Pyramid Kubernetes on GCP Guide

Version 1.2

Table of Contents

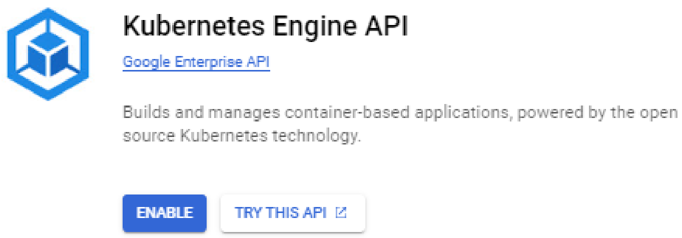
Overview

The following guide is provided to customers to setup and launch a Pyramid Kubernetes cluster on Google Cloud (GCP). The guide provides a standard walkthrough but is NOT exhaustive and does not cover every available option.

1. Instantiating Kubernetes on GCP

If you have no prior deployment of Google Kubernetes engine start here. Otherwise please start on step 12. You can either use your existing cluster or choose to deploy one just for Pyramid.

Log into the Google Admin. From the Google Cloud Engine, please choose Kubernetes Engine, if you don't already have it enabled. You will be presented with the following image.

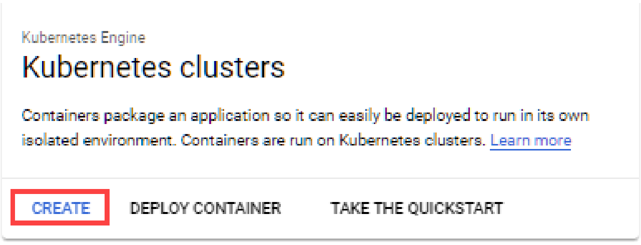


Kubernetes Engine API

Google Enterprise API

Builds and manages container-based applications, powered by the open source Kubernetes technology.

a) Once enabled, you can create your first Kubernetes cluster.
Click on "Create"



Kubernetes Engine

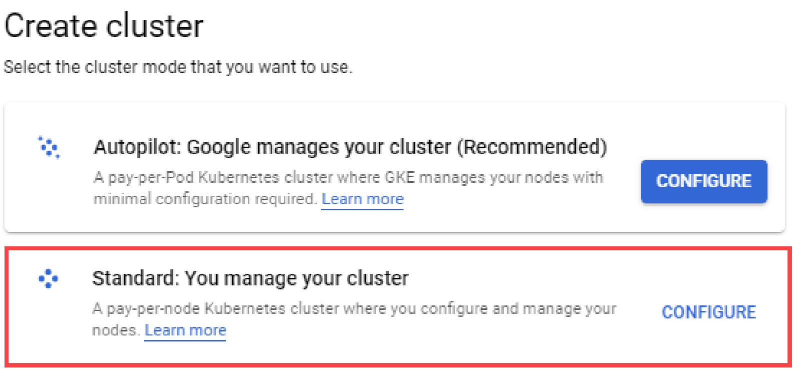
Kubernetes clusters

Containers package an application so it can easily be deployed to run in its own isolated environment. Containers are run on Kubernetes clusters. Learn more

b) Deploy a 'Standard' Google Kubernetes cluster

Currently the Google autopilot cluster only considers request limits, so we do not recommend using it. See here for more details.

Choose "Standard: You manage your cluster."



Create cluster

Select the cluster mode that you want to use.

Autopilot: Google manages your cluster (Recommended)

A pay-per-Pod Kubernetes cluster where GKE manages your nodes with minimal configuration required. Learn more

Standard: You manage your cluster

A pay-per-node Kubernetes cluster where you configure and manage your nodes. Learn more

Cluster basics

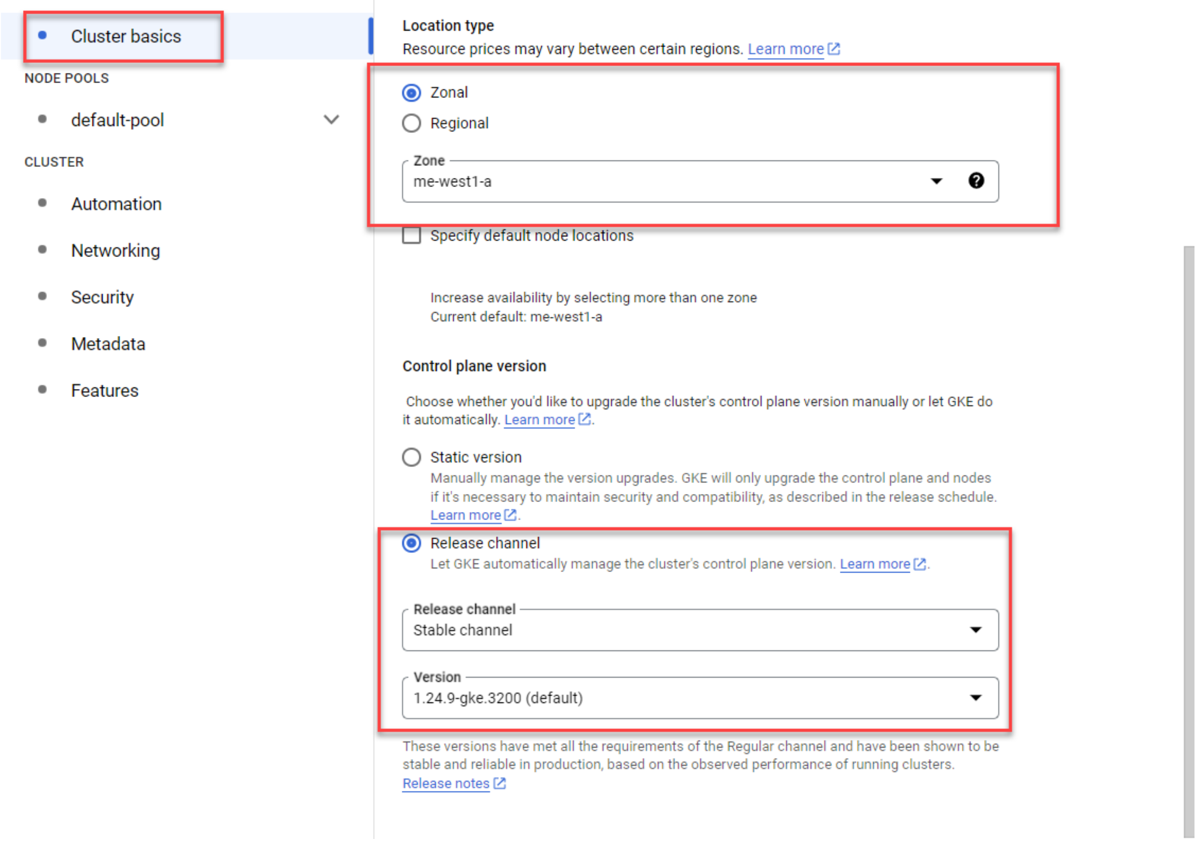
Settings that are required to be set are:

a. **Name** – set a name for your cluster.

b. **Zonal -** set to the zone closest to your users.

c. **Release channel-** set to Stable channel.

d. **Version –** can be left on the default option.



Location type

Resource prices may vary between certain regions. Learn more

● Zonal

○ Regional

Zone: me-west1-a

□ Specify default node locations

Increase availability by selecting more than one zone
Current default: me-west1-a

Control plane version

Choose whether you'd like to upgrade the cluster's control plane version manually or let GKE do it automatically. Learn more.

○ Static version
Manually manage the version upgrades. GKE will only upgrade the control plane and nodes if it's necessary to maintain security and compatibility, as described in the release schedule. Learn more.

● Release channel
Let GKE automatically manage the cluster's control plane version. Learn more.

Release channel: Stable channel

Version: 1.24.9-gke.3200 (default)

These versions have met all the requirements of the Regular channel and have been shown to be stable and reliable in production, based on the observed performance of running clusters.
Release notes

Nodes

This depends a lot on the expected usage.

Settings that are required to be set are:

a. **Series** – suggested start is to use E2 series.

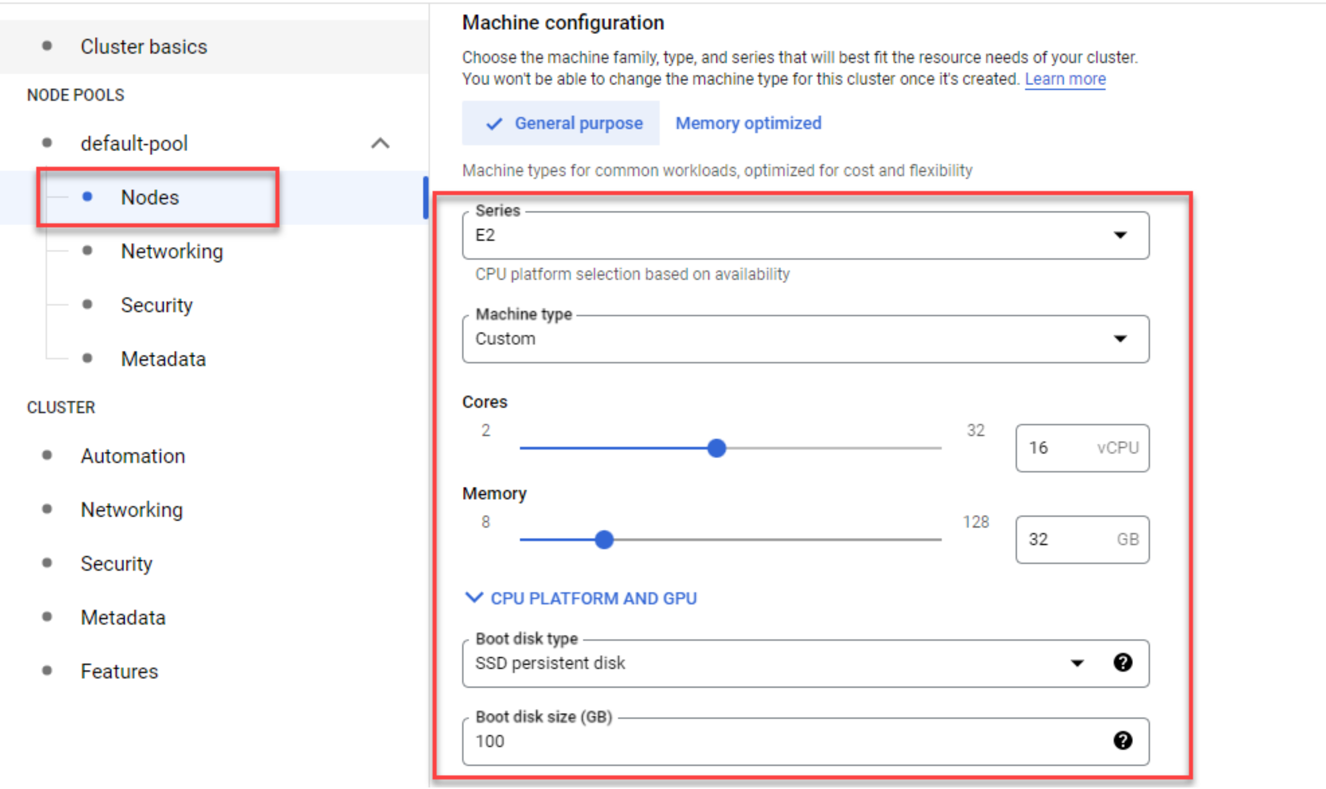
b. **Machine type –** set to custom.

c. **Cores and memory –** it is suggested to set the minimum node side to at least 16 cores and 32GB of memory *.

d. **Boot disk type-** SSD persistent disks

e. **Boot disk size (GB) –** set to 100GB

* When using the Pyramid Kubernetes Configurator (described below) to generate the cluster, a minimum initial node size of 16 CPUs and 24GB of RAM is assumed.



