

## Lab Worksheet

ชื่อ-นามสกุล นาย เอกราช ชัยสงค์ รหัสนักศึกษา 653380355-5 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

## Lab Worksheet

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

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

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

PS C:\Users\computer\Desktop> mkdir Lab8_1

Directory: C:\Users\computer\Desktop

Mode                LastWriteTime         Length Name
----                -
d-----          1/23/2025   9:28 AM                Lab8_1

PS C:\Users\computer\Desktop> cd Lab8_1
PS C:\Users\computer\Desktop\Lab8_1> docker pull busybox
Using default tag: latest
latest: Pulling from library/busybox
9c0abc9c5bd3: Pull complete
Digest: sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbbf1
Status: Downloaded newer image for busybox:latest
docker.io/library/busybox:latest

PS C:\Users\computer\Desktop\Lab8_1> docker images
REPOSITORY          TAG         IMAGE ID      CREATED        SIZE
aixam                latest     ec5ada23d7f1  3 months ago  7.64GB
busybox              latest     af4709625109  3 months ago  4.27MB
synthesizedio/whalesay latest     07da125a0bc8  6 months ago  45.2MB
PS C:\Users\computer\Desktop\Lab8_1> |
```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร ชื่อของ Docker image หรือแหล่งที่มาของ image ซึ่งระบุว่า image นั้นถูกสร้างขึ้นมาจาก repository ไหน
- (2) Tag ที่ใช้บ่งบอกถึงอะไร บ่งบอกเวอร์ชันเฉพาะของ Docker image นั้น เช่น latest, 1.0, หรือ alpine
5. ป้อนคำสั่ง \$ docker run busybox
6. ป้อนคำสั่ง \$ docker run -it busybox sh
7. ป้อนคำสั่ง ls

## Lab Worksheet

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

```

PS C:\Users\computer\Desktop\Lab8_1> docker run busybox
PS C:\Users\computer\Desktop\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 02:36 .
drwxr-xr-x  1 root    root          4096 Jan 23 02:36 ..
-rwxr-xr-x  1 root    root           0 Jan 23 02:36 .dockerenv
drwxr-xr-x  2 root    root        12288 Sep 26 21:31 bin
drwxr-xr-x  5 root    root         360 Jan 23 02:36 dev
drwxr-xr-x  1 root    root         4096 Jan 23 02:36 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 387 root    root           0 Jan 23 02:36 proc
drwx----- 1 root    root         4096 Jan 23 02:36 root
dr-xr-xr-x 11 root    root           0 Jan 23 02:36 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\computer\Desktop\Lab8_1> docker run busybox echo "Hello Aekkarat Chaisong from busybox"
Hello Aekkarat Chaisong from busybox
PS C:\Users\computer\Desktop\Lab8_1> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS
PORTS         NAMES
847dd7529a34   busybox    "echo 'Hello Aekkara..." 10 seconds ago Exited (0) 9 seconds ago
c920c6d3dc00   busybox    "sh"                      About a minute ago Exited (0) About a minute a

```

## Lab Worksheet

```

Windows PowerShell
drwx----- 1 root root 4096 Jan 23 02:36 root
dr-xr-xr-x 11 root root 0 Jan 23 02:36 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\computer\Desktop\Lab8_1> docker run busybox echo "Hello Aekkarat Chaisong from busybox"
Hello Aekkarat Chaisong from busybox
PS C:\Users\computer\Desktop\Lab8_1> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS
PORTS         NAMES
847dd7529a34   busybox        "echo 'Hello Aekkara..." 10 seconds ago Exited (0) 9 seconds ago
c920c6d3dc00   busybox        "sh"                     About a minute ago Exited (0) About a minute a
go
fdff0cde95f4   busybox        "sh"                     2 minutes ago   Exited (0) 2 minutes ago
24f9f74670e7   synthesizedio/whalesay:latest   "/usr/local/bin/cows..." 23 minutes ago   Exited (0) 23 minutes ago
1059509bff05   synthesizedio/whalesay:latest   "/usr/local/bin/cows..." 25 minutes ago   Exited (0) 25 minutes ago
d56c1db53528   synthesizedio/whalesay:latest   "/usr/local/bin/cows..." 26 minutes ago   Exited (0) 26 minutes ago
b40d7dcc3cb7   aixam:latest   "jupyter lab --ip=0..." 3 months ago     Exited (0) 32 minutes ago
ea79e7806688   aixam:latest   "jupyter lab --ip=0..." 3 months ago     Created
43001d811426   aixam:latest   "jupyter lab --ip=0..." 3 months ago     Exited (255) 54 minutes ago
0.0.0.0:8000->8000/tcp modest_cohen
PS C:\Users\computer\Desktop\Lab8_1>

```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

เมื่อใช้ -it จะช่วยให้สามารถเข้าสู่ shell ภายใน container เพื่อรันคำสั่งต่าง ๆ ได้แบบ interactive

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

"Up x seconds/minutes": container กำลังทำงานอยู่

"Exited (0) x seconds/minutes ago": container หยุดทำงานแล้ว พร้อมแสดง exit code

"Created": container ถูกสร้างแต่ยังไม่เริ่มทำงาน

## Lab Worksheet

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

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

