

Lab Worksheet

ชื่อ-นามสกุล นายภูวนาท โคตรชนะ รหัสนักศึกษา 653380340-8 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

Lab Worksheet

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

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

<input type="checkbox"/>	Name	Tag	Image ID ↑	Created	Size
<input type="checkbox"/>	busybox	latest	a5d0ce49aa80	4 months ago	6.55

Terminal

```
PS C:\work\Software_years3\Lab8_1> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
PS C:\work\Software_years3\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Download complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest
PS C:\work\Software_years3\Lab8_1> docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
busybox latest a5d0ce49aa80 4 months ago 6.56MB
PS C:\work\Software_years3\Lab8_1> ^C
PS C:\work\Software_years3\Lab8_1> []
```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร busybox
- (2) Tag ที่ใช้บ่งบอกถึงอะไร ระบุสถานะของ docker images
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 Poowanath Kotchana from busybox"
11. ป้อนคำสั่ง \$ docker ps -a

Lab Worksheet

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

<input type="checkbox"/>	Name	Tag	Image ID	Created	Size
<input type="checkbox"/>	busybox	latest	a5d0ce49aa80	4 months ago	6

Terminal

```

busybox latest a5d0ce49aa80 4 months ago 6.56MB
PS C:\work\Software_years3\Lab8_1> ^C
PS C:\work\Software_years3\Lab8_1> ^C
drwxr-xr-x 1 root root 4096 Jan 27 13:33 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 267 root root 0 Jan 27 13:33 proc
drwx----- 1 root root 4096 Jan 27 13:33 root
dr-xr-xr-x 11 root root 0 Jan 27 13:33 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:\work\Software_years3\Lab8_1> docker run busybox echo Hello Poowanath Kotchana from busybox
Hello Poowanath Kotchana from busybox
PS C:\work\Software_years3\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS              PORTS          NAMES
d1566a64f46f   busybox   "echo Hello Poowanath..." 7 seconds ago   Exited (0) 6 seconds ago           priceless_booth
e499052c5e6c   busybox   "sh"                     About a minute ago   Exited (0) 36 seconds ago         pedantic_einstein
09ff2666450d   busybox   "sh"                     About a minute ago   Exited (0) About a minute ago         sleepy_pare
PS C:\work\Software_years3\Lab8_1>

```

- (1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
ส่งผลให้เราสามารถโต้ตอบกับคอนเทนเนอร์ได้โดยตรงผ่าน terminal
- (2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร
แสดงถึงสถานะของการทำงานว่าเกิดอะไรขึ้น

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

Lab Worksheet

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

☰
Only show running containers

<input type="checkbox"/>	Name	Container ID	Image	Port(s)	CPU (%)	Last started
<input type="checkbox"/>	sleepy_pare	09ff2666450d	busybox		N/A	9 minutes ago
<input type="checkbox"/>	pedantic_einstein	e499052c5e6c	busybox		N/A	9 minutes ago
<input type="checkbox"/>	priceless_booth	d1566a64f46f	busybox		N/A	8 minutes ago
<input type="checkbox"/>	objective_taussig	93fca0f30b3	busybox		N/A	6 minutes ago

Terminal

```

+ option -lt
+ CategoryInfo          : ObjectNotFound: (option:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\work\Software_years3\Lab8_1> docker run busybox
PS C:\work\Software_years3\Lab8_1> docker run -it busybox sh
/ # option -lt
sh: option: not found
/ # ^C

/ # exit
PS C:\work\Software_years3\Lab8_1> docker rm 8d9d8e41bc18bb8f2ee4a5ea577051be45e5cbf6a8df446daa23198bdccff868
8d9d8e41bc18bb8f2ee4a5ea577051be45e5cbf6a8df446daa23198bdccff868
PS C:\work\Software_years3\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"

EOF

Lab Worksheet

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

```
=> exporting to image
=> exporting layers
=> exporting manifest sha256:92a82d2eaca61e6c097434e909e644209dc9ce8062c91dea614036c37c8d4b5
=> exporting config sha256:b965c9ba7c320c5abaa28ffc44676f00b98980e97a25d1400ebb5474a533fae3
=> exporting attestation manifest sha256:95574ad09b555be722f61aed3c34cc0180974db71ca7fc3f29d88705351ee6a
=> exporting manifest list sha256:22077a38e23b1e019cdd8ae96cc452f45aa64347012e42d5ec0c3cb4c3f92d0
=> naming to docker.io/library/myinfo:latest
=> unpacking to docker.io/library/myinfo:latest

0.1s
0.0s
0.0s
0.0s
0.0s
0.0s
0.0s
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 3)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 3)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 5)
PS C:\work\Software_years3\Lab8_2> docker run myinfo
"Poowanath Kotchana 65380340-8 Pat"
PS C:\work\Software_years3\Lab8_2> docker run myinfo
```

- (1) คำสั่งที่ใช้ในการ run คือ

docker run myinfo

- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
-t ในคำสั่งนี้มีหน้ากำหนดชื่อ หรือกำหนด tag ให้กับ docker image

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"

Lab Worksheet

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
```

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

The screenshot shows the Docker Desktop interface. At the top, there is a search bar and icons for settings and resources. Below this is a table listing Docker images:

