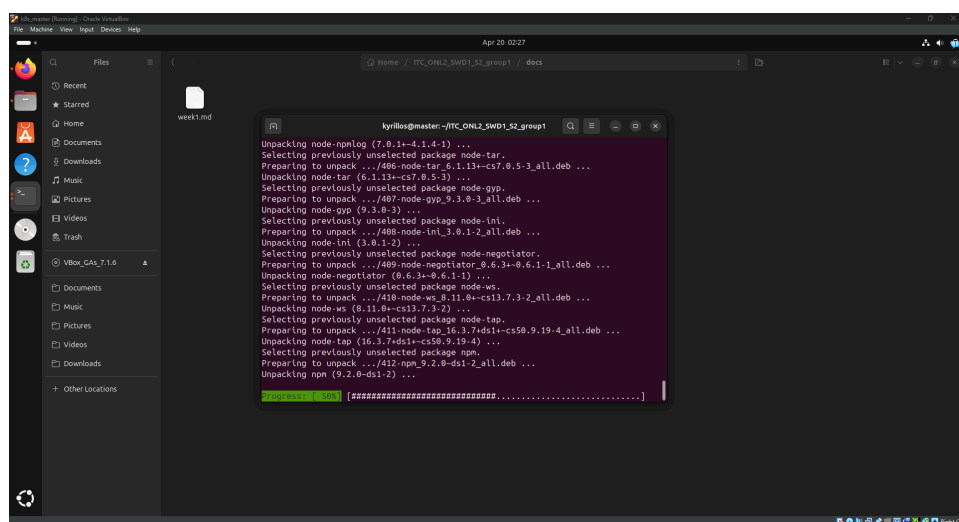


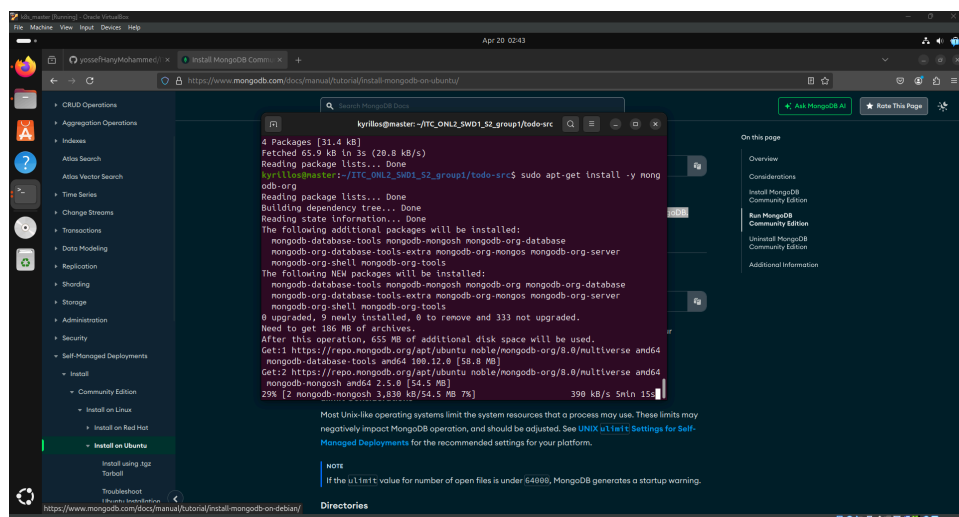
# Week 1: Environment Setup, App Initialization & Dockerization

1. Set up the local development environment: Install and configure necessary DevOps tools (manual)
2. Choose and prepare a simple project to work with
3. Clone the project repo locally
4. Set up GitHub and classic token
5. Reload the local packages using `sudo apt update`
6. Install project requirements and dependencies
7. Install npm on machine



A screenshot of a terminal window running on a virtual machine. The terminal shows the output of the command `sudo apt update`. The output lists various packages being updated, including `node-npmlog`, `node-tar`, `node-gyp`, `node-lint`, `node-negotiator`, `node-ws`, `node-tap`, `node-npm`, and `node-npm`. The terminal also shows the output of `sudo apt install -y mongodb`, which encounters an error: "Package 'mongodb' 8. has no installation candidate".

