Server Team’s Installation Manual

How to Build Server Environment for CSE 455

**Server Team:**

Garrett Cuillier, Yakira Dixon, Paul Cummings

**Prerequisites:**

The steps described in this guide are performed on a server running Debian Linux. This guide will show the steps to first setup mysql/php, subversion, and then how to setup Redmine.

**Setup Mysql/php:**

1) Run the following to install mysql and php

$ sudo apt-get install apt-get -y install php5 php5-cli php5-curl php5-dev php5-gd php5-mcrypt php-pear php5-mysql mysql-server

2) set the root password to mysql with the following:

$ sudo mysqladmin -u root -h localhost password 'mypassword'

3) login to mysql with the command:

$ mysql -u root -p

**setup project databases**

Project databases need to be set up and users created with specific to that database. The user will only be allowed to connect from the vpn network and localhost. An example of a sql script looks like:

create database studentadvisingdb;

create user 'sadbuser'@'139.182.20%.%' identified by 'password';

grant all privileges on studentadvisingdb.\* to 'sadbuser'@'139.182.20%.%';

create user 'sadbuser'@'localhost' identified by 'passwordl';

grant all privileges on studentadvisingdb.\* to 'sadbuser'@'localhost';

**Setup SVN:**

1) First Install Subversion. This can be done with the following command:

$ sudo apt-get install subversion libapache2-svn

2) Once svn is installed, create svn repositories. This can be done with the following command:

$ svnadmin create /path/to/repos/project

In the case of this guide the svn repositories were located in /var/svn/projectname

So used the command as follows:

$ svnadmin create /var/svn/museum

$ svnadmin create /var/svn/sodexo

We did this for each of the projects

3) For each of the svn repositories that were created change the ownership to the www-data user(Debian’s http user). This allows users to import and commit repository files over http. To change ownership, use this command for each repository:

$ sudo chown -R www-data:www-data /var/svn

4) On your own machine, make sure that Tortoise SVN is installed. Now create a new folder. This folder can be named anything. Right click on the folder and choose “SVN Checkout”. You should be prompted to enter the server URL with the path to the repository. Example:

http://serverURL/svn/sodexo

After clicking “Okay”, you should see a prompt for username and password. Enter the username and password and it should checkout the contents of the SVN repository that was selected. If this works, svn is setup correctly

**Setup SVN hooks**

1) Once svn is setup and functioning correctly, setting up svn hooks will make sure that whenever a svn commit is performed, the www directory is updated as well. On the server, the www directory contains everything as it would appear on an actual web server. The svn hooks are scripts that perform such updates after actions like a commit.

2) The svn hooks are located in the repositories directories, for example /var/svn/museum/hooks. We will be using the post-commit hook.

3) The following steps will need to be repeated for each svn repository Repeat steps 4-9 in order for each of the repositories. First switch to the root user with the command:

$ su

4) Now use this command to switch to the www-data user:

note: the repos must be checked out by apache first otherwise you will run into permission issues.

$ su - www-data -c /bin/bash

5) Change to the directory /var/www and run this command:

$ svn co file:///var/svn/museum

And be sure to use the name of the project that this process is being done for.

If the above command was successful, a revision number should be shown.

6) After running the above command, type ‘exit’ to get out of the www-data shell. Now go to the hooks directory of the project, for example: var/svn/museum/hooks

7) In this directory, rename the post-commit script with the following command:

$ mv post-commit.tmp post-commit

And change the permissions of this script:

$ chmod 744 post-commit

8) Edit post-commit and make sure everything in the file is commented out, then add the following line:

/usr/bin/svn update /var/www/wri

9) Repeat steps 4-8 in order for each project.

10) To test that the post-commit hook is working, first create a folder on your local system and name it something like “cse455svn”. Then in this folder create a new folder for each project and give each folder it’s corresponding project name. In each project folder, create 2 additional folders called “tag” and “trunk”.

