

Configuration serveur pour WebRSA 4

| | |
|------------------------------|--|
| Client | CD |
| Projet | WebRSA |
| Auteur | E. DUCHESNE / E. PRITCHARD / C. POYER / P. LAVIGNE / P. BRUN (DevOps) Atol C&D |
| Objet de l'installation | Configuration serveur |
| Date de l'intervention | |
| Version(s) installée(s) | Version 4 |
| Date de révision du document | 11 juin 2020 |
| ID machine | |
| Contenu | Voir détail ci-après |

Table des matières

| | | |
|----------|---|-----------|
| 1 | Spécifications serveur | 4 |
| 1.1 | Configuration de base | 4 |
| 1.2 | Schéma de partitionnement conseillé | 4 |
| 2 | Prérequis | 5 |
| 2.1 | Utilisateur et dossier | 5 |
| 2.2 | Dépôts supplémentaires | 5 |
| 2.3 | Paquets | 6 |
| 2.4 | Désactiver SELinux | 6 |
| 2.5 | Désactiver firewalld | 6 |
| 3 | Apache | 7 |
| 3.1 | Installation | 7 |
| 3.2 | Configuration | 7 |
| 4 | Php | 9 |
| 4.1 | Installation | 9 |
| 4.2 | Configuration | 9 |
| 4.2.1 | Php | 9 |
| 4.2.2 | Apache | 10 |
| 5 | Postgresql | 12 |
| 5.1 | Installation | 12 |
| 5.2 | Création d'une base et d'un utilisateur | 12 |
| 5.3 | Extension | 13 |
| 5.4 | Configuration | 13 |
| 5.5 | Exception | 14 |
| 5.5.1 | Client SQL | 14 |
| 5.5.2 | pg_dump | 14 |
| 6 | Tomcat-Python-Clouddoo, Golem, Openoffice-Headless | 16 |
| 6.1 | Paquets complémentaires | 16 |
| 6.2 | Openoffice-Headless | 16 |
| 6.3 | Tomcat-Clouddoo | 16 |
| 6.3 | PDFTK | 22 |
| 6.4 | Pear XML | 22 |
| 6.5 | Goleem | 23 |
| 7 | Logrotate | 25 |
| 8 | Composer | 26 |

1 Spécifications serveur

1.1 Configuration de base

- * vhost nommé webrsa-web1
- * apache 2.4
- * php fpm 7.2
- * postgresql 10
- * SE linux et firewalld sont désactivés sur la machine.

1.2 Schéma de partitionnement conseillé

- * / : 10 Go
- * /var/log : 2Go
- * /var/www : 5 Go
- * /var/lib/postgres : 25Go

2 Prérequis

2.1 Utilisateur et dossier

Créer l'utilisateur qui fera fonctionner l'application WebRSA :

```
mkdir -p /var/www/webrsa-web1  
useradd -d /var/www/webrsa-web1/ -s /bin/false webrsa-web1
```

Ne pas tenir compte du message suivant :

```
useradd : attention, le répertoire personnel existe déjà.  
Aucun fichier du répertoire « skels » n'y sera copié.
```

Créer les dossiers qui contiendront le vhost et les logs :

```
mkdir -p /var/www/webrsa-web1/{public_html,logs}  
chown -R webrsa-web1:webrsa-web1 /var/www/webrsa-web1/
```

2.2 Dépôts supplémentaires

Afin d'installer les logiciels dans les bonnes versions, il est nécessaire d'ajouter des dépôts complémentaires.

Ajouter le dépôt epel :

```
yum install epel-release
```

Ajouter le dépôt ius :

Centos :

```
yum install https://centos7.iuscommunity.org/ius-release.rpm
```

Ajouter le dépôt postgresql :

Centos :

```
yum install  
https://download.postgresql.org/pub/repos/yum/10/redhat/rhel-7-x86_64/pgdg-centos10-10-2.noarch  
.rpm
```

2.3 Paquets

Installer les dépendances suivantes :

```
yum install java-1.8.0-openjdk
```

Paquets complémentaires d'administration (optionnel) :

```
yum install wget vim bash-completion htop man man-pages man-pages-fr bind-utils iotop iftop rsync  
mailx screen tcpdump telnet lsof git xz zip unzip lynx iperf nmap ntp ntpdate nc jnettop yum-utils  
deltarpm mtr nano
```

