BeenVerified.com™

Rollout Features with Confidence!

Arjun Anand
@yodarjun

Context

- BeenVerified has been around for 13 years
- Millions of visitors per week
- Bootstrapped, very profitable
- Proudly made in NYC with satellite office in Costa Rica
- Multiple full time remote devs
- We’re hiring full stack devs :)

Context

- One core system part is Flowrida
- Selects flows for users to go through before hitting the paywall
- Allows for A/B testing
- Co-ordinates campaign efforts
- Strong test coverage

The Challenge

- Needed to make a fundamental change
- Needed to test it in production
- Needed to measure its success against the way things were
- Needed it to be easily extensible to future tests
- Wanted to do iterative development
- Did not want to maintain a long running branch
- Needed an easy switch to turn on feature for production environment
- Needed the code to be clean
- But needed to all of this without affecting business as usual

Initial Thoughts

- Conditionals based on a user id
  - Not enough control
  - Not clean code
  - No easy way to set up ids before a user signs up
- Conditionals based on environment
  - Could work
  - Not clean code
  - Cleanup would be hard

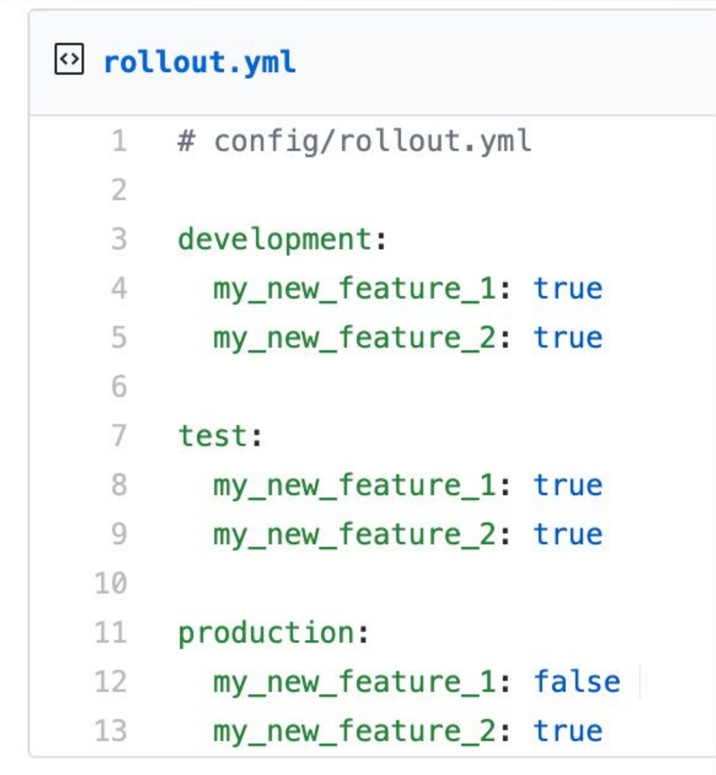
Third party libraries

- https://github.com/fetlife/rollout
  - Great tool
  - Very flexible
  - Has way more functionality than we needed

Enter Rollout (no, our version)

- `config/rollout.yml`
- `config/initializers/rollout.rb`
- `lib/rollout.rb`

config/rollout.yml



rollout.yml

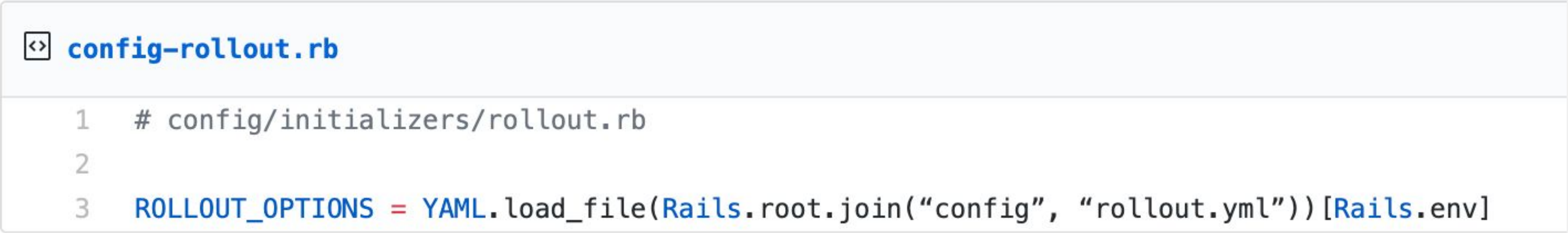
```
# config/rollout.yml

development:
  my_new_feature_1: true
  my_new_feature_2: true

test:
  my_new_feature_1: true
  my_new_feature_2: true

production:
  my_new_feature_1: false
  my_new_feature_2: true
```

