

รหัสนักศึกษา : 61070346

Setup Lab สำหรับผู้ใช้ Lab environment บน Katacoda

1. เข้าไปที่ <https://katacoda.com/saranonuan/scenarios/ansible101> และเริ่มที่ข้อ 3 ได้เลย

Setup Lab สำหรับผู้ใช้เครื่องตนเองในการ Lab

1. ต้องติดตั้ง Ansible ในเครื่อง
 1. Windows
<https://fakhriddarmawan.medium.com/install-ansible-on-windows-subsystem-linux-ac9245219077>
 2. OSX
<https://hvops.com/articles/ansible-mac-osx>
2. Download source lab project จากใน Google Class Room ไฟล์ชื่อ sdte-lab7.zip และ Extract ให้เรียบร้อย
3. สร้าง Repository ใน Github โดยให้ตั้งชื่อว่า sdte-lab7 และให้เลือกเป็น Public Repository
4. ทำการ Initial Git, Commit ไฟล์ใน Folder ที่ดาวโหลดมาในข้อ 1 (ยังไม่ต้องแก้ไขไฟล์ใด) และ push ขึ้น Remote repository ที่สร้างในข้อที่ 3
 - ▶ git init
 - ▶ git add .
 - ▶ git commit -m "First commit"
 - ▶ git remote add origin REMOTE_REPO_URL.git
 - ▶ เปลี่ยน REMOTE_REPO_URL.git ให้เป็น URL Git repo ที่สร้างในข้อ 3
 - ▶ สำหรับคนที่ใช้ Katacoda ให้ใช้ URL Git เป็น https
 - ▶ git branch -m main
 - ▶ git push -u origin main

บันทึกผล

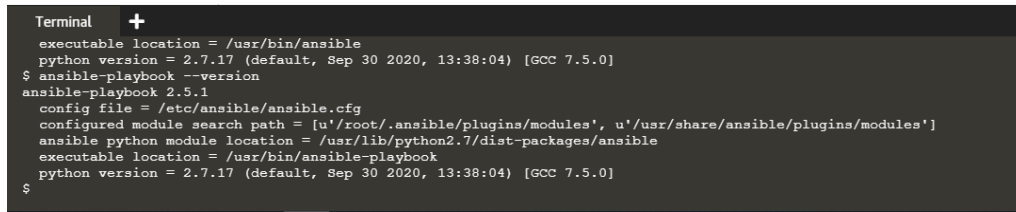
- A. Git URL : <https://github.com/lalitaSa/sdte-lab7>
- B. รัน “ansible --version” เพื่อเช็คว่าได้ติดตั้ง ansible แล้ว และตรวจสอบ Version ด้วย, ทำการ Capture screen output

```

16 | - ~/.keys/authorized_keys:/root/.ssh/authorized_keys"
17 | host03:
18 |   image: rastasheep/ubuntu-sshd:18.04
19 |   ports:
20 |     - "2203:22"
21 |     - "8083:80"
22 |   volumes:
23 |     - ~/.keys/authorized_keys:/root/.ssh/authorized_keys"
24 |
Terminal +
$ cd .
$ cd ..
$ ansible --version
ansible 2.5.1
config file = /etc/ansible/ansible.cfg
configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/dist-packages/ansible
executable location = /usr/bin/ansible
python version = 2.7.17 (default, Sep 30 2020, 13:38:04) [GCC 7.5.0]
$

```

- C. รัน “ansible-playbook --version” เพื่อเช็คว่าได้ติดตั้ง ansible-playbook แล้ว และตรวจสอบ Version ด้วย, ทำการ Capture screen output



```
Terminal +
executable location = /usr/bin/ansible
python version = 2.7.17 (default, Sep 30 2020, 13:38:04) [GCC 7.5.0]
$ ansible-playbook --version
ansible-playbook 2.5.1
config file = /etc/ansible/ansible.cfg
configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']
ansible python module location = /usr/lib/python2.7/dist-packages/ansible
executable location = /usr/bin/ansible-playbook
python version = 2.7.17 (default, Sep 30 2020, 13:38:04) [GCC 7.5.0]
$
```

