

[Docs](#)[Set Up](#)[Cloud Deploy](#)

GCP Deployment

Before You Start

This guide assumes that:

- You have already tried [Pachyderm locally](#), and have some familiarity with [Kubectl](#), [Helm](#), [Google Cloud SDK](#) and `jq`
- You have access to a Google Cloud account linked to an active billing account ([gcloud alpha billing accounts list](#))



Kubernetes & OpenShift Version Support

- **Kubernetes:** Pachyderm supports the three most recent minor [release versions](#) of Kubernetes. If your Kubernetes version is not among these, it is End of Life (EOL) and unsupported. This ensures Pachyderm users access to the latest Kubernetes features and bug fixes.
- **OpenShift:** Pachyderm is compatible with [OpenShift versions](#) within the “Full Support” window.

Configure Variables

- 1 Configure the following variables.

> Optional Variables for Scripting

- 2 Save to an `.env` file.
 - 3 Source them by inputting `source .env` into the terminal before starting the installation guide.
-

The following steps use a template to create a GKE cluster, a Cloud SQL instance, and a static IP address. The template also creates a service account for Pachyderm and Loki, and grants the service account the necessary permissions to access the Cloud SQL instance and storage buckets. You do not have to use this template, but it's a good outline for understanding how to create your own set up.

1. Create a New Project

- 1 Create a new project (e.g., `pachyderm-quickstart-project`).

You can pre-define the project ID using between 6-30 characters, starting with a lowercase letter. This ID will be used to set up the cluster and will be referenced throughout this guide.

```
gcloud projects create ${PROJECT_ID} --name=${PROJECT_ID} Copy E
gcloud alpha billing projects link ${PROJECT_ID} --billing-account=${
```

- 2 Enable the following APIs:

```
gcloud services enable container.googleapis.com Copy
gcloud services enable sqladmin.googleapis.com
gcloud services enable compute.googleapis.com
```

2. Create a Static IP Address

- 1 Create the static IP Address:

```
gcloud compute addresses create ${STATIC_IP_NAME} --re Copy ;{
```

2 Get the static IP address:

```
STATIC_IP_ADDR=$(gcloud compute addresses describe $ Copy C
```

3. Create a GKE Cluster

1 Create a GKE cluster with the following command:

```
gcloud container clusters create ${CLUSTER_NAME} \ Copy  
  --region=${GCP_REGION} \  
  --machine-type=${CLUSTER_MACHINE_TYPE} \  
  --workload-pool=${PROJECT_ID}.svc.id.goog \  
  --enable-ip-alias \  
  --create-subnetwork="" \  
  --logging=${LOGGING} \  
  --enable-dataplane-v2 \  
  --enable-shielded-nodes \  
  --release-channel="regular" \  
  --workload-metadata="GKE_METADATA" \  
  --enable-autorepair \  
  --enable-autoupgrade \  
  --disk-type="pd-ssd" \  
  --image-type="COS_CONTAINERD"
```

2 Grant your user account the privileges needed for the

helm install **to work properly:**

```
# By default, GKE clusters have RBAC enabled. To allow the Copy ↵  
# the requisite privileges via clusterrolebindings, you will need to gra  
# needed to create those clusterrolebindings.  
#  
# Note that this command is simple and concise, but gives your user  
# https://docs.pachyderm.io/en/latest/deploy-manage/deploy/rbac/ fc  
# Pachydermserviceaccount needs.  
kubectI create clusterrolebinding cluster-admin-binding --clusterrole
```

3 Connect to the cluster:

```
gcloud container clusters get-credentials ${CLUSTER_NAME} Copy ↵ gi
```

4. Create Storage Buckets

```
gsutil mb -l ${GCP_REGION} gs://${BUCKET_NAME} Copy  
gsutil mb -l ${GCP_REGION} gs://${LOKI_BUCKET_NAME}
```

