LAB EP1 Preparation

เนื่องจากการ Training ครั้งนี้จะไม่มี UDF Lab ให้ ทางผู้เข้าอบรบจำเป็นต้องเตรียมเครื่องสำหรับนำมาทำ Lab ด้วยตัวเอง ซึ่งจะมีขั้นตอนการเตรียมดังต่อไปนี้

Software ที่จะใช้ในการทำ Lab

1. VSCode: เป็น code editor ใช้สำหรับแก้ไข config files
2. VirtualBox: Hypervisor สำหรับสร้าง Virtual Machine
3. Vagrant

VM Spec

Master Node: 2vCPU RAM 2GB

Worker Node: 2vCPU RAM 2GB

ขั้นตอนการติดตั้ง

VSCode

Link Download https://code.visualstudio.com/download

ขั้นตอนการติดตั้งสำหรับ Windows : https://code.visualstudio.com/docs/setup/windows

ขั้นตอนการติดตั้งสำหรับ macOS : https://code.visualstudio.com/docs/setup/mac

VirtualBox

Link Download:

Windows: https://download.virtualbox.org/virtualbox/7.0.12/VirtualBox-7.0.12-159484-Win.exe

macOS: https://download.virtualbox.org/virtualbox/7.0.12/VirtualBox-7.0.12-159484-OSX.dmg

ขั้นตอนการติดตั้ง: เปิดไฟล์และติดตั้งตาม Standard software ทั่วไปได้เลย ไม่มี advance config

ติดตั้ง VirtualBox Extension สามารถใช้ไฟล์เดียวกันได้ทั้ง Windows และ macOS สามารถ download ได้จาก link :

https://download.virtualbox.org/virtualbox/7.0.12/Oracle_VM_VirtualBox_Extension_Pack-7.0.12.vbox-extpack

Vagrant

Link Download:

Windows: https://releases.hashicorp.com/vagrant/2.4.0/vagrant_2.4.0_windows_amd64.msi

macOS: https://releases.hashicorp.com/vagrant/2.4.0/vagrant_2.4.0_darwin_amd64.dmg

ขั้นตอนการติดตั้ง: เปิดไฟล์และติดตั้งตาม Standard software ทั่วไปได้เลย ไม่มี advance config

