

UNIVERZITET DŽEMAL BIJEDIĆ U MOSTARU
FAKULTET INFORMACIJSKIH TEHNOLOGIJA MOSTAR

SEMINAR

***Agilni software development,
test & deployment***

Student: *Ernad Husremović, DL 2792*

Mentor: *mr. Adil Joldić*

ver: 0.9.0

Mostar, novembar 2012.

SADRŽAJ

1. Uvod	1
1.1. <i>Gitlab</i> projekat <i>case study</i>	1
1.2. <i>Gitlab</i> arhitektura	1
1.3. <i>Devops</i> metode	2
2. Testno okruženje	3
2.1. <i>Vagrant</i>	3
2.2. Instalacija testnog <i>gitlab</i> servera	3
2.3. <i>Chef cookbooks</i> - knjige recepata za instalaciju sistema	5
2.3.1. <i>Opscode</i> standardni <i>cookbook</i> - <i>mysql</i>	5
2.3.2. Projektni <i>cookbook</i> - <i>gitlab</i>	5
2.3.3. <i>Cookbook</i> principi	6
3. <i>Git</i> i <i>chef</i>	7
3.1. <i>Gitlab</i> "merge" from upstream	7
3.2. Instalacija različitih verzija <i>gitlab</i> -a	10
4. Instalacija na produkcijski server	11
4.1. <i>Rackspace cloud</i>	11
4.2. <i>Bootstrap</i> i instalacija <i>rackspace</i> servera	12
4.2.1. Kako <i>bootstrap</i> funkcioniše	14
5. Zaključak	15
6. Literatura	16
A. <i>Gitlab cookbook</i> parametri	17

Abstract

U ovom radu se na bazi konkretnog primjera¹ prezentuje infrastruktura za testiranje i instalaciju u produkcijskom okruženju ("test" & "deploy" infrastructure). Obradićemo instalaciju "Gitlab" servera koristeći sljedeće tehnologije i servise:

- Github/git <https://github.com/hernad/gitlabhq>
- Vagrant <http://vagrantup.com/>
- Chef opencode <http://www.opscode.com/chef>
- Rackspace cloud <http://www.rackspace.com/cloud>

Keywords: open source software, OSS, chef, vagrant, rackspace, git, github

¹"HOWTO" stil

1. Uvod

1.1. *Gitlab* projekat *case study*

Gitlab je web servis koji obezbeđuje upravljanje softverskim projektima. Gitlab obezbeđuje sljedeće funkcije:

- Git source code management - hostiranje izvornog koda unutar git repozitorija
- Issue management - elektronsko praćenje aktivnosti developera
- Milestone management - pojedine aktivnosti (issues) se mogu vezati uz odgovarajuću verziju (iteraciju) - "milestone"
- Code snippets (code templates) - publikovanje uzoraka izvornog koda koje će tim koristiti
- Code/Commit review - komentarisanje izvornog koda
- Wikies - wikiji omogućavaju timski razvoj projektne dokumentacije

1.2. *Gitlab* arhitektura

"Gitlab" je složen softverski sistem sastavljen od sljedećih komponenti:

1. ubuntu linux server
2. openssh server za razmjenu git repozitorija putem ssh protokola
3. "nginx" web server
4. "postfix" email server
5. "ruby on rails" framework je korišten za izradu web front-end-a.
6. "resque" za background jobs <https://github.com/defunkt/resque>
7. "redis" nosql (koristi ga "resque")
8. "mysql" database backend¹

¹može se koristiti i postgresql

9. "gitolite" perl za upravljanje git repozitorijima
10. "pygments" python "source code highlighter"

1.3. *Devops metode*

Instalacija ovako složene infrastruktura traži značajno vrijeme i resurse za postavljanje, kako testnog tako i produkcijskog okruženja. Agilni pristup potencira mogućnost brzih promjena softverskog sistema. Konvencionalne metode instalacije i održavanja (manuelne procedure) su u slučajevima ovako kompleksnih sistema neadekvatne u tom kontekstu. Agilni pristup ovom problemu potencira automatizaciju testnog i produkcijskog okruženja. Manuelne operacije se zamjenjuju atomatiziranim procedurama. Operacije instalacije i održavanja sistema i same postaju sofverski projekti. Agilnog pristupa ovoj tematici je rezultirao "Devops"² metodama i praksama instalacije.

