

เล็ก

## Lab Worksheet

ชื่อ-นามสกุล\_โจนาทาน ด้วยอง\_รหัสนักศึกษา\_653380322-0\_\_Section\_\_4\_\_

## 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>)

เล็ก

## Lab Worksheet

4. ป้อนคำสั่ง \$ docker images

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

```
PS C:\Users\KKU650001> cd .\Lab8_1\
PS C:\Users\KKU650001\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c13
8bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
PS C:\Users\KKU650001\Lab8_1> docker images
>>
REPOSITORY    TAG        IMAGE ID      CREATED       SIZE
busybox       latest     af4709625109  4 months ago  4.27MB
```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

ตอบ busybox แสดงชื่อของ Docker Image ที่ถูกดาวน์โหลดมาใช้งาน

(2) Tag ที่ใช้บ่งบอกถึงอะไร

ตอบ แสดงชื่อของ Docker Image ที่ถูกดาวน์โหลดมาใช้งาน

5. ป้อนคำสั่ง \$ docker run busybox

6. ป้อนคำสั่ง \$ docker run -it busybox sh

7. ป้อนคำสั่ง ls

8. ป้อนคำสั่ง ls -la

9. ป้อนคำสั่ง exit

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

11. ป้อนคำสั่ง \$ docker ps -a

เล็ก

## Lab Worksheet

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

```
PS C:\Users\KKU650001\Lab8_1> docker run busybox
PS C:\Users\KKU650001\Lab8_1> docker run -it busybox sh
/ # ls
bin      etc      lib      proc     sys      usr
dev      home    lib64    root     tmp      var
/ # ls -la
total 48
drwxr-xr-x  1 root    root           4096 Jan 29 13:15 .
drwxr-xr-x  1 root    root           4096 Jan 29 13:15 ..
-rwxr-xr-x  1 root    root           0 Jan 29 13:15 .docker
renv
drwxr-xr-x  2 root    root          12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root           360 Jan 29 13:15 dev
drwxr-xr-x  1 root    root           4096 Jan 29 13:15 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 237 root    root           0 Jan 29 13:15 proc
drwx----- 1 root    root           4096 Jan 29 13:15 root
dr-xr-xr-x 11 root    root           0 Jan 29 13:15 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
PS C:\Users\KKU650001\Lab8_1> docker run busybox echo "Hello Jo
nathan Doillon from busybox"
Hello Jonathan Doillon from busybox
```

เล็ก

## Lab Worksheet

```

PS C:\Users\KKU650001\Lab8_1> docker ps -a
acaf2ed2f37c    busybox    "echo 'Hello Jonatha..."    16 seconds ago
o              Exited (0) 15 seconds ago    charming_hawk
ing
ac8756e3ca29    busybox    "Hello Jonathan Doil..."    About a minut
e ago    Created    distracted_sw
irles
8c051be05eaf    busybox    "sh"    About a minut
e ago    Exited (0) About a minute ago    reverent_hawk
ing
238611a84d44    busybox    "sh"    About a minut
e ago    Exited (0) About a minute ago    peaceful_hofs
tadter
52e2b8d30b2e    busybox    "echo 'Hello Joonath..."    11 minutes ag
o    Exited (0) 11 minutes ago    serene_edison
ce89ee35c0db    busybox    "sh"    12 minutes ag
o    Exited (0) 12 minutes ago    nervous_jang
4e8367aeb7b4    busybox    "sh"    13 minutes ag
o    Exited (0) 13 minutes ago    awesome_heyro
vsky

```

```

PS C:\Users\KKU650001\Lab8_1> docker ps -a
>> C:\Users\KKU650001\Lab8_1>
CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
acaf2ed2f37c    busybox    "echo 'Hello Jonatha..."    20 minutes ago    Exited (0) 5 minutes ago    charming_hawking
ac8756e3ca29    busybox    "Hello Jonathan Doil..."    21 minutes ago    Created    distracted_swirles
8c051be05eaf    busybox    "sh"    21 minutes ago    Exited (0) 6 minutes ago    reverent_hawking
238611a84d44    busybox    "sh"    21 minutes ago    Exited (0) 6 minutes ago    peaceful_hofstadter
52e2b8d30b2e    busybox    "echo 'Hello Joonath..."    31 minutes ago    Exited (0) 16 minutes ago    serene_edison
ce89ee35c0db    busybox    "sh"    32 minutes ago    Exited (0) 17 minutes ago    nervous_jang
4e8367aeb7b4    busybox    "sh"    33 minutes ago    Exited (0) 17 minutes ago    awesome_heyrovsky
PS C:\Users\KKU650001\Lab8_1> docker rm acaf2ed2f37c

