

## Lab Worksheet

ชื่อ-นามสกุล นางสาวศุภวรรณ เพ็ญศรี นักศึกษา 653380027-2 Section 2

## 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

```

Terminal

Directory: D:\mykku\selab

ode                LastWriteTime                Length Name
---                -
-----          1/28/2025  11:54 PM                Lab8_1

S D:\mykku\selab> cd Lab8_1
S D:\mykku\selab\Lab8_1> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
S D:\mykku\selab\Lab8_1> docker pull
docker pull" requires exactly 1 argument.
See 'docker pull --help'.

Usage:  docker pull [OPTIONS] NAME[:TAG|@DIGEST]

Download an image from a registry
S D:\mykku\selab\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
c0abc9c5bd3: Already exists
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
S D:\mykku\selab\Lab8_1> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
busybox latest af4709625109 4 months ago 4.27MB
S D:\mykku\selab\Lab8_1>

```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร

Ans. Image name

(2) Tag ที่ใช้บ่งบอกถึงอะไร

Ans. version

## Lab Worksheet

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

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

```

Terminal
-D, --debug          Enable debug mode
-H, --host list       Daemon socket to connect to
-L, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls                Use TLS; implied by --tlsverify
--tlscacert string    Trust certs signed only by this CA (default "C:\Users\Supawan.p\docker\ca.pem")
--tlscert string      Path to TLS certificate file (default "C:\Users\Supawan.p\docker\cert.pem")
--tlskey string       Path to TLS key file (default "C:\Users\Supawan.p\docker\key.pem")
--tlsverify           Use TLS and verify the remote
-v, --version         Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

PS D:\mykku\seLab\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
edac33019347   busybox   "echo 'Hello supawan...'" 41 seconds ago   Exited (0) 41 seconds ago           eager_hodgkin
e6647ca08ae37   busybox   "echo 'Hello supawan...'" About a minute ago Exited (0) About a minute ago       interesting_liskov
0bc2fc8b6f6f   busybox   "sh"                     12 minutes ago   Exited (0) 10 minutes ago           vigilant_booth
15b2223d8373   busybox   "sh"                     12 minutes ago   Exited (0) 12 minutes ago           lucid_sutherland
PS D:\mykku\seLab\Lab8_1>
  
```

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

Ans. จะสามารถใช้ Command ใน container ได้โดยตรง

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

Ans. แสดงสถานะของแต่ละ container โดยบอกถึง สถานะการทำงานปัจจุบันหรือที่ผ่านมา ของ container นั้น ๆ

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

## Lab Worksheet

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

```

Terminal
-D, --debug          Enable debug mode
-H, --host list      Daemon socket to connect to
-l, --log-level string Set the logging level ("debug", "info", "warn", "error", "fatal") (default "info")
--tls               Use TLS; implied by --tlsverify
--tlscacert string   Trust certs signed only by this CA (default "C:\\Users\\Supawan.p\\\\.docker\\ca.pem")
--tlscert string     Path to TLS certificate file (default "C:\\Users\\Supawan.p\\\\.docker\\cert.pem")
--tlskey string      Path to TLS key file (default "C:\\Users\\Supawan.p\\\\.docker\\key.pem")
--tlsverify         Use TLS and verify the remote
-v, --version        Print version information and quit

Run 'docker COMMAND --help' for more information on a command.

For more help on how to use Docker, head to https://docs.docker.com/go/guides/

PS D:\mykku\selab\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
edac33019347   busybox   "echo 'Hello supawan..'" 41 seconds ago   Exited (0) 41 seconds ago           eager_hodgkin
e6647ca8ae37   busybox   "echo 'Hello supawan..'" About a minute ago Exited (0) About a minute ago       interesting_liskov
8bc2fc8b0f6f   busybox   "sh"                     12 minutes ago   Exited (0) 10 minutes ago           vigilant_booth
15b2223d8373   busybox   "sh"                     12 minutes ago   Exited (0) 12 minutes ago           lucid_sutherland

PS D:\mykku\selab\Lab8_1> docker rm edac33019347
edac33019347
PS D:\mykku\selab\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 และป้อนคำสั่งต่อไปนี้

\$ cat > Dockerfile << EOF

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

## Lab Worksheet

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t &lt;ชื่อ Image&gt; .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```

Terminal

Start a build
PS D:\mykku\selab\Lab8_2> docker build -t jib .
[+] Building 0.2s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 151B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent uninte 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be u 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent uninte 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:b2b4a43e3aba86adb42abb7009b5e6688fffe6ad5a2ddb451bb8d87 0.0s
=> => naming to docker.io/library/jib                           0.0s

View build details: docker-desktop:///dashboard/build/desktop-linux/desktop-linux/wdc7sj9sn
z60h6izw1mh8i03k

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 sam
e 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:\mykku\selab\Lab8_2> Docker run jib
"supawanpensri 653380027-2 jib"
PS D:\mykku\selab\Lab8_2>

```

## Lab Worksheet

(1) คำสั่งที่ใช้ในการ run คือ

Docker run jib

(2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
ตั้งชื่อและติด tag ให้กับ Docker image ที่สร้างขึ้น

