

# Installation uPortal v3 sur une centOS

## ⚠ Préambule

Nous décrivons ici l'installation d'un hôte Linux Vserver [http://linux-vserver.org/Welcome\\_to\\_Linux-VServer.org](http://linux-vserver.org/Welcome_to_Linux-VServer.org) sur lequel nous déployons le socle esup-portail.

Le système "invité" est une CentOS 5.4 d'architecture 32 bits, avec un nom inclut dans le DNS de l'établissement et des ACL réseau laissant passer les flux.

Nous supposons que vous possédez une infrastructure en place avec au minimum un DNS, CAS et un serveur openLDAP en production.

Dans ce qui va suivre, notre machine s'appelle **gaspar-dev.int-evry.fr** et a pour adresse IP **157.159.10.200**.

---

- 1/ Firewall
  - # Connexions sortantes:
  - # Connexions entrantes:
- 2/ Installation système de base
  - 2.1/ Récupération des repositories java
    - 2.1.1/ jpackage
    - 2.1.2/ remi
  - 2.2/ Installation packages de base via RPM
  - 2.3/ Configuration spécifique à Linux-VServer
  - 2.4/ Installation des packages java pour ESUP
  - 2.5/ Démarrage automatique de ces services/daemons:
  - 2.6/ Récupération d'esupdev et de uPortal-esup-DLM
  - 2.7/ Ajout d'un compte local et positionnement des droits
  - 2.8/ Mysql
    - 2.8.1/ Authentification et autorisations
    - 2.8.2/ Modification pour régler les problèmes de casse
  - 2.9/ Monitoring nagios (spécifique à mon établissement mais ça peut rappeler que...)
  - 2.10/ Installation de maven
- 3/ Certificats, keystore et autres soucis
  - 3.1/ Création d'un bundle comprenant la chaine de notre PKI et celle de TCS/addTrust
  - 3.2/ Génération de la clé et du certificate request pour TCS
  - 3.3/ Création d'un Keystore pkcs12 et vérification
  - 3.4/ Modification de Tomcat
- 4/ Installation du package esup-portail à proprement parler
  - 4.1/ Décompression des packages
  - 4.2/ Configuration
    - config.properties
    - build.properties
  - 4.3/ Premières tâches ant et création de l'arborescence
  - 4.4/ Modification des droits
  - 4.5/ Configuration de uPortal
  - 4.6/ Utilisation du ant embarqué
  - 4.7/ Deuxièmes tâches ant
  - 4.8/ Création d'un script de lancement automatique
- 5/ On reprend tout suite à l'analyse des différents logs
  - 5.1/ APR
    - Premier essai
    - Second essai
  - 5.2/ Modification de Tomcat
    - 5.2.1/ server.xml
    - 5.2.3/ tomcat-users.xml

---

## 1/ Firewall

Il convient de correctement configurer votre firewall. Dans le cas d'un vserver installé avec les vs-tools (<http://dokuwici.univ-reims.fr/wiki/doku.php?id=public:projets:vs-tools:startvs-tools>), le fichier est accessible sur la machine hôte/physique dans `/etc/vservers/nomduvserver/firewall.conf`.

Dans le cas d'une machine physique traditionnelle, il faut modifier le fichier `/etc/sysconfig/iptables`.

