

Lab Worksheet

ชื่อ-นามสกุล นางสาวชญาณิศ ธานี รหัสนักศึกษา 633021124-4 Section 1

Lab#8 – Software Deployment Using Docker

วัตถุประสงค์การเรียนรู้

1. ผู้เรียนสามารถอธิบายเกี่ยวกับ Software deployment ได้
2. ผู้เรียนสามารถสร้างและรัน Container จาก Docker image ได้
3. ผู้เรียนสามารถสร้าง Docker files และ Docker images ได้
4. ผู้เรียนสามารถนำซอฟต์แวร์ที่พัฒนาขึ้นให้สามารถรันบนสภาพแวดล้อมเดียวกันและทำงานร่วมกันกับสมาชิกในทีมพัฒนาซอฟต์แวร์ผ่าน Docker hub ได้
5. ผู้เรียนสามารถเริ่มต้นใช้งาน Jenkins เพื่อสร้าง Pipeline ในการ Deploy งานได้

Pre-requisite

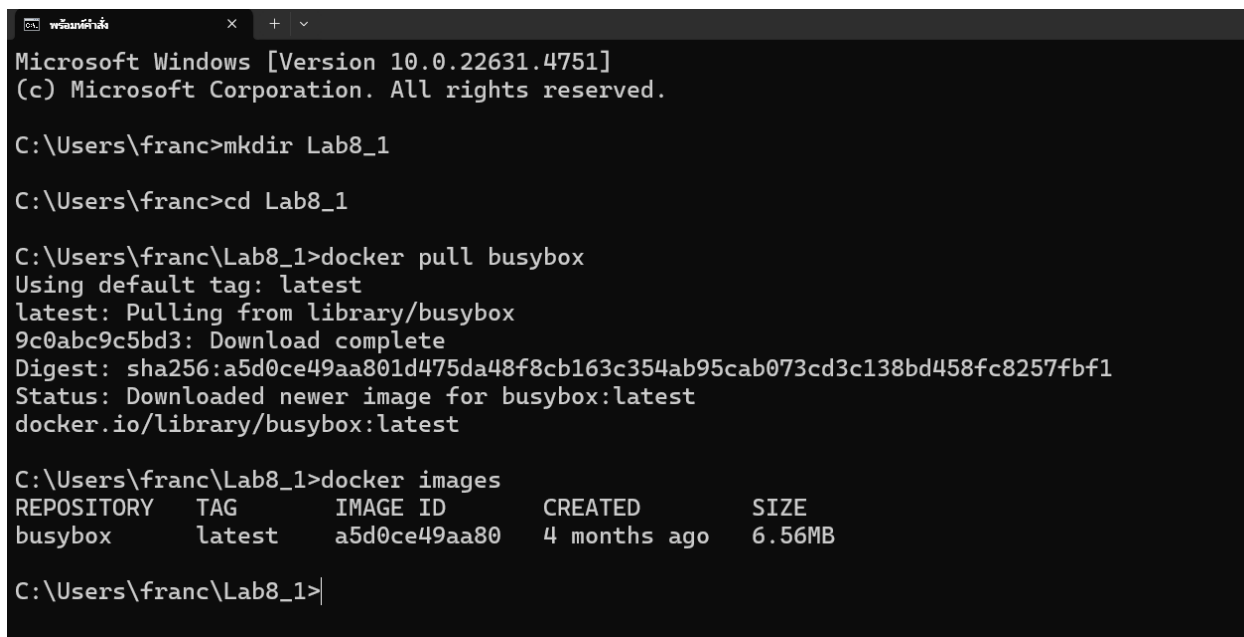
1. ติดตั้ง Docker desktop ลงบนเครื่องคอมพิวเตอร์ โดยดาวน์โหลดจาก <https://www.docker.com/get-started>
2. สร้าง Account บน Docker hub (<https://hub.docker.com/signup>)
3. กำหนดให้ \$ หมายถึง Command prompt และ <> หมายถึง ให้ป้อนค่าของพารามิเตอร์ที่กำหนด

แบบฝึกปฏิบัติที่ 8.1 Hello world - รัน Container จาก Docker image

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
1. เปิด Command line หรือ Terminal บน Docker Desktop จากนั้นสร้าง Directory ชื่อ Lab8_1
2. ย้ายตำแหน่งปัจจุบันไปที่ Lab8_1 เพื่อใช้เป็น Working directory
3. ป้อนคำสั่ง \$ docker pull busybox หรือ \$ sudo docker pull busybox สำหรับกรณีที่ติดปัญหา Permission denied
(หมายเหตุ: BusyBox เป็น software suite ที่รองรับคำสั่งบางอย่างบน Unix - <https://busybox.net>)
4. ป้อนคำสั่ง \$ docker images

Lab Worksheet

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



```

Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\franc>mkdir Lab8_1

C:\Users\franc>cd Lab8_1

C:\Users\franc\Lab8_1>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Download complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

C:\Users\franc\Lab8_1>docker images
REPOSITORY    TAG       IMAGE ID      CREATED        SIZE
busybox        latest    a5d0ce49aa80  4 months ago  6.56MB

C:\Users\franc\Lab8_1>
  
```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

ตอบ ค่าภายใต้คอลัมน์ Repository คือ busybox ซึ่งหมายถึงชื่อของ Docker image ที่ถูกดึง (pulled) มาและเก็บอยู่ใน local repository ของเครื่อง

(2) Tag ที่ใช้บ่งบอกถึงอะไร

ตอบ Tag ใช้บ่งบอก เวอร์ชัน (version) ของ Docker image ที่ถูกดึงมา ในกรณีนี้ที่ค่า Tag เป็น latest หมายความว่าได้ดึงเวอร์ชันล่าสุดของ BusyBox จาก Docker Hub มาใช้งาน

[Check point#1]

5. ป้อนคำสั่ง \$ docker run busybox
6. ป้อนคำสั่ง \$ docker run -it busybox sh
7. ป้อนคำสั่ง ls
8. ป้อนคำสั่ง ls -la
9. ป้อนคำสั่ง exit

Lab Worksheet

10. ป้อนคำสั่ง \$ docker run busybox echo "Hello ชื่อและนามสกุลของนักศึกษา from busybox"

