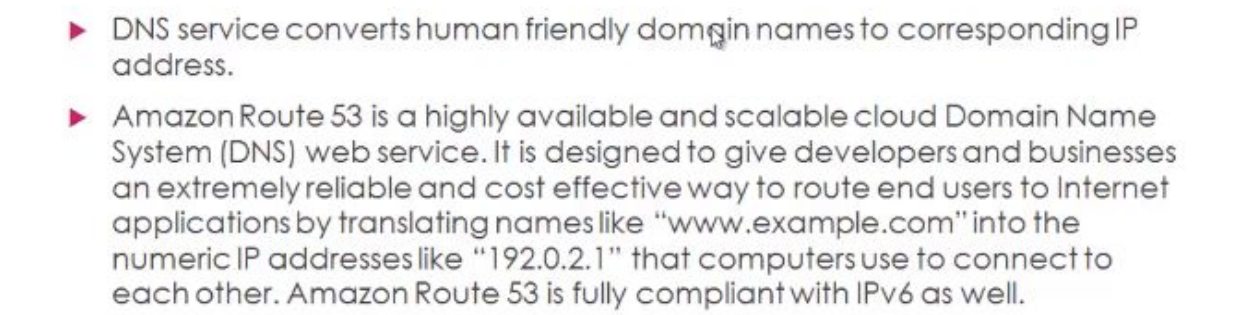
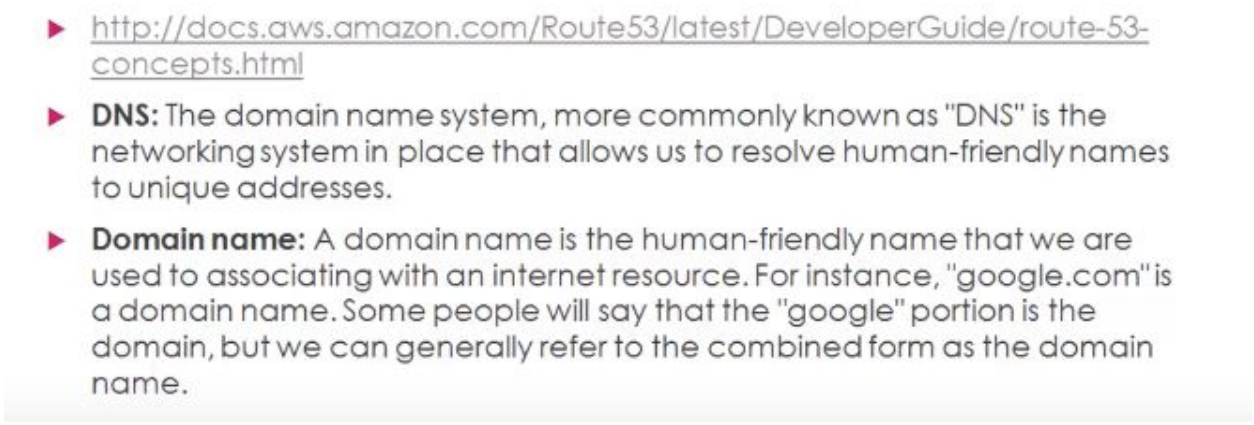
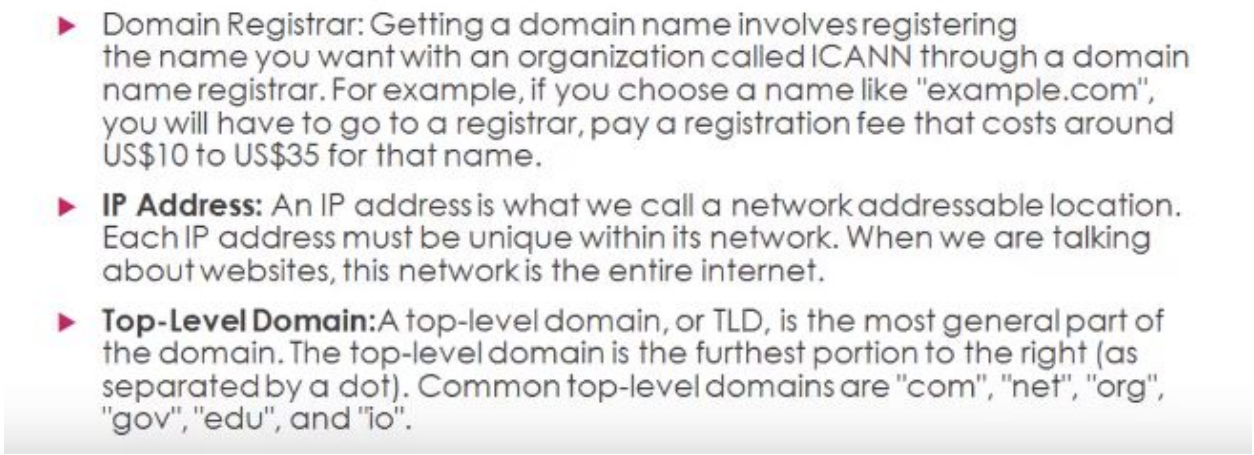
What is Route53?

- Route 53 is a domain management service (DNS hosting solution) provided by AWS
- Route53 is commonly used with an ELB to direct traffic from the domain to the instances (and thus have traffic evenly distributed among servers running your applications.







- **Hosts:** Within a domain, the domain owner can define individual hosts, which refer to separate computers or services accessible through a domain. For instance, most domain owners make their web servers accessible through the bare domain (example.com) and also through the "host" definition "www" (www.example.com).
- **Name Server:** A name server is a computer designated to translate domain names into IP addresses. These servers do most of the work in the DNS system. Since the total number of domain translations is too much for any one server, each server may redirect request to other name servers or delegate responsibility for a subset of subdomains they are responsible for.

- **Zone File:** A zone file is a simple text file that contains the mappings between domain names and IP addresses. This is how the DNS system finally finds out which IP address should be contacted when a user requests a certain domain name.
- **Records:** Within a zone file, records are kept. In its simplest form, a record is basically a single mapping between a resource and a name. These can map a domain name to an IP address, define the name servers for the domain, define the mail servers for the domain, etc.

- **SOA Records:** The Start of Authority, or SOA, record is a mandatory record in all zone files. It must be the first real record in a file (although $ORIGIN or $TTL specifications may appear above). It is also one of the most complex to understand.
- **A and AAAA Records:**Both of these records map a host to an IP address. The "A" record is used to map a host to an IPv4 IP address, while "AAAA" records are used to map a host to an IPv6 address.
- **CNAME Records:** CNAME records define an alias for canonical name for your server (one defined by an A or AAAA record).

- **MX Records:** MX records are used to define the mail exchanges that are used for the domain. This helps email messages arrive at your mail server correctly.
- **NS Records:** This record type defines the name servers that are used for this zone.
- Hosted Zone, Alias Records,
- **Zone apex record:** An apex record is one at the root of a DNS zone. Sometimes called "naked domains".

Route 53 Main functions

Key features of Route53

1) Domain Registration
2) DNS service
3) Health Checking

- **Register domain names** – Your website needs a name, such as example.com. Amazon Route 53 lets you register a name for your website or web application, known as a *domain name*. For an overview, see Registering Domain Names.
- **Route Internet traffic to the resources for your domain** – When a user opens a web browser and enters your domain name in the address bar, Amazon Route 53 helps the Domain Name System (DNS) connect the browser with your website or web application. For an overview, see Routing Internet Traffic to Your Website or Web Application.
- **Check the health of your resources** – Amazon Route 53 sends automated requests over the Internet to a resource, such as a web server, to verify that it's reachable, available, and functional. You also can choose to receive notifications when a resource becomes unavailable and choose to route Internet traffic away from unhealthy resources. For an overview, see How Amazon Route 53 Checks the Health of Your Resources.

Mandatory Requirements

- Registered Domain name
- Instances/Resources to host your service/Application
- DNS service equipped with routing policy/protocol to suite your needs.

Types of Routing Policy

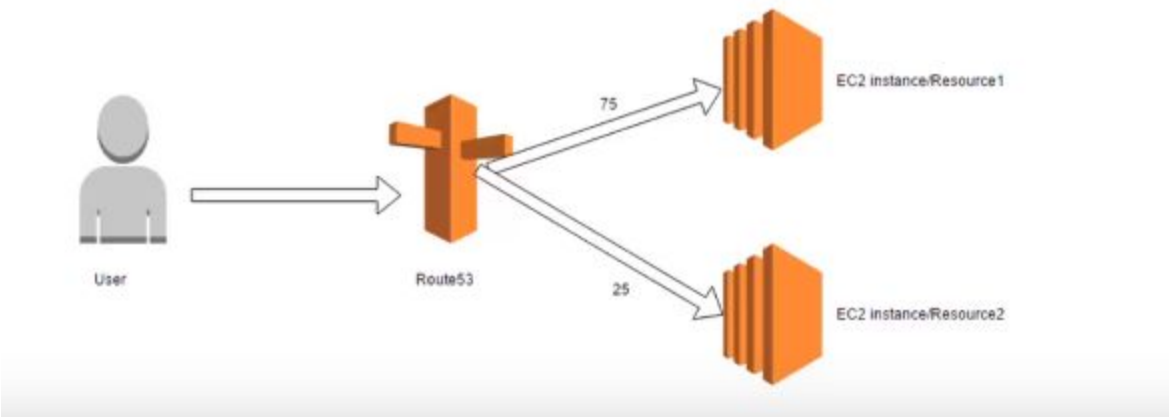
1. Simple
2. Weighted
3. Latency-Based
4. Failover
5. Geo location

http://docs.aws.amazon.com/Route53/latest/DeveloperGuide/routing-policy.html

Simple
Route all traffic to one end point

Weighted:
Route traffic to multiple endpoints (manual Load balancing)

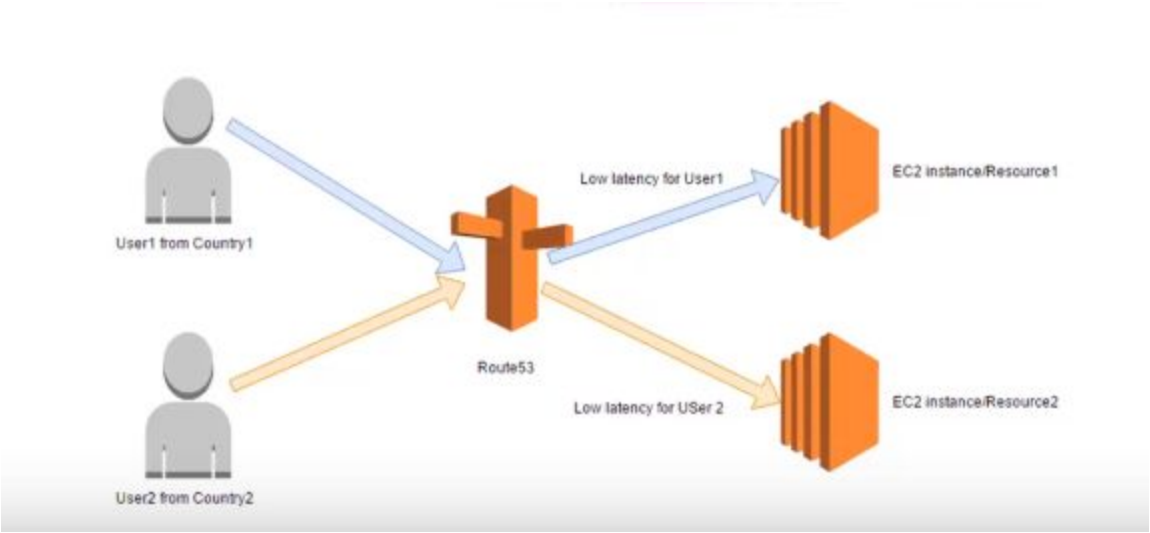
- Use the weighted routing policy when you have multiple resources that perform the same function (for example, web servers that serve the same website) and you want Amazon Route 53 to route traffic to those resources in proportions that you specify (for example, one quarter to one server and three quarters to the other). For more information about weighted resource record sets.



Latency Routing

Route traffic to an endpoint based on the users latency to various endpoints.

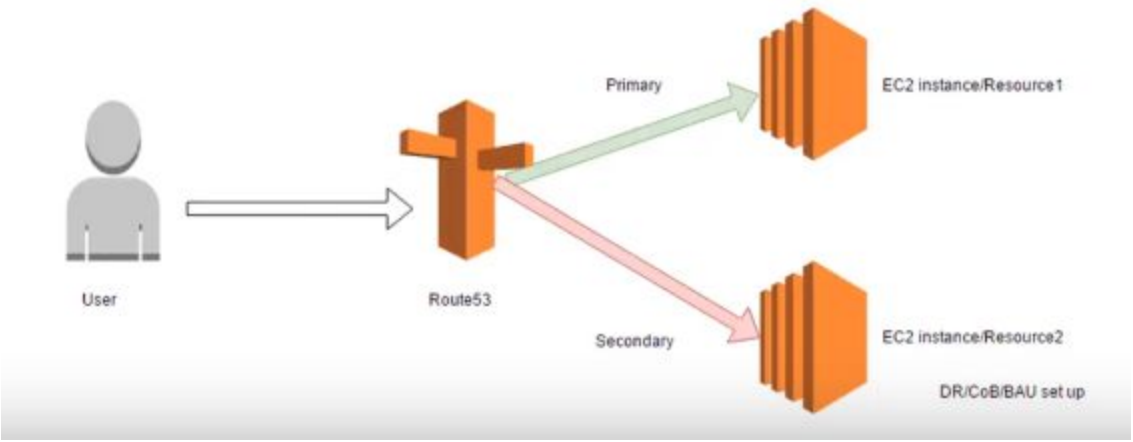
- Use the latency routing policy when you have resources in multiple Amazon EC2 data centers that perform the same function and you want Amazon Route 53 to respond to DNS queries with the resources that provide the best latency. For example, you might have web servers for example.com in the Amazon EC2 data centers in Ireland and in Tokyo. When a user browses to example.com, Amazon Route 53 chooses to respond to the DNS query based on which data center gives your user the lowest latency. For more information about latency resource record sets.



Failover routing

Route traffic to a “secondary” endpoint if the “primary” endpoint is unavailable

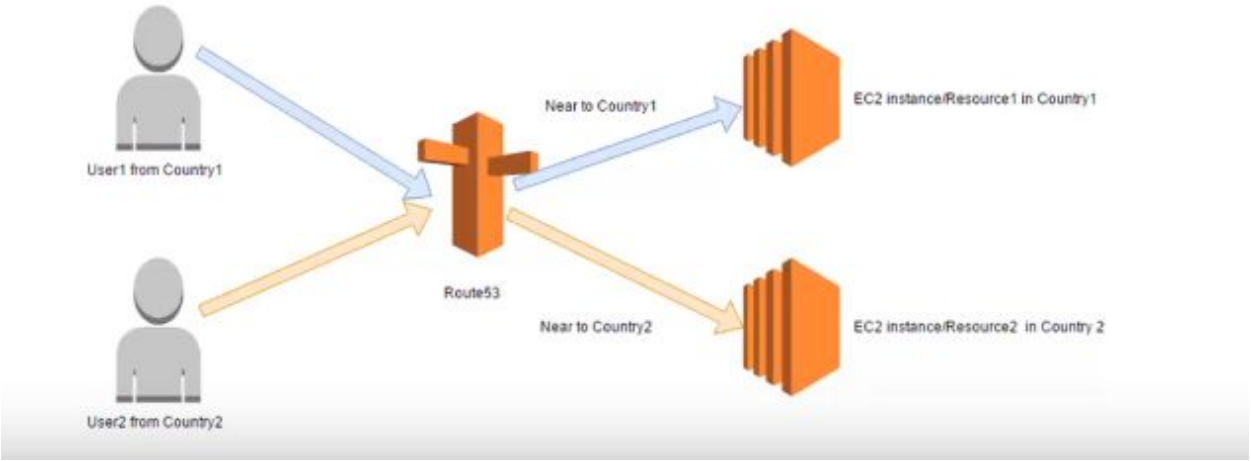
- Use the failover routing policy when you want to configure active-passive failover, in which one resource takes all traffic when it's available and the other resource takes all traffic when the first resource isn't available. For more information about failover resource record sets, see Configuring Active-Passive Failover by Using Amazon Route 53 Failover and Failover Alias Resource Record Sets. For information about creating failover resource record sets in a private hosted zone.



