INFORME – ACTIVIDAD 4: Instalación y configuración de Docker Engine (Debian)

4.1 INSTALACIÓN DE DOCKER ENGINE EN LINUX (DEBIAN)

En esta sección se documenta el proceso de instalación del Docker Engine nativo en Debian, cumpliendo los requerimientos de la guía.

1.1 Actualizar el sistema

```
vboxuser@DEBIAN:~$ sudo apt update
[sudo] contraseña para vboxuser:
Obj:1 http://deb.debian.org/debian trixie InRelease
Obj:2 http://deb.debian.org/debian trixie-updates InRelease
Obj:3 https://download.docker.com/linux/debian trixie InRelease
Obj:4 http://security.debian.org/debian-security trixie-security InRelease
Todos los paquetes están actualizados.
vboxuser@DEBIAN:~$ sudo apt upgrade -y
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
vboxuser@DEBIAN:~$
```

1.2 Instalar dependencias necesarias

- ca-certificates: Permite validar certificados HTTPS.
- curl: Descarga archivos desde la terminal.
- gnupg: Gestiona claves GPG necesarias para firmar repositorios

```
vboxuser@DEBIAN:~$ sudo apt install -y ca-certificates curl gnupg lsb-release
ca-certificates ya está en su versión más reciente (20250419).
curl ya está en su versión más reciente (8.14.1-2+deb13u2).
gnupg ya está en su versión más reciente (2.4.7-21).
lsb-release ya está en su versión más reciente (12.1-1).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

1.3 Agregar el repositorio oficial de Docker

- Crea el directorio donde se almacenarán las claves GPG del repositorio.
- Descarga la clave oficial de Docker y la convierte al formato que usa APT.
- Asigna permisos para que APT pueda leer la clave.

```
vboxuser@DEBIAN:~$ sudo install -m 0755 -d /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
sudo chmod a+r /etc/apt/keyrings/docker.gpg
El fichero '/etc/apt/keyrings/docker.gpg' ya existe. ¿Sobreescribir? (s/N) s
```

1.4 Agregar el repositorio

- Añade el repositorio estable de Docker según la versión de Debian instalada.

```
vboxuser@DEBIAN:~$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] \
  https://download.docker.com/linux/debian \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

1.5 Actualizar nuevamente

```
vboxuser@DEBIAN:~$ sudo apt update
Obj:1 http://deb.debian.org/debian trixie InRelease
Obj:2 http://deb.debian.org/debian trixie-updates InRelease
Obj:3 http://security.debian.org/debian-security trixie-security InRelease
Obj:4 https://download.docker.com/linux/debian trixie InRelease
Todos los paquetes están actualizados.
```

1.6 Instalar Docker y sus componentes

- docker-ce: Motor de Docker.
- docker-ce-cli: Herramientas de comando.
- containerd.io: Motor de contenedores en segundo plano.
- buildx: Permite construir imágenes avanzadas.
- docker compose plugin: Permite usar docker compose.

```
vboxuser@DEBIAN:~$ sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
docker-ce ya está en su versión más reciente (5:29.1.2-1~debian.13~trixie).
docker-ce-cli ya está en su versión más reciente (5:29.1.2-1~debian.13~trixie).
containerd.io ya está en su versión más reciente (2.2.0-2~debian.13~trixie).
docker-buildx-plugin ya está en su versión más reciente (0.30.1-1~debian.13~trixie).
docker-compose-plugin ya está en su versión más reciente (5.0.0-1~debian.13~trixie).
Summary:
  Upgrading: 0, Installing: 0, Removing: 0, Not Upgrading: 0
```

```
vboxuser@DEBIAN:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Thu 2025-12-11 18:06:31 -05; 4min 36s ago
 Invocation: fb12d3f5732c4581a447d6aa455d9626
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 6781 (dockerd)
      Tasks: 11
     Memory: 28.1M (peak: 28.9M)
        CPU: 1.775s
     CGroup: /system.slice/docker.service
             └─6781 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

dic 11 18:06:29 DEBIAN dockerd[6781]: time="2025-12-11T18:06:29.718445142-05:00" level=info msg="Restoring containers: start."
dic 11 18:06:29 DEBIAN dockerd[6781]: time="2025-12-11T18:06:29.876010664-05:00" level=info msg="Deleting nftables IPv4 rules" error="exit >
dic 11 18:06:29 DEBIAN dockerd[6781]: time="2025-12-11T18:06:29.918073424-05:00" level=info msg="Deleting nftables IPv6 rules" error="exit >
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.371476923-05:00" level=info msg="Loading containers: done."
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.402967284-05:00" level=info msg="Docker daemon" commit=de45c2a containerd-s>
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.403198949-05:00" level=info msg="Initializing buildkit"
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.542845816-05:00" level=info msg="Completed buildkit initialization"
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.578332738-05:00" level=info msg="Daemon has completed initialization"
dic 11 18:06:31 DEBIAN systemd[1]: Started docker.service - Docker Application Container Engine.
dic 11 18:06:31 DEBIAN dockerd[6781]: time="2025-12-11T18:06:31.578864905-05:00" level=info msg="API listen on /run/docker.sock"
```

1.7 Verificar instalación

- Ejecuta una imagen de prueba para confirmar que Docker funciona correctamente.

```
vboxuser@DEBIAN:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
ea52d2000f90: Download complete
Digest: sha256:d4aaab6242e0cace87e2ec17a2ed3d779d18fbfd03042ea58f2995626396a274
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

4.2 CREACIÓN DE IMAGEN DOCKER PERSONALIZADA (DOCKERFILE)

1. Crear directorio del proyecto

- Crea la carpeta donde estará tu Dockerfile.
- Entra en dicha carpeta.

