

Informe de Configuración de un Ambiente de Computación Distribuido

Rui Lei Marco Riascos Benjamin Ortiz Julian Galvis

November 14, 2024

Contents

1	Objetivos	1
2	Descripción del Ejercicio y Configuración	2
2.1	Apache en Ubuntu Server	2
2.2	Nginx como Proxy Inverso en Ubuntu Desktop	2
2.3	Docker-Compose en Ubuntu Server	3
3	Errores y Resolución de Problemas	4
4	Pruebas y Verificación	4
5	Recursos Utilizados	4
6	Conclusión	5

1 Objetivos

Los objetivos principales de este ejercicio son los siguientes:

- Demostrar habilidades en la utilización de herramientas de virtualización como VirtualBox o KVM/QEMU.
- Instalar y configurar un servidor web Apache en una máquina virtual con Ubuntu Server.
- Configurar Nginx como proxy inverso en otra máquina virtual con Ubuntu Desktop.
- Ejecutar una aplicación basada en contenedores de Docker utilizando `docker-compose` en una máquina virtual adicional.

2 Descripción del Ejercicio y Configuración

En este ejercicio, hemos implementado una configuración distribuida utilizando tres máquinas virtuales (VMs) configuradas de la siguiente manera:

2.1 Apache en Ubuntu Server

- **VM:** Ubuntu Server
- **IP Asignada:** 192.168.28.32
- **Descripción de la Configuración:**
 1. Instalamos Apache utilizando el siguiente comando:

```
sudo apt update
sudo apt install apache2 -y
```
 2. Verificamos que el servicio de Apache esté activo y habilitado para que inicie al arrancar la máquina:

```
sudo systemctl status apache2
sudo systemctl enable apache2
```
 3. Configuramos el archivo de configuración de Apache para escuchar en la IP asignada:

```
sudo nano /etc/apache2/ports.conf
# Cambiamos o a adimos la linea Listen:
Listen 192.168.28.32:80
```
 4. Para verificar el acceso, abrimos un navegador e ingresamos `http://192.168.28.32`. Si el servidor estaba configurado correctamente, debía verse la página de bienvenida de Apache.

2.2 Nginx como Proxy Inverso en Ubuntu Desktop

- **VM:** Ubuntu Desktop (normal)
- **IP Asignada:** 192.168.28.30
- **Descripción de la Configuración:**
 1. Instalamos Nginx en Ubuntu Desktop con:

```
sudo apt update
sudo apt install nginx -y
```
 2. Configuramos Nginx para funcionar como proxy inverso, de modo que:

- Las solicitudes a `http://192.168.28.30/compose` se redirigen a la VM de Docker en la IP 192.168.28.31.
 - Las solicitudes a `http://192.168.28.30/apache` se redirigen a la VM de Apache en la IP 192.168.28.32.
3. Edición del archivo de configuración de Nginx:


```
sudo nano /etc/nginx/sites-available/default
```
 4. Añadimos la siguiente configuración en el archivo:


```
server {
    listen 80;
    location /compose {
        proxy_pass http://192.168.28.31;
    }
    location /apache {
        proxy_pass http://192.168.28.32;
    }
}
```
 5. Verificamos la configuración y reiniciamos el servicio de Nginx:


```
sudo nginx -t
sudo systemctl restart nginx
```

2.3 Docker-Compose en Ubuntu Server

- **VM:** Ubuntu Server
- **IP Asignada:** 192.168.28.31
- **Descripción de la Configuración:**
 1. Instalamos Docker y Docker Compose:


```
sudo apt update
sudo apt install docker.io -y
sudo apt install docker-compose -y
```
 2. Ejecutamos la aplicación utilizando `docker-compose`. El archivo de `docker-compose.yml` provisto se colocó en el directorio `example`.


```
cd example
docker-compose up -d
```

3 Errores y Resolución de Problemas

Durante el ejercicio, se presentaron varios errores que resolvimos de la siguiente manera:

- **Error con Apache en la IP asignada:** Inicialmente, Apache no estaba escuchando en 192.168.28.32. Esto se resolvió modificando el archivo `ports.conf` y reiniciando el servicio de Apache.
- **Nginx Proxy Inverso sin conexión:** Después de configurar el archivo de Nginx, el proxy inverso no funcionaba correctamente. Verificamos la sintaxis del archivo y corregimos un error en las directivas `proxy_pass`.
- **Permisos de Docker-Compose:** Al ejecutar `docker-compose`, surgieron problemas de permisos en el directorio donde se encontraba el archivo `docker-compose.yml`. Esto se solucionó asignando permisos correctos al usuario actual:

```
sudo chmod +x script.sh
./script.sh
```

4 Pruebas y Verificación

Para validar la funcionalidad del sistema distribuido, realizamos las siguientes pruebas:

1. **Prueba de conexión directa con Apache:**
 - Abrimos `http://192.168.28.32` en un navegador y verificamos que se mostraba la página de inicio de Apache.
2. **Prueba de redirección con Nginx:**
 - Accedimos a `http://192.168.28.30/apache` para verificar que la solicitud se redirigía correctamente a Apache.
 - Accedimos a `http://192.168.28.30/compose` para asegurar que se redirigía al servicio en Docker.
3. **Comprobación del Docker-Compose:**
 - Verificamos que el servicio de Docker estaba funcionando correctamente en 192.168.28.31.

5 Recursos Utilizados

Para completar este ejercicio, utilizamos los siguientes recursos:

- Documentación oficial de Apache y Nginx.

- Tutoriales de instalación y configuración en Ubuntu Server y Ubuntu Desktop.
- El archivo `docker-compose.yml` provisto por el instructor.
- Guías de configuración de red virtual de VirtualBox para el tipo `host-only adapter` y configuración de red de KVM con interfaz tipo `bridge`.

6 Conclusión

Este ejercicio permitió la implementación de un ambiente de computación distribuida con múltiples servicios en una red virtual. Documentamos los problemas encontrados y las soluciones aplicadas, y verificamos el correcto funcionamiento del sistema mediante pruebas en cada etapa.