**Dockerizing an Application**

Run the **javascript** application on localhost

Pull images of **mongo** and **mongo-express** from DockerHub in local

Create a Docker Network “**mongo-network**”

Start Mongo DB

**docker run -d –p 27017:27017 -e MONGO\_INITDB\_ROOT\_USERNAME=admin –e MONGO\_INITDB\_ROOT\_PASSWORD=password –network mongo-network –name mongodb mongo**

Start Mongo Express

**Docker run –d –p 8081:8081 –e ME\_CONFIG\_MONGODB\_ADMINUSERNAME=admin –e ME\_CONFIG\_MONGODB\_ADMINPASSWORD=password –e ME\_CONFIG\_MONGODB\_SERVER= mongodb –network mongo-network –name mongo-express mongo-express**

Mongo-express UI will open at [**http://localhost:8081**](http://localhost:8081)

Create **“my-db”** database and **“users”** collection in mongo-express UI

Start your nodejs app locally by going in the **/app** directory and running

**npm install**

**node server.js**

Access the nodejs application from browser [**http://localhost:3000**](http://localhost:3000)

Alternative way to run the application is by using **Docker-Compose**

Create a **docker-compose.yaml** file for both the services mongo and mongo-express

**docker-compose –f docker-compose.yaml up** to spin up the containers

Mongo-express UI will open at [**http://localhost:8081**](http://localhost:8081)

Create **“my-db”** database and **“users”** collection in mongo-express UI

Start your nodejs app locally by going in the **/app** directory and running

**npm install**

**node server.js**

Access the nodejs application from browser [**http://localhost:3000**](http://localhost:3000)

Create a **Dockerfile** which will be used to build an image of the application

**docker build –t my\_app:1.0 .** (Build an image from Dockerfile)