



GESTION INTEGREE DE LA TERRE ET DE L'EAU POUR L'ADAPTATION A LA VARIABILITE ET AU CHANGEMENT CLIMATIQUES au MALI

Guide Technique DU SI-GDT

Document d'installation et de configuration des applications



PREAMBULE

Contexte

La présente documentation se rapporte à une installation des applications sur un serveur exploitant une distribution Linux Debian « Wheezy » 7.2, version Serveur 64 bits.

Dans le cas de systèmes de la famille Debian (autres versions de Debian ou distributions apparentées, comme Ubuntu), cela ne devrait guère différer. Cependant, il est possible que certains ajustements soient nécessaires. Les logiciels évoluent, et les paramètres de configurations aussi. Il faut garder l'esprit ouvert, et ne pas hésiter à solliciter les moteurs de recherche en cas de problème. L'aide en ligne est également précieuse !

Pré-requis

Les logiciels serveurs Apache 2 et Tomcat 7 sont supposés être déjà installés et configurés pour la production. Nous suggérerons peut-être dans la suite quelques modifications à leur apporter, mais pour l'essentiel de leur installation et configuration, se reporter aux documentations officielles Debian (<http://www.debian.org/doc/manuals/debian-reference/>), Apache 2 (<http://httpd.apache.org/>) et Tomcat 7 (<http://tomcat.apache.org/tomcat-6.0-doc/index.html>).

De même, le SGBD PostGreSQL sera installé, configuré et sécurisé.

Spécificités

Partitionnement

Nous supposons que le système est fractionné en deux partitions (plus la partition swap, bien entendu) : partition / de 40Go en ext4, suffisant pour héberger le système, partition /home occupant le reste de l'espace disponible.

Les données seront autant que possible stockées dans la partition /home, alors que les applications seront installées dans /

Une taille inférieure pour / est envisageable, mais nécessite quelques optimisations. Lorsque cela s'avérera pratique, nous proposerons de telles optimisations.

Répertoire racine des données

On va choisir de regrouper toutes les données dédiées à l'application dans un répertoire /home/ilwac.

Utilisateur

Dans tout ce qui suit, on considérera que l'administrateur a créé un compte d'utilisateur nommé 'aedd' lors de l'installation, et opérera depuis ce compte. Dans le cas où le nom différerait, il convient d'ajuster toutes les références à 'aedd'...

Vue d'ensemble

Le géoportail est une version retravaillée du logiciel open source GeoNetwork.

Comme GeoNetwork, il s'agit d'une application devant être hébergée par un serveur (internet) de servlets Java, en l'occurrence Tomcat.

Par ailleurs, ce logiciel stocke les données qu'il exploite dans une base de donnée. En l'occurrence, il s'agira d'une base de données PostGreSQL.

Les données cartographiques, que l'on pourra visualiser dans les espaces cartographiques du géocatalogue

seront traitées par un serveur cartographique, GeoServer. Il est lui aussi hébergé par Tomcat.

En plus des fonctions de catalogage (outil de gestion documentaire) apportées par GeoNetwork, le géoportail ILWAC offre également des fonctionnalités cartographiques, permettant de naviguer au sein d'un ensemble de données cartographiques. L'essentiel d'entre elles sont stockées et desservies par l'instance geoserver de notre serveur. D'autres proviennent de sources externes. Nous nous sommes efforcés de faire les choix les plus pertinents, pour chaque donnée.

Cet aspect « client cartographique » est l'héritier du SIGDT 2D de la précédente version. Il exploite essentiellement des bibliothèques javascript, et quelques scripts PHP. Pour une optimisation des ressources, il est hébergé en partie avec le géocatalogue et en partie sous Apache. La base de données associée est elle aussi hébergée dans une base PostgreSQL.

Une partie des données cartographiques exploitées par le Système d'Information sont stockées dans une base PostgreSQL/PostGIS. L'outil qui transforme les données PostGIS en images exploitables par le SIGDT est geoserver, notre serveur cartographique.