Voici les flux à laisser transiter à travers votre firewall

# Connexions sortantes:

```

allow to 0/0 to udp/53      # DNS request
allow to 0/0 to udp/123    # Network Time Protocol
allow to 157.159.0.0/16 to tcp/22 # SSH vers le réseau de l'établissement
#
# Récupération de packages
allow to 0/0 to tcp/21      # FTP
allow from 0/0 from tcp/20 # FTP-data
allow to 0/0 to tcp/80      # HTTP YUM
allow to 0/0 to tcp/11371   # PGP-keys
#
# Portail
allow to 157.159.10.29/32 to tcp/443 # Accès https au CAS de l'établissement
allow to 157.159.10.76/32 to tcp/3306 # Requêtes mysql vers un serveur central (pas nécessaire a priori)
allow to 157.159.0.0/16 to tcp/389   # LDAP (tcp/636 pour du LDAPS)
allow to 157.159.0.0/16 to tcp/25    # SMTP
allow to 0/0 to tcp/8443             # HTTPS tomcat pour shibboleth
allow to 0/0 to tcp/443             # HTTPS pour shibboleth
allow to 0/0 to tcp/445             # CIFS stockage
allow to 0/0 to tcp/137             # CIFS stockage
allow to 0/0 to tcp/138             # CIFS stockage
allow to 0/0 to tcp/139             # CIFS stockage
allow to 157.159.10.88/32 to tcp/143 # Accès au serveur de boîtes aux lettres pour Canal IMAP

```

# Connexions entrantes:

```

allow from 0/0 to tcp/80          # HTTP
allow from 157.159.0.0/16 to tcp/8080 # HTTP tomcat
allow from 0/0 to tcp/443        # HTTPS
allow from 0/0 to tcp/8443       # HTTPS tomcat
allow from 0/0 to tcp/22         # Local SSH
allow from 0/0 to udp/161        # SNMP

```

## 2/ Installation système de base

### 2.1/ Récupération des repositories java

#### 2.1.1/ jpackage

```

cd /etc/yum.repos.d/
wget http://www.jpackage.org/jpackage50.repo
wget http://www.jpackage.org/jpackage.asc
rpm --import jpackage.asc

```

#### 2.1.2/ remi

```

wget http://download.fedora.redhat.com/pub/epel/5/i386/epel-release-5-3.noarch.rpm
wget http://rpms.famillecollet.com/RPM-GPG-KEY-remi
rpm --import RPM-GPG-KEY-remi
wget http://rpms.famillecollet.com/enterprise/remi-release-5.rpm
rpm -ihv remi-release-5.rpm

```

### 2.2/ Installation packages de base via RPM

On vérifie qu'on dispose que de packages 32 bits:

```
rpm -qa --queryformat %-{name}-{version}-{release}.{arch}"\n" | grep x86_64
```

On récupère via yum quelques packages utiles:

```
yum install vim-enhanced rootfiles vixie-cron anacron logrotate system-config-date sudo wget tar gzip diffutils
yum install which bind-utils
yum install openssh.i386 openssh-clients.i386 openssh-server.i386
yum install dos2unix
yum install mlocate
yum install yum-utils
yum install rpm-build
```

On met à jour mlocate afin de retrouver facilement un fichier:

```
updatedb
```

## 2.3/ Configuration spécifique à Linux-VServer

On met le système à l'heure:

```
[root@gaspar-dev ~]# system-config-date
```

On vérifie:

```
[root@gaspar-dev ~]# date
```

Modification de fichiers PAM (/etc/pam.d/crond et /etc/pam.d/sshd) afin de rester dans le contexte du vservers. Dans les deux cas, on met en commentaire la ligne suivante:

```
#session    required    pam_loginuid.so
```

Il peut survenir des soucis avec la localhost sur les vservers (plusieurs localhosts sur la même machine physique). On modifie /etc/hosts en conséquence:

```
[root@gaspar-dev ~]# more /etc/hosts
157.159.10.200      gaspar-dev.int-evry.fr localhost
```

## 2.4/ Installation des packages java pour ESUP

```
yum install tomcat6
yum install ant-nodeps
yum install tomcat6-admin-webapps.noarch tomcat6-docs-webapp.noarch tomcat6-webapps.noarch
yum update tomcat6.noarch tomcat6-javadoc.noarch velocity
yum install java-1.6.0-sun-jdbc ant-apache-log4j
```



### Attention

Afin de ne pas obtenir une erreur de type `"/usr/bin/rebuild-security-providers is needed by package java-1.4.2-gcj-compat.."`, éditez /etc/yum.repos.d/jpackage50.repo et mettez le repository [jpackage-generic](#) à `enabled=0`.

