

# Documentation finale - Installation serveur AudioVidéoCast sur RedHat RHEL6



Cette documentation n'a pas encore été testée "from scratch". Par conséquent, si vous identifiez des manques ou des erreurs ou que vous avez des suggestions à faire, merci de les envoyer à [nicolas.truchaud@univ-lyon2.fr](mailto:nicolas.truchaud@univ-lyon2.fr).

- 1. Installation de la distribution
  - a. Installation des packages nécessaires au fonctionnement du site :
  - b. Paquets pour l'encodage
    - Ajoût du RPMforge
    - Install lame
    - Install ffmpeg
    - Install eyeD3
    - Install ffmpeg2theora
    - Install yamdi
    - Install gcc-c++
    - Install atomicparsley
  - c. Modification des presets ipod
- 2. Paramétrage du ftp
  - Créer un répertoire à la racine du serveur :
  - Ajouter un nouvel utilisateur du système « ftpuser » :
  - Des sous-répertoires doivent également être présents dans le répertoire FTP:
  - Modifier les lignes de /etc/vsftpd/vsftpd.conf :
- 3. Récupération des sources
- 4. Création de la base de données
- 5. Configuration de tomcat
  - Configuration du tomcat-users.xml
  - Configuration du server.xml :
  - Configuration des politiques :
- 6. Stockage des répertoires de cours
- 7. Déploiement de l'application AudioVideoCast :
- 8. Configuration APACHE
  - a. Installation des modules
  - b. Configuration d'apache
  - c. Connexion avec un sous-répertoire : [http://avcast-test.univ-lyon2.fr/univ-r\\_av/](http://avcast-test.univ-lyon2.fr/univ-r_av/)
- 9. Activation du streaming
- 10. Séparer l'encodage des médias
- 11. Suppression des tests
- 12. Configuration du live
- 13. Page d'administration
- 14. Annexes
  - a. Personnalisation de l'interface
  - b. Détail du fichier "univrav.properties"
  - c. Détail du filtre CAS du fichier "web.xml"

## 1. Installation de la distribution

### a. Installation des packages nécessaires au fonctionnement du site :

```
$ yum install postgresql pgadmin3 tomcat6 httpd vsftpd zip unzip mailx libpg-java  
libcommons-fileupload-java libcommons-io-java libcommons-lang-java  
libcommons-beanutils-java libcommons-collections3-java libcommons-logging-java ant  
subversion liblog4j1.2-java openjdk-6-jdk
```

### b. Paquets pour l'encodage

Les paquets nécessaires pour l'encodage sont :

```
vorbis-tools lame ffmpeg libavcodec-extra-53 eyeD3 ffmpeg2theora yamdi zip unzip python2.6 python-report  
python-psycopg2 atomicparsley
```

Un certain nombre sont inclus par défaut dans la distribution :

```
$ yum install vorbis-tools libavcodec-extra-53 zip unzip
```

D'autres non, il convient donc de les installer comme suit.

## Ajout du RPMforge

Documentation : <http://repoforge.org/use/>

```
$ rpm -Uhv
http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
```

## Install lame

Documentation : <http://lame.cvs.sourceforge.net/viewvc/lame/lame/USAGE>

```
$ yum install ffmpeg
```

### Test

```
$ wget http://www.villagegeek.com/downloads/webwavs/adios.wav
$ lame -b128 adios.wav adios.mp3

LAME 3.99.5 64bits (http://lame.sf.net)
polyphase lowpass filter disabled
Encoding adios.wav to adios.mp3
Encoding as 11.025 kHz single-ch MPEG-2.5 Layer III (1.4x) 64 kbps qval=3
  Frame      | CPU time/estim | REAL time/estim | play/CPU |   ETA
  65/65   (100%)|   0:00/   0:00|   0:00/   0:00|  113.20x|   0:00
-----
-----
-----
  kbps      mono %      long switch short %
  64.0      100.0      80.0  12.3   7.7
Writing LAME Tag...done
ReplayGain: -5.1dB
```

## Install ffmpeg

Documentation : <http://www.ffmpeg.org/ffmpeg.html>

```
$ yum install ffmpeg
```

### Test

```
$ wget
http://mirrorblender.top-ix.org/peach/bigbuckbunny_movies/big_buck_bunny_720p_surro
und.avi
$ ffmpeg -i big_buck_bunny_720p_surround.avi -acodec aac -ab 128kb -strict
experimental -vcodec mpeg4 -b 1200kb -mbd 2 -cmp 2 -subcmp 2 -s 320x180
final_video.mp4
```

## Install eyeD3

Documentation : <http://eyed3.nicfit.net/#documentation-index>

```
$ yum install python-eyed3.noarch
```

