

ชื่อ-นามสกุล **มณีสุขญาณี ธรรมนันท์พงษา** รหัสนักศึกษา **653380323-8** Section 4

Lab#8 – Software Deployment Using Docker

วัตถุประสงค์การเรียนรู้

- ผู้เรียนสามารถอธิบายเกี่ยวกับ Software deployment ได้
- ผู้เรียนสามารถสร้างและรัน Container จาก Docker image ได้
- ผู้เรียนสามารถสร้าง Docker files และ Docker images ได้
- ผู้เรียนสามารถนำซอฟต์แวร์ที่พัฒนาขึ้นให้สามารถรันบนสภาพแวดล้อมเดียวกันและทำงานร่วมกันกับสมาชิกในทีมพัฒนาซอฟต์แวร์ผ่าน Docker hub ได้
- ผู้เรียนสามารถเริ่มต้นใช้งาน Jenkins เพื่อสร้าง Pipeline ในการ Deploy งานได้

Pre-requisite

- ติดตั้ง Docker desktop ลงบนเครื่องคอมพิวเตอร์ โดยดาวน์โหลดจาก <https://www.docker.com/get-started>
- สร้าง Account บน Docker hub (<https://hub.docker.com/signup>)
- กำหนดให้ \$ หมายถึง Command prompt และ <> หมายถึง ให้ป้อนค่าของพารามิเตอร์ที่กำหนด

แบบฝึกปฏิบัติที่ 8.1 Hello world - รัน Container จาก Docker image

- เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
- เปิด Command line หรือ Terminal บน Docker Desktop จากนั้นสร้าง Directory ชื่อ Lab8_1

Lab Worksheet

3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_1 เพื่อใช้เป็น Working directory
4. ป้อนคำสั่ง \$ docker pull busybox หรือ \$ sudo docker pull busybox สำหรับกรณีที่ติดปัญหา Permission denied
(หมายเหตุ: BusyBox เป็น software suite ที่รองรับคำสั่งบางอย่างบน Unix - <https://busybox.net>)
5. ป้อนคำสั่ง \$ docker images

[Check point#1] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ พร้อมกับตอบคำถามต่อไปนี้

```
[MacBook-Air--Manitchaya:~ manitchaya$ docker ps
CONTAINER ID        IMAGE       COMMAND       CREATED          STATUS          PORTS      NAMES
MacBook-Air--Manitchaya:~ manitchaya$ docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
[MacBook-Air--Manitchaya:~ manitchaya$ docker images
REPOSITORY          TAG       IMAGE ID      CREATED          SIZE
busybox              latest    fc0179a204e2   3 months ago   4.04MB
mcr.microsoft.com/azure-sql-edge  latest    9d0e27694fc9   23 months ago  1.84GB
MacBook-Air--Manitchaya:~ manitchaya$
```

(1) สิ่งที่อยู่ภายในได้คอลัมน์ Repository คืออะไร

busybox

- ชื่อของ Image Repository ที่ดึงมาจาก Docker Hub (library/busybox)
- เป็น Image สำหรับชุดคำสั่ง Unix พื้นฐาน ที่มีขนาดเล็กมาก (ประมาณ 4 MB)

mcr.microsoft.com/azure-sql-edge

- เป็นชื่อ Repository สำหรับ Microsoft Azure SQL Edge
- Image นี้ใช้สำหรับการรัน SQL Server เวอร์ชัน Edge ซึ่งเหมาะสมสำหรับ IoT และการทำงานบนอุปกรณ์ Edge

Lab Worksheet

(2) Tag ที่ใช้บ่งบอกถึงอะไร

Tag ใน Docker ใช้เพื่อ ระบุเวอร์ชัน หรือ ตัวแปรเฉพาะ ของ Docker Image ใน Repository นั้นๆ ทำให้ง่ายต่อการจัดการและดึง Image ที่ต้องการได้อย่างเจาะจงแทนที่จะใช้ Image ทั่วไปเท่านั้น

6. ป้อนคำสั่ง \$ docker run busybox
7. ป้อนคำสั่ง \$ docker run -it busybox sh
8. ป้อนคำสั่ง ls
9. ป้อนคำสั่ง ls -la
10. ป้อนคำสั่ง exit
11. ป้อนคำสั่ง \$ docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"
12. ป้อนคำสั่ง \$ docker ps -a

[Check point#2] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ตั้งแต่ขั้นตอนที่ 6-12 พร้อมกับตอบคำถามต่อไปนี้

```
MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox
MacBook-Air--Manitchaya:~ manitchaya$ docker run -it busybox sh
/ # 
/ # ls
/ # ls -la
total 48
drwxr-xr-x  1 root   root    4096 Jan 23 04:51 .
drwxr-xr-x  1 root   root    4096 Jan 23 04:51 ..
-rwxr-xr-x  1 root   root    8 Jan 23 04:51 .dockerenv
drwxr-xr-x  2 root   root   12240 Sep 26 21:31 bin
drwxr-xr-x  5 root   root   358 Jan 23 04:51 dev
drwxr-xr-x  1 root   root    4096 Jan 23 04:51 etc
drwxr-xr-x  2 nobody nobody  4096 Sep 26 21:31 home
drwxr-xr-x  2 root   root    4096 Sep 26 21:31 lib
lrwxrwxrwx  1 root   root    3 Sep 26 21:31 lib64 -> lib
dr-xr-xr-x  184 root   root    8 Jan 23 04:51 proc
drwxr-xr-x  13 root   root   4096 Jan 23 04:51 root
drwxr-xr-x  2 root   root    8 Jan 23 04:51 sbin
drwxrwxrwt  2 root   root    4096 Sep 26 21:31 tmp
drwxr-xr-x  4 root   root    4096 Sep 26 21:31 usr
drwxr-xr-x  4 root   root    4096 Sep 26 21:31 var
/ # exit
MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox echo "Hello Manitchaya Thamanunpong"
MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox echo "Hello Manitchaya Thamanunpong from busybox"
MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox echo "Hello Manitchaya Thamanunpong from busybox"
MacBook-Air--Manitchaya:~ manitchaya$ docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS               NAMES
a3606e73bd70        busybox             "echo \"Hello Manitch...\" 10 seconds ago   Exited (0) 9 seconds ago
4ea3f95fbed2        busybox             "sh"                2 minutes ago     Exited (0) About a minute ago
66c3fa1862eb        busybox             "sh"                3 minutes ago     Exited (0) 3 minutes ago
959739bc2844        mcr.microsoft.com/azure-sql-edge   "/opt/mssql/bin/perm..." 12 months ago      Exited (255) 5 months ago   1401/tcp, 0.0.0.0:1433->1433/tcp   sql_server
```

Lab Worksheet

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสั้นๆ

Option -it ในคำสั่ง docker run มีผลดังนี้:

- -i (Interactive) : ทำให้สามารถโต้ตอบกับ Standard Input (STDIN) ของ Container ได้
- -t (TTY) : จำลอง Terminal (TTY) เพื่อให้การแสดงผลใน Shell อ่านง่ายขึ้น

สรุป : การใช้ -it ช่วยให้คุณเข้าสู่ Interactive Shell ของ Container (ในตัวอย่างคือ sh) และสามารถรันคำสั่งภายใน Container ได้เหมือนทำงานบนระบบ Linux ปกติ

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

คอลัมน์ **STATUS** ในคำสั่ง docker ps -a แสดงสถานะของ Container โดยบอกว่า

- **Exited (code):** Container หยุดทำงานแล้ว เช่น Exited (0): การทำงานสำเร็จ, Exited (255): เกิดข้อผิดพลาด
- **Up:** Container กำลังทำงานอยู่
- **Restarting:** Container กำลังรีสตาร์ทใหม่
- **Created:** Container ถูกสร้างแต่ยังไม่ได้เริ่มทำงาน

Lab Worksheet

13. ป้อนคำสั่ง \$ docker rm <container ID ที่ต้องการลบ>

[Check point#3] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 13

```
|MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox echo "Hello Manitchaya Thamanunpong" |MacBook-Air--Manitchaya:~ manitchaya$ docker run busybox echo "Hello Manitchaya Thamanunpong" from busybox |MacBook-Air--Manitchaya:~ manitchaya$ docker ps -a |CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES |a3606ef3bd70 busybox "echo \"Hello Manitchaya Thamanunpong\" from busybox" 10 seconds ago Exited (0) 9 seconds ago |4ea3f95fbed0 busybox "sh" 2 minutes ago Exited (0) About a minute ago |66c3fa1862e1 busybox "sh" 3 minutes ago Exited (0) 3 minutes ago |9597390c2044 microsoft/com.azure-sql-edge "/opt/mssql/bin/perm..." 12 months ago Exited (255) 5 months ago |MacBook-Air--Manitchaya:~ manitchaya$ docker rm |"docker rm" requires at least 1 argument. |See 'docker rm --help'. |Usage: docker rm [OPTIONS] CONTAINER [CONTAINER...] |Remove one or more containers |MacBook-Air--Manitchaya:~ manitchaya$ docker rm a3606ef3bd70 |MacBook-Air--Manitchaya:~ manitchaya$ docker ps -a |CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES |4ea3f95fbed0 busybox "sh" 47 minutes ago Exited (0) 46 minutes ago |66c3fa1862e1 busybox "sh" 48 minutes ago Exited (0) 48 minutes ago |9597390c2044 microsoft/com.azure-sql-edge "/opt/mssql/bin/perm..." 12 months ago Exited (255) 5 months ago |MacBook-Air--Manitchaya:~ manitchaya$ |
```

แบบฝึกปฏิบัติที่ 8.2: สร้าง Docker file และ Docker image

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้

2. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_2

3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_2 เพื่อใช้เป็น Working directory

4. สร้าง Dockerfile.swp ไว้ใน Working directory

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ (Windows) บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

FROM busybox

CMD echo "Hi there. This is my first docker image."

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

\$ cat > Dockerfile << EOF

FROM busybox

CMD echo "Hi there. This is my first docker image."

