Reference Sheet

Docker Command Line Interface

All instances of <container name> can be replaced with <container id> for unnamed containers. Containers and images can be tagged and accessed by specifying <container name:container tag> instead of <container name>

Command	Option	Details
docker build -t <image name> <relative path>		The relative path should point to the folder containing a Dockerfile
docker commit <container name> <image name>		Create an image from a container
docker push <image name>		Push an image to a repository
docker run [options] <image> <process>		Launch a new container running process. The process is optional if CMD is specified in the Dockerfile
	-d	Run the container in the background (detach)
	-i	Launches the process in interactive mode
	--name <container name>	Assigns a name to the container
	-p <host port>:<container port>	Publish a container port to the host
	-P	Publish all container ports to random ports on the host in the range defined by /proc/sys/net/ipv4/ip_local_port_range
	-t	Launches a pseudo-terminal so that you can detach and have the process continue to run in the container
	-v <host directory>:<container directory>	Bind mount a volume between the host and container
	other options	You can also specify the resources to allocate to the container; and other options. https://docs.docker.com/reference/commandline/run
docker attach <container name>		Attach the terminal to the container
docker exec [options] <container name> <process>		Execute a process on a running container
	-d	Run the container in the background (detach)
	-i	Launches the process in interactive mode
	-t	Launches a pseudo-terminal so that you can detach and have the process continue to run in the container
docker diff <container name>		Show a list of filesystem changes from the image to the container https://docs.docker.com/reference/commandline/diff
docker cp <container name>:<container path> <host path>		Copy a path from the container to the host
docker cp <host path> <container name>:<container path>		Copy a path from the host to the container (only for Docker 1.8 and later)
docker port [container port]		Show the host to container port mapping
docker ps		Show a list of running containers and their metadata
docker images		Show a list of images and their metadata

docker stop [options] <container name> [<container name> ...]		Stop running containers by sending *SIGTERM* before *SIGKILL*
	-t <seconds>	Period between terminate and kill
docker kill <container name> [<container name> ...]		Kill running containers by sending *SIGKILL*
docker restart [options] <container name> [<container name> ...]		Restart running containers using *SIGTERM* before *SIGKILL*
	-t <seconds>	Period between terminate and kill
docker rm <container name> [<container name> ...]		Remove stopped containers
docker rmi <image name> [<image name> ...]		Remove images

Dockerfile

A Dockerfile contains a series of instructions used to create a Docker image. The final image is composed of the base image and the series of diffs obtained after running each instruction. Each diff is cached.

Instruction	**Details**
FROM <base image>	Choose the image to use as a starting point.
MAINTAINER <name> <\<email\>>	Set the maintainer metadata for the image.
ENV <environment variable> <value>	Set persistent environment variables in the container.
ADD <local path or URL> <container path>	Copy a file or folder to the container. If the file is in a recognized compression, the contents will be extracted.
COPY <local path> <container path>	Copy a file or folder to the container.
USER <username>	Set the UID to be used when running the image.
WORKDIR <container directory>	Set the working directory for RUN, CMD and ENTRYPOINT instructions.
RUN <command line>	Execute a command to create a ***new*** image of the diff from the previous image. Note that the new image is an image of the filesystem, not of the container state. If you RUN a background daemon and then RUN a process that depends on the daemon, the daemon will no longer be running during the second RUN instruction because the second RUN runs in a new container. To run a process that depends on a daemon, they must be run in the same command line using the && shell operator. You can also put the commands in a shell script and RUN the shell script.
CMD \[<"executable">[, <"param1">, <"param2">, ...]\]	Provide a default command to run when the container is launched. ie. if you call 'docker run image' without specifying the process, the container will run using your CMD. If you provide a process in 'docker run ...', it will override this CMD. Only the last CMD in a Dockerfile will take effect.

ENTRYPOINT \[<"executable">[, <"param1">, <"param2">, ...]\]	Provide a mandatory instruction that will be run when the container is launched. Additional parameters can be appended from 'docker run ...', or overridden by specifying the --entrypoint option. ie. docker run image --entrypointOption will append --entrypointOption to your ENTRYPOINT, and run that command to launch the container. Only the last ENTRYPOINT in a Dockerfile will take effect, and will override any CMD.
VOLUME <host directory>:<container directory>	Bind mount a volume between the host and container. This is useful for storing log files and to persist data from the database. Mounted volumes are available during run-time, but not during build-time. ie. you cannot mount a volume of json files to import during the build process.
EXPOSE <container port> [<container port> ...]	Expose the container ports to the host. Alternatively, the ports can be exposed setting the --expose option on the command line.

Note: Try to use RUN instead of CMD or ENTRYPOINT to import data into the database because Docker will cache an image with the data loaded in the database. If you use CMD or ENTRYPOINT to run your database import tool, the data will be imported every time you start that container, which could take a long time if you have a lot of data. And if you base a new image on an image that was not built with the data cached, then all subsequent images will also require the CMD or ENTRYPOINT to import the data.