**Service de positionnement en ligne**

Un service de positionnement PPP en ligne a été implémenté afin de remplacer la diffusion du logiciel PPP open-source et de valoriser autrement la technologie du PPP-Wizard développée par le CNES. Ce service effectue un calcul PPP avec résolution d'ambiguïté à partir d'un fichier de mesures GNSS fourni par l'utilisateur. Il est accessible sur le site ppp-wizard.net, à l’adresse <http://www.ppp-wizard.net/upload.html>.

Ce document détaille l’implémentation du service.

Le glossaire ci-dessous précise les sigles présents dans le document.

|  |  |
| --- | --- |
| **Sigle / abréviation** | **Définition** |
| CGI | Common Gateway Interface |
| EXIM | Serveur de messagerie |
| HTML | HyperText Markup Language |
| IP | Internet Protocol |
| JSON | JavaScript Object Notation |
| GNSS | Global Navigation Satellite System |
| NMEA | National Marine Electronics Association |
| OVH | Entreprise proposant un hébergement de serveurs |
| PPP | Precise Point Positioning |
| PPP-Wizard | Precise Point Positioning With Integer and Zero-difference  Ambiguity Resolution Demonstrator |
| RINEX | Receiver Independant EXchange Format |

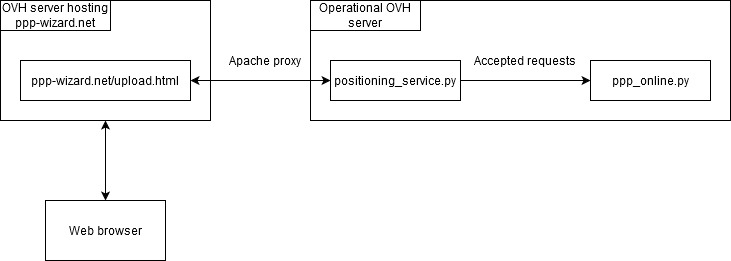
1. **Fonctionnement général**

Le service repose sur deux machines OVH utilisant un serveur Apache. La première machine (adresse IP : 94.23.202.142) héberge le site ppp-wizard.net et la seconde (adresse IP : 135.125.97.149) effectue les traitements pour gérer les requêtes des utilisateurs. La requête soumise est transférée entre les deux machines grâce au module proxy du serveur Apache.

La gestion des requêtes s’effectue par deux scripts python :

* positioning\_service.py : valide les requêtes et archive celles qui sont acceptées.
* ppp\_online.py : traite les rinex des requêtes acceptées avec les binaires du PPP-User.

Le schéma ci-dessous détaille le fonctionnement technique :



Un compte utilisateur nommé « apache » a été créé sur les deux machines. De plus, un dossier  « /home/apache/UPLOADS » a été créé sur la machine 135.125.97.149 pour stocker les requêtes.

1. **Mise en place de l’interface web sur la machine 94.23.202.142**

L’interface web est un formulaire HTML : l’utilisateur fournit un fichier de mesures au format RINEX (qui peut être compressé et/ou zippé) et une adresse email pour recevoir les résultats. Il choisit également des options de traitement (statique/cinématique et type de trames NMEA en sortie du cas cinématique).

Cette page HTML appelée upload.html est dans /home/apache/ sur la machine 94.23.202.142. L’action liée à ce formulaire est une requête POST exécutant un script python « positioning\_service.py » :

<form enctype = "multipart/form-data" name="test" action = "positioning\_service.py" method = "post">

Ce script permet de gérer les requêtes des utilisateurs et se trouve sur la machine 135.125.97.149. La requête POST est redirigée vers cette machine grâce au module proxy du serveur Apache.

Une fois la requête soumise sur le formulaire HTML, une page de résultats, correspondant aux traitements du script positioning\_service.py, est affichée à l’utilisateur.

