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

(Version 234.8.2)

**Table of Contents**

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

[Cloud Infrastructure 3](#_j55aqvm3h5ep)

[Network Environment Requirements 3](#_e9vylsnysmfk)

[**Deliverables 4**](#_mqlyoi33hzb5)

[**Architecture 5**](#_chto247rp9sq)

[**Infrastructure Creation through terraform 6**](#_osi6luonpr8l)

[**Infrastructure Creation through this User Guide 6**](#_yo3cntos4hdy)

[PostgreSQL Instance creation 6](#_rk7rhripkwu4)

[Bucket creation 6](#_kps4vs2t50mm)

[Kubernetes cluster creation 6](#_h55v6u6x57ma)

[**Deployment 8**](#_gf16gdbwsbt7)

[Setup of environment 9](#_s6x4yrccpcji)

[Vectice namespace 9](#_mxuozpchkny)

[TLS ingress secret  
If you don’t have the certificate of your own 9](#_ypem02sy50h1)

[Install cert-manager and csi-driver (only if they are not installed on the cluster) 9](#_4723uuqtbcbv)

[Generate a custom certificate authority and create its associated secret 10](#_iqc1gqhvdlnn)

[Install the vectice stack 11](#_srjzjzw8ue5v)

[Through the marketplace 11](#_zgczmo35tz6w)

[Through helm, from the git repository 11](#_skewvyz16upb)

[**Appendix 1: Creating the SQL instance 12**](#_cjhnek5bd0t4)

[**Appendix 2: Bucket Creation 16**](#_opp616jbvjzz)

[**Appendix 3: Cluster Creation 17**](#_k06jjylf33to)

**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)
* Access to https://docs.vectice.com/
* computer/server with access to GCP and cluster through the command-lines tools listed below

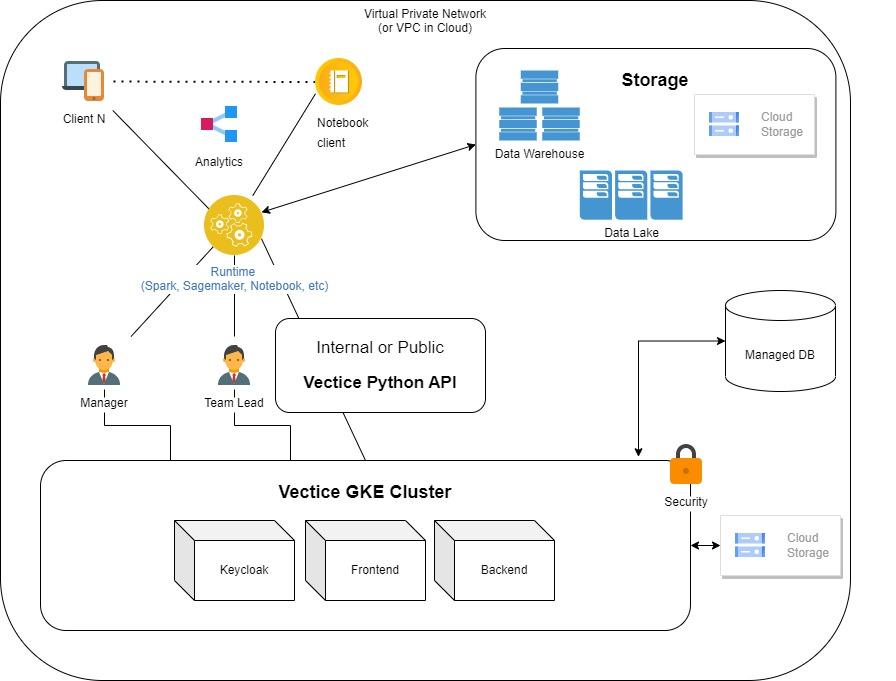
Command-lines tools:

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

# Deliverables

* helm vectice chart
* 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

Two ways to create the infrastructure necessary for running Vectice, through terraform or manually.  
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 go to the next section [Infrastructure Creation through this User Guide](#_yo3cntos4hdy)

