**Installing DSpace**

Nason Bimbe

September 2017

**Important!**

Always consult the official DSpace documentation on installation. Please see [https://wiki.duraspace.org/display/DSDOC6x/Installing+DSpace](https://wiki.duraspace.org/display/DSDOC6x/Installing%2BDSpace)

# Introduction

These instructions are based on installing DSpace 6.x on Ubuntu Server 16.04 LTS.

# Prerequisites

DSpace 6.x has the following software prerequisites for all platforms:

* Java JDK v7
* Apache Maven 3.0.5 or later (Java build tool)
	+ Maven is necessary in the first stage of the build process to assemble the installation package for your DSpace instance. It gives you the flexibility to customize DSpace using the existing Maven projects found in the [dspace-source]/dspace/modules directory or by adding in your own Maven project to build the installation package for DSpace, and apply any custom interface "overlay" changes.
* Apache Ant 1.8 or later
	+ Apache Ant is required for the second stage of the build process. It is used once the installation package has been constructed in [dspace-source]/dspace/target/dspace-installer and still uses some of the familiar ant build targets found in the 1.4.x build process.
* PostgreSQL 9.0 or later
* Apache Tomcat 7 or later
* Perl 5
	+ Only required for [dspace]/bin/dspace-info.pl

# Installing Ubuntu Server 16.04 LTS

Once the installation is complete, please test to make sure the setup is working properly and also that the server can be accessed from a client computer using SSH.

Login using your SSH credentials

## Create user

DSpace and Tomcat will need to run as the same user so make sure that you created that user that DSpace and Tomcat will run under.

# Installing Prerequisites

## Update the server

First, ensure that the system is up to date:

sudo apt-get update

sudo apt-get upgrade

## Java JDK

Install Java dependencies. Since the Java is already installed, we still need to make sure that the default jdk aligns with the Java version we need, so do:

sudo apt-get install default-jdk default-jre

List alternatives:

sudo update-java-alternatives -l

You should see something like this:

java-1.8.0-openjdk-amd64 1081 /usr/lib/jvm/java-1.8.0-openjdk-amd64

Set JAVA\_HOME environment variable:

sudo nano /etc/environment

And add the following at the top:

export JAVA\_HOME=/usr/lib/jvm/default-java

And at the end of PATH string:

:$JAVA\_HOME/jre/bin

And should look like this after the processes

export JAVA\_HOME=/usr/lib/jvm/default-java

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:$JAVA\_HOME/jre/bin"

Run the following to reload the environment, notice the space after the fullstop:

. /etc/environment

Check if java installed okay

echo $JAVA\_HOME

This will display the $JAVA\_HOME path on the console

java -version

This will display the Java installed and its version number on the console

## Ant and Maven

Install Ant and Maven:

sudo apt-get install ant maven

## Configure Maven

You will need to create an environment variable for the location of maven.

sudo nano /etc/environment

And add the following:

export M3\_HOME=/usr/share/maven

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:$JAVA\_HOME/jre/bin:$M3\_HOME/bin"

Create a symbolic link:

sudo ls -s /usr/share/maven/bin/mvn /usr/bin/mvn

Run the following to reload the environment:

. /etc/environment

Check if maven installed okay

echo $M3\_HOME

mvn -version

***HTTP Proxy***

If you need to run maven through a HTTP proxy you will need to configure this in the central settings.xml file. This should be live under the Maven installation directory:

cd $M3\_HOME/conf

sudo nano settings.xml

Uncomment the proxies section and add your details as illustrated in the highlighted section:

<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"

 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

 xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0

 http://maven.apache.org/xsd/settings-1.0.0.xsd">

 <localRepository/>

 <interactiveMode/>

 <usePluginRegistry/>

 <offline/>

 <pluginGroups/>

 <servers/>

 <mirrors/>

 <proxies>

 <proxy>

 <id>{my-name-for-maven-settings}</id>

 <active>true</active>

 <protocol>http</protocol>

 <host>{my-proxy-hostname}</host>

 <port>{my-proxy-port}</port>

 <username>{my-proxy-username}</username>

 <password>{my-proxy-password}</password>

 <nonProxyHosts></nonProxyHosts>

 </proxy>

 </proxies>

 <profiles/>

 <activeProfiles/>

</settings>

The username and password settings are only required if your proxy server requires authentication.

You can test the Maven installation and network connectivity by running:

mvn clean

This will give that message BUILD FAILURE because it requires a pom.xml file but the output will indicate whether maven is able to download packages from its repositories.

