

# Ansible Tower ワークショップ

## ～ 『脱!? 自動化初級』 編 ～

レッドハット株式会社  
パートナーソリューションアーキテクト部  
岡野浩史  
最終更新日 : 2021年 9月

# 注意事項

- ・ このウェビナーではチャットを多用します。アクセス先のURLなどはチャットに貼り付ける事も多くありますので、チャットを確認できる様にしておいてください。
- ・ ご質問はチャットで頂くことも可能です。その際は “全員” 宛にご質問ください。“RH 事務局” は使わないようにお願いします。
- ・ オンラインですので、中々皆様の進捗が分かりません。オペレーションについていけない場合は遠慮なくご連絡ください。

# 質問に関して

## 遠慮なくどんどんご質問ください

不明点は音声もしくはチャットでご質問ください

チャットで質問いただく際は "全員" 宛でお願いします

"RH 事務局" は使わない様お願いします

# Experience



## Senior Solution Architect

Red Hat  
2016年9月 - 現在・(4年3ヶ月)  
Ebisu Tokyo Japan

Solution Architect for Partners



## Lead Systems Engineer

VMware  
2008年7月 - 2016年8月・(8年2ヶ月)  
Tokyo

2014.7 : Promoted to Lead Systems Engineer

Lead internal SE team

- Collect the new (not published) technologies in the internal site
- Pick up the remarkable technology



## Global Solution Team

AMD  
2006年8月 - 2008年6月・(1年11ヶ月)  
Tokyo

AMD Manager (Solution Engineer) 8/06 - 6/08

AMD Japan Ltd., Manager, duties and responsibilities include:

- Working with products such as CPU/ GPU/ High performance computing



## Advanced Systems Group

Dell  
2001年5月 - 2006年7月・(5年3ヶ月)  
日本 東京都 23 区内

Pre-Sales Consultant (4Years) and SE Manager (one year).

Dell Senior System Consultant 5/01-7/06

Dell Japan Ltd., Senior System Consultant, duties and responsibilities include:



## Research And Development Engineer

TOTO・正社員・職員  
1994年4月 - 2001年4月・(7年1ヶ月)  
神奈川県茅ヶ崎市

## 自己紹介

## 岡野 浩史

仕事: パートナーSA

SI 担当 Ansible 大好き

趣味: 星を見る事・見せる事

星のソムリエ®の資格所有 (^^)

ボランティアで天体観望会実施

その他、山登り、ジョギングも

毎朝 40km → 15km 走ってます。



プライベートのブログ(星のみ)  
<http://coral-hiro.asablo.jp/blog/>



# 自己紹介

- お名前
- 普段やっている事
- このトレーニングに期待する事

可能であれば、カメラオンをお願いします！！（\_.\_）

# 本日のハンズオントレーニング

13:00-13:20 はじめに、環境説明など

13:20-14:00 Ansible Automation Platform 座学トレーニング

14:00-14:20 LAB 2.1 ~ 2.2

14:20-14:30 休憩

14:30-15:50 LAB 2.3 ~ 2.6 (LAB 2.7はオプション)

15:50-16:00 まとめ

# 自動化への期待

# 自動化へのモヤモヤした期待

繰り返し作業では若い人が入社してくれない。クリエイティブな仕事が必要

自動化っつーか構成の確認が結構大変

DXに対応したITっていったってね基本塩漬けでしょ？

ワークフローと連携した仮想環境確認とインスタンスの払い出し？

顧客からの運用費用の削減要求がきつい

設定変更って怖いよね、基本夜中

スクリプトベースでやられるとね・・・

システム多すぎ。独自の管理ツールなんて覚えらんねえ～。

自動化の横連携ができない  
同じようなことやってのに  
自動化がサイロ化してる？

目視確認って今時本当に必要あるのか・・・

ネットワークとパブリッククラウドの設定は別の管理者へ依頼

依頼すると結構時間かかるんだよね。





# IT人材の不足が今後さらに深刻化

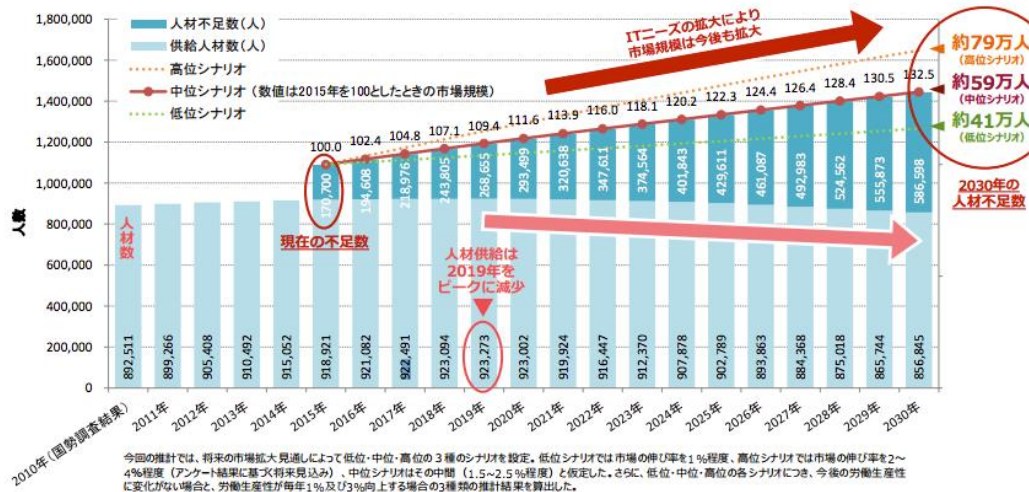
## IT人材の「不足規模」に関する推計結果

- IT関連産業の産業人口に関する将来推計（マクロ推計）の一環として、人材の不足状況や今後の見通しに関するアンケート調査結果に基づき、現在及び将来の人材不足数に関する推計も実施。
- マクロ推計によれば、**2015年時点で約17万人のIT人材が不足している**という結果になった。さらに、前頁で示されたとおり、今後IT人材の供給力が低下するにもかかわらず、ITニーズの拡大によってIT市場は今後も拡大を続けることが見込まれるため、IT人材不足は今後ますます深刻化し、**2030年には、（中位シナリオの場合で）約59万人程度まで人材の不足規模が拡大する**との推計結果が得られた。

### 2 今後のIT人材の不足規模

#### IT人材の不足規模に関する予測

- 2015年の人材不足規模：約17万人
- 2030年の人材不足規模：約59万人（中位シナリオ）  
⇒ IT人材不足は、今後ますます深刻化



IT人材は、2030年  
までに79万人不足！

2019年4月、  
働き方改革法案施行

時間外労働の上限  
月45時間、年360時間を原則

\*厚労省による概要資料より抜粋

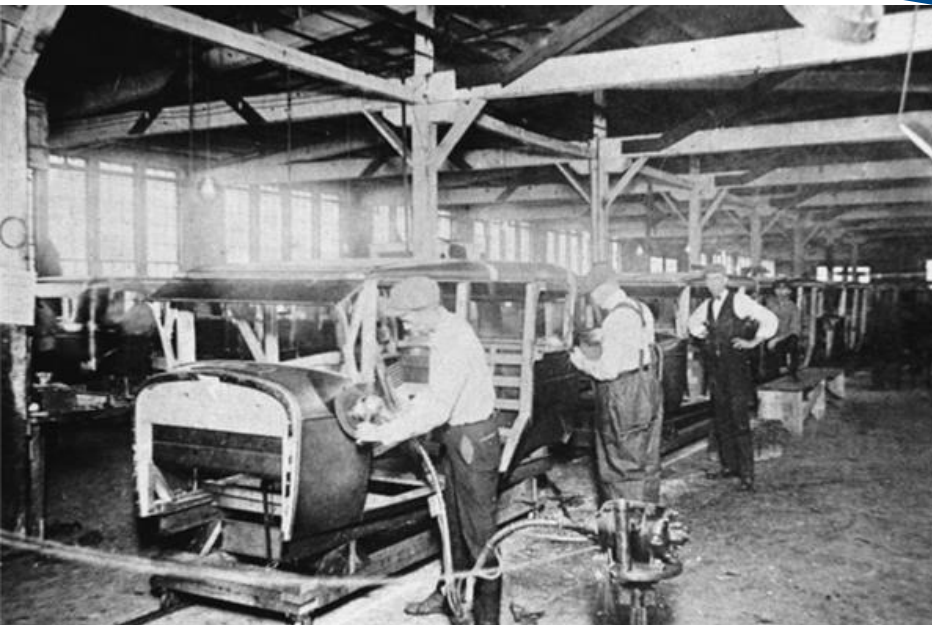
# Ansible Automation Platform概要

## ～ 自動化の課題と自動化 2.0 ～

# 自動化のパラダイムシフト 運用のサービス化 自動化1.0 → 2.0

人が作業  
(機械で補助する)

機械が作業  
(人が管理する)



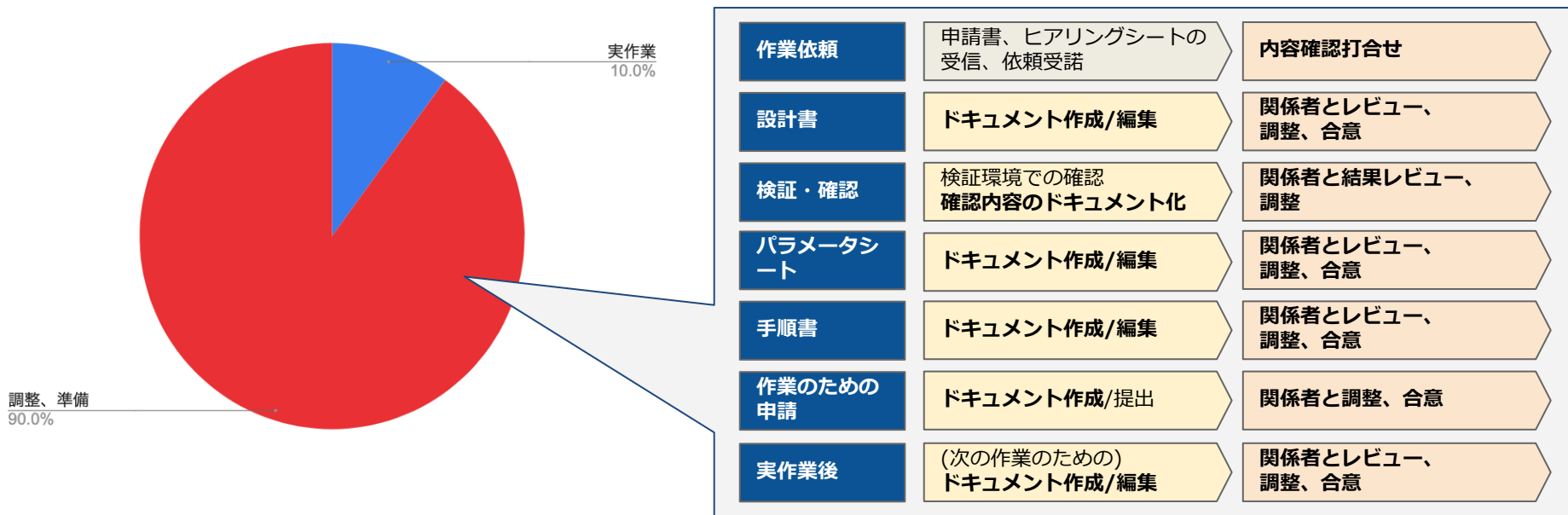
自動化以前～自動化1.0



自動化2.0

# 肥大化するコミュニケーションコスト

- ・ インフラ関係者の大半の時間は「人と人との調整(打合せ)」と「レビュー/確認/調整のためのドキュメント作成」という**間接的な作業（コミュニケーション）に費やされている**。
- ・ システム規模が大きくなり、関係者が増えることで調整量は増大していく。
- ・ 現代のインフラ作業ではこの**コミュニケーションにフォーカスした自動化**でなければ効果が出ない。



