Ansibleではじめる インフラ構築自動化と構成管理

新日鉄住金ソリューションズ(株)
技術本部 システム研究開発センター

小野寺　大地
安久　隼人

自己紹介

- 小野寺　大地
  - 北海道生まれ、北海道育ち
  - ID @onodes
  - Think ITでAnsible入門の連載を執筆中(4月頃公開)
  - Ruby, Python, Linuxを中心にして
  - 自宅OpenStack, Docker,機械学習あたりを家でやってます。

- 安久　隼人
  - 関西生まれ　海育ち
  - ID @hayato1226
  - Think ITでAnsible入門の連載を執筆中(4月頃公開)
  - 自社クラウドサービス開発、マルチクラウドの研究
  - 自宅FCSANなど

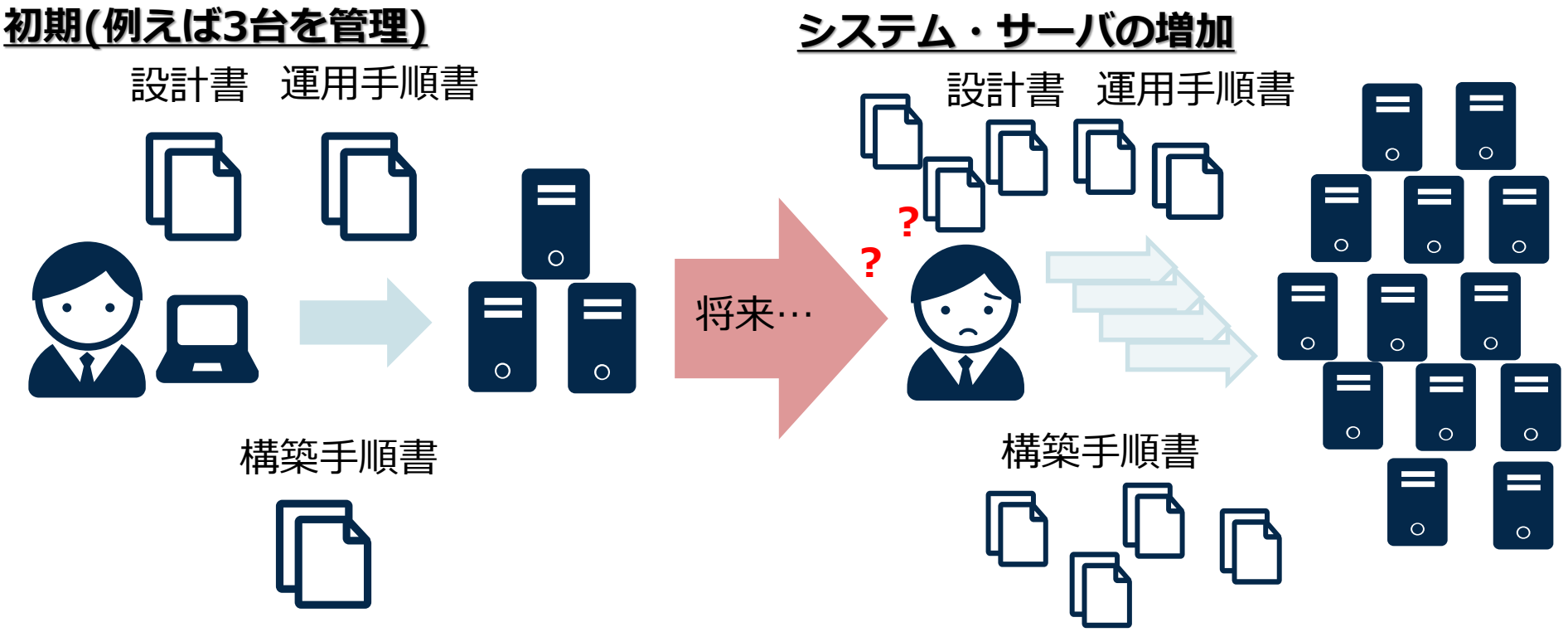
1 はじめに

サーバの構築・運用で

こういうことはありませんか？

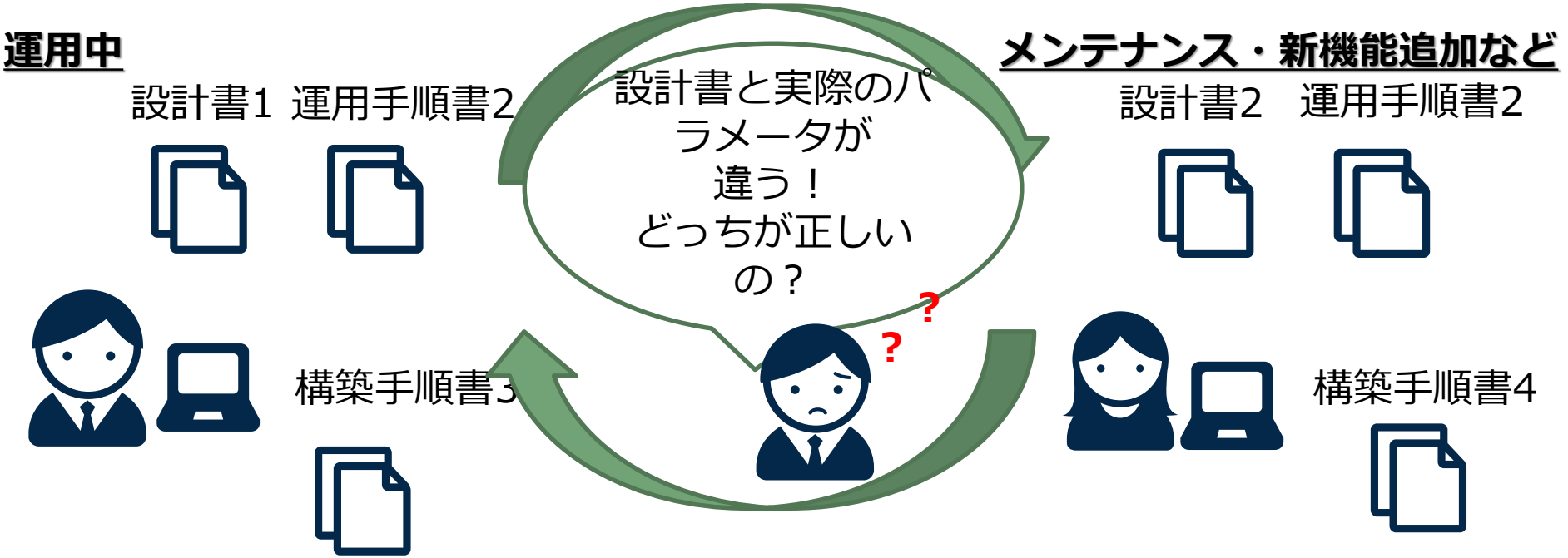
こういうことはありませんか？①

- 初期には少ないサーバ台数だったため、管理可能でした。
- サービスが流行った！システムの数が増えた！を繰り返すと…
- それを支えるサーバ台数の増加で管理が困難になります。
- でも人員増加はないです…
