

## Lab Worksheet

ชื่อ-นามสกุล วรรณะ เสี่ยมแหลม รหัสนักศึกษา 653380280-0 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
2. เปิด Command line หรือ Terminal บน Docker Desktop จากนั้นสร้าง Directory ชื่อ Lab8\_1
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

## Lab Worksheet

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

```

Terminal

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox

Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

What's next:
  View a summary of image vulnerabilities and recommendations →docker scout quickview busybox

```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 4

```

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
jodloei-app         latest          0f8347a9fa8d   13 days ago    1.42GB
busybox              latest          af4709625109   3 months ago   4.27MB

```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 5

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

ตอบ ชื่อของ image ที่ถูกใช้เพื่อสร้างคอนเทนเนอร์ โดยชื่อในคอลัมน์นี้จะช่วยให้เรารู้ว่า image นั้นมาจากที่ไหน เช่น nginx หรือ ubuntu ซึ่งเป็นชื่อของ image ที่ถูกดึงมาจาก Docker Hub หรือ registry อื่น ๆ

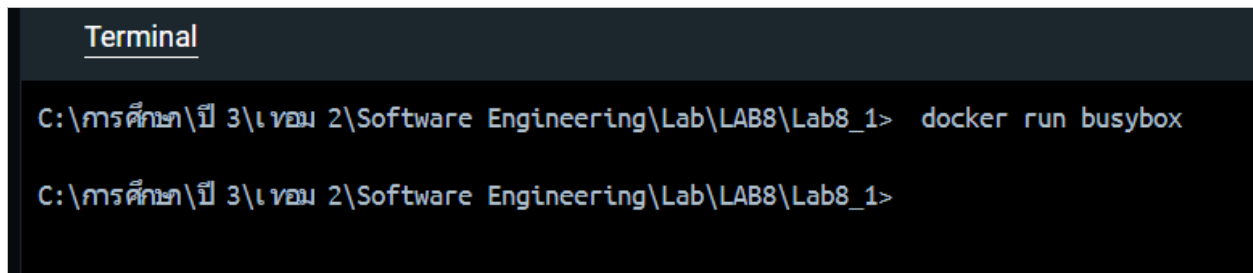
(2) Tag ที่ใช้บ่งบอกถึงอะไร

ตอบ ใช้สำหรับบ่งบอกเวอร์ชันหรือรุ่นของ image นั้น ๆ เช่น ถ้ามี tag เป็น latest จะหมายถึง image เวอร์ชันล่าสุด หรือถ้ามี tag เป็นตัวเลข เช่น 1.20 ก็จะมีหมายถึง image ที่ใช้เวอร์ชัน 1.20

## Lab Worksheet

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



```
Terminal
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1> docker run busybox
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>
```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 6

## Lab Worksheet

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search  ☒ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	keen_chandrasekhar	1718476198e2	busybox		0%	1 minute ago

**Terminal**

```
docker.io/library/busybox:latest

What's next:
  View a summary of image vulnerabilities and recommendations →docker scout quickview busybox

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
jodloei-app   latest    0f8347a9fa8d   13 days ago    1.42GB
busybox       latest    af4709625109   3 months ago   4.27MB

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker run busybox

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker run -it busybox sh
/ #
```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 7

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search  ☒ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	keen_chandrasekhar	1718476198e2	busybox		0%	2 minutes ago

**Terminal**

```
What's next:
  View a summary of image vulnerabilities and recommendations →docker scout quickview busybox

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
jodloei-app   latest    0f8347a9fa8d   13 days ago    1.42GB
busybox       latest    af4709625109   3 months ago   4.27MB

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker run busybox


C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker run -it busybox sh
/ # ls
bin  dev  etc  home  lib  lib64  proc  root  sys  tmp  usr  var
/ #
```


## Lab Worksheet

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 8

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search   ☒ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	 keen_chandrasekhar	1718476198e2	busybox		0%	2 minutes ago

**Terminal**


```
ln dev etc home lib lib64 proc root sys tmp usr var
# ls -la
total 48
-rwxr-xr-x 1 root root 4096 Jan 22 04:23 .
-rwxr-xr-x 1 root root 4096 Jan 22 04:23 ..
-rwxr-xr-x 1 root root 0 Jan 22 04:23 .dockerenv
-rwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
-rwxr-xr-x 5 root root 360 Jan 22 04:23 dev
-rwxr-xr-x 1 root root 4096 Jan 22 04:23 etc
-rwxr-xr-x 2 nobody nobody 4096 Sep 26 21:31 home
-rwxr-xr-x 2 root root 4096 Sep 26 21:31 lib
-rwxrwxrwx 1 root root 3 Sep 26 21:31 lib64 -> lib
-r-xr-xr-x 238 root root 0 Jan 22 04:23 proc
-rwx----- 1 root root 4096 Jan 22 04:25 root
```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 9

## Lab Worksheet

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

Search  ☒ Only show running containers

<input type="checkbox"/>	Name	Container ID	Image ↑	Port(s)	CPU (%)	Last started	Act
No running containers found							

**Terminal**

```

drwxr-xr-x 5 root root 360 Jan 22 04:23 dev
drwxr-xr-x 1 root root 4096 Jan 22 04:23 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 238 root root 0 Jan 22 04:23 proc
drwx----- 1 root root 4096 Jan 22 04:25 root
dr-xr-xr-x 11 root root 0 Jan 22 04:23 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:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8\_1>

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 10

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Act
<input type="checkbox"/>	quizzical_yalow	ad16c20a6515	jodloei-app	4200:4200	N/A	13 days ago	▶
<input type="checkbox"/>	serene_curie	d3093e6c2562	busybox		N/A	5 minutes ago	▶
<input type="checkbox"/>	keen_chandrasekhar	1718476198e2	busybox		N/A	4 minutes ago	▶
<input type="checkbox"/>	keen_bassi	abce47ea221d	busybox		N/A	10 seconds ago	▶

**Terminal**

```

rwxr-xr-x 2 root root 4096 Sep 26 21:31 lib
rwxrwxrwx 1 root root 3 Sep 26 21:31 lib64 -> lib
r-xr-xr-x 238 root root 0 Jan 22 04:23 proc
rwx----- 1 root root 4096 Jan 22 04:25 root
r-xr-xr-x 11 root root 0 Jan 22 04:23 sys
rwxrwxrwt 2 root root 4096 Sep 26 21:31 tmp
rwxr-xr-x 4 root root 4096 Sep 26 21:31 usr
rwxr-xr-x 4 root root 4096 Sep 26 21:31 var
# exit

```

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8\_1>docker run busybox echo "Hello Matthana Saimlaem from busybox"

Hello Matthana Saimlaem from busybox"

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8\_1>

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 11

## Lab Worksheet

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

	name	Container ID	image	Ports	CPU (%)	Last started	Actions
	quizzical_yalow	ad16c20a6515	jodloei-app	4200:4200	N/A	13 days ago	
	serene_curie	d3093e6c2562	busybox		N/A	5 minutes ago	
	keen_chandrasekhar	1718476198e2	busybox		N/A	4 minutes ago	
	keen_bassi	abce47ea221d	busybox		N/A	10 seconds ago	

**Terminal**

```
"Hello Watthana Saimlaem from busybox"
```

```
C:\การศีกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
abce47ea221d	busybox	"echo "Hello Watthan..."	46 seconds ago	Exited (0) 46 seconds ago		keen_bassi
1718476198e2	busybox	"sh"	4 minutes ago	Exited (0) About a minute ago		keen_chandrasekhar
d3093e6c2562	busybox	"sh"	5 minutes ago	Exited (0) 5 minutes ago		serene_curie
0fcfb381d3b7	jodloei-app	"docker-entrpoint.s..."	13 days ago	Exited (137) 13 days ago		goofy_dubinsky
af2120a71ab6	jodloei-app	"docker-entrpoint.s..."	13 days ago	Exited (137) 13 days ago		suspicious_margulis
b5eaabad37c1	jodloei-app	"docker-entrpoint.s..."	13 days ago	Exited (137) 13 days ago		compassionate_rubin
bd595e6f27db	jodloei-app	"docker-entrpoint.s..."	13 days ago	Created		modest_banach
ad16c20a6515	jodloei-app	"docker-entrpoint.s..."	13 days ago	Exited (137) 13 days ago		quizzical_yalow

```
C:\การศีกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>
```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขึ้นตอนที่ 12

