Ensure Access & Identity in Google Cloud: Challenge Lab

GSP342

Google Cloud Self-Paced Labs

Overview

![/fragments/challenge-overview]

This lab is recommended for students enrolled in the Ensure Access & Identity in Google Cloud quest. Are you ready for the challenge?
Topics tested

- Create a custom security role.
- Create a service account.
- Bind IAM security roles to a service account.
- Create a private Kubernetes Engine cluster in a custom subnet.
- Deploy an application to a private Kubernetes Engine cluster

Setup

Before you click the Start Lab button

Read these instructions. Labs are timed and you cannot pause them. The timer, which starts when you click **Start Lab**, shows how long Google Cloud resources will be made available to you.

This Qwiklabs hands-on lab lets you do the lab activities yourself in a real cloud environment, not in a simulation or demo environment. It does so by giving you new, temporary credentials that you use to sign in and access Google Cloud for the duration of the lab.

What you need

To complete this lab, you need:

- Access to a standard internet browser (Chrome browser recommended).
- Time to complete the lab.
  Note: If you already have your own personal Google Cloud account or project, do not use it for this lab.

Note: If you are using a Pixelbook, open an Incognito window to run this lab.

Challenge scenario

You have started a new role as a junior member of the security team for the Orca team in Jooli Inc. Your team is responsible for ensuring the security of the Cloud infrastucture and services that the company's applications depend on.

You are expected to have the skills and knowledge for these tasks, so don't expect step-by-step guides to be provided.

Your challenge

You have been asked to deploy, configure, and test a new Kubernetes Engine cluster that will be used for application development and pipeline testing by the the Orca development team.

As per the organisation's security standards you must ensure that the new Kubernetes Engine cluster is built according to the organisation's most recent security standards and thereby must comply with the following:

- The cluster must be deployed using a dedicated service account configured with the least privileges required.
- The cluster must be deployed as a Kubernetes Engine private cluster, with the public endpoint disabled, and the master authorized network set to include only the ip-address of the Orca group's management jumphost.
- The Kubernetes Engine private cluster must be deployed to the `orca-build-subnet` in the Orca Build VPC.
  From a previous project you know that the minimum permissions required by the service account that is specified for a Kubernetes Engine cluster is covered by these three built in roles:

- `roles/monitoring.viewer`
- `roles/monitoring.metricWriter`
- `roles/logging.logWriter`
  These roles are sepcified in the documentation for hardening cluster security.
  You must bind the above roles to the service account used by the cluster as well as a custom role that you must create in order to provide access to any other services specified by the development team. Initially you have been told that the development team requires that the service account used by the cluster should have the permissions necessary to add and update objects in Google Cloud Storage buckets. To do this you will have to create a new custom IAM role that will provide the following permissions:

- `storage.buckets.get`
- `storage.objects.get`
- `storage.objects.list`
- `storage.objects.update`
- `storage.objects.create`

Once you have created the new private cluster you must test that it is correctly configured by connecting to it from the jumphost, `orca-jumphost`, in the management subnet `orca-mgmt-subnet`. As this compute instance is not in the same subnet as the private cluster you must make sure that the master authorized networks for the cluster includes the internal ip-address for the instance, and you must specify the `--internal-ip` flag when retrieving cluster credentials using the `gcloud container clusters get-credentials` command.

All new cloud objects and services that you create should include the "orca-" prefix.

Your final task is to validate that the cluster is working correctly by deploying a simple application to the cluster to test that management access to the cluster using the `kubectl` tool is working from the `orca-jumphost` compute instance.

Task 1: Create a custom security role.

Your first task is to create a new custom IAM security role called `orca_storage_update` that will provide the Google Cloud storage bucket and object permissions required to be able to create and update storage objects.

If you don't get a green check mark, click on the **Score** fly-out on the top right and click **Run Step** on the relevant step. A hint pop up opens to give you advice.

Task 2: Create a service account.

Your second task is to create the dedicated service account that will be used as the service account for your new private cluster. You must name this account `orca-private-cluster-sa`.

If you don't get a green check mark, click on the **Score** fly-out on the top right and click **Run Step** on the relevant step. A hint pop up opens to give you advice.

Task 3: Bind a custom security role to a service account.

You must now bind the Cloud Operations logging and monitoring roles that are required for Kubernetes Engine Cluster service accounts as well as the custom IAM role you created for storage permissions to the Service Account you created earlier.

If you don't get a green check mark, click on the **Score** fly-out on the top right and click **Run Step** on the relevant step. A hint pop up opens to give you advice.

Task 4: Create and configure a new Kubernetes Engine private cluster

You must now use the service account you have configured when creating a new Kubernetes Engine private cluster. The new cluster configuration must include the following:

- The cluster must be called `orca-test-cluster`
- The cluster must be deployed to the subnet `orca-build-subnet`
- The cluster must be configured to use the `orca-private-cluster-sa` service account.
- The private cluster options `enable-master-authorized-networks`, `enable-ip-alias`, `enable-private-nodes`, and `enable-private-endpoint` must be enabled.

Once the cluster is configured you must add the internal ip-address of the `orca-jumphost` compute instance to the master authorized network list.

If you don't get a green check mark, click on the **Score** fly-out on the top right and click **Run Step** on the relevant step. A hint pop up opens to give you advice.

Task 5: Deploy an application to a private Kubernetes Engine cluster.

You have a simple test application that can be deployed to any cluster to quickly test that basic container deployment functionality is working and that basic services can be created and accessed. You must configure the environment so that you can deploy this simple demo to the new cluster using the jumphost `orca-jumphost`.

```
kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0
content_copy
```

This deploys an application that listens on port 8080 that can be exposed using a basic load balancer service for testing.

If you don't get a green check mark, click on the **Score** fly-out on the top right and click **Run Step** on the relevant step. A hint pop up opens to give you advice.

Tips and Tricks

- **Tip 1**. When retrieving the `kubectl` credentials for a private cluster that has the `enable-private-endpoint` option set you must specify the `--internal-ip` flag.

```
gcloud container clusters get-credentials [CLUSTER_NAME] --internal-ip --zone=[ZONE]
content_copy
```

- **Tip 2**. When adding the internal ip-address of the `orca-jumphost` machine to the list of authorized adderesses for the private Kubernetes Engine cluster you should use a `/32` netmask to ensure that only the specific compute instance is authorized.

- **Tip 3**. You cannot connect directly to a Kubernetes Engine private cluster from a VPC or other network outside of the VPC the private cluster has been deployed to if the `enable-private-endpoint` option has been specified. This represents the highest security option for a private cluster and you must use a jumphost, or a proxy within the same VPC as the cluster, and you must use that jumphost or proxy to connect to the internal managment ip-address for the cluster.

Congratulations!

Earn Your Next Skill Badge

This self-paced lab is part of the Ensure Access & Identity in Google Cloud skill badge quest. Completing this skill badge quest earns you the badge above, to recognize your achievement. Share your badge on your resume and social platforms, and announce your accomplishment using #GoogleCloudBadge.
See other available Qwiklabs Quests available in the catalog.

Google Cloud Training & Certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.
Manual Last Updated February 23, 2021
Lab Last Tested October 12, 2020

Solution: commands   Video