

Principe de fonctionnement HAProxy

HAProxy reçoit le trafic entrant via ses “**frontends**”, Une fois le trafic reçu, **HAProxy** le distribue vers des “**backends**”. En cas de panne d'un serveur, HAProxy **redirige automatiquement** le trafic vers les serveurs restants, assurant ainsi une continuité de service.

La configuration de **HAProxy** se fait via le fichier de configuration **haproxy.cfg** situé à **/etc/haproxy/haproxy.cfg**

Il est constitué de sections

- **Section global**

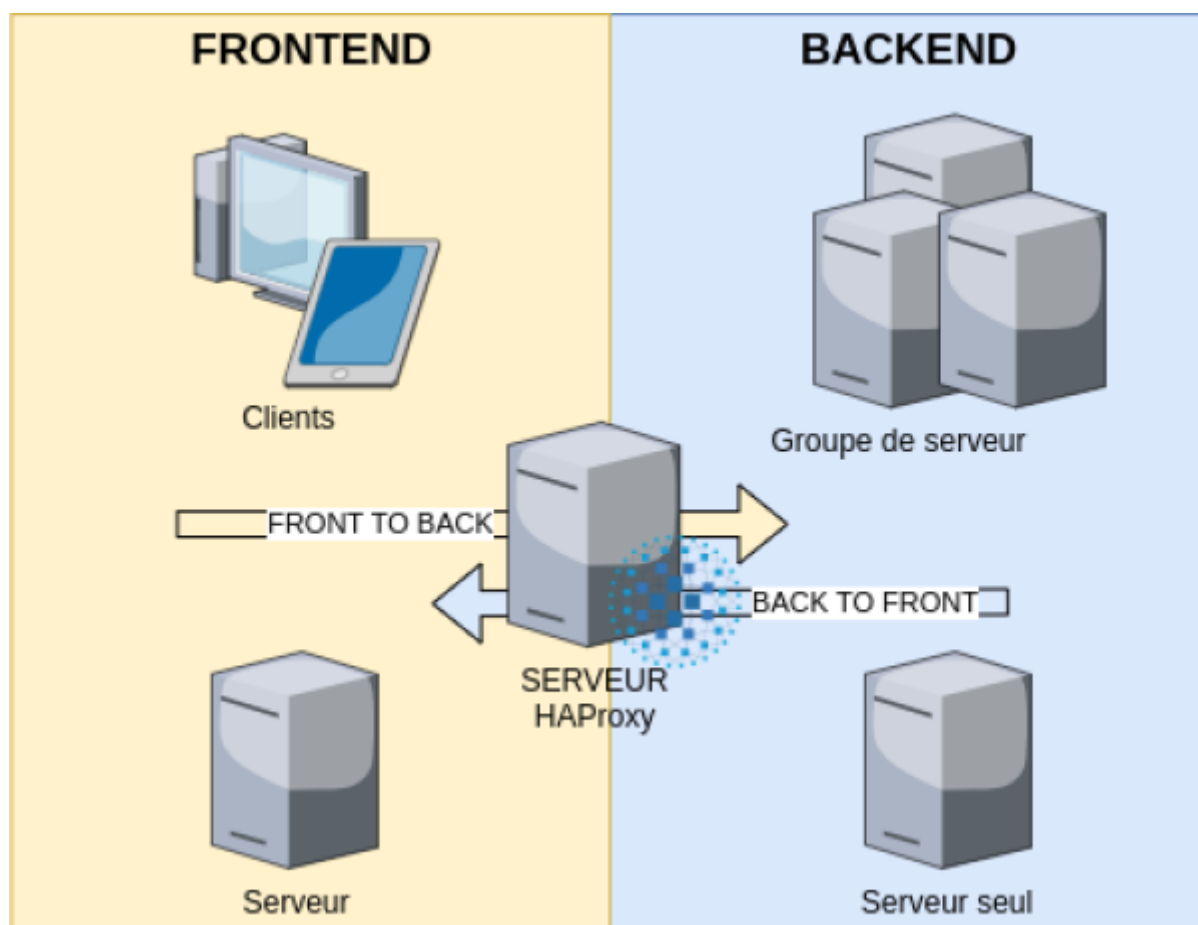
Cette section contient les **paramètres globaux** affectant le processus principal d'HAProxy, comme la **journalisation** et les **droits d'accès**.