#### Test

```
// Avant : afficher métadonnées du fichier
$ eyeD3 adios.mp3

eyed3.plugins:WARNING: Plugin '('statistics.py',
'/usr/lib/python2.6/site-packages/eyed3/plugins')' requires packages that are not
installed: cannot import name Counter
adios.mp3      [ 26.94 KB ]
-----
Time: 00:03      MPEG2, Layer III      [ 64 kb/s @ 11025 Hz - Mono ]
-----
No ID3 v1.x/v2.x tag found!

// affectation de métadonnées
$ eyeD3 -a MonArtiste -A "MonAlbum" -t "MonMorceau" -n 4 adios.mp3

// résultat :
eyed3.plugins:WARNING: Plugin '('statistics.py',
'/usr/lib/python2.6/site-packages/eyed3/plugins')' requires packages that are not
installed: cannot import name Counter
adios.mp3      [ 28.03 KB ]
-----
Setting artist: MonArtiste
Setting album: MonAlbum
Setting title: MonMorceau
Time: 00:03      MPEG2, Layer III      [ 64 kb/s @ 11025 Hz - Mono ]
-----
ID3 v2.4:
title: MonMorceau
artist: MonArtiste
album: MonAlbum
track: 4
Writing ID3 version v2.4
-----
```

## Install ffmpeg2theora

Documentation : <http://v2v.cc/~j/ffmpeg2theora/examples.html>

```
$ wget http://v2v.cc/~j/ffmpeg2theora/ffmpeg2theora-0.29.linux64.bin
$ install -m 755 ffmpeg2theora-0.29.linux64.bin /usr/local/bin/ffmpeg2theora
```

#### Test

```
$ wget
http://sourceforge.net/projects/libdv/files/examples/examples/pond.dv/download
$ ffmpeg2theora pond.dv

[dv @ 0x2d84b20] Estimating duration from bitrate, this may be inaccurate
Input #0, dv, from 'pond.dv':
  Duration: 00:00:30.53, start: 0.000000, bitrate: 28771 kb/s
    Stream #0:0: Video: dvvideo, yuv411p, 720x480 [SAR 8:9 DAR 4:3], 28771 kb/s,
    29.97 tbr, 29.97 tbn, 29.97 tbc
    Stream #0:1: Audio: pcm_s16le, 44100 Hz, 2 channels, s16, 1411 kb/s
    Pixel Aspect Ratio: 0.89/1   Frame Aspect Ratio: 1.33/1
[swscaler @ 0x2db84a0] Warning: data is not aligned! This can lead to a speedloss
0:00:30.53 audio: 81kbps video: 6975kbps, time elapsed: 00:00:40
0:00:30.53 audio: 81kbps video: 6975kbps, time elapsed: 00:00:40
```

## Install yamdi

Documentation : <http://yamdi.sourceforge.net/>

```
$ yum install yamdi
```

### Test

```
$ wget
http://developer.longtailvideo.com/trac/export/944/trunk/html5/test/files/bunny.flv
$ yamdi -i bunny.flv -o bunny_with_metadata.flv - "John Doe"
```

## Install gcc-c++

```
$ yum install gcc-c++
```

## Install atomicparsley

Documentation : <http://atomicparsley.sourceforge.net/> & <https://bitbucket.org/wez/atomicparsley>

```
$ wget https://bitbucket.org/wez/atomicparsley/get/9183fff907bf.zip
$ unzip 9183fff907bf.zip
$ cd wez-atomicparsley-9183fff907bf/
$ sudo yum install automake
```

### Test

```
???
```

## c. Modification des presets ipod

```
$ sed -i 's/^profile/vprofile/' /usr/share/ffmpeg/libx264-ipod320.ffpreset
$ sed -i 's/^profile/vprofile/' /usr/share/ffmpeg/libx264-ipod640.ffpreset
```

## 2. Paramétrage du ftp

Créer un répertoire à la racine du serveur :

```
$ mkdir /audiovideocours
```

Ajouter un nouvel utilisateur du système « ftpuser »:

```
$ useradd -m -d /audiovideocours/ftp ftpuser  
$ passwd ftpuser
```

Des sous-répertoires doivent également être présents dans le répertoire FTP:

```
$ mkdir -p /audiovideocours/ftp/canceled /audiovideocours/ftp/client_update  
/audiovideocours/ftp/live /audiovideocours/ftp/releases
```

- canceled : sert à stocker les cours enregistrés par le client mais annulés avant la publication.
- client\_update : permet de stocker la dernière version du client d'enregistrement des cours, afin qu'il se mette à jour automatiquement dans les amphis.
- live : est le répertoire de stockage des diapositives d'un cours en direct envoyées par FTP.
- releases : stocke les programmes d'installation du client AudioVideoCast.