```

PS C:\Users\computer\Desktop\Lab8_1> docker ps -a
CONTAINER ID   IMAGE          NAMES                  COMMAND                  CREATED        STATUS
PORTS
c920c6d3dc00   busybox        determined_lamport     "sh"                     8 minutes ago  Exited (0) 7 minutes ago
fdff0cde95f4   busybox        objective_noyce        "sh"                     8 minutes ago  Exited (0) 8 minutes ago
24f9f74670e7   synthesizedi... zealous_shirley        "/usr/local/bin/cows..." 30 minutes ago  Exited (0) 30 minutes ago
1059509bfff05   synthesizedi... whalesay                "/usr/local/bin/cows..." 31 minutes ago  Exited (0) 31 minutes ago
d56c1db53528   synthesizedi... inspiring_hodgkin       "/usr/local/bin/cows..." 32 minutes ago  Exited (0) 32 minutes ago
b40d7dcc3cb7   aixam:latest   exciting_torvalds      "jupyter lab --ip=0..." 3 months ago   Exited (0) 39 minutes ago
ea79e7806688   aixam:latest   xenodochial_chaplygin  "jupyter lab --ip=0..." 3 months ago   Created
43001d811426   aixam:latest   modest_cohen          "jupyter lab --ip=0..." 3 months ago   Exited (255) About an hour ago
0.0.0.0:8000->8000/tcp
PS C:\Users\computer\Desktop\Lab8_1> docker rm c920c6d3dc00
c920c6d3dc00
PS C:\Users\computer\Desktop\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

## Lab Worksheet

FROM busybox

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

CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา ชื่อเล่น"

EOF

หรือใช้คำสั่ง

\$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

\$ docker build -t <ชื่อ Image> .

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```
PS C:\Users\computer\Desktop\Akkcs\Lab8_2> docker build -t dockering .
[+] Building 0.0s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 155B
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load metadata for docker.io/library/busybox:latest
=> [1/1] FROM docker.io/library/busybox
=> exporting to image
=> => exporting layers
=> => writing image sha256:34a151eca925e0247e4e0418a6331d1ce707724d43e85b73b67c997ea5840a3c
=> => naming to docker.io/library/dockering
PS C:\Users\computer\Desktop\Akkcs\Lab8_2> docker run dockering
"Aekkarat Chaisong 65338055-5 A"
PS C:\Users\computer\Desktop\Akkcs\Lab8_2> |
```

- (1) คำสั่งที่ใช้ในการ run คือ

**docker run dockering**

- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

Option -t (tag) ใช้สำหรับกำหนดชื่อและแท็กให้กับ Docker image ที่สร้างขึ้น

ช่วยให้สามารถระบุ Image ได้ง่ายเมื่อมีการใช้งานหรือจัดการ Image ในภายหลัง เช่น การเรียกดู

(docker images) หรือการรัน (docker run)

รูปแบบของ Option -t คือ <ชื่อ Image>:<tag> ซึ่ง tag เป็นทางเลือก (default คือ latest)

## Lab Worksheet

## แบบฝึกปฏิบัติที่ 8.3: การแชร์ Docker image ผ่าน Docker Hub

1. เปิดใช้งาน Docker desktop และ Login ด้วย Username และ Password ที่ลงทะเบียนกับ Docker Hub เอาไว้
2. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8\_3
3. ย้ายตำแหน่งปัจจุบันไปที่ Lab8\_3 เพื่อใช้เป็น Working directory
4. สร้าง Dockerfile.swp ไว้ใน Working directory

สำหรับเครื่องที่ใช้ระบบปฏิบัติการวินโดวส์ บันทึกคำสั่งต่อไปนี้ลงในไฟล์ โดยใช้ Text Editor ที่มี

```
FROM busybox
```

```
CMD echo "Hi there. My work is done. You can run them from my Docker image."
```

```
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"
```

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

```
$ cat > Dockerfile << EOF
```

```
FROM busybox
```

```
CMD echo "Hi there. My work is done. You can run them from my Docker image."
```

