LET'S MAKE A GEM

WHAT IS A GEM?

LET'S MAKE A GEM

1. Creating a Gem with Bundler

LET'S MAKE A GEM

1. Creating a Gem with Bundler
2. Testing it with rspec

LET'S MAKE A GEM

1. Creating a Gem with Bundler
2. Testing it with rspec
3. Setting up continuous integration

LET'S MAKE A GEM

1. Creating a Gem with Bundler
2. Testing it with rspec
3. Setting up continuous integration
4. Publishing to Rubygems

LET'S MAKE A GEM

1. Creating a Gem with Bundler
2. Testing it with rspec
3. Setting up continuous integration
4. Publishing to Rubygems
5. Publishing to a private gem server

1. Creating a Gem

RVM

BUNDLER

RUBYGEMS

INITIALIZING OUR GEM

```
$ rvm use 2.1.0
$ gem install bundler
$ bundle gem eleventh
$ cd eleventh
$ rvm --rvmrc --create 2.1.0@eleventh
$ cd ..; cd -
```

GENERATED STRUCTURE

```
$ ls
eleventh
|--- README.md
|--- eleventh.gemspec
|--- Rakefile
|--- Gemfile
|--- lib
      |--- eleventh.rb
      |--- eleventh
            |--- version.rb
```

EXPLORING THE GENERATED STRUCTURE

```
$ vim eleventh.gemspec
$ vim Gemfile
$ vim lib/eleventh.rb
$ vim lib/eleventh/version.rb
$ rake -T
```

MAKE IT DO SOMETHING

```
$ vim lib/eleventh/array_access.rb
$ vim lib/eleventh.rb
$ git add .
$ git commit -m "Version 1"
```

MAKE IT DO SOMETHING

```ruby
# ./lib/eleventh/array_access.rb
class Array
  def eleventh
    self[10]
  end
end

# ./lib/eleventh.rb
require "eleventh/version"
require "eleventh/array_access"

module Eleventh
end
```

2. Testing our gem with rspec

TESTING OUR GEM

```
$ mkdir spec
$ vim spec/spec_helper.rb
$ vim spec/array_access_spec.rb
$ rspec spec/array_access.rb
```

TESTING OUR GEM

```
# ./spec/spec_helper.rb
require 'rubygems'
require 'bundler/setup'

Bundler.setup

require 'eleventh'

RSpec.configure do |config|
end
```

TESTING OUR GEM

```ruby
# ./spec/array_access.rb
require 'spec_helper'

describe Array do
  describe '#eleventh' do
    it 'should return the eleventh element' do
      arr = (1..15).to_a

      arr.eleventh.should eq(11)
    end

    it 'should return nil if the eleventh element not not exist' do
      arr = (1..9).to_a

      arr.eleventh.should eq(nil)
    end
  end
end
```

TESTING OUR GEM

```
$ rspec spec/array_access.rb
..
Finished in 0.00135 seconds
2 examples, 0 failures
```

TESTING OUR GEM

```
$ vim Rakefile

# ./Rakefile
require "bundler/gem_tasks"
require 'rspec/core/rake_task'

RSpec::Core::RakeTask.new('spec')

task :default => :spec
```

