**How to deploy Node.js application on AWS with Github**

In this article, we will learn about how to deploy a NodeJS application on an AWS EC2 Linux instance and access in browser.

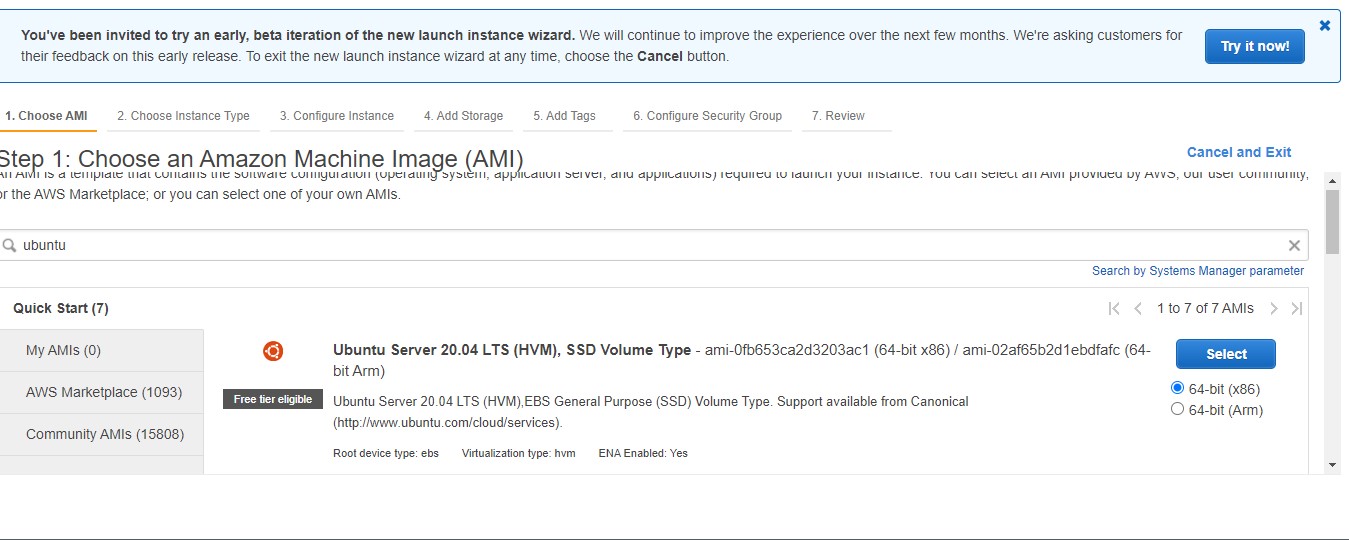


As a very first step, we need to configure an EC2 instance and securely SSH into it. To learn how to create an EC2 Linux(Amazon Linux) instance, please refer to my previous article on “[Securely SSH into AWS EC2 Linux instance](https://medium.com/@sumantmishra/securely-ssh-into-aws-ec2-linux-instance-42ad8a322ac5)”.

After successful SSH into EC2 instance, we will follow the below steps to deploy a sample NodeJS application on the EC2 instance:

* Install NodeJS and NPM using nvm
* Install Git and clone repository from GitHub
* Install dependencies
* Run the application
* Configure security group to access via public URL

**Step 1 : Create launch instance of the ubuntu AMI**

****

# 

# 

# 

# Setp2:- Create Elastic IP Address allocated

# 

# Setp3:- SSH Access to Visudo studio code

# 

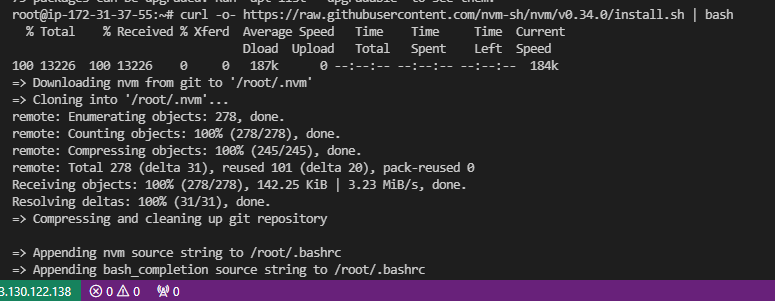
# Apt update

# 

# Setp3:- Install NodeJS and NPM using nvm

Install node version manager (nvm) by typing the following at the command line.

curl -o- <https://raw.githubusercontent.com/nvm-sh/nvm/v0.34.0/install.sh> | bash



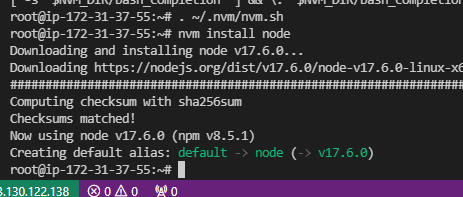
Activate nvm by typing the following at the command line.

. ~/.nvm/nvm.sh



Use nvm to install the latest version of Node.js by typing the following at the command line.

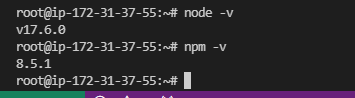
nvm install node



Test that node and npm are installed and running correctly by typing the following at the terminal:

node -v

npm -v



# Install Git and clone repository from GitHub

To install git, run below commands in the terminal window:

sudo yum update -ysudo yum install git -y

Just to verify if system has git installed or not, please run below command in terminal:

git — version



This command will print the git version in the terminal.

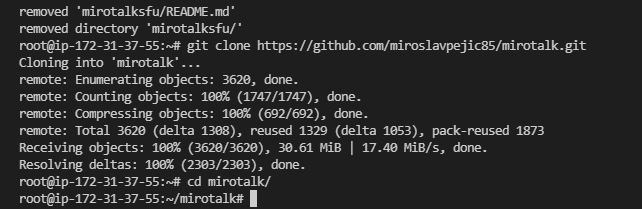
Run below command to clone the code repository from Github:

