



**NOUS ÉCLAIRONS.  
VOUS BRILLEZ.**

**420-W45-SF Installation de serveurs et  
de services**

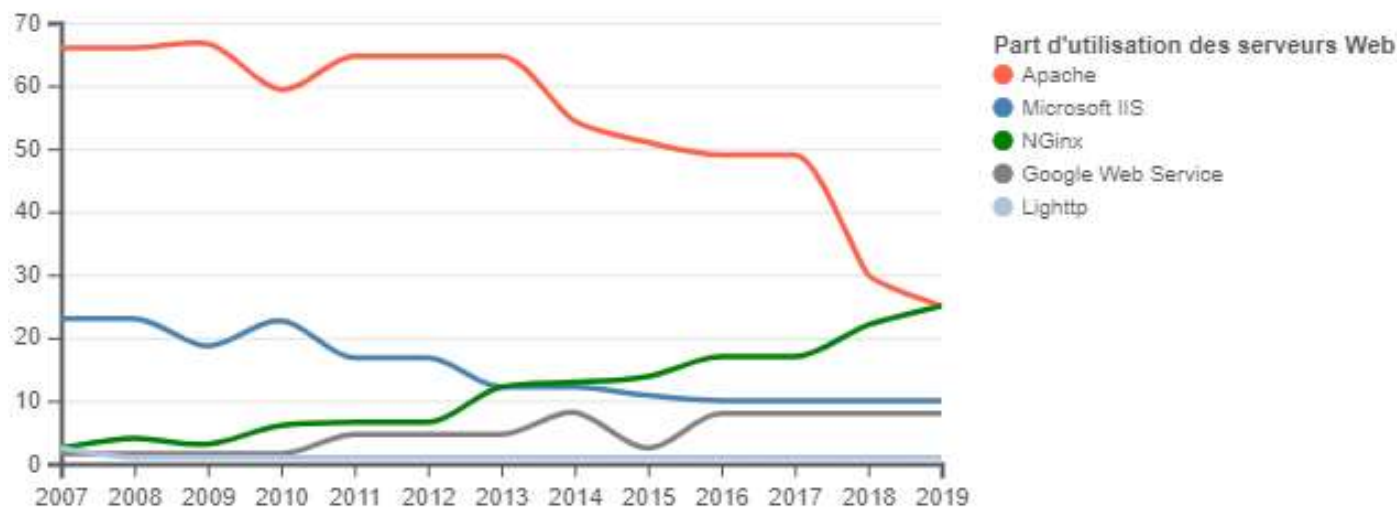
**Claude Roy et Jean-Pierre Duchesneau  
Serveur Web Nginx**



**FORMATION CONTINUE  
ET SERVICES AUX ENTREPRISES**

# Les sites Web et Nginx

- En avril 2019, il devient le serveur web le plus utilisé, à la fois en nombre de sites et en nombre d'ordinateurs<sup>b</sup>.



# Introduction

- nginx [engine x] est un serveur HTTP,
  - un serveur proxy inverse,
  - un serveur proxy de messagerie
  - un serveur proxy TCP/UDP générique,
- Un serveur Web occupé doit répondre à plusieurs milliers de requêtes simultanées.
- La plupart du temps nécessaire pour traiter chaque demande est passé à attendre que les données arrivent du réseau ou du disque.
- Le temps passé à traiter activement la demande est court par comparaison.

# Nginx

- **Serveur de type asynchrone :**

Apache est synchrone:  
1 processus par client.

- Chaque processus gère  $n$  clients
- Meilleure performance
- Moins de consommation mémoire
- Lorsqu'une demande ou une réponse (un événement) est prête à être servie, un processus de travail termine rapidement le traitement avant de revenir pour gérer l'événement suivant.

- **Structure modulaire (en compilation statique).**

Apache a des modules:  
dynamique.

- Nécessite de recompiler pour ajouter ou enlever un module.
- Surtout, NGINX vise à éviter le blocage sur les E/S réseau ou disque par une excellente cache mémoire.

# Processus maître (master)

- Deux processus: un maître et un travailleur.
  - Le maître effectue des tâches administratives telles que l'ouverture de sockets, la lecture de la configuration et le maintien en cours d'exécution des autres processus NGINX.

# Processus travailleur (worker)

- Les travailleurs font la plupart des gros travaux en recevant et en traitant les demandes.
- Certaines configurations utilisent des processus supplémentaires dédiés à la mise en cache.

# Privilèges

- Comme dans Apache, le processus maître s'exécute en tant que root afin qu'il puisse ouvrir des sockets pour tous les ports inférieurs à 1024.
- Les autres processus s'exécutent en tant qu'utilisateur moins privilégié.