CREATION DE /HOME/ILWAC

Lorsque l'on a partitionné le disque, on a créé deux partitions, une pour /, l'autre pour /home. C'est cette dernière que l'on va exploiter pour stocker les données. C'est d'ailleurs à elle que l'on a attribué l'essentiel de l'espace disque. Dans /home, on va créer un répertoire 'ilwac', racine de toutes les données de l'application.

En mode console, on crée le répertoire :

```
sudo mkdir /home/ilwac
```

On attribue, pour simplifier les manipulations ultérieures, la propriété du répertoire et de ses sous-répertoires à aedd :

```
sudo chown -R aedd:aedd /home/ilwac
```

On procédera de même pour créer un répertoire /home/gdtweb, qui hébergera les fichiers du site web (repris tel quel de la version précédente)

SERVEUR WEB APACHE

Une partie des fichiers de l'application seront servis directement par le serveur web Apache2. Le reste transitera également par Apache2 (ce qu'on réglera plus loin, voir AJP)

Apache est en principe déjà installé sur le serveur. Nous allons juste adapter quelques éléments de sa configuration à nos besoins.

Une partie des fichiers des applications web sont à placer directement dans le répertoire www d'Apache. Cela étant, si l'on veut découpler le système des données (faciliter les mises à jour, réinstallation et sauvegardes), on va chercher à placer les fichiers plutôt dans un disque dur différent de celui utilisé par le système. En l'occurrence, on utilisera /home/ilwac

« Hors-sujet » : (Ré-)Installation du site web gdtmali.org

Copie des fichiers

On extrait les fichiers en préservant les droits d'utilisateurs d'origine. Depuis /home/gdtweb/www, on fait `tar zxvfp [support installation]/sources/www.tar.gz`

On vérifie par un `ls -la` que les droits sont bons. On devrait avoir un utilisateur geocat, et groupe www-data.

On va fixer le groupe pour tout le répertoire de la façon suivante :

```
sudo chown -R aedd:www-data /home/gdtweb/www
sudo chmod g+s -R /home/gdtweb/www
```

Changement du répertoire de base d'Apache

Le répertoire par défaut d'Apache est /var/www. On va le changer, tout simplement.

On édite le fichier /etc/apache2/sites-available/default (en mode super-utilisateur – `sudo gedit /etc/apache2/sites-available/default`), et on change /var/www par /home/gdtweb/www

```
# DocumentRoot /var/www
DocumentRoot /home/gdtweb/www
<Directory />
  Options FollowSymLinks
  AllowOverride All
</Directory>
# <Directory /var/www/>
<Directory /home/gdtweb/www/>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  allow from all
</Directory>
```

Puis on redémarre le serveur :

```
sudo /etc/init.d/apache2 restart
```

On peut vérifier dans le navigateur (<http://localhost>, ou bien <http://www.gdtmali.org>), on verra un message informant que *Drupal est en maintenance*. C'est normal, on n'a pas encore configuré sa base de données.

Configuration de MySQL et insertion des données du site web

Il faut d'abord installer MySQL si ce n'est déjà fait. Pour cela, installer les paquets nécessaires :

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

Ensuite, on va redéfinir le mot de passe de l'utilisateur root de MySQL (il arrive que tout ne soit pas parfaitement bien configuré au début) :

```
sudo dpkg-reconfigure mysql-server-5.5
```

Ensuite, mysql propose un script pour sécuriser l'installation (l'installation par défaut n'est pas orientée serveur et sécurité) :

```
sudo mysql_secure_installation
```

Ca va afficher une série de questions :

```
Change the root password? [Y/n] n
Remove anonymous users? [Y/n] y
Disallow root login remotely? [Y/n] y
Remove test database and access to it? [Y/n] y
Reload privilege tables now? [Y/n] y
```

Puis on va créer l'utilisateur gdtweb utilisé pour la base de donnée :

- on se connecte avec l'utilisateur "root" mysql : `mysql -u root -p`

- on crée une base de données vide : `CREATE DATABASE ml_gdtmali;`

- on importe la sauvegarde dans la base : dans une autre console, on se met dans le répertoire de la sauvegarde, là où se trouvent les bases de données (sous-répertoire databases), et on fait :

```
mysql -u root -p ml_gdtmali < 20131118-mysql-sitewebDrupal.sql
```

- on retourne dans la console où on est connecté sur mysql
- on crée l'utilisateur gdtweb : CREATE USER gdtweb IDENTIFIED BY 'gdtweb';
- on lui donne les droits sur la base ml_gdtmali : GRANT ALL PRIVILEGES ON ml_gdtmali.* TO gdtweb;

Et ça devrait être bon. Vérifier dans le navigateur...

