

Foreman - Bug #1367

Ensure all gems are defined in Bundler Gemfile

12/04/2011 04:21 AM - Ohad Levy

Status: Closed	
Priority: Normal	
Assignee:	
Category: Rails	
Target version: 1.0	
Difficulty:	Fixed in Releases:
Triaged:	Found in Releases:
Bugzilla link:	Red Hat JIRA:
Pull request:	
Description	
Currently some dependencies of Foreman, especially gems for database access like `pg` or `mysql`, are missing from our `Gemfile` because we have no good way of handling them selectively.	
Related issues:	
Related to Foreman - Bug #1382: no such file to load -- pg	Duplicate 12/07/2011
Related to Foreman - Bug #1380: /has_many_polymorphs.git (at master) is not c...	Closed 12/07/2011

Associated revisions

Revision e6d6111b - 01/03/2012 10:37 AM - Ohad Levy

refs #1367 - a better version of our gemfiles, not perfect yet.

History

#1 - 12/29/2011 09:55 AM - Jochen Schalanda

- Subject changed from ensure all gems are defined in bundler gemlock file to Ensure all gems are defined in Bundler Gemfile

Short relevant discussion on <https://github.com/ohadlevy/foreman/pull/23>

Comment

ok, a few points then:

1. we cant ship bundle lock file in the app
2. we either need to manage the config file in the package, or run in post section of each db package bundle --without

what do you think?

From the perspective of creating Debian packages for Foreman the "database gems" (`mysql`, `pg`, `sqlite3`) don't have to be listed in the `Gemfile` at all. They're all available as packages in the Debian-based distributions we're supporting.

These three gems are special in the way that they need additional packages or header files installed on the system and they are bound to the respective Ruby version they've been installed for since they contain binary extensions.

But of course it's not a good practice to have an incomplete manifest of dependencies (which `Gemfile` basically is) for any other user which would want to install Foreman from source or from the git repository.

In my opinion the best option currently is having separate groups for these dependencies (like the [pull request](#) introduced) plus adding Bundler's `.bundle/config` file in which these groups are excluded.

For example if a user wants to use `sqlite3` she can run `bundle install --without mysql postgresql`. A simple `bundle install` without additional parameters would install neither `mysql`, `postgresql`, nor `sqlite3`.

#2 - 12/29/2011 10:15 AM - Ohad Levy

Jochen Schalanda wrote:

From the perspective of creating Debian packages for Foreman the "database gems" (`mysql`, `pg`, `sqlite3`) don't have to be listed in the `Gemfile` at all. They're all available as packages in the Debian-based distributions we're supporting.

True, but bundler has two responsibilities, it download and require the gem, so AFIAK, if its not in the Gemfile, then its not loaded at all.

These three gems are special in the way that they need additional packages or header files installed on the system and they are bound to the respective Ruby version they've been installed for since they contain binary extensions.

But of course it's not a good practice to have an incomplete manifest of dependencies (which `Gemfile` basically is) for any other user which would want to install Foreman from source or from the git repository.

In my opinion the best option currently is having separate groups for these dependencies (like the [pull request](#) introduced) plus adding Bundler's `.bundle/config` file in which these groups are excluded.

For example if a user wants to use `sqlite3` she can run `bundle install --without mysql postgresql`. A simple `bundle install` without additional parameters would install neither `mysql`, `postgresql`, nor `sqlite3`.

I agree, what about removing the lock file from the repo?

#3 - 12/29/2011 11:03 AM - Jochen Schalanda

Ohad Levy wrote:

True, but bundler has two responsibilities, it download and require the gem, so AFIAK, if its not in the Gemfile, then its not loaded at all.

I have to check that again, but if a library is installed in Ruby's library path (not necessarily by RubyGems or intermediately by Bundler) it should be possible to load them by simply require-ing them. In case of Foreman these libraries are being loaded by ActiveRecord, aren't they?

I agree, what about removing the lock file from the repo?

I'm not sure about this. On the one hand having a working set of definitely working dependency versions is good, but on the other hand it might be a problem if the Ruby version or architecture changes. Currently I'm for keeping the Gemfile.lock in the repository.

#4 - 12/29/2011 01:03 PM - Ohad Levy

Jochen Schalanda wrote:

Ohad Levy wrote:

True, but bundler has two responsibilities, it download and require the gem, so AFIAK, if its not in the Gemfile, then its not loaded at all.

I have to check that again, but if a library is installed in Ruby's library path (not necessarily by RubyGems or intermediately by Bundler) it should be possible to load them by simply require-ing them. In case of Foreman these libraries are being loaded by ActiveRecord, aren't they?

Yes, but db gems are special imho, not sure who loads first.

I agree, what about removing the lock file from the repo?

I'm not sure about this. On the one hand having a working set of definitely working dependency versions is good, but on the other hand it might be a problem if the Ruby version or architecture changes. Currently I'm for keeping the Gemfile.lock in the repository.

Well, what about different versions of puppet, or other 3rd party gems that are already in various versions in the different distributions?

#5 - 06/03/2012 03:12 AM - Ohad Levy

- *Status changed from New to Closed*

this has been resolved by the various packages.