ANZNN Installation and Deployment Guide

Version 1.1 - 2012-07-12 1530

Background

The preferred environment for hosting the web application is:

- A Linux server, preferably RHEL/CentOS 6.x or later (5.x will also work), with the standard "Server" packages installed.
- 4GB of memory.
- A MySQL (5.x or later) installation (on the same or different server).

The recommended method of deployment is to have a local copy of the application, and to use Capistrano to deploy from here to the target server. This provides many benefits and makes subsequent updates much easier.

These instructions are tailored to and tested on RHEL/CentOS 6.x (Specifically, CentOS 6.2), however there should be very little effort required to successfully deploy on EL 5.x distros (including Fedora).

From hereon in, RHEL/CentOS 5.x will be referred to as el5, similarly RHEL/CentOS 6.x as el6

Installation Instructions

Assumed preparation

- You have el5/6 or later installed with the standard "Server" packages.
- The EPEL yum repository is installed.
    - see http://fedoraproject.org/wiki/EPEL/FAQ#How_can_I_install_the_packages_from_the_EPEL_software_repository.3F
- Postfix (or equivalent MTA), or else an SMTP server that will accept connections from this server.
- Disable selinux, unless you are willing to add all the policy allow rules for the application. The list of rules is not covered here. Use a tool like audit2allow to obtain this once you have setup the application.
    - To quickly disable it, see http://www.electrictoolbox.com/switch-off-selinux-centos-5/
- The firewall should allow incoming **tcp** traffic to ssh (22), http (80) and https (443) ports.
    - In el5, use `system-config-securitylevel-tui`
    - In el6 this is now `system-config-firewall-tui`
    - You may need to run `yum install system-config-securitylevel-tui` first

Step 1: Local Environment Preparation

You will want to have a Linux VM handy that can be used for deployment (not the server!)

- We recommend using a VM as it is portable and doesn't have to be left running
- You can deploy from basically any Unix-based system, however these instructions only cover el5/6 and MacOS

- These instructions have been tested on MacOS 10.6.8 and CentOS 6.2 (the latest currently available).

Tips:

- Read each block of instructions first before copying and pasting.
- Ensure you are logged in as the correct user@host for each command
- If you don't have convenient access to a local VM, we recommend using the cloud (NeCTAR, Rackspace, Amazon EC2 etc)
- If you're running the VM locally, include the VM image in your backups once you are done to save time in the future
- SSH into your VM so you have access to the local clipboard
- Make use of snapshots in your VM host
- This VM can be used to redeploy other rails apps that utilise Capistrano unless there is a major OS package conflict.
    - Again, snapshot before and after

Create a user account with sudo privileges

You should never deploy (or do any other day-to-day activity) as root!

```
root@deployvm # useradd deploy
root@deployvm # passwd deploy
```

Give that user sudo powers

```
root@deployvm # visudo
```

Add this line. it doesn't matter where, but if you want to keep it tidy then under 'root ALL=(ALL) ALL'

- press 'i' to insert text, 'ESC' to exit insert mode, and SHIFT+ZZ to save and quit.

```
deploy      ALL=(ALL)      ALL
```

Switch to your new user

```
su - deploy
```

Install the following packages:

el5 & el6:

- These can all be installed automatically in one go with Yum:

```
deploy@deployvm $ sudo yum install -y gcc gcc-c++ patch readline readline-devel zlib zlib-devel
libyaml-devel libffi-devel openssl openssl-devel make bzip2 autoconf automake libtool bison
httpd httpd-devel apr-devel apr-util-devel mod_ssl mod_xsendfile curl curl-devel openssl
openssl-devel tzdata libxml2 libxml2-devel libxslt libxslt-devel sqlite-devel git mysql-server
mysql mysql-devel
```

MacOS:

- Command Line Tools for Xcode http://developer.apple.com/downloads** Requires a (free) Apple Developer ID https://developer.apple.com/programs/register/** Not required nor recommended if you have Xcode installed.
- Homebrew - http://mxcl.github.com/homebrew/** If you have an old version of Xcode installed, Homebrew may recommend you upgrade. This *probably* won't make a difference, but if you have trouble building Ruby or Postgres, this is where to start troubleshooting.
- MySQL
    - Do not use the downloadable version from the website.
    - After installing homebrew, run:

```
user@mac $ brew install mysql
```