```

- (1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

ตอบ -it เป็นการรวมกันของสอง flag

-i ทำให้ container ทำงานใน **interactive mode** ซึ่งรอรับ input จากผู้ใช้

-t จำลอง terminal ให้สามารถพิมพ์โต้ตอบกับ container ได้

- (2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

ตอบ คอลัมน์ STATUS แสดง สถานะของ container ว่าอยู่ในสถานะใด

เล็ก

## Lab Worksheet

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

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

### 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
```

```
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

เล็ก

## Lab Worksheet

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

```
PS C:\Users\KKU650001> cd .\Lab8_2\
PS C:\Users\KKU650001\Lab8_2> docker build -t my-first-image .
>>
[+] Building 0.8s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 155B                             0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
=> [internal] load .dockerignore                                0.1s
=> => transferring context: 2B                                    0.0s
=> [1/1] FROM docker.io/library/busybox:latest                 0.0s
=> exporting to image                                           0.1s
=> => exporting layers                                           0.0s
=> => writing image sha256:39eb523ba92170e2d533782c458b5e931bf89269a5f20f97c44edeb38753fde4 0.0s
=> => naming to docker.io/library/my-first-image                0.0s

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)
PS C:\Users\KKU650001\Lab8_2>
```

(1) คำสั่งที่ใช้ในการ run คือ

ตอบ docker run my-first-image

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

ตอบ Option -t ในคำสั่ง \$ docker build ใช้กำหนดชื่อและแท็กให้กับ Docker Image ที่สร้างขึ้น ทำให้สามารถอ้างอิงและเรียกใช้ Image ได้ง่ายขึ้น แทนที่จะใช้ Image ID ซึ่งเป็นค่าอัตโนมัติที่อ่านยาก

### 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

เล็ก

## Lab Worksheet

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 &gt; Dockerfile &lt;&lt; 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

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t &lt;username ที่ลงทะเบียนกับ Docker Hub&gt;/lab8

5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

\$ docker run &lt;username ที่ลงทะเบียนกับ Docker Hub&gt;/lab8

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

```
PS C:\Users\KKU650001\Lab8_3> docker build -t joe3220/lab8 .
>>
[+] Building 0.2s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.1s
=> => transferring dockerfile: 178B                             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           0.0s
=> exporting to image                                           0.0s
=> => exporting layers                                           0.0s
=> => writing image sha256:11eccc68f9599d59f26016e1f1f03d560e4b0c4ea13dfbb2811dcededf712b0 0.0s
=> => naming to docker.io/joe3220/lab8                          0.0s

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)
PS C:\Users\KKU650001\Lab8_3> docker run joe3220/lab8
"Jonathan Doillon 653380322-0"
```

เล็ก

## Lab Worksheet

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยใช้คำสั่ง

```
$ 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

```
PS C:\Users\KKU650001\Lab8_3> docker push joe3220/lab8
Using default tag: latest
The push refers to repository [docker.io/joe3220/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:276d351efb601aa6c2918ade9f65abc1fd108a8c6d902c89c73ea82d567740ea size: 527
PS C:\Users\KKU650001\Lab8_3>
```

[Docker Hub](#) / [joe3220/lab8](#)

<

**joe3220/lab8**
Tag  
latest ✓
Pull Run

☆ 0
📄 0

Overview
Tags

Sort by  
Newest
Filter tags

Tag: latest  
Last pushed 4 minutes ago by [joe3220](#)

Digest	OS/ARCH	Size	Last pull
276d351efb60	linux / amd64	2.15 MB	4 minutes ago <a href="#">Pull image</a>

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



เล็ก

## Lab Worksheet

**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

```
PS C:\Users\KKU650001> cd .\Lab8_4\  
PS C:\Users\KKU650001\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 | 8.42 MiB/s, done.  
Resolving deltas: 100% (523/523), done.
```

เล็ก

## Lab Worksheet

```
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     "sqlite3": "^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 }
```

เล็ก

## Lab Worksheet

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

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

