ANSIBLE ご紹介
～AUTOMATION FOR EVERYONE～

レッドハット株式会社

Agenda

1. Ansible
2. Ansible Tower
3. 活用シーン と 事例
4. Ansible と Red Hat製品

Ansible の ラインナップ

ANSIBLE
by Red Hat®

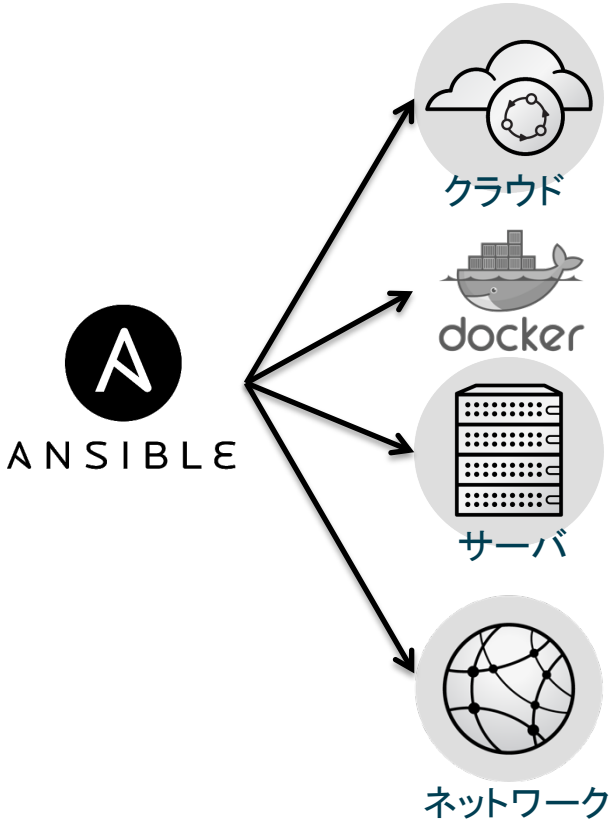
- OSSで提供されている Ansible (Core)
- Red Hatの製品としての提供はされていません

ANSIBLE
TOWER
by Red Hat®

- Red Hat がサブスクリプション(サポート)を提供する製品
- OSSの Ansible (Core) を包含し、多くの便利な機能が追加されています
- 2015年10月に買収、2016年6月より日本でも取扱開始

1. ANSIBLE

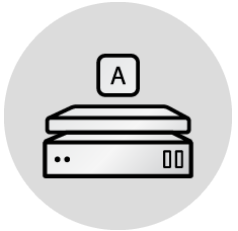
Ansible とは…

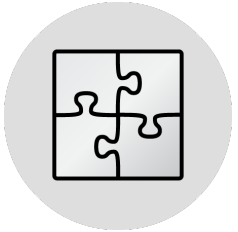


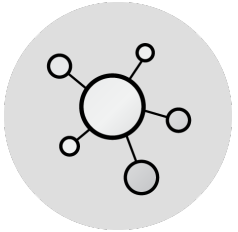
- **いわゆる構成管理ツール**
  あらゆる作業をシンプルに自動化
  - サーバ構成やネットワーク構成
  - 各サーバ内のソフトウェアの構成
  - 外部サービスとの連携等
- **メリット**
  - 手順書管理の手動オペレーションを自動化
  - 安全かつ効率的にシステム構成を管理／維持

Ansible によるオートメーション

- ブートストラップ
  - IaaSの操作APIや各種コマンドを叩いてOS環境やネットワーク設定
- 設定管理
  - OSの設定
    - ユーザ、グループの作成など
  - 各種ミドルウェアのセットアップ
  - 各種サービス、デーモンの起動管理
  - アプリケーションのデプロイメント
    - ソースコード/ビルド成果物の配置
    - 設定ファイルの展開
- オーケストレーション
  - 複数の構成をまとめて一つのシステムとして協調動作させる
    - 負荷状況に応じてマシン数をスケールさせる
    - サービスの新規追加やダウンを検知する







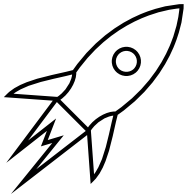
構成管理ツール導入のメリット

1. 安全性が上がる

- 実行時にミスしない
- 作業者に依存しない (属人性の排除)
- ファイルに書くことで変更履歴を管理できる：誰が、いつ、何を？
- 手順書と実環境の乖離が発生しない

2. 作業効率が上がる

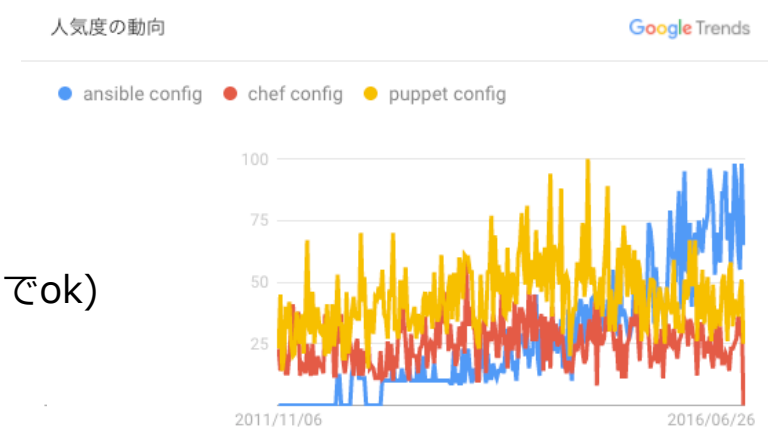
- 何台でも同じ環境を構築できる、並列実行もできる
- 長時間作業や深夜帯の人員配置が不要になる
- リリース作業が素早くなる



3. 他ツールと連携して更なる自動化・効率化が実現できる

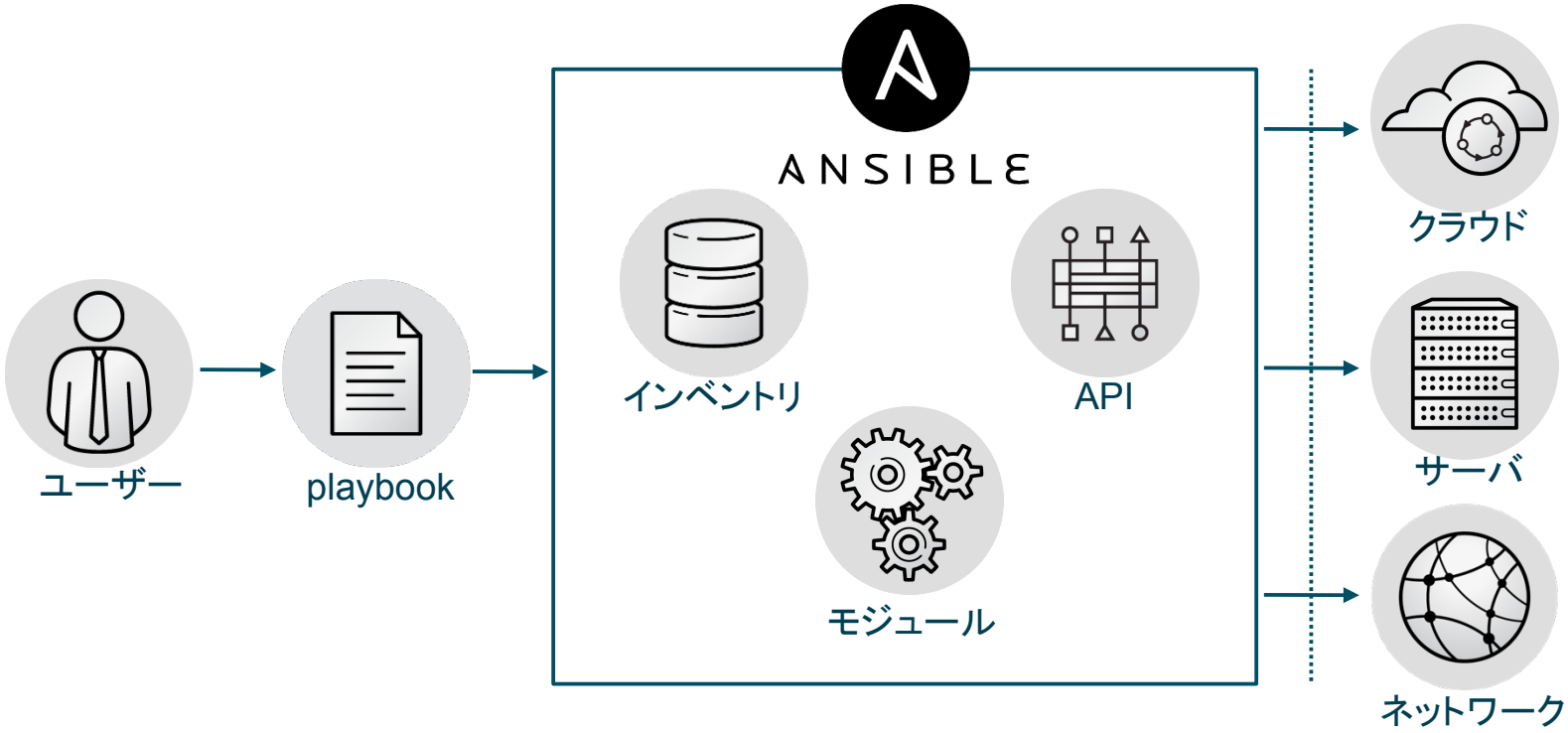
- バージョン管理ツール(git, svn…)による手順/設定の管理
- 自動テストツールによる環境テスト(serverspec等)
- 各種CIツールとの自動連携(jenkins等)
- 監視システムと連携した障害対応自動化(zabbix, nagios等)
- Slack等と連携してチャットベースでの運用作業実行

