

ชื่อ-นามสกุล นายฐาปกรณ์ เมืองจันทร์ รหัสนักศึกษา 653380194-3 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

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

Lab Worksheet

- ย้ายตำแหน่งปัจจุบันไปที่ Lab8_1 เพื่อใช้เป็น Working directory
- ป้อนคำสั่ง \$ docker pull busybox หรือ \$ sudo docker pull busybox สำหรับกรณีที่ติดปัญหา Permission denied (หมายเหตุ: BusyBox เป็น software suite ที่รองรับคำสั่งบางอย่างบน Unix - <https://busybox.net>)
- ป้อนคำสั่ง \$ docker images

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

```
Mode                LastWriteTime         Length Name
----                -
d-----          1/23/2025   9:56 AM                Lab8_1

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

What's next:
  View a summary of image vulnerabilities and recommendations →docker scout quickview busybox
PS C:\Users\computer\Lab8_1> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
busybox       latest   af4709625109   3 months ago   4.27MB
PS C:\Users\computer\Lab8_1>
```

- (1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร
repository ที่ download มา
- (2) Tag ที่ใช้บ่งบอกถึงอะไร
บ่งบอกถึงเวอร์ชันของ repository

5. ป้อนคำสั่ง \$ docker run busybox
6. ป้อนคำสั่ง \$ docker run -it busybox sh
7. ป้อนคำสั่ง ls

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

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

```
Terminal
PS C:\Users\computer\Lab8_1> docker run busybox
PS C:\Users\computer\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 03:06 .
drwxr-xr-x 1 root root      4096 Jan 23 03:06 ..
-rwxr-xr-x 1 root root        0 Jan 23 03:06 .dockerenv
drwxr-xr-x 2 root root    12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root      368 Jan 23 03:06 dev
drwxr-xr-x 1 root root      4096 Jan 23 03:06 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 329 root root        0 Jan 23 03:06 proc
drwx----- 1 root root      4096 Jan 23 03:06 root
dr-xr-xr-x 11 root root        0 Jan 23 03:06 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\Lab8_1> docker run busybox echo "Hello ฐานรณเมืองจันท์ from busybox"
Hello ฐานรณเมืองจันท์ from busybox
PS C:\Users\computer\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS          PORTS          NAMES
8f6de6b316ec   busybox   "echo 'Hello ฐานรณเมืองจันท์'" 13 seconds ago   Exited (0) 12 seconds ago           compassionate_beaver
2f383ce61afd   busybox   "sh"                     About a minute ago   Exited (0) About a minute ago           nostalgic_fernat
2277886db552   busybox   "sh"                     2 minutes ago     Exited (0) 2 minutes ago           intelligent_volhard
PS C:\Users\computer\Lab8_1>
```

- (1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
เข้าไปใน container และใช้งาน shell เพื่อ interact กับ container ได้ เปิดโหมด interactive
- (2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร
สถานะของ Docker container ในปัจจุบัน

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

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

```
Terminal
drwxr-xr-x 1 root root 4096 Jan 23 03:06 ..
-rwxr-xr-x 1 root root 0 Jan 23 03:06 .dockerenv
drwxr-xr-x 2 root root 12288 Sep 26 21:31 bin
drwxr-xr-x 5 root root 360 Jan 23 03:06 dev
drwxr-xr-x 1 root root 4096 Jan 23 03:06 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 329 root root 0 Jan 23 03:06 proc
drwx----- 1 root root 4096 Jan 23 03:06 root
dr-xr-xr-x 11 root root 0 Jan 23 03:06 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\Lab8_1> docker run busybox echo "Hello ภาณุวัฒน์เมืองจันทร์ from busybox"
Hello ภาณุวัฒน์เมืองจันทร์ from busybox
PS C:\Users\computer\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
8f6de6b316ec   busybox   "echo 'Hello ภาณุวัฒน์'" 13 seconds ago Exited (0)    12 seconds ago           compassionate_beaver
2f383ce61afd   busybox   "sh"                    About a minute ago Exited (0)    About a minute ago       nostalgic_fernat
2277886db552   busybox   "sh"                    2 minutes ago Exited (0)    2 minutes ago           intelligent_volhard
PS C:\Users\computer\Lab8_1> docker rm ^C
PS C:\Users\computer\Lab8_1> docker rm 8f6de6b316ec
8f6de6b316ec
PS C:\Users\computer\Lab8_1> docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS          NAMES
2f383ce61afd   busybox   "sh"                    4 minutes ago Exited (0)    3 minutes ago           nostalgic_fernat
2277886db552   busybox   "sh"                    4 minutes ago Exited (0)    4 minutes ago           intelligent_volhard
PS C:\Users\computer\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"

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

Lab Worksheet

สำหรับเครื่องที่ใช้ระบบปฏิบัติการ MacOS หรือ Linux บนหน้าต่าง Terminal และป้อนคำสั่งต่อไปนี้

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

