

TP1 R3D17 - Vagrant

@Bastien-Helec

1 Vagrant :

1.1 Creation d'une première VM Ubuntu :

1. Créer un répertoire Vagrant à la racine de votre compte.
2. Placez-vous dans ce répertoire et créez un répertoire VM1. Version rapide qui crée les 2 répertoires simultanément : `mkdir -p ~/Vagrant/VM1`.
3. Dans VM1, tapez la commande : `vagrant init ubuntu/xenial64` qui crée un fichier de configuration Vagrantfile dans le répertoire, que nous allons inspecter plus tard.

```
Vagrant
└── VM1
    ├── Vagrantfile
    └── ...

vagrant init ubuntu/xenial64
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.
bastien_fedora@fedora:~/Vagrant/VM1$
```

4. Démarrez la VM avec `vagrant up`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant init ubuntu/xenial64 --box-version 20211001.0.0
`Vagrantfile` already exists in this directory. Remove it before
running `vagrant init`.
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant up
Bringing machine 'default' up with 'libvirt' provider...
==> default: Box 'ubuntu/xenial64' could not be found. Attempting to find and install...
default: Box Provider: libvirt
default: Box Version: 20211001.0.0
==> default: Loading metadata for box 'ubuntu/xenial64'
default: URL: https://vagrantcloud.com/ubuntu/xenial64
The box you're attempting to add doesn't support the provider
you requested. Please find an alternate box or use an alternate
provider. Double-check your requested provider to verify you didn't
simply misspell it.

If you're adding a box from HashiCorp's Vagrant Cloud, make sure the box is
released.

Name: ubuntu/xenial64
Address: https://vagrantcloud.com/ubuntu/xenial64
Requested provider: [:libvirt]
bastien_fedora@fedora:~/Vagrant/VM1$
```

Celle-ci ne fonctionne pas en demandant via libvirt c'est pour cela que j'ai essayer de faire `vagrant up --`

provider=virtualbox

```

4 déc. 16:44
Tili: bastien_fedora@fedora:~/Vagrant/VM1

2: bastien_fedora@fedora:~/Vagrant/VM1
Requested provider: [virtualbox]
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant up --provider=virtualbox
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Box 'ubuntu/xenial64' could not be found. Attempting to find and install...
    default: Box Provider: virtualbox
    default: Box Version: 20211001.0.0
==> default: Loading metadata for box 'ubuntu/xenial64'
    default: URL: https://vagrantcloud.com/ubuntu/xenial64
==> default: Adding box 'ubuntu/xenial64' (v20211001.0.0) for provider: virtualbox
    default: Downloading: https://vagrantcloud.com/ubuntu/boxes/xenial64/versions/20211001.0.0/providers/virtualbox/unknown/vagrant-box
Download redirected to host: cloud-images.ubuntu.com
==> default: Successfully added box 'ubuntu/xenial64' (v20211001.0.0) for 'virtualbox'!
==> default: Importing base box 'ubuntu/xenial64'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'ubuntu/xenial64' version '20211001.0.0' is up to date...
==> default: Setting the name of the VM: VM1_default_1701704648688_12074
Vagrant is currently configured to create VirtualBox synced folders with
the 'SharedFoldersEnableSymlinksCreate' option enabled. If the Vagrant
guest is not trusted, you may want to disable this option. For more
information on this option, please refer to the VirtualBox manual:
https://www.virtualbox.org/manual/ch04.html#sharedfolders

This option can be disabled globally with an environment variable:
    VAGRANT_DISABLE_VBOXSYMLINKCREATE=1
or on a per folder basis within the Vagrantfile:
    config.vm.synced_folder '/host/path', '/guest/path', SharedFoldersEnableSymlinksCreate: false
==> 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: Running 'pre-boot' VM customizations...
==> 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:
    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
  
```

5. Logez-vous sur la VM en tapant **vagrant ssh**. Quelles sont les interfaces réseaux de la machine ?

```

bastien_fedora@fedora:~/Vagrant/VM1$ vagrant ssh
Welcome to Ubuntu 16.04.7 LTS (GNU/Linux 4.4.0-210-generic x86_64)

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

UA Infra: Extended Security Maintenance (ESM) is not enabled.

0 updates can be applied immediately.

45 additional security updates can be applied with UA Infra: ESM
Learn more about enabling UA Infra: ESM service for Ubuntu 16.04 at
https://ubuntu.com/16-04

New release '18.04.6 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

vagrant@ubuntu-xenial:~$
  
```

6. Placez-vous dans le répertoire (de la machine physique) où VirtualBox stocke les fichiers de la VM.

 vagrant1.6

Quel est le type de disque (en se basant sur l'extension du fichier) ? Cherchez dans <https://www.virtualbox.org/manual/ch05.html> pour comprendre les différentes extensions et notamment celles de VM1 et de la machine Ubuntu créée en section ?

Le fichier est en VMDK , ce qui signifie que c'est un disque dur virtuel utiliser dans plusieurs outils de virtualisation comme VMware ou VirtualBox.

2 Vagrant Customisation VM et construction d'image :

2.1 Customisation de l'instance - Partie 1 :

Votre fichier Vagrantfile contient surtout des commentaires comme peut vous en convaincre un simple :

```
egrep -v -e"\"#|^$" Vagrantfile
```

Cette commande permet d'inverser la recherche de ne pas selectionner tout ce qui commence par "#" et "^\$" qui sont des commentaires.

1. Editer le fichier Vagrantfile et d'é-commenter les lignes en rapport avec vm.provision :

```
apt-get update
apt-get install -y apache2
SHELL
```

```
Vagrant.configure("2") do |config|
  config.vm.box = "ubuntu/xenial64"
  config.vm.box_version = "20211001.0.0"
  config.vm.provision "shell", inline: <<-SHELL
  apt-get update
  apt-get install -y apache2
  SHELL
end
bastien_fedora@fedora:~/Vagrant/VM1$
```

Le "provisioning" aura lieu soit lors du prochain démarrage, soit à chaud.

Nous allons prendre la seconde option en tapant la commande : `vagrant provision`