単純な作業の自動化だけでは効率が上がらない

# 自動化 2.0 に要求されるツールの性格

～ Ansible Automation Platform の特徴 ①



低い学習コスト

## Simple

誰もが読める標準化  
された自動化言語



ツールの統一

## Powerful

多様な制御対象を  
統一的手法で自動化



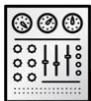
導入が容易

## Agentless

追加も簡単  
セキュリティ懸念無し

# 自動化 2.0 に要求されるツールの性格

## ～ Ansible Automation Platform の特徴 ②



ボタンの作成

### Interface

使い易い共通インターフェースで複雑な作業を『ボタン化』



ボタンを連結

### Control

ワークフローで『ボタン』を連結



ボタンを実行

### Delegation

必要な人に権限を委譲『ボタン』を実行

# シンプル - 分かりやすさの例 (プレイブック)

TARGET  
セクション

```
---  
- name: Apacheのインストールと起動  
  hosts: app  
  become: yes
```

#Playbookの説明  
#appグループが対象 (インベントリ)  
#権限昇格の有無

実行順序

TASKS  
セクション

```
tasks:
  - name: httpdのインストール
    yum: pkg=httpd state=latest
  - name: httpdを起動
    service: name=httpd state=running
```

#実行する手順の内容  
#実行時に処理毎に表示される名前



モジュール

ansible-playbookコマンドの実行  
**\$ ansible-playbook -i inventory\_file playbook.yml**



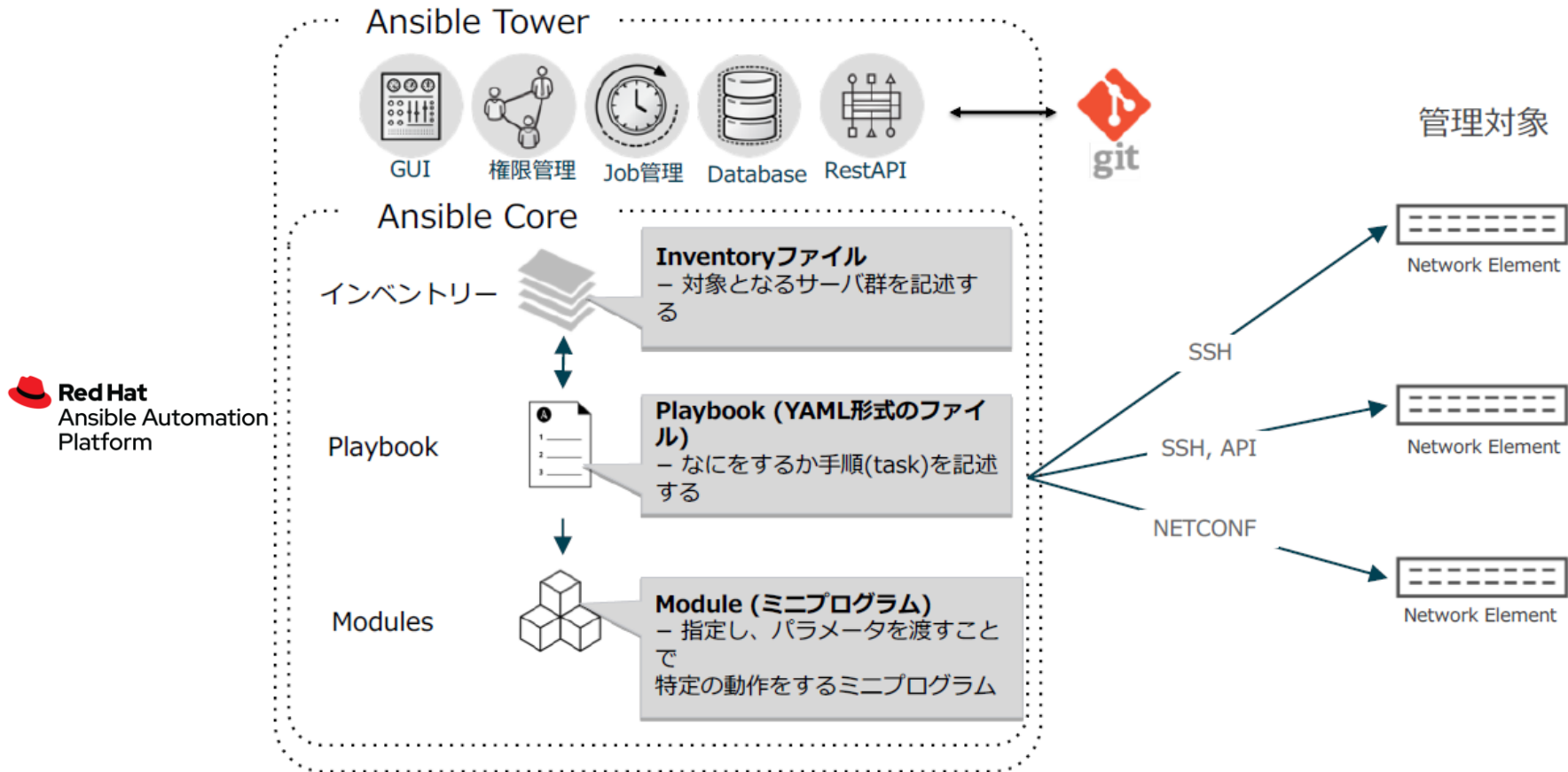
# パワフル - 様々なシステムを自動化



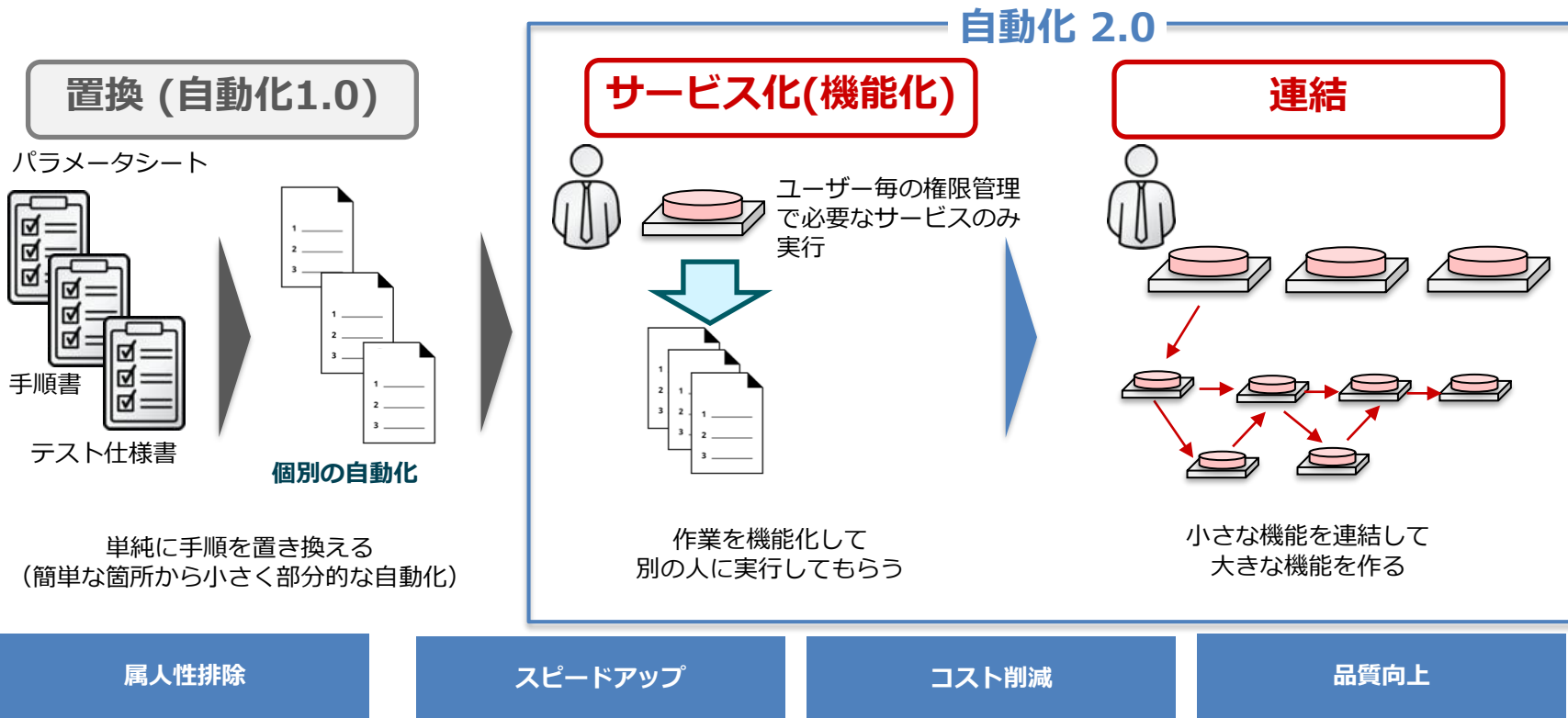
他にもたくさんあります。最新の情報はこちらをご確認ください。  
[http://docs.ansible.com/ansible/list\\_of\\_all\\_modules.html](http://docs.ansible.com/ansible/list_of_all_modules.html)



