

Documentation d'Installation d'Apache sur VirtualBox avec Ubuntu Server + PHP

Documentation d'Installation d'Apache sur VirtualBox avec Ubuntu Server + PHP	1
Prérequis	1
Installation de la Machine Virtuelle	2
1/ Créer une Nouvelle Machine Virtuelle	2
2/ Quelques spécifications	3
3/ Démarrer la Machine Virtuelle	4
4/ Configuration du profil utilisateur	5
Mise à Jour du Système	5
1/ Mise à Jour du Système	6
2/ Configuration du Réseau avec Netplan	6
3/ Installation d'Apache	8
4/ Vérifier le Statut d'Apache	8
5/ Redémarrer Apache	9
6/ Vérification d'Apache	9
Installation de PHP	11
1/ Installer le Module PHP pour Apache	11
2/ Vérifier que PHP est bien installé	12
3/ Créer un Fichier de Test PHP	12
Résumé des commandes utilisées	13

Prérequis

Oracle VM VirtualBox :

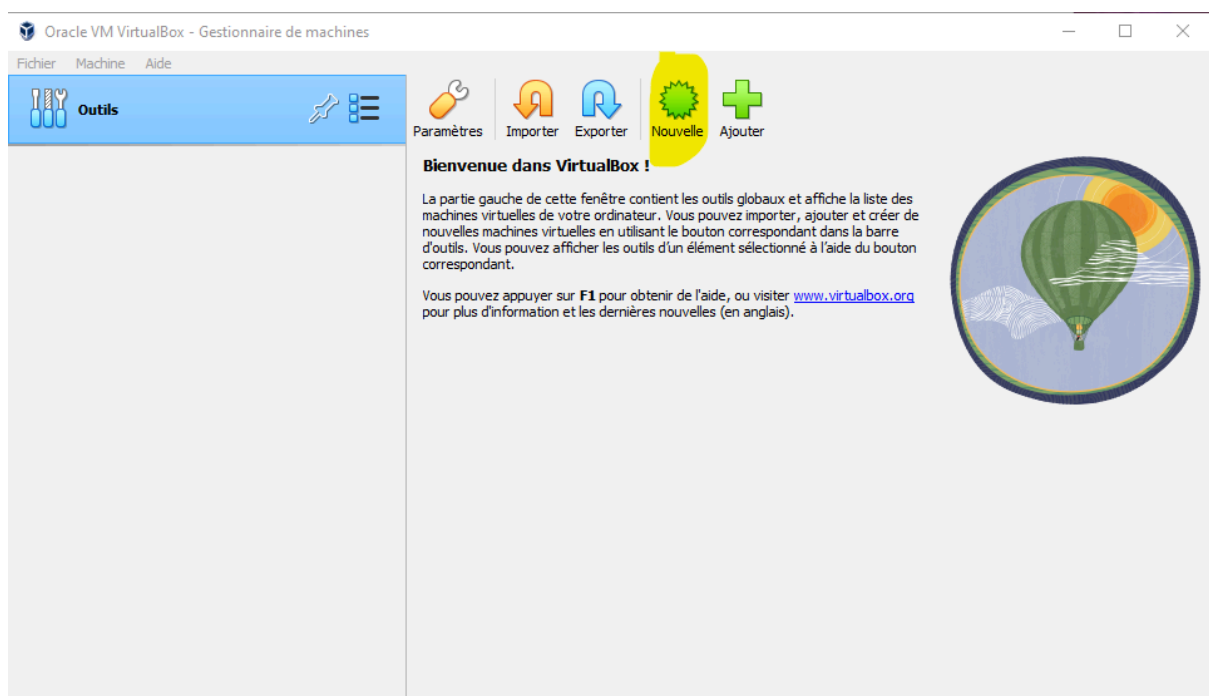
- Assurez-vous d'avoir installé Oracle VM VirtualBox. Vous pouvez le télécharger [ici](#).
- Lancez le programme et suivez les étapes d'installation.

ISO d'Ubuntu Server :

- Téléchargez l'image ISO d'Ubuntu Server à partir du [site officiel](#).

Installation de la Machine Virtuelle

Créer une Nouvelle Machine Virtuelle :



- Précisez le nom, le dossier et sélectionnez l'ISO d'Ubuntu Server.
- Cochez "Skip Unattended Installation" et cliquez sur Suivant.

