

# INSTALACION CLUSTER Google Cloud

## 1. Entramos a Kubernetes Engine

Kubernetes Engine

Clústeres

Clústeres

Cargas de trabajo

Services e Ingress

Aplicaciones

Configuración

Almacenamiento

Navegador de objetos

Migrar a contenedores

Kubernetes Engine

Clústeres de Kubernetes

Los contenedores sirven para empaquetar aplicaciones, de modo que se puedan desplegar y ejecutar fácilmente en su propio entorno aislado. Se administran en clústeres que automatizan la creación y el mantenimiento de VMs. [Más información](#)

Crear clúster

Desplegar contenedor

Realizar el tutorial de inicio rápido

## 2. Clic a crear cluster

← Crear un clúster de Kubernetes

AÑADIR GRUPO DE NODOS

ELIMINAR GRUPO DE NODOS

Información básica de los clústeres

TIPOS DE NODOS

- default-pool

ÚSTER

- Automatización
- Redes
- Seguridad
- Metadatos
- Características

Información básica de los clústeres

El clúster nuevo tendrá el nombre, la versión y la ubicación que especifiques aquí. Después de crear el clúster, su nombre y ubicación no se pueden cambiar.

1

Para experimentar con un clúster asequible, prueba [Mi primer clúster en Guías de configuración de clústeres](#).

Nombre

cluster-1

Tipo de ubicación

☒ Zona

☐ Regional

Zona

us-central1-c

☐ Especificar ubicaciones de nodo predeterminadas

Valor predeterminado actual: us-central1-c

Versión maestra

Elige un canal de lanzamiento para que la versión y la periodicidad de las actualizaciones de tu clúster se gestionen automáticamente. Elige una versión estática para gestionar de forma más directa la versión de tu clúster. [Más información](#)

☒ Versión estática

☐ Canal de lanzamiento

Versión estática

1.16.13-gke.401 (predeterminado)

Guías de configuración de clústeres

Mi primer clúster

Un clúster asequible para experimentar

### 3. Configuración básica

Información básica de los clústeres

UPOS DE NODOS

- default-pool

CLÚSTER

- Automatización
- Redes
- Seguridad
- Metadatos
- Características

El clúster nuevo tendrá el nombre, la versión y la ubicación que especifiques aquí. Después de crear el clúster, su nombre y ubicación no se pueden cambiar.

1

Para experimentar con un clúster asequible, prueba **Mi primer clúster** en **Guías de configuración de clústeres**.

Nombre

sopes-cluster

Tipo de ubicación

☒ Zona

☐ Regional

Zona

us-central1-c

☐ Especificar ubicaciones de nodo predeterminadas

Valor predeterminado actual: us-central1-c

Versión maestra

Elige un canal de lanzamiento para que la versión y la periodicidad de las actualizaciones de tu clúster se gestionen automáticamente. Elige una versión estática para gestionar de forma más directa la versión de tu clúster. [Más información](#)

☐ Versión estática

☒ Canal de lanzamiento

Canal de lanzamiento

Canal habitual (predeterminado)

Versión

1.17.12-gke.1504(predeterminado)

Estas versiones han superado la validación interna y se considera que ofrecen calidad de producción. Sin embargo, el historial de datos no es lo suficientemente extenso como para asegurar su estabilidad. Los problemas conocidos suelen tener solución. [Notas de la versión](#)

CREAR

CANCELAR

[REST](#) o [línea de comando](#) equivalente

Guías de configuración de clúster

Mi primer clúster

Un clúster asequible para experimentar

#### 4. Detalle nodos

Google Cloud Platform

SOPES2

Buscar productos y recursos

← Crear un clúster de Kubernetes

+ AÑADIR GRUPO DE NODOS

ELIMINAR GRUPO DE NODOS

Información básica de los clústeres

POS DE NODOS

workers

Nodos

Seguridad

Metadatos

CLÚSTER

Automatización

Redes

Seguridad

Metadatos

Características

Detalles de grupo de nodos

El clúster se creará con al menos un grupo de nodos, que funciona como una plantilla para los grupos de nodos creados en este clúster. Después de crear el clúster, se pueden añadir o quitar grupos de nodos.

