



OSC 2016 Tokyo Fall

Ansibleで始める運用自動化

2016/11/5

株式会社 日立ソリューションズ
通信クラウド基盤本部 サービスビジネス第2部
山本 慎悟

Contents

- 1 . Infrastructure as CodeとOSS構成管理ツール
- 2 . Ansible とは
- 3 . Ansible Demonstration
- 4 . Ansible Tower
- 5 . まとめ



1 . Infrastructure as Code と OSS構成管理ツール

仮想化・クラウド化に伴い、サーバの台数が増加
多くのサーバを効率的に管理する仕組みが必要

ビジネスのスピード要求の高まり

設計書作成、手順書作成、手順実行、結果確認といった従来型オペレーションには限界。

人間は必ずミスをする

コンピュータであれば必ず同じ処理を実行する。

構成管理ツールを使いこなすことはITエンジニアとしては必須のスキルとなりつつある。

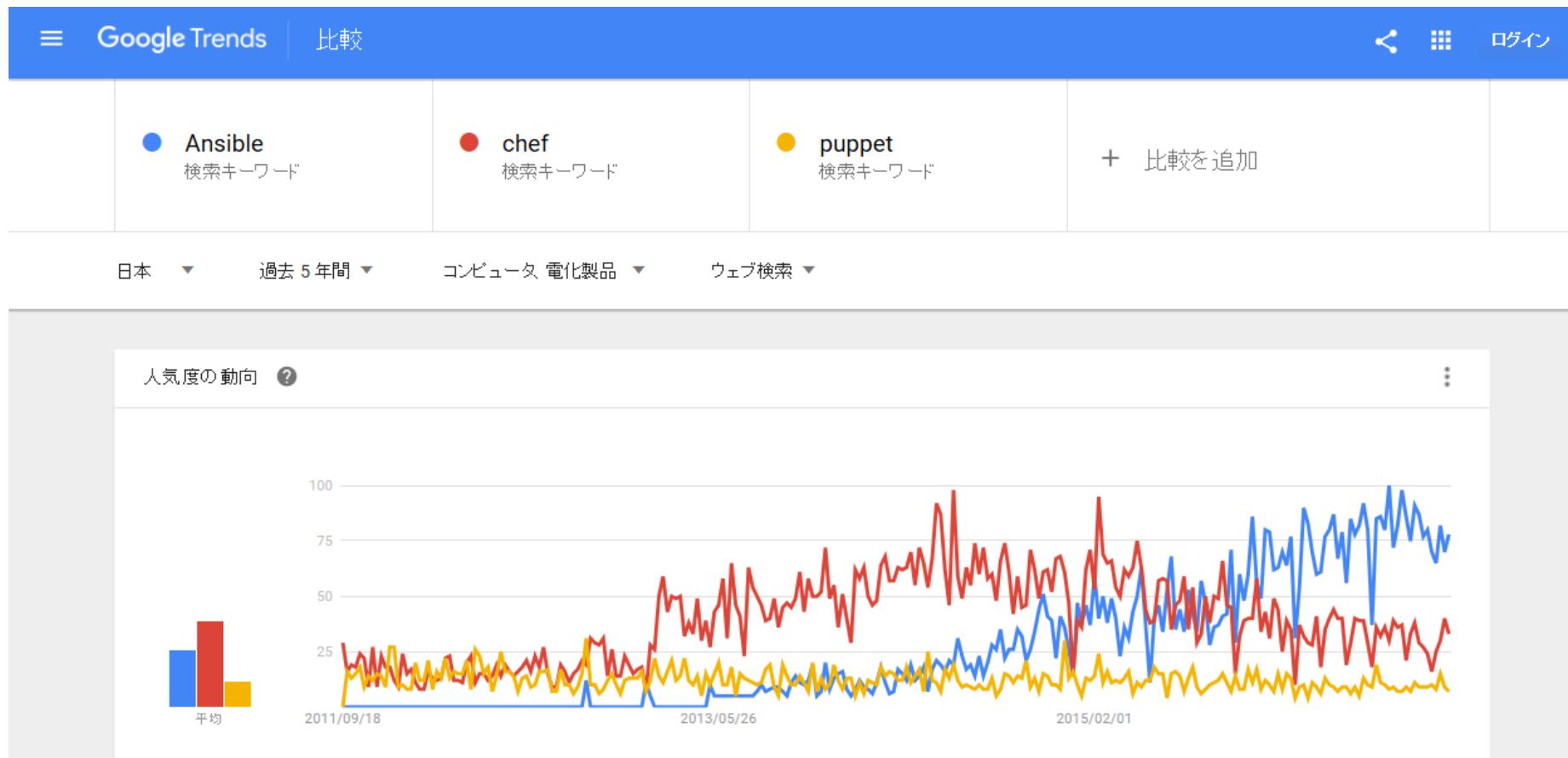
1-2. 主なOSS構成管理ツールと特徴

	Puppet	Chef	Ansible
リリース	2005年	2009年	2012年
言語	Ruby	Ruby	Python
設定言語	DSL	DSL/Ruby	YAML
エージェント要否	要	要	不要

Ansibleは後発のため、既存の構成管理ツールのデメリットを解消。特に、**エージェントレスが非常に強力**であり、既存のシステムにも導入しやすい。

YAML：構造化されたデータを表現するためのフォーマット。読みやすい。書きやすい。わかりやすい。