Une fois ceci fait, vous pouvez continuer à installer les packages java et les packages nécessaires au fonctionnement du portail:

```
yum install ant java-sun java-sun-devel
yum install ant-apache-regexp ant-contrib
yum install java-1.6.0-sun-fonts java-1.6.0-openjdk-devel java-1.6.0-openjdk
yum install httpd mod_ssl
yum install mysql-server
```

## 2.5/ Démarrage automatique de ces services/daemons:

```
chkconfig --level 345 mysqld on
chkconfig --level 345 tomcat6 on
chkconfig --level 345 httpd on
```

## 2.6/ Récupération d'esupdev et de uPortal-esup-DLM

```
cd ~
wget http://sourcesup.cru.fr/frs/download.php/2874/esupdev-3.1.2-esup-0.5-RC3.tar.gz
wget http://sourcesup.cru.fr/frs/download.php/2875/esup-3.1.2-esup-0.5-RC3.tar.gz
```

Évidemment, la version peut avoir changé au moment où vous lisez ce How-To...

Référez vous à [http://sourcesup.cru.fr/frs/?group\\_id=173](http://sourcesup.cru.fr/frs/?group_id=173) .

## 2.7/ Ajout d'un compte local et positionnement des droits

Ajout du compte esup:

```
useradd esup
```

Copie du package esup et esupdev dans le répertoire du compte local:

```
cd ~
cp esup-3.1.2-esup-0.5-RC3.tar.gz /home/esup/.
cp esupdev-3.1.2-esup-0.5-RC3.tar.gz /home/esup/.
```

Positionnement des droits:

```
cd /home/esup/
chgrp -R apache .bash_logout .bash_profile .bashrc .mozilla
chown esup:tomcat esup-3.1.2-esup-0.5-RC3.tar.gz esupdev-3.1.2-esup-0.5-RC3.tar.gz
```

## 2.8/ Mysql

### 2.8.1/ Authentification et autorisations

```

/etc/init.d/mysqld start
/usr/bin/mysqladmin -u root password 'motdepasse'
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.77 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> create database esup;
mysql> GRANT ALL PRIVILEGES ON esup.* TO 'esup'@'localhost' IDENTIFIED BY 'motdepasse' WITH GRANT OPTION;
mysql> GRANT ALL PRIVILEGES ON esup.* TO 'esup'@'gaspar-dev.int-evry.fr' IDENTIFIED BY 'motdepasse' WITH GRANT OPTION;

```

## 2.8.2/ Modification pour régler les problèmes de casse

Pour citer Vincent Repain:

---

uPortal 3 ne supporte pas les bases mySQL en case sensitive : <http://www.ja-sig.org/wiki/display/UPM31/03+MySQL+Configuration> parce qu'ils ne garantissent pas que toutes les requêtes soient écrites avec la même casse. On s'en aperçoit très vite dès l'init de la base : les scripts Cernunnos sont écrits en minuscules. Si on veut passer en case insensitive sur un serveur hébergeant déjà des bases, il faut migrer les bases (il y a en effet transformation physique des noms de tables en minuscules) :

- Faire un dump de chaque base
- Changer la config mySql pour passer en lowercase
- Réimporter chaque base
- Observer les résultats

---

Ici, on crée les bases MySQL, on ne se soucie donc pas de bases existantes. Voici un lien qui décrit comment faire: <http://dev.mysql.com/doc/refman/5.0/fr/name-case-sensitivity.html>.