git clone https://github.com/miroslavpejic85/mirotalk.git

Install dependencies

Now, move to the folder node-app by running below command in the

cd mirotalk



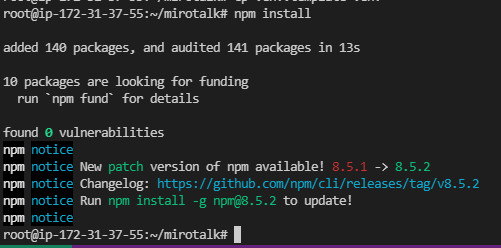
copy .env.template to .env

$ cp .env.template .env



# install dependencies

$ npm install



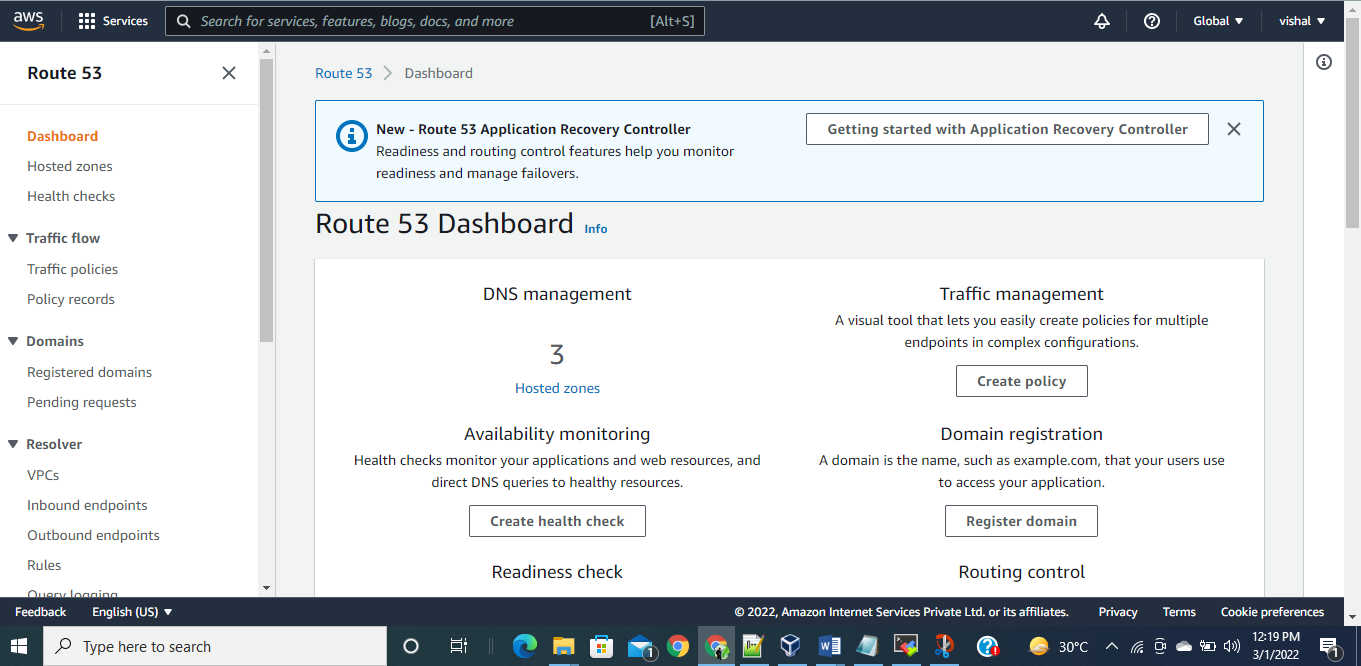
# start the server

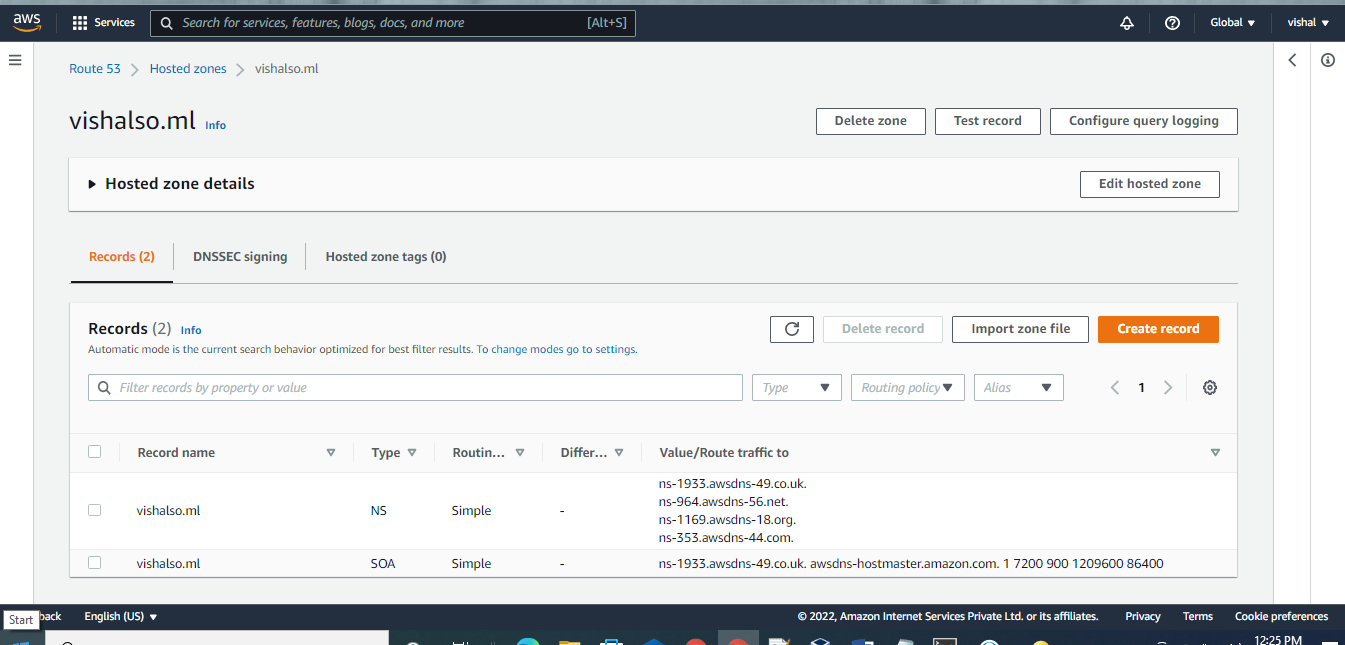
