

# Introduction à Vagrant

Avant de commencer, assurez-vous que vous disposez déjà d'une solution de virtualisation sur votre système. Les solutions qui fonctionnent avec Vagrant incluent VirtualBox, VMware (payant), Docker, Hyper-V et des solutions personnalisées.

## Installation

1. Pour trouver la dernière version de Vagrant, utilisez un navigateur Web pour accéder à sa page Web officielle:

<https://www.vagrantup.com/downloads.html>

2. Vous verrez une liste de tous les différents systèmes d'exploitation pris en charge, avec un package 32 bits et 64 bits pour chacun. Téléchargez le fichier approprié pour votre système d'exploitation, puis exécutez le programme d'installation. Vous pouvez également utiliser git for windows pour une ligne de commande bash sous Windows 10 disponible sur le lien : <https://gitforwindows.org/>

3. Il existe deux façons de vérifier si l'installation a réussi:

Vous pouvez soit utiliser cmd ou git-bash sous Windows (<https://gitforwindows.org/>) :

```
$ vagrant -v
Vagrant 2.2.7
```

qui devrait montrer le numéro de version en cours d'exécution sur votre ordinateur. La dernière version en date est Vagrant 2.2.7.

Ou vous pouvez taper la commande suivante dans le terminal:

```
$ vagrant.exe
Usage: vagrant [options] <command> [<args>]

    -v, --version          Print the version and exit.
    -h, --help             Print this help.

Common commands:
    box                    manages boxes: installation, removal, etc.
    cloud                  manages everything related to Vagrant Cloud
    destroy                stops and deletes all traces of the vagrant machine
    global-status          outputs status Vagrant environments for this user
    ...

For help on any individual command run `vagrant COMMAND -h`

Additional subcommands are available, but are either more advanced
or not commonly used. To see all subcommands, run the command
`vagrant list-commands`.
```

Cette sortie vous affichera une liste des commandes fréquemment utilisées si l'outil était installé correctement.

## Configuration du projet Vagrant

Aujourd'hui, nous allons utiliser une des différentes boxes que Vagrant met à disposition sur son site : <https://app.vagrantup.com/boxes/search>

1. Commencez par créer un répertoire pour stocker votre fichier Vagrant:

```
$ mkdir projet_vagrant
$ cd projet_vagrant
```

2. Téléchargez la distribution Centos7 à partir d'une bibliothèque commune et créez un Vagrantfile de base avec:

```
$ vagrant cloud search centos
+-----+-----+-----+-----+
| NAME | VERSION | DOWNLOADS | PROVIDERS |
+-----+-----+-----+-----+
| centos/7 | 1905.1 | 5,149,818 | virtualbox,vmware_desktop,libvirt,hyper |
| bento/centos-6.7 | 2.2.7 | 1,548,291 | parallels,vmware_desktop,virtualbox |
| generic/centos7 | 2.0.6 | 1,066,214 | vmware_desktop,virtualbox,parallels,libvirt,hyper |
| bento/centos-7.2 | 2.3.1 | 1,055,677 | vmware_desktop,virtualbox,parallels |
| puppet/centos65-x64 | 20161102 | 846,916 | vmware_desktop,virtualbox,parallels |
| bento/centos-7.1 | 2.2.2 | 793,656 | parallels,virtualbox,vmware_desktop |
| generic/centos6 | 2.0.6 | 708,919 | vmware_desktop,virtualbox,parallels,libvirt,hyper |
| scalefactory/centos6 | 1.4.14 | 553,757 | libvirt,virtualbox |
| geerlingguy/centos7 | 1.2.21 | 458,725 | virtualbox |
| puppetlabs/centos-6.6-64-puppet | 1.0.3 | 440,965 | vmware_desktop,virtualbox |
+-----+-----+-----+-----+

$ vagrant box add centos/7
==> box: Loading metadata for box 'centos/7'
    box: URL: https://vagrantcloud.com/centos/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

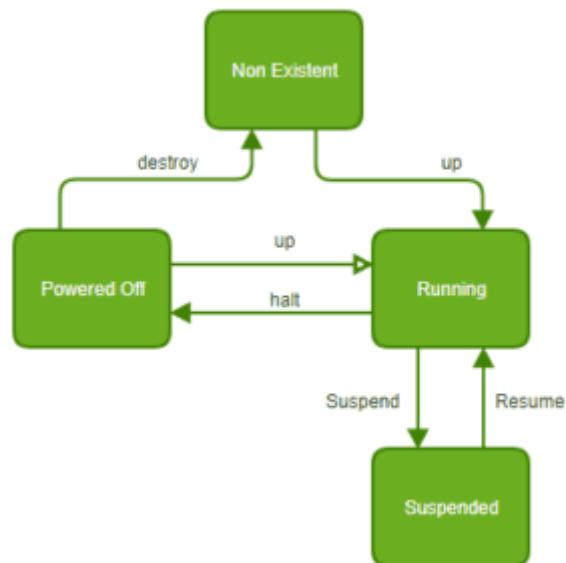
Enter your choice: 3
==> box: Adding box 'centos/7' (v1905.1) for provider: virtualbox
    box: Downloading:
https://vagrantcloud.com/centos/boxes/7/versions/1905.1/pr
    box: Download redirected to host: cloud.centos.org
    box:
==> box: Successfully added box 'centos/7' (v1905.1) for 'virtualbox'!

$ vagrant init centos/7 --provider virtualbox
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.
```

