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

Red Hat

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)
神奈川県茅ケ崎市

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 ~~10km~~ → 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/

Red Hat

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、カメラオンでお願いします！！（_._）

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 Ansble Automatio 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

DXに対応したITっていったってね基本塩漬けでしょ？

自動化っつーか構成の確認が結構大変

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

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ぇ～。

自動化の横連携ができない同じようなことやってん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 ～

Red Hat

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）

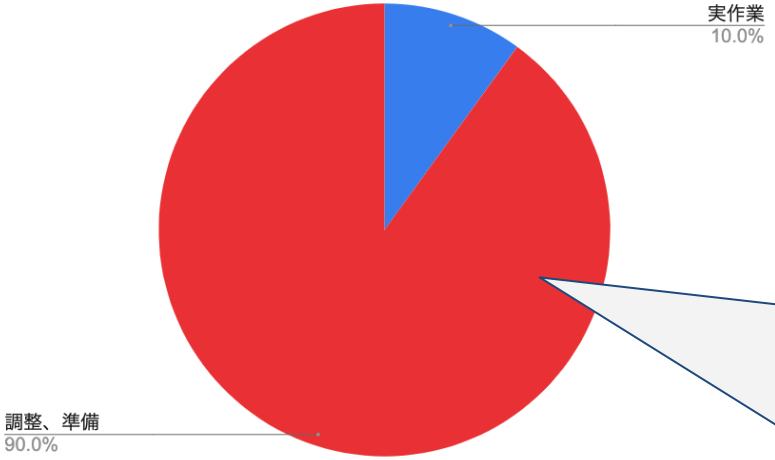
機械が作業
（人が管理する）

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。

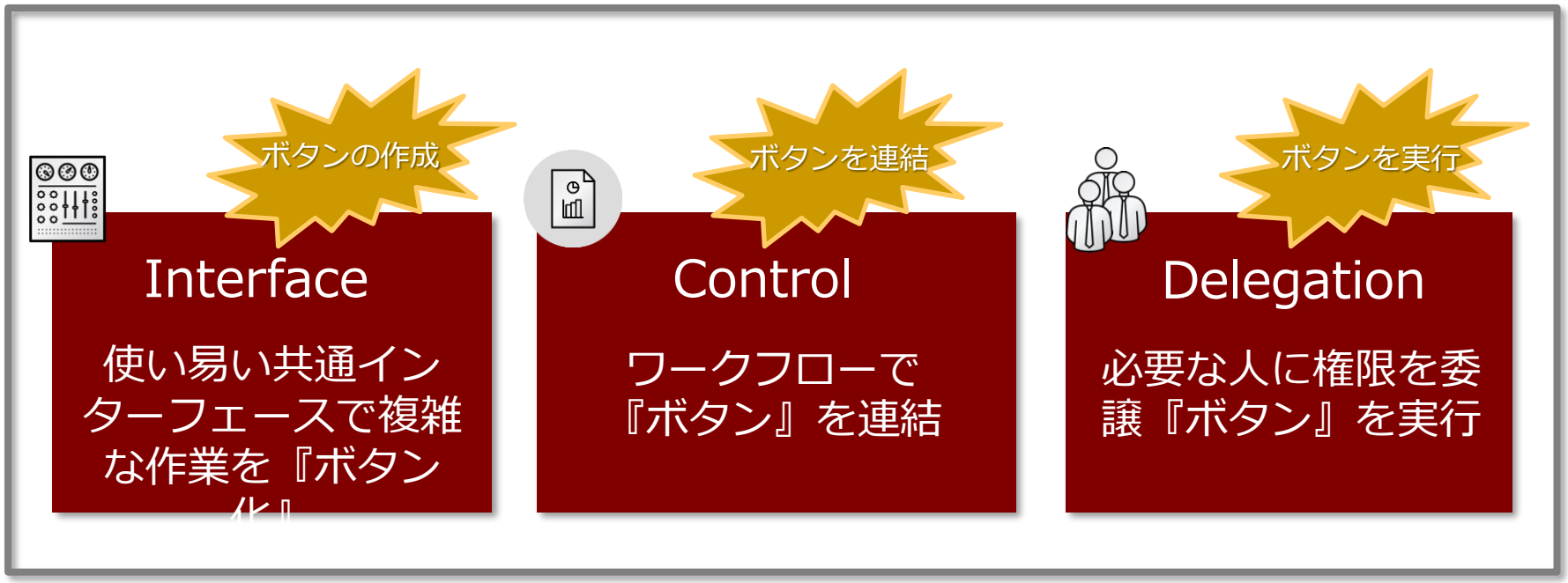


作業依頼	申請書、ヒアリングシートの受信、依頼受諾	**内容確認打合せ**
設計書	**ドキュメント作成/編集**	**関係者とレビュー、調整、合意**
検証・確認	検証環境での確認 **確認内容のドキュメント化**	**関係者と結果レビュー、調整**
パラメータシート	**ドキュメント作成/編集**	**関係者とレビュー、調整、合意**
手順書	**ドキュメント作成/編集**	**関係者とレビュー、調整、合意**
作業のための申請	**ドキュメント作成**/提出	**関係者と調整、合意**
実作業後	(次の作業のための) **ドキュメント作成/編集**	**関係者とレビュー、調整、合意**

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

自動化 2.0 に要求されるツールの性格
～ Ansible Automation Platform の特徴 ②



シンプル - 分かりやすさの例（プレイブック）

TARGET
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（インベントリ）
  become: yes                               #権限昇格の有無

