renovate-bot chore(deps): Update Terraform GoogleCloudPlatform/cloud-run/google to... 87c6b0a · 4 months ago

Name		Name	Last commit date
..			
README.md	fix: remove duplicate variable key (#15...	2 years ago	
main.tf	chore(deps): Update Terraform Googl...	4 months ago	
outputs.tf	Feat/integration tests for secure cloud...	3 years ago	
terraform.example.tfvars	add suport multiple domains loadbala...	3 years ago	
variables.tf	fix: remove duplicate variable key (#15...	2 years ago	

README.md

Secure Cloud Run Example

This example showcases the deployment of Secure Cloud Run, along with domain mapping and IAM policy for the service.

The resources/services/activations/deletions that this example will create/trigger are:

- Creates Firewall rules on your **VPC Project**.
  - Serverless to VPC Connector
  - VPC Connector to Serverless
  - VPC Connector to LB
  - VPC Connector Health Checks
- Creates a sub network to VPC Connector usage purpose.
- Creates Serverless Connector on your **VPC Project** or **Serverless Project**. Refer the comparison below:
  - Advantages of creating connectors in the VPC Project
  - Advantages of creating connectors in the Serverless Project
- Grant the necessary roles for Cloud Run are able to use VPC Connector on your Shared VPC when creating VPC Connector in host project.
  - Grant Network User role to Cloud Services service account.
  - Grant VPC Access User to Cloud Run Service Identity when deploying VPC Access.
  - Creates KMS Keyring and Key for customer managed encryption keys in the **KMS Project** to be used by Cloud Run.
  - Enables Organization Policies related to Cloud Run in the **Serverless Project**.

- Allow Ingress only from internal and Cloud Load Balancing.
- Allow VPC Egress to Private Ranges Only.
- Creates a Cloud Run Service.
- Creates a Load Balancer Service using Google-managed SSL certificates.
- Creates Cloud Armor Service only including the pre-configured rules for SQLi, XSS, LFI, RCE, RFI, Scannerdetection, Protocolattack and Sessionfixation.

Organization Policies

By default, this example will apply 2 organization policies at the project level for the **Serverless Project**.

- Allow Ingress only from internal and Cloud Load Balancing.
- Allow VPC Egress to Private Ranges Only.

To the organization policies to be applied at folder or organization level, the `policy_for` variable needs to be changed. Possible values: ["project", "folder", "organization"] and the variables `folder_id` or `organization_id` need to be be filled up, respectively.

Usage

To provision this example, run the following from within this directory:

- Rename `terraform.example.tfvars` to `terraform.tfvars` by running `mv terraform.example.tfvars terraform.tfvars` and update the file with values from your environment.
- `terraform init` to get the plugins
- `terraform plan` to see the infrastructure plan
- `terraform apply` to apply the infrastructure build

Clean up

- Run `terraform destroy` to clean up your environment.

Assumptions and Prerequisites

This example assumes that below mentioned pre-requisites are in place before consuming the example.

- All required APIs are enabled in the GCP Project.
- An Organization.
- A Billing Account.

Inputs

Name	Description	Type
cloud_armor_policies_name	Cloud Armor policy name already created in the project. If `create_cloud_armor_policies` is `false` , this variable must be provided, If	`string`

Name	Description	Type
	`create_cloud_armor_policies` is `true` , this variable will be ignored.	
cloud_run_sa	Service account to be used on Cloud Run.	`string`
create_cloud_armor_policies	When `true` , the terraform will create the Cloud Armor policies. When `false` , the user must provide their own Cloud Armor name in `cloud_armor_policies_name` .	`bool`
domain	Domain list to run on the load balancer. Used if `ssl` is `true` .	`list(string)`
folder_id	The folder ID to apply the policy to.	`string`

groups	Groups which will have roles assigned.
	The Serverless Administrators email group which the following roles will be added: Cloud Run Admin, Compute Network Viewer and Compute Network User.
	The Serverless Security Administrators email group which the following roles will be added: Cloud Run Viewer, Cloud KMS Viewer and Artifact Registry Reader.
	The Cloud Run Developer email group which the following roles will be added: Cloud Run Developer, Artifact Registry Writer and Cloud KMS CryptoKey Encrypter.
	The Cloud Run User email group which the following roles will be added: Cloud Run Invoker.
		`object({`
		`group_serverless_administrator`
		`group_serverless_security_administ`
		`group_cloud_run_developer`
		`group_cloud_run_user`
		`})`

ip_cidr_range	The range of internal addresses that are owned by the subnetwork and which is going to be used by VPC Connector. For example, 10.0.0.0/28 or 192.168.0.0/28. Ranges must	`string`

Name	Description	Type
	be unique and non-overlapping within a network. Only IPv4 is supported.	
kms_project_id	The project where KMS will be created.	`string`
organization_id	The organization ID to apply the policy to.	`string`
policy_for	Policy Root: set one of the following values to determine where the policy is applied. Possible values: ["project", "folder", "organization"].	`string`
resource_names_suffix	A suffix to concat in the end of the network resources names.	`string`
serverless_project_id	The project where cloud run is going to be deployed.	`string`
shared_vpc_name	Shared VPC name which is going to be re-used to create Serverless Connector.	`string`
vpc_project_id	The project where shared vpc is.	`string`

Outputs

Name	Description
cloud_services_sa	Service Account for Cloud Run Service.
connector_id	VPC serverless connector ID.
domain	Domain name to run the load balancer on. Used if `ssl` is `true`.
domain_map_id	Unique Identifier for the created domain map.
domain_map_status	Status of Domain mapping.
folder_id	The folder ID to apply the policy to.
gca_vpcaccess_sa	Service Account for VPC Access.
key_name	Key name.
keyring_name	Keyring name.
kms_project_id	The project where KMS will be created.
load_balancer_ip	IP Address used by Load Balancer.
organization_id	The organization ID to apply the policy to.

Name	Description
policy_for	Policy Root: set one of the following values to determine where the policy is applied. Possible values: ["project", "folder", "organization"].
project_id	The project where Cloud Run will be created.
revision	Deployed revision for the service.
run_identity_services_sa	Service Identity to run services.
service_id	Unique Identifier for the created service.
service_status	Status of the created service.
service_url	The URL on which the deployed service is available.
shared_vpc_name	Shared VPC name which is going to be re-used to create Serverless Connector.
vpc_project_id	The project where VPC Connector is going to be deployed.

Requirements

These sections describe requirements for using this example.

Software

- Terraform ~> v0.13+
- Terraform Provider for GCP >= 3.53, < 5.0
- Terraform Provider for GCP Beta >= 3.53, < 5.0

Service Account

A service account can be used with required roles to execute this example:

- Compute Shared VPC Admin: `roles/compute.xpnAdmin`
- Security Admin: `roles/compute.securityAdmin`
- Serverless VPC Access Admin: `roles/vpcaccess.admin`
- Cloud KMS Admin: `roles/cloudkms.admin`
- Serverless VPC Access Admin: `roles/vpcaccess.admin`
- Cloud Run Developer: `roles/run.developer`
- Compute Network User: `roles/compute.networkUser`
- Artifact Registry Reader: `roles/artifactregistry.reader`