Lab Worksheet

ชื่อ-นามสกุล นางสาวมุตธิตา จันดาวงค์ รหัสนักศึกษา 653380279-5 Section 2

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) แสดงผลลัพธ์ที่ได้พร้อมกับตอบคำถามต่อไปนี้

```
PS C:\Users\acer> cd C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> mkdir Lab8_1


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/22/2025   2:23 PM                Lab8_1


PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> cd lab8_1
```

(1) สิ่งที่อยู่ภายใต้คอลัมน์ Repository คืออะไร ชื่อ Docker image

(2) Tag ที่ใช้บ่งบอกถึงอะไร เวอร์ชั่นของ Docker image

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\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








                                                                                   What's next:
                               What's next:                                                             What's next:
                                                     What's next:                                                      Wh
at's next:                                                          What's next:
         What's next:                                                       What's next:
              What's next:                                                      View a summary of image vulnerabiPS C:\Users\acer\
Documents\Year3\SoftwereEngineer\Lab8\lab8_1> do
REPOSITORY       TAG       IMAGE ID       CREATED        SIZE


busybox          latest    af4709625109   3 months ago   4.27MB
```

5. ป้อนคำสั่ง $ docker run busybox
6. ป้อนคำสั่ง $ docker run -it busybox sh
7. ป้อนคำสั่ง ls
8. ป้อนคำสั่ง ls -la
9. ป้อนคำสั่ง exit
10. ป้อนคำสั่ง $ docker run busybox echo “Hello ชื่อและนามสกุลของนักศึกษา from busybox”
11. ป้อนคำสั่ง $ docker ps -a

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

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\lab8_1> docker run busybox
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\lab8_1> docker run -it                                        >
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\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 22 07:37 .
drwxr-xr-x    1 root     root          4096 Jan 22 07:37 ..
Hello Muttita Chandawong from busybox
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\lab8_1> docker ps -a
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS                       PORTS      NAMES
5f75824b2c62   busybox                  "echo 'Hello Muttita…"   12 seconds ago   Exited (0) 11 seconds ago               heuristic_montalcini
2fce203a0795   busybox                  "sh"                     2 minutes ago    Exited (0) 53 seconds ago               optimistic_germain
174dc7fb4cea   busybox                  "sh"                     3 minutes ago    Exited (0) 3 minutes ago                condescending_hermann
758c1d6faa96   ailabexam_image          "jupyter lab --ip=0.…"   3 months ago     Exited (0) 3 months ago                 zealous_tharp
61157b3173a0   ailabexam_image:latest   "jupyter lab --ip=0.…"   3 months ago     Exited (255) 3 months ago   8000/tcp    reverent_wozniak
6d091b2ae8d1   ailabexam_image          "jupyter lab --ip=0.…"   3 months ago     Exited (0) 3 months ago                 blissful_booth
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\lab8_1>
```

(1) เมื่อใช้ option -it ในคำสั่ง run ส่งผลต่อการทำงานของคำสั่งอย่างไรบ้าง อธิบายมาพอสังเขป

เปิดใช้งาน interactive mode (-i) เพื่อให้ป้อนข้อมูลเข้าได้

สร้าง terminal จำลอง (-t) สำหรับโต้ตอบกับคอนเทนเนอร์

-it ทำให้คอนเทนเนอร์รองรับการโต้ตอบผ่าน terminal ได้เหมือนการใช้งานเครื่องในโหมดปกติ

(2) คอลัมน์ STATUS จากการรันคำสั่ง docker ps -a แสดงถึงข้อมูลอะไร

STATUS แสดงสถานะคอนเทนเนอร์ เช่น **รันอยู่**(Running), **หยุด**(Exited), **หยุดชั่วคราว**(Paused) พร้อมบอกระยะเวลาที่เปลี่ยนสถานะนั้น

12. ป้อนคำสั่ง $ docker rm <container ID ที่ต้องการลบ>

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

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_> docker ps -a
CONTAINER ID   IMAGE                    COMMAND
  CREATED              STATUS                          PORTS      NAMES
511d4effabd9   busybox                  "echo 'Hello Muttita…"   49 seconds ago       Exited (0) 49 seconds ago
   festive_wozniak
2783bc9a7803   busybox                  "sh"
  About a minute ago   Exited (0) About a minute ago
   inspiring_hertz
05a478d0e484   busybox                  "sh"
  2 minutes ago        Exited (0) 2 minutes ago
   thirsty_carson
5f75824b2c62   busybox                  "echo 'Hello Muttita…"   5 days ago           Exited (0) 5 days ago
   heuristic_montalcini
2fce203a0795   busybox                  "sh"
  5 days ago           Exited (0) 5 days ago
   optimistic_germain
758c1d6faa96   ailabexam_image          "jupyter lab --ip=0.…"   3 months ago         Exited (0) 3 months ago                  zealous_tharp
61157b3173a0   ailabexam_image:latest   "jupyter lab --ip=0.…"   3 months ago         Exited (255) 3 months ago      8000/tcp  reverent_woznia
k
6d091b2ae8d1   ailabexam_image          "jupyter lab --ip=0.…"   3 months ago         Exited (0) 3 months ago                  blissful_booth
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_1>^C
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_1> docker rm 511d4effabd9
511d4effabd9
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\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

หรือใช้คำสั่ง

$ touch Dockerfile

แล้วใช้ Text Editor ในการใส่เนื้อหาแทน

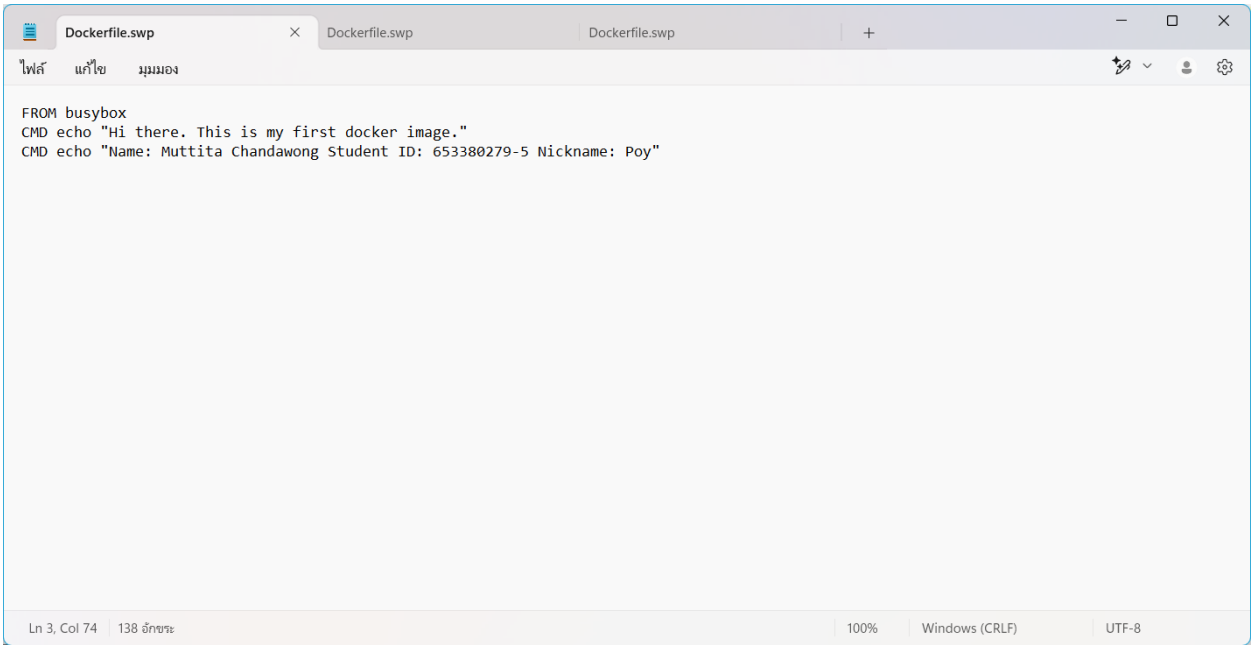
5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้

   $ 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\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_1> cd ..
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> mkdir Lab8_2


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/27/2025  10:00 PM                Lab8_2


PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> cd Lab8_2
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2> New-Item -ItemType File -Name Dockerfile.swp


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/27/2025  10:22 PM              0 Dockerfile.swp
```




