

## Lab Worksheet

ชื่อ-นามสกุล สิริยากร อายางคำ

รหัสนักศึกษา 653380348-2

Section 3

## 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

```

Terminal

PS C:\Users\Lenovo> mkdir Lab8_1
>>

Directory: C:\Users\Lenovo

Mode                LastWriteTime         Length Name
----                -
d-----          1/23/2025   9:07 AM                Lab8_1

```

```

PS C:\Users\Lenovo\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

```

```

PS C:\Users\Lenovo\Lab8_1> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
busybox       latest    af4709625109   3 months ago   4.27MB

```

- (1) สิ่งที่อยู่ภายใต้คอนเทนเนอร์ Repository คืออะไร busybox
- (2) Tag ที่ใช้บ่งบอกถึงอะไร latest
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

```
PS C:\Users\Lenovo\Lab8_1> docker run busybox
PS C:\Users\Lenovo\Lab8_1> docker run -it busybox sh
/ # ls
bin      dev      etc      home     lib      lib64    proc     root     sys      tmp      usr      var
```

```
/ # ls -la
total 48
drwxr-xr-x  1 root    root      4096 Jan 23 02:53 .
drwxr-xr-x  5 root    root      360 Jan 23 02:53 dev
drwxr-xr-x  1 root    root      4096 Jan 23 02:53 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 265 root    root        0 Jan 23 02:53 proc
drwx----- 1 root    root      4096 Jan 23 02:54 root
dr-xr-xr-x 11 root    root        0 Jan 23 02:53 sys
drwxrwxrwt  2 root    root      4096 Sep 26 21:31 tmp
drwxr-xr-x  4 root    root      4096 Sep 26 21:31 usr
drwxr-xr-x  4 root    root      4096 Sep 26 21:31 var
```

```
/ # exit
PS C:\Users\Lenovo\Lab8_1> docker run busybox echo Hello สิริยากร อาจามคำ from busybox
Hello สิริยากร อาจามคำ from busybox
PS C:\Users\Lenovo\Lab8_1> docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
48f30b5fff1a	busybox	"echo Hello สิริยากร..."	17 seconds ago	Exited (0) 16 seconds ago		goofy_sutherland
8530d689ea36	busybox	"sh"	6 minutes ago	Exited (0) About a minute ago		romantic_edison
b9fa68e7c755	busybox	"sh"	7 minutes ago	Exited (0) 7 minutes ago		objective_haslett

```
PS C:\Users\Lenovo\Lab8_1>
```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

สามารถเข้าสู่ shell ภายใน container และพิมพ์คำสั่งโต้ตอบได้

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

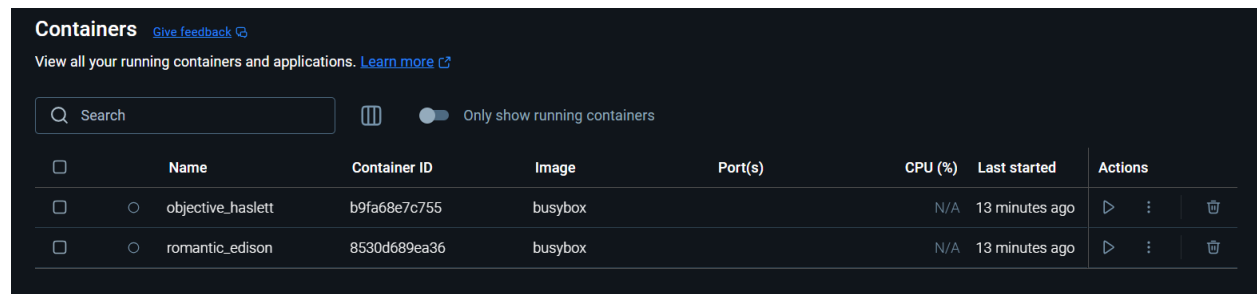
แสดงข้อมูลใน Container

## Lab Worksheet

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

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

```
PS C:\Users\Lenovo\Lab8_1> docker rm 48f30b5fff1ae78c7229aa4258a8b5d4dcc9710143b454ee3e809f2b80b77968
>>
48f30b5fff1ae78c7229aa4258a8b5d4dcc9710143b454ee3e809f2b80b77968
PS C:\Users\Lenovo\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

```
PS C:\Users\Lenovo> mkdir Lab8_2

Directory: C:\Users\Lenovo

Mode                LastWriteTime         Length Name
----                -
d-----          1/23/2025  10:07 AM                Lab8_2

PS C:\Users\Lenovo> cd Lab8_2
```

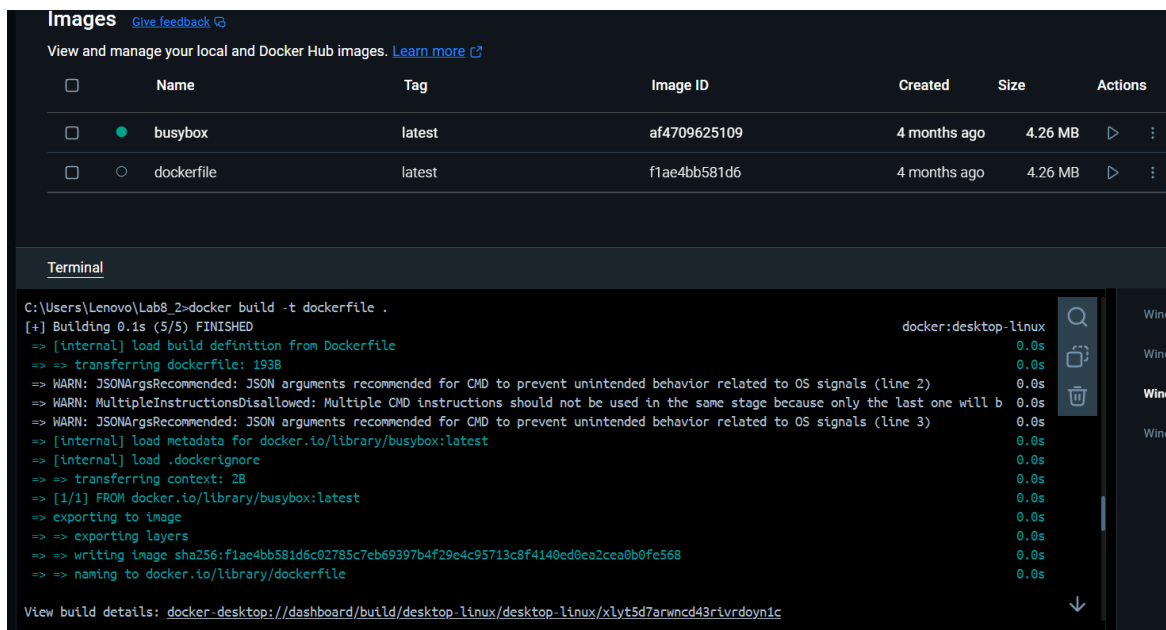
```
Terminal

PS C:\Users\Lenovo\Lab8_2> New-Item Dockerfile.swp -ItemType File
>>

Directory: C:\Users\Lenovo\Lab8_2

Mode                LastWriteTime         Length Name
----                -
-a-----          1/23/2025  10:13 AM                0 Dockerfile.swp
```

## Lab Worksheet



```
C:\Users\Lenovo\Lab8_2>docker run dockerfile
สวัสดี ยาวนาน 653380348-2 ก็ฟ
```

(1) คำสั่งที่ใช้ในการ run คือ

Docker run dockerfile

(2) Option -t ในคำสั่ง `$ docker build` ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป  
ใช้เพื่อกำหนดชื่อและแท็ก 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."

