Docker CLI

2023-04-24

Docker Run

Run a container from an image:

```
docker run nginx
```

Run a container and assign a name:

```
docker run --name my_nginx nginx
```

Run a container in detached mode:

```
docker run -d nginx
```

Run a container and publish a container’s port to the host:

```
docker run -p 8080:80 nginx
```

Run a container and mount a host directory:

```
docker run -v /path/to/host/dir:/usr/share/nginx/html nginx
```

Docker Build

Build an image from a Dockerfile:

```
docker build -t my_image:latest .
```

Build an image without cache:

```
docker build --no-cache -t my_image:latest .
```

Build an image and compress the build context:

```
docker build -t my_image:latest --compress .
```

Specify a Dockerfile name and path:

```
docker build -t my_image:latest -f /path/to/Dockerfile .
```

Build an image with build arguments:

```
docker build -t my_image:latest --build-arg ARG_NAME=ARG_VALUE .
```

Managing Containers

List running containers:

```
docker ps
```

List all containers:

```
docker ps -a
```

Stop a running container:

```
docker stop CONTAINER_ID_OR_NAME
```

Remove a stopped container:

```
docker rm CONTAINER_ID_OR_NAME
```

Remove all stopped containers:

```
docker container prune
```

Managing Images

List images:

```
docker images
```

Remove an image:

```
docker rmi IMAGE_ID_OR_NAME
```

Remove unused images:

```
docker image prune
```

Remove all unused images:

```
docker image prune -a
```

Save an image to a tar file:

```
docker save -o IMAGE_NAME.tar IMAGE_NAME:TAG
```

```
Docker CLI (continued)
```

Managing Networks

List networks:

```
docker network ls
```

Create a network:

```
docker network create NETWORK_NAME
```

Connect a container to a network:

```
docker network connect NETWORK_NAME CONTAINER_ID_OR_NAME
```

Disconnect a container from a network:

```
docker network disconnect NETWORK_NAME CONTAINER_ID_OR_NAME
```

Remove a network:

```
docker network rm NETWORK_NAME
```

Inspect a network:

```
docker network inspect NETWORK_NAME
```

Managing Volumes

List volumes:

```
docker volume ls
```

Create a volume:

```
docker volume create VOLUME_NAME
```

Inspect a volume:

```
docker volume inspect VOLUME_NAME
```

Remove a volume:

```
docker volume rm VOLUME_NAME
```

Remove unused volumes:

```
docker volume prune
```

Run a container with a named volume:

```
docker run -v VOLUME_NAME:/path/in/container IMAGE_NAME
```

Docker Compose

Start containers defined in a docker-compose.yml file:

```
docker-compose up
```

Start containers in detached mode:

```
docker-compose up -d
```

Stop containers defined in a docker-compose.yml file:

```
docker-compose down
```

Build images as defined in the docker-compose.yml file:

```
docker-compose build
```

View logs for containers defined in the docker-compose.yml file:

```
docker-compose logs
```

Scaling containers defined in a docker-compose.yml file:

```
docker-compose up -d --scale SERVICE_NAME=NUM_INSTANCES
```

Docker Exec

Execute a command inside a running container:

```
docker exec CONTAINER_ID_OR_NAME COMMAND
```

Execute an interactive shell inside a running container:

```
docker exec -it CONTAINER_ID_OR_NAME /bin/bash
```

Docker Logs

Display logs for a container:

```
docker logs CONTAINER_ID_OR_NAME
```

Display logs for a container and follow new logs:

```
docker logs -f CONTAINER_ID_OR_NAME
```

Display logs for a container with timestamps:

```
docker logs --timestamps CONTAINER_ID_OR_NAME
```

Docker Attach

Attach to a running container:

```
docker attach CONTAINER_ID_OR_NAME
```

Attach to a running container and detach without stopping the container:

```
docker attach --detach-keys="ctrl-p,ctrl-q" CONTAINER_ID_OR_NAME
```