## Configure Ant

You will need to create an environment variable for the location of ant:

sudo nano /etc/environment

And add the following:

export ANT\_HOME=/usr/share/ant

PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:$JAVA\_HOME/jre/bin:$M3\_HOME/bin:$ANT\_HOME/bin"

Create a symbolic link

sudo ls -s /usr/share/ant/bin/ant /usr/bin/ant

Run the following to reload the environment

. /etc/environment

Check if ant installed okay:

echo $ANT\_HOME

ant -version

***HTTP Proxy***

Part of the DSpace installation needs to download files form web. If you are accessing the internet through a proxy you may need to tell ant the proxy details. Edit /etc/environment

and add the following lines, filling the details of your HTTP proxy:

export ANT\_OPTS="-Dhttp.proxyHost=<proxy address> -Dhttp.proxyPort=<proxy port>"

- Not that the proxy address should contain only the domain name (or IP address) and not include 'http://'

## Tomcat

DSpace will run on other servlet engines but for our installation, we will use Apache Tomcat.

Install Tomcat 8: It has been reported that DSpace does to work well with a Tomcat 8.0.32 which is comes with Ubuntu 16.04 LTS therefore a version higher must be installed manually instead.

The Tomcat 8 installation instructions described below are based on

<https://www.digitalocean.com/community/tutorials/how-to-install-apache-tomcat-8-on-ubuntu-16-04>

First, create a new tomcat group:

sudo groupadd tomcat

Add dspace user to the tomcat group:

sudo adduser dspace tomcat

The best way to install Tomcat 8 is to download the latest binary release then configure it manually.

Find the latest version of Tomcat 8 at the [Tomcat 8 Downloads page](http://tomcat.apache.org/download-80.cgi). At the time of writing, the latest version is **8.5.20**, but you should use a later stable version if it is available. Under the **Binary Distributions** section, then under the **Core** list, copy the link to the "tar.gz".

Next, change to the /tmp directory on your server. This is a good directory to download ephemeral items, like the Tomcat tar ball, which we won't need after extracting the Tomcat contents:

cd /tmp

Use curl to download the link that you copied from the Tomcat website:

curl -O http://www-us.apache.org/dist/tomcat/tomcat-8/v8.5.23/bin/apache-tomcat-8.5.23.tar.gz

Install Tomcat to the /opt/tomcat directory. Create the directory, then extract the archive to it with these commands:

sudo mkdir /opt/tomcat

sudo tar xzvf apache-tomcat-8\*tar.gz -C /opt/tomcat --strip-components=1

Next, set up the proper user permissions for the installation.

Change to the directory where we unpacked the Tomcat installation:

cd /opt/tomcat

Give the tomcat group ownership over the entire installation directory:

sudo chgrp -R tomcat /opt/tomcat

Next, give the tomcat group read access to the conf directory and all its contents, and **execute** access to the directory itself:

sudo chmod -R g+r conf

sudo chmod g+x conf

Make the tomcat user (which in our case is dspace user) the owner of the webapps, work, temp, and logs directories:

sudo chown -R dspace webapps/ work/ temp/ logs/

We want to be able to run Tomcat as a service, so we will set up systemd service file.

Tomcat needs to know where Java is installed. This path is commonly referred to as "JAVA\_HOME". The easiest way to look up that location is by running this command:

sudo update-java-alternatives -l

Output

java-1.8.0-openjdk-amd64 1081 /usr/lib/jvm/java-1.8.0-openjdk-amd64

The correct JAVA\_HOME variable can be constructed by taking the output from the last column and appending /jre to the end. Given the example above, the correct JAVA\_HOME for this server would be:

JAVA\_HOME

/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre

Your JAVA\_HOME may be different.

With this piece of information, we can create the systemd service file. Open a file called tomcat.service in the /etc/systemd/system directory by typing:

sudo nano /etc/systemd/system/tomcat.service

Paste the following contents into your service file. Modify the value of JAVA\_HOME if necessary to match the value you found on your system. Also change the tomcat user to be dspace user.