## Lab Worksheet

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
```

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

```
PS C:\Users\Lenovo> mkdir Lab8_3

Directory: C:\Users\Lenovo

Mode                LastWriteTime         Length Name
----                -
d-----          1/23/2025   8:37 PM             Lab8_3

PS C:\Users\Lenovo> cd Lab8_3
```

```
Dockerfile.swp.txt
File Edit View

FROM busybox
CMD echo "Hi there. My work is done. You can run them from my Docker image."
CMD echo "สิริยากร อายางคำ 653380348-2"
```

## Lab Worksheet

```
C:\Users\Lenovo\Lab8_3>docker build -t siriyakorn3482/lab8 .
[+] Building 0.1s (5/5) FINISHED                                docker:desktop-L
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 206B
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
=> WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will b
=> WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
=> [internal] load metadata for docker.io/library/busybox:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/busybox:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:85542047b48f1839c6a534f78c301420ddf14caff3022292d1aa99e8fcbfc4ce
=> => naming to docker.io/siriyakorn3482/lab8
```

```
PS C:\Users\Lenovo\Lab8_3> docker run siriyakorn3482/lab8
สัณยการ ฉายาณาค่า 653380348-2
```

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

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

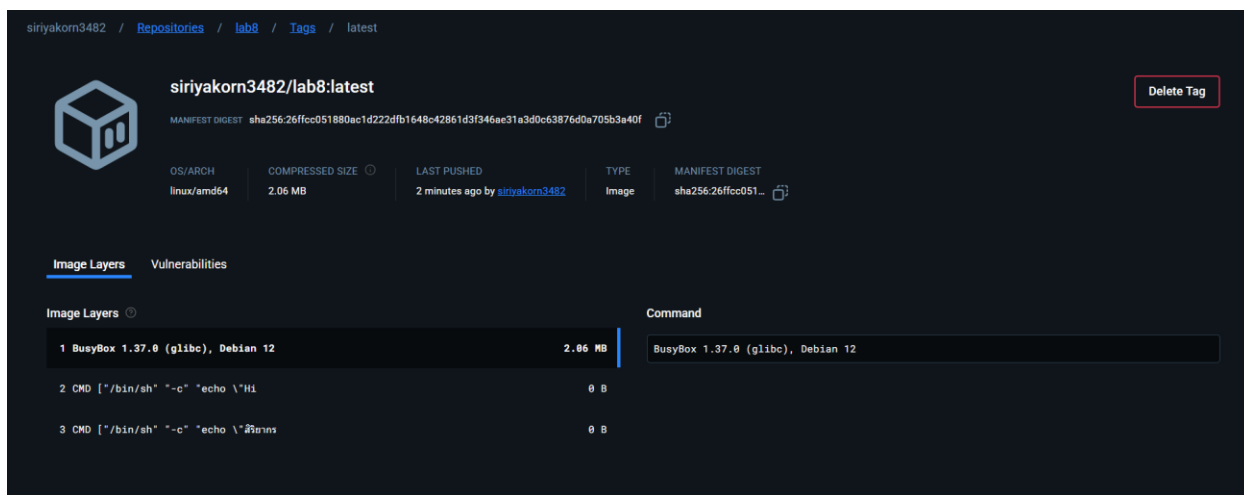
```
PS C:\Users\Lenovo> docker login -u siriyakorn3482
Password:

Login Succeeded
PS C:\Users\Lenovo>
```

```
PS C:\Users\Lenovo> docker push siriyakorn3482/lab8
Using default tag: latest
The push refers to repository [docker.io/siriyakorn3482/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:26ffcc051880ac1d222dfb1648c42861d3f346ae31a3d0c63876d0a705b3a40f size: 527
PS C:\Users\Lenovo>
```



## Lab Worksheet



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

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_4
2. ทำการ Clone ซอร์สโค้ดของเว็บแอปพลิเคชันจาก GitHub repository  
<https://github.com/docker/getting-started.git> ลงใน Directory ที่สร้างขึ้น โดยใช้คำสั่ง  
`$ git clone https://github.com/docker/getting-started.git`
3. เปิดดูองค์ประกอบภายใน getting-started/app เมื่อพบไฟล์ package.json ให้ใช้ Text editor ในการเปิดอ่าน

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

```
PS C:\Users\Lenovo> mkdir Lab8_4

Directory: C:\Users\Lenovo

Mode                LastWriteTime         Length Name
----                -
d-----          1/29/2025   8:21 PM             Lab8_4

PS C:\Users\Lenovo> cd Lab8_4
PS C:\Users\Lenovo\Lab8_4>
```

## Lab Worksheet

```
PS C:\Users\Lenovo\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 | 123.00 KiB/s, done.
Resolving deltas: 100% (523/523), done.
PS C:\Users\Lenovo\Lab8_4>
```

```
PS C:\Users\Lenovo\Lab8_4> cd getting-started/app
PS C:\Users\Lenovo\Lab8_4\getting-started\app> dir

Directory: C:\Users\Lenovo\Lab8_4\getting-started\app

Mode                LastWriteTime         Length Name
----                -
d-----          1/29/2025   8:23 PM             spec
d-----          1/29/2025   8:23 PM             src
-a-----          1/29/2025   8:23 PM        678 package.json
-a-----          1/29/2025   8:23 PM       150541 yarn.lock

