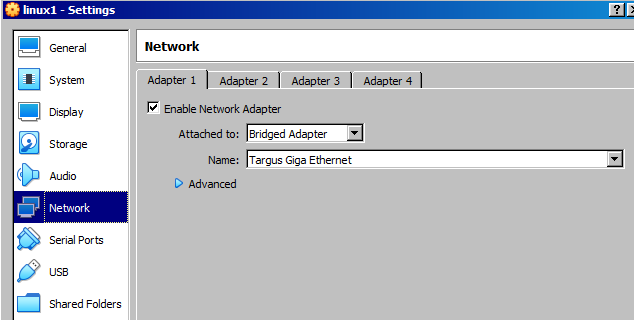
# Setting up K8s cluster locally on windows in VM

## Software Requirement

1. Get virtual box (VirtualBox-6.0.14-133895-Win.exe)
2. Download a copy of Ubuntu iso (ubuntu-18.04.3-desktop-amd64.iso)
3. Use this STS-TokenGen.pyand copy to shared drive and share with VM
4. Install aws iam authenticator <https://docs.aws.amazon.com/eks/latest/userguide/install-aws-iam-authenticator.html> ( to be done later)

## Step 1 Install Virtual Box

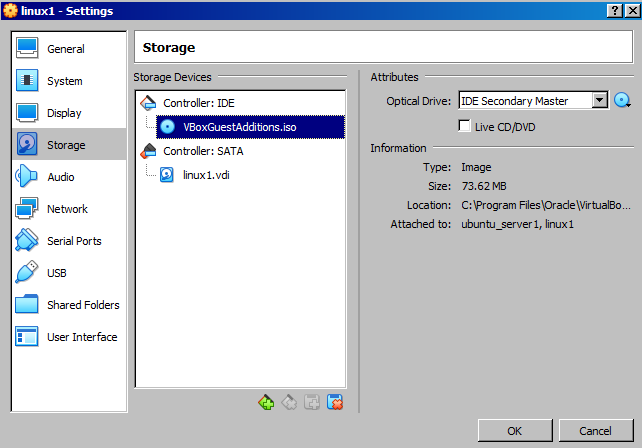
Setup virtual box with bridge networking, please make sure to use the right network to bridge to ex:



## Step 2 Create VM from Ubuntu image

Create a virtual machine with at least 4 gig ram and 40gb space.

Boot VM with Ubuntu iso as the image. To select image to go vm settings>storage and change the ide cd/dvd.



## Step 3 Setup networking

**Set Proxy Server**

Set proxy server in your bashrc and terminal . If you use sudo or bash sudo please do in that shell as well.

SET HTTP\_PROXY = "http:// username:password @corp-eq5-proxy.mhc:8080"

export HTTPS\_PROXY=$HTTP\_PROXY

export NO\_PROXY=".spglobal.com"

Or try with this:

export HTTP\_PROXY=http:// username:password@sqpx.mge.spratingsvpc.com:10041

export HTTPS\_PROXY=$HTTP\_PROXY

export NO\_PROXY=".spglobal.com"

**Setup DNS**

vi /etc/resolv.conf

Add dns server from your local desktop (on windows do ipconfig /all)

**Setup APT**

vi /etc/apt/apt.conf

Add proxy server as using these instructions <https://www.serverlab.ca/tutorials/linux/administration-linux/how-to-set-the-proxy-for-apt-for-ubuntu-18-04/>

## At this point machine is ready to install additional software

## Step 4 Install required software using APT

sudo apt update

sudo apt install docker.io

sudo apt install yum

sudo apt install python3

sudo apt install anaconda

sudo apt install curl

sudo apt install npm

sudo apt install node

sudo apt install -y dkms build-essential

## Step 5 Install Anaconda

**Use wget to Download and Install anaconda from** <https://repo.anaconda.com/archive/Anaconda3-2019.10-Linux-x86_64.sh>

**Set anaconda to path**

export PATH=/home/mav/anaconda3/bin/:$PATH

**Install additional packages and libs.**

conda install lxml

conda install bot3

conda install boto3

pip install lxml

pip install lxml

## Step 5 Install aws software

sudo apt install snapd

Sudo snap install aws-cli –classic

## Step 7 Install k8s

**Using snap to install microk8s**

snap install microk8s –classic

OR

Install full k8s using kadm instructions from kubernetes site.

## Step8 Install serverless

snap install --edge node --classic

snap install --edge npm –classic

npm install npm@latest –g

npm install -g serverless

## Step 9 Disable Ubuntu Gui (optional if vm is slow)

sudo systemctl set-default multi-user.target

sudo vi /etc/default/grub

sudo apt remove plymouth-theme-ubuntu-text

sudo update-grub2

vi /etc/default/grub

And follow instructions here to disable xserver:[**https://askubuntu.com/questions/16371/how-do-i-disable-x-at-boot-time-so-that-the-system-boots-in-text-mode**](https://askubuntu.com/questions/16371/how-do-i-disable-x-at-boot-time-so-that-the-system-boots-in-text-mode)

#reboot

shutdown -r now

## Step 10 Install aws kubectl

<https://docs.aws.amazon.com/eks/latest/userguide/install-kubectl.html#w243aac27b9b9b3>

## Step 11: Test the VM

export AWS\_PROFILE=031791575028-ADFS-MHF-PowerUser

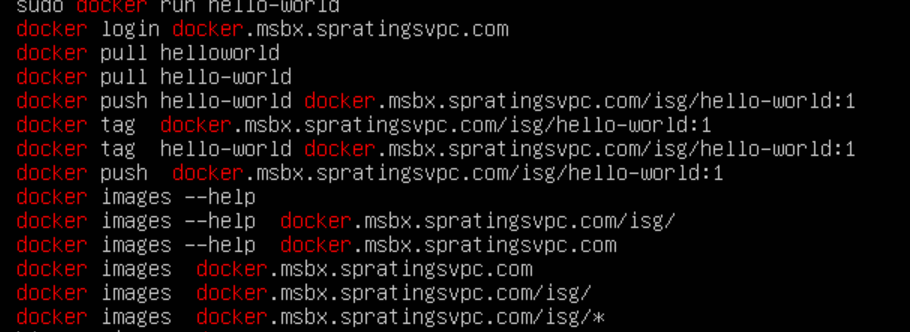
python ./STS-TokenGen.py

aws s3 ls --profile 031791575028-ADFS-MHF-PowerUser

aws eks --region us-east-1 update-kubeconfig --name isg-eks-poc-cluster

kubectl get svc

## Step 12 Working with harbor container repo



## Step 13 Installing virtual box tool inside guest (vm image)(TODO)

## Step 14 Sharing folder as mount inside VM. (TODO)