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

**ตอบ** เมื่อใช้ option -it ในคำสั่ง docker run จะเป็นการทำให้เราสามารถโต้ตอบกับคอนเทนเนอร์ได้โดยตรงผ่าน terminal โดย -i จะเปิดการรับข้อมูลจากผู้ใช้ และ -t จะเปิด terminal แบบจำลอง เพื่อให้การแสดงผลเหมือนใช้งาน terminal จริง ตัวอย่างเช่น ถ้าเราใช้คำสั่ง docker run -it ubuntu /bin/bash จะทำให้เราเข้าไปยัง shell ของคอนเทนเนอร์ Ubuntu และสามารถพิมพ์คำสั่งเพื่อโต้ตอบได้ทันที

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

**ตอบ** ใช้แสดงสถานะของคอนเทนเนอร์ในปัจจุบัน เช่น คอนเทนเนอร์กำลังทำงานอยู่จะแสดงคำว่า Up พร้อมระยะเวลาที่ทำงาน เช่น Up 5 minutes หรือถ้าคอนเทนเนอร์หยุดทำงานแล้วจะแสดงคำว่า Exited พร้อม Exit Code และระยะเวลาที่หยุด เช่น Exited (0) 10 seconds ago และในกรณีที่คอนเทนเนอร์ถูกหยุดชั่วคราวจะแสดงคำว่า Paused

## Lab Worksheet

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

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

The screenshot shows the Docker Desktop interface. At the top, there's a 'Containers' section with a link to 'Give feedback'. Below it, a message says 'View all your running containers and applications. Learn more'. A table lists four containers:

	name	container id	image	Ports	CPU (%)	Last started	Actions
<input type="checkbox"/>	modest_banach	bd595e6f27db	jodloei-app	4200:4200	N/A		
<input type="checkbox"/>	serene_curie	d3093e6c2562	busybox		N/A	17 minutes ago	
<input type="checkbox"/>	keen_chandrasekhar	1718476198e2	busybox		N/A	16 minutes ago	
<input type="checkbox"/>	keen_bassi	abce47ea221d	busybox		N/A	12 minutes ago	

Below the table, it says 'Showing 7 of 7 containers'. At the bottom, there's a 'Terminal' section with a list of containers and their status:

Container ID	Image	Command	Status	Exit Code	Exit Time	Container Name
af2128a71ab6	jodloei-app	"docker-entrypoint.s..."	Exited (137)	137	13 days ago	suspicious_margulis
b5eaabad37c1	jodloei-app	"docker-entrypoint.s..."	Exited (137)	137	13 days ago	compassionate_rubin
bd595e6f27db	jodloei-app	"docker-entrypoint.s..."	Created			modest_banach
ad16c20a6515	jodloei-app	"docker-entrypoint.s..."	Exited (137)	137	13 days ago	quizzical_yellow

The terminal window shows the following commands and output:

```
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker rm <ad16c20a6515>
The syntax of the command is incorrect.
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>docker rm ad16c20a6515
ad16c20a6515
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_1>
```

รูปจากแบบฝึกปฏิบัติที่ 8.1 ขั้นตอนที่ 13



## Lab Worksheet

## 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 และบันทึกคำสั่งต่อไปนี้

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

```
FROM busybox
```

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

```
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"
```

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

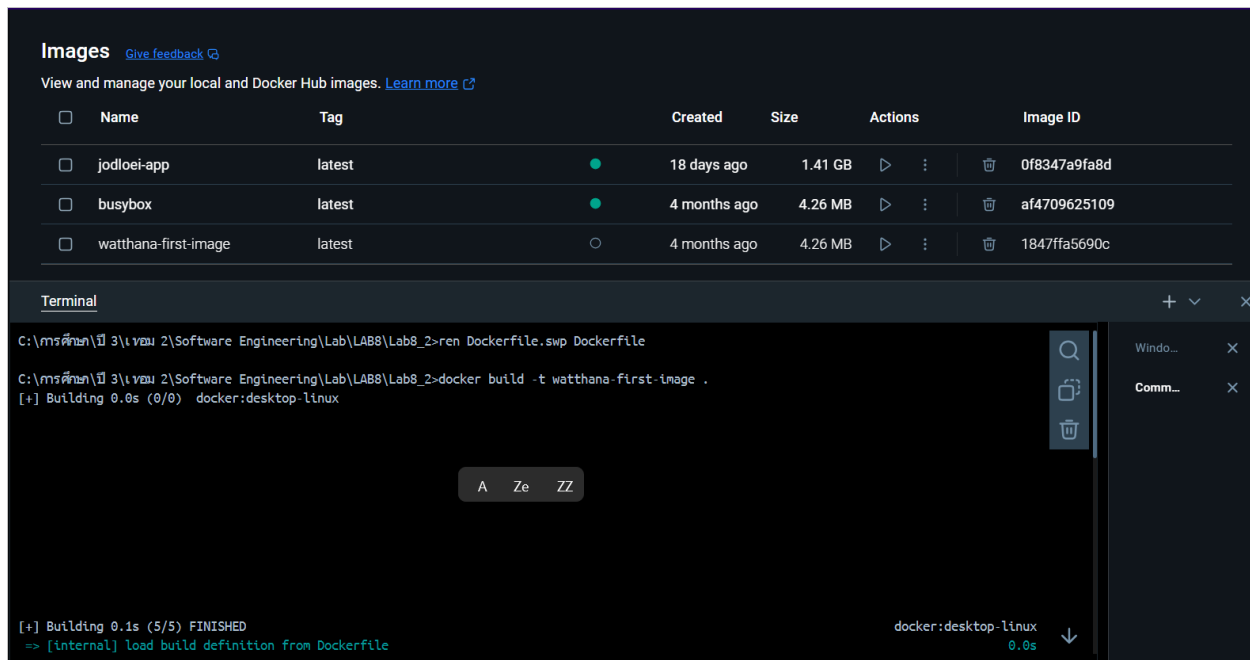
5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

```
$ docker build -t <ชื่อ Image> .
```

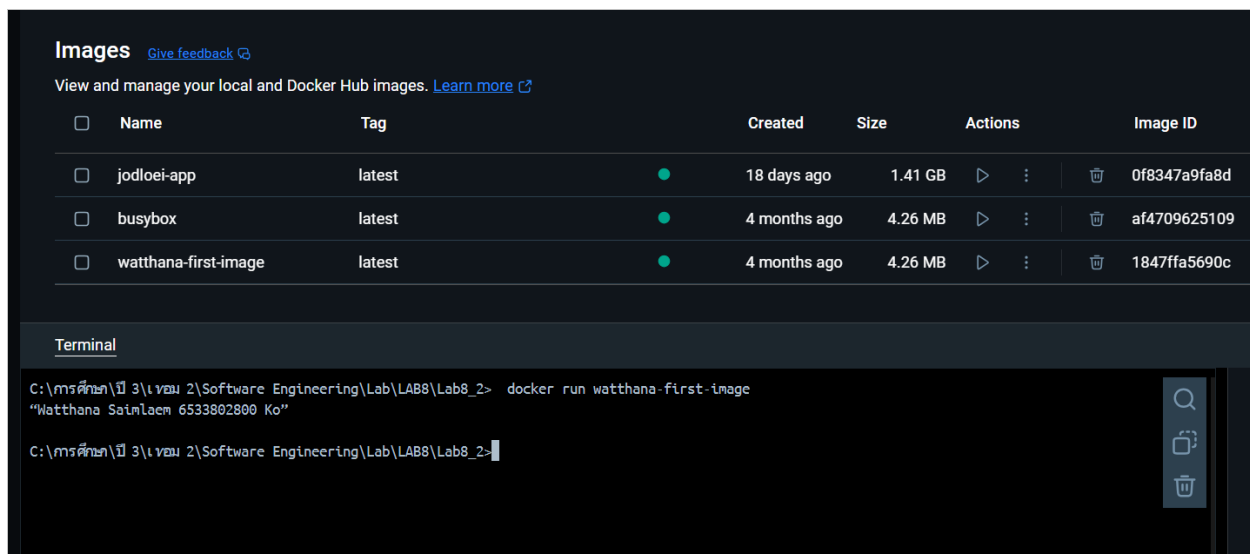
6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

## Lab Worksheet



รูปจากแบบฝึกปฏิบัติที่ 8.2 ขั้นตอนที่ 5



รูปจากแบบฝึกปฏิบัติที่ 8.2 ขั้นตอนที่ 6

## Lab Worksheet

(1) คำสั่งที่ใช้ในการ run คือ

ตอบ docker run watthana-first-image

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

ตอบ ใช้สำหรับการตั้งชื่อและแท็ก ให้กับ Docker image ที่สร้างขึ้น โดยการตั้งชื่อ image ช่วยให้เราสามารถอ้างอิงถึง image นั้นได้ง่าย