# Ansible Tower でのネットワーク機器の管理



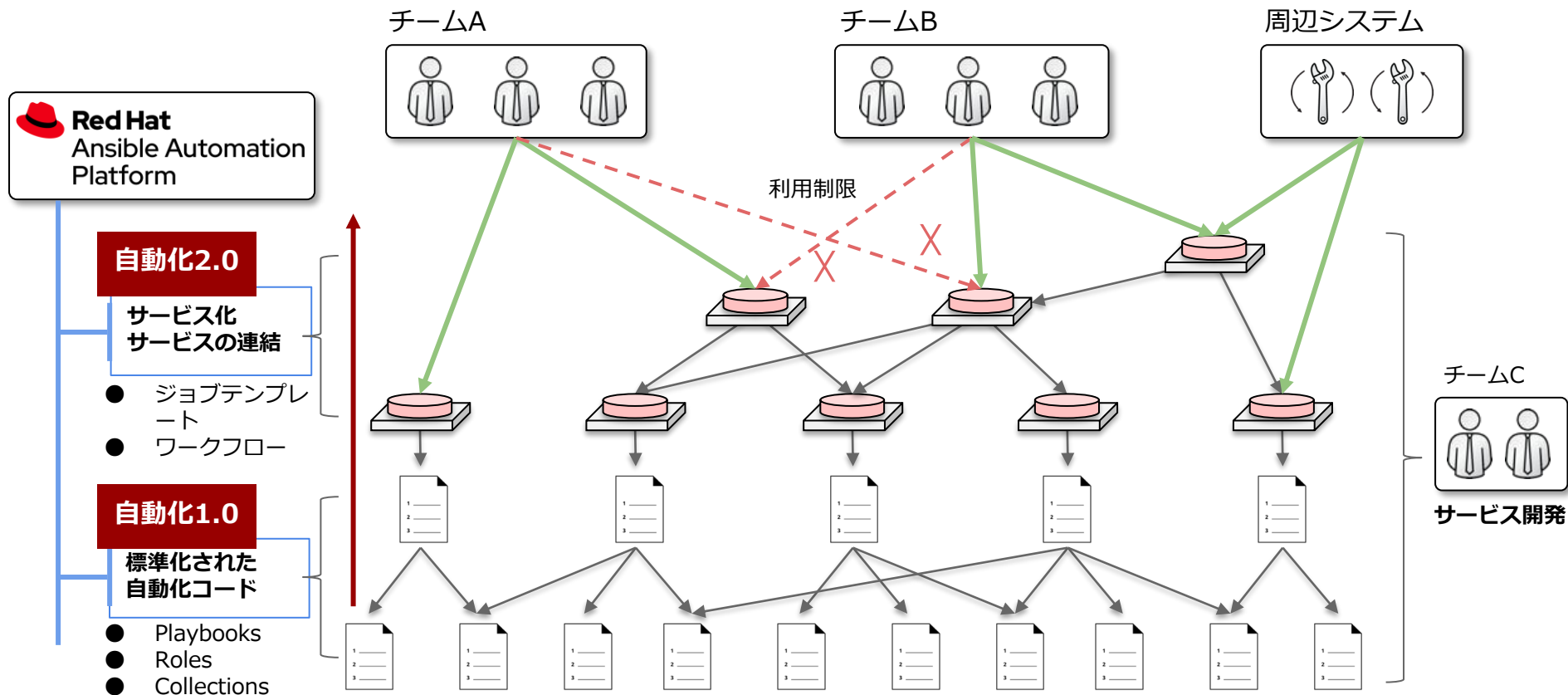
# 自動化 1.0 → 2.0 へ - 目指すは運用のサービス化



# Ansible Automation Platform

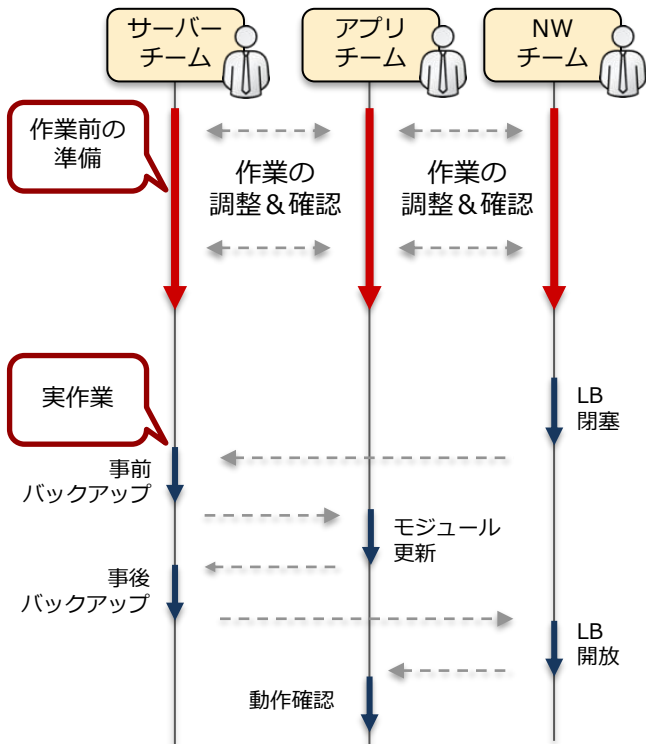
## ～ Ansible Tower 機能概要 ～

# Ansible Automation Platform による自動化2.0の実現

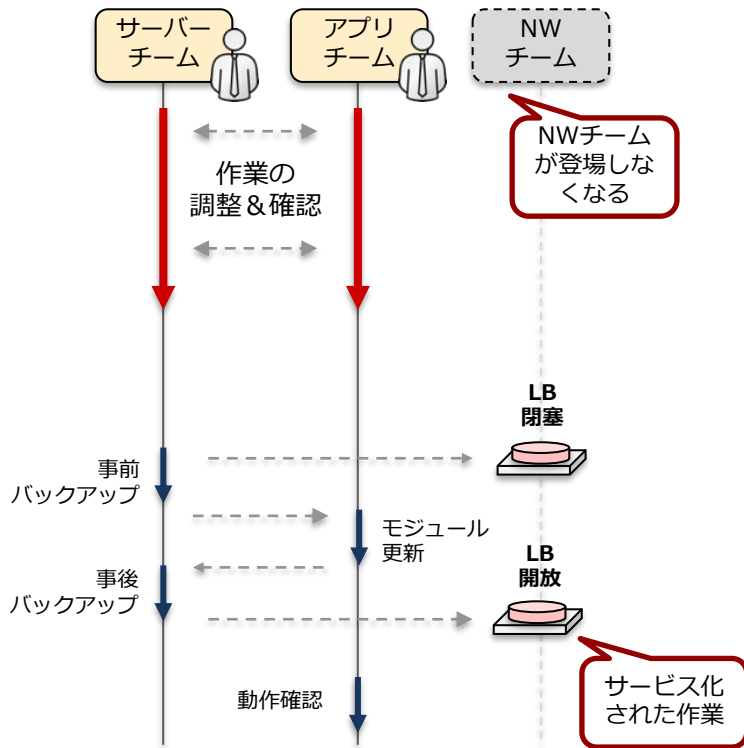


# 自動化のサービス化による『赤い部分』の削減

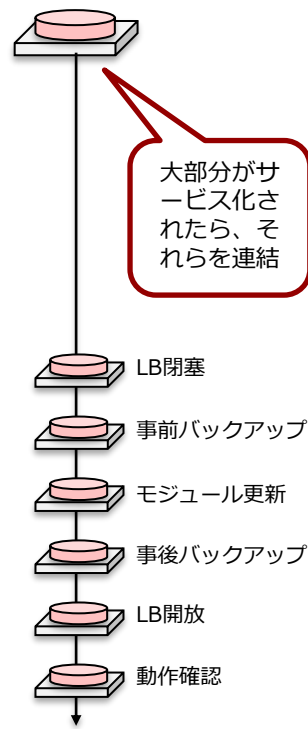
## 手作業 or 自動化1.0



## 一部サービス化された状態



## 完全自動化



WEBアプリケーションのリリース作業例

# Ansible Tower - 統合 GUI

## 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

## 監査ログ

Ansibleジョブの実行履歴をドリルダウンで監視

## 権限管理

作業実行者への権限の移譲

## ワンクリック実行

ジョブ実行をワンクリックで開始

## API & CLI

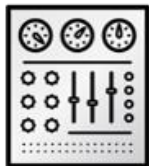
RESTful API を提供しているため外部からAPI連携可能。また、コマンドラインでの実行も可能

