Automating Infrastructure and Software Configuration with Terraform, Ansible, Jenkins, and Docker

==========

Requiremnt:

1. Execution to be under one GCP project per environment (Environments : DEV & PROD) (Remarks: No individual projects) Group has to work on one GCP project

   Required 2 Projects:
   project_name: gcp_adq_pocproject_dev
   project_name: gcp_adq_pocproject_prd

2. Folder Structure & Server names for the application and jenkins server

    Jenkins Server : Terraform folder should be renamed to "adq-jenkins-box", Server name/folder name/label - application, label (jenkins-server), environment (dev/prd)

        Application Server (ubuntu desktop) : Folder to created - "adq_ubuntudesktop"

        adq_ubuntudesktop
                --- terraform
                                --- code (dev & prd)
                --- ansible
                        --- code (dev & prd)
                --- Jenkinsfile
                --- Dockerfile
                --- ReadMe (Complete flow of the Job is explained here - adq_ubuntudesktop)

3. Under Ansible

   Install Apache, Tomcat Server, Python, Notepad++, java
   Restart Services
   Application Samples (Python helloworld/notebook/dairy) and Java application

4. Under Jenkins Server:

a. Application Pipelines
        1. Create Applications Pipelines - Java Application CICD approach
        2. Create Applications Pipelines - Python Application CICD approach
        3. above 1 & 2 Pipelines should have below scenarios
            -- when you run the pipeline, it should build the code from the Github , upload the code to either GCS Bucket/ Nexus/jfrog - Version : py_app_v1.0 (v1.1,1.2.....etc)
                -- you should ask the prompt whether to deploy the application or not, if not , job should be closed, if yes , it should deploy the code to ubuntu server (adq_ubuntudesktop)

b. IaC Pipelines

Create Job *** "adq_ubuntudesktop" , jenkinsfile path : adq_ubuntudesktop\Jenkinsfile, Add Parameters - Environment (DEV& PRD), Terraform (Skip, Plan, Apply, Destroy), Terraform Extra Args (Txt box), Ansible (Skip, Apply, Destroy), Ansible Playbook (txt box), Ansible Extra Args (txt box), Build Button

Build Scenarios:

1. Environment - DEV/PRD, Terraform Apply , Ansible Apply :- it should be execute the Terraform & Ansible Script.

   Limitations: When you execute again terraform apply -> it should333 the throw the exception/Ask the Prompt whether to recreate . If Yes, it should recreate the server and further..., If no, Job should be failed and no more further stages to execute.

