

## LAB 6 Microservices Architecture

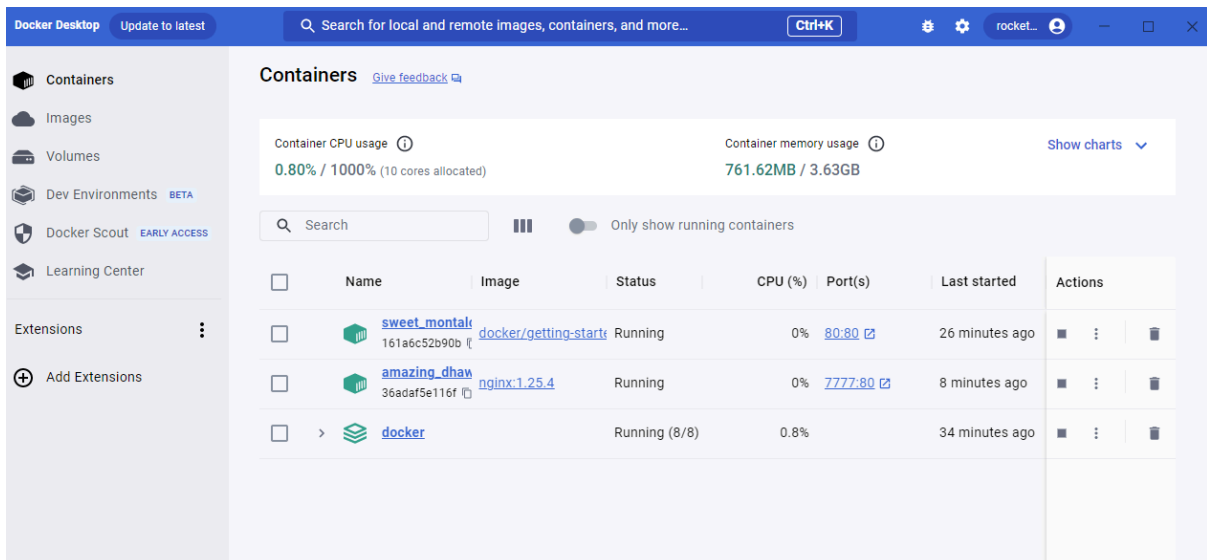
ติดตั้ง WSL

wsl --install

wsl --version

```
C:\Users\ERP-ST01>wsl -l -v
NAME                STATE      VERSION
* docker-desktop    Running    2
  Ubuntu             Running    2
  docker-desktop-data Running    2
```

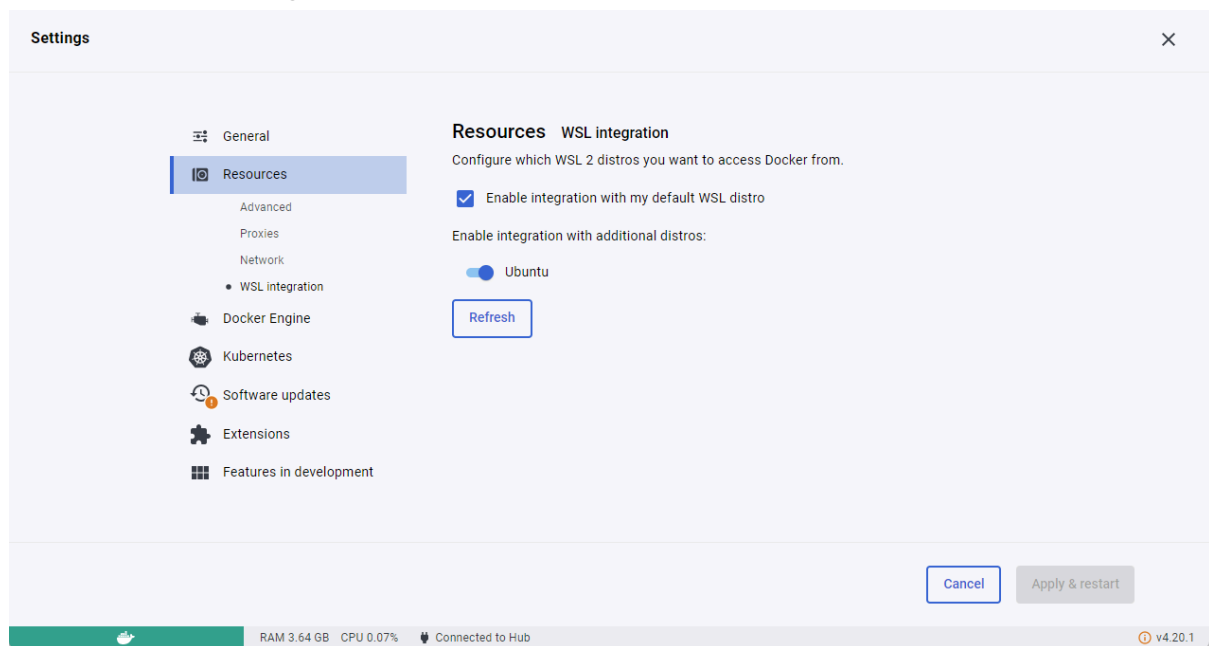
ติดตั้ง Docker Desktop



The screenshot shows the Docker Desktop application window. The left sidebar contains navigation options: Containers, Images, Volumes, Dev Environments (BETA), Docker Scout (EARLY ACCESS), and Learning Center. The main area is titled 'Containers' and displays system usage: Container CPU usage at 0.80% / 1000% (10 cores allocated) and Container memory usage at 761.62MB / 3.63GB. Below this is a search bar and a toggle for 'Only show running containers'. A table lists the running containers:

Name	Image	Status	CPU (%)	Port(s)	Last started	Actions
sweet_montali 161a6c52b90b	docker/getting-start	Running	0%	80:80	26 minutes ago	Stop, Restart, Delete
amazing_dhaw 36adaf5e116f	nginx:1.25.4	Running	0%	7777:80	8 minutes ago	Stop, Restart, Delete
> docker		Running (8/8)	0.8%		34 minutes ago	Stop, Restart, Delete