Typiquement, il suffit d'écrire dans `my.cnf.txt/etc/my.cnf` :

```
lower_case_table_names=1
```

Et de relancer mysql:

```
/etc/init.d/mysqld restart
```

## 2.9/ Monitoring nagios (spécifique à mon établissement mais ça peut rappeler que...)

```

yum install nrpe
yum install nagios-of-plugins nagios-plugins-ntp nagios-plugins-load nagios-plugins-users nagios-int-plugins
nagios-plugins nagios-plugins-disk nagios-plugins-perl nagios-plugins-procs
vi /etc/nagios/nrpe.cfg
chkconfig --level 345 nrpe on
/etc/init.d/nrpe start

```

## 2.10/ Installation de maven

Après téléchargement (<http://maven.apache.org/>) ou récupération de maven via "ant init":

```
cp apache-maven-2.2.1-bin.tar.gz /usr/local/.
cd /usr/local/
gunzip apache-maven-2.2.1-bin.tar.gz
tar xvf apache-maven-2.2.1-bin.tar
rm apache-maven-2.2.1-bin.tar
chmod +x /usr/local/apache-maven-2.2.1/bin/mvn
```

En tant qu'utilisateur esup, il faut rajouter ceci à votre path:

```
su - esup
vi /home/esup/.bash_profile

...
# User specific environment and startup programs
PATH=$PATH:$HOME/bin:/usr/local:/usr/local/apache-maven-2.2.1/bin
```

### 3/ Certificats, keystore et autres soucis

On va supposer que votre établissement à obtenu l'agrément afin de se faire signer par Terena Certificate Service (TCS). Pour en savoir plus sur TCS, allez par exemple à: <http://www.cru.fr/services/tcs/index>.

Dans notre établissement, on a également une PKI (tout ce qui contient les caractères "it" ou "tmstp" dans la suite).

On va créer un keystore qui comprend la chaîne de notre PKI et celle de Terena. Si ce n'est pas votre cas, ça ne peut être que plus simple 😊



#### Attention

Il faut que votre serveur CAS soit également signé par Terena.

#### 3.1/ Création d'un bundle comprenant la chaîne de notre PKI et celle de TCS/addTrust

On récupère la chaîne TCS/addtrust à [http://www.cru.fr/\\_media/services/tcs/cachain.pem](http://www.cru.fr/_media/services/tcs/cachain.pem)

```
[root@gaspar-dev /etc/pki/tls/certs] wget http://www.cru.fr/_media/services/tcs/cachain.pem
```

On récupère et met les certificats de notre PKI (ca-chain-tmstpca2\_itca2\_itrootca2.crt, tmstpca.crt.1, cat itca.crt.1 et cat itrootca.crt.1) dans le même répertoire. On concatène le tout dans le bundle:

```
[root@gaspar-dev /etc/pki/tls/certs] ( cat ca-chain-tmstpca2_itca2_itrootca2.crt ; cat tmstpca.crt.1 ; cat itca.crt.1 ; cat itrootca.crt.1 ; cat cachain-tcs-addtrust.pem ) > ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.crt
```

#### 3.2/ Génération de la clé et du certificate request pour TCS

Dans notre cas, on va sur un autre serveur afin de générer la clé et le csr (certificate request).

On crée ceci sur un autre serveur pour des raisons "historiques" car c'est celui qui contient notre PKI. Le logiciel qu'on utilise est CSP dont voici quelques références:

<http://www.rajeevnet.com/crypto/ca/ca-paper.html>

<http://devel.it.su.se/pub/jsp/polopoly.jsp?d=1026&a=3290>

On crée donc la clé privée (protégée par un mot de passe) et le csr via openssl:

```
[pkits@pki ~/tcs-tmstp]$ openssl req -config opensslTCS-TMSP-gaspardev.cnf -new -extensions v3_req -keyout private/gaspardev_pass.key -out req/gaspardev.req -days 720
```