Name	Tag	Image ID	Created	Size
busybox	latest	a5d0ce49aa80	4 months ago	6.55 MB
myinfo	latest	22077a38e23b	4 months ago	6.55 MB
poowanath/lab8	latest	429db725b3c6	4 months ago	6.55 MB

Below the table is a terminal window titled "Terminal" showing the output of the Docker build and run commands. The build process is successful, and the container is running. The terminal output includes:

```
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
PS C:\work\Software_years3\Lab8_3> docker build -t poowanath/lab8 .
[+] Building 2.3s (5/6) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 186B
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 3)
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 4)
=> [internal] load metadata for docker.io/library/busybox:latest
=> => exporting config sha256:73679228ee0473185501880c3363e8b5cfe439843b0fae0b09a8535b08df133e
=> => exporting attestation manifest sha256:2a87adb882d225758ffdc86741839c2b9bbf876f60c4e314cde3229d4b4a538
=> => exporting manifest list sha256:429db725b3c632bf1f8821c59db3eaddc48b4459a565608453dc8f05ca884a8e
=> => naming to docker.io/poowanath/lab8:latest
=> => unpacking to docker.io/poowanath/lab8:latest

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 3)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 4)
PS C:\work\Software_years3\Lab8_3> docker run poowanath/lab8
"Poowanath Kotchana 653388348-8."
```

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

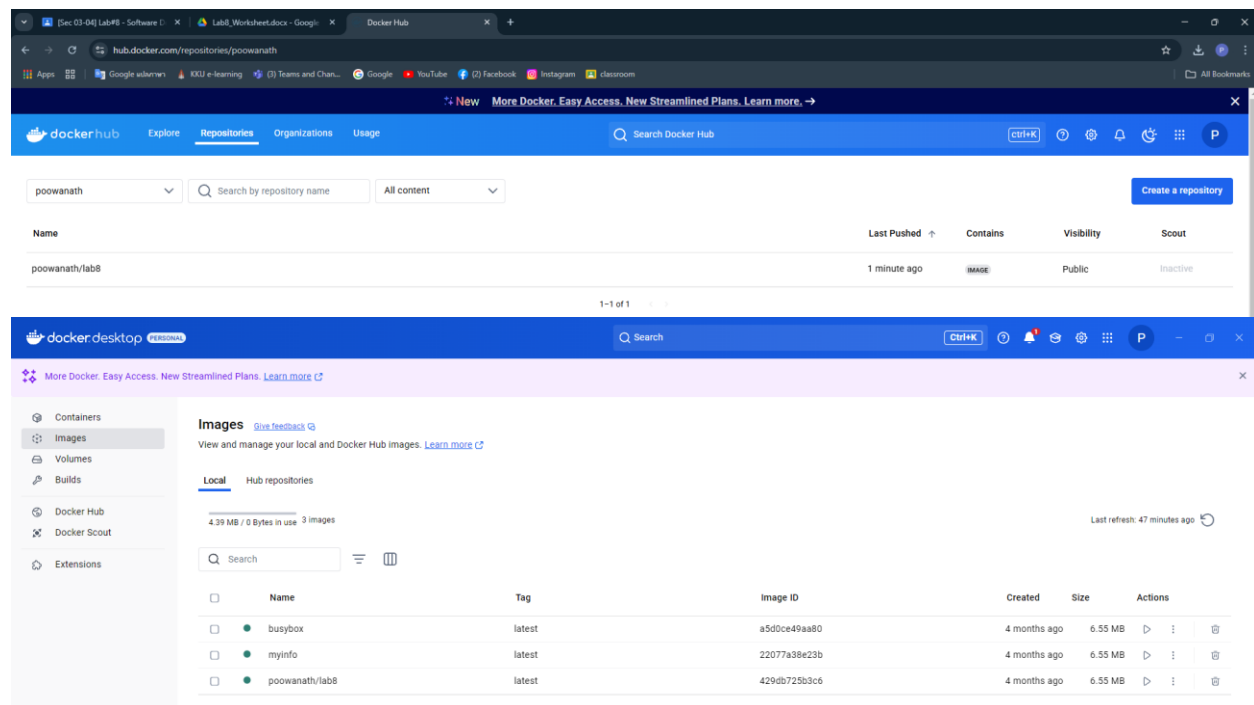
Lab Worksheet

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

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

- 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

- เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4
- 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
- เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

