

Linux + IA 2025/05/25

Docker

- Script de instalación:

```
◦ wget -C  
https://nextcloud.openit.dev/s/6rBskz35qHgkDAN/download/docker-  
install.sh  
chmod u+x docker-install.sh  
sudo ./docker-install.sh
```

- Dejar descargando el image OpenWebUI

```
◦ docker pull openwebui/open-webui:latest
```

Docker Model Runner (DMR)

- Instalación Ubuntu/Debian

```
◦ sudo apt-get update  
sudo apt-get install docker-model-plugin
```

- Verifica la instalación:

```
◦ docker model version
```

Maquina virtual o sin GPU

- Para asegurar que DMR use solo CPU y no intente acceder a GPUs:

```
◦ docker model install-runner --gpu none
```

- Listar modelos

```
◦ docker model search ai
```

- Descargar modelo

```
◦ docker model pull ai/smollm2
```

- Ejecutar el Modelo

```
◦ docker model run ai/smollm2 "Escribe una función en Python que  
calcule el factorial de un número"
```

Maquina con GPU

- nvidia tools

```
◦ # Agregar el repositorio y la clave GPG de NVIDIA
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | \
  sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-
  toolkit-keyring.gpg

curl -s -L https://nvidia.github.io/libnvidia-
  container/stable/deb/nvidia-container-toolkit.list | \
  sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-
  container-toolkit-keyring.gpg] https://#g' | \
  sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

# Instalar el toolkit
sudo apt update
sudo apt install -y nvidia-container-toolkit
```

- Configurar

```
◦ # Configurar el runtime de NVIDIA
sudo nvidia-ctk runtime configure --runtime=docker

# Reiniciar Docker para aplicar los cambios
sudo systemctl restart docker
docker model install-runner --backend vllm --gpu cuda
docker model status
```

- Verificación

```
◦ docker run --rm --gpus all nvidia/cuda:12.2.2-base-ubuntu22.04
  nvidia-smi
```

- Descargar modelo:

```
◦ docker model pull ai/mistral
docker model install-runner --gpu auto
docker model run ai/mistral "Escribe una función en Python que
  calcule el factorial de un número"
```

OpenWeb UI + (DMR)

- Ejecución simple

```
◦ docker run -d -p 3000:8080 --name open-webui openwebui/open-webui
```

- Verificar si ya levanto el contenedor este demorara porque tiene que descargar 30 files:

- `docker logs -f open-webui`

- Ingresar al IP <http://192.168.56.103:3000>
- Cambio manual al servidor <http://172.17.0.1:12434>
- Al finalizar detener y borrar el contenedor

```
docker ps
docker stop open-webui
docker rm open-webui
```

- Con docker compose
 - Crear al carpeta openwebui

```
mkdir openwebui && cd openwebui
```

- Crear el archivo docker-compose.yml

```
nano docker-compose.yml
```

- Copiar el contenido de:

[docker-compose.yml](#)

```
services:
  open-webui:
    image: openwebui/open-webui:latest
    container_name: open-webui
    restart: unless-stopped
    ports:
      - "3000:8080"
    volumes:
      - open-webui-data:/app/backend/data
    environment:
      - DMR_BASE_URL=http://172.17.0.1:12434
      - OLLAMA_BASE_URL=http://172.17.0.1:12434
      - HF_HUB_OFFLINE=1 # ← Impide
descargas externas # ← Vacío
      - HF_TOKEN=
para no autenticar
    extra_hosts:
      - "host.docker.internal:host-gateway"
    networks:
      - dmr-network

networks:
  dmr-network:
    driver: bridge

volumes:
  open-webui-data:
```

- Ejecutar:

- `docker compose up -d`

- Ingresamos directo al <http://192.168.56.103:3000>

From:

<https://dw.openit.dev/> - **DokuWikiOpenIT**

Permanent link:

<https://dw.openit.dev/lia20260525?rev=1779758137>

Last update: **2026/05/26 01:15**