Machine configuration

Choose the machine family, type, and series that will best fit the resource needs of your cluster. You won't be able to change the machine type for this cluster once it's created. Learn more

✓ General purpose | Memory optimized

Machine types for common workloads, optimized for cost and flexibility

Series: E2

CPU platform selection based on availability

Machine type: Custom

Cores: 2 – 32: 16 vCPU

Memory: 8 – 128: 32 GB

CPU PLATFORM AND GPU

Boot disk type: SSD persistent disk

Boot disk size (GB): 100

Networking

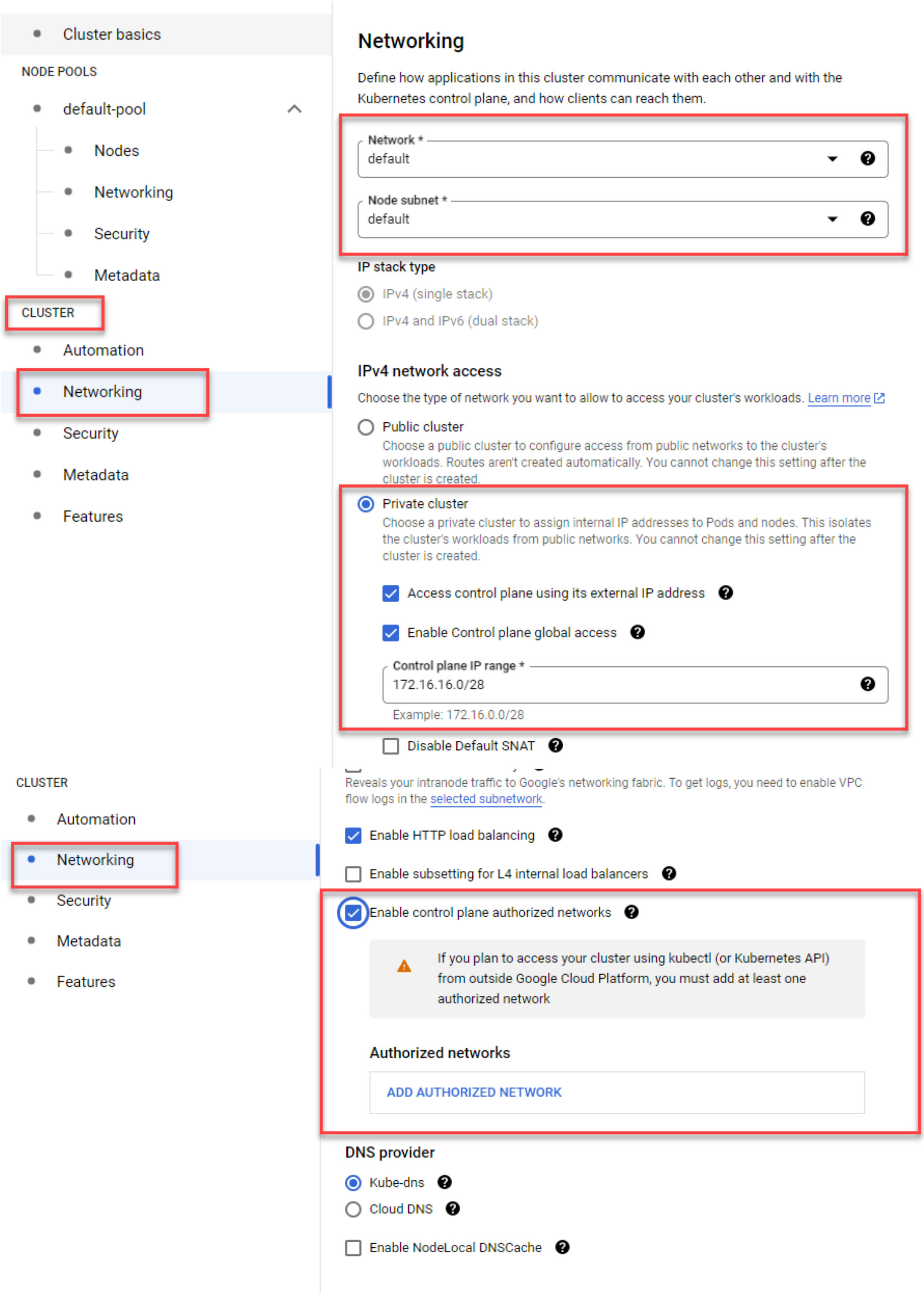
Settings that are required to be set are:

a. **Network** – use the default or one of your other networks.

b. **Node subnet –** use the default or one of your subnets.

c. **Private cluster –** set as a private cluster. The webserver is exposed through a load balancer (configured in the Pyramid YAML).

d. **Enable control panel authorized networks-** set this if you want to connect to the cluster from outside of the google cloud platform.



Networking

Define how applications in this cluster communicate with each other and with the Kubernetes control plane, and how clients can reach them.

Network * default

Node subnet * default

IP stack type

● IPv4 (single stack)

○ IPv4 and IPv6 (dual stack)

IPv4 network access

Choose the type of network you want to allow to access your cluster's workloads. Learn more

○ Public cluster
Choose a public cluster to configure access from public networks to the cluster's workloads. Routes aren't created automatically. You cannot change this setting after the cluster is created.

● Private cluster
Choose a private cluster to assign internal IP addresses to Pods and nodes. This isolates the cluster's workloads from public networks. You cannot change this setting after the cluster is created.

☑ Access control plane using its external IP address

☑ Enable Control plane global access

Control plane IP range * 172.16.16.0/28
Example: 172.16.0.0/28

□ Disable Default SNAT

Reveals your intranode traffic to Google's networking fabric. To get logs, you need to enable VPC flow logs in the selected subnetwork.

☑ Enable HTTP load balancing

□ Enable subsetting for L4 internal load balancers

☑ Enable control plane authorized networks

If you plan to access your cluster using kubectl (or Kubernetes API) from outside Google Cloud Platform, you must add at least one authorized network

Authorized networks

ADD AUTHORIZED NETWORK

DNS provider

● Kube-dns

○ Cloud DNS

□ Enable NodeLocal DNSCache

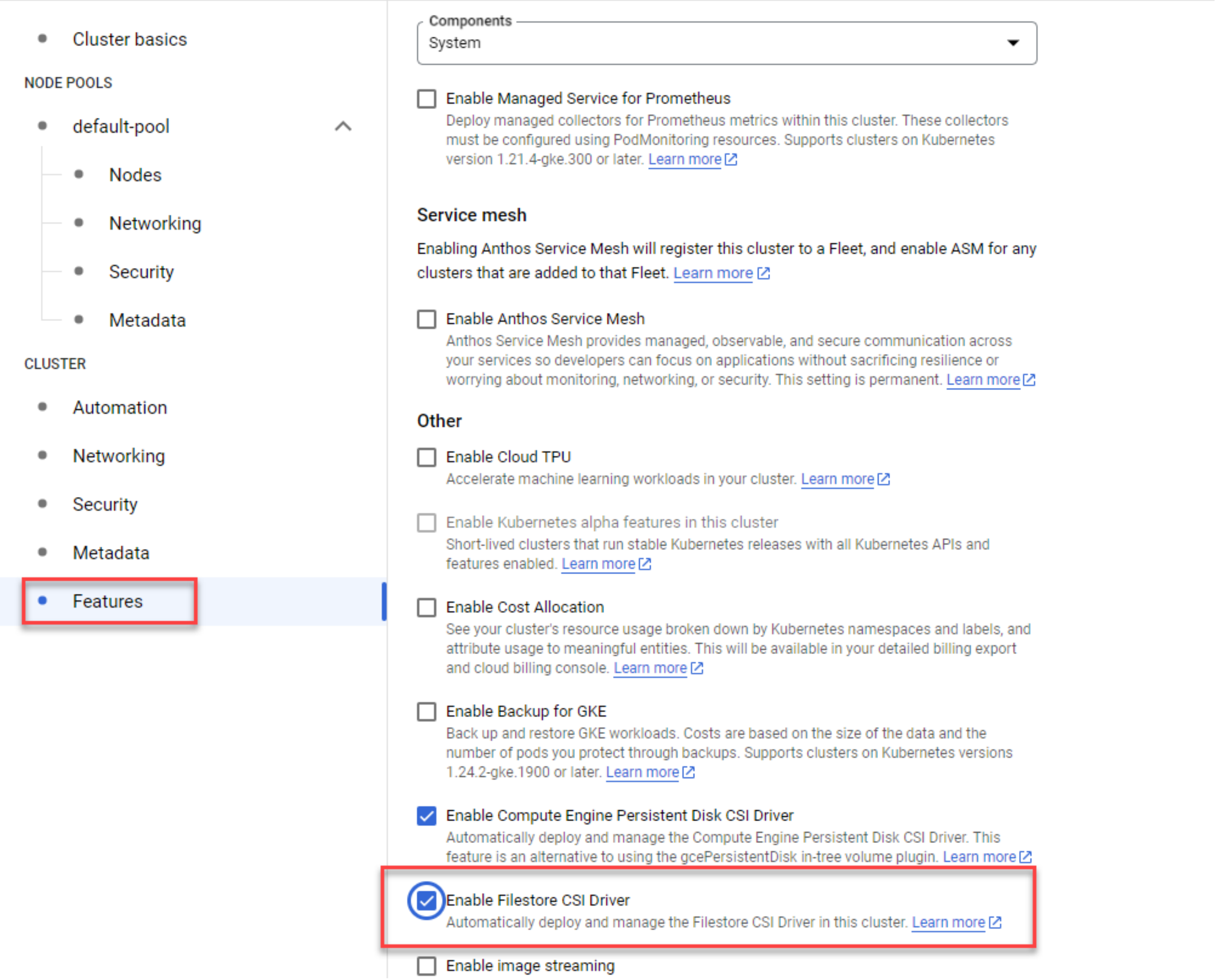
Features

Settings that are required to be set are:

a. **Enable Filestore CSI Driver –** Must be enabled for google persistent storage to work.

b. **Leave all other settings as default unless required otherwise.**

c. **Finally, click on "create"**



- Cluster basics

NODE POOLS

- default-pool
  - Nodes
  - Networking
  - Security
  - Metadata

CLUSTER

- Automation
- Networking
- Security
- Metadata
- Features

Components: System

☐ Enable Managed Service for Prometheus
Deploy managed collectors for Prometheus metrics within this cluster. These collectors must be configured using PodMonitoring resources. Supports clusters on Kubernetes version 1.21.4-gke.300 or later. Learn more

Service mesh

Enabling Anthos Service Mesh will register this cluster to a Fleet, and enable ASM for any clusters that are added to that Fleet. Learn more

☐ Enable Anthos Service Mesh
Anthos Service Mesh provides managed, observable, and secure communication across your services so developers can focus on applications without sacrificing resilience or worrying about monitoring, networking, or security. This setting is permanent. Learn more

Other

☐ Enable Cloud TPU
Accelerate machine learning workloads in your cluster. Learn more

☐ Enable Kubernetes alpha features in this cluster
Short-lived clusters that run stable Kubernetes releases with all Kubernetes APIs and features enabled. Learn more

