User Manual of the Pre-built Lubuntu Virtual Machine

1 Overview

Using VirtualBox, a pre-built virtual machine image for Lubuntu has been created. This virtual machine will be used for all labs in this course. The purpose of this document is to describe the configuration of this virtual machine. The virtual machine is available online from the course web page.

Additional documentation on VirtualBox can be found on the VirtualBox home page:
https://www.virtualbox.org/manual/

2 Virtual Machine Configuration

While following any of the subsections below, if you feel like starting over, see Section 3 for instructions on cleaning up.

2.1 Connecting to a Development Machine

The VirtualBox software has been installed on the mcXX machines managed by the CS department. These machines are:

- mc19.cs.purdue.edu
- mc20.cs.purdue.edu
- mc21.cs.purdue.edu

You must have received an email with machine and port allocation. You are supposed to use the mcXX machine allocated to you.

Each of you have been given space in the directory /scratch/$USERID where $USERID should be replaced with your CS user account name. All work for the lab assignments should be done in this directory. Feel free to create sub directories as needed.

NOTE: Do NOT use your home directory for saving virtual machine images. Use the space allocated in /scratch/$USERID

You will first need to connect to the mcXX machine using the ssh command. To do that, open a terminal on your workstation and run the command:

```
ssh mcXX.cs.purdue.edu
```

Replace mcXX with the name of the machine allocated to you.

Once connected to the mcXX machine (you will need to enter your CS account user name and password), you can start your virtual machine setup.

2.2 Configuration of the Virtual Machine

The main configuration of the virtual machine is summarized here. Using VirtualBox you can adjust the configuration according to the resources of your host machine (e.g., you can assign more memory to this virtual machine if your host machine has enough memory):

- Operating system: `Lubuntu` with the `Linux` kernel v4.2.0-35-generic.
- Memory: 512M RAM.
- Disk space: Maximum 5G disk space.

A single account has been created for the virtual machine. The user and password for the account is:

- User ID: `cs528user`, Password: `cs528pass`

2.3 Importing the Virtual Machine

To import the virtual machine image:

1. It is suggested to create a separate directory for each virtual machine instance to keep things organized. For example for lab1 you can create a directory for a virtual machine image using the following:

   ```
   mkdir -p /scratch/$USERID/lab1/$VMNAME/
   ```

   For example, for user `jdoe` creating the vm with name `CS528_vm1`, run:

   ```
   mkdir -p /scratch/jdoe/lab1/CS528_vm1/
   ```

2. Import the virtual machine image using the following command:

   ```
   VBoxManage import \
       /homes/cs528/vms/cs528_vm.ova \
       --vsys 0 \
       --vmname $VMNAME \
       --unit 9 \
       --disk /scratch/$USERID/lab1/$VMNAME/$VMNAME.vmdk
   ```

   For example, for user `jdoe` importing the vm with name `CS528_vm1`, run:

   ```
   VBoxManage import \
       /homes/cs528/vms/cs528_vm.ova \
       --vsys 0 \
       --vmname CS528_vm1 \
       --unit 9 \
       --disk /scratch/jdoe/lab1/CS528_vm1/CS528_vm1.vmdk
   ```

   NOTE: The import of the virtual machine image will take a while so be patient.

   NOTE: Make sure you specify a path within your scratch directory for the virtual disk image. Do NOT save it to your home directory.

Once the virtual machine is imported, it needs to be registered with `VirtualBox`. The import will create a `VirtualBox` configuration file (.vbox) in your home directory in the following location:

`/homes/$USERID/VirtualBox\ VMs/$VMNAME`

Where $USERID is your log in identity and $VMNAME is the name you gave your virtual machine.

2.4 Registering the Virtual Machine

Register the virtual machine using the following command:

```
VBoxManage registervm \
    /homes/$USERID/VirtualBox\ VMs/$VMNAME/$VMNAME.vbox
```

NOTE: Make sure you replace $USERID with your user and $VMNAME with the name of your virtual machine. For example if you created a virtual machine called CS528_vm1 and your user name is jdoe the register command would be:

```
VBoxManage registervm \
    /homes/jdoe/VirtualBox\ VMs/CS528_vm1/CS528_vm1.vbox
```

To see if your virtual machine was successfully registered use the following command:

```
VBoxManage list vms
```

NOTE: You will only need to run the `registervm` command after importing the first virtual machine.

NOTE: If you see an error like the following

```
Trying to open a VM config
'/homes/jdoe/VirtualBox\ VMs/CS528_vm1/CS528_vm1.vbox'
which has the same UUID as an existing virtual machine
```

you can proceed if $VMNAME shows up when you run `VBoxManage list vms`.

