

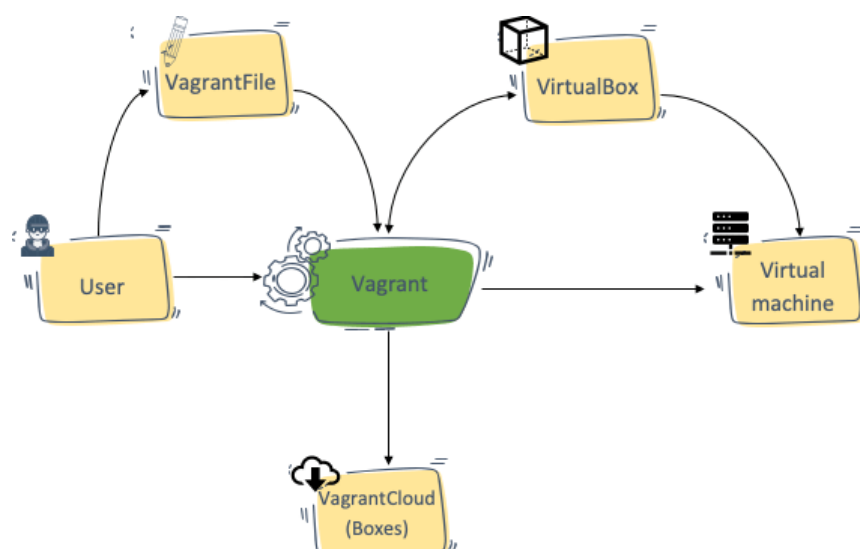
Installation et configuration de Vagrant

Vagrant est un outil open-source qui vous permet de créer, configurer et gérer des boîtes de machines virtuelles via une interface de commande facile à utiliser. Il s'agit essentiellement d'une couche de logiciel installée entre un outil de virtualisation (tel que VirtualBox, Docker, Hyper-V) et une VM.

Il est souvent utilisé dans le cadre du développement de logiciels pour s'assurer que tous les membres de l'équipe travaillent sur la même configuration. Non seulement il partage les environnements, mais il partage également le code. Cela permet au code d'un développeur de fonctionner sur le système d'un autre, rendant possible le développement collaboratif et coopératif.

Ce tutoriel vous explique tout ce que vous devez savoir sur la configuration et la gestion de Vagrant.

Architecture de vagrant :



- **Vagrantfile :**

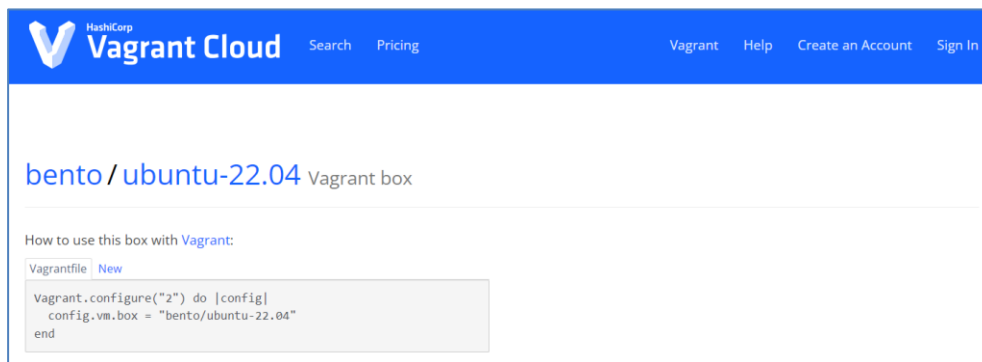
Grâce à Vagrant, vous pouvez facilement créer des environnements de développement virtuels à partir des images de VM et disposer de toutes les configurations de la VM dans un fichier appelé Vagrantfile.

Pour faire simple, vous définissez tout ce dont vous avez besoin dans la VM et Vagrant se charge de la création et la configuration de la VM.

Si quelqu'un a configuré un VagrantFile sur sa machine, il peut le partager avec n'importe qui afin de construire une VM similaire à celle qu'il a créée.

- **Vagrant Cloud (Boxes) :** <https://app.vagrantup.com>

Le vagrant Box est une image de machine virtuelle pré-cuite (Ubuntu, Centos, etc.). Ce format de boîte peut être partagé avec tous ceux qui utilisent Vagrant. Vous pouvez également créer une machine virtuelle personnalisée et ensuite l'empaqueter avec toutes les configurations et la partager avec les autres membres de l'équipe.