Geo-Location ROuting

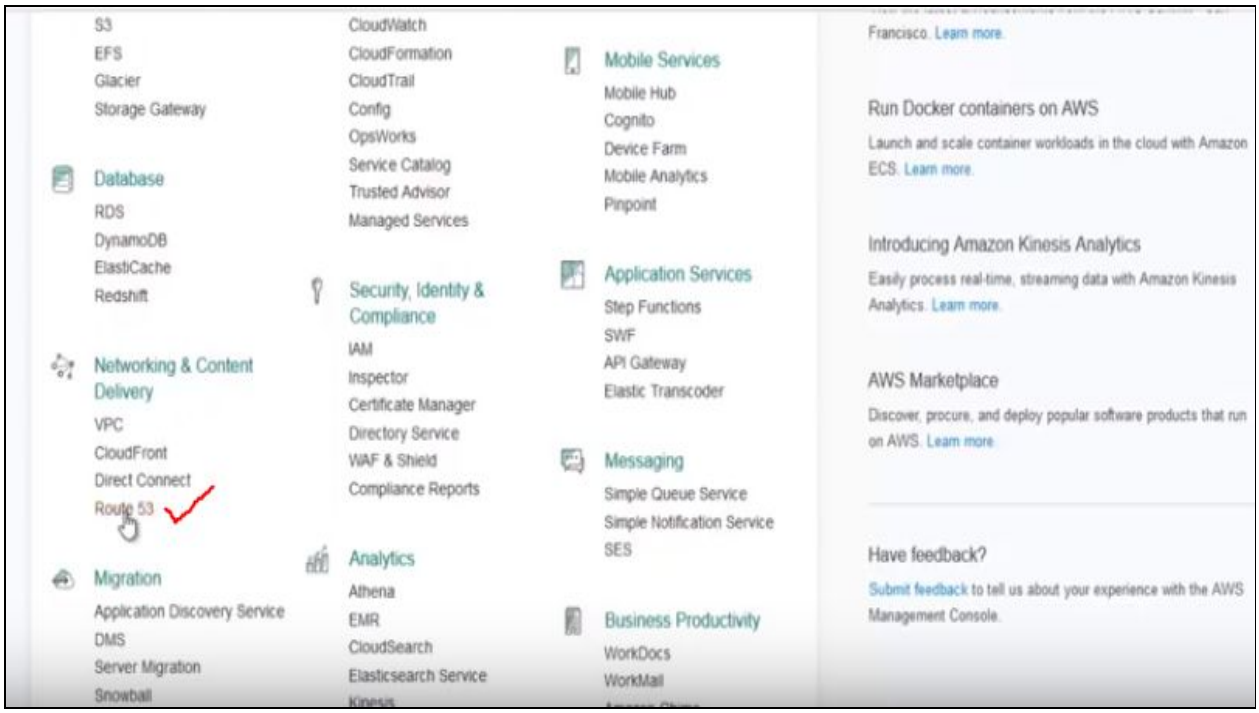
Route traffic to an endpoint based on the geographical location of the user

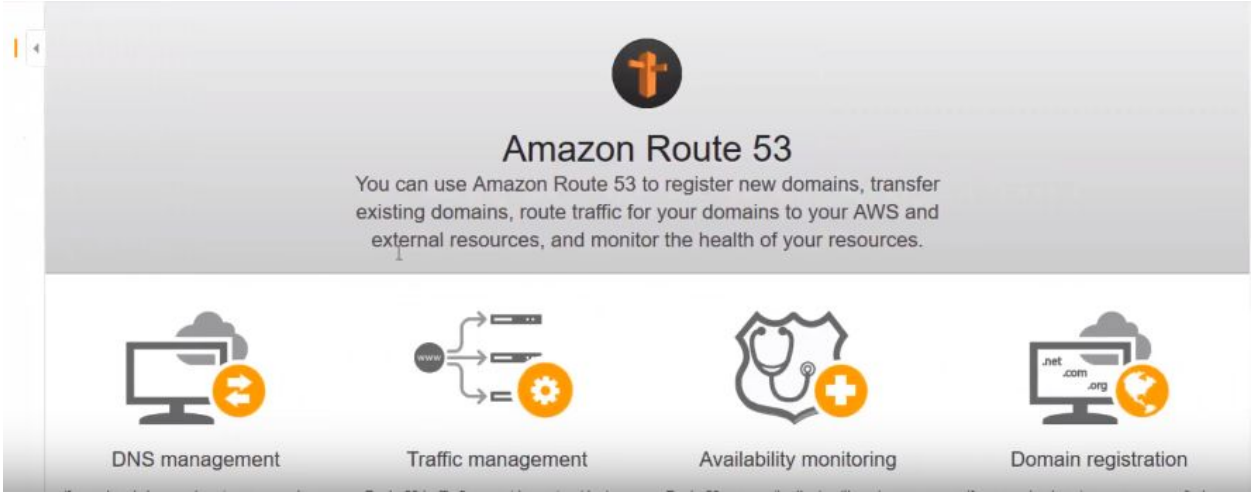
- Use the geolocation routing policy when you want Amazon Route 53 to respond to DNS queries based on the location of your users. For more information about geolocation resource record sets.



LAB:

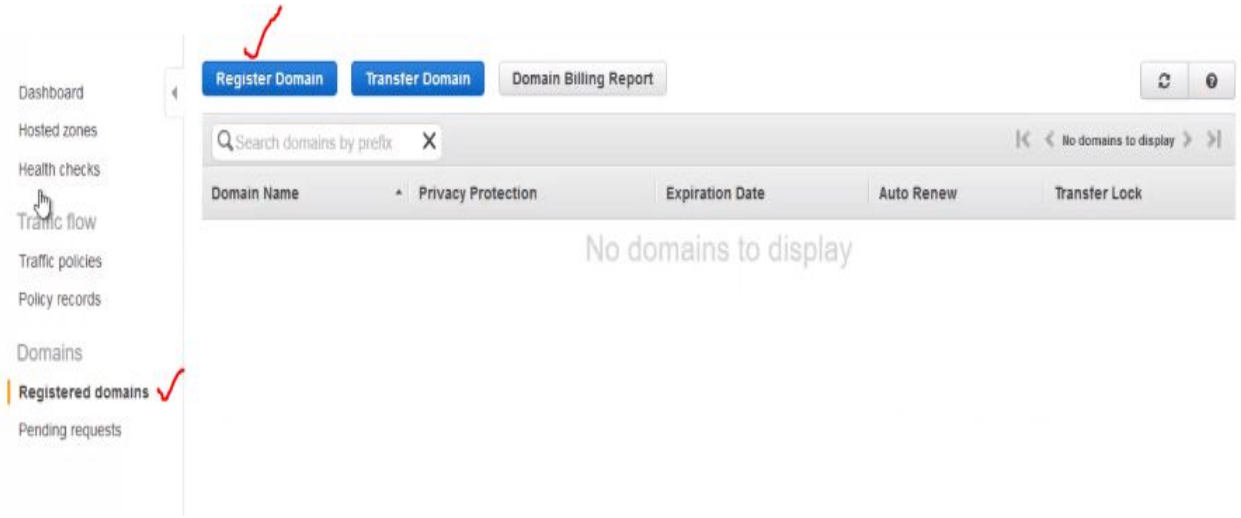
First login to AWS Management COnsole and Click on ROute 53 in Network and Delivery Section



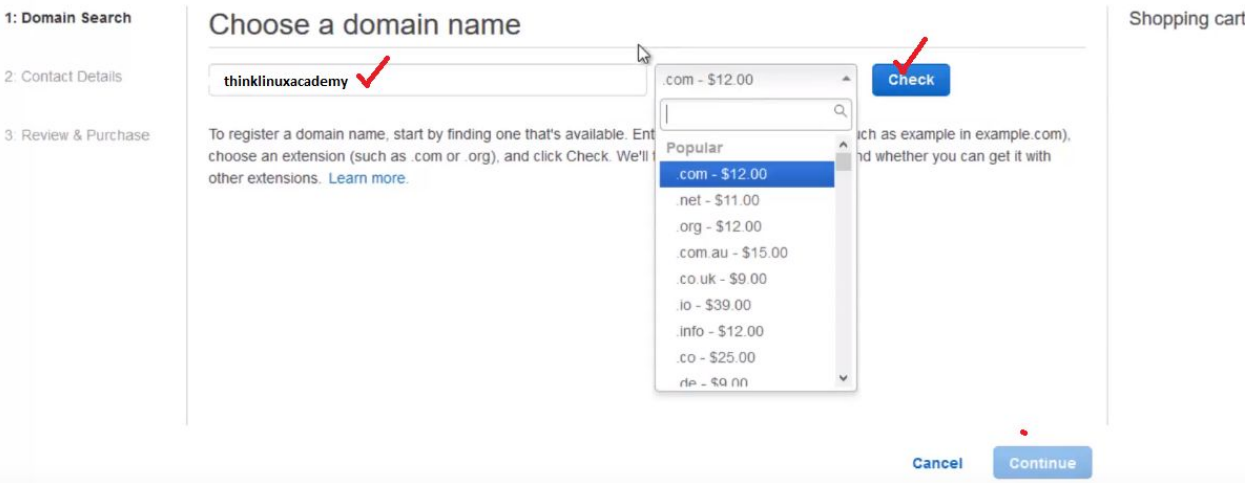


First Time if we are using then we will see this screen, So click as below

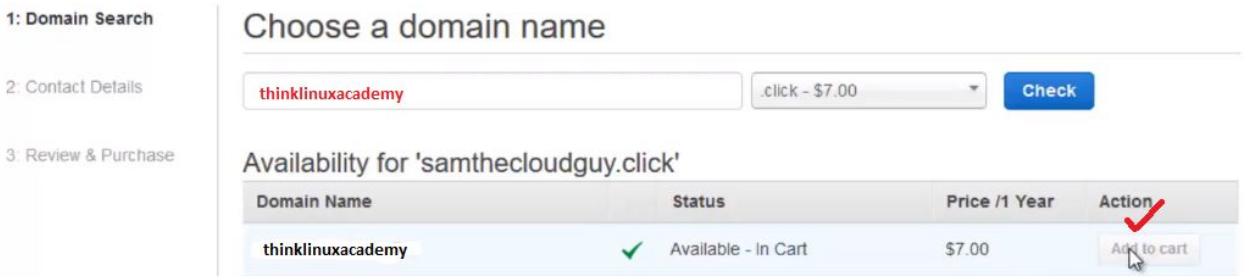
TASK1: Register your Domain with AWS Registrar



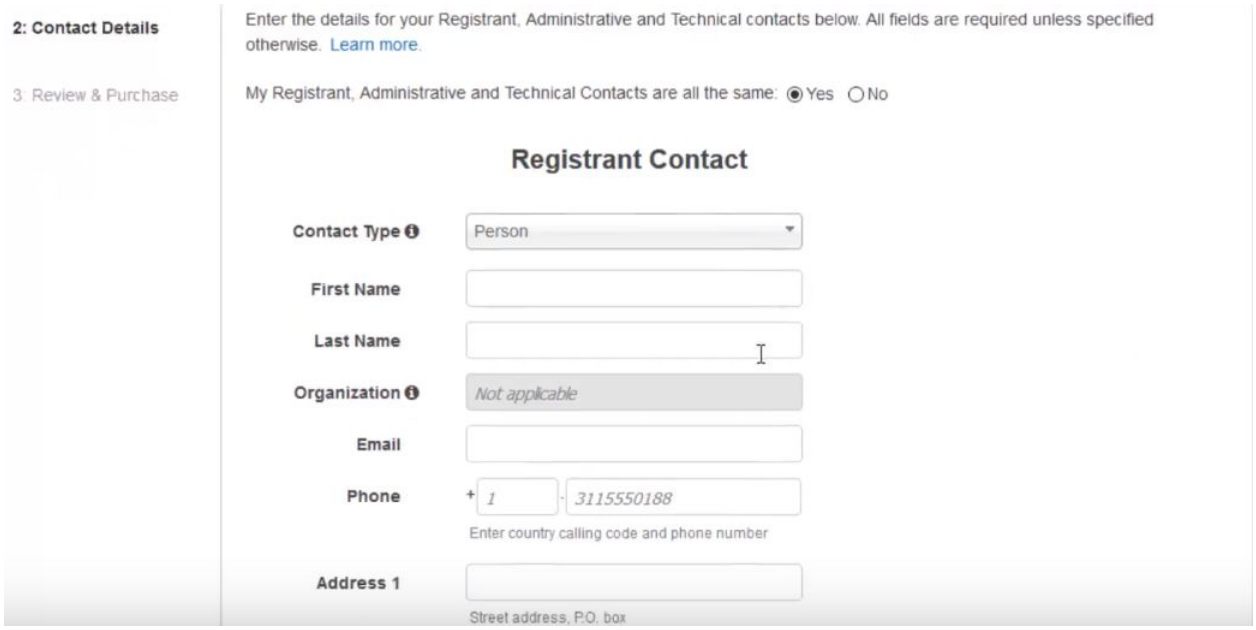
Check your domain is available in registrar or not



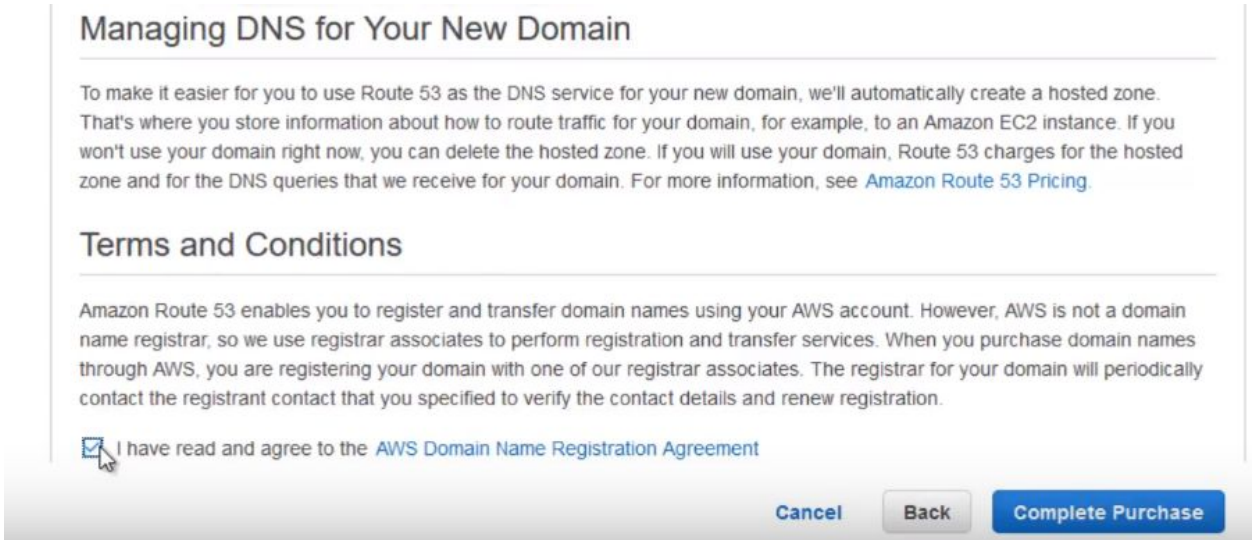
If avaialable, you can register



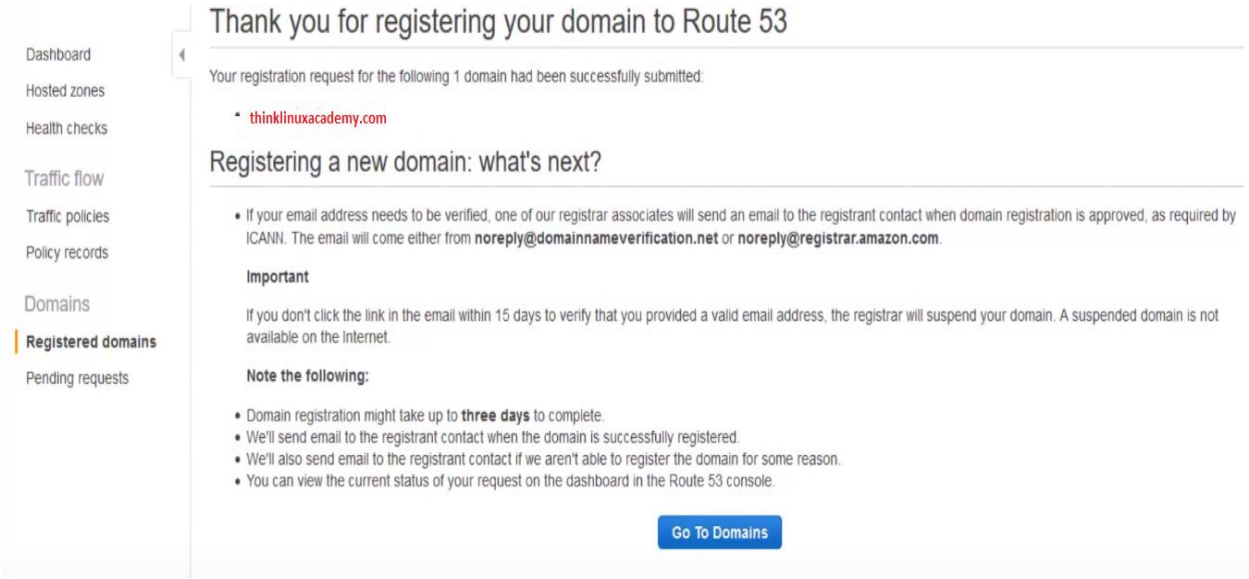
Enter your contact Details



Clock on Continue

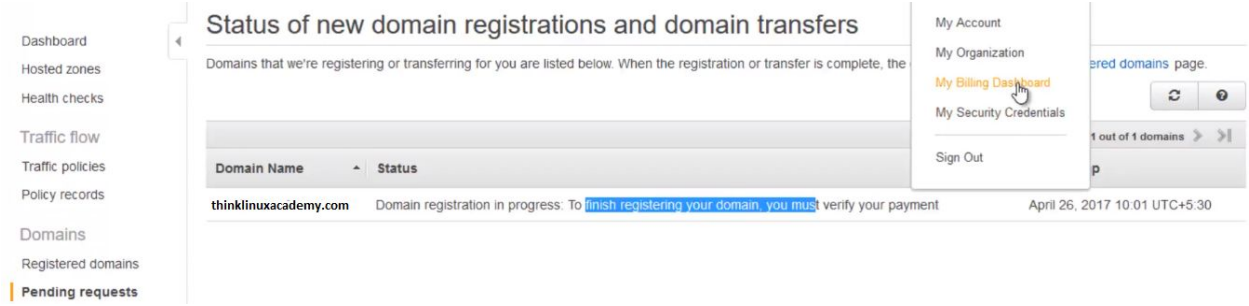


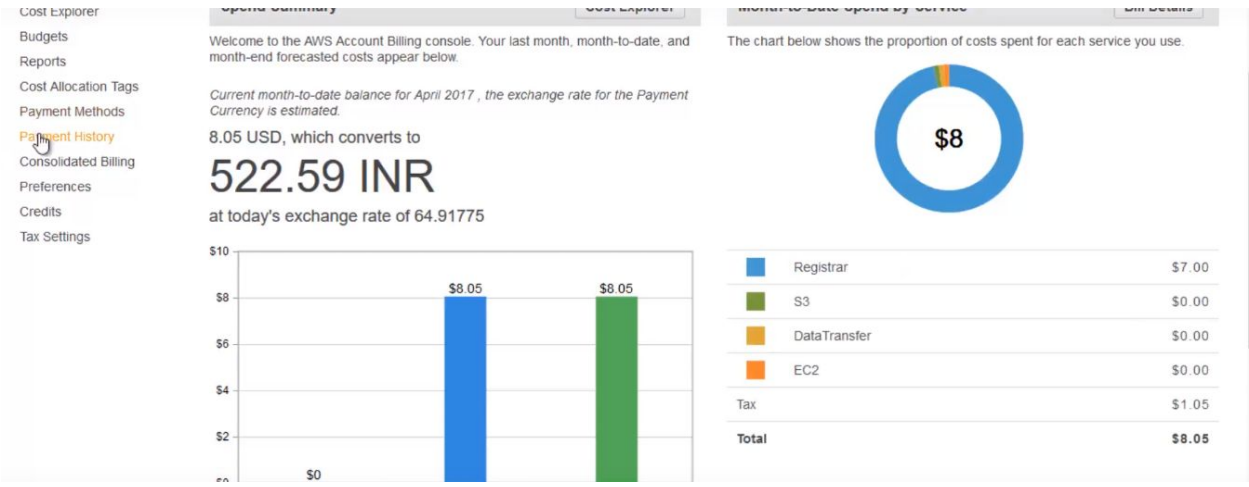
Agree to terms and condition and click on Complete Purchase