** หากเพิ่งติดตั้งแล้วรันคำสั่งไม่สำเร็จ แนะนำให้ปิดและเปิด Terminal ใหม่ดูหนึ่งครั้งเพื่อให้ Terminal อัปเดต paths **

Lab 1 - spin up lab environment, สร้าง Ubuntu ขึ้นมา 3 ตัว เพื่อจำลองเป็นเครื่องที่สร้างขึ้นใหม่ ยังไม่ได้ติดตั้งอะไร

1. เปิด Terminal หรือ PowerShell (สำหรับผู้ใช้เครื่องตนเอง)
2. Change directory (cd) เข้าไปที่ folder “environments”
3. รัน "docker-compose up -d" เพื่อสร้าง Ubuntu จำลองขึ้นมาในเครื่องด้วย docker
4. สำหรับผู้ใช้ katacoda ให้รันคำสั่งต่อไปนี้เพิ่มเติม
 “docker-compose exec host01 ln --symbolic /usr/bin/python3 /usr/bin/python”
 “docker-compose exec host02 ln --symbolic /usr/bin/python3 /usr/bin/python”
 “docker-compose exec host03 ln --symbolic /usr/bin/python3 /usr/bin/python”
5. รัน “docker-compose ps” เพื่อตรวจสอบว่า Ubuntu ถูกรันขึ้นมาเรียบร้อยแล้วหรือไม่

B. Capture Screen output ที่ได้ในข้อ 4

```
Terminal +
2b1920b96f11: Pull complete
Digest: sha256:1df808ccf95c13b8e62548ad434829bc28ee701a65624e5c0440fc24482e4a62
Status: Downloaded newer image for rastasheep/ubuntu-sshd:18.04
Creating environments_host01_1 ... done
Creating environments_host02_1 ... done
Creating environments_host03_1 ... done
$ docker-compose exec host01 ln --symbolic /usr/bin/python3 /usr/bin/python
$ docker-compose exec host02 ln --symbolic /usr/bin/python3 /usr/bin/python
$ docker-compose exec host03 ln --symbolic /usr/bin/python3 /usr/bin/python
$ docker-compose ps
```

Name	Command	State	Ports
environments_host01_1	/usr/sbin/sshd -D	Up	0.0.0.0:2201->22/tcp, 0.0.0.0:8081->80/tcp
environments_host02_1	/usr/sbin/sshd -D	Up	0.0.0.0:2202->22/tcp, 0.0.0.0:8082->80/tcp
environments_host03_1	/usr/sbin/sshd -D	Up	0.0.0.0:2203->22/tcp, 0.0.0.0:8083->80/tcp

C. Lab Quiz

1. Q> Ubuntu Version อะไร
 A> Description: Ubuntu 18.04.5 LTS
 Release: 18.04
2. Q> จากตั้งค่าใน docker-compose.yml จะสามารถ ssh เพื่อเข้าถึง host01,host02,host03 ได้ที่ port ไต
 A> host01 port : "8081:80"
 host02 port : "8082:80"
 host03 port : "8083:80"

Lab 2 - สร้าง Ansible Inventory file

1. Change directory (cd) ไปที่ folder “lab”
2. สร้างไฟล์ “lab/inventory.yml” ทำการใส่ content ในไฟล์ จาก

<https://gist.github.com/saranonuan/928f307eb012ef8ae51845325961499e>

ในไฟล์นี้จะระบุ host โดยให้ชื่อว่า host01 โดย ssh เข้าถึงได้ที่ 127.0.0.1 ที่ port 2201 โดยใช้ user “root” และ private key ที่ path environments/keys/id_rsa. โดย host01 อยู่ในกลุ่มที่ชื่อว่า “web”

D. รัน “ansible -i inventory.yml -m ping host01” แล้ว Capture Screen Output

(** cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)