Installation de VirtualBox :

Avant l'installation de Vagrant, Vous devez avoir une boîte virtuelle installée. Vous pouvez télécharger Virtualbox en suivant les instructions suivantes :

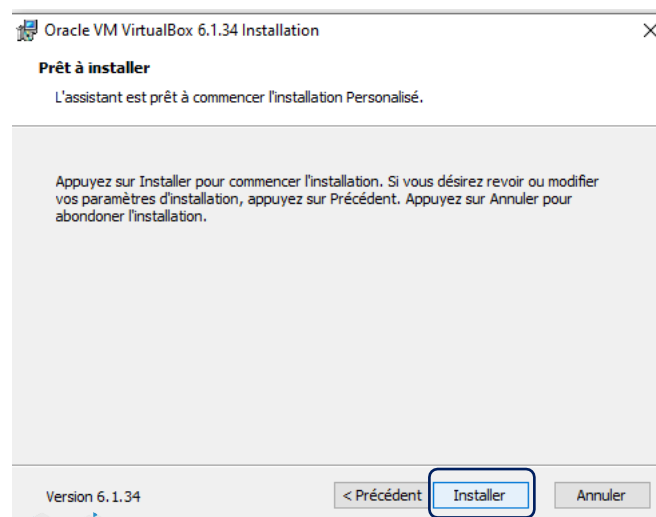
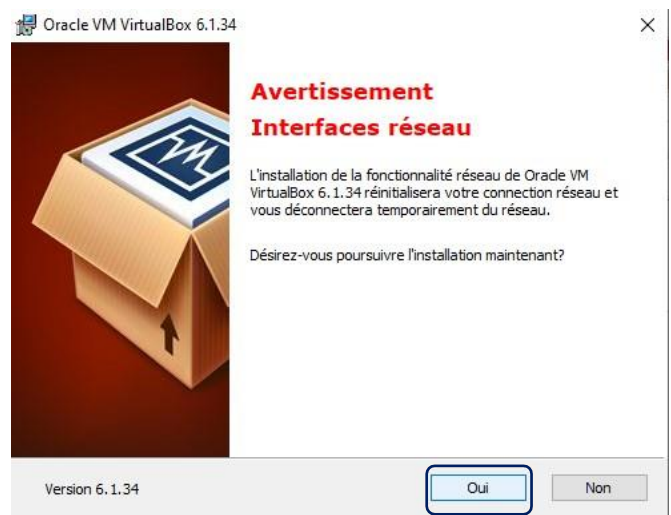
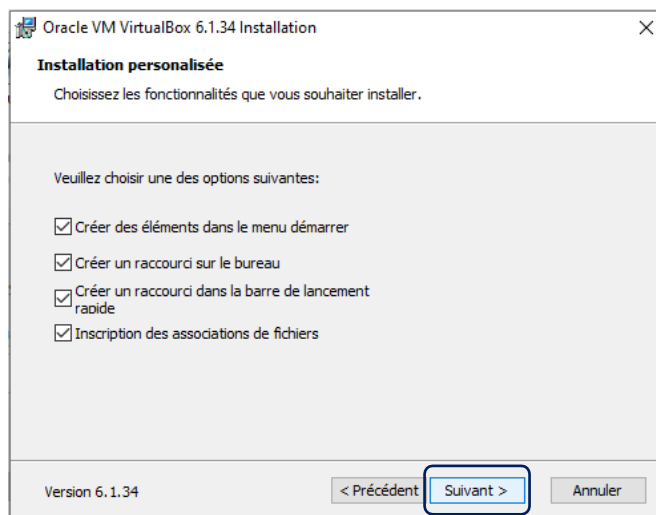
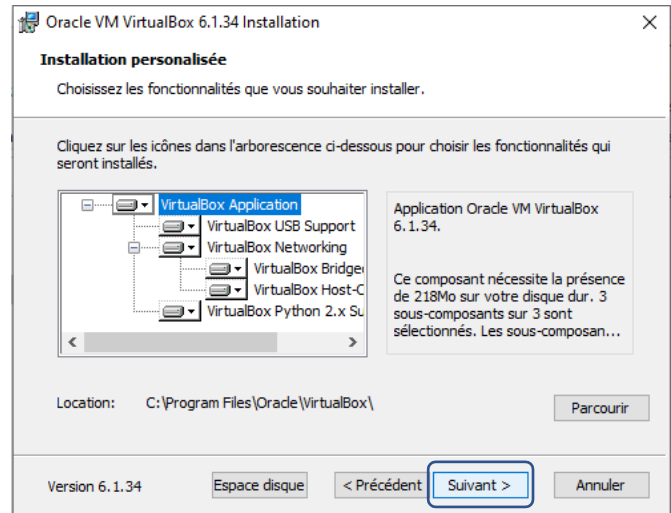
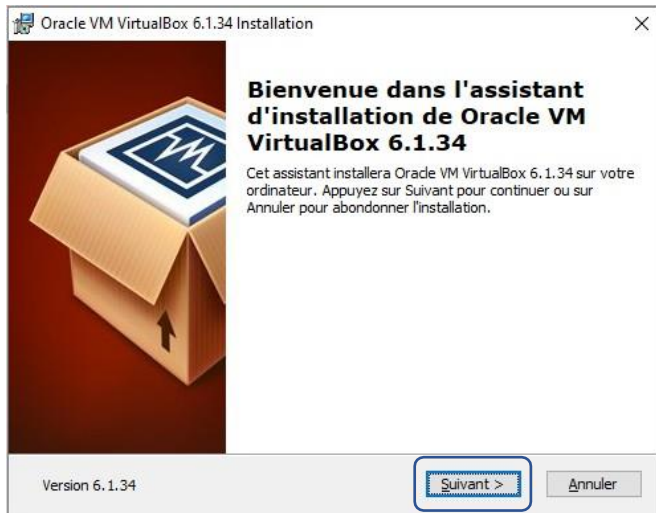
Accédez à ce lien : <https://www.virtualbox.org/wiki/Downloads>