config/initializers/rollout.rb

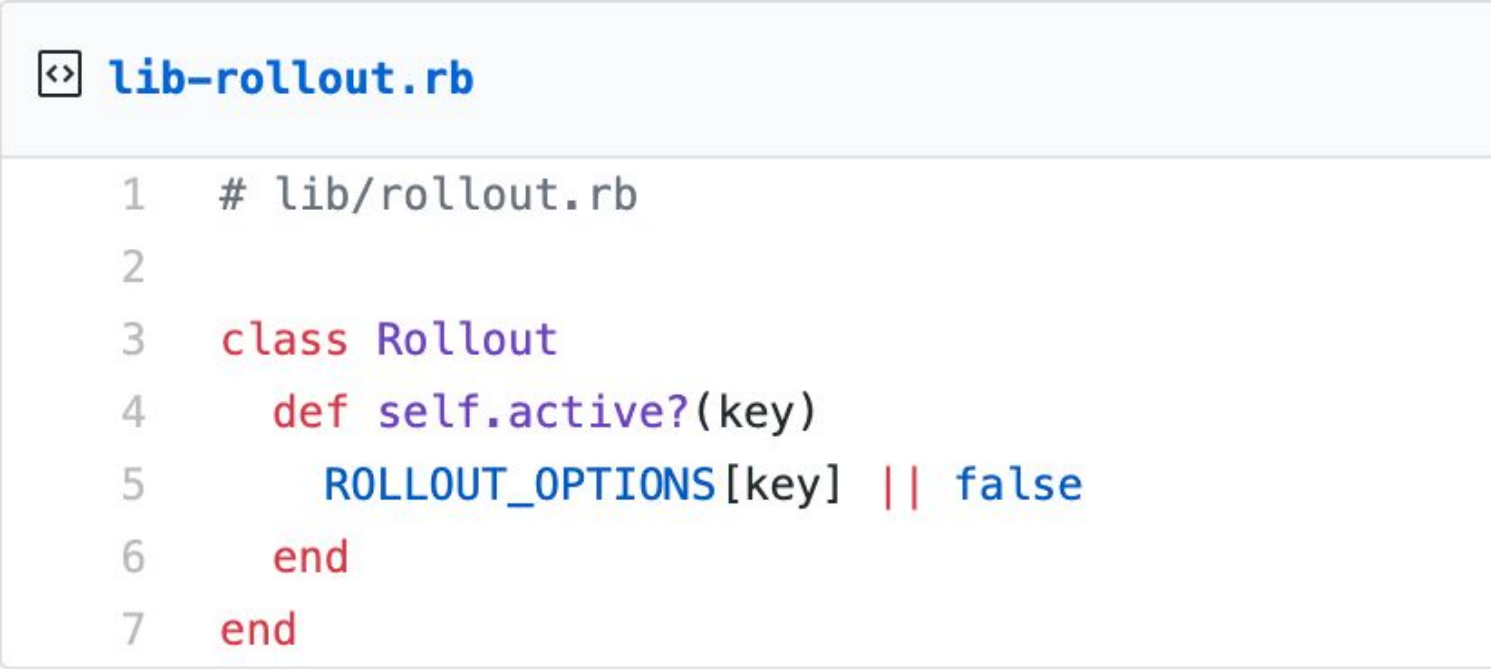


config-rollout.rb

```ruby
# config/initializers/rollout.rb

ROLLOUT_OPTIONS = YAML.load_file(Rails.root.join(“config”, “rollout.yml”))[Rails.env]
```