TESTING OUR GEM

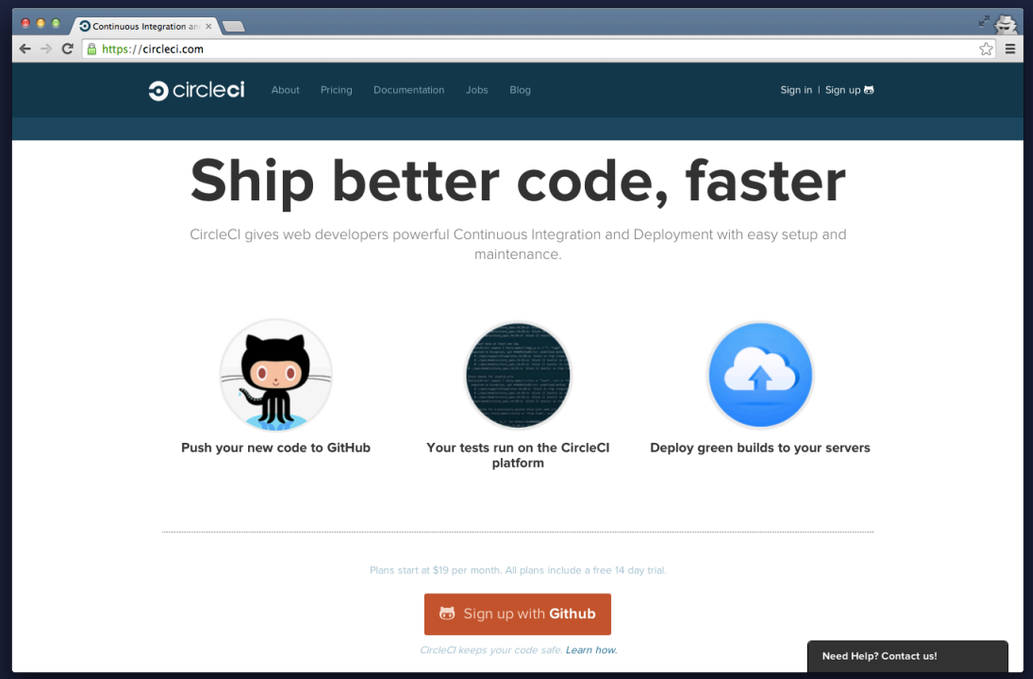
```
$ rake
/Users/bmorris/.rvm/rubies/ruby-2.1.0-p247/bin/ruby -S rspec ./spec/array_access_spec.rb
..

Finished in 0.00066 seconds
2 examples, 0 failures
```

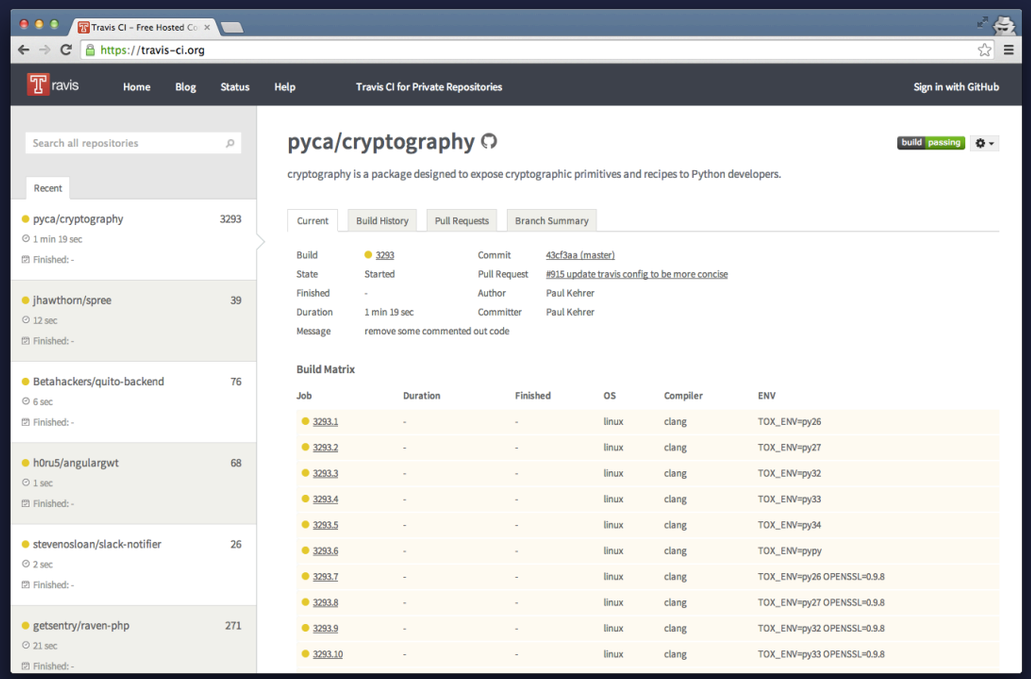
3. Setting up continuous integration

Welcome to Jenkins CI!
jenkins-ci.org
Jenkins
An extendable open source continuous integration server
Fork me on GitHub
BLOG
CONNECT
BUG TRACKER
WIKI
CI
TUTORIALS
ARCHIVES
DONATION
ABOUT
Meet Jenkins
Find out what Jenkins is and get started.
Use Jenkins
See how to get more out of your Jenkins.
Customize Jenkins
Choose from over 600 plugins to customize Jenkins exactly to your needs.
Extend Jenkins
Learn how to build Jenkins or extend Jenkins by writing plugins.
Download Jenkins
Release
Long-Term Support Release
Java Web Archive (.war)
Latest and greatest (1.559)
changelog | past releases | RC
upgrading from Hudson?
Or native package
Windows
Ubuntu/Debian
Red Hat/Fedora/CentOS
Mac OS X
openSUSE
FreeBSD
OpenBSD
Solaris/OpenIndiana
Gentoo

