Contents

[General Diagram 2](#_Toc50869598)

[Prerequisites 2](#_Toc50869599)

[Labs10: Lab Prerequisites (Option 1: Using Cloudformation) 2](#_Toc50869600)

[Labs10: Lab Prerequisites (Option 2: Using CLI) 3](#_Toc50869601)

[Network Infrastructure and Creation of Instance 3](#_Toc50869602)

[Differentiate EC2 Instances 6](#_Toc50869603)

[Lab 10c1: Created Simple and Geolocation Routing Policy 7](#_Toc50869604)

[Create Hosted Zone 7](#_Toc50869605)

[External Registrar (GoDaddy) 9](#_Toc50869606)

[Create the Simple Routing Policy 11](#_Toc50869607)

[Create a Geolocation Reouting Policy 15](#_Toc50869608)

[Tested the routing policies 20](#_Toc50869609)

[Evidences to send 21](#_Toc50869610)

[Clean Resources 22](#_Toc50869611)

# General Diagram

Diagrama, Esquemático

Descripción generada automáticamente

Create a well know infrastructure with 2 Public Instances with different messages (Europe and Asia), and later create a Hosted Zone on Route 53, link then nameserver from previous Hosted Zone on the Domain Registrar (GoDaddy) and finally, create a Geolocation Routing Policy on the same domain ([www.servicar.club](http://www.servicar.club)) to identify Source IPs and oriented to specific instance.

The first infrastructure is done using CLI, but you can do it using Web Management Console. The remainder instructions are done using Web.

# Prerequisites

Labs1c1 have to be done and the context for Administrative user have to activated on Command Line Session.

Labs4c1 has the context to create infrastructure: Network (VPC, Subnets), Sec Groups and Instances.

Labs5c1 can created containers from Command Line.

Labs9c1 can create common infrastructure using Cloudformation and you see there how to delete the stack after finishing the lab.

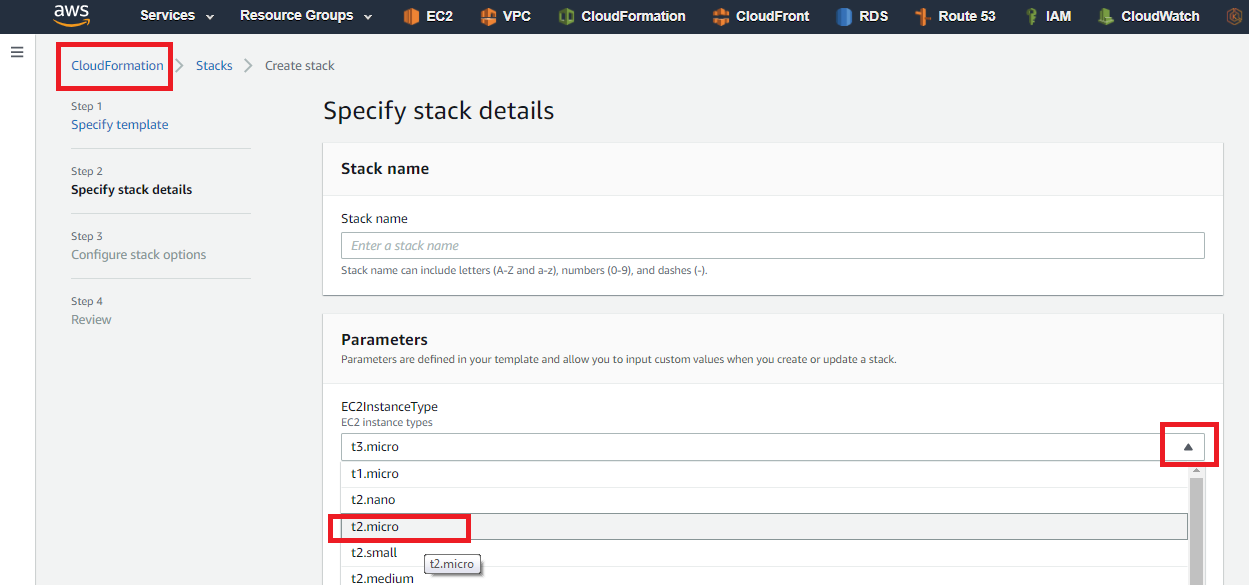
To acquire knowledge about: DNS Concepts, Route 53 Hosted Zones, and Simple and Geolocation Routing Policies.

# Labs10: Lab Prerequisites (Option 1: Using Cloudformation)

It is mandatory to have a keypair before to launch this template.

You can use the instructions of previous laboratory (Labs9c1) to create common infrastructure, with the recommendation to use t2.micro instances to use free tier for this lab.

There are subtle difference from the message that appear on the containers.



# Labs10: Lab Prerequisites (Option 2: Using CLI)

## Network Infrastructure and Creation of Instance

rem Prerequisito tener el putty completo en el path

rem Infrastructura de Red e Instancias Publicas

rem Se van a crear 2 instancias publicas en una misma AZ y subnet. Sera necesario ingresar con Docker para ejecutar nombre diferentes y diferenciarlas.

rem Clave del usuario

set AWS\_ACCESS\_KEY\_ID=AKIA3FYYCIHBXIBW5W25

set AWS\_SECRET\_ACCESS\_KEY=49d6BXXmTuY4MiuUGtg+pKsoXTO0Vv7ImPLspT53

set AWS\_DEFAULT\_REGION=us-east-1

rem Variables de configuracion

set vpcn\_Mask="10.0.0.0/16"

set pbsn1\_Mask="10.0.0.0/24"

set first\_az="us-east-1a"

set instance\_type="t2.micro"

rem Crear la VPC y habilitar resolucion DNS

aws ec2 create-vpc --cidr-block %vpcn\_Mask%|jq ".Vpc.VpcId" >tmpFile

set /p vpcn\_Id= < tmpFile

aws ec2 modify-vpc-attribute --vpc-id %vpcn\_Id% --enable-dns-hostnames "{\"Value\":true}"

rem Crear subred Publica 1

aws ec2 create-subnet --vpc-id %vpcn\_Id% --cidr-block %pbsn1\_Mask% --availability-zone %first\_az%|jq ".Subnet.SubnetId" >tmpFile

set /p pbsn1\_Id= < tmpFile

aws ec2 modify-subnet-attribute --subnet-id %pbsn1\_Id% --map-public-ip-on-launch

rem Crear el Internet Gateway IGW y asignarlo a la VPC

aws ec2 create-internet-gateway|jq ".InternetGateway.InternetGatewayId"  >tmpFile

set /p IGW\_Id= < tmpFile

aws ec2 attach-internet-gateway --vpc-id %vpcn\_Id% --internet-gateway-id %IGW\_Id%

rem Crear tabla de ruteo publica y asignarle IGW como ruta por defecto

aws ec2 create-route-table --vpc-id %vpcn\_Id%|jq ".RouteTable.RouteTableId" >tmpFile

set /p Public\_RT\_Id= < tmpFile

aws ec2 create-route --route-table-id %Public\_RT\_Id% --destination-cidr-block 0.0.0.0/0 --gateway-id %IGW\_Id%

rem Asociar la tabla de ruta a la subred

aws ec2 associate-route-table  --subnet-id %pbsn1\_Id% --route-table-id %Public\_RT\_Id%

rem Crear las llaves para el SSH a las nuevas instancias y convertirlas a PPK para usar Putty ya sea con puttygen o winscp

aws ec2 create-key-pair --key-name Lab10a --query "KeyMaterial" --output text > Lab10a.pem

winscp.com /keygen "Lab10a.pem" /output="Lab10a.ppk"

rem Crear los Security Groups para esa instancia

aws ec2 create-security-group --group-name "SecGroup\_A" --description "Security group for Instance A" --vpc-id %vpcn\_Id% |jq ".GroupId">tmpFile

set /p SecGroup\_A\_Id= < tmpFile

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 22 --cidr 0.0.0.0/0

aws ec2 authorize-security-group-ingress --group-id %SecGroup\_A\_Id% --protocol tcp --port 80 --cidr 0.0.0.0/0

rem En el laboratorio de EC2 Inicial se mostrar la importancia de buscar una AMI correcto.

