EDEM
Escuela de Empresarios

EDEM
Escuela de Empresarios

DevSecOps

Ignacio Asín Calamonte

Curso 22-23 – 4ª Edición

Fecha 28/01/2023

ABOUT ME

- Working in **SIEMENS** as a **Cloud DevSecOps in the Global Cybersecurity Cloud Protection unit**
- Located in the SIEMENS **Madrid Cybersecurity Hub**

- Previously I was:
  - Cloud Architect & Blockchain Specialist
  - Big Data Engineer
  - DevOps Engineer
  - Senior Java Programmer

- **Computer Science**
- **Master in Big Data & Visual Analytics**
- **Cloud Certified** in Several Clouds & other Certs.

@iasinDev

ignacioasincalamonte

iasin.dev@gmail.com

- Favorite topics: **Automation, Google Cloud, Blockchain, latest Tech trends, …**

- Project Architect/Developer, Research, Formation courses, Meetups, …

AGENDA

EDEM
Escuela de Empresarios

1. FORMAT OF THE CLASS
2. LOCAL ENVIRONMENT

3. WHAT IS DEVOPS & DEVSECOPS
    1. Everything as Code
    2. Communication & Collaboration
    3. Security Management
    4. Continuous Integration (CI) & Continuous Deployment (CD)
    5. Platform as Code
    6. Infrastructure as Code
    7. Testing
    8. Destroy All

09,00h – 14,00h

BREAKS
10,30h – 10,45h
12,15h – 12,30h

Questions??

AGENDA

EDEM
Escuela de Empresarios

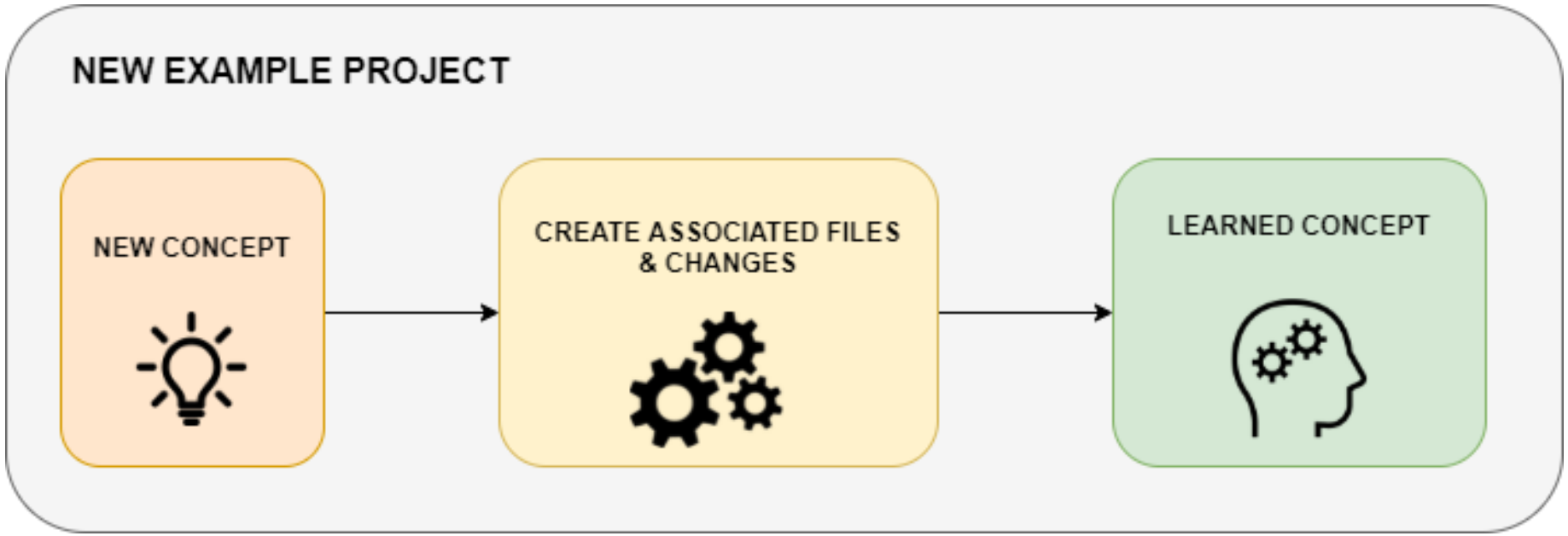
1. FORMAT OF THE CLASS

FORMAT OF THE CLASS

“Learning by doing”

FORMAT OF THE CLASS

“Learning by doing”



AGENDA

EDEM
Escuela de Empresarios

2. LOCAL ENVIRONMENT

LOCAL ENVIRONMENT

- Visual Studio Code
- GitHub Account
- Google Cloud Platform (GCP) account
- Google Chrome / similar browser
- GitHub Desktop (Recommended) Git for Windows (PRO users)
- Notepad++

AGENDA

EDEM
Escuela de Empresarios

3. WHAT IS DEVOPS & DEVSECOPS?

What **IS NOT** DevOps?

New
Superhero

Profession
(not only)

Culture
(not only)

What IS DevOps?

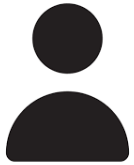
Velocity

Faster releases

Reliability

Scale

Improved collaboration



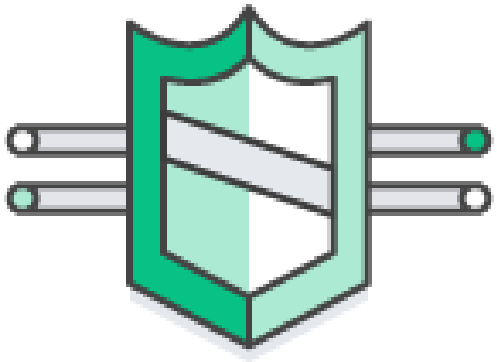
Scripting + Communicative + Process reengineering +
Experience with certain tools / languages + …

What IS DevOps?

DIFFICULT TO FIND !!

What **IS** DevOps?

What **IS** Dev**Sec**Ops?



Security

Everything as Code

Infrastructure as Code (IaC)

(Application as Code)

Configuration as Code

Security as Code

Policy as Code

… as Code

WHAT IS DEVOPS & DEVSECOPS?