Install RVM and the appropriate ruby

NB: RVM is constantly evolving and this step may change. If you have trouble, follow the instructions at https://rvm.io/rvm/install/

```
deploy@deployvm $ curl -L http://get.rvm.io | bash -s stable --ruby=1.9.3-p194
deploy@deployvm $ source ~/.rvm/scripts/rvm
```

Download the source repository

```
deploy@deployvm $ git clone https://github.com/IntersectAustralia/anznn.git
```

Set up gemset on the local machine

RVM will ask you to trust the .rvmrc file.
We recommend that you glance over any new rvmrc quickly before accepting it.

```
deploy@deployvm $ cd anznn
```

Ensure that you are in the correct gemset

```
deploy@deployvm $ rvm current
ruby-1.9.3-p194@anznn
```

install bundler and the required environment for anznn

```
deploy@deployvm $ gem install bundler
deploy@deployvm $ bundle install
```

Configure your git identity if you haven't already

Global config:

```
deploy@deployvm $ git config --global user.email "you@example.com"
deploy@deployvm $ git config --global user.name "Your Name"
```

Just for ANZNN:

```
deploy@deployvm $ pwd
/home/user/anznn
deploy@deployvm $ git config user.email "you@example.com"
deploy@deployvm $ git config user.name "Your Name"
```

Step 2: Server Preparation

SSH into your server

- You probably want to use a new tab on your local computer rather than from the deploy VM because you will be switching between them a lot

Create a user for application to run as, e.g. 'dc21'

- Follow the same steps you used on your deploy vm to create a user:

```
root@server # useradd anznn
root@server # passwd anznn
```

Give that user sudo powers

```
root@server # visudo
```

Add this line. it doesn't matter where, but if you want to keep it tidy then under 'root ALL=(ALL) ALL'

- press 'i' to insert text, 'ESC' to exit insert mode, and SHIFT+ZZ to save and quit.

```
anznn      ALL=(ALL)      ALL
```

Step 3: Deploy Preparation

This section is likely to change in the future

On your local machine:

These commands should all be executed from within the repository that you just checked out, i.e.:

```
deploy@deployvm $ pwd
/home/user/anznn
```

The codebase has configuration files with some default settings that need to altered.

In `config/deploy/production.rb`

- Supply your server's FQDN (hostname)
  - This will also be used to configure an Apache virtualhost
  - If you would not normally access the server via the name you supply here, you will need to manually edit /etc/httpd/conf.d/rails-anznn.conf
- Set the el6 flag to false if using el5 (approx line 16)
- If you have a proxy server, fill in the details here.
  - If you have no proxy, leave it as 'nil'
  - The url should look like http://user:pass@proxy.example.com:8080
  - or without a username/pass: http://proxy.example.com:8080

```
# Your HTTP server, Apache/etc
set :web_server, 'hostname.com'
# This may be the same as your Web server
set :app_server, 'hostname.com'
# This is where Rails migrations will run
set :db_server, 'hostname.com'
# The user configured to run the rails app
set :user, 'anznn'

# If you are using RHEL/CentOS 6 or later, set this to true
set :el6, true

# If you have a proxy server, enter the address here including "inverted commas", eg:
#set :proxy, "http://user:pass@proxy.example.com:8080" # with a user/password
#set :proxy, "http://proxy.example.com:8080" # without a user/pass
set :proxy, nil
```

In `deploy_templates/config/anznn_config.yml` edit the values under the 'defaults' section.
You will likely want to change the following:

- password_reset_email_sender (The FROM: address in password reset emails)
- account_request_user_status_email_sender (The FROM: address in 'You have been approved/rejected' emails)
- account_request_admin_notification_sender (The FROM: address in 'Dear Admin, a new account has been requested' emails).
  - This address may be a 'noreply' address
  - There is no reason why these cannot all be the same address, but the option for customisation is there.

The Batch file settings can be changed if required, but the defaults should suffice.

