

## Lab Worksheet

ชื่อ-นามสกุล\_ณัฐชา ธิปาบุตร\_รหัสนักศึกษา\_653380196-9\_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>)
4. ป้อนคำสั่ง \$ docker images

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

## Lab Worksheet

```
PS D:\TySoftware\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Image is up to date for busybox:latest
docker.io/library/busybox:latest
```

**What's next:**

View a summary of image vulnerabilities and recommendations → `docker scout quickview busybox`

```
PS D:\TySoftware\Lab8_1> docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
jenkins/jenkins	lts-jdk17	dc56634cc8fa	3 weeks ago	779MB
busybox	latest	a5d0ce49aa80	4 months ago	6.56MB

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร ชื่อของ images
- (2) Tag ที่ใช้บ่งบอกถึงอะไร บอกถึง version ของ images หรือ บอกชื่อได้

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

6.

```
PS D:\TySoftware\Lab8_1> docker run -it busybox sh
/ # ls
```

## Lab Worksheet

7.

```
/ # ls
bin    dev    etc    home   lib    lib64  proc   root   sys    tmp    usr    var
```

8.

```
/ # ls -la
total 48
drwxr-xr-x  1 root   root   4096 Jan 29 15:11 .
drwxr-xr-x  1 root   root   4096 Jan 29 15:11 ..
-rwxr-xr-x  1 root   root    0 Jan 29 15:11 .dockerenv
drwxr-xr-x  2 root   root  12288 Sep 26 21:31 bin
drwxr-xr-x  5 root   root   360 Jan 29 15:11 dev
drwxr-xr-x  1 root   root   4096 Jan 29 15:11 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
```

9.

```
/ # exit
```

10.

```
PS D:\TySoftware\Lab8_1> docker run busybox echo "Hello Natcha Thapabut from busybox"
Hello Natcha Thapabut from busybox
```

11.

```
PS D:\TySoftware\Lab8_1> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
3d8eb80fd413	busybox	"echo 'Hello Natcha ..."	8 seconds ago	Exited (0) 7 seconds ago
d549707d335c	busybox	"sh"	About a minute ago	Exited (0) 54 seconds

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

รันคอนเทนเนอร์ในโหมด Interactive และให้มี Terminal เพื่อป้อนคำสั่ง

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

Status ของ Container ว่าตอนนี้เป็นสถานะอะไร

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

```
PS D:\TySoftware\Lab8_1> docker rm d5
d5
```

## Lab Worksheet

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

```
PS D:\TySoftware\Lab8_1> docker rm d5
d5
```

### 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

## Lab Worksheet

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

```
PS D:\TySoftware\Lab8_2> docker build -t lab8_2 -f Dockerfile.swp .
[+] Building 6.4s (6/6) FINISHED
=> [internal] load build definition from Dockerfile.swp
=> => transferring dockerfile: 154B
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior relate
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stag
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior relate
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
```

```
PS D:\TySoftware\Lab8_2> docker run lab8_2
Natcha Thapabut 653380196-9 Natty
```

- (1) คำสั่งที่ใช้ในการ run คือ `docker run lab8_2`
- (2) Option -t ในคำสั่ง `$ docker build` ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
กำหนดชื่อของ images หรือ tag

### 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"
```

## Lab Worksheet

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 D:\TySoftware\Lab8_3> docker build -t nattynatcha/lab8 -f Dockerfile.swp .
[+] Building 0.4s (5/5) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 175B                             0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior relate 0.0s
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stag 0.0s
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior relate 0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 0.0s
```

```
PS D:\TySoftware\Lab8_3> docker run nattynatcha/lab8
Natcha Thapabut 653380196-9
```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

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

\$ docker login -u &lt;username&gt; -p &lt;password&gt;

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)

```
PS D:\TySoftware\Lab8_3> docker push nattynatcha/lab8
```

```
Using default tag: latest
```

```
The push refers to repository [docker.io/nattynatcha/lab8]
```

```
9c0abc9c5bd3: Mounted from peeraphong/lab8
```

```
6d8948c1a3d8: Pushed
```

```
latest: digest: sha256:ed8ddafdc6a5c5b4506debf1474d22930ed8066e3f2498a69b7406c708623dfe size: 855
```

## nattynatcha/lab8

Last pushed 2 minutes ago • Repository size: 2.1 MB

Add a description

Add a category

General Tags Builds Collaborators Webhooks Settings

## Tags

This repository contains 1 tag(s).

