Installing Dspace 1.8 on Ubuntu 12.04

This is an abridged version of the dspace 1.8 installation guide, specifically targeted at getting a basic server running from scratch using Ubuntu. More information can be found in the full dspace 1.8.x installation guide located here: https://wiki.duraspace.org/display/DSDOC18/Installation

First, prepare an Ubuntu server for Dspace deployment. This document assumes a base Ubuntu 12.04 LTS server install. Make sure that networking is up, with a valid IP and route to the internet.

Install and configure openssh and firewall – Done from console

```
sudo apt-get install openssh-server
```

Enter your sudo password if needed

```
sudo service ssh restart

sudo ufw allow 22

sudo ufw allow 80

sudo ufw allow 443

sudo ufw enable
```

Note that this opens up to the world connections on ssh, http and https. If a stricter policy is needed, it should be configured here.

Also, sshd should be configured to match the site standards, including things like public key vs password login, norootlogon options, etc.

Now SSH into your server. The rest of the config will be run from there.

Installation of prerequisite applications for Dspace

```
sudo apt-get -y install openjdk-6-jdk tasksel
```

Use Tasksel to configure server components

```
sudo tasksel
```

Select the following packages.

Use space bar for select applications from list.

[*] LAMP server

[*] PostgreSQL database

[*] Tomcat Java server

Use tab to select OK button and enter

You will need to select a MySQL root password during this procedure. IThis is a shortcut to get the correct packages installed; we will be configuring Dspace to use PostgreSQL.

Install Ant and Maven

```
sudo apt-get -y install ant maven2
```

Create the database user (dspace)

```
sudo su postgres
```

```
createuser -U postgres -d -A -P dspace
```

Enter password for new role. The default password expected by the dspace server is “dspace”. Note: If you choose anything different (recommended), make sure you change it in the dspace.cfg file in the build folder before you deploy the install with ant.

Shall the new role be allowed to create more new roles? (y/n) *n*

```
exit
```

Allow the database user (dspace) to connect to the database

The version number in the line below may be different than what you are using.

```
sudo vi /etc/postgresql/9.1/main/pg_hba.conf
```

Add this line to the configuration file at the end

```
local all dspace md5
```

Save and close the file

Restart PostgreSQL

```
sudo service postgresql restart
```

Configure local Unix user ‘dspace’:

```
sudo useradd -m dspace
```

```
sudo passwd dspace
```

Enter your chosen password for dspace user – be careful as the dspace user by default is given a shell login, which means that if someone guesses the dspace password, they will be able to log on remotely

Create the /usr/local/dspace directory if it does not yet exist and assign permissions

```
sudo mkdir -p /usr/local/dspace
```

```
sudo chown dspace:dspace /usr/local/dspace
```

Create the PostgreSQL 'dspace' database.

```
sudo -u dspace createdb -U dspace -E UNICODE dspace
```

Configure Tomcat to know about the DSpace webapps.

Make a copy of the original tomcat config (for later upgrade/diff comparisons)

```
sudo cp /etc/tomcat6/server.xml /etc/tomcat6/server.xml.orig
sudo vi /etc/tomcat6/server.xml
```

Insert the following chunk of text just above the closing </Host>

```
<!-- Define a new context path for all DSpace web apps -->

<Context path="/xmlui" docBase="/usr/local/dspace/webapps/xmlui" allowLinking="true"/>

<Context path="/sword" docBase="/usr/local/dspace/webapps/sword" allowLinking="true"/>

<Context path="/oai" docBase="/usr/local/dspace/webapps/oai" allowLinking="true"/>

<Context path="/jspui" docBase="/usr/local/dspace/webapps/jspui" allowLinking="true"/>

<Context path="/lni" docBase="/usr/local/dspace/webapps/lni" allowLinking="true"/>

<Context path="/solr" docBase="/usr/local/dspace/webapps/solr" allowLinking="true"/>
```

Save and close file

Download and unpack the current version of Dspace. If a newer version exists, simply replace the “1.8.2” with the new version number in the following lines. Newer versions, especially full point releases (i.e. 1.9) may need additional changes to these directions. Consult the dspace installation manual online for more information.

```
sudo mkdir /usr/local/src/dspace

sudo chmod -R 777 /usr/local/src/dspace

cd /usr/local/src/dspace

wget http://sourceforge.net/projects/dspace/files/DSpace%20Stable/1.8.2/dspace-1.8.2-src-release.tar.bz2

tar -xvjf dspace-1.8.2-src-release.tar.bz2

cd /usr/local/src/dspace/dspace-1.8.2-src-release
```

Make a backup of dspace.cfg and edit the dspace.dir line to reflect new location of /usr/local/dspace

```
cp dspace/config/dspace.cfg dspace/config/dspace.cfg.orig

vi /usr/local/src/dspace/dspace-1.8.2-src-release/dspace/config/dspace.cfg
```

If you made the change to the default password for the PostgreSQL server, you will want to make that here as well.

If you know what your dspace configuration needs to look like, go through the config directory and make changes so the system will deploy with these changes. Otherwise they can be changed on the running system. At a minimum, you'll want to change the hostname, name, mail server and email contact addresses. See the dspace installation manual for more information.

