AnsibleVagrant

Vagrant + VirtualBox+ Ansible

Architecture

Introduction

Dans ce document, nous allons détailler les étapes nécessaires pour la mise en place d'un environnement de virtualisation et d'automatisation de configuration en utilisant **VirtualBox, Vagrant et Ansible**.

- **VirtualBox** : Permet de créer et gérer des machines virtuelles sur un environnement local.
- **Vagrant** : Simplifie la gestion des machines virtuelles en automatisant leur création et configuration via un simple fichier de configuration (**Vagrantfile**).
- **Ansible** : Outil d'automatisation qui facilite le déploiement, la gestion de la configuration et l'orchestration des serveurs sans nécessiter d'agent sur les machines cibles.

Ce projet vise à **provisionner plusieurs machines virtuelles**, à les configurer automatiquement et à y installer des services spécifiques grâce à Ansible.

Rôles des composants du projet :

- **Contrôleur Ansible** : Machine dédiée à l'exécution des playbooks Ansible pour gérer les autres machines.
- **Machines hôtes (Hosts)** : Machines virtuelles sur lesquelles Ansible va exécuter des tâches (ex. : installation de services comme Nginx).

- **Vagrant** : Outil utilisé pour automatiser la création et la gestion des machines virtuelles.
- **VirtualBox** : Hyperviseur utilisé pour exécuter les machines virtuelles.

Ce guide détaillera toutes les étapes, depuis l'installation des outils jusqu'à l'exécution des playbooks Ansible, afin d'automatiser complètement la configuration de notre infrastructure

---

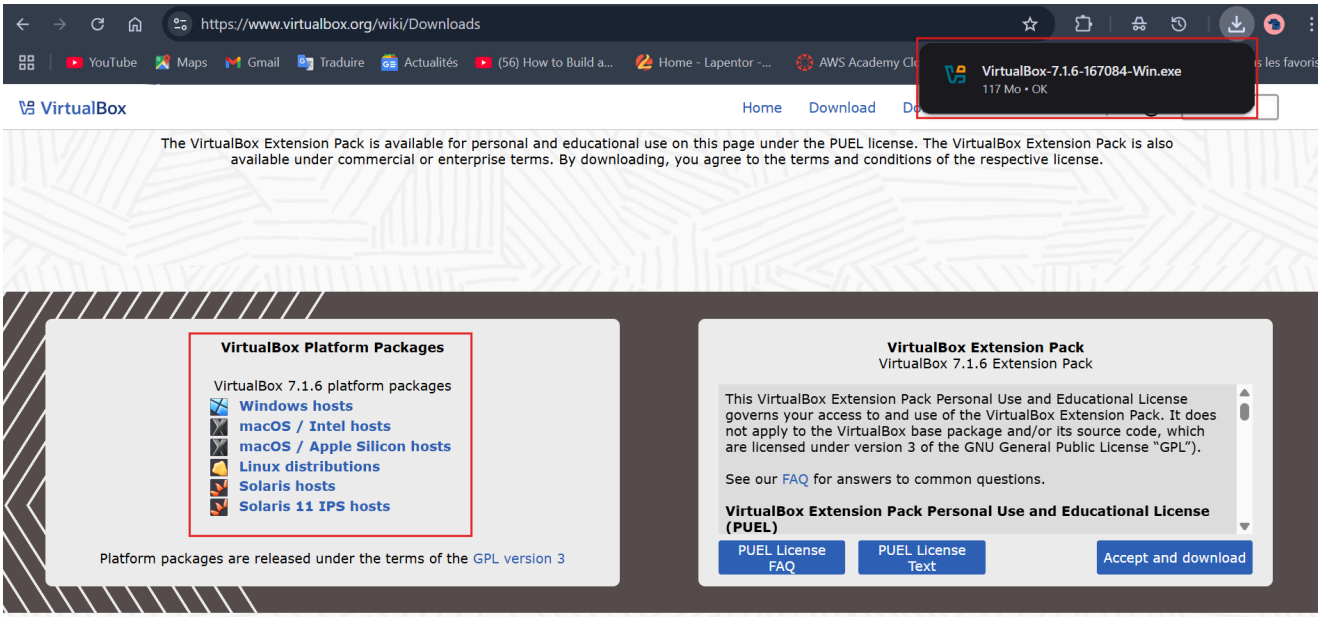
1. Prérequis

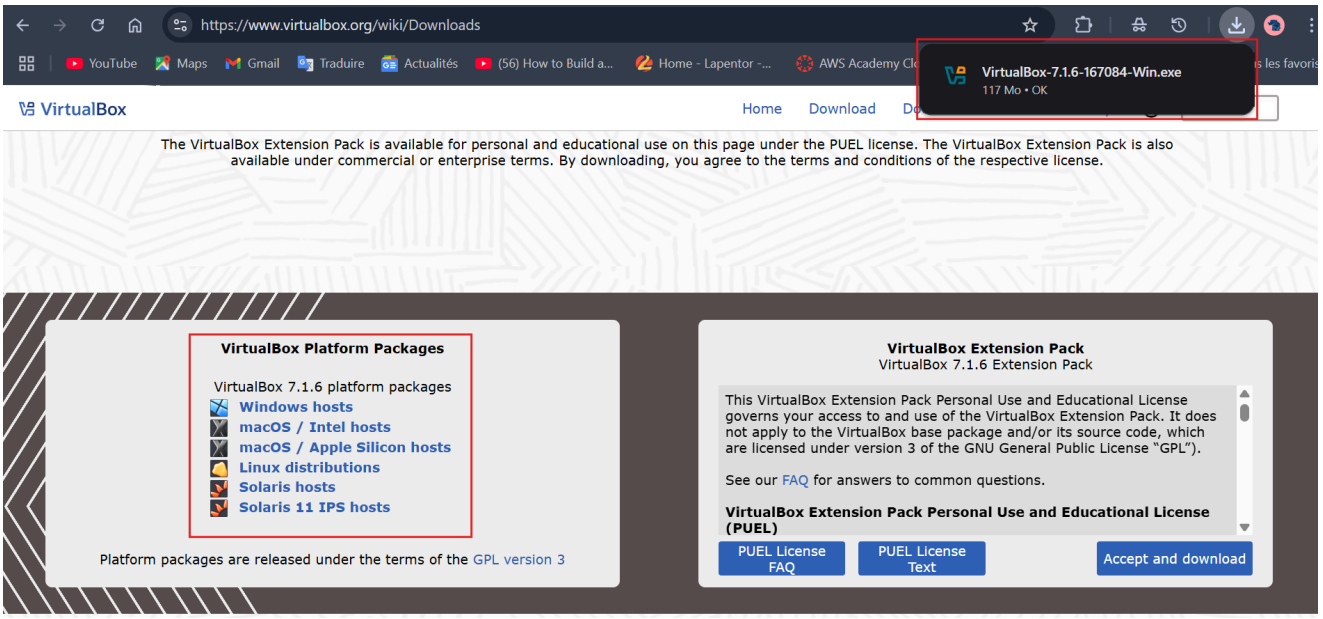
Avant de commencer, assurez-vous d'avoir installé les outils suivants :

