**🔐 What is RBAC in Kubernetes?**

**RBAC (Role-Based Access Control) controls who can do what within your Kubernetes cluster.**

* **Who (subjects) can do what (verbs) on which resources in what namespace.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**It uses 4 core resources:**

**A screenshot of a computer

AI-generated content may be incorrect.**

Role is used to controlled permissions within namespace and ClusterRole is used to controlled permissions for cluster level.

And once roles are created and you can bind it to individual user/group/serviceaccount.

**A screenshot of a computer

AI-generated content may be incorrect.**

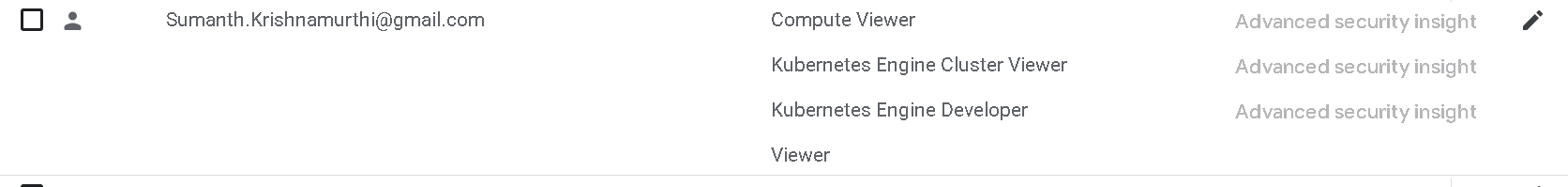
**Now let’s take the use case that I am adding a new user to access Kubernetes console**

**✅ Step 1: Add the user to your GCP project**

**Go to GCP Console → IAM & Admin → IAM  
Or use this direct link: https://console.cloud.google.com/iam-admin/iam**

**Click “Grant Access” and fill:**

* **New Principal: sumanth.krishnamurthi@gmail.com**
* **Roles:**
  + ✅ Kubernetes Engine Cluster Viewer (read access to GKE clusters)
  + ✅ Kubernetes Engine Developer .
  + ✅ Compute Viewer (read-only access to VM instances)
  + ✅ *(Optional)* Viewer for overall read-only access



* **✅ Check the box: "Grant access to all current and future resources"**
* **Click Save**

**Check the IAM roles are added properly**

gcloud projects get-iam-policy dynatrace-fiserv \

--flatten="bindings[].members" \

--filter="bindings.members:sumanth.krishnamurthi@gmail.com" \

--format="table(bindings.role)"

If not, add them via:

gcloud projects add-iam-policy-binding dynatrace-fiserv \

--member="user:sumanth.krishnamurthi@gmail.com" \

--role="roles/container.developer"

**✅ Step 2: (Optional but recommended) Enable GKE API**

**gcloud services enable container.googleapis.com**

**📁 Step 4: Create ClusterRole (cluster-wide read-only)**

# clusterrole-readonly.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

name: cluster-read-only

rules:

- apiGroups: ["", "apps", "batch", "extensions"]

resources: ["pods", "services", "deployments", "replicasets", "nodes", "namespaces", "events"]

verbs: ["get", "list", "watch"]

**📁 Step 5: Create ClusterRoleBinding (map role to user)**

# clusterrolebinding-readonly.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRoleBinding

metadata:

name: cluster-readonly-binding

subjects:

- kind: User

name: sumanth.krishnamurthi@gmail.com

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: ClusterRole

name: cluster-read-only

apiGroup: rbac.authorization.k8s.io

**kubectl apply -f clusterrole-readonly.yaml**

**kubectl apply -f clusterrolebinding-readonly.yaml**

**🔍 Test RBAC Access**

**kubectl auth can-i list pods --as=sumanth.krishnamurthi@gmail.com --namespace=dev**

Now Let’s test this from Cloud-shell console ,

**gcloud auth login**

**Login in as sumanth.krishnamurthi**

**gcloud config set project <your-project-id>**

**gcloud config set project dynatrace-fiserv**

**gcloud container clusters get-credentials <your-cluster-name> --zone <your-zone>**

**gcloud container clusters get-credentials cluster-1 --zone us-central1-c --project dynatrace-fiserv**

kubectl run nginx --image=nginx:latest --port=8081

Error from server (Forbidden): pods is forbidden: User "Sumanth.Krishnamurthi@gmail.com" cannot create resource "pods" in API group "" in the namespace "default": requires one of ["container.pods.create"] permission(s).