Si vous le souhaitez, vous pouvez accéder à <https://app.vagrantup.com/boxes/search> et télécharger une Vagrantbox de votre choix.

Lorsque vous exécutez la commande *vagrant init*, Vagrant installe la boîte dans le répertoire actuel. Le Vagrantfile est placé dans le même répertoire et peut être édité ou copié.

Vous pouvez ouvrir le fichier Vagrantfile précédemment créé et vous inspirer des commentaires pour adapter sa configuration selon vos besoins de RAM, CPU, @IP, fichiers partagés ou commandes post création...



### Cycle de vie des Boxes Vagrant

Dès lors, il ne vous reste qu'à utiliser la commande *vagrant up* pour créer la machine virtuelle :

```
$ vagrant up --provider virtualbox
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Checking if box 'centos/7' version '1905.1' 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: 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: Machine booted and ready!
==> default: Checking for guest additions in VM...
default: No guest additions were detected on the base box for this VM! Guest
default: additions are required for forwarded ports, shared folders, host only
default: networking, and more. If SSH fails on this machine, please install
default: the guest additions and repackage the box to continue.
default:
default: This is not an error message; everything may continue to work
properly,
default: in which case you may ignore this message.
==> default: Rsyncing folder: /cygdrive/d/VirtualBox_VMs/projet_vagrant/ =>
/vagrant
==> default: Machine already provisioned. Run `vagrant provision` or use
the `--provision`
==> default: flag to force provisioning. Provisioners marked to run always
will still run.
```

Pour expliquer ce qui se passe ici, Vagrant importe d'abord la boîte de base si vous ne l'avez pas déjà (vous l'avez déjà, sinon elle serait téléchargée en premier). Il vérifie ensuite si votre box est à jour.

**Remarque:** ces boîtes de base existent dans votre dossier `%userprofile%\.vagrant.d/boxes`. Vous pouvez répertorier toutes les boîtes installées en tapant `vagrant box list`. Vous pouvez supprimer des boîtes avec `vagrant box remove <box name>`.

```
$ vagrant box list
centos/7 (virtualbox, 1905.1)
```

Vous pouvez voir qu'il configure le port de transfert SSH comme "2222": vous en aurez besoin pour utiliser un client SSH comme PuTTY.

