

Projet Nginx

Introduction

Nginx est l'un des serveurs web les plus populaires au monde, il est aussi responsable d'héberger certains des sites les plus gros et les plus visités d'internet. Dans la plupart des cas, il utilise moins de ressources qu'Apache et peut être utilisé en tant que serveur web ou proxy inverse.

Dans ce guide, nous discuterons de la manière d'installer Nginx sur votre serveur Ubuntu 18.04.

Prérequis

Avant que vous ne débutiez ce guide, vous devriez avoir configuré sur votre serveur, un utilisateur régulier, qui n'est pas un utilisateur root, mais qui a des privilèges sudo. Vous pouvez apprendre comment configurer un profil d'utilisateur régulier en suivant notre « Guide de configuration initiale du serveur Ubuntu 18.04 » [initial server setup guide for Ubuntu 18.04](#).

Lorsque vous avez un profil de disponible, connectez-vous en tant qu'utilisateur régulier (et non en tant qu'utilisateur root) pour débiter.

Étape 1 — Installer Nginx

Comme Nginx est disponible dans les répertoires par défaut de Ubuntu, il est possible de l'installer à partir de ceux-ci en utilisant l'outil de paquetage avancé `apt` [APT packaging system].

Comme c'est notre première interaction avec l'outil de paquetage 'apt' sur cette session, nous allons mettre à jour notre index de paquet local pour avoir accès aux plus récents. Ensuite, nous pourrons installer `Nginx`:

- `sudo apt update`
-
- `sudo apt install nginx`
-

Après avoir accepté la procédure, `apt` installera Nginx et tous ses prérequis à votre serveur.

Étape 2 — Ajustement du pare-feu

Avant d'essayer Nginx, le logiciel pare-feu doit être ajusté pour laisser l'accès au service. Nginx s'enregistrera en tant que service avec `ufw` au moment de l'installation, ce qui lui simplifie l'accès.

Pour une liste complète des configurations que `ufw` sait utiliser, entrer :

- `sudo ufw app list`
-

Vous devriez avoir une liste des profils d'application :

Output

```
Available applications: (applications disponibles)
```

```
Nginx Full
```

```
Nginx HTTP
```

```
Nginx HTTPS
```

```
OpenSSH
```

Comme vous pouvez le voir, il y a trois profils disponibles pour Nginx :

- **Nginx Full:** Ce profil ouvre à la fois le port 80 (normal, trafic internet non-encrypté) et le port 443 (trafic internet encrypté par TLS/SSL)
- **Nginx HTTP:** Ce profil ouvre seulement le port 80 (normal, trafic internet non-encrypté)
 - **Nginx HTTPS:** Ce profil ouvre seulement le port 443 (trafic internet encrypté par TLS/SSL) Il est recommandé que vous choisissiez le profil le plus restrictif qui vous permettra tout de même le trafic que vous avez configuré. Comme nous n'avons pas encore configuré SSL pour nos serveurs dans ce guide, nous n'aurons qu'à donner l'accès au trafic sur le port 80. Vous pouvez le faire en entrant : `command`
 - `sudo ufw allow 'Nginx HTTP'`
 - Vous pouvez vérifier le changement en entrant : `command`
 - `sudo ufw status`
 -

Vous devriez voir le trafic internet HTTP permis sous le format d'affichage:

Output

```
Status: active
```

```
To Action From
```

```
--
```

```
OpenSSH ALLOW Anywhere
```

```
Nginx HTTP ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
Nginx HTTP (v6) ALLOW Anywhere (v6)
```

Étape 3 — Vérification de votre serveur web

À la fin du processus d'installation, Ubuntu 18.04 lance Nginx. Le serveur web devrait déjà être opérationnel.

Nous pouvons vérifier avec le system init `systemd` pour nous assurer du bon fonctionnement en entrant :

- `systemctl status nginx`
-

Output

```
• nginx.service - A high performance web server and a reverse proxy server
Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset:
enabled)
Active: active (running) since Fri 2018-04-20 16:08:19 UTC; 3 days ago
Docs: man:nginx(8)
Main PID: 2369 (nginx)
Tasks: 2 (limit: 1153)
CGroup: /system.slice/nginx.service
└─2369 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
   └─2380 nginx: worker process
```

Comme vous pouvez le voir plus haut, le service semble avoir démarré avec succès. Cependant, le meilleur moyen de faire la vérification est de concrètement requêter une page à Nginx.

