



入門Chef Solo

落ち穂拾い



入門 Chef Solo

INFRASTRUCTURE AS CODE



伊藤 直也

登録情報

フォーマット： Kindle版

ファイルサイズ： 374 KB

並列的なデバイスの利用： 無制限

出版社： 伊藤直也; 1版 (2013/3/11)

販売： Amazon Services International, Inc.

言語 日本語

ASIN: B00BSPH158

おすすめ度： [カスタマーレビューを書きませんか？](#)

Amazon ベストセラー商品ランキング: Kindleストア 有料タイトル - 2位 ([Kindleストア 有料タイトルのベストセラーを見る](#))

1位 - [Kindleストア](#) > [Kindle本](#) > [科学・テクノロジー](#) > [工学](#)

1位 - [Kindleストア](#) > [Kindle本](#) > [コンピュータ・IT](#) > [OS](#)

1位 - [本](#) > [コンピュータ・IT](#) > [OS](#) > [Linux](#)

[画像に対するフィードバックを提供する、またはさらに安い価格について知らせる](#)

本日の話題

- 出版後の反応から
- 落ち穂拾い

facebook と chef

- #ChefConf2013

- facebookの中の人発表

<http://tkak.hatenablog.com/entry/2013/04/29/192603>

- 4 people
- Tens of thousands of heterogeneous systems
- Opscode Private Chef
- “Testing against real system is useful and necessary”

Chefでどこまでやるの？

- Chefで全部やるの？
- デプロイもChefで？
- sshして直接変更はするの？

理屈じゃねえだろ？



- 基本はChefだけでやる
 - インスタンス起動後の構成変更は全てChefで行う。直接変更しない
- アプリケーションのデプロイは例外
 - capistrano etc

- 大事な考え方
 - [NG] Chef = 自動化ツール
 - [OK] Chef = ノードの状態管理

convergence
(収束)

- ノードの状態はChefで記述されていない
なければならない
 - ⇒sshで構成変更すると記述されていない
状態が存在することに
- アプリケーションデプロイは特殊
 - rollback やブルーグリーン デプロイ
 - それ専用のツールに任せる

Chef Solo でどこまで？

- Chef Solo はどのレベルまで?
- 複数台に Chef Solo?
- Chef Server vs Chef Solo

- 大規模 (数百～10,000台)
 - おそらく Chef Server 一択
- 中規模 (～数十台)
 - Chef Solo でも頑張れる
 - Chef Server 入れるか悩みどころ
- 小規模 (1 ～ 十数台)
 - Chef Solo で十分

数台ならこれで

```
$ echo host1 host2 host3 | xargs -n 1 knife solo cook
```

xargsで辛くなったら...

```
$ cap provision
```

Chef Server は node をで検索できる

```
$ knife search node "platform:centos AND languages_ruby_version:  
1.8*"
```

Chef Server では Client が自律更新できる

```
$ sudo chef-client -d -i 900
```

- Chef Serverの導入が面倒?
 - chef-zero っていうのが出たそうです
 - DBや認証がないので運用が楽
 - ※ 未検証

Fabric vs Chef

冷やせよ



- 比較軸がずれていますが...
- Fabric(と sh) / Chef はコンセプトが大きく異なる

- 【二度目】 大事な考え方
 - [NG] Chef = 自動化ツール
 - [OK] Chef = ノードの状態管理

convergence
(収束)

- Chef は状態を管理する
 - レシピと異なる状態を、その状態になるようにする
 - 場合によっては「戻す」
 - 「サーバー構築自動化」はその一面
- Fabric は状態は管理しない
 - 書かれた通りに「前進」するだけ

レシピを Git で管理すれば「ノードの状態」をバージョン管理することができる。管理しているのは単なるコードではなく「状態」。

Search or type a command ? ? Explore Gist Blog Help

naoya / [sandbox.bloghackers.net](#) Pull Request

Code Network Pull Requests 0 Issues 0

branch: master Files Commits Branches 1

[sandbox.bloghackers.net](#) / [site-cookbooks](#) / [sandbox](#) / [recipes](#) / [atig.rb](#)

naoya 24 days ago install atig

1 contributor

file | 24 lines (20 sloc) | 0.399 kb

```
1 %w{sqlit
2 packag
3 acti
4 end
5 end
6
7 gem_pack
8 gem_b
9 action
10 end
11
12 director
13 owner
14 group
```

[sandbox.bloghackers.net](#) / Commit History

Keyboard shortcuts available

Apr 21, 2013

Fix bloghackers.net 404 9c64c7b89e

naoya authored 4 days ago [Browse code](#)

Apr 19, 2013

Update nginx.conf for hosting bloghackers.net 88938ca88a

naoya authored 5 days ago [Browse code](#)

VM を共有するの？
レシピを配るの？

- 開発環境の標準化にChefを使おう
- できあがったVMイメージを共有すべきか、レシピを共有すべきか・・・？

- レシピを共有しよう！
 - 自分で convergence してもらおう
 - コードで状態を外部化したのだから
 - レッツ Social Coding

テストどうするの？



髪型がかっこいい割に
わかりません

- before convergence なテスト
- Integration テスト

- before convergence なテスト
 - knife cookbook test
 - シンタックスチェックとか
 - foodcritic
 - LINTER みたいなもの
 - chefspec
 - ユニット (?) テスト
 - strainer
 - 以上のツールの実行を補助するもの

chefspec (<http://qiita.com/items/48fe830d2ee3b6c87bf5> より)

```
require 'chefspec'

describe 'http_server::default' do
  let (:chef_run) { ChefSpec::ChefRunner.new.converge 'http_server::default' }
  it 'should install nginx' do
    chef_run.should start_service 'nginx'
  end

  it 'should start nginx' do
    chef_run.should start_service 'nginx'
  end
end
```

- Integration テスト
 - minitest-chef-handler
 - serverspec
 - Test Kitchen
 - 複数環境を Vagrant で立ち上げて minitest でテスト走らせたりする

minitest-chef-handler

```
require 'minitest/spec'

describe_recipe 'spec_examples::default' do
  describe "files" do
    it "creates the config file" do
      file("/etc/fstab").must_exist
    end

    it "ensures that the foobar file is removed if present" do
      file("/etc/foobar").wont_exist
    end
  end
end
```

serverspec

/* 他の人の発表見てね */

- 定番は test-kitchen + minitest
 - opscore 公式だし
- serverspec 推し
 - Integrationテストは処理計に依存しない方が良い
 - 将来何かに乗り換える時もテストをそのまま流用できる
 - test-kitchen + serverspec 良さそう。ただし未検証
- 悩ましいところ
 - chefspec + Integration Test は結構冗長・・・
同じような記述が多い

- Chef 盛り上がってる
 - facebook が数万台とか激しい
- 使うならChefのコンセプトに従おう
 - 「ノードの状態」を管理しよう
- serverspec 推し。他にも候補あるよ