1. **Configuration du serveur Apache**

Un serveur Apache2 a été installé sur les deux machines OVH avec la commande :

apt-get install apache2

Le fichier de configuration du site (/etc/apache2/sites-available/000-default) a été modifié spécifiquement au besoin de chaque machine, comme présenté ci-dessous.

* 1. Configuration sur la machine 94.23.202.142

Le fichier de configuration a été modifié pour définir le chemin /home/apache comme racine des fichiers servis par le serveur Apache avec la ligne suivante :

DocumentRoot /home/apache

De plus, pour cette machine, le module proxy d’Apache a été activé avec les commandes suivantes (à effectuer en étant root) :  
a2enmod proxy  
a2enmod proxy\_http

Le fichier de configuration a été adapté avec l’ajout des lignes suivantes :   
ProxyPass /positioning\_service.py http:// 135.125.97.149/positioning\_service.py

ProxyPass /UPLOADS/ http:// 135.125.97.149/UPLOADS/

De cette façon, lorsque la requête POST présentée à la partie précédente est effectuée, Apache redirige la requête vers la machine 135.125.97.149.

* 1. Configuration sur la machine 135.125.97.149

Des alias ont été ajoutés dans le fichier de configuration pour mettre à jour les chemins des dossiers et fichiers liés au service de positionnement :

Alias /UPLOADS/ /home/apache/UPLOADS/

ScriptAlias /positioning\_service.py /home/apache/positioning\_service.py

De plus, pour cette machine, la technologie CGI, qui est une interface entre Apache et Python, a été activée avec la commande suivante (à effectuer en étant root) :   
a2enmod mpm\_prefork cgi

Le fichier de configuration a été adapté avec les lignes suivantes :

<Directory /home/apache>  
 Options +ExecCGI  
 Require all granted  
 </Directory>  
  
AddHandler cgi-script .py

Enfin, le fichier de configuration a été modifié pour limiter la taille des fichiers RINEX à 52 Mo, avec l’ajout de la ligne suivante :  
LimitRequestBody 52428800

Le tutoriel suivant récapitule les instructions à effectuer :

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-apache-mysql-and-python-lamp-server-without-frameworks-on-ubuntu-14-04>

* 1. Redémarrage du serveur

Après les commandes effectuées en étant root et les modifications du fichier de configuration, il faut redémarrer le serveur Apache avec la commande suivante :

systemctl restart apache2

1. **Description des scripts Python sur la machine 135.125.97.149**

Les scripts Python utilisés se trouvent dans le dossier /home/apache. Les deux principaux sont positioning\_service.py et ppp\_online.py. Deux autres scripts, comportant des fonctions génériques, sont également utilisés (julian.py et utility.py).

* 1. positioning\_service.py

Ce script s’exécute sans argument. Il récupère les informations de la requête POST avec la technologie CGI puis valide la requête. Pour cela, il détermine le nombre de requêtes actuellement traitées par le service en comptant le nombre de processus en cours.

Si celui-ci est supérieur ou égal à 50 alors la requête est refusée et un message demandant à l’utilisateur de réitérer plus tard sa demande est affiché sur la page de résultats.

Sinon, la requête est acceptée et un message de confirmation est affiché sur la page de résultats. Si le nombre de processus dépasse 3, le message indique également que la requête est mise en file d’attente et précise le nombre de requêtes actuellement en attente.

Les requêtes acceptées sont archivées dans le dossier UPLOADS avec la création d’un .tgz de nom unique : <dateCourante>\_<heureCourante>\_<nomDuFichierSoumis>.tgz

Ce nom unique permet de gérer des requêtes qui comportent le même nom de RINEX. L’archive contient le fichier RINEX et un fichier JSON avec les options de la requête (adresse email, type de traitement, type de trame NMEA).

Enfin, le script appelle en tâche de fond le script ppp\_online.py avec la commande subprocess.Popen

* 1. ppp\_online.py
     1. Démarrage du script

La commande pour exécuter ce script est : python3 ppp\_online.py <nomArchive> [mail]

L’argument optionnel « mail » indique si un mail de résultats sera envoyé à la fin de l’exécution : il est donc présent lors de l’appel par positioning\_service.py. Mais on peut également vouloir faire du rejeu sans envoi de mail.

