

利用者視点でみる 10分でAnsible

2017.2.23

Hideki Saito

whom



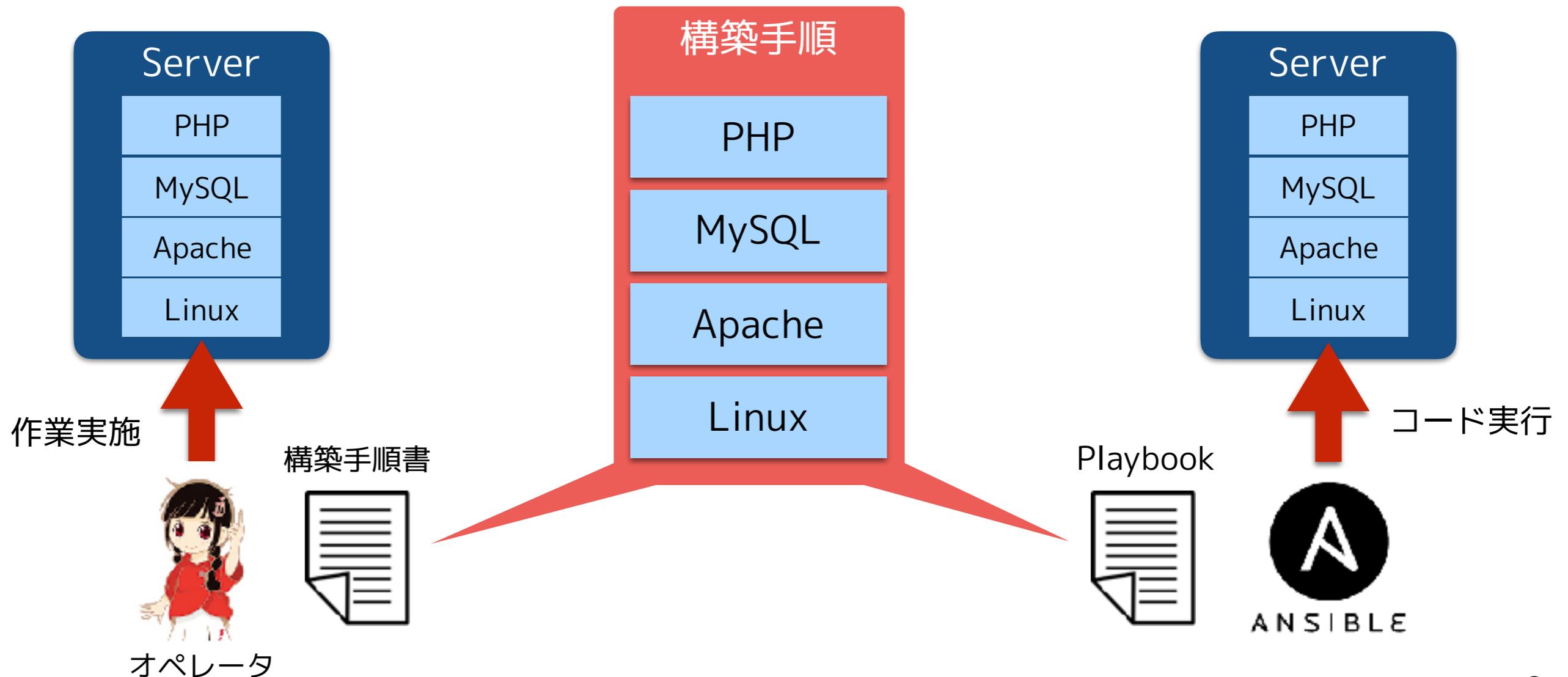
氏名: 齊藤 秀喜 (さいとう ひでき)

TwitterID: @saito_hideki

-
- 所属:
 - 日本OpenStackユーザ会
 - OSSとの関わり
 - OpenStack Active Technical Contributor
 - Ansibleコミュニティモジュールメンテナ
 - 趣味: OpenStack, Ansible, IT Automation