Nombre

workers

Versión maestra: 1.17.12-gke.1504

Tamaño

Número de nodos \*

1

El intervalo de direcciones del pod limita el tamaño máximo del clúster. [Más información](#)

☒ Habilitar autoescalado

Número mínimo de nodos \*

0

Número máximo de nodos \*

3

☐ Especificar ubicaciones de nodos

Predeterminado: us-central1-c

Automatización

☒ Habilitar actualización automática

Es posible que los clústeres con menos de 3 nodos experimenten periodos inactivos durante las actualizaciones

☒ Habilitar reparación automática

CREAR

CANCELAR

[REST o línea de comando equivalente](#)

## 5. Nodos

← Crear un clúster de Kubernetes

[+ AÑADIR GRUPO DE NODOS](#)

[ELIMINAR GRUPO DE NODOS](#)

Información básica de los clústeres

GRUPOS DE NODOS

- workers
  - Nodos**
  - Seguridad
  - Metadatos


CLÚSTER

- Automatización
- Redes
- Seguridad
- Metadatos
- Características

Serie  
E2

La plataforma de CPU se elige según las que haya disponibles

Tipo de máquina  
e2-medium (2 vCPU, 4 GB de memoria)



**vCPU**  
1 núcleo compartido

**Memory**  
4 GB

▼ PLATAFORMA DE CPU Y GPU

Tipo de disco de arranque  
Disco persistente estándar

Tamaño de disco de arranque (GB)  
20

☐ Habilitar el encriptado del disco de arranque gestionado por el cliente

Discos SSD locales

☐ Habilitar nodos interrumpibles

**Redes**

En la configuración del clúster, se especifica un máximo de 110 pods por nodo, pero puedes anular este ajuste a nivel de grupo de nodos.

Número máximo de pods por nodo

Máscara para el intervalo de direcciones de pods por nodo: /24

Etiquetas de red  
todoin todoout

CREAR

CANCELAR

[REST o línea de comando](#) equivalente

## 6. Seguridad

Google Cloud Platform

SOPES2

Buscar productos y recursos

← Crear un clúster de Kubernetes

[+ AÑADIR GRUPO DE NODOS](#)

[ELIMINAR GRUPO DE NODOS](#)

Información básica de los clústeres

GRUPOS DE NODOS

- workers
  - Nodos
  - Seguridad**
  - Metadatos

CLÚSTER

- Automatización
- Redes
- Seguridad
- Metadatos
- Características

**Seguridad de los nodos**

Esta configuración de seguridad de nodos se utilizará cuando se creen nodos con este grupo de nodos.

Cuenta de servicio  
Compute Engine default service account

**Permisos de acceso**

☐ Permitir el acceso predeterminado

☒ Permitir el acceso completo a todas las API de Cloud

☐ Definir el acceso a cada API

☐ Habilitar zona de pruebas con gVisor

**Opciones blindadas**

☒ Habilitar monitorización de integridad

☐ Habilitar arranque seguro

## CREAR MAQUINA VIRTUAL

### 7. INSTALAR KUBECTL

- a. `wget curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.18.0/bin/linux/amd64/kubectl`
- b. `chmod +x ./kubectl`
- c. `sudo mv ./kubectl /usr/local/bin/kubectl`
  - i. REF: <https://kubernetes.io/docs/tasks/tools/install-kubectl/>

### 8. CREAR Y CONFIGURAR CLUSTER EN GOOGLE

- a. `curl -O https://dl.google.com/dl/cloudsdk/channels/rapid/downloads/google-cloud-sdk-297.0.0-linux-x86\_64.tar.gz`
- b. `tar zxvf [ARCHIVE_FILE] google-cloud-sdk`
  - i. `tar zxvf google-cloud-sdk-297.0.0-linux-x86_64.tar.gz`
- c. `./google-cloud-sdk/install.sh`
- d. logout and login to reload the new gcloud command (Verificamos si funciona el commando gcloud)

e. `gcloud init`<<Seguir instrucciones de pantalla>>

```
Choose the account you would like to use to perform operations for
this configuration:
[1] 1029710161861-compute@developer.gserviceaccount.com
[2] Log in with a new account
Please enter your numeric choice: 2

You are running on a Google Compute Engine virtual machine.
It is recommended that you use service accounts for authentication.

You can run:

  $ gcloud config set account 'ACCOUNT'

to switch accounts if necessary.

Your credentials may be visible to others with access to this
virtual machine. Are you sure you want to authenticate with
your personal account?

Do you want to continue (Y/n)? y

Go to the following link in your browser:

  https://accounts.google.com/o/oauth2/auth?response_type=code&client_id=32555940559.apps.googleusercontent.com
  scope=openid+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fuserinfo.email+https%3A%2F%2Fwww.googleapis.com%2Fauth%2F
  cloud-platform+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww.googleapis.com%2Fauth%
  Fcompute+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Faccounts.reauth&redirect_uri=urn%3Aietf%3Awg%3Aoauth%3A2.0%3A
  ob&code_challenge=lv297kUbZjzuZjpG06SZRIQoDcdrcEAmWht83oT6UG0&access_type=offline&prompt=select_account&code_cha
  llenge_method=S256

Enter verification code: █
```

