demo.dspace.org Notes

This page contains information about how demo.dspace.org server is setup/configured. This demo.dspace.org server is managed jointly by the DSpace Committer Team. Any Committer can request server access.

If major issues occur or something needs to be installed requiring root access, contact Tim Donohue or "sysadmin AT duraspace DOT org"

General Server Setup / Info

Here's an overview of how everything is setup on the 'demo.dspace.org' server:

Base Software

- The server is currently running on DuraSpace's Amazon EC2 account with the following configurations:
  - Amazon AMI (EBS Image), currently m3.medium
  - Ubuntu Linux 16.04 LTS server (64-bit)
  - Puppet Setup scripts: https://github.com/DSpace-Labs/puppet-dspace-demo
- All DSpace requirements/software are installed using our puppet-dspace module: https://github.com/DSpace/puppet-dspace
  - Puppet Setup script which installs these prerequisites: https://github.com/DSpace-Labs/puppet-dspace-demo/blob/master/manifests/site.pp
  - Open JDK 8
  - Ant
  - Maven
  - PostgreSQL
  - Tomcat: `~/tomcat`
    - Tomcat is configured to run on port 8080
  - Apache web server
    - Forwards all requests to Tomcat via AJP
  - DSpace Source: `~/dspace-src`
  - DSpace Install: `~/dspace`

'dspace' user account

- The 'dspace' user has FULL 'sudo' access on system.

- The 'dspace' user's `~/bin/` includes various useful scripts

Who to Contact

Only a DuraSpace employee can do the following:

- Recreate the server (using Puppet scripts). **NOTE: The instructions for recreating the server are in the https://github.com/DSpace-Labs/puppet-dspace-demo repository README**
- Upgrade Ubuntu to next version of Ubuntu
- Create Snapshots of server volumes and restore server based on one of those Snapshots

Contact sysadmin@duraspace.org or Tim Donohue if you need any of these tasks performed.

Getting SSH access to demo.dspace.org

This is how you provide a DSpace Committer with command-line access to this server.

1. Have Committer generate an SSH Key on their computer and send you their PUBLIC Key.
2. Append their PUBLIC Key on the end of the 'dspace' user's `~/.ssh/authorized_keys` file
    - NOTE: Please add a comment regarding who's key this is, so that it makes it easier to clean up later on. For example:

    ```
    # Tim Donohue's SSH Key
    ssh-rsa ....
    ```

3. They should now be able to connect as follows:

    ```
    ssh dspace@demo.dspace.org
    ```

Updating / Upgrading DSpace installation

To ensure we are consistently updating DSpace in the same manner, please perform the following steps when updating any configuration or making any customization to DSpace.

(If you have updates/suggestions, please let us know – we can change these processes, but we just need to make sure we are all consistently following the same general steps)

Make all Configuration/File Changes in '~/dspace-src/' FIRST

The `~/dspace-src/` folder is a Git clone of the DSpace-demo GitHub Repository: https://github.com/DSpace-Labs/demo.dspace.org

- This is a public fork of the main DSpace/DSpace GitHub Repository, with a few small tweaks specific to our Demo site

In this local Git repository, we are running off of a **branch named "demo"**. You can see all the branches by running

```
git branch
```

Changes that you wish to keep should be committed to this "demo" branch.

At any one time, you can compare this 'demo' branch to any version of DSpace. For example, to compare 'demo' to DSpace 5.5 run:

```
cd ~/dspace-src
git checkout demo
git diff dspace-5.5
```

WARNING: If you make direct config edits to ~/dspace/config/ you can expect that they may be overwritten in future (unless you also copy them to ~/dspace-src/dspace/config/)
You have been warned! Again, if your changes don't make it to ~/dspace-src/ then THEY WILL BE LOST during the next update!

Upgrade DSpace Source

If you are upgrading to the next stable version of DSpace, you can use `git merge` to help you merge all changes.

For Example:

```
cd ~/dspace-src
# Fetch new branches (eg. minor releases)
git fetch --all
# Pull down all latest changes
git checkout master
git pull
# Merge them into our "demo" branch
git checkout demo
git merge dspace-6.0 (or 'git merge master')
```

NOTE: You should make sure to pay close attention to whether any Conflicts occurred. If so, you will need to resolve them manually.

