Proceso de instalación Docker y

Configuración

MAQUINA VIRTUAL MINIOS-BASE DOCKER (0-4)

Creación de MV miniOS-docker para el proyecto final

Nombre: miniOS-baseD

Almacenamiento: 35GB

Redes: NAT, solo anfitrión y adaptador puente (los dos últimos en modo promiscuo: permitir todo)

IP: 192.168.56.202/24

Procesos de Instalación

1. Instalar miniOS al encender la MV

Para la instalación hay que irnos al menú del SO y buscamos ‘Install MiniOS’

Se nos mostrara un menú donde tenemos que configurar lo siguiente:

Select file system: ext4

EFI: esta opción tiene que estar marcada

Y le damos a ‘install’ luego nos pedirá reinciar

2. Configurar las Redes

Para la configuración de las redes, dirigirse a la parte inferior derecha y configurar las redes de NAT, solo anfitrión y adaptador puente.

eth0: Esta es la red NAT, la cual no debemos dejar en modo automático ya que es la que proporciona internet.

eth1: Esta es la red solo anfitrión. Vamos a la sección de configuración IPv4, establecemos el 'method' en manual y asignamos la IP 192.168.56.202/24.

eth2: Esta es la red adaptador puente, la dejamos como viene por defecto en esta instancia.

3. Instalar Docker

Para la instalación, ejecutamos el siguiente archivo que contiene los siguientes comandos:

```
sudo apt update

sudo apt install apt-transport-https ca-certificates curl software-properties-common -y

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

sudo apt install docker-ce -y

sudo apt install docker-compose-plugin -y

sudo usermod -aG docker ${USER}

su - live -c "docker info"
```

Después de tener este archivo con los códigos mencionados, ejecutamos el siguiente comando (asegurándonos de estar en la ruta correcta):

bash docker-install.sh

4. Actualizar el Sistema

Para actualizar el sistema, ejecutamos los siguientes comandos:

sudo apt update

sudo apt upgrade

5. Habilitar el SSH

Este comando se puede ejecutar en cualquier ruta:

sudo service ssh start

6. Verificar la Instalación de Docker

Para verificar si Docker se instaló correctamente, ejecutamos:

docker run hello-world

MAQUINA VIRTUAL MINIOS-DOCKER-DNS DOCKER (CLONADO DE BASE) (1-4)

Configuración de la Máquina Virtual

- **IP:** 192.168.56.203/24

Procesos de Instalación

1. Crear Estructura de Carpetas

Crear la carpeta **Docker** y dentro de ella, la carpeta **dns**.

2. Configurar las Redes

Para la configuración de las redes, sigue estos pasos:

- Ve a la parte inferior derecha y configura las redes de NAT y solo anfitrión.
    - **eth0**: Esta es la red NAT. No debe estar en modo automático ya que proporciona internet.
    - **eth1**: Esta es la red solo anfitrión. En la configuración de IPv4, establece el 'method' en manual y asigna la IP **192.168.56.252/24**.

Después de hacer estos cambios, puede ser necesario reiniciar la máquina virtual para que los cambios se apliquen.

3. Crear el Archivo Docker Compose en la Carpeta dns

Contenido del archivo **docker-compose.yml** para Pi-hole:

```
version: "3"


services:
  pihole:
    container_name: pihole
```

```
image: pihole/pihole:latest
ports:
  - "53:53/tcp"
  - "53:53/udp"
  - "67:67/udp" # Only required if you are using Pi-hole as your DHCP server
  - "80:80/tcp"
environment:
  TZ: 'America/La_Paz'
  WEBPASSWORD: '123123'
volumes:
  - './etc-pihole:/etc/pihole'
  - './etc-dnsmasq.d:/etc/dnsmasq.d'
cap_add:
  - NET_ADMIN # Required if you are using Pi-hole as your DHCP server,
                        else not needed
restart: unless-stopped
```

4. Limpiar el Caché para que Funcione Pi-hole

Deshabilitar el servicio **systemd-resolved**:

sudo systemctl disable systemd-resolved

Editar el archivo **/etc/NetworkManager/NetworkManager.conf** (puede hacerse de manera gráfica por el Snowflake para más rapidez). Dentro de ese archivo, cambiar lo siguiente:

dns=dnsmasq

por

dns=default

Reiniciar el sistema:

sudo shutdown -r now

5. Verificar Acceso a la Página de Administración de Pi-hole

La página es **http://192.168.56.10/admin** con contraseña: **123123**.