Installation et configuration de geoserver (serveur carto)

CONFIGURATION DE TOMCAT7

Pour des instructions d'installation de Tomcat, voir en annexe.

Configurer un utilisateur tomcat pour accéder à l'interface manager

Ouvrir en édition le fichier /etc/tomcat7/tomcat-users.xml, et déclarer un utilisateur pour le rôle manager

```
<role rolename="tomcat"/>
<role rolename="manager-gui"/>
<role rolename="admin-gui"/>
<user username="gn_tomcat" password="gn_tomcat" roles="tomcat,manager-gui,admin"/>
```

(donner, bien sûr, un mot de passe plus sûr !)

Paramétrages mémoire

On augmente les ressources mémoire allouées à tomcat : dans /etc/default/tomcat7, on ajuste les paramètres :

```
JAVA_OPTS="-Djava.awt.headless=true -Xms256M -Xmx1024M -XX:MaxPermSize=256m -XX:PermSize=128m"
```

GEOSERVER

Installation

On va de suite installer GeoServer.

Récupérer dans le CD d'installation le fichier geoserver.war (ou télécharger la dernière archive WAR sur le site de Geoserver). Dans le navigateur, aller sur localhost:8080/manager/html (il faudra vous identifier avec l'identifiant créé ci-dessus), et dans « WAR file to deploy », aller chercher le fichier WAR. Déployer.

Geoserver devrait être à présent listé dans les applications disponibles.

Les fichiers ont été placés dans /var/lib/tomcat7/webapps, répertoire geoserver.

Dans l'idée de stocker les applications installées pour le portail autant que possible dans le répertoire /home/ilwac, on va déplacer geoserver. Pour cela :

- éteindre tomcat : sudo service tomcat7 stop
- créer l'arborescence de destination :

```
mkdir /home/ilwac/tomcat7
```

```
mkdir /home/ilwac/tomcat7/webapps
```

- déplacer le répertoire geoserver :

```
sudo mv /var/lib/tomcat7/webapps/geoserver /home/ilwac/tomcat7/webapps/
```

- déclarer le changement à tomcat : dans le fichier /etc/tomcat7/server.xml, tout à fait à la fin juste avant la balise </Host> on insère :

```
<Context path="/geoserver" docBase="/home/ilwac/tomcat7/webapps/geoserver" crossContext="false" reloadable="false" />
```

- et on redémarre tomcat : `sudo service tomcat7 start`
- Vérifier en visitant la page <http://localhost:8080/geoserver>.

Dernier point : on va rajouter l'extension ImagePyramid utilisée pour plusieurs gros jeux de données raster. Dézipper `geoserver-2.4.1-pyramid-plugin.zip`, copier le fichier `gt-imagepyramid-10.1.jar` dans `/home/ilwac/tomcat7/webapps/geoserver/WEB-INF/lib`, puis changer le propriétaire du fichier : `sudo chown tomcat7:tomcat7 gt-imagepyramid-10.1.jar`.

Configuration

Data dir

On va sortir le répertoire data de GeoServer. Cela est recommandé, afin de ne pas risquer d'écraser les données lors d'une mise à jour de GeoServer.

Par ailleurs, le portail est livré avec un certain nombre de couches carto préconfigurées qu'on va mettre dans ce nouveau répertoire.