²<http://en.wikipedia.org/wiki/DevOps>

2. Testno okruženje

2.1. Vagrant

”Chef” je klasičan primjer ”devops” alata. ”Chef” recepti (”cookies”) sadrže programe kojim se instaliraju pojedine komponente sistema. Skup tih recepata definiše instalacijsku proceduru.

Naravno, produkcijskoj instalaciji sistema¹ predhodi testiranje. ”Vagrant” je popularni alat koji su mnogi ”devops” pobornici prihvatili kao alat za izgradnju testnog okruženja. ”Vagrant” za automatizaciju koristi ”Virtual Box” virtuelne sesije². Konfiguracija tih sesija se vrši sa ”chef configuration management” sistemom³.

2.2. Instalacija testnog gitlab servera

Preuzmimo vagrant_gitlab projekat sa Github-a:

```
$ git clone git://github.com/hernad/vagrant_gitlab.git
```

Install skripta instalira sve potrebne ruby pakete koje će vagrant koristiti⁴, kao i sve ”cookbooks” koje će koristiti ”chef”:

```
vagrant_gitlab$ ./install.sh
```

Vagrantfile⁵ sadrži konfiguraciju testnog gitlab servera. Vagrantfile nije ništa drugo nego ”ruby” program. Konkretno, vagrant_gitlab koristi niz sistemskih varijabli (OS environment) kao parametre instalacije sistema⁶

Komandom `up.sh` setujemo sistemske varijable, te kreiramo virutelnu sesiju:

```
vagrant_gitlab$ source up.sh
```

```
OS_AUTH_SYSTEM=rackspace
OS_AUTH_URL=https://identity.api.rackspacecloud.com/v2.0/
OS_DNS_DOMAIN=test.out.ba
```

¹instalaciji sistema namjenjenoj krajnjem korisniku

²<https://www.virtualbox.org>

³može da se koristi i ”Puppet” configuration managment sistem

⁴Naravno, prije toga je potrebno imati instalirano ”ruby” programsko okruženje, preporučeno ”rvm” ruby okruženje

⁵https://github.com/hernad/vagrant_gitlab/blob/master/Vagrantfile

⁶Parametri sadrže sve ono što su privatni podaci korisnika i što ne treba biti javno dostupno - imena hostova, korisnika, lozinke i sl.

```

OS_ENVARS='GMAIL_USER=bring.out.sa@gmail.com GMAIL_PASSWORD=pwd1 ↵
  MYSQL_ROOT_PWD=pwd2 MYSQL_PWD=pwd3 GITHUB_USER=hernad GITHUB_PROJECT↵
  =vagrant_gitlab OS_SERVER_NAME=gitlab-stable.test.out.ba ↵
  OS_DNS_DOMAIN=test.out.ba GITLAB_VERSION=v3.0.2 GITLAB_REPOS=git://↵
  github.com/hernad/gitlabhq.git'
OS_NO_CACHE=1
OS_PASSWORD=<rackspacecloud_api_key>
OS_PROJECT_ID=<rackspace_user>
...

```

Nakon kreiranja gitlab "VirtualBox" testne sesije, slijedi pokretanje "chef-solo" procesa unutar sesije. "chef-solo"⁷ je zadužen za instalaciju i konfiguraciju testne sesije:

```
vagrant_gitlab$ source ./up.sh
```

```

[default] Clearing any previously set forwarded ports...
[default] Forwarding ports...
[default] -- 22 => 2222 (adapter 1)
[default] Creating shared folders metadata...
...
[default] Waiting for VM to boot. This can take a few minutes. <<<<<<<< ↵
  virtualbox sesija kreirana
[default] VM booted and ready for use!
[default] Detected Virtualbox Guest Additions 4.2.4 --- OK.
[default] Configuring and enabling network interfaces...
[default] Mounting shared folders...
[default] -- v-root: /vagrant
[default] -- v-csc-1: /tmp/vagrant-chef-1/chef-solo-1/cookbooks
[default] Running provisioner: Vagrant::Provisioners::ChefSolo...
[default] Generating chef JSON and uploading...
[default] Running chef-solo... <<<<<<<<<<<< Pokrece se chef-solo na ↵
  virtualbox sesiji
stdin: is not a tty
[Mon, 19 Nov 2012 11:33:15 +0000] INFO: *** Chef 0.10.10 ***
[Mon, 19 Nov 2012 11:33:15 +0000] INFO: Setting the run_list to ["↵
  recipe[build-essential]", "recipe[rvm::system]", "recipe[mysql::↵
  client]", "recipe[mysql::ruby]", "recipe[postgresql::client]", "↵
  recipe[postgresql::ruby]", "recipe[gitlab::default]", "recipe[gitlab↵
  ::nginx]", "recipe[gitlab::database]"] from JSON
[Mon, 19 Nov 2012 11:33:15 +0000] INFO: Run List is [recipe[build↵
  essential], recipe[rvm::system], recipe[mysql::client], recipe[mysql↵
  ::ruby], recipe[postgresql::client], recipe[postgresql::ruby], ↵
  recipe[gitlab::default], recipe[gitlab::nginx], recipe[gitlab::↵
  database]]
[Mon, 19 Nov 2012 11:33:15 +0000] INFO: Run List expands to [build↵
  essential, rvm::system, mysql::client, mysql::ruby, postgresql::↵