Tag	OS	Type	Pulled	Pushed
latest	linux/amd64	Image	2 minutes ago	2 minutes ago

[See all](#)



## nattynatcha/lab8:latest

INDEX DIGEST sha256:ed8ddafdc6a5c5b4506debf1474d22930ed8066e3f2498a69b7406c708623dfe

OS/ARCH	COMPRESSED SIZE	LAST PUSHED	TYPE	MANIFEST DIGEST
linux/amd64	2.06 MB	2 minutes ago by nattynatcha	Image	sha256:1e86eb1f...

## Image Layers

Vulnerabilities

Image Layers

1	BusyBox 1.37.0 (glibc), Debian 12	2.06 MB	Cor
2	CMD ["/bin/sh" "-c" "echo \"Hi	0 B	
3	CMD ["/bin/sh" "-c" "echo \"Natcha	0 B	

## 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
```

## Lab Worksheet

- 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 D:\TySoftware\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 | 8.18 MiB/s, done.
Resolving deltas: 100% (523/523), done.
```

Name	Date modified	Type	Size
.git	1/29/2025 10:32 PM	File folder	
.github	1/29/2025 10:32 PM	File folder	
app	1/29/2025 10:32 PM	File folder	
docs	1/29/2025 10:32 PM	File folder	
.dockerignore	1/29/2025 10:32 PM	DOCKERIGNORE F...	1 KB
.gitignore	1/29/2025 10:32 PM	Git Ignore Source ...	1 KB
build.sh	1/29/2025 10:32 PM	sh_auto_file	1 KB
docker-compose	1/29/2025 10:32 PM	Yaml Source File	1 KB
Dockerfile	1/29/2025 10:32 PM	File	2 KB
LICENSE	1/29/2025 10:32 PM	File	12 KB
makedocs	1/29/2025 10:32 PM	Yaml Source File	3 KB
README	1/29/2025 10:32 PM	MD File	2 KB
requirements	1/29/2025 10:32 PM	Text Document	1 KB

```
{
  "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. ภายใต้งetting-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 D:\TySoftware\Lab8_4\getting-started\app> new-item Dockerfile
```

```
Directory: D:\TySoftware\Lab8_4\getting-started\app
```

```
Mode                LastWriteTime         Length Name
```

```
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000
|
```

```
PS D:\TySoftware\Lab8_4\getting-started\app> docker build -t myapp_6533801969 .
```

```
[+] Building 6.6s (10/10) FINISHED
```

```
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 156B
=> [internal] load metadata for docker.io/library/node:18-alpine
=> [auth] library/node:pull token for registry-1.docker.io
=> [internal] load .dockerignore
```

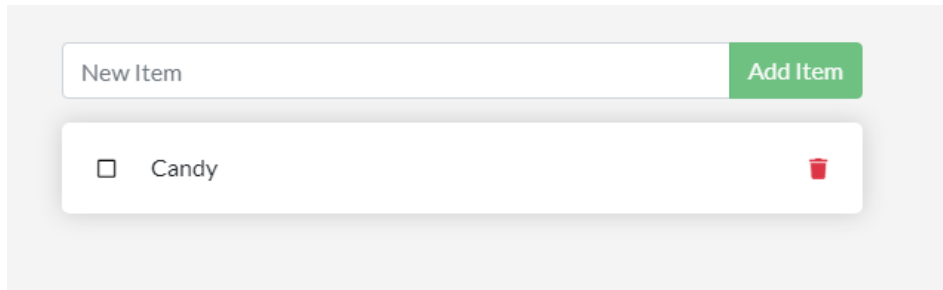
```
docker:desktop-linux
0.0s
0.0s
3.0s
0.0s
0.0s
```

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

```
PS D:\TySoftware\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_65338019696613d44c60faa20614ec7dda681172f1f979677e0aca0ab06ae8d63eb98ec3a0
```



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) แสดงคำสั่งและผลลัพธ์ที่ได้ทาง หน้าจอ พร้อมกับตอบคำถามต่อไปนี้

```
<React.Fragment>|
  <AddItemForm onNewItem={onNewItem} />
  {items.length === 0 && (
    <p className="text-center">There is no TODO item. Please add one to the
list. By Natcha Thapabut</p>
  )}
  {items.map(item => (
```

## Lab Worksheet

```

PS D:\TySoftware\Lab8_4\getting-started\app> docker build -t myapp_6533801969 .
[+] Building 41.7s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 156B                                              0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine                 1.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059d 0.0s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059d 0.0s
=> [internal] load build context                                                  0.0s

```

```

PS D:\TySoftware\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533801969
16a6a036cefc8c76b5885f316c3ea210242e66f6781403946d2ccf55bf569ea
docker: Error response from daemon: driver failed programming external connectivity on endpoint heuristic_ardighelli (d2ebf3b67b057eabc958de738788b27c6861cedd584ece652f49c0ce004e63e1): Bind for 0.0.0.0:3000 failed: port is already allocated.