Dans le fichier de configuration [opensslTCS-TMSP-gaspardev.cnf](#) (en pièce jointe), les points importants sont:

```
nsCertType = server, client
keyUsage = nonRepudiation, digitalSignature, keyEncipherment, dataEncipherment
extendedKeyUsage = serverAuth, clientAuth
subjectAltName = @altnamesgaspardev
\[ altnamesgaspardev \]
DNS.1 = "gaspar-dev.it-sudparis.eu"
```

On récupère la clé sur le serveur où tournera uPortal, par exemple en utilisant scp:

```
[root@gaspar-dev /etc/pki/tls/certs] scp root@carthage:/vservers/pki/home/pkitcs/tcs-tmsp/private
/gaspardev_pass.key .
```

On retire la passphrase de la clé privée:

```
openssl rsa -in gaspardev_pass.key -out gaspardev.key
```

On soumet à TCS notre csr et, après approbation, on récupère le certificat signé et la chaîne de certification. Dans notre cas, il s'agit respectivement de cert-593-gaspar-dev.int-evry.fr.pem et chain-593-gaspar-dev.int-evry.fr.pem.

On copie le tout aux emplacements traditionnels:

```
cp gaspardev.key /etc/pki/tls/private/.
cp cert-593-gaspar-dev.int-evry.fr.pem /etc/pki/tls/certs/.
cp chain-593-gaspar-dev.int-evry.fr.pem /etc/pki/tls/certs/.
```

### 3.3/ Création d'un Keystore pkcs12 et vérification

On crée le keystore pkcs12:

```
openssl pkcs12 -export -in cert-593-gaspar-dev.int-evry.fr.pem -inkey gaspardev.key -out gaspardev_ca-pkiit1-
pkiit2-chain-tcs-addtrust-bundle.p12 -name tomcat -CAfile ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.crt -
caname root -chain
```

On vérifie ce que ça donne:

```
keytool -list -storetype pkcs12 -keystore gaspardev_ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.p12 -v
```

On copie ce keystore où il convient:

```
cp gaspardev_ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.p12 /etc/pki/tls/certs/.
```

### 3.4/ Modification de Tomcat

Le fichier à modifier est /etc/tomcat6/server.xml :

```
<Connector port="8443" maxHttpHeaderSize="8192"
    maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
    enableLookups="false" disableUploadTimeout="true"
    acceptCount="100" scheme="https" secure="true"
    keystoreFile="/etc/pki/tls/certs/gaspardev_ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.p12"
    keystorePass="motdepasse" keystoreType="pkcs12"
    clientAuth="false" sslProtocol="TLS" />
```

De la documentation peut être trouvée à <http://tomcat.apache.org/tomcat-6.0-doc/ssl-howto.html> .

## 4/ Installation du package esup-portail à proprement parler



### Attention

Faites tout ce qui suit en tant que l'utilisateur esup (su - esup)

### 4.1/ Décompression des packages

```
cd
gunzip esup-3.1.2-esup-0.5-RC3.tar.gz
tar xvf esup-3.1.2-esup-0.5-RC3.tar
mkdir Archive
gzip -9 esup-3.1.2-esup-0.5-RC3.tar
mv esup-3.1.2-esup-0.5-RC3.tar.gz Archive/.
```

### 4.2/ Configuration

```
cd esup-3.1.2-esup-0.5-RC3
cp config.sample.properties config.properties
cp build.sample.properties build.properties
```

config.properties

Les parties modifiées du fichier config.properties sont en mauve ci-dessous.