Lab Worksheet

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

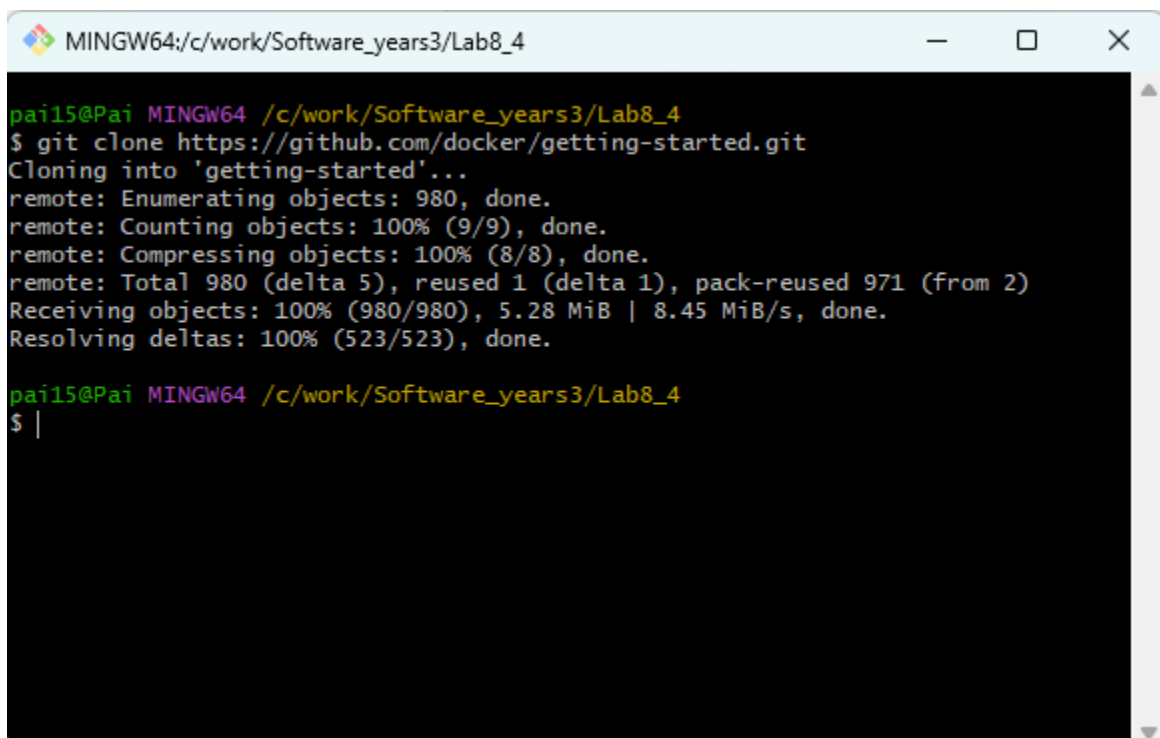
Terminal

```
PS C:\work\Software_years3\Lab8_3> cd ..  
PS C:\work\Software_years3> cd .\Lab8_4\  
PS C:\work\Software_years3\Lab8_4> ls
```

Directory: C:\work\Software_years3\Lab8_4

Mode	LastWriteTime	Length	Name
d-----	1/27/2025 9:10 PM		getting-started

```
PS C:\work\Software_years3\Lab8_4>
```



The screenshot shows a Windows terminal window titled "MINGW64:/c/work/Software_years3/Lab8_4". The user is logged in as "pai15@Pai". The terminal shows the execution of the command `git clone https://github.com/docker/getting-started.git`. The output of the command is as follows:

```
pai15@Pai MINGW64 /c/work/Software_years3/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 | 8.45 MiB/s, done.  
Resolving deltas: 100% (523/523), done.  
  
pai15@Pai MINGW64 /c/work/Software_years3/Lab8_4  
$ |
```


Lab Worksheet



```
{
  "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"
  }
}
```

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
```

## Lab Worksheet

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

The screenshot shows a file explorer window with the following files and folders:

| Name       | Date modified     | Type             | Size   |
|------------|-------------------|------------------|--------|
| spec       | 1/27/2025 9:10 PM | File folder      |        |
| src        | 1/27/2025 9:10 PM | File folder      |        |
| package    | 1/27/2025 9:10 PM | JSON Source File | 1 KB   |
| yarn.lock  | 1/27/2025 9:10 PM | LOCK File        | 148 KB |
| Dockerfile | 1/27/2025 9:17 PM | File             | 1 KB   |

Below the file explorer is a terminal window showing the Docker build process:

```
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker build -t myapp_6533803408 .
[+] Building 23.4s (18/18) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 171B
=> [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@sha256:974af6bcb0314dc6502b14243b8a39fbb2d04d975e959dd866be3e274fbb25
=> resolve docker.io/library/node:18-alpine@sha256:974af6bcb0314dc6502b14243b8a39fbb2d04d975e959dd866be3e274fbb25
=> sha256:6504e29690c8d5213b52cda800370abb3d12639802d06b40b6fce368990ca771_444B / 444B
=> sha256:37892ffbfca871a10f813883949d18c3015a482051d51b7e0da02525e63167c_40.01MB / 40.01MB
=> sha256:5650d6de56f90bb419872b876ac1df28f577b39573c3b72fbd0d15b426d01bc1_1.26MB / 1.26MB
=> sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0_3.64MB / 3.64MB
=> extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0
=> extracting sha256:37892ffbfca871a10f813883949d18c3015a482051d51b7e0da02525e63167c
=> extracting sha256:5650d6de56f90bb419872b876ac1df28f577b39573c3b72fbd0d15b426d01bc1
=> extracting sha256:6504e29690c8d5213b52cda800370abb3d12639802d06b40b6fce368990ca771
=> [internal] load build context
=> => transferring context: 4.62MB
=> [2/4] WORKDIR /app
```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

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

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

## Lab Worksheet

|                          |  |                  |        |              |               |         |   |
|--------------------------|--|------------------|--------|--------------|---------------|---------|---|
| <input type="checkbox"/> |  | busybox          | latest | a5d0ce49aa80 | 4 months ago  | 6.55 MB | ▶ |
| <input type="checkbox"/> |  | myinfo           | latest | 22077a38e23b | 4 months ago  | 6.55 MB | ▶ |
| <input type="checkbox"/> |  | poowanath/lab8   | latest | 429db725b3c6 | 4 months ago  | 6.55 MB | ▶ |
| <input type="checkbox"/> |  | myapp_6533803408 | latest | 43778cba00e2 | 8 minutes ago | 342 MB  | ▶ |

Walkthroughs

Terminal

