Keith & Mario’s Guide To CONTINUOUS DEPLOYMENT

OR

Why automatically deploying your master branch to production is a good idea.
Previously...

Keith and Mario’s Guide to FAST WEBSITES

@keithpitt @mariovisic
A community of handpicked designers and developers just for you.
Secure and flexible continuous integration.

Buildbox hosts the tools, and run the tests on your own infrastructure giving you complete control.

Learn More  Sign up for Free
ALRIGHT, LISTEN UP
NOT SURE IF CONTINUOUS DELIVERY
OR DEPLOYMENT
WHY CONTINUOUS DEPLOYMENT?
FAST FEEDBACK
CONFIDENCE
PREDICTABILITY
INGREDIENTS
INGREDIENTS

Feature Planning
Code Reviews
Feature Toggles
Automated Tests
Smoke Tests
Zero Downtime Deploys
Rollback Strategy
Handling Database Migrations
Application Monitoring
FEATURE PLANNING
Feature Planning

SMALLEST THING THAT YOU CAN SHIP TODAY
Feature Planning

How do you deploy it with zero downtime?
HOW WILL WE HANDLE DATABASE MIGRATIONS?

Feature Planning
Reviewing Code

GITHUB PULL REQUESTS
add flux capacitor #15

Merged cavalle merged 1 commit into bebanjo:master from nashby:flux-capacitor over 1 year ago

nashby commented 2 years ago

add flux capacitor

jwo commented on the diff 2 years ago
Reviewing Code

GITHUB.COM/SQUARE/CANE
Reviewing Code

GITHUB.COM/BBATSOV/RUBOCOP
FEATURE TOGGLES
Feature Toggles

KEEP IT GHETTO
Feature Toggles

GITHUB.COM/FETLIFE/ROLLOUT
Feature Toggles

GITHUB.COM/PDA/FLIP
before do
  LoginPage.login(user)
end

t "shows the output of the build on the project view page" do
  project_show_page = ProjectShowPage.visit(build.project.account)
  expect(project_show_page).to have_build_with_message("Oh hai!")
end

t "can be navigated to via the build history and viewed" do
  project_history_page = ProjectHistoryPage.visit(build.project.
  project_history_page.click_build_with_message("Oh hai!")
  build_show_page = buildShowPage.new
  expect(build_show_page).to have_build_with_message("Oh hai!")
end
Automated Tests

Write just enough tests to trust the code.
Automated Tests

SO THEY’RE FAST
Automated Tests

Corey Haines Fast Rails Tests
SMOKE TESTS
Smoke Tests

**RUNS** POST DEPLOY AGAINST **REMOTE SERVER**
BONUS POINTS: THE SAME TESTS RUN AGAINST REMOTE SERVER & LOCAL
ZERO DOWNTIME DEPLOYS
Zero Downtime Deploys

Deploying new code without maintenance page
Zero Downtime Deploys

**HEROKU PRO TIP:**
**HEROKU LABS:ENABLE PREBOOT**
Zero Downtime Deploys

#373 Zero-Downtime Deployment (pro) – RailsCasts

Here I show how to accomplish zero-downtime deployment using Unicorn. I also cover gotchas when working with migrations and how to put up a maintenance page when you do need to take down the site.

Subscribe to Watch Episode

Subscribe to RailsCasts Pro to access this episode and other pro and revised episodes. Only $9 per month.

Learn more about RailsCasts Pro
DATABASE MIGRATIONS
Database Migrations

HARDEST PART
class RemovingNotes < ActiveRecord::Migration
  def self.up
    remove_column :users, :notes
  end
end
PGError: ERROR: column "notes" does not exist
Database Migrations

EASY WAY:
MAINTENANCE PAGE
HARD WAY: DEPLOY MIGRATIONS FIRST
Rails migrations with no downtime

JUL 13 2011

Lets understand the challenge of changing a Rails database without introducing any downtime with a simple, apparently harmless migration:

```ruby
class ApparentlyHarmlessMigration < ActiveRecord::Migration
  def self.up
    remove_column :users, :notes
  end
end
```

I learned this kind of migration is not really harmless the hard way: during a production deploy. Since the column was not being referred anywhere in the code, I assumed it was a migration that I could just run at will.
Database Migrations