# Infrastructure Creation through this User Guide

## PostgreSQL Instance creation

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

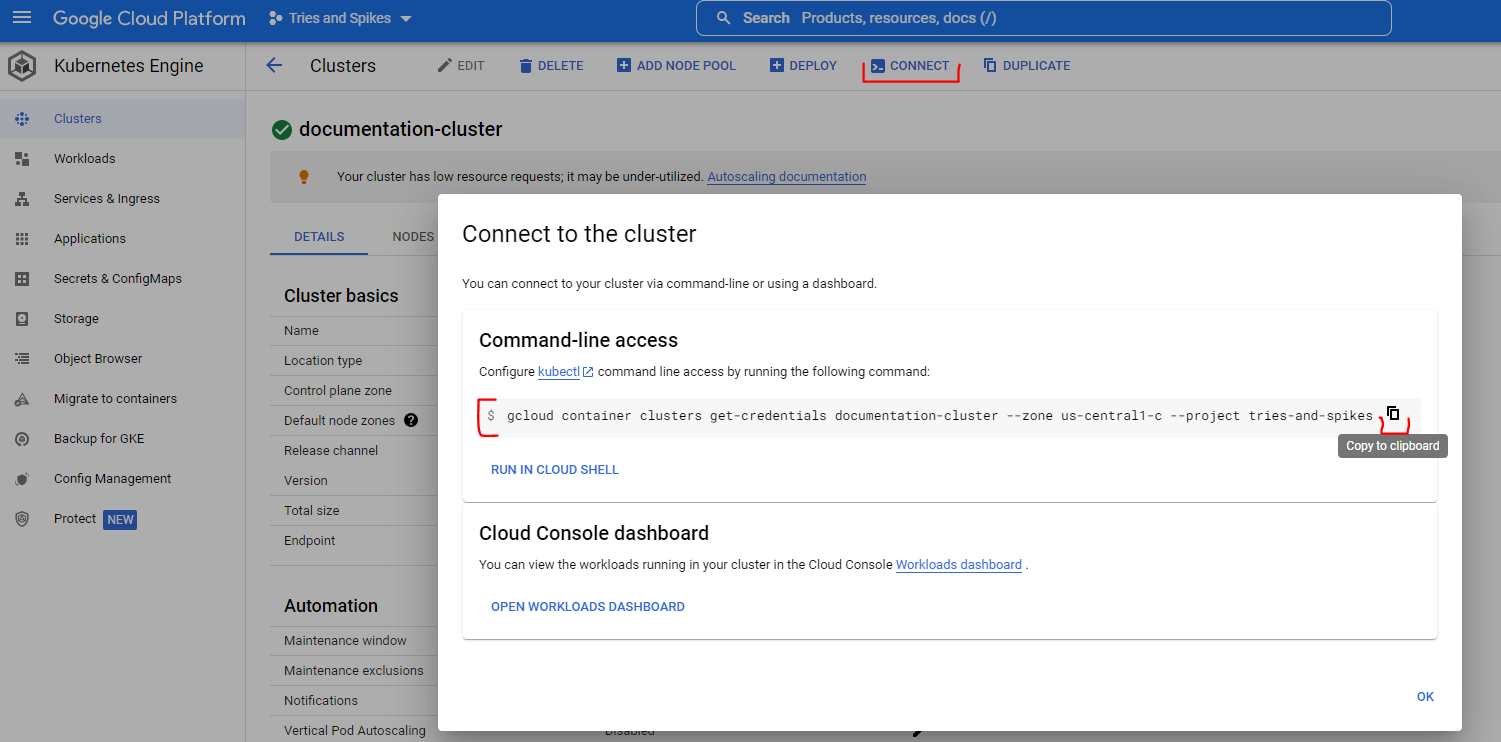
## Bucket creation

[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. |
| --- |

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** |
| --- |

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** |
| --- |

### 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"** |
| --- |

### 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"** |
| --- |

## 

## 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** |
| --- |

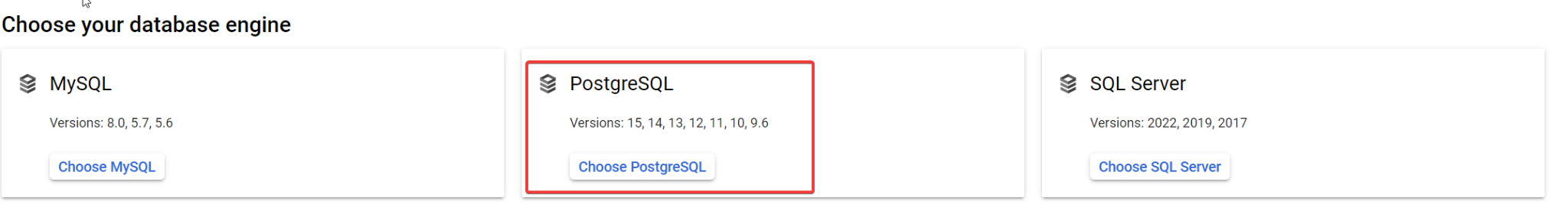
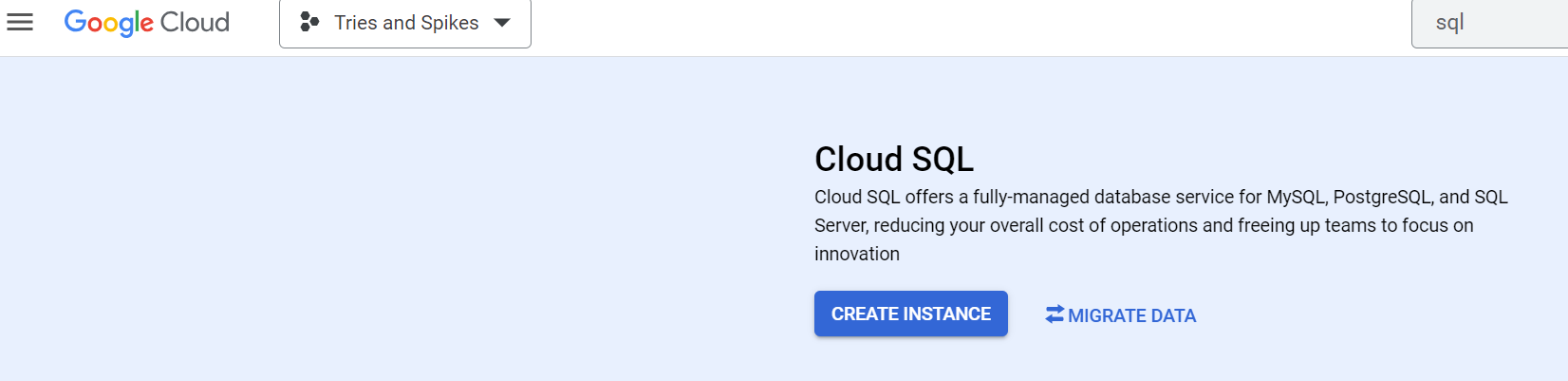
Vectice installation workflow for SaaS version is set to deploy all components simultaneously.

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

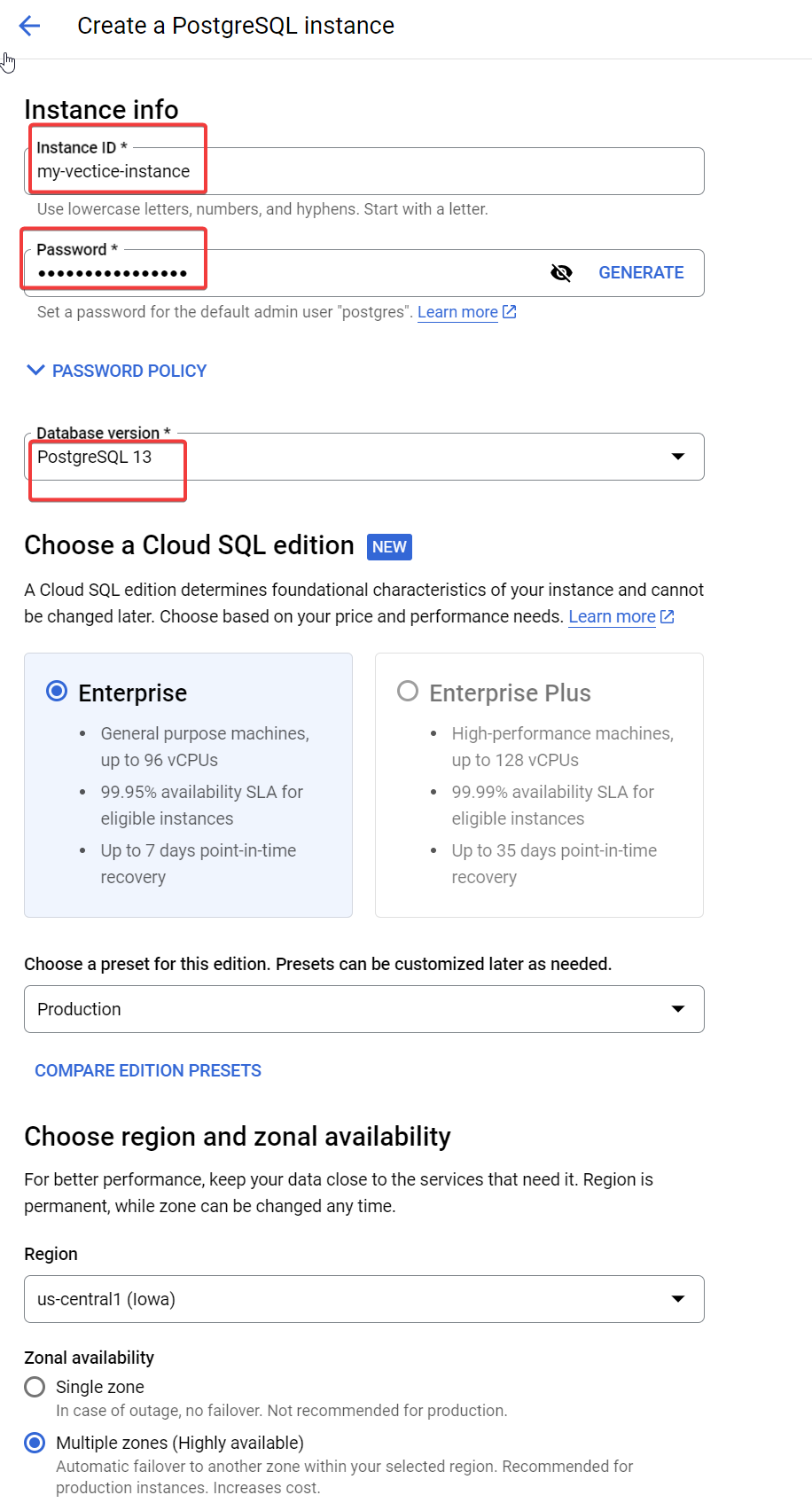
| **cd chart helm --kube-context $CONTEXT upgrade --install vectice vectice -f vectice/myvalues.yaml -n vectice --create-namespace --wait** |
| --- |