rem AWS sugiere que se tome el AMI Amazon Linux 2 y se instale docker desde linea de comandos: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html#install\_docker

aws ec2 describe-images --owners amazon --filters "Name=name,Values=amzn2-ami-hvm-2.0.????????.?-x86\_64-gp2" "Name=state,Values=available" --query "reverse(sort\_by(Images, &CreationDate))[:1].ImageId" --output text >tmpFile

set /p AMI= < tmpFile

rem Se solicitan instancias y se adiciona un bootstrap para comprobar que el docker fue instalado

aws ec2 run-instances --image-id %AMI% --count 1 --instance-type %instance\_type% --key-name Lab10a --security-group-ids %SecGroup\_A\_Id% --subnet-id %pbsn1\_Id% --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=A}]"  --user-data file://bootstrap.sh |jq "[.Instances|.[].InstanceId|.]"|jq ".[0]" >tmpFile

set /p Instance1Id= <tmpFile

aws ec2 run-instances --image-id %AMI% --count 1 --instance-type %instance\_type% --key-name Lab10a --security-group-ids %SecGroup\_A\_Id% --subnet-id %pbsn2\_Id% --tag-specifications "ResourceType=instance,Tags=[{Key=Name,Value=B}]"  --user-data file://bootstrap.sh |jq "[.Instances|.[].InstanceId|.]"|jq ".[0]" >tmpFile

set /p Instance2Id= <tmpFile

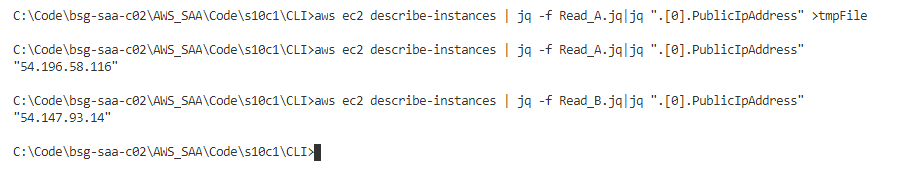
rem Traer Datos especificos de instancia A. Revisar contenido describe-instances y Read\_A.jq ya que es diferente a lo de anteriores laboratorios.

aws ec2 describe-instances | jq -f Read\_A.jq|jq ".[0].PublicIpAddress" >tmpFile

set /p A\_IP= < tmpFile

aws ec2 describe-instances | jq -f Read\_B.jq|jq ".[0].PublicIpAddress" >tmpFile

set /p B\_IP= < tmpFile



## Differentiate EC2 Instances

rem Ingresar a ambas instancias publica por SSH. Ejecutar las mismas acciones y despues ir al navegador a ver que funcionan las IPs

putty.exe -i "Lab10a.ppk" ec2-user@%A\_IP%

putty.exe -i "Lab10a.ppk" ec2-user@%B\_IP%

rem Para ambas instancias. Comprobar la instalacion de Docker y borramos cualquier contenedor anterior

sudo yum update -y

sudo amazon-linux-extras install docker -y

sudo service docker start

sudo usermod -a -G docker ec2-user

rem Desconectarse del Putty y reconectarse.

rem Si necesita eliminar el contenedor anterior.

docker ps -a

docker stop $(docker ps -aq)

docker rm $(docker ps -aq)

rem Ejercutar variables comunes para cada una de las instancias

export AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)

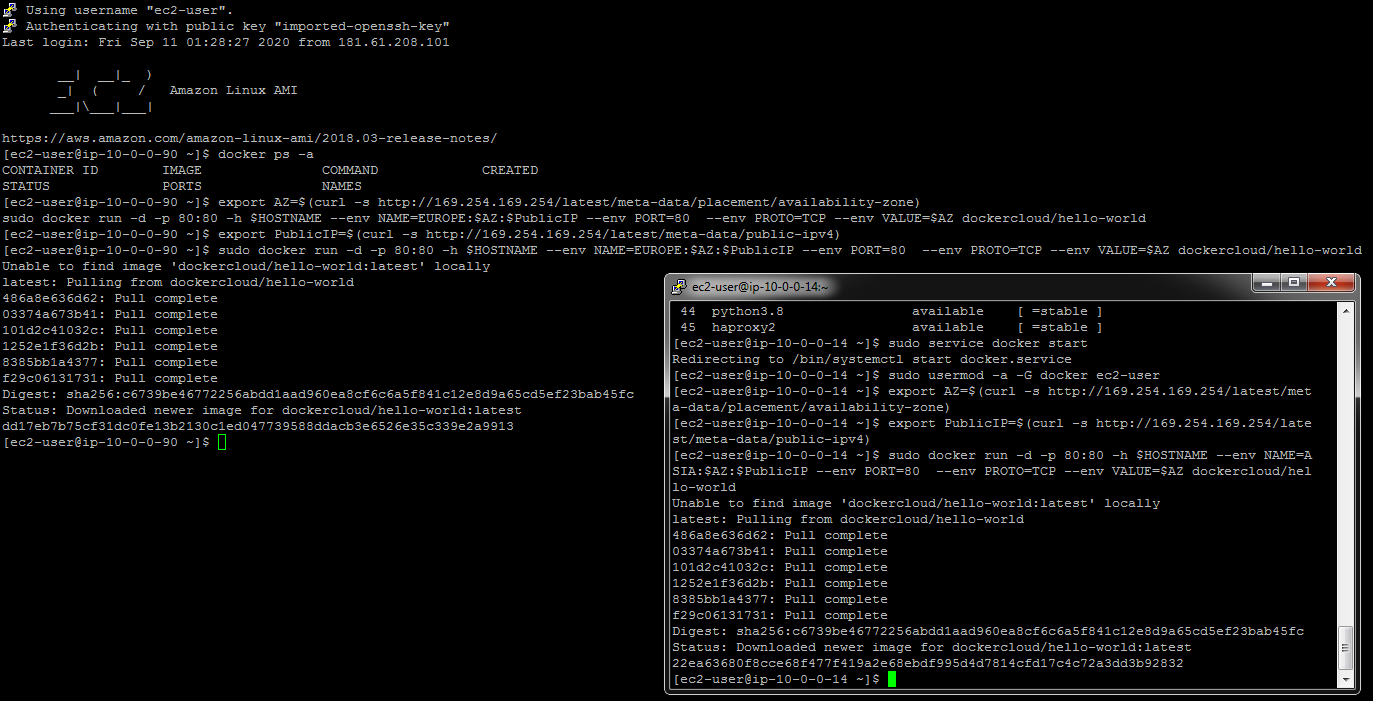
export PublicIP=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)