Effectuez l'installation comme suit :



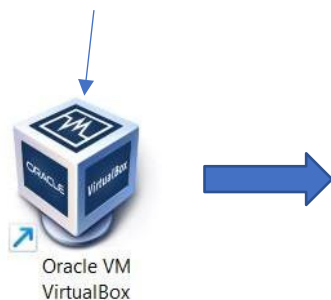


Double cliquer sur l'installation



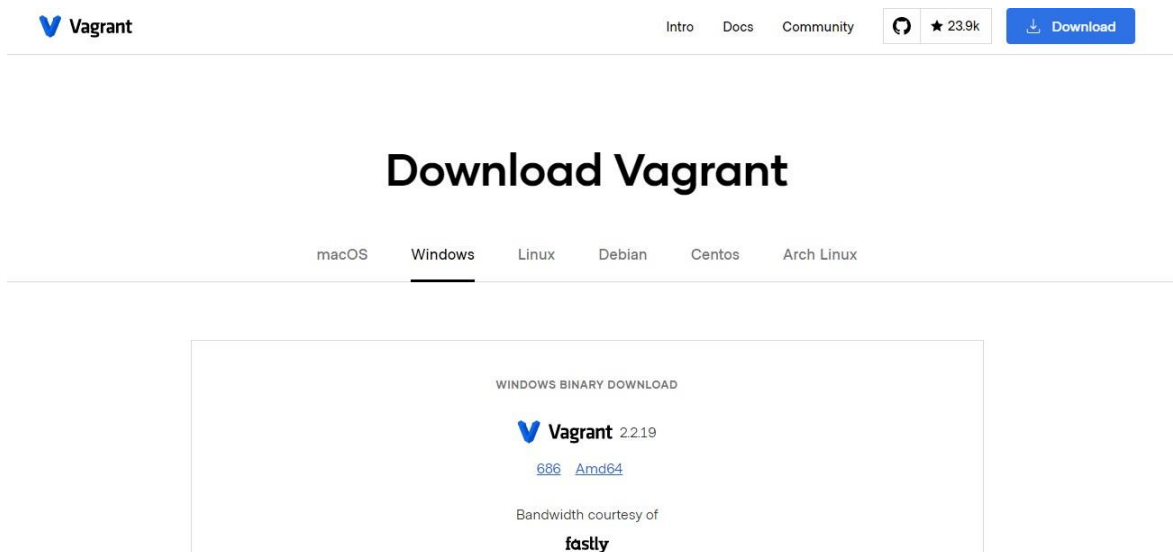
Le résultat final :

Double cliquer sur l'exécutable

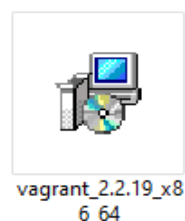


Installation de Vagrant :