```

PS C:\Users\KKU650001\Lab8_4> docker build -t myapp_6533803220 .
[+] Building 0.2s (1/1) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 2B
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
PS C:\Users\KKU650001\Lab8_4> ^C
PS C:\Users\KKU650001\Lab8_4> cd .\getting-started\app
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker build -t myapp_6533803220 .
[+] Building 29.6s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB
=> => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aece8647b17a7bcebb385e 1.72kB / 1.72kB
=> => sha256:dcfb7b337595be6f4d214e4eed84f230ee0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB
=> => sha256:37892ffbfcaaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 1.3s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B
=> => extracting sha256:37892ffbfcaaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 2.4s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 0.1s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 0.0s
=> [internal] load build context
=> => transferring context: 4.62MB 1.0s
=> [2/4] WORKDIR /app 0.9s
=> [3/4] COPY . . 0.4s
=> [4/4] RUN yarn install --production 0.1s
=> exporting to image 16.6s
=> => exporting layers 1.3s
=> => writing image sha256:7cdfefc9c837756960df51b7ff869e4443ad2ced9c3508a8b407a2a2dd40d5c6 1.2s
=> => naming to docker.io/library/myapp_6533803220 0.0s

```

เล็ก

## Lab Worksheet

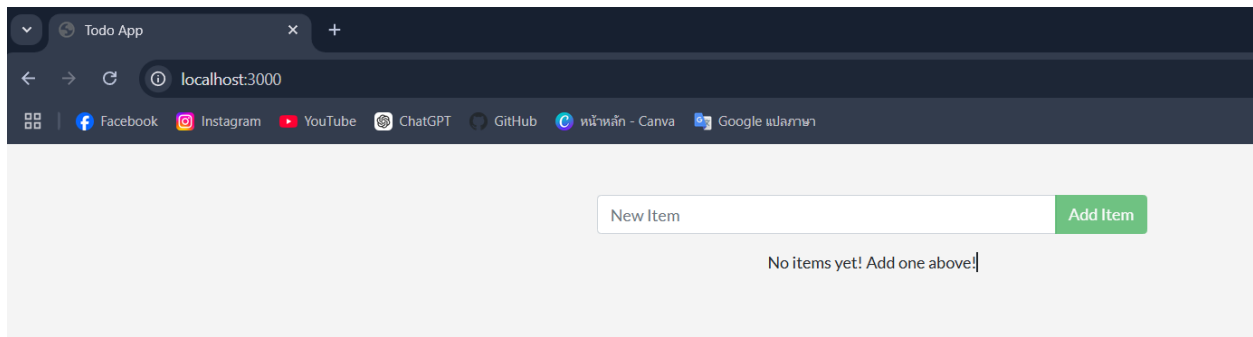
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

```
$ 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

```
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803220
>>
31cd95e863ae3efac7d477096a7b7b0cfd46b34d8048d250bcf4019ceebea1044
PS C:\Users\KKU650001\Lab8_4\getting-started\app>
```



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

เล็ก

## Lab Worksheet

```

52     return (
53         <React.Fragment>
54             <AddItemForm onNewItem={onNewItem} />
55             {items.length === 0 && (
56                 <p className="text-center">There is no TODO item. Please add one to the list. By Jonathan Doillon 653380322-0</p>
57             )}
58             {items.map(item => (
59                 <ItemDisplay
60                     item={item}
61                     key={item.id}
62                     onItemUpdate={onItemUpdate}
63                     onItemRemoval={onItemRemoval}
64                 />
65             ))}
66         </React.Fragment>
67     );

```

```
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker build -t myapp_6533803220 .
```

```

[+] Building 26.0s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 154B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> [internal] load build context
=> => transferring context: 8.10kB
=> CACHED [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> => exporting layers
=> => writing image sha256:24c952bf1d2620b8c886b03d4e161a4862d5d0075b52f338f62651e8c1f568ab
=> => naming to docker.io/library/myapp_6533803220
PS C:\Users\KKU650001\Lab8_4\getting-started\app>

```

```
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803220
```

```

72b2f5736395669f49f789af5f1c57b96a3e47d5e1b587e1e077df48f7102f4d
docker: Error response from daemon: driver failed programming external connectivity on endpoint naughty_lederberg (916282ae0e270b10ac187af042df2
2e16355c9b64218de826f83f567a4efad28): Bind for 0.0.0.0:3000 failed: port is already allocated.