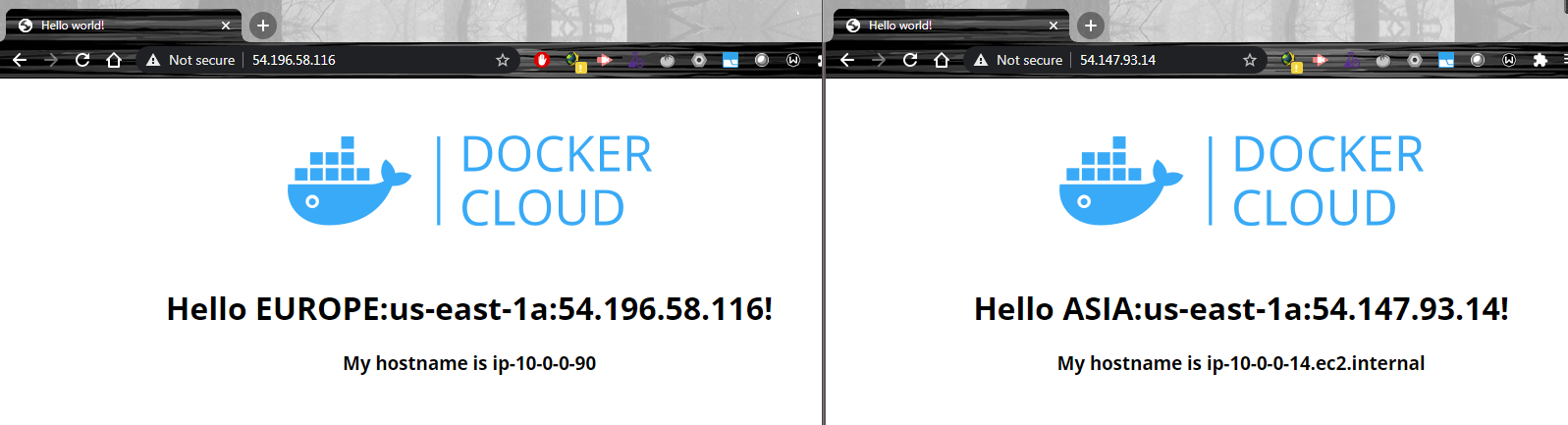
rem para la Instancia A

sudo docker run -d -p 80:80 -h $HOSTNAME --env NAME=EUROPE:$AZ:$PublicIP --env PORT=80  --env PROTO=TCP --env VALUE=$AZ dockercloud/hello-world

rem para la Instancia B

sudo docker run -d -p 80:80 -h $HOSTNAME --env NAME=ASIA:$AZ:$PublicIP --env PORT=80  --env PROTO=TCP --env VALUE=$AZ dockercloud/hello-world

