Docker Commands

Docker PS / docker ps --all

Shows running containers

Docker ps –all

Shows all the containers that have ever run

Docker run = docker create + docker start

Docker start -a <container-id> (the -a is to watch for output and show it)

Docker system prune

Deletes stopped containers and build cache

Docker logs <container-id>

Shows what the container would have output

Docker stop <container-id>

Docker kill <container-id>

Docker exec -it <container-id> <command>

Start a second program in a container. The -it flag does not listen for text input. -it = -i -t . -i is for input and -t is for terminal format.

To create an image

1. Docker File
2. Docker Client
3. Docker Server
4. Usable Image!

To tag an image we have to change the build command

Docker build -t username/projectname:version context

Docker build -t kevingreenb1/redis:latest .

Docker run kevingreenb1/redis

Docker run -p 8080:8080 kevingreenb1/simpleweb

Listens from localport/containerport

WORKDIR is an instruction to set the working directory where files should be copied

docker run -it kevingreenb1/simpleweb sh

docker run myimage = docker-compose up

docker build. + docker run myimage = docker-compose up –build

docker-compose down

npm run start

starts the app

npm run test

tests the app

npm run build

builds the app

docker build -f Dockerfile.dev .

This specifies the name of the dockerfile to use while building the app

docker run -p 3000:3000 -v $(pwd):/app <contianer-id>

This command is run so if you build the docker container and make changes to it, it automatically refreshes without having to rebuild

docker run -p 3000:3000 -v /app/node\_modules -v $(pwd):/app <contianer-id>

this says not to try to map anything for node\_modules

docker run <container-id> npm run test

To Test

1. Build

Docker build -f Dockerfile.dev .

1. Run for test

docker run <container-id> npm run test