

MANUALE INSTALLAZIONE OPENMAINT IN UBUNTU 16.04

OpenMAINT è una soluzione open source per il **Property & Facility Management**.

E' cioè una applicazione per la **gestione di asset mobili** (impianti, elementi tecnici, arredamento, ecc) ed immobili (edifici, infrastrutture sul territorio, ecc) e delle relative attività manutentive (programmate e a guasto), logistiche ed economiche. Una applicazione che aiuta a conoscere, organizzare e mantenere aggiornati i dati relativi al patrimonio immobiliare, a supporto dell'attività decisionale ed operativa dell'ente gestore.

Rappresenta pietra miliare per la possibile integrazione con dispositivi IOT e Domotici di prossima generazione. Pensate solo alla possibile supervisione di ogni guasto dell'edificio da remoto, con una panoramica in tempo reale dei possibili inconvenienti e anche con una "difesa predittiva" dei possibili malfunzionamenti che accorcerebbero i tempi di ripristino e i danni che vengono generati da incidenti improvvisi. I sensori IOT vanno in questa direzione e posso essere integrati in un discorso complessivo di gestione degli asset.

Utile soprattutto a chi naturalmente fa della gestione di questi impianti o asset mobili il proprio business, e a chi ha la responsabilità di tutelare e proteggere alcuni beni, privati o pubblici che siano.

Strumenti quali **OPENMAINT** forniscono almeno informazioni complete ai decisori-protempore di turno, magari collegati ad altri programmi che consentano l'interfacciamento con l'utente del servizio reso dal bene, per una sua possibile collaborazione (magari anche denunciando malfunzionamenti in corso).

openMAINT può essere scaricato liberamente dal sito web dedicato, essendo rilasciato con licenza open source AGPL 3, e può essere utilizzato gratuitamente. Sono previsti naturalmente servizi aggiuntivi a pagamento.

Per chi volesse approfondire rimandiamo al manuale completo <http://www.openmaint.org/files/manuale-pdf>.

Come già fatto con il nostro [Manuale GLPI per il parco informatico](#) riteniamo che la diffusione delle competenze e della conoscenza di questi prodotti contribuisca alla crescita del sentimento comune di gestione tecnica e consapevole dei beni di cui si ha la responsabilità.

Abbiamo rilevato che uno dei limiti nell'utilizzo di questo strumento è l'installazione difficoltosa, sia in ambiente linux che Windows, per cui ci sembrava utile fornire alcune indicazioni step-by-step affinché fosse meglio valutato e la nostra esperienza di installazione andasse a beneficio di altri possibili fruitori di questo fantastico prodotto. Per quella che è la nostra esperienza di lavoro le strutture piu' avanzate avranno comunque la necessità di supporti tecnici e di affidabilità piu' evoluti, anche pagando la loro tranquillità con canoni di assistenza adeguati. Con questo manuale vorremmo solo aiutare i tecnici IT che hanno responsabilità in questo settore a conoscere una soluzione utile per la loro attività.

Il nostro obiettivo è solo quello di fornire utili indicazioni di installazione, presentando la nostra esperienza... non neghiamo esistano altre soluzioni, forse migliori... ma ci auspichiamo che queste indicazioni siano utili.

INSTALLAZIONE OPENMAINT IN UBUNTU 16.04

Effettuata l'installazione di Ubuntu (in un PC o macchina virtuale) dotato di almeno 8 Gb di RAM – con hard disk da almeno 120 gb e quad core cpu. (occorre tener conto che volendo sfruttare appieno le caratteristiche di OPENMAINT, soprattutto relative alla grafica e al rendering 3D, la dotazione dovrebbe essere molto piu' "pesante" sia in termini di ram che per la scheda grafica)

Ci colleghiamo con "putty" [scaricabile gratuitamente](#), indicando l'IP del server ed elevandoci al livello di root con sudo su e la password indicata nell'installazione:

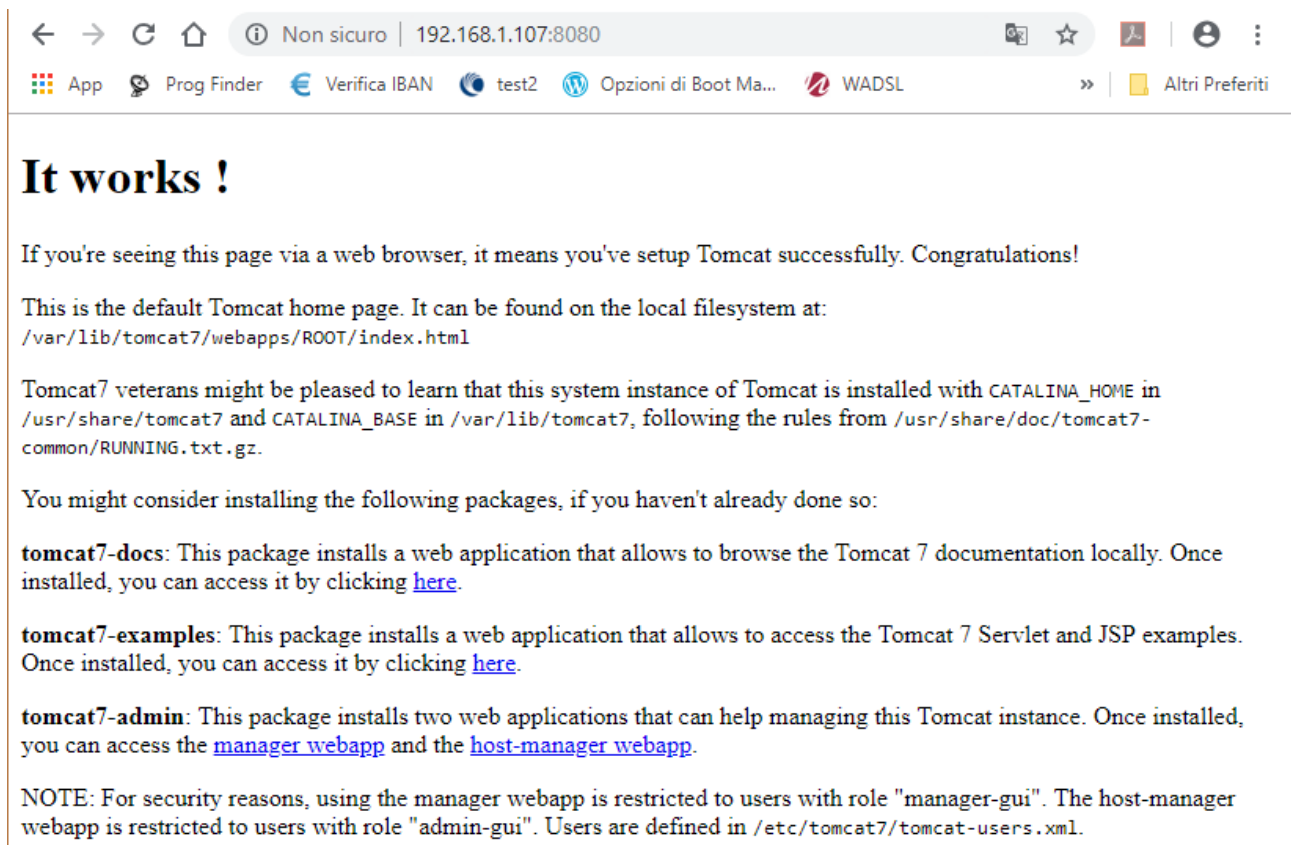
```
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu:/home/ubuntu# apt-get update
Hit:1 http://it.archive.ubuntu.com/ubuntu xenial InRelease
Get:2 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]
Hit:3 http://it.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://it.archive.ubuntu.com/ubuntu xenial-backports InRelease
Fetched 109 kB in 0s (203 kB/s)
Reading package lists... Done
root@ubuntu:/home/ubuntu#
```

La prima fase è l'installazione di Tomcat (in questo caso abbiamo utilizzato la versione 8)

```
sudo apt-get install tomcat8 tomcat8-admin tomcat8-examples
```

potremo testarne il funzionamento richiamando l'ip del pc e la porta 8080 su un browser

<http://192.168.1.xxx:8080>



Importante! Dobbiamo configurare l'utente Tomcat come root (questo per una esigenza del programma Tecnoteca per le caratteristiche di **shark**)

```
nano /etc/default/tomcat8
```

```
# Run Tomcat as this user ID. Not setting this or leaving it blank will use the
# default of tomcat8.
TOMCAT8_USER=root

# Run Tomcat as this group ID. Not setting this or leaving it blank will use
# the default of tomcat8.
TOMCAT8_GROUP=tomcat8
```

Occorre inoltre incrementare la memoria jvm editando la linea JAVA_OPTS

```
JAVA_OPTS="-Djava.awt.headless=true -Xms128m -Xmx1024m -XX:+UseConcMarkSweepGC"
```

```
# exactly one CPU chip that contains one or two cores, you should also add
# the "-XX:+CMSIncrementalMode" option.
# JAVA_OPTS="-Djava.awt.headless=true -Xmx128m -XX:+UseConcMarkSweepGC"
JAVA_OPTS="-Djava.awt.headless=true -Xms128m -Xmx1024m -XX:+UseConcMarkSweepGC"

# To enable remote debugging uncomment the following line.
```

Start & Stop Tomcat

```
sudo systemctl start tomcat8
```

```
sudo systemctl stop tomcat8
```

In caso di memoria insufficiente avremo un messaggio

A questo punto installo la gui grafica per Ubuntu (vista l'esigenza di utilizzare Pgadmin3 con un'interfaccia grafica)

```
sudo apt-get install xubuntu-desktop
```

Facciamo quindi ripartire il server (in questo modo tomcat avra' l'utente root)

```
Shutdown -r now
```

Installiamo PostgreSQL 9.5 e Postgis 2.2

```
apt-get install postgresql-9.5 postgresql-contrib-9.5 postgresql-9.5-postgis-2.2 postgresql-9.5-postgis-scripts
```

Questo creerà un nuovo utente sul tuo PC con il nome "postgres". Imposta la tua password per questo account utilizzando il comando seguente e prendine nota (nel mio caso ho utilizzato la password 'postgres')

```
passwd postgres
```

```
root@ubuntu:/home/ubuntu# passwd postgres
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

Installiamo pgadmin3