11. ป้อนคำสั่ง \$ docker ps -a

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

```

C:\Users\franc\Lab8_1>docker run busybox
C:\Users\franc\Lab8_1>docker run -it busybox sh
/ # ls
total 48
drwxr-xr-x 1 root root 4096 Jan 28 12:00 .
drwxr-xr-x 1 root root 4096 Jan 28 12:00 ..
-rwxr-xr-x 1 root root 0 Jan 28 12:00 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 28 12:00 dev
drwxr-xr-x 1 root root 4096 Jan 28 12:00 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 239 root root 0 Jan 28 12:00 proc
drwx----- 1 root root 4096 Jan 28 12:01 root
dr-xr-xr-x 11 root root 0 Jan 28 12:00 sys
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

C:\Users\franc\Lab8_1>docker run busybox echo "Hello ชญณิศ ธาณี from busybox"
Hello ชญณิศ ธาณี from busybox

C:\Users\franc\Lab8_1>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
c217b25dd43c   busybox   "echo 'Hello ชญณิศ ...'" 17 seconds ago Exited (0) 16 seconds ago           nostalgic_lampport
debcc84d0d3a   busybox   "sh"                    3 minutes ago Exited (0) About a minute ago           nostalgic_satoshi
491c0f31a3ff   busybox   "sh"                    3 minutes ago Exited (0) 3 minutes ago           nostalgic_keen_davinci
keen_davinci
  
```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

ตอบ ช่วยให้โต้ตอบกับ Container ผ่าน terminal ได้

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

ตอบ Container ยังรันอยู่หรือหยุดไปแล้ว พร้อมระบุเวลาที่ล่าสุดที่มีการเปลี่ยนแปลงสถานะ

[Check point#2]

Lab Worksheet

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

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

```

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

C:\Users\franc\Lab8_1>docker run busybox echo "Hello ชญณิ ศ ธาณี from busybox"
Hello ชญณิ ศ ธาณี from busybox

C:\Users\franc\Lab8_1>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
c217b25dd43c   busybox   "echo 'Hello ชญณิ ศ ..." 17 seconds ago Exited (0) 16 seconds ago           great_lampport
debcc84d0d3a   busybox   "sh"                    3 minutes ago  Exited (0) About a minute ago
nostalgic_satoshi
491c0f31a3ff   busybox   "sh"                    3 minutes ago  Exited (0) 3 minutes ago
keen_davinci

C:\Users\franc\Lab8_1>docker rm c217b25dd43c
c217b25dd43c

C:\Users\franc\Lab8_1>

```

[Check point#3]

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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ (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

Lab Worksheet

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

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

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\franc>mkdir Lab8_2

C:\Users\franc>cd Lab8_2

C:\Users\franc\Lab8_2>docker build -t myfirstimage -f Dockerfile.swp .
[+] Building 2.3s (6/6) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 190B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                                0.0s
=> transferring context: 2B                                     0.0s
=> [1/1] FROM docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc82 2.2s
=> => resolve docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc82 2.2s
=> [auth] library/busybox:pull token for registry-1.docker.io 0.0s
=> exporting to image                                          0.1s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:5ee17acb6daf4901f73a89154f547096855b06cb3e21637557a216b7f5de122 0.0s
=> => exporting config sha256:08e1193b0800e4f77366341b3d0d6d265e2fde661337bd63ffb001da9620aebf 0.0s
=> => exporting attestation manifest sha256:9bd748037df4f145035106721d3300647e650d30a84624df5126652940bab49f7 0.0s
=> => exporting manifest list sha256:0f4f5f3978bc593478cf089449888cb6b562d2ddb9851176c181f6b1b34cc2f3 0.0s
=> => naming to docker.io/library/myfirstimage:latest          0.0s
=> => unpacking to docker.io/library/myfirstimage:latest       0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/u7ic95v6semczvbmjv7ybim

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)

C:\Users\franc\Lab8_2>docker run myfirstimage
"ชญาณ ศ ธานี 633021124-4 พ้อง"
```

- (1) คำสั่งที่ใช้ในการ run คือ

ตอบ คำสั่งที่ใช้ run Docker image ที่สร้างขึ้นคือ: docker run myfirstimage

Lab Worksheet

(2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

ตอบ คำสั่งที่ใช้ docker build -t myfirstimage .

- ใช้กำหนด ชื่อและแท็กของ Docker Image
- ทำให้อ้างอิง Image ได้ง่ายขึ้น
- ถ้าไม่ใช้ -t → Docker จะกำหนดเป็น IMAGE ID แทน

[Check point#4]

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

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ 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"
```

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

Lab Worksheet

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

```
Microsoft Windows [Version 10.0.22631.4751]
(c) Microsoft Corporation. All rights reserved.

C:\Users\franc>docker login
Authenticating with existing credentials...
Login Succeeded

C:\Users\franc>mkdir Lab8_3

C:\Users\franc>cd Lab8_3

C:\Users\franc\Lab8_3>docker build -t chayanitt/lab8 -f Dockerfile.swp .
[+] Building 2.4s (6/6) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 183B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS si 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                                0.0s
=> transferring context: 2B                                     0.0s
=> CACHED [1/1] FROM docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd 2.2s
=> => resolve docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc82 2.2s
=> [auth] library/busybox:pull token for registry-1.docker.io 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:53e9b3fc4d22a6319f2f0304cb8d23b37636b820887b42802804a53d622f5c22 0.0s
=> => exporting config sha256:022728f2881732c19d12eba85c4676f83ad90b257a579f3c77c0016ca20b7bd 0.0s
=> => exporting attestation manifest sha256:4786087d62d55afc4b4f68a5717c7417a9832aca95721cca0bd9e3e85ff00f63 0.0s
=> => exporting manifest list sha256:cc39f881bae33ef1a9bc2df8346e64738864364152d7e5a2724348114686f1a6 0.0s
=> => naming to docker.io/chayanitt/lab8:latest                 0.0s
=> => unpacking to docker.io/chayanitt/lab8:latest              0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/zvjz64q858tnmzcubdie2avgk

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)

C:\Users\franc\Lab8_3>docker run chayanitt/lab8
"Chayanit Thane 633021124-4"

C:\Users\franc\Lab8_3>
```

[Check point#5]