### 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

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

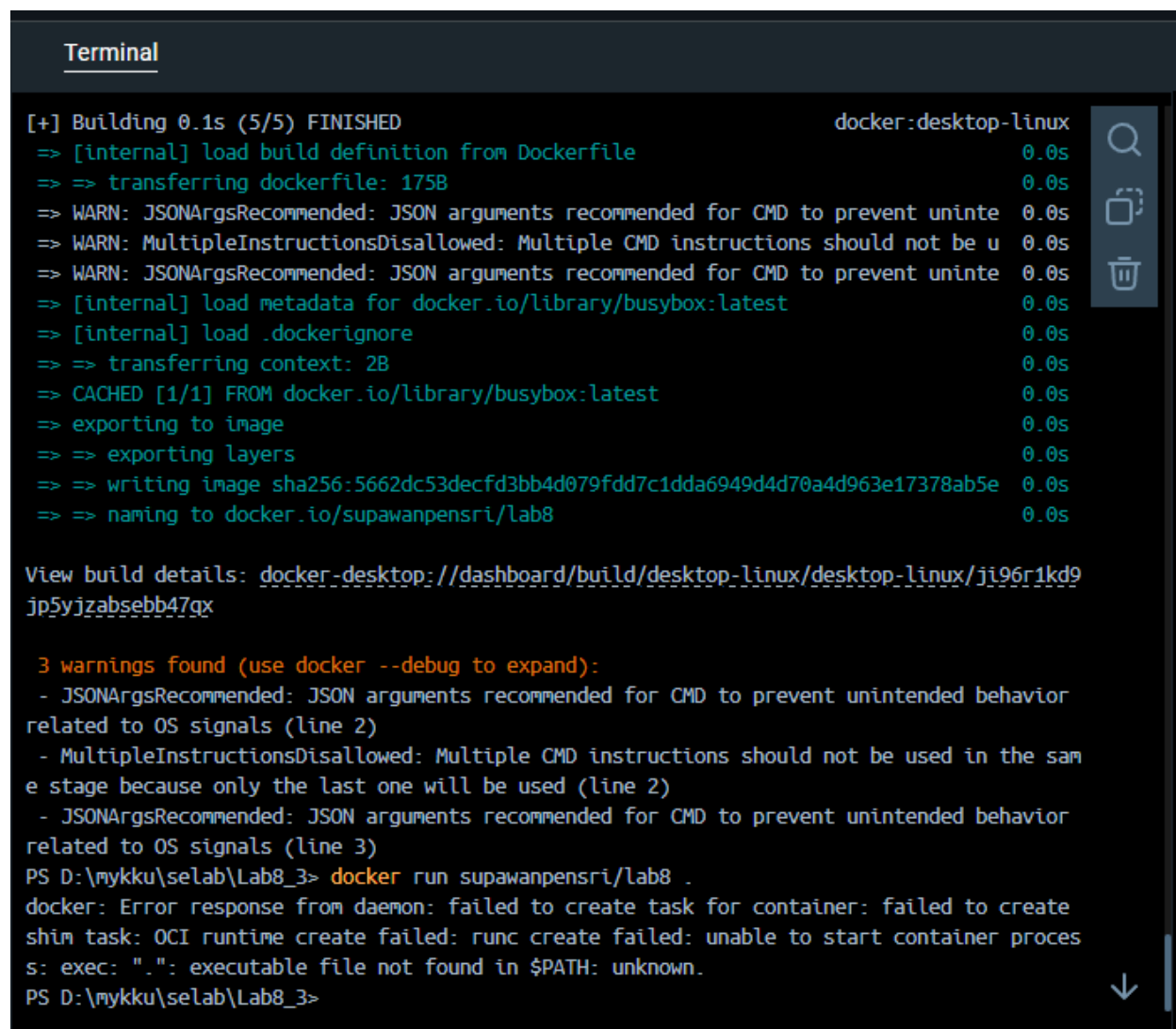
## Lab Worksheet

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

5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

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

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



```

Terminal
[+] Building 0.1s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile              0.0s
=> => transferring dockerfile: 175B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent uninte 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be u 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent uninte 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 sha256:5662dc53decfd3bb4d079fdd7c1dda6949d4d70a4d963e17378ab5e 0.0s
=> => naming to docker.io/supawanpensri/lab8                    0.0s

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/ji96r1kd9
jp5yvjzabsebb47qx

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 sam
e 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:\mykku\selab\Lab8_3> docker run supawanpensri/lab8 .
docker: Error response from daemon: failed to create task for container: failed to create
shim task: OCI runtime create failed: runc create failed: unable to start container proces
s: exec: ".": executable file not found in $PATH: unknown.
PS D:\mykku\selab\Lab8_3>
  
```

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

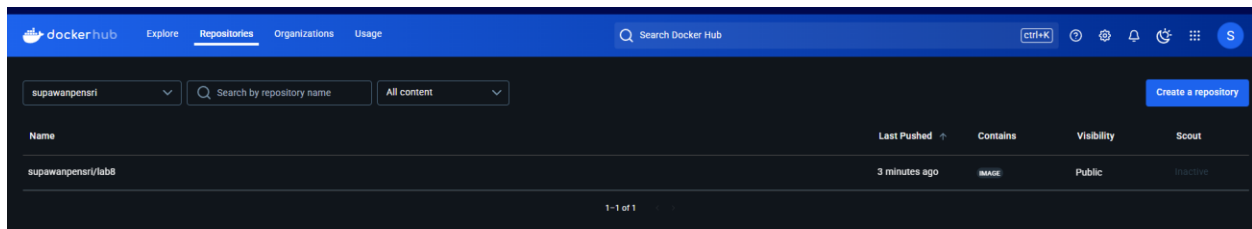
## Lab Worksheet

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

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

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

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



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

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_4

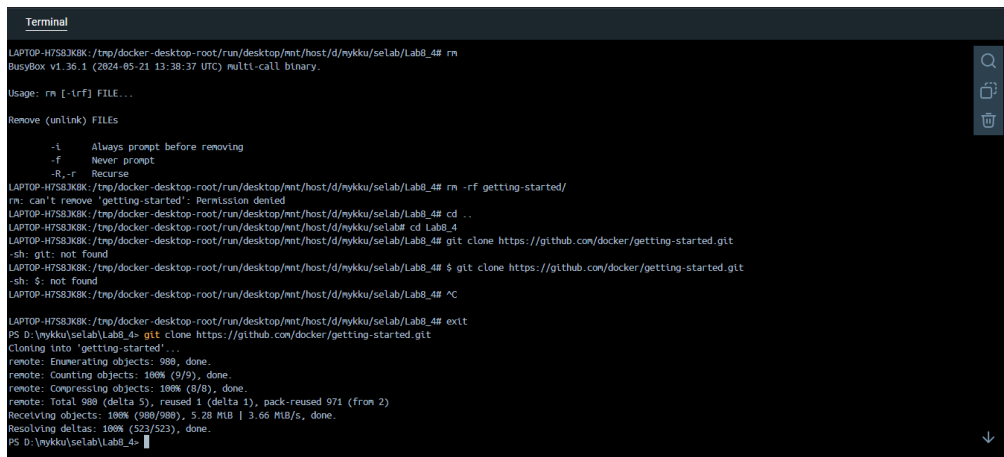
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก GitHub repository