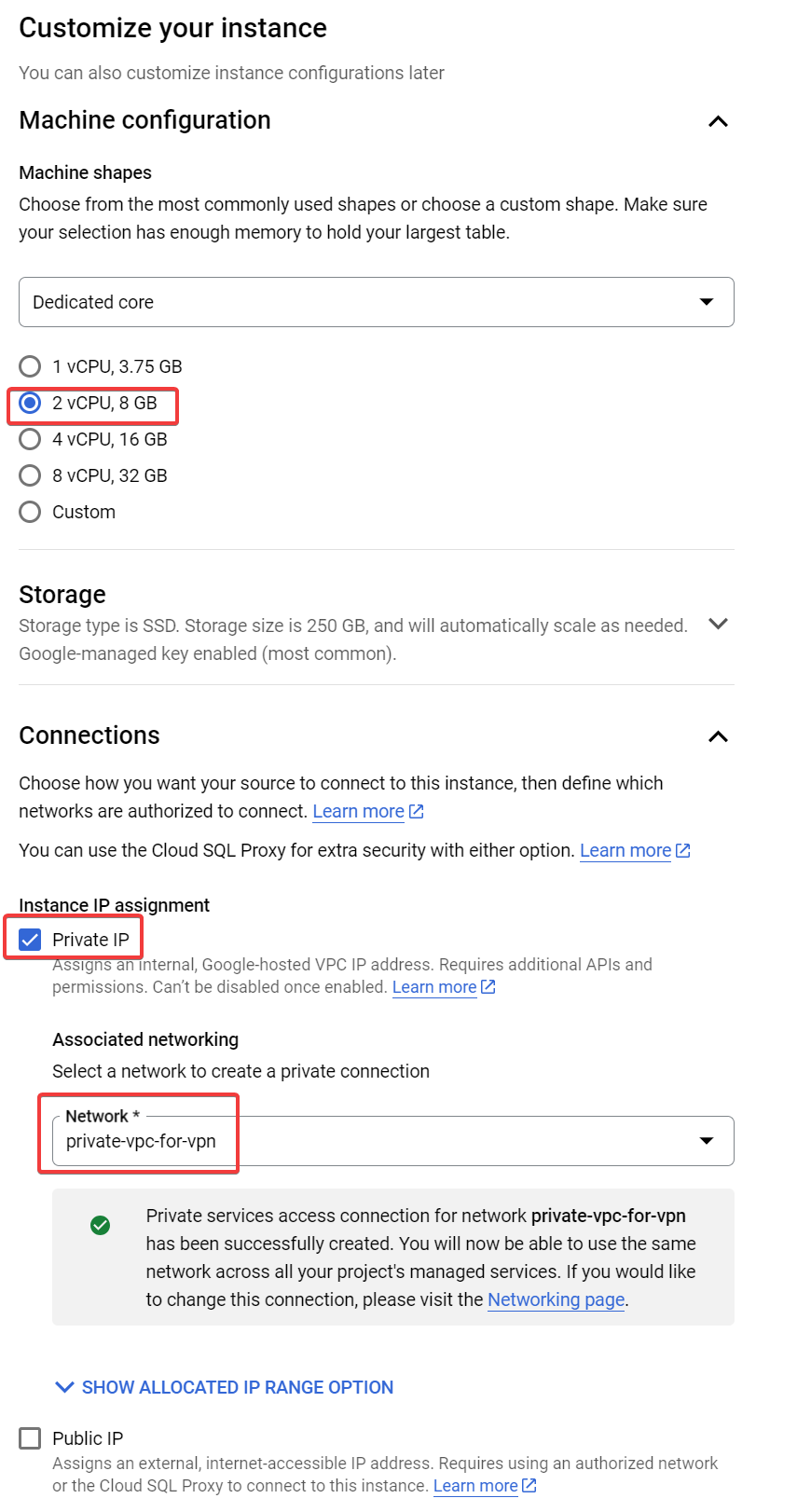
***Please reach out to*** [***support@vectice.com***](mailto:support@vectice.com) ***if you have any questions or comments.***