$ npm start



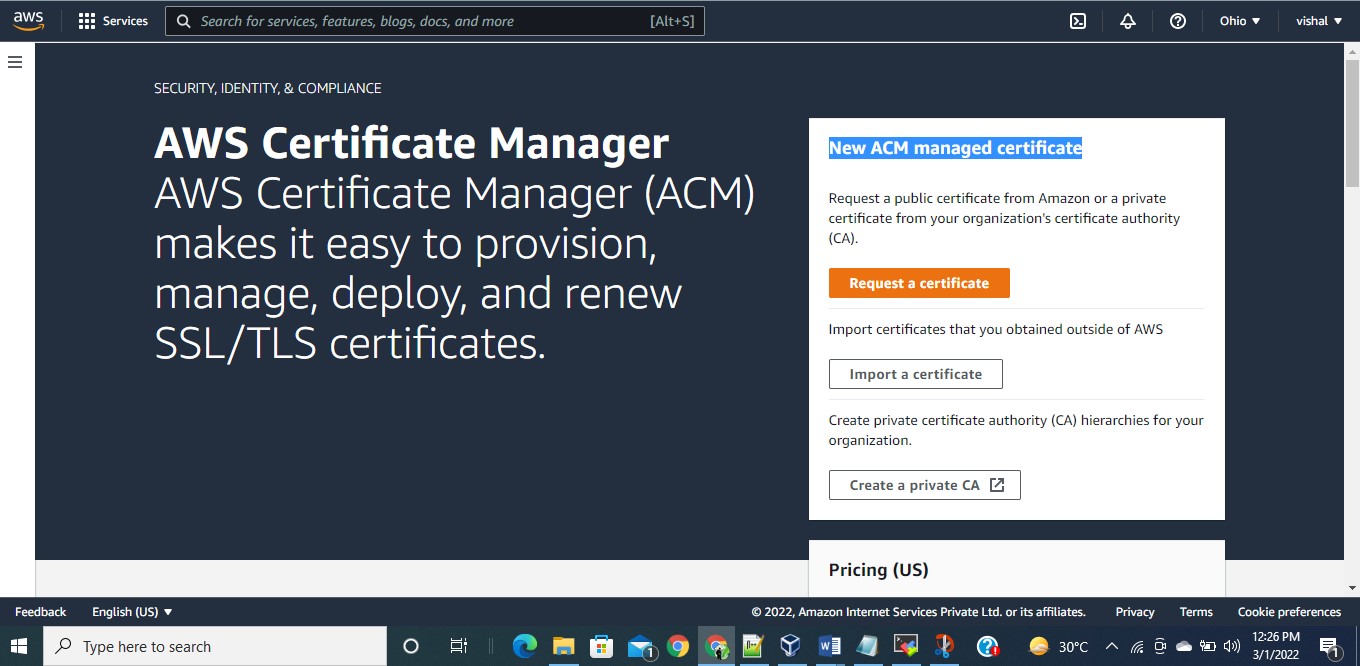
Setp4:- Create Rotuer 53:-

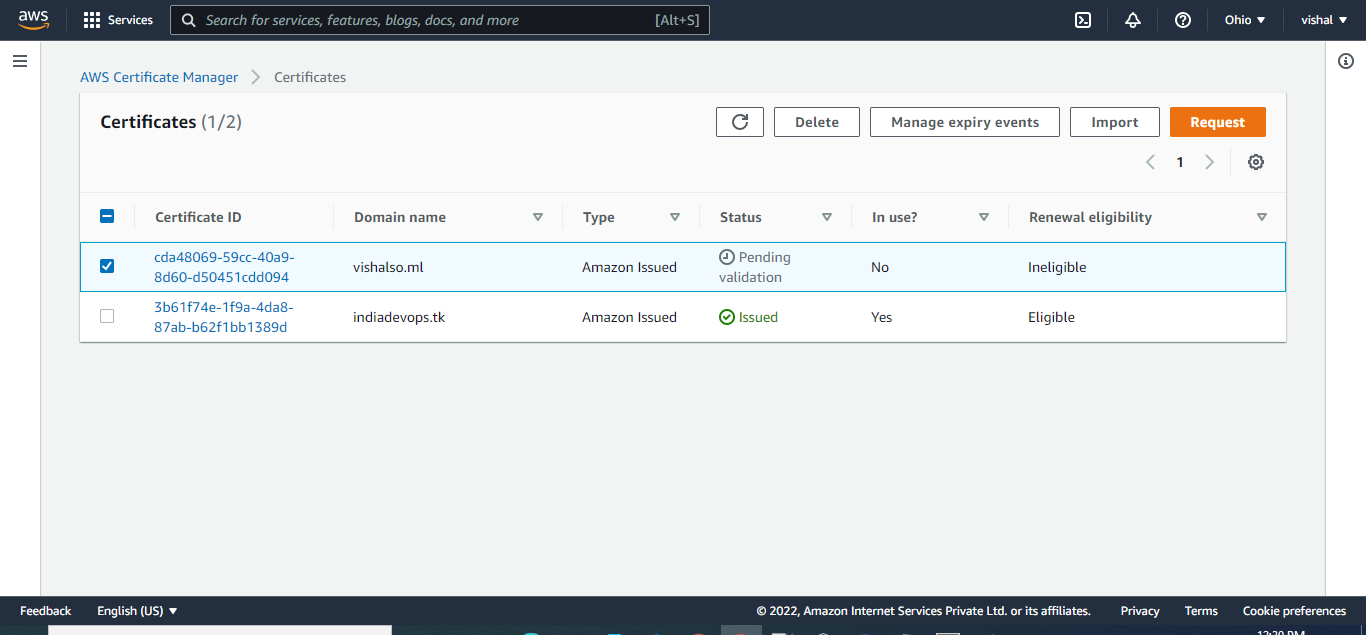
Amazon Route 53 is a highly available and scalable cloud [Domain Name System (DNS)](https://aws.amazon.com/route53/what-is-dns/) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names 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.





