HHS Voice of the Customer (VOC) Deployment Requirements Guide

Software Requirements:

Ruby Version Manager (RVM)
https://rvm.io/

RVM is the industry standard method of supporting multiple ruby environments on the same box or virtual machine while keeping each of them in complete isolation from each other. RVM enables the deployment of multiple projects, each with its own completely self-contained and dedicated ruby environment, from the specific version of ruby, all the way down to the precise set of required gems application. RVM prevents the issue of gem version conflicts between projects. RVM permits easy testing of gems and gemsets for upgrades, by switching to a new clean set of gems to test with, while leaving the original current live set intact. It is flexible enough to even maintain a set of gems per environment, or per development branch, or even per individual developer's taste!

Ruby MRI 1.9.3-p484, Bundler v.1.3.5
Ruby version 1.9.3-p484, which includes Bundler v. 1.3.5, which is a standard gem used to manage and build gemsets for Ruby projects. Bundler is installed by RVM when installing most Ruby versions.

Gem Dependencies
Rails 3.0.20 – Ruby on Rails web framework
jQuery-rails 2.1.4 – jQuery javascript library bridge for Ruby on Rails
Kaminari 0.14.1 – Pagination plugin
Authlogic 3.2.0 – Authentication plugin
Memcache-client 1.8.5 – Memcached interface for Ruby on Rails
Paperclip 3.4.0 – Document attachment management
Resque 1.24.1 – Asynchronous jobs processor
Resque_mailer 2.2.3 – Asynchronous Mailer processor
Resque-status 0.4.1 – Makes redis/resque jobs trackable
Bson_ext 1.9.0 – Provides BSON support for mongo database
Escape_utils 0.3.2 – Fast string escaping for rails
Mongoid 2.2.6– Provides ORM support for mongo database
Open_uri_redirections 0.1.1 – Handles redirection between http / https fluidly
Ranked-model 0.2.1 – Database row sorting library
Redis-objects 0.7.0 – Redis Queue Integration
Pdfkit 0.5.4 – PDF generation utility
Best_in_place 0.2.4– Javascript / support for in place editing of tables

Database Requirements

MySQL 5.5 or later should be installed with a custom application user with read/write and create table permissions.

MongoDB 2.2.0 or later should be installed with a custom application user with read/write/create permissions.

Redis 2.8 or later should be installed.

Redis is an open source, BSD licensed, advanced key-value store. It is often referred to as a data structure server since keys can contain strings, hashes, lists, sets and sorted sets. Redis runs as a deamon like MySQL or MongoDB. Full documentation can be found at http://redis.io/

Redis installation directions (http://redis.io/topics/quickstart):

1. wget http://download.redis.io/releases/redis-2.8.5.tar.gz
   other versions available here, (http://redis.io/download)
2. tar xzf redis-2.8.5.tar.gz
3. cd redis-2.8.5.tar.gz
4. make
5. run with src/redis-server

For configuration details, see http://redis.io/topics/config

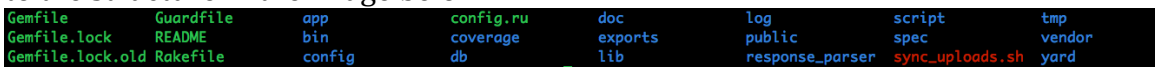
Build Requirements:

Build Server

A build server, independent of existing host infrastructure, should be used to pull updates of the deployed code from the source controlled code repository. The code repository will be the HHS Github locations established for this project. Since there is no compilation process for Ruby, the resulting code pulled from the Github is ready for deployment. This code may then be compressed and transferred to the development/staging/production environment so that those environments may be kept secure from outside connections.

Private Githubs:
https://github.com/HHS/voc-public.git
https://github.com/HHS/voc-admin.git

The file structure on the Application Server in each environment should look similar to the structure in the image below.



The public folder contains all of the html and html support static files, which should be pushed to the front facing web server/CDN.

The build server can most easily constructed on top of any of the modern flavors of Linux.

Deployment Requirements:

Configuration

RVM should have a configuration file for each project directory generated by the command line:

rvm use ruby-1.9.3-p484 --rvmrc

Validation can be done by switching to the project directory in the terminal and reading the ruby version report string output upon the directory switch.

Once the Github code tree is deployed to Application Server, run the bundle install from the root of the project directory to install the required gems via the command line:

bundle install –local

In the VOC/config directory the files database.yml, mongoid.yml, secret_settings.yml, and app_config.yml should be modified in the following ways. (The Github repository will have *.example files for reference and syntax requirements.)

a. database.yml – Change the User/Database connection information
b. mongoid.yml – Change the user/database connection information
c. secret_settings.yml – Uncomment (delete the hash symbol) and generate a new secret on the command line. This is used to hash cookies for users.
d. app_config.yml – Set the host url to target of the ajax calls for each distinct environment dev/staging/production. Set the redis server location and port; along with expire value (86400 recommended). Branding information can be applied here, defining a custom header/footer file as well as a custom CSS file.

Configure the Web servers. VOC consists of two distinct applications, each of which should have its own virtual host section. These virtual host sections will segment these applications away from existing and future Ruby applications running on the same server. As VOC will be hosted using Phusion Passenger, indicate the rvm and ruby info for Phusion Passenger in the same way as other Ruby on Rails applications within the virtual host section.

Example VirtualHost Config Assuming Passenger 4.0.0 or greater:

```
<VirtualHost *:80>
 ServerName voc.example.gov
 DocumentRoot "/webapps/voc-app/public"
```

```
 RailsEnv production
 PassengerRuby /Users/rubyapp/.rvm/wrappers/ruby-1.9.3-p448@comment/ruby
</VirtualHost>
```

If multiple app servers are needed, they can be configured independently and placed behind a load balancing solution.

Run the initialization rake tasks in a single command line:

rake db:create db:migrate db:seed

Assuming the settings in database.yml are correct rake will create the database, load the schema, and populate with the required seed data.

Run the initialization rake tasks in a single command line with

RAILS_ENV=production, rake db:create db:migrate db:seed

Assuming the settings in database.yml are correct rake will create the database, load the schema, and populate with the required seed data.

Touch the /tmp/restart.txt file to signal to Phusion Passenger to restart.

Further Reading:

The VOC User Guide will tell them how to create a survey. All surveys will have a survey ID associated with them.