Rose-Hulman Institute of Technology

University of Wisconsin–Madison | Department of Computer Sciences
Human-Computer Interaction Laboratory

THE UNIVERSITY of WISCONSIN MADISON

Deployment Guide

Trey Cahill Katie Greenwald Samad Jawaid Kevin Risden

1 Introduction

This document will provide a very good sense of how to deploy the Participant Scheduling System, but it will not be a definite guide. Though most instructions include concrete examples, some steps may be missing. Pay special attention when configuring Nginx, Apache, and PostgreSQL. Furthermore, some values will need to be changed, like the references to `rose-hulman.edu`. Good luck!

2 Install Nginx, Apache, mod_wsgi, PostgreSQL, and Postfix

2.1 Definitely

Command:

```
yum install nginx apache2 libapache2-mod-wsgi postgresql python-psycopg2 postfix
```

2.2 Maybe

Command:

```
yum install httpd mod_wsgi postgresql-server
```

2.3 Configure Nginx

File:

```
/etc/nginx/proxy.conf
```

Contents:

```
proxy_redirect          off;
proxy_set_header        Host            $host;
proxy_set_header        X-Real-IP       $remote_addr;
proxy_set_header        X-Forwarded-For $proxy_add_x_forwarded_for;
client_max_body_size    10m;
client_body_buffer_size 128k;
proxy_connect_timeout   90;
proxy_send_timeout      90;
proxy_read_timeout      90;
proxy_buffers           32 4k;
```

File:

```
/etc/nginx/sites-available/pss.csse.rose-hulman.edu
```

Contents:

```
server {
    listen 137.112.40.203:80;
    server_name pss.csse.rose-hulman.edu;

    access_log /var/log/nginx/pss.csse.rose-hulman.edu.access.log;
    error_log /var/log/nginx/pss.csse.rose-hulman.edu.error.log;
    location / {
        proxy_pass http://127.0.0.1:80/;
        include /etc/nginx/proxy.conf;
    }

    location /static/admin {
        alias /opt/django/django/contrib/admin/static/admin;
        expires 24h;
    }

    location /static {
        alias /srv/django/participant-scheduling-system/src/static;
        expires 24h;
    }
}
```

Command (note line break to remove):

```
ln -s /etc/nginx/sites-available/pss.csse.rose-hulman.edu \
/etc/nginx/sites-enabled/pss.csse.rose-hulman.edu
```

2.4 Configure Apache

File:

```
/etc/httpd/conf.d/pss.csse.rose-hulman.edu
```

Contents:

```
<VirtualHost 127.0.0.1:80>
    ServerName pss.csse.rose-hulman.edu
    ServerAdmin jawaidss@rose-hulman.edu

    ErrorLog /var/log/httpd/pss.csse.rose-hulman.edu-error_log
    CustomLog /var/log/httpd/pss.csse.rose-hulman.edu-access_log vhost_combined

    ErrorDocument 403 "/403/"
    ErrorDocument 404 "/404/"

    WSGIScriptAlias / /srv/django/participant-scheduling-system/src/pss.wsgi
</VirtualHost>
```

2.5 Configure PostgreSQL

Command (to edit the file):

```
su postgres
```

File:

```
/var/lib/pgsql/data/pg_hba.conf
```

Contents (to append to the file):

```
local   all         postgres                          ident
local   pss         pss                               md5
```

Command:

```
/etc/init.d/postgresql restart
```

3 Install Git, Subversion, and Mercurial

Command:

```
yum install git subversion mercurial
```

4 Install Django, Django Registration, Django Uni-Form, Django Ajax Validation, and Django Extensions

Commands:

```
svn co http://code.djangoproject.com/svn/django/trunk/ /opt/django
hg clone https://bitbucket.org/ubernostrum/django-registration /opt/.
git clone https://github.com/pydanny/django-uni-form.git /opt/.
git clone https://github.com/alex/django-ajax-validation.git /opt/.
git clone https://github.com/django-extensions/django-extensions.git /opt/.
```

Command (example return value is `/usr/lib/python2.7/site-packages`):

```
python -c "from distutils.sysconfig import get_python_lib; print get_python_lib()"
```

File:

```
/usr/lib/python2.7/site-packages/ajax_validation.pth
```

Contents:

```
/opt/django-ajax-validation
```

File:

```
/usr/lib/python2.7/site-packages/django_extensions.pth
```

Contents:

```
/opt/django-extensions
```

File:

```
/usr/lib/python2.7/site-packages/django.pth
```

Contents:

```
/opt/django
```

File:

```
/usr/lib/python2.7/site-packages/registration.pth
```

Contents:

```
/opt/django-registration
```

File:

```
/usr/lib/python2.7/site-packages/uni_form.pth
```

Contents:

```
/opt/django-uni-form
```

Command:

```
ln -s /opt/django/django/bin/django-admin.py /usr/bin
```

5 Install Participant Scheduling System

Commands (note line break to remove):

```
mkdir /srv/django
git clone https://github.com/compuwizard123/participant-scheduling-system.git \
/srv/django/.
```

5.1 To reset

Commands:

```
cd /srv/django/participant-scheduling-system/src
./production-django-admin resetall
```

6 Everything Else

File:

```
~/pss.py
```

Contents:

```
#!/usr/bin/env python

path_to_settings = '/srv/django/participant-scheduling-system/src/pss/settings.py'

changes = (
    ('DEBUG = True', 'DEBUG = False'),
    ("'USER': 'jawaidss'", "'USER': 'pss'"),
)

old_settings = open(path_to_settings)
settings = old_settings.read()
old_settings.close()

for change in changes:
    settings = settings.replace(*change)

new_settings = open(path_to_settings, 'w')
```

```
new_settings.write(settings)
new_settings.close()
```

```
chmod +x ~/update.py
```

File:

```
~/update-pss
```

Contents (note line break to remove):

```
#!/bin/bash
if [ "$(whoami)" == "root" ]; then
    cd /var/backups/postgres
    sudo -u postgres pg_dump --no-owner --format=c pss > \
pss.dump.$(date +"%Y-%m-%d-%H-%M-%S")
    FOLDER=/srv/django/participant-scheduling-system
    cd $FOLDER
    git reset --hard HEAD
    git pull
    ./pss.py
    chown -R apache:apache $FOLDER
    /etc/init.d/httpd reload
    /etc/init.d/nginx reload
else
    echo "sudo?"
fi
```

File:

```
~/update.py
```

Contents:

```
#!/usr/bin/env python

import datetime
import os

IGNORE = (
)

WORKSPACE = '/opt'
SVN = 'svn'
HG = 'hg'
GIT = 'git'

def _print_line():
    print '-' * 80

def _execute(program, *args):
    os.system('%s %s' % (program, ' '.join(args)))

def _update(folder):
    print folder
    os.chdir(folder)
    files = os.listdir(folder)
    if '.' + SVN in files:
        _execute(SVN, 'up')
    if '.' + HG in files:
        _execute(HG, 'pull')
        _execute(HG, 'update')
    if '.' + GIT in files:
        _execute(GIT, 'pull')
    _print_line()

def main():
    now = datetime.datetime.now()
    _print_line()
    for folder in os.listdir(WORKSPACE):
        if not folder.startswith('.') and not folder in IGNORE:
            _update(os.path.join(WORKSPACE, folder))
    then = datetime.datetime.now()
    print (then - now).seconds, 'seconds'
    _print_line()

if __name__ == '__main__':
    main()
```

6.1 To update Participant Scheduling System

Command:

```
./update-pss
```

6.2 To update everything else

Command:

```
./update.py
```