

TASK-TIDE

Website for Google keep

TEAM MEMBERS:

- Sandeep K 21ADL065
- Sankar V 21ADL066

DOCKER

Docker is a platform that allows developers to build package, and run applications in lightweight, portable containers. It's used to ensure consistent environments across different systems, enabling easier deployment and scalability.

STEP 1: Listing the files in the directory to confirm the presence of essential files like docker-compose.yml and Dockerfile

Command: **ls**

STEP 2: Attempting to start a MongoDB container. If the image isn't found locally, Docker will download it.

Command: **docker run -d --name mongodb -p 27017:27017 mongo**

```
student@ITP-CC1-1:~$ ls
Anaconda3-2020.11-Linux-x86_64.sh  minikube_latest_amd64.deb
apple                               Music
shiva                              myfile3.tar
Desktop                             naveen
dockerfile                          newfile.gz
Documents                           Pictures
dockerfile                          Public
Downloads                            R
eclipse-workspace                  rstudio-1.2.1578-amd64.deb
file.tar                             snap
file.txt                             sparkfiles
file.txt.save                       studentcount.py
google-chrome-stable_current_amd64.deb  swe
ITM-Page                             templates
itn-mate                             Videos

student@ITP-CC1-1:~$ cd Desktop
student@ITP-CC1-1:~/Desktop$ ls
react-todo-master  react-todo-master.zip
student@ITP-CC1-1:~/Desktop$ cd ^C
student@ITP-CC1-1:~/Desktop$ cd react-todo-master
student@ITP-CC1-1:~/Desktop/react-todo-master$ ls
api  application  docker-compose.yml  README.md
student@ITP-CC1-1:~/Desktop/react-todo-master$ sudo su
[sudo] password for student:
(base) root@ITP-CC1-1:~/home/student/Desktop/react-todo-master# docker run -d --name mongodb -p 27017:27017 mongo
Unable to find image 'mongo:latest' locally
latest: Pulling from library/mongo
3c645b31de29: Pull complete
bfa196f67a92: Pull complete
2eb794c1d8eb: Pull complete
3a415fad0b1a: Pull complete
ebbca0d8db42: Pull complete
7341c0351e26: Pull complete
11062490c406: Pull complete
e953fad04d18: Pull complete
Digest: sha256:f25b7229483f523d624c939118579a8cc2b37da62ee502e82888dd0a4d18ec4
Status: Downloaded newer image for mongo:latest
348dc5cf8bc04225741d0ffb234c11f89930baa2bd09d13066e0c8d14a02d722
(base) root@ITP-CC1-1:~/home/student/Desktop/react-todo-master# ls
api  application  docker-compose.yml  README.md
(base) root@ITP-CC1-1:~/home/student/Desktop/react-todo-master# cd api
(base) root@ITP-CC1-1:~/home/student/Desktop/react-todo-master/api# ls
coverage  dockerfile  jest.config.js  __mocks__  package.json  src  __tests__
(base) root@ITP-CC1-1:~/home/student/Desktop/react-todo-master/api# cat jest.config.js
module.exports = {
```

STEP 3:

- Building the frontend Docker image.
- Navigating to the appropriate directory and using the Dockerfile to create the image.

Command: **docker build -t frontend-app .**

```

module.exports = {
  verbose: true,
};(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/apl# ls
coverage  dockerfile  jest.config.js  __mocks__  package.json  src  __tests__
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/apl# nano dockerfile
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/apl# cat dockerfile
FROM node:10

WORKDIR /var/todo

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3333

CMD [ "npm", "run", "start" ]
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/apl# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master# tree

Command 'tree' not found, but can be installed with:

snap install tree # version 1.8.0+pkg-3fd6, or
apt install tree # version 1.8.0-1

See 'snap info tree' for additional versions.

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master# apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
 tree
0 upgraded, 1 newly installed, 0 to remove and 590 not upgraded.
Need to get 43.0 kB of archives.
After this operation, 115 kB of additional disk space will be used.
Get:1 http://in.archive.ubuntu.com/ubuntu focal/universe amd64 tree amd64 1.8.0-1 [43.0 kB]
Fetched 43.0 kB in 1s (56.9 kB/s)
Selecting previously unselected package tree.
(Reading database ... 266993 files and directories currently installed.)
Preparing to unpack .../tree_1.8.0-1_amd64.deb ...
Unpacking tree (1.8.0-1) ...
Setting up tree (1.8.0-1) ...
Processing triggers for man-db (2.9.1-1) ...