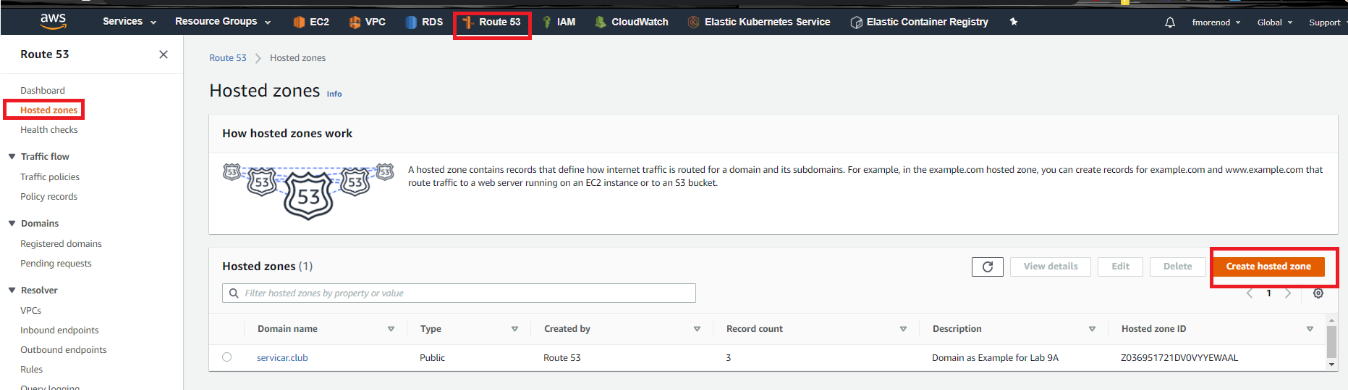


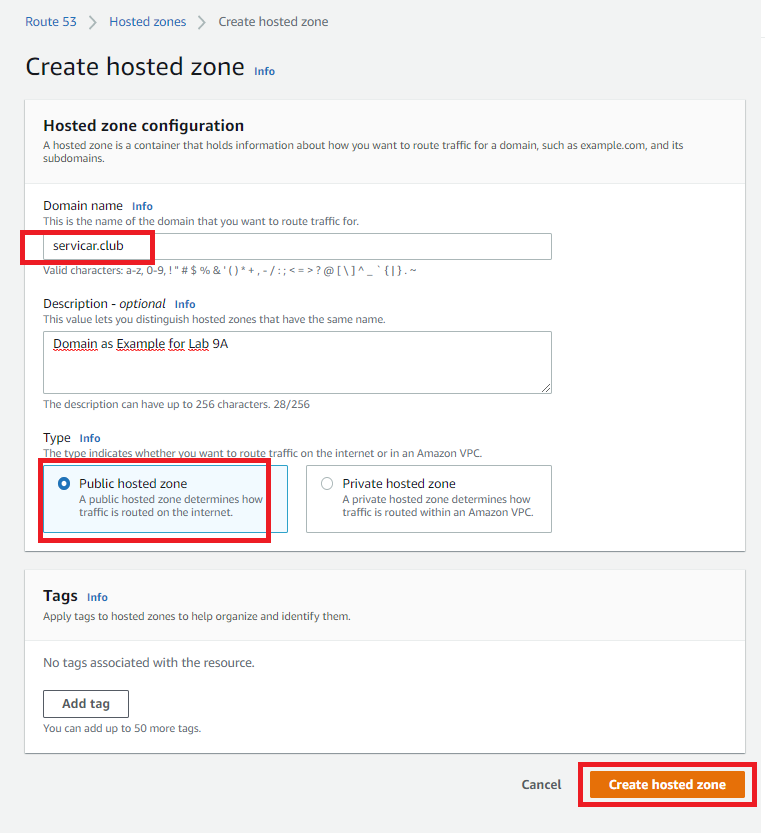


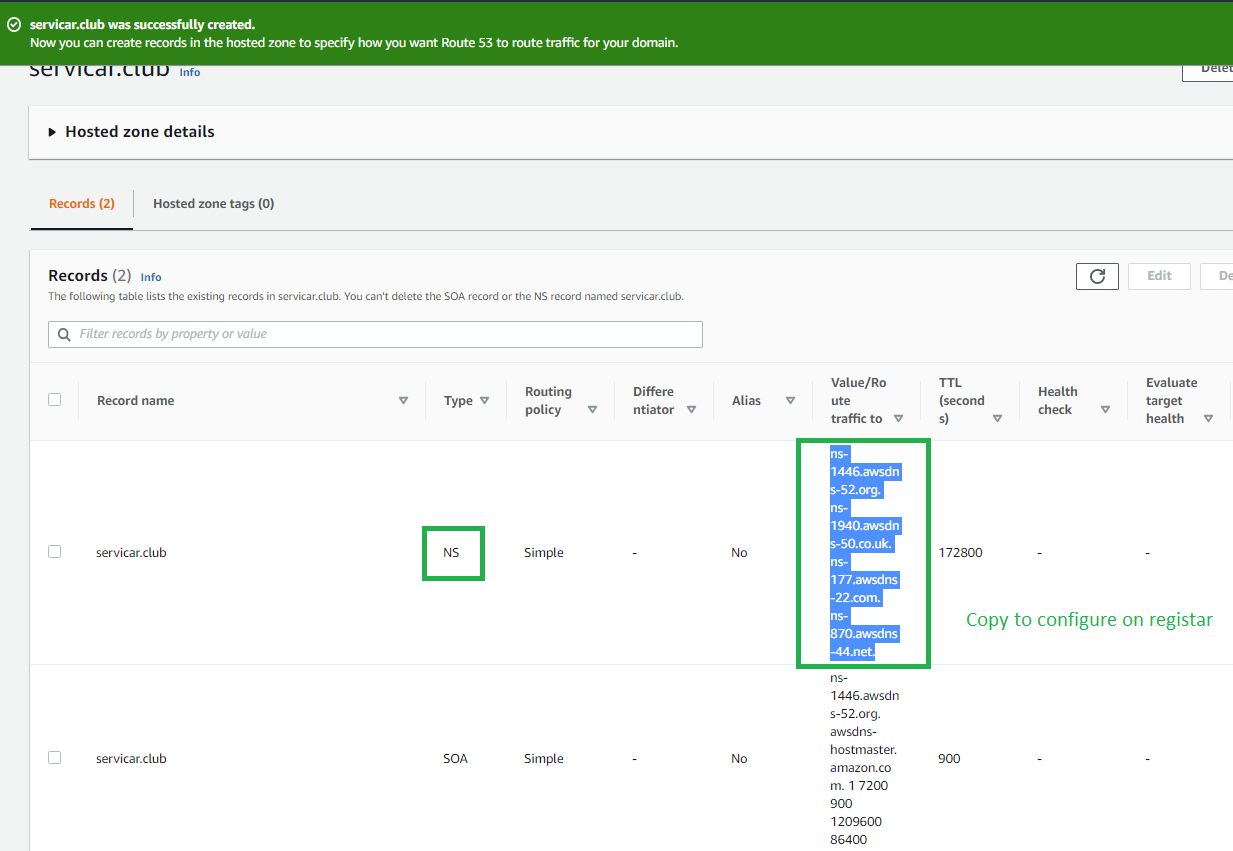
# Lab 10c1: Created Simple and Geolocation Routing Policy

## Create Hosted Zone

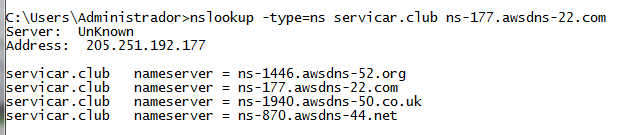
You have to create a Hosted Zone with same name as domain that you bought.