```

bastien_fedora@fedora:~/Vagrant/VM1$ vagrant provision
==> default: Running provisioner: shell...
default: Running: inline script
default: Hit:1 http://archive.ubuntu.com/ubuntu xenial InRelease
default: Hit:2 http://security.ubuntu.com/ubuntu xenial-security InRelease
default: Hit:3 https://esm.ubuntu.com/infra/ubuntu xenial-infra-security InRelease
default: Hit:4 http://archive.ubuntu.com/ubuntu xenial-updates InRelease
default: Hit:5 https://esm.ubuntu.com/infra/ubuntu xenial-infra-updates InRelease
default: Hit:6 http://archive.ubuntu.com/ubuntu xenial-backports InRelease
default: Reading package lists...
default: Reading package lists...
default: Building dependency tree...
default: Reading state information...
default: The following additional packages will be installed:
default:  apache2-bin apache2-data apache2-utils libapr1 libaprutil1
default:  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
default: Suggested packages:
default:  www-browser apache2-doc apache2-suexec-pristine | apache2-suexec-custom
default:  openssl-blacklist
default: The following NEW packages will be installed:
default:  apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1
default:  libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.1-0 ssl-cert
default: 0 upgraded, 10 newly installed, 0 to remove and 5 not upgraded.
default: Need to get 1,559 kB of archives.
default: After this operation, 6,448 kB of additional disk space will be used.
default: Get:1 http://archive.ubuntu.com/ubuntu xenial/main amd64 libapr1 amd64 1.5.2-3 [86.0 kB]
default: Get:2 http://archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1 amd64 1.5.4-1build1 [77.1 kB]
default: Get:3 http://archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1-dbd-sqlite3 amd64 1.5.4-1build1 [10.6 kB]
default: Get:4 http://archive.ubuntu.com/ubuntu xenial/main amd64 libaprutil1-ldap amd64 1.5.4-1build1 [8,720 B]
default: Get:5 http://archive.ubuntu.com/ubuntu xenial/main amd64 liblua5.1-0 amd64 5.1.5-8ubuntu1 [102 kB]
default: Get:6 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-bin amd64 2.4.18-2ubuntu3.17 [927 kB]
default: Get:7 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-utils amd64 2.4.18-2ubuntu3.17 [81.9 kB]
default: Get:8 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2-data all 2.4.18-2ubuntu3.17 [162 kB]
default: Get:9 http://archive.ubuntu.com/ubuntu xenial-updates/main amd64 apache2 amd64 2.4.18-2ubuntu3.17 [86.8 kB]
default: Get:10 http://archive.ubuntu.com/ubuntu xenial/main amd64 ssl-cert all 1.0.37 [16.9 kB]
default: dpkg-preconfigure: unable to re-open stdin: No such file or directory
default: Fetched 1,559 kB in 0s (2,154 kB/s)
default: Selecting previously unselected package libapr1:amd64.
(Reading database ... 54424 files and directories currently installed.)
default: Preparing to unpack .../libapr1_1.5.2-3_amd64.deb ...
default: Unpacking libapr1:amd64 (1.5.2-3) ...
default: Selecting previously unselected package libaprutil1:amd64.
default: Preparing to unpack .../libaprutil1_1.5.4-1build1_amd64.deb ...
default: Unpacking libaprutil1:amd64 (1.5.4-1build1) ...
default: Selecting previously unselected package libaprutil1-dbd-sqlite3:amd64.
default: Preparing to unpack .../libaprutil1-dbd-sqlite3_1.5.4-1build1_amd64.deb ...
default: Unpacking libaprutil1-dbd-sqlite3:amd64 (1.5.4-1build1) ...
default: Selecting previously unselected package libaprutil1-ldap:amd64.
default: Preparing to unpack .../libaprutil1-ldap_1.5.4-1build1_amd64.deb ...
default: Unpacking libaprutil1-ldap:amd64 (1.5.4-1build1) ...
default: Selecting previously unselected package liblua5.1-0:amd64.
default: Preparing to unpack .../liblua5.1-0_5.1.5-8ubuntu1_amd64.deb ...
default: Unpacking liblua5.1-0:amd64 (5.1.5-8ubuntu1) ...
default: Selecting previously unselected package apache2:amd64.

```

2. Prouvez que vous avez bien un serveur Web qui tourne dans la VM en montrant : (a) que le service apache2 est actif (quelle commande avez-vous tapé ?). Dans la VM : `systemctl status apache2`

```

vagrant@ubuntu-xenial:~$ systemctl status apache2
● apache2.service - LSB: Apache2 web server
   Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Mon 2023-12-04 16:12:50 UTC; 2min 32s ago
     Docs: man:systemd-sysv-generator(8)
   CGroup: /system.slice/apache2.service
           └─13175 /usr/sbin/apache2 -k start
             └─13178 /usr/sbin/apache2 -k start
               └─13179 /usr/sbin/apache2 -k start
vagrant@ubuntu-xenial:~$

```

- (b) que le programme apache2 écoute bien sur le port 80 (utilisez netstat du paquet net-tools)

```
netstat -tulpn | grep apache2
```

```
vagrant@ubuntu-xenial:~$ sudo !!
sudo netstat -tulnp | grep apache2
tcp6      0      0 :::*          LISTEN      13175/apache2
vagrant@ubuntu-xenial:~$
```

(c) en téléchargeant la page d'accueil (comment ? Quelle commande avez-vous tapé ?).

`curl localhost:80`

```
vagrant@ubuntu-xenial:~$ curl localhost:80
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional
">
<html xmlns="http://www.w3.org/1999/xhtml">
  <!--
    Modified from the Debian original for Ubuntu
    Last updated: 2014-03-19
    See: https://launchpad.net/bugs/1288690
  -->
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Apache2 Ubuntu Default Page: It works</title>
    <style type="text/css" media="screen">
      * {
        margin: 0px 0px 0px 0px;
        padding: 0px 0px 0px 0px;
      }

      body, html {
        padding: 3px 3px 3px 3px;
        background-color: #D9D9E3;
      }
    </style>
  </head>
  <div style="text-align: center; border: 1px solid black; padding: 10px 10px 10px 10px; width: 80%; margin: auto;">
    <div style="display: inline-block; width: 90%; text-align: left; vertical-align: top;">
      <h1 style="margin: 0; font-size: 2em; font-weight: normal;">It works!
      <div style="margin-top: 10px; font-family: monospace; font-size: 0.8em; text-align: left;">
        <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 80%; margin-bottom: 5px;">
          <pre>
            <!--
              Modified from the Debian original for Ubuntu
              Last updated: 2014-03-19
              See: https://launchpad.net/bugs/1288690
            -->
          </pre>
        </div>
        <div style="border: 1px solid black; padding: 5px; display: inline-block; width: 80%; margin-bottom: 5px;">
          <pre>
            <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
            <title>Apache2 Ubuntu Default Page: It works</title>
            <style type="text/css" media="screen">
              * {
                margin: 0px 0px 0px 0px;
                padding: 0px 0px 0px 0px;
              }

              body, html {
                padding: 3px 3px 3px 3px;
                background-color: #D9D9E3;
              }
            </style>
          </pre>
        </div>
      </div>
    </div>
  </div>
</html>
```

3. Visualiser la page d'accueil depuis le navigateur de la machine physique est impossible. Pourquoi ? (vous pouvez essayer pour vous en convaincre)

Car l'adresse ip de la VM en 127.0.0.1 est une adresse locale qui ne peut être accéder que depuis la VM elle même. Il serait possible de pouvoir communiquer avec la VM depuis une adresse ip dans le reseau

4. La configuration réseau est bien compliquée. Analysez le fichier Vagrantfile et montrez qu'il y a 3 solutions (vous devez être capable de dire où elles se situent dans le fichier) pour permettre un accès simple au serveur Apache dans la VM.

1ere solution : faire un mapping port de la VM vers le port de la machine physique qui autorise la connexion. l:29 `config.vm.network "forwarded_port", guest: 80, host: 8080`

2eme solution : reseau priver entre les deux machines avec une adresse ip fixe l:36

`config.vm.network "private_network", ip:192.168.24.12` pour cette situation je dois creer une nouvelle carte réseau dans la machine physique pour pouvoir communiquer avec la VM :

3eme solution : reseau public avec une adresse VM accessible depuis le reseau l:41


`config.vm.network "public_network"`

`vagrant reload`

