

Hardware recommendations:

- 2 CPUs
- 4gb of memory
- 50gb of hard disk space (things to consider: # of documents per service request and size of mysql instance)

Software requirements:

- apache (sometimes referred to as httpd)
- apache-devel (sometimes referred to as httpd-devel)
- curl-devel
- mysql
- ruby-1.9.3-p286
- rubygems

From a brand new server:

- Install Apache
- Install MySQL
- Install the dependencies for Ruby:
 - build-essential vim git-core curl bison openssl libreadline6
 - libreadline6-dev zlib1g zlib1g-dev libssl-dev libyaml-dev
 - libxml2-dev libxslt-dev autoconf libc6-dev ncurses-dev
 - libcurl4-openssl-dev libopenssl-ruby apache2-prefork-dev
 - libapr1-dev libaprutil1-dev libx11-dev libffi-dev tcl-dev tk-dev
- Install Ruby 1.9.3p286 either from source or by using RVM (instructions for installing RVM can be found at rvm.io)
- Verify that ruby is the correct version, `ruby -version` should return 'ruby 1.9.3p286'
- Clone the git repository to a location on the server (`/var/www/rails` is a good location)
- Set the application configurations to your liking (see `doc/multi_institutional_documentation.md` for instructions)
- Depending on the environment that is configured (staging/production/testing/etc) go into the application folder and run
 - `RAILS_ENV=environment_chosen bundle install`
 - `RAILS_ENV=environment_chosen rake db:create`
 - `RAILS_ENV=environment_chosen rake db:migrate`
- Install the phusion passenger gem: `gem install passenger`
- Install the passenger apache module: `passenger-install-apache2-module`
- Follow the instructions provided by the passenger apache module install script.
- Create a virtualhost entry for the application which points to the public folder (so if your app is at `/var/www/rails/sparc-rails` then the

DocumentRoot and Directory should be /var/www/rails/sparc-rails/public)

- You will need to set:
 - Options -MultiViews
- And (depending on your settings) you may need to set:
 - AllowOverride all
 - Allow from all
- Depending on your settings you may need to chmod the application directory (I had to chmod -R 777 the directory to get write permissions for log and cache files to work with apache)
- Restart apache

The application can easily be setup to deploy using Capistrano or any other automated deployment method. RVM can also be used to manage the ruby versions used. See doc/musc_installation_example.txt for an example.

Once the Installation is complete you will need to create your initial institution and user. To do this follow these steps:

- While in the application folder (if the examples here were followed then /var/www/rails/sparc-rails) run the command:
RAILS_ENV=chosen_environment rails console (where chosen_environment is the environment whose database you want to access)
- Once in the console enter: require './lib/initial_cm_creation', then enter: 'run_initial_setup' and follow the instructions.
- NOTE: This tool should only be used to set up your initial institution and user, as it creates users with admin privileges.

Once you have created an initial institution and set yourself as a catalog manager, you will be able to access the catalog manager part of the application and begin inputting your organizational structure and service offerings. This tool can be found at <root_url>/catalog_manager. In order to access the other administration tool, the admin portal (found at <root_url>/portal/admin) you will need to either be a service provider or a super user for the organizational level you need admin control over in order to gain access to the portal. These rights can be granted in the catalog manager under the 'Service Providers' and 'Super Users' sections for each organizational entity.