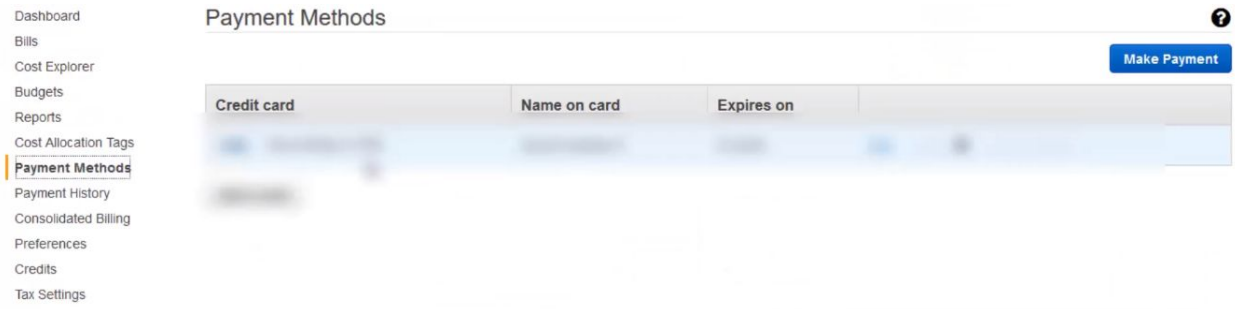
Completed but you have to wait

Go to Payments Section and pay the money

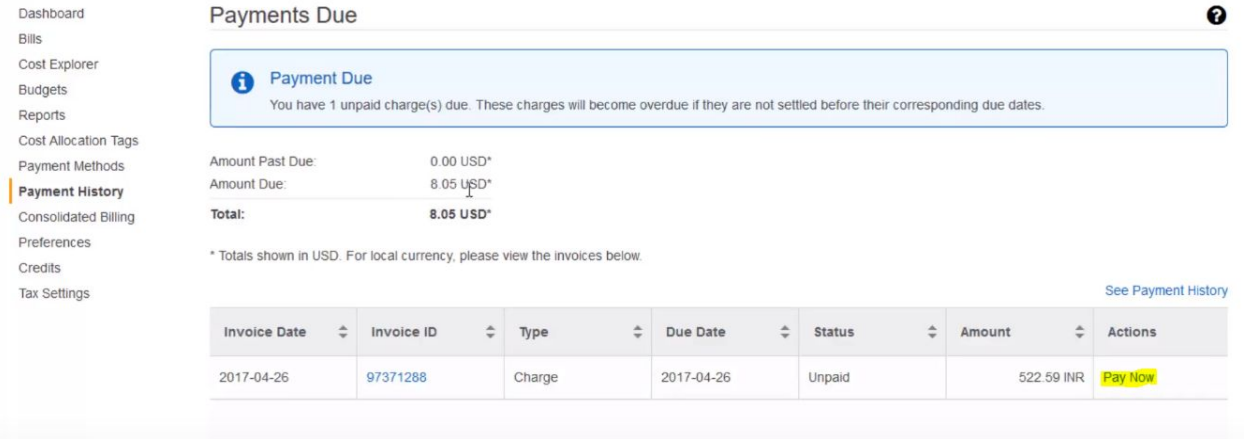




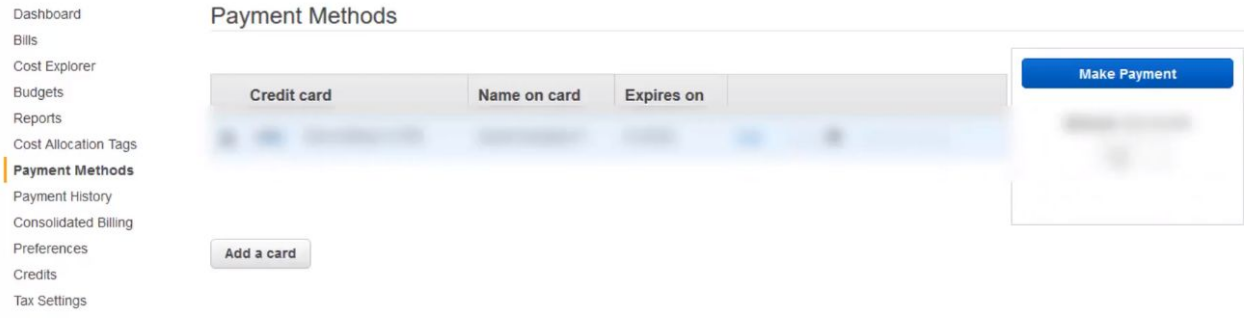
Go to Payment Method



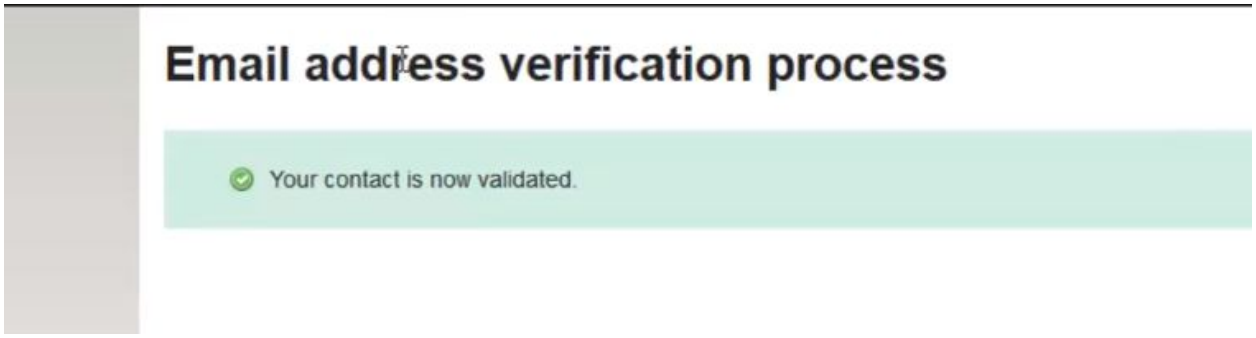
Click on Make Payment



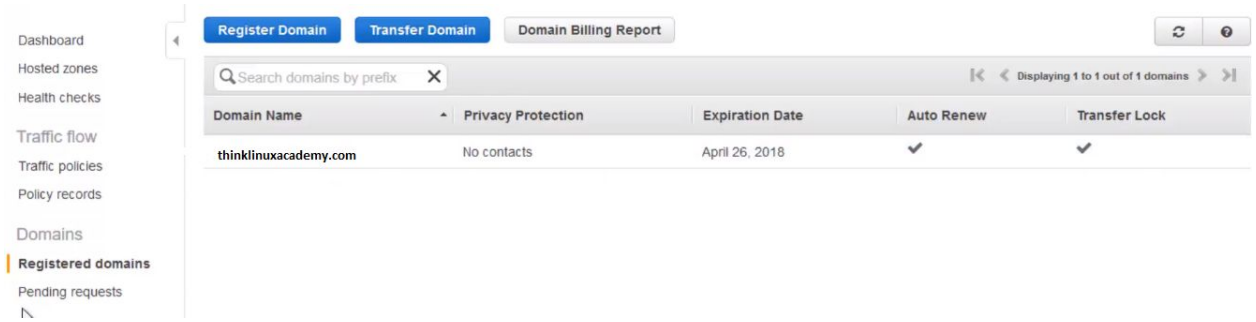
Pay Now



Verify you Email address also



Verify now in Register Domain section

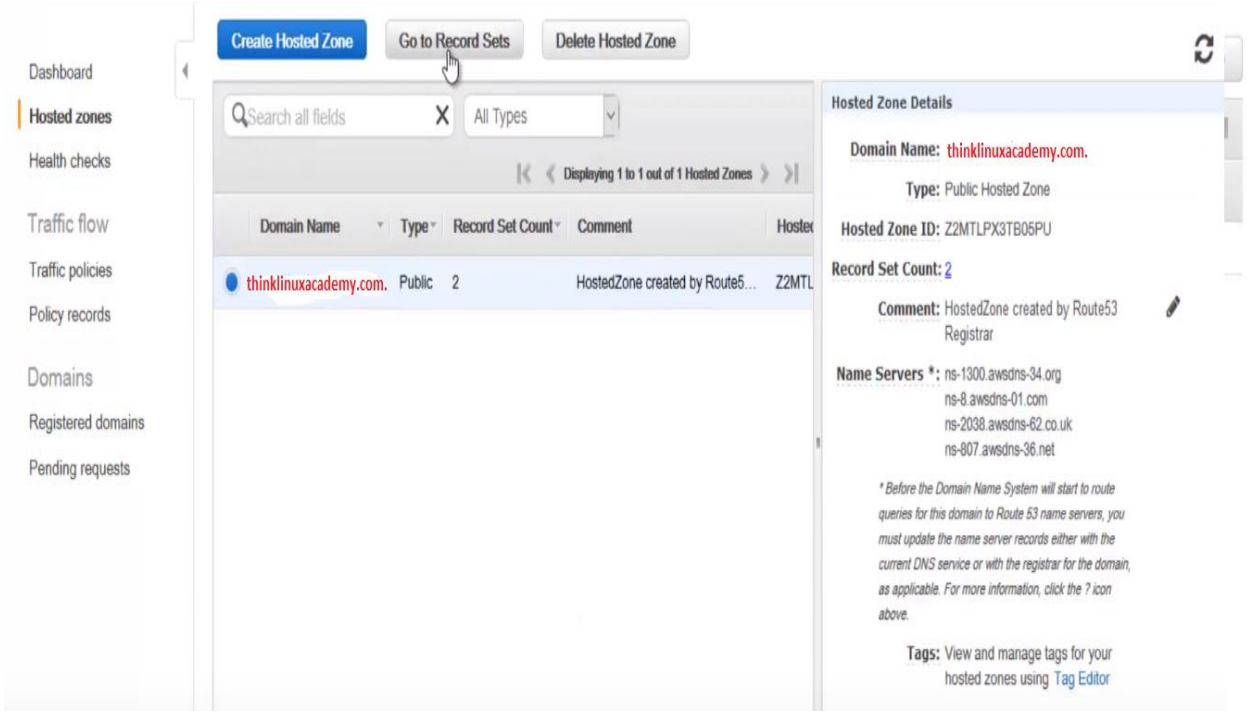


HOSTED ZONE:

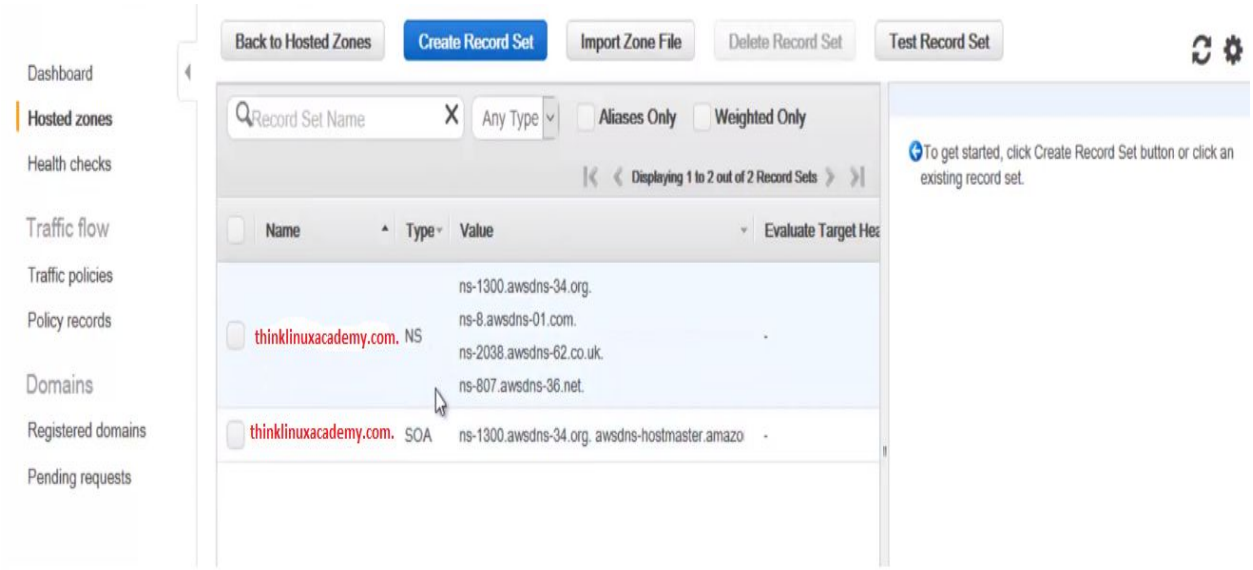
- A Hosted Zones stores DNS records for your domain
- Basically, it contains all the rules (record sets) that tells Route 53 what to do with DNS request.
- There are both public and private hosted zones
    - A Public Hosted Zone is a container that holds information about how you want to route traffic on the internet for a domain, such as thinklinuxacadmey.com, and its subdomains.

- A Private Hosted Zone is a container that holds information about how you want to route traffic for a domain and its subdomains within one or more Amazon Virtual Private Clouds.

- After you create a Hosted Zone for your domain like thinklinuxacademy.com, you create resource record sets to tell the Domain Name System (DNS) how you want traffic to be routed for that domain
- By default, Hosted zone will be pre-populated with NS (name server) and SOA (start of authority) record sets.



