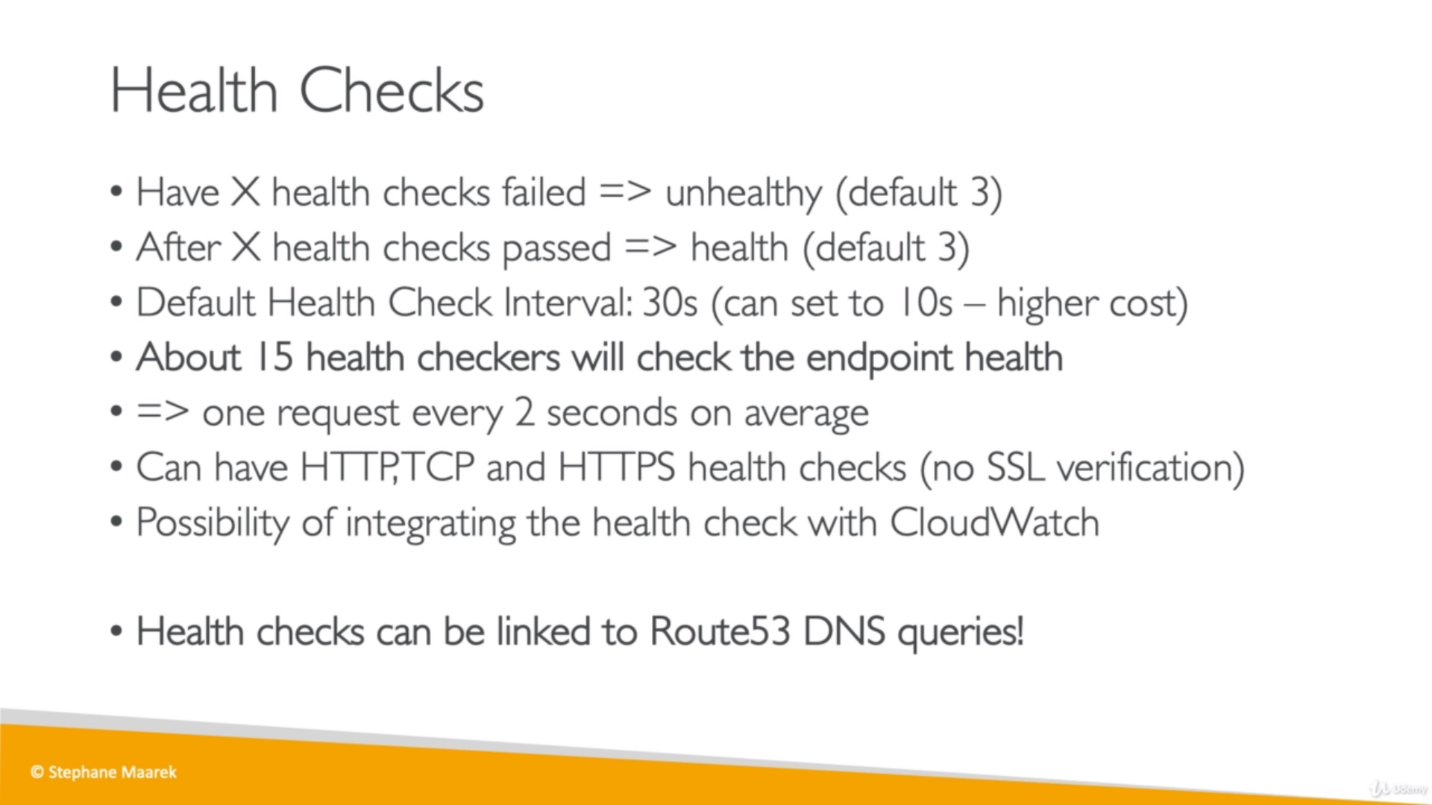
We must create multiple record set for multiple instances.

