

UNANTES 20_212_Dispositif orientation active - Procédure d'installation générale

- Infrastructure d'hébergement
 - Spécificité
 - Schéma général des composants
- Détail des composants
 - Serveur frontal
 - Serveur d'application
 - Dimensionnement
 - Prérequis logiciels
 - Serveur de bases de données
 - Serveur de fichiers
- Procédure d'installation de l'application
 - PostgreSQL
 - Installation du paquet Debian
 - Création de la base de données
 - Service Aply
 - Installation de Java
 - Déclaration du service
 - Activation du service
 - Optionnel : Installation et configuration de GeolP
 - Mise en place de la base de données
 - Mise à jour automatique de la base de données
 - Apache
 - Installation
 - Configuration
 - HAProxy
 - Installation
 - Configuration
 - Redémarrage
 - Configuration de l'application
 - Profil Spring
 - Base de données
 - Mails
 - Checklist de validation
 - Actions après le premier lancement de l'application
 - Modifier le mot de passe des utilisateurs

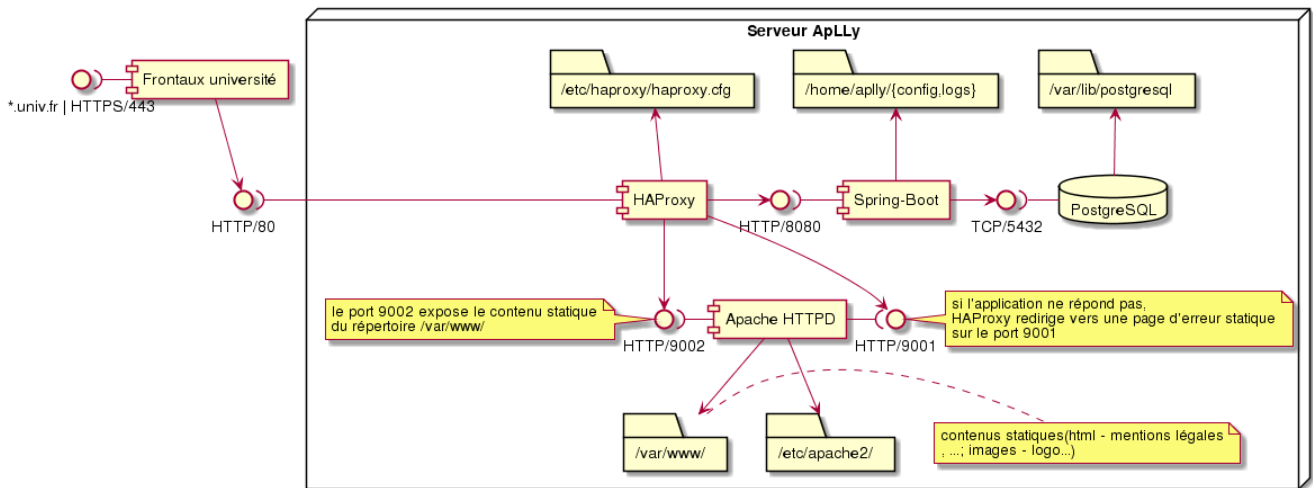
Infrastructure d'hébergement

Spécificité

Certains frontaux HTTP d'université ne savent traiter que des services sur les ports standards HTTP (80 et 443). Comme il est difficile de faire tourner un service Java Spring Boot sur le port 80 (contrainte de sécurité des systèmes Linux), un reverse proxy (HAProxy) est positionné pour rediriger 80 <-> 8080.

Schéma général des composants

Diagramme de composants



Détail des composants

Pour chacun des composants nous détaillons ici le dimensionnement et les prérequis logiciels.

Serveur frontal

Pas de VM dédiée dans l'infra actuelle

Serveur d'application

A ce stade, cette VM héberge le frontal HAProxy, l'application, ainsi que la base de données et les fichiers.

Elle s'appuie sur un serveur HAProxy, un serveur Web Apache, et un serveur de base de données PostgreSQL.

Dimensionnement

- Processeur : 4 CPU
- Mémoire : 8 Go de RAM
- Espace disque : Partition unique de 50 Go

Prérequis logiciels

- Système d'exploitation : Debian 10 64 bits
- HAProxy 1.8
- Apache 2.4
- OpenJDK 11 (64-Bit Server VM)
- PostgreSQL 13.1

Serveur de bases de données

Pas de VM dédiée dans l'infra actuelle

Serveur de fichiers

Pas de VM dédiée dans l'infra actuelle

Procédure d'installation de l'application

PostgreSQL

Installation du paquet Debian

<https://www.postgresql.org/download/linux/debian/>

```
# Create the file repository configuration:
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" >
/etc/apt/sources.list.d/pgdg.list'

# Import the repository signing key:
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo apt-key add -

# Update the package lists:
sudo apt-get update

# Install the latest version of PostgreSQL.
# If you want a specific version, use 'postgresql-12' or similar instead of 'postgresql':
sudo apt-get -y install postgresql
```