## 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



# Ansible Tower でエンタープライズへ



## CONTROL

自動化処理の集中管理

- Web GUIから集中管理
- スケジューリング実行
- 通知機能
- 複数のAnsible Towerサーバからの分散処理
- REST API連携



## KNOWLEDGE

監査と  
コンプライアンス

- 実行履歴ログ管理
- 認証情報の暗号化
- ワークフロー機能を活用し、共通のPlaybookを利用
- Gitとの連携



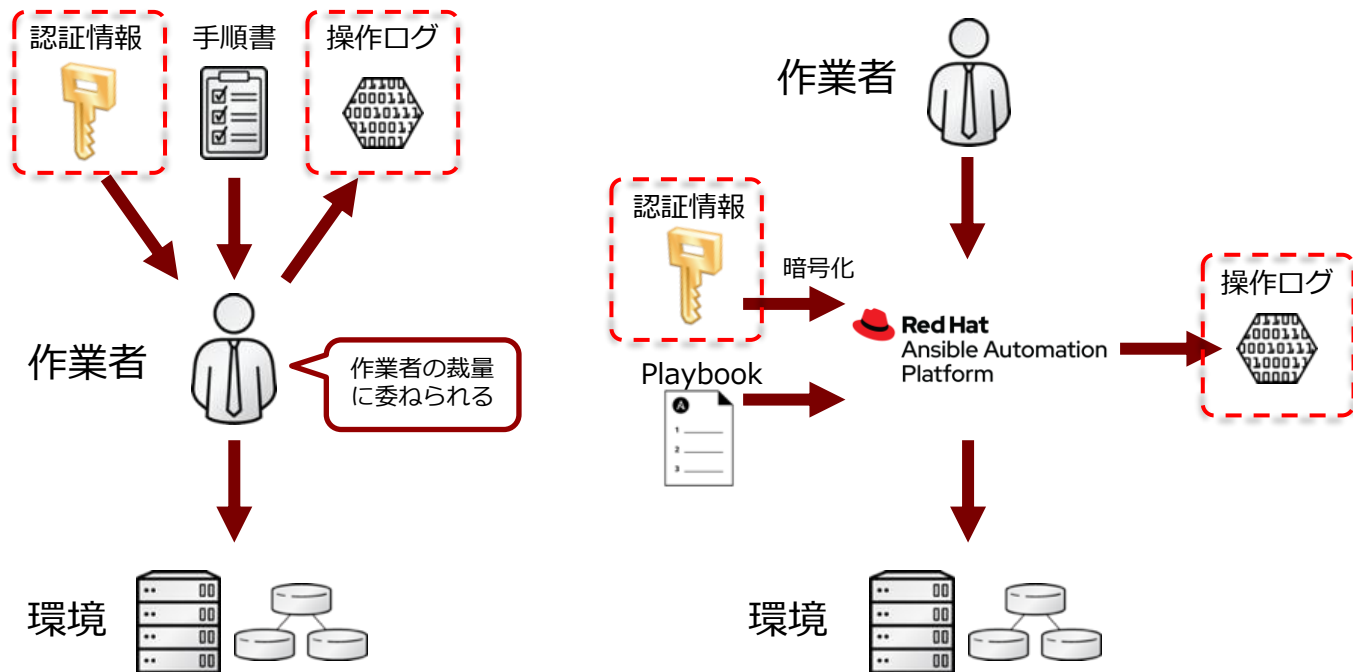
## DELEGATION

ロールベースの権限管理と  
ユーザへのセルフサービス

- 適切な人に適切な作業を実施するための権限管理設定
- セルフサービスポータル（サーベイ機能）

# 認証情報の分離と証跡管理

- 認証情報を暗号化して作業員に対して隠蔽します。
- 作業内容はPlaybookとして履歴管理され、「いつ」「誰が」「何を」「どこに」実施したかを記録します。

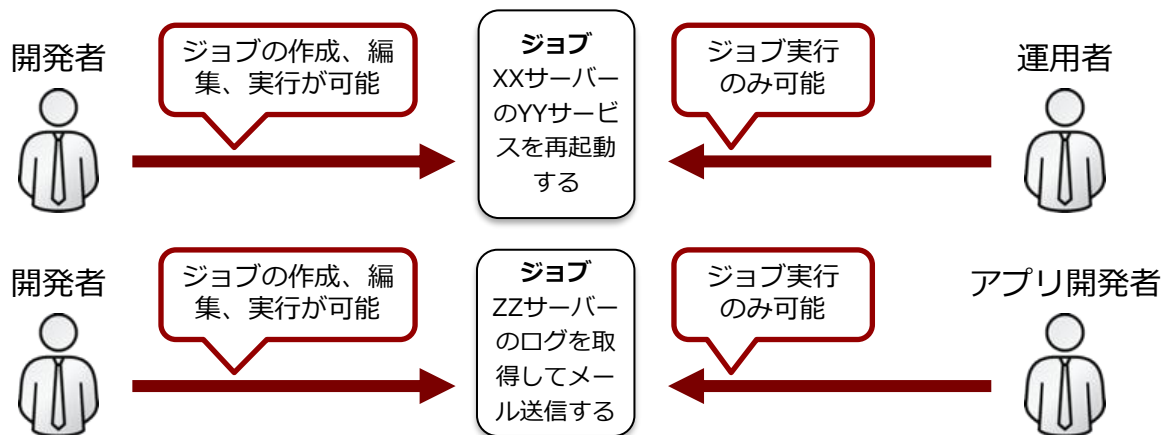




# 権限管理 (RBAC)

作成した自動化の権限を委譲することが可能。

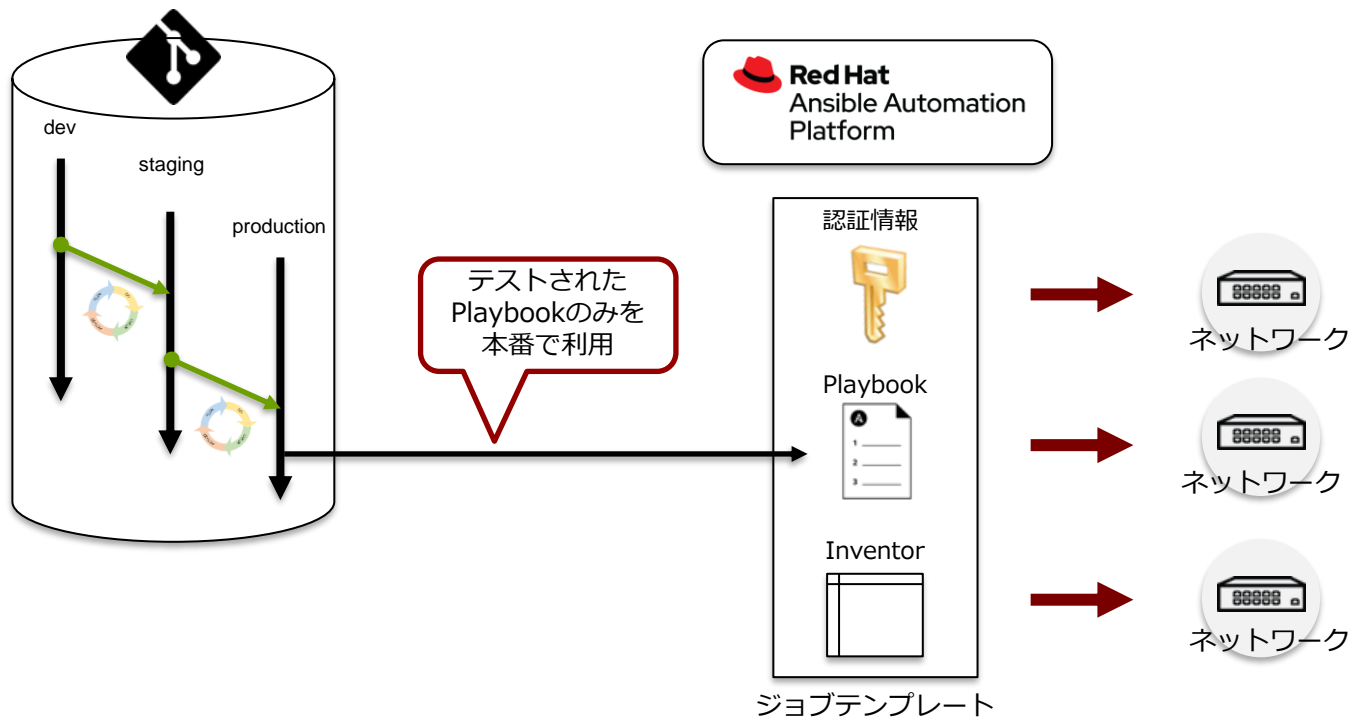
- オペレーターにサービスの自動再起動の操作「だけ」を実行「だけ」させたい。
- アプリ開発者にサーバーの自動ログ収集の操作「だけ」を実行「だけ」させたい。



- RBAC対象
- Playbook
  - インベントリ
  - 認証情報
  - ジョブ

# SCM連携

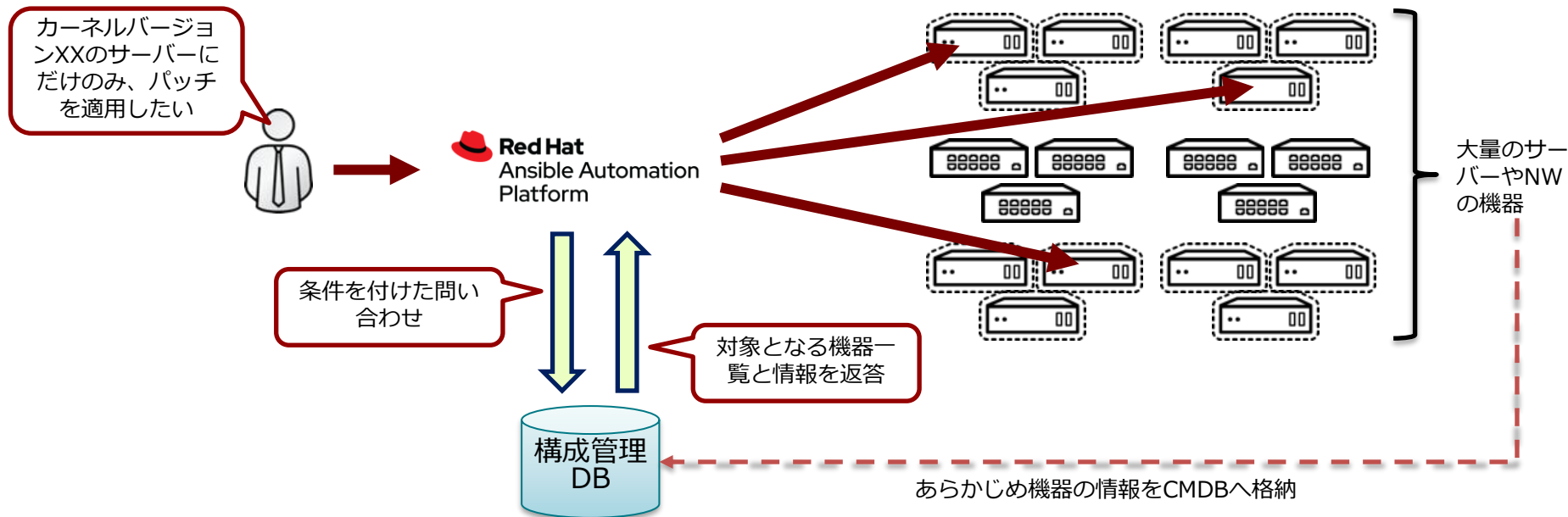
バージョン管理されていない野良Playbookや、CI等による品質管理の連携が可能。