```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

map port ไม่ได้ port ถูกใช้ไปแล้ว เกิดจากการ run container บน port เดียวกันซ้ำจากตัวที่ run ไปก่อนหน้า

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

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

b. ผ่าน Docker desktop

- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

## Lab Worksheet

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

```
PS D:\TySoftware\Lab8_4\getting-started\app> docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS
6613d44c60fa   600025f26534                        "docker-entrypoint.s... 6 minutes ago  Up 6 minutes  0.0.0.0:3000->3000/tcp
143a0899833e   jenkins/jenkins:lts-jdk17          "/usr/bin/tini -- /u... 28 hours ago   Up 38 minutes  0.0.0.0:8080->8080/tcp
PS D:\TySoftware\Lab8_4\getting-started\app> docker stop 661
661
PS D:\TySoftware\Lab8_4\getting-started\app> docker rm 661
661
PS D:\TySoftware\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533801969ef4e582d51ecf98b1c547ed691501e5f7632a44421655b981c062cea960868c7
```

New Item

Add Item

There is no TODO item. Please add one to the list. By Natcha Thapabut

### 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

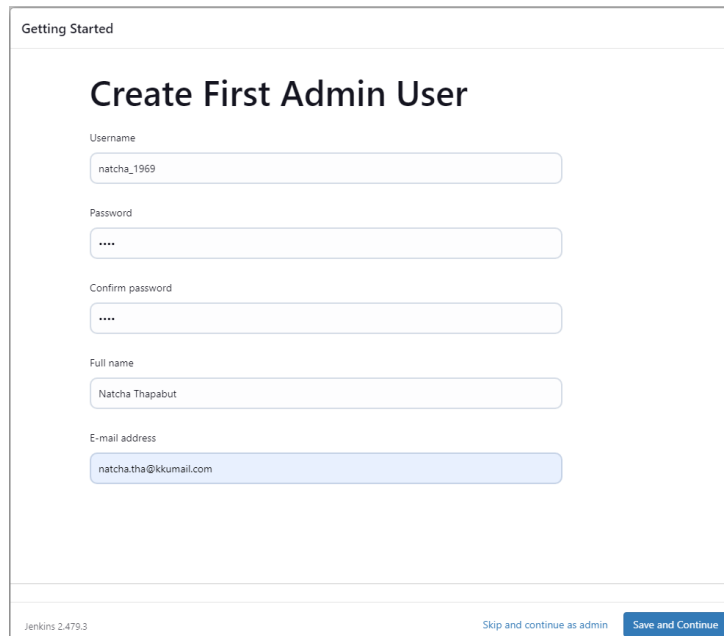
```
-----
eaeaa1143f63412c8ef5424806ee9590
```