```

⁷ Chef-solo koristi lokalno instalirane "cookbooks". Standardna opcija je client-server, u kojoj chef agent (klijent) koji sa centralog chef-servera uzima "cookbooks"

```

client, postgresql::ruby, gitlab::default, gitlab::nginx, gitlab::←
database]
[Mon, 19 Nov 2012 11:33:15 +0000] INFO: Starting Chef Run for precise32
...
[Mon, 19 Nov 2012 11:36:20 +0000] INFO: Processing package[ssl-cert] ←
action install (/tmp/vagrant-chef-1/chef-solo-1/cookbooks/rvm/←
providers/ruby.rb line 156)
[Mon, 19 Nov 2012 11:36:22 +0000] INFO: package[ssl-cert] installed ←
version 1.0.28ubuntu0.1
[Mon, 19 Nov 2012 11:36:22 +0000] INFO: Building rvm_ruby[ruby-1.9.3-←
p327], this could take awhile...
...
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: Processing service[nginx] ←
action reload (nginx::default line 39)
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: service[nginx] reloaded
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: template[/etc/init.d/gitlab] ←
sending enable action to service[gitlab] (delayed)
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: Processing service[gitlab] ←
action enable (gitlab::default line 183)
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: Chef Run complete in 994.730999←
seconds
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: Running report handlers
[Mon, 19 Nov 2012 14:18:58 +0000] INFO: Report handlers complete

```

Nakon 15-tak minuta imamo u potpunosti instaliran "gitlab" server.

2.3. *Chef cookbooks* - knjige recepata za instalaciju sistema

2.3.1. Opscode standardni cookbook - mysql

Veliki broj "chef" knjiga recepata već postoji spreman. Uzmimo primjer `mysql8` cookbook-a. Sve što je potrebno jesu tri jednostavna koraka:

1. Uvrsti [mysql cookbook](#) u knjigu recepata koje gitlab projekat koristi
2. Uključi "recipe" u [Vagrantfile](#)
3. Definiši [parametre instalacije](#)

2.3.2. Projektni cookbook - gitlab

Za potrebe našeg projekta pravimo sopstveni cookbook⁹ (ili više njih, vodeći se "reusable" principima).

⁸<http://community.opscode.com/cookbooks/mysql>

⁹https://github.com/hernad/chef_gitlab

Unutar njega koristimo komande iz standardnih cookbook-ova, kao što je gore navedeni mysql ili jednostavno navodimo operacije koje su potrebne za instalaciju i konfiguraciju naše aplikacije.