```
vagrant@ubuntu-xenial:~$ ip -br a
lo          UNKNOWN      127.0.0.1/8  ::1/128
enp0s3      UP            10.0.2.15/24 fe80::be:82ff:fe6b:cc1d/64
enp0s8      UP            192.168.24.12/24 fe80::a00:27ff:fe10:7dc4/64
enp0s9      UP            192.168.1.34/24 2a01:cb1d:8aac:8500:a00:27ff:fee9:50ee/64 fe80::a00:27ff:fee9:50ee/64
vagrant@ubuntu-xenial:~$
```

localhost:8080

Dev Recette Github Heimdall Cloud Divertissement Life RP Logement Actu Edit



ubuntu

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```


- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers **`a2enmod`**, **`a2dismod`**, **`a2ensite`**, **`a2disssite`**, and **`a2enconf`**, **`a2disconf`**. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **`public_html`** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in

Non sécurisé | 192.168.24.12

Dev Recette Github Heimdall Cloud Divertissement Life RP Logement Actu Edit



Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in [/usr/share/doc/apache2/README.Debian.gz](#)**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers **`a2enmod`**, **`a2dismod`**, **`a2ensite`**, **`a2disssite`**, and **`a2enconf`**, **`a2disconf`**. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **public_html** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in



1. Quels sont (en quelques mots) les avantages et inconvénients de ces 3 solutions d'apres vous ?

1ere solution :	Avantages	Inconvénients
Facile à mettre en place, communications interne	Pas de communication entre les deux machines comme si c'etait 2 pc differents , ouvre un port sur la machine physique	
2eme solution :	Avantages	Inconvénients
Communication securisé entre les deux machines (aucun moyen de communication avec l'exterieur) pas de ouverture de port indesirable	mise en place complexe car demande a toucher a la configuration de virtualbox	
3eme solution :	Avantages	Inconvénients
Communication entre pc et VM (depuis un réseau physique), ouverture de port uniquement sur la VM	mise en place legerement complexe car passe par un bridge sur l'ordinateur physique	

2.2 Customisation de l'instance - Partie 2 :

1. Lors du reboot de la machine, Vagrant vous a demandé sur quelle interface physique de la machine doit être fait le pont (eth1 en l'occurrence). Pour ne pas avoir à faire cela à chaque reboot, on peut modifier le fichier Vagrantfile en complétant la ligne sur l'interface réseau par le nom exact de l'interface. Cela donne (à adapter si besoin): `config.vm.network "public_network", bridge: "wlp0s20f3"`

1: bastien_fedora@fedora:~/Vagrant/VM1 ▾

GNU nano 7.2

Vagrantfile

```
# using a specific IP.
config.vm.network "private_network", ip: "192.168.24.12"

# Create a public network, which generally matched to bridged network.
# Bridged networks make the machine appear as another physical device on
# your network.
config.vm.network "public_network", bridge: "wlp0s20f3"

# Share an additional folder to the guest VM. The first argument is
# the path on the host to the actual folder. The second argument is
# the path on the guest to mount the folder. And the optional third
# argument is a set of non-required options.
# config.vm.synced_folder "../data", "/vagrant_data"

# Provider-specific configuration so you can fine-tune various
```

2. Vagrant utilise les possibilités de répertoire partagé fournies par VirtualBox. Nous allons l'illustrer ici pour permettre de modifier les fichiers du serveur web de VM1.

(a) Dans le répertoire où se trouve le fichier Vagrantfile, créez un répertoire vagrantsite et placez dedans le fichier index.html suivant :

```
<html>
<head>
<title>TP1 R3D17 - Vagrant</title>
</head>
<body>
<div align="center"><h1>TP1 R3D17 - Vagrant</h1></div>
</body>
</html>
```

```
1: bastien_fedora@fedora:~/Vagrant/VM1 ▼
bastien_fedora@fedora:~/Vagrant/VM1$ nano Vagrantfile
bastien_fedora@fedora:~/Vagrant/VM1$ mkdir vagrantsite
bastien_fedora@fedora:~/Vagrant/VM1$ nano vagrantsite/index.html
bastien_fedora@fedora:~/Vagrant/VM1$ cat vagrantsite/index.html
<html>
<head>
<title>TP1 R3D17 - Vagrant</title>
</head>
<body>
<div align="center"><h1>TP1 R3D17 - Vagrant</h1></div>
</body>
</html>
bastien_fedora@fedora:~/Vagrant/VM1$
```

(b) Puis éditez le fichier Vagrantfile, et modifiez la partie montage de fichier et la partie liée au montage de fichier ainsi que celle concernant le script de configuration.

```
...
config.vm.synced_folder "vagrantsite", "/opt/vagrantsite"
...
config.vm.provision "shell", inline: <<-SHELL
apt-get update
apt-get install -y apache2
ln -s /opt/vagrantsite /var/www/html/vagrantsite
SHELL
```

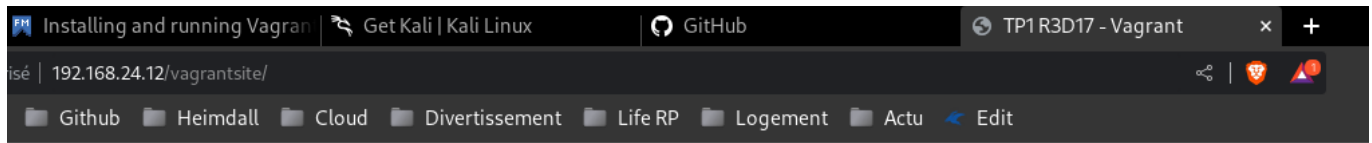
3. Redémarrez votre instance avec la commande `vagrant reload --provision`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant reload --provision
==> default: Attempting graceful shutdown of VM...
==> default: Checking if box 'ubuntu/xenial64' version '20211001.0.0' is up to date...
==> default: Clearing any previously set forwarded ports...
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
    default: Adapter 1: nat
    default: Adapter 2: hostonly
    default: Adapter 3: bridged
==> default: Forwarding ports...
    default: 80 (guest) => 8080 (host) (adapter 1)
    default: 22 (guest) => 2222 (host) (adapter 1)
==> default: Running 'pre-boot' VM customizations...
==> 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
```

Attention : il faut faire reload pour redémarrer une VM et `--provision` pour que la partie provisioning se fasse. Pourquoi est-ce que la partie provisioning est-elle optionnelle et non exécutée à chaque boot ?