(** ต้อง Ping host01 ได้ output เป็นสีเขียว และสถานะ SUCCESS)

```
Terminal +
environments host01 1 /usr/sbin/sshd -D Up 0.0.0.0:2201->22/tcp, 0.0.0.0:8081->80/tcp
environments host02 1 /usr/sbin/sshd -D Up 0.0.0.0:2202->22/tcp, 0.0.0.0:8082->80/tcp
environments host03 1 /usr/sbin/sshd -D Up 0.0.0.0:2203->22/tcp, 0.0.0.0:8083->80/tcp
$ cd ..
$ cd lab/
$ ls -l ../environments/keys/id_rsa
-rw-r--r-- 1 root root 2610 Mar 8 08:19 ../environments/keys/id_rsa
$ chmod 600 ../environments/keys/id_rsa
$ ls -l ../environments/keys/id_rsa
-rw----- 1 root root 2610 Mar 8 08:19 ../environments/keys/id_rsa
$ ansible -i inventory.yml -m ping host01
host01 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

3. ทำการแก้ไขไฟล์ inventory.yml เพิ่ม host02,host03 โดย ansible_port ต้องชี้ให้ถูกต้อง โดย
 - host02 ไปที่ port 2202 และค่า hostname เป็น host02
 - host03 ไปที่ port 2203 และค่า hostname เป็น host03

E. รัน “ansible -i inventory.yml -m ping web” แล้ว Capture Screen Output

(** cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)

(** ต้อง Ping host01,host02,host03 ได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 ตัว)

```
Terminal +
$ cd lab/
$ ls -l ../environments/keys/id_rsa
-rw-r--r-- 1 root root 2610 Mar 8 08:19 ../environments/keys/id_rsa
$ chmod 600 ../environments/keys/id_rsa
$ ls -l ../environments/keys/id_rsa
-rw----- 1 root root 2610 Mar 8 08:19 ../environments/keys/id_rsa
$ ansible -i inventory.yml -m ping host01
host01 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
$ ansible -i inventory.yml -m ping web
host01 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
host02 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
host03 | SUCCESS => {
  "changed": false,
  "ping": "pong"
}
```

```
host01 | FAILED! => {  
  "changed": false,  
  "module_stderr": "Shared connection to 127.0.0.1 closed.\r\n",  
  "module_stdout": "bash: warning: setlocale: LC_ALL: cannot change  
  "msg": "MODULE FAILURE",  
  "rc": 127  
}
```

**** ใครที่เจอปัญหาต่อไปนี ให้ร่นคำสั่ง Lab1 ข้อที่ 4 เพิ่มเติม ****

Lab 3 - สร้าง Ansible playbook สำหรับติดตั้ง Nginx ในแต่ละ hosts

1. Change directory (cd) ไปที่ folder “lab”
2. สร้างไฟล์ “lab/setup_nginx.yml” , ทำการใส่ content ในไฟล์ setup_nginx.yml จาก <https://gist.github.com/saranonuan/a8b27dad938d4563277aa872f9065540>
3. สร้าง folder ชื่อ “templates”
4. สร้างไฟล์ “lab/templates/site.conf” ทำการใส่ content ในไฟล์ site.conf จาก <https://gist.github.com/saranonuan/ac9d75e12f0d8c7b59e8ad6f5a8a82b8>

F. รัน “ansible-playbook -i inventory.yml setup_nginx.yml” แล้ว Capture Screen Output

(** cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)

(** ต้องได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 hosts)

```
Terminal +
$ cd ..
$ cd lab/
$ mkdir templates
$ ansible-playbook -i inventory.yml setup_nginx.yml

PLAY [web] *****

TASK [Gathering Facts] *****
ok: [host03]
ok: [host02]
ok: [host01]

TASK [apt-get update] *****
```

```
Terminal +
TASK [Gathering Facts] *****
ok: [host03]
ok: [host02]
ok: [host01]

TASK [apt-get update] *****
ok: [host02]
ok: [host03]
ok: [host01]

TASK [install nginx] *****
changed: [host02]
changed: [host01]
changed: [host03]

TASK [create www directory] *****
changed: [host01]
changed: [host03]
changed: [host02]

TASK [delete default nginx site] *****
changed: [host01]
changed: [host02]
changed: [host03]

TASK [copy nginx site.conf] *****

TASK [copy nginx site.conf] *****
changed: [host02]
changed: [host01]
changed: [host03]

