

Informe del Ejercicio: Despliegue de un Ambiente Computacional Distribuido

Juan Diego Collazos Mejia, Juan Sebastian Garizao

November 22, 2024

Contents

1	Introducción	2
2	Descripción del Ejercicio	2
3	Objetivos	3
4	Metodología	3
4.1	Configuración de la Red en VirtualBox	3
4.2	Instalación y Configuración del Proxy Inverso	3
4.3	Despliegue del Aplicativo con Docker Compose	5
4.4	Instalación y Configuración del Servidor Web Apache	6
5	Razones para el Cambio de Direcciones IP	7
6	Desafíos y Soluciones	8
6.1	Problema con el Arranque Seguro en VirtualBox y Solución mediante Firma	8
7	Recursos Utilizados	10
7.1	Video de muestra	10
8	Conclusión	10

1 Introducción

En este ejercicio, el objetivo es realizar el despliegue de un ambiente computacional distribuido, que consiste en configurar e instalar varios servicios en máquinas virtuales (VM). Estos servicios incluyen un proxy inverso, un aplicativo ejecutado con Docker Compose, y un servidor web Apache, los cuales interactúan entre sí y con otros elementos de la infraestructura.

El ejercicio se lleva a cabo utilizando VirtualBox como plataforma de virtualización. Se crearon tres máquinas virtuales Kali Linux, y para permitir la comunicación entre ellas, se configuró una red "Host-only Network", conectando todas las máquinas con la configuración "All Done". Además, se utilizaron configuraciones de red específicas para cada servicio, garantizando que los servicios pudieran interactuar entre sí correctamente.

2 Descripción del Ejercicio

El entorno a desplegar está compuesto por los siguientes elementos:

- **Tu PC:** El equipo local desde donde se ejecutan las acciones.
- **Web Browser (Navegador):** Utilizado para interactuar con los servicios desplegados.
- **Proxy Inverso (IP: 192.168.28.30):** Ejecución en una máquina virtual Kali Linux que se encarga de redirigir las solicitudes.
- **Aplicativo con Docker Compose (IP: 192.168.28.31):** Máquina virtual Kali Linux donde se ejecuta el aplicativo dentro de un contenedor Docker.
- **Servidor Web Apache (IP: 192.168.28.32):** Máquina virtual Kali Linux que ejecuta un servidor Apache, que provee contenido web.

El desafío principal es realizar la configuración de los tres elementos restantes en las máquinas virtuales correspondientes y asegurarse de que interactúen correctamente dentro del ambiente distribuido.

3 Objetivos

Los objetivos principales del ejercicio son:

1. Configurar una máquina virtual para ejecutar un **proxy inverso**.
2. Desplegar una máquina virtual con **Docker Compose** para ejecutar un aplicativo.
3. Instalar y configurar un **servidor web Apache** en otra máquina virtual.
4. Asegurar la comunicación entre estos servicios para lograr el funcionamiento de la arquitectura distribuida.

4 Metodología

4.1 Configuración de la Red en VirtualBox

El ejercicio se realizó utilizando VirtualBox y configurando tres máquinas virtuales Kali Linux. Para permitir que estas máquinas se conectaran entre sí, se utilizó la red **Host-only Network**. Se creó una red y se conectaron todas las máquinas virtuales con la configuración de red **All Done**, asegurando que pudieran comunicarse sin problemas.

4.2 Instalación y Configuración del Proxy Inverso

El proxy inverso se configura en una máquina virtual con la IP 192.168.56.3. Para ello, se utilizó **nginx** como servidor para gestionar las solicitudes entrantes y redirigirlas a los servicios correspondientes. Los pasos fueron los siguientes:

- Instalación de **nginx**:

```
sudo apt update  
sudo apt install nginx -y
```

- Configuración del archivo de configuración de **nginx** (/etc/nginx/sites-available/default)

```
server {
    listen 80;
    server_name 192.168.56.3;

    # Configuración para la raíz
    location / {
        root /usr/share/nginx/html;
        index index.html index.htm;
    }

    # Configuración para "/apache"
    location /apache /{
        proxy_pass http://192.168.56.4/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }

    # Configuración para "/compose"
    location /compose/ {
        proxy_pass http://192.168.56.5:8000/;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    }
}
```

- Reiniciar **nginx** para aplicar los cambios:

```
sudo systemctl restart nginx
```

- Habilitar el servicio de **nginx** para que se inicie automáticamente al arrancar la máquina:

```
sudo systemctl enable nginx
```

El proxy inverso fue configurado para redirigir las solicitudes HTTP que lleguen al servidor de proxy inverso (192.168.56.3) hacia el servidor Apache (192.168.56.4) y al servicio de Docker Compose (192.168.56.5).

