### VagrantFile

*Vagrant*.configure("2") do |config|

if *Vagrant*.has\_plugin? "vagrant-vbguest"

config.vbguest.no\_install = true

config.vbguest.auto\_update = false

config.vbguest.no\_remote = true

end

config.vm.define :servidor do |servidor|

config.vm.box = "bento/centos-stream-9"

config.vm.box\_version = "202301.20.0"

servidor.vm.provision "shell", path: "servidor.sh.txt"

servidor.vm.network :private\_network, ip: "192.168.50.4"

servidor.vm.hostname = "servidor"

servidor.vm.provider "virtualbox" do |v|

v.cpus = 2

end

end

end

### 

### Aprovisionamiento

#!/bin/bash

echo "Instalación de las herramientas necesarias"

sudo yum update -y

sudo yum install -y vim

sudo yum install -y net-tools

sudo yum install -y epel-release

sudo yum install -y python3-pip

sudo pip3 install Flask

wget https://repo.mysql.com//mysql80-community-release-el9-1.noarch.rpm

sudo rpm -Uvh mysql80-community-release-el9-1.noarch.rpm

sudo dnf install -y mysql mysql-server

sudo systemctl start mysqld

sudo systemctl enable mysqld

sudo yum install -y mysql-devel

sudo yum install -y python3-devel

sudo yum install -y gcc

sudo pip3 install flask-mysqldb

sudo yum install -y git

echo "configurando el selinux para desactivarlo"

cat <<TEST> /etc/selinux/config

SELINUX=disabled

SELINUXTYPE=targeted

TEST

### Punto 1

#### Punto a

1. Levante y aprovisiona la máquina:

**vagrant up**

1. Iniciar sesión SSH en el servidor:

**vagrant ssh servidor**

1. clone el repositorio en: <https://github.com/omondragon/APIRestFlaskMySQL>

**git clone** [**https://github.com/omondragon/APIRestFlaskMySQL**](https://github.com/omondragon/APIRestFlaskMySQL)

1. Ingrese al directorio del proyecto clonado: cd APIRestFlaskMySQL/

**cd APIRestFlaskMySQL/**

1. Una password aleatoria se genera cuando instalamos la maquina, en mi caso fue “L#N\_Thfkp0cs”, para ver esta password podemos verla en los registros de la siguiente manera

qrgpH/:Wj6:j

**sudo vim /var/log/mysqld.log**

1. Instalar mysql seguro

**sudo mysql\_secure\_installation**

**new pass: \_Houselins200212**

**\_Santiago2022**

1. Acceder a mysql

**mysql -u root -p**

pide password: **\_Houselins200212**

creamos la base de datos: **create database myflaskapp;**

accedemos a la base de datos: **use myflaskapp;**

creamos las tablas a usar: **CREATE TABLE books (**

**id INT AUTO\_INCREMENT PRIMARY KEY,**

**title VARCHAR(255) NOT NULL,**

**description TEXT,**

**author VARCHAR(255) NOT NULL**

**);**

**CREATE TABLE books (**

**id int NOT NULL AUTO\_INCREMENT PRIMARY KEY,**

**title varchar(255),**

**description varchar(255),**

**author varchar(255)**

**);**

**INSERT INTO books VALUES(null, "La hojarasca", "Interesante", "Gabo"),**

**(null, "El principito", "Brillante", "Antoine de Saint");**

**~**

1. Ejecutar el Código:

**export FLASK\_APP=apirest\_mysql.py**

**python3 -m flask run --host=0.0.0.0**

1. Compruebe el funcionamiento del código, el cual consiste en un API Rest en Python Flask con persistencia de datos usando una base de datos MySQL. Para eso use herramientas como curl o postman

**GET ALL BOOKS**

**curl -i http://localhost:5000/books**

**GET BOOK BY ID**

**curl -i** [**http://localhost:5000/books/{id**](http://localhost:5000/books/%7Bid)**}**

**ADD BOOK**

**curl -i -H "Content-Type: application/json" -X POST -d '{"title": "Luna de pluton v2", "author": "Dross", "description": "Mi libro luna de pluton v2"}'** [**http://localhost:5000/books**](http://localhost:5000/books)

**EDIT A BOOK**

**curl -i -H "Content-Type: application/json" -X PUT -d '{"author":"Jorgito"}' http://localhost:5000/books/{id}**

**DELETE A BOOK**

**curl -i -H "Content-Type: application/json" -X DELETE http://localhost:5000/books/{id}**

#### Punto b

1. Iniciar proyecto de Node.js:

**git clone** [**https://github.com/BastianTian/nodeConsumoApiBasico.git**](https://github.com/BastianTian/nodeConsumoApiBasico.git)