Notez que le nom d'utilisateur est «vagrant», ce qu'il sera toujours. Vous remarquerez également qu'il génère un nouvel ensemble de clés. La clé publique est automatiquement copiée dans la machine virtuelle et y remplace la clé par défaut. Il place également la clé privée dans un nouveau dossier appelé «.vagrant» dans votre dossier de projet. Si vous explorez ce dossier, vous trouverez «private\_key», qui sera utile pour une utilisation ultérieure dans PuTTY.

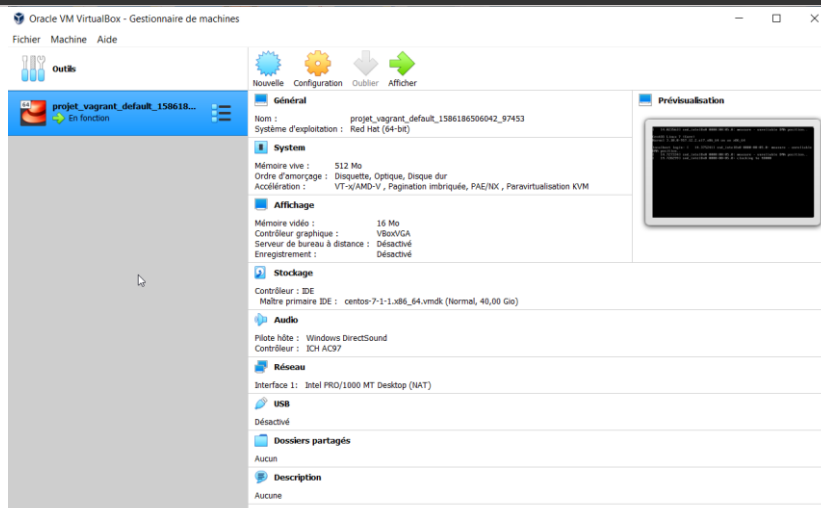
Remarquez ensuite qu'il vérifie les *Guest additions* et monte le dossier partagé entre l'hôte et l'invité. Cela peut être modifié comme vous le souhaitez, ou des dossiers partagés supplémentaires peuvent être ajoutés. Le premier dossier «/vagrant» est sur la VM; il sera partagé avec mon dossier de projet local. Les fichiers seront synchronisés entre les dossiers pendant que vous travaillez.

En vérifiant sous VirtualBox, ou par la commande suivante, on voit bien que la machine a bien été créée.

```
$ vagrant status
Current machine states:

default                                running (virtualbox)

The VM is running. To stop this VM, you can run `vagrant halt` to
shut it down forcefully, or you can run `vagrant suspend` to simply
suspend the virtual machine. In either case, to restart it again,
simply run `vagrant up`.
```



## Se connecter à la machine virtuelle

Un utilisateur nommé vagrant dont le mot de passe est vagrant peut être utilisé pour se connecter à la machine. Ce dernier dispose des droits *sudo* et peut donc passer en root (*sudo su -*).

Vous pouvez vous connecter à votre machine virtuelle (et vérifier qu'elle fonctionne) en utilisant une connexion SSH:

```
$ vagrant ssh
[vagrant@localhost ~]$
```

Cela ouvre une connexion shell sécurisée à la nouvelle machine virtuelle. Votre invite de commande passera à *vagrant@localhost* pour indiquer que vous êtes connecté à la machine virtuelle.