สำหรับ macOS สามารถติดตั้งผ่าน brew ได้โดยใช้

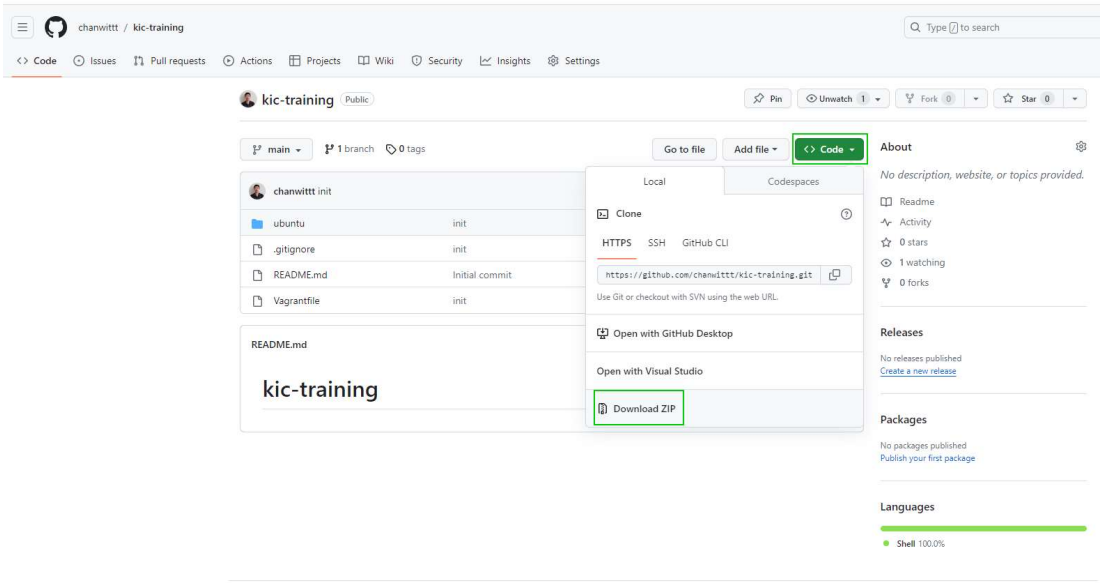
brew tap hashicorp/tap

brew install hashicorp/tap/hashicorp-vagrant

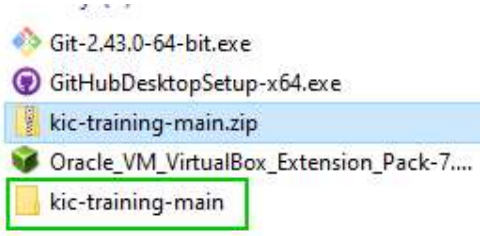
Build Kubernetes Cluster

ใน Lab นี้จะใช้ k8s จำนวน 2 Node โดยจะเป็น 1 Master node และ 1 Worker Node

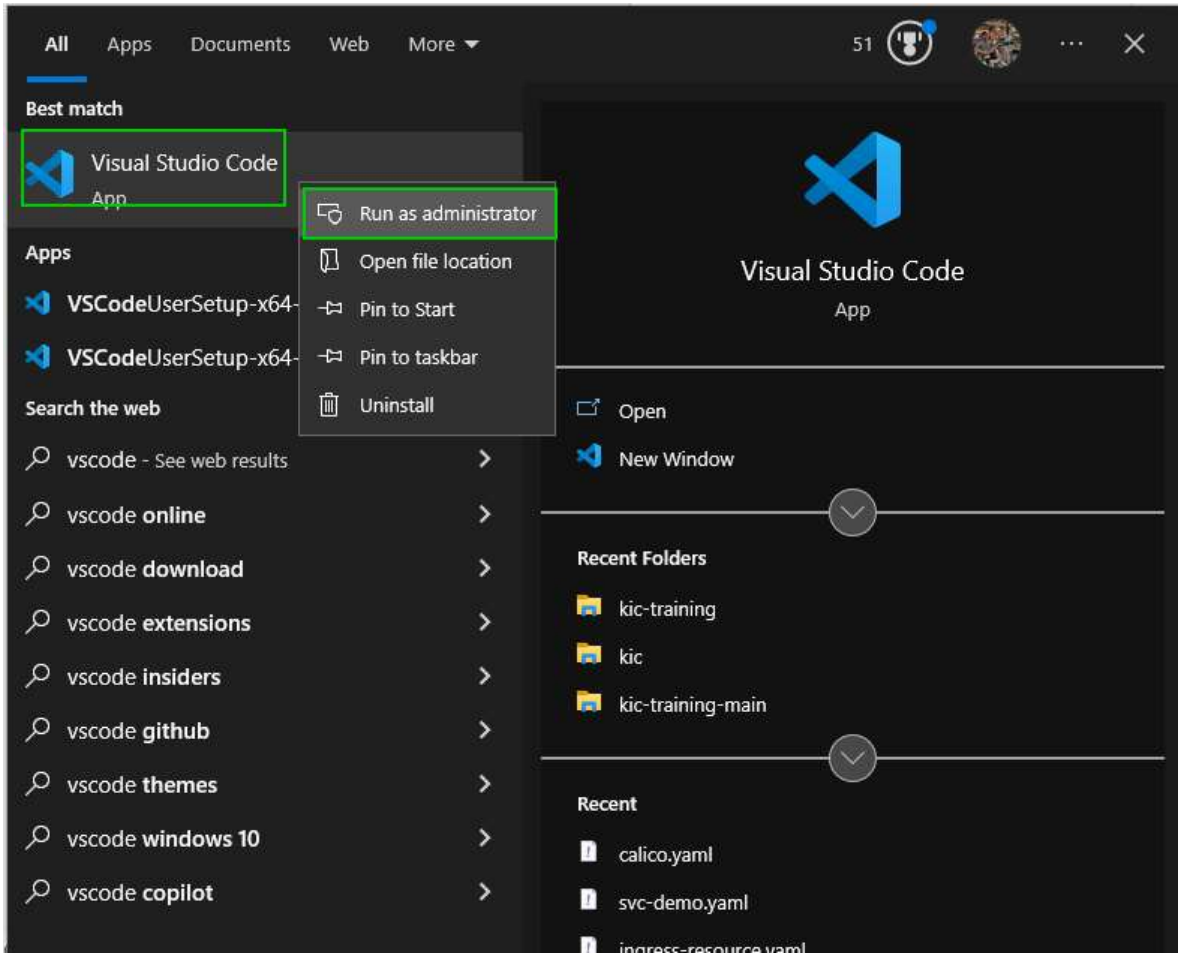
1. Download vagrant file ได้จาก https://github.com/chanwittt/kic-training click ที่ Code จากนั้นเลือก Download Zip file