1-3.OSS構成管理ツールのトレンド



トレンド的にもAnsibleをおさえておけばOK！



2. Ansibleとは

2-1. Ansibleとは

- デプロイメントツール
パッケージのインストール、設定変更、ファイルの配置、サービスの起動停止などを遠隔操作
- オーケストレーションツール
サーバ、ネットワークスイッチ、ロードバランサ、ファイアウォールなどの設定およびデプロイを自動化
- 構成管理ツール
構築作業や日々の定型作業、運用ファイルをテキストファイル化

2015年にRed Hat 社が買収

※ Red Hat 社からサブスクリプションとして提供されるのは、後述のAnsible Towerとなります

- エージェントレス
UNIX系プラットフォームの場合、標準的なSSHサービスが利用できればよい
- YAML形式の設定ファイル
可読性が高く、習得コスト低
- 豊富なモジュールを標準提供
- コマンドラインでの実行が可能
(アドホックコマンド)
- 署等性（べきとうせい）を担保
など実行しても同じ結果となる。

AGENTLESS

SIMPLE

POWERFUL

最小構成では、Inventory File とPlaybookの2つのコードからなる。

1 . Inventory File

管理対象ノードの情報を記述するファイル

2 . Playbook

ノードの構成や設定手順を記述するYAML形式

実行例

```
$ ansible-playbook -i production site.yml
```

単一のコマンドであれば、Playbookを書かずにアドホックコマンドとしてコマンドラインでの実行も可能。(Inventory は必要)