  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: httpdを起動
    service: name=httpd state=running
```

タスク

実行順序

モジュール

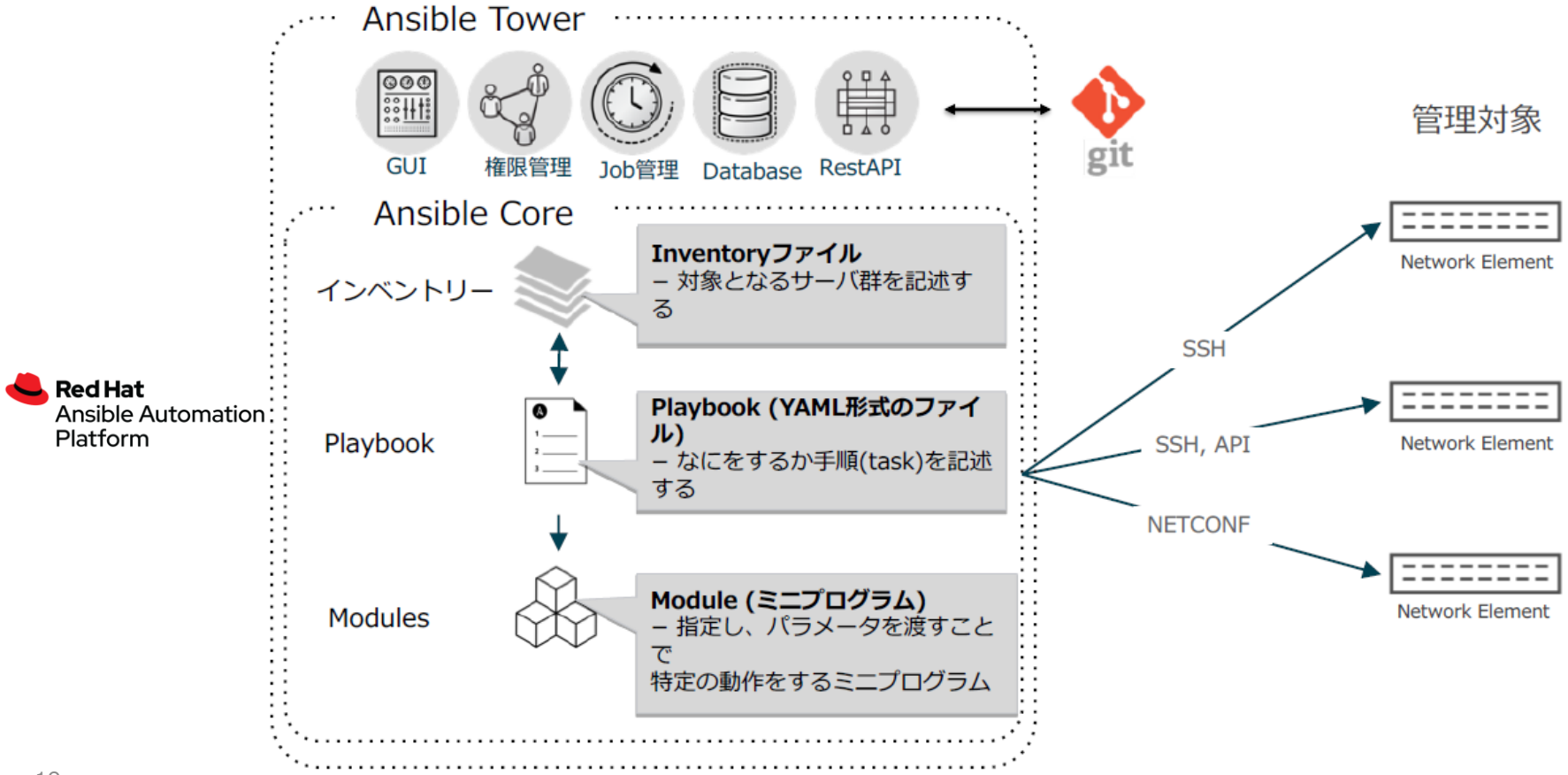
ansible-playbookコマンドの実行

$ ansible-playbook –i inventory_file playbook.yml

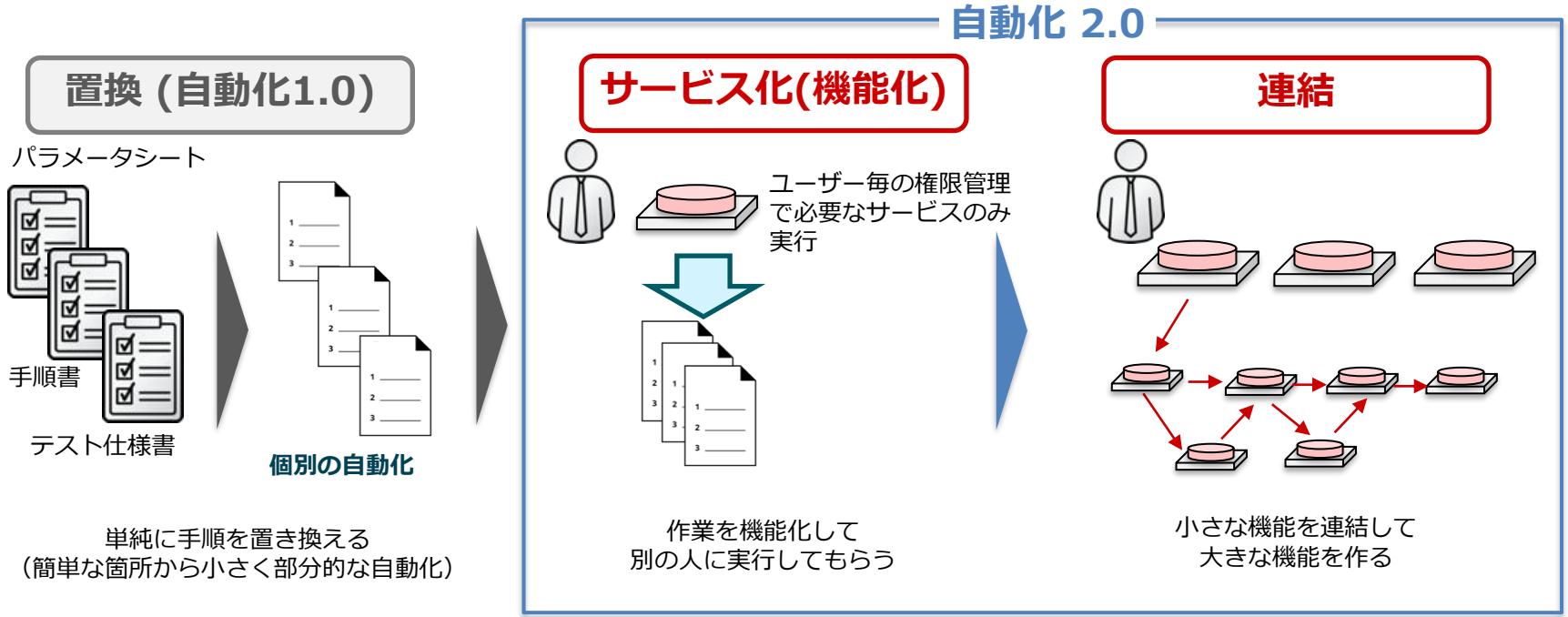
パワフル - 様々なシステムを自動化

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

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

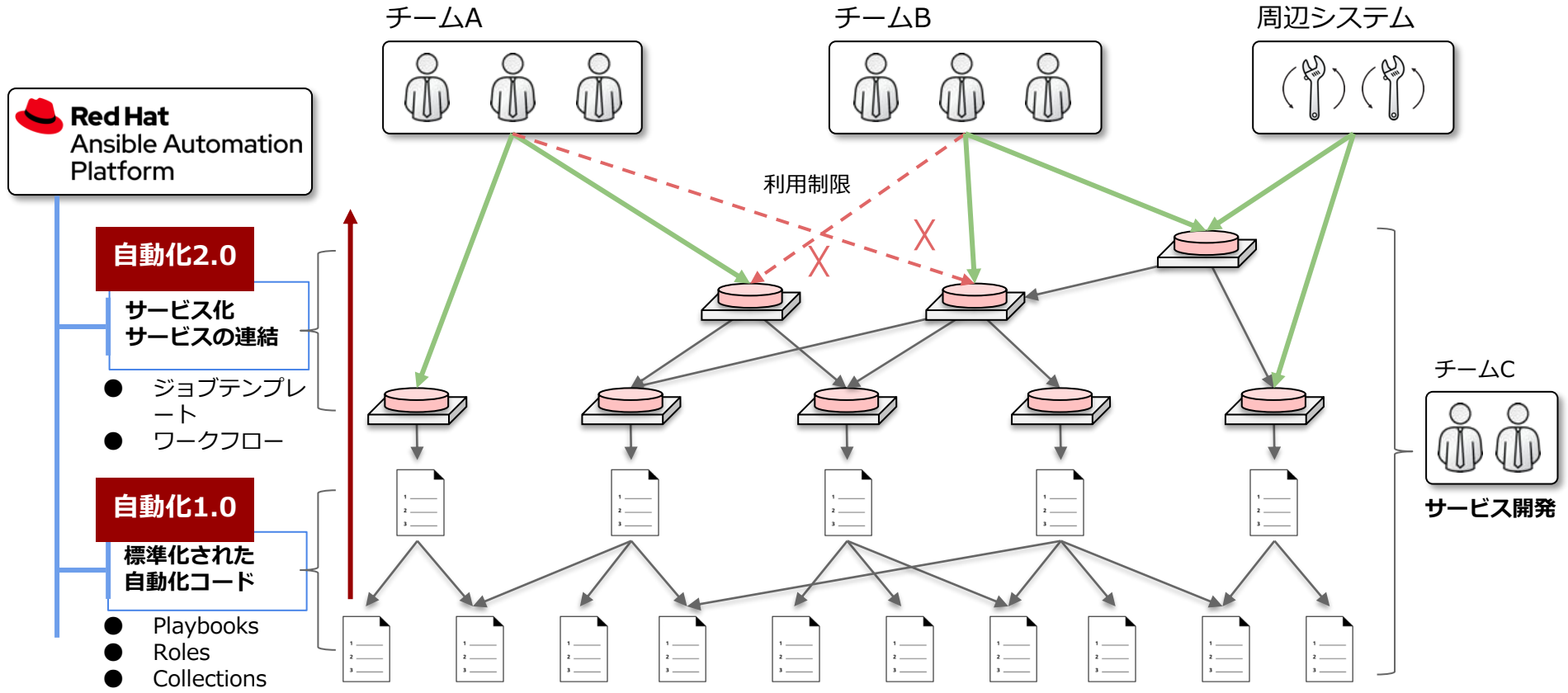


Ansible Automation Platform

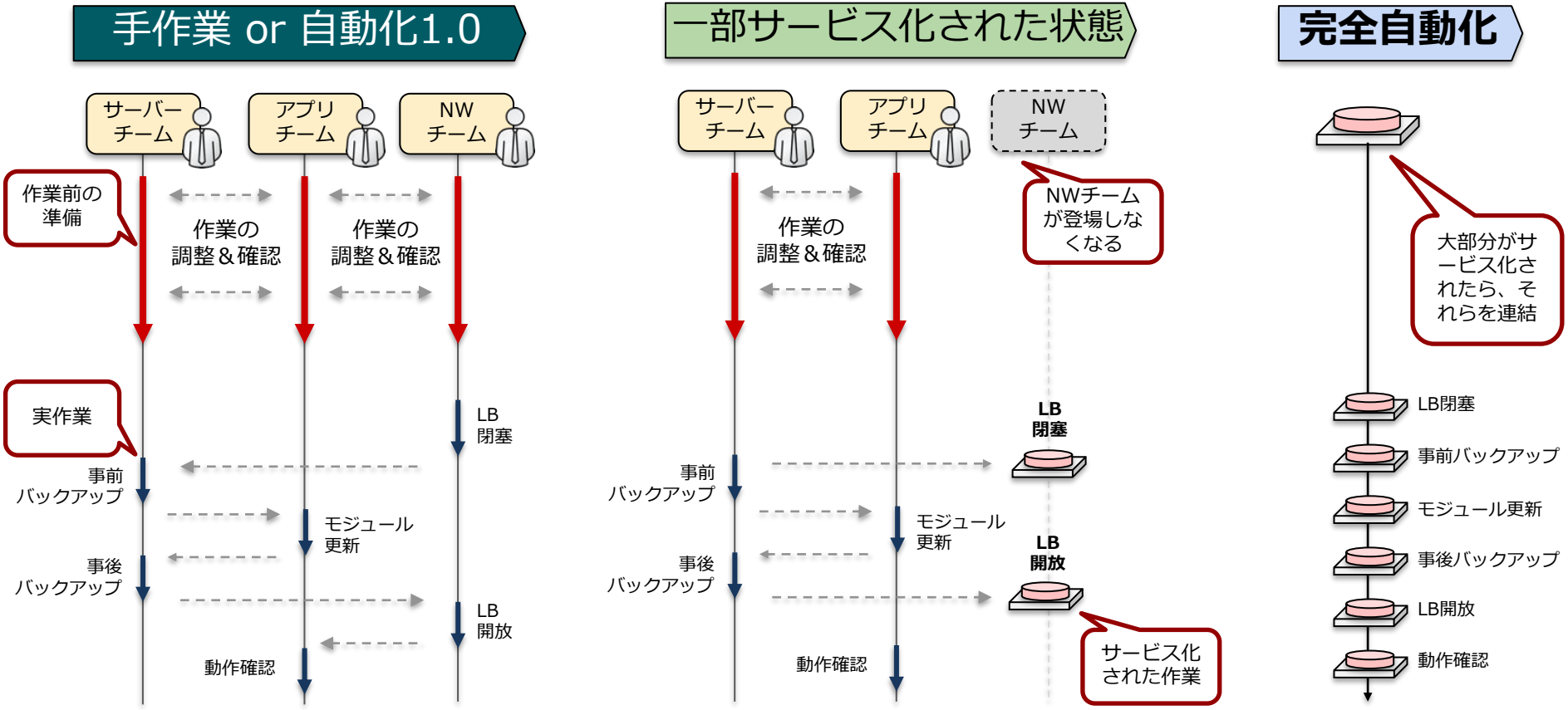
～ Ansible Tower 機能概要 ～

Red Hat

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



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

権限管理
作業実行者への権限の移譲