Ansibleとは(1)

Ansibleは、手作業のコード化と自動実行を実現するためのITオートメーションツールです。



Ansibleとは(2)

Ansibleは、ITオートメーションを実現する優秀なツールですが、他の選択肢も存在しています。

他の代表的なプロダクトとして、ChefやPuppetがあげられます。



Ansibleの特徴

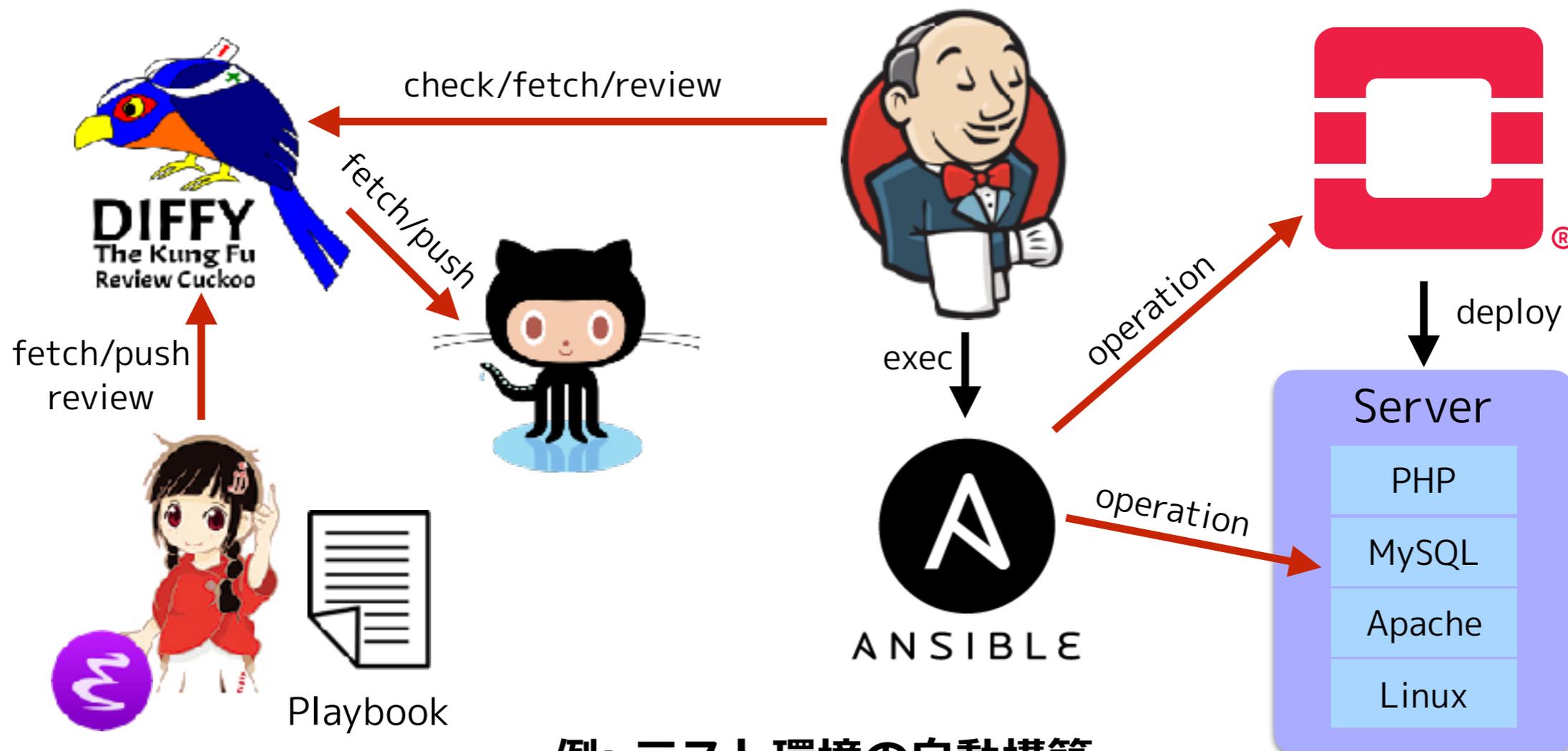
ChefとPuppetはAnsibleに比べて歴史も長く、多くの利用実績があります。また、自動化を実現するためのコンセプトも似通っています。

