Proyecto Final CPD

Por: Rubén Calvo Villazán

Índice:

Realizado:

Nodos: 3 Servidores + 1 Balanceador + 1 Monitor = 5 Nodos

Redundancia: rsync entre los 3 servidores + rsync en el balanceador

Monitorizacion: Ansible en el nodo monitor

Seguridad: fail2ban, rkhunter, chrootkit, lynis

Despliegue realizado con **Vagrant (**https://www.vagrantup.com/**)**

Pasos a seguir:

El total de ficheros incluye un Vagrantfile y una serie de scripts de configuración.

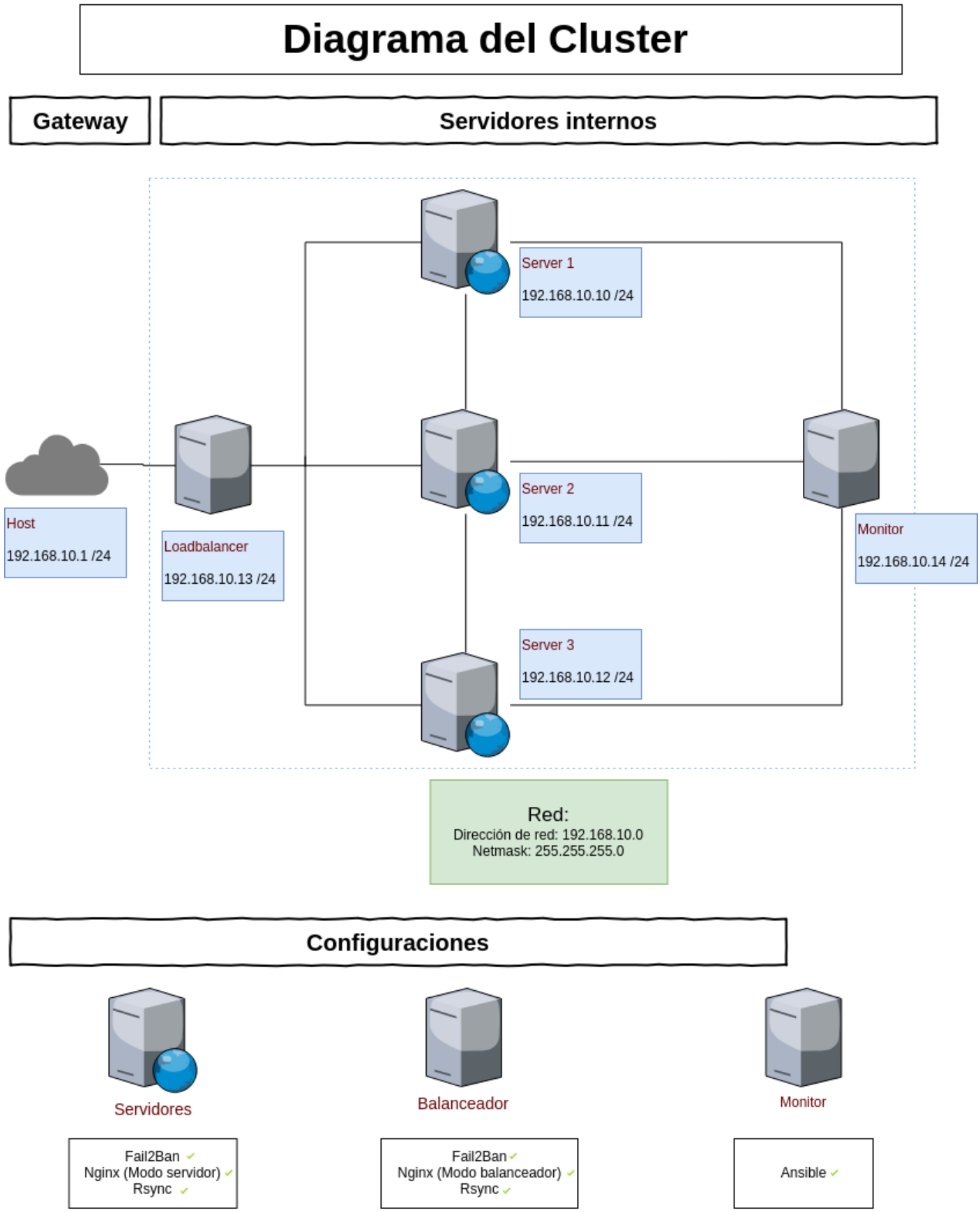
Crear un directorio, desplazar todos los ficheros dentro del directorio y lanzar Vagrant.
Ejecutar los scripts dentro de su nodo correspondiente para configurar el servicio.