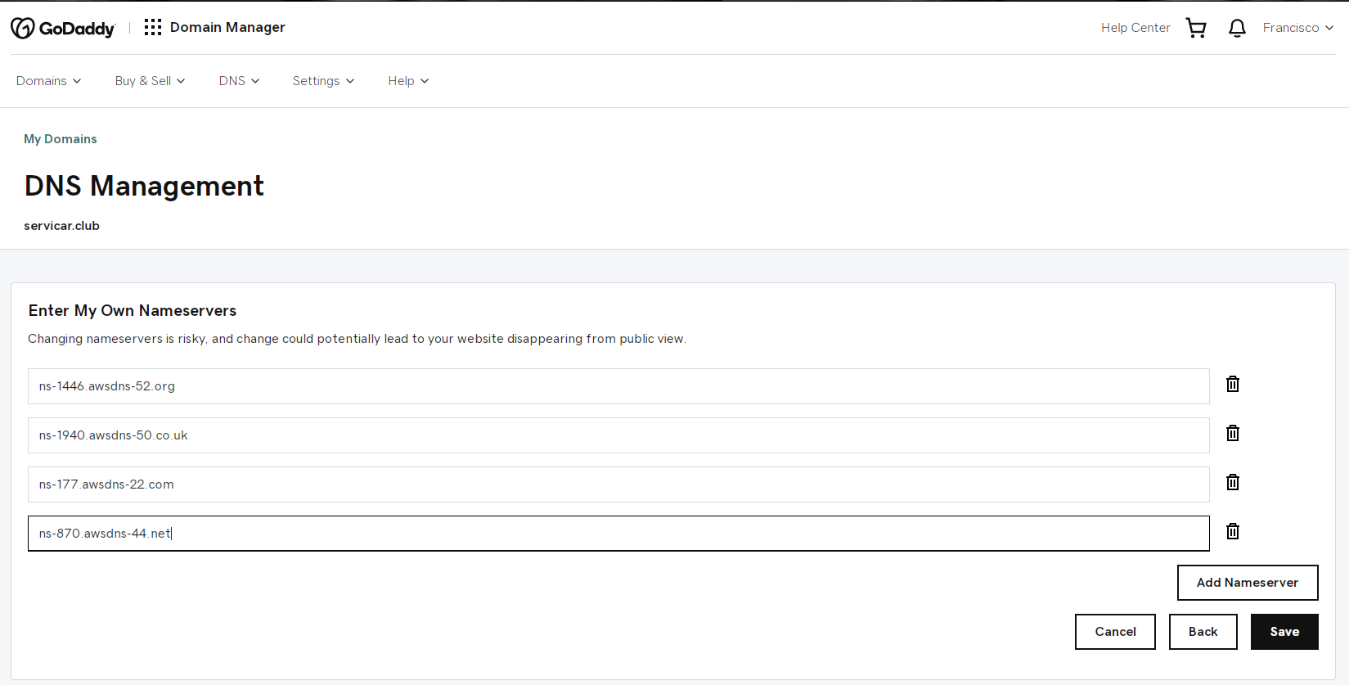


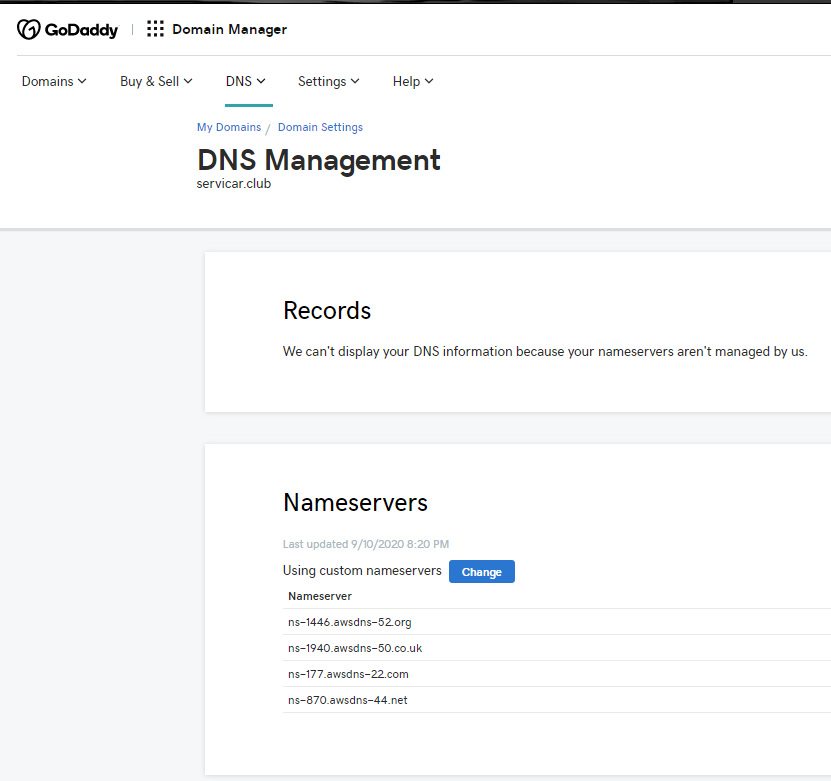
You get nameservers that managed that domain but those aren’t public, you have to publish using the domain registar that you have.



## External Registrar (GoDaddy)

Going to GoDaddy (Domain Registrar) and configure the new nameservers that you get on the Hosted Zone. In addition, you can buy the domain on Route 53 on Domains >> Registered Domains instead of having this step.

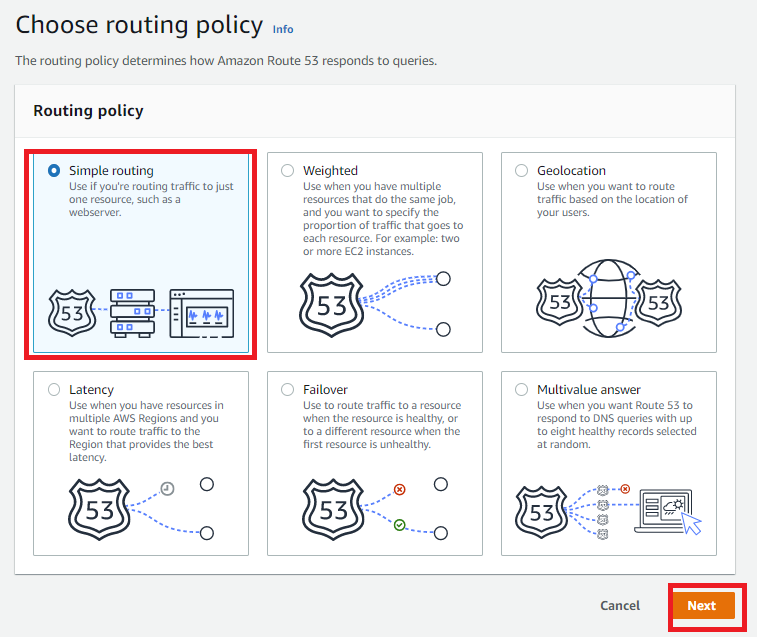




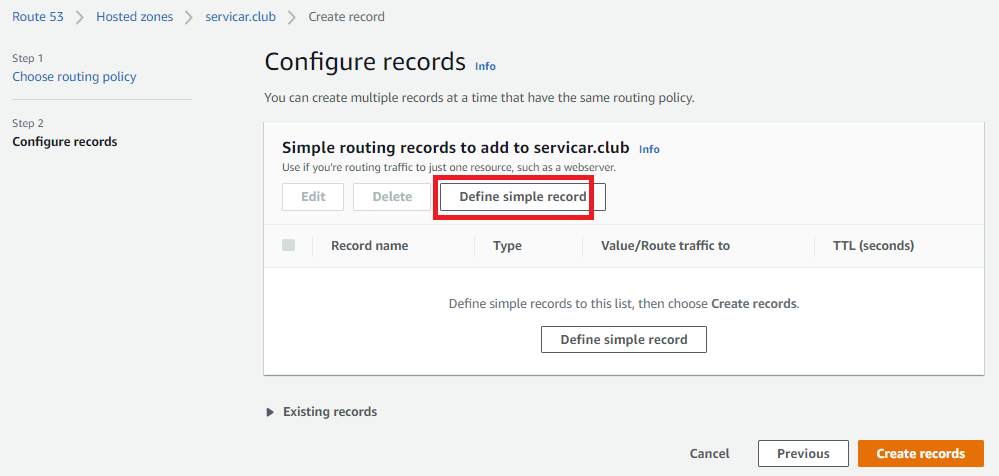
Now Route 53 have the control to create records and manage the domain.

## Create the Simple Routing Policy

When you are on the Hosted Zone, click on Create Record and then, Choose Simple Routing.

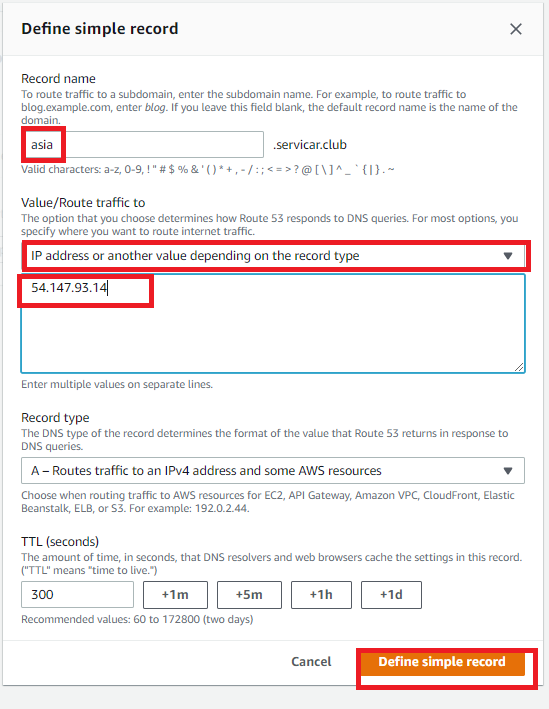


Click on Define Simple Records

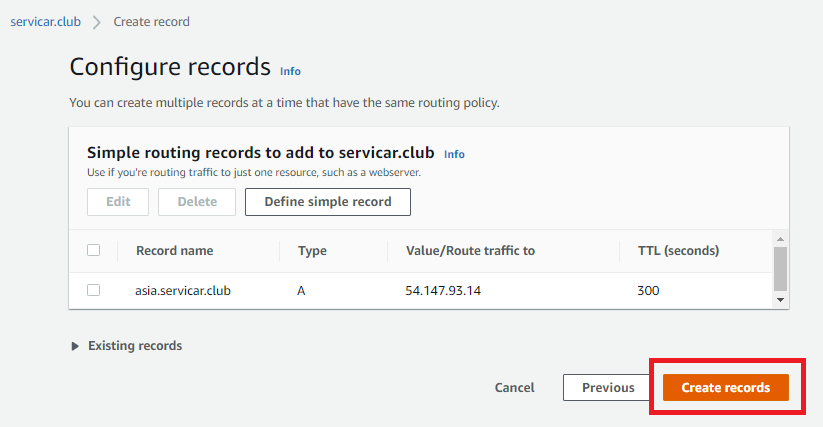


Then, fill the form using the new subdomain, IP that you directed and let Record type on A.

