GMOペパボ

Roadmap for RubyGems 4 and Bundler 3

The integration for package ecosystem

Hiroshi SHIBATA / GMO Pepabo, Inc.

2020.04.13 Ruby3さみっと

GMOペパボ



Hiroshi SHIBATA @hsbt

Executive Officer VP of Engineering
Technical Director
at GMO Pepabo, Inc. @pepabo

https://www.hsbt.org

Agenda

- The introduction for RubyGems 3.1 and Bundler 2.1
- Recent Update for RubyGems and Bundler
- The feature plan and Roadmap for Ruby 3.0



What's new in RubyGems 3.1?

- I released RubyGems 3.1 at 16 Dec 2019
- https://blog.rubygems.org/2019/12/16/3.1.0-released.html
  - This version dropped the compatibility code of Ruby 1.8 and 1.9
  - Bundle Bundler 2.1
  - Multi-factor authentication for yank command



What's new in Bundler 2.1?

- Bundler 2.1 has been released at 15 Dec 2019.
- https://github.com/bundler/bundler/releases/tag/v2.1.4
  - There is no incompatible feature from Bundler 2.0.x(maybe…)
  - Cleanup the compatibility code for Ruby 1.8 and 1.9
  - Tuned on the deprecation warnings for Bundler 3



Tuned on the deprecation warnings for v3

```
Bundler::SharedHelpers.major_deprecation 2
```

- `bundle console`
- `bundle vis`
- `bundle inject`
- Bundler.environment → Bundler.load
- Bundler.clean_{env,exec} → Bundler.unbundled_{env,exec}
- Bundler.with_clean_env → Bundler.with_unbundled_env



The feature flag for Bundler 3

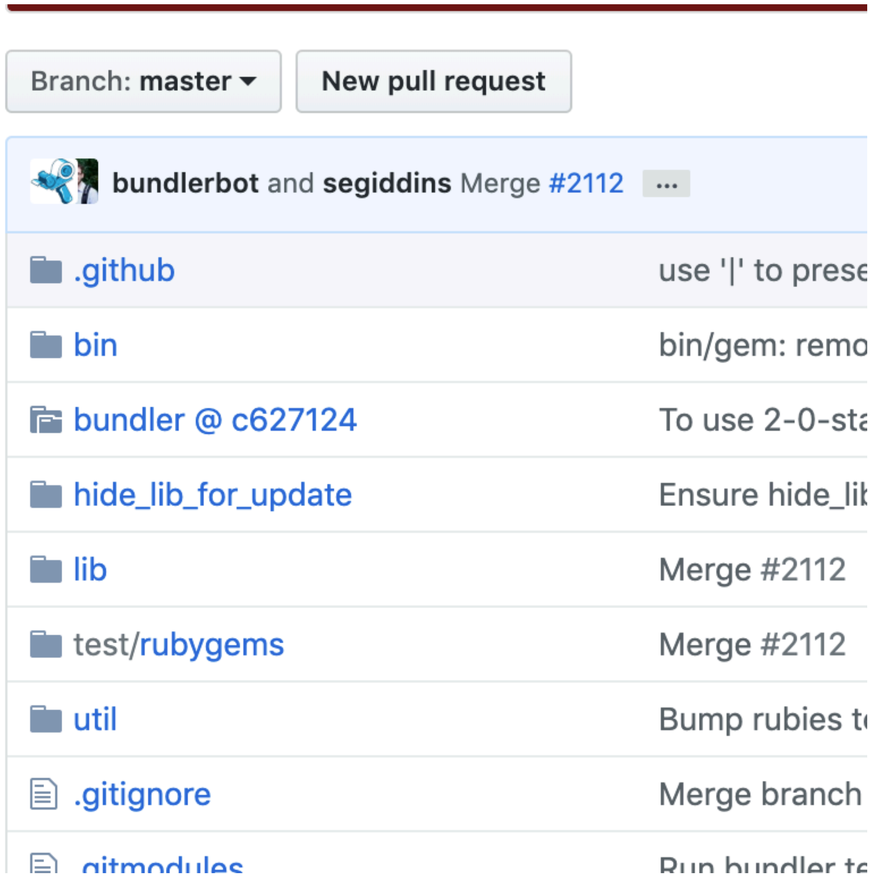
```
    settings_flag(:allow_bundler_dependency_conflicts) { bundler_3_mode? }
    settings_flag(:allow_offline_install) { bundler_3_mode? }
    settings_flag(:auto_clean_without_path) { bundler_3_mode? }
    settings_flag(:auto_config_jobs) { bundler_3_mode? }
    settings_flag(:cache_all) { bundler_3_mode? }
    settings_flag(:default_install_uses_path) { bundler_3_mode? }
    settings_flag(:deployment_means_frozen) { bundler_3_mode? }
    settings_flag(:disable_multisource) { bundler_3_mode? }
    settings_flag(:forget_cli_options) { bundler_3_mode? }
    settings_flag(:global_gem_cache) { bundler_3_mode? }
    settings_flag(:only_update_to_newer_versions) { bundler_3_mode? }
    settings_flag(:path_relative_to_cwd) { bundler_3_mode? }
    settings_flag(:plugins) { @bundler_version >= Gem::Version.new("1.14") }
    settings_flag(:print_only_version_number) { bundler_3_mode? }
    settings_flag(:setup_makes_kernel_gem_public) { !bundler_3_mode? }
    settings_flag(:skip_default_git_sources) { bundler_3_mode? }
    settings_flag(:specific_platform) { bundler_3_mode? }
    settings_flag(:suppress_install_using_messages) { bundler_3_mode? }
    settings_flag(:unlock_source_unlocks_spec) { !bundler_3_mode? }
    settings_flag(:update_requires_all_flag) { bundler_4_mode? }
    settings_flag(:use_gem_version_promoter_for_major_updates) { bundler_3_mode? }

    settings_option(:default_cli_command) { bundler_3_mode? ? :cli_help : :install }
```