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

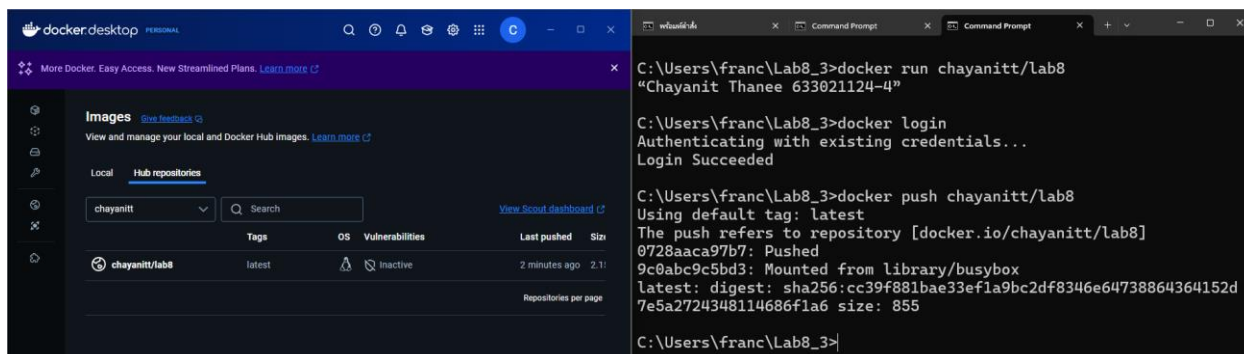
\$ docker login แล้วป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้คำสั่ง

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

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

Lab Worksheet

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



[Check point#6]

แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก 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

Lab Worksheet

```

(c) Microsoft Corporation. All rights reserved.

C:\Users\franc>mkdir Lab8_4

C:\Users\franc>cd Lab8_4

C:\Users\franc\Lab8_4>git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 980, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 980 (delta 5), reused 1 (delta 1), pack-reused 971 (from 2)
Receiving objects: 100% (980/980), 5.28 MiB | 15.63 MiB/s, done.
Resolving deltas: 100% (523/523), done.

C:\Users\franc\Lab8_4>cd getting-started/app

C:\Users\franc\Lab8_4\getting-started>ls
'ls' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\franc\Lab8_4\getting-started>dir
Volume in drive C is OS
Volume Serial Number is 8098-4CA5

Directory of C:\Users\franc\Lab8_4\getting-started\app

01/28/2025  08:32 PM  <DIR>          .
01/28/2025  08:32 PM  <DIR>          ..
01/28/2025  08:32 PM             678 package.json
01/28/2025  08:32 PM  <DIR>          spec
01/28/2025  08:32 PM  <DIR>          src
01/28/2025  08:32 PM      150,541 yarn.lock
                2 File(s)      151,219 bytes
                4 Dir(s)  22,838,206,464 bytes free

C:\Users\franc\Lab8_4\getting-started>app
  
```

Visual Studio Code Explorer shows the following structure:

- LAB8_4
 - getting-started
 - build.yml
 - workflows
 - app
 - spec
 - src
 - package.json
 - yarn.lock
 - docs
 - .dockerignore
 - .gitignore
 - build.sh
 - docker-compose.yml
 - Dockerfile
 - LICENSE
 - mkdocs.yml
 - README.md
 - requirements.txt

The package.json file content is as follows:

```

1  {
2    "name": "101-app",
3    "version": "1.0.0",
4    "main": "index.js",
5    "license": "MIT",
6    "scripts": {
7      "dev": "nodemon src/index.js",
8      "test": "jest",
9      "prettify": "prettier -l --write \"**/*.js\"",
10     "wait-port": "wait-port",
11     "dependencies": {
12       "express": "^4.18.2",
13       "mysql2": "^2.3.3",
14       "sqlite3": "^5.1.2",
15       "uuid": "^9.0.0",
16     },
17     "resolutions": {
18       "ansi-regex": "5.0.1"
19     },
20     "prettier": {
21       "trailingComma": "all",
22       "tabWidth": 4,
23       "useTabs": false,
24       "semi": true,
25       "singleQuote": true
26     },
27     "devDependencies": {
28       "jest": "^29.3.1",
29       "nodemon": "^2.0.20",
30       "prettier": "^2.7.1"
31     }
32   }
33
34
  
```

[Check point#7]

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดชื่อ image เป็น

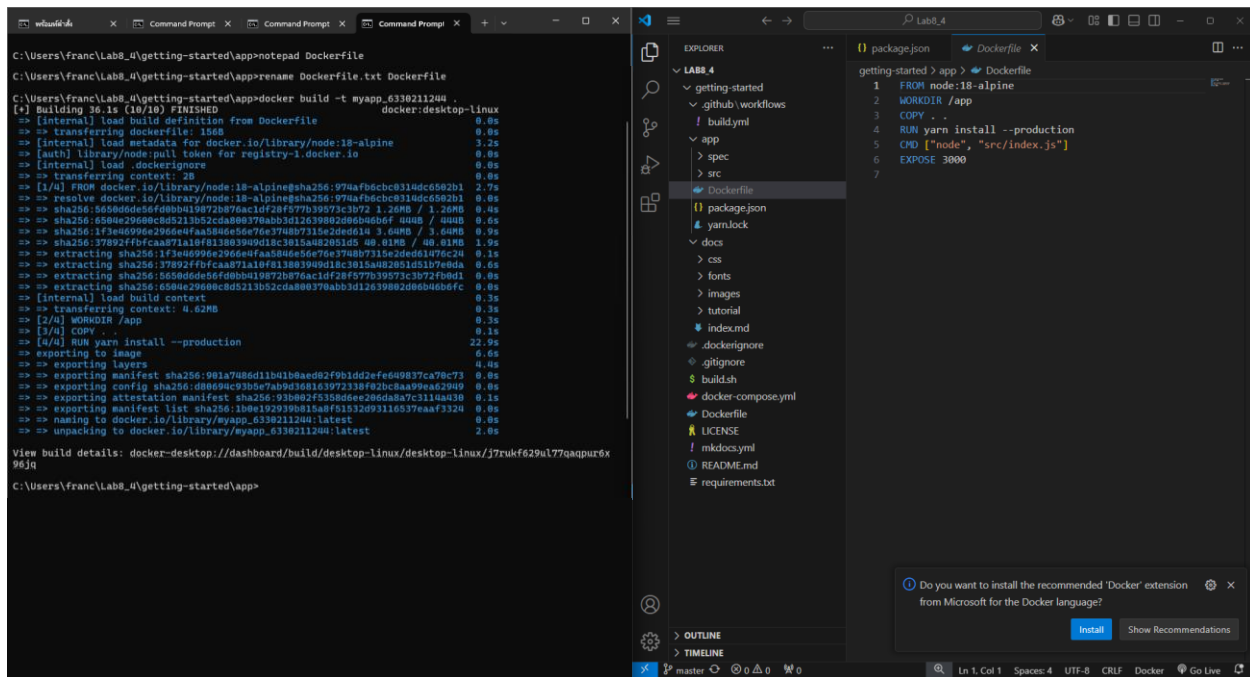
myapp_รหัสนศ. ไม่มีขีด

\$ docker build -t <myapp_รหัสนศ. ไม่มีขีด> .