BEWARE OF DATABASE LOCKING
Database Migrations

GITHUB.COM/SOUNDCLOUD/LHM

GitHub repository for Online MySQL schema migrations.
ROLLING BACK DEPLOYS
Rolling Back Deploys

JUST ROLL FORWARD
Rolling Back Deploys

Just turn the feature off.
Rolling Back Deploys

If you have to: Git Revert and deploy
Rolling Back Deploys

TURN ON MAINTENANCE PAGE
APPLICATION MONITORING
Exception Tracking

Application Monitoring

Detect and diagnose crashes in your applications

Bugsnag’s cross platform error monitoring helps you ship with confidence

LEARN MORE
TRY IT FOR FREE
Application Monitoring

PERFORMANCE MONITORING

New Relic

YOU CAN MONITOR
PRICING
ABOUT
ENTERPRISE

Application Performance Management & Monitoring | New Relic

App performance increases happiness

Get immediate code-level visibility and build faster, more reliable web and mobile applications.

Create Free Account

No credit card required • No commitment
Application Monitoring

DAVID GOODLAD BUSINESS METRIC MONITORING

vimeo

Monitorama EU 2013 – David Goodlad on Vimeo

Compu’er says no
Getting Notified

Application Monitoring

Alert the right person, every time.

PagerDuty provides SaaS IT on-call schedule management, alerting and incident tracking.

- Add scheduling to existing monitoring solutions
- Alert via SMS, phone, email, iOS or Android push
- Respond or escalate issues via your mobile device

SIGN UP NOW
GET A FREE T-SHIRT!

Plug PagerDuty into any of these monitoring tools to enable phone, SMS and mobile push

Nagios®  Zenoss®  pingdom  splunk®  New Relic®  ZABBIX®  AppDynamics®  keynoter®
INGREDIENTS

Feature Planning
Code Reviews
Feature Toggles
Automated Tests
Smoke Tests
Zero Downtime Deploys
Rollback Strategy
Handling Database Migrations
Application Monitoring
TIME TO MIX
1 #!/bin/bash
2 set -x
3 set -e
4
5 git push heroku master
#!/bin/bash
set -x
set -e
rake quality
rake test
git push heroku master
rake smoke:production
#!/bin/bash
set -x
set -e
rake quality
rake test
git push heroku-staging master
rake smoke:staging
git push heroku master
rake smoke:production
THE FINAL PRODUCT
git push
Added social media buttons to everything possible. #240

keithpitt wants to merge 1 commit into master from social-media-buttons

1 participant

This pull request can be automatically merged.
You can also merge branches on the command line.
Merge pull request
#!/bin/bash

set -x
set -e

rake quality
rake test
git push heroku-staging master
rake smoke:staging
git push heroku master
rake smoke:production
Adding social media buttons to everything possible

master / a4064d36

keithpitt

finished 3 months ago

lollies
+ [ l -d .git ]
+ git clean -fd
+ git fetch -q
+ git checkout -qf a4064d36ea453a7971f6dd04f0156d27119e5362
+ bundle install --local
Using rake (10.1.0)
Using i18n (0.6.5)
Using minitest (4.7.5)
Using multi_json (1.8.2)
Using atomic (1.1.14)
Using thread_safe (0.1.3)
Using tzinfo (0.3.38)
Using activesupport (4.0.1)
Using builder (3.1.4)
Using erubis (2.7.0)
Using rack (1.5.2)
$rollout.activate_group(:social_media_buttons,:developers)
$rollout.activate_group(
   :social_media_buttons,
   :all
)
}
FAST FEEDBACK

CONFIDENCE

PREDICTABILITY
FAST FEEDBACK
CONFIDENCE
PREDICTABILITY
AVERAGE NUMBER OF DEPLOYS PER DAY

- Manual Deploys
- Continuous Deploys
AVG NUMBER OF DEPLOYS PER DAY

4.7
130 Additions Per Deploy

1.9
245 Additions Per Deploy

Manual Deploys

Continuous Deploys
A always
B be
C continuously
D deploying
CONTINUOUS DEPLOYMENT

FIN.

@keithpitt
@mariovisic