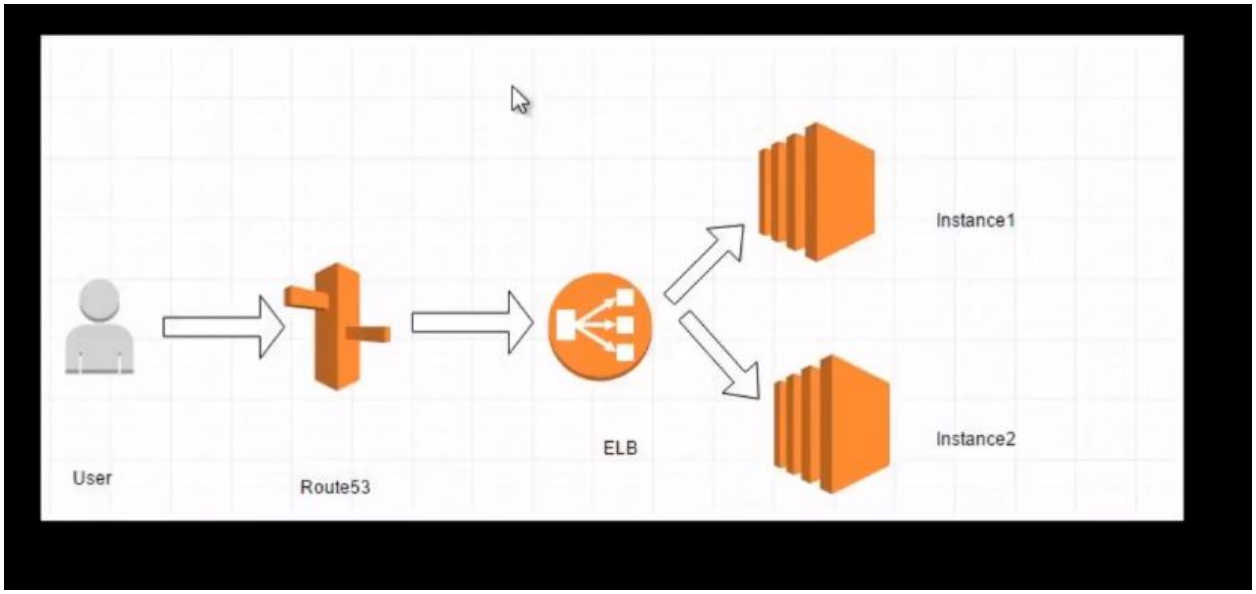
Click on Go to Record Sets



There were 2 record set already available

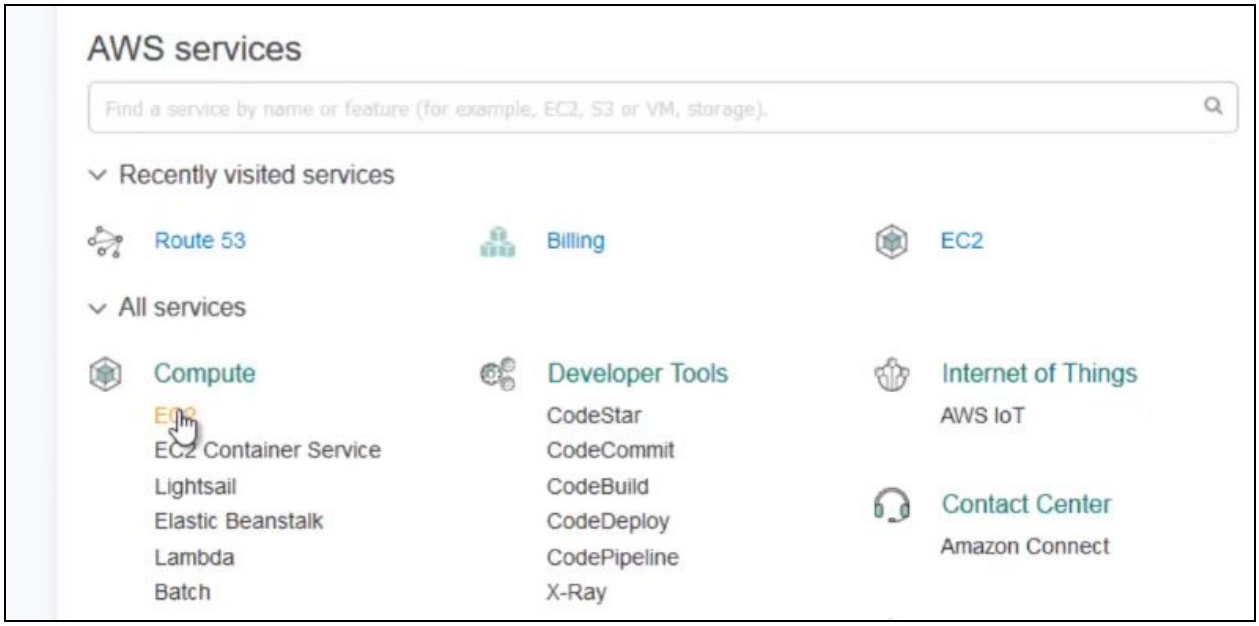
1) NS
2) SOA

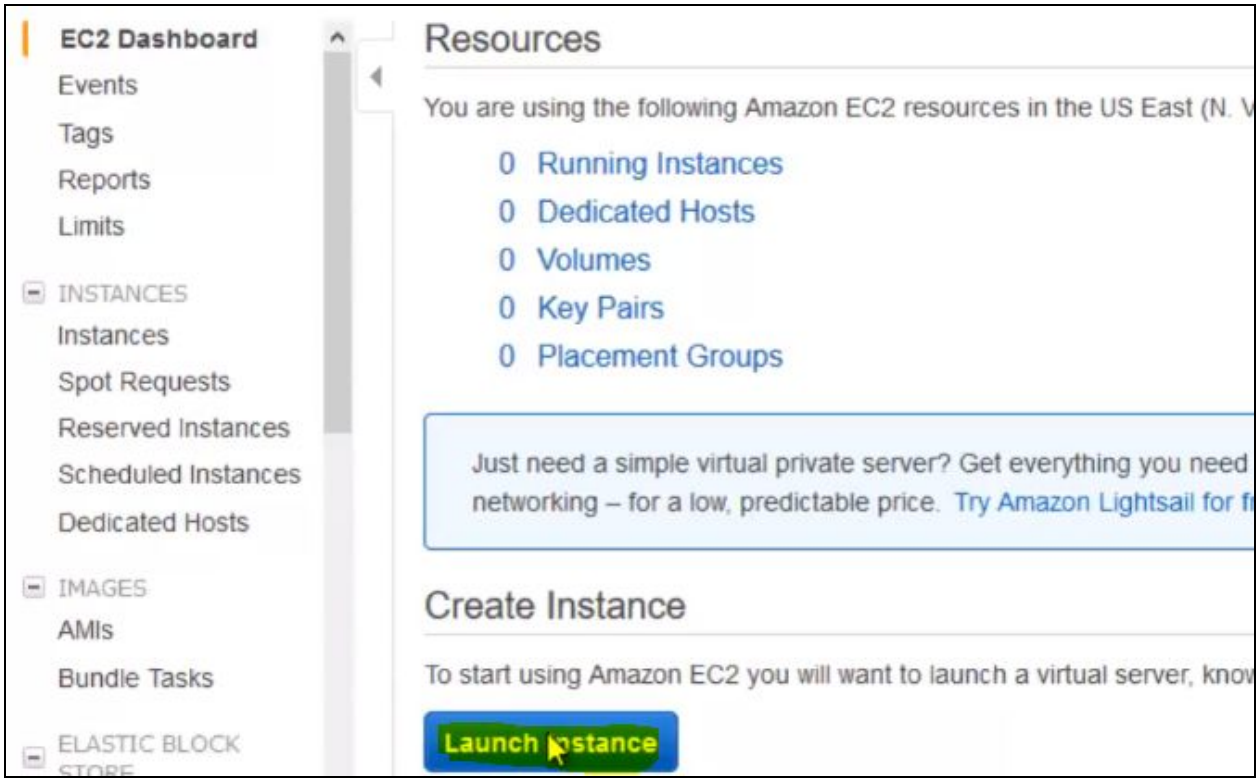
Before going Furthur, We will implement this below diagram in our Route53 Practical

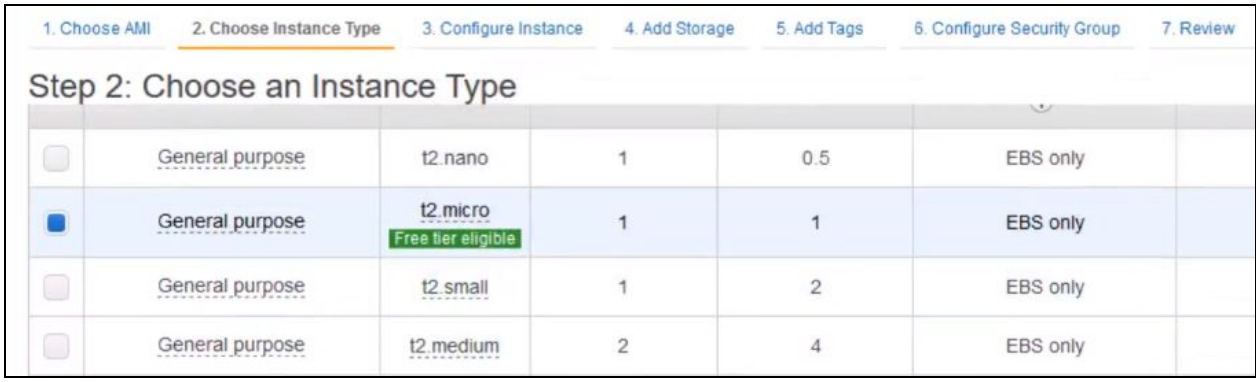


SO whenever the user , type the thinklinuxacademy.com , Route53 comes into action and it reaches the ELB and behind ELB we have 2 EC2 instances who is serving the page.

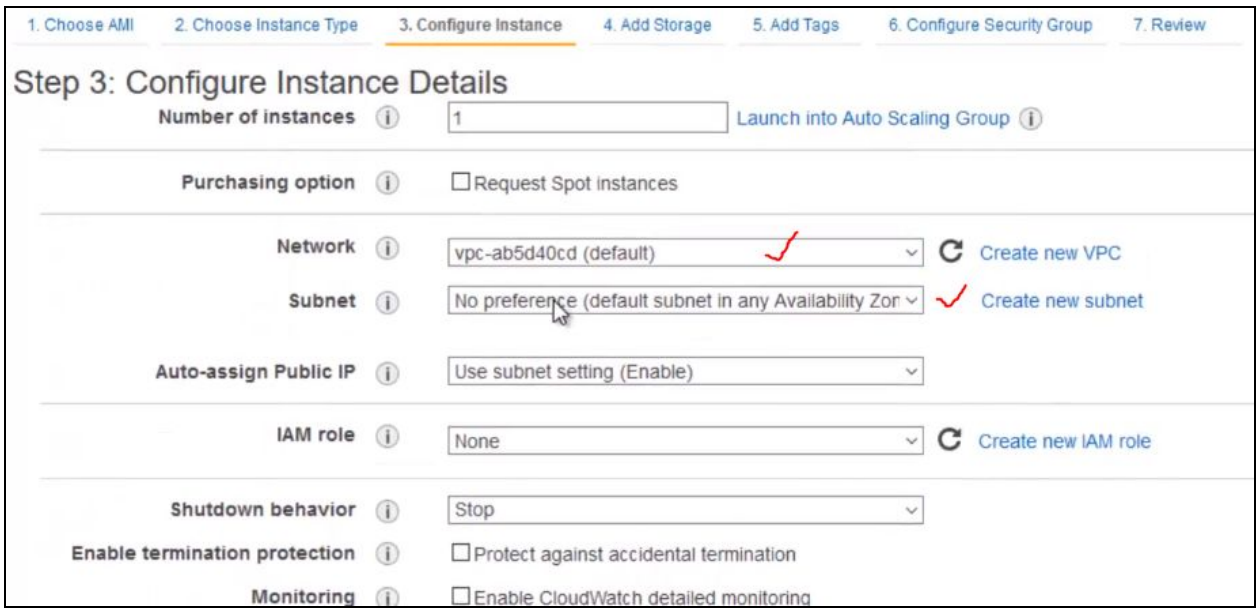
First Create 2 EC2 instance
Go to EC2 Dashboard







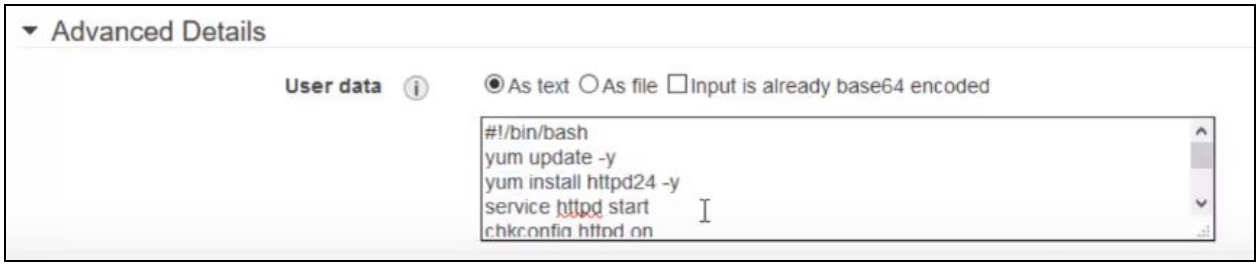
Leave with defaults settings

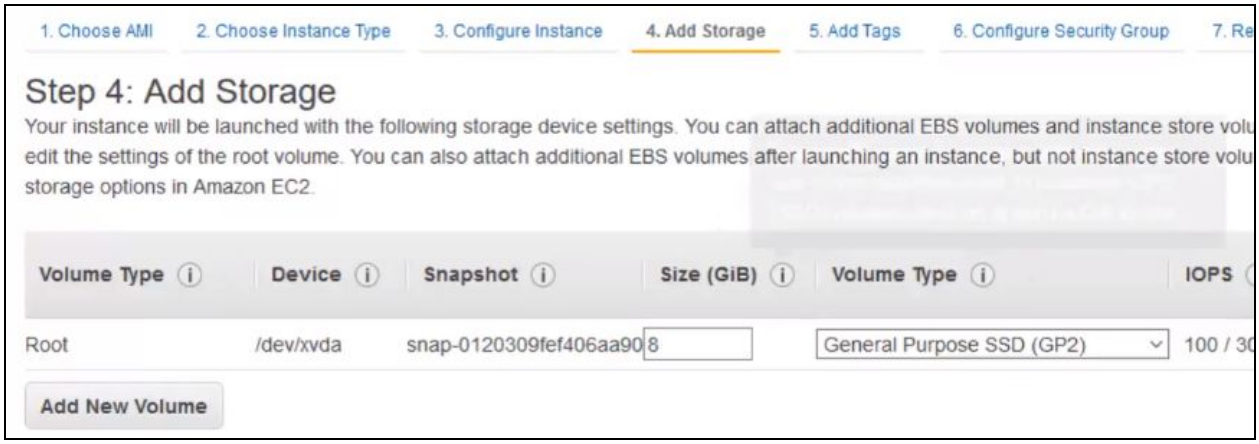


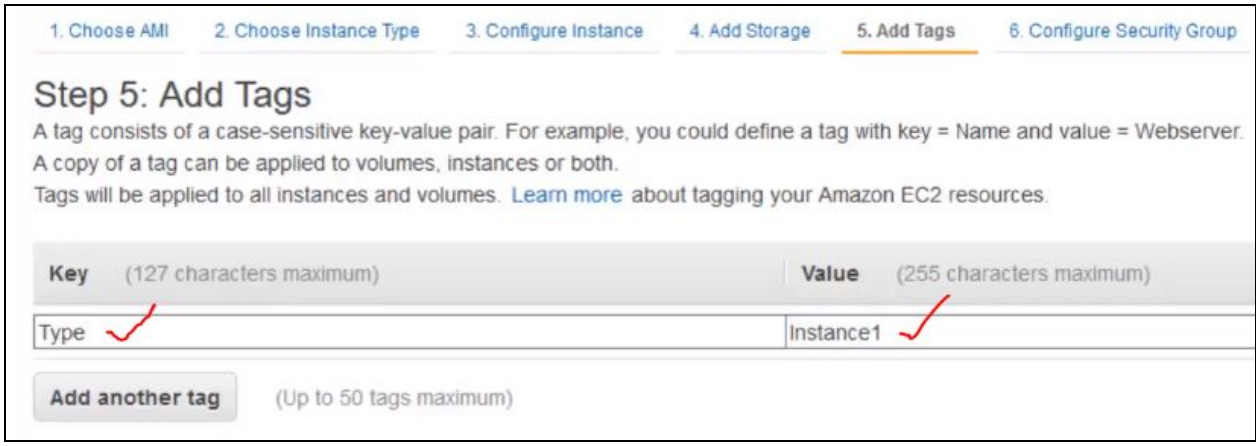
In advance Details, put a boot script which will create a website and installation of httpd server.

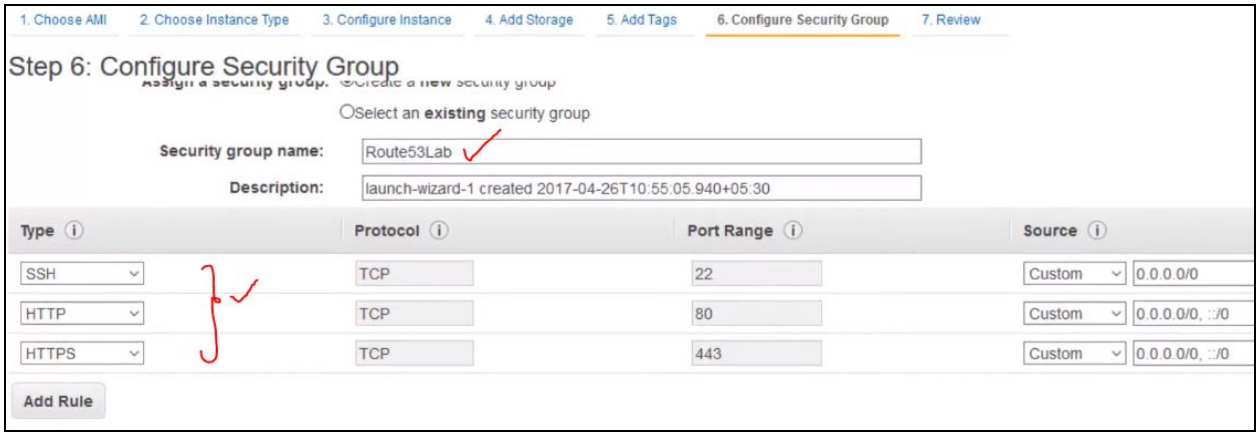
#!/bin/bash
yum update -y
yum install httpd24 -y

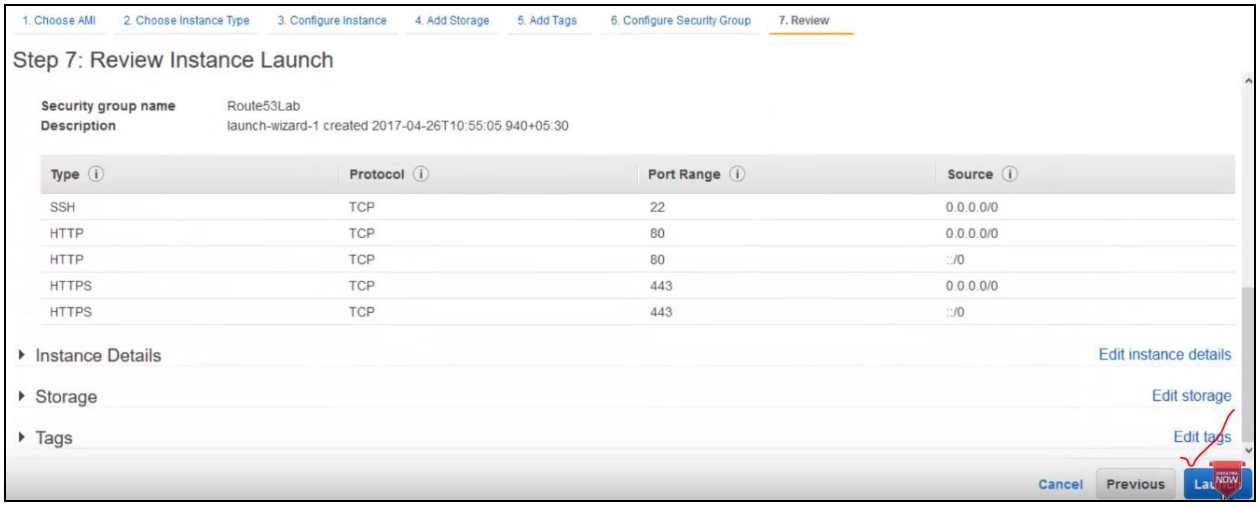
Service httpd start
chkconfig httpd on
echo “<html><body><h1>Welcome to TLA from Instance1 </h1></body></html>” >
/var/www/html/index.html