Car le provisioning est une action qui peut prendre du temps et qui n'est pas forcément utile a chaque reboot de la VM.

4. Montrer que les modifications escomptées ont bien été faites en affichant la page d'accueil du site vagrantsite (<http://yyy/html/vagrantsite/index>)



TP1 R3D17 - Vagrant

2.3 Préparation d'une image :

Supposons que l'image actuelle nous satisfasse en terme de configuration. Nous allons en faire un template.

1. Faire un template va nous permettre de créer d'autres VMs suivant la même configuration. Cela pose un problème avec l'utilisation de la redirection de port que nous avons utilisé. Pourquoi ?

Car la redirection de port est une configuration qui est propre a la VM et qui ne peut pas être dupliquer sur une autre VM.

2. Il faut modifier la directive ainsi : `config.vm.network "forwarded_port", guest: 80, host: 8080, auto_correct: true`

```

1: bastien_fedora@fedora:~/Vagrant/VM1
GNU nano 7.2 Vagrantfile
# Disable automatic box update checking. If you disable this, then
# boxes will only be checked for updates when the user runs
# `vagrant box outdated`. This is not recommended.
# config.vm.box_check_update = false

# Create a forwarded port mapping which allows access to a specific port
# within the machine from a port on the host machine. In the example below,
# accessing "localhost:8080" will access port 80 on the guest machine.
# NOTE: This will enable public access to the opened port
config.vm.network "forwarded_port", guest: 80, host: 8080, auto_correct: true

# Create a forwarded port mapping which allows access to a specific port
# within the machine from a port on the host machine and only allow access
# via 127.0.0.1 to disable public access
# config.vm.network "forwarded_port", guest: 80, host: 8080, host_ip: "127.0.0.1"

# Create a private network, which allows host-only access to the machine
# using a specific IP.
config.vm.network "private_network", ip: "192.168.24.12"

# Create a public network, which generally matched to bridged network.

```

3. Créer la box avec la commande `vagrant package`

Si il y a un problème, il se peut que ce soit à cause du manque des Guest additions dans VM1, un complément indispensable pour Vagrant, voir <https://www.vagrantup.com/docs/virtualbox/boxes.html>.

```

1: bastien_fedora@fedora:~/Vagrant/VM1
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant package
==> default: Attempting graceful shutdown of VM...
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: /home/bastien_fedora/Vagrant/VM1/package.box
bastien_fedora@fedora:~/Vagrant/VM1$

```

4. Une fois la box faite , il faut l'ajouter a votre liste avec la commande : `vagrant box add --name vagrantTP1 ./package.box`

```
1: bastien_fedora@fedora:~/Vagrant/VM1 ▾
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant package
==> default: Attempting graceful shutdown of VM...
==> default: Clearing any previously set forwarded ports...
==> default: Exporting VM...
==> default: Compressing package to: /home/bastien_fedora/Vagrant/VM1/package.box
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant box add --name vagrantTP1 ./package.box
==> box: Box file was not detected as metadata. Adding it directly...
==> box: Adding box 'vagrantTP1' (v0) for provider:
    box: Unpacking necessary files from: file:///home/bastien_fedora/Vagrant/VM1/package.box
==> box: Successfully added box 'vagrantTP1' (v0) for 'virtualbox!'
bastien_fedora@fedora:~/Vagrant/VM1$
```

5. Listez vos boxes avec :

`vagrant box list`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant box list
fedora/32-cloud-base (libvirt, 32.20200422.0)
ubuntu/xenial64      (virtualbox, 20211001.0.0)
vagrantTP1           (virtualbox, 0)
bastien_fedora@fedora:~/Vagrant/VM1$
```

6. Vagrant stocke les images (boxes) dans un répertoire particulier, `~/.vagrant.d/boxes`. Faites un `ls -R ~/.vagrant.d/boxes` pour lister ce répertoire. Quelles sont les images ?

```

bastien_fedora@fedora:~/Vagrant/VM1$ ls -R ~/.vagrant.d/boxes/
/home/bastien_fedora/.vagrant.d/boxes/:
fedora-VAGRANTSLASH-32-cloud-base  ubuntu-VAGRANTSLASH-xenial64  vagrantTP1

/home/bastien_fedora/.vagrant.d/boxes/fedora-VAGRANTSLASH-32-cloud-base:
32.20200422.0  metadata_url

/home/bastien_fedora/.vagrant.d/boxes/fedora-VAGRANTSLASH-32-cloud-base/32.20200422.0:
libvirt

/home/bastien_fedora/.vagrant.d/boxes/fedora-VAGRANTSLASH-32-cloud-base/32.20200422.0/libvirt:
box.img  box_update_check  metadata.json  Vagrantfile

/home/bastien_fedora/.vagrant.d/boxes/ubuntu-VAGRANTSLASH-xenial64:
20211001.0.0  metadata_url

/home/bastien_fedora/.vagrant.d/boxes/ubuntu-VAGRANTSLASH-xenial64/20211001.0.0:
virtualbox

/home/bastien_fedora/.vagrant.d/boxes/ubuntu-VAGRANTSLASH-xenial64/20211001.0.0/virtualbox:
box.ovf          metadata.json          ubuntu-xenial-16.04-cloudimg.mf  Vagrantfile
box_update_check  ubuntu-xenial-16.04-cloudimg-configdrive.vmdk  ubuntu-xenial-16.04-cloudimg.vmdk