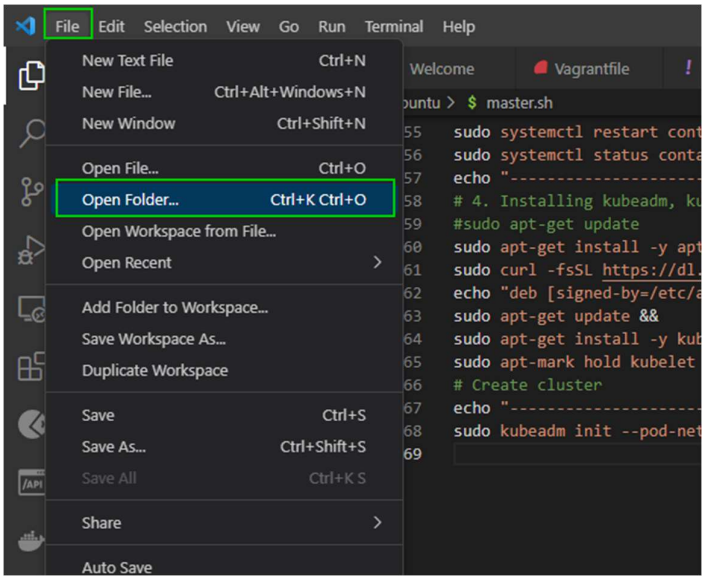
2. Unzip file kic-training-main



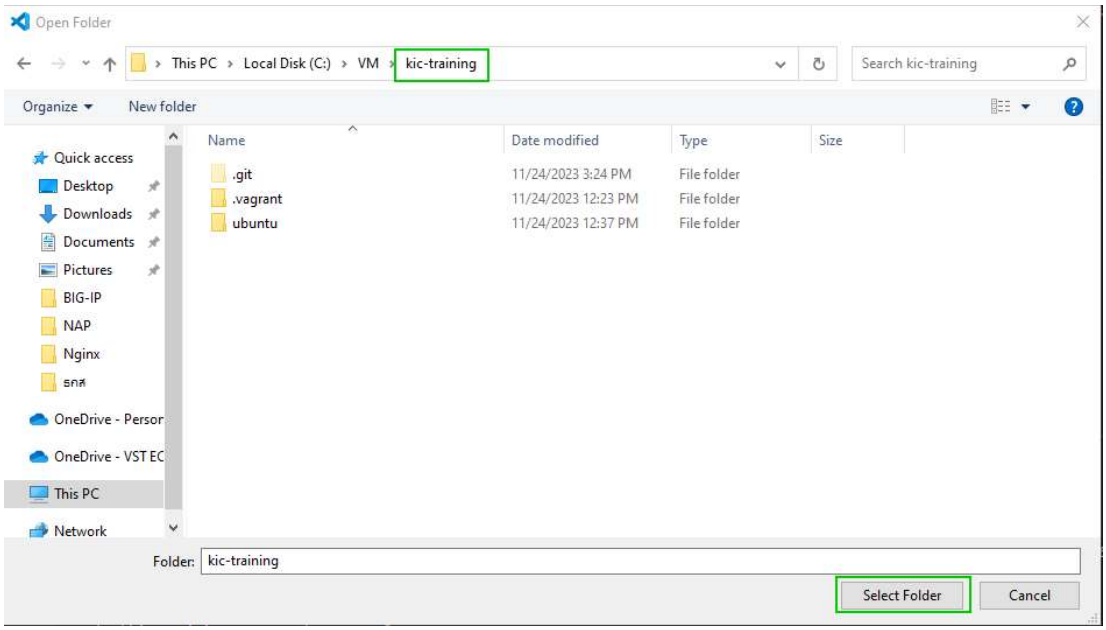
3. เปิด VSCode (Run as Administrator)