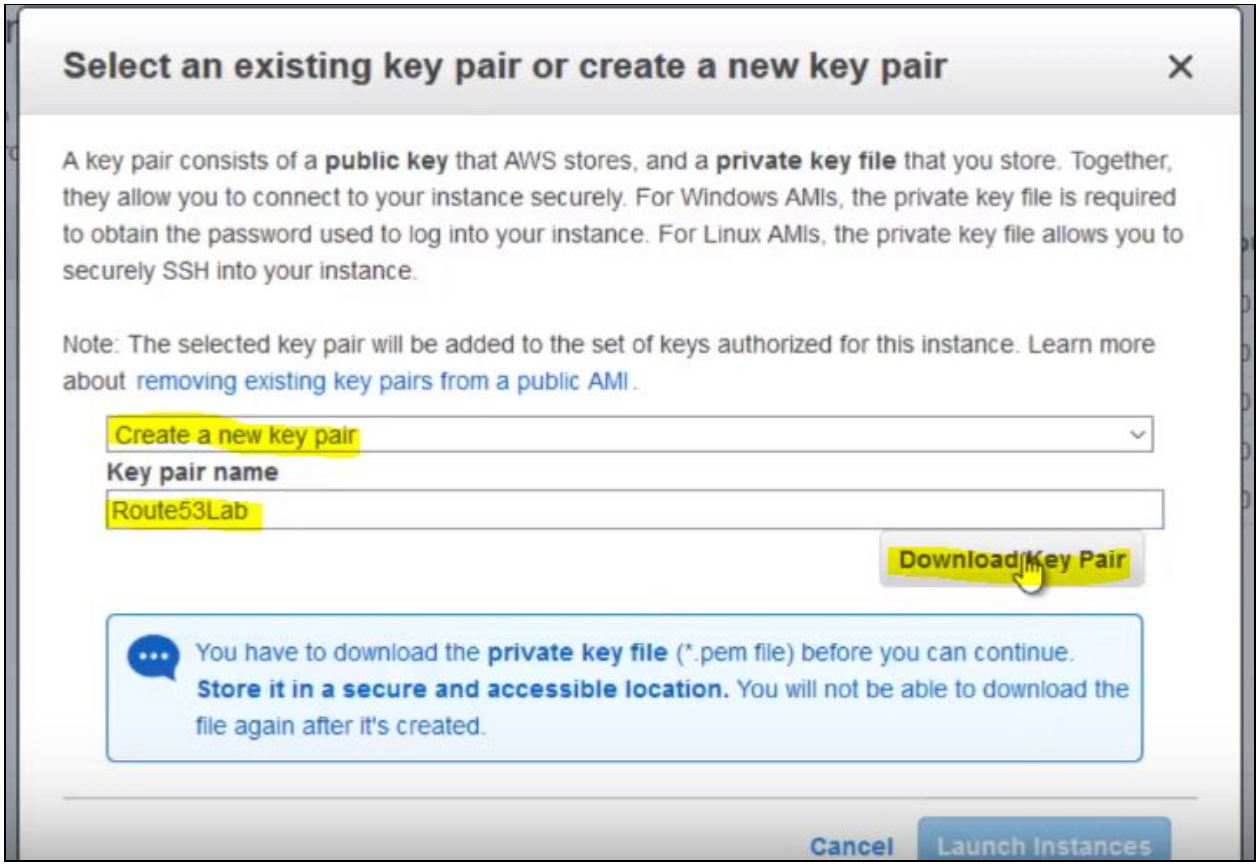




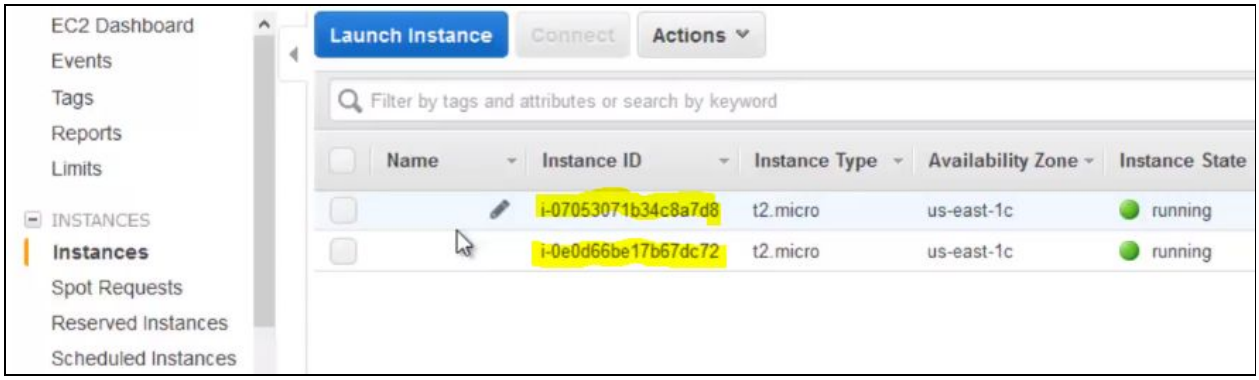




Now launch it



Similarly Launch one more instance



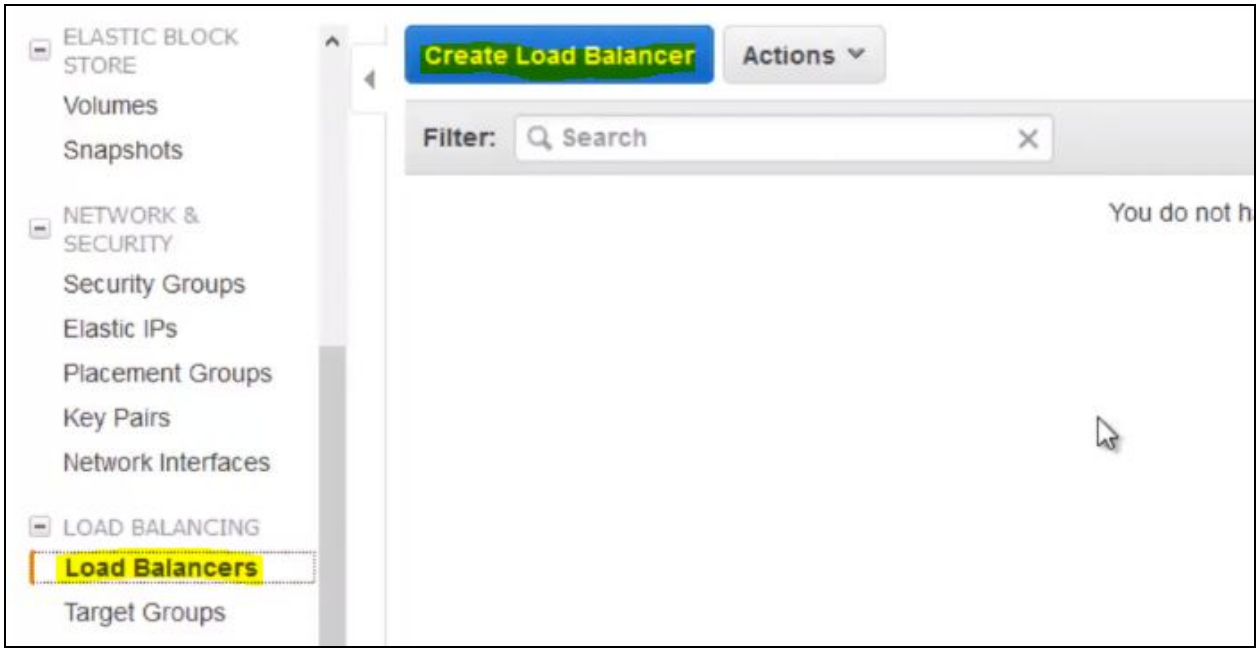
See you can see 2 instances. Make sure httpd is running in both of them with similar boot script (index.html pages)

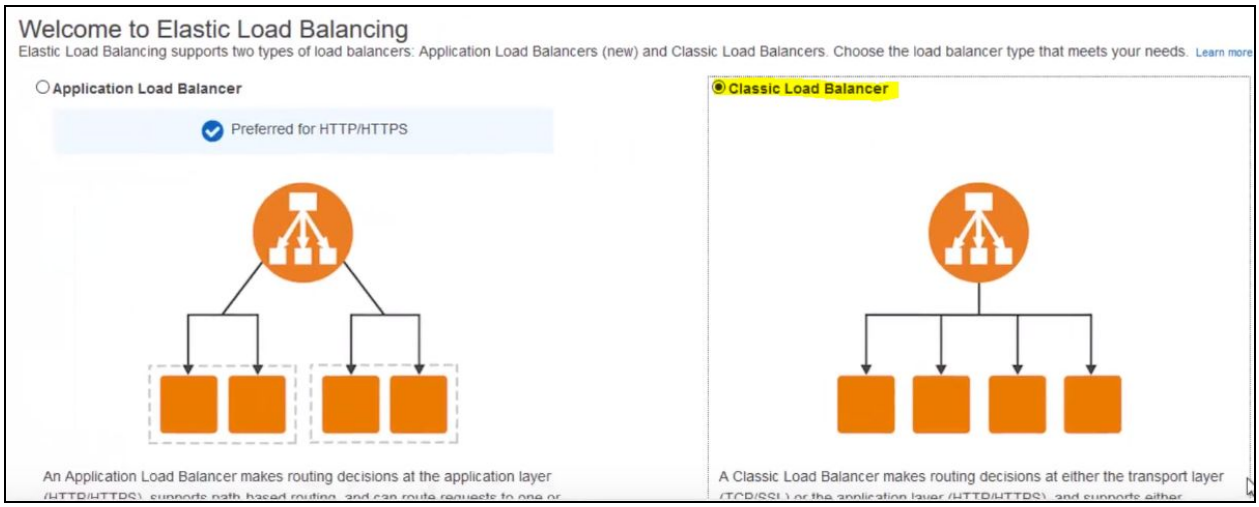
You can also verify, by copy and paste the public DNS names of EC2 instance in browser.

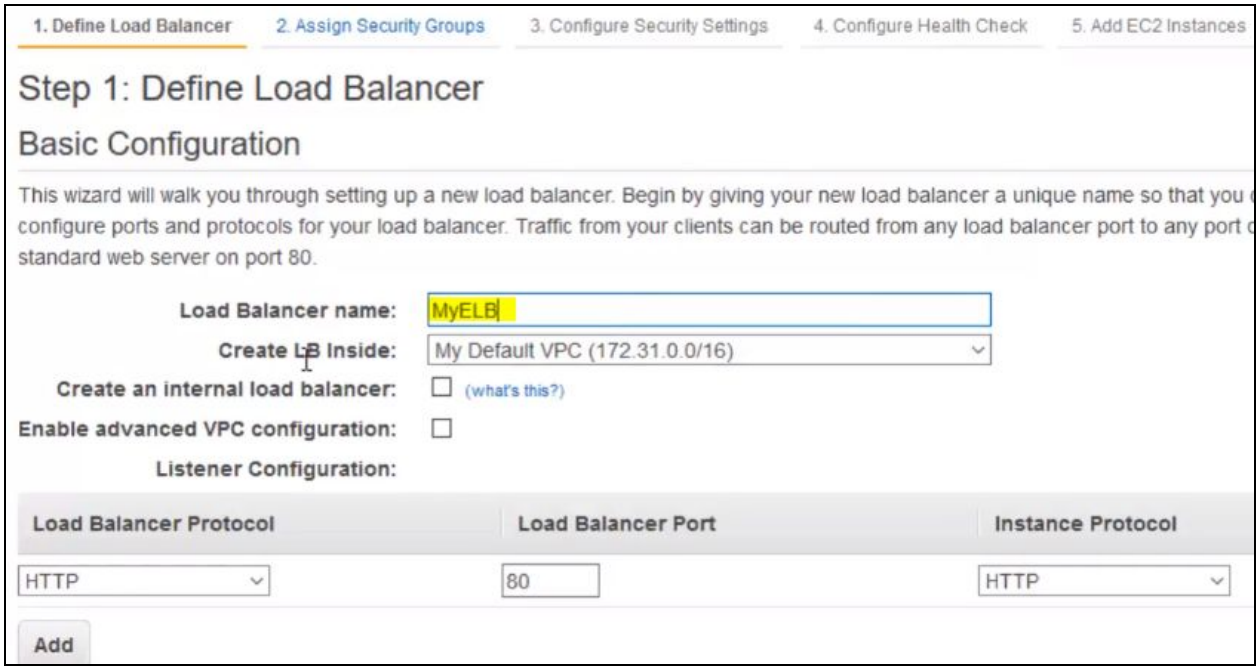

ec2-54-146-169-131.compute-1.amazonaws.com
Search
Welcome to TLA from Instance1

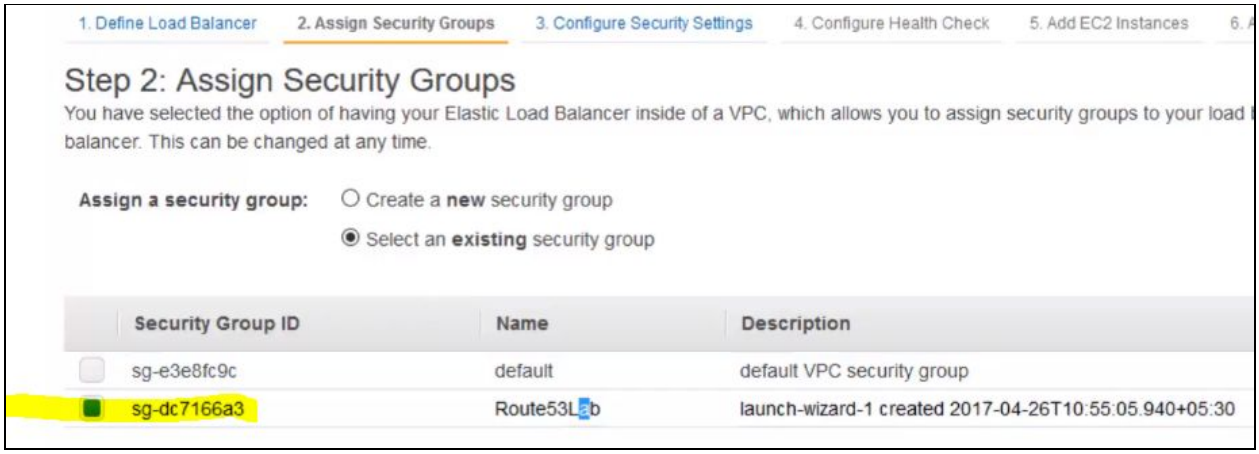

ec2-54-146-169-112.compute-1.amazonaws.com
Welcome to TLA from Instance2

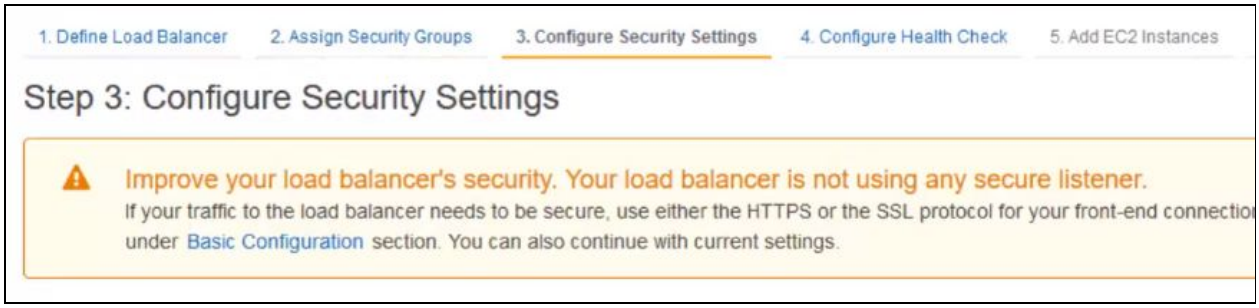
Create a ELB:

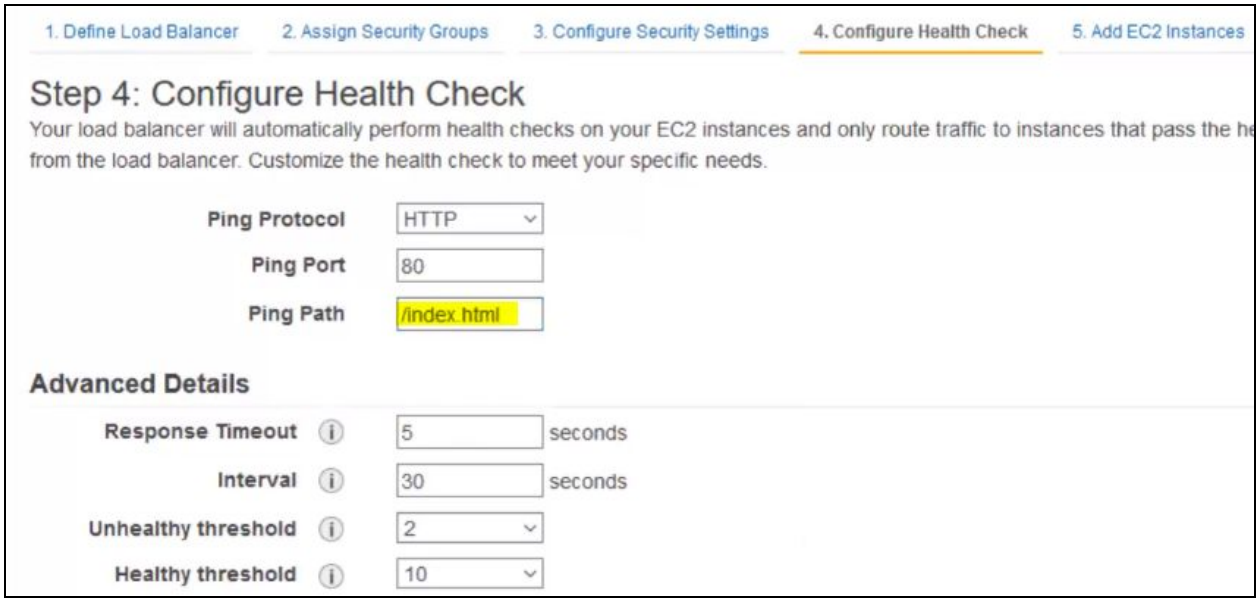

ELASTIC BLOCK STORE
Volumes
Snapshots
NETWORK & SECURITY
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
LOAD BALANCING
Load Balancers
Target Groups
Create Load Balancer
Actions
Filter:
Search
You do not h


