

ChatOps: Operations meet chat

Jose Luis Salas

April 12, 2015

Outline

whoami

What is ChatOps

What does ChatOps provide?

Questions I haven't asked since ChatOps

ChatOps in practice

What do you need?

Hubot

Lita

\$ lita new bot

\$ lita handler myhandler

Handler usage

Err

Examples

Nagios

Deploying

More examples

ChatOps @ peerTransfer

Bibliography

whoami

I'm Jose Luis Salas

Work as SRE at peerTransfer, programming in Ruby and Shell mostly

Open source contributor: josacar/chef-babushka, josacar/lita-campfire, josacar/lita-jobs, fog/fog, sethvargo/chefspec, rumblelabs/asset_sync, aetrion/chef-dnsimple, jesseadams/sphinx-cookbook, opscience-cookbooks/client-rekey, librato/papertrail-cookbook, sstephenson/ruby-build, hw-cookbooks/route53, josephruscio/twke, geokit/geokit, collectiveidea/tinder, berkshelf/berkshelf, librato/collectd-librato-cookbook, chef/ohai, opscience-cookbooks/rsyslog, peter-murach/github

Twitter: @josacar

GitHub: josacar

What is ChatOps

Doing Operations in a Chat tool

What does ChatOps provide?

1. Everyone sees all of that happen since their first day
2. Place tools in the middle of the conversation
3. Teaching by doing by making things visible
4. Communicate by doing

Questions I haven't asked since ChatOps

- How is the deployment doing?
- Should I deploy or are you deploying that?
- Can I deploy to this environment?
- Is the smoke suite green?
- Is anyone checking the sensu alert?
- Has CI passed in that branch?
- Is the system load ok?
- Is anyone using this environment?

ChatOps in practice

What do you need?

- A chat client: IRC, HipChat, Slack, Flowdock, and Campfire
- Chat bots (all FLOSS):
 - Hubot
 - Lita
 - Err

Hubot

- Written in CoffeeScript and JavaScript
- Runs on top of NodeJS
- Created and used by GitHub
- Supports multiple chats: Campfire, Slack, HipChat
- Storage (brain) : Redis, PostgreSQL, MySQL
- Supports chat and HTTP routes

pugme.coffee

```
1 module.exports = (robot) ->
2
3   robot.respond /pug me/i, (msg) ->
4     msg.http("http://pugme.herokuapp.com/random")
5       .get() (err, res, body) ->
6         msg.send JSON.parse(body).pug
7
8   robot.respond /pug bomb( (\d+))?.i, (msg) ->
9     count = msg.match[2] || 5
10    msg.http("http://pugme.herokuapp.com/bomb?count=" + count)
11      .get() (err, res, body) ->
12        msg.send pug for pug in JSON.parse(body).pugs
13
14  robot.respond /how many pugs are there/i, (msg) ->
15    msg.http("http://pugme.herokuapp.com/count")
16      .get() (err, res, body) ->
17      msg.send "There are #{JSON.parse(body).pug_count} pugs."
```

Lita

- Written in Ruby (requires Ruby \geq 2.0)
- Runs on top of MRI / JRuby / Rubinius
- Created by JimmyCuadra
- Supports multiple chats: Campfire, Slack, HipChat, IRC
- Storage: Redis
- Supports chat and HTTP routes
- Testable with RSpec
- Extensible using plugins through Bundler

\$ lita new bot

```
1 # lita_config.rb
2
3 Lita.configure do |config|
4   # The name your robot will use.
5   config.robot.name = 'bot'
6
7   # The locale code for the language to use.
8   config.robot.locale = :en
9
10  # The severity of messages to log. Options are:
11  # :debug, :info, :warn, :error, :fatal
12  # Messages at the selected level and above will be logged.
13  config.robot.log_level = :info
14
15  # An array of user IDs that are considered administrators. These users
16  # the ability to add and remove other users from authorization groups.
17  # What is considered a user ID will change depending on which adapter you use.
18  config.robot.admins = ["1", "2"]
19
20  # The adapter you want to connect with. Make sure you've added the
21  # appropriate gem to the Gemfile.
```

