

翟志军

ChatOps实战

08 OCT 2017

ansible chatops devops



ChatOps概念在国内已经有一些文章谈过，但是都处于理论范畴。而本文则是一篇ChatOps实践的文章。

有必要说明我对ChatOps的理解，ChatOps表面上就是在一个聊天窗口中，发送一个命令给运维机器人bot，然后bot根据我们预定义的操作进行执行，并返回执行结果。至于更深层次的作用，就是将重复性的手工的运维工作自动化了，开发人员、运维人员可以按需执行一些运维操作。

另外，我做到了自动化搭建这一套东西（感谢Github上那么多开源项目，让我少写很多Ansible脚本）。为什么要自动化搭建呢？因为我懒，我不想每次通过一条条shell手工搭建。

本文主题

在RocketChat的聊天窗口中命令Hubot执行一次Jenkins构建任务。

工具介绍

有必要简单说明一下我们此次实现ChatOps的这几个工具。

RocketChat

可以把RocketChat想像成一个具有更多功能的IRC或者微信。它依赖于MongoDB，所以，我们还将自动化安装MongoDB。

如果你了解过Slack的话，它可以作为Slack的开源替代表。

Hubot

Hubot是Github出品的一个运维机器人。本质上就是一个接收命令消息，执行预定义操作的一个程序。而接收命令消息的这个组件在Hubot中被称为Adapter。比如我们希望Hubot接收来自RocketChat聊天窗口里的消息，我们就必须为Hubot安装一个RocketChat的Adapter。市面上，已经有很多Adapter了，我们很少需要自己实现自定义Adapter。

那么，Hubot接收到命令消息后，怎么知道执行哪些操作呢？这部分是需要我们实现了。本质上就是通过正则表达式匹配命令消息，然后操作。实际上通过写Coffescript脚本实现。比如：

```
robot.respond /open the (.*) doors/i, (res) ->
  doorType = res.match[1]
  if doorType is "pod bay"
    res.reply "I'm afraid I can't let you do that."
  else
    res.reply "Opening #{doorType} doors"
```

Jenkins

就这个就不用多介绍了。值得一提是Github已经有不少自动化搭建Jenkins的Ansible脚本了（完全不需要人工干预），本文使用的是geerlingguy的。

Ansible

能让开发人员快速上手的自动化运维工具。我们使用Ansible实现自动化。想简单了解Ansible，可以看看简单易懂Ansible系列——解决了什么。

准备环境

需要准备几台机器：

IP	OS	安装
192.168.61.11	CentOS7	Jenkins,Openresty(for Jenkins)
192.168.61.14	CentOS7	Openresty(for RocketChat)

IP	OS	安装
192.168.61.15	CentOS7	RocketChat Server, MongoDB, Hubot

因为我是在本地做实验的，所以需要在本机虚拟化3台机器。我使用Vagrant + VirtualBox的方式来实现。具体Vagrant如何使用，不在本文讨论范围。你也可以手工在VirtualBox或Vmware上创建相应的虚拟机。Vagrant只不过是自动化了这个过程。Vagrant会基于一个称为 `Vagrantfile` 的文件来创建机器。

Vagrantfile部分内容如下（[想看全文件点这](#)）：

```
Vagrant.configure(2) do |config|

  ANSIBLE_RAW_SSH_ARGS = []
  VAGRANT_VM_PROVIDER = "virtualbox"
  machine_box = "CentOS-7.1.1503-x86_64-netboot"
  config.vm.define "p1" do |machine|
    machine.vm.box = machine_box
    machine.vm.network "private_network", ip: "192.168.61.11"
    machine.vm.provider "virtualbox" do |node|
      node.name = "p1"
      node.memory = 2000
      node.cpus = 2
    end
  end
  ##### 此处省略其它机器的配置
end
```

因为我本地已经存在相应的Vagrant box了，所以，直接使用命令就可以启动这几台机器：

```
vagrant up p1
vagrant up p4
vagrant up p5
```

搭建环境

1. clone 项目

```
git clone https://github.com/zacker330/devops-platform.git
cd devops-platform
```

2. 执行Ansible自动化部署所有的应用及配置

```
ansible-playbook -i chatops-inventory chatops-playbook.yml
```

`chatops-inventory` 是一个类ini文件，用于描述机器，其实就是对机器进行分组。`chatops-playbook.yml` 是一个yaml文件，用于描述如何部署我们的应用及配置。