Dentro de esa página, ir a la sección de **Local DNS** y luego a **DNS Records**. Agregar:

- **Domain:** dns.leon.lan
- **IP Address:** 192.168.56.10

Luego, hacer clic en **ADD**.

Instalar las herramientas DNS:

sudo apt install dnsutils

Verificar la configuración con el comando:

dig @192.168.56.10 dns.leon.lan

6. Modificar la Configuración de Redes

Modificar la configuración de NAT en IPv4, en la sección de **Additional DNS servers**, agregar: **192.168.56.10**.

Puede haber errores, por lo que es necesario editar el archivo **/etc/resolv.conf**:

sudo nano /etc/resolv.conf

Agregar la siguiente línea:

nameserver 192.168.56.10

Para probar, abrir un navegador y acceder a la siguiente ruta:

http://192.168.56.10/admin

Si no funciona, intentar con:

http://dns.leon.lan/admin

MAQUINA VIRTUAL MINIOS-DOCKER-DOKUWIKI DOCKER (CLONADO DE BASE) (2-4)

Configuración de la Máquina Virtual

1. Crear Estructura de Carpetas

Crear la carpeta **docker** y dentro de ella, la carpeta **dokuwiki**.

2. Configurar las Redes

Para la configuración de las redes, sigue estos pasos:

- Ve a la parte inferior derecha y configura las redes de NAT y solo anfitrión.
    - **eth0**: Esta es la red NAT. No debe estar en modo automático ya que proporciona internet.
    - **eth1**: Esta es la red solo anfitrión. En la configuración de IPv4, establece el 'method' en manual y asigna la IP **192.168.56.250/24**.

Después de hacer estos cambios, puede ser necesario reiniciar la máquina virtual para que los cambios se apliquen.

3. Crear el Archivo Docker Compose

Contenido del archivo **docker-compose.yml** para DokuWiki:

```
version: "2.1"
services:
  dokuwiki:
    image: lscr.io/linuxserver/dokuwiki:latest
    container_name: dokuwiki
    environment:
      - PUID=1000
```

```
    - PGID=1000
    - TZ=Bolivia/La_Paz
  volumes:
    - ./config:/config
  ports:
    - 80:80
    - 443:443 #optional
  restart: unless-stopped
```

- **Prender el miniOS-docker-base y también el miniOS-docker-dns.**

La idea es que como ya configuramos el DNS con el dominio y todo en la MV docker base, tiene que entrar. Nos dirigimos al navegador y probamos primero con la IP:

http://192.168.56.10/admin

Esa ruta debería funcionar, pero al probar con el dominio **dns.leon.lan**, es posible que no lo encuentre. Para resolver esto, edita el archivo **/etc/resolv.conf** en la MV docker-base:

sudo nano /etc/resolv.conf

Modifica el contenido de la siguiente manera:

#nameserver 1.1.1.1

#nameserver 8.8.8.8

nameserver 192.168.56.10

Comentamos los dos primeros y dejamos el que nos importa de la VM docker-dns. Luego, en el navegador ponemos:

http://dns.leon.lan/admin

Migrar el Data DokuWiki a la MV MINIOS-DOCKER-DUKUWIKI (CLONADO)

Preparativos

Prender las máquinas de **miniOS-docker-dns**, **miniOS-docker** y la MV clonada de **miniOS-docker-dokuwiki**.

En la Máquina MINIOS-DOCKER-DNS

1. Crear el dominio **dk.leon.lan**. Acceder a la página:

   http://192.168.56.10 o dns.leon.lan/admin

Dentro de la página, ir a la sección de **Local DNS** y crear el dominio:

dk.leon.lan

En la Máquina MINIOS-DOCKER-DOKUWIKI

2. Crear la estructura de carpetas:

mkdir -p docker/dokuwiki

Descargar el archivo **docker-compose-dokuwiki.yml** del repositorio del docente y moverlo a la carpeta **dokuwiki**. Luego, ejecutar el comando para activar el contenedor:

docker compose up -d

Acceder a la ruta:

http://192.168.56.250/install.php

Completar con:

- **Usuario:** jhonnydleon
- **Contraseña:** 8446

En la Máquina MINIOS-DOCKER

3. Navegar a la ruta:

cd "/home/live/docker/dokuwiki/config/dokuwiki"

Comprimir la carpeta **data**:

sudo tar -czvf data.tar.gz data

Usar Snowflake para transferir el archivo comprimido a la carpeta:

/home/live/docker/dokuwiki/config/dokuwiki

de la MV **MINIOS-DOCKER-DOKUWIKI**.