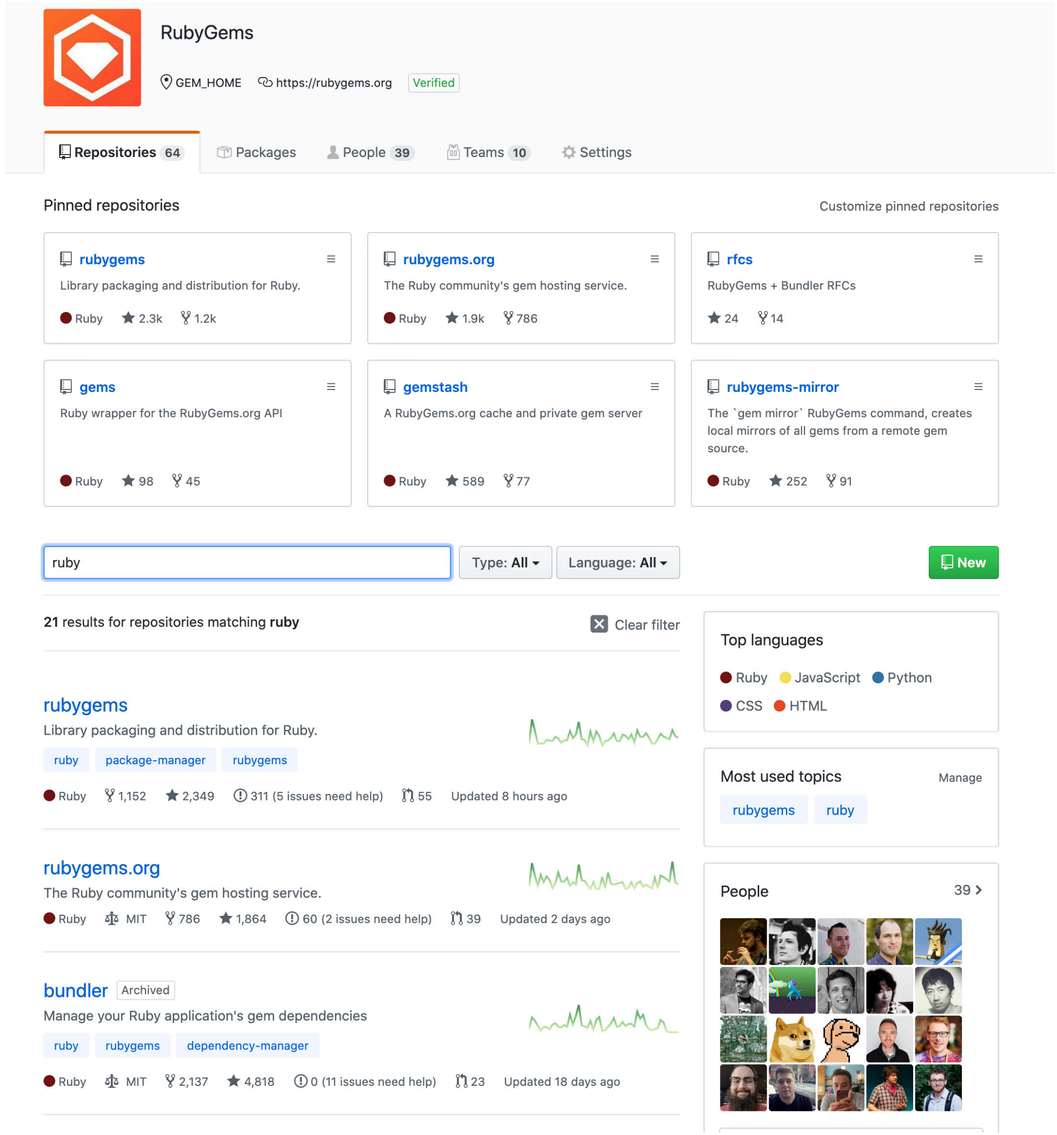
RubyGems/Bundler integration

- We put the bundler as submodule in rubygems repository in 2019



RubyGems/Bundler integration

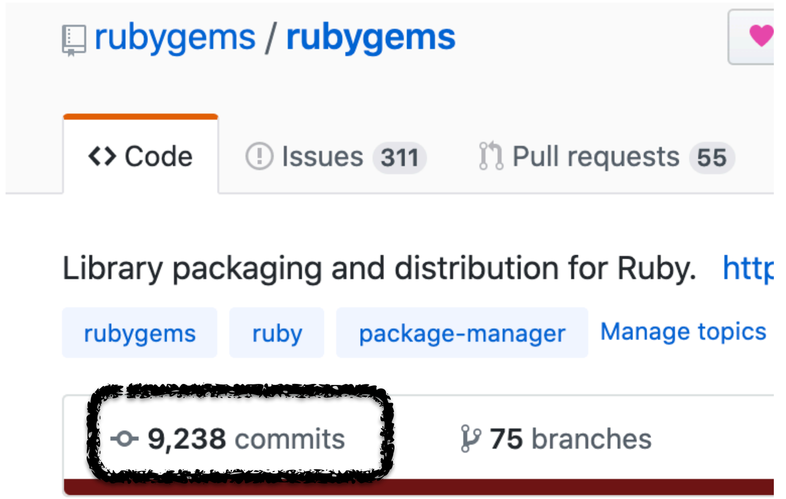
- We merged bundler org into rubygems org.
