**Vectice Private Deployment for Kubernetes on GCP Installation & Integration Guide**

(Version 234.8.2)

**Table of Contents**

[**Requirements**](#_fnjk6vnqzbqk) **3**

[**Deliverables**](#_mqlyoi33hzb5) **3**

[**Architecture**](#_chto247rp9sq) **4**

[Kubernetes cluster creation](#_h55v6u6x57ma) 5

[Setup](#_s6x4yrccpcji) 10

[Vectice namespace](#_mxuozpchkny) 10

[Kubernetes vectice’s docker and helm registry](#_ilgx03nxlki1) 10

[TLS ingress secret  
If you don’t have the certificate of your own](#_ypem02sy50h1) 11

[Install cert-manager and csi-driver (only if they are not installed on the cluster)](#_4723uuqtbcbv) 12

[Generate a custom certificate authority and create its associated secret](#_iqc1gqhvdlnn) 12

[Activating the service account](#_m5zcuw3rs30y) 12

[Install the Vectice stack](#_1lv79sue5wq) 13

[**Appendix 1: Authentication with gcloud with serviceaccount.**](#_umhc324a9cve) **13**

**Vectice**

short line

# Requirements

You should receive an archive link to download. If you did not receive an archive link, contact us at [support@vectice.com](mailto:support@vectice.com).

### **Cloud Infrastructure**

* Kubernetes cluster v1.16+ deployed, 3 nodes with e2-standard-4
* External PostgreSQL instance: with databases vectice and keycloak
* Google Cloud Bucket on client Google storage

### **Network Environment Requirements**

1. GCP
   1. Recommended:
      1. VPC with access to Public Vectice API libraries for PyPi
      2. Same VPC for both Vectice server and user client
      3. Security Groups - Port 443, 3128 Outbound, SMTP Port\*\*
      4. Same VPC for Vectice server and Database
   2. Certificate for SSL (No self-signed certificates supported)
   3. Domain name or static IP (e.g.: <https://vectice.my-company.com>)

\*\*Security Groups  
- Port 443 for app’s HTTPS access  
- Port 3128 for pip install  
- SMTP port (e.g 2525)

* TLS certificates for ingresses are generated
* Service Account from client GCP project who has permissions to read/write the Google Cloud Bucket on client Google storage (JSON key file will be used in the configuration)
* Provided by Vectice: Service account is created and has the permission to pull Vectice Docker images and Helm charts, for example with Roles Artifact Registry Reader and Container Registry Service Agent like on example below
* Access to https://docs.vectice.com/



* computer/server with access to GCP and cluster through the command-lines tools listed below

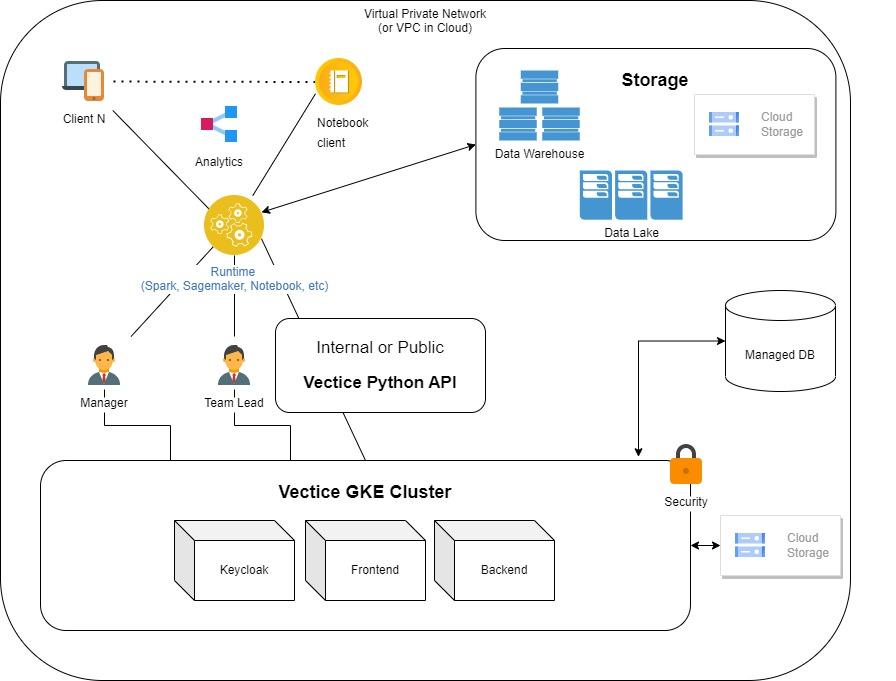
Command-lines tools:

* Helm v3+ <https://helm.sh/docs/intro/install/>
* Kubectl <https://kubernetes.io/docs/tasks/tools/>
* Gcloud <https://cloud.google.com/sdk/docs/install>
* Gsutil <https://cloud.google.com/storage/docs/gsutil_install>
* Openssl

# Deliverables

* helm vectice chart
* JSON file of Service Account which has the permission to pull Vectice image
* script to encode easily in base64 the Client Service Account

# Architecture



Vectice’s GKE Cluster could be directly placed in the customer’s private environment with direct connection or through a proxy server connected to the public Vectice SDK libraries for Python client integration.

For an air-gapped environment with no Internet access in the customer’s environment, Vectice can provide an SDK tarball to be placed in the customer’s VPN (VPC).

# Infrastructure Creation through terraform

You can find our terraform scripts in [our public repository](https://github.com/vectice/vectice-gke-marketplace), along with the instructions.  
After the creation, go to the Deployment [section.](#_gf16gdbwsbt7)Or you can follow the instruction in this document