Everything as Code 

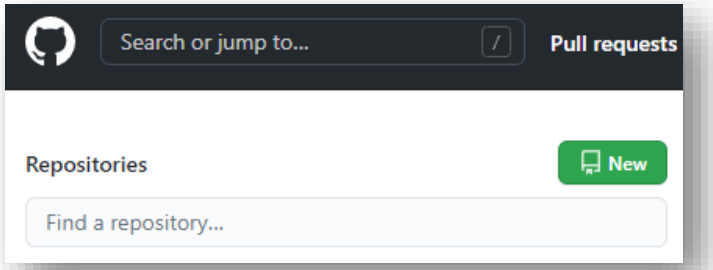
Solution here

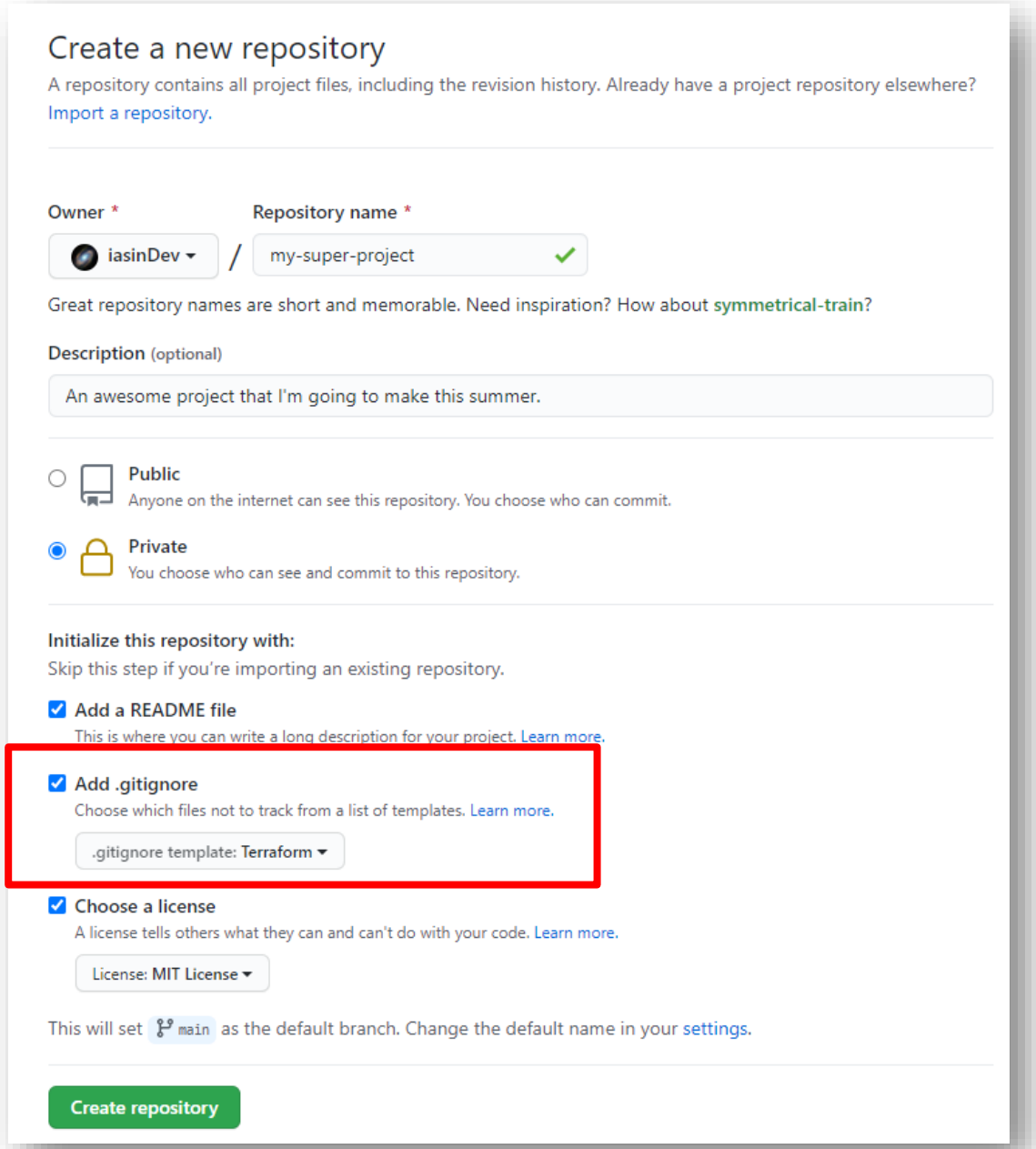
1. Create a new Project in GitHub (Private project)

How to … in next slide …

WHAT IS DEVOPS & DEVSECOPS?

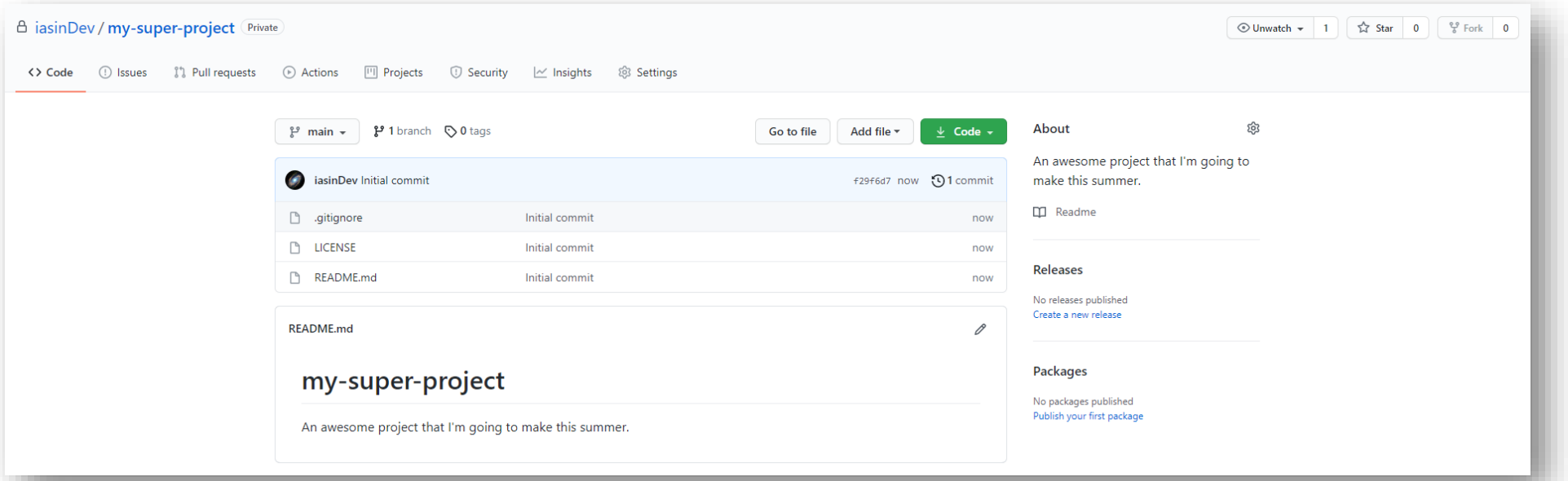
Everything as Code





WHAT IS DEVOPS & DEVSECOPS?

