**รหัสนักศึกษา :**

**Setup Lab สำหรับผู้ที่ใช้ Lab environment บน Katacoda**

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

**Setup Lab สำหรับผู้ที่ใช้เครื่องตนเองในการ Lab**

1. ต้องติดตั้ง Ansible ในเครื่อง
   1. Windows  
      <https://fakhridarmawan.medium.com/install-ansible-on-windows-subsystem-linux-ac9245219077>
   2. OSX

<https://hvops.com/articles/ansible-mac-osx>

1. Download source lab project จากใน Google Class Room ไฟล์ชื่อ sdte-lab7.zip และ Extract ให้เรียบร้อย
2. สร้าง Repository ใน Github โดยให้ตั้งชื่อว่า sdte-lab7 และให้เลือกเป็น Public Repository
3. 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**

1. **Git URL : https://github.com/ptydvp/sdte-lab7**
2. **รัน “ansible --version” เพื่อเช็คว่าได้ติดตั้ง ansible แล้ว และตรวจสอบ Version ด้วย, ทำการ Capture screen outputText

   Description automatically generated**
3. **รัน “ansible-playbook --version” เพื่อเช็คว่าได้ติดตั้ง ansible-playbook แล้ว และตรวจสอบ Version ด้วย, ทำการ Capture screen outputText

   Description automatically generated**

\*\* หากพึ่งติดตั้งแล้วรันคำสั่งไม่สำเร็จ แนะนำให้ปิดและเปิด Terminal ใหม่ดูหนึ่งครั้งเพื่อให้ Terminal อัพเดต paths \*\*

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

1. เปิด Terminal หรือ PowerShell (สำหรับผู้ที่ใช้เครื่องตนเอง)
2. Change directory (cd) เข้าไปที่ folder “environments”
3. รัน "docker-compose up -d” เพื่อสร้าง Ubuntu จำลองขึ้นมาในเครื่องด้วย docker
4. **สำหรับผู้ที่ใช้ 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**

**A picture containing graphical user interface

Description automatically generated**

**C. Lab Quiz**

* 1. **Q> Ubuntu Version อะไร  
     A> 18.04**
  2. **Q> จากตั้งค่าใน docker-compose.yml จะสามารถ ssh เพื่อเข้าถึง host01,host02,host03 ได้ที่ port ใด  
     A> port 2201, 2202, 2203**

**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”

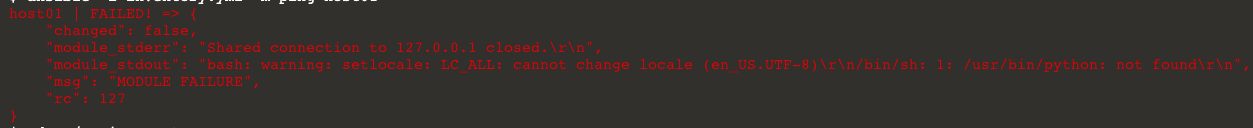
**D. รัน “ansible -i inventory.yml -m ping host01” แล้ว Capture Screen OutputText

Description automatically generated  
(\*\*\* cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)  
(\*\*\* ต้อง Ping host01 ได้ output เป็นสีเขียว และสถานะ SUCCESS)**

1. 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 OutputText

Description automatically generated  
(\*\*\* cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)  
(\*\*\* ต้อง Ping host01,host02,host03 ได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 ตัว)**

**\*\*\*\* ใครที่เจอปัญหาต่อไปนี้ ให้รันคำสั่ง 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 OutputA picture containing text

Description automatically generated  
(\*\*\* cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)  
(\*\*\* ต้องได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 hosts)**

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

**Port 8081: Graphical user interface, text, application

Description automatically generated**

**Port 8082: Graphical user interface, text, application

Description automatically generated**

**Port 8083: Graphical user interface, text, application

Description automatically generated  
(\*\*\* จะขึ้นเป็นหน้า Error ของ nginx ซึ่งถูกต้องแล้ว เพราะยังไม่ได้ใส่หน้าเว็บอะไรเข้าไปใน Nginx)**

**Lab 4 - สร้าง Ansible playbook สำหรับ Copy หน้าเว็บที่ทำขึ้นเพื่อนำไปใส่ใน 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 OutputGraphical user interface, text

Description automatically generated  
(\*\*\* cd ให้อยู่ใน folder lab ก่อนที่จะรันคำสั่ง)  
(\*\*\* ต้องได้ output เป็นสีเขียว และสถานะ SUCCESS ทั้ง 3 hosts)**

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

**Port 8081 :**

**Graphical user interface, text, application

Description automatically generatedPort 8082 :**

Graphical user interface, text, application

Description automatically generated

**Port 8083 : Graphical user interface, text, application

Description automatically generated**

**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/**](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**](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 ออกมาทำ ให้ Save ไฟล์เป็น PDF แล้วอัพโหลดส่งใน Google Classroom  
   หากทำใน GoogleDocs บนไฟล์ของ Google Classroom ให้ส่งด้วยไฟล์นั้นได้เลย