Pour cela, on va :

- créer le répertoire de stockage des répertoires data (on y mettra aussi celui de geonetwork) :

```
mkdir /home/ilwac/tomcat7/data
```

- décompresser l'archive fournie :

```
tar zxvfp [support d'installation]/sources/geoserver-datadir.tar.gz
```

- déclarer le changement d'emplacement dans GeoServer. On édite `/var/lib/tomcat7/webapps/geoserver/WEB-INF/web.xml`, et on y entre :

```
<context-param>
  <param-name>GEOSERVER_DATA_DIR</param-name>
  <param-value> /home/ilwac/tomcat7/data/geoserver-datadir</param-value>
</context-param>
```

- On recharge GeoServer (via le manager tomcat, par exemple) et on teste que tout s'est bien passé.
- Maintenant, on peut rajouter le reste des données. Ça représente un gros volume, donc du temps.
- Décompresser `pays.tgz` et `abn.tgz` (`tar -zxvfp pays.tgz`). Attention, `pays.tgz`, décompressé, fait 70Go ! Prévoir la place !
- déplacer les répertoires `pays` et `abn` dans `/home/ilwac/tomcat7/data/geoserver-datadir/data/`
- Redémarrer geoserver, ça devrait marcher.

Couple utilisateur/mot de passe

A l'installation, GeoServer n'est pas du tout sécurisé (couple utilisateur/mot de passe : `admin/geoserver`). Il faut donc changer le couple utilisateur/mot de passe (et éventuellement, en créer un par utilisateur si plusieurs personnes doivent intervenir dessus).

Cela se fait dans l'interface graphique de Geoserver. pour pouvoir changer le mot de passe admin, il faut d'abord créer un nouvel utilisateur avec droits d'administration, se déconnecter et se reconnecter avec ce nouvel utilisateur, et là, on peut changer le mot de passe d'admin.

POSTGRESQL

C'est le SGBD utilisé pour le géoportail, tant pour les données du catalogue que pour celles de la partie carto.

Installation

On installe le paquet **postgresql** (postgresql-9,1), le paquet **php5_pgsq** et éventuellement **pgadmin3** (interface graphique d'administration)

Configuration

cf. <http://doc.ubuntu-fr.org/postgresql>

On va se connecter en tant que postgres :

```
$ sudo -i -u postgres
$ psql
```

Et on va créer un utilisateur avec suffisamment de droits pour se passer de postgres par la suite :

```
postgres=# CREATE USER aedd;
CREATE ROLE
postgres=# ALTER ROLE aedd WITH CREATEDB;
ALTER ROLE
postgres=# ALTER USER aedd WITH ENCRYPTED PASSWORD 'server';
ALTER ROLE
postgres=# ALTER ROLE aedd WITH CREATEUSER;
ALTER ROLE
```

Ensuite, on va créer un 2e utilisateur à droits très réduits, soit à partir de pgAdmin3 (on peut se connecter, à présent, en utilisant l'utilisateur aedd juste créé), ou bien en ligne de commande, puis on se déconnecte :

```
postgres=# CREATE USER geonetwork;
CREATE ROLE
postgres=# ALTER USER geonetwork WITH ENCRYPTED PASSWORD 'geonetwork';
ALTER ROLE
postgres=# \q
postgres@qdtserver:~$ exit
déconnexion
aedd@qdtserver:~$
```

Pour pouvoir se connecter aux bases depuis la console, avec un rôle différent du nom d'utilisateur courant, on va modifier légèrement les droits d'accès postgresql.

On édite `/etc/postgresql/9,1/main/pg_hba.conf`, et on remplace

```
#local all all ident
par
```

```
local all all md5
```

Et on redémarre postgresql :

```
sudo service postgresql restart
```

Installation de postGIS

On rajoute PostGIS, qui est l'extension spatiale de postgresQL :

```
sudo apt-get install postgis
```

Import des bases

On va importer les bases de données. Il y en a trois : une base pour les données géospatiales publiées dans

le portail, une pour les données non-géospatiales utilisées dans les services web, et la base du géocatalogue.