8. `cd todo-src`
9. `npm install`
10. Try `sudo apt install -y mongodb` but encountered an error (Package 'mongodb' 8. has no installation candidate)
11. Head to MongoDB official documentation for installation guide:  
<https://www.mongodb.com/docs/manual/tutorial/install-mongodb-on-ubuntu/>



A screenshot of a terminal window running on a virtual machine. The terminal shows the output of the command `sudo apt-get install -y mongodb`. The output lists various packages being installed, including `mongodb-database-tools`, `mongodb-mongosh`, `mongodb-org-database`, `mongodb-org-database-tools-extra`, `mongodb-org-mongos`, `mongodb-org-server`, `mongodb-org-shell`, and `mongodb-org-tools`. The terminal also shows the output of `sudo apt-get install -y mongodb`, which encounters an error: "Package 'mongodb' 8. has no installation candidate".

From a terminal, install `gnupg` and `curl` if they are not already available:

```
sudo apt-get install gnupg curl
```

To import the MongoDB public GPG key, run the following command:

```
curl -fsSL https://www.mongodb.org/static/pgp/server-8.0.asc | \
sudo gpg -o /usr/share/keyrings/mongod
b-server-8.0.gpg \ --dearmor
```

Create the list file for Ubuntu 24.04 (Noble):

```
echo"deb [ arch=amd64,arm64 signed-by=/usr/share/keyrings/mongodb-server-8.0.gpg ] https://repo.m
ongodb.org/apt/ubuntu noble/mongodb-org/8.0 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-or
g-8.0.list
```

You can install the latest stable version of MongoDB **Community Server**:

```
sudo apt-get install -y mongodb-org
```

12. Install production dependencies (where `package.json` lives):

```
npm install bcrypt chalk cors dotenv express mongoose \
connect-mongodb-session express-session validator ejs --save

# Tidy up low-risk issues:
npm audit fix
```

13. Install development tools:

```
npm install nodemon eslint prettier eslint-config-prettier --save-dev
eslint --init
```

14. Add a dev-script to your `package.json` :

```
"dev": "nodemon -L server.js",
"start": "node server.js",
```

15. **Create a** `.env` (in the same folder as `src/index.js`):

```
MONGO_URI=mongodb://todo-database:27017/todo-app

SESSION_SECRET=my-super-secret

PORT=3000
```

16. Run MongoDB (already installed it):

```
sudo systemctl start mongod
```

17. Launch app in dev mode:

```
npm run dev
```

18. **Open** <http://localhost:3000> in browser.

19. Encountered this error:

```
kyrillos@master:~/ITC_ONL2_SWD1_S2_group1/todo-src$ npm run dev
> todoapp-backend@1.0.0 dev
> nodemon -L server.js

[nodemon] 3.1.9
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,cjs,json
[nodemon] starting 'node server.js'
express-session deprecated req.secret; provide secret option file:/home/kyrillos/ITC_ONL2_SWD1_S2_group1/todo-src/app.js:27:2
Connection to MongoDB failed!
MongooseError: The 'uri' parameter to 'openUri()' must be a string, got "undefined". Make sure the first parameter to 'mongoose.connect()' or 'mongoose.createConnection()' is a string.
Listening on port undefined
```

20. Try installing mongo using Docker Compose instead

21. First: Dockerfile for backend

```
FROM node:20.9-alpine

# 1st: Create app directory
RUN mkdir -p /usr/src/app
WORKDIR /usr/src/app

# 2nd: Copy package.json and package-lock.json to the working directory
COPY package*.json ./

# 3rd: Install npm
RUN npm install --silent

# 4th: Copy source code
COPY . .

# 5th: Exports
EXPOSE 8082

# 6th: Start node server
CMD ["npm","start"]
```