TASK [restart nginx] *****
changed: [host03]
changed: [host02]
changed: [host01]

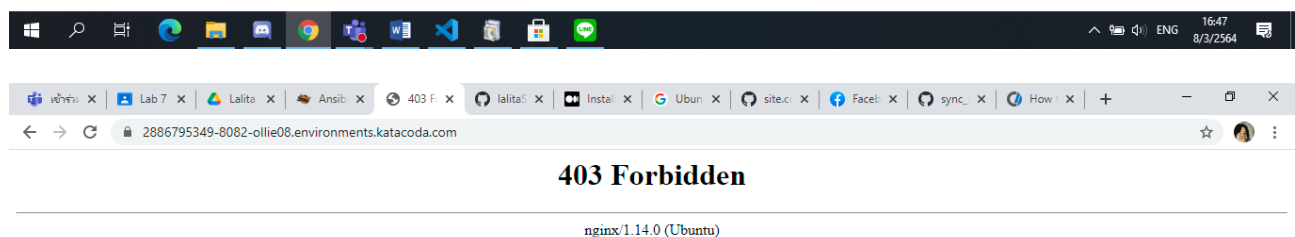
TASK [set INIT status] *****
ok: [host01]
ok: [host02]
ok: [host03]

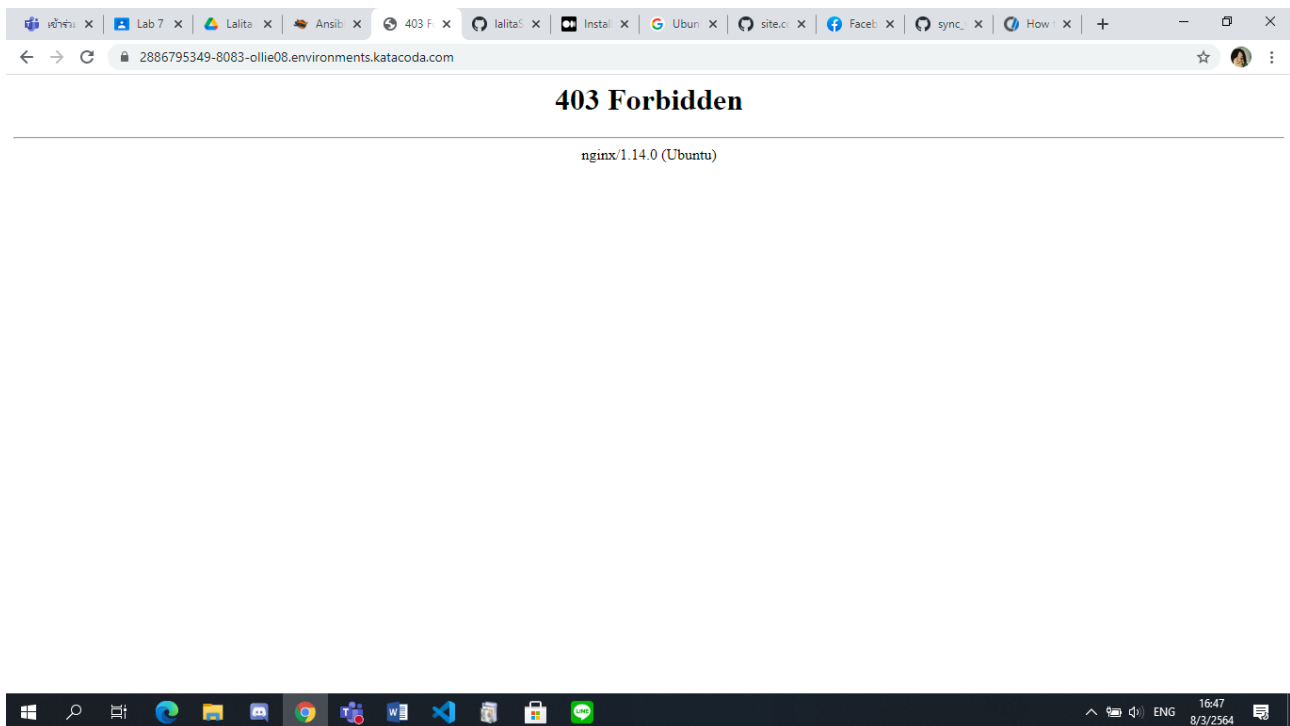
TASK [start the Nginx service, if necessary] *****
ok: [host02]
ok: [host01]
ok: [host03]