Resolving Conflicts: Here are some hints on how to resolve / manage conflicts encountered during a merge:

- If there were a lot of conflicts and you just want to accept the "master" or tagged version (and overwrite any local changes), you can use:

  ```
  git checkout --theirs [full-path-to-file]
  git add [full-path-to-file]
  ```

- If you need to completely delete a file that caused conflict, just use:

  ```
  git rm [full-path-to-file]
  ```

- If you need to abort an in-process merge that had conflicts, just run:

  ```
  git merge --abort
  ```

Rebuild DSpace

```
cd ~/dspace-src
# Build DSpace using Mirage 2 theme
mvn -U clean package -Dmirage2.on=true
```

Update the Splash Page and News

These contain the DSpace version number, and should match what we are running! See Managing Website Content below.

Push out Updates

WARNING: this overwrites existing configs in `~/dspace/config/`

```
sudo service tomcat7 stop
cd ~/dspace-src/dspace/target/dspace-installer/
ant update
sudo service tomcat7 start
```

Double check everything still looks to be working.

Also make sure your changes made it to `~/dspace/` (and that you didn't remove previous settings, especially configs)

An easy way to double check config changes is to do a 'diff' of the latest dspace.cfg with the most recent '.old' one.

Commit your changes to "demo" branch

Assuming your changes are already over in `~/dspace-src/` this is easy...

```
cd ~/dspace-src/
git commit [file]

# OR, to commit all changed files
git commit -a

# Push those changes up to GitHub!
git push origin demo
```

DSpace Tweaks (for Demo site)

Disabling the editing of all EPerson Email addresses

In May/June 2015, we ran into several scenarios where users were logging in as a demo Admin account and promptly changing the email address associated with that account. In order to avoid this, it is HIGHLY recommended to disable editing of email addresses on demo.dspace.org.

Here's how it's done:

- In Mirage2, the following jQuery can be added to the ~/dspace-src/dspace-xmlui-mirage2/src/main/webapp/xsl/core/page-structure.xsl:

```
<xsl:template name="buildHead">
<head>
...

   <!-- CUSTOM FOR DEMO.DSPACE.ORG: Don't allow EPerson Emails to be edited, so no one can change
default admin acct emails. -->
   <script type="text/javascript">
       jQuery(function() {
           // Change label for email field in "Edit E-Person"
           jQuery("label[for='aspect_administrative_eperson_EditEPersonForm_field_email_address']").text
("Email Address (editing is disabled on demo.dspace.org)");
           // Make email field in "Edit E-Person" READ-ONLY
           jQuery("#aspect_administrative_eperson_EditEPersonForm_field_email_address").prop("readonly",
true);
        });
   </script>
</head>
</xsl:template>
```

- In JSPUI, the following jQuery can be added to the ~/dspace-src/dspace-jspui/src/main/webapp/layout/header-submission.jsp:

```
<head>
...
   <!-- CUSTOM FOR DEMO.DSPACE.ORG: Don't allow EPerson Emails to be edited, so no one can change
default admin acct emails. -->
   <script type="text/javascript">
       jQuery(function() {
           // Change label for email field in "Edit E-Person"
           jQuery("label[for='temail']").text("Email (editing disabled on demo.dspace.org):");
           // Make email field in "Edit E-Person" READ-ONLY
           jQuery("#temail").prop("readonly", true);
       });
   </script>
</head>
```

Managing Website Content

Website / Splash page

- Tomcat is configured to run on port 8080
    - Apache runs on port 80, and forwards all requests to Tomcat via AJP
- DSpace Webapps are run from `~/dspace/webapps/` (configured in Tomcat's context fragments in `~/tomcat/conf/Catalina/localhost/`)

- The main "splash" page (http://demo.dspace.org) is served by Tomcat and is located at: `~/tomcat/webapps/ROOT/index.html`
    - Its content is also managed via the GitHub Repository at: https://github.com/DSpace-Labs/demo.dspace.org-site
    - Info on updating & pushing to GitHub can be found in the README at https://github.com/DSpace-Labs/demo.dspace.org-site

JavaDocs page

- The JavaDocs pages (http://demo.dspace.org/javadocs/) are static pages served by Tomcat and are located at: `~/tomcat/webapps/javadocs/`
- These JavaDocs can be regenerated at any time by running the following (from the root source directory, `[dspace-source]`):
    - `mvn javadoc:aggregate`
    - If you're generating javadoc of a snapshot version of DSpace, the above would fail. Use `mvn install javadoc:aggregate && rm -rf ~/.m2/repository/org/dspace` instead.
    - The "javadoc:aggregate" command generates a single set of javadocs which aggregate the APIs of all DSpace modules. See http://maven.apache.org/plugins/maven-javadoc-plugin/plugin-info.html
    - The resulting javadoc is in `[dspace-source]/target/site/apidocs`. Upload it to `dspace@demo.dspace.org:/home/dspace/tomcat/webapps/javadocs/[dspace-major-version]/`.
    - NOTE: We've encountered some oddities with the results when this is run from demo.dspace.org itself (the resulting CSS isn't applied). So, it's recommended to run this command from your local machine.
        - It worked fine on 2 machines running Java 6, Maven 2.2.1 and 3.0.3, respectively. It didn't work on demo, which was running Java 7 and Maven 2.2.1.
        - Later, it worked fine on demo running Java 8u181 and Maven 3.3.9.