```
vboxuser@DEBIAN:~$ mkdir miapp
vboxuser@DEBIAN:~$ cd miapp
vboxuser@DEBIAN:~/miapp$
```

2. Crear el Dockerfile

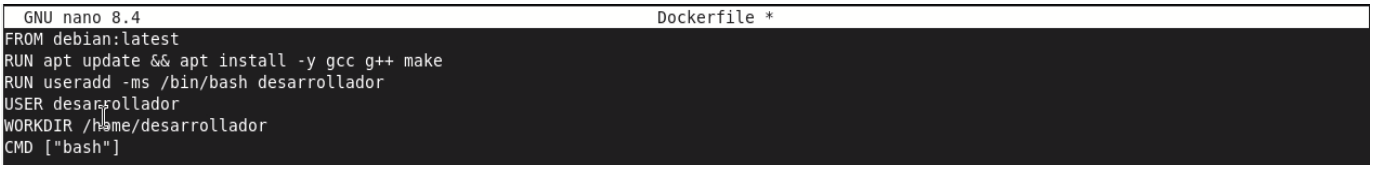
- Abre el editor nano para crear el archivo Dockerfile.

```
vboxuser@DEBIAN:~/miapp$ nano Dockerfile
```

3. Contenido del Dockerfile (explicado brevemente)

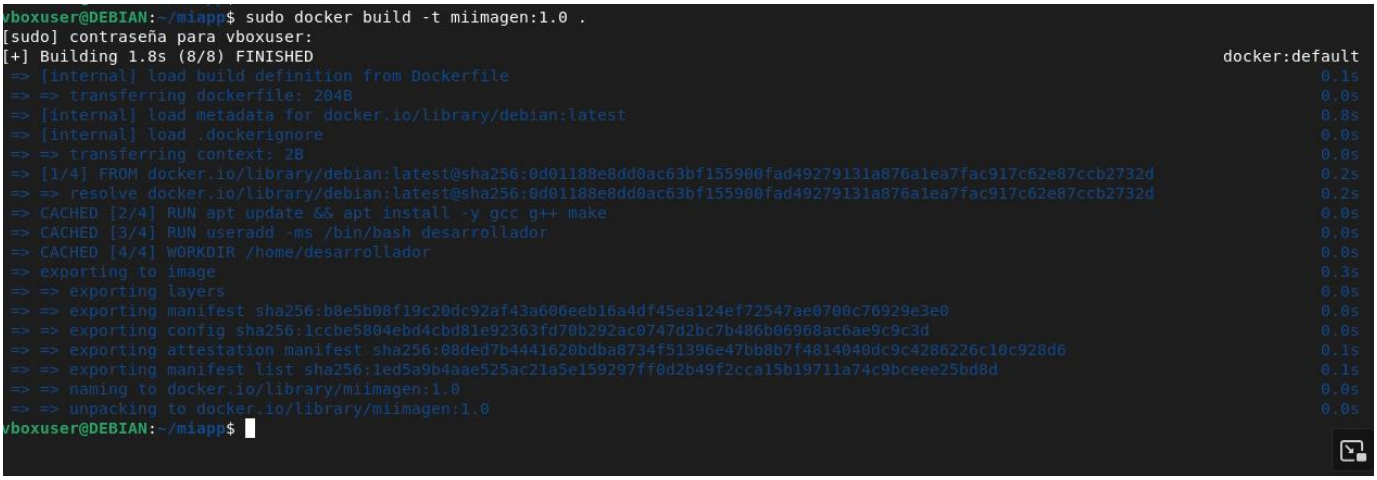
- Usa Debian como imagen base.

- Instala herramientas de compilación (build-essential).
- Crea un usuario sin privilegios.
- Define que el contenedor se ejecute con este usuario.
- Establece el directorio de trabajo.
- Ejecuta una terminal Bash al iniciar el contenedor.



4. Construir la imagen

- Construye la imagen con nombre miimagen y versión 1.0
- El punto (.) significa: usa el Dockerfile en este directorio.



4.3 CREACIÓN Y GESTIÓN DE CONTENEDORES — (VERSIÓN DEBIAN CON DESCRIPCIONES)

1. Crear y ejecutar un contenedor interactivo

- docker run → crea y ejecuta un contenedor.
- -i → modo interactivo (recibe entrada de teclado).
- -t → asigna una terminal tipo TTY.
- --name → asigna un nombre personalizado.
- miimagen:1.0 → imagen base que tú creaste.

- **Ctrl + P + Q** → desconecta la terminal, pero el contenedor sigue en ejecución.
- **exit** → cierra la sesión y detiene el contenedor.