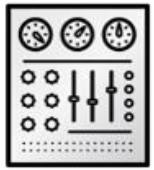
ワンクリック実行
ジョブ実行をワンクリックで開始

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、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

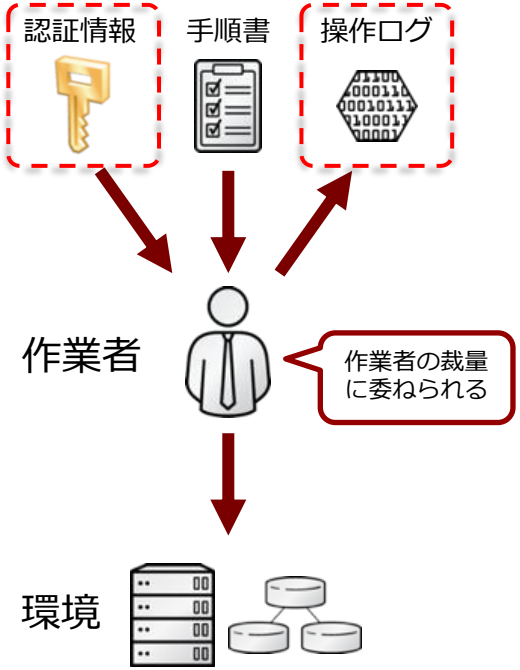
DELEGATION

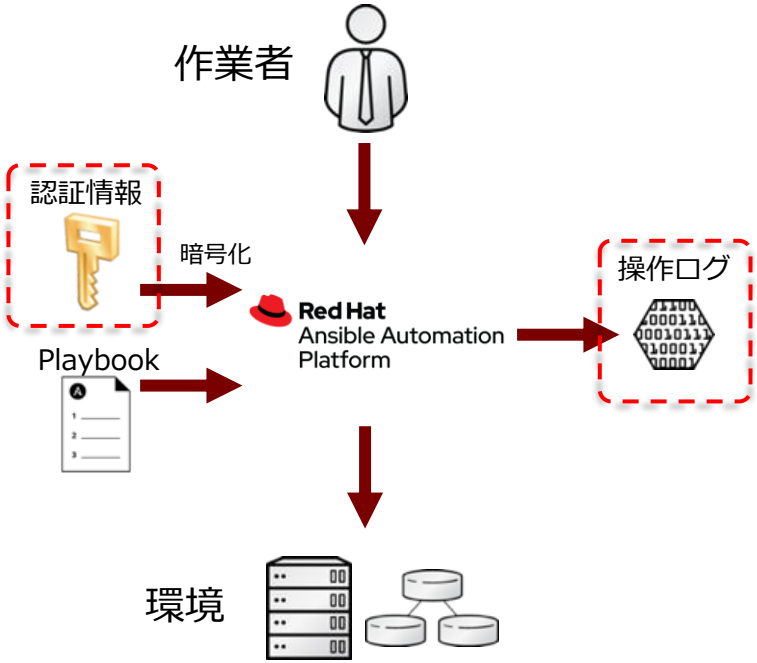
ロールベースの権限管理と
ユーザへのセルフサービス

- 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者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。

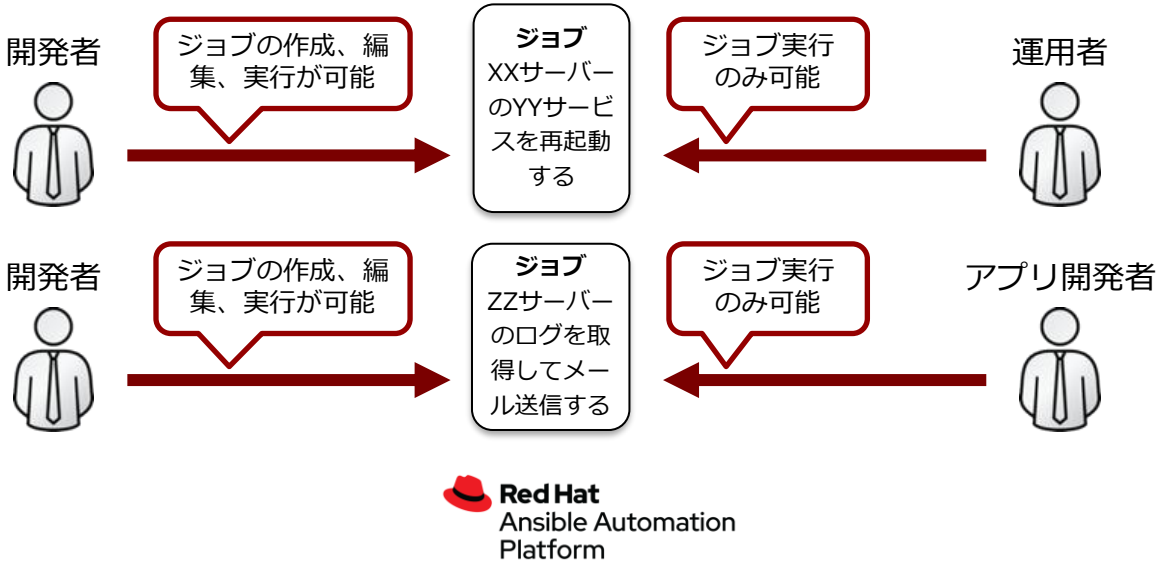




権限管理（RBAC）

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。

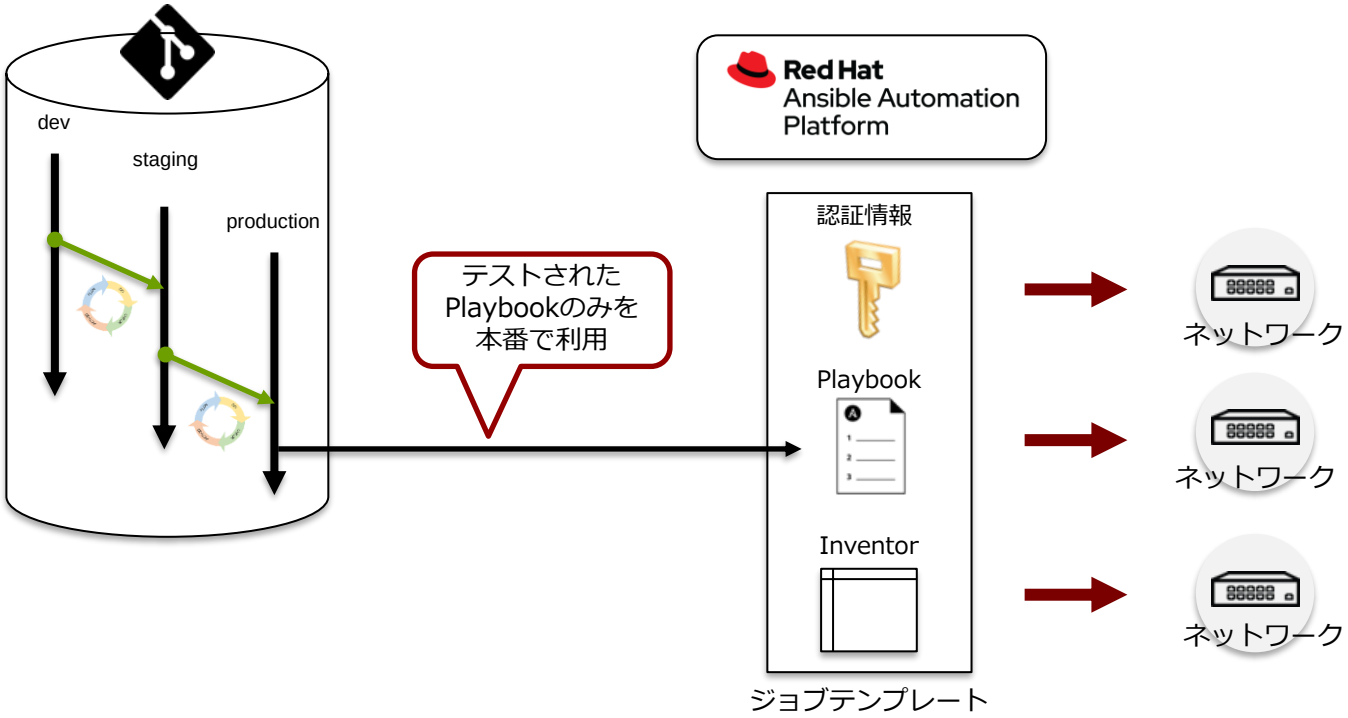


RBAC対象

- Playbook
- インベントリ
- 認証情報
- ジョブ

SCM連携

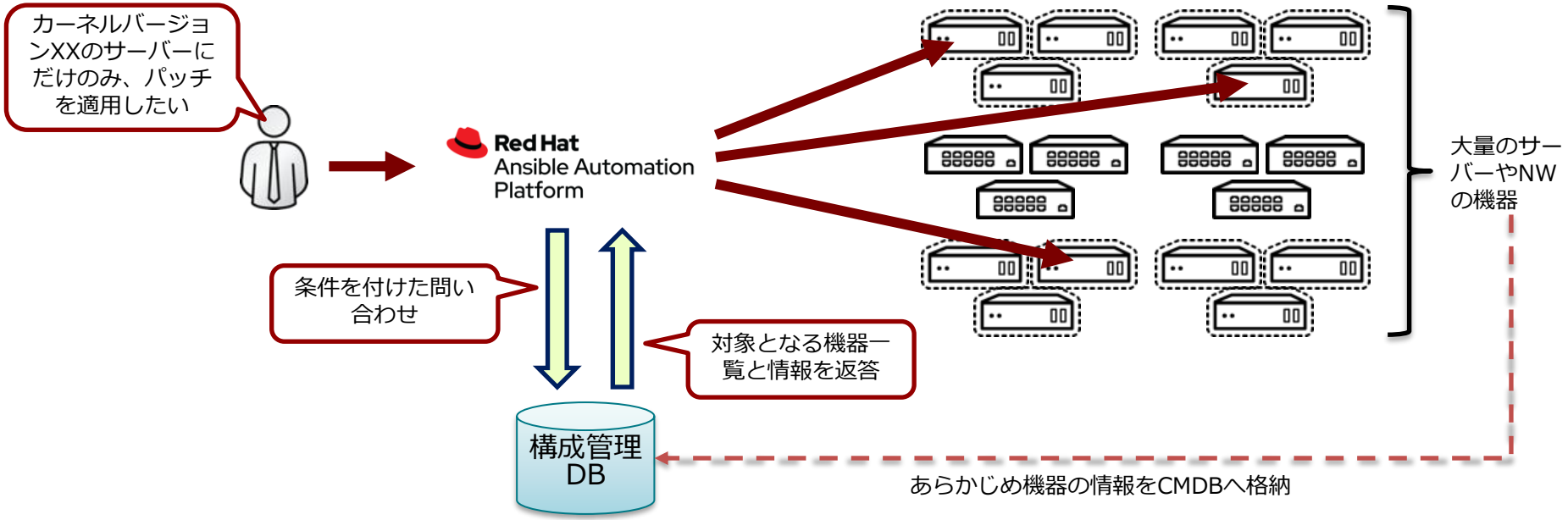
バージョン管理されていない野良Playbookや、CI等による品質管理の連携が可能。