2.3.3. Cookbook principi

Kod izrade cookbook-a treba imati na umu sljedeće principe:

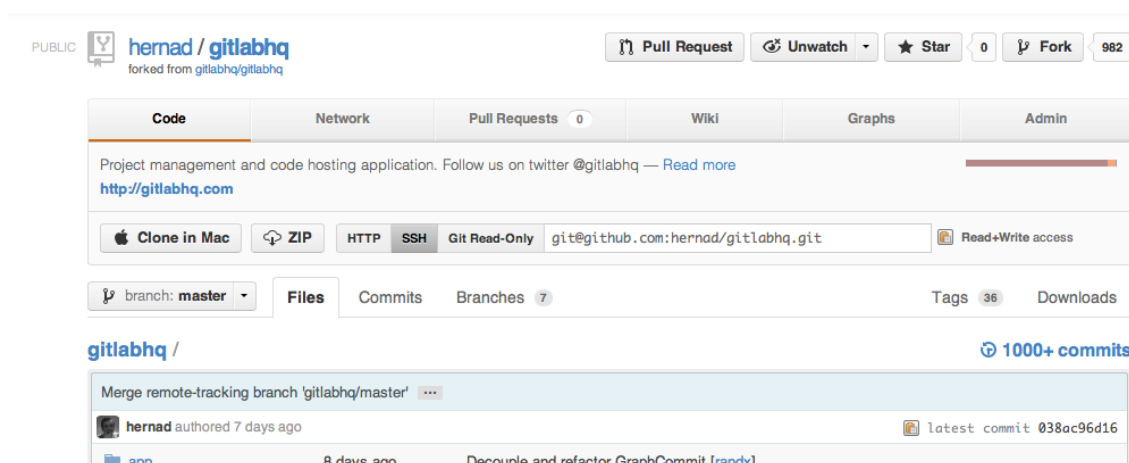
- Idempotentnost - cookbook se može bez problema pokrenuti više puta, ishod instalacije treba biti "identičan"
- Multi-environment pristup - predvidjeti sva ciljna okruženja (testno/produksijsko, operativni sistemi, ciljni servis provajderi). Tehnički, ovaj princip se realizira parametrizacijom
- Parametrizacija - Svi parametri sistema koji su predmet promjena prilikom različitih scenarija instalacije treba parametrizirati.

3. *Git i chef*

U ovom poglavlju ćemo demonstrirati efekte pravilne parametrizacije "cookbook"-ova. Takođe ćemo demonstrirati korištenje git repozitorija unutar chef receptata. "Chef", treba li to uopšte naglasiti, ima odličnu podršku za git repozitorije.

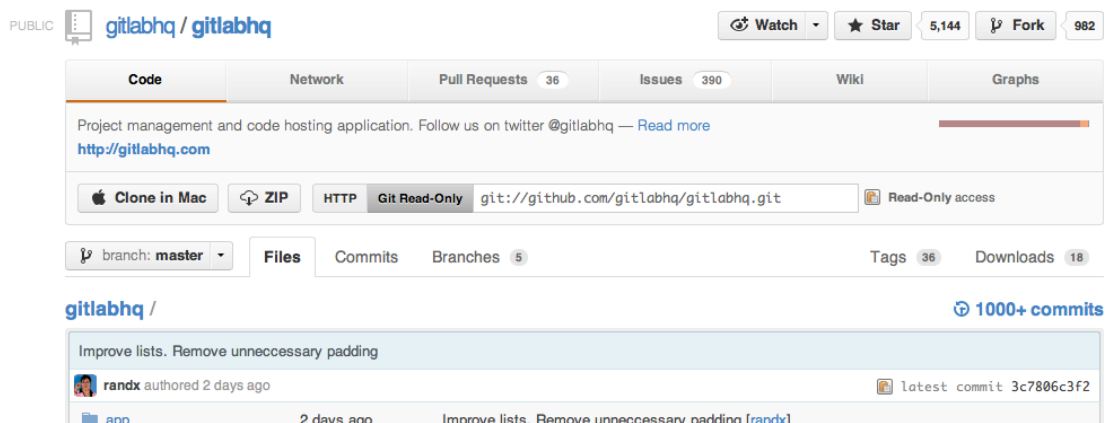
3.1. Gitlab "merge" from upstream

Ključni software repozitorij je naravno izvorni kod same gitlab aplikacije <https://github.com/hernad/gitlabhq>. Ovaj repozitorij predstavlja "fork" glavnog - "upstream" projekta.



Slika 3.1: Fork "gitlabhq" od strane developera "hernad"

Glavni ("upstream") repozitorij projekta nalazi se na <https://github.com/gitlabhq/gitlabhq>.