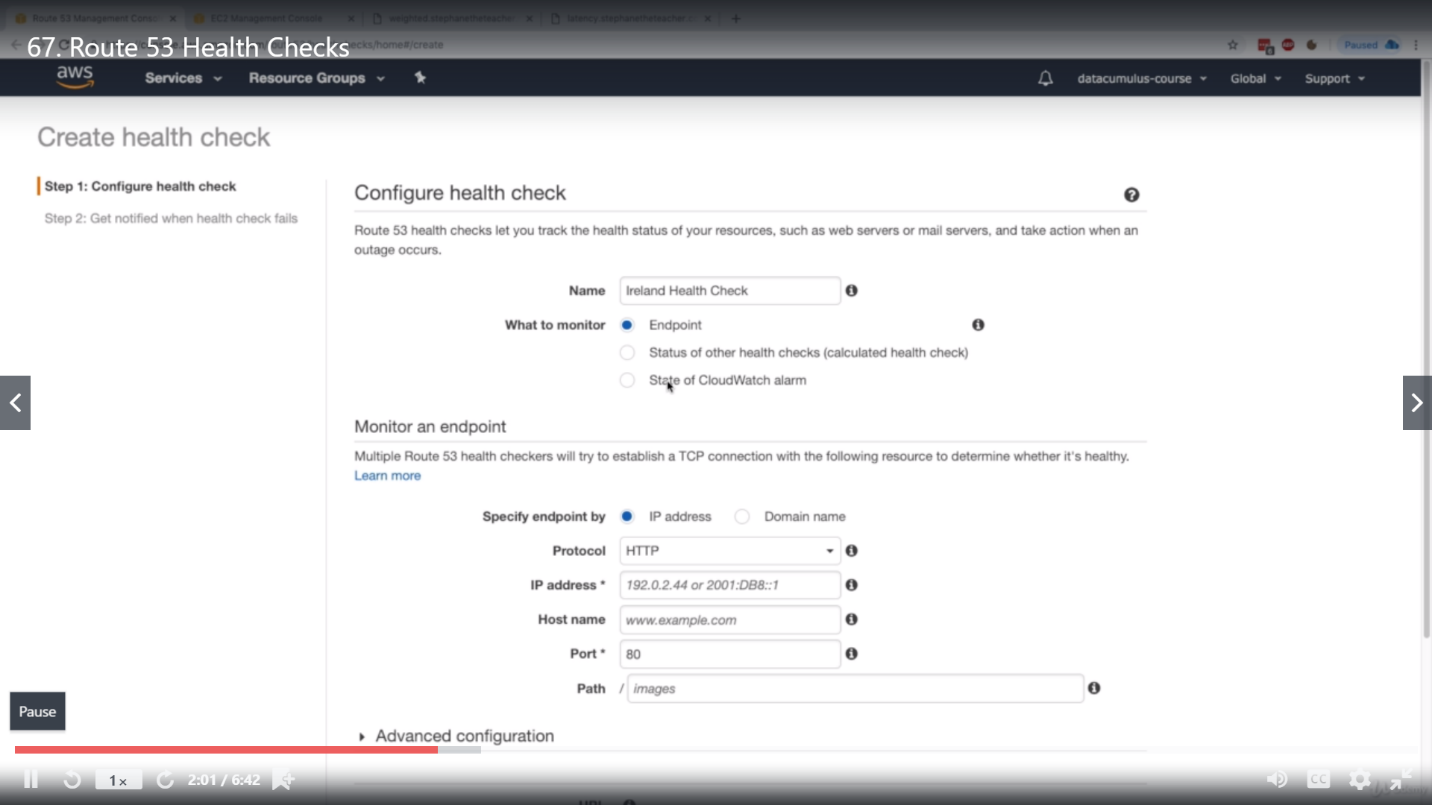


**Health Check:**

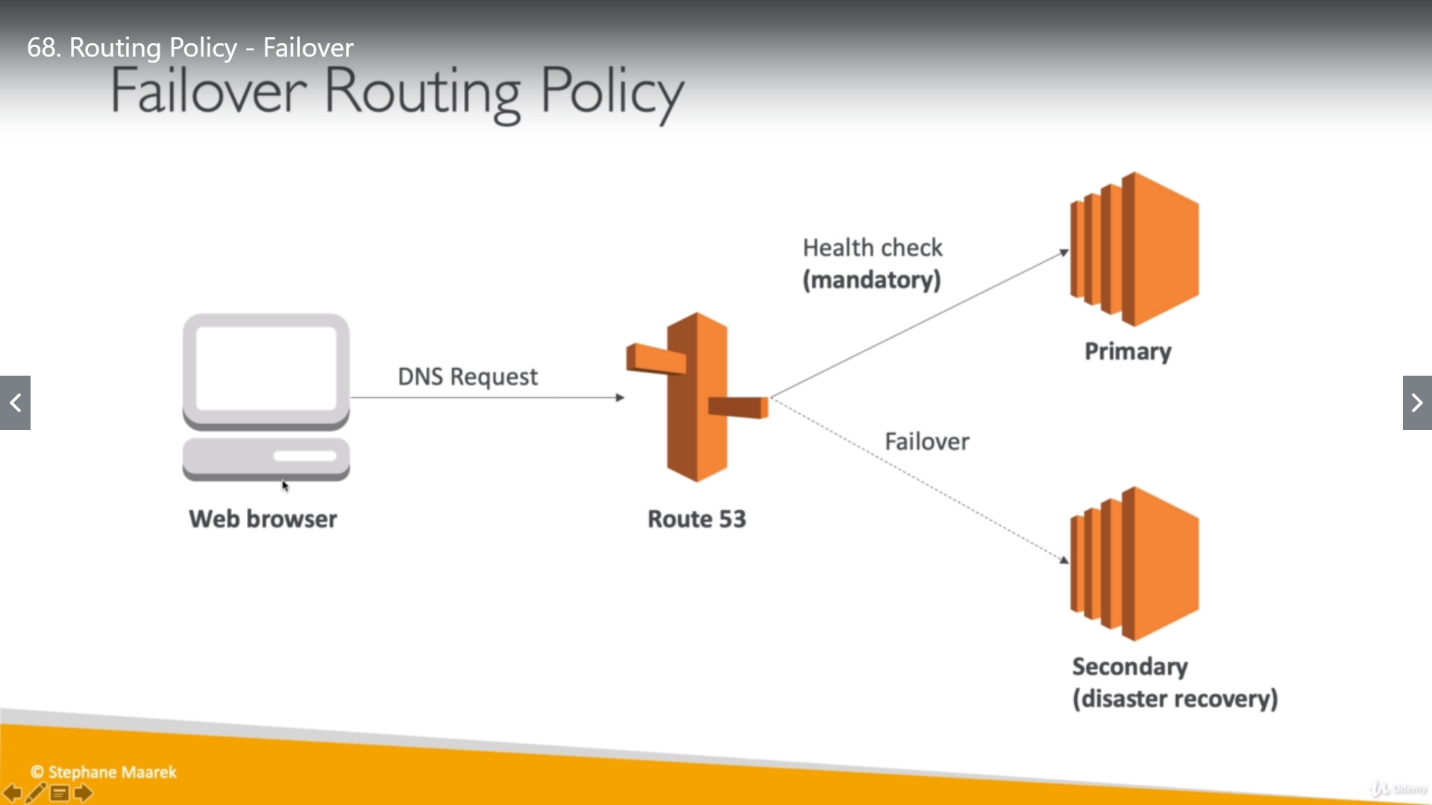
When you have two or more resources that perform the same function, such as two or more web servers for example.com, you can use the following health-checking features to route traffic only to the healthy resources.

Health check is required for failover policy implementation so first we have to create health check for all the instances that we are using for route-53. So, we can configure it according to our requirements as mentioned below:





**Failover routing policy:**



We can have only one primary and one secondary Failover record type.