Vous pouvez accéder la page d'accueil par défaut de Nginx pour confirmer que le logiciel opère sans problèmes en naviguant vers l'adresse IP de votre serveur. Si vous ne connaissez pas l'adresse IP de votre serveur, il y a différentes méthodes pour l'obtenir.

Essayez d'entrer ceci dans votre utilitaire de commande :

- `ip addr show eth0 | grep inet | awk '{ print $2; }' | sed 's/\./.*$//'`
-

Vous aurez une réponse de quelques lignes. Vous pouvez essayer dans chacun de vos navigateurs pour voir s'ils fonctionnent.

Une alternative est d'entrer ceci, qui devrait vous fournir votre adresse IP publique tel que vu d'un emplacement différent de l'internet :

- `curl -4 icanhazip.com`
-

Quand vous avez l'adresse IP de votre serveur, entrer la dans la barre de recherche de votre navigateur :

`http://your_server_ip`

Vous devriez voir la page d'accueil par défaut de Nginx :

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.

Cette page est incluse avec Nginx pour démontrer que le serveur fonctionne correctement.

Étape 4 — Gestion des procédés Nginx

Maintenant que votre serveur web est opérationnel, revoyons quelques commandes de gestion de base.

Pour arrêter votre serveur web, entrez :

- `sudo systemctl stop nginx`
-

Pour démarrer votre serveur web lorsqu'il est arrêté, entrez :

- `sudo systemctl start nginx`
-

Pour arrêter, puis redémarrer le service, entrez :

- `sudo systemctl restart nginx`
-

Si vous apportez seulement des modifications de configuration, Nginx peu souvent redémarrer sans interrompre la connexion. Pour ce faire, entrez :

- `sudo systemctl reload nginx`
-

Par défaut, Nginx est configuré pour démarrer automatiquement quand le serveur s'allume. Si ce n'est pas ce que vous voulez, vous pouvez désactiver ce comportement en entrant :

- `sudo systemctl disable nginx`
-

Pour réactiver le démarrage automatique, vous pouvez entrer :

- `sudo systemctl enable nginx`
-

Étape 5 — Configuration des blocs de serveur (Recommandé)

Lorsque vous utilisez le serveur web Nginx, *server blocks* (semblable à «hôte virtuel» dans Apache) peut être utilisé pour encapsuler des détails de configuration afin d'être hôte à plus d'un serveur. Nous allons mettre en place un domaine appelé **example.com** mais vous devriez **remplacer ceci par votre propre nom de domaine**. Pour en savoir plus à propos de la procédure de mise en place d'un tel serveur avec DigitalOcean, veuillez voir Introduction à DigitalOcean DNS [Introduction to DigitalOcean DNS](#).

Nginx sur Ubuntu 18.04 a un bloc de serveur d'actif par défaut qui est configuré pour desservir des documents à partir d'un répertoire à `/var/www/html`. Malgré que ceci fonctionne bien pour un site unique, il pourrait vite devenir lourd si vous hébergez de multiples sites. Au lieu de modifier `/var/www/html`, créons une structure de répertoire à l'intérieur de `/var/www/html` pour notre site **example.com**, ce qui laissera `/var/www/html` en place comme le répertoire par défaut à être desservi si la requête d'un client ne correspond à aucuns autres sites.

Créez le répertoire pour **example.com** comme suit, en utilisant `-p` pour créer n'importe quels autres répertoires apparentés :

- `sudo mkdir -p /var/www/example.com/html`
-

Ensuite, attribuez les droits de propriétaire du répertoire avec l'environnement variable `$USER` :

- `sudo chown -R $USER:$USER /var/www/example.com/html`
-

Les permissions devrait être correctes si vous n'avez pas modifié votre valeur ``umask'`, mais vous pouvez vous en assurer en entrant :

- `sudo chmod -R 755 /var/www/example.com`
-

Ensuite, créez un échantillon de page `index.html` en utilisant `nano` sur votre éditeur préféré :

- `nano /var/www/example.com/html/index.html`
-

À l'intérieur, entrez l'échantillon HTML suivant :

```
/var/www/example.com/html/index.html

<html>
<head>
<title>Welcome to Example.com!</title>
</head>
<body>
<h1>Success! The example.com server block is working!</h1>
</body>
</html>
```

Sauvegardez et fermez le fichier lorsque vous avez terminé.

