

Ansibleハンズオンセミナー (RHEL編) 補足資料

レッドハット株式会社
テクニカルセールス本部
西日本ソリューションアーキテクト部

田中 耕輔 <ktanaka@redhat.com>

2019.9.2 / Rev 2.0

Section 1

Ansible Engine 演習

Section 1 - Ansible Engineの演習 - もくじ

演習 1.1 - 要件を確認してみよう

演習 1.2 - Ad-hoc コマンドを実行しよう

演習 1.3 - 初めての Playbook 作成

演習 1.4 - 変数を使ってみよう

演習 1.5 - 条件式、ハンドラ、ループを使う

演習 1.6 - テンプレートを使う

演習 1.7 - Roles: Playbook を再利用可能にする

演習 1.8 - ボーナストラボ

演習 1.1

環境確認のため、ハンズオン環境へ ログイン

- 各自に割り当てられたサーバーとユーザ名で
 - ssh studentXX@aa.bb.cc.dd
- sudo動作の確認
- ansibleのバージョン確認

演習 1.2

インベントリ(管理対象ホストの情報)の確認と、
Ad-Hocコマンド

- インベントリに登録されたホスト情報の確認
- Ansibleの設定ファイル ~/.ansible.cfg
- Ad-Hocコマンドの実行 - ワンライナーとしての使い方
 - `ansible -m ping`
 - `ansible -m command -a '...'`
 - `ansible -m copy -a '...'`
- 権限昇格
- マニュアルの引き方 ~ `ansible-doc`

演習 1.3

Playbookの基本的な書き方

- YAML形式のお約束:
 - 先頭行は '---'
 - インデントは正しく
- Playbookのお約束:
 - name:, hosts:, become:
 - tasks:
- Playbook実行 ~ ansible-playbook
- グループまとめて実行

```
---  
- name: Apache server installed  
  hosts: node1  
  become: yes  
  tasks:  
    - name: latest Apache version installed  
      yum:  
        name: httpd  
        state: latest  
    - name: Apache enabled and running  
      service:  
        name: httpd  
        enabled: true  
        state: started  
    - name: copy index.html  
      copy:  
        src: ~/ansible-files/index.html  
        dest: /var/www/html/
```

playbookの構文チェックもやってみましょう:

- `ansible-playbook --syntax-check apache.yml` (構文チェック)
- `ansible-playbook --check apache.yml` (dry-run)

演習 1.4

変数とAnsibleファクト

- 変数の定義
 - var
 - ホスト変数 `./host_vars/...`
 - グループ変数 `./group_vars/...`
- ファクトの確認
 - 対象ノードのプロパティ情報
 - 実行時に自動収集
 - `ansible_xxx`
- 変数、ファクトの利用
 - `{{ ... }}`

```
---  
stage: dev
```

```
---  
stage: prod
```

```
---  
- name: Copy index.html  
  hosts: web  
  become: yes  
  tasks:  
    - name: copy index.html  
      copy:  
        src: ~/ansible-files/{{ stage }}_index.html  
        dest: /var/www/html/index.html
```

演習 1.5

条件分岐、ハンドラ、ループ

- 条件分岐: when
- ハンドラ
 - notify と handlers
 - 後で実行したい処理
- くりかえし
 - loop
 - {{ item }}

```
---
- name: manage httpd.conf
  hosts: web
  become: yes
  tasks:
    - name: Copy Apache configuration file
      copy:
        src: httpd.conf
        dest: /etc/httpd/conf/
      notify:
        - restart_apache
  handlers:
    - name: restart_apache
      service:
        name: httpd
        state: restarted
```

```
---
- name: Ensure users
  hosts: node1
  become: yes

  tasks:
    - name: Ensure three users are present
      user:
        name: "{{ item.username }}"
        state: present
        groups: "{{ item.groups }}"
      loop:
        - { username: 'dev_user', groups: 'ftp' }
        - { username: 'qa_user', groups: 'ftp' }
        - { username: 'prod_user', groups: 'apache' }
```


演習 1.6

テンプレートの使用

- xxxxx.j2 ファイルを用意
 - Jinja2 テンプレートエンジン
- template モジュール
 - src:
 - dest:

```
---  
- name: Fill motd file with host data  
  hosts: node1  
  become: yes  
  tasks:  
    - template:  
      src: motd-facts.j2  
      dest: /etc/motd  
      owner: root  
      group: root  
      mode: 0644
```

```
Welcome to {{ ansible_hostname }}.  
{{ ansible_distribution }} {{ ansible_distribution_version }}  
deployed on {{ ansible_architecture }} architecture.
```