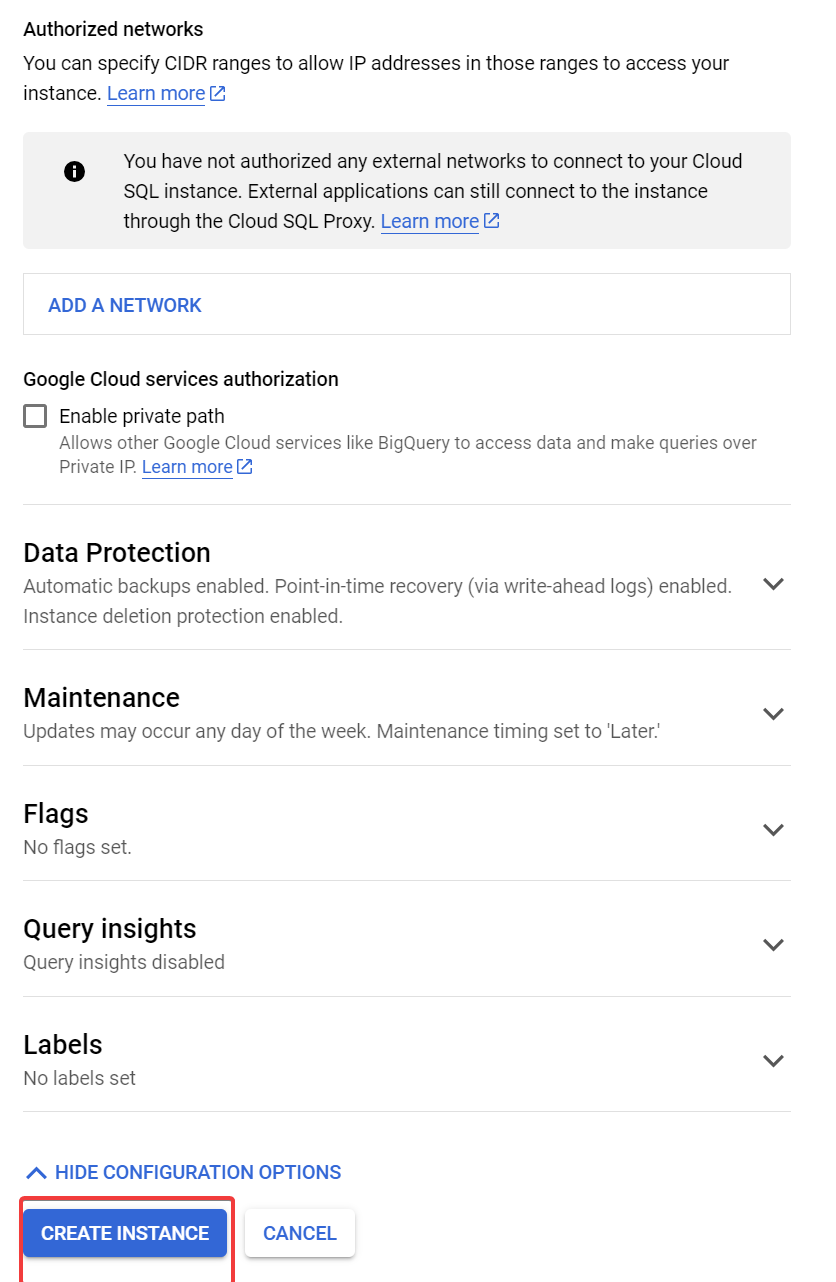
# Appendix 1: Creating the SQL instance

Go the SQL menu, and press CREATE INSTANCE

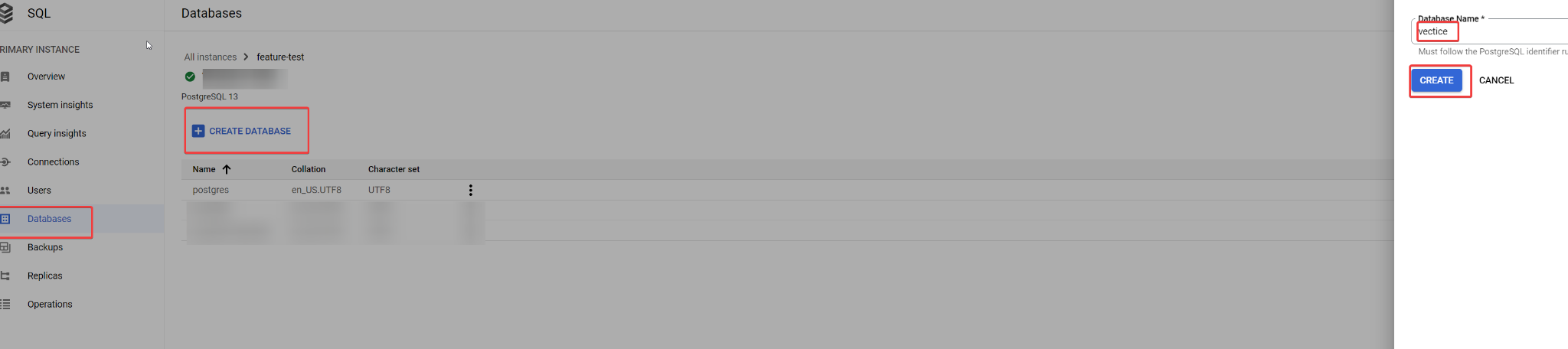
We’ll then input the instance name, the master user, and password





