

## Ejercicio 1: Instalación de Docker en Ubuntu 24.04

**Objetivo:** Instalar Docker Engine en Ubuntu 24.04 siguiendo el procedimiento oficial del fabricante.

### Pasos:

**1. Actualizar el índice de paquetes:**

```
sudo apt update
```

**2. Instalar los paquetes necesarios para permitir que apt use repositorios sobre HTTPS:**

```
sudo apt install ca-certificates curl gnupg
```

**3. Crear el directorio para las claves GPG:**

```
sudo install -m 0755 -d /etc/apt/keyrings
```

**4. Descargar la clave GPG oficial de Docker:**

```
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --  
dearmor -o /etc/apt/keyrings/docker.gpg
```

**5. Dar permisos de lectura a la clave GPG para el usuario actual:**

```
sudo chmod a+r /etc/apt/keyrings/docker.gpg
```

**6. Añadir el repositorio de Docker a las fuentes de APT:**

```
echo \  
"deb [arch="$(dpkg --print-architecture)"  
signed-by=/etc/apt/keyrings/docker.gpg]  
https://download.docker.com/linux/ubuntu \  
"$(. /etc/os-release && echo "$VERSION_CODENAME)" stable" | \  
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

**7. Actualizar el índice de paquetes nuevamente con el repositorio de Docker añadido:**

```
sudo apt update
```

**8. Instalar Docker Engine, Docker CLI y Containerd:**

```
sudo apt install docker-ce docker-ce-cli containerd.io docker-buildx-  
plugin docker-compose-plugin
```

**9. Verificar que el servicio Docker esté activo y funcionando:**

```
sudo systemctl status docker
```

**10.(Opcional) Permitir a un usuario no-root ejecutar comandos Docker:**

- Crear el grupo docker si no existe: `sudo groupadd docker`
- Añadir tu usuario al grupo docker: `sudo usermod -aG docker $USER`
- Cerrar la sesión y volver a iniciarla para que los cambios surtan efecto.

---

**Ejercicio 2: Instalación de Docker en Rocky Linux 9**

**Objetivo:** Instalar Docker Engine en Rocky Linux 9 siguiendo el procedimiento oficial del fabricante.

**Pasos:**

**1. Actualizar el índice de paquetes:**

```
sudo dnf update -y
```

**2. Instalar las utilidades DNF (necesarias para config-manager):**

```
sudo dnf install -y dnf-utils
```

**3. Añadir el repositorio oficial de Docker:**

```
sudo dnf config-manager --add-repo  
https://download.docker.com/linux/rhel/docker-ce.repo
```

**4. Instalar Docker Engine, Docker CLI, Containerd y plugins:**

```
sudo dnf install docker-ce docker-ce-cli containerd.io docker-buildx-  
plugin docker-compose-plugin -y
```

**5. Iniciar el servicio Docker:**

```
sudo systemctl start docker
```

**6. Habilitar el servicio Docker para que se inicie automáticamente en el arranque:**

```
sudo systemctl enable docker
```

**7. Verificar que el servicio Docker esté activo y funcionando:**

```
sudo systemctl status docker
```

**8. (Opcional) Permitir a un usuario no-root ejecutar comandos Docker:**

- Crear el grupo `docker` si no existe: `sudo groupadd docker`
- Añadir tu usuario al grupo `docker`: `sudo usermod -aG docker $USER`
- Cerrar la sesión y volver a iniciarla para que los cambios surtan efecto.

---

### Ejercicio 3: Ejecución de un Contenedor Nginx con Docker

**Objetivo:** Descargar y ejecutar una imagen de Nginx como un contenedor Docker.

**Pasos:**

**1. Descargar la imagen oficial de Nginx desde Docker Hub:**

```
docker pull nginx
```

*(Si no añadiste tu usuario al grupo `docker` en los ejercicios anteriores, deberás usar `sudo docker pull nginx`)*

**2. Ejecutar un contenedor basado en la imagen de Nginx, mapeando el puerto 80 del host al puerto 80 del contenedor:**