```
FROM busybox
```

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

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

```
EOF
```

หรือใช้คำสั่ง

```
$ touch Dockerfile
```

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

```
$ docker build -t <ชื่อ Image> .
```

6. เมื่อ Build สำเร็จแล้ว ให้ทำการรัน Docker image ที่สร้างขึ้นในขั้นตอนที่ 5

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

```
PS D:\2-2567\Lab8_2> docker build -t my-image -f Dockerfile.swp .
>>
[+] Building 5.3s (6/6) FINISHED                                docker:desktop-linux
=> [internal] load build definition from Dockerfile.swp        0.0s
=> => transferring dockerfile: 211B                             0.0s
=> [internal] load metadata for docker.io/library/busybox:latest 4.0s
=> [auth] library/busybox:pull token for registry-1.docker.io  0.0s
=> [internal] load .dockerignore                               0.0s
=> => transferring context: 2B                                    0.0s
=> [1/1] FROM docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1 0.7s
=> => resolve docker.io/library/busybox:latest@sha256:a5d0ce49aa801d475da48f8cb163c354ab95cab073cd3c138bd458fc8257fbf1 0.0s
=> => sha256:9c0abc9c5bd3a7854141800ba1f4a227baa88b11b49d8207eadc483c3f2496de 2.16MB / 2.16MB 0.6s
=> exporting to image                                          0.9s
=> => exporting layers                                           0.0s
=> => exporting manifest sha256:0adcd1c9f505409d3cf9077f2d202eb9e83cf118a01371268211a8c10cabf1fd 0.0s
=> => exporting config sha256:860d0bf0830174215935c3dc04c74927b7dc695d328978f3e5bfa3bfc384fa91 0.0s
=> => exporting attestation manifest sha256:37ed3bec813f6fb22e6a25920f26675d2e99bb02c720f82225a52186e99e6841 0.0s
=> => exporting manifest list sha256:ee18c5f9556854998835e8d7fa7d76345c001682dc94b0cb540f351dcb4749c8 0.0s
=> => naming to docker.io/library/my-image:latest              0.0s
=> => unpacking to docker.io/library/my-image:latest           0.7s
```

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

Lab Worksheet

```
3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)
- MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will be used (line 2)
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)
PS D:\2-2567\Lab8_2> docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
my-image       latest    ee18c5f95568   4 months ago   6.56MB
PS D:\2-2567\Lab8_2>
```

- (1) คำสั่งที่ใช้ในการ run คือ
Docker run <ชื่อ Image>
- (2) Option -t ในคำสั่ง \$ docker build ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป
-t ในคำสั่ง docker build ช่วยกำหนดชื่อและ tag ให้กับ Docker image ที่จะถูกสร้างขึ้น

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 บันทึกคำสั่งต่อไปนี้ลงในไฟล์โดยใช้ 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

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

Lab Worksheet

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

