**Date :** 11/03/2024

https://[www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-](http://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-20-) 04-fr

**Tutoriel :**

**[Installer un Serveur Web Apache avec Ubuntu]**

**Auteur :** Le Groupe De Sacha

**Version :** V.1

Sommaire :

1. [- Étape 1 — Installation d’Apache 3](#_bookmark0)
2. [– Étape 2 — Réglage du pare-feu 3](#_bookmark1)
3. [– Étape 3 - Vérification de votre serveur Web 4](#_bookmark2)
4. [– Étape 4 — Gestion du processus Apache 6](#_bookmark3)
5. [– Étape 5 — Configuration des hôtes virtuels 6](#_bookmark4)

# - Étape 1 — Installation d’Apache

L’installer à l’aide d’outils classiques de gestion des packages.

Commençons par mettre à jour l’index local des packages pour refléter tout nouveau changement en amont :

sudo apt update

Ensuite, installez le package apache2 :

sudo apt install apache2

Après avoir confirmé l’installation, apt installera Apache et toutes les dépendances requises.

# – Étape 2 — Réglage du pare-feu

Avant de tester Apache, il est nécessaire de modifier les paramètres du pare-feu pour permettre à l’extérieur d’accéder aux ports web par défaut. En supposant que vous ayez suivi les instructions des conditions préalables, vous devriez avoir un pare-feu UFW configuré pour restreindre l’accès à votre serveur.

Lors de l’installation, Apache s’enregistre auprès de l’UFW pour fournir quelques profils

d’application qui peuvent être utilisés pour activer ou désactiver l’accès à Apache à travers le pare-feu.

Listez les profils d’application ufw en tapant :

sudo ufw app list

Vous recevrez une liste des profils d’application :

Résulta :

Available applications: Apache

Apache Full Apache Secure

OpenSSH

Comme le montre la sortie, il existe trois profils pour Apache :

**Apache :** ce profil n’ouvre que le port 80 (trafic web normal, non crypté).

**Apache Full :** ce profil ouvre à la fois le port 80 (trafic web normal, non crypté) et le port 443 (trafic crypté TLS/SSL).

**Apache Secure :** ce profil n’ouvre que le port 443 (trafic crypté TLS/SSL).

Nous vous recommandons d’activer le profil le plus restrictif qui autorisera tout de même le trafic que vous avez configuré. Comme nous n’avons pas encore configuré le SSL pour notre serveur dans ce guide, nous devrons seulement autoriser le trafic sur le port 80 :

sudo ufw allow 'Apache'

# – Étape 3 - Vérification de votre serveur Web

À la fin du processus d’installation, Ubuntu 20.04 lance Apache. Le serveur Web devrait déjà être opérationnel.

Vérifiez avec le système systemd init pour vous assurer que le service fonctionne en tapant

:

**Résulta :**

* apache2.service - The Apache HTTP Server

Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled) Active: active (running) since Thu 2020-04-23 22:36:30 UTC; 20h ago

Docs: https://httpd.apache.org/docs/2.4/ Main PID: 29435 (apache2)

Tasks: 55 (limit: 1137) Memory: 8.0M

CGroup: /system.slice/apache2.service

├─29435 /usr/sbin/apache2 -k start

├─29437 /usr/sbin/apache2 -k start

└─29438 /usr/sbin/apache2 -k start

sudo systemctl status apache2

Comme cette sortie le confirme, le service a démarré avec succès. Cependant, la meilleure façon de le tester consiste à demander une page à Apache.

Vous pouvez accéder à la page d’accueil par défaut d’Apache pour confirmer que le logiciel fonctionne correctement grâce à votre adresse IP : Si vous ne connaissez pas l’adresse IP de votre serveur, vous pouvez l’obtenir de plusieurs façons différentes à partir de la ligne de commande.

Essayez de taper ceci à l’invite de commande de votre serveur :

hostname -I

Vous obtiendrez quelques adresses séparées par des espaces. Vous pouvez essayer chacune d’elles dans votre navigateur web pour déterminer s’ils fonctionnent.

Utiliser l’outil Icanhazip, qui devrait vous donner votre adresse IP publique telle qu’elle est lue depuis un autre endroit sur Internet, constitue une autre option :

curl -4 icanhazip.com **(Si la commande ne fonctionne pas , installer curl)**

Une fois que vous avez l’adresse IP de votre serveur, saisissez-la dans la barre d’adresse de votre navigateur :

http://your\_server\_ip

Vous devriez voir la page web par défaut d’Apache Ubuntu 20.04 :



# – Étape 4 — Gestion du processus Apache

Maintenant que votre serveur web est opérationnel, passons en revue quelques commandes de gestion de base en utilisant systemctl.