[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ

Lab Worksheet



[Check point#8]

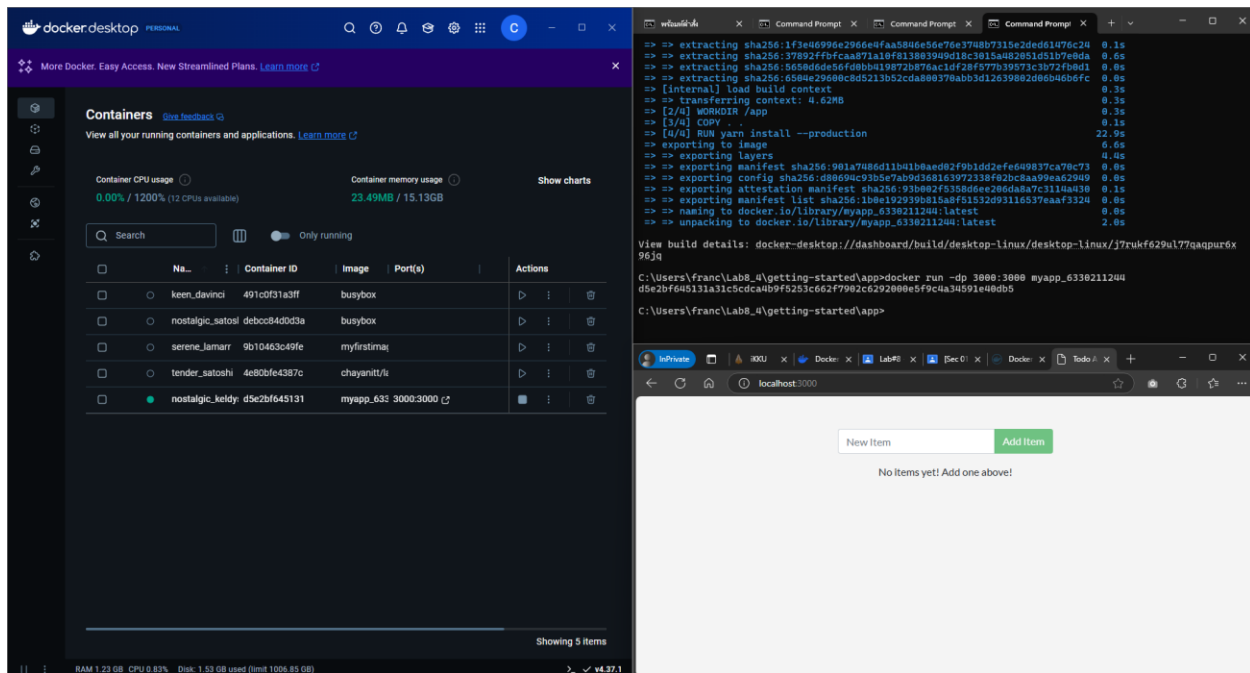
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

\$ docker run -dp 3000:3000 <myapp_รหัสนศ. ไม่มีขีด>

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

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

Lab Worksheet



[Check point#9]

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้
 - a. เปิดไฟล์ src/static/js/app.js ด้วย Editor และแก้ไขบรรทัดที่ 56 จาก

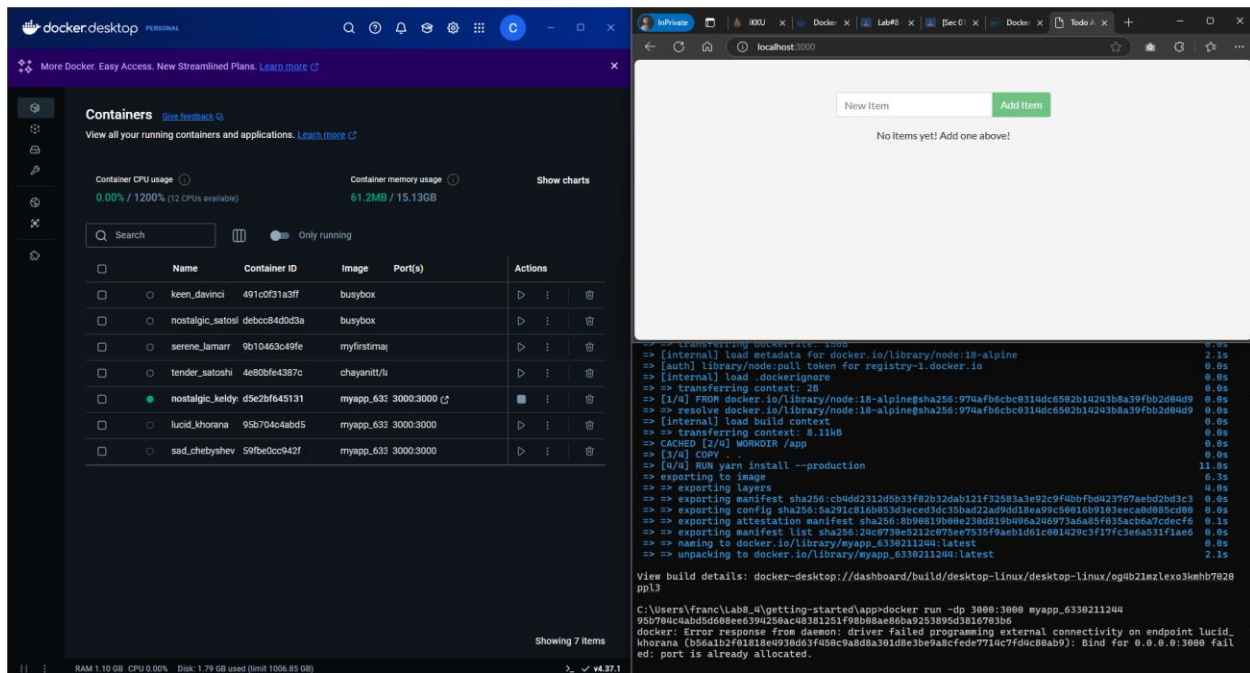

```
<p className="text-center">No items yet! Add one above!</p>
```

 เป็น


```
<p className="text-center">There is no TODO item. Please add one to the list. By
          ชื่อและนามสกุลของนักศึกษา</p>
```
 - b. Save ไฟล์ให้เรียบร้อย
9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5
10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

Lab Worksheet



[Check point#10]

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

ตอบ Error ที่เกิดขึ้น

C:\Users\franc\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6330211244
95b704c4abd5d608ee6394250ac48381251f98b08ae86ba9253895d3816703b6

docker: Error response from daemon: driver failed programming external connectivity on endpoint
lucid_khorana (b56a1b2f01818e4930d63f450c9a8d8a301d8e3be9a8cfede7714c7fd4c80ab9):

Bind for 0.0.0.0:3000 failed: port is already allocated.