1. Nos dirigimos al directorio

**cd nodeConsumoApiBasico**

**sudo yum install -y nodejs**

**npm install express axios**

**npm install body-parser**

1. Instalamos librerías

**npm install**

1. Corremos el proyecto

**node app.js**

### 

### 

### Punto 2)

[1.5 Puntos (Funcionamiento + Sustentación)] SERVICIO + FIREWALL. Instalar el servidor de streaming Streama protegido por Firewall como se muestra en la figura. Todas las solicitudes hacia el servidor Streama deberán ser realizadas al firewall y no directamente al servicio configurado. El firewall debe redirigir las peticiones al servicio. Compruebe el funcionamiento desde el navegador del anfitrión y del SmartPhone

# -- mode: ruby --

# vi: set ft=ruby :

Vagrant.configure("2") do |config|

config.vm.define :firewall do |firewall|

firewall.vm.provision "shell", path: "firewall.sh.txt"

firewall.vm.box = "generic/centos8"

firewall.vm.network :public\_network, ip: "192.168.20.118"

firewall.vm.network :private\_network, ip: "192.168.50.2"

firewall.vm.hostname = "firewall"

end

config.vm.define :stream do |stream|

stream.vm.provision "shell", path: "stream.sh.txt"

stream.vm.box = "generic/centos8"

stream.vm.network :private\_network, ip: "192.168.50.3"

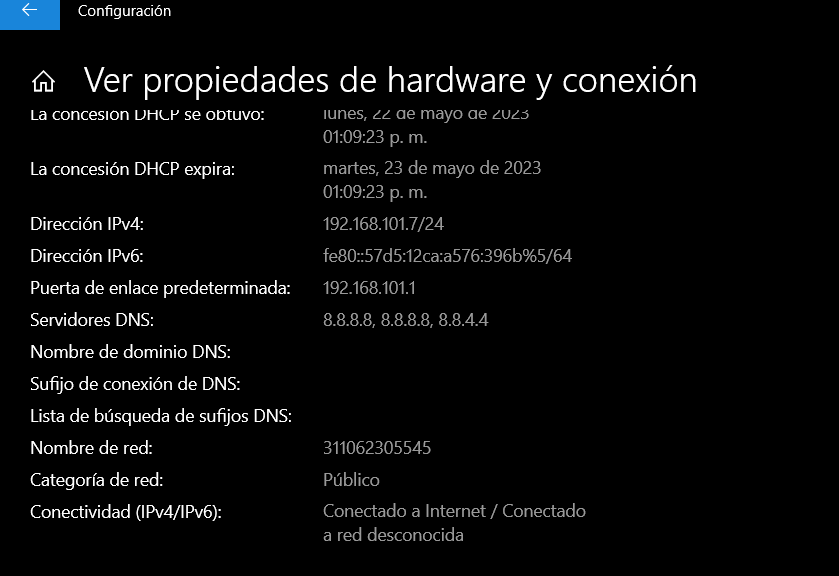
stream.vm.hostname = "stream"

end

end

PONER ATENCION!!!!!!!!!

firewall.vm.network :public\_network, ip: "192.168.20.118" esta línea cambia a tu IP, debe ser la ip de tu router/wifiiii



entras a propiedades de red y buscas tu wifi y buscas puerta de enlace predeterminada!!! cambias el ultimo numero y ya

aprovisionamiento firewalld:

#!/bin/bash

echo "configurando el selinux para desactivarlo"

sudo cat <<TEST> /etc/selinux/config

# This file controls the state of SELinux on the system.

# SELINUX= can take one of these three values:

# enforcing - SELinux security policy is enforced.

# permissive - SELinux prints warnings instead of enforcing.

# disabled - No SELinux policy is loaded.

SELINUX=disabled

# SELINUXTYPE= can take one of three values:

# targeted - Targeted processes are protected,

# minimum - Modification of targeted policy. Only selected processes are protected.

# mls - Multi Level Security protection.

SELINUXTYPE=targeted

TEST

echo "instalando vim"

sudo yum install vim -y

sudo yum install bind-utils bind-libs bind-\* -

echo "detener servicios de networkmanager"

sudo service NetworkManager stop

sudo chkconfig NetworkManager off

echo "iniciar firewall"