Crée une machine virtuelle

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Nom :

Folder :

ISO Image :

Edition :

Type :

Version :

☒ Skip Unattended Installation

You have selected to skip unattended guest OS install, the guest OS will need to be installed manually.

Aide Mode expert Précédent **Suivant** Annuler

Quelques spécifications :

- Chaque ordinateur peut avoir des spécifications différentes
- La taille du Disque Dur Virtuel peut-être modifiée.

Crée une machine virtuelle

Hardware

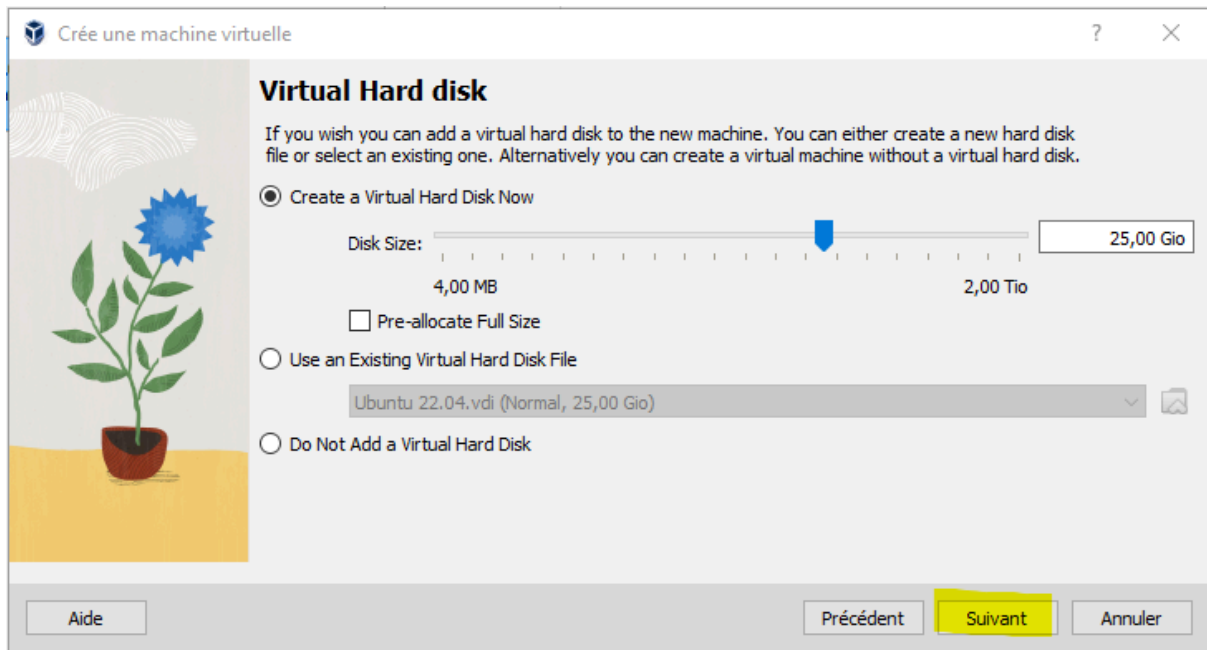
You can modify virtual machine's hardware by changing amount of RAM and virtual CPU count. Enabling EFI is also possible.

Mémoire vive :

Processors :