```
docker run --name my-nginx -p 80:80 -d nginx
```

- `--name my-nginx`: Asigna el nombre "my-nginx" al contenedor para facilitar su identificación.
- `-p 80:80`: Mapea el puerto 80 del host al puerto 80 dentro del contenedor.
- `-d`: Ejecuta el contenedor en modo "detached" (en segundo plano).
- `nginx`: Especifica la imagen a utilizar. *(De nuevo, usa `sudo docker run ...` si es necesario)*

**3. Verificar que el contenedor se está ejecutando:**

```
docker ps
```

Deberías ver un contenedor con el nombre `my-nginx` y el estado "Up".

**4. Verificar que se puede acceder a la página por defecto de Nginx desde un navegador web o usando `curl` en el host:**

```
curl http://localhost
```

Deberías recibir el código HTML de la página de bienvenida de Nginx.

**5. Detener y eliminar el contenedor (opcional):**

```
docker stop my-nginx
docker rm my-nginx
```

*(Usa `sudo` si es necesario)*

---

**Ejercicio 4: Instalación de Podman**

**Objetivo:** Instalar Podman en Ubuntu 24.04 y Rocky Linux 9.

**Instalación en Ubuntu 24.04:**

Podman está disponible en los repositorios por defecto de Ubuntu.

**1. Actualizar el índice de paquetes:**

```
sudo apt update
```

**2. Instalar Podman:**

```
sudo apt install podman -y
```

**3. Verificar la instalación:**

```
podman --version
```

**Instalación en Rocky Linux 9:**

Podman también está disponible en los repositorios por defecto de Rocky Linux.

**1. Actualizar el índice de paquetes:**

```
sudo dnf update -y
```

**2. Instalar Podman:**

```
sudo dnf install podman -y
```

**3. Verificar la instalación:**

```
podman --version
```

---

## Ejercicio 5: Instalación de LXC

**Objetivo:** Instalar LXC en Ubuntu 24.04 y Rocky Linux 9.

### Instalación en Ubuntu 24.04:

LXC está disponible en los repositorios por defecto de Ubuntu.

#### 1. Actualizar el índice de paquetes:

```
sudo apt update
```

#### 2. Instalar LXC:

```
sudo apt install lxc -y
```

#### 3. Verificar la instalación:

```
lxc --version
```

*(Nota: Es posible que `lxc-create`, `lxc-start`, etc., sean los comandos principales dependiendo de la versión y configuración, pero `lxc --version` debería mostrar la versión de las herramientas instaladas).*

### Instalación en Rocky Linux 9:

La instalación de LXC en Rocky Linux generalmente implica el uso de Snap, ya que no siempre está disponible en los repositorios por defecto o su versión puede estar desactualizada.

#### 1. Asegurar SELinux en modo permisivo (puede ser necesario para Snap y LXC):

```
sudo sed -i 's/^SELINUX=enforcing$/SELINUX=permissive/'  
/etc/selinux/config  
sudo reboot # Se requiere reiniciar para aplicar el cambio de SELinux
```

Después de reiniciar, verificar con `getenforce` que esté en modo Permissive.

#### 2. Instalar el repositorio EPEL (si no está ya instalado):

```
sudo dnf install epel-release -y
```

#### 3. Instalar Snapd:

```
sudo dnf install snapd -y
```

#### 4. Habilitar el socket de Snapd:

```
sudo systemctl enable --now snapd.socket
```

**5. Crear el enlace simbólico necesario para el soporte clásico de Snap:**

```
sudo ln -s /var/lib/snapd/snap /snap
```

**6. Reiniciar el sistema nuevamente para que los cambios de Snap surtan efecto:**

```
sudo reboot
```

Después de reiniciar, asegúrate de que el servicio snapd está activo (`sudo systemctl status snapd`).

**7. Instalar LXD (que incluye las herramientas LXC) usando Snap:**