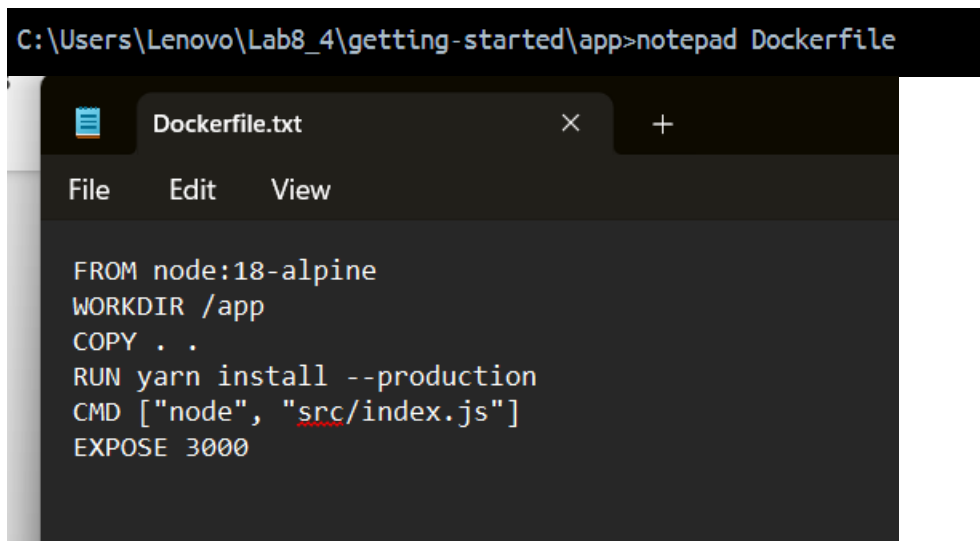
PS C:\Users\Lenovo\Lab8_4\getting-started\app>
```

```
{} package.json X
C: > Users > Lenovo > 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  > Debug
7  "scripts": {
8    "prettify": "prettier -l --write \"**/*.js\"",
9    "test": "jest",
10   "dev": "nodemon src/index.js"
11 },
12 "dependencies": {
13   "express": "^4.18.2",
14   "mysql2": "^2.3.3",
15   "sqlite3": "^5.1.2",
16   "uuid": "^9.0.0",
17   "wait-port": "^1.0.4"
18 },
19 "resolutions": {
20   "ansi-regex": "5.0.1"
21 },
22 "prettier": {
23   "trailingComma": "all",
24   "tabWidth": 4,
25   "useTabs": false,
26   "semi": true,
27   "singleQuote": true
28 },
29 "devDependencies": {
30   "jest": "^29.3.1",
31   "nodemon": "^2.0.20",
32   "prettier": "^2.7.1"
33 }
34 }
```

## 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

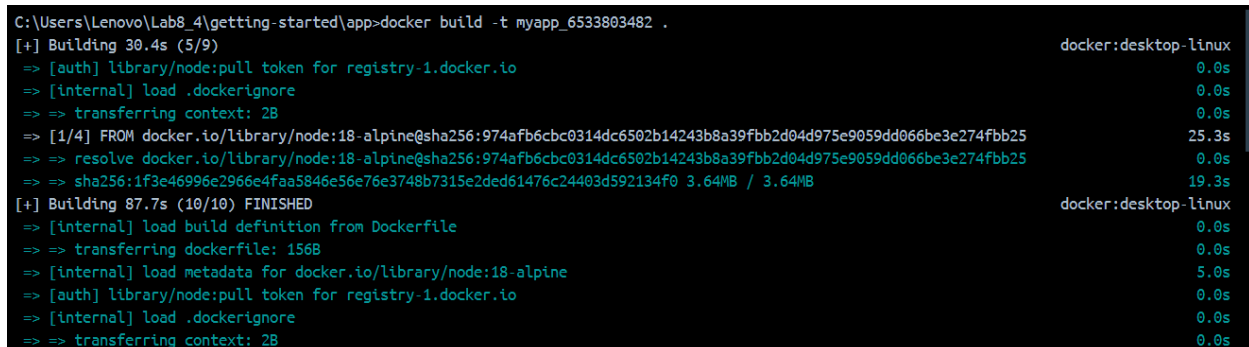


C:\Users\Lenovo\Lab8\_4\getting-started\app>notepad Dockerfile

```

Dockerfile.txt
File Edit View
FROM node:18-alpine
WORKDIR /app
COPY . .
RUN yarn install --production
CMD ["node", "src/index.js"]
EXPOSE 3000

```



```

C:\Users\Lenovo\Lab8_4\getting-started\app>docker build -t myapp_6533803482 .
[+] Building 30.4s (5/9)
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 25.3s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB 19.3s
[+] Building 87.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile 0.0s
=> => transferring dockerfile: 156B 0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine 5.0s
=> [auth] library/node:pull token for registry-1.docker.io 0.0s
=> [internal] load .dockerignore 0.0s
=> => transferring context: 2B 0.0s

```

## Lab Worksheet

```

=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 63.9s
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> => sha256:1f3e46996e2966e4faa5846e56e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB 19.3s
=> => sha256:37892ffbfc8a871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB 62.8s
=> => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB 0.0s
=> => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aece8647b17a7bcebbbc385e 1.72kB / 1.72kB 0.0s
=> => sha256:dcbf7b337595be6f4d214e4eed84f230eef0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB 0.0s
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB 16.9s
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B 17.7s
=> => extracting sha256:1f3e46996e2966e4faa5846e56e3748b7315e2ded61476c24403d592134f0 0.1s
=> => extracting sha256:37892ffbfc8a871a10f813803949d18c3015a482051d51b7e0da02525e63167c 0.9s
=> => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 0.0s
=> => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 0.0s
=> [internal] load build context 0.7s
=> => transferring context: 4.62MB 0.7s
=> [2/4] WORKDIR /app 0.1s
=> [3/4] COPY . . 0.1s
=> [4/4] RUN yarn install --production 18.0s
=> exporting to image 0.6s
=> => exporting layers 0.6s
=> => writing image sha256:da9aa85a8ee8a8dca158ee02bbfb7b4f3cd7305981e36d3d6a82e55ef1a945ec 0.0s
=> => naming to docker.io/library/myapp_6533803482 0.0s

```

6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

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

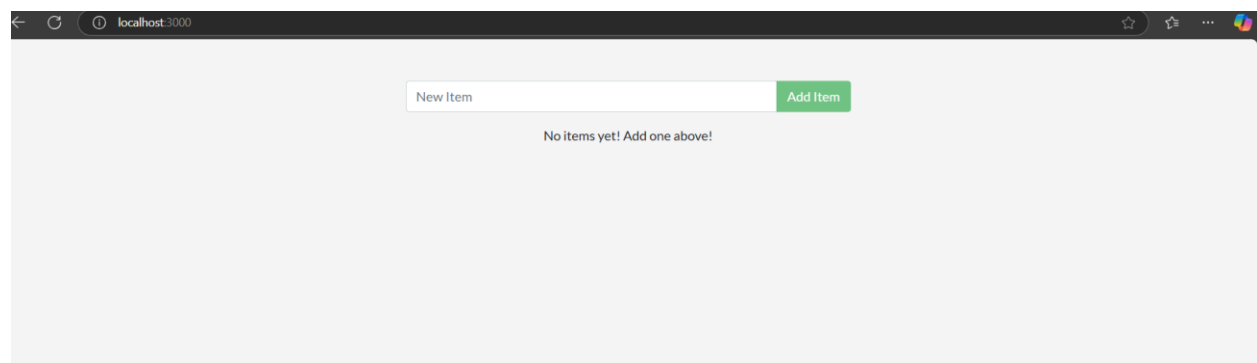
7. เปิด Browser ไปที่ URL = <http://localhost:3000>

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

