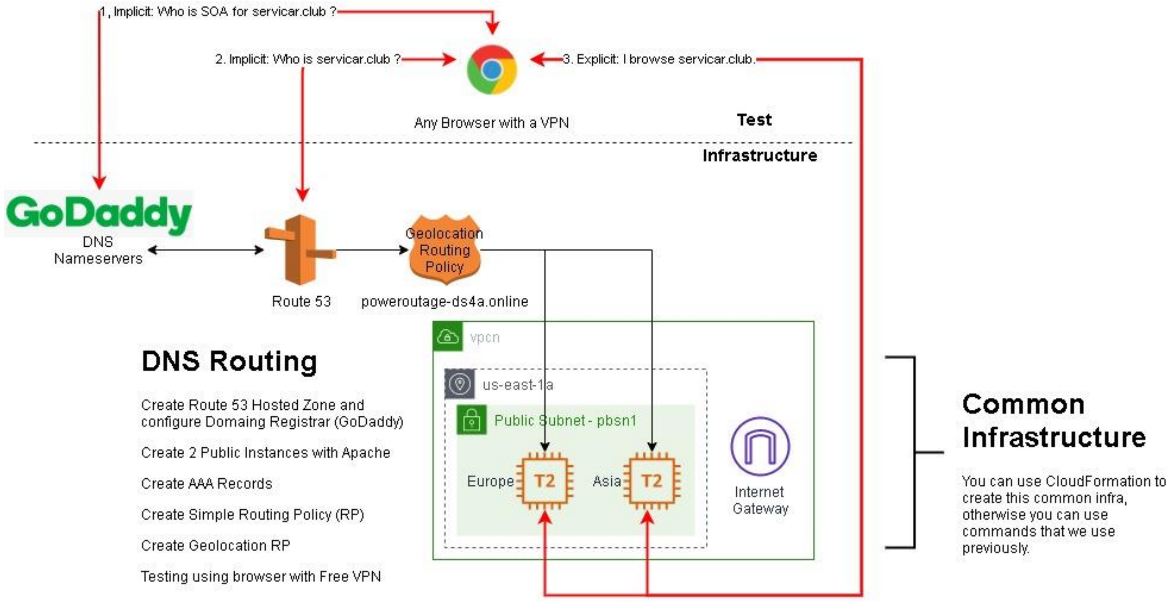
Contents

General Diagram



DNS Routing

Create Route 53 Hosted Zone and configure Domaing Registrar (GoDaddy)

Create 2 Public Instances with Apache

Create AAA Records

Create Simple Routing Policy (RP)

Create Geolocation RP

Testing using browser with Free VPN

Common Infrastructure

You can use CloudFormation to create this common infra, otherwise you can use commands that we use previously.

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) 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.

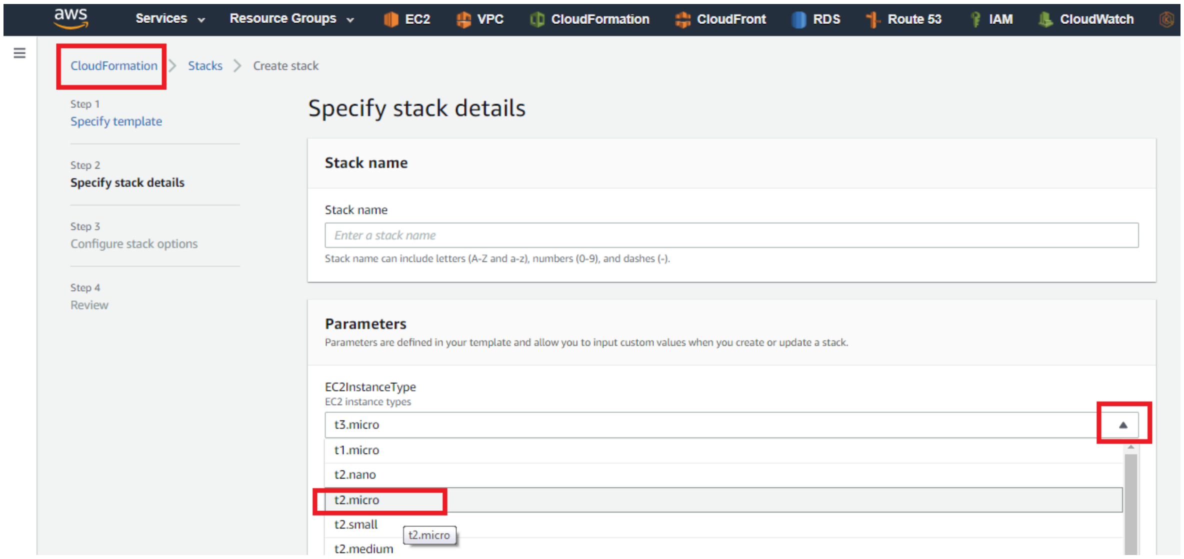
To acquire knowledge about: DNS Concepts, Route 53 Hosted Zones, and Simple and Geolocation Routing Policies.

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.



Specify stack details

Stack name

Stack name

Enter a stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EC2InstanceType

EC2 instance types

t3.micro

t1.micro

t2.nano

t2.micro

t2.small

t2.micro

t2.medium

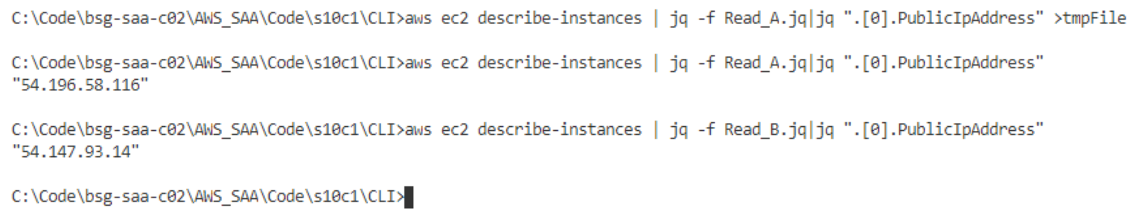
Labs10: Lab Prerequisites (Option 2: Using CLI)

Network Infrastructure and Creation of Instance

```
rem Prerequisito tener el putty completo en el path
rem Infraestructura de Red e Instancias Publicas
rem Se van a crear 2 instancias publicas en una misma AZ y subnet. Sera nece
sario ingresar con Docker para ejecutar nombre diferentes y diferenciarlas.

rem Clave del usuario
set AWS_ACCESS_KEY_ID=AKIA3FYYCIHBXIBW5W25
set AWS_SECRET_ACCESS_KEY=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
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
```

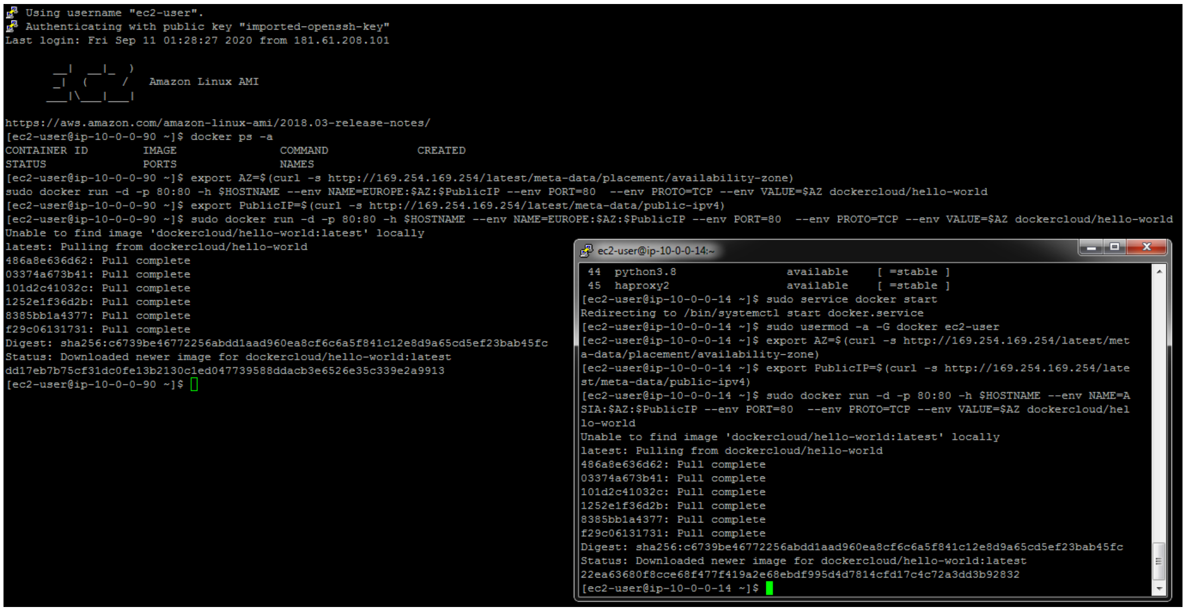
```
rem En el laboratorio de EC2 Inicial se mostrar la importancia de buscar una AMI correcto.
rem AWS sugiere que se tome el AMI Amazon Linux 2 y se instale docker desde linea de comandos: https://docs.aws.amazon.com/AmazonECS/latest/developerguide/docker-basics.html#install_docker
aws ec2 describe-images --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
```



```
C:\Code\bsg-saa-c02\AWS_SAA\Code\s10c1\CLI>aws ec2 describe-instances | jq -f Read_A.jq|jq ".[0].PublicIpAddress" >tmpFile

C:\Code\bsg-saa-c02\AWS_SAA\Code\s10c1\CLI>aws ec2 describe-instances | jq -f Read_A.jq|jq ".[0].PublicIpAddress"
"54.196.58.116"

C:\Code\bsg-saa-c02\AWS_SAA\Code\s10c1\CLI>aws ec2 describe-instances | jq -f Read_B.jq|jq ".[0].PublicIpAddress"
"54.147.93.14"

C:\Code\bsg-saa-c02\AWS_SAA\Code\s10c1\CLI>
```