2.4 Désactiver SELinux

Editer `/etc/selinux/config` :

```
SELINUX=disabled
```

Redémarrer :

```
reboot
```

2.5 Désactiver firewalld

Arrêter le service :

```
systemctl stop firewalld
```

Le désactiver au démarrage :

```
systemctl disable firewalld
```

Redémarrer :

```
reboot
```

3 Apache

3.1 Installation

Installer apache :

```
yum install httpd
```

Ajouter apache dans le groupe de l'utilisateur webrsa-web1 :

```
usermod -aG webrsa-web1 apache
```

3.2 Configuration

Créer le fichier `/etc/httpd/conf.d/10-webrsa-web1.conf` pour configurer le vhost apache :

```
<VirtualHost *:80>
  ServerName webrsa-web1.local

  ## Vhost docroot
  DocumentRoot "/var/www/webrsa-web1/public_html"

  ## Directories
  <Directory "/var/www/webrsa-web1">
    AllowOverride All
    Require all granted
  </Directory>

  ## Php-fpm
  <FilesMatch "\.php$">
    Require all granted
    SetHandler proxy:fcgi://127.0.0.1:9467
  </FilesMatch>

  ## Logging
  ErrorLog "/var/www/webrsa-web1/logs/error.log"
  ServerSignature Off
  CustomLog "/var/www/webrsa-web1/logs/access.log" "combined" env=!forwarded
  CustomLog "/var/www/webrsa-web1/logs/access.log" "forwarded" env=forwarded
  SetEnvIf X-Forwarded-For "^.*\.*\.*\.*" forwarded

</VirtualHost>
```

Relancer apache :

```
systemctl restart httpd
```

Activer le service au démarrage :

```
systemctl enable httpd
```


4 Php

4.1 Installation

Installer les paquets php suivants :

```
yum install php72u-fpm-httpd php72u-devel php72u-mbstring php72u-soap php72u-pgsql php72u-pdo  
php72u-bcmath php72u-gd php72u-pear php72u-memcached php72u-pspell php72u-snmp php72u-xml  
php72u-xmlrpc
```

4.2 Configuration

4.2.1 Php

S'assurer que les lignes suivantes sont présentes et décommentées dans `/etc/php-fpm.conf`

```
include=/etc/php-fpm.d/*.conf  
[global]  
pid = /var/run/php-fpm/php-fpm.pid  
error_log = /var/log/php-fpm/error.log  
syslog.facility = daemon  
syslog.ident = php-fpm  
log_level = notice  
emergency_restart_threshold = 0  
emergency_restart_interval = 0  
process_control_timeout = 0  
daemonize = yes
```

Créer le fichier de configuration spécifique au site webrsa-web1 `/etc/php-fpm.d/webrsa-web1.conf`

```
[webrsa-web1]  
listen = 127.0.0.1:9467  
listen.backlog = -1  
listen.allowed_clients = 127.0.0.1  
user = webrsa-web1  
group = webrsa-web1  
pm = dynamic  
pm.max_children = 50  
pm.start_servers = 5  
pm.min_spare_servers = 5  
pm.max_spare_servers = 35  
pm.max_requests = 0  
ping.response = pong  
request_slowlog_timeout = 1  
slowlog = /var/www/webrsa-web1/logs/php-slow.log  
catch_workers_output = no
```

```
php_admin_value[error_log] = /var/www/webrsa-web1/logs/php-error.log
php_admin_flag[log_errors] = on
php_admin_value[date.timezone] = Europe/Paris
php_admin_value[max_input_vars] = 4000
php_admin_value[session_save_path] = /var/lib/php
php_admin_value[upload_max_filesize] = 10M
```

Relancer le service :

```
systemctl start php-fpm.service
```

Activer le service au démarrage :

```
systemctl enable php-fpm.service
```

4.2.2 Apache

Dans le fichier de configuration d'apache `/etc/httpd/conf.d/10-webrsa-web1.conf`, vous devez ajouter les lignes en gras sous les lignes en italique.

Dans les deux blocs les numéros de port doivent correspondre :

```
<FilesMatch "\.php$">
    Require all granted
    SetHandler proxy:fcgi://127.0.0.1:9815
</FilesMatch>

<Proxy fcgi://127.0.0.1:9815>
    ProxySet timeout=300
</Proxy>
```

Dans le fichier de configuration d'apache `/etc/httpd/conf/httpd.conf`, vous devez modifier la ligne du timeout de 60 à 300 comme ceci :

```
Timeout 300
```

Relancer le service :

```
systemctl start php-fpm.service
```

Relancer apache :

```
systemctl restart httpd
```

5 Postgresql

5.1 Installation

Installer postgresql server :

```
yum install postgresql10-server postgresql10
```