## Enable WSL integration เพื่อให้ใช้ Docker บน Ubuntu ผ่าน WSL ได้



### ทดสอบ

docker ps

จากนั้นลองรัน container nginx ด้วย port 7777

docker container run -d -p 7777:80 nginx:1.25.4

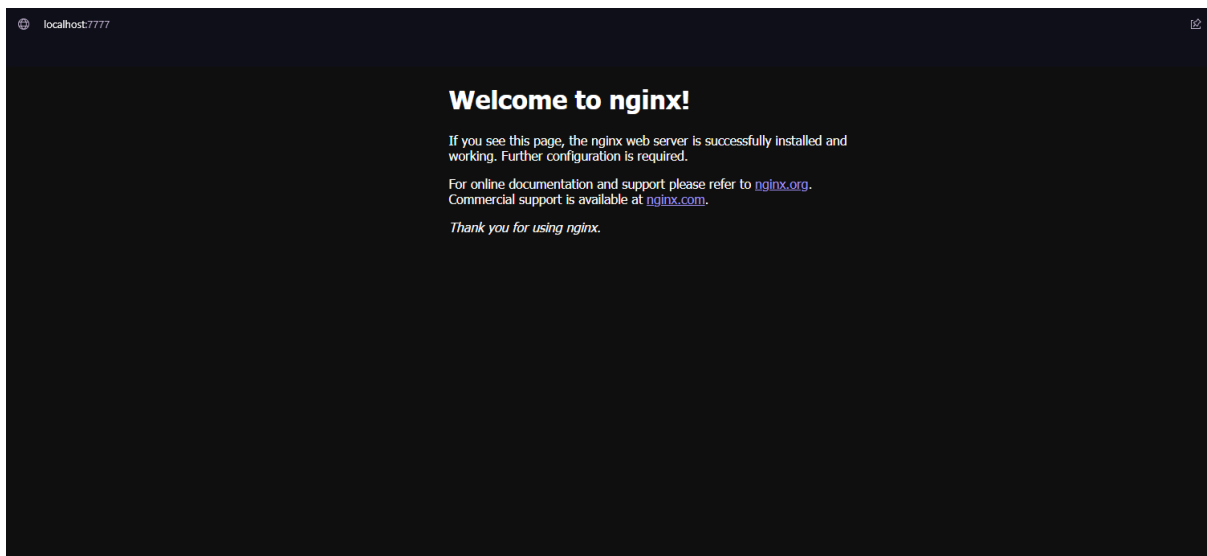
ลองเข้าผ่าน port 7777

http://localhost:7777

or

http://127.0.0.1:7777

```
devuser@DESKTOP-UN2PSA8:~$ docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
PORTS         NAMES
161a6c52b90b   docker/getting-started              "/docker-entrypoint..." 17 minutes ago Up 17 minutes
0.0.0.0:80->80/tcp      sweet_montalcini
7059fb99fd57   docker-grafana                      "/run.sh"                15 months ago Up 24 minutes
0.0.0.0:3000->3000/tcp   user-grafana
e620bbf15ceb   nodered/node-red                    "./entrypoint.sh"         15 months ago Up 24 minutes (heal
thy) 0.0.0.0:1880->1880/tcp user-nodered
67705629b023   docker-php-apache-environment       "docker-php-entrypoint..." 15 months ago Up 24 minutes
0.0.0.0:8080->80/tcp     user-php-apache
e59ecc025d25   phpmyadmin/phpmyadmin               "/docker-entrypoint..." 15 months ago Up 24 minutes
0.0.0.0:8000->80/tcp     user-myadmin
820cb6eb5874   eclipse-mosquitto:1.6.9             "/docker-entrypoint..." 15 months ago Up 24 minutes
0.0.0.0:1883->1883/tcp, 0.0.0.0:9001->9001/tcp user-mosquitto
b21a4b4ae07f   mysql                               "docker-entrypoint.s..." 15 months ago Up 24 minutes
0.0.0.0:3306->3306/tcp, 33060/tcp user-db
28bb8b6f8b3b   influxdb:1.8.10                     "/entrypoint.sh infl..." 15 months ago Up 24 minutes
0.0.0.0:8086->8086/tcp   user-influxdb
fcb4129c07f9   adminer:latest                      "entrypoint.sh php -..." 15 months ago Up 24 minutes
0.0.0.0:8888->8080/tcp   user-adminer
devuser@DESKTOP-UN2PSA8:~$ docker container run -d -p 7777:80 nginx:1.25.4
Unable to find image 'nginx:1.25.4' locally
1.25.4: Pulling from library/nginx
13808c22b207: Pull complete
```



container stop id อันเก่า

แล้วรัน container ใหม่ที่ config มาในชื่อ nginx-demo

```
devuser@DESKTOP-UN2PSA8:~$ docker container ls -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS
bbb5a24875e	nginx:1.25.4	"/docker-entrypoint..."	30 seconds ago	Created	
nginx-demo					

สร้างโฟลเดอร์ engse207 แล้ว clone git repository ลงในโฟลเดอร์

```
devuser@DESKTOP-UN2PSA8:~/engse207$ git clone https://github.com/se-rmutl/engse207-docker.g
Cloning into 'engse207-docker'...
remote: Enumerating objects: 86, done.
remote: Counting objects: 100% (86/86), done.
remote: Compressing objects: 100% (72/72), done.
remote: Total 86 (delta 5), reused 82 (delta 4), pack-reused 0 (from 0)
Receiving objects: 100% (86/86), 5.29 MiB | 3.48 MiB/s, done.
Resolving deltas: 100% (5/5), done.
devuser@DESKTOP-UN2PSA8:~/engse207$ ls
engse207-docker
```

Pull MSSQL Image ลง Docker