```
apt-get install pgadmin3
```

Creiamo quindi un ruolo Openmaintuser e shark (per non dimenticarcelo)

```
sudo -i -u postgres
```

```
psql
```

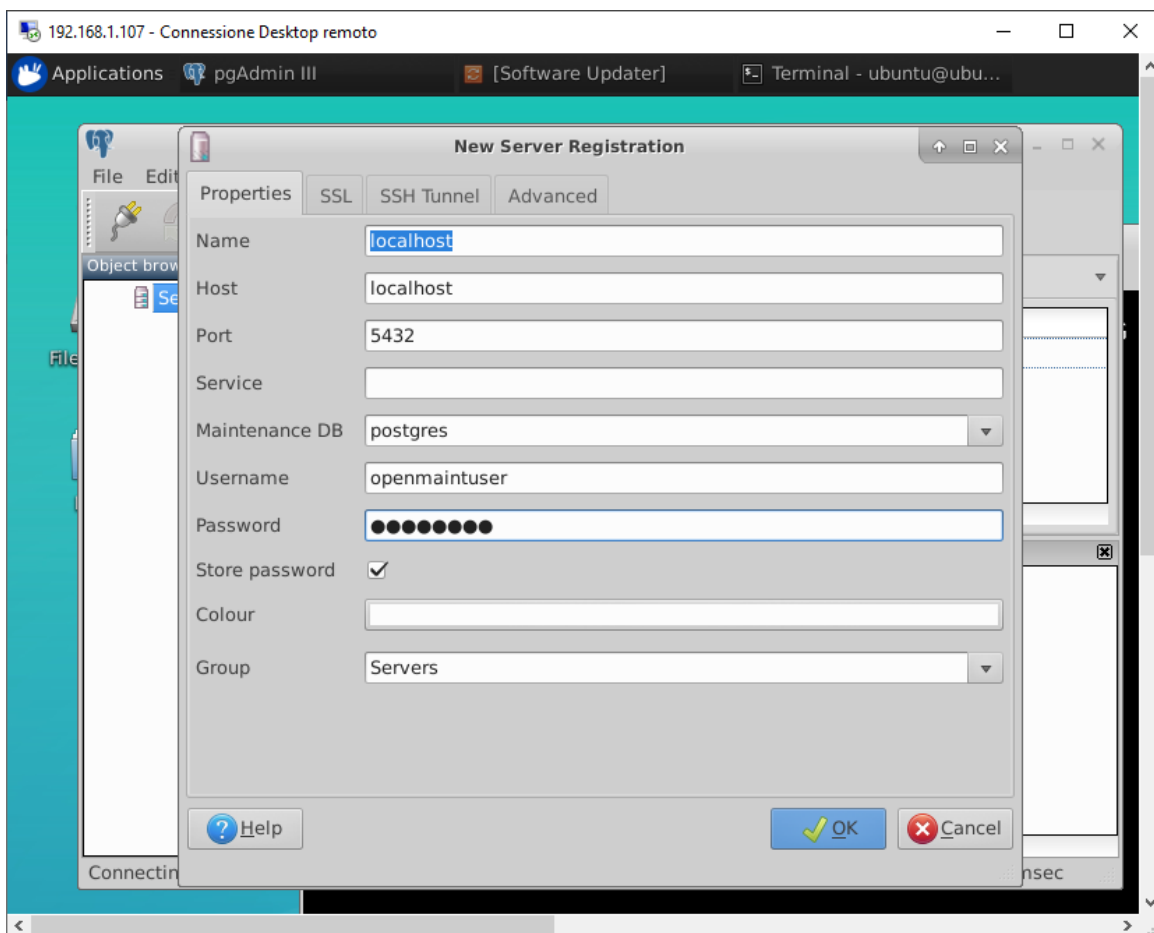
```
create user openmaintuser with password 'postgres' superuser;
```

```
create user shark with password 'shark';
```

```
postgres=# create user openmaintuser with password 'postgres' superuser;
CREATE ROLE
postgres=# create user shark with password 'shark';
CREATE ROLE
postgres=#
```

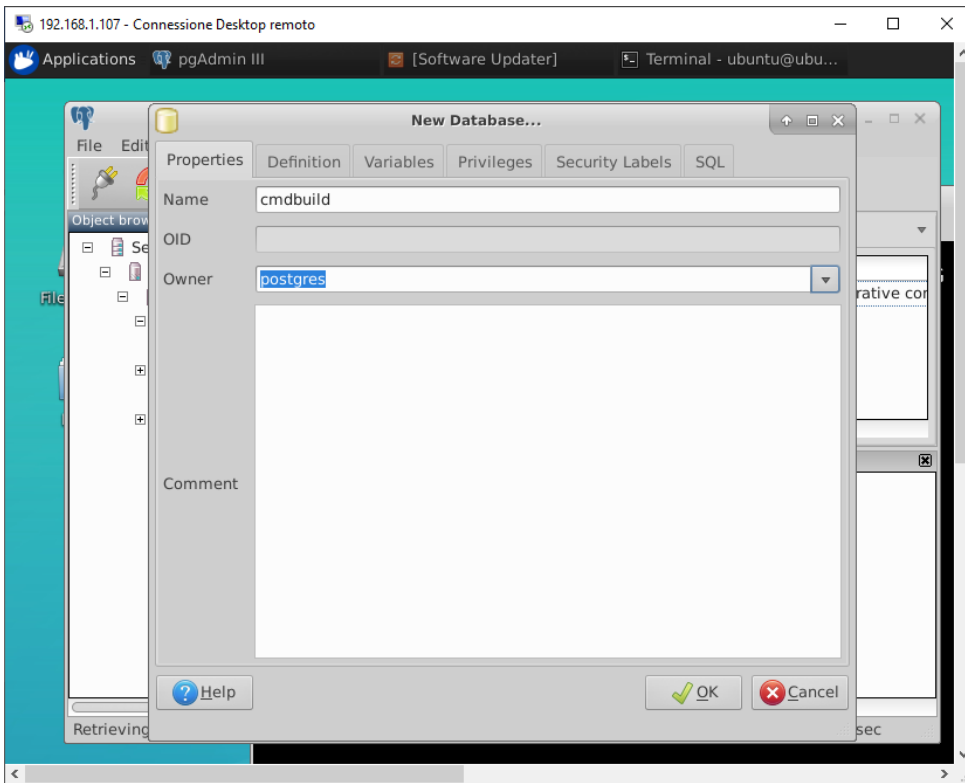
Potremo quindi utilizzare Pgadmin3

```
pgadmin3
```

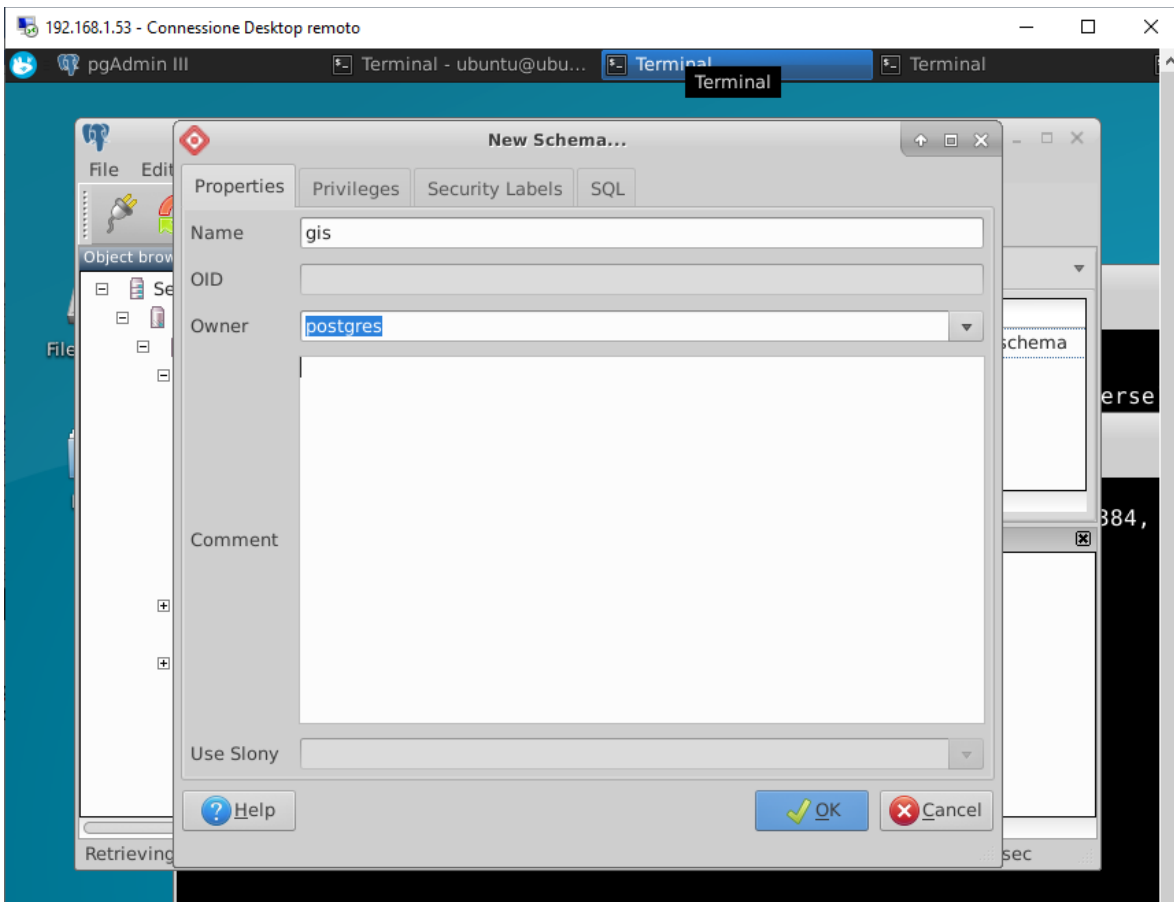


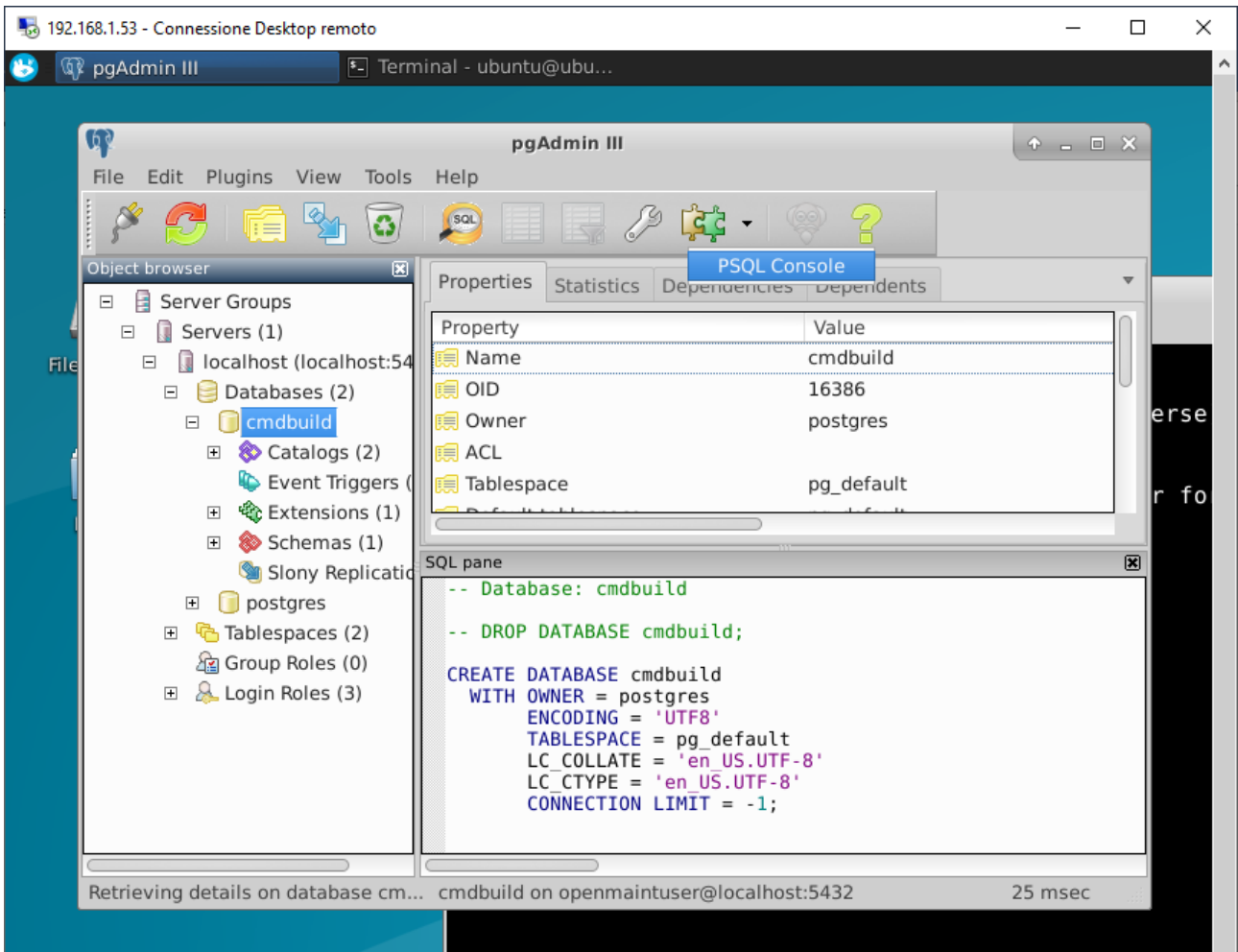
per:

```
create database cmdbuild with owner postgres
```

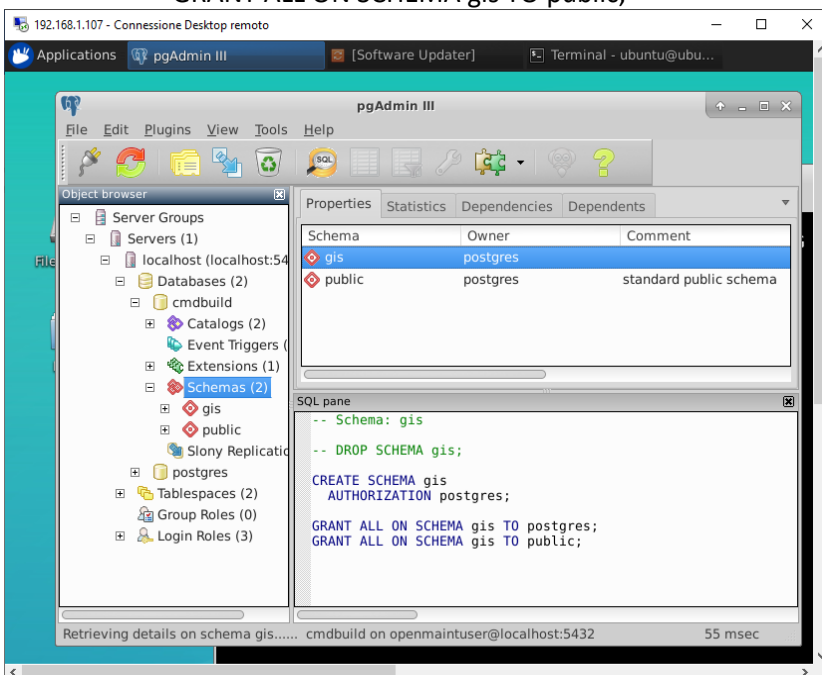


Creare schema gis

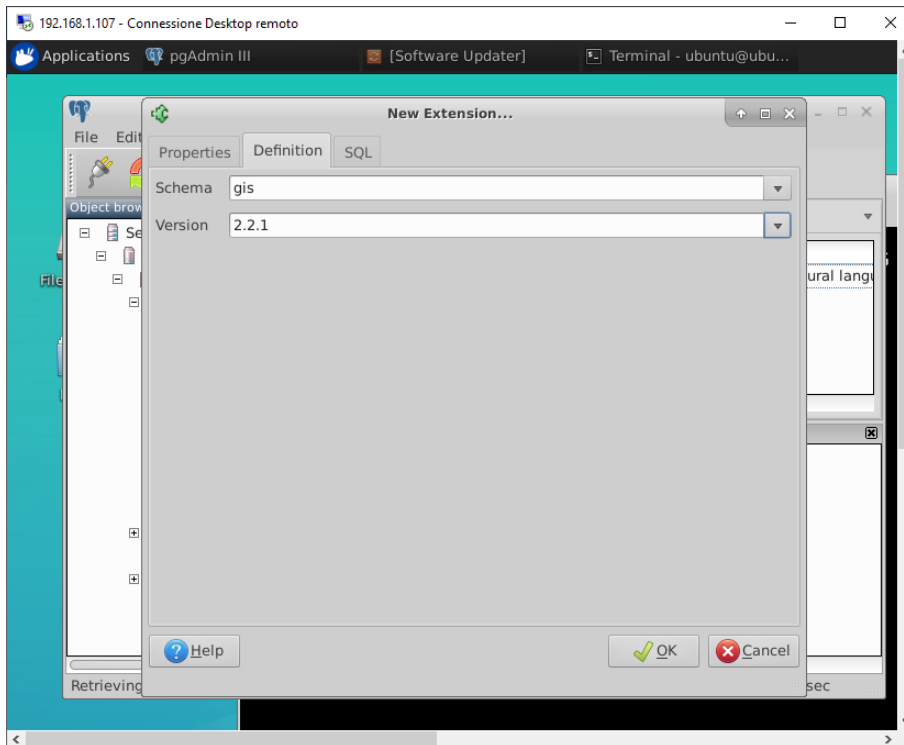




GRANT ALL ON SCHEMA gis TO postgres;
GRANT ALL ON SCHEMA gis TO public;



Creare l'estensione postgis con il collegamento allo schema gis



Usando psql provendiamo ad alcune integrazioni:

```
sudo su postgres  
psql cmdbuild
```

Settiamo il path su gis e installiamo le funzioni legacy

```
SET SEARCH_PATH TO gis, public;  
\i /usr/share/postgresql/9.5/contrib/postgis-2.2/legacy.sql;
```

```
TRUNCATE TABLE gis.spatial_ref_sys;
```

Add std OpenMaint row back into spatial_ref_sys