<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

```
package.json > ...
Michael Irwin, 2 years ago | 2 authors (Michael Irwin and one other)
{
  "name": "101-app",
  "version": "1.0.0",
  "main": "index.js",
  "license": "MIT",
  "scripts": {
    "prettify": "prettier -l --write \"**/*.js\"",
    "test": "jest",
    "dev": "nodemon src/index.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "mysql2": "^2.3.3",
    "sqlite3": "^5.1.2",
    "uuid": "^9.0.0",
    "wait-port": "^1.0.4"
  },
  "resolutions": {
    "ansi-regex": "5.0.1"
  },
  "prettier": {
    "trailingComma": "all",
    "tabWidth": 4,
    "useTabs": false,
    "semi": true,
    "singleQuote": true
  },
  "devDependencies": {
    "jest": "^29.3.1",
    "nodemon": "^2.0.20",
    "prettier": "^2.7.1"
  }
}
```

## Lab Worksheet

4. ภายใต้ getting-started/app ให้สร้าง Dockerfile 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

```

Terminal
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Supawan.p> cd D:\mykku\selab\Lab8_4\getting-started\app
PS D:\mykku\selab\Lab8_4\getting-started\app> docker build -t myapp_6533880272.
[+] Building 0.0s (0/0)                                docker:desktop-linux
ERROR: unable to prepare context: path "6533880272." not found
PS D:\mykku\selab\Lab8_4\getting-started\app> docker build -t myapp_6533880272. .
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build
PS D:\mykku\selab\Lab8_4\getting-started\app> ^C
PS D:\mykku\selab\Lab8_4\getting-started\app> docker build -t myapp_6533880272 .
[+] Building 17.7s (18/18) FINISHED
==> [internal] load build definition from Dockerfile
==> [internal] load build context
==> ==> transferring context: 4.62MB
==> [2/4] WORKDIR /app
==> [3/4] COPY . .
==> [4/4] RUN yarn install --production
==> exporting to image
==> ==> exporting layers
==> ==> writing image sha256:387de6766954a6c52a23bc17ae56653c646442ab9c77e34e1f5704c517a76034
==> ==> naming to docker.io/library/myapp_6533880272

View build details: docker-desktop://dashboard/build/desktop-linux/desktop-linux/11x71ghjx64711114niburj1
PS D:\mykku\selab\Lab8_4\getting-started\app>
  
```

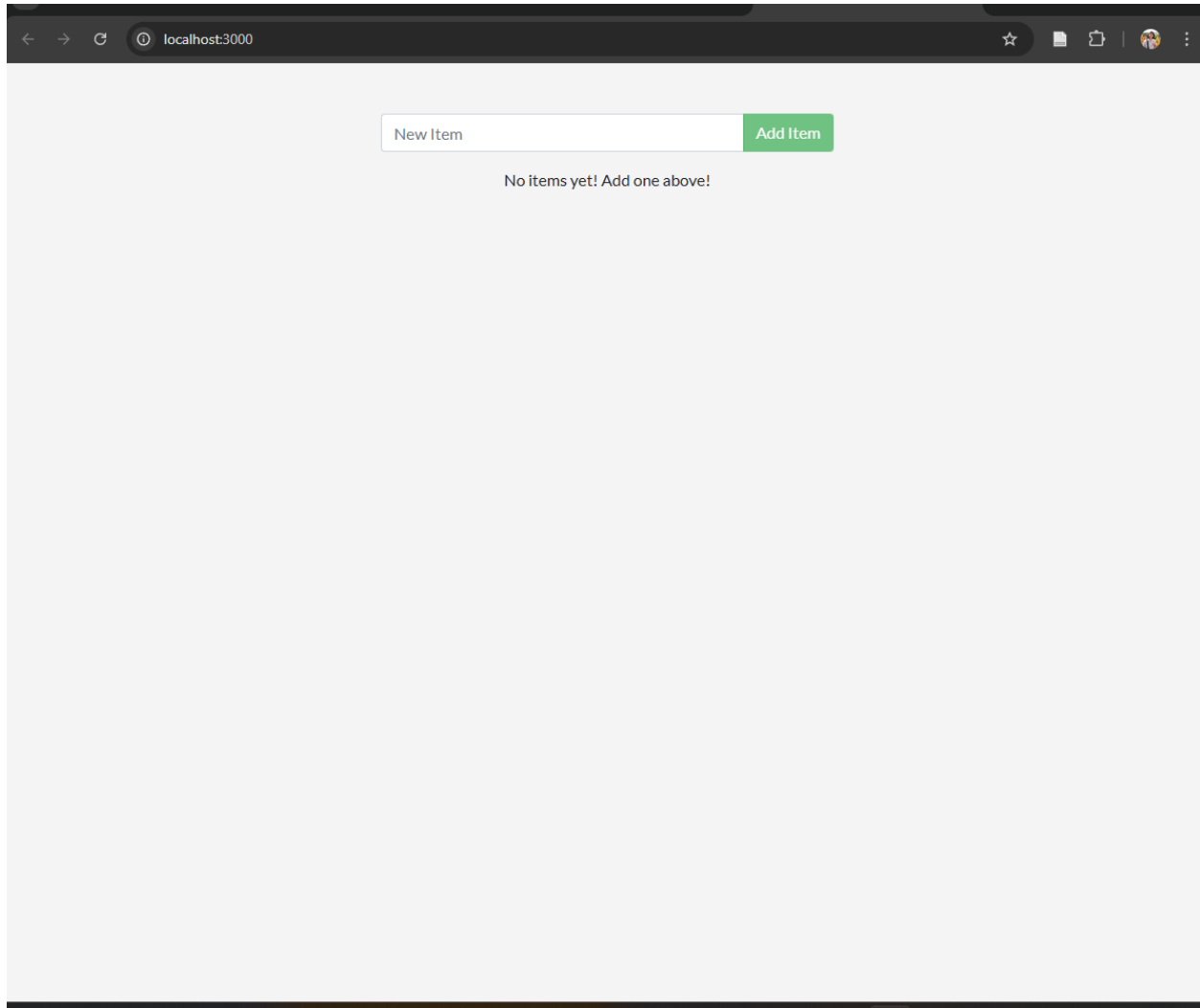
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