ChatOps: Operations meet chat

```
22 config.robot.adapter = :slack
23
24 ## Example: Set options for the chosen adapter.
25 # config.adapter.username = "myname"
26 # config.adapter.password = "secret"
27 config.adapters.slack.token = 'xoxb-3242497751-XHFJhTNa87987dhADff7873A'
28
29 ## Example: Set options for the Redis connection.
30 # config.redis.host = "127.0.0.1"
31 # config.redis.port = 1234
32
33 ## Example: Set configuration for any loaded handlers. See the handler's
34 ## documentation for options.
35 # config.handlers.some_handler.some_config_key = "value"
36 end
```

ChatOps: Operations meet chat

```
1 # Gemfile
2
3 source 'https://rubygems.org'
4
5 gem 'lita'
6 gem 'lita-slack'
7
8 # Uncomment to use the HipChat adapter
9 # gem "lita-hipchat"
10
11 # Uncomment to use the IRC adapter
12 # gem "lita-irc"
13
14 # Add handlers to give Lita new functionality.
15 # For example:
16 gem "lita-google-images"
17 # gem "lita-karma"
```

\$ lita handler myhandler

Do you want to test your plugin on Travis CI? yes

Do you want to generate code coverage information with SimpleCov and Coveralls.io? yes

If your plugin's Git repository will be hosted on GitHub, build status and code coverage badges can be \ automatically added to your README.

Would you like to add these badges? ("yes" or "no", default is "no") yes

What is your GitHub username? josacar

```
create lita-myhandler/lib/lita/handlers/myhandler.rb
```

```
create lita-myhandler/lib/lita-myhandler.rb
```

```
create lita-myhandler/spec/lita/handlers/myhandler\_spec.rb
```

```
create lita-myhandler/spec/spec\_helper.rb
```

```
create lita-myhandler/locales/en.yml
```

```
create lita-myhandler/templates/.gitkeep
```

```
create lita-myhandler/Gemfile
```

```
create lita-myhandler/lita-myhandler.gemspec
```

```
create lita-myhandler/.gitignore
```

```
create lita-myhandler/.travis.yml
```

```
create lita-myhandler/Rakefile
```

```
create lita-myhandler/README.md
```

lib/lita/handlers/myhandler.rb

```
1 module Lita
2   module Handlers
3     class Myhandler < Handler
4       end
5
6     Lita.register_handler(Myhandler)
7   end
8 end
```

spec/lita/handlers/myhandler_spec.rb

```
1 require "spec_helper"
2
3 describe Lita::Handlers::Myhandler, lita_handler: true do
4 end
```

Handler usage

```
1 route(/^echo\s+(.+)/, :echo, command: true, restrict_to: [:testers, :committers], help: {  
2   'echo TEXT' => 'Replies back with TEXT.'  
3 })
```

```
1 http.get "/builds/:id", :build_info  
2  
3 def build_info(request, response)  
4   id = request.env["router.params"][:id]  
5   build = MyBuildSystem.find(id)  
6   response.headers["Content-Type"] = "application/json"  
7   response.write(MultiJson.dump(build))  
8 end
```

ChatOps: Operations meet chat

```
1 on :connected, :greet
2 def greet(payload)
3   target = Source.new(room: payload[:room])
4   robot.send_message(target, "Hello #{payload[:room]}!")
5 end
```

```
1 module Lita
2   module Handlers
3     class HandlerWithConfig < Handler
4       config :api_key
5       route(/call api/, command: true) do |response|
6         response.reply(ThirdPartyAPI.new(config.api_key).call)
7       end
8     end
9     Lita.register_handler(HandlerWithConfig)
10  end
11 end
```

