Docker Cheat Sheet
Launch your own website with unlimited hosting. Use HostingerTutorials discount code and get up to 83% OFF any web hosting plan.

**Docker Cheat Sheet**

- **Build Commands**
  - `docker build`
    Builds an image from a Dockerfile located in the current directory
  - `docker build https://github.com/docker/rootfs.git#container:docker`
    Builds an image from a remote Git repository
  - `docker build -t imagename/tag`
    Builds, and tags an image for easier tracking
  - `docker build https://yourserver/file.tar.gz`
    Builds an image from a remote tar archive
  - `docker build -t image:1.0 -<<EOF
    FROM busybox
    RUN echo “hello world”
    EOF`
    Builds an image via a Dockerfile that is passed through STDIN

- **Clean Up Commands**
  - `docker image prune`
    Clears an unused image
  - `docker image prune -a`
    Clears all images that are not being used by containers
  - `docker system prune`
    Removes all stopped containers, all networks not used by containers, all dangling images, and all build cache
  - `docker image rm image`
    Removes an image
  - `docker rm container`
    Removes a running container
  - `docker kill $(docker ps -q)`
    Stops all running containers
<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker start container</code></td>
<td>Starts a new container</td>
</tr>
<tr>
<td><code>docker stop container</code></td>
<td>Stops a container</td>
</tr>
<tr>
<td><code>docker pause container</code></td>
<td>Pauses a container</td>
</tr>
<tr>
<td><code>docker unpause container</code></td>
<td>Unpauses a container</td>
</tr>
<tr>
<td><code>docker restart container</code></td>
<td>Restarts a container</td>
</tr>
<tr>
<td><code>docker wait container</code></td>
<td>Blocks a container</td>
</tr>
<tr>
<td><code>docker exec -ti container script.sh</code></td>
<td>Runs a command in a container</td>
</tr>
<tr>
<td><code>docker commit container image</code></td>
<td>Creates a new image from a container</td>
</tr>
<tr>
<td><code>docker export container</code></td>
<td>Exports container contents to a tar archive</td>
</tr>
</tbody>
</table>

**Container Interaction Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker attach container</code></td>
<td>Attaches to a running container</td>
</tr>
<tr>
<td><code>docker wait container</code></td>
<td>Waits until the container is terminated and shows the exit code</td>
</tr>
<tr>
<td><code>docker commit -m &quot;commit message&quot; -a &quot;author&quot; container username/image_name: tag</code></td>
<td>Saves a running container as an image</td>
</tr>
<tr>
<td><code>docker logs -ft container</code></td>
<td>Follows container logs</td>
</tr>
<tr>
<td><code>docker exec -ti container script.sh</code></td>
<td>Runs a command in a container</td>
</tr>
<tr>
<td><code>docker commit container image</code></td>
<td>Creates a new image from a container</td>
</tr>
<tr>
<td><code>docker create image</code></td>
<td>Creates a new container from an image</td>
</tr>
</tbody>
</table>

**Additional Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker volume rm $(docker volume ls -f dangling=true -q)</code></td>
<td>Removes all dangling volumes</td>
</tr>
<tr>
<td><code>docker kill $(docker ps -q)</code></td>
<td>Stops all running containers</td>
</tr>
<tr>
<td><code>docker rm $(docker ps -a -q)</code></td>
<td>Removes all stopped containers</td>
</tr>
<tr>
<td><code>docker stack rm stackname</code></td>
<td>Removes a swarm</td>
</tr>
<tr>
<td><code>docker volume rm $(docker volume ls -f dangling=true -q)</code></td>
<td>Removes all dangling volumes</td>
</tr>
<tr>
<td><code>docker rm $(docker ps -a -q)</code></td>
<td>Removes all stopped containers</td>
</tr>
<tr>
<td><code>docker kill $(docker ps -q)</code></td>
<td>Stops all running containers</td>
</tr>
<tr>
<td><code>docker stack rm stackname</code></td>
<td>Removes a swarm</td>
</tr>
</tbody>
</table>