Pour des tests, on utilise dans un premier temps le ant, le maven et le tomcat embarqué (3 premières valeurs fixées à "true" ci-dessous). Par ailleurs, pour la même raison, on utilise la base hsql et non mysql. Pour utiliser mysql, il suffit de commenter/décommenter ce qui se trouve en mauve dans la section "DB Options" de ce fichier. Attention, il est normal que la base hsql est comme login "sa" et pas de mot de passe...

```
# Components Options
use.embedded.ant=true
use.embedded.maven=true
use.embedded.tomcat=true
# Tomcat Options
tomcat.update=true
tomcat.reloadable=true
tomcat.port.shutdown=8005
tomcat.port.http=8080
tomcat.port.jk=8009
# URL Options
esup.public.host=gaspar-dev.int-evry.fr
esup.public.proto=http
esup.public.port=:8080
esup.public.uri=/uPortal
esup.public.guest.uri=/Guest
esup.public.ressource.uri=/ResourceServingWebapp
esup.real.host=${esup.public.host}
esup.real.uri=${esup.public.uri}
esup.real.port.https=
```



```

# LDAP Options
esup.ldap.auth=false
esup.ldap.url=ldap://ldap4.int-evry.fr:389 ldap://ldap3.int-evry.fr:389
esup.ldap.baseDN=dc=int-evry,dc=fr
esup.ldap.bindDN=cn=mcibind,ou=system,dc=int-evry,dc=fr
esup.ldap.bindPasswd=motdepassebindldap
esup.ldap.persondirs.use=true
esup.ldap.uid=uid
# CAS Options
esup.cas.auth=true
esup.cas.proto=https
esup.cas.proxy=true
esup.cas.host=cas.it-sudparis.eu
esup.cas.port.https=
esup.cas.port.http=
esup.cas.uri=/cas
esup.cas.withoutSSL=false
esup.cas.uri.login=/login
esup.cas.uri.validate=/serviceValidate
esup.cas.uri.proxy=/proxy
esup.cas.uri.logout=/logout
esup.real.uri.callbackCas=/CasProxyServlet
# DB Options
esup.db.auth=true
esup.db.persondirs.use=true
esup.db.username=sa
esup.db.password=
#esup.db.username=esup
#esup.db.password=motdepassemysql
# db type : mysql, hsql, postgresql
esup.db.type=hsql
esup.db.url=jdbc:hsqldb:hsql://localhost:8887
esup.db.className=org.hsqldb.jdbcDriver
#esup.db.type=mysql
#esup.db.className=com.mysql.jdbc.Driver
#esup.db.jdbcDriverJar=mysql-connector-java-5.1.6
#esup.db.db-version=5.0.77
#esup.db.poolPreparedStatements=true
# XSL Options
esup.xsl.debug=false
esup.xsl.cache=false
esup.xsl.help.url=http://www.esup-portail.org/display/ESUPMU/Documentations+%28esup-3%29
esup.xsl.help.login=http://www.esup-portail.org/display/ESUPMU/Documentations+%28esup-3%29
esup.xsl.use.sidebar=false
esup.xsl.use.sidebar.focused=false
esup.xsl.use.sidebar.guest=false
# LOG Options
esup.log.dir=${esup.root}/logs
esup.log.file=${esup.log.dir}/portal.log
esup.log.size=500000KB
esup.log.level=DEBUG
# STATS Options
esup.stats.file=${esup.log.dir}/stats.log
esup.stats.attributeType=eduPersonPrimaryAffiliation
esup.stats.channelFnames=all
esup.stats.size=200000KB
# MISC Optionsesup.title.main=eSup DEVEL ${uportal.ver} - [${esup.host.logicalName}]
esup.host.logicalName=gaspar-dev
esup.session.lifetime=30
esup.languages=fr_FR,en_US
esup.default.language=fr
esup.upload-maxsize=300000
esup.ws.groups=trueesup.monitor=true
esup.userprefs.save=false
esup.keystore=/etc/pki/tls/certs/gaspardev_ca-pkiit1-pkiit2-chain-tcs-addtrust-bundle.p12