Modifier les lignes de /etc/vsftpd/vsftpd.conf :

```
anonymous_enable=NO  
local_enable=YES  
write_enable=YES  
local_umask=022  
chroot_local_user=YES  
local_root=/audiovideocours/ftp
```

Redémarrez le serveur FTP :

```
$ service vsftpd restart
```

## 3. Récupération des sources

Vous aurez besoin d'utiliser git pour récupérer les dernières sources stables du projet via la commande :

```
$ git clone https://github.com/unistra/avc-server.git univ-r_av && cd univ-r_av &&  
git checkout release-2.54 && cd ..
```

L'utilisation de Git ayant échouée (erreur type "Unable to find remote helper for 'https'"), nous avons récupéré l'archive zip sur GitHub à l'adresse <https://github.com/unistra/avc-server/archive/master.zip> :

```
$ wget https://github.com/unistra/avc-server/archive/master.zip  
$ unzip master.zip  
$ mv master avc-server-master
```

Il vous faut également récupérer le player flash JW player, en acceptant la licence « Creative Commons : Attribution-Noncommercial-Share

Alike 3.0 Unported » (voir <http://creativecommons.org/licenses/by-nc-sa/3.0/>) :

```
$ wget http://audiovideocast.unistra.fr/releases/player.swf
$ mv player.swf
/usr/local/share/applications/avc-server-master/WebContent/files/jwflvplayer/
```

## 4. Création de la base de données



Pour déterminer si votre CentOS est en 32-bit ou 64-bit : tapez

```
$ uname -a
```

Si le résultat contient i686 ou i386, votre CentOS est en 32-bit.

Si le résultat contient x86\_64, votre CentOS est en 64-bit.

Installation de PostgreSQL :

```
$ yum install postgresql-server.x86_64
```

Initialisation de la base de données :

```
$ /etc/init.d/postgresql initdb
Initialisation de la base de données :
[ OK ]
```

Démarrage de la base de données :

```
$ service postgresql start
Starting postgresql service:
[ OK ]
```

Mise en place de l'autostart en cas de reboot :

```
$ chkconfig postgresql on
```

Rappel des commandes de base pour postgresql :

```
$ postgres=# help
Vous utilisez psql, l'interface en ligne de commande de PostgreSQL.
Saisissez:
    \copyright pour les termes de distribution
    \h pour l'aide-mémoire des commandes SQL
    \? pour l'aide-mémoire des commandes psql
    \g ou point-virgule en fin d'instruction pour exécuter la requête
    \q pour quitter
```

Création d'un utilisateur "sqluser"

```
$ su - postgres --command="createuser -P sqluser"
Enter password for new role:
Enter it again:
Shall the new role be a superuser? (y/n) n
Shall the new role be allowed to create databases? (y/n) n
Shall the new role be allowed to create more new roles? (y/n) n
```

Création d'une base de données "univrav" en utf-8 pour cet utilisateur :

```
$ su - postgres --command="createdb -O sqluser -E UTF8 univrav"
```

Lancer le script de création de la base se trouvant dans le répertoire des sources :

```
$ psql -h localhost -U sqluser -W -d univrav <
/tmp/avc-server-master/WebContent/scripts/script_creation_database.sql
```



Si une erreur "psql: FATAL: Ident authentication failed for user ..." est retournée, éditez le fichier "pg\_hba.conf" comme suit en gardant une copie de sauvegarde :

```
$ cp /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/data/pg_hba.conf.back
$ vi /var/lib/pgsql/data/pg_hba.conf
```

Avant :

```
# TYPE  DATABASE  USER          CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local   all             all                      ident
# IPv4 local connections:
host    all             all          127.0.0.1/32    ident
# IPv6 local connections:
host    all             all          ::1/128         ident
```

Après :

```
# TYPE  DATABASE  USER          CIDR-ADDRESS  METHOD
# "local" is for Unix domain socket connections only
local   all             all                      password
# IPv4 local connections:
host    all             all          127.0.0.1/32    password
# IPv6 local connections:
host    all             all          ::1/128         password
```

Pour prendre en compte les changements, redémarrez le service :

```
$ sudo service postgresql restart
```

pour plus d'infos : <http://www.postgresql.org/docs/8.2/static/auth-pg-hba-conf.html>

Vérifier que les tables ont bien été créées :

```
// connexion à la base
$ psql -h localhost -d univrav -U sqluser -W
// lister les tables
univrav=> \dt
```

Liste des relations			
Schéma	Nom	Type	Propriétaire
public	amphi	table	sqluser
public	building	table	sqluser
public	course	table	sqluser
public	discipline	table	sqluser
public	job	table	sqluser
public	level	table	sqluser
public	log_user_action	table	sqluser
public	selection	table	sqluser
public	slide	table	sqluser
public	tag	table	sqluser
public	user	table	sqluser

(11 lignes)

