# AWS Portfolio

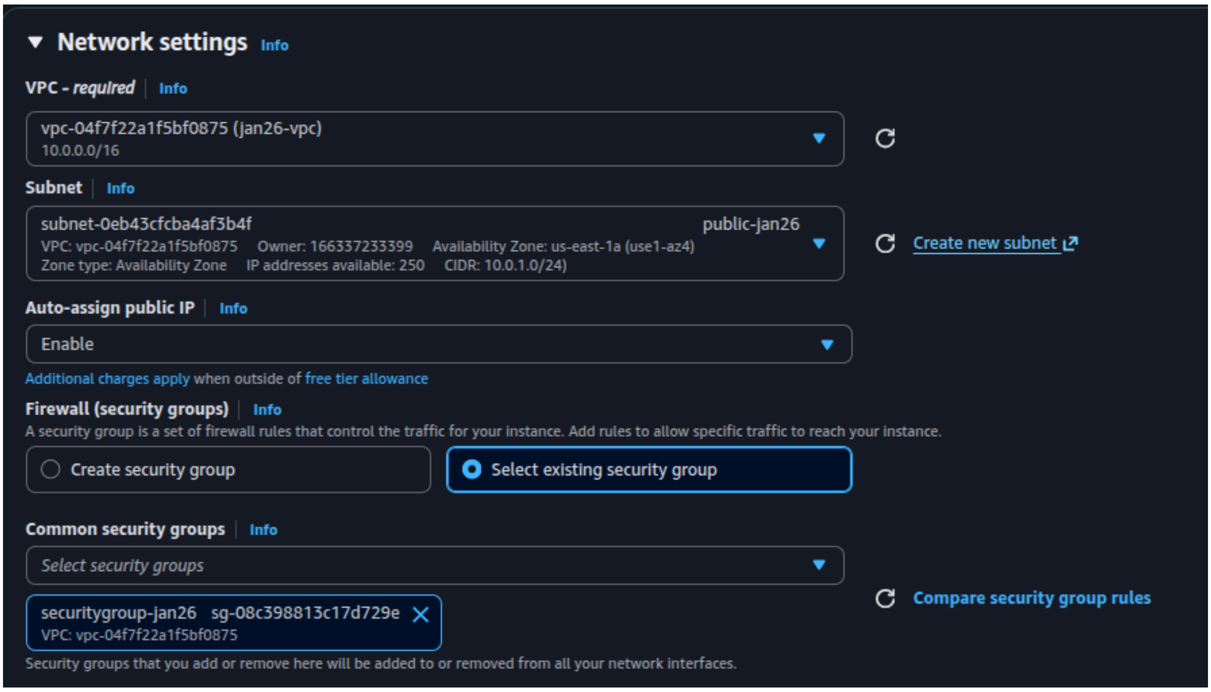
## Creating portfolio

First, you need to create a portfolio with data, you can provide data as a .pdf cv or fetch it from google, for example from GitHub, Linkedin, etc.

ChatGPT canvas is a nice tool for looking how it would look before deploying.

## EC2

Later, create an EC2 instance with the following settings:



**Network settings** Info

**VPC - *required*** | Info

vpc-04f7f22a1f5bf0875 (jan26-vpc)
10.0.0.0/16

**Subnet** | Info

subnet-0eb43cfcba4af3b4f public-jan26
VPC: vpc-04f7f22a1f5bf0875 Owner: 166337233399 Availability Zone: us-east-1a (use1-az4)
Zone type: Availability Zone IP addresses available: 250 CIDR: 10.0.1.0/24)

Create new subnet

**Auto-assign public IP** | Info

Enable

Additional charges apply when outside of free tier allowance

**Firewall (security groups)** | Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

**Common security groups** | Info

*Select security groups*

securitygroup-jan26 sg-08c398813c17d729e
VPC: vpc-04f7f22a1f5bf0875

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

## Install Git

Use:

```
sudo yum update
sudo yum install git
```

## Git pull

Use:

```
git clone <repo_url>
```

## run.sh

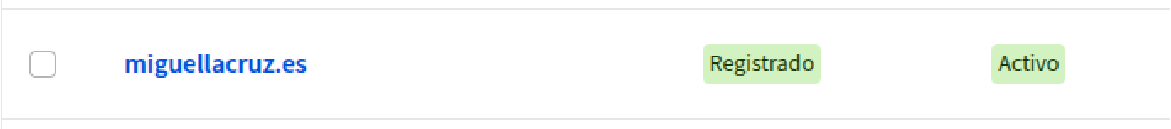
Update run.sh to work with your portfolio, as I used react will need to use a more robust run sh:

```bash
#!/usr/bin/env bash
set -euo pipefail

# ========= Config =========
APP_NAME="portfolio"
SERVER_NAME="_"          # set your domain, e.g. "miguellacruz.es"
PORT="80"
WEB_ROOT="/var/www/${APP_NAME}"
BUILD_DIR="/tmp/${APP_NAME}-vite"

# Resolve repository location from this script path
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"

# Input files/folders in repo
PORTFOLIO_FILE="${SCRIPT_DIR}/portfolio.tsx"
IMG_DIR="${SCRIPT_DIR}/img"
CV_DIR="${SCRIPT_DIR}/cv"

NGINX_SITE="/etc/nginx/conf.d/${APP_NAME}.conf"

need_cmd() { command -v "$1" >/dev/null 2>&1; }

install_pkgs() {
 if need_cmd dnf; then
   sudo dnf -y update
   sudo dnf -y install nginx curl rsync ca-certificates tar gzip
 elif need_cmd yum; then
   sudo yum -y update
   sudo yum -y install nginx curl rsync ca-certificates tar gzip
 else
   echo "ERROR: Neither dnf nor yum found. This script targets Amazon Linux."
   exit 1
 fi
}

ensure_node() {
 if need_cmd node; then
   local node_major
   node_major="$(node -v | sed -E 's/^v([0-9]+).*/\1/')"
   if [ "${node_major}" -ge 20 ]; then
     return
   fi
 fi

 echo "==> Installing Node.js 20.x..."
 curl -fsSL https://rpm.nodesource.com/setup_20.x | sudo bash -

 if need_cmd dnf; then
   sudo dnf -y install nodejs
 else
   sudo yum -y install nodejs
 fi
}

echo "==> Validating input..."
if [ ! -f "$PORTFOLIO_FILE" ]; then
 echo "ERROR: ${PORTFOLIO_FILE} not found. Ensure portfolio.tsx is next to run.sh."
 exit 1
fi

echo "==> Installing system dependencies for Amazon Linux..."
install_pkgs
```

```
ensure_node

echo "==> Node: $(node -v) | npm: $(npm -v)"

echo "==> Creating Vite React+TS project in ${BUILD_DIR}..."
rm -rf "$BUILD_DIR"
mkdir -p "$BUILD_DIR"
cd /tmp

npm create vite@latest "${APP_NAME}-vite" -- --template react-ts
cd "$BUILD_DIR"

echo "==> Installing dependencies..."
npm install

echo "==> Installing required libraries..."
npm install framer-motion lucide-react react-github-calendar

echo "==> Copying portfolio.tsx into src/Portfolio.tsx..."
cp -f "$PORTFOLIO_FILE" "$BUILD_DIR/src/Portfolio.tsx"

echo "==> Writing src/main.tsx to render Portfolio..."
cat > "$BUILD_DIR/src/main.tsx" <<'EOF_MAIN'
import React from "react";
import ReactDOM from "react-dom/client";
import Portfolio from "./Portfolio";
import "./index.css";

ReactDOM.createRoot(document.getElementById("root")!).render(
 <React.StrictMode>
   <Portfolio />
 </React.StrictMode>
);
EOF_MAIN

echo "==> Preparing static assets (public/img, public/cv)..."
mkdir -p "$BUILD_DIR/public/img" "$BUILD_DIR/public/cv"

if [ -d "$IMG_DIR" ]; then
 echo "==> Copying img -> public/img ..."
 rsync -av --delete "$IMG_DIR"/ "$BUILD_DIR/public/img"/
else
 echo "==> No img folder found (ok)."
fi

if [ -d "$CV_DIR" ]; then
 echo "==> Copying cv -> public/cv ..."
 rsync -av --delete "$CV_DIR"/ "$BUILD_DIR/public/cv"/
else
 echo "==> No cv folder found (ok)."
fi

echo "==> Building..."
npm run build

if [ ! -d "$BUILD_DIR/dist" ]; then
 echo "ERROR: dist/ not found after build."
 exit 1
fi

echo "==> Deploying dist/ to ${WEB_ROOT}..."
sudo mkdir -p "$WEB_ROOT"
sudo rsync -av --delete "$BUILD_DIR/dist"/ "$WEB_ROOT"/

echo "==> Writing Nginx config..."
sudo tee "$NGINX_SITE" >/dev/null <<EOF_NGINX
server {
 listen ${PORT};
 server_name ${SERVER_NAME};
 root ${WEB_ROOT};
 index index.html;

 location / {
   try_files \$uri \$uri/ /index.html;
 }

 location ~* \\.(?:css|js|mjs|map|jpg|jpeg|png|gif|svg|webp|ico|woff2|woff|ttf)$ {
   expires 30d;
   add_header Cache-Control "public, max-age=2592000, immutable";
   try_files \$uri =404;
 }

 add_header X-Content-Type-Options nosniff;
 add_header X-Frame-Options SAMEORIGIN;
 add_header Referrer-Policy strict-origin-when-cross-origin;
}
EOF_NGINX

echo "==> Testing Nginx config..."
sudo nginx -t

echo "==> Enabling and restarting Nginx..."
sudo systemctl enable nginx
sudo systemctl restart nginx

echo "✅ Done"
echo "   Served from: ${WEB_ROOT}"
echo "   Open: http://<EC2_PUBLIC_IP>/"
echo
echo "Notes:"
echo " - Open inbound port 80 in your EC2 Security Group"
echo " - SSL is intentionally manual for practice (configure later on your own)"
```

## nginx.conf

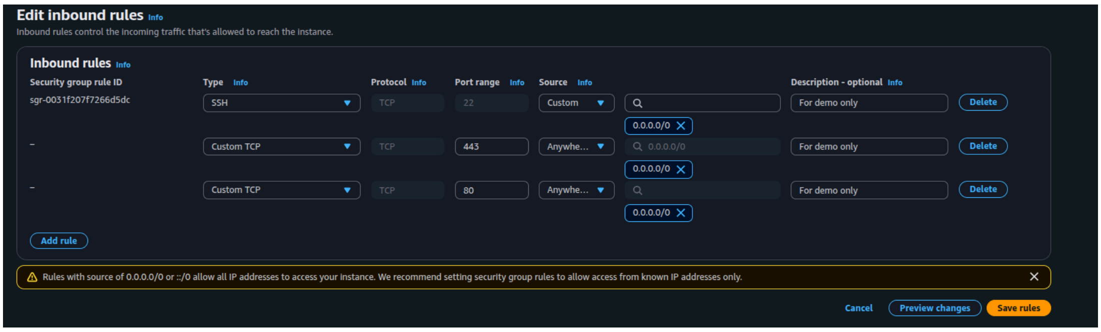
Is necessary to update the nginx.conf too in order to point to your own domain, in my case I use OVH cloud as provider, we don’t need proxy_pass http://localhost:8000 neither as we deploy in the server directly (ports 80, 443 if have SSL).



| | | | |
|---|---|---|---|
| □ | miguellacruz.es | Registrado | Activo |

## Inbound rules

In the security group add (for demo only), be sure opening the ports you need on inbound rules (443 https, 80 http, ect.):



**Edit inbound rules** Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

**Inbound rules** Info

| Security group rule ID | Type | Protocol | Port range | Source | | Description - optional | |
|---|---|---|---|---|---|---|---|
| sgr-0031f207f7266d5dc | SSH | TCP | 22 | Custom | 0.0.0.0/0 | For demo only | Delete |
| - | Custom TCP | TCP | 443 | Anywhe... | 0.0.0.0/0 | For demo only | Delete |
| - | Custom TCP | TCP | 80 | Anywhe... | 0.0.0.0/0 | For demo only | Delete |

⚠ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

## Permissions

```
chmod u+x run.sh
```

To be able to run run.sh

## Install nginx

```
sudo yum install nginx
```

## nginx.conf setup

Move your config to /etc/nginx/nginx.conf, can save a backup in case of problems:

- Save backup:

```
sudo cp /etc/nginx/nginx.conf /etc/nginx/nginx-backup.conf
```

- New conf:

```
sudo cp /home/ec2-user/<portfolio_aws>/nginx.conf /etc/nginx/nginx.conf
```

- Test if everything is right:

```
sudo nginx -t
systemctl status nginx
```

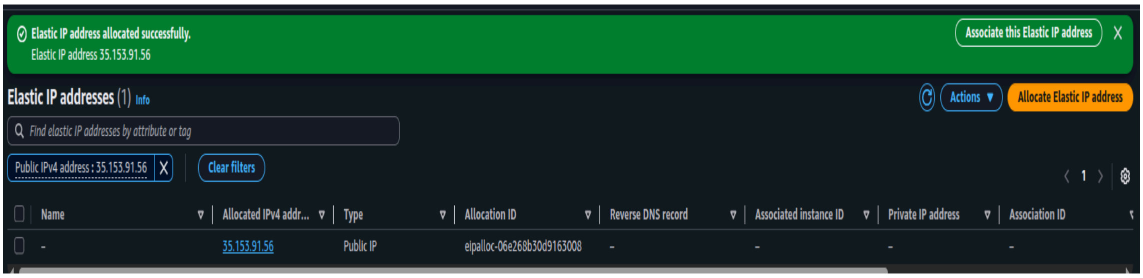
- If inactive:

```
sudo systemctl enable --now nginx
sudo systemctl reload nginx
```

Remember asking any AI to review your nginx file as this is out of the scope of this practice.

## Create Elastic IP

- Allocate the elastic IP



Elastic IP address allocated successfully.
Elastic IP address 35.153.91.56

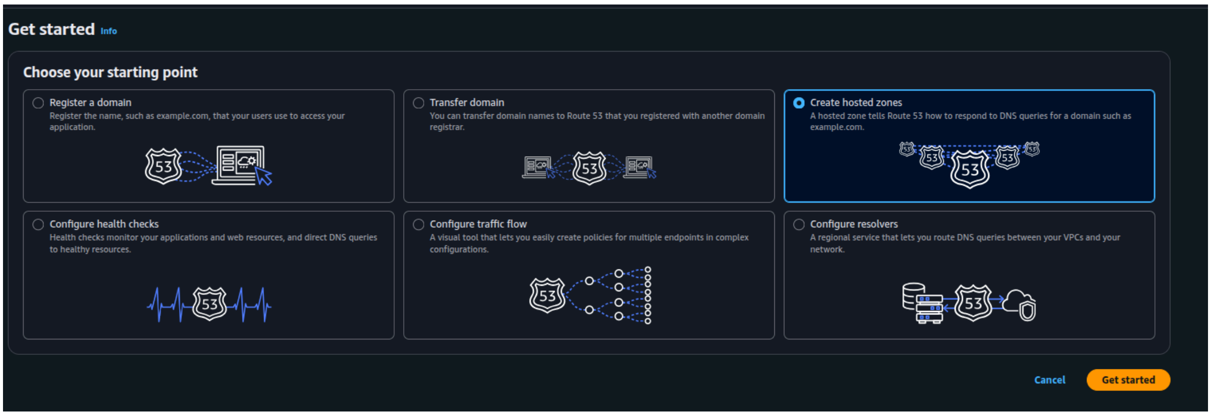
Elastic IP addresses (1)

Public IPv4 address : 35.153.91.56

| □ | Name | Allocated IPv4 addr... | Type | Allocation ID | Reverse DNS record | Associated instance ID | Private IP address | Association ID |
|---|---|---|---|---|---|---|---|---|
| □ | - | 35.153.91.56 | Public IP | eipalloc-06e268b30d9163008 | - | - | - | - |

Associate this Elastic IP address
X
Info
Actions
Allocate Elastic IP address
Find elastic IP addresses by attribute or tag
Clear filters
1

- Associate to EC2 instance



Get started Info

Choose your starting point

○ Register a domain
Register the name, such as example.com, that your users use to access your application.

○ Transfer domain
You can transfer domain names to Route 53 that you registered with another domain registrar.

● Create hosted zones
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

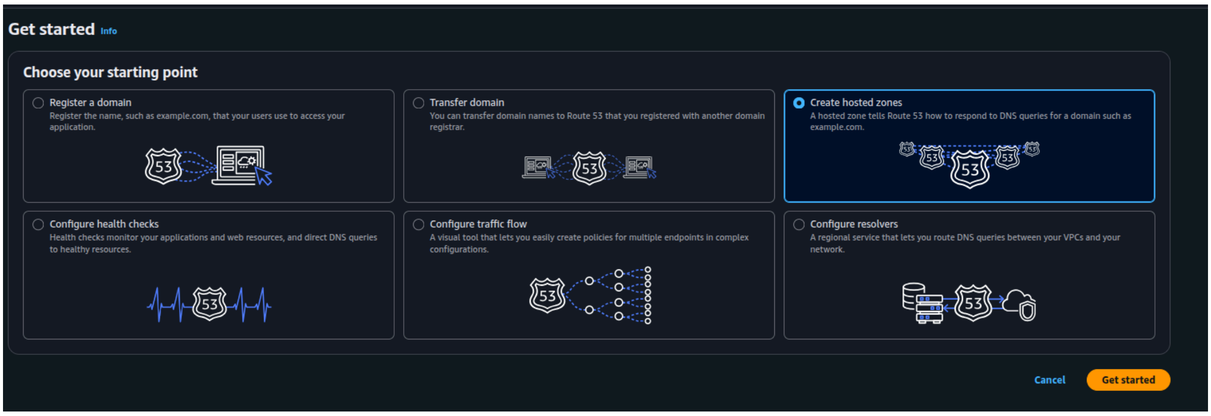
○ Configure health checks
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

○ Configure traffic flow
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

○ Configure resolvers
A regional service that lets you route DNS queries between your VPCs and your network.

## Point domain name to instance

- Create hosted zones



Get started Info

Choose your starting point

○ Register a domain
Register the name, such as example.com, that your users use to access your application.

○ Transfer domain
You can transfer domain names to Route 53 that you registered with another domain registrar.

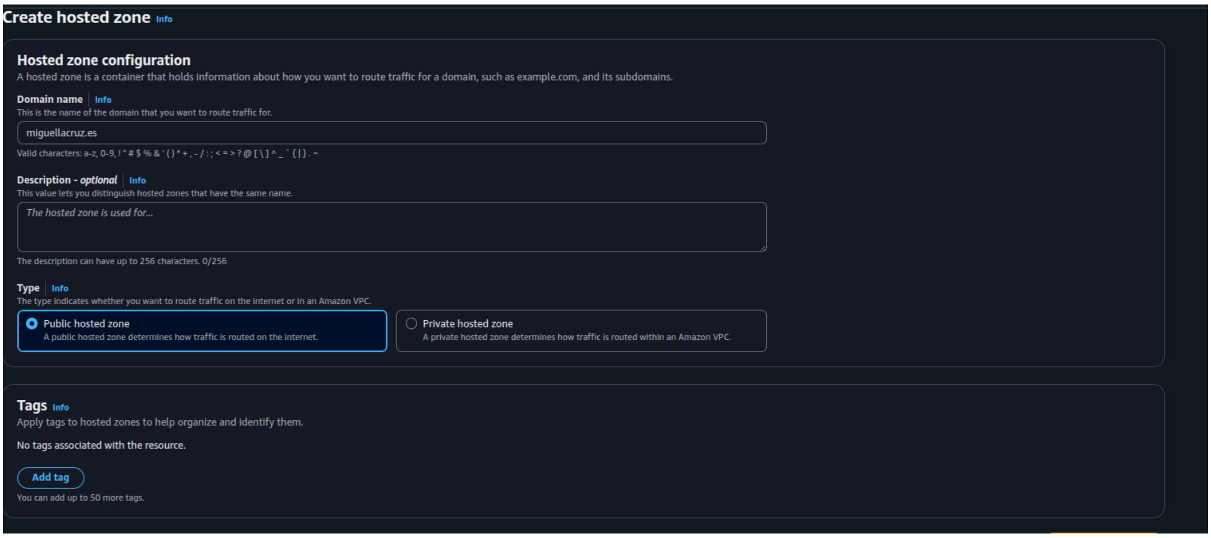
● Create hosted zones
A hosted zone tells Route 53 how to respond to DNS queries for a domain such as example.com.

○ Configure health checks
Health checks monitor your applications and web resources, and direct DNS queries to healthy resources.

○ Configure traffic flow
A visual tool that lets you easily create policies for multiple endpoints in complex configurations.

○ Configure resolvers
A regional service that lets you route DNS queries between your VPCs and your network.

Cancel Get started



Create hosted zone Info

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 | Info
This is the name of the domain that you want to route traffic for.
miguellacruz.es
Valid characters: a-z, 0-9, ! " # $ % & ' ( ) * + , - / : ; < = > ? @ [ \ ] ^ _ ` { | } . ~

Description - *optional* | Info
This value lets you distinguish hosted zones that have the same name.
*The hosted zone is used for...*
The description can have up to 256 characters. 0/256

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

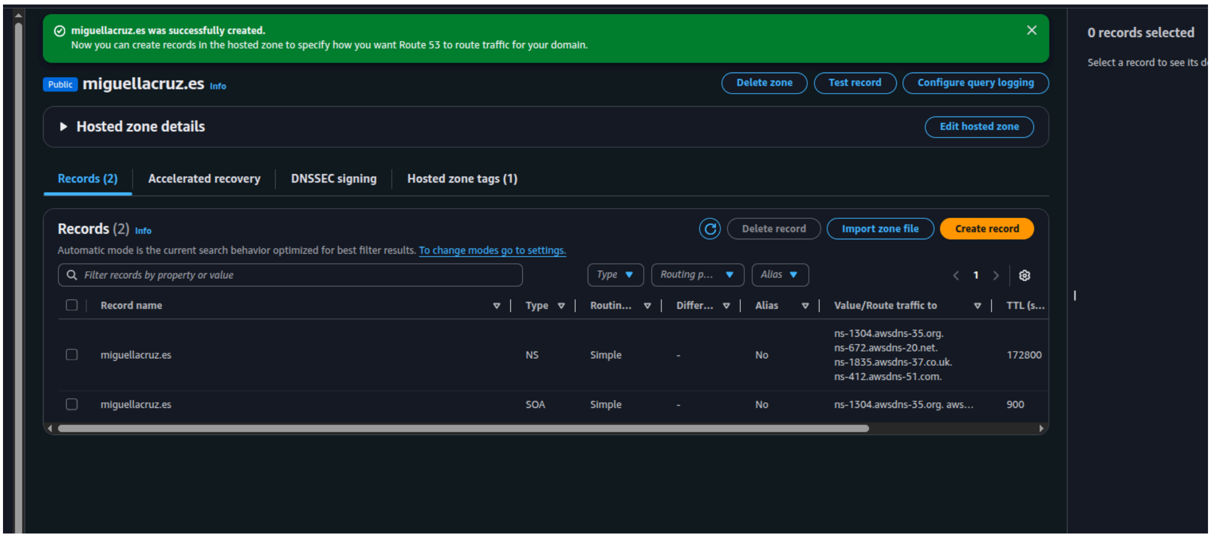
● Public hosted zone
A public hosted zone determines how traffic is routed on the internet.

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

Tags Info
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.



miguellacruz.es 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.

Public **miguellacruz.es** Info

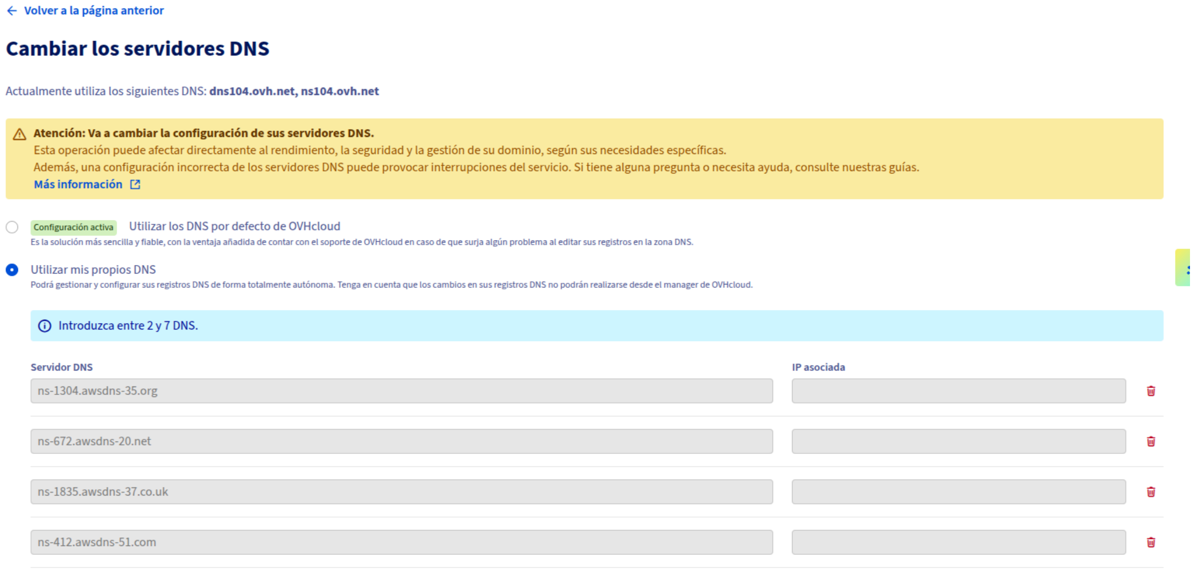
Records (2) Info

Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

Filter records by property or value
Type
Routing p...
Alias
0 records selected
Select a record to see its d

| □ | Record name | Type | Routin... | Differ... | Alias | Value/Route traffic to | TTL (s... |
|---|---|---|---|---|---|---|---|
| □ | miguellacruz.es | NS | Simple | - | No | ns-1304.awsdns-35.org.<br>ns-672.awsdns-20.net.<br>ns-1835.awsdns-37.co.uk.<br>ns-412.awsdns-51.com. | 172800 |
| □ | miguellacruz.es | SOA | Simple | - | No | ns-1304.awsdns-35.org. aws... | 900 |

- Register the values in Values/Route traffic to on the domain provider, in my case OVH, don't need to add an IP



← Volver a la página anterior

**Cambiar los servidores DNS**

Actualmente utiliza los siguientes DNS: **dns104.ovh.net, ns104.ovh.net**

**Atención: Va a cambiar la configuración de sus servidores DNS.**
Esta operación puede afectar directamente al rendimiento, la seguridad y la gestión de su dominio, según sus necesidades específicas.
Además, una configuración incorrecta de los servidores DNS puede provocar interrupciones del servicio. Si tiene alguna pregunta o necesita ayuda, consulte nuestras guías.
Más información

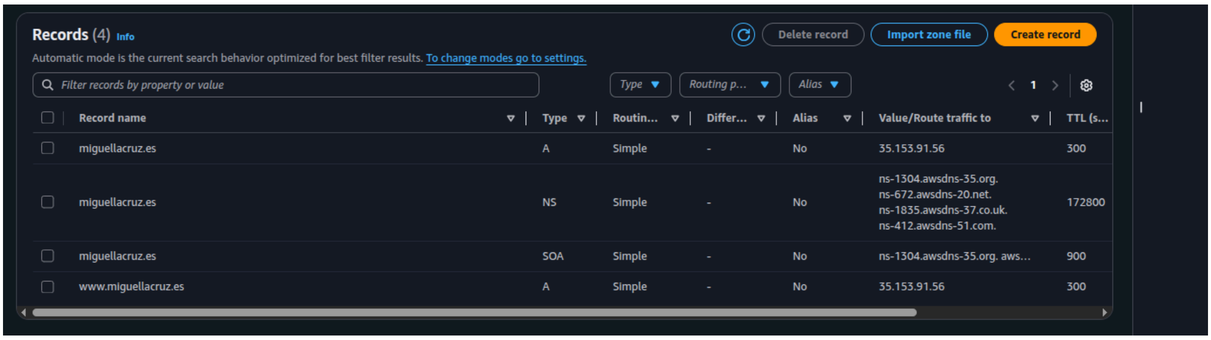
○ Configuración activa Utilizar los DNS por defecto de OVHcloud
Es la solución más sencilla y fiable, con la ventaja añadida de contar con el soporte de OVHcloud en caso de que surja algún problema al editar sus registros en la zona DNS.

● Utilizar mis propios DNS
Podrá gestionar y configurar sus registros DNS de forma totalmente autónoma. Tenga en cuenta que los cambios en sus registros DNS no podrán realizarse desde el manager de OVHcloud.

Introduzca entre 2 y 7 DNS.

| Servidor DNS | IP asociada |
|---|---|
| ns-1304.awsdns-35.org | |
| ns-672.awsdns-20.net | |
| ns-1835.awsdns-37.co.uk | |
| ns-412.awsdns-51.com | |

- Create A records with the IP value and domain name



Records (4) Info

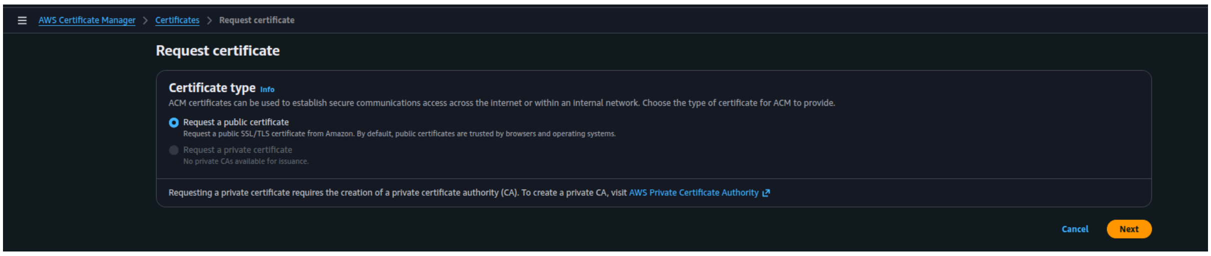
Automatic mode is the current search behavior optimized for best filter results. To change modes go to settings.

| □ | Record name | Type | Routin... | Differ... | Alias | Value/Route traffic to | TTL (s... |
|---|---|---|---|---|---|---|---|
| □ | miguellacruz.es | A | Simple | - | No | 35.153.91.56 | 300 |
| □ | miguellacruz.es | NS | Simple | - | No | ns-1304.awsdns-35.org.<br>ns-672.awsdns-20.net.<br>ns-1835.awsdns-37.co.uk.<br>ns-412.awsdns-51.com. | 172800 |
| □ | miguellacruz.es | SOA | Simple | - | No | ns-1304.awsdns-35.org. aws... | 900 |
| □ | www.miguellacruz.es | A | Simple | - | No | 35.153.91.56 | 300 |

Note: Could last 1-2hrs to propagate NS from provider to AWS NS.

## Request a certificate

Go to “Certificate manager” and then “request certificate”.



AWS Certificate Manager > Certificates > Request certificate

Request certificate

Certificate type Info

ACM certificates can be used to establish secure communications access across the internet or within an internal network. Choose the type of certificate for ACM to provide.

● Request a public certificate
Request a public SSL/TLS certificate from Amazon. By default, public certificates are trusted by browsers and operating systems.

○ Request a private certificate
No private CAs available for issuance.

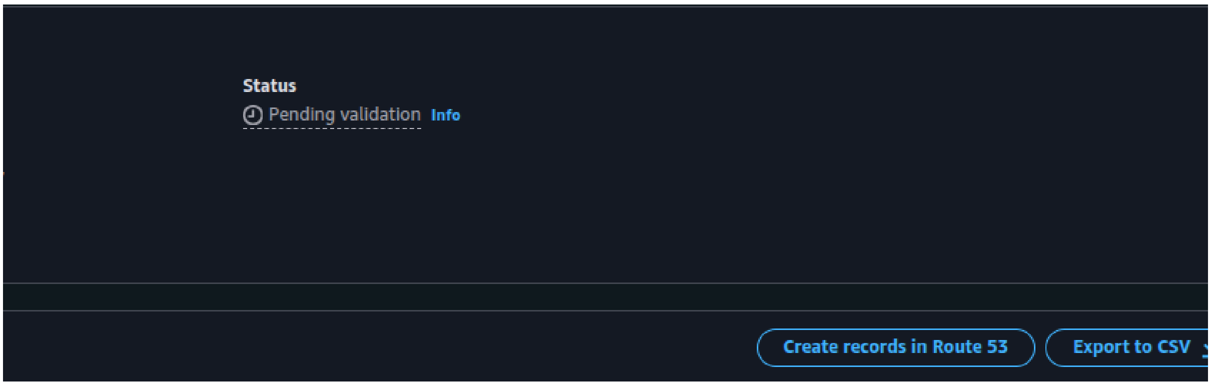
Requesting a private certificate requires the creation of a private certificate authority (CA). To create a private CA, visit AWS Private Certificate Authority

Settings:

- Public
- domain created before
- enable export
- validation by DNS
- RSA algorithm

## Create routes in Route 53

- Click on create records in Route 53.



Status
Pending validation Info

It will say to create it within a route (domain already created) click ok and continue.

- When status of certificate is “issued” export the certificate (3 files).
- Send 3 files to EC2.
- Unblock (no encrypt) the private key.

## Server config ssl

- Create folder for ssl

sudo mkdir -p /etc/nginx/ssl

- Put the certificates on the folder

sudo cp certificate.txt /etc/nginx/ssl/certificate.crt
sudo cp certificate_chain.txt /etc/nginx/ssl/ca_bundle.crt

```
sudo cp private_key.txt /etc/nginx/ssl/private.key
```

- Permissions for security

```
sudo chmod 600 /etc/nginx/ssl/private.key
sudo chmod 644 /etc/nginx/ssl/certificate.crt /etc/nginx/ssl/ca_bundle.crt
```

## Final

- Check and reload nginx

```
sudo nginx -t
sudo systemctl reload nginx
```

- Run app

In my case use run.sh with the command:

```
npm run dev -- --host 0.0.0.0 --port 5173
```