```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2> docker build -t myFirstDocker .
[+] Building 0.0s (0/0)                                                                    docker:desktop-linux
ERROR: invalid tag "myFirstDocker": repository name must be lowercase
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2> docker build -t myfirstdocker .
[+] Building 0.1s (1/1) FINISHED                                                           docker:desktop-linux
 => [internal] load build definition from Dockerfile                                                   0.0s
 => => transferring dockerfile: 2B                                                                     0.0s
ERROR: failed to solve: failed to read dockerfile: open Dockerfile: no such file or directory
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2> docker build -t myfirstdocker -f Dockerfile.swp .
 => [internal] load build definition from Dockerfile.swp                                               0.0s
 => => transferring dockerfile: 183B                                                                   0.0s
 => WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 2)   0.0s
 => WARN: MultipleInstructionsDisallowed: Multiple CMD instructions should not be used in the same stage because only the last one will b  0.0s
 => WARN: JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 3)   0.0s
 => [internal] load metadata for docker.io/library/busybox:latest                                      0.0s
 => [internal] load .dockerignore                                                                      0.0s
 => => transferring context: 2B                                                                        0.0s
 => [1/1] FROM docker.io/library/busybox:latest                                                        0.0s
 => exporting to image                                                                                 0.0s
 => => exporting layers                                                                                0.0s
 => => writing image sha256:f664f2ccd496bd68e9a24f887b8fb03508d8ed0e0297ec25ea27c0d8af41d0f1           0.0s
 => => naming to docker.io/library/myfirstdocker                                                       0.0s

3 warnings found (use docker --debug to expand):
- JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 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)
```

(1) คำสั่งที่ใช้ในการ run คือ

docker run <ชื่อ image> ในที่นี้คือ myfirstdocker

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_2> docker run myfirstdocker
Name: Muttita Chandawong Student ID: 653380279-5 Nickname: Poy
```

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

Option -t ใช้ตั้งชื่อและแท็ก Docker image เพื่อให้อ้างอิงและจัดการ image ได้ง่าย

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
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

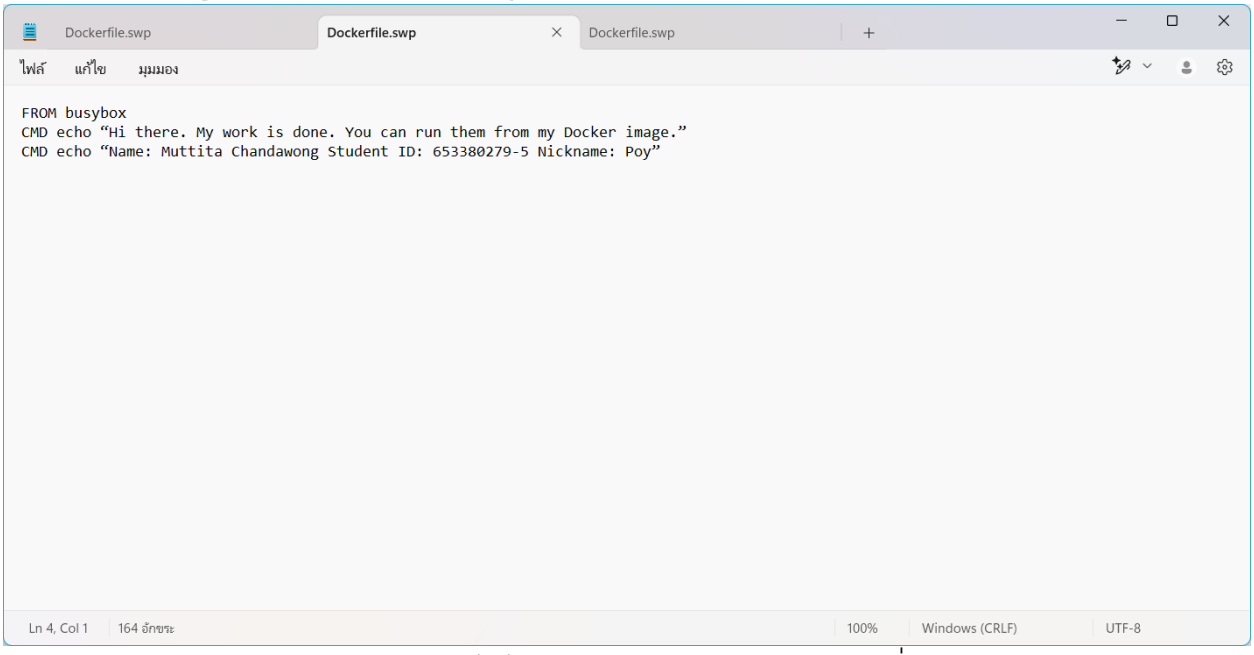
5. ทำการรัน Docker image บน Container ในเครื่องของตัวเองเพื่อทดสอบผลลัพธ์ ด้วยคำสั่ง

   $ docker run <username ที่ลงทะเบียนกับ Docker Hub>/lab8

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

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> mkdir Lab8_3
```

```
Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/27/2025  11:06 PM                Lab8_3


PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> cd Lab8_3
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_3> New-Item -ItemType File -Name Dockerfile.swp


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_3


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/27/2025  11:08 PM              0 Dockerfile.swp


PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_> docker build -t muttita/lab8 -f Dockerfile.swp .
 => [internal] load .dockerignore                          0.0s
 => => transferring context: 2B                            0.0s
 => CACHED [1/1] FROM docker.io/library/busybox:latest     0.0s
 => exporting to image                                     0.0s
 => => exporting layers                                    0.0s
 => => writing image sha256:de7b4e11e52719108eae2486b43b   0.0s
 => => naming to docker.io/muttita/lab8                    0.0s

 3 warnings found (use docker --debug to expand):
 - JSONArgsRecommended: JSON arguments recommended for CMD to prevent unintended behavior related to OS signals (line 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)

What's next:
    View a summary of image vulnerabilities and recommendations →docker scout quickview
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_> docker run muttita/lab8
“Name: Muttita Chandawong Student ID: 653380279-5 Nickname: Poy”
```



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

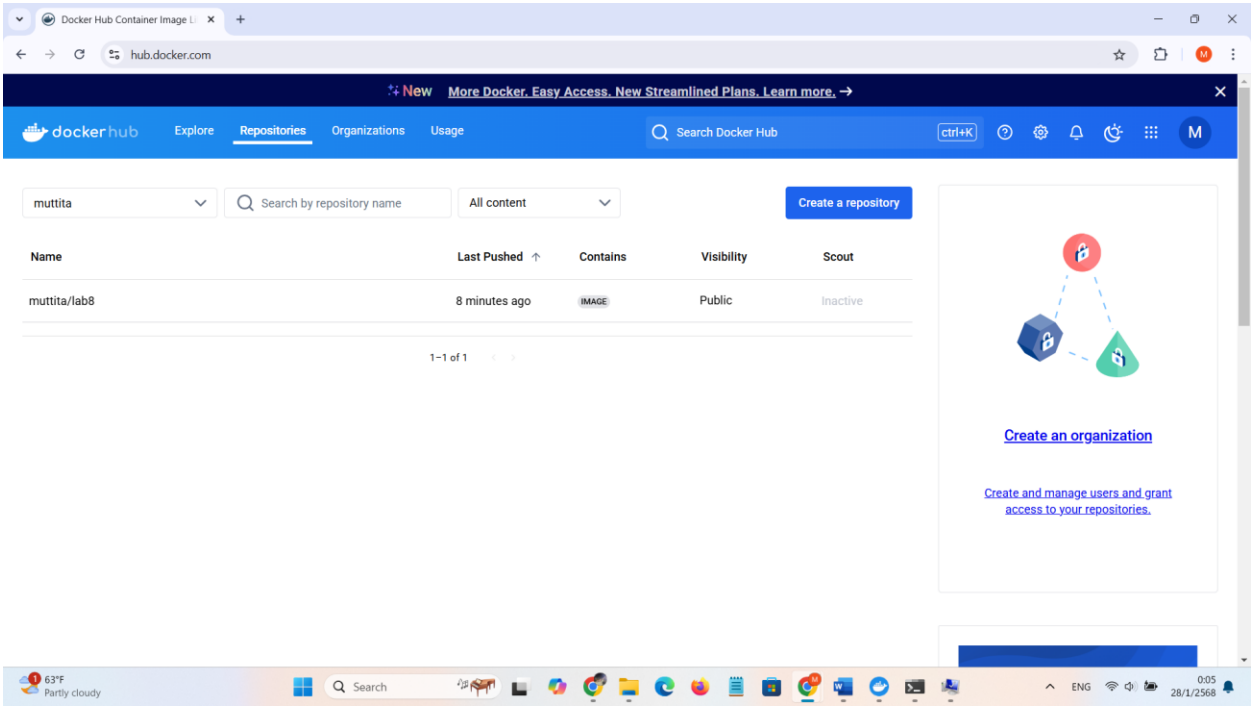
   $ docker login แล้วป้อน Username และ Password ตามที่ระบุใน Command prompt หรือใช้คำสั่ง

   $ 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)