[Unit]

Description=Apache Tomcat Web Application Container

After=network.target

[Service]

Type=forking

Environment=JAVA\_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64/jre

Environment=CATALINA\_PID=/opt/tomcat/temp/tomcat.pid

Environment=CATALINA\_HOME=/opt/tomcat

Environment=CATALINA\_BASE=/opt/tomcat

Environment='CATALINA\_OPTS=-Xms1024M -Xmx2048M -server -XX:+UseParallelGC'

Environment='JAVA\_OPTS=-Djava.awt.headless=true -Djava.security.egd=file:/dev/./urandom -Dfile.encoding=UTF-8'

ExecStart=/opt/tomcat/bin/startup.sh

ExecStop=/opt/tomcat/bin/shutdown.sh

User=dspace

Group=tomcat

UMask=0007

RestartSec=10

Restart=always

[Install]

WantedBy=multi-user.target

When you are finished, save and close the file.

Next, reload the systemd daemon so that it knows about our service file:

sudo systemctl daemon-reload

Start the Tomcat service by typing:

sudo systemctl start tomcat

Double check that it started without errors by typing:

sudo systemctl status tomcat

= [End of Tomcat 8 installation] =

~~Set some Java options to ensure Tomcat is compatible with DSpace:~~

~~sudo nano /etc/environment~~

~~And add this to the bottom:~~

~~JAVA\_OPTS="-Djava.awt.headless=true –Xmx4096m –Xms1024m -Dfile.encoding=UTF-8"~~

~~Run the following to reload the environment:~~

~~. /etc/environment~~

~~Also tell the Tomcat connector to support UTF-8 encoding in URIs. Edit server.xml and make the changes as shown:~~

sudo nano /opt/tomcat/conf/server.xml

Change as follows:

<Connector port="8080" protocol="HTTP/1.1"

 enableLookups="false"

 maxThreads="150"

 minSpareThreads="25"

 maxSpareThreads="75"

 acceptCount="100"

 disableUploadTimeout="true"

 connectionTimeout="20000"

 URIEncoding="UTF-8"

 redirectPort="8443" />

Restart tomcat:

sudo systemctl restart tomcat

Let tomcat start on server start/reboot:

sudo systemctl enable tomcat

Setup Tomcat admin users by typing the following:

sudo nano /opt/tomcat/conf/tomcat-users.xml

Delete all the contents of the file and add the following admin and manager roles with a password.

<?xml version='1.0' encoding='utf-8'?>

<tomcat-users>

 <role rolename="manager-gui"/>

 <role rolename="manager-jmx"/>

 <role rolename="admin-gui"/>

 <user username="{username-here}" password="{password-here}" roles="manager-gui,manager-jmx,admin-gui"/>

</tomcat-users>

Please note the username and password set.

Restart tomcat:

sudo systemctl restart tomcat

## Perl

Perl should already be installed on your Ubuntu server installation

# Setting up DSpace

## Download DSpace software

DSpace can be downloaded from GitHub (https://github.com/DSpace/DSpace/releases). We will be using the source release, so copy the distribution file to the installation directory /home, for example and unpack it:

cd $home

wget https://github.com/DSpace/DSpace/releases/download/dspace-6.2/dspace-6.2-src-release.tar.gz

tar -zxf dspace-6.2-src-release.tar.gz

This will create a directory called dspace-<version>-src-release where <version> is the number of the DSpace version downloaded. Following the convention of the DSpace manual, we will refer to this directory as [dspace-source] in the reminder of this document.

Change the ownership of this directory to the dspace user:

sudo chown -R dspace:dspace dspace-6.2-src-release

## Install/Configure PostgreSQL and Setup DSpace database in PostgreSQL

**Install PostgrSQL**

Create a file at /etc/apt/sources.list.d/postgresql.list with the following command:

sudo nano /etc/apt/sources.list.d/postgresql.list

Add the following line according to your distribution (Ubuntu 16.04 codename is trusty):

deb http://apt.postgresql.org/pub/repos/apt/ xenial-pgdg main

Save it and Close it! Now you just have to do:

sudo wget http://apt.postgresql.org/pub/repos/apt/ACCC4CF8.asc

sudo apt-key add ACCC4CF8.asc

sudo apt-get update

sudo apt-get upgrade

sudo apt-get install postgresql postgresql-contrib

sudo apt-get update

sudo apt-get upgrade

The PostgreSQL database will need to be configured for use by DSpace.

A PostgreSQL JDBC driver is configured as part of the default DSpace build. You no longer need to copy any PostgreSQL jars to get PostgreSQL installed.

Create a dspace database user (this user can have any name, but we'll assume you name them "dspace"). This is entirely separate from the dspace operating-system user created above:

sudo su - postgres

createuser -U postgres -d -A -P dspace

At this point, you will be asked to enter a password. Make it secure and remember it.