Lab Worksheet

CMD echo “ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น”

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t <ชื่อ Image> .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

[Check point#4] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 5 พร้อมกับตอบคำถามต่อไปนี้

```
MacBook-Air--Manitchaya:~ manitchaya$ mkdir Lab8_2
MacBook-Air--Manitchaya:~ manitchaya$ 
MacBook-Air--Manitchaya:~ manitchaya$ cd Lab8_2
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ touch Dockerfile.swp
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ cat > Dockerfile << EOF
> FROM busybox
> CMD echo "Hi there. This is my first docker image."
> CMD echo "Manitchaya Thamanunpong 653380323-8 May"
> EOF
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ docker build -t <Manitchaya_images> .
-bash: Manitchaya_images: No such file or directory
[MacBook-Air--Manitchaya:Lab8_2 manitchaya$ docker build -t Manitchaya_images .
invalid argument "Manitchaya_images" for "-t, --tag" flag: invalid reference format: repository name must be lowercase
See 'docker build --help'.
[MacBook-Air--Manitchaya:Lab8_2 manitchaya$ docker build -t manitchaya_images .
[+] Building 0.1s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 308B
=> [internal] load .dockerrcignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [1/1] FROM docker.io/library/busybox
=> exporting to image
=> => exporting layers
=> => writing image sha256:59b6e09c73802865cb7562c900cb4677e404435047a9f58b7de9d3ea16449ce3
=> => naming to docker.io/library/manitchaya_images

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ docker images
REPOSITORY          TAG      IMAGE ID      CREATED     SIZE
busybox              latest   fc0179a204e2   3 months ago   4.04MB
manitchaya_images    latest   59b6e09c7380   3 months ago   4.04MB
mcr.microsoft.com/azure-sql-edge  latest   9d0e27694fc9   23 months ago  1.84GB
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ docker run manitchaya_images
"Manitchaya Thamanunpong 653380323-8 May"
MacBook-Air--Manitchaya:Lab8_2 manitchaya$
```

(1) คำสั่งที่ใช้ในการ run คือ

Lab Worksheet

`docker run <ชื่อ Image>`

- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสั้นๆ
Option -t ช่วยตั้งชื่อและแท็กให้ Image ที่สร้าง ทำให้ง่ายต่อการจัดการและเรียกใช้งาน Image ในคำสั่งอื่นๆ เช่น docker run หรือ docker push

แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
 2. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_3
 3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_3 เพื่อใช้เป็น Working directory
 4. สร้าง Dockerfile.swp ไว้ใน Working directory
- สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดว์ บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

FROM busybox

CMD echo "Hi there. My work is done. You can run them from my Docker image."

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

```
$ cat > Dockerfile << EOF
```

FROM busybox

CMD echo "Hi there. My work is done. You can run them from my Docker image."

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"

Lab Worksheet

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t <username> ที่ลงทะเบียนกับ Docker Hub>/lab8

5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

\$ docker run <username> ที่ลงทะเบียนกับ Docker Hub>/lab8

[Check point#5] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้ในขั้นตอนที่ 5

```
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ mkdir Lab8_3
MacBook-Air--Manitchaya:Lab8_2 manitchaya$ cd Lab8_3
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ touch Dockerfile.swp
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ cat > Dockerfile << EOF
> FROM busybox
> CMD echo "Hi there. My work is done. You can run them from my Docker image."
>
> CMD echo "Manitchaya Thamanunpongsa 653380323-8"
> EOF
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ touch Dockerfile
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ docker build -t manitchaya/lab8 .
[+] Building 0.1s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] transfer dockerfile: 192B
=> [internal] load .dockerignore
=> [internal] transfer context: 2B
=> [internal] load metadata for docker.io/library/busybox:latest
=> CACHED [1/1] FROM docker.io/library/busybox
=> exporting to image
[ => => exporting layers
=> => writing image sha256:816c83893c7170c6317a9baf8963957c1447184f3462bf5a7e0e548a5cb2041f
[ => => naming to docker.io/manitchaya/lab8
[

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ docker run manitchaya/lab8
"Manitchaya Thamanunpongsa 653380323-8"
MacBook-Air--Manitchaya:Lab8_3 manitchaya$
```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

\$ docker push <username> ที่ลงทะเบียนกับ Docker Hub>/lab8
ในการณ์ที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

Lab Worksheet

\$ docker login แล้วป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้คำสั่ง

\$ docker login -u <username> -p <password>

7. ไปที่ Docker Hub กด Tab ชื่อ Tags หรือไปที่ Repository ก็ได้

[Check point#6] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดง Repository ที่มี Docker image (<username>/lab8)

The screenshot shows the Docker Hub interface. In the search bar, 'manitchaya' is selected. The 'Repositories' tab is active, showing a list of repositories under the 'manitchaya' user. One repository, 'manitchaya/lab8', is listed with the following details: Last Pushed 5 minutes ago, Contains IMAGE, Public visibility, and Inactive status. To the right of the repository list, there are sections for creating an organization and managing users.

```
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ touch Dockerfile
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ docker build -t manitchaya/lab8 .
[+] Building 0.1s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> [internal] load .dockerrcignore
=> [internal] load .dockerrc
=> [internal] load metadata for docker.io/library/busybox:latest
=> CACHED [1/1] FROM docker.io/library/busybox
=> exporting to image
=> => exporting layers
=> => writing image sha256:816c83893c7170c6317a9baf8963957c1447184f3462bf5a7e0e548a5cb2041f
[ => => naming to docker.io/manitchaya/lab8
[

|Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ docker run manitchaya/lab8
"Manitchaya Thamanunpongsa 653380323-8"
MacBook-Air--Manitchaya:Lab8_3 manitchaya$ docker push manitchaya/lab8
Using default tag: latest
The push refers to repository [docker.io/manitchaya/lab8]
613e5fc506b9: Mounted from library/busybox
latest: digest: sha256:61a337538f8946b495b65e8d50bc9db8d19a0de82f6caf2bbe90d11d2be4c07f size: 527
|MacBook-Air--Manitchaya:Lab8_3 manitchaya$
```

Lab Worksheet

แบบฝึกปฏิบัติที่ 8.4: การ Build และ Update แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
2. ทำการ Clone ซอฟต์แวร์โค้ดของเว็บแอปพลิเคชันจาก GitHub repository <https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง

```
$ git clone https://github.com/docker/getting-started.git
```
3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

[Check point#7] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงที่อยู่ของ Source code ที่ Clone มาและเนื้อหาของไฟล์ package.json

```
1 {
  "name": "101-app",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "prettify": "prettier -l --write \"**/*.js\"",
    "test": "jest",
    "dev": "nodemon src/index.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "mysql2": "^2.3.3",
    "sqlite3": "5.1.2",
    "uuid": "3.0.0",
    "wait-port": "1.0.4"
  },
  "resolutions": {
    "ansi-regex": "5.0.1"
  },
  "prettier": {
    "trailingComma": "all",
    "tabWidth": 4,
    "useTabs": false,
    "semi": true,
    "singleQuote": true
  },
  "devDependencies": {
    "jest": "^29.3.1",
    "nodemon": "2.0.20",
    "prettier": "^2.7.1"
  }
}
```

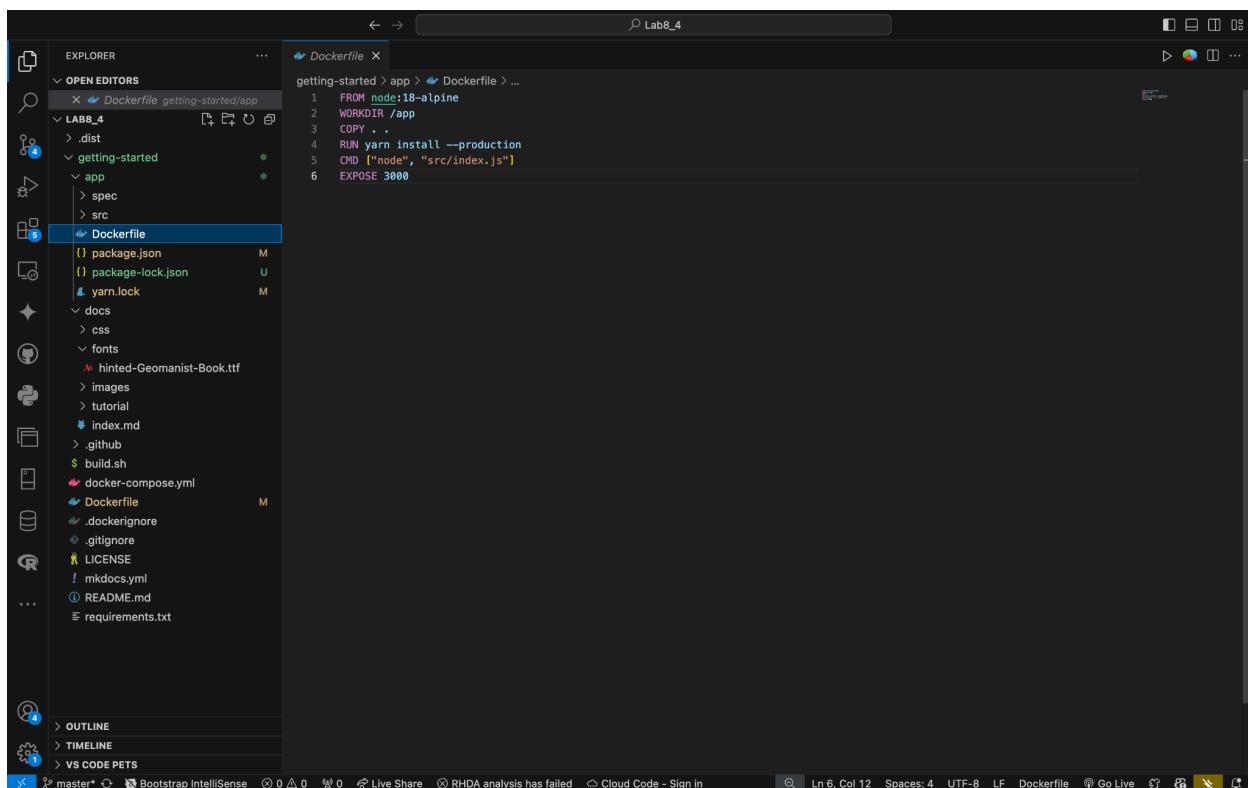
Lab Worksheet

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปในไฟล์

```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
```

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น myapp_รหัสสั้นๆ ไม่มีขีด
\$ docker build -t <myapp_รหัสสั้นๆ ไม่มีขีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ



The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure. The root folder contains a `Dockerfile`, `package.json`, `package-lock.json`, `yarn.lock`, `docs` (with `css`, `fonts`, `index.md`, `.github`, `build.sh`, `Dockerfile`, `.dockerignore`, `.gitignore`, `LICENSE`, `mkdocs.yml`, `README.md`, and `requirements.txt`), and `getting-started` (with `app` containing `spec` and `src`).
- Editor View:** Displays the `Dockerfile` content in the `getting-started/app/Dockerfile` tab:

```
1 FROM node:18-alpine
2 WORKDIR /app
3 COPY . .
4 RUN yarn install --production
5 CMD ["node", "src/index.js"]
6 EXPOSE 3000
```
- Bottom Status Bar:** Shows the current file is `Dockerfile`, line 6, column 12, spaces: 4, encoding: UTF-8, and mode: LF.

