**Docker Commands**

**Containers**

**Lifecycle**

* [docker create](https://docs.docker.com/engine/reference/commandline/create) creates a container but does not start it.
* [docker rename](https://docs.docker.com/engine/reference/commandline/rename/) allows the container to be renamed.
* [docker run](https://docs.docker.com/engine/reference/commandline/run) creates and starts a container in one operation.
* [docker rm](https://docs.docker.com/engine/reference/commandline/rm) deletes a container.
* [docker update](https://docs.docker.com/engine/reference/commandline/update/) updates a container's resource limits.

**Starting and Stopping**

* [docker start](https://docs.docker.com/engine/reference/commandline/start) starts a container so it is running.
* [docker stop](https://docs.docker.com/engine/reference/commandline/stop) stops a running container.
* [docker restart](https://docs.docker.com/engine/reference/commandline/restart) stops and starts a container.
* [docker pause](https://docs.docker.com/engine/reference/commandline/pause/) pauses a running container, "freezing" it in place.
* [docker unpause](https://docs.docker.com/engine/reference/commandline/unpause/) will unpause a running container.
* [docker wait](https://docs.docker.com/engine/reference/commandline/wait) blocks until running container stops.
* [docker kill](https://docs.docker.com/engine/reference/commandline/kill) sends a SIGKILL to a running container.
* [docker attach](https://docs.docker.com/engine/reference/commandline/attach) will connect to a running container.

**Info**

* [docker ps](https://docs.docker.com/engine/reference/commandline/ps) shows running containers.
* [docker logs](https://docs.docker.com/engine/reference/commandline/logs) gets logs from container. (You can use a custom log driver, but logs is only available for json-file and journald in 1.10).
* [docker inspect](https://docs.docker.com/engine/reference/commandline/inspect) looks at all the info on a container (including IP address).
* [docker events](https://docs.docker.com/engine/reference/commandline/events) gets events from container.
* [docker port](https://docs.docker.com/engine/reference/commandline/port) shows public facing port of container.
* [docker top](https://docs.docker.com/engine/reference/commandline/top) shows running processes in container.
* [docker stats](https://docs.docker.com/engine/reference/commandline/stats) shows containers' resource usage statistics.
* [docker diff](https://docs.docker.com/engine/reference/commandline/diff) shows changed files in the container's FS.

**Import / Export**

* [docker cp](https://docs.docker.com/engine/reference/commandline/cp) copies files or folders between a container and the local filesystem.
* [docker export](https://docs.docker.com/engine/reference/commandline/export) turns container filesystem into tarball archive stream to STDOUT.

**Executing Commands**

* [docker exec](https://docs.docker.com/engine/reference/commandline/exec) to execute a command in container.

**Images**

**Lifecycle**

* [docker images](https://docs.docker.com/engine/reference/commandline/images) shows all images.
* [docker import](https://docs.docker.com/engine/reference/commandline/import) creates an image from a tarball.
* [docker build](https://docs.docker.com/engine/reference/commandline/build) creates image from Dockerfile.
* [docker commit](https://docs.docker.com/engine/reference/commandline/commit) creates image from a container, pausing it temporarily if it is running.
* [docker rmi](https://docs.docker.com/engine/reference/commandline/rmi) removes an image.
* [docker load](https://docs.docker.com/engine/reference/commandline/load) loads an image from a tar archive as STDIN, including images and tags (as of 0.7).
* [docker save](https://docs.docker.com/engine/reference/commandline/save) saves an image to a tar archive stream to STDOUT with all parent layers, tags & versions (as of 0.7).

**Info**

* [docker history](https://docs.docker.com/engine/reference/commandline/history) shows history of image.
* [docker tag](https://docs.docker.com/engine/reference/commandline/tag) tags an image to a name (local or registry).
* docker image prune - for removing unused images

### Load/Save image

Load an image from file:

docker load < my\_image.tar.gz

Save an existing image:

docker save my\_image:my\_tag | gzip > my\_image.tar.gz

### Import/Export container

Import a container as an image from file:

cat my\_container.tar.gz | docker import - my\_image:my\_tag

Export an existing container:

docker export my\_container | gzip > my\_container.tar.gz

**Networks**

### Lifecycle

* [docker network create](https://docs.docker.com/engine/reference/commandline/network_create/)
* [docker network rm](https://docs.docker.com/engine/reference/commandline/network_rm/)

### Info

* [docker network ls](https://docs.docker.com/engine/reference/commandline/network_ls/)
* [docker network inspect](https://docs.docker.com/engine/reference/commandline/network_inspect/)

### Connection

* [docker network connect](https://docs.docker.com/engine/reference/commandline/network_connect/)
* [docker network disconnect](https://docs.docker.com/engine/reference/commandline/network_disconnect/)

You can specify a [specific IP address for a container](https://blog.jessfraz.com/post/ips-for-all-the-things/):

# create a new bridge network with your subnet and gateway for your ip block

docker network create --subnet 203.0.113.0/24 --gateway 203.0.113.254 iptastic

# run a ngin

x container with a specific ip in that block

$ docker run --rm -it --net iptastic --ip 203.0.113.2 nginx

# curl the ip from any other place (assuming this is a public ip block duh)

$ curl 203.0.113.2

## **Registry & Repository**

A repository is a hosted collection of tagged images that together create the file system for a container.

