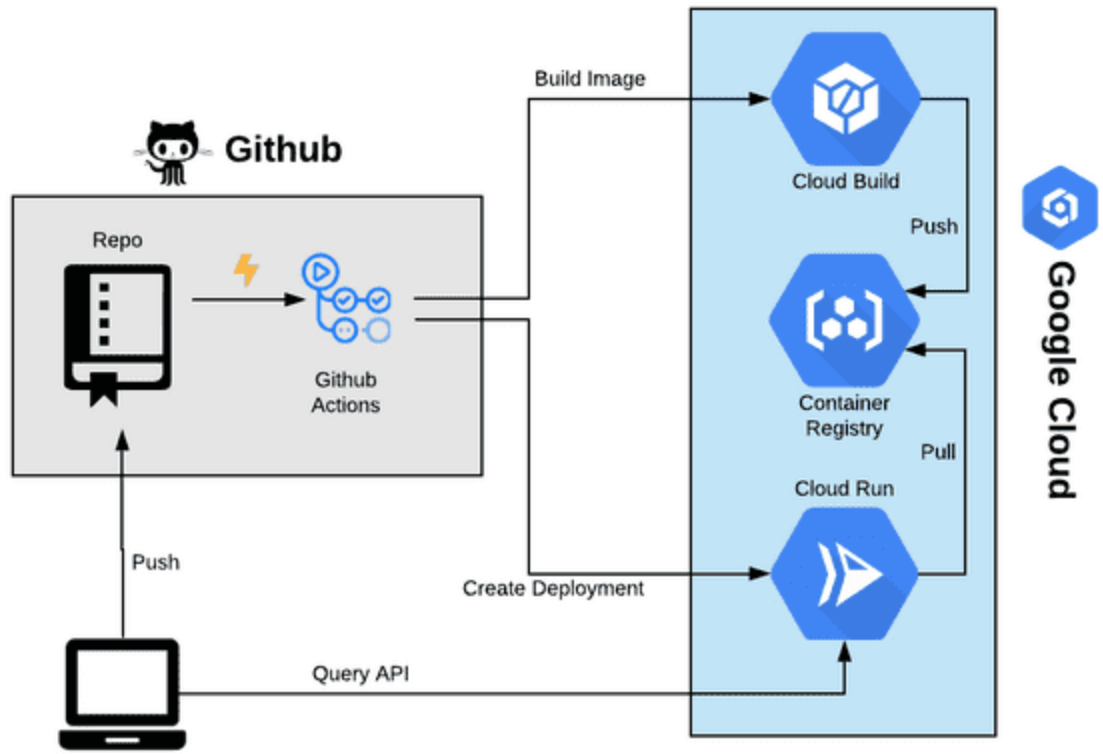
Introducción a los despliegues automáticos con GitHub Actions

- Vamos a automatizar el despliege del API.
- Para ello usaremos GitHub Actions, Google Cloud Build y Cloud Run.
- GitHub Actions provisiona una máquina en la que podemos ejecutar una serie de acciones cuando ocurre algún evente en nuestro repositorio.
- En nuestro claso desplegaremos una nueva versión cuando hagamos un push a main.


Github
Repo
Github
Actions
Push
Query API
Build Image
Create Deployment
Cloud Build
Push
Container
Registry
Pull
Cloud Run
Google Cloud

- Definimos el proceso de despliege en una fichero YAML:

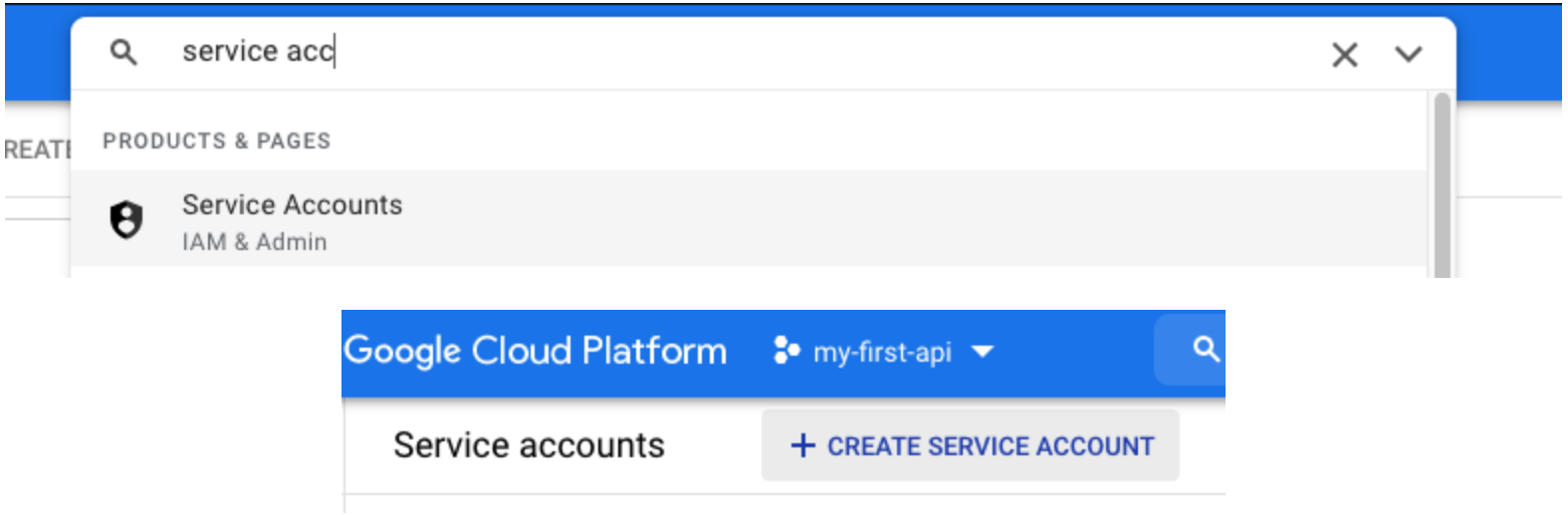
```yaml
name: Build and Deploy to Cloud Run

on:
  push:
    branches:
    - main

env:
  PROJECT_ID: ${{ secrets.RUN_PROJECT }}
  RUN_REGION: europe-west1
  SERVICE_NAME: api-cloud-run

jobs:
  setup-build-deploy:
    name: Setup, Build, and Deploy to Cloud Run
    runs-on: ubuntu-latest

    steps:
    - name: Checkout
      uses: actions/checkout@v2

    # Setup gcloud CLI
    - uses: GoogleCloudPlatform/github-actions/setup-gcloud@master
      with:
        version: '290.0.1'
        service_account_key: ${{ secrets.RUN_SA_KEY }}
        project_id: ${{ secrets.RUN_PROJECT }}

    # Build and push image to Google Container Registry
    - name: Build
      run: |-
        gcloud builds submit \
          --quiet \
          --tag "gcr.io/$PROJECT_ID/$SERVICE_NAME:$GITHUB_SHA"

    # Deploy image to Cloud Run
    - name: Deploy
      run: |-
        gcloud run deploy "$SERVICE_NAME" \
          --quiet \
          --region "$RUN_REGION" \
          --image "gcr.io/$PROJECT_ID/$SERVICE_NAME:$GITHUB_SHA" \
          --platform "managed" \
          --allow-unauthenticated
```

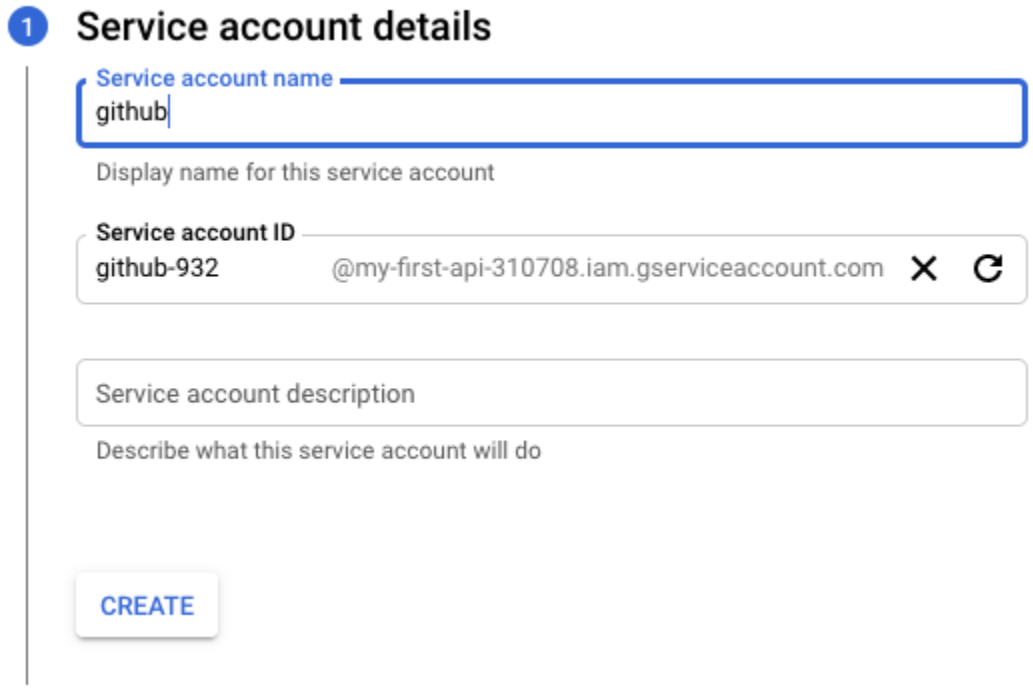
- Este fichero tiene que estar en la carpeta .github/workflows.

