

Lab Worksheet

ชื่อ-นามสกุล นายวริศ ทองนาม รหัสนักศึกษา 653380147-2 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

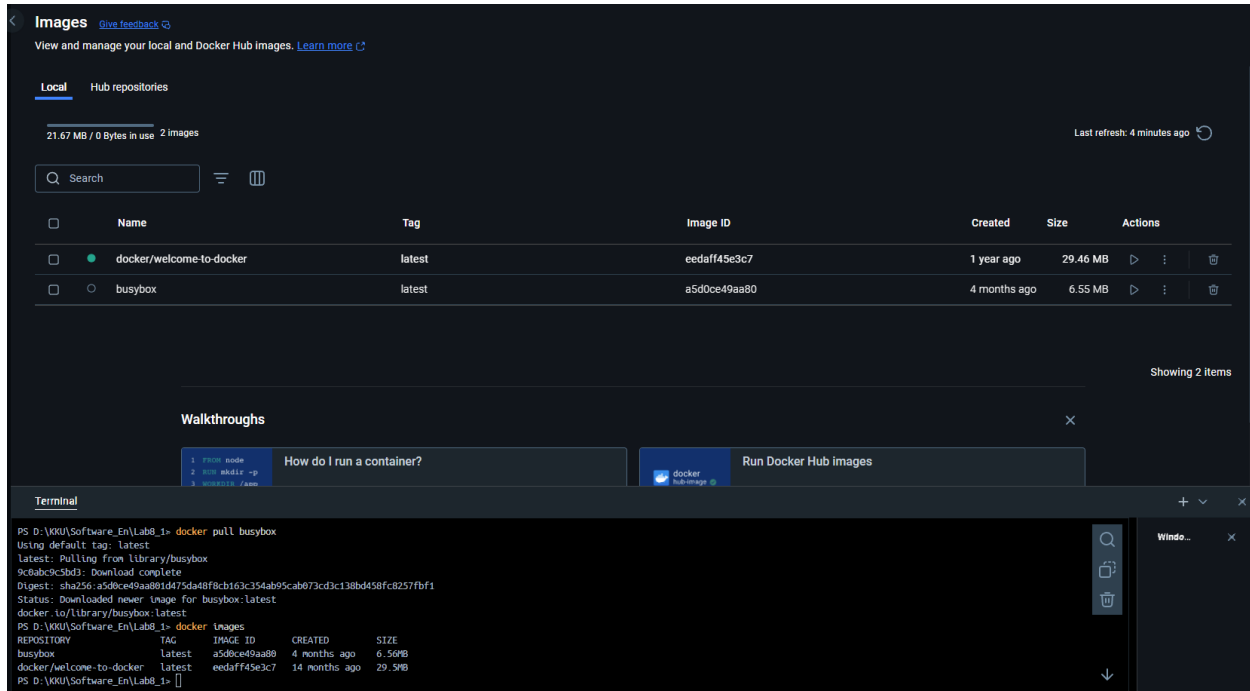
1. ติดตั้ง Docker desktop ลงบนเครื่องคอมพิวเตอร์ โดยดาวน์โหลดจาก <https://www.docker.com/get-started>
2. สร้าง Account บน Docker hub (<https://hub.docker.com/signup>)
3. กำหนดให้ \$ หมายถึง Command prompt และ <> หมายถึง ให้ป้อนค่าของพารามิเตอร์ที่กำหนด

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



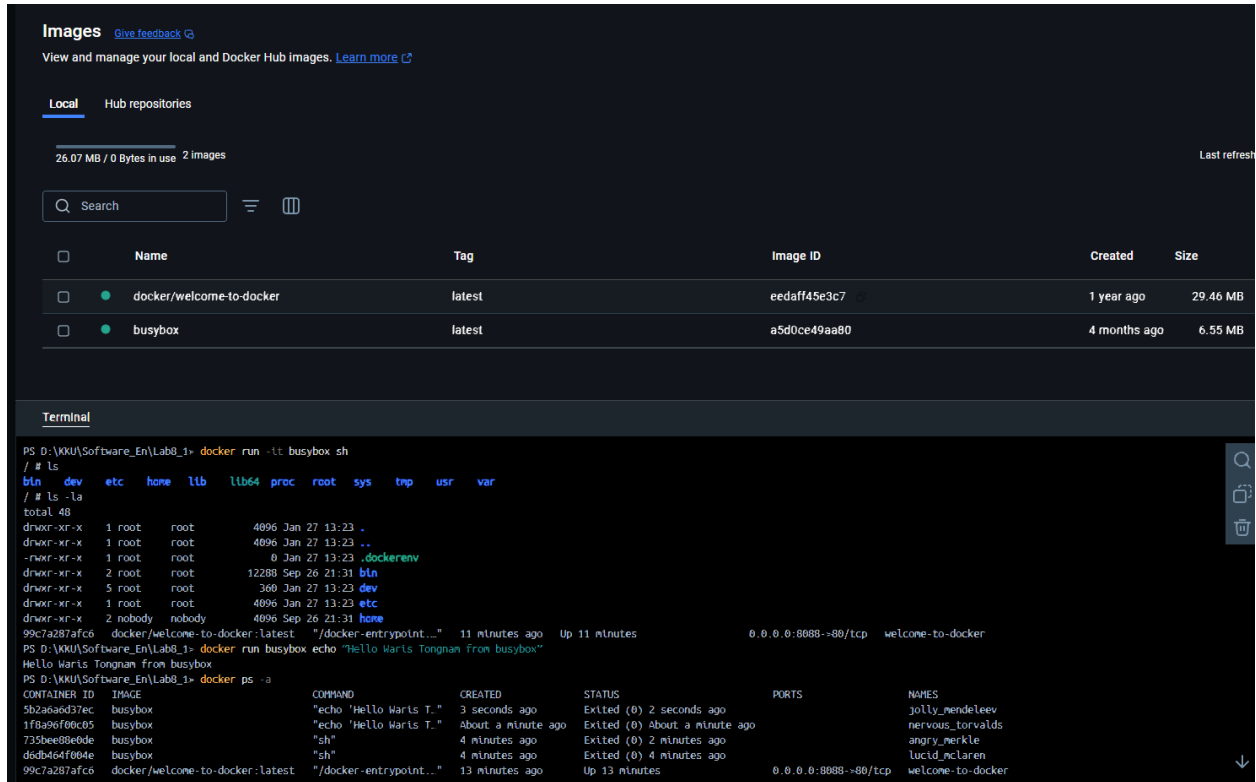
(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร แสดงชื่อของ Docker image ที่เก็บไว้ในระบบหรือที่ถูกดึงมาจาก Docker Hub