☐ Enable Cost Allocation
See your cluster's resource usage broken down by Kubernetes namespaces and labels, and attribute usage to meaningful entities. This will be available in your detailed billing export and cloud billing console. Learn more

☐ Enable Backup for GKE
Back up and restore GKE workloads. Costs are based on the size of the data and the number of pods you protect through backups. Supports clusters on Kubernetes versions 1.24.2-gke.1900 or later. Learn more

☑ Enable Compute Engine Persistent Disk CSI Driver
Automatically deploy and manage the Compute Engine Persistent Disk CSI Driver. This feature is an alternative to using the gcePersistentDisk in-tree volume plugin. Learn more

☑ Enable Filestore CSI Driver
Automatically deploy and manage the Filestore CSI Driver in this cluster. Learn more

☐ Enable image streaming

2. Enabling Internet Access

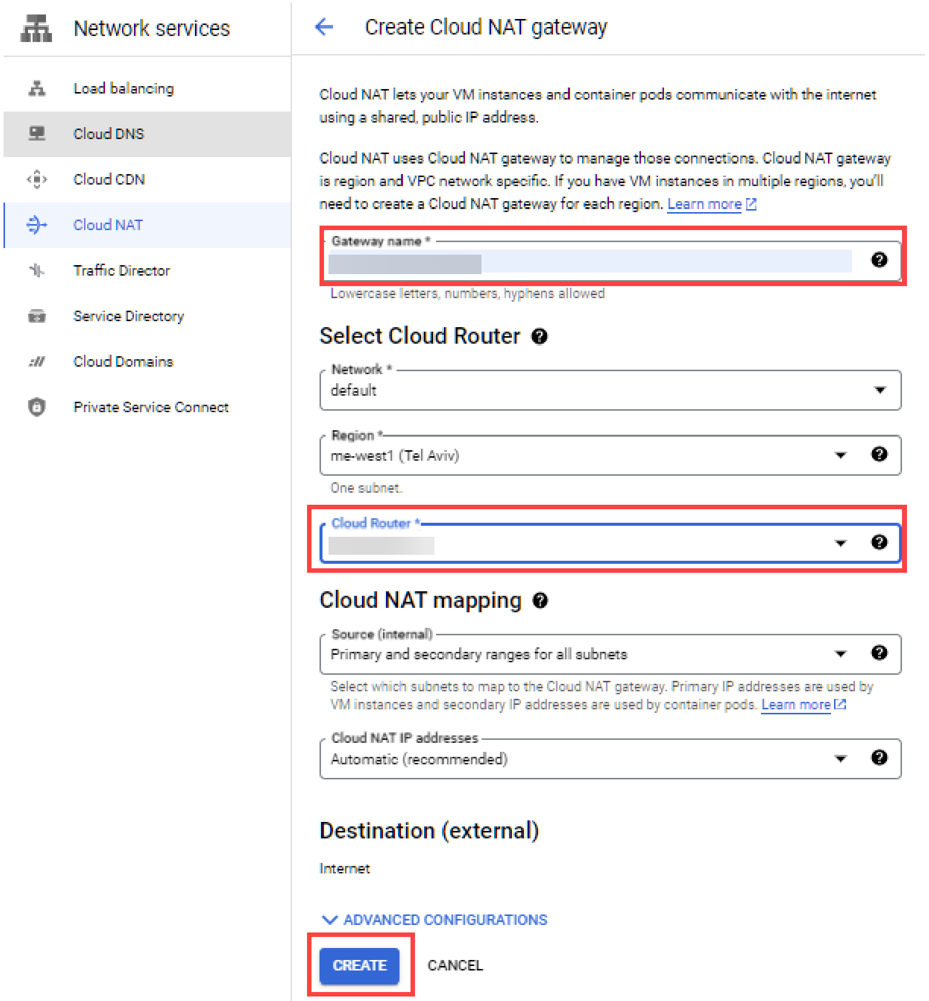
The nodes must have internet access to be able to pull down Pyramid's containers.

To give outgoing internet access outgoing to the nodes see this google article - see step 6: Create a NAT configuration using Cloud Router.

Example setup:

Settings that a required to be set are:

- **Gateway name** – choose a name for the gateway.
- **Network –** leave as default (should be the same as what your Kubernetes cluster uses)
- **Region-** should be set to the same region as the Kubernetes cluster.
- **Cloud Router** – If you do not have one already, it will ask you to create one. Follow the on-screen instructions.



Cloud NAT lets your VM instances and container pods communicate with the internet using a shared, public IP address.

Cloud NAT uses Cloud NAT gateway to manage those connections. Cloud NAT gateway is region and VPC network specific. If you have VM instances in multiple regions, you'll need to create a Cloud NAT gateway for each region. Learn more

Gateway name *: [redacted]

Lowercase letters, numbers, hyphens allowed

Select Cloud Router

Network *: default

Region *: me-west1 (Tel Aviv)

One subnet.

Cloud Router *: [redacted]

Cloud NAT mapping

Source (internal): Primary and secondary ranges for all subnets

Select which subnets to map to the Cloud NAT gateway. Primary IP addresses are used by VM instances and secondary IP addresses are used by container pods. Learn more

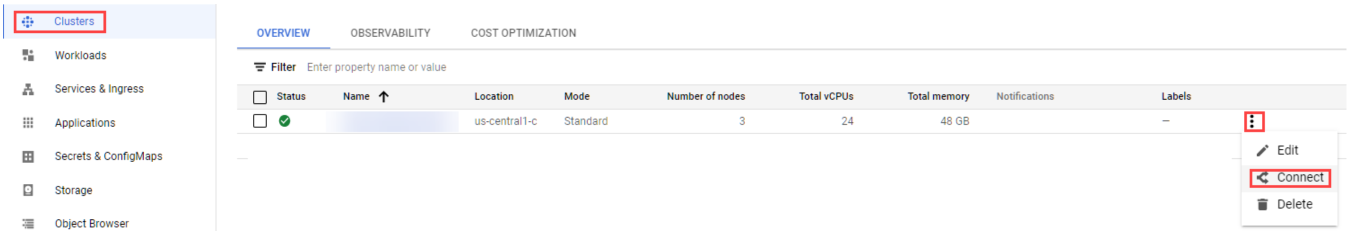
Cloud NAT IP addresses: Automatic (recommended)

Destination (external)

Internet

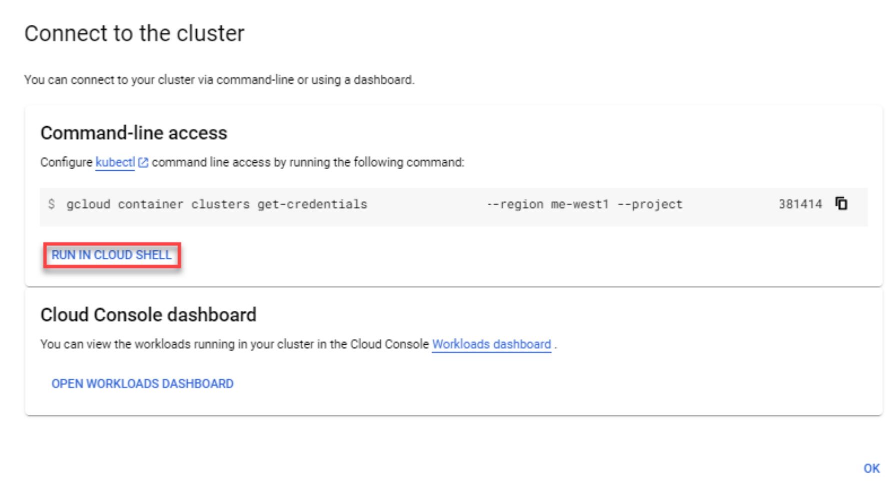
3. Connecting to the Cluster

a) Once the cluster has finished being created, click on connect as shown below.



□	Status	Name ↑	Location	Mode	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
□	✓		us-central1-c	Standard	3	24	48 GB		–

b) Click on Run in Cloud Shell



Connect to the cluster

You can connect to your cluster via command-line or using a dashboard.

Command-line access

Configure kubectl command line access by running the following command:

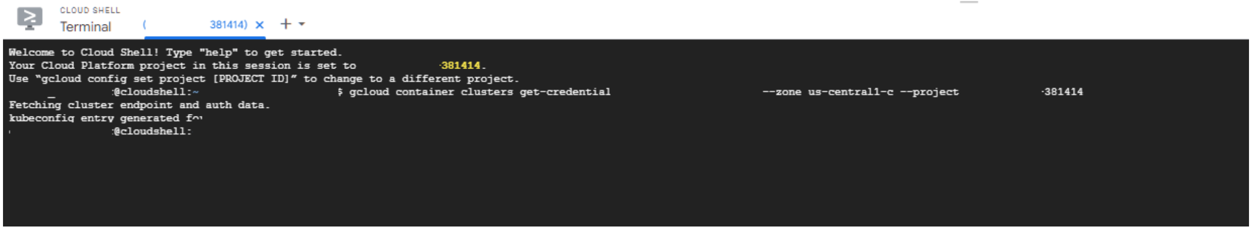
```
$ gcloud container clusters get-credentials                --region me-west1 --project            381414
```

Cloud Console dashboard

You can view the workloads running in your cluster in the Cloud Console Workloads dashboard .

c) Click on “continue”

d) Press Enter to connect



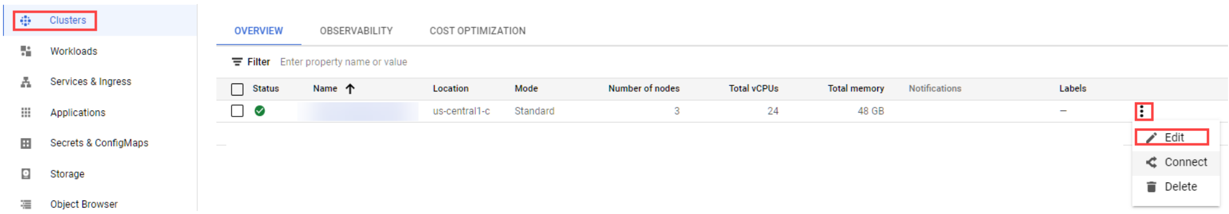
```
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to             381414.
Use "gcloud config set project [PROJECT ID]" to change to a different project.
             @cloudshell:~                  $ gcloud container clusters get-credential                   --zone us-central1-c --project          381414
Fetching cluster endpoint and auth data.
kubeconfig entry generated for
             @cloudshell:
```

e) Note that if you ticked the box “enable control plane authorized networks, then you will not be able connect from the shell to your cluster until you add its external IP to the list of “Authorized networks”. To get its IP run the below from the “Cloud Shell terminal (**NOTE THAT EACH TIME YOU CONNECT TO THE TERMINAL THE IP MIGHT CHANGE, so it needs to be updated in the authorization list**)”

```
curl -4 ifconfig.co
```