JENKINS

CIRCLE CI


Continuous Integration
https://circleci.com
circleci
About
Pricing
Documentation
Jobs
Blog
Sign in | Sign up
Ship better code, faster
CircleCI gives web developers powerful Continuous Integration and Deployment with easy setup and maintenance.
Push your new code to GitHub
Your tests run on the CircleCI platform
Deploy green builds to your servers
Plans start at $19 per month. All plans include a free 14 day trial.
Sign up with Github
CircleCI keeps your code safe. Learn how.
Need Help? Contact us!


Travis CI - Free Hosted Co
https://travis-ci.org
travis
Home
Blog
Status
Help
Travis CI for Private Repositories
Sign in with GitHub
Search all repositories
Recent
pyca/cryptography 3293
1 min 19 sec
Finished: -
jhawthorn/spree 39
12 sec
Finished: -
Betahackers/quito-backend 76
6 sec
Finished: -
h0ru5/angulargwt 68
1 sec
Finished: -
stevenosloan/slack-notifier 26
2 sec
Finished: -
getsentry/raven-php 271
21 sec
Finished: -
pyca/cryptography
build passing
cryptography is a package designed to expose cryptographic primitives and recipes to Python developers.
Current
Build History
Pull Requests
Branch Summary
Build 3293
State Started
Finished -
Duration 1 min 19 sec
Message remove some commented out code
Commit 43cf3aa (master)
Pull Request #915 update travis config to be more concise
Author Paul Kehrer
Committer Paul Kehrer
Build Matrix
Job Duration Finished OS Compiler ENV
3293.1 - - linux clang TOX_ENV=py26
3293.2 - - linux clang TOX_ENV=py27
3293.3 - - linux clang TOX_ENV=py32
3293.4 - - linux clang TOX_ENV=py33
3293.5 - - linux clang TOX_ENV=py34
3293.6 - - linux clang TOX_ENV=pypy
3293.7 - - linux clang TOX_ENV=py26 OPENSSL=0.9.8
3293.8 - - linux clang TOX_ENV=py27 OPENSSL=0.9.8
3293.9 - - linux clang TOX_ENV=py32 OPENSSL=0.9.8
3293.10 - - linux clang TOX_ENV=py33 OPENSSL=0.9.8

TRAVIS CI

SETTING UP CONTINUOUS INTEGRATION

```
$ vim travis.yml
$ vim eleventh.gemspec
$ vim README.md
$ vim git commit -m "Setting up Travis CI"
$ git push
```

SETTING UP CONTINUOUS INTEGRATION

```
# ./travis.yml

language: ruby
rvm:
  - 2.1.0
```

4. Publishing to Rubygems

PUBLISHING TO RUBYGEMS

```
$ rake -T
$ rake release
```

```
$ gem build eleventh.gemspec
$ gem push eleventh-0.0.1.gem

$ open rubygems.org
```

TA-DA!

