

ひでお (仮)

こんにちは

- にくです
- コンサドーレ札幌が大好きです
- @niku_name
- <http://niku.name/>

コンサドーレホーム開幕戦情報

3/9(日)13:00 @ 札幌ドーム vs モンテディオ山形



ひでおとは

- 「ひ」 とり
- 「で」 ぶ (Dev)
- 「お」 ぶす (Ops)

ひとりとは

- ひとりです
- 開発してもひとり
- 運用してもひとり

DevOpsおさらい

- “Dev”elopment(開発)
- “Op”eration(運用)

を繋いだ言葉です

DevOpsってなんなの

- 開発 「（多少動かなくても）機能をのせていきたい」
- 運用 「（多少機能が少なくても）ちゃんと動くようにしたい」

開発と運用はあんまり相性がよくないようにみえる

DevもOpsも向かう先は一緒だった

- 開発 「（プロダクトをよりよくするため）（多少動かなくても）機能をのせていきたい」
- 運用 「（プロダクトをよりよくするため）（多少機能が少なくても）ちゃんと動くようにしたい」

プロダクトをよりよくする

- プロダクト全体の最適化
- 全体を良くするにはお互いを理解する

DevOpsは壮大なストーリー

- 大きすぎて今回はうまく話せないことがわかった
- すまん

ひでお

- ひとりで開発と運用をやるのでお互いは理解できている
- プロダクトをよりよくするため
 - 運用の圧縮
 - 人が増えたときのノウハウの共有
 - 記憶や手順書からの脱却

ひでお(旅立ち編)

- CI環境作るためのツール
 - VirtualBox
 - Vagrant
 - Packer
 - Chef

ケース

- ソースコードをGithubに置ける
- ソースコードをGithubに置けない

ソースコードをGithubに置く

- Github連携サービスを使おう
 - Travis CI
 - Circle CI
- サービスの面倒をみなくていい
- おすすめ

ソースコードをGithubに置けない

- Jenkinsを使おう
- 面倒をみないといけない
- 運用の手間を減らすために運用対象を増やすの？

既にJenkinsが使える

- おめでとう！
- 使い倒そう

サーバーあるけどJenkins入ってない

```
$ wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -  
$ sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'  
$ sudo apt-get update  
$ sudo apt-get install jenkins
```

[https://wiki.jenkins-ci.org/
display/JENKINS/Installing
+Jenkins#InstallingJenkins-Unix
%2FLinuxInstallation](https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins#InstallingJenkins-Unix%2FLinuxInstallation)

サーバーがない

- 仮想環境に作ろう
 - 実際のサーバー用意するより作ったり壊したりが簡単

仮想環境を作る

- 必要なもの
 - 自分のPC
 - 仮想化ソフト
 - インストールしたことある人に聞ける環境か調べる根性

仮想化ソフト

- インストーラーでインストールできる
 - VirtualBox
 - <https://www.virtualbox.org/>
 - VMWare Fusion(Mac) / VMWare Player(Windows)
 - <https://my.vmware.com/jp/>
- 個人的にはVirtualBoxを使って

仮想化環境を作ろう

- 仮想化ソフトとは、使っていないPC（ハードウェア）のようなもの
- 中身（OS）をインストールする

CI(Jenkins)を入れる

```
$ wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -  
$ sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'  
$ sudo apt-get update  
$ sudo apt-get install jenkins
```

[https://wiki.jenkins-ci.org/
display/JENKINS/Installing
+Jenkins#InstallingJenkins-Unix
%2FLinuxInstallation](https://wiki.jenkins-ci.org/display/JENKINS/Installing+Jenkins#InstallingJenkins-Unix%2FLinuxInstallation)