On crée d'abord ces trois bases :

- Base sigdt : données non-geospatiales des services web

```
createdb -E UTF8 --owner geonetwork sigdt
```

- Base ilwac_geodata : données geospatiales postGIS

```
createdb -E UTF8 --owner geonetwork lwac_geodata
```

- Base ml_ilwac_gn2_10 : base geonetwork

```
createdb -E UTF8 --owner geonetwork ml_ilwac_gn2_10
```

On dézippe les dumps postgresql se trouvant dans le répertoire sources/databases, et on importe les contenus :

```
$ psql -d sigdt <sigdt .sql
$ psql -d lwac_geodata <lwac_geodata .sql
$ psql -d ml_ilwac_gn2_10 <ml_ilwac_gn2_10.sql
```

A noter que pour la base ilwac_geodata, deux version sont disponibles selon que l'on a une version 1.5.x ou 2.x de postgis. Il s'agit de prendre la bonne !

Des messages d'erreur du type 'le rôle « sigdt » n'existe pas' vont s'afficher. Ne pas en tenir compte, c'est normal, on va régler ça maintenant : les différences de configuration d'un serveur à un autre, font que geonetwork ne va pas être toujours identifié comme propriétaire de toutes les tables des bases. Par défaut, suite à ces erreurs de rôle, c'est le rôle aedd, l'exécutant, qui est mis comme propriétaire. Ce qui nous pose problème, car le rôle geonetwork n'aura alors pas le droit d'accéder à certaines d'entre elles. On va donc tout réaffecter à geonetwork :

```
aedd@gdtserver:~$ psql -d sigdt
psql (9.1.9)
Saisissez « help » pour l'aide.

sigdt=# REASSIGN OWNED BY aedd TO geonetwork;
REASSIGN OWNED
sigdt=# REASSIGN OWNED BY aedd TO geonetwork;^C
sigdt=# \q
aedd@gdtserver:~$ psql -d ilwac_geodata
psql (9.1.9)
Saisissez « help » pour l'aide.

ilwac_geodata=# REASSIGN OWNED BY aedd TO geonetwork;
REASSIGN OWNED
ilwac_geodata=# \q
aedd@gdtserver:~$ psql -d ml_ilwac_gn2_10
psql (9.1.9)
Saisissez « help » pour l'aide.

ml_ilwac_gn2_10=# REASSIGN OWNED BY aedd TO geonetwork;
REASSIGN OWNED
ml_ilwac_gn2_10=#
```

Installation des scripts de service web (php) pour le portail

installation des scripts de services web pour le portail

- on installe gdal si ce n'est déjà fait :

```
sudo apt-get install gdal-bin
```

- On crée un répertoire /home/ilwac/www/services. On décompresse les 3 archives public_html_* dedans. Attention, c'est gros ; au total 37Go.
- On règle bien les droits sur les fichiers :

```
sudo chown -R aedd:www-data /home/ilwac/www
sudo chmod g+s -R /home/ilwac/www
```

- on rajoute une déclaration dans /etc/apache2/sites-available/default :

```
Alias /services "/home/ilwac/www/services"
<Directory /home/ilwac/www/services/>
  Options Indexes FollowSymLinks MultiViews
  AllowOverride None
  Order allow,deny
  allow from all
</Directory>
```

- On ajuste les paramètres de connexion dans les fichiers sigdt-config/config-ilwac/db/pg_*.php.
- On redémarre apache (sudo service apache2 restart)

On teste que ça marche en visitant <http://localhost/services/sigdt-config/services-ilwac/getCercles.php> par exemple.

Installation et configuration du geoportail (géocatalogue + interface web)

GEOCATALOGUE & SI-GDT

Copier les fichiers

On copie le géocatalogue. On va lui aussi le placer dans le répertoire /home/ilwac/tomcat7/webapps

```
cd /home/ilwac/tomcat7/webapps
tar zxvfp [support d'installation]/sources/ml-ilwac-gn2_10.tgz
```

et le répertoire data (stocke les données téléchargées, etc) :

```
cd /home/ilwac/tomcat7/data
tar zxvfp [support d'installation]/sources/ml-ilwac-gn2_10-datadir.tgz
```

Et enfin, on crée un répertoire pour les fichiers à attacher aux métadonnées, mais trop gros pour passer par l'interface de téléchargement de geonetwork. Il existe déjà, il se trouve dans /home/ilwac/www/services et s'appelle gn_depot. On va le déplacer et le déclarer dans la conf apache, de sorte à y accéder via http://gdtmali.org/gn_depots :

```
sudo mv /home/ilwac/www/services/gn_depot /home/ilwac/www/
```

Et on déclare ce répertoire auprès d'apache, exactement comme pour le répertoire 'services'.

