Deploying the CSDT Community Site

GK-12 @ RPI

June 5, 2014

1 Objective

To document the procedure for deploying an updated version of the CSDT Community Site (CCS) server package. This document is intended for developers of the CCS or CSnap.

1.1 Definitions

CSDT Community Site (CCS) The CSDT Community Site software is a Django-based application which allows users to easily share CSnap projects, ideas, and other material related to the GK-12 grant at RPI. For more information, please see the document (Insert bilbo reference here)

CSnap CSnap is a drag-and-drop visual programming environment. It integrates with the CCS to help students learn programming skills. For more information, please see the document (Insert bilbo reference here)

1.2 Big Picture

The procedure works as follows:

a. Schedule downtime, bring down the live server

b. Clone the server hosting the live CCS

c. Perform updates on the cloned server

d. Run all unit tests

e. Run all manual tests

f. Bring up the live server

2 Clone the live server

The live server contains CCS application, live database, and uploaded user files. Before performing an upgrade, we will clone the live server so if something goes wrong, we don't lose any data.

2.1 Procedure

Schedule time with Ethan Coppenrath (`COPPEE@rpi.edu`) to update the community site. During this time, Ethan will need to:

a. Disconnect the live server from the internet

b. Clone the live server

c. Give the developer time to update the cloned server

d. Connect the cloned server to the internet as the live server

If something goes wrong and results in a longer than anticipated down time (more than an hour), the old live server should be brought back online while the developer debugs the changes.

3 Updating on the cloned server

The first step in updating is to gain access to the server. After gaining access, update the application, static files, and migrate the database.

3.1 Gaining access

Login information is stored seperately in the csdt_doc_private respository, which the developer will need access to. To gain access to this repository, contact Ron Eglash (`eglash@rpi.edu`).

3.2 Updating the application

This section depends a lot on what changes were made. In general, it will involve:

a. Performing a git-pull from the master repository on Github

b. Updating all installed Django/Python modules

c. Performing updates for any linked projects (such as CSnap)

d. Updating the collected static files

e. Migrating the database

3.2.1 Updating CCS

Run the commands in listing 1 one at a time, and deal with errors as they appear. Expected output is displayed below each command. Large outputs are truncated.

```
# Change to the directory that contains the source code
cd /var/www
# Update the code
git pull
# Activate the python virtual environment
. ./activate
# Update all the installed packages
pip install --upgrade -r libraries.txt

# Perform updates for linked projects, if desired
git submodule update
# You may have to perform special updates for the linked
    projects
# See that projects documentation for more details

# Update the collected static files
#  These are kept cached to help with performance
python ./manage.py collectstatic
# You will be prompted, type yes to continue

# Migrate the database
# !! This is the most dangerous part! And why we keep backups
# If there are errors here, they could be hard to fix
#  might be worth having the old site be put back up while
#  you investiage
python ./manage.py migrate
```

Listing 1: Updating the application

4 Running all unit tests

Run the commands in listing 2 to perform the built-in unit tests.

```
# Change to the directory that contains the source code
cd /var/www
# Activate the python virtual environment
. ./activate

#Run the tests
python manage.py test

#Expected output looks like this
Ran 22 tests in 0.221s

OK
```

Listing 2: Running unit tests

If the unit test reports any failures, fix the CCS so that all tests pass before continuing.

5 Run manual tests

Connect to the server by adjusting the your configuration so that `http://community.csdt.rpi.edu` points to the IP of the server you are configuring. Run the tests documented in (insert reference to the manual testing document here).

6 Bringing up the live site

Once all updates and tests have been completed, bring the site back up. To do this, simply update the IP on the cloned machine so that it replaces the old machine.