```
REPOSITORY          TAG         IMAGE ID        CREATED         SIZE
jajajajochx47/lab8  latest     d369f20d1ae3   4 months ago   6.56MB
my-image             latest     ee18c5f95568   4 months ago   6.56MB
PS D:\2-2567\Lab8_3> docker run -t jajajajochx47/lab8
"นายฐาปกรณ์เมืองจันทร์ 653380194-3 sec.3"
```

6. ทำการ Push ตัว Docker image ไปไว้บน Docker Hub โดยการใช้คำสั่ง

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

ในกรณีที่ติดปัญหาไม่ได้ Login ไว้ก่อน ให้ใช้คำสั่งต่อไปนี้ เพื่อ Login ก่อนทำการ Push

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

CP353004/SC313 004 Software Engineering (2/2567)

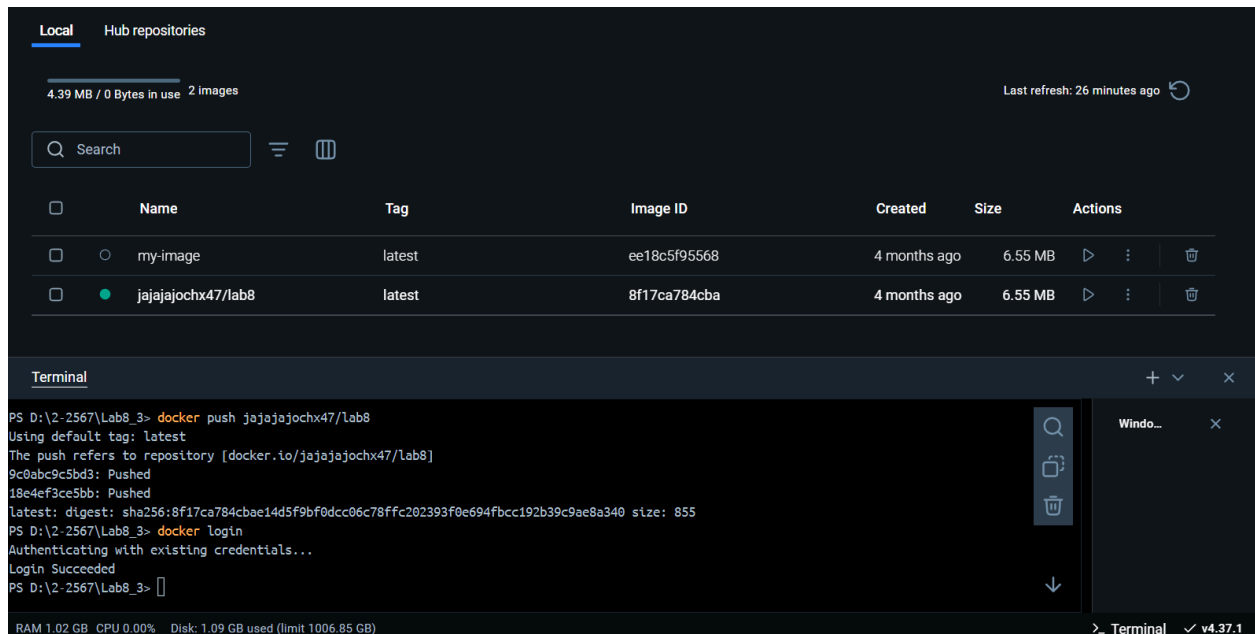
ผศ.ดร.ชิตสุธา สุ่มเล็ก

Lab Worksheet

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

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

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



แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

1. เปิด Command line หรือ Terminal จากนั้นสร้าง Directory ชื่อ Lab8_4

CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

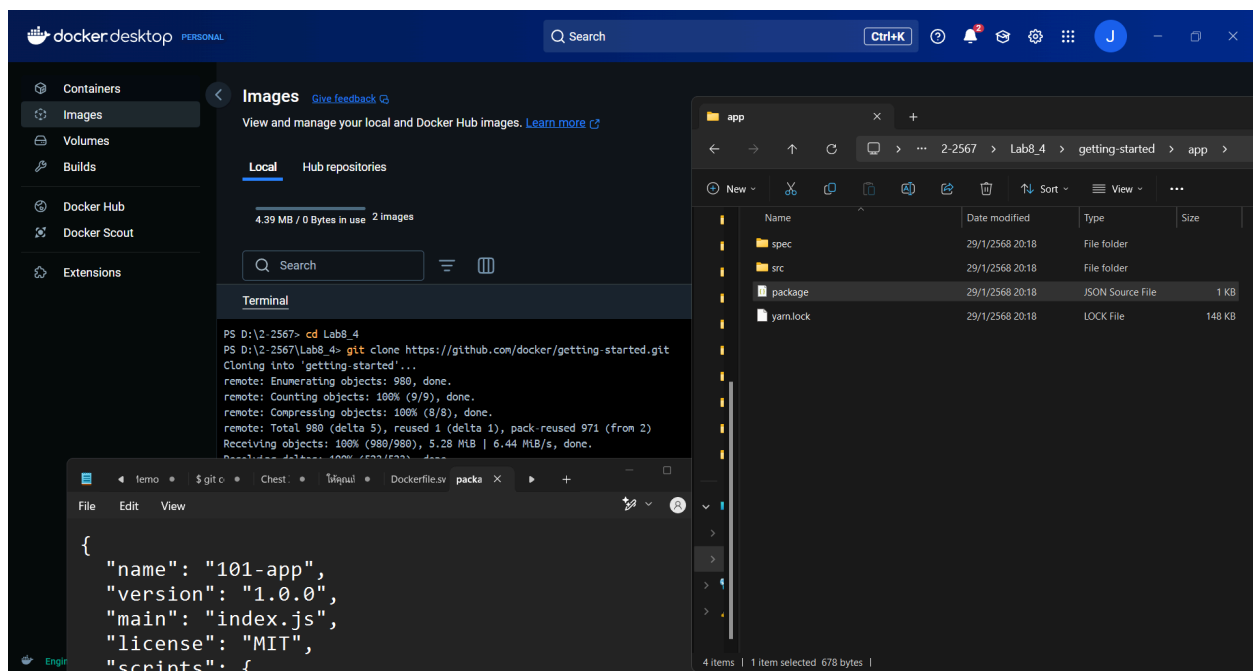
Lab Worksheet

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 (ทั้งหน้าตาและทุกหน้าต่างที่เกี่ยวข้อง) แสดงที่อยู่ของ Source code ที่ Clone มาและเนื้อหาของไฟล์ package.json



4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปไฟล์
- ```
FROM node:18-alpine
WORKDIR /app
COPY ..
```

## CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

### Lab Worksheet

RUN yarn install --production

CMD ["node", "src/index.js"]

EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น myapp\_รหัสสนศ. ไม่มีขีด

\$ docker build -t <myapp\_รหัสสนศ. ไม่มีขีด> .

**[Check point#8] Capture หน้าจอ (ทั้งหน้าต่างและทุกหน้าต่างที่เกี่ยวข้อง) แสดงคำสั่งและผลลัพธ์ที่ได้ทางหน้าจอ**