**kubectl get clusterroles | grep read-only**

**View All ClusterRoleBindings**

**kubectl get clusterrolebindings**

**🔹 View Roles in a Namespace (e.g., dev)**

**kubectl get roles -n dev**

**🔹 View RoleBindings in a Namespace (e.g., dev)**

**kubectl get rolebindings -n dev**

**🔎 View Detailed Info of a Role/ClusterRole**

**kubectl describe role <role-name> -n <namespace>**

**kubectl describe clusterrole <clusterrole-name>**

**kubectl describe rolebinding <binding-name> -n <namespace>**

**kubectl describe clusterrolebinding <binding-name>**

**🗑️ Delete RBAC Resources**

**kubectl delete clusterrole <clusterrole-name>**

**kubectl delete clusterrolebinding <binding-name>**

**kubectl delete role <role-name> -n dev**

**kubectl delete rolebinding <binding-name> -n dev**

**🧱 Real-Time Use Case: Dev Namespace Access for a Developer**

**Allow developer-level access to only the dev namespace for the user:**

* **Can create/update/delete workloads like Pods, Deployments, Services.**
* **But not touch other namespaces or cluster-level resources.**

gcloud projects add-iam-policy-binding dynatrace-fiserv \

--member="user:sumanth.krishnamurthi@gmail.com" \

--role="roles/container.developer"

**Add IAM Role as ✅ Kubernetes Engine Developer.**

**Create Role / ClusterRole**

**Step1: Developer access to only one namespace (dev)**

# dev-namespace-developer-role.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

namespace: dev

name: dev-namespace-developer

rules:

- apiGroups: [""]

resources: ["pods", "services", "configmaps", "secrets", "persistentvolumeclaims"]

verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

- apiGroups: ["apps"]

resources: ["deployments", "replicasets", "statefulsets"]

verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

- apiGroups: ["batch"]

resources: ["jobs", "cronjobs"]

verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]

**🟦 Step 2: Create a RoleBinding (assign role to user) Bind the Role to Sumanth**

# dev-namespace-developer-rolebinding.yaml

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

name: dev-namespace-developer-binding

namespace: dev

subjects:

- kind: User

name: sumanth.krishnamurthi@gmail.com

apiGroup: rbac.authorization.k8s.io

roleRef:

kind: Role

name: dev-namespace-developer

apiGroup: rbac.authorization.k8s.io

**kubectl apply -f dev-namespace-developer-role.yaml**

**kubectl apply -f dev-namespace-developer-rolebinding.yaml**

**Validate:**

**kubectl auth can-i create pods --namespace=dev --as=sumanth.krishnamurthi@gmail.com**

**yes**

**kubectl auth can-i create deployment --namespace=dev --as=sumanth.krishnamurthi@gmail.com**

**yes**

**Check All RBAC Entries for That User**

**kubectl get rolebindings,clusterrolebindings --all-namespaces -o yaml | grep -A5 sumanth.krishnamurthi**

**🎯 Step 5: Ask Sumanth to Test Access**

gcloud auth login

gcloud config set project dynatrace-fiserv

gcloud container clusters get-credentials <your-cluster-name> --zone <zone>

kubectl get pods -n dev

# Or try creating a pod (if he has create permission)

kubectl run nginx --image=nginx -n dev

kubectl create deployment mynginx --image=nginx -n dev

A screenshot of a computer

AI-generated content may be incorrect.

A close up of words

AI-generated content may be incorrect.

**🔐 Best Practice (SRE Tip):**

* Use Role + RoleBinding ➜ When limiting users to a specific team namespace
* Use ClusterRole + ClusterRoleBinding ➜ For admin/devops who need cluster-wide access.