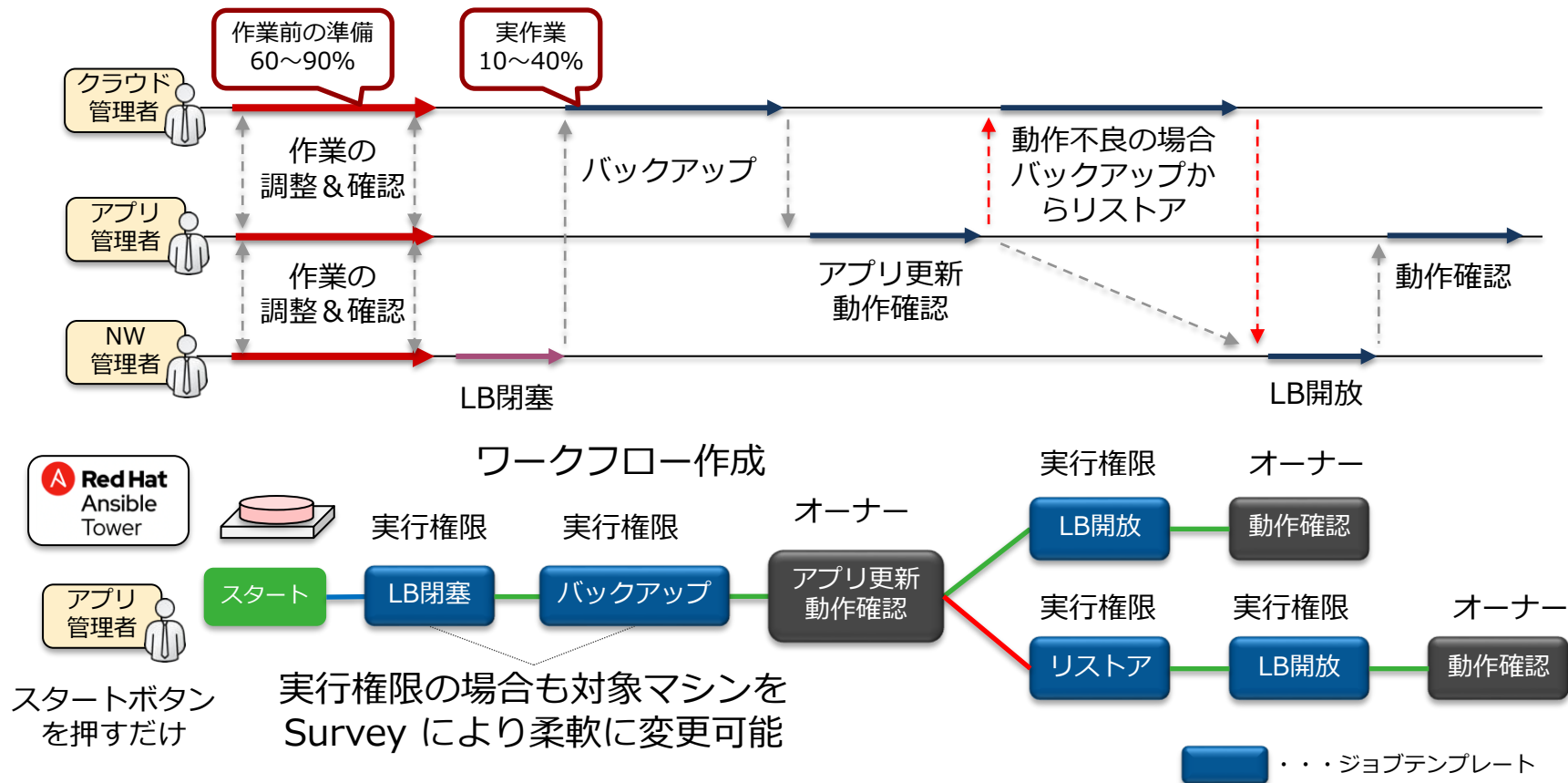
# 多数の自動化の対象を CMDB と連携して管理

多数の機器の中から特定条件に合致する対象のみに自動化を「安全、確実」に実行する。

- OSのバージョン、インストールされているパッケージ、特定のコンフィグが設定されている、など

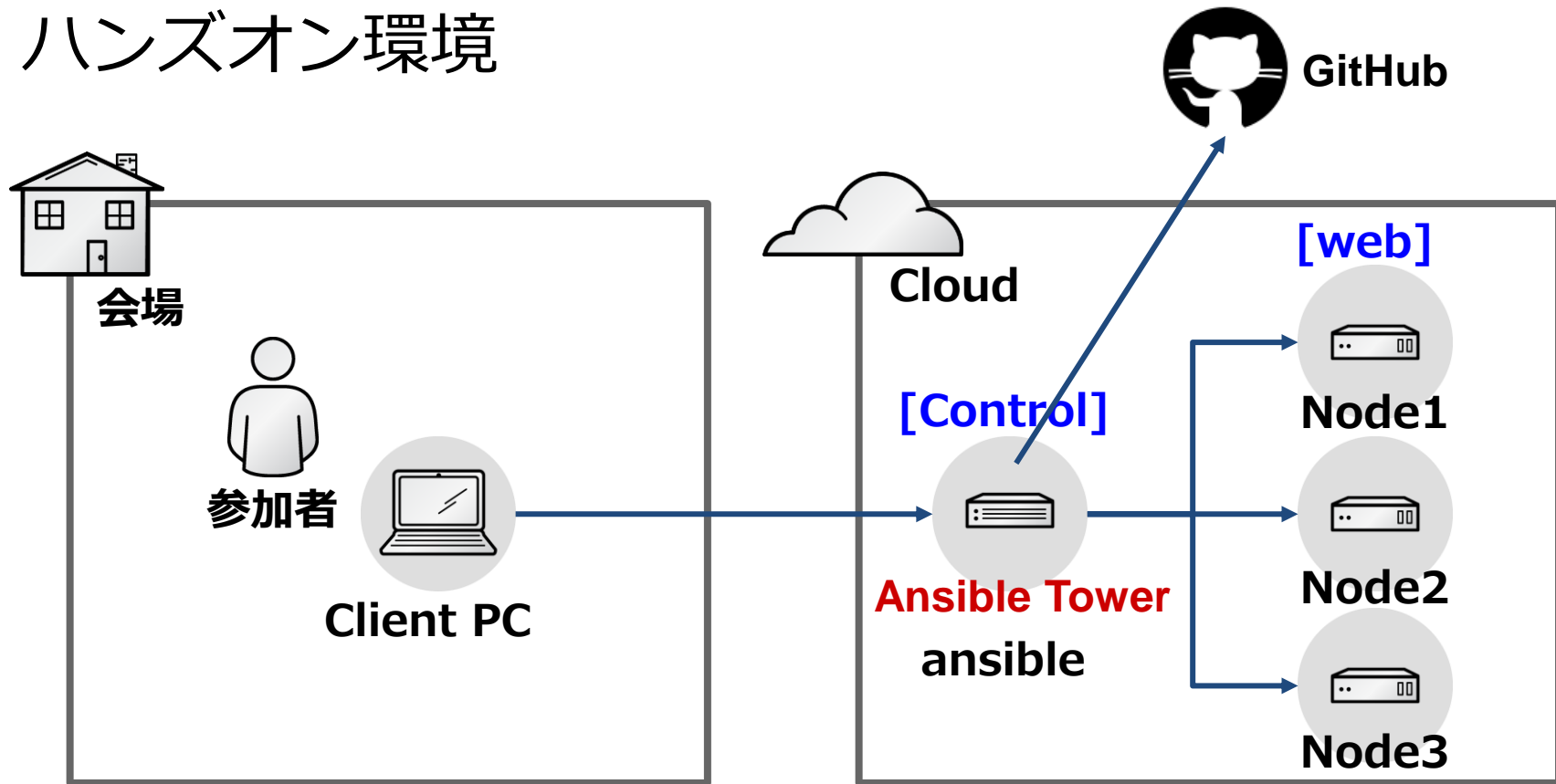


# 自動化 2.0 の実現（ワークフローによる運用サービスの連結）



# Ansible Tower Workshop

# ハンズオン環境



1人1環境 (Control Node 1台 & web Node 3台) をクラウド上に用意

# ハンズオン内容

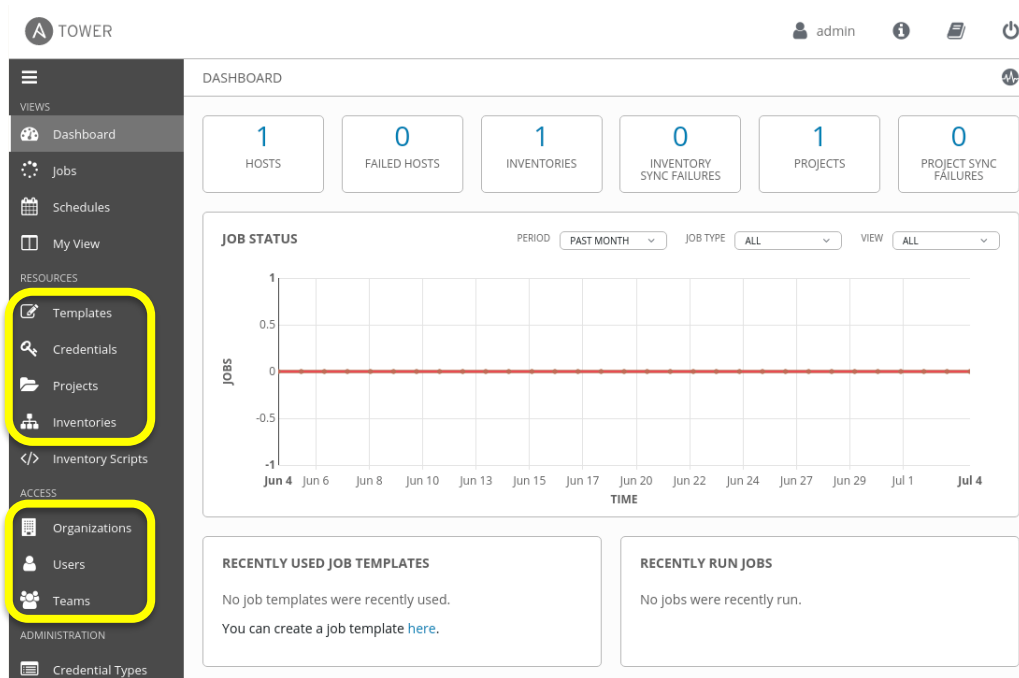
## Section 2 - Ansible Tower

- ・ 演習 2.1 - Tower の紹介
- ・ 演習 2.2 - インベントリ、認証情報、Ad Hoc コマンド
- ・ 演習 2.3 - プロジェクトとジョブテンプレート
- ・ 演習 2.4 - Survey 機能
- ・ 演習 2.5 - ロールベースのアクセス制御
- ・ 演習 2.6 - ワークフロー
- ・ 演習 2.7 - まとめ（オプション）

[https://ansible.github.io/workshops/exercises/ansible\\_rhel/README.ja.html](https://ansible.github.io/workshops/exercises/ansible_rhel/README.ja.html)

# 演習 2.1 - Tower の紹介

Ansible Tower には、Ansible にはない GUI ダッシュボードや、『プロジェクト』、『認証情報』など、独自のオブジェクトがあります。演習2.1ではそれらについて学習します。



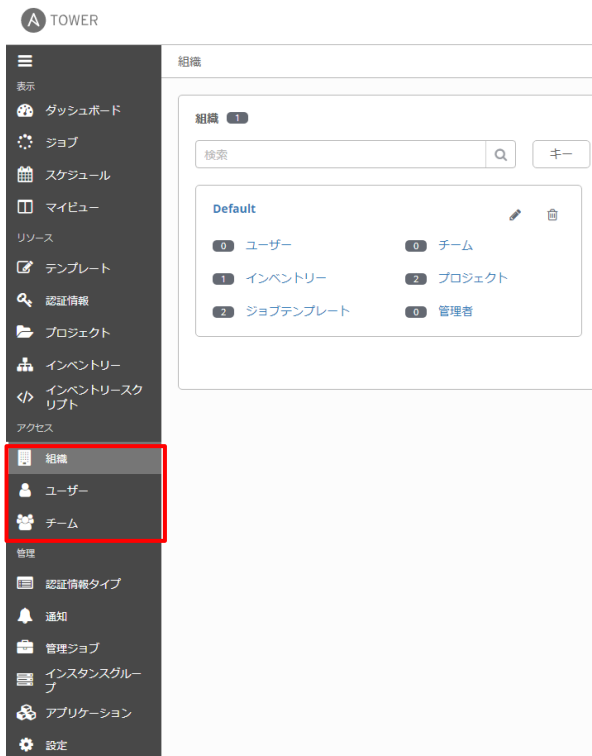
Ansible Tower のオブジェクトは沢山ありますがよく使うのは以下5つです