Ansible の利点

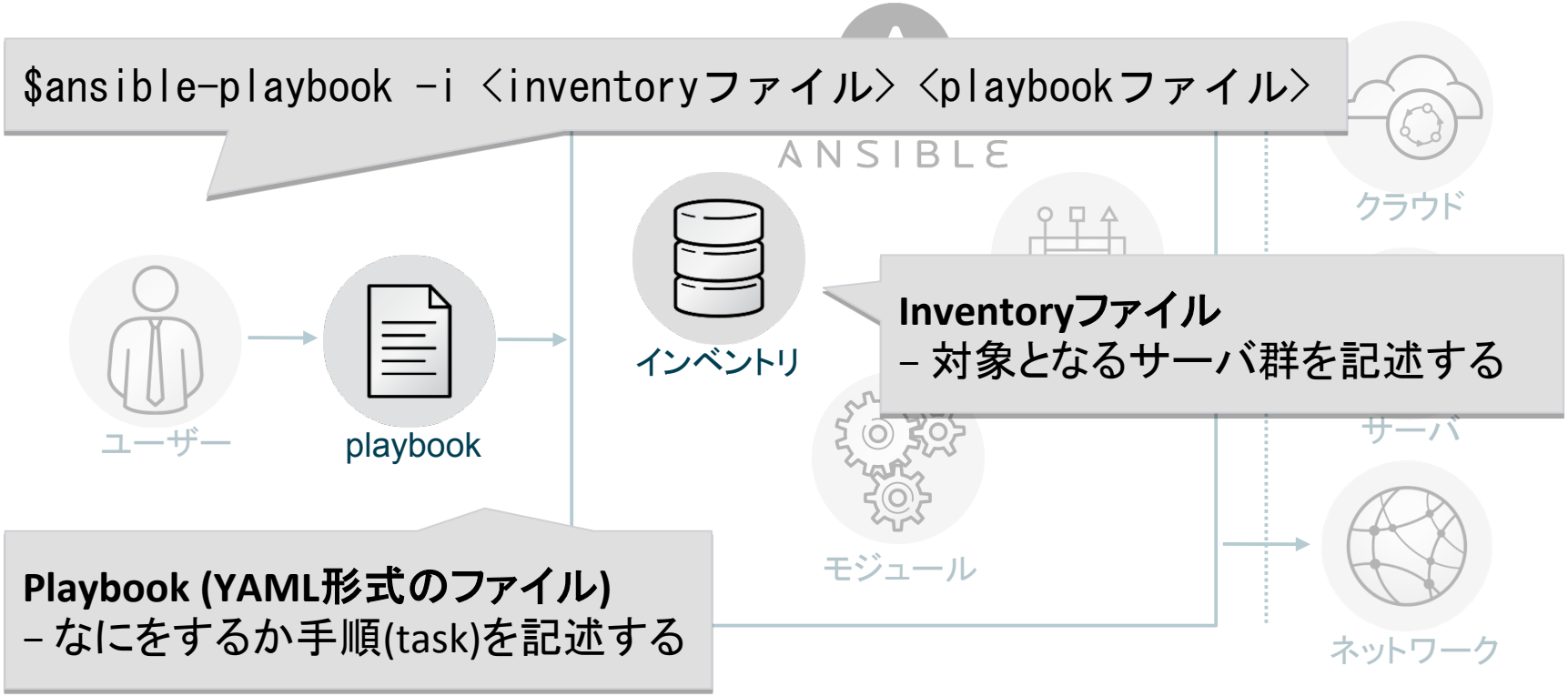


1. **エージェントレス**
   - 対象ホストに何もインストールする必要がない (sshでok)
2. **シンプル**
   - YAML形式で読みやすく書きやすい
     - インデントで構造化され、XMLのようなタグもなく、JSONのように閉じカッコ忘れもない
     - 実行順序が明確：上から順に書いた順番に実行される
   - 非プログラマである管理者や運用担当者も理解しやすい
   - 一定の制約に従った作りとなるため属人化しにくい
3. **パワフル**
   - 多数の製品・機器に対応
     - 800以上のモジュール、更に急増、たった一つの書式で扱える
   - 多数の対象ホストに同時実行
   - ブートストラップから設定変更までをワンストップに実行
     - IaaS上に複数VM立ち上げ、NW設定をし、各VM内の設定変更を行う等
   - 活発なコミュニティ：今後新しいものにどれだけ追随していけるか

Ansible の動作



Ansible の設定ファイル



Inventoryファイル

- 管理対象サーバを記述
  - ホスト名
  - IPアドレス
  - sshのユーザ名
- グループ化できる
- ansible-playbookコマンドの –i オプションで指定する

Playbook の例

ansible-playbookコマンドの実行
$ ansible-playbook –i inventory_file playbook.yml

TARGET セクション

VARS セクション

TASKS セクション

```
---
- name: Apacheのインストールと起動      #Playbookの説明
  hosts: app                            #appグループが対象
  remote_user: root                     #リモートユーザ
  vars:                                 #変数
    http_port: 80
    max_clients: 200

  tasks:                                #実行する手順の内容
  - name: httpdのインストール            #実行時に処理毎に表示される名前
    yum: pkg=httpd state=latest
  - name: Apache configファイルに変数を設定して展開
    template: src=/srv/httpd.j2 dest=/etc/httpd.conf
  - name: httpdを起動
    service: name=httpd state=running
```

タスク

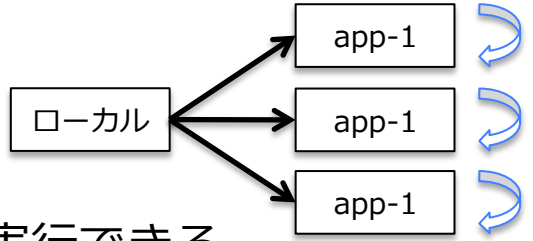
モジュール

実行順序

Playbook のその他の機能

- 繰り返し (with_item, with nested, until …)
- 条件分岐 (when, register, …)
- 他のplaybookの読み込み (include, role, …)
- 外部情報の参照
  - 環境変数、ファイル など (environment, lookup, vars_prompt,…)
- カスタムモジュールを書いて拡張も可能

参考：その他 便利なところ



- **過去資産を活用できる(シェルスクリプト)**
  「いまこの構築スクリプトを使ってるんですよ」
  → “script”モジュールで既存スクリプトを送って実行できる
  → 複数のサーバで実行でき、“creates” で二度実行を防げる

```
- name: 秘伝のスクリプトを実行
  script: files/hiden.sh creates=/tmp/done.txt
```

ファイルやフォルダが既にあるならスキップされる

- **運用時などに使えるアドホックコマンド**

```
$ ansible webservers -m service -a "name=httpd state=stoped"
```