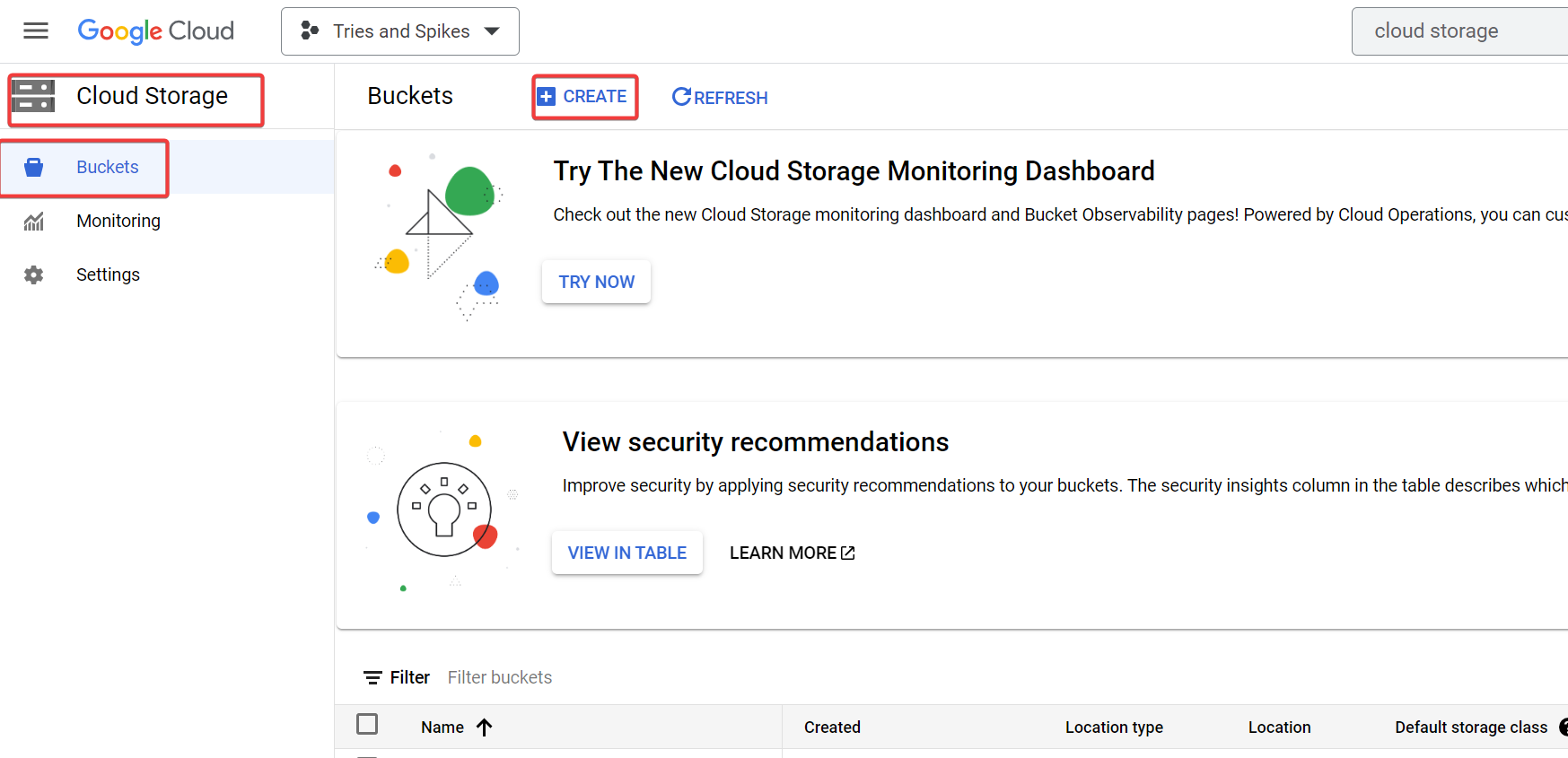


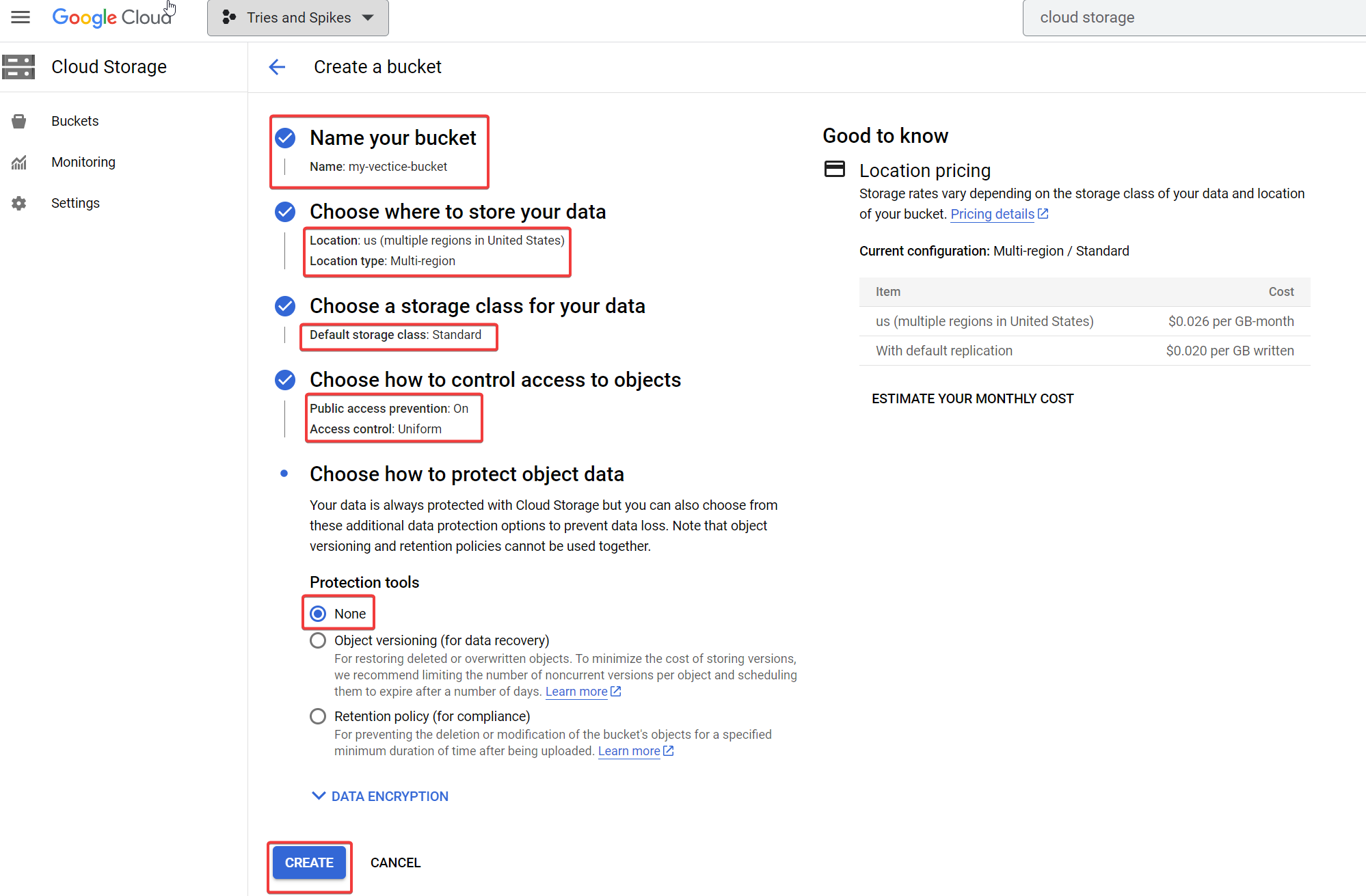
We’ll then create the two databases, vectice and keycloak  
From the menu “Databases”, CREATE DATABASE, we create vectice, then redo the operation for keycloak