11) After completing the above process, right click on each of the project folders and do a SVN Checkout for the corresponding project repositories. Now go to the truck folder and if it doesn’t exist already, create a index.html file. Make a change to this file and the right click on the project folder and do a SVN Commit.

12) To test that the post-commit hook worked, browse to the server’s URL with the project directory specified and make sure the changed index.html appears. Server URL example:

serverURL/projectName

**Redmine Installation**

Redmine <<http://redmine.org>> is a Ruby application for software project management. Focus on understanding these features: SCM integration, LDAP authentication, automatic account creation, database support for MySQL, PostgreSQL, SQLite.

The Redmine package“redmine” and its dependencies can be downloaded and installed using your system's package manager (aptitude or apt-get for Debian). During the installation, you will be asked to configure Redmine's database and to create a Redmine admin user.

**Configuring LDAP authentication for SVN**

You'll need to enable the module mod\_authnz\_ldap for Apache webserver version 2.2+. Also, make sure mod\_dav\_svn is enabled. To configure this, you’ll need to modify the config file /etc/apache2/mods-available/dav\_svn.conf. This file contains several Apache directives that relate to SVN and LDAP.

Example dav\_svn.conf:

<Location /svn>  
  
 # Uncomment this to enable the repository,  
 DAV svn  
  
 # Set this to the path to your repository  
 SVNParentPath /var/lib/svn

SVNPathAuthz on

# Basic authentication with LDAP

AuthType Basic

AuthBasicProvider ldap

AuthName " CSE 455 Subversion Repository"

AuthzLDAPAuthoritative on

# Enable the following two lines if you are not performing an anonymous bind

#AuthLDAPBindDN "ou=People,dc=csusb,dc=edu"

#AuthLDAPBindPassword password

AuthLDAPUrl "ldap://139.182.5.179:389/ou=People,dc=csusb,dc=edu?uid,userPassword?sub?(objectClass=\*)"

# This directive will return the attribute that will identify the remote user

AuthLDAPRemoteUserAttribute uid

# To enable authorization via mod\_authz\_svn

AuthzSVNAccessFle /etc/apache2/dav\_svn.authz

</Location>

The command apache2ctl -M will output a list of enabled modules.

References: SVNBook - mod\_dav\_svn: <<http://svnbook.red-bean.com/en/1.7/svn.ref.mod_dav_svn.conf.html>>

Apache documentation: <<http://httpd.apache.org/docs/2.2/>> and directive index <<http://httpd.apache.org/docs/2.2/mod/directives.html>>

Use tools like ldapsearch to determine the directory schema and to help you construct valid queries.

**Configuring LDAP authentication for Redmine**

LDAP configuration can be done through the graphical interface. Go to the Administration menu and select "LDAP authentication." You'll need to know the following information about the LDAP server you are using:

* the server’s IP address
* the port you're connecting to (default port is 389)
* the base distinguished name (DN) used for the search
* the attributes that the search needs to return

Select "on-the-fly user creation" to create a Redmine account for any user who has authenticated for the first time.

**Path-Based Authorization with dav\_svn**

To configure path-based authorization, you'll need to edit the file

/etc/apache2/dav\_svn.authz

Users are placed into groups, and groups are given access to repositories. You can specify whether the groups have read (r) or write (w) access. The asterisk variable (\*) represents all users.

Example dav\_svn.authz:

[/]

\* = r

[groups]

managers = cderyll

developers = jsissel, mhashimoto, rtyler

sysadmins = ydixon, gcullier, pcummings

qa = mbates, rsill, kdouglas

a-team = @managers, @developers

[coderepo:/]

@a-team = rw

@sysadmins = r

@qa = r

When you restart apache, the configuration file may fail to load. Check apache’s error log to determine if the file was loaded or not.

References: SVNBook: Path-Based Authorization <http://svnbook.red-bean.com/en/1.7/svn.serverconfig.pathbasedauthz.html>