```
An image does not exist locally with the tag: muttia/lab8
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_> docker push muttita/lab8
Using default tag: latest
The push refers to repository [docker.io/muttita/lab8]
59654b79daad: Mounted from library/busybox
latest: digest: sha256:d9a57e242834bf5128a8031656a4210ddbb699e7ab19a483c6cbfd4ac24a76fc size: 527
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_3>
```



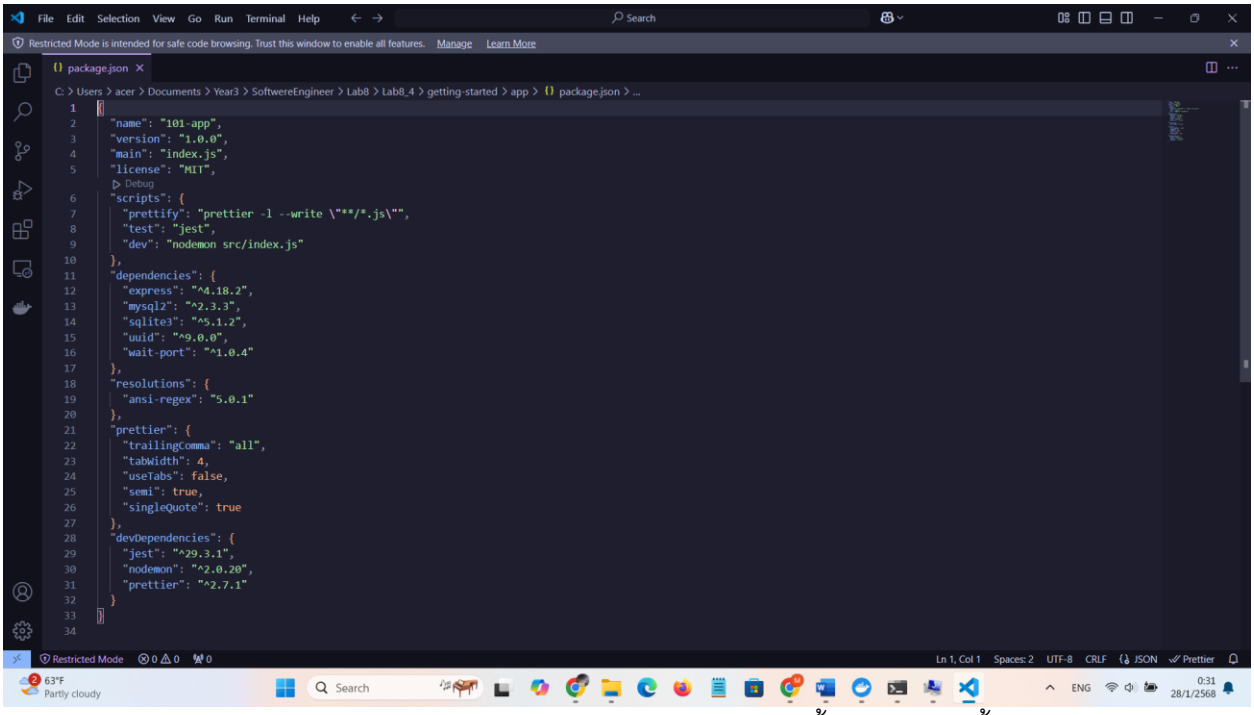
แบบฝึกปฏิบัติที่ 8.4: การ Build แอปพลิเคชันจาก Container image และการ Update แอปพลิเคชัน

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\acer\Documents\Year3\SoftwereEngineer\Lab8> mkdir Lab8_4


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         1/28/2025  12:14 AM                Lab8_4


PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8> cd Lab8_4
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4> git clone https://github.com/docker/getting-started.git
Cloning into 'getting-started'...
remote: Enumerating objects: 980, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 980 (delta 5), reused 1 (delta 1), pack-reused 971 (from 2)
Receiving objects: 100% (980/980), 5.28 MiB | 8.58 MiB/s, done.
Resolving deltas: 100% (523/523), done.
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4>
```



4. ภายใต้ getting-started/app ให้สร้าง Dockerfile พร้อมกับใส่เนื้อหาดังต่อไปนี้ลงไปในไฟล์

   FROM node:18-alpine

   WORKDIR /app

   COPY . .

   RUN yarn install --production

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

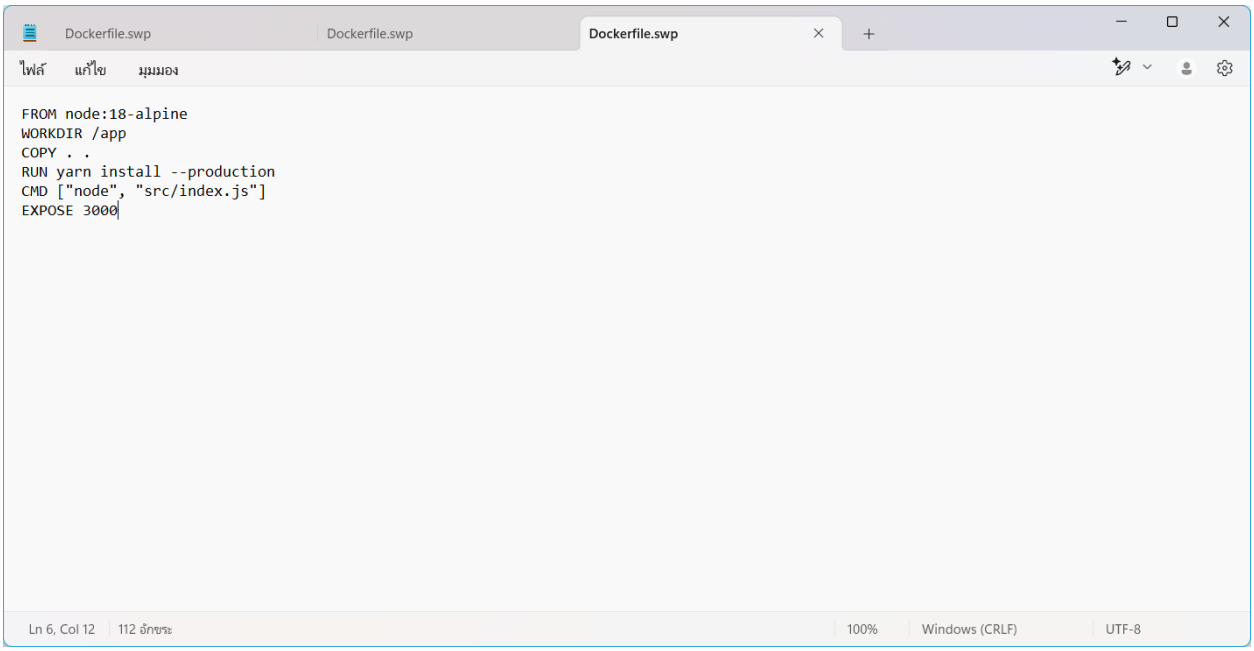
EXPOSE 3000

5. ทำการ Build Docker image ที่สร้างขึ้นด้วยคำสั่งต่อไปนี้ โดยกำหนดใช้ชื่อ image เป็น myapp_รหัสนศ. ไม่มีขีด

$ docker build -t <myapp_รหัสนศ. ไม่มีขีด> .

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

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4> cd C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app

PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> New-Item -ItemType File -Name Dockerfile.swp


    Directory: C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         1/28/2025  12:37 AM              0 Dockerfile.swp
```



```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker build -t myapp_6533802795 -f Dockerfile.swp .
[+] Building 21.7s (10/10) FINISHED                                                                                    docker:desktop-linux
 => [internal] load build definition from Dockerfile.swp                                                                              0.0s
 => => transferring dockerfile: 158B                                                                                                  0.0s
 => [internal] load metadata for docker.io/library/node:18-alpine                                                                     3.2s
 => [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               5.9s
 => => resolve docker.io/library/node:18-alpine@sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25               0.0s
 => => sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1 1.26MB / 1.26MB                                        0.8s
 => => sha256:974afb6cbc0314dc6502b14243b8a39fbb2d04d975e9059dd066be3e274fbb25 7.67kB / 7.67kB                                        0.0s
 => => sha256:6e804119c3884fc5782795bf0d2adc89201c63105aece8647b17a7bcebbc385e 1.72kB / 1.72kB                                        0.0s
 => => sha256:dcbf7b337595be6f4d214e4eed84f230eefe0e4ac03a50380d573e289b9e5e40 6.18kB / 6.18kB                                        0.0s
 => => sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0 3.64MB / 3.64MB                                        0.7s
 => => sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c 40.01MB / 40.01MB                                      4.7s
 => => extracting sha256:1f3e46996e2966e4faa5846e56e76e3748b7315e2ded61476c24403d592134f0                                             0.1s
 => => sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771 444B / 444B                                            1.0s
 => => extracting sha256:37892ffbfcaa871a10f813803949d18c3015a482051d51b7e0da02525e63167c                                             1.0s
 => => extracting sha256:5650d6de56fd0bb419872b876ac1df28f577b39573c3b72fb0d15bf426d01bc1                                             0.0s
 => => extracting sha256:6504e29600c8d5213b52cda800370abb3d12639802d06b46b6fce368990ca771                                             0.0s
 => [internal] load build context                                                                                                    0.4s
 => => transferring context: 4.62MB                                                                                                   0.3s
 => [2/4] WORKDIR /app                                                                                                                1.5s
 => [3/4] COPY . .                                                                                                                    0.0s
 => [4/4] RUN yarn install --production                                                                                              10.2s
 => exporting to image                                                                                                                0.7s
 => => exporting layers                                                                                                               0.7s
 => => writing image sha256:b23de478a29e5e7c2b0991361b6492c486690dafd3383de6f11f419d794991ca                                          0.0s
 => => naming to docker.io/library/myapp_6533802795                                                                                   0.0s