```

#### build.properties

On utilise là encore le tomcat embarqué afin de faire des tests. Pour utiliser le tomcat v.6 récupéré, il faut commenter/décommenter le "esup.deploy" et "server.home" de build.properties comme ci-dessous.

```
#java_home=/usr/java/jdk1.6.0_16
java_home=/usr/lib/jvm/java
# CURRENT FOLDER
esup.root=${basedir.conv}
esup.base=${esup.root}/Portail
ant_home=${basedir.conv}/resources/${ant.package.final}
mvn_home=${esup.base}/${mvn.package.final}
esup.sources=${esup.base}/uPortal_rel-${uportal.ver}
esup.deploy=${esup.base}/webapps
#esup.deploy=/var/lib/tomcat6/webapps
server.home=${esup.base}/apache-tomcat-${tomcat.ver}
#server.home=/usr/share/tomcat6
server.temp=${server.home}/temp
esup.custom=${esup.root}/custom
esup.update=${esup.root}/update
esup.resources=${basedir.conv}/resources
esup.packages=${esup.root}/packages
esup.tmp=${esup.root}/temp
config.file=${esup.root}/config.properties
mvn_repo=${esup.base}/maven-repo
mvn_offline=false
mvn_skiptest=true
mvn_esuprepo=${esup.resources}/esup-maven-repo
```

### 4.3/ Premières tâches ant et création de l'arborescence

Les options de ant:

```
ant -p
```

On lance la séquence décrite par le consortium:

```
cd ~/esup-3.1.2-esup-0.5-RC3
ant getcomponents
ant unzip
```

### 4.4/ Modification des droits

```
ls logs/
ls -al logs
touch logs/portal.log
touch logs/stats.log
```

### 4.5/ Configuration de uPortal

```
cd /home/esup/esup-3.1.2-esup-0.5-RC3/Portail/uPortal-3.1.1
vi build.properties
```

Il faut modifier les paramètres server.webapps et maven.home de ce fichier [build.properties](#) .

Allez ensuite dans le répertoire où se situe ant:

```
cd ../../..
```

## 4.6/ Utilisation du ant embarqué

```
cd /home/esup/esup-3.1.2-esup-0.5-RC3/resources/
gunzip apache-ant-1.7.1-bin.tar.gz
tar xvf apache-ant-1.7.1-bin.tar
rm apache-ant-1.7.1-bin.tar
cp ant.sample.sh ant.sh
vi ant.sh

----
#!/bin/sh

export LANG=fr_FR
# export JAVA_HOME=/usr/java/jdk1.6
export JAVA_HOME=/usr/lib/jvm/java
export ANT_HOME=./resources/apache-ant-1.7.1

# chmod 755 $ANT_HOME/bin/ant
$ANT_HOME/bin/ant $@

----
chmod u+x ant.sh
dos2unix ant.sh
```

## 4.7/ Deuxièmes tâches ant

```
cd /home/esup/esup-3.1.2-esup-0.5-RC3
./ant.sh init
./ant.sh getcomponents
./ant.sh unzip
```

Il faut faire un choix de base de données. Nous avons choisi mysql mais pour commencer les tests, on va prendre hsql. Pour ceux qui ne connaissent pas, hsqldb est une base, écrite en java, embarquée dans l'application. Ca a la réputation d'être plus rapide et de taille moins importante que mysql. En effet, toutes les données sont écrites en mémoire pour améliorer la vitesse d'accès mais ça ne permet donc pas le traitement d'énormes bases de données. Lors de la fermeture de la connexion à la base de données, hsql va écrire toutes ses données volatiles dans des fichiers (il est cependant aussi possible de créer des tables en dur sur le disque).

Si vous voulez en savoir plus sur le sujet, consultez <http://baptiste-wicht.developpez.com/tutoriel/java/hsql/> .

