US-RS7G
Sécurité des réseaux
Le Rebours Paris XIII

Configuration d'un ReverseProxy `Nginx`
avec HTTPS & Load Balancing sous `Debian`
Durée : **2 heures**
TP du **9 décembre 2025**

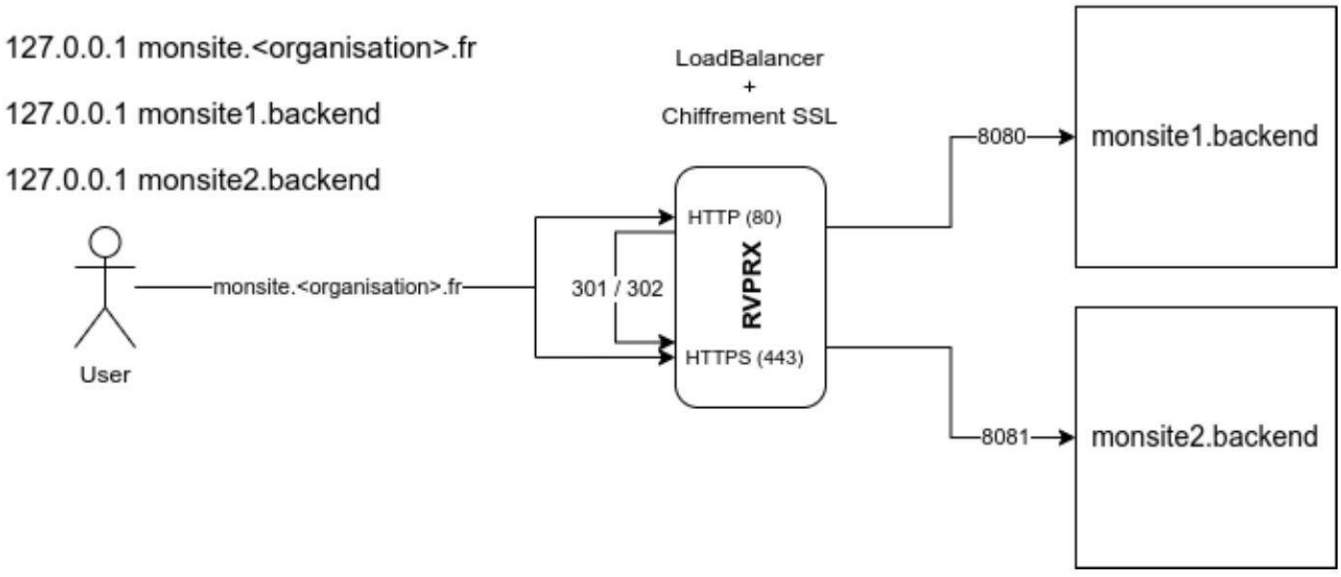
Introduction

`Nginx` est largement utilisé comme *ReverseProxy* (RVPRX) pour rediriger le trafic vers plusieurs serveurs ou applications. Dans ce TP, nous explorerons comment configurer `Nginx` pour servir deux sites web HTTP via un RVPRX HTTPS tout en équilibrant la charge entre eux.

Objectifs

- Installer `Nginx` sur `Debian`.
- Configurer `Nginx` en tant que ReverseProxy HTTPS pour deux sites HTTP.
- Utiliser la directive 'upstream' pour le load balancing.
- Tester et comprendre différentes méthodes de load balancing.
- Protéger les accès grâce à l'authentification 'AuthBasic'

/etc/hosts



1. [2 points] Installation de `Nginx`
   (a) Mettez à jour les paquets de votre système `Debian` et installez des outils de diagnostique :

```
sudo apt update && sudo apt upgrade
sudo apt install vim terminator curl wget lynx net-tools telnet
```

   (b) Installez Nginx :

```
sudo apt install nginx
```

2. [10 points] Configuration du Load Balancing en HTTPS
   Dans le cadre de ce TP, nous utiliserons un certificat SSL auto-signé.
   (a) Créez les répertoires :

```
sudo mkdir -p /etc/nginx/ssl /etc/ssl/private
```