**Setp5:- Create AWS Certificate Manager**

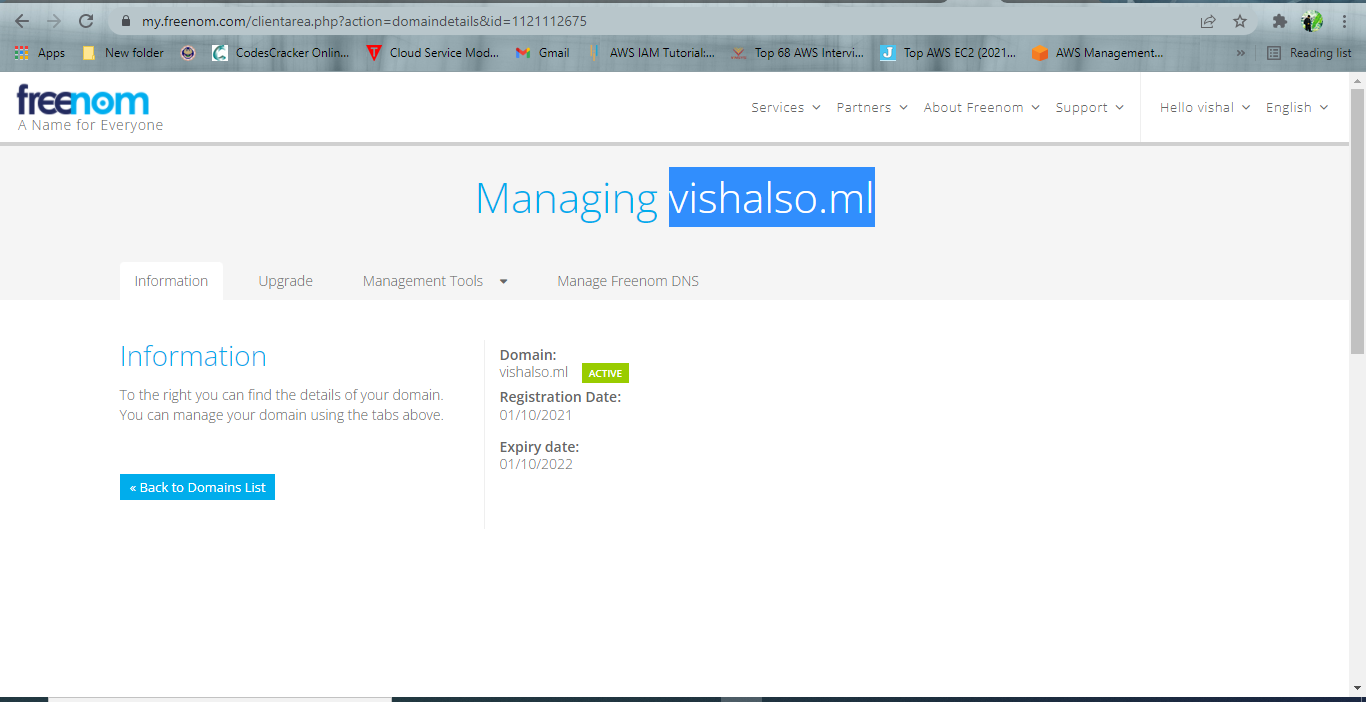




Setp5:- Website Freenom Domain

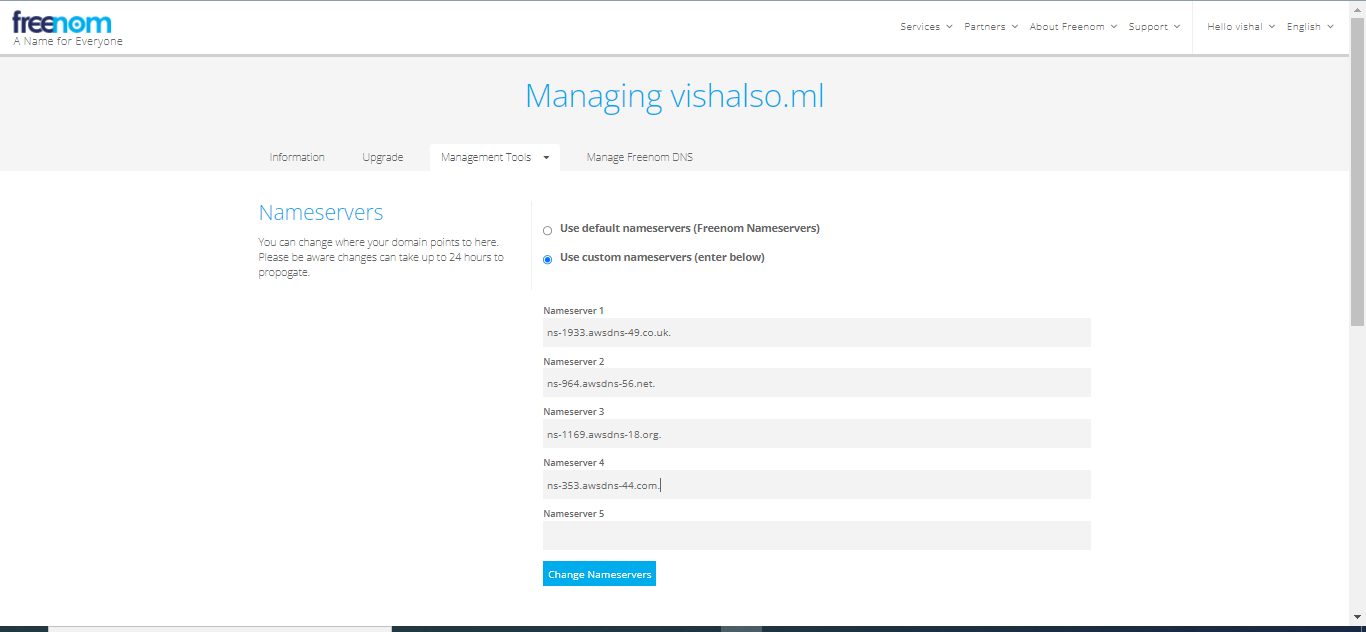
purchase to freenom website

