

Enchère Hostein Québec

Guide de gestion serveur

Table of contents

SSH	2
Apache	3
Nodejs	7
PM2	8

SSH

La gestion du serveur websocket nécessite un accès SSH root. Celle-ci peut se faire de différente manière, mais les panneaux admin inclus habituellement un bouton pour ouvrir une console dans le navigateur.

Apache

Qu'est-ce?

Apache est un proxy pour pointer toute requête entrant vers sa bonne ressource (`enchereholsteinquebec.com` -> `/home/encheres/www`). On peut également ajouter des règles plus précises pour des routes particulières (`enchereholsteinquebec.com/api/ws` -> Port `9002`).

Objectif

Notre objectif pour le fonctionnement du websocket est d'ajouter un lien entre la route `/api/ws` et le port `9002`. Il faut également s'assurer que les modifications ne sont pas écrasées par le c-panel.

La configuration désirée se résume au suivant:

Domaine	Path	Port	Resource
enchereholsteinquebec.com	*	80	/home/encheres/www
enchereholsteinquebec.com	*	443	/home/encheres/www
enchereholsteinquebec.com	/api/ws	443	Port 9002
staging-kz.enchereholsteinquebec.com	*	80	/home/encheres/domains/staging-kz.enchereholsteinquebec.com
staging-kz.enchereholsteinquebec.com	*	443	/home/encheres/domains/staging-kz.enchereholsteinquebec.com
staging-kz.enchereholsteinquebec.com	/api/ws	443	Port 9002

Virtual Host

Chaque règle dans apaches est définis par un Virtual Host. Celles-ci sont défini par un fichier de configuration qui sera auto-géré par le C-panel. Ce fichier peut être configuré différemment d'un serveur à l'autre, alors la première chose qu'il faut faire est une analyse.

Emplacement

L'endroit de ce fichier peut varier d'un serveur à un autre, alors la première étape est de l'identifier. Voici quelques d'exemples connu:

```
/etc/apache2/conf/httpd.conf # path standard
/usr/local/apps/apache2/etc/conf.d/webuzoVH.conf # path likuid
```

Structure

Dans le fichier de configuration, on retrouve plusieurs Virtual Host. Le suivant est un exemple qui est automatiquement généré par likuid.

```
<VirtualHost 67.43.227.82:443> # Port 443 (https)
    ServerName encheresholsteinquebec.com # Domaine primaire
    ServerAlias www.encheresholsteinquebec.com
mail.encheresholsteinquebec.com # Domaines secondaires
    DocumentRoot /home/encheres/public_html # Emplacement des fichiers
web/php
    # ...
    IncludeOptional /var/webuzo-
data/apache2/custom/domains/encheresholsteinquebec.com.conf # Configuration
persistant
</VirtualHost>
```

Puisque ce fichier est automatiquement généré, on ne peut pas faire de changement sans qu'ils soient à risque d'être écrasé. Heureusement, les c-panel offrent une structure pour ajouter configurations persistents. La configuration clé est `IncludeOptional`. L'emplacement indiqué sera importé dans la configuration et est sous la forme `/var/webuzo-data/apache2/custom/domains/{ServerName}.conf`.

Fichier persistant

Maintenant qu'on connaît l'emplacement des fichiers persistants, on peut créer le dossier parent s'il n'existe pas déjà.

Ceci peut se faire via client SFTP, ou bien en ligne de commande:

1. s'assurer que le dossier parent `/var/webuzo-data/apache2/custom/domains` existe.
2. Créer un fichier pour chaque domaine:

```
/var/webuzo-data/apache2/custom/domains/encheresholsteinquebec.com.conf
/var/webuzo-data/apache2/custom/domains/staging-
kz.encheresholsteinquebec.com.conf
```

Le contenu de ces fichiers est la règle pour diriger toute requête sur `/api/ws` vers notre serveur websocket. Le port correspond à celle dans le `index.js` du serveur.

```
<Location "/api/ws">  
    ProxyPass "ws://localhost:9002"  
</Location>
```

Mise-à-jour apache

Suite à une modification dans la configuration apache, il faut le valider et redémarrer le service apache.

Valider:

```
/usr/local/apps/apache2/bin/apachectl configText
```

Redémarrer:

```
/usr/local/apps/apache2/bin/apachectl graceful
```

Nodejs

Nodejs & npm

Afin de partir le serveur, nodejs est nécessaire d'être installé. Celle-ci est souvent disponible pour télécharger directement sur le c-panel et inclus la commande `npm`.

En SSH, on peut ensuite valider l'installation avec `node -v` et `npm -v`

Initialiser le serveur

Emplacement

L'emplacement du serveur websocket n'a pas d'importance. Cependant, il est préférable de le garder avec le code wordpress pour l'inclure dans les backups.

Port

le port doit absolument être unique pour chaque instance et peut être modifier dans le `index.js`.

Pour une configuration simple, on peut partager le serveur websocket entre staging et prod. Cependant, si on veut faire la maintenance sur le serveur websocket, il est fortement recommandé de partir une deuxième instance sur un autre port (ex. 9001) et modifier le Virtual Host du domaine de staging en conséquence.

Dépendances

Les dépendances du serveur se retrouve dans le dossier `node_modules`. Si celles-ci n'existe pas, il suffit d'exécuter `npm install` dans le même dossier que `index.js`.

Démarrer le serveur (temporaire)

Dans le dossier qui se retrouve le serveur:

```
node index.js
```

Celle-ci va partir le serveur et validé s'il y a des erreurs. Cependant, le moment qu'on quitte le processus, ou ferme la connexion SSH, le serveur va se fermer. Pour le maintenir, il nous faut une gestion de processus node: `pm2`

PM2

PM2 (<https://pm2.keymetrics.io/>) est un gestionnaire de processus complet pour les projets nodes.

Installation

Installation global avec npm

```
npm install pm2 -g
```

Validé l'installation

```
pm2 status
```

À noter que si ce dernier ne fonctionne pas. Il se peut que la commande `pm2` n'est pas disponible globalement. Pour ce faire il faut exécuter la commande suivant:

```
ln -s /usr/local/apps/nodejs18/bin/pm2 /usr/bin/pm2
```

Redémarrage automatique

La commande suivant configure le serveur pour exécuter les processus nodes suite à un redémarrage du système. Elle est nécessaire que la première fois.

```
pm2 startup
```

Partir un processus

Naviguer à l'emplacement du serveur, et exécuter la commande suivant:

```
pm2 start index.js
```

et pour le rendre persistant:

```
pm2 save
```

Gestion

Les commandes suivants sont les plus commun pour manipuler les processus:

```
pm2 status (voir les process en ligne, ainsi que leur index)
pm2 start index.js (à partir du path du websocket)
pm2 save (Sauvegarde un "snapshot" des processus actifs. Ils seront
réinitialiser si le serveur redémarre)
pm2 restart <index>
pm2 del <index>
pm2 log --lines=20 (affiche les logs de tous les process, ça l'affiche
dans block un à la suite de l'autre, aloes c'est possible qu'il faut scroll:
process-log, errors-log, info-log)
```