Everything as Code



WHAT IS DEVOPS & DEVSECOPS?

Everything as Code 

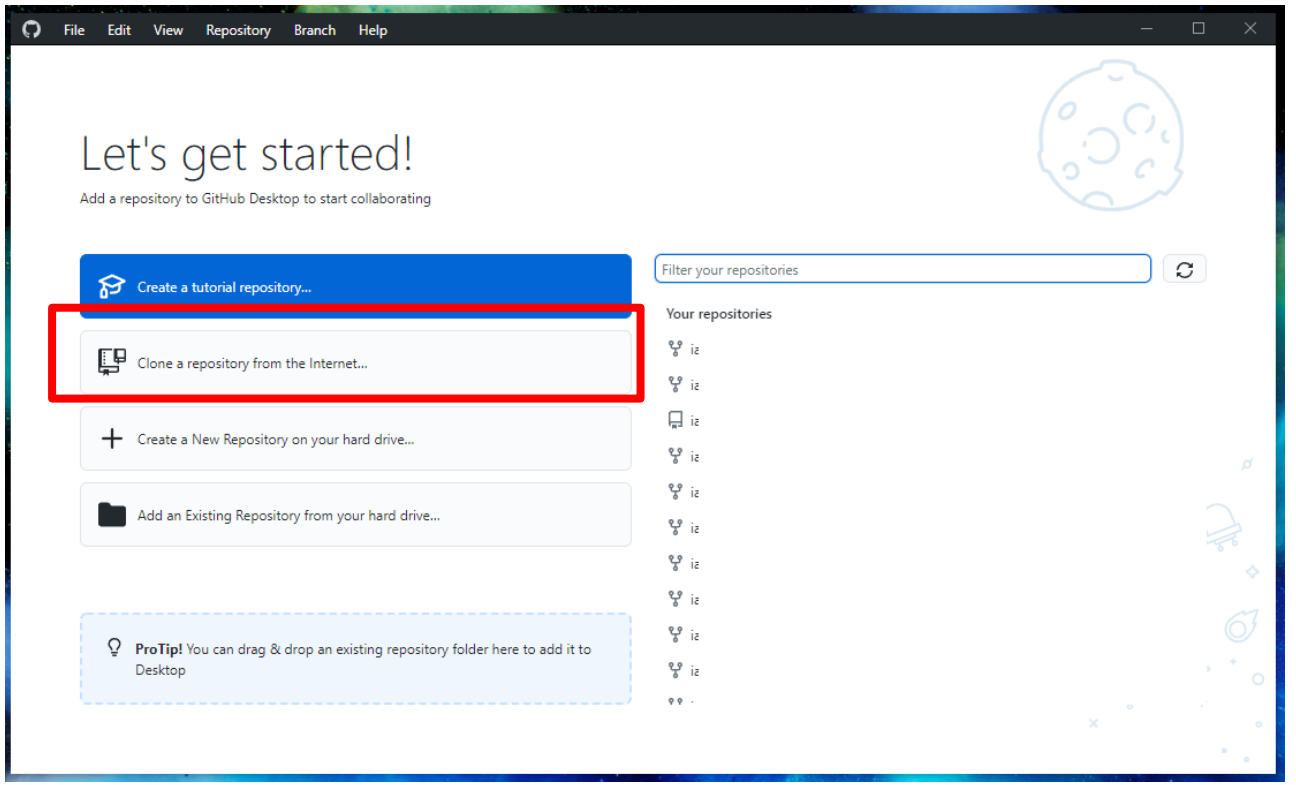
Solution here

2. Clone the new Project in local

How to … in next slide …

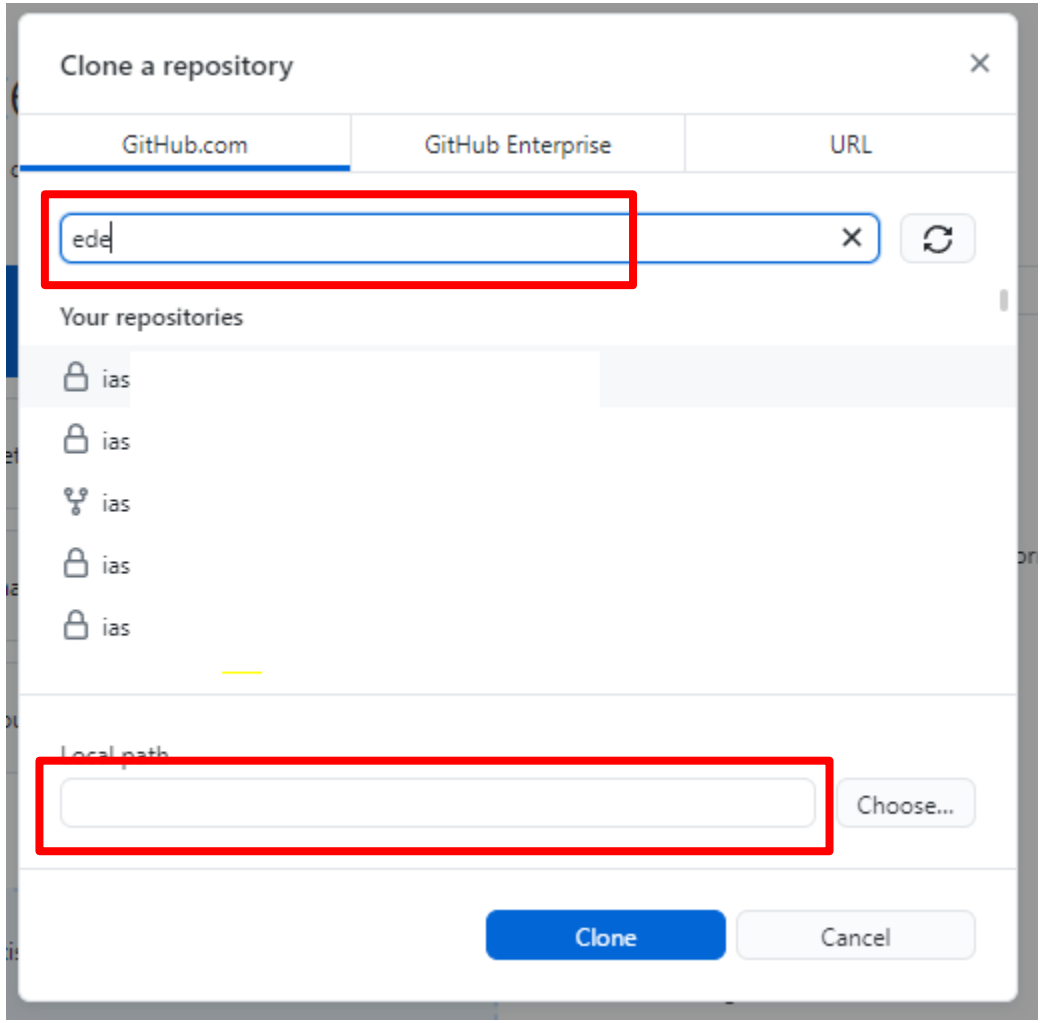
WHAT IS DEVOPS & DEVSECOPS?