就这样，我们的Jenkins，RocketChat，Hubot就已经搭建完成了。没错，只需要扫行一条命令。是不是很爽~

RocketChat web客户端：<http://192.168.61.14:3000/>，**初次登录时，需要先注册一个超级管理员**。Jenkins: <http://192.168.61.11/jenkins>，默认账号密码：`admin/admin`

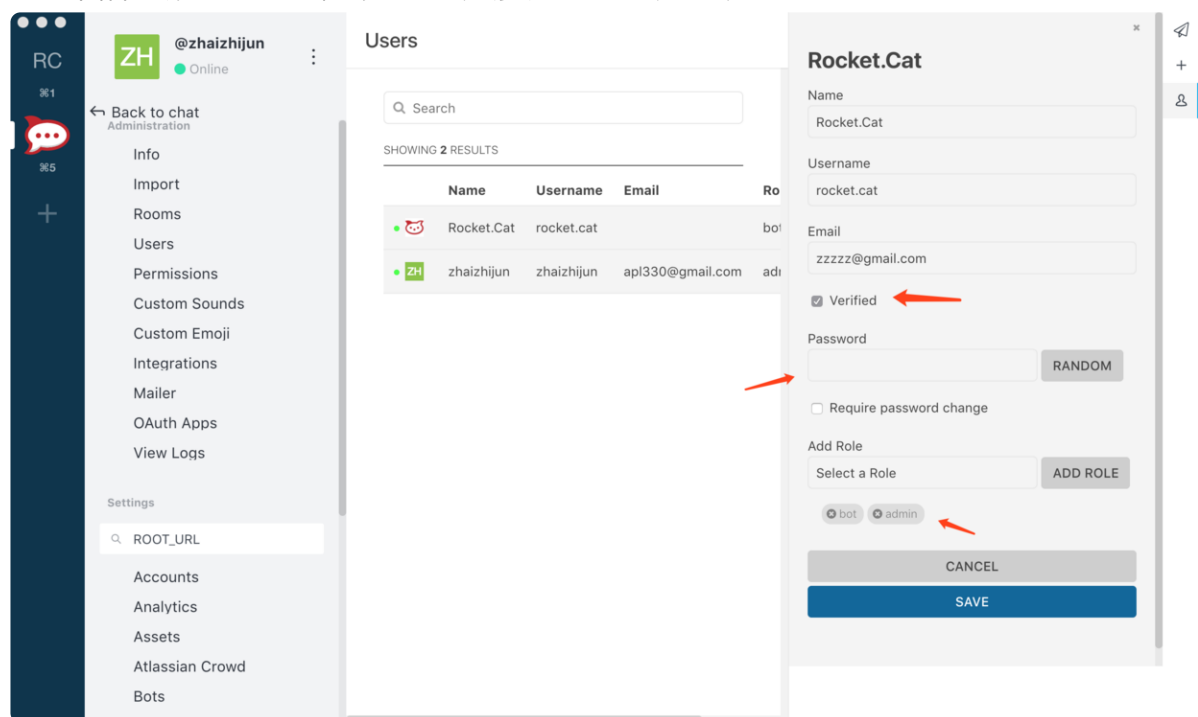
至于是如何搭建的，感兴趣的同学可以看Ansible代码。

以下是集成方法及需要注意的地方：

Hubot与RocketChat集成

1. 设置Hubot运维机器人 现在需要在RocketChat中添加一个User作为运维机器人，我们选择 RocketChat默认用户rocket.cat作为运维机器人，这里需要注意的是：

- rocket.cat必须具有的角色：admin、bot
- rocket.cat必须设置密码，我设置为123456
- 邮箱必须verified，设置时只要勾选上就可以了