Création de la base de données

<https://www.postgresql.org/docs/12/tutorial-createdb.html> (Remplacer %PASSWORD% par le mot de passe souhaité pour la base de données).

```
sudo su - postgres
createdb aply
psql aply
create user aply with password '%PASSWORD%';
GRANT ALL PRIVILEGES ON DATABASE aply TO aply ;
# quitter la console postgres
\q
# se déconnecter de la session utilisateur postgres
exit
# se connecter sur une session root
sudo su
# créer un utilisateur aply avec le même mot de passe que celui défini lors de la création de l'utilisateur dans
la console postgres
adduser aply
#
```

Service Aply

Installation de Java

```
sudo apt-get install default-jdk
```

Déclaration du service

/etc/systemd/system/aply.service

```
[Unit]
Description=Aply
After=syslog.target

[Service]
User=aply
WorkingDirectory=/home/aply
ExecStart=java -jar orientactive.jar
SuccessExitStatus=143

[Install]
WantedBy=multi-user.target
```

Activation du service

```
sudo systemctl enable aply.service
```

Optionnel : Installation et configuration de GeoIP

Cette fonctionnalité permet de récolter des informations complémentaires sur les utilisateurs qui répondent aux formulaires. La réglementation RGPD n'est pas respectée dans le cas où cette fonction serait utilisée en parallèle avec l'authentification de l'utilisateur sur un formulaire.

Cette fonctionnalité nécessite un compte sur <https://www.maxmind.com/en/home> afin d'avoir accès aux différentes bases de données GeoIP.

Mise en place de la base de données

Pour activer la fonctionnalité, il faut installer la base de données GeoIP sur l'environnement, et la référencer dans les propriétés de l'application.

- Télécharger la base de données "GeoLite2 City" depuis le site <https://www.maxmind.com/>;
- Extraire l'archive afin de récupérer le fichier "GeoLite2-City.mmdb" qui correspond à la base de données;
- Placer l'archive sur le serveur de l'application, par exemple dans "/var/lib/geoip/GeoLite2-City.mmdb";
- Référencer la base de données dans les propriétés de l'application :

```
application.geoip-database=file:/var/lib/geoip/GeoLite2-City.mmdb
```

Mise à jour automatique de la base de données

Afin de permettre la mise à jour automatique de la base de données, il faut mettre en place le script de mise à jour fourni par Maxmind.

Cette documentation n'est peut-être plus à jour, de préférence, se référer au README.md de <https://github.com/maxmind/geoipupdate>.

Télécharger la dernière version pour votre système d'exploitation sur <https://github.com/maxmind/geoipupdate/releases> (exemple : `geoipupdate_4.6.0_linux_arm64.tar.gz` pour linux).

Décompresser l'archive dans un répertoire de votre choix (il devra être accessible en lecture par l'utilisateur qui exécute le service aply)

Modifier le fichier de configuration afin de renseigner la licence, le code utilisateur, l'emplacement de la base de données et la base de données à utiliser

- Pour trouver le code du compte (accountId), il faut se connecter. Le code est situé dans l'URL : <https://www.maxmind.com/en/accounts/<AccountID>>;
- La clé de licence est présente à cette adresse : <https://www.maxmind.com/en/accounts/<AccountID>/license-key>
- Pour renseigner l'emplacement de la base de données, il faut "décommenter" la ligne suivante

```
DatabaseDirectory /chemin/vers/la/base/de/donnée
```

- Il faut utiliser la base de donnée "lite city" :

```
EditionIds GeoLite2-City
```

La commande pour lancer l'utilitaire est la suivante : `./geoipupdate -f path/to/GeoIP.conf`

Apache

Remplacer "aply-prod.univ.fr" par votre nom de domaine.

Installation

```
sudo apt install apache2
sudo a2enmod rewrite
```

Configuration

/etc/apache2/ports.conf

```
Listen 9001
Listen 9002
```

/etc/apache2/sites-available/aplly-erreur.conf

```
<VirtualHost *:9001>

    ServerName aplly-prod.univ.fr
    LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
    ErrorLog /var/log/apache2/aplly-error.log
    CustomLog /var/log/apache2/aplly-access.log combined

    DocumentRoot /var/www/html/
    <Directory /var/www/html/>
        Options -Indexes
    </Directory>

    RewriteEngine On
    RewriteCond %{DOCUMENT_ROOT}/unavailable.html -f
    RewriteCond %{SCRIPT_FILENAME} !unavailable.html
    RewriteCond %{SCRIPT_FILENAME} !^.*\.png
    RewriteCond %{SCRIPT_FILENAME} !^.*\.ico
    RewriteRule ^.*$ /unavailable.html [R=503,L]
    ErrorDocument 503 /unavailable.html

</VirtualHost>
```