## 5. Configuration de tomcat

Ajouter les bibliothèques suivantes dans le répertoire `/usr/share/tomcat6/lib`:

```
$ ln -s /usr/share/java/commons-beanutils-1.9.1.jar
/usr/share/tomcat6/lib/commons-beanutils.jar
$ ln -s /usr/share/java/commons-collections-3.2.1.jar
/usr/share/tomcat6/lib/commons-collections3.jar
$ ln -s /usr/share/java/commons-collections.jar
/usr/share/tomcat6/lib/commons-collections.jar
$ ln -s /usr/share/java/commons-dbc3.jar /usr/share/tomcat6/lib/commons-dbc3.jar
$ ln -s /usr/share/java/commons-fileupload-1.3.1.jar
/usr/share/tomcat6/lib/commons-fileupload.jar
$ ln -s /usr/share/java/commons-io-2.4.jar /usr/share/tomcat6/lib/commons-io.jar
$ ln -s /usr/share/java/commons-lang-2.6.jar
/usr/share/tomcat6/lib/commons-lang.jar
$ ln -s /usr/share/java/commons-logging.jar
/usr/share/tomcat6/lib/commons-logging.jar
$ ln -s /usr/share/java/commons-pool.jar /usr/share/tomcat6/lib/commons-pool.jar
$ ln -s /usr/share/java/postgresql-jdbc3.jar
/usr/share/tomcat6/lib/postgresql-jdbc3.jar
$ ln -s /usr/share/java/log4j-1.2.14.jar /usr/share/tomcat6/lib/log4j1.2.jar
```



Il peut être nécessaire d'installer les bibliothèques Apache Commons à la main au préalable :



```

yum install tomcat6-admin-webapps.noarch
yum install tomcat6-webapps.noarch

wget
http://apache.crihan.fr/dist/commons/fileupload/binaries/commons-fileupload-1.3.1-bin.tar.gz
wget
http://mirrors.ircam.fr/pub/apache/commons/lang/binaries/commons-lang3-3.3.2-bin.tar.gz
wget
http://mirrors.ircam.fr/pub/apache/commons/io/binaries/commons-io-2.4-bin.tar.gz
wget
http://apache.websitebeheerjd.nl/commons/collections/binaries/commons-collections-3.2.1-bin.tar.gz
wget
http://apache.lehtivihrea.org/commons/beanutils/binaries/commons-beanutils-1.9.1-bin.tar.gz
for fichier in $(ls commons*.gz); do tar -xzf $fichier; done
mv commons-*/{lib/,}*jar /usr/share/java/

# vérif :

find / -name "commons-*"

yum install postgresql-jdbc.noarch

# vérif :

find / -name "postgresql-jdbc*"

```

## Configuration du tomcat-users.xml

Création d'un utilisateur tomcat `ulpmm` de la manière suivante :

```

$ cd /usr/share/tomcat6/conf/
$ sudo cp tomcat-users.xml tomcat-users.xml.back
$ sudo vi tomcat-users.xml

```

Contenu :

```

<tomcat-users>
  <role rolename="tomcat"/>
  <role rolename="manager"/>
  <role rolename="ulpmm"/>
  <user username="ulpmm" password="s3cret" roles="tomcat,manager,ulpmm"/>
</tomcat-users>

```

Note: le rôle `ulpmm` est nécessaire pour accéder aux pages d'administration d'AudioVideoCast.

## Configuration du server.xml :

```
$ cd /usr/share/tomcat6/conf/  
// copie de sauvegarde  
$ cp server.xml server.xml.back
```

Parties du fichier à modifier :

```
// configuration du fichier server.xml  
<Connector port="8080" protocol="HTTP/1.1"  
  connectionTimeout="20000"  
  redirectPort="443" />
```

Mise en place du certificat pour Tomcat

```
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"  
maxThreads="150" scheme="https" secure="true"  
clientAuth="false" sslProtocol="TLS"  
keystoreFile="/etc/tomcat6/tomcatkeystore.jks" keystorePass="*****" />
```

```
<!-- Define an AJP 1.3 Connector on port 8009 -->  
<Connector port="8009" protocol="AJP/1.3" redirectPort="443" />
```

```
<Host name="localhost" appBase="webapps" unpackWARs="true" autoDeploy="true"  
  xmlValidation="false" xmlNamespaceAware="false">  
  <Context path="" docBase="univr_av"> <!-- laisser vide le contexte path pour  
accéder au portail avec une adresse du type http://mon-domaine.fr -->  
    <Environment name="volume" value="1" type="java.lang.Short" override="false"/>  
    <Resource  
      name="jdbc/postgres"  
      auth="Container"  
      type="javax.sql.DataSource"  
      driverClassName="org.postgresql.Driver"  
      url="jdbc:postgresql://127.0.0.1:5432/univrav"  
      username="sqluser"  
      password="*****"  
      maxActive="20"  
      maxIdle="10"  
      maxWait="1"  
    />  
    <Resource name="ldap/ox" auth="Container"  
      type="com.sun.jndi.ldap.LdapCtx"  
      factory="org.ulpmm.univrav.dao.LdapFactory"  
      java.naming.factory.initial="com.sun.jndi.ldap.LdapCtxFactory"  
      java.naming.provider.url="your_ldap_url"  
      java.naming.security.authentication="simple"  
      java.naming.security.principal="ldap_user"  
      java.naming.security.credentials="ldap_password"  
      com.sun.jndi.ldap.connect.pool="true"  
      java.naming.security.protocol="ssl"  
      com.sun.jndi.ldap.connect.timeout="5000"  
      com.sun.jndi.ldap.read.timeout="5000" />  
    </Context>  
</Host>
```

## Configuration des politiques :

Attention : certaines versions de tomcat sont plus stricte que d'autres d'un point de vue sécurité au niveau des politiques.