```
PS D:\2-2567\Lab8_4> docker build -t myapp_6533801943 -f D:\2-2567\Lab8_4\getting-started\app\Dockefile.swp .
[+] Building 8.8s (10/10) FINISHED
=> [internal] load build definition from Dockerfile.swp
=> => transferring dockerfile: 160B
=> [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@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25
=> => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B
=> => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB
=> => sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7c0da02525e63167c 40.01MB / 40.01MB
=> => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB
=> => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0
=> => extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7c0da02525e63167c
RAM 0.99 GB CPU 0.08% Disk: 1.31 GB used (limit 1006.85 GB)
```

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**

หมายเหตุ: นศ.สามารถทดลองเล่น Web application ที่ทำงานอยู่ได้

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

## Lab Worksheet

Images [Give feedback](#)

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

| <input type="checkbox"/> | Name               | Tag    | Image ID     | Created        | Size      | Actions |
|--------------------------|--------------------|--------|--------------|----------------|-----------|---------|
| <input type="checkbox"/> | my-image           | latest | ee18c5f95568 | 4 months ago   | 6.55 MB   |         |
| <input type="checkbox"/> | jajajajochx47/lab8 | latest | 8f17ca784cba | 4 months ago   | 6.55 MB   |         |
| <input type="checkbox"/> | myapp_6533801943   | latest | df40c98e5c69 | 57 seconds ago | 341.99 MB |         |

Showing 3 items

Walkthroughs

1 FROM node

2 RUN mkdir -p

3 WORKDIR /app

How do I run a container?

Run Docker Hub images

Terminal

```
=> => exporting manifest sha256:7ee8c8f2cd82efb63d350cc79a411f8f6f73da35be2f062e95a629c99e036478 0.0s
=> => exporting config sha256:a3ac0431763d412a7f50ef892334270e3f4898219d7e6ad614de74e183427331 0.0s
=> => exporting attestation manifest sha256:dbb7b9a051abbac67e353cc02216c620380c9793ee08f55ad64b01a41ca6279d 0.0s
=> => exporting manifest list sha256:df40c98e5c6966de8c8ec02dd464282cdcc8756c3a227095d8c1d57033494253 0.0s
=> => naming to docker.io/library/myapp_6533801943:latest 0.0s
=> => unpacking to docker.io/library/myapp_6533801943:latest 2.7s
PS D:\2-2567\Lab8_4> docker run -dp 3000:3000 myapp_6533801943
>>
72b57019a52139ee4651a4f32e969e464210e35bffa933f0482dd7c72fdbcd329
PS D:\2-2567\Lab8_4>
```

localhost:3000

No items yet! Add one above!

## CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

### 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</p>`

b. Save ไฟล์ให้เรียบร้อย

9. ทำการ Build Docker image โดยใช้คำสั่งเดียวกันกับข้อ 5

10. Start และรัน Container ตัวใหม่ โดยใช้คำสั่งเดียวกันกับข้อ 6

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