$ mkdir ProyectoFinal
$ cd ProyectoFinal
$ ls ProyectoFinal
Vagrantfile scripts.sh
$ vagrant up

Dentro de cada máquina, los scripts se encuentran en el directorio vinculado del host
vagrant@server1 **$** ls /vagrant/
Vagrantfile scripts.sh

Para ejecutar un script por ejemplo el que configura nginx en el balanceador:
vagrant@loadbalancer **$** cp /vagrant/script.sh .
vagrant@loadbalancer **$** ./script.sh

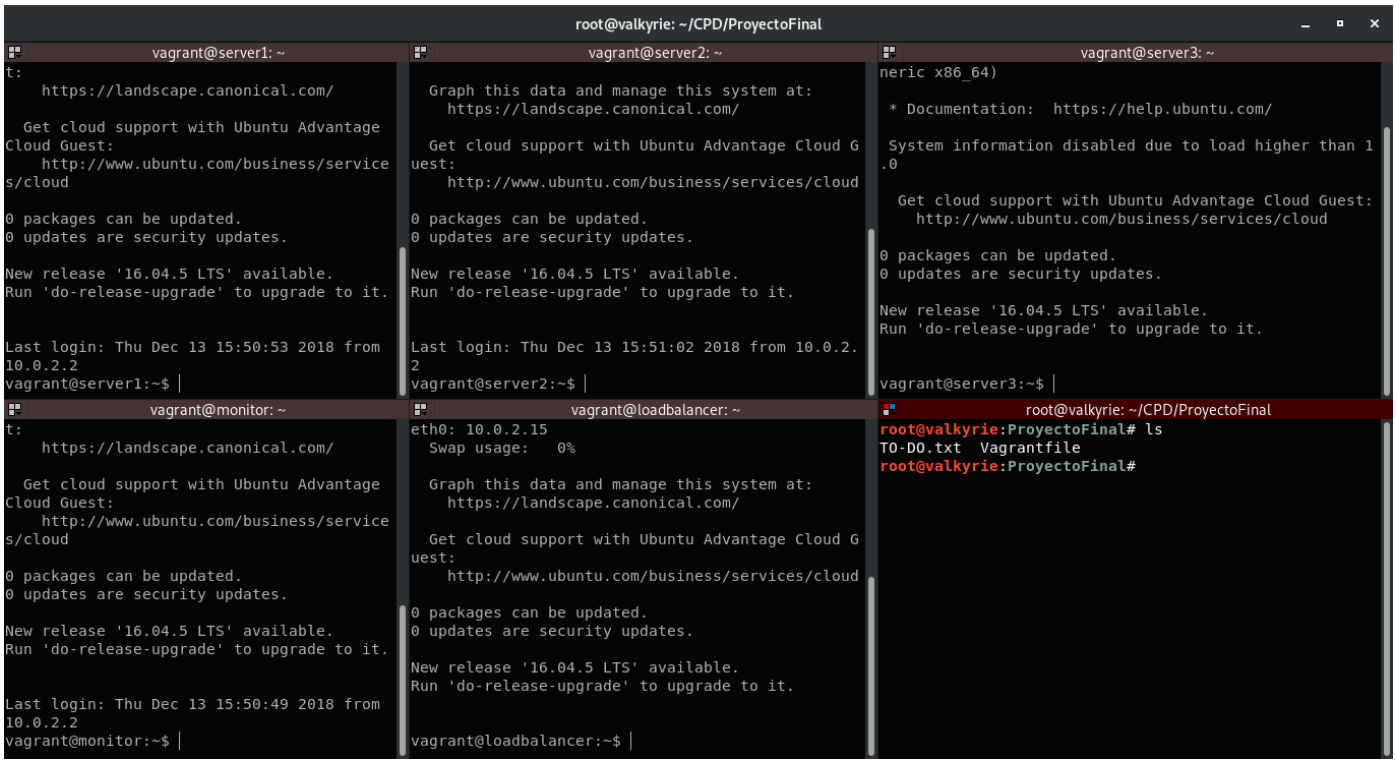
Diagrama del Cluster:



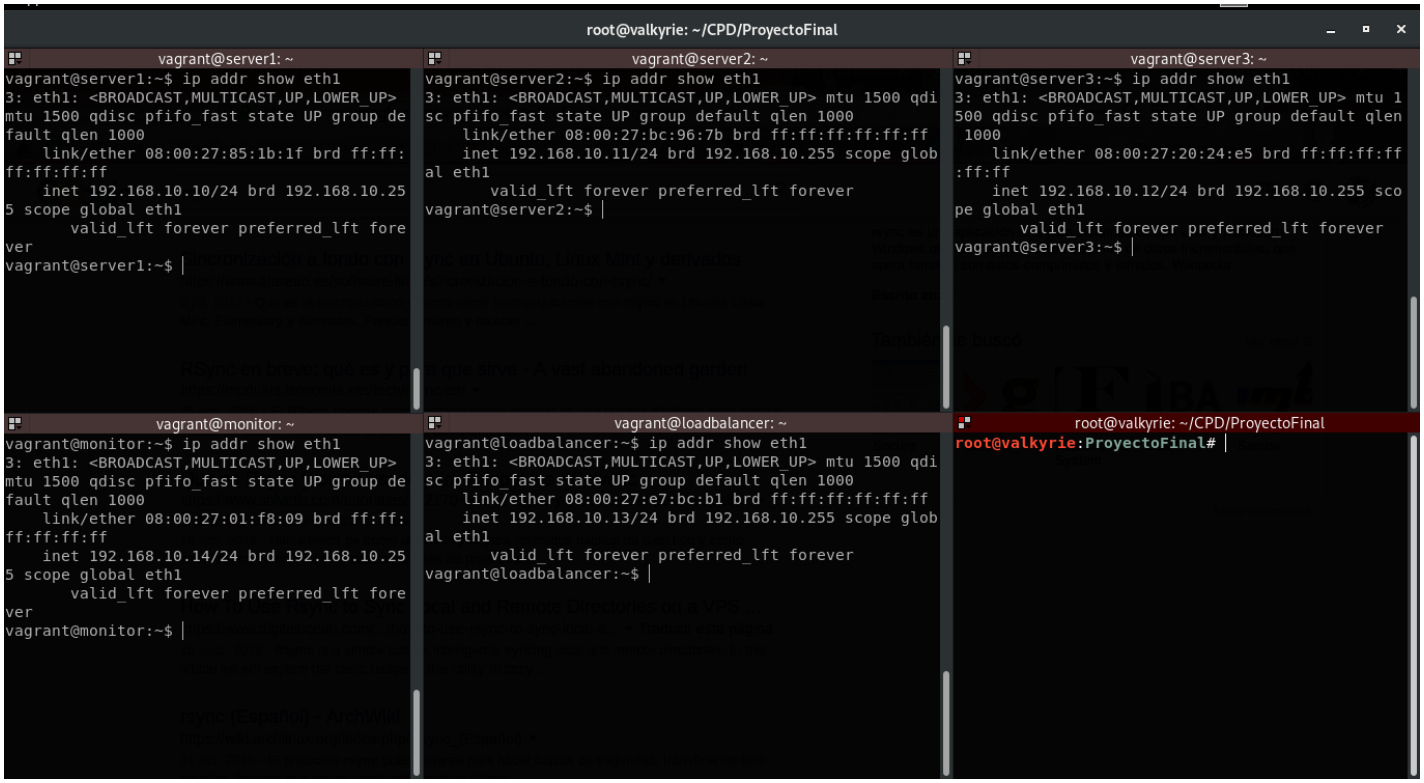
Vagrantfile:

```
## Cluster de servidores

identificadores = [ 'server1', 'server2', 'server3', 'loadbalancer', 'monitor'] # Añadir/Quitar nombres
servidores = []
num_servers = 5 # Cambiar si se añaden/quitan nodos (de forma dinamica)

dir_red = '192.168.10.' # 192.168.2. Para que no colapse con ninguna otra red WLAN
ip_inicial = 10 # Primera IP: 192.168.2.10


for i in 0..num_servers-1 do
  name = identificadores[i]
  ip = dir_red + (ip_inicial + i).to_s
  servidores << {'name' => name,
                 'ip' => ip,
                }
end

Vagrant::Config.run do |config|
  servidores.each do |server|
    config.vm.define server['name'] do |config_server|
      config_server.vm.box = "ubuntu/trusty64"
      config_server.vm.host_name = server['name']
      config_server.vm.network :hostonly, server['ip']

      if server['name'] == 'monitor'   # Si el nodo es monitor, instalamos ansible (monitorizar)
        config_server.vm.provision :shell, :inline => "
        echo 'SOY EL MONITOR'"
        # apt install software-properties-common
        # apt-add-repository --yes --update ppa:ansible/ansible
        # apt install ansible"
      else

        # DESCOMENTAR ESTAS LINEAS PARA PROVISIONAR LAS MAQUINAS (TARDA MAS)
        config_server.vm.provision :shell, :inline => "

        echo 'INSTALANDO: FAIL2BAN, NGINX, RSYNC'"


        # echo vagrant | sudo su -S ;
        # apt update; apt install fail2ban -y ;
        # apt install nginx -y;
        # apt install rsync -y;


        # apt install rkhunter -y;  # Descomentar para añadir mas funcionalidad
        # apt install chrootkit -y;
        # apt install lynis -y;
        #"
      end



      # NOTA:
      # Aqui solo realizamos las instalaciones, no configuramos nada porque un solo fallo en configuracion de
      # una maquina ya evita que se lancen y configuren las demas

      # Ejecutar scripts o configurar manualmente cada maquina

    end
  end
end
```