4.3 Despliegue del Aplicativo con Docker Compose

En la máquina virtual con la IP 192.168.56.5, se desplegó el aplicativo utilizando `docker-compose`, que permite ejecutar contenedores de Docker de manera fácil. Los pasos fueron:

- Instalación de Docker y Docker Compose:

```
sudo apt update
sudo apt install docker.io -y
sudo apt install docker-compose -y
```

- Creación de un archivo de script para descargar el aplicativo desde Google Drive y configurarlo:

```
nano script.sh
```

- Dentro del archivo `script.sh`, se copió el siguiente código:

```
#!/usr/bin/env bash
CURRENT_WORKDIR=$(pwd)
FILE_ID="1EH1zS1wQ_5WmvUEar2eXcB4QMj1Mo49g"
FILENAME="docker-compose-example.tgz"
mkdir example && cd example
wget --no-check-certificate "https://docs.google.com/uc?export=download&id=${FILE_ID}"
tar xzf ${FILENAME}
cd ${CURRENT_WORKDIR}
```

- Cambiar permisos de ejecución y ejecutar el script:

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

- Acceder al directorio `example` y ejecutar `docker-compose`:

```
cd example && docker-compose up
```

- Iniciar el servicio de Docker Compose:

```
sudo systemctl start docker
```

Este script descarga un archivo comprimido, lo descomprime, y configura el entorno necesario para ejecutar el aplicativo. Posteriormente, el aplicativo se ejecuta dentro de contenedores Docker mediante `docker-compose`.

4.4 Instalación y Configuración del Servidor Web Apache

En la máquina virtual con la IP 192.168.56.4, se instaló el servidor Apache para servir contenido web. Los pasos fueron los siguientes:

- Instalación de Apache:

```
sudo apt update
sudo apt install apache2 -y
```

- Cambiar el título de la página por defecto a "Apache server":

```
sudo nano /var/www/html/index.html
```

Luego, cambiar el contenido de la página HTML a:

```
<html>
  <head><title>Apache server</title></head>
  <body><h1>Welcome to Apache server</h1></body>
</html>
```

- Reiniciar el servicio de Apache:

```
sudo systemctl restart apache2
```

- Habilitar el servicio de `apache2` para que se inicie automáticamente al arrancar la máquina:

```
sudo systemctl enable apache2
```

Se configuró Apache para servir contenido estático desde el directorio `/var/www/html`, el cual puede ser accesible a través del proxy inverso configurado previamente.

5 Razones para el Cambio de Direcciones IP

En el ejercicio, las direcciones IP especificadas en el enunciado inicialmente no eran adecuadas debido a posibles conflictos con otras direcciones en el entorno de red de las máquinas virtuales y el dispositivo host. A continuación, se detallan las razones por las cuales se cambiaron las direcciones IP y cómo se solucionó este problema:

- **Conflictos con Direcciones Existentes:** Las direcciones IP proporcionadas en el enunciado original podrían haber coincidido con direcciones IP ya asignadas en la red interna de VirtualBox o con otras direcciones de dispositivos ya conectados al host. Este tipo de conflicto puede causar problemas de conectividad, ya que los paquetes de red pueden enviarse a la dirección incorrecta, afectando el funcionamiento del sistema distribuido.
- **Red "Host-only Network":** Al utilizar una red de tipo "Host-only Network" en VirtualBox, las direcciones IP deben estar dentro de un rango específico que no se solape con las direcciones asignadas automáticamente a otras interfaces de red del host o máquinas virtuales. Por lo tanto, se eligieron direcciones dentro de un rango que asegurara que no hubiera interferencia con la red del host o de otras máquinas virtuales.

- **Aislamiento de la Red Virtual:** Al asignar estas direcciones IP, también se logra el aislamiento entre las máquinas virtuales y otros dispositivos de la red externa, lo que mejora la seguridad y la estabilidad del sistema distribuido. Esto es especialmente importante cuando se prueban configuraciones de red en entornos controlados, ya que se minimiza la posibilidad de conflictos con la red local.

El cambio de las direcciones IP a un rango adecuado, junto con la asignación manual de estas direcciones a cada máquina virtual, permitió evitar conflictos y garantizar la correcta comunicación entre los distintos servicios del entorno distribuido.