### 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 และบันทึกคำสั่งต่อไปนี้

```
$ 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

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

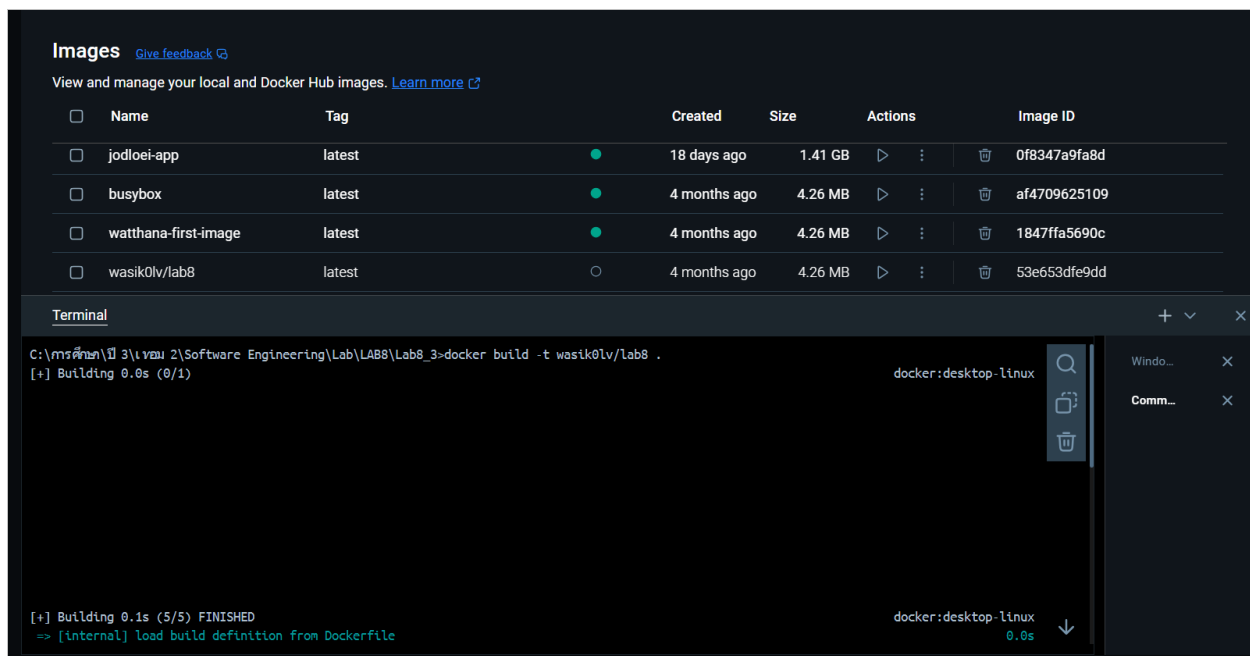
## Lab Worksheet

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

6. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

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

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



รูปจากแบบฝึกปฏิบัติที่ 8.3 ขั้นตอนที่ 5

## Lab Worksheet

**Images** [Give feedback](#)

View and manage your local and Docker Hub images. [Learn more](#)

<input type="checkbox"/>	Name	Tag		Created	Size	Actions	Image ID
<input type="checkbox"/>	jodloei-app	latest	●	18 days ago	1.41 GB	▶ ⋮	0f8347a9fa8d
<input type="checkbox"/>	busybox	latest	●	4 months ago	4.26 MB	▶ ⋮	af4709625109
<input type="checkbox"/>	watthana-first-image	latest	●	4 months ago	4.26 MB	▶ ⋮	1847ffa5690c
<input type="checkbox"/>	wasik0lv/lab8	latest	●	4 months ago	4.26 MB	▶ ⋮	53e653dfe9dd

**Terminal**

```
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab_3>docker run wasik0lv/lab8
"Watthana Sainlaem 6833802800"

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab_3>
```

รูปจากแบบฝึกปฏิบัติที่ 8.3 ขั้นตอนที่ 6

7. 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>
```

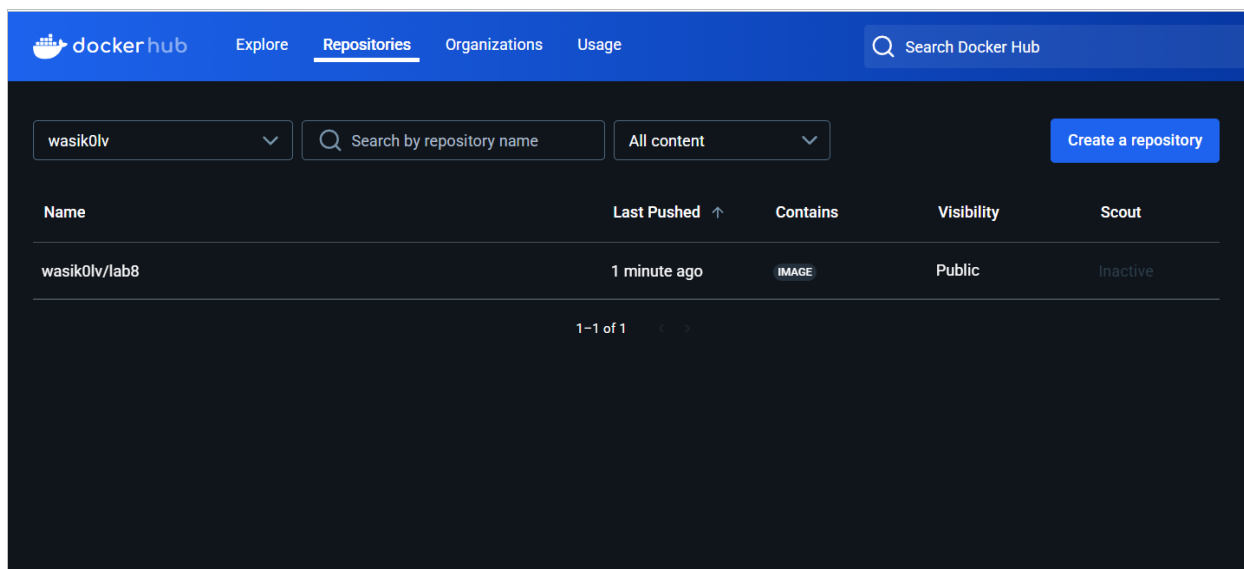
8. ไปที่ Docker Hub กด Tab ชื่อ Tags หรือไปที่ Repository ก็ได้

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

## Lab Worksheet

```
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab_3>docker push wasik0lv/lab8
Using default tag: latest
The push refers to repository [docker.io/wasik0lv/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:13e3754d691b4467b6ae00f4d7c149a4bbd839f7a03f73c72d52a667e433aee1 size: 527
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab_3>
```

รูปจากแบบฝึกปฏิบัติที่ 8.3 ขั้นตอนที่ 7



รูปจากแบบฝึกปฏิบัติที่ 8.3 ขั้นตอนที่ 8

#### 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

## Lab Worksheet

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

```

Terminal
Volume Serial Number is 58DD-2234

Directory of C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app

01/27/2025  10:39 AM  <DIR>          .
01/27/2025  10:39 AM  <DIR>          ..
01/27/2025  10:39 AM                678 package.json
01/27/2025  10:39 AM  <DIR>          spec
01/27/2025  10:39 AM  <DIR>          src
01/27/2025  10:39 AM            150,541 yarn.lock
                2 File(s)        151,219 bytes
                4 Dir(s)    101,072,834,560 bytes free

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app> code package.json

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app>
  
```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 3

```

1 {
2   "name": "101-app",
3   "version": "1.0.0",
4   "main": "index.js",
5   "license": "MIT",
6   "scripts": {
7     "prettify": "prettier -l --write \"**/*.js\"",
8     "test": "jest",
9     "dev": "nodemon src/index.js"
10  },
11  "dependencies": {
12    "express": "^4.18.2",
13    "mysql2": "^2.3.3",
14    "salute": "^5.1.2",
15    "uuid": "^9.0.0",
16    "wait-port": "^1.0.4"
17  },
18  "resolutions": {
19    "ansi-regex": "5.0.1"
20  },
21  "prettier": {
22    "trailingComma": "all",
23    "tabWidth": 4,
24    "useTabs": false,
25    "semi": true,
26    "singleQuote": true
27  },
28  "devDependencies": {
29    "jest": "^29.3.1",
30    "nodemon": "^2.0.20",
31    "prettier": "^2.7.1"
32  }
33 }
  
```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 3

- ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์  
FROM node:18-alpine  
WORKDIR /app  
COPY . .