sudo service firewalld start

sudo firewall-cmd --zone=public --remove-interface=eth0 --permanent

sudo firewall-cmd --zone=public --remove-interface=eth2 --permanent

sudo firewall-cmd --zone=public --remove-interface=eth1 --permanent

sudo firewall-cmd --zone=public --add-interface=eth1 --permanent

sudo firewall-cmd --zone=public --add-service=ftp --permanent

sudo firewall-cmd --zone=public --add-service=dns --permanent

sudo firewall-cmd --zone=public --add-masquerade --permanent

sudo firewall-cmd --permanent --zone=public --add-forward-port=port=80:proto=tcp:toport=8080:toaddr=192.168.50.3

firewall-cmd --zone=public --add-service=http --permanent

sudo firewall-cmd --zone=internal --add-masquerade --permanent

sudo firewall-cmd --reload

sudo firewall-cmd --list-all-zones

echo "configurando el resolv.conf con cat"

sudo cat <<TEST> /etc/resolv.conf

nameserver 192.168.50.3

TEST

Aprovisionamiento stream:

#!/bin/bash

echo "Instalación de las herramientas necesarias"

sudo yum update -y

sudo yum install -y vim

sudo yum install -y net-tools

sudo yum -y install java-1.8.0-openjdk

wget https://github.com/dularion/streama/releases/download/v1.1/streama-1.1.war

echo "Creando las carpetas y archivos necesarios"

mkdir /opt/streama

mv streama-1.1.war /opt/streama/streama.war

mkdir /opt/streama/media

chmod 664 /opt/streama/media

echo "creando y modificando archivo"

sudo cat <<TEST> /etc/systemd/system/streama.service

[Unit]

Description=Streama Server

After=syslog.target

After=network.target

[Service]

User=root

Type=simple

ExecStart=/bin/java -jar /opt/streama/streama.war

Restart=always

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=Streama

[Install]

WantedBy=multi-user.target

TEST

echo "servicio streama"

systemctl start streama

systemctl enable streama

systemctl status streama

service firewalld stop

echo "configurando el selinux para desactivarlo"

cat <<TEST> /etc/selinux/config

SELINUX=disabled

SELINUXTYPE=targeted

TEST

vagrant ssh stream

sudo -i

DESACTIVAR SELINUX

vim /etc/selinux/config

PARAR EL SERVICIO FIREWALLD

service firewalld stop

yum -y update

yum -y install wget

sudo yum -y install java-1.8.0-openjdk

java -version \*\*verificar java

wget <https://github.com/dularion/streama/releases/download/v1.1/streama-1.1.war> \*\* descargar el stream esa mierda

mkdir /opt/streama

mv streama-1.1.war /opt/streama/streama.war

*java -jar /opt/streama/streama.war*

Corremos

mkdir /opt/streama/media \*\* crearemos directorio para ejecución

chmod 664 /opt/streama/media\*\*permisos

creamos archivo

vim /etc/systemd/system/streama.service

En el archivo creado metemos este codigo

[Unit]

Description=Streama Server

After=syslog.target

After=network.target

[Service]

User=root

Type=simple

ExecStart=/bin/java -jar /opt/streama/streama.war

Restart=always

StandardOutput=syslog

StandardError=syslog

SyslogIdentifier=Streama

[Install]

