GH ACTIONS – ANSIBLE

GH ACTIONS – ANSIBLE

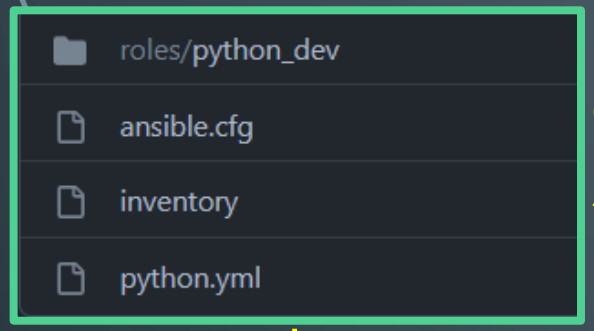
Task – Iimplementing a Deployment Pipeline, using ansible and GitHub Actions for invocation.

All codes are stored in own public GitHub repository Actions-Ansible .

Below, provided overview of current working tree in GitHub:

ANSIBLE – WORKING TREE

In ansible working directory consist of next: roles, configuration files (ansible.cfg), inventory file (inventory) and ansible-playbook itself calling python.yml




```
1  [defaults]
2  host_key_checking       = false
3  inventory               = inventory
4  roles_path              = $HOME/$USER/ghActions-Ansible/ansible/roles
```

In ansible.cfg as a default were defined:

- host_key_checking – to avoiding prompt key confirmation
- path to inventory file
- and path for roles location

```
---
- name: Installing Python Environment
  hosts: developers
  become: true
  roles:
     - python_dev
```

```
1  [developers]
2  localhost ansible_connection=local
```

Inventory:

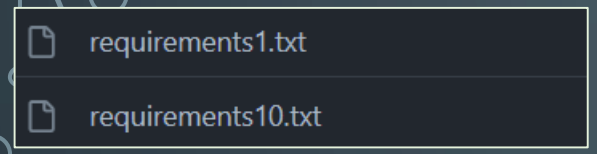
- – given a name of host (localhost) and connection type (otherwise used IP address or DNS name)

Ansible playbook:

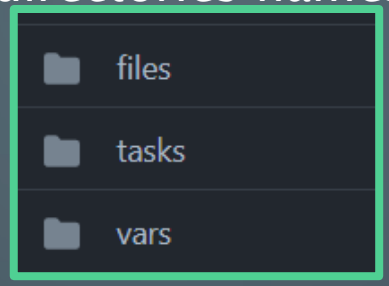
- playbook will be run on a group of hosts with name developers;
- running tasks in **privilege** mode;
- roles to be started – python_dev.

ANSIBLE – ROLES

Ansible roles structure have been generated by using ansible-galaxy.
For that project, its enough to have only 3 directories names: files, tasks and vars.



Files **requirements.txt** to be installed as py-venvs



Declared variable – **{{ dest_path }}** for location of py-venvs

```yaml
---
dest_path: /opt/python_venvs/
```

```yaml
---
- name: Update Ubuntu cache
  apt:
    update_cache: true
    force_apt_get: true
    cache_valid_time: 3600
- name: Install and upgrade pip
  pip:
    name: pip
    extra_args: --upgrade
    executable: pip3
- name: Install Python tools
  apt:
    pkg:
      - python3-pip
      - python3-setuptools
      - python3.8-venv
    state: present
    update_cache: true
- name: Install Python venvs from requirements.txt
  pip:
    requirements: "{{ item }}"
    virtualenv: "{{ dest_path }}/{{ item | basename| splitext | first }}"
    virtualenv_command: "{{ ansible_python.executable }} -m venv"
  with_fileglob: "*.txt"
  loop_control:
    label: "{{ item | basename | splitext | first }}"
```

1st task – update Ubuntu cache if it is older than 1hour

2nd task – Install & upgrade pip

3rd task – Install python and necessary tools

4th task – install py-venvs from requirements.txt
For this task used lookup plugin **with_fileglob** to find a files which a matching to a pattern "***.txt**".
Command **item | basename | splitext | first** carry out segregation of only name "**requirements***" which would be used later in the loop for creating python venvs.