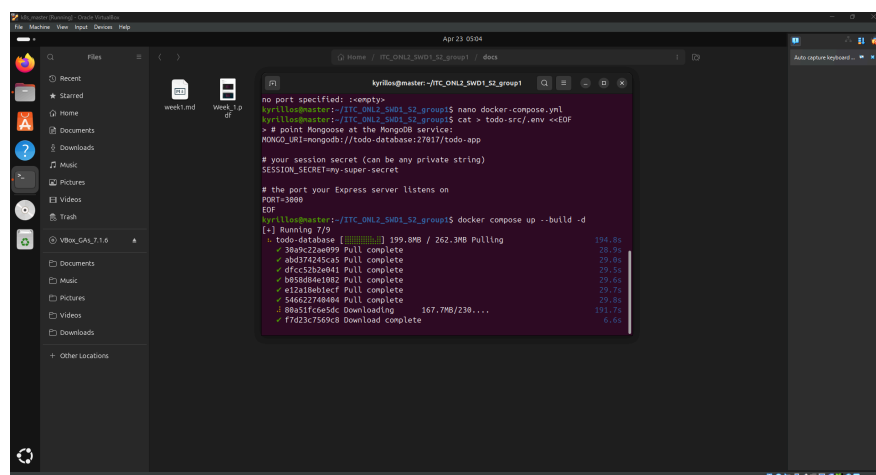
## 22. Then: docker-compose file

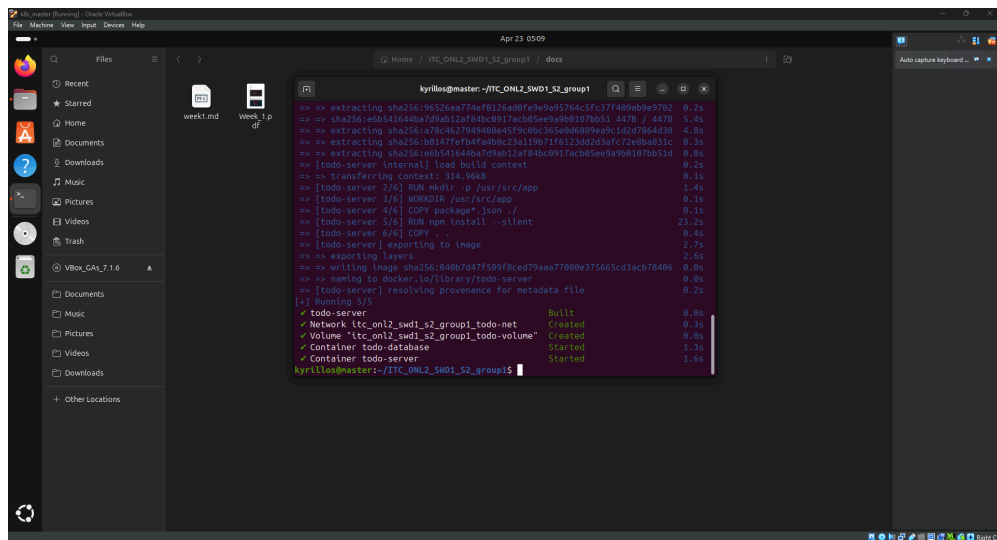
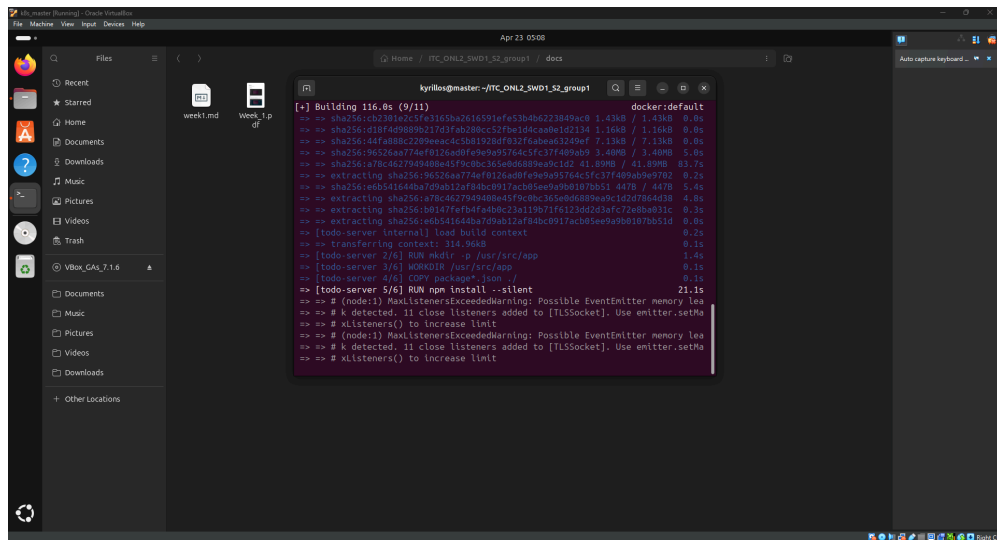
```
version: '3.9'
services:
  todo-database:
    image: mongo
    container_name: todo-database
    restart: always
    ports:
      - "27017:27017"
    volumes:
      - todo-volume:/data/db
    networks:
      - todo-net

  todo-server:
    build:
      context: ./todo-src
      dockerfile: Dockerfile
    image: todo-server
    container_name: todo-server
    restart: on-failure
    ports:
      - "3000:3000"
    env_file:
      - ./todo-src/.env
    depends_on:
      - todo-database
    networks:
      - todo-net

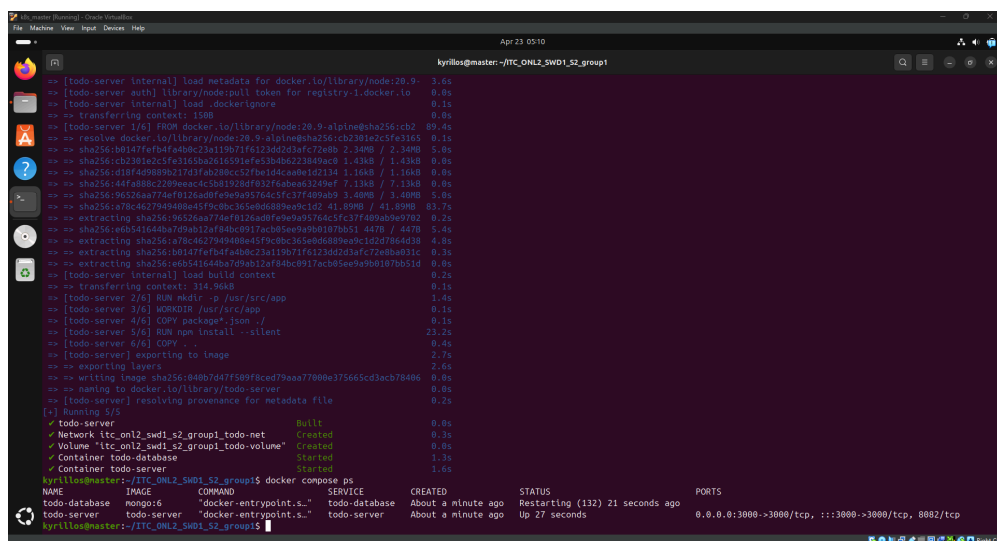
volumes:
  todo-volume:

networks:
  todo-net:
    driver: bridge
```