```
C:\Users\ERP-ST01>docker pull mcr.microsoft.com/mssql/server:2017-latest
2017-latest: Pulling from mssql/server
699451820735: Pull complete
ac40f1754a69: Pull complete
a6ec4b3b25ec: Pull complete
Digest: sha256:896972291ff7bd8b7fc172df994fb8b35135b0a7614e97b9e892c18328bc1208
Status: Downloaded newer image for mcr.microsoft.com/mssql/server:2017-latest
mcr.microsoft.com/mssql/server:2017-latest
```

รัน MSSQL แล้วเริ่มใช้คำสั่ง sql ด้วย bash

docker exec -it mssql2017 "bash"

```
devuser@DESKTOP-UN2PSA8:~/engse207/engse207-docker/docker-image$ docker run -e "ACCEPT_EULA=Y" -e "MSSQL_SA_
PASSWORD=P@ssw0rd" -p 1433:1433 --name mssql2017 -d mcr.microsoft.com/mssql/server:2017-latest
97d6c185b00a69bfe84245478f87b995584b0fc1eeb354801845afb313762ca5
devuser@DESKTOP-UN2PSA8:~/engse207/engse207-docker/docker-image$ docker exec -it mssql2017 "bash"
root@97d6c185b00a:/# /opt/mssql-tools/bin/sqlcmd -S localhost -U sa -p
Password:
1> SELECT Name from sys.databases;
2> go
Name
```

## สร้าง Database , เพิ่มตารางและ insert ข้อมูลลงในตาราง

```
(4 rows affected)

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total      46  avg   46.0 (21.7 xacts per sec.)
1> CREATE DATABASE team2_web_labDB;
2> go

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total      413  avg  413.0 (2.4 xacts per sec.)
1> CREATE TABLE dbo.Inventory ( id INT, name NVARCHAR(50), quantity INT, PRIMARY KEY (id));
2> go

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total       8  avg   8.0 (125.0 xacts per sec.)
1> INSERT INTO dbo.Inventory VALUES (1, 'banana', 150);
2> INSERT INTO dbo.Inventory VALUES (2, 'orange', 154);
3> go

(1 rows affected)

(1 rows affected)

Network packet size (bytes): 4096
```

## ทดสอบ

```
Clock Time (ms.): total      13  avg   13.0 (76.9 xacts per sec.)
1> SELECT * FROM dbo.Inventory;
2> go

id          name          quantity
-----
1 banana          150
2 orange          154

(2 rows affected)
```

## Import MSSQL Data และทดสอบ

```
Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total      10  avg  10.0 (100.0 xacts per sec.)

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total       6  avg   6.0 (166.7 xacts per sec.)

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total       3  avg   3.0 (333.3 xacts per sec.)
root@9bdadc90a2ee:/# use team2_web_labDB;
bash: use: command not found
root@9bdadc90a2ee:/# /opt/mssql-tools/bin/sqlcmd -S localhost -U sa -p
Password:
1> use team2_web_labDB;
2> go
Changed database context to 'team2_web_labDB'.

Network packet size (bytes): 4096
1 xact[s]:
Clock Time (ms.): total       9  avg   9.0 (111.1 xacts per sec.)
1> SELECT * FROM dbo.OnlineAgents;
2> go













OnlineAgent_id agent_id  agent_code  DIDQueueID  StartOnline  uuid  AgentName
AgentStatus IsLogin LastUpdated
-----
1 NULL 9999 1 2023-08-31 13:56:26.033 NULL 2023-08-31 13:56:26.033 cb40f193-1317-4b17-a360-506cd89700f7 Dev2
2 NULL 3891 1 2023-09-05 14:27:35.483 NULL 2023-09-05 14:27:35.483 0a9c6366-5ec7-4727-8698-045b977afe61 Dev2
3 NULL 9998 1 2023-09-07 14:05:20.360 NULL 2023-09-07 14:05:20.360 5ac956de-560b-4888-8b69-d77dfd3191d Dev2
```

## Pull Mongodb image มาใช้

```
root@DESKTOP-02C4R72:~/engse207/engse207-docker/docker-image# docker pull mongodb/mongodb-community-server:latest
latest: Pulling from mongodb/mongodb-community-server
c43c894ea6e9: Download complete
4a38c01437f4: Download complete
923f72ee8d96: Pulling fs layer
3561320b1d22: Download complete
5414378b6477: Download complete
2579002e66c4: Downloading [=====>] 49.28MB/160.6MB
09218eb43544: Download complete
dbf20f635577: Pulling fs layer
4f4fb700ef54: Download complete
ea9f76730df0: Download complete
5892f94b5cc5: Download complete
```

## รัน Container บน Docker

```
root@DESKTOP-02C4R72:~/engse207/engse207-docker/docker-image# docker run --name mongodb -p 27017:27017 -d mongodb/mongodb-community-server:latest
cc4e328c939aedef1989408b7e2f7ef001d5d2fc5ae93e92d74528f36280defcb
```

<input type="checkbox"/>	Name	Tag	Status	Created	Size	Actions
<input type="checkbox"/>	<a href="#">nginx</a> 9ff236ed47fe 	1.25.4	<a href="#">In use</a>	8 months ago	271.73 MB	  
<input type="checkbox"/>	<a href="#">mcr.microsoft.com/mssql/se</a> 896972291ff7 	2017-latest	<a href="#">In use</a>	25 days ago	1.81 GB	  
<input type="checkbox"/>	<a href="#">mongodb/mongodb-commun</a> 9c913685357d 	latest	<a href="#">In use</a>	9 hours ago	1.67 GB	  

Showing 3 items

## สร้าง DB ใน MongoDB

```
test> use wallboarddb
switched to db wallboarddb
wallboarddb> db.createUser({user: "wallboarduser",pwd: "WB1qazxsw2",roles: [ { role: "readWrite", db: "wallboarddb" } ]})
{ ok: 1 }
wallboarddb> show users
[
  {
    _id: 'wallboarddb.wallboarduser',
    userId: UUID('f2c2e065-6926-42f0-9a55-3c0f199bbce4'),
    user: 'wallboarduser',
    db: 'wallboarddb',
    roles: [ { role: 'readWrite', db: 'wallboarddb' } ],
    mechanisms: [ 'SCRAM-SHA-1', 'SCRAM-SHA-256' ]
  }
]
wallboarddb> █
```