```

```
Activities [ Terminal ] Apr 25, 11:59 AM •
root@ITP-C11-1: /home/student/Desktop/react-todo-master

29 directories, 56 files
(base) root@ITP-C11-1:/home/student/Desktop/react-todo-master# ^C
(base) root@ITP-C11-1:/home/student/Desktop/react-todo-master# ls
api application docker-compose.yml README.md
(base) root@ITP-C11-1:/home/student/Desktop/react-todo-master# cd application
(base) root@ITP-C11-1:/home/student/Desktop/react-todo-master/application# docker run -d --name mongoddb -p 27017:27017 mongo
docker: Error response from daemon: Conflict. The container name "/mongoddb" is already in use by container "348dc5cfabc0425741d0ffbb234c1f8993bba2bd69d13066e8cd14a027d22". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
(base) root@ITP-C11-1:/home/student/Desktop/react-todo-master/application# docker build -t frontend-app .
[+] Building 248.4s (10/10) FINISHED
=> [internal] load build definition from dockerfile 0.0s
=> transferring dockerfile: 150B 0.0s
=> [internal] load .dockerignore 0.0s
=> transferring context: 2B 0.0s
=> [internal] load metadata for docker.io/library/node:18 0.0s
[1/5] FROM docker.io/library/node:18@sha256:59531d283edd5161cf9512f 200.35s
=> resolve docker.io/library/node:18@sha256:59531d283edd5161cf9512f 0.0s
=> sha256:70b8ef87096fa726adbe8f073ef69b5604bac19 45.38MB / 45.38MB 61.3s
=> sha256:59531d283edd5161cf9512f9e095b1836f7a1fcb0ab73 776B / 776B 0.0s
=> sha256:680e08593f82f8bf641e1627549b9cd0ad74f222 2.21KB / 2.21KB 0.0s
=> sha256:20e46e642d302aeebdc16101d0996f001e0d0e92e 7.83KB / 7.83KB 0.0s
=> sha256:2e2baf8ea0f4d509cc1024908f72686ea662e437 11.29MB / 11.29MB 11.6s
=> sha256:b53c1fd2746e8d2037f1b0b91dde0cc7411eb3e 4.34MB / 4.34MB 200.9s
=> sha256:84a83c1bd5887cc4ba9e1f286f9de9e31ce12dba9 49.79MB / 49.79MB 32.2s
=> sha256:7a803dc0b40fcd10faee3fb3ebb2d7aa8580 214.35MB / 214.35MB 139.6s
=> extracting sha256:70b8ef87096fa726adbe8f073ef69b5604bac19474c5cce 0.8s
=> sha256:680e08593f82f8bf641e1627549b9cd0ad74f222 4.19KB / 4.19KB 60.2s
=> extracting sha256:2e2baf8ea0f4d509cc1024908f72686ea662e437f10e9598 0.2s
=> sha256:0da9f9f60d485c74d153bfc7562f34533550e3dd 21.42MB / 21.42MB 89.8s
=> sha256:04dcce9934cf0daab9d303fe73b3bcb4ed84329d9d 2.48MB / 2.48MB 92.3s
=> sha256:73269890f9f6184d0fdba0afe3cf97b0c8dbe1e7a07 294B / 294B 92.6s
=> extracting sha256:b53c1fd2746e8d2037f1b0b91dde0cc7411eb3e5949f1 0.0s
=> extracting sha256:84a83c1bd5887cc4ba9e1f286f9de9e31ce12dba9f6813ef1 0.9s
=> extracting sha256:7a803dc0b40fcd10faee3fb3ebb2d7aa8580052e0e2952 0.0s
=> extracting sha256:b800e94e77303e276b8fb4911a40bfe28f46180d997022c69 0.0s
=> extracting sha256:0da9f9f60d485c74d153bfc7562f34533550e3ddbaa78f5e 0.6s
=> extracting sha256:04dcce9934cf0daab9d303fe73b3bcb4ed84329d9d562049 0.0s
=> extracting sha256:73269890f9f6184d0fdba0afe3cf97b0c8dbe1e7a07f7f 0.0s
[internal] load build context 0.0s
=> transferring context: 27.30kB 0.0s
[2/5] WORKDIR /var/todo 0.1s
[3/5] COPY package*.json . 0.0s
[4/5] RUN npm install 36.7s
[5/5] COPY . . 0.0s
```

