**Section 2: Setting up the development Environment**

**Development Machine Requirements & Setup options**

1. **Min machine configuration**
2. **Setup options**

**Windows:**

Create a ubuntu VM to carry out testing/exercises

**Ubuntu:**

* Install binaries directly on the machine
* OR create a VM to carry out testing

**For creating virtual Machine:**

You may use:

* Vagrant (it is used here)
* Create VM directly on Hypervisor

**Introduction to Hypervisor & setup of Virtual Box and Vagrant**

1. Basics of virtualization
2. Installation of oracle virtual box and vagrant

Virtualization is the software simulation of the physical hardware, You have the **physical machine** and you can have one or more **virtual machines** which simulate isolated physical machines.

To make this magic happen.

There is a layer of software that sits right on top of the physical machine, and this software is known as the **hypervisor.**

Virtualization is a set of technologies that allows a single physical computer or server to be divided into multiple virtual machines, operating systems, or containers.

Depending on the operating system, there are multiple hypervisor softwares available.

I'll be using the Oracle VirtualBox, which is a free software for creating **virtual machines** on Windows,Mac OS and even Linux.

**Quick note for the Windows 10 pro users.**

If your machine is already enabled to use Hyper-V, then please do not install VirtualBox as it will result in unexpected issues.

You may skip the installation of the VirtualBox if you already have it installed on your machine. Otherwise you need to download and install VirtualBox

Now let's talk about Vagrant.

**“Vagrant is a tool for building and managing virtual machine environments in a single workflow.”**

* Think of it as just a utility that makes it easy to work with virtual machines. (CLI tool for managing VM)
* You can use the vagrant file for automating the workflow and that is what we are going to use in our setup.

**Download Vagrant:**

* Vagrant is available on almost all major platforms and you can download it.
* Depending on your OS, you have to pick up the right version of Vagrant to be downloaded.
* Once the package has been downloaded, just click on it and follow the instructions.
* Click next and then accept the agreement.
* Click next again, use the default location and click next.
* Just hit install and this will install vagrant on your machine.

**Relationship between vagrant and the virtualization layer:**

Vagrant reads the vagrant file, which is a configuration file to create virtual machines.

And to do that it connects with the virtualization layer, which in our case is the VirtualBox and then creates the virtual machine as per the specification in the vagrant file.

**Commands to be used:**

1. vagrant up to start the virtual machine.
2. Vagrant ssh to log into the virtual machine.
3. Vagrant halt to gracefully shut down the virtual machine and
4. vagrant destroy to delete the virtual machine.

**Visual Studio Code & HLF2 Project Folders Overview**

1. **how to set up certain useful features in Visual Studio Code.**

* Once you have installed Visual Studio code**, set up the autosave** by clicking on the file menu and then checking the autosave. This way you won't have to explicitly save the files. When you will make changes, it'll make things easier for you.
* Another setup that you must do on Windows on Visual Studio code is setting up of the **end of line termination**.
* Click on the crlf or the bottom right of the Visual Studio code window.

This will open up a menu on the top which will show you Lf and Crlf select lf.

The reason we have to do this is because on windows the end of line termination is by way of carriage return line feed.

And if you run the script after making changes on Windows, Unix does not like it, so you will start to see errors in the format with messages saying that there is a slash character at the end.

So make sure that if you are using Windows you make this change to your Visual Studio code Environment.

Visual Studio code allows extensions to be added.

**I have installed:**

* Vagrant extension,
* Yaml extension,
* Golang extension and
* the Docker extension.

Open the project( added in project folder)

* open the project folder in the Visual Studio code.
* As you can see here, this is the setup folder with all the scripts that we have executed .
* To go through the code, simply click on the script and you can go through the script here to log into the virtual machine.
* You don't have to open up the PowerShell. You can simply click on view, click on Terminal, and right here you can say Vagrant Ssh, provide the password.( first create VM by command vagrant up)
* And now we are logged into the vagrant in a terminal that's opened up within the Visual Studio code.
* Here you will find there are multiple folders.
* These three folders were created as part of the execution of the fabric setup script.
* Other folders are used in different lectures.

As you will go through the lectures, you will learn about all of these folders in details.

**Course Project Repository Setup**

This cover steps that you need to carry out for **downloading the project** **repository** and you will also learn about the various development environment setup options.

**Download Project Repository:**

All of the sample code and shell scripts for setting up the environment are available in a project repository that may be downloaded from the link provided in the **folder** titled **“Download project Repository.”**

Once you have downloaded the archive for the Project repository, use the unzip tool for extracting the files in a local folder.

Right click on the project repository, zip file and select extract.

You may use the unzip command.

**Dev Environment Setup Options:**

There are two options to set up the development environment.

1. **VirtualBox Express install.**

* This option is available to those students who are using VirtualBox as their hypervisor.
* So if you are not using the VirtualBox as a hypervisor, then this option is not applicable to you.
* This option requires you to execute only one script, to set up the entire environment and it is goingto be 20 times faster to set up compared to the option number two.