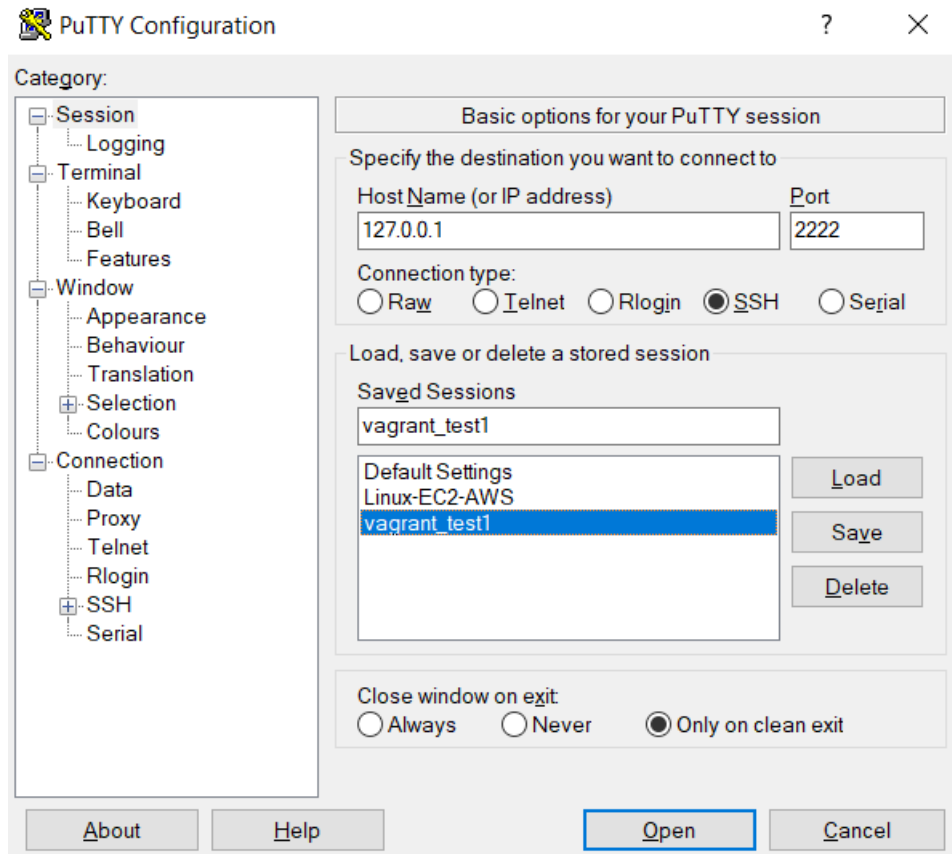
Une fois que vous avez fini d'explorer la machine virtuelle, vous pouvez quitter la session avec CTRL-D . La machine virtuelle fonctionnera toujours en arrière-plan, mais la connexion SSH sera fermée.

Tapez la commande suivante pour obtenir vos informations SSH chaque fois que vous en avez besoin, elle vous indiquera le port IP et SSH à utiliser et quelques autres informations si vous oubliez.

```
$ vagrant ssh-config
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile
D:/VirtualBox_VMs/projet_vagrant/.vagrant/machines/default/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL
```

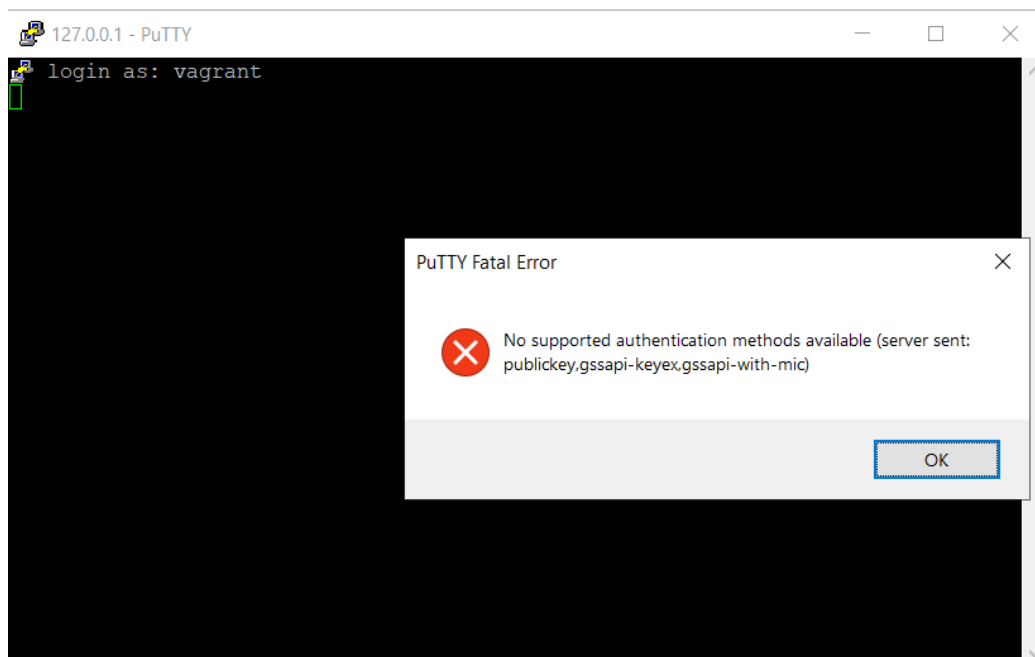
Une autre façon de se connecter à la VM est d'utiliser PUTTY ou MobaXterm ou un autre client GUI SSH.