```

wallboarddb> use wallboarddb
already on db wallboarddb
wallboarddb> db.createCollection("log", { capped : true, size : 5242880, max : 5000 } )
{ ok: 1 }
wallboarddb> db.log.insertOne({ capped : true, size : 5242880, max : 5000 })
{
  acknowledged: true,
  insertedId: ObjectId('67127e339e37747227fe6911')
}
wallboarddb> db.log.find()
[
  {
    _id: ObjectId('67127e339e37747227fe6911'),
    capped: true,
    size: 5242880,
    max: 5000
  }
]
wallboarddb> show collections
log
wallboarddb> show dbs
admin      148.00 KiB
config     12.00 KiB
local      40.00 KiB
wallboarddb 8.00 KiB
wallboarddb> █

```

## กำหนดรหัสผ่านยืนยัน ถ้าจะจัดการไฟล์ DB

```

mongod@cc4e328c939a:/$ exit
exit
root@DESKTOP-02C4R72:~/engse207/engse207-docker/docker-image# docker exec -it mongod mongosh -u wallboarduser -p MB1qazxsw2 --authenticationDatabase wallboarddb
Current Mongosh Log ID: 67127ee9866d4cd6afe6910
Connecting to:  mongod://ccredentials@127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&authSource=wallboarddb&appName=mongosh+2.3.2
Using MongoDB:  7.0.14
Using Mongosh:  2.3.2

For mongosh info see: https://www.mongodb.com/docs/mongosh-shell/

-----
The server generated these startup warnings when booting
2024-10-18T15:23:23.114+00:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2024-10-18T15:23:23.740+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2024-10-18T15:23:23.740+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never' in this binary version
2024-10-18T15:23:23.741+00:00: vm.max_map_count is too low
-----
test> █

```

## Docker Pull Nodejs Image มา

```

root@DESKTOP-02C4R72:~/engse207/engse207-docker/docker-image# docker pull node:20
20: Pulling from library/node
8ba0b3d08b81: Downloading [=====>] 64.39MB/64.39MB
d0c5f3b3f727: Download complete
5f57b3eca258: Download complete
27bc85537ac7: Downloading [=====>] 59.77MB/211.3MB
898ee3ab8d89: Downloading [=====>] 48.22MB/48.22MB
7d98d813d54f: Download complete
2c430d81b800: Download complete
e07e01c8fd93: Download complete
█

```

## Clone Repository มา

```

root@DESKTOP-02C4R72:~/engse207# git clone https://github.com/se-rmutl/engse207.git
Cloning into 'engse207'...
remote: Enumerating objects: 723, done.
remote: Counting objects: 100% (62/62), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 723 (delta 17), reused 48 (delta 10), pack-reused 661 (from 1)
Receiving objects: 100% (723/723), 7.06 MiB | 5.88 MiB/s, done.
Resolving deltas: 100% (202/202), done.
root@DESKTOP-02C4R72:~/engse207# ls
engse207  engse207-docker
root@DESKTOP-02C4R72:~/engse207# cd engse207
root@DESKTOP-02C4R72:~/engse207/engse207# ls
README.md  microservices  mvc_architecture  n-tier_architecture
root@DESKTOP-02C4R72:~/engse207/engse207# █

```

## ติดตั้ง package ของ endpoint

```
root@DESKTOP-02C4R72: ~/engse207/engse207-docker/engse207/n-tier_architecture/backend/endpoint-api
engse207-docker
root@DESKTOP-02C4R72:~/engse207# cd engse207-docker/
root@DESKTOP-02C4R72:~/engse207/engse207-docker# ls
README.md  docker  docker-compose  docker-image  dockerfile  engse207  mssql_scripts
root@DESKTOP-02C4R72:~/engse207/engse207-docker# cd engse207/
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207# ls
README.md  microservices  mvc_architecture  n-tier_architecture
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207# cd n-tier_architecture/
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207/n-tier_architecture# cd backend/
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207/n-tier_architecture/backend# cd endpoint-api/
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207/n-tier_architecture/backend/endpoint-api# ls
package.json  sqlConfig.js  webreport-api-https.js
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207/n-tier_architecture/backend/endpoint-api# npm i
npm warn deprecated har-validator@5.1.5: this library is no longer supported
npm warn deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm warn deprecated uuid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain
circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.