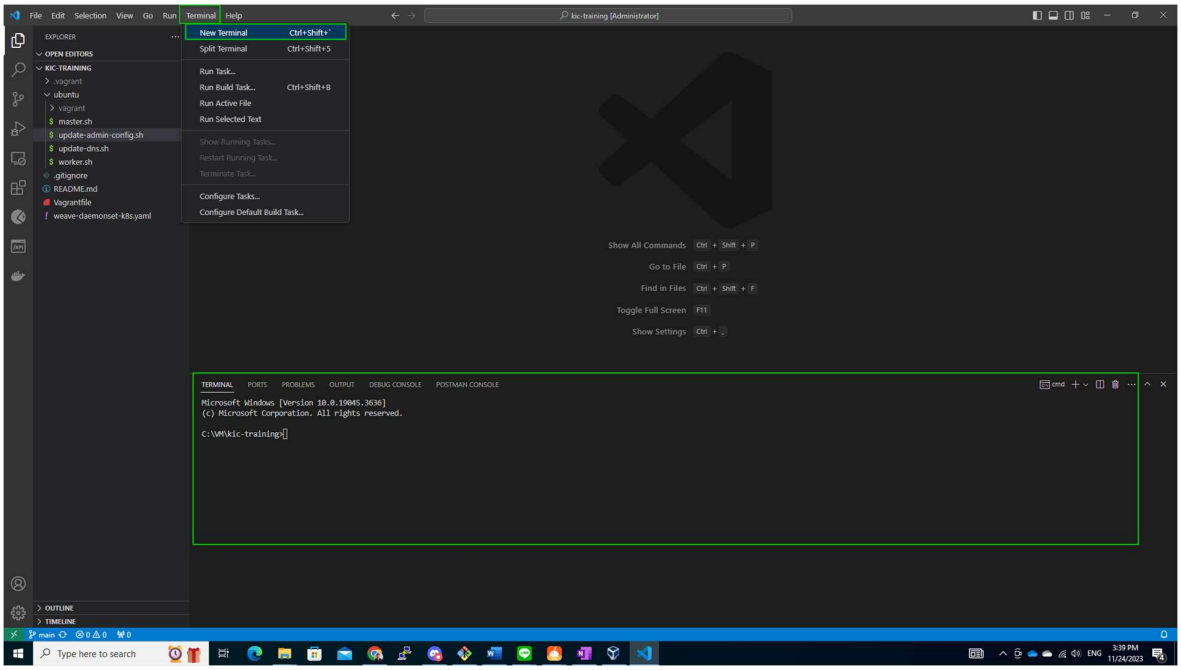
ใช้ VSCode เปิด folder ที่ Unzip ในขั้นตอนที่ 2 โดยไปที่เมนู File > Open Folder