Lab Worksheet

```
MacBook-Air--Manitchaya:app chonthicha$ docker build -t myapp_6533803238 .
[+] Building 19.2s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> [internal] load build context
=> => transferring context: 4.80MB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY .
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => writing image sha256:ba4c6f2a0eb12263850f9ded9fb902bab9f1f67efe8a79a6456c4da6421a36eb
=> => naming to docker.io/library/myapp_6533803238

Use 'docker scan' to run Snyk tests against images to find vulnerabilities and learn how to fix them
MacBook-Air--Manitchaya:app chonthicha$
```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

\$ docker run -dp 3000:3000 <myapp_รหัส> ไม่มีชีด>

7. เปิด Browser ไปที่ URL = <http://localhost:3000>

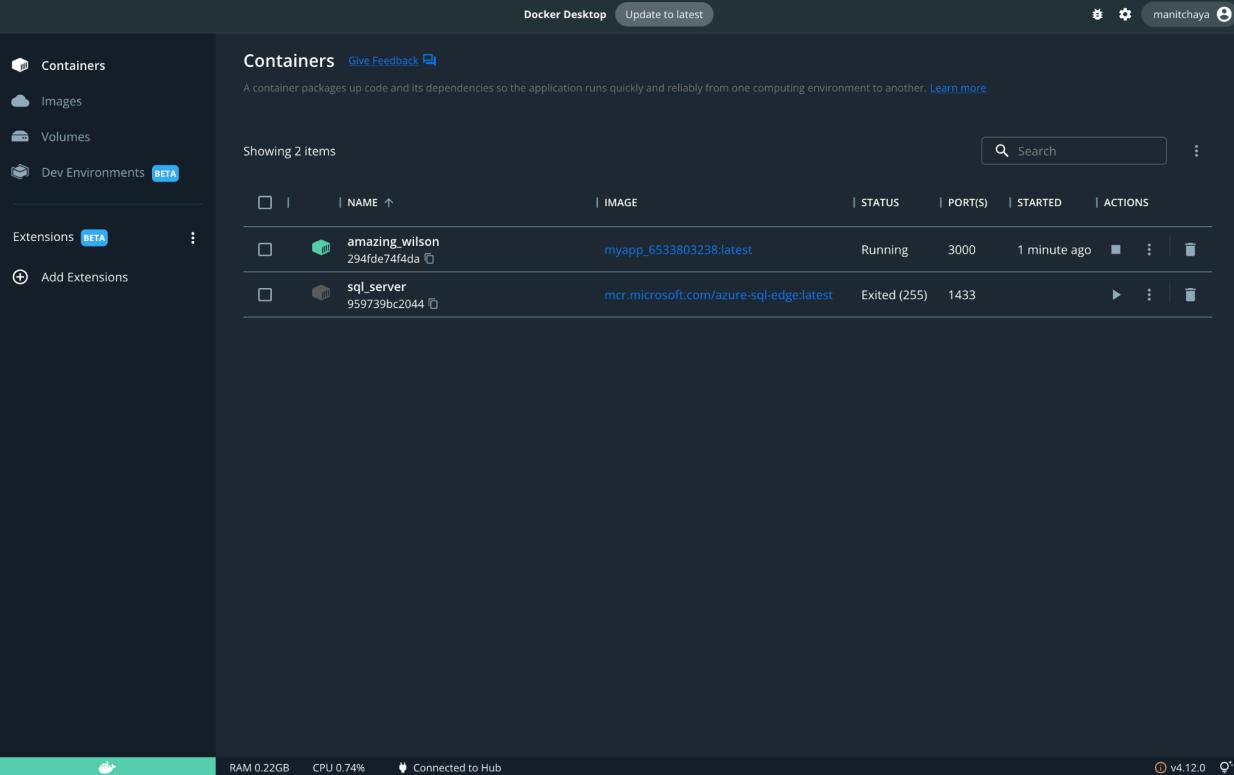
[Check point#9] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

```
MacBook-Air--Manitchaya:getting-started chonthicha$ docker run -dp 3000:3000 myapp_6533803238
74def5af77fc986a1ce3346a90b97ea1405b7af2999a3e214ff0a6041c9ab8
MacBook-Air--Manitchaya:getting-started chonthicha$
```

CP353004/SC313 004 Software Engineering (2/2567)

ผศ. ดร. ชิตสุชา สุ่มเล็ก

Lab Worksheet



Docker Desktop Update to latest manitchaya

Containers [Give Feedback](#)

A container packages up code and its dependencies so the application runs quickly and reliably from one computing environment to another. [Learn more](#)

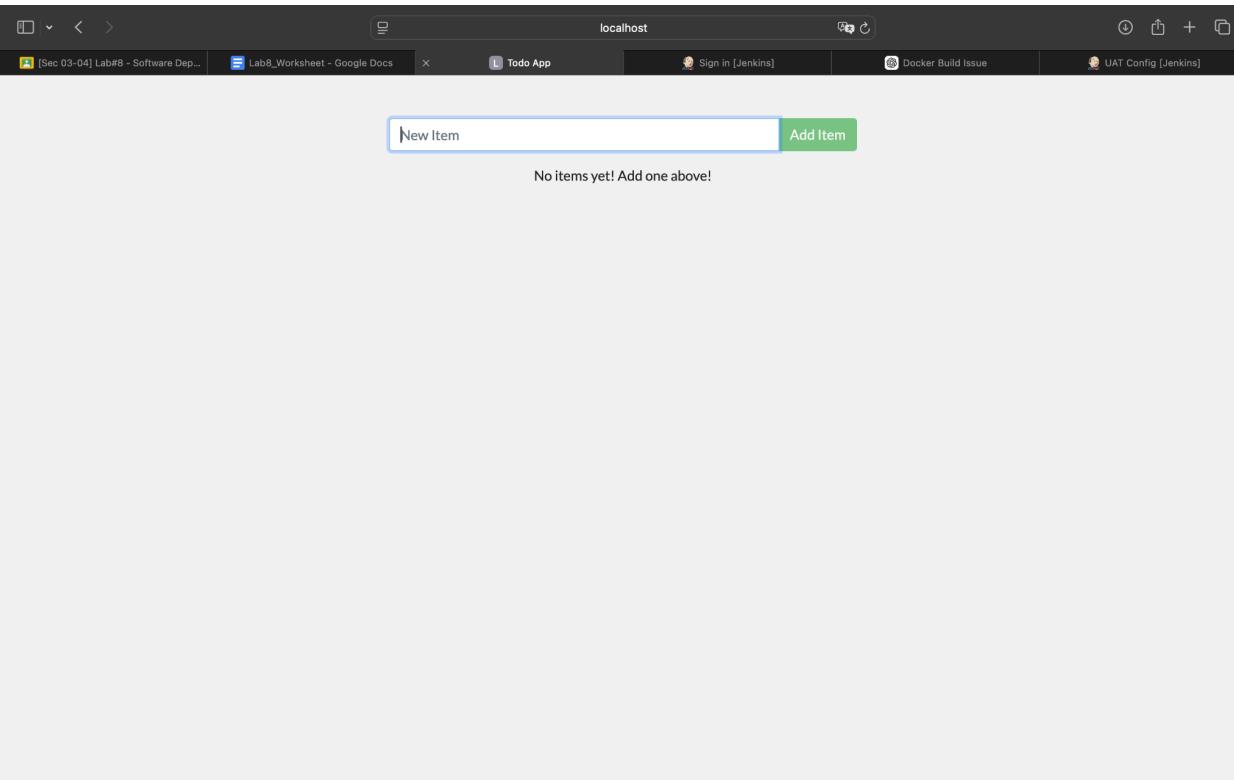
Showing 2 items

| | NAME ↑ | IMAGE | STATUS | PORT(S) | STARTED | ACTIONS |
|--------------------------|--------------------------------|---|--------------|---------|--------------|---|
| <input type="checkbox"/> | amazing_wilson 294fde74f4da | myapp_6533803238:latest | Running | 3000 | 1 minute ago | Stop ⋮ Remove |
| <input type="checkbox"/> | sql_server 959739bc2044 | mcr.microsoft.com/azure-sql-edge:latest | Exited (255) | 1433 | | Start ⋮ Remove |

Extensions [BETA](#) ...

+ Add Extensions

RAM 0.22GB CPU 0.74% Connected to Hub v4.12.0



localhost

New Item Add Item

No items yet! Add one above!