Domain nameserver.

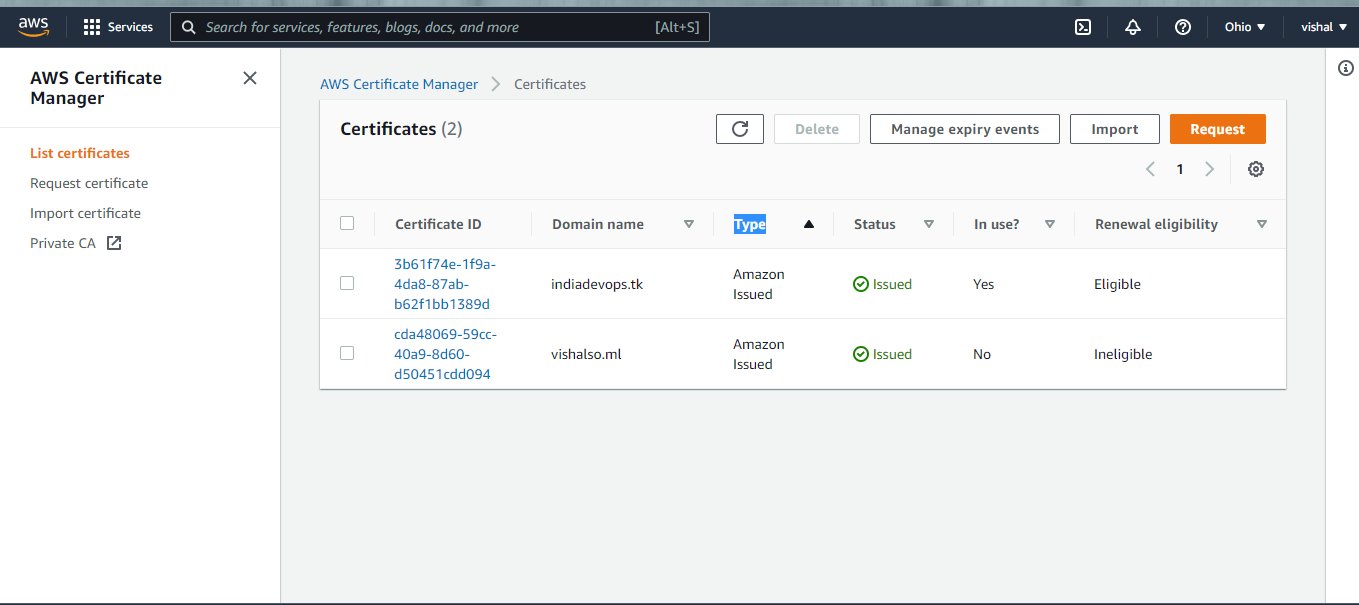


freenom website

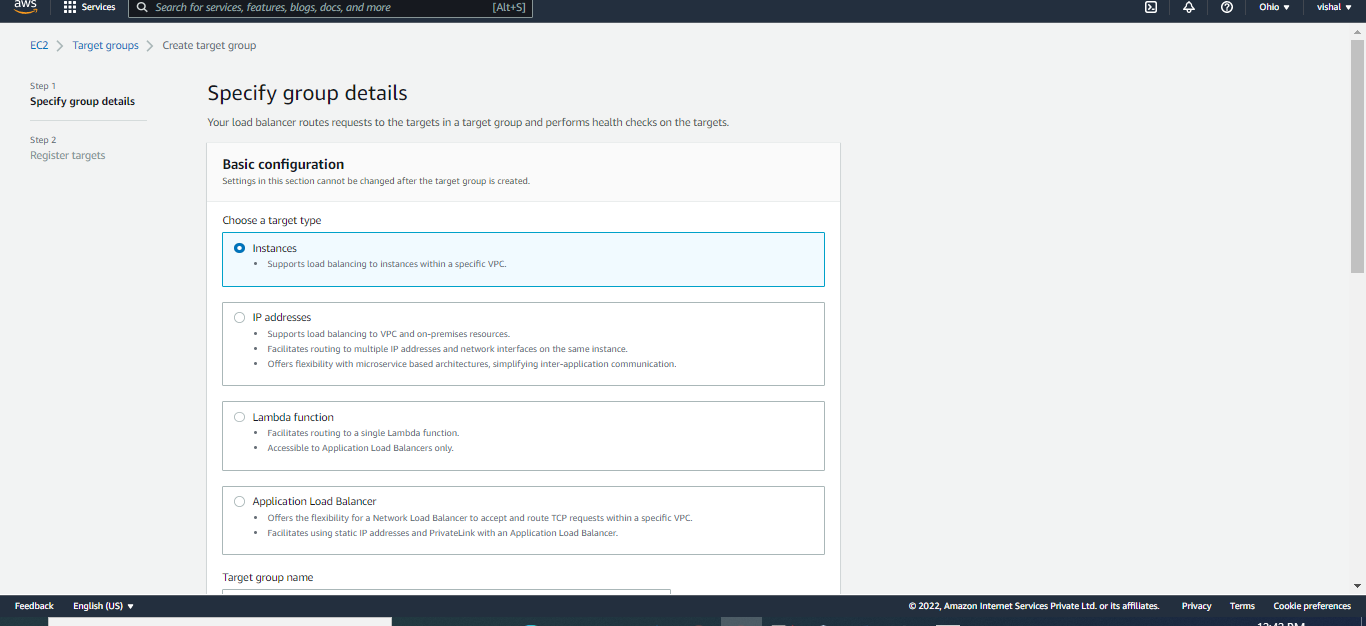
Nameserver added to Router53 record ns upload form freenom website