STEP 4: Running the frontend application in Docker, mapping port **8080** on the host to port **3000** in the container

Command: **docker run -d -p 8080:3000 frontend-app**



STEP 5: Checking Docker Network Configuration

Command: **docker network inspect bridge**

```
Activities Terminal Apr 25 11:59 AM
root@ITP-CC1-1: /home/student/Desktop/react-todo-master

=> exporting to image
=> exporting layers
=> writing image sha256:20178eac90f3d4e934870f41a901dfe4c3d277f3d 0.0s
=> saving to registry
root@ITP-CC1-1: /home/student/Desktop/react-todo-master# docker run -d -p 8080:3000 frontend-app
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# ls
dockerfile package.json public src
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# cat dockerfile
FROM node:18

WORKDIR /var/todo

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3000

CMD ["npm", "start"]
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# ^C
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# ^C
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# docker run -d -p 8080:3000 frontend-app
6aaa73f249f35f407edddeb9c3a79ec974e9b35bb29a25ddab774ed30554075
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/application# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master# ls
api application docker-compose.yml README.md
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master# cd api
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# docker network inspect my-network
[]
Error response from daemon: network my-network not found
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# docker network inspect bridge
[
  {
    "Name": "bridge",
    "Id": "44982374878c05a1525847b7776783c96dbe1193856987e77b545a5ee319f050",
    "Created": "2024-04-25T10:43:55.530495037+05:30",
    "Scope": "local",
    "Driver": "bridge",
    "EnableIPv6": false,
    "IPAM": {
      "Driver": "default",
      "Options": null,
      "Config": [
        {
          "Subnet": "172.17.0.0/16"
        }
      ]
    }
  }
]
```

```
root@ITP-CC1-1: /home/student/Desktop/react-todo-master

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# ls
core enums functions index.js models repositories routes types.d.ts
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# cat index.js
const express = require('express');
const bodyParser = require('body-parser');
const Database = require('../core/Database');
const Routes = require('../core/Routes');
const { PORT } = require('../enums/commonEnums');

const app = express();
new Database().connect();

// Added for temporary another solution will be checked later
app.use((request, response, next) => {
  response.header('Access-Control-Allow-Origin', '*');
  response.header('Access-Control-Allow-Methods', 'GET, PUT, POST, DELETE');
  response.header('Access-Control-Allow-Headers', 'Origin, X-Requested-With, Content-Type, Accept');
  next();
});
app.use(bodyParser.json());
app.use('/', Routes);
app.listen(PORT);

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# ^C
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# ls
coverage dockerfile jest.config.js mocks package.json src tests
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# cd src
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# ls
enums functions index.js models repositories routes types.d.ts
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# cd functions
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src/functions# ls
index.js
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src/functions# cat index.js
/**
 * @param {*} value
 * @returns {boolean}
 */
const isNull = value => value === null;

module.exports = {
  isNull,
};
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src/functions# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
cat: nano: No such file or directory
PORT=3000
DB=mongodb://127.17.0.2:27017

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# ls
coverage dockerfile jest.config.js mocks package.json src tests
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# cat dockerfile
FROM node:10

WORKDIR /var/todo

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3333

CMD [ "npm", "run", "start" ]
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano dockerfile
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
Use "fg" to return to nano.
[!]+ Stopped nano .env
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# cat .env
PORT=3333
DB=mongodb://127.17.0.2:27017

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# docker build -t backend-app .
[*] Building 19.3s (10/10) FINISHED
[+] backend-app: Read build information from Dockerfile
```

STEP 6: Building the Docker image for the backend application.

Command: **docker build -t backend-app .**

```
root@ITP-CC1-1: /home/student/Desktop/react-todo-master

* @returns {boolean}
*/
const isNull = value => value === null;

module.exports = {
  isNull,
};
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src/functions# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api/src# cd ..
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
cat: nano: No such file or directory
PORT=3000
DB=mongodb://127.17.0.2:27017

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# ls
coverage dockerfile jest.config.js mocks package.json src tests
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# cat dockerfile
FROM node:10

WORKDIR /var/todo

COPY package*.json ./

RUN npm install

COPY . .

EXPOSE 3333

CMD [ "npm", "run", "start" ]
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano dockerfile
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
Use "fg" to return to nano.
[!]+ Stopped nano .env
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# nano .env
(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# cat .env
PORT=3333
DB=mongodb://127.17.0.2:27017

(base) root@ITP-CC1-1:/home/student/Desktop/react-todo-master/api# docker build -t backend-app .
[*] Building 19.3s (10/10) FINISHED
[+] backend-app: Read build information from Dockerfile
```

STEP 7:Running the backend application in Docker, mapping port **3002** on the host to port **3333** in the container

Command: **docker run -d -p 3002:3333 backend-app**

STEP 8: Listing the running Docker containers to confirm everything is running properly.

