

# Installation, configuration et exploitation de Grouper

## Introduction

Après plusieurs années d'utilisation dans différents établissements, [Grouper d'Internet2](#) est considéré comme une solution de gestion de groupes mature et satisfaisante ; notamment dans sa version actuelle, la **2.3.0**.

D'architecture complexe mais complète, sa mise en place dans un établissement n'est pas des plus aisées, mais une fois en fonctionnement, Grouper d'Internet2 est une solution qui sait se faire apprécier.

Son installation et sa configuration sont en effet délicates, à sa décharge Grouper est hautement configurable et peut s'adapter à des cas d'usages avancés et variés.

Dès les prémices de l'installation plusieurs choix s'offrent à l'intégrateur : installation depuis Grouper Installer, depuis une archive, depuis les sources, ...

Les possibilités de configuration et de mises en place des synchronisations sont ensuite sujettes également à bon nombre de variantes.

A côté de cela, l'uniformisation croissante de nos Systèmes d'Information (supann, openldap, shibboleth, fédération d'identités, CAS) ainsi que les expériences et retours des établissements ayant sauté le pas (cf [Contributions d'établissements](#) et présentations [ESUP-Days n°20](#), [ESUP-Days n°22](#), ...) nous confortent dans l'idée que nos installations de Grouper peuvent être finalement similaires.

La gestion d'un package Grouper 'ESUP' nous paraît cependant trop ambitieuse et à terme contre-productive.

Aussi, et à l'image de ce qui avait été fait pour la version 2.1.5 de Grouper au travers du [Workshop Grouper 2.1.5 – 15/04/2014 – Paris Descartes](#), avons-nous décidé d'élaborer une documentation actualisée de mise en oeuvre de Grouper 2.3.0, documentation se voulant simple, **concise**, efficace et répondant à un contexte d'usage très spécifique et restreint : usage simple constaté dans nos établissements jusque-là.

Pour faciliter l'installation de Grouper, ce document propose ainsi de partir d'un "fork" github de Grouper (correspondant initialement aux travaux de configuration de Grouper de l'Université de Paris1) à reparamétrer. Ce document vous indique ce qu'il faut reparamétrer et comment installer les services correspondants. Cette démarche a par ailleurs fait l'objet d'une présentation en 2018 lors d'Esup-Days n°25, présentation que nous vous invitons à consulter ; diaporama et vidéo disponible ici : [Installation rapide de Grouper à la ESUP](#)

Une fois que vous aurez une installation "simple" et fonctionnelle intégrée dans votre système d'information, libre à vous par la suite d'aller (éventuellement) plus loin dans l'usage de Grouper.

Grouper peut cependant tout à fait convenir à un usage circonscrit comme proposé ici ; cette intégration ayant alors le mérite de rester relativement simple, de ne pas sortir des clous par rapport à ce qui se fait "basiquement" avec Grouper ; on retire alors les avantages d'une maintenance facilitée par exemple.

Toujours à l'image de ce qui avait été proposé lors du [Workshop Grouper 2.1.5 – 15/04/2014 – Paris Descartes](#), nous vous proposons une VM (téléchargement disponible depuis la page [VM Grouper, Shibboleth, LDAP](#)) sous virtualbox complètement autonome !

Cela vous permet ainsi d'avoir à disposition une installation complète de grouper qui fonctionne. Pour la partie Grouper, cette VM a été réalisée en suivant cette documentation. Elle embarque un IdP Shibboleth (v3) et un (open)LDAP Supann.