เลือก Folder kic-training-main

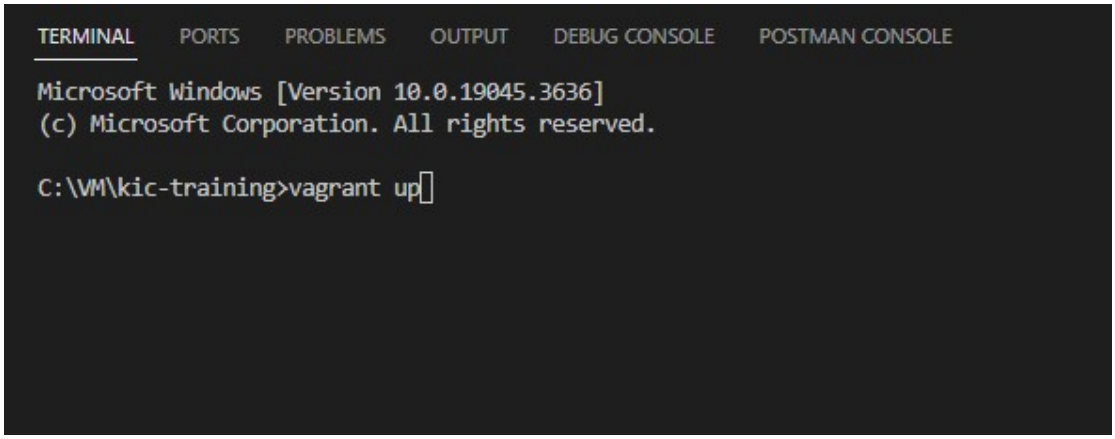


เปิด terminal โดยไปที่เมนู Terminal > New Terminal สามารถเพิ่ม terminal ได้โดยคลิกที่ + ด้านขาว



4. สร้าง K8S Cluster ขั้นตอนนี้จำเป็นต้องใช้ Internet เพื่อ Download images file โดยการพิมพ์ command:

vagrant up



Vagrant จะทำการสร้าง VM บน VirtualBox ติดตั้ง Ubuntu และติดตั้ง K8S ให้อัตโนมัติ ขั้นตอนนี้จะใช้เวลาค่อนข้างนาน ขึ้นอยู่กับความเร็วของ Internet

```
TERMINAL  PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  POSTMAN CONSOLE

Microsoft Windows [Version 10.0.19045.3636]
(c) Microsoft Corporation. All rights reserved.

C:\VM\kic-training>vagrant up
Bringing machine 'master' up with 'virtualbox' provider...
Bringing machine 'worker01' up with 'virtualbox' provider...
Bringing machine 'worker02' up with 'virtualbox' provider...
==> master: Importing base box 'ubuntu/focal64'...
==> master: Matching MAC address for NAT networking...
==> master: Setting the name of the VM: master
==> master: Clearing any previously set network interfaces...
==> master: Preparing network interfaces based on configuration...
    master: Adapter 1: nat
    master: Adapter 2: hostonly
==> master: Forwarding ports...
    master: 22 (guest) => 22101 (host) (adapter 1)
    master: 22 (guest) => 2222 (host) (adapter 1)
==> master: Running 'pre-boot' VM customizations...
==> master: Booting VM...
```

*หากไม่สามารถใช้ command vagrant ได้ให้ทำการ restart เครื่องและทดลอง run command อีกครั้ง

หากติดตั้งเรียบร้อยแล้วจะมีข้อความ Install successful

```
TERMINAL  PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  POSTMAN CONSOLE

    worker01: Selecting previously unselected package kubectl.
    worker01: Preparing to unpack .../6-kubectl_1.28.2-00_amd64.deb ...
    worker01: Unpacking kubectl (1.28.2-00) ...
    worker01: Selecting previously unselected package kubeadm.
    worker01: Preparing to unpack .../7-kubeadm_1.28.2-00_amd64.deb ...
    worker01: Unpacking kubeadm (1.28.2-00) ...
    worker01: Setting up conntrack (1:1.4.5-2) ...
    worker01: Setting up kubectl (1.28.2-00) ...
    worker01: Setting up ebtables (2.0.11-3build1) ...
    worker01: Setting up socat (1.7.3.3-2) ...
    worker01: Setting up cri-tools (1.26.0-00) ...
    worker01: Setting up kubernetes-cni (1.2.0-00) ...
    worker01: Setting up kubelet (1.28.2-00) ...
    worker01: Created symlink /etc/systemd/system/multi-user.target.wants/kubelet.service → /lib/systemd/system/kubelet.service.
    worker01: Setting up kubeadm (1.28.2-00) ...
    worker01: Processing triggers for man-db (2.9.1-1) ...
    worker01: kubelet set on hold.
    worker01: kubeadm set on hold.
    worker01: kubectl set on hold.
    worker01: -------------------------Install Worker Node successfully----------------------------------
```