En cas d'erreur provenant de « java.security », il faut :

- Soit désactiver le « security manager » dans `/etc/init.d/tomcat6: TOMCAT6_SECURITY=no`
- Soit configurer les politiques du répertoire `/etc/tomcat6/policy.d/`

Après avoir effectuées toutes les modifications ci-dessus, redémarrer tomcat via :

```
/etc/init.d/tomcat6 restart
```

## 6. Stockage des répertoires de cours

Créer un répertoire `/audiovideocours/cours` :

```
mkdir /audiovideocours/cours
```

Ce répertoire permet de stocker les cours.

Il contient un répertoire (par défaut 1) appelé « volume de stockage ». Il est possible de modifier la valeur du volume dans le contexte de tomcat, si l'on veut monter un nouveau filesystem (pour raison de place par exemple).

Chaque cours se trouve dans une arborescence de sous-répertoires basée sur l'id du cours (formaté sur 8 chiffres).

Par exemple, pour le cours n°4209, l'arborescence sera :

```
/audiovideocours/cours/1/00/00/42/09
```

On donne les droits d'écriture à l'ensemble du répertoire `/audiovideocours` :

```
chmod -R 777 /audiovideocours
```

## 7. Déploiement de l'application AudioVideoCast :



Vérifiez que Ant soit installé sur votre serveur :

```
$ yum install ant
```

Pour créer un fichier `.war` de l'application, il suffit d'utiliser le fichier `build.xml`, qui est à la racine du projet, avec ANT :

- Configurer l'application par le fichier `univrav.properties` dans le répertoire `WebContent/conf` des sources et par le fichier `WebContent/WEBINF/web.xml` (voir Annexe).
- Génération du `.war` de l'application via les sources récupérées du SVN, en exécutant la commande suivante à la racine du projet : `ant`
  - En cas de problème dans `Application.java` > erreur : "package org.apache.commons.lang does not exist "
    1. éditer fichier `Application` : `vi /usr/local/share/applications/avc-server-master/src/org/ulpm/univrav/web/Application.java`
    2. changer classe `commons-lang` chargée : `import org.apache.commons.lang3.text.WordUtils;`
    3. mettre en commentaire ligne 47 : `import org.apache.commons.lang.WordUtils;`
  - En cas d'erreur : "the class org.apache.tools.ant.taskdefs.optional.junit.JUnitTask was not found"
    1. installer junit : `sudo yum install ant-junit.x86_64`
  - Génération du `war` :

```
$ ant
Buildfile: build.xml
prep:
init:
compile:
    [echo] Compilation des sources
test:
makeWar:
    [echo] Création de l'archive univ-r_av.war dans /root/dev/war
    [war] Building war: /root/dev/war/univ-r_av.war
BUILD SUCCESSFUL
Total time: 1 second
```

- Déploiement du .war sur Tomcat (url : <http://localhost:8180/manager/html>)

```
$ cp /root/dev/war/univ-r_av.war /usr/share/tomcat6/webapps/univ-r_av.war
$ service tomcat6 restart
```



Attention : il est préférable de supprimer le cache de Tomcat lors de son redémarrage car il peut y avoir des problèmes de rafraîchissement de pages lorsqu'une nouvelle version est mise en place.

Accéder à l'application via [http://localhost/univ-r\\_av](http://localhost/univ-r_av) ou via l'url apache que vous avez configurée : [http://avcast-test.univ-lyon2.fr:8080/univ-r\\_av/](http://avcast-test.univ-lyon2.fr:8080/univ-r_av/)

## 8. Configuration APACHE

### a. Installation des modules

Les modules qui doivent être installés sont : proxy proxy\_ajp proxy\_balancer proxy\_http rewrite ssl

La plupart sont installés par défaut sur RHEL6. Le seul manquant est le module SSL.

Pour le vérifier :

```
$ grep -E 'rewrite|proxy|ssl' /etc/httpd/conf{.d,}/*.conf | grep LoadModule
/etc/httpd/conf/httpd.conf:LoadModule rewrite_module modules/mod_rewrite.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_module modules/mod_proxy.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_balancer_module
modules/mod_proxy_balancer.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_ftp_module modules/mod_proxy_ftp.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_http_module modules/mod_proxy_http.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_ajp_module modules/mod_proxy_ajp.so
/etc/httpd/conf/httpd.conf:LoadModule proxy_connect_module
modules/mod_proxy_connect.so
```

Le seul module manquant est mod\_ssl.