- **Section defaults**

Définissez ici les **valeurs par défaut** pour les autres sections, évitant ainsi la répétition de paramètres communs

- Section **frontend**
- Section **backend**

Le Frontend et le Backend



Frontend

Un **frontend** dans **HAProxy** définit comment les **connexions entrantes** sont gérées. C'est le point d'entrée pour le trafic client et spécifie :

- Les adresses **IP** et ports sur lesquels HAProxy **écoute**
- Le mode de fonctionnement (**HTTP ou TCP**)
- Les règles de **roulage** des requêtes

Par exemple, un frontend typique pourrait ressembler à ceci :

```
frontend front_webrowsers
  bind *:443
  default_backend backend_webrowsers
  option forwardfor
```

Dans cet exemple le frontend écoute sur le port 443 et dirige le trafic vers un backend nommé **"backend_webrowsers"**

Backend

Un **backend** dans **HAProxy** définit un groupe de serveurs qui traitent les **requêtes reçues** par le

frontend. Il comprend :

- La liste des serveurs et leurs **adresses**
- L'algorithme d'**équilibrage de charge** à utiliser
- Les **options** de configuration spécifiques aux serveurs

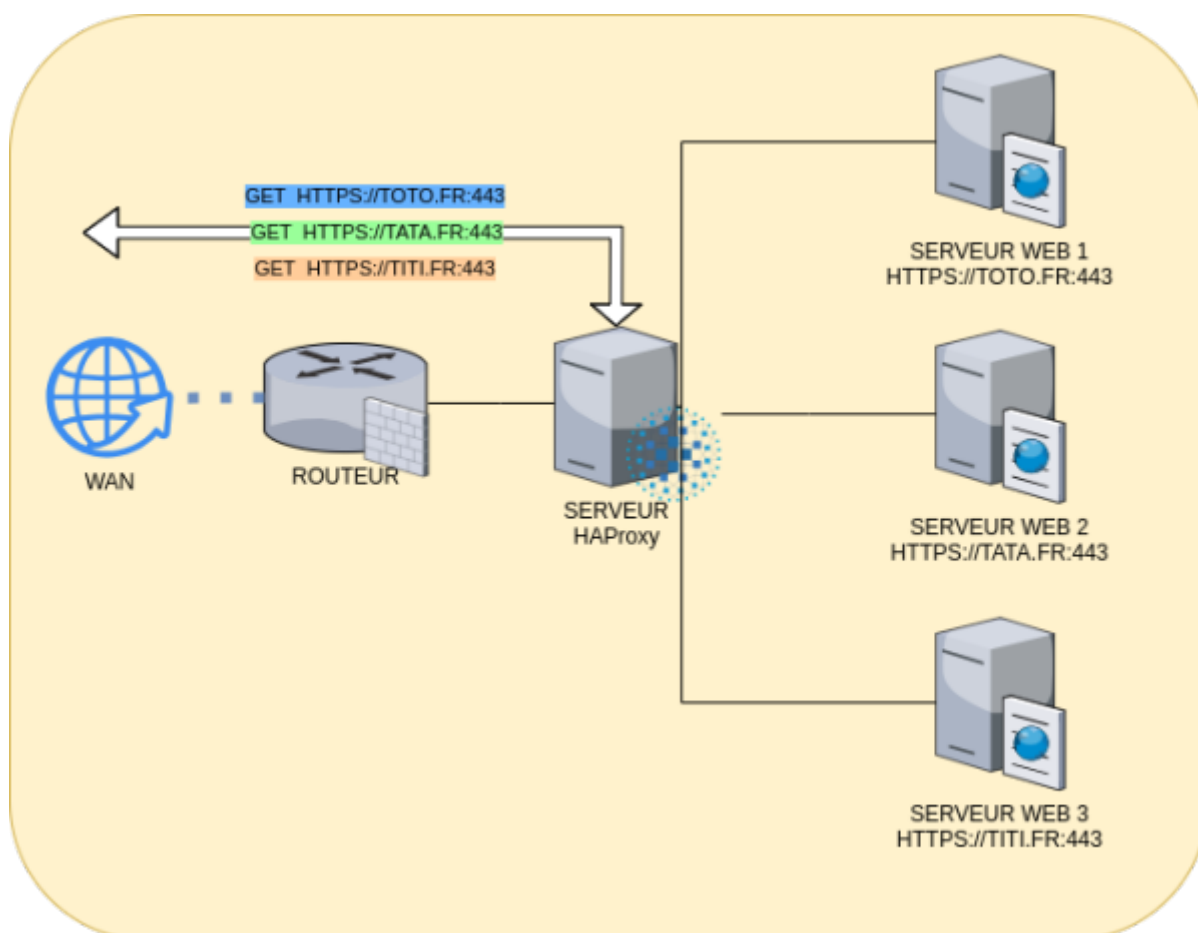
Voici un exemple de configuration de backend :

```
backend backend_webservers
  balance roundrobin
  server srv-web-1 192.168.1.10:443 check
  server srv-web-2 192.168.1.11:443 check
```