```
INSERT INTO "spatial_ref_sys" ("srid","auth_name","auth_srid","srtext","proj4text") VALUES  
(900913,'spatialreferencing.org',900913,','+'proj=merc +a=6378137 +b=6378137 +lat_ts=0.0  
+lon_0=0.0 +x_0=0.0 +y_0=0 +units=m +k=1.0 +nadgrids=@null +no_defs');
```

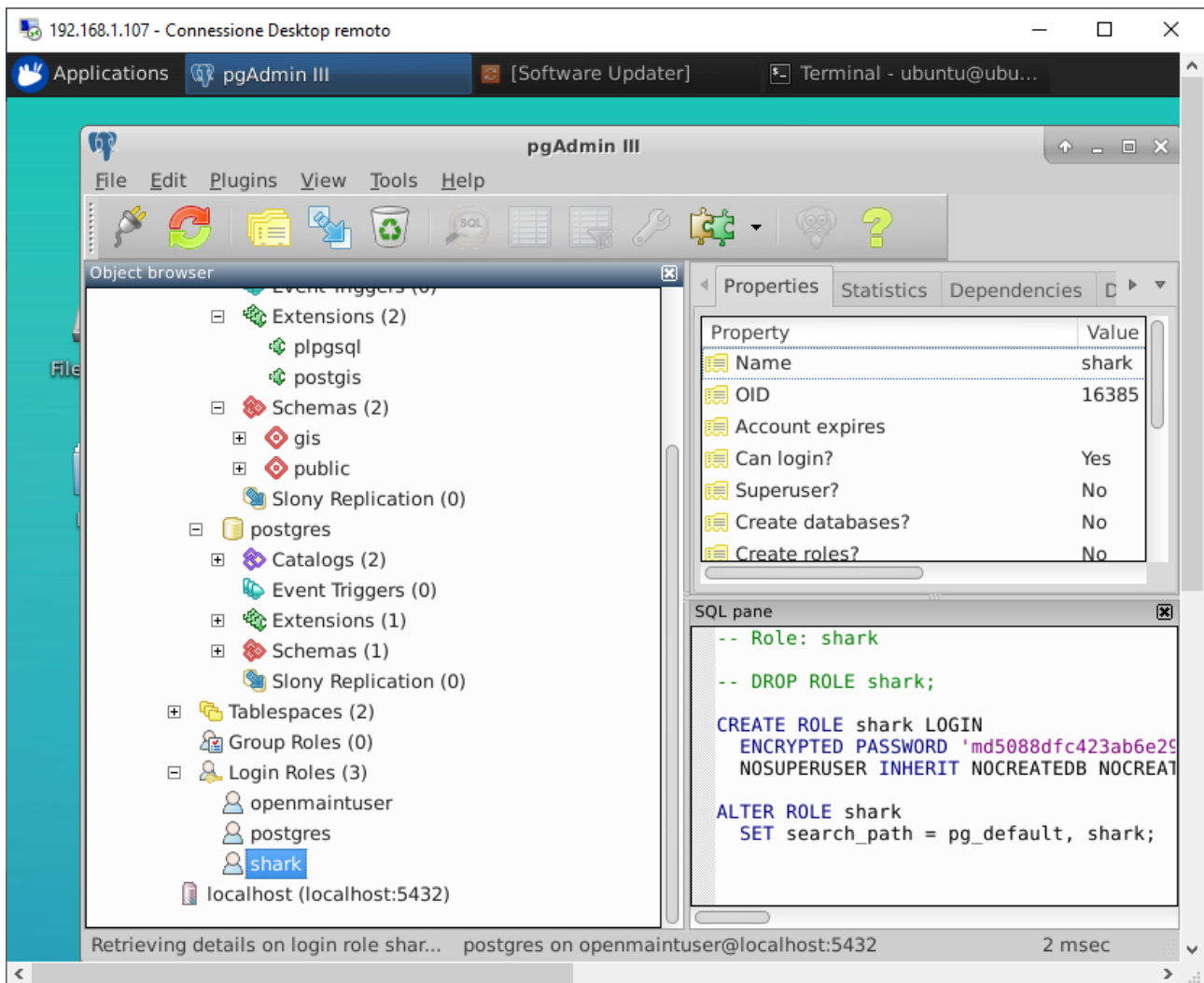
Reset il Path:

```
ALTER DATABASE "cmdbuild" SET search_path="$user", public, gis;
```

Definiamo anche Shark

```
ALTER ROLE shark SET search_path=pg_default,shark;
```

e verifichiamo:



Installare Java

```
sudo apt-get install default-jre
```

Installiamo anche alcune utility relative:

```
apt-get install ant git
```

Java è neutrale rispetto alla piattaforma, si tratta semplicemente di scaricare il file JAR appropriato e rilasciarlo nel classpath. Scarichiamo la versione richiesta, che può essere scaricata utilizzando il [link al driver JDBC PostgreSQL](#).

Useremo il comando seguente per ottenere il driver JDBC richiesto.

```
wget https://jdbc.postgresql.org/download/postgresql-42.1.4.jre6.jar
```

Quindi, usando il comando di sotto, copia questo nella directory 'lib' di tomcat.

```
cp postgresql-42.1.4.jre6.jar /usr/share/tomcat8/lib
```


Scarica e installa OpenMaint

L'ultima versione di OpenMaint può essere scaricata dalla loro [pagina web ufficiale](#) .

```
wget https://downloads.sourceforge.net/project/openmaint/1.1/openmaint-1.1-2.4.2.zip
```

Una volta scaricato il pacchetto, estrarlo, inserire la directory e copiare le librerie in "extras / tomcat-libs" usando i comandi sottostanti.

```
unzip openmaint-1.1-2.4.2.zip
```

```
cd openmaint-1.1-2.4.2
```

```
cp ./extras/tomcat-libs/6.0\ or\ higher/postgresql-9.4.1207.jar /usr/share/tomcat8/lib/
```

Quindi copia openmaint- {version} .war su 'openmaint.war' usando i comandi sottostanti.

```
cp openmaint-1.1-2.4.2.war /var/lib/tomcat8/webapps/openmaint.war
```

Quindi copieremo cmdbuild-shark-server- {versione} .war in 'shark.war' usando il comando di sotto.

```
cp openmaint-shark-server-1.1-2.4.2.war /var/lib/tomcat8/webapps/shark.war
```

I file .war verranno estratti stoppiamo quindi tomcat

```
service tomcat8 stop
```

dopodiché potrai eliminarli inserendo il seguente comando:

```
rm -r /var/lib/tomcat8/webapps/shark.war /var/lib/tomcat8/webapps/openmaint.war
```

Settiamo il linguaggio su Openmaint:

```
sudo nano /var/lib/tomcat8/webapps/openmaint/WEB-INF/conf/cmdbuild.conf
```

e decommentiamo il linguaggio

```
## Language
language=it
languageprompt=true
```

Settiamo la connessione con il database

```
sudo nano /var/lib/tomcat8/webapps/openmaint/WEB-INF/conf/database.conf
```

decommentiamo e modifichiamo i collegamenti

```
db.url=jdbc\:postgresql\://localhost/cmdbuild
db.username=openmaintuser
db.password=12345abcde
```

Settiamo la connessione shark a openmaint

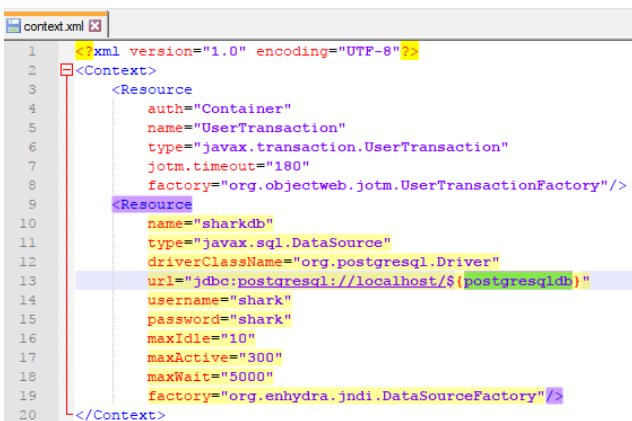
`sudo nano /var/lib/tomcat8/webapps/shark/conf/Shark.conf`

e (alla fine del documento) cambiamo la connessione a openmaint

```
# CMDBuild connection settings
org.cmdbuild.ws.url=http://localhost:8080/openmaint/
org.cmdbuild.ws.username=workflow
org.cmdbuild.ws.password=changeme

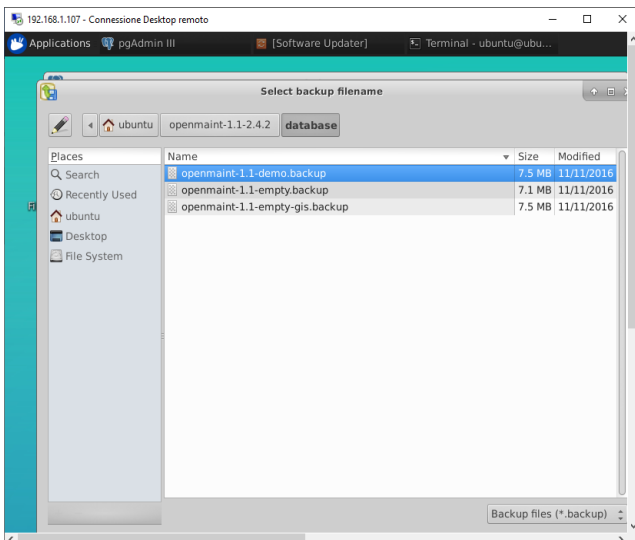
# GroovyToolAgent settings
#CMDBuild.Groovy.Repository=
```

`nano /var/lib/tomcat8/webapps/shark/META-INF/context.xml`
sostituiamo `$postgresqldb` con `cmdbuild`