Starting / Stopping DSpace (and related services)

The 'dspace' user can easily start/stop PostgreSQL and Tomcat using the corresponding service scripts:

```
sudo service postgresql start
sudo service tomcat7 start
~/dspace/bin/start-handle-server

sudo service tomcat7 stop
sudo service postgresql stop
```

Debugging Tips

Log file locations

- DSpace: `~/dspace/log/`
- Tomcat: `~/tomcat/logs/`
- PostgreSQL: `/var/lib/postgresql/9.5/main/pg_log/`
- Apache: `/var/log/apache2/`

Database access

```
# Login to 'dspace' database as dspace (password: dspace)
psql dspace
# Login to 'dspace' database as Postgres Admin (no password needed)
psql -h localhost -U postgres dspace
```

Important Cron Jobs

Obviously, you can get the latest information on the existing Cron jobs by logging into the demo.dspace.org server and running:

```
crontab -l
```

However, here's a brief overview of a few of the more important Cron jobs.

Reset DSpace Content (based on AIPs) every Saturday

EVERY SATURDAY NIGHT (currently at 23:59 UTC), all existing DSpace content is automatically REMOVED and reset to the AIPs located at `~/AIP-restore/`

This is controlled by the `~/bin/reset-dspace-content` script (source code in GitHub)

This is a BASH script that essentially does the following:

1. Stops Tomcat
2. Backs up current DB & Assetstore to `~/tmp/data-backup` (This backup is performed just in case something goes wrong and we need to quickly restore DSpace.)
3. Deletes existing DB & Assetstore
4. Resets DSpace back to a 'fresh_install' state (by restoring database tables, sequences, registries, and initial Admin user acct)
5. Imports the AIPs in `~/AIP-restore`into DSpace as default content (This also autocreates the demo EPeople and Groups)
    - SEE README in ~/AIP-restore/ for info on updating these AIPs
6. Refreshes all Indexes (Lucene/DB & Discovery) & Restarts Tomcat
7. A log of the entire 'reset' process is written to `~/AIP-restore/reset-dspace-content.log`

How to update the set of demo AIPs

The set of demo AIPs are all stored in the `~/AIP-restore/` directory.

To update these AIPs, you must use the DSpace AIP Backup & Restore tools as described at:
AIP Backup and Restore

You can regenerate / update these AIPs by doing the following:

1. Install a fresh (empty) copy of DSpace on your local server.
2. Configure it to have the same handle prefix as demo.dspace.org (handle prefix: 10673) & setup an initial administrative user (ideally 'dspacedemo+admin@gmail.com' which is the Demo Administrator on demo.dspace.org).
3. Download the existing AIPs from this directory, e.g.

    ```
    scp dspace@demo.dspace.org:~/AIP-restore/* .
    ```

4. Use the downloaded AIPs to "restore" content to your local server's empty DSpace, e.g.

    ```
    [dspace]/bin/dspace packager -r -a -f -t AIP -e [admin-email] -i 10673/0 /full/path/to/SITE@10673-0.zip
    ```

5. Update your DSpace's content as you see fit (adding/removing/changing objects)
6. Export a fresh set of AIPs, by performing a full SITE export e.g.

    ```
    [dspace]/bin/dspace packager -d -a -t AIP -e [admin-email] -i 10673/0 -o includeBundles=ORIGINAL,LICENSE
    -o passwords=true SITE@10673-0.zip
    ```

    - The above example just exports ORIGINAL & LICENSE bundles into AIPs, and also exports user passwords into AIPs (so that they can also be restored).
7. Upload those newly updated AIPs to demo.dspace.org, e.g.

    ```
    scp . dspace@demo.dspace.org:~/AIP-restore/
    ```

    - NOTE: Before putting them on demo.dspace.org, you may want to do your own test restore using these AIPs, just to ensure there are no issues.

How to share AIPs for public download

At this time, sharing AIPs is not automated. It's also not currently possible to share them from the default `~/AIP-restore/` location, so this is a bit of a "temporary hack" that needs fixing in the future.