2.5 Importing Multiple Virtual Machines

When importing the same virtual machine image multiple times it is important to give each:

1. Their own virtual machine name (set using the `--vmname` flag - see above)
2. Their own directory for storing the virtual disk image in your `/scratch` directory

To import multiple virtual machines, follow the steps above making sure to create a separate directory in your `/scratch` directory and giving each virtual machine their own name.

NOTE: When you import multiple virtual machines you do not need to run the `registervm` command after importing the first virtual machine.

2.6 Network Setup

Currently the "Network connection" is set to "NAT Network" which is a special type of Network Address Translation (NAT) that also creates an internal network for all of your virtual machines. This allows direct communication between the virtual machines as well as allows them to communicate with the outside world. Before you can start your virtual machine you need to set up the natnetwork. This is done using a couple of commands.

The supplied virtual machine image contains one network adapter that is set to be a part of the "cs528net" network. However you will need to create and use your own internal network for your virtual machines. To create the network run the following command:

```
VBoxManage natnetwork add --netname $USERID \
    --network "192.168.15.0/24" --enable --dhcp on
```

NOTE: Make sure you replace $USERID with your user ID. For example, if your user id was jdoe the command to run would be:

```
VBoxManage natnetwork add --netname jdoe \
    --network "192.168.15.0/24" --enable --dhcp on
```

This will create an internal network with the name jdoe and enable Dynamic Host Configuration Protocol (DHCP) for the 192.168.15.0/24 range of internet addresses.

The network must then be started in order to be used. Start the network using the following command:

```
VBoxManage natnetwork start --netname $USERID
```

To view information about the internal network run the following command:

```
VBoxManage list natnets
```

It should print out the following:

```
NetworkName:    $USERID
IP:             192.168.15.1
Network:        192.168.15.0/24
IPv6 Enabled:   No
IPv6 Prefix:
DHCP Enabled:   Yes
Enabled:        Yes
loopback mappings (ipv4)
        127.0.0.1=2
```

NOTE: $USERID would be replaced with the name of your network (your user ID) in the output.

Recall that when the network was created, the `-dhcp on` flag was used to enable a DHCP server for the network. `VirtualBox` has an internal process that manages DHCP for your virtual machines automatically. To view information about the DHCP server run the following command:

```
VBoxManage list dhcpservers
```

It should print out the following:

```
NetworkName:    $USERID
IP:             192.168.15.3
NetworkMask:    255.255.255.0
lowerIPAddress: 192.168.15.4
upperIPAddress: 192.168.15.254
Enabled:        Yes
```

This output shows the network name for the DHCP server and the internet address (IP) of the server itself on the virtual network among other useful information.

NOTE: the `lowerIPAddress` and `upperIPAddress` fields show IP address range that the DHCP server will give out to clients.

Now modify the virtual machine to use your NAT network by running the following command:

```
VBoxManage modifyvm $VMNAME --nat-network1 $USERID
```

Replace $VMNAME with the name of the virtual machine and $USERID with your user ID (the name of your NAT network). For example if your user ID is `jdoe` and you used `CS528_vm1` for the virtual

machine, the command would be:

```
VBoxManage modifyvm CS528_vm1 --nat-network1 jdoe
```

2.7 Running the Virtual Machine

Once the virtual machine image is imported and the network configured, you can view the status by running the following command:

```
VBoxManage showvminfo $VMNAME
```

Replacing $VMNAME with the name of your virtual machine image. To see the the names of all your virtual machine images run the command:

```
VBoxManage list vms
```

To start the virtual machine use the following command:

```
VBoxManage startvm $VMNAME --type headless
```

This will start the virtual machine $VMNAME. Replace $VMNAME with the actual name of your virtual machine.

After the VM has started (in about 1-2 minutes), run the following command for the IP address assigned by the DHCP server.

```
VBoxManage guestproperty enumerate $VMNAME
```

NOTE: This shows a lot of details along with the IP address.

2.8 Port Forwarding

You need to set up NAT port forwarding to use `ssh` command to access the virtual machine command line. You should have been given a range of port numbers from the TA which you can use for this purpose. To set the SSH port forwarding for a VM, use the following command:

```
VBoxManage natnetwork modify --netname $USERID \
    --port-forward-4 "guestssh1:tcp:[]:$TARGETPORT:[$VMIP]:22"
```

Replacing $VMIP with the IP address of your VM image and $TARGETPORT with one of your assigned port numbers. If you look at the output from the `VBoxManage list dhcpservers` command, the first IP that the DHCP server gives out will be the IP for the first VM. For example, if you are setting up your first VM, it will have IP address 192.168.15.4 . If one of your port numbers are 10023, use the following command:

```
VBoxManage natnetwork modify --netname $USERID \
--port-forward-4 "guestssh1:tcp:[]:10023:[192.168.15.4]:22"
```

The above command will forward SSH traffic from localhost (127.0.0.1) port 10023 to the VM with IP 192.168.15.4 on ssh port 22.