```
7/25/019a52139ee4651adocker build -t myapp_6533801943 -f D:\2-2567\Lab8_4\getting-started\app\Dockerfile.swp .
[+] Building 24.1s (10/10) FINISHED
=> [internal] load build definition from Dockerfile.swp
=> => transferring dockerfile: 190B
=> [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
=> [internal] load build context
=> => exporting attestation manifest sha256:0ce4e8bd43e0f5b8d24b6ea6804c76365a01db39162bcf6b81f7726882b4def
=> => exporting manifest list sha256:9ce2b0544a49409d189598e9e64efccfaf4238fe162a97da13209f2b35d24e9f
=> => naming to docker.io/library/myapp_6533801943:latest
=> => unpacking to docker.io/library/myapp_6533801943:latest
PS D:\2-2567\Lab8_4> docker run -dp 3000:3000 myapp_6533801943
>>
1b3a0158d5666e5f9a1cee4ec8b08890b13ea891249c784b5b9f2a8e5a06f6e9
docker: Error response from daemon: driver failed programming external connectivity on endpoint happy_swanson (7c9d01e598d524b187d04857cfc75c1af
c69e1c841315488eb2489860b0b6809): Bind for 0.0.0.0:3000 failed: port is already allocated.
PS D:\2-2567\Lab8_4>
```

(1) Error ที่เกิดขึ้นหมายความว่าอย่างไร และเกิดขึ้นเพราะอะไร  
พอร์ต 3000 มีการใช้งานอยู่แล้ว

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

a. ผ่าน Command line interface

## CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

### Lab Worksheet

- i. ใช้คำสั่ง \$ docker ps เพื่อดู Container ID ที่ต้องการจะลบ
    - ii. Copy หรือบันทึก Container ID ไว้
    - 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) แสดงผลลัพธ์ที่ได้บน Browser และ Dashboard ของ Docker desktop

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตส์ฐา สุ่มเล็ก

## Lab Worksheet

### Containers [Give feedback](#)

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

0.00% / 1200% (12 CPUs available) 18.57MB / 3.25GB

Only show running containers

| <input type="checkbox"/> | Name                | Container ID | Image              | Port(s)   | CPU (%) | Last started   | Actions |
|--------------------------|---------------------|--------------|--------------------|-----------|---------|----------------|---------|
| <input type="checkbox"/> | goofy_keller        | b3948940eeb8 | jajajajochx47/lab8 |           | 0%      | 49 minutes ago |         |
| <input type="checkbox"/> | eager_faraday       | 822258f1ae4a | jajajajochx47/lab8 |           | 0%      | 47 minutes ago |         |
| <input type="checkbox"/> | awesome_lamarr      | d8ad79f4f47b | jajajajochx47/lab8 |           | 0%      | 46 minutes ago |         |
| <input type="checkbox"/> | heuristic_northcutt | e8e6172e7f18 | myapp_6533801943   | 3000:3000 | 0%      | 22 seconds ago |         |

Showing 4 items

#### Terminal

RAM 1.61 GB CPU 0.33% Disk 1.80 GB used (limit 1006.85 GB)

> Terminal v4.37.1

2.9s

>

PS D:\2-2567\Lab8\_4> docker run -dp 3000:3000 myapp\_6533801943

>

1b3a0158d5666e5f9a1cee4ec8b08890b13ea891249c784b5b9f2a8c5a06f8e9

docker: Error response from daemon: driver failed programming external connectivity on endpoint happy\_swanson (7c9d01c598d524b187d04857cfc75c1afc69e1c841315488eb2489860b0b6809): Bind for 0.0.0.0:3000 failed: port is already allocated.

PS D:\2-2567\Lab8\_4> docker run -dp 3000:3000 myapp\_6533801943

>

e8e6172e7f18602f294ec06304768f5f8eb3284a6ad8f022008c789bc40c18a0

PS D:\2-2567\Lab8\_4>

localhost:3000

New Item

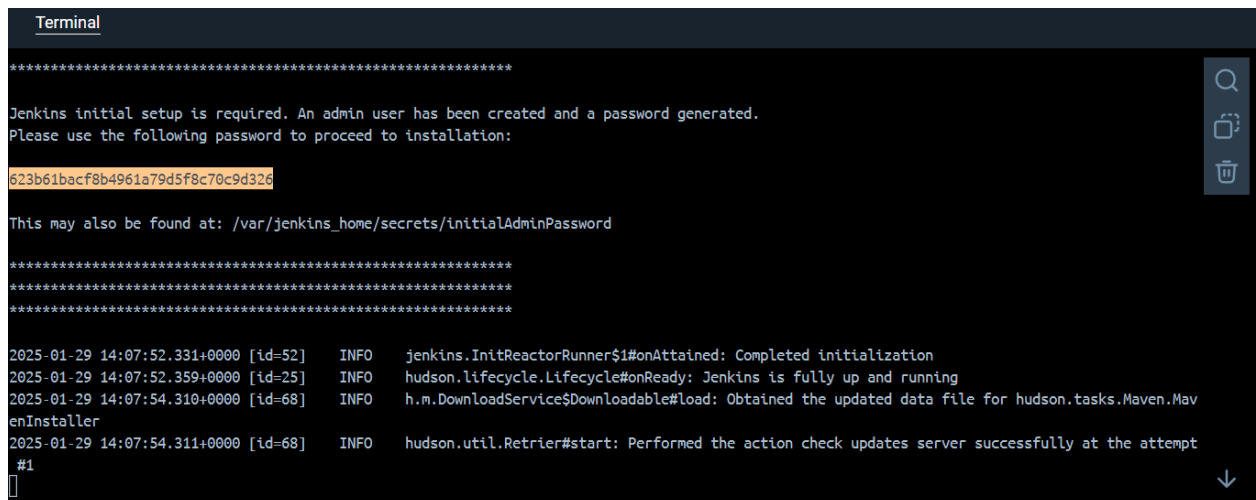
Add Item

There is no TODO item. Please add one to the list. By สุวิมลกร เมืองจันทร์ 653380194-3

14

### 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**