PLAY RECAP *****
host01      : ok=9    changed=5    unreachable=0    failed=0
host02      : ok=9    changed=5    unreachable=0    failed=0
host03      : ok=9    changed=5    unreachable=0    failed=0
$
```

G. เปิด Browser แล้วไปที่ localhost:8081 และ localhost:8082 และ localhost:8083 แล้ว Capture Screen Output ในแต่ละ Port ว่าได้ติดตั้ง nginx แล้วเรียบร้อยทั้ง 3 เครื่อง

(** จะขึ้นเป็นหน้า Error ของ nginx ซึ่งถูกต้องแล้ว เพราะยังไม่ได้ใส่หน้าเว็บอะไรเข้าไปใน Nginx)





Lab 4 - สร้าง Ansible playbook สำหรับ Copy หน้าเว็บที่สร้างขึ้นเพื่อนำไปใส่ใน Nginx ในแต่ละ hosts

1. Change directory (cd) ไปที่ folder “lab”
2. สร้าง folder ชื่อ “site”
3. สร้างไฟล์ “lab/site/index.html” ทำการใส่ content เป็น HTML ตามแต่นักศึกษาต้องการ
4. สร้างไฟล์ “lab/sync_web.yml” , ทำการใส่ content ในไฟล์ sync_web.yml จาก <https://gist.github.com/saranonuan/5a6214e4a40448738faac4bf8b47588c>

H. รัน “ansible-playbook -i inventory.yml sync_web.yml” แล้ว Capture Screen Output

(** cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)

(** ต้องได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 hosts)

```
Terminal
$ cd lab/
$ ansible-playbook -i inventory.yml sync_web.yml

PLAY [web] *****

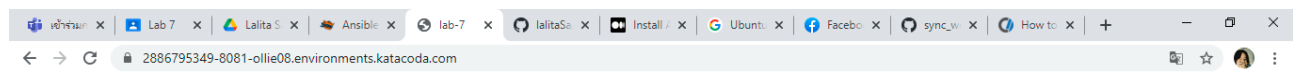
TASK [Gathering Facts] *****
ok: [host01]
ok: [host02]
ok: [host03]

TASK [copy website] *****
ok: [host01]
ok: [host02]
ok: [host03]

PLAY RECAP *****
host01 : ok=2    changed=0    unreachable=0    failed=0
host02 : ok=2    changed=0    unreachable=0    failed=0
host03 : ok=2    changed=0    unreachable=0    failed=0