- ・テンプレート
- ・認証情報
- ・プロジェクト
- ・インベントリー
- ・組織

ユーザー / チーム



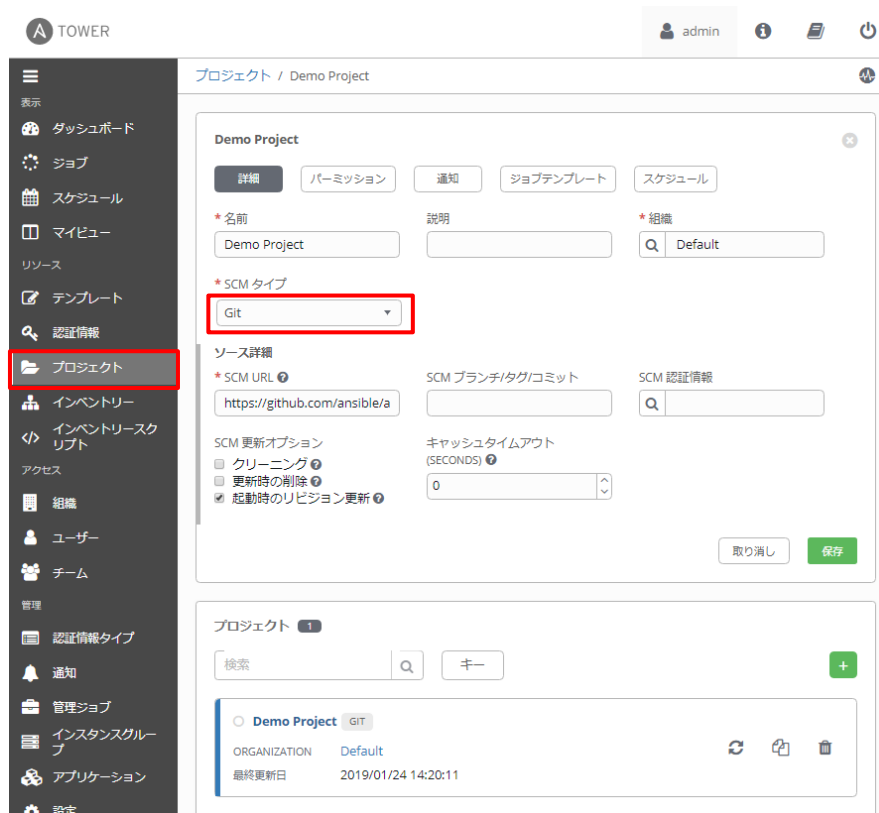
# 演習 2.1 Tower の紹介 - 組織



## 組織 (Organization)

- Ansible Tower の1番上位の概念です。ジョブを実行するために、Ansible Tower のオブジェクトをまとめて管理する仕組みを提供します。ユーザーは少なくとも1つの組織に対してアクセスが許可されている必要があります。
- 組織の下にユーザーとチーム（グループ）オブジェクトが見えていますが、これらのユーザーを組織の管理者や閲覧者などで定義します
- Ansible Tower 管理者は複数の組織を管理できます。マルチテナントという概念で考えると、組織が1つのテナントとなります。

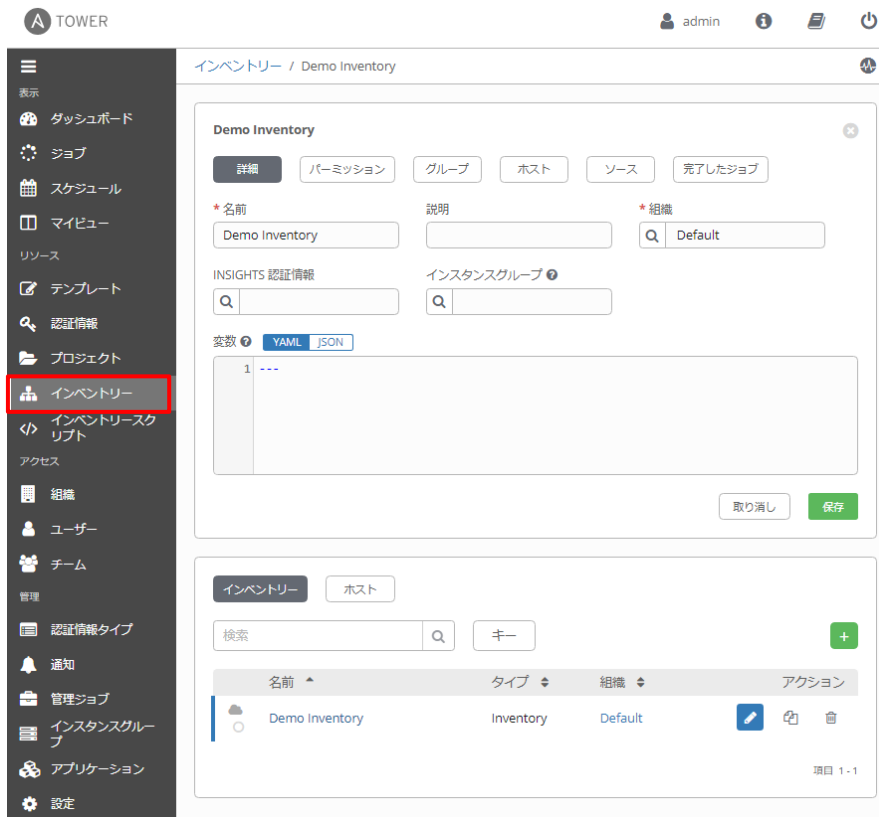
# 演習 2.1 Tower の紹介 - プロジェクト



## プロジェクト (Project)

- ・プレイブックのパスを管理する仕組み。1つのプロジェクトには、1つのプレイブックの保存場所（ローカルディレクトリ、Gitアドレスなど）が割り当てられます。ローカルディレクトリの場合は各プロジェクトのパスを `/var/lib/awx/projects/` 配下に作成します。
- ・プロジェクトではプレイブックファイルまでは指定しません。あくまでパスのみです。

# 演習 2.1 Tower の紹介 - インベントリー



## インベントリー (Inventory)

- 被管理ホストのホスト名やホストを含むグループ名などの情報を定義します。※ 認証に関する情報（ユーザー名やパスワード）は別管理
- 接続ホストに関する接続方法などを変数で定義することも可能です（Windows ホストへの WinRM 接続など）
- VMware や AWS の仮想マシン、インスタンスの情報を取得して自動的に登録するダイナミックインベントリにも対応しています。

# 演習 2.1 Tower の紹介 - 認証情報

The screenshot displays the Ansible Tower web interface. On the left sidebar, the 'Credentials' link is highlighted with a red box. The main content area shows the 'Localhost' credential configuration form. The form includes fields for 'Name' (Localhost), 'Description', 'Organization', 'Credential Type' (Machine), and 'Type Details' (User: root, Password: [masked], SSH Key: [empty]). There are also sections for 'Secrets' and 'Privilege' with 'Show' buttons. At the bottom, there are 'Cancel' and 'Save' buttons.

## 認証情報（Credential）

- ・ホストへのssh接続や、AWS、VMwareなどのクラウド環境へのID・パスワードなどの認証情報をまとめて管理するための仕組みを提供します。ssh接続のキー認証では、プライベートキーを登録することも可能です。
- ・仮想化やクラウドでは、例えば、vCenter の認証情報を登録すると、vCenterが管理する仮想マシンインベントリを入手し、インベントリに登録するダイナミックインベントリも利用可能になります。AWSなども同様です。

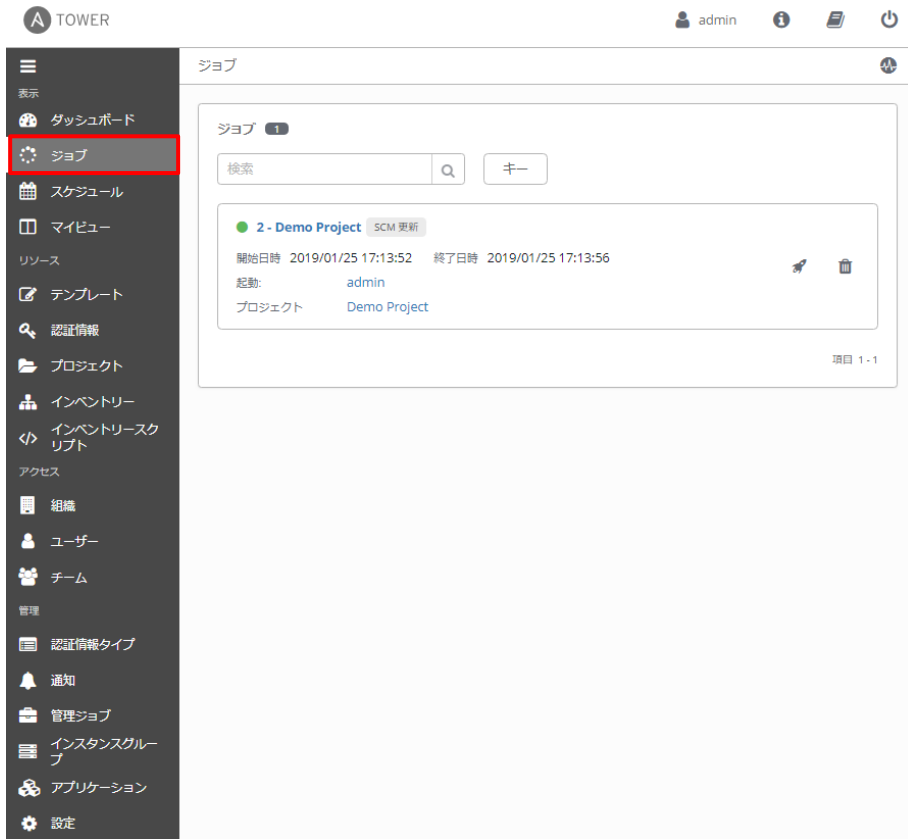
# 演習 2.1 Tower の紹介 - テンプレート

## テンプレート (Template)

- ・ ジョブテンプレートとワークフローテンプレートの統合名称です
- ・ ジョブテンプレート  
プレイブックを実行するために必要な、プロジェクト、インベントリ、認証情報、プレイブックファイル名をまとめて定義したもの
- ・ ワークフローテンプレート  
複数のジョブテンプレートをまとめて定義・実行するための仕組み
- ・ プレイブック内で記述された変数への入力 (サーベイ) にも対応しています

The screenshot displays the 'Demo Job Template' configuration interface in Ansible Tower. The left-hand navigation menu is visible, with the 'テンプレート' (Template) option highlighted. The main panel shows the configuration for a specific job template. At the top, there are tabs for '詳細' (Details), 'パーミッション' (Permissions), '通知' (Notifications), '完了したジョブ' (Completed Jobs), 'スケジュール' (Schedules), and 'SURVEY の追加' (Add Survey). The configuration fields are organized into sections: '名前' (Name) with a value of 'Demo Job Template'; '説明' (Description) which is empty; 'ジョブタイプ' (Job Type) set to '実行' (Run); 'インベントリ' (Inventory) set to 'Demo Inventory'; 'プロジェクト' (Project) set to 'Demo Project'; 'PLAYBOOK' set to 'hello\_world.yml'; '認証情報' (Credential) set to 'Demo Credential'; 'フォーク' (Forks) set to 'DEFAULT'; '制限' (Limit) which is empty; '詳細' (Details) set to '0 (Normal)'; 'ジョブタグ' (Job Tags) which is empty; 'スキップタグ' (Skip Tags) which is empty; 'ラベル' (Labels) which is empty; 'インスタンスグループ' (Instance Groups) set to '1'; and 'JOB SLICING' set to '1'. There is also an 'オプション' (Options) section with checkboxes for '権限昇格の有効化' (Privilege Escalation), 'プロビジョニングコールドバックの許可' (Allow Provisioning Cold Back), '同時実行ジョブの有効化' (Allow Concurrent Jobs), and 'ファクトのキャッシュの使用' (Use Fact Caching). At the bottom, there is a section for '追加変数' (Additional Variables) with tabs for 'YAML' and 'JSON', and a text area containing '1 ---'.

# 演習 2.1 Tower の紹介 - ジョブ



## ジョブ (Job)

テンプレートの実行状況、実行結果（成功・失敗など）を確認することができます。プロジェクトでGitなどのSCMを指定した場合、実行結果にプレイブックのバージョン情報が含まれ、以下のような情報を過去にさかのぼり確認することができます。

いつ、誰が、どのシステムに対して、どのバージョンのプレイブックを実行したか

# 演習 2.2 - インベントリー、認証情報、アドホックコマンド

## インベントリー

Ansible Tower 環境に既にセットアップされているインベントリーと認証情報についてLABガイドを見ながら確認します。インベントリーでは、"関連するグループ" に何が指定されているか、"グループ" → "Web" → "ホスト"で何が見えるかを確認してみましょう。

The image displays three screenshots of the Ansible Tower web interface, illustrating the inventory structure and navigation.