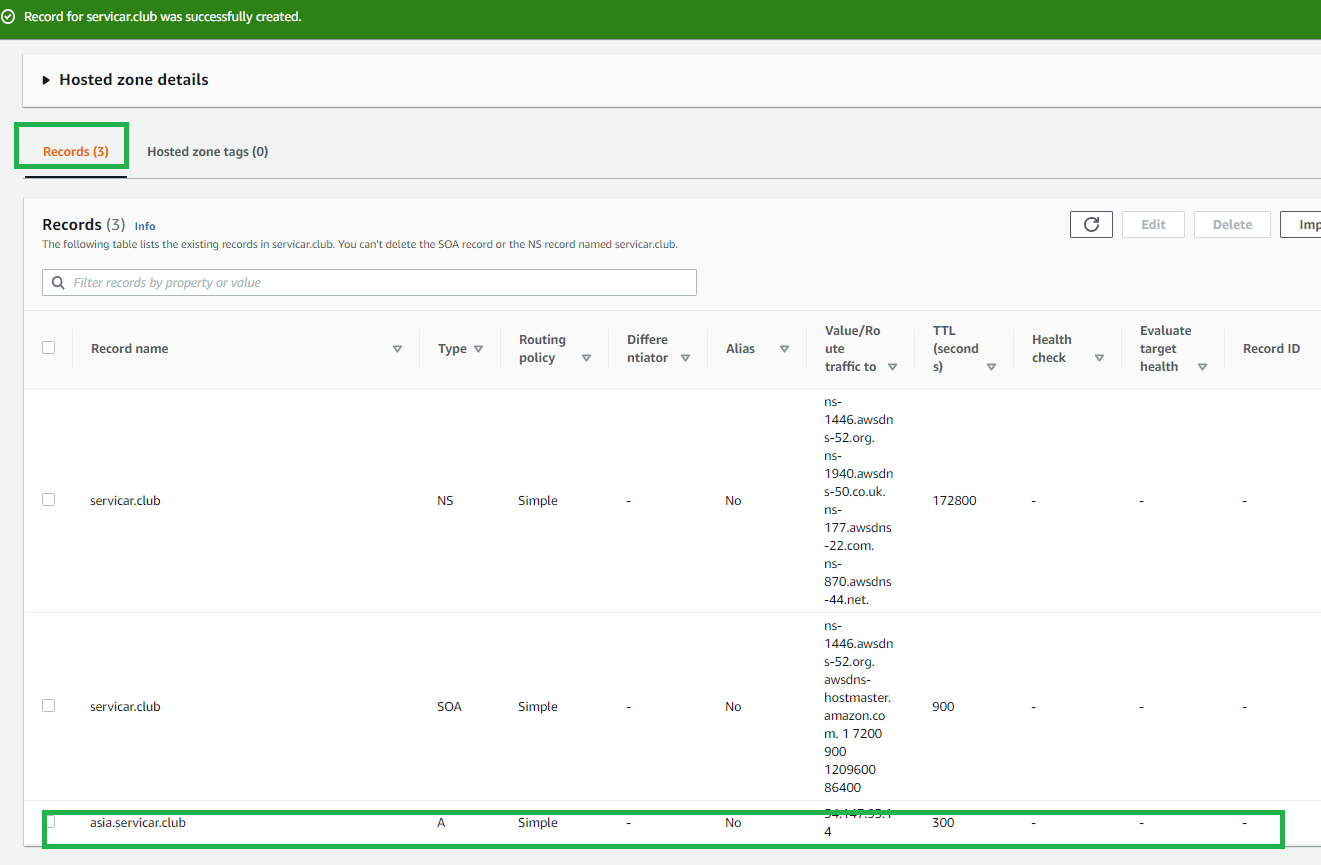
Click on Define Simple Record.