/home/bastien_fedora/.vagrant.d/boxes/vagrantTP1:
0

/home/bastien_fedora/.vagrant.d/boxes/vagrantTP1/0:
virtualbox

/home/bastien_fedora/.vagrant.d/boxes/vagrantTP1/0/virtualbox:
box-disk001.vmdk  box-disk002.vmdk  box.ovf  metadata.json  Vagrantfile  vagrant_private_key
bastien_fedora@fedora:~/Vagrant/VM1$

```

7. Est-ce que l'on peut effacer ~/Vagrant/VM1/package.box (avec une commande rm) ? Dit autrement, est-ce que cela a une incidence sur l'image dans vagrant ?

Non car l'image est stocker dans le repertoire ~/.vagrant.d/boxes

3 Vagrant advenced version :

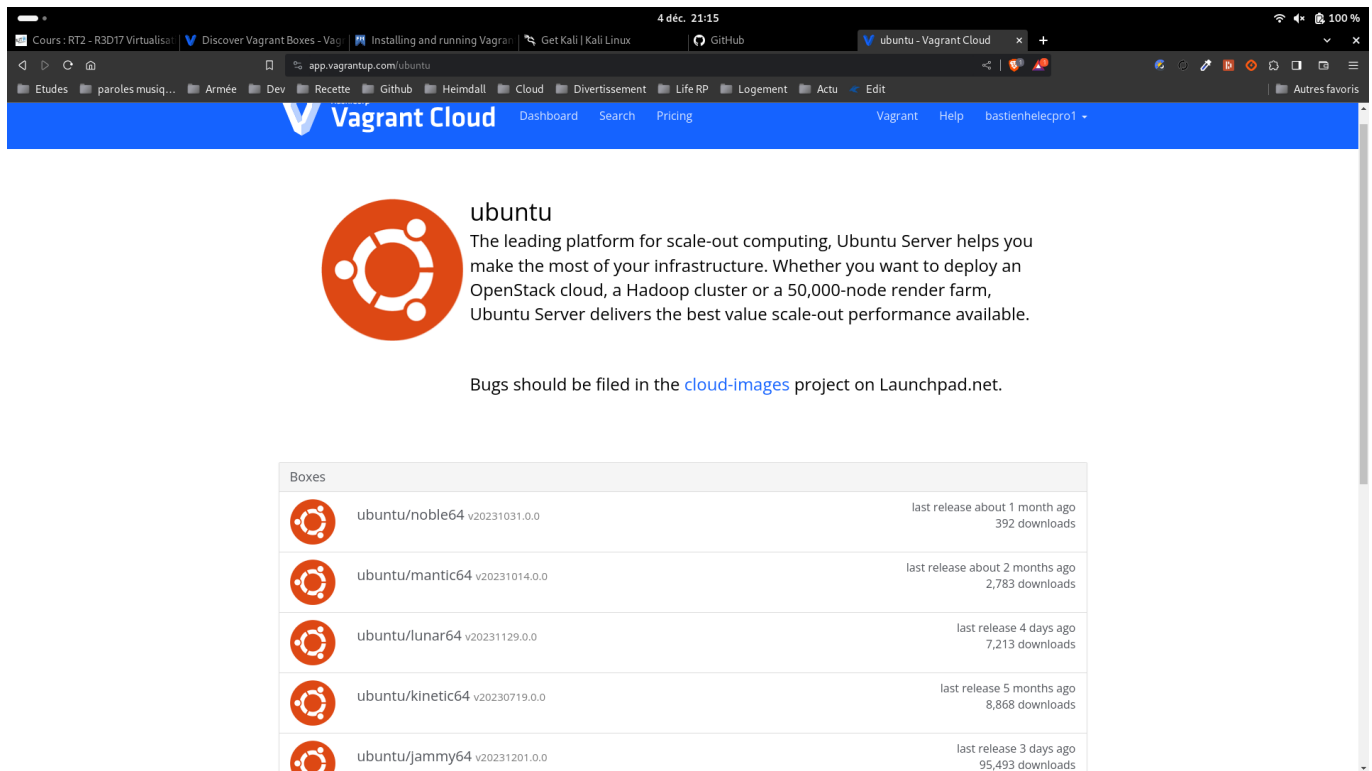
3.1 Vagrant hub :

Hashicorp fournit un service de partage des images : <https://app.vagrantup.com/boxes/search> que l'on atteint aussi en partant du site principal de vagrant en cliquant sur Find Boxes <https://www.vagrantup.com/>.

1. Les images sont associées à un compte utilisateur et ont un nom. Par exemple ubuntu/xenial64.

Ubuntu est le nom du compte et xenial64 est le nom de l'image.






(a) Choisissez l'image ubuntu/xenial64 et cliquez dessus. Cliquez ensuite sur le nom du compte ubuntu pour avoir le profil de l'utilisateur. Est-ce une image en laquelle on peut avoir confiance ?



ubuntu

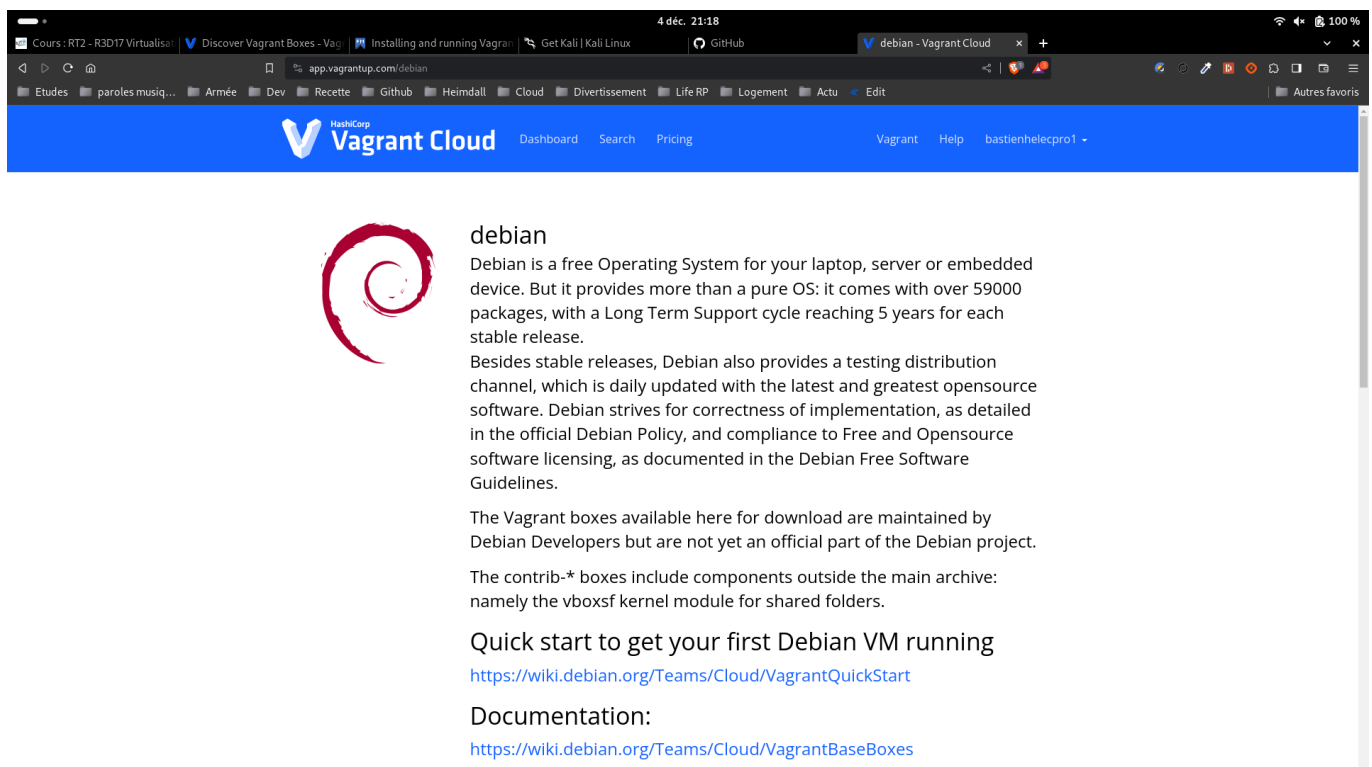
The leading platform for scale-out computing. Ubuntu Server helps you make the most of your infrastructure. Whether you want to deploy an OpenStack cloud, a Hadoop cluster or a 50,000-node render farm, Ubuntu Server delivers the best value scale-out performance available.

Bugs should be filed in the [cloud-images](#) project on Launchpad.net.

Boxes		
	ubuntu/noble64 v20231031.0.0	last release about 1 month ago 392 downloads
	ubuntu/mantic64 v20231014.0.0	last release about 2 months ago 2,783 downloads