# Infrastructure Creation through this User Guide

## PostgreSQL Instance creation

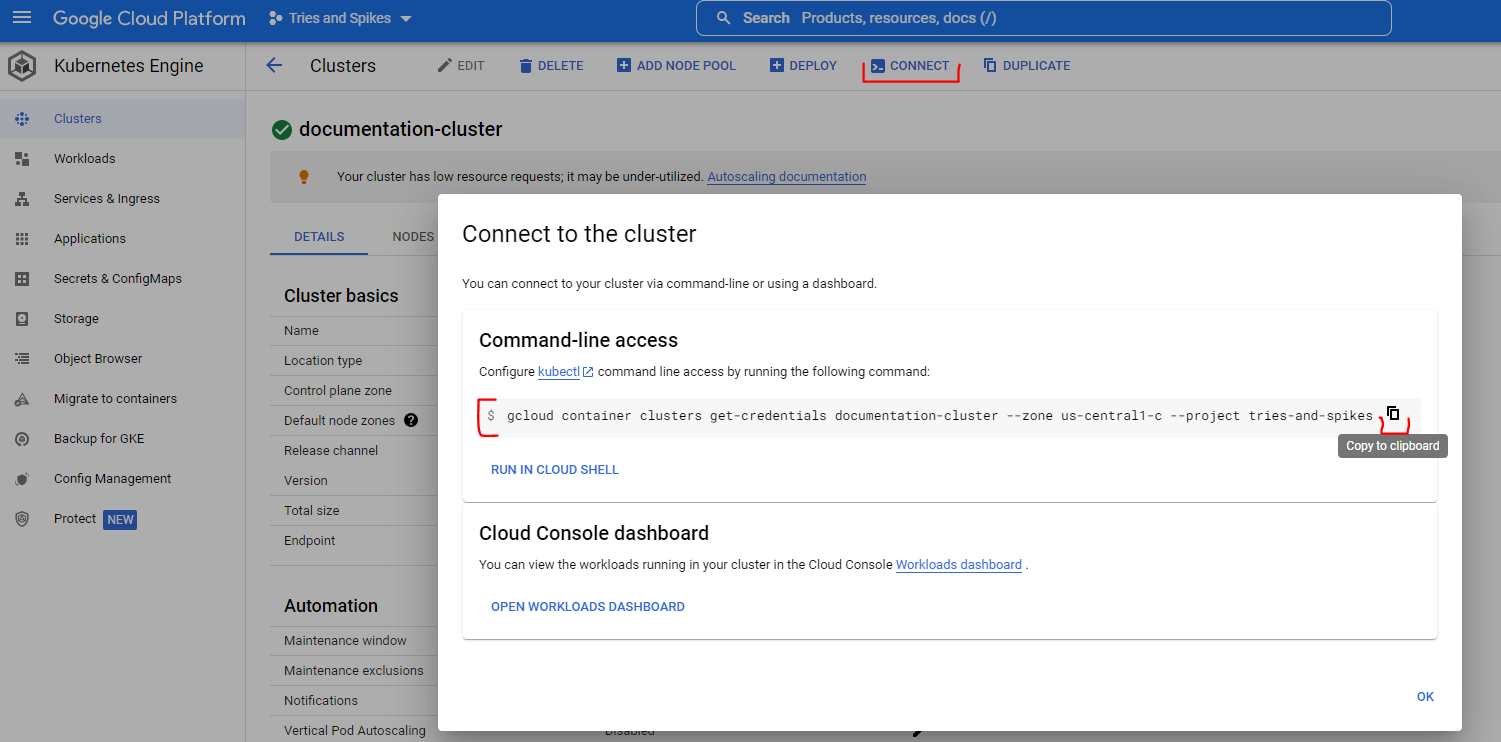
[See Appendix 1: Creating the SQL Instance](#_cjhnek5bd0t4)

## Bucket creation [See Appendix 2: Creating the GCS Bucket](#_opp616jbvjzz)

## Kubernetes cluster creation

[See Appendix 3: Cluster Creation](#_k06jjylf33to)

You can then go to the Cluster, and connect to it



# 

# Deployment

| **gcloud container clusters get-credentials documentation-cluster --zone us-central1-c --project tries-and-spikes**  # Fetching cluster endpoint and auth data.  # kubeconfig entry generated for documentation-cluster. |
| --- |

We’ll then set the context and set it as a variable

| **kubectl config get-contexts | grep '\*'**  \* tries-and-spikes\_us-central1-c\_documentation-cluster tries-and-spikes\_us-central1-c\_documentation-cluster tries-and-spikes\_us-central1-c\_documentation-cluster **CONTEXT=gke\_tries-and-spikes\_us-central1-c\_documentation-cluster** |
| --- |

You can then list the namespaces

| **kubectl --context $CONTEXT get namespaces**  NAME STATUS AGE  default Active 28m  kube-node-lease Active 28m  kube-public Active 28m  kube-system Active 28m |
| --- |

## 

## 

## 

## Setup of environment

### Vectice namespace

Create Vectice namespace where applications will be deployed.

| **kubectl --context $CONTEXT create namespace vectice** |
| --- |

### Kubernetes vectice’s docker and helm registry

Go in the directory where you have the JSON file provided by the Vectice team.

| **${PROJECT}-${CLIENT}-docker-image-puller.json** |
| --- |

Run the following commands to create the secret:

| **PROJECT="vectice-public"**  **CLIENT="<CLIENT>"**  **FILE="${PROJECT}-${CLIENT}-docker-image-puller.json"** |
| --- |

In order to be able to pull the helm charts:

| **kubectl --context $CONTEXT create secret docker-registry vectice-gcr-secrets -n vectice \**  **--docker-server=gcr.io \**  **--docker-username=\_json\_key \**  **--docker-password="$(cat $FILE)" \**  **--docker-email=$CLIENT-docker-image-puller@$PROJECT.iam.gserviceaccount.com** |
| --- |

### TLS ingress secret If you don’t have the certificate of your own

| **openssl req -x509 -nodes -newkey rsa:2048 -days 3650 -keyout /tmp/vectice-cert.key -out /tmp/vectice-cert.crt -subj "/CN=demo.vectice.com"** |
| --- |

# If you have your own certificate, you need to name them vectice-cert.crt and vectice-cert.key and store them in /tmp in order to install them with the command below.  
# Considering you have the certificate and its associated key in /tmp directory.

| **kubectl --context $CONTEXT create secret tls vectice-private-https -n vectice --cert=/tmp/vectice-cert.crt --key=/tmp/vectice-cert.key** |
| --- |

### Install cert-manager and csi-driver (only if they are not installed on the cluster)

| **helm --kube-context $CONTEXT repo add jetstack https://charts.jetstack.io**  **helm --kube-context $CONTEXT repo update**  **helm --kube-context $CONTEXT install cert-manager jetstack/cert-manager -n cert-manager --create-namespace --set installCRDs=true**  **helm --kube-context $CONTEXT install cert-mananager-csi-driver jetstack/cert-manager-csi-driver --create-namespace -n cert-manager** |
| --- |

### Generate a custom certificate authority and create its associated secret

| **openssl req -x509 -nodes -newkey rsa:4096 -days 3650 -keyout /tmp/ca.key -out /tmp/ca.crt -subj '/CN=vectice-internal-ca' -addext "keyUsage = keyCertSign"**  **kubectl --context $CONTEXT create secret tls vectice-internal-ca -n vectice --cert=/tmp/ca.crt --key=/tmp/ca.key** |
| --- |

### Activating the service account

| **gcloud auth activate-service-account $CLIENT-docker-image-puller@$PROJECT.iam.gserviceaccount.com --key-file=$FILE**  Activated service account credentials |
| --- |

If you have issues with kubectl/helm commands after activating the service account, you have to logout and login again, as described in [Appendix 4](#_umhc324a9cve)

## Install the vectice stack

### Through the marketplace

Fill in the fields in the deployment form and press deploy

### Through helm, from the git repository

Define the version you want to deploy and set it as a variable  
  
Clone the public GitHub repository [vectice-gke-marketplace](https://github.com/vectice/vectice-gke-marketplace)

Clone it and position yourself on the version you want to deploy

| **git clone** [**git@github.com**](mailto:git@github.com)**:vectice/vectice-gke-marketplace.git git checkout v$VERSION** |
| --- |

Copy the values file and fill in the values according to your environment deployment

| **cp chart/vectice/values.yaml chart/vectice/myvalues.yaml** |
| --- |

Vectice installation workflow for SaaS version is set to deploy all components in one time

