dockerexp1 Cheat Sheet

by mcz via cheatography.com/85452/cs/20039/

Start

docker version	get the docker version
docker run hello--world	check if everething works
docker run -d redis	-d for run in background (detach mode)
docker ps	show all runs. `ps` stands for 'process status'
docker ps -a	show all run / paused / stoped
docker stop CONTNAME	stop a running container with his name
docker rm CONTNAME	remove the instance
docker rm $(docker ps -a -q)	remove all listed instances
docker system prune	Erase all /!\ use for big clean
docker kill CONTNAME	stop close nicely with maybe some save, message etc Kill will kill now

DOCKER IMAGE

docker image ls	List all your images

Useful flags

ls	List running
ls --all	List all
ls -aq	List all in quiet mode

REDIS

docker run redis	will start the redis image
docker exec -it 0c04f252f8d2 redis-cli	In an other terminal run this command to access the redis instance with redis-cli

exec : execute multi command
-it : interactive mode == -i -t : (-i attach to stdin) (-t is use to format nicely what you see)

USEFUL COMMANDS

docker cp folder/test.txt CONTNAME:/path	CP - Copy file to/from docker container:
docker image inspect IMAGENAME	INSPECT - use to inspect info about an image
docker logs	Access to logs useful when build, run fail

DELETE IMAGES & CONTAINERS

docker rmi <your-image-id>	delete docker image (no container linked)
docker container rm <container_id_1>	Delete container
docker image prune -a	delete all images
docker container prune	delete all stopped containers

Working with volume

VOLUME ["app/tempfile"]	Add anonymous volume for temp files in Dockerfile
docker volume ls	List all volumes available
docker run -p 3000:3000 --rm --name test-app -v tosave:app/saveme imageName:info	Create a named volume to get data persistant
docker volume rm VOL_NAME	To delete a specific volume
docker volume prune	To clear al volumes

ENV & ARG

ENV *name value*	is use on docker file
--env PORT=3000 or -e PORT=3000	is use on `docker run` (overwrite dockerfile)
--env-file .env	on`docker run` use a file to define env variable
ARG DEFAULT_PORT=80	is use on dockerfile for the image build
ENV PORT $DEFAULT_PORT	ex of use ARG in dockerfile
docker build -app:1 --build-arg DEFAULT_PORT=3000	Change ARG in dockerfile on build with `--build-arg`

TIPS 1 - CREATE START LOGS

docker run = docker create + docker start

docker create hello-world	return a container id
docker start -a 456789123dj..	-a mean get output to my terminal
docker logs 456789123dj..	print the logs when not use the -a

456789123dj.. is the container id

Cheatography

dockerexp1 Cheat Sheet

by mcz via cheatography.com/85452/cs/20039/

TIPS 2 - EXEC -it SH

docker exec -it 0c04f252f8d2 sh	open a shell in a running container
cmd + d or type `exit`	exit the shell
docker run -it busybox sh	attach stdin stdout & run shell

DOCKERFILE

Create a `Dockerfile`	FROM WORKDIR COPY RUN EXPOSE CMD
docker build .	run this command to build your image
=> Successfully built b3cc42a9ef20	Return you the image id
docker build -t myredis .	Add a tag to the image
mcz/redis:latest	tag convention: dockerid/projectname:version
FROM node:alpine	take the node base image
COPY ./ ./	copy all file inside our project
WORKDIR /user/app	Set a default safe directory to pull files
RUN npm install	node classic action
ENV PORT 3000	Use env variable in your docker file `EXPOSE $PORT`
CMD ["npm", "start"]	Script start in package.json
docker build -t john/node-server .	Build image with a tag name
docker run -p 8080:8080 john/nodeserver	Set port mapping to enable incoming traffic
docker ps	list running container with id
docker exec -it c66323-38e690 sh	Get a shell inside your running container
docker build -f Dockerfile.dev	To run a dev version called Dockerfile.dev

DOCKER COMPOSE

docker-compose	List all cmd available
docker-compose up	like `docker run`
docker-compose up --build	build image + run
docker-compose up -d	With -d run in background
docker-compose down	Stop the containers
restart: on-failure	LIST OF RESART CMD
"no"	never restart
always	restart anyway
on-failure	restart on process.exit(34) value > 0
unless-stoped	always unless the dev stop
docker-compose ps	Only works in the current directory

Terminology

SIGTERM	terminate signal
SIGKILL	Kill signal