Differentiate EC2 Instances

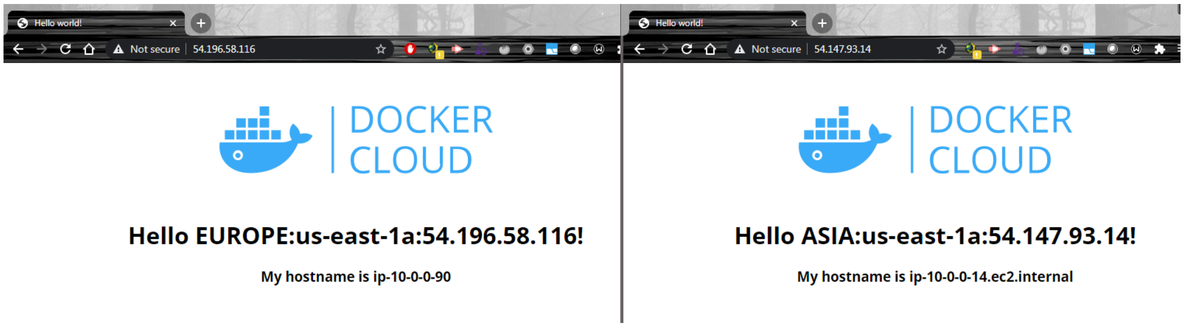
```
rem Ingresar a ambas instancias publica por SSH. Ejecutar las mismas accione
s y despues ir al navegador a ver que funcionan las IPs
putty.exe -i "Lab10a.ppk" ec2-user@%A_IP%
putty.exe -i "Lab10a.ppk" ec2-user@%B_IP%
rem Para ambas instancias. Comprobar la instalacion de Docker y borramos cua
lquier 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
```



```
Using username "ec2-user".
Authenticating with public key "imported-openssh-key"
Last login: Fri Sep 11 01:28:27 2020 from 181.61.208.101

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2018.03-release-notes/
[ec2-user@ip-10-0-0-90 ~]$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS               NAMES
[ec2-user@ip-10-0-0-90 ~]$ export AZ=$(curl -s http://169.254.169.254/latest/meta-data/placement/availability-zone)
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
[ec2-user@ip-10-0-0-90 ~]$ export PublicIP=$(curl -s http://169.254.169.254/latest/meta-data/public-ipv4)
[ec2-user@ip-10-0-0-90 ~]$ 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
Unable to find image 'dockercloud/hello-world:latest' locally
latest: Pulling from dockercloud/hello-world
486a8e636d62: Pull complete
03374a673b41: Pull complete
101d2c41032c: Pull complete
1252e1f36d2b: Pull complete
8385bb1a4377: Pull complete
f29c06131731: Pull complete
Digest: sha256:c6739be46772256abdd1aad960ea8cf6c6a5f841c12e8d9a65cd5ef23bab45fc
Status: Downloaded newer image for dockercloud/hello-world:latest
dd17eb7b75cf31dc0fe13b2130c1ed047739588ddacb3e6526e35c339e2a9913
[ec2-user@ip-10-0-0-90 ~]$
```

ec2-user@ip-10-0-0-14:~

```
 44  python3.8                available    [ =stable ]
 45  haproxy2                 available    [ =stable ]
[ec2-user@ip-10-0-0-14 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-10-0-0-14 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-10-0-0-14 ~]$ export AZ=$(curl -s http://169.254.169.254/latest/met
a-data/placement/availability-zone)
[ec2-user@ip-10-0-0-14 ~]$ export PublicIP=$(curl -s http://169.254.169.254/late
st/meta-data/public-ipv4)
[ec2-user@ip-10-0-0-14 ~]$ sudo docker run -d -p 80:80 -h $HOSTNAME --env NAME=A
SIA:$AZ:$PublicIP --env PORT=80  --env PROTO=TCP --env VALUE=$AZ dockercloud/hel
lo-world
Unable to find image 'dockercloud/hello-world:latest' locally
latest: Pulling from dockercloud/hello-world
486a8e636d62: Pull complete
03374a673b41: Pull complete
101d2c41032c: Pull complete
1252e1f36d2b: Pull complete
8385bb1a4377: Pull complete
f29c06131731: Pull complete
Digest: sha256:c6739be46772256abdd1aad960ea8cf6c6a5f841c12e8d9a65cd5ef23bab45fc
Status: Downloaded newer image for dockercloud/hello-world:latest
22ea63680f8cce68f477f419a2e68ebdf995d4d7814cfd17c4c72a3dd3b92832
[ec2-user@ip-10-0-0-14 ~]$
```



Hello EUROPE:us-east-1a:54.196.58.116!

My hostname is ip-10-0-0-90

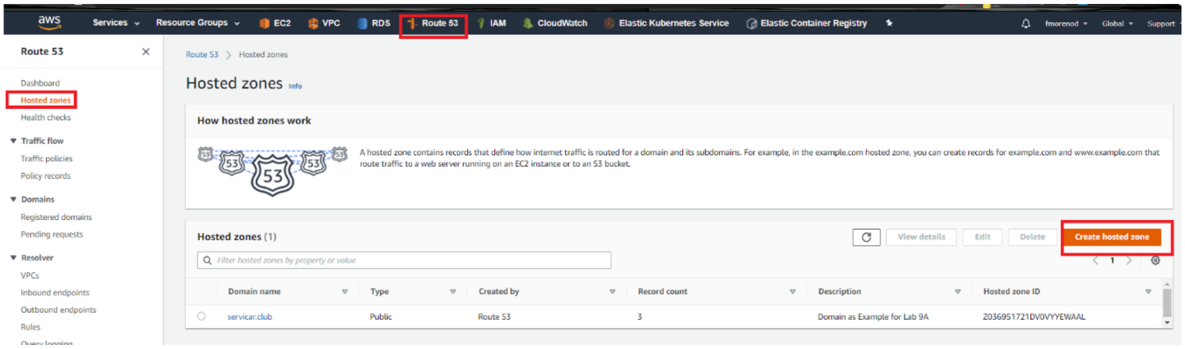
Hello ASIA:us-east-1a:54.147.93.14!

My hostname is ip-10-0-0-14.ec2.internal

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.



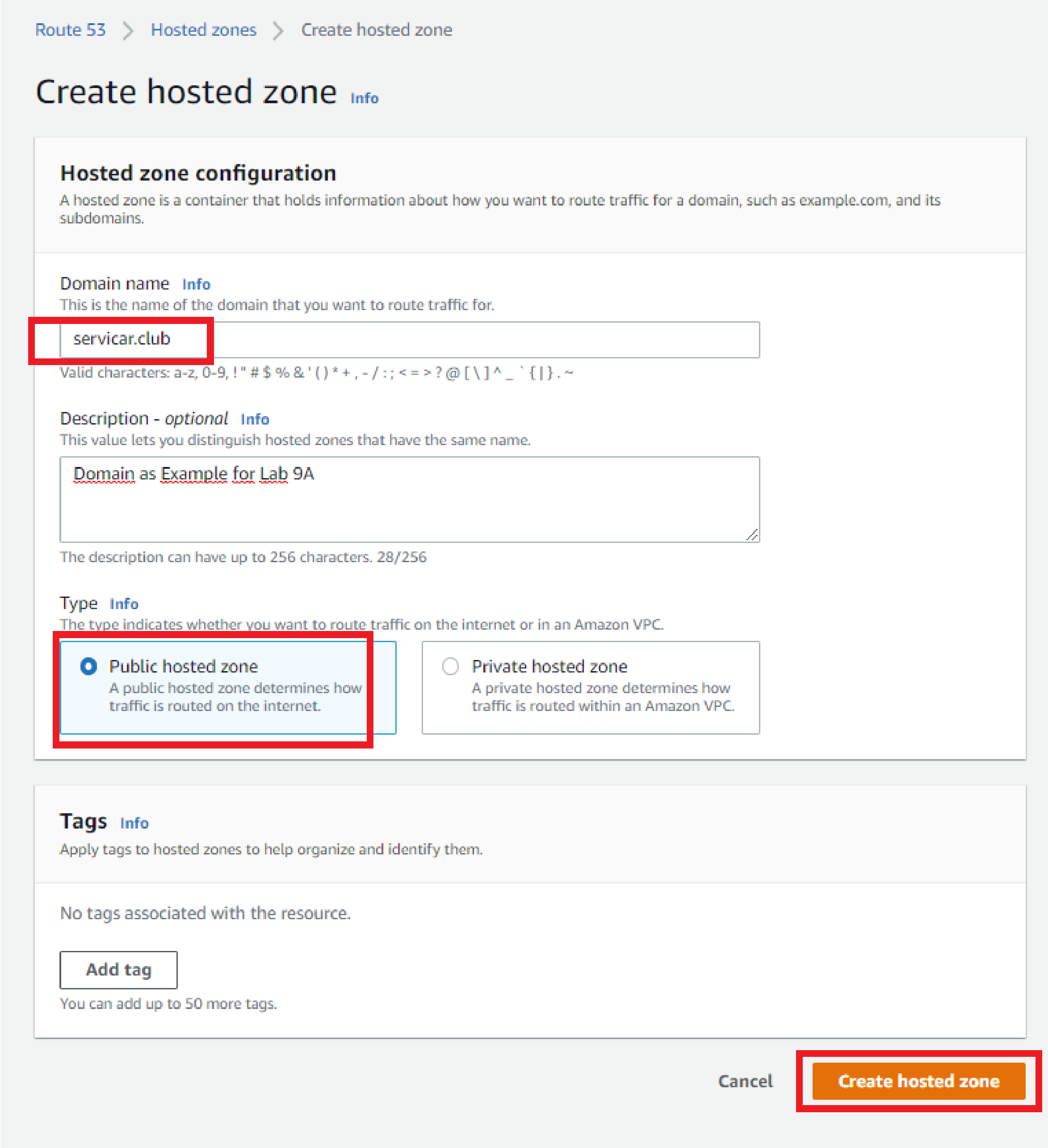
Hosted zones

How hosted zones work

A hosted zone contains records that define how internet traffic is routed for a domain and its subdomains. For example, in the example.com hosted zone, you can create records for example.com and www.example.com that route traffic to a web server running on an EC2 instance or to an S3 bucket.

Hosted zones (1)

	Domain name	Type	Created by	Record count	Description	Hosted zone ID
○	servicar.club	Public	Route 53	3	Domain as Example for Lab 9A	Z036951721DV0VYYEWAAL



Create hosted zone

Hosted zone configuration

A hosted zone is a container that holds information about how you want to route traffic for a domain, such as example.com, and its subdomains.

Domain name
This is the name of the domain that you want to route traffic for.

servicar.club

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Description - *optional*
This value lets you distinguish hosted zones that have the same name.

Domain as Example for Lab 9A

The description can have up to 256 characters. 28/256

Type
The type indicates whether you want to route traffic on the internet or in an Amazon VPC.

● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

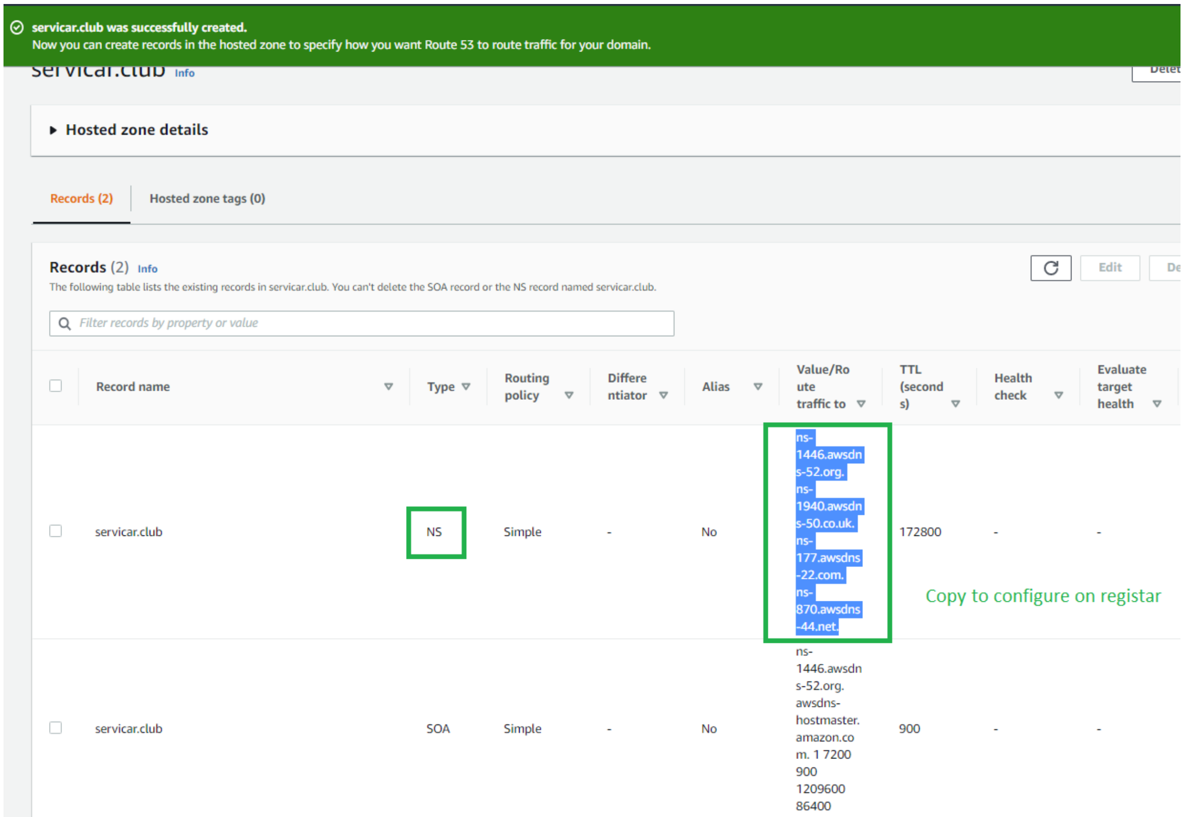
○ Private hosted zone
A private hosted zone determines how traffic is routed within an Amazon VPC.

Tags

Apply tags to hosted zones to help organize and identify them.

No tags associated with the resource.

You can add up to 50 more tags.



servicar.club was successfully created.
Now you can create records in the hosted zone to specify how you want Route 53 to route traffic for your domain.

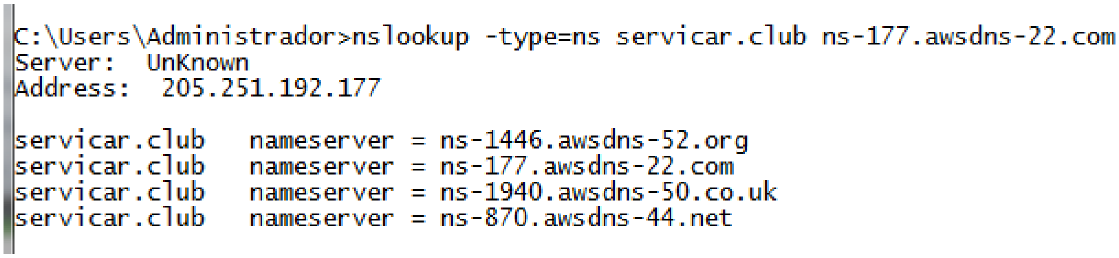
Records (2) Info

The following table lists the existing records in servicar.club. You can't delete the SOA record or the NS record named servicar.club.

☐	Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health
☐	servicar.club	NS	Simple	-	No	ns-1446.awsdns-52.org. ns-1940.awsdns-50.co.uk. ns-177.awsdns-22.com. ns-870.awsdns-44.net.	172800	-	-
☐	servicar.club	SOA	Simple	-	No	ns-1446.awsdns-52.org. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	900	-	-

Copy to configure on registar

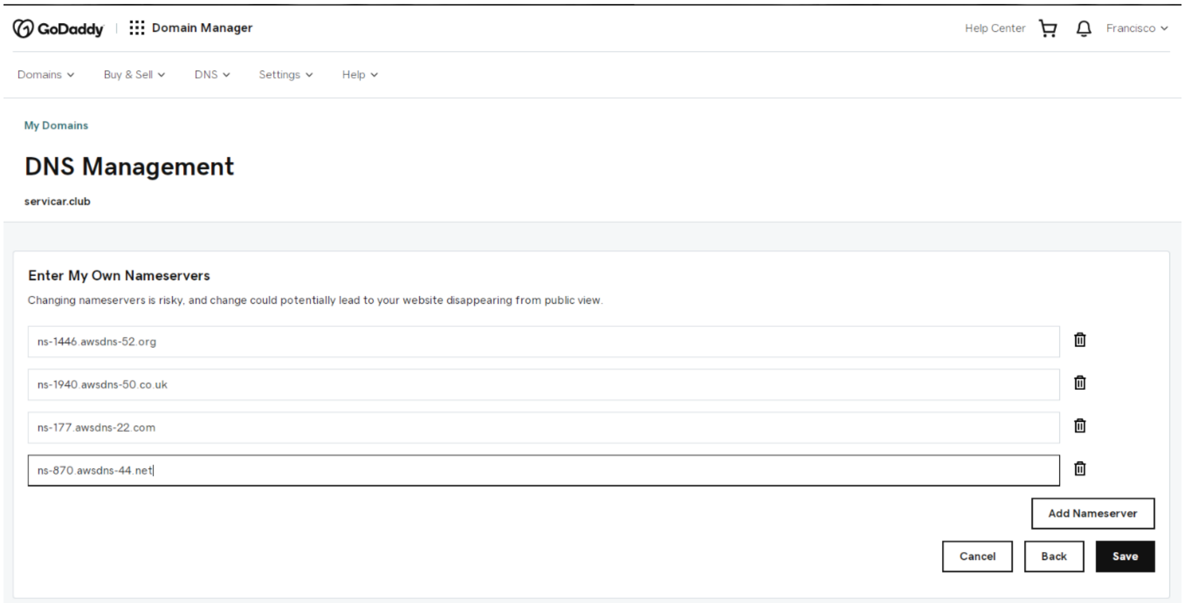
You get nameservers that managed that domain but those aren't public, you have to publish using the domain registar that you have.



```
C:\Users\Administrador>nslookup -type=ns servicar.club ns-177.awsdns-22.com
Server:  UnKnown
Address:  205.251.192.177

servicar.club   nameserver = ns-1446.awsdns-52.org
servicar.club   nameserver = ns-177.awsdns-22.com
servicar.club   nameserver = ns-1940.awsdns-50.co.uk
servicar.club   nameserver = ns-870.awsdns-44.net
```

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.



DNS Management

servicar.club

Enter My Own Nameservers

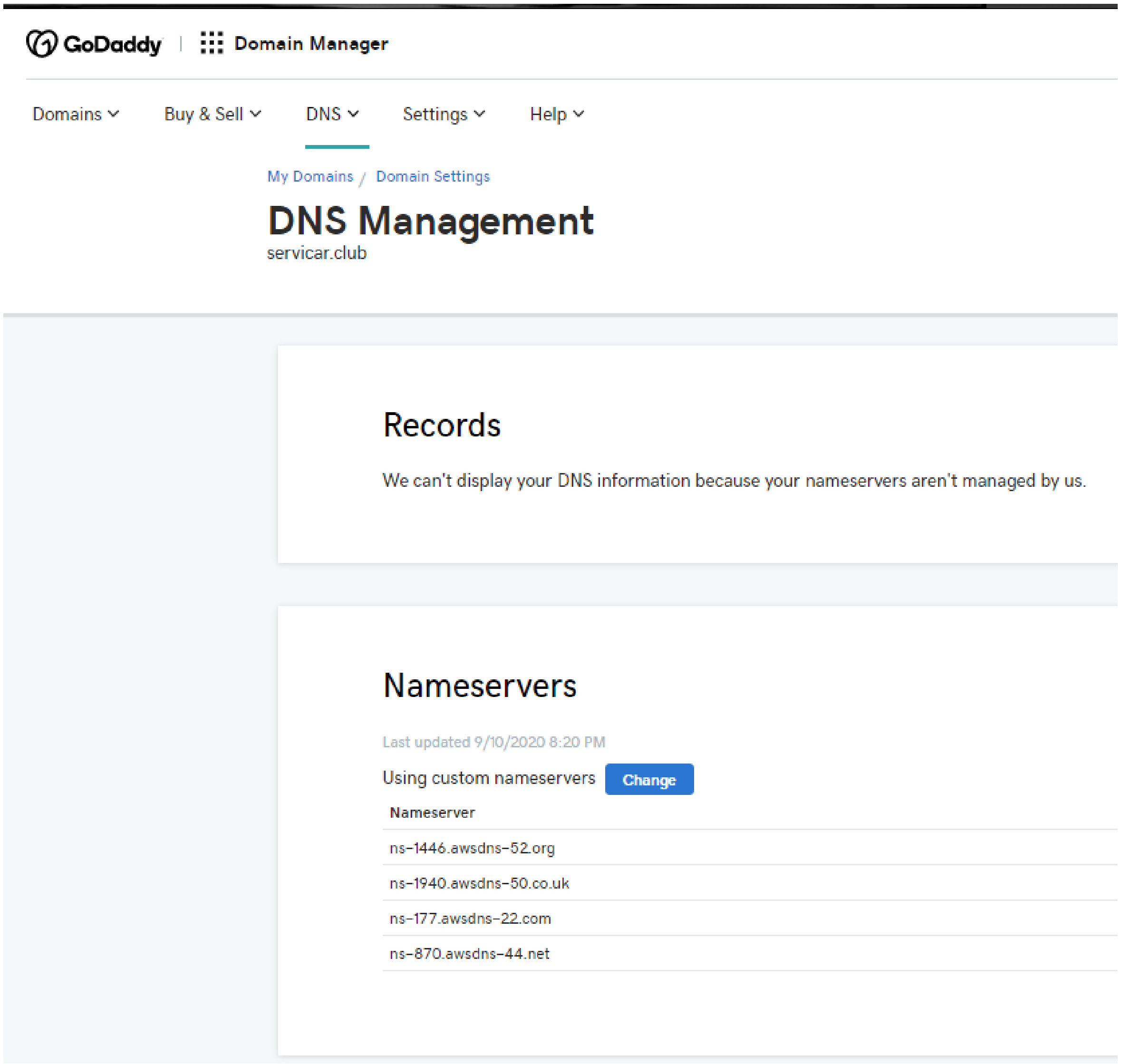
Changing nameservers is risky, and change could potentially lead to your website disappearing from public view.

ns-1446.awsdns-52.org

ns-1940.awsdns-50.co.uk

ns-177.awsdns-22.com

ns-870.awsdns-44.net



DNS Management

servicar.club

Records

We can't display your DNS information because your nameservers aren't managed by us.