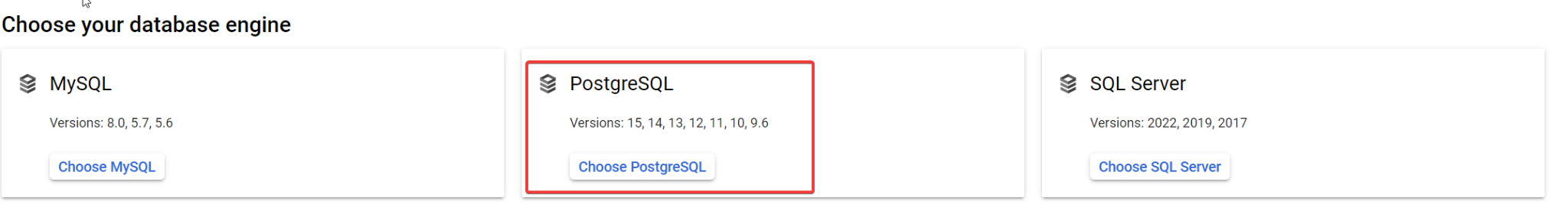
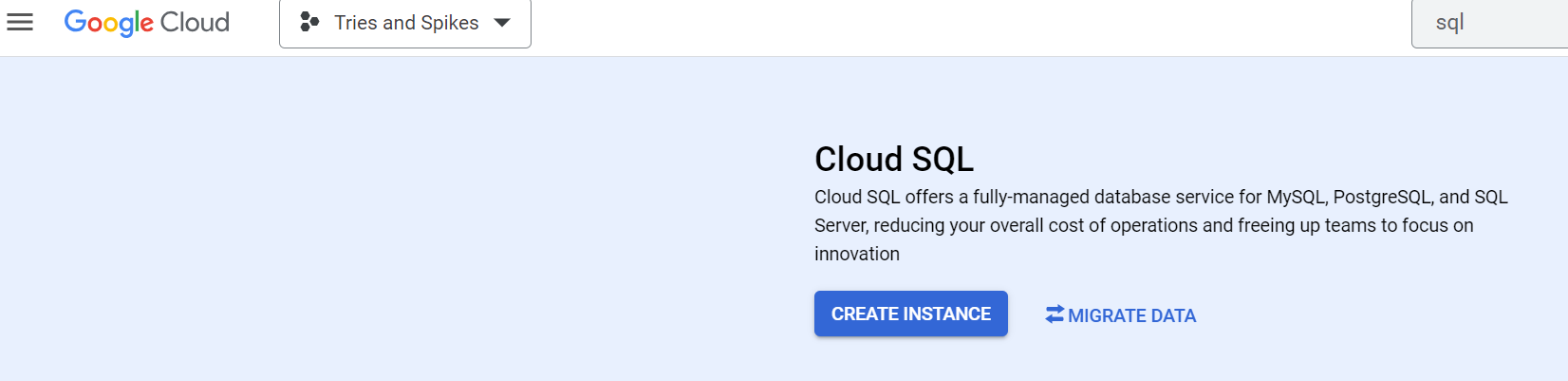
* Position yourself in the chart directory and run the helm deployment command

| **cd chart helm --kube-context $CONTEXT upgrade --install vectice vectice -f vectice/myvalues.yaml -n vectice --create-namespace --wait** |
| --- |

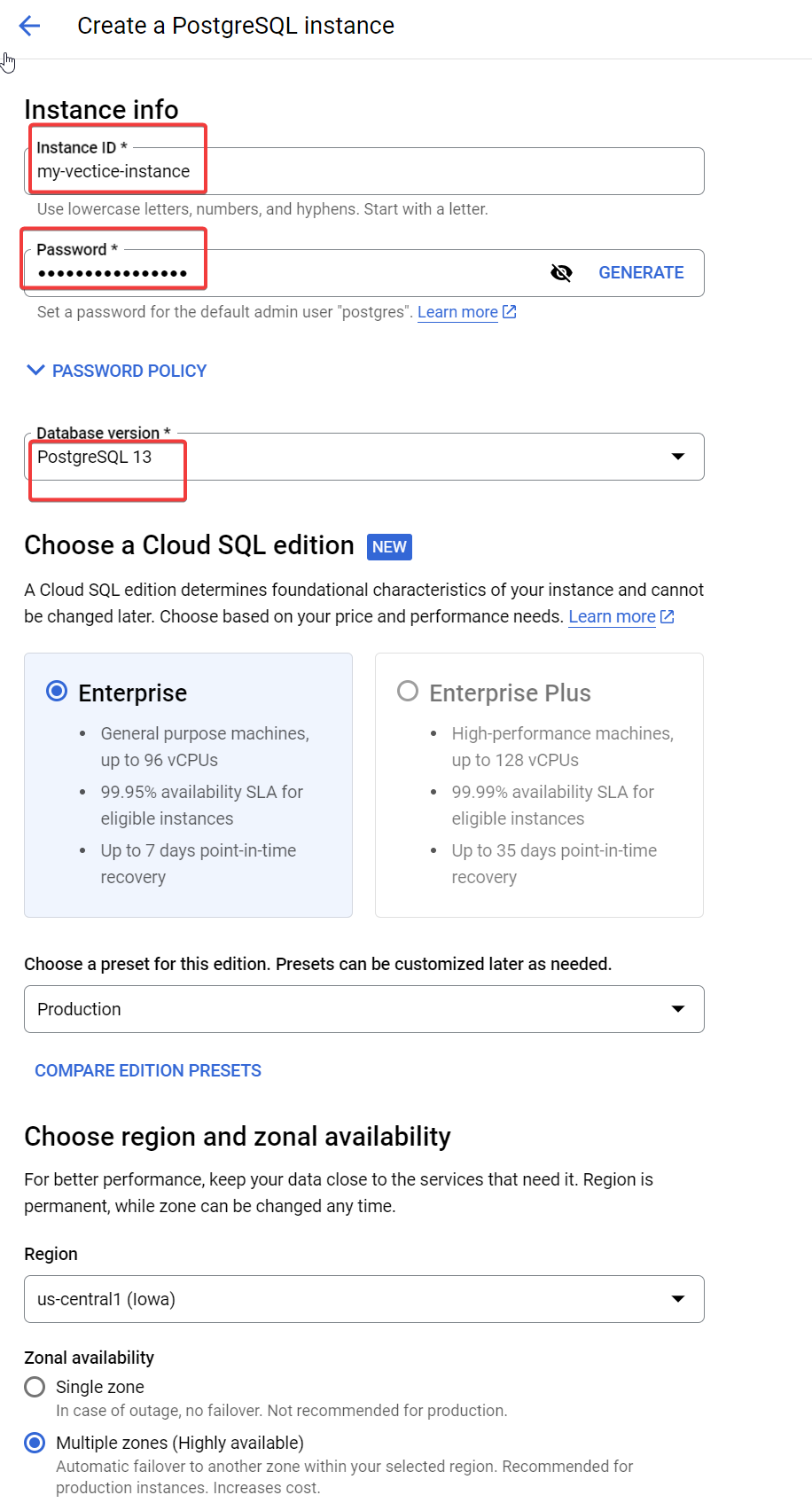
***Please reach out to*** [***support@vectice.com***](mailto:support@vectice.com) ***if you have any questions or comments.***