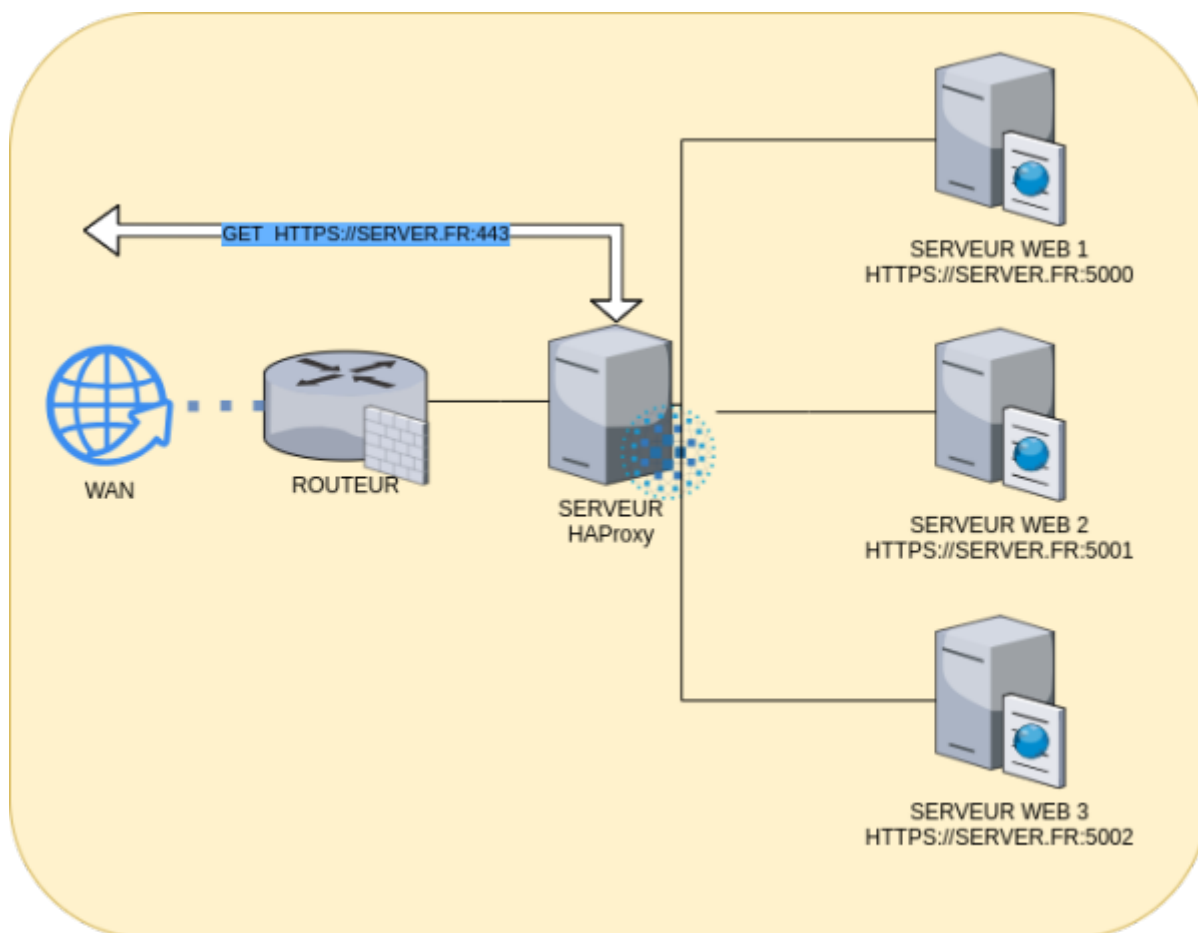
Ce backend utilise l'algorithme **roundrobin** pour distribuer les requêtes entre deux serveurs web.