-mでモジュールを指定

サービスが停止していることを確認している

Ansible モジュール

- Module：対象ホストで実行するライブラリ群
- 800以上※のModuleが予め提供、Ansibleコミュニティから日々新しいModuleが公開


Amazon EC2の設定
50種類以上


Linuxの各種設定
80種類以上


CISCOスイッチの設定
15種類以上


OpenStackの設定
40種類以上


Azure の設定
18種類以上


DevOps関連の設定
10種類以上


OSS DBMSの設定
7種類以上


Verticaの設定
5種類以上


VMWareの設定
20種類以上


Big-IPの設定
19種類以上


NetAppの設定
15種類以上


ネットワークの設定
15種類以上

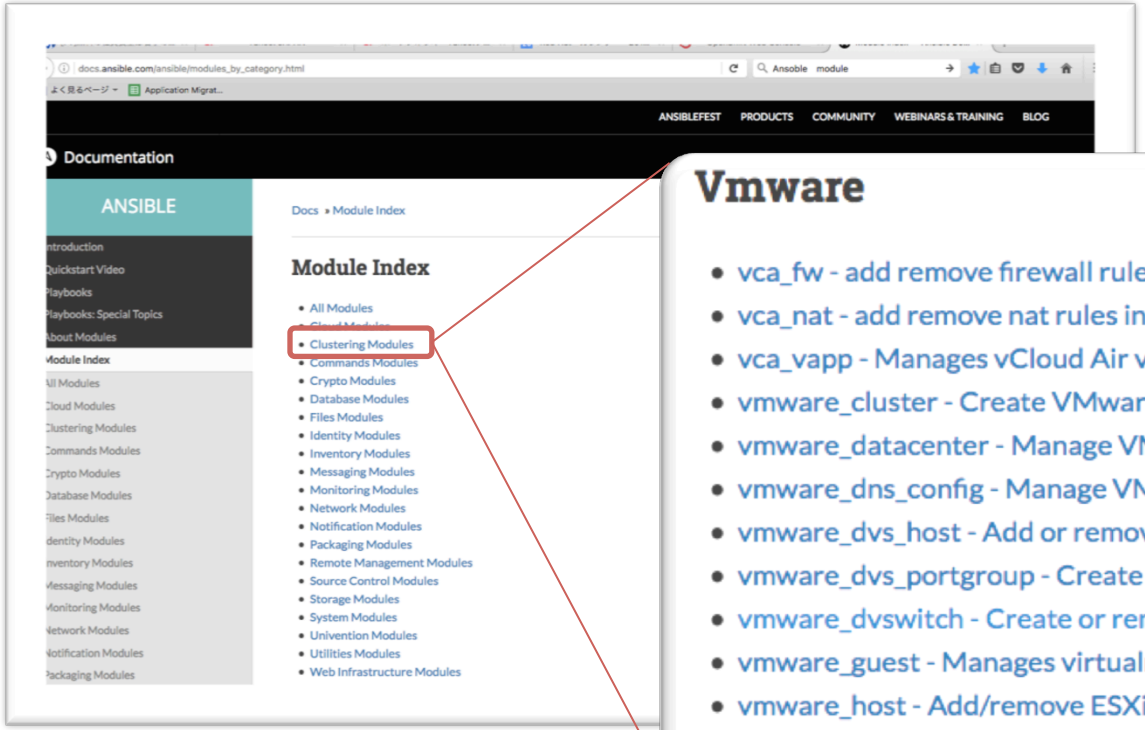

Windowsの設定
40種類以上


Dockerの設定
8種類以上


その他多数

http://docs.ansible.com/ansible/list_of_all_modules.html
※2017年1月時点

モジュール一覧



Vmware

- vca_fw - add remove firewall rules in a gateway in a vca
- vca_nat - add remove nat rules in a gateway in a vca
- vca_vapp - Manages vCloud Air vApp instances.
- vmware_cluster - Create VMware vSphere Cluster
- vmware_datacenter - Manage VMware vSphere Datacenters
- vmware_dns_config - Manage VMware ESXi DNS Configuration
- vmware_dvs_host - Add or remove a host from distributed virtual switch
- vmware_dvs_portgroup - Create or remove a Distributed vSwitch portgroup
- vmware_dvswitch - Create or remove a distributed vSwitch
- vmware_guest - Manages virtualmachines in vcenter
- vmware_host - Add/remove ESXi host to/from vCenter
- vmware_local_user_manager - Manage local users on an ESXi host

代表的な モジュール の例

- パッケージ管理
  - yum, apt
    指定パッケージ(およびその依存パッケージ)のインストール
- サービス制御
  - service
    サービスの起動/停止など
- ファイル処理
  - file, copy, fetch, template
    ファイルの配布(copy, template)、ファイルの収集(fetch)など
- コマンド実行
  - command, shell
    外部コマンドの実行と、その出力結果のとりこみなど

Ansible で管理できる管理対象例

OS

Network製品

Cloud製品等

モジュールサポートについて

- Ansibleは全てののモジュールがコアプロジェクトのコミッターによって維持されているわけではありません。
- 各モジュールには、次のカテゴリに分けられています。

- Core
  - Ansible Coreチームによるメンテナンスされているモジュールであり、常に安全な状態で出荷されます。
- Curated
  - これらのモジュールは現在、Ansibleに同梱されていますが、将来は別途出荷される可能性があります。 主にコミュニティによって管理されていますが、コアコミッターは変更を監視したり、発生した問題を処理します。
- Community
  - これらのモジュールは現在、Ansibleに同梱されていますが、将来は別途出荷される可能性があります。 コミュニティによって維持されています。 問題への対応はコミュニティに依存します。

http://docs.ansible.com/ansible/modules_support.html

Ansible を組織で使う上での課題

様々なPlaybookを共有し、分担してして利用するようになると...

- 複数のユーザが Playbook を編集できてしまう
- Playbookの編集履歴が管理されていない

- 「誰が、いつ、どのシステムを対象に、どんな変更を加えたのか？」追跡したい
  - 実行履歴やその他の操作履歴
- 障害時に原因や影響範囲を特定できない
  - Playbookの履歴と実行履歴の紐付け

ユーザー

playbook

インベントリ

API

サーバ

ユーザー

ネットワーク

- 人や組織によって閲覧できる情報を制限したい (Host情報、ユーザー名、パスワード etc.)
- 実行可能なPlaybookとインベントリを制限したい
- パッケージ導入など、実行にはroot権限が必要

redhat.

2. ANSIBLE TOWER

Ansible Tower の強化ポイント

- 管理者／ユーザーの権限分離
- 履歴管理・監査機能
- 管理用の機能強化・追加

Ansible Tower の強化ポイント

- **管理者／ユーザの権限分離**
  - Job／Project／Inventory単位などUser／Team毎に権限管理ができる(LDAPやADも使える)
- **履歴管理・監査機能**
  - 誰が、いつ、どのシステムを対象に、何をやったか、ダッシュボード表示や変更通知、履歴管理ができる
- **管理用の機能強化・追加**
  - Playbookの実行をWebブラウザから数クリックで実行できる
  - AWSやOpenStackのようなCloudやIaaSとホスト情報を同期しGUIからのInentoryエディタを提供する。オートスケールにも対応
  - Playbookの実行をスケジューリングできる
  - GUIで操作できることは全てRESTful APIでも操作できる

Ansible Tower

アクセス制御
ロールベースのACL、
LDAPとの連携

カタログ管理
Playbookの種類や対象リソースをグラフィカルに管理

ワンクリック実行
ジョブ実行をワンクリックで開始

権限管理
作業実行者の権限管理

監査ログ
Ansibleジョブの実行履歴
をドリルダウンで監視

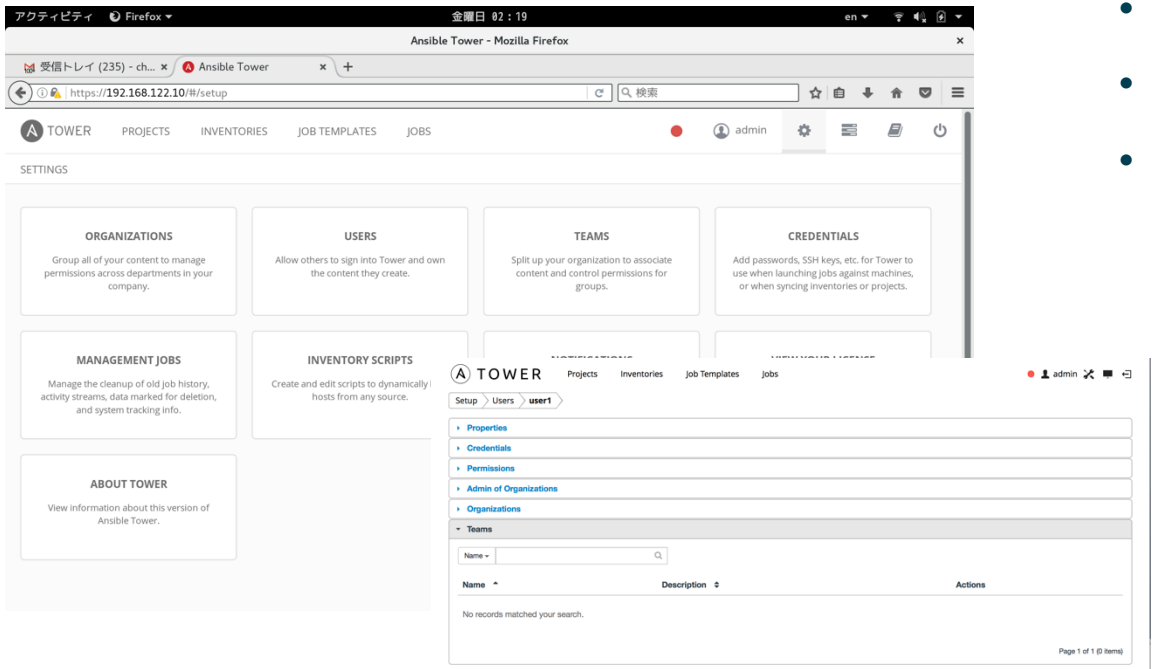
API & CLI
RESTful API を提供しているため外部から
API連携可能。また、Tower コマンドライン
インタプリタを提供しているため、独自の
スクリプトから実行指示が可能

スケジューリング
各種ジョブの
スケジューリング
や自動実行、
状態の一覧

redhat.

