SDN project: Tutorial on Environment Set-Up

1. Brief Introduction

This is a tutorial on how to set up the sdn project's environment on Virtual Machine (VM) we provided. In this tutorial, we show detailed steps and some common problems on environment set-up which are not included in the document of *sdn-vagrant-vm.pdf*. You are highly recommended to go through this tutorial before you start your project so that you won't get troubled by insignificant problems.

In this tutorial, we **only focus on the provider as virtual box**. If you would like to use other providers, we are afraid that it is not able to provide too much help.

2. Downloading

Please download all the softwares and documents listed to make sure you are able to build up a VM

- [ ] **VirtualBox**: https://virtualbox.org
- [ ] **Vagrant**: https://www.vagrantup.com
- [ ] **VM Box**: *cs-dev-vm.box* on https://owncloud.sustech.edu.cn/s/ZfxpnDKwB4pb2ry
- [ ] **Vagrant Profile**: *Vagrantfile* (without extension) on https://owncloud.sustech.edu.cn/s/ZfxpnDKwB4pb2ry

- [ ] **Starter Repository**: The folder *sdn-script/* on https://owncloud.sustech.edu.cn/s/ZfxpnDKwB4pb2ry

3. Modify the Vagrant Profile (*Vagrantfile*)

You are **required to change the location of box** in the profile to make the VM work. As for following two other configurations, we recommend but not force you to change them. By the way, you are free to modify the profile as you like as long as it works best with your machine.

3.1 Change the location of box

```
22    # You need to change the box's location here
23    config.vm.box = "cs-dev-vm.box"
```

Change the "cs-dev-vm.box" as the box location in your computer

3.2 * Enable synchronous folders

This is set to synchronize the specific folder on VM with the folder in your computer so that you could code on your own computer with familiar IDE or Editors. See comments for details

```
50    # Share an additional folder to the guest VM. The first argument is
51    # the path on the host to the actual folder. The second argument is
52    # the path on the guest to mount the folder. And the optional third
53    # argument is a set of non-required options.
54
55    # You need to change folders' location here
56    config.vm.synced_folder "sdn-script", "/home/vagrant/sdn-code"
57
```

3.3 * Enable X11 forward

You could change the profile to allow it using X11 forwarding so that you do not need to add the parameter `-X` or `-Y` every time you connect to the VM. You should add these two statements on the file:

```
1  config.ssh.forward_agent = true
2  config.ssh.forward_x11 = true
```

4. Work on VM

4.0 * Configure environment variables manually (windows only)

Since **vagrant** automatically configure the environment, you only need to make new configuration for **virtual box**. Find the directory which contains `VBoxManage.exe` and add it into your PATH (e.g `E:\\virtual box`). Then **restart** the cmd or other command line tools if you have opened it before. Type `VBoxManage` to see if it works well

4.1 Get access to the VM

In the directory with `Vagrantfile`,

- Type in `vagrant up` to set up the VM for the first time or when the VM is shut down
- Then type in `vagrant ssh` to get access to the VM

```
(base) wu@Wus-MacBook-Pro vagrant % ls
Vagrantfile
(base) wu@Wus-MacBook-Pro vagrant % vagrant ssh
Welcome to Ubuntu 18.04.1 LTS (GNU/Linux 4.15.0-29-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Tue Dec 10 10:32:02 UTC 2019

  System load:  0.0               Processes:           93
  Usage of /:   4.2% of 61.80GB   Users logged in:     0
  Memory usage: 13%               IP address for eth0: 10.0.2.15
  Swap usage:   0%

 * Overheard at KubeCon: "microk8s.status just blew my mind".

     https://microk8s.io/docs/commands#microk8s.status

 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

368 packages can be updated.
199 updates are security updates.


Last login: Tue Dec 10 08:47:01 2019 from 10.0.2.2
vagrant@vagrant:~$
```

4.2 Install extra libraries

Use `sudo apt-get install mininet python3-ryu iputils-arping` to download and install extra libraries and please **do not use other package manager to install** these (e.g. pip etc.). Also, please **do not install other python libraries.**

4.3 Run the code

In the same terminal/cmd, run the Ryu application as follow:

```
vagrant@vagrant:~/sdn-code$ ryu-manager --observe-links shortest_paths.py
loading app shortest_paths.py
loading app ryu.controller.ofp_handler
loading app ryu.topology.switches
loading app ryu.controller.ofp_handler
instantiating app shortest_paths.py of ShortestPathSwitching
instantiating app ryu.controller.ofp_handler of OFPHandler
instantiating app ryu.topology.switches of Switches
```

Then open a new terminal, get access to the VM and run the python script as follow:

```
vagrant@vagrant:~/sdn-code$ sudo python run_mininet.py single 3
*** Creating network
*** Adding controller
Connecting to remote controller at 127.0.0.1:6653
*** Adding hosts:
h1 h2 h3
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1)
*** Configuring hosts
h1 h2 h3
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Sending ARPing from host h1
*** Sending ARPing from host h2
*** Sending ARPing from host h3
*** Starting CLI:
mininet>
```

5. * Some Helpful Commands of Vagrant

Here lists some useful vagrant commands, you could find more information on Vagrant docs [1]

- `vagrant up`: **Turn on** the virtual machine. Whatever you want to do on VM, you must turn on it at first
- `vagrant ssh`: Once you have turned on the VM, you could **connect** to it with this command
  - `vagrant ssh -- [extra_ssh_args]`: allow you pass extra arguments to the ssh
    - `vagrant ssh -- -v`: You could use command to know debug message of your VM.

