Deployment

Introduction

As we currently do not use a containerized solution, we need to release the code in the homolog and prod environment by sending the code using the rsync or git pull command.

This requires the user to have access permission on the servers to run the command.

Setup

In this section, we will carry out the first configuration of the system.

Systemctl

First of all, we must upload deployment/github_server.service to our server.
It must be saved in a directory that systemctl can access and manage, example: /lib/systemd/system
After that, we need to configure the user and directory where the source code was uploaded.

Source code

We currently do not use a containerized solution. So we need to send the code to the server via rsync or git pull command.
If you choose to use rsync, you will need to be careful with the version you are sending to the machine so as not to place an invalid version in the environment. You can create an alias for this.

Example: alias github_sync_dev='git branch master; rsync -azPr --exclude .idea --exclude Gemfile.lock --exclude Gemfile --exclude .git/ ...'

Packages

It is important to check if the packages below are installed, either via rvm / bundler or system packages.

- Ruby 3.0
- PostgreSQL

Gems

- ruby-sinatra
- ruby-octokit
- ruby-netrc
- ruby-jwt
- ruby-json
- ruby-activerecord
- puma

Bamboo CI

You must create a file called .netrc and configure the GITHUB-APP key with the login (github app id) and password.

The password is used to create a SHA256 signature that we use to send events to the server.

Database

Now we need to configure config/database.yml to allow the application to access the database and create the necessary tables.

To do this, just copy the database_template.yml file to config/database.yml and configure the file with the correct data.

Rakefile

If the database does not exist, run the following command:
RACK_ENV=test rake db:create

This will create a new database for the test environment you need to run the command for the environment you want, the name must match the config/database.yml file.
To run the migrations just run the command
RACK_ENV=test rake db:migrate

General Settings

In the root of the project there is a file called config.yml, you need to copy the config_template.yml file to config.yml and configure the data.

```
debug:

port:

auth_signature:
 password:

github_app:
 - login:
   cert:
```

About file:

- debug - true or false. Allows system logs to be passed to debug. **When modified, you do not need to restart the system.**
- port - Port number the system will run.
- auth_signature - Key used to generate a SHA256 to validate data submitted by GitHub or CI.
- Github_app - Login would be the GitHub App ID and cert is the location of the certificate (.pem) generated by GitHub.

Plan

In the database, we have a table called Plan, where we relate a GitHub repository with a Bamboo CI execution plan.
We need to configure it according to the environment to run the plan otherwise it will run a default plan (TESTING-FRRCRAS)