Connecting an Express JS app with MongoDB

1)

Open Command Line and get into git repository folder , then run the following command to build the Docker Image

- **docker build -t [IMAGE_NAME]:[VERSION] .**

  Note --> '.' in the command states that , Dockerfile is in present Working Directory

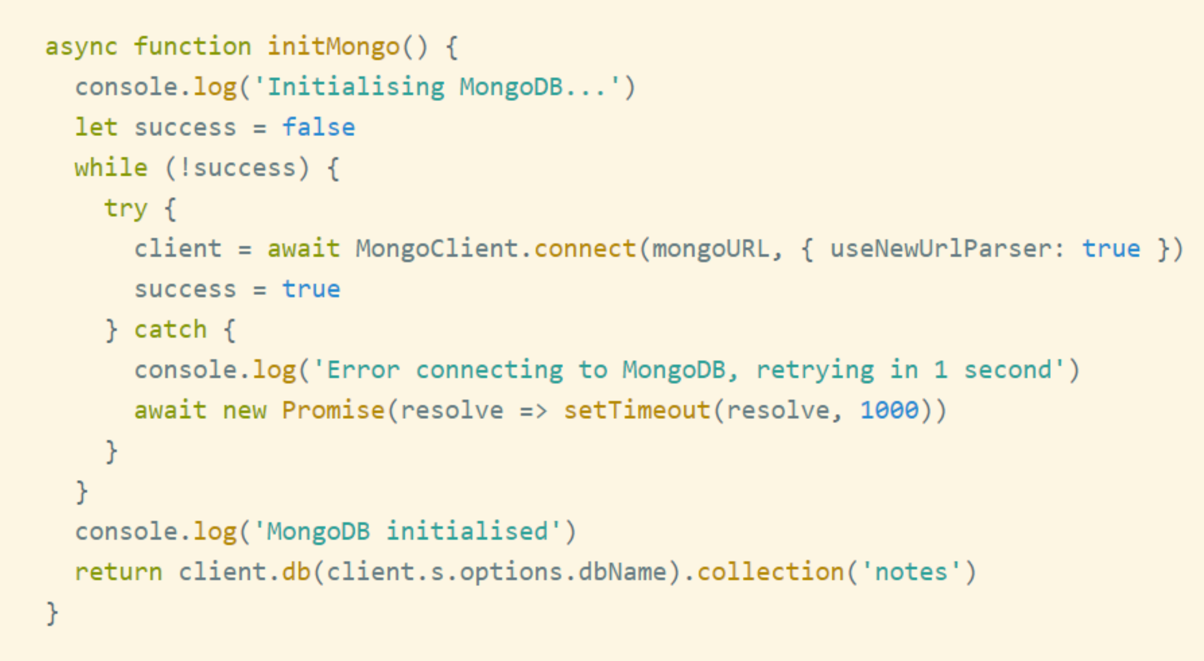
2) Navigate to the knote-js app folder

- **cd knote-js/01**

3) Create a Docker file inside the folder with the following contents

```
FROM node:12.0-slim
RUN mkdir /test
WORKDIR /test
COPY . /test/
RUN npm install
CMD [ "node", "index.js" ]
```

- This function in Javascript app keeps trying to connect to a MongoDB database at the given URL until the connection succeeds.



```
async function initMongo() {
  console.log('Initialising MongoDB...')
  let success = false
  while (!success) {
    try {
      client = await MongoClient.connect(mongoURL, { useNewUrlParser: true })
      success = true
    } catch {
      console.log('Error connecting to MongoDB, retrying in 1 second')
      await new Promise(resolve => setTimeout(resolve, 1000))
    }
  }
  console.log('MongoDB initialised')
  return client.db(client.s.options.dbName).collection('notes')
}
```

4) Hence Deploy your app with a mongo dB database

Before creating the db, lets create a separate network.

- **docker network create knote**

5) Now run a mongodb docker container inside the created network.

- **docker run --name=mongo-demo --network=knote mongo**

6) Check if your container has created

- **docker ps**
- **docker logs mongo-demo**

7) Now run the Knote app with the docker image that was build

- **docker run --name=knote --network=knote -p 9011:3000 -e MONGO_URL=mongodb://mongo-demo:27017/dev knote**

Open your browser and hit http://localhost:9011 and you can see your app running.