Ansible Tower の権限管理

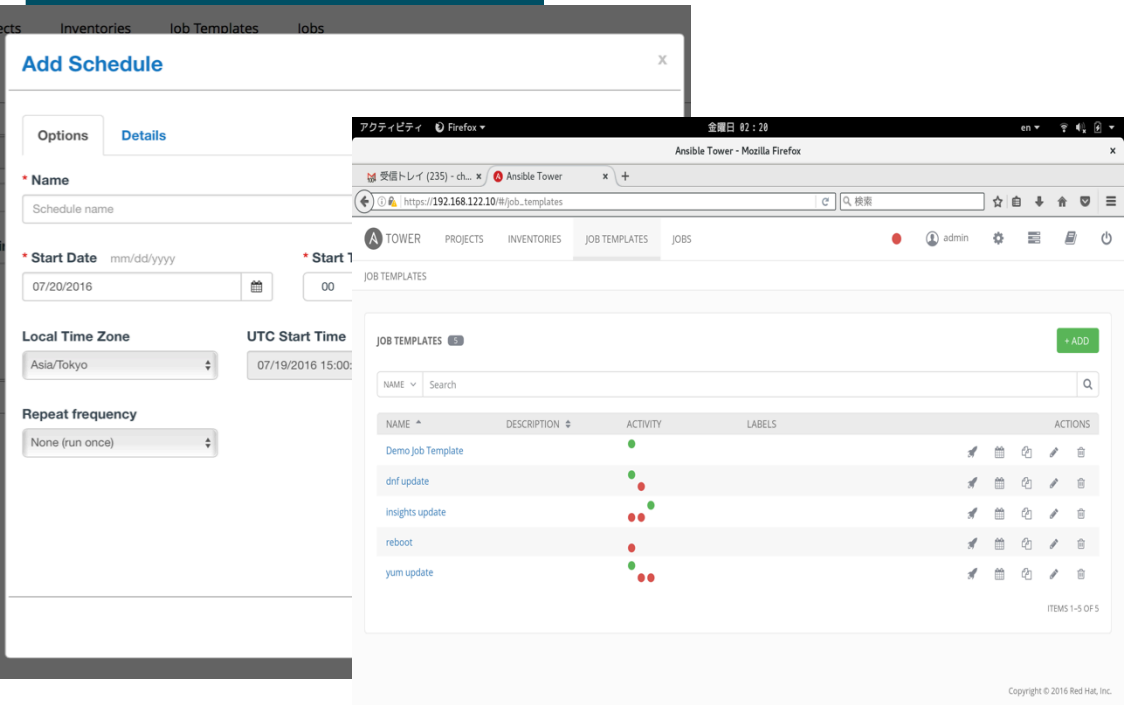
権限管理



- Ansible TowerではWeb UIで操作
- ユーザの認証機能 （ログイン UI ）
- ユーザのTeam へ配置
- ユーザ / Team 毎に権限設定が可能
  - 利用可能な Inventory
  - 利用可能な Credentials
    (パスワード / SSH Public Key)
  - 利用可能な Job Template
    （ Playbook とパラメータのセット ）
  - 指定可能な Project
    (Git などのplaybookレポジトリ)

Ansible Towerのジョブコントロール

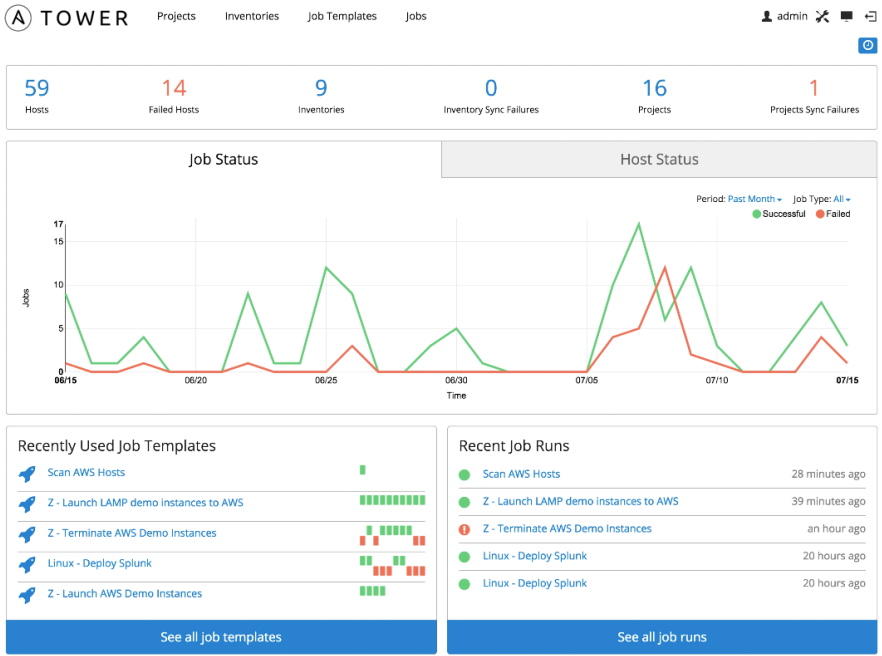
ジョブコントロール



- Job のスケジュール実行
- ジョブの集中管理 / 一括実行
- Ad-hoc Command の実行
- Job Template としての抽象化
  (ユーザが Playbook を直接編集
  することはできません)
- Scan Job と System Tracking

Ansible Tower の可視化機能

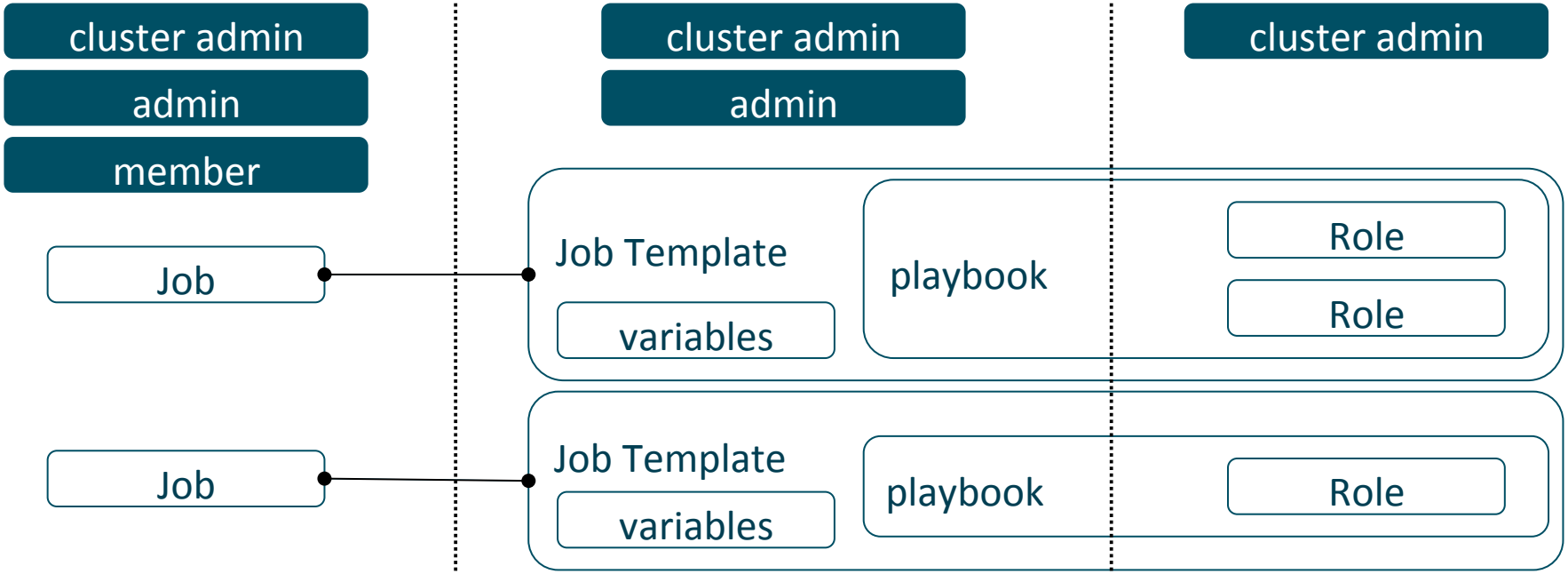
可視化



- ダッシュボード
  - 全体の実行結果
- 各 Job Template の実行結果一覧
  - 各Job の Task ごとの結果
  - ログ（ログレベルの指定可）
- Job Template 毎の実行結果
- Inventory / Host 毎の実行結果