Everything as Code



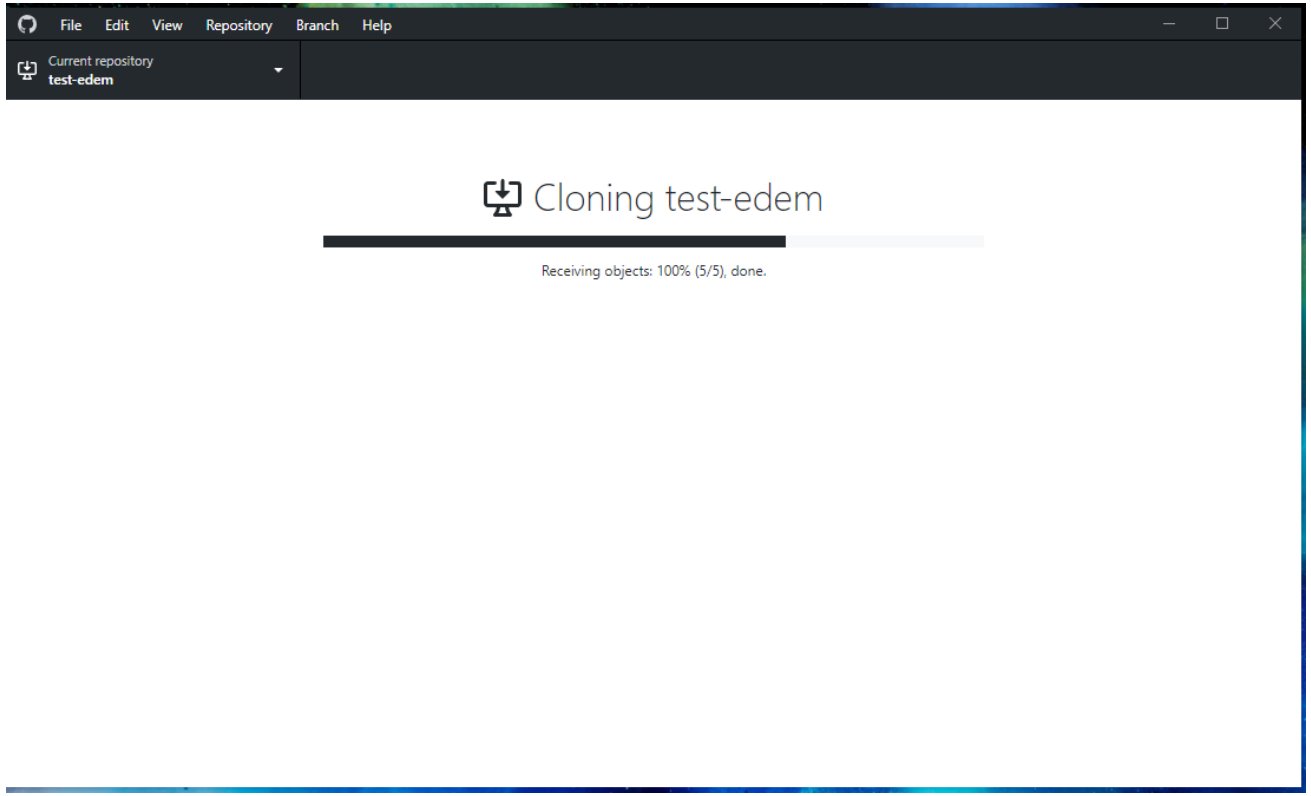
WHAT IS DEVOPS & DEVSECOPS?

Everything as Code



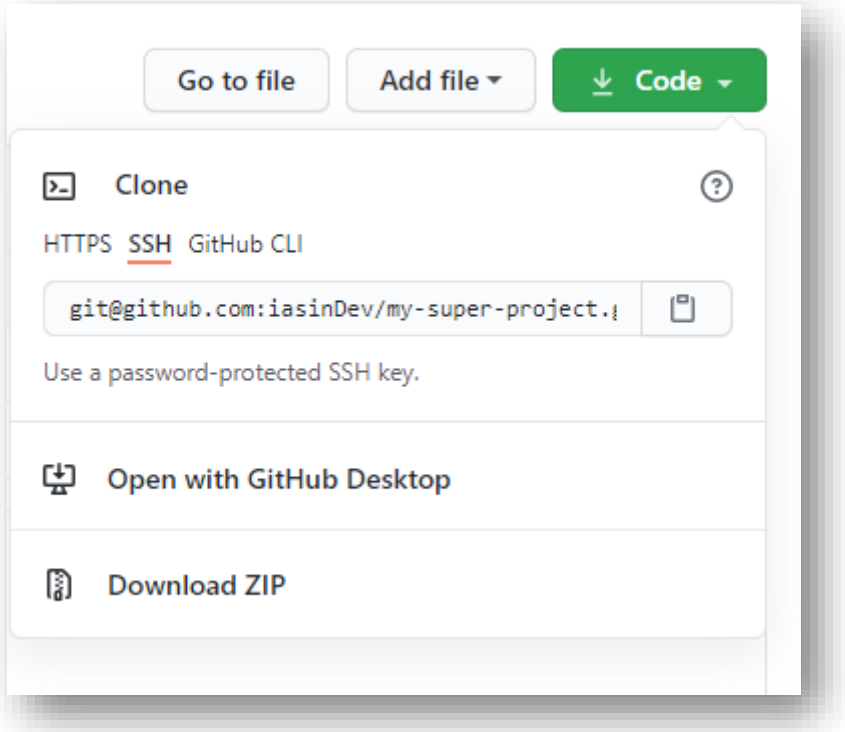
WHAT IS DEVOPS & DEVSECOPS?

Everything as Code



WHAT IS DEVOPS & DEVSECOPS?