added 242 packages, and audited 243 packages in 31s

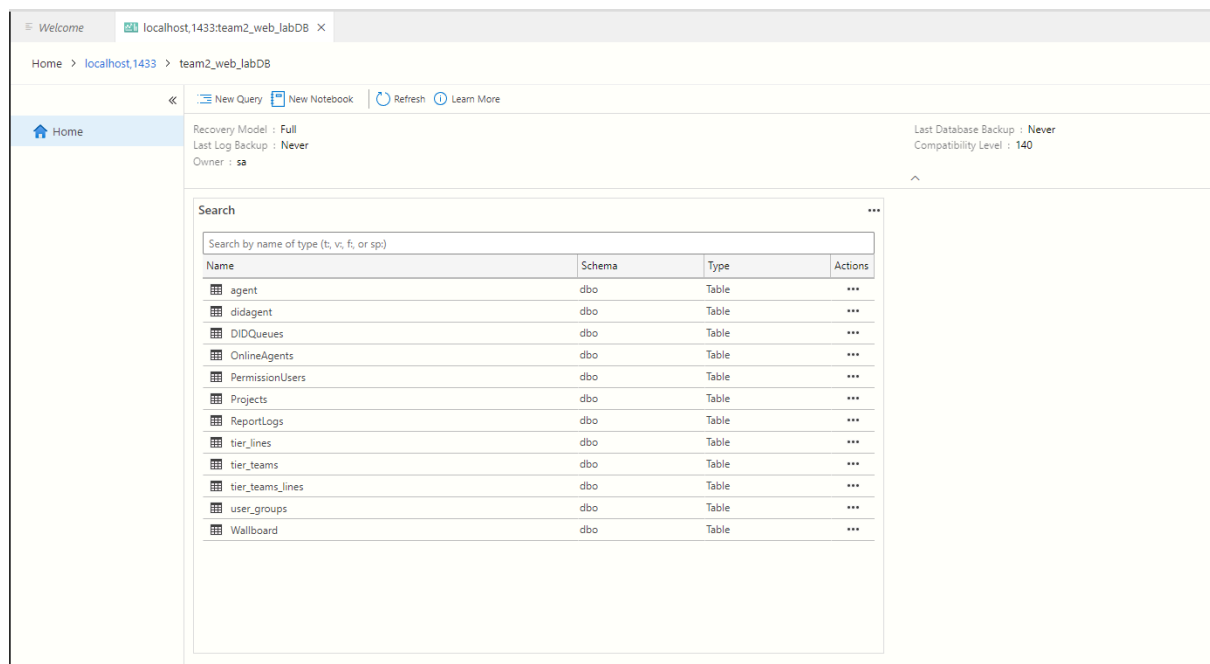
24 packages are looking for funding
  run `npm fund` for details

2 moderate severity vulnerabilities

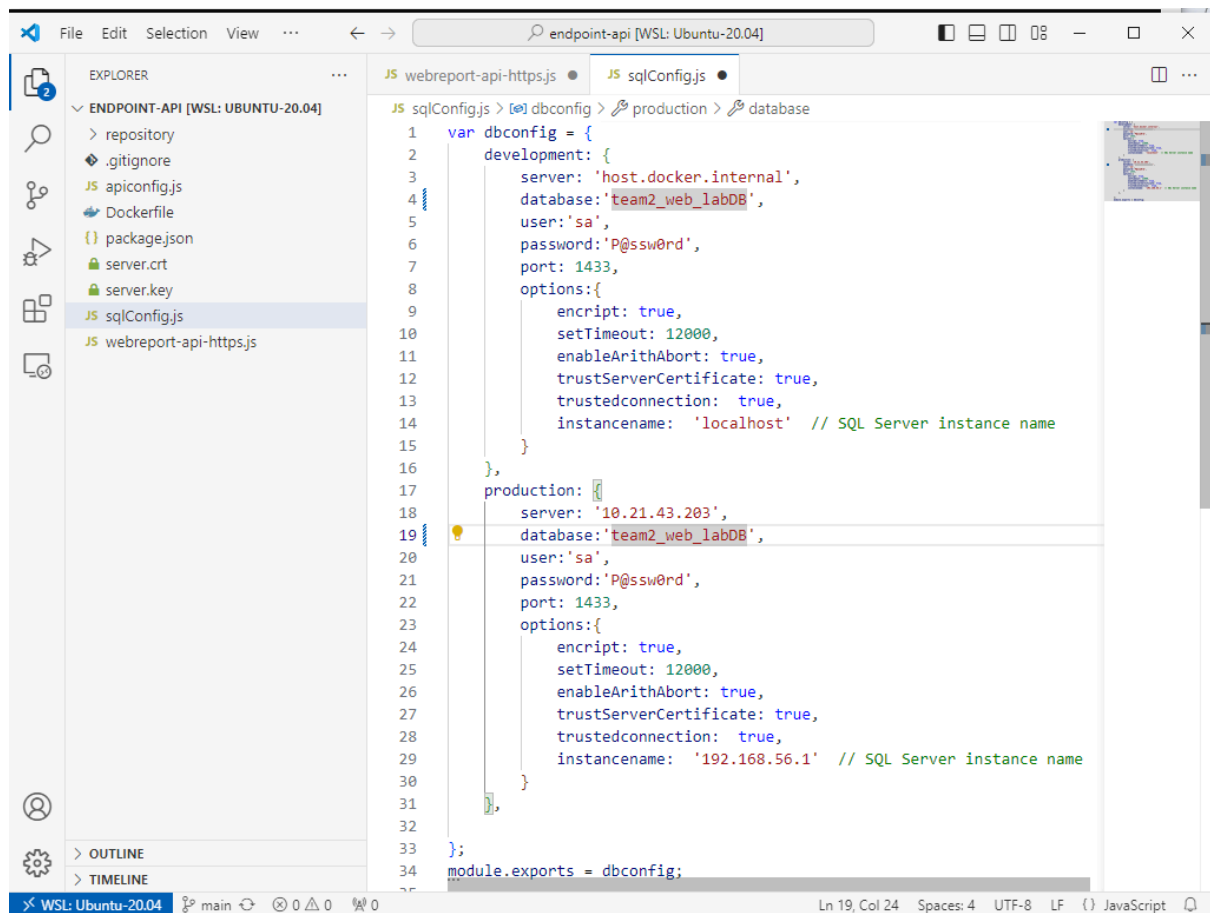
Some issues need review, and may require choosing
a different dependency.