En la Máquina MINIOS-DOCKER-DOKUWIKI

4. Navegar a la ruta y descomprimir el archivo:

cd "/home/live/docker/dokuwiki/config/dokuwiki"

tar -xzvf /home/live/docker/dokuwiki/config/data.tar.gz -C
/home/live/docker/dokuwiki/config/dokuwiki/

Modificar el Archivo resolv.conf

5. Editar el archivo **resolv.conf**:

sudo nano /etc/resolv.conf

Dejarlo de la siguiente manera:

#nameserver 1.1.1.1

#nameserver 8.8.8.8

nameserver 192.168.56.10

Reiniciar el sistema y acceder a la ruta:

http://dk.leon.lan/

Si no carga el DokuWiki con la migración, probar de nuevo después de reiniciar la MV.

MAQUINA VIRTUAL MINIOS-DOCKER-CMS CON FORMS DE PHP

DOCKER (CLONADO DE BASE) (3-4)

Características

- **Clonación del miniOS-docker-base**: hay que clonarlo con los cambios.
    - **Nombre**: minios-docker-cms
    - **IP**: 192.168.56.251
    - **Dominio**: cms.leon.lan
    - **Archivo a utilizar**: docker-compose.php.yml del ingeniero en su repositorio, renombrarlo a docker-compose.yml y modificar el puerto a 8080.

Configuración Inicial

1. Crear Estructuras de Carpetas

Crear la carpeta **docker** y dentro de ella las carpetas **cms** y **php**.

2. Configurar las Redes

Para la configuración de las redes, sigue estos pasos:

- Ve a la parte inferior derecha y configura las redes de NAT y solo anfitrión.
    - **eth0**: Esta es la red NAT. No debe estar en modo automático ya que proporciona internet.
    - **eth1**: Esta es la red solo anfitrión. En la configuración de IPv4, establece el 'method' en manual y asigna la IP **192.168.56.251/24**.

Después de hacer estos cambios, puede ser necesario reiniciar la máquina virtual para que los cambios se apliquen.

Instalar CMS-Simple

2. Crear el archivo docker-compose.yml en la carpeta cms

Contenido del archivo **docker-compose.yml**:

```
version: '3.8'


services:
  # Apache and PHP service
  web:
    image: php:8-apache
    container_name: php-apache-container
    ports:
      - "80:80"
    volumes:
      - ./html:/var/www/html
# Define networks
networks:
  my-network:
```

3. Ejecutar el Servicio

```
docker compose up -d
```

4. Clonar el CMS en la Carpeta html

```
sudo git clone https://github.com/risingisland/GetSimpleCMS-CE-3.3.20.git
```

5. Configuración Adicional

live@minios:~/docker/cms/html$ sudo mv GetSimpleCMS-CE-3.3.20/* .

live@minios:~/docker/cms/html$ cd ..

live@minios:~/docker/cms$ sudo chown -R live:live html/

6. Acceder al Navegador

http://cms.leon.lan/admin

7. Dar Permisos a las Carpetas de data y backups

live@minios:~/docker/cms$ sudo chmod -R 777 html/

8. Cambiar la Contraseña

Al acceder a la página del CMS, se pedirá que cambies la contraseña.

Instalar PHP-Forms

1. Navegar a la Carpeta de php

cd docker/php

2. Crear el archivo docker-compose.yml para PHP

Contenido del archivo **docker-compose.yml**:

```
version: '3.8'

services:

  # Apache and PHP service

  web:

    image: php:8-apache

    container_name: php-forms-container

    ports:

      - "8080:80"

    volumes:

      - ./html:/var/www/html

# Define networks

networks:

  my-network:
```

3. Levantar el Servicio

docker compose up -d

4. Entrar a la Carpeta html que se Crea al Levantar el Servicio

cd docker/php/html

5. Clonar el Simple-CMS

sudo git clone https://github.com/gnat/simple-php-form.git

6. Renombrar la Carpeta de la Clonación

Cambiar el nombre a **frm**.

7. Dar Permisos a la Carpeta de frm

cd docker/php

sudo chown -R live:live html/

8. Verificar la Página

Para ver los forms, poner la ruta completa más el puerto:

http://cms.leon.lan:8080/frm/examples/basic.php

MAQUINA VIRTUAL MINIOS-DOCKER-WORDPRESS CON PHPMYADMIN Y MARIADB (CLONADO DE BASE) (4-4)

1. Crear Estructuras de Carpetas

Crear la carpeta **docker** y dentro de ella la carpeta **wordpress**.