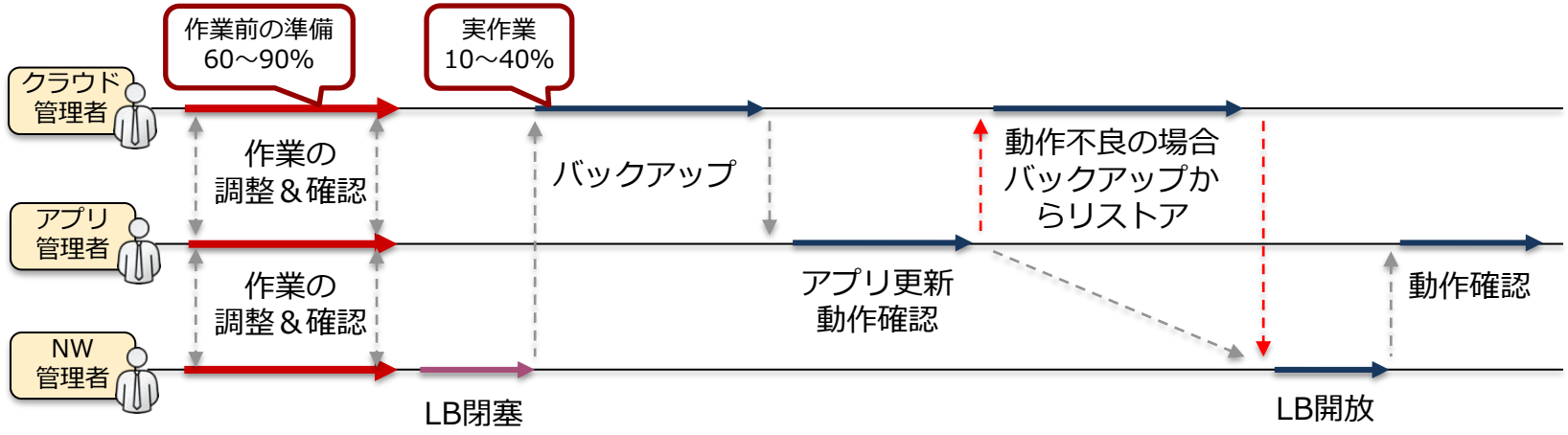
多数の自動化の対象を CMDB と連携して管理

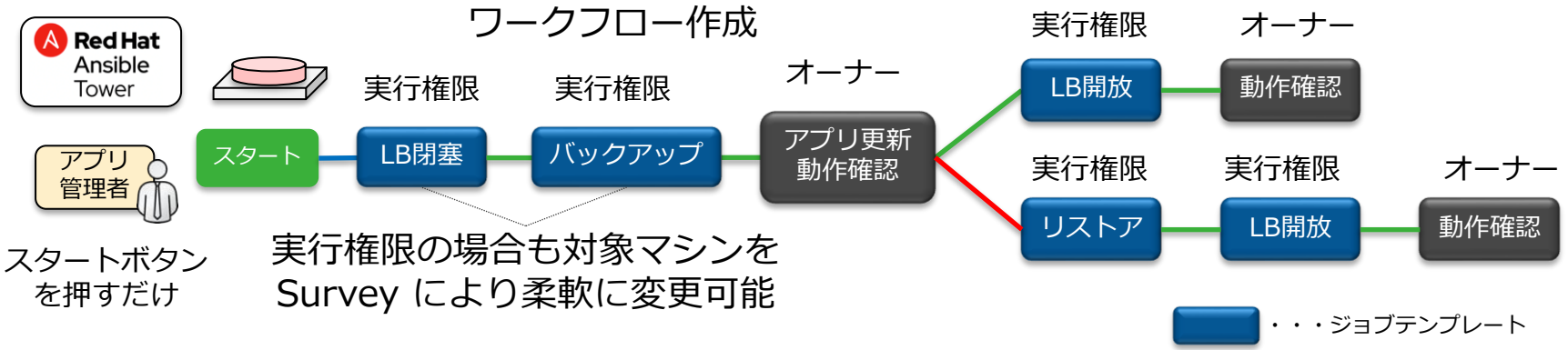
多数の機器の中から特定条件に合致する対象のみに自動化を「安全、確実」に実行する。

- 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）

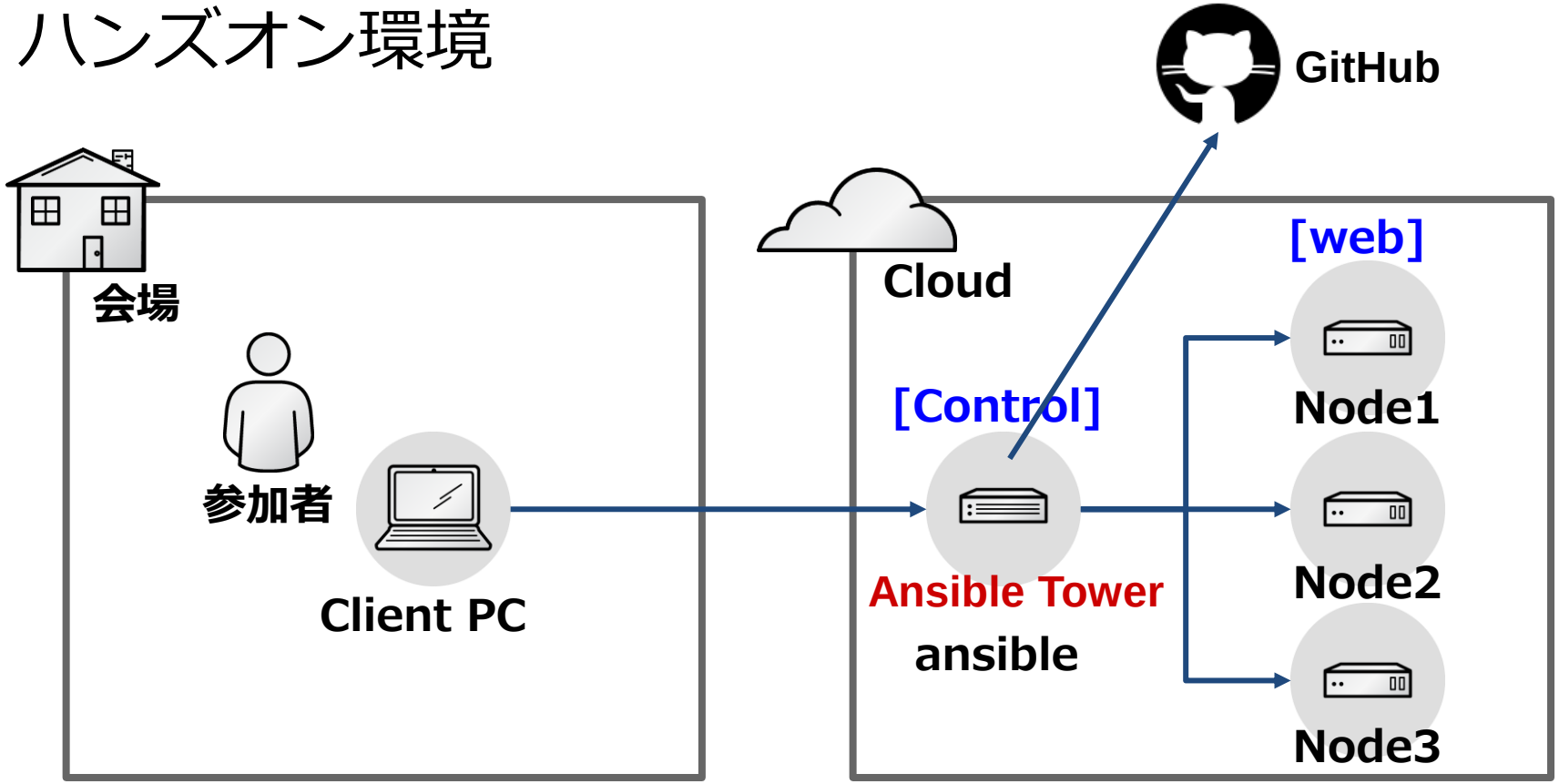




Ansible Tower Workshop

Red Hat

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

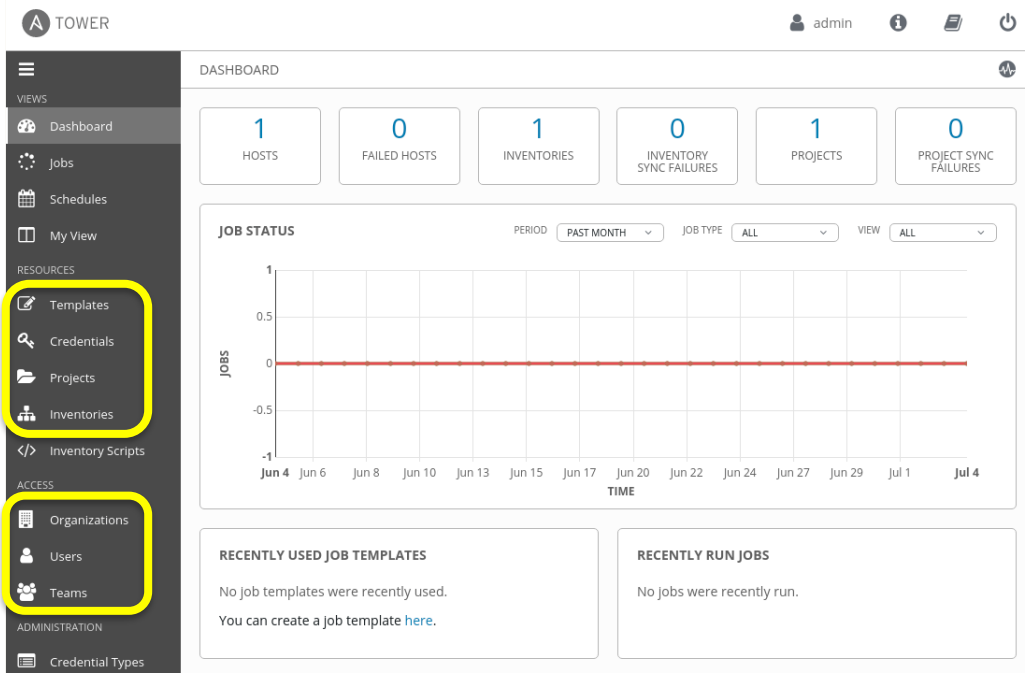
Section 2 - Ansible Tower

- 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

演習 2.1 - Tower の紹介

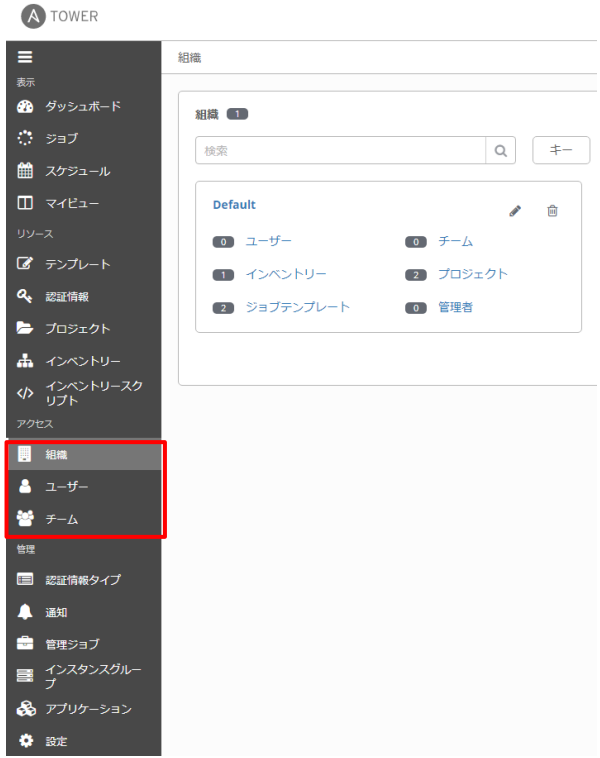
Ansible Tower には、Ansible にはない GUI ダッシュボードや、『プロジェクト』、『認証情報』など、独自のオブジェクトがあります。演習2.1 ではそれらについて学習します。



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