## Lab Worksheet

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



## Lab Worksheet

```

Terminal
=> => transferring context: 4.62MB
=> [2/4] WORKDIR /app
=> [3/4] COPY . .
=> [4/4] RUN yarn install --production
=> exporting to image
=> writing image sha256:3b7de6766954a6c52a23bc17ae56653c646442ab9c77e34e1f5704c517a76034
=> naming to docker.io/library/myapp_6533800272

View build details: docker-desktop://dashboard/build/desktop:linux/desktop:linux/11x71ghjx6471111dn1bur31
PS D:\wykku\selab\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_653380027
docker: invalid reference format.
See 'docker run --help'.
PS D:\wykku\selab\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_653380027
Unable to find image 'myapp_653380027:latest' locally
docker: Error response from daemon: pull access denied for myapp_653380027, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
PS D:\wykku\selab\Lab8_4\getting-started\app> docker login -u supawanpensri
Password:
Login Succeeded
PS D:\wykku\selab\Lab8_4\getting-started\app> docker login -u supawanpensri
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
PS D:\wykku\selab\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_653380027
Unable to find image 'myapp_653380027:latest' locally
docker: Error response from daemon: pull access denied for myapp_653380027, repository does not exist or may require 'docker login': denied: requested access to the resource is denied.
See 'docker run --help'.
PS D:\wykku\selab\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_653380027
dd9304cd4a45c98b290321054380432e7f16b570e5a6de97ab4e6f34583b376
PS D:\wykku\selab\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

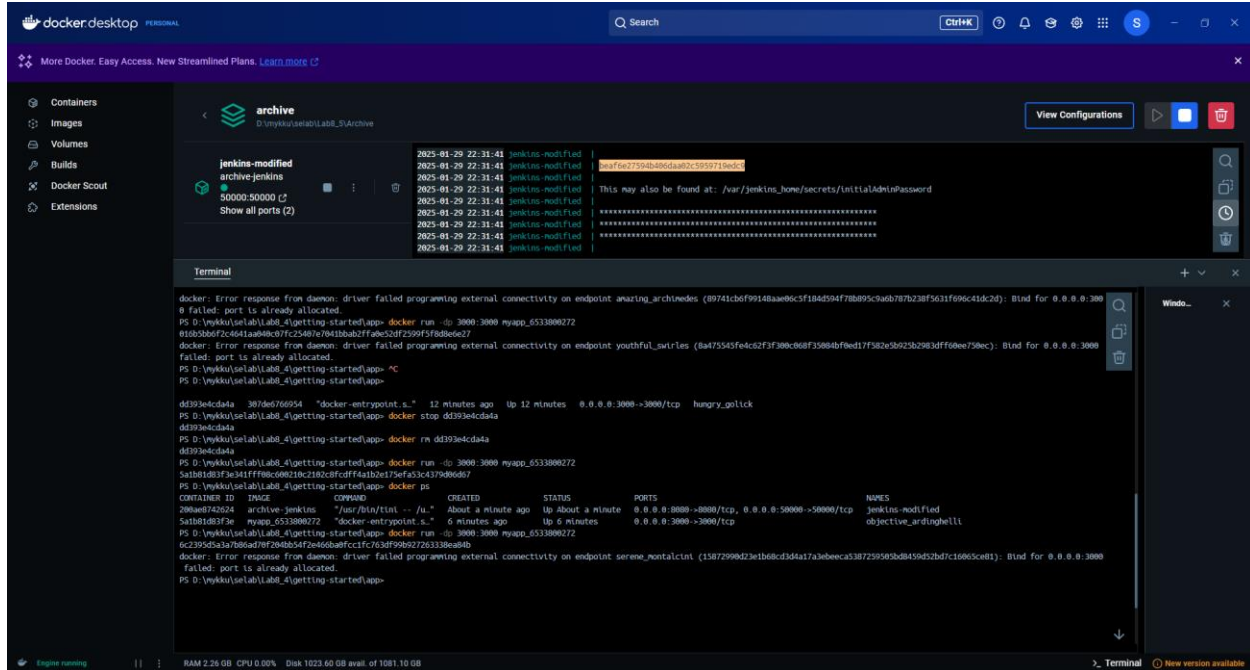
### 9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

### 10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

## Lab Worksheet

[Check point#10] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง)

แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้



(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

ตอบ มี images ที่ run port 3000 อยู่จึงไม่สามารถ run ได้

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

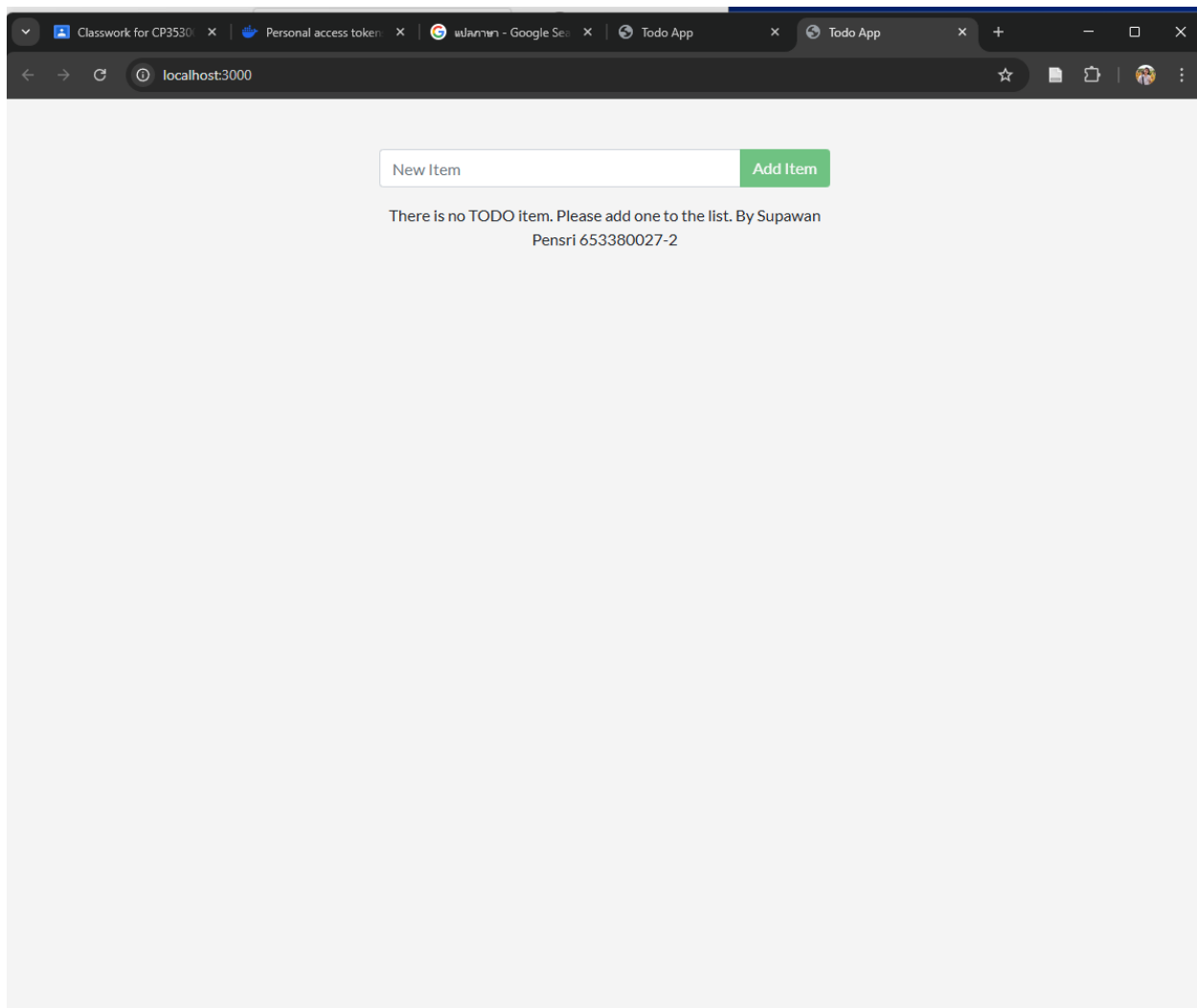
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

## 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

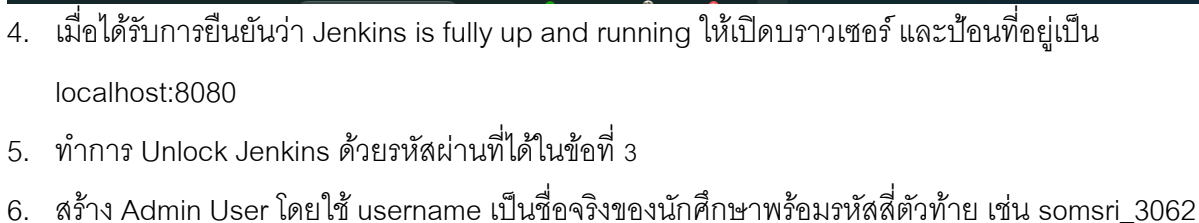
2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:its-jdk17
```