- **一人ひとりが管理するシステム数が膨大になり、作業に時間がかかる。**



こういうことはありませんか？②

- システムに対して変更作業を実施すると、各種手順書も修正が必要となります。
- これを何度も繰り返すと、実際のシステムと設計書、さらには同じ役割のサーバ同士の設定内容が異なってしまうことが発生します。
- どこを変更したのかがわからない！
- **<u>現在のシステム構成</u>を管理し続ける負荷が増大していく。**



ここから生まれる課題

一人ひとりが管理するシステム数が膨大になり、作業に時間がかかる。

現在のシステムの構成の管理が困難になる。

システム開発の速度の低下

＝

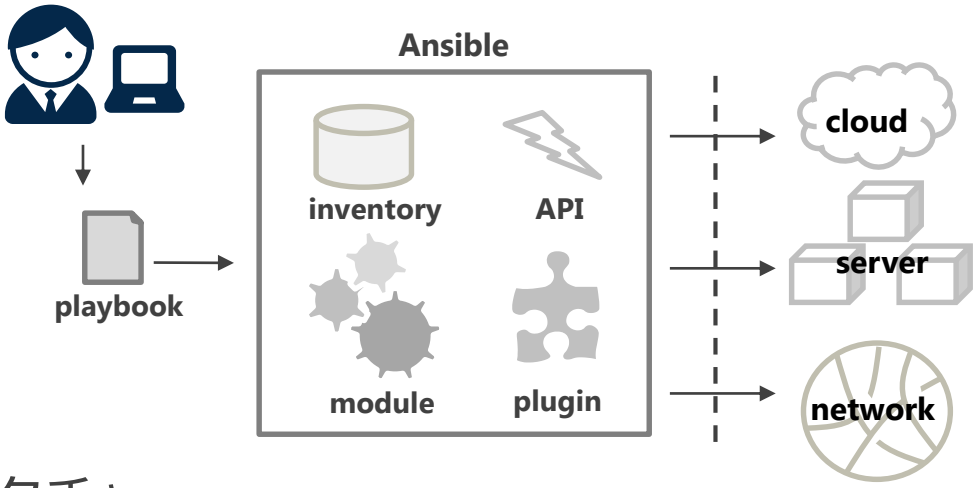
機会の損失！！

こうした問題を解決する手法としてコードによるインフラ管理手法
“Infrastructure as Code (IaC)” に注目が集まっています

2 IaCを具現化するAnsible

Ansibleの役割と特徴

- Ansibleの3つの役割
  - **構成管理ツール**
  - デプロイメントツール
  - オーケストレーションツール
- Ansibleの特徴
  - **エージェントレス**のアーキテクチャ
  - モジュールによる**拡張性**の高さ
  - 後発の製品のため、過去の構成管理ツールの**弱点を克服**！