5. Create a Cloud SQL Instance

- 1 Create a Cloud SQL instance with the following command:

```
gcloud sql instances create ${CLOUDSQL_INSTANCE_NAME} Copy
--database-version=POSTGRES_14 \
--cpu=${SQL_CPU} \
--memory=${SQL_MEM} \
--zone=${GCP_ZONE} \
--availability-type=ZONAL \
--storage-size=50GB \
--storage-type=SSD \
--storage-auto-increase \
--root-password=${SQL_ADMIN_PASSWORD}
```

- 2 Create a databases for Pachyderm and Dex:

```
gcloud sql databases create pachyderm -i ${CLOUDSQL_INSTANCE_NAME} Copy :E
gcloud sql databases create dex -i ${CLOUDSQL_INSTANCE_NAME}
```

- 3 Get the Cloud SQL connection name:

```
CLOUDSQL_CONNECTION_NAME=$(gcloud sql instances Copy)e
```

6. Create Service Accounts

Create a service account for Pachyderm and Loki.

```
gcloud iam service-accounts create ${GSA_NAME}
```

[Copy](#)

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \  
  --member="serviceAccount:${SERVICE_ACCOUNT}" \  
  --role="${ROLE1}"
```

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \  
  --member="serviceAccount:${SERVICE_ACCOUNT}" \  
  --role="${ROLE2}"
```

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \  
  --member=serviceAccount:${SERVICE_ACCOUNT} \  
  --role="${ROLE3}"
```

```
gcloud iam service-accounts add-iam-policy-binding ${SERVICE_ACCOL \  
  --role roles/iam.workloadIdentityUser \  
  --member "${PACH_WI}"
```

```
gcloud iam service-accounts add-iam-policy-binding ${SERVICE_ACCOL \  
  --role roles/iam.workloadIdentityUser \  
  --member "${PACH_WI}"
```

```
--member "${SIDECAR_WI}"
```

```
gcloud iam service-accounts add-iam-policy-binding ${SERVICE_ACCOUNT_NAME} \
  --role roles/iam.workloadIdentityUser \
  --member "${CLOUDSQLAUTHPROXY_WI}"
```

```
gcloud iam service-accounts create ${LOKI_GSA_NAME}
```

```
gcloud iam service-accounts keys create "${LOKI_GSA_NAME}-key.json"
```

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${LOKI_SERVICE_ACCOUNT}" \
  --role="${ROLE2}"
```

```
gcloud projects add-iam-policy-binding ${PROJECT_ID} \
  --member="serviceAccount:${LOKI_SERVICE_ACCOUNT}" \
  --role="${ROLE3}"
```

7. Create a Loki Secret

```
kubectl create secret generic loki-service-account --from-file=" Copy 39
```

8. Build a Helm Values File



Warning

Setting up [Authentication](#)?

Do not use mockIDP for clusters that will be deployed into production. If you do upgrade a cluster with mockIDP enabled, you must revoke the default mockIDP admin user by running the following command:

```
pachctl auth revoke --user kilgore@kilgore.trout
```

Copy

- 1 Create a `values.yaml` file, inserting the variables we've created in the previous steps:

> **Values.yaml**



Info

If your Postgres deployment requires SSL, you may need to update the parameters in the **global** section of your Helm Chart Values (HCVs)

- 2 Install using the following command:

```
helm repo add pachyderm https://helm.pachyderm.com  
helm repo update  
helm install pachyderm -f ./${NAME}.values.yaml pachyderm/pachyd
```

Copy

9. Connect to Cluster

You'll need your organization's cluster URL (proxy.host) value to connect.

- 1 Run the following command to get your cluster URL:

```
kubectl get services | grep pachyderm-proxy | awk '{print $4}'
```

Copy

- 2 Connect to your cluster:

HTTP

HTTPS (TLS)

```
pachctl connect grpc://<your-proxy.host-value>:80
```

Copy

3 You can optionally run `port-forward` to connect to console in your dashboard at `http://localhost:4000/` .

Azure Deployment

Console Setup