Ouvrez PuTTY, entrez l'IP et le port, donnez un nom à la connexion et enregistrez-la. Vous pouvez maintenant vous connecter au nouveau serveur via SSH avec PuTTY. Le nom d'utilisateur est «vagrant» et le mot de passe doit également être «vagrant».



Vagrant choisira automatiquement un autre port si le 2222 par défaut est pris. Toutes vos machines virtuelles recevront un nouveau port, alors assurez-vous de noter quelle machine virtuelle est laquelle!

Si le message d'erreur suivant apparaît c'est que votre box est par défaut configurée pour une authentification à base de clés, suivez les étapes suivantes.

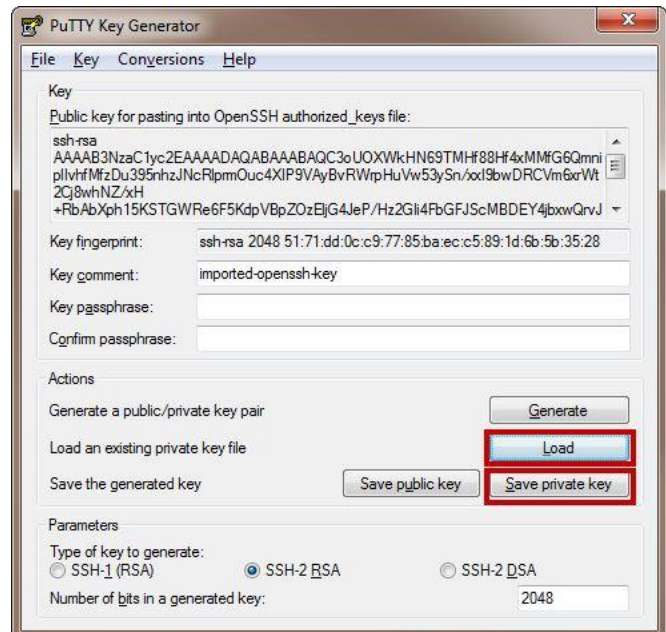


## Utilisation de clés publiques et privées

Vous pouvez utiliser la nouvelle clé privée créée par Vagrant, car la clé privée a été stockée sur votre ordinateur local. Cependant, PuTTY ne peut pas utiliser la clé de style OpenSSH. C'est pourquoi nous avons téléchargé PuTTYGen, alors ouvrez-le maintenant.

Cliquez sur «Charger» et accédez à votre dossier de projet jusqu'à  
\*\*\\.\vagrant\machines\default\virtualbox\*\*.  
Basculez pour voir «tous les fichiers»,  
sélectionnez «private\_key» et cliquez sur  
Ouvrir.

Vous pouvez immédiatement cliquer sur le bouton «Enregistrer la clé privée», sélectionner «Oui» pour enregistrer sans mot de passe. Nommez-le quelque chose de similaire comme "private\_key\_putty" dans le même dossier. Cela créera un fichier avec une extension .PPK. Fermez PuTTYGen et revenez à PuTTY maintenant.