lib/rollout.rb

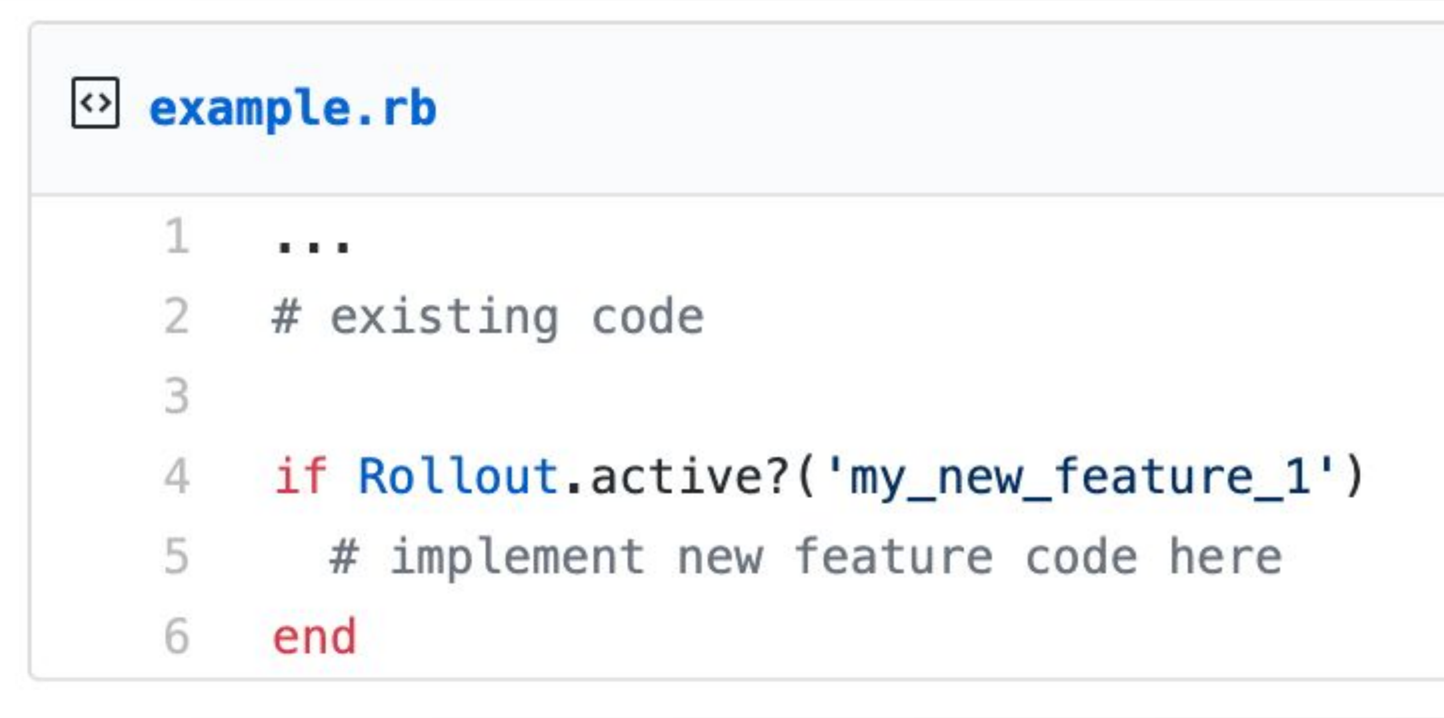


lib-rollout.rb

```ruby
# lib/rollout.rb

class Rollout
  def self.active?(key)
    ROLLOUT_OPTIONS[key] || false
  end
end
```

Usage



example.rb

```ruby
...
# existing code

if Rollout.active?('my_new_feature_1')
  # implement new feature code here
end
```

Post processing

- When you want to turn on the feature in production env
    - config/rollout.yml

```
Production:

    Feature: true
```

Cleanup

- Remove instances of `if Rollout.active?(feature)`
- Remove the feature from `config/rollout.yml`

Conclusions

- Great way to do iterative development
- Simple and easy setup and cleanup
- Shipping features progressively while not affecting production env
- One quick flip of a switch to turn it on in production

Will this solve all of your rollout problems?

- No
- This worked for us for this instance
- YMMV depending on need and context

Questions?

Links

- Code in this slideshow can be found here:
  - https://gist.github.com/yodarjun/afe54709534e8afec8ec87da2bce2f60
- Rollout gem:
  - https://github.com/fetlife/rollout

Thank you

BeenVerified.com™