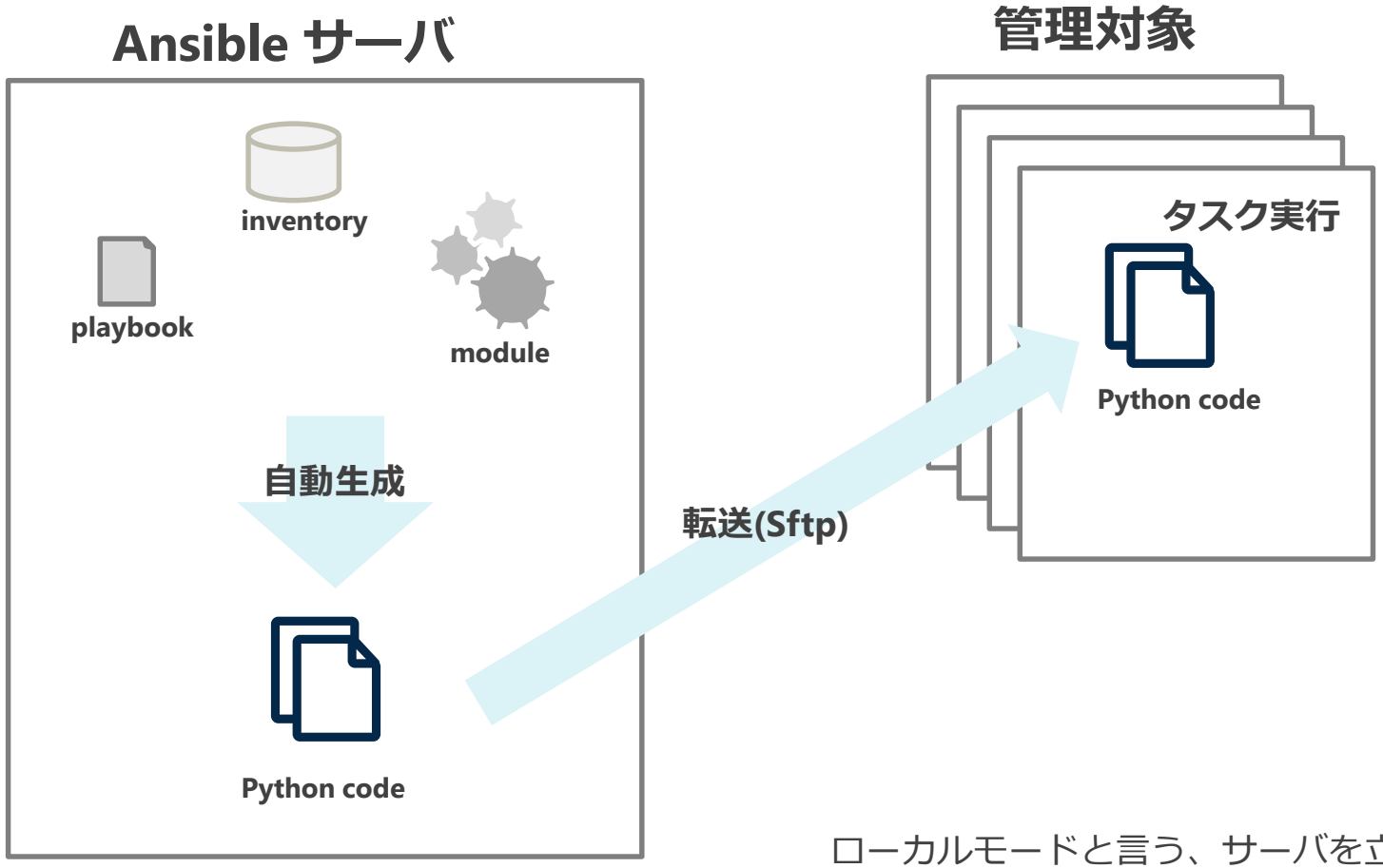


	Puppet	Chef	Salt	Ansible
最初のリリース	2005	2009	2011	2012
設定言語	DSL	Ruby/DSL	YAML	**YAML**
エージェント要否	必要	必要	必要	**不要※**

※ 管理対象にPythonがインストールされている必要があります。

[参考] Ansibleの仕組み

- Playbookを元にAnsibleサーバがpython codeを生成した後、管理対象に対して転送し、ssh経由で実行する仕組み



ローカルモードと言う、サーバを立てず自身をAnsibleで管理するモードもある

Ansibleをオススメする理由

- 自動化ツール・構成管理ツールを使ったことがない人にAnsibleをオススメする理由

1. **入門しやすい**
   - 構築手順や構成情報を構造化されたデータを表現しやすい、汎用のフォーマット(YAML形式)で記述するため、初学者でも扱いやすい