Nameservers

Last updated 9/10/2020 8:20 PM

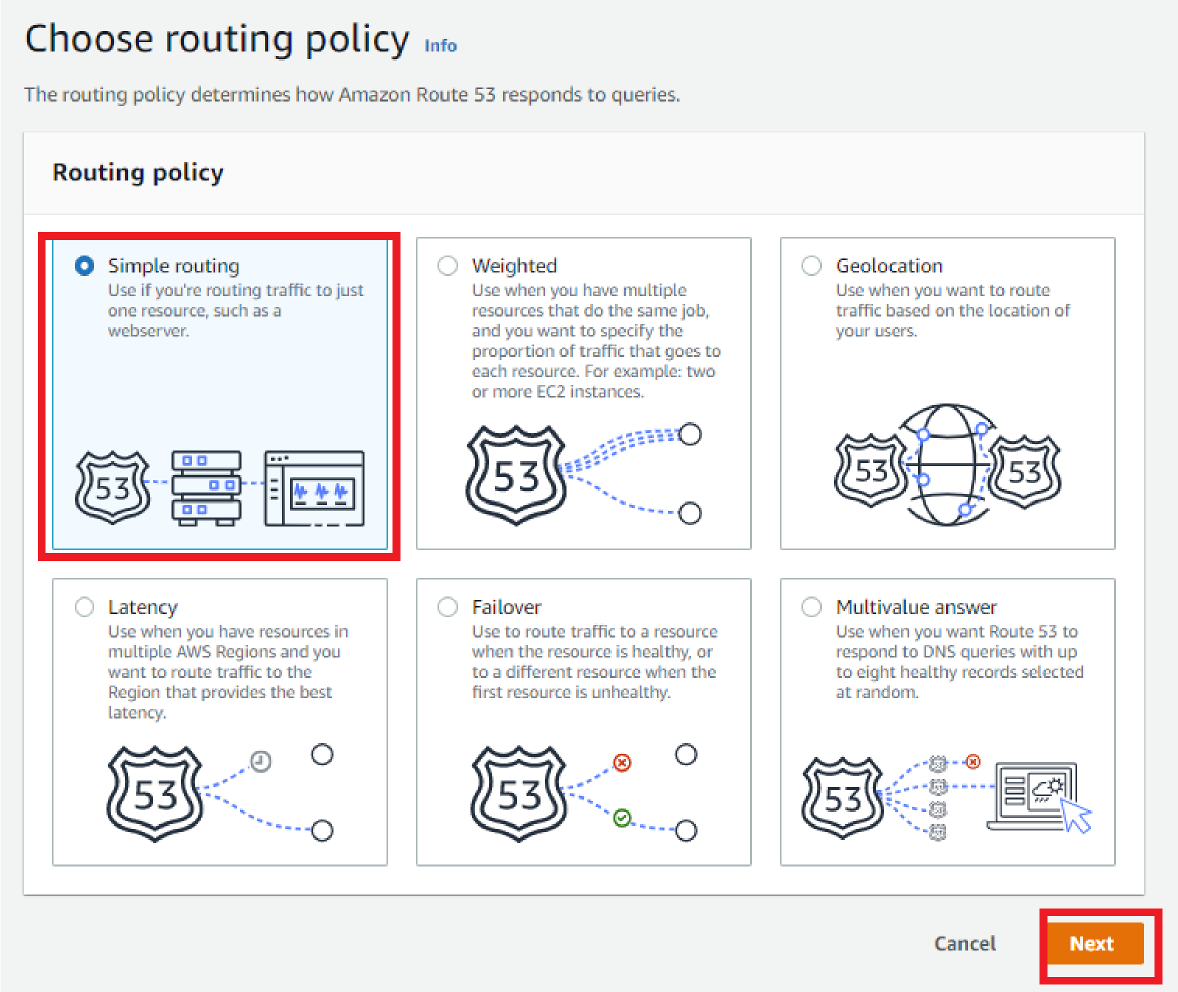
Using custom nameservers

Nameserver
ns-1446.awsdns-52.org
ns-1940.awsdns-50.co.uk
ns-177.awsdns-22.com
ns-870.awsdns-44.net

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.



Choose routing policy Info

The routing policy determines how Amazon Route 53 responds to queries.

Routing policy

● Simple routing
Use if you're routing traffic to just one resource, such as a webserver.

○ Weighted
Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

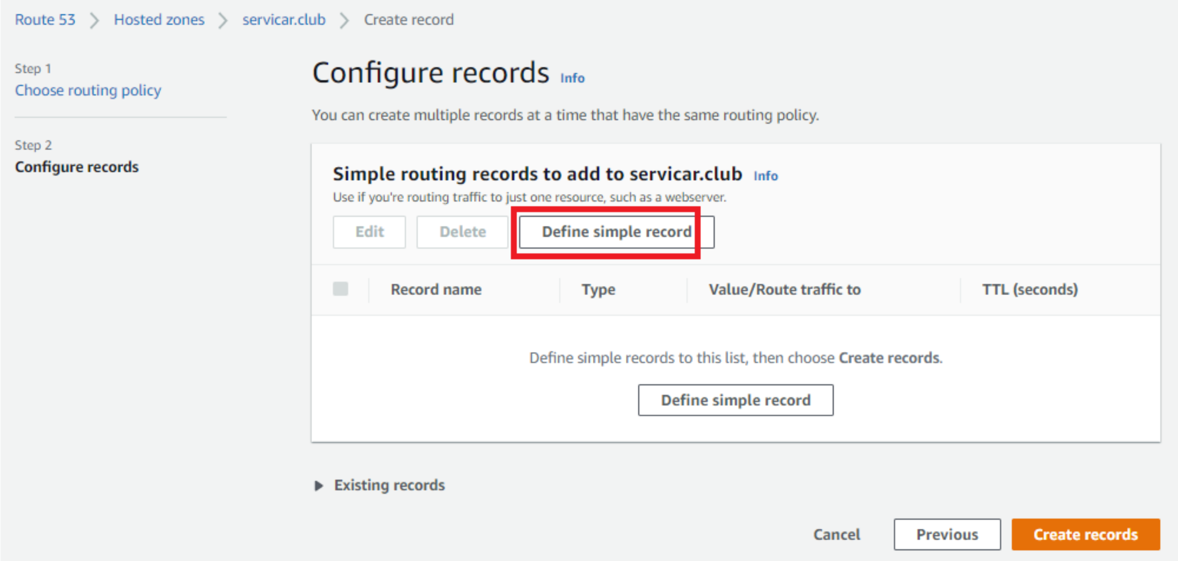
○ Geolocation
Use when you want to route traffic based on the location of your users.

○ Latency
Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

○ Failover
Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

○ Multivalue answer
Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

Click on Define Simple Records



Step 1
Choose routing policy

Step 2
Configure records

Configure records Info

You can create multiple records at a time that have the same routing policy.

Simple routing records to add to servicar.club Info

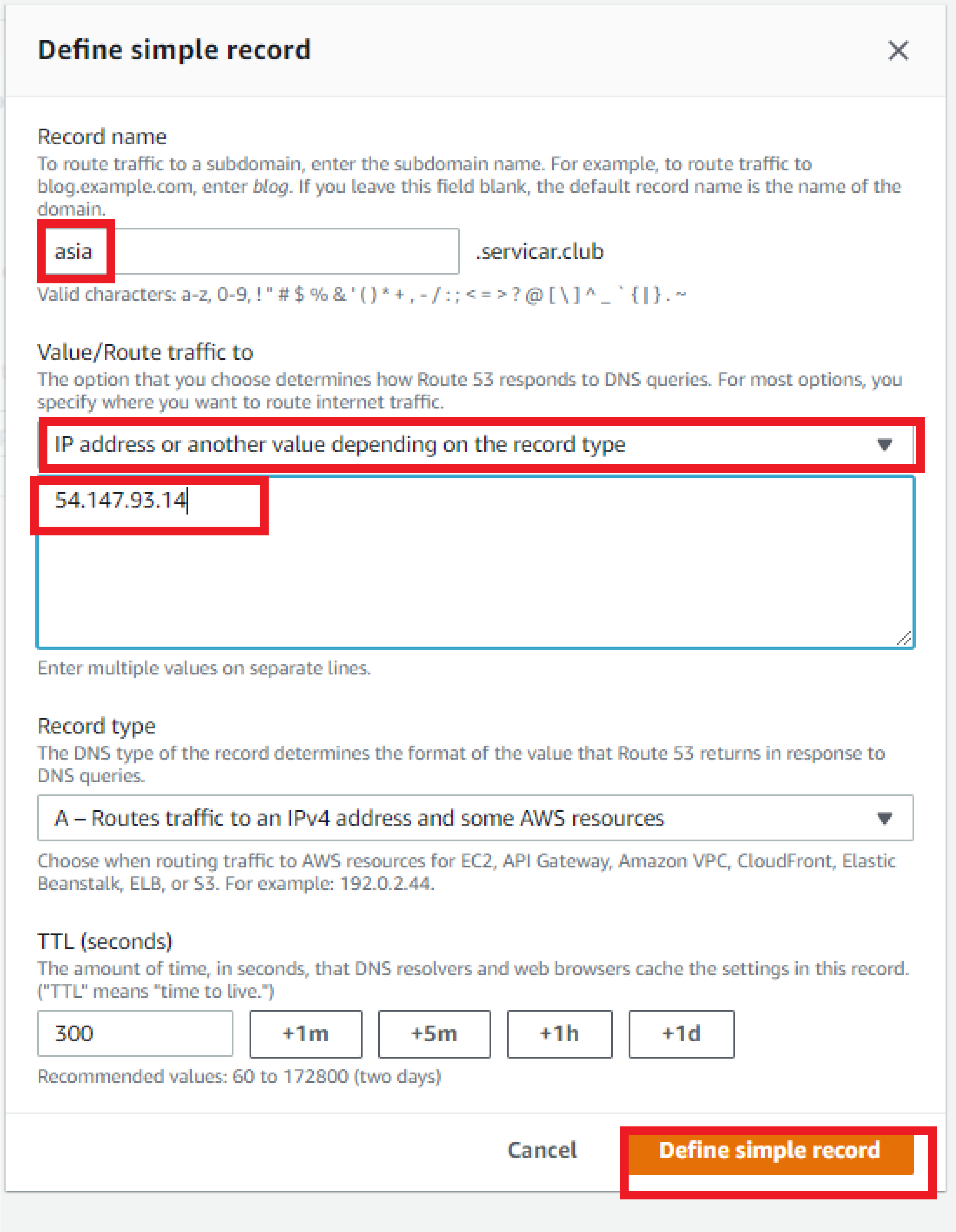
Use if you're routing traffic to just one resource, such as a webserver.

□	Record name	Type	Value/Route traffic to	TTL (seconds)

Define simple records to this list, then choose **Create records**.

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

Click on Define Simple Record.



