

# GitHub Actionsで Ansible roleのCIを 実行する

2020/6/23 『Ansible Night オンライン！ 2020.06』

 @ma2muratomonori

# 自己紹介

@ma2muratomonori

- 週3～4日保線工事に従事
- 週3日開発
- Ansible歴5年目



# 本日お話しする事

- なぜCIが必要か？
- GitHub Actionsとは？
- サンプルコード
- 次のステップ

なぜCIが  
必要か？

# なぜCIが必要か？

- Ansibleのバージョンアップ
- 実行対象環境の変化
- Galaxy role、Collectionsのアップデート
- コードの変化

# GitHub Actions とは？

# GitHub Actionsとは？

- GitHubリポジトリと直接連携
- 実行環境にLinux、Windows、MacOSが指定可能
- Actionsライブラリを用いて処理の共通化が可能
- パラメータを指定して並列にCIを実行出来る
- 公開リポジトリでは無料で利用できる

サンプル  
コード

# CentOS7へcowsayをインストールする

- Ansibleのバージョン
  - 2.9.10
  - 2.10.0b1
- Pythonのバージョン
  - 2.7
  - 3.6
  - 3.7
  - 3.8

# パイプラインの流れ

1. バージョン違いのPythonをインストール
2. バージョン違いのAnsibleをインストール
3. sudoがインストールされたCentOS7イメージをビルド
4. コンテナを起動
5. Galaxyの依存関係をインストール
6. RoleをPlaybookとして実行
7. 幕等性のチェック
8. Playbook実行後のコマンドテスト

```
.github
└── workflows
    └── playbook.yml
── README.md
── defaults
    └── main.yml
── meta
    └── main.yml
── tasks
    └── main.yml
── tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
── test.yml
```

# ファイル構成

```
.github
└── workflows
    └── playbook.yml
── README.md
── defaults
└── main.yml
── meta
└── main.yml
── tasks
└── main.yml
── tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
    └── test.yml
```

**.github/workflows/playbook.yml**

GitHub Actionsの実行ファイル

```
.github
└── workflows
    └── playbook.yml
-- README.md
-- defaults
    └── main.yml
-- meta
    └── main.yml
-- tasks
    └── main.yml
-- tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
-- test.yml
```

Ansible roleのファイル

```
.github
└── workflows
    └── playbook.yml
-- README.md
-- defaults
    └── main.yml
-- meta
    └── main.yml
-- tasks
    └── main.yml
-- tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
    └── test.yml
```

## tasks/main.yml

cowsayをインストール

```
---
- name: Install cowsay with yum
  yum:
    name: cowsay
    state: present
```

```
.github
└── workflows
    └── playbook.yml
── README.md
── defaults
    └── main.yml
── meta
    └── main.yml
── tasks
    └── main.yml
── tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
    └── test.yml
```

## **tests/Dockerfile**

Ansibleを実行するコンテナを作成

```
.github
└── workflows
    └── playbook.yml
── README.md
── defaults
    └── main.yml
── meta
    └── main.yml
── tasks
    └── main.yml
── tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
── test.yml
```

## **tests/roles/requirements.yml**

Galaxy roles、collectionsをインストール

```
---
roles:
- name: geerlingguy.repo-epel
- name: cowsay
  src: https://github.com/..
collections:
- name: community.general
```

```
.github
└── workflows
    └── playbook.yml
── README.md
── defaults
    └── main.yml
── meta
    └── main.yml
── tasks
    └── main.yml
── tests
    ├── Dockerfile
    └── roles
        └── requirements.yml
    └── test.yml
```

## **tests/test.yml**

Roleのテストの実行ファイル

```
---
- hosts: all
  become: true
  roles:
    - gearlingguy.repo-epel
    - cowsay
```

# tests/Dockerfile

- イメージを最新版へアップデート
- sudoをインストール
- user 「ansible」を追加



## tests/Dockerfile

```
FROM centos:7
```

```
# Install requirements
```

```
RUN yum clean all \  
    && yum update -y \  
    && yum install -y sudo
```

```
# Create `ansible` user with sudo permissions
```

```
ENV ANSIBLE_USER=ansible SUDO_GROUP=wheel
```

```
RUN set -xe \  
    && groupadd -r ${ANSIBLE_USER} \  
    && useradd -m -g ${ANSIBLE_USER} ${ANSIBLE_USER} \  
    && usermod -aG ${SUDO_GROUP} ${ANSIBLE_USER} \  
    && sed -i "/^%${SUDO_GROUP}/s/ALL\$/NOPASSWD:ALL/g" /etc/sudoers
```