23. Verify the two containers have started using `docker compose ps`



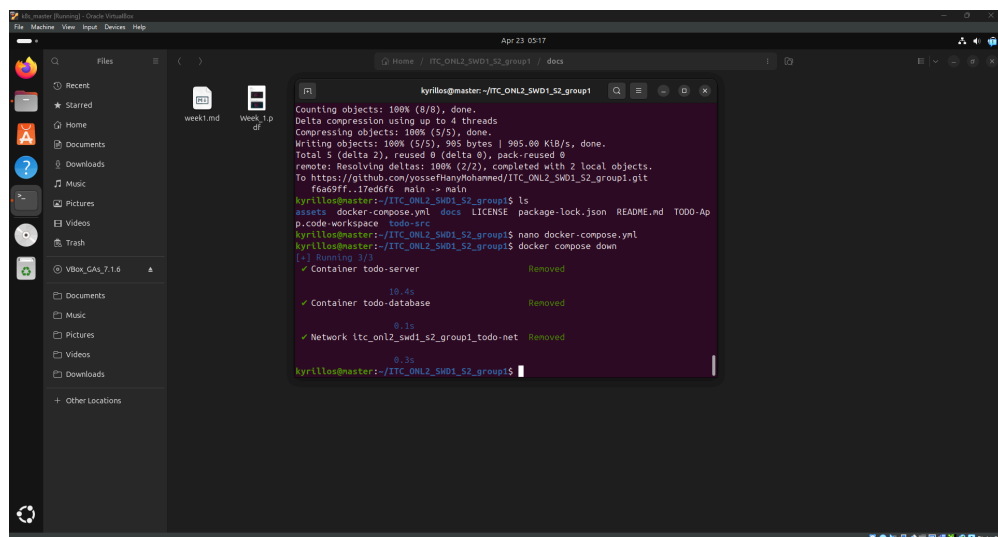
## 24. Problem database container exiting and restarting

```
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
do-database |
do-database | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
do-database |
do-database | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
do-database |
do-database | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
do-database |
do-database | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
do-database |
do-database | WARNING: MongoDB 5.0+ requires a CPU with AVX support, and your current system does not appear to have that!
do-database | see https://jira.mongodb.org/browse/SERVER-54407
do-database | see also https://www.mongodb.com/community/forums/t/mongodb-5-0-cpu-intel-g4650-compatibility/116610/2
do-database | see also https://github.com/docker-library/mongo/issues/485#issuecomment-891991814
```

25. "Restarting (132)" paired with that AVX warning means CPU (or VM) doesn't support the AVX instructions that Mongo 5+ now requires, so `mongod` is crashing immediately. The quickest fix is to fall back to a 4.x Mongo image that doesn't need AVX.