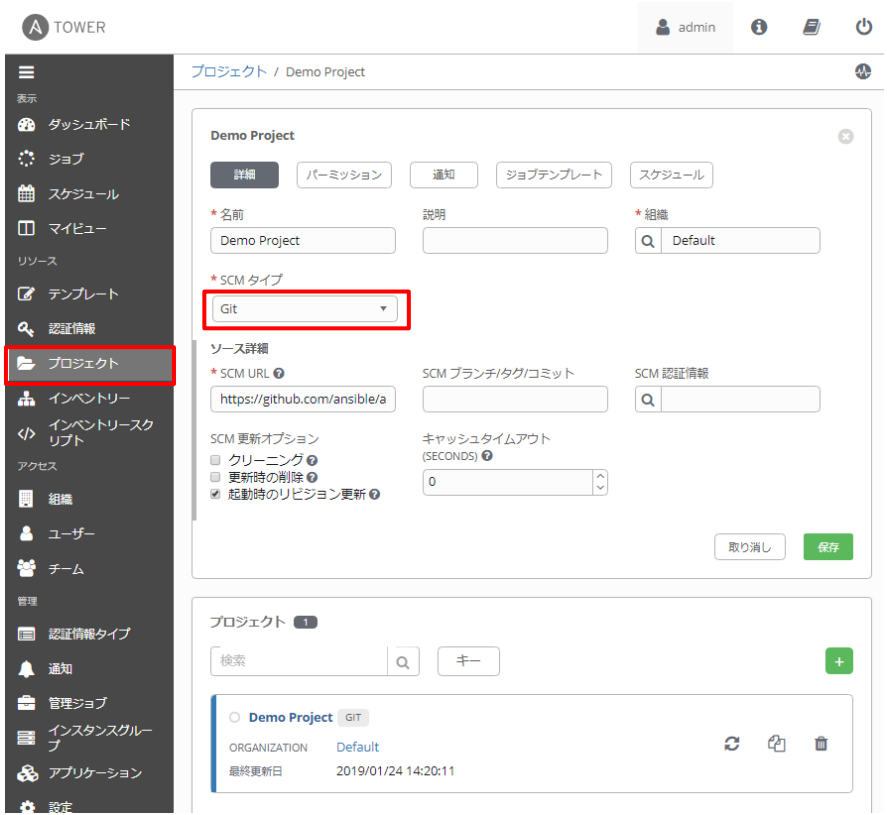
演習 2.1 Tower の紹介 - 組織



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。

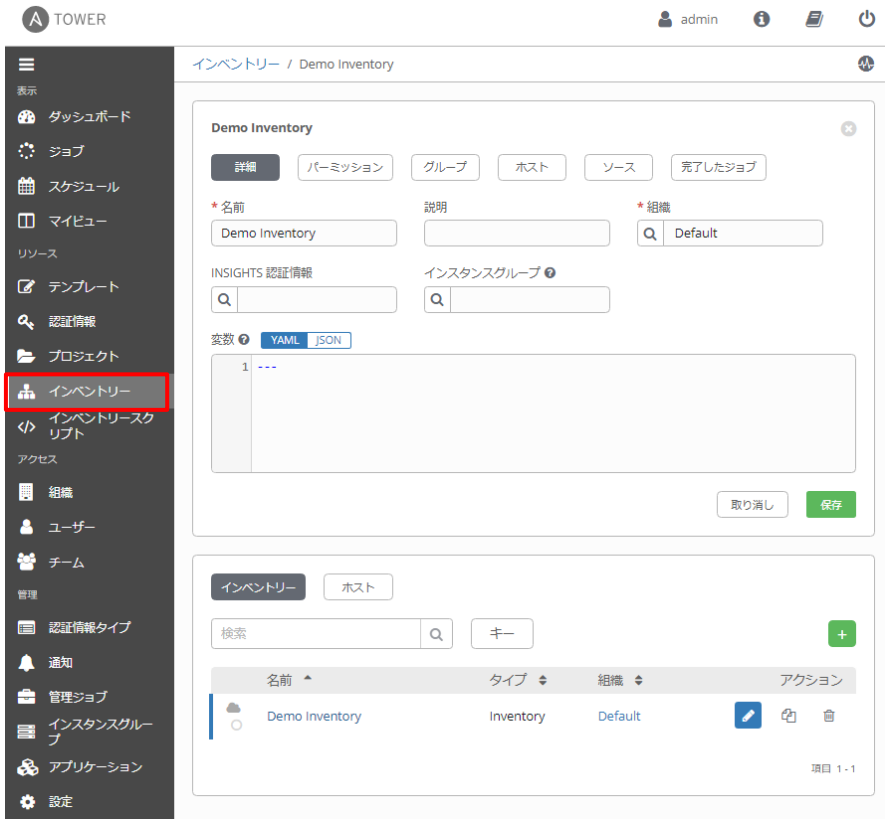
演習 2.1 Tower の紹介 - プロジェクト



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。

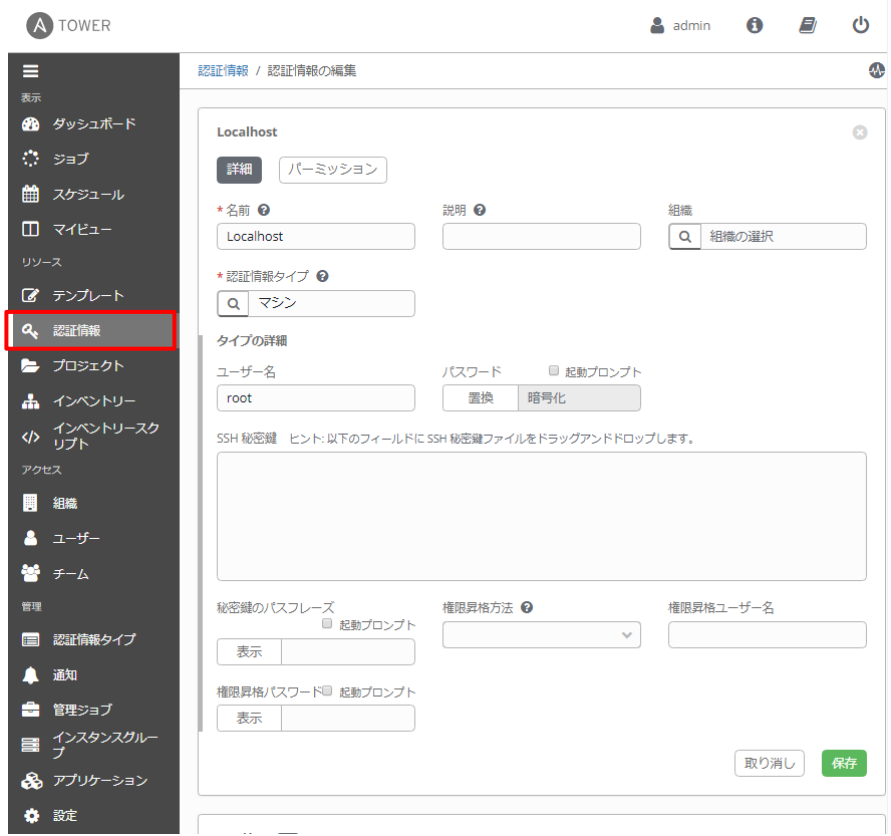
演習 2.1 Tower の紹介 - インベントリー



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。

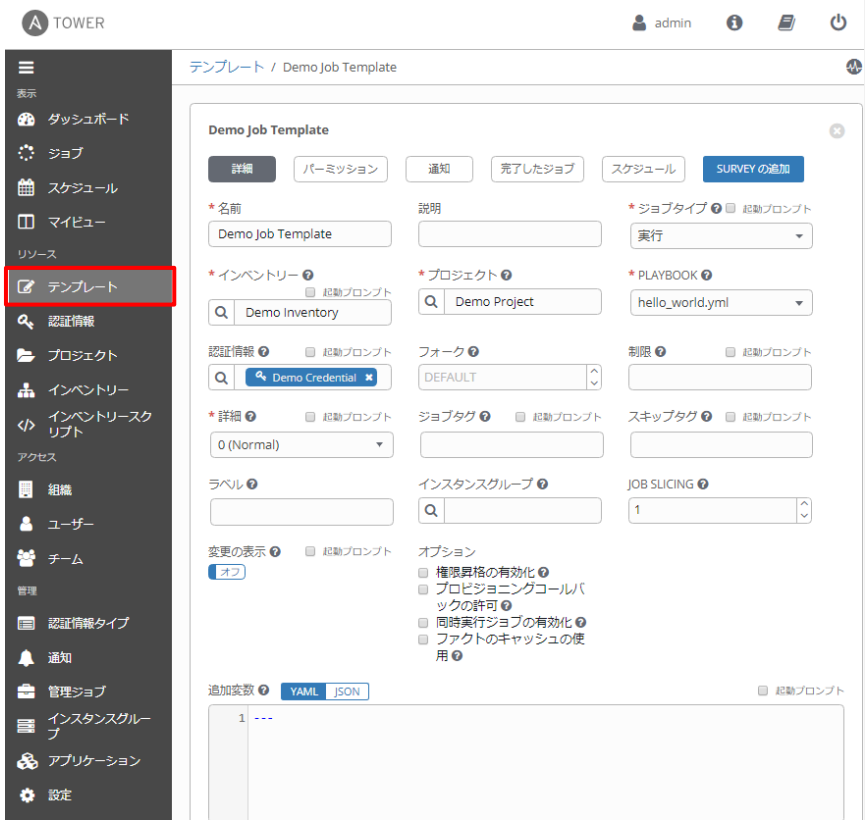
演習 2.1 Tower の紹介 - 認証情報



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。

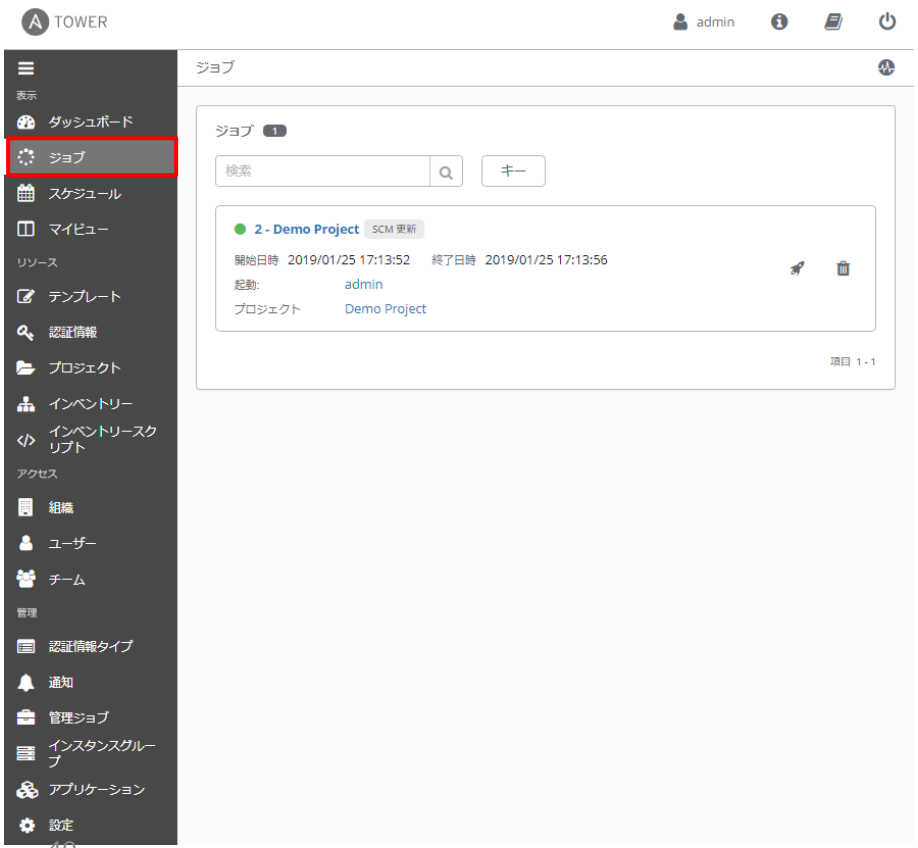
演習 2.1 Tower の紹介 - テンプレート



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

演習 2.1 Tower の紹介 - ジョブ



ジョブ（Job）

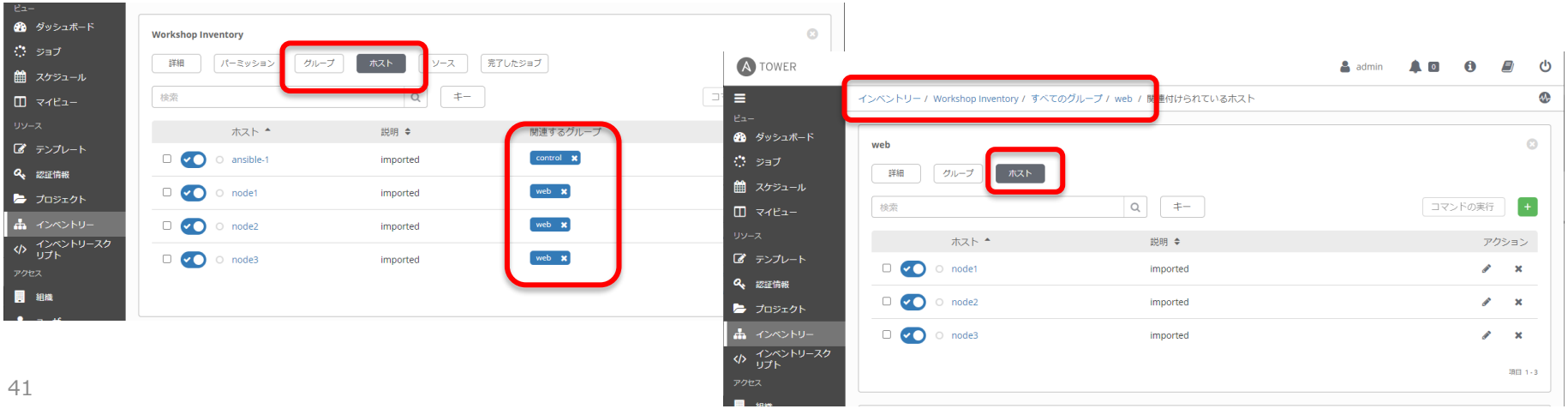
