

Dive Into Docker: The Beginner's Handbook

Build, Run, and Manage Applications with Confidence and Clarity

by: github.com/nathanaelsanilo

Docker Commands

Command	Action
<code>docker run [image]</code>	Runs a command in a new container, creating it if it doesn't exist.
<code>docker ps</code>	Lists currently running containers. Add <code>-a</code> to show all containers (running and stopped).
<code>docker stop [container_id/name]</code>	Stops a running container gracefully.
<code>docker start [container_id/name]</code>	Starts a stopped container.
<code>docker rm [container_id/name]</code>	Removes one or more stopped containers.
<code>docker rmi [image_id/name]</code>	Removes one or more images. Use with caution.
<code>docker build -t [image_name] .</code>	Builds a Docker image from a Dockerfile in the current directory.
<code>docker images</code>	Lists all local Docker images.
<code>docker pull [image_name]</code>	Downloads an image from a registry (e.g., Docker Hub).
<code>docker exec -it [container_id/name] [command]</code>	Executes a command inside a running container. <code>-it</code> provides an interactive terminal.
<code>docker volume create [volume_name]</code>	Creates a new named volume for persistent data storage.
<code>docker volume ls</code>	Lists all Docker volumes.
<code>docker volume rm [volume_name]</code>	Removes one or more volumes. Be careful, this deletes data.
<code>docker network create [network_name]</code>	Creates a user-defined bridge network for containers to communicate on.
<code>docker network ls</code>	Lists all Docker networks.
<code>docker network rm [network_name]</code>	Removes one or more networks.
<code>docker inspect [container_id/name/image_id/name]</code>	Returns low-level information on Docker objects (containers, images, volumes, networks).
<code>docker cp [source_path] [container_id]:[destination_path]</code>	Copies files/folders between a container and the local filesystem.
<code>docker save -o [tar_file] [image_name]</code>	Saves one or more images to a tar archive, useful for offline transfer.
<code>docker load -i [tar_file]</code>	Loads images from a tar archive, typically created by <code>docker save</code> .

Docker Compose

Command	Action
<code>docker-compose up</code>	Builds, (re)creates, starts, and attaches to containers for a service. Add <code>-d</code> to run in detached mode.
<code>docker-compose down</code>	Stops and removes containers, networks, and volumes defined in the <code>docker-compose.yml</code> file.
<code>docker-compose ps</code>	Lists all services defined in the <code>docker-compose.yml</code> file and their status.
<code>docker-compose build [service_name]</code>	Builds or rebuilds services. Useful after making changes to a Dockerfile.
<code>docker-compose stop [service_name]</code>	Stops running containers without removing them.
<code>docker-compose start [service_name]</code>	Starts stopped services.
<code>docker-compose restart [service_name]</code>	Restarts services.
<code>docker-compose logs [service_name]</code>	Displays log output from services.
<code>docker-compose exec [service_name] [command]</code>	Executes an arbitrary command in a running service container.
<code>docker-compose config</code>	Validates and displays the Compose file configuration.

Dockerfile

Argument	Action
<code>FROM [image]</code>	Specifies the base image for subsequent instructions. Always the first instruction.
<code>WORKDIR [path]</code>	Sets the working directory for any <code>RUN</code> , <code>CMD</code> , <code>ENTRYPOINT</code> , <code>COPY</code> , or <code>ADD</code> instructions that follow.
<code>COPY [source] [destination]</code>	Copies new files or directories from the source and adds them to the filesystem of the container at the destination path.
<code>RUN [command]</code>	Executes commands in a new layer on top of the current image, typically used for installing packages or configuring the environment.
<code>CMD ["executable", "param1", "param2"]</code>	Provides default commands or arguments for an executing container. Only the last <code>CMD</code> instruction in a Dockerfile takes effect.
<code>EXPOSE [port]</code>	Informs Docker that the container listens on the specified network ports at runtime. Does not actually publish the port.
<code>ENV [key] [value]</code>	Sets environment variables. These variables are available to subsequent instructions and at runtime.
<code>ARG [name][=default_value]</code>	Defines build-time variables that users can pass to the builder with the <code>docker build --build-arg</code> command.
<code>ENTRYPOINT ["executable", "param1", "param2"]</code>	Configures a container that will run as an executable. Often used in conjunction with <code>CMD</code> .
<code>LABEL [key]="value"</code>	Adds metadata to an image. Useful for organizing images.
<code>ARG [name]</code>	Declares a build-time variable that can be passed during the <code>docker build</code> process.
<code>COPY --from=[builder_stage] [source] [destination]</code>	Multi-stage build instruction: copies files from a previous build stage, reducing final image size.