```
sudo snap install lxd
```

**8. Inicializar LXD (configuración inicial):**

```
sudo lxd init
```

Sigue las indicaciones. Para una configuración básica de prueba, generalmente puedes aceptar los valores por defecto.

**9. Verificar la instalación de LXD y las herramientas LXC:**

```
lxc version
```

Deberías ver información sobre la versión del cliente y el servidor LXD.

---

## Ejercicio 6: Ejecución de un Contenedor Nginx con Podman

**Objetivo:** Descargar y ejecutar una imagen de Nginx como un contenedor Podman.

**Pasos:**

**1. Descargar la imagen oficial de Nginx desde Docker Hub (Podman puede usar imágenes de Docker Hub directamente):**

```
podman pull nginx
```

**2. Ejecutar un contenedor basado en la imagen de Nginx, mapeando el puerto 80 del host al puerto 80 del contenedor:**

```
podman run --name my-nginx-podman -p 80:80 -d nginx
```

- `--name my-nginx-podman`: Asigna un nombre al contenedor.
- `-p 80:80`: Mapea el puerto 80 del host al puerto 80 dentro del contenedor.
- `-d`: Ejecuta el contenedor en segundo plano.
- `nginx`: Especifica la imagen a utilizar.

**3. Verificar que el contenedor se está ejecutando:**

```
podman ps
```

Deberías ver un contenedor con el nombre `my-nginx-podman` y el estado "Up".

**4. Verificar que se puede acceder a la página por defecto de Nginx desde un navegador web o usando `curl` en el host:**

```
curl http://localhost
```

Deberías recibir el código HTML de la página de bienvenida de Nginx.

**5. Detener y eliminar el contenedor (opcional):**

```
podman stop my-nginx-podman  
podman rm my-nginx-podman
```

---

## Ejercicio 7: Ejecución de un Contenedor Nginx con LXC

**Objetivo:** Crear y ejecutar un contenedor LXC con Nginx instalado.

**Pasos:**

**1. Crear un nuevo contenedor LXC (por ejemplo, basado en Ubuntu):**

```
sudo lxc init ubuntu my-nginx-lxc
```

- `ubuntu`: Especifica la imagen base (una imagen de Ubuntu en este caso).
- `my-nginx-lxc`: Asigna un nombre al contenedor.

**2. Iniciar el contenedor:**

```
sudo lxc start my-nginx-lxc
```

**3. Verificar que el contenedor se está ejecutando:**

```
sudo lxc list
```

Deberías ver el contenedor `my-nginx-lxc` con el estado `RUNNING`.

**4. Acceder al shell del contenedor:**

```
sudo lxc exec my-nginx-lxc -- /bin/bash
```

**5. Dentro del contenedor, actualizar el índice de paquetes e instalar Nginx:**

```
apt update
apt install nginx -y
exit # Salir del shell del contenedor
```

**6. Obtener la dirección IP del contenedor:**

```
sudo lxc list my-nginx-lxc
```

Busca la dirección IP asignada en la salida.

**7. Verificar que se puede acceder a la página por defecto de Nginx utilizando la IP del contenedor desde el host:**

```
curl http://[DIRECCIÓN_IP_DEL_CONTENEDOR]
```

Reemplaza `[DIRECCIÓN_IP_DEL_CONTENEDOR]` con la IP que obtuviste en el paso anterior. Deberías recibir el código HTML de la página de bienvenida de Nginx.

**8. Detener y eliminar el contenedor (opcional):**

```
sudo lxc stop my-nginx-lxc
sudo lxc delete my-nginx-lxc
```

Estos ejercicios cubren la instalación básica y la ejecución de contenedores web simples utilizando las tres tecnologías principales mencionadas en el objetivo 102.6, proporcionando experiencia práctica con Linux en entornos contenerizados.