A registry is a host -- a server that stores repositories and provides an HTTP API for [managing the uploading and downloading of repositories](https://docs.docker.com/engine/tutorials/dockerrepos/).

Docker.com hosts its own [index](https://hub.docker.com/) to a central registry which contains a large number of repositories. Having said that, the central docker registry [does not do a good job of verifying images](https://titanous.com/posts/docker-insecurity) and should be avoided if you're worried about security.

* [docker login](https://docs.docker.com/engine/reference/commandline/login) to login to a registry.
* [docker logout](https://docs.docker.com/engine/reference/commandline/logout) to logout from a registry.
* [docker search](https://docs.docker.com/engine/reference/commandline/search) searches registry for image.
* [docker pull](https://docs.docker.com/engine/reference/commandline/pull) pulls an image from registry to local machine.
* [docker push](https://docs.docker.com/engine/reference/commandline/push) pushes an image to the registry from local machine.

## **Dockerfile**

[The configuration file](https://docs.docker.com/engine/reference/builder/). Sets up a Docker container when you run docker build on it. Vastly preferable to docker commit.

Here are some common text editors and their syntax highlighting modules you could use to create Dockerfiles:

* If you use [jEdit](http://jedit.org), I've put up a syntax highlighting module for [Dockerfile](https://github.com/wsargent/jedit-docker-mode) you can use.
* [Sublime Text 2](https://packagecontrol.io/packages/Dockerfile%20Syntax%20Highlighting)
* [Atom](https://atom.io/packages/language-docker)
* [Vim](https://github.com/ekalinin/Dockerfile.vim)
* [Emacs](https://github.com/spotify/dockerfile-mode)
* [TextMate](https://github.com/docker/docker/tree/master/contrib/syntax/textmate)
* [VS Code](https://github.com/Microsoft/vscode-docker)
* Also see [Docker meets the IDE](https://domeide.github.io/)

### Instructions

* [.dockerignore](https://docs.docker.com/engine/reference/builder/#dockerignore-file)
* [FROM](https://docs.docker.com/engine/reference/builder/#from) Sets the Base Image for subsequent instructions.
* [MAINTAINER (deprecated - use LABEL instead)](https://docs.docker.com/engine/reference/builder/#maintainer-deprecated) Set the Author field of the generated images.
* [RUN](https://docs.docker.com/engine/reference/builder/#run) execute any commands in a new layer on top of the current image and commit the results.
* [CMD](https://docs.docker.com/engine/reference/builder/#cmd) provide defaults for an executing container.
* [EXPOSE](https://docs.docker.com/engine/reference/builder/#expose) informs Docker that the container listens on the specified network ports at runtime. NOTE: does not actually make ports accessible.
* [ENV](https://docs.docker.com/engine/reference/builder/#env) sets environment variable.
* [ADD](https://docs.docker.com/engine/reference/builder/#add) copies new files, directories or remote file to container. Invalidates caches. Avoid ADD and use COPY instead.
* [COPY](https://docs.docker.com/engine/reference/builder/#copy) copies new files or directories to container. By default this copies as root regardless of the USER/WORKDIR settings. Use --chown=<user>:<group> to give ownership to another user/group. (Same for ADD.)
* [ENTRYPOINT](https://docs.docker.com/engine/reference/builder/#entrypoint) configures a container that will run as an executable.
* [VOLUME](https://docs.docker.com/engine/reference/builder/#volume) creates a mount point for externally mounted volumes or other containers.
* [USER](https://docs.docker.com/engine/reference/builder/#user) sets the user name for following RUN / CMD / ENTRYPOINT commands.
* [WORKDIR](https://docs.docker.com/engine/reference/builder/#workdir) sets the working directory.
* [ARG](https://docs.docker.com/engine/reference/builder/#arg) defines a build-time variable.
* [ONBUILD](https://docs.docker.com/engine/reference/builder/#onbuild) adds a trigger instruction when the image is used as the base for another build.
* [STOPSIGNAL](https://docs.docker.com/engine/reference/builder/#stopsignal) sets the system call signal that will be sent to the container to exit.
* [LABEL](https://docs.docker.com/config/labels-custom-metadata/) apply key/value metadata to your images, containers, or daemons.