What's next:
    View a summary of image vulnerabilities and recommendations →docker scout quickview
```

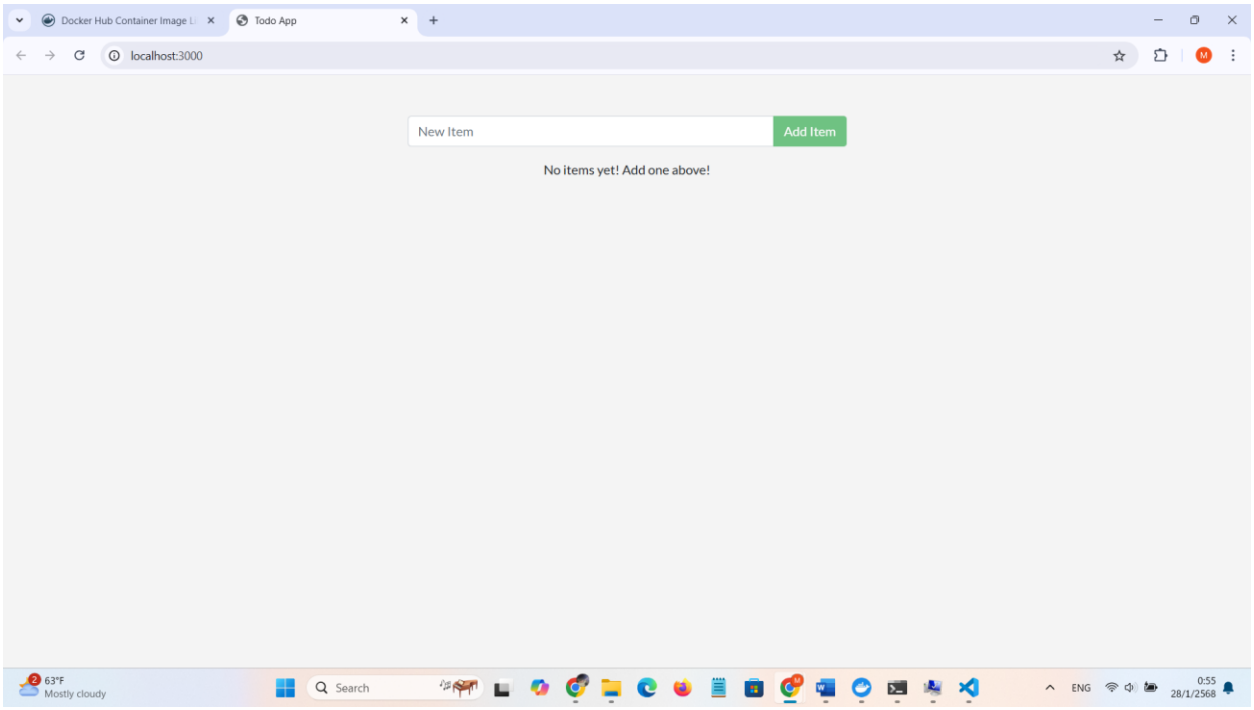
6. ทำการ Start ตัว Container ของแอปพลิเคชันที่สร้างขึ้น โดยใช้คำสั่ง

   $ docker run -dp 3000:3000 <myapp_รหัสนศ. 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\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802795
993c8d60eb42b55b6382b1abbc08cbb55375a1fd7f01edb82c517d7bf1ac8053
```



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
    No items yet! Add one above! เป็น
    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

```
53          <React.Fragment>
54              <AddItemForm onNewItem={onNewItem} />
55              {items.length === 0 && (
56                  <p className="text-center">There is no TODO item. Please add one to the list. By Muttita Chandawong</p>
57              )}
```

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker build -t myapp_6533802795 -f Dockerfile.swp .
[+] Building 19.3s (10/10) FINISHED                                                                                  docker:desktop-linux
 => [internal] load build definition from Dockerfile.swp                                                                         0.0s
 => => transferring dockerfile: 158B                                                                                             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                                                                                                0.0s
 => => transferring context: 8.10kB                                                                                              0.0s
 => CACHED [2/4] WORKDIR /app                                                                                                    0.0s
 => [3/4] COPY . .                                                                                                               0.1s
 => [4/4] RUN yarn install --production                                                                                         15.7s
 => exporting to image                                                                                                           0.9s
 => => exporting layers                                                                                                          0.9s
 => => writing image sha256:5b9e41695c41d1a18296b7fc53c2d5e178b435e1a1125e748aa54c054565ef6f                                     0.0s
 => => naming to docker.io/library/myapp_6533802795                                                                              0.0s

What's next:
    View a summary of image vulnerabilities and recommendations →docker scout quickview
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802795
6641612c880dfbed8c4857f974b4e99db2b4d0eb57542adb638c826d9914b558
docker: Error response from daemon: driver failed programming external connectivity on endpoint gifted_easley (6fb3ccc4f45eb76b32140133a38ad1a9e3d19396526cd73ca7122
365b43aa791): Bind for 0.0.0.0:3000 failed: port is already allocated.
```

(1) Error ที่เกิดขึ้นหมายความอย่างไร และเกิดขึ้นเพราะอะไร

เกิดขึ้นเพราะพอร์ต 3000 ถูกใช้งานอยู่ วิธีแก้คือ หยุดโปรเซสหรือคอนเทนเนอร์ที่ใช้พอร์ตนี้ หรือเปลี่ยนไปใช้พอร์ตอื่นสำหรับคอนเทนเนอร์

11. ลบ Container ของ Web application เวอร์ชันก่อนแก้ไขออกจากระบบ โดยใช้วิธีใดวิธีหนึ่งดังต่อไปนี้

    a. ผ่าน Command line interface

        i. ใช้คำสั่ง $ docker ps เพื่อดู Container ID ที่ต้องการจะลบ

        ii. Copy หรือบันทึก Container ID ไว้

        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

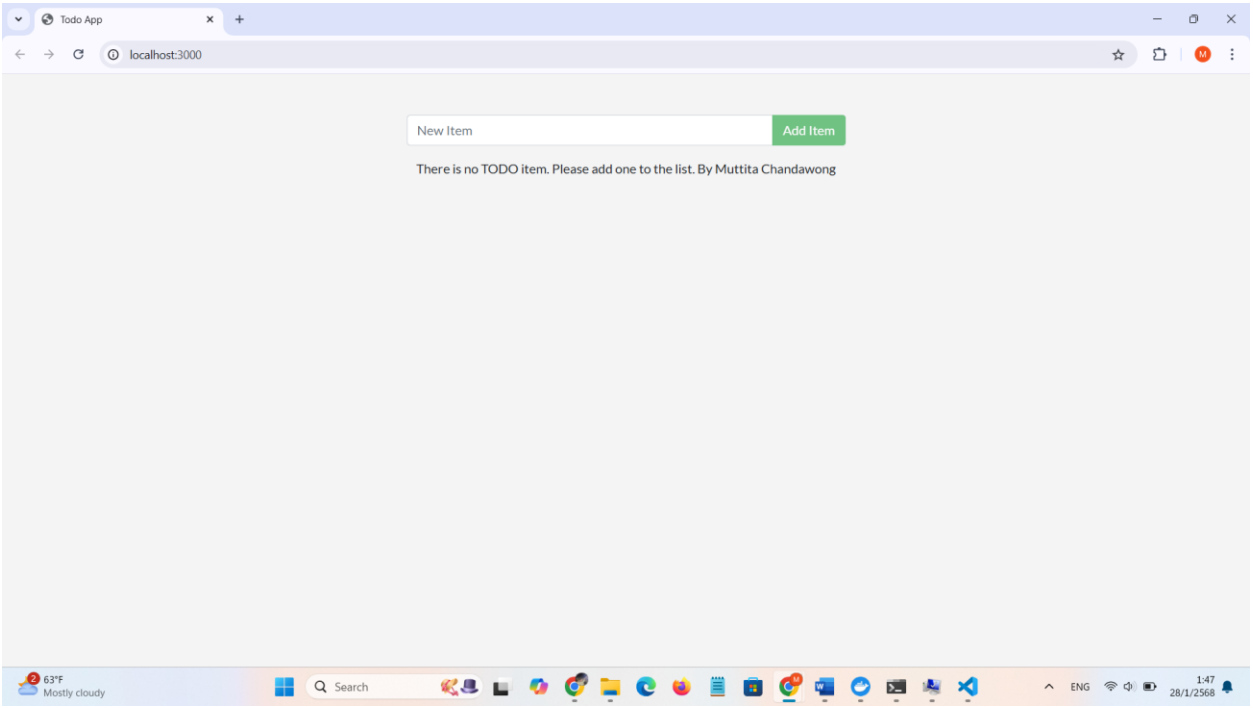
12. Start และรัน Container ตัวใหม่อีกครั้ง โดยใช้คำสั่งเดียวกันกับข้อ 6

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

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

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                    NAMES
993c8d60eb42   b23de478a29e   "docker-entrypoint.s…"   47 minutes ago   Up 47 minutes   0.0.0.0:3000->3000/tcp   focused_rosalind
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> ^C
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker stop 993c8d60eb42
993c8d60eb42
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker rm 993c8d60eb42
993c8d60eb42
```

```
PS C:\Users\acer\Documents\Year3\SoftwereEngineer\Lab8\Lab8_4\getting-started\app> docker run -dp 3000:3000 myapp_6533802795
0c6d28af30ecf04754fc34e24251f6a08f06e52f3489f7c396743e41687c2246
```



