**BSafe Project Steps**

**Github URL:** [**https://github.com/rajib2k5/BSafeProject**](https://github.com/rajib2k5/BSafeProject)

**Here are list of steps taken for this project:**

1. **Created EC2 Instance(s)**

Graphical user interface, table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. **Associated Elastic IP with them**



1. **Testing connectivity on the EC2 instance via elastic ip**

Text

Description automatically generated

1. **Jenkins installation status on AWS Ubuntu 20.04**

A screenshot of a computer

Description automatically generated with medium confidence

1. **Added Jenkins support in Security Group**

Table

Description automatically generated

1. **Accessing Jenkins via the web**

Graphical user interface, text, application

Description automatically generated

1. **Installing plugins**

Graphical user interface, application, table

Description automatically generated

1. **Ansible installation & configuration output**

Text

Description automatically generated

Text

Description automatically generated

1. **Docker installation status**

Graphical user interface, text, application

Description automatically generated

1. **Ansible playbooks for this project**

::::::::::::::

0\_clone\_repo.yaml

::::::::::::::

- hosts: localhost

tasks:

- name: Clone BSafeProject github repository

git:

repo: https://github.com/rajib2k5/BSafeProject

dest: /project/ansible\_project/BSafeProject/

clone: yes

update: yes

::::::::::::::

1\_build\_repo.yaml

::::::::::::::

- hosts: localhost

tasks:

- name: Build BSafeProject Python app

# replicate production build and config tasks

shell: |

cd /project/ansible\_project/BSafeProject/

cp /project/ansible\_project/BSafeProject/conf/env.properties.j2 /project/ansible\_project/BSafeProject/conf/env.properties

docker build . -t bsafeproject

::::::::::::::

2\_deploy.yaml

::::::::::::::

- hosts: localhost

tasks:

- name: Deploy Production Config

shell: |

echo Deploying production configuration file

docker cp /project/ansible\_project/BSafeProject/conf/env.properties.j2 80e53a3e0c34:/app/conf/env.properties

- name: Post Deployment cleanup

#become: true

#become\_user: root

shell: |

echo Go back to project directory and stash local git so we can deploy next time

cd /project/ansible\_project/BSafeProject/

git stash

1. **Pipeline setup**

pipeline {

agent any

stages {

stage('Ansible Playbook to clone code...') {

steps {

echo 'Clone repo'

ansiblePlaybook installation: 'Ansible', playbook: '/project/ansible\_project/0\_clone\_repo.yaml'

}

post {

success {

echo 'Done with cloning...'

}

}

}

stage('Ansible Playbook to build code...') {

steps {

echo 'Build code using maven...'

ansiblePlaybook installation: 'Ansible', playbook: '/project/ansible\_project/1\_build\_repo.yaml'

}

post {

success {

echo 'Done with building...'

}

}

}

stage('Ansible Playbook to deploy production configuration & cleanup...') {

steps {

echo 'Deploy production artifacts...'

ansiblePlaybook installation: 'Ansible', playbook: '/project/ansible\_project/2\_deploy.yaml'

}

post {

success {

echo 'Done with deploying...'

}

}

}

}

}

Graphical user interface, text, application, email

Description automatically generated

1. **Test run of Ansible playbook as Jenkins user**

A picture containing timeline

Description automatically generated

1. **Successful output of the CICD process in Jenkins**
2. Table

   Description automatically generated**D**

**This project required setting up proper permission for Jenkins user to run docker container, adding jenkins user the ability to pull from git repo, run ansible playbook. Some time, I had to stash git repo to restart the cicd process successfully. Some of the steps were probably not the best practice. The stages are functional for this POC project.**