[Sec 03-04] Lab#8 - Software Dep... Lab8_Worksheet - Google Docs Todo App Sign in [Jenkins] Docker Build Issue UAT Config [Jenkins]

Lab Worksheet

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้

- เปิดไฟล์ src/static/js/app.js ด้วย Editor และแก้ไขบรรทัดที่ 56 จาก

< p className="text-center">No items yet! Add one above! </p> เป็น

< p className="text-center">**There is no TODO item.**

Please add one to the list. By ชื่อและนามสกุลของนักศึกษา

</p>

- Save ไฟล์ให้เรียบร้อย

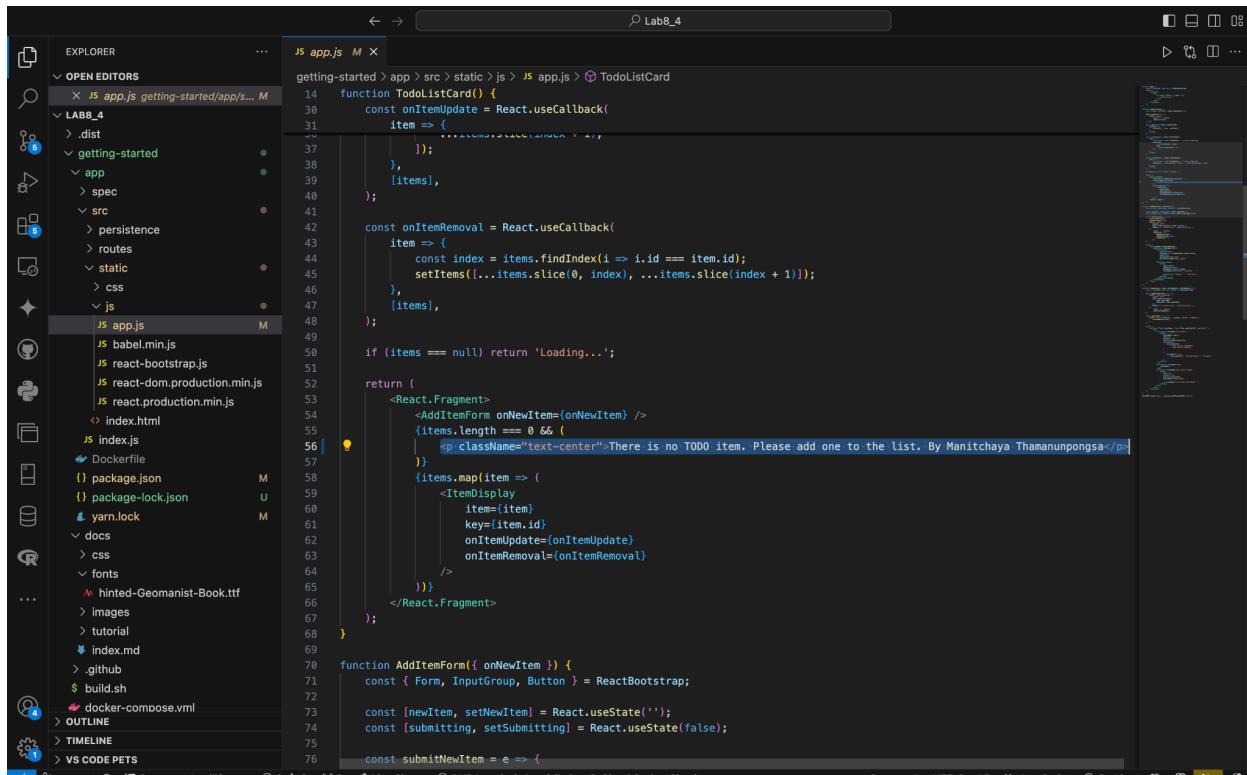
9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