      -v Verbose mode. Causes **ssh** to print debugging messages about its progress. This is helpful in debugging connection, authentication, and configuration problems. Multiple **-v** options increase the verbosity. The maximum is 3.
- `vagrant reload`: If you have **changed** the *Vagrantfile*, you should remember to use this command to let the change **take effect**
- `vagrant halt`: **Turn off** the VM. If you do not use this command, the VM will be always running on the back-end in which you could use `ps -ef | grep VBOx` to capture or see it on `top` in linux/Unix/MacOs, or you could use `Taskmgr.exe` in Windows.
- `vagrant destroy`: **Destroy** the VM. If you want to **initiate the VM**. You could just destroy it and then `vagrant up` which gives you a brand new VM.

6. Run Wireshark

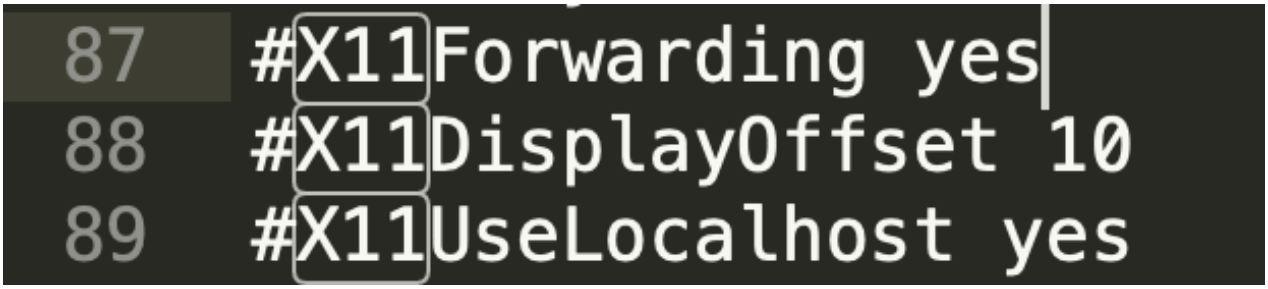
Different operating systems have their own constrains and requirements. Of course, you could install the GUI Ubuntu `sudo apt-get install --no-install-recommends ubuntu-desktop gnome-terminal virtualbox-guest-dkms` which might takes more memories and ignore the following instructions.

6.1 MacOS

- Download **XQuartz**: Since X11 is no longer included with Mac [2] , if your Mac OS X is later than 10.7, please download **XQuartz** [3]
  - In the terminal, use `print $DISPLAY` or `echo $DISPLAY` to make sure you installed it properly.

```
(base) wu@Wus-MacBook-Pro ~ % print $DISPLAY
/private/tmp/com.apple.launchd.uqRzxEgrdD/org.macosforge.xquartz:0
```

- Change the ssh profile: Change the *ssh profile* in */etc/ssh/sshd_config*, allowing **X11Forwang** in ssh.



- Get access to the VM (`vagrant ssh -- -X` or `vagrant ssh -- -Y` or `vagrant ssh` for the configured profile)
  - Then you should see the XQuartz launched for `vagrant ssh -- -X` or `vagrant ssh -- -Y`
  - You could also type `echo $DISPLAY` to see the display, it should be `localhost:10.0` where `10.0` is the DisplayOffset motioned before
  - Then type in `Wireshark`, you would see the GUI of the application (Actually, first the XQuartz launched and the Wireshark runs on it)

6.2 Ubuntu

Just **check the configuration of */etc/ssh/sshd_config*** and make sure you get access to the VM **with X11 forward mode**

6.3 Windows

It is more complicated than MacOS X and Ubuntu to enable X11 Forwarding on the Windows, because the feasible solution using **openssh** and X11 server only has not been found yet. Hence, we recommend you to use GUI VM instead.

However, if you are interested in how to forward X11 applications, here goes some reference:

- Firstly, you need to be able to get access into the VM by using `ssh` command rather than `vagrant ssh` we used before [4] .
- Then, you could combine the X11 server (e.g Xming) and ssh client (e.g putty, Xshell) to

complete the connection with X11 forwarding [5].

7 Common Problems

7.1 `vagrant up` error: no usable default provider

- If you have already dowloaded virtual box or other providers
  - Please check your vagrant **vision** using `vagrant --version` and make sure it is latest (2.2.6)
  - Please make sure the Virtual Box has **been added into the PATH** (refer to 4.0)
- If not, please download virtual box or other providers

---

1. https://www.vagrantup.com/docs/cli/↩
2. https://support.apple.com/en-us/HT201341↩
3. https://www.xquartz.org/index.html↩
4. https://www.jianshu.com/p/ca73a93d5843↩
5. https://hpcf.umbc.edu/general-productivity/running-x-windows-programs-remotely/↩