## Lab Worksheet

RUN yarn install --production

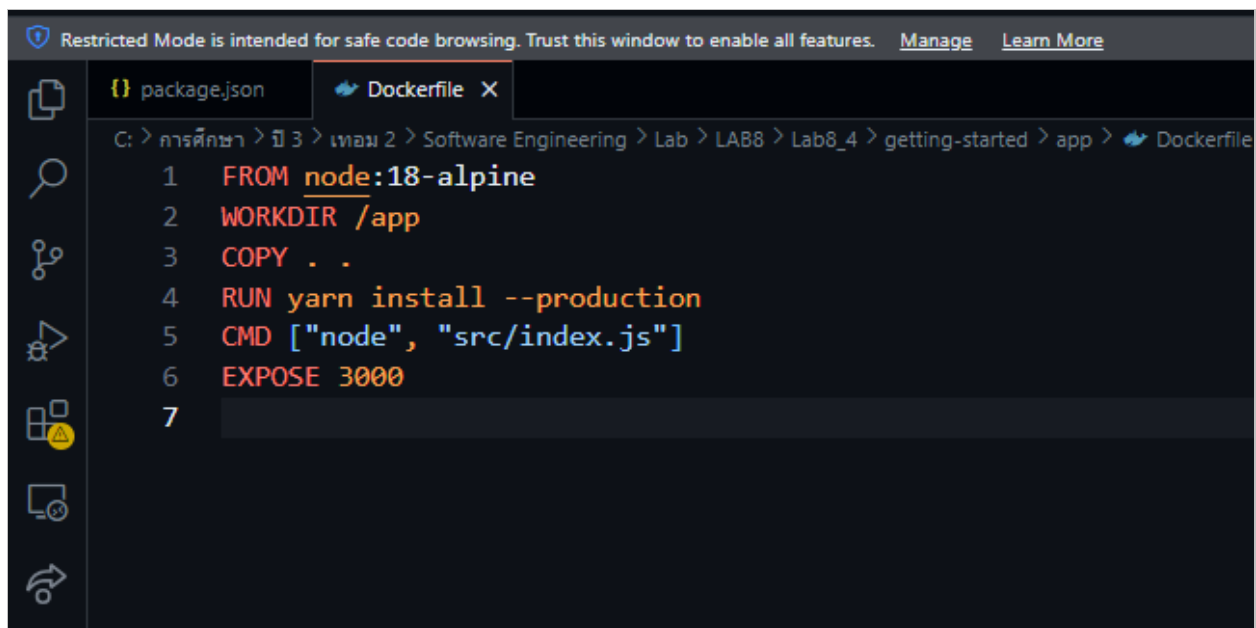
CMD ["node", "src/index.js"]

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดให้ชื่อ image เป็น myapp\_รหัสสนศ. ไม่มีขีด

\$ docker build -t <myapp\_รหัสสนศ. ไม่มีขีด> .

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



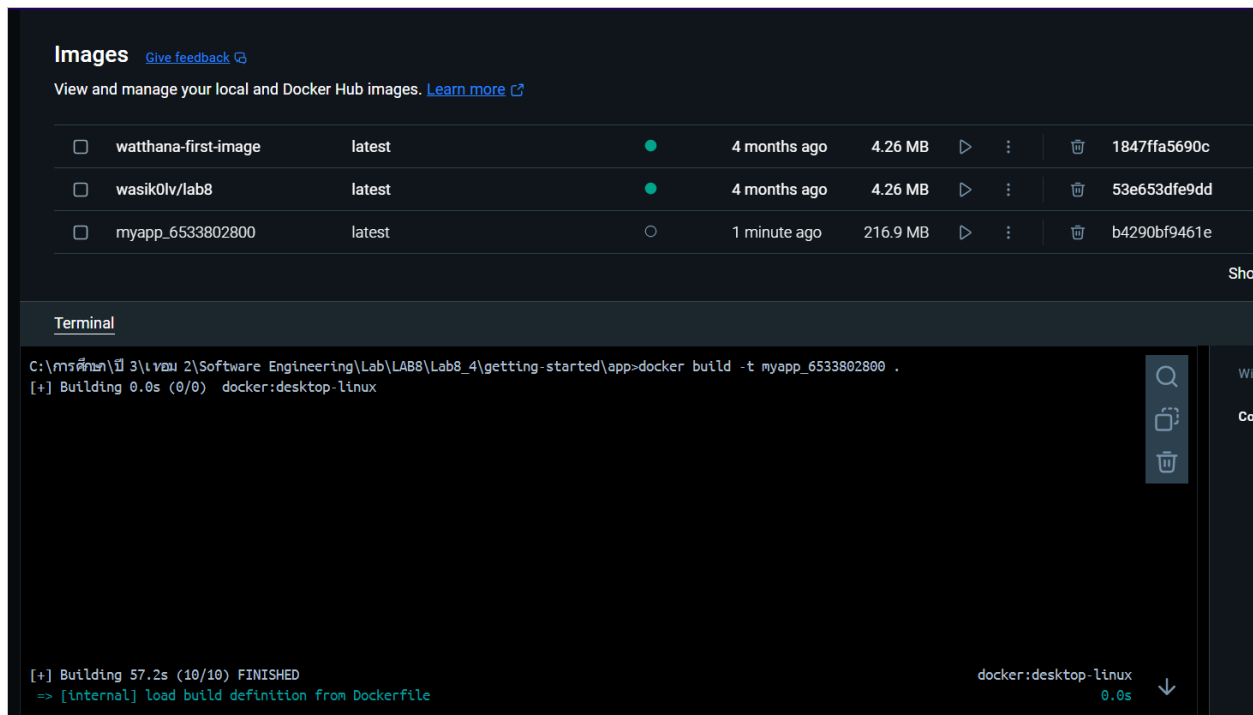
```
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

package.json Dockerfile X
C: > การศึกษา > ปี 3 > เทอม 2 > Software Engineering > Lab > LAB8 > Lab8_4 > getting-started > app > Dockerfile
1 FROM node:18-alpine
2 WORKDIR /app
3 COPY . .
4 RUN yarn install --production
5 CMD ["node", "src/index.js"]
6 EXPOSE 3000
7
```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 4



## Lab Worksheet



รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 5

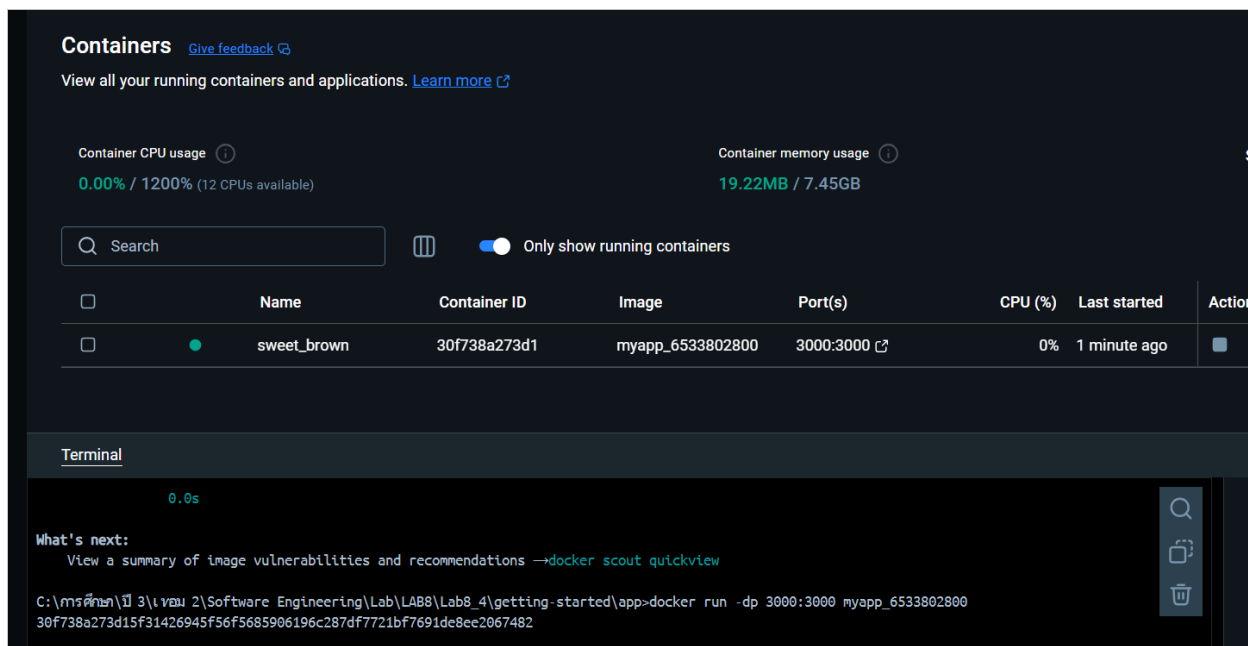
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