- The repository, team, maintainers was merged.



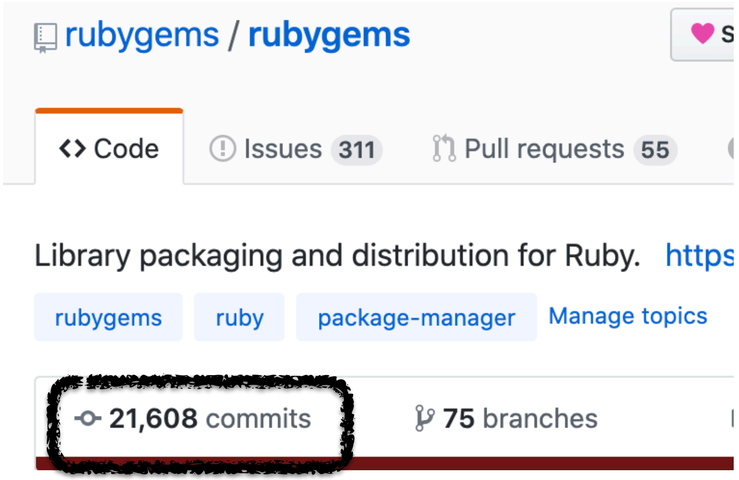
RubyGems/Bundler integration

- We also merged bundler repo into rubygems repo as monorepo.
- The commits, CI, mainainers documents was merged