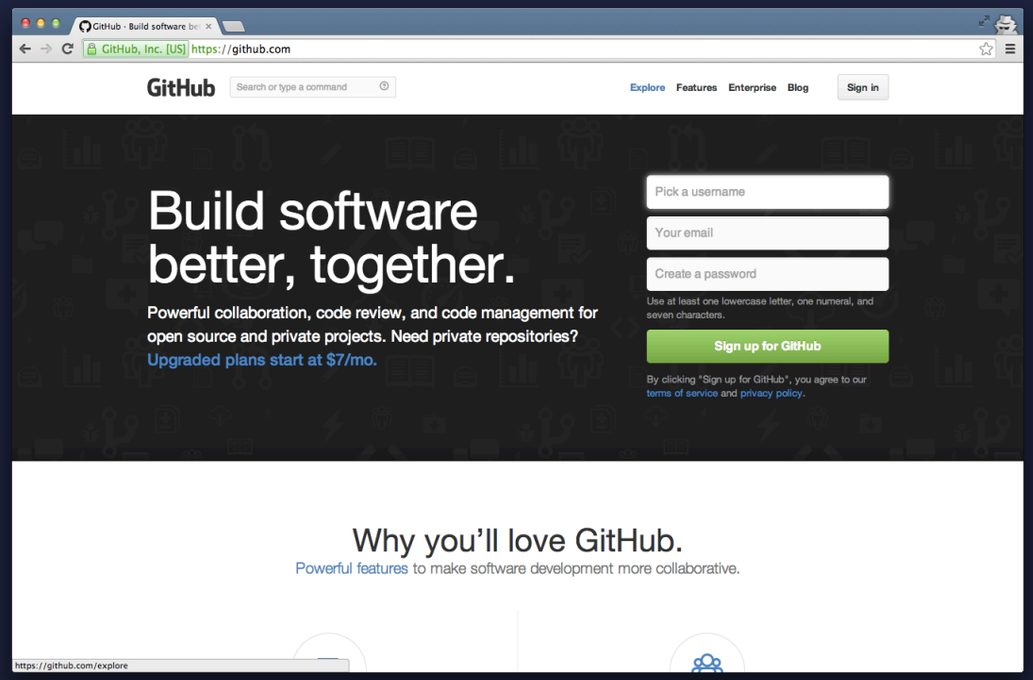
REMOVING THE GEM

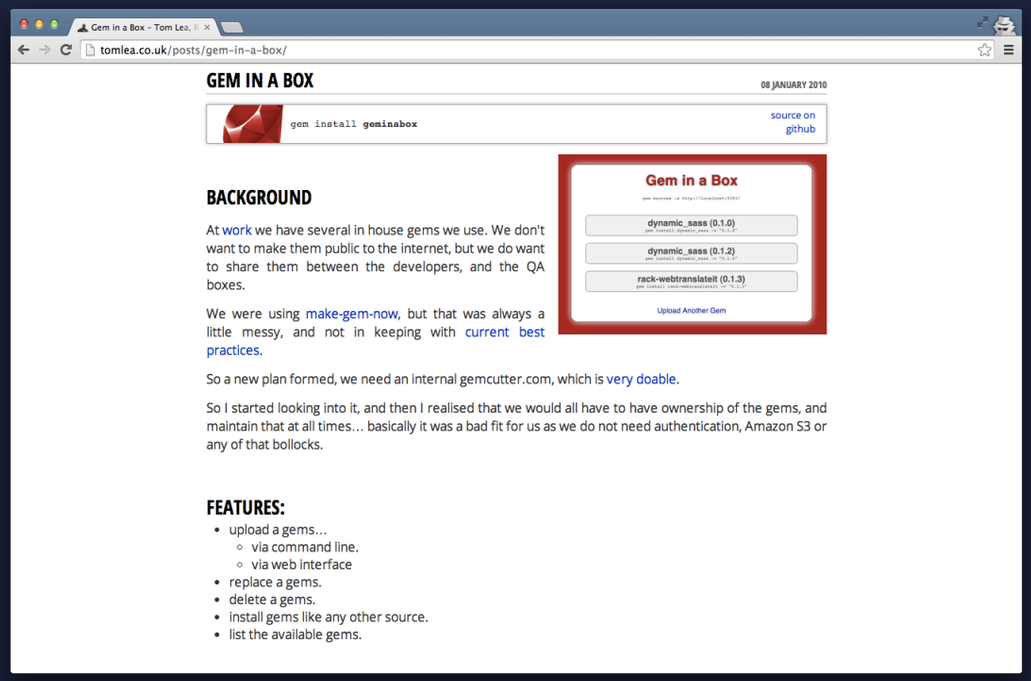
```
$ gem yank eleventh -v 0.0.1
```

5. Publishing to a private gem server

WHY PRIVATE?

