

## 6.x Crowd

- Installation
  - PostgreSQL
  - Setup JDBC4 Driver and JDK 1.x
  - Configuring your Application Server to Connect to PostgreSQL
  - Configure CrowdID to use PostgreSQL
  - Starting Crowd
    - Setup Wizard
    - Options
    - Mail Server - SMTP
- Integrating Crowd with Apache
  - Installing the Crowd Apache Connector on Ubuntu-Linux
- Integrating Crowd with Subversion
- Integrating Crowd with Confluence
- References

### Installation

You can Download Crowd [here](#).

Crowd will be installed using serveradmin

```
su - serveradmin
```

After grabbing the tar file with the wget command use the tar command to unpack it

```
tar -xvpf atlassian-crowd-2.3.2.tar.gz
```

Define the crowd-home Directory by editing the crowd-init.properties file located in /crowd-webapp/WEB-INF/classes/  
Uncomment the following line

```
crowd.home=/var/crowd-home
```

Change the line to the directory you want to specify as crowd home in our case it was changed to the opt directory

```
crowd.home=/opt/crowd-data
```

After create the crowd-data directory and and allow serveradmin ownership

```
sudo mkdir /opt/crowd-data/  
sudo chown -R serveradmin:serveradmin /opt/crowd-data
```

### PostgreSQL

PostgreSQL is needed to start so if you haven't installed it already go to [PostgreSQL](#).

We need to create a database user so change back to your log in because serveradmin cannot use sudo.  
First log in to PostgreSQL then: (remember to actually put a password to replace putYourPasswordHere)

```
CREATE ROLE crowduserdb LOGIN ENCRYPTED PASSWORD 'putYourPasswordHere'
NOINHERIT
VALID UNTIL 'infinity';
COMMENT ON ROLE crowduserdb IS 'Account used by the Crowd application.';
```

After this is entered you'll see

```
CREATE ROLE
COMMENT
```

Create the Database called crowddb

```
CREATE DATABASE crowddb
WITH ENCODING='UTF8'
OWNER=crowduserdb
CONNECTION LIMIT=-1;
```

If this was created correctly you should see

```
CREATE DATABASE
```

Next we need to create a second database to store the employee id data so this database will be called crowdidb  
crowdidb

```
CREATE DATABASE crowdidb
WITH ENCODING='UTF8'
OWNER=crowduserdb
CONNECTION LIMIT=-1;
```

After everything is completely successfully we exit the Postgresql prompt

```
\q
```

## Setup JDBC4 Driver and JDK 1.x

First we need to grab the JDK 1.x file and after the JDBC4 Driver file. The JDBC4 Driver file needs to be in the lib folder in apache-tomcat.

This section should reference or include a java 32-bit installer instructions... which I think already exists.

So we start by going to the [Java SE](#) site and use the wget command to get the file.

```
sudo apt-get install ia32-libs
su - serveradmin
cd /opt/crowd/apache-tomcat/lib/
wget http://jdbc.postgresql.org/download/postgresql-8.4-701.jdbc4.jar
```

Unpack the JDK1.x file into the atlassian-crowd-2.3.2/apache-tomcat/ and then create a symbolic link to java

```
chmod u+x jdk-6u27-linux-i586.bin
./jdk-6u27-linux-i586.bin
sudo ln -s ./jdk1.6.0_27 ./java
```

After we need to edit the startup.sh and shutdown.sh to be only used by serveradmin.  
Followed by editing setenv.sh file in the atlassian-crowd-2.3.2/apache-tomcat/bin/ so that JAVA\_HOME can be defined.  
Make sure to check it works with the version.sh file.

## Configuring your Application Server to Connect to PostgreSQL

Using your favorite editor edit the file apache-tomcat/conf/Catalina/localhost/openidserver.xml and customise the **username**, **password**, **driverClassName** and **url** parameters for the datasource.

```
<Context path="/openidserver" docBase="../../crowd-openidserver-webapp"
debug="0">

<Resource name="jdbc/CrowdIDDS" auth="Container"
type="javax.sql.DataSource"
username="[enter db username here]"
password="[enter db password here]"
driverClassName="org.postgresql.Driver"
url="jdbc:postgresql://host:port/crowdiddb" [ see also
http://jdbc.postgresql.org/doc.html) ]"
[ delete the minEvictableIdleTimeMillis, timeBetweenEvictionRunsMillis and
maxActive params here ]
/>

<Manager className="org.apache.catalina.session.PersistentManager"
saveOnRestart="false"/>

</Context>
```

Because this step is repeated in the final setup of Crowd. This step will be excluded to see if it is necessary.

Delete the **minEvictableIdleTimeMillis**, **timeBetweenEvictionRunsMillis** and **maxActive** attributes. (These are only needed for HSQL database, and degrade performance otherwise.)

## Configure CrowdID to use PostgreSQL

Edit the build.properties file located in the root of Crowd, and modify the hibernate.dialect to the following

```
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
```

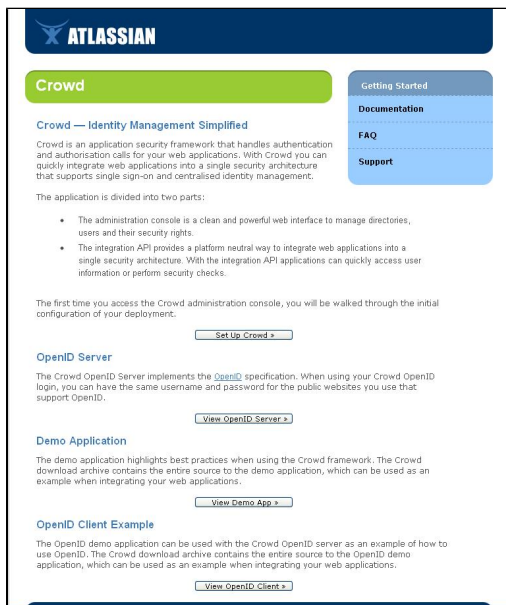
Next modify the jdbc.properties file located at: crowd-openidserver-webapp/WEB-INF/classes/jdbc.properties to

```
# - Crowd Configuration Options

hibernate.connection.datasource=java\:comp/env/jdbc/CrowdIDDS
hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
hibernate.transaction.factory_class=org.hibernate.transaction.JDBCTransactionFactory
```