Dans PuTTY, chargez votre connexion sauvegardée ou saisissez-la à nouveau si vous ne l'avez pas enregistrée. Ensuite, allez dans Connexion-> SSH-> Auth dans la barre latérale et cliquez sur «Parcourir» pour trouver la clé privée, vous y verrez la version PPK. Assurez-vous maintenant d'enregistrer votre configuration afin de pouvoir continuer à l'utiliser plus tard.

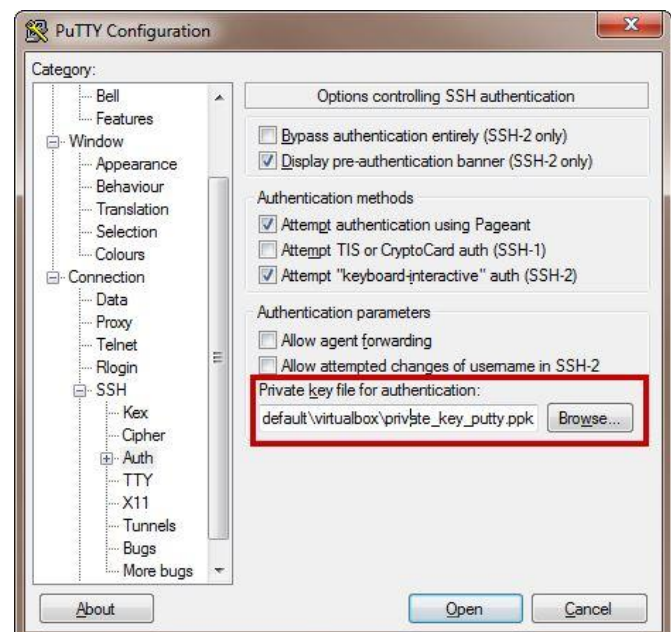
## Le fichier Vagrantfile

Le fichier Vagrantfile contient la description de la machine virtuelle et la manière avec laquelle elle sera créée et configurée. Ce fichier est prévu pour être partagé facilement et poussé dans un gestionnaire de versions tel que GIT permettant ainsi aux différents développeurs d'un projet de le récupérer et avoir un environnement d'exécution identique à l'aide d'une simple commande.

Il est possible de configurer d'autres éléments de notre VM grâce aux paramètres suivants :

**config.vm.network** : Si vous désirez transférer le port 80 au cas où vous feriez du travail de développement Web.

**config.vm.hostname (chaîne)** : Le nom d'hôte que la machine doit avoir, le nom d'hôte sera défini au démarrage et Vagrant mettra à jour /etc/hosts sur le guest avec le nom d'hôte configuré.



```

$ mkdir webapp1
$ cd webapp1/

$ vi vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 512
    vb.cpus = 1
  end
  config.vm.box = "centos/7"
  config.vm.define :webapp1 do |webapp1_config|
    webapp1_config.vm.host_name = "webapp1"
    webapp1_config.vm.network "private_network", ip: "192.168.10.10"
    webapp1_config.vm.network "forwarded_port", guest: 80, host: 8080
    webapp1_config.vm.hostname = "webapp1"
  end
end

$ vagrant up
Bringing machine 'webapp1' up with 'virtualbox' provider...
==> webapp1: Importing base box 'centos/7'...
==> webapp1: Matching MAC address for NAT networking...
==> webapp1: Checking if box 'centos/7' version '1905.1' is up to date...
==> webapp1: Setting the name of the VM:
webapp1_webapp1_1586225876832_72645
==> webapp1: Clearing any previously set network interfaces...
==> webapp1: Preparing network interfaces based on configuration...
webapp1: Adapter 1: nat
webapp1: Adapter 2: hostonly
==> webapp1: Forwarding ports...
webapp1: 80 (guest) => 8080 (host) (adapter 1)
...

$ vagrant ssh
[vagrant@webapp1 ~]$ ip a
...
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 52:54:00:8a:fe:e6 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global noprefixroute dynamic eth0
        valid_lft 86136sec preferred_lft 86136sec
    inet6 fe80::5054:ff:fe8a:fee6/64 scope link
        valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group
default qlen 1000
    link/ether 08:00:27:91:8a:a6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.10.10/24 brd 192.168.10.255 scope global noprefixroute eth1
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe91:8aa6/64 scope link
        valid_lft forever preferred_lft forever

$ 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)