Ajuster les droits d'accès

Certains répertoires du dossier de geonetwork (ml-ilwac-gn2_10) doivent être la propriété de tomcat7 pour que l'application fonctionne correctement (il doit pouvoir écrire dans ces répertoires). On va donc les lui ré-attribuer si nécessaire :

```
sudo chown -R tomcat7:tomcat7 /home/ilwac/tomcat7/webapps/ml-ilwac-gn2_10
```

```
sudo chown -R tomcat7:tomcat7 /home/ilwac/tomcat7/data/ml-ilwac-gn2_10-datadir
```

Paramétrer le géocatalogue (vérifier les paramètres)

Data dir

Dans `/home/ilwac/tomcat7/webapps/ml-ilwac-gn2_10/WEB-INF/config-overrides-prod.xml`, on ajuste le chemin vers le répertoire des données :

```
<addXML xpath="/appHandler"><param name="geonetwork.dir" value="/home/ilwac/tomcat7/data/ml-ilwac-gn2_10-datadir" /></addXML>
```

Fichier de log

Dans `/home/ilwac/tomcat7/webapps/ml-ilwac-gn2_10/WEB-INF/log4j.cfg`, on ajuste le chemin du fichier de log :

```
log4j.appender.jeeves.file = /home/ilwac/tomcat7/data/ml-ilwac-gn2_10-datadir/logs/geonetwork.log
```

Et on crée le répertoire `/home/ilwac/tomcat7/data/ml-ilwac-gn2_10-datadir/logs`

Fichiers de configuration du portail

On doit aussi mettre à jour l'adresse (nom de domaine) du portail en plusieurs endroits :

- ouvrir le fichier `/home/ilwac/tomcat7/webapps/ml-ilwac-gn2_10/apps/geoportal/index.html`, et avec un éditeur de texte supportant le chercher/remplacer, remplacer toutes les occurrences de '<http://localhost/sigdt-config/>' par '<http://gdtmali.org/services/sigdt-config/>' et toutes les occurrences de 'ilwac.ige.fr' par 'gdtmali.org'. Remplacer également la ligne 'offline:true,' par 'offline:false,'
- ouvrir le fichier `/home/ilwac/www/services/sigdt-config/config-ilwac/infotree.js` et remplacer toutes les occurrences de '<http://ilwac.ige.fr/sigdt-config/>' par '<http://gdtmali.org/services/sigdt-config/>'
- ouvrir le fichier `/home/ilwac/www/services/sigdt-config/config-ilwac/layertree.js` et remplacer toutes les occurrences de '<http://ilwac.ige.fr/geoserver-prod/>' par '<http://gdtmali.org/geoserver/>'

Paramétrer Tomcat

Comme lors de l'installation de geoserver, on doit déclarer l'application à tomcat : à la fin de `/etc/tomcat7/server.xml`, on rajoute la ligne

```
<Context path="/ml-ilwac-gn2_10" docBase="/home/ilwac/tomcat7/webapps/ml-ilwac-gn2_10" crossContext="false" reloadable="false" />
```

Cela signifie que maintenant (lorsqu'on aura redémarré tomcat), l'application sera trouvée par tomcat, et identifiée comme "ml-ilwac-gn2_10". On pourra donc y accéder à l'adresse suivante : `http://localhost:8080/ml-ilwac-gn2_10`

Il est possible de redémarrer tomcat dès maintenant et de la vérifier (`sudo service tomcat7 restart`)

Connexion Tomcat / Apache

Certains éléments du géoportail (en particulier pour le SI-GDT) exploitent du code fonctionnant directement sous Apache. En outre, il est plus propre de s'affranchir de la mention du port 8080 qui encombre les adresses utilisant tomcat.