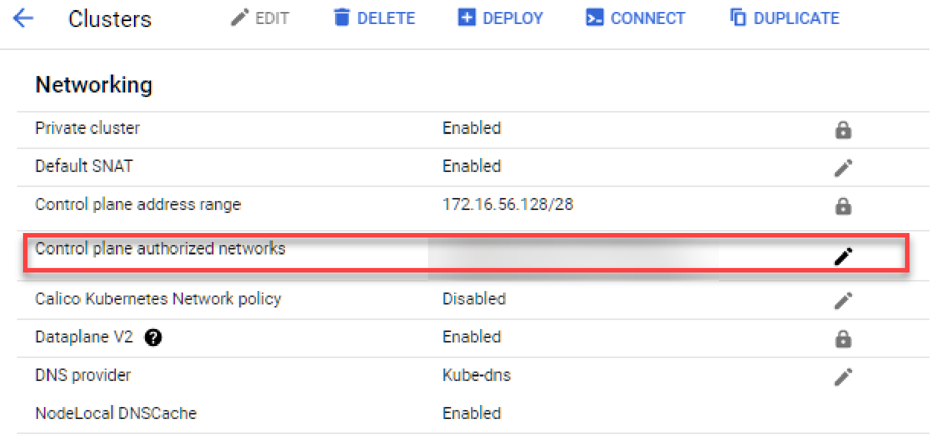
Then copy the IP the above command returns and update the authorized network list as shown below:

Click on edit on your cluster



□	Status	Name ↑	Location	Mode	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
□	✓		us-central1-c	Standard	3	24	48 GB		–

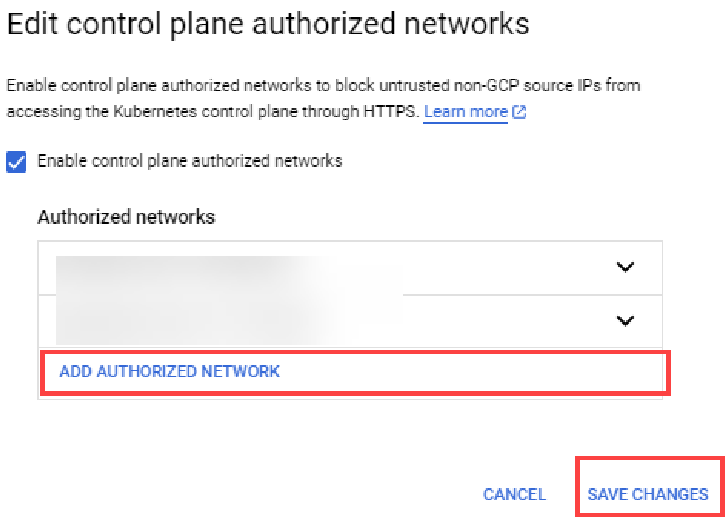
Edit the “Control plane aurthorized networks”.



Networking

Private cluster	Enabled
Default SNAT	Enabled
Control plane address range	172.16.56.128/28
Control plane authorized networks	[redacted]
Calico Kubernetes Network policy	Disabled
Dataplane V2	Enabled
DNS provider	Kube-dns
NodeLocal DNSCache	Enabled

Click on “Add authorized network”, add the IP of the cloud shell and save your changes.



Edit control plane authorized networks

Enable control plane authorized networks to block untrusted non-GCP source IPs from accessing the Kubernetes control plane through HTTPS. Learn more

☑ Enable control plane authorized networks

Authorized networks

[redacted]

Note that for the last step, you limit the IP addresses that can use the k8 control panel. To work out what IP to input use a Subnet Calculator, by inputting your public IP and getting back the Input.
If it’s one IP just add /32 to the end of it.

4. Generating the Pyramid YAML

The setup for Pyramid is *best* driven through a YAML configuration file. This can be manually created. However, it is simpler to use Pyramid's YAML configurator.

Helm Charts

As an alternative, you can deploy the cluster using Helm charts, as described here. However, with the complexity of the cluster, and the numerous settings required, the YAML configurator approach is simpler and faster.

The rest of this guide is designed around the use of the **configurator approach**.

Configurator

Login to Pyramid's customer portal, go to the Kubernetes setup page: https://customers.pyramidanalytics.com/kubernetes/ and generate a YAML file for your Pyramid config. If using Google storage, choose that option from the Persistence Storage dropdown. If you elect to use Google storage, then complete step 5 below. Otherwise, you can skip it. More info on the configurator can be found here .

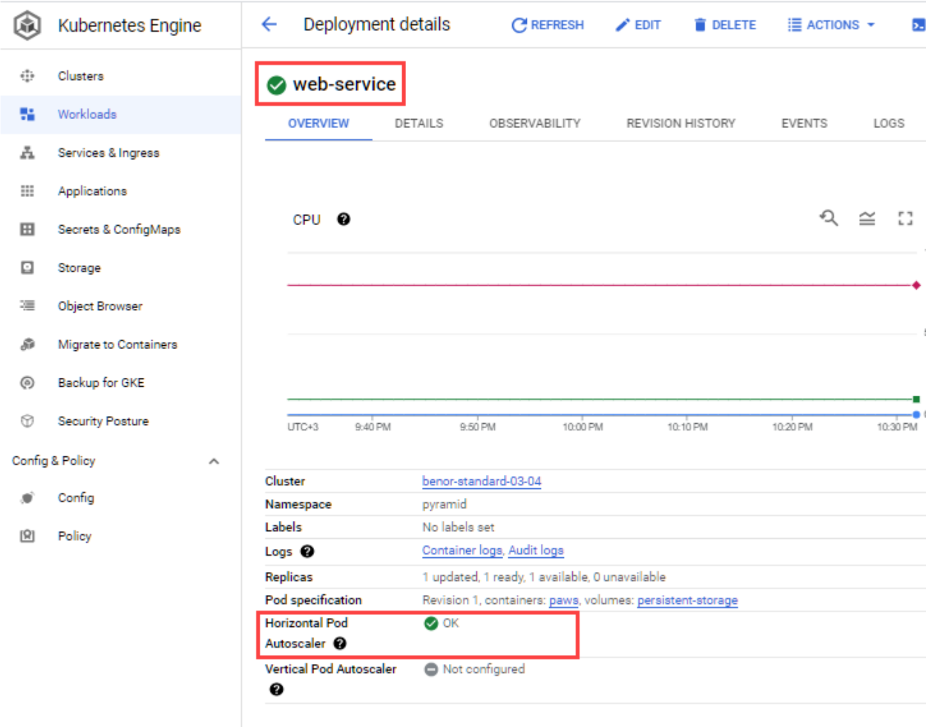
Autoscaling the pods:

Pyramid gives you the option of scaling the pods Horizontally (Horizontal Pod Autoscaling).
You can choose the maximum number of replicas(pods) to spawn by ticking the Elastic Scaling option when creating the Pyramid YAML and entering in the max number of pods that can be spawned.

To enable the auto scaling to work, please run the following commands on your cluster:

```
kubectl apply -f https://github.com/kedacore/keda/releases/download/v2.10.0/keda-2.10.0-core.yaml
```

Please note, that for the pods that you choose to auto scale (as set when creating the Pyramid YAML), it will show a green OK for "Horizontal pod Autoscaler". It can take up to 20mins for this to become active and show the status as green. These settings can be found under Workloads>choose pod>Overview

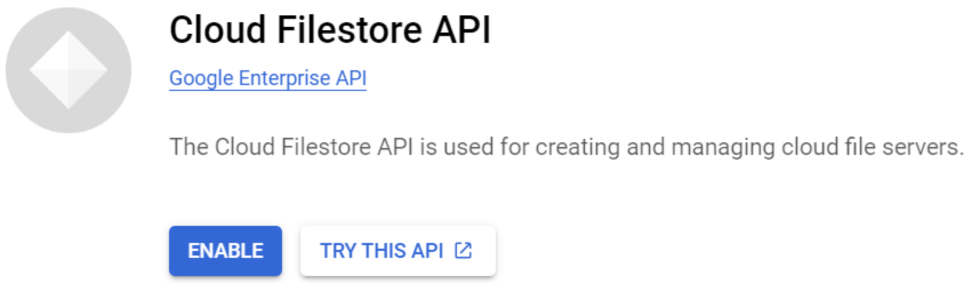


Cluster	benor-standard-03-04
Namespace	pyramid
Labels	No labels set
Logs	Container logs, Audit logs
Replicas	1 updated, 1 ready, 1 available, 0 unavailable
Pod specification	Revision 1, containers: paws, volumes: persistent-storage
Horizontal Pod Autoscaler	✓ OK
Vertical Pod Autoscaler	⊖ Not configured

5. Enabling Cloud Filestore

This step is only required when using Google Persistent Storage. Otherwise, move to step 6 below.

The Cloud Filestore API needs to be enabled in your workspace or it will fail to provision the storage when the YAML gets run.

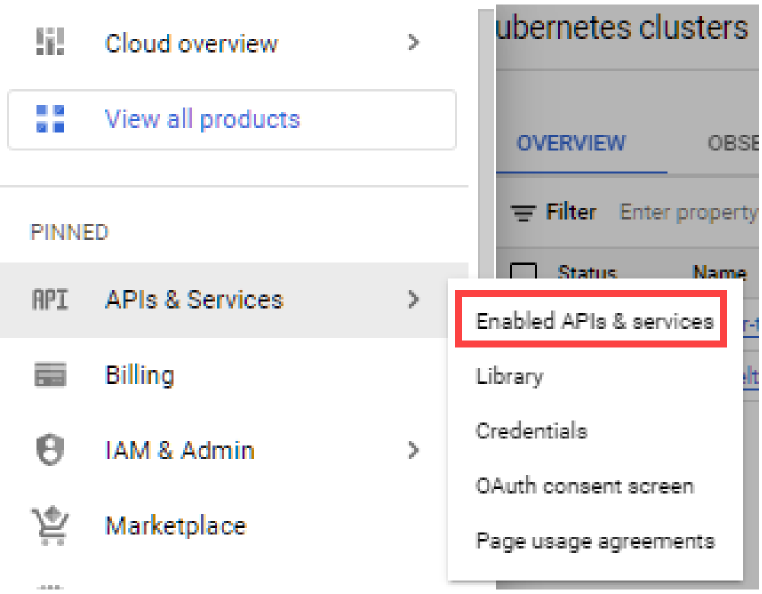


Cloud Filestore API

Google Enterprise API

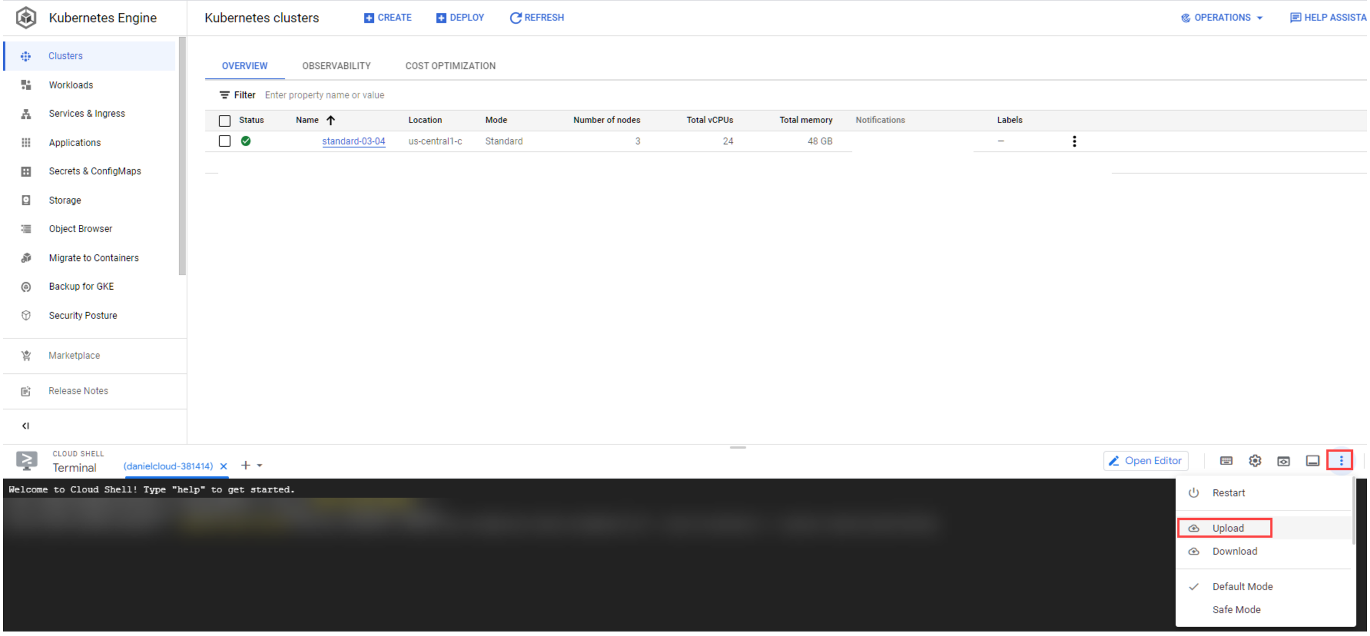
The Cloud Filestore API is used for creating and managing cloud file servers.