Command: **docker ps**

STEP 10: Building Docker Services

- Ensuring whether the Docker Compose setup is correct, and then building the services specified in the docker-compose.yml file

Command: **docker compose up**

```
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# docker rm mongodb-mongod
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# nano docker-compose.yml
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# nano docker-compose.yml
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# docker compose up
validating /home/student/Desktop/react-todo-master/docker-compose.yml: services.container_name must be a mapping
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# nano docker-compose.yml
(base) root@ITP-CCI-1:/home/student/Desktop/react-todo-master# docker compose up
(-) Running yml
✓ Container mongodbg_serve created OK
✓ Container api created OK
✓ Container application created OK
Attaching to api, application, mongodbg_serve
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.311+00:00"}, {"$": "I", "c": "CONTROL", "id": 123285, "ctx": "main", "msg": "Automatically disabling TLS 1.0, to force-en"}
able TLS 1.0 specify --sslDisabledProtocols 'none'"}
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.311+00:00"}, {"$": "I", "c": "NETWORK", "id": 14915701, "ctx": "main", "msg": "Initialized wire specification", "attr": {"spe"}
{"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 21}, "incomingInternalClient": {"minWireVersion": 10, "maxWireVersion": 21}, "outgoing": {"minWireVersion": 6,
maxWireVersion": 21}, "isInternalClient": true}}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.316+00:00"}, {"$": "I", "c": "NETWORK", "id": 14648001, "ctx": "main", "msg": "Implicit TCP FastOpen unavailable. If TCP Fa}
stoppen is required, see tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.319+00:00"}, {"$": "I", "c": "REPL", "id": 15123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService",
"attr": {"service": "TenantMigrationDonorService", "namespace": "config.tenantMigrationDonors"}}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.319+00:00"}, {"$": "I", "c": "REPL", "id": 15123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService",
"attr": {"service": "TenantMigrationAccepterService", "namespace": "config.tenantMigrationAccepters"}}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.319+00:00"}, {"$": "I", "c": "CONTROL", "id": 15945603, "ctx": "main", "msg": "Multithreading initialized"}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.319+00:00"}, {"$": "I", "c": "TENANT_M", "id": 70916000, "ctx": "main", "msg": "Starting TenantMigrationAccessBlockerRegistr}
y"}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.320+00:00"}, {"$": "I", "c": "CONTROL", "id": 14615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 1,
port": 27017, dbPath: "/data/db", architecture: "64-bit", host": "5read7fbcce"}]}
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.320+00:00"}, {"$": "I", "c": "CONTROL", "id": 123403, "ctx": "initandlisten", "msg": "Build info", "attr": {"buildInfo": {"v"}
ersion": "7.0.8", gitVersion": "c5d3e5ba3bd8ae274bf65ec4e5533bd9744a64", opensslVersion": "OpenSSL 3.0.2 15 Mar 2022", modules": [], allocator: "tcmalloc", environme"}
nt": {"distro": "ubuntu", distarch: "x86_64", target_arch: "x86_64"}}]}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.320+00:00"}, {"$": "I", "c": "CONTROL", "id": 151765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"na}
me": "Ubuntu", version": "22.04"}}]}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.320+00:00"}, {"$": "I", "c": "CONTROL", "id": 121951, "ctx": "initandlisten", "msg": "Options set by command line", "attr"}
{("options": {"net": {"bindip": ""}})}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.321+00:00"}, {"$": "I", "c": "STORAGE", "id": 122299, "ctx": "initandlisten", "msg": "Using the XFS filesystem is strongl}
y recommended with the WiredTiger storage engine. See http://docs.mongodb.org/coproductates/filesystem", "tags": ["startupWarnings"]}]}]
mongodbg_serve | [{"$date": "2024-04-25T06:27:25.321+00:00"}, {"$": "I", "c": "STORAGE", "id": 122011, "ctx": "initandlisten", "msg": "Opening WiredTiger", "attr": {"confi}
g": {"create_cache_line_size: 16384, create_collection_base_files_per_seg: 10, create_statistics_fast: 1, log_enable_wiredtiger_remove_true_path_journal_conf}
essor_snappy, buffer_in_extension_config({zstd: {compression_level: 6}}, file_manager: {close_idle_time: 60, close_scan_interval: 10, close_handle_minium: 2000}, statistics_log: {
wait: 0}, json_output: {error, message}, verbose: {recovery_progress: 1, checkpoint_progress: 1, compact_progress: 1, backup0_checkpoint: 0, compact0_evict: 0, history_store: 0, recove}
ry_0_rts: 0, salvage: 0, tiered: 0, timestamp: 0, transaction: 0, verify: 0, log: 0, ...}}]}]
```

```