Une page spécifique donne quelques indications sur celle-ci : [VM Grouper, Shibboleth, LDAP](#)

## Plan du document

- [Introduction](#)
- [Plan du document](#)
- [Version de Grouper](#)
- [Contexte d'usage et intégration souhaitée de Grouper dans le SI de l'établissement.](#)
- [LDAP, prérequis/ajustements techniques](#)
  - [overlay memberOf](#)
  - [Indexes](#)
  - [Tri des attributs member](#)
  - [Ajustement de la RAM pour openldap avec LMDB](#)
- [Dimensionnement serveurs](#)
- [Éléments techniques à installer sur le serveur qui héberge Grouper.](#)
  - [Java / Tomcat](#)
    - [Ant](#)
    - [Java](#)
    - [Tomcat](#)
  - [PostgreSQL](#)
  - [Apache et mod\\_shib](#)
- [Grouper ESUP](#)
  - [Récupération de la version Grouper d'Esup](#)
  - [Modifications apportées](#)
  - [Jars supplémentaires](#)
  - [Déploiement](#)
  - [Scripts de démarrage](#)
  - [Backup de la BD](#)
  - [Tâches CRON de synchronisation](#)
- [Démarrage de Grouper et prise en main](#)
  - [Initialisation de la base de données de Grouper](#)
  - [Peuplement du groupe etc:sysadmingroup](#)
  - [Démarrage des services](#)
  - [Interface Web](#)

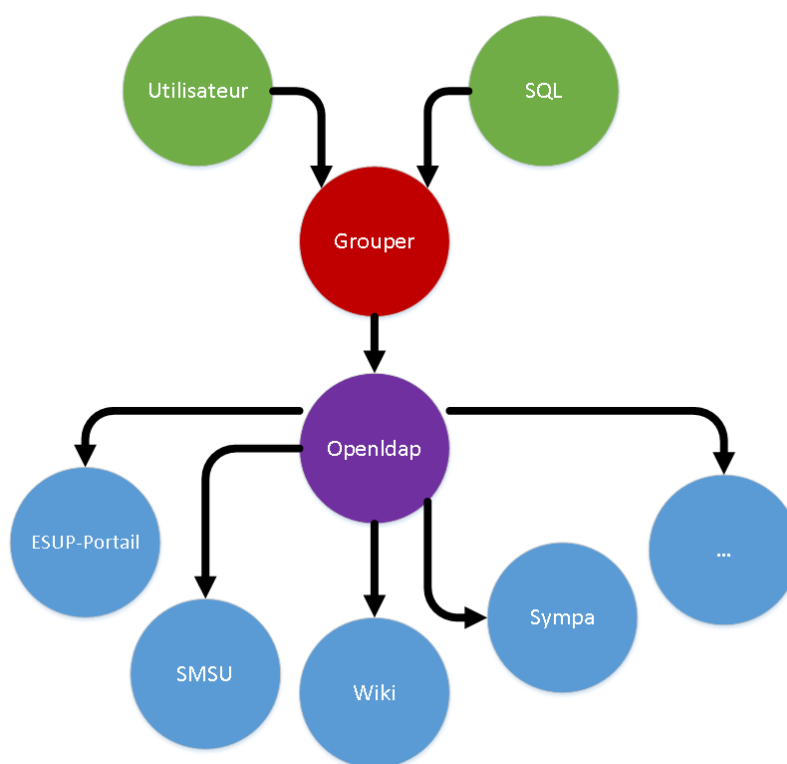
## Version de Grouper

La version de Grouper actuellement traitée par ce document est la **2.3.0**.

## Contexte d'usage et intégration souhaitée de Grouper dans le SI de l'établissement.

Le contexte d'usage et l'intégration de Grouper dans le système d'information de l'établissement sont contraints de cette façon :

- authentification shibboleth sur les interfaces web Grouper - même si ici seuls les personnels de l'établissement sont amenés à s'authentifier sur celles-ci dans cette configuration.
- le LDAP de l'établissement est un openldap basé sur les recommandations (et schémas) Supann
- les potentiels membres des groupes (source des individus pour grouper) sont tous issus du LDAP de l'établissement et on utilise comme identifiant non pas leur uid ldap mais leur eppn (eduPersonPrincipalName)
- les groupes sont poussés/synchronisés dans le Ldap comme groupOfNames ('plus supann' et 'plus standard' que les posixGroup) côté ou=groups et éventuellement (conseillé) memberOf côté ou=people (via un overlay) par Grouper
- la nomenclature des groupes dans ldap reprend la nomenclature hiérarchique des groupes de Grouper - les ":" sont transformés en "." dans le passage de l'id de Grouper au dn ldap
- les applications utilisant les groupes ne conversent pas avec Grouper mais avec LDAP (cf schéma ci-dessous)



## LDAP, prérequis/ajustements techniques

L'intégration de Grouper dans le Système d'Information est portée par l'annuaire LDAP openldap construit selon les recommandations supann.

