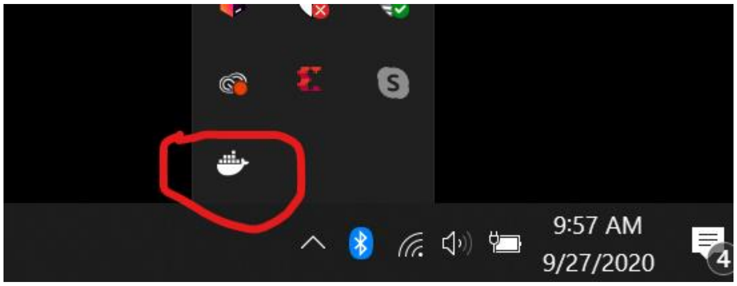
Step 1: First create a docker account from URL(https://hub.docker.com/)

Step 2 :Download desktop for docker from URL(https://www.docker.com/get-started)

Step 3: Install desktop docker

Step 4: A little docker desktop will appear in your taskbar:

Picture:



Step 5: Login to docker hub and create a repository to get started with repository.

Step 6: open your git bash terminal (recommended) in new folder Docker.

Step 7 : run :

git clone https://github.com/dockersamples/node-bulletin-board

cd node-bulletin-board/bulletin-board-app

this will install all the dependencies in your machine on that current folder required to build an images.

step 8: Make sure you’re in the directory node-bulletin-board/bulletin-board-app and run :

```
docker build --tag bulletinboard:1.0 .
```

this will build an image name bulletinboard that will tag in 1.0.

step 9: To start a container run:

```
docker run --publish 8000:8080 --detach --name bb bulletinboard:1.0
```

Explanation:

There are a couple of common flags here:

- `--publish` asks Docker to forward traffic incoming on the host’s port 8000 to the container’s port 8080. Containers have their own private set of ports, so if you want to reach one from the network, you have to forward traffic to it in this way. Otherwise, firewall rules will prevent all network traffic from reaching your container, as a default security posture.
- `--detach` asks Docker to run this container in the background.
- `--name` specifies a name with which you can refer to your container in subsequent commands, in this case `bb`.

Step 10: you can delete your container by running:

```
docker rm --force bb
```

Remember:

The **--force** option stops a running container, so it can be removed. If you stop the container running with docker stop bb first, then you do not need to use **--force** to remove it.

Step 11: How to Share Your Image in Docker Hub:

1. Click on the Docker icon in your menu bar, and navigate to Repositories > Create. You'll be redirected to the Create Repository page on Docker Hub.
2. Type the repository name as **bulletinboard** and click Create at the bottom of the page. Do not fill any other details for now.
3. Run: **docker tag bulletinboard:1.0 <Your Docker ID>/bulletinboard:1.0**
4. Run: **docker push <Your Docker ID>/bulletinboard:1.0**
5. To make sure you can now delete your image running:
   docker rmi --force <your Docker ID>/bulletinboard:1.0
6. To Pull that that image form your docker hub repository run:
   docker pull <your Docker ID>/bulletinboard:1.0

Step 12: create your docker image, cointainer from node.js

1. Download latest node.js from URL(https://nodejs.org/en/)
2. Create a new folder name node-docker for your dependencies
3. Direct your git terminal to this folder
4. Run**: npm init -y**
5. Run: **npm install ronin-server ronin-mocks**
6. Run: **touch server.js**
7. Now, open server.js file and type:

```
const ronin     = require( 'ronin-server' )
const mocks     = require( 'ronin-mocks' )


const server = ronin.server()



server.use( '/', mocks.server( server.Router(), false, true ) )

server.start()
```

Explanation: The mocking server is called Ronin.js and will listen on port 8000 by default. You can make POST requests to the root (/) endpoint and any JSON structure you send to the server will be saved in memory. You can also send GET requests to the same endpoint and receive an array of JSON objects that you have previously POSTed.

8. Run to start your application: **node server.js**
9. Create a docker file name **Dockerfile in your editor.**
10. Write the following code in that file to build an image form node file.

```
FROM node:12.18.1

ENV NODE_ENV=production



WORKDIR /app



COPY ["package.json", "package-lock.json*", "./"]
```

```
RUN npm install

COPY . .

CMD [ "node", "server.js" ]
```

Explanation: 1st line of code: it takes node image form 12.18.1 to our file.

2nd line of code: It will create a node environment. production node environment is recommended.

3rd line of code: It will create s working directory called app to run our command.

4th line of code: It will copy package.json into package-lock.json* of our image.

5th line of code: It will install npm locally in our computer.

6th line of code: It will add our source code into our image.

7th line of code: It will tell our docker container to run **node** and **server.js** command inside our container.

11. Run to build your image: docker build --tag node-docker .
12. Run to tag your image: docker tag node-docker:latest node-docker:v1.0.0
13. Run to see docker images: docker images
14. Run to remove image: docker rmi node-docker:v1.0.0

Step 12: To run your Container:

1. Simple run: **docker run node-docker**
2. To publish a port to our container run**:**

   docker run --publish 8000:8000 node-docker
3. To run in a detached mode or in background run:

   docker run -d -p 8000:8000 node-docker
4. To stop a container run:

   docker stop <container id / container name>

Step 13: To restart your container run:

docker restart <container id / container name>

Step 14: some common command used in docker with their explanation:

docker build --tag <imagemane> : This command is to build image but to run this command your need to have installed dependencies or have node.js file .go to step 12.

docker images: To see the list of images.

docker run – help: To get help on run commands

docker ps: To see what container is running.

docker ps-a: To run all the container.

docker rm <container id/ container name>: To remove container.

docker rmi <image id/ image name>: **to remove image**

I get all the sources from URL (https://docs.docker.com/get-started/)