- i. Elegimos loguearnos con una cuenta nueva
- ii. Le damos aceptar y nos tirará un enlace, el cual entramos, con la cuenta con la que tenemos creadas nuestras maquinas de gcloud
- iii. Después de loguearnos nos dará un código, el cual copiamos y pegamos en donde nos indica.



Acceder

Copia este código, cambia a tu aplicación y pégalo allí:

4/1AY0e-  
g7Yt1jMrPHWBCMMUwS79xSHRISLygW9UkiHJl3bJlZl8f0

```
Enter verification code: 4/1AY0e-g7Yt1jMrPHWBcMMUwS79xSHRISLygW9UkiHJl3bJLZlBf0GiqgN4zc
You are logged in as: [2618007790101@ingenieria.usac.edu.gt].

Pick cloud project to use:
[1] solt3-280619
[2] sopas2-295218
[3] Create a new project
Please enter numeric choice or text value (must exactly match list
item):
```

Ahora estamos logueados y listos para continuar.

```
Enter verification code: 4/1AY0e-g7Yt1jMrPHWBcMMUwS79xSHRISLygW9UkiHJl3bJLZlBf0GiqgN4zc
You are logged in as: [2618007790101@ingenieria.usac.edu.gt].

Pick cloud project to use:
[1] solt3-280619
[2] sopas2-295218
[3] Create a new project
Please enter numeric choice or text value (must exactly match list
item): 2

Your current project has been set to: [sopas2-295218].

Do you want to configure a default Compute Region and Zone? (Y/n)? y
```

Elegimos el proyecto, en este caso es el proyecto 2 y configuramos la región

En este caso escogemos la misma región en la que creamos el cluster

us-central1-c

y finalizamos con la instalación

```
did not print [24] options.
Too many options [74]. Enter "list" at prompt to print choices fully.
Please enter numeric choice or text value (must exactly match list
item): 7

Your project default Compute Engine zone has been set to [us-central1-c].
You can change it by running [gcloud config set compute/zone NAME].

Your project default Compute Engine region has been set to [us-central1].
You can change it by running [gcloud config set compute/region NAME].

Created a default .boto configuration file at [/home/g2618007790101/.boto]. See this file and
https://cloud.google.com/storage/docs/gsutil/commands/config for more
information about configuring Google Cloud Storage.
Your Google Cloud SDK is configured and ready to use!

Commands that require authentication will use 2618007790101@ingenieria.usac.edu.gt by default
Commands will reference project 'sopes2-295218' by default
Compute Engine commands will use region 'us-central1' by default
Compute Engine commands will use zone 'us-central1-c' by default

Run 'gcloud help config' to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple
accounts and/or projects.
Run 'gcloud topic configurations' to learn more.

Some things to try next:

Run 'gcloud --help' to see the Cloud Platform services you can interact with. And run 'gcloud help COMMAND' to
get help on any gcloud command.
Run 'gcloud topic --help' to learn about advanced features of the SDK like arg files and output formatting
2618007790101@backend:~$
```

iv. referencia: <https://cloud.google.com/sdk/docs/quickstart-linux>

## CONFIGURAR KUBECTL PARA ACCEDER AL CLUSTER

- f. Vamos al cluster creado anteriormente, entramos y en la opción que dice Endpoint le damos clic a —ver certificado— Clusters->MyCluster->Details->Show Cluster Certificate