```

=> sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded51476c24483d592134f8 3.64MB / 3.64MB 2.1s
=> extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded51476c24483d592134f8 0.1s
=> extracting sha256:378927fbfca871a18f013863049d18c3815a482851d51b7e6da82525e63157c 0.7s
=> extracting sha256:56504d6e56f8bb419872b876ac1df28f577b39573c3b72fb6d15bf426d81bc1 0.6s
=> extracting sha256:6504e29608cd5213b52cda808370abb3d12639882d86b46b6fce368990ca771 0.6s
[internal] load build context
=> transferring context: 4.62MB 0.4s
[2/4] WORKDIR /app 0.2s
[3/4] COPY . . 0.1s
[4/4] RUN yarn install --production 9.6s
=> exporting to image 4.0s
=> exporting layers 2.7s
=> exporting manifest sha256:ade814aab9a7bf40ec9d99b018d652583d7362a64ecb813bd584692720119c9 0.0s
=> exporting config sha256:66093fe93cf7317c606660cc14a3b5e6950bec06e90e2027f5adb472001c99 0.0s
=> exporting attestation manifest sha256:735d729704092ba197cf8cc5c061a620b36d4c70cdd44b79da11017afa14777 0.0s
=> exporting manifest list sha256:43778cba00e237580a07ba411e44cdda90744dc2ed08fe56d47997efe365ccb0 0.0s
=> naming to docker.io/library/myapp_6533803408:latest 0.0s
=> unpacking to docker.io/library/myapp_6533803408:latest 1.1s
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803408
9486e1d3b80d26c97215c329864ca7ae4e39b37e708f1982890dc4c25635eeab
PS C:\work\Software_years3\Lab8_4\getting-started\app>

```

New Item

Add Item

No items yet! Add one above!

Docker Hub

Todo App

+

Google YouTube (2) Facebook Instagram classroom

New Item

Add Item

☐ Carrot

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

## 8. ทำการแก้ไข Source code ของ Web application ดังนี้

## Lab Worksheet

- 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

```

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 Poowanath Kotchana</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);
68 }

```

```

=> => transferring context: 28
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afbcbcb0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> resolve docker.io/library/node:18-alpine@sha256:974afbcbcb0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> [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
=> exporting manifest sha256:e19cc1ce710026f1bc50ba3d839bb2dcccabbdd290f08c64891ef3b56786bfff
=> exporting config sha256:528458798979029989c82308364039a07ac9c3df38dc1fe1a4ec2c2c7260
=> exporting attestation manifest sha256:cf262ecae9417c849ab8690847389c8f15d4da372232935c5691f78f044a11
=> exporting manifest list sha256:43195a3b17e7f8c112d7347c216b747f5ead9024446c2c3705718400a91f8ca
=> naming to docker.io/library/myapp_6533803408:latest
=> unpacking to docker.io/library/myapp_6533803408:latest
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803408
5fad5657f40e6cd505775e42e622cd613c4a632613e02a649148b3579751a6f
docker: Error response from daemon: driver failed programming external connectivity on endpoint trusting_bartik (die21ica04836590bcaa87962d4d3f214403a6950615b48b66d735b44aaa664): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS C:\work\Software_years3\Lab8_4\getting-started\app>

```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

หมายความว่าใช้งาน port ซ้ำกัน เกิดจาก port 3000 ถูกใช้งานไปแล้ว

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

## Lab Worksheet

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) แสดงผลลัพธ์ที่ได้นบน Browser และ Dashboard ของ Docker desktop

The screenshot shows the Docker Desktop interface. At the top, there's a search bar and icons for containers and images. Below is a table of containers:

Name	Tag	Image ID	Created	Size	Actions
busybox	latest	a5d0ce49aa80	4 months ago	6.55 MB	[Stop] [Refresh] [Delete]
myinfo	latest	22077a38e23b	4 months ago	6.55 MB	[Stop] [Refresh] [Delete]
poowanath/lab8	latest	429db725b3c6	4 months ago	6.55 MB	[Stop] [Refresh] [Delete]
myapp_6533803408	latest	43195a3b17e7	6 minutes ago	342 MB	[Stop] [Refresh] [Delete]

Below the table, there's a 'Walkthroughs' section. A terminal window is open, showing the following commands and output:

```
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
9486e1d3b08d 43778cba00e2 "docker-entrypoint.s..." 11 minutes ago Up 11 minutes 0.0.0.0:3000->3000/tcp vigilant_satoshi
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker stop 9486e1d3b08d
9486e1d3b08d
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker rm 9486e1d3b08d
9486e1d3b08d
PS C:\work\Software_years3\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803408
bccf52998577a91fe9629938fadfdde198c3030e2c5b58e1d24798be3d51c49e
PS C:\work\Software_years3\Lab8_4\getting-started\app>
```

Below the terminal, there's a browser window showing the 'localhost:3000' address. The browser displays a 'New Item' input field and an 'Add Item' button. Below this, it says: 'There is no TODO item. Please add one to the list. By Poowanath Kotchana'.

## แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop

## Lab Worksheet

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

<input type="checkbox"/>	laughing_curie	bccf52998577	myapp_6533803408	3000:3000	0.19%	12 minutes ago
<input type="checkbox"/>	condescending_meninsky	61c18df650f7	jenkins/jenkins:lts-jdk17	50000:50000 Show all ports (2)	0.09%	1 minute ago

Terminal

```


Jenkins initial setup is required. An admin user has been created and a password generated.
Please use the following password to proceed to installation:

79f6908ab72448e8965ad8f90c1b47c9

This may also be found at: /var/jenkins_home/secrets/initialAdminPassword


```

🔍

📄