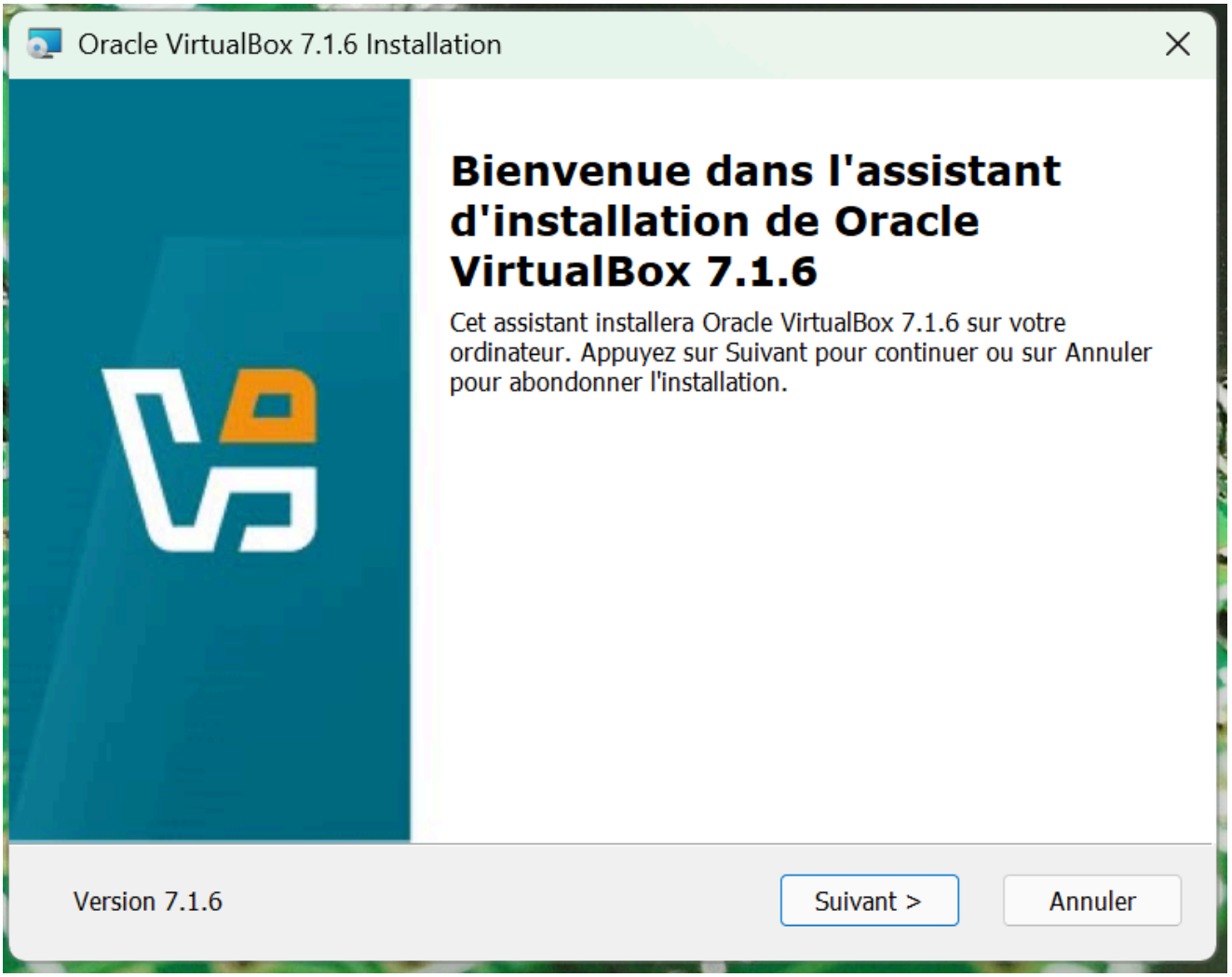
- VirtualBox
- Vagrant
- Ansible

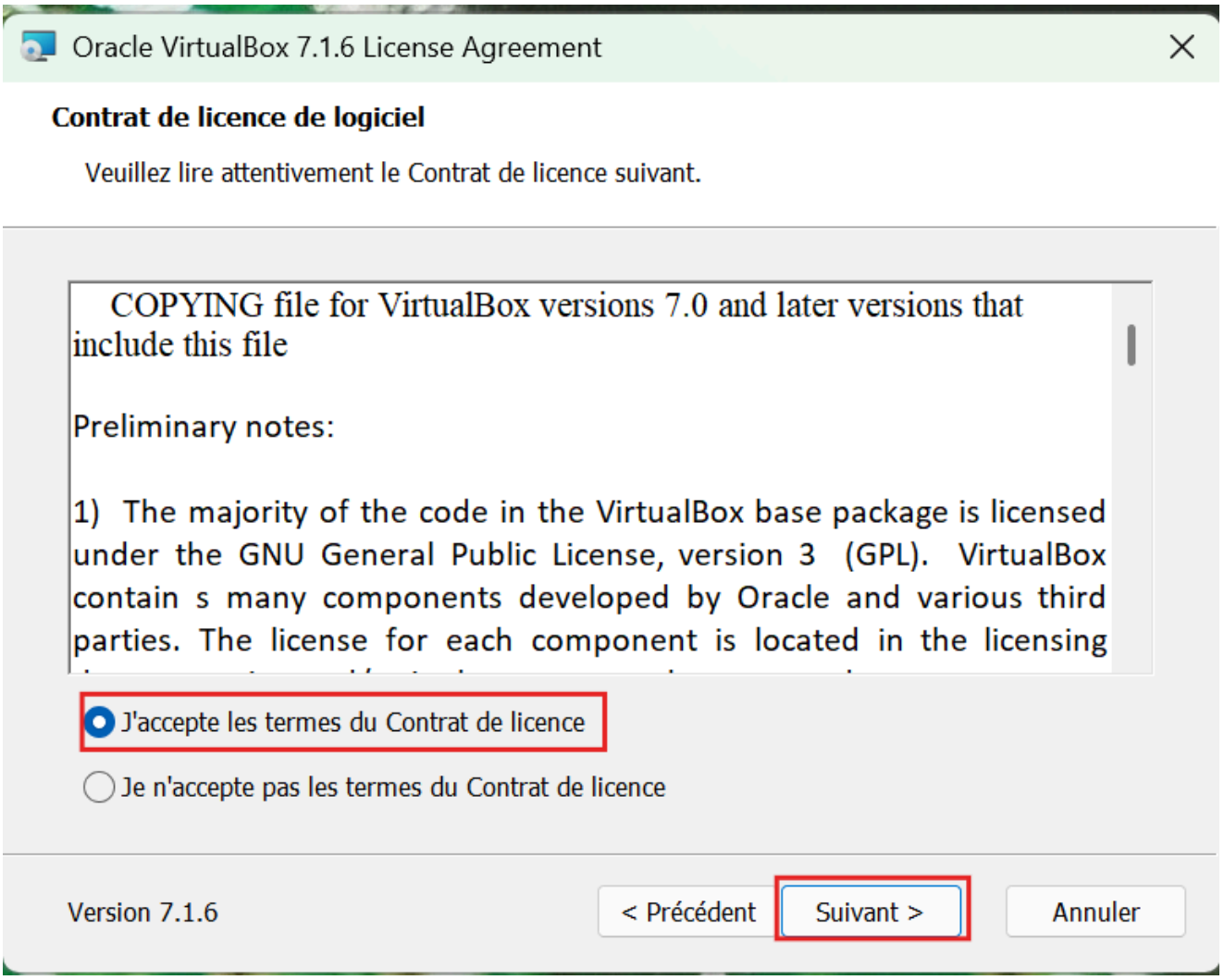
Lien de téléchargement de VirtualBox :