```

C:\Users\Lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533803482
486a5e5c74f4cb4c94612c6674a583ef53b2baf616287c81aa0ead93b027a69b

```



หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

## Lab Worksheet

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
  - 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

```

55     {items.length === 0 && (
56         <p className="text-center">There is no TODO item. Please add one to the list. By [สิริยากร อายวงค์ 653380348-2]</p>
57     )
58 }

```

```

C:\Users\Lenovo\Lab8_4\getting-started\app>docker build -t myapp_6533803482 .
[+] Building 45.1s (10/10) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 156B                                              0.0s
=> [internal] load metadata for docker.io/library/node:18-alpine                2.4s
=> [auth] library/node:pull token for registry-1.docker.io                     0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                    0.0s
=> [1/4] FROM docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 0.0s
=> [internal] load build context                                                  0.1s
=> => transferring context: 8.14kB                                                0.0s
=> CACHED [2/4] WORKDIR /app                                                     0.0s
=> [3/4] COPY . .                                                                0.0s
=> [4/4] RUN yarn install --production                                          41.9s
=> exporting to image                                                            0.6s
=> => exporting layers                                                            0.6s
=> => writing image sha256:f752e0e5cf19d8a9b4420cd52d9862ac652071f3fccfa61d385950638aa72b67 0.0s
=> => naming to docker.io/library/myapp_6533803482                             0.0s

```

View build details: [docker\\_desktop://dashboard/build/desktop-linux/desktop-linux/gii3kp3esqdmyardm74r9lkkd](https://dashboard/build/desktop-linux/desktop-linux/gii3kp3esqdmyardm74r9lkkd)

## Lab Worksheet

```
C:\Users\Lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533803482
3e7c31c12d795f187346e2dbd73936d26011dac6db14fbd6cee6e380f9fad21e
docker: Error response from daemon: driver failed programming external connectivity on endpoint modest_meitner (b72afaf9a73ca863ae6676df0a35acb2
28560259d6cc1b982c81aa32c732be03): Bind for 0.0.0.0:3000 failed: port is already allocated.
```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

Docker พยายามที่จะเชื่อมต่อ (bind) พอร์ต 3000 บนเครื่อง แต่ไม่สามารถทำได้เพราะพอร์ตนั้นถูกใช้งานโดยโปรแกรมอื่นๆ หรือคอนเทนเนอร์อื่นแล้ว

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>

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

```
C:\Users\Lenovo\Lab8_4\getting-started\app>docker ps
```

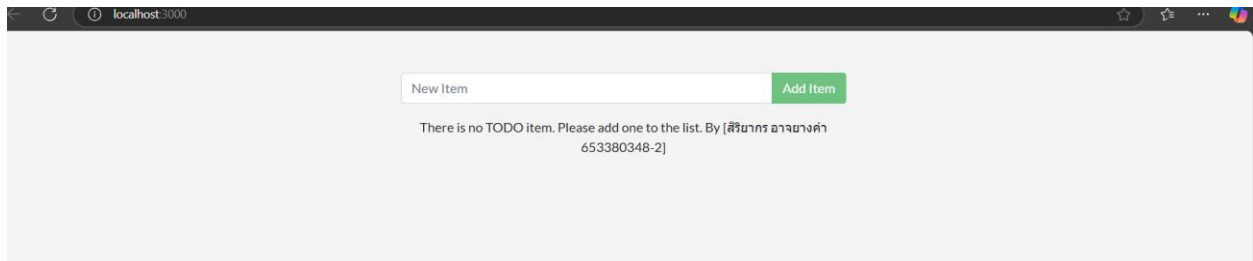
CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
486a5e5c74f4	da9aa85a8ee8	"docker-entrypoint.s..."	14 minutes ago	Up 14 minutes	0.0.0.0:3000->3000/tcp	vigilant_leavitt

```
C:\Users\Lenovo\Lab8_4\getting-started\app>docker stop 486a5e5c74f4
486a5e5c74f4
```

```
C:\Users\Lenovo\Lab8_4\getting-started\app>docker rm 486a5e5c74f4
486a5e5c74f4
```

## Lab Worksheet