**หากมี error ระหว่างการติดตั้งให้ลองใช้ command: vagrant destroy –force และ vagrant up ใหม่อีกครั้ง

SSH เข้าไปที่ Master Node เพื่อ config เพิ่มเติม โดยใช้ command

vagrant ssh master

```
    worker01: kubeadm set on hold.
    worker01: kubectl set on hold.
    worker01: -------------------------Install Worker Node successfully----------------------------------

C:\VM\kic-training>vagrant ssh master
```

Run commands

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

```
vagrant@master:~$ mkdir -p $HOME/.kube
vagrant@master:~$ sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
vagrant@master:~$ sudo chown $(id -u):$(id -g) $HOME/.kube/config
vagrant@master:~$
vagrant@master:~$
vagrant@master:~$ kubectl get node
NAME     STATUS     ROLES           AGE   VERSION
master   NotReady   control-plane   18m   v1.28.2
vagrant@master:~$
```

Verify master node โดยใช้ command kubectl get node สังเกตุ Status ยังเป็น NotReady เนื่องจากยังไม่ได้

ติดตั้ง Network Plug-in

ทำการติดตั้ง Network Plug-in โดยใช้ command

git clone https://github.com/chanwittt/kic-training

cd kic-training

kubectl apply -f weave-daemonset-k8s.yaml

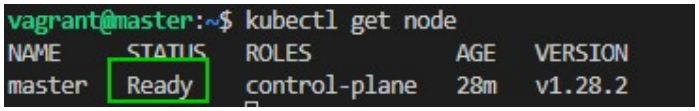
ขั้นตอนนี้จะใช้เวลาประมาณ 1-2 นาทีในการ deploy weaver สามารถตรวจสอบการทำงานโดยใช้ command

kubectl get pod -n kube-system รอจนกว่า pod ที่ชื่อ weaver จะเปลี่ยน status เป็น running

```
vagrant@master:~$ kubectl get pod -n kube-system
NAME                              READY   STATUS    RESTARTS      AGE
coredns-5dd5756b68-djmpl          1/1     Running   0             26m
coredns-5dd5756b68-ldbhz          1/1     Running   0             26m
etcd-master                       1/1     Running   0             26m
kube-apiserver-master             1/1     Running   0             26m
kube-controller-manager-master    1/1     Running   0             26m
kube-proxy-hr2x8                  1/1     Running   0             26m
kube-scheduler-master             1/1     Running   0             26m
weave-net-4sl4c                   2/2     Running   1 (35s ago)   45s
```

ตรวจสอบ Node status อีกครั้ง โดยใช้ command kubectl get node

Status ของ node จะเปลี่ยนเป็น Ready

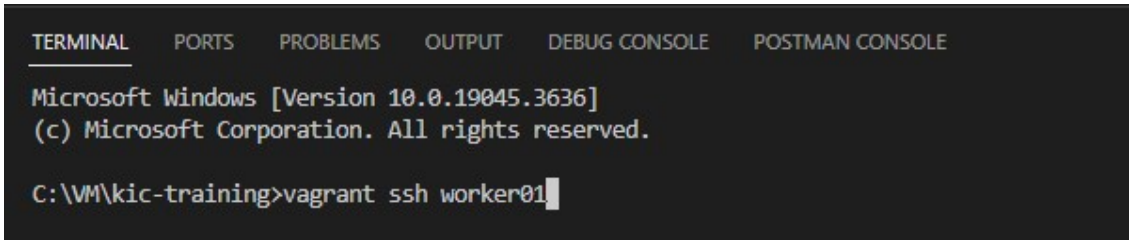


ทำการ get join command สำหรับการเพิ่ม worker node เข้ามาใน cluster โดยใช้ command

kubeadm token create --print-join-command และ copy result ใส nodepad หรือ vscode เพื่อนำไป run ใน worker node