```
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:

623b61bacf8b4961a79d5f8c70c9d326

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

2025-01-29 14:07:52.331+0000 [id=52] INFO jenkins.InitReactorRunner$1#onAttained: Completed initialization
2025-01-29 14:07:52.359+0000 [id=25] INFO hudson.lifecycle.Lifecycle#onReady: Jenkins is fully up and running
2025-01-29 14:07:54.310+0000 [id=68] INFO h.m.DownloadService$Downloadable#load: Obtained the updated data file for hudson.tasks.Maven.MavenInstaller
2025-01-29 14:07:54.311+0000 [id=68] INFO hudson.util.Retrier#start: Performed the action check updates server successfully at the attempt #1
```

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

**[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า**

## CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

### Lab Worksheet

#### Getting Started

Username

Thapakorn\_1943

Password

.....

Confirm password

.....

Full name

Thapakorn Muangchan

E-mail address

thapakon.m@kkumail.com

Jenkins 2.479.3

[Skip and continue as admin](#) [Save and Continue](#)

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

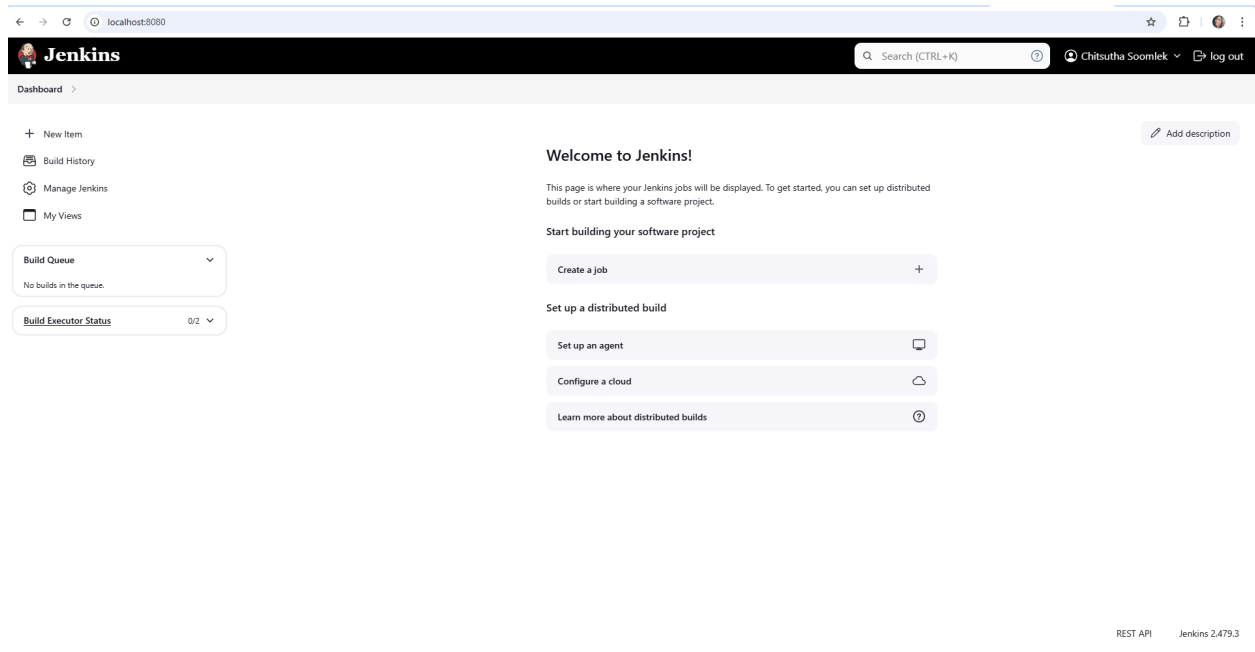
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ



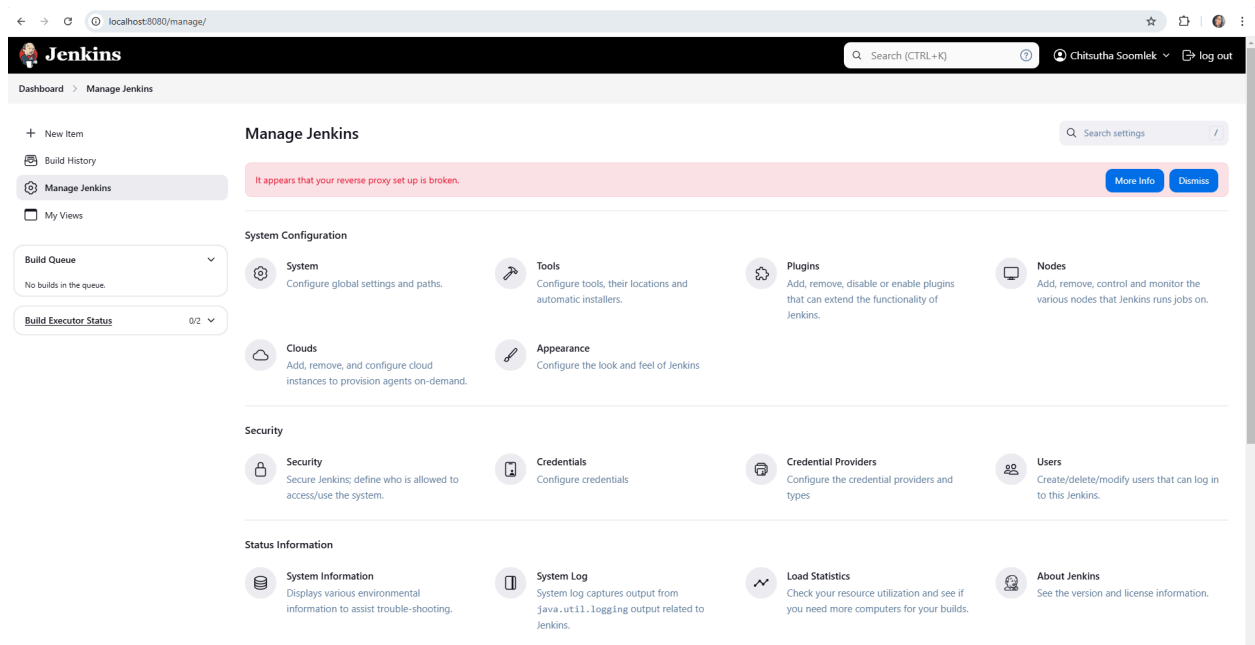
# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

## Lab Worksheet



## 9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



# CP353004/SC313 004 Software Engineering (2/2567)

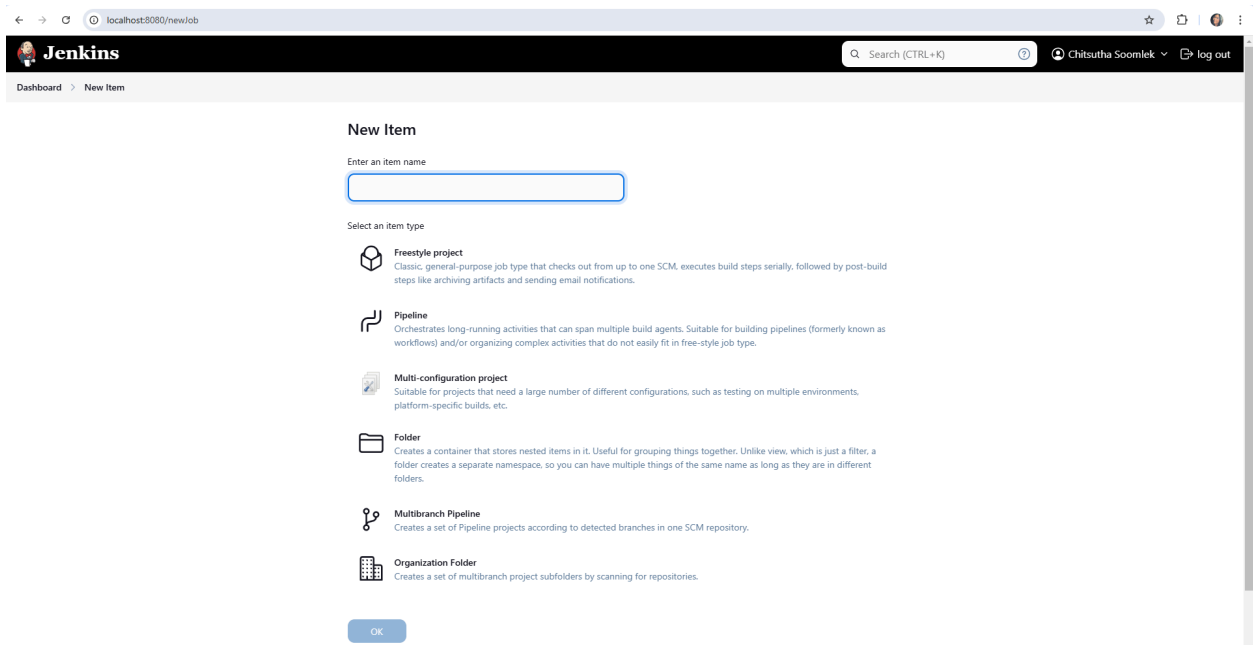
ผศ.ดร.ชิตสุธา สุ่มเล็ก

## 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

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

## Lab Worksheet

**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)

**[Check point#14]** Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้

General

Enabled 

Description

Lab8.5

Plain text

Preview

☐ Discard old builds ?

☒ GitHub project

Project url ?

https://github.com/Thapakorn194-3/Lab8/

Advanced ▾

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

## Lab Worksheet

### Build Triggers

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

☐ Build after other projects are built ?

☒ Build periodically ?

Schedule ?

H/15 \* \* \* \*

Would last have run at Wednesday, January 29, 2025 at 3:05:05 PM Coordinated Universal Time; would next run at Wednesday, January 29, 2025 at 3:20:05 PM Coordinated Universal Time.

☐ GitHub hook trigger for GITScm polling ?

☐ Poll SCM ?

### Build Steps

≡ Execute shell ?

Command

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