Initier la base de données :

```
/usr/pgsql-10/bin/postgresql-10-setup initdb
```

Démarrer postgresql :

```
systemctl start postgresql-10.service
```

Activer le service au démarrage :

```
systemctl enable postgresql-10.service
```

5.2 Création d'une base et d'un utilisateur

Se connecter au prompt postgres pour pouvoir utiliser les commandes sql :

```
su - postgres  
psql
```

Créer un utilisateur afin d'accéder à la base :

```
CREATE ROLE webrsa WITH LOGIN PASSWORD 'passWORD123*!';
```

Créer la base de données webrsa-web1 :

```
CREATE DATABASE webrsa_web1 OWNER webrsa;
```

5.3 Extension

Installation de l'extension :

```
yum install postgresql10-contrib
```

Connexion à la base de données depuis le compte postgres :

```
sudo su postgres  
psql -d webrsa_web1
```

Création de l'extension :

```
UPDATE pg_language SET lanpltrusted = true WHERE lanname LIKE 'c';  
CREATE EXTENSION fuzzysrmatch;
```

5.4 Configuration

Ajouter la ligne en gras dans le fichier `/var/lib/pgsql/10/data/pg_hba.conf` :

```
# TYPE DATABASE USER ADDRESS METHOD  
  
# "local" is for Unix domain socket connections only  
local all all peer  
# IPv4 local connections:  
host all webrsa localhost md5  
host all all 127.0.0.1/32 ident  
# IPv6 local connections:  
host all all ::1/128 ident  
# Allow replication connections from localhost, by a user with the  
# replication privilege.  
local replication all peer  
host replication all 127.0.0.1/32 ident  
host replication all ::1/128 ident
```

Redémarrer postgresql :

```
systemctl restart postgresql-10.service
```

5.5 Exception

Si des erreurs de connexion se produisent, vous pouvez tester les configurations suivantes.

5.5.1 Client SQL

Pour un département la configuration a dû être modifiée comme ci-dessous pour que celui-ci puisse utiliser un client SQL.

`/var/lib/pgsql/10/data/pg_hba.conf :`

| # | TYPE | DATABASE | USER | ADDRESS | METHOD |
|--|-------------|----------|--------|--------------|--------|
| # "local" is for Unix domain socket connections only | | | | | |
| local | all | | all | | peer |
| # IPv4 local connections: | | | | | |
| host | all | | webrsa | 0.0.0.0/0 | md5 |
| host | all | | all | 127.0.0.1/32 | ident |
| # IPv6 local connections: | | | | | |
| host | all | | all | ::1/128 | md5 |
| # Allow replication connections from localhost, by a user with the | | | | | |
| # replication privilege. | | | | | |
| local | replication | | all | | peer |
| host | replication | | all | 127.0.0.1/32 | ident |
| host | replication | | all | ::1/128 | ident |

`/var/lib/pgsql/10/data/postgresql.conf :`

| | |
|----------------------------------|--|
| #----- | |
| # CONNECTIONS AND AUTHENTICATION | |
| #----- | |
| # - Connection Settings - | |
| listen_addresses = '*' | # what IP address(es) to listen on; |
| | # comma-separated list of addresses; |
| | # defaults to 'localhost'; use '*' for all |
| | # (change requires restart) |

5.5.2 pg_dump

Pour utiliser la commande `pg_dump` depuis un terminal.

`/var/lib/pgsql/10/data/pg_hba.conf :`

| # | TYPE | DATABASE | USER | ADDRESS | METHOD |
|---|------|----------|------|---------|--------|
|---|------|----------|------|---------|--------|

```
# "local" is for Unix domain socket connections only
local all all trust
# IPv4 local connections:
host all webrsa 0.0.0.0/0 md5
host all all 127.0.0.1/32 ident
# IPv6 local connections:
host all all ::1/128 md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local replication all peer
host replication all 127.0.0.1/32 ident
host replication all ::1/128 ident
```

6 Tomcat-Python-Cloudoo, Golem, Openoffice-Headless

6.1 Paquets complémentaires

Paquets complémentaires nécessaires au fonctionnement :

```
yum install gcc libreoffice-writer libreoffice-draw libreoffice-impress libreoffice-calc python-devel  
python-setuptools python-lxml python-psutil python-paste-script wget xorg-x11-font-utils
```