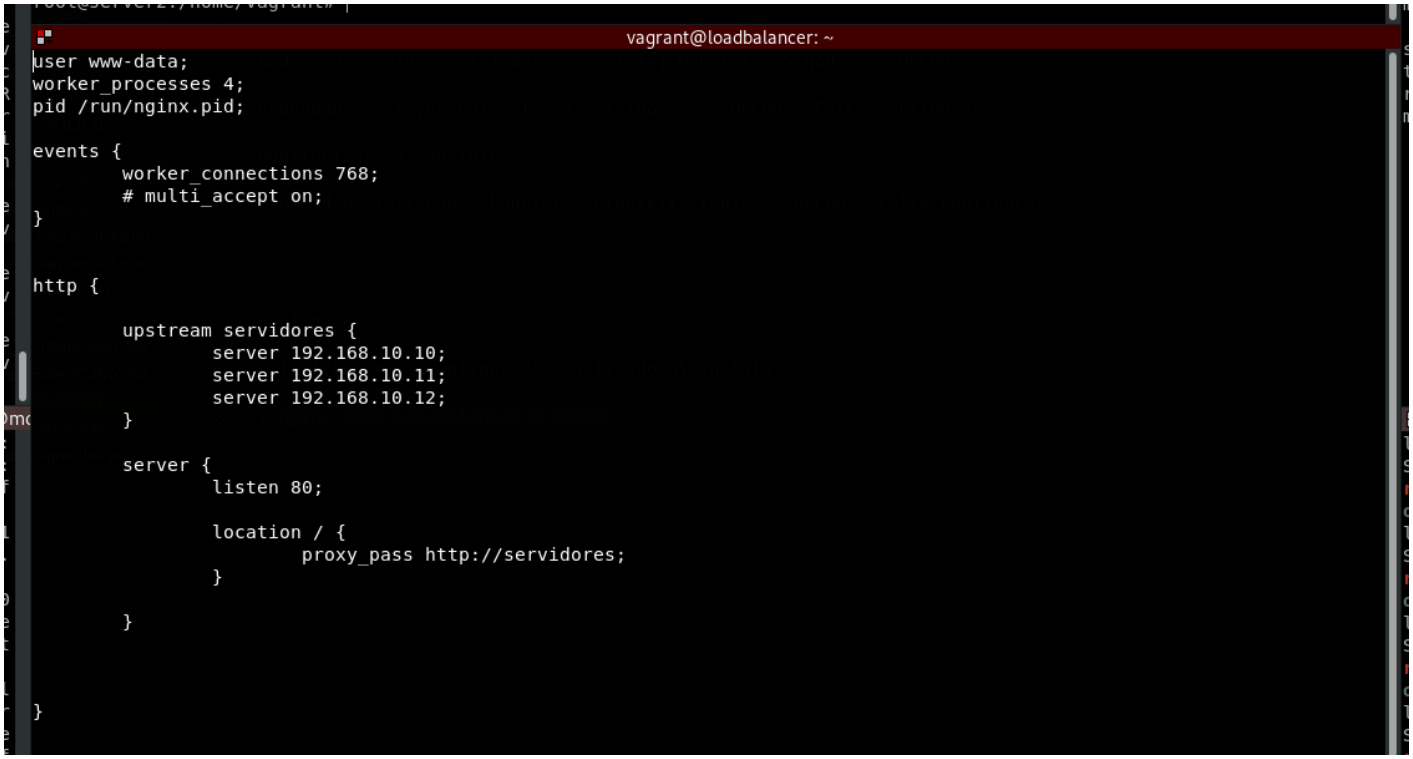
Nodos activos (SSH)