```
CMD echo "ชื่อ-นามสกุล รหัสนักศึกษา"
```

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

7. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

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

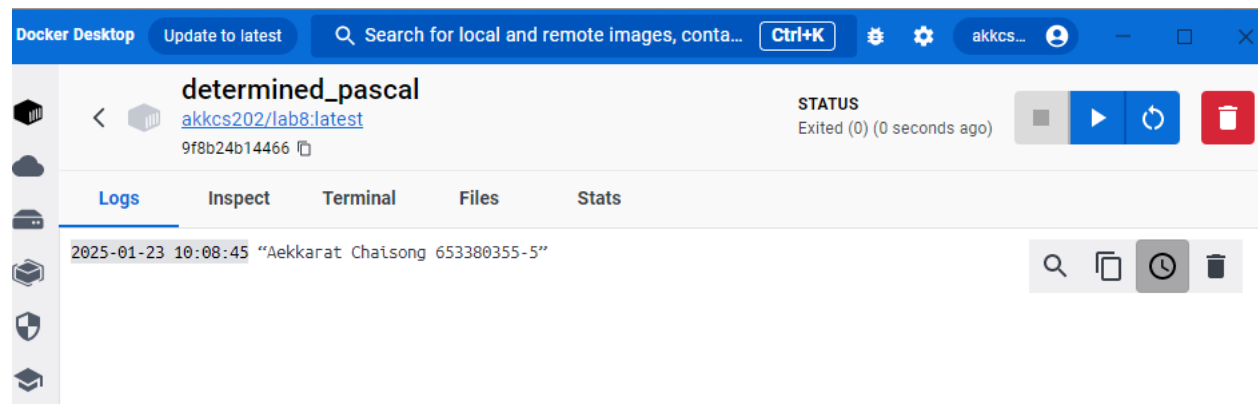
5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

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

## Lab Worksheet

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

```
PS C:\Users\computer\Desktop\Akkcs\Lab8_3> docker build -t akkcs202/lab8 .
[+] Building 0.0s (5/5) FINISHED
=> [internal] load build definition from Dockerfile                                0.0s
=> => transferring dockerfile: 181B                                              0.0s
=> [internal] load .dockerignore                                                  0.0s
=> => transferring context: 2B                                                  0.0s
=> [internal] load metadata for docker.io/library/busybox:latest                0.0s
=> CACHED [1/1] FROM docker.io/library/busybox                                  0.0s
=> exporting to image                                                            0.0s
=> => exporting layers                                                            0.0s
=> => writing image sha256:c43abba27cca6a921cf2ba5b754d71659c4572e480893939bbf4eff37c675e2b 0.0s
=> => naming to docker.io/akkcs202/lab8                                         0.0s
PS C:\Users\computer\Desktop\Akkcs\Lab8_3> |
```



6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

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

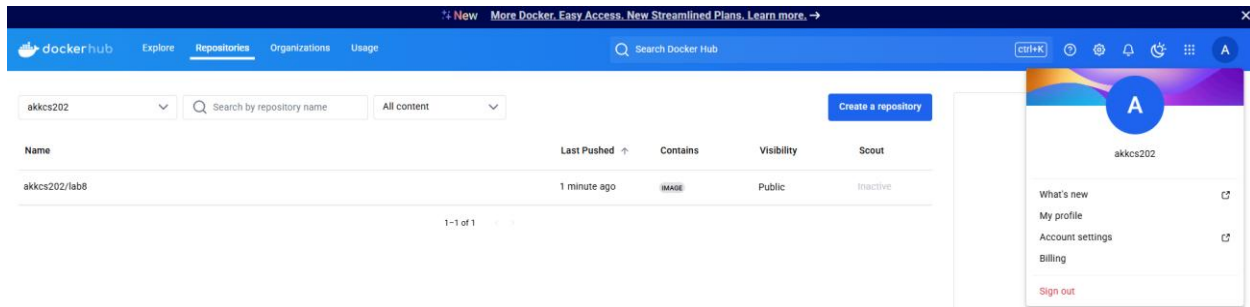
\$ docker login -u <username> -p <password>

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



## Lab Worksheet

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



### 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

## Lab Worksheet

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

name	Date modified	type	size
.git	1/23/2025 10:13 AM	File folder	
.github	1/23/2025 10:13 AM	File folder	
app	1/23/2025 10:13 AM	File folder	
docs	1/23/2025 10:13 AM	File folder	
.dockerignore	1/23/2025 10:13 AM	DOCKERIGNORE F...	1 KB
.gitignore	1/23/2025 10:13 AM	Git Ignore Source ...	1 KB
build.sh	1/23/2025 10:13 AM	SH Source File	1 KB
docker-compose.yml	1/23/2025 10:13 AM	Yaml Source File	1 KB
Dockerfile	1/23/2025 10:13 AM	File	2 KB
LICENSE	1/23/2025 10:13 AM	File	12 KB
mkdocs.yml	1/23/2025 10:13 AM	Yaml Source File	3 KB
README.md	1/23/2025 10:13 AM	Markdown Source...	2 KB
requirements.txt	1/23/2025 10:13 AM	Text Document	1 KB

