BSc (Hons) in Information Technology
Specializing in Software Engineering
Year 3 - 2021
SE3040 – Application Frameworks
Lab 11 – Docker

1. Install Docker to your local machine.
    a. If you are using Windows, follow all the instructions as you need to install a Linux kernel.

2. Start Docker service (you can start it by opening the Docker, once started you should see a whale like icon in the taskbar where background processes are). You should be able to see the Docker dashboard as well. Also, you will be able to log into your Docker Hub account from the Docker desktop application.

3. Create a file named Dockerfile in you Koa REST application. Add the following to the file.

```
FROM node:14-alpine3.13

WORKDIR /app

COPY . .

RUN npm install

EXPOSE 3000/tcp

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

   Your image is based on node:14-alpine3.13. You are setting your work directory and then copy all the content in your project to that directory. After that you are downloading all npm dependencies. Last two commands tell Docker on what port it is exposed from the container and how to start the container.

4. Now let's build the image.

   docker build -t sample-rest-service .
   This command should run from your source directory where the Dockerfile is. Note the 'period/full stop' at the end of the command (used to indicate 'this' directory). The image name is sample-rest-service. You can change it as per your project.
5. Now you can run the image.

   docker run -d --name sample-rest-service -p 3000:3000 sample-rest-service:latest
   -d : Detached mode (run in background do not hold the console)
   -name : Name of the container
   -p : Container port and host port
   Image name and the version.
6. You can check the status of your container by running,

   docker ps -a
   Your container should be in running state.
7. You can check the logs of your container by running,

   docker logs <container-id>
8. To stop your container,

   docker stop <container-id>

9. To remove your container

   docker rm <container-id>
10. You can ssh into your container by,

    docker exec -it <container id> sh
11. Create a file named docker-compose.yml

```yaml
version: "3.7"
services:
  app:
    image: sample-rest-service
    ports:
      - 3000:3000
    working_dir: /app
    deploy:
      resources:
        limits:
          cpus: '0.25'
          memory: 300M
```

12. You can run your app using docker-compose up and shutdown using docker-compose down.

Note: If your application is connecting to a DB on localhost change the IP address to your local machine IP address (use ipconfig in Windows, ifconfig in Linux).

Ref:

1. https://docs.docker.com/engine/reference/builder/
2. https://docs.docker.com/compose/compose-file/
3. https://docs.docker.com/engine/api/v1.41/#
4. https://docs.docker.com/engine/reference/run/
5. https://docs.docker.com/compose/reference/