root@ITP-CC1-1: /home/student/Desktop/react-todo-master
root@ITP-CC1-1: /home/student/Desktop/react-todo-master
student@ITP-CC1-1: /Desktop/react-todo-master
student@ITP-CC1-1: ~/Desktop/react-todo-master

(base) root@ITP-CC1-1: /home/student/Desktop/react-todo-master# nano docker-compose.yml
(base) root@ITP-CC1-1: /home/student/Desktop/react-todo-master# docker compose up
validating /home/student/Desktop/react-todo-master/docker-compose.yml: services.container_name must be a mapping
(base) root@ITP-CC1-1: /home/student/Desktop/react-todo-master# nano docker-compose.yml
(base) root@ITP-CC1-1: /home/student/Desktop/react-todo-master# docker compose up
[+] Running 3/0
✔ Container mongodb_server created                                0.0m
✔ Container api created                                             0.0m
✔ Container application created                                    0.0m
Attaching to api, application, mongodb_server
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.311+00:00"},"s":1,"c":["CONTROL","id":"23285","ctx":"main","msg":"Automatically disabling TLS 1.0, to force-en
able TLS 1.0 specify --sslDisabledProtocols 'none'"]}
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.311+00:00"},"s":1,"c":["NETWORK","id":"4915701","ctx":"main","msg":"Initialized wire specification","attr":{"spe
c":{"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":21},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":121},"outgoing":{"minWireVersion":0,
maxWireVersion":121},"isInternalClient":true}}]}
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.319+00:00"},"s":1,"c":["NETWORK","id":"4648001","ctx":"main","msg":"Implicit TCP Fastopen unavailable. If TCP Fa
stopen is required, see tcpFastOpenServer, tcpFastOpenClient, and tcpFastOpenQueueSize."}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.319+00:00"},"s":1,"c":["REPL","id":"1523008","ctx":"main","msg":"Successfully registered PrimaryOnlyService."}
"attr":{"service":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.319+00:00"},"s":1,"c":["REPL","id":"1523008","ctx":"main","msg":"Successfully registered PrimaryOnlyService."}
"attr":{"service":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.319+00:00"},"s":1,"c":["CONTROL","id":"5945603","ctx":"main","msg":"Multi threading initialized"}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.319+00:00"},"s":1,"c":["TENANT_M","id":"7691600","ctx":"main","msg":"Starting TenantMigrationAccessBlockerRegistr
y"}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.320+00:00"},"s":1,"c":["CONTROL","id":"4615611","ctx":"initandlisten","msg":"MongoDB starting","attr":{"pid":1,
port":27017,"dbPath":"/data/db","architecture":"64-bit","host":"5teeed7fbcce"}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.320+00:00"},"s":1,"c":["CONTROL","id":"23403","ctx":"initandlisten","msg":"Build Info","attr":{"buildInfo":{"v
ersion":"7.0.8","gitVersion":"c5d33e53ba3d98be2f4876sec4e5338d67a4e64","opensslVersion":"OpenSSL 3.0.2 15 Mar 2022","modules":[],"allocator":"tcmalloc","environment":{"
distro":"ubuntu2204","distarch":"x86_64","target_arch":"x86_64"}}}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.320+00:00"},"s":1,"c":["CONTROL","id":"51765","ctx":"initandlisten","msg":"Operating System","attr":{"os":{"na
me":"Ubuntu","version":"22.04"}}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.320+00:00"},"s":1,"c":["CONTROL","id":"21951","ctx":"initandlisten","msg":"Options set by command line","attr
":{"options":{"net":{"bindip":"","}}}]}
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.321+00:00"},"s":1,"c":["STORAGE","id":"22297","ctx":"initandlisten","msg":"Using the XFS filesystem is strongl
y recommended with the WiredTiger storage engine. See http://docs.mongodb.org/manual/products/filesystem","tags":{"startupWarnings":1}}]
mongodb_server | {"t":{"$date":"2024-04-25T06:27:25.321+00:00"},"s":1,"c":["STORAGE","id":"22315","ctx":"initandlisten","msg":"Opening WiredTiger","attr":{"config
":{"create,cache-size=3353M,session_max=31000,evictions:(threads_min=4,threads_max=4),config_base=false,statistics=(fast),log(enabled=true,remove=true,path.journal.com
pressor=snappy),builtin_extension_config=(zstd-compression-level=6),file_manager=(close_idle_time=000,close_scan_interval=10,close_handle_minimum=2000),statistics_log(=
{"enabled":0,honour_abort_level=0}),verbose=(quiet,error_progress=1),checkpoint_progress=1,compact_progress=1,backup=0,checkpoint=0,compaction=0,evict=0,historical_store=0,recove
ry=0,rts=0,salvage=0,tiered=0,timestampt=0,transaction=0,verify=0,log=0,...}}]}
Error response from daemon: driver failed programming external connectivity on endpoint application [3dcdecab0feef7a628bbeec973dc9cc5bb7bfac5299ae7fb84b4cee02df2]:
Bind for 0.0.0.0:8080 failed: port is already allocated
(base) root@ITP-CC1-1: /home/student/Desktop/react-todo-master# sudo netstat -tuln | grep 8080