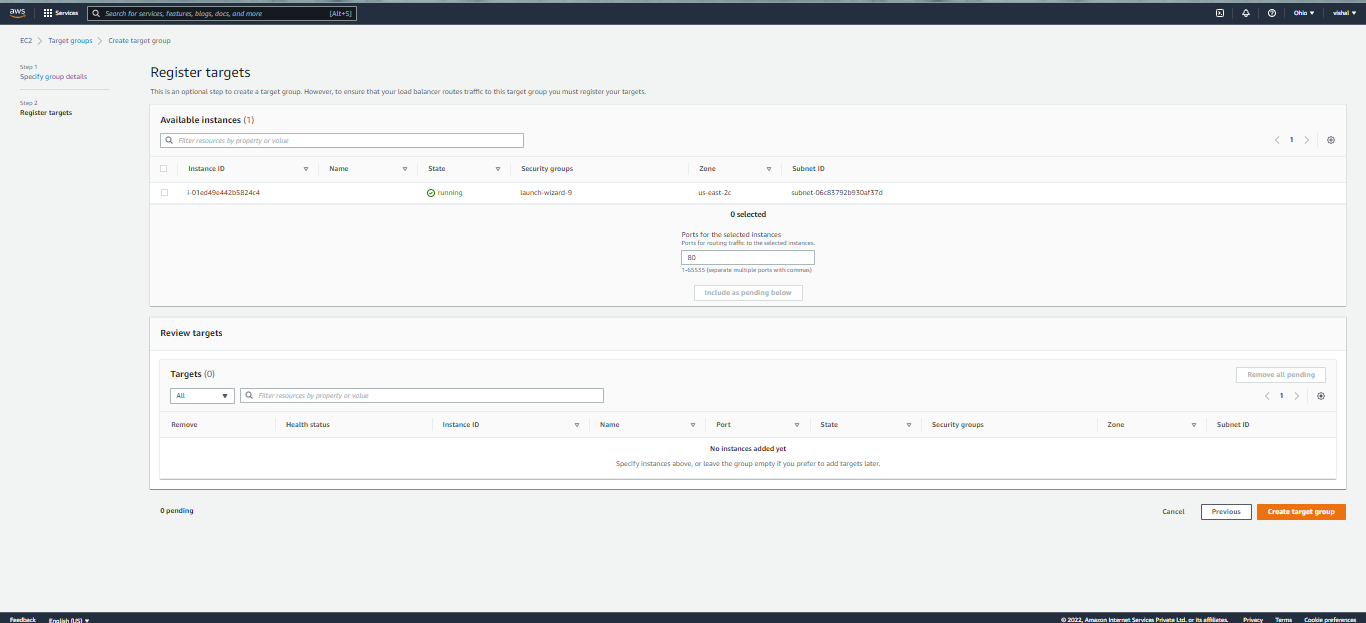


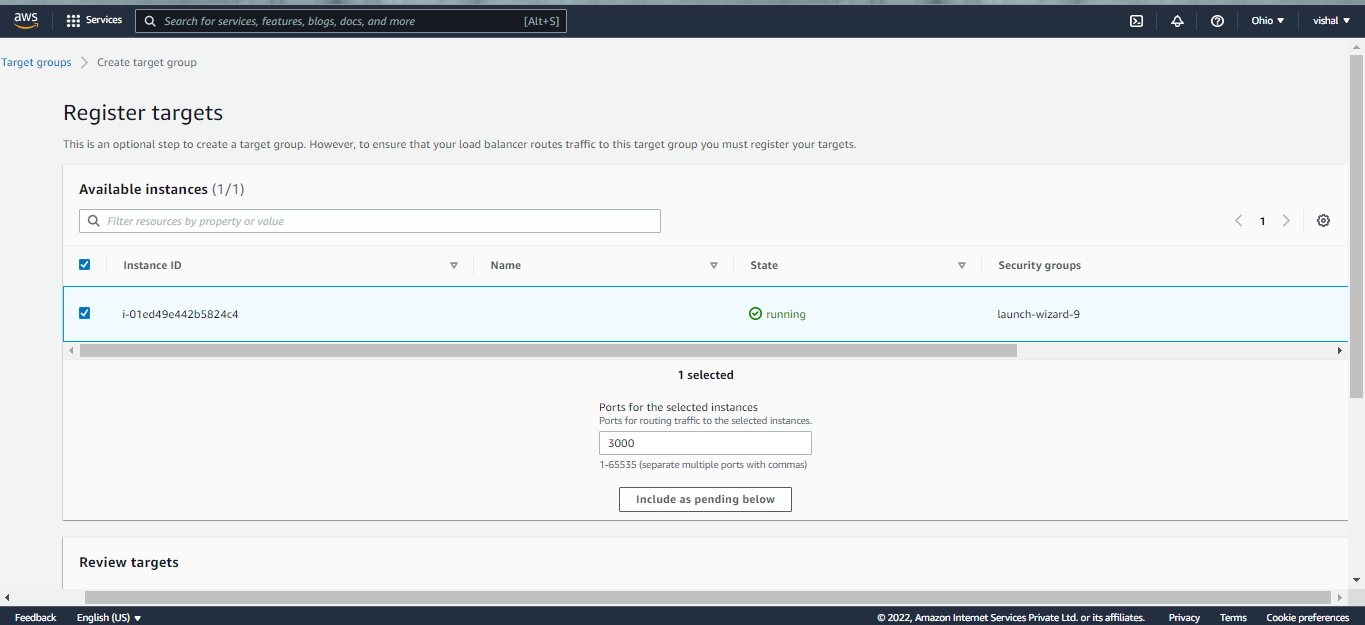
Setup6:- AWS CERTIFICATE MANAAGES ISSUED .

