

# Docker

- [Determining and Clearing Container Disk Usage in Overlay2](#)
- [Display live container resource usage](#)
- [Install Docker and Docker Compose](#)

# Determining and Clearing Container Disk Usage in Overlay2

Show disk usage using docker df:

```
docker system df
```

Should output similar to below:

TYPE	TOTAL	ACTIVE	SIZE	RECLAIMABLE
Images	23	22	11.15GB	799.2MB (7%)
Containers	28	27	2.751GB	0B (0%)
Local Volumes	19	19	477.2MB	0B (0%)
Build Cache	0	0	0B	0B

Show disk usage using du:

```
du -shc /var/lib/docker/overlay2*/diff | grep total
```

should output similar to below:

```
15G    total
```

Show disk usage using du:

```
du -h /var/lib/docker | sort -h
```

Script to clear overlay2 storage and volumes :

```
#!/bin/bash

# remove exited containers:
docker ps --filter status=dead --filter status=exited -aq | xargs -r docker rm -v

# remove unused images:
docker images --no-trunc | grep '<none>' | awk '{ print $3 }' | xargs -r docker rmi

# remove unused volumes:
find '/var/lib/docker/volumes/' -mindepth 1 -maxdepth 1 -type d | grep -vFf <(
  docker ps -aq | xargs docker inspect | jq -r '.[[] | .Mounts | .[] | .Name | select(.)'
) | xargs -r rm -fr
```

Docker Logs can also take up a significant amount of space. Logs are located under `/var/lib/docker/containers/<container-id>/<container-id>.json.log`.

Logs can be set to maximum size and number by setting the following in `docker-compose.yml`:

```
services:
  my-app:
    image: my-app:latest
    container_name: mycontainer
    logging:
      options:
        max-size: 10m
    ....
```

Additionally, you can setup logging driver default options at the daemon level as described in the URL below:

<https://docs.docker.com/config/containers/logging/configure/#configure-the-default-logging-driver>

# Display live container resource usage

Run the following command to view CPU and memory usage of all your containers in real-time:

```
docker stats --all --format "table {{.ID}}\t{{.Name}}\t{{.CPUPerc}}\t{{.MemUsage}}"
```

# Install Docker and Docker Compose

## Install Docker Engine and Compose

Run these in a shell on your Ubuntu 24.04 box:

```
# 1) Prereqs and keyring directory
sudo apt update
sudo apt install -y ca-certificates curl gnupg

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

# 2) Add Docker's official GPG key
curl -fsSL https://download.docker.com/linux/ubuntu/gpg \
  | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg

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

# 3) Add Docker's apt repo (stable channel)
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

# 4) Update package index
sudo apt update

# 5) Install Docker Engine + CLI + containerd + Buildx + Compose plugin
```

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

# Enable and test Docker

```
# Start and enable Docker service  
sudo systemctl enable --now docker  
  
# Optional: run Docker without sudo by adding your user to docker group  
sudo usermod -aG docker $USER  
# Then log out and back in, or:  
newgrp docker  
  
# Verify Docker  
docker version  
  
# Verify the Compose plugin (note: space, not hyphen)  
docker compose version
```

## Optional: legacy `docker-compose` binary

If you explicitly need the old standalone `docker-compose` binary (for older scripts/CI), you can add it alongside the plugin:

```
sudo curl -L \  
"https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-  
$(uname -m)" \  
-o /usr/local/bin/docker-compose  
  
sudo chmod +x /usr/local/bin/docker-compose  
  
docker-compose version
```