```

Todos

Add Todo

complete the task

KUBERNETES

Kubernetes is an orchestration system for managing containerized applications, providing tools for scaling, balancing loads, and automating deployment and recovery.

STEP 11: Starting Minikube to set up a local Kubernetes cluster.

Command: **minikube start**

```
student@ITP-CC1-1:~$ sudo su
[sudo] password for student:
(base) root@ITP-CC1-1:/home/student# docker ps -aq
64f9eb5f6d5f
276873d5253b
3b7ea98bf475
(base) root@ITP-CC1-1:/home/student# cd Desktop
(base) root@ITP-CC1-1:/home/student/Desktop# ls
(base) root@ITP-CC1-1:/home/student/Desktop# cd ..
(base) root@ITP-CC1-1:/home/student# cd project
(base) root@ITP-CC1-1:/home/student/project# ls
docker_nodejs_mongodb
(base) root@ITP-CC1-1:/home/student/project# cd docker_nodejs_mongodb
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb# ls
deployment.yaml docker-compose.yml frontend README.md server service.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb# cd frontend
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/frontend# nano frontend-deployment.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/frontend# docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS                               NAMES
64f9eb5f6d5f   docker_nodejs_mongodb_server        "docker-entrypoint.s..." About an hour ago Up About an hour 0.0.0.0:3000->3000/tcp, :::3000->3000/tcp   server
276873d5253b   mongo:4.2.0                         "docker-entrypoint.s..." About an hour ago Up About an hour 0.0.0.0:27017->27017/tcp, :::27017->27017/tcp   mongo
3b7ea98bf475   docker_nodejs_mongodb_frontend      "docker-entrypoint.s..." About an hour ago Up About an hour 0.0.0.0:3002->3002/tcp, :::3002->3002/tcp   frontend
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/frontend# nano frontend-service.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/frontend# cd ..
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb# cd service
bash: cd: service: No such file or directory
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb# cd server
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/server# nano backend-deployment.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/server# nano backend-deployment.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/server# nano backend-service.yaml
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb/server# cd ..
(base) root@ITP-CC1-1:/home/student/project/docker_nodejs_mongodb# exit
exit
student@ITP-CC1-1:~$ minikube start
🐳 minikube v1.12.0 on Ubuntu 20.04
🔧 Using the virtualbox driver based on existing profile
👉 Starting control plane node minikube in cluster minikube
🔄 Updating the running virtualbox "minikube" VM ...
🔧 Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
🔧 Verifying Kubernetes components...
```

STEP 12: Applying Kubernetes Deployments

- Deploying Kubernetes resources using kubectl apply with the respective deployment files.

Command:

- **kubectl apply -f frontend-deployment.yaml**
- **kubectl apply -f backend-deployment.yaml**

STEP 13: Applying Kubernetes Services

- Similarly, applying the corresponding service configurations to expose the deployments.