Pour l'installer :

```
$ yum install mod_ssl.x86_64
```

### b. Configuration d'apache

Il faut créer un fichier `monsite.conf` dans le répertoire `/etc/httpd/conf.d/` qui doit contenir les hôtes virtuels pour le site web (un sur le port 80 et un sur le port SSL 443).

Il faut alors monter le dossier racine afin d'avoir accès à l'arborescence du site, le dossier /audiovideocours/cours/ pour pouvoir accéder aux cours depuis l'extérieur, les dossiers /audiovideocours/ftp/live/ et /audiovideocours/ftp/releases mettant à disposition des fichiers via FTP.

Pour SSL, il faut créer un certificat et l'activer depuis l'hôte virtuel 443. Puis, depuis l'hôte virtuel 80, on redirige les URLs que l'on désire sécuriser vers le port 443, via le mode rewrite.

// création des fichiers mon-site.conf et mon-site.include

```
$ cd /etc/httpd/conf.d/
$ touch avcast-test.conf
$ touch avcast-test.include
```

// Fichier mon-site.conf

```
<VirtualHost *:80>
    ServerName mon-site
    ServerAlias mon-site-alias
    RewriteEngine On
    RewriteCond %{HTTPS} off
    RewriteRule ^(.*)/avc/myspace(.*)$ https://%{HTTP_HOST}%{REQUEST_URI}
[R=301,L]
    #RewriteRule ^(.*)/avc/myspace(.*)$
https://avcast-test.univ-lyon2.fr/avc/myspace_home [R=301,L]
    RewriteRule ^(.*)/admin(.*)$ https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^(.*)/avc/admin(.*)$ https://%{HTTP_HOST}%{REQUEST_URI}
[R=301,L]
    RewriteRule ^(.*)/avc/publication(.*)$
https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    RewriteRule ^(.*)/avc/authentification(.*)$
https://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
    #RewriteRule ^(.*)/avc/authentification(.*)$
https://avcast-test.univ-lyon2.fr/avc/myspace_home [R=301,L]
    Include /etc/httpd/conf.d/mon-site.include
</VirtualHost>

<VirtualHost *:443>
    ServerName mon-site
    ServerAlias mon-site
    SSLEngine On
    SSLCipherSuite (ici les algorithmes de chiffrement générés)
    SSLCertificateFile /etc/pki/tls/certs/mon-site.crt
    SSLCertificateKeyFile /etc/pki/tls/private/mon-site.key
    SSLCertificateChainFile /etc/pki/tls/certs/mon-site.crt
    Include /etc/httpd/conf.d/mon-site.include
    RewriteEngine On
    RewriteCond %{HTTPS} on
    RewriteRule ^(.*)/avc/courseaccess(.*)$ http://%{HTTP_HOST}%{REQUEST_URI}
[R=301,L]
    RewriteRule ^(.*)/avc/liveaccess(.*)$ http://%{HTTP_HOST}%{REQUEST_URI} [R=301,L]
</VirtualHost>
```

// Fichier mon-site.conf

```

ServerAdmin webmaster@localhost

DocumentRoot /var/lib/tomcat6/webapps/univ-r_av
Options -Indexes

<Proxy *>
  Allow from all
</Proxy>

<Proxy balancer://tomcat/univ-r_av>
  BalancerMember ajp://127.0.0.1:8009 route=localhost connectiontimeout=1000
</Proxy>

ProxyPass /coursv2 !
Alias /coursv2 "/audiovideocours/cours/"

ProxyPass /live !
Alias /live "/audiovideocours/ftp/live/"

ProxyPass /releases !
Alias /releases "/audiovideocours/ftp/releases/"

ProxyPass /balancer-manager !
<Location /balancer-manager>
  SetHandler balancer-manager
  Order Deny,Allow
  Deny from all
  Allow from 127.0.0.1
</Location>

ProxyPass / balancer://tomcat/ stickysession=JSESSIONID|jsessionid
ProxyPassReverse / balancer://tomcat/ stickysession=JSESSIONID|jsessionid

ErrorLog /var/log/httpd/error-univ-rav.log
LogLevel warn
CustomLog /var/log/httpd/access-univ-rav.log combined

```

Il faut, après tout changement au niveau de l'un de ces fichiers, redémarrer Apache :

```
$ service httpd restart
```

### c. Connexion avec un sous-répertoire : [http://avcast-test.univ-lyon2.fr/univ-r\\_av/](http://avcast-test.univ-lyon2.fr/univ-r_av/)

Modification du fichier `/usr/share/tomcat6/conf/server.xml` pour faire pointer la racine non plus sur le serveur Tomcat mais sur un sous-répertoire :

```
<Context path="univ-r_av" docBase="univ-r_av"> puis restart Tomcat
```

Le site est donc accessible à l'adresse : [http://avcast-test.univ-lyon2.fr/univ-r\\_av/](http://avcast-test.univ-lyon2.fr/univ-r_av/)



Il convient de vérifier toutes les URLs d'accès, notamment dans le fichier `univrav.properties`.