(2) Tag ที่ใช้บ่งบอกถึงอะไร คอลัมน์ Tag ใช้บ่งบอก เวอร์ชัน หรือ ตัวระบุ ของ image

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

Lab Worksheet

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



(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
สามารถโต้ตอบกับ container ได้โดยตรงในลักษณะ interactive mode

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร
คอลัมน์ STATUS บอกสถานะของ container เช่น กำลังรันหรือหยุด พร้อม exit code และเวลาล่าสุดที่เปลี่ยนแปลง

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

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

Lab Worksheet

Containers [Give feedback](#)

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

Container CPU usage 0.00% / 1200% (12 CPUs available) Container memory usage 9.43MB / 15.15GB

Search Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	● welcome-to-docker	99c7a287afc6	docker/welcome-to-docker:latest	8088:80	0%	23 minutes ago
<input type="checkbox"/>	○ lucid_mclaren	d6db464f004e	busybox		0%	15 minutes ago
<input type="checkbox"/>	○ angry_merkle	735bee88e0de	busybox		0%	14 minutes ago
<input type="checkbox"/>	○ nervous_torvalds	1f8a96f00c05	busybox		0%	12 minutes ago

Terminal

```
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS          NAMES
5b2a6ad37ec   busybox       "echo 'Hello Warts T..." 3 seconds ago   Exited (0)    2 seconds ago   jolly_wendeleev
1f8a96f00c05   busybox       "echo 'Hello Warts T..." About a minute ago   Exited (0)    About a minute ago   nervous_torvalds
735bee88e0de   busybox       "sh"                     4 minutes ago   Exited (0)    2 minutes ago   angry_merkle
d6db464f004e   busybox       "sh"                     4 minutes ago   Exited (0)    4 minutes ago   lucid_mclaren
99c7a287afc6   docker/welco-  "sh"                     13 minutes ago   Up 13 minutes   0.0.0.0:8088->80/tcp   welcome-to-docker

PS D:\KKU\Software_En\Lab8_1> docker create --name test_container test
PS D:\KKU\Software_En\Lab8_1> docker create --name test_container test
PS D:\KKU\Software_En\Lab8_1>
Inable to find image 'test:latest' locally
Error response from daemon: pull access denied for test, repository does not exist or may require 'docker login'
PS D:\KKU\Software_En\Lab8_1> docker rm 5b2a6ad37ecbd8c9246488bed7563b57289748cdaa658f66964ddd819d3361
5b2a6ad37ecbd8c9246488bed7563b57289748cdaa658f66964ddd819d3361
PS D:\KKU\Software_En\Lab8_1>
```

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

Lab Worksheet

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

The screenshot shows the Docker Desktop interface. At the top, there's a 'Local' tab under 'Images'. Below it, a table lists three local images:

Name	Tag	Image ID	Created	Size
docker/welcome-to-docker	latest	eedaff45e3c7	1 year ago	29.46 MB
busybox	latest	a5d0ce49aa80	4 months ago	6.55 MB
testlab8	latest	9293B5c6a8ef	4 months ago	6.55 MB

Below the table, there's a 'Walkthroughs' section with two options: 'FROM scratch' and 'How do I run a container?'. The 'Terminal' tab is active, showing the following output:

```
=> => naming to docker.io/library/testlab8:latest
=> => unpacking to docker.io/library/testlab8:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/915b7969hgxnbua8dfaw5lo9

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 D:\XXU\Software_En\Lab8_2> docker run testlab8
"ชื่อ-นามสกุล นามจริง นามสกุล รหัสนักศึกษา 653380147-2 ชื่อเล่น นอรา"
PS D:\XXU\Software_En\Lab8_2> docker run -it testlab8
"ชื่อ-นามสกุล นามจริง นามสกุล รหัสนักศึกษา 653380147-2 ชื่อเล่น นอรา"
PS D:\XXU\Software_En\Lab8_2>
```

(1) คำสั่งที่ใช้ในการ run คือ `docker run testlab8`

Lab Worksheet

- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
-t ใช้เพื่อให้สามารถกำหนด ชื่อ (name) และ แท็ก (tag) ให้กับ Docker image ที่กำลังสร้างได้

