EPAM University Programs

DevOps education program

Module 2 Virtualization and Cloud Basic

TASK 2.2

PART 1. WORK WITH VIRTUALBOX

1. First run VirtualBox and Virtual Machine (VM).

1.1 Get acquainted with the structure of the user manual VirtualBox [1]

1.2 From the official VirtualBox site [2] download the latest stable version of VirtualBox according to the host operating system (OS) installed on the student's workplace. For Windows, the file may be called, for example, VirtualBox-6.1.10-138449-Win.exe. Install VirtualBox.

1.2 Download the latest stable version of Ubuntu Desktop or Ubuntu Server from the official site [3].

1.3 Create VM1 and install Ubuntu using the instructions [1, chapter 1.8]. Set machine name as "host machine name"_"student last name"

1.4 Get acquainted with the possibilities of VM1 control - start, stop, reboot, save state, use Host key and keyboard shortcuts, mouse capture, etc. [1, ch.1.9].

1.5 Clone an existing VM1 by creating a VM2 [1, ch.1.14].

1.6 Create a group of two VM: VM1, VM2 and learn the functions related to groups [1, п.1.10].

1.7 For VM1, changing its state, take several different snapshots, forming a branched tree of snapshots [1, ch.1.11].

1.8 Export VM1. Save the *.ova file to disk. Import VM from *.ova file [1, ch.1.15].

2. Configuration of virtual machines

2.1 Explore VM configuration options (general settings, system settings, display, storage, audio, network, etc.).

2.2 Configure the USB to connect the USB ports of the host machine to the VM [1, ch.3.11].

2.3 Configure a shared folder to exchange data between the virtual machine and the host [1, ch.4.3].

2.4 Configure different network modes for VM1, VM2. Check the connection between VM1, VM2, Host, Internet for different network modes. You can use the ping command to do this. Make a table of possible connections.

3. Work with CLI through VBoxManage.

3.1 Run the cmd.exe command line.

3.2 Examine the purpose and execute the basic commands of VBoxManage list, showvminfo, createvm, startvm, modifyvm, clonevm, snapshot, controlvm [1, ch.8].

PART 2. WORK WITH VAGRANT

1. Download the required version of Vagrant according to the instructions [5] and according to the host operating system (OS) installed on the student's workplace. For Windows, the file may be called, for example, vagrant_2.2.0_x86_64.msi. Install Vagrant. Check the path to Vagrant bin in the Path variable (My computer -> Properties -> Advanced system settings -> Advanced -> Environment Variables).

2. Run the powershell. Create a folder "student name" (in English). In this example, create a folder vagrant_test. Next, go to the folder.

```
PS C:\Users\Andrii> cd C:\
PS C:\> mkdir vagrant_test


    Каталог: C:\


Mode                 LastWriteTime     Length Name
----                 -------------     ------ ----
d----         31.10.2018      1:02            vagrant_test


PS C:\> cd .\vagrant_test
```

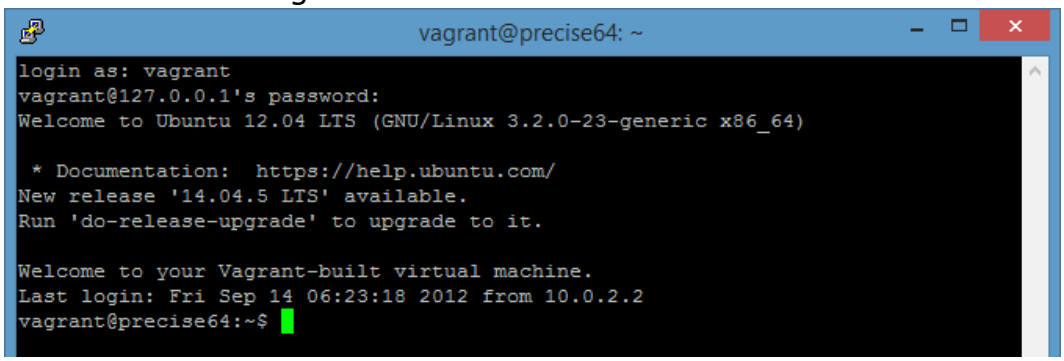
3. Initialize the environment with the default Vagrant box: init hashicorp/precise64

```
PS C:\vagrant_test> vagrant init hashicorp/precise64
A 'Vagrantfile' has been placed in this directory. You are now
ready to 'vagrant up' your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
'vagrantup.com' for more information on using Vagrant.
```

4. Run vagrant up and watch for messages during VM boot and startup.

```
PS C:\vagrant_test> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'hashicorp/precise64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'hashicorp/precise64' is up to date...
==> default: Setting the name of the VM: vagrant_test_default_1540940755138_18078
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
==> default: Forwarding ports...
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Booting VM...
==> default: Waiting for machine to boot. This may take a few minutes...
    default: SSH address: 127.0.0.1:2222
    default: SSH username: vagrant
    default: SSH auth method: private key
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Remote connection disconnect. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Remote connection disconnect. Retrying...
    default: Warning: Connection aborted. Retrying...
    default: Warning: Connection reset. Retrying...
    default: Warning: Connection aborted. Retrying...
    default:
    default: Vagrant insecure key detected. Vagrant will automatically replace
    default: this with a newly generated keypair for better security.
    default:
    default: Inserting generated public key within guest...
    default: Removing insecure key from the guest if it's present...
    default: Key inserted! Disconnecting and reconnecting using new SSH key...
==> default: Machine booted and ready!
==> default: Checking for guest additions in VM...
    default: The guest additions on this VM do not match the installed version of
    default: VirtualBox! In most cases this is fine, but in rare cases it can
    default: prevent things such as shared folders from working properly. If you see
    default: shared folder errors, please make sure the guest additions within the
    default: virtual machine match the version of VirtualBox you have installed on
    default: your host and reload your VM.
    default:
    default: Guest Additions Version: 4.2.0
    default: VirtualBox Version: 5.2
==> default: Mounting shared folders...
    default: /vagrant => C:/vagrant_test
```

5. Connect to the VM using the program PuTTY (can be downloaded from [6]), using SSH, IP address and port listed above (127.0.0.1:2222). By default, login - *vagrant* and password are also *vagrant*



6. Record the date and time by executing the date command

```
vagrant@precise64:~$ date
Tue Oct 30 23:49:50 UTC 2018
```

7. Stop and delete the created VM.

```
PS C:\vagrant_test> vagrant halt
==> default: Attempting graceful shutdown of VM...
PS C:\vagrant_test> vagrant destroy
    default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Destroying VM and associated drives...
```

8. Create your own Vagrant box [7]

9. *(optional)* Create a test environment from a few servers. Servers' parameters are chosen independently by the student.

REFERENCES

1. Oracle VM VirtualBox.User Manual https://www.virtualbox.org/manual/
2. Official page VirtualBox https://www.virtualbox.org/
3. Download page Ubuntu https://ubuntu.com/download
4. Documentation page Vagrant https://www.vagrantup.com/docs/index.html
5. Installation instructions page Vagrant https://www.vagrantup.com/docs/installation/index.html
6. Download page PuTTY https://www.putty.org/
7. O'Reilly - Vagrant: Up and Running.
8. Vagrant Workflows http://czerasz.com/2015/01/06/vagrant-workflows/
9. How To Use Vagrant To Create Small Virtual Test Lab on a Linux / OS X / MS-Windows https://www.cyberciti.biz/cloud-computing/use-vagrant-to-create-small-virtual-lab-on-linux-osx/