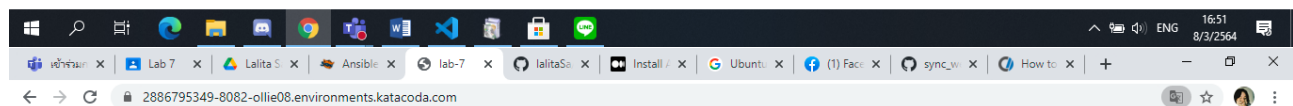
$
```


I. เปิด Browser แล้วไปที่ localhost:8081 และ localhost:8082 และ localhost:8083 แล้ว Capture Screen Output ในแต่ละ Port โดยจะแสดงเป็นหน้าเว็บที่นักศึกษาได้ทำในข้อที่ 3



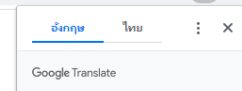
61070346

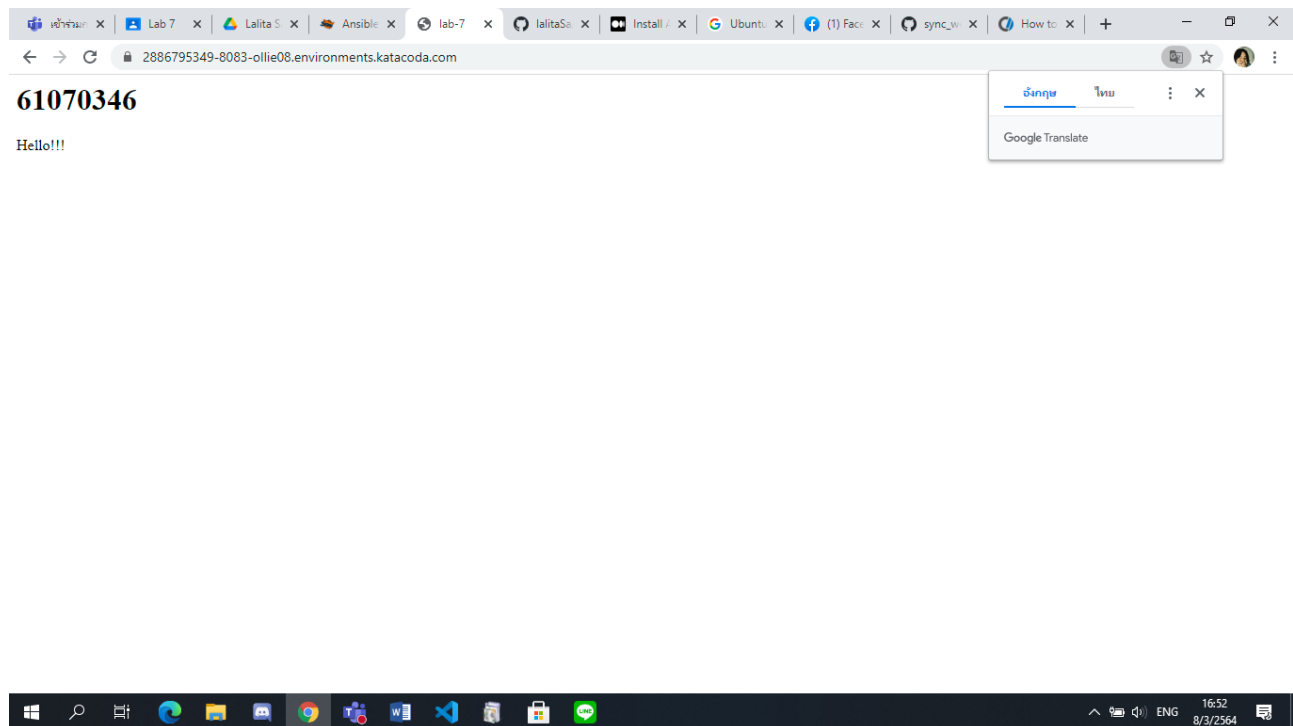
Hello!!!



61070346

Hello!!!





Lab 5 / CHALLENGE - ทำอย่างไรก็ได้ให้ Output ที่ได้จากหน้าเว็บในแต่ละ hosts นั้นแตกต่างกัน

Guide line

Way 1 - One HTML, Replace content by hostname variable

<https://www.middlewareinventory.com/blog/ansible-replace-line-in-file-ansible-replace-examples/>

Way 2 - Playbook command set target individual host

https://docs.ansible.com/ansible/latest/user_guide/intro_patterns.html

J. Capture ไฟล์ที่แก้ไขหรือเพิ่มเข้ามา ที่ทำให้ Challenge นี้สำเร็จ

K. เปิด Browser แล้วไปที่ localhost:8081 และ localhost:8082 และ localhost:8083 แล้ว Capture Screen Output ในแต่ละ Port เพื่อให้เห็นผลลัพธ์ที่ต่างกันในแต่ละ Port

Post lab

1. เมื่อเสร็จแล้ว ให้ add, commit, push ไฟล์ทั้งหมดขึ้น Repository
2. หาก Download ออกมาทำให้ Save ไฟล์เป็น PDF แล้วอัปโหลดส่งใน Google Classroom หากทำใน GoogleDocs บนไฟล์ของ Google Classroom ให้ส่งด้วยไฟล์นั้นได้เลย