```



## Provisioning

Très bien, nous avons donc une machine virtuelle exécutant une copie de base de Centos et nous pouvons modifier les fichiers de notre machine et les synchroniser dans la machine virtuelle. Servons maintenant ces fichiers à l'aide d'un serveur Web.

Vagrant a une prise en charge intégrée pour l'approvisionnement automatisé. En utilisant cette fonctionnalité, Vagrant installera automatiquement le logiciel lorsque vous faites *vagrant up* afin que la machine invitée puisse être créée de manière répétée et prête à l'emploi.

Cependant, étant donné que la même commande «*vagrant up*» peut être utilisée pour réveiller ou démarrer une machine virtuelle qui a déjà été créée, Vagrant doit être informé si nous souhaitons détruire et provisionner à nouveau la machine en utilisant l'option *--provision* ou au contraire *--no-provision* sinon.

## Installer Apache

Nous allons simplement configurer Apache pour notre projet de base, et nous le ferons en utilisant un script shell. Créez le script shell suivant et enregistrez le fichier *bootstrap.sh* dans le même répertoire que votre Vagrantfile:

```
$ vi bootstrap.sh
#!/usr/bin/env bash

yum update -y
yum install -y httpd
if ! [ -L /var/www/html ]; then
    rm -rf /var/www/html
    ln -s /vagrant /var/www/html
fi
systemctl enable httpd --now
setenforce 0
```

Ensuite, nous configurons Vagrant pour exécuter ce script shell lors de la configuration de notre machine. Nous le faisons en éditant le Vagrantfile, qui devrait maintenant ressembler à ceci:

```
$ vi Vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

Vagrant.configure("2") do |config|
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 512
    vb.cpus = 1
  end
  config.vm.box = "centos/7"
  config.vm.define :webapp1 do |webapp1_config|
    webapp1_config.vm.host_name = "webapp1"
    webapp1_config.vm.network "private network", ip: "192.168.10.10"
    webapp1_config.vm.network "forwarded port", guest: 80, host: 8080
    webapp1_config.vm.hostname = "webapp1"
    config.vm.provision :shell, path: "bootstrap.sh"
  end
end
```

```
$ vi index.html
<!DOCTYPE html>
<html>
  <body>
    <h1>Getting started with Vagrant!</h1>
  </body>
</html>

$ vagrant reload --provision
==> webapp1: Attempting graceful shutdown of VM...
==> webapp1: Checking if box 'centos/7' version '1905.1' is up to date...
...
==> webapp1: Setting hostname...
==> webapp1: Rsyncing folder:
/cygdrive/d/VirtualBox_VMs/projet_vagrant/webapp1/ => /vagrant
==> webapp1: Machine not provisioned because '--no-provision' is specified.
```