```
PS C:\Users\computer\Desktop\Akkcs\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 | 12.70 MiB/s, done.
Resolving deltas: 100% (523/523), done.
PS C:\Users\computer\Desktop\Akkcs\Lab8_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

```
Windows PowerShell
PS C:\Users\aeikka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app> docker build -t myapp_6533803555 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 4.658MB
Step 1/6 : FROM node:18-alpine
18-alpine: Pulling from library/node
1f3e46996e29: Pull complete
37892ffbfcaa: Pull complete
5650d6de56fd: Pull complete
6504e29600c8: Pull complete
Digest: sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
Status: Downloaded newer image for node:18-alpine
--> dcbf7b337595
Step 2/6 : WORKDIR /app
--> Running in ab5ad3addec2
--> Removed intermediate container ab5ad3addec2
--> 8723a73a2423
Step 3/6 : COPY . .
--> d4eccc46234
Step 4/6 : RUN yarn install --production
--> Running in 9e8531ce1160
yarn install v1.22.22
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 8.63s.
```

```
Windows PowerShell
Step 2/6 : WORKDIR /app
--> Running in ab5ad3addec2
--> Removed intermediate container ab5ad3addec2
--> 8723a73a2423
Step 3/6 : COPY . .
--> d4eccc46234
Step 4/6 : RUN yarn install --production
--> Running in 9e8531ce1160
yarn install v1.22.22
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 8.63s.
--> Removed intermediate container 9e8531ce1160
--> 7ada579a6c77
Step 5/6 : CMD ["node", "src/index.js"]
--> Running in efada2dbd221
--> Removed intermediate container efada2dbd221
--> 19ee6db62493
Step 6/6 : EXPOSE 3000
--> Running in b0febef2b02c
--> Removed intermediate container b0febef2b02c
--> 5cedb5929084
Successfully built 5cedb5929084
Successfully tagged myapp_6533803555:latest
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and director
ies added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissions f
or sensitive files and directories.
PS C:\Users\aeikka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app>
```

## Lab Worksheet

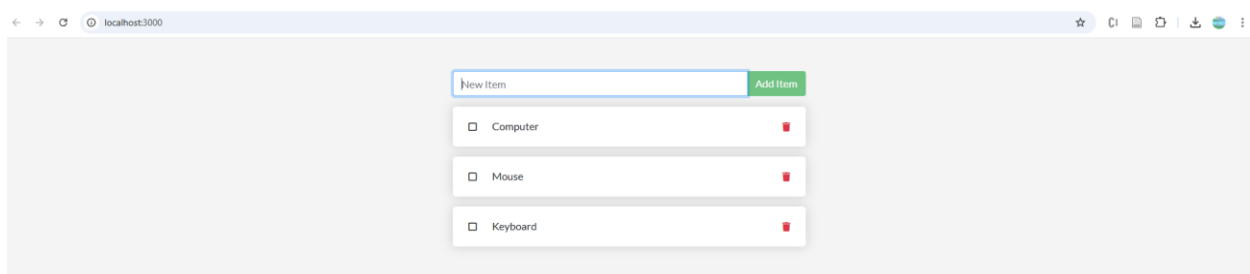
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

```
$ 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

```
PS C:\Users\aecka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_653380355529b3aa3176bd48ecfa2e72f469221f52912bd7d83e517b7daf343a89248b2b8e
PS C:\Users\aecka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app> |
```



หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

8. ทำการแก้ไข Source code ของ Web application ดังนี้

- a. เปิดไฟล์ src/static/js/app.js ด้วย Editor และแก้ไขบรรทัดที่ 56 จาก

```
<p className="text-center">No items yet! Add one above!</p> เป็น
```

```
<p className="text-center">There is no TODO item. Please add one to the list.
```

**By ชื่อและนามสกุลของนักศึกษา**</p>

- b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

**[Check point#10]** Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ พร้อมกับตอบคำถามต่อไปนี้

## Lab Worksheet

```
PS C:\Users\aeikka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app> docker build -t myapp_6533803555 .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
Install the buildx component to build images with BuildKit:
https://docs.docker.com/go/buildx/