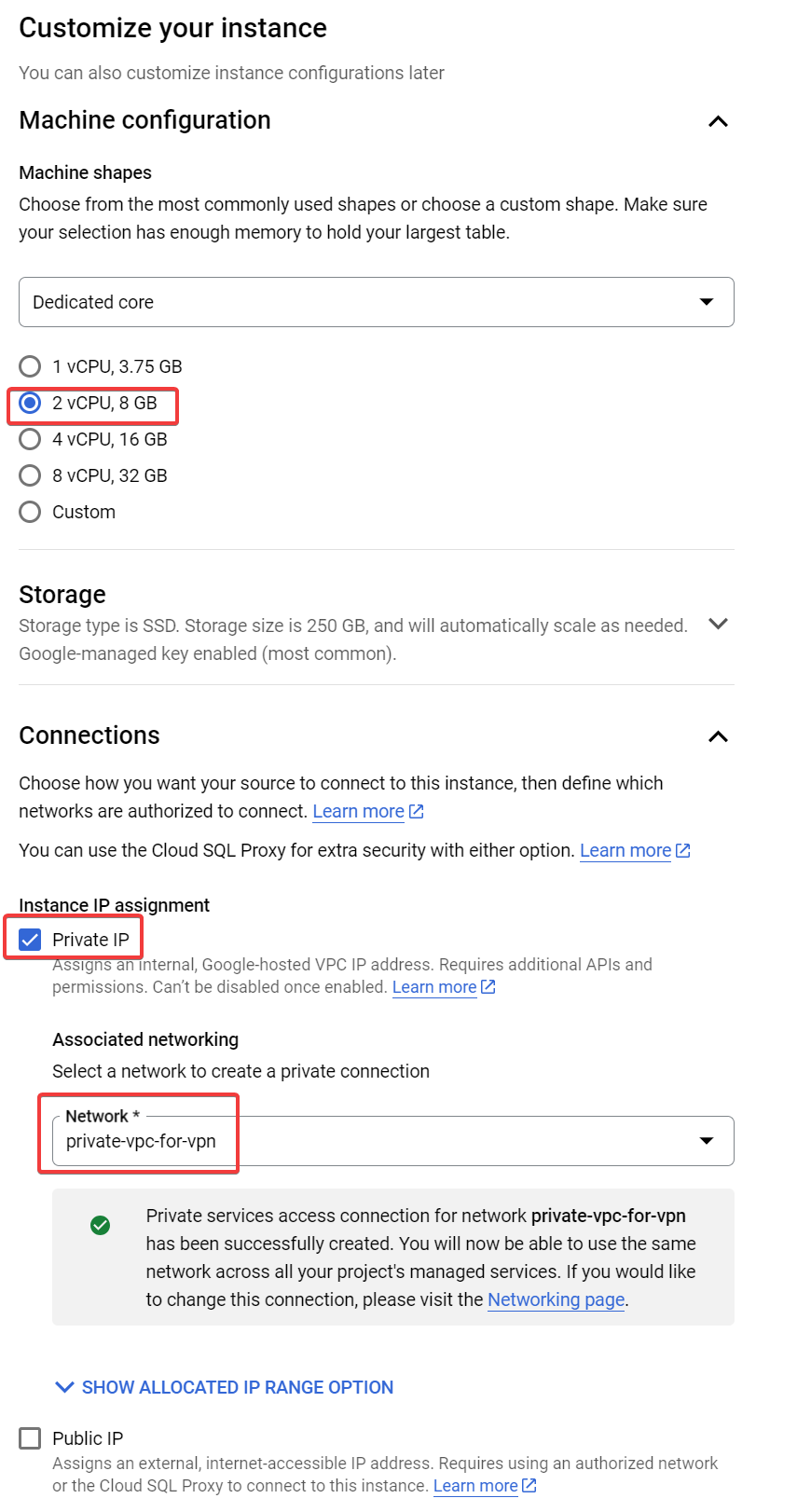
# 

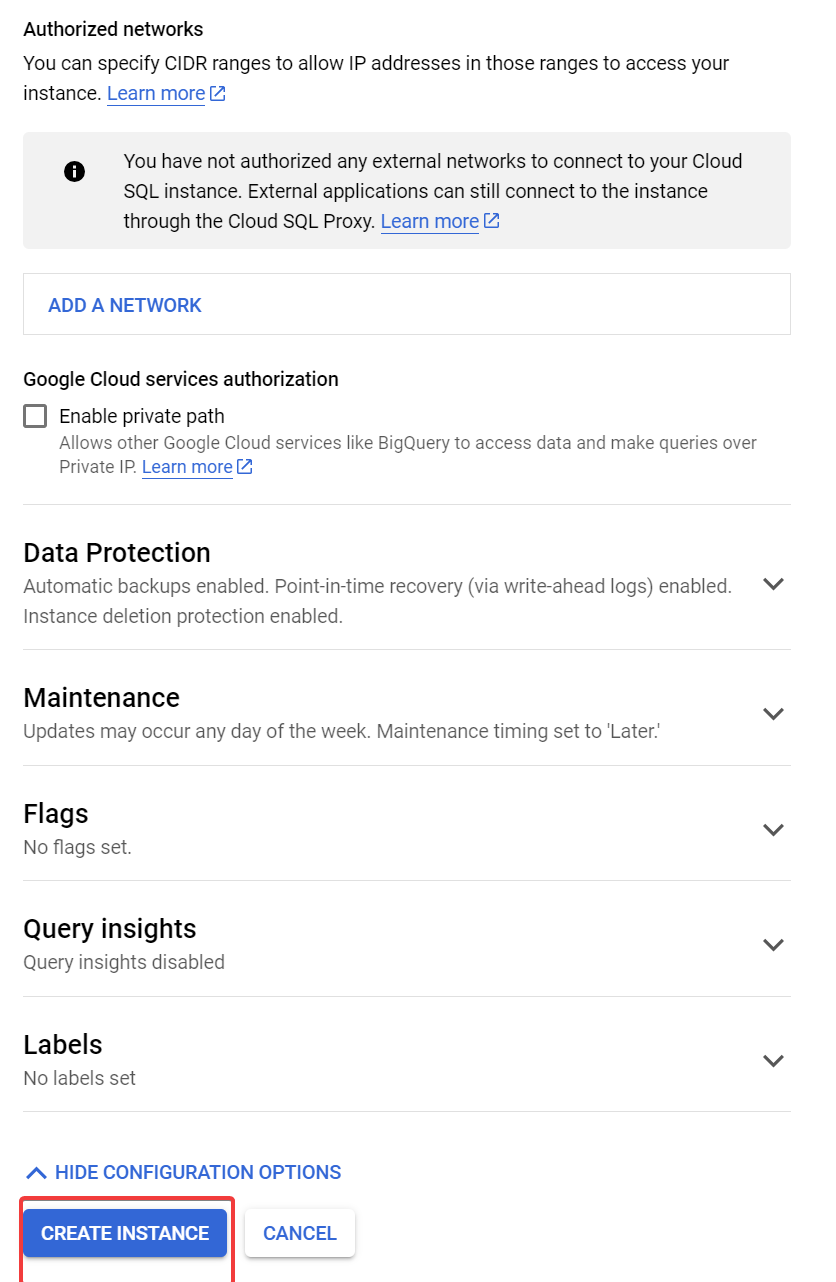
# Appendix 1: Creating the SQL instance

Go the SQL menu, and press CREATE INSTANCE

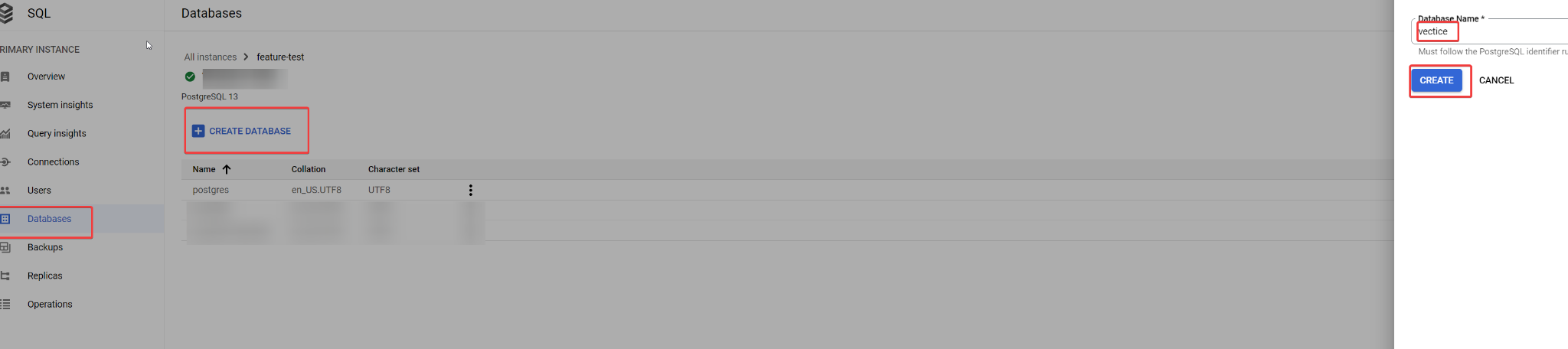
We’ll then input the instance name, the master user, and password