[Check point#10] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุเมลล์ก Lab Worksheet



```
JS app.js M X
getting-started > app > src > static > js > JS app.js > TodoListCard
  14 function TodoListCard() {
  15   const onItemUpdate = React.useCallback(
  16     item => {
  17       const index = items.findIndex(i => i.id === item.id);
  18       setItems([...items.slice(0, index), ...items.slice(index + 1)]);
  19     },
  20     [items]
  21   );
  22
  23   const onItemRemoval = React.useCallback(
  24     item => {
  25       const index = items.findIndex(i => i.id === item.id);
  26       setItems([...items.slice(0, index), ...items.slice(index + 1)]);
  27     },
  28     [items]
  29   );
  30
  31   if (items === null) return 'Loading...';
  32
  33   return (
  34     <React.Fragment>
  35       <AddItemForm onNewItem={onNewItem} />
  36       {items.length === 0 && (
  37         <p className="text-center">There is no TODO item. Please add one to the list. By Manitchaya Thamanunpongsa</p>
  38       )}
  39       {items.map(item => (
  40         <ItemDisplay
  41           item={item}
  42           key={item.id}
  43           onItemUpdate={onItemUpdate}
  44           onItemRemoval={onItemRemoval}
  45         />
  46       ))}
  47     </React.Fragment>
  48   );
  49
  50   function AddItemForm({ onNewItem }) {
  51     const [Form, InputGroup, Button] = ReactBootstrap;
  52
  53     const [newItem, setNewItem] = React.useState('');
  54     const [submitting, setSubmitting] = React.useState(false);
  55
  56     const submitNewItem = e => {
  57       e.preventDefault();
  58       if (newItem === '') return;
  59
  60       onNewItem(newItem);
  61       setNewItem('');
  62       setSubmitting(true);
  63       setTimeout(() => setSubmitting(false), 500);
  64     };
  65
  66     return (
  67       <Form>
  68         <InputGroup>
  69           <Input type="text" value={newItem} onChange={e => setNewItem(e.target.value)} />
  70           <Button type="button" onClick={submitNewItem}>Add</Button>
  71         </InputGroup>
  72       </Form>
  73     );
  74   }
  75
  76   const [onNewItem, setNewItem] = React.useState('');
  77   const [submitting, setSubmitting] = React.useState(false);
  78
  79   const submitNewItem = e => {
  80     e.preventDefault();
  81     if (onNewItem === '') return;
  82
  83     onNewItem(onNewItem);
  84     setNewItem('');
  85     setSubmitting(true);
  86     setTimeout(() => setSubmitting(false), 500);
  87   };
  88
  89   return (
  90     <Form>
  91       <InputGroup>
  92         <Input type="text" value={onNewItem} onChange={e => setNewItem(e.target.value)} />
  93         <Button type="button" onClick={submitNewItem}>Add</Button>
  94       </InputGroup>
  95     </Form>
  96   );
  97 }
  98
  99
 100
 101
 102
 103
 104
 105
 106
 107
 108
 109
 110
 111
 112
 113
 114
 115
 116
 117
 118
 119
 120
 121
 122
 123
 124
 125
 126
 127
 128
 129
 130
 131
 132
 133
 134
 135
 136
 137
 138
 139
 140
 141
 142
 143
 144
 145
 146
 147
 148
 149
 150
 151
 152
 153
 154
 155
 156
 157
 158
 159
 160
 161
 162
 163
 164
 165
 166
 167
 168
 169
 170
 171
 172
 173
 174
 175
 176
 177
 178
 179
 180
 181
 182
 183
 184
 185
 186
 187
 188
 189
 190
 191
 192
 193
 194
 195
 196
 197
 198
 199
 200
 201
 202
 203
 204
 205
 206
 207
 208
 209
 210
 211
 212
 213
 214
 215
 216
 217
 218
 219
 220
 221
 222
 223
 224
 225
 226
 227
 228
 229
 230
 231
 232
 233
 234
 235
 236
 237
 238
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293
 294
 295
 296
 297
 298
 299
 300
 301
 302
 303
 304
 305
 306
 307
 308
 309
 310
 311
 312
 313
 314
 315
 316
 317
 318
 319
 320
 321
 322
 323
 324
 325
 326
 327
 328
 329
 330
 331
 332
 333
 334
 335
 336
 337
 338
 339
 340
 341
 342
 343
 344
 345
 346
 347
 348
 349
 350
 351
 352
 353
 354
 355
 356
 357
 358
 359
 360
 361
 362
 363
 364
 365
 366
 367
 368
 369
 370
 371
 372
 373
 374
 375
 376
 377
 378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431
 432
 433
 434
 435
 436
 437
 438
 439
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450
 451
 452
 453
 454
 455
 456
 457
 458
 459
 460
 461
 462
 463
 464
 465
 466
 467
 468
 469
 470
 471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500
 501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522
 523
 524
 525
 526
 527
 528
 529
 530
 531
 532
 533
 534
 535
 536
 537
 538
 539
 540
 541
 542
 543
 544
 545
 546
 547
 548
 549
 550
 551
 552
 553
 554
 555
 556
 557
 558
 559
 559
 560
 561
 562
 563
 564
 565
 566
 567
 568
 569
 570
 571
 572
 573
 574
 575
 576
 577
 578
 579
 580
 581
 582
 583
 584
 585
 586
 587
 588
 589
 589
 590
 591
 592
 593
 594
 595
 596
 597
 598
 599
 599
 600
 601
 602
 603
 604
 605
 606
 607
 608
 609
 609
 610
 611
 612
 613
 614
 615
 616
 617
 618
 619
 619
 620
 621
 622
 623
 624
 625
 626
 627
 628
 629
 629
 630
 631
 632
 633
 634
 635
 636
 637
 638
 639
 639
 640
 641
 642
 643
 644
 645
 646
 647
 648
 649
 649
 650
 651
 652
 653
 654
 655
 656
 657
 658
 659
 659
 660
 661
 662
 663
 664
 665
 666
 667
 668
 669
 669
 670
 671
 672
 673
 674
 675
 676
 677
 678
 679
 679
 680
 681
 682
 683
 684
 685
 686
 687
 688
 689
 689
 690
 691
 692
 693
 694
 695
 696
 697
 698
 699
 699
 700
 701
 702
 703
 704
 705
 706
 707
 708
 709
 709
 710
 711
 712
 713
 714
 715
 716
 717
 718
 719
 719
 720
 721
 722
 723
 724
 725
 726
 727
 728
 729
 729
 730
 731
 732
 733
 734
 735
 736
 737
 737
 738
 739
 739
 740
 741
 742
 743
 744
 745
 746
 747
 748
 749
 749
 750
 751
 752
 753
 754
 755
 756
 757
 758
 759
 759
 760
 761
 762
 763
 764
 765
 766
 767
 768
 769
 769
 770
 771
 772
 773
 774
 775
 776
 777
 778
 779
 779
 780
 781
 782
 783
 784
 785
 786
 787
 788
 789
 789
 790
 791
 792
 793
 794
 795
 796
 797
 798
 799
 799
 800
 801
 802
 803
 804
 805
 806
 807
 808
 809
 809
 810
 811
 812
 813
 814
 815
 816
 817
 818
 819
 819
 820
 821
 822
 823
 824
 825
 826
 827
 828
 829
 829
 830
 831
 832
 833
 834
 835
 836
 837
 838
 839
 839
 840
 841
 842
 843
 844
 845
 846
 847
 848
 849
 849
 850
 851
 852
 853
 854
 855
 856
 857
 858
 859
 859
 860
 861
 862
 863
 864
 865
 866
 867
 868
 869
 869
 870
 871
 872
 873
 874
 875
 876
 877
 878
 879
 879
 880
 881
 882
 883
 884
 885
 886
 887
 888
 889
 889
 890
 891
 892
 893
 894
 895
 896
 897
 898
 899
 899
 900
 901
 902
 903
 904
 905
 906
 907
 908
 909
 909
 910
 911
 912
 913
 914
 915
 915
 916
 917
 918
 919
 919
 920
 921
 922
 923
 924
 925
 926
 927
 928
 929
 929
 930
 931
 932
 933
 934
 935
 936
 937
 938
 939
 939
 940
 941
 942
 943
 944
 945
 946
 947
 948
 949
 949
 950
 951
 952
 953
 954
 955
 956
 957
 958
 959
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968
 969
 969
 970
 971
 972
 973
 974
 975
 976
 977
 978
 979
 979
 980
 981
 982
 983
 984
 985
 986
 987
 988
 989
 989
 990
 991
 992
 993
 994
 995
 996
 997
 997
 998
 999
 999
 1000
 1000
 1001
 1002
 1003
 1004
 1005
 1006
 1007
 1008
 1009
 1009
 1010
 1011
 1012
 1013
 1014
 1015
 1016
 1017
 1018
 1019
 1019
 1020
 1021
 1022
 1023
 1024
 1025
 1026
 1027
 1028
 1029
 1029
 1030
 1031
 1032
 1033
 1034
 1035
 1036
 1037
 1038
 1039
 1039
 1040
 1041
 1042
 1043
 1044
 1045
 1046
 1047
 1048
 1049
 1049
 1050
 1051
 1052
 1053
 1054
 1055
 1056
 1057
 1058
 1059
 1059
 1060
 1061
 1062
 1063
 1064
 1065
 1066
 1067
 1068
 1069
 1069
 1070
 1071
 1072
 1073
 1074
 1075
 1076
 1077
 1078
 1079
 1079
 1080
 1081
 1082
 1083
 1084
 1085
 1086
 1087
 1088
 1089
 1089
 1090
 1091
 1092
 1093
 1094
 1095
 1096
 1097
 1097
 1098
 1099
 1099
 1100
 1101
 1102
 1103
 1104
 1105
 1106
 1107
 1108
 1109
 1109
 1110
 1111
 1112
 1113
 1114
 1115
 1116
 1117
 1118
 1119
 1119
 1120
 1121
 1122
 1123
 1124
 1125
 1126
 1127
 1128
 1129
 1129
 1130
 1131
 1132
 1133
 1134
 1135
 1136
 1137
 1138
 1139
 1139
 1140
 1141
 1142
 1143
 1144
 1145
 1146
 1147
 1148
 1149
 1149
 1150
 1151
 1152
 1153
 1154
 1155
 1156
 1157
 1158
 1159
 1159
 1160
 1161
 1162
 1163
 1164
 1165
 1166
 1167
 1168
 1169
 1169
 1170
 1171
 1172
 1173
 1174
 1175
 1176
 1177
 1178
 1178
 1179
 1180
 1181
 1182
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1189
 1190
 1191
 1192
 1193
 1194
 1195
 1196
 1197
 1198
 1198
 1199
 1199
 1200
 1201
 1202
 1203
 1204
 1205
 1206
 1207
 1208
 1209
 1209
 1210
 1211
 1212
 1213
 1214
 1215
 1216
 1217
 1218
 1219
 1219
 1220
 1221
 1222
 1223
 1224
 1225
 1226
 1227
 1228
 1229
 1229
 1230
 1231
 1232
 1233
 1234
 1235
 1236
 1237
 1238
 1239
 1239
 1240
 1241
 1242
 1243
 1244
 1245
 1246
 1247
 1248
 1249
 1249
 1250
 1251
 1252
 1253
 1254
 1255
 1256
 1257
 1258
 1259
 1259
 1260
 1261
 1262
 1263
 1264
 1265
 1266
 1267
 1268
 1269
 1269
 1270
 1271
 1272
 1273
 1274
 1275
 1276
 1277
 1278
 1279
 1279
 1280
 1281
 1282
 1283
 1284
 1285
 1286
 1287
 1288
 1289
 1289
 1290
 1291
 1292
 1293
 1294
 1295
 1296
 1297
 1297
 1298
 1299
 1299
 1300
 1301
 1302
 1303
 1304
 1305
 1306
 1307
 1308
 1309
 1309
 1310
 1311
 1312
 1313
 1314
 1315
 1316
 1317
 1318
 1319
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1329
 1330
 1331
 1332
 1333
 1334
 1335
 1336
 1337
 1338
 1339
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389
 1389
 1390
 1391
 1392
 1393
 1394
 1395
 1396
 1397
 1398
 1398
 1399
 1400
 1401
 1402
 1403
 1404
 1405
 1406
 1407
 1408
 1409
 1409
 1410
 1411
 1412
 1413
 1414
 1415
 1416
 1417
 1418
 1418
 1419
 1420
 1421
 1422
 1423
 1424
 1425
 1426
 1427
 1428
 1429
 1429
 1430
 1431
 1432
 1433
 1434
 1435
 1436
 1437
 1438
 1439
 1439
 1440
 1441
 1442
 1443
 1444
 1445
 1446
 1447
 1448
 1449
 1449
 1450
 1451
 1452
 1453
 1454
 1455
 1456
 1457
 1458
 1459
 1459
 1460
 1461
 1462
 1463
 1464
 1465
 1466
 1467
 1468
 1469
 1469
 1470
 1471
 1472
 1473
 1474
 1475
 1476
 1477
 1478
 1478
 1479
 1480
 1481
 1482
 1483
 1484
 1485
 1486
 1487
 1488
 1489
 1489
 1490
 1491
 1492
 1493
 1494
 1495
 1496
 1497
 1498
 1498
 1499
 1500
 1501
 1502
 1503
 1504
 1505
 1506
 1507
 1508
 1509
 1509
 1510
 1511
 1512
 1513
 1514
 1515
 1516
 1517
 1518
 1519
 1519
 1520
 1521
 1522
 1523
 1524
 1525
 1526
 1527
 1528
 1529
 1529
 1530
 1531
 1532
 1533
 1534
 1535
 1536
 1537
 1538
 1539
 1539
 1540
 1541
 1542
 1543
 1544
 1545
 1546
 1547
 1548
 1549
 1549
 1550
 1551
 1552
 1553
 1554
 1555
 1556
 1557
 1558
 1559
 1559
 1560
 1561
 1562
 1563
 1564
 1565
 1566
 1567
 1568
 1569
 1569
 1570
 1571
 1572
 1573
 1574
 1575
 1576
 1577
 1578
 1578
 1579
 1580
 1581
 1582
 1583
 1584
 1585
 1586
 1587
 1588
 1589
 1589
 1590
 1591
 1592
 1593
 1594
 1595
 1596
 1597
 1598
 1599
 1599
 1600
 1601
 1602
 1603
 1604
 1605
 1606
 1607
 1608
 1609
 1609
 1610
 1611
 1612
 1613
 1614
 1615
 1616
 1617
 1618
 1619
 1619
 1620
 1621
 1622
 1623
 1624
 1625
 1626
 1627
 1628
 1629
 1629
 1630
 1631
 1632
 1633
 1634
 1635
 1636
 1637
 1638
 1639
 1639
 1640
 1641
 1642
 1643
 1644
 1645
 1646
 1647
 1648
 1649
 1649
 1650
 1651
 1652
 1653
 1654
 1655
 1656
 1657
 1658
 1659
 1659
 1660
 1661
 1662
 1663
 1664
 1665
 1666
 1667
 1668
 1669
 1669
 1670
 1671
 1672
 1673
 1674
 1675
 1676
 1677
 1678
 1678
 1679
 1680
 1681
 1682
 1683
 1684
 1685
 1686
 1687
 1688
 1689
 1689
 1690
 1691
 1692
 1693
 1694
 1695
 1696
 1697
 1698
 1699
 1699
 1700
 1701
 1702
 1703
 1704
 1705
 1706
 1707
 1708
 1709
 1709
 1710
 1711
 1712
 1713
 1714
 1715
 1716
 1717
 1718
 1719
 1719
 1720
 1721
 1722
 1723
 1724
 1725
 1726
 1727
 1728
 1729
 1729
 1730
 1731
 1732
 1733
 1734
 1735
 1736
 1737
 1738
 1739
 1739
 1740
 1741
 1742
 1743
 1744
 1745
 1746
 1747
 1748
 1749
 1749
 1750
 1751
 1752
 1753
 1754
 1755
 1756
 1757
 1758
 1759
 1759
 1760
 1761
 1762
 1763
 1764
 1765
 1766
 1767
 1768
 1769
 1769
 1770
 1771
 1772
 1773
 1774
 1775
 1776
 1777
 1778
 1779
 1779
 1780
 1781
 1782
 1783
 1784
 1785
 1786
 1787
 1788
 1789
 1789
 1790
 1791
 1792
 1793
 1794
 1795
 1796
 1797
 1798
 1799
 1799
 1800
 1801
 1802
 1803
 1804
 1805
 1806
 1807
 1808
 1809
 1809
 18
```

Lab Worksheet

(1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร
failed to solve with frontend dockerfile.v0: failed to create LLB
definition: the Dockerfile cannot be empty.

