Carton 1.0
managing CPAN deps the right way

Tatsuhiko Miyagawa
@miyagawa
O'Reilly OSCON 2013
Me

- Tatsuhiko Miyagawa
- Lives in San Francisco
- {github,twitter,CPAN}/miyagawa
Managing CPAN Dependencies
How many CPAN modules your app depends on?
When is the last time upgrading a CPAN module broke your app?
Case Study: Web App Development
You’re writing a new web app using as many CPAN modules.
Get them from CPAN, install on your machine.
john@local> cpanm Web::Framework
installed LWP-5.912
installed Plack-0.9980
installed Web-Framework-1.10
Test it...
Works? Ship it!
What if ...
• Jul 2nd: Started working on project using Web::Framework 1.1
• Jul 9th: Finished version 1.0
• Jul 10-15th: internal beta, QA
• Jul 16th: Deploy to the production
• Jul 2nd: Started working on project
• using Web::Framework 1.1
• Jul 9th: Finished version 1.0
• Jul 10-15th: internal beta, QA
• Jul 15th: Web::Framework 1.2 is released
• Jul 16th: Deploy to the cloud/production
root@prod> cpanm Web::Framework
installed LWP-5.912
installed Plack-0.9980
installed Web-Framework-1.20
Web::Framework 1.2

API changes!
9.2 Deprecations

- **Deprecate** ActiveSupport::TestCase#pending method, use skip from MiniTest instead.

- ActiveSupport::Benchmarkable#silence has been **deprecated** due to its lack of thread safety. It will be removed without replacement in Rails 4.1.

- ActiveSupport::JSON::Variable is **deprecated**. Define your own `as_json` and `encode_json` methods for custom JSON string literals.

- **Deprecates** the compatibility method Module#local_constant_names, use Module#local_constants instead (which returns symbols).

- BufferedLogger is **deprecated**. Use ActiveSupport::Logger, or the logger from Ruby standard library.

- **Deprecate** `assert_present` and `assert_blank` in favor of `assert object.blank?` and `assert object.present?`
“Upgrading CPAN modules broke my app”
Dependencies are part of your app.
a solution

cpanm 1.6
> cpanm Web::Framework@1.10
installed Web-Framework-1.10
cpanm Web::Framework~">= 1.0, < 1.2" installed Web-Framework-1.19
a solution
cpanfile + cpanm 1.6
> cat cpanfile
requires 'Web::Framework', '== 1.10';

> cpanm --installdeps .
installed Web-Framework-1.10
cpanfile

DSL to describe prereqs
requires 'Catalyst', '5.8000';

recommends 'JSON::XS', '2.0';

on 'test' => sub {
    requires 'Test::More', '>= 0.96'
};

on 'develop' => sub {
    recommends 'Devel::NYTProf';
};

feature 'sqlite' => sub {
    requires 'DBD::SQLite';
};
inspired by:

gemfile(5)
Backward compatible to:
Module::Install(::DSL)
Converted to to:
CPAN::Meta::Prereqs
Toolset

Module::CPANfile
Module::Install::CPANfile
Dist::Zilla plugins
Supported by
dzil, Milla & cpanm 1.6
cpanfile + cpanm

• Simple
• Yet powerful and flexible way to describe dependencies and version requirements
• Version control cpanfile
caveats

- Locking each dependency with specific version is tedious work
- Can only lock direct dependencies
- MetaCPAN as SPOF
- No easy/reliable way to fallback
Many other solutions

Needs servers, Needs databases
Too simple, Too complicated, etc.
the Solution
Inspired by...
The best way to manage your application's dependencies

Bundler manages an **application's dependencies** through its entire life across many machines systematically and repeatably.

**I am interested in**

- Troubleshooting
- Understanding Bundler
- Gemfile Manual
- CLI Manual

**Getting Started**

Getting started with bundler is easy

```
$ gem install bundler
```
Basic idea:

Describe CPAN dependencies
snapshot tarball pathnames
Built on top of:
cpanfile + cpanm 1.6
• App-specific local environment

• Fast and safe install with caches

• Dep-tree analysis, including versions

• Freezing module versions

• Conservative updates

• Easy Redeployment, Rollback

• Single-file, VCS friendly
Local perl environment

Using local::lib and cpanm -L
Each app has an isolated local library path
Fast and safe install

Saves MYMETA.json and install meta info
Dep tree analysis

Rebuild the dependency tree from snapshot
Checks if anything is missing/superfluous
Freezing versions

Versions are saved in snapshots including dependencies
Easy Redeployment

Reinstall exactly the same set of modules on another prod/development machines.
Conservative Update

Modules won't be upgraded unless it is required, or manually updated.
Single-file, VCS friendly

You can add cpanfile.snapshot to git update whenever you update modules
"Dependencies are part of your app."
Safe and easy rollback

revert the lock file and redeploy
Deployment with Carton
@local> carton install
@local> git commit cpanfile.snapshot
@local> git push

@remote> carton install --deployment
@remote> carton exec plackup ...
Example:

github.com/miyagawa/cpanmetadb-perl
capistrano, Server::Starter,
carton, plackup (twiggy)
PaaS/Cloud

github.com/miyagawa/heroku-buildpack-perl
Creating warm-peak-9823... done, stack is cedar
BUILDPACK_URL=https://github.com/miyagawa/heroku-buildpack-perl.git#carton
http://warm-peak-9823.herokuapp.com/ | git@heroku.com:warm-peak-9823.git
Git remote heroku added

- heroku-perl-sample git:(carton) git push heroku master

Counting objects: 4, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 384 bytes, done.
Total 4 (delta 0), reused 0 (delta 0)

-----> Fetching custom git buildpack... done
-----> Perl/PSGI app detected
-----> Vending Perl
Carton and perl version support: miyagawa/heroku-buildpack-perl

Dependencies can be declared using cpanfile and frozen with carton install. The buildpack will install these dependencies using Carton into a cached .Local directory.

I implemented the experimental buildpack update to enable Carton and perl versions support.

This enables the use of Carton (dev release on v0.9) to deploy dependencies recorded in carton.lock (you have to check it in to the git repo), and then use the file .perl-version ala plenv to specify the perl runtime version. Only the latest minor point release of all stable releases are supported. See the README for details.

This buildpack is for FFI2013 built on Windows, Silicon Valley.

http://weblog.bulknews.net
Support to come!

Dokku, DotCloud, Travis CI, Cloud Foundry etc.
carton 1.0
later this week
> cpanm --dev Carton
Towards 1.1

- Inject patched versions (DarkPAN)
- Install from github
- bootstrap with fatpack
github.com/miyagawa/carton
irc.perl.org #carton
Questions?