Des ajustements techniques doivent/peuvent être effectués (si non présents dans votre ldap).

### overlay memberOf

Pour faciliter l'usage des groupes ldap dans les applications, il est intéressant de proposer l'information d'appartenance d'un individu à un groupe dans l'objet de l'individu (supannPerson positionné dans ou=people) en plus de cette information portée par le groupOfNames (supannGroupe) lui-même.

Pour ce faire, l'idée est de mettre en place le overlay memberOf.

Vous pouvez suivre cette documentation par exemple : <http://idmoim.blogspot.fr/2014/05/enabling-memberof-attribute-in-openldap.html>

## Indexes

Pour optimiser les performances des applicatifs et de Grouper on vous conseille d'ajouter des indexes, notamment sur

- eduPersonPrincipalName et memberOf côté ou=people (Attention constaté une dégradation des perfs en indexant l'attribut interne memberOf (D. Lalot))
- member côté ou=groups

<https://www.vincentiefoghe.net/content/ajout-dindex-sur-un-annuaire-openldap>

## Tri des attributs member

Sur des groupes de plusieurs milliers ou dizaines de milliers d'individus, il est impératif que l'attribut member soit trié, notamment pour que la suppression d'un individu à un groupe ne pose pas de problème.

Il faut donc demander à openldap de trier (olcSortVals) cet attribut member ... sous peine d'obtenir des temps de réponse catastrophiques, de récupérer des DB\_LOCK\_DEADLOCK au niveau de la base ldap, voir de mettre à terre votre serveur ldap !

Pour ce faire, vous pouvez suivre cette documentation :

<https://www.vincentiefoghe.net/content/optimisation-openldap-pour-la-suppression-dun-membre-dun-groupe>



### moteur backend

Cet ajustement de tri est effectivement utile lorsqu'on utilise un backend comme BDB (Berkeley DataBase)

Avec d'autres moteurs plus récents/évolués comme LMDB, il semblerait que ce ne soit plus nécessaire.

Ce backend plus récent se montre plus performant ; openldap recommande maintenant son usage : <https://www.vincentiefoghe.net/content/openldap-changer-moteur-backend>

## Ajustement de la RAM pour openldap avec LMDB

*Rappel : cf plus haut, (L)MDB est recommandé et est maintenant le backend donné par défaut dans la plupart des distributions.*

- LMDB monte la base en RAM pour pouvoir l'exploiter ;
- grouper fait des requêtes assez conséquentes (1 pour trouver un groupe puis 1 par membre d'un groupe) qui, si vous avez des groupes comprenant finalement l'ensemble des individus, correspondent finalement à requêter très régulièrement toute la base ;
- les requêtes, notamment sur les membres, sont mises en cache par grouper (via ehcache) en RAM ...
  - on peut ajuster cette configuration d'ehcache
  - cependant, le script export-modified-groups-to-LDAP fonctionne en lançant une nouvelle instance de gsh (== grouper) à chaque modification de groupe : le cache ehcache de grouper est alors impuissant
- si votre serveur LDAP dispose d'un espace RAM plus petit que la taille de votre base, alors :
  - LMDB se verra contraint de flusher une partie de la base pour monter la partie qu'il doit lire et qu'il n'avait pas pu monter en RAM faute d'espace ; si vous consultez l'ensemble de la base très régulièrement, vous provoquez ce mécanisme de manière cyclique !
  - en conséquence, vous pouvez alors constater un **usage du disque en lecture (IO READ) très important**.

en LMDB, et notamment avec grouper qui peut solliciter beaucoup le ldap, il est vivement recommandé de fournir assez de RAM au serveur pour que l'ensemble de la base (ou au moins la base "utile") puisse être montée en RAM :

en disposant d'un volume de **RAM supérieur à la taille (effective) du .mdb** (dans /var/lib/ldap/ généralement), on s'assure d'éviter de l'IO (lecture) trop important. Privilégiez également des disques SSD pour les VMs du LDAP (imports LDIFs et réindexations).

