Bug, port non identifiable Elixir, serveur QLM (Ubuntu): (30/1/24) (1)

Problème

> Erreur rencontré dans /var/log/nginx/error.log
> upstream prematurely closed connection while reading response header fr
> om upstream,
> client: 102.16.82.18, server: focicom.re, request: "GET / HTTP/1.1",
> upstream: "http://127.0.0.1:4010/", host: "focicom.re

- `upstream prematurely closed connection while reading response header from upstream` : Cette partie du message d'erreur indique qu'il y a eu un problème avec la connexion entre Nginx et le serveur en amont (dans ce cas, un serveur fonctionnant à `http://127.0.0.1:4010/`). La connexion a été fermée de manière inattendue avant que l'en-tête de réponse ne puisse être entièrement lu.
- `client: 102.16.82.18` : L'adresse IP du client qui a effectué la demande.
- `server: focicom.re` : Le nom du serveur où l'erreur s'est produite.
- `request: "GET / HTTP/1.1"` : Le type de requête HTTP qui a été effectué. Dans ce cas, c'est une requête GET pour le chemin racine ("/") en utilisant HTTP/1.1.
- `upstream: "http://127.0.0.1:4010/"` : Le serveur en amont auquel Nginx transmet les demandes. Dans ce cas, c'est un serveur fonctionnant à l'adresse `http://127.0.0.1:4010/`.
- `host: "focicom.re"` : L'en-tête d'hôte de la demande du client.

Causes

1. **Plantage de l'application :** Si l'application en amont plante ou rencontre une erreur, elle peut fermer la connexion de manière inattendue.
2. **Épuisement des ressources :** Le serveur en amont peut manquer de ressources (par exemple, mémoire, descripteurs de fichiers), entraînant une

fermeture prématurée de la connexion.

3. **Délais d'attente :** S'il existe des configurations de délai d'attente, la connexion peut être fermée si la réponse met trop de temps à être générée.
4. **Problèmes de réseau :** Des problèmes avec le réseau entre Nginx et le serveur en amont pourraient entraîner une fermeture prématurée des connexions.

Analyse personnel:

Bug: Il semble que l'application elixir (dans ce cas le site focicom) utilisant le port 4010 s'est crashé pendant un petit moment puis s'est continué a fonctionné normalement (or le processus d'elixir {beam.smp} continuait de fonctionner sur ce même port).

Ainsi, lorsque le site décida de fonctionner normalement en utilisant le port 4010, Nginx ne peut pas le faire fonctionner sur ce même port car le processus d'elixir {beam.smp} l'utilise déjà

Solution

Processus de fonctionnement des sites: focicom, fracomex, bbmay, cheinmalt

> Client va entrer l'url des sites: ex: focicom.re → DNS: focicom.re:135-125-87.eu:8006:serveur QLM → Nginx (dans serveur QLM) → reverse proxy (port 80 et 443 redirigé en 4030) → site local

> 💡Difference entre proxy et proxy inverse
>
> - **Proxy :** Agit au nom du client, transmettant les demandes au serveur.
> - **Proxy inverse :** Agit au nom du serveur, recevant les demandes et les dirigeant vers le serveur backend approprié.

Reflèxe:

- En cas d'erreur, vérifier le log de Ngnix dans /var/log/nginx/error.log

- Assigner au site web un autre port (4030, le site fonctionne sans aucun problème)

- Editer le fichier de configuration de Nginx pour changer le port, fichier contenu dans

```
/etc/nginx/sites-available/mgbi.conf
```

```
# UPSTREAMS

##### Voici les lignes à éditer pour changer les ports de Nginx
upstream cheinmalt{
      server 127.0.0.1:8069;
}

upstream bbmay{
       server 127.0.0.1:4000;
}

upstream focicom{
       server 127.0.0.1:4030;
}

upstream fracomex{
       server 127.0.0.1:4020;
}
```

- Editer le fichier de configuration du site en question (dans notre cas focicom.re) pour changer son port de lancement, fichier contenu dans

```
/home/mgbi/elixir/focicom/master/focicom/config/dev.exs
```

```
   ##### Voici la ligne à modifier pour changer le port du site focic
om.re
  http: [ip: {0, 0, 0, 0}, port: 4030],
  check_origin: false,
  code_reloader: true,
  debug_errors: true,
  secret_key_base: "a47qN/Cgy6ikhBek4f55bQWq42gjbWE30sVVx
2PgLzbsn9IYIFXUrq9/HL0Dq9r8",
```

```
  watchers: [
    # Start the esbuild watcher by calling Esbuild.install_and_run(:def
ault, args)
    esbuild: {Esbuild, :install_and_run, [:default, ~w(--sourcemap=inli
ne --watch --loader:.woff=file --loader:.woff2=file --loader:.eot=fil
e --loader:.woff2=file --loader:.svg=file --loader:.ttf=file)]}
  ]
```

- Kill le processus utilisant le port 4010, puis relance du site

```
#Liste les ports ouverts et les processus (à l'aide de leurs PID: PROCES
S ID) utilisant ces ports
netstat -tulpn

#Liste tous les processus active
ps -aux
#Liste un processus avec un nom spécifique
ps -aux | grep "<nom_du_processus ou PID>"

#Pour killer un processus avec son PID
kill <PID>

#Pour forcer le kill d'un processus
kill -9 <PID>

#Pour vérifier si le processus n'est plus présent
Apres la commande kill, faire echo $?
#Ou bien, rechercher si le PID utilise toujours ce port
netstat -tulpn
#Ou bien, liste les processus si le processus est toujours présent
ps -aux
```