2. 安装hubot-rocketchat adapter

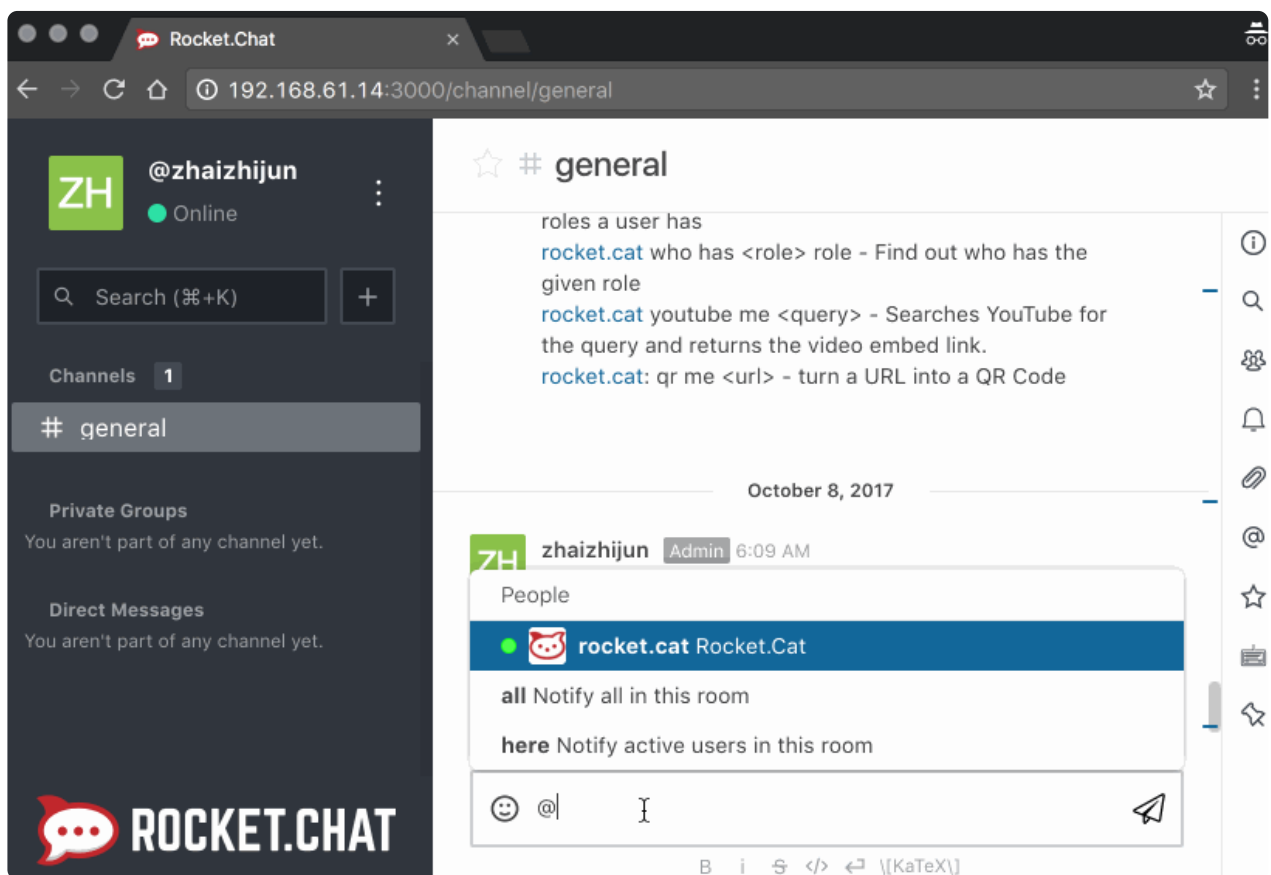
3.

启动时需要指定这几个环境变量以便Hubot能登录上RocketChat:

```
export ROCKETCHAT_URL="http://192.168.61.15:3000"  
export ROCKETCHAT_ROOM=''  
export LISTEN_ON_ALL_PUBLIC=true  
export ROCKETCHAT_USER=rocket.cat  
export ROCKETCHAT_PASSWORD=123456  
export ROCKETCHAT_AUTH=password
```

4.