## 9. Activation du streaming



Partie à venir

## 10. Séparer l'encodage des médias

! Partie à venir

## 11. Suppression des tests

! Partie à venir

## 12. Configuration du live

! Partie à venir

## 13. Page d'administration

La page d'administration permet de gérer différents éléments du site : `http://mon-site/admin`

1. Gestion des cours (éditer, supprimer)
2. Gestion des cours Univ-r (éditer, supprimer)
3. Gestion des bâtiments et des salles (ajouter, éditer, supprimer)
4. Gestion des utilisateurs (ajouter, éditer, supprimer)
5. Statistiques sur les auteurs
6. Statistiques générales (taille du disque, commande findTracks et findStats, version des clients)
7. *Page dédié aux tests (code d'accès spécifiques modifiables dans le fichier « `univrav.properties` » via « `testKeyword` »): `http://mon-site/tests`*

! Cette partie là ne fonctionne pas dans la configuration déployée chez nous.

8. Gestion des sélections et des collections de cours
9. Test des versions des clients des amphithéâtres

NB: Pour les statistiques, il y a la possibilité d'utiliser google analytics (voir fichier « `WEB-INF/views/include/google_analytics.jsp` » pour y coller votre script ga)

## 14. Annexes

### a. Personnalisation de l'interface

Fichier master : `/usr/local/share/applications/avc-server-master/WebContent/WEB-INF/views/include/footer.jsp`

Ligne 14 modifiée :

```
<a href="mailto:nicolas.truchaud@univ-lyon2.fr"><fmt:message key="Assistance"/></a>
- <a href="mailto:nicolas.truchaud@univ-lyon2.fr.fr"><fmt:message
key="Contact"/></a> - <a href="c:out value="\${thick_legal}" />"
title="<fmt:message key="Informations l'acute;gales"/>"
class="thickbox"><fmt:message key="Informations l'acute;gales"/></a>
```

Emplacement final : `/usr/share/tomcat6/webapps/univ-r_av/WEB-INF/views/include/`

### b. Détail du fichier "univrav.properties"

Chemin d'accès au fichier : `/usr/share/tomcat6/webapps/univ-r_av/conf/univrav.properties`

```
# The Url of the server
serverUrl = http://localhost/univ-r_av
## Il s'agit de l'url du serveur. Elle doit être changée.

# The Url to access to the course on internet Prod :
```

```

http://[HOSTNAME].u-strasbg.fr/
coursesUrl = http://avcast-test.univ-lyon2.fr/coursv2/
# Il s'agit de l'url permettant d'accéder aux enregistrements. Elle doit être
changée. Il est possible d'utiliser une fonction RAND[110] pour faire du load
balancing sur plusieurs serveurs. Ex: streamRAND[1-10].example.fr va load balancer
sur stream1,stream2,...,stream10. Il est également possible d'utiliser une fonction
[HOSTNAME] pour récupérer le nom du serveur.

# Folders on the file system Prod : /audiovideocours/cours/ /audiovideocours/ftp/
/audiovideocours/ftp/live/
coursesFolder = /audiovideocours/cours/
# Dossier contenant les cours de l'application.
ftpFolder = /audiovideocours/ftp/
# Dossier contenant les cours zippés envoyés par le client.
liveFolder = /audiovideocours/ftp/live/
# Dossier servant à recevoir la capture d'écran envoyé par le client pendant le
Live.

# Default media filenames in the archive sent by the client
defaultMp3File = enregistrement-micro.mp3
defaultFlashFile = enregistrement-video.flv
defaultMp4File = enregistrement-video.mp4
defaultAudioMacFile1 = enregistrement-micro.flv
defaultAudioMacFile2 = enregistrement-micro.aac
# Ces paramètres ne doivent pas être changés pour le bon fonctionnement avec le
client actuel.

# Copyright comment
comment = Owned by the author
# Commentaire pour les tags des médias.

# IP address of the Flash Server for the video and audio live
flashServerIp = vod-flash-avc.univ-lyon2.fr
# Adresse du serveur flash pour la diffusion en live. Doit être changée.

# The settings of the RSS files, of the permalien (interface flash) and of emails
rssTitle = AudioVideoCast
rssName = AudioVideoCast
rssDescription = Universit? Lyon 2
rssImageUrl =
http://avcast-test.univ-lyon2.fr/univ-r_av/files/img/univr-av-logo-rss.png
rssCategory = Enseignement
language = fr
# Paramètres pour les flux RSS. Peut être changé.
recordedInterfaceUrl = http://avcast-test.univ-lyon2.fr/univ-r_av/avc/courseaccess
# Lien utilisé pour les accès directs aux enregistrements. Doit être changé.

# The setting of the RSS files for iTunes
itunesAuthor = Universit? Lyon 2
itunesSubtitle = Enregistrement culturel
itunesSummary = Retrouvez les supports synchronis?s sur avcast-test.univ-lyon2.fr
itunesImage = http://avcast-test.univ-lyon2.fr/releases/Illustration_itunes_AVC.jpg
itunesCategory = Enseignement
itunesKeywords = science,culture,soci?t?,technologies,?ducation
# Paramètres pour les flux RSS sous iTunes. Peut être changé.

# University parameters
univName = Universite Lyon 2
univAcronym = UL2
univLink = http://www.univ-lyon2.fr/
# Informations concernant l'université sur la page d'accueil

# Publication free