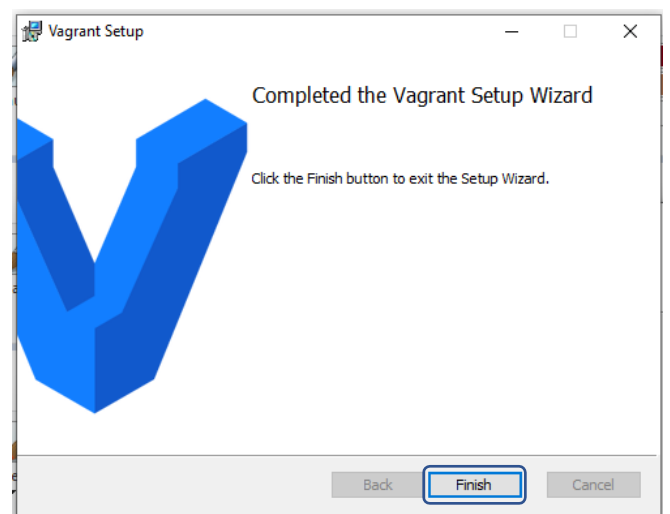
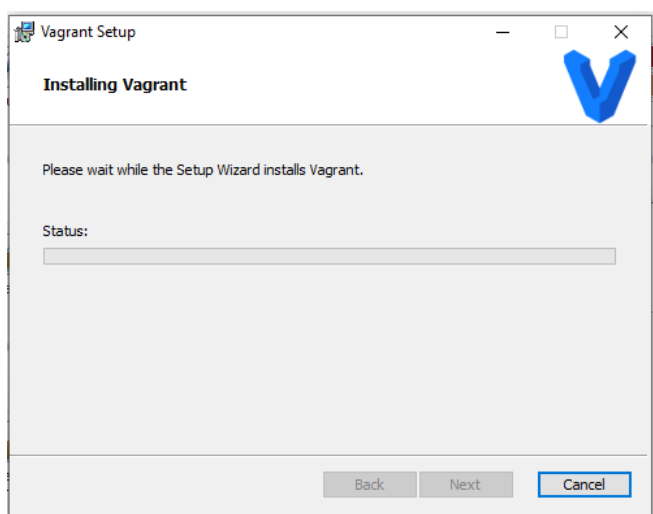
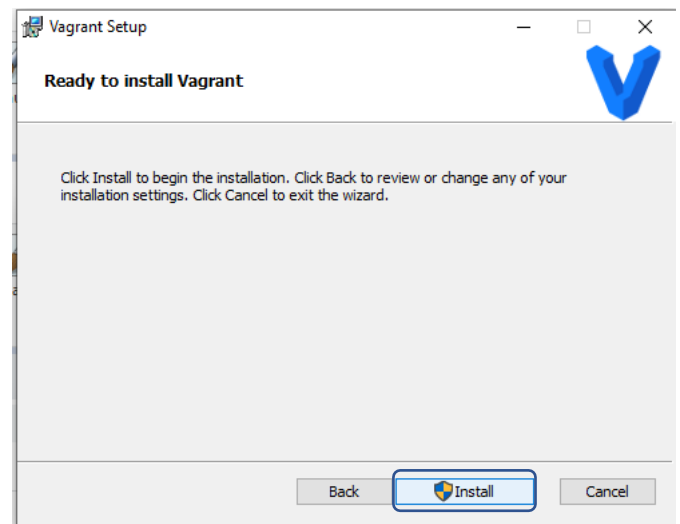
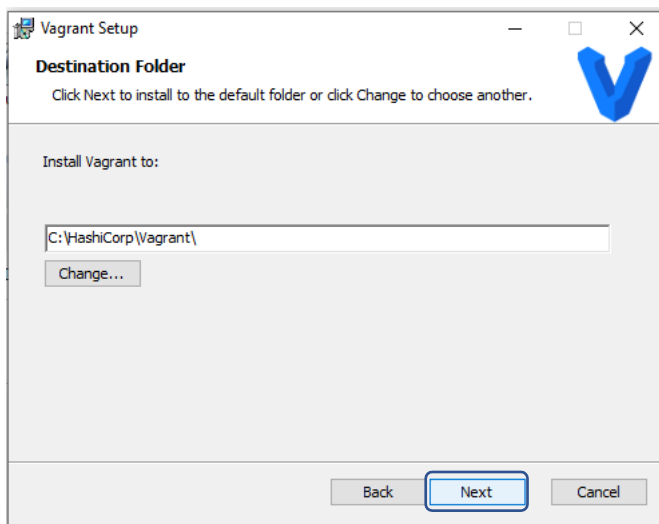
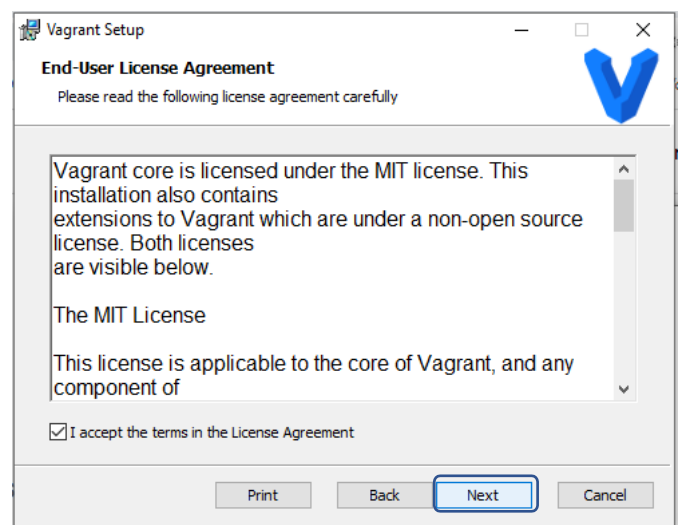