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

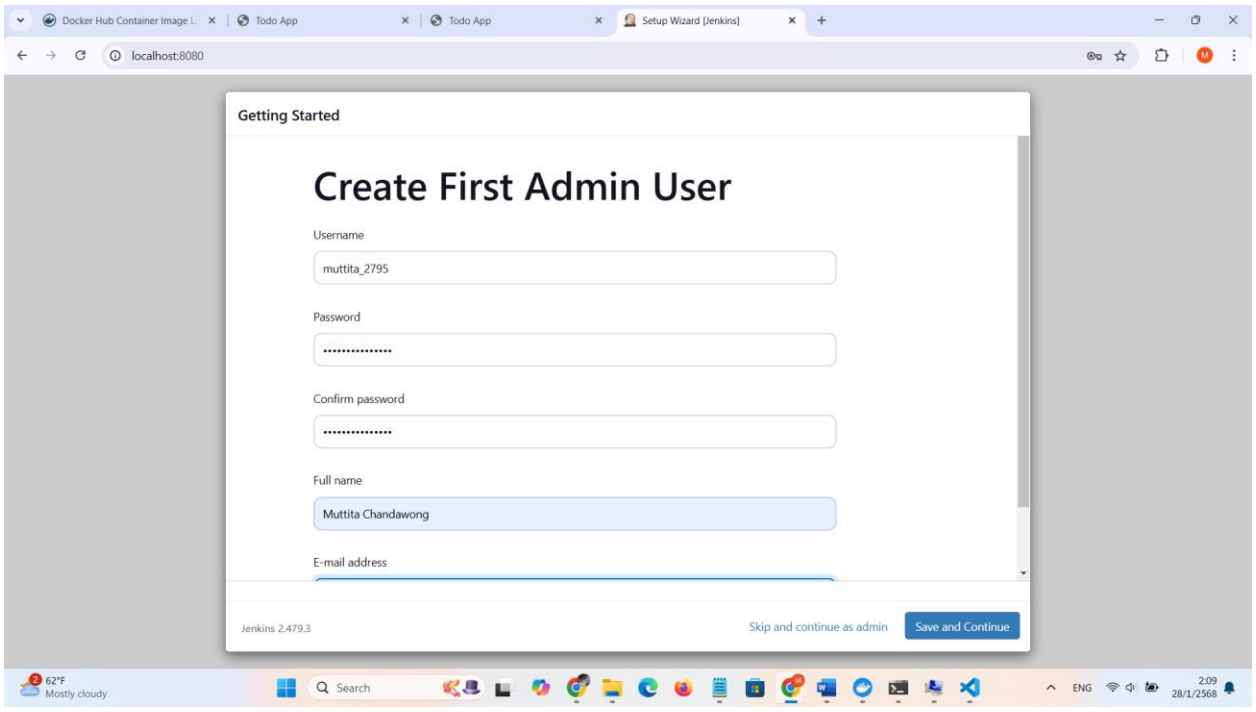
   jenkins_home:/var/jenkins_home jenkins/jenkins:lts-jdk1
3. บันทึกรหัสผ่านของ Admin user ไว้สำหรับ log-in ในครั้งแรก