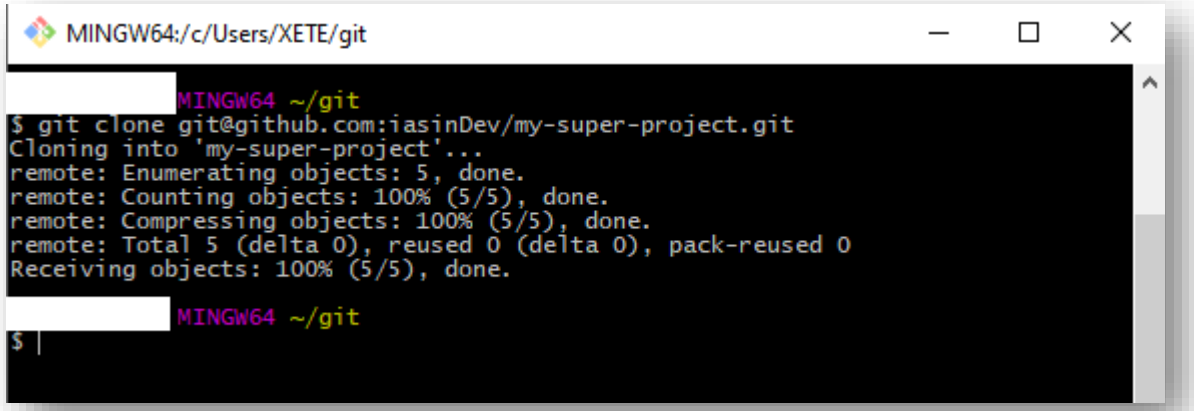
Everything as Code



Pro Mode

Using ssh keys in console

See here and here for more info



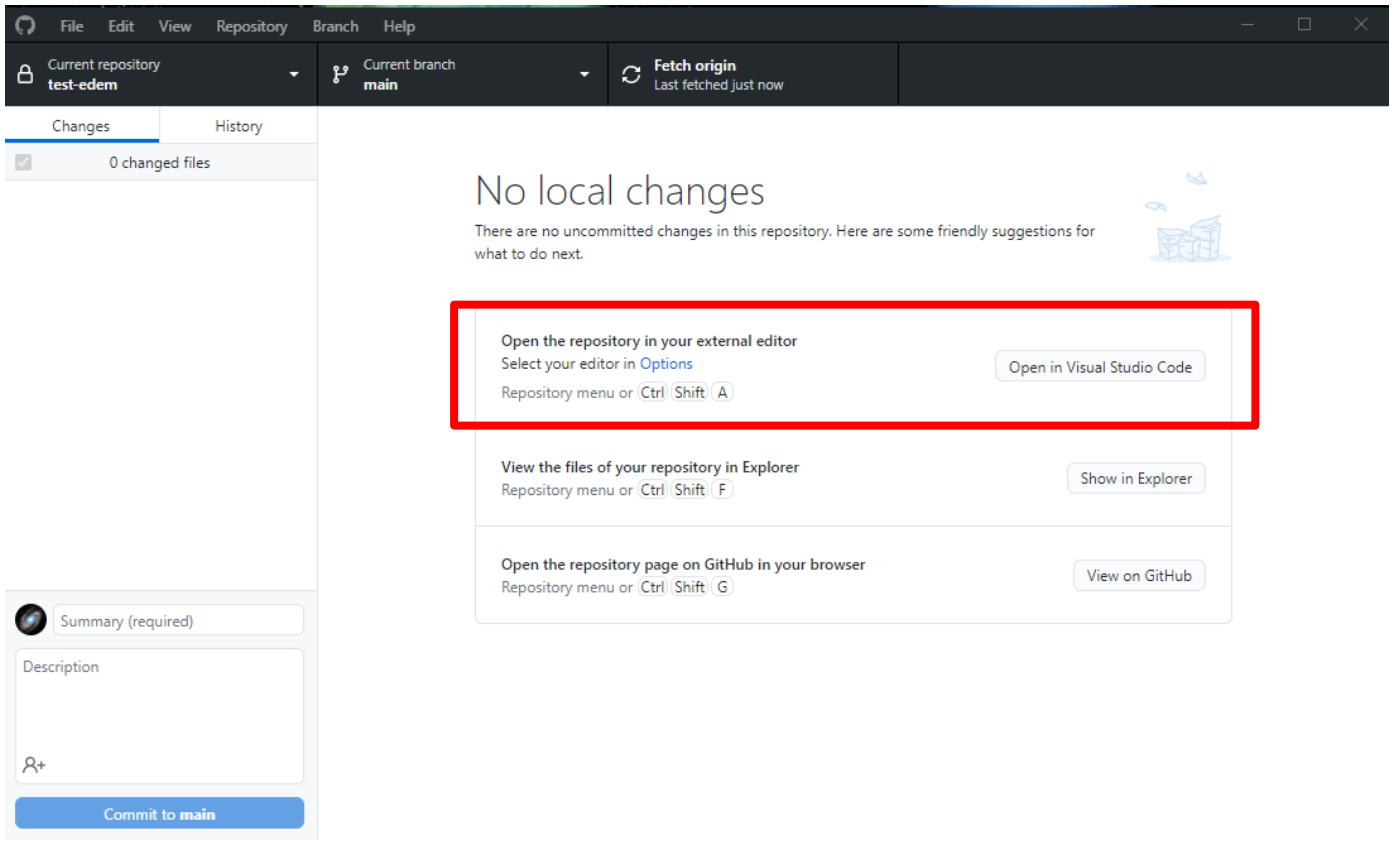
Everything as Code 

Solution here

3. Open the new Project with Visual Studio Code

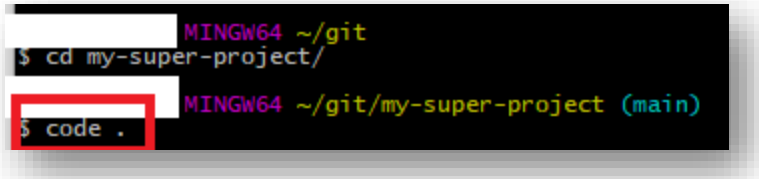
WHAT IS DEVOPS & DEVSECOPS?

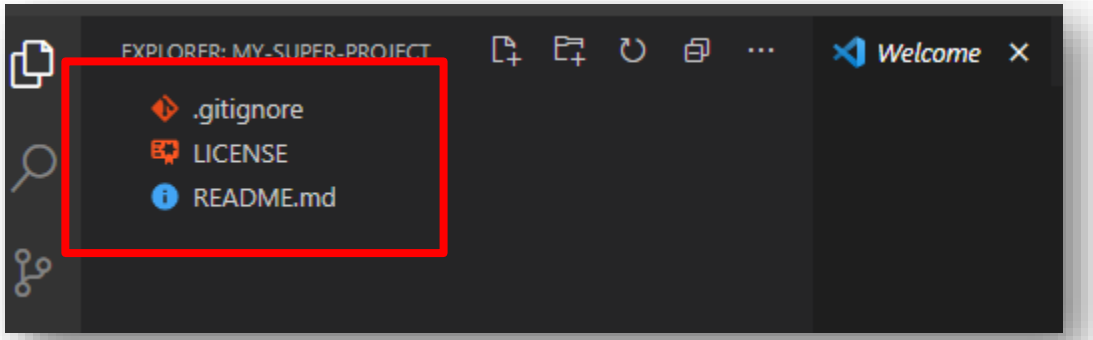
Everything as Code



WHAT IS DEVOPS & DEVSECOPS?