1. First, copy all the AIPs to a shareable location. Below, we chose `/usr/share/dspace/AIP-restore` folder:

```
# Create share location
sudo mkdir -p /usr/share/dspace/AIP-restore

# Manually copy all existing AIPs over there (TODO: This should be automated or synced in future)
cd /usr/share/dspace/AIP-restore/
sudo cp ~dspace/AIP-restore/* .
sudo chown -R dspace:dspace /usr/share/dspace/AIP-restore/

# Add DSpace to www-data user group (to give Apache read access)
sudo usermod -a -G www-data dspace

# Give Apache group rights on directory
sudo chgrp www-data /usr/share/dspace/AIP-restore/
sudo chmod g+rxs /usr/share/dspace/AIP-restore/
```

2. Next, update the Apache configuration for demo.dspace.org to provide access to that shareable location:

```
sudo nano /etc/apache2/sites-available/25-demo.dspace.org.conf

## ADD THE FOLLOWING INTO THAT FILE (inside the <VirtualHost>)
<VirtualHost *:80>
  ...
  # Define path /aip to point at shareable AIP-restore location
  Alias "/aip" "/usr/share/dspace/AIP-restore"
  <Directory "/usr/share/dspace/AIP-restore">
    # Allow viewing file listing
    Options Indexes
    # Don't allow access to README, logs or parent link (..)
    IndexIgnore README* *.log ..
    # Allow access to all
    Order allow,deny
    Allow from all
  </Directory>
  # Don't proxy /aip paths to Tomcat
  ProxyPass /aip !
  ...
</VirtualHost>
```

3. Reload Apache and test it out:

```
sudo service apache2 reload
```

4. Assuming everything works, here's a `wget` command that can be used to download the AIPs to a local computer

```
# This recursively downloads all files (except index.html file) into an "aip" directory
wget -r -np -nH -R "index.html*" --execute="robots=off" http://demo.dspace.org/aip/
```

Reset "News" sections every night

Since the "News" sections are editable via the JSPUI, there is a cron job that automatically resets them each night.

It's a rather simple cron job that just copies the original "news-*" files from the `~/dspace-src/` directory:

```
05 0 * * * cp $HOME/dspace-src/dspace/config/news-* $HOME/dspace/config/ > /dev/null
```

Reset Demo User Passwords every hour

Since people have been known to change our demo user passwords on this demo.dspace.org server, we now reset them to the default password every hour.

This functionality is just a simple set of SQL UPDATE commands that are run via the `~/bin/reset-demo-passwords` script.

kompewter IRC bot

The kompewter IRC bot is on the server at `~/kompewter`.

It's source code is managed in GitHub at https://github.com/DSpace-Labs/kompewter

Starting / Stopping kompewter

To start kompewter just run:

```
cd ~/kompewter
nohup ./jenni > kompewter.log &
```

(NOTE: The "nohup" command ensures that kompewter will keep running even after you log off the server.)

Slack / IRC integration bot

As we now have a DSpace Slack setup, this bot integrates our DSpace Slack with IRC (per the below configuration). It allows messages to be sent from Slack to IRC and vice versa.

Installation

Currently, this installation is NOT automated via Puppet (That should be changed at some point)

We are using this tool: https://github.com/ekmartin/slack-irc

Installation is rather simple:

```
# Ensure we have NPM & Node
# NOTE: "nodejs-legacy" ensures the 'node' command maps to 'nodejs'
sudo apt-get install npm nodejs nodejs-legacy
# Install slack-irc tool
sudo npm install -g slack-irc
# Create a folder where we can store its config, etc.
mkdir ~/slack-irc
```

Configuration

Per the documentation at https://github.com/ekmartin/slack-irc , we just need a **valid** JSON config file to configure this bot.

Here's the current config (save it to `~/slack-irc/config.json`)

```
[
  {
    "nickname": "DSpaceSlackBot",
    "server": "irc.freenode.net",
    "token": "xoxb-147848164820-lkHcW1gt1C01X4kxx3EKtQR4",
    "channelMapping": {
      "#dev-mtg": "#duraspace",
      "#irc": "#dspace"
    },
    "ircOptions": {
      "port": 6697,
      "sasl": true,
      "secure": true,
      "selfSigned": true,
      "certExpired": true,
      "nick": "DSpaceSlackBot",
      "userName": "DSpaceSlackBot",
      "password": "[Ask Tim Donohue for it]"
    },
    "ircStatusNotices": {
      "join": false,
      "leave": true
    }
  }
]
```

This configuration ensures messages on #duraspace IRC are also on the Slack #dev-mtg channel (and vice versa). It also ensures messages on #dspace IRC are also on the Slack #irc channel (and vice versa). Finally, it also authenticates as the registered "DSpaceSlackBot" account with Freenode, which ensures the account is trusted (i.e. won't be blocked). This account is managed by Tim Donohue, so contact him for more info.