[Check point#12] Capture หน้าจอที่แสดงผล Admin password

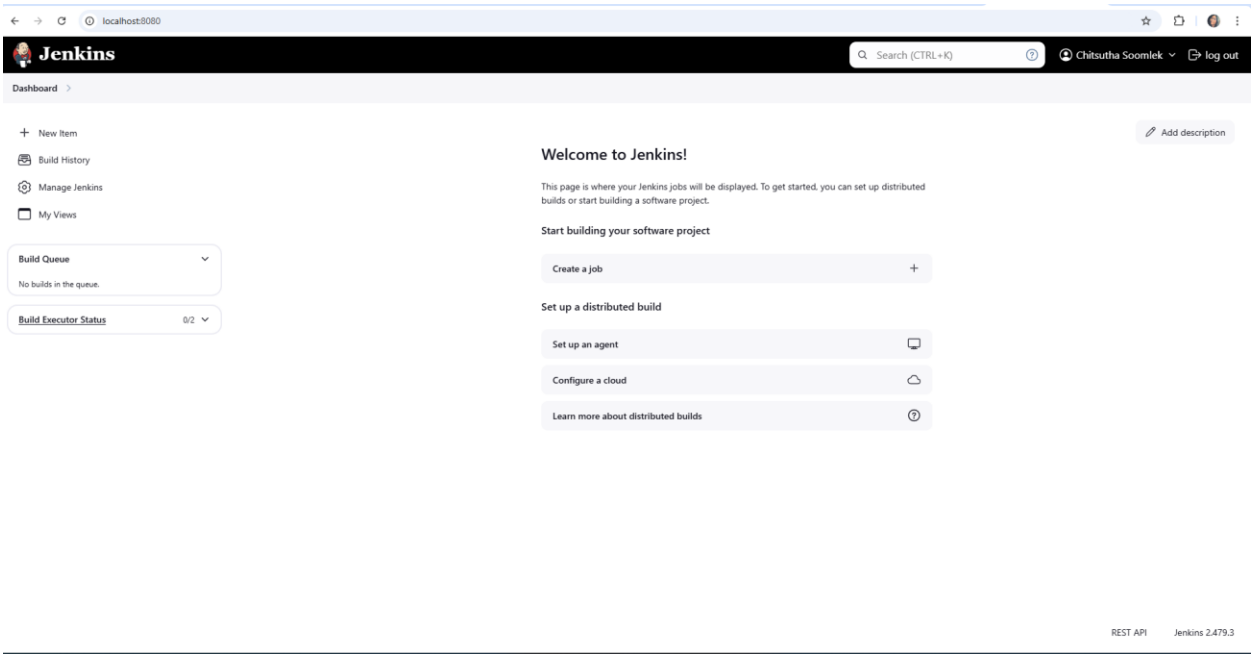
```
2025-01-27 19:04:00.297+0000 [id=45]    INFO    jenkins.InitReactorRunner$1#onAttained: System config adapted
2025-01-27 19:04:00.298+0000 [id=40]    INFO    jenkins.InitReactorRunner$1#onAttained: Loaded all jobs
2025-01-27 19:04:00.299+0000 [id=47]    INFO    jenkins.InitReactorRunner$1#onAttained: Configuration for all jobs updated
2025-01-27 19:04:00.330+0000 [id=60]    INFO    hudson.util.Retrier#start: Attempt #1 to do the action check updates server
2025-01-27 19:04:00.589+0000 [id=47]    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:

3211d97f882d45efa1777ee35433cad1

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

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

4. เมื่อได้รับการยืนยันว่า Jenkins is fully up and running ให้เปิดบราวเซอร์ และป้อนที่อยู่เป็น localhost:8080
5. ทำการ Unlock Jenkins ด้วยรหัสผ่านที่ได้ในข้อที่ 3
6. สร้าง Admin User โดยใช้ username เป็นชื่อจริงของนักศึกษาพร้อมรหัสสี่ตัวท้าย เช่น somsri_3062

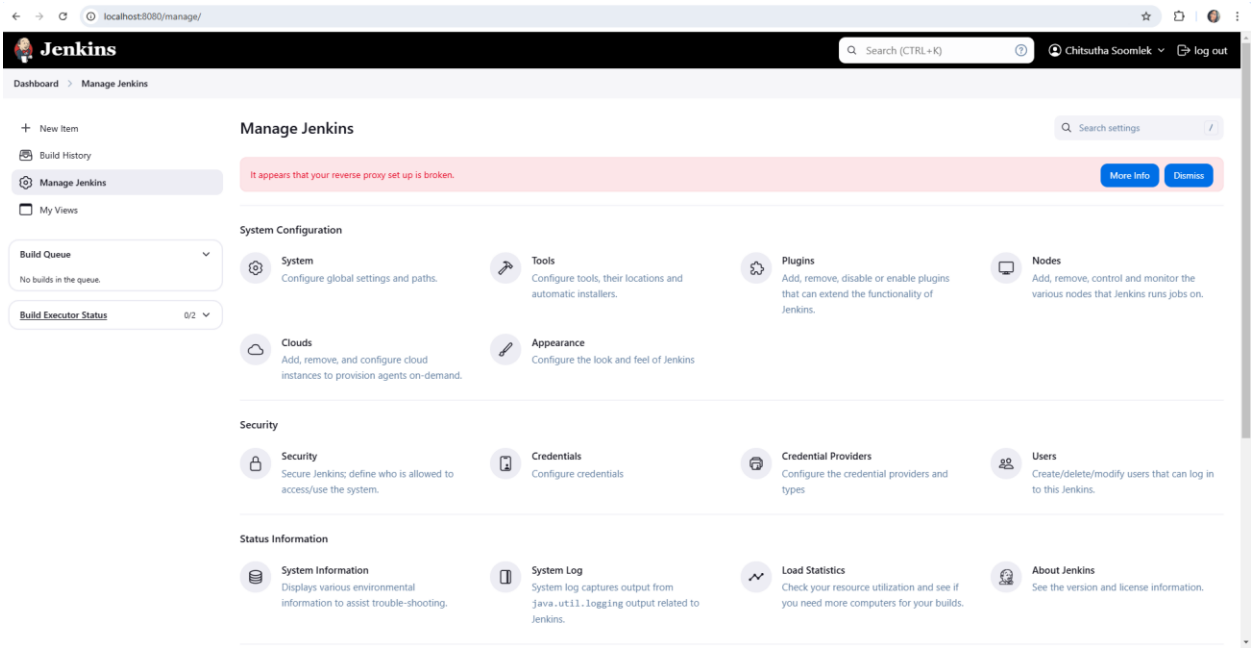
[Check point#13] Capture หน้าจอที่แสดงผลการตั้งค่า



7. กำหนด Jenkins URL เป็น http://localhost:8080/lab8
8. เมื่อติดตั้งเรียบร้อยแล้วจะพบกันหน้า Dashboard ดังแสดงในภาพ

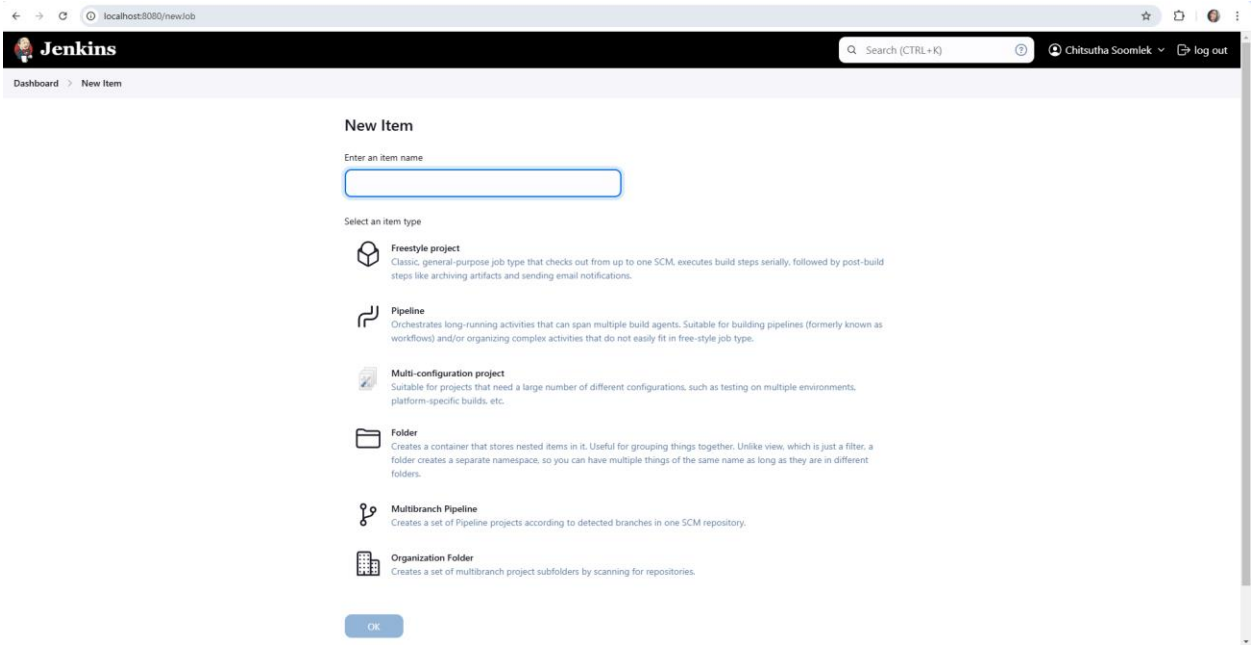


9. เลือก Manage Jenkins แล้วไปที่เมนู Plugins



10. ไปที่เมนู Available plugins แล้วเลือกติดตั้ง Robotframework เพิ่มเติม

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



12. นำไฟล์ .robot ที่ทำให้แบบฝึกปฏิบัติที่ 7 (Lab#7) ไปไว้บน Repository ของนักศึกษา จากนั้นตั้งค่าที่จำเป็นในหน้านี้ทั้งหมด ดังนี้

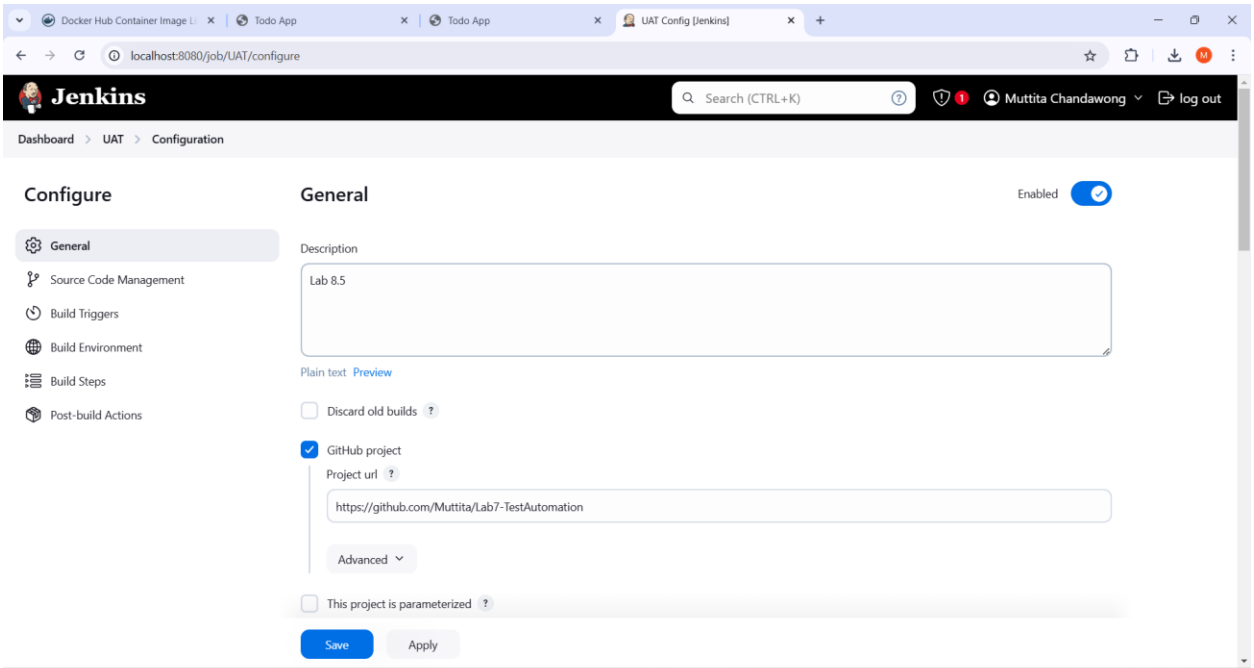
Description: Lab 8.5

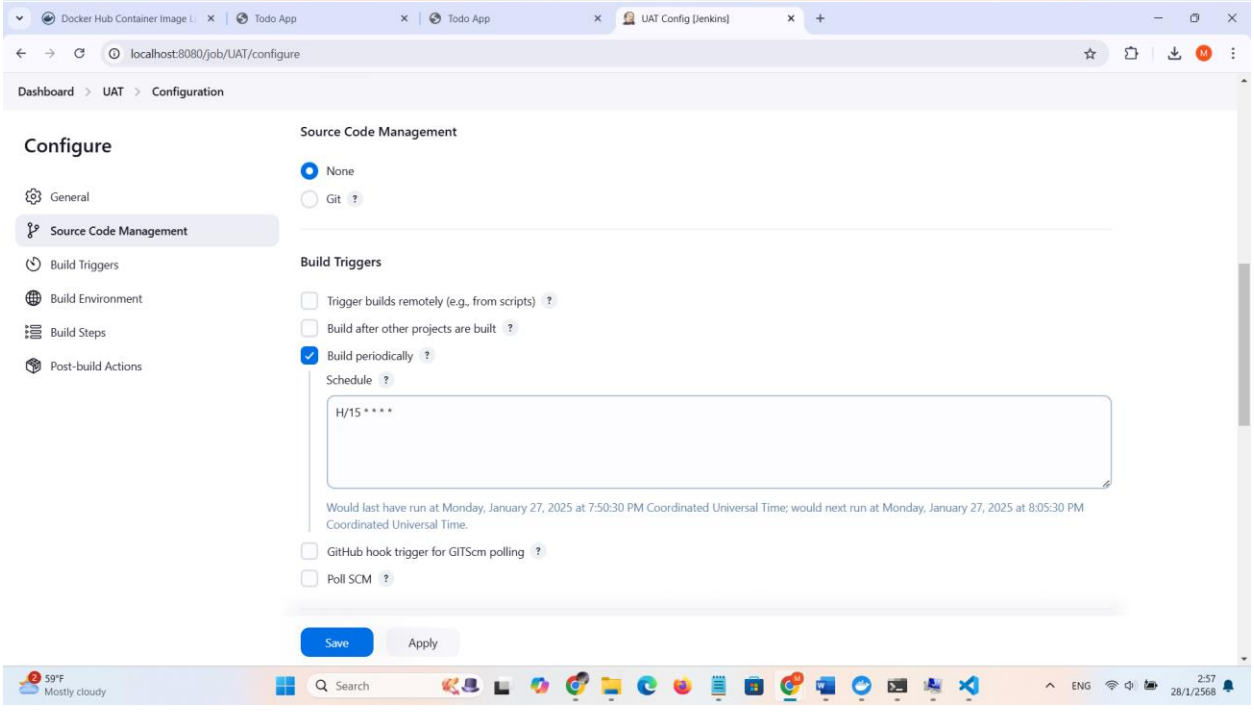
GitHub project: กดเลือก แล้วใส่ Project URL เป็น repository ที่เก็บโค้ด .robot (ดูขั้นตอนที่ 12)