Before



After



Finally, We start to merge the code-base

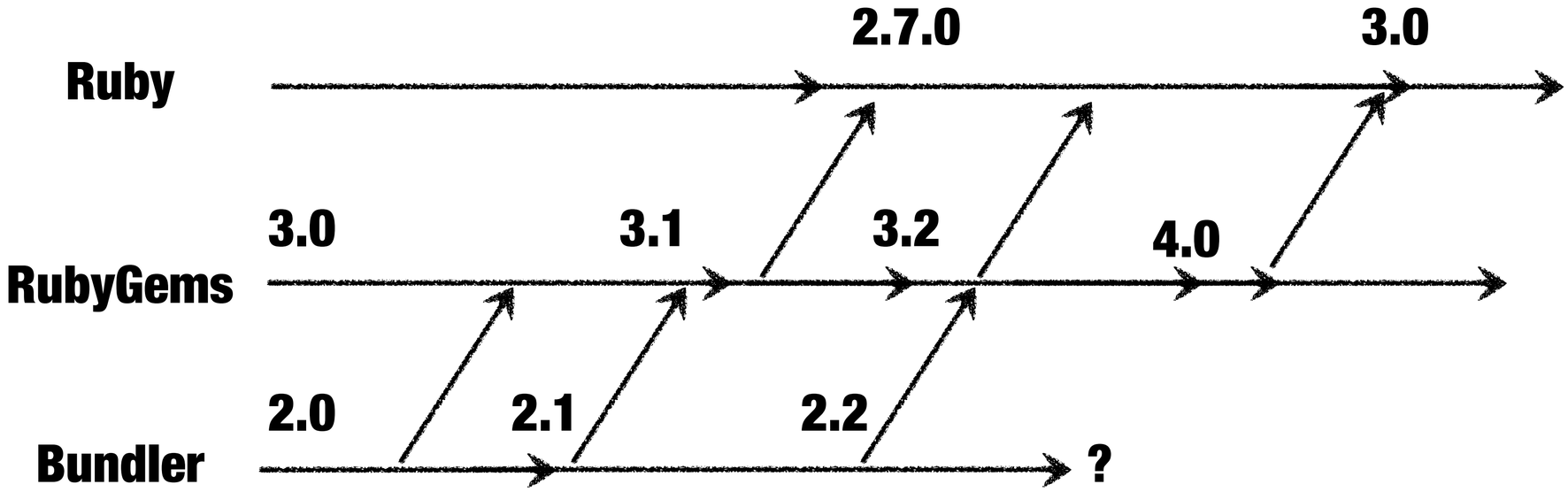
Feature works

- We still use separated release workflow about RubyGems and Bundler
- We hope to release simultaneously like this.
  - Release RubyGems 3.2 and Bundler 2.2
  - After that, RubyGems 3.2.1 and Bundler 2.2.1 and RubyGems 3.3 and Bundler 2.3
- Or, Bump Version to RubyGems 4.0 and Bundler 4.0



Bump up RubyGems/Bundler

- We will merge into RubyGems 3.2 into Ruby 2.8.0-dev. After that, RubyGems 4.0 will be merge Ruby 3.



Dependency Resolver incompatible

- RubyGems 2.x and 3.x uses Molinillo-0.5.7

```
~/D/g/r/rubygems (master) > ls lib/rubygems/resolver/molinillo/lib/molinillo
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
dependency_graph    errors.rb           modules             resolver.rb
```

- Bundler 1.x and 2.x also uses Molinillo-0.6.4

```
~/D/g/b/bundler (master) > ls lib/bundler/vendor/molinillo/lib/molinillo
compatibility.rb    dependency_graph    errors.rb           modules             resolver.rb
delegates           dependency_graph.rb gem_metadata.rb     resolution.rb       state.rb
```

- These are different versions and behavior of dependency resolver.



Gemification for Ruby 3

Gemification for Ruby 3.0

This matrix shows number of standard libraries and their classifications in Ruby 2.8/3.0.

	Standard Libraries	Default Gems	Bundled Gems
Pure Ruby	22	36	6
C extensions	12	17	0



Gemification for Ruby 3.0

- Notable Changes
  - `net-telnet` was removed from the bundled gems
  - `rexml` and `rss` promoted the bundled gems from the default gems.

- I'll work to publish the all of stdlibs to the default gems for Ruby 3.0
  - Except to use `internal.h` or `internal/*.h` and depends on Ruby version
  - mkmf.rb, objspace, pty, ripper, socket, coverage



Gamification on Ruby 3.0(TBD)

We will extract the standard libraries to the default gems.

```
Abbrev
Base64
DEBUGGER__
DRb
ERB
Find
OpenURI
OptionParser
PP
PrettyPrinter
resolv-replace.rb
Resolv
Rinda
SecureRandom
Set
Shellwords
Time
TSort
un.rb
Digest
IO
Monitor
NKF
Pathname
Syslog
WIN32OLE
```



Activation issues about default gems

- You couldn’t use the specified version of default gems like json when RubyGems/Bundler activated them.
- When rubygems uses json-2.1.0, You couldn’t use json 1.8.x. Because ruby gems and rubygems.org never uses JSON format.
- We can resolve it with `vendoring` approach. But json, psych, and openssl is C extension library.



Gamification on Ruby 3.0(TBD)

We will extract the default gems to the bundled gems without them.

```
English
base64
benchmark
cgi
digest
erb
etc
fileutils
find
io/console
json
monitor
net/http
net/https
openssl
optparse
pathname
pp
psych
rdoc
resolv
set
shellwords
stringio
strscan
tempfile
thread
time
timeout
tmpdir
tsort
uri
webrick
yaml
zlib
```



We should remove the dependency of stdlib from rubygems and bundler

Conclusion

- Finally, We will start to merge RubyGems and Bundler for unifying the one command.
- I’m working to extract stdlib to the default gems in Ruby3
- We should remove the current dependency from rubygems and bundler.

Let’s join to #rubygems in ruby-jp slack




See you in
Matsumoto