GH ACTIONS – WORKFLOW

```yaml
name: Ansible
on:
  push:
    branches:
    - main

  workflow_dispatch:

jobs:
  validate:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v3.0.0
      - name: Run ansible-lint
        uses: ansible-community/ansible-lint-action@v6.0.2
        with:
          args: "ansible"

  deployAnsible:
    runs-on: ubuntu-latest
    needs: [validate]
    steps:
      - uses: actions/checkout@v3.0.0
      - name: Run Ansible playbook
        uses: dawidd6/action-ansible-playbook@v2.5.0
        with:
          playbook: python.yml
          directory: ansible
        # key: ${{ secrets.SSH_KEY }}

      - name: Checking adjusted Python venvs
        run: ls -la "/opt/python_venvs/"
```

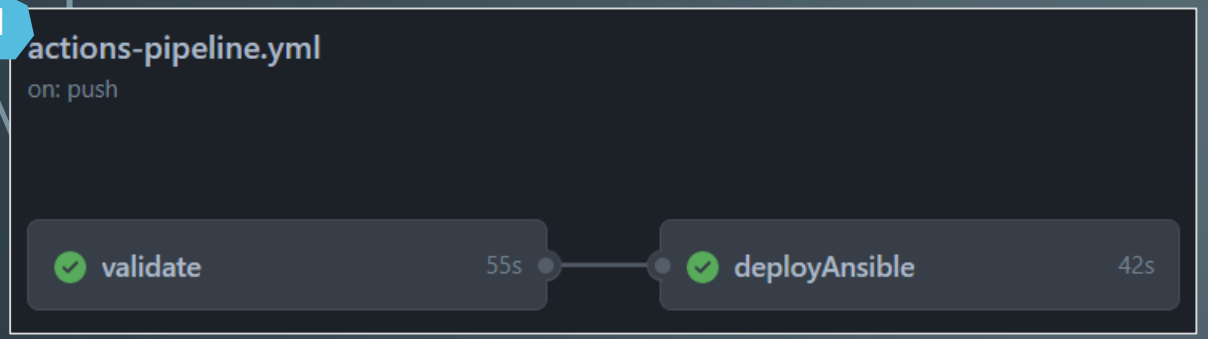
Creating of pipeline begin with name assignment of the workflow and defining event that triggers of the workflow. In our case – on push (main branch) and manually.

First job naming – validate, performing checking for a quality and respectability to a proven practices of an ansible files.
As a runner – using GH Actions runner.
Linux – Ubuntu.
Before start to run ansbile-lint from marketplace, need to checkout repository to GH runner.

Second job naming – deployAnsbile, responsible for running ansible playbook. But, first of all it would start only if previous job "validate" finished successfully.
After that, deployment of ansible-playbook using special tool from GH marketplace.
Finally, verification if python-vevns installed configured properly.

GH ACTIONS – PIPELINE OVERVIEW

1



2

```
16  DEBUG    Examining ansible/python.yml of type yaml
17  DEBUG    Examining ansible of type role
18  DEBUG    Examining ansible/roles/python_dev/tasks/main.yml of type tasks
19  DEBUG    Examining ansible/roles/python_dev/vars/main.yml of type vars
```

1. Overall view to GHA pipeline.
2. Examining by ansible-lint ansbile ".yml" files.
3. Run playbook.
4. Checking presence of python_venvs in mentioned path.

3

```
▶ Run dawidd6/action-ansible-playbook@v2.5.0
/opt/pipx_bin/ansible-playbook python.yml

PLAY [Installing Python Environment] ***************************************

TASK [Gathering Facts] *****************************************************
ok: [localhost]

TASK [python_dev : Update Ubuntu cache] ************************************
changed: [localhost]

TASK [python_dev : Install and upgrade pip] ********************************
changed: [localhost]

TASK [python_dev : Install Python tools] ***********************************
ok: [localhost]

TASK [python_dev : Install Python venvs from requirements.txt] *************
changed: [localhost] => (item=requirements10)
changed: [localhost] => (item=requirements1)

PLAY RECAP *****************************************************************
localhost                  : ok=5    changed=3    unreachable=0    failed=0    skip
```

4

```
▶ Run ls -la "/opt/python_venvs/"
total 16
drwxrwxrwx+  4 root root 4096 Sep  7 15:58 .
drwxrwxrwx+ 16 root root 4096 Sep  7 15:57 ..
drwxrwxrwx+  6 root root 4096 Sep  7 15:58 requirements1
drwxrwxrwx+  6 root root 4096 Sep  7 15:57 requirements10
```