Create the dspace database

createdb --username=postgres --owner=dspace --encoding=UNICODE dspace

Enable the pgcrypto extension on your new dspace database. Again, this can only be enabled by a "superuser" account (e.g. postgres)

# Login to the database as a superuser, and enable the pgcrypto extension on this database

psql --username=postgres dspace -c "CREATE EXTENSION pgcrypto;"

exit

## Configure DSpace (Initial configuration)

**Initial Configuration (local.cfg):** Create your own [dspace-source]/dspace/config/local.cfg configuration file (you may wish to simply copy the provided [dspace-source]/dspace/config/local.cfg.EXAMPLE). This local.cfg file can be used to store any configuration changes that you wish to make which are local to your installation (see [local.cfg configuration file](https://wiki.duraspace.org/display/DSDOC6x/Configuration%2BReference%22%20%5Cl%20%22ConfigurationReference-Thelocal.cfgConfigurationPropertiesFile) documentation). ANY setting may be copied into this local.cfg file from the dspace.cfg or any other \*.cfg file to override the default setting (see note below).  For the initial installation of DSpace, there are some key settings you'll likely want to override, those are provided in the [dspace-source]/dspace/config/local.cfg.EXAMPLE. (NOTE: Settings followed with an asterisk (\*) are highly recommended, while all others are optional during initial installation and may be customized at a later time)

cp [dspace-source]/dspace/config/local.cfg.EXAMPLE [dspace-source]/dspace/config/local.cfg

nano [dspace-source]/dspace/config/local.cfg

* dspace.dir\* - must be set to the [dspace] (installation) directory  (NOTE: On Windows be sure to use forward slashes for the directory path!  For example: "C:/dspace" is a valid path for Windows.)
* dspace.hostname - fully-qualified domain name of web server (or "localhost" if you just want to run DSpace locally for now)
* dspace.baseUrl\* - complete URL of this server's DSpace home page (including port), but without any context eg. /xmlui, /oai, etc.
* dspace.name - "Proper" name of your server, e.g. "My Digital Library".
* solr.server\* - complete URL of the Solr server. DSpace makes use of [Solr](http://lucene.apache.org/solr/) for indexing purposes.
* default.language - Default language for all metadata values (defaults to "en\_US")
* db.url\* - The full JDBC URL to your database (examples are provided in the local.cfg.EXAMPLE)
* db.driver\* - Which database driver to use, based on whether you are using PostgreSQL or Oracle
* db.dialect\* - Which database dialect to use, based on whether you are using PostgreSQL or Oracle
* db.username\* - the database username used in the previous step.
* db.password\* - the database password used in the previous step.
* db.schema\* - the database scheme to use (examples are provided in the local.cfg.EXAMPLE)
* mail.server - fully-qualified domain name of your outgoing mail server.
* mail.from.address - the "From:" address to put on email sent by DSpace.
* mail.feedback.recipient - mailbox for feedback mail.
* mail.admin - mailbox for DSpace site administrator.
* mail.alert.recipient - mailbox for server errors/alerts (not essential but very useful!)
* mail.registration.notify- mailbox for emails when new users register (optional)

Next create the DSpace installation directory. This should be the directory specified in dspace.dir configuration option above and should be owned by the dspace user. Assuming a dspace directory of /local/usr/dspace:

cd $home

sudo mkdir dspace

sudo chown -R dspace:dspace dspace

Again, following the convention of the DSpace manual we will refer to this installation directory as [dspace]

## Build and install DSpace

We will build DSpace with the responsive user interface Mirage2.

**Enable Mirage2 responsive UI**

Install git if not yet installed

sudo apt-get install git

Just in case force git to use port 443:

git config –-global url.”https://”.insteadOf git://

Modify xmlui.xconf in [dspace-source]/dspace/config with the following: -

<!-- Mirage theme, @mire contributed theme, default since DSpace 3.0 -->

<!--<theme name="Atmire Mirage Theme" regex=".\*" path="Mirage/" />-->

<!-- Mirage2 theme, @mire contributed theme -->

<theme name="Atmire Mirage2 Theme" regex=".\*" path="Mirage2/" />

Run the Maven package for DSpace as the dspace user:

cd [dspace-source]

mvn package -Dmirage2.on=true

This may download a large number of additional packages in order to generate the Dspace installation.