   (b) Générez le bi-clef RSA et le certificat auto-signé avec, par exemple, les informations suivantes :
   FR – Ile de France – PARIS – Le Rebours Paris XIII – CYBER – *monsite.lerebours.fr* – *christophe.maudoux@cnam.fr*

```
sudo openssl req -x509 -newkey rsa:2048 -keyout /etc/ssl/private/nginx-selfsigned.key -
    ↪ out /etc/nginx/ssl/nginx-selfsigned.crt -nodes -days 365
```

   (c) Créez un fichier ‘loadbalancer.conf‘ contenant un bloc ‘upstream‘ pour gérer vos serveurs backend :

```
sudo vi /etc/nginx/conf.d/loadbalancer.conf
```

   (d) Ajoutez la configuration suivante :

```
upstream upstream_lerebours {
   server monsite1.backend:8080;
   server monsite2.backend:8081;
}

server {
   listen 443 ssl;
   server_name monsite.lerebours.fr;

   # include /etc/nginx/security/restrict;
   add_header Organization 'lerebours';
   ssl_certificate /etc/nginx/ssl/nginx-selfsigned.crt;
   ssl_certificate_key /etc/ssl/private/nginx-selfsigned.key;

   location / {
      proxy_pass http://upstream_lerebours;
      proxy_set_header Host $host;
      proxy_set_header X-Real-IP $remote_addr;
   }
}
```

   (e) Testez puis redémarrez `Nginx` pour prendre en compte les modifications :

```
sudo nginx -t
sudo systemctl restart nginx
```

   (f) Affichez toute la configuration `Nginx` :

```
sudo nginx -T
```

   Expliquez pourquoi le fichier ‘loadbalancer.conf‘ est bien pris en compte par `Nginx`.
   (g) Éditez votre fichier ‘/etc/hosts‘ pour associer ‘monsite.lerebours.fr‘ à ‘127.0.0.1‘.

(h) Dans votre navigateur, naviguez vers `https://monsite.lerebours.fr` et faites 'F5' plusieurs fois. Que constatez-vous ?

(i) Expliquez l'exception de sécurité levée par le navigateur.

(j) Affichez dans le navigateur les caractéristiques de votre certificat SSL et faites une capture d'écran. Quelle est sa date d'expiration ? Pour quel sujet a-t-il été délivré et quelle autorité de certification (AC) l'a signé ? Quelle est la taille de la clef publique ?

3. [6 points] Exploration des méthodes de *Load Balancing*

(a) Éditez le fichier 'loadbalancer.conf' pour que la charge de travail soit distribuée de manière *non uniforme* entre les serveurs :

```
upstream upstream_lerebours {
   server monsite1.backend:8080 weight=3;
   server monsite2.backend:8081;
}
```

(b) Rechargez puis testez la configuration et notez les différences.

(c) Éditez le fichier 'loadbalancer.conf' pour utiliser la méthode 'least_conn' :

```
upstream upstream_lerebours {
   least_conn;
   server monsite1.backend:8080;
   server monsite2.backend:8081;
}
```

(d) Testez cette configuration et observez le comportement.

(e) De retour dans le fichier de configuration, explorez la méthode 'ip_hash' pour une session persistante basée sur l'adresse IP du client.

```
upstream upstream_lerebours {
   ip_hash;
   server monsite1.backend:8080;
   server monsite2.backend:8081;
}
```

(f) Testez à nouveau la configuration et notez les différences. Supprimez la ligne 'ip_hash ;' puis rechargez la configuration `Nginx` pour revenir à une répartition équilibrée de la charge (méthode 'Round-Robin' par défaut).

4. [5 points] Mise en place de la redirection HTTP vers HTTPS

(a) Éditez le fichier 'loadbalancer.conf' et ajouter le *BlockServer* (BS) suivant afin de rediriger les requêtes HTTP vers le BS HTTPS :

```
## HTTP redirection to HTTPS
server {
   listen 80;
   server_name monsite.lerebours.fr;
   return 302 https://$host$request_uri;
   #return 301 https://$host$request_uri;
}
```

$host & $request_uri sont des variables d'environnement positionnées *automatiquement* par `Nginx`. Il en existe plein d'autres[1] !!!

