terraform-google-cloud-run / modules / secure-cloud-run / Add file

release-please[bot] chore(main): release 0.17.4 (#338) 90b0e7d · 2 days ago

Name	Name	Last commit date
..		
README.md	fix: remove duplicate variable key (#15...	2 years ago
main.tf	fix!: upgrade modules to use provider ...	6 months ago

terraform-google-cloud-run / modules / secure-cloud-run / Top

metadata.yaml	chore(main): release 0.17.4 (#338)	2 days ago
outputs.tf	feat!: changes net module to be serve...	2 years ago
variables.tf	fix: remove duplicate variable key (#15...	2 years ago
versions.tf	chore(main): release 0.17.4 (#338)	2 days ago

README.md

Secure Cloud Run

This module handles the deployment required for Cloud Run usage. Secure-cloud-run module will call the secure-cloud-run-core, secure-serverless-net and secure-cloud-run-security modules.

When using a Shared VPC, you can chose where to create the VPC Connector.

Note: When using a single VPC you should provides VPC and Serverless project id with the same value and the value for `connector_on_host_project` variable must be `false` .

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

- secure-serverless-net module will apply:
  - 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 be 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.

- secure-cloud-run-security module will apply:
    - 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.
    - When groups emails are provided, this module will grant the roles for each persona.
        - Serverless Administrator - **Service Project**
            - Cloud Run Administrator: `roles/run.admin`
            - Cloud Compute Network Viewer: `roles/compute.networkViewer`
            - Cloud Compute Network User: `compute.networkUser`
        - Servervless Security Administrator - **Security Project**
            - Cloud Run Viewer: `roles/run.viewer`
            - Cloud KMS Viewer: `roles/cloudkms.viewer`
            - roles/artifactregistry.reader
        - Cloud Run developer - **Security Project**
            - Cloud Run Develper: `roles/run.developer`
            - Cloud Run: `roles/artifactregistry.writer`
            - Cloud Run KMS Encrypter: `roles/cloudkms.cryptoKeyEncrypter`
        - Cloud Run user - **Security Project**
            - Cloud Run Invoker: `roles/run.invoker`

- secure-cloud-run-core module will apply:
    - 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.

Usage

Basic usage of this module is as follows:

```
module "secure_cloud_run" {
  source = "../modules/secure-cloud-run"

  vpc_project_id                          = <VPC Project ID>
  kms_project_id                          = <KMS Project ID>
  serverless_project_id                   = <Serverless Project ID>
  domain                                  = <Domain>
```

```
  shared_vpc_name                       = <Shared VPC Name
  ip_cidr_range                         = <IP CIDR Range>
  service_name                          = <Service Name>
  location                              = <Location>
  region                                = <Region>
  image                                 = <Image>
  cloud_run_sa                          = <Cloud Run Service Account>
  artifact_registry_repository_location   = <Artifact Registry Repository Location>
  artifact_registry_repository_name       = <Artifact Registry Repository Name>
  artifact_registry_repository_project_id = <Artifact Registry Repository Project ID>
}
```

Inputs

Name	Description	
artifact_registry_repository_location	Artifact Registry Repository location to grant serverless identity viewer role.	`string`
artifact_registry_repository_name	Artifact Registry Repository name to grant serverless identity viewer role	`string`
artifact_registry_repository_project_id	Artifact Registry Repository Project ID to grant serverless identity viewer role.	`string`
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 `create_cloud_armor_policies` is `true` , this variable will be ignored.	`string`
cloud_run_sa	Service account to be used on Cloud Run.	`string`
connector_name	The name for the connector to be created.	`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`
create_subnet	The subnet will be created with the subnet_name variable if true. When false, it will use the subnet_name for the subnet.	`bool`
env_vars	Environment variables (cleartext)	`list(object({ value = string name = string }))`
folder_id	The folder ID to apply the policy to.	`string`
grant_artifact_register_reader	When true it will grant permission to read an image from your artifact registry. When true,	`bool`

Name	Description	
	you must provide `artifact_registry_repository_project_id`, `artifact_registry_repository_location` and `artifact_registry_repository_name`.	