監査機能

社内標準化での利用

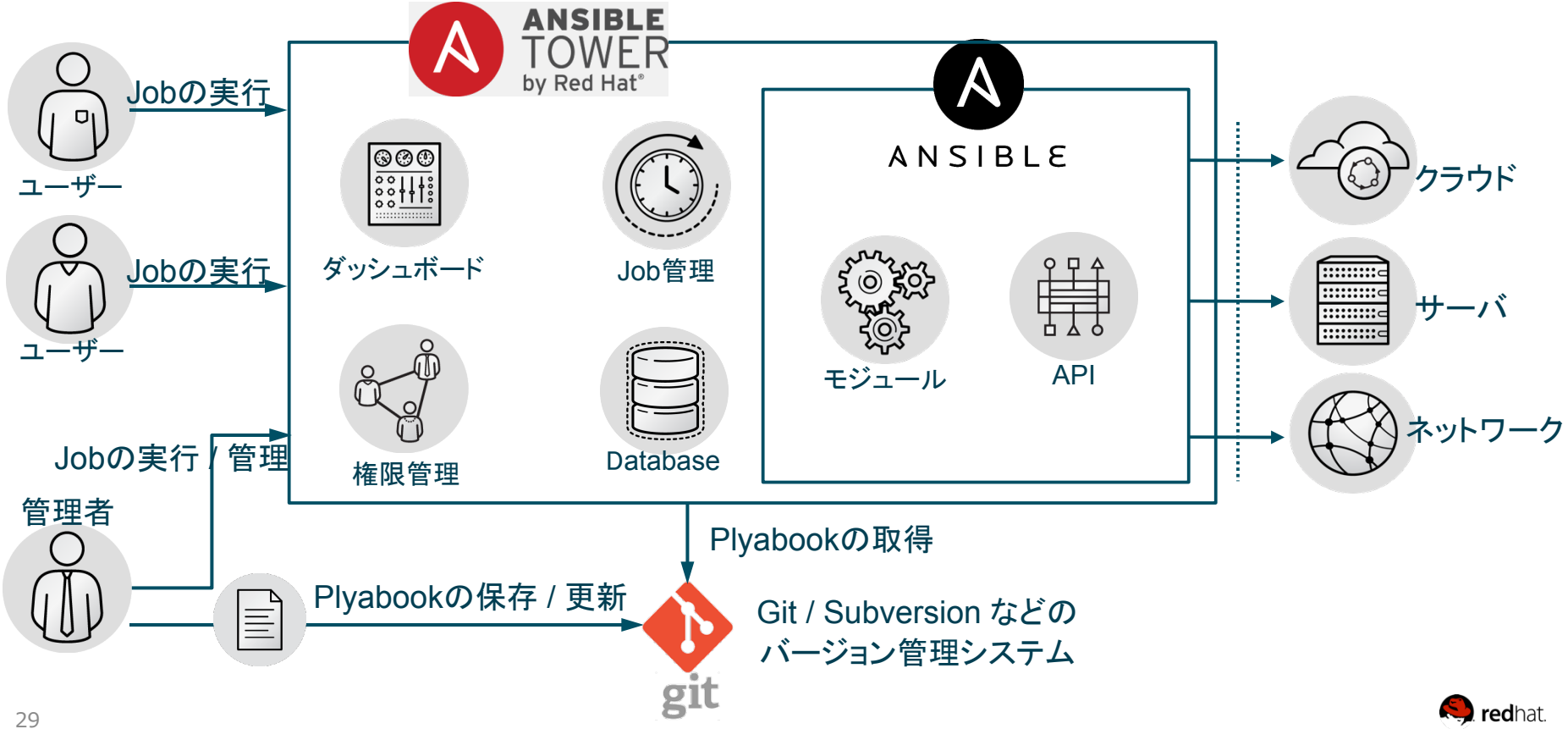


- Jobの実行
- 実行時選択変数の入力

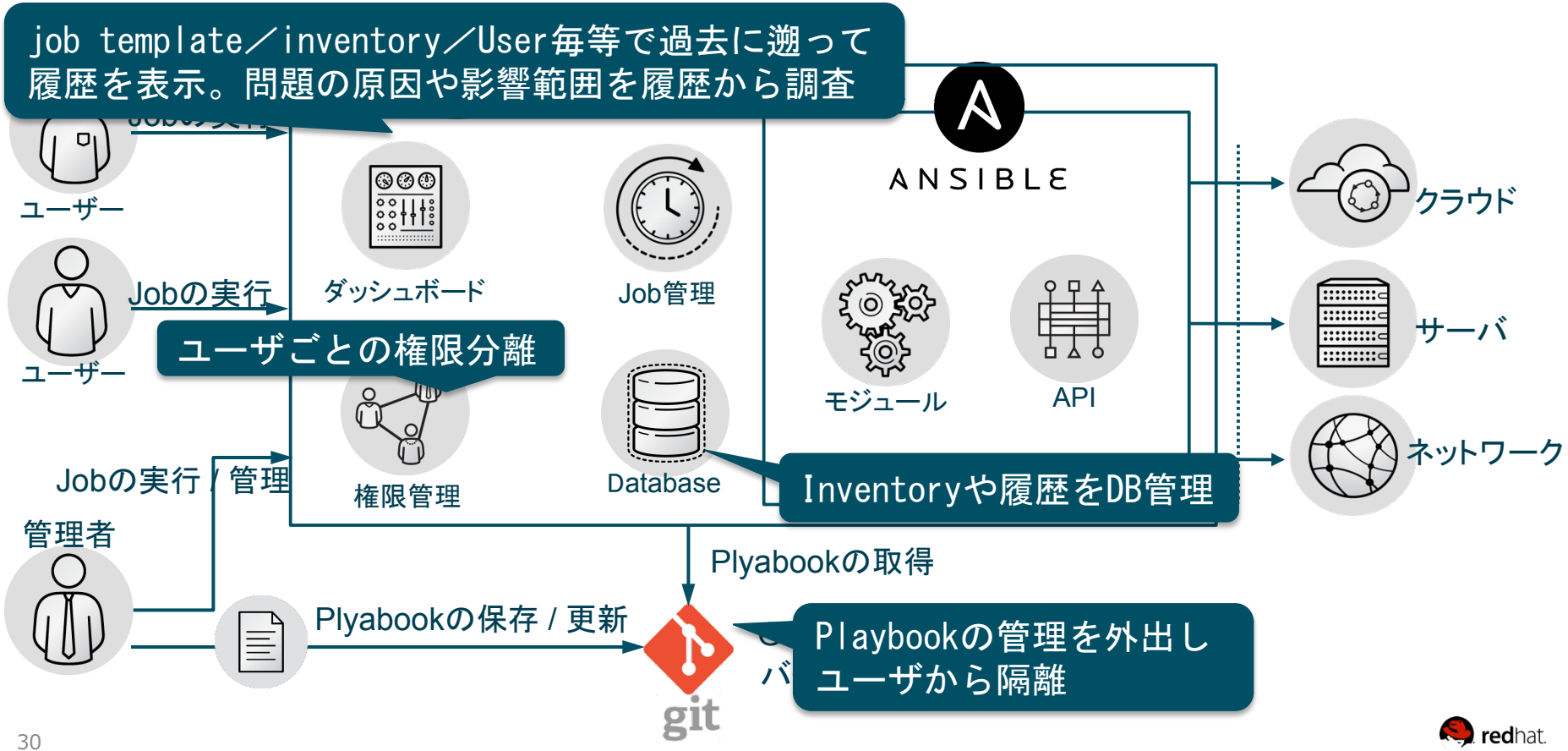
- Playbookの作成・管理
- Job templateの作成・管理
- 変数の設定