Dans une fenêtre séparée (ceci sert à lancer la base de données hsqldb):

```
./ant.sh hsql.start
```

Dans la première fenêtre:

```
./ant.sh db.test
./ant.sh db.init
./ant.sh deploy
```

## 4.8/ Création d'un script de lancement automatique

```

vi ~/restart_esup.sh
----
#!/bin/bash
#
/usr/bin/sudo /etc/init.d/tomcat6 stop
cd ~/esup-3.1.2-esup-0.5-RC3
./ant.sh init
./ant.sh db.import -Dchannel=all
./ant.sh db.import -Dusername=monuser-lo
./ant.sh deploy
/usr/bin/sudo /etc/init.d/tomcat6 start
cd
vi /etc/sudoers
chmod u+x restart_esup.sh
~/restart_esup.sh
----

```

## 5/ On reprend tout suite à l'analyse des différents logs

### 5.1/ APR

Dans /var/log/tomcat6/catalina.out, on voit au lancement de Tomcat une phrase un rien inquiétante si la mise en production du portail presse:

```

INFO: The APR based Apache Tomcat Native library which allows optimal performance in production environments
was not found on the java.library.path: ...

```

Il faut installer la "bibliothèque APR" (a priori seuls les 5 derniers packages sont nécessaires):

```

yum install apr-util apr-util-devel apr apr-devel libapreq2 libapreq2-devel tomcat-native openssl-devel

```

Argh, tomcat-native est en version 1.1.8 alors qu'il faut une version 1.1.9 😞

Il faut donc l'effacer avec:

```

yum remove tomcat-native

```

Il faut récupérer le source de tomcat-native et le recompiler.

### Premier essai

```

wget http://archive.apache.org/dist/tomcat/tomcat-connectors/native/1.1.9/source/tomcat-native-1.1.9-src.tar.gz
tar xvf tomcat-native-1.1.9-src.tar.gz
cd tomcat-native-1.1.9-src/jni/native/
./configure --with-apr=/usr/bin/apr-1-config --with-java-home=/usr/lib/jvm/java --with-ssl=yes
make
make install

```

Modification de /home/esup/esup-3.1.2-esup-0.5-RC3/env.sh dans env.sh avec rajout des lignes (en dessous de ANT\_HOME):

```

export CATALINA_HOME=/usr/share/tomcat6
export LD_LIBRARY_PATH=/usr/local/apr/lib

```

Ca ne marche toujours pas 😞

## Second essai

On va récupérer un package source de Fedora, le recompiler et l'installer. C'est pour ça qu'on a installé au départ le package "rpm-build"..

```
wget ftp://ftp.muug.mb.ca/mirror/fedora/linux/development/source/SRPMS/tomcat-native-1.1.19-1.fc13.src.rpm
rpm -ivh --nomd5 tomcat-native-1.1.19-1.fc13.src.rpm
cd /usr/src/redhat/SPECS/
rpmbuild -bb tomcat-native.spec
cd ../RPMS/i386/
rpm -ivh tomcat-native-1.1.19-1.i386.rpm
```

Voici ce qu'on voit au redémarrage de Tomcat dans catalina.out:

```
12 févr. 2010 17:14:30 org.apache.catalina.core.AprLifecycleListener init
GRAVE: An incompatible version 1.1.9 of the APR based Apache Tomcat Native library is installed, while Tomcat
requires version 1.1.17
```

Bon, on va repartir sur la version donnée avec la distribution alors (1.1.18)..

```
rpm -e tomcat-native-1.1.19-1
yum install tomcat-native
\[root@gaspar-dev /\]# rpm -qa |grep native
tomcat-native-1.1.18-1.el5
```

Voici ce qu'on voit au redémarrage de Tomcat dans catalina.out:

```
12 févr. 2010 17:24:13 org.apache.catalina.core.AprLifecycleListener init
INFO: An older version 1.1.18 of the APR based Apache Tomcat Native library is installed, while Tomcat
recommends version greater than 1.1.19
```

## 5.2/ Modification de Tomcat

Je m'inspire ici de la documentation se trouvant à [http://www.esup-portail.org/display/ESUPMU/Tomcat en production](http://www.esup-portail.org/display/ESUPMU/Tomcat+en+production) .

### 5.2.1/ server.xml

### 5.2.3/ tomcat-users.xml