```
C:\Users\Lenovo\Lab8_4\getting-started\app>docker run -dp 3000:3000 myapp_6533803482
2ee7399094ce17e273a900a0de98e1173301ca7ae39c831c5145355429785118
```



### แบบฝึกปฏิบัติที่ 8.5: เริ่มต้นสร้าง Pipeline อย่างง่ายสำหรับการ Deploy ด้วย Jenkins

1. เปิด Command line หรือ Terminal บน Docker Desktop
2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต  
`$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure jenkins/jenkins:lts-jdk17`  
 หรือ  
`$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk17`
3. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

**[Check point#12] Capture หน้าจอที่แสดงผล Admin password**

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

3eea46630e3542c996599177cd6040d4

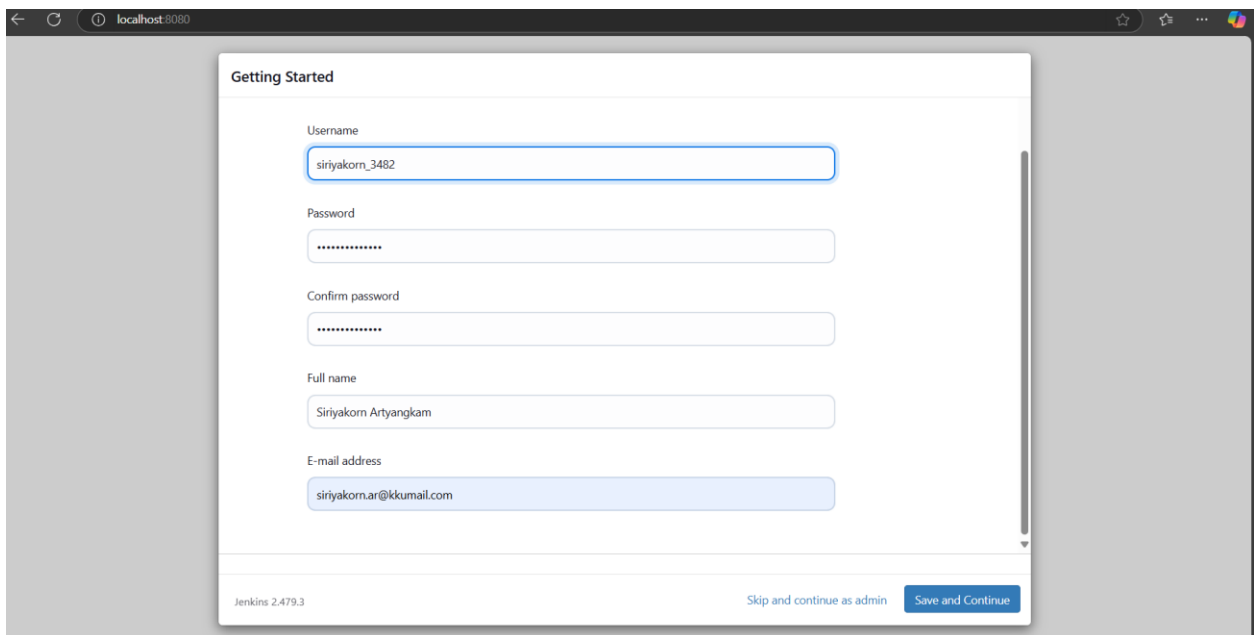
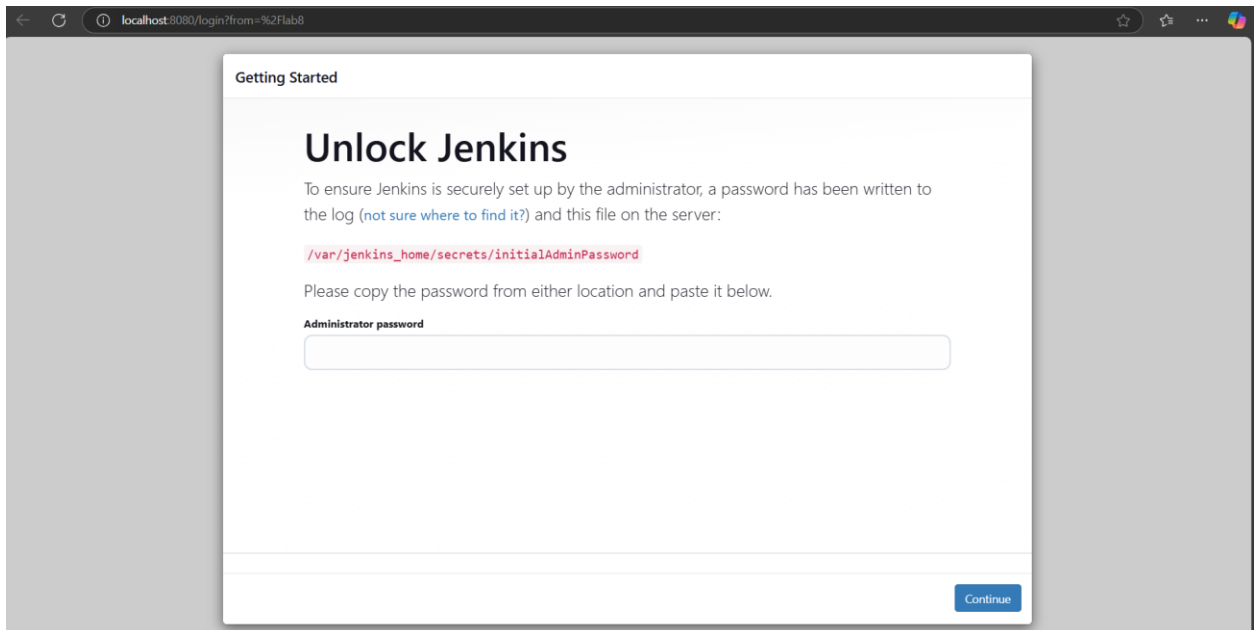
This may also be found at: /var/jenkins_home/secrets/initialAdminPassword

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

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

## Lab Worksheet

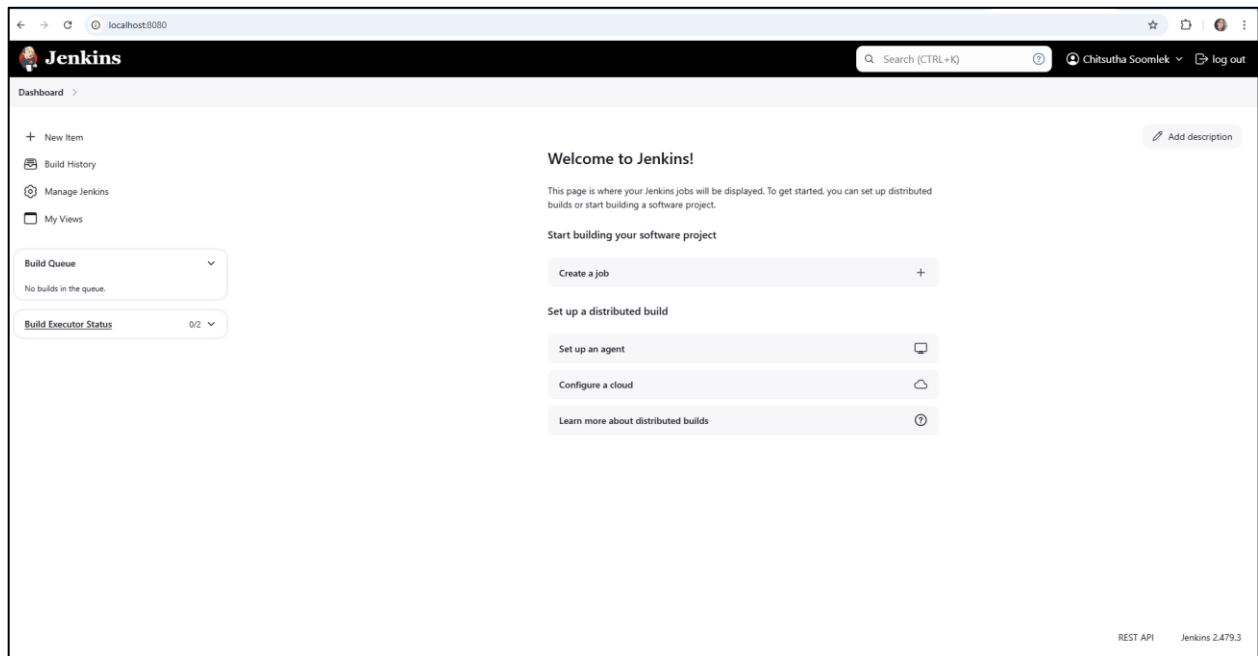
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



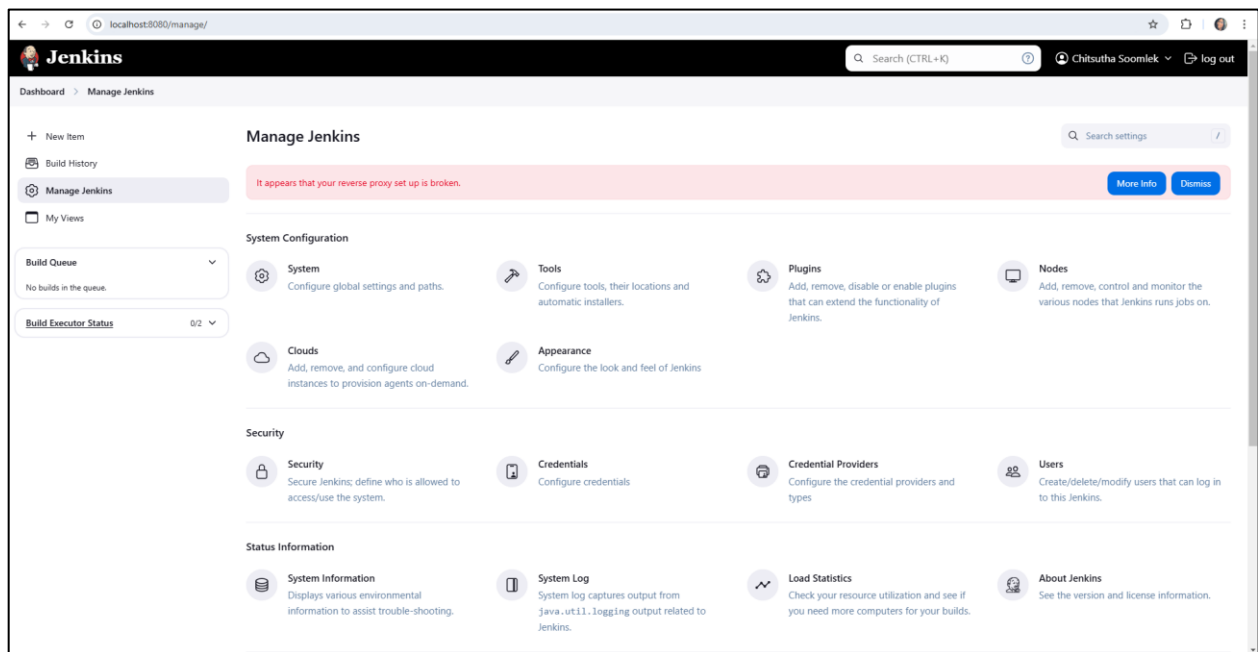
7. กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบหน้าจอ Dashboard ดังแสดงในภาพ



## Lab Worksheet

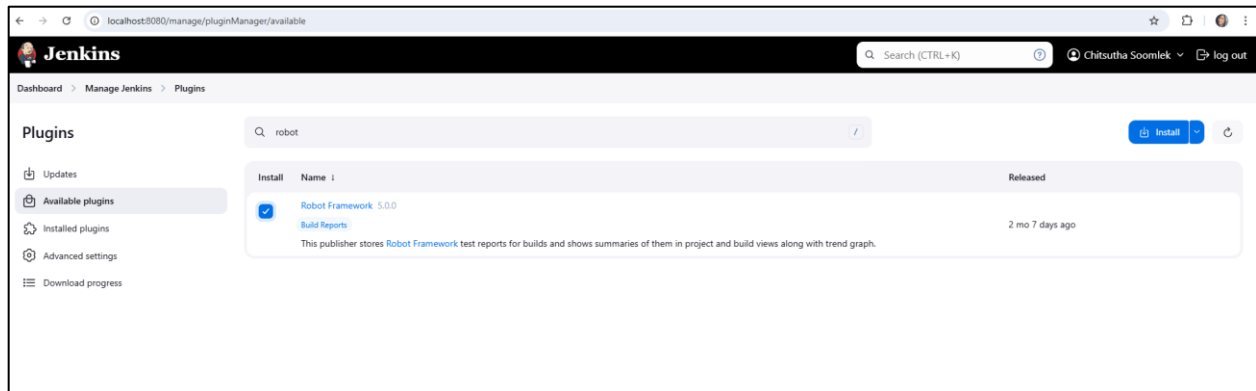


## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins

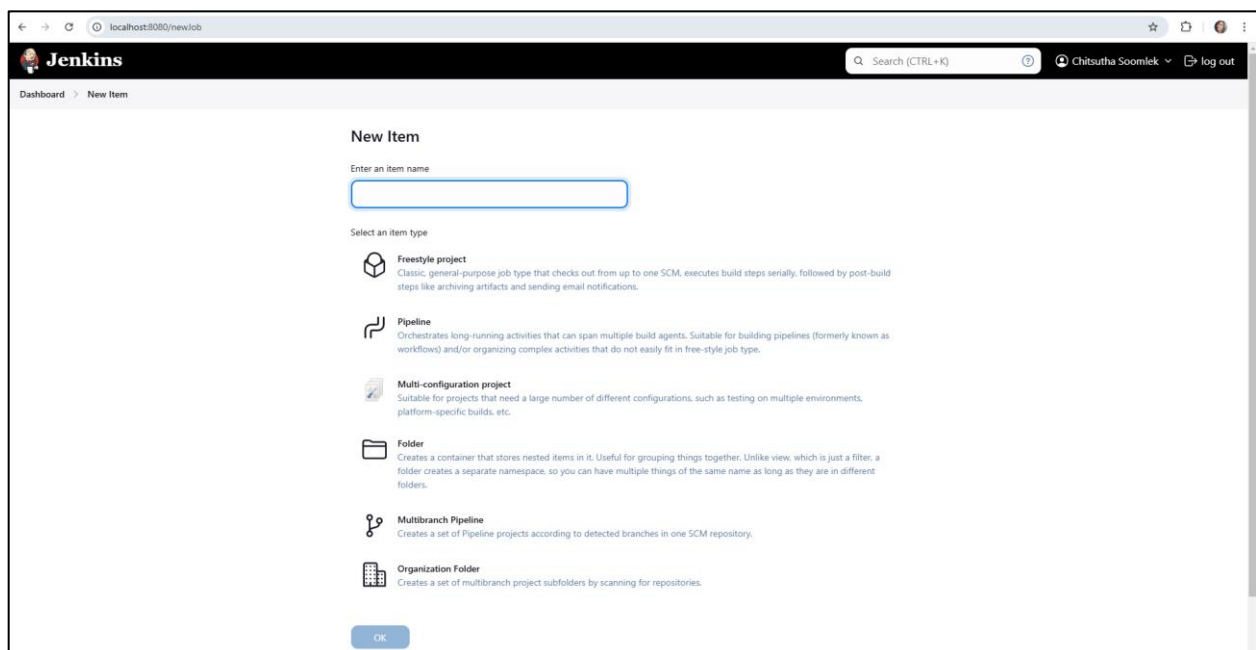


## Lab Worksheet

10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



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



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 ไปถึงไฟล์ให้เรียบร้อยด้วย)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

The screenshot shows the Jenkins 'Getting Started' page with the 'Instance Configuration' section. The 'Jenkins URL' field is filled with 'http://localhost:8080/lab4'. Below the field, there is explanatory text about the Jenkins URL and a 'Save and Finish' button. The bottom of the page shows the Jenkins dashboard with a sidebar for 'Plugins' and a 'Download progress' section listing various plugins and their installation status.