イメージを最新版へアップデート

## tests/Dockerfile

```
FROM centos:7
```

### # Install requirements

```
RUN yum clean all \  
    && yum update -y \  
    && yum install -y sudo
```

### # Create `ansible` user with sudo permissions

```
ENV ANSIBLE_USER=ansible SUDO_GROUP=wheel  
RUN set -xe \  
    && groupadd -r ${ANSIBLE_USER} \  
    && useradd -m -g ${ANSIBLE_USER} ${ANSIBLE_USER} \  
    && usermod -aG ${SUDO_GROUP} ${ANSIBLE_USER} \  
    && sed -i "/^%${SUDO_GROUP}/s/ALL\$/NOPASSWD:ALL/g" /etc/sudoers
```

SUDOをインストール

## tests/Dockerfile

```
FROM centos:7
```

```
# Install requirements
```

```
RUN yum clean all \  
    && yum update -y \  
    && yum install -y sudo
```

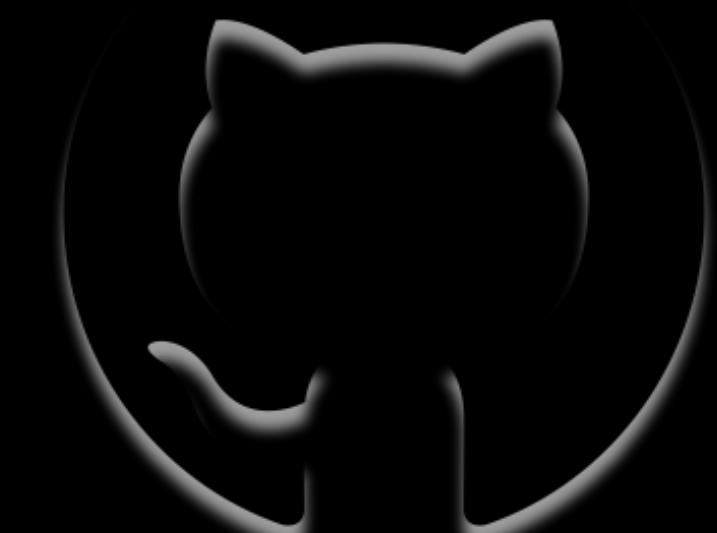
```
# Create `ansible` user with sudo permissions
```

```
ENV ANSIBLE_USER=ansible SUDO_GROUP=wheel  
RUN set -xe \  
    && groupadd -r ${ANSIBLE_USER} \  
    && useradd -m -g ${ANSIBLE_USER} ${ANSIBLE_USER} \  
    && usermod -aG ${SUDO_GROUP} ${ANSIBLE_USER} \  
    && sed -i "/^%${SUDO_GROUP}/s/ALL\$/NOPASSWD:ALL/g" /etc/sudoers
```

user 「ansible」 を追加、 権限を付与

# .github/workflow/playbook.yml

1. 異なるバージョンのPythonをインストール
2. 異なるバージョンのAnsibleをインストール
3. コードをチェックアウト
4. DockerイメージをDockerfileからビルド
5. Dockerコンテナを起動
6. Galaxy roleをインストール
7. Playbookを実行
8. 幕等性のチェック
9. コマンドのチェック



## .github/workflows/playbook.yml

```
name: playbook
on: [deployment, push]

jobs:
  playbook:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version:
          - 2.7
          - 3.6
          - 3.7
          - 3.8
        ansible-version:
          - 'ansible==2.9.10'
          - 'ansible-base'
```

Trigger Typeに

- deployment

- push

を指定

## .github/workflows/playbook.yml

```
---
```

```
name: playbook
on: [deployment, push]
```

```
jobs:
  playbook:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version:
          - 2.7
          - 3.6
          - 3.7
          - 3.8
        ansible-version:
          - 'ansible==2.9.10'
          - 'ansible-base'
```

**Ansibleを実行する  
Pythonのversionを  
複数指定**

## .github/workflows/playbook.yml

```
---
name: playbook
on: [deployment, push]

jobs:
  playbook:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version:
          - 2.7
          - 3.6
          - 3.7
          - 3.8
        ansible-version:
          - 'ansible==2.9.10'
          - 'ansible-base'
```

インストールする  
Ansibleのversionを  
複数指定

## .github/workflows/playbook.yml

```
---
name: playbook
on: [deployment, push]

jobs:
  playbook:
    runs-on: ubuntu-latest
    strategy:
      matrix:
        python-version:
          - 2.7
          - 3.6
          - 3.7
          - 3.8
        ansible-version:
          - 'ansible==2.9.10'
          - 'ansible-base'
```