- ロールの作成・管理

Ansible Tower の構成



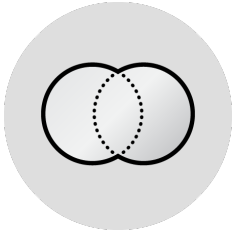
Ansible Tower の構成



Ansible Tower の便利な機能①

Scan Job

- OS に対してのステータス収集を行う特殊 Job
- Package / Service / Status



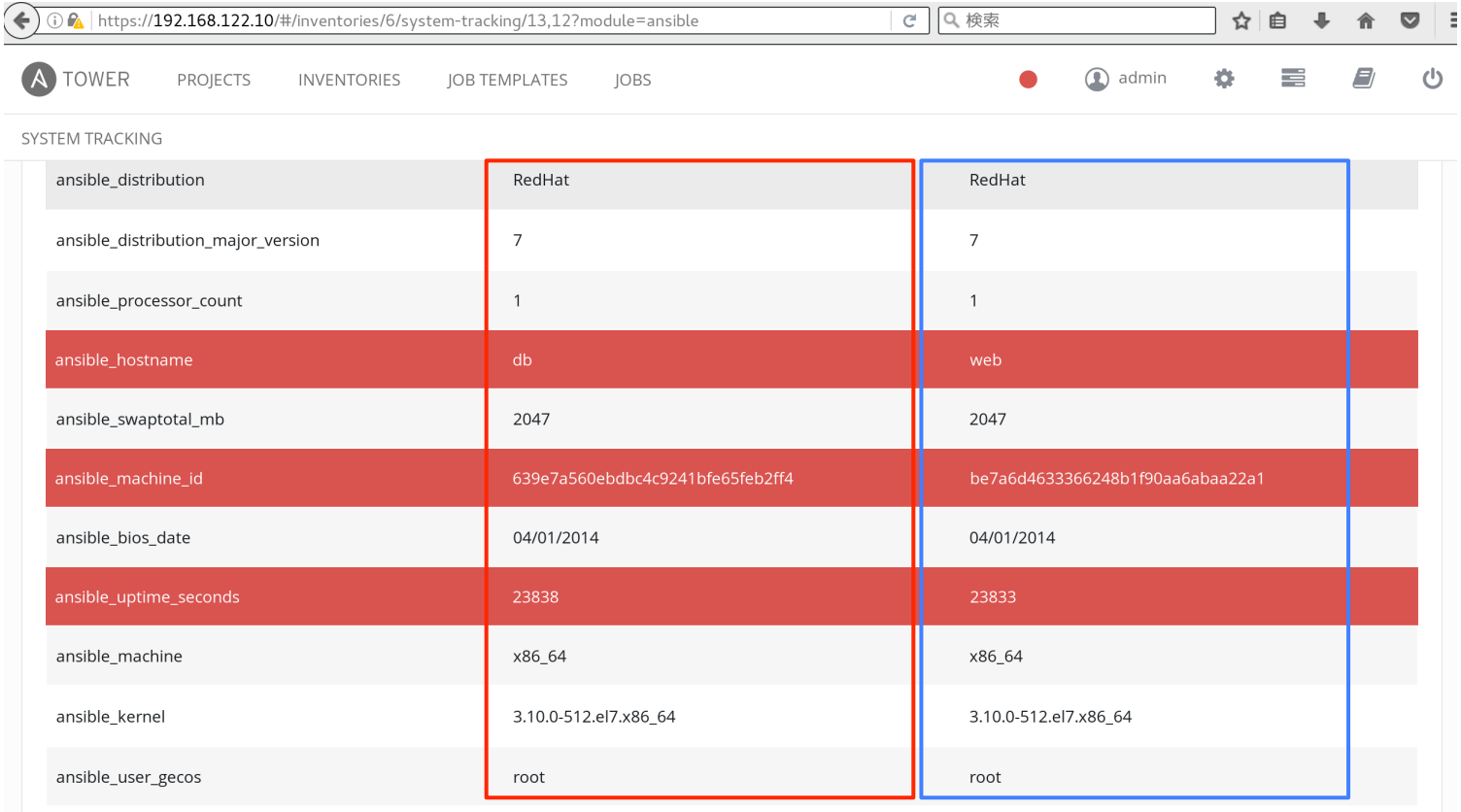
System Tracking

- Scan Job の結果比較機能
- Scan Job が同時実行された複数のHost 間 / 同一の Host の過去結果との比較

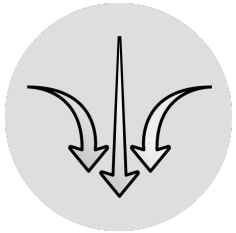
Activity Stream

- 全操作(User追加や権限変更など含め)に対する実行履歴の時系列表示機能

System Tracking の画面例



Ansible Tower の便利な機能②



Restful API

- 外部からRestful API経由でAnsible Towerの機能を実行
- 外部ツール連携やイベントをトリガーとした処理が可能

Notification

- Job 実行 / playbook 更新などの通知機能
- Email / Slack / Twilio (SMS) / PagerDuty / HipChat / Webhook

パスワード管理

- 一度登録すると暗号化され、編集時にも過去の登録内容は一切表示されません。

Ansible Tower のその他の機能③

- Active / Passive 冗長構成
- 画面のカスタマイズ
- SAML / RADIUS 認証
- LDAP 対応
- マルチテナント対応（ 複数 Organization の作成が可能）
- Surveys (ユーザのJob実行時に、特定変数の値を選択させるダイアログ作成機能)
- etc…