Slika 3.2: Gitlabhq "upstream" projekat

Git "merge proces" omogućava spajanje različitih branch-ova unutar jednog ili više git repozitorija (Husremović, 2012). Definišimo lokaciju upstream repozitorija:

```
gitlabhq$
```

```
git remote add gitlabhq git://github.com/gitlabhq/gitlabhq.git
```

Preuzmimo promjene sa upstream-a:

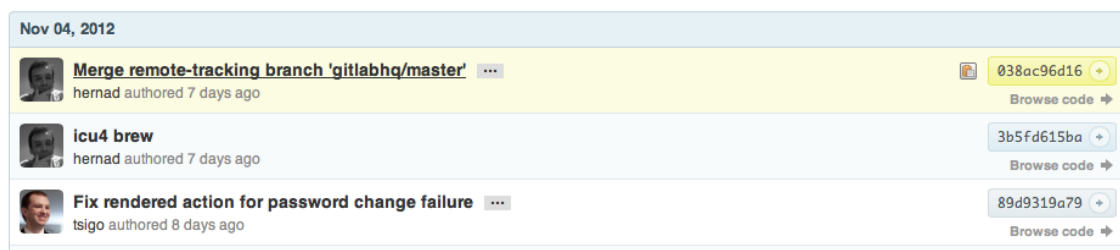
```
gitlabhq$ git fetch gitlabhq
```

```
remote: Counting objects: 570, done.
remote: Compressing objects: 100% (174/174), done.
remote: Total 342 (delta 262), reused 237 (delta 165)
Receiving objects: 100% (342/342), 43.70 KiB, done.
Resolving deltas: 100% (262/262), completed with 122 local objects.
From git://github.com/gitlabhq/gitlabhq
 dd4d124..3c7806c master -> gitlabhq/master
```

Pogledajmo koji je naš trenutni branch:

```
gitlabhq$ git branch -l
```

```
* master
```



Slika 3.3: "gitlabhq" repozitorij - hernad fork prije merge operacije

Naš lokalni branch ne sadrži promjene koje su na "upstream" projektu rađene posljednjih par dana. "Merge" operacijom te promjene ubacujemo u naš repozitorij:

```
gitlabhq$ git merge gitlabhq/master
```

```

Removing lib/gitlab/encode.rb
Removing gitlab
Auto-merging doc/development.md
Removing Procfile.production
Merge made by the recursive strategy
.travis.yml | 2 +-
CHANGELOG | 2 +-
....
VERSION | 2 +-
app/assets/images/event_filter_comments.png | Bin 0 -> 750 bytes
app/assets/images/event_filter_merged.png | Bin 0 -> 463 bytes
app/assets/images/event_filter_push.png | Bin 0 -> 632 bytes
...
app/controllers/application_controller.rb | 9 +++
app/controllers/blob_controller.rb | 10 +--
app/controllers/dashboard_controller.rb | 11 +-
app/controllers/profile_controller.rb | 2 +-
...
app/views/blame/show.html.haml | 4 +-
app/views/commits/_commit.html.haml | 4 +-
app/views/commits/_head.html.haml | 5 ++
app/views/dashboard/index.html.haml | 9 +-
...

delete mode 100644 Procfile.production
create mode 100644 app/assets/images/event_filter_comments.png
create mode 100644 app/assets/images/event_filter_merged.png
...
delete mode 100644 lib/gitlab/encode.rb
create mode 100644 lib/gitlab/git_stats.rb
create mode 100644 vendor/assets/javascripts/g.bar-min.js
create mode 100644 vendor/assets/javascripts/g.raphael-min.js

```

Nije bilo nikakvih konflikata, tako da je automatsko merđžiranje bilo dovoljno. U slučaju konflikata, moralo bi se obaviti ručno spajanje fjalova koji su u konfliktu¹.

Pushiramo promjene na "origin" remote lokaciju, što je ustvari fork repozitorij githab-a developera "hernad": `gitlabhq$ git push origin master`

```

Counting objects: 581, done.
Delta compression using up to 8 threads.
Compressing objects: 100\% (85/85), done.
Writing objects: 100\% (350/350), 45.00 KiB, done.
Total 350 (delta 267), reused 342 (delta 262)
To git@github.com:hernad/gitlabhq.git

```

¹Npr. ako se i na "hernad" fork objektu radilo na nekom source fajlu, git prijavljuje konflikt i upućuje na ručne ispravke tog fajla da bi se merđžiranje završilo

```
038ac96..28f5807 master -> master
```

3.2. Instalacija različitih verzija gitlab-a

Kod izgradnje "gitlab" cookbook-a su predviđeni parametri "gitlab_url" i "gitlab_branch". Unutar Vagrantfile-a se ti parametri popunjavaju sa varijablama GITLAB_VERSION i GITLAB_REPOS.

