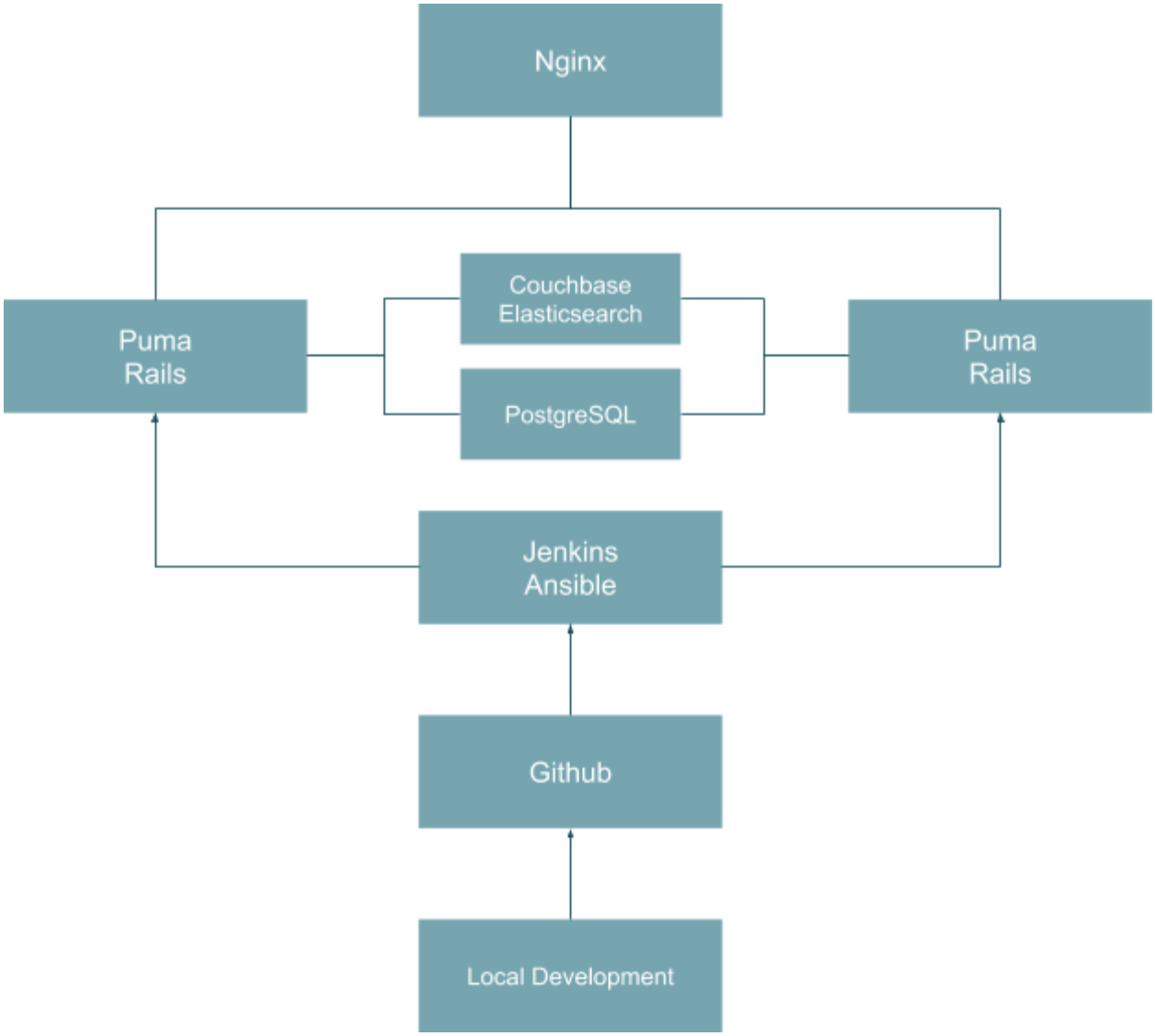
Deployment Architecture v0.2

StoryWeaver platform deployment

Author	Version/Date	Change
Manoj Sukhavasi	**v0.1** 9th Feb 2018	First draft
Manoj Sukhavasi	**V0.2** 12th Feb 2018	Added detailed installation steps for development server.

Table of Contents

Architecture Diagram



The above diagram shows how our current setup is configured.

