

## Site web

### Notes des mails des Enrique concernant le serveur

Le virtual host nginx utilisé pour le site web est dans  
**/etc/nginx/sites-enabled/https-oporctunite.cfg**

vérification que les fichiers vhosts sont valides syntaxiquement  
**sudo nginx -t**

Redémarrage Nginx :  
[Linux Restart Nginx WebServer Command - nixCraft](#)

**sudo systemctl reload nginx**

Ce que j'ai fait hier soir pour git clone

```
cd /var/www/html/  
sudo rm -r oporctunite  
sudo git  
clone https://nom_utilisateur_gitlab_envt:"mot_de_passe_utilisateur_gitlab_envt"@gi  
tlab.envt.fr/oporctunite/oporctunite.git  
sudo chown -R www-data:www-data /var/www/html/oporctunite/  
sudo chmod -R 755 /var/www/html/oporctunite/
```

NB : j'ai eu qqs effets bizarres avec le cache des navigateurs  
ne pas hésiter à l'effacer après un pull du site

Ce qui posait pb pour le shell:

```
# set expiration of assets to MAX for caching  
location ~* \.(ico|css|js|gif|jpe?g|png)(\?[0-9]+)?$ {  
    expires max;  
    log_not_found off;  
}
```

## API

Projet cloné à la racine du projet, dans oporctunite-api

## PM2 process manager

PM2 permet de lancer ou éteindre un serveur node sans avoir à garder la fenêtre du terminal ouverte constamment.

### Commandes que j'ai utilisé pour installer PM2 :

```
sudo npm i pm2 -g
sudo pm2 start server
```

La 1ere permet d'installer pm2 globalement sur le serv avec npm, la 2eme permet de lancer l'api (on remplace npm par pm2 et server correspond à server.js)

### Autres commandes utiles :

```
sudo pm2 show server
sudo pm2 status
sudo pm2 restart server
sudo pm2 stop server
sudo pm2 logs (Show log stream)
sudo pm2 flush (Clear logs)
```

### Pour créer un script permettant de lancer le serv au redémarrage du serveur de l'envt :

```
sudo pm2 startup ubuntu
```

En savoir plus : <https://pm2.keymetrics.io/docs/usage/pm2-doc-single-page/>

## NGINX proxy setup

```
sudo nano /etc/nginx/sites-available/https-oporctunite.cfg
```

L'api écoute sur le port 5000.

Hors ce port est bloqué pour tout utilisateur autre que localhost par le firewall.

Il a donc fallu utiliser un proxy-pass pour rediriger les requêtes du port navigateur (80) vers le 5000 :

```
location /oporctunite-api/ {
    proxy_pass http://localhost:5000/;

    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection 'upgrade';
    proxy_set_header Host $host;
    proxy_cache_bypass $http_upgrade;
}
```

Remarque importante : le SSL n'est pas encore configuré pour l'api, donc pas de https. Enfin, on peut très bien mettre https dans l'url de la requête mais ce sera converti en http.