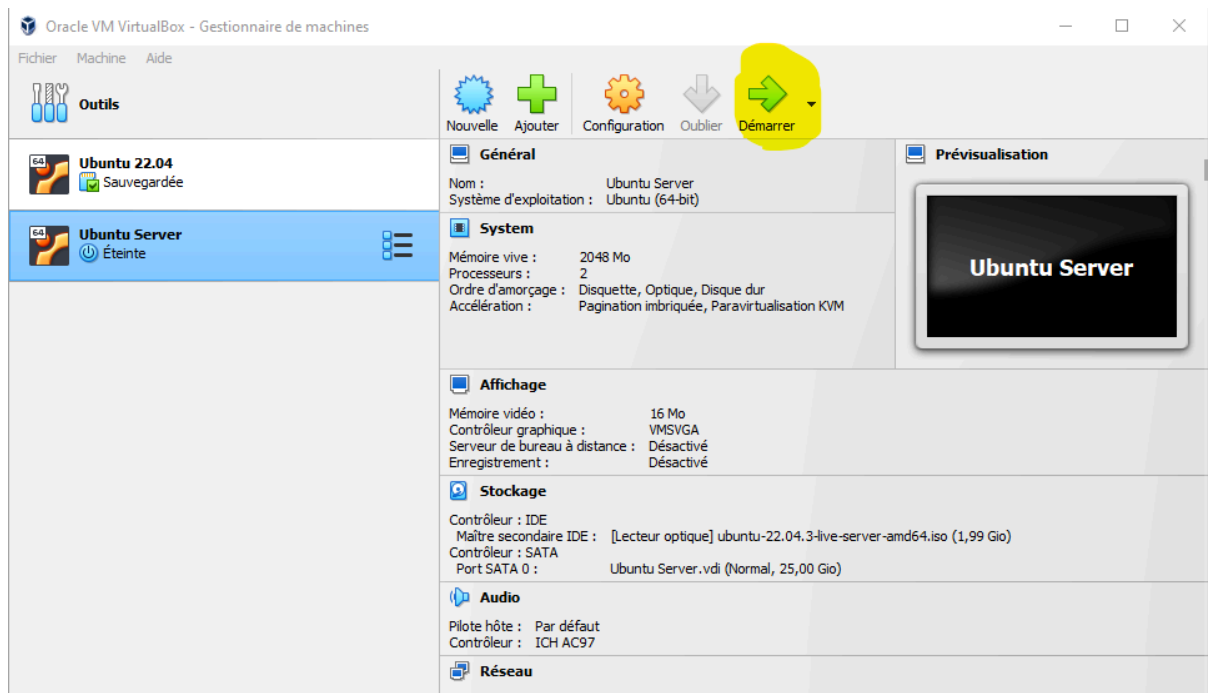
☐ Enable EFI (special OSes only)

Aide Précédent **Suivant** Annuler

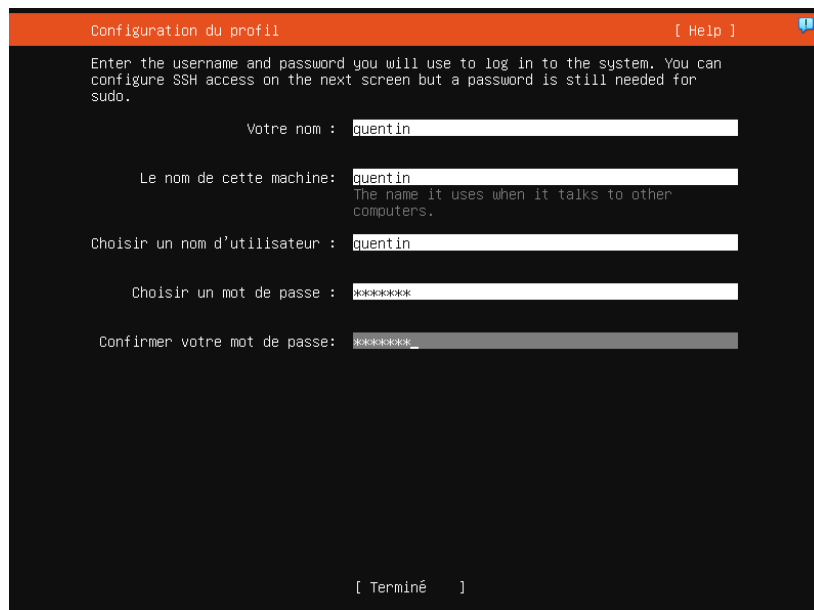


Démarrer la Machine Virtuelle :

- Cliquez sur "Terminer".
- Démarrez la machine virtuelle.



Configuration du profil utilisateur :



Mise à Jour du Système

1/ Mise à Jour du Système :

Commandes :

- **sudo apt update**
- **sudo apt upgrade**

Résultat :

```
quentin@quentin:~$ sudo apt update
[sudo] password for quentin:
Atteint :1 http://fr.archive.ubuntu.com/ubuntu jammy InRelease
Atteint :2 http://fr.archive.ubuntu.com/ubuntu jammy-updates InRelease
Atteint :3 http://fr.archive.ubuntu.com/ubuntu jammy-backports InRelease
Atteint :4 http://fr.archive.ubuntu.com/ubuntu jammy-security InRelease
Réception de :5 http://fr.archive.ubuntu.com/ubuntu jammy/main Translation-fr [486 kB]
Réception de :6 http://fr.archive.ubuntu.com/ubuntu jammy/restricted Translation-fr [4 760 B]
Réception de :7 http://fr.archive.ubuntu.com/ubuntu jammy/universe Translation-fr [3 564 kB]
Réception de :8 http://fr.archive.ubuntu.com/ubuntu jammy/multiverse Translation-fr [93,5 kB]
4 149 ko réceptionnés en 5s (865 ko/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
52 paquets peuvent être mis à jour. Exécutez « apt list --upgradable » pour les voir.
quentin@quentin:~$
```