```
cd WebDemo-master/login_tests
robot UAT-Lab7-001.robot
robot UAT-Lab7-002.robot
```

Advanced ▾

Add build step ▾

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

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตสุธา สุ่มเล็ก

## Lab Worksheet

### Build Steps

≡ Execute shell ?

Command

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

```
cd WebDemo-master/login_tests
robot UAT-Lab7-001.robot
robot UAT-Lab7-002.robot
```

Advanced ▾

Add build step ▾

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

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตส์ฐา สุ่มเล็ก

## Lab Worksheet

Status

</> Changes

Console Output

Edit Build Information

Delete build '#2'

Timings

Git Build Data

Previous Build

Console Output

Started by user Thapakorn Muangchan  
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/Thapakorn194-3/Lab8.git # timeout=10  
Fetching upstream changes from https://github.com/Thapakorn194-3/Lab8.git  
> git --version # timeout=10  
> git --version # 'git version 2.39.5'  
> git fetch --tags --force --progress -- https://github.com/Thapakorn194-3/Lab8.git +refs/heads/\*:refs/remotes/origin/\* # timeout=10  
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10  
Checking out Revision c526934d661777e686415edb94a54091a1f12e4b (refs/remotes/origin/main)  
> git config core.sparsecheckout # timeout=10  
> git checkout -f c526934d661777e686415edb94a54091a1f12e4b # timeout=10  
Commit message: "Add files via upload"  
> git rev-list --no-walk c526934d661777e686415edb94a54091a1f12e4b # timeout=10  
[UAT] \$ /bin/sh -xe /tmp/jenkins16276196406339097073.sh  
+ cd WebDemo-master/login\_tests  
+ robot UAT-Lab7-001.robot  
/tmp/jenkins16276196406339097073.sh: 3: robot: not found

Dashboard > UAT > Workspace of UAT on Built-In Node

Status

</> Changes

Workspace

Wipe Out Current Workspace

Build Now

Configure

Delete Project

Robot Results

GitHub

Rename

Workspace of UAT on Built-In Node

UAT /

.git

WebDemo-master

(all files in zip)

Builds

Filter

# CP353004/SC313 004 Software Engineering (2/2567)

ผศ.ดร.ชิตส์ฐา สุ่มเล็ก

## Lab Worksheet

### Build History of Jenkins

| S                                                                                 | Build  | Time Since <sup>1</sup> | Status                  |                                                                                     |
|-----------------------------------------------------------------------------------|--------|-------------------------|-------------------------|-------------------------------------------------------------------------------------|
|  | UAT #8 | 3.3 sec                 | broken for a long time  |  |
|  | UAT #7 | 23 sec                  | broken for a long time  |  |
|  | UAT #6 | 23 sec                  | broken for a long time  |  |
|  | UAT #5 | 25 sec                  | broken for a long time  |  |
|  | UAT #4 | 26 sec                  | broken for a long time  |  |
|  | UAT #3 | 2 min 28 sec            | broken for a long time  |  |
|  | UAT #2 | 13 min                  | broken for a long time  |  |
|  | UAT #1 | 15 min                  | broken since this build |  |

Icon: S M L

ooo