テンプレートの実行状況、実行結果（成功・失敗など）を確認することができます。プロジェクトでGitなどのSCMを指定した場合、実行結果にプレイブックのバージョン情報が含まれ、以下のような情報を過去にさかのぼり確認することができます。

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。



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。

演習 2.2 - インベントリー、認証情報、アドホックコマンド

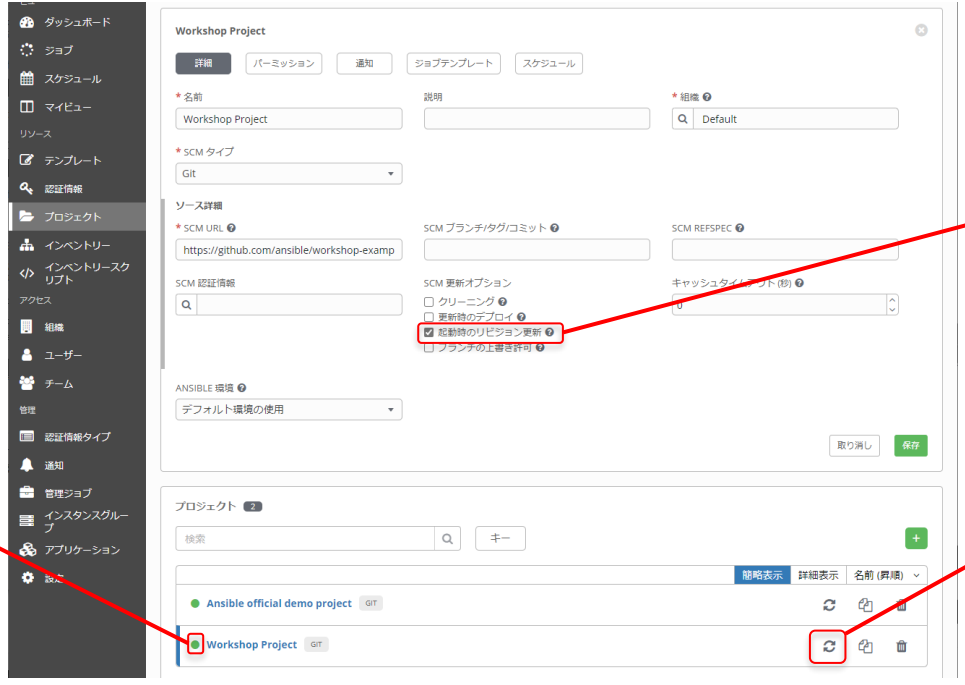
アドホックコマンド

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。



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

SCM 同期(Playbook ダウンロード) 済み

手動同期ボタン

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

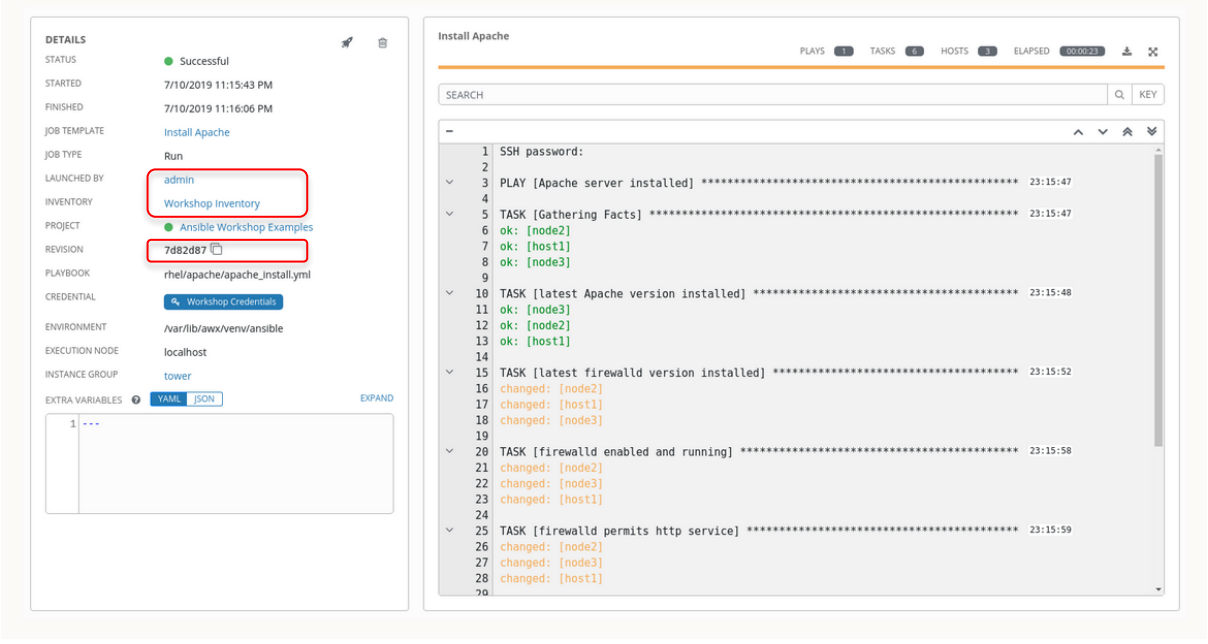
Playbook では "hosts : all" が指定されていますが、ジョブテンプレートの "制限" で web (グループ) が指定されていますので、実際に実行されるホストは、"web" に所属する3台となります

