Docker CLI cheatsheet

docker build

```
docker build [options] .
  -t "app/container_name"    # name
```

Create an image from a Dockerfile.

docker run

```
docker run [options] IMAGE
  # see `docker create` for options
```

Example

```
$ docker run -it debian:buster /bin/bash
```

Run a command in an image.

#Manage containers

docker create

```
docker create [options] IMAGE
  -a, --attach               # attach stdout/err
  -i, --interactive          # attach stdin (interactive)
  -t, --tty                  # pseudo-tty
      --name NAME            # name your image
  -p, --publish 5000:5000    # port map
      --expose 5432          # expose a port to linked containers
  -P, --publish-all          # publish all ports
```

```
    --link container:alias # linking
 -v, --volume `pwd`:/app    # mount (absolute paths needed)
 -e, --env NAME=hello       # env vars
```

Example

```
$ docker create --name app_redis_1 \
 --expose 6379 \
 redis:3.0.2
```

Create a container from an image.

docker exec

```
docker exec [options] CONTAINER COMMAND
 -d, --detach        # run in background
 -i, --interactive   # stdin
 -t, --tty           # interactive
```

Example

```
$ docker exec app_web_1 tail logs/development.log
$ docker exec -t -i app_web_1 rails c
```

Run commands in a container.

docker start

```
docker start [options] CONTAINER
  -a, --attach        # attach stdout/err
  -i, --interactive   # attach stdin

docker stop [options] CONTAINER
```

Start/stop a container.

docker ps

```
$ docker ps
$ docker ps -a
$ docker kill $ID
```

Manage containers using ps/kill.

#Images

docker images

```
$ docker images
 REPOSITORY   TAG        ID
 ubuntu       12.10      b750fe78269d
 me/myapp     latest     7b2431a8d968
```

```
$ docker images -a   # also show intermediate
```

Manages images.

docker rmi

```
docker rmi b750fe78269d
```

Deletes images.

#Clean up

Clean all

```
docker system prune
```

Cleans up dangling images, containers, volumes, and networks (ie, not associated with a container)

```
docker system prune -a
```

Additionally remove any stopped containers and all unused images (not just dangling images)

Containers

```
# Stop all running containers
docker stop $(docker ps -a -q)

# Delete stopped containers
docker container prune
```

Images

```
docker image prune [-a]
```

Delete all the images

Volumes

```
docker volume prune
```

Delete all the volumes