これらのプロダクトに対して、Ansibleは以下のような特徴を持っています。

- ◎ 運用しやすいコマンドラインベースのシンプルなアーキテクチャ
- ◎ エージェントレスなうえに、CMDBもなしで身軽
- 標準提供される数多くのモジュール群をもっている
- × 外部から制御可能なAPIをもっていない(CLIベースなので)
- × ユーザの細かな権限管理をしにくい
- × Ansibleの設定やログなどを中央で管理する方法は提供されない

Infrastructure as Codeを実現

ITオートメーションツールと、他の役割のツール群を連携させることで、更に大きな導入効果を得られます。



例: テスト環境の自動構築

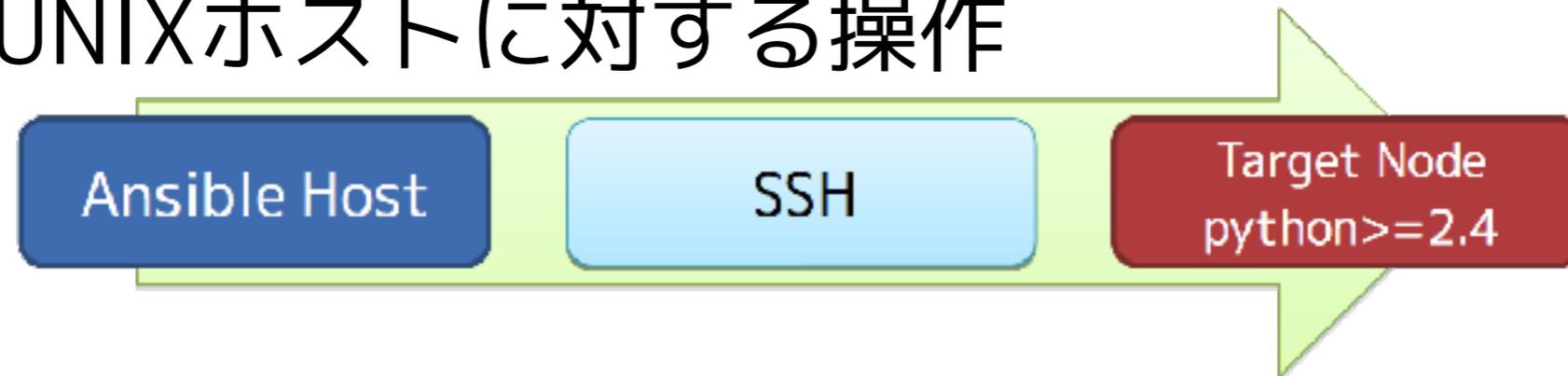
Ansibleの構成要素

Ansibleはコマンドラインベースのツールで、主に以下の要素から構成されています。

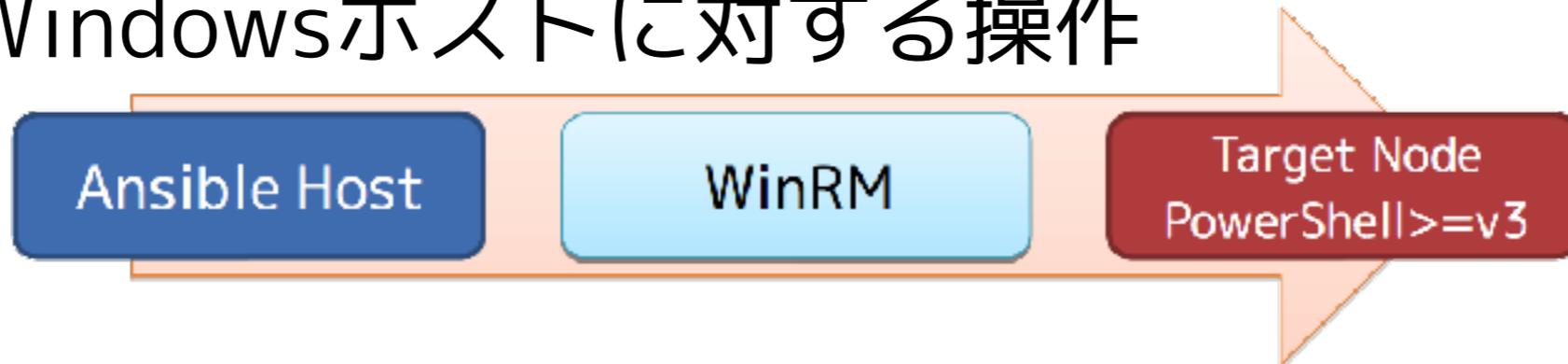
#	構成要素	概要
1	Configuration	Ansibleの振る舞いを決める設定ファイル
2	Inventory	管理対象ホストの一覧が記述されたファイル
3	Module	タスクとしてAnsibleが実行するプログラム
4	Command	タスクやPlaybookを実行するためのコマンド群
5	Playbook	複数のタスクから構成されるワークフロー定義ファイル
6	Plugin	コア機能を拡張するためのプラグイン