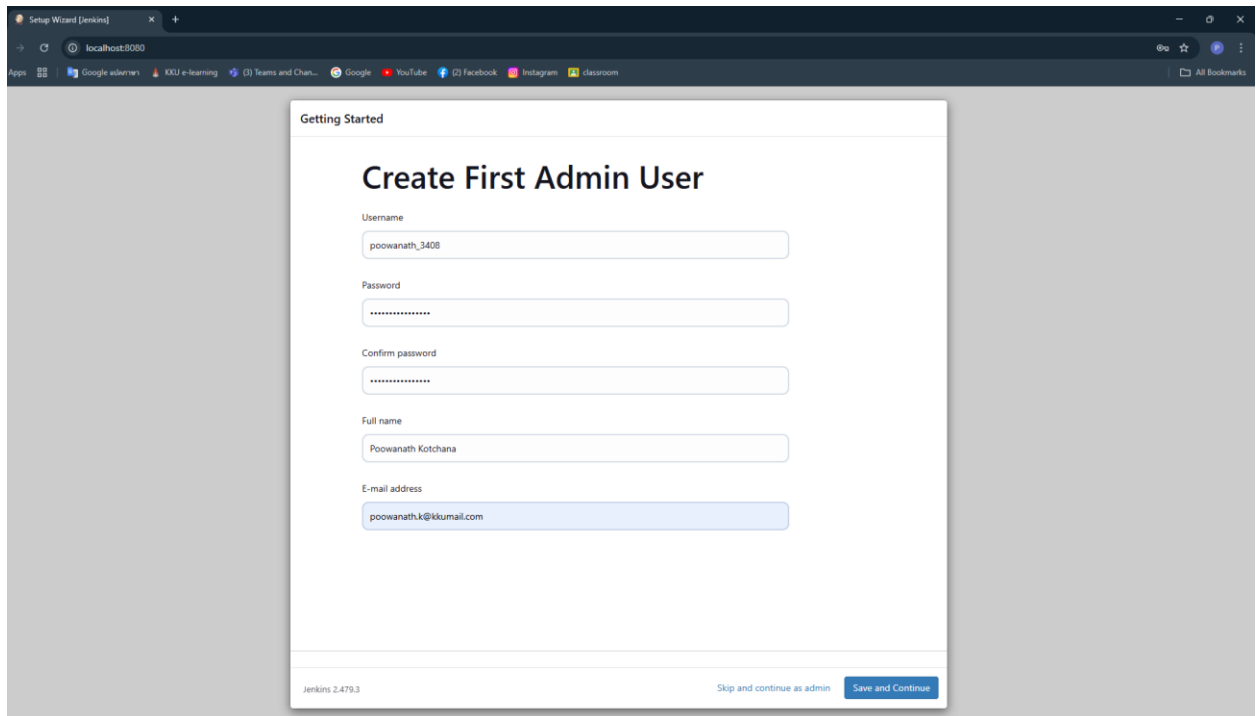
🗑️

⬇️

4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062

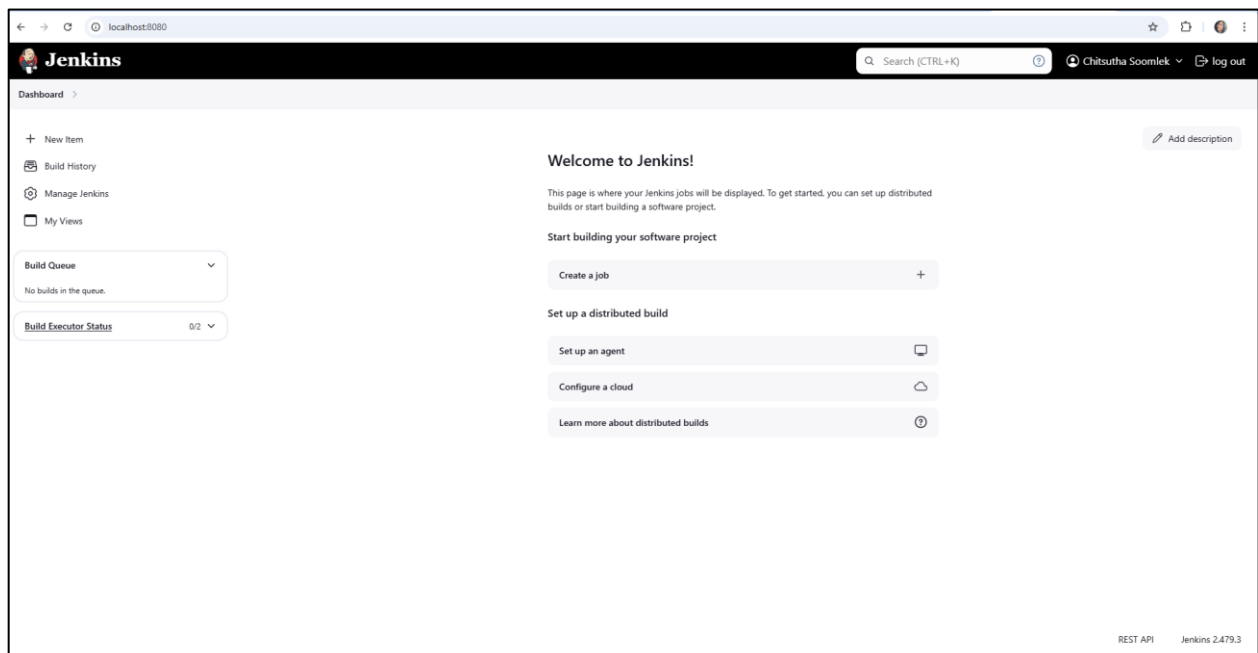
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

## Lab Worksheet



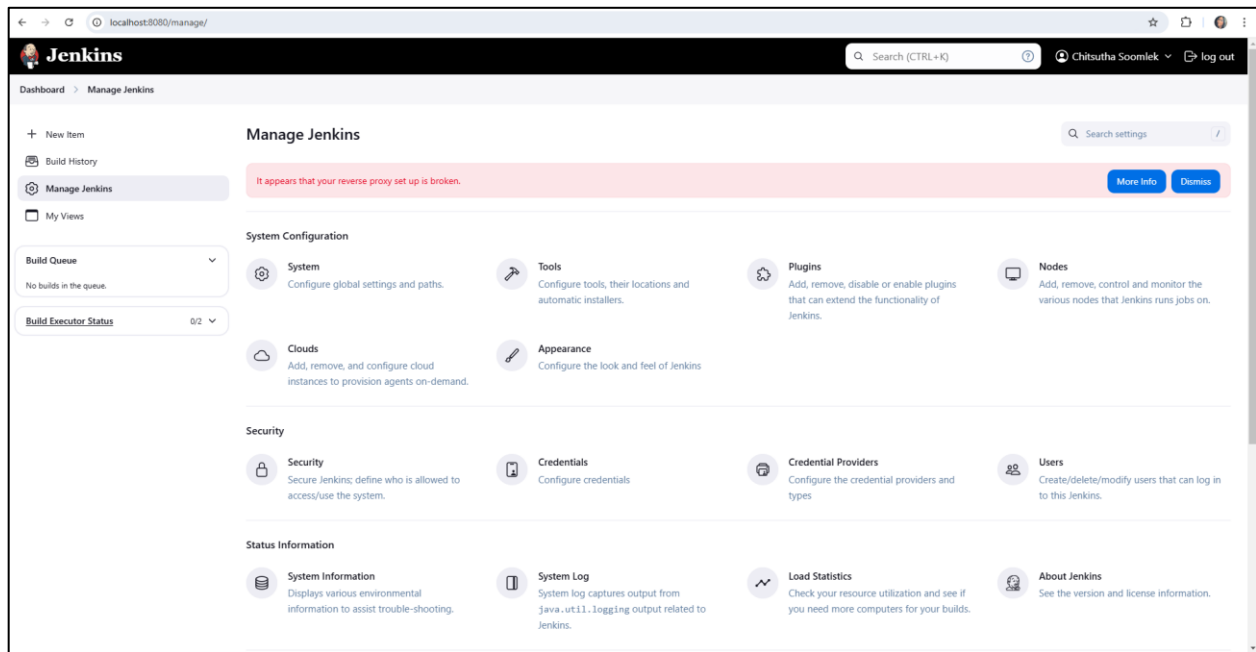
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

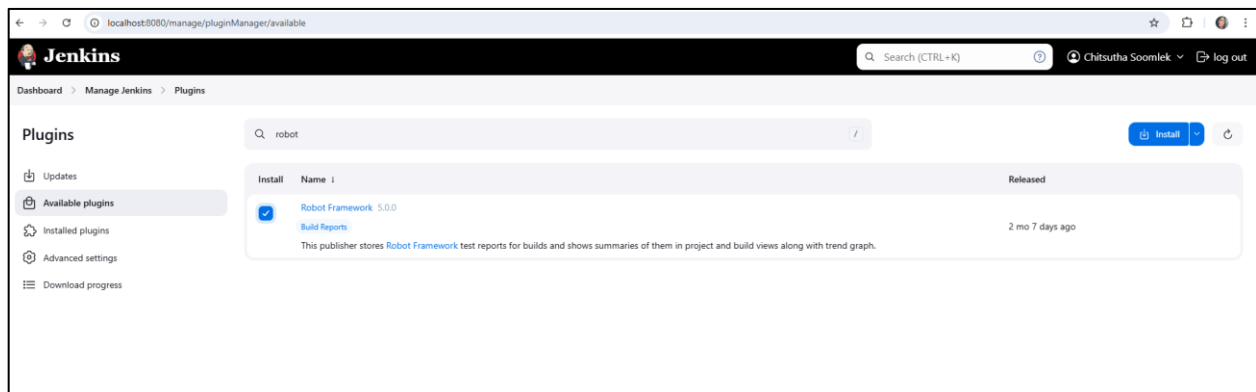


9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

## Lab Worksheet



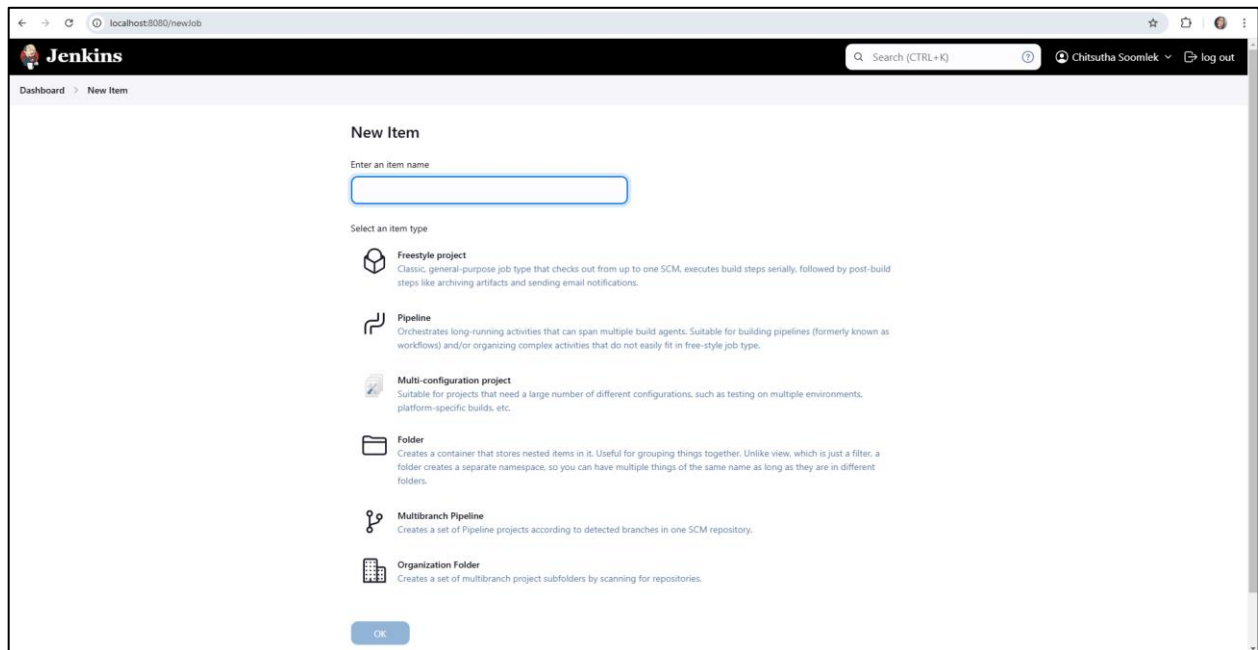
10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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 image shows two screenshots of the Jenkins web interface. The top screenshot is the 'Getting Started' page for 'Instance Configuration'. It features a text input field for 'Jenkins URL' containing 'http://localhost:8080/lab8'. Below the input field, there is explanatory text about the Jenkins URL and a 'Save and Finish' button. The bottom screenshot shows the 'Installed plugins' page. A search bar at the top contains the text 'robot'. Below the search bar, a table lists installed plugins. The first entry is 'Robot Framework plugin' version 5.0.0, which is marked as 'Enabled' with a blue toggle switch. The table also includes links for 'Report an issue with this plugin' and a description of the plugin's functionality.

**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

---

**Installed plugins - Plugins [jen]**

Search (CTRL+K)  log out

Dashboard > Manage Jenkins > Plugins