For primary Failover record type:

this policy must need to associate with health check. (check-in the Health check option while creating the record-set and then associate a health check with this (this associated health check should be the one, which is using the instance similar to IP of current record set)).

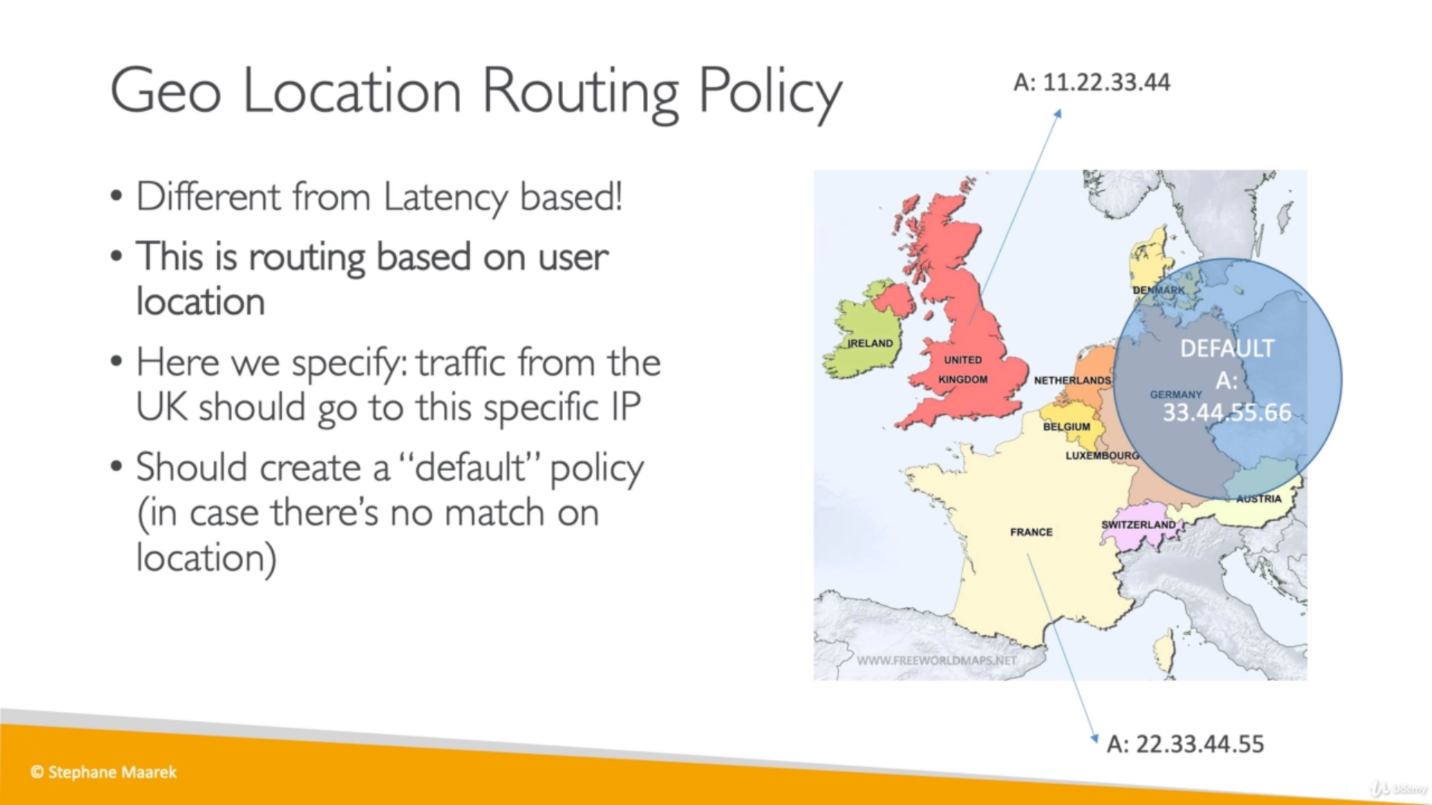
For secondary Failover record type:

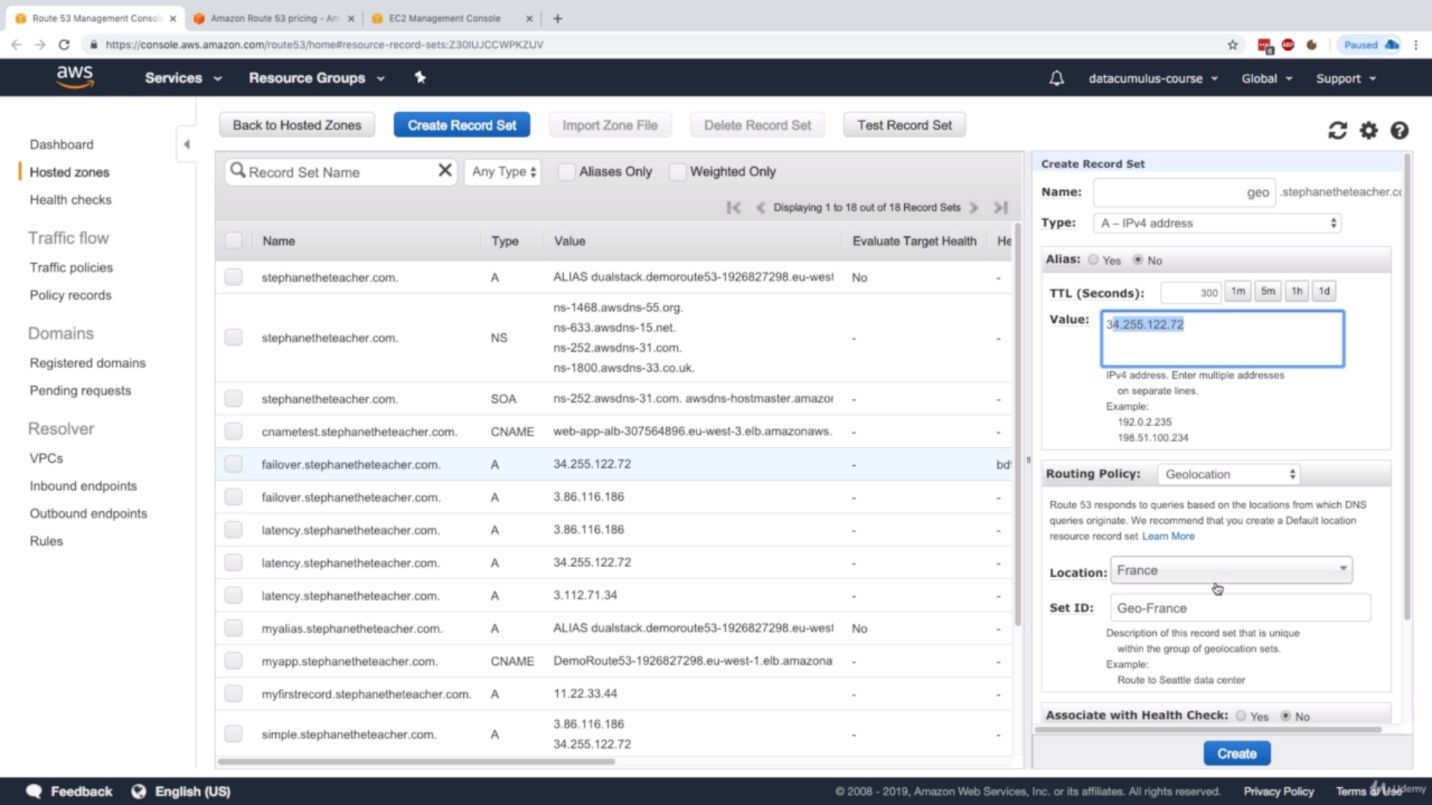
No need to have associated health check. So whenever primary record is not healthy, the request will come to this instance.