演習 1.7

Roles による Playbookの再利用

- roles ディレクトリ
- ansible-galaxy init roles/xxx
 - スケルトン作製
- これまでの演習で作ったplaybookをroleとして再実装
 - バラバラに配置し直すイメージ

```
apache/  
├─ defaults  
│   └─ main.yml  
├─ files  
├─ handlers  
│   └─ main.yml  
├─ meta  
│   └─ main.yml  
├─ README.md  
├─ tasks  
│   └─ main.yml  
├─ templates  
├─ tests  
│   ├── inventory  
│   └─ test.yml  
└─ vars  
    └─ main.yml
```

演習 1.8

ボーナスラボ: 腕試し問題

- Ad-Hocコマンド実行
- テンプレートとホスト変数/グループ変数を使い、
httpd.confを配置
するplaybook

```
---  
- name: Apache httpd.conf  
  hosts: web  
  become: yes  
  tasks:  
    - name: Create Apache configuration file from template  
      template:  
        src: httpd.conf.j2  
        dest: /etc/httpd/conf/httpd.conf  
      notify:  
        - restart apache  
  handlers:  
    - name: restart apache  
      service:  
        name: httpd  
        state: restarted
```

Section 2

Ansible Tower演習

※Engineの演習が終わってない方も
ご一緒にお願いします

Section 2 - Ansible Towerの演習 - もくじ

演習 2.1 - Tower の紹介

演習 2.2 - インベントリー、認証情報、アドホックコマンド

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

演習 2.1

- Ansible Towerの価値を理解する
 - ユーザフレンドリーな Web GUI
 - ユーザアクセス権限による安全な運用
 - REST API
 - etc.
- ラボ環境を(再)確認
- ログイン→ダッシュボードを見る
- 用語・概念：
 - プロジェクト (≡ リポジトリ)
 - インベントリー
 - 認証情報 (Credentials)
 - テンプレート
 - ジョブ

演習 2.2

Ansible Towerを利用する準備

- インベントリ(Inventory)の作成
 - 手作業での作製
 - コマンドでimportも可能

```
$ sudo awx-manage inventory_import \
--source="/home/student1/lab_inventory/hosts" --inventory-name="test"
```
- 認証情報(Credential)の作成
 - node1 ~ node3へのアクセス情報
- Tower上でのAd-Hocコマンド実行
 - 権限昇格

演習 2.3

Ansible Tower 上での Playbook 実行

- プロジェクトの作製
 - github リポジトリの確認
 - プロジェクト作製
- ジョブテンプレート作製 ~ これまでに準備したものを指定
 - インベントリー
 - プロジェクト
 - playbook
 - 認証情報、権限昇格指定
- ジョブテンプレートの実行 → ジョブ
- 結果の確認と、次作業の準備
 - node1 (Webserver) への apache httpdインストール

演習 2.4

サーベイ(Survey)機能の利用

- role_apache ロールの確認:
 - httpdの導入 → httpd.conf のコピー → index.htmlの作製
 - index.html.j2 の確認
- サーベイ付きのジョブテンプレートの作製
 - ジョブテンプレートの作製
 - 「Surveyの追加」
 - {{ first_line }}, {{ second_line }} の入力
- 作ったジョブテンプレートの実行
- インベントリー(Webserver)を変更して、ジョブテンプレートの再実行

演習 2.5

ロールベースのアクセス制御 (RBAC)

- ユーザーを作製
 - wweb / Normal User
- チームの作製
 - Web Content / wwebをメンバーとして追加
- ジョブテンプレートへのパーミッション付与
 - Web Contentチームに対して、
Create index.html の実行権限を追加
- パーミッションの確認
 - wwebユーザーとしてTowerへログイン
 - どのジョブテンプレートが見えるか？

演習 2.6

ワークフロー機能

- プロジェクトを2つ作製
 - Webops Git Repo
 - Webdev Git Repo
- ジョブテンプレートを2つ作製
 - Tomcat Deploy - rhel/webops/tomcat.yml
 - Web App Deploy - rhel/webdev/create_jsp.yml
- ワークフローテンプレートの作製:
 - Deploy Webapp Server
 - Tomcat Deploy → Web App Deploy
- ワークフローの実行

演習 2.7

チャレンジ問題

- インベントリーの作製: Webserver
 - グループ変数、ホスト変数の設定
 - Webserver → stage: dev
 - node2 → stage: prod
- ジョブテンプレートの作製: Create Web Content
 - rhel/apache/webcontent.yml
 - 追加変数:
 - dev_content: default dev content
 - prod_content: default prod content
- サーベイの追加 → dev_content, prod_contentの設定
- wwebユーザーによる実行

おつかれさまでした！