หมายความว่า Dockerfile ที่ใช้ในการสร้าง Docker image ว่างเปล่าหรือ
ไม่มีเนื้อหาภายใน ซึ่งทำให้ Docker ไม่สามารถสร้าง image ได้จาก
Dockerfile ดังกล่าว

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออก
จากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

- i. ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
- ii. Copy หรือบันทึก Container ID ไว้
- iii. ใช้คำสั่ง \$ docker stop <Container ID ที่ต้องการจะลบ> เพื่อหยุดการทำงานของ Container ดังกล่าว
- iv. ใช้คำสั่ง \$ docker rm <Container ID ที่ต้องการจะลบ> เพื่อทำการลบ

b. ผ่าน Docker desktop

- i. ไปที่หน้าต่าง Containers
- ii. เลือกไอคอนถังขยะในแท็บของ Container ที่ต้องการจะลบ
- iii. ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับ
ข้อ 6

13. เปิด Browser ไปที่ URL = <http://localhost:3000>

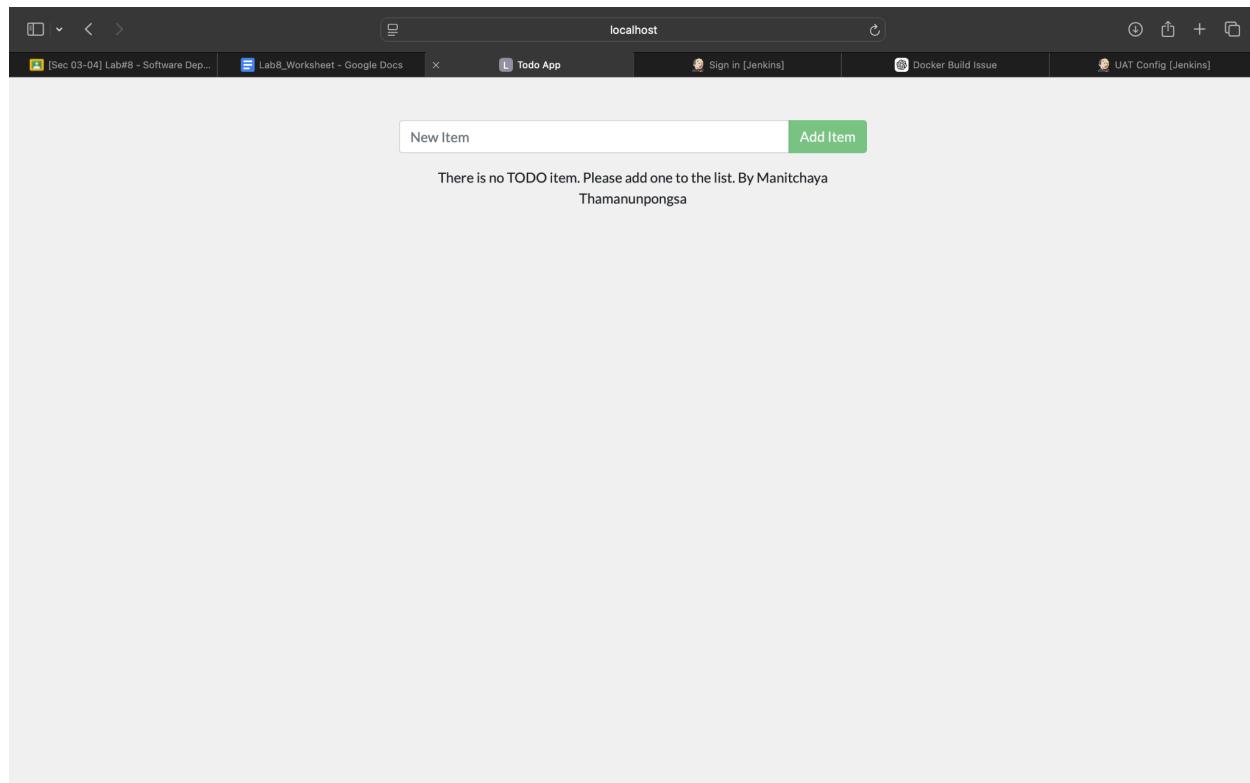
[Check point#11] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่
เกี่ยวข้อง) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ
Docker desktop

CP353004/SC313 004 Software Engineering (2/2567)

ผศ. ดร. ชิตสุชา สุเมลักษ์

Lab Worksheet

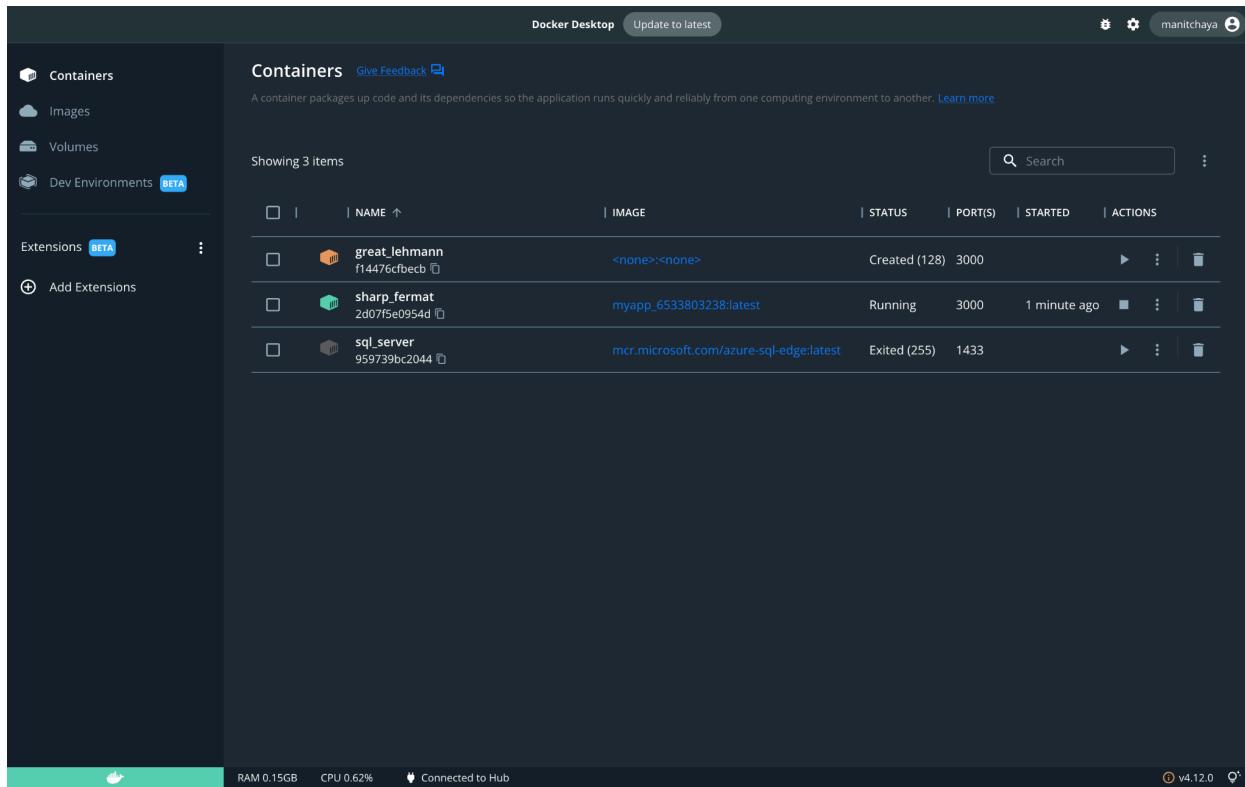
```
[MacBook-Air:~/Manitchaya:Lab8_4 chonthicha]$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
4ea3f95f8ed2        busybox            "sh"                   2 days ago         Up 9 minutes      0.0.0.0:1433->1433/tcp   sql_server
959739bc2844        mcr.microsoft.com/azure-sql-edge   "/opt/mssql/binperm..."   12 months ago       Up 9 minutes      1401/tcp, 0.0.0.0:1433->1433/tcp   sql_server
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$ docker stop 4ea3f95f8ed2
4ea3f95f8ed2
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$ docker rm 4ea3f95f8ed2
4ea3f95f8ed2
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$ docker ps -a
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
d289378d3b99        myapp   6533803238:latest   "docker-entrypoint.s..." 6 minutes ago     Exited (1) 6 minutes ago
c798558da10         myapp   6533803238:latest   "docker-entrypoint.s..." 12 minutes ago    Exited (1) 12 minutes ago
30051969d08b        myapp   6533803238:latest   "docker-entrypoint.s..." 17 minutes ago    Exited (1) 12 minutes ago
67ebfc4bb67         myapp   6533803238:latest   "docker-entrypoint.s..." 20 minutes ago   Exited (1) 20 minutes ago
db9b271ba5c2        myapp   6533803238:latest   "docker-entrypoint.s..." 21 minutes ago   Exited (1) 2 minutes ago
87311d1595f5        myapp   6533803238:latest   "docker-entrypoint.s..." 41 minutes ago   Exited (1) 12 minutes ago
a9671599467b        myapp   6533803238:latest   "docker-entrypoint.s..." 50 minutes ago   Exited (1) 12 minutes ago
8c2997366afdf       myapp   6533803238:latest   "docker-entrypoint.s..." 41 hours ago    Exited (1) 12 minutes ago
0f59cadad9fc       myapp   6533803238:latest   "docker-entrypoint.s..." 41 hours ago    Exited (1) 41 hours ago
e4d421d4e5f5        manitchaya:lab8   "/bin/sh -c 'echo 'M... 46 hours ago   Exited (0) 12 minutes ago
24a1137d5cc0        manitchaya_images   "/bin/sh -c 'echo 'M... 46 hours ago   Exited (0) 12 minutes ago
60c3fa1b62eb        busybox            "sh"                   2 days ago       Exited (0) 12 minutes ago
959739bc2845        mcr.microsoft.com/azure-sql-edge   "/opt/mssql/binperm..." 12 months ago      Up 12 minutes      1401/tcp, 0.0.0.0:1433->1433/tcp   sql_server
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$ docker run -dp 3000:3000 myapp_6533803238
98b0912a4711b2342f4204329c2fbae5b897274b0b0b70dcfaef63925a485
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS
959739bc2844        mcr.microsoft.com/azure-sql-edge   "/opt/mssql/binperm..."   12 months ago       Up 14 minutes      1401/tcp, 0.0.0.0:1433->1433/tcp   sql_server
MacBook-Air:~ Manitchaya:Lab8_4 chonthicha$
```



CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุ่มเล็ก

Lab Worksheet



แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop
2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต

```
$ docker run -p 8080:8080 -p 50000:50000
```

```
--restart=on-failure jenkins/jenkins:lts-jdk17
```

หรือ

```
$ docker run -p 8080:8080 -p 50000:50000
```

```
--restart=on-failure -v jenkins_home:/var/jenkins_home  
jenkins/jenkins:lts-jdk17
```

3. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก
- [Check point#12] Capture หน้าจอที่แสดงผล Admin password**

```
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

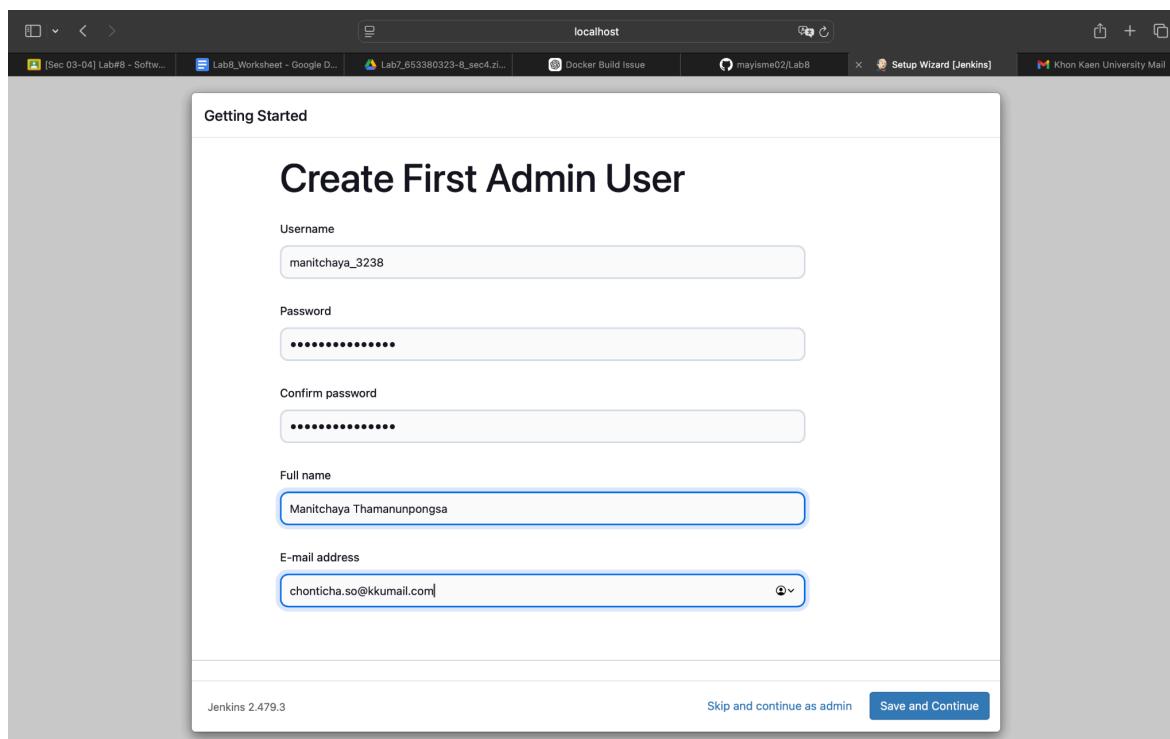
cfb23284de6e41afa87778ced630785c

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

*****
*****
```

4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิด
ราเวชอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษา
พร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุ่มเล็ก

Lab Worksheet

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

The screenshot shows the Jenkins dashboard at [localhost:8080](#). The top navigation bar includes links for 'Dashboard', 'Build History', 'Manage Jenkins', and 'My Views'. The main content area features a 'Welcome to Jenkins!' message and a 'Start building your software project' section. It includes sections for 'Build Queue' (empty), 'Create a job' (with a '+' button), 'Set up a distributed build' (with 'Set up an agent' and 'Configure a cloud' buttons), and 'Build Executor Status' (0/2). A 'Learn more about distributed builds' link is also present. The bottom right corner shows 'REST API' and 'Jenkins 2.479.3'.

9. เลือก Manage Jenkins และไปที่เมนู Plugins

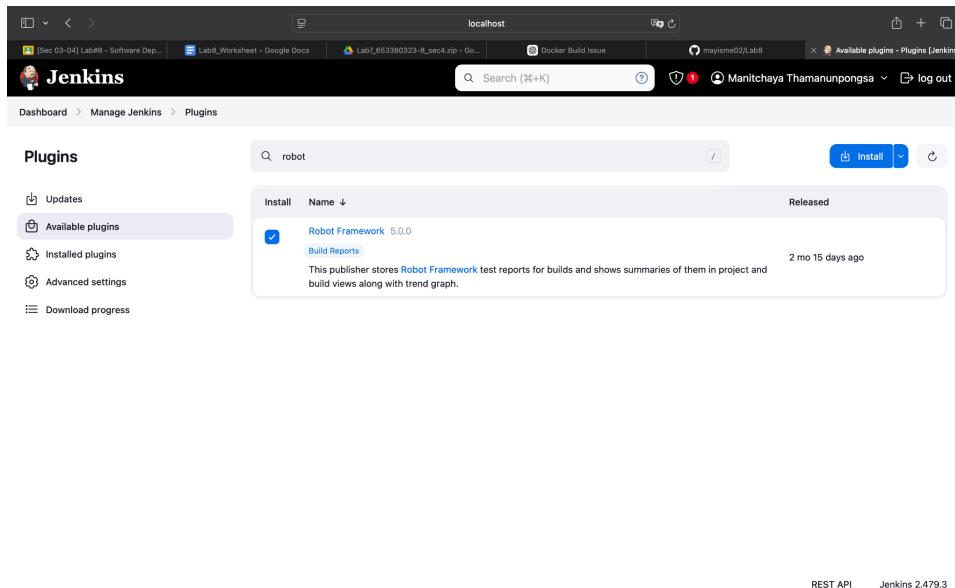
The screenshot shows the 'Manage Jenkins' page at [localhost:8080/jenkins/manage](#). The left sidebar has a 'Manage Jenkins' tab selected. The main area displays a message about a reverse proxy setup being broken, with 'More info' and 'Dismiss' buttons. Below this, there's a 'System Configuration' section with links for 'System', 'Nodes', 'Clouds', 'Tools', 'Plugins', 'Appearance', 'Security', 'Credentials', and 'Credential Providers'. Each link has a brief description and a corresponding icon.

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุ่มเล็ก

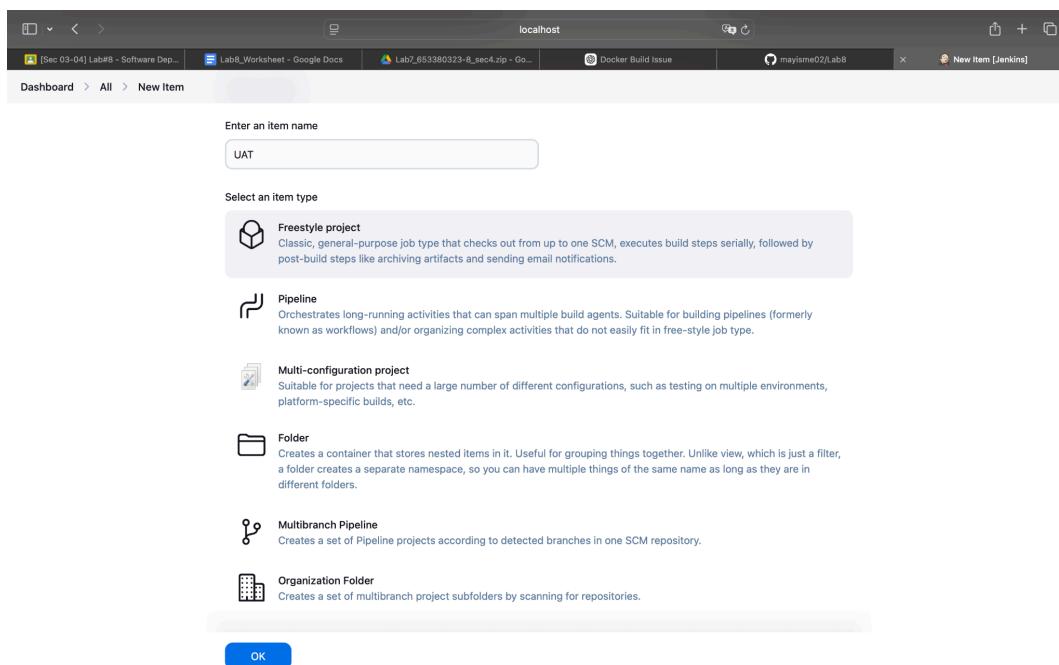
Lab Worksheet

10. ไปที่เมนู Available plugins และเลือกติดตั้ง Robotframework เพิ่มเติม



The screenshot shows the Jenkins Plugins page. In the top navigation bar, there are several tabs: [Sec 03-04] Lab#8 - Software Dep., Lab8_Worksheet - Google Docs, Lab7_653380323-B_sec4.zip - Go..., Docker Build Issue, mayisme02/Lab8, Available plugins - Plugins [Jenkins], and a user profile. Below the tabs, there is a search bar with the placeholder "Search (%+K)" and a "Log out" button. The main content area is titled "Plugins". On the left, there is a sidebar with links: "Updates", "Available plugins" (which is selected and highlighted in grey), "Installed plugins", "Advanced settings", and "Download progress". The main panel shows a table with columns: "Install", "Name", and "Released". A single row is visible for "Robot Framework 5.0.0" by "Build Reports". The row includes a "Install" button, a "Released" timestamp ("2 mo 15 days ago"), and a brief description: "This publisher stores Robot Framework test reports for builds and shows summaries of them in project and build views along with trend graph.". At the bottom right of the main panel, there are links for "REST API" and "Jenkins 2.479.3".