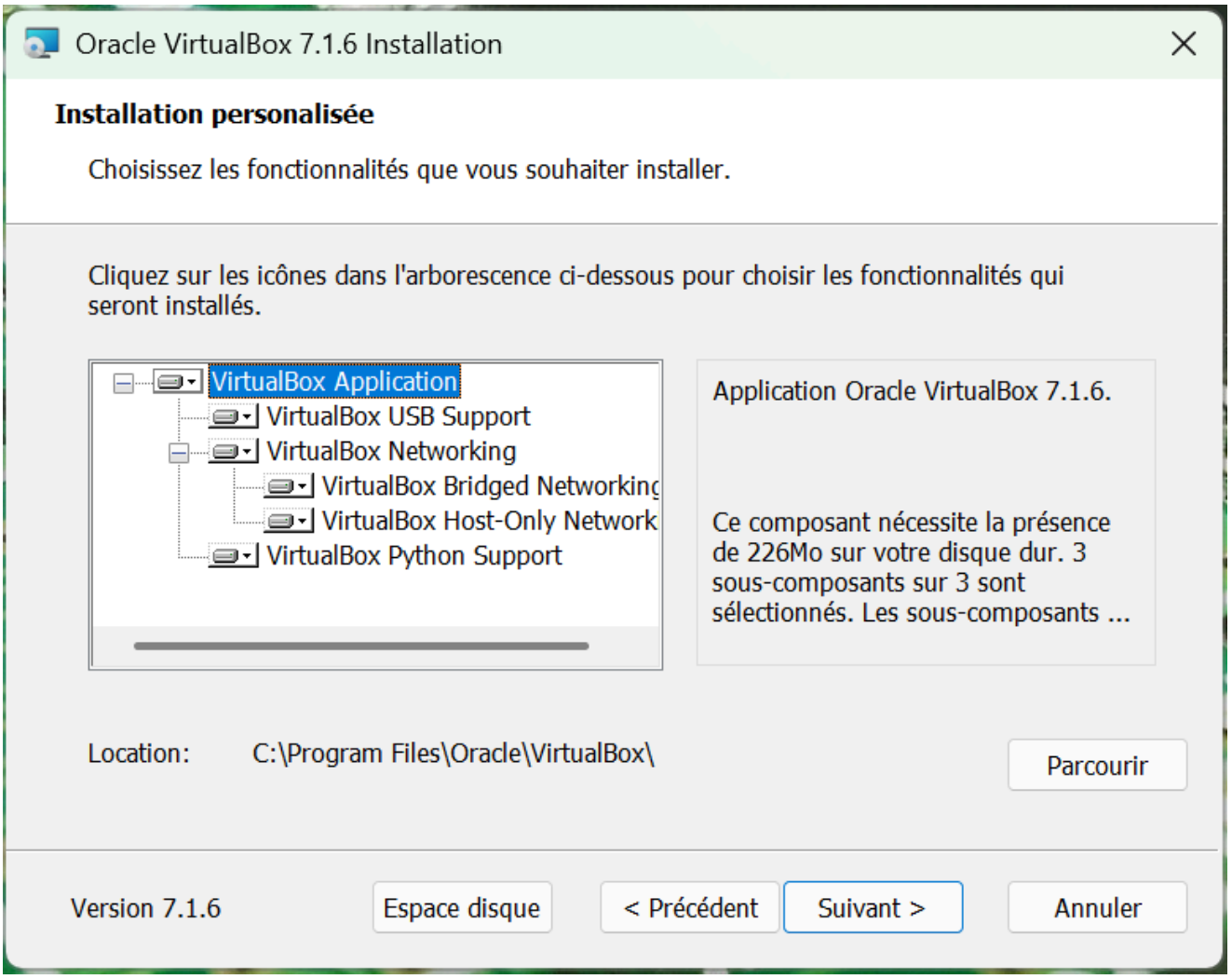
https://www.virtualbox.org/



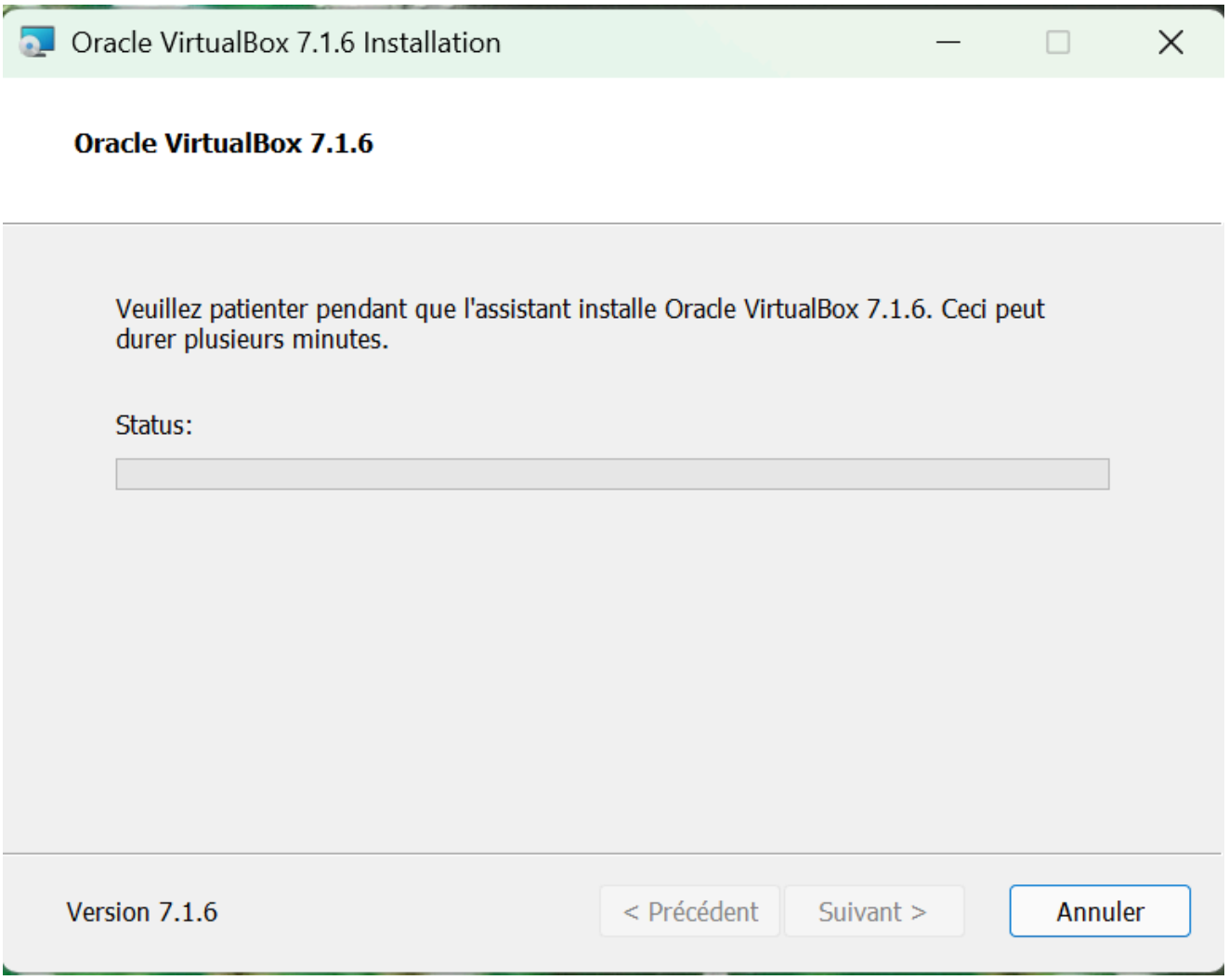

https://www.virtualbox.org/wiki/Downloads
YouTube
Maps
Gmail
Traduire
Actualités
(56) How to Build a...
Home - Lapentor -...
AWS Academy Cl
VirtualBox-7.1.6-167084-Win.exe
117 Mo • OK
VirtualBox
Home
Download
The VirtualBox Extension Pack is available for personal and educational use on this page under the PUEL license. The VirtualBox Extension Pack is also available under commercial or enterprise terms. By downloading, you agree to the terms and conditions of the respective license.
VirtualBox Platform Packages
VirtualBox 7.1.6 platform packages
Windows hosts
macOS / Intel hosts
macOS / Apple Silicon hosts
Linux distributions
Solaris hosts
Solaris 11 IPS hosts
Platform packages are released under the terms of the GPL version 3
VirtualBox Extension Pack
VirtualBox 7.1.6 Extension Pack
This VirtualBox Extension Pack Personal Use and Educational License governs your access to and use of the VirtualBox Extension Pack. It does not apply to the VirtualBox base package and/or its source code, which are licensed under version 3 of the GNU General Public License "GPL").
See our FAQ for answers to common questions.
VirtualBox Extension Pack Personal Use and Educational License (PUEL)
PUEL License FAQ
PUEL License Text
Accept and download


Oracle VirtualBox 7.1.6 Installation
Bienvenue dans l'assistant d'installation de Oracle VirtualBox 7.1.6
Cet assistant installera Oracle VirtualBox 7.1.6 sur votre ordinateur. Appuyez sur Suivant pour continuer ou sur Annuler pour abondonner l'installation.
Version 7.1.6
Suivant >
Annuler


Oracle VirtualBox 7.1.6 License Agreement
Contrat de licence de logiciel
Veuillez lire attentivement le Contrat de licence suivant.
COPYING file for VirtualBox versions 7.0 and later versions that include this file
Preliminary notes:
1) The majority of the code in the VirtualBox base package is licensed under the GNU General Public License, version 3 (GPL). VirtualBox contain s many components developed by Oracle and various third parties. The license for each component is located in the licensing
J'accepte les termes du Contrat de licence
Je n'accepte pas les termes du Contrat de licence
Version 7.1.6
< Précédent
Suivant >
Annuler