**Shell Commands**

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>docker kill $(docker ps -q)</code></td>
<td>Stops all running containers</td>
</tr>
<tr>
<td><code>docker rm $(docker ps -a -q)</code></td>
<td>Removes all stopped containers</td>
</tr>
<tr>
<td><code>docker volume rm $(docker volume ls -f dangling=true -q)</code></td>
<td>Removes all dangling volumes</td>
</tr>
<tr>
<td><code>docker stack rm stackname</code></td>
<td>Removes a swarm</td>
</tr>
<tr>
<td><code>docker volume rm $(docker volume ls -f dangling=true -q)</code></td>
<td>Removes all dangling volumes</td>
</tr>
<tr>
<td><code>docker rm $(docker ps -a -q)</code></td>
<td>Removes all stopped containers</td>
</tr>
<tr>
<td><code>docker kill $(docker ps -q)</code></td>
<td>Stops all running containers</td>
</tr>
<tr>
<td><code>docker stack rm stackname</code></td>
<td>Removes a swarm</td>
</tr>
<tr>
<td><code>docker volume rm $(docker volume ls -f dangling=true -q)</code></td>
<td>Removes all dangling volumes</td>
</tr>
<tr>
<td><code>docker rm $(docker ps -a -q)</code></td>
<td>Removes all stopped containers</td>
</tr>
<tr>
<td><code>docker kill $(docker ps -q)</code></td>
<td>Stops all running containers</td>
</tr>
</tbody>
</table>
Container Inspection Commands

- `docker ps`
  Lists all running containers

- `docker -ps -a`
  Lists all containers

- `docker diff container`
  Inspects changes to directories and files in the container filesystem

- `docker top container`
  Shows all running processes in an existing container

- `docker inspect container`
  Displays low-level information about a container

- `docker logs container`
  Gathers the logs for a container

- `docker stats container`
  Shows container resource usage statistics

Manage Images Commands

- `docker image ls`
  Lists images

- `docker image rm mysql`
  Removes an image

- `docker tag image tag`
  Tags an image

- `docker history image`
  Displays the image history

- `docker inspect image`
  Displays low-level information about an image
Run Commands

Docker uses the run command to create containers from provided images. The default syntax for this command:

```
docker run [options] image [command] [arg...]
```

Then you can use one of the following flags:

- `--detach , -d`
  Runs a container in the background and prints the container ID

- `--env , -e`
  Sets environment variables

- `--hostname , -h`
  Sets a hostname to a container

- `--label , -l`
  Creates a meta data label for a container

- `--name`
  Assigns a name to a container

- `--network`
  Connects a container to a network

- `--rm`
  Removes container when it stops

- `--read-only`
  Sets the container filesystem as read-only

- `--workdir , -w`
  Sets a working directory in a container

Registry Commands

- `docker login`
  Logs in to a registry

- `docker logout`
  Logs out from a registry

- `docker pull mysql`
  Pulls an image from a registry

- `docker push repo/ rhel-httpd:latest`
  Pushes an image to a registry

- `docker search term`
  Searches Docker Hub for images with the specified term
**Service Commands**

- `docker service ls`
  Lists all services running in a swarm

- `docker stack services stackname`
  Lists all running services

- `docker service ps servicename`
  Lists the tasks of a service

- `docker service update servicename`
  Updates a service

- `docker service create image`
  Creates a new service

- `docker service scale servicename=10`
  Scales one or more replicated services

- `docker service logs stackname servicename`
  Lists all service logs

**Network Commands**

- `docker network create networkname`
  Creates a new network

- `docker network rm networkname`
  Removes a specified network

- `docker network ls`
  Lists all networks

- `docker network connect networkname container`
  Connects a container to a network

- `docker network disconnect networkname container`
  Disconnects a container from a network

- `docker network inspect networkname`
  Displays a detailed information about a network