Welcome to Elastic Load Balancing
Elastic Load Balancing supports two types of load balancers: Application Load Balancers (new) and Classic Load Balancers. Choose the load balancer type that meets your needs. Learn more
Application Load Balancer
Preferred for HTTP/HTTPS
An Application Load Balancer makes routing decisions at the application layer
Classic Load Balancer
A Classic Load Balancer makes routing decisions at either the transport layer

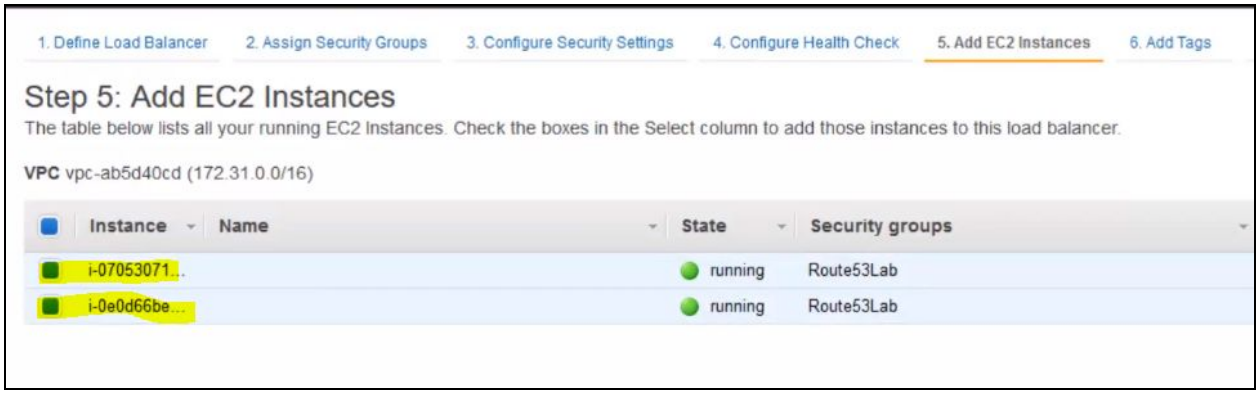

1. Define Load Balancer
2. Assign Security Groups
3. Configure Security Settings
4. Configure Health Check
5. Add EC2 Instances
Step 1: Define Load Balancer
Basic Configuration
This wizard will walk you through setting up a new load balancer. Begin by giving your new load balancer a unique name so that you
configure ports and protocols for your load balancer. Traffic from your clients can be routed from any load balancer port to any port
standard web server on port 80.
Load Balancer name:
MyELB
Create LB Inside:
My Default VPC (172.31.0.0/16)
Create an internal load balancer:
(what's this?)
Enable advanced VPC configuration:
Listener Configuration:
Load Balancer Protocol
Load Balancer Port
Instance Protocol
HTTP
80
HTTP
Add

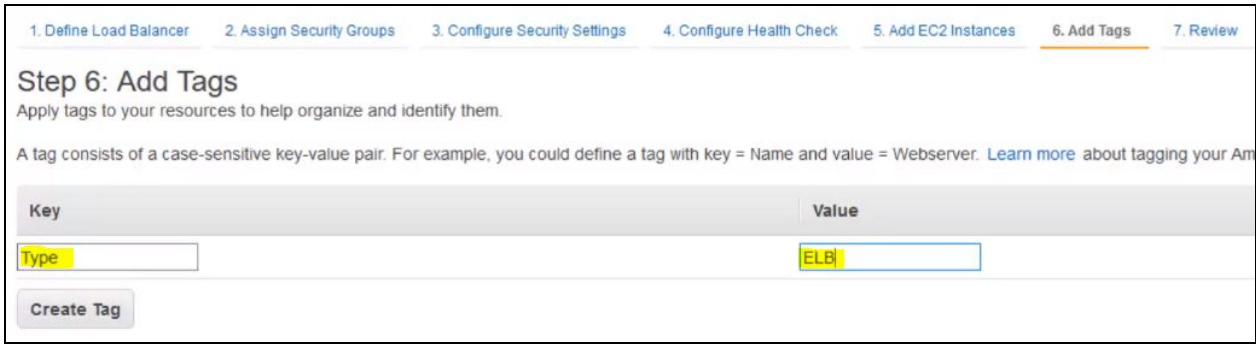

1. Define Load Balancer
2. Assign Security Groups
3. Configure Security Settings
4. Configure Health Check
5. Add EC2 Instances
Step 2: Assign Security Groups
You have selected the option of having your Elastic Load Balancer inside of a VPC, which allows you to assign security groups to your load
balancer. This can be changed at any time.
Assign a security group:
Create a new security group
Select an existing security group
Security Group ID
Name
Description
sg-e3e8fc9c
default
default VPC security group
sg-dc7166a3
Route53Lab
launch-wizard-1 created 2017-04-26T10:55:05.940+05:30

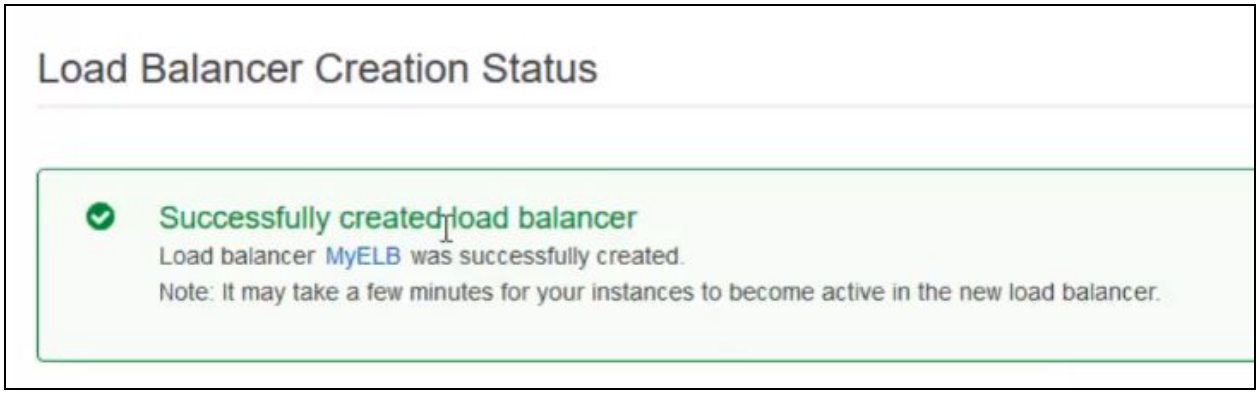

1. Define Load Balancer
2. Assign Security Groups
3. Configure Security Settings
4. Configure Health Check
5. Add EC2 Instances
Step 3: Configure Security Settings
Improve your load balancer's security. Your load balancer is not using any secure listener.
If your traffic to the load balancer needs to be secure, use either the HTTPS or the SSL protocol for your front-end connectio
under Basic Configuration section. You can also continue with current settings.



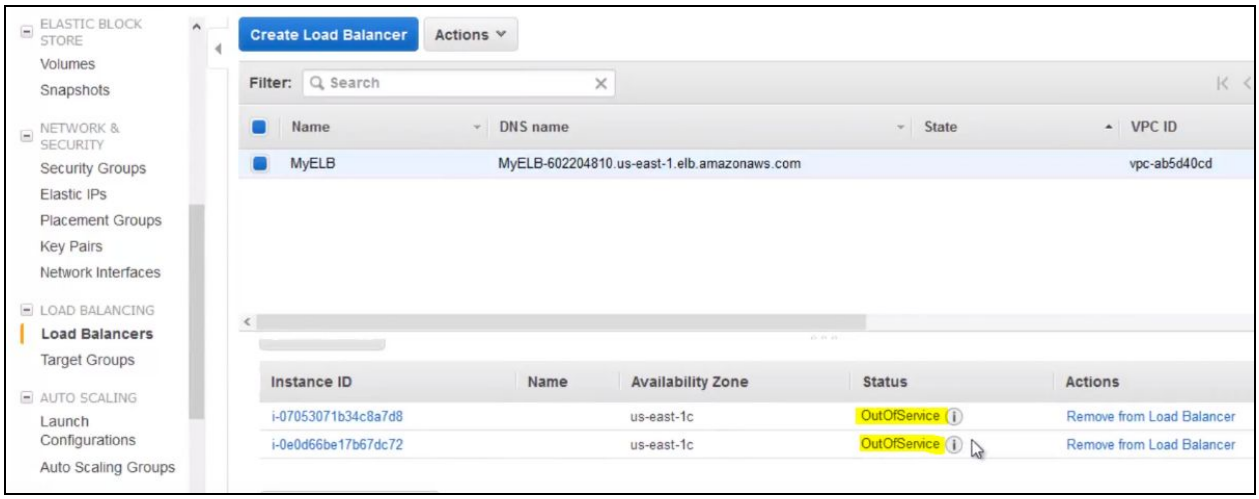
Add both the EC2 into ELB







Verify now



After sometime, Instances status will be in-service

You can verify by copy and paste the ELB DNS address in browser

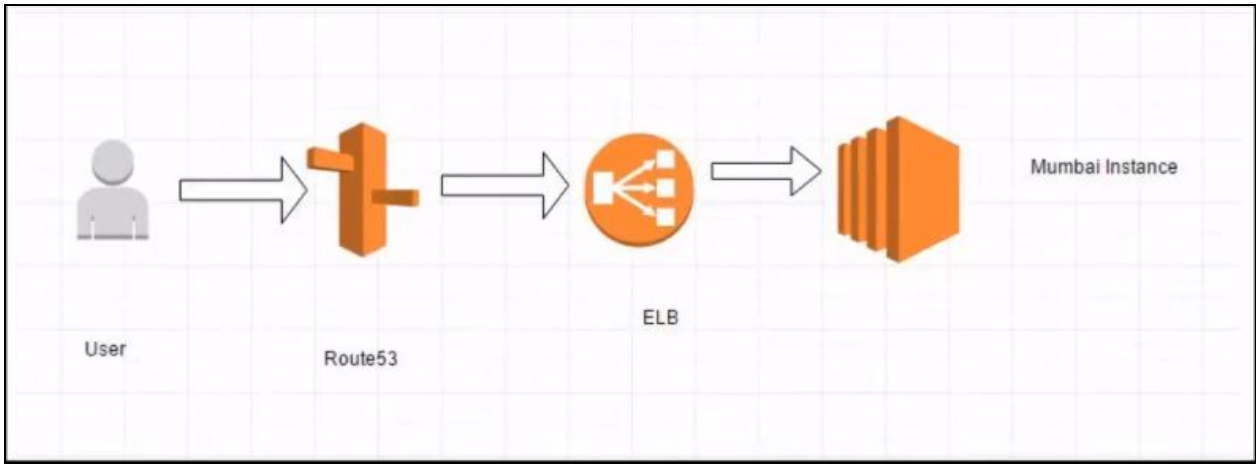
http://myelb-89010101.us-east-1.elb.amazonaws.com

You will the page as “Welcome to TLA from Instance1” one time

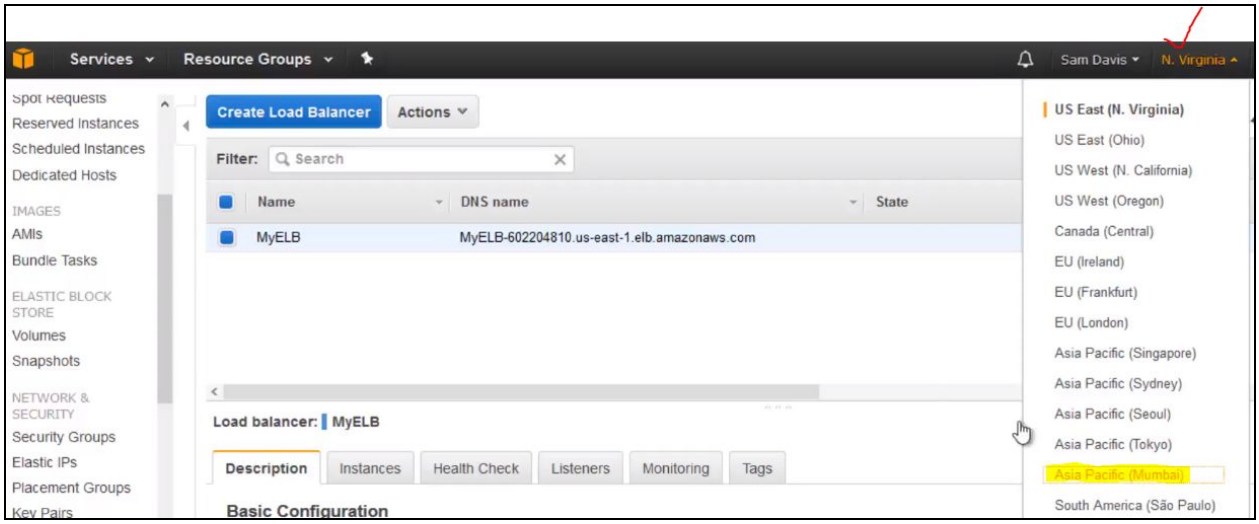
You will the page as “Welcome to TLA from Instance2” second time

Same ELB is serving the traffic from both instances

You can implement the below setup also.



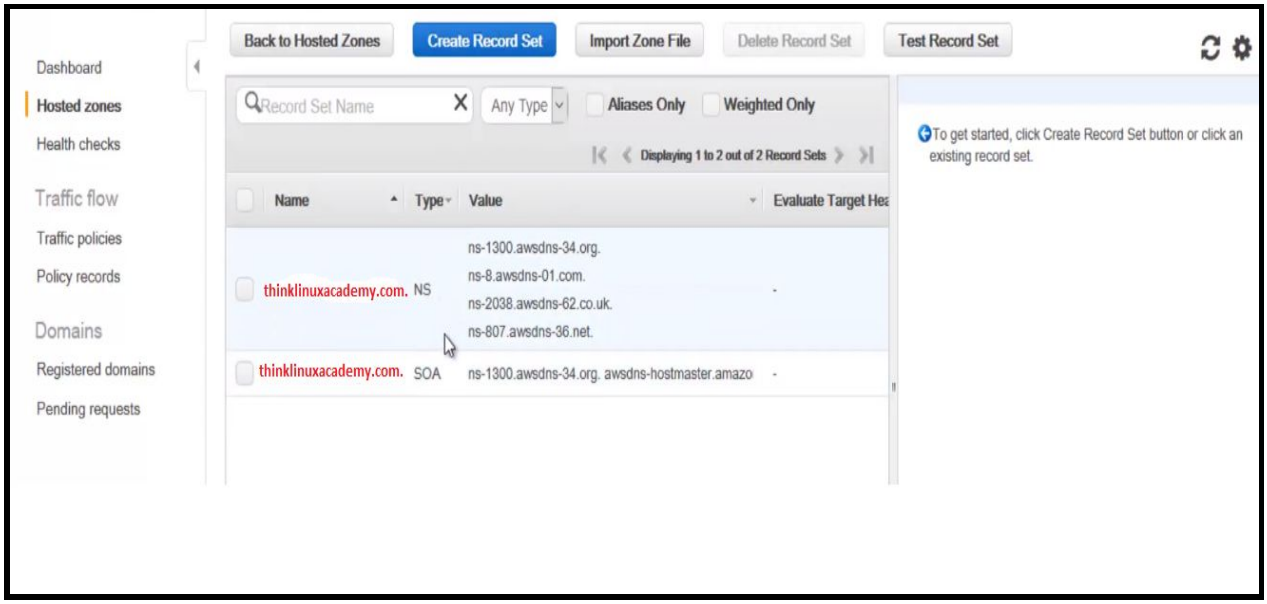
Switch your account to mumbai region



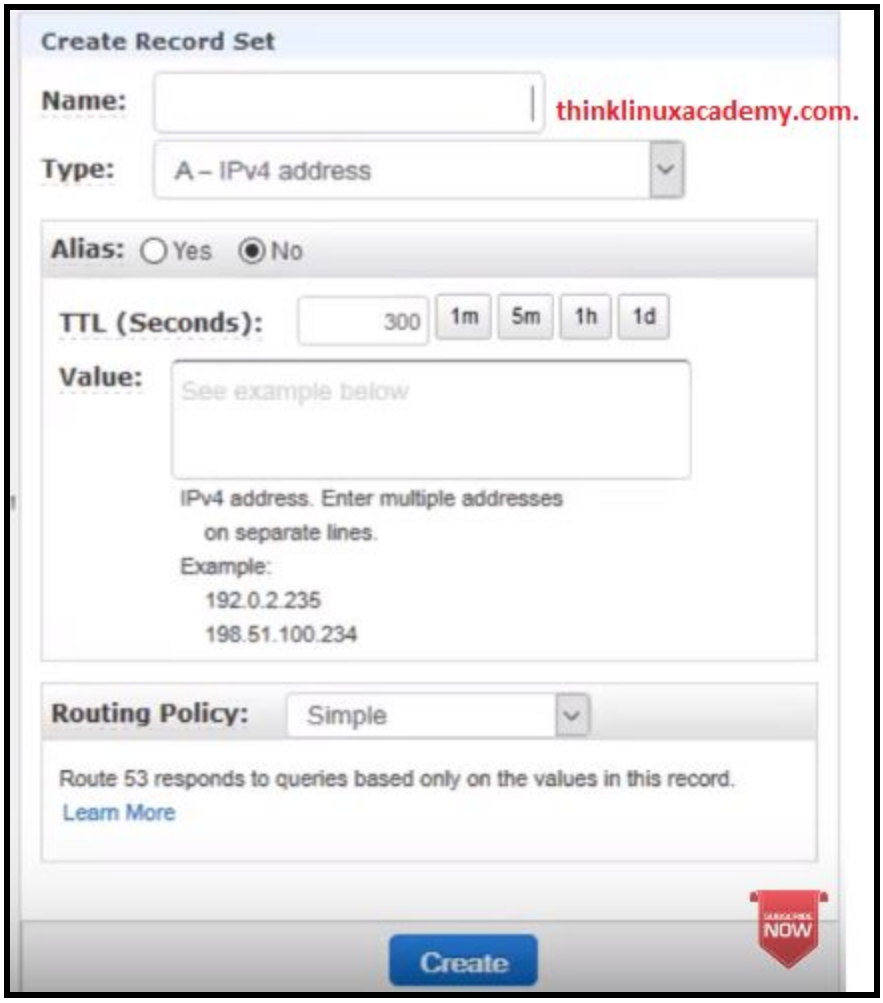
Create one EC2 and then ELB and add that EC2 into this ELB
Verify it is working or not using EC2 DNS ip and using ELB DNS ip