Command:

- **kubectl apply -f frontend-service.yaml**
- **kubectl apply -f backend-service.yaml**


```

student@ITP-CC1-1: ~/project/docker_nodejs_mongodb
root@ITP-CC1-1: /home/student/project/docker_nodejs_mongodb

minikube v1.32.0 on ubuntu 20.04
Using the virtualbox driver based on existing profile
Starting control plane node minikube in cluster minikube
Updating the running virtualbox "minikube" VM ...
Preparing Kubernetes v1.28.3 on Docker 24.0.7 ...
Verifying Kubernetes components...
  ■ Using image registry.k8s.io/nginx-kube-webhook-certgen:v20231011-8b53cabe0
  ■ Using image registry.k8s.io/nginx-kube-webhook-certgen:v20231011-8b53cabe0
  ■ Using image registry.k8s.io/nginx-kube-webhook-certgen:v20231011-8b53cabe0
  ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  ■ Using image docker.io/kubernetes/dashboard:v2.7.0
  ■ Using image docker.io/kubernetes/metrics-scraper:v1.0.8
Verifying Ingress addon...
Some dashboard features require the metrics-server addon. To enable all features please run:
minikube addons enable metrics-server

Enabled addons: storage-provisioner, dashboard, default-storageclass, ingress
Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
student@ITP-CC1-1: $ cd project
student@ITP-CC1-1: ~/project $ cd docker_nodejs_mongodb
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb $ cd frontend
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/frontend $ kubectl apply -f frontend-deployment.yaml
deployment.apps/frontend created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/frontend $ kubectl apply -f frontend-service.yaml
service/frontend-service created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/frontend $ cd ..
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb $ cd server
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl apply -f backend-service.yaml
service/backend-service created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl get deployments

```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
backend	0/1	1	0	51s
frontend	0/1	1	0	97s
lltn-deployment	1/1	1	1	11d
mongo-learn	2/2	2	2	16d
nyapp-deployment	0/3	1	0	40m
nginx-deployment	2/2	1	2	18d
nginx-sru	2/2	2	2	18d
nginx	0/1	1	0	74m

```

student@ITP-CC1-1: ~/project/docker_nodejs_mongodb
root@ITP-CC1-1: /home/student/project/docker_nodejs_mongodb

student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/frontend $ kubectl apply -f frontend-service.yaml
service/frontend-service created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/frontend $ cd ..
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb $ cd server
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl apply -f backend-service.yaml
service/backend-service created
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl get deployments

```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
backend	0/1	1	0	51s
frontend	0/1	1	0	97s
lltn-deployment	1/1	1	1	11d
mongo-learn	2/2	2	2	16d
nyapp-deployment	0/3	1	0	40m
nginx-deployment	2/2	1	2	18d
nginx-sru	2/2	2	2	18d
nginx	0/1	1	0	74m

```

student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ kubectl get services

```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
backend-service	ClusterIP	10.107.165.57	<none>	80/TCP	64s
frontend-service	NodePort	10.105.183.166	<none>	80:31199/TCP	108s
lltn-service	LoadBalancer	10.107.164.129	<pending>	4000:32689/TCP	11d
kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	18d
mongo-learn-service	LoadBalancer	10.99.228.176	<pending>	3000:31331/TCP	16d
mongolearn-service	LoadBalancer	10.105.238.65	<pending>	3000:30656/TCP	16d
nyapp-service	NodePort	10.109.132.74	<none>	3002:30520/TCP	41m
nginx	ClusterIP	10.102.33.186	<none>	80/TCP	17d
nginx-lg	LoadBalancer	10.108.230.252	<pending>	80:31093/TCP	17d

```

student@ITP-CC1-1: ~/project/docker_nodejs_mongodb/server $ cd ..
student@ITP-CC1-1: ~/project/docker_nodejs_mongodb $ kubectl get deployments

```

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
backend	0/1	1	0	87s
frontend	0/1	1	0	2m13s
lltn-deployment	1/1	1	1	11d
mongo-learn	2/2	2	2	16d
nyapp-deployment	0/3	1	0	41m
nginx-deployment	2/2	1	2	18d
nginx-sru	2/2	2	2	18d
nginx	0/1	1	0	74m

STEP 14: Verifying Kubernetes Deployments and Services

- Checking the status of the deployments and services to ensure that they are running as expected.
- List Deployments - This command shows the readiness, availability, and age of each deployment. Look for READY, UP-TO-DATE, and AVAILABLE fields to assess their status.

