ezAgimus (traitement conjoint avec ezPaarse)

ezAgimus est l'intégration dans un tableau de bord **Agimus** des données produites par une instance locale d'**ezPAARSE** (http://ezpaarse.couperin.org) qui filtre et enrichit sur les traces de passage vers la documentation électronique payante. Les fichiers de traces analysés proviennent d'un reverse-proxy comme **EZproxy** (https://www.oclc.org/ezproxy.en.html), **Biblio-PAM** (http://www.alixen.fr/partage_documentaire_securise.html) ou autre (squid, apache, etc.).

Vous pouvez utiliser 2 traitements différents suivant la récupération ou non du login dans les logs ezproxy. Nous vous conseillons la première méthode qui permet de conserver les possibilités de retraitement ultérieurs fournis par ezPaarse.

Récupération du login

Le traitement présenté ici peut être adapté avec le cookie en ajoutant simplement un lien symbolique vers le fichier permettant la correspondance trace <-> uid et en modifiant l'analyse grok de ce traitement.

Cette version est conseillée car elle permet un traitement indépendant des traitements ezPaarse. En effet, lorsque de nouveaux parseurs sont créés dans ezPaarse, il est nécessaire de retraiter les anciens logs.

Le SI de l'établissement évoluant en permanence, il faut faire les enrichissements agimus-ng sur les logs bruts ezproxy afin que le fichier résultat soit enregistré pour pouvoir servir de base aux retraitements ultérieurs.

Ce traitement diverge donc des autres dans le sens où l'enrichissement et l'enregistrement sont décorrélés.

L'enrichissement se fait grâce au traitemement ezproxy. La génération du fichier quotidien et son envoi vers le serveur ezpaarse se fait avec le script traitement-ezproxy.sh. Ces fichiers sont enrichis puis anonymisés et doivent être stocké sur le serveur ezPaarse.

L'envoi des résultats elasticsearch est quant à lui réalisé avec la configuration ezpaarse. Lorsqu'il y a un retraitement ezPaarse quelle que soit la date des fichiers traités, ceux-ci sont renvoyés vers le serveur agimus. Pour détecter et retraiter ces nouveaux fichiers, il faut utiliser le script traitement-ezpaarse.sh. Ce script est indépendant du script quotidien et peut être lancé aussi souvent que souhaité.

Récupération du cookie de trace uniquement

> Cette version ne permet pas un retraitement ezPaarse. Si vous ne disposez pas du login, nous vous conseillons d'adapter la configuration ci-dessus pour transformer le cookie en uid et d'utiliser le traitement en 2 temps proposé ci-dessus.

Voici l'ensemble des outils/paramétrages pour la mise en place d'ezAgimus (basé sur l'intégration de l'université de Lille, Sciences et technologies).

EZproxy

Dans la configuration d'EZproxy, il faut mettre en place de l'authentification Shibboleth et la configuration des logs :

- dans le fichier de config, pour les logs, voici les instructions :

configuration pour les logs

```
Option LogUser
LogFormat %h "%{ezproxy-groups}i" %u %t "%r" %s %b "%{Referer}i" "%{user-agent}i" "%{Cookie}i" %{ezproxy-session}i
LogFile -strftime /var/log/ezproxy/ezp%Y%m%d.log
```

Un fichier quotidien est créé avec l'ensemble des informations nécessaires.

> Ce fichier est à envoyer toutes les nuits sur le serveur Agimus selon le même principe que les autres fichiers de logs : [...]/YYYY/MM/DD/ezp.log

Dans le traitement journalier (daily_batch) d'Agimus, il faut ajouter deux instructions :

- le traitement du fichier de log par ezPAARSE
- le traitement du résultat par Agimus.

Pour le traitement ezPAARSE, il faut faire en curl avec cette instruction :

/opt/agimus-ng/build/scripts/daily_batch.sh

```
[...]

echo "$LINE_SEPARATOR"
echo "#### ezPAARSE - EZproxy logs : "`date +'%F %R'`
if [ -f "$LOG_DIR/$DATE/ezp.log" ]; then
    curl -X POST --proxy "" --no-buffer -H 'Log-Format-ezproxy: %h "%{ezproxy-groups}<.*>" %u %t "%r" %s %b "%
{Referer}<[^ ]+>" "%{user-agent}<.*>"
"%{Cookie}<.*>" %{ezproxy-session}<[a-zA-Z0 -9\)-]+>' \
        -H 'Accept: text/csv' \
        -H 'Traces-Level: error' \
        -H 'Output-Fields: +year,+institution,+datetime' \
        -H 'Relative-Domain: docproxy.univ.fr' \
        -H 'Double-Click-Removal: true' \
        --data-binary @$LOG_DIR/$DATE/ezp.log  http://127.0.0.1:59599 -v \
        -o $LOG_DIR/$DATE/ezp.csv
else
        echo "ERR : NO file logs EZP" >&2
fi
echo ""

[...]
```

A la suite de ce traitement, un fichier CSV (ezp.csv) est créé.

Voici, toujours dans le fichier daily_batch, le traitement de ce fichier pour injection dans Agimus :

/opt/agimus-ng/build/scripts/daily_batch.sh

```
[...]

echo "$LINE_SEPARATOR"
echo "#### Import ezPAARSE CSV logs : "`date +'%F %R'`
if [ -f "$LOG_DIR/$DATE/ezp.csv" ]; then
    echo "#### Number of lines in file "`cat $LOG_DIR/$DATE/ezp.csv | wc -l`
    cat $LOG_DIR/$DATE/ezp.csv | $LOGSTASH_DIR/bin/logstash --quiet -f $BUILD_HOME/logstash/logstash-ezPAARSE.
conf >&2
else
        echo "ERR : NO file logs EZP csv" >&2
fi
echo ""

[...]
```

Vous trouverez le fichier de conf logstash pour ezPAARSE sur le github : https://raw.githubusercontent.com/EsupPortail/agimus-ng/v2.0/logstash/logstash-ezpaarse.conf
ainsi que les deux dashboards générés https://github.com/EsupPortail/agimus-ng/tree/v2.0/kibana/visualization.

Vous trouverez des exemples de rendu des dashboards sur la page : 5 - Les tableaux de bord disponibles