### Plugins

- Updates
- Available plugins
- Installed plugins**
- Advanced settings
- Download progress

Name	Enabled
<b>Robot Framework plugin</b> 5.0.0 This publisher stores <b>Robot Framework</b> test reports for builds and shows summaries of them in project and build views along with trend graph. <a href="#">Report an issue with this plugin</a>	<input checked="" type="checkbox"/>

## Lab Worksheet

Dashboard > UAT > Configuration

### Configure

**General**

Description: Lab8.5

Plain text [Preview](#)

☐ Discard old builds ?

☒ GitHub project

Project url ?

[Advanced](#) ▾

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

Would last have run at Tuesday, January 28, 2025 at 1:35:24 PM Coordinated Universal Time; would next run at Tuesday, January 28, 2025 at 1:35:24 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

### Build Steps

**Execute shell** ?

Command

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

[Advanced](#) ▾

- (1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ
- robot UAT\_Lab7\_001.robot UAT\_Lab7\_002.robot

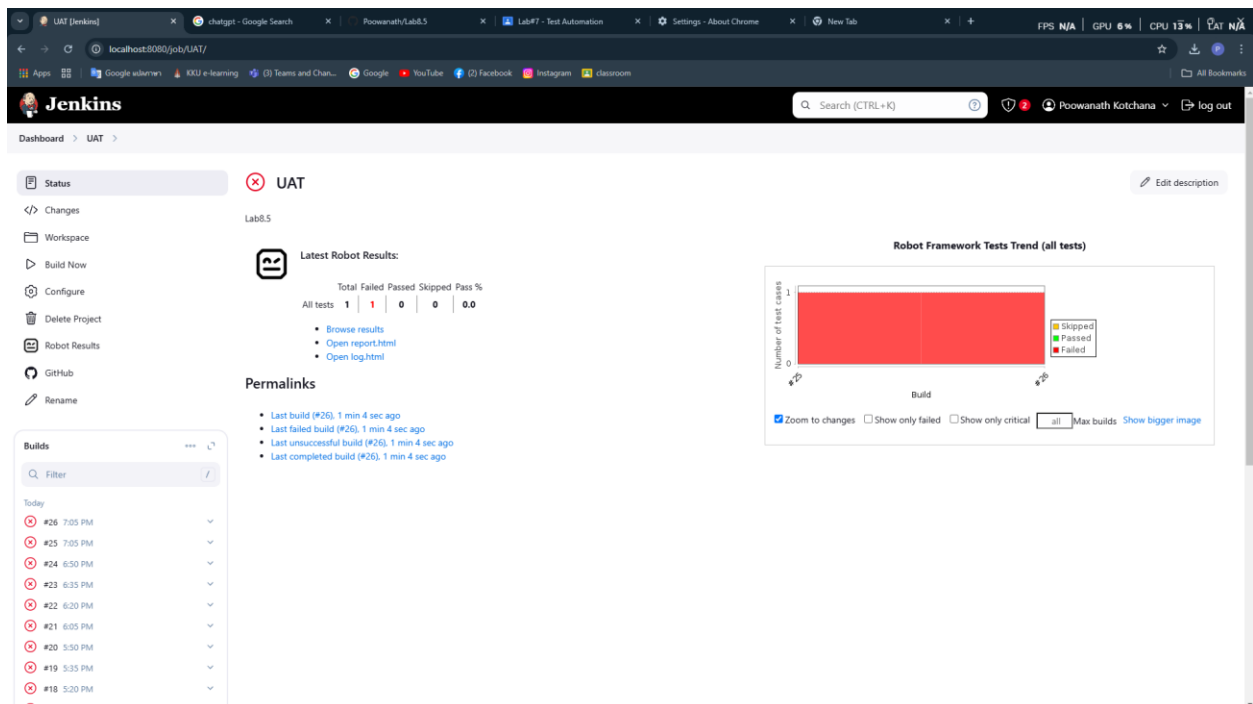
## Lab Worksheet

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

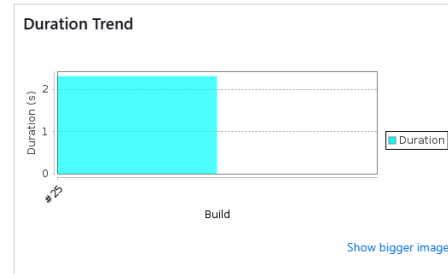
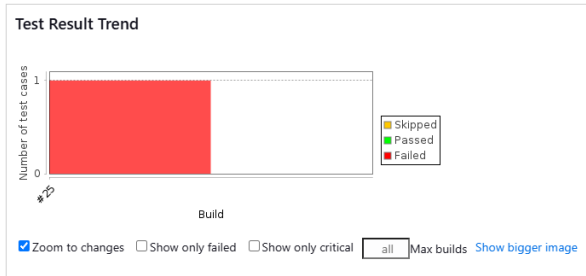
**[Check point#15]** Capture หน้าจอแสดงหน้าหลักของ Pipeline และ Console Output



## Lab Worksheet

## Robot Framework Test Results

**Executed:** 2025-01-29T01:55:27.707329  
**Duration:** 0:00:02.315 (+0:00:02.315)  
**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](#)



## Failed Test Cases

Name	Crit.	Duration	Age
UAT Lab7 001.Form Submission with Valid Data	no	0:00:02.052	1

## Test Suites

Dashboard > UAT > #25 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build #25

Timings

Git Build Data

Robot Results

Previous Build

Next Build

**Console Output**

Download Copy View as plain text

```
Started by user Poowanath Kotchana
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
Cloning the remote Git repository
Cloning repository https://github.com/Poowanath/Lab8.5.git
> git init /var/jenkins_home/workspace/UAT # timeout=10
Fetching upstream changes from https://github.com/Poowanath/Lab8.5.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/Poowanath/Lab8.5.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git config remote.origin.url https://github.com/Poowanath/Lab8.5.git # timeout=10
> git config --add remote.origin.fetch +refs/heads/*:refs/remotes/origin/* # timeout=10
Avoid second fetch
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision c916a89709c3547e3265b0a5b217c4bc4f88734b (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f c916a89709c3547e3265b0a5b217c4bc4f88734b # timeout=10
Commit message: "Add files via upload"
First time build. Skipping changelog.
[UAT] $ /bin/sh -xe /tmp/jenkins12455565478658880634.sh
+ robot UAT_lab7_001.robot UAT_lab7_002.robot
/tmp/jenkins12455565478658880634.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:
```