หมายถึง พอร์ต 3000 ที่พยายามใช้กับ Container ใหม่ ถูกใช้งานอยู่โดย Container อื่น หรือ โดยโปรเซสอื่นใน
ระบบ โดย Docker ไม่สามารถกำหนดพอร์ตเดียวกันให้หลาย Container/โปรเซสได้ เนื่องจากการชนกันของ
การเชื่อมต่อ (port conflict)

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

i. ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ

Lab Worksheet

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

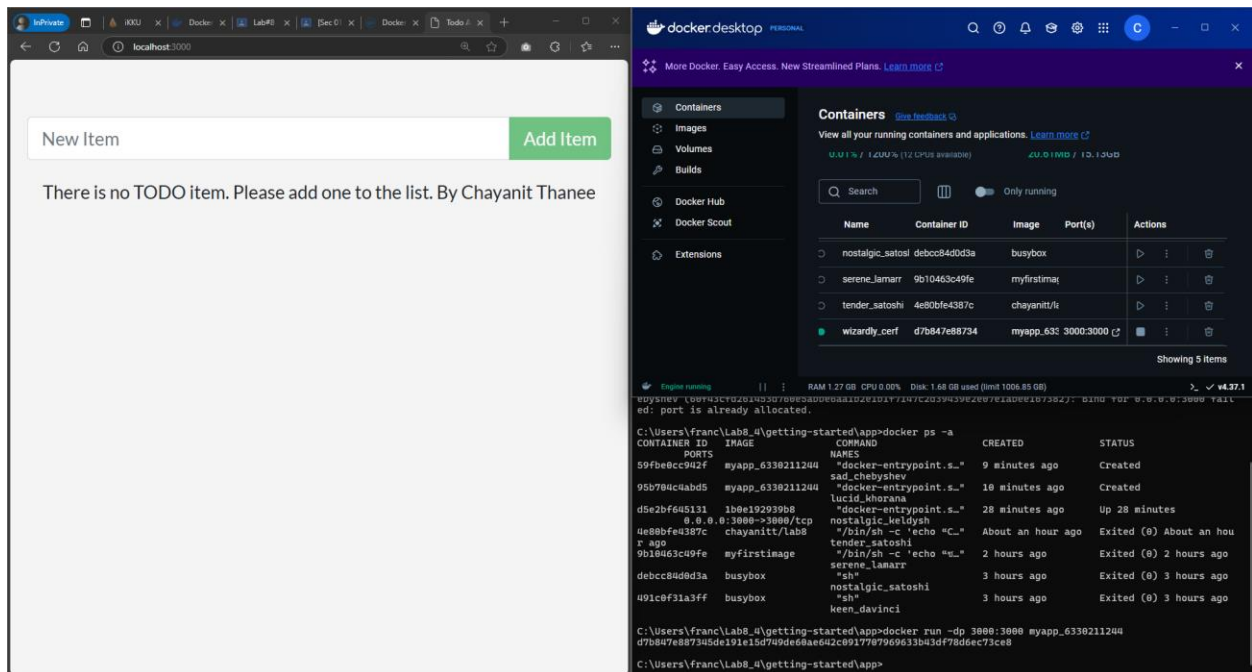
b. ผ่าน Docker desktop

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

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

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



*ลบ Container ของ Web application ผ่าน Docker desktop

[Check point#11]

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
```

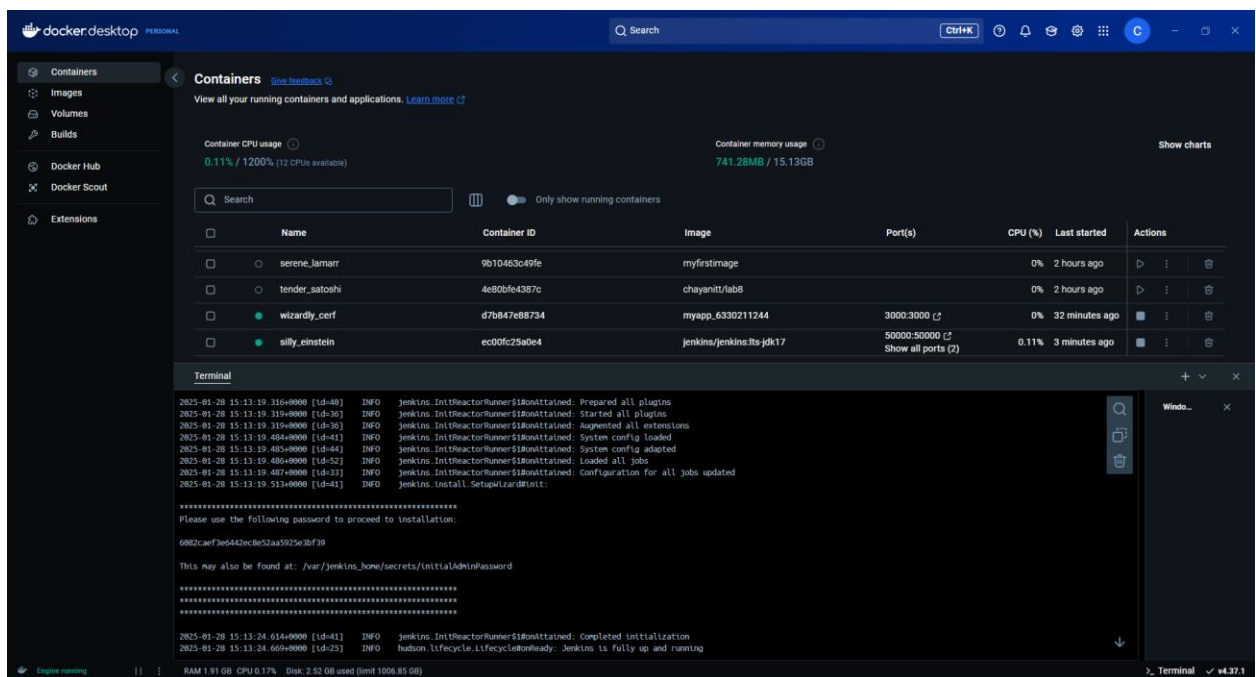
หรือ

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v
```

```
jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17
```

3. บันทึกการรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

[Check point#12] Capture หน้าจอที่แสดงผล Admin password



6082caef3e6442ec8e52aa5925e3bf39

[Check point#12]

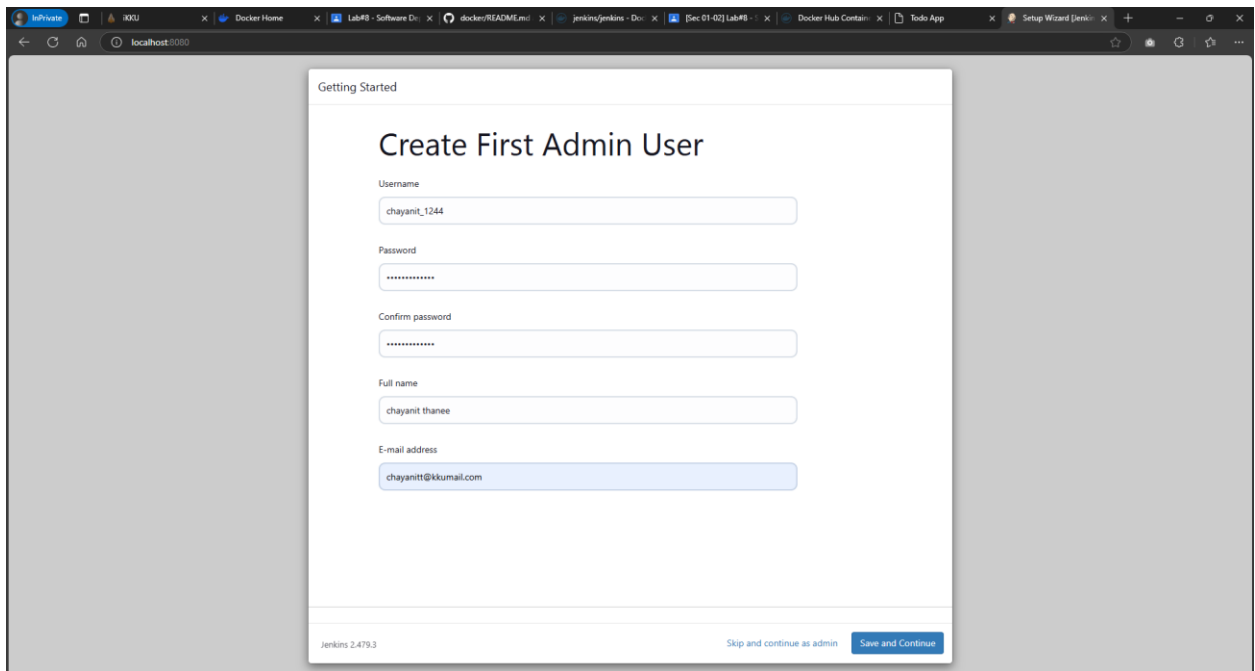
4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น

localhost:8080

Lab Worksheet

5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

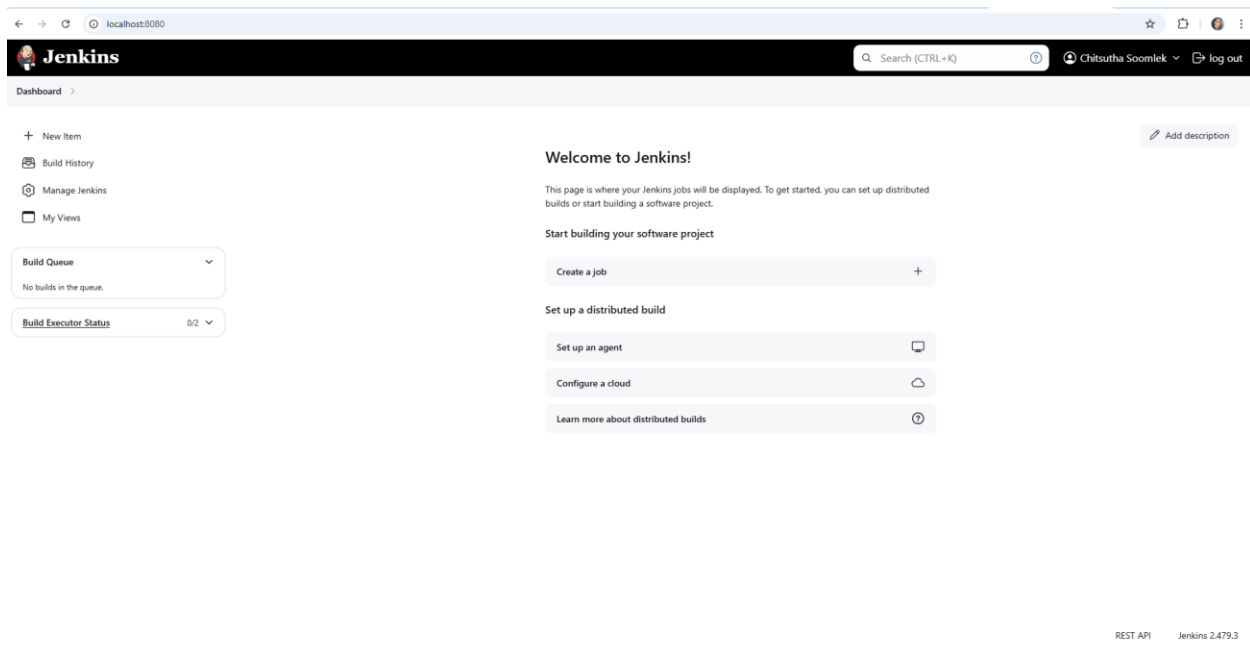
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



[Check point#13]