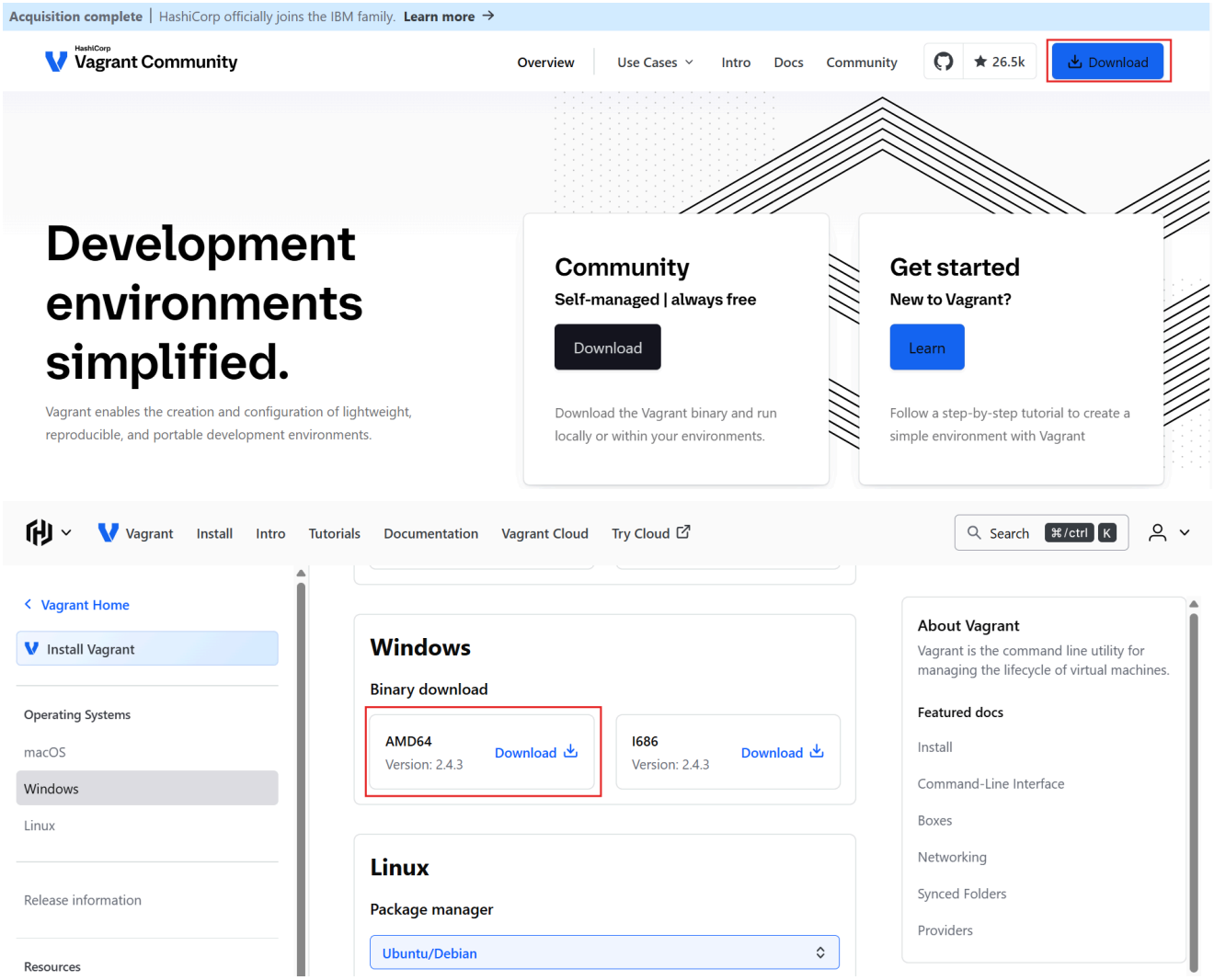


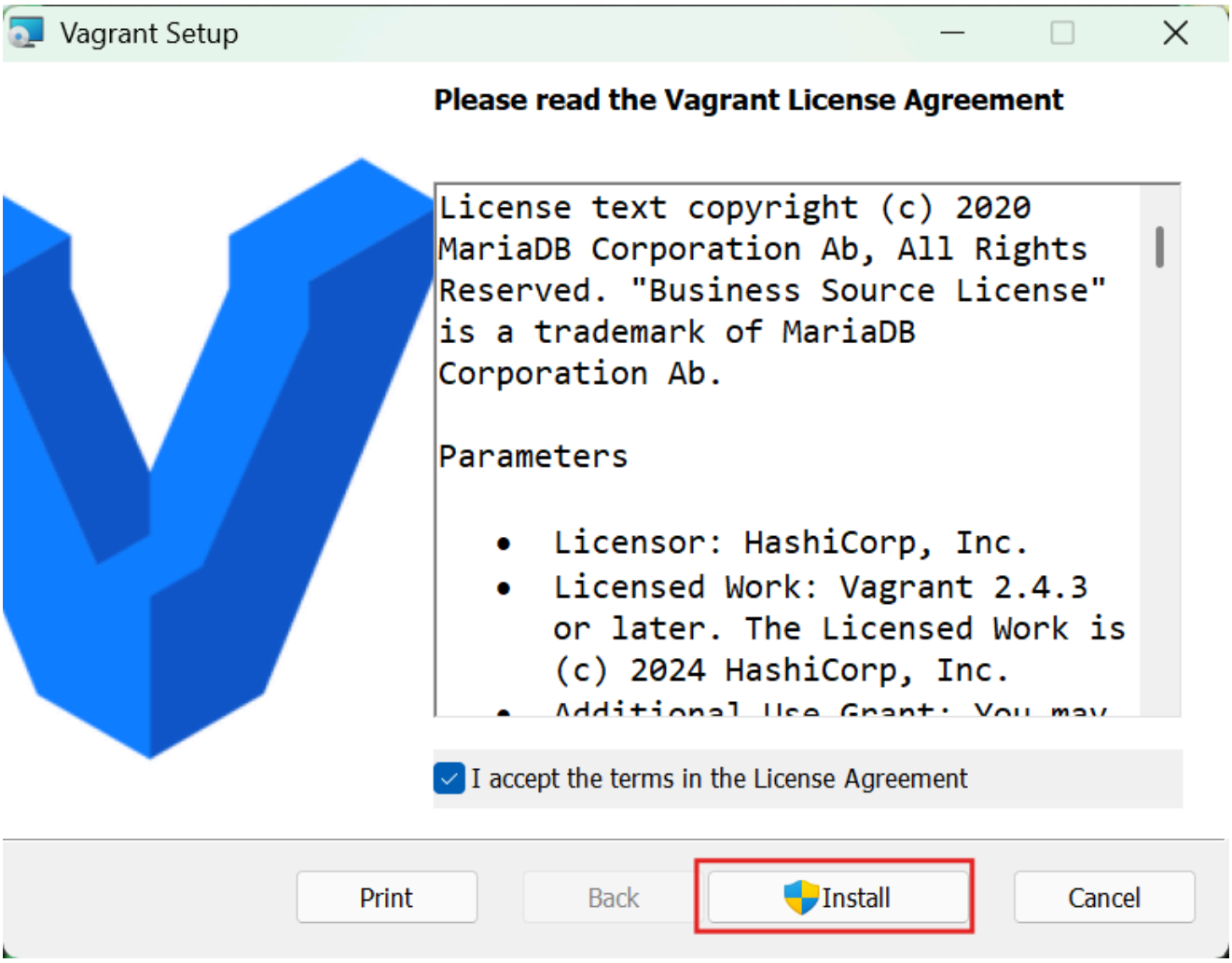
Ensuite, passez aux étapes suivantes.

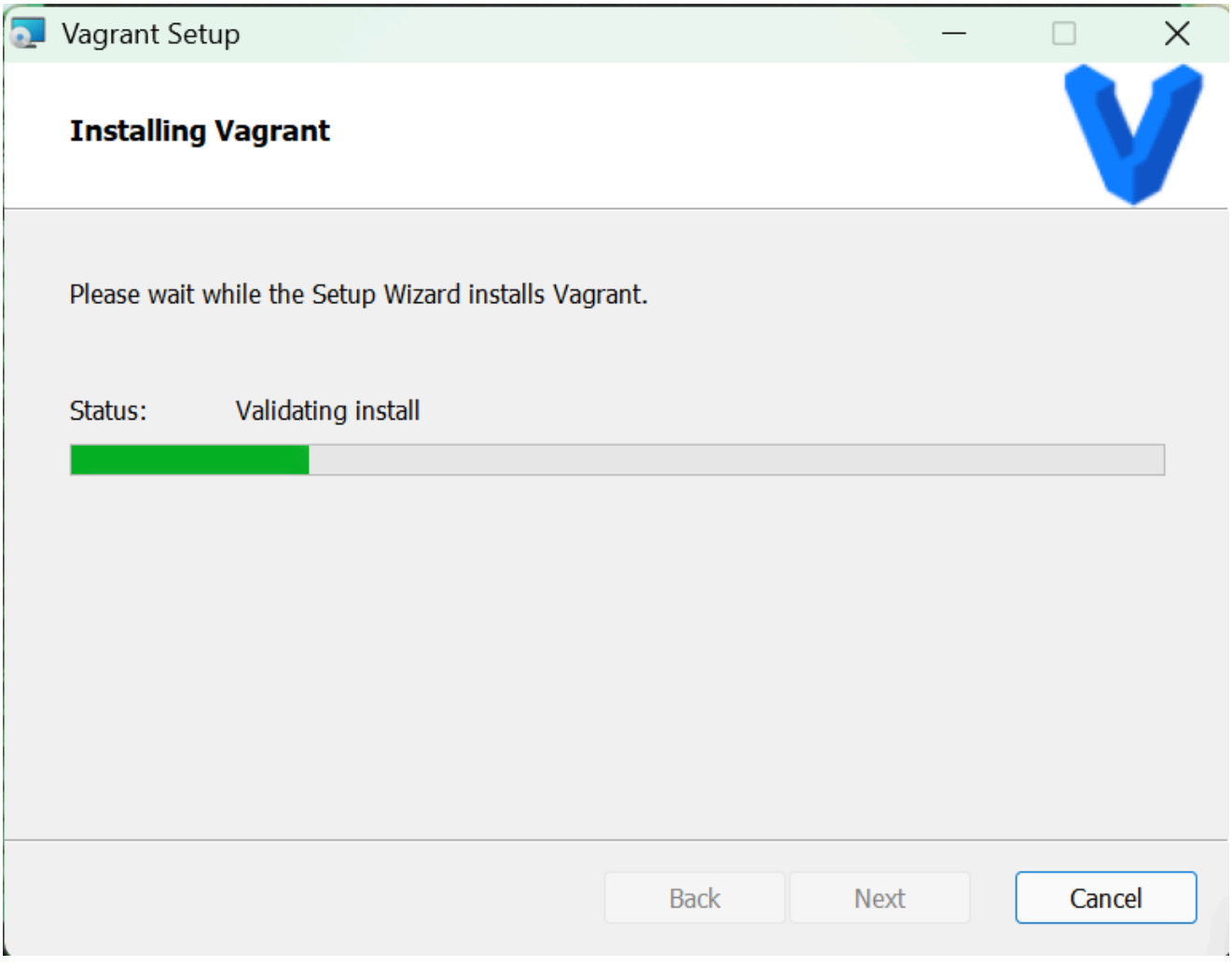


Lien de téléchargement de Vagrant :

https://www.vagrantup.com/


Acquisition complete | HashiCorp officially joins the IBM family. Learn more →
HashiCorp
Vagrant Community
Overview
Use Cases
Intro
Docs
Community
★ 26.5k
Download
Development environments simplified.
Vagrant enables the creation and configuration of lightweight, reproducible, and portable development environments.
Community
Self-managed | always free
Download
Download the Vagrant binary and run locally or within your environments.
Get started
New to Vagrant?
Learn
Follow a step-by-step tutorial to create a simple environment with Vagrant
Vagrant
Install
Intro
Tutorials
Documentation
Vagrant Cloud
Try Cloud
Search ⌘/ctrl K
Vagrant Home
Install Vagrant
Operating Systems
macOS
Windows
Linux
Release information
Resources
Windows
Binary download
AMD64
Version: 2.4.3
Download
I686
Version: 2.4.3
Download
Linux
Package manager
Ubuntu/Debian
About Vagrant
Vagrant is the command line utility for managing the lifecycle of virtual machines.
Featured docs
Install
Command-Line Interface
Boxes
Networking
Synced Folders
Providers


Vagrant Setup
Please read the Vagrant License Agreement
License text copyright (c) 2020 MariaDB Corporation Ab, All Rights Reserved. "Business Source License" is a trademark of MariaDB Corporation Ab.
Parameters
Licensor: HashiCorp, Inc.
Licensed Work: Vagrant 2.4.3 or later. The Licensed Work is (c) 2024 HashiCorp, Inc.
I accept the terms in the License Agreement
Print
Back
Install
Cancel


Vagrant Setup
Installing Vagrant
Please wait while the Setup Wizard installs Vagrant.
Status:
Validating install
Back
Next
Cancel

Vérifier son installation

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant>vagrant --version
Vagrant 2.4.3
```

---

2. Configuration des VM Hosts

Création des répertoires nécessaires