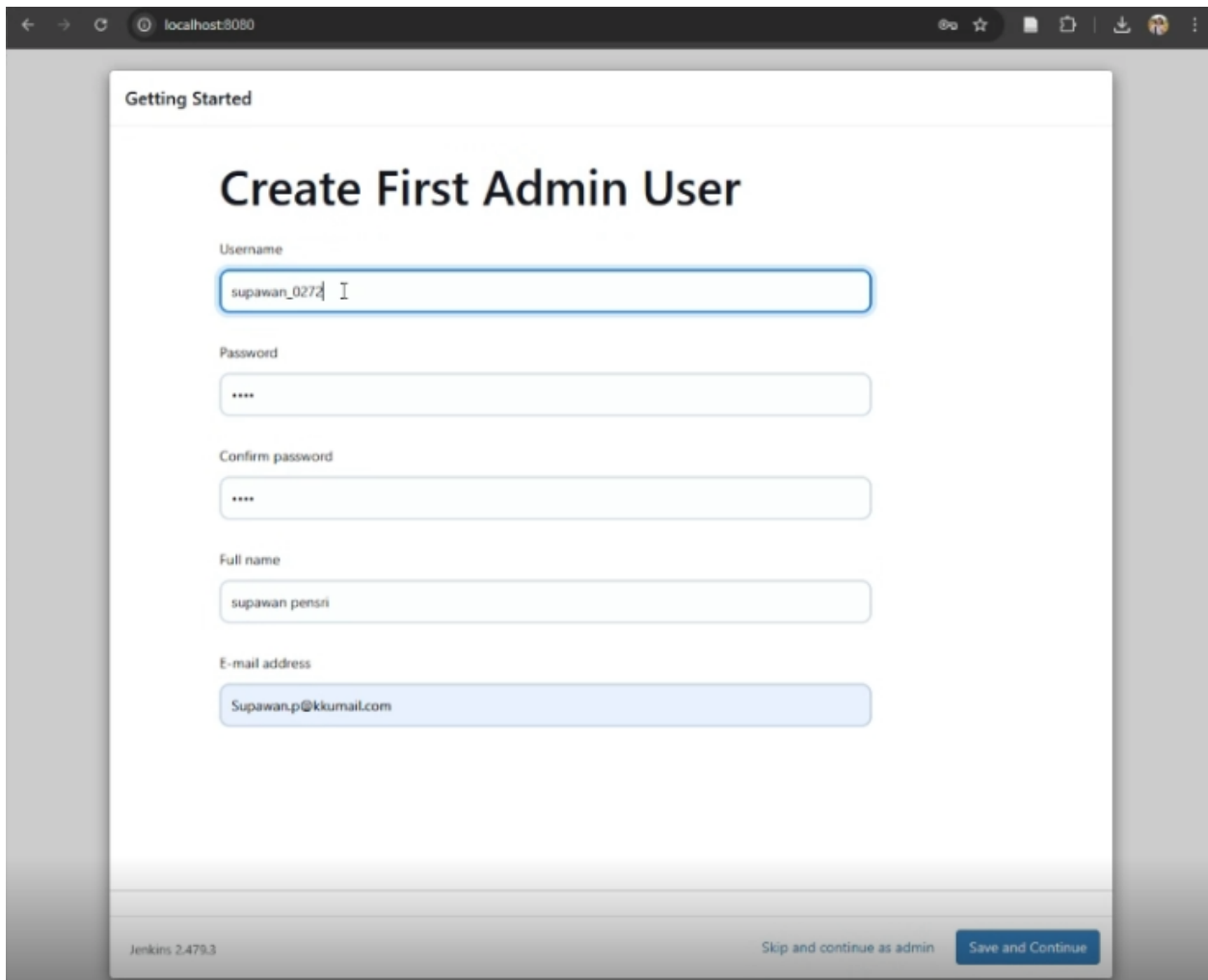
หรือ

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v
jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17
```

- [Check point#12] Capture หน้าจอที่แสดงผล Admin password



[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



The screenshot shows a web browser window at localhost:8080 displaying the 'Getting Started' page for Jenkins. The main heading is 'Create First Admin User'. The form contains the following fields and values:

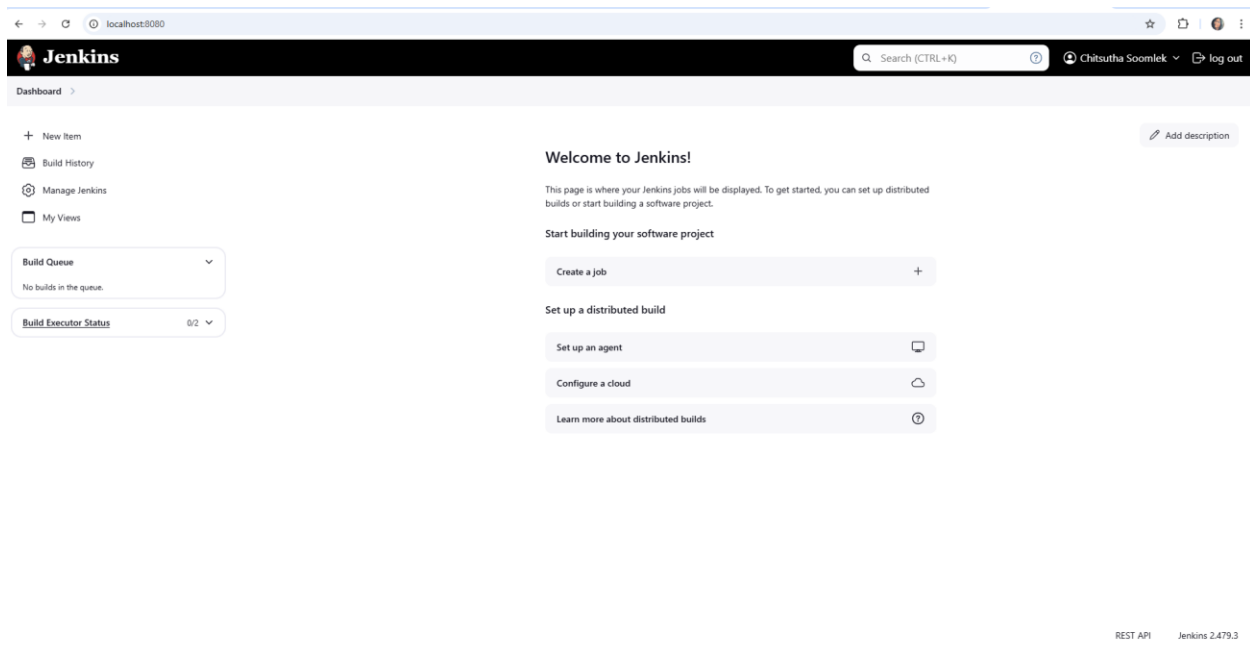
- Username:** supawan\_0272
- Password:** (masked with four asterisks)
- Confirm password:** (masked with four asterisks)
- Full name:** supawan pensri
- E-mail address:** Supawan.p@kkumail.com

At the bottom of the form, there is a 'Save and Continue' button. The footer of the page indicates 'Jenkins 2.479.3' and a link to 'Skip and continue as admin'.

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

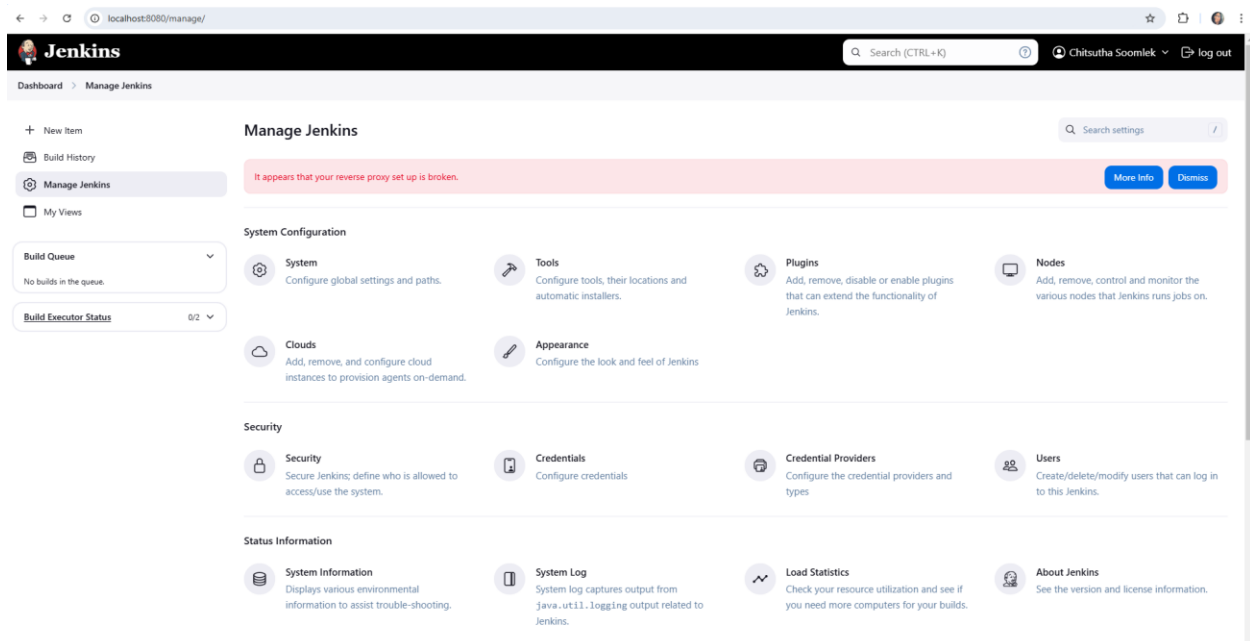


## Lab Worksheet



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

## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



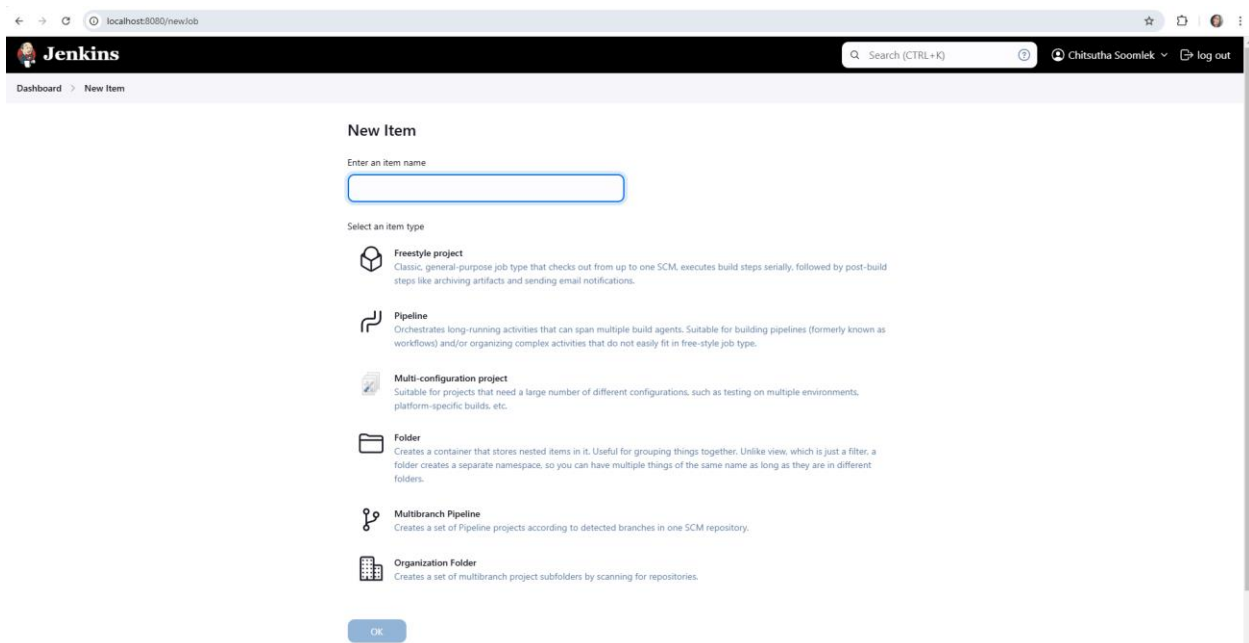