One su ugniježdene u OS envar "OS_ENVARS"².

Definišimo instalaciju verzije 3.0.2³, sa repozitorija "hernad":

```
OS_ENVARS='... GITLAB_VERSION=v3.0.2 GITLAB_REPOS=git://github.com/↔  
hernad/gitlabhq.git'
```

Rezultat će biti instalacija verzije 3.0.2 gitlab aplikacije na ciljni server. Ako želimo instalirati posljednju verziju, jednostavno navodimo GITLAB_VERSION=master. Na taj način možemo na testni server instalirati posljednju verziju, a na produkcijski posljednju stabilnu verziju (npr. v3.0.2):



Slika 3.4: Gitlabhq nakon 'merge' operacije sa 'upstream' projektom - "master" branch je verzija 3.1.0-pre

²Ovo "ugnježdvanje" je napravljeno iz tehničkih razloga. Naime ti parametri nam trebaju biti dostupni i kod instalacije udaljenog produkcijskog servera (Instalacija na rackspace cloud).

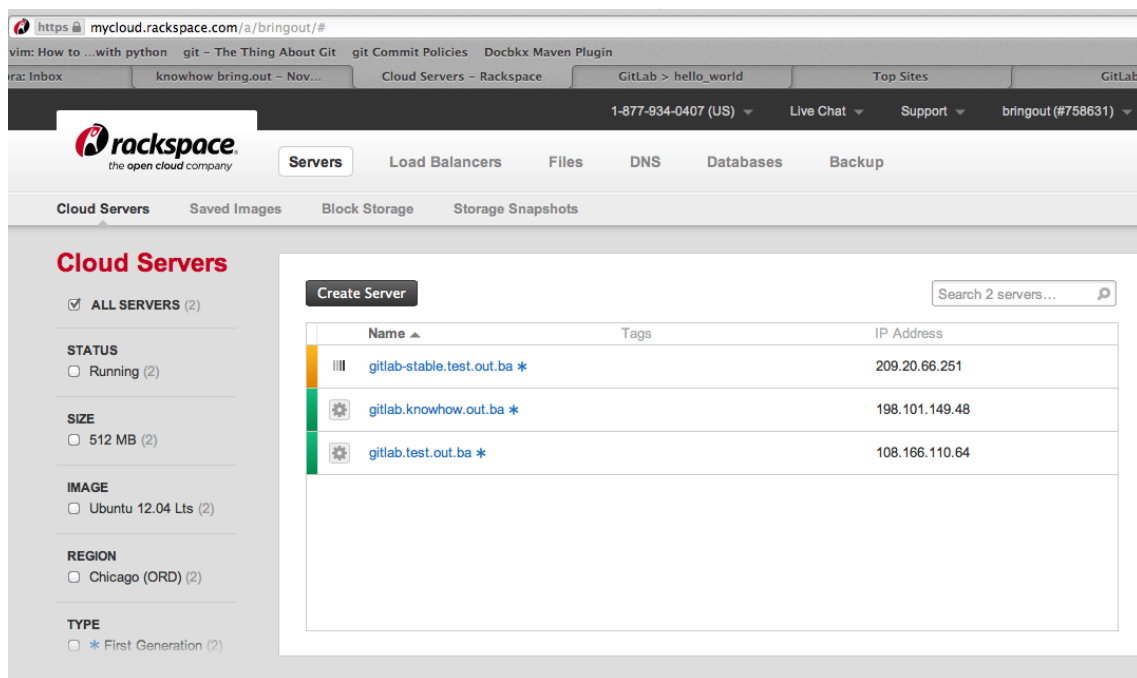
³U repozitoriju se nalazi git tag "v3.0.2"

4. Instalacija na produkcijski server

Jednostavna uspostava testnog okruženja uz pomoć Vagrant-a je bitna stvar. Međutim, ako se produkcijsko okruženje mora uspostaviti manuelno, nismo dobili previše na agilnosti. Dobra je stvar da se "Chef" može primijeniti i na "produkcijsko" okruženje bez ikakvih značajnih intervencija.