```
vboxuser@DEBIAN:~$ sudo docker run -it --name contenedor_prueba miimagen:1.0 bash
[sudo] contraseña para vboxuser:
desarrollador@a5a80aafda1c:~$ vboxuser@DEBIAN:~$
```

2. Listar contenedores en ejecución

- Muestra solo los contenedores que están activos.

```
desarrollador@a1d7e5decf2a:~$ vboxuser@DEBIAN:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED              STATUS                  PORTS     NAMES
a1d7e5decf2a   miimagen:1.0   "bash"     14 seconds ago       Up 13 seconds                     contenedor_prueba1
a5a80aafda1c   miimagen:1.0   "bash"     About a minute ago   Up About a minute                 contenedor_prueba
f6b0509e3731   hello-world    "/hello"   2 hours ago          Exited (0) 2 hours ago            great_dhawan
```

3. Volver a entrar al contenedor

- docker exec → ejecuta un comando dentro de un contenedor ya creado.
- -it → te permite entrar como si fuera una terminal real.

```
vboxuser@DEBIAN:~$ sudo docker exec -it contenedor_prueba bash
desarrollador@a5a80aafda1c:~$ read escape sequence
```

4. Listar contenedores en ejecución

- Muestra solo los contenedores que están activos.

```
vboxuser@DEBIAN:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED              STATUS                  PORTS     NAMES
a1d7e5decf2a   miimagen:1.0   "bash"     About a minute ago   Up About a minute                 contenedor_prueba1
a5a80aafda1c   miimagen:1.0   "bash"     2 minutes ago        Up 2 minutes                      contenedor_prueba
f6b0509e3731   hello-world    "/hello"   2 hours ago          Exited (0) 2 hours ago            great_dhawan
```

5. Detener un contenedor

- Apaga correctamente un contenedor en ejecución.

```
vboxuser@DEBIAN:~$ sudo docker stop contenedor_prueba
contenedor_prueba
```

6. Listar contenedores en ejecución

- Muestra solo los contenedores que están activos

```
vboxuser@DEBIAN:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED         STATUS                          PORTS     NAMES
a1d7e5decf2a   miimagen:1.0   "bash"     3 minutes ago   Up 3 minutes                              contenedor_prueba1
a5a80aafda1c   miimagen:1.0   "bash"     4 minutes ago   Exited (137) 16 seconds ago               contenedor_prueba
f6b0509e3731   hello-world    "/hello"   2 hours ago     Exited (0) 2 hours ago                    great_dhawan
```

7. Iniciar un contenedor detenido

- Arranca un contenedor que se encontraba detenido.

```
vboxuser@DEBIAN:~$ sudo docker start contenedor_prueba
contenedor_prueba
```

8. Listar contenedores en ejecución

- Muestra solo los contenedores que están activos.

```
vboxuser@DEBIAN:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED         STATUS                     PORTS     NAMES
a1d7e5decf2a   miimagen:1.0   "bash"     4 minutes ago   Up 4 minutes                         contenedor_prueba1
a5a80aafda1c   miimagen:1.0   "bash"     5 minutes ago   Up 16 seconds                        contenedor_prueba
f6b0509e3731   hello-world    "/hello"   2 hours ago     Exited (0) 2 hours ago               great_dhawan
```

9. Detener un contenedor

- Apaga correctamente un contenedor en ejecución.

```
vboxuser@DEBIAN:~$ sudo docker stop contenedor_prueba1
contenedor_prueba1
```

10. Eliminar un contenedor

- docker rm → borra un contenedor.
- Solo funciona si está detenido.

```
vboxuser@DEBIAN:~$ sudo docker rm contenedor_prueba1
contenedor_prueba1
```

11. Listar contenedores en ejecución

- Muestra solo los contenedores que están activos.

```
vboxuser@DEBIAN:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND    CREATED         STATUS                     PORTS     NAMES
a5a80aafda1c   miimagen:1.0   "bash"     8 minutes ago   Up 3 minutes                         contenedor_prueba
f6b0509e3731   hello-world    "/hello"   2 hours ago     Exited (0) 2 hours ago               great_dhawan
```

12. Listar imágenes disponibles

- Muestra todas las imágenes instaladas en tu sistema.

```
vboxuser@DEBIAN:~$ sudo docker images
                                                                                    i Info →  U In Use
IMAGE                ID             DISK USAGE   CONTENT SIZE   EXTRA
hello-world:latest   d4aaab6242e0       25.9kB         9.52kB    U
miimage:1.0          792561e5975c        620MB          171MB    U
miimagen:1.0         1ed5a9b4aae5        620MB          171MB    U
vboxuser@DEBIAN:~$ sudo docker images
```