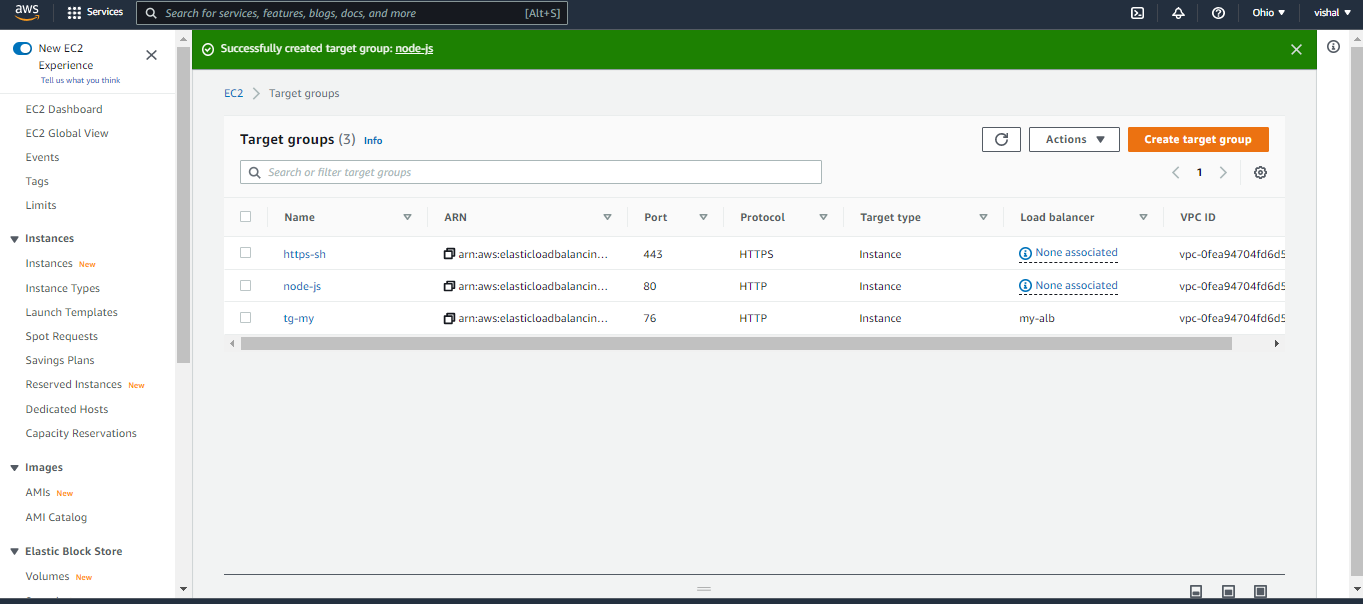


**Setp7: Create target groups:**

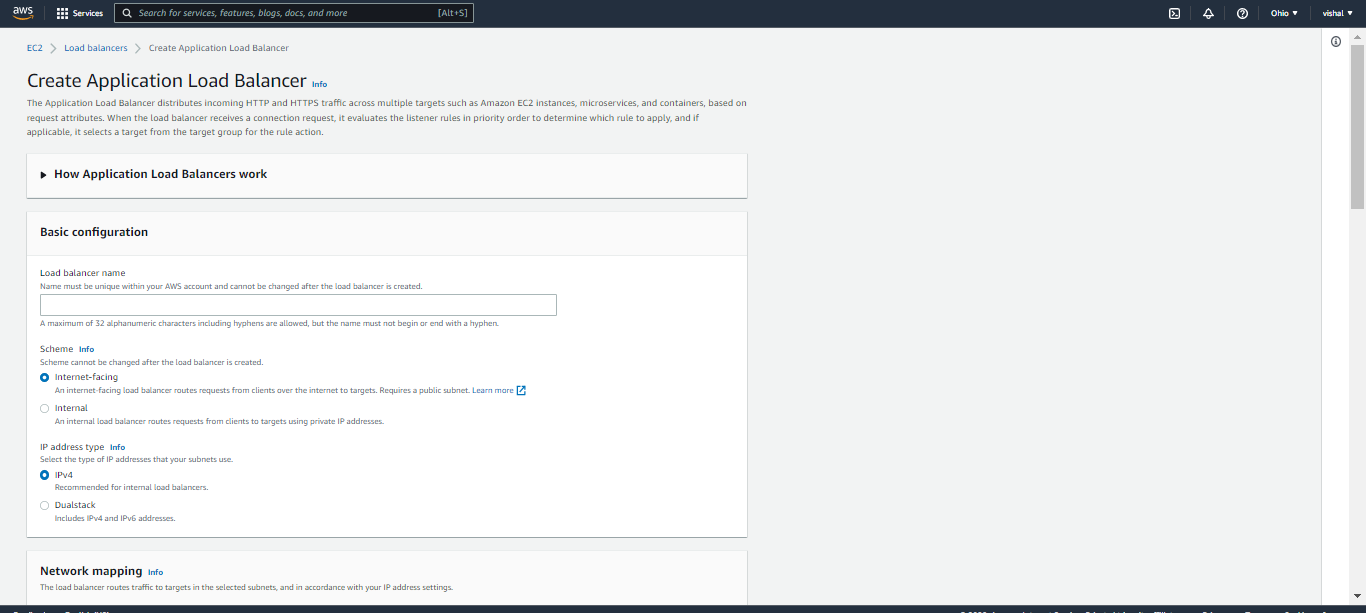
****

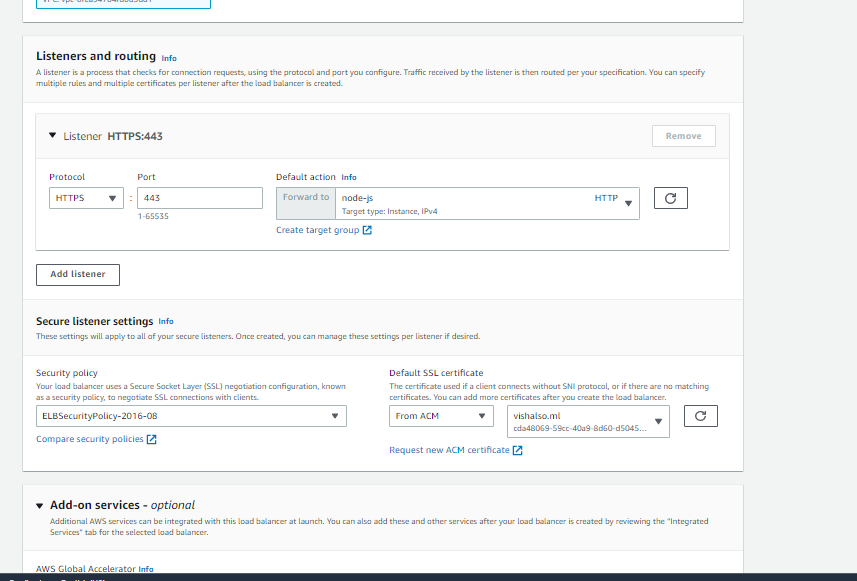
****

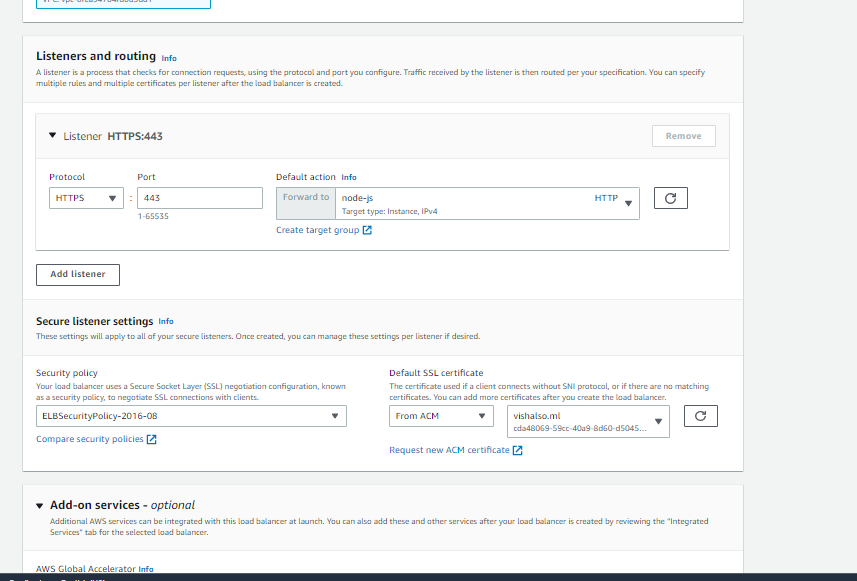
****

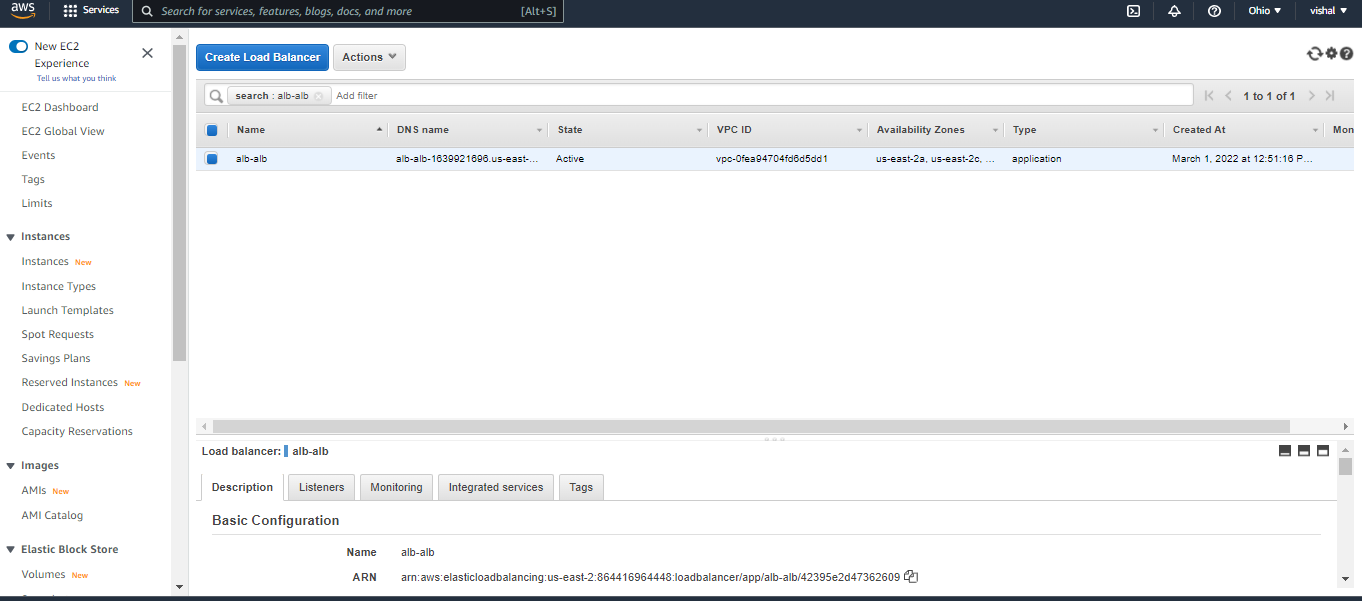
****

**Setup8:- Create** [Load Balancers](https://us-east-2.console.aws.amazon.com/ec2/v2/home?region=us-east-2#LoadBalancers:)