4.1. Rackpace cloud

Mi ćemo za produkcijsko okruženje koristiti rackspace cloud:



Slika 4.1: Rackpace cloud web upravljačka konzola

"Rackpace" je odabran zato što postoji kvalitetan developerski API¹ za upravljanje svim cloud resursima. "Chef" recepti nas vode korak do potpune automatske instalacije produkcijskog servera.

¹<http://docs.rackspace.com/api/>

4.2. *Bootstrap* i instalacija rackspace servera

U odnosu na naše "Vagrant" testno okruženje trebamo sljedeće nove funkcije:

- Kreirati rackspace server
- Instalirati na njega chef radno okruženje²
- Pokrenuti željene chef cookbooks, sa zadanim parametrima (OS_ENVARS)³
- Registrovati novokreirani server na rackspace DNS sistemu⁴

Sve ove funkcije realizira "ubuntu_bootstrap_chef"⁵ projekat.

Komanda `setup Rack Server` u jednom koraku realizira sve gornje korake:

```
Server: gitlab.knowhow.out.ba IP: 198.101.149.48
vrsim update domena knowhow.out.ba, zapis gitlab.knowhow.out.ba sa ↵
  javnom adresom 198.101.149.48
./manage_domains.py -d knowhow.out.ba -r gitlab.knowhow.out.ba -i ↵
  198.101.149.48
dns_domain: knowhow.out.ba dns_record: gitlab.knowhow.out.ba ip_address↵
  198.101.149.48

....

[2012-11-12T11:15:43+00:00] INFO: template[/etc/rvmrc] mode changed to ↵
  644
[2012-11-12T11:15:43+00:00] INFO: Processing execute[install system↵
  wide RVM] action run (rvm::system_install line 76)
[2012-11-12T11:15:43+00:00] INFO: Processing execute[upgrade system↵
  wide RVM to none] action run (rvm::system_install line 110)
[2012-11-12T11:15:43+00:00] INFO: Processing rvm_ruby[ruby-1.9.3-p327] ↵
  action install (rvm::system line 170)
/root/vagrant_gitlab/cookbooks/rvm/libraries/rvm_chef_user_environment.↵
  rb:36: warning: class variable access from toplevel
[2012-11-12T11:15:45+00:00] INFO: Processing package[build-essential] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
  line 156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[openssl] action ↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
  156)
```

²https://github.com/hernad/ubuntu_bootstrap_chef/blob/master/bootstrap.sh

³https://github.com/hernad/ubuntu_bootstrap_chef/blob/master/run_solo.sh

⁴https://github.com/hernad/ubuntu_bootstrap_chef/blob/master/rackspace/manage_domains.py

⁵https://github.com/hernad/ubuntu_bootstrap_chef

```
[2012-11-12T11:15:45+00:00] INFO: Processing package[libreadline6] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[libreadline6-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[zlib1g] action ↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[zlib1g-dev] action↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[libssl-dev] action↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[libyaml-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
...
[2012-11-12T11:15:45+00:00] INFO: Processing package[libsqlite3-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:45+00:00] INFO: Processing package[sqlite3] action ↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[libxml2-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[libxslt-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:46+00:00] INFO: package[libxslt-dev] is a virtual ↵
  package, actually acting on package[libxslt1-dev]
[2012-11-12T11:15:46+00:00] INFO: Processing package[autoconf] action ↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[libc6-dev] action ↵
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ↵
    156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[ncurses-dev] ↵
  action install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb↵
    line 156)
[2012-11-12T11:15:46+00:00] INFO: package[ncurses-dev] is a virtual ↵
  package, actually acting on package[libncurses5-dev]
```



```

[2012-11-12T11:15:46+00:00] INFO: Processing package[automake] action ←
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ←
  156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[libtool] action ←
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ←
  156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[bison] action ←
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ←
  156)
[2012-11-12T11:15:46+00:00] INFO: Processing package[ssl-cert] action ←
  install (/root/vagrant_gitlab/cookbooks/rvm/providers/ruby.rb line ←
  156)
[2012-11-12T11:15:47+00:00] INFO: Building rvm_ruby[ruby-1.9.3-p327], ←
  this could take awhile...

....

bootstrap chef for vagrant_gitlab is finished (0.9.9 :)
restartujem server ...
Restartujem server gitlab.knowhow.out.ba

real15m29.472s
user0m3.928s
sys0m0.387s

```

Nakon 30-tak minuta naš novokreirani rackspace server je konfigurisan i spreman da odmah obrađuje zahtjeve internet korisnika !