/etc/apache2/sites-available/aplly-images.conf

```
<VirtualHost *:9002>

    ServerName aplly-prod.univ.fr
    LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
    ErrorLog /var/log/apache2/aplly-images.log
    CustomLog /var/log/apache2/aplly-access-images.log combined
    DocumentRoot /var/www/
    Alias /images "/var/www/images/"
    <Directory "/var/www/images">
        Options +Indexes
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>
    Alias /static "/var/www/html/"
    <Directory "/var/www/html">
        Options +Indexes
        AllowOverride None
        Order allow,deny
        Allow from all
    </Directory>

</VirtualHost>
```

```
sudo a2dissite default-ssl.conf
sudo a2dissite 000-default.conf
sudo a2ensite aplly-erreur.conf
sudo a2ensite aplly-images.conf
rm /etc/apache2/sites-available/default-ssl.conf
rm /etc/apache2/sites-available/000-default.conf
sudo mkdir -p /var/www/images
sudo service apache2 restart
```

HAProxy

Installation

```
sudo apt install haproxy hatop
```

Configuration

/etc/haproxy/haproxy.cfg

```
global
    log /dev/log      local0
    log /dev/log      local1 notice
    chroot /var/lib/haproxy
    stats socket /run/haproxy/admin.sock mode 660 level admin expose-fd listeners
    stats timeout 30s
    user haproxy
    group haproxy
    daemon

    # Default SSL material locations
    ca-base /etc/ssl/certs
    crt-base /etc/ssl/private

    # Default ciphers to use on SSL-enabled listening sockets.
    # For more information, see ciphers(1SSL). This list is from:
    # https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
    # An alternative list with additional directives can be obtained from
    # https://mozilla.github.io/server-side-tls/ssl-config-generator/?server=haproxy
    ssl-default-bind-ciphers
ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:DH+AES:RSA+AESGCM:RSA+AES:!aNULL:!MD5:!DSS
    ssl-default-bind-options no-ssl-v3

defaults
    log          global
    mode         http
    option       httplog
    option       dontlognull
    timeout      connect 5000
    timeout      client  50000
    timeout      server  50000
    errorfile    400 /etc/haproxy/errors/400.http
    errorfile    403 /etc/haproxy/errors/403.http
    errorfile    408 /etc/haproxy/errors/408.http
    errorfile    500 /etc/haproxy/errors/500.http
    errorfile    502 /etc/haproxy/errors/502.http
    errorfile    503 /etc/haproxy/errors/503.http
    errorfile    504 /etc/haproxy/errors/504.http

frontend req_http
    bind        *:80
    acl path_images path_beg /images
    acl path_static path_beg /static
    use_backend static if path_images
    use_backend static if path_static
    default_backend ap1ly
    capture cookie JSESSIONID= len 51
    capture request  header Referer len 200
    capture request  header User-Agent len 200
    capture request  header X-Forwarded-For len 15

backend ap1ly
    option httpchk GET /management/health HTTP/1.1\r\nHost:\ www
    server ap1ly 127.0.0.1:8080 check port 8080 maxconn 500
    server error 127.0.0.1:9001 backup

backend static
    server static 127.0.0.1:9002
```

Redémarrage

```
sudo service haproxy restart
```

Configuration de l'application

Profil Spring

- Ajouter un fichier /home/ap1ly/config/application.properties contenant :

```
spring.profiles.active=prod
```

Ce fichier permet seulement de définir le profil spring utilisé par l'application.

Base de données

- Ajouter un fichier `/home/aplly/config/application-prod.properties` contenant (en ajoutant votre mot de passe pour la base de données) :

```
spring.datasource.url=jdbc:postgresql://localhost:5432/aplly
spring.datasource.username=aplly
spring.datasource.password=
```

Ce fichier contiendra toutes les propriétés de configuration de l'application.

Mails

- La déclaration du serveur smtp est faite via les propriétés Spring dans le fichier `/home/aplly/config/application-prod.properties` (Remplacer pour les valeurs souhaitées) :

```
##
# SMTP
##
spring.mail.host=smtp.univ.prive
spring.mail.port=25
jhipster.mail.from=aplly@univ.fr
jhipster.mail.base-url=https://aplly.univ.fr
```

Checklist de validation

- Connexion à la base de données (cf. procédure d'exploitation)
- Démarrage du service aplly
 - `/management/health`
 - le connecter au back-office puis : `https://aplly-prod.univ.fr/admin/jhi-health`
- Démarrage de HAProxy :
 - `sudo service haproxy status`
- Démarrage de Apache
 - `sudo service apache status`

Actions après le premier lancement de l'application

Modifier le mot de passe des utilisateurs

Après avoir lancé pour la première fois l'application, la base de données a été initialisée. Il faut changer le mot de passe des utilisateurs par un mot de passe que vous avez définie via la commande suivante :

```
update jhi_user set password_hash = '$2y$10$G2eXSo/p6DDYVfYNIoNLO2i6Sljul8dEG6kkNlnJPl3aPYG87IAi' where true;
```

Vous pouvez remplacer le hash avec le mot de passe que vous souhaitez, encodé en bcrypt (vous pouvez utiliser [ce site](#) pour encoder). Le hash dans la commande, au-dessus, correspond à "adminuniv". À noter que le mot de passe doit faire plus de 8 caractères.

Vous devez désormais pouvoir vous connecter avec l'identifiant "admin" sur "/login" pour accéder au backoffice de l'application.