Starting / Stopping slack-irc

To start the slack-irc bot just run:

```
cd ~/slack-irc
nohup slack-irc --config config.json > slack-irc.log &
```

(NOTE: The "nohup" command ensures that slack-irc will keep running even after you log off the server.)

Java Profiling using YourKit

> ⚠ **Remote Profiling Using YourKit**
>
> Full instructions available at: http://www.yourkit.com/docs/95/help/profiling_j2ee_remote.jsp

In order to locate potential memory issues in DSpace, we've installed YourKit on demo.dspace.org at `~/yjp/`.

It can be accessed remotely so that we can perform various Java profiling tasks.

On your local computer:

- Download & Install YourKit Profiler. Put in your open source license key (available to all DSpace Committers).
- Open up YourKit, select "Connect to remote application..." option.
- Point it at "demo.dspace.org:10001" and start doing some profiling!
- If it's not running, start it using ~/yjp/bin/yjp.sh
- If needed, logs are in ~/.yjp/log/

Let's Encrypt free DV X.509 certificate

TODO: add to puppet scripts (install package, pull configuration from S3, create cron file)

First-time installation will validate domain ownership and generate a private key. Any subsequent certificate requests will reuse the private key. The `/etc/letsencrypt` directory should be backed up in private S3 storage (TODO).

The certificate is issued for 3 months. The script that checks for renewals needed is running twice a day on a random minute from `/etc/cron.d/certbot`.

```
# Latest install instructions available at: https://certbot.eff.org/lets-encrypt
sudo apt-get update
sudo apt-get install software-properties-common
sudo add-apt-repository universe
sudo add-apt-repository ppa:certbot/certbot
sudo apt-get update
sudo apt-get install certbot python-certbot-apache

# register and request first certificate, but do not change Apache configuration (we'll do it manually)
sudo letsencrypt --apache certonly

Enter email address (used for urgent notices and lost key recovery)
sysadmin@duraspace.org

Which names would you like to activate HTTPS for?
[*] demo.dspace.org

IMPORTANT NOTES:
 - If you lose your account credentials, you can recover through
   e-mails sent to sysadmin@duraspace.org.
 - Congratulations! Your certificate and chain have been saved at
   /etc/letsencrypt/live/demo.dspace.org/fullchain.pem. Your cert will
   expire on 2017-01-04. To obtain a new version of the certificate in
   the future, simply run Let's Encrypt again.
 - Your account credentials have been saved in your Let's Encrypt
   configuration directory at /etc/letsencrypt. You should make a
   secure backup of this folder now. This configuration directory will
   also contain certificates and private keys obtained by Let's
   Encrypt so making regular backups of this folder is ideal.

# replace self-signed certificates with Let's Encrypt certificates
sudo vim /etc/apache2/sites-enabled/25-ssl-demo.dspace.org.conf
  ## SSL directives
  SSLEngine on
#  SSLCertificateFile      "/etc/ssl/certs/ssl-cert-snakeoil.pem"
#  SSLCertificateKeyFile   "/etc/ssl/private/ssl-cert-snakeoil.key"
#  SSLCACertificatePath    "/etc/ssl/certs"
  SSLCertificateFile    /etc/letsencrypt/live/demo.dspace.org/cert.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/demo.dspace.org/privkey.pem
  SSLCACertificateFile  /etc/letsencrypt/live/demo.dspace.org/fullchain.pem

# test renewal (dry run)
sudo letsencrypt renew --dry-run --agree-tos

# set up renewal from cron
sudo vim /etc/cron.d/certbot

# /etc/cron.d/certbot: crontab entries for the certbot package
#
# Upstream recommends attempting renewal twice a day
#
# Eventually, this will be an opportunity to validate certificates
# haven't been revoked, etc.  Renewal will only occur if expiration
# is within 30 days.
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
0 */12 * * * root test -x /usr/bin/letsencrypt && perl -e 'sleep int(rand(3600))' && letsencrypt -n renew --
agree-tos
```

Papertrail log viewer

The logs of demo can be consulted in a webUI through https://papertrailapp.com/systems/demo/events. Ask on #dev for the credentials if you want to have a look.

Installation of this viewer required a SyslogAppender appender to be added to **/dspace/config/log4j.properties**