26. EDIT docker compose, first `docker compose down` then `docker compose up --build -d`

image: mongo:4.4



```

kyrillos@master:~/ITC_ONL2_SWD1_S2_group1
✓ Container todo-database Removed 0.1s
✓ Network ltc_onl2_sw1_s2_group1_todo-net Removed 0.3s
kyrillos@master:~/ITC_ONL2_SWD1_S2_group1$ docker compose up --build -d
[+] Running 9/9
  ✓ todo-database Pulled 90.7s
    ✓ d4c3c9e5f8 Pull complete 26.2s
    ✓ bca5893fe8bd Pull complete 26.3s
    ✓ 35ec036951f8 Pull complete 26.9s
    ✓ ddb77e597b02 Pull complete 27.2s
    ✓ 7ab9eb5a409d Pull complete 27.2s
    ✓ a6c1ba219414 Pull complete 27.3s
    ✓ 83b651df5384 Pull complete 87.5s
    ✓ e233f2d13609 Pull complete 87.6s
[+] Building 2.0s (13/13) FINISHED
  => [todo-server internal] load build definition from Dockerfile
  => [todo-server internal] load metadata for docker.io/library/node:28.9-alpine
  => [todo-server auth] library/node:pull token for registry-1.docker.io
  => [todo-server internal] load dockertignore
  => [todo-server internal] transfering context: 120B
  => [todo-server 1/6] FROM docker.io/library/node:28.9-alpine@sha256:cb2301e2c5fe3165ba2616591ef53b4b6223849ac0871c138f56d5f7aebbedb
  => [todo-server internal] load build context
  => [todo-server internal] transfering context: 1.21kB
  => CACHED [todo-server 2/6] RUN mkdir -p /usr/src/app
  => CACHED [todo-server 3/6] WORKDIR /usr/src/app
  => CACHED [todo-server 4/6] COPY package*.json ./
  => CACHED [todo-server 5/6] RUN npm install --silent
  => CACHED [todo-server 6/6] COPY .
  => [todo-server] exporting to image
  => [todo-server] exporting layers
  => [todo-server] writing image sha256:040b7d47f599f8ced79aaa77800e375665cd3ac378406035e9d1d28767d47446
  => [todo-server] naming to docker.io/library/todo-server
  => [todo-server] resolving provenance for metadata file
[+] Running 4/4
  ✓ todo-server Built 0.0s
  ✓ Network ltc_onl2_sw1_s2_group1_todo-net Created 0.4s
  ✓ Container todo-database Started 3.0s
  ✓ Container todo-server Started 2.0s
kyrillos@master:~/ITC_ONL2_SWD1_S2_group1$ docker compose ps

```

27. `docker compose ps` and now both are running

```

kyrillos@master:~/ITC_ONL2_SWD1_S2_group1
[+] Running 9/9
  ✓ todo-database Pulled 90.7s
    ✓ d4c3c9e5f8 Pull complete 26.2s
    ✓ bca5893fe8bd Pull complete 26.3s
    ✓ 35ec036951f8 Pull complete 26.9s
    ✓ ddb77e597b02 Pull complete 27.2s
    ✓ 7ab9eb5a409d Pull complete 27.2s
    ✓ a6c1ba219414 Pull complete 27.3s
    ✓ 83b651df5384 Pull complete 87.5s
    ✓ e233f2d13609 Pull complete 87.6s
[+] Building 2.0s (13/13) FINISHED
  => [todo-server internal] load build definition from Dockerfile
  => [todo-server internal] load metadata for docker.io/library/node:28.9-alpine
  => [todo-server auth] library/node:pull token for registry-1.docker.io
  => [todo-server internal] load dockertignore
  => [todo-server internal] transfering context: 120B
  => [todo-server 1/6] FROM docker.io/library/node:28.9-alpine@sha256:cb2301e2c5fe3165ba2616591ef53b4b6223849ac0871c138f56d5f7aebbedb
  => [todo-server internal] load build context
  => [todo-server internal] transfering context: 1.21kB
  => CACHED [todo-server 2/6] RUN mkdir -p /usr/src/app
  => CACHED [todo-server 3/6] WORKDIR /usr/src/app
  => CACHED [todo-server 4/6] COPY package*.json ./
  => CACHED [todo-server 5/6] RUN npm install --silent
  => CACHED [todo-server 6/6] COPY .
  => [todo-server] exporting to image
  => [todo-server] exporting layers
  => [todo-server] writing image sha256:040b7d47f599f8ced79aaa77800e375665cd3ac378406035e9d1d28767d47446
  => [todo-server] naming to docker.io/library/todo-server
  => [todo-server] resolving provenance for metadata file
[+] Running 4/4
  ✓ todo-server Built 0.0s
  ✓ Network ltc_onl2_sw1_s2_group1_todo-net Created 0.4s
  ✓ Container todo-database Started 3.0s
  ✓ Container todo-server Started 2.0s
kyrillos@master:~/ITC_ONL2_SWD1_S2_group1$ docker compose ps