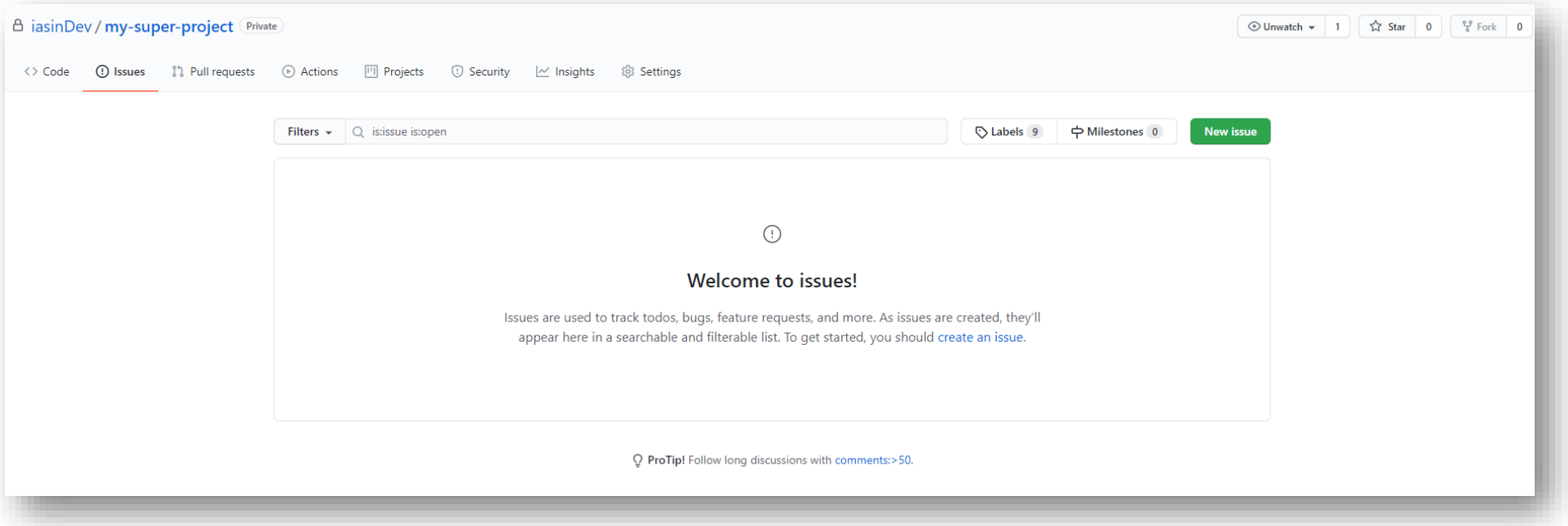
Everything as Code





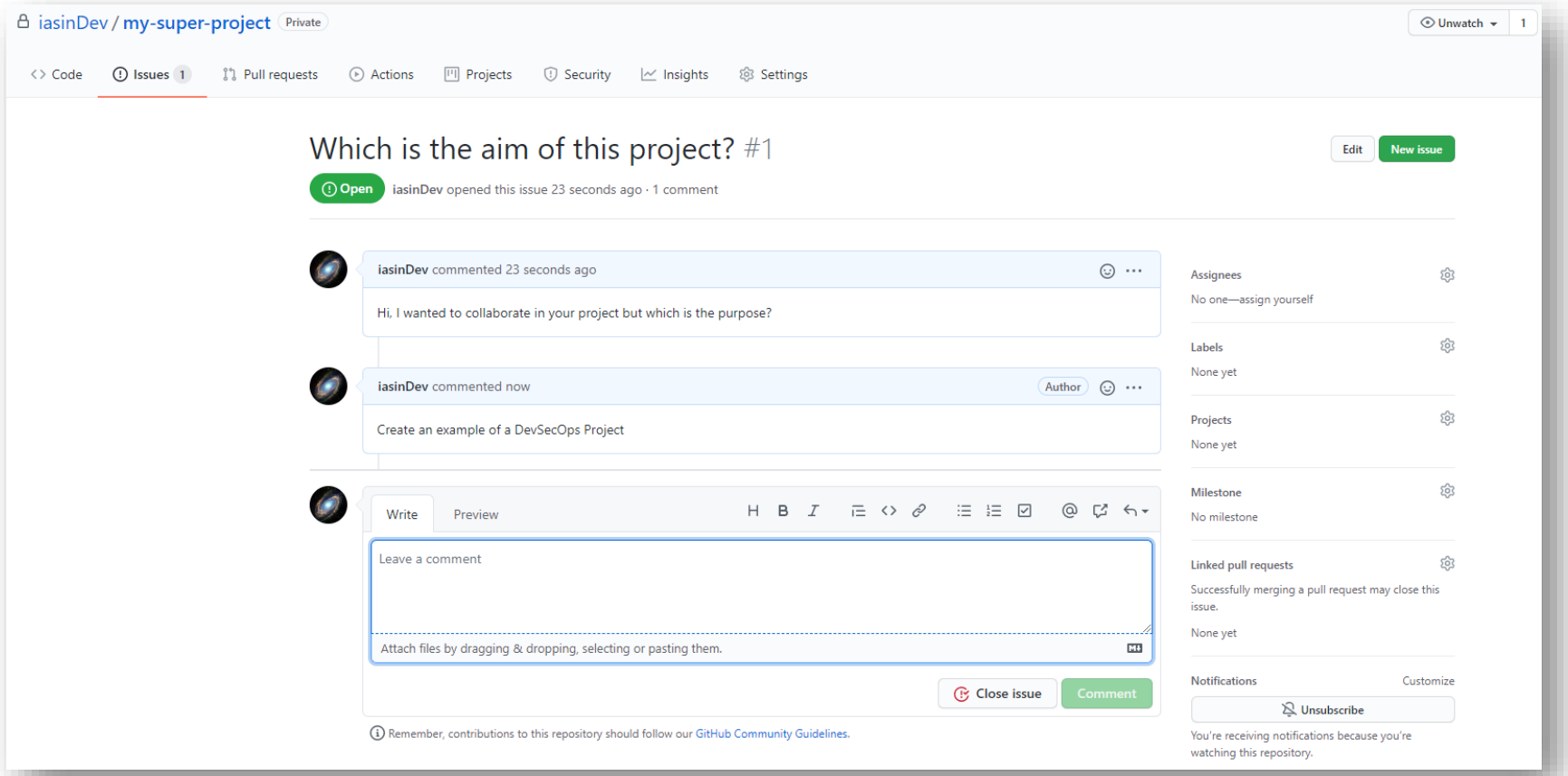
WHAT IS DEVOPS & DEVSECOPS?

Communication & Collaboration



WHAT IS DEVOPS & DEVSECOPS?

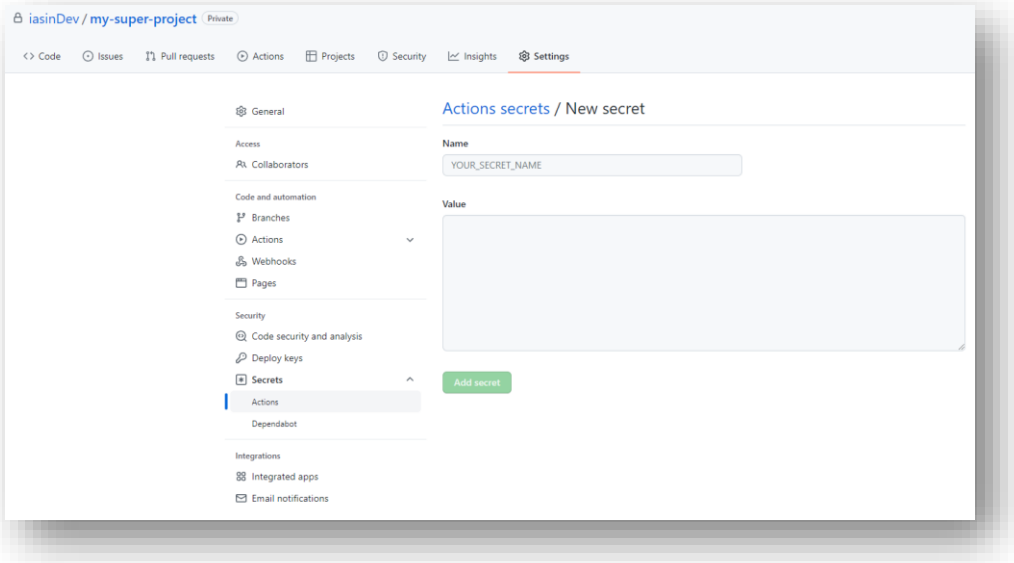
Communication & Collaboration



Security Management

We need to obtain GCP credentials to interact with the Cloud; this is called, a “service account”.

C a roach w be
d ad file w he
d tials, a it in he
P and even e in
th b Project.



GitHub Secrets

Security Management

Create a new Google Cloud Project called edemdevsecops

Create a Google Cloud service account

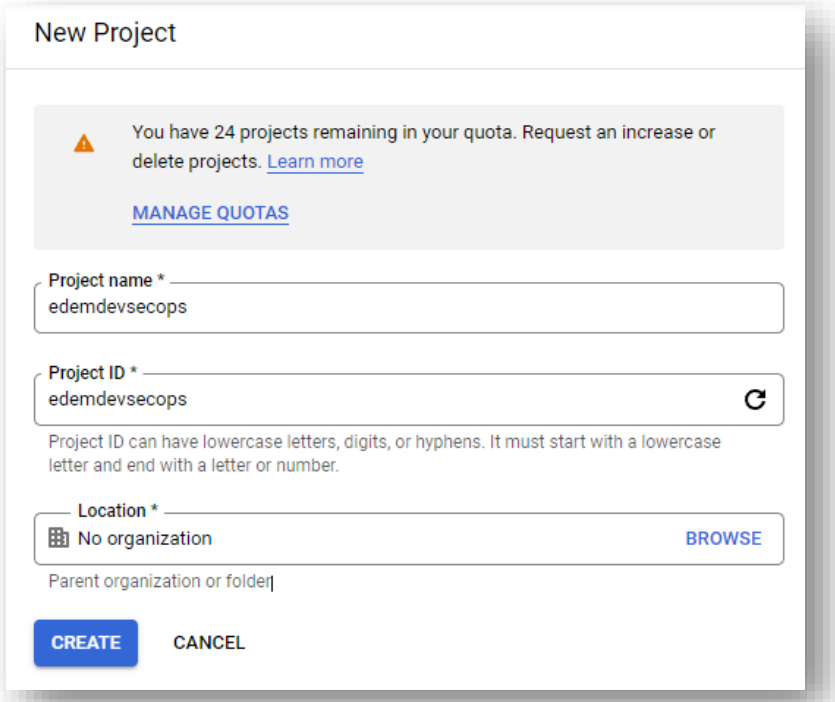
Call it “edemdevsecops” with the description “Account used for the course of DevSecOps of EDEM”

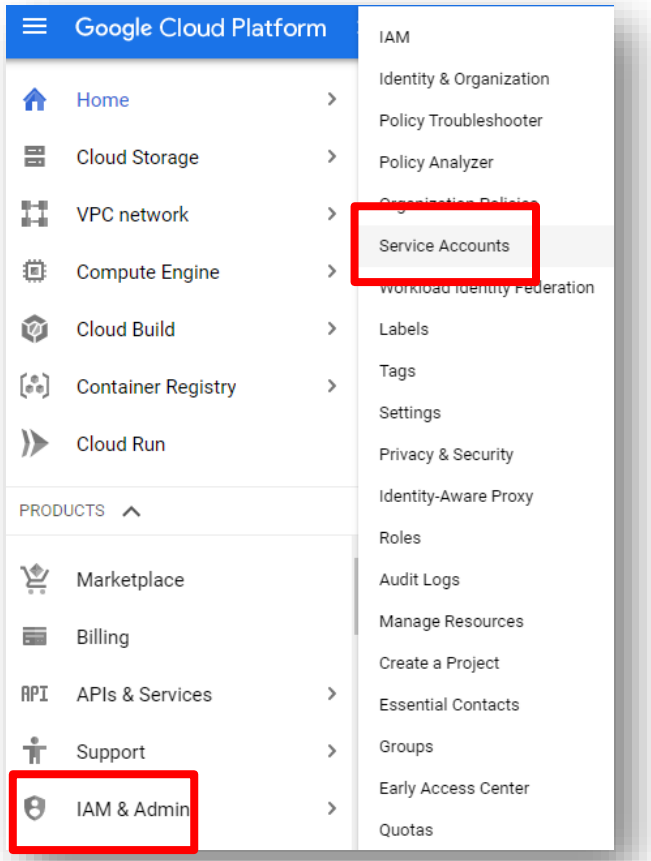
With the following roles (PoLP, Principle of Less Privilege):