4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080

## Lab Worksheet

5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062

[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

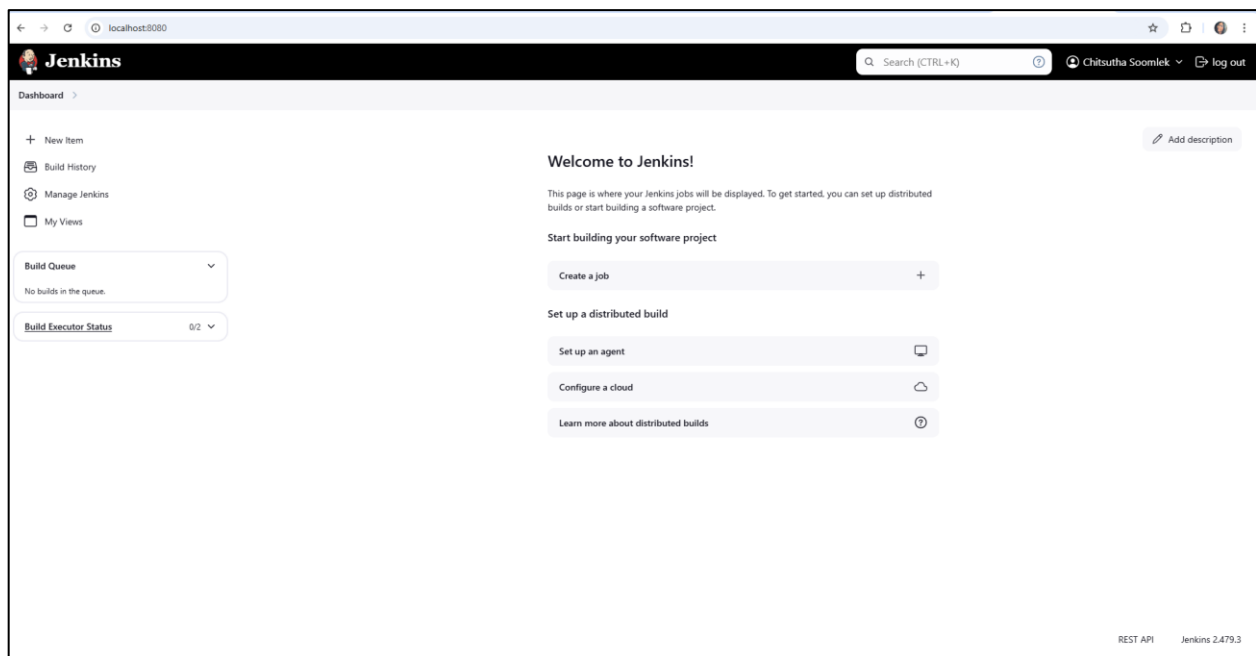


The screenshot shows the 'Getting Started' page in Jenkins with the 'Create First Admin User' form. The form fields are filled with the following information:

- Username: natcha\_1969
- Password: \*\*\*\*
- Confirm password: \*\*\*\*
- Full name: Natcha Thapabut
- E-mail address: natchatha@kkumail.com

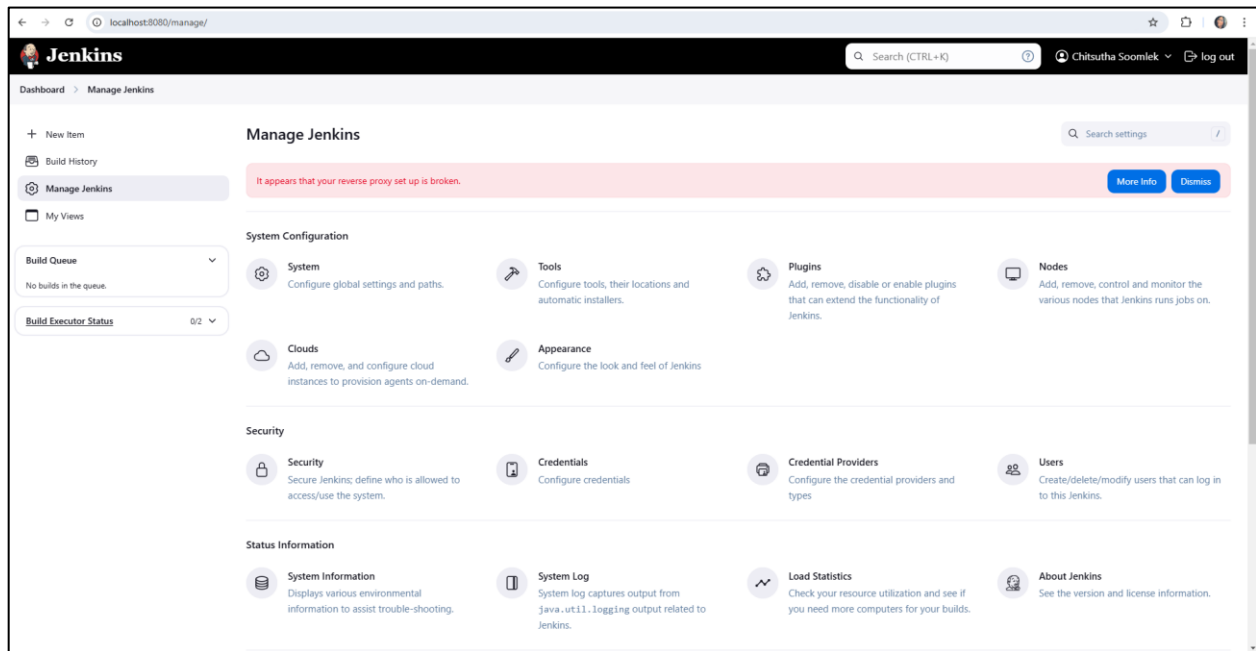
At the bottom of the form, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'.