```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

ตอบ

Error : port 3000 ถูกใช้งานอยู่แล้ว โดยมี container อื่น กำลังใช้พอร์ตนี้

Docker ไม่สามารถ Bind พอร์ต 3000 บนเครื่อง Host ได้ เพราะมี Process อื่นจับ พอร์ตอยู่  
เกิดขึ้นเพราะ :

มี Container อื่นที่ยังรันอยู่ และใช้พอร์ต 3000

Process อื่นนอก Docker (เช่น Node.js, React, หรือแอปอื่น) กำลังใช้พอร์ต 3000

เล็ก

## Lab Worksheet

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ 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. 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

```
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker ps -a
>>
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                    NAMES
72b2f5736395   myapp_6533803220 "docker-entrypoint.s..." 9 minutes ago   Created                                naughty_lederber
rg
31cd95e863ae   7cdfec9c837    "docker-entrypoint.s..." 17 minutes ago  Up 17 minutes    0.0.0.0:3000->3000/tcp    inspiring_mclar
en
12cfd33c9b2a   joe3220/lab8    "/bin/sh -c 'echo "J..." 43 minutes ago  Exited (0) 43 minutes ago                crazy_robinson
42590e3c0598   my-first-image  "/bin/sh -c 'echo "J..." 54 minutes ago  Exited (0) 54 minutes ago                eager_nobel
ac8756e3ca29   busybox         "Hello Jonathan Doil..." 2 hours ago     Created                                distracted_swir
les
8c051be05eaf   busybox        "sh"                     2 hours ago     Exited (0) 2 hours ago                reverent_hawkin
g
238611a84d44   busybox        "sh"                     2 hours ago     Exited (0) 2 hours ago                peaceful_hofsta
dter
52e2b8d30b2e   busybox        "echo 'Hello Joonath..." 2 hours ago     Exited (0) 2 hours ago                serene_edison
ce89ee35c0db   busybox        "sh"                     2 hours ago     Exited (0) 2 hours ago                nervous_jang
4e8367aeb7b4   busybox        "sh"                     2 hours ago     Exited (0) 2 hours ago                awesome_heyrovs
ky
```

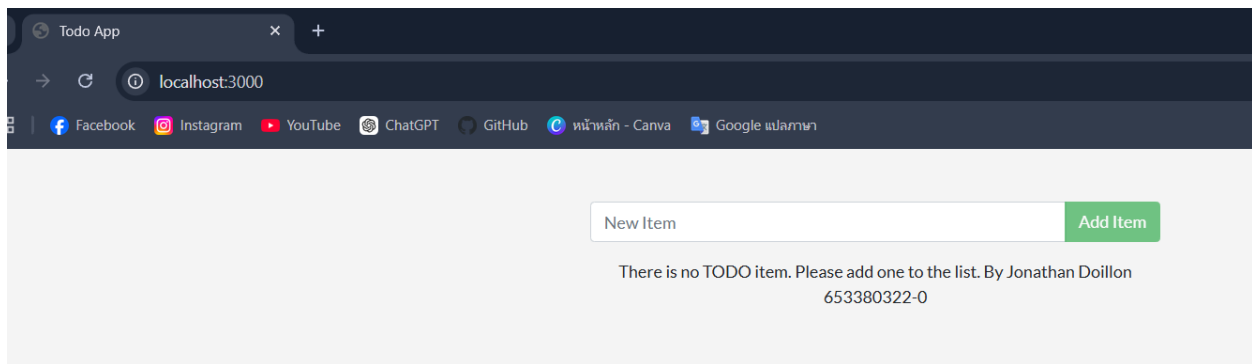
เล็ก

## Lab Worksheet

```

PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker stop 31cd95e863ae
31cd95e863ae
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker rm 31cd95e863ae
31cd95e863ae
PS C:\Users\KKU650001\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803220
>>
8d27037f534cc79ec119284eab32ace967ea06d7540b32900fd7746211f6bb89
PS C:\Users\KKU650001\Lab8_4\getting-started\app>

```



Containers

Images

Volumes

Builds

Docker Hub

Docker Scout

Extensions

Containers

[Give feedback](#)

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

Container CPU usage0.00% / 1200% (12 CPUs available)