仮想化環境でのCIできた

- もっともっと
- 簡単にノウハウを配布したい
- 記憶や手順書からの脱却

Vagrant

- <http://www.vagrantup.com/>
- Vagrantfileに設定を書く
 - あらかじめ仮想化してある環境(Box)の取得
 - ↑の環境へ事前処理(Provision)を適用
- 仮想環境の起動/終了を簡単なコマンドから

Vagrantのインストール

- インストーラーで簡単にインストールできる
- <http://www.vagrantup.com/downloads.html>
- Windows/OSX/Debian/RHEL

Vagrantfile

```
VAGRANTFILE_API_VERSION = "2"

Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "hideo"
  config.vm.box_url = "http://files.vagrantup.com/precise64.box"
  config.vm.network "forwarded_port", guest: 8080, host: 18080
  config.vm.provision "shell", inline: <<__EOS__
wget -q -O - http://pkg.jenkins-ci.org/debian/jenkins-ci.org.key | sudo apt-key add -
sudo sh -c 'echo deb http://pkg.jenkins-ci.org/debian binary/ > /etc/apt/sources.list.d/jenkins.list'
sudo apt-get update
sudo apt-get install jenkins -y
__EOS__
end
```

帰ったらやってみよう Vagrant

- Vagrantfileを書いて
- `vagrant up`
- ブラウザからJenkinsにアクセスできる

帰ったらやってみよう Vagrant(2)

- いるもの
- VirtualBox
- Vagrant

帰ったらやってみよう Vagrant(3)

```
$ mkdir hideo
$ cd hideo
$ curl -o Vagrantfile https://gist.githubusercontent.com/niku/8867832/raw/5b7cafeedd37b775000b3fa87a1076f8980f67dd/Vagrantfile
$ vagrant up
# 処理が終わったら、ブラウザから http://localhost:18080/ にアクセスすると
# Jenkins が動いている
$ vagrant halt
```

Vagrantでノウハウの配布 できた

- けど

- 「誰か」があらかじめ作ってくれたBoxを使っている
- それエンタープライズでも同じこと言えんの？

Vagrantの課題解消

- もっともっと
- Boxも自分で作れるようになりたい
- <http://docs.vagrantup.com/v2/boxes/base.html>

Packer

- <http://www.packer.io/>
- Template(JSON)に設定を書く
 - OSのインストールCD/DVDを取得
 - OSの自動インストール
 - 環境に処理を適用

OSの自動インストール (Debian)

- Debian => preseed.cfg
- [http://www.debian.org/
releases/wheezy/mipsel/
apbs01.html.ja](http://www.debian.org/releases/wheezy/mipsel/apbs01.html.ja)

OSの自動インストール(Red Hat Enterprise Linux)

- RHEL => kickstart
- https://access.redhat.com/site/documentation/ja-JP/Red_Hat_Enterprise_Linux/6/html/Installation_Guide/ch-kickstart2.html

OSの自動インストール (Windows)

- Windows =>
Autounattend.xml
- [http://
technet.microsoft.com/ja-jp/
library/cc749415%
28v=ws.10%29.aspx](http://technet.microsoft.com/ja-jp/library/cc749415%28v=ws.10%29.aspx)