```



```
pubFree = true
#Publication test
pubTest = true
# Autoriser ou non la publication libre de contenu

# The numbers of courses to display at the same time
lastCourseNumber = 10
selectionCourseNumber = 10
collectionCourseNumber = 10
recordedCourseNumber = 10
# Permet de définir le nombre de cours à afficher dans la page d'accueil et dans la
page des enregistrements.

# The default style
defaultStyle = orange-theme
# Thème du site par défaut.

# The keyword to identify the tests to delete (genre is equal to this keyword)
testKeyWord1 = Suppression
# The keyword to identify the tests to hide (title begins with this keyword)
testKeyWord2 = Testul2
# testKeyWord3 = Essai
# Mots clé pour le code d'accès d'un enregistrement permettant de le masquer.

# The client port for the Univ-R integration
#clientSocketPort = 3737

#Admin email for notification (ex: user@domain.fr)
adminEmail1=admin1@fai.fr
adminEmail2=admin2@fai.fr
adminEmail3=admin3@fai.fr
# Adresse email des administrateurs. Doit être changé.

#CAS Logout
casLogoutUrl=https://casl2.univ-lyon2.fr/cas/logout
# Adresse de déconnexion du serveur cas. Doit être changé.

#Additional document formats
addDocFormats=pdf html swf ppt pptx odp docx doc odt xls xlsx ods rtf txt jpg jpeg
png gif bmp zip bz bz2 ark rar 7z ac3 avi divx flv m3u m4a mov movie mp2 mp3 mp4
mpg mpeg ogg ra rm rv wav wma wmv aac
# Formats de fichiers autorisés pour l'upload de documents additionnels.

#Upload media formats
uploadFormats=mp3 ogg wav wma avi divx mp4 mpg mpeg mov wmv mkv flv ogv webm m4v
# Formats de fichiers autorisés pour l'upload de fichiers audio et video.

#Link for support (help page)
supportLink=http://dsi.univ-lyon2.fr
helpLink=https://sites.google.com/site/wikiaudiovideocast/
clientLink=https://sites.google.com/site/wikiaudiovideocast/client-logiciel-fr
tracLink=http://sourcesup.cru.fr/projects/audiovideocours
docLink=http://dsi.univ-lyon2.fr/acces-et-utilisation-des-outils-486843.kjsp
# Lien d'aide pour l'application (page help). Peut être changé.

#LDAP search properties
ldapBaseDn=ou=people,dc=univ-lyon2,dc=fr
ldapSearchFilter=uid
ldapMail=mail
ldapFirstname=givenName
ldapLastname=sn
ldapProfile=eduPersonPrimaryAffiliation
ldapAffectation=supannetablissement
```

```
ldapEtpPrimaryCode=udsPrimaryEtpCode
ldapPrimaryInstitute=udsMainDepartmentCode
ldapSecondaryInstitute=supannEntiteAffectationPrincipale
# Propriétés de recherche du ldap. À modifier en fonction du ldap.

# To separate medias encodage
sepEnc=false
# Si false, l'application web encode les médias (par défaut). Si true, l'encodage
des medias doit se faire séparément (via tâche cron qui appelle les scripts).

# default record interface (flash or html5)
defaultRecordInterface=html5
# html5 ou flash pour l'interface de visualisation par défaut

# Log user action stats
logstats=true
# logger les actions des utilisateurs pour les statistiques

# Google analytics
googleAnalyticsAccount=

# Show Contact Us
```

```
contactUs=true
```

### c. Détail du filtre CAS du fichier "web.xml"

Chemin d'accès au fichier : /usr/share/tomcat6/webapps/univ-r\_av/WEB-INF/web.xml

```
<filter>
  <filter-name>CAS filter</filter-name>
  <filter-class>edu.yale.its.tp.cas.client.filter.CASFilter</filter-class>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.loginUrl</param-name>
    <param-value>https://cas.example.fr:443/cas/login</param-value>
    Url de login du serveur CAS. À changer.
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.validateUrl</param-name>
    <param-value>https://cas.example.fr:443/cas/serviceValidate</param-value>
    Url de validation du serveur CAS. À changer.
  </init-param>
  <init-param>
    <param-name>edu.yale.its.tp.cas.client.filter.serverName</param-name>
    <param-value>localhost</param-value>
    Url de retour après connexion au serveur CAS. Il s'agit de l'adresse de votre
    site. À changer.
  </init-param>
</filter>
```