The screenshot shows the 'Manage Jenkins' page at localhost:8080/manage/. The top navigation bar is similar to the dashboard. The left sidebar highlights 'Manage Jenkins'. The main content area is titled 'Manage Jenkins' and includes a warning message: 'It appears that your reverse proxy set up is broken.' Below this, there are several configuration categories: System Configuration (System, Tools, Plugins, Nodes, Clouds, Appearance), Security (Security, Credentials, Credential Providers, Users), and Status Information (System Information, System Log, Load Statistics, About Jenkins). Each category has a brief description of its function.

## Lab Worksheet

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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



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

Description: Lab 8.5

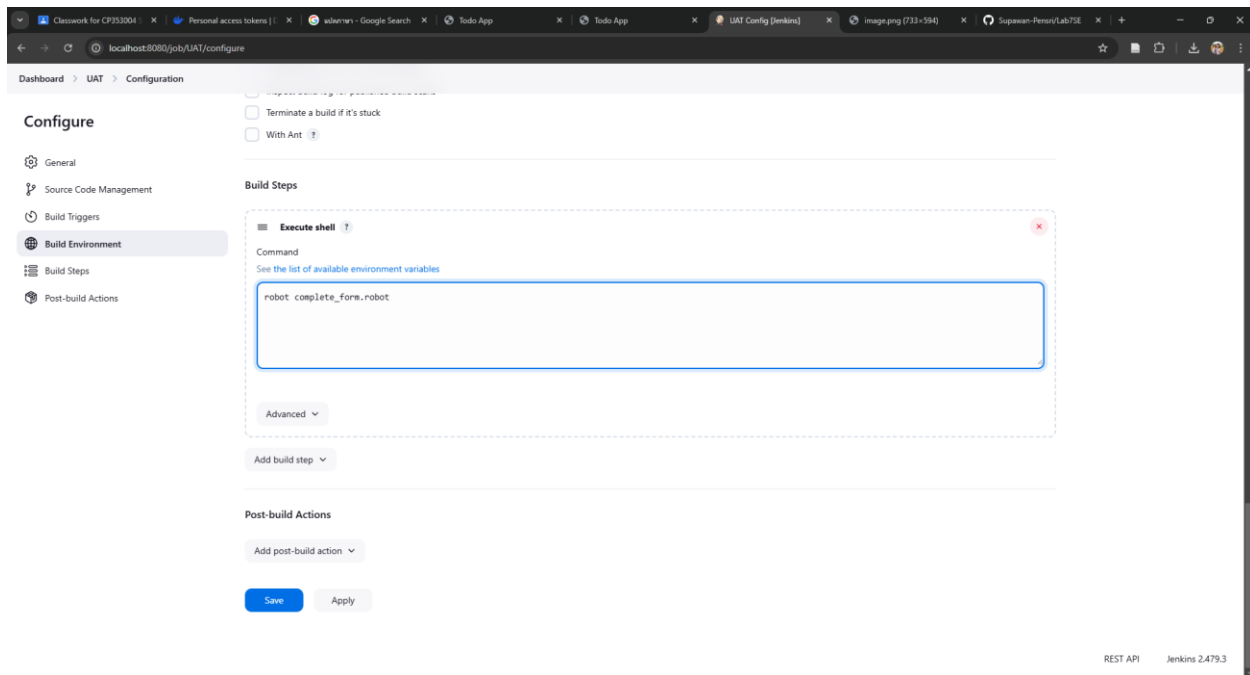
GitHub project: กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

## Lab Worksheet

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

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้



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

ตอบ robot complete\_form.robot

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