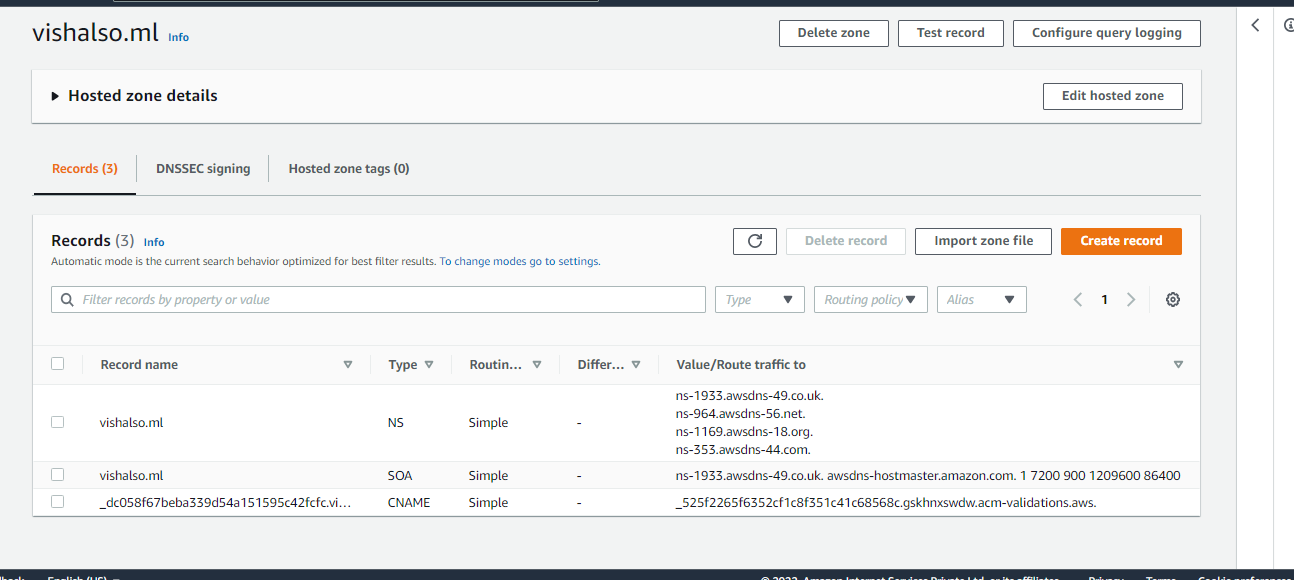
****

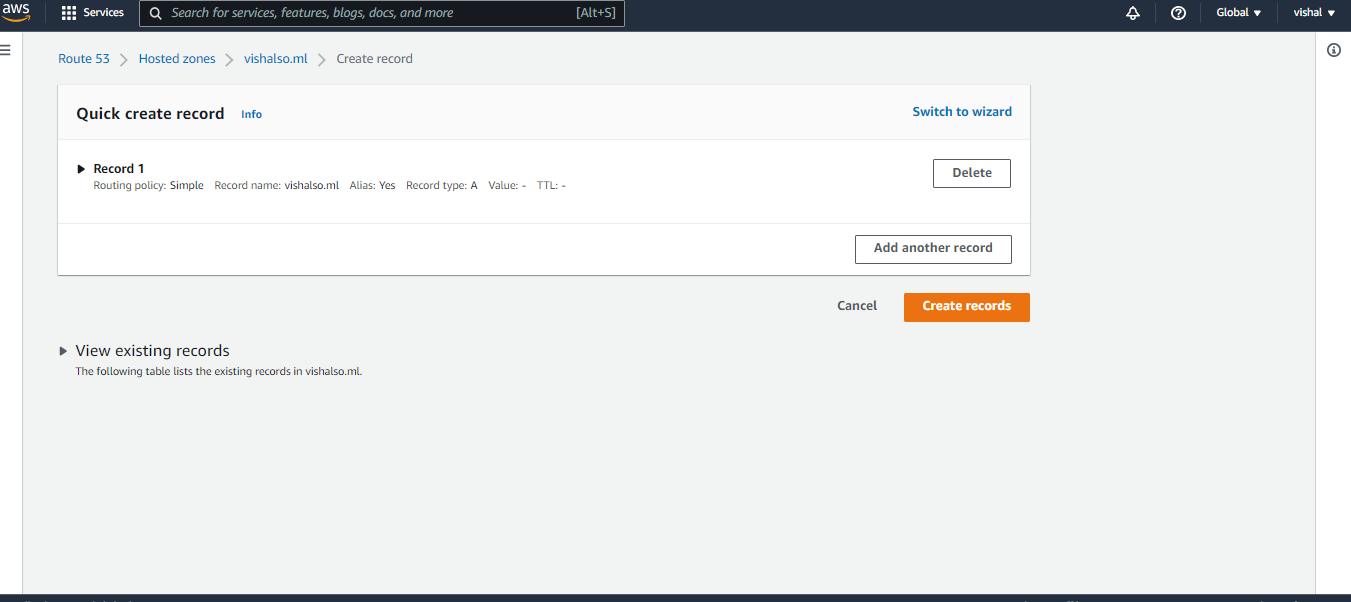
****

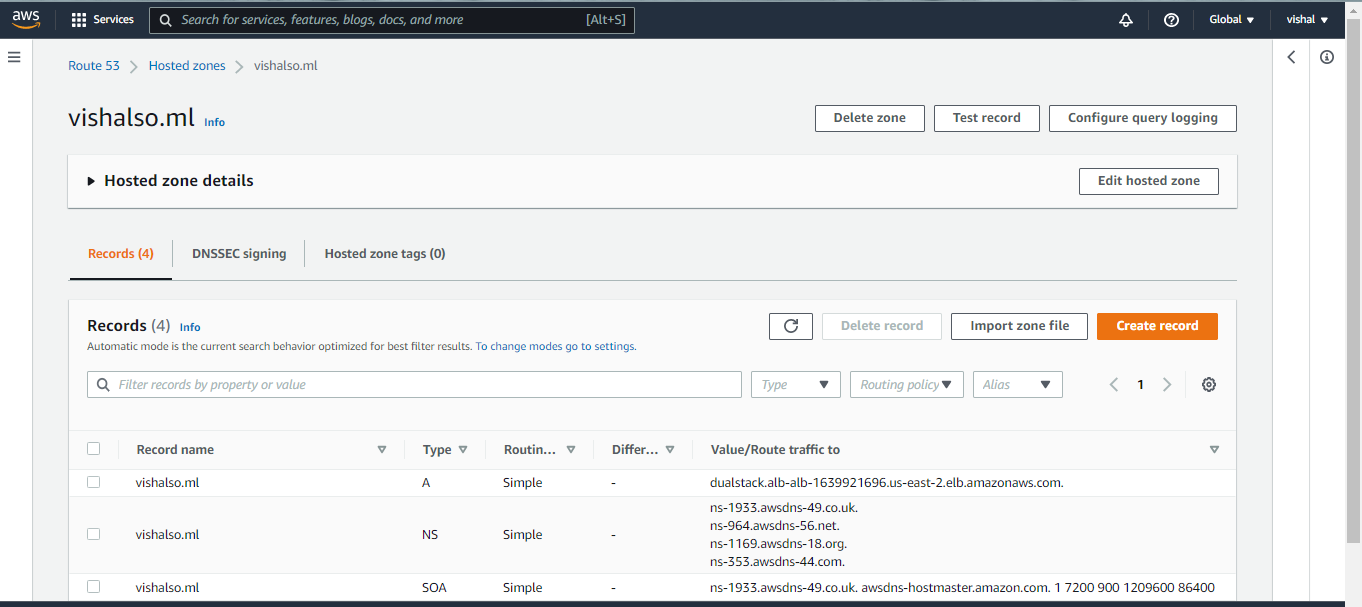
****

****

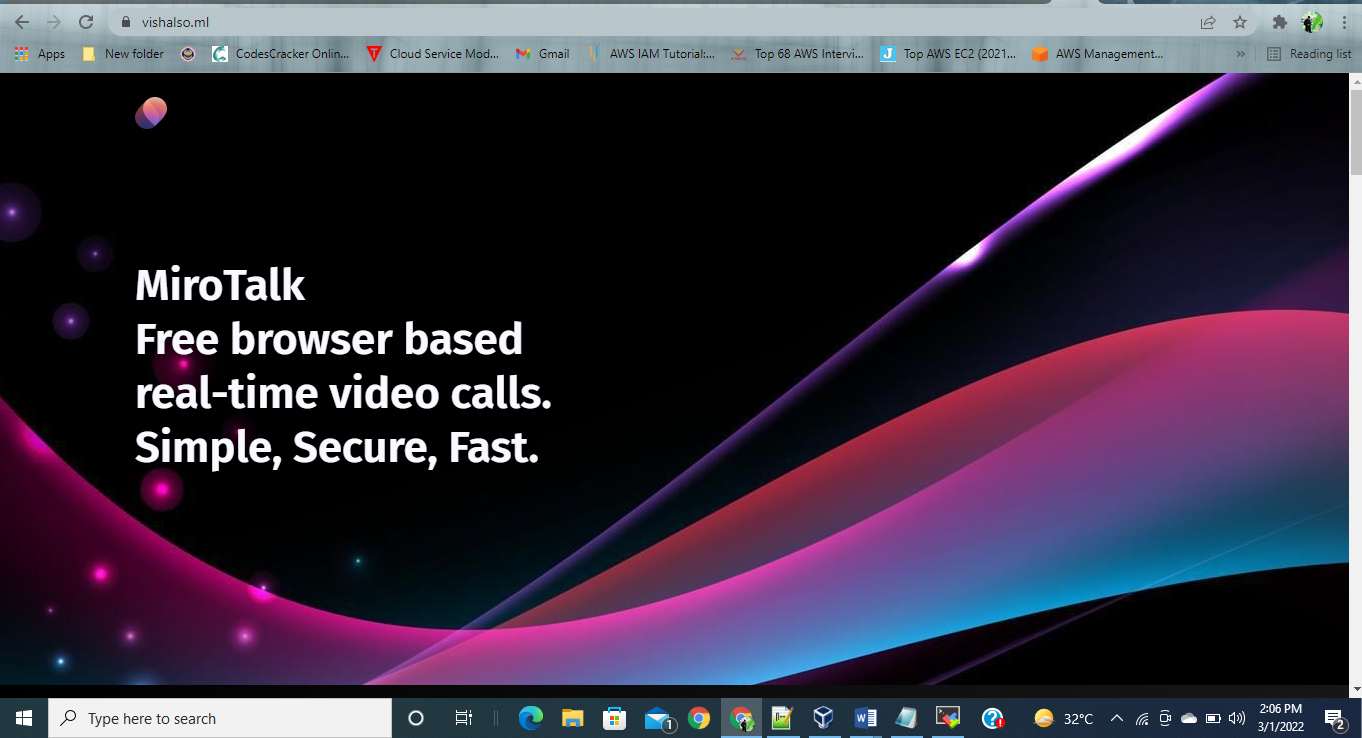
**Setup9:- Create router 53 A-record**

****

****

****

**Setup 10:- ping the https://vishalso.ml**

****