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

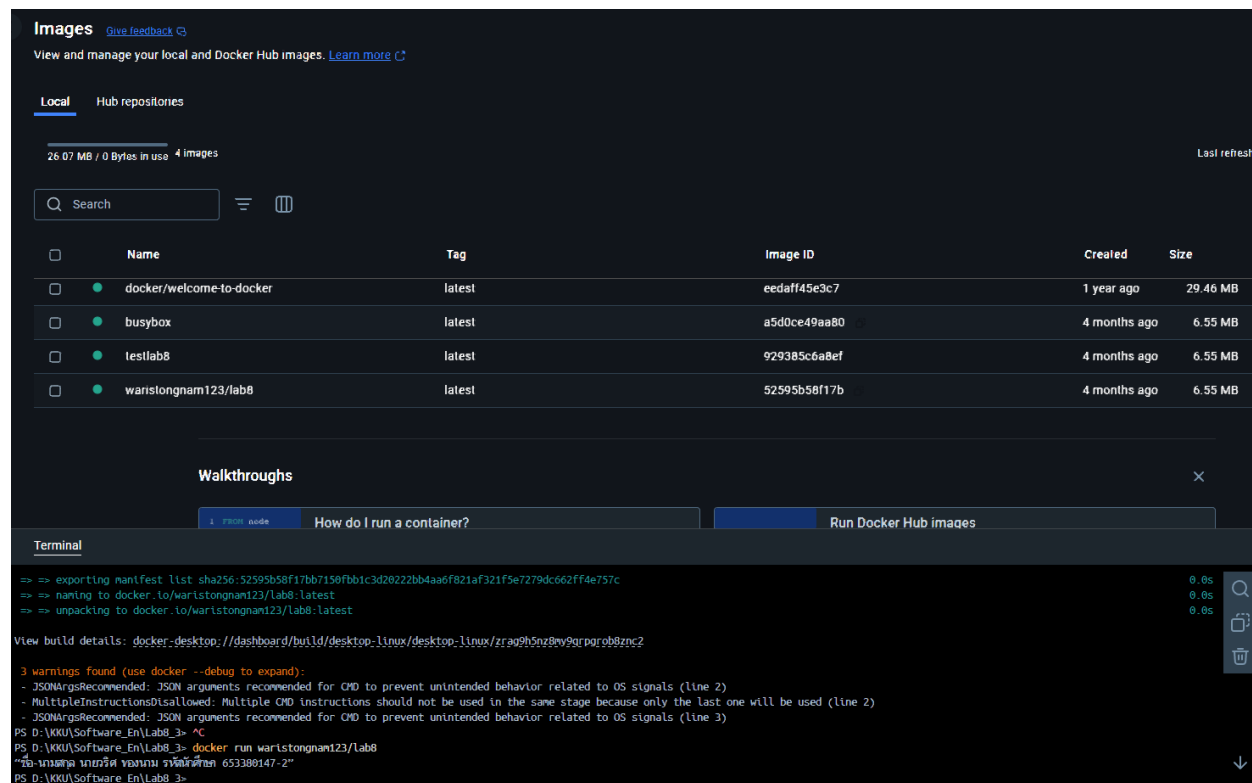
```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

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

Lab Worksheet

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

The screenshot shows the Docker Desktop interface. The top section is titled "Images" and includes a search bar and a table of local images. Below this is a "Walkthroughs" section with a "Terminal" tab. The terminal window shows the execution of Docker commands to run and push an image.

Images Section:

Name	Tag	Image ID	Created	Size	Actions
docker/welcome-to-docker	latest	eedaff45e3c7	1 year ago	29.46 MB	▶
busybox	latest	a5d0ce49aa80	4 months ago	6.55 MB	▶
testlab8	latest	929385c6a8ef	4 months ago	6.55 MB	▶
waristongnam123/lab8	latest	52595b58f17b	4 months ago	6.55 MB	▶

Terminal Section:

```

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 D:\KKU\Software_En\Lab8_3> ^C
PS D:\KKU\Software_En\Lab8_3> docker run waristongnam123/lab8
"ชื่อ-นามสกุล นามจริง นามนาม รหัสประจำตัว 653380147-2"
PS D:\KKU\Software_En\Lab8_3> docker push waristongnam123/lab8
Using default tag: latest
The push refers to repository [docker.io/waristongnam123/lab8]
adc1733492fb: Pushed
9c0abc9c5bd3: Mounted from library/busybox
latest: digest: sha256:52595b58f17bb7158fbb1c3d20222bb4aa6f821af321f5e7279dc662ff4e757c size: 855
PS D:\KKU\Software_En\Lab8_3>
  
```

Hub repositories Section:

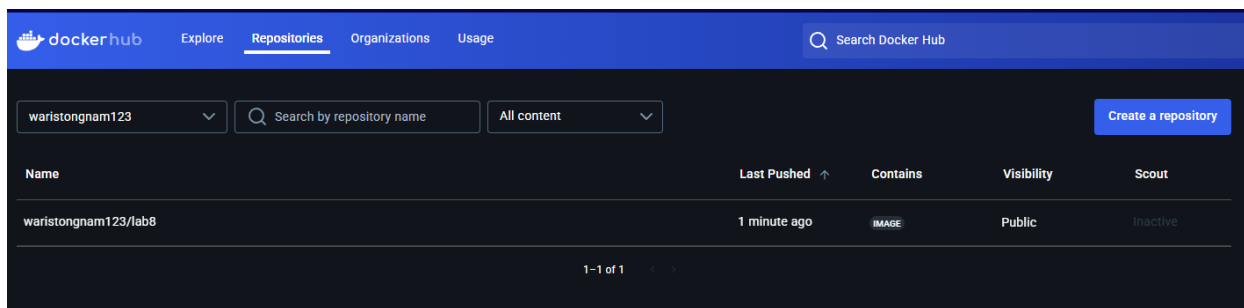
Repository	Tag	OS	Vulnerabilities	Last pushed	Size
waristongnam123/lab8	latest	Linux	Inactive	2 minutes ago	2.15 MB

Terminal Section (continued):