Command: **kubectl get deployments**

- List Services - This command lists the services and their associated ports, cluster IPs, and external IPs. Check for any pending external IPs or port conflicts.

Command: **kubectl get services**

STEP 15: Monitoring Minikube and Kubernetes Components

- Ensuring whether the Minikube and the Kubernetes components are running smoothly by checking the cluster status.

Command: **kubectl get nodes -o wide**

This command provides information about the nodes in your Kubernetes cluster, including their internal IP, external IP, and other configurations.

```
student@ITP-CC1-11: /project/docker_nodejs_mongodb$ cd server
student@ITP-CC1-11: /project/docker_nodejs_mongodb/server$ kubectl apply -f backend-deployment.yaml
deployment.apps/backend created
student@ITP-CC1-11: /project/docker_nodejs_mongodb/server$ kubectl apply -f backend-service.yaml
service/backend-service created
student@ITP-CC1-11: /project/docker_nodejs_mongodb/server$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend       1/1     1             0           8s
frontend      1/1     1             0           93s
litn-deploy   1/1     1             1          11d
mongo-learn   2/2     2             2          16d
nyapp-deploy  0/3     1             0          40m
nginx-deploy  2/2     1             2          18d
nginx-sru     2/2     2             2          18d
nginx         0/1     1             0          74m
student@ITP-CC1-11: /project/docker_nodejs_mongodb/server$ kubectl get services
NAME          TYPE          CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
backend-service  ClusterIP     10.107.105.57 <none>        80/TCP           64s
frontend-service NodePort       10.105.183.166 <none>        80:31199/TCP     108s
litn-service    LoadBalancer 10.107.164.129 <pending>     4000:32689/TCP   11d
kubernetes      ClusterIP     10.96.0.1     <none>        443/TCP          18d
mongo-learn-service LoadBalancer 10.99.228.176 <pending>     3000:31331/TCP   16d
mongolearn-service LoadBalancer 10.105.238.65 <pending>     3000:30656/TCP   16d
nyapp-service   NodePort       10.109.132.74 <none>        3002:30520/TCP   41m
nginx           ClusterIP     10.102.33.186 <none>        80/TCP          17d
nginx-ig        LoadBalancer 10.108.230.252 <pending>     80:31093/TCP     17d
student@ITP-CC1-11: /project/docker_nodejs_mongodb/server$ cd ..
student@ITP-CC1-11: /project/docker_nodejs_mongodb$ kubectl get deployments
NAME          READY   UP-TO-DATE   AVAILABLE   AGE
backend       1/1     1             0           87s
frontend      1/1     1             0          2m13s
litn-deploy   1/1     1             1          11d
mongo-learn   2/2     2             2          16d
nyapp-deploy  0/3     1             0          41m
nginx-deploy  2/2     1             2          18d
nginx-sru     2/2     2             2          18d
nginx         0/1     1             0          74m
student@ITP-CC1-11: /project/docker_nodejs_mongodb$ kubectl get nodes -o wide
NAME        STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE             KERNEL-VERSION   CONTAINER-RUNTIME
minikube   Ready    control-plane  18d   v1.28.3   192.168.59.108 <none>        Bullseye 2021.02.12 5.10.57             docker://24.0.7
student@ITP-CC1-11: /project/docker_nodejs_mongodb$
```

STEP 16: Creating Ingress Resource

Applying an Ingress resource to route external traffic to the services.

Command: **kubectl apply -f ingress.yaml**

```
... student@ITP-CC1-11: /project/docker_nodejs_mongodb$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/todo-task created
```

STEP 17: Displaying a list of services running in a Minikube Kubernetes cluster.

Command: **minikube service list**

STEP 18: Mapped domain names to specific IP addresses can be viewed

Command: **sudo gedit /etc/hosts**

```
front-end-service NodePort 10.109.87.99 <none> 80:30340/TCP 117n
kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 3h4m
student@ITP-CC1-1: ~/Desktop/react-todo-master$ curl www.todo-task.com
<!doctype html><html lang="en"><head><meta charset="utf-8"/><link rel="icon" href="/favicon.ico"/> <link
href="https://cdnis.cl *https://cdnis loudflare.com/ajax/libs/flowbite/1.6.5/flowbite.min.css" rel="styleshee
```

Todos