**Docker Practice**

1. **Create docker hub account – Sign Up Free**
   1. <https://hub.docker.com>
2. **Install Docker**
   1. sudo apt update
   2. sudo apt install apt-transport-https ca-certificates curl software-properties-common
   3. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –
   4. sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable"
   5. sudo apt update
   6. apt-cache policy docker-ce
   7. sudo apt install docker-ce
   8. sudo chmod 666 /var/run/docker.sock
   9. sudo systemctl status docker
3. **Executing docker commands**
   1. docker – lists all commands
   2. docker <commands> --help
   3. docker info
4. **Working with docker images**
   1. docker run hello-world
   2. docker search ubuntu
   3. docker pull ubuntu
   4. docker images
5. **Running a docker container**
   1. Run a container using the latest image of Ubuntu. The combination of the **-i** and **-t** switches gives you interactive shell access into the container:
      1. docker run -it ubuntu
         1. Inside docker – apt update, apt install nodejs, node -v
6. **Managing docker containers**
   1. docker ps
   2. docker ps -a
   3. docker ps -l
   4. docker start <container\_id>
   5. docker stop <name>
   6. docker rm <name>
7. **Committing your changes to docker image**
   1. docker commit -m "added Node.js" -a "sammy" d9b100f2f636 sammy/ubuntu-nodejs
   2. docker images
8. **Pushing docker images to a Docker repository**
   1. docker login -u docker-registry-username
   2. Enter password at prompt
   3. docker push docker-registry-username/docker-image-name
      1. docker push sammy/ubuntu-nodejs
   4. Check in docker hub for this image from Browser
9. **Create a docker file and build docker image**

# Add below lines to a file named - **Dockerfile**

FROM node:12-alpine

RUN apk add --no-cache python g++ make

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

* Run below commands to build a docker image from above Dockerfile

$ docker build -t yourusername/my\_trng\_repo .

$ docker images

$ docker run -p80:80 yourusername/my\_trng\_repo