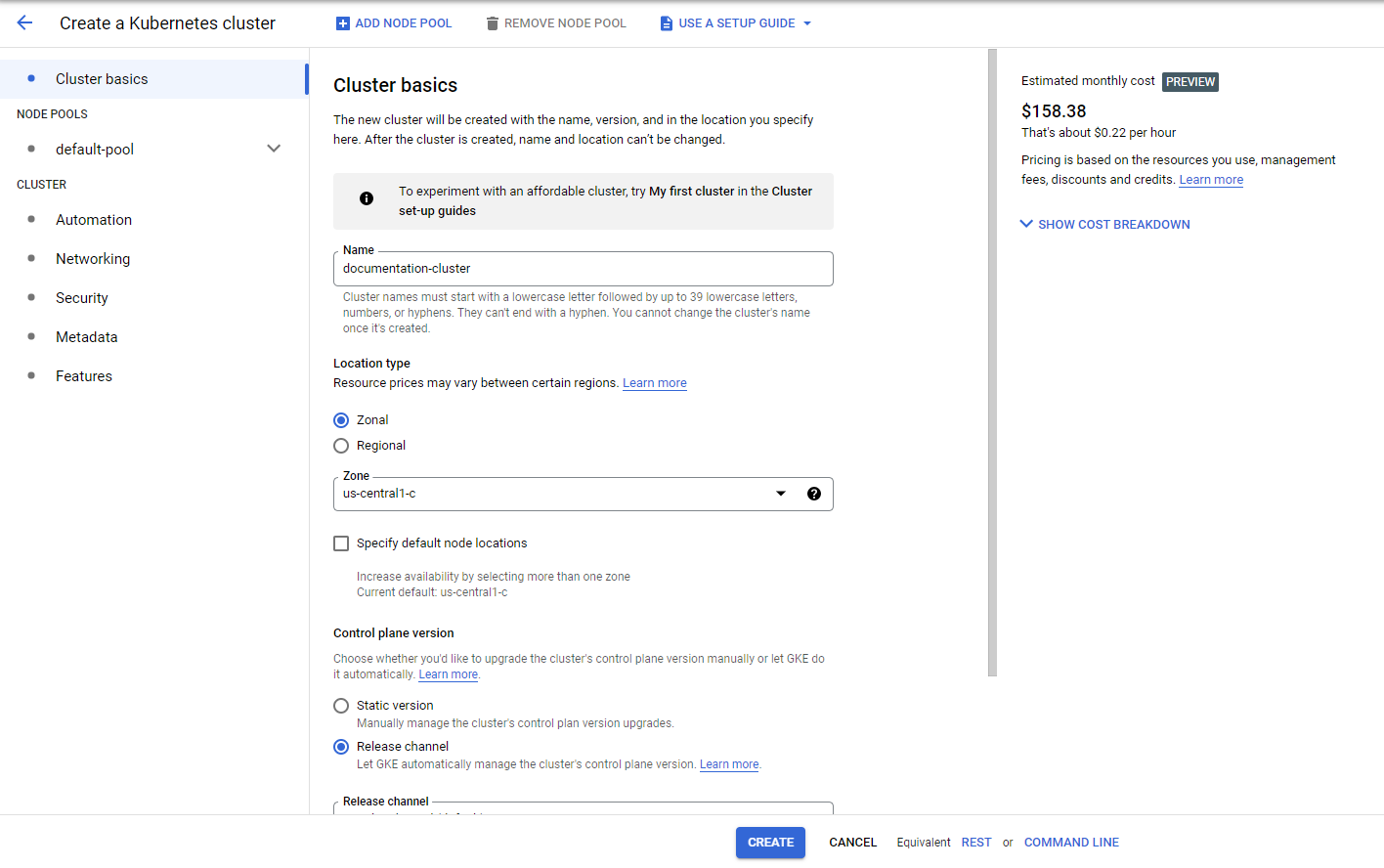
# Appendix 2: Bucket Creation

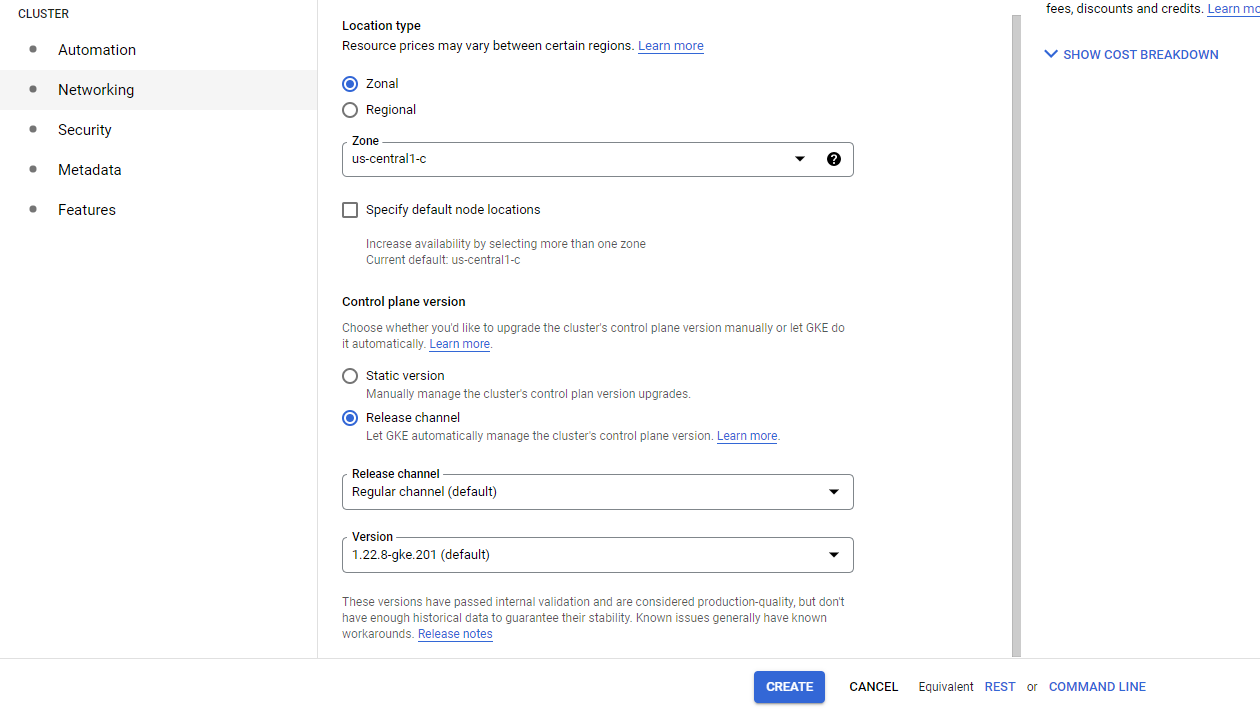
Go to Cloud Storage, Buckets menu, and Press CREATE