Run `npm audit` for details.
root@DESKTOP-02C4R72:~/engse207/engse207-docker/engse207/n-tier_architecture/backend/endpoint-api#
```

## เตรียมข้อมูลใน Azure Data Studio ให้พร้อม



## เตรียมการเชื่อมต่อ MSSQL

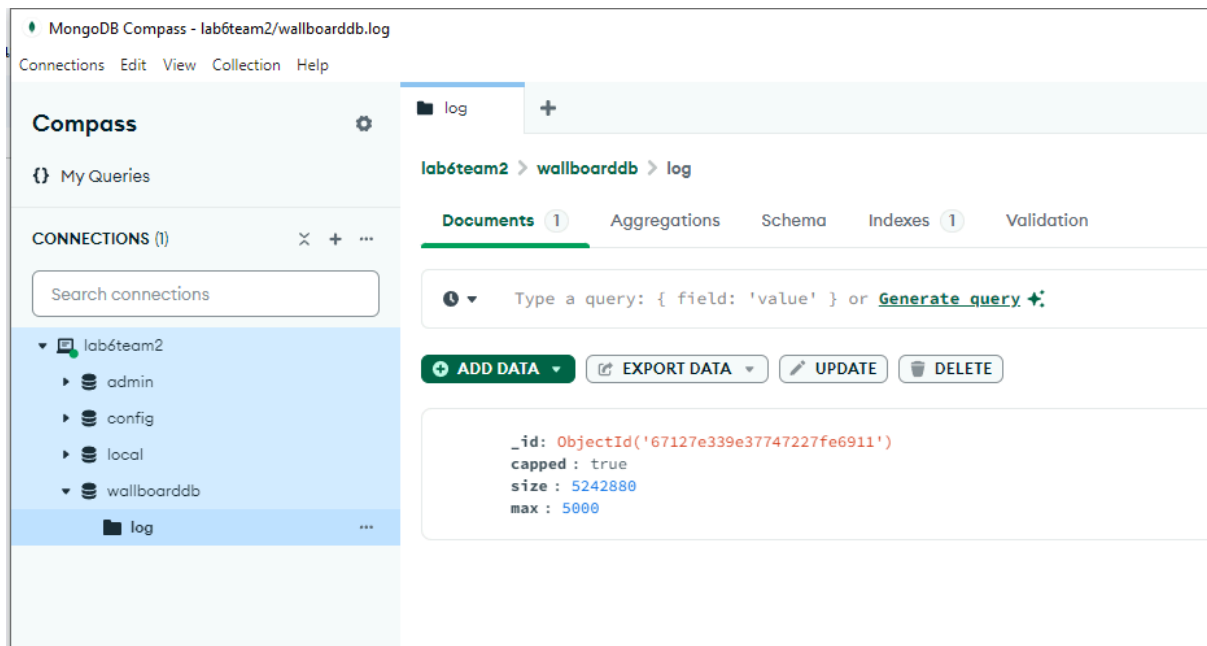


The screenshot shows a Visual Studio Code editor window with the file explorer on the left and the code editor on the right. The file explorer shows the project structure for 'ENDPOINT-API [WSL: UBUNTU-20.04]'. The code editor displays the 'sqlConfig.js' file, which defines a 'dbconfig' object with 'development' and 'production' environments. The 'development' environment is configured to connect to a local MSSQL instance, while the 'production' environment is configured to connect to a remote MSSQL instance.

```
1 var dbconfig = {
2   development: {
3     server: 'host.docker.internal',
4     database: 'team2_web_labDB',
5     user: 'sa',
6     password: 'P@ssw0rd',
7     port: 1433,
8     options: {
9       encrypt: true,
10      setTimeout: 12000,
11      enableArithAbort: true,
12      trustServerCertificate: true,
13      trustedconnection: true,
14      instancename: 'localhost' // SQL Server instance name
15    }
16  },
17  production: {
18    server: '10.21.43.203',
19    database: 'team2_web_labDB',
20    user: 'sa',
21    password: 'P@ssw0rd',
22    port: 1433,
23    options: {
24      encrypt: true,
25      setTimeout: 12000,
26      enableArithAbort: true,
27      trustServerCertificate: true,
28      trustedconnection: true,
29      instancename: '192.168.56.1' // SQL Server instance name
30    }
31  },
32 };
33 module.exports = dbconfig;
```

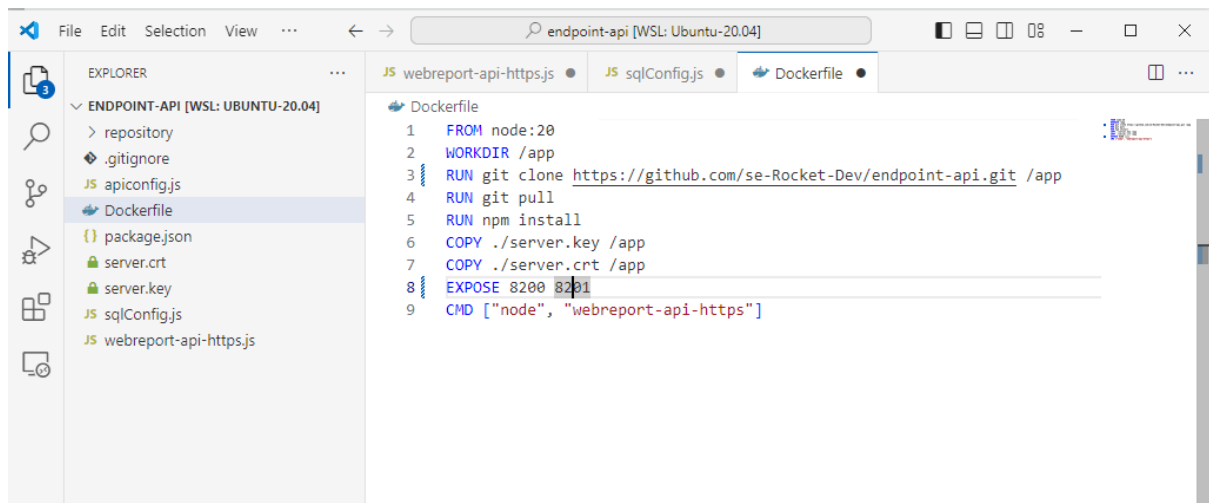


## เตรียม MongoDB ด้วย MongoCompass



## สร้าง Endpoint Container

### สร้าง Dockerfile







สร้าง Image : docker build -t endpointapi



ได้ image มาแล้วรัน

`docker run -d -p 8200:8200 -p 8201:8201 endpointapi`

 <b>focused_mcclintock</b> 014bec00af96 	<code>endpointapi:&lt;none&gt;</code>	Running	<a href="#">8200:8200</a>  <a href="#">8201:8201</a>  <a href="#">Show less</a>
---	---------------------------------------	---------	---

test api ใน postman

GET Endpoint-api

+

ENGSE207 Final / Endpoint-api

GET

https://localhost:8200

Overview

Params

Authorization

Headers (8)

Body

Scripts

Settings

Auth Type

Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Heads up! These parameters hold sensitive data. To keep this data secure while working in a collaborat variables.

Token

00D5D0000001aaZiARgAQGuQzp.mOv2jml...

Body

Cookies

Headers (10)

Test Results

Pretty

Raw

Preview

Visualize

HTML

Test Hello, from Endpoint Web Report API.

test get user by id

GET Endpoint-api

HTTP ENGSE207 Final / Endpoint-api

GET https://192.168.56.1:8200/api/v1/getOnlineAgentByAgentCode?agentcode=9998

Overview Params Authorization Headers (8) Body Scripts Settings

Endpoint-api

Add request description...

Body Cookies Headers (10) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "error": false,
3   "statusCode": 200,
4   "data": {
5     "OnlineAgent_id": 3,
6     "agent_id": null,
7     "agent_code": "9998",
8     "DIDQueueID": null,
9     "StartOnline": "2023-09-07T14:05:20.360Z",
10    "uuid": "5ac956de-560b-4888-8b69-d77dfdf3191d",
11    "AgentName": "Dev2",
12    "AgentStatus": "1",
13    "IsLogin": "1",
14    "LastUpdated": "2023-09-07T14:05:20.360Z"
15  }
16 }
```