**Screenshot 1: Workshop Inventory**

- The "Hosts" tab is selected in the top navigation bar.
- The "関連するグループ" (Related Groups) section is highlighted with a red box, showing the following groups:

  - control
  - web
  - web
  - web

**Screenshot 2: web group**

- The "web" group is selected in the top navigation bar.
- The "Hosts" tab is selected in the sub-navigation bar.
- The "Hosts" section is highlighted with a red box, showing the following hosts:

  - node1
  - node2
  - node3

**Screenshot 3: web group**

- The "web" group is selected in the top navigation bar.
- The "Hosts" tab is selected in the sub-navigation bar.
- The "Hosts" section is highlighted with a red box, showing the following hosts:

  - node1
  - node2
  - node3

# 演習 2.2 - インベントリー、認証情報、アドホックコマンド

## 認証情報

Ansible Tower 環境に既にセットアップされている認証情報を確認します。ssh 接続では、パスワード認証の他、SSH秘密キー認証にも対応しています。また、認証情報タイプの🔍アイコンをクリックし様々な認証情報タイプに対応していることを確認してみましょう。

ビュー

- ダッシュボード
- ジョブ
- スケジュール
- マイビュー
- リソース
- テンプレート
- 認証情報**
- プロジェクト
- インベントリー
- インベントリースクリプト
- アクセス
- 組織
- ユーザー
- チーム

### Workshop Credential

**詳細** | パーミッション

\* 名前 ⓘ Workshop Credential

説明 ⓘ

組織 🔍 Default

\* 認証情報タイプ ⓘ 🔍 マシン

タイプの詳細

ユーザー名 🔍 ec2-user

パスワード 🔍  ☐ 起動プロンプト

SSH 秘密鍵

🔍 暗号化



## 演習 2.2 - インベントリー、認証情報、アドホックコマンド

### アドホックコマンド

`ansible -m <module_name>` コマンドで実行できるアドホックコマンドを使うと、Playbook を書くことなくモジュールを実行できます。Ansible Tower でもこの機能が実装されていますので、演習ガイドに沿って実機で確かめてみましょう。権限昇格の必要なコマンド実行方法についても確認します。

## 演習 2.3 - プロジェクトとジョブテンプレート プロジェクト

このLABでは、SCM として Git を指定。Playbook は一度 Git から Ansible Tower にダウンロードされてから実行されます。

Workshop Project

詳細 | パーミッション | 通知 | ジョブテンプレート | スケジュール

\* 名前: Workshop Project | 説明: | \* 組織: Default

\* SCM タイプ: Git

ソース詳細

\* SCM URL: https://github.com/ansible/workshop-examp | SCM ブランチタグ/コミット: | SCM REFSPEC:

SCM 認証情報: | SCM 更新オプション: ☐ クリーニング ☒ 更新時のデプロイ ☒ 起動時のリビジョン更新 ☐ プランタの正番許可

キャッシュタイプ (使用):

ANSIBLE 環境: デフォルト環境の使用

取り消し | 保存

プロジェクト 2

検索 | キー

Ansible official demo project | GIT

Workshop Project | GIT

起動時に SCM とダウンロード済みの物を比較。SCMが新しい場合、再ダウンロードし Playbook を実行

手動同期ボタン

SCM 同期(Playbook  
ダウンロード) 済み

## 演習 2.3 - プロジェクトとジョブテンプレート

### ジョブテンプレート

Playbook では3種類のモジュールを使って以下を行います。

- yum モジュール  
httpd と firewalld → 最新 Ver
- service モジュール  
firewalld と httpd → 稼働状態
- firewalld モジュール  
http ポートの開放

Playbook では "hosts : all" が指定されていますが、ジョブテンプレートの "制限" で web (グループ) が指定されていますので、実際に実行されるホストは、"web" に所属する3台となります

```
---
- name: Apache server installed
  hosts: all

  tasks:
    - name: latest Apache version installed
      yum:
        name: httpd
        state: latest

    - name: latest firewalld version installed
      yum:
        name: firewalld
        state: latest

    - name: firewalld enabled and running
      service:
        name: firewalld
        enabled: true
        state: started

    - name: firewalld permits http service
      firewalld:
        service: http
        permanent: true
        state: enabled
        immediate: yes

    - name: Apache enabled and running
      service:
        name: httpd
        enabled: true
        state: started
```

# 演習 2.3 - プロジェクトとジョブテンプレート

## ジョブテンプレート

いつ誰がどのホストに対して、どのバージョンの Playbook を実行したのかなど、ジョブ履歴として確認可能

The screenshot displays the Ansible Tower interface. On the left, the 'DETAILS' panel shows job information: STATUS is 'Successful', STARTED and FINISHED times are from 7/10/2019, JOB TEMPLATE is 'Install Apache', JOB TYPE is 'Run', LAUNCHED BY is 'admin', INVENTORY is 'Workshop Inventory', PROJECT is 'Ansible Workshop Examples', REVISION is '7d82d87', PLAYBOOK is 'rhel/apache/apache\_install.yml', CREDENTIAL is 'Workshop Credentials', ENVIRONMENT is '/var/lib/awx/venv/ansible', EXECUTION NODE is 'localhost', and INSTANCE GROUP is 'tower'. On the right, the 'Install Apache' job execution log is shown, detailing the steps: SSH password, PLAY [Apache server installed], TASK [Gathering Facts], TASK [latest Apache version installed], TASK [latest firewallld version installed], TASK [firewalld enabled and running], and TASK [firewalld permits http service].

DETAILS	VALUES
STATUS	Successful
STARTED	7/10/2019 11:15:43 PM
FINISHED	7/10/2019 11:16:06 PM
JOB TEMPLATE	Install Apache
JOB TYPE	Run
LAUNCHED BY	admin
INVENTORY	Workshop Inventory
PROJECT	Ansible Workshop Examples
REVISION	7d82d87
PLAYBOOK	rhel/apache/apache_install.yml
CREDENTIAL	Workshop Credentials
ENVIRONMENT	/var/lib/awx/venv/ansible
EXECUTION NODE	localhost
INSTANCE GROUP	tower

```
1 SSH password:
2
3 PLAY [Apache server installed] ***** 23:15:47
4
5 TASK [Gathering Facts] ***** 23:15:47
6 ok: [node2]
7 ok: [host1]
8 ok: [node3]
9
10 TASK [latest Apache version installed] ***** 23:15:48
11 ok: [node3]
12 ok: [node2]
13 ok: [host1]
14
15 TASK [latest firewallld version installed] ***** 23:15:52
16 changed: [node2]
17 changed: [host1]
18 changed: [node3]
19
20 TASK [firewalld enabled and running] ***** 23:15:58
21 changed: [node2]
22 changed: [node3]
23 changed: [host1]
24
25 TASK [firewalld permits http service] ***** 23:15:59
26 changed: [node2]
27 changed: [node3]
28 changed: [host1]
29
```

## 演習 2.4 - Survey

### **Survey**

Ansible Tower 独自のオブジェクト名ですが、変数に値を入力するための仕組みのことを Survey と言います。この演習では、先ほどインストールした httpd のデフォルトページの表示内容 (index.html) を Survey に入力した内容によりカスタマイズする方法を学びます。利用するのは template モジュールです。

## 演習 2.4 - Survey

The image shows a sequence of Ansible playbooks in a dark-themed editor. The first panel shows the 'main.yml' file with a 'roles' list containing 'role\_apache'. The second panel shows the 'role\_apache/tasks/main.yml' file, which includes tasks for installing Apache, updating the firewall, and deploying the index.html file. The third panel shows the 'role\_apache/templates/index.html.j2' file, which contains HTML code with Jinja2 placeholders for survey data. A yellow box highlights the survey field names in the HTML template, and an arrow points from the explanatory text to this box.

workshop-examples / rhel / apache / apache\_role\_install.yml

IPVSean forcing more specific hosts declaration and updating logo

2 contributors

6 lines (5 sloc) | 80 Bytes

```
1 ---
2 - name: Ensure Apache installation
3   hosts: web
4
5   roles:
6     - role_apache
```

workshop-examples / rhel / apache / roles / role\_apache / tasks / main.yml

liquidat Add apache role playbook

1 contributor

42 lines (36 sloc) | 791 Bytes

```
1 ---
2 - name: latest Apache version installed
3   yum:
4     name: Mtpd
5     state: latest
6     notify: apache-restart
7
8 - name: latest firewall version installed
9   yum:
10    name: firewall
11    state: latest
12
13 - name: firewall enabled and running
14   service:
15     name: firewall
16     enabled: true
17     state: started
18
19 - name: firewall permits http service
20   firewall:
21     service: Mtpd
22     permanent: true
23     state: enabled
24     immediate: yes
25
26 - name: Apache enabled and running
27   service:
28     name: httpd
29     enabled: true
30     state: started
31
32 - name: Ensure proper Apache configuration
33   copy:
34     src: httpd.conf
35     dest: /etc/httpd/conf/httpd.conf
36     notify: apache-restart
37
38 - name: deploy index.html
39   template:
40     src: index.html.j2
41     dest: /var/www/html/index.html
42     notify: apache-restart
```

workshop-examples / rhel / apache / roles / role\_apache / templates / index.html.j2

liquidat Add apache role playbook

1 contributor

7 lines (7 sloc) | 187 Bytes

```
1 <html>
2 <body>
3 <h1>Apache is running fine</h1>
4 <h1>This is survey field "First Line": {{ first_line }} </h1>
5 <h1>This is survey field "Second Line": {{ second_line }}</h1>
6 </body>
7 </html>
```

Role を使っているので階層が深くなっていますが、template モジュールによってコピーされるのはこちらのファイル。その際、単なるコピーではなく first\_line と second\_line という2つの変数に、Survey の値が入力されてホストの /var/www/html/index.html として保存されます。

## 演習 2.5 - ロールベースのアクセス制御

Ansible Tower の優れた機能として柔軟な権限の設定と委譲があります。こちらを演習で確認しましょう。演習内容は以下の通りです。

- ・新しいユーザー (wweb) を作成
- ・新しいチーム (web Content) の作成  
作成したチームに新しいユーザーを追加  
web Content チームにジョブテンプレート "Create Index.html" の "実行" 権限を付与
- ・wwebでAnsible Towerにログインし、ジョブテンプレート Create Index.html を実行