ChatOps: Operations meet chat

pugme.rb

```
1 module Lita
2   module Handlers
3     class Pugbomb < Handler
4       route(/\bpug me\b/i, :pug, command: true, help: {
5         "pug me" => "Display a single pug"
6       })
7       route(/\bpug ?bomb( (\d+))?/i, :bomb, command: true, help: {
8         "pug bomb" => "Display five pugs", "pug bomb N" => "Display N pugs"
9       })
10      route(/how many pugs are there/i, :count, command: true, help: {
11        "how many pugs are there" => "Gets the current number of pugs"
12      })
13
14      BASE_URL = "http://pugme.herokuapp.com"
15
16      def pug(response)
17        data = MultiJson.load(http.get(BASE_URL + "/random").body)
18        response.reply data['pug']
19      end
20    end
  end
end
```

ChatOps: Operations meet chat

```
21  def bomb(response)
22    count = response.matches[0][1] || 5
23    data = MultiJson.load(http.get(BASE_URL + "/bomb", count: count).body)
24    data['pugs'].each do |pug|
25      response.reply pug
26    end
27  end
28
29  def count(response)
30    data = MultiJson.load(http.get(BASE_URL + "/count").body)
31    pug_count = data['pug_count']
32    response.reply "There are #{pug_count} pugs."
33  end
34  end
35  Lita.register_handler(Pugbomb)
36  end
37end
```

Err

- Written in Python
- Runs on top of Python 2.7 and 3.2+
- Created by Guillaume Binet
- Supports multiple chats: Campfire, HipChat, IRC
- Storage (brain) : File
- Supports chat and HTTP routes

ChatOps: Operations meet chat

hello.py

```
1 from errbot import BotPlugin, botcmd
2
3 class HelloWorld(BotPlugin):
4     """Example 'Hello, world!' plugin for Err"""
5
6     @botcmd
7     def hello(self, msg, args):
8         """Say hello to the world"""
9         return "Hello, world!"
```

Examples

Nagios

- bot nagios ack database01/mysqld
- bot nagios check database01/mysqld
- bot nagios status database01/mysqld
- bot nagios downtime database01/mysqld
- bot nagios mute database01/mysqld
- bot nagios unmute database01/mysqld

Deploying

- bot deploy myapp to prod/front01
- bot log me smoke front01
- bot status yellow Issue deploying.
- bot lbctl disable front01
- bot deploy myapp to prod/front01
- bot log me smoke front01
- bot status green Everything is ok.
- bot lbctl enable front01

More examples

- bot pingdom checks
- bot who is on call
- bot conns loadbalancer
- bot graph me -10min @collectd.load(rabbitmq*)
- bot procs unicorn
- bot whois 1.2.3.4
- bot ci status rails/rails
- bot log me hooks front01

ChatOps @ peerTransfer



Jokin Aspiazu 23:00

bot ci build platform



R2D2 BOT 23:00

Going ham on platform/master



bot BOT 23:16

Build #19655 of platform/master was successful

Janky

<http://strong-robot-2524.herokuapp.com/19655/output>

Commit

[db3e79f](#)

ChatOps: Operations meet chat



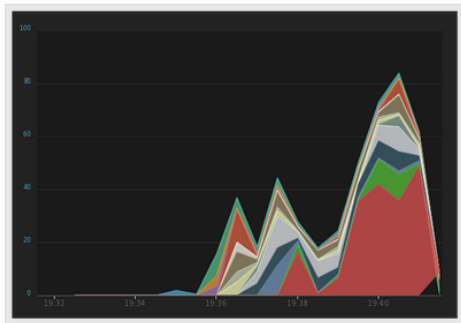
Jose Luis Salas 21:41

bot graph me cpu0 user load over the last 10 minutes



R2D2 BOT 21:41 ★

jose Luis: <http://snapshots.librato.com/instrument/d67q8n4z-14760.png> (50KB) ▼



ChatOps: Operations meet chat



Jose Luis Salas 22:00

bot pager incidents



R2D2 BOT 22:00

No open incidents



Jose Luis Salas 22:01

bot who is on call



R2D2 BOT 22:01

* Vicent Gozalbes is on call for On-call - <https://peeralarm.pagerduty.com/schedules#PULUXNK>