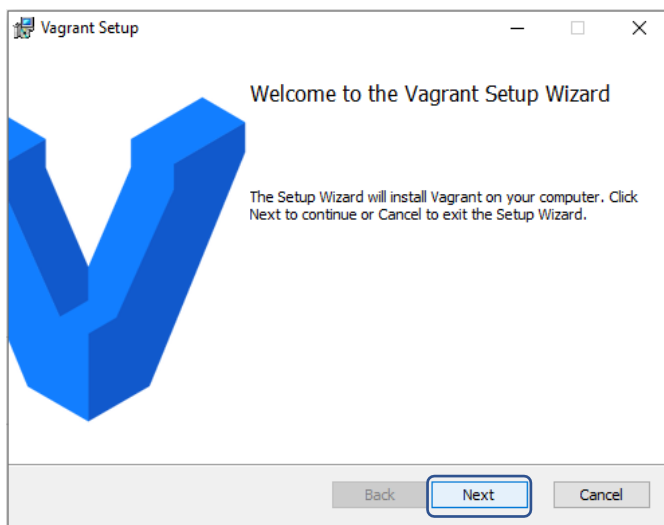
Passons maintenant à l'installation de vagrant. La première étape à faire est de télécharger l'installation en accédant à ce lien : <https://www.vagrantup.com/downloads>