```

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 D:\KKU\Software_En\Lab8_3> ^C
PS D:\KKU\Software_En\Lab8_3> docker run waristongnam123/lab8
"ชื่อ-นามสกุล นามจริง นามนาม รหัสประจำตัว 653380147-2"
PS D:\KKU\Software_En\Lab8_3> docker push waristongnam123/lab8
Using default tag: latest
The push refers to repository [docker.io/waristongnam123/lab8]
adc1733492fb: Pushed
9c0abc9c5bd3: Mounted from library/busybox
latest: digest: sha256:52595b58f17bb7158fbb1c3d20222bb4aa6f821af321f5e7279dc662ff4e757c size: 855
PS D:\KKU\Software_En\Lab8_3>
  
```

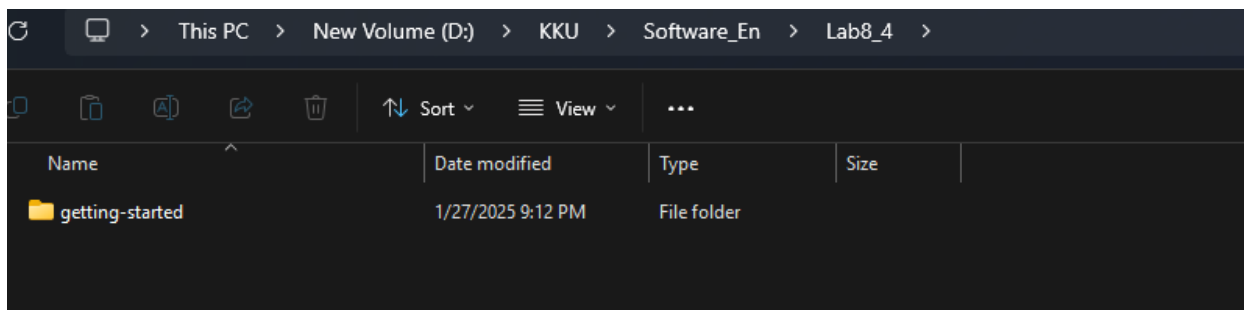

Lab Worksheet



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



Lab Worksheet

```
PS D:\KKU\Software_En\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 | 1.46 MiB/s, done.
Resolving deltas: 100% (523/523), done.
PS D:\KKU\Software_En\Lab8_4> ls
```

Directory: D:\KKU\Software_En\Lab8_4

Mode	LastWriteTime	Length	Name
d-----	1/27/2025 9:14 PM		getting-started

Lab Worksheet

```

D: > KKU > Software_En > Lab8_4 > getting-started > app > {} package.json > ...
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 }
34

```

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์

FROM node:18-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

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

Lab Worksheet

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

```

FROM node:22-alpine

WORKDIR /app
COPY . .

# ติดตั้งแพ็คเกจโดยไม่ติดตั้ง devDependencies
RUN npm install --omit=dev

CMD ["node", "src/index.js"]
EXPOSE 3000
  
```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

Lab Worksheet

\$ docker run -dp 3000:3000 <myapp_รหัสสนศ. ไม่มีชี้>

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

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

The screenshot shows the Docker Desktop interface. At the top, there's a search bar and a toggle for 'Only show running containers'. Below this is a table of containers:

Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
lucid_mclaren	d6db464f004e	busybox		0%	3 hours ago	Stop, Restart, Kill, Logs, Details
angry_merkle	735bee88e0de	busybox		0%	3 hours ago	Stop, Restart, Kill, Logs, Details
nervous_torvalds	1f8a96f00c05	busybox		0%	3 hours ago	Stop, Restart, Kill, Logs, Details
nice_curran	a1f49a0031c2	testlab8		0%	2 hours ago	Stop, Restart, Kill, Logs, Details
adoring_bhaskara	c68c513abcb3	testlab8		0%	2 hours ago	Stop, Restart, Kill, Logs, Details
pedantic_hofstadter	069523e0f08f	waristongnam123/lab8		0%	2 hours ago	Stop, Restart, Kill, Logs, Details
competent_bell	0545206c37f0	myapp_6533801472	3000:3000	0%	45 seconds ago	Stop, Restart, Kill, Logs, Details

Below the table is a terminal window showing the following commands and output:

```

=> [4/4] RUN npm install --omit=dev
=> exporting to image
=> exporting layers
=> exporting manifest sha256:d78eb926165238a27618a213725074b07fae844bee964f61583da09d472b0479
=> exporting config sha256:e9e172ee30a129d9cd25f0b3aa93669fb752fdd88f551251d828d445ad543a81
=> exporting attestation manifest sha256:668bd9e0d1edab940217e836caecb450ec9bf99ab768205a0105563ab46ed49
=> exporting manifest list sha256:ea9071aa42edc8c0e549fa067dafeaea9366302d5d810607238cd934b9e94083
=> naming to docker.io/library/myapp_6533801472:latest
=> unpacking to docker.io/library/myapp_6533801472:latest
  
```

At the bottom, there's a 'Terminal' tab with a search bar and a list of build details:

```

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/b80c910901f0b041xmkqndx
PS D:\XKU\Software_En\lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533801472
0545206c37f09a6613fe32b1065764136ed5d453674bfbab77758a1be996f00
PS D:\XKU\Software_En\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

Lab Worksheet

<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

Images [Give feedback](#)

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

Local Hub repositories

341.26 MB / 0 Bytes in use 5 Images Last refresh

	Name	Tag	Image ID	Created	Size
<input type="checkbox"/>	docker/welcome-to-docker	latest	eedaff45e3c7	1 year ago	29.46 MB
<input type="checkbox"/>	busybox	latest	a5d0ce49aa80	4 months ago	6.55 MB
<input type="checkbox"/>	testlab8	latest	929385c6a8ef	4 months ago	6.55 MB
<input type="checkbox"/>	waristongnam123/lab8	latest	52595b58f17b	4 months ago	6.55 MB
<input type="checkbox"/>	myapp_6533801472	latest	34266da00a34	30 seconds ago	406.68 MB