Sending build context to Docker daemon 4.658MB
Step 1/6 : FROM node:l8-alpine
----> dcbf7b337595
Step 2/6 : WORKDIR /app
----> Using cache
----> 8723a73a2423
Step 3/6 : COPY . .
----> be7e13055648
Step 4/6 : RUN yarn install --production
----> Running in 2b0aa4a65d58
yarn install v1.22.22
[1/4] Resolving packages...
[2/4] Fetching packages...
[3/4] Linking dependencies...
[4/4] Building fresh packages...
Done in 8.79s.
----> Removed intermediate container 2b0aa4a65d58
----> a58f26df8015
Step 5/6 : CMD ["node", "src/index.js"]
```

```
PS C:\Users\aeikka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533803555
c60ca176748908f8dd162f8aa4c9a6b73102001e874b1816dd1e8162aac3cf08
docker: Error response from daemon: driver failed programming external connectivity on endpoint hungry_grothendieck (1462091a6e768e931ac5e4fb6b1bfcce35fe13bfdd2c1f0bd757f75e275a8c61): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS C:\Users\aeikka\OneDrive\Desktop\Software Engineer\Lab8\Lab8_4\getting-started\app>
```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร

เพราะชื่อของ Container มันซ้ำกับข้อ 5 และมันกำลังรันอยู่เลยรันตัวที่ Build ใหม่ไม่ได้

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

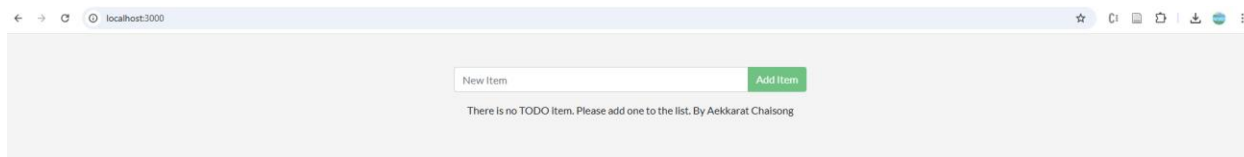
- ไปที่หน้าต่าง Containers
- เลือกไอคอนถังขยะในแถวของ Container ที่ต้องการจะลบ
- ยืนยันโดยการกด Delete forever

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



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

1. เปิด Command line หรือ Terminal บน Docker Desktop

2. ป้อนคำสั่งและทำการรัน container โดยผูกพอร์ต

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

หรือ

```
$ docker run -p 8080:8080 -p 50000:50000 --restart=on-failure -v
```

```
jenkins_home:/var/jenkins_home jenkins/jenkins:its-jdk17
```

3. บันทึกการรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

[Check point#12] Capture หน้าจอที่แสดงผล Admin password

```
Windows PowerShell
2025-01-29 12:31:37.261+0000 [id=72] INFO jenkins.InitReactorRunner$1#onAttained: Started all plugins
2025-01-29 12:31:37.262+0000 [id=49] INFO jenkins.InitReactorRunner$1#onAttained: Augmented all extensions
2025-01-29 12:31:37.426+0000 [id=49] INFO jenkins.InitReactorRunner$1#onAttained: System config loaded
2025-01-29 12:31:37.427+0000 [id=69] INFO jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-29 12:31:37.427+0000 [id=41] INFO jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-29 12:31:37.431+0000 [id=50] INFO jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-29 12:31:37.457+0000 [id=85] INFO hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-01-29 12:31:37.848+0000 [id=62] INFO jenkins.install.SetupWizard#init:

*****
*****
*****

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

33faac7b5c5a4b4dbcece6f70c9d9c43

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

*****
*****
*****

2025-01-29 12:31:43.869+0000 [id=62] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 12:31:43.886+0000 [id=33] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-29 12:31:45.793+0000 [id=85] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
```

## Lab Worksheet

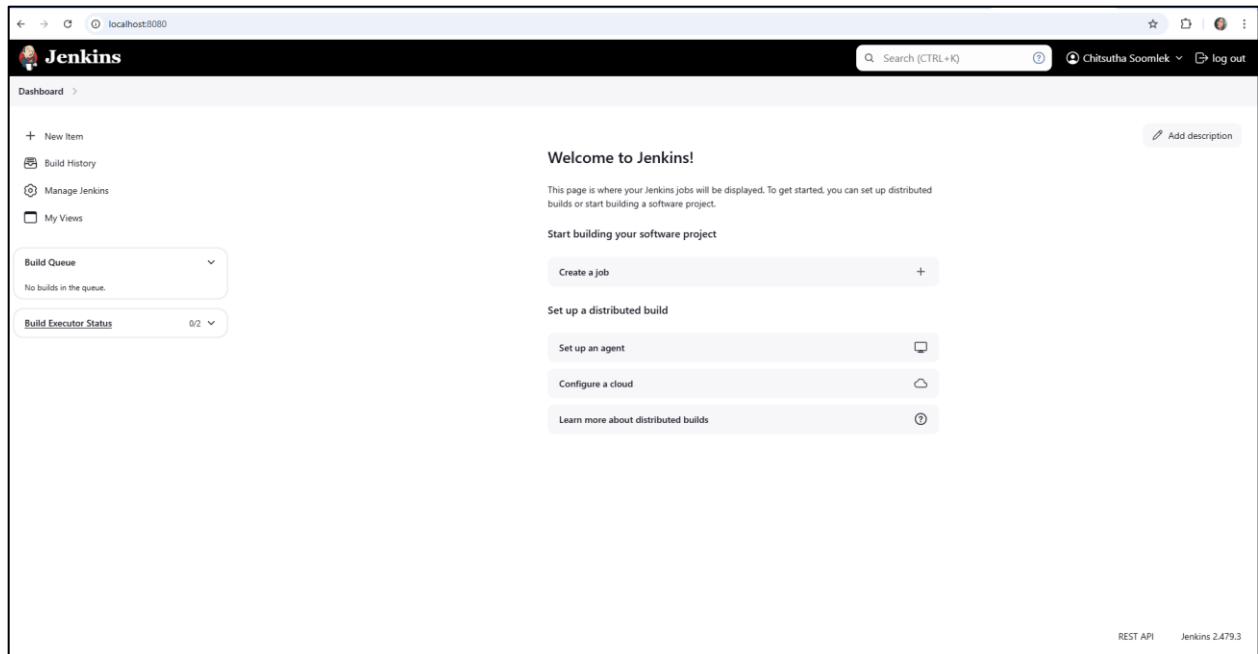
- เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
  - ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
  - สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri\_3062
- [Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า

The image displays two screenshots of the Jenkins Setup Wizard interface. The top screenshot shows the 'Getting Started' page where the user is prompted to enter their details. The fields are filled with: Username: Aekkarat\_3555, Password: (masked with dots), Confirm password: (masked with dots), Full name: Aekkarat chaisong, and E-mail address: aekkarat.c@kkumail.com. At the bottom right, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The bottom screenshot shows the 'Instance Configuration' page. It features a large heading 'Instance Configuration' and a field for 'Jenkins URL' which is set to 'http://localhost:8080/lab8'. Below this field, there is explanatory text about the Jenkins URL. At the bottom right, there are two buttons: 'Not now' and 'Save and Finish'.

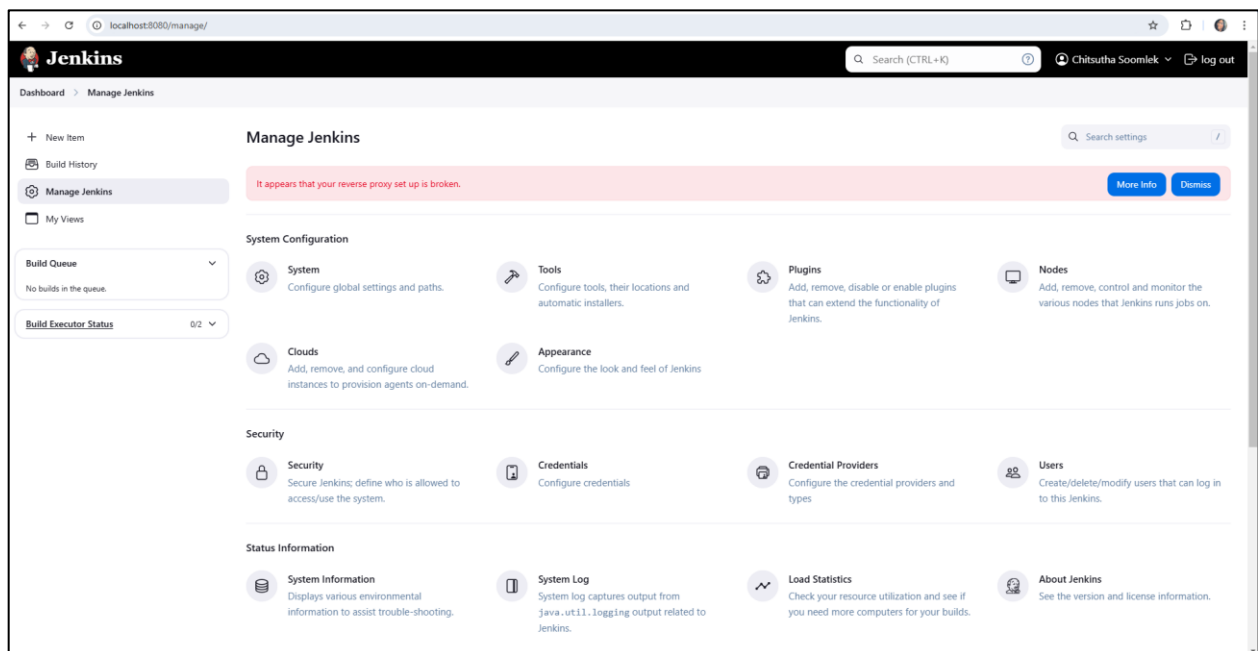
- กำหนด Jenkins URL เป็น <http://localhost:8080/lab8>

## Lab Worksheet

## 8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



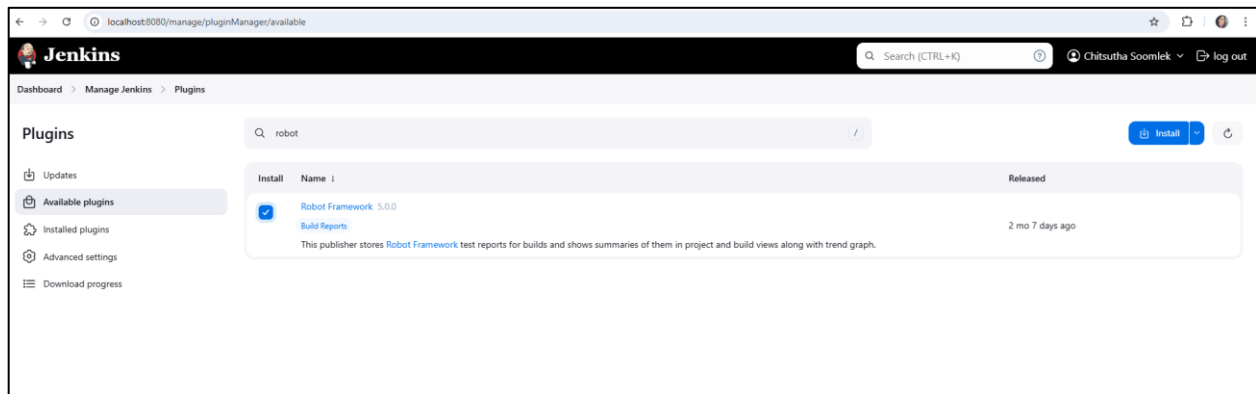
## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



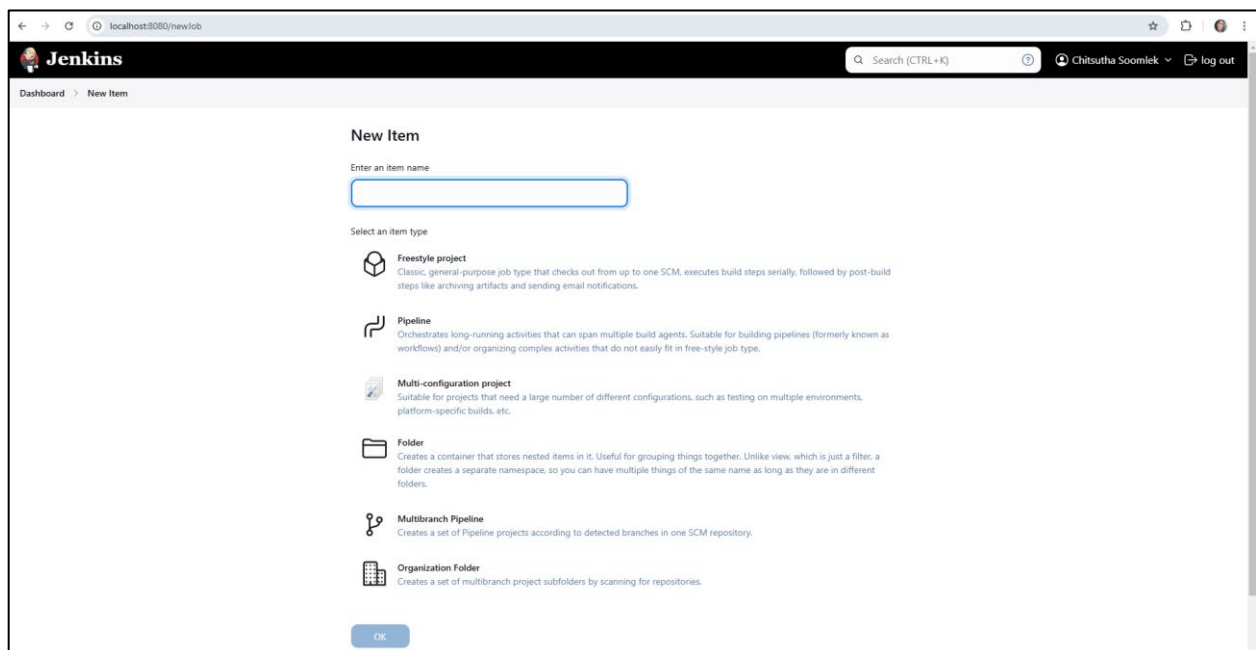
## 10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม



## Lab Worksheet



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



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้นตั้งค่าที่จำเป็นในหน้านั้นทั้งหมด 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 ไปถึงไฟล์ให้เรียบร้อยด้วย)

## Lab Worksheet

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

☐ Discard old builds ?

☒ GitHub project

Project url ?

`https://github.com/Aekkarat1729/softenlab8/`

Advanced ▾