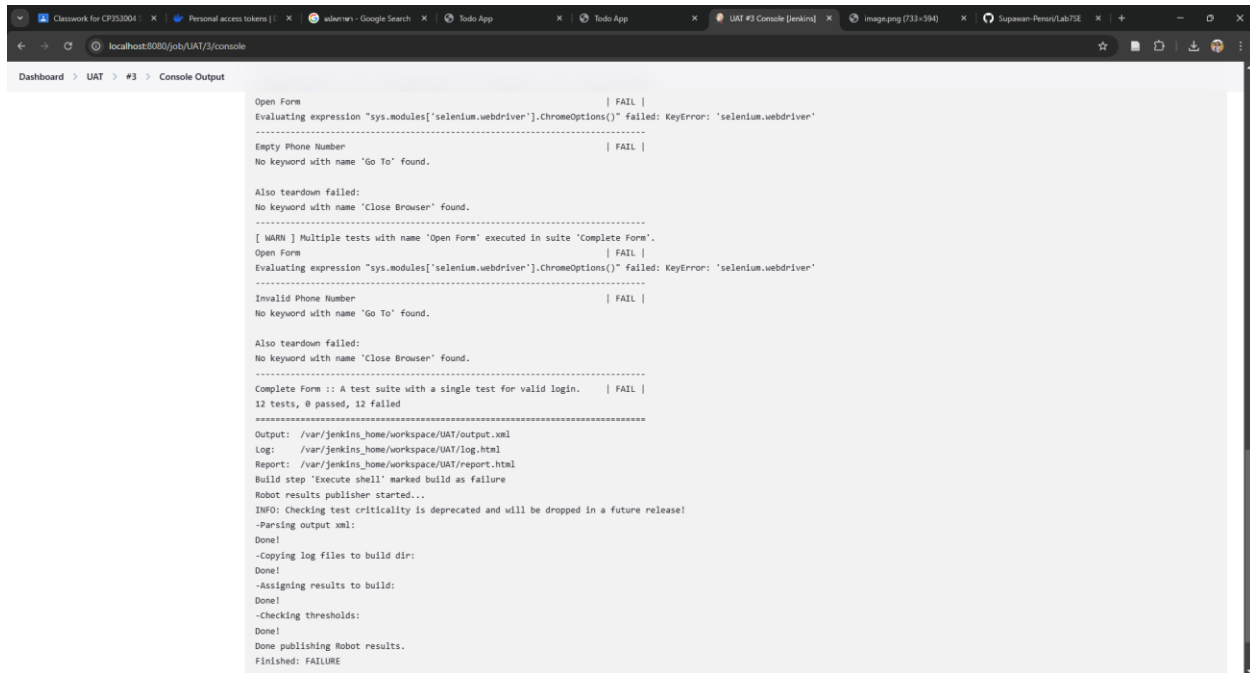
ระบุไดเรกทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

13. กด Apply และ Save

14. สั่ง Build Now

## Lab Worksheet

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



```

Dashboard > UAT > #3 > Console Output

Open Form | FAIL |
Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: KeyError: 'selenium.webdriver'
-----
Empty Phone Number | FAIL |
No keyword with name 'Go To' found.

Also teardown failed:
No keyword with name 'Close Browser' found.
-----
[ WARN ] Multiple tests with name 'Open Form' executed in suite 'Complete Form'.
Open Form | FAIL |
Evaluating expression "sys.modules['selenium.webdriver'].ChromeOptions()" failed: KeyError: 'selenium.webdriver'
-----
Invalid Phone Number | FAIL |
No keyword with name 'Go To' found.

Also teardown failed:
No keyword with name 'Close Browser' found.
-----
Complete Form :: A test suite with a single test for valid login. | FAIL |
12 tests, 0 passed, 12 failed
-----
Output: /var/jenkins_home/workspace/UAT/output.xml
Log: /var/jenkins_home/workspace/UAT/log.html
Report: /var/jenkins_home/workspace/UAT/report.html
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

```