1. **Standard Install**

* Option number two requires you to execute multiple installation scripts.

Let me help you make a decision on which option suits you best.

First thing you need to think is are you going to install the development environment directly on Ubuntu?

If that is the case, then you need to go with the standard installation option.

If you are going to install the development environment on virtual machine.

And if your hypervisor is VirtualBox, then you have options.

If you're using a different hypervisor, not the VirtualBox, then you don't have much of an option.

You would go with the standard install.

The next thing you need to think about is would you like to go through the installation process and

go through the scripts used for installing the components?

If the answer is yes, then you need to go with the standard install.

Otherwise you go with the VirtualBox express install.

Now here is the thing.

My **recommendation** is to go with the VirtualBox express install first because it will give you a quick way to start learning Hyperledger fabric and you can always go back to standard install to learn the installation process.

At this point I'm hoping that you have made a decision as to which option you will use.

Now, one important note if you have decided to go with the standard installation directly on Ubuntu,

* then please be aware that there may be challenges due to the environmental differences.
* The upgrades of the project repository will require you to manually clean up the earlier versions and
* upgrade of the fabric binaries and tools may lead to version conflicts that you will have to resolve on your own.

So my recommendation go with the virtual machine and once you are comfortable, then you can start to learn native installation.

At this time, we are ready to roll forward with the installation process for VirtualBox Express install.

**Virtual Box Express Install**

**Part1: Setup the virtual machine**

**Part2: Initialize the dev environment setup**

**Part3: Validate the setup**

**Part1: Setup the virtual machine**

we need to carry out two steps.

1. validate the setup in the vagrant file.

Update the **Vagrantfile**

Config.vm.box=”acloudfan/hlfdev2=.0-0”

* You need to set up this particular property to point to a virtual box with the Pre-configured Hyperledger fabric environment.
* And then you have to execute Vagrant up to create the virtual machine.

**Command to run terminal:**

1. vagrant init acloudfan/hlfdev2.0-0
2. config.vm.box = "acloudfan/hlfdev2.0-0"
3. vagrant up

**Part2: Initialize the dev environment setup**

**Command to run terminal:**

1. vagrant ssh
2. cd setup
3. sudo ./init-vexpress.sh

**Part3: Validate the setup**

1. log out of the VM-**exit**
2. log in **-vagrant ssh**
3. After that you change the folder to the setup folder in the virtual machine terminal and execute validate **./validate-prereqs.sh**

* Now this will validate all of the components installed on the virtual machine.
* So go through the messages printed out by the script to make sure there are no errors.
* If you see any error, we will need to address it.
* If you don't see any error, you are good at this point and you can move to the next section.

**Command to run terminal:**

1. Exit
2. Vagrant ssh
3. ./validate-prereqs.sh

**Summary**

**Part1: Setup the virtual machine**

1. vagrant init acloudfan/hlfdev2.0-0
2. config.vm.box = "acloudfan/hlfdev2.0-0"
3. vagrant up

**Part2: Initialize the dev environment setup**

1. vagrant ssh
2. cd setup
3. sudo ./init-vexpress.sh

**Part3: Validate the setup**

1. Exit
2. Vagrant ssh
3. ./validate-prereqs.sh

**HLF2 - Course Project Setup**

**Part1: Set up the Virtual Machine**

**Part2: install the software/HLF and hyperledger Binaries**

Note:If you are not using native installation and also vagrant then you can skip part 1

**Part1: Set up the Virtual Machine**

**Command to run terminal:**

1. config.vm.box = "bento/ubuntu-18.04"-> in vagrant file
2. Open terminal → Windows Powershell
3. Vagrant up
4. Vagrant ssh
5. Cd /vagrant
6. Ls -la
7. Cd setup
8. Ls -la

**Part2: install the software/HLF and hyperledger Binaries**

**We will install:**

1. Docker
2. GoLang
3. Fabric binaries and samples
4. CA Server
5. /jq utility

Chmod 755\*.sh will give permission for executing scripts

**1.Install Docker:**

**Command to run terminal:**

1. sudo ./docker.sh
2. exit
3. vagrant ssh
4. docker version

**2. install GOLang:**

**Command to run terminal:**

1. cd setup
2. sudo ./go.sh
3. exit
4. vagrant ssh
5. go version

**3.Install Fabric binaries and samples**

**Command to run terminal:**

1. cd setup
2. sudo -E ./fabric-setup.sh
3. orderer version(to validate)
4. cd vagrant
5. ls -la
6. cdvagrant/setup
7. ls -la

**4.install CA server**

1. sudo -E ./caserver-setup.sh
2. fabric-ca-server version
3. fabric-ca-client version
4. cd vagrant/setup

**5.Install jq**

1. sudo ./jq.sh
2. jq