Terminal

```

=> => exporting layers
=> exporting manifest sha256:002253e896cec7c15e54dc7a21c6d6254ff5b535af52cbaf717eb49c636beaf7
=> exporting config sha256:2b26eed477252c8f3aca8bc90089922e31e42405f294e43b2db68390d945035
=> exporting attestation manifest sha256:aefb7f08b8cc626563c51b3b37f997550a84adbe83dcdab5c7b78cd2933d1c62
=> exporting manifest list sha256:34266da00a34f0296ef6e758f69f7c332c7c3151f4b385abf01d4f9e9742f652
=> naming to docker.io/library/myapp_6533801472:latest
=> unpacking to docker.io/library/myapp_6533801472:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/29kosbez7nw1tpfxj97plsnl5
PS D:\VXU\Software\Enl\lab8_4\getting-started> docker run -dp 3000:3000 myapp_6533801472
15ba9fa75c247ac7691fc380b4ed461bd16a2ecf8d5cd9d5203fac8492bbcd
docker: Error response from daemon: driver failed programming external connectivity on endpoint lucid_ferml (0112823e7d47780e87d6867a55a35070231cefa2a5c5bf6ea7bdf000128019c4): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS D:\VXU\Software\Enl\lab8_4\getting-started>

```

```

48   });
49
50   if (items === null) return 'Loading...';
51
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 นายวิศ ทองนาม</p>
57       )}
58       {items.map(item => (
59         <ItemDisplay
60           item={item}
61           key={item.id}
62           onItemUpdate={onItemUpdate}
63           onItemRemoval={onItemRemoval}
64         />

```

Lab Worksheet

Containers [Give feedback](#)

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

Container CPU usage 0.00% / 1200% (12 CPUs available)

Container memory usage 67.4MB / 15.15GB

Search Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	lucid_mclaren	d6db464f004e	busybox		0%	3 hours ago
<input type="checkbox"/>	angry_merkle	735bee88e0de	busybox		0%	3 hours ago
<input type="checkbox"/>	nervous_torvalds	1f8a96f00c05	busybox		0%	3 hours ago
<input type="checkbox"/>	nice_curran	a1f49a0031c2	testlab8		0%	3 hours ago
<input type="checkbox"/>	adoring_bhaskara	c68c513abcb3	testlab8		0%	3 hours ago
<input type="checkbox"/>	pedantic_hofstadter	069523e0f08f	waristongnam123/lab8		0%	3 hours ago
<input checked="" type="checkbox"/>	competent_bell	0545206c37f0	myapp_6533801472	3000:3000	0%	49 minutes ago

Terminal

```

=> => exporting layers
=> => exporting manifest sha256:002253e896cec7c15e54dc7a21c6d6254ff5b535af52cbaf717eb49c636beaf7
=> => exporting config sha256:2b26eed477252c8f3aca8bc90009922e31e42485f294e453b2db60309d945035
=> => exporting attestation manifest sha256:aefb7f08b8cc626563c51b3b37f997556a84adb83dcdab5c7b78cd2933d1c62
=> => exporting manifest list sha256:34266da00a34f0296ef6e758f69f7c332c7c3151f4b385abf81d4f9e9742f652
=> => naming to docker.io/library/myapp_6533801472:latest
=> => unpacking to docker.io/library/myapp_6533801472:latest

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/29k04bez7nw1tpfxj07p1sm15
PS D:\XXU\Software_En\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533801472
15ba0fa75c247ac7691fc380b4ed461bd16a2ecfad5cd9d5203fac0492bbcd
docker: Error response from daemon: driver failed programming external connectivity on endpoint lucid_fermt (8112823e7d47780e87d6867a55a35070231cefa2a5c5bf6ea7bdf000128019c4): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS D:\XXU\Software_En\Lab8_4\getting-started\app>

```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

หมายความว่า port 3000 ถูกใช้งานอยู่ เกิดขึ้นเพราะมีการใช้ port ที่ซ้ำกัน

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

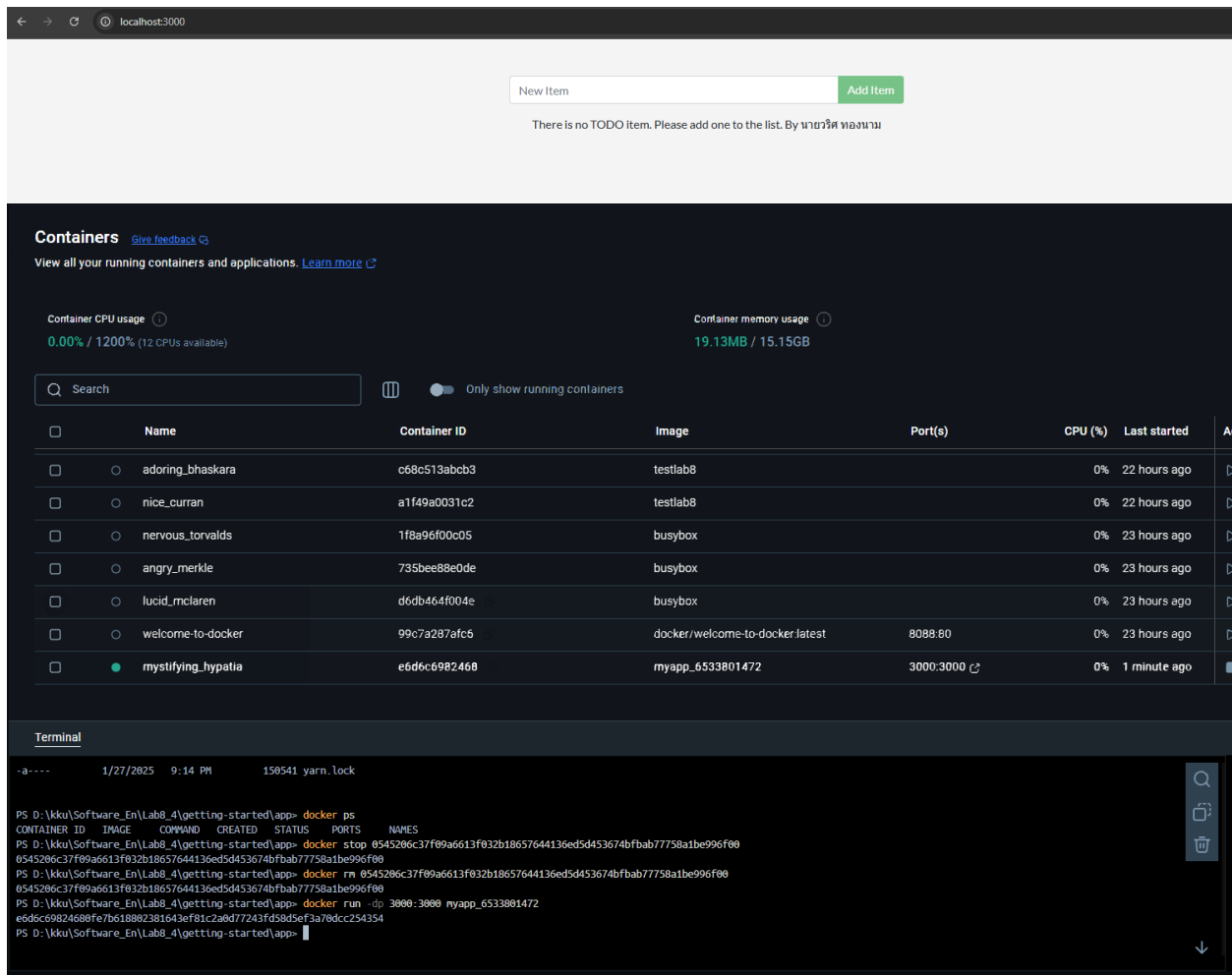
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