Container memory usage19.76MB / 2.76GB

Show charts

Search

Only show running containers

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> serene_edison	52e2b8d30b2e	busybox		0%	2 hours ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> nervous_jang	ce89ee35c0db	busybox		0%	2 hours ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> awesome_heyrovsky	4e8367aeb7b4	busybox		0%	2 hours ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> peaceful_hofstadter	238611a84d44	busybox		0%	2 hours ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> reverent_hawking	8c051be05eaf	busybox		0%	2 hours ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> distracted_swirles	ac8756e3ca29	busybox		0%		<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> eager_nobel	42590e3c0598	my-first-image		0%	1 hour ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> crazy_robinson	12cfd33c9b2a	joe3220/lab8		0%	51 minutes ago	<div><div></div><div></div><div></div></div>
<input type="checkbox"/>	<input type="radio"/> naughty_leiderberg	72b2f5736395	myapp_6533803220	3000:3000	0%		<div><div></div><div></div><div></div></div>
<input checked="" type="checkbox"/>	<input checked="" type="radio"/> keen_sanderson	8d27037f534c	myapp_6533803220	3000:3000	0%	5 minutes ago	<div><div></div><div></div><div></div></div>

เล็ก

## 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:

```
9b455d8b666a47c694b402b0d74cf7ff
```

This may also be found at: /var/jenkins\_home/secrets/initialAdminPassword

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

```
2025-01-29 15:00:45.798+0000 [id=55] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 15:00:45.822+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-29 15:00:47.811+0000 [id=68] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2025-01-29 15:00:47.813+0000 [id=68] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt
_#1
```

Activate

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



เล็ก

## Lab Worksheet

The screenshot shows the 'Getting Started' section of the Jenkins installation wizard. The main heading is 'Instance Configuration'. Below it, there is a text input field for 'Jenkins URL' with the value 'http://localhost:8080/lab8'. A paragraph explains that the Jenkins URL is used for absolute links and environment variables. Another paragraph notes that the default value is 'not saved yet' and is generated from the current request. At the bottom, there are two buttons: 'Not now' and 'Save and Finish'.

Getting Started

## Instance Configuration

Jenkins URL:

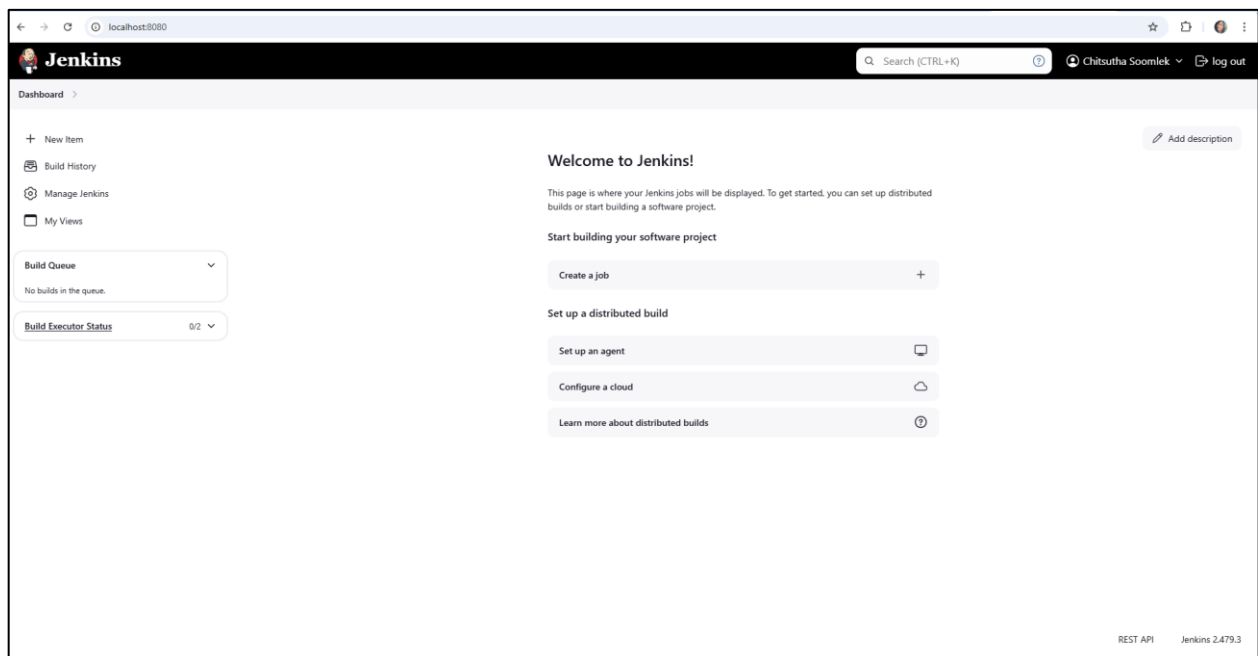
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD\_URL environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.479.3 Not now Save and Finish