To enable it: APIs & services>enabled APIs & services , then search for “Cloud Filesstore API” and enable it.



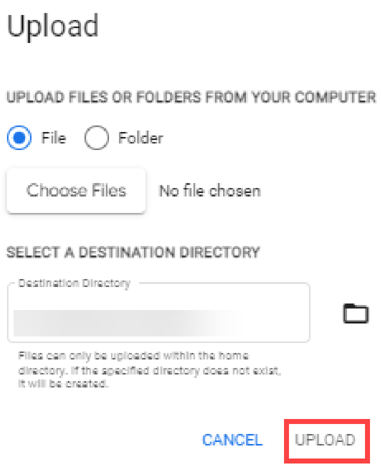
6. Deploying Pyramid YAML configuration

Upload your YAML file (from previous steps) to your cluster as shown below:



Status	Name	Location	Mode	Number of nodes	Total vCPUs	Total memory	Notifications	Labels
✓	standard-03-04	us-central1-c	Standard	3	24	48 GB		–

```
Welcome to Cloud Shell! Type "help" to get started.
[redacted]
```



Upload

UPLOAD FILES OR FOLDERS FROM YOUR COMPUTER

● File ○ Folder

Choose Files No file chosen

SELECT A DESTINATION DIRECTORY

Destination Directory [redacted]

Files can only be uploaded within the home directory. If the specified directory does not exist, it will be created.

Once you upload the YAML run it as below to pull down the pyramid pods

```
kubectl apply -f pyramid-analytics-config.yaml
```

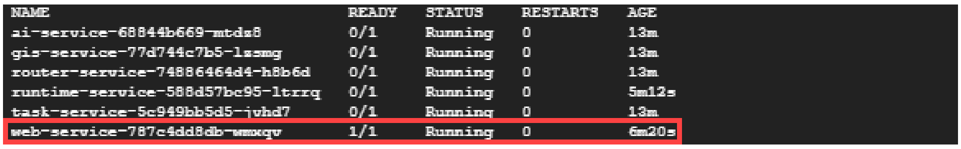
Then run the below command to see the pods generating or look at the Google control panel under “Workloads” (it will also show the pods as incomplete until after the full deployment has finished)

```
kubectl -n pyramid get pods -w
```

or

```
kubectl -n pyramid get pods
```

Its normal that only the web-service pod will show 1/1 until the full deployment has finished (after until after you have finished the setup in the browser)

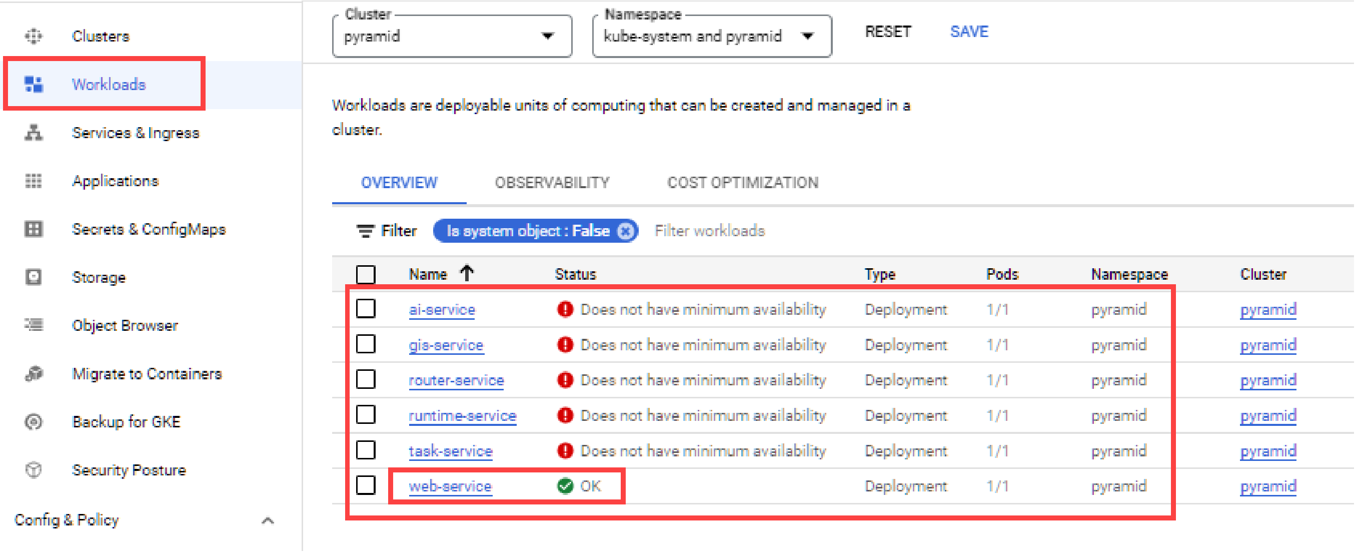


```
NAME                                  READY   STATUS    RESTARTS   AGE
ai-service-68844b669-mtds8            0/1     Running   0          13m
gis-service-77d744c7b5-lssmg          0/1     Running   0          13m
router-service-74886464d4-h8b6d       0/1     Running   0          13m
runtime-service-588d57bc95-ltrrq      0/1     Running   0          5m12s
task-service-5c949bb5d5-jvhd7         0/1     Running   0          13m
web-service-787c4dd8db-wmxgv          1/1     Running   0          6m20s
```

Wait until you see that all pods show as “running.”

From the Google console, it will look as below:

Once you see that the web-service shows as “OK”, continue to the next step.



Cluster: pyramid
Namespace: kube-system and pyramid

Workloads are deployable units of computing that can be created and managed in a cluster.

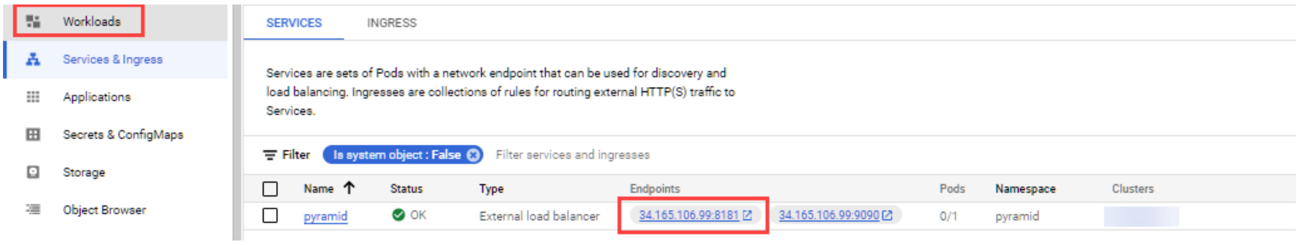
Filter: Is system object : False ⊗ Filter workloads

□	Name ↑	Status	Type	Pods	Namespace	Cluster
□	ai-service	❗ Does not have minimum availability	Deployment	1/1	pyramid	pyramid
□	gis-service	❗ Does not have minimum availability	Deployment	1/1	pyramid	pyramid
□	router-service	❗ Does not have minimum availability	Deployment	1/1	pyramid	pyramid
□	runtime-service	❗ Does not have minimum availability	Deployment	1/1	pyramid	pyramid
□	task-service	❗ Does not have minimum availability	Deployment	1/1	pyramid	pyramid
□	web-service	✓ OK	Deployment	1/1	pyramid	pyramid

Once you see that the web-service shows as “OK”, continue to the next step.

External IP Access for the Pyramid Kubernetes Instance

To get the external IP to access the Pyramid application on, click on “Services & Ingress” and click on the endpoint.



Services are sets of Pods with a network endpoint that can be used for discovery and load balancing. Ingresses are collections of rules for routing external HTTP(S) traffic to Services.

Filter: Is system object : False ⊗ Filter services and ingresses

□	Name ↑	Status	Type	Endpoints		Pods	Namespace	Clusters
□	pyramid	✓ OK	External load balancer	34.165.106.99:8181	34.165.106.99:9090	0/1	pyramid	[redacted]

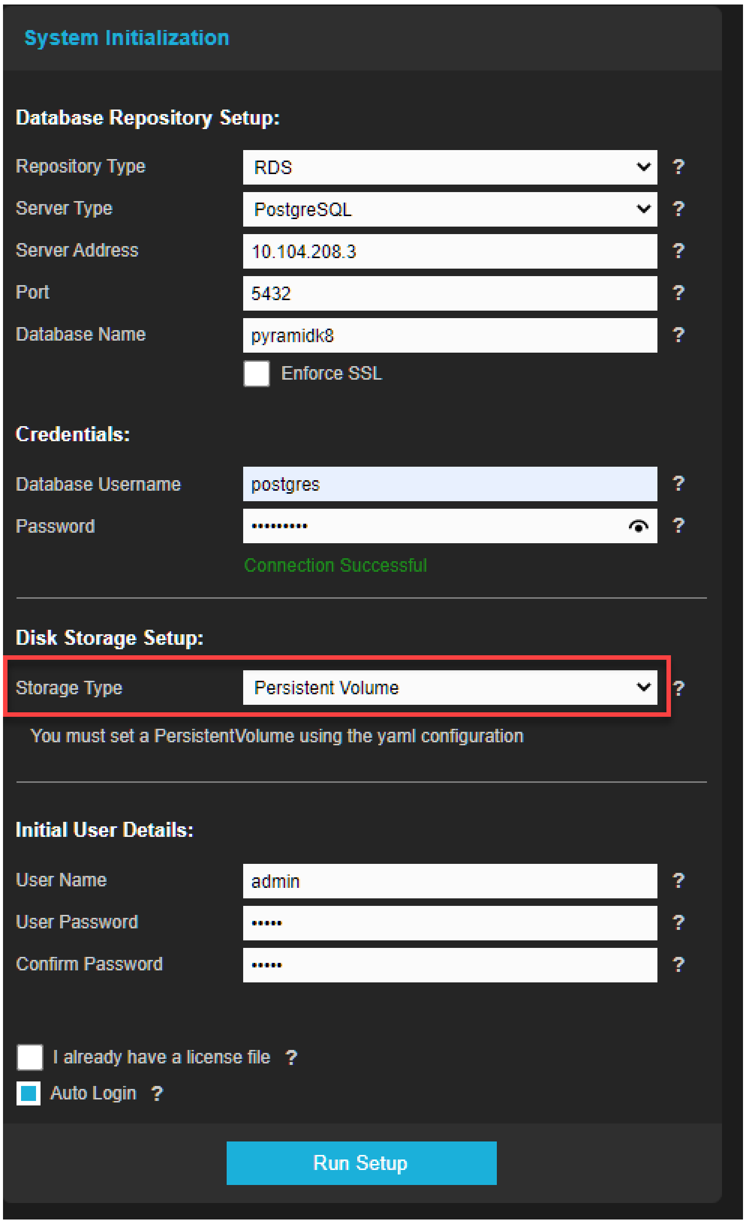
Clicking the above link will bring you to the below page, where you can fill out all the needed info to finish the Pyramid deployment.