Define simple record

Record name

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

asia .servicar.club

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Value/Route traffic to

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value depending on the record type ▼

54.147.93.14

Enter multiple values on separate lines.

Record type

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources ▼

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

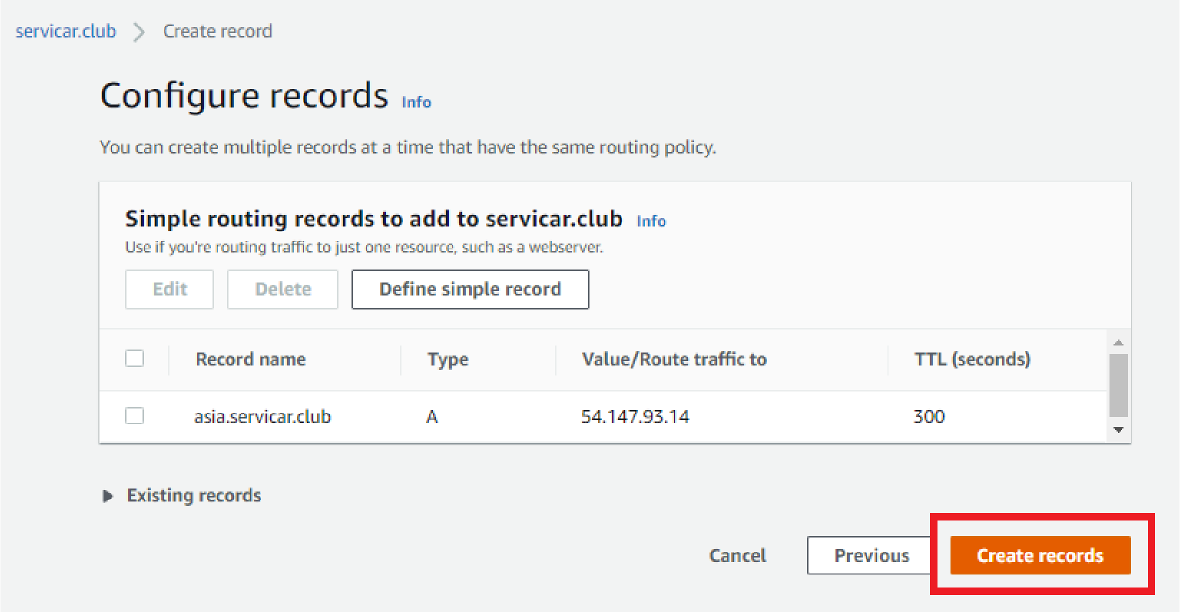
TTL (seconds)

The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. ("TTL" means "time to live.")

300

Recommended values: 60 to 172800 (two days)

On main screen, click on Create Records



servicar.club > Create record

Configure records Info

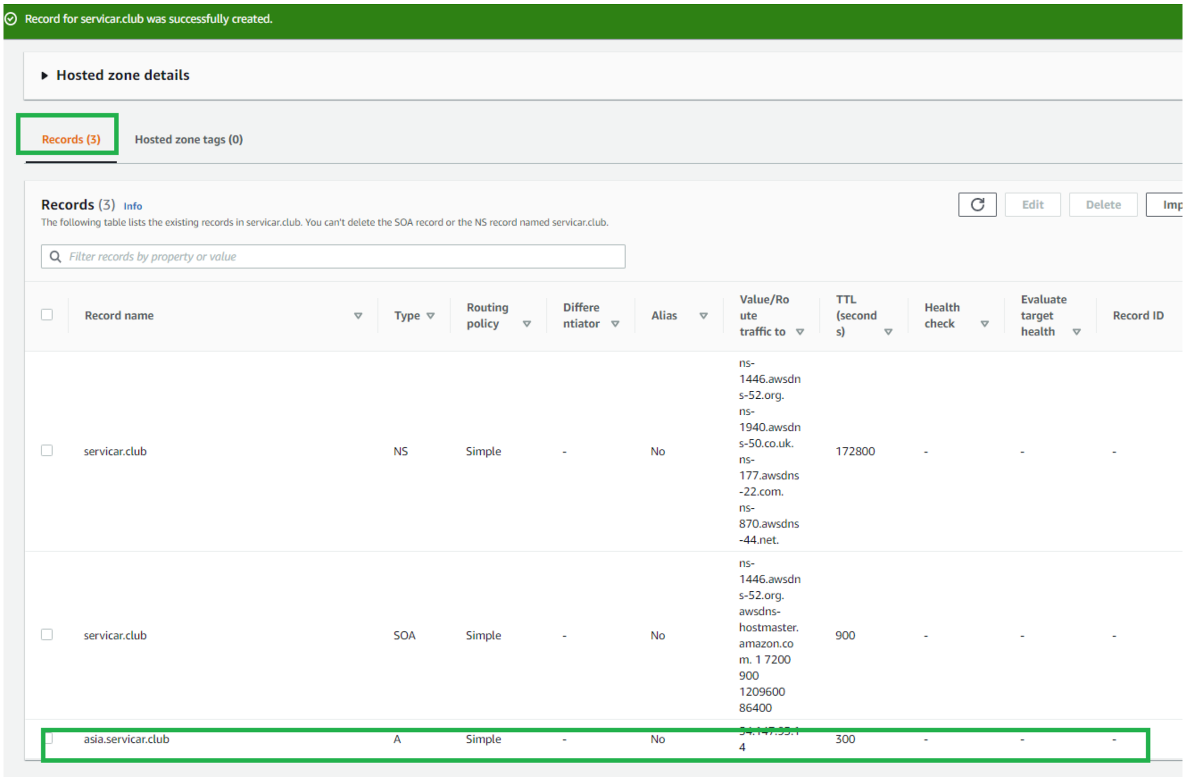
You can create multiple records at a time that have the same routing policy.

Simple routing records to add to servicar.club Info

Use if you're routing traffic to just one resource, such as a webserver.

□	Record name	Type	Value/Route traffic to	TTL (seconds)
□	asia.servicar.club	A	54.147.93.14	300

A New Record appear on Main Screen of Hosted Zone,



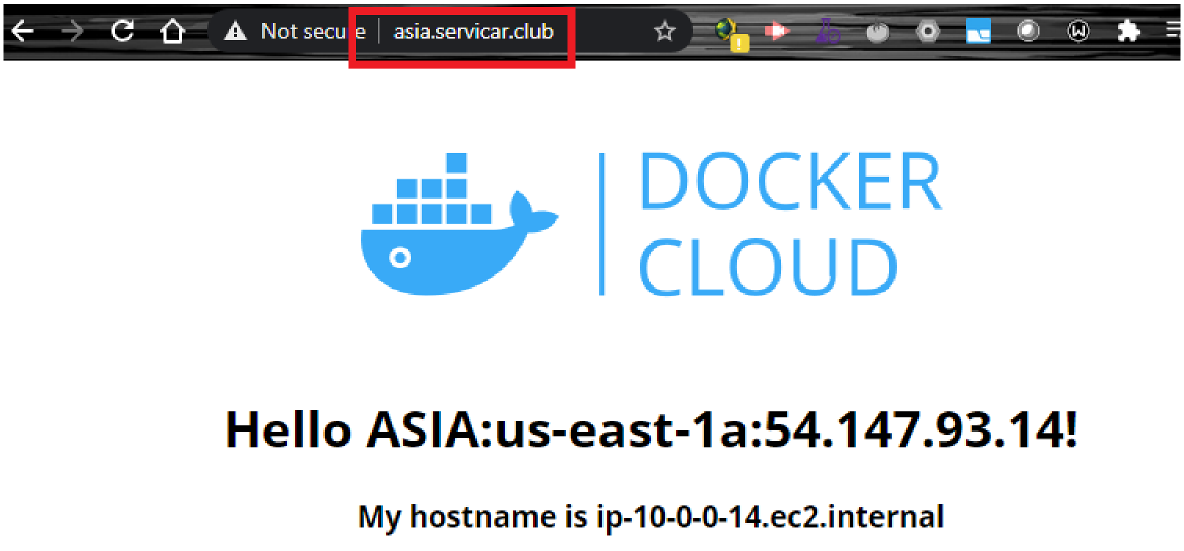
Record for servicar.club was successfully created.

Records (3) Info

The following table lists the existing records in servicar.club. You can't delete the SOA record or the NS record named servicar.club.

□	Record name	Type	Routing policy	Differentiator	Alias	Value/Route traffic to	TTL (seconds)	Health check	Evaluate target health	Record ID
□	servicar.club	NS	Simple	-	No	ns-1446.awsdns-52.org. ns-1940.awsdns-50.co.uk. ns-177.awsdns-22.com. ns-870.awsdns-44.net.	172800	-	-	-
□	servicar.club	SOA	Simple	-	No	ns-1446.awsdns-52.org. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400	900	-	-	-
□	asia.servicar.club	A	Simple	-	No	54.147.93.14	300	-	-	-