アドホックコマンドの実行例

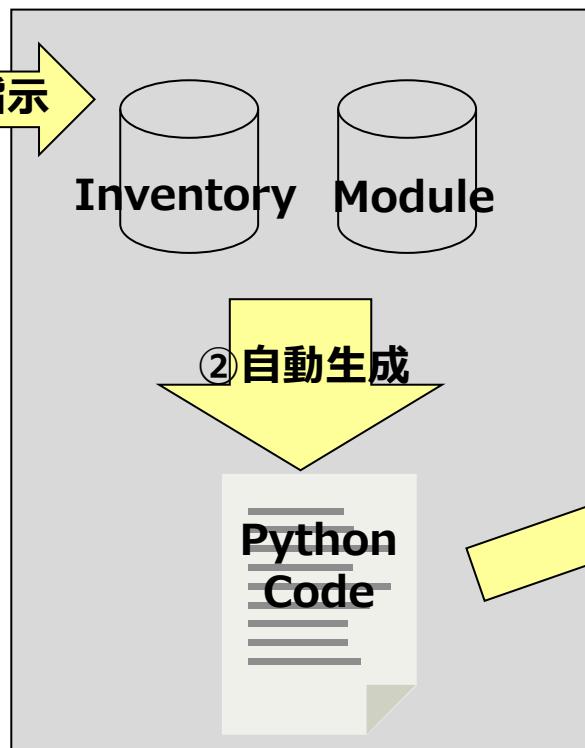
```
$ ansible all -m ping
```

2-4. Ansibleの動作方式

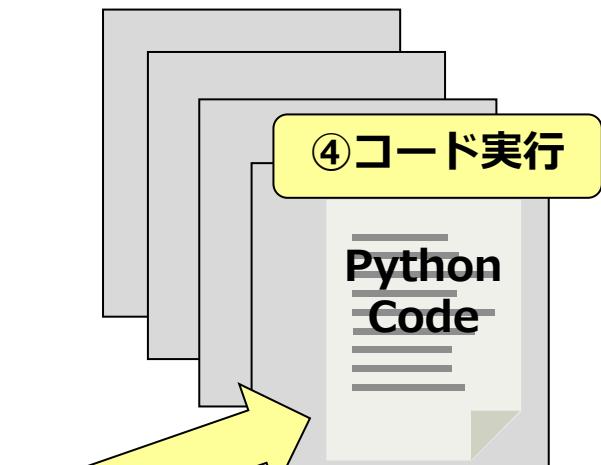
エンジニア



Ansible



管理対象ノード



Playbookを元に、AnsibleがPython Codeを自動生成。
生成したCodeを管理対象ノードに転送し、ssh経由でCodeを実行する。



3 . Ansible Demonstration



4 . Ansible Tower

4-1.Ansible Tower とは

WEBベースで利用できるAnsibleの管理製品。

The image shows two screenshots of the Ansible Tower web interface. The left screenshot displays a dashboard with metrics: 3 Hosts, 0 Failed Hosts, 1 Inventory, 0 Inventory Sync Failures, 2 Projects, and 0 Project Sync Failures. It includes a 'Job Status' chart showing job counts over time and a 'Recent Job Runs' table. The right screenshot shows a specific job run for 'Server Setup' with details like timing, plays, tasks, host events, and a host summary with a donut chart.

Run ID	Status	Playbook Run
44	Successful	2016/02/09 18:08:09
43	Successful	2016/02/09 18:08:139
42	Failed	2016/02/09 18:08:149
41	Successful	2016/02/09 18:08:149
40	Failed	2016/02/09 18:08:149
39	Successful	2016/02/09 18:08:159
38	Successful	2016/02/09 18:08:159
37	Successful	2016/02/09 18:08:159
36	Failed	2016/02/09 18:08:159
35	Successful	2016/02/09 18:08:159
34	Failed	2016/02/09 18:08:159
33	Failed	2016/02/09 18:08:159
32	Successful	2016/02/09 18:08:159
31	Successful	2016/02/09 18:08:159
30	Successful	2016/02/09 18:08:159
29	Successful	2016/02/09 18:08:159
28	Successful	2016/02/09 18:08:159
27	Successful	2016/02/09 18:08:159
26	Successful	2016/02/09 18:08:159
25	Successful	2016/02/09 18:08:159
24	Failed	2016/02/09 18:08:159

- ロールベースのアクセス制御を採用し、システムを安全かつ効率的に管理
- オペレータなどのITスキルの低いユーザでも簡単な操作で構成管理が可能
- Ansibleの実行結果をサーバで集中管理。証跡の監査・追跡を容易に実現。
- スケジューリングとジョブの定期実行をサポート
- RESTful APIを提供
- 現時点ではOSSではない（サポート付有償製品）

特に運用フェーズにおける構成管理・自動化を強力にサポート



5. まとめ

- ・構成管理ツールは使わないともったいない。
むしろＩＴエンジニアとして必須スキルと考えるべき。
- ・構成管理ツールとして Ansibleを推奨
- ・Ansibleの管理ツールとしてAnsible Towerがある。
GUI管理ツールであり、運用フェーズに有効。

AnsibleはOSSのため、気軽に試してみることが可能です。



おまけ

Ansible ノウハウ

Playbookにはベストプラクティスがあります。

Playbookを作成する際は、ベストプラクティスに従って作成しましょう
http://docs.ansible.com/ansible/playbooks_best_practices.html

Git上にもたくさんのPlaybookが掲載されています。

Ansible Tower のセットアップもPlaybookを利用しているので、
いろいろと参考にしてみてください。

おすすめ書籍

Ansible 徹底活用ガイド

<http://book.impress.co.jp/books/1116101065>

2016/10/6 発売

当社の技術者も著者となっています。

END

OSC 2016 Tokyo Fall
Ansibleで始める運用自動化

2016/11/5

株式会社 日立ソリューションズ
通信クラウド基盤本部 サービスビジネス第2部
山本 慎悟

HITACHI
Inspire the Next[®]