2. **動作環境の準備が容易**
   - 管理対象に特別なツールをインストールする必要がない
     - ⇒エージェントレス
   - 管理対象にはsshでログインできる環境であれば利用可能
     - Pythonが必要だが、主要Linuxディストリビューションにはデフォルトで入っている。

3 Ansibleのサンプル

Ansibleのサンプル

- Ansibleは基本的に2つのコードがあれば動作します。※

1. **Inventory file**
   - 管理対象の役割とホスト情報を記述する部分

2. **Playbook**
   - 管理対象の構成や構築手順を記載する部分
   - YAML形式で記述する

※ アドホックモードは除く。

Ansibleのサンプル(Inventory file)

- Web3層アプリケーションを想定したInventoryfile
  - Webサーバ1台
  - APサーバ　2台
  - DBサーバ　3台
- 下記サンプルはIPで書いていますが、
  名前解決できるホスト名でも問題ありません。

```
[webservers]
192.168.0.40

[apservers]
192.168.0.30
192.168.0.31

[dbservers]
192.168.0.20
```

管理対象の台数が多い時は、

[apservers]
192.168.0.[30:40]

の様にIPアドレス(やホスト名)の連番を省略したり、Inventory fileではなく外部データソースからInventoryを動的に生成するdynamic inventoryと言う機能もあります

Ansibleのサンプル(Playbook)

- yumで色々インストールしたい！
  - Apacheをインストールするサンプル

```
- hosts: webservers
  tasks:
  - name: install httpd httpd-devel gcc
    yum: name="{{ item }}" state=latest
    with_items:
      - httpd
      - httpd-devel
      - gcc
```

  - 各要素の意味

hosts	実行対象のサーバグループを指定
name	手順が何を実施しているかを記述（optional）
yum	yumコマンドを実行するAnsible module
state	インストールするパッケージのバージョンを指定
with_items	イテレータ(YAML形式の配列を指定)
item	with_itemsで定義した配列が展開される

私たちは、このように活用しています！！

- ShellShockを覚えていますか?
  - 我々が管理している**数百台のサーバ**に対して**1時間程度**でパッチ適用を完了させた
    - それまでAnsibleで管理していなかった環境へ導入(導入しやすさ)
    - 検証環境の構築10分
    - Ansible Playbookをコーディング 10分
    - 実行30分(CPUやNW負荷を考えて並列度を落として実行)
    - 確認10分　（これも自動で行いました）
- OpenStackの構築もAnsibleで
  - 社内の研究用クラウド(NSXIC)で利用している**OpenStack**もAnsibleで**自動構築・構成管理**しています
  - Ansibleだけではなく、他のツールとも組み合わせ、物理サーバのセットアップからOpenStackリソースへの組み込みまで自動化しています
  - 自動化されたことで環境構築の速度が上がったことはもちろん、運用手順がシンプルになったことでオペレーションミスの削減にも繋がっています
- Dockerとの組み合わせ
  - Ansible2.0と共に搭載されたAnsible Docker Pluginを利用したDockerの構成管理方法も検討・整理をしています

4 まとめ

まとめ1/2

- Ansibleはサーバ構築を**自動化する構成管理ツール**
- 類似のツールに比べてAnsibleを採用する理由
  1. エージェントレス
  2. モジュールによる拡張性
  3. 後発ならではのかゆい所に手が届く改善がされている
- Ansibleを実行するには？
  1. 実行元ホストにAnsibleがインストールされている
  2. 管理対象にPythonがインストールされている
  3. 実行元から管理対象にsshでログインできること
- Ansibleで必要なソースコードは？
  1. Inventory file
  2. Playbook

まとめ2/2

一人ひとりが管理するサーバ数が膨大になり、作業に時間がかかる。

Ansibleは自動化ツール！

現在のシステムの構成を判断するのに時間がかかる。

Ansibleは構成管理ツール！

- Ansibleのようなツールは**課題解決の糸口**になります
- 中でも**Ansibleは入門しやすい**ツールです
- Ansibleを利用して、開発速度・更改速度を上げて、**より価値があるシステム**を提供しましょう！

商標等の記載について
- NS（ロゴ）、NS Solutions、NSSOLは、新日鉄住金ソリューションズ株式会社の登録商標です。
- NSXICは新日鉄住金ソリューションズ株式会社の登録商標です。
- Linuxは Linus Torvalds氏の、米国およびその他の国における登録商標 あるいは商標です。
- その他本文記載の会社名及び製品名は、それぞれ各社の商標又は登録商標です。