Readying a Variations server for Variations-Web - 6.0

Readying a Variations server for Variations-Web - 6.0

This page describes the steps that need to be taken in order to run any part of Variations-Web including the web player and the access manager. These instructions assume that you will be installing Variations-Web on the the same system where the Variations server and its various components (e.g. Perl, Apache, etc.) are already running. These instructions are written assuming the system's operating system is Red Hat Enterprise Linux 5.

Contents

Apache

Configure SSL for Apache

HTTPS using SSL authentication is needed in order to protect private information including user profiles and group/course membership information.

1. Find the DNS hostname that refers to Apache
    - You can find the IP apache is using by looking at the Listen statement in the Apache conf file: **/etc/httpd/conf/httpd.conf**.
2. Ensure that the SSL certificate for Apache is valid by making sure that the Subject Common Name (CN) is the same as the full hostname HTTPS requests will be coming to
    - You can see the CN using: `openssl x509 -in /etc/pki/tls/certs/localhost.crt -noout -text`
    - If it is not valid create a new one (follow the process at your institution or create a self-signed certificate)
        - You can create a self-signed certificate by deleting the one that exists at **/etc/pki/tls/certs/localhost.crt** then running `sudo make testcert` in **/etc/pki/tls/certs** . When prompted for the Common Name, fill in the DNS hostname (the certificate will NOT work if an IP address is used for the Common Name.)
3. Install mod_ssl: `yum install mod_ssl`
4. If necessary, edit **/etc/httpd/conf.d/ssl.conf** by changing **SSLCertificateFile, SSLCertificateKeyFile, and SSLCertificateChainFile** to their proper paths.

Install mod_jk

For Tomcat to be accessed on normal HTTP ports (80 and 443), mod_jk needs to be installed. mod_jk passes all requests on given paths to Tomcat along with SSL credentials if present. Using mod_jk allows a single entry point for all HTTP requests and avoids running Tomcat on higher ports.

1. Install apxs: `sudo yum install httpd-devel`
2. Download latest mod_jk sources from http://tomcat.apache.org/download-connectors.cgi
3. Untar the sources and cd into native: `tar xvzf tomcat-connectors-1.2.32-src.tar.gz; cd tomcat-connectors-1.2.32-src/native`
4. Compile and install mod_jk:

```
CFLAGS="-O2 -g -Wall -fno-strict-aliasing" ./configure --with-apxs=/usr/sbin/apxs --disable-trace --
enable-flock
make
sudo make install
```

Configure mod_jk

- Create **/etc/httpd/conf.d/jk.conf**

```
#########################################
# Mod_jk stuff
#########################################
LoadModule jk_module modules/mod_jk.so
JkWorkersFile /etc/httpd/conf.d/workers.properties
JkLogFile /etc/httpd/logs/mod_jk.log
JkLogLevel warn
JkMountCopy All

JkMount /variations-ws-server/* default
JkMount /variations-ui-web/* default
JkMount /variations-mgmt-web/* default
```

- Create **/etc/httpd/conf.d/workers.properties**

```
worker.list=default

worker.default.port=8009
worker.default.host=localhost
worker.default.type=ajp13
```

Restart Apache

After installing and configuring mod_jk, you will need to restart apache: `sudo /etc/init.d/httpd restart`

Tomcat

Install SSL Certificate

If your SSL certificate is self-signed, then in order for the Web UI and Access Manager connect to the Web Services you will need to add it to the trusted certificates of the newly installed JDK. Run the following as **root**, adjusting the JDK path (both the keytool program and keystore) if necessary:

```
$JAVA_HOME/bin/keytool -import  -trustcacerts -alias apacheLocalhostCA \
-file /etc/pki/tls/certs/localhost.crt -keystore $JAVA_HOME/jre/lib/security/cacerts \
-storepass changeit -keypass changeit
```

Install Tomcat

Do the following steps as **root**, adjusting tomcat's filename if necessary:

1. Download the most recent 6.0 tomcat core from http://tomcat.apache.org/download-60.cgi.
2. Copy it to **/usr/local**: `cp apache-tomcat-6.0.33.tar.gz /usr/local`
3. Untar it: `cd /usr/local; tar xvfz apache-tomcat-6.0.33.tar.gz`
4. Create a general symlink to it: `ln -s apache-tomcat-6.0.33 tomcat`
5. As root, create the tomcat user and change ownership of the tomcat installation to tomcat:
    - useradd tomcat -r -d /usr/local/tomcat
    - chown -R tomcat:tomcat apache-tomcat-6.0.33/
6. Become the new tomcat user: `su - tomcat`
7. Create **.bash_profile** with the following contents (adjusting the jdk path if necessary):

    ```
    export JAVA_OPTS="-Xmx1024m -Dcom.sun.management.jmxremote -XX:MaxPermSize=256m"
    ```

8. Logout and log back in to make sure the new settings take hold by checking the version reported by java (it should be the JDK installed earlier):

    ```
    exit
    su - tomcat
    echo $JAVA_OPTS
    ```

Configure Tomcat

1. Edit **conf/server.xml**
    - Change the redirectPort attribute in the 8009 AJP connector tag to 443.
2. For Variations webapps to connect to the Variations Web Services, you need to create a user with role "wsconsumer" by adding the following lines to **conf/tomcat-users.xml** (replace client_auth_username and actual_password with your own values, which you will later also put in Variations webapp config files):

    ```
    <role rolename="wsconsumer"/>
    <user username="client_auth_username" password="actual_password" roles="wsconsumer"/>
    ```

Starting Tomcat

As the tomcat user, run bin/startup.sh. Check that tomcat has started up properly by opening a browser and navigating to http://tomcat.host:8080/ (where tomcat.host is the hostname where tomcat is running). If tomcat is running, a welcome page will be displayed.

Variations Server Access for Webapps

The Variations Web Services need to connect to the Variations server to access metadata and user profiles. The Web Services require a superuser connection to Variations. Access control to this information is handled by the Variations server access policy file and an access policy file bundled with the Web Services.

1. Create an account that can access your Variations server by making a service or guest account through your institution. **Do not use a personal account.** If your institution does not provide this capability, then you can create a system account on the Variations server and turn on system authentication in Variations.
2. Add the user to the Variations server Administrators group using the varGroup.sh command.