You can tested on your favorite browser,

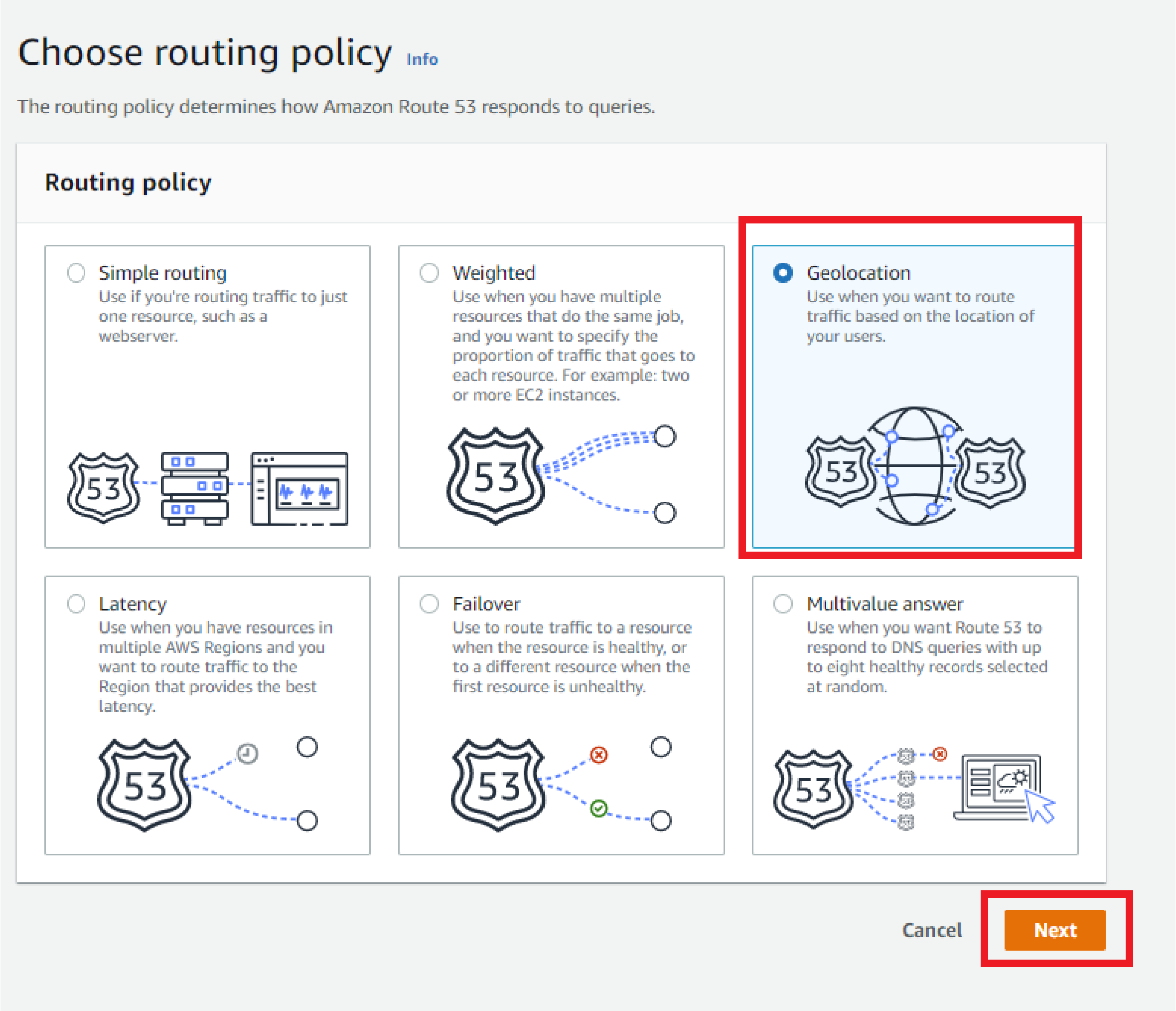


Hello ASIA:us-east-1a:54.147.93.14!

My hostname is ip-10-0-0-14.ec2.internal

Create a Geolocation Reouting Policy

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



Choose routing policy Info

The routing policy determines how Amazon Route 53 responds to queries.

Routing policy

○ Simple routing
Use if you're routing traffic to just one resource, such as a webserver.

○ Weighted
Use when you have multiple resources that do the same job, and you want to specify the proportion of traffic that goes to each resource. For example: two or more EC2 instances.

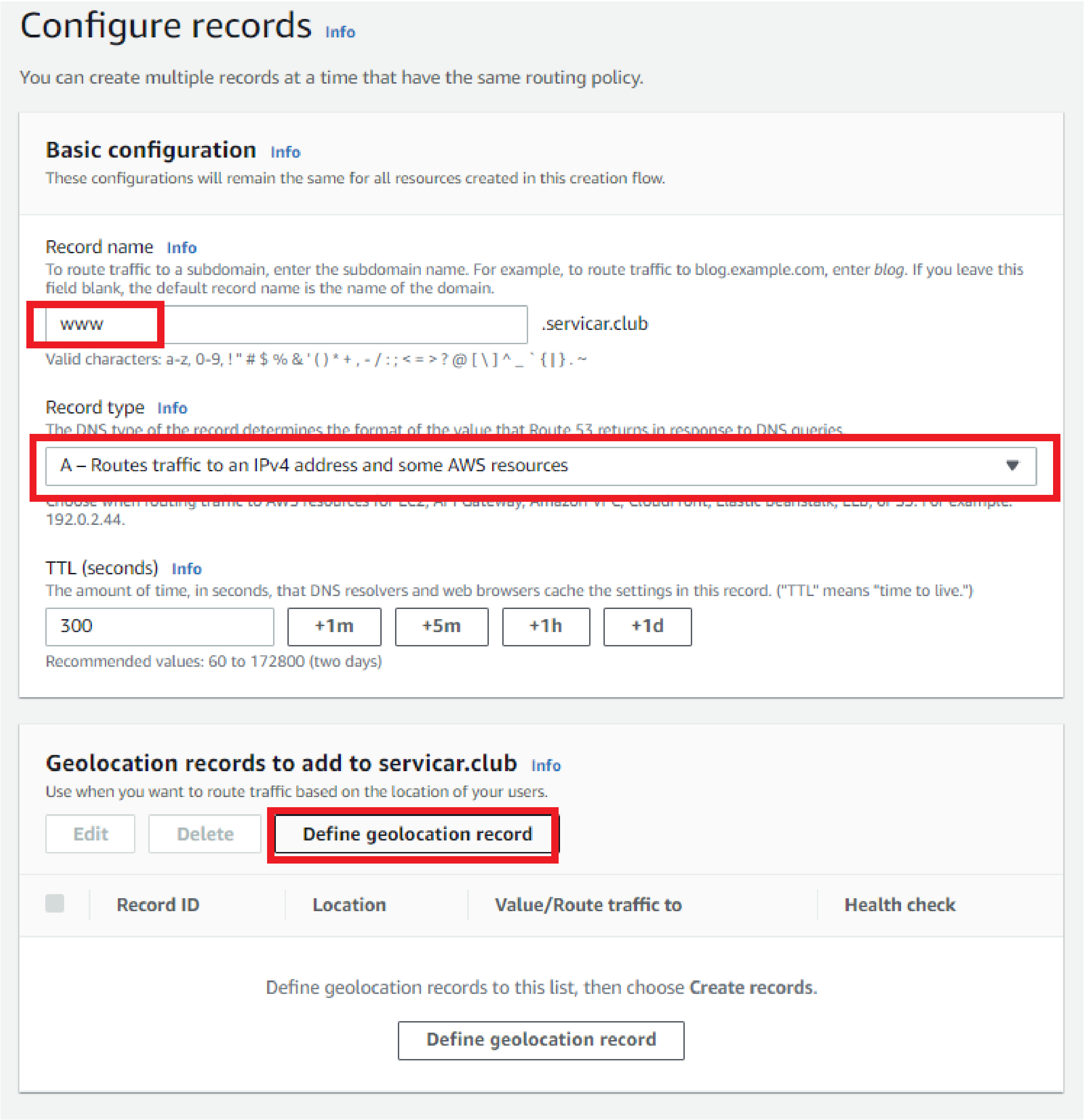
● Geolocation
Use when you want to route traffic based on the location of your users.

○ Latency
Use when you have resources in multiple AWS Regions and you want to route traffic to the Region that provides the best latency.

○ Failover
Use to route traffic to a resource when the resource is healthy, or to a different resource when the first resource is unhealthy.

○ Multivalue answer
Use when you want Route 53 to respond to DNS queries with up to eight healthy records selected at random.

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



Configure records Info

You can create multiple records at a time that have the same routing policy.

Basic configuration Info

These configurations will remain the same for all resources created in this creation flow.

Record name Info

To route traffic to a subdomain, enter the subdomain name. For example, to route traffic to blog.example.com, enter *blog*. If you leave this field blank, the default record name is the name of the domain.

www .servicar.club

Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [\ ] ^ _ ` { | } . ~

Record type Info

The DNS type of the record determines the format of the value that Route 53 returns in response to DNS queries.

A – Routes traffic to an IPv4 address and some AWS resources

Choose when routing traffic to AWS resources for EC2, API Gateway, Amazon VPC, CloudFront, Elastic Beanstalk, ELB, or S3. For example: 192.0.2.44.

TTL (seconds) Info

The amount of time, in seconds, that DNS resolvers and web browsers cache the settings in this record. ("TTL" means "time to live.")

300

Recommended values: 60 to 172800 (two days)

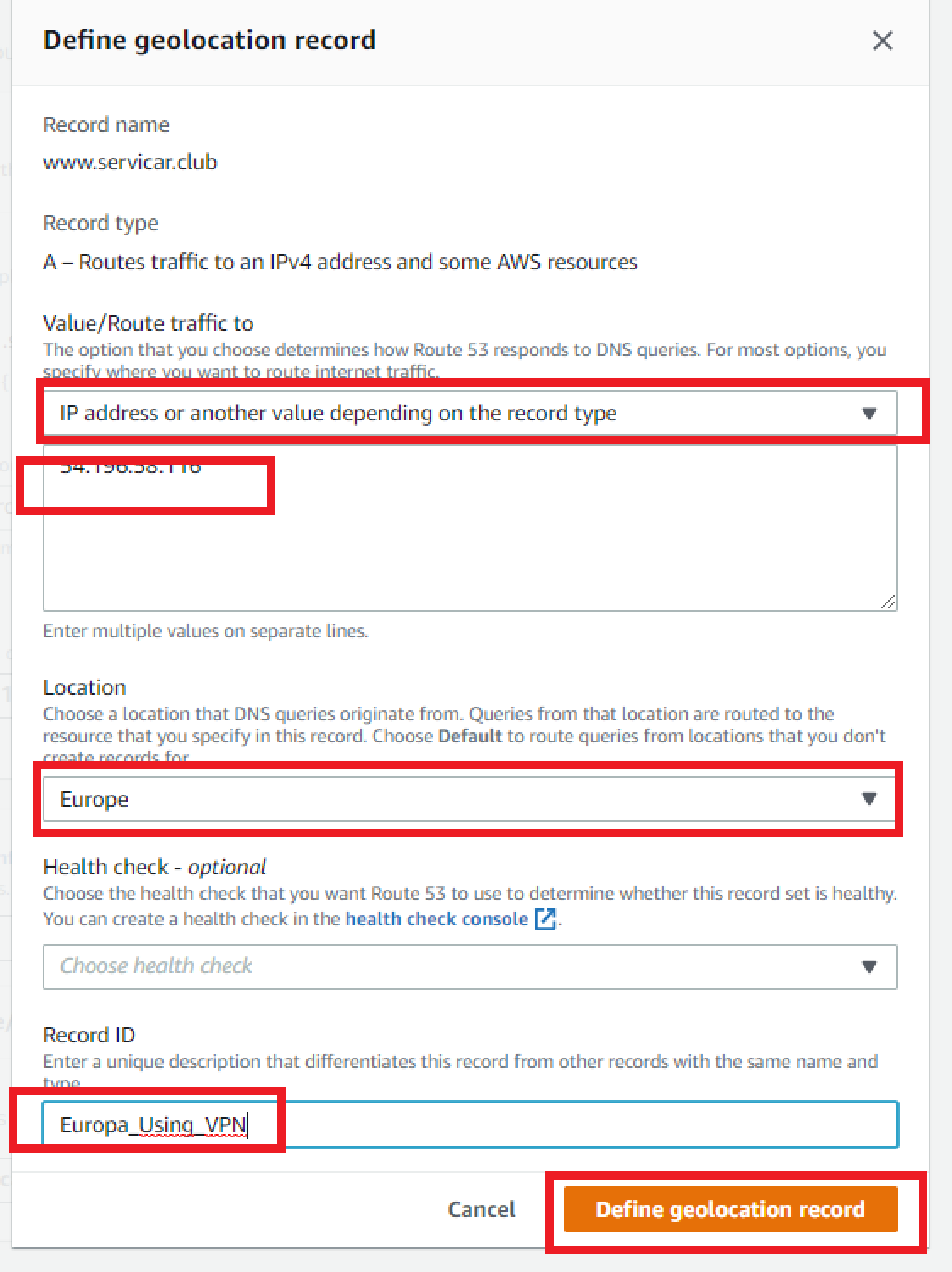
Geolocation records to add to servicar.club Info

Use when you want to route traffic based on the location of your users.