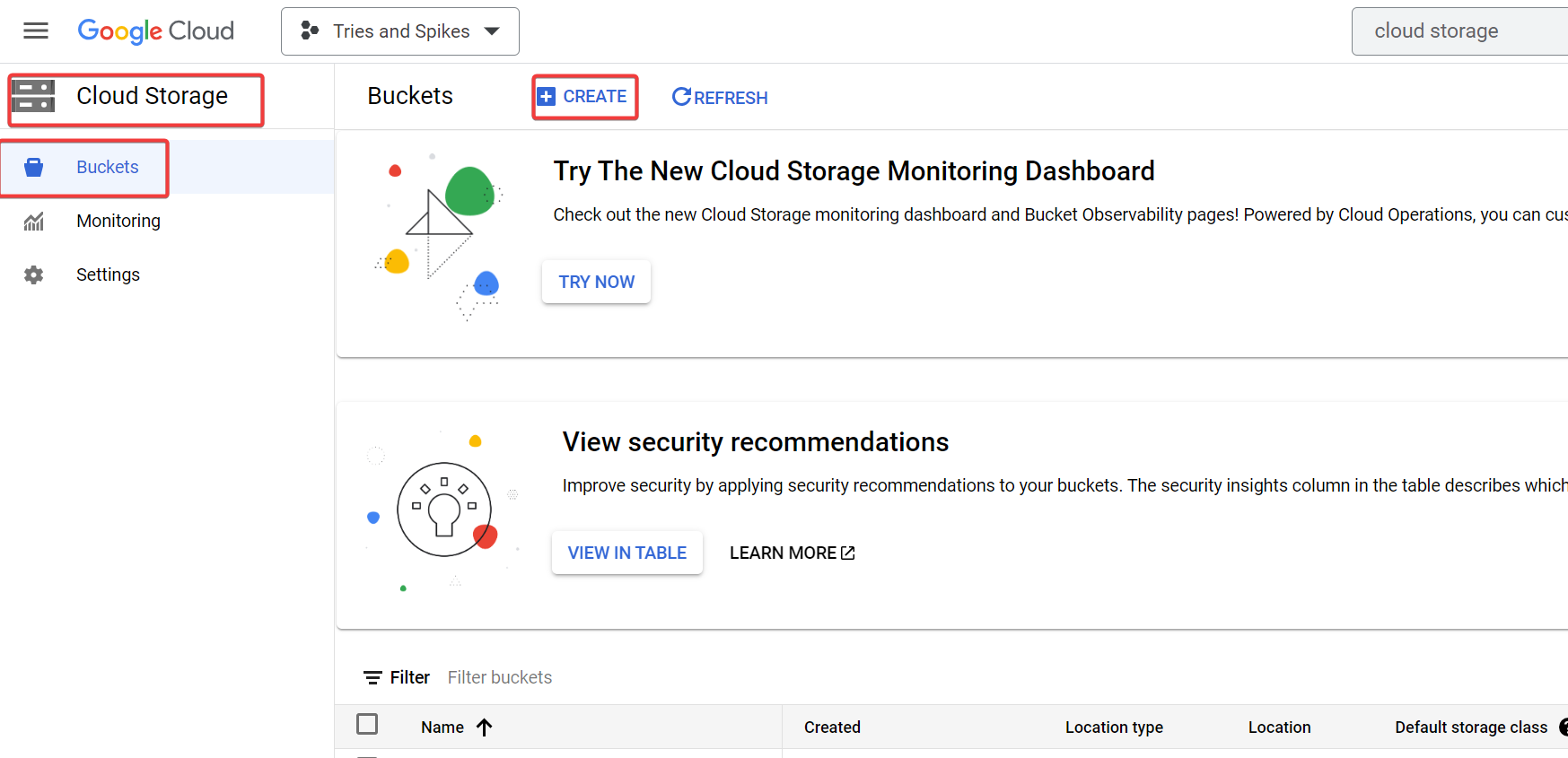


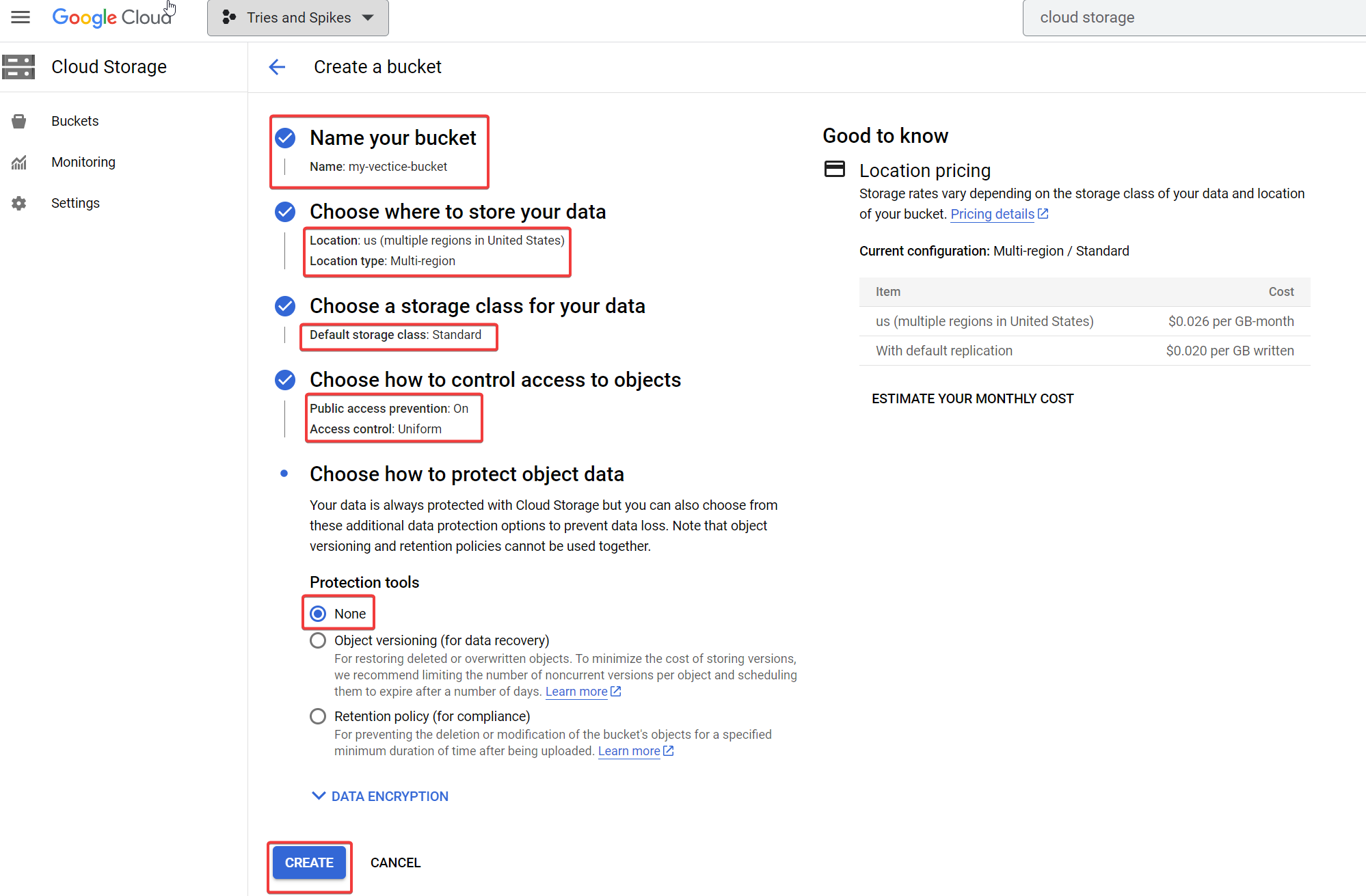
We’ll then create the two databases, vectice and keycloak  
From the menu “Databases”, CREATE DATABASE, we create vectice, then redo the operation for keycloak