**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

**Jenkins** Search (CTRL+K) 🔒 🔴 👤 Siriyakorn Artyangkam 🚪 log out

Dashboard > Manage Jenkins > Plugins

### Plugins

- 🔄 Updates
- 📦 Available plugins
- 🔧 Installed plugins
- ⚙️ Advanced settings
- 📋 **Download progress**

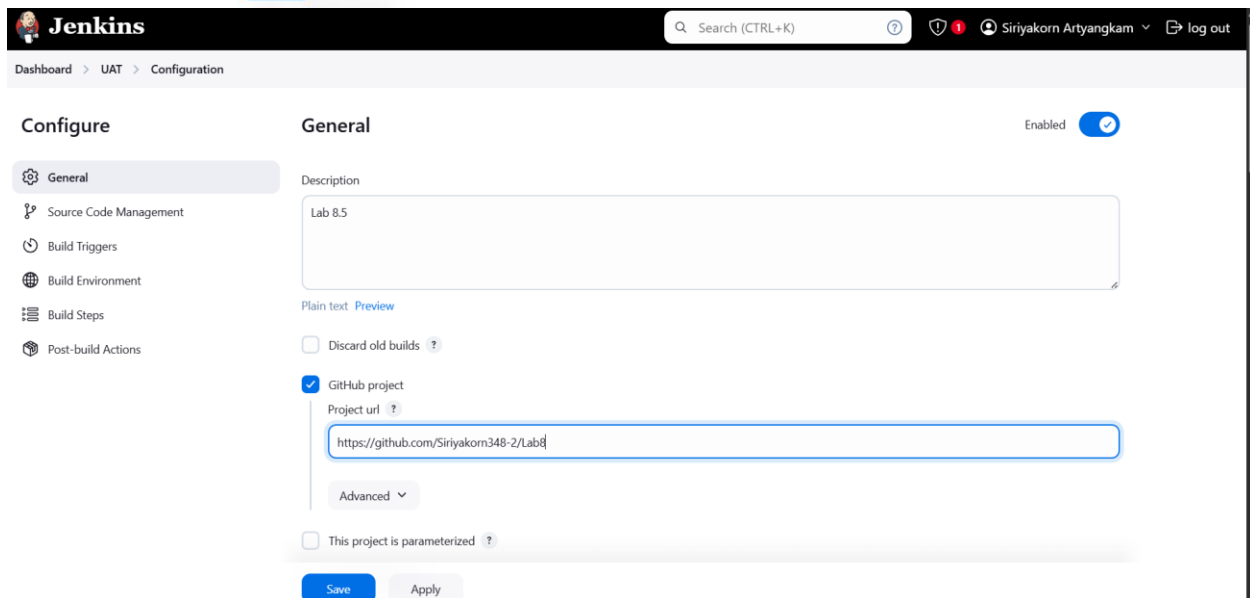
### Download progress

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Icons API	🟢 Success
Folders	🟢 Success
OWASP Markup Formatter	🟢 Success
ASM API	🟢 Success
JSON Path API	🟢 Success
Structs	🟢 Success
Pipeline: Step API	🟢 Success
Token Macro	🟢 Success
Build Timeout	🟢 Success
bouncycastle API	🟢 Success
Credentials	🟢 Success
Plain Credentials	🟢 Success
Variant	🟢 Success
SSH Credentials	🟢 Success
Credentials Binding	🟢 Success
SCM API	🟢 Success

## Lab Worksheet



## Lab Worksheet

Dashboard > UAT > Configuration

### Configure

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

#### Source Code Management

☒ None  
☐ Git ?

#### Build Triggers

☐ Trigger builds remotely (e.g., from scripts) ?  
☐ Build after other projects are built ?  
☒ Build periodically ?

Schedule ?

H/15 \* \* \* \*

Would last have run at Wednesday, January 29, 2025 at 2:50:58 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 3:05:58 PM Coordinated Universal Time.

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

Save Apply

### Configure

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

#### Git ?

Repositories ?

Repository URL ?

https://github.com/Siriyakorn348-2/Lab8.git

Credentials ?

- none -

+ Add

Advanced ▾

Add Repository

Branches to build ?

Branch Specifier (blank for 'any') ?

\*/main

Save Apply

## 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 4:05:54 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 4:20:54 PM Coordinated Universal Time.

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

## Lab Worksheet

**Build Steps**

Execute shell ?

Command

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