	Record ID	Location	Value/Route traffic to	Health check
☐				

Define geolocation records to this list, then choose **Create records.**

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



Define geolocation record

Record name

www.servicar.club

Record type

A – Routes traffic to an IPv4 address and some AWS resources

Value/Route traffic to

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value depending on the record type ▼

54.196.58.116

Enter multiple values on separate lines.

Location

Choose a location that DNS queries originate from. Queries from that location are routed to the resource that you specify in this record. Choose **Default** to route queries from locations that you don't create records for.

Europe ▼

Health check - *optional*

Choose the health check that you want Route 53 to use to determine whether this record set is healthy. You can create a health check in the **health check console**.

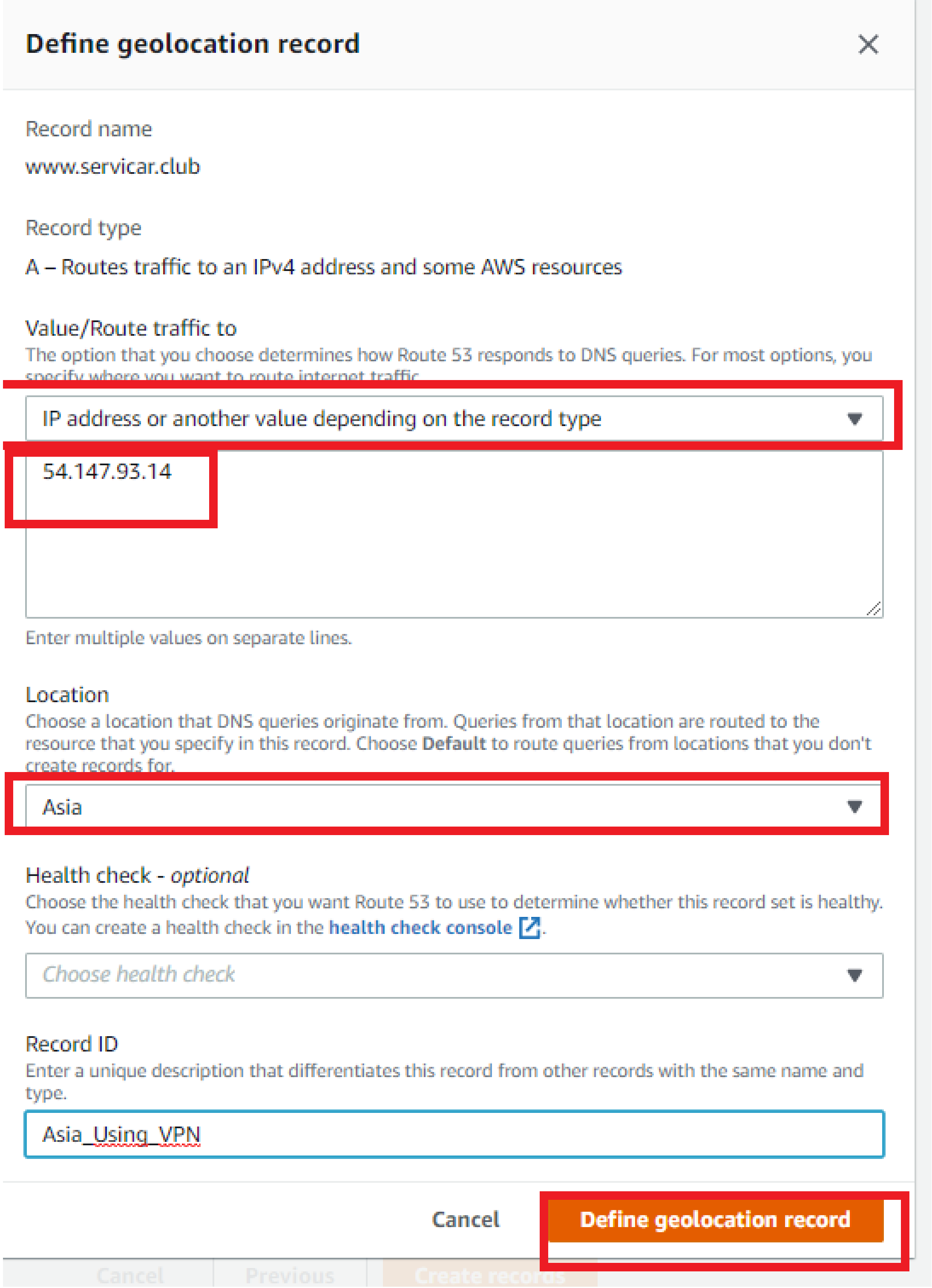
Choose health check ▼

Record ID

Enter a unique description that differentiates this record from other records with the same name and type.

Europa_Using_VPN

For Asia, the same thing,



Define geolocation record

Record name

www.servicar.club

Record type

A – Routes traffic to an IPv4 address and some AWS resources

Value/Route traffic to

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value depending on the record type

54.147.93.14

Enter multiple values on separate lines.

Location

Choose a location that DNS queries originate from. Queries from that location are routed to the resource that you specify in this record. Choose **Default** to route queries from locations that you don't create records for.

Asia

Health check - *optional*

Choose the health check that you want Route 53 to use to determine whether this record set is healthy. You can create a health check in the **health check console**.

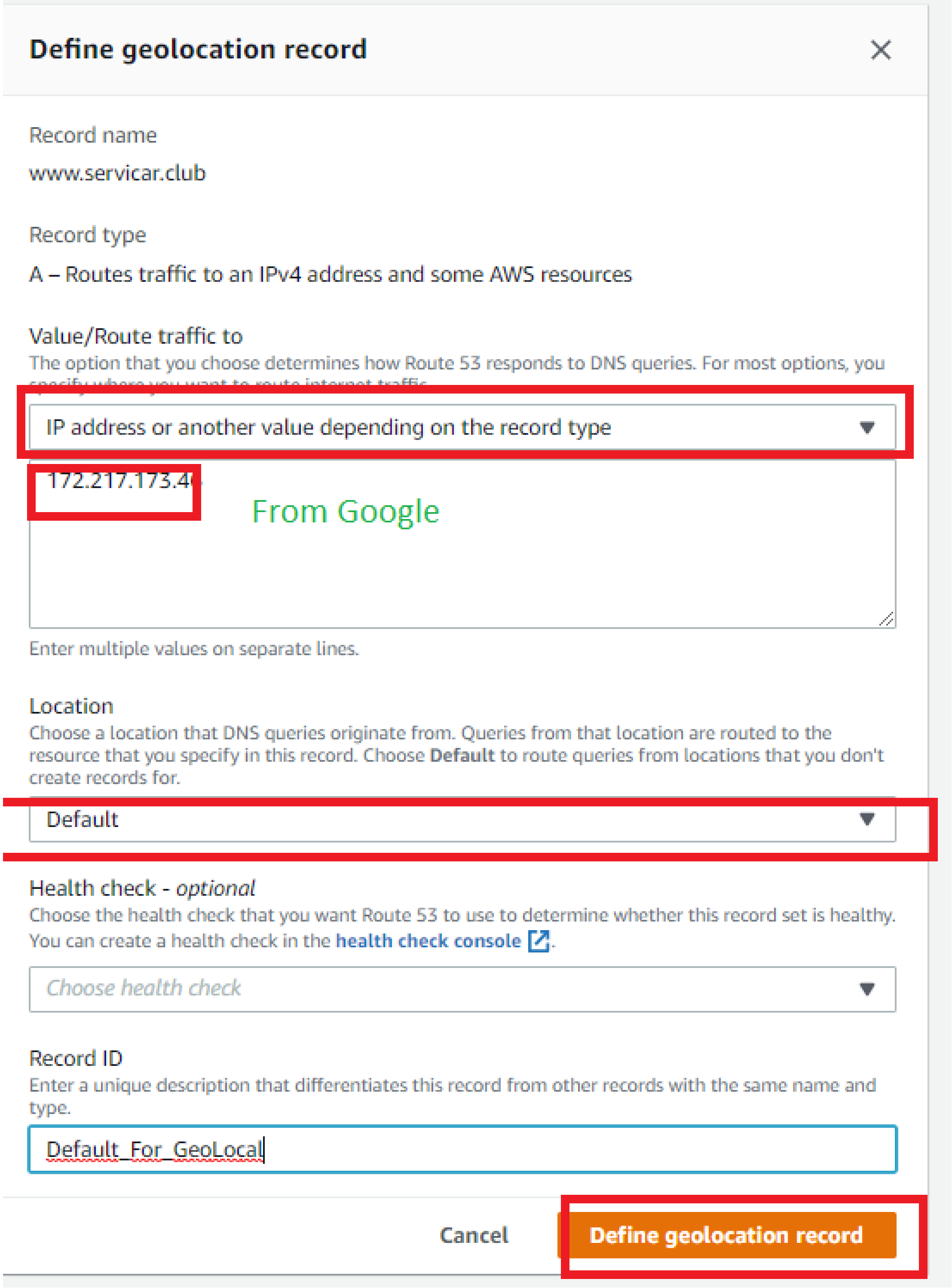
Choose health check

Record ID

Enter a unique description that differentiates this record from other records with the same name and type.

Asia_Using_VPN

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



Define geolocation record

Record name

www.servicar.club

Record type

A – Routes traffic to an IPv4 address and some AWS resources

Value/Route traffic to

The option that you choose determines how Route 53 responds to DNS queries. For most options, you specify where you want to route internet traffic.

IP address or another value depending on the record type ▼

172.217.173.4

From Google

Enter multiple values on separate lines.

Location

Choose a location that DNS queries originate from. Queries from that location are routed to the resource that you specify in this record. Choose **Default** to route queries from locations that you don't create records for.

Default ▼

Health check - *optional*

Choose the health check that you want Route 53 to use to determine whether this record set is healthy. You can create a health check in the **health check console**.

