Packaging training

2. Setup

What's needed

To create deb/rpm packages, we need a specific linux distribution (centos, ubuntu, ….)
To create python, gem, npm, it is often easy to do it directly on our computer, though it can be nice to test the package on different distributions and on clean systems

VirtualBox/Vagrant

Vagrant will work above VirtualBox and create virtual machines from a “build” file, and you will be able to easily connect via ssh to the vm. Vagrant is available for Windows/Mac/Linux

So, the first thing to do is to install VirtualBox then Vagrant on your computer

https://www.vagrantup.com/ , https://www.virtualbox.org/

Create and connect to the VM

Once VirtualBox and Vagrant are installed.

$mkdir pkg-training

$ cd pkg-training

$git clone https://github.com/osallou/packaging-training.git

$cd ubuntu <- contains a *Vagrantfile* file

$vagrant up <- the first *up* command will download base image and provision the vm according to the Vagrantfile setup, next *up* commands will only start/update the vm

$vagrant halt <- now the vm is initialized, we can stop it

$cd centos

$vagrant up

$vagrant halt

vm: virtual machine
Vagrantfile: used by Vagrant to create a vm. It defined a "base" image and how to provision it

Playing with your new system

I won’t explain here how to use Vagrant, there is a nice documentation on Vagrant site for this, I only explain how to use it for the purpose of this training

Connect to a Vagrant VM

```
$ cd ubuntu
#### Start the VM
$ vagrant up
#### Starts an interactive ssh session in the vm
$ vagrant ssh   vagrant@vagrant-ubuntu-trusty-64:~$  <- you are at your Ubuntu prompt
vagrant@vagrant-ubuntu-trusty-64:~$ ls /vagrant/
#### The /vagrant directory is an automatic shared mount with the directory where the Vagrantfile is. This means that
#### if you can put your code, data, ... in this directory, it will be available in the vm, and vice-versa.
quiltrc.example  Vagrantfile
vagrant@vagrant-ubuntu-trusty-64:~$
#### Exit from the VM like for a basic ssh session
vagrant@vagrant-ubuntu-trusty-64:~$ exit
#### Stop the vm
$ vagrant halt
```

We’re ready

Ok, now we have a CentOS and an Ubuntu server, ready for deb/rpm packaging.
As you have seen, it is easy to setup multiple other environments (Fedora, Suse, …) to create/test your packages.