**Matrixのパラメータの  
数だけVMが起動する  
 $(4 \times 2 = 8)$**

## .github/workflows/playbook.yml

```
steps:
  - name: Show Ansible version
    run: |
      ansible --version

  - name: Set up Python
    uses: actions/setup-python@v1
    with:
      python-version: ${{ matrix.python-version }}

  - name: Install Ansible
    run: |
      pip install ${{ matrix.ansible-version }}
```

異なるバージョンのPythonを複数指定しVMを起動

## .github/workflows/playbook.yml

```
steps:
  - name: Show Ansible version
    run: |
      ansible --version

  - name: Set up Python
    uses: actions/setup-python@v1
    with:
      python-version: ${{ matrix.python-version }}

  - name: Install Ansible
    run: |
      pip install ${{ matrix.ansible-version }}
```

Ansible 2.9.10、ansible-baseをインストール

## .github/workflows/playbook.yml

```
- name: Checkout
  uses: actions/checkout@v2

- name: Build CentOS7 image sudo installed
  run: docker build tests/. -t centos7

- name: Docker run CentOS7 image sudo installed
  run: docker run -d -t --name=centos7 centos7

- name: Install Ansible galaxy roles
  run: |
    ansible-galaxy install \
    -r tests/roles/requirements.yml
```

Dockerイメージをビルド

## .github/workflows/playbook.yml

```
- name: Checkout
  uses: actions/checkout@v2

- name: Build CentOS7 image sudo installed
  run: docker build tests/. -t centos7

- name: Docker run CentOS7 image sudo installed
  run: docker run -d -t --name=centos7 centos7

- name: Install Ansible galaxy roles
  run: |
    ansible-galaxy install \
    -r tests/roles/requirements.yml
```

Dockerコンテナを起動

## .github/workflows/playbook.yml

```
- name: Checkout
  uses: actions/checkout@v2

- name: Build CentOS7 image sudo installed
  run: docker build tests/. -t centos7

- name: Docker run CentOS7 image sudo installed
  run: docker run -d -t --name=centos7 centos7

- name: Install Ansible galaxy roles
  run: |
    ansible-galaxy install \
      -r tests/roles/requirements.yml
```

依存関係をインストール

## .github/workflows/playbook.yml

```
- name: Execute ansible-playbook command
  env:
    ANSIBLE_FORCE_COLOR: '1'
  run: |
    ansible-playbook -i 'centos7,' \
      -c docker -u ansible tests/test.yml
```

```
- name: Check idempotency
  env:
    ANSIBLE_FORCE_COLOR: '1'
  run: |
    ansible-playbook -i 'centos7,' \
      -c docker -u ansible tests/test.yml
```

# Playbookを実行

## .github/workflows/playbook.yml

```
- name: Execute ansible-playbook command
  env:
    ANSIBLE_FORCE_COLOR: '1'
  run: |
    ansible-playbook -i 'centos7,' \
      -c docker -u ansible tests/test.yml
```

```
- name: Check idempotency
  env:
    ANSIBLE_FORCE_COLOR: '1'
  run: |
    ansible-playbook -i 'centos7,' \
      -c docker -u ansible tests/test.yml
```

冪等性をチェック

## .github/workflows/playbook.yml

```
- name: Check idempotency
  env:
    ANSIBLE_FORCE_COLOR: '1'
  run: |
    ansible-playbook -i 'centos7,' \
      -c docker -u ansible tests/test.yml
```

```
- name: Make sure the cowsay command works
  run: |
    docker exec --tty centos7 env TERM=xterm \
      cowsay hello!
```

コマンドのチェック

次の  
ステップ

## 次のステップ

- 1.Moleculeを採用する
- 2.Playbook実行対象コンテナの自動更新
- 3.Ansible-lint、yamllintの適用
- 4.CIの定期実行
- 5.複数のOSへの対応

# 参考資料

- [\*\*TomonoriMatsumura/ansible-night-2020-06: Ansible Night 2020年6月の資料 | GitHub\*\*](#)  
(本スライドのPDF、サンプルコード)
- [\*\*GitHub Actions Documentation - GitHub Help\*\*](#)
- [\*\*GitHub Actions APIでJobを再実行する - Qiita\*\*](#)  
(CIの定期実行について)
- [\*\*GitHub Actions + CircleCI でDocker HubへDockerイメージを登録、定期的にre-buildする\*\*](#)  
(Dockerイメージの定期ビルドの手法)
- [\*\*ansible-roles-matsumura/scheduled-executor\*\*](#)  
(GitHub Actions + Molecule サンプルコード集)