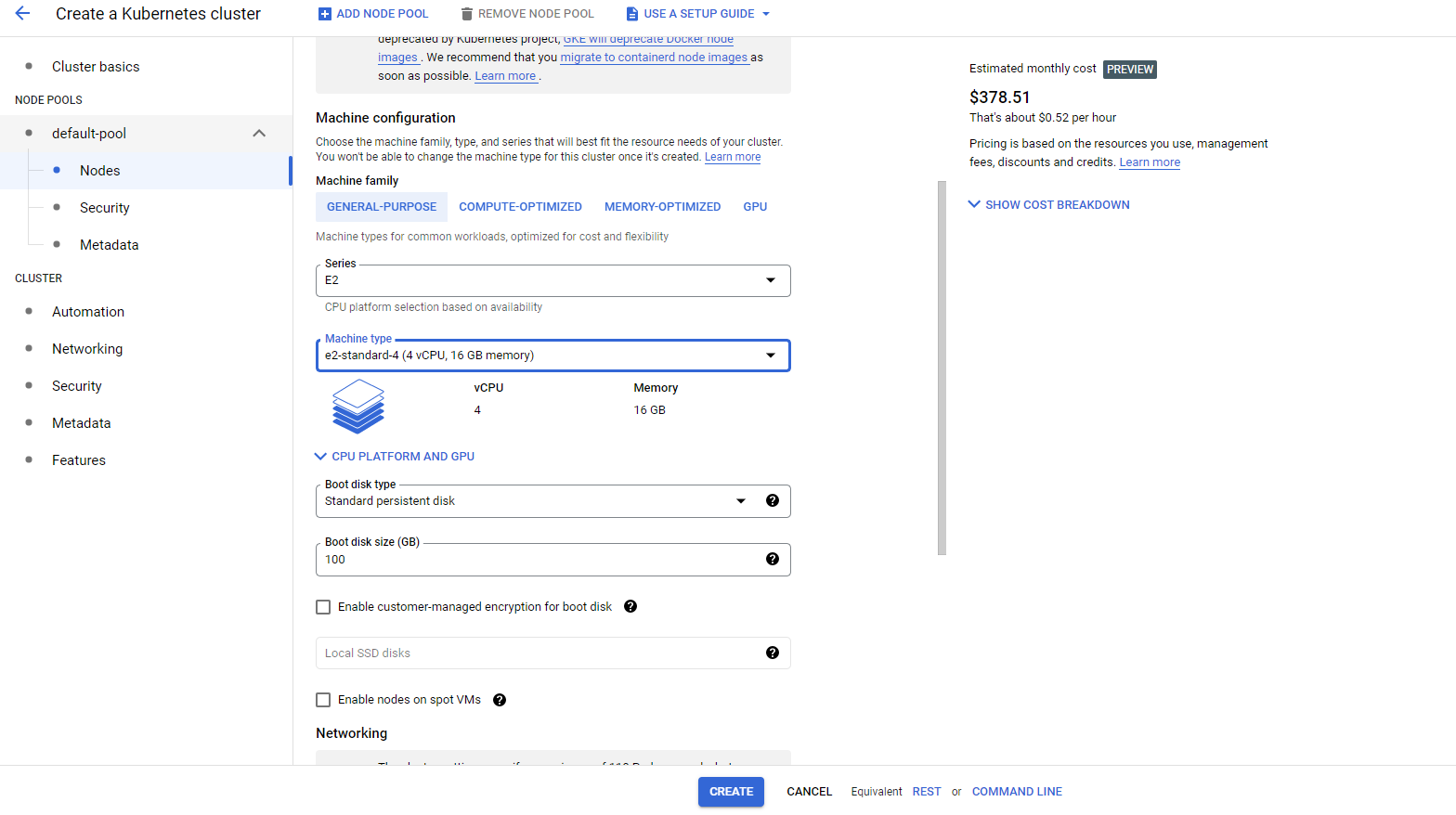
  
Fill in the settings as on screenshot below

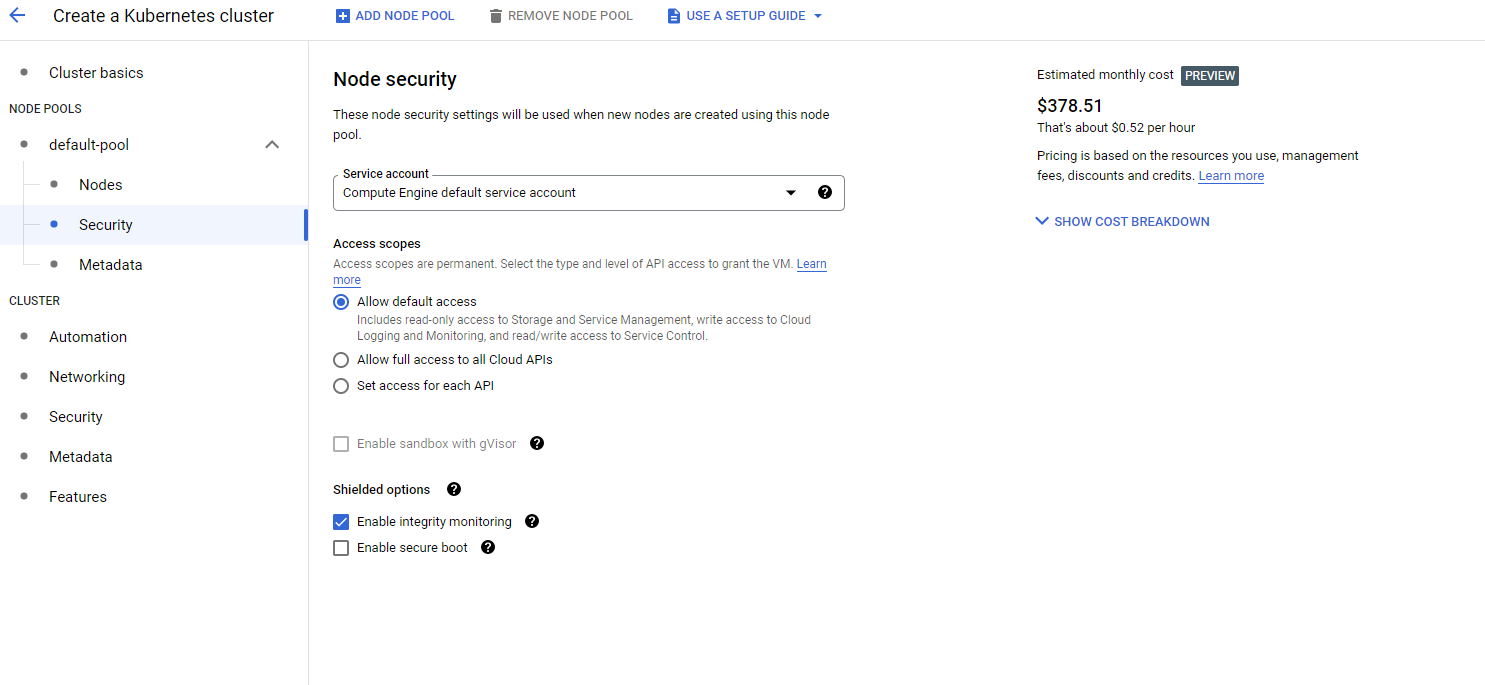


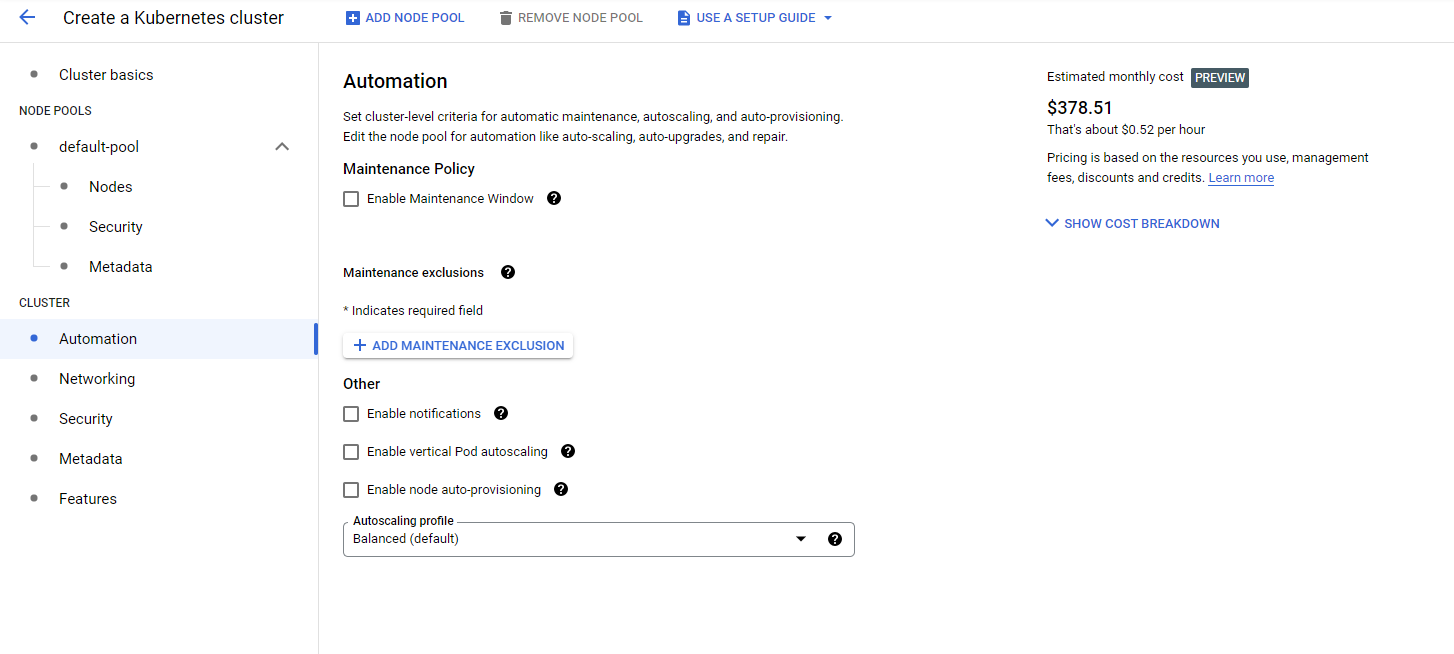