Next, run the Ant build (still logged in as the dspace user):

cd [dspace-source]/dspace/target/dspace-installer

ant fresh\_install

For an update, please use instead

ant update

## Configure Tomcat to use DSpace webapps

Decide which DSpace Web Applications you want to install. DSpace comes with a variety of web applications (in [dspace]/webapps), each of which provides a different "interface" to your DSpace. Which ones you install is up to you, but there are a few that DSpace manual highly recommend (see below):

* *"xmlui"* = This is the XML-based User Interface (XMLUI), based on Apache Cocoon. It comes with a variety of out-of-the-box themes, including Mirage 1 (the default) and Mirage 2 (based on Bootstrap). Between the "xmlui" and "jspui", you likely only need to choose one.
* *"jspui"* = This is the JSP-based User Interface (JSPUI), which is based on Bootstrap. Between the "xmlui" and "jspui", you likely only need to choose one.
* *"solr" (required)* = This is Apache Solr web application, which is used by the "xmlui" and "jspui" (for search & browse functionality), as well as the OAI-PMH interface. It must be installed in support of either UI.
* *"oai"* = This is the DSpace OAI interface. It allows for Metadata and Bitstream (content-file) harvesting, supporting OAI-PMH (Protocol for Metadata Harvest) and OAI-ORE (Object Reuse and Exchange) protocols
* *"rdf"* ( new ) = This is the DSpace RDF interface supporting Linked (Open) Data.
* *"rest"* = This is the DSpace REST API
* *"sword"* = This is the DSpace SWORDv1 interface. More info on SWORD protocol and its usage.
* *"swordv2"* = This is the DSpace SWORDv2 interface. More info on SWORD protocol and its usage.
* *"lni"* (deprecated) = This is the DSpace Lightweight Networking Interface (LNI), supporting WebDAV / SOAP / RPC API. It is disabled by default as we recommend using REST or SWORD for most activities. In order to build it you must rebuild DSpace with the following flag: mvn package -Pdspace-lni

Then tell your Tomcat installation where to find your DSpace web application(s):

sudo nano /opt/tomcat/conf/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="/home/dspace/dspace/webapps/xmlui" reloadable="true" cachingAllowed="false"/>

<!-- jspui webapp disabled as it is not needed. We will use xmlui for the UI -->

<!--<Context path="/jspui" docBase="/home/dspace/dspace/webapps/jspui" reloadable="true" cachingAllowed="false"/>-->

<Context path="/sword" docBase="/home/dspace/dspace/webapps/sword" reloadable="true" cachingAllowed="false"/>

<Context path="/swordv2" docBase="/home/dspace/dspace/webapps/swordv2" reloadable="true" cachingAllowed="false"/>

<Context path="/oai" docBase="/home/dspace/dspace/webapps/oai" reloadable="true" cachingAllowed="false"/>

<Context path="/solr" docBase="/home/dspace/dspace/webapps/solr" reloadable="true" cachingAllowed="false"/>

<Context path="/rest" docBase="/home/dspace/dspace/webapps/rest" reloadable="true" cachingAllowed="false"/>

<Context path="/rdf" docBase="/home/dspace/dspace/webapps/rdf" reloadable="true" cachingAllowed="false"/>

restart tomcat:

sudo systemctl restart tomcat

## Initialise the DSpace database (optional)

- While this step is optional (as the DSpace database will auto-initialize itself on first startup), it's always good to verify one last time that your database connection is working properly. To initialize the database run (logged in as dspace user) but first stop Tomcat:

[dspace]/bin/dspace database migrate

## Administrator Account

Create an initial administrator account from the command line (logged in as dsapce user) but first stop Tomcat:

[dspace]/bin/dspace create-administrator

## Initial Startup!

Now the moment of truth! Start up (or restart) Tomcat/Jetty/Resin. Visit the base URL(s) of your server, depending on which DSpace web applications you want to use. You should see the DSpace home page. Congratulations! Base URLs of DSpace Web Applications:

* ~~JSP User Interface - (e.g.) http://dspace.myu.edu:8080/jspui~~
* XML User Interface (aka. Manakin) - (e.g.) http://dspace.myu.edu:8080/xmlui
* OAI-PMH Interface - (e.g.) http://dspace.myu.edu:8080/oai/request?verb=Identify (Should return an XML-based response)

In order to set up some communities and collections, you'll need to login as your DSpace Administrator (which you created with create-administrator above) and access the administration UI in XML user interface as this is the one we will be using instead of JSP based one.