11. กลับไปที่หน้า Dashboard และสร้าง Pipeline อย่างง่าย โดยกำหนด New item เป็น Freestyle project และตั้งชื่อเป็น UAT



The screenshot shows the Jenkins "New Item" creation dialog. At the top, there is a header with tabs: "Dashboard", "All", and "New Item". Below the header, there is a text input field labeled "Enter an item name" containing the text "UAT". Underneath the name input, there is a section titled "Select an item type" with a list of options:

- Freestyle project**: Described as a "Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications."
- Pipeline**: Described as "Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type."
- Multi-configuration project**: Described as "Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc."
- Folder**: Described as "Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders."
- Multibranch Pipeline**: Described as "Creates a set of Pipeline projects according to detected branches in one SCM repository."
- Organization Folder**: Described as "Creates a set of multibranch project subfolders by scanning for repositories."

At the bottom of the dialog, there is a blue "OK" button.

12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้นตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

Description: Lab 8.5

GitHub project: กดเลือก และใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

Build Trigger: เลือกแบบ Build periodically และกำหนดให้ build ทุก 15 นาที

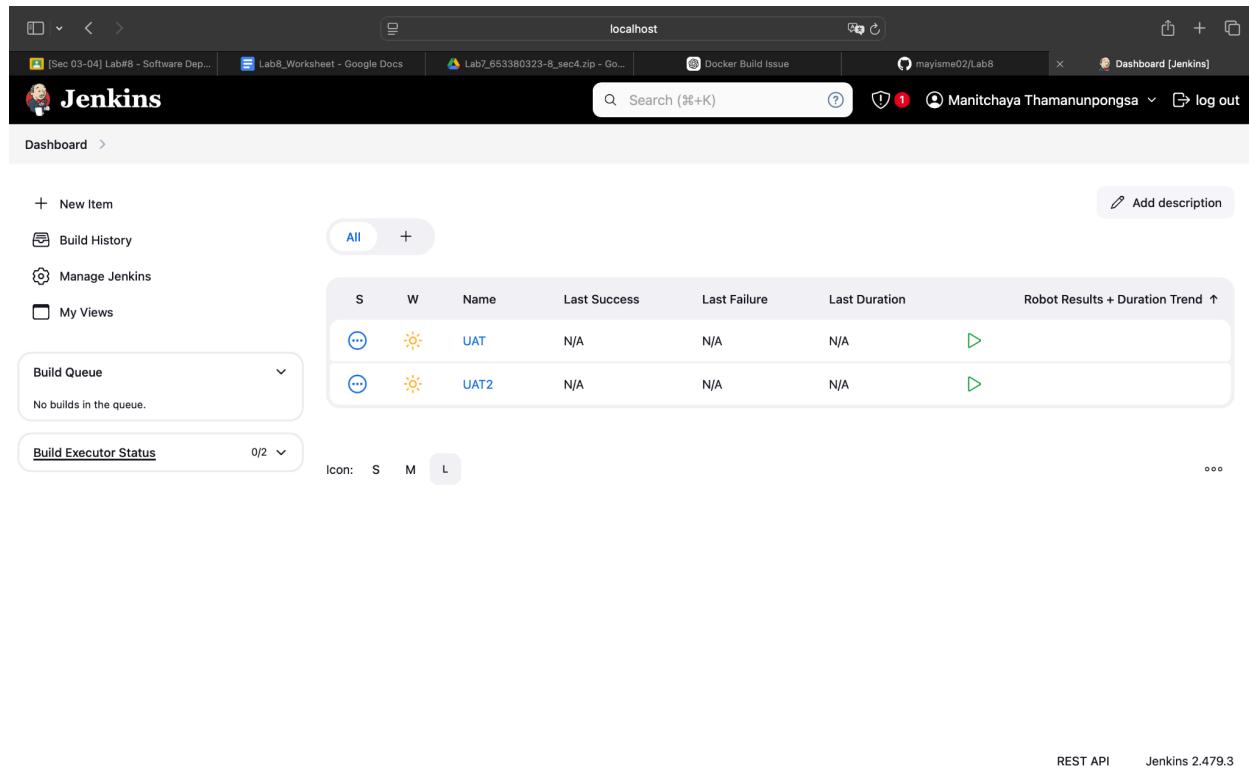
Build Steps: เลือก Execute shell และใส่คำสั่งในการรันไฟล์ .robot (หากไฟล์ไม่ได้อยู่ในหน้าแรกของ repository ให้ใส่ Path ไปถึงไฟล์ให้เรียบร้อยด้วย)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุ่มเล็ก

Lab Worksheet



The screenshot shows the Jenkins dashboard at localhost. The dashboard header includes tabs for [Sec 03-04] Lab#8 - Software Dep..., Lab8_Worksheet - Google Docs, Lab7_853380323-8_sec4.zip - Go..., Docker Build Issue, mayisme02/Lab8, and Dashboard [Jenkins]. The user Manitchaya Thamanunpongsa is logged in. The main area displays a table of builds:

| S | W | Name | Last Success | Last Failure | Last Duration | Robot Results + Duration Trend ↑ |
|-----|-----|------|--------------|--------------|---------------|----------------------------------|
| ... | ... | UAT | N/A | N/A | N/A | > |
| ... | ... | UAT2 | N/A | N/A | N/A | > |

On the left sidebar, there are links for New Item, Build History, Manage Jenkins, My Views, Build Queue (No builds in the queue), and Build Executor Status (0/2). At the bottom, there are icons for S, M, and L, and a REST API link.

(1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ [robot path/to/testsuite.robot](#)

Post-build action: เพิ่ม Publish Robot Framework test results -> ระบุไดเร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้ว นับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

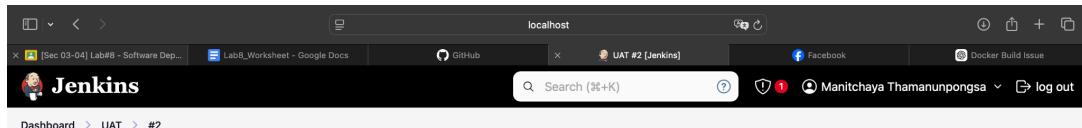
13. กด Apply และ Save
14. สั่ง Build Now

[Check point#15] Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุชา สุ่มเล็ก

Lab Worksheet



#2 (29 ม.ค. 2568 15:27:02)

Add description

Keep this build forever

</> Changes

Console Output

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Next Build

Started by user Manitchaya Thamanunpongsa

Started 1 hr 4 min ago

Took 1.6 sec

This run spent:

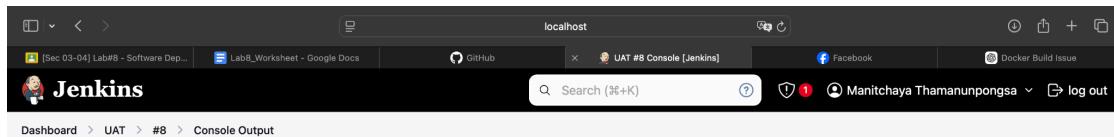
- 10 ms waiting;
- 1.6 sec build duration;
- 1.6 sec total from scheduled to completion.

Revision: a991c64b1abbe3727620af9cc04d11b8876abc0e
Repository: https://github.com/mayisme02/Lab8.git

- origin/main
- origin/main

</> No changes.

REST API Jenkins 2.479.3



Console Output

Download

Copy

View as plain text

</> Changes

Console Output

Edit Build Information

Delete build '#8'

Timings

Git Build Data

Previous Build

Next Build

Started by user Manitchaya Thamanunpongsa

Running as SYSTEM

Building in workspace /var/jenkins_home/workspace/UAT

The recommended git tool is: NONE

No credentials specified

No credentials specified

> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/UAT/.git # timeout=10

Fetching changes from 2 remote Git repositories

> git config remote.origin.url https://github.com/mayisme02/Lab8.git # timeout=10

Fetching upstream changes from https://github.com/mayisme02/Lab8.git

> git --version # timeout=10

> git --version # 'git' version 2.39.5'

> git fetch --tags --force --progress -- https://github.com/mayisme02/Lab8.git +refs/heads/*\:refs/remotes/origin/* # timeout=10

> git config remote.origin.url https://github.com/mayisme02/Lab8.git # timeout=10

Fetching upstream changes from https://github.com/mayisme02/Lab8.git

> git fetch --tags --force --progress -- https://github.com/mayisme02/Lab8.git +refs/heads/*\:refs/remotes/origin/* # timeout=10

Seen branch in repository origin/main

Seen branch in repository origin1/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

Checking out Revision a991c64b1abbe3727620af9cc04d11b8876abc0e (origin1/main, origin/main)

> git config core.sparsecheckout # timeout=10

> git checkout -f a991c64b1abbe3727620af9cc04d11b8876abc0e # timeout=10

Commit message: "Add files via upload"

> git rev-list --no-walk a991c64b1abbe3727620af9cc04d11b8876abc0e # timeout=10