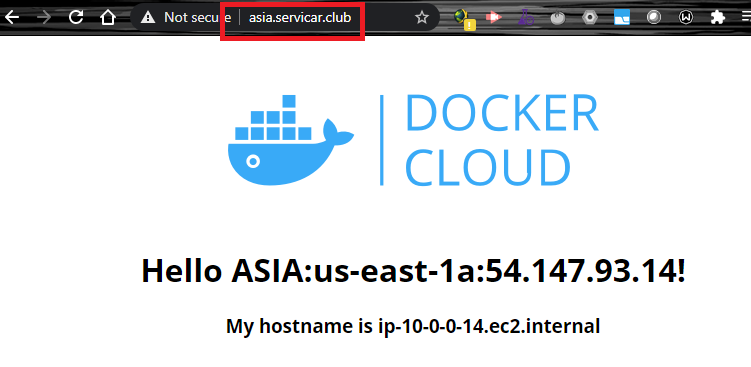
On main screen, click on Create Records



A New Record appear on Main Screen of Hosted Zone,

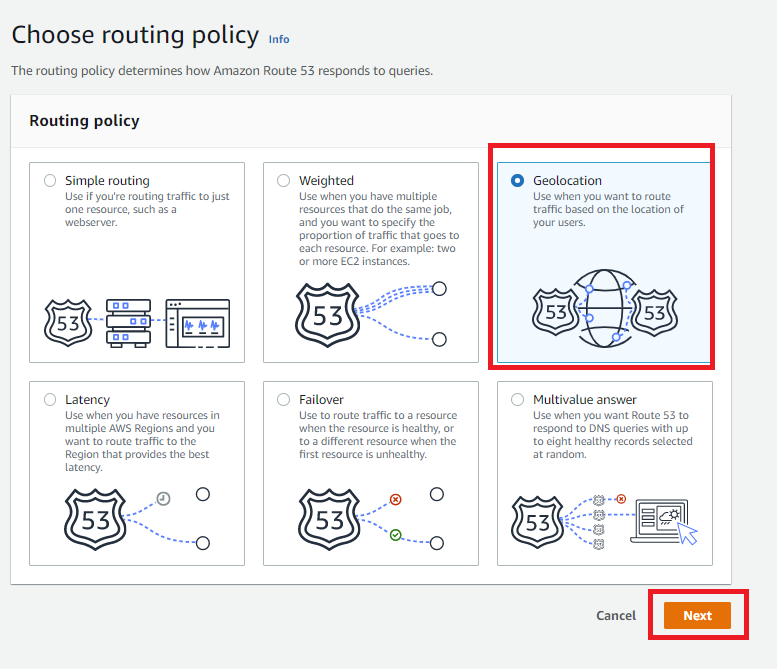


You can tested on your favorite browser,

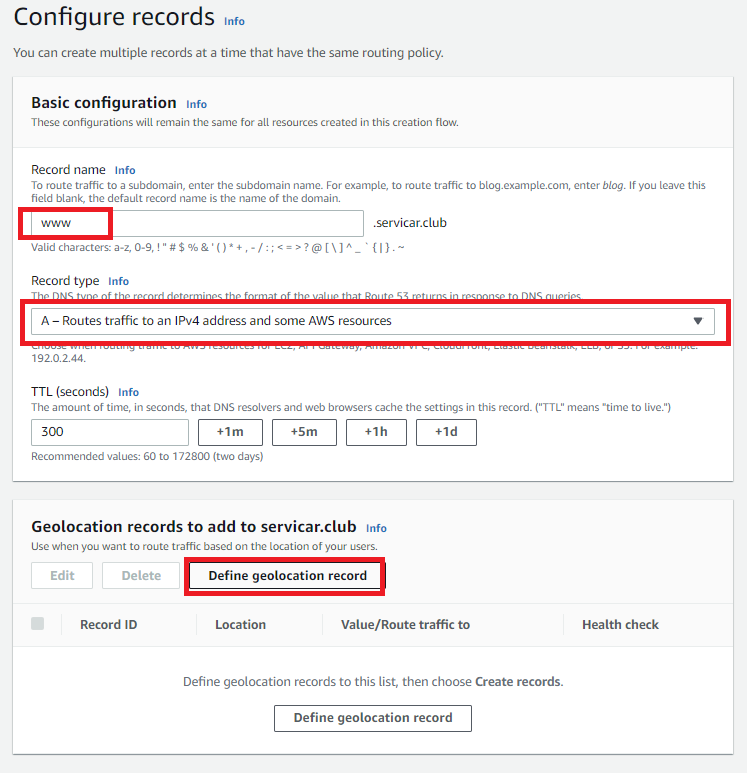


## Create a Geolocation Reouting Policy

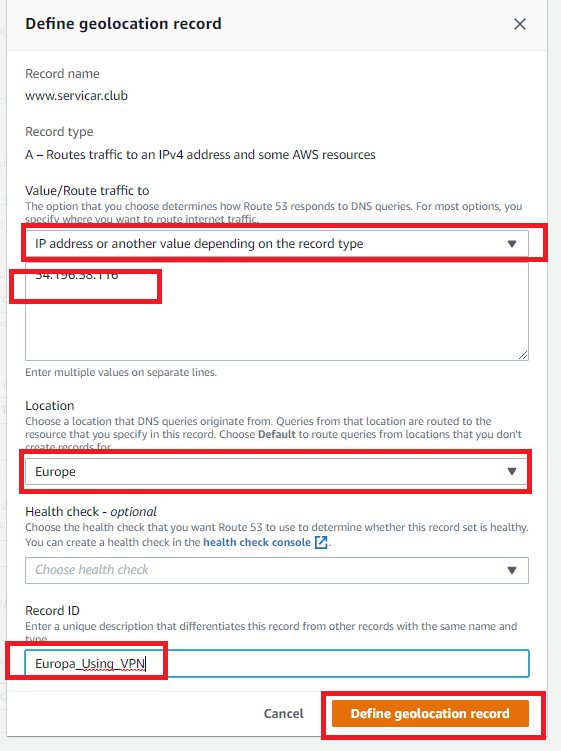
On Route 53 for the Hosted Zone, click on Create Record, and then click on Geolocation.



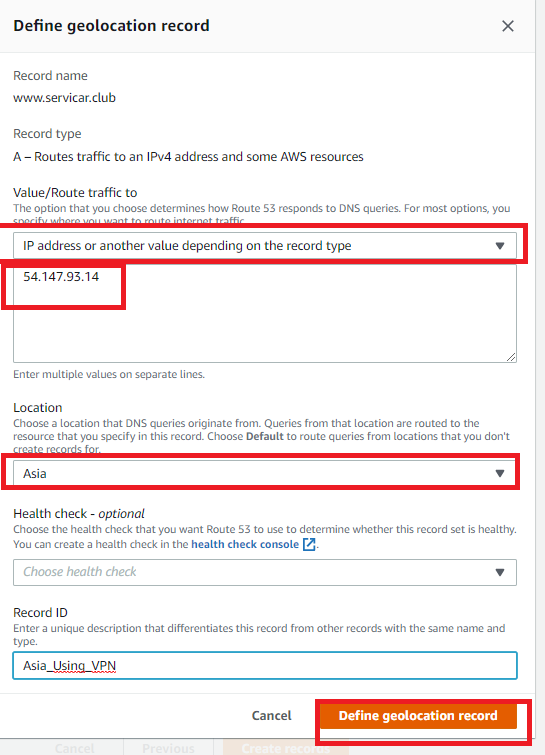
Configure the Subdomain that you like to manage and its record type (A); then click on Define geolocation record where you stablish the routes based on source ip



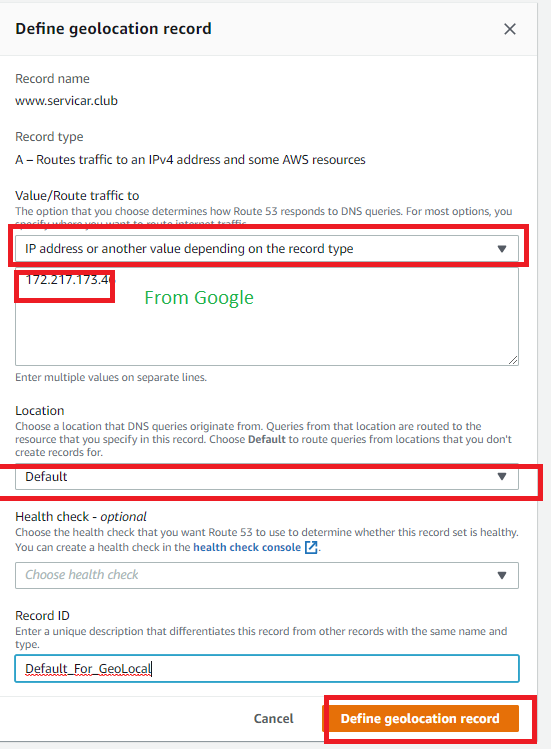
For Source IPs coming from Europe configure the public IP of Europe Instance,



For Asia, the same thing,



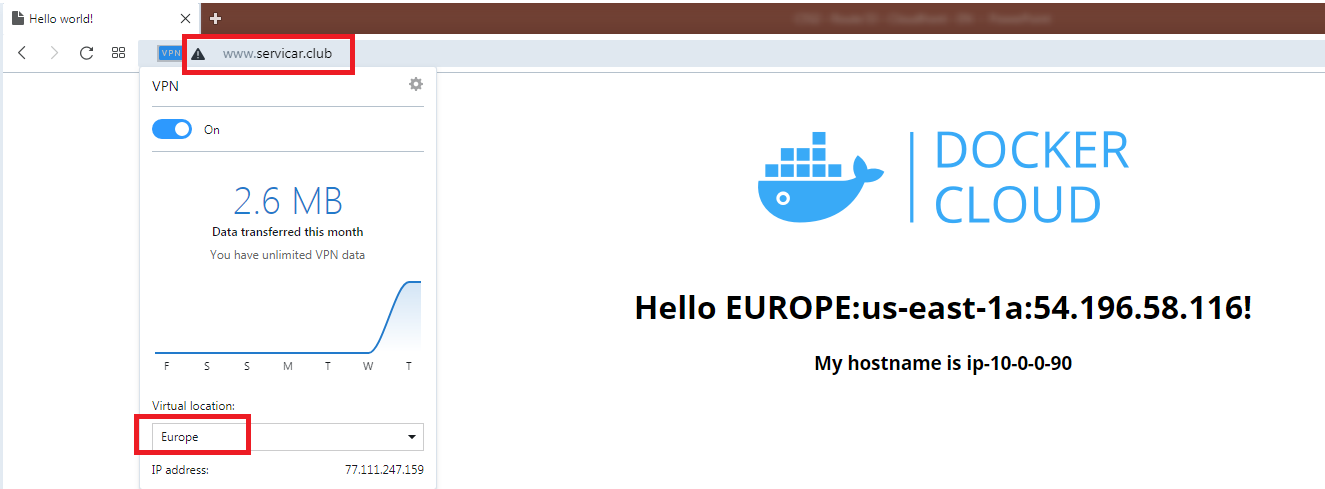
On default, I get a public IP from Google but you can use anything else,