付録. Ansible Demonstration

データベースの内容を表示するWEBシステムを作成する

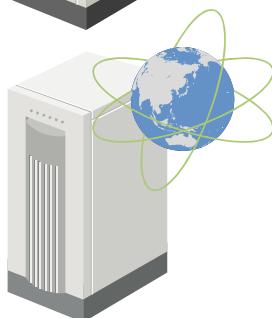
環境



Ansible実行サーバ

OS : RHEL7.2

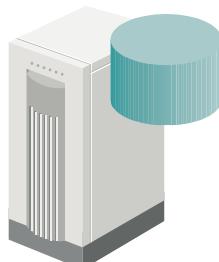
Ansible : ansible-2.1.0.0-1.el7



WEB/APサーバ

OS : RHEL7.2

ミドルウェア : Apache、 PHP



DBサーバ

OS : RHEL7.2

RDBMS : PostgreSQL

一般的な作業手順

WEB/APサーバ

1. パッケージをインストールする (httpd, php, php-pgsql)
2. Apacheのコンフィグファイルを設定する
3. Apacheのコンテンツを配置する
4. SELinuxを設定する
5. サービスを起動する
6. OS起動時のサービス起動を有効にする
7. FirewallでHTTPの通信を許可する

DBサーバ

1. パッケージをインストールする (postgresql-server)
2. initdbを実行する
3. postgresql.confを設定する
4. サービスを起動する
5. データベースを作成する
6. データベースにデータを初期データを投入する
7. OS起動時のサービス起動を有効にする
8. pg_hba.confを設定する
9. FirewallでWEBサーバからのPostgreSQL通信を許可する

※ OSのインストールおよび初期設定については今回は割愛

Ansibleを利用した場合の実行手順

Ansible実行環境

1. Playbookを作成する
2. Playbookを実行する

```
# ansible-playbook -i production site.yml

PLAY [webservers] ****
TASK [setup] ****
ok: [192.168.0.82]

TASK [common : install rpms] ****
changed: [192.168.0.82] => (item={'name': 'httpd'})
changed: [192.168.0.82] => (item={'name': 'php'})

略

PLAY [dbservers] ****
TASK [setup] ****
ok: [192.168.0.81]

TASK [common : install rpms] ****
changed: [192.168.0.81] => (item={'name': 'postgresql-server'})

略

PLAY RECAP ****
192.168.0.81      : ok=14  changed=12  unreachable=0   failed=0
192.168.0.82      : ok=6   changed=5   unreachable=0   failed=0

#
```

もう一度実行してみる

```
# ansible-playbook -i production site.yml

PLAY [webservers] ****
TASK [setup] ****
ok: [192.168.0.82]

TASK [common : install rpms] ****
ok: [192.168.0.82] => (item={'name': 'httpd'})
ok: [192.168.0.82] => (item={'name': 'php'})

略

PLAY [dbservers] ****
TASK [setup] ****
ok: [192.168.0.81]

TASK [common : install rpms] ****
ok: [192.168.0.81] => (item={'name': 'postgresql-server'})

略

PLAY RECAP ****
192.168.0.81      : ok=8    changed=0   unreachable=0   failed=0
192.168.0.82      : ok=6    changed=0   unreachable=0   failed=0

#
```

何度実行しても、最終的な状態は変わらない（べきとう性）

テスト用にもう1つ環境を作つてほしい。予算が足りないからテスト環境は1台構成で。（今回はデモなので、DBサーバをテスト用に流用）

テスト環境作成の流れ

1. Playbookにインベントリと変数定義ファイルを追加する（既存ファイルの修正は不要）
2. Playbookを実行する

```
# ansible-playbook -i staging site.yml

PLAY [webservers] ****
TASK [setup] ****
ok: [192.168.0.81]

TASK [common : install rpms] ****
changed: [192.168.0.81] => (item={'name': 'httpd'})
changed: [192.168.0.81] => (item={'name': 'php'})

略

PLAY [dbservers] ****
TASK [setup] ****
ok: [192.168.0.81]

TASK [common : install rpms] ****
changed: [192.168.0.81] => (item={'name': 'postgresql-server'})

略

PLAY RECAP ****
192.168.0.81      : ok=20  changed=17  unreachable=0  failed=0

#
```