- El proceso necesita dos variables de entorno:
  - RUN_PROJECT: Nombre del proyecto donde vamos a desplegar nuestra aplicación.
  - RUN_SA_KEY: Key de un service account con los siguientes permisos:
    - Service Account User
    - Cloud Build Editor
    - Cloud Run Admin
    - Viewer

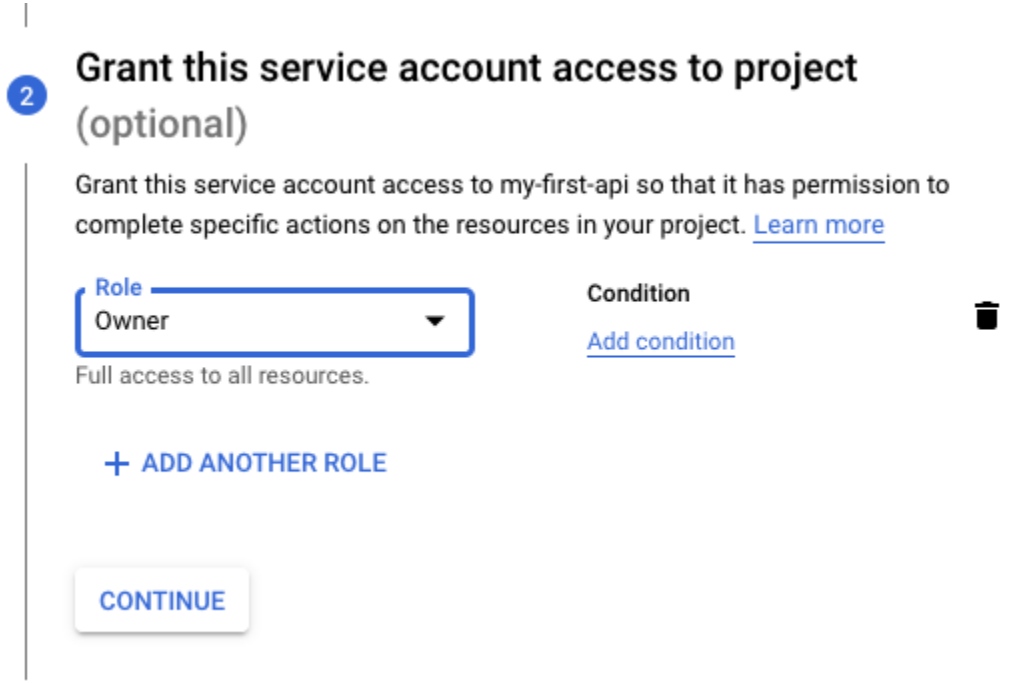
Ejemplo

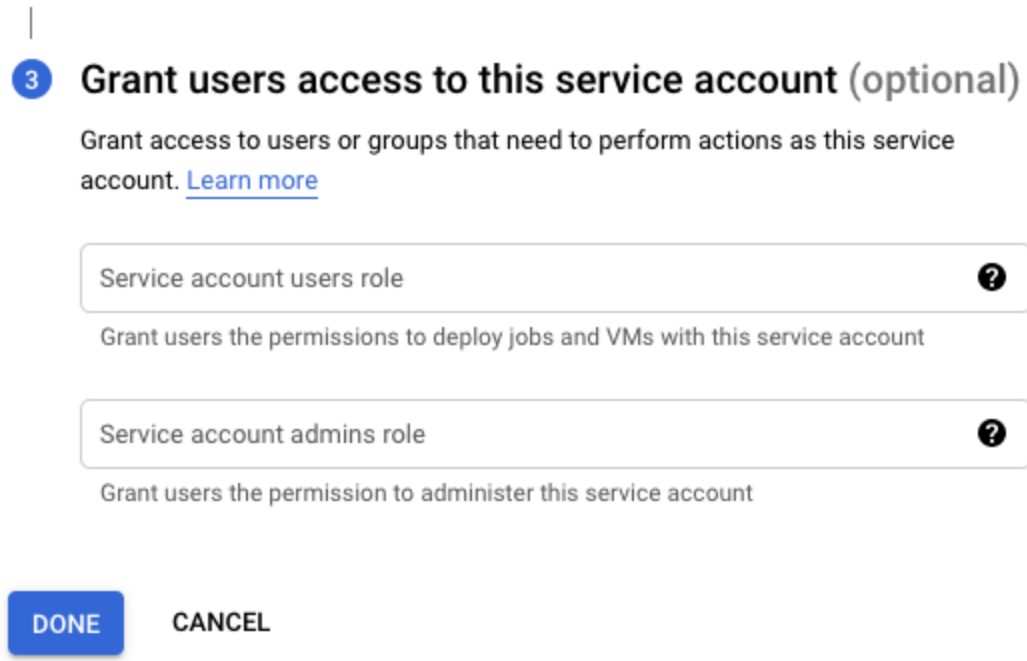
- Primero generamos el service account con los permisos necesarios.



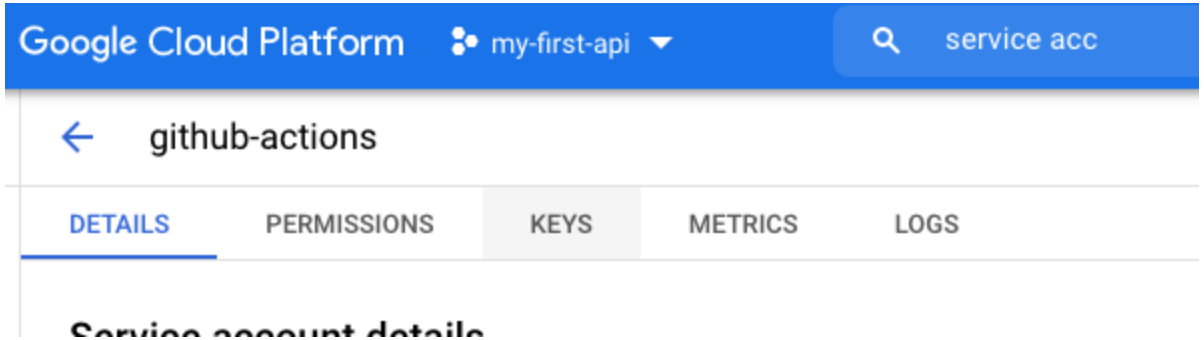

1
Service account details
Service account name
github
Display name for this service account
Service account ID
github-932
@my-first-api-310708.iam.gserviceaccount.com
Service account description
Describe what this service account will do
CREATE

- Podemos poner el permiso de owner aunque no es lo correcto.




3
Grant users access to this service account (optional)
Grant access to users or groups that need to perform actions as this service account. Learn more
Service account users role
Grant users the permissions to deploy jobs and VMs with this service account
Service account admins role
Grant users the permission to administer this service account
DONE
CANCEL

- Generamos el json con la key del service account.



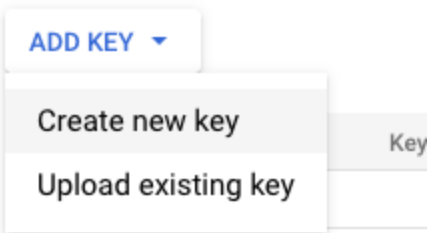
Keys

Add a new key pair or upload a public key certificate from an existing key pair. Please note that public certificates need to be in RSA_X509_PEM format. Learn more about upload key formats

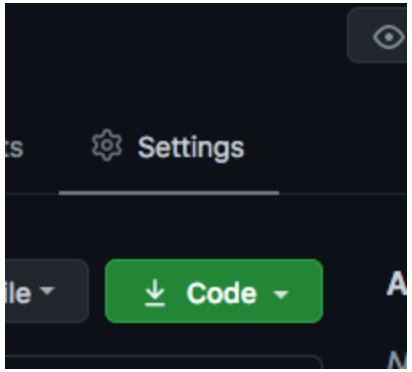
Block service account key creation using organization policies.
Learn more about setting organization policies for service accounts

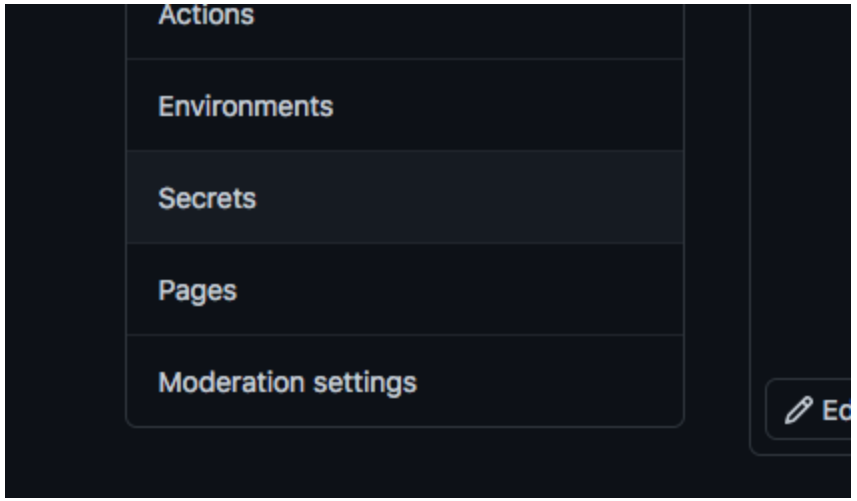
ADD KEY

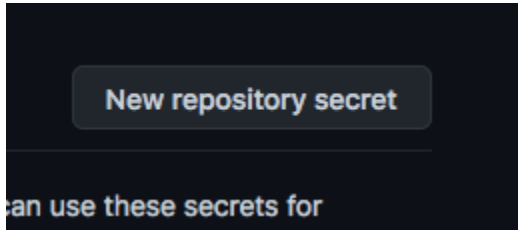
Type	Status	Key	Key creation date	Key expiration date
No rows to display				

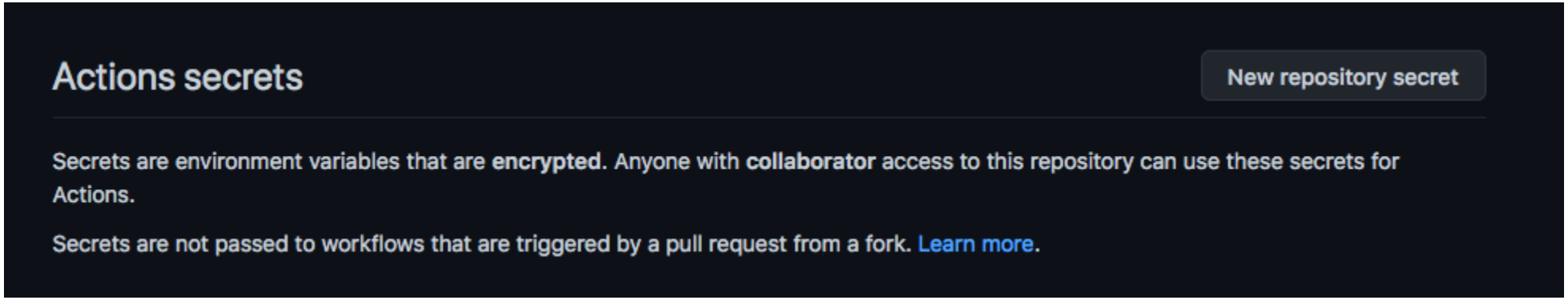


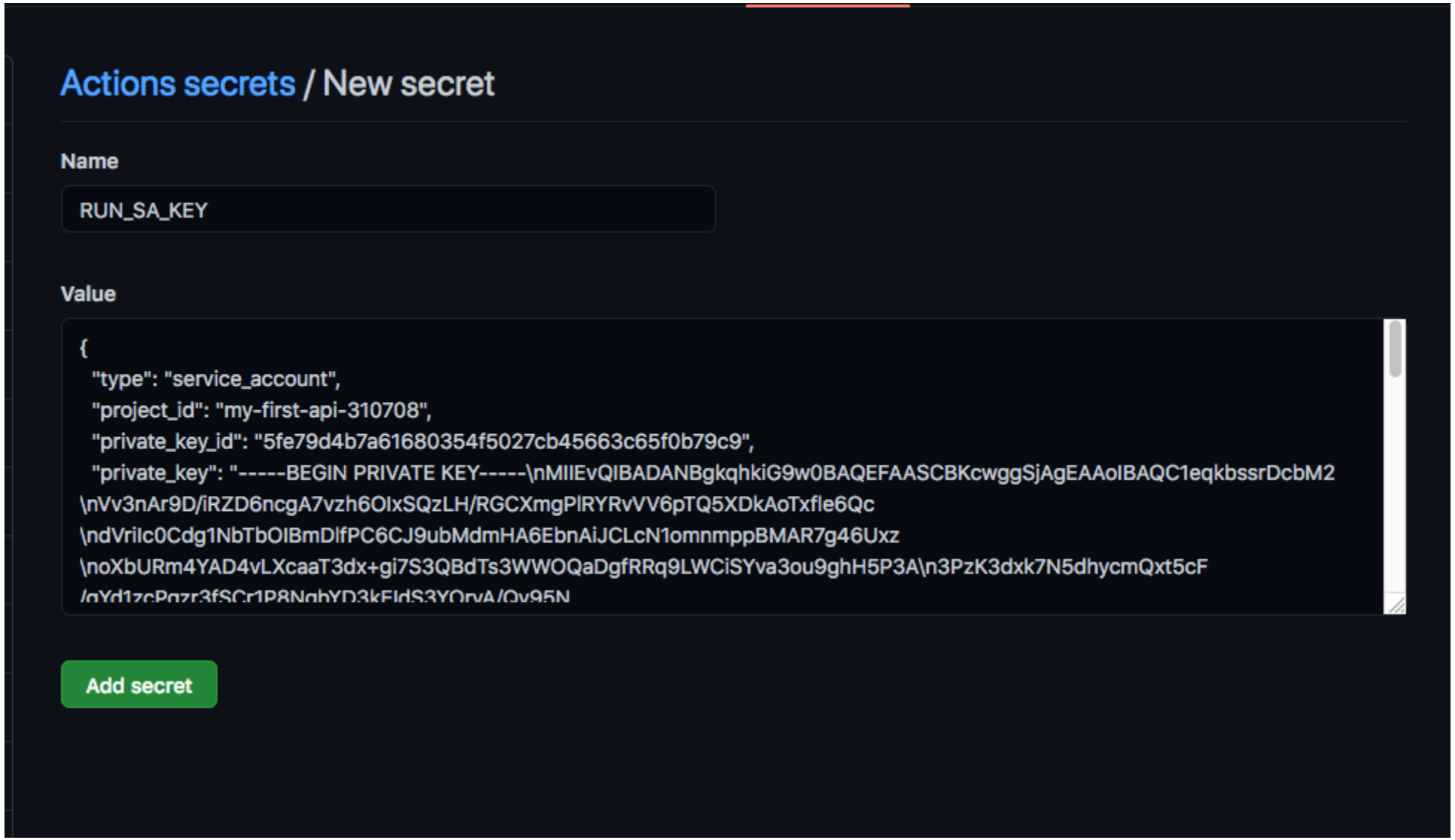
- Añadimos las variables a GitHub.

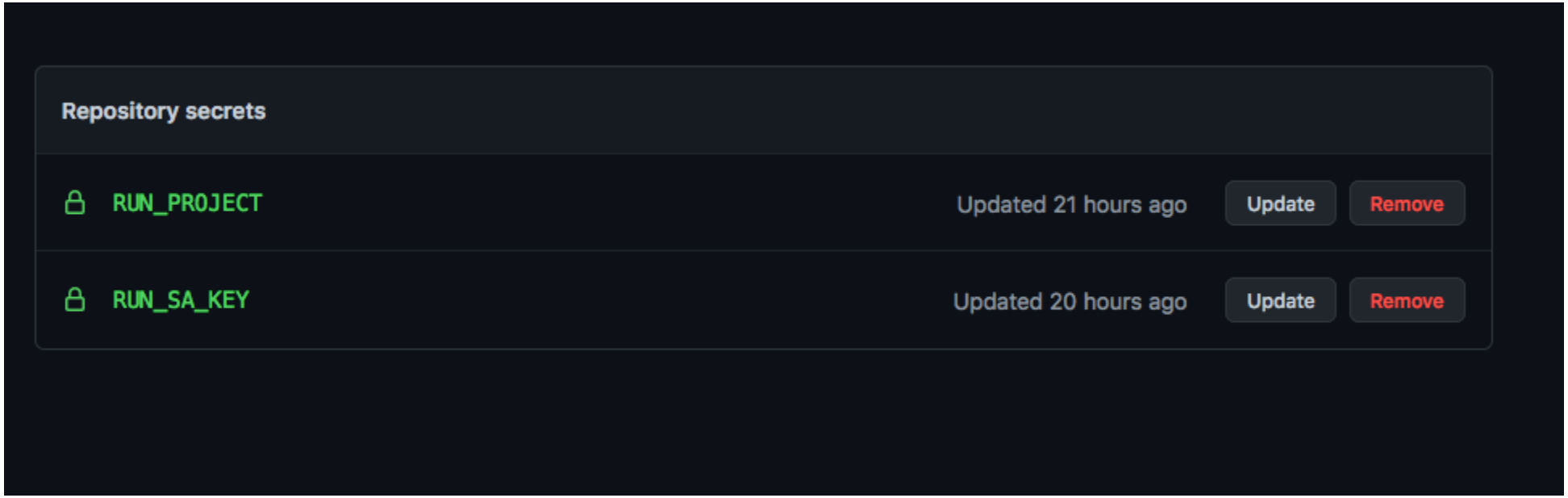




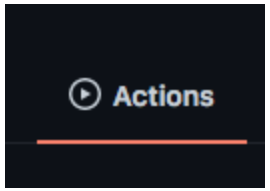

New repository secret
an use these secrets for

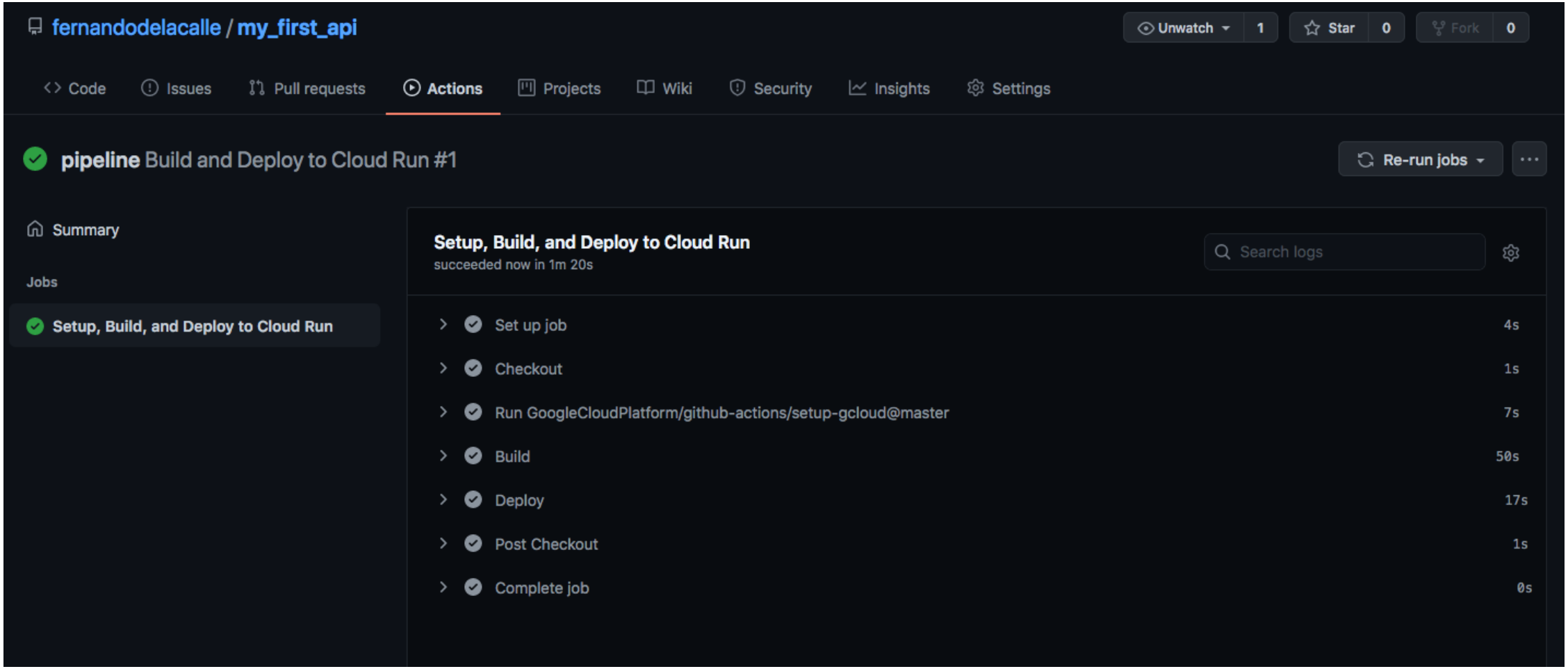

Actions secrets
New repository secret
Secrets are environment variables that are encrypted. Anyone with collaborator access to this repository can use these secrets for Actions.
Secrets are not passed to workflows that are triggered by a pull request from a fork. Learn more.


Actions secrets / New secret
Name
RUN_SA_KEY
Value
{
"type": "service_account",
"project_id": "my-first-api-310708",
"private_key_id": "5fe79d4b7a61680354f5027cb45663c65f0b79c9",
"private_key": "-----BEGIN PRIVATE KEY-----\nMIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQC1eqkbssrDcbM2
\nVv3nAr9D/iRZD6ncgA7vzh6OIxSQzLH/RGCXmgPlRYRvVV6pTQ5XDkAoTxfle6Qc
\ndVriIc0Cdg1NbTbOIBmDlfPC6CJ9ubMdmHA6EbnAiJCLcN1omnmppBMAR7g46Uxz
\noXbURm4YAD4vLXcaaT3dx+gi7S3QBdTs3WWOQaDgfRRq9LWCiSYva3ou9ghH5P3A\n3PzK3dxk7N5dhycmQxt5cF
Add secret

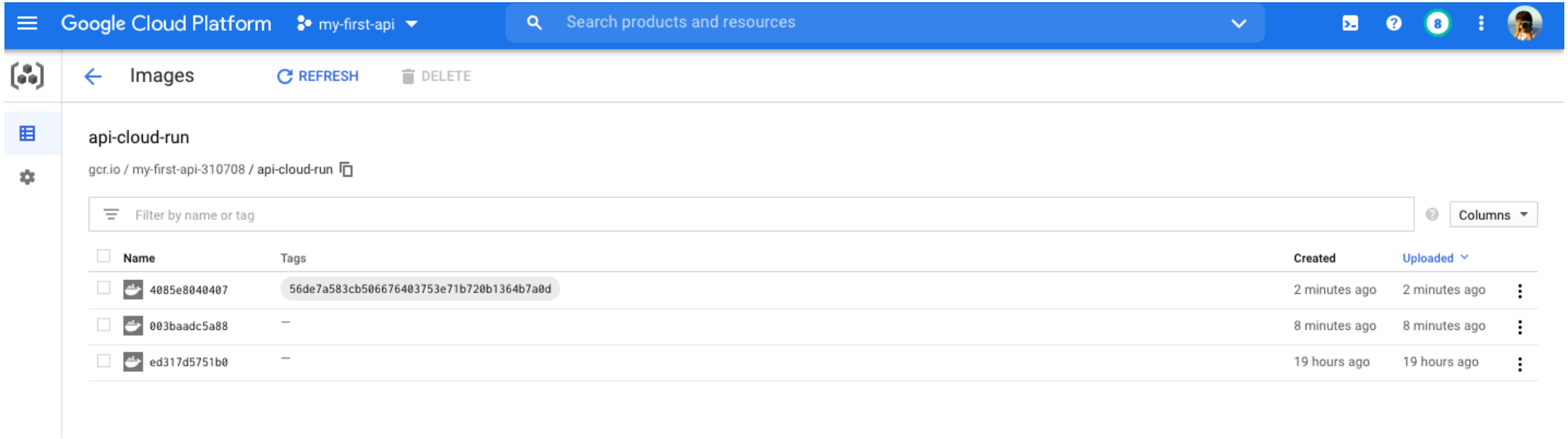

Repository secrets
RUN_PROJECT
Updated 21 hours ago
Update
Remove
RUN_SA_KEY
Updated 20 hours ago
Update
Remove

- Añadimos el fichero al repositorio.
- Si todo está correcto al hacer el push la aplicación se deplegará automáticamente.
- Podemos verlo en el menú de github actions.

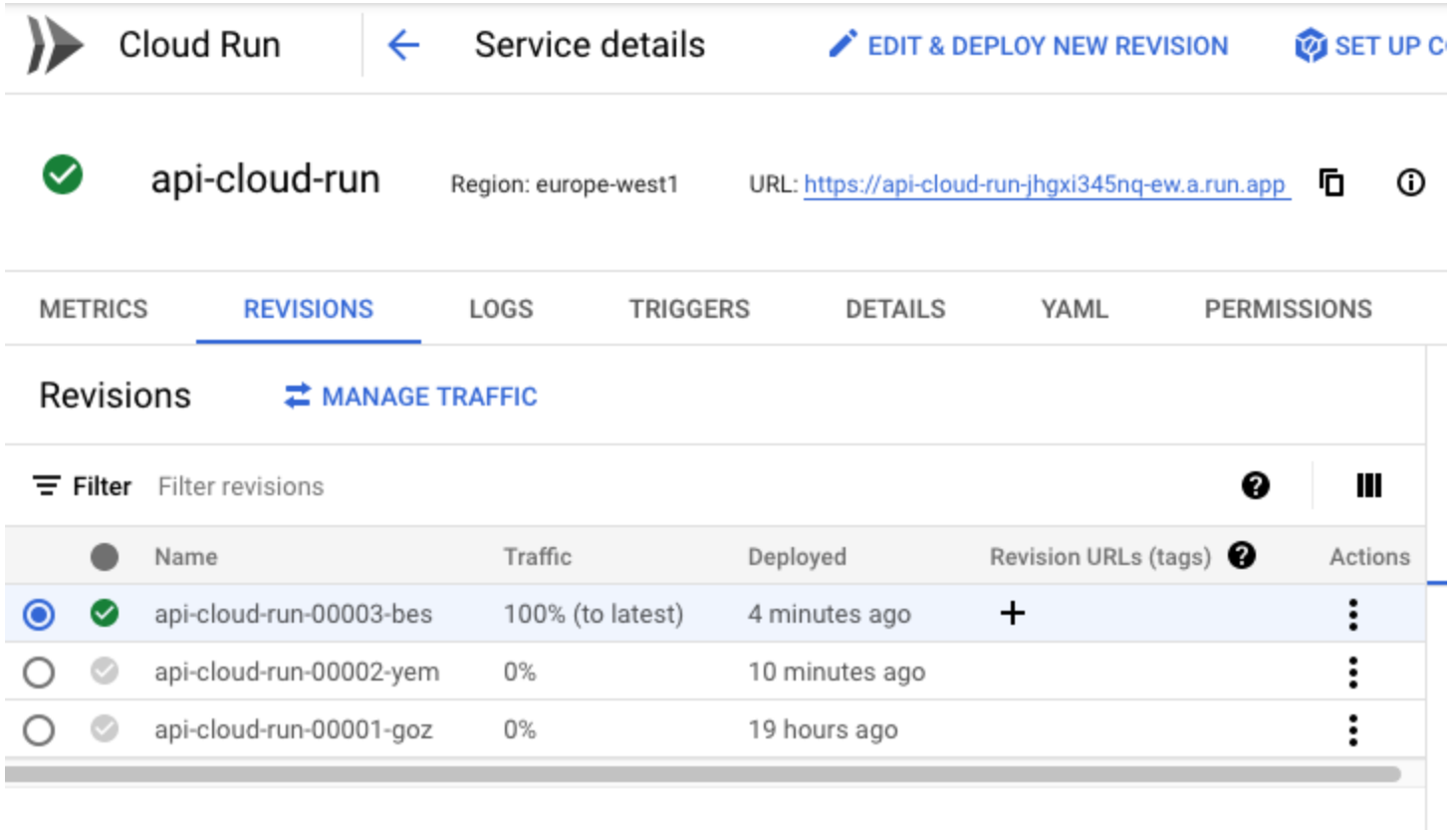



fernandodelacalle / my_first_api
Unwatch 1
Star 0
Fork 0
Code
Issues
Pull requests
Actions
Projects
Wiki
Security
Insights
Settings
pipeline Build and Deploy to Cloud Run #1
Re-run jobs
Summary
Jobs
Setup, Build, and Deploy to Cloud Run
Setup, Build, and Deploy to Cloud Run
succeeded now in 1m 20s
Search logs
Set up job 4s
Checkout 1s
Run GoogleCloudPlatform/github-actions/setup-gcloud@master 7s
Build 50s
Deploy 17s
Post Checkout 1s
Complete job 0s

- Si entramos en el registy veremos que hay una nueva imagen.



- Tendremos una nueva revisión en nuestro servico de cloud run.



Ejercicio

- Configura el despliege automático para tu dash.