# Nombres de processus

- Le nombre de processus de travail est configurable.
- Une bonne règle d'or consiste à exécuter autant de processus de travail que le système a de cœurs de processeur.



# Nombres de processus

- Debian et Ubuntu configurent NGINX de cette façon par défaut s'il est installé à partir d'un paquet.
- FreeBSD et RHEL utilisent par défaut un seul processus de travail.

# Fichier de configuration

- Le fonctionnement de nginx et de ses modules est déterminé dans le fichier de configuration.
- Fichier de configuration par défaut :  
`nginx.conf`.
- Pour lancer nginx avec un fichier de configuration personnalisé :  
`- nginx -c my_nginx.conf`

# Démarrage, arrêt et rechargement de la configuration

- Nginx peut être contrôlé en invoquant l'exécutable avec le paramètre `-s` :
  - `nginx -s signal`
    - *signal* peut être l'un des suivants:
      - **stop** - arrêt rapide
      - **quit** - arrêt normal
      - **reload** - recharger le fichier de configuration
      - **reopen** - rouvrir les fichiers journaux
  - Préférer la commande **systemctl** dans un environnement de production.
    - `systemctl status nginx`
    - `Systemctl start | stop | nginx`

# Informations nginx

- Sous Debian/Ubuntu

Nom du package	nginx
Path du daemon	/usr/sbin/nginx
Configuration root	/etc/nginx
Fichier de configuration	nginx.conf
Utilisateur par défaut	www-data

# Structure de nginx.conf

- nginx se compose de modules contrôlés par des directives spécifiées dans le fichier de configuration.
- Les directives sont divisées en directives simples et directives de bloc.

# Exemple de fichier de configuration

## Directive

Toute directive se termine par un ;

## Bloc :

events

Délimité par accolade

## Bloc :

http

Donc s'applique à tous ce qui concerne http.

## Bloc :

serveur

Serveur virtuel

```
# Utilisateur groupe
user nginx nginx;
# Nombre de clients par processus
events {
    worker_connections 1024;
}
# Serveur HTTP par défaut
http {
    include /etc/nginx/mime.types;
    index index.html;
    server {
        listen 80;
        server_name localhost;
        root /usr/share/nginx/html;
    }
}
```

Paramètres  
user group

Directive et paramètre.  
Nb. Client gérable par un processus.

Nous avons ici le fichier de configuration minimum par défaut.

# Directives

- Une directive simple se compose du nom et des paramètres séparés par des espaces et se termine par un point-virgule (;).
- Une directive de bloc a la même structure qu'une directive simple, mais au lieu du point-virgule, elle se termine par un ensemble d'instructions supplémentaires entourées d'accolades ({ et }).
- Si une directive de bloc peut avoir d'autres directives entre accolades, elle est appelée un contexte (exemples: events, http, server et location).

# Exemple

```
events {}  
http {  
    server {  
        server_name www.monsite.com;  
        root /var/www/admin.com;  
    }  
}
```

- Voici une configuration minimale.
- Le contexte le plus externe (appelé `main`) est implicite et configure la fonctionnalité de base.



# Events

- Les contextes `events` et `http` vivent dans `main`.
- `events` est un contexte obligatoire qui configure la gestion des connexions.
- Puisqu'il est vide dans cet exemple, les valeurs par défaut sont implicites.

# Valeurs par défauts de `events`

- Exécutez un processus de travail (utilisez le compte d'utilisateur non privilégié).
- Écoutez sur le port 80 s'il est démarré en tant que root ou sur le port 8000 sinon.
- Écrire les journaux dans `/var/log/nginx` (choisi au moment de la compilation).

# http

- Contiens toutes les directives relatives aux services de proxy Web et HTTP.

# server

- Définis des hôtes virtuels.
- Imbriqués dans `http`.
- Plusieurs contextes `server` dans `http` configureraient plusieurs hôtes virtuels.

# Vérification du fichier de configuration

- Il est possible (voir nécessaire) de vérifier le fichier de configuration avant de le recharger :

`- nginx -t`

```
# nginx -t
```

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
```

```
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Il existe une variante avec :

`nginx -T | more`

Vérifie la syntaxe est ajoute tous les include.