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

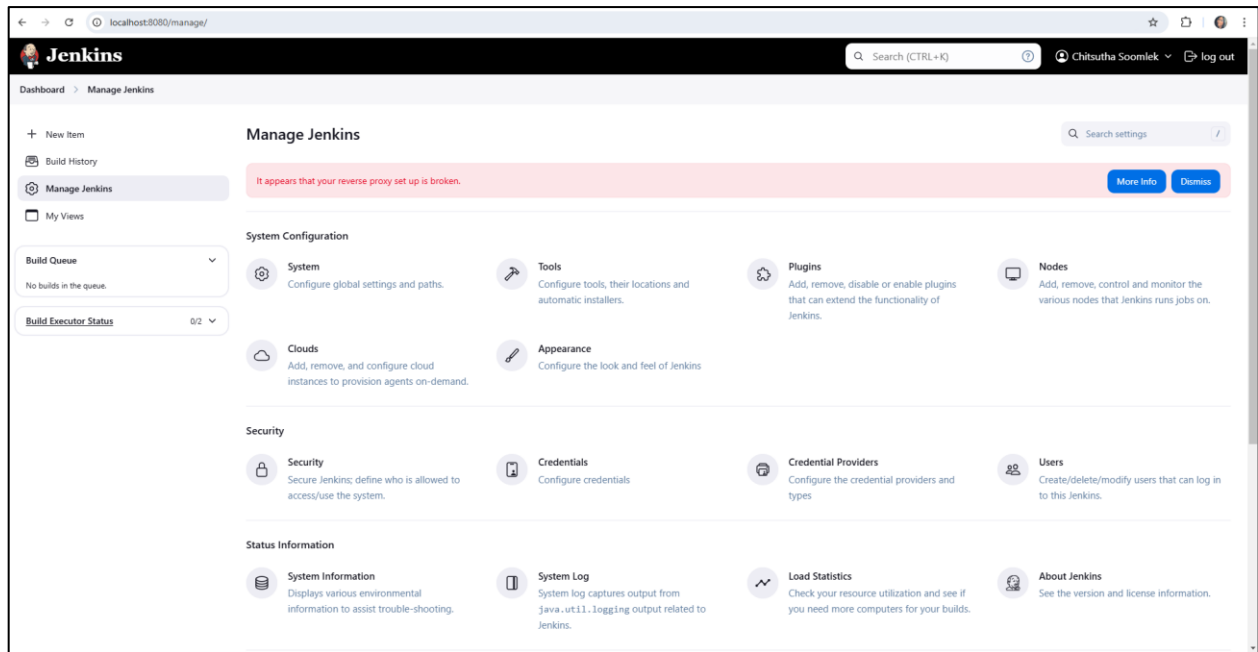
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



