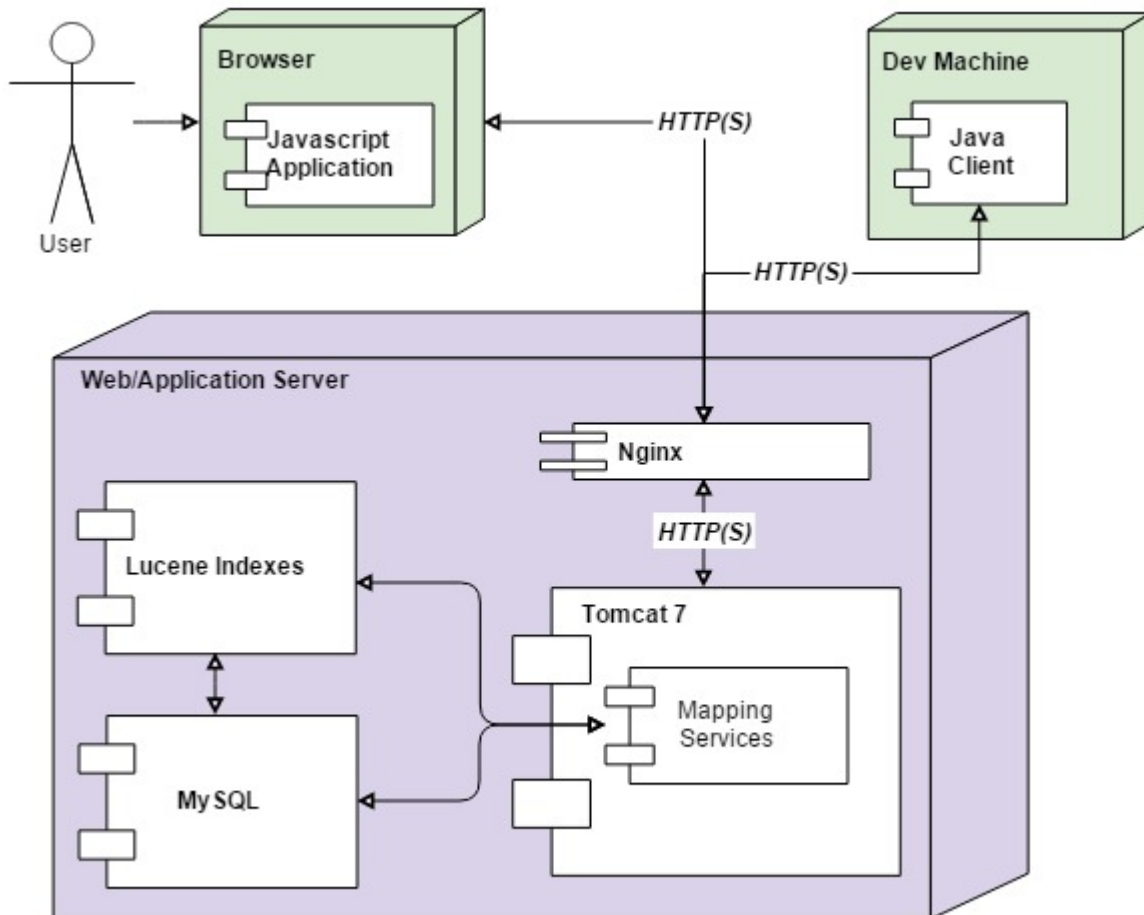


Deploy Instructions

1.1.1. Overview

Describes how to set up and deploy the mapping tool to the IHTSDO mapping.ihtsdotools.org server.