Pour exécuter ce script, il a fallu installer le module python numpy.

Le script met en place une file d’attente avec le mécanisme de synchronisation Linux : 3 fichiers lock sont utilisés, ce qui permet de limiter le nombre de processus exécutés en même temps à 3 et ainsi de ne pas submerger la machine. Un processus en attente vérifie chaque seconde si un fichier lock est disponible pour pouvoir démarrer.

Une fois démarré, le processus extrait l’archive et créé un répertoire de travail avec un nom unique : <nomArchiveExtraite>\_<dateCourante>\_<heureCourante>

Cela permet d’avoir des répertoires de travail distincts et notamment de pouvoir faire du rejeu.

Exemple :

* Un utilisateur soumet un fichier RINEX nommé « GNSS\_1Hz.rnx » le 01/03/2021 à 9h30. L’archive est alors créée par le script positioning\_service.py : 01032021\_093000\_GNSS\_1Hz.rnx.tgz
* Le répertoire de travail est ensuite créé par le script ppp\_online.py, deux secondes plus tard :

01032021\_093000\_GNSS\_1Hz.rnx\_01032021\_093002

* Si on souhaite faire du rejeu de cette archive le 02/03/2021 à 10h en utilisant le script ppp\_online.py, alors un nouveau répertoire de travail sera créé par le script : 01032021\_093000\_GNSS\_1Hz.rnx\_02032021\_100000

Les répertoires de travail datant d’au moins 30 jours sont supprimés avec la commande find, par le script positoning\_service.py. En revanche, toutes les archives sont gardées.

* + 1. Binaires utilisés

Le script vérifie ensuite le format du fichier RINEX. Si besoin, le fichier est décompressé avec le binaire CRX2RNX qui effectue la décompression HATANAKA. Ce binaire est placé dans le dossier /home/apache/bin.

Enfin, le script traite le fichier RINEX en appelant les binaires du PPP-User. Ceux-ci ont été également placés dans le dossier /home/apache/bin. Les binaires sont les suivants :

* convertODTS : Convertit le fichier RINEX dans un format synthétique listant les mesures de code, de phase et de Doppler.
* generateLowLevel : Génère un fichier de mesures à partir des observables du fichier RINEX, des éphémérides et des biais.
* processLowLevel : Effectue le calcul PPP à partir du fichier de mesures généré.

Le fichier de configuration pour processLowLevel se nomme conf\_process\_PPP\_AR\_GPSGLOGAL\_generic.txt et se trouve dans le dossier /home/apache.

* + 1. Produits utilisés dans le calcul PPP

Les produits nécessaires en entrée de generateLowLevel (éphémérides et biais) ont été archivés sur le compte archive de la machine 51.75.129.104 depuis de début de l’année 2020. Il y a les produits post-traités (post\_<jourJulien>.tgz) ainsi que les produits temps réel (tr\_<jourJulien>.tgz).

Pour pouvoir accéder à ces produis, une clé SSH a été installée sur la machine 135.125.97.149 avec les deux commandes suivantes :

ssh-keygen

ssh-copy-id archive@51.75.129.104

* + 1. Envoi des résultats par email

La commande mailx est utilisée pour envoyer les résultats par mail, au format HTML, depuis l’adresse « no-reply@ppp-wizard.net ».

Pour cela, il a fallu configurer le serveur de messagerie Exim4 en suivant les instructions de la documentation d’Exim4 : <https://wiki.debian.org/Exim#Example_stand-alone_example>

De plus, il a fallu configurer les protocoles d’authentification afin que les mails ne soient pas considérés comme indésirables en suivant les instructions de la documentation OVH : <https://docs.ovh.com/fr/dedicated/optimiser-envoi-emails/>

Ainsi, le SPF, DKIM et reverse IP ont été configurés à partir de la page manager OVH.