```
mkdir AnsibleWork
```

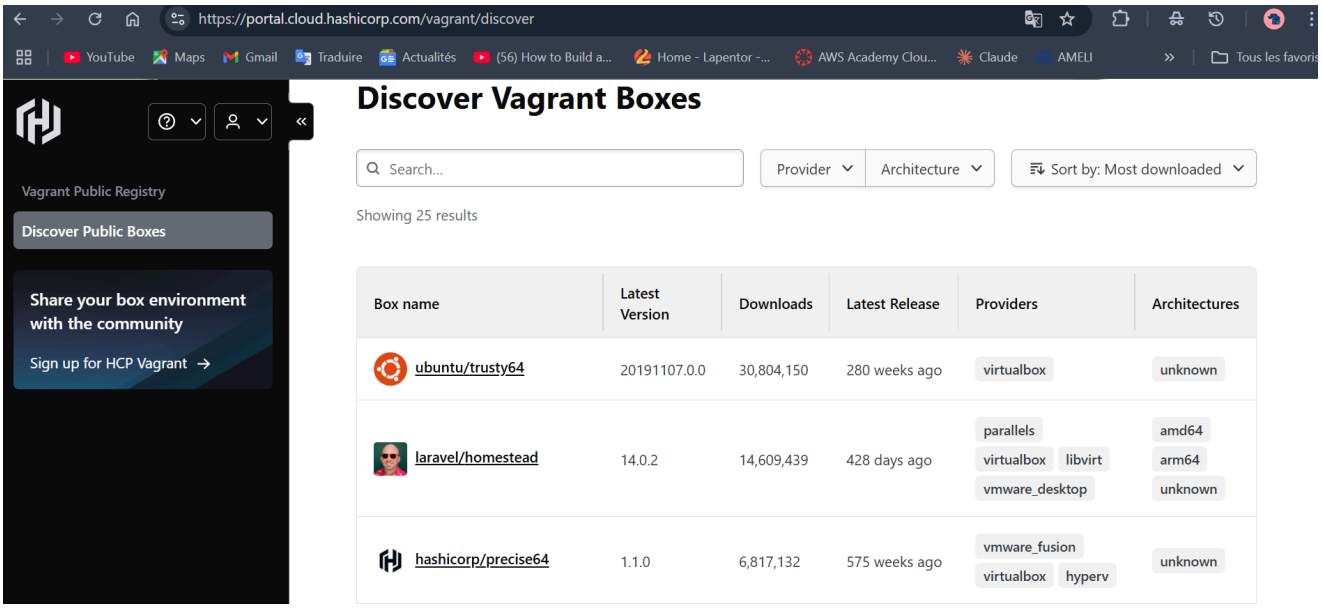
```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant>mkdir AnsibleWork

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant>cd AnsibleWork

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork>
```

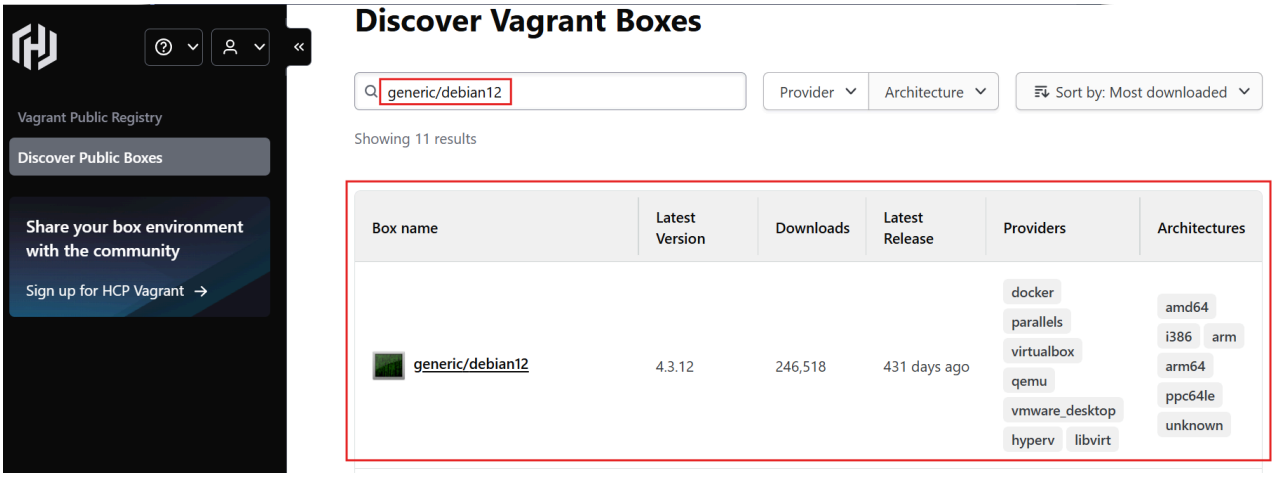
```
mkdir ansible-hosts
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork>mkdir ansible-hosts

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork>cd ansible-hosts

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>
```

Sélection de l’image pour nos VM

Nous utiliserons **HashiCorp Cloud Platform** et choisirons le module **generic/debian12**.





Initialisation de la machine Vagrant

```
vagrant init generic/debian12
```

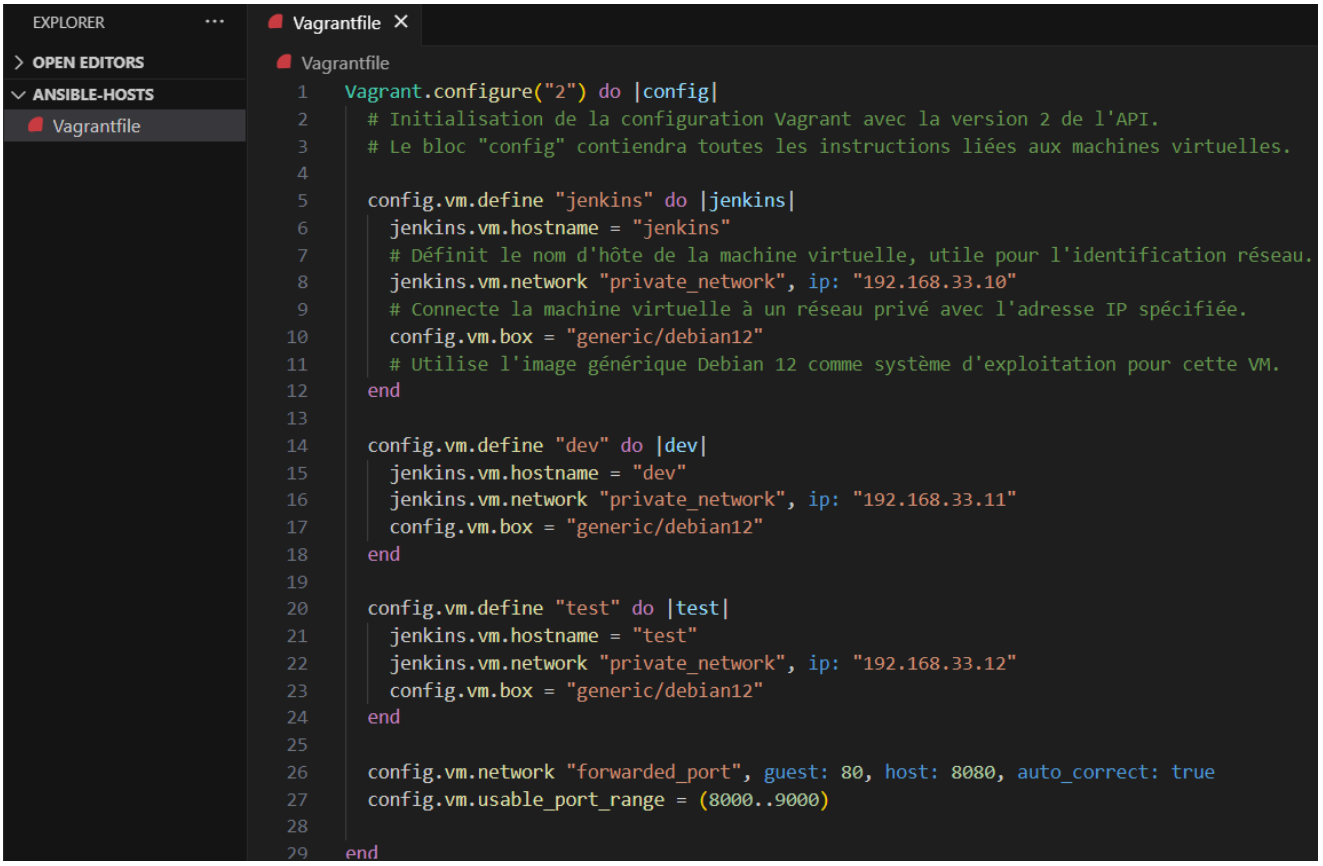
```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>vagrant init generic/debian12
A `Vagrantfile` has been placed in this directory. You are now
ready to `vagrant up` your first virtual environment! Please read
the comments in the Vagrantfile as well as documentation on
`vagrantup.com` for more information on using Vagrant.
```

Vérification du Vagrantfile

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>dir
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 8456-C647

 Répertoire de C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts

23/03/2025  02:21    <DIR>          .
23/03/2025  01:52    <DIR>          ..
23/03/2025  02:21             3 467 Vagrantfile
               1 fichier(s)              3 467 octets
               2 Rép(s)  87 423 791 104 octets libres
```

Voici la configuration du **Vagrantfile** :

```
Vagrant.configure("2") do |config|
  config.vm.define "jenkins" do |jenkins|
    jenkins.vm.hostname = "jenkins"
    jenkins.vm.network "private_network", ip: "192.168.33.10"
    config.vm.box = "generic/debian12"
  end

  config.vm.define "dev" do |dev|
    dev.vm.hostname = "dev"
    dev.vm.network "private_network", ip: "192.168.33.11"
    config.vm.box = "generic/debian12"
  end

  config.vm.define "test" do |test|
    test.vm.hostname = "test"
    test.vm.network "private_network", ip: "192.168.33.12"
    config.vm.box = "generic/debian12"
  end

  config.vm.network "forwarded_port", guest: 80, host: 8080, auto_correct:
true
end
```