Si on désire synchroniser un autre répertoire que /vagrant on utilisera les paramètres de configuration de dossiers synchronisés de Vagrant comme suit :

```
$ mkdir data
$ touch data/datafile
$ vim Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 512
    vb.cpus = 1
  end
  config.vm.box = "centos/7"
  config.vm.define :webapp1 do |webapp1_config|
    webapp1_config.vm.host_name = "webapp1"
    webapp1_config.vm.network "private_network", ip: "192.168.10.10"
    webapp1_config.vm.network "forwarded_port", guest: 80, host: 8080
    webapp1_config.vm.hostname = "webapp1"
    webapp1_config.vm.synced_folder "data/", "/mnt"
    config.vm.provision :shell, path: "bootstrap.sh"
  end
end

$ vagrant.exe reload --provision
==> webapp1: Attempting graceful shutdown of VM...
==> webapp1: Checking if box 'centos/7' version '1905.1' is up to date...
==> webapp1: Clearing any previously set forwarded ports...
==> webapp1: Clearing any previously set network interfaces...
==> webapp1: Preparing network interfaces based on configuration...
...
==> webapp1: Mounting shared folders...
   webapp1: /mnt => D:/VirtualBox_VMs/projet_vagrant/webapp1/data
Vagrant was unable to mount VirtualBox shared folders. This is usually
because the filesystem "vboxsf" is not available. This filesystem is
made available via the VirtualBox Guest Additions and kernel module.
Please verify that these guest additions are properly installed in the
guest. This is not a bug in Vagrant and is usually caused by a faulty
Vagrant box. For context, the command attempted was:

mount -t vboxsf -o uid=1000,gid=1000 mnt /mnt

The error output from the command was:

mount: unknown filesystem type 'vboxsf'
```

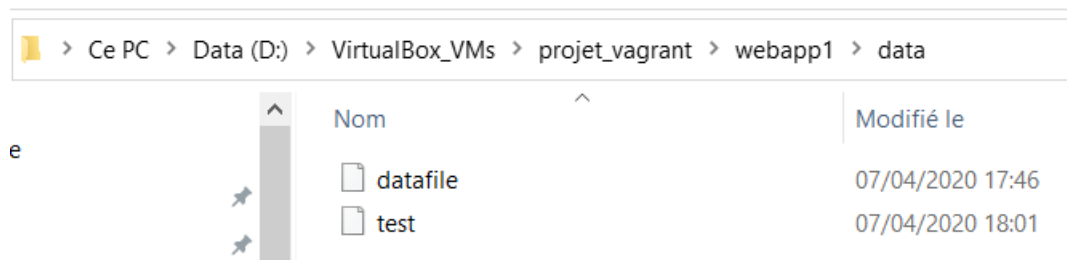
Comme le message d'erreur l'indique, il faut installer VirtualBox Guest Additions les pour améliorer l'intégration des Guests.

```
$ vagrant plugin install vagrant-vbguest
Installing the 'vagrant-vbguest' plugin. This can take a few minutes...
Installed the plugin 'vagrant-vbguest (0.23.0)'.

$ vagrant reload --provision
```

```
$ vagrant.exe ssh
Last login: Tue Apr  7 16:39:18 2020 from 10.0.2.2
[vagrant@webapp1 ~]$ touch /mnt/test
[vagrant@webapp1 ~]$ ll /mnt/
total 0
-rwxrwxrwx. 1 vagrant vagrant 0 Apr  7 16:46 datafile
-rwxrwxrwx. 1 vagrant vagrant 0 Apr  7 17:01 test
```

Vous trouverez le fichier à peine créé sous Windows dans le dossier data de votre projet.



## Nettoyer Vagrant

Une fois que vous avez fini de travailler sur votre système invité, vous avez quelques options pour terminer la session.

1. Pour arrêter la machine et enregistrer son exécution en cours:

```
$ vagrant suspend
```

Vous pouvez reprendre en exécutant à nouveau *vagrant up*. C'est un peu comme mettre la machine en mode veille.

2. Pour arrêter la machine virtuelle, utilisez la commande:

```
$ vagrant halt
```

Encore une fois, *vagrant up* redémarrera la même machine virtuelle et vous pourrez reprendre là où vous vous étiez arrêté. Cela ressemble beaucoup à l'arrêt d'une machine ordinaire.

3. Pour supprimer toutes les traces de la machine virtuelle de votre système, tapez ce qui suit:

```
$ vagrant destroy
```

Tout ce que vous avez enregistré dans la machine virtuelle sera supprimé. Cela libère les ressources système utilisées par Vagrant. La prochaine fois que vous ferez *vagrant up*, la machine devra être réimportée et réapprovisionnée. C'est un peu comme formater un disque dur sur un système, puis recharger une nouvelle image .