# Advanced Installation

The above installation steps are sufficient to set up a test server to play around with, but there are a few other steps and options you should probably consider before deploying a DSpace production site.

## Cron Jobs

A number of cron jobs are required in order to carry out scheduled tasks such as sending subscription emails; generating thumbnails images of media and generating the full-text of documents for indexing; testing the bitstreams in the repository for corruption; enhancing the ability of major search engines to index the content and make it findable; search and browse; OAI-PMH; Usage statistics; and generally for regular optimisation. First create a set of cron jobs as the dspace user.

crontab -e

Enter the following lines:

#-----------------

# GLOBAL VARIABLES

#-----------------

# Full path of your local DSpace Installation (e.g. /home/dspace or /dspace or similar)

# MAKE SURE TO CHANGE THIS VALUE!!!

DSPACE = /home/dspace/dspace

#--------------

# HOURLY TASKS (Recommended to be run multiple times per day, if possible)

# At a minimum these tasks should be run daily.

#--------------

# Regenerate DSpace Sitemaps every 8 hours (12AM, 8AM, 4PM).

# SiteMaps ensure that your content is more findable in Google, Google Scholar, and other major search engines.

0 0,8,16 \* \* \* $DSPACE/bin/dspace generate-sitemaps > /dev/null

#----------------

# DAILY TASKS

# (Recommended to be run once per day. Feel free to tweak the scheduled times below.)

#----------------

# Update the OAI-PMH index with the newest content (and re-optimize that index) at midnight every day

# NOTE: ONLY NECESSARY IF YOU ARE RUNNING OAI-PMH

# (This ensures new content is available via OAI-PMH and ensures the OAI-PMH index is optimized for better performance)

0 0 \* \* \* $DSPACE/bin/dspace oai import -o > /dev/null

# Clean and Update the Discovery indexes at midnight every day

# (This ensures that any deleted documents are cleaned from the Discovery search/browse index)

0 0 \* \* \* $DSPACE/bin/dspace index-discovery > /dev/null

# Re-Optimize the Discovery indexes at 12:30 every day

# (This ensures that the Discovery Solr Index is re-optimized for better performance)

30 0 \* \* \* $DSPACE/bin/dspace index-discovery -o > /dev/null

# run the index-authority script once a day at 12:45 to ensure the Solr Authority cache is up to date

45 0 \* \* \* $DSPACE/bin/dspace index-authority > /dev/null

 # Cleanup Web Spiders from DSpace Statistics Solr Index at 01:00 every day

# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS

# (This removes any known web spiders from your usage statistics)

0 1 \* \* \* $DSPACE/bin/dspace stats-util -i

# Re-Optimize DSpace Statistics Solr Index at 01:30 every day

# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS

# (This ensures that the Statistics Solr Index is re-optimized for better performance)

30 1 \* \* \* $DSPACE/bin/dspace stats-util -o

# Send out subscription e-mails at 02:00 every day

# (This sends an email to any users who have "subscribed" to a Collection, notifying them of newly added content.)

0 2 \* \* \* $DSPACE/bin/dspace sub-daily

# Run the media filter at 03:00 every day.

# (This task ensures that thumbnails are generated for newly add images,

# and also ensures full text search is available for newly added PDF/Word/PPT/HTML documents)

0 3 \* \* \* $DSPACE/bin/dspace filter-media

# Run any Curation Tasks queued from the Admin UI at 04:00 every day

# (Ensures that any curation task that an administrator "queued" from the Admin UI is executed

# asynchronously behind the scenes)

0 4 \* \* \* $DSPACE/bin/dspace curate -q admin\_ui

#----------------

# WEEKLY TASKS

# (Recommended to be run once per week, but can be run more or less frequently, based on your local needs/policies)

#----------------

# Run the checksum checker at 04:00 every Sunday

# By default it runs through every file (-l) and also prunes old results (-p)

# (This re-verifies the checksums of all files stored in DSpace. If any files have been changed/corrupted, checksums will differ.)

