

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行に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

パイプラインの流れ

1. sudoがインストールされたCentOS7イメージをビルド
2. コンテナを起動
3. Galaxyの依存関係をインストール
4. RoleをPlaybookとして実行
5. 幕等性のチェック
6. 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
```

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

Ansible galaxy roleをインストール

```
.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

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. Ansibleをインストール
2. コードをチェックアウト
3. DockerイメージをDockerfileからビルド
4. Dockerコンテナを起動
5. Galaxy roleをインストール
6. Playbookを実行
7. 幕等性のチェック
8. 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
```

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を実行する
Pythonのversionを
複数指定**

.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 2.9.10
    run: |
      pip install ansible==2.9.10
```

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 2.9.10
    run: |
      pip install ansible==2.9.10
```

Ansible 2.9.10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 \
      -p tests/roles
```

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 \
 - p tests/roles

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 \
      -p tests/roles
```

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の適応
- 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)