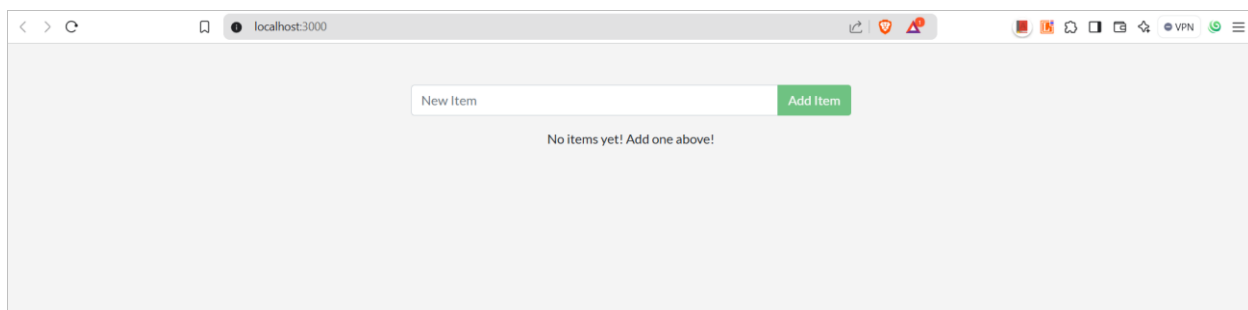
7. เปิด Browser ไปที่ URL = <http://localhost:3000>

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

## Lab Worksheet



รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 6



รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 7

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>

## Lab Worksheet

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

```
return (
  <React.Fragment>
    <AddItemForm onNewItem={onNewItem} />
    {items.length === 0 && (
      <p className="text-center">There is no TODO item. Please add one to the list. By Watthana Saimlaem</p>
    )}
    {items.map(item => (
      <ItemDisplay
        item={item}
        key={item.id}
        onItemUpdate={onItemUpdate}
        onItemRemoval={onItemRemoval}
      />
    ))}
  </React.Fragment>
);
```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 8

```
Terminal
C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app> docker build -t myapp_6533802800 .
[+] Building 0.0s (0/0) docker:desktop-linux

[+] Building 36.5s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 156B                                              0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine                 2.2s
=> [auth] library/node:pull token for registry-1.docker.io                      0.0s
=> [4/4] RUN yarn install --production                                          33.2s
=> exporting to image                                                            0.9s
=> => exporting layers                                                            0.9s
=> => writing image sha256:35424c06afb2a623d82bf9646988ed9eac15c86deacb2a184caf1357d3ba152f 0.0s
```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 9

## Lab Worksheet

```

Terminal
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => writing image sha256:35424c06afb2a623d82bf9646988ed9eac15c86deacb2a184caf1357d3ba152f
=> => naming to docker.io/library/myapp_6533802800
33.2s
0.9s
0.9s
0.0s
0.0s

What's next:
View a summary of image vulnerabilities and recommendations →docker scout quickview

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802800
3d93146730c79d97b144d4606c0a65dc6762e91bb0b0d771f83fe4c2a5585ded
docker: Error response from daemon: driver failed programming external connectivity on endpoint laughing_spence (7d54cd8bd8ae19c5d5c056909d3174d
25ee448875d3a0273176db46c36f91adc): Bind for 0.0.0.0:3000 failed: port is already allocated.

C:\การศึกษา\ปี 3\เทอม 2\Software Engineering\Lab\LAB8\Lab8_4\getting-started\app>

```

รูปจากแบบฝึกปฏิบัติที่ 8.4 ขึ้นตอนที่ 10

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

**ตอบ** Error ที่เกิดขึ้นหมายความว่า พอร์ต 3000 บนเครื่องคอมพิวเตอร์เครื่องนี้กำลังถูกใช้งานโดยโปรแกรมอื่นอยู่แล้ว ดังนั้น Docker ไม่สามารถเชื่อมต่อพอร์ตนั้นกับคอนเทนเนอร์ที่กำลังพยายามรันได้

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

- ใช้คำสั่ง `$ docker ps` เพื่อดู Container ID ที่ต้องการจะลบ
- Copy หรือบันทึก Container ID ไว้
- ใช้คำสั่ง `$ docker stop <Container ID ที่ต้องการจะลบ>` เพื่อหยุดการทำงานของ Container ดังกล่าว
- ใช้คำสั่ง `$ docker rm <Container ID ที่ต้องการจะลบ>` เพื่อทำการลบ

b. ผ่าน Docker desktop

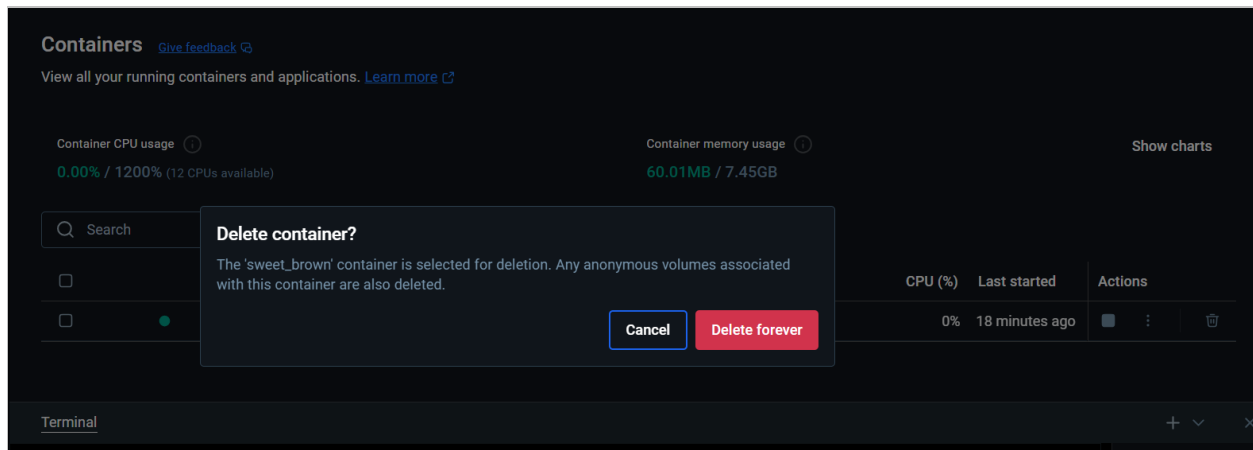
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

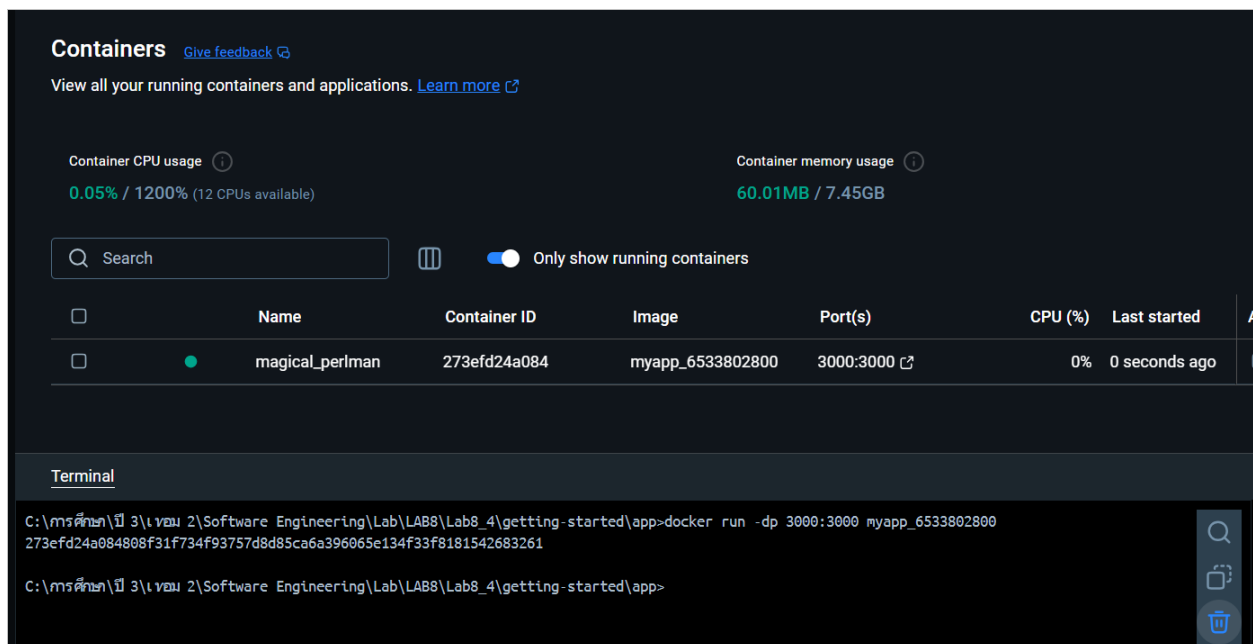
13. เปิด Browser ไปที่ URL = <http://localhost:3000>