1.1.2. Diagram



1.1.3. Details

1. Log into the server mapping.ihtsdotools.org
 - a. Ensure there is an "ihtsdo" user - everything will run as this user
2. Install and configure software as root (nginx, tomcat7, libapache2_mod_jk, mysql-server-5.5, maven, git, etc)
 - a. You'll need the root user password
 - b. Set up your maven settings.xml according to [Settings.xml Page](#).
 - c. Edit the /etc/init.d/tomcat7 script to comment out the top part that forces running as root

```
#if [ `id -u` -ne 0 ]; then
# echo "You need root privileges to run this script"
# exit 1
#fi
```

- d. Edit the /etc/init.d/tomcat7 script to use this:

```
CATALINA_PID="/var/lib/tomcat7/$NAME.pid"
```

- e. Set permissions of the tomcat installation to the "ihtsdo" user:

```
** This must be done while tomcat is NOT running.
chown -R ihtsdo /var/log/tomcat7
chgrp -R ihtsdo /var/log/tomcat7
chown -R ihtsdo /var/cache/tomcat7
chgrp -R ihtsdo /var/cache/tomcat7
chown -R ihtsdo /var/lib/tomcat7
chgrp -R ihtsdo /var/lib/tomcat7
```

Edit /etc/init.d/tomcat7 to use the following

```
* TOMCAT7_USER=ihtsdo
* TOMCAT7_GROUP=ihtsdo
```

Edit /etc/default/tomcat7

```
* TOMCAT7_USER=ihtsdo
* TOMCAT7_GROUP=ihtsdo
```

- f. See ~root/README.txt [file not present]

3. Tomcat must be configured to pass "run.config" environment variable to the application
 - a. One way to do this is to invoke the java command with -Drun.config=/opt/mapping-service/conf/config/config.properties (e.g. edit this in /etc/init.d/tomcat7 and add to JAVA_OPTS)
 - b. Also add -Xmx2500M to JAVA_OPTS in the tomcat startup script.
 - c. NOTE: development environments must also be configured as such
4. Tomcat must be configured to support POST requests of size > 2MB
 - a. Edit /var/lib/tomcat7/conf/server.xml and add maxPostSize="50000000" to each <Connector> element.
5. Tomcat must be configured to support following of links. Make sure any Context tags in the configuration contain allowLinking="true".
 - a. Edit /var/lib/tomcat7/conf/context.xml to add this parameter to the Context tag.
6. Make sure you have MAVEN_OPTS set
 - a. e.g. -XX:MaxPermSize=512m -Xmx3072m
7. Nginx Configuration
 - a. Install the SSL keys as root in the appropriate places
 - b. Edit /etc/nginx/sites-available/mapping.ihtsdotools.org (see the webapp/src/main/nginx directory for configurations).
 - c. Link /etc/nginx/sites-enabled/mapping.ihtsdotools.org to the file above
 - d. Add a Valve to tomcat to handle proxy headers.
 - e. Edit /etc/nginx/nginx.conf, edit or add "client_max_body_size 50M;" to the "http" context.
 - f. [See the attachment for more details](#)
8. MySQL Configuration
 - a. When configuring the database URL in config.properties, use ?autoconnect=true to avoid connection issues
 - b. Also (as root) edit the my.cnf file for your platform and add this to the mysqld section:

```
[mysqld]
...
wait_timeout = 1048576
interactive_timeout = 1048576
```

- c. It may also be worth considering using the innodb table per file option

```
[mysqld]
...
innodb_file_per_table
```

This is useful to better manage the overall size of the database because individual tables can then be optimized rather than having to dump/reload the entire database.

- d. Consider other optimizations to mysql parameters (e.g. <http://www.percona.com/blog/2014/01/28/10-mysql-settings-to-tune-after-installation/>)

- e. Restart MySQL for the changes to take effect.

```
/etc/init.d/mysql restart
```

9. As the MySQL root user, create an OTF user with the mysql client:

```
CREATE USER 'otf'@'localhost' IDENTIFIED BY 'otfpwd';  
GRANT ALL privileges ON *.* TO 'otf'@'localhost';
```

10. As the MySQL root user, create a database in the mysql client:

```
CREATE DATABASE mappingservicedb  
  DEFAULT CHARACTER SET utf8  
  DEFAULT COLLATE utf8_bin;  
GRANT ALL ON mappingservicedb.* TO 'otf';
```