Lab Worksheet

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



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

Lab Worksheet

Getting Started

Create First Admin User

Username

waris_1472

Password

.....

Confirm password

Full name

E-mail address

Jenkins 2.479.3

Skip and continue as admin

Save and Continue

Getting Started

Instance Configuration

Jenkins URL:

http://localhost:8080/lab8

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.

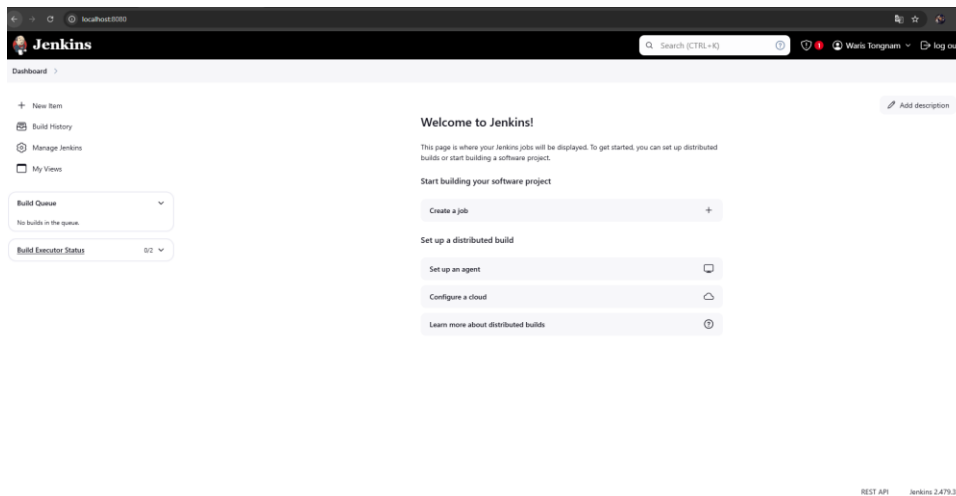
Jenkins 2.479.3

Not now

Save and Finish

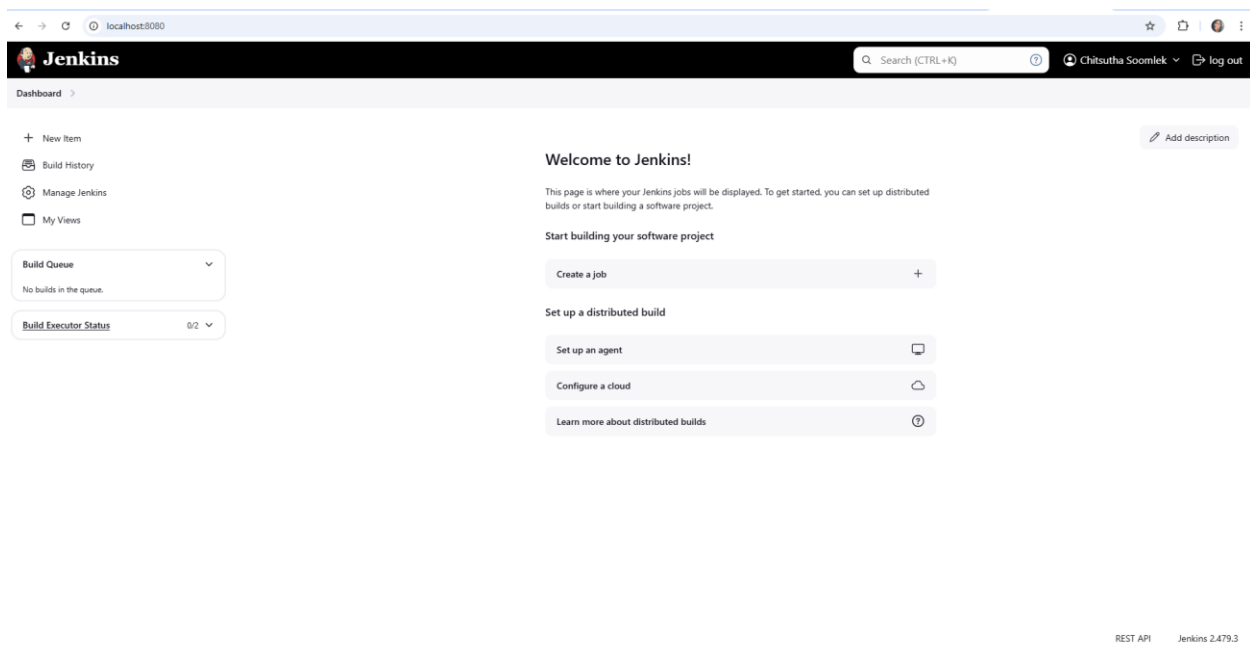
18

Lab Worksheet



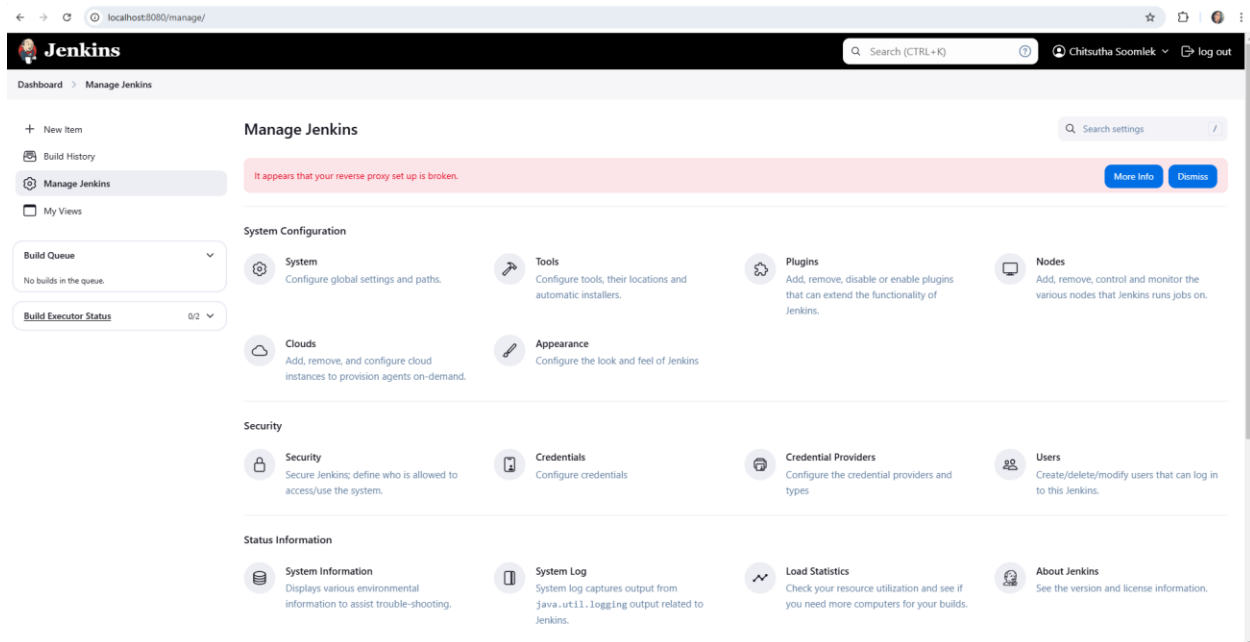
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