Configure Route53

Go to ROute53
Goto Hosted Zone



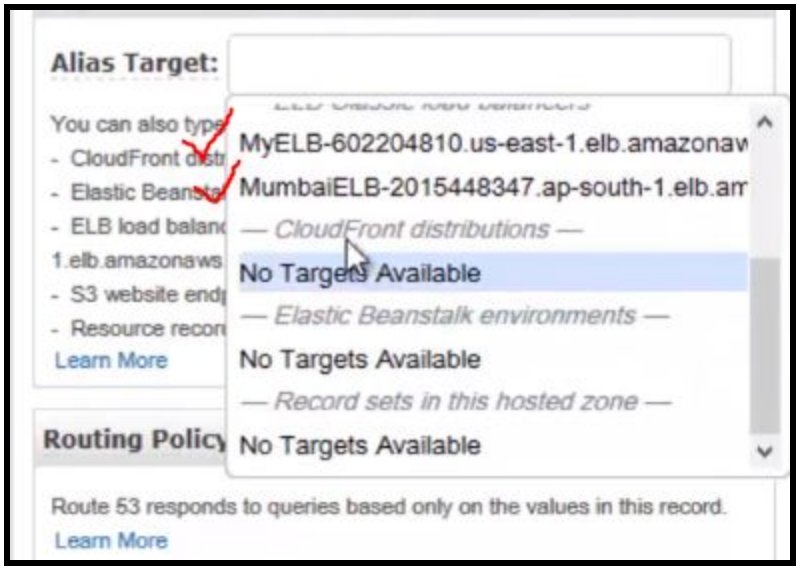
Click on “Create Record Set”



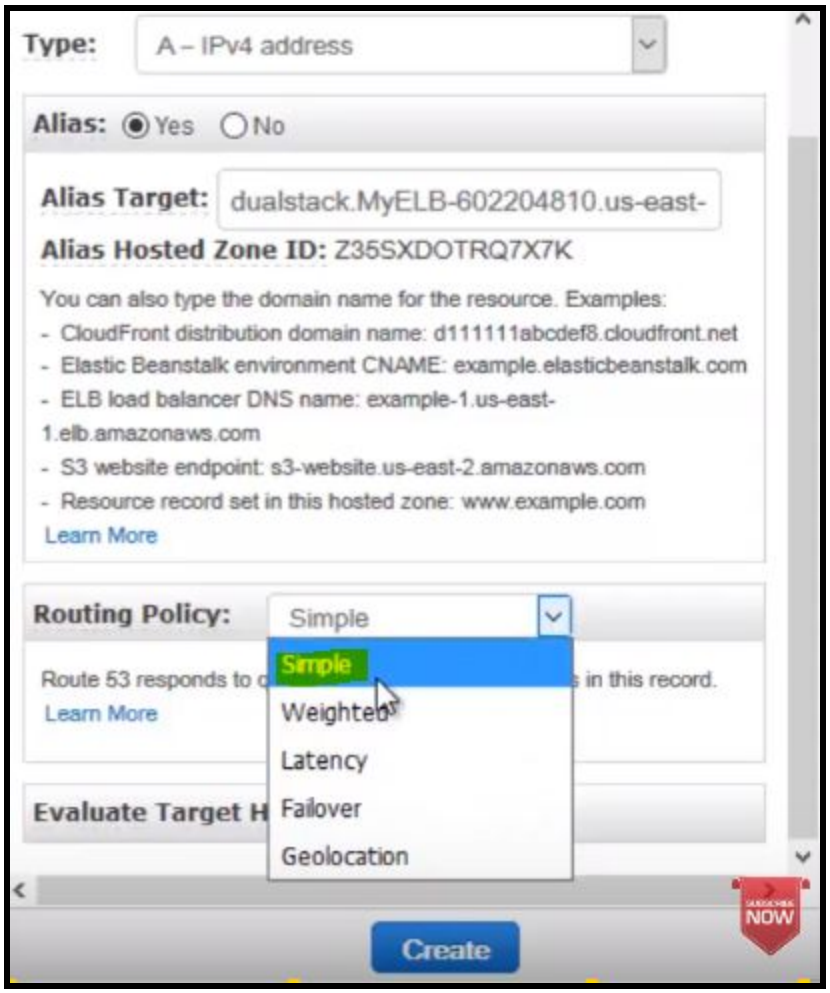
Name:
You can enter www in the Name section
Or you can leave empty (it is called naked domain) so that you can access the site as thinklinuxacademy.com

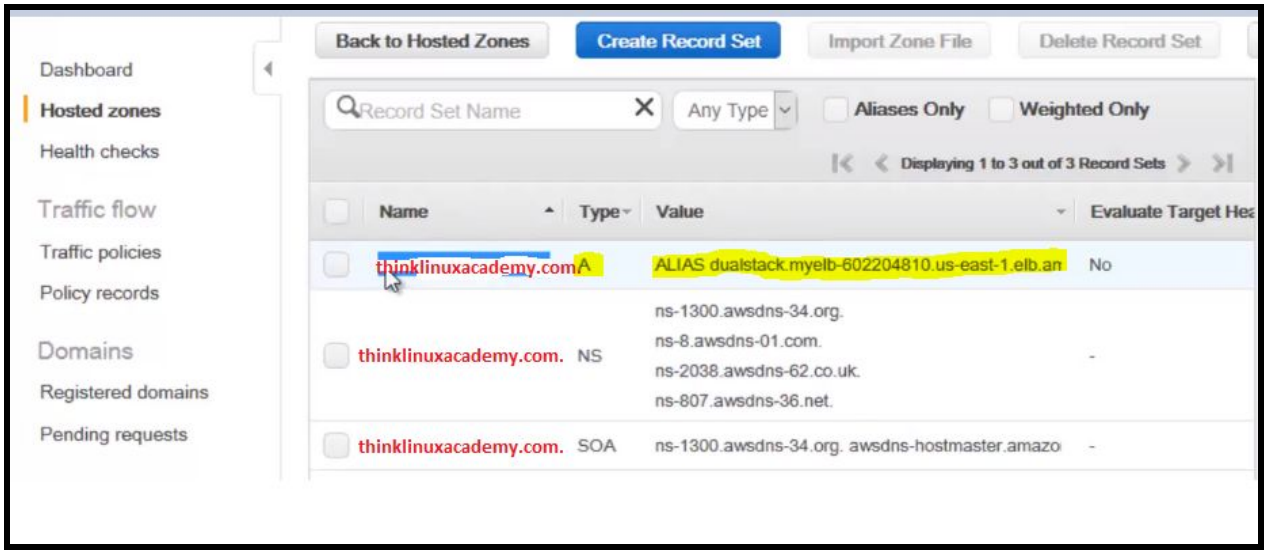
Alias Target:

Here you can enter the ELB address



Routing Policy:

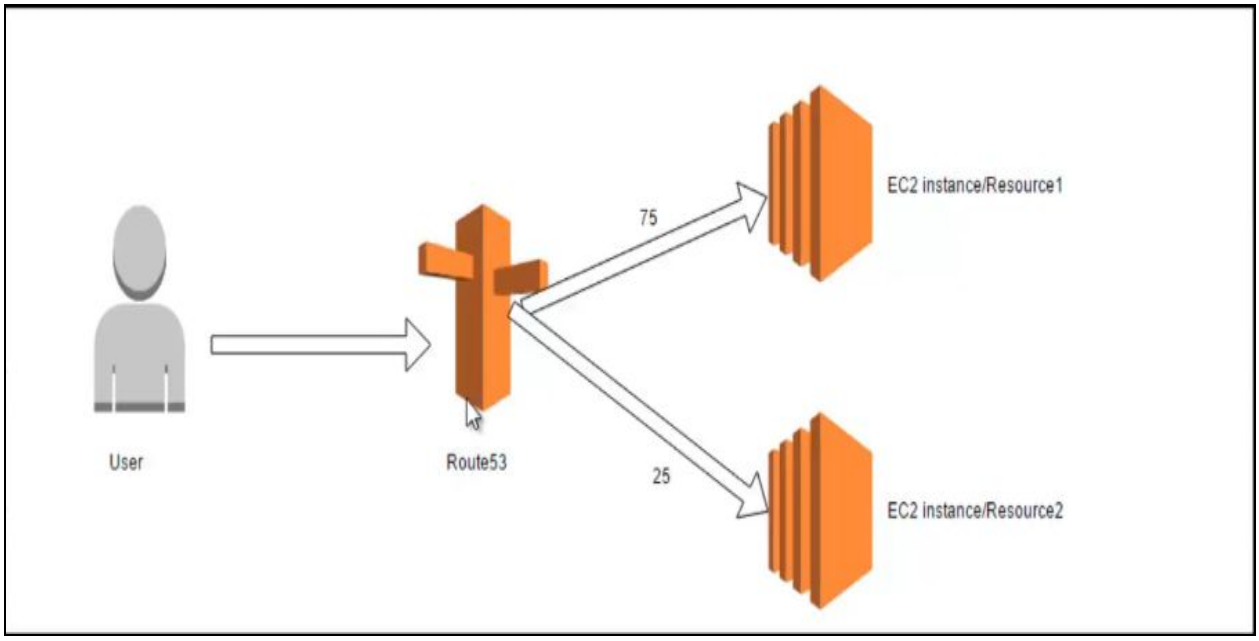




Because of this “A” Record, whenever you click on website it goes to this ELB as per this configuration.

Second time

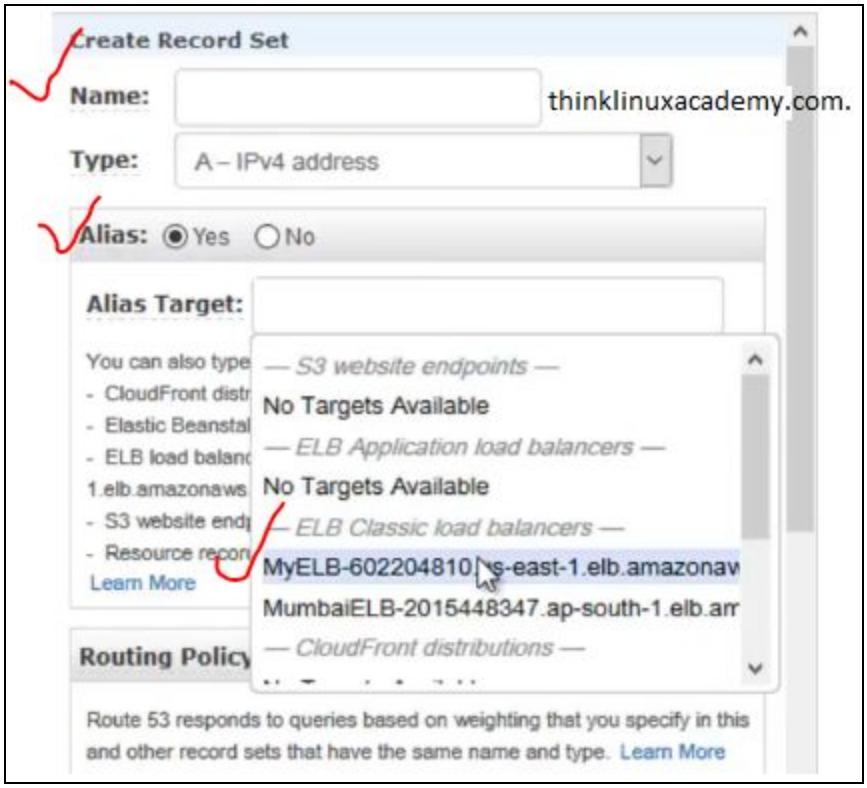
weighted:



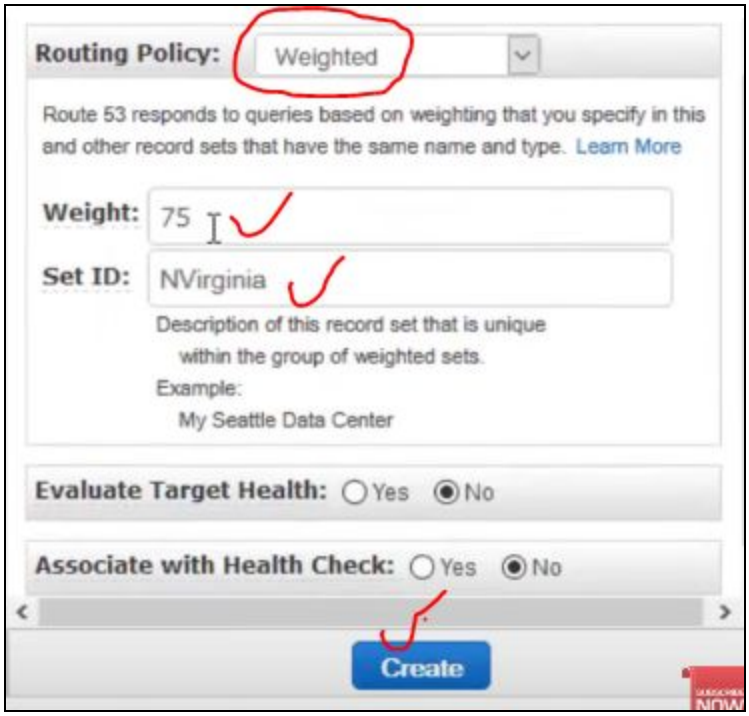
User will request for a specific alias record or website or domain name. \\\\\\\\\\\\\route53 will jump into action. It will try to resolve the DNS record. Based on the weight which we have allocated to EC2 instances who is serving the traffic.
1st set of EC2 instances is having more weight in our above case, so definitely, most of the traffic is taken care by 1set of EC2 instances. If less weight then it will take care next.

Click on **“Create Record Set”**

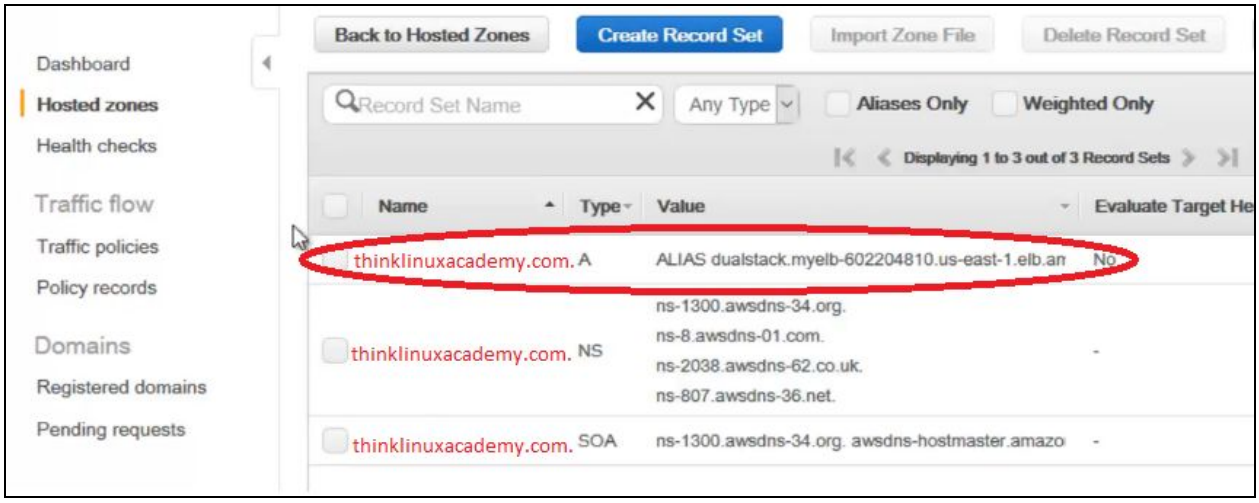
Name: (keep it empty called as naked domain)
Alias: yes and select the proper ELB as below