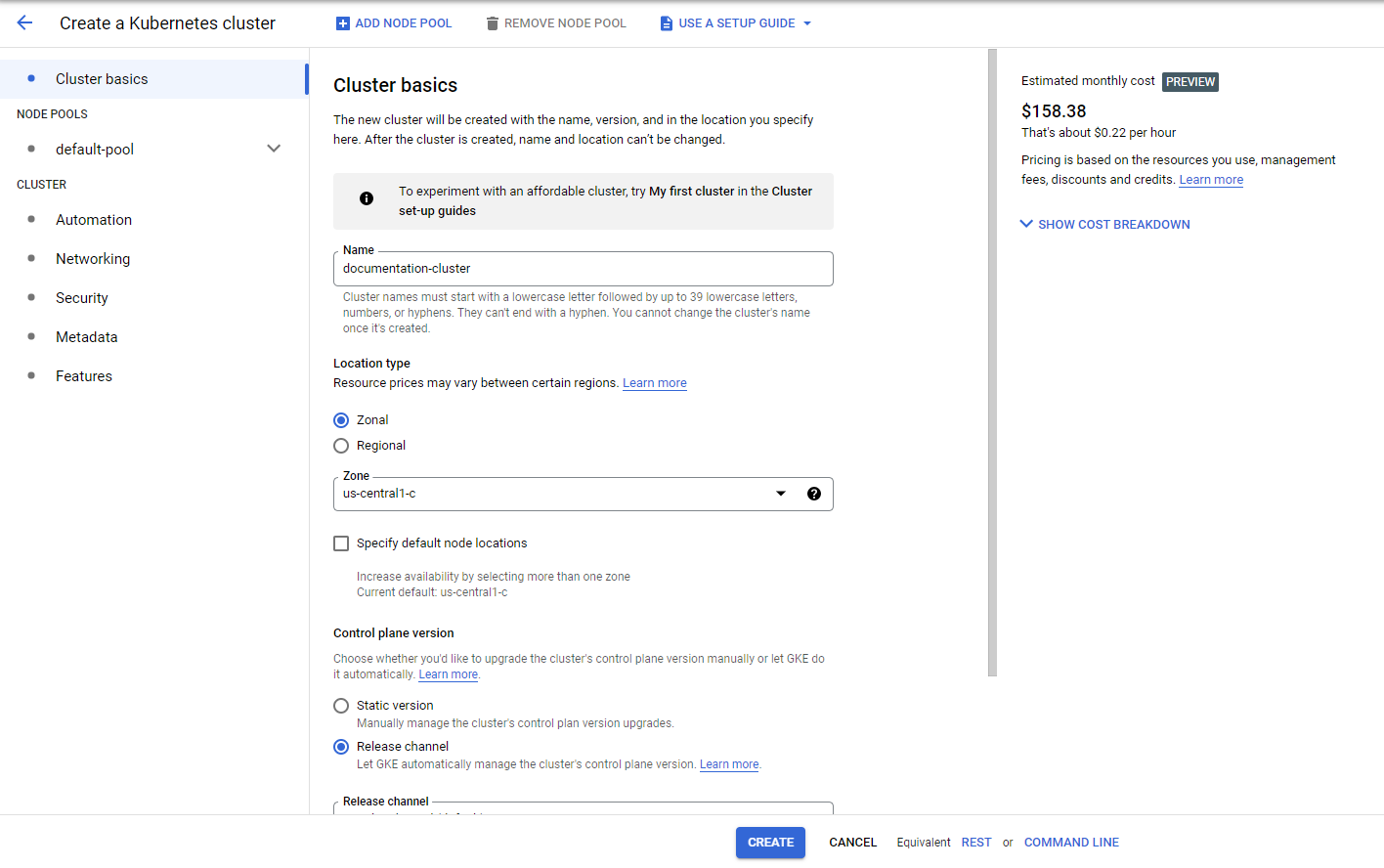
# Appendix 2: Bucket Creation

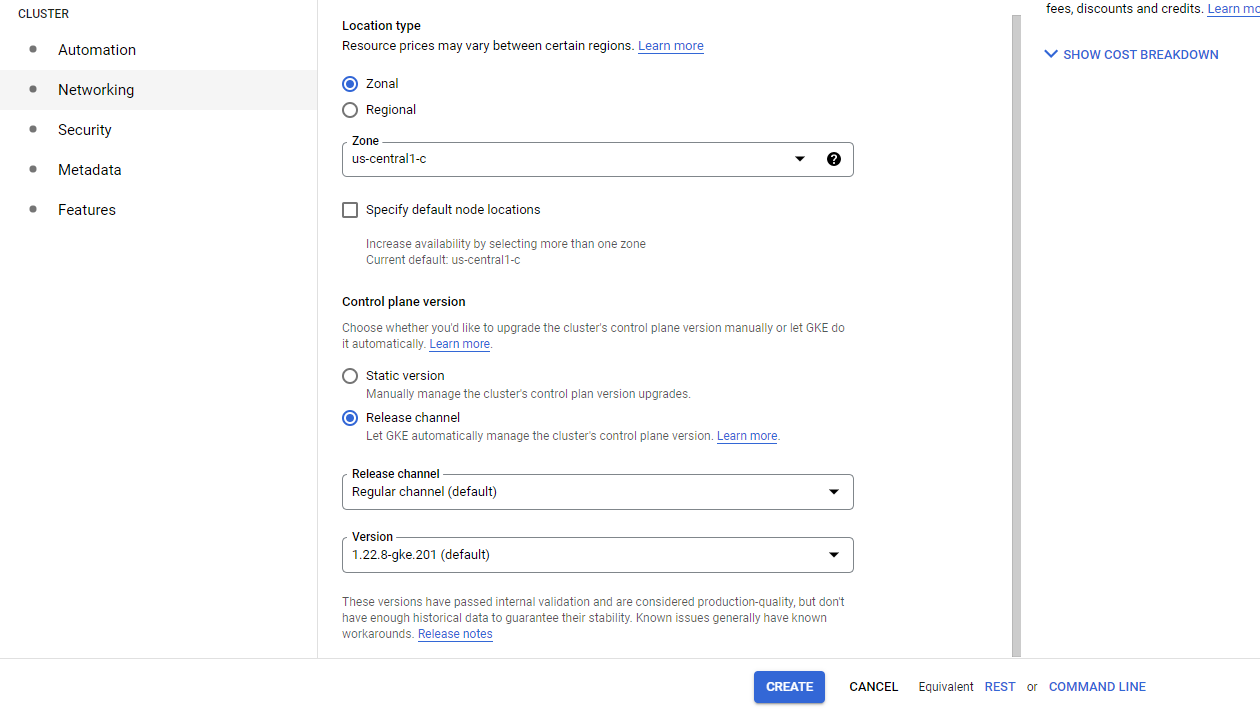
Go to Cloud Storage, Buckets menu, and Press CREATE