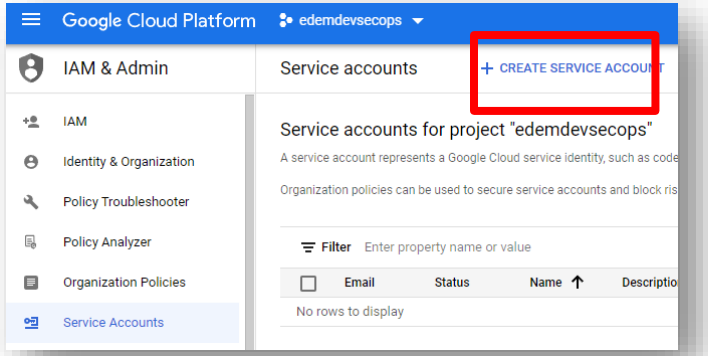
- `Cloud Run Admin` - allows for the creation of new Cloud Run services
- `Service Account User` - required to deploy to Cloud Run as service account
- `Storage Admin` - allow push to Google Container Registry

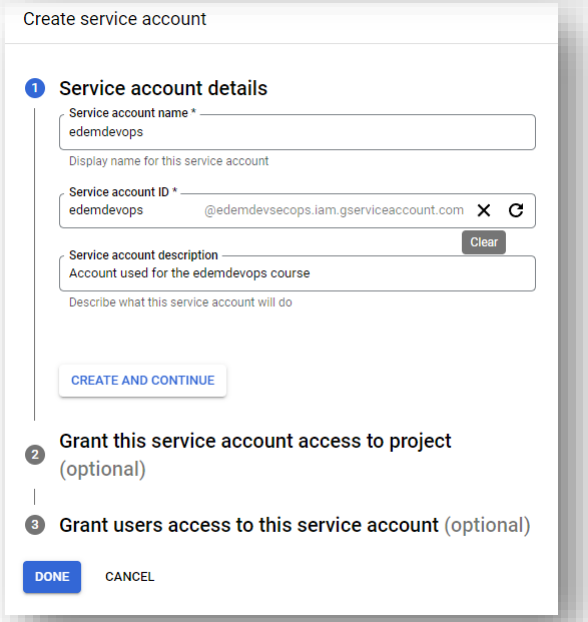
WHAT IS DEVOPS & DEVSECOPS?

Security Management



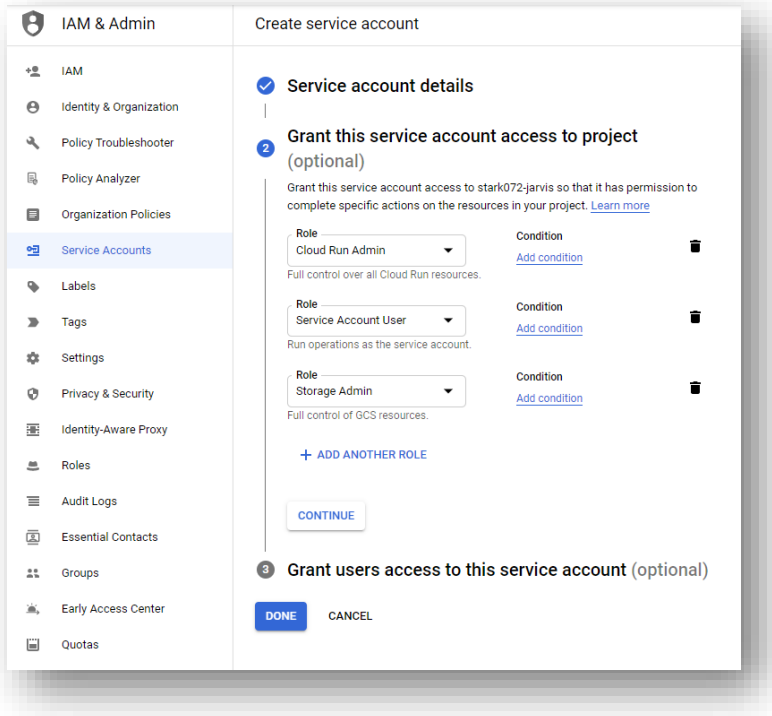


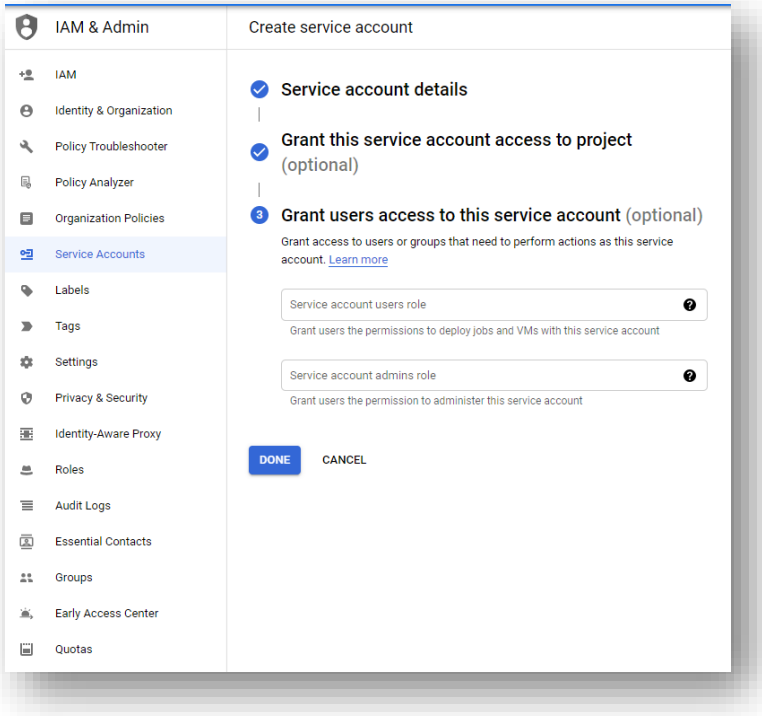


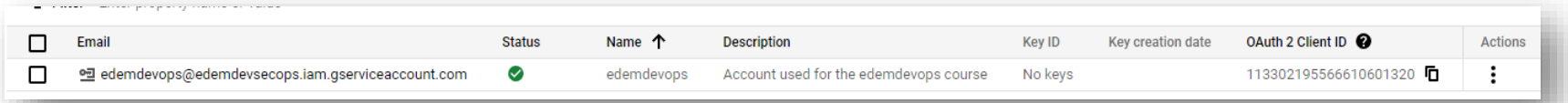


WHAT IS DEVOPS & DEVSECOPS?

Security Management