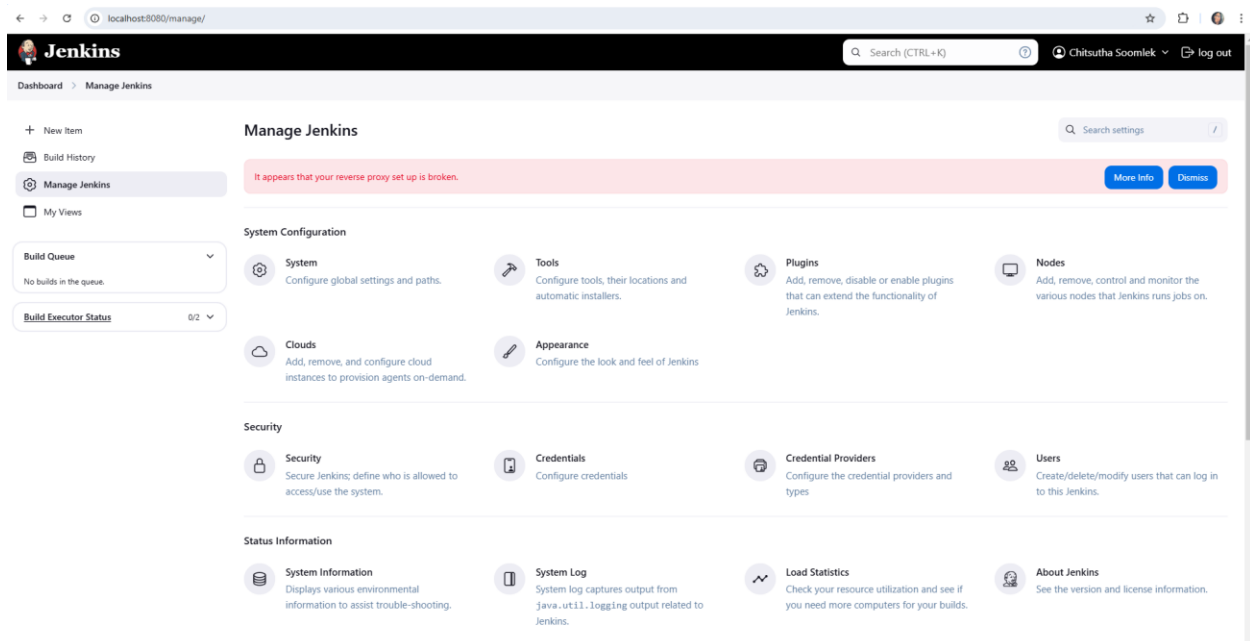
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบหน้าจอ Dashboard ดังแสดงในภาพ

Lab Worksheet



The screenshot shows the Jenkins Dashboard at localhost:8080. The top navigation bar includes the Jenkins logo, a search bar, and user information (Chitsutha Soomlek) with a log out button. The left sidebar contains links for New Item, Build History, Manage Jenkins, and My Views. The main content area features a 'Welcome to Jenkins!' message and instructions on how to start building a software project. It includes buttons for 'Create a job', 'Set up a distributed build', 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'. The bottom right corner displays 'REST API' and 'Jenkins 2.479.3'.

9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



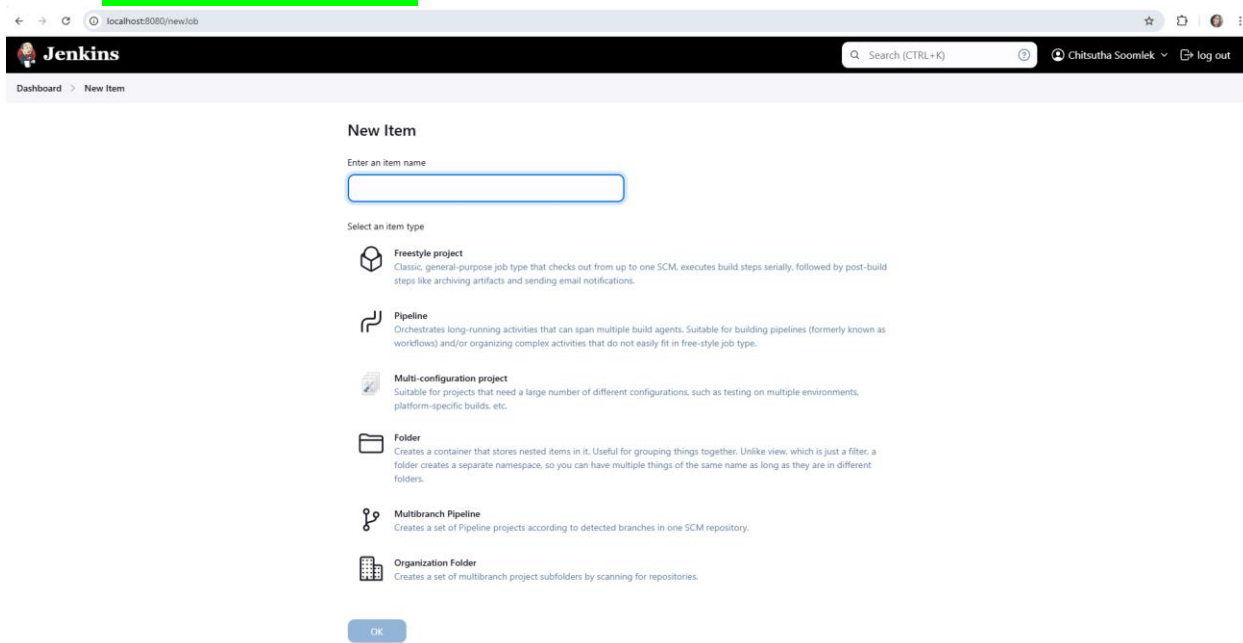
The screenshot shows the Jenkins 'Manage Jenkins' page at localhost:8080/manage/. The top navigation bar is similar to the dashboard. The left sidebar highlights 'Manage Jenkins'. The main content area displays a warning message: 'It appears that your reverse proxy set up is broken.' Below this, the 'System Configuration' section is visible, containing links for System, Tools, Plugins, Nodes, Clouds, and Appearance. The 'Security' section includes links for Security, Credentials, Credential Providers, and Users. The 'Status Information' section includes links for System Information, System Log, Load Statistics, and About Jenkins.