# Appendix 3: Cluster Creation

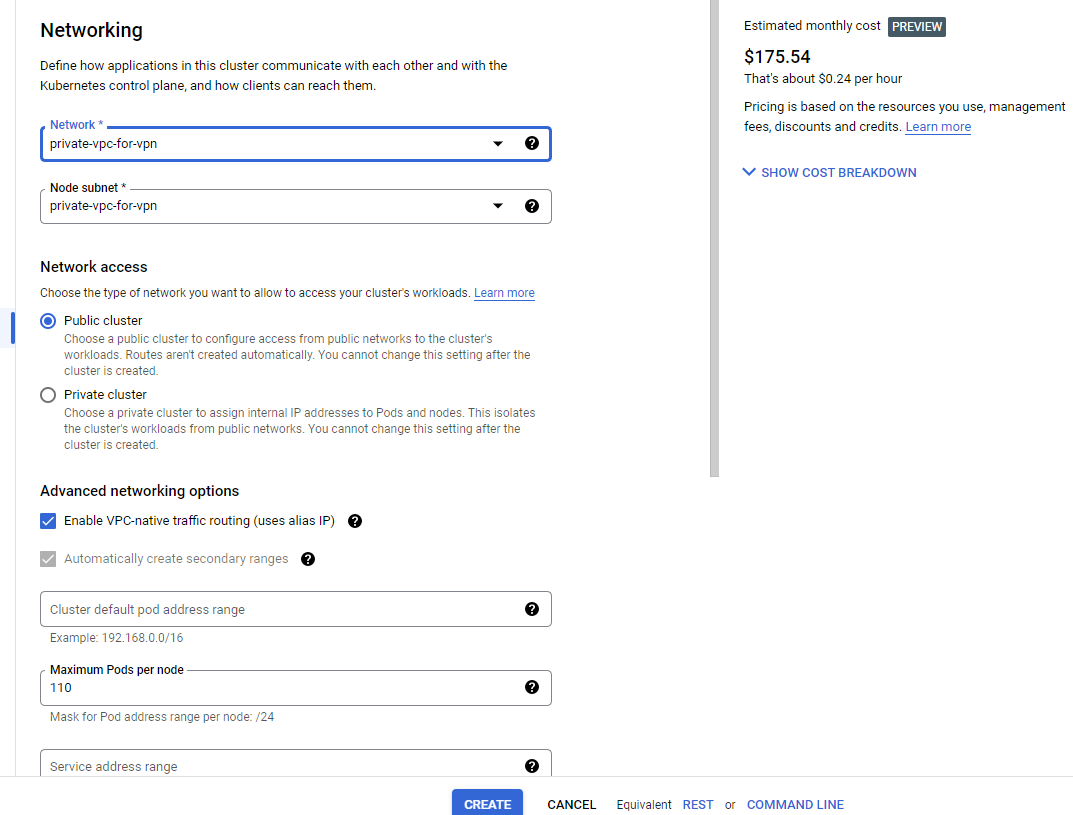












Once you push the create button, there will be a period of 30 min of creation of Cluster, Control Plan, NodePool, and Nodes.