## 演習 2.5 - ロールベースのアクセス制御 Tips

### - Ansible Tower (管理者・監査担当・ユーザー)

Tower 管理者・・・複数の組織を一括管理・閲覧する権限を持ちます。

Tower 監査担当・・・全てのオブジェクトの閲覧権限を持ちます。

Tower ユーザー・・・別途与えられた権限のみ利用可能となります。

### - 組織 (管理者・メンバー・読み込み)

組織内に存在する以下のオブジェクトに対する管理・閲覧権限を定義

プロジェクト、インベントリ、テンプレート、ホスト認証情報

単一の組織の管理者は、**Towerのユーザー + 組織の管理者権限** で設定します

例えば、以下の様な権限の設定が可能です

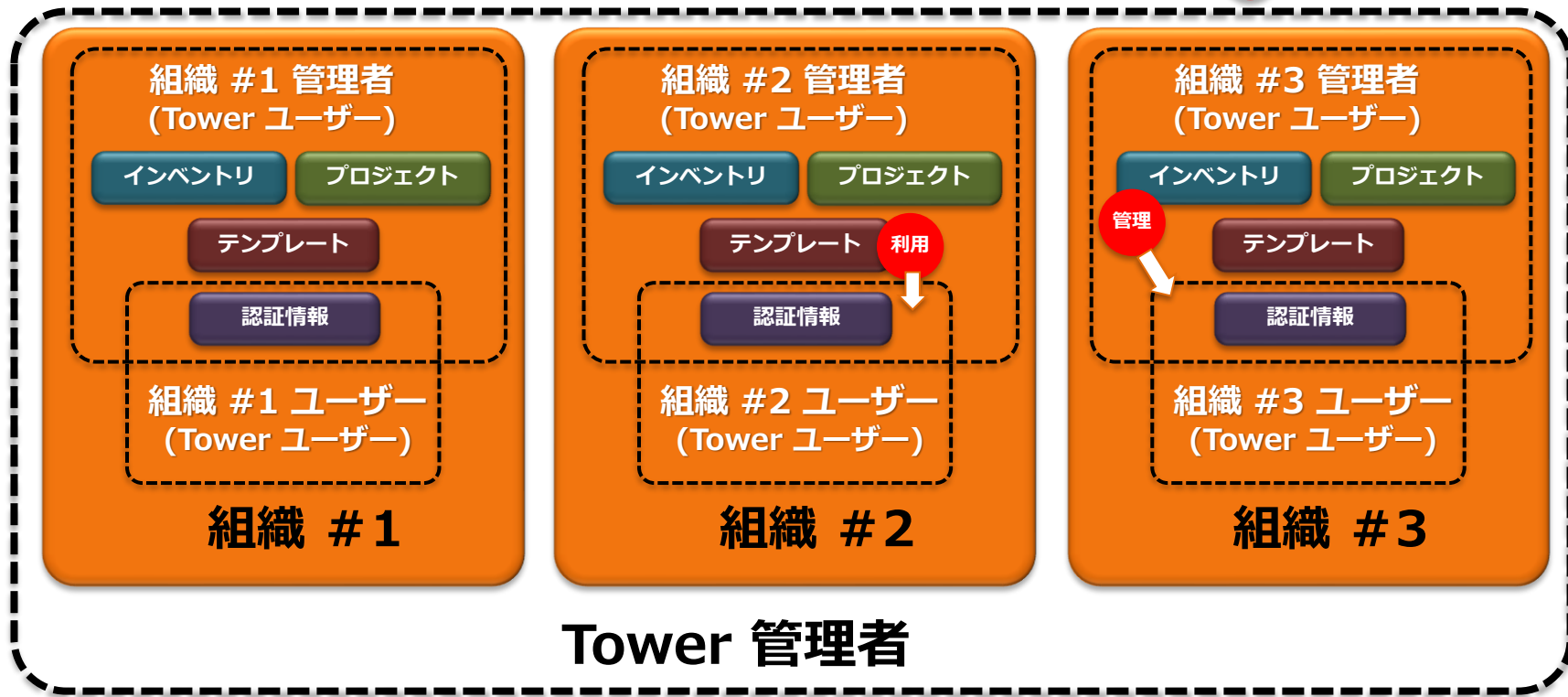
- ・ A・・・Tower の管理者
- ・ B・・・Tower ユーザー、組織 A 管理者
- ・ C・・・Tower ユーザー、組織 A メンバー、組織 A 特定テンプレートの実行
- ・ D・・・Tower ユーザー、組織 B 管理者、組織 A 特定インベントリの管理者



## 演習 2.5 - ロールベースのアクセス制御 Tips

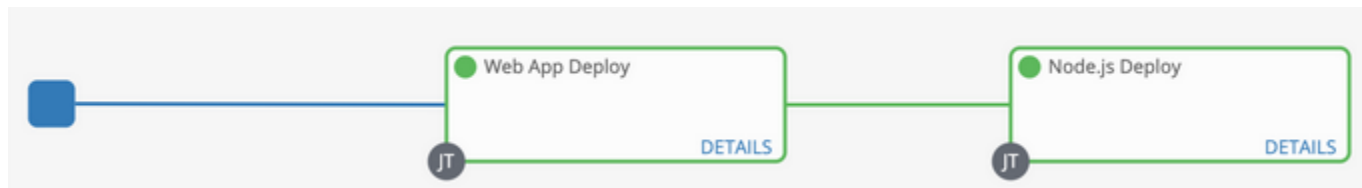
権限の及ぶ範囲

権限移譲の例



## 演習 2.6 - ワークフロー

その他、Ansible Tower の優れた機能としてワークフローがあります。複数の Playbook（ジョブテンプレート）を連結して実行できる機能です。ジョブテンプレート実行の正否によって次に実行されるジョブテンプレートを変える事も容易に定義可能です。演習ガイドに沿って、ワークフローを作成し、2つのジョブテンプレートをワンクリックで実行してみてください。



## 演習 2.7 - まとめ

node1 & node3 . . . 開発用の Web Server

node2 . . . 本番用の Web Server

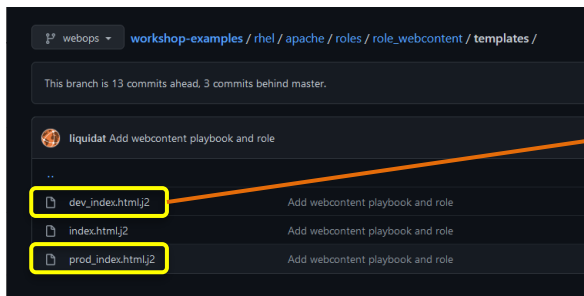
と定義し、デフォルトで表示される index.html 開発用、本番用のコメントが表示されるように変数を定義してみましようという演習です。

main.yml

```
[...]  
- name: Deploy index.html from template  
  template:  
    src: "{{ stage }}_index.html.j2"  
    dest: /var/www/html/index.html  
    notify: apache-restart
```

stage 変数の値で呼び出す  
template ファイルを選別  
値は "prod" or "dev" です

template ホルダ



dev\_index.html.j2

```
4 lines (4 sloc) 85 Bytes  
1 <body>  
2 <h1>This is a development webserver, have fun!</h1>  
3 {{ dev_content }}  
4 </body>
```

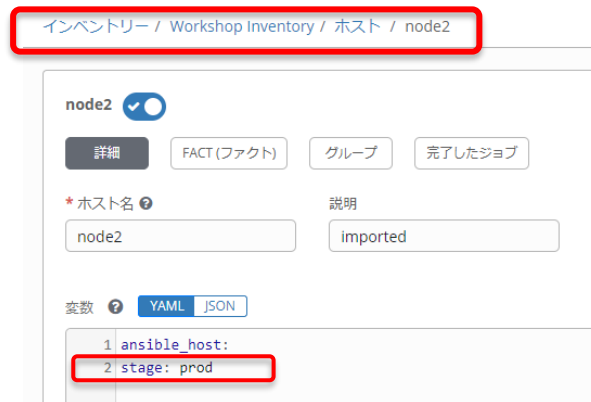
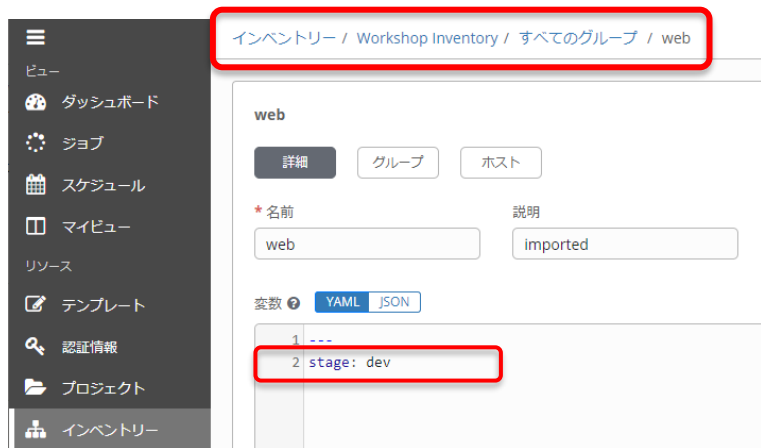
dev\_content 変数がありま  
す。htmlの中身なので、開発  
用のWeb Serverであることを  
示す値を入力しましょう。

## 演習 2.7 - まとめ

変数の定義方法は沢山ありますが、私はこのようにしました。

stage: dev ---> インベントリーの web グループ全体変数

stage: prod ---> node2 のインベントリー変数（上記は上書きされます）



## 演習 2.7 - まとめ

template ファイル内の変数は、ジョブテンプレートの中で指定しました。

dev\_content: 開発用です、ご自由に！

prod\_content: 本番用です、ご注意ください！

Dashboard  
ジョブ  
スケジュール  
マイビュー  
リソース  
テンプレート  
認証情報  
プロジェクト  
インベントリ  
インベントリースクリプト  
アクセス  
組織  
ユーザー  
チーム  
管理  
認証情報タイプ  
通知  
管理ジョブ  
インスタンスグループ  
アプリケーション  
設定

### Create Web Content

詳細 | パーミッション | 通知 | 完了したジョブ | スケジュール | SURVEY の追加

\* 名前: Create Web Content | 説明: | \* ジョブタイプ: 実行 | ☐ 起動プロンプト

\* インベントリ: Workshop Inventory | ☐ 起動プロンプト | \* プロジェクト: Workshop Project | \* PLAYBOOK: rhel/apache/webcontent.yml | ☐ 起動プロンプト

認証情報: | ☐ 起動プロンプト | フォーク: 0 | 制限: web | ☐ 起動プロンプト

\* 詳細: 0 (Normal) | ☐ 起動プロンプト | ジョブタグ: | ☐ 起動プロンプト | スキップタグ: | ☐ 起動プロンプト

ラベル: | ANSIBLE 環境: デフォルト環境の使用 | インスタンスグループ: |

ジョブスライス: 1 | タイムアウト: 0 | 変更の表示: ☐ 起動プロンプト

オプション

- ☒ 権限昇格の有効化
- ☐ プロビジョニングコールバックの有効化
- ☐ WEBHOOK の有効化
- ☐ 同時実行ジョブの有効化
- ☐ ファクトキャッシュの有効化

追加変数: ☐ YAML ☐ JSON | ☐ 起動プロンプト

```
1 ---
2 dev_content: 開発用です、ご自由に！
3 prod_content: 本番用です、ご注意ください！
```

これで全て終了です！  
お疲れ様でした！！

# Thank you

Red Hat is the world's leading provider of enterprise open source software solutions. Award-winning support, training, and consulting services make Red Hat a trusted adviser to the Fortune 500.