Terraform Script to Configure Google Cloud Composer Environment

Overview

This documentation provides a detailed overview of a Terraform script designed to configure a Google Cloud Composer environment. The script includes enabling the Composer API, creating a custom service account, assigning necessary IAM roles, and creating the Cloud Composer environment with specific configurations.

Variables

The script uses several input variables to customize the configuration:

project_id: The ID of the Google Cloud project.
region: The region for the resources.
sa_name: The name for the custom service account.
project_number: The project number.
composer_env_name: The name for the Cloud Composer environment.
Additionally, local variables are used for environment size and various workload configurations, such as worker, scheduler, triggerer, and web server configurations.

Resources

1. Enable Composer API

The script enables the Composer API for the specified project. The disable_on_destroy parameter is set to false to prevent breaking other environments if the API is disabled.

2. Create Custom Service Account

A custom service account is created with a specified account ID and display name.

3. Assign IAM Roles to Service Account

The script assigns necessary IAM roles to the custom service account:

The **roles/composer.worker** role is assigned at the project level.
The **roles/composer.ServiceAgentV2Ext** role is granted to the custom service account, allowing it to interact with Cloud Composer.

4. Create Cloud Composer Environment

A Cloud Composer environment is created with the following configurations:

Environment Size: Set using a local variable.
Software Configuration: Specifies the image version and required PyPI packages.
Node Configuration: Uses the custom service account.

Workloads Configuration: Specifies the resource requirements for various components such as workers, schedulers, triggerers, and web servers.

Conclusion

This Terraform script provides a structured approach to setting up a Google Cloud Composer environment with customized configurations. It ensures proper access controls and resource allocations through the use of service accounts and IAM roles. The flexibility of the script allows for easy scalability and adjustments to meet specific needs.