RUBYCONF
O mesmo Ruby, tudo novo.
ChatOps in Ruby
Gabriel Mazetto
Full-Stack Developer

@brodock
github.com/brodock
blog.gabrielmazetto.eti.br
Resultados Digitais
Operations and Performance Team
# /etc/init.d/daemon stop
Some Numbers...
10.1M+ requests
116.6M+ background transactions
120M+ e-mails delivered

this month!
35K+ commits
3.2K+ closed PRs
2 bots 24/7
10+ deploys per day
What is ChatOps?
Doing Operations in a chat
Capybot BOT 8:56 PM
Starting deployment of rdstation/fix-analytics-business-funnel-html to production that was requested 19s ago
Updated the quantity of web dynos from 1 to 2
Deployment of rdstation/fix-analytics-business-funnel-html to production created

Capybot BOT 8:56 PM
#1823714 - rdstation / fix-analytics-business-funnel-html / production
  #1823714: lleos is deploying rdstation to production (compare)

  #1823714: lleos's production deployment of rdstation is done!

Capybot BOT 8:58 PM
Waiting Heroku preboot...
Deployment succeeded!
Your deployment should be available in production
I'll restore dynos' quantity after 2 minutes from now just for precaution
Published rdstation/fix-analytics-business-funnel-html to production that was requested 7m 36s ago
Restoring the quantity of web dynos from 2 to 1

salvador 9:04 PM
capybot merge it!
Using your chat as an automated CLI
dig resultadosdigitais.com.br txt

;; Answer received from 8.8.8.8:53 (304 bytes)
;;
;; HEADER SECTION
;; id = 44493
;; qr = 1  opCode: QUERY  aa = 0  tc = 0  rd = 1
;; ra = 1  ad = 0  cd = 0  rcode = NoError
;; qdCount = 1  anCount = 3  nsCount = 0  arCount = 0

;; QUESTION SECTION (1 record):
;; resultadosdigitais.com.br.  IN  TXT

;; ANSWER SECTION (3 records):
resultadosdigitais.com.br.  14399  IN  TXT  "v=spf1 a mx include:_spf.google.com include:sendgrid.net include:mail.zendesk.com ~all"
resultadosdigitais.com.br.  14399  IN  TXT  "google-site-verification=D_TyNsVRtbMlaaZJoL3mqKzufV04jDjpDEiDG41qiwg"
resultadosdigitais.com.br.  14399  IN  TXT  "google-site-verification=x9qQ_f6Nd3JA5D952yQ9RbmQo5kyH6yRRhjxXP6Uj0E"
Conversation Driven Development
Forget TDD, BDD
It's all about CDD!
Just kidding!

Keep testing, please!
Conversation Driven Development
It's about Communicating what, when, why
geisonbiazus 4:49 PM
capybot queue next!

Capybot  BOT 4:49 PM ⭐
@brodock: You're next!

geisonbiazus: Removed geisonbiazus from queue

nathan.thiesen 4:58 PM
capybot queue me

Capybot  BOT 4:58 PM
[brodock, nathan.thiesen]
Reinforces organizational culture!
Alguém bate um olho, só pra ideia de vai ter alguém chamada lenta

@pedro.brentan: ?

E dai, faz o deploy :)

e, tem isso tbm

Capybot  BOT  10:58 AM

Empowering Non Ops

it should be easy...
Everybody deploys even designers
Lita
Robot companion in ruby
Eventmachine based
Plugins are rubygems too
Redis as your datastore
Integrated Webserver for webhooks
Rspec Integration

so you can test, seriously!
How to Install
and first steps
$ gem install lita

Successfully installed lita-4.6.1
Parsing documentation for lita-4.6.1
Installing ri documentation for lita-4.6.1
Done installing documentation for lita after 1 seconds
1 gem installed
$ lita --help

Commands:

  lita adapter NAME      # Generates a new Lita adapter
  lita extension NAME    # Generates a new Lita extension
  lita handler NAME      # Generates a new Lita handler
  lita help [COMMAND]    # Describe available commands or one specific command
  lita new NAME          # Generates a new Lita project (default name: lita)
  lita start             # Starts Lita
  lita version           # Outputs the current version of Lita
$ lita new confbot

create confbot
create confbot/Gemfile
create confbot/lita_config.rb
$ cat Gemfile

source "https://rubygems.org"

gem "lita"

# Uncomment to use the HipChat adapter
# gem "lita-hipchat"

# Uncomment to use the IRC adapter
# gem "lita-irc"

# Add handlers to give Lita new functionality.
# For example:
# gem "lita-google-images"
# gem "lita-karma"
$ cat lita_config.rb
Lita.configure do |config|
  # The name your robot will use.
  config.robot.name = "ConfBot"

  # The locale code for the language to use.
  # config.robot.locale = :en

  # The severity of messages to log. Options are:
  # :debug, :info, :warn, :error, :fatal
  # Messages at the selected level and above will be logged.
  config.robot.log_level = :info

  ...