```
robot valid_login.robot
```

Advanced ▾

---

**Post-build Actions**

Publish Robot Framework test results ?

Directory of Robot output

Path to directory containing robot xml and html files (relative to build workspace)

Advanced ▾

Thresholds for build result ?

🟡 %

20.0

🟢 %

80.0

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

☐ Include skipped tests in total count for thresholds

Save Apply

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

```
robot <path_to_your_robot_file>
```

**Post-build action:** เพิ่ม Publish Robot Framework test results -> ระบุได้เร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่าน

## Lab Worksheet

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

**Jenkins** Search (CTRL+K) ? [User: Siriyakorn Artyangkam] [log out]

Dashboard > UAT > #1

**Status** #1 (Jan 29, 2025, 4:20:05 PM) [Add description] [Keep this build forever]

</> Changes

📄 Console Output

📄 Edit Build Information

🗑️ Delete build '#1'

🕒 Timings

🔗 Git Build Data

🤖 Robot Results

🕒 Started by user Siriyakorn Artyangkam

🕒 This run spent:

- 3 ms waiting;
- 5.3 sec build duration;
- 5.3 sec total from scheduled to completion.

🔗 Revision: b089152e6b3ffc047c9f4c7baaab4fee06d3a3e1

🔗 Repository: <https://github.com/Siriyakorn348-2/Lab8.git>

- refs/remotes/origin/main

🤖 Robot Test Summary:

	Total	Failed	Passed	Skipped	Pass %
All tests	6	0	6	0	100.0

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

</> No changes.

**Jenkins** Search (CTRL+K) ? [User: Siriyakorn Artyangkam] [log out]

Dashboard >

+ New Item

📄 Build History

⚙️ Manage Jenkins

📄 My Views

Build Queue

No builds in the queue.

Build Executor Status

0/2

All +

S	W	Name	Last Success	Last Failure	Last Duration	Robot Results + Duration Trend
🔴	🌤️	UAT	N/A	53 sec #1	5.3 sec	🟢 6/6 passed

Icon: S M L

## Lab Worksheet

Containers
Images
Volumes
Builds
Docker Hub
Docker Scout
Extensions

### Containers

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

Only show running containers

	Name	Container ID	Image	Port(s)	CPU (%)	Last started	Actions
<input type="checkbox"/>	<input type="radio"/> festive_gagarin	070dd6dec1df6	jenkins/jenkins-its-jdk17	50000:50000 Show all ports (2)	0%	2 hours ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<input type="radio"/> suspicious_dijkstra	7bcbde7d94a3	jenkins/jenkins-its-jdk17	50000:50000 Show all ports (2)	0%	2 hours ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<input type="radio"/> nifty_black	6d5535dee8f9	jenkins/jenkins-its-jdk17	50000:50000 Show all ports (2)	0%	2 hours ago	<div></div> <div></div> <div></div>
<input type="checkbox"/>	<input checked="" type="radio"/> wonderful_fermi	1772cede7dc4	jenkins/jenkins-its-jdk17	50000:50000 Show all ports (2)	0.23%	2 hours ago	<div></div> <div></div> <div></div>

Showing 4 items

Terminal