Jenkins & Ansible:

Jenkins helps to automate some parts of software development process, with continuous integration and facilitating the process of continuous delivery. We use Jenkins to automate the process of testing, building and deploying the code to servers. We manually configure Jenkins to do various tasks for us. We currently use jenkins version 1.656. Install the jenkins thought the following steps :

```
wget -q -O - https://pkg.jenkins.io/debian/jenkins-ci.org.key | sudo apt-key add -
echo deb https://pkg.jenkins.io/debian-stable binary/ | sudo tee
/etc/apt/sources.list.d/jenkins.list
sudo apt-get update
sudo apt-get install jenkins
```

Ansible is software that automates software provisioning, configuration management, and application deployment. We use Ansible to setup new servers or deploy the code to the existing production servers. Scripts for the creating a new server are located at https://github.com/PrathamBooks/spp-delivery . Based on the needs of your server you need to modify the existing scripts to setup a new server.
Steps to create a new server :

- Add a new server config to the ansible/group_vars. Setup the config details for ip, port, API keys as in sample file ansible/group_vars/app2-e2e
- Add credentials for the server access in ansible/inventories. Add ssh keys to the server as in sample file ansible/inventories/production
- Edit the ansible/production.yml file to specify the config, roles needed .
- Run the "ansible-playbook ./ansible/production.yml -vvvv -i ./ansible/inventories/production". Modify the command to add your production.yml file and inventory you modified.

Scripts for deploying code to production servers are located at
https://github.com/PrathamBooks/spp-deploy .

Nginx:

Nginx is a is a web server which can also be used as a reverse proxy, load balancer and HTTP cache. This is first point of contact for a request in our setup. Nginx does load-balancing of multiple servers(currently we have 2 main servers) and serves static assets. To create a new server with the Nginx on it, follow the scripts in spp-delivery as mentioned above and following steps :

- Add new server config details and credentials as mentioned above.
- Edit the roles section in the ./ansible/production.yml(or your own file) to include only nginx and comment out the pre-tasks.
- Any further modification to Nginx installation can be made in the roles/nginx

Puma:

Puma is an application server that enables our Rails application to process requests concurrently. As Puma is not designed to be accessed by users directly, we will use Nginx as a reverse proxy that will buffer requests and responses between users and our Rails application. You can think of Rack as a common language that Ruby web frameworks (like Rails) and app servers both speak. Because each side knows the same language, it means Rails can talk to Puma and Puma to Rails, without having either Rails or Puma know anything about the other. Puma is generally installed within Rails server. To install a Rails server and Puma follow the instructions in spp-delivery and following steps:

- Add new server config details and credentials as mentioned above.
- Edit the roles section in the ./ansible/production.yml(or your own file) to include only utils, rvm1, vsftpd, phantom, newrelic and un-comment the pre-tasks if commented out.

Couchbase:

Couchbase is multi-modal NoSQL database. We use Couchbase for caching our pages and search queries. We currently setup Elasticsearch and Couchbase in one server. To install these, follow the guidelines as mentioned above in spp-delivery and following instructions:

- Add new server config details and credentials as mentioned above.
- Edit the roles section in the ./ansible/production.yml(or your own file) to include only couchbase and elasticsearch and comment out the pre-tasks.
- Any further modification to couchbase/elasticsearch installation can be made in their respective directories in ./roles.

Elasticsearch:

Elasticsearch is a full-text search engine with an HTTP web interface and schema-free JSON documents. We use this for search functionality in our website. This allows us for faster retrieval of the data. We currently setup Elasticsearch and Couchbase in one server. To install these, follow the guidelines as mentioned above in couchbase section.

PostgreSQL:

PostgreSQL is an object-relational database management system . This is our main database. We have a seperate server dedicated to this. To setup a new server, follow the guidelines as mentioned in spp-delivery and following guidelines :

- Add new server config details and credentials as mentioned above.
- Edit the roles section in the ./ansible/production.yml(or your own file) to include only postgres and comment out the pre-tasks.
- Any further modification to postgres installation can be made in the roles/postgres directory.

Development installation guide

Following steps detail out the instructions for installing Storyweaver application and running a local server.

Requirements

You need to have an 8GB system and Ubuntu 14.04.

