1. List containers  
   docker ps
2. List all containers (including stopped ones)  
   docker ps –all (--a, -a)
3. Remove all containers from disk  
   docker system prune
4. docker run = docker create + docker start
5. docker start + <container\_id>
6. Start container and print last process (command)  
   docker start -a <container\_id>  
   docker start -a 54fsdf234dfas echo Hello World (cannot override default command used at container creation)
7. **Note to point above:**  
   docker run <image\_id> npm run test (here we can override default command which was specified in Dockerfile as container is not yet created)
8. Get logs from container  
   docker logs <container\_id>
9. Execute an additional command in a container  
   docker exec -it <container\_id> <command>
10. Dockerfile most important instructions:
    1. FROM
    2. RUN
    3. CMD
11. Remove all images  
    docker rmi $(docker images -a -q)
12. Remove single image  
    docker rmi <image\_id>
13. docker run -it <image\_id> sh (overriding default command on run with „sh”)
14. **WORKDIR /usr/app –** any following command will be executed relative to this path in the container
15. **SECTION: 5**
16. Docker compose
    1. docker run <image> = docker-compose up
    2. docker build . + docker run <image> = docker-compose up –build
    3. Launch in the background → docker-compose up -d
    4. Stop containers → docker-compose down
    5. docker ps = docker-compose ps (it can be run only in the folder that contains   
       docker-compose.yml)
17. docker build with custom file name  
    docker build -f <file\_name>
18. **SECTION: 6**
19. Running docker with custom Dockerfile name  
    docker build -f <Dockerfile\_file> .
20. Docker volume:  
    Inside container we no longer copy content from local directory.   
    We esentially set up a „placeholder” of sorts inside of a docker container. And so we’re no longer going to copy over the entire „src” directory or the entire „public” directory. Instead, you can kind of imagine that we’re going to put in a kind of reference here. The volume is esentially going to set up a reference that’s going to point back to our local machine.  
    docker run -p 3000:3000 -v /app/node\_modules (put a bookmark on that folder, it’s necessary as locally I deleted node\_modules folder and container reference would point to nothing is such case and by doing that syntax, we tell docker that we want this folder to be a placeholder for a folder that it’s inside a container – we have node\_modules inside a container after execution npm install command)   
    -v $(pwd):/app (mark, connect the „pwd” into the „/app” folder) <image\_id>