**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 2:05:08 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 2:20:08 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

**Build Steps**

Execute shell ?

Command

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

`robot login_test/lab7.robot`

Advanced ▾

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

**robot login\_test/lab7.robot**

**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

## Lab Worksheet

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

## 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 ?

🟡%

🟢%

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

☐ Include skipped tests in total count for thresholds

The screenshot shows the Jenkins web interface for a pipeline named 'UAT'. The top navigation bar includes the Jenkins logo, a search bar, and user information. The left sidebar contains a 'Status' tab and a list of actions: Changes, Workspace, Build Now, Configure, Delete Project, Robot Results, GitHub, and Rename. The main content area displays the 'UAT' pipeline status as 'Lab 8.5'. Below this, there is a section for 'Latest Robot Results' which states 'No results available yet.' and a 'Permalinks' section with a list of build links. At the bottom left, a 'Builds' panel shows a list of recent builds, including build #8 at 2:44 PM and build #7 at 2:43 PM, both marked with red 'X' icons indicating failure.

## Lab Worksheet

The screenshot shows the Jenkins web interface. The breadcrumb navigation is Dashboard > UAT > #8 > Console Output. The left sidebar contains links for Status, Changes, Console Output (selected), Edit Build Information, Delete build '#8', Timings, and Previous Build. The main area is titled 'Console Output' with a red error icon. It shows the build was started by user 'Aekkarat chaisong' and is running as SYSTEM. The build is in workspace /var/jenkins\_home/workspace/UAT. The console output shows a shell command being executed, which fails because a file named 'robot' is not found. The error message is 'robot: not found'. The build step 'Execute shell' is marked as failure. The publisher started, but the build failed due to a 'hudson.AbortException: No files found in path /var/jenkins\_home/workspace/UAT with configured filemask: output.xml'. The stack trace shows the error occurred in the RobotPublisher plugin.

