

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

100 lines (90 loc) · 4.71 KB ·

Code Blame

Raw

```
1  /**
2   * Copyright 2022 Google LLC
3   *
4   * Licensed under the Apache License, Version 2.0 (the "License");
5   * you may not use this file except in compliance with the License.
6   * You may obtain a copy of the License at
7   *
8   *     http://www.apache.org/licenses/LICENSE-2.0
9   *
10  * Unless required by applicable law or agreed to in writing, software
11  * distributed under the License is distributed on an "AS IS" BASIS,
12  * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
13  * See the License for the specific language governing permissions and
14  * limitations under the License.
15  */
16
17  locals {
18      location      = "us-west1"
19      region        = "us-west1"
20      repository_name = "rep-secure-cloud-run"
21
22      hello_image = "us-docker.pkg.dev/cloudrun/container/hello:latest"
23  }
24
25  resource "random_id" "random_folder_suffix" {
26      byte_length = 2
27  }
28
29  module "secure_harness" {
30      source = "GoogleCloudPlatform/cloud-run/google//modules/secure-serverless-harness"
31      version = "~> 0.16"
32
33      billing_account      = var.billing_account
34      security_project_name = "prj-kms-secure-cloud-run"
35      serverless_project_names = ["prj-secure-cloud-run"]
36      org_id                = var.org_id
37      parent_folder_id      = var.parent_folder_id
38      serverless_folder_suffix = random_id.random_folder_suffix.hex
39      region                 = local.region
40      location               = local.location
41      vpc_name                = "vpc-secure-cloud-run"
42      subnet_ip              = "10.0.0.0/28"
43      private_service_connect_ip = "10.3.0.5"
44      create_access_context_manager_access_policy = var.create_access_context_manager_access_policy
```

```

45     access_context_manager_policy_id      = var.access_context_manager_policy_id
46     access_level_members                  = var.access_level_members
47     key_name                              = "key-secure-artifact-registry"
48     keyring_name                          = "krg-secure-artifact-registry"
49     prevent_destroy                       = false
50     artifact_registry_repository_name      = local.repository_name
51     egress_policies                       = var.egress_policies
52     ingress_policies                      = var.ingress_policies
53     base_serverless_api                   = "run.googleapis.com"
54     project_deletion_policy                = "DELETE"
55     folder_deletion_protection             = false
56 }
57
58 resource "null_resource" "copy_image" {
59     provisioner "local-exec" {
60         command = "gcloud container images add-tag ${local.hello_image} ${local.location}-docker.pkg.dev/${module.sec
61     }
62
63     depends_on = [
64         module.secure_harness
65     ]
66 }
67
68 module "secure_cloud_run" {
69     source = "GoogleCloudPlatform/cloud-run/google//modules/secure-cloud-run"
70     version = "~> 0.16"
71
72     location          = local.location
73     region            = local.region
74     serverless_project_id = module.secure_harness.serverless_project_ids[0]
75     vpc_project_id     = module.secure_harness.network_project_id[0]
76     kms_project_id     = module.secure_harness.security_project_id
77     key_name           = "key-secure-cloud-run"
78     keyring_name       = "krg-secure-cloud-run"
79     service_name       = "srv-secure-cloud-run"
80     image              = "${local.location}-docker.pkg.dev/${module.secure_harness.security_pr
81     cloud_run_sa       = module.secure_harness.service_account_email[module.secure_harness.sei
82     connector_name     = "con-secure-cloud-run"
83     subnet_name       = module.secure_harness.service_subnet[0]
84     create_subnet      = false
85     shared_vpc_name    = module.secure_harness.service_vpc[0].network_name
86     ip_cidr_range      = "10.0.0.0/28"
87     prevent_destroy    = false
88     artifact_registry_repository_location = local.location
89     artifact_registry_repository_project_id = module.secure_harness.security_project_id
90     artifact_registry_repository_name      = local.repository_name
91     env_vars                              = [{ name = "TEST", value = "true" }]
92     ssl_certificates = {
93         generate_certificates_for_domains = var.domain
94         ssl_certificates_self_links      = []
95     }
96
97     depends_on = [
98         null_resource.copy_image
99     ]
100 }

```