Lab Worksheet

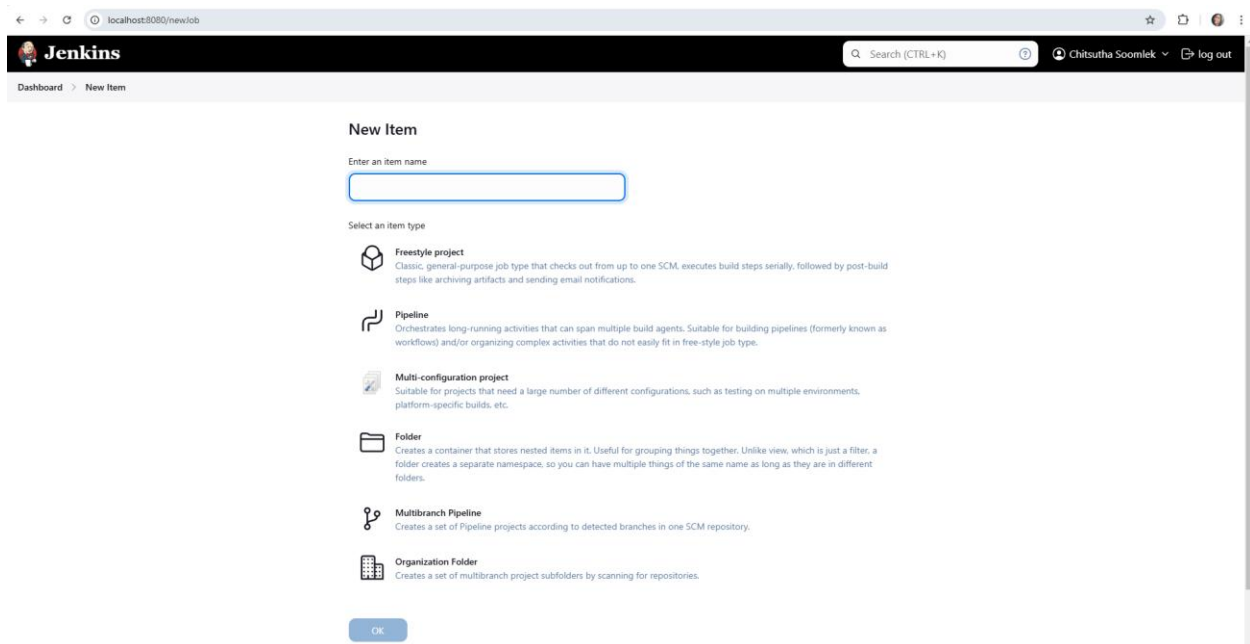


10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

Lab Worksheet



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

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 job named 'UAT'. The page is divided into two main sections: 'General' and 'Source Code Management'.

General Section:

- Description:** Lab 8.5
- Plain text:** Preview
- ☐ Discard old builds ?
- ☒ GitHub project
 - Project url ?** https://github.com/waristongnam/Lab8.5.git
 - Advanced** (dropdown)
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Execute concurrent builds if necessary ?
- Advanced** (dropdown)

Source Code Management Section:

- ☐ Git ?