6.2 Openoffice-Headless

Installation d'Open Office en headless pour le fonctionnement des créations de documents texte, PDF et des impressions :

```
cd /tmp  
wget http://ressources.libriciel.fr/deploiement/l/LibreOffice_4.2.6_Linux_x86-64_rpm.tar.gz  
tar -zxf LibreOffice_4.2.6_Linux_x86-64_rpm.tar.gz  
cd LibreOffice_4.2.6.2_Linux_x86-64_rpm/RPMS/  
rm *gnome*  
rpm -Uvh *.rpm
```

Répondre oui à la demande de confirmation de la commande :

```
rm : supprimer fichier « libobasis4.2-gnome-integration-4.2.6.2-2.x86_64.rpm » ?
```

6.3 Tomcat-Cloudooo

Télécharger Tomcat :

```
cd /tmp  
wget http://mirror.linux-ia64.org/apache/tomcat/tomcat-8/v8.5.42/bin/apache-tomcat-8.5.42.tar.gz
```

Nous utilisons la version 8.5.35, si cette version n'est plus disponible, vous devez chercher la dernière version de Tomcat 8 ou Tomcat 9 disponible à l'url <http://mirror.linux-ia64.org/apache/tomcat> (ou vous pouvez également utiliser le .tar.gz fourni en annexe par Atol).

Décompresser les fichier téléchargés :


```
tar -xvf apache-tomcat-8.5.42.tar.gz  
rm apache-tomcat*.tar.gz  
mv apache-tomcat* /opt/tomcat
```

Répondre oui à la demande de confirmation de la commande :

```
rm : supprimer fichier « apache-tomcat-8.5.35.tar.gz » ?
```

Ajouter les permissions d'exécution :

```
chmod +x /opt/tomcat/bin/*sh
```

Créer le dossier :

```
mkdir /opt/tomcat/scripts/
```

Créer le fichier `/opt/tomcat/scripts/create_admin_user.sh` :

```
#!/bin/bash  
  
if [ -f /.tomcat_admin_created ]; then  
    echo "Tomcat 'admin' user already created"  
    exit 0  
fi  
  
#generate password  
PASS=${TOMCAT_PASS:-$(cat /dev/urandom | tr -dc 'a-zA-Z0-9' | fold -w 10 | head -n 1)}  
_word=$( [ ${TOMCAT_PASS} ] && echo "preset" || echo "random" )  
  
echo "=> Creating and admin user with a ${_word} password in Tomcat"  
sed -i -r 's/<\/tomcat-users>/' "${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<role rolename="manager-gui"/>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<role rolename="manager-script"/>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<role rolename="manager-jmx"/>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<role rolename="admin-gui"/>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<role rolename="admin-script"/>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo "<user username=\"admin\" password=\"${PASS}\"  
roles=\"manager-gui,manager-script,manager-jmx,admin-gui, admin-script\"/>" >>  
"${CATALINA_HOME}/conf/tomcat-users.xml  
echo '<\/tomcat-users>' >> "${CATALINA_HOME}/conf/tomcat-users.xml  
echo "=> Done!"  
touch "${CATALINA_HOME}/scripts/.tomcat_admin_created  
  
echo "=====  
echo "You can now configure to this Tomcat server using:"  
echo ""
```

```
echo "  admin:${PASS}"  
echo ""  
echo "=====
```

Créer un utilisateur pour Tomcat :

```
groupadd -r tomcat  
  
useradd -g tomcat -d /opt/tomcat -s /sbin/nologin -c "Tomcat user" tomcat
```

Ne pas tenir compte du message suivant :

```
useradd : attention, le répertoire personnel existe déjà.  
Aucun fichier du répertoire « skels » n'y sera copié.
```

Changer les droits d'accès :

```
chown -R tomcat:tomcat /opt/tomcat
```

Lancer le script de création d'admin :

```
cd /opt/tomcat
```

```
chmod +x scripts/create_admin_user.sh
```

Cette commande génère un mot de passe pour le serveur Tomcat, vous devez le conserver en cas de besoin :

```
./scripts/create_admin_user.sh
```

```
exec /opt/tomcat/bin/catalina.sh run &
```

Récupérer Cloudooo :

```
cd /tmp  
  
wget https://ressources.libriciel.fr/deploiement/c/cloudooo-v1.25-xenial.tgz  
  
tar xvzf cloudooo-v1.25-xenial.tgz
```