2. Configurar las Redes

Para la configuración de las redes, sigue estos pasos:

- Ve a la parte inferior derecha y configura las redes de NAT y solo anfitrión.
    - **eth0**: Esta es la red NAT. No debe estar en modo automático ya que proporciona internet.
    - **eth1**: Esta es la red solo anfitrión. En la configuración de IPv4, establece el 'method' en manual y asigna la IP **192.168.56.252/24**.

Después de hacer estos cambios, puede ser necesario reiniciar la máquina virtual para que los cambios se apliquen.

3. Crear el archivo docker-compose.yml en la carpeta wordpress

Contenido del archivo **docker-compose.yml**:

```
# Versión de Docker Compose
# https://docs.docker.com/compose/compose-file/compose-file-v3/
version: '3'

# Inicio de los servicios
services:

  # Define el servicio 'wordpress'.
  wordpress:
    # Usa la imagen oficial de WordPress en Docker.
    image: wordpress
    # Expone el puerto 8080 para la comunicación dentro de la red.
    expose:
      - 80
    ports:
      # Mapea el puerto 8080 del host al puerto 80 del contenedor para acceso externo.
      - 80:80
    networks:
      # Conéctate a la red 'internal' para la comunicación.
      - internal
    # Configura las variables de entorno para el servicio de WordPress.
    environment:
      # Configura el host de la base de datos.
      WORDPRESS_DB_HOST: mariadb
      # Configura el nombre de la base de datos.
      WORDPRESS_DB_NAME: wp_db_001
      # Configura el usuario y la contraseña de la base de datos.
      WORDPRESS_DB_USER: jhonny
      WORDPRESS_DB_PASSWORD: 123123 # cambia esto!
    volumes:
      # Monta un directorio local como volumen en el contenedor.
      - ./wordpress:/var/www/html
```

```
  # Define el servicio de MariaDB.
  mariadb:
    # Usa la imagen oficial de MariaDB en Docker.
    image: mariadb
    expose:
      # Expone el puerto 3306 para la comunicación dentro de la red.
      - 3306
    ports:
      # Mapea el puerto 3306 del host al puerto 3306 del contenedor para acceso externo.
      - 3306:3306
    networks:
      # Conéctate a la red 'internal' para la comunicación.
      - internal
    # Configura las variables de entorno para el servicio de MariaDB.
    environment:
      # Crea la base de datos de WordPress.
      MYSQL_DATABASE: wp_db_001
      # Crea el usuario y la contraseña de WordPress.
      MYSQL_USER: jhonny
      MYSQL_PASSWORD: 123123 # cambia esto!
      # Configura la contraseña de root de la base de datos.
      MYSQL_ROOT_PASSWORD: 123123 # cambia esto!
      # Configura MARIADB_MYSQL_LOCALHOST_USER con un valor no vacío para crear el usuario mysql@localhost.
      # Este usuario es especialmente útil para una variedad de comprobaciones de estado y scripts de respaldo.
      # https://hub.docker.com/_/mariadb
      # MARIADB_MYSQL_LOCALHOST_USER: true
    volumes:
      # Monta un directorio local como volumen en el contenedor.
      - ./db:/var/lib/mysql


  # Define el servicio 'pma' para phpMyAdmin.
  pma:
    # Usa la imagen oficial de phpMyAdmin en Docker.
    image: phpmyadmin
    # Asegura que el contenedor se reinicie automáticamente si se detiene.
```

```
    restart: always
    # Expone el puerto 8181 para la comunicación dentro de la red.
    expose:
      - 8181
    # Mapea el puerto 8181 del host al puerto 80 del contenedor para acceso externo.
    ports:
      - 8181:80
    # Conéctate a la red 'internal' para la comunicación.
    networks:
      - internal
    # Configura las variables de entorno para el servicio de phpMyAdmin.
    environment:
      # Desactiva la conexión arbitraria al servidor; usa el host y el puerto configurados.
      - PMA_ARBITRARY=0
      # Configura el host al que phpMyAdmin debe conectarse.
      - PMA_HOST=mariadb
      # Configura el puerto al que phpMyAdmin debe conectarse.
      - PMA_PORT=3306

# Define la red 'internal' para la comunicación entre contenedores.
networks:
  internal:

# Define volúmenes con nombre para el almacenamiento de datos.
volumes:
  wordpress:
  db:
```

4. Levantar el Servicio

docker compose up -d

5. Verificar las Páginas

Para WordPress

http://wp.leon.lan/wp-admin/install.php

Después de instalarlo la primera vez, podrás entrar de la siguiente manera:

http://wp.leon.lan

Para phpMyAdmin

http://wp.leon.lan:8181/admin

En este caso, se usa el puerto 8181 ya que el puerto por defecto está siendo utilizado por WordPress.