3. 活用シーンと事例

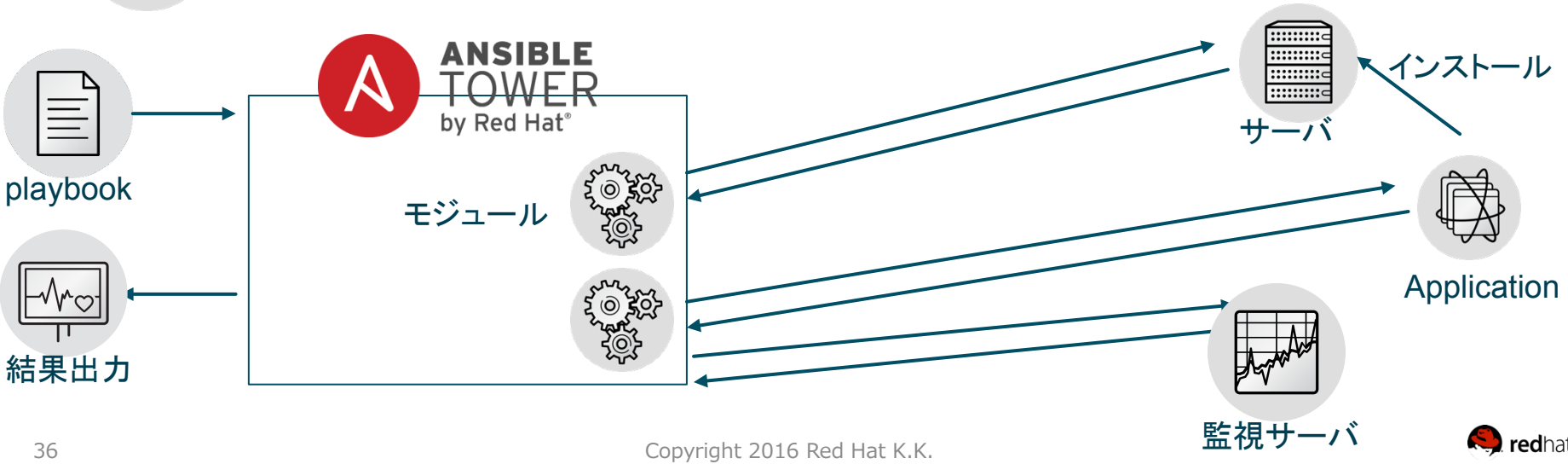
Server 構築と監視設定の自動化

地味に工数がかかる

手動での監視設定によるヒューマンエラー

Server 構築（ Cloud / 仮想化環境でのデプロイ or アプリケーションインストール）を Ansible から実行

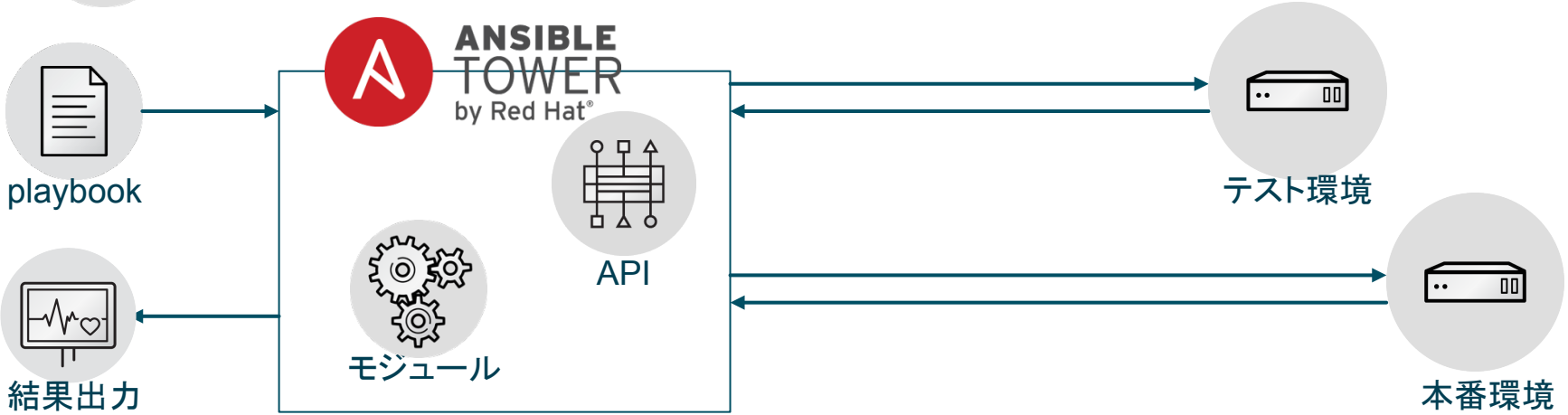
監視ツールへの設定も Ansible から同時に実行し、 監視設定の抜け漏れがなく、工数も短縮



手順書の置き換え

テスト環境に構築・テスト後、手順書を作成してレビューの後、手順書を見ながら本番環境の構築
ヒューマンエラーが発生しやすく、人件費が嵩み、俊敏性に欠ける

Ansible のplaybookを使えば、手順書を作成して繰り返すよりも再現性が高くなる
ヒューマンエラーを除外でき、迅速なデプロイが可能。 エラーが発生しても素早く再実行できる
手順書をplaybookで代替でき、作業結果をAnsible Tower のログ出力で代替できる

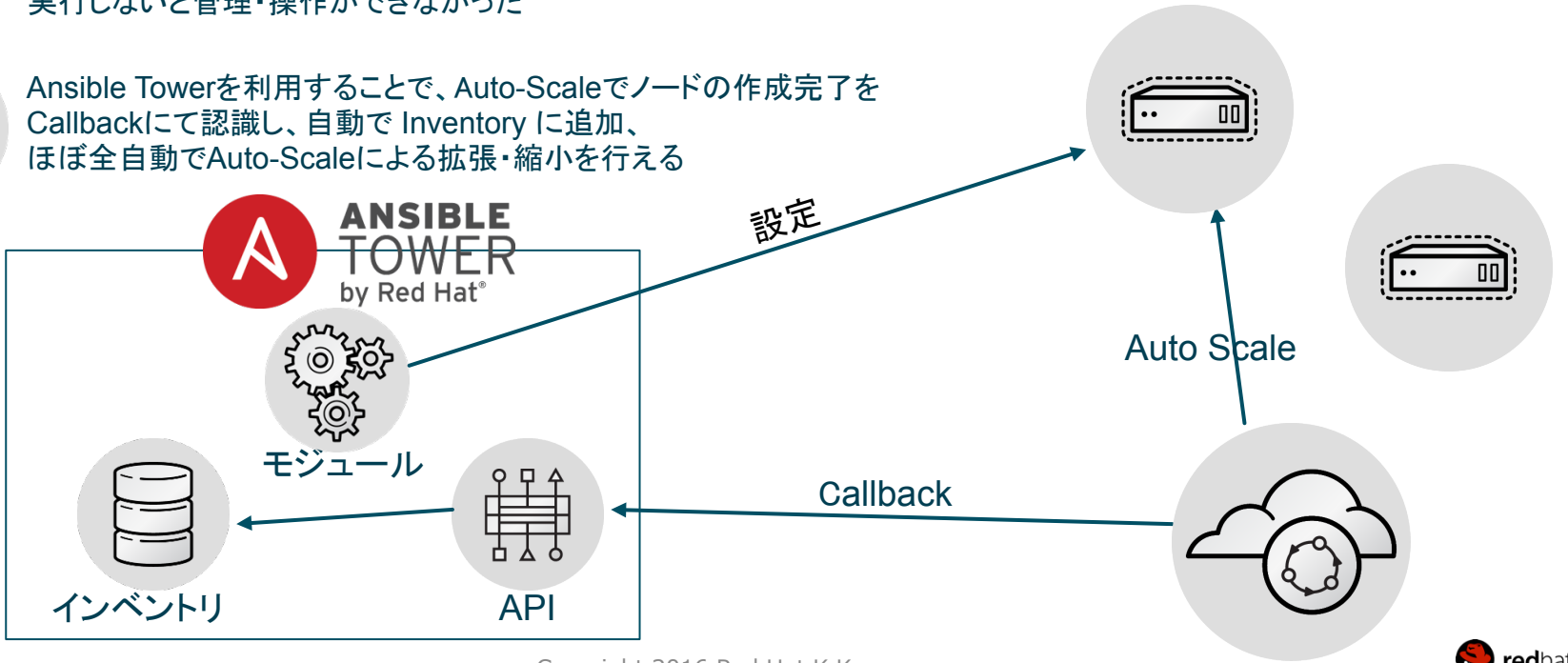


Cloud の Auto-Scale 機能との連携

Auto-Scale機能で追加されたHostに対して限定的な設定のみ

Ansible であっても、Auto-Scaleで追加されたノードのIPをInventoryに追加し、
実行しないと管理・操作ができなかった

Ansible Towerを利用することで、Auto-Scaleでノードの作成完了を
Callbackにて認識し、自動で Inventory に追加、
ほぼ全自動でAuto-Scaleによる拡張・縮小を行える



Ansible Tower の導入実績

世界で、既に600社を超える
企業に導入されています。

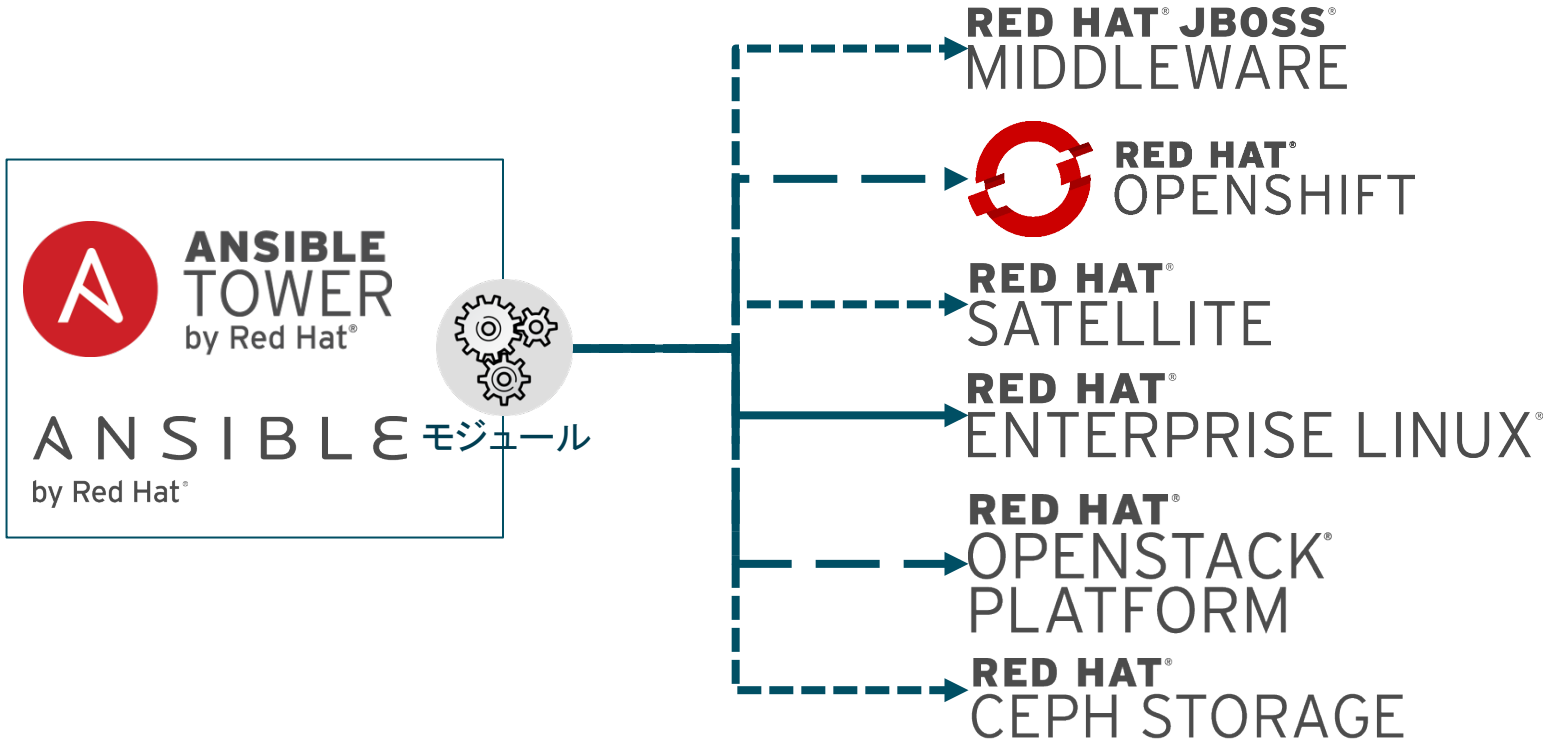
redhat.