Pour avoir la taille effective: ll -sh \*mdb

```
1,9G -rw----- 1 openldap openldap 6,6G nov.  8 12:06 data.mdb
```

```
4,0K -rw----- 1 openldap openldap 8,0K nov.  8 12:08 lock.mdb
```

## Dimensionnement serveurs

Le service Grouper peut fonctionner sur un unique serveur sur lequel on fait tourner

- les services java/tomcat correspondant aux applications web grouper et tâches de fond de synchronisation de groupes avec ldap
- le frontal apache/mod\_shib
- la base de données postgresql

La gestion de synchronisation de calcul et synchronisation de groupes de quelques milliers d'individus peut solliciter beaucoup le serveur ldap mais également le serveur Grouper.

Aussi pour un établissement de quelques dizaines de milliers de personnes, vous pouvez envisager l'usage d'un serveur (VM) avec 4 CPUs et 8 GB de RAM.

Pour l'espace disque, il faut compter simplement quelques GB pour la base de données et quelques GB pour les sources et binaires Grouper.

Pour la disponibilité du service, notre contexte d'usage fait qu'une indisponibilité de Grouper ne permet plus de mettre à jour les groupes simplement.

L'usage des groupes par les applicatifs est en effet porté par LDAP.

Une indisponibilité de Grouper n'est donc pas critique.

## Éléments techniques à installer sur le serveur qui héberge Grouper.

## Java / Tomcat

Les service Grouper correspondent à des applications Java / Tomcat.

Ceux-ci fonctionnent avec un JDK 1.7 et Tomcat 6.

- vous pouvez prendre le JDK 1.7 distribué par Oracle ou plutôt ([recommandation](#)) **la version openjdk 1.7 de votre distribution** ([yum install java-1.7.0-openjdk-devel](#) pour CentOS 7) ... en effet, le JDK 1.7 distribué par Oracle n'est plus disponible/maintenu publiquement (gratuitement).
- pour Tomcat, prenez la dernière version 6 proposée - <https://archive.apache.org/dist/tomcat/tomcat-6/v6.0.39/bin/>

Pour construire les binaires depuis les sources, nous avons également besoin de ANT

### Ant

Si vous avez installé le JDK de votre distribution, installez dans la foulée le Apache Ant de la même façon ([yum install ant](#) pour CentOS 7).

### Java

On édite un fichier `/opt/grouper-env` qui est un fichier d'environnement utilisé par les différents scripts de démarrage de service et par un utilisateur 'grouper' (à créer) sous lequel on travaille (lancement du tomcat, scripts de synchro, client shell grouper).

```
#!/bin/sh
JAVA_HOME=/usr/lib/jvm/java
JAVA_OPTS="-Dgrouper.home=/opt/grouper-esup/grouper -Xms1512m -Xmx1512m -XX:MaxPermSize=512m"
```

Pour faciliter la prise en compte de ce fichier d'environnement dans un environnement bash, on réalise également un fichier `/opt/env.sh` :

```
#!/bin/sh
set -o allexport
source /opt/grouper-env
PATH=$PATH:$JAVA_HOME/bin
set +o allexport
```

Ainsi dans `~grouper/.bashrc` en fin de fichier on ajoute :

```
. /opt/env.sh
```

### Tomcat

Dézippez le tomcat dans `/opt` et faites le appartenir à l'utilisateur 'grouper'.

Un lien symbolique est fait `/opt/tomcat-grouper -> /opt/apache-tomcat-6.0.35` - on utilise ensuite `/opt/tomcat-grouper` comme chemin d'accès au tomcat dans les fichiers de configuration.

Dans `/opt/tomcat-grouper/conf/server.xml` on ajoute dans la balise `Server < Service < Host` les 2 contextes suivants :

```
<Context docBase="/opt/grouper-esup/grouper-ui/dist/grouper" path="/grouper" reloadable="false"/>
<Context docBase="/opt/grouper-esup/grouper-ws/grouper-ws/build/dist/grouper-ws" path="/grouper-ws" reloadable="false"/>
```

Pour que l'authentification avec shibboleth fonctionne (passe), il est important (**requis**) d'ajouter l'attribut **`tomcatAuthentication="false"`** dans le connector utilisé (ajp a priori avec apache devant en proxypass ajp) :

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443" tomcatAuthentication="false"/>
```

## PostgreSQL

Grouper utilise une base de données SQL. Installez le serveur postgresql fourni par votre distribution puis créez une base pour grouper :

```
create database grouper;
create USER grouper with password 'esup';
grant ALL ON DATABASE grouper to grouper;
```

## Apache et mod\_shib

Comme pour la plupart des applications 'shibbolethisées', l'authentification/identification shibboleth est effectuée par Apache et le mod\_shib.

Il faut donc configurer un SP shibboleth prêt à l'emploi pour Grouper et un virtualhost en <https://grouper.univ-ville.fr>

Différentes documentations peuvent vous aider :

- doc et tutos Renater : [https://services.renater.fr/federation/docs/installation#installer\\_un\\_sp\\_shibboleth](https://services.renater.fr/federation/docs/installation#installer_un_sp_shibboleth)
- doc plus concise (mais plus tout à fait à jour par rapport aux versions de distributions) : <https://aresu.dsi.cnrs.fr/spip.php?article188>

Une fois le Virtualhost apache et SP shibboleth paramétré comme il se doit, on doit accéder et tester la bonne configuration de l'ensemble via une url du type : <https://grouper.univ-ville.fr/secure>

Votre fichier de configuration Apache pouvant ressembler à cela :

```
<VirtualHost *:80>

    ServerName grouper.univ-ville.fr
    ServerAlias grouper

    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/access.log combined

    Redirect / https://grouper.univ-ville.fr/

</VirtualHost>

<VirtualHost *:443>

    ServerName grouper.univ-ville.fr
    ServerAlias grouper

    LogLevel warn
    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/access.log combined
    ErrorDocument 503 /503/index.html

    ProxyPass /503 !

    RewriteEngine On
    RewriteRule ^/$ /grouper [L,R,NE]

    SSLEngine on

    ProxyPass /Shibboleth.sso !
    ProxyPass /secure !
    ScriptAlias /secure /var/www/printenv.pl

    <Location /secure>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
    </Location>

    <Location /grouper>
        AuthType shibboleth
        ShibRequestSetting requireSession 1
        require shib-session
        ShibUseHeaders On
    </Location>

    ProxyPass / ajp://localhost:8009/ ttl=10 timeout=3600 retry=1

</VirtualHost>
```

## Grouper ESUP

Le 'fork' ESUP est disponible depuis le repository github Esup <https://github.com/EsupPortail/grouper-esup> - vous noterez qu'il est bien vu comme un fork de Grouper d'Internet2 au niveau de Github.

Il consiste en la configuration de Grouper pour fonctionner de la manière décrite ci-dessus : ldap, shibboleth, postgresql, etc.

Cette version ('fork') ESUP est basée sur le tag GROUPER\_2\_3\_0. Aussi l'usage de git vous permet d'identifier les modifications apportées ("configurations", scripts, traductions ...).

Cette version correspond à une véritable installation de Grouper dans un établissement, nous avons simplement renommé les noms des serveurs, changé les mots de passe, ... ce "paramétrage" que l'on a fait tenir en un seul commit correspond à un travail conséquent de plusieurs établissements que l'on remercie énormément.

## Récupération de la version Grouper d'Esup

```
cd /opt
git clone https://github.com/EsupPortail/grouper-esup.git
```

## Modifications apportées

La commande suivante vous permet d'avoir une bonne idée des fichiers modifiés ou ajoutés par rapport au tag GROUPER\_2\_3\_0 récupéré initialement et utilisé donc comme point de départ :

```
cd /opt/grouper-esup
git diff GROUPER_2_3_0 --name-only | grep -v jar
```

L'interface de github permet également de voir directement ce qui a été modifié, d'autant que tout a été compilé en 1 seul commit : <https://github.com/EsupPortail/grouper-esup/commit/69a6cac9f25e217722997934f8df0210925948f5>

*Vous pouvez aussi regarder en quoi consiste ces modifications véritablement en refaisant un git diff sans le name-only.*

```
grouper-ui/.gitignore
grouper-ui/conf/.gitignore
grouper-ui/conf/grouper-ui.properties
grouper-ui/conf/grouperText/grouper.text.en.us.base.properties
grouper-ui/conf/grouperText/grouper.text.fr.fr.properties
grouper-ui/webapp/WEB-INF/struts-config.xml
grouper-ui/webapp/WEB-INF/web.ajax.xml
grouper-ui/webapp/WEB-INF/web.core.xml
grouper-ui/webapp/grouperExternal/public/assets/images/logo_univ-ville.png
grouper/.gitignore
grouper/bin/export-modified-groups-to-LDAP
grouper/bin/grouper-loader-daemon
grouper/bin/grouper-loader-problem-with-subjectIdentifier.cgi
grouper/bin/gsh
grouper/bin/gsh-psp-bulkSync
grouper/bin/sync-grouper-loader-group-and-export-to-LDAP
grouper/bin/sync-grouper-loader-group-and-export-to-LDAP.cgi
grouper/build.properties
grouper/conf/ehcache.xml
grouper/conf/grouper-loader.properties
grouper/conf/grouper.client.properties
grouper/conf/grouper.hibernate.properties
grouper/conf/grouper.properties
grouper/conf/ldap.properties
grouper/conf/log4j.properties
grouper/conf/morphString.properties
grouper/conf/psp-internal.xml
grouper/conf/psp-resolver.xml
grouper/conf/psp-services.xml
grouper/conf/psp.xml
grouper/conf/server.properties
grouper/conf/sources.xml
grouper/ext/lib/temp.txt
```

Cependant les modifications essentielles à effectuer correspondent aux fichiers qui contiennent simplement un univ-ville.fr ...

```
/opt/grouper-esup$ fgrep -lR univ-ville *
grouper/conf/ldap.properties
grouper/conf/sources.xml
grouper/conf/grouper.properties
grouper/conf/psp-resolver.xml
grouper/conf/psp.xml
grouper/conf/grouper-loader.properties
```

Les modifications des fichiers ci-dessus correspondent à du paramétrage LDAP sauf pour grouper/conf/grouper-loader.properties.



#### grouper-loader.properties

**grouper-loader.properties** permet en effet de paramétrer des bases de données permettant de réaliser des groupes dynamiques par requêtes sql (ces requêtes étant données / paramétrées via l'interface homme machine).

Une fois votre Grouper installé, c'est le seul fichier que vous aurez à modifier de temps en temps pour pouvoir requêter de nouvelles bases de données ; la requête en elle-même est saisie dans l'IHM mais pas les paramètres d'accès aux bases de données ... en tout cas en Grouper 2.3 car en 2.4 il se pourrait que ce soit une nouvelle possibilité de l'IHM de Grouper !

Suivant les bases de données utilisées, vous devrez rajouter vous-même le driver (bibliothèque java jar) dans le répertoire /opt/grouper-esup/grouper/lib/custom/ (driver oracle type ojdbc6\_g.jar par exemple).

Vous aurez également besoin de modifier éventuellement les autres fichiers comme grouper/conf/grouper.hibernate.properties par exemple si vous avez pris un autre mot de passe ou nom de base que ceux indiqués ci-dessus ...

## Jars supplémentaires

L'installation / configuration proposée utilise actuellement Grouper PSP (module de synchronisation de groupes avec ldap).

Pour que cela fonctionne, vous devez récupérer un ensemble de jars à ajouter dans /opt/grouper-esup/grouper/lib/custom/ Cet ensemble de jars se trouve dans le répertoire lib/custom dans le tar-gz suivant : <https://software.internet2.edu/grouper/release/2.3.0/grouper.psp-2.3.0.tar.gz>

## Déploiement

Pour que les modifications de fichiers (configurations) soient prises en compte, en plus de redémarrer les services Grouper (au sens unix / daemon; cf [ci-dessous](#) cela correspond aux services tomcat-grouper.service, grouper-loader-daemon.service et export-modified-groups-to-LDAP.service) vous devez redéployer les briques Grouper.

Vous stoppez les [3 services Grouper](#) puis vous lancez successivement les redéploiements suivants :

```
cd /opt/grouper-esup/grouper-ui/
ant clean dist
cd /opt/grouper-esup/grouper-ws/grouper-ws/
ant clean dist
```

Vous redémarrez les [3 services](#) précédemment stoppés.

*Présent dans les sources Grouper, nous avons ici déployé naturellement grouper-ws ... même si celui-ci est complètement optionnel. Si vous ne vous en servez pas, vous n'êtes donc pas obligé de le déployer et de déclarer son contexte dans la configuration tomcat.*

## Tâches d'exploitation

### Scripts de démarrage

L'installation des services postgresql, apache, sp shibboleth ont dû engendrer l'ajout des scripts de démarrages (initd/systemctl/systemd).

Il faut y ajouter les services installés 'manuellement'. Sous Centos 7 on a ainsi :

- /usr/lib/systemd/system/tomcat-grouper.service

```
# Systemd unit file for tomcat
[Unit]
Description=Apache Tomcat Web Application Container
After=syslog.target network.target
[Service]
Type=forking
EnvironmentFile=/opt/grouper-env
Environment=CATALINA_PID=/opt/tomcat-grouper/temp/tomcat.pid
Environment=CATALINA_HOME=/opt/tomcat-grouper
ExecStart=/opt/tomcat-grouper/bin/startup.sh
ExecStop=/bin/kill -15 $MAINPID
User=grouper
Group=grouper
[Install]
WantedBy=multi-user.target
```

- /usr/lib/systemd/system/grouper-loader-daemon.service

```
# Systemd unit file for grouper-loader-daemon
[Unit]
Description=Synchro incrementale grouperloader (bases de données) vers grouper
After=syslog.target network.target
[Service]
EnvironmentFile=/opt/grouper-env
ExecStart=/opt/grouper-esup/grouper/bin/grouper-loader-daemon
User=grouper
Group=grouper
[Install]
WantedBy=multi-user.target
```

- /usr/lib/systemd/system/export-modified-groups-to-LDAP.service

```
# Systemd unit file for export-modified-groups-to-LDAP
[Unit]
Description=Synchro incrementale grouper vers ldap
After=syslog.target network.target postgresql.service
[Service]
EnvironmentFile=/opt/grouper-env
ExecStart=/opt/grouper-esup/grouper/bin/export-modified-groups-to-LDAP
User=grouper
Group=grouper
[Install]
WantedBy=multi-user.target
```

On active les services :

```
systemctl enable tomcat-grouper.service
systemctl enable grouper-loader-daemon.service
systemctl enable export-modified-groups-to-LDAP.service
```

Vérifiez dans /var/log/messages la bonne exécution de l'ensemble. Pour une bonne mise en oeuvre de ces services, vous pouvez les tester individuellement avec l'utilisateur grouper en les lançant depuis une console directement.

En effet, certains scripts peuvent requérir la présence de certaines bibliothèques supplémentaires : export-modified-groups-to-LDAP est un script perl qui utilise des bibliothèques perl comme DBI ou encore DBD-PG (*yum install perl-DBI perl-DBD-Pg* pour Centos).

## Backup de la BD

Pour sauvegarder votre base de données de manière consistante, ajoutez un dump postgresql dans /etc/crontab simplement pour que le dump soit fait avant le passage de votre robot de sauvegarde.