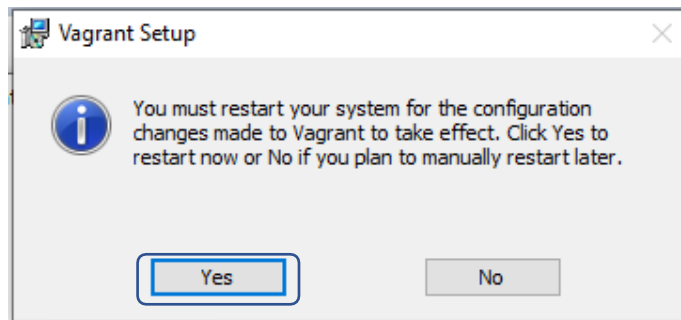
Choisissez la version selon le système d'exploitation que vous utilisez et procédez à l'installation en suivant les instructions suivantes :



Double cliquer sur l'installation



Redémarrez l'ordinateur après l'installation.



Pour vérifier que Vagrant est installé, ouvrez un PowerShell et exécutez la commande suivante :

vagrant --version

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

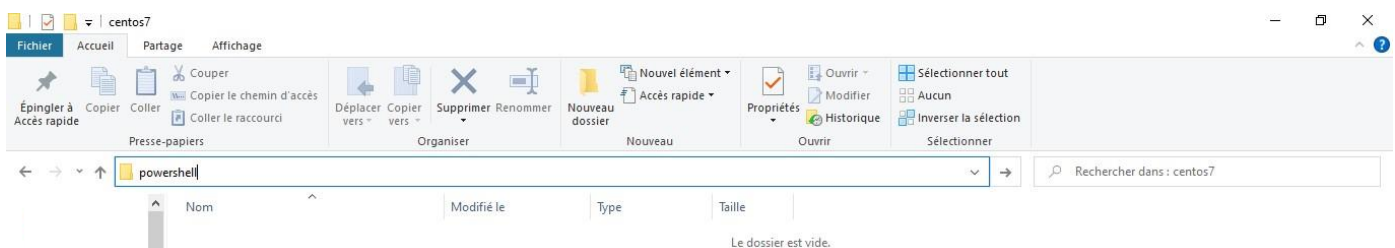
Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS C:\Users\Sirine> vagrant --version
Vagrant 2.2.19
```

Créer et configurer une machine virtuelle avec Vagrant :

1/ Par souci d'homogénéité, vous pouvez créer un dossier **D:\Vagrant**, dans lequel, vous allez trouver toutes les machines virtuelles dont vous aurez besoin par la suite.

2/ Créez un dossier nommé « **ubuntu** » et ouvrez un powershell.

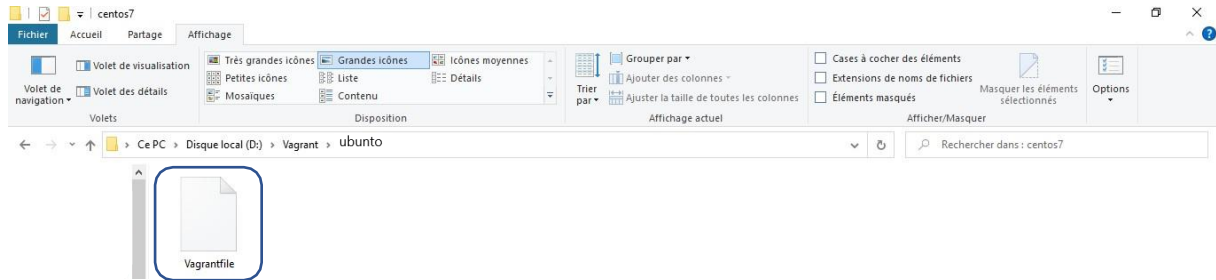


3/ Exécutez la commande **vagrant init** pour créer le fichier de configuration **Vagrantfile**

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Testez le nouveau système multiplateforme PowerShell https://aka.ms/pscore6

PS D:\Vagrant\ubuntu> vagrant init
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/ Ouvrez le fichier **Vagrantfile** où vous trouverez une configuration par défaut avec les explications des différents champs en commentaire.

```

1  # -*- mode: ruby -*-
2  # vi: set ft=ruby :
3
4  # All Vagrant configuration is done below. The "2" in Vagrant.configure
5  # configures the configuration version (we support older styles for
6  # backwards compatibility). Please don't change it unless you know what
7  # you're doing.
8  Vagrant.configure("2") do |config|
9    # The most common configuration options are documented and commented below.
10   # For a complete reference, please see the online documentation at
11   # https://docs.vagrantup.com.
12
13   # Every Vagrant development environment requires a box. You can search for
14   # boxes at https://vagrantcloud.com/search.
15   config.vm.box = "base"
16
17   # Disable automatic box update checking. If you disable this, then
18   # boxes will only be checked for updates when the user runs
19   # 'vagrant box outdated'. This is not recommended.
20   # config.vm.box_check_update = false
21
22   # Create a forwarded port mapping which allows access to a specific port
23   # within the machine from a port on the host machine. In the example below,
24   # accessing "localhost:8080" will access port 80 on the guest machine.
25   # NOTE: This will enable public access to the opened port
26   # config.vm.network "forwarded_port", guest: 80, host: 8080

```

Pour créer une machine virtuelle **Ubuntu** avec une configuration basique, vous pouvez utiliser le **Vagrantfile** ci-dessous :

```

Vagrantfile
Vagrant.configure("2") do |config|
  config.vm.box = "bento/ubuntu-22.04"
  config.vm.network "private_network", ip: "192.168.33.10"
  config.vm.provider "virtualbox" do |vb|
    vb.memory = "4000"
    vb.cpus = "2"
  end
end

```

Configurer la machine qui sera utilisée. Cette valeur doit être le nom d'une machine installée ou le nom abrégé d'une machine dans le « VagrantCloud ».

Configurer les réseaux sur la machine

Configurer le hardware de la machine (Mémoire, CPU, ...)

- **Vagrant provisioner**

Les provisionneurs dans Vagrant vous permettent d'installer automatiquement des logiciels, de modifier les configurations, ... sur la machine virtuelle en cours de création.

Voici un exemple d'utilisation :