**Geolocation routing policy:** Use when we want to route traffic based on the location of our users. Geolocation routing lets you choose the resources that serve your traffic based on the geographic location of your users, meaning the location that DNS queries originate from. For example, you might want all queries from Europe to be routed to an ELB load balancer in the Frankfurt region.

When you use geolocation routing, you can localize your content and present some or all of your website in the language of your users. You can also use geolocation routing to restrict distribution of content to only the locations in which you have distribution rights. Another possible use is for balancing load across endpoints in a predictable, easy-to-manage way, so that each user location is consistently routed to the same endpoint.

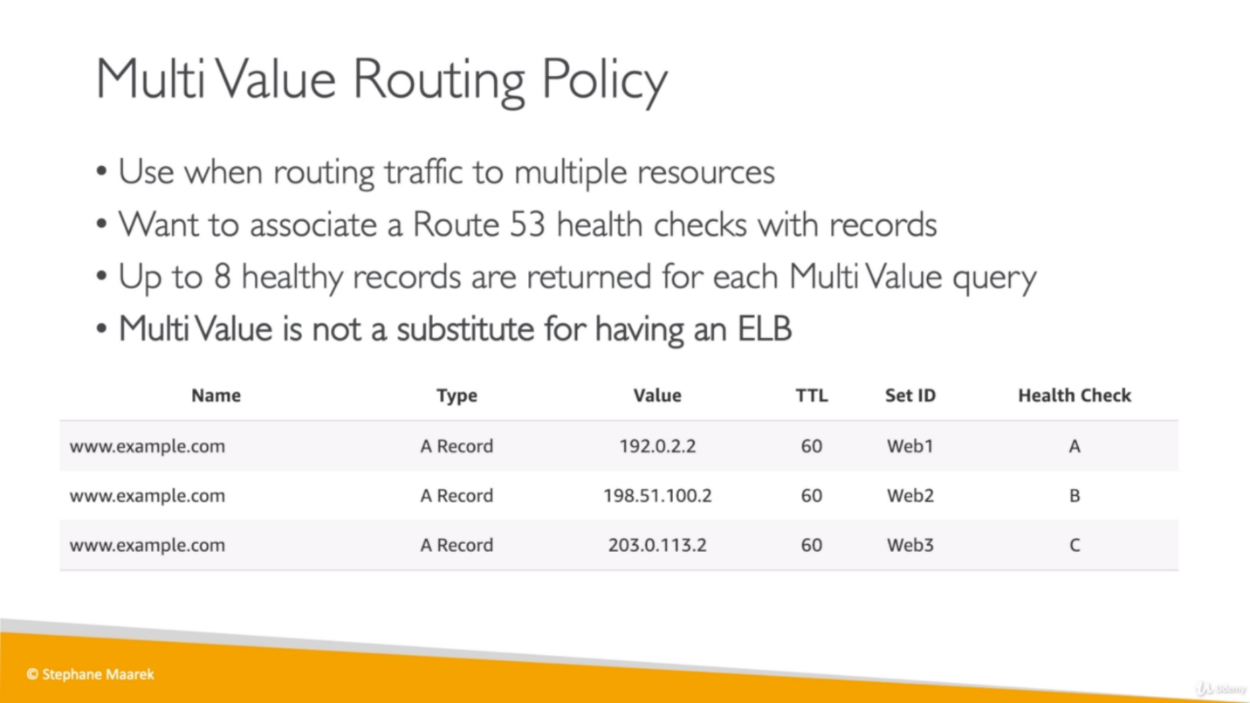
When a DNS request falls under two categories, it will be served by smaller geolocation. As a request in a continent and country will have priority for country.