```

NAME	IMAGE	COMMAND	SERVICE	CREATED	STATUS	PORTS
todo-database	mongo:4.4	"docker-entrypoint.s..."	todo-database	About a minute ago	Up About a minute	0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
todo-server	todo-server	"docker-entrypoint.s..."	todo-server	About a minute ago	Up 4 seconds	0.0.0.0:3000->3000/tcp, :::3000->3000/tcp, 8082/tcp

```

kyrillos@master:~/ITC_ONL2_SWD1_S2_group1$

```

28. SECRET .env problem:

```

localhost:3000
Error: secret option required for sessions
    at session (/usr/src/app/node_modules/express-session/index.js:204:12)
    at Layer.handle [as handle_request] (/usr/src/app/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/usr/src/app/node_modules/express/lib/router/index.js:328:13)
    at /usr/src/app/node_modules/express/lib/router/index.js:286:9
    at Function.process_params (/usr/src/app/node_modules/express/lib/router/index.js:346:12)
    at next (/usr/src/app/node_modules/express/lib/router/index.js:280:10)
    at SendStream.error (/usr/src/app/node_modules/serve-static/index.js:121:7)
    at SendStream.emit (node:events:514:28)
    at SendStream.error (/usr/src/app/node_modules/send/index.js:270:17)
    at SendStream.onStatError (/usr/src/app/node_modules/send/index.js:417:12)

```

`process.env.SESSION_KEY` but we set `SESSION_SECRET` in `.env`. Since `SESSION_KEY` is undefined, express-session throws that "secret option required" error.