7. System Initialization

Once the pods have finished being created, and you click on the link as explained above, you will be prompted with the screen below. This initializes the system, with persistent storage, the Pyramid repository database and creates the first initial user within Pyramid. For more information on this stage please see this link.

- See the appendix for details on how to setup a database repository on GCP.
- For more information on this stage please see this link.

For the storage type choose "Persistent volume," if you chose any of the persistent storage options in the Pyramid YAML.



System Initialization

Database Repository Setup:

Repository Type	RDS
Server Type	PostgreSQL
Server Address	10.104.208.3
Port	5432
Database Name	pyramidk8
	□ Enforce SSL

Credentials:

Database Username	postgres
Password	•••••••••
	Connection Successful

Disk Storage Setup:

Storage Type	Persistent Volume

You must set a PersistentVolume using the yaml configuration

Initial User Details:

User Name	admin
User Password	•••••
Confirm Password	•••••

□ I already have a license file ?

☑ Auto Login ?

8. Finished

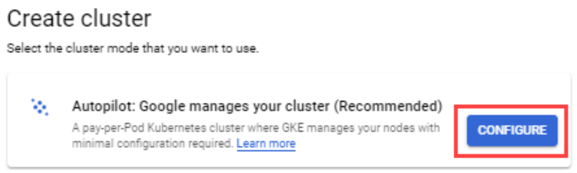
Once the initialization setup has finished running (normally around 5-10 mins) it will redirect you to the fully installed Pyramid application.

Appendix

1. How to create an Autopilot cluster

Not currently recommended by Pyramid

a) Click on “Configure” for the Autopilot option. (GKE version 1.24 and later.) Alternatively, use the standard cluster option. See appendix on “How to deploy a standard cluster”. For more information on how to decide what cluster type is best for you see this link



Create cluster

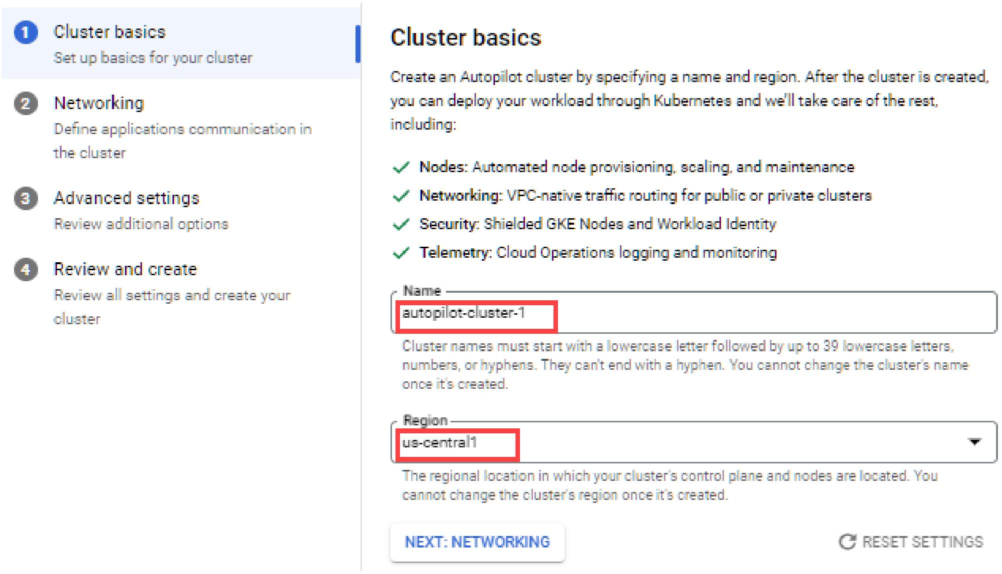
Select the cluster mode that you want to use.

Autopilot: Google manages your cluster (Recommended)

A pay-per-Pod Kubernetes cluster where GKE manages your nodes with minimal configuration required. Learn more

b) Click on “Let’s get started”.

c) Give your cluster a name (note it must start with lowercase characters) and choose the appropriate region.



1 Cluster basics
Set up basics for your cluster

2 Networking
Define applications communication in the cluster

3 Advanced settings
Review additional options

4 Review and create
Review all settings and create your cluster

Cluster basics

Create an Autopilot cluster by specifying a name and region. After the cluster is created, you can deploy your workload through Kubernetes and we'll take care of the rest, including:

- ✓ Nodes: Automated node provisioning, scaling, and maintenance
- ✓ Networking: VPC-native traffic routing for public or private clusters
- ✓ Security: Shielded GKE Nodes and Workload Identity
- ✓ Telemetry: Cloud Operations logging and monitoring

Name: autopilot-cluster-1

Cluster names must start with a lowercase letter followed by up to 39 lowercase letters, numbers, or hyphens. They can't end with a hyphen. You cannot change the cluster's name once it's created.

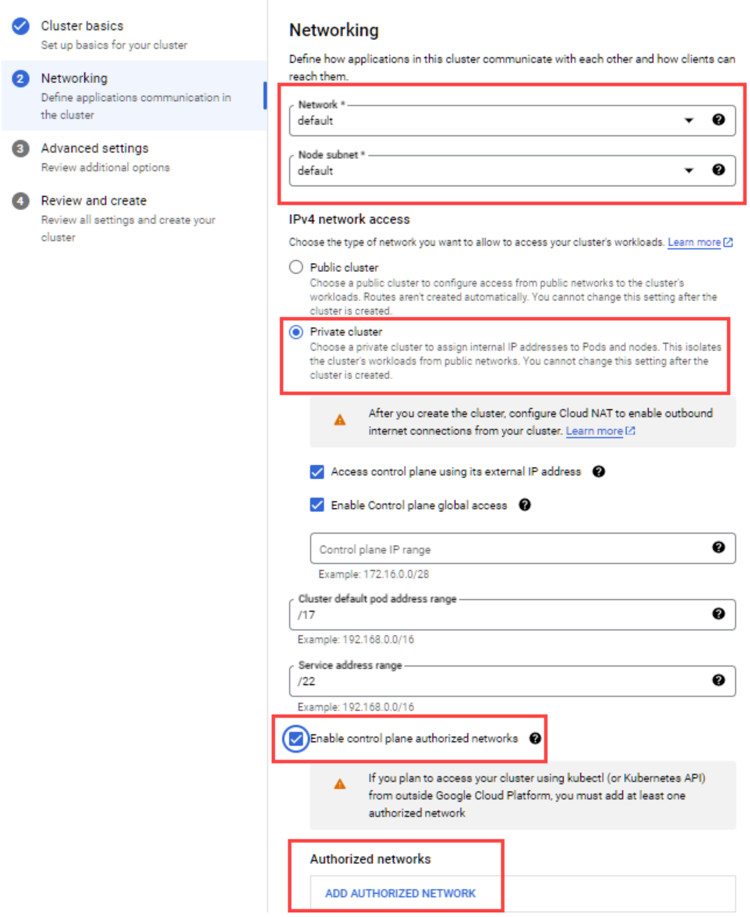
Region: us-central1

The regional location in which your cluster's control plane and nodes are located. You cannot change the cluster's region once it's created.

d) Setup the networking as shown below (or as per your requirements).
Settings that a required to be set are:

a. **Network** – set as “default” or create/us your own one
b. **Node subnet -** set as “default” or create/us your own one
c. **Private cluster** – the cluster should be a private one, as the webserver is exposed through a load balancer created by the Pyramid YAML.
d. **Cluster default pod address range** – can be left on its default setting
e. **Service address range** – can be left on its default setting
f. **Enable control plane authorized networks** – should be checked for better security

Note that for the last step, you limit the IP addresses that can use the k8 control panel. To work out what IP to input use a Subnet Calculator, by inputting your public IP and getting back the Input.
If it’s one IP just add /32 to the end of it.



Cluster basics
Set up basics for your cluster

2 Networking
Define applications communication in the cluster

3 Advanced settings
Review additional options

4 Review and create
Review all settings and create your cluster

Networking

Define how applications in this cluster communicate with each other and how clients can reach them.

Network * default

Node subnet * default

IPv4 network access

Choose the type of network you want to allow to access your cluster's workloads. Learn more

○ Public cluster
Choose a public cluster to configure access from public networks to the cluster's workloads. Routes aren't created automatically. You cannot change this setting after the cluster is created.

● Private cluster
Choose a private cluster to assign internal IP addresses to Pods and nodes. This isolates the cluster's workloads from public networks. You cannot change this setting after the cluster is created.

After you create the cluster, configure Cloud NAT to enable outbound internet connections from your cluster. Learn more

☑ Access control plane using its external IP address

☑ Enable Control plane global access

Control plane IP range
Example: 172.16.0.0/28

Cluster default pod address range /17
Example: 192.168.0.0/16

Service address range /22
Example: 192.168.0.0/16

☑ Enable control plane authorized networks

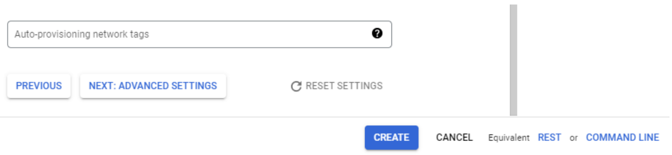
If you plan to access your cluster using kubectl (or Kubernetes API) from outside Google Cloud Platform, you must add at least one authorized network

