Canvas LMS Testing Framework Report

Don’t Test Me

Kenneth Dela Cruz, Kyle Glick, Sam Word

Table of Contents

Chapter 1: Installation

I. Installation Report

Problems and difficulties were encountered when attempting to build Canvas on an Ubuntu image. On their GitHub page, Canvas provided a Quick-Start Guide which included a set of instructions for using their quick-start script and another set without using the script. Our first hurdle came from attempting to use their automated script. The script failed at some point during its execution, and being unaware of the details of the script, we chose to set up Canvas manually. Most of the issues we encountered while manually installing Canvas were a result of our inexperience with installing Ruby on Rails projects and their dependencies. For instance, two of us installed an incorrect version of Ruby, causing problems later on in the process, and we all encountered issues with permissions in PostgreSQL. We often resolved these issues by searching online for a way to undo our mistakes and trying again.

Unfortunately not all of the obstacles we encountered were as trivial as installing the wrong Ruby version. In fact, the most difficult issue we encountered surfaced while running a built in Rake task to install all of the required assets. Canvas uses a custom fork of jQuery 1.7.2 with a very, very small change. When Canvas’s asset management tool, Yarn, tried to retrieve the code from that jQuery repository, it complained about an Invalid Tar Header coming from the source of the repository. We were originally unable to get this version of jQuery to run, so we tested various jQuery releases, eventually finding that version 2.2.4 would compile. However, when running the application, nothing would load on the page. It turned out that jQuery wasn’t being defined using version 2.2.4. After struggling with this issue for awhile, we decided to write an Issue on Canvas’s Github page. While writing the issue, we reset the jQuery version used by Yarn to get a screenshot of the error to find that the error no longer appeared. Surprised, once we finished compiling the assets again and ran the application, everything functioned as intended. We’re still not entirely sure what fixed this issue, but we believe that adding several asset folders and files, a step only mentioned in the Production Start Guide, may have resolved our issue.

On our team Github wiki, we provided step-by-step instructions for settings up Canvas. Many of the steps were taken from Canvas’s Quick Start Guide, though a few additions were made that were required to get the application running. Some of these were manual edits that we believe are Ubuntu-only issues, such as providing access to the PostgreSQL server, while others were steps in the Canvas Production Start Guide that were not in the Quick Start Guide.

II. Custom Installation Guide

This is a guide on setting up Canvas LMS by Instructure Inc. in an Ubuntu environment based on our experiences. The official development and production start guides can be found on the Canvas GitHub:

- Official Quick Start Guide
- Official Production Start Guide

Set Up Git

```
sudo apt-get install git
git clone https://github.com/instructure/canvas-lms.git canvas
cd canvas
```

Install Curl

```
sudo apt install curl
```

Install NodeJS and PostgreSQL

```
curl -sL https://deb.nodesource.com/setup_6.x | sudo bash -
sudo apt-get install nodejs postgresql
```

Install Ruby and Ruby Dependencies

```
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:brightbox/ruby-ng
sudo apt-get update
sudo apt-get install ruby2.4 ruby2.4-dev zlib1g-dev libxml2-dev
libsqlite3-dev libpq-dev libxmlsec1-dev curl make g++
```

Create PostgreSQL Superuser

```
sudo -u postgres createuser $USER
sudo -u postgres psql -c "ALTER USER $USER WITH SUPERUSER" postgres
```

Configure Gem Installation Directory

Note: You will have to export the GEM_HOME directory for each new bash session. If you don't want to do this, the Quick Start guide explains how to permanently link to the gem installation directory.

```
mkdir ~/.gems
export GEM_HOME=~/.gems
```

Install Bundler

```
gem install bundler -v 1.15.3
```

Install Yarn

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee
/etc/apt/sources.list.d/yarn.list
sudo apt-get update && sudo apt-get install yarn
```

Install Python (for Yarn)

```
sudo apt-get install python
```

Install Gems

```
$GEM_HOME/bin/bundle install
```

Install CoffeeScript

```
sudo npm install -g coffee-script@1.6.2
```

Set Up Config YML Files

```
for config in amazon_s3 delayed_jobs domain file_store outgoing_mail
security external_migration; do cp -v config/$config.yml.example
config/$config.yml; done
```

Set Up Database Config YML File

```
cp config/database.yml.example config/database.yml
```

Create Tables and Populate Data

Note: The email and password you enter here will be your login credentials later.

```
$GEM_HOME/bin/bundle exec rake db:initial_setup
```

Create Test Database Admin

```
psql -c 'CREATE USER canvas' -d postgres
psql -c 'ALTER USER canvas CREATEDB' -d postgres
```

Give PostgreSQL Permissions

```
sudo gedit /etc/postgresql/9.5/main/pg_hba.conf
```

Change `peer` to `trust` where it's used.

Create Test Database

```
createdb -U canvas canvas_test
psql -c 'GRANT ALL PRIVILEGES ON DATABASE canvas_test TO canvas' -d
canvas_test
psql -c 'GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO canvas' -d
canvas_test
psql -c 'GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA public TO canvas'
-d canvas_test
psql -c "ALTER USER canvas WITH PASSWORD 'password'" -d canvas_test
```

Configure Test Database Credentials

```
gedit config/database.yml
```

Add `password: password` inside of the test environment YML `test:`

Set Up Test Database Tables

```
RAILS_ENV=test $GEM_HOME/bin/bundle exec rake db:test:reset
```

Create Asset Files and Directories

```
mkdir -p log tmp/pids public/assets app/stylesheets/brandable_css_brands
touch app/stylesheets/_brandable_variables_defaults_autogenerated.scss
touch Gemfile.lock
sudo chown -R canvasuser config/environment.rb log tmp public/assets
app/stylesheets/_brandable_variables_defaults_autogenerated.scss
app/stylesheets/brandable_css_brands Gemfile.lock config.ru
```

Compile Assets

```
yarn install
$GEM_HOME/bin/bundle exec rake canvas:compile_assets
$GEM_HOME/bin/bundle exec rake brand_configs:generate_and_upload_all
sudo chown -R canvasuser public/dist/brandable_css
```

Run Server

```
$GEM_HOME/bin/bundle exec rails server
```

Glossary of Terms

JavaScript:
A high-level object-oriented programming language developed by Brendan Eich and released in 1995. It is used primarily in World Wide Web content to make interactive pages and online applications.

jQuery:
A JavaScript library built originally by John Resig and released in 2006. It’s primary purpose is to make manipulating HTML documents easier and provide improved interactivity with document elements.

PostgreSQL:
A popular Database Management System (DBMS) released in 1996.

Rake (Ruby Make):
A build utility for Ruby, similar to the “make” command.

Rake Task:
A Ruby script built to run using Rake syntax.

Ruby:
A dynamic, general-purpose programming language developed in the mid-1990s by Yukihiro Matsumoto. It was designed with ease-of-use and flexibility in mind. It can be used for functional, object-oriented, and imperative programming.

Ruby on Rails (Rails):
A server-side MVC web application framework in Ruby developed by David Heinemeier Hansson in 2005. It uses HTML, CSS, and JavaScript for user interfacing and JSON and XML for data transfer. It also provides a built-in test suite and a variety of external libraries for developers.

Yarn:
A Dependency Management System (DMS) used to compile and control packages and assets.