2. Execution of Ansible Packages : Environment - DEV/PRD, Terraform Skip , Ansible Apply :- it should be execute only Ansible Script
3. Destroy of the Server: Environment - DEV/PRD, Terraform Destroy , Ansible - not required (optional whatever): it should destroy the infrastructure under the environment
4. Check the Terraform Code -> Environment - DEV/PRD, Terraform Plan , Ansible (not required).
5. Destroy the Ansible Stufff -> Environment - DEV/PRD, Terraform Skip , Ansible Destroy :- It should uninstall the packages /applications -- LEAST Priority
6. Terraform Extra Args
7. Ansible Playbook : Example - python_deployment.yml
7. Ansible Extra Args -- tags/version

```
================================================================================
================================================
get-ubuntudesktop-iac
.
├── adq-jenkins-box
│   ├── adq-java-app  ------> JAVA APP DEPLOYMENT JENKINS FILE JOB1
│   │   ├── Jenkinsfile
│   │   └── sonar.sh
│   ├── adq-python-app -----> PYTHON DEPLOYMENT JENKINS FILE JOB2 <
│   │   └── Jenkinsfile
│   ├── main.tf        ----------> JENKINS VM CREATION --> CREATED BY BASE
VM(EXECUTION) CONTAINS STARTUP SCRIPT FOR DOCKER AND TERRAFORM INSTALL
│   ├── modules
│   │   └── adq-jenkins-box --->  JENKINS VM CREATION <---
│   │       ├── main.tf
│   │       └── variables.tf
│   ├── terraform.tfvars    -------------------------> Values for JENKINS VM with
N/W name created at the Root
│   └── variables.tf
--------------------------------------------------------------------------------
```

```
-------------------------------------------------------
================================================================================
================
├── adq-ubuntudesktop  -> CREATES A TRAGET VM VIA JENKINS AND USES ANSILE CONFIG AND
CREATES THE SOFTWARE INSTALLATION ON THE TARGET VM USING DOCKER AGENT AS SLAVENODE
│   ├── Dockerfile  ----->  CLOUD SDK INSTALL AND USER CREATION JENKINS FOR AGENT
--------------
│   ├── Jenkinsfile ----------------------------------------> CREATES INFRA FOR
TARGET VM ----CREATED BY JENKINS VM USING DOCKER CONTAINER-------
│   ├── ansible
│   │   ├── ansible.cfg  ---------------------------------------> MAKE CONNCTION TO
THE TRAGET VM AND RUNS PLAYBOOK
│   │   ├── environments
│   │   │   ├── dev
│   │   │   │   ├── groups
│   │   │   │   │   ├── adq_ubuntudesktop.yaml ---- TO PARTICULAR VM TO CONFIGURE
│   │   │   │   │   └── all.yaml      ----------------------------------- TO ALL
HOSTS IN THE GCP VM HAVE THE SSH KEY
│   │   │   │   └── inventory.gcp.yaml  ---------------------->CONNECTION THROUGH
DYNAMIC<------------------
│   │   │   └── prod
│   │   │       ├── groups
│   │   │       │   └── all_hosts.yaml
│   │   │       └── inventory.gcp.yaml
│   │   ├── main.yaml
│   │   ├── playbook_deployment.yml
│   │   ├── playbook_destroy.yml
│   │   ├── playbook_services.yml
│   │   ├── playbook_softwares.yml
│   │   └── roles
│   │       ├── java
│   │       │   └── tasks
│   │       │       └── main.yaml  -> INSTALL JAVA 11 DEFAULT OR ELSE TAKES VERSION
FROM JENKINS PARAMETERS
│   │       ├── java-d
│   │       │   └── tasks
│   │       │       └── main.yaml  -> UINSTALL JAVA 11 DEFAULT OR ELSE TAKES VERSION
FROM JENKINS PARAMETERS
│   │       ├── notepad++
│   │       │   └── tasks
│   │       │       └── main.yaml  -> INSTALL NOTEPAD ++
│   │       ├── notepad-d
│   │       │   └── tasks
│   │       │       └── main.yaml  -> UNINSTALL NOTEPAD ++
│   │       ├── python
│   │       │   └── tasks
│   │       │       └── main.yaml   -> INSTALL PYTHON 3
│   │       ├── python-d
│   │       │   └── tasks
│   │       │       └── main.yaml   -> UNINSTALL PYTHON 3
```

```
|   |       ├── tomcat
|   |       |   ├── handlers
|   |       |   |   └── main.yaml  ------------------> TOMCAT INSTALL WITH JENKINS
PARAMETER FOR VERSION USING HANDLER CONCEPT
|   |       |   └── tasks
|   |       |       └── main.yaml
|   |       ├── tomcat-d
|   |       |   └── tasks
|   |       |       └── main.yaml  ----------------> TOMACT UNINSTALL
|   |       ├── ubuntudesktop
|   |       |   ├── handlers
|   |       |   |   └── main.yaml   ---------> UBUNTU CHROME REMOTE DESKTOP
INSTALLATION USING HANDLERS------->
|   |       |   └── tasks
|   |       |       └── main.yaml
|   |       └── ubuntudesktop-d
|   |           └── tasks      ---------> UBUNTU CHROME REMOTE DESKTOP
UNINSTALLATION USING HANDLERS------->
|   └──
terraform==============================================================================
==========================================
|
                                      CREATE TRAGET VM AT DEV PROJECT OR PROD PROJECT
VIA JENKINS WITH N/W & SA CREATED BY ROOT TERRAFORM MDOULE - U CREATE A N/W IN ONE
PROJECT AND USE THE NAME                  HERE INSTEAD OF N/W MODULE
|
|       ├── environments
|       |   ├── dev
|       |   |   ├── backend.tf
|       |   |   ├── main.tf
|       |   |   ├── terraform.tfvars
|       |   |   └── variables.tf
|       |   └── prod
|       |       ├── backend.tf
|       |       ├── main.tf
|       |       ├── terraform.tfvars
|       |       └── variables.tf
|       └── modules
|           └── ubuntu-desktop
|               ├── main.tf
|               └── variables.tf
|
└── terraform
================================================================================
=================
    ├── environments
    |   ├── dev                     ------------------ DEV PROJECT N/W & SA
VARIABLES ----------------------
    |   |   ├── backend.tf
    |   |   ├── main.tf
    |   |   ├── terraform.tfvars
```

```
    │   │   └── variables.tf
    │   └── prod                      ------------------- PROD PROJECT N/W & SA
VARIABLES ------------------
    │       ├── backend.tf
    │       ├── main.tf
    │       ├── terraform.tfvars
    │       └── variables.tf
    └── modules
        ├── networking     ---------------------------------------N/W FOR JENKINS
-----------
        │   ├── main.tf
        │   ├── outputs.tf
        │   └── variables.tf
        └── service-account------------------------------------ SA FOR WORKING ON
ENTIRE INFRA BY IMPERSONATING JENKINS VM --------------------
            ├── main.tf
            ├── outputs.tf
            └── variables.tf
```