ตัวอย่าง kubeadm join 192.168.100.2:6443 --token vh80my.437sxv5ycexsnm5s --discovery-token-ca-cert-hash sha256:0a434d3f18986d5b12c09da7eadd6ca3ef78c3bf58b68535b6bd74a746e7978a

SSH เข้าไปที่ Worker Node เพื่อ join Worker Node เข้าใน Cluster โดยใช้ command vagrant ssh worker01



นำ command ที่ได้จาก step ก่อนหน้านี้มา run บน worker01 โดยเพิ่ม sudo ด้านหน้า sudo kubeadm join 192.168.100.2:6443 --token vh80my.437sxv5ycexsnm5s --discovery-token-ca-cert-hash sha256:0a434d3f18986d5b12c09da7eadd6ca3ef78c3bf58b68535b6bd74a746e7978a

หาก join สำเร็จจะมีผลลัพธ์ตามรูปด้านล่าง

```
vagrant@worker01:~$ sudo kubeadm join 192.168.100.2:6443 --token ur3rly.ut8347ow1ev6hgmd --disco
[preflight] Running pre-flight checks
[preflight] Reading configuration from the cluster...
[preflight] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-co
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubelet-start] Writing kubelet environment file with flags to file "/var/lib/kubelet/kubeadm-fl
[kubelet-start] Starting the kubelet
[kubelet-start] Waiting for the kubelet to perform the TLS Bootstrap...

This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

ตรวจสอบอีกครั้งโดย SSH ไปที่ Master Node และ run command kubectl get node

```
vagrant@master:~$ kubectl get node
NAME       STATUS   ROLES           AGE     VERSION
master     Ready    control-plane   3d15h   v1.28.2
worker01   Ready    <none>          3m6s    v1.28.2
vagrant@master:~$
```

จะมี Node ที่ 2 เพิ่มเข้ามาใน Cluster

ทดสอบ Deploy NGINX ด้วย command

kubectl run nginx --image=nginx --labels="app=nginx"

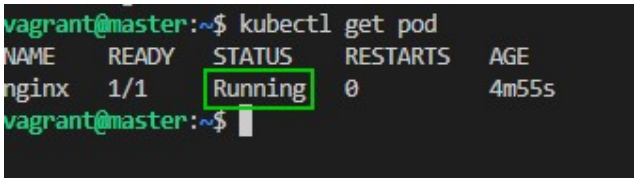
kubectl create service nodeport nginx --tcp=80:80 --node-port=30080

**Run บน Master Node

```
TERMINAL  PORTS  PROBLEMS  OUTPUT  DEBUG CONSOLE  POSTMAN CONSOLE
vagrant@master:~$ kubectl run nginx --image=nginx --labels="app=nginx"
pod/nginx created
vagrant@master:~$ kubectl create service nodeport nginx --tcp=80:80 --node-port=30080
service/nginx created
```

**การ deploy ครั้งแรกหากยังไม่มี image บนเครื่องจะใช้เวลานานเนื่องจากต้อง pull images จาก internet

ตรวจสอบ status ของ pod ด้วย command kubectl get pod



หาก status เป็น running แล้ว ลองทดสอบเข้า browser ใส่ URL http://192.168.100.3:30080 จะได้ผลลัพธ์ตามรูป

เพิ่มเติม

หากต้องการลบ app ที่ทดสอบ deploy ไป สามารถใช้ command

Kubectl delete pod nginx

Kubectl delete svc nginx

Shutdown VM ด้วย command

Vagrant suspend จะเป็นการ save state ล่าสุดไว้ (recommend) หรือ vagrant halt จะเป็นการ shutdown VM เลย

**ต้องออกจาก ssh ก่อน หรือเปิด terminal ใหม่ขึ้นมาเพื่อ run command บนเครื่อง host windows หรือ mac

หากต้องการลบ VM สามารถใช้ command vagrant destroy หรือ vagrant destroy --force