Connecting to the VPS

To connect your VPS server, you can use your server IP, you can create a root password and enter the server with your IP address and password credentials. But the more secure way is using an SSH key.

Creating SSH Key

For MAC OS / Linux / Windows 10 (with openssh)

Launch the Terminal app.

ssh-keygen -t rsa

Press ENTER to store the key in the default folder /Users/lamadev/.ssh/id\_rsa).

Type a passphrase (characters will not appear in the terminal).

Confirm your passphrase to finish SSH Keygen. You should get an output that looks something like this:

Your public key has been saved in /Users/lamadev/.ssh/id\_rsa.pub.

The key fingerprint is:

ae:89:72:0b:85:da:5a:f4:7c:1f:c2:43:fd:c6:44:30 lamadev@mac.local

The key's randomart image is:

+--[ RSA 2048]----+

| |

| . |

| E . |

| . . o |

| o . . S . |

| + + o . + |

|. + o = o + |

| o...o \* o |

|. oo.o . |

+-----------------+

Copy your public SSH Key to your clipboard using the following code: pbcopy < ~/.ssh/id\_rsa.pub

For Windows

Download PuTTY and PuTTYgen.

Open up PuTTYgen and click the Generate.

Copy your key.

Enter a key passphrase and confirm.

Save the private key.

Connection

After copying the SSH Key go the to hosting service provider dashboard and paste your key and save. After,

For MAC OS / Linux

ssh root@<server ip address>

For Windows

Open the PuTTY app.

Enter your IP address.

Open the following section: Connection - SSH - Auth

Browse the folders and choose your private key.

First Configuration

Deleting apache server

systemctl stop apache2

systemctl disable apache2

apt remove apache2

to delete related dependencies:

apt autoremove

Cleaning and updating server

apt clean all && sudo apt update && sudo apt dist-upgrade

rm -rf /var/www/html

Installing Nginx

apt install nginx

Installing and configure Firewall

apt install ufw

ufw enable

ufw allow "Nginx Full"

First Page

Delete the default server configuration

rm /etc/nginx/sites-available/default

rm /etc/nginx/sites-enabled/default

First configuration

nano /etc/nginx/sites-available/netflix

server {

listen 80;

location / {

root /var/www/netflix;

index index.html index.htm;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

try\_files $uri $uri/ /index.html;

}

}

ln -s /etc/nginx/sites-available/netflix /etc/nginx/sites-enabled/netflix

Write your fist message

nano /var/www/netflix/index.html

Start Nginx and check the page

systemctl start nginx

Uploading Apps Using Git

apt install git

mkdir netflix

cd netflix

git clone <your repository>

Nginx Configuration for new apps

nano /etc/nginx/sites-available/netflix

location /api {

proxy\_pass http://45.90.108.107:8800;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

If you check the location /api you are going to get "502" error which is good. Our configuration works. The only thing we need to is running our app

apt install nodejs

apt install npm

cd api

npm install

nano .env

Copy and paste your env file

node index.js

But if you close your ssh session here. It's gonna kill this process. To prevent this we are going to need a package which is called pm2

npm i -g pm2

Let's create a new pm2 instance

pm2 start --name api index.js

pm2 startup ubuntu

React App Deployment

cd ../client

nano .env

Paste your env file.

npm i

Let's create the build file

npm run build

Right now, we should move this build file into the main web file

rm -rf /var/www/netflix/\*

mkdir /var/www/netflix/client

cp -r build/\* /var/www/netflix/client

Let's make some server configuration

location / {

root /var/www/netflix/client/;

index index.html index.htm;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

try\_files $uri $uri/ /index.html;

}

Adding Domain

1 - Make sure that you created your A records on your domain provider website.

2 - Change your pathname from Router

3 - Change your env files and add the new API address

4 - Add the following server config

server {

listen 80;

server\_name safakkocaoglu.com www.safakkocaoglu.com;

location / {

root /var/www/netflix/client;

index index.html index.htm;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

try\_files $uri $uri/ /index.html;

}

}

server {

listen 80;

server\_name api.safakkocaoglu.com;

location / {

proxy\_pass http://45.90.108.107:8800;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

}

}

server {

listen 80;

server\_name admin.safakkocaoglu.com;

location / {

root /var/www/netflix/admin;

index index.html index.htm;

proxy\_http\_version 1.1;

proxy\_set\_header Upgrade $http\_upgrade;

proxy\_set\_header Connection 'upgrade';

proxy\_set\_header Host $host;

proxy\_cache\_bypass $http\_upgrade;

try\_files $uri $uri/ /index.html;

}

}

SSL Certification

apt install certbot python3-certbot-nginx

Make sure that Nginx Full rule is available

ufw status

certbot --nginx -d example.com -d www.example.com

Let’s Encrypt’s certificates are only valid for ninety days. To set a timer to validate automatically:

systemctl status certbot.timer

Additional Command

Systemctl –failed

System <service> restart