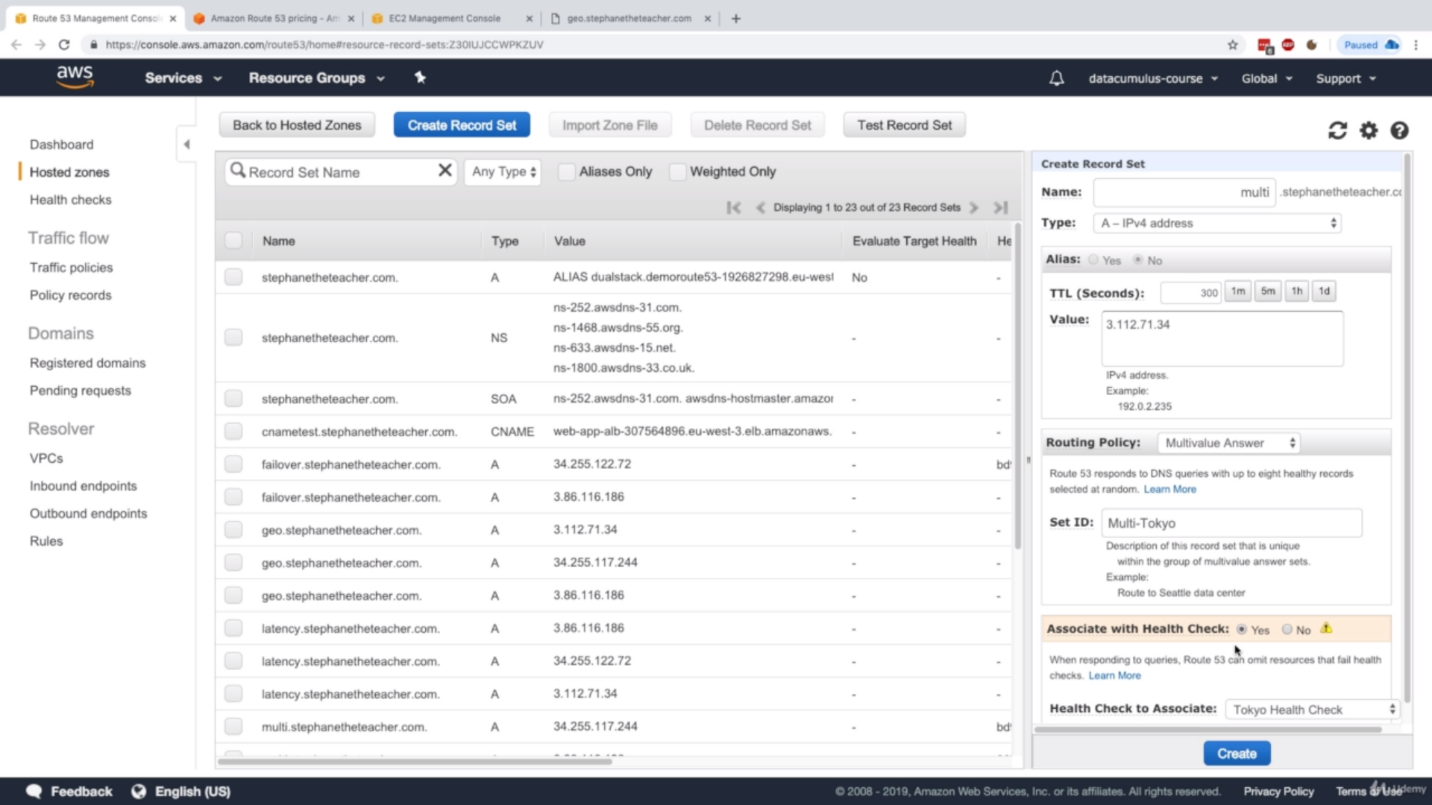




* **Multi Value Routing Policy:**

Improvement of simple routing policy. Use when we want Route 53 to respond to DNS queries with up to eight healthy records selected at random. (it is random not load balanced)





A script where we are giving availability zone details along with installing Apache server.