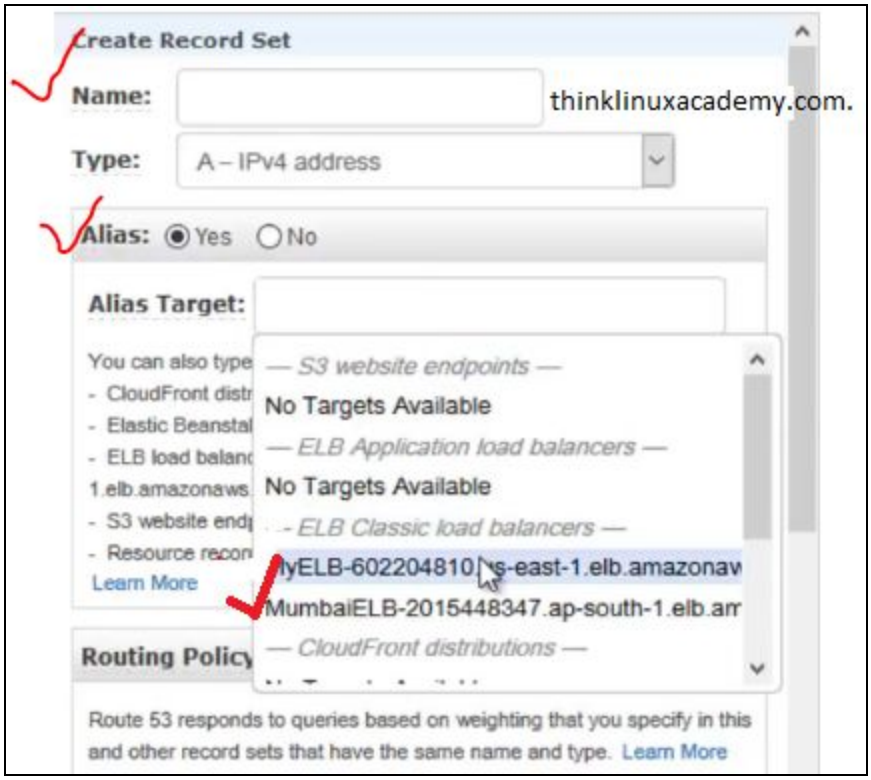
Routing Policy:

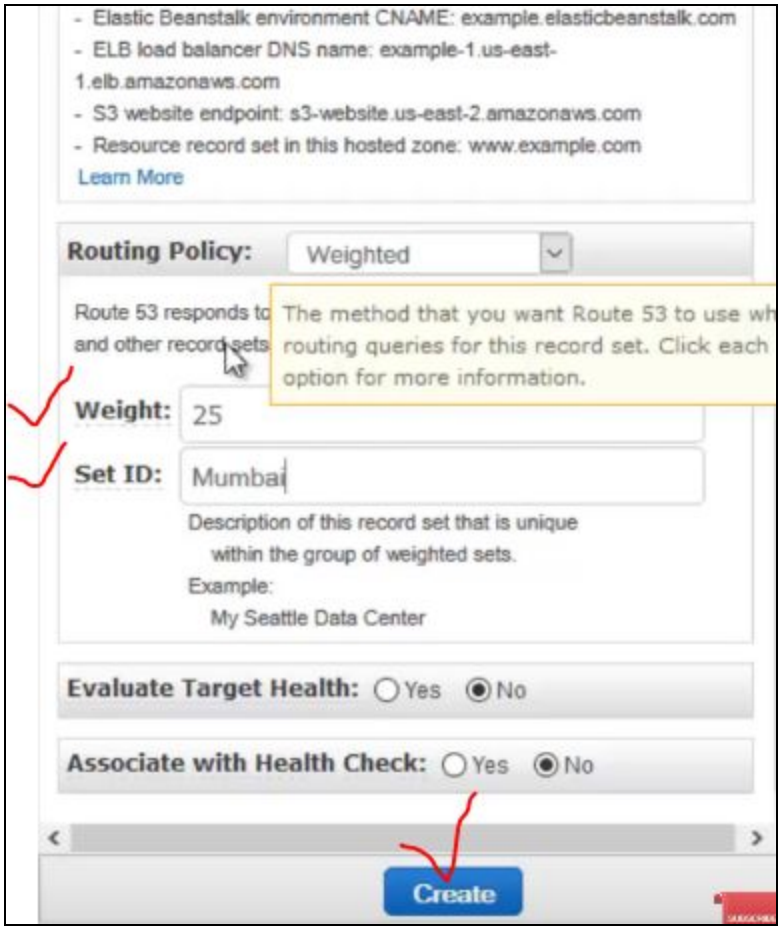


Then you can see the below entry in the main configuration

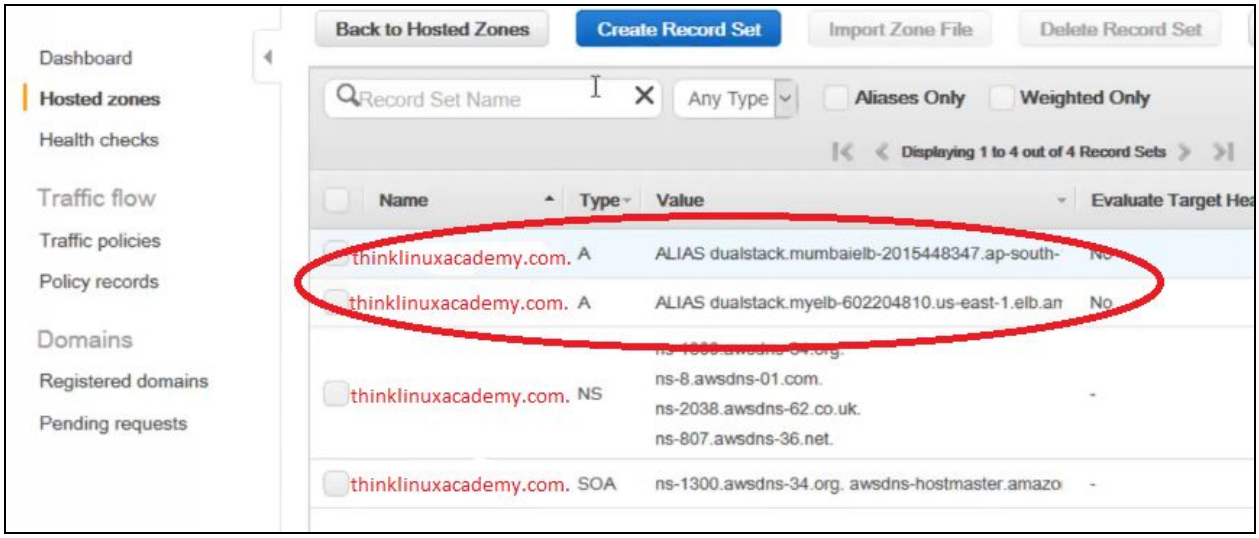


Then go back and create one more record set



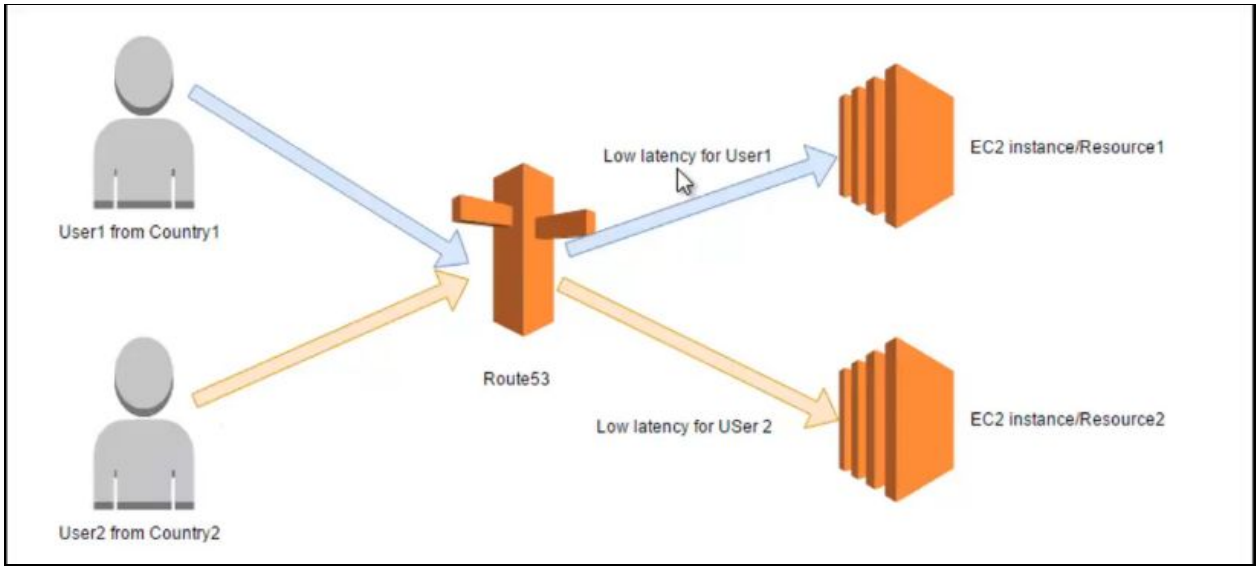


Now you can see 2 entry of A Record



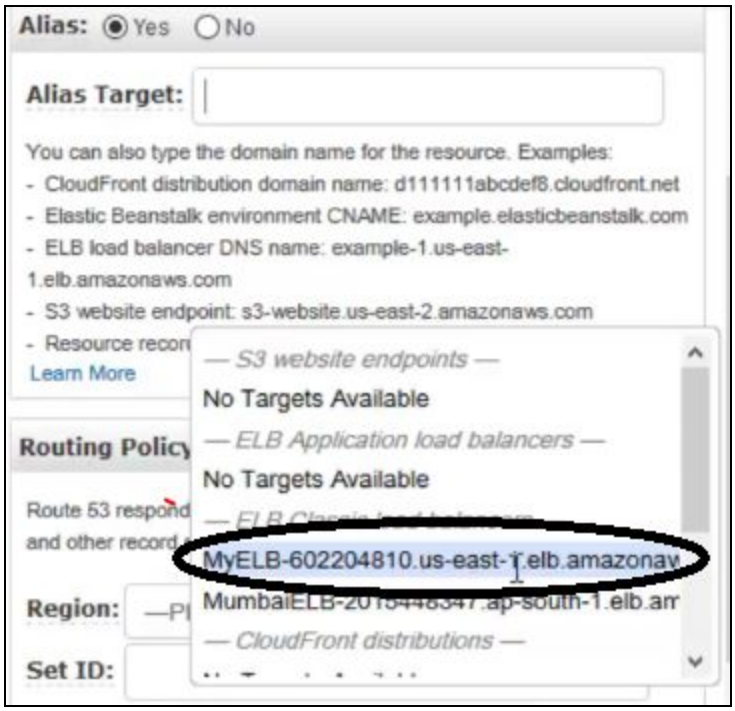
Due to this configuration, most of the traffic handle by north virginia ELB (75% of time)
Few times the traffic is handled by mumbai ELB (25% of time). Refresh many times

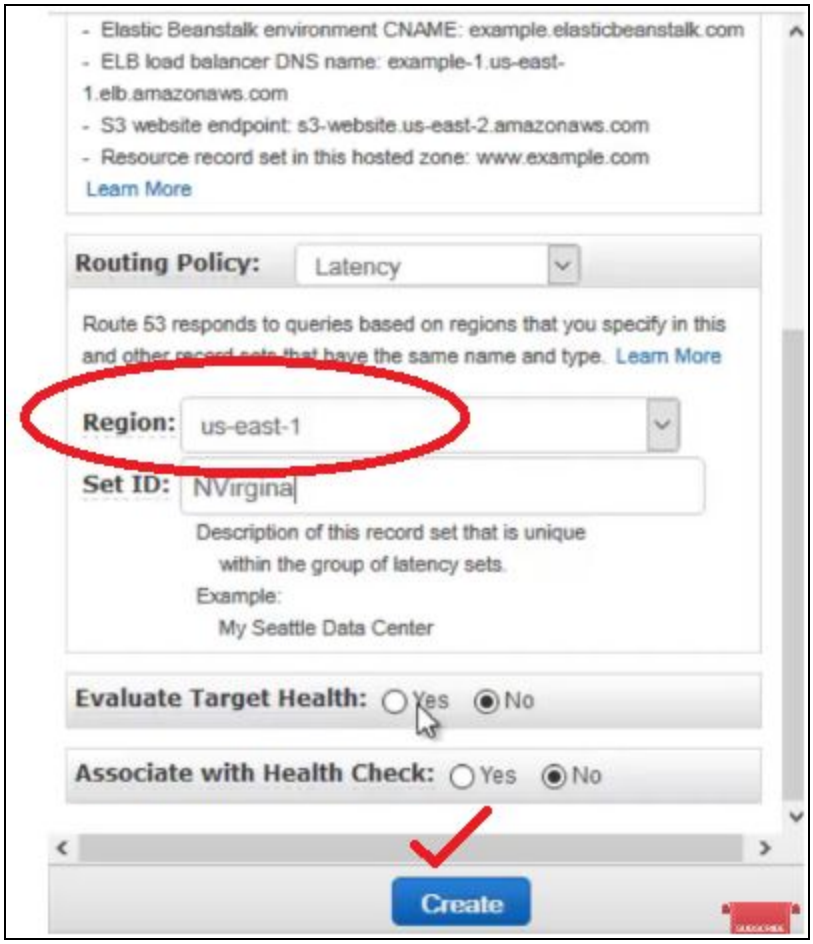
Latency based Routing:



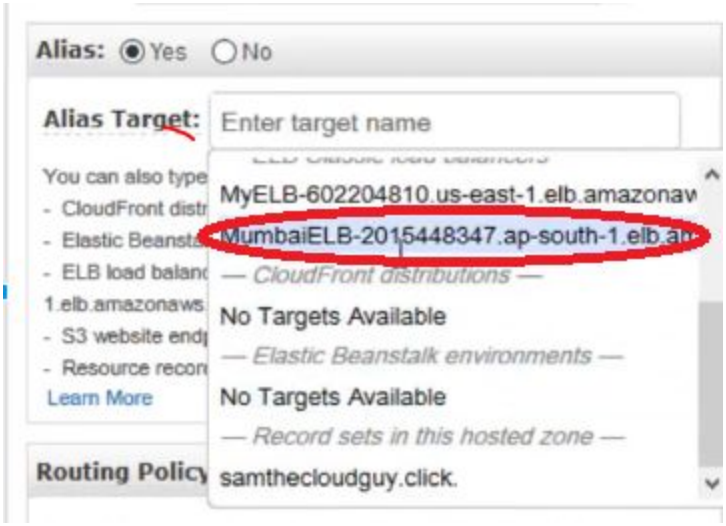
Deleted Record of last weighted based routing

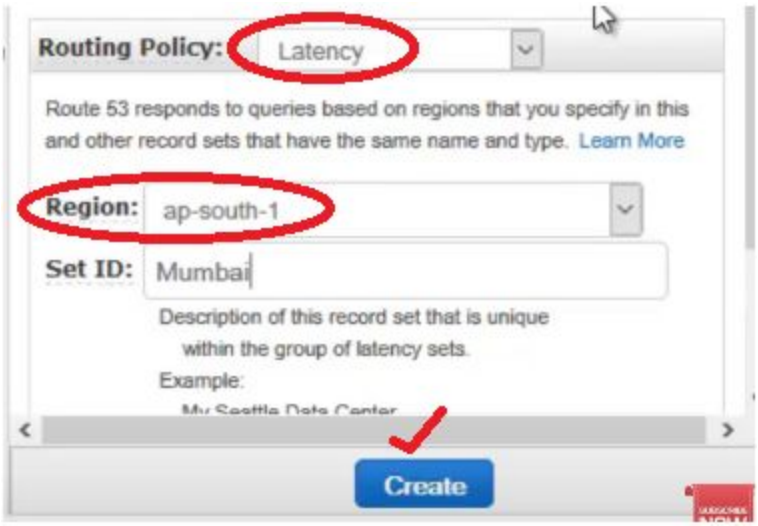
Now create a new record for Latency based routing



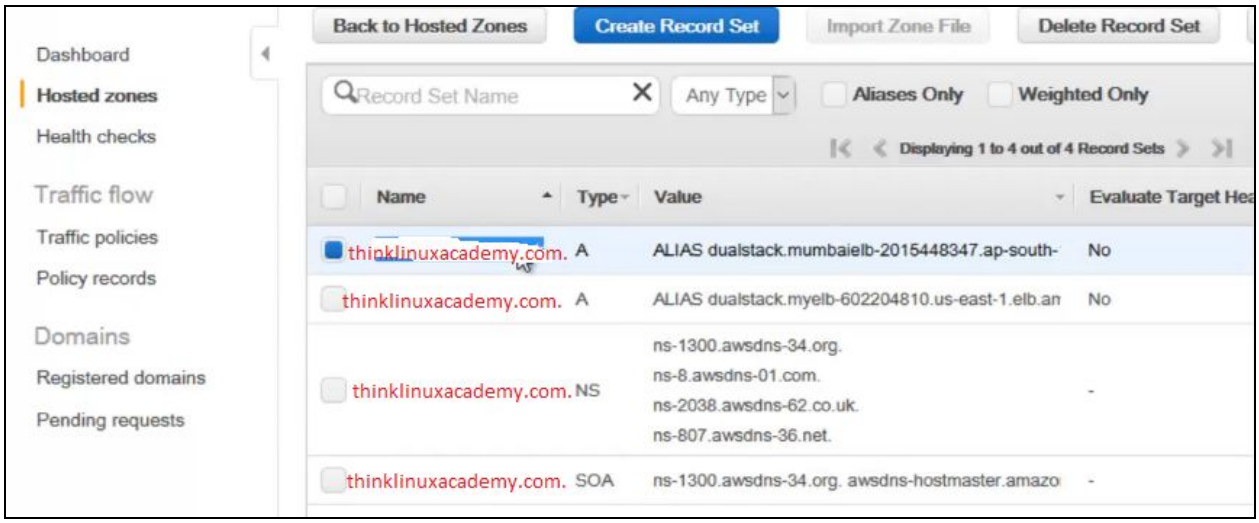


CREATE ONE MORE RECORD IN LATENCY





Now you can see the configuration as below:



Now you can check from india location . Try to open the website of thinklinuxacademy.com Then you will see the message from Mumbai instance, that means it is serving from mumbai location.