(b) Testez puis rechargez la configuration `Nginx`.

(c) Dans votre navigateur, ouvrez la console de debug avec 'F12' dans l'onglet 'Réseau' et naviguez vers `http://monsite.lerebours.fr`. Que constatez-vous ? Expliquez les différentes requêtes et étapes correspondantes (code HTTP, entêtes transmis, ...).

1. `http://nginx.org/en/docs/varindex.html`

(d) Expliquez la différence de comportement du navigateur entre la redirection 301 et 302 en modifiant votre BS de redirection.

(e) A votre avis, quel code de redirection est le plus pertinent à utiliser dans ce cas ?

5. [10 points] Mise en place de l'authentification 'AuthBasic' et du filtrage par adresse IP

(a) Dé-commentez la directive 'include'.

(b) Créez le fichier '/etc/nginx/security/restrict' et ajouter le contenu suivant :

```
## AuthBasic & Filtrage IP
allow 1.2.3.4;
deny all;
auth_basic "Access restricted";
auth_basic_user_file /etc/nginx/security/.htpasswd;
```

— 'auth_basic' définit un message personnalisé qui apparaîtra dans la boîte de dialogue d'authentification.

— 'auth_basic_user_file' indique le chemin vers le fichier de mots de passe créé à l'étape précédente.

(c) Exécutez les commandes suivantes pour créer un fichier de mots de passe :

```
sudo apt install apache2-utils
htpasswd -c /etc/nginx/security/.htpasswd lerebours
```

et entrez un mot de passe lorsque vous y êtes invité.

(d) Testez puis rechargez la configuration `Nginx`.

(e) Dans votre navigateur, naviguez vers `https://monsite.lerebours.fr`. Que constatez-vous ?

(f) Modifiez la configuration du filtrage afin de permettre l'accès à `https://monsite.lerebours.fr` et testez à nouveau.

(g) A présent, testez l'accès avec la commande suivante :

```
curl -kv https://monsite.lerebours.fr
```

(h) Quels sont le code HTTP et l'entête d'authenfication retournés par le serveur ? Fournissez une capture d'écran et expliquez.

(i) Testez à nouveau l'accès au serveur avec la commande suivante :

```
curl -kvu 'lerebours:<mot de passe>' https://monsite.lerebours.fr
```

(j) Expliquez les options -k, -v et -u à l'aide de la commande 'man curl'.

(k) Quel est l'entête utilisé pour transmettre l'identifiant et le mot de passe *AuthBasic* au serveur ? Affichez son contenu. Celui-ci est-il chiffré ?

6. [7 points] Discussion

(a) Quels avantages apporte de faire "porter" le HTTPS par le ReverseProxy ?

(b) Pourquoi utiliser `Nginx` en tant que LoadBalancer ?

(c) Quels sont les avantages et inconvénients des différentes méthodes de load balancing que nous avons explorées ?

(d) Comment éviter l'exception de sécurité due au certificat ?

(e) Expliquez la directive 'proxy_set_header'.

(f) Quelle est l'utilité de la directive 'add_header' ? Fournissez une capture d'écran et expliquez en analysant la réponse du serveur à l'aide de la console du navigateur.

(g) Pourquoi est-il recommandé d'utiliser la méthode d'authentification basique en combinaison avec HTTPS ?

Conclusion

Vous avez maintenant une compréhension approfondie de la manière dont `Nginx` peut être configuré comme ReverseProxy avec capacité de load balancing en HTTPS et abordé l'authentification basique. Cette connaissance est essentielle pour assurer sécurité, haute disponibilité et performance des applications Web dans un environnement de production.

Question:	1	2	3	4	5	6	Total
Points:	2	10	6	5	10	7	40
Score:							