NOTE: Some parts of your assignment require multiple virtual machines to be running at the same time. You will need to use a different port number for SSH port-forwarding to each running virtual machine. Also, for each port-forwarding rule you have to use a different 'rule name'. The above example uses the name 'guestssh1'. It is suggested that you use 'guestssh2' for the next VM and so on.

NOTE: The IP addresses are assigned to the VMs when they are started for the first time. So, the machine needs to be started once (Section 2.6 explains how to run the VMs) to know the IP address assigned to it. Otherwise, you can guess the IP address of the next machine, because the dhcp server here allocates IP addresses in sequence. For example, the first machine gets IP address 192.168.15.4, the second machine gets IP address 192.168.15.5 etc.

A port forwarding rule can be deleted by using the following command:

```
VBoxManage natnetwork modify --netname $USERID \
  --port-forward-4 delete $RULENAME
```

Replacing $RULENAME with the name of the forwarding rule to be deleted. For example:

```
VBoxManage natnetwork modify --netname $USERID \
--port-forward-4 delete guestssh1
```

Will delete the port forwarding rule `guestssh1`.

2.9 SSH into the virtual machine

To connect to the virtual machine once it is running use the following ssh command:

```
ssh -p $TARGETPORT cs528user@localhost
```

Replacing $TARGETPORT with the port number you used in your port-forwarding rule.

Recall that the VM password is `cs528pass`

2.10 Force Power Off of a Virtual Machine

If you find that you need to reboot or force power off a virtual machine, this can be done using the following command:

To force a power off use the following command:

```
VBoxManage controlvm $VMNAME poweroff
```

To force a reboot use the following command:

```
VBoxManage controlvm $VMNAME reset
```

Replace $VMNAME with the actual name of your virtual image.

NOTE: The poweroff command will have the same affect as pulling the power on an actual computer while the reset command has the same affect as pressing the hardware reset button on an actual computer. All unsaved information will be lost.

NOTE: The poweroff/reset command also releases the DHCP assigned to the VM. When restarting the VMs again, get the IPs of the VM, and use the correct port associated with the port forwarding rule to SSH into the VM.

Please make sure that you do not leave virtual machines running on the 'mc' machines when you are done working.

3 Cleaning Up (Optional)

3.1 Remove Old Virtual Machines

If you want to have a clean VirtualBox configuration before you begin this lab, you can follow the steps mentioned below:

1. Start by making sure you have no running virtual machines on your mcXX machine. To do that run the following command:

   ```
   VBoxManage list runningvms
   ```

   If this command shows any running virtual machines stop them with the following command:

   ```
   VBoxManage controlvm $VMNAME poweroff
   ```

   Replace `$VMNAME` with the name of the virtual machine to stop.

2. Now make sure that there are no VirtualBox processes running:

   ```
   ps aux | grep -v grep | grep VBox | grep $USER
   ```

   If this command output any running VBox processes make sure to kill them.

3. Now that you do not have any running virtual machines, to ensure you have a clean configuration run the following command and make it doesn't output anything:

   ```
   VBoxManage list vms
   ```

   If the command outputs the existence of any virtual machines make sure you unregister and delete them with the following command:

   ```
   VBoxManage unregistervm $VMNAME --delete
   ```

   Replace `$VMNAME` with the name of the virtual machine to delete.

   NOTE: this command will remove all contents of the virtual machine. Make sure to backup anything you want to save from the virtual machine before unregistering it.

3.2 Remove Old NAT Networks

Make sure you have removed all existing NAT networks before you begin the lab. To ensure you have a clean network configuration run the following command and make sure it doesn't output anything:

```
VBoxManage list natnets
```

If the command outputs the existence of any NAT network make sure you remove it with the following commands:

```
VBoxManage natnetwork stop --netname $NETNAME
VBoxManage natnetwork remove --netname $NETNAME
VBoxManage dhcpserver remove --netname $NETNAME
```

Replace `$NETNAME` with the name of the network you want to remove.

3.3 Stop VirtualBox Processes

At this point there shouldn't be any VirtualBox processes running, but to make sure you have a clean configuration run the following command to check for VirtualBox processes:

```
ps aux | grep $USER | grep VBox | grep -v grep
```

If any VirtualBox processes are listed kill them with the following command:

```
kill PID
```

Replace `PID` with the process identifier output from the `ps` command.

4 Submission

This exercise does not require you to submit anything. However, you should finish this exercise because you will need this setup for most of the lab assignments in this course.