```
---
- 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



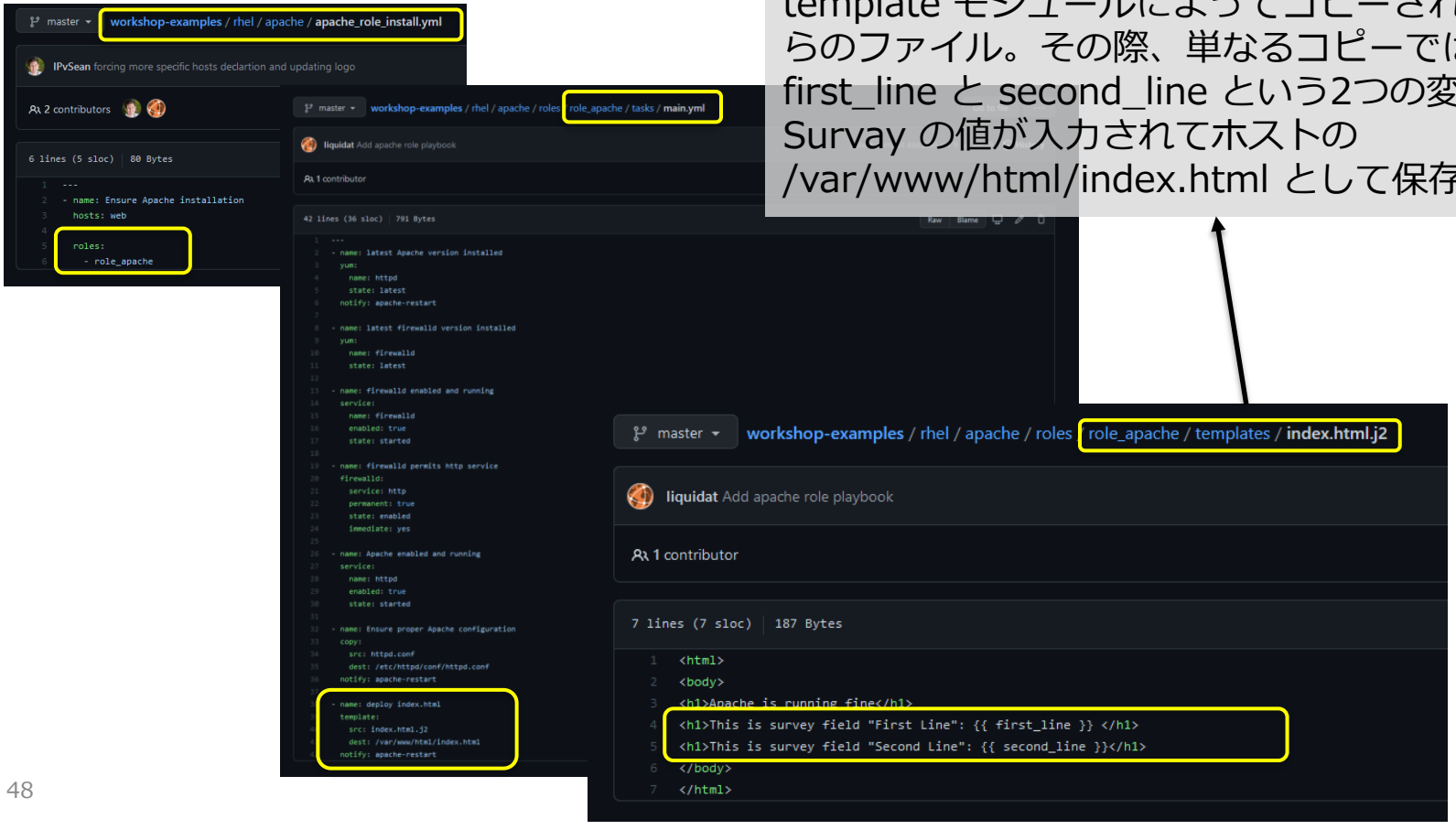
演習 2.4 - Survay

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 - Survay

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、Survay 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。

