



CDW Documentation

Docker Basics - Command Reference

Docker Basics - Command Reference

Docker Installation and Health Checks

docker version

Purpose

Verifies that the Docker CLI is installed and whether it can communicate with the Docker daemon.

What it shows

Client version information and server (daemon) version information.

Usage

```
docker version
```

Notes

If the Docker daemon is not running, only client information is shown and the server section fails.

docker info

Purpose

Displays detailed system-wide Docker configuration and runtime information.

What it shows

Docker root directory, storage driver, number of images and containers, OS and architecture details.

Usage

```
docker info
```

Working With Containers

docker run

Purpose

Creates and starts a new container from an image.

Common usage

```
docker run hello-world
```

```
docker run -it ubuntu bash
```

```
docker run -d --name my-nginx -p 8080:80 nginx
```

Key options

-it runs the container interactively

-d runs the container in the background

--name assigns a readable name to the container

-p maps host ports to container ports

docker ps

Purpose

Lists containers.

Usage

```
docker ps
```

```
docker ps -a
```

docker ps shows running containers only.

docker ps -a includes stopped containers.

docker stop and docker start

Purpose

Stops or starts an existing container.

Usage

```
docker stop my-container
```

```
docker start my-container
```

docker rm

Purpose

Removes a stopped container.

Usage

```
docker rm my-container
```

docker inspect

Purpose

Displays low-level JSON metadata for containers or images.

Typical use cases

Inspect environment variables, network settings, mounts, ports, and runtime configuration.

Usage

```
docker inspect my-container
```

Executing Commands in Running Containers

docker exec

Purpose

Runs a command inside a running container.

Interactive shell

```
docker exec -it my-container bash
```

```
docker exec -it my-container sh
```

One-off command

```
docker exec my-container ls /etc
```

Logs and Runtime Monitoring

docker logs

Purpose

Displays standard output and error logs from a container.

Usage

```
docker logs my-container
```

```
docker logs -f my-container
```

```
docker logs -tail 50 my-container
```

docker stats

Purpose

Shows real-time CPU, memory, network, and disk usage for containers.

Usage

```
docker stats
```

```
docker stats my-container
```

Images and Registries

docker pull

Purpose

Downloads an image from a container registry.

Usage

```
docker pull ubuntu
```

```
docker pull nvcr.io/nvidia/pytorch:24.04-py3
```

docker images

Purpose

Lists all locally available images.

Usage

```
docker images
```

docker tag

Purpose

Creates an additional name and tag for an image, commonly used before pushing to a registry.

Usage

```
docker tag nginx myuser/nginx:demo
```

docker push

Purpose

Uploads an image to a container registry.

Usage

```
docker push myuser/nginx:demo
```

Prerequisite

You must authenticate with the registry using docker login.

Cleanup and Disk Management

docker system df

Purpose

Displays Docker disk usage information.

Usage

```
docker system df
```

docker container prune

Purpose

Removes all stopped containers.

Usage

```
docker container prune
```

docker image prune

Purpose

Removes dangling and unused images.

Usage

```
docker image prune
```

docker system prune

Purpose

Removes unused containers, networks, images, and build cache.

Usage

docker system prune

Aggressive cleanup

docker system prune -a --volumes

This removes all unused images and volumes and should be used with caution.

Docker Contexts

docker context ls

Purpose

Lists available Docker contexts such as local, desktop, or remote environments.

Usage

docker context ls

docker context use

Purpose

Switches the active Docker context.

Usage

docker context use desktop-linux