```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <Context>
3   <Resource
4     auth="Container"
5     name="UserTransaction"
6     type="javax.transaction.UserTransaction"
7     jtm.timeout="180"
8     factory="org.objectweb.jtm.UserTransactionFactory"/>
9   <Resource
10    name="sharkdb"
11    type="javax.sql.DataSource"
12    driverClassName="org.postgresql.Driver"
13    url="jdbc:postgresql://localhost/${postgresqldb}"
14    username="shark"
15    password="shark"
16    maxIdle="10"
17    maxActive="300"
18    maxWait="5000"
19    factory="org.enhydra.jndi.DataSourceFactory"/>
20 </Context>
```

Proviamo al restore-ripristino dei database (demo) :

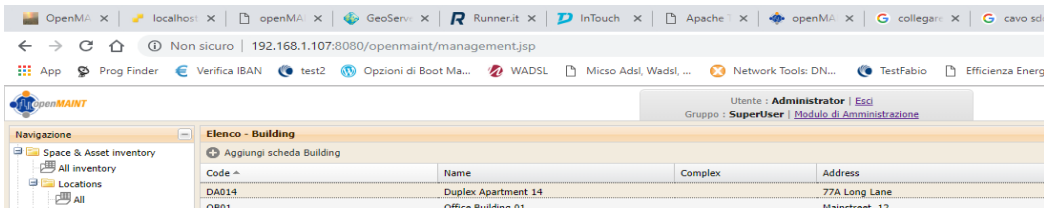


Dopodichè possiamo provvedere al riavvio del nostro server

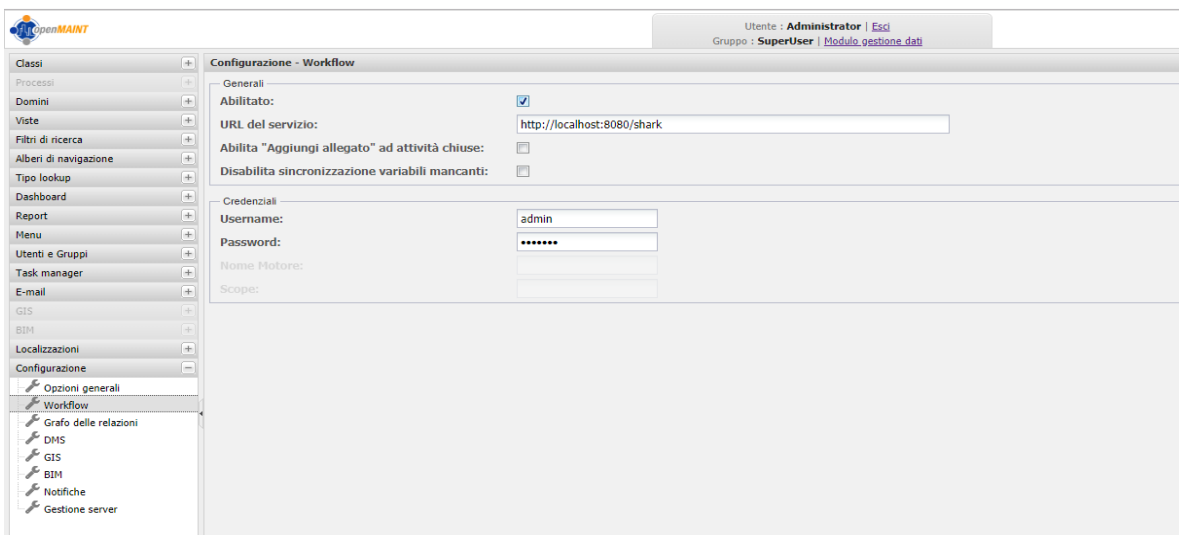
`shutdown -r now`

Al riavvio del server è opportuno entrare in amministrazione per configurare il workflow

Quindi <http://xxx.xxx.xxx.xxx:8080/openmaint>



Eventuali messaggi di errore relativi a class non trovate saranno risolti dall'attivazione del workflow shark



INSTALLAZIONE GEOSERVER 2.15.0

GeoServer 2.15.0

Released on February 28th, 2019

- [Changelog](#)
JIRA release 16736
- [Announcement](#)
GeoServer blog.

Packages

- [Platform Independent Binary](#)
Operating system independent runnable binary.
- [Windows Installer](#)
Installer for Windows platforms.
- [Web Archive](#)
Web Archive (war) for servlet containers.

Documentation

- [User Guide UTM](#)
- [Tutorials](#)

Da Web archive:

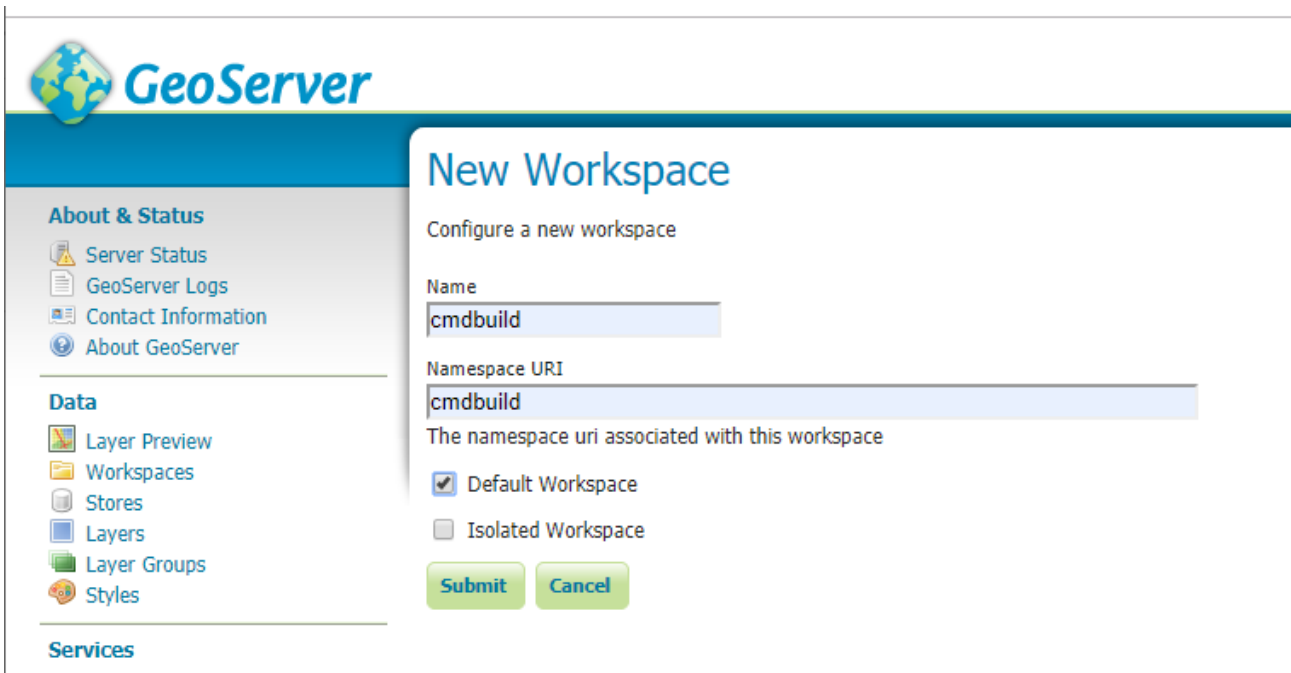
wget <http://sourceforge.net/projects/geoserver/files/GeoServer/2.15.0/geoserver-2.15.0-war.zip>

unzip e copia su /var/lib/tomcat8/webapps

```
root@ubuntu:/home/ubuntu# unzip geoserver-2.15.0-war.zip
Archive:  geoserver-2.15.0-war.zip
  inflating: geoserver.war
  inflating: GPL.txt
  inflating: LICENSE.txt
   creating: target/
  inflating: target/VERSION.txt
root@ubuntu:/home/ubuntu# cp geoserver.war /var/lib/tomcat8/webapps
root@ubuntu:/home/ubuntu#
```

Una volta attivo entrarci via web e modificare il workspaces

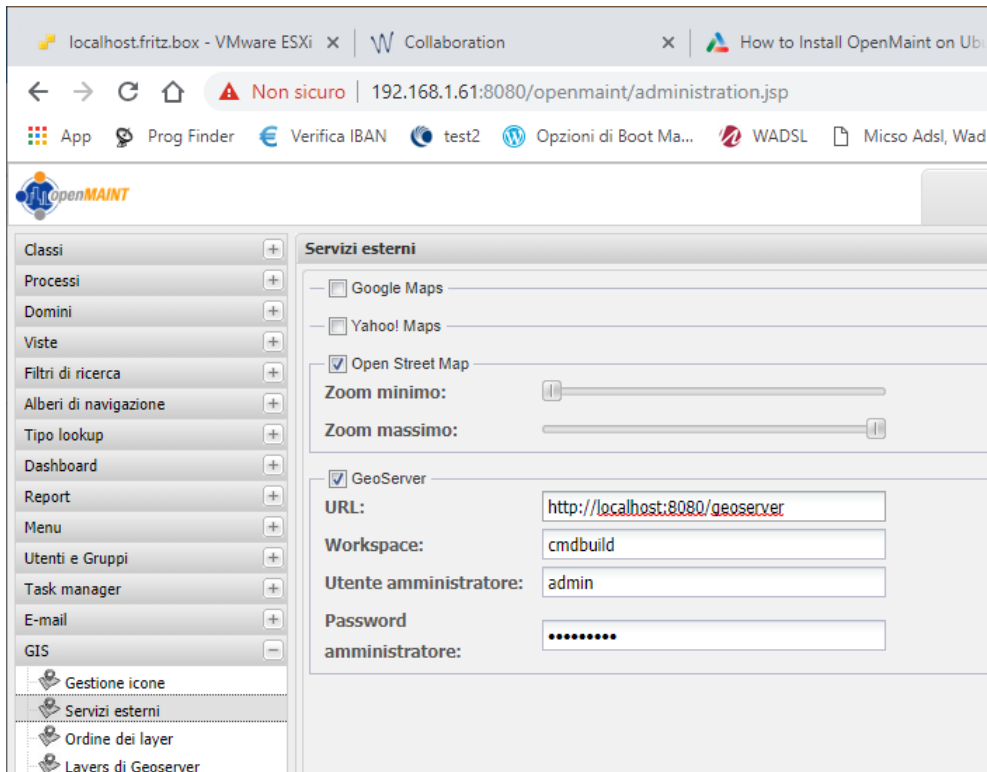
- **Data > Workspaces**
- Creare workspace cmdbuild
- Name cmdbuild
- Uri cmdbuild
- Make Default



Dopodiche' occorre configurare Openmaint

Configure OpenMaint

- **Admin Module > GIS > External Services**
- Tick Geoserver
- URL <http://localhost:8080/geoserver>
- Workspace cmdbuild
- User admin Pw geoserver
- Save



Installare Alfresco

wget <https://download.alfresco.com/release/community/201707-build-00028/alfresco-community-installer-201707-linux-x64.bin>

I am going to make a special directory for the alfresco app

```
> sudo mkdir -p /10x13/apps/alfresco
```

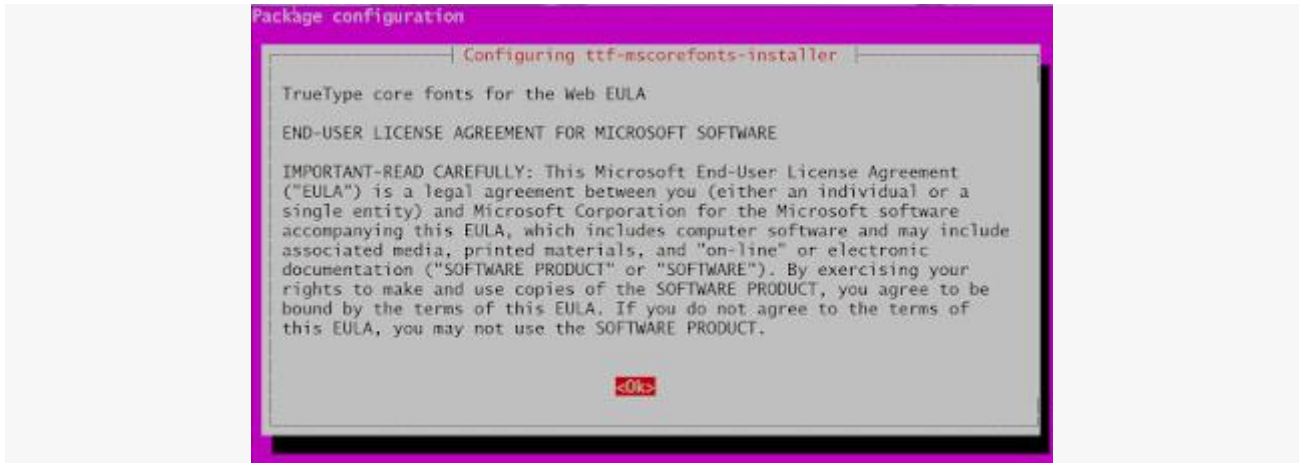
First we need to install some libraries that Alfresco will need.

```
> sudo add-apt-repository ppa:opencpn/opencpn  
> sudo apt-get update
```

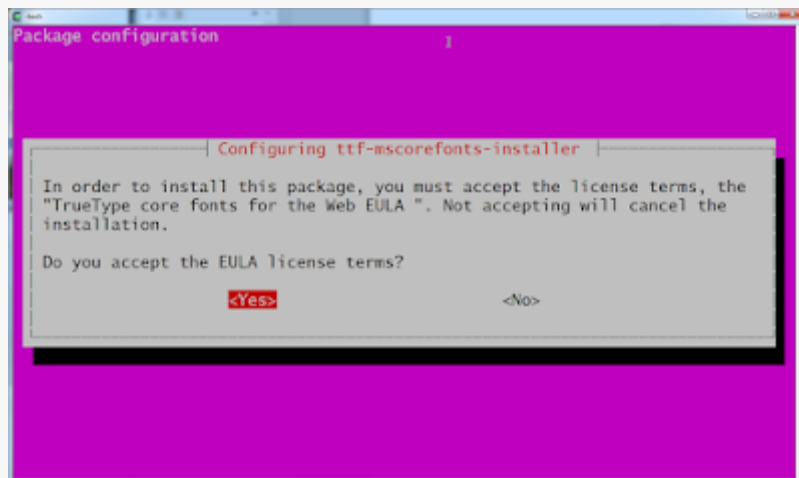
Installare le necessarie librerie

```
sudo apt -y install fontconfig libsm-dev libice-dev libxt-dev libxrender-dev libxext-dev cups libglu1-mesa-dev libcairo2-dev libgl-dev
```

```
sudo apt-get install -y libfontconfig1 libsm6 libice6 \libxrender1 libxt6 libcups2 opencpn libcairo2 \ttf-mscorefonts-installer
```



Have to accept the license for Microsoft fonts.



Chmod and run it

```
> chmod u+x alfresco-community-installer-201707-linux-x64.bin  
> sudo ./alfresco-community-installer-201707-linux-x64.bin
```

You will get this warning ignore it and enter y

```
Language Selection
Please select the installation language
[1] English - English
[2] French - Français
[3] Spanish - Español
[4] Italian - Italiano
[5] German - Deutsch
[6] Japanese - 日本語
[7] Dutch - Nederlands
[8] Russian - 俄罗斯语
[9] Simplified Chinese - 简体中文
[10] Norwegian - Norsk bokmål
[11] Brazilian Portuguese - Português Brasileiro
Please choose an option [1] :
```

Language select 1

```
Installation Type
[1] Easy - Install using the default configuration.
[2] Advanced - Configure server ports and service properties.: Choose optional components to install.
Please choose an option [1] : 2
```

Select Advanced

For the next few choices

Java	- yes
PostgreSQL	- yes
LibreOffice	- yes
Alfresco Community Edition	- yes
Solr1	- no
Solr4	- yes
Alfresco Office Services	- yes
Web Quick Start	- yes
Google Docs Integration	- yes

```
Choose a folder to install Alfresco Community.
Select a folder: [/opt/alfresco-community]: /10x13/apps/alfresco
```

```
root@ubuntu: /home/ubuntu
Attenzione: Impossibile stabilire un'associazione al numero di porta
specificato. Selezionare un'altra porta di arresto di Tomcat.
Premi [ Invio ] per continuare :
-----
Configurazione porte Tomcat

Immettere i parametri di configurazione Tomcat.

Dominio server web: [127.0.0.1]:
Porta server Tomcat: [8080]: 8081
Porta di arresto di Tomcat: [8005]: 8006
Porta SSL di Tomcat: [8443]: 8444
Porta AJP di Tomcat: [8009]: 8010
-----
Porta server LibreOffice

Immettere la porta di cui il server LibreOffice resterà in ascolto.
Porta server LibreOffice: [8100]: █
```

```
-----
Admin Password

Specify a password for the Alfresco Content Services administrator
account.

Admin Password: :
Repeat Password: :
-----
```

For the admin password I put in "admin" I will switch it later.

```
Install as a service

If you register Alfresco Community as a service it will automatical
ly start
Alfresco Community on machine startup.

Install Alfresco Community as a service? [Y/n]: |
```

Enter Y and install

It will take several minutes to get fully deployed
I like to tail the tomcat catalina.log


```
> cd /10x13/apps/alfresco  
> tail -f tomcat/logs/catalina.out
```

Test it

Il mio server usa 192.168.1.61 così' <http://192.168.1.61:8081/share>

192.168.1.61:8081/share/page/

ifica IBAN test2 Opzioni di Boot Ma... WADSL Micso Adsl, Wadsl... Network Tools: DN... TestFabio Efficienza Energetic... S HP, IBM, Cisco, J

 Alfresco

Alfresco Share
Alfresco Community Edition

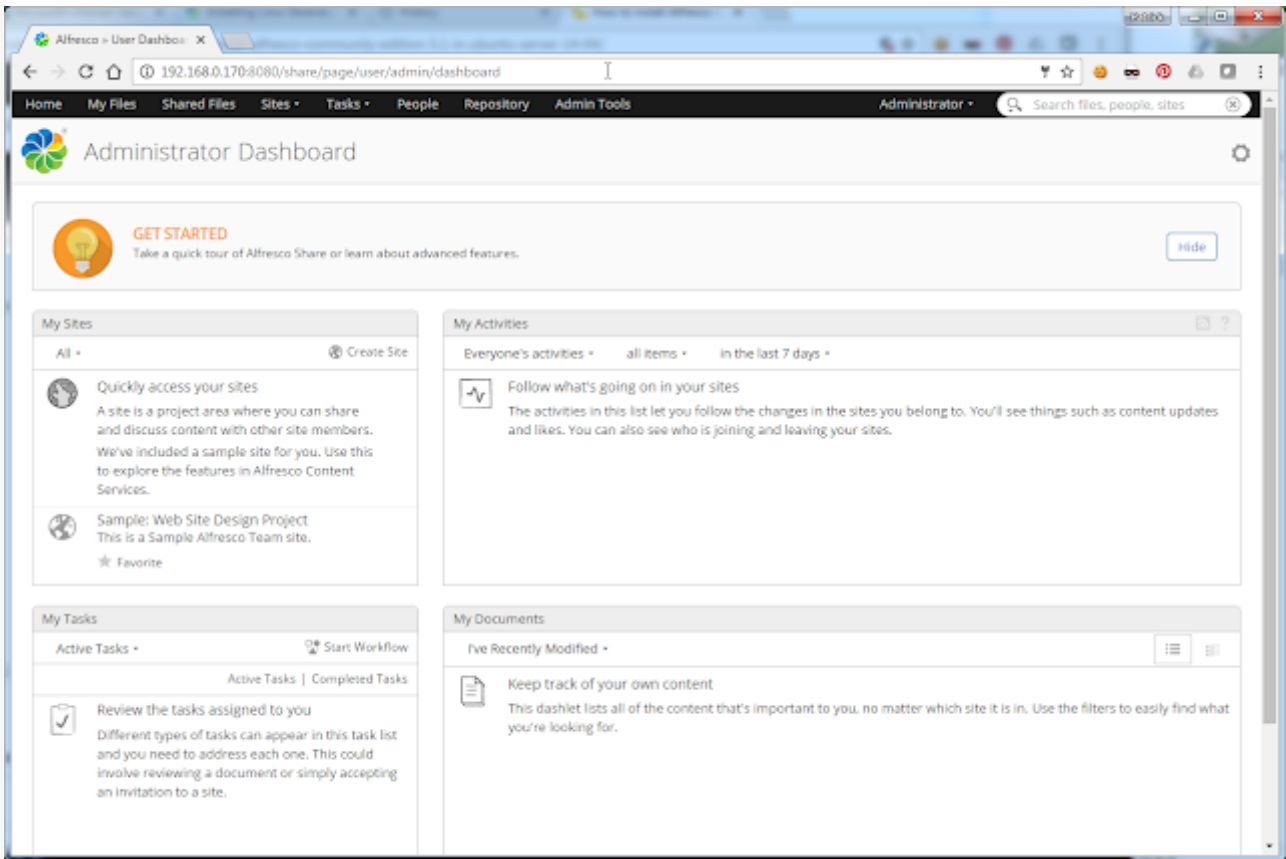
admin

....

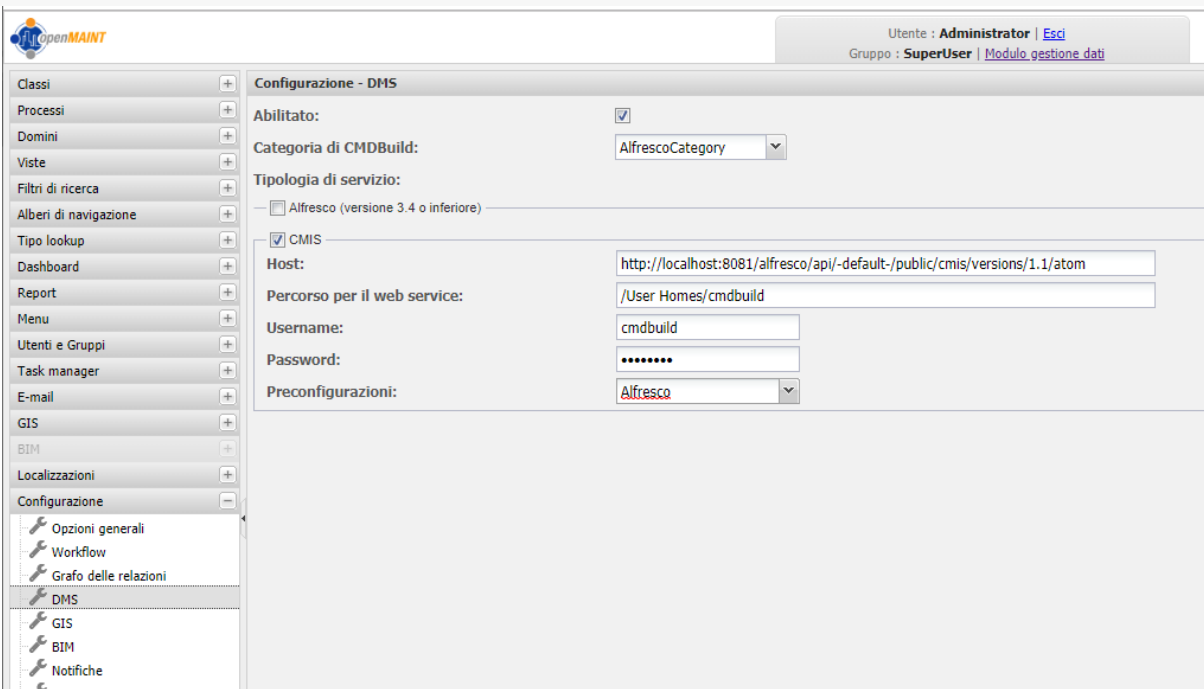
Accedi

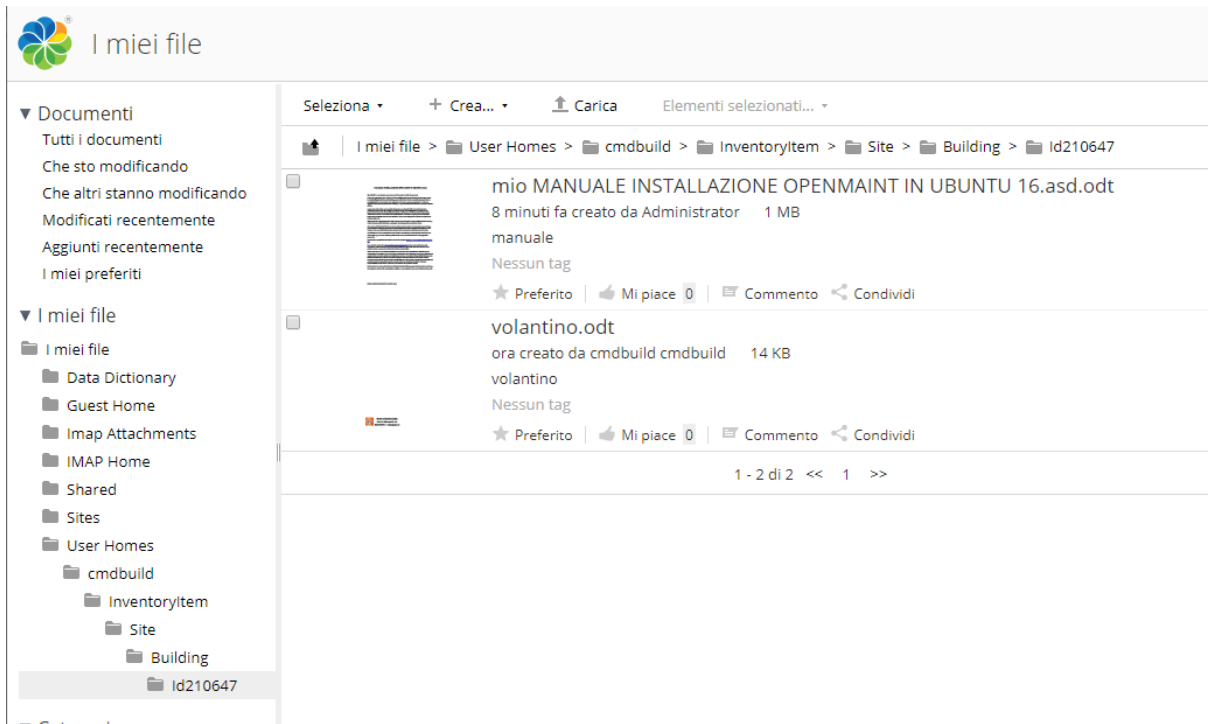
© 2005-2017 Alfresco Software Inc. Tutti i diritti riservati.

Log in con admin:admin



Dopo aver creato l'utente (nel mio caso cmdbuild con pass cmdbuild) e il sito (cmdbuild) è possibile caricare allegati e documenti per poterli poi raccogliere in alfresco, dopo aver configurato il collegamento di openmaint con alfresco





Installazione BIMSERVER

Nel mio caso per testare il funzionamento corretto ho fatto l'installazione con la versione JAR:

wget <https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/bimserver/bimserver-1.2.jar>

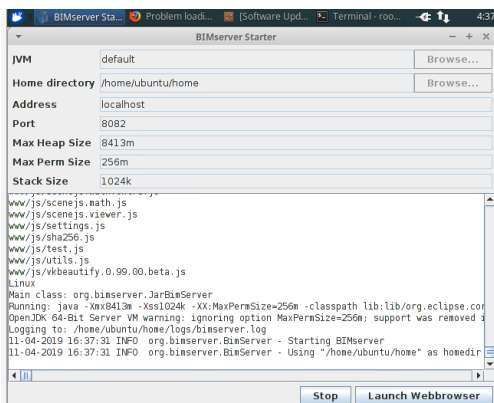
dopodiche' l'ho rinominato in bimserver.jar

```
mv bimserver-1.2.jar bimserver.jar
```

lanciando quindi l'installazione (direttamente sul server)