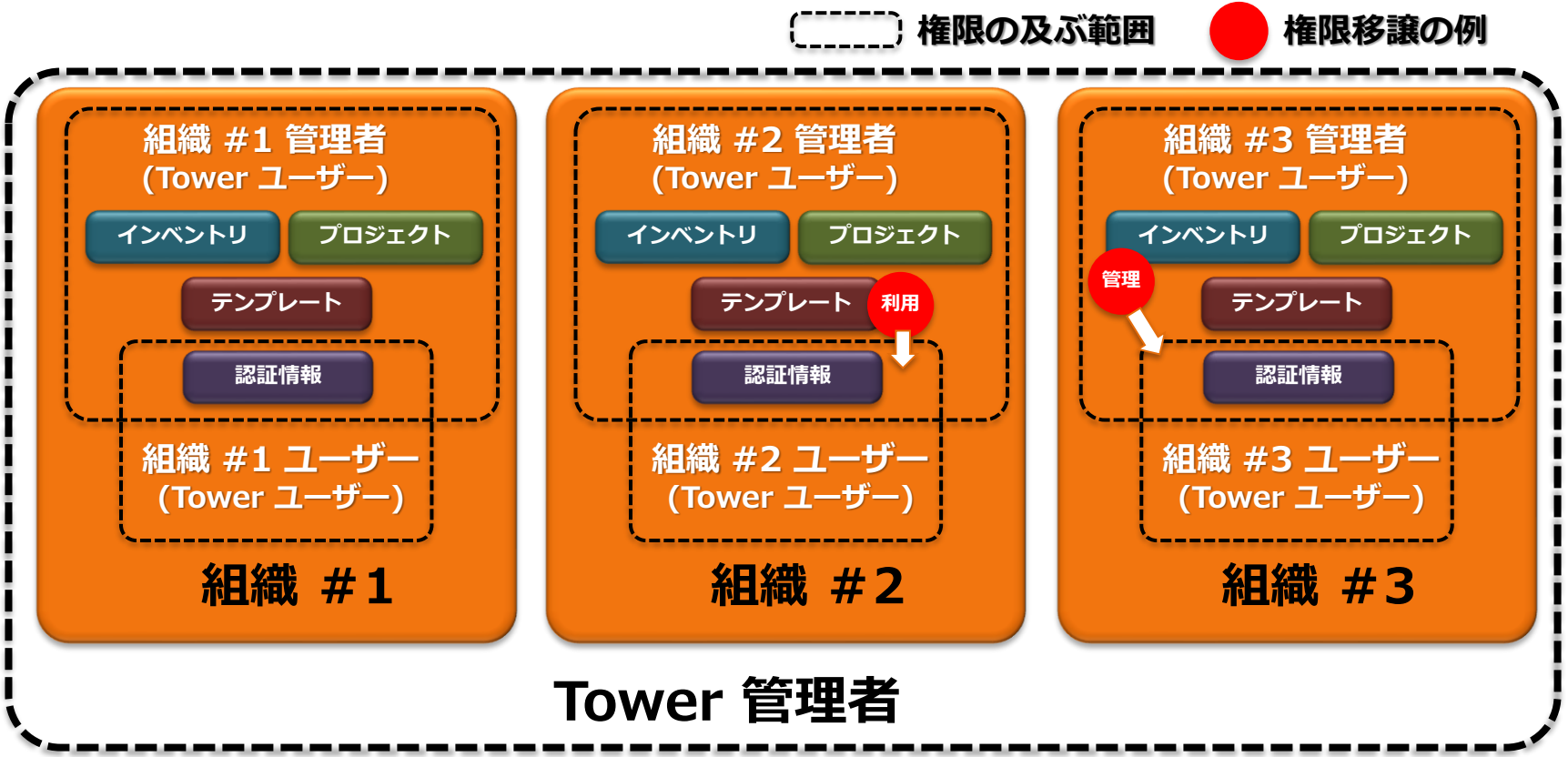
・新しいユーザー（wweb）を作成
・新しいチーム（web Content）の作成
　作成したチームに新しいユーザーを追加
　web Content チームにジョブテンプレート "Create Index.html" の "実行" 権限を付与
・wwebでAnsible Towerにログインし、ジョブテンプレート Create Index.html を実行

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、インベントリー、テンプレート、ホスト認証情報

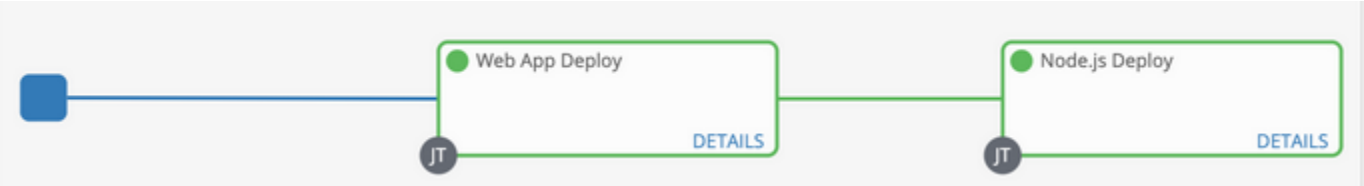
単一の組織の管理者は、Towerのユーザー + 組織の管理者権限 で設定します
例えば、以下の様な権限の設定が可能です
・A・・・Tower の管理者
・B・・・Tower ユーザー、組織 A 管理者
・C・・・Tower ユーザー、組織 A メンバー、組織 A 特定テンプレートの実行
・D・・・Tower ユーザー、組織 B 管理者、組織 A 特定インベントリの管理者

演習 2.5 - ロールベースのアクセス制御 Tips



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下さい。



演習 2.7 - まとめ

node1 & node3・・・開発用の Web Server

node2・・・本番用の Web Server

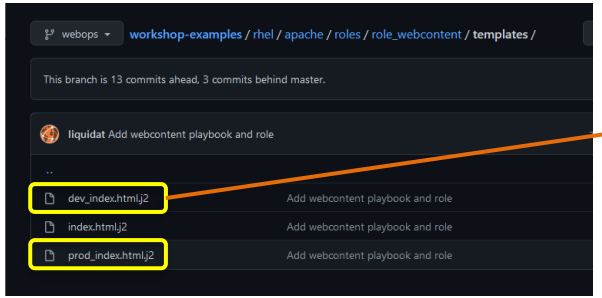
と定義し、デフォルトで表示される index.html 開発用、本番用のコメントが表示されるように変数を定義してみましょうという演習です。

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

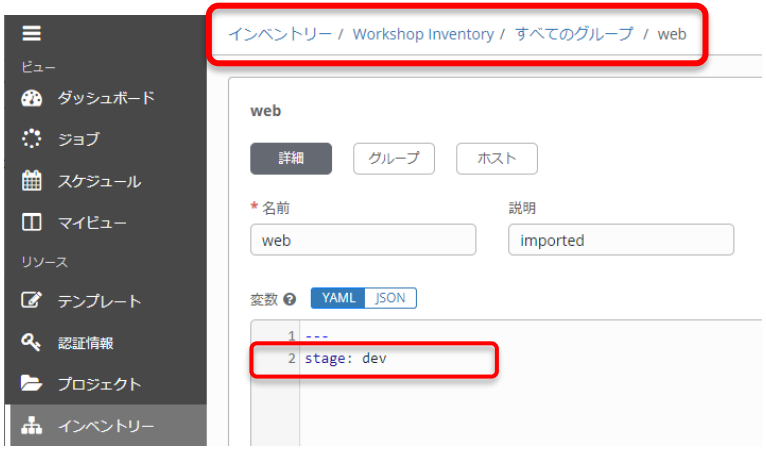
dev_content 変数があります。htmlの中身なので、開発用のWeb Serverであることを示す値を入力しましょう。

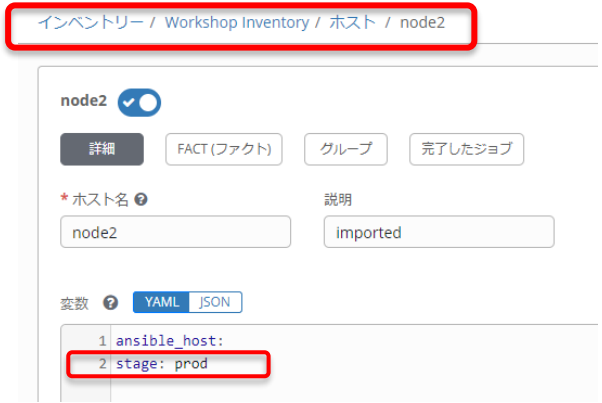
演習 2.7 - まとめ

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）



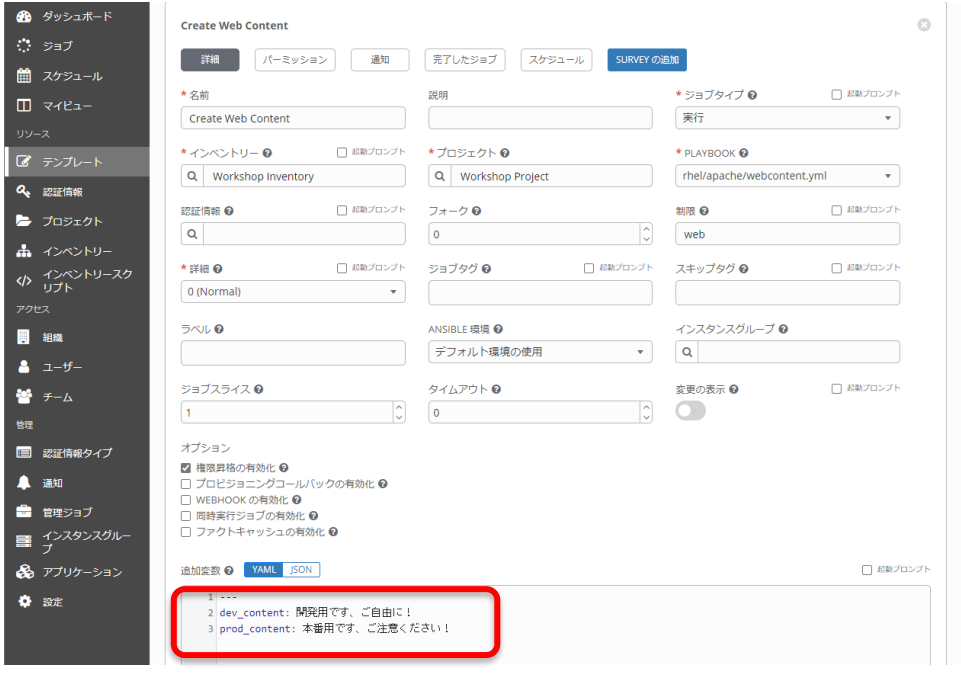


演習 2.7 - まとめ

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！



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.