## Lab Worksheet

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

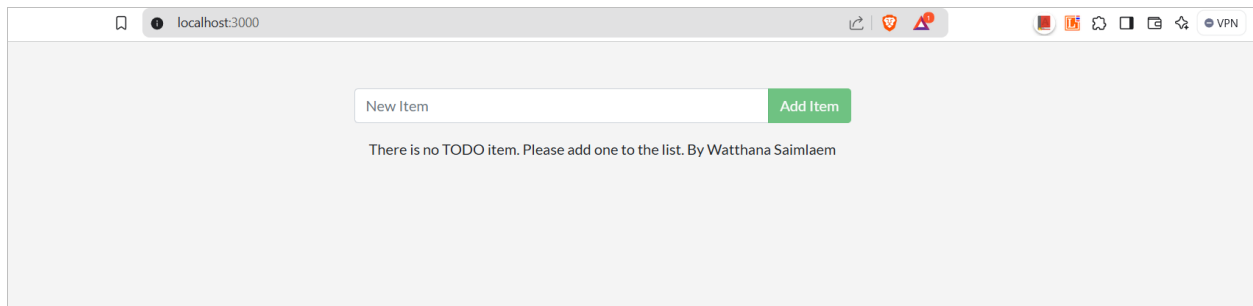


รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 11



รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 12

## Lab Worksheet



รูปจากแบบฝึกปฏิบัติที่ 8.4 ขั้นตอนที่ 13

### 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

## Lab Worksheet

**Containers** [Give feedback](#)

View all your running containers and applications. [Learn more](#)

<input type="checkbox"/>	<span style="color: green;">●</span>	magical_perlman	273efd24a084	myapp_6533802800	3000:3000 ↗	0%	14 minutes ago	<input type="checkbox"/>
<input type="checkbox"/>	<span style="color: green;">●</span>	ecstatic_wright	4a8bf1d8c97c	jenkins/jenkins:its-jdk17	50000:50000 ↗ Show all ports (2)	0.6%	18 seconds ago	<input type="checkbox"/>

---

**Terminal**

```

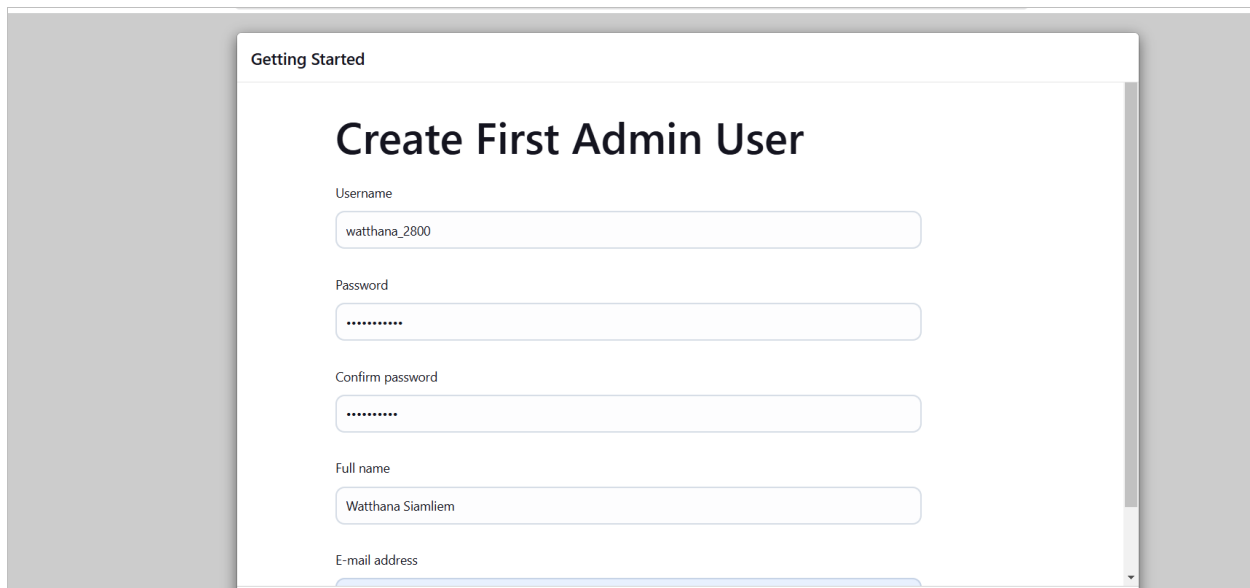
*****
*****
*****
Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:
bc8ea030b5464bee8ec249e75441412f
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword
*****
*****
*****

```

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 3

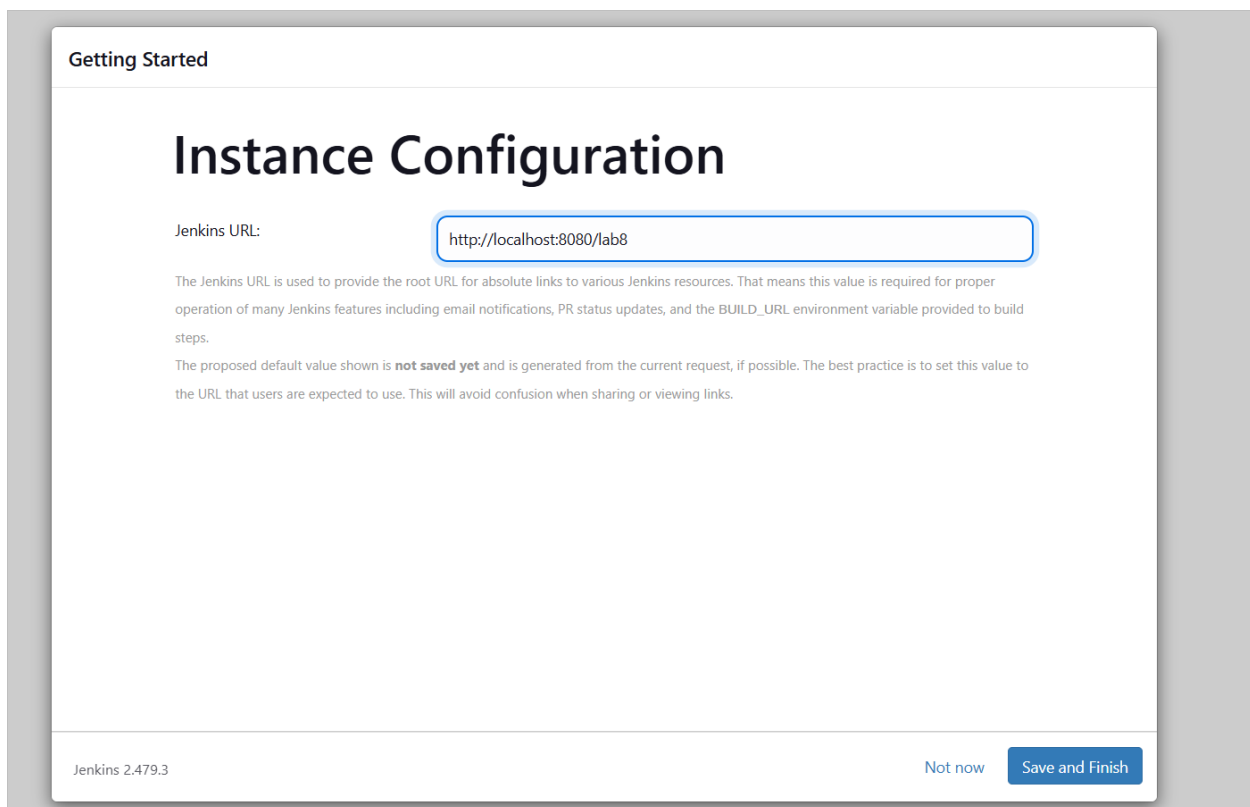
4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
  5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
  6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062
- [Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

## Lab Worksheet



The screenshot shows the 'Getting Started' page of Jenkins with the 'Create First Admin User' form. The form includes fields for Username (wathana\_2800), Password (masked with dots), Confirm password (masked with dots), Full name (Wathana Siamliem), and E-mail address (empty). The form is set against a light gray background with a white card-like container.