## CREAR PODS

### CREAR NAMESPACE

Kubectl create ns proyecto

Instalamos Helm

wget <https://get.helm.sh/helm-v3.4.0-linux-amd64.tar.gz>

tar -xzf helm-v3.4.0-linux-amd64.tar.gz

sudo mv linux-amd64/helm /sbin

agregar repositorio generic de Helm

helm repo add stable https://charts.helm.sh/stable

busca los paquetes

helm search repo stable

Buscamos el paquete nginx, el cual nos va a servir para publicar nuestra app ya sea backend o frontend y lo instalamos en nuestro namespace creado anteriormente

helm install nginx-ingress stable/nginx-ingress -n proyecto

para desinstalarlo solo ejecutamos el siguiente comando

helm uninstall nginx-ingress -n nginx-ingress

helm list -n proyecto

Verificamos la instalación con los pods y los servicios

kubectl get pods -n proyecto

```
g2618007790101@backend:~$ kubectl get pods -n proyecto
NAME                                READY   STATUS    RESTARTS   AGE
nginx-ingress-controller-85fcf4d47c-kqfck   1/1     Running   0           64s
nginx-ingress-default-backend-5b967cf596-h4j1h 1/1     Running   0           64s
g2618007790101@backend:~$ kubectl get services -n proyecto
```

kubectl get services -n proyecto

```
g2618007790101@backend:~$ kubectl get services -n proyecto
NAME                                TYPE           CLUSTER-IP   EXTERNAL-IP   PORT(S)                                     AGE
nginx-ingress-controller           LoadBalancer   10.6.219.174  34.121.190.143  80:30028/TCP, 443:31874/TCP              90s
nginx-ingress-default-backend      ClusterIP       10.6.219.104  <none>        80/TCP                                    90s
g2618007790101@backend:~$
```

La ip externa del balanceador de carga es el que nos va a servir para publicar nuestra app

## CREAR IMAGE DOCKER DE NUESTRAS APPS

*Descargamos nuestra app desde github a nuestra maquina virtual*

*Instalamos Docker*

`Sudo apt install docker.io`

*Entramos a la carpeta del proyecto que queremos publicar y ejecutamos*

`sudo sh build.sh`

Loguearse como sudo, para evitar escribir siempre sudo antes de cada comando

Nos logueamos a Docker

Docker login

push imagen a docker

`sudo docker push wildering/backend-image-so2`

```
root@backend:/home/g2618007790101/proyecto_grupo31_so2/backend# sudo docker push wildering/backend-image-so2
The push refers to repository [docker.io/wildering/backend-image-so2]
c18e83b5c5dd: Pushed
5253f630fcb6: Pushed
4ffb5f425c49: Pushed
cdaf513b2fba: Pushed
19571ff21ce3: Mounted from library/node
579834dd79ff: Mounted from library/node
0b6541f04a0d: Mounted from library/node
3211c12c1c23: Mounted from library/node
1d3ec06e3d4f: Mounted from library/node
9e5330403dba: Mounted from library/node
3bd20dc0b7e5: Mounted from library/node
94b70b410c2a: Mounted from library/node
3567db1eb737: Mounted from library/node
latest: digest: sha256:1defdcc8f65d6997760d08750326de476d3b23b95d60a222baf5ea3914aece18 size: 3051
root@backend:/home/g2618007790101/proyecto_grupo31_so2/backend#
```

Publicar en Kubernetes

*Creamos del deployment –Nota: este deployment se crea con archivo yaml ya que se crean tres replicas, si no se desean crear replicas puede hacerse un deployment con el siguiente comando :*

*kubectl create deployment miapp-python-n --image=wildering/python-nats -n proyecto obteniendo la imagen de Docker creada anteriormente.*

```
kubectl create -f ingress-backend.yaml
```

Crear el service

`kubectl expose deployment backend-deploy --name=miapp-backend-gnix-srv --port=3000 --target-port=3000 --type=LoadBalancer -n proyecto`

*verificamos que se haya creado exitosamente tanto el servicio como el proyecto*

kubectl get pods -n proyecto

kubectl get services -n proyecto

Crear el Ingress ya sea para nginx o para Contour

Una vez creado lo ejecutamos

kubectl create -f ingress-nginx.yaml

enlazamos la ip de nuestro loadbalancer con nuestro dns creado anteriormente.

Creamos el deploy del front end

```
kubectl create -f deploy-frontend.yaml
```

kubectl expose deployment frontend-deploy --name=miapp-frontend-nginx-srv --port=4000 --target-port=4000 --type=LoadBalancer -n proyecto

Creamos el ingress

kubectl create -f ingress-nginx.yaml

BACKEND

<http://backend.proy2so2.tk>

FRONTEND

<http://frontend.proy2so2.tk>

<http://34.122.182.4:4000/all>