Installer les dépendances supplémentaires :

```
yum install libmspack
```

```
cd /tmp

wget dl.fedoraproject.org/pub/epel/7/x86_64/Packages/c/cabextract-1.5-1.el7.x86_64.rpm

rpm -ivh cabextract-1.5-1.el7.x86_64.rpm

rpm -i
https://downloads.sourceforge.net/project/mscorefonts2/rpms/msttcore-fonts-installer-2.6-1.noarch.
rpm
```

mscorefonts2 peut être télécharger ici si le lien ci-dessus ne fonctionne pas :

<https://sourceforge.net/projects/mscorefonts2/>

```
cd /tmp/cloudooo-v1.25/cloudooo-v1.25 && python setup.py install
```

```
mkdir -p /opt/cloudooo/conf && mkdir -p /opt/cloudooo/pid && mkdir -p /var/log/cloudooo && ln -s
/var/log/cloudooo /opt/cloudooo/log
```

Créer le fichier **/opt/cloudooo/conf/cloudooo.conf** :

```
[app:main]
use = egg:cloudooo
#
## System config
#
debug_mode = True
# Folder where pid files, lock files and virtual frame buffer mappings
# are stored. In this folder is necessary create a folder tmp, because this
# folder is used to create all temporary documents.
working_path = /tmp/cloudooo
# Folder where UNO library is installed
# uno_path = /usr/lib64/libreoffice/ure-link/bin
uno_path = /usr/lib64/libreoffice/program
# Folder where soffice.bin is installed
office_binary_path = /usr/lib64/libreoffice/program
#
## Monitor Settings
#
# Limit to use the Openoffice Instance. if pass of the limit, the instance is
# stopped and another is started.
limit_number_request = 100
# Interval to check the factory
monitor_interval = 10
timeout_response = 1800
```

```
enable_memory_monitor = True
# Set the limit in MB
# e.g 1000 = 1 GB, 100 = 100 MB
limit_memory_used = 3000
#
## OoFactory Settings
#
# The pool consist of several OpenOffice.org instances
application_hostname = localhost
# OpenOffice Port
openoffice_port = 4062
openoffice_user_interface_language = fr
#
## Environment Variable Settings (env-KEY = value)
#
# specify preferable library locations
# env-LD_LIBRARY_PATH =
/opt/erp5/trunk/parts/fontconfig/lib:/opt/erp5/trunk/parts/libpng/lib:/opt/erp5/trunk/parts/zlib/lib
#
# specify preferable executable locations
# env-PATH = /opt/erp5/trunk/parts/imagemagick/bin:/opt/erp5/trunk/parts/w3m/bin
#
# Mimetype Registry
# It is used to select the handler that will be used in conversion.
# Priority matters, first match take precedence on next lines.
mimetype_registry =
  application/vnd.openxmlformats-officedocument.wordprocessingml.document * ooo
  application/doc * ooo
  application/pdf * ooo
  application/vnd.oasis.opendocument* * ooo
  application/vnd.sun.xml* * ooo
  text/* * ooo
  image/* image/* imagemagick
  video/* * ffmpeg
  * application/vnd.oasis.opendocument* ooo
  * * ooo
# This is used to disable ooo filters
# Below is an example list of spreadsheet filter names.
# http://wiki.openoffice.org/wiki/Documentation/DevGuide/Spreadsheets/Filter\_Options
#ooo_disable_filter_name_list =
# Microsoft Excel 5.0
# Microsoft Excel 95
# Microsoft Word 6.0
# Microsoft Word 95

[server:main]
```

```
use = egg:PasteScript#wsgiutils  
host = 0.0.0.0  
port = 8011
```

Lancer le serveur cloudooo grâce à cette commande :

```
/usr/bin/paster serve /opt/cloudooo/conf/cloudooo.conf --daemon --pid-file  
/opt/cloudooo/pid/cloudooo.pid --log-file=/opt/cloudooo/log/cloudooo.log
```

Tester que la commande est bien lancée en consultant les logs :

```
tail -f /var/log/cloudooo/cloudooo.log
```

Vous obtenez un message du type :

```
2019-02-07 13:54:07 - Cloudooo - DEBUG - Test OpenOffice 4062 - Pid 3487  
2019-02-07 13:54:07 - Cloudooo - DEBUG - Testing Openoffice Instance 4062  
2019-02-07 13:54:08 - Cloudooo - DEBUG - Instance 4062 works  
2019-02-07 13:54:08 - Cloudooo - DEBUG - Process Started openoffice, Port 4062. Pid 3487  
2019-02-07 13:54:08 - Cloudooo - DEBUG - Start MonitorRequest  
2019-02-07 13:54:08 - Cloudooo - DEBUG - Start MonitorMemory  
2019-02-07 13:54:08 - Cloudooo - DEBUG - OpenOffice localhost, 4062 unlocked  
Starting server in PID 3478.  
Starting HTTP server on http://0.0.0.0:8011
```

Installer net-tools :

```
yum provides */netstat  
yum install net-tools
```

Vérifier que le port 8011 est en écoute :

```
netstat -ntapu |grep 8011
```

Cette commande doit retourner la ligne suivante :

```
tcp 0 0 0.0.0.0:8011 0.0.0.0:* LISTEN 3751/python
```

Vérifier le port 4062 est en écoute localement :

```
netstat -ntapu |grep 4062
```

Cette commande doit retourner la ligne suivante :

```
tcp 0 0 127.0.0.1:4062 0.0.0.0:* LISTEN 3781/soffice.bin
```

6.3 PDKT

Lancer la commande d'installation de PDKT :

```
yum localinstall https://www.linuxglobal.com/static/blog/pdftk-2.02-1.el7.x86_64.rpm
```

Si la ligne ci-dessus ne fonctionne pas il est possible d'utiliser le fichier pdftk-2.02-1.el7.x86_64.rpm fourni en pièce jointe par ATOL CD en le déposant dans le répertoire /tmp puis en lançant la commande :

```
yum localinstall /tmp/pdftk-2.02-1.el7.x86_64.rpm
```

6.4 Pear XML

Créer le dossier temporaire d'installation :

```
mkdir -p /tmp/pear/install
```

Récupération de GO-PEAR :

```
cd /tmp  
wget https://objects-us-east-1.dream.io/kbfiles/pear/go-pear.phar --no-check-certificate
```

Si la ligne ci-dessus ne fonctionne pas il est possible d'utiliser le fichier go-pear.phar fourni en pièce jointe par ATOL CD.

Exécuter la commande :

```
php go-pear.phar
```

Appuyer sur la touche entrée pour lancer l'installation.

Déposer le fichier PHP-PEAR-RPC-XML.zip fourni par Atol CD sur votre serveur dans répertoire /tmp.

Décompresser et copier ces fichiers vers leur répertoire d'utilisation :

```
cd /tmp
```

```
unzip PHP-PEAR-RPC-XML.zip  
cp PHP-PEAR-RPC-XML/RPC.php /usr/share/pear/XML/  
mkdir /usr/share/pear/XML/RPC  
cp PHP-PEAR-RPC-XML/RPC/* /usr/share/pear/XML/RPC
```

6.5 Goolem

Déposer le fichier source de ODFgedooo.war fournis par ATOLCD dans le répertoire /tmp de votre serveur.

Copier le fichier ODFgedooo.war dans le répertoire de fonctionnement de Tomcat :

```
cp /tmp/ODFgedooo.war /opt/tomcat/webapps/
```

Redémarrer :

```
reboot
```

Exécuter les commandes du serveur tomcat :

```
exec /opt/tomcat/bin/catalina.sh run&  
  
/usr/bin/paster serve /opt/cloudooo/conf/cloudooo.conf --daemon --pid-file /opt/cloudooo/pid/cloudooo.pid  
--log-file=/opt/cloudooo/log/cloudooo.log && exec /opt/tomcat/bin/catalina.sh run&
```

Vérifier que le port 8080 est en écoute :

```
netstat -ntapu |grep 8080
```

Cette commande doit retourner la ligne suivante :

| | | | | | | |
|------|---|---|-------------------|-------------------|-----------|------------|
| tcp6 | 0 | 0 | :::8080 | :::* | LISTEN | 23637/java |
| tcp6 | 0 | 0 | 172.30.3.170:8080 | 172.31.4.34:55402 | TIME_WAIT | - |

Tester le bon fonctionnement de Goolem en utilisant cette url dans un navigateur :

```
[IPSERVEUR]:8080/ODFgedooo/OfficeService?wsdl
```


7 Logrotate

Afin de faire tourner les logs d'apache, créer le fichier suivant `/etc/logrotate.d/httpd-lamp`

```
/var/www/*/logs/*.log {  
  compress  
  create 640 root adm  
  daily  
  delaycompress  
  missingok  
  rotate 5  
  sharedscripts  
  postrotate  
    /etc/init.d/httpd reload &> /dev/null || true  
  endscript  
}
```

8 Composer

Afin d'installer composer, suivre les instructions suivantes :

```
sudo yum -y update  
cd /tmp  
sudo curl -sS https://getcomposer.org/installer | php  
mv composer.phar /usr/local/bin/composer
```