Pour arrêter votre serveur Web, tapez :

sudo systemctl stop apache2

Pour démarrer le serveur web lorsqu’il est arrêté, tapez :

sudo systemctl start apache2

Pour arrêter puis redémarrer le service, tapez :

sudo systemctl restart apache2

Si vous procédez uniquement à des modifications de configuration, il se peut qu’Apache recharge souvent sans interrompre les connexions. Pour ce faire, utilisez cette commande :

sudo systemctl reload apache2

Par défaut, Apache est configuré pour un lancement automatique au démarrage du

serveur. Si ce n’est pas ce que vous souhaitez, désactivez ce comportement en tapant :

sudo systemctl disable apache2

Pour réactiver le service de lancement automatique au démarrage, tapez :

sudo systemctl enable apache2

Désormais, Apache devrait démarrer automatiquement au redémarrage du serveur.

# – Étape 5 — Configuration des hôtes virtuels

Lorsque vous utilisez le serveur web Apache, vous pouvez utiliser des hôtes virtuels

(similaires aux blocs de serveurs dans Nginx) pour encapsuler les détails de la configuration et héberger plusieurs domaines à partir d’un seul serveur. Nous allons configurer un

domaine appelé your\_domain, mais vous devez le remplacer par le nom de votre domaine. Si vous configurez un nom de domaine avec DigitalOcean, veuillez consulter notre

Documentation de mise en réseau.

Apache sur Ubuntu 20.04 a un bloc serveur activé par défaut qui est configuré pour servir des documents à partir du répertoire /var/www/html. Même si cela fonctionne bien pour un seul site, cela peut devenir ingérable si vous hébergez plusieurs sites. Au lieu de

modifier /var/www/html, créons une structure de répertoire au sein de /var/www pour un your\_domain, en laissant /var/www/html en place comme répertoire par défaut qui sera utilisé si une demande du client ne correspond à aucun autre site.

Créez le répertoire pour your\_domain comme suit :

sudo mkdir /var/www/your\_domain

Ensuite, attribuez la propriété du répertoire avec la variable d’environnement $USER

sudo chown -R $USER:$USER /var/www/your\_domain

Les autorisations de la racine de votre site Internet devraient être correctes si vous n’avez pas modifié votre valeur umask, qui définit les autorisations de fichier par défaut. Pour vous assurer que vos autorisations sont correctes et permettre au propriétaire de lire,

écrire et exécuter les fichiers tout en accordant uniquement des autorisations de lecture et d’exécution aux groupes et aux autres, vous pouvez saisir la commande suivante :

sudo chmod -R 755 /var/www/your\_domain

Ensuite, créez une page index.html à l’aide de nano ou de votre éditeur favori :

sudo nano /var/www/your\_domain/index.html

À l’intérieur, ajoutez l’exemple de HTML suivant :

<html>

<head>

<title>Welcome to Your\_domain!</title>

</head>

<body>

<h1>Success! The your\_domain virtual host is working!</h1>

</body>

</html>

Enregistrez et fermez le fichier lorsque vous avez terminé.

Pour qu’Apache puisse servir ce contenu, il est nécessaire de créer un fichier d’hôte virtuel avec les directives correctes. Au lieu de modifier directement le fichier de configuration par défaut situé dans /etc/apache2/sites-available/000-default.conf, créons-en un nouveau dans /etc/apache2/sites-available/your\_domain.conf :

sudo nano /etc/apache2/sites-available/your\_domain.conf

Collez le bloc de configuration suivant, qui est similaire à celui par défaut, mais mis à jour pour notre nouveau répertoire et notre nom de domaine :

<VirtualHost \*:80>

ServerAdmin webmaster@localhost ServerName your\_domain

ServerAlias www.your\_domain

DocumentRoot /var/www/your\_domain ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Notez que nous avons mis à jour le DocumentRoot dans notre nouveau répertoire et le ServerAdmin dans un e-mail auquel l’administrateur du site your\_domain peut accéder.

Nous avons également ajouté deux directives : ServerName, qui établit le domaine de base qui doit correspondre à cette définition d’hôte virtuel, et ServerAlias, qui définit d’autres noms qui doivent correspondre comme s’ils étaient le nom de base.

Enregistrez et fermez le fichier lorsque vous avez terminé. Activons le fichier avec l’outil a2ensite :

sudo a2ensite your\_domain.conf

Désactivez le site par défaut défini dans 000-default.conf :

sudo a2dissite 000-default.conf

Ensuite, effectuons un test à la recherche d’éventuelles erreurs de configuration :

sudo apache2ctl configtest

Vous devriez voir la sortie suivante :

Résulta :

Syntax OK

Redémarrez Apache pour implémenter vos modifications :

sudo systemctl restart apache2

Apache devrait maintenant vous présenter votre nom de domaine. Vous pouvez le vérifer en allant sur http://your\_domain où vous devriez voir quelque chose similaire à ceci :



Pour modifier la page , utiliser cette commande :

sudo nano /var/www/your\_domain/index.html

**Puis modifier comme bon vous semble votre page, exemple :**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bienvenue sur le Serveur Web 2 - Test</title>

</head>

<body>

<h1>Bienvenue sur le Serveur Web 2 - Test</h1>

</body>

</html>