6 Desafíos y Soluciones

Durante el desarrollo del ejercicio, se presentaron varios desafíos. Uno de los principales fue la configuración del proxy inverso, ya que las reglas de redirección de `nginx` no funcionaban correctamente debido a un error de configuración en las rutas. Este problema se resolvió ajustando las configuraciones del archivo `nginx.conf` y utilizando herramientas como `curl` y `telnet` para probar las conexiones.

6.1 Problema con el Arranque Seguro en VirtualBox y Solución mediante Firma

Durante la ejecución del ejercicio, se presentó un problema relacionado con el arranque seguro en VirtualBox, lo que impidió que las máquinas virtuales pudieran iniciarse correctamente. Este problema es común en sistemas operativos modernos, como Fedora, que tienen habilitada la opción de "Arranque Seguro" (Secure Boot) en la BIOS/UEFI, la cual verifica que solo se inicien sistemas operativos o software con una firma digital válida.

A continuación, se detallan las causas del problema y la solución adoptada:

- **Causa del Problema:** El arranque seguro en Fedora estaba impidiendo que VirtualBox ejecutara las máquinas virtuales, ya que las aplicaciones no firmadas no pueden ejecutarse bajo este modo de seguridad. VirtualBox no está firmado para ser utilizado en un entorno con

arranque seguro habilitado, lo que provocó que el sistema no pudiera iniciar las máquinas virtuales.

- **Solución: Firma de VirtualBox:** En lugar de deshabilitar completamente el arranque seguro, lo cual podría generar preocupaciones de seguridad, opté por crear una firma para VirtualBox. Esta solución permitió que las máquinas virtuales se ejecutaran sin necesidad de desactivar el arranque seguro. El proceso para firmar VirtualBox fue el siguiente:

- **Generar una clave de firma:** Utilicé los siguientes comandos para crear una clave privada y un certificado de firma:

```
sudo openssl req -new -newkey rsa:2048 -days 3650 -nodes  
-x509 -keyout /var/lib/shim-signed/mok.priv -out  
/var/lib/shim-signed/mok.der
```

Esto generó una clave privada ('mok.priv') y un certificado de firma ('mok.der') que se utilizarían para firmar los módulos de VirtualBox.

- **Registrar la clave en el sistema:** Después de crear la clave, registré la firma utilizando el siguiente comando:

```
sudo mokutil --import /var/lib/shim-signed/mok.der
```

Esto agregó el certificado de firma a la base de datos de claves de arranque seguro de Fedora.

- **Reiniciar y completar la firma:** Tras registrar la clave, reinicié el sistema y durante el arranque, Fedora solicitó que confirmara la importación de la clave, lo cual completé para permitir que los módulos de VirtualBox se ejecutaran sin restricciones.
- **Resultado:** Tras realizar este proceso de firma, las máquinas virtuales pudieron iniciarse correctamente con el arranque seguro habilitado, lo que permitió continuar con la configuración del entorno distribuido sin desactivar ninguna función de seguridad importante.

- **Consideraciones Adicionales:** Esta solución permitió mantener la protección de arranque seguro mientras habilitaba la ejecución de VirtualBox. Esta opción es más segura en comparación con la desactivación total del arranque seguro, ya que asegura que solo el software autorizado pueda ejecutarse en el sistema.

El problema con el arranque seguro se resolvió con éxito mediante la firma de VirtualBox, lo que permitió continuar con la configuración de las máquinas virtuales y el despliegue de los servicios requeridos en el ejercicio.

Otro desafío fue la comunicación entre las máquinas virtuales. Asegurarse de que las direcciones IP y las rutas de red estuvieran correctamente configuradas fue clave para garantizar que los servicios pudieran interactuar entre sí.

7 Recursos Utilizados

Para llevar a cabo la configuración y resolución de problemas se utilizaron diversos recursos, entre ellos:

- Documentación oficial de Nginx
- Documentación oficial de Docker
- Documentación oficial de Apache
- Pagina Oficial de Kali Linux
- Documentacion de VirtualBox

7.1 Video de muestra

- evidencia del funcionamiento del sistema

8 Conclusión

En este ejercicio se logró configurar un ambiente distribuido con un servidor proxy inverso, un aplicativo ejecutado en contenedores Docker y un servidor web Apache, garantizando la correcta interacción entre los servicios desplegados. Los desafíos fueron superados con éxito gracias al uso de herramientas de diagnóstico y pruebas.