```
## Backup de la base Grouper
30 01 * * * postgres rm -rf /opt/backup/grouper-dump && pg_dump -b -f /opt/backup/grouper-dump grouper
```

Au préalable on crée le répertoire de backup pour postgres :

```
mkdir /opt/backup && chown -R postgres:postgres /opt/backup
```

## Tâches CRON de synchronisation

Pour assurer les synchronisations au mieux, on ajoute également dans `/etc/crontab` :

```
# Synchro ldap totale toutes les heures - consolide la synchro incrémentale effectuée par export-modified-  
groups-to-LDAP.service  
15 * * * * grouper /opt/grouper-esup/grouper/bin/gsh-psp-bulkSync  
  
# redémarrage de grouper-loader-daemon.service pour prise en compte ajout/modif groupes dynamiques  
00,30 * * * * root systemctl restart grouper-loader-daemon.service
```

## Démarrage de Grouper et prise en main

### Initialisation de la base de données de Grouper

On initialise la base de données avec gsh ainsi :

```
/opt/grouper-esup/grouper/grouper/bin$ ./gsh -registry -check
```

Il vous demande si vous êtes sûr, vous y répondez puis il vous dit de vérifier un script sql et de le passer à gsh, ce que vous faites :

```
/opt/grouper-esup/grouper/grouper/bin$ gsh -registry -runsqlfile /opt/grouper-esup/grouper/ddlScripts  
/grouperDdl_20171122_15_16_30_079.sql
```

Nouvelle question y/n où on répond que l'on est sûr.

Le script initialise la base de données, on peut voir dans postgresql que la base grouper contient plus de 100 tables.

### Peuplement du groupe etc:sysadmingroup

Une fois Grouper démarré, vous pourrez l'utiliser via ses interfaces web directement. Pour ce faire cependant, vous avez besoin de vous déclarer comme administrateur de Grouper. Vous faites cela en démarrant un shell grouper, en créant le groupe configuré comme étant le groupe des administrateurs et en ajoutant votre user à ce groupe :

```
/opt/grouper-esup/grouper/grouper/bin$ ./gsh.sh  
gsh 0% addGroup("etc", "sysadmingroup", "Administrateurs de Grouper")  
gsh 1% addMember("etc:sysadmingroup", "joe@univ-ville.fr")
```

### Démarrage des services

```
systemctl restart tomcat-grouper.service  
systemctl restart grouper-loader-daemon.service  
systemctl restart export-modified-groups-to-LDAP.service
```

## Interface Web

Vous pouvez maintenant utiliser votre Grouper et initier votre arborescence de groupes.

La page wiki des [Contributions d'établissements](#) pourra vous donner des idées.

Un usage simple peut être de scinder vos arborescences de groupes en trois ensemble ainsi :

- un ensemble des groupes constitués dynamiquement et issus de requêtes SQL
- un ensemble de groupes dédiés à des applicatifs spécifiques
- un ensemble de groupes autre qui correspondra aux groupes ne rentrant pas dans les 2 ensembles précédents

Dans un tout premier temps, un groupe à créer peut être le groupe DSI qui sera membre de etc:sysadmingroup par exemple !

## Usage de Grouper

L'objet de ce document n'est pas de faire une documentation de Grouper véritablement, l'idée est juste de faciliter son installation et sa mise en œuvre.

Aussi pour l'usage de Grouper, on vous réfère à la documentation officielle : <https://spaces.internet2.edu/display/Grouper> même si pour un usage sommaire, la nouvelle interface de Grouper en 2.3.0 est plutôt intuitive.

Vous y trouverez cependant les explications nécessaires pour notamment :

- réaliser des groupes dynamiques issus de requêtes sql sur des bases externes : <https://spaces.internet2.edu/display/Grouper/Grouper++Loader>
- réaliser des groupes composites : <https://spaces.internet2.edu/display/Grouper/Grouper+UI+composites>
- accorder des privilèges à un utilisateur ou à un groupe sur **un dossier incluant tous les sous-dossiers et sous-groupes** : <https://spaces.internet2.edu/display/Grouper/Grouper+rules+privileges+inheritance+on+UI>
- purger les membres supprimés dans le ldap mais affectés à des groupes dans grouper ("Entity not found") : <https://spaces.at.internet2.edu/pages/viewpage.action?pageId=14517820>
- ...