This page explains how to create Cloud Identity and Access Management (Cloud IAM) (/iam) policies for authorization in Google Kubernetes Engine.

Every Google Cloud, GKE, and Kubernetes API call requires that the account making the request has the necessary permissions. By default, no one except you can access your project or its resources. You can use Cloud IAM to manage who can access your project and what they are allowed to do. Cloud IAM permissions work alongside Kubernetes RBAC (/kubernetes-engine/docs/role-based-access-control), which provides granular access controls for specific objects in a cluster or Namespace. Cloud IAM has a stronger focus on permissions at the level of the Google Cloud project and organization, though it does provide several pre-defined roles specific to GKE.

To grant users and service accounts access to your Google Cloud project, you add them as project team members (/iam/docs/granting-changing-revoking-access), then assign roles (#managing) to the team members. Roles define which Google Cloud resources an account can access and which operations they can perform.

In GKE, you can use Cloud IAM to manage which users and service accounts can access, and perform operations in, your clusters.

Before you start, make sure you have performed the following tasks:

- Ensure that you have enabled the Google Kubernetes Engine API.

  Enable Google Kubernetes Engine API (https://console.cloud.google.com/apis/library/container.googleapis.com?q=kubernetes%20engine)

- Ensure that you have installed the Cloud SDK (/sdk/downloads).

Set up default `gcloud` settings using one of the following methods:

- Using `gcloud init`, if you want to be walked through setting defaults.
- Using `gcloud config`, to individually set your project ID, zone, and region.

Kubernetes' native role-based access control (RBAC) (/kubernetes-engine/docs/role-based-access-control/) system also manages access to your cluster. RBAC controls access on a cluster and namespace level, while Cloud IAM works on the project level.

Cloud IAM and RBAC can work in concert, and an entity must have sufficient permissions at either level to work with resources in your cluster.

The following sections describe the Cloud IAM Roles available in Google Cloud.

Cloud IAM provides predefined Roles (/iam/docs/understanding-roles#predefined_roles) that grant access to specific Google Cloud resources and prevent unauthorized access to other resources.

Cloud IAM offers the following predefined roles for GKE:

Role	Title	Description	Lowest resource
`roles/container.admin`	Kubernetes Engine Admin	Provides access to full management of Container Clusters and their Kubernetes API objects.	Project
`roles/container.clusterAdmin`	Kubernetes Engine Cluster Admin	Provides access to management of Container Clusters.	Project
`roles/container.clusterViewer`	Kubernetes Engine Cluster Viewer	Read-only access to Kubernetes Clusters.	
`roles/container.developer`	Kubernetes Engine Developer	Provides full access to Kubernetes API objects inside Container Clusters.	Project
`roles/container.hostServiceAgentUser`	Kubernetes Engine Host Service Agent User	Use access of the Kubernetes Engine Host Service Agent.	
`roles/container.viewer`	Kubernetes Engine Viewer	Provides read-only access to GKE resources.	Project

To learn about permissions granted by each Cloud IAM role, refer to Permissions granted by Cloud IAM roles (#permissions).

Primitive Cloud IAM roles grant users global, project-level access to all Google Cloud resources. To keep your project and clusters secure, use predefined Roles (#predefined) whenever possible.

To learn more about primitive roles, refer to Primitive roles (/iam/docs/understanding-roles#primitive_roles) in the Cloud IAM documentation.

Service Account User (/iam/docs/service-accounts#the_service_account_user_role) grants a Google Cloud user account the permission to perform actions as though a service account were performing them.

- Granting the `iam.serviceAccountUser` role to a user for a **project** gives the user all of the roles granted to all service accounts in the project, including service accounts that may be created in the future.
- Granting the `iam.serviceAccountUser` role to a user for a specific service account gives a user all of the roles granted to that service account.

This role includes the following permissions:

- `iam.serviceAccounts.actAs`
- `iam.serviceAccounts.get`
- `iam.serviceAccounts.list`

- `resourcemanager.projects.get`
- `resourcemanager.projects.list`

For more information about the ServiceAccountUser role, see ServiceAccountUser (/iam/docs/service-accounts#the_service_account_user_role) in the Cloud IAM documentation.

The following command shows the syntax for granting the Service Account User role:

The Host Service Agent User role is only used in Shared VPC (/kubernetes-engine/docs/how-to/cluster-shared-vpc) clusters. This role includes the following permissions:

- `compute.firewalls.get`
- `container.hostServiceAgent.*`

If predefined roles (#predefined) don't meet your needs, you can create custom roles (/iam/docs/understanding-custom-roles) with permissions that you define.

To learn how to create and assign custom roles, refer to Creating and managing custom roles (/iam/docs/creating-custom-roles).

You can view the permissions granted by each Role using the `gcloud` command-line tool or Cloud Console.

To learn how to manage Cloud IAM roles and permissions for human users, refer to Granting, changing, and revoking access to project members (/iam/docs/granting-changing-revoking-access) in the Cloud IAM documentation.

For *service accounts*, refer to Granting roles to service accounts (/iam/docs/granting-roles-to-service-accounts).

Here are a few examples of how Cloud IAM works with GKE:

- A new employee has joined a company. They need to be added to the Google Cloud project, but they only need to view the project's clusters and other Google Cloud resources. The project owner assigns them the project-level Compute Viewer (/compute/docs/access/iam#compute.viewer) role. This role provides read-only access to get and list nodes, which are Compute Engine resources.
- The employee is working in operations, and they need to update a cluster using `gcloud` or Google Cloud Console. This operation requires the `container.clusters.update` permission, so the project owner assigns them the Kubernetes Engine Cluster Admin role. The employee now has the permissions granted by both the Kubernetes Engine Cluster Admin and Compute Viewer roles.
- The employee needs to investigate why a Deployment is having issues. They need to run `kubectl get pods` to see Pods running in the cluster. The employee already has the Compute Viewer role, which is not sufficient for listing Pods. The employee needs the Kubernetes Engine Viewer role.
- The employee needs to create a new cluster. The project owner grants the employee the Service Account User role for the `[PROJECT_NUMBER]-compute@developer.gserviceaccount.com` service account, so that the employee's account can access Compute Engine's default service account. This service account has the Editor role, which provides a broad set of permission.

- Read the access control overview (/kubernetes-engine/docs/concepts/access-control).
- Learn some best practices for Cloud IAM policies (/solutions/prep-kubernetes-engine-for-prod#managing_identity_and_access).