Authorized networks

ADD AUTHORIZED NETWORK

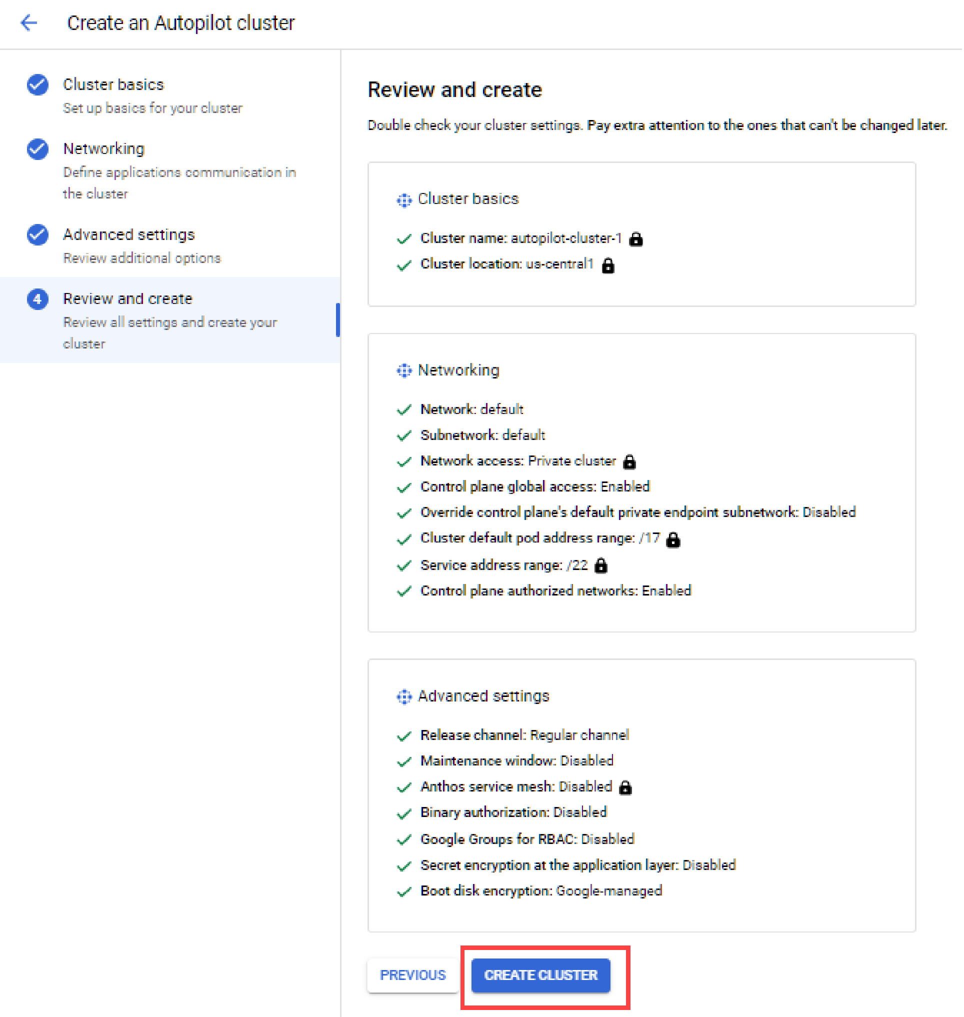
NOTE if you enable the option “**enable control panel authorized networks**” then you must add the public IP address from where you will connect from to your Authorized networks lists.

Click on “NEXT: ADVANCED SETTINGS”


Auto-provisioning network tags
PREVIOUS
NEXT: ADVANCED SETTINGS
RESET SETTINGS
CREATE
CANCEL
Equivalent REST or COMMAND LINE

e) Next, leave all options as default (unless your requirements are otherwise e.g., setting a maintenance windows) and give the cluster a description of your choosing.

f) Review and create your cluster.



Create an Autopilot cluster

Cluster basics
Set up basics for your cluster
Networking
Define applications communication in the cluster
Advanced settings
Review additional options
4 Review and create
Review all settings and create your cluster

Review and create

Double check your cluster settings. Pay extra attention to the ones that can't be changed later.

Cluster basics

- ✓ Cluster name: autopilot-cluster-1 🔒
- ✓ Cluster location: us-central1 🔒

Networking

- ✓ Network: default
- ✓ Subnetwork: default
- ✓ Network access: Private cluster 🔒
- ✓ Control plane global access: Enabled
- ✓ Override control plane's default private endpoint subnetwork: Disabled
- ✓ Cluster default pod address range: /17 🔒
- ✓ Service address range: /22 🔒
- ✓ Control plane authorized networks: Enabled

Advanced settings

- ✓ Release channel: Regular channel
- ✓ Maintenance window: Disabled
- ✓ Anthos service mesh: Disabled 🔒
- ✓ Binary authorization: Disabled
- ✓ Google Groups for RBAC: Disabled
- ✓ Secret encryption at the application layer: Disabled
- ✓ Boot disk encryption: Google-managed

To connect to the cluster and begin the deployment of Pyramid see section 3 “Connecting to the cluster.”

2. Deploying an MS-SQL or PostgreSQL Instance

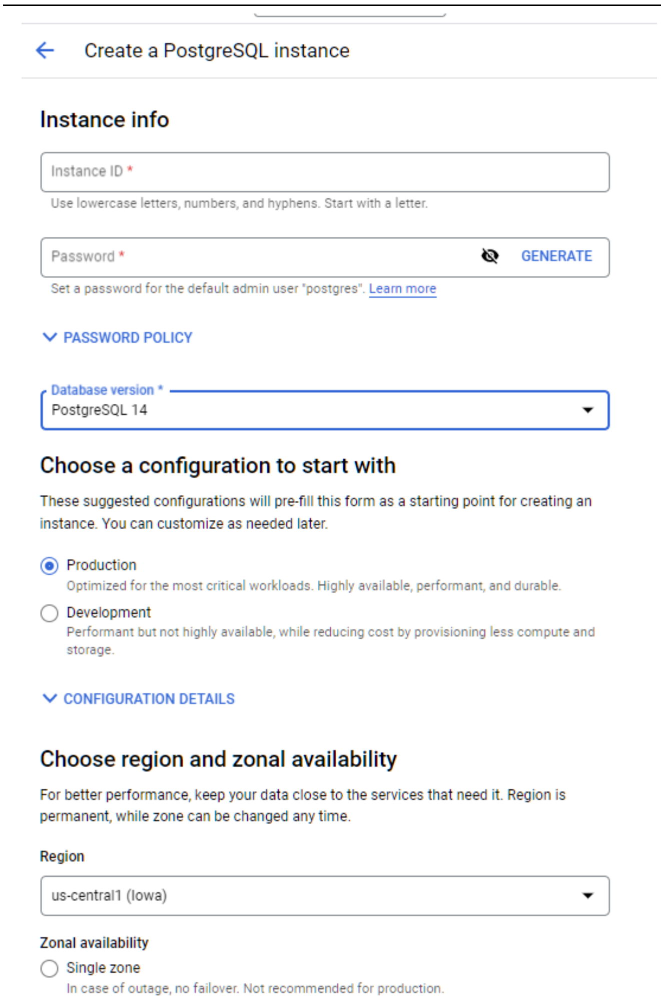
The steps below guide you in the Google Console for creating an MS-SQL or PostgreSQL database instance to host the Pyramid repository.

Notes: it should be a private instance as it does not need to be accessed from outside of your network. It should be in the same Zone and network as your Kubernetes cluster.

Settings that a required to be set are:

- **Instance ID** – a name for your Postgres instance
- **Password** -a password used to connect to the instance.
- **Database version** – set as PostgreSQL 14, but any version can be used.
- **Production** – this option should be selected.
- **Region** – should be the same region as the Kubernetes cluster is in
- **Private IP** – the instance should be set a private. There is no reason to give public access to the instance.
- **Region** – should be the same region as the Kubernetes cluster is in
- **Network** – set as default or your own network, making sure that the Kubernetes cluster can access this.

Ensure that the database is not underpowered. It should not be less than 4 CPU’s (8 is the recommended minimum) and 12-16Gb of Memory.



← Create a PostgreSQL instance

Instance info

Instance ID *

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password * GENERATE

Set a password for the default admin user "postgres". Learn more

PASSWORD POLICY

Database version *: PostgreSQL 14

Choose a configuration to start with

These suggested configurations will pre-fill this form as a starting point for creating an instance. You can customize as needed later.

● Production
Optimized for the most critical workloads. Highly available, performant, and durable.

○ Development
Performant but not highly available, while reducing cost by provisioning less compute and storage.

CONFIGURATION DETAILS

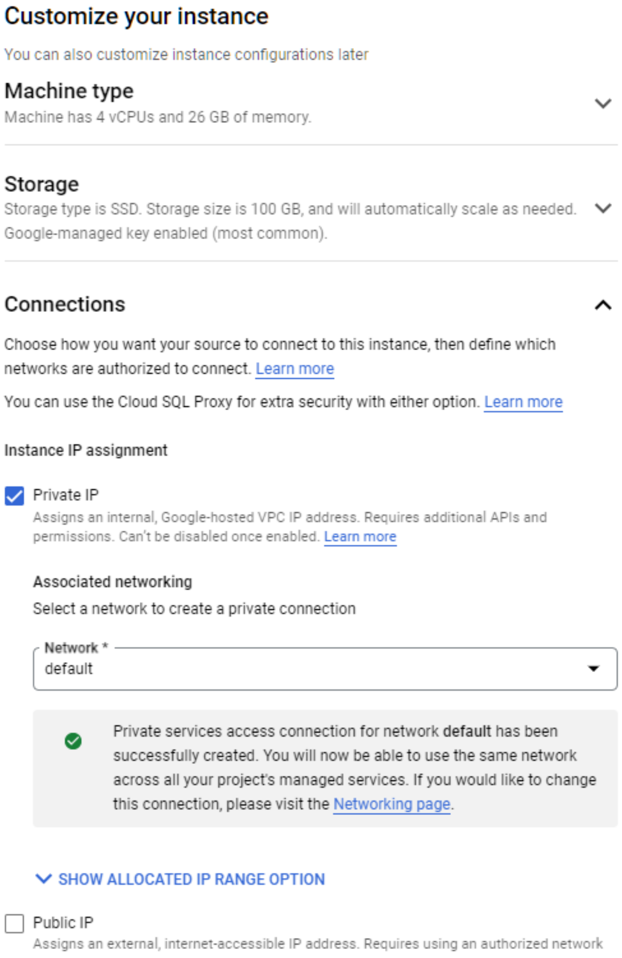
Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region: us-central1 (Iowa)

Zonal availability

○ Single zone
In case of outage, no failover. Not recommended for production.



Customize your instance

You can also customize instance configurations later

Machine type
Machine has 4 vCPUs and 26 GB of memory.

Storage
Storage type is SSD. Storage size is 100 GB, and will automatically scale as needed. Google-managed key enabled (most common).

Connections

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. Learn more

You can use the Cloud SQL Proxy for extra security with either option. Learn more

Instance IP assignment

☑ Private IP
Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. Learn more

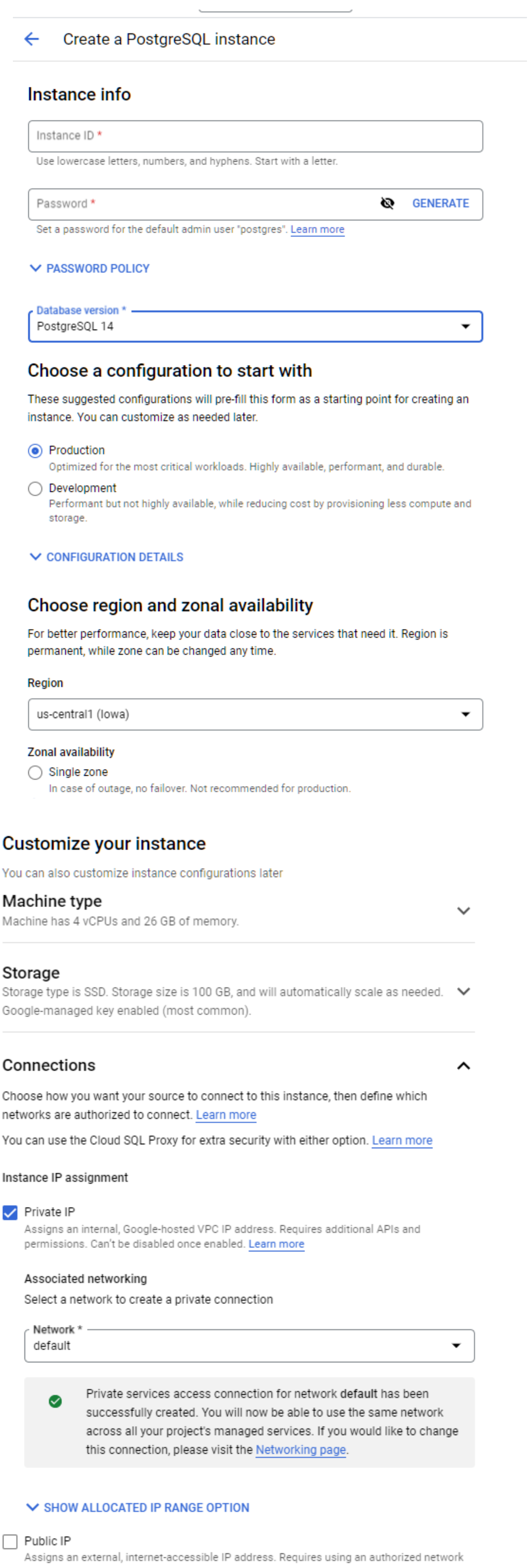
Associated networking
Select a network to create a private connection

Network *: default

✓ Private services access connection for network **default** has been successfully created. You will now be able to use the same network across all your project's managed services. If you would like to change this connection, please visit the Networking page.

