tutoriel

Nginx - Guide du débutant

Ce guide décrit :

1. comment démarrer et arrêter nginx, et recharger sa configuration,
2. la structure du fichier de configuration
3. et comment :
    1. configurer nginx pour servir du contenu statique,
    2. configurer nginx comme serveur proxy,
    3. le connecter avec une application FastCGI.

Nginx a un processus maitre et plusieurs processus de travail. Le processus maitre lit et évalue la configuration et gère les processus de travail qui font le traitement réel des demandes. Le nombre de processus de travail, défini dans le fichier de configuration, peut être fixé pour une configuration donnée ou automatiquement ajusté au nombre de cœurs de processeur disponibles.

Par défaut, le fichier de configuration est **/usr/local/nginx/conf/nginx.conf**, **/etc/nginx/nginx.conf** ou **/usr/local/etc/nginx/nginx.conf**.

Démarrage, arrêt et rechargement de la configuration

Pour démarrer **nginx**, lancez le fichier exécutable.

Une fois démarré, nginx peut être contrôlé en lançant l'exécutable avec le paramètre -s :

```
USER@MACHINE:~$ nginx -s signal
```

où **signal** peut être :

1. **stop** : arrêt rapide
2. **quit** : arrêt en douceur
3. **reload** : rechargement du fichier de configuration
4. **reopen** : réouverture des fichiers de log

Par exemple, pour arrêter les processus nginx en attendant que les processus de travail aient fini de servir les demandes en cours :

```
USER@MACHINE:~$ nginx -s quit
```

Cette commande doit être exécutée par l'utilisateur qui a lancé nginx.

Les modifications du fichier de configuration ne seront appliquées qu'après la commande

```
USER@MACHINE:~$ nginx -s reload
```

ou un redémarrage.

Pour obtenir la liste de tous les processus nginx en cours d'exécution :

```
...@...:~$ ps -ax | grep nginx
```

Structure du fichier de configuration

Les modules de Nginx sont contrôlés par des directives du fichier de configuration.

Les directives sont :

- **des directives simples**, composées du nom et des paramètres séparés par des espaces et terminées par un point-virgule (;).
- et **des directives de bloc**, de même structure qu'une directive simple, mais au lieu du point-virgule, se terminant par un ensemble d'instructions supplémentaires entourées d'accolades ({ et }).

Une directive de bloc peut contenir d'autres directives entre accolades, elle est alors appelée contexte (exemples : events, http, server, et location).

Les directives placées dans le fichier de configuration en dehors de tout contexte sont considérées comme étant dans le contexte principal.

Les directives events et http résident dans le contexte principal, le server dans http, et location dans server.

Le reste d'une ligne après un # est un commentaire.

Servir du contenu statique

Le serveur web a la tâche importante de distribuer des fichiers (des images ou des pages HTML statiques, par exemple).

Vous allez mettre en œuvre un exemple dans lequel, selon la requête, les fichiers seront servis à partir de différents répertoires locaux : /data/www (qui peut contenir des fichiers HTML) et

/data/images (qui contient des images).

Pour ce faire, il faut modifier le fichier de configuration et mettre en place un bloc server à l'intérieur du bloc http avec deux blocs location.

Créez d'abord le répertoire /data/www et placez-y un fichier index.html avec un contenu textuel quelconque, puis créez le répertoire /data/images et placez-y des images.

Puis, ouvrez le fichier de configuration. Le fichier de configuration par défaut inclut déjà plusieurs exemples du bloc server, la plupart commentés. Pour l'instant, commentez tous ces blocs et commencez un nouveau bloc serveur :

```
http {
    server {
    }
}
```

Le fichier de configuration peut généralement comprendre plusieurs blocs server qui se distinguent par les ports sur lesquels ils écoutent et par les noms des serveurs.

Dès que nginx a décidé quel serveur doit traiter une requête, il compare l'URI spécifié dans l'en-tête de la requête avec les paramètres des directives location définies dans le bloc server.

Ajoutez le bloc “location” suivant au bloc “server” :

```
location / {
    root /data/www;
}
```

Ce bloc location spécifie le préfixe “/” comparé à l'URI de la requête.

Pour les demandes concordantes, l'URI sera ajouté au chemin spécifié dans la directive root, c'est-à-dire à /data/www, pour former le chemin d'accès au fichier demandé sur le système de fichiers local.

Si plusieurs blocs location correspondent, nginx sélectionne celui qui a le préfixe le plus long.

Le bloc location ci-dessus fournit le préfixe le plus court, de longueur un, et ce n'est donc que si tous les autres blocs location ne fournissent pas de correspondance que ce bloc sera utilisé.

Ajoutez maintenant le deuxième bloc location :

```
location /images/ {
    root /data;
}
```

Il concordera avec les requêtes commençant par /images/ (location / concorde également avec ces requêtes, mais son préfixe est plus court).

La configuration résultante du bloc server devrait ressembler à ceci :

```
server {
```

```
    location / {
        root /data/www;
    }

    location /images/ {
        root /data;
    }
}
```

C'est déjà une configuration fonctionnelle d'un serveur qui écoute sur le port standard 80 et est accessible sur la machine locale à l'adresse http://localhost/.

En réponse aux requêtes dont l'URI commence par /images/, le serveur enverra des fichiers du répertoire /data/images.

Par exemple, en réponse à la requête http://localhost/images/example.png, nginx enverra le fichier /data/images/example.png.

Si ce fichier n'existe pas, nginx répondra en indiquant l'erreur 404.

Les demandes dont l'URI ne commence pas par /images/ seront transférées dans le répertoire /data/www.

Par exemple, en réponse à la requête http://localhost/some/example.html, nginx enverra le fichier /data/www/some/example.html.

To apply the new configuration, start nginx if it is not yet started or send the reload signal to the nginx's master process, by executing:

```
...@...:~$ nginx -s reload
```

In case something does not work as expected, you may try to find out the reason in access.log and error.log files in the directory /usr/local/nginx/logs or /var/log/nginx.

Mise en place d'un proxy simple

Mise en place du proxy FastCGI

Voir aussi

- **(en)** Beginner's Guide

Basé sur « Beginner's Guide » par nginx.org.

From:
https://doc.wikis.frapp.fr/ - **doc**

Permanent link:
https://doc.wikis.frapp.fr/doku.php?id=tutoriel:reseau:http:serveur:nginx:debutant:start

Last update: **2024/09/21 10:26**