	ubuntu/lunar64 v20231129.0.0	last release 4 days ago 7,213 downloads
	ubuntu/kinetic64 v20230719.0.0	last release 5 months ago 8,868 downloads
	ubuntu/jammy64 v20231201.0.0	last release 3 days ago 95,493 downloads

Oui car l'image est officiel et est maintenu par l'equipe de ubuntu. pour ubuntu server.

(b) Même question pour debian/jessie64



debian

Debian is a free Operating System for your laptop, server or embedded device. But it provides more than a pure OS: it comes with over 59000 packages, with a Long Term Support cycle reaching 5 years for each stable release.

Besides stable releases, Debian also provides a testing distribution channel, which is daily updated with the latest and greatest opensource software. Debian strives for correctness of implementation, as detailed in the official Debian Policy, and compliance to Free and Opensource software licensing, as documented in the Debian Free Software Guidelines.

The Vagrant boxes available here for download are maintained by Debian Developers but are not yet an official part of the Debian project.

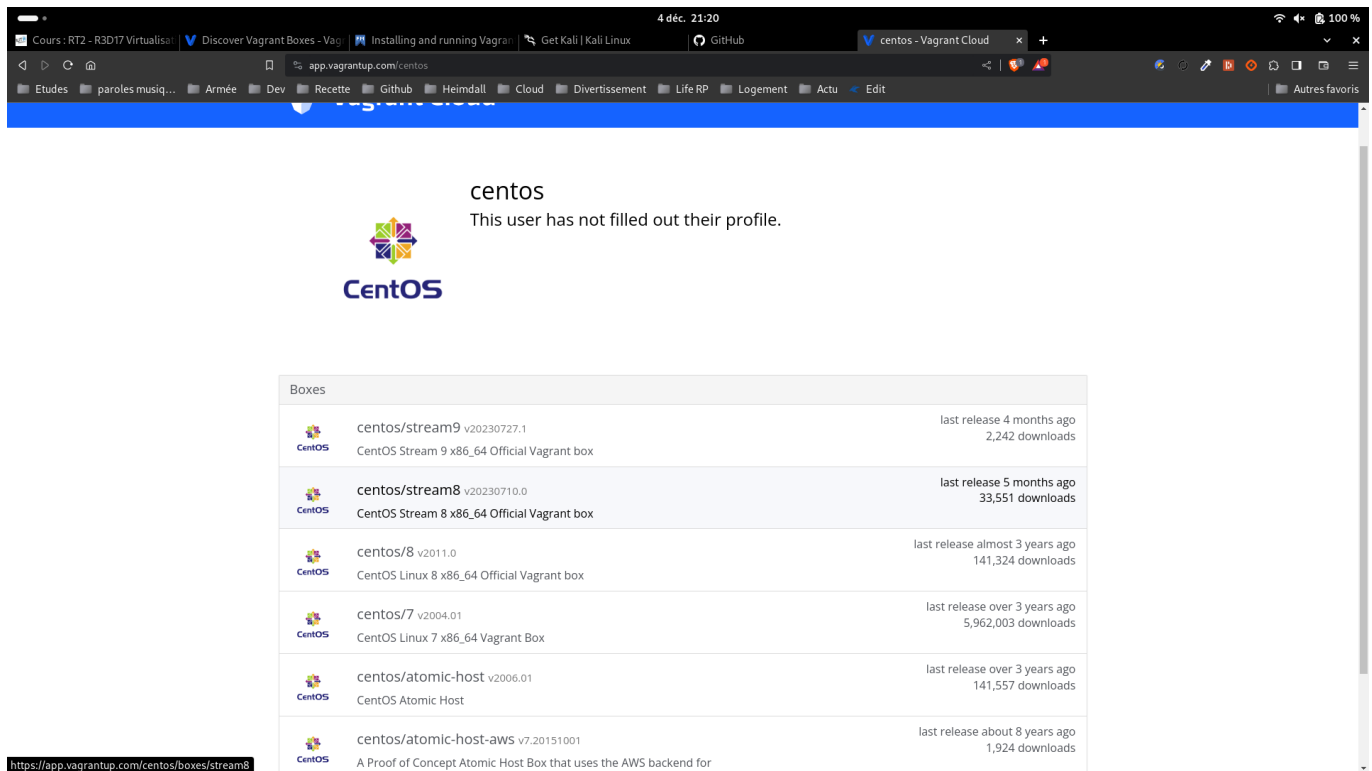
The contrib-* boxes include components outside the main archive: namely the vboxsf kernel module for shared folders.

Quick start to get your first Debian VM running
<https://wiki.debian.org/Teams/Cloud/VagrantQuickStart>

Documentation:
<https://wiki.debian.org/Teams/Cloud/VagrantBaseBoxes>

Oui car l'image n'est pas officiel au projet debian mais est maintenu par l'equipe de debian.

(c) Même question pour centos/7 (il faut chercher un peu plus...)



Oui car l'image est officiel mais n'a pas d'indication sur son profil

2. Ajouter les images debian/jessie64 et centos/7 à votre bibliothèque locale (vagrant box --help)