```
Vagrant.configure("2") do |config|

  config.vm.box = "bento/ubuntu-22.04"

  config.vm.network "private_network", ip: "192.168.33.10"
  config.vm.network "public_network"
  config.vm.network "forwarded_port", guest: 80, host: 8080

  config.vm.provider "virtualbox" do |vb|
    vb.memory = "4000"
    vb.cpus = "2"
  end

  config.vm.provision "shell", inline: <<-SHELL
    sudo yum update
    sudo yum install gedit
  SHELL
end
```

NB :

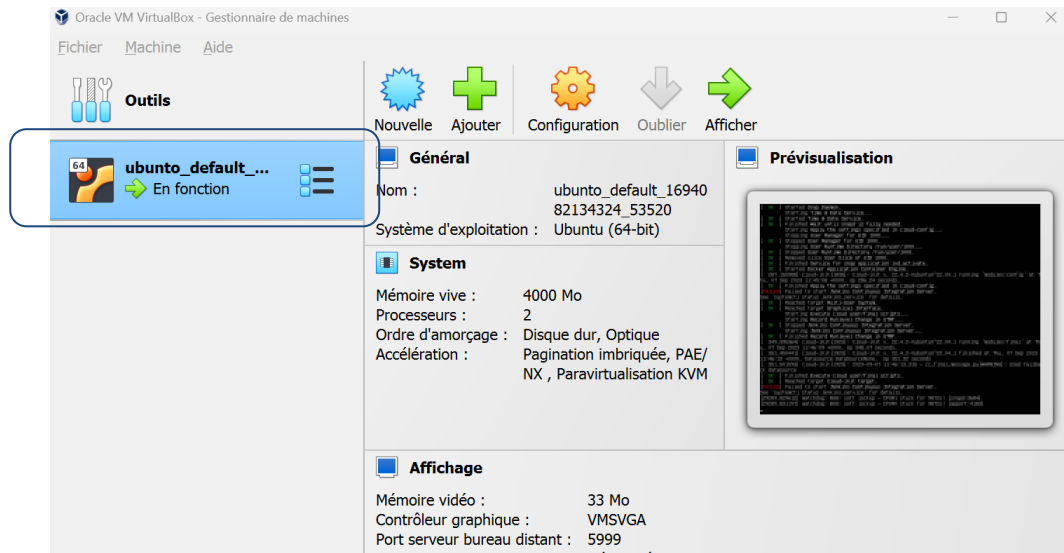
Après chaque modification du fichier **Vagrantfile**, vous devez exécuter la commande **vagrant reload**.

Si vous avez ajouté une partie de provisionnement dans un **Vagrantfile** d'une machine virtuelle déjà créée, vous devez exécuter la commande **vagrant reload --provision** pour forcer le provisionnement.

5/ Après, ouvrez VirtualBox et exécutez la commande **vagrant up** dans le powershell pour créer la machine.



```
vagrant@localhost:~
==> default: Destroying VM and associated drives...
PS D:\Vagrant\ubuntu> vagrant up
Bringing machine 'default' up with 'virtualbox' provider...
==> default: Importing base box 'centos/7'...
==> default: Matching MAC address for NAT networking...
==> default: Checking if box 'centos/7' version '2004.01' is up to date...
==> default: Setting the name of the VM: centos7_default_1656884160580_5914
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
default: Adapter 1: nat
default: Adapter 2: bridged
==> 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
```

Pour connaître l'état de la machine virtuelle, il suffit de taper dans le powershell la commande

vagrant status

```
PS D:\DevOpsVagrant\ubuntu> 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`.
PS D:\DevOpsVagrant\ubuntu>
```

6/ Pour accéder à la machine et créer un client vagrant, il suffit d'exécuter dans powerShell la commande **vagrant ssh**

```
vagrant@localhost:~
PS D:\Vagrant\ubuntu> vagrant ssh
[vagrant@localhost ~]$
```

Pour se connecter en tant que « root », le mot de passe par défaut est 'vagrant'. Si vous souhaitez le modifier, il suffit d'ajouter les instructions suivantes dans le fichier de configuration « Vagrantfile »

```
config.ssh.username = 'root'
config.ssh.password = 'vagrant'
```

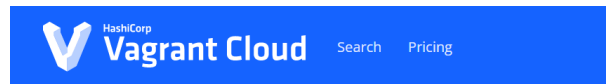
Pour consulter la liste des boîtes disponibles, vous devez exécuter la commande **vagrant box list**