Validation et démarrage des VM

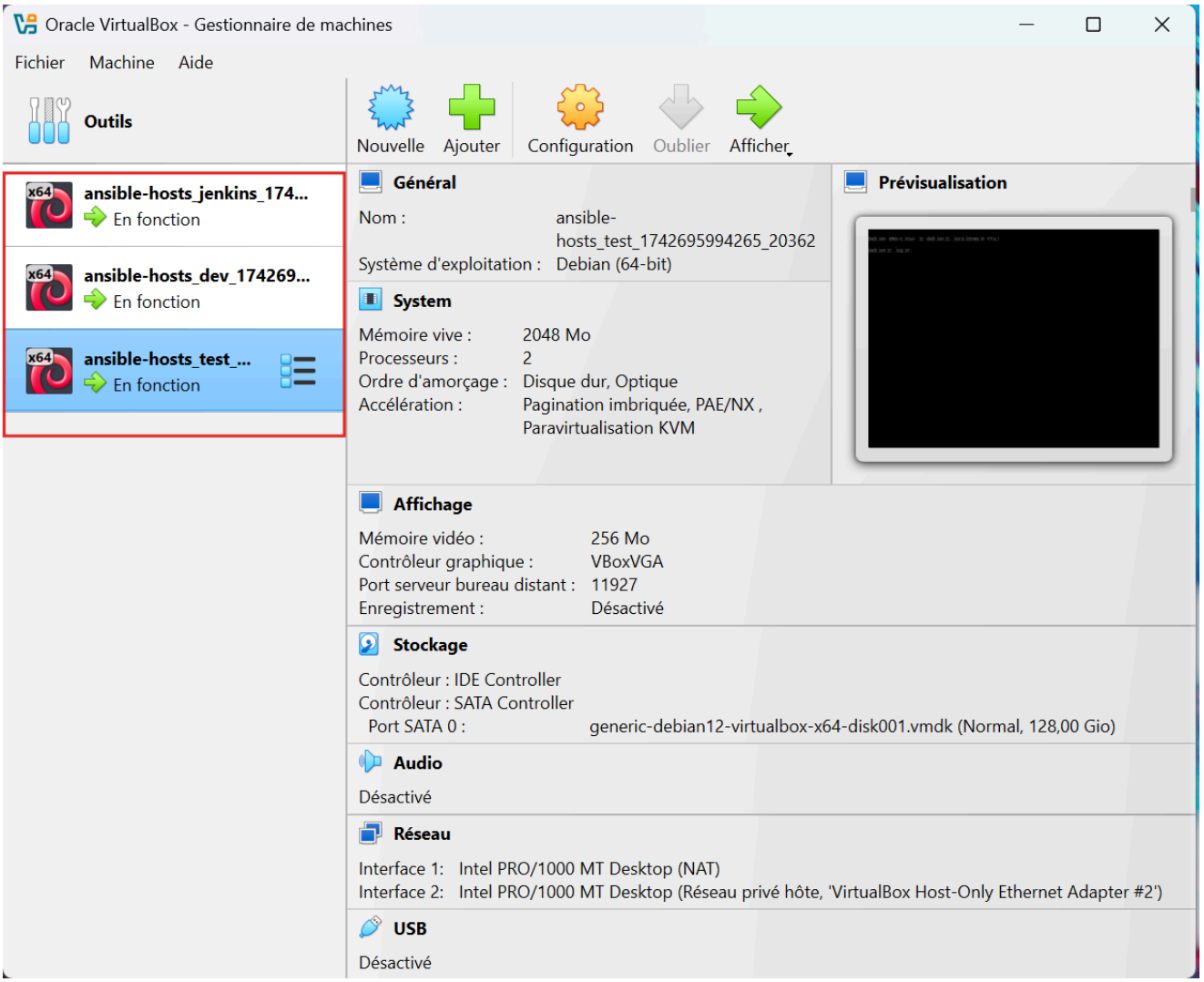
```
vagrant validate
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>vagrant validate
Vagrantfile validated successfully.
```

```
vagrant up
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>vagrant up
Bringing machine 'jenkins' up with 'virtualbox' provider...
Bringing machine 'dev' up with 'virtualbox' provider...
Bringing machine 'test' up with 'virtualbox' provider...
==> jenkins: Box 'generic/debian12' could not be found. Attempting to find and install...
    jenkins: Box Provider: virtualbox
    jenkins: Box Version: >= 0
==> jenkins: Loading metadata for box 'generic/debian12'
    jenkins: URL: https://vagrantcloud.com/api/v2/vagrant/generic/debian12
==> jenkins: Adding box 'generic/debian12' (v4.3.12) for provider: virtualbox (amd64)
    jenkins: Downloading: https://vagrantcloud.com/generic/boxes/debian12/versions/4.3.12/providers/virtualbox/amd64/vagrant.box
    jenkins:
    jenkins: Calculating and comparing box checksum...
==> jenkins: Successfully added box 'generic/debian12' (v4.3.12) for 'virtualbox (amd64)'!
==> jenkins: Importing base box 'generic/debian12'...
==> jenkins: Matching MAC address for NAT networking...
==> jenkins: Checking if box 'generic/debian12' version '4.3.12' is up to date...
==> jenkins: Setting the name of the VM: ansible-hosts_jenkins_1742695713660_91358
==> jenkins: Fixed port collision for 80 => 8080. Now on port 8000.
==> jenkins: Clearing any previously set network interfaces...
==> jenkins: Preparing network interfaces based on configuration...
    jenkins: Adapter 1: nat
    jenkins: Adapter 2: hostonly
==> jenkins: Forwarding ports...
    jenkins: 80 (guest) => 8000 (host) (adapter 1)
    jenkins: 22 (guest) => 2222 (host) (adapter 1)
==> jenkins: Running 'pre-boot' VM customizations...
==> jenkins: Booting VM...
==> jenkins: Waiting for machine to boot. This may take a few minutes...
    jenkins: SSH address: 127.0.0.1:2222
    jenkins: SSH username: vagrant
```

vérification des machines des VirtualBox



Vérification du statut des VM

```
vagrant status
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>vagrant status
Current machine states:

jenkins                   running (virtualbox)
dev                       running (virtualbox)
test                      running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
```

Connexion à une VM spécifique

```
ssh vagrant@192.168.33.11
```

Le mot de passe est **vagrant**.

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-hosts>ssh vagrant@192.168.33.11
vagrant@192.168.33.11's password:
Last login: Sun Mar 23 02:26:06 2025 from 10.0.2.2
vagrant@dev:~$
```

```
vagrant@dev:~$ exit
logout
Connection to 192.168.33.11 closed.
```

---

3. Configuration du Controller Ansible

Création et initialisation de la VM Controller

```
mkdir ansible-controller
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork>mkdir ansible-controller

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork>cd ansible-controller
```

```
vagrant init generic/debian12
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>vagrant init generic/debian12
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.

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>dir
 Le volume dans le lecteur C n'a pas de nom.
 Le numéro de série du volume est 8456-C647

 Répertoire de C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller

23/03/2025  03:47    <DIR>          .
23/03/2025  03:47    <DIR>          ..
23/03/2025  03:47             3 467 Vagrantfile
               1 fichier(s)            3 467 octets
               2 Rép(s)  75 477 639 168 octets libres

C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>
```

Configuration du **Vagrantfile**

```
Vagrant.configure("2") do |config|
  config.vm.define "ansible-controller" do |controller|
    controller.vm.hostname = "controller"
    controller.vm.box = "generic/debian12"
  end

  config.vm.provision "shell", inline: <<-SHELL
    sudo apt-get update
    sudo apt-get install -y software-properties-common
    apt-add-repository --yes --update ppa:ansible/ansible
```

```
    sudo apt-get install -y ansible
  SHELL
end
```

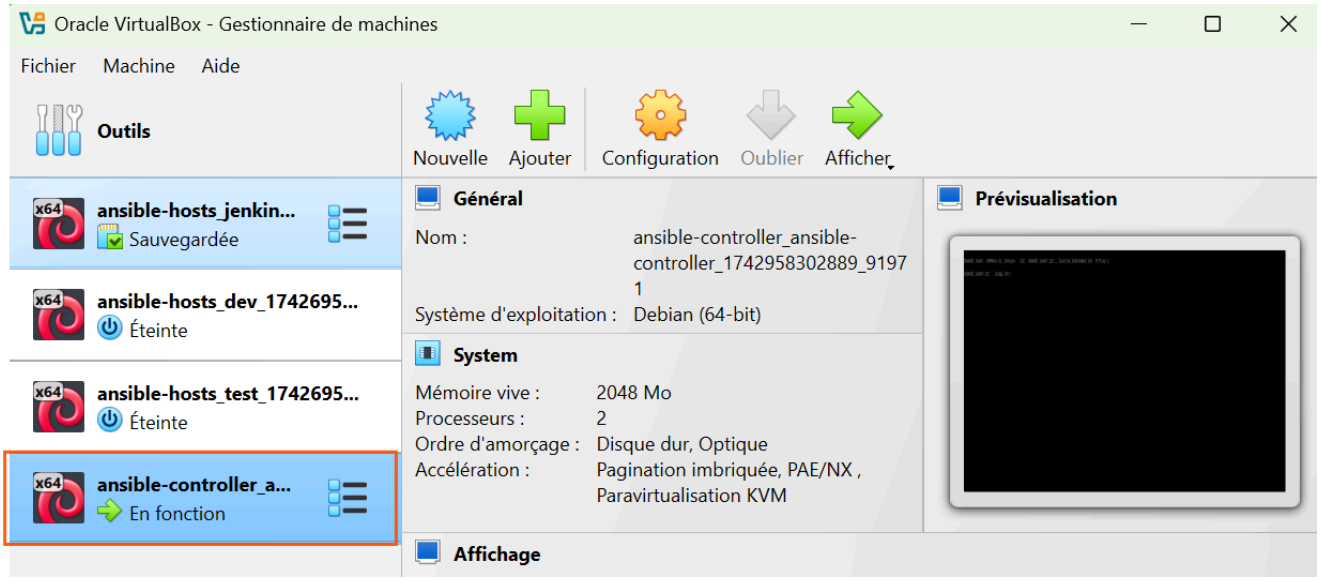
Validation et démarrage de la VM Controller