0 4 \* \* \* $DSPACE/bin/dspace checker -l -p

# NOTE: LARGER SITES MAY WISH TO USE DIFFERENT OPTIONS. The above "-l" option tells DSpace to check \*everything\*.

# If your site is very large, you may need to only check a portion of your content per week. The below commented-out task

# would instead check all the content it can within \*one hour\*. The next week it would start again where it left off.

#0 4 \* \* 0 $DSPACE/bin/dspace checker -d 1h -p

# Mail the results of the checksum checker (see above) to the configured "mail.admin" at 05:00 every Sunday.

# (This ensures the system administrator is notified whether any checksums were found to be different.)

0 5 \* \* 0 $DSPACE/bin/dspace checker-emailer

#----------------

# MONTHLY TASKS

# (Recommended to be run once per month, but can be run more or less frequently, based on your local needs/policies)

#----------------

# Permane ntly delete any bitstreams flagged as "deleted" in DSpace, on the first of every month at 01:00

# (This ensures that any files which were deleted from DSpace are actually removed from your local filesystem.

# By default they are just marked as deleted, but are not removed from the filesystem.)

0 1 1 \* \* $DSPACE/bin/dspace cleanup > /dev/null

#----------------

# YEARLY TASKS (Recommended to be run once per year)

#----------------

# At 2:00AM every January 1, "shard" the DSpace Statistics Solr index.

# This ensures each year has its own Solr index, which improves performance.

# NOTE: ONLY NECESSARY IF YOU ARE RUNNING SOLR STATISTICS

# NOTE: This is scheduled here for 2:00AM so that it happens \*after\* the daily cleaning & re-optimization of this index.

0 2 1 1 \* $DSPACE/bin/dspace stats-util -s

Log in as the postgres user and create another crontab to clean the database regularly:

sudo su - postgres

crontab -e

Add these lines (for example):

#Clean up the database nightly at 4.20am

20 4 \* \* \* vaxuumdb --analyze dspace > /dev/null 2>&1

## Proxy DSpace/Tomcat through Apache (running on port 80)

Sometimes we don't want to have to type the port number in the URL in order to access the repository so the next step is to set up proxying through Apache using mod\_proxy and mod\_proxy\_ajp Apache modules which will redirect requests to Tomcat (via AJP Connector).

Install the apache web server:

sudo apt-get install apache2

https://help.ubuntu.com/lts/serverguide/httpd.html

Enable AJP Connector

sudo nano /opt/tomcat/conf/server.xml

Uncomment the AJP 1.3 Connector

<!-- Define an AJP 1.3 Connector on port 8009 -->

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

Restart Tomcat

sudo service tomcat restart

Install mod\_proxy and mod\_proxy\_ajp

Please note that these are installed by default when installing Apache HTTP Server on Debian and Ubuntu, theresore you just need to enable them:

sudo a2enmod proxy proxy\_ajp

Create a new Virtual Host in Apache HTTP server represent the DSpace site:

sudo nano /etc/apache2/sites-available/000-default.conf

Modify as follows:

<VirtualHost \*:80>

 ServerName {your-server-name}

 #Add AJP proxy directives

 <proxy \*>

 AddDefaultCharset Off

 Order deny,allow

 Allow from all

 </proxy>

 ProxyPass /ignored\_path !

 ProxyPass / ajp://localhost:8009/

 ProxyPassReverse / ajp://localhost:8009/

</VirtualHost>

Restart Apache

sudo service apache2 restart

### Test:

XML User Interface (aka. Manakin) - (e.g.) http://dspace.myu.edu/xmlui

OAI-PMH Interface - (e.g.) http://dspace.myu.edu/oai/request?verb=Identify (Should return an XML-based response)

### Redirect traffic that come through to the root i.e. http://dspace.myu.edu to http://dspace.myu.edu/xmlui

Make a copy of index.html:

sudo cp /opt/tomcat/webapps/ROOT/index.jsp /opt/tomcat/webapps/ROOT/indexcopy.jsp

Edit index.html and add the redirect code as follows:

sudo nano /opt/tomcat/webapps/ROOT/index.jsp

Amend and file should look like this:

<html>

 <head>

 <title>{server-name}</title>

 <meta http-equiv=refresh content="1;url=http://{repository-domain-name-address}/xmlui" />

 </head>

 <body>

 <p>This is the {server-name}</p>

 <p>For the {service-name}, please <a href="/xmlui">click here</a></p>

 </body>

</html>

OR

Set Tomcat to serve up DSpace by default:

cd /opt/tomcat/webapps

mv ROOT ROOT\_hold

ln -s /home/dspace/dspace/webapps/xmlui ROOT

Restart Tomcat:

sudo systemctl restart tomcat

### Test:

<http://dspace.myu.edu>

## Google and HTML sitemaps

To aid web crawlers index the content within the repository, make use of sitemaps. There are currently two forms of sitemaps included in DSpace: Google sitemaps and HTML sitemaps.