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 6



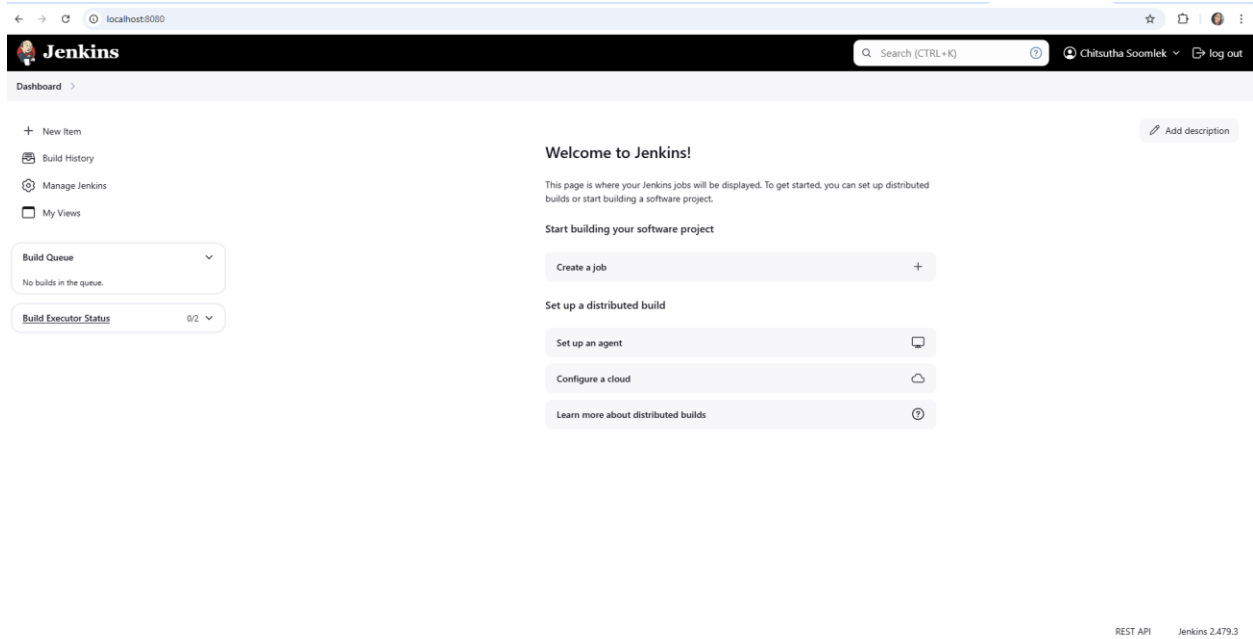
The screenshot shows the 'Getting Started' page of Jenkins with the 'Instance Configuration' section. The 'Jenkins URL' field is filled with 'http://localhost:8080/lab8'. Below the field, there is explanatory text about the Jenkins URL and its use. At the bottom, there are two buttons: 'Not now' and 'Save and Finish'. The Jenkins version '2.479.3' is displayed in the bottom left corner.