Build Triggers Section:

- ☐ 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 1:50:25 PM Coordinated Universal Time; would next run at Tuesday, January 28, 2025 at 2:05:25 PM Coordinated Universal Time.
- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

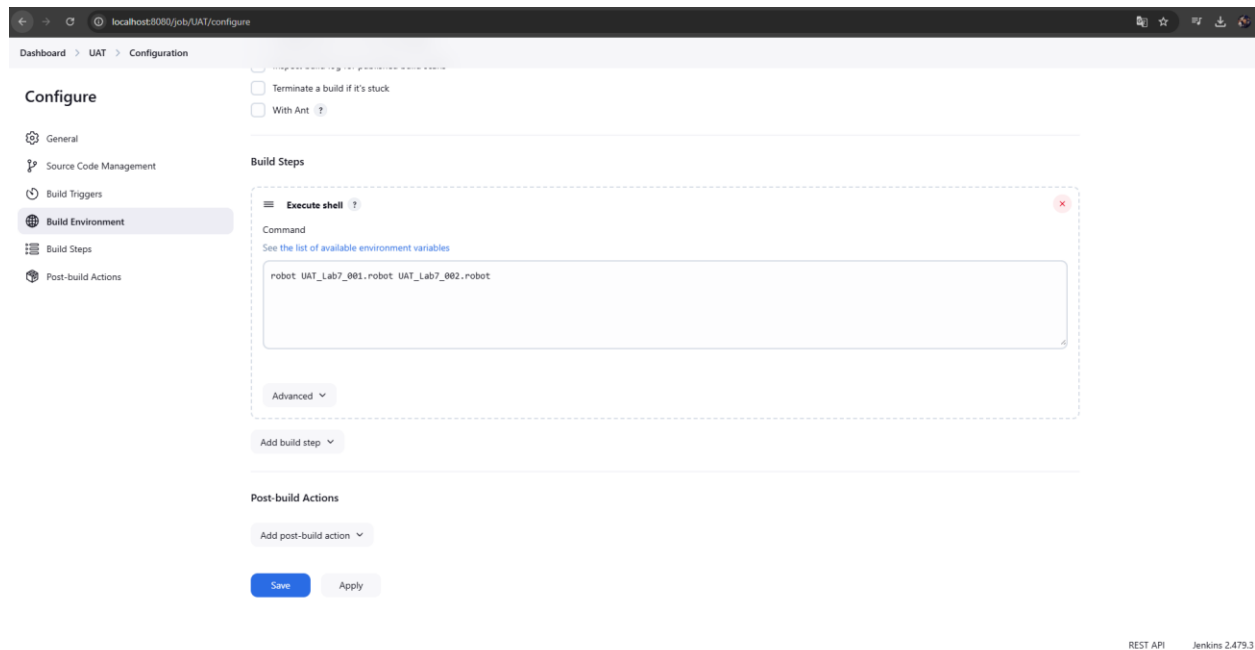
Build Environment Section:

- ☐ 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 Name Section:

- Save** **Apply**

Lab Worksheet



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

`robot UAT_Lab7_001.robot`

`robot UAT_Lab7_002.robot`

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

localhost:8080/job/UAT/

Jenkins Search (CTRL+K) Waris Tongnam log out

Dashboard > UAT >

Status **UAT** Edit description

Changes

Workspace

Build Now

Configure

Delete Project

Robot Results

GitHub

Rename

Builds Filter /

28 snmru 2568

#40 16:12

#39 16:10

#38 16:05

#37 16:04

#36 16:01

#35 15:56

#34 15:54

#33 15:54

#32 15:51

Lab 8.5

Latest Robot Results:

Total Failed Passed Skipped Pass %

All tests 1 1 0 0 0.0

- Browse results
- Open report.html
- Open log.html

Permalinks

- Last build (#40), 12 sec ago
- Last failed build (#40), 12 sec ago
- Last successful build (#40), 12 sec ago
- Last completed build (#40), 12 sec ago

Robot Framework Tests Trend (all tests)

Number of test cases

Build

Zoom to changes Show only failed Show only critical all Max builds Show bigger image

Robot Framework Test Results

Executed: 2025-01-28T22:32:35.435788

Duration: 0:00:09.340 (±0)

Status: 1 critical test, 0 passed, 1 failed, 0 skipped
1 test total (±0), 0 passed, 1 failed, 0 skipped

Results: [report.html](#)
[log.html](#)
[Original result files](#)

Test Result Trend

Number of test cases

Build

Zoom to changes Show only failed Show only critical all Max builds Show bigger image

Duration Trend

Duration (s)

Build

Show bigger image

Failed Test Cases

Name	Crit.	Duration	Age
UAT Lab7 001.Test Form Submission with Valid Data	no	0:00:08.953	2

Test Suites

Lab Worksheet

The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information (Waris Tongnam). The breadcrumb trail is Dashboard > UAT > #40 > Console Output. On the left sidebar, the 'Console Output' tab is selected. The main area displays the console output for build #40, which includes the following text:

```
Started by user Waris Tongnam
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/UAT/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/waristongnam/Lab8.5.git # timeout=10
Fetching upstream changes from https://github.com/waristongnam/Lab8.5.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/waristongnam/Lab8.5.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 90ec991178ac54572c019eaca70c9b5aab9c1322 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 90ec991178ac54572c019eaca70c9b5aab9c1322 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 90ec991178ac54572c019eaca70c9b5aab9c1322 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins4132367249580022902.sh
+ robot UAT_Lab7_001.robot
/tmp/jenkins4132367249580022902.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 xml:
Done!
WARNING! Could not find file: log.html
WARNING! Could not find file: report.html
-Copying log files to build dir:
Done!
-Assigning results to build:
-
```