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_serverle`
		`group_serverle`
		`group_cloud_ru`
		`group_cloud_ru`
		`})`

image	Image url to be deployed on Cloud Run.	`string`
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 be unique and non-overlapping within a network. Only IPv4 is supported.	`string`
key_name	The name of KMS Key to be created and used in Cloud Run.	`string`
key_protection_level	The protection level to use when creating a version based on this template. Possible values: ["SOFTWARE", "HSM"]	`string`
key_rotation_period	Period of key rotation in seconds.	`string`
keyring_name	Keyring name.	`string`
kms_project_id	The project where KMS will be created.	`string`
location	The location where resources are going to be deployed.	`string`
max_scale_instances	Sets the maximum number of container instances needed to handle all incoming requests or events from each revison from Cloud Run. For more information, access this documentation.	`number`

Name	Description	
members	Users/SAs to be given invoker access to the service with the prefix `serviceAccount:'` for SAs and user: for users.	`list(string)`
min_scale_instances	Sets the minimum number of container instances needed to handle all incoming requests or events from each revison from Cloud Run. For more information, access this documentation.	`number`
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`
prevent_destroy	Set the `prevent_destroy` lifecycle attribute on the Cloud KMS key.	`bool`
region	Location for load balancer and Cloud Run resources.	`string`
resource_names_suffix	A suffix to concat in the end of the network resources names being created.	`string`
serverless_project_id	The project to deploy the cloud run service.	`string`
service_name	Shared VPC name.	`string`
shared_vpc_name	Shared VPC name which is going to be re-used to create Serverless Connector.	`string`
ssl_certificates	A object with a list of domains to auto-generate SSL certificates or a list of SSL Certificates self-links in the pattern `projects/<PROJECT-ID>/global/sslCertificates/<CERT-NAME>` to be used by Load Balancer.	`object({ ssl_certificat generate_certi })`
subnet_name	Subnet name to be re-used to create Serverless Connector.	`string`
verified_domain_name	List of Custom Domain Name	`list(string)`
volumes	[Beta] Volumes needed for environment variables (when using secret).	`list(object({ name = string secret = set(o secret_name items }) }))`

Name	Description	
vpc_egress_value	Sets VPC Egress firewall rule. Supported values are all-traffic, all (deprecated), and private-ranges-only. all-traffic and all provide the same functionality. all is deprecated but will continue to be supported. Prefer all-traffic.	`string`
vpc_project_id	The host project for the shared vpc.	`string`

Outputs

Name	Description
cloud_services_sa	Service Account for Cloud Run Service.
connector_id	VPC serverless connector ID.
domain_map_id	Unique Identifier for the created domain map.
domain_map_status	Status of Domain mapping.
gca_vpcaccess_sa	Service Account for VPC Access.
key_self_link	Name of the Cloud KMS crypto key.
keyring_self_link	Name of the Cloud KMS keyring.
load_balancer_ip	IP Address used by Load Balancer.
revision	Deployed revision for the service.
run_identity_services_sa	Service Identity to run services.
service_id	ID of the created service.
service_status	Status of the created service.
service_url	Url of the created service.

Requirements

Software

The following dependencies must be available:

- Terraform >= 0.13.0
- Terraform Provider for GCP < 5.0

APIs

The Secure-cloud-run module will enable the following APIs to the Serverlesss Project:

- Google VPC Access API: `vpcaccess.googleapis.com`
- Compute API: `compute.googleapis.com`

- Container Registry API: `container.googleapis.com`
- Cloud Run API: `run.googleapis.com`

The Secure-cloud-run module will enable the following APIs to the VPC Project:

- Google VPC Access API: `vpcaccess.googleapis.com`
- Compute API: `compute.googleapis.com`

The Secure-cloud-run module will enable the following APIs to the KMS Project:

- Cloud KMS API: `cloudkms.googleapis.com`

Service Account

A service account with the following roles must be used to provision the resources of this module:

- VPC Project
  - Compute Shared VPC Admin: `roles/compute.xpnAdmin`
  - Network Admin: `roles/compute.networkAdmin`
  - Security Admin: `roles/compute.securityAdmin`
  - Serverless VPC Access Admin: `roles/vpcaccess.admin`
- KMS Project
  - Cloud KMS Admin: `roles/cloudkms.admin`
- Serverless Project
  - Security Admin: `roles/compute.securityAdmin`
  - 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`

Note: Secret Manager Secret Accessor role must be granted to the Cloud Run service account to allow read access on the secret.