# Les modules de Nginx

```
nginx -V
nginx version: nginx/1.6.0
...
--with-http_ssl_module
--with-http_realip_module
--with-http_addition_module
--with-http_sub_module
--with-http_dav_module
--with-http_flv_module
--with-http_mp4_module
--with-http_gunzip_module
...
```

- De base (obligatoires)
- Standards (généralement intégrés)

Non affichés dans -V

Ceux-ci oui :

- Optionnels
- Tierces-parties

# Exemples de modules

Nom	Description	Type
HTTP core	Serveur HTTP de Nginx	Standard
Access	Contrôle d'accès	Standard
Auth Basic	Authentification de base	Standard
FastCGI	Gestion des scripts FastCGI (PHP, Python, Perl, etc.)	Standard
Gzip	Compression	Standard
Log	Gestion des logs	Standard
SSI	Gestion du SSI	Standard
Proxy	Gestion de Proxy	Optionnel
SSL	Gestion SSL	Optionnel
Mail	Serveur de mail	Optionnel
POP3	Serveur POP3	Optionnel
IMAP	Serveur IMAP	Optionnel
SMTP	Serveur SMTP	Optionnel
GeoIP	Géolocalisation de l'adresse IP	Tierce partie
Mongo	Bases de données MongoDB	Tierce partie
UnZip	Décompression à la volée	Tierce partie

# Les hôtes virtuels sur adresse IP/numéros de port

```
server {  
    listen 10.0.0.1:80 default;  
    server_name localhost;  
    root /usr/share/nginx/html/public;  
}  
  
server {  
    listen 8080;  
    server_name localhost;  
    root /usr/share/nginx/html/admin;  
}
```

Réponse  
par défaut.



# Les hôtes virtuels sur nom d'hôte

```
server {  
    listen 80;  
    server_name www.compta.masoc.com;  
    root /usr/share/nginx/html/public/compta;  
}  
server {  
    listen 80;  
    server_name www.marketing.masoc.com;  
    root /usr/share/nginx/html/public/mkg;  
}
```

# Les filtres de traitement

- Bloc de type location, dans un bloc **server**
- Associe des traitements aux URIs  
(Uniform Ressource Identifier)  
demandé par le client
- S'applique à tous ou partie de l'URI
- Correspond à un et un seul URI ou à un ensemble
- Syntaxe souple et puissante :
  - Expression régulières de Perl

# Les filtres de traitement

- Syntaxe :  
location Expression\_Selection {  
Directives  
}

Arborescence  
de :

```
./
./compta/
./compta/cptfra.html
./rh/
./rh/drhappli.html
./index.html
./gesttemps/
./gesttemps/compta/
./gesttemps/compta/conges/
./gesttemps/compta/conges/index.html
./gesttemps/conges/
./gesttemps/conges/index.html
./gesttemps/rh/
./gesttemps/rh/conges/
./gesttemps/rh/conges/index.html
[root@phases7 intranet1# ls -l
```

Exactement la chaîne

Chaîne débutant par

Tilde = Expression régulière

- Indépendante de la case.
- Peut importe ou.

```
server {
    listen      80;
    # Un serveur virtuel pour tous les noms d'hôtes de .intranet
    server_name *.intranet;
    root _/var/nginx/intranet/;
    # Accueil de l'intranet
    location = / {
        index index.html;
    }
    # Accueil de la compta
    location /compta {
        index cptfra.html;
    }
    # Accueil de la drh
    location /rh {
        index drhappli.html;
    }
    # Accueil demandes de congés (dans le dossier gesttemps)
    location ~* congés {
        root _/var/nginx/intranet/gesttemps;
        index index.html;
    }
}
```

# Contrôle d'accès utilisateur

- Contrôle par adresse IP -> Module de base **ngx\_http\_access**
- Autorise ou interdit l'accès
- Directives allow et deny

```
server {  
    listen 80;  
    server_name drh.soc.fr;  
    root /usr/share/nginx/html/intra/;  
    allow 10.0.1.0/24;  
    deny all;  
}
```

# Contrôle d'accès utilisateur

- Contrôle par authentification locale -> Module de base **ngx\_http\_auth\_basic**
- Utilise une base de compte locale htpasswd
- Directives auth\_basic et auth\_basic\_user\_file



# Contrôle d'accès utilisateur

- Contrôle par authentification LDAP
- Via PAM (Pluggable Authentication Module)
- Ou module tierce partie ngx\_http\_auth\_pam

# Utilisation de SSL

```
server {  
    listen 443 ssl ;  
    server_name www.masoc.com;  
    ssl_certificate /usr/share/nginx/ssl/cert.pem;  
    ssl_certificate_key /usr/share/nginx/ssl/cert.key;  
    root /usr/share/nginx/html/ssl/compta;  
}
```

# Site Web de référence

- <https://nginx.org/>
- [https://hub.docker.com/\\_/nginx](https://hub.docker.com/_/nginx)