Terraform Example for CI/CD with GitHub Actions

1. Login to GCP: Authenticate with Google Cloud

D:\Terraform-Training\terraform\_1.9.2\_windows\_amd64\GCP\demo-1>gcloud auth application-default login

1. Verify the Authentication

D:\Terraform-Training\terraform\_1.9.2\_windows\_amd64\GCP\demo-1>gcloud auth application-default print-access-token

ya29.a0AcM612xih9TwIRzezhSuMfpVmEAu6JNiykRQOj0cn2YS\_T8YeeZPh19KmKIZMQwLt9MdZUS5vngVboN\_vQxQctwTGv7hmWrxI9MymcXB2kEfzSolqYpeFxADTmASrplz52RbkXNBUyoYB2OPSLS0R3DtlTSdv6l\_HoaaaCgYKAbISARASFQHGX2Mi7NgFnhdnFxlGmj9Q-mRBNw0171

**Step 1: Setting Up Terraform Configuration**

First, ensure you have your Terraform configuration files set up. For instance, you might have a file main.tf to define your infrastructure:

**provider "google" {**

**project = "projectkk-terraform"**

**region = "us-central1"**

**}**

**resource "google\_compute\_instance" "vm\_instance" {**

**name = "my-instance"**

**machine\_type = "n1-standard-1"**

**zone = "us-central1-a"**

**boot\_disk {**

**initialize\_params {**

**image = "** **debian-cloud/debian-10-buster-v20210817"**

**}**

**}**

**network\_interface {**

**network = "default"**

**access\_config {}**

**}**

**}**

A screenshot of a computer error

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

**Step 2: Complete Terraform Setup with CI/CD**

**To run the .github/workflows/terraform.yml workflow in GitHub Actions, you need to follow these steps:**

**Step-by-Step Guide**

1. **Add the Workflow File: Ensure the .github/workflows/terraform.yml file is present in your repository.**
2. **Create a Service Account and Generate a Key:**
   * **Go to the Google Cloud Console.**
   * **Navigate to IAM & Admin > Service Accounts.**
   * **Create a new service account or use an existing one.**
   * **Grant the necessary permissions to the service account (e.g., roles like Editor or specific roles like Compute Admin).**
   * **Create a JSON key for this service account and download it.**
3. **Add the Service Account Key to GitHub Secrets:**
   * **Go to your GitHub repository.**
   * **Navigate to Settings > Secrets and variables > Actions.**
   * **Click on New repository secret.**
   * **Add a secret with the name GCP\_CREDENTIALS and paste the content of the JSON key file.**
4. **Push Code to GitHub:**
   * **Commit and push your Terraform configuration and the workflow file to your repository. The workflow will automatically trigger on the specified events (e.g., push or pull request to the main branch).**