验证 因为我们安装了hubot-friendly脚本, hey一下hubot, 它有回应, 就说明我们成功集成了RocketChat和Hubot。



Hubot与Jenkins集成

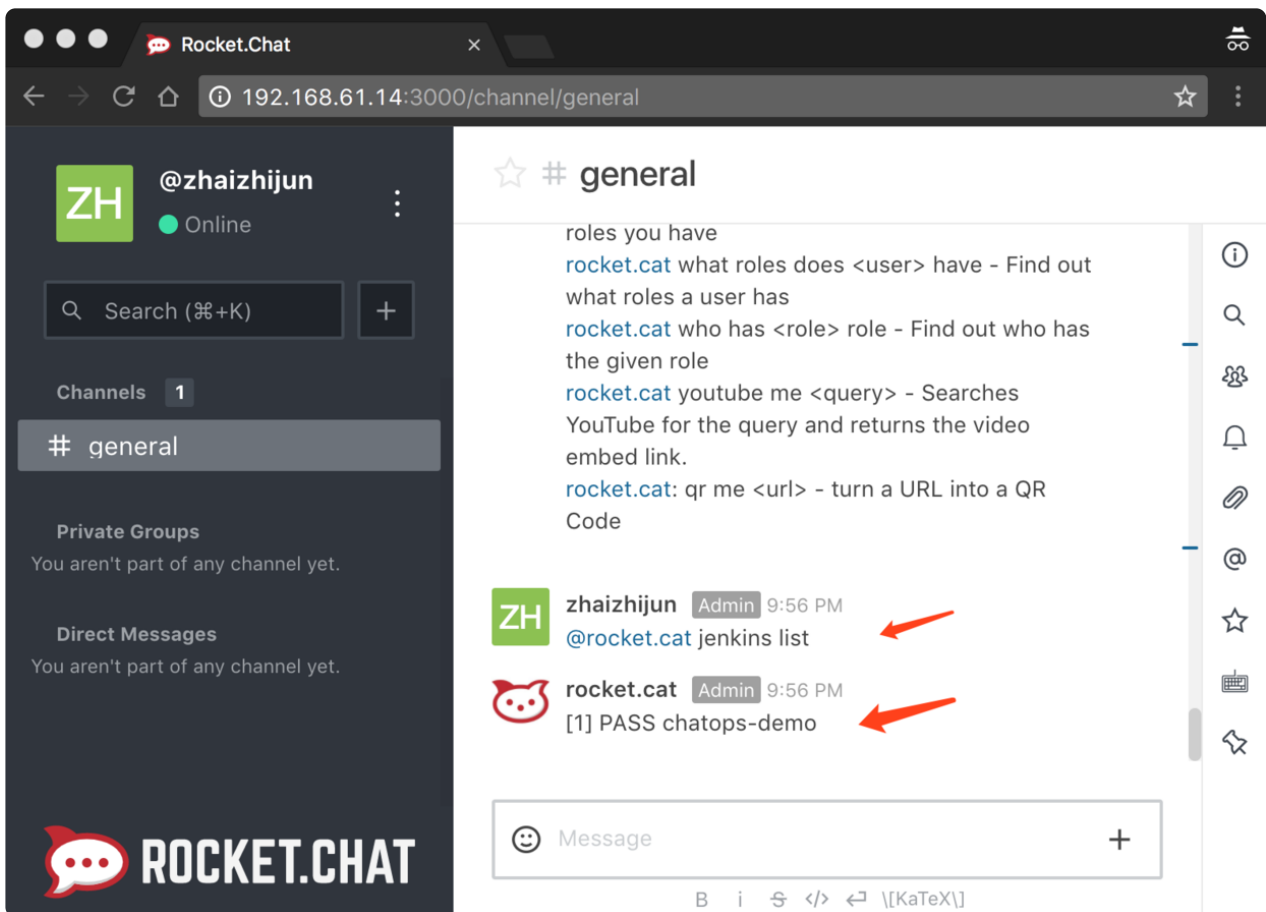
1. 安装hubot脚本: [hubot-jenkins](#)

2. 配置hubot连接Jenkins的环境变量:

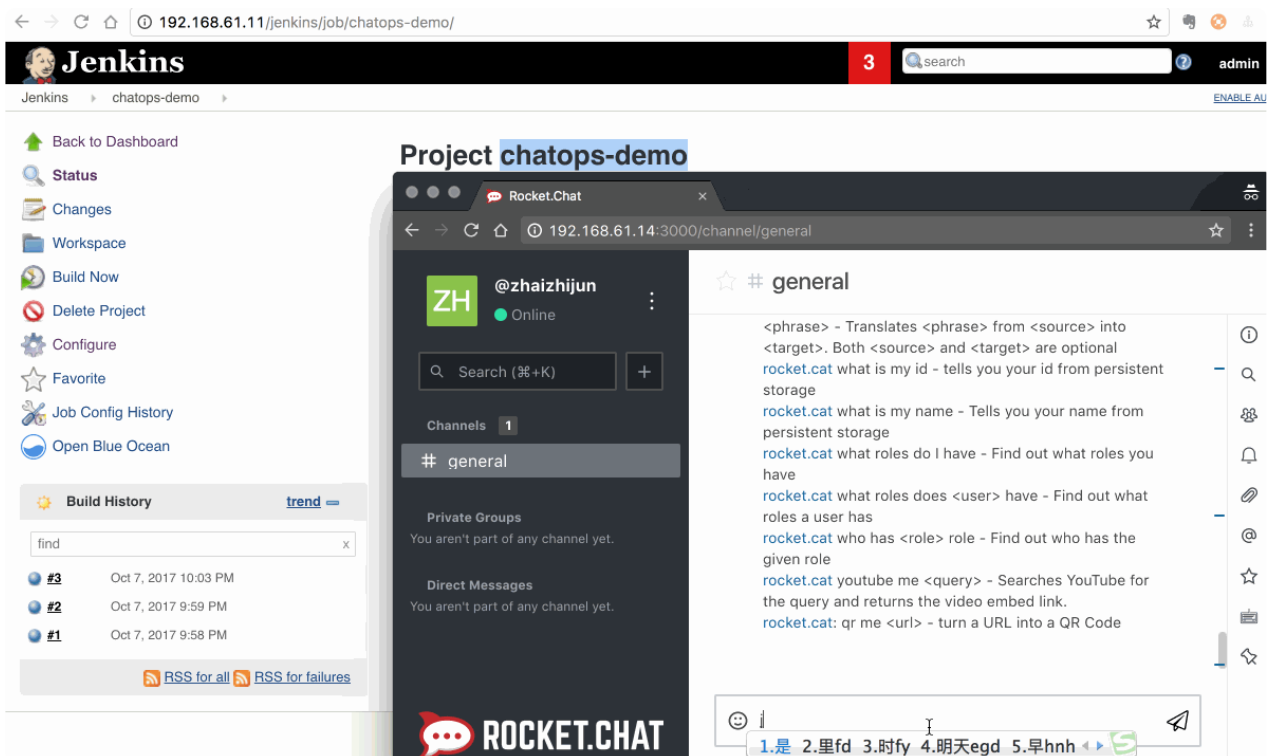
```
export HUBOT_JENKINS_URL=192.168.61.14/jenkins  
export HUBOT_JENKINS_AUTH=admin:admin
```

3.

在RocketChat中, 操作Jenkins的job: 比如列出当前Jenkins的job列表:



再比如执行chatops-demo这个job:



Jenkins与RocketChat集成

Jenkins与RocketChat集成主要用于当Jenkins的job发生变化时主动推送消息到RocketChat中。

1.

在Jenkins中安装Jenkins插件rocketchatnotifier

2. 在系统设置中，设置rocketchatnotifier参数：

Global RocketChat Notifier Settings

Rocket Server URL	<input type="text" value="http://192.168.61.14:3000"/>
Login Username	<input type="text" value="rocket.cat"/>
Login password	<input type="password" value="....."/>
Channel	<input type="text" value="ci"/>
Build Server URL	<input type="text" value="http://192.168.61.11/jenkins"/>
Success <input type="button" value="Test Connection"/>	

3. 在构建job中设置post build action:

chatops-demo Config [Jenkins x]

192.168.61.11/jenkins/job/chatops-demo/configure

Jenkins > chatops-demo >

General Source Code Management Build Triggers Build Environment Build **Post-build Actions**

Post-build Actions

RocketChat Notifications X

- Notify Build Start ☐
- Notify Aborted ☐
- Notify Failure ☒
- Notify Not Built ☐
- Notify Success ☒
- Notify Unstable ☐
- Notify Back To Normal ☐

Advanced...

Add post-build action ▼

如果你使用的是Jenkins pipeline，rocketchatnotifier也支持

```
rocketSend channel: 'general', emoji: ':sob:', message: 'My message', rawMessage: true
```


4. 验证 在Jenkins上手工点击构建按钮，RocketChat的ci channel应该会有消息提醒：

The screenshot displays the Jenkins web interface at 192.168.61.11/jenkins/. The main area shows a table of build jobs. The 'chatops-demo' job is highlighted, showing a successful build with a duration of 32 ms. Below the table, there are links for 'Legend', 'RSS for all', 'RSS for failures', and 'RSS for just latest builds'. On the left sidebar, there are links for 'New Item', 'People', 'Build History', 'Manage Jenkins', 'My Views', and 'Job Config History'. Below the Jenkins interface, the Rocket.Chat interface is shown, displaying the '# ci' channel. The channel contains three messages: a welcome message from 'zhaizhijun' at 6:42 AM, a message from 'rocket.cat' at 6:42 AM stating 'RocketChat/Jenkins plugin: you're all set on http://192.168.61.11/jenkins', and a message from 'rocket.cat' at 6:54 AM stating 'chatops-demo - #9 Success after 0 ms (Open)'. The Rocket.Chat interface also shows a search bar, a list of channels ('# ci', '# general'), and a 'Private Groups' section.

小结

本文如有不足，欢迎来邮讨论。

至此，我们简单的ChatOps框架算是搭好了。剩下的就是根据你们自己业务进行改造了。

另外多说一句：思维模式不应该被职位所局限。

End

为你的收获买单



© 2016 翟志军. All rights reserved.