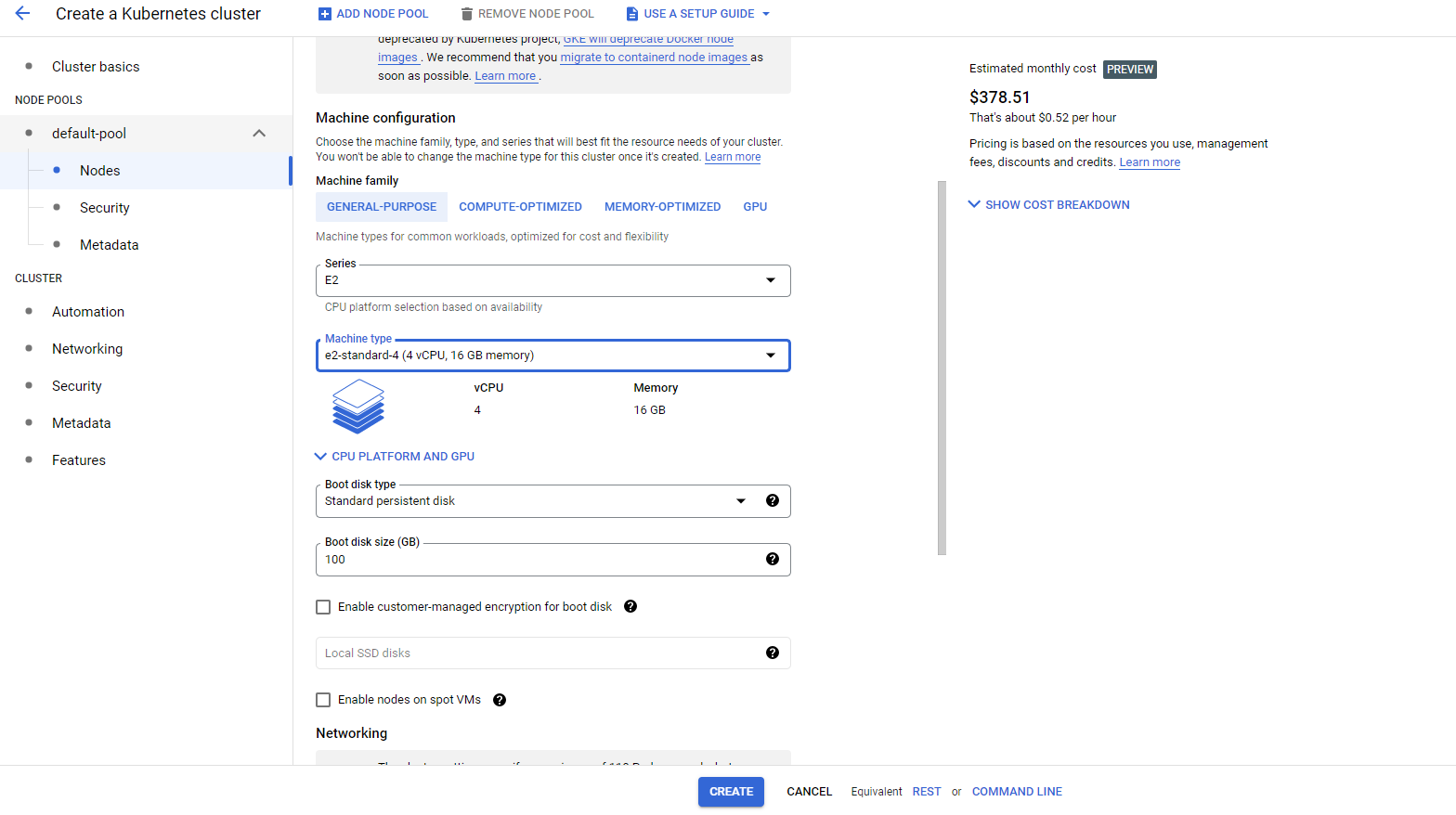
  
Fill in the settings as on screenshot below

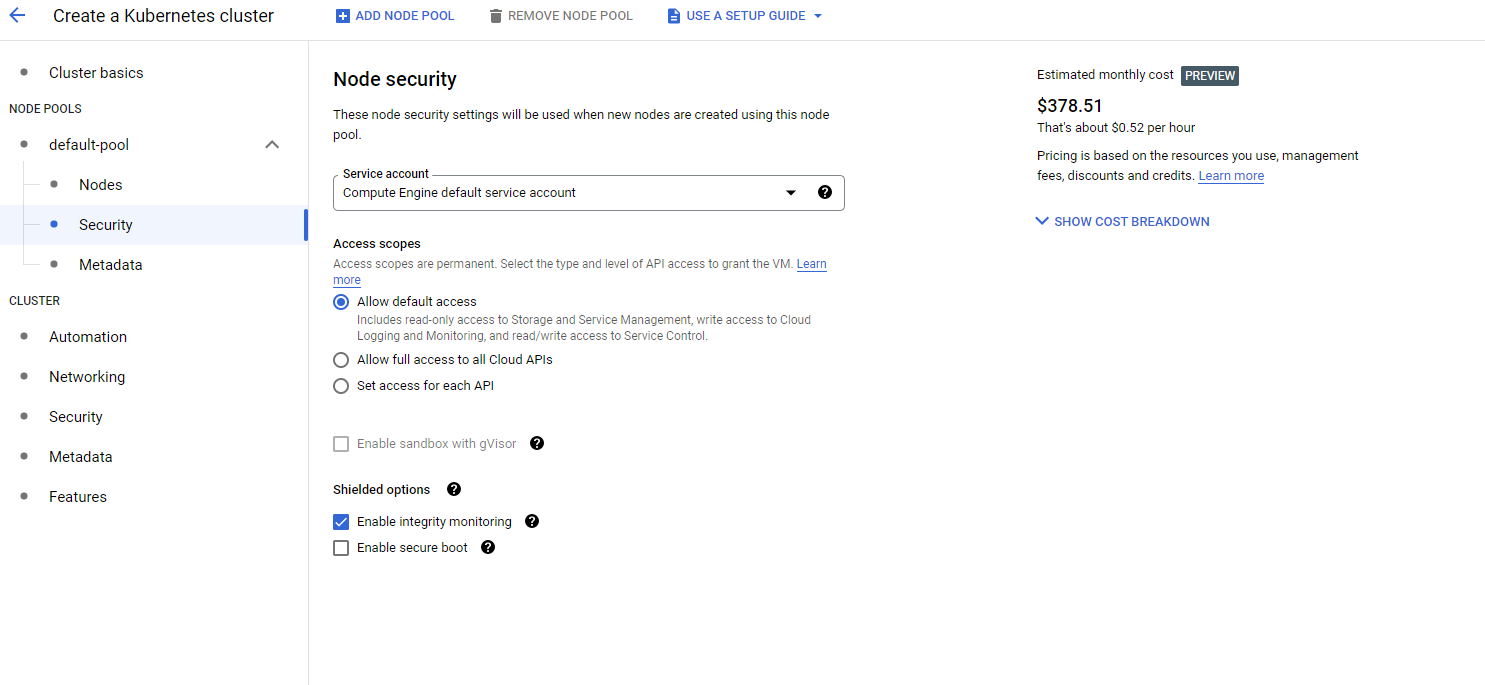


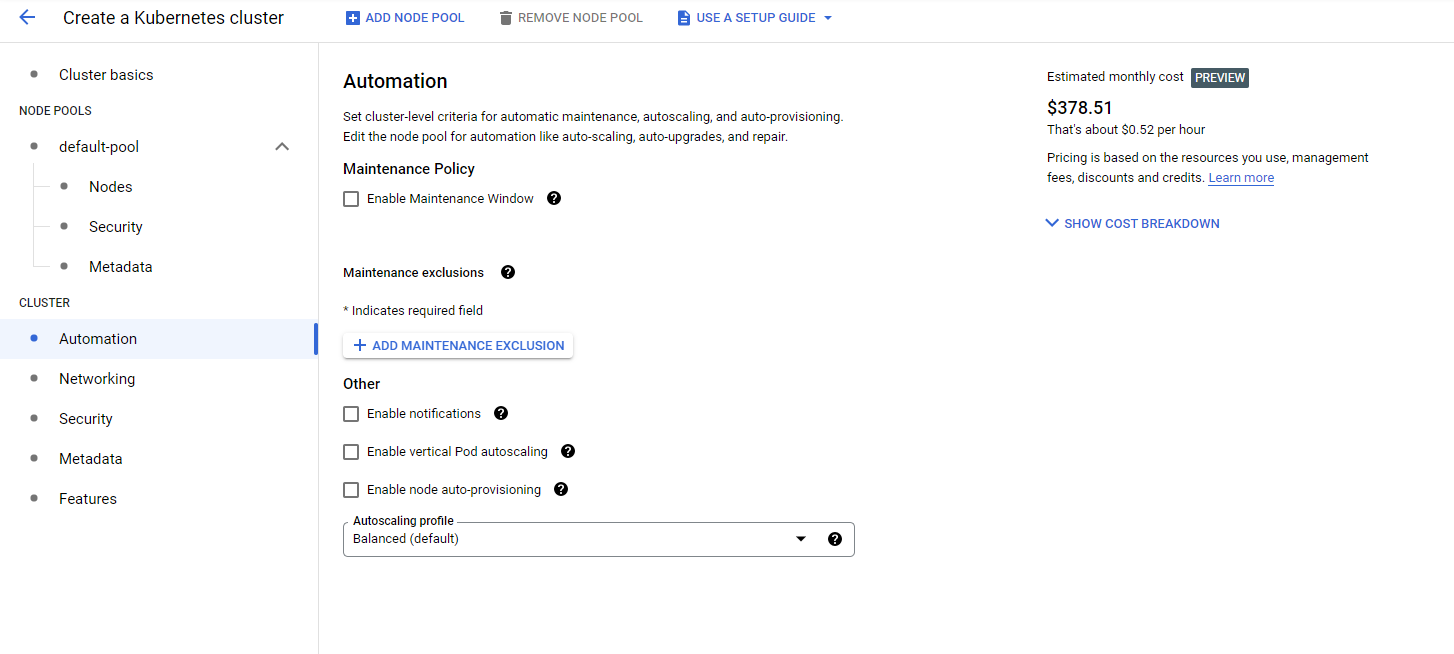
# Appendix 3: Cluster Creation

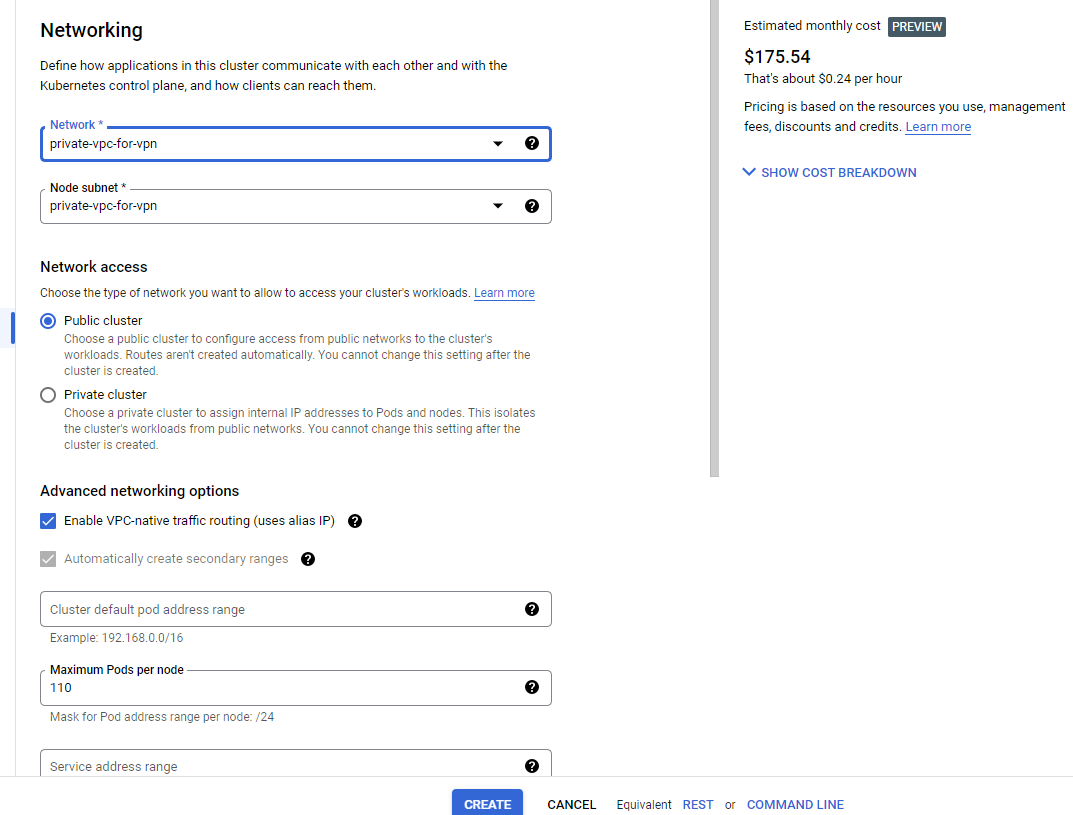












Once you push the create button, there will be a period of 30 min of creation of Cluster, Control Plan, NodePool, and Nodes.

# 

# Appendix 4: Authentication with gcloud with serviceaccount.

| **gcloud auth activate-service-account $CLIENT-docker-image-puller@$PROJECT.iam.gserviceaccount.com --key-file=$FILE**  Activated service account credentials |
| --- |

This is about getting connected with usual IAM account vs getting connected with a service account, and using kubectl/helm commands after getting connected with a service account.

Solution, and more context:

You need to revoke the service account authentication and re-authenticate to the cluster

| **gcloud auth list**  **gcloud config set account**  **gcloud container clusters get-credentials <my-cluster> --zone <clusters-zone>"** |
| --- |

source: <https://stackoverflow.com/questions/60330165/how-to-easily-switch-gcloud-kubectl-credentials>