In `deploy_templates/config/initializers/devise.rb` edit the line for the sender email address (around line 21):

```
config.mailer_sender = "noreply@HOSTNAME"
```

Apply proxy settings to the server (if you have a proxy)

In `config/deploy/production.rb` supply your server's FQDN (hostname):
NB: This file is *not* in the deploy_templates directory

```
deploy@deployvm $ cap production server_setup:set_proxies
```

Run initial setup:

You will need to supply the password you used to set up the **anznn** user on your **server**

```
deploy@deployvm $ cap production deploy:setup
```

Then on the server

Change the MySQL root password, create the database and a user.

- This user is being given all privileges.
- If the server is being used by other applications you will want to set more restrictive permissions
- Change the root and anznn user passwords ('RootPassword'/'ANZNNPassword' respectively in this example) to something else more secure
- Use a different password to the actual root / anznn user passwords

```
root@server # mysql -u root -p
Your MySQL connection id is 3
 --- snip ---
Welcome to the MySQL monitor.  Commands end with ; or \g.
mysql> use mysql;
mysql> update user set password=PASSWORD("RootPassword") where User='root';
mysql> CREATE USER 'anznn'@'localhost' IDENTIFIED BY 'anznn';
Query OK, 0 rows affected (0.00 sec)

mysql> grant all on *.* to 'anznn'@'localhost' identified by 'ANZNNPassword';
Query OK, 0 rows affected (0.00 sec)

mysql> SHOW GRANTS FOR 'anznn';
+-------------------------------------------------------------------------------------+
| Grants for anznn@%                                                                  |
+-------------------------------------------------------------------------------------+
| GRANT ALL PRIVILEGES ON *.* TO 'anznn'@'%' IDENTIFIED BY PASSWORD '293733bd176c1f46' |
+-------------------------------------------------------------------------------------+
1 row in set (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
```

Enable services at boot

```
root@server # /sbin/chkconfig mysql on
root@server # /sbin/chkconfig httpd on
```

Step 4: Deploy the app

This all happens from your local instance

Deploy the code

- When running each of these tasks, you will first be prompted for the sudo password for your anznn user
- When running deploy:update, at the `Database Password:` prompt enter the password used to connect to the database
    - If deployment stalls running 'bundle:install' in deploy:update, this is likely due to a misconfigured proxy
    - The quick-fix solution is to copy the command in yellow and run it directly on the server as the dc21 user while the deploy is hung, then restart the deploy.

```
deploy@deployvm $ cap production deploy:update
deploy@deployvm $ cap production deploy:schema_load
deploy@deployvm $ cap production deploy:seed
```

Create an initial admin user:

```
deploy@deployvm $ cap production deploy:generate_initial_user
```

Reminder: you should change the password when you first login.

Step 5: Start the server

```
root@server # service httpd start
Starting httpd:                                [  OK  ]
```

You may see this warning:

```
httpd: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1 for
ServerName
```

For an initial deploy, that's fine, however before the server goes live you should fix a couple of things in `/etc/httpd/conf/httpd.conf`

```
ServerAdmin you@example.com
ServerName www.example.com:80
```

- More information can be found within `/etc/httpd/conf/httpd.conf`

Step 6: Post Installation

Check that you can successfully visit the site at the hostname or IP you configured.

If you have problems you can check both the Apache logs and also web application logs in the logs directory under the anznnapp directory (e.g. under /home/anznn/anznn/shared/log).

Updating The Application

These commands should all be run from your checked-out local instance

```
deploy@deployvm $ pwd
/home/deploy/anznn
```

Update your copy of the repo:

```
deploy@deployvm $ git stash
deploy@deployvm $ git pull
deploy@deployvm $ git stash pop
```

There may be some changes to the templated files which result in merge conflicts. resolve them as necessary.
Check over all of the files in the deploy_templates directory, making any changes as required.

Rebundle on your local machine

```
deploy@deployvm $ bundle install
```

Redeploy the code

- You will first be prompted for the sudo password for your anznn user
- The second prompt will be to enter the password used to connect to the database

```
user@local $ cap production deploy:update
user@local $ cap production deploy:migrate
user@local $ cap production deploy:restart
```