## Starting Crowd

Start Crowd with startup.sh and go into your browser and use the server IP with the port 8095(port Crowd uses) if all is well you should see this page. If you cannot hit the page check that you have port 8095 open.



## Setup Wizard

1. Start by hitting the Start Crowd button
2. Enter the license key you have bought or generated from starting up an account with Atlassian
3. Click **JDBC Connection**
  - Database: Pick PostgreSQL
  - Enter the username: crowduserdb (in our case)
  - Enter the password you decided to use

## Options

The **deployment title** is a unique name for your Crowd instance. The deployment title is used by default in the subject line of email notifications. You can change this value later, via the Crowd Administration Console.

The **session timeout** determines how long a session will be considered valid during any period of inactivity. This value is specified in minutes and must be greater than 0.

You can change this value later, via the Crowd Administration Console.

The **base URL** is the website address of the Crowd server. This value is used during startup to put the correct values into the crowd.properties file for the Crowd Administration Console.

There is no option to change this value via the Crowd Administration Console, because the URL must not be changed while Crowd is running.

## Mail Server - SMTP

Notification Email Address — The email address which will receive notifications about server events.

From Email Address — Crowd will add this email address as the 'sender' on the emails generated by Crowd and sent to users.

Subject Prefix — The prefix which will appear at the start of the email subject, for all emails generated by Crowd. This can be useful for email client programs that offer filtering rules.

Mail Server Type — Select the 'SMTP Server' radio button.

SMTP Host — The hostname of the SMTP mail server, e.g. 'localhost' or 'smtp.acme.com'.

SMTP Port — The port on which the SMTP mail server listens. The default is '25'.

Username — The username that your Crowd server will use when it logs in to your mail server.

Password — The password that your Crowd server will use when it logs in to your mail server.

Use Secure Sockets Layer (SSL) — Select this check-box if you want to access your mail server over SSL (Secure Sockets Layer). This ensures that all email communications between Crowd and your mail server are encrypted, provided your mail server supports SSL.

## Integrating Crowd with Apache

### Installing the Crowd Apache Connector on Ubuntu-Linux

```
sudo apt-get -y install autoconf libtool apache2-threaded-dev libsvn-dev  
libcurl4-gnutls-dev libxml2-dev apache2-mpm-worker libapache2-svn
```

Enter the following commands

```
wget  
https://studio.plugins.atlassian.com/svn/CWDAPACHE/tags/2.0.2/mod_authnz_c  
rowd-2.0.2.tar.gz  
tar xzf mod_authnz_crowd-2.0.2.tar.gz  
cd mod_authnz_crowd-2.0.2  
autoreconf --install  
./configure  
make  
sudo make install
```

Restart Apache after

## Integrating Crowd with Subversion

This setup requires that you have installed the crowd apache connector.

If you are using Apache to manage access to a subversion repository (instructions), you can use the same configuration method to delegate user authentication to Crowd.

To restrict Subversion repository access to certain groups and/or users, you can add the Require group and Require user directives, described in the page on [integrating Crowd with Apache](#).

For more fine-grained access, Crowd provides the AuthzSVNCrowdAccessFile directive which allows you to define path-based access rules.

These modifications need to be made to the default file in /etc/apache2/sites-available/

Also the code added must be located before the error log code in the file.

```
<Location /svn>

    AuthName "Atlassian Crowd"
    AuthType Basic
    AuthBasicProvider crowd

    CrowdAppName subversion
    CrowdAppPassword yourpassword
    CrowdURL http://localhost:8095/crowd/

    CrowdCreateSSO off # Improves performance when using Subversion clients
    that

    DAV svn

    # Set this to the path to your repository
    SVNPath /opt/svn-data/hamsters

    AuthzSVNCrowdAccessFile /etc/apache2/dav_svn.authz
    Require valid-user

</Location>
```

Next you should create your repository... steps to go here.

After the repository is created, in order for the WebDAV process to work with the repository you need to change the permissions,

```
cd /opt/svn-data
sudo chown -R www-data:subversion hamsters
sudo chmod -R g+rws hamsters
```

You will need to do this for all **new** repositories or else you will get the error (to document).

## Integrating Crowd with Confluence

...

## References

<http://confluence.atlassian.com/display/CROWD/PostgreSQL+for+CrowdID>

<http://confluence.atlassian.com/display/CROWD011/3.2.1+Integrating+Crowd+with+Apache+or+Subversion>

This article helped with permission of repositories - <https://help.ubuntu.com/community/Subversion>

Crowd Authorization options use the same syntax in SVN - <http://svnbook.red-bean.com/en/1.6/svn.serverconfig.httpd.html#svn.serverconfig.httd.authz>

This talks specifically about using the authz paths - <http://svnbook.red-bean.com/en/1.6/svn.serverconfig.pathbasedauthz.html>

Confluence 4.1 removes Legacy Crowd SSO Authenticator - <http://confluence.atlassian.com/display/DOC/Confluence+4.1+Upgrade+Notes>