2/ Configuration du Réseau avec Netplan :

Éditez le fichier de configuration Netplan :

Modifiez le fichier pour spécifier l'adresse IP, le masque de sous-réseau et la passerelle.

Commande :

- **sudo nano /etc/netplan/*yaml**

Résultat de base :

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
  version: 2
```

Modification à apporter :

```
# This is the network config written by 'subiquity'
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s3:
      dhcp4: false
      addresses:
        - 10.29.156.71/16
      gateway4: 10.29.0.254
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

(Chaque adresse pour 'addresses' et 'gateway' seront différentes pour chaque utilisateur.)

3/ Installation d'Apache :

Commandes :

- **sudo apt install apache2**

```
Enabling module mpm_event.
Enabling module authz_core.
Enabling module authz_host.
Enabling module authn_core.
Enabling module auth_basic.
Enabling module access_compat.
Enabling module authn_file.
Enabling module authz_user.
Enabling module alias.
Enabling module dir.
Enabling module autoindex.
Enabling module env.
Enabling module mime.
Enabling module negotiation.
Enabling module setenvif.
Enabling module filter.
Enabling module deflate.
Enabling module status.
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-htcacheclean.service.
invoke-rc.d: policy-rc.d denied execution of start.
Traitement des actions différées (« triggers ») pour ufw (0.36-6ubuntu1.1) ...
Traitement des actions différées (« triggers ») pour systemd (245.4-4ubuntu3.23) ...
Traitement des actions différées (« triggers ») pour man-db (2.9.1-1) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-0ubuntu9.14) ...
```

4/ Vérifier le Statut d'Apache :

Confirme le fonctionnement d'Apache, identifie les problèmes, affiche les erreurs et facilite la maintenance du serveur web.

Commande :

- **sudo systemctl status apache2**

Exemple :

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
   Active: inactive (dead)
     Docs: https://httpd.apache.org/docs/2.4/
```

(Optionnel) Autoriser le Trafic HTTP dans le Pare-feu :

Commande :

- **sudo ufw allow 80**

5/ Redémarrer Apache :

Commande :

- **systemctl start apache2.service**

```
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to restart 'apache2.service'.
```

Demande du mot de passe pour la confirmation

```
==== AUTHENTICATION COMPLETE ====
```

6/ Vérification d'Apache :

Ouvrez un navigateur et accédez à l'adresse IP de la machine virtuelle.
Vous devriez voir la page par défaut d'Apache



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`, `a2dissite`, 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.

Tips :

Plusieurs commande à propos d'Apache :

- Démarrer Apache : **sudo systemctl start apache2**
- Arrêter Apache : **sudo systemctl stop apache2**
- Redémarrer Apache : **sudo systemctl restart apache2**

Installation de PHP

1/ Installer le Module PHP pour Apache :

Commande :

- **sudo apt install libapache2-mod-php7.***

