

US-RS7G

Sécurité des réseaux

Le Rebours Paris XIII

Découverte & Compréhension du serveur Web Nginx sous Debian

Durée : 2 heures

TP du 18 novembre 2025

Pré-requis

Pour réaliser ce TP, il est impératif de maîtriser les commandes de base **Linux**. De plus, je vous recommande fortement de prendre un peu de votre temps pour suivre le tutoriel **Vim** accessible en lançant la commande `'vimtutor'`. **Vim** (Vi IMproved) est un éditeur de texte directement inspiré de **vi** (un éditeur très répandu sur les systèmes d'exploitation de type **Unix**), dont il est le clone le plus populaire. **Vim** se différencie de la plupart des autres éditeurs par son fonctionnement modal, hérité de **vi**. En effet, il possède trois modes de base : le mode *normal* ou *commande* (lorsque Vim démarre), le mode *insertion*, et le mode *exécution*. La touche `<ESC>` permet de revenir au mode normal.



Introduction

Dans ce TP, nous allons explorer le serveur Web **Nginx** sous un système d'exploitation **Debian**. **Nginx** est l'un des serveurs web les plus populaires et est particulièrement apprécié pour sa performance et sa simplicité.

Objectifs

- Installer **Nginx** sur **Debian**
- Tester le fonctionnement d'un serveur Web
- Configurer des sites simples & Comprendre les blocs de configuration
- Explorer les fichiers journaux (logs) et les configurations par défaut

→ Merci de lire le TP *en entier* au moins une fois avant de commencer !

1. [5 points] Installation de Nginx

- Démarrez votre machine **Debian**.
- Ouvrez un terminal et exécutez les commandes suivantes pour installer **Nginx** et divers outils de diagnostic :

```
1 sudo apt update && sudo apt upgrade
2 sudo apt purge apache2 lighttpd
3 sudo apt install vim terminator curl wget lynx net-tools telnet nginx
```

- (c) Une fois l'installation terminée, vérifiez que Nginx est en cours d'exécution avec :

```
1 sudo systemctl start nginx
2 sudo systemctl status nginx
3 sudo netstat -ntlp | grep nginx
4 sudo ss -ptl4 | grep nginx # process, TCP, listening, IPv4
```

Que constatez-vous (port(s) ouvert(s) et service associé) ?

- (d) Testez le fonctionnement du serveur Web Nginx avec différents outils en CLI :

```
1 cd /tmp
2 sudo wget 127.0.0.1
3 sudo cat index.html
4 sudo lynx index.html
5 sudo curl http://127.0.0.1
6 sudo telnet 127.0.0.1 80
7 HEAD / HTTP/1.0
8 sudo telnet 127.0.0.1 80
9 GET / HTTP/1.0 # puis GET / HTTP/1.1
```

Quelle est la différence entre les méthodes HEAD & GET ?

Quelle est la différence entre HTTP/1.0 & HTTP/1.1 ?

2. [5 points] Configuration de base

- (a) Lancez votre navigateur Web et saisissez l'adresse `http://localhost`.
Vous devriez voir la page par défaut de Nginx.

- (b) Naviguez dans le répertoire de configuration de Nginx :

```
1 cd /etc/nginx/sites-available/
```

- (c) Créez un nouveau fichier de configuration pour servir vos sites appelé 'sites' :

```
1 sudo vi sites
```

- (d) Ajoutez le contenu suivant :

```
1 server {
2     listen 8080;
3     server_name monsite1.backend;
4     root /var/www/monsite1;
5     index index.html;
6 }
7
8 server {
9     listen 8081;
10    server_name monsite2.backend;
11    root /var/www/monsite2;
12    index index.html;
13 }
```

- (e) Créez les répertoires '/var/www/monsite1' et '/var/www/monsite2' avec 'mkdir'.

3. [5 points] Configuration d'un site de base

- (a) Créez les fichiers HTML pour vos sites en exécutant les commandes suivantes :

```
1 echo '<center><h1>Bienvenue sur mon magnifique <b>Site1</b> Nginx !</h1></center>' >
2 /var/www/monsite1/index.html
3 echo '<center><h1>Bienvenue sur mon superbe <u>Site2</u> Nginx !</h1></center>' >
4 /var/www/monsite2/index.html
```

- (b) Ouvrez à nouveau `http://localhost` dans votre navigateur.
Quels changements observez-vous ?

4. [5 points] Activer les sites et tester

- (a) Activez les sites en créant le lien symbolique :

```
1 sudo ln -s /etc/nginx/sites-available/sites /etc/nginx/sites-enabled/
```

(b) Testez puis rechargez Nginx :

```
1 sudo nginx -t
2 sudo nginx -s reload
```

(c) Vérifiez à nouveau les ports ouverts :

```
1 sudo netstat -ntlp
2 sudo ss -ptl4 | grep nginx
```

Que constatez-vous ?

(d) Éditez votre fichier ‘/etc/hosts’ pour associer ‘monsite1.backend’ et ‘monsite2.backend’ à ‘127.0.0.1’.

(e) Dans votre navigateur, naviguez vers <http://monsite1.backend:8080> et <http://monsite2.backend:8081> pour voir vos nouveaux sites web !

5. [5 points] Exploration des configurations et fichiers journaux

(a) Explorez la configuration par défaut de Nginx et le site par défaut :

```
1 sudo vi /etc/nginx/nginx.conf
2 sudo vi /etc/nginx/sites-available/default
```

Quelles sont les principales sections que vous observez ? Expliquez.

Il faut donner le moins d’information possible à un éventuel pirate. Comment faire pour ne pas afficher la version du serveur Nginx utilisé ?

(b) Accédez aux fichiers journaux de Nginx :

```
1 sudo tail -n 20 /var/log/nginx/access.log
2 sudo tail -n 20 /var/log/nginx/error.log
3 sudo tail -f /var/log/nginx/access.log
4 # Generez des acces avec votre navigateur
```

Que pouvez-vous déduire des entrées inscrites dans les journaux ?

6. [5 points] Discussion

- (a) Expliquez pourquoi il est nécessaire de modifier le fichier ‘/etc/hosts’.
- (b) Quelles sont les alternatives pour éviter cette étape ?
- (c) Pourquoi est-il essentiel de surveiller les fichiers journaux (logs) de Nginx ?
- (d) Comment pouvez-vous sécuriser davantage votre installation de Nginx ?
- (e) Quels autres serveurs web connaissez-vous ?

Conclusion

Félicitations ! Vous avez maintenant une compréhension de base du fonctionnement de Nginx sur Debian. Il existe de nombreuses fonctionnalités avancées et configurations possibles avec Nginx. Je vous encourage à explorer la documentation officielle pour en savoir plus^[1].

Question:	1	2	3	4	5	6	Total
Points:	5	5	5	5	5	5	30
Score:							

1. <https://nginx.org/en/docs/>