Lab Worksheet

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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



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

Description: Lab 8.5

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

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

Lab Worksheet

Build Steps: เลือก Execute shell แล้วใส่คำสั่งในการรันไฟล์ .robot (หากไฟล์ไม่ได้อยู่ในหน้าแรกของ repository ให้ใส่ Path ไปถึงไฟล์ให้เรียบร้อยแล้ว)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

Lab Worksheet

The screenshot shows the Jenkins configuration page for a project named 'Lab 8.5'. The page is divided into several sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' section is currently selected and shows the following configuration:

- Description:** Lab 8.5
- Plain text:** [Preview](#)
- ☐ Discard old builds [?](#)
- ☒ GitHub project
 - Project url:** [?](#) `https://github.com/Chayanit124-4/Lab7-TestAutomation`
 - Advanced:** [?](#)
- ☐ This project is parameterized [?](#)
- ☐ Throttle builds [?](#)
- ☐ Execute concurrent builds if necessary [?](#)
- Advanced:** [?](#)

Source Code Management

- ☒ None
- ☐ Git [?](#)

Build Triggers

- ☐ Trigger builds remotely (e.g., from scripts) [?](#)
- ☐ Build after other projects are built [?](#)
- ☒ Build periodically [?](#)
 - Schedule:** [?](#) `H/15 * * * *`
 - Would last have run at Tuesday, January 28, 2025 at 4:20:29 PM Coordinated Universal Time; would next run at Tuesday, January 28, 2025 at 4:35:29 PM Coordinated Universal Time.
- ☐ GitHub hook trigger for GITScm polling [?](#)
- ☐ Poll SCM [?](#)

Build Environment

- ☐ Delete workspace before build starts
- ☐ Use secret text(s) or file(s) [?](#)
- ☐ Add timestamps to the Console Output
- ☐ Inspect build log for published build scans
- ☐ Terminate a build if it's stuck
- ☐ With Ant [?](#)

Build Steps

- Execute shell** [?](#)
 - Command:** [See the list of available environment variables](#)
 - `robot lab7_1.robot`
 - Advanced:** [?](#)
- Add build step:** [?](#)

Post-build Actions

- Add post-build action:** [?](#)

Buttons: [Save](#) [Apply](#)

REST API Jenkins 2.479.3

Lab Worksheet

(1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ

ตอบ robot lab7_1.robot

[Check point#14]

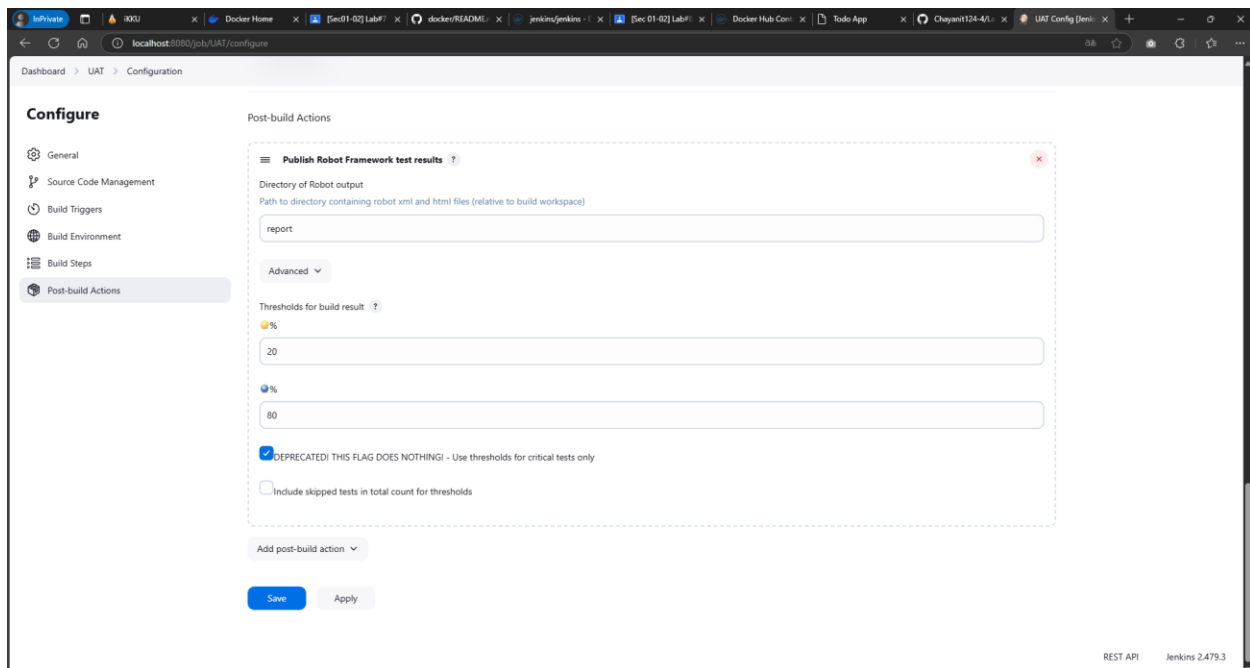
Post-build action: เพิ่ม Publish Robot Framework test results ->

ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> 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



Lab Worksheet

Jenkins Dashboard

S	W	Name	Last Success	Last Failure	Last Duration	Robot Results + Duration Trend
🔴	🔴	UAT	N/A	4 min 36 sec #1	44 ms	▶

Build Queue
No builds in the queue.

Build Executor Status
(0 of 2 executors busy)

Jenkins Console Output

```

Started by user chayanit thanee
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/sh -xe /tmp/jenkins1512256177221381933.sh
+ robot lab7.3.robot
/tmp/jenkins1512256177221381933.sh: 2: robot: not found
Build step 'Execute shell' marked build as failure
Robot results publisher started...
INFO: Checking test criticality is deprecated and will be dropped in a future release!
-Parsing output url:
ERROR: Build step failed with exception
/var/jenkins_home/workspace/UAT/report does not exist.
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:512)
at org.apache.tools.ant.types.AbstractFileSet.getDirectoryScanner(AbstractFileSet.java:489)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:76)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:52)
at hudson.FilePath.act(FilePath.java:1234)
at hudson.FilePath.act(FilePath.java:1217)
at PluginClassLoader for robot/hudson.plugins.robot.RobotParser.parse(RobotParser.java:48)
at PluginClassLoader for robot/hudson.plugins.robot.RobotPublisher.parse(RobotPublisher.java:262)
at PluginClassLoader for robot/hudson.plugins.robot.RobotPublisher.perform(RobotPublisher.java:286)
at hudson.tasks.BuildStepCompatibilityLayer.perform(BuildStepCompatibilityLayer.java:88)
at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:20)
at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
at hudson.model.AbstractBuild$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:767)
at hudson.model.Build$BuildExecution.post2(Build.java:179)
at hudson.model.AbstractBuild$AbstractBuildExecution.post(AbstractBuild.java:711)
at hudson.model.Run.execute(Run.java:1854)
at hudson.model.FreeStyleBuild.run(FreeStyleBuild.java:44)
at hudson.model.ResourceController.execute(ResourceController.java:103)
at hudson.model.Executor.run(Executor.java:449)
Build step 'Publish Robot Framework test results' marked build as failure
Finished: FAILURE
  
```

** ส่งงานให้อาไฟล์pdf หรือ word ใส่ใน github แล้วเอาลิงค์ github ส่ง**