GITHUB

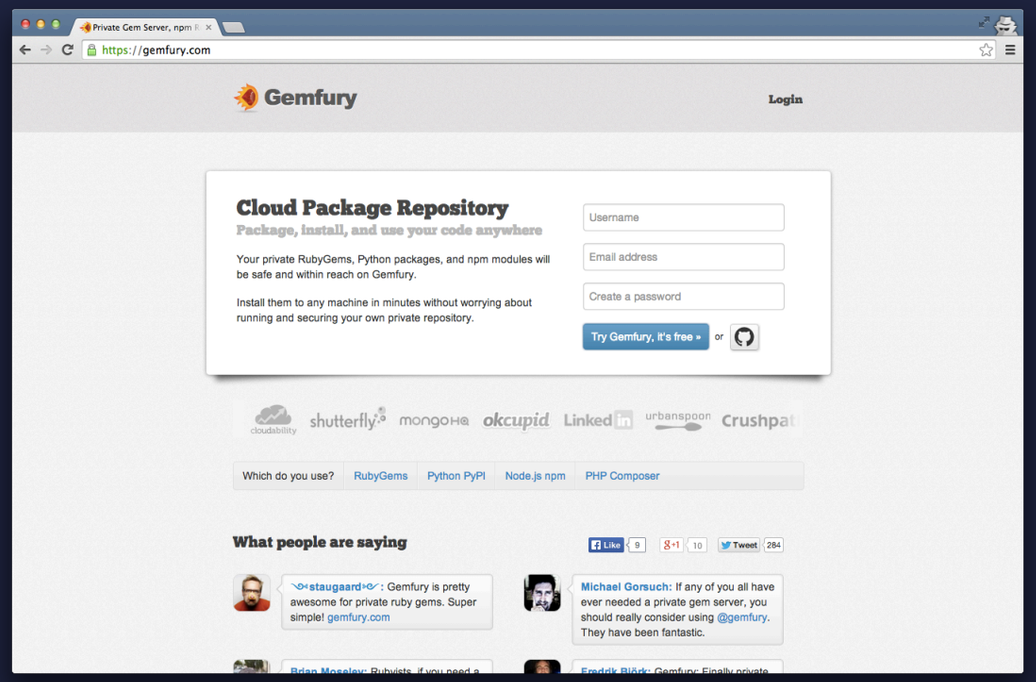

GitHub · Build software
GitHub, Inc. [US] https://github.com
GitHub
Search or type a command
Explore Features Enterprise Blog
Sign in
Build software better, together.
Powerful collaboration, code review, and code management for open source and private projects. Need private repositories?
Upgraded plans start at $7/mo.
Pick a username
Your email
Create a password
Use at least one lowercase letter, one numeral, and seven characters.
Sign up for GitHub
By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy.
Why you'll love GitHub.
Powerful features to make software development more collaborative.
https://github.com/explore


Gem in a Box - Tom Lea
tomlea.co.uk/posts/gem-in-a-box/
GEM IN A BOX
08 JANUARY 2010
gem install geminabox
source on github
BACKGROUND
At work we have several in house gems we use. We don't want to make them public to the internet, but we do want to share them between the developers, and the QA boxes.
We were using make-gem-now, but that was always a little messy, and not in keeping with current best practices.
Gem in a Box
dynamic_sass (0.1.0)
dynamic_sass (0.1.2)
rack-webtranslateit (0.1.3)
Upload Another Gem
So a new plan formed, we need an internal gemcutter.com, which is very doable.
So I started looking into it, and then I realised that we would all have to have ownership of the gems, and maintain that at all times… basically it was a bad fit for us as we do not need authentication, Amazon S3 or any of that bollocks.
FEATURES:
upload a gems…
via command line.
via web interface
replace a gems.
delete a gems.
install gems like any other source.
list the available gems.

GEMINABO

X

GEMFURY


Private Gem Server, npm
https://gemfury.com
Gemfury
Login
Cloud Package Repository
Package, install, and use your code anywhere
Your private RubyGems, Python packages, and npm modules will be safe and within reach on Gemfury.
Install them to any machine in minutes without worrying about running and securing your own private repository.
Username
Email address
Create a password
Try Gemfury, it's free »
or
cloudability
shutterfly
mongoHQ
okcupid
LinkedIn
urbanspoon
Crushpath
Which do you use?
RubyGems
Python PyPI
Node.js npm
PHP Composer
What people are saying
Like 9
+1 10
Tweet 284
staugaard: Gemfury is pretty awesome for private ruby gems. Super simple! gemfury.com
Michael Gorsuch: If any of you all have ever needed a private gem server, you should really consider using @gemfury. They have been fantastic.

PUBLISHING TO GEMFURY

```
$ gem install gemfury
$ gem build eleventh.gemspec
$ fury push eleventh-0.0.1.gem

$ git remote add fury https://bnmrrs@git.fury.io/bnmrrs/eleventh.git
$ git push fury master
```

PUBLISHING TO GEMFURY

```
$ cd ../other-project
$ vim Gemfile
```

```
# ../other-project/Gemfile
source 'https://TOKEN@gem.fury.io/bnmrrs/'
```

```
$ bundle install
```

PUBLISHING A NEW VERSION

```
$ vim lib/eleventh/version.rb
$ gem build eleventh.gemspec
$ git add .
$ git commit -m "Version 2"
$ fury push eleventh-0.0.2.gem
$ fury list
```

QUESTIONS?

@bnmrrs
www.boltmade.com

https://github.com/bnmrrs/talks