Install RVM, it is compatible for bash shell only

```
$ gpg --keyserver hkp://keys.gnupg.net --recv-keys
409B6B1796C275462A1703113804BB82D39DC0E3 7D2BAF1CF37B13E2069D6956105BD0E739499BDB

$ \curl -sSL https://get.rvm.io | bash -s stable --path
<custom_path_for_installation>
```

Install Ruby 2.1.4

```
$ rvm install 2.1.4
$ rvm use 2.1.4 --default
```

Install PostgreSQL

```
$ sudo apt-get install postgresql postgresql-contrib
```

PostgreSQL configuration

Create *spp_user* and grant access to it,

```
$ sudo -i
$ su - postgres
$ psql
psql (9.3.17)
```

```
Type "help" for help.
postgres=# CREATE ROLE spp_user superuser;
postgres=# ALTER ROLE spp_user WITH LOGIN;
postgres=# GRANT ALL PRIVILEGES ON ALL FUNCTIONS IN SCHEMA public TO spp_user;
postgres=# GRANT ALL ON ALL FUNCTIONS IN SCHEMA public TO spp_user;
postgres=# GRANT ALL ON ALL SEQUENCES IN SCHEMA public TO spp_user;
postgres=# GRANT ALL ON ALL TABLES IN SCHEMA public TO spp_user;
postgres=# \q
```

For password less login to database, Open the below file,

```
$ sudo vim /etc/postgresql/9.3/main/pg_hba.conf
```

(or)

```
$ sudo vim /etc/postgresql/<version>/main/pg_hba.conf
```

Add the following line (in green) just before below line in the pg_hba.conf file

```
# "local" is for Unix domain socket connections only
+ local   all             spp_user                                trust
  local   all             all                                     peer
```

Restart postgres,

```
$ sudo service postgresql restart
```

Install imagemagick

```
$ sudo apt-get install imagemagick
```

Install wkhtmltopdf

```
$ sudo add-apt-repository ppa:ecometrica/servers
$ sudo apt-get install wkhtmltopdf
$ sudo ln /usr/bin/wkhtmltopdf /usr/local/bin/wkhtmltopdf
```

Install Java 8

This is required for Elasticsearch installation

```
$ sudo add-apt-repository ppa:webupd8team/java
$ sudo apt-get update
$ sudo apt-get install oracle-java8-installer
```

Install elasticsearch version 5.2.0 or above

- https://www.elastic.co/downloads/elasticsearch
- Download the Debian Package file
- Go to the download directory and run,

```
$ sudo dpkg -i elasticsearch-<version>.deb
```

Rails Install Steps

Clone repo

```
$ git clone https://github.com/PrathamBooks/spp
```

Install bundler

```
$ sudo apt-get install bundler
$ gem install bundler
```

Rest of the following commands needs to to be executed in project directory spp.

Install gems

```
$ bundle install
```

Create db

For developers

```
$ bundle exec rake db:create
$ bundle exec rake db:migrate
$ bundle exec rake db:seed
$ bundle exec rake db:seed:development:users
```

For non-developers

```
$ bundle exec rake db:create
```

```
Restore the database from a database dump provided(currently database dump is not public) with the following command.
```

```
$ psql spp < {Replace with database dump file}
```

Index search engine

```
$ bundle exec rake searchkick:reindex:all
```

rake jobs

```
$ bundle exec rake jobs:work
```

Start server

```
$ rails s
```

or

```
$ puma
```

Point your browser to http://localhost:3000

Known issues?

While installing gems via 'bundle install'

- Gem pg installation error while bundling, `Can't find the 'libpq-fe.h header` Solution:

```
$ sudo apt-get install libpq-dev
```

- `Can't install RMagick 2.13.4. Can't find MagickWand.h`, Solution:

```
$ sudo apt-get install libmagickwand-dev
```

- `*You must install libcouchbase >= 2.4.0`

```
$ wget
http://packages.couchbase.com/releases/couchbase-release/couchbase-release-1.0-3-amd6
4.deb
$ sudo dpkg -i couchbase-release-1.0-3-amd64.deb
$ sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 6EF1EAC479CF7903
$ sudo apt-get update
$ sudo apt-get install libcouchbase-dev libcouchbase2-bin build-essential
```

Installing React environment:

Install nodejs and npm

```
$ curl -sL https://deb.nodesource.com/setup_8.x | sudo -E bash -
$ sudo apt-get install nodejs
```

Clone the Repo sw-js

```
$ git clone https://github.com/PrathamBooks/sw-js.git
```

Install yarn

Move to the sw-js directory

```
$ npm install -g yarn
```

Integrate React with Rails

```
$ REACT_PATH=/scratch/sw-js RAILS_PATH=/scratch/spp bash
/scratch/spp/lib/scripts/integrate_rails_with_react.sh
```

Update the above paths with your custom paths.

Move to the Rails directory(spp) and run '`rails server -b 0.0.0.0`'.