```
vagrant box add --name debian/jessie64
```

```
https://app.vagrantup.com/debian/boxes/jessie64
```

```
vagrant box add --name centos/7 https://app.vagrantup.com/centos/boxes/7
```

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant box add --name debian/jessie64 https://app.vagrantup.com/debian/boxes/jessie64
==> box: Loading metadata for box 'https://app.vagrantup.com/debian/boxes/jessie64'
==> box: Adding box 'debian/jessie64' (v8.11.1) for provider: virtualbox
        box: Downloading: https://vagrantcloud.com/debian/boxes/jessie64/versions/8.11.1/providers/virtualbox/unknown/vagrant.box
==> box: Box download is resuming from prior download progress
==> box: Successfully added box 'debian/jessie64' (v8.11.1) for 'virtualbox'!
bastien_fedora@fedora:~/Vagrant/VM1$
```

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant box add --name centos/7 https://app.vagrantup.com/centos/boxes/7
==> box: Loading metadata for box 'https://app.vagrantup.com/centos/boxes/7'
This box can work with multiple providers! The providers that it
can work with are listed below. Please review the list and choose
the provider you will be working with.

1) hyperv
2) libvirt
3) virtualbox
4) vmware_desktop

Enter your choice: 2
==> box: Adding box 'centos/7' (v2004.01) for provider: libvirt
        box: Downloading: https://vagrantcloud.com/centos/boxes/7/versions/2004.01/providers/libvirt/unknown/vagrant.box
Download redirected to host: cloud.centos.org
        box: Calculating and comparing box checksum...
==> box: Successfully added box 'centos/7' (v2004.01) for 'libvirt'!
bastien_fedora@fedora:~/Vagrant/VM1$
```

3.2 Autres commandes Vagrant et interaction avec VirtualBox :

1. Pour créer l'image à partir de VM1, Vagrant a stoppé VM1. Vous pouvez le vérifier avec la commande `vagrant status`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant status
Current machine states:

default                                poweroff (virtualbox)

The VM is powered off. To restart the VM, simply run `vagrant up`
```

Vous pouvez aussi voir le status de toutes les VMs actives et pas seulement celle du répertoire courant avec la commande `vagrant global-status`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant global-status
id      name      provider  state    directory
-----
d387189 default virtualbox poweroff /home/bastien_fedora/Vagrant/VM1

The above shows information about all known Vagrant environments
on this machine. This data is cached and may not be completely
up-to-date (use "vagrant global-status --prune" to prune invalid
entries). To interact with any of the machines, you can go to that
directory and run Vagrant, or you can use the ID directly with
Vagrant commands from any directory. For example:
"vagrant destroy 1a2b3c4d"
```

2. Redémarrez VM1. (avec quelle commande ?)

```
vagrant up
```

3. Vous pouvez faire des snapshots(instantané) de VM1 avec la commande `vagrant snapshot save $(date +%Y%m%d%H%M%S)`

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant snapshot save $(date +%Y%m%d%H%M%S)
==> default: Snapshotting the machine as '20231204215421'...
==> default: Snapshot saved! You can restore the snapshot at any time by
==> default: using `vagrant snapshot restore`. You can delete it using
==> default: `vagrant snapshot delete`.
bastien_fedora@fedora:~/Vagrant/VM1$
```

Quels nouveaux fichiers sont créés dans le répertoire de VM1 de virtualbox (lister)

```

bastien_fedora@fedora:~/VirtualBox VMs/VM1_default_1701704648608_12974$ tree
.
├── Logs
│   ├── VBox.log
│   ├── VBox.log.1
│   ├── VBox.log.2
│   ├── VBox.log.3
│   └── VBoxUI.log
├── Snapshots
│   ├── 2023-12-04T20-54-24-696645000Z.sav
│   ├── {63282adc-46f8-40a5-a2f0-54ef6bc72297}.vmdk
│   ├── {9c2943db-091d-4dbf-99fa-68581d32b805}.vmdk
│   ├── ubuntu-xenial-16.04-cloudimg-configdrive.vmdk
│   ├── ubuntu-xenial-16.04-cloudimg.vmdk
│   ├── VM1_default_1701704648608_12974.vbox
│   └── VM1_default_1701704648608_12974.vbox-prev
└──
3 directories, 12 files
bastien_fedora@fedora:~/VirtualBox VMs/VM1_default_1701704648608_12974$

```

Note : la commande inverse est `vagrant snapshot restore [nom_du_snapshot]` et repart dans l'état précédent.

Vous pouvez aussi faire un `vagrant snapshot delete [nom_du_snapshot]` pour supprimer un snapshot.

Vous pouvez aussi lister les snapshots avec `vagrant snapshot list`

4. Vous pouvez mettre la VM en veille avec

`vagrant suspend`

Vous pouvez verifier dans l'interface VirtualBox l'état de la VM.



5. Mettre en veille la VM ou faire un instantané sont des operations quasiment similaires. Quels fichiers ont été créés par Virtualbox pour cette opération et quelle est la différence avec l'operation snapshot ?

```
bastien_fedora@fedora:~/VirtualBox VMs/VM1_default_1701704648608_12974$ tree
.
├── Logs
│   ├── VBox.log
│   ├── VBox.log.1
│   ├── VBox.log.2
│   ├── VBox.log.3
│   └── VBoxUI.log
├── Snapshots
│   ├── 2023-12-04T20-54-24-696645000Z.sav
│   ├── 2023-12-04T20-58-24-439239000Z.sav
│   ├── {63282adc-46f8-40a5-a2f0-54ef6bc72297}.vmdk
│   ├── {9c2943db-091d-4dbf-99fa-68581d32b805}.vmdk
│   ├── ubuntu-xenial-16.04-cloudimg-configdrive.vmdk
│   ├── ubuntu-xenial-16.04-cloudimg.vmdk
│   ├── VM1_default_1701704648608_12974.vbox
│   └── VM1_default_1701704648608_12974.vbox-prev
3 directories, 13 files
bastien_fedora@fedora:~/VirtualBox VMs/VM1_default_1701704648608_12974$
```

La différence est que la VM est en veille et non pas en pause comme avec un snapshot, le fichier est donc temporaire et sera supprimé au prochain reboot de la VM.

6. Vous pouvez reprendre la VM avec la commande `vagrant resume` Vous pouvez voir les redirections de port avec la commande `vagrant port` Que voyez vous ?

```
bastien_fedora@fedora:~/Vagrant/VM1$ vagrant port
The forwarded ports for the machine are listed below. Please note that
these values may differ from values configured in the Vagrantfile if the
provider supports automatic port collision detection and resolution.

    22 (guest) => 2222 (host)
    80 (guest) => 8080 (host)
bastien_fedora@fedora:~/Vagrant/VM1$
```

7. Vagrant interagit avec l'hyperviseur qui gère les VMs, ici VirtualBox. Souvent, il peut utiliser des fonctions avancées. Dans le fichier Vagrantfile suivant, quelles fonctions avancées sont utilisées et quel est leur intérêt ?

On peut ainsi configurer la machine complète via le fichier Vagrantfile et non pas via des commandes shell.