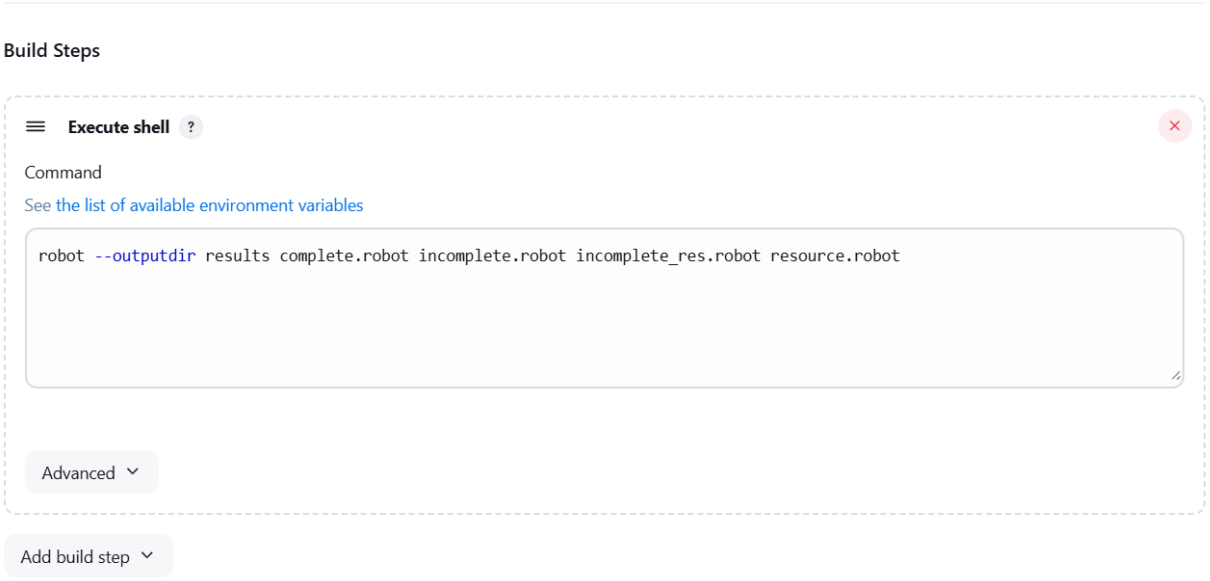
Build Trigger: เลือกแบบ Build periodically แล้วกำหนดให้ build ทุก 15 นาที

Build Steps: เลือก Execute shell แล้วใส่คำสั่งในการรันไฟล์ .robot (หากไฟล์ไม่ได้อยู่ในหน้าแรกของ repository ให้ใส่ Path ไปถึงไฟล์ให้เรียบร้อยด้วย)

[Check point#14] Capture หน้าจอแสดงการตั้งค่า พร้อมกับตอบคำถามต่อไปนี้







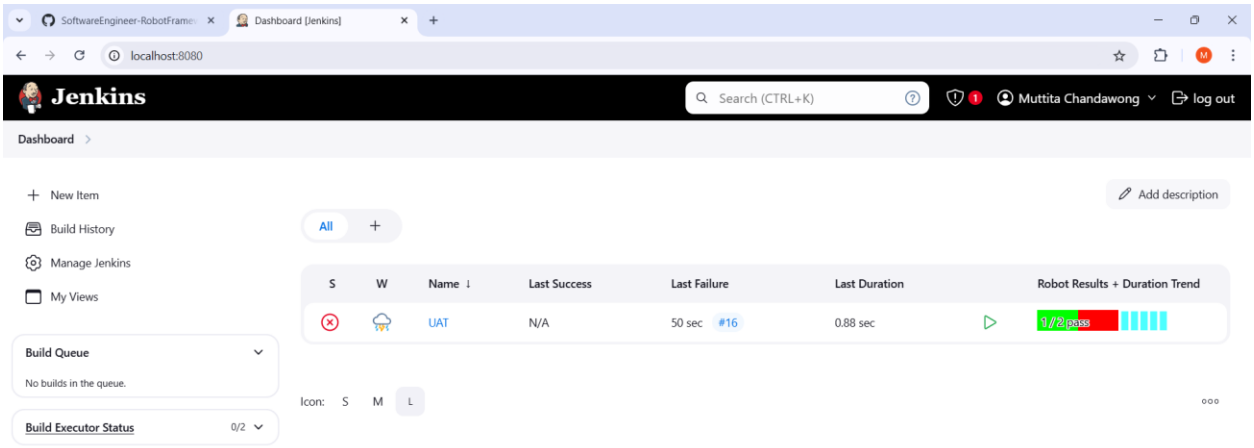
(1) คำสั่งที่ใช้ในการ Execute ไฟล์ .robot ใน Build Steps คือ

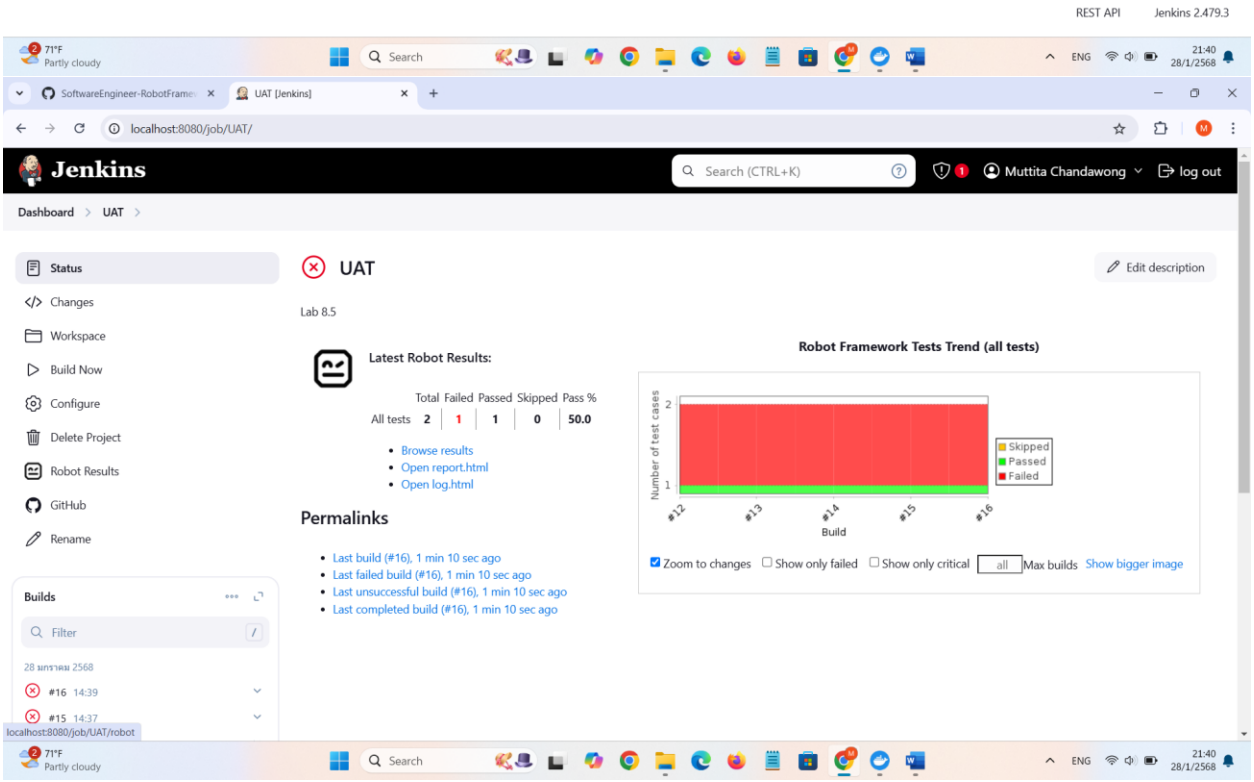
robot complete.robot incomplete.robot incomplete_res.robot resource.robot

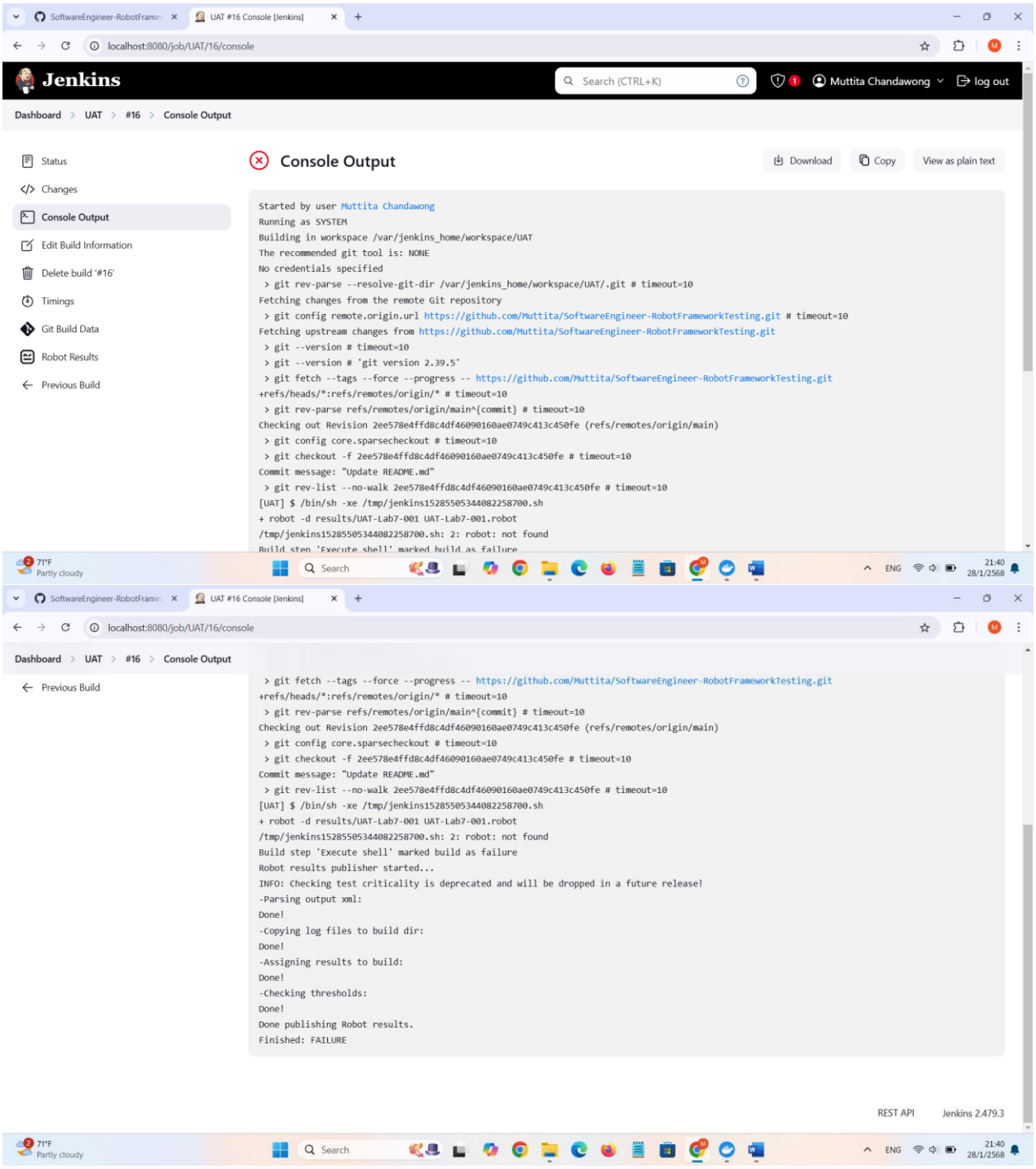
Post-build action: เพิ่ม Publish Robot Framework test results -> ระบุไดเร็คทอรีที่เก็บไฟล์ผลการทดสอบโดย Robot framework ในรูป xml และ html -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ไม่ผ่านแล้วนับว่าซอฟต์แวร์มีปัญหา -> ตั้งค่า Threshold เป็น % ของการทดสอบที่ผ่านแล้วนับว่าซอฟต์แวร์มีอยู่ในสถานะที่สามารถนำไปใช้งานได้ (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)
Muttita Chandawong
log out
Dashboard
New Item
Build History
Manage Jenkins
My Views
Build Queue
No builds in the queue.
Build Executor Status
Add description
All
S
W
Name
Last Success
Last Failure
Last Duration
Robot Results + Duration Trend
UAT
N/A
50 sec #16
0.88 sec
Icon: S M L


REST API
Jenkins 2.479.3
Jenkins
Search (CTRL+K)
Muttita Chandawong
log out
Dashboard
UAT
Status
Changes
Workspace
Build Now
Configure
Delete Project
Robot Results
GitHub
Rename
Builds
Filter
UAT
Edit description
Lab 8.5
Latest Robot Results:
Total Failed Passed Skipped Pass %
All tests 2 1 1 0 50.0
Browse results
Open report.html
Open log.html
Permalinks
Last build (#16), 1 min 10 sec ago
Last failed build (#16), 1 min 10 sec ago
Last unsuccessful build (#16), 1 min 10 sec ago
Last completed build (#16), 1 min 10 sec ago