```
PS C:\Vagrant\ubuntu> vagrant box list
bento/ubuntu-22.04 (virtualbox, 202303.13.0)
```

Utiliser une machine virtuelle (box) à partir du « Vagrant Cloud » :

Au lieu de construire une machine virtuelle à partir de zéro, Vagrant utilise une image de base pour cloner rapidement une machine virtuelle. Ces images de base sont appelées "**box**" dans Vagrant.

Vous pouvez installer un box sans créer un nouveau Vagrantfile. Pour cela, vous devez utiliser la commande **vagrant box add <nom de la machine>**.



[gusztavvargadr/windows-11](#) Vagrant box

How to use this box with Vagrant:

Vagrantfile | [New](#)

```
Vagrant.configure("2") do |config|
  config.vm.box = "gusztavvargadr/windows-11"
end
```

```
PS D:\DevOpsVagrant\windows 11> vagrant box add gusztavvargadr/windows-11
==> box: Loading metadata for box 'gusztavvargadr/windows-11'
box: URL: https://vagrantcloud.com/gusztavvargadr/windows-11
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) virtualbox
Enter your choice:
Invalid choice. Try again:
Invalid choice. Try again: 2
==> box: Adding box 'gusztavvargadr/windows-11' (v2102.0.2206) for provider: virtualbox
box: Downloading: https://vagrantcloud.com/gusztavvargadr/boxes/windows-11/versions/2102.0.2206/providers/virtualbox.box
==> box: Box download is resuming from prior download progress
box:
box: Calculating and comparing box checksum...
==> box: Successfully added box 'gusztavvargadr/windows-11' (v2102.0.2206) for 'virtualbox'!
```

```
PS D:\DevOpsVagrant\windows 11> vagrant box list
centos/7 (virtualbox, 2004.01)
gusztavvargadr/windows-11 (virtualbox, 2102.0.2206)
ubuntu/focal64 (virtualbox, 20220706.0.0)
```

Arrêter une machine virtuelle :

Vous pouvez exécuter la commande **vagrant halt** pour arrêter la machine virtuelle une fois votre travail terminé :

```
PS D:\DevOpsVagrant\ubuntu> vagrant halt
==> default: Attempting graceful shutdown of VM...
```

Si vous voulez arrêter la machine virtuelle tout en préservant son état, utilisez la commande **vagrant suspend**

```
PS D:\DevOpsVagrant\ubuntu> vagrant suspend
==> default: Saving VM state and suspending execution...
```

Redémarrer une machine virtuelle :

Si vous voulez redémarrer la machine virtuelle, utilisez la commande **vagrant reload**

```
PS D:\DevOpsVagrant\ubuntu> vagrant reload
==> default: Attempting graceful shutdown of VM...
==> default: Checking if box 'centos/7' version '2004.01' is up to date...
==> default: Clearing any previously set forwarded ports...
==> default: Fixed port collision for 22 => 2222. Now on port 2200.
==> default: Clearing any previously set network interfaces...
==> default: Preparing network interfaces based on configuration...
```

Supprimer une machine virtuelle :

Si vous voulez détruire la machine virtuelle, utilisez la commande **vagrant destroy**

```
PS D:\Vagrant\ubuntu> vagrant destroy
default: Are you sure you want to destroy the 'default' VM? [y/N] y
==> default: Forcing shutdown of VM...
==> default: Destroying VM and associated drives...
```

Vous pouvez aussi utiliser la commande **vagrant box remove <Nom de la machine>**

```
PS D:\DevOpsVagrant\ubuntu> vagrant box remove bento/ubuntu-22.04
Box 'bento/ubuntu-22.04 (v202303.13.0)' with provider 'virtualbox' appears
to still be in use by at least one Vagrant environment. Removing
the box could corrupt the environment. We recommend destroying
these environments first:

default (ID: 6502ef464cab49b1866cd4eae8197d50)
Are you sure you want to remove this box? [y/N]
```

Conclusion :

Vagrant est un outil puissant que de nombreux spécialistes utilisent. Le fait qu'il ne dispose pas de grandes alternatives accroît encore son importance. Ce workshop a fourni un guide détaillé sur le processus d'installation de vagrant. Il montre également les étapes que nous pouvons utiliser pour créer des machines virtuelles sur Ubuntu.