```
vagrant validate
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>vagrant validate
Vagrantfile validated successfully.
```

```
vagrant up
```

```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>vagrant up
Bringing machine 'ansible-controller' up with 'virtualbox' provider...
==> ansible-controller: Importing base box 'generic/debian12'...
==> ansible-controller: Matching MAC address for NAT networking...
==> ansible-controller: Checking if box 'generic/debian12' version '4.3.12' is up to date...
==> ansible-controller: Setting the name of the VM: ansible-controller_ansible-controller_1742958302889_91971
==> ansible-controller: Fixed port collision for 22 => 2222. Now on port 2200.
==> ansible-controller: Clearing any previously set network interfaces...
==> ansible-controller: Preparing network interfaces based on configuration...
    ansible-controller: Adapter 1: nat
==> ansible-controller: Forwarding ports...
    ansible-controller: 22 (guest) => 2200 (host) (adapter 1)
==> ansible-controller: Running 'pre-boot' VM customizations...
==> ansible-controller: Booting VM...
==> ansible-controller: Waiting for machine to boot. This may take a few minutes...
    ansible-controller: SSH address: 127.0.0.1:2200
    ansible-controller: SSH username: vagrant
    ansible-controller: SSH auth method: private key
    ansible-controller: Warning: Connection reset. Retrying...
    ansible-controller:
```



Vérification de l’installation d’Ansible

```
vagrant ssh
ansible --version
```

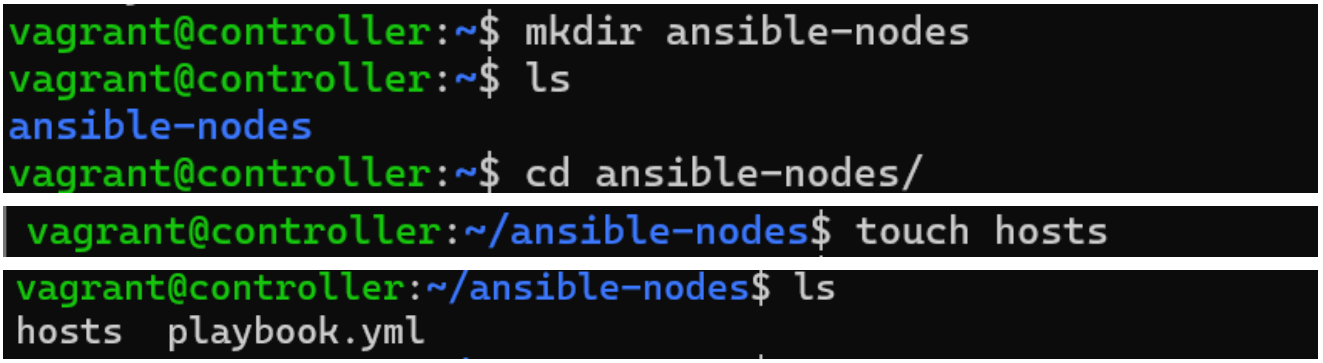
```
C:\Users\hp\Desktop\WORK\TP\AnsibleVagrant\AnsibleWork\ansible-controller>vagrant ssh
vagrant@controller:~$ ansible --version
ansible [core 2.14.18]
  config file = None
  configured module search path = ['/home/vagrant/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/vagrant/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
vagrant@controller:~$ S
```

---

4. Configuration d'Ansible

Création des fichiers de configuration

```
mkdir ansible-nodes
cd ansible-nodes/
touch hosts playbook.yml
```



Édition du fichier **hosts**

```
sudo nano hosts
```

```
vagrant@controller:~/ansible-nodes$ sudo nano hosts
```

Vérification du fichier **hosts**

```
cat hosts
```

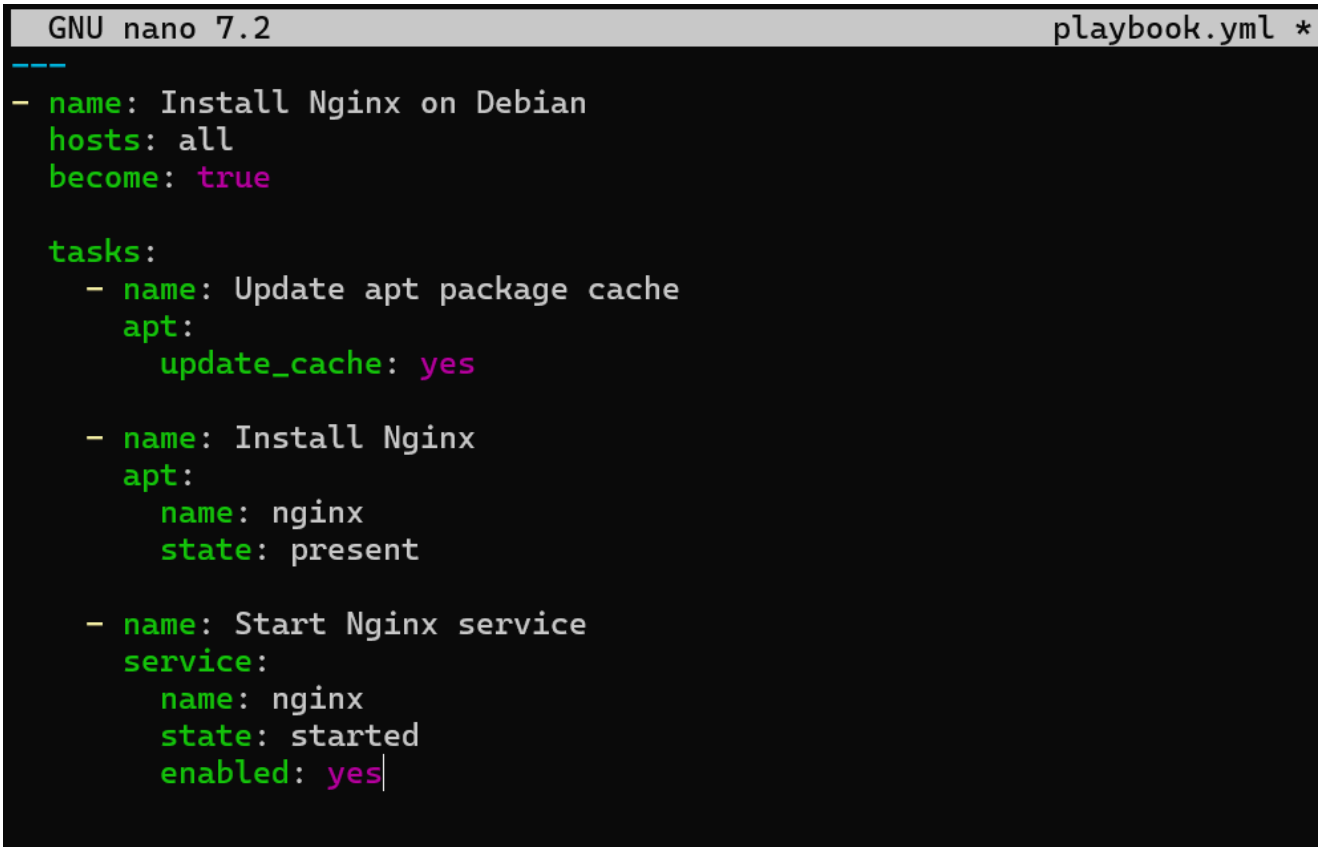
```
vagrant@controller:~/ansible-nodes$ cat hosts
[cicd]
192.168.33.10

[devs]
192.168.33.11

[tests]
192.168.33.12
vagrant@controller:~/ansible-nodes$
```

Édition du fichier **playbook.yml**

```
sudo nano playbook.yml
```

```
---
- name: Install Nginx on Debian
  hosts: all
  become: true

  tasks:
    - name: Update apt package cache
      apt:
```

```
        update_cache: yes

    - name: Install Nginx
      apt:
        name: nginx
        state: present

    - name: Start Nginx service
      service:
        name: nginx
        state: started
        enabled: yes
```



Vérification du fichier **playbook.yml**

```
cat playbook.yml
```

```
vagrant@controller:~/ansible-nodes$ cat playbook.yml
---
- name: Install Nginx on Debian
  hosts: all
  become: true

  tasks:
    - name: Update apt package cache
      apt:
        update_cache: yes

    - name: Install Nginx
      apt:
        name: nginx
        state: present

    - name: Start Nginx service
      service:
        name: nginx
        state: started
        enabled: yes
vagrant@controller:~/ansible-nodes$
```

---

5. Déploiement et Tests

Génération de la clé SSH