SHOW ALLOCATED IP RANGE OPTION

□ Public IP
Assigns an external, internet-accessible IP address. Requires using an authorized network



← Create a PostgreSQL instance

Instance info

Instance ID *

Use lowercase letters, numbers, and hyphens. Start with a letter.

Password * GENERATE

Set a password for the default admin user "postgres". Learn more

PASSWORD POLICY

Database version *
PostgreSQL 14

Choose a configuration to start with

These suggested configurations will pre-fill this form as a starting point for creating an instance. You can customize as needed later.

● Production
Optimized for the most critical workloads. Highly available, performant, and durable.

○ Development
Performant but not highly available, while reducing cost by provisioning less compute and storage.

CONFIGURATION DETAILS

Choose region and zonal availability

For better performance, keep your data close to the services that need it. Region is permanent, while zone can be changed any time.

Region
us-central1 (Iowa)

Zonal availability

○ Single zone
In case of outage, no failover. Not recommended for production.

Customize your instance

You can also customize instance configurations later

Machine type
Machine has 4 vCPUs and 26 GB of memory.

Storage
Storage type is SSD. Storage size is 100 GB, and will automatically scale as needed. Google-managed key enabled (most common).

Connections

Choose how you want your source to connect to this instance, then define which networks are authorized to connect. Learn more

You can use the Cloud SQL Proxy for extra security with either option. Learn more

Instance IP assignment

☑ Private IP
Assigns an internal, Google-hosted VPC IP address. Requires additional APIs and permissions. Can't be disabled once enabled. Learn more

Associated networking

Select a network to create a private connection

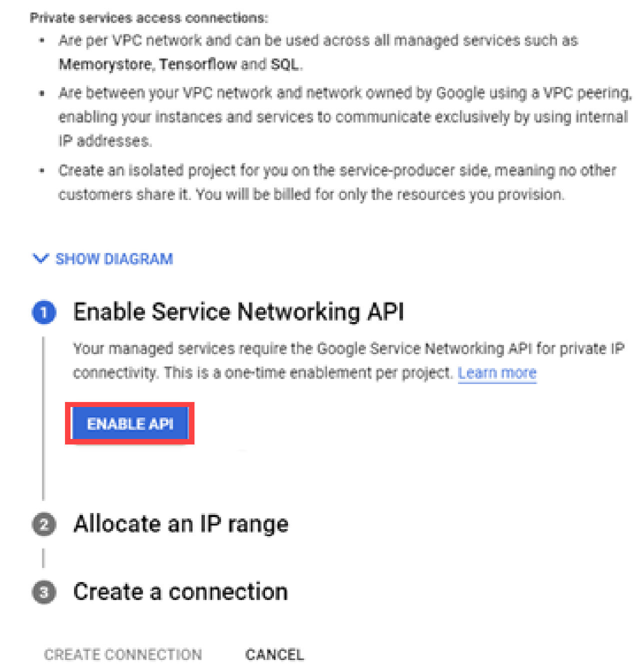
Network *
default

Private services access connection for network **default** has been successfully created. You will now be able to use the same network across all your project's managed services. If you would like to change this connection, please visit the Networking page.

SHOW ALLOCATED IP RANGE OPTION

□ Public IP
Assigns an external, internet-accessible IP address. Requires using an authorized network

Note that you will get asked to enable the below API's (if not already enabled). To do this, click on "ENABLE API"



Private services access connections:

- Are per VPC network and can be used across all managed services such as **Memorystore**, **Tensorflow** and **SQL**.
- Are between your VPC network and network owned by Google using a VPC peering, enabling your instances and services to communicate exclusively by using internal IP addresses.
- Create an isolated project for you on the service-producer side, meaning no other customers share it. You will be billed for only the resources you provision.

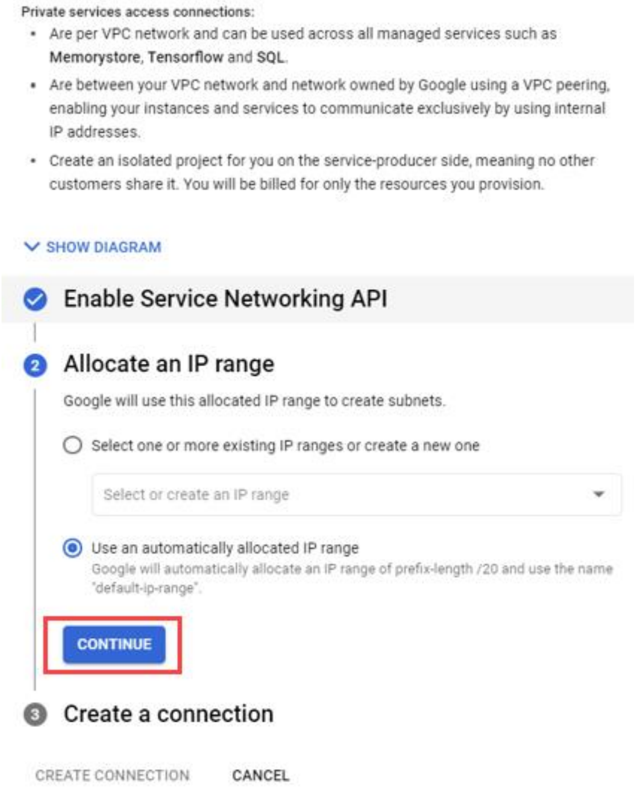
1 Enable Service Networking API

Your managed services require the Google Service Networking API for private IP connectivity. This is a one-time enablement per project. Learn more

2 Allocate an IP range

3 Create a connection

Leave on "use automatically allocated IP range" and click "continue."



Private services access connections:

- Are per VPC network and can be used across all managed services such as **Memorystore**, **Tensorflow** and **SQL**.
- Are between your VPC network and network owned by Google using a VPC peering, enabling your instances and services to communicate exclusively by using internal IP addresses.
- Create an isolated project for you on the service-producer side, meaning no other customers share it. You will be billed for only the resources you provision.

✓ Enable Service Networking API

2 Allocate an IP range

Google will use this allocated IP range to create subnets.

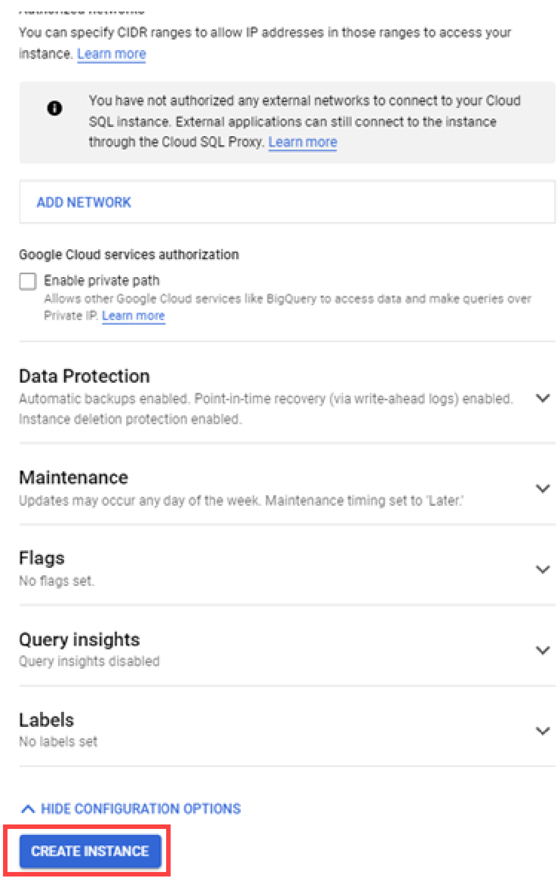
○ Select one or more existing IP ranges or create a new one

Select or create an IP range

● Use an automatically allocated IP range
Google will automatically allocate an IP range of prefix-length /20 and use the name "default-ip-range".

3 Create a connection

Finally, click on create instance. You then create a database (see the next step).



You can specify CIDR ranges to allow IP addresses in those ranges to access your instance. Learn more

You have not authorized any external networks to connect to your Cloud SQL instance. External applications can still connect to the instance through the Cloud SQL Proxy. Learn more

Google Cloud services authorization

□ Enable private path

Allows other Google Cloud services like BigQuery to access data and make queries over Private IP. Learn more

Data Protection

Automatic backups enabled. Point-in-time recovery (via write-ahead logs) enabled. Instance deletion protection enabled.

Maintenance

Updates may occur any day of the week. Maintenance timing set to 'Later.'

Flags

No flags set.

Query insights

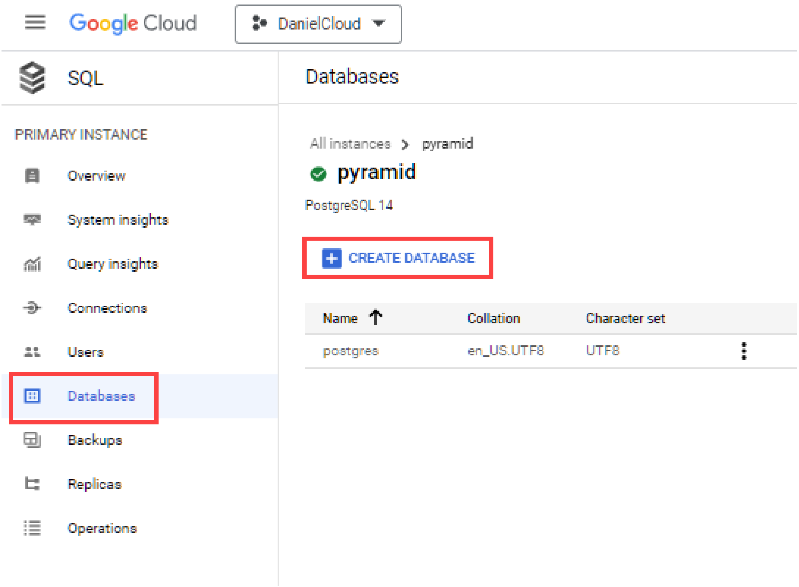
Query insights disabled

Labels

No labels set

3. Creating a new Pyramid Repository Database

Once the setup has completed, click on your new SQL instance and create a new blank database under SQL>Databases



Name ↑	Collation	Character set
postgres	en_US.UTF8	UTF8

To get the connection details click on overview and take the private IP of the instance. To connect using the username and password created on install of the instance.