```
Started by user Aekkarat chaisong
Running as SYSTEM
Building in workspace /var/jenkins_home/workspace/UAT
[UAT] $ /bin/sh -xe /tmp/jenkins8554889590164521666.sh
+ robot login_test/lab7.robot
/tmp/jenkins8554889590164521666.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:
Failed!
hudson.AbortException: No files found in path /var/jenkins_home/workspace/UAT with configured filemask: output.xml
    at PluginClassLoader for robot//udson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:81)
    at PluginClassLoader for robot//udson.plugins.robot.RobotParser$RobotParserCallable.invoke(RobotParser.java:52)
    at hudson.FilePath.act(FilePath.java:1234)
    at hudson.FilePath.act(FilePath.java:1217)
    at PluginClassLoader for robot//udson.plugins.robot.RobotParser.parse(RobotParser.java:48)
    at PluginClassLoader for robot//udson.plugins.robot.RobotPublisher.parse(RobotPublisher.java:262)
    at PluginClassLoader for robot//udson.plugins.robot.RobotPublisher.perform(RobotPublisher.java:286)
    at hudson.tasks.BuildStepCompatibilityLayer.perform(BuildStepCompatibilityLayer.java:80)
    at hudson.tasks.BuildStepMonitor$1.perform(BuildStepMonitor.java:20)
    at hudson.model.AbstractBuild$AbstractBuildExecution.perform(AbstractBuild.java:818)
    at hudson.model.AbstractBuild$AbstractBuildExecution.performAllBuildSteps(AbstractBuild.java:767)
```