Robot Framework Tests Trend (all tests)
Number of test cases
Skipped
Passed
Failed
Build
Zoom to changes
Show only failed
Show only critical
Max builds
Show bigger image


UAT #16 Console [Jenkins]
localhost:8080/job/UAT/16/console
Jenkins
Search (CTRL+K)
Muttita Chandawong
log out
Dashboard > UAT > #16 > Console Output
Status
Changes
Console Output
Edit Build Information
Delete build '#16'
Timings
Git Build Data
Robot Results
Previous Build
Console Output
Download
Copy
View as plain text
Started by user Muttita Chandawong
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/Muttita/SoftwareEngineer-RobotFrameworkTesting.git # timeout=10
Fetching upstream changes from https://github.com/Muttita/SoftwareEngineer-RobotFrameworkTesting.git
> git --version # timeout=10
> git --version # 'git version 2.39.5'
> git fetch --tags --force --progress -- https://github.com/Muttita/SoftwareEngineer-RobotFrameworkTesting.git
+refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 2ee578e4ffd8c4df46090160ae0749c413c450fe (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 2ee578e4ffd8c4df46090160ae0749c413c450fe # timeout=10
Commit message: "Update README.md"
> git rev-list --no-walk 2ee578e4ffd8c4df46090160ae0749c413c450fe # timeout=10
[UAT] $ /bin/sh -xe /tmp/jenkins15285505344082258700.sh
+ robot -d results/UAT-Lab7-001 UAT-Lab7-001.robot
/tmp/jenkins15285505344082258700.sh: 2: robot: not found
Build step 'Execute shell' marked build as failure
Robot results publisher started...
INFO: Checking test criticality is deprecated and will be dropped in a future release!
-Parsing output xml:
Done!
-Copying log files to build dir:
Done!
-Assigning results to build:
Done!
-Checking thresholds:
Done!
Done publishing Robot results.
Finished: FAILURE
REST API
Jenkins 2.479.3