test post user

# สร้าง Endpoint Container

## สร้าง Dockerfile

EXPLORER

▼ PARSE-SERVER-API [WSL: UBUNTU-20.04]

> cloud

> logs

🔍 .gitignore

JS api-dashboard-https.js

JS api-wallboard-https.js

Dockerfile

{ } package.json

📖 README.md

Dockerfile > ...

```
1 FROM node:20
2 WORKDIR /app
3 RUN git clone https://github.com/se-Rocket-Dev/Parse-server-API.git /app
4 RUN git pull
5 RUN npm install
6 COPY ./server.key /app
7 COPY ./server.crt /app
8 EXPOSE 8205
9 CMD ["node", "api-wallboard-https.js"]
10
11
```

สร้าง Image : docker build -t parse-server .


 **parse-server**

59cd9fc1769c

latest

สร้าง Container ให้ parse-server

docker run -d -p 8205:8205 parse-server

 **great\_euclid**

b5df249dec19

parse-server:<none>

Running

8205:8205

## ลองทดสอบ POST ด้วย Postman

ENGINE207 Final / Parse-API

POST https://127.0.0.1:8205/api/functions/postOnlineAgentListByTeam?AgentCode=9999&AgentName=Test Agent2&Queue=5&AgentStatus=4&AgentStatusCode=5&IsLogin=1

Send

Overview Params Authorization Headers (11) Body Scripts Settings

Query Params

Key	Value	Description
AgentCode	9999	
AgentName	Test Agent2	
Queue	5	
AgentStatus	4	
AgentStatusCode	5	
IsLogin	1	

Body Cookies Headers (11) Test Results

200 OK 49 ms 684 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "result": 0
3 }
```

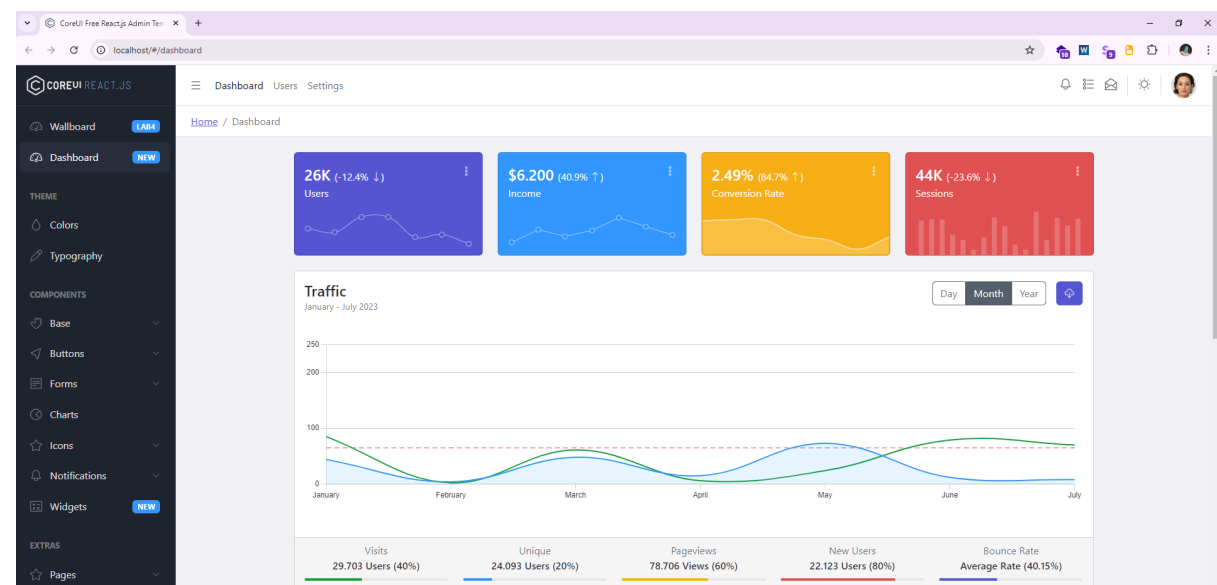
# สร้าง CoreUI Container

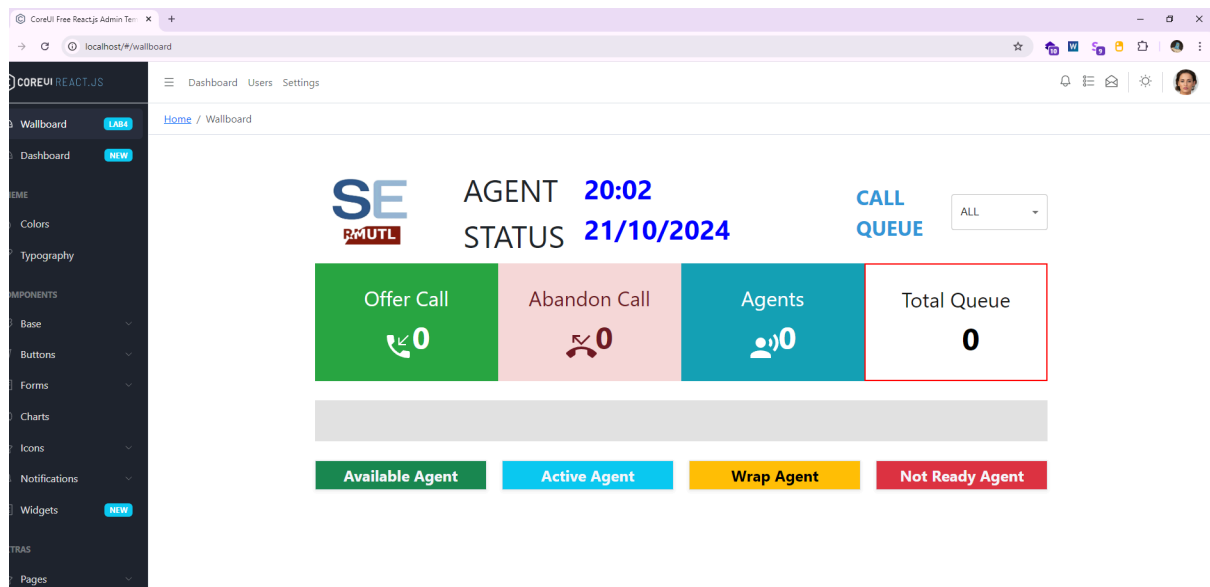
## สร้าง Dockerfile