WantedBy=multi-user.target

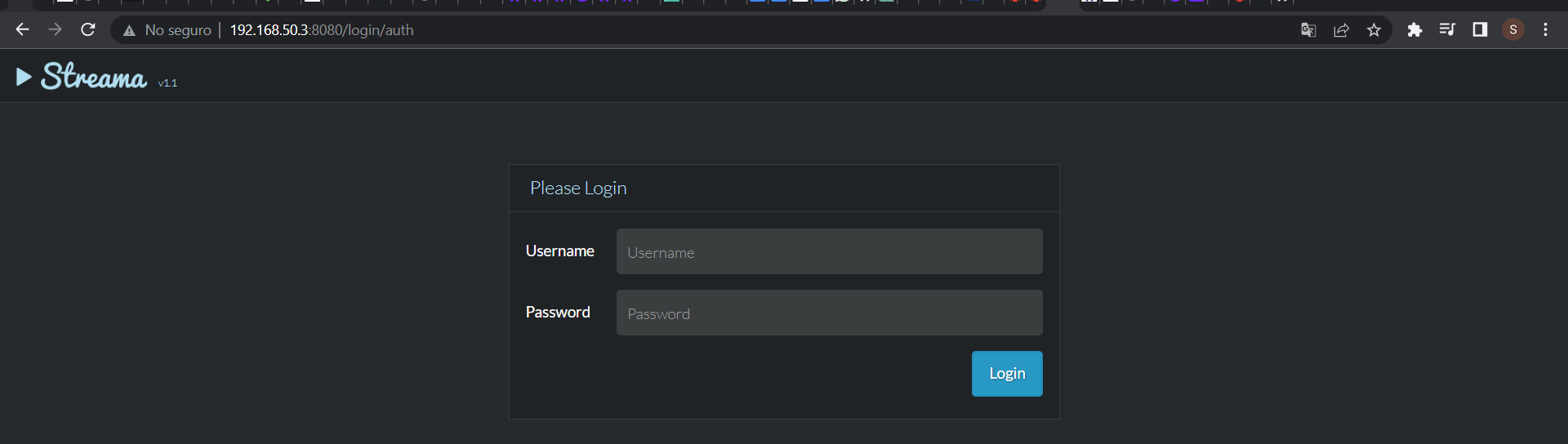
--------------------------

systemctl start streama

systemctl enable streama

systemctl status streama

<http://192.168.50.3:8080/>



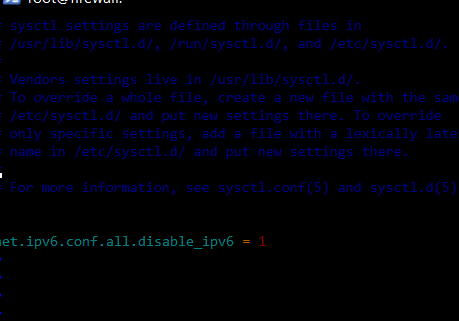
Auth: admin

Auth: admin

vagrant ssh firewall

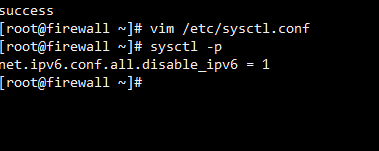
sudo -i

vim vim /etc/sysctl.conf



cambiamos a 1

sysctl -p



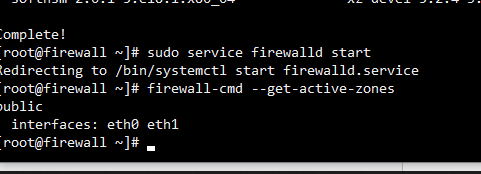
selinux desactivar

sudo service NetworkManager stop

sudo chkconfig NetworkManager off

sudo service firewalld start

firewall-cmd --get-active-zones



active

sudo firewall-cmd --zone=public --remove-interface=eth2 --permanent

sudo firewall-cmd --zone=public --remove-interface=eth1 --permanent

\*\*Removemos porque vienen en defecto en public/ puede que no esten entonces este paso no lo podemos omitir si no estan

sudo firewall-cmd --zone=public --add-interface=eth1 --permanent

sudo firewall-cmd --zone=public --add-service=ftp --permanent

sudo firewall-cmd --zone=public --add-service=dns --permanent

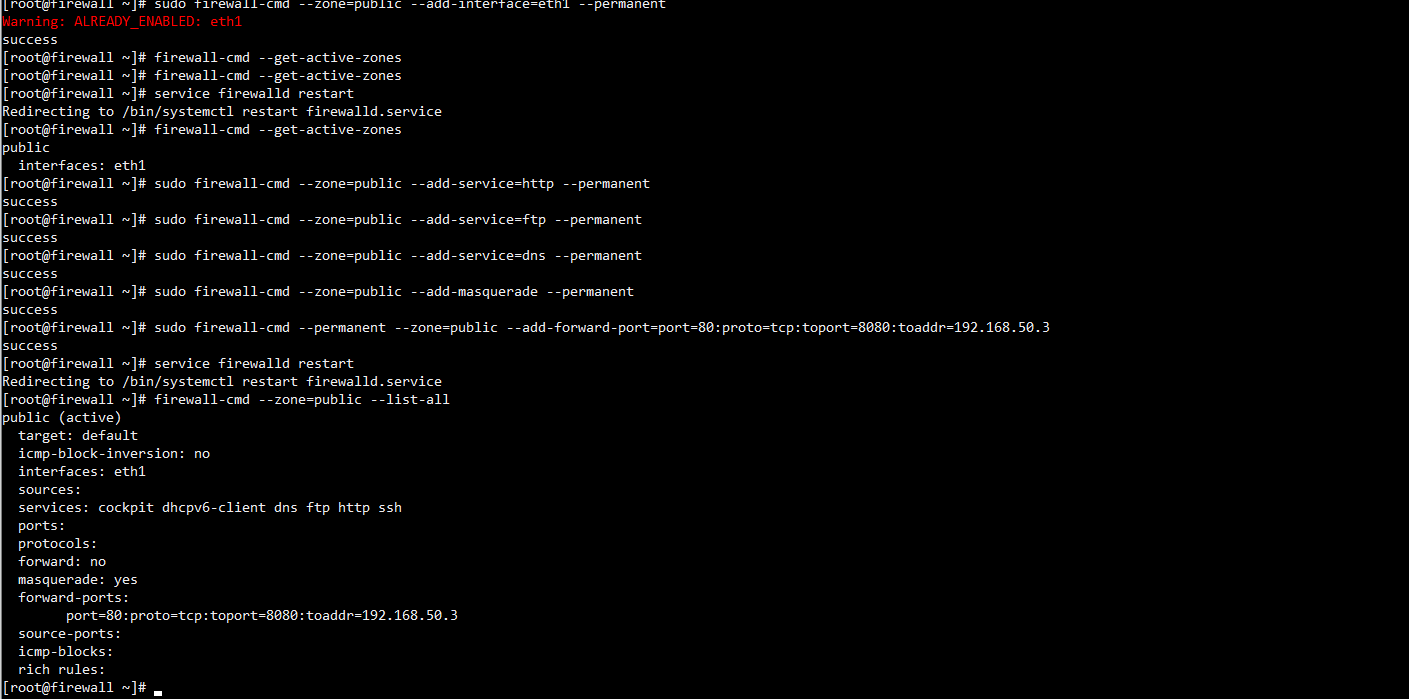
sudo firewall-cmd --zone=public --add-masquerade --permanent