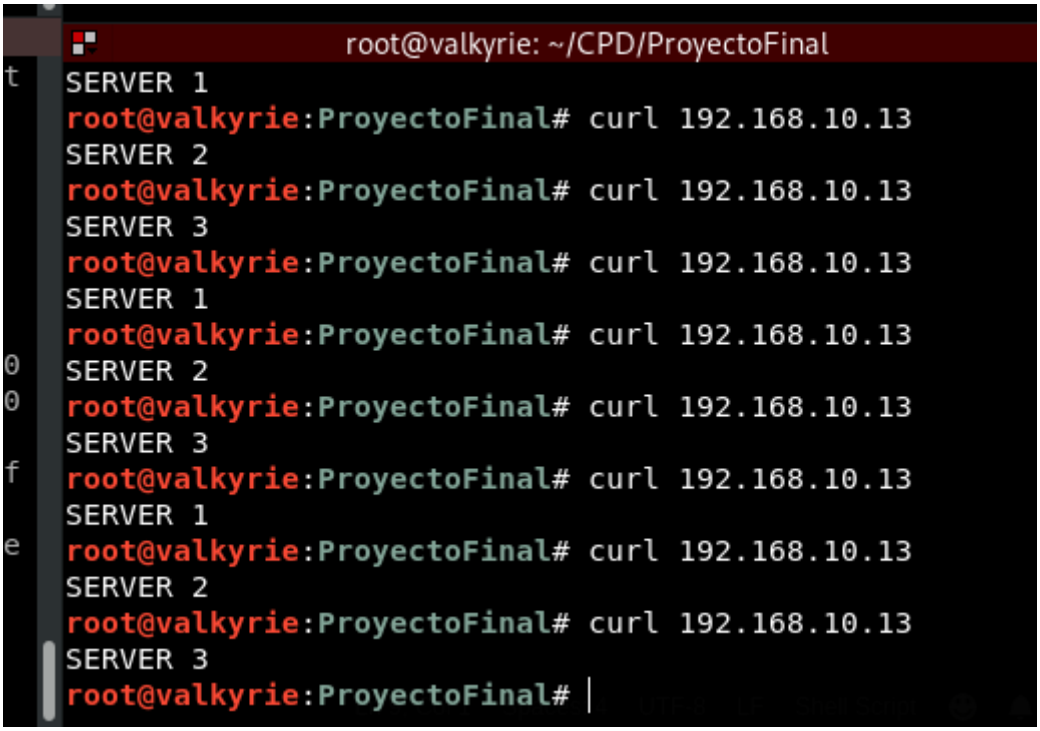
Red de los nodos (IP)



Configuración Nginx Load Balancer



Prueba de funcionamiento del Load Balancer



Script de configuración de rsync (redundancia)

```
#!/bin/bash

# Script para configurar rsync

# Ejecutar en un nodo y pasarle la IP del otro nodo a vincular como argumento


sudo su
mkdir /var/www/

exec rsync -avzhie "ssh" vagrant@$1:/var/www/ /var/www/


# Crontab que lo ejecuta cada minuto

echo "* * * * *     root    rsync -avzhie 'ssh' vagrant@$1:/var/www/ /var/www/" >> /etc/crontab
```

Script de configuración de Nginx como Load Balancer

```
#!/bin/bash


# Script para configurar nginx como un balanceador

# Ejecutar solo en el nodo loadbalancer


sudo su
cp /etc/nginx/nginx.conf /etc/nginx/nginxBK.conf

echo '
        user www-data;
        worker_processes 4;
        pid /run/nginx.pid;

        events {
                worker_connections 768;
                # multi_accept on;
        }


        http {

                upstream servidores {
                        server 192.168.10.10;
                        server 192.168.10.11;
                        server 192.168.10.12;
                }

                server {
                        listen 80;

                        location / {
                                proxy_pass http://servidores;
                        }

                }
        }' > /etc/nginx/nginx.conf

service nginx restart
```