11. Create "code", "config", and "data" directories

- a. ~/code - for the code
- b. ~/data - for the data files (only needed on initial load)
- c. /opt/mapping-service/config - for the configuration files and configurable admin scripts:

```
cd  
mkdir code  
mkdir data  
mkdir config [may already exist in /opt/mapping-service/]
```

12. Get code:

```
cd ~/code  
git clone https://github.com/IHTSDO/OTF-Mapping-Service.git .  
  
# switch to dev branch - for UAT  
git checkout develop  
  
# switch to dev branch - for PROD  
git checkout master
```

13. Prepare the environment (set your shell up with these ENV variables):

```
JAVA_HOME=/path/to/java  
MAVEN_OPTS=-Xmx3000M
```

14. Build all of the code, making sure to choose the correct "config.artifactId" for your environment. Shown is "prod" configuration:

```
cd ~/code  
mvn -Dconfig.artifactId=mapping-config-prod clean install
```

15. If have a new project go to 16 to get the data, otherwise jump to 17.

16. Get data:

```
cd ~/data  
git clone https://git.ihtsdotools.org/ihtsdo/ihtsdo-mapping-tool-data.git data  
# set up doc directories, for uploading mapping handbook files  
mkdir data/doc  
mkdir data/doc/archive  
chmod -R ga+rw data/doc
```

17. Prepare indexes to be maintainable by ihtsdo user:

```
# To clean up the indexes do this (as root)
/bin/rm -rf /var/lib/tomcat7/indexes
mkdir /var/lib/tomcat7/indexes
mkdir /var/lib/tomcat7/indexes/lucene
mkdir /var/lib/tomcat7/indexes/lucene/indexes
chmod -R ga+rwX /var/lib/tomcat7/indexes
chown -R ihtsdo:ihtsdo /var/lib/tomcat7/indexes
```

18. Unpack the config artifact corresponding to your environment The example shown below is for "prod":

```
cd ~/config
unzip ~/code/config/prod/target/mapping-config-prod*.zip
```

19. Edit the configuration file and scripts.

- a. Edit config.properties and replace the "EDIT_THIS" entries with real values (mostly passwords, etc).
- b. Edit the mail notification settings to list email addresses of those admins or maintenance staff that should be notified of errors and/or automation outcomes.
- c. Edit the scripts in bin/ as needed for this environment (default setup requires no changes).

20. Edit the load script for your environment. Run it to load the data:

```
cd ~/config/bin
./load.csh >&! load.log
```

21. Deploy war files:

```
/bin/cp -f ~/code/rest/target/mapping-rest*.war /var/lib/tomcat7/webapps/mapping-rest.war
```

22. Start service and link "doc" directory:

```
service tomcat7 start

# wait for deploy to create the ROOT directory
sleep 10
# link doc directory
cd /var/lib/tomcat7/webapps/mapping-rest
ln -s ~/data/doc
```

23. Setup cron jobs for nightly/weekly automations. For example,

```
# * * * * * command to execute
# ? ? ? ? ?
# ? ? ? ? ?
# ? ? ? ? ? day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is Sunday, the
same as 0)
# ? ? ? ? ? month (1 - 12)
# ? ? ? ? ? day of month (1 - 31)
# ? ? ? ? ? hour (0 - 23)
# ? ? ? ? ? min (0 - 59)

0 23 * * 4 csh /home/ihtsdo/config/bin/loadDelta.csh > /home/ihtsdo/logs/loadDelta.log 2>&1
0 0 * * 0,1,2,4,5,6 csh /home/ihtsdo/config/bin/qaCron.csh > /home/ihtsdo/logs/qaCron.log 2>&1
0 0 * * 0,1,2,4,5,6 csh /home/ihtsdo/config/bin/dailyReports.csh > /home/ihtsdo/logs/dailyReports.
log 2>&1
```

1.1.4. References/Links

- n/a