```
sudo apt install libapache2-mod-php7.*
```

```
Creating config file /etc/php/7.4/mods-available/sysvshm.ini with new version
Creating config file /etc/php/7.4/mods-available/tokenizer.ini with new version
Paramétrage de php7.4-readline (7.4.3-4ubuntu2.19) ...
Creating config file /etc/php/7.4/mods-available/readline.ini with new version
Paramétrage de php7.4-opcache (7.4.3-4ubuntu2.19) ...
Creating config file /etc/php/7.4/mods-available/opcache.ini with new version
Paramétrage de php7.4-json (7.4.3-4ubuntu2.19) ...
Creating config file /etc/php/7.4/mods-available/json.ini with new version
Paramétrage de php7.4-cli (7.4.3-4ubuntu2.19) ...
update-alternatives: utilisation de « /usr/bin/php7.4 » pour fournir « /usr/bin/php » (php) en mode automatique
update-alternatives: utilisation de « /usr/bin/phar7.4 » pour fournir « /usr/bin/phar » (phar) en mode automatique
update-alternatives: utilisation de « /usr/bin/phar.phar7.4 » pour fournir « /usr/bin/phar.phar » (phar.phar) en mode automatique
Creating config file /etc/php/7.4/cli/php.ini with new version
Paramétrage de libapache2-mod-php7.4 (7.4.3-4ubuntu2.19) ...
Creating config file /etc/php/7.4/apache2/php.ini with new version
Module mpm_event disabled.
Enabling module mpm_prefork.
apache2_switch_mpm Switch to prefork
invoke-rc.d: policy-rc.d denied execution of restart.
apache2_invoke: Enable module php7.4
invoke-rc.d: policy-rc.d denied execution of restart.
Traitement des actions différées (« triggers ») pour man-db (2.9.1-1) ...
Traitement des actions différées (« triggers ») pour php7.4-cli (7.4.3-4ubuntu2.19) ...
Traitement des actions différées (« triggers ») pour libapache2-mod-php7.4 (7.4.3-4ubuntu2.19) ...
```

2/ Vérifier que PHP est bien installé :

Plusieurs commandes à effectuer :

- **cd /etc/php/7.*/**
- **ls**

```
apache2 cli mods-available
```

- Vérifier la présence de 'apache2'. On peut donc faire :
- **cd apache2**
- **ls**

```
conf.d php.ini
```

Si vous avez 'php.ini', tout est bien installé.

3/ Créer un Fichier de Test PHP :

On créer un fichier de test :

Commande :

- **sudo nano /var/www/html/phpinfo.php**

Ajoutez le contenu suivant dans le fichier phpinfo.php :



```
<?php
    phpinfo();
?>
```

```
GNU nano 4.8
<?php
    phpinfo();
?>
```

Enregistrez le fichier, en faisant CTRL + X

Redémarrer Apache à l'aide de la commande **sudo systemctl restart apache2**

Maintenant, allez sur votre navigateur, écrire votre IP/ + le nom du dossier, en l'occurrence : **phpinfo.php**

PHP Version 7.4.3-4ubuntu2.19	
	
System	Linux parisc 5.4.0-169-generic #187-Ubuntu SMP Thu Nov 23 14:52:28 UTC 2023 x86_64
Build Date	Jun 27 2023 15:49:59
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gdlib.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-xml.ini, /etc/php/7.4/apache2/conf.d/20-zip.ini, /etc/php/7.4/apache2/conf.d/20-zlib.ini, /etc/php/7.4/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.4/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini, /etc/php/7.4/apache2/conf.d/20-yaml.ini, /etc/php/7.4/apache2/conf.d/20-zip.ini, /etc/php/7.4/apache2/conf.d/20-zlib.ini, /etc/php/7.4/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.4/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.4/apache2/conf.d/20-xsl.ini, /etc/php/7.4/apache2/conf.d/20-yaml.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902.NTS
PHP Extension Build	API20190902.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*
<small>This program makes use of the Zend Scripting Language Engine: Zend Engine v3.4.0, Copyright (c) Zend Technologies with Zend OPcache v7.4.3-4ubuntu2.19, Copyright (c), by Zend Technologies</small>	
	

Voilà ! Tout est bien installé.

Résumé des commandes utilisées :

Ces commandes installent, configurent Apache et PHP, vérifient le statut d'Apache et créent un fichier de test PHP.

Mise à Jour du Système :

```
sudo apt update
```

```
sudo apt upgrade
```

Configuration du Réseau avec Netplan :

```
sudo nano /etc/netplan/*.yaml
```

Installation d'Apache :

```
sudo apt install apache2
```

Vérifier le Statut d'Apache :

```
sudo systemctl status apache2
```

Autoriser le Trafic HTTP dans le Pare-feu (Optionnel) :

```
sudo ufw allow 80
```

Démarrer Apache : `sudo systemctl start apache2`

Arrêter Apache : `sudo systemctl stop apache2`

Redémarrer Apache : `sudo systemctl restart apache2`

Installation de PHP :

```
sudo apt install libapache2-mod-php7.*
```

Créer un Fichier de Test PHP :

```
sudo nano /var/www/html/phpinfo.php
```

Ajoutez :

```
<?php  
    phpinfo();  
?>
```

Vérifier l'Installation de PHP dans le Navigateur :

Accédez à l'adresse IP/phpinfo.php