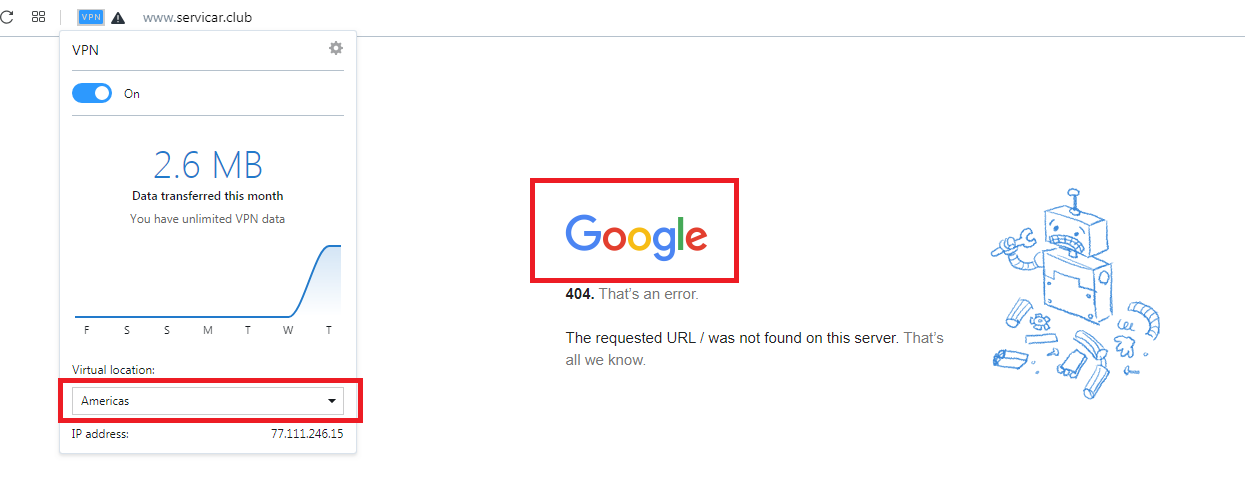
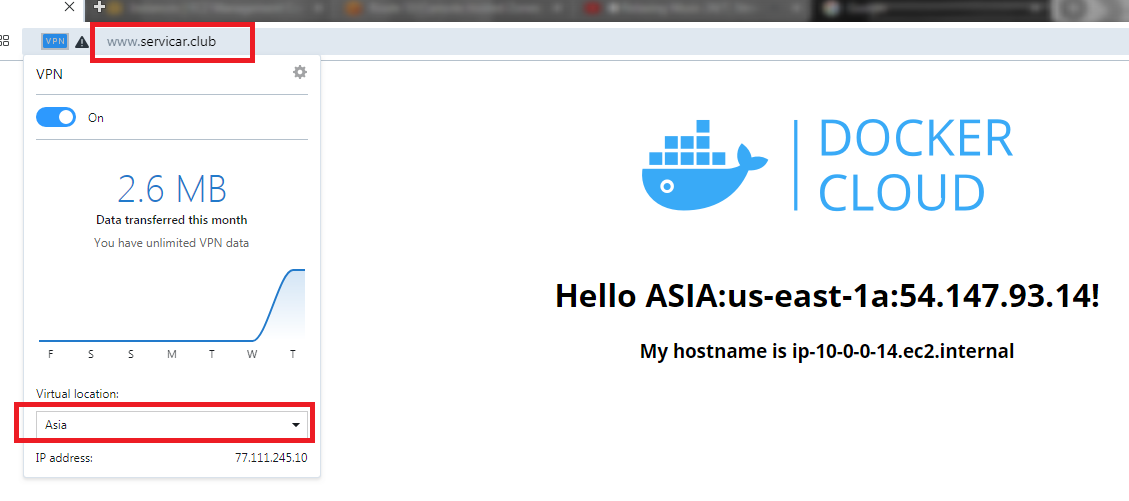


When you come back to main screen of Records you click on Create Records.

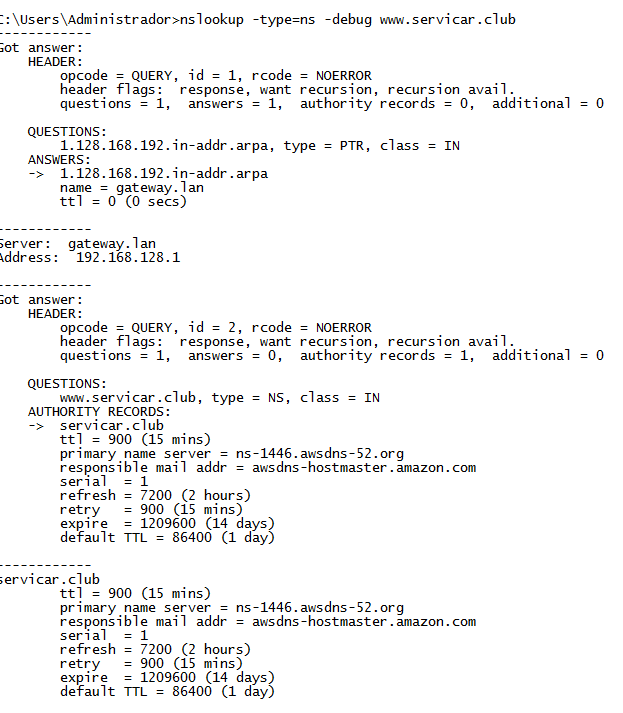
## Tested the routing policies

Using Opera with Free VPN Activated, you can tested the routing policies





Even though, if you like to know which server reply your DNS query you can use CLI:



# Evidences to send

To have a review, the student has to send some screenshots to instructor email:

1. Simple Routing record from Route 53.
2. Working Docker Containers using the Simple Routing.
3. One of the records of the Geolocation Routing Policy.
4. One of Opera using VPN of Asia or Europe pointing to the right Docker Container.

# Clean Resources

rem ----- ELIMINAR RECURSOS ----

rem Entiendo que despues de 12h se empieza a cobrar Route 53

rem Eliminar Route 53 Hosted Zone como primer paso

rem En Cloudformation, seleccionar el Stack y borrarlo.

aws ec2 terminate-instances --instance-ids <Codigo de las Instancias>

aws ec2 delete-security-group --group-id %SecGroup\_A\_Id%

aws ec2 detach-internet-gateway --internet-gateway-id %IGW\_Id% --vpc-id %vpcn\_Id%

aws ec2 delete-internet-gateway --internet-gateway-id %IGW\_Id%

aws ec2 delete-subnet --subnet-id %pbsn1\_Id%

aws ec2 delete-route-table --route-table-id %Public\_RT\_Id%

aws ec2 delete-vpc --vpc-id %vpcn\_Id%

aws ec2 delete-key-pair --key-name Lab10a