```
ssh-keygen
```

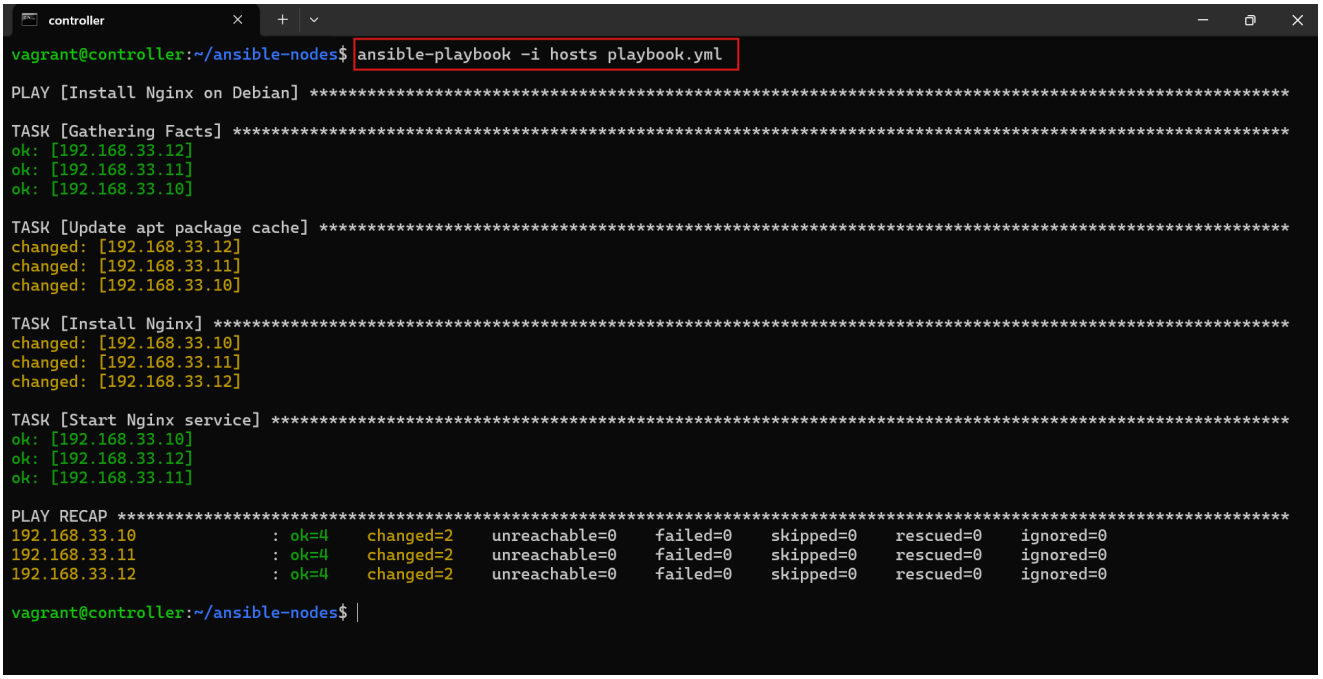
```
vagrant@controller:~/ansible-nodes$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/vagrant/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/vagrant/.ssh/id_rsa
Your public key has been saved in /home/vagrant/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:zD63ECeMl/zg8wYSzaTBbm5wn8yRbhSDlk4uTbsR3Qw vagrant@controller
The key's randomart image is:
+---[RSA 3072]----+
|      . +E+      |
|       X = o     |
|      O O +      |
|     o %**.      |
|      *.XSo.     |
|       =+O*      |
|      . o*.o     |
|          *..    |
|           .o    |
+----[SHA256]-----+
vagrant@controller:~/ansible-nodes$
```

Transfert de la clé publique vers les VM Hosts

```
ssh-copy-id vagrant@192.168.33.10
ssh-copy-id vagrant@192.168.33.11
ssh-copy-id vagrant@192.168.33.12
```

Exécution du Playbook Ansible

```
ansible-playbook -i hosts playbook.yml
```



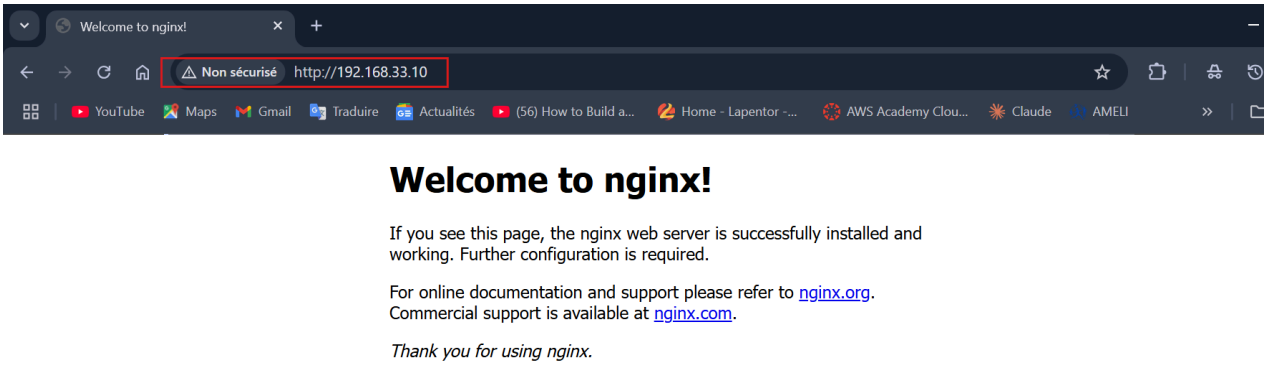
Vérification des installations

```
vagrant@jenkins:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Wed 2025-03-26 15:30:23 UTC; 9min ago
       Docs: man:nginx(8)
    Process: 3035 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
    Process: 3036 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/SUCCESS)
   Main PID: 3061 (nginx)
      Tasks: 3 (limit: 2307)
     Memory: 2.3M
        CPU: 85ms
     CGroup: /system.slice/nginx.service
             ├─3061 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             ├─3063 "nginx: worker process"
             └─3064 "nginx: worker process"

Mar 26 15:30:22 jenkins systemd[1]: Starting nginx.service - A high performance web server and a reverse proxy server...
Mar 26 15:30:23 jenkins systemd[1]: Started nginx.service - A high performance web server and a reverse proxy server.
vagrant@jenkins:~$
```

Vérification via le navigateur :

- VM1 : http://192.168.33.10/

  -



Welcome to nginx!

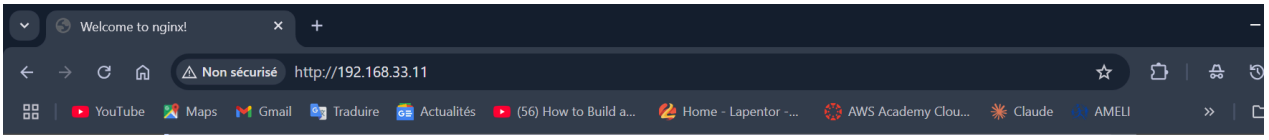
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- VM2 : http://192.168.33.11/

  -



Welcome to nginx!

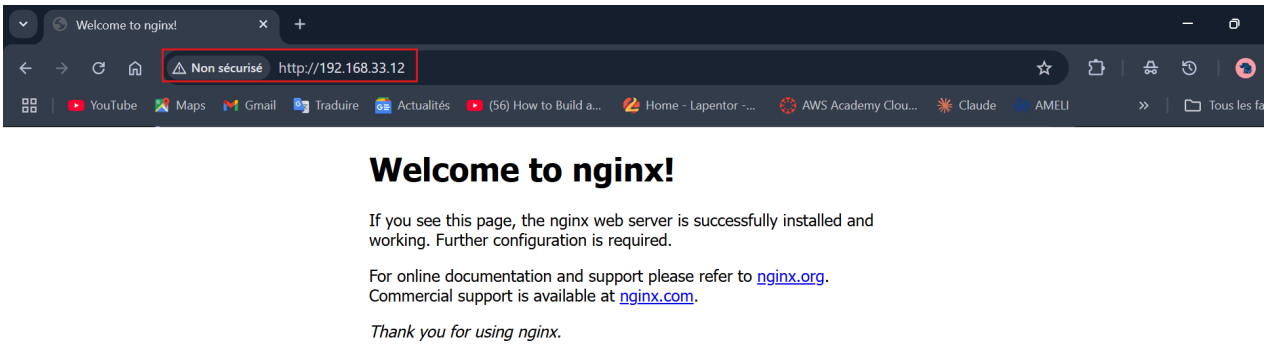
If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

- VM3 : http://192.168.33.12/

  -



Voici une conclusion pour ton document :

Conclusion

Grâce à ce TP, nous avons mis en place une infrastructure automatisée et reproductible en utilisant **Vagrant** pour la gestion des machines virtuelles et **Ansible** pour leur configuration. Nous avons appris à :

- Installer et configurer **VirtualBox** et **Vagrant**
- Définir un **Vagrantfile** pour créer plusieurs machines virtuelles
- Installer et configurer **Ansible** sur un contrôleur
- Utiliser **Ansible Playbook** pour automatiser l'installation et la configuration des services sur les machines hôtes
- Vérifier l'accès aux machines via **SSH** et tester la connectivité

L'automatisation de l'infrastructure avec ces outils permet d'éviter les configurations manuelles fastidieuses, d'améliorer la cohérence des environnements et de faciliter le déploiement des services. Cette approche est essentielle pour les pratiques DevOps modernes et constitue un excellent point de départ pour approfondir l'orchestration et la gestion des infrastructures à plus grande échelle.
