

Command	Description	Usage	Important Options	Example
<b>build</b>	Create a new <b>image</b> from a Dockerfile.	<b>[OPTIONS] DIR</b>	<b>-t</b> Define image name and tag	<b>docker build -t myapp:0.0.1 /home/user/app</b> Create a new image called myapp with tag 0.0.1 from the Dockerfile in directory /home/user/app. The new image is now listed in <b>docker images</b> .
<b>create</b>	Create a new <b>container</b> from an image.	<b>[OPTIONS] IMAGE [CMD] [CMDARG...]</b>	<b>-e</b> Set environment variables <b>-i</b> Interactive (allow to attach local stdin, stdout, and stderr) <b>-p</b> Define port mapping <b>-t</b> Set up a terminal in the container <b>-v</b> Define a volume	<b>docker create -i -t -e F00=bar -p 8080:80 -v /home/user:/root ubuntu:16.04 bash</b> Create a new <b>Ubuntu 16.04</b> container with an <b>interactive terminal</b> running <b>Bash</b> . Set the environment variable <b>FOO</b> in the container. Expose <b>port 80</b> of the container, an map it to port <b>8080</b> on the host. Map the <b>/home/user</b> directory on the host to <b>/root</b> in the container. The new container is now listed in <b>docker ps -a</b> .
<b>start</b>	Start a container.	<b>[OPTIONS] CONTAINER...</b>	<b>-a</b> Attach <b>stdout</b> and <b>stderr</b> (show stdout and stderr of container on host) <b>-i</b> Attach <b>stdin</b> (read stdin for container from host)	<b>docker start -ai f585a987fae2</b> Start a previously created container and attach local stdout, stderr, and stdin to it. The <b>-a</b> and <b>-i</b> options together have the same effect as running <b>docker attach</b> afterwards. The container is now listed in <b>docker ps</b> .
<b>attach</b>	Attach local <b>stdin</b> , <b>stdout</b> , and <b>stderr</b> to running <b>container</b> .	<b>[OPTIONS] CONTAINER</b>		<b>docker attach f585a987fae2</b> Attach local stdout, stderr, and stdin to a previously started container. If the container runs a shell, this is like logging in to the container.
<b>stop</b>	Stop a running container.	<b>[OPTIONS] CONTAINER...</b>		<b>docker stop f585a987fae2</b> Stop a running container. The container is now still listed in <b>docker ps -a</b> , but not anymore in <b>docker ps</b> .
<b>run</b>	<b>Create</b> a new container from an image, <b>start</b> it, and <b>attach</b> to it.	<b>[OPTIONS] IMAGE [CMD] [CMDARG...]</b>	<b>-d</b> Don't attach to the container <b>-e</b> Set environment variables <b>-i</b> Interactive (allow to attach local stdin, stdout, and stderr) <b>-p</b> Define port mapping <b>-t</b> Set up a terminal in the container <b>-v</b> Define a volume	<b>docker run -i -t -e F00=bar -p 8080:80 -v /home/user:/root ubuntu:16.04 bash</b> Same example as for <b>create</b> , but now also start the container and attach local stdin, stdout, and stderr to it. The <b>run</b> command merges the three commands <b>create</b> , <b>start</b> , and <b>attach</b> into one.
<b>ps</b>	List running containers	<b>[OPTIONS]</b>	<b>-a</b> List all containers, not just running ones <b>-q</b> List only IDs of containers	<b>docker ps -aq</b> List IDs of all existing containers.
<b>images</b>	List images.	<b>[OPTIONS]</b>	<b>-a</b> Also list intermediate images <b>-q</b> List only IDs of images	<b>docker images -q</b> List IDs of all main images.
<b>rm</b>	Remove containers.	<b>[OPTIONS] CONTAINER...</b>	<b>-f</b> Force removal <b>-v</b> Also remove volumes	<b>docker rm -f \$(docker ps -aq)</b> Remove all existing containers.
<b>rmi</b>	Remove images	<b>[OPTIONS] IMAGE...</b>	<b>-f</b> Force removal	<b>docker rmi -f \$(docker images -q)</b> Remove all existing images.