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

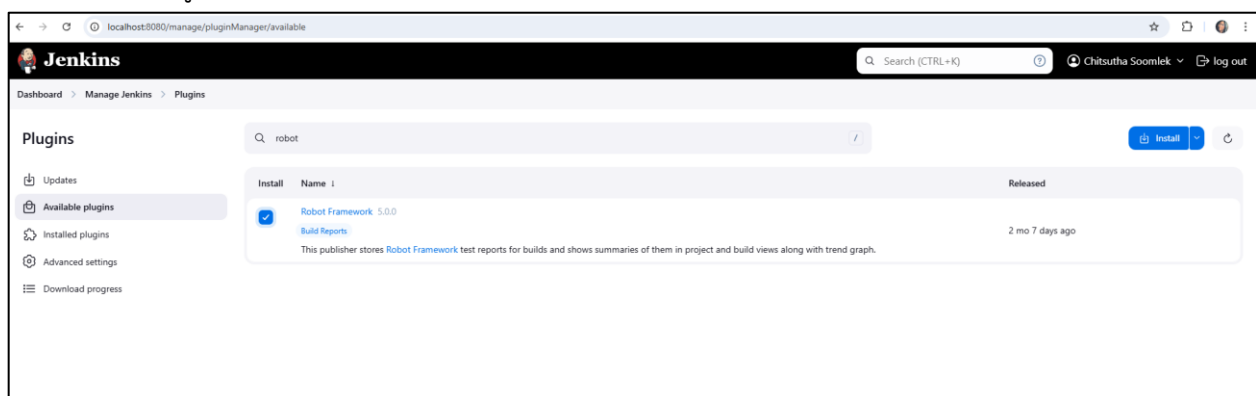


## Lab Worksheet

## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

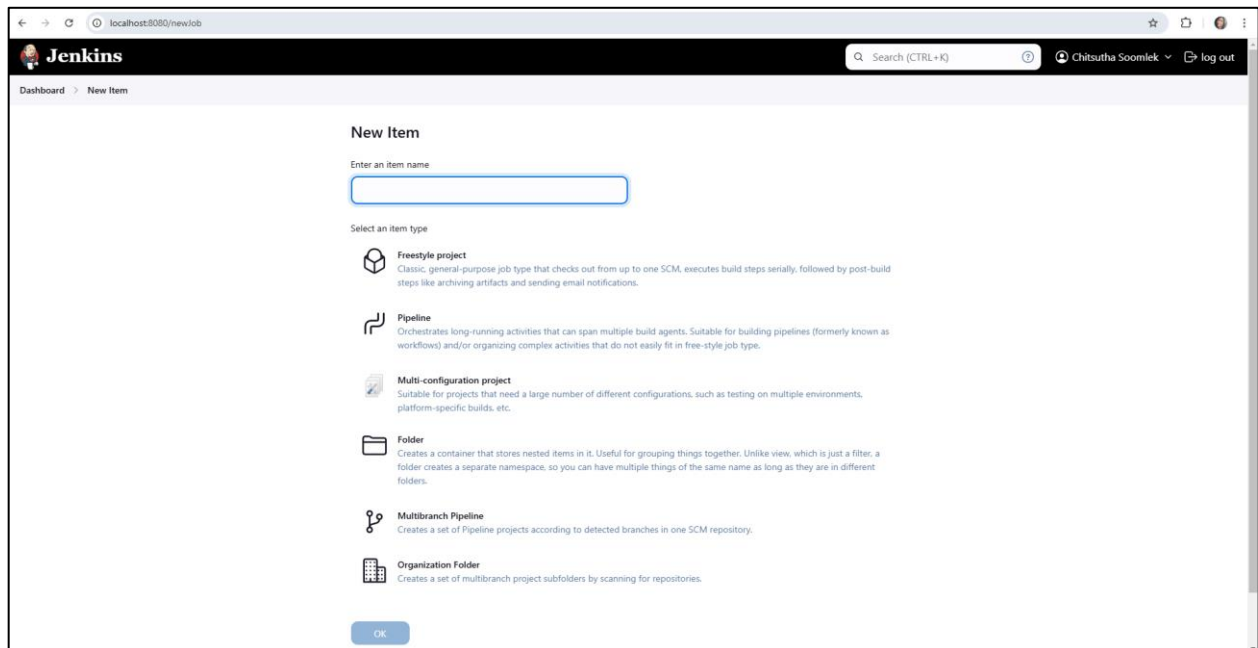


## 10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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

## 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

Description

Lab 8.5

Plain text [Preview](#)☐ Discard old builds [?](#)☒ GitHub projectProject url [?](#)<https://github.com/NatchaThapabut/Lab8.git>Advanced [v](#)☐ This project is parameterized [?](#)☐ Throttle builds [?](#)☐ Execute concurrent builds if necessary [?](#)Advanced [v](#)