$ bundle install
Fetching gem metadata from https://rubygems.org/...........
Resolving dependencies...
Using multipart-post 2.0.0
Using bundler 1.7.6
Using i18n 0.7.0
Using rb-readline 0.5.2
Using ice_nine 0.11.1
Using multi_json 1.11.0
Using redis 3.2.1
Using thor 0.19.1
Using faraday 0.9.1
Using redis-namespace 1.5.2
Installing rack 1.6.1
...
running it on the terminal
$ lita start
Type "exit" or "quit" to end the session.
ConfBot > ConfBot help
ConfBot: help - Lists help information for terms and command the robot will respond to.
ConfBot: help COMMAND - Lists help information for terms or commands that begin with COMMAND.
ConfBot: info - Replies with the current version of Lita.
ConfBot: users find SEARCH_TERM - Find a Lita user by ID, name, or mention name.
ConfBot > tem alguem ai?
ConfBot >
Creating a plugin
$ lita handler rubyconf

create lita-rubyconf/lib/lita/handlers/rubyconf.rb
create lita-rubyconf/lib/lita-rubyconf.rb
create lita-rubyconf/spec/lita/handlers/rubyconf_spec.rb
create lita-rubyconf/spec/spec_helper.rb
create lita-rubyconf/locales/en.yml
create lita-rubyconf/templates/.gitkeep
create lita-rubyconf/Gemfile
create lita-rubyconf/lita-rubyconf.gemspec
create lita-rubyconf/.gitignore
create lita-rubyconf/Rakefile
create lita-rubyconf/README.md
module Lita
  module Handlers
    class Rubyconf < Handler
      end
    end
  end
end

Lita.register_handler(Rubyconf)
end
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/keynote$/i, :next_keynote, command: true,
        help: {"keynote" => "Display the next keynote"})

      def next_keynote(message)
        keynote = Keynote.get_next
        message.reply "Next keynote will be: #{keynote.title}"
      end

    end

    Lita.register_handler(Rubyconf)
  end
end
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/keynote$/i, :next_keynote, command: true, help: {"keynote" => "Display the next keynote"})
      
      def next_keynote(message)
        keynote = Keynote.get_next
        message.reply "Next keynote will be: #{keynote.title}"
      end
      
    end
  end
end

Lita.register_handler(Rubyconf)
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/keynote$/i, :next_keynote, command: true,
          help: {"keynote" => "Display the next keynote"})

      def next_keynote(message)
        keynote = Keynote.get_next
        message.reply "Next keynote will be: #{keynote.title}"
      end

      end
    end

    Lita.register_handler(Rubyconf)
  end
end
Continuous Integration

Checking Status
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/status ci (.+)/i, :check_ci, command: true,
        help: {"status ci" => "Display build status of informed branch"})

      def check_ci(message)
        ci = XundaCI.get_status(message.match_data[1])
        response.reply "Status do build: #{ci.status}"
      end
    end
  end
end
Lita.register_handler(Rubyconf)
end
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/status ci (.+)/i, :check_ci, command: true, help: {
        "status ci" => "Display build status of informed branch"
      })

      def check_ci(message)
        ci = XundaCI.get_status(message.match_data[1])
        response.reply "Status do build: #{ci.status}"
      end
    end
  end
end
Lita.register_handler(Rubyconf)
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/status ci (.+)/i, :check_ci, command: true,
            help: {"status ci" => "Display build status of informed branch"})

      def check_ci(message)
        ci = XundaCI.get_status(message.match_data[1])
        response.reply "Status do build: #{ci.status}"
      end

      end
    end
  end
end

Lita.register_handler(Rubyconf)
end
Automating code deployment
Github Deployment Flow

Heaven + Heroku
Heaven is an API that integrates with GitHub's Deployment API. It receives deployment events from GitHub and pushes code to your servers.

Heaven currently supports Capistrano, Fabric, and Heroku deployments. It also has a notification system for broadcasting deployment status events to chat services (e.g. Campfire, Hipchat, SlackHQ, and Flowdock). It can be hosted on Heroku for a few dollars a month.

Documentation

- Overview
- Installation
- Deployment Providers
- Deployment Notifications
- Environment Locking

github.com/atmos/heaven
Heaven is a rails app that receives Deployment events from GitHub and deploys your code. It works best with a hubot, and gives you a chat-ops style workflow. It receives GitHub webhooks and runs deployment jobs as background tasks with resque. Heaven captures the standard input and output streams and posts the results to a gist.
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/deploy (\w*) (\w*)$/i, :deploy_branch, command: true, help:
        {"deploy [repo] [branch]" => "Deploys code according to informed params"})

      def deploy_branch(message)
        repo, branch = message.match_data[1], message.match_data[2]
        github.create_deployment(repo, branch)

        response.reply "Started deploying #{repo}/#{branch}!"
      end

      # continua...
    end
  end
end

Lita.register_handler(Rubyconf)
end
module Lita
  module Handlers
    class Rubyconf < Handler
      route(/deploy \(\w*\) \(\w*\)$/i, :deploy_branch, command: true,
      help: {
        "deploy [repo] [branch]" => "Deploys code according to informed
        params"
      })

      def deploy_branch(message)
        repo, branch = message.match_data[1], message.match_data[2]
        github.create_deployment(repo, branch)
        response.reply "Started deploying #{repo}/#{branch}!"
      end
      # there is more...
    end
  end
end

Lita.register_handler(Rubyconf)
module Lita
  module Handlers
    class Rubyconf < Handler
      # more here :)
      config :token, type: String, required: true

      def github
        @github ||= Octokit::Client.new(access_token: config.token)
      end
    end
  end
end

Lita.register_handler(Rubyconf)
$ lita start
Type "exit" or "quit" to end the session.

ConfBot > ConfBot deploy xunda-app/xunda-branch
ConfBot: Started deploying xunda-app/xunda-branch!
ConfBot >
Questions?
github.com/brodock
@brodock
blog.gabrielmazetto.eti.br
shipit.resultadosdigitais.com.br