```
Dockerfile > ...
1 FROM node:18-alpine AS build
2 RUN apk add --no-cache git
3 WORKDIR /app
4 RUN git clone https://github.com/se-Rocket-Dev/CoreUI-Wallboard.git .
5 RUN git pull
6 RUN npm install --legacy-peer-deps
7 RUN npm run build
8 FROM nginx:alpine
9 COPY --from=build /app/build /usr/share/nginx/html
10 EXPOSE 80
11 CMD ["nginx", "-g", "daemon off;"]
```

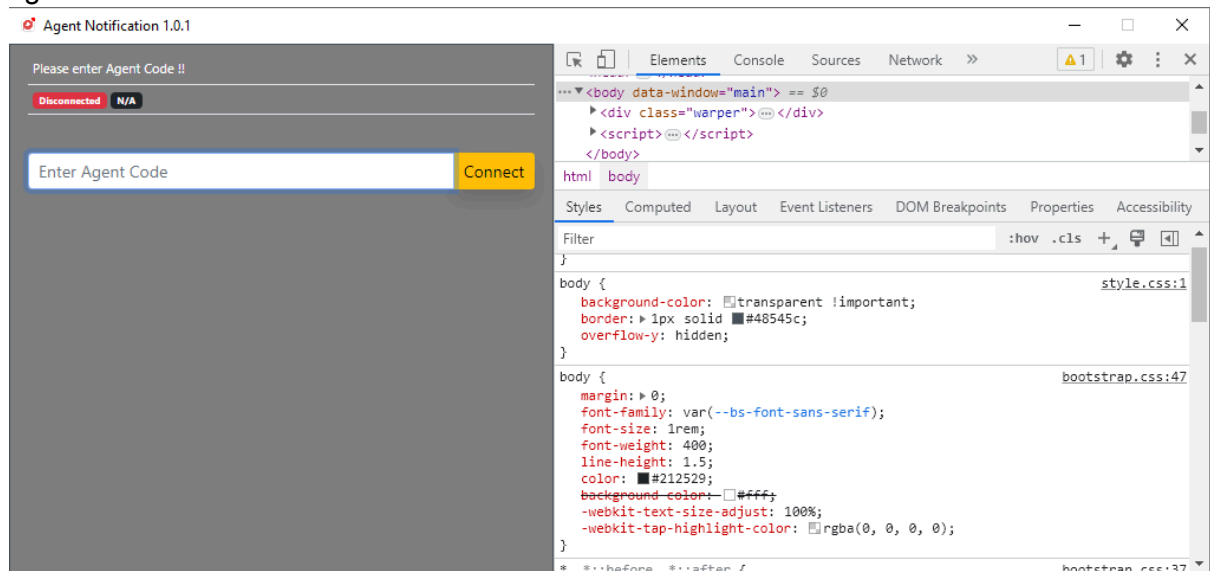
สร้าง image : docker build -t coreui .

<a href="#">coreui</a>	latest	Unused
01fef3a99e50		
<a href="#">cranky_lederberg</a>	coreui:<none>	Running 80:80
94dbd3a1f72b		





## Agent Notification แสดงผล



Agent Notification 1.0.1

[9998] Starts at: 2024-10-21 20:09:12

Connected

Available

Pop-up Chrome

Disconnect

Show Notification

Agent Status:

Available

Send Message:

Send to Agent Code

message

Send

Elements

Console

Sources

Network

1

body data-window="main" == \$0

<div class="warper">

<script>

</script>

</div>

</body>

html body

Styles

Computed

Layout

Event Listeners

DOM Breakpoints

Properties

Accessibility

Filter

:hov .cls

body {

background-color: transparent !important;

border: 1px solid #48545c;

overflow-y: hidden;

}

body {

margin: 0;

font-family: var(--bs-font-sans-serif);

font-size: 1rem;

font-weight: 400;

line-height: 1.5;

color: #212529;

background-color: #fff;

-webkit-text-size-adjust: 100%;

-webkit-tap-highlight-color: rgba(0, 0, 0, 0);

}

\*.::before, \*.::after {