BinckBank

- 企業
  - BinckBankは、オランダで最大のオンライン投資銀行で、76万以上の口座を保有しています。
- 課題
  - データセンターの複雑さが課題でした。
  - 自動化とともに製品を使用するにあたってトレーニングが不要なシンプルさが必要でした。
  - 独自のスクリプトを駆使しており、作成やデバッグにかなりの時間を費やしていました。
- 解決策
  - Linux / Unix Serverに対してAnsible Towerを使用しました。
- 効果
  - Ansible Towerで過去履歴が残ることで、正しく環境が構築できていることや問題が起きても影響範囲が示せるようになりました。これまで問題が発生するとインストールは正しく行われたか、他の設定は間違っていないか等の疑念から長時間の確認作業を強いられていました。Tower導入後は、原因となったPlaybook、設定、影響するサーバ等を追えるようになりました。
  - 500台以上のサーバに対して、事前の設定なく(エージェントのインストールが不要で)すぐに設定を行えるようになりました。
  - Ansible Towerを導入したことで社内の非技術者もAnsibleを利用できるようになりました。

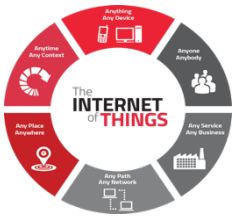
4. ANSIBLE と RED HAT製品

Ansible から操作が可能なRed Hat 製品



参考： https://www.ansible.com/red-hat

まとめ



ARISTA CISCO f5 JUNIPER NETWORKS

何の作業を自動化したいか？
アイデアはあなた次第

IBM AIX solaris

THINK BIG

docker kubernetes

自動化／効率化できる対象は様々

Open Stack

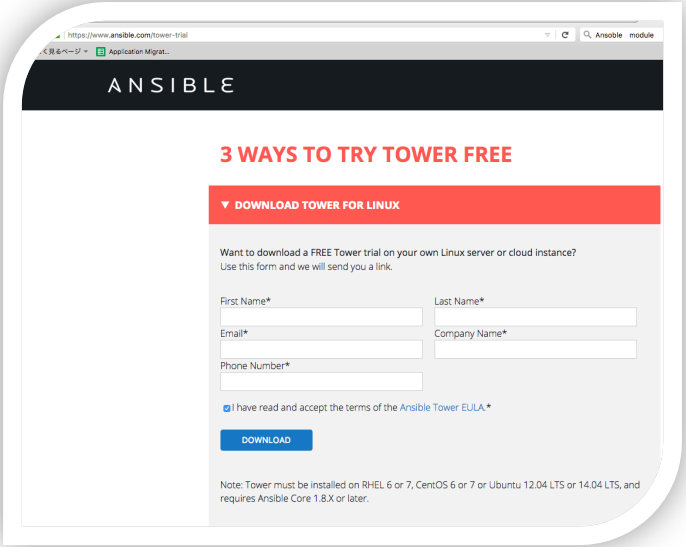
Satellite CloudForms Insight

amazon web services
Microsoft Azure
Google Cloud Platform

redhat.

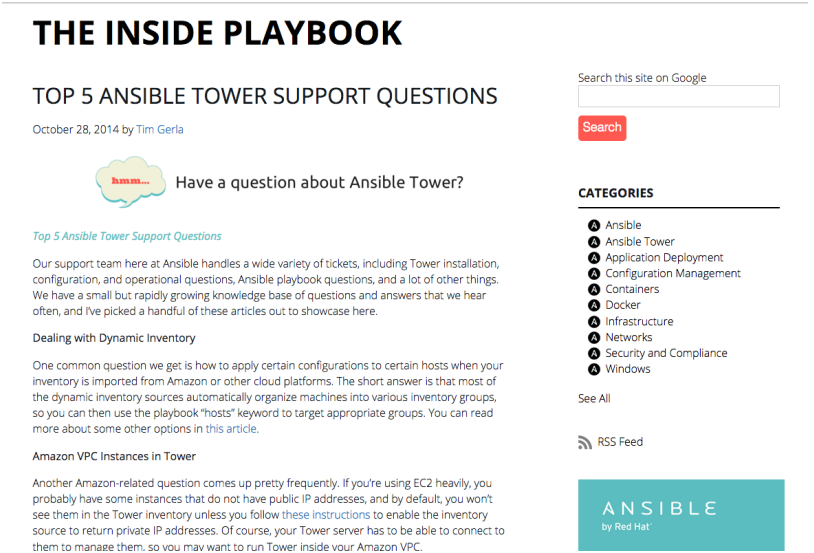
評価版について

- 評価版登録サイトにて、必要な情報を登録後の評価版DLが可能となります。
  - https://www.ansible.com/tower-trial



Ansible Towerのよくある質問

- https://www.ansible.com/blog/ansible-tower-support
  - ユーザのLDAPとの連携は？
  - 既存のインベントリファイルをTowerへ移行は？
  - ・・・



THE INSIDE PLAYBOOK

TOP 5 ANSIBLE TOWER SUPPORT QUESTIONS

October 28, 2014 by Tim Gerla

Have a question about Ansible Tower?

Top 5 Ansible Tower Support Questions

Our support team here at Ansible handles a wide variety of tickets, including Tower installation, configuration, and operational questions, Ansible playbook questions, and a lot of other things. We have a small but rapidly growing knowledge base of questions and answers that we hear often, and I've picked a handful of these articles out to showcase here.

Dealing with Dynamic Inventory

One common question we get is how to apply certain configurations to certain hosts when your inventory is imported from Amazon or other cloud platforms. The short answer is that most of the dynamic inventory sources automatically organize machines into various inventory groups, so you can then use the playbook "hosts" keyword to target appropriate groups. You can read more about some other options in this article.

Amazon VPC Instances in Tower

Another Amazon-related question comes up pretty frequently. If you're using EC2 heavily, you probably have some instances that do not have public IP addresses, and by default, you won't see them in the Tower inventory unless you follow these instructions to enable the inventory source to return private IP addresses. Of course, your Tower server has to be able to connect to them to manage them, so you may want to run Tower inside your Amazon VPC.

Search this site on Google

Search

CATEGORIES

- Ansible
- Ansible Tower
- Application Deployment
- Configuration Management
- Containers
- Docker
- Infrastructure
- Networks
- Security and Compliance
- Windows

See All

RSS Feed

ANSIBLE by Red Hat

教育について

- Automation with Ansible コース (DO407:4 日間)
  - このコースでは、受講者はハンズオンラボを通じて、**Ansible** による管理対象ホスト上のシステム管理タスクの自動化、**Ansible Playbook** の作成とタスク実行の標準化、**Playbook** の集中管理、そして**Ansible Tower** を使用して **Web** インターフェイスの反復実行をスケジューリングする方法を学びます。また受講者は、**Ansible Vault** により **Ansible** の暗号化を管理したり、**Ansible Tower** をデプロイしたり、それを使用してシステムを管理したり、**Vagrant** とともに **DevOps** 環境で **Ansible** を使用する方法についても学びます。尚、**Ansible Tower** の内容は全**13**章中、第**11**章で触れております。
  - https://www.redhat.com/ja/services/training/do407-automation-ansible

redhat.

THANK YOU

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos