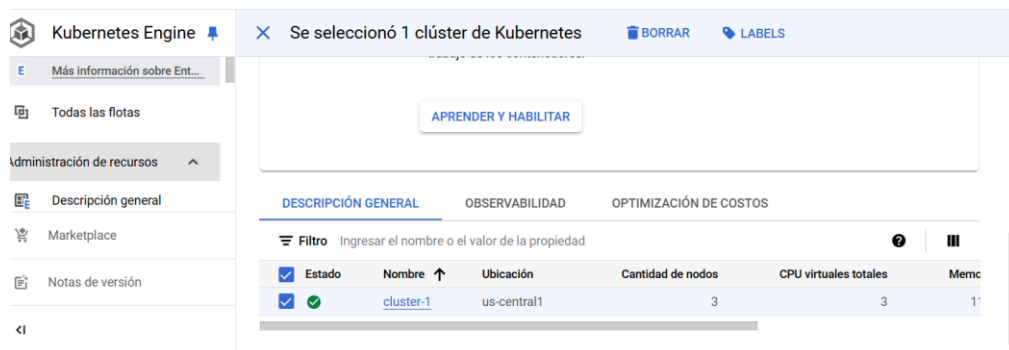
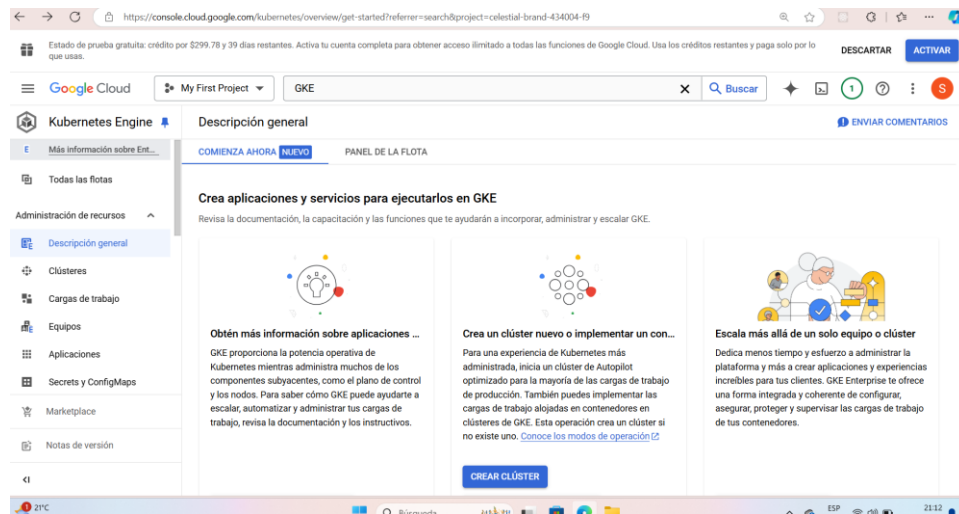


Creación de clúster



Creación de main.go y dockerfile

```
Your Cloud Platform project in this session is set to celestial-brand-434004-f9.
Welcome to Cloud Shell! Type "help" to get started.
Your Cloud Platform project in this session is set to celestial-brand-434004-f9.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
sharotago@cloudshell:~ (celestial-brand-434004-f9) $ gcloud container clusters get-credentials cluster-1 --zone us-central1
Fetching cluster endpoint and auth data.
kubeconfig entry generated for cluster-1.
sharotago@cloudshell:~ (celestial-brand-434004-f9) $ mkdir go-app
sharotago@cloudshell:~ (celestial-brand-434004-f9) $ cd go-app
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9) $ nano main.go
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9) $ nano Dockerfile
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9) $ gcloud auth configure-docker
```

```
GNU nano 7.2                                main.go
package main

import (
    "encoding/json"
    "fmt"
    "log"
    "net/http"
)

type Student struct {
    Student string `json:"student"`
    Age     int    `json:"age"`
    Faculty string `json:"faculty"`
    Discipline int  `json:"discipline"`
}

func handleData(w http.ResponseWriter, r *http.Request) {
    var student Student
    err := json.NewDecoder(r.Body).Decode(&student)
    if err != nil {
        http.Error(w, err.Error(), http.StatusBadRequest)
        return
    }
    fmt.Fprintf(w, "Received: %v\n", student)
}

func main() {
    http.HandleFunc("/api-endpoint", handleData)
    log.Println("Server starting on port 8080...")
    log.Fatal(http.ListenAndServe(":8080", nil))
}

[ Read 31 lines ]
^G Help      ^O Write Out ^W Where Is  ^R Cut       ^T Execute   ^C Location  M-U Undo     M-A Set Mark
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo     M-G Copy
```

CLOUD SHELL

Terminal (celestial-brand-434004-f9) x + ▾ [Abrir editor](#)

```
GNU nano 7.2                                Dockerfile
# Usa una imagen base de Go
FROM golang:1.19-alpine

# Establece el directorio de trabajo en el contenedor
WORKDIR /app

# Copia el archivo de la aplicación al contenedor
COPY . .

# Compila la aplicación Go
RUN go build -o main .

# Expone el puerto 8080
EXPOSE 8080

# Comando para ejecutar cuando el contenedor se inicie
CMD ["./main"]
```

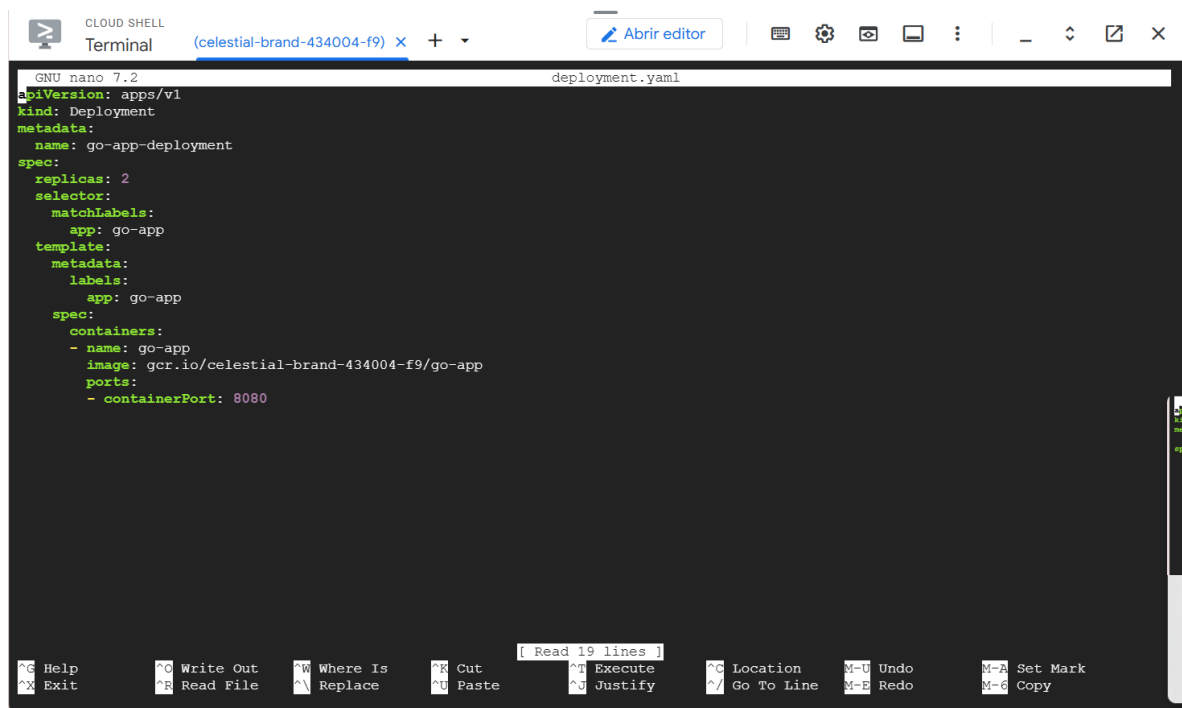
Construyendo imagen

```
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9)$ docker build -t gcr.io/$(gcloud config get-value project)/go-app .
Your active configuration is: [cloudshell-8733]
[+] Building 2.1s (9/9) FINISHED
=> [internal] load build definition from Dockerfile                                docker:default
=> => transferring dockerfile: 380B                                              0.0s
=> [internal] load metadata for docker.io/library/golang:1.19-alpine             0.0s
=> [internal] load .dockerignore                                                 0.0s
=> => transferring context: 2B                                                  0.0s
=> [1/4] FROM docker.io/library/golang:1.19-alpine@sha256:0ec0646e208ea58e5d29e558e39f2e59fccf39b7bda306cb53bbaff91919eca5 0.0s
=> [internal] load build context                                                0.0s
=> => transferring context: 117B                                                0.0s
=> CACHED [2/4] WORKDIR /app                                                    0.0s
=> [3/4] COPY . .                                                              0.0s
=> [4/4] RUN go build -o main .                                                1.1s
=> exporting to image                                                           0.2s
=> => exporting layers                                                         0.1s
=> => writing image sha256:66180f1673a080eef656a0ebbe08f75c54118584c08da71e24284252d14c51bd 0.0s
=> => naming to gcr.io/celestial-brand-434004-f9/go-app                       0.0s
```

Subiendo imagen

```
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9)$ docker push gcr.io/$(gcloud config get-value project)/go-app
Your active configuration is: [cloudshell-8733]
Using default tag: latest
The push refers to repository [gcr.io/celestial-brand-434004-f9/go-app]
7cd069240157: Pushed
e333606d37c0: Pushed
372334916228: Pushed
ffbff62e3514: Layer already exists
9bfd5e9c96bf: Layer already exists
f4285c491509: Layer already exists
4693057ce236: Layer already exists
latest: digest: sha256:186d60ccb4b865a888c0759960bceda92ac55fedfbf51b0d930c81007116508f size: 1782
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9)$
```

Creación de yaml



```
GNU nano 7.2 deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: go-app-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      app: go-app
  template:
    metadata:
      labels:
        app: go-app
    spec:
      containers:
      - name: go-app
        image: gcr.io/celestial-brand-434004-f9/go-app
        ports:
        - containerPort: 8080
```

Prueba con envío de datos del json

```
Received: {Student:Arvalo Garcia Age:20 Faculty:Ingenieria Discipline:1}
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9)$ curl -X POST http://35.238.237.253:8080/api-endpoint \
-H "Content-Type: application/json" \
-d '{"student": "Sharon Tagual", "age": 23, "faculty": "Ingeniería", "discipline": 1}'
Received: {Student:Sharon Tagual Age:23 Faculty:Ingenieria Discipline:1}
sharotago@cloudshell:~/go-app (celestial-brand-434004-f9)$
```