ChatOps: Operations meet chat



Jose Luis Salas 21:46
bot log me school/wadus



R2D2 BOT 21:46
"school/wadus" found 3 matches – <https://papertrailapp.com/events?q=school/wadus>

Apr 11 19:46:12 ext-front-payschool1.infra.peertransfer.com /data/payschool/shared/log/production.log: ActionController::RoutingError (No route matches [GET] "/school/wadus"):

Apr 11 19:46:12 ext-front-payschool1.infra.peertransfer.com /var/log/nginx/payschool.access.log: 10.230.42.125 - - [11/Apr/2015:19:46:06 +0000] "GET /school/wadus HTTP/1.1" 404 625 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.118 Safari/537.36" "90.161.233.142"

Apr 11 19:46:12 ext-front-payschool1.infra.peertransfer.com /var/log/nginx/payschool.access.log: 10.230.42.125 - - [11/Apr/2015:19:46:06 +0000] "GET /stylesheets/404.css HTTP/1.1" 200 454 "<https://www.peertransfer.com/school/wadus>" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2272.118 Safari/537.36" "90.161.233.142"

ChatOps: Operations meet chat



peertransfer.me BOT 18:41

marcoalacot is deploying SHA ea09ce2 to [peertransfer.me](https://github.com/peertransfer/platform/compare/d739b26...ea09ce2) <https://github.com/peertransfer/platform/compare/d739b26...ea09ce2>

Successfully deployed to [peertransfer.me](#) 🤖



Marco Alacot 18:48

staging baywatch smoke



staging BOT 18:48

Running smoke suite in staging jid: 9310

Running smoke suite in staging demo jid: 9313



staging BOT 18:53

smoke suite is GREEN in staging demo

smoke suite is GREEN in staging



peertransfer.com BOT 18:58

marcoalacot is deploying SHA ea09ce2 to [peertransfer.com](https://github.com/peertransfer/platform/compare/d739b26...ea09ce2) <https://github.com/peertransfer/platform/compare/d739b26...ea09ce2>



peertransfer.com BOT 19:08 ★

Successfully deployed to [peertransfer.com](#) 🤖



Marco Alacot 19:08

production baywatch smoke



production BOT 19:08

Running smoke suite in production jid: 20791

Running smoke suite in production demo jid: 20794

smoke suite is GREEN in production

smoke suite is GREEN in production demo



Marco Alacot 19:15



ChatOps: Operations meet chat



Jose Luis Salas 21:50

apache deploy list



apache BOT 21:50

Getting list of deployed issues (jid: 32105)

- 21:50 ★
- Merged 3016 : [origin/feature/3016-random-testimonials](#)
 - Merged 3081 : [origin/bug/3081-hide-cny-id-name-error](#)
 - Merged 3082 : [origin/bug/3082_wrong_resources_links](#)
 - Merged 3084 : [origin/chore/3084-fix-payease-offline](#)



Jose Luis Salas 21:50

apache deploy 3080



apache BOT 21:50

Deploying incremental issues 3080 (jid: 32195)



apache BOT 21:56

Successfully incremental deploy issues 3080 (366.286240053 seconds)

Bibliography

<http://www.pagerduty.com/blog/what-is-chatops/>

<http://blog.librato.com/posts/fire-in-the-hole>

<http://blog.librato.com/posts/confessions-of-a-chatbot>

<http://slides.com/carolnichols/chatops>

<http://tech.toptable.co.uk/blog/2013/11/22/beginning-a-journey-to-chatops-with-hubot/>

<http://docs.lita.io/plugin-authoring/handlers/>

Questions?

Thanks!

Valencia DevOps

April 2015