Choose health check ▼

Record ID

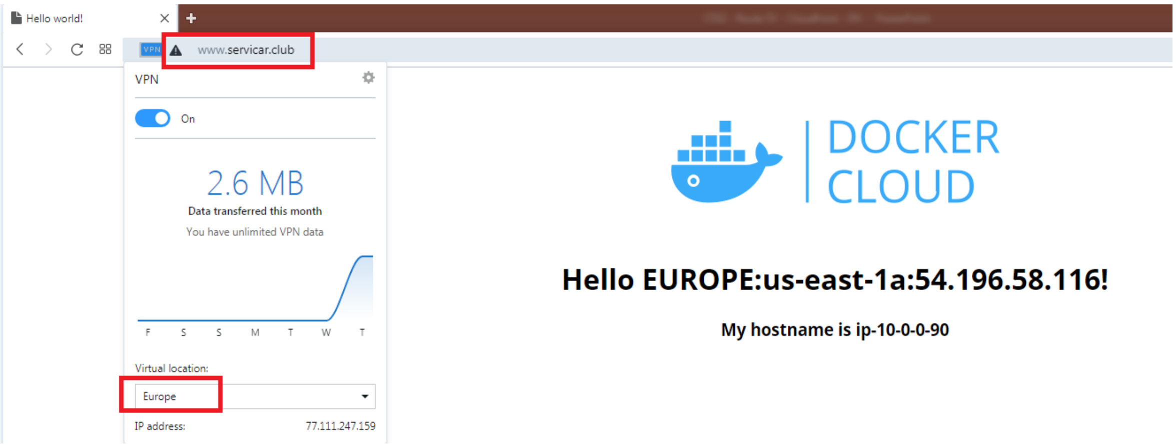
Enter a unique description that differentiates this record from other records with the same name and type.

Default_For_GeoLocal

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


Hello world!

www.servicar.club

VPN

On

2.6 MB

Data transferred this month

You have unlimited VPN data

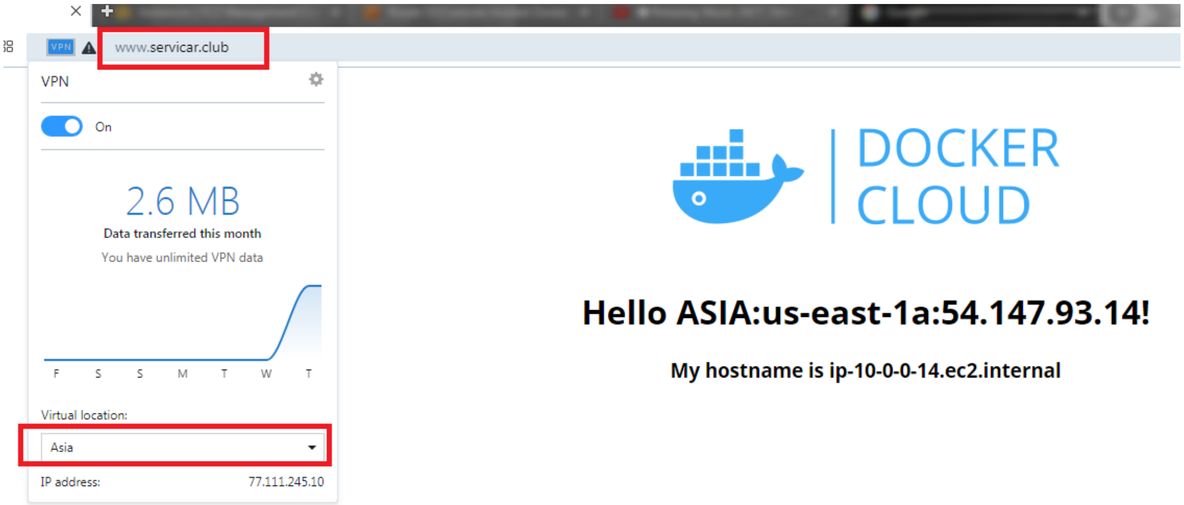
Virtual location: Europe

IP address: 77.111.247.159

DOCKER CLOUD

Hello EUROPE:us-east-1a:54.196.58.116!

My hostname is ip-10-0-0-90


www.servicar.club

VPN

On

2.6 MB

Data transferred this month

You have unlimited VPN data

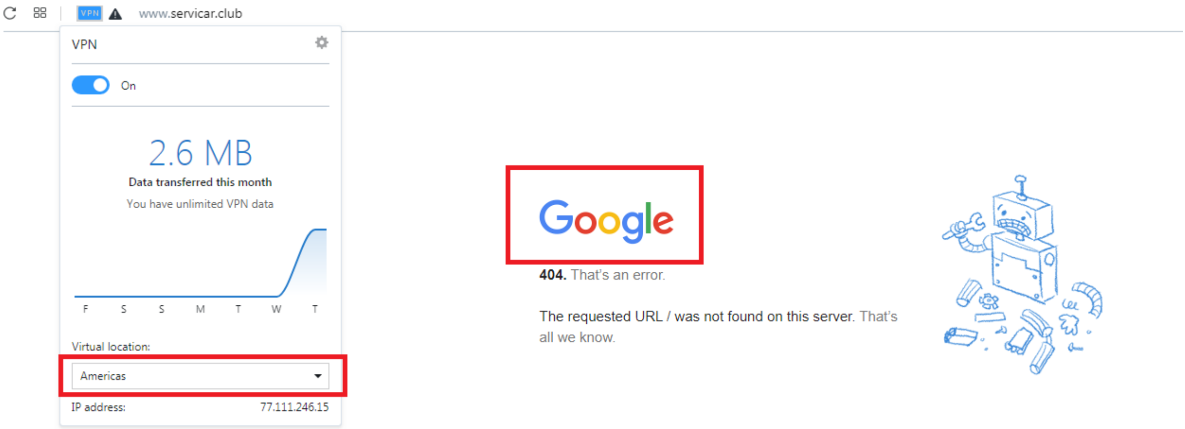
Virtual location: Asia

IP address: 77.111.245.10

DOCKER CLOUD

Hello ASIA:us-east-1a:54.147.93.14!

My hostname is ip-10-0-0-14.ec2.internal


www.servicar.club

VPN

On

2.6 MB

Data transferred this month

You have unlimited VPN data

Virtual location: Americas

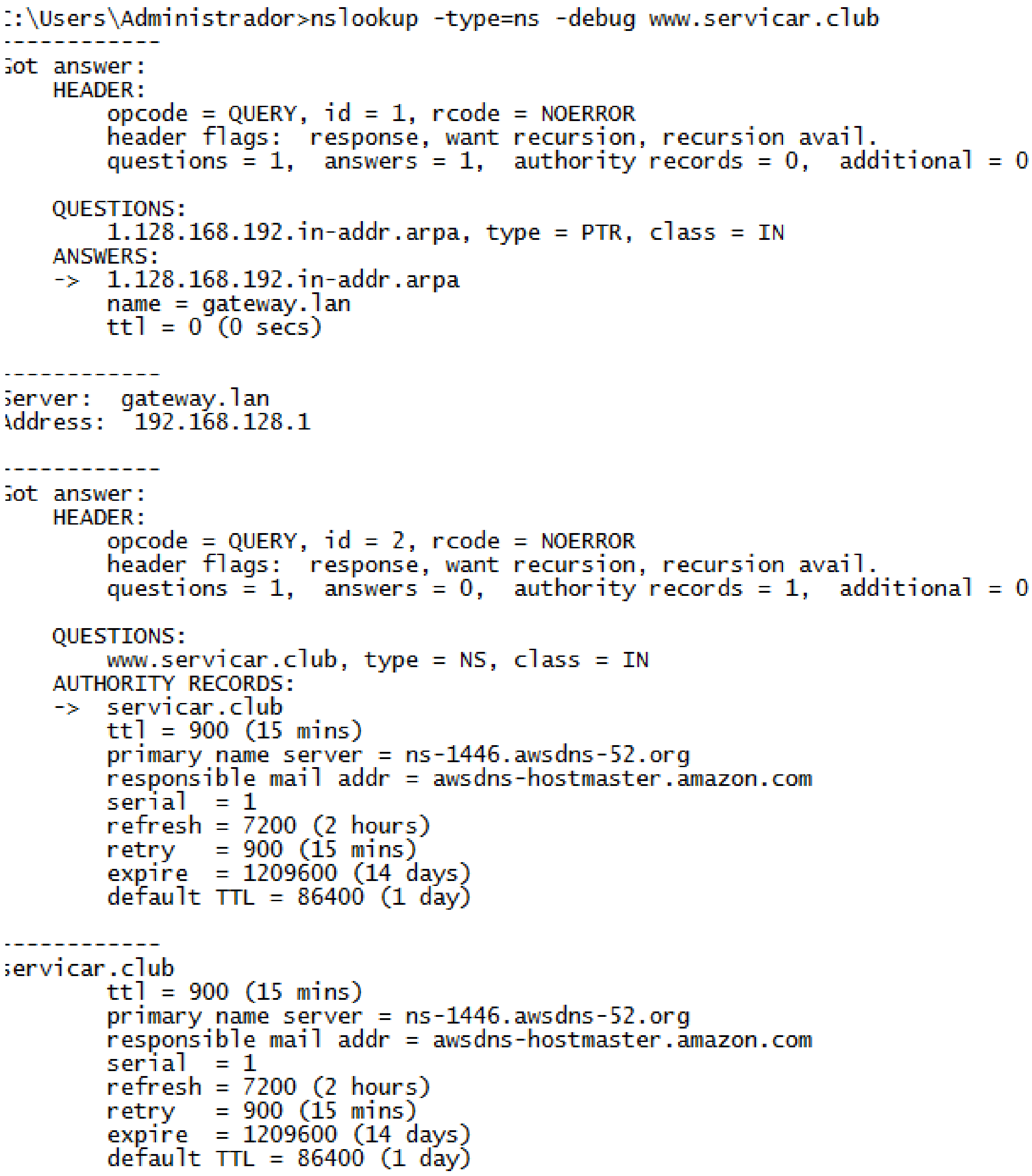
IP address: 77.111.246.15

Google

404. That's an error.

The requested URL / was not found on this server. That's all we know.

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



```
C:\Users\Administrador>nslookup -type=ns -debug www.servicar.club
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 1, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 1,  authority records = 0,  additional = 0

    QUESTIONS:
        1.128.168.192.in-addr.arpa, type = PTR, class = IN
    ANSWERS:
    ->  1.128.168.192.in-addr.arpa
        name = gateway.lan
        ttl = 0 (0 secs)

------------
Server:  gateway.lan
Address:  192.168.128.1

------------
Got answer:
    HEADER:
        opcode = QUERY, id = 2, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 0,  authority records = 1,  additional = 0

    QUESTIONS:
        www.servicar.club, type = NS, class = IN
    AUTHORITY RECORDS:
    ->  servicar.club
        ttl = 900 (15 mins)
        primary name server = ns-1446.awsdns-52.org
        responsible mail addr = awsdns-hostmaster.amazon.com
        serial  = 1
        refresh = 7200 (2 hours)
        retry   = 900 (15 mins)
        expire  = 1209600 (14 days)
        default TTL = 86400 (1 day)

------------
servicar.club
        ttl = 900 (15 mins)
        primary name server = ns-1446.awsdns-52.org
        responsible mail addr = awsdns-hostmaster.amazon.com
        serial  = 1
        refresh = 7200 (2 hours)
        retry   = 900 (15 mins)
        expire  = 1209600 (14 days)
        default TTL = 86400 (1 day)
```

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
```