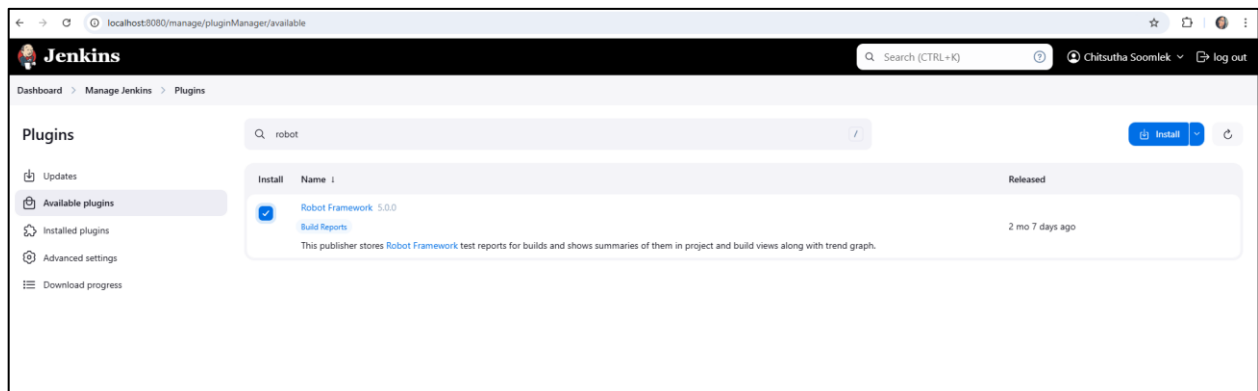
เล็ก

## Lab Worksheet

## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



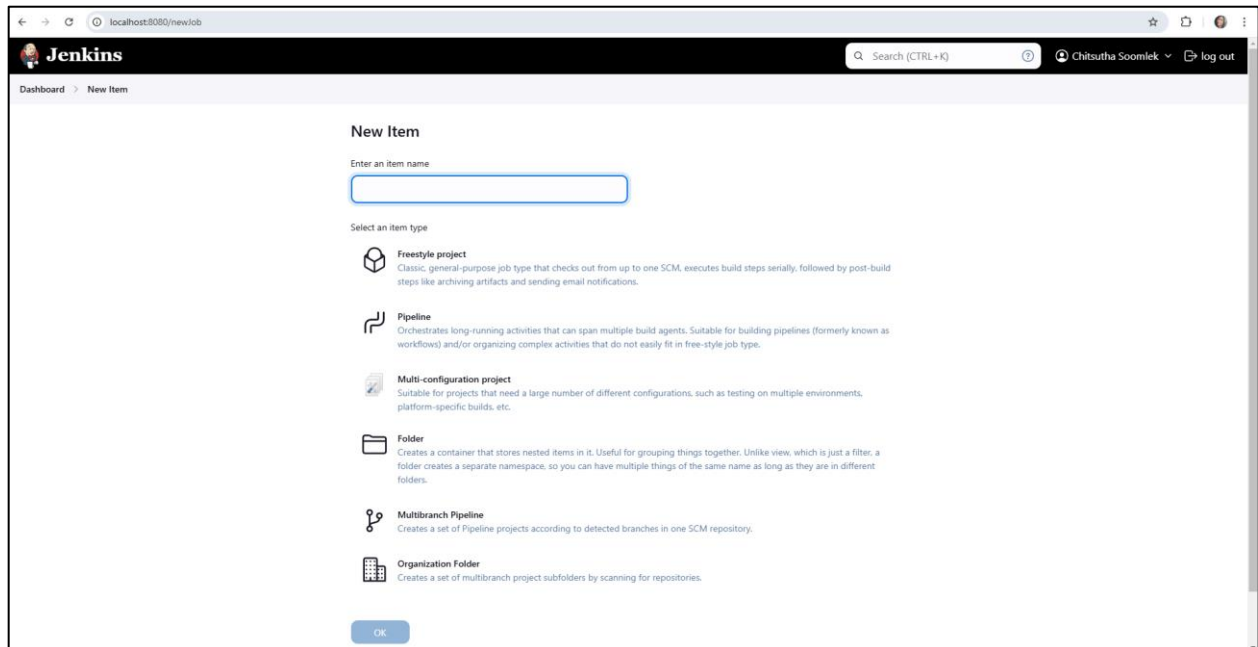
## 10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

เล็ก

## Lab Worksheet



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้น  
ตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด 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

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

เล็ก

## Lab Worksheet

**Jenkins** Search (CTRL+K) Jonathan Doillon log out

Dashboard > UAT > Configuration

### Configure

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

#### General

Description: Lab 8.5

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ? <https://github.com/kku-computer-science/configuration-management-Joey322-0.git>

[Advanced](#) ▾

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

[Advanced](#) ▾

[Save](#) [Apply](#)

Activate Windows  
Go to Settings to activate Windows.

---

Dashboard > UAT > Configuration

### Configure

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

#### 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 Wednesday, January 29, 2025 at 3:35:31 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 3:50:31 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) ?

[Save](#) [Apply](#)

Activate Windows  
Go to Settings to activate Windows.

เล็ก

## Lab Worksheet

**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

```
cd Lab8_653380322-0
mkdir -p results
robot --outputdir results *.robot
```

Advanced

Add build step

**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)

results

Save Apply

Activate Windows  
Go to Settings to activate Windows.

---

Dashboard > UAT > Configuration

**Configure**

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

Add build step

**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)

results

Advanced Edited

Thresholds for build result

20.0

80.0

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

☐ Include skipped tests in total count for thresholds

Add post-build action

Save Apply

Activate Windows  
Go to Settings to activate Windows.