On va donc configurer Apache et Tomcat pour que ce dernier passe systématiquement par Apache. Apache va donc fonctionner comme proxy pour les applications Tomcat.

Cela se fait assez simplement avec le module apache `mod_proxy_ajp`

Activer mod_proxy_ajp

Dans la console :

```
$ sudo a2enmod proxy_ajp
```

Si l'activation se passe bien, on aura un message du type :

```
Considering dependency proxy for proxy_ajp:
Enabling module proxy.
Enabling module proxy_ajp.
Run '/etc/init.d/apache2 restart' to activate new configuration!
```

Avant de redémarrer apache2, on va d'abord déclarer le site ml-ilwac-gn2_10

Activer le connecteur AJP sur Tomcat

Dans le fichier /etc/tomcat7/server.xml, dé-commenter la ligne

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

Puis redémarrer Tomcat (sudo /etc/init.d/tomcat7 restart)

Déclarer les sites

Créer un nouveau fichier /etc/apache2/sites-available/tomcat_apps contenant :

```
<IfModule mod_proxy_ajp.c>
ProxyRequests On
ProxyVia On
ProxyPreserveHost On

ErrorLog /var/log/apache2/apache-tomcat.error.log
CustomLog /var/log/apache2/apache-tomcat.log combined

<Location /geoserver>
Order allow,deny
Allow from all
ProxyPass ajp://localhost:8009/geoserver
ProxyPassReverse ajp://localhost:8009/geoserver
</Location>
<Location /ml-ilwac-gn2_10>
Order allow,deny
Allow from all
ProxyPass ajp://localhost:8009/ml-ilwac-gn2_10
ProxyPassReverse ajp://localhost:8009/ml-ilwac-gn2_10
</Location>
</IfModule>
```

Et activer le site :

```
$ sudo a2ensite tomcat_apps
```

Redémarrer Apache

```
sudo /etc/init.d/apache2 force-reload
```

Le serveur doit à présent être opérationnel. On peut le vérifier en visitant http://www.gdtmali.org/ml-ilwac-gn2_10.

On va effectuer, d'ailleurs, deux derniers changements :

- dans l'interface d'administration de geonetwork. Se connecter (identifiant admin, mot de passe ilwac2013), aller dans l'interface d'administration, et aller dans 'Configuration du système'. Changer le champ 'hôte' pour www.gdtmali.org, et sauvegarder.

Le nom d'hôte est utilisé pour construire les adresses d'accès aux données liées aux fiches, imageries, etc.

Vous pouvez en profiter pour changer le mot de passe d'admin.

- un changement sur la base de données ml_ilwac_gn2_10, en rapport avec la première modif : celle-ci influe sur les métadonnées à venir. Mais il faut encore retoucher les métadonnées existantes, dont on va actualiser les liens.
 - se connecter dans une console, sur la base postgresql ml_ilwac_gn2_10 :

```
psql -d ml_ilwac_gn2_10
```

- exécuter une commande de remplacement de chaînes de caractères (chercher-remplacer) :

```
UPDATE metadat SET data=replace(data, 'http://ilwac.ige.fr/ml-ilwac-geocat',  
'http://www.gdtmali.org/ml_ilwac_gn2_10');
```

Et on va vérifier que ça a marché. Ouvrir la page du portail dans le navigateur (http://www.gdtmali.org/ml-ilwac-gn2_10), lancer une recherche, ouvrir une fiche de métadonnées, et regarder le lien de téléchargement proposé. Il doit commencer par <http://www.gdtmali.org>. Sinon, repérer le texte à remplacer, et refaire une requête rechercher/remplacer.

Penser aussi à mettre à jour le lien dans le site web, pour remplacer les anciens liens vers le si-gdt et le géocatalogue par un lien vers ce nouveau portail.

Pour tout ce qui touche à l'utilisation et l'administration des applications, se référer aux documentations utilisateur et administrateurs correspondants.