Sitemaps allow DSpace to expose its content without the crawlers having to index every page. HTML sitemaps provide a list of all items, collections and communities in HTML format, whilst Google sitemaps provide the same information in gzipped XML format.

To generate the sitemaps, you need to run [dspace]/bin/dspace generate-sitemaps This creates the sitemaps in [dspace]/sitemaps/

The sitemaps can be accessed from the following URLs:

* http://dspace.example.com/dspace/sitemap - Index sitemap
* http://dspace.example.com/dspace/sitemap?map=0 - First list of items (up to 50,000)
* http://dspace.example.com/dspace/sitemap?map=n - Subsequent lists of items (e.g. 50,0001 to 100,000) etc...

HTML sitemaps follow the same procedure:

* http://dspace.example.com/dspace/htmlmap - Index HTML based sitemap
* etc...

When running [dspace]/bin/dspace generate-sitemaps the script informs Google that the sitemaps have been updated. For this update to register correctly, you must first register your

Google sitemap index page (/dspace/sitemap) with Google at http://www.google.com/webmasters/sitemaps/. If your DSpace server requires the use of a HTTP proxy to connect to the internet, ensure that you have set http.proxy.host and http.proxy.port in dspace]/config/dspace.cfg

The URL for pinging Google, and in future, other search engines, is configured in [dspace]/config/dspace.cfg using the sitemap.engineurls setting where you can provide a comma-separated list of URLs to 'ping'.

You can generate the sitemaps automatically every day using an additional cron job:

# Generate sitemaps at 6:00 am local time each day

0 6 \* \* \* [dspace]/bin/dspace generate-sitemaps

More information on why we highly recommend enabling sitemaps can be found at Search Engine Optimization (SEO): https://wiki.duraspace.org/display/DSDOC5x/Search+Engine+Optimization

# Sources

DSpace 6.x Documentation - [https://wiki.duraspace.org/display/DSDOC6x/DSpace+6.x+Documentation](https://wiki.duraspace.org/display/DSDOC6x/DSpace%2B6.x%2BDocumentation)

Installing DSpace 6.x - [https://wiki.duraspace.org/display/DSDOC6x/Installing+DSpace](https://wiki.duraspace.org/display/DSDOC6x/Installing%2BDSpace)

Stellenbosch University Library's SUNScholar/DSpace - <http://wiki.lib.sun.ac.za/index.php/SUNScholar/DSpace>

David Cohen's Installing DSpace 4.2 on Ubuntu Server 14.04.1 LTS - [https://wiki.duraspace.org/display/DSPACE/Installing+DSpace+4.2+on+Ubuntu+Server+14.04.1+LTS](https://wiki.duraspace.org/display/DSPACE/Installing%2BDSpace%2B4.2%2Bon%2BUbuntu%2BServer%2B14.04.1%2BLTS)

Santhosh Kumar's Installing DSpace 4.1 In Ubuntu 12.04 - <http://dspace.cusat.ac.in/jspui/handle/123456789/7831>

Rob Ingram’s Installing DSpace on Debian 5 - <http://rspproject.wordpress.com/2010/11/10/installing-dspace-on-debian-5/>

# DSpace Resources

Here is a list of DSpace resources. Please note that the list is not by all means exhaustive therefore I would urge use of the web searches for up to date information.

How To guides:

[https://wiki.duraspace.org/display/DSPACE/XMLUI+How+To+Guides](https://wiki.duraspace.org/display/DSPACE/XMLUI%2BHow%2BTo%2BGuides)

<https://wiki.duraspace.org/display/DSPACE/DSpaceResources>

XMLUI:

 [https://wiki.duraspace.org/display/DSPACE/Repository+of+XMLUI+themes](https://wiki.duraspace.org/display/DSPACE/Repository%2Bof%2BXMLUI%2Bthemes)

<http://www.slideshare.net/tdonohue/making-dspace-xmlui-your-own>

<http://cocoon.apache.org/2.1/userdocs/>

DSpace and Google scholar:

<http://lists.webjunction.org/wjlists/web4lib/2004-November/001942.html>

DSpace training material:

<http://www.dspace.org/training-grid/configurable-submission-system-for-dspace-dp6.html>

<http://www.dspace.org/training-grid/configurable-submission-system-for-dspace.html>

Or:

<http://cadair.aber.ac.uk/dspace/handle/2160/615>

<http://www.docstoc.com/docs/2133962/DSpace-Basic-Tutoria>