Packerのインストール

- <http://www.packer.io/docs/installation.html>
- ファイルを1つダウンロードして
パス通すだけ

Packerのテンプレート

```
{
  "builders": [
    {
      "type": "virtualbox-iso",
      "iso_checksum_type": "sha512",
      "iso_checksum": "2ed8a7ec7ef3abe59b825f7c8d7176322b87341800277c70dce807c5c9ab1c8f18a9e677766d2a12df334024fe27580cf208a305afaf09e96cfe9e762e4f94df",
      "iso_url": "http://cdimage.debian.org/debian-cd/7.3.0/amd64/iso-cd/debian-7.3.0-amd64-netinst.iso",
      "ssh_username": "vagrant",
      "ssh_password": "vagrant",
      "http_directory": "./",
      "boot_command": [
        "cscs",
        "wait",
        "install",
        "preseed/url=http://{{ .HTTPIP }}:{{ .HTTPPort }}/preseed.cfg ",
        "debian-installer=en_US ",
        "auto ",
        "locale=en_US ",
        "kbd-chooser/method=us ",
        "netcfg/get_hostname={{ .Name }} ",
        "netcfg/get_domain=vagrantup.com ",
        "fb=false ",
        "debconf/frontend=noninteractive ",
        "console-setup/ask_detect=false ",
        "console-keymaps-at/keymap=us ",
        "keyboard-configuration/xb-keymap=us ",
        "<enter>"
      ],
      "shutdown_command": "sudo shutdown -h now",
      "guest_os_type": "Debian_64"
    }
  ],
  "provisioners": [
    {
      "type": "shell",
      "scripts": [
        "base.sh",
        "chef.sh",
        "vagrant.sh"
      ]
    }
  ],
  "post-processors": [
    {
      "type": "vagrant"
    }
  ]
}
```

Packerのテンプレート作るのつらそう

- 「Packerを使ってVagrantのBoxを作る方法を一つずつ説明する」という良いコンテンツがある
- [packer vagrant] で検索

帰ったらやってみよう Packer

- コマンドが終わったらDebianベースでVagrantに取り込めるBoxができています

帰ったらやってみよう Packer(2)

- いるもの
- VirtualBox
- Packer

帰ったらやってみよう

Packer(3)

```
$ git clone git@github.com:niku/my-packer-and-vagrant-example.git  
$ cd my-packer-and-vagrant-example  
$ packer build wheezy64.json
```

Vagrantでやってたこと

- あらかじめ仮想化してある環境 (Box)の取得
- ↑の環境へ事前処理(Provision)を適用

Boxの取得はできた

サーバーへの処理をShellでやるとしんどなってくる

- パッケージ化されていないものやソースから最新版を入れてメンテすること
- 設定ファイルの内容を書くこと
- 何回も動かしても大丈夫なようにすること

Chef(Chef Solo)

- <http://www.getchef.com/>
- サーバーが「こうあるべき」というのを書いておく
- 「こうじゃないところ」を「こう」にしてくれる
- 「こう」になっているところは変わらない

入門Chef Soloを読もう

- 全部書いてある
- <http://tatsu-zine.com/books/chef-solo>

帰ったらやってみようChef (1)

- サーバーが自分のidでログインできるようになっている
- Ruby2.1.0が使えるようになっている

帰ったらやってみようChef (2)

- いるもの
- VirtualBox
- Vagrant
- git
- Ruby実行環境
- `gem(knife-solo)`

帰ったらやってみようChef (3)

```
$ git clone git@github.com:niku/server-settings-with-chef-on-debian.git
$ cd server-settings-with-chef-on-debian
$ bundle install
$ mv data_bags/users/niku.json data_bags/users/{自分のid}.json
$ emacs data_bags/users/{自分のid}.json
# id を自分のidに変更する
# key を自分の公開鍵に変更する
$ vagrant init
$ vagrant up
$ vagrant ssh
# vagrantでsshログインできることを確認する
$ ssh 127.0.0.1 -p 2222
# 自分のidでログインできないことを確認する
$ bundle exec knife solo prepare 127.0.0.1 --ssh-user vagrant --ssh-password vagrant --ssh-port 2222
$ bundle exec knife solo cook 127.0.0.1 nodes/default.json --ssh-user vagrant --ssh-password vagrant --ssh-port 2222
# password "vagrant" を何回か打つ
$ ssh 127.0.0.1 -p 2222
```


まとめ

- VirtualBox 仮想化ツール
- Vagrant 仮想環境をBoxとProvisionから構築するツール
- Packer Boxを作るツール
- Chef Provision

宿題

- Vagrant動かしてみる
- Packer動かしてみる
- Chef動かしてみる

俺たちのひでおはこれからだ

にく先生の次回作にご期待ください