3.3 Vagrant Multi-machines :

soit le vagrantfile ci-après :

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
# Vagrantfile API/syntax version. Don't touch unless you know what you're
doing!
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
# All Vagrant configuration is done here. The most common configuration
```

```
# options are documented and commented below. For a complete reference,
# please see the online documentation at vagrantup.com.
# Every Vagrant virtual environment requires a box to build off of.
config.vm.define "database" do |db|
  db.vm.box = "puppetlabs/ubuntu-14.04-64-nocm"
  db.vm.hostname = "db01"
  db.vm.network "private_network", ip: "192.168.55.100"
end
config.vm.define "web" do |web|
  web.vm.box = "puppetlabs/ubuntu-14.04-64-nocm"
  web.vm.hostname = "web01"
  web.vm.network "private_network", ip: "192.168.55.101"
  web.vm.provision "shell",
  inline: "echo '127.0.0.1 localhost web01\n192.168.55.100 db01' >
/etc/hosts"
end
end
```

1. Modifiez le script de manière à ce que ce soit l'image de ubuntu/xenial64 qui soit utilisée.

```
# -*- mode: ruby -*-
# vi: set ft=ruby :
# Vagrantfile API/syntax version. Don't touch unless you know what you're
doing!
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  # All Vagrant configuration is done here. The most common configuration
  # options are documented and commented below. For a complete reference,
  # please see the online documentation at vagrantup.com.
  # Every Vagrant virtual environment requires a box to build off of.
  config.vm.define "database" do |db|
    db.vm.box = "ubuntu/xenial64"
    db.vm.hostname = "db01"
    db.vm.network "private_network", ip: "192.168.55.100"
  end

  config.vm.define "web" do |web|
    web.vm.box = "ubuntu/xenial64"
    web.vm.hostname = "web01"
    web.vm.network "private_network", ip: "192.168.55.101"
    web.vm.provision "shell",
    inline: "echo '127.0.0.1 localhost web01\n192.168.55.100 db01' >
/etc/hosts"
  end
end
```

2. Vous pouvez vous logez sur une machine ou sur l'autre avec la commande :

```
vagrant ssh [nom_de_la_machine]
```

Logez vous sur *web* et montrer que l'on peut ping 192.168.55.100

```
vagrant@web01:~$ ping 192.168.55.100
PING 192.168.55.100 (192.168.55.100) 56(84) bytes of data.
64 bytes from 192.168.55.100: icmp_seq=1 ttl=64 time=0.359 ms
64 bytes from 192.168.55.100: icmp_seq=2 ttl=64 time=0.599 ms
64 bytes from 192.168.55.100: icmp_seq=3 ttl=64 time=0.703 ms
64 bytes from 192.168.55.100: icmp_seq=4 ttl=64 time=0.702 ms
^C
--- 192.168.55.100 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3317ms
rtt min/avg/max/mdev = 0.359/0.590/0.703/0.143 ms
vagrant@web01:~$
```

3. Pourquoi est ce que :

ping db01 fonctionne aussi ?

```
vagrant@web01:~$ ping db01
PING db01 (192.168.55.100) 56(84) bytes of data.
64 bytes from db01 (192.168.55.100): icmp_seq=1 ttl=64 time=0.374 ms
64 bytes from db01 (192.168.55.100): icmp_seq=2 ttl=64 time=0.526 ms
64 bytes from db01 (192.168.55.100): icmp_seq=3 ttl=64 time=0.520 ms
^C
--- db01 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.374/0.473/0.526/0.072 ms
vagrant@web01:~$
```

Car dans le fichier hosts de la machine web01 il y a une ligne qui permet de faire le lien entre db01 et l'adresse ip

4. Peut-on pinguer les 2 VMs depuis la machine physique ? Pourquoi ?

```

bastien_fedora@fedora:~/Vagrant/VM1$ ping 192.168.55.100 -c 4 && ping 192.168.55.101 -c 4
PING 192.168.55.100 (192.168.55.100) 56(84) octets de données.
64 octets de 192.168.55.100 : icmp_seq=1 ttl=64 temps=0.491 ms
64 octets de 192.168.55.100 : icmp_seq=2 ttl=64 temps=0.376 ms
64 octets de 192.168.55.100 : icmp_seq=3 ttl=64 temps=0.513 ms
64 octets de 192.168.55.100 : icmp_seq=4 ttl=64 temps=0.388 ms

--- statistiques ping 192.168.55.100 ---
4 paquets transmis, 4 reçus, 0% packet loss, time 3081ms
rtt min/avg/max/mdev = 0.376/0.442/0.513/0.060 ms
PING 192.168.55.101 (192.168.55.101) 56(84) octets de données.
64 octets de 192.168.55.101 : icmp_seq=1 ttl=64 temps=0.470 ms
64 octets de 192.168.55.101 : icmp_seq=2 ttl=64 temps=0.494 ms
64 octets de 192.168.55.101 : icmp_seq=3 ttl=64 temps=0.573 ms
64 octets de 192.168.55.101 : icmp_seq=4 ttl=64 temps=0.527 ms

--- statistiques ping 192.168.55.101 ---
4 paquets transmis, 4 reçus, 0% packet loss, time 3070ms
rtt min/avg/max/mdev = 0.470/0.516/0.573/0.038 ms
bastien_fedora@fedora:~/Vagrant/VM1$

```

Oui car les VMs sont en réseau privé et on a un bridge via le pc physique mais ne sont pas accessibles depuis l'extérieur.

5. Notez, en remarque finale, que l'on peut être plus générique en créant n machines où n est un nombre fixé par l'administrateur, comme on peut le voir dans le script ci-après. Quelle sera la contrainte physique du serveur à prendre en compte dans le choix de n ?

La limite sera la puissance de la machine physique qui va devoir supporter n VMs.

```

# -*- mode: ruby -*-
# vi: set ft=ruby :
# Vagrantfile API/syntax version. Don't touch unless you know what you're
# doing!
VAGRANTFILE_API_VERSION = "2"
# Define a variable - the number of web nodes.
$cluster_nodes = 3
$consul_server_ip = "192.168.30.130"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  # Define a global box file to be used by all machines.
  config.vm.box = "puppetlabs/ubuntu-14.04-64-puppet"
  # Create and provision a Consul server machine.
  config.vm.define "consul" do |consul|
    consul.vm.hostname = "consul"
    consul.vm.network "private_network", ip: $consul_server_ip
    consul.vm.provision "shell", inline: ["apt-get update && apt-get install",
    "unzip"]
    consul.vm.provision "puppet" do |puppet|
      puppet.manifests_path = "puppet/manifests"
      puppet.module_path = "puppet/modules"
      puppet.manifest_file = "site.pp"
    end
  end
  # Create a number of cluster members.

```



```
(1..$cluster_nodes).each do |i|
  config.vm.define vm_name = "cluster%02d" % i do |cluster|
    cluster.vm.hostname = vm_name
    cluster.vm.provision "shell", inline: "apt-get update && apt-get
install -y unzip"
    cluster.vm.provision "puppet" do |puppet|
      puppet.manifests_path = "puppet/manifests"
      puppet.module_path = "puppet/modules"
      puppet.manifest_file = "site.pp"
    end
    cluster.vm.provision "shell", inline: "consul join #{$consul_server_ip}
"
  end
end
end
```