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

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

ตอบ cd Lab8\_653380322-0  
robot \*.robot

เล็ก

## 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)

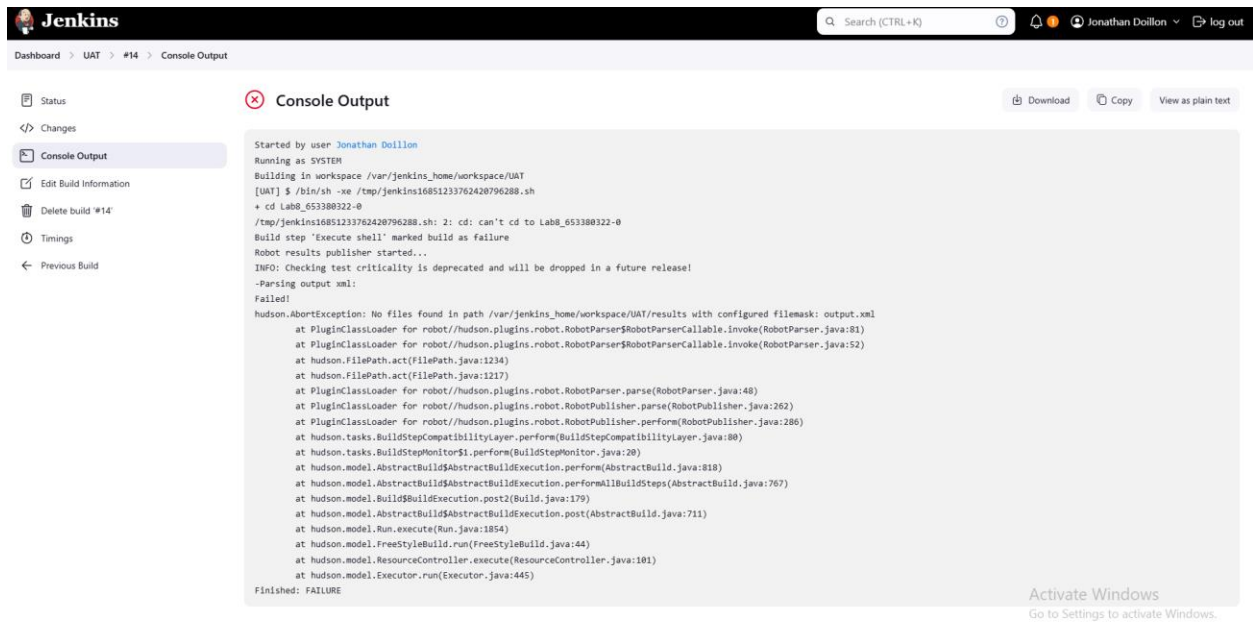
13. กด Apply และ Save

14. สั่ง Build Now

The screenshot shows the Jenkins web interface. At the top is the Jenkins logo and a search bar. Below the header, the breadcrumb 'Dashboard > UAT' is visible. The main content area is divided into two columns. The left column contains a sidebar with icons for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', 'Robot Results', 'GitHub', and 'Rename'. The right column displays the 'UAT' project details, including 'Lab 8.5', 'Latest Robot Results' (showing 'No results available yet'), and 'Permalinks' (listing links for the last build, failed build, unsuccessful build, and completed build, all from 37 seconds ago). A 'Builds' panel on the left shows a list of recent builds, all marked as failed (red 'X' icon) with timestamps ranging from 4:19 PM to 4:50 PM. An 'Activate Windows' watermark is visible in the bottom right corner.

เล็ก

## Lab Worksheet



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information for Jonathan Doillon. The breadcrumb trail is Dashboard > UAT > #14 > Console Output. On the left sidebar, the 'Console Output' tab is selected. The main area displays the console output for build #14, which has failed. The output text is as follows:

```
Started by user Jonathan Doillon
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/sh -xe /tmp/jenkins16851233762420796288.sh
+ cd Lab0_653380322-0
/tmp/jenkins16851233762420796288.sh: 2: cd: can't cd to Lab0_653380322-0
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:
Failed!
hudson.AbortException: No files found in path /var/jenkins_home/workspace/UAT/results with configured filemask: output.xml
    at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:81)
    at PluginClassLoader for robot/hudson.plugins.robot.RobotParser$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:101)
    at hudson.model.Executor.run(Executor.java:445)
Finished: FAILURE
```

At the bottom right of the console output area, there is a watermark that says 'Activate Windows. Go to Settings to activate Windows.'

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