4.2.1. Kako *bootstrap* funkcioniše

Generalno, bootstrap projekat nakon instalacije servera preuzima cookbooks iz projekta zadatog parametrom `GITHUB_PROJECT=vagrant_gitlab`. To znači da je ova procedura primjenljiva i na različite projekte (npr. `GITHUB_PROJECT=vagrant_redmine`), naravno pod uslovom da se u njima obezbjedi sadržaj i struktura kakvu bootstrap okruženje traži.

U "cookbook" projektu se moraju nalaziti [solo.rb](#) i [node.json](#). Oni imaju identičnu funkciju kakav ima Vagrantfile u testnoj instalaciji⁶

⁶Lahko je uočiti, da je [roles/gitlab.rb](#) dobijen sa copy-paste sadržaja [Vagrantfile-a](#)

5. Zaključak

Agilni softverski projekat je sposoban brzo reagovati na promjene (J. Shore i S. Warden, 2008). Da bi to bilo moguće, agilnost se mora primjeniti u svim fazama životnog ciklusa softverskog projekta.

Devops prakse i procedure su zadatke instalacije, konfiguracije i nadogradnje softverskog rješenja pretvorile u zasebne softverske projekte. Zadaci koji se pred sistemске inženjere postavljaju kod instalacije *cloud* infrastrukture daju agilnim metodama još veći značaj. Agilne *devops* prakse su već sada "must be" pristup građenju IT sistema u *cloud*-u. Treba imati na umu da ovaj koncept već sada "ulazi" i u konvencionalna IT odjeljenja. Brzi trend integracije privatne i javne *cloud* infrastrukture će bitnost ovog pristupa *software deployment* operacijama još više uvećati.

6. Literatura

Ernad Husremović. *Agilni software development, Git SCM*, 2012.

J.Shore i S.Warden. *The Art of Agile Development*. O'Reilly, 2008.

Dodatak A

Gitlab *cookbook* parametri

Cookbook parametre možemo smjestiti u `~/ .bash_profile`. Primjer sadržaja:

```
export OS_SERVER_NAME="gitlab-stable.test.out.ba"
export OS_DNS_DOMAIN="test.out.ba"

export GITLAB_REPOS="git://github.com/hernad/gitlabhq.git"
export GITLAB_VERSION="v3.0.3"

# OS_ENVARS sadrzi sve varijable koje chef-solo treba kod ↔
# konfiguracije remote sistema
OS_ENVARS="GMAIL_USER=bring.out.sa@gmail.com_GMAIL_PASSWORD=↔
pwd_bring_sa"
OS_ENVARS="$OS_ENVARS_MYSQL_ROOT_PWD=pwd_my_root_MYSQL_PWD=pwd_my"
OS_ENVARS="$OS_ENVARS_GITHUB_USER=hernad_GITHUB_PROJECT=vagrant_gitlab"
OS_ENVARS="$OS_ENVARS_OS_SERVER_NAME=$OS_SERVER_NAME_OS_DNS_DOMAIN=↔
$OS_DNS_DOMAIN"
OS_ENVARS="$OS_ENVARS_GITLAB_VERSION=$GITLAB_VERSION_GITLAB_REPOS=↔
$GITLAB_REPOS"
export OS_ENVARS
```

Značenje parametara:

- OS_SERVER_NAME - hostname servera
- OS_DNS_DOMAIN - pripadnost servera DNS domeni (koristi za automatski update rackspace DNS sistema)
- GITLAB_REPOS, GITLAB_VERSION - git repozitorij "gitlab" projekta, verzija (branch, tag) koja se instalira
- GMAIL_USER, GMAIL_PASSWORD - postfix email ser podešava za relaying preko gmail-a
- GITHUB_USER, GITHUB_PROJECT - odavde se preuzimaju cookbook-ovi
- MYSQL_ROOT_PWD, MYSQL_PWD - root password mysql servera, password "gitlab" mysql usera