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 7

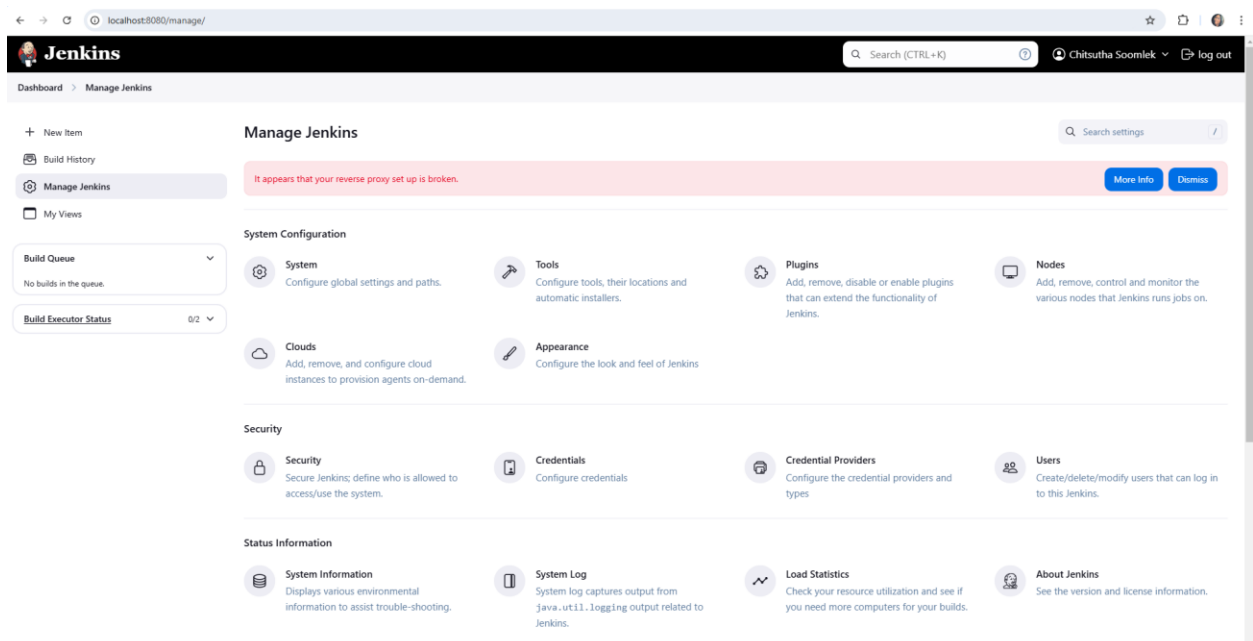


## Lab Worksheet

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



9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

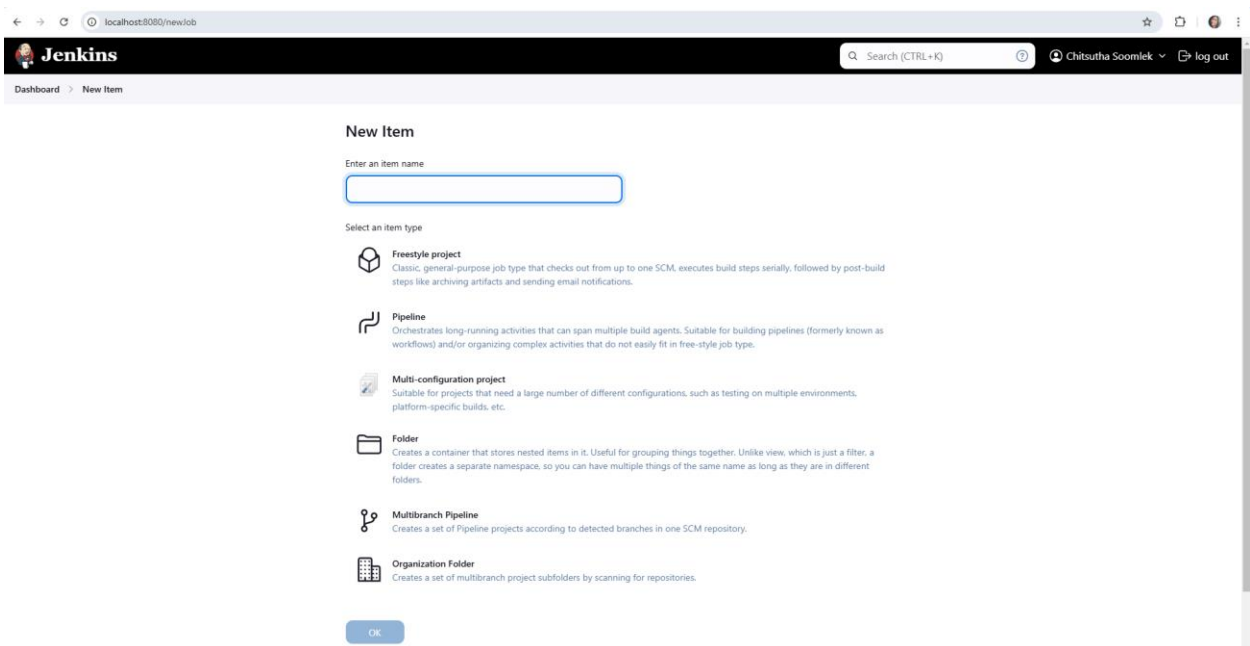


## Lab Worksheet

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

## Lab Worksheet

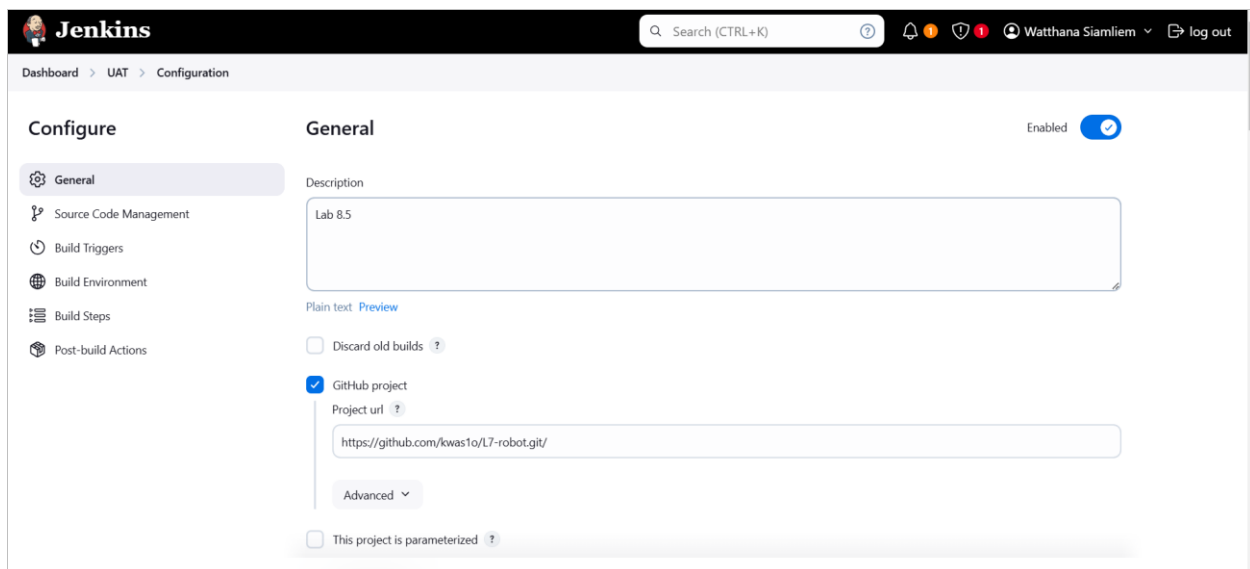
Description: Lab 8.5

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

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้



รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 12

## Lab Worksheet

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Build Steps
- Post-build Actions

☐ Trigger builds remotely (e.g., from scripts) ?  
☐ Build after other projects are built ?  
☒ Build periodically ?  
 Schedule ?  
 H/15 \* \* \* \* \*  
 Would last have run at Monday, January 27, 2025 at 10:20:34 AM Coordinated Universal Time; would next run at Monday, January 27, 2025 at 10:35:34 AM 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

Save Apply

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 12

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps**
- Post-build Actions

**Build Steps**

Execute shell ?

Command

See the list of available environment variables

```
#!/bin/bash
cd $WORKSPACE
if [ ! -d "L7-robot" ]; then
  git clone https://github.com/kwas10/L7-robot.git
fi

cd L7-robot/demoapp

if ! command -v python3 && /dev/null; then
  echo "Python 3 is not installed. Please install Python 3."
  exit 1
fi

if [ ! -d "venv" ]; then
  python3 -m venv venv
fi

source venv/bin/activate
```

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 12

## Lab Worksheet

Dashboard > UAT > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions**

#### Post-build Actions

##### Publish Robot Framework test results ?

Directory of Robot output  
Path to directory containing robot xml and html files (relative to build workspace)

Advanced ▾ Edited

Thresholds for build result ?

🟡%

🟢%

☒ DEPRECATED! THIS FLAG DOES NOTHING! - Use thresholds for critical tests only

☐ Include skipped tests in total count for thresholds

Save Apply

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 12

## Lab Worksheet

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

```
#!/bin/bash
cd $WORKSPACE
if [ ! -d "L7-robot" ]; then
    git clone https://github.com/kwas1o/L7-robot.git
fi

cd L7-robot/demoapp

if ! command -v python3 &> /dev/null; then
    echo "Python 3 is not installed. Please install Python 3."
    exit 1
fi

if [ ! -d "venv" ]; then
    python3 -m venv venv
fi

source venv/bin/activate

if ! command -v pip &> /dev/null; then
    echo "pip is not installed. Installing pip..."
    python3 -m ensurepip --upgrade
fi

if ! pip show robotframework &> /dev/null; then
    echo "Robot Framework is not installed. Installing..."
    pip install robotframework
fi

if ! pip show robotframework-seleniumlibrary &> /dev/null; then
    echo "SeleniumLibrary is not installed. Installing..."
    pip install robotframework-seleniumlibrary
fi

export PATH=$PATH:$WORKSPACE/L7-robot/ChromeForTesting/chromedriver
export CHROME_BIN=$WORKSPACE/L7-robot/ChromeForTesting/chrome.exe

chmod +x /var/jenkins_home/workspace/UAT/L7-robot/ChromeForTesting/chromedriver/chromedriver.exe
chmod +x /var/jenkins_home/workspace/UAT/L7-robot/ChromeForTesting/chrome/chrome.exe

echo "Starting the server..."
nohup python3 server.py &

cd ../input_tests
echo "Running the Robot Framework test..."
robot /var/jenkins_home/workspace/UAT/L7-robot/input_tests/completed_input.robot
```

## Lab Worksheet

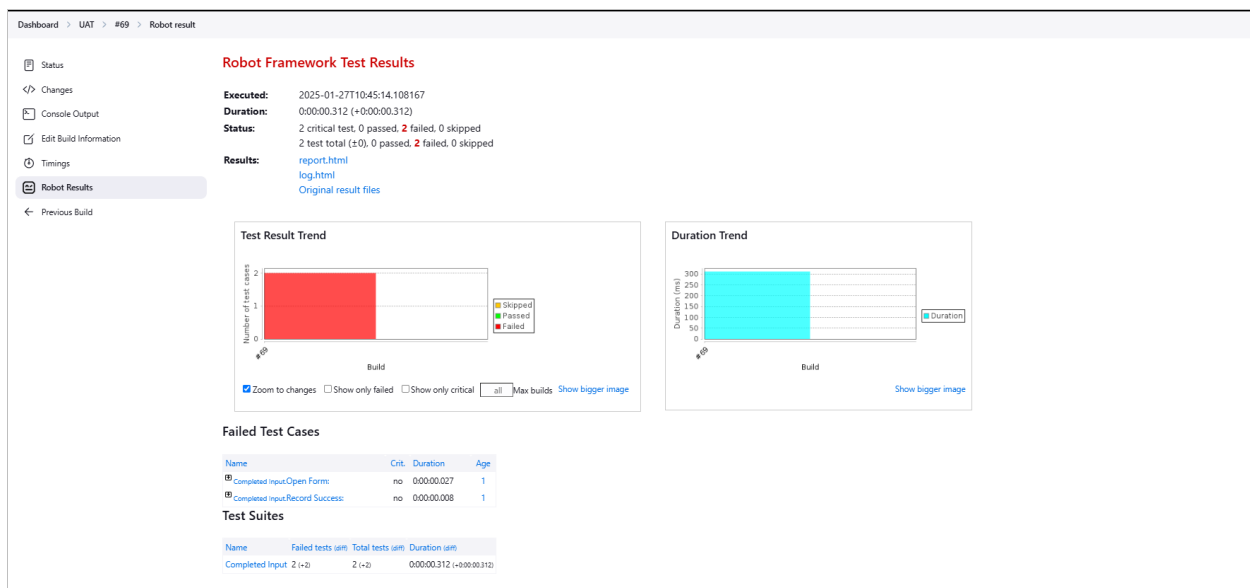
Post-build action: เพิ่ม Publish Robot Framework test results ->

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



รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 14

## Lab Worksheet

The screenshot shows the Jenkins 'Console Output' for a build named 'UAT' (number 969). The output log indicates a failure during the 'Execute shell' step. The error message is: 'WebDriverException: Message: Service /var/jenkins\_home/workspace/UAT/L7-robot/input\_tests/.../ChromeForTesting/chromedriver/chromedriver.exe unexpectedly exited. Status code was: 1'. The log also shows that 2 tests passed and 2 failed. The build is marked as 'FAILURE'.

```

Dashboard > UAT > #69 > Console Output

Status
Changes
Console Output
Edit Build Information
Delete build #69
Timings
Robot Results
Previous Build

Console Output

Started by user Matthaeus SiamSiam
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/bash /tmp/jenkins918877532833181482.sh
Starting the server...
Running the Robot Framework test...
=====
Completed Input
=====
Open Form: | FAIL |
WebDriverException: Message: Service /var/jenkins_home/workspace/UAT/L7-robot/input_tests/.../ChromeForTesting/chromedriver/chromedriver.exe unexpectedly exited. Status code was: 1
=====
Record Success: | FAIL |
No browser is open.
=====
Completed Input
2 tests, 0 passed, 2 failed | FAIL |
=====
Output: /var/jenkins_home/workspace/UAT/L7-robot/input_tests/output.xml
Log: /var/jenkins_home/workspace/UAT/L7-robot/input_tests/log.html
Report: /var/jenkins_home/workspace/UAT/L7-robot/input_tests/report.html
Process leaked file descriptors. See https://www.jenkins.io/redirect/troubleshooting/process-leaked-file-descriptors for more information
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 xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE
  
```

รูปจากแบบฝึกปฏิบัติที่ 8.5 ขั้นตอนที่ 14