## Source Code Management

☐ None☒ Git [?](#)Repositories [?](#)Repository URL [?](#)<https://github.com/NatchaThapabut/Lab8.git>Credentials [?](#)

- none -

+ Add

Advanced [v](#)

Add Repository

Branches to build [?](#)Branch Specifier (blank for 'any') [?](#)

\*/main

Add Branch



## Lab Worksheet

Adds links in "changes" views within Jenkins to an external system for browsing the details of those changes. The "Auto" selection attempts to infer the repository browser from other jobs, if supported by the SCM and a job with matching SCM details can be found.

(Auto) ▼

Additional Behaviours

Add ▼

## 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 3:50:38 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 4:05:38 PM Coordinated Universal Time.

- ☐ GitHub hook trigger for GITScm polling ?
- ☐ Poll SCM ?

## Build Environment

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

## Build Steps

## Execute shell ?

Command

See [the list of available environment variables](#)

```
robot UAT_Lab7_001.robot
robot UAT_Lab7_002.robot
```

Advanced ▼

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

robot UAT\_Lab7\_001.robot

robot UAT\_Lab7\_002.robot

## Lab Worksheet

**Post-build action:** เพิ่ม Publish Robot Framework test results -> ระบุได้เร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่าน แล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (เช่น 20, 80)

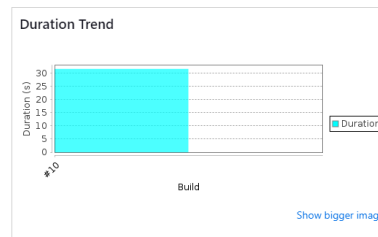
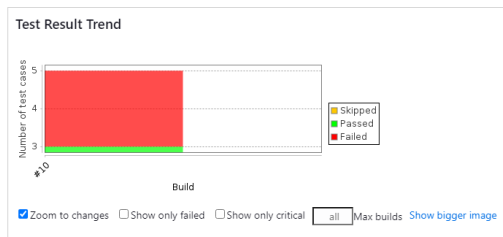
13. กด Apply และ Save

14. สั่ง Build Now

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

## Robot Framework Test Results

**Executed:** 2025-01-29T23:33:15.810962  
**Duration:** 0:00:31.615 (+0:00:31.615)  
**Status:** 5 critical test, 3 passed, **2** failed, 0 skipped  
 5 test total (±0), 3 passed, **2** failed, 0 skipped  
**Results:** [report.html](#)  
[log.html](#)  
[Original result files](#)



## Failed Test Cases

Name	Crit.	Duration	Age
⊞ UAT-Lab7-002.Validate Destination Field	no	0:00:04.879	1
⊞ UAT-Lab7-002.Validate Invalid Phone Field	no	0:00:04.884	1

## Lab Worksheet

```

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/NatchaThapabut/Lab8.git # timeout=10
Fetching upstream changes from https://github.com/NatchaThapabut/Lab8.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/NatchaThapabut/Lab8.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 35b31de1920971db3fab95dedd73c8157a536b63 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 35b31de1920971db3fab95dedd73c8157a536b63 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk 35b31de1920971db3fab95dedd73c8157a536b63 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins5482853714816246632.sh
+ robot UAT_Lab7_001.robot
/tmp/jenkins5482853714816246632.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:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE

```

 UAT

 Edit description

Lab 8.5



Latest Robot Results:

Total	Failed	Passed	Skipped	Pass %	
All tests	5	2	3	0	60.0

- [Browse results](#)
- [Open report.html](#)
- [Open log.html](#)

## Permalinks

- [Last build \(#11\). 38 sec ago](#)
- [Last failed build \(#11\). 38 sec ago](#)
- [Last unsuccessful build \(#11\). 38 sec ago](#)
- [Last completed build \(#11\). 38 sec ago](#)

Robot Framework Tests Trend (all tests)