sudo firewall-cmd --permanent --zone=public --add-forward-port=port=80:proto=tcp:toport=8080:toaddr=192.168.50.3

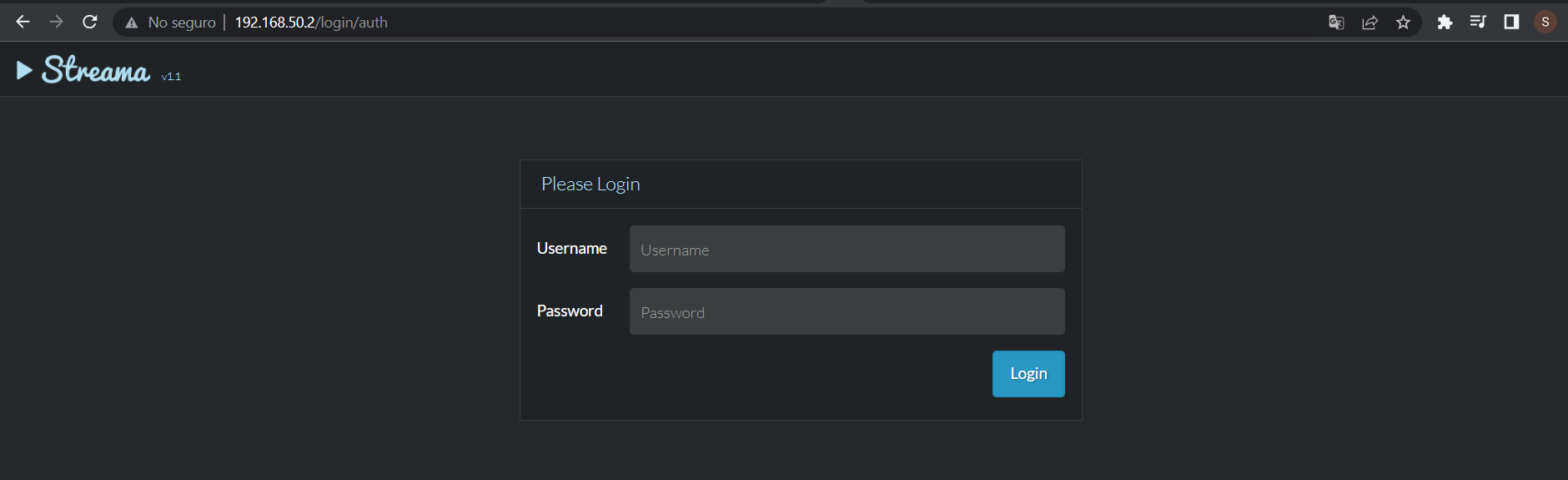
firewall-cmd --zone=public --add-service=http --permanent

firewall-cmd --zone=public --list-all

service firewalld restart \*\*para ver los cambiooosssss



<http://192.168.50.2/> debe redirrecionar a stream



Las estimaciones de software se pueden catalogar en 4 tipos tr

ERROR: Usar el AFP en momentos preliminares del ciclo de vida para estimar y nunca en la configuración final.