29. Edit the .env file:

MONGO\_URI=mongodb://todo-database:27017/todo-app

SESSION\_KEY=my-super-secret

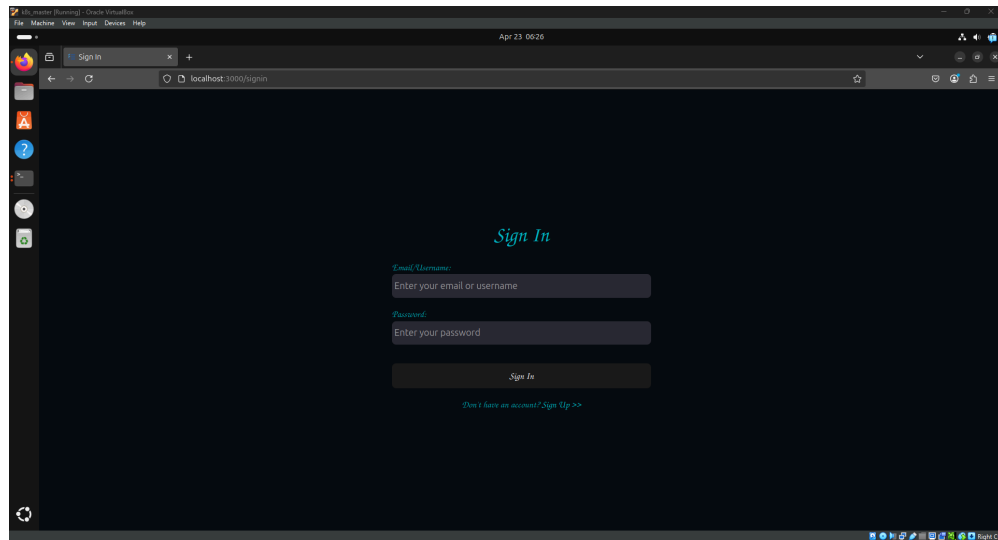
PORT=3000

```
kyrillos@master: ~/ITC_ONL2_SWd1_S2_group1/todo-src
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1$ docker compose ps
NAME                IMAGE             COMMAND                  SERVICE    CREATED         STATUS         PORTS
todo-database       mongo:4.4         "docker-entrypoint.s..." todo-database 55 minutes ago  Up 2 minutes   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
todo-server         todo-server       "docker-entrypoint.s..." todo-server  55 minutes ago  Up 7 seconds   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1$ ls
assets  docker-compose.yml  docs  LICENSE  package-lock.json  README.md  TODO-App.code-workspace  todo-src
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1$ cd todo-src/
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ ls
app.js  Dockerfile  node_modules  package.json  package-lock.json  public  README.md  server.js  src  views
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ ls
app.js  Dockerfile  node_modules  package.json  package-lock.json  public  README.md  server.js  src  views
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ nano .env
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ nano .env
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ docker compose down
[+] Running 3/3
  ✓ Container todo-server          Removed          12.5s
  ✓ Container todo-database       Removed          0.8s
  ✓ Network itc_onl2_sw_d1_s2_group1_todo-net Removed          0.3s
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$
```

```
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src
[+] Building 0.8s
  > writing image sha256:25c179ca4a45d3f268230c91b5fcede2c7adf5aad99680eb066d01a99822dfad0
  > naming to docker.io/library/todo-server
  > [todo-server] resolving provenance for metadata file
[+] Building 0.8s
  > todo-server 0.8s Built
  > Network itc_onl2_sw_d1_s2_group1_todo-net Created
  > Container todo-database 0.1s Started
  > Container todo-server 0.8s Started
  > Container todo-server 1.1s Started
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$ docker compose ps
NAME                IMAGE             COMMAND                  SERVICE    CREATED         STATUS         PORTS
todo-database       mongo:4.4         "docker-entrypoint.s..." todo-database 13 seconds ago  Up 13 seconds   0.0.0.0:27017->27017/tcp, :::27017->27017/tcp
todo-server         todo-server       "docker-entrypoint.s..." todo-server  13 seconds ago  Up 12 seconds   0.0.0.0:3000->3000/tcp, :::3000->3000/tcp, 8082/tcp
kyrillos@master:~/ITC_ONL2_SWd1_S2_group1/todo-src$
```



30. The app is dockerized and runninggg



31. Jenkins CI/CD Pipeline plan draft

Stage	Trigger	Tasks	Tooling	Deliverable
<b>Build</b>	Push to <code>main</code> or <code>dev</code>	<ul style="list-style-type: none"> <li>Checkout repo ( <code>git clone ...</code> ).</li> <li>Build Docker image from <code>todo-src/Dockerfile</code>.</li> <li>Validate &amp; lint Dockerfile.</li> </ul>	Docker, Docker Compose	<code>todo-server:ci-&lt;SHA&gt;</code> image
<b>Test</b>	On successful build	<ul style="list-style-type: none"> <li>Run <code>npm run lint</code> on <code>todo-src</code>.</li> <li>(If you add tests) run <code>npm test</code>.</li> <li>Run <code>npm audit</code> for vulns.</li> </ul>	ESLint, Prettier, npm	Pass/Fail report in CI console
<b>Publish</b>	Tests pass	<ul style="list-style-type: none"> <li>Tag image with CI build number ( <code>ci-&lt;BUILD_ID&gt;</code> ).</li> <li>Push images ( <code>mongo</code> comes from off-the-shelf ).</li> </ul>	Docker CLI, Docker Hub	Images in central registry
<b>Deploy</b>	Image push (or manual)	<ul style="list-style-type: none"> <li>Pull new image on target host.</li> <li>Run <code>docker compose pull &amp;&amp; docker compose up -d</code>.</li> <li>Smoke-test HTTP.</li> </ul>	Docker Compose, curl	Live app at <code>http://&lt;host&gt;:3000</code>
<b>Notify</b>	Deploy success/failure	<ul style="list-style-type: none"> <li>Send Slack/Email alert.</li> <li>On failure, trigger rollback playbook (e.g. re-deploy previous tag).</li> </ul>	CI Notifications plugin	Channel/Email updated with pipeline status