```

_3482
2025-01-29 14:30:43.429+0000 [id-354] INFO h.m.UpdateCenter$UpdateCenterConfiguration#download: Downloading robot
2025-01-29 14:30:46.630+0000 [id-373] INFO jenkins.InitReactorRunner$1onAttained: Prepared all plugins
2025-01-29 14:30:46.631+0000 [id-377] INFO jenkins.InitReactorRunner$1onAttained: Started all plugins
2025-01-29 14:30:46.632+0000 [id-382] INFO jenkins.InitReactorRunner$1onAttained: Augmented all extensions
2025-01-29 14:30:46.634+0000 [id-387] INFO jenkins.InitReactorRunner$1onAttained: System config loaded
2025-01-29 14:30:46.635+0000 [id-392] INFO jenkins.InitReactorRunner$1onAttained: System config adapted
2025-01-29 14:30:46.636+0000 [id-366] INFO jenkins.InitReactorRunner$1onAttained: Loaded all jobs
2025-01-29 14:30:46.636+0000 [id-368] INFO jenkins.InitReactorRunner$1onAttained: Configuration for all jobs updated
2025-01-29 14:30:46.636+0000 [id-372] INFO jenkins.InitReactorRunner$1onAttained: Completed initialization
2025-01-29 14:30:46.637+0000 [id-354] INFO h.m.UpdateCenter$CompleteBatchJob#run: Completed installation of 1 plugins in 3.2 sec

```

Search

Windows

RAM 2.74 GB CPU 0.00% Disk 2.98 GB used (limit 1006.85 GB)

Search (CTRL+K)

Siriya Korn Artyangkarn

log out

Dashboard > UAT > #4 > Console Output

Status

Changes

Console Output

Edit Build Information

Delete build '#4'

Timings

Git Build Data

Robot Results

Previous Build

Console Output

Download

Copy

View as plain text

```

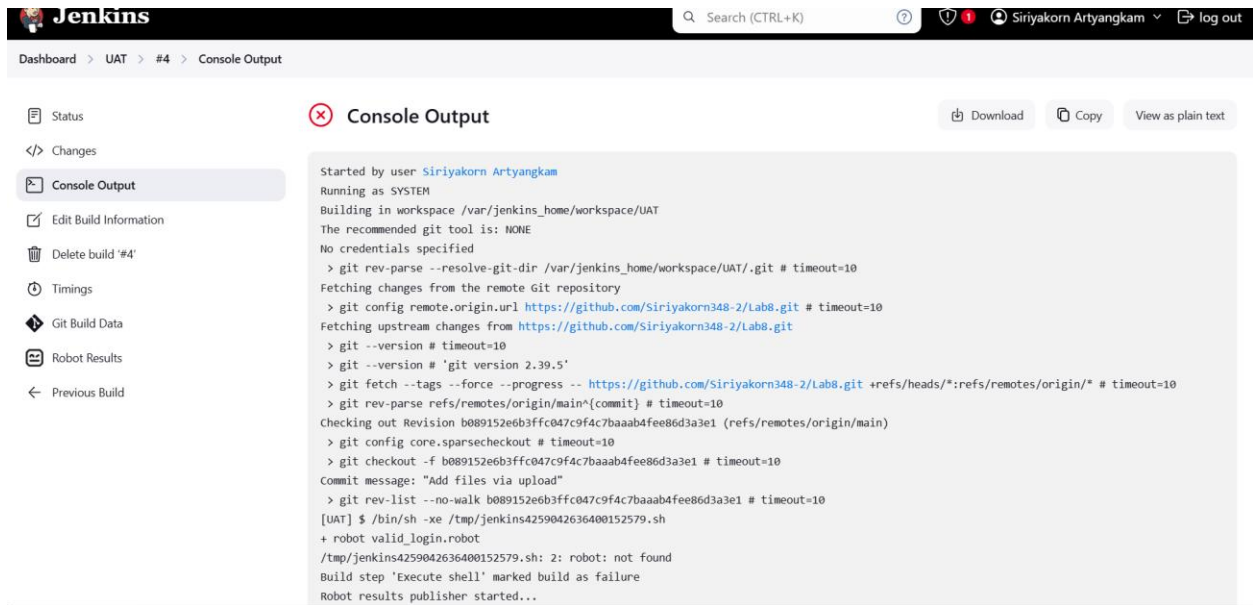
Started by user Siriya Korn Artyangkarn
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/jenkins_home/workspace/UAT/.git # timeout=10
Fetching changes from the remote git repository
> git config remote.origin.url https://github.com/SiriyaKorn348-2/Lab8.git # timeout=10
Fetching upstream changes from https://github.com/SiriyaKorn348-2/Lab8.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/SiriyaKorn348-2/Lab8.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins4259042636400152579.sh
+ robot valid_login.robot
/tmp/jenkins4259042636400152579.sh: 2: robot: not found
Build step 'Execute shell' marked build as failure
Robot results publisher started...

```

24



## Lab Worksheet



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and the user name 'Siriya Korn Artyangkarn' with a 'log out' link. The breadcrumb trail is 'Dashboard > UAT > #4 > Console Output'. On the left sidebar, the 'Console Output' tab is selected. The main content area displays the console output for build #4, which failed. The output text is as follows:

```
Started by user Siriya Korn Artyangkarn
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/SiriyaKorn348-2/Lab8.git # timeout=10
Fetching upstream changes from https://github.com/SiriyaKorn348-2/Lab8.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/SiriyaKorn348-2/Lab8.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 # timeout=10
Commit message: "Add files via upload"
> git rev-list --no-walk b089152e6b3ffc047c9f4c7baaab4fee86d3a3e1 # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins4259042636400152579.sh
+ robot valid_login.robot
/tmp/jenkins4259042636400152579.sh: 2: robot: not found
Build step 'Execute shell' marked build as failure
Robot results publisher started...
```