Afin que Nginx desserve ce contenu, il est nécessaire de créer un bloc serveur avec les bonnes directives. Au lieu de modifier le fichier de configuration par défaut directement, faites-en un nouveau a `/etc/nginx/sites-available/example.com`:

- `sudo nano /etc/nginx/sites-available/example.com`
-

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

```
/etc/nginx/sites-available/example.com

server {
listen 80;
listen [::]:80;
root /var/www/example.com/html;
index index.html index.htm index.nginx-debian.html;
server_name example.com www.example.com;
location / {
try_files $uri $uri/ =404;
}
}
```

Veuillez noter que nous avons fait la mise à jour la configuration `root` de notre nouveau répertoire et le `server_name` de notre nom de domaine.

Pour poursuivre, activons le fichier en créant un lien vers celui-ci dans le répertoire `sites-enabled`, que Nginx lit au démarrage :

- `sudo ln -s /etc/nginx/sites-available/example.com /etc/nginx/sites-enabled/`
-

Deux blocs serveurs sont maintenant activés et configurés pour répondre aux requêtes en se basant sur leurs directives `listen` et `server_name` (vous pouvez en lire davantage sur la manière dont Nginx procède ces directives ici [here](#)):

- `example.com`: Répondra aux requêtes pour `example.com` et `www.example.com`.
- `default`: Répondra à n'importe quelle sur le port 80 qui ne correspond pas aux deux autres blocs. Pour éviter un problème avec la mémoire de hachage qui pourrait survenir lors de l'ajout de nom de serveur additionnels, il est nécessaire d'ajuster une seule valeur dans le fichier `/etc/nginx/nginx.conf`. Ouvrez le fichier : `command`
- `sudo nano /etc/nginx/nginx.conf`
- Trouvez la directive `server_names_hash_bucket_size` et enlevez le symbole `#` pour décommenter la ligne :
- `[label /etc/nginx/nginx.conf]`
- `...`
- `http {`
- `...`
- `server_names_hash_bucket_size 64;`
- `...`
- `}`
- `...`
- Ensuite, faite la vérification pour vous assurer qu'il n'y a pas d'erreurs de syntaxe dans vos fichiers Nginx : `command`
- `sudo nginx -t`
- Enregistrez et fermez le fichier lorsque vous avez terminé. S'il n'y a pas de problèmes, redémarrez Nginx pour activer vos changements : `command`
- `sudo systemctl restart nginx`
- Nginx devrait maintenant desservir votre nom de domaine. Vous pouvez vérifier ceci en naviguant à `http://example.com`, ou vous verrez quelque chose comme ceci :

Success! The example.com server block is working!

Étape 6 — Se familiariser avec les fichiers et répertoires importants de Nginx

Maintenant que vous savez comment gérer le service Nginx, vous devriez prendre quelques minutes pour vous familiariser avec quelques fichiers et répertoires importants.

Contenu

- `/var/www/html`: Le contenu web qui, par défaut est seulement la page d'accueil de Nginx que vous avez vu plus tôt, est desservi à travers le répertoire `/var/www/html`. Ceci peut être changé en modifiant les fichiers de configuration de Nginx.

Configuration du serveur

- `/etc/nginx`: Le répertoire de configuration de Nginx. Tous les fichiers de configuration y résident.
- `/etc/nginx/nginx.conf`: Le fichier de configuration principal de Nginx. Il peut être modifié pour apporter des changements globaux à la configuration de Nginx.
- `/etc/nginx/sites-available/`: Le répertoire où tous les blocs serveurs, propres à chaque site, peuvent être stockés. Nginx n'utilisera pas les fichiers de configuration qui s'y trouvent à moins qu'ils ne soient liés à un répertoire `sites-enabled`. Typiquement, toutes les configurations de blocs serveurs sont faites à partir de ce répertoire, puis activées en les liant à d'autres répertoires.
- `/etc/nginx/sites-enabled/`: Le répertoire où les blocs serveurs (par site) activés sont stockés. Typiquement, la création se fait en les liant à des fichiers de configuration qui se retrouvent dans le répertoire `sites-available`.
- `/etc/nginx/snippets`: Ce répertoire contient des fragments de configuration qui peuvent être inclus ailleurs dans la configuration de Nginx. Les segments qui ont le potentiel de se répéter sont des candidats parfaits pour les remoduler en extraits [snippets].

Journal du serveur

- `/var/log/nginx/access.log`: Toutes les requêtes faites à votre serveur web sont enregistrées dans ce journal à moins d'avoir configuré Nginx autrement.
- `/var/log/nginx/error.log`: Toute erreur qui survient dans Nginx sera enregistrée dans ce journal.

Conclusion

Maintenant que vous avez un serveur web d'installé, vous avez plusieurs options pour le type de contenu à desservir et les technologies que vous voulez utiliser pour créer

une expérience plus riche.