Ansibleによる自動化の仕組み

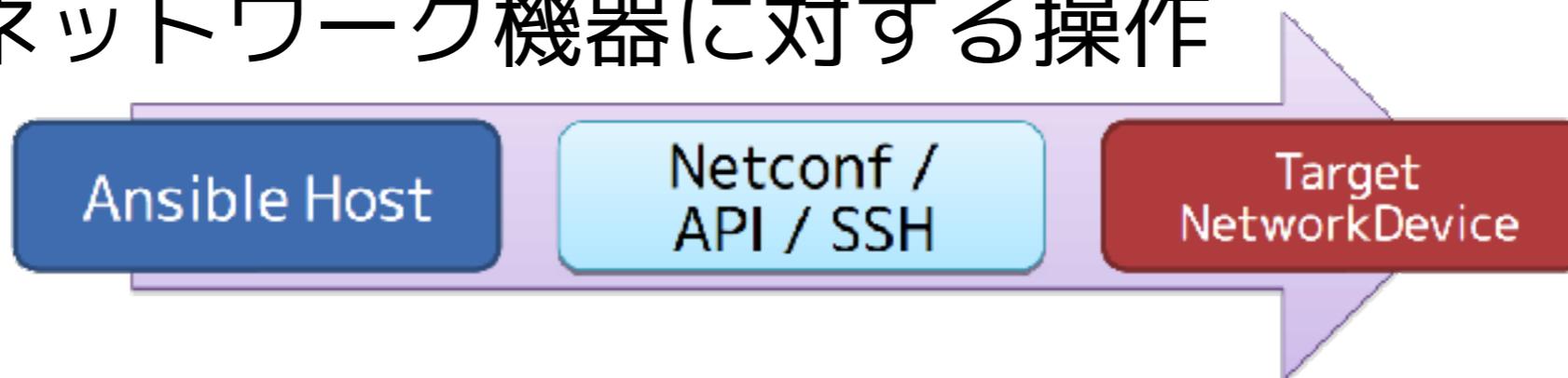
(a) UNIXホストに対する操作



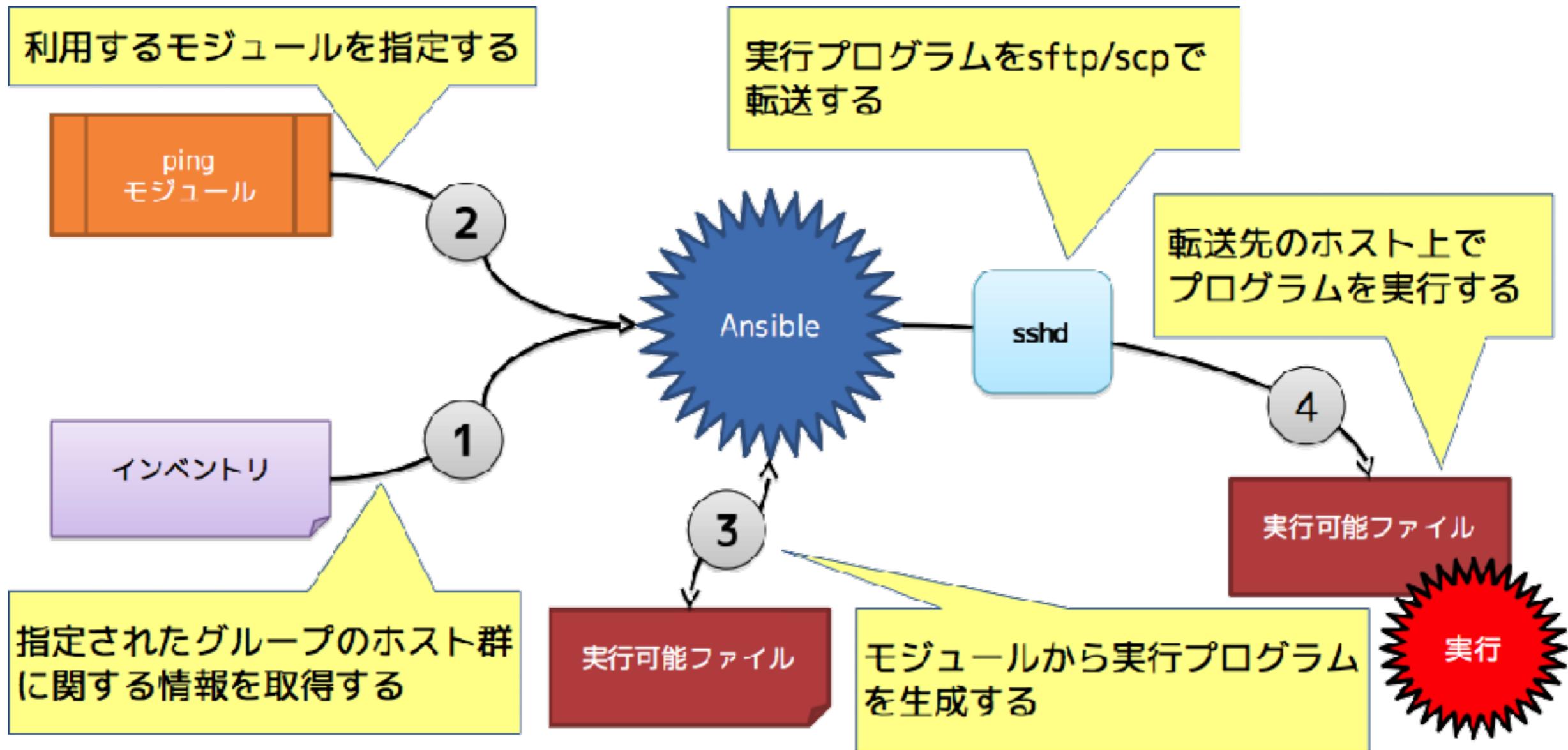
(b) Windowsホストに対する操作



(c) ネットワーク機器に対する操作



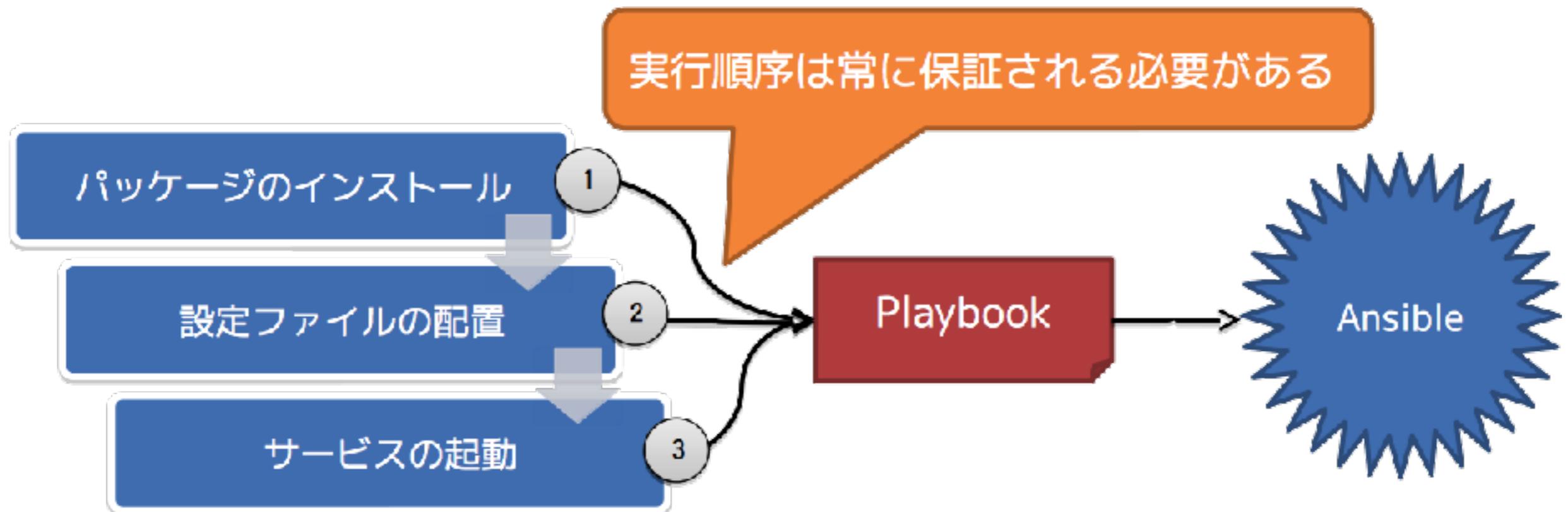
ansibleによるタスク実行



ansible-playbookによるシナリオ実行

AnsibleはPlaybookにより、オペレーションのコード化を実現しています。

Playbookは、実施する一連のタスクを、順序立ててYAML形式で記述したものです。



最後に...

Playbookを書く際の心得

- (1) ちゃんとしたYAMLの書式で書こう
- (2) できるだけBest Practiceに従った構造にしよう
- (3) タスクの先頭にはname:セクションをちゃんと書こう
- (4) タスク内で即値を書かずにvarsファイルに外出ししよう
- (5) 名前重要。Playbookの変数の名前空間はフラットです
- (6) debugモジュールをつかう場合はverbosityも設定しよう
- (7) jinja2テンプレートはできるだけシンプルに書こう
- (8) 巨大な1つのファイルに書かずに分割してincludeしよう
- (9) Playbookをコードレビューできる仕組みを導入しよう

Inventoryを書く際の心得

- (1) IPアドレスやFQDNよりも、わかりやすい名前をつけよう
- (2) 環境毎に分けて誤爆を防ごう
- (3) 適切なパーミッションを設定しよう
- (4) --limitを多用するくらいなら、専用にグループを作ろう
- (5) Dynamic Inventoryは単一の情報ソースから生成しよう

ご静聴ありがとうございました