Save file

```
mvn -U package

cd dspace/target/dspace-1.8.2-build

sudo ant fresh_install
```

Set Tomcat to run as dspace user and restart the Tomcat server

```
sudo vi /etc/default/tomcat6
```

Change TOMCAT_USER and TOMCAT_GROUP from tomcat6 to dspace

```
sudo chown -R root:dspace /etc/tomcat6

sudo chmod 775 /var/cache/tomcat6

sudo chown -R dspace:dspace /var/cache/tomcat6

sudo chown -R root:dspace /var/lib/tomcat6/webapps

sudo chown -R root:dspace /var/log/tomcat6

sudo chown -R dspace:dspace /usr/local/dspace

sudo service tomcat6 stop
```

Check for running tomcat6 service with

```
ps -ef | grep java
```

Look for any java process running as the “tomcat” user. If one exists, end it with

```
sudo kill whateverthepidis

sudo service tomcat6 start
```

Make an initial administrator user account in DSpace:

Switch to the dspace user

```
sudo su – dspace
```

Create the initial Dspace Admin user

```
/usr/local/dspace/bin/dspace create-administrator
```

Enter data for initial Dspace admin. This probably should not be an individual.

```
exit
```

The next section covers configuration of the Apache Java Proxy to allow Dspace to be accessible from port 80/443 and to setup rewrite for our base url.

Configure SSL

Enable SSL Module

```
sudo a2enmod ssl

sudo service apache2 restart
```

Create a Self-Signed SSL certificate. If you have a certificate from a CA, install them here instead. A proper DNS name should be used instead of “server” if you are getting your own certificates.

```
cd /tmp

openssl genrsa -out server.key 2048

openssl req -new -key server.key -out server.csr
```

Here you will have to fill out the CSR as follows. You will be asked several questions here.

```
openssl x509 -req -days 3650 -in server.csr -signkey server.key -out server.crt

sudo cp server.crt /etc/ssl/certs/
sudo cp server.key /etc/ssl/private/
```

Change default-ssl file to enable ssl

```
sudo vi /etc/apache2/sites-available/default-ssl
```

Add following line below Server Admin line:

```
ServerName example.com:443
```

Verify the following lines are correctly configured:

```
SSLEngine on

SSLCertificateFile /etc/ssl/certs/server.crt

SSLCertificateKeyFile /etc/ssl/private/server.key
```

Activate SSL Site

```
sudo a2ensite default-ssl
```

Enable AJP in Tomcat

```
sudo vi /etc/tomcat6/server.xml
```

Remove the “<!- -“ and the “- ->” from line <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />

Remove the “<!- -“ and the “- ->” from line <Connector port="8443" protocol="HTTP/1.1" redirectPort="8443" …../>

Enable proper apache2 modules

```
sudo a2enmod proxy_ajp rewrite

sudo vi /etc/apache2/conf.d/proxy_ajp.conf
```

Add the following lines into the file (it should be empty):

```
#

# When loaded, the mod_proxy_ajp module adds support for

# proxying to an AJP/1.3 backend server (such as Tomcat).

# To proxy to an AJP backend, use the "ajp://" URI scheme;

# Tomcat is configured to listen on port 8009 for AJP requests

# by default.

#

# Uncomment the following lines to serve the ROOT webapp

# under the /tomcat/ location, and the jsp-examples webapp

# under the /examples/ location.

#
```

```
#ProxyPass /tomcat/ ajp://localhost:8009/

#ProxyPass /examples/ ajp://localhost:8009/jsp-examples/


ProxyPass /jspui             ajp://localhost:8009/jspui

ProxyPass /oai               ajp://localhost:8009/oai

ProxyPass /xmlui             ajp://localhost:8009/xmlui

ProxyPass /solr              ajp://localhost:8009/solr
```

Configure apache2 rewrite

```
sudo vi /etc/apache2/httpd.conf
```

Add the following lines into the file: *Note – replace “localhost” with your proper server name

```
RedirectMatch ^/$ https://localhost/xmlui/

RewriteEngine On

RewriteCond %{SERVER_PORT} !443

RewriteRule (/xmlui/.*) https://localhost%{REQUEST_URI}
```

Perform any additional customizations in the current dspace.cfg file

```
sudo vi /usr/local/dspace/config/dspace.cfg
```

The most commonly altered lines to customize your install are:

```
dspace.hostname = localhost

dspace.baseUrl = http://localhost:8080

dspace.name = DSpace at My University

mail.server = smtp.uoregon.edu

mail.from.address = dspace@admin.com

feedback.recipient = dspace-help@myu.edu

mail.admin = dspace-help@myu.edu
```

These changes should all be done once you have all of the pertinent information like email address and DNS names.

Restart Tomact and then Apache2

```
sudo service tomcat6 restart

sleep 20

sudo service apache2 restart
```

That is all that is required to install DSpace on Ubuntu. Test in your browser now at https://yourservername

Cron Jobs – once the configuration is set up, these cron jobs should be set

```
sudo su – dspace

crontab –e
```

Add the following lines. Adjust times to your own specs. Note that the vacuumdb may not be needed with recent versions of postgresql as long as postgresql autovacuum is enabled.

```
# Send out subscription e-mails at 01:00 every day
0 1 * * *  /usr/local/dspace/bin/dspace sub-daily
# Run the media filter at 02:00 every day
0 2 * * *  /usr/local/dspace/bin/dspace filter-media
# Run the checksum checker at 03:00
0 3 * * *  /usr/local/dspace/bin/dspace checker -lp
# Mail the results to the sysadmin at 04:00
0 4 * * *  /usr/local/dspace/bin/dspace checker-emailer -c


# Clean up the database nightly at 4.20am
20 4 * * * vacuumdb --analyze dspace > /dev/null 2>&1


# Run stat analysis
0 1 * * * /usr/local/dspace/bin/dspace stat-general
0 1 * * * /usr/local/dspace/bin/dspace stat-monthly
0 2 * * * /usr/local/dspace/bin/dspace stat-report-general
0 2 * * * /usr/local/dspace/bin/dspace stat-report-monthly
```