```
java-jar bimserver.jar
```

L'installazione propone una configurazione di default che ho lasciato invariata



Entrando nella Pagina principale si ha uno Status: NOT_SETUP di Bimserver

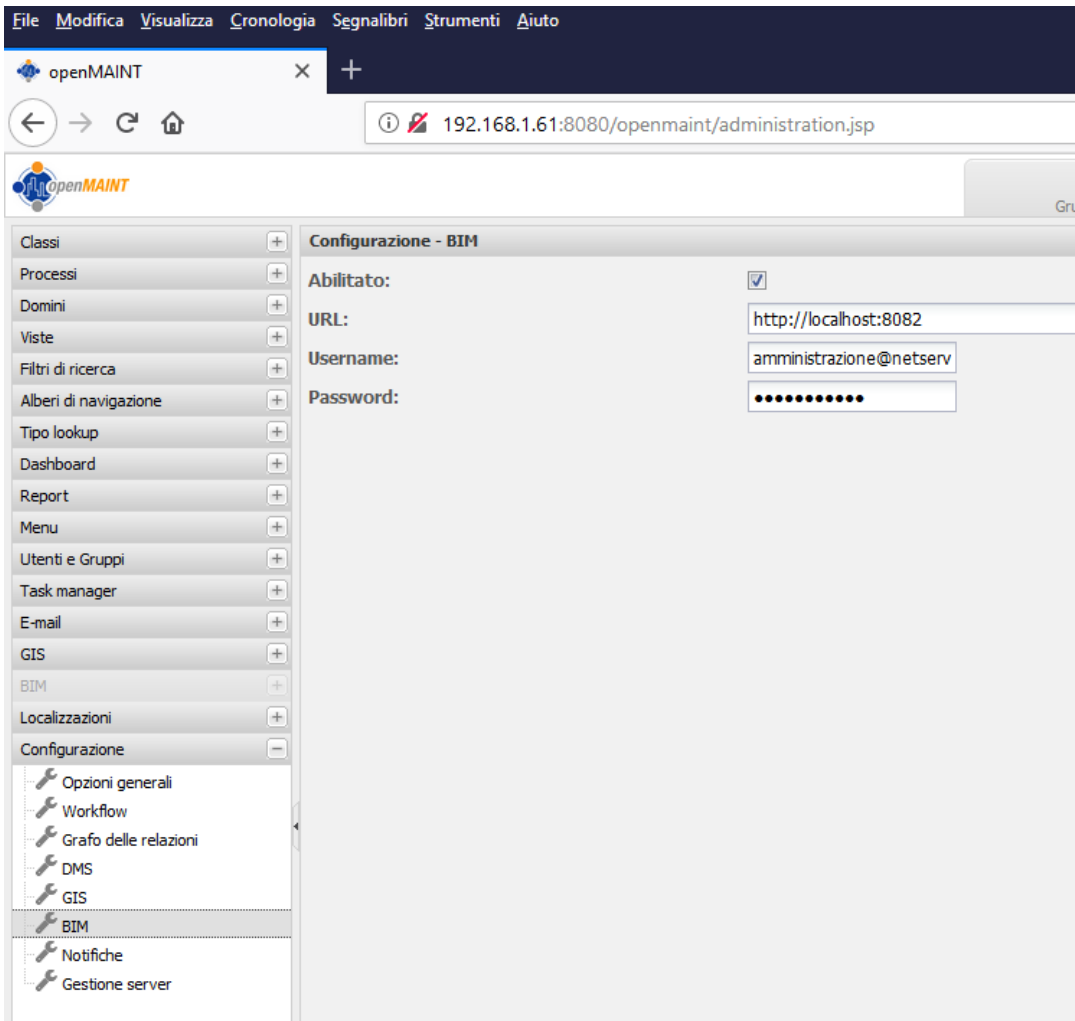
Occorre pertanto entrare nella pagina di admin e inserire i dati necessari

The screenshot shows a web browser window with the address bar displaying "192.168.1.61:8082/admin/". The page content includes the BIMserver logo and the heading "Setup BIMserver". Below the heading, a message states: "This BIMserver has not yet been setup. You can setup this BIMserver right now." The form contains the following fields:

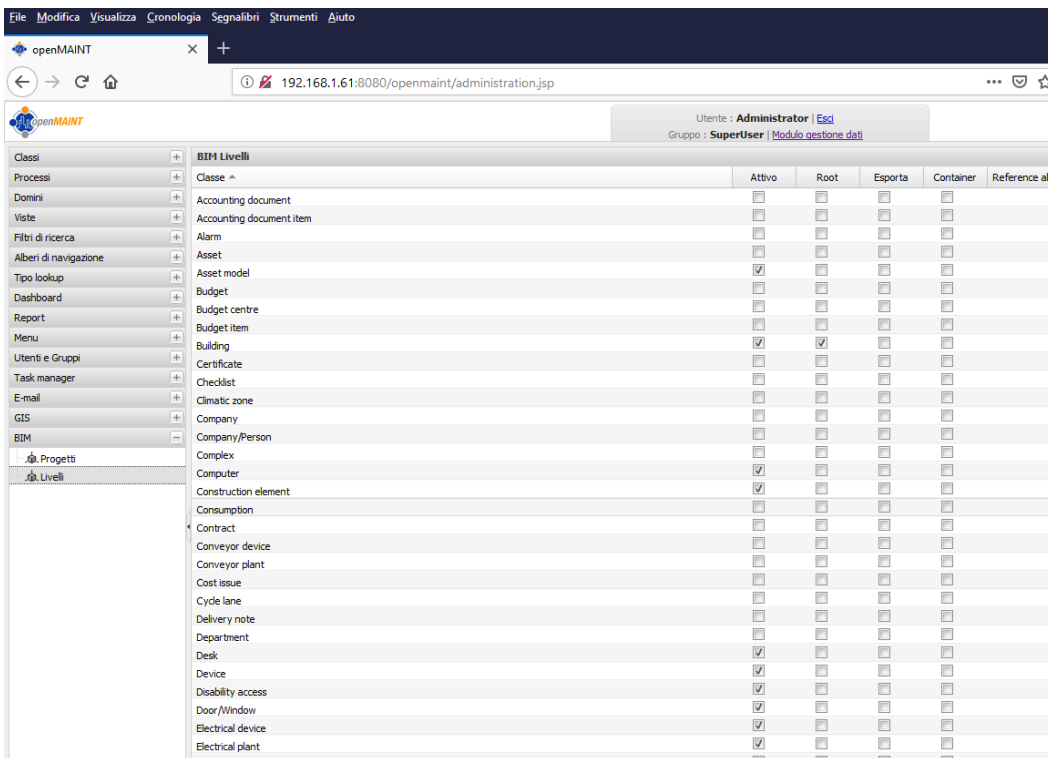
- Site address:** A text input field containing "http:// 192.168.1.61:8082".
- E-mail section:**
 - SMTP Server:** A text input field with the placeholder "Address of the SMTP server to use".
 - Email sender address:** A text input field with a "@" symbol and the placeholder "E-mail adres from which to send e-n".
- Administrator section:**
 - Admin name:** A text input field containing "Administrator".
 - Admin username:** A text input field with a "@" symbol and the placeholder "Username (e-mail address)".
 - Admin password:** A password input field with a lock icon.

A "Setup" button is located at the bottom of the form.

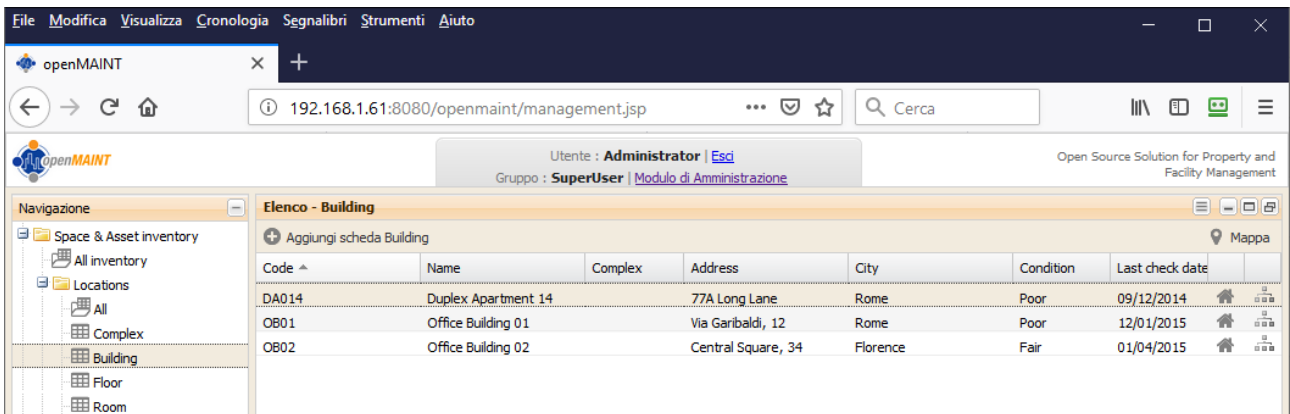
Occorre quindi creare il collegamento al server in openmaint



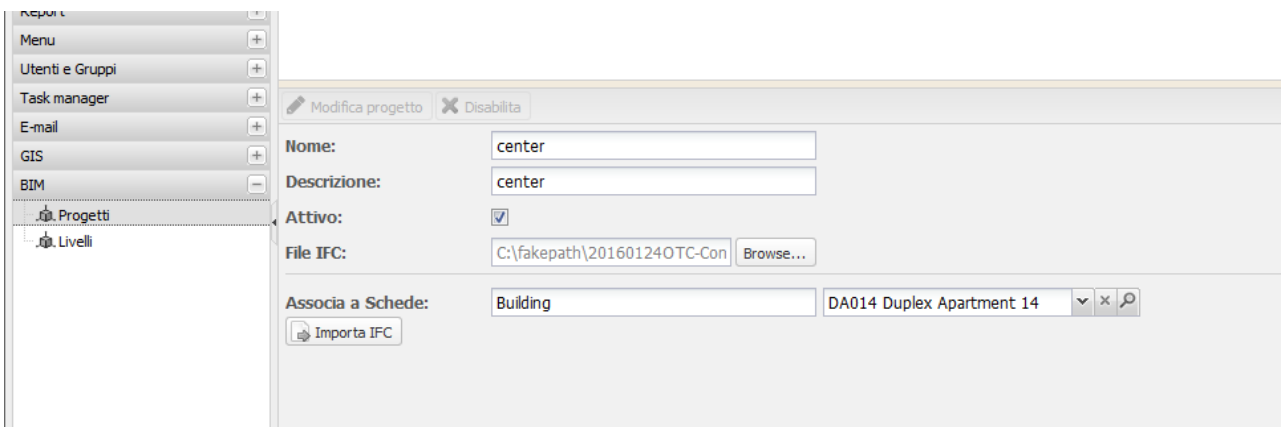
Occorre quindi definire i livelli di BIM (identificando il livello root – nel mio caso gli edifici/buildings):



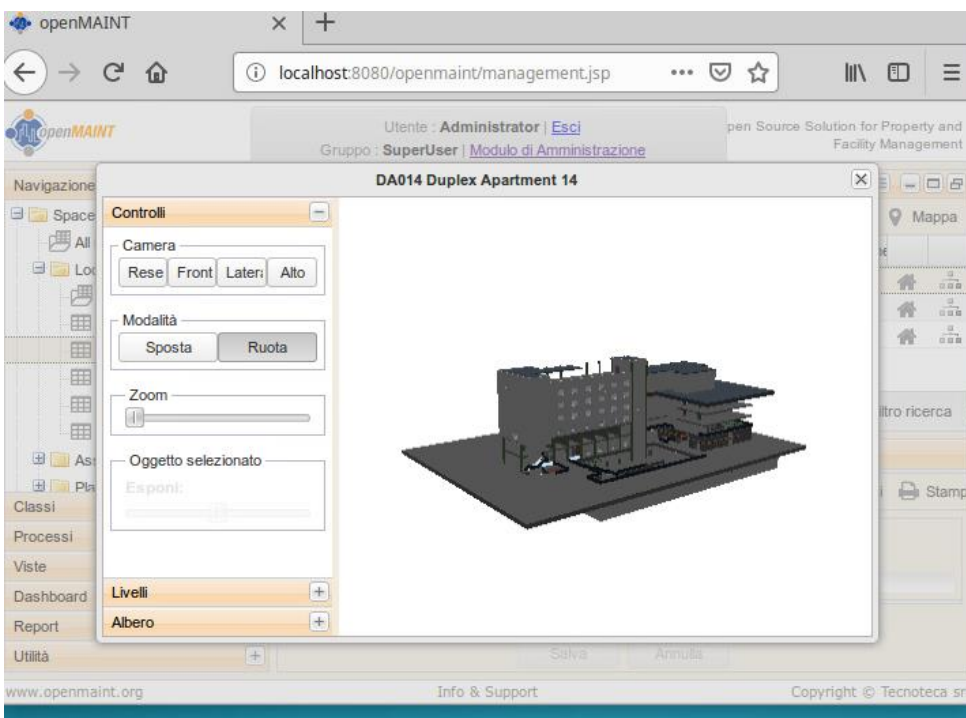
Possiamo quindi iniziare a caricare il nostro progetto BIM sull'edificio di interesse – è opportuno verificare prima che l'iconcina 3d (la casettina a destra prima dei collegamenti) sia presente sull'edificio:



Carichiamo quindi il nostro progetto



Che potremo visualizzare in 3d entrando in ogni caratteristica in modo approfondito



CONCLUSIONI

Esistono ancora possibili collegamenti del programma con software esterno, fra cui OCSinventor (a cui preferisco GLPI con FusionInventory) e (a mio avviso) NAGIOS. Peraltro credo che molte opportunità emergeranno con la prossima revisione del programma prevista in uscita entro quest'anno, e credo che con l'utilizzo di sensori e periferiche IOT le possibili implementazioni saranno molteplici.

Spero con questa documentazione (che come potete vedere ha riassunto alcune soluzioni già presenti in rete) di essere utile a chi vuole approfondire la conoscenza del prodotto OPENMAINT.

Molte Aziende e strutture pubbliche potrebbero poi collegare l'utilizzo di questo prodotto con alcuni programmi di finanziamento per l'ottimizzazione della loro struttura.

Se il Manuale vi è stato utile contribuite alla sua diffusione e visitate il nostro sito web www.pinet.it

Alessandria (Italy) 11/04/2019

Giancarlo Finesso