Exemples simples d'utilisation

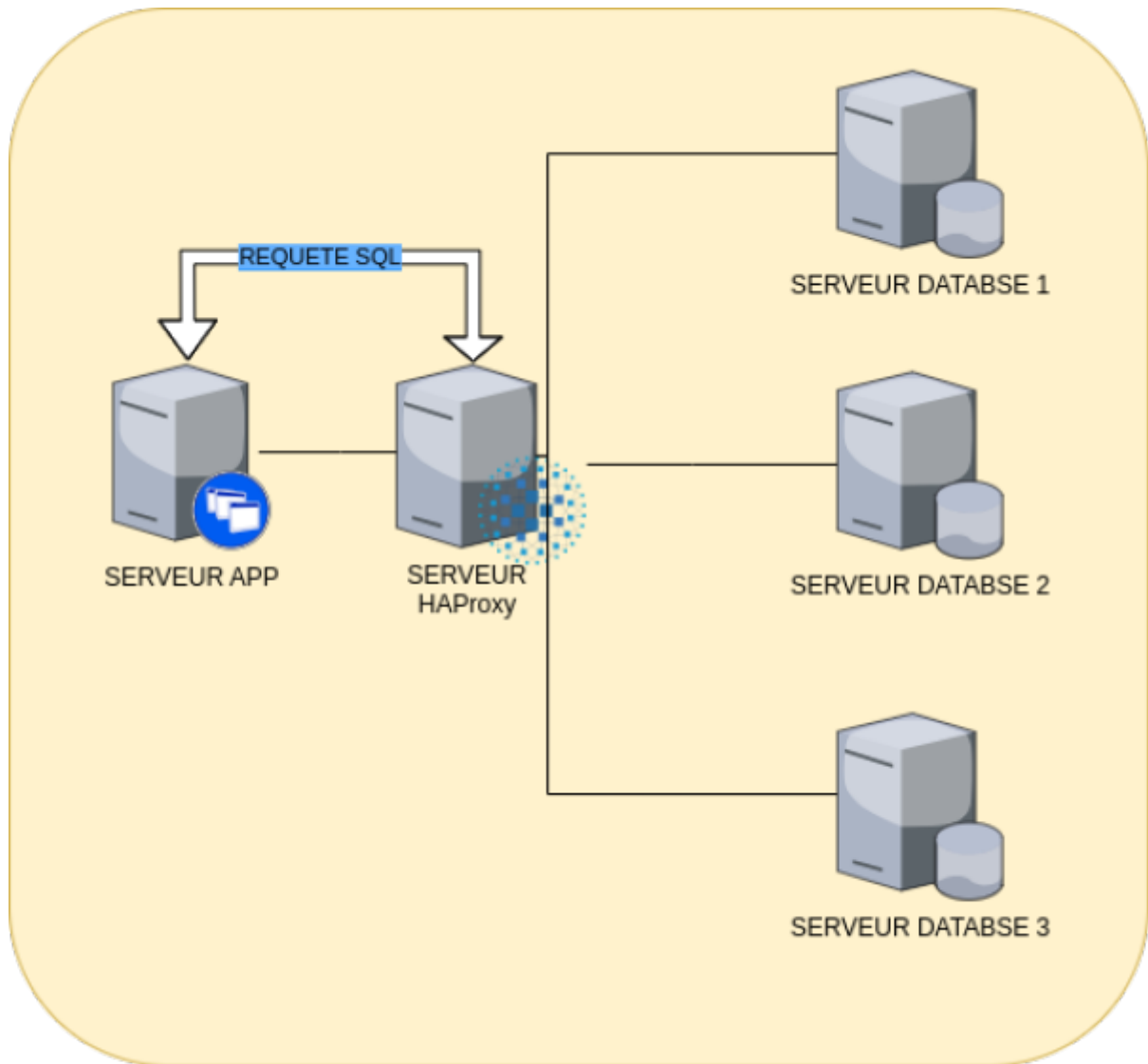
URL Forwarding



Load balncing WEB



Load balancing TLS



From:
<https://www.hugo-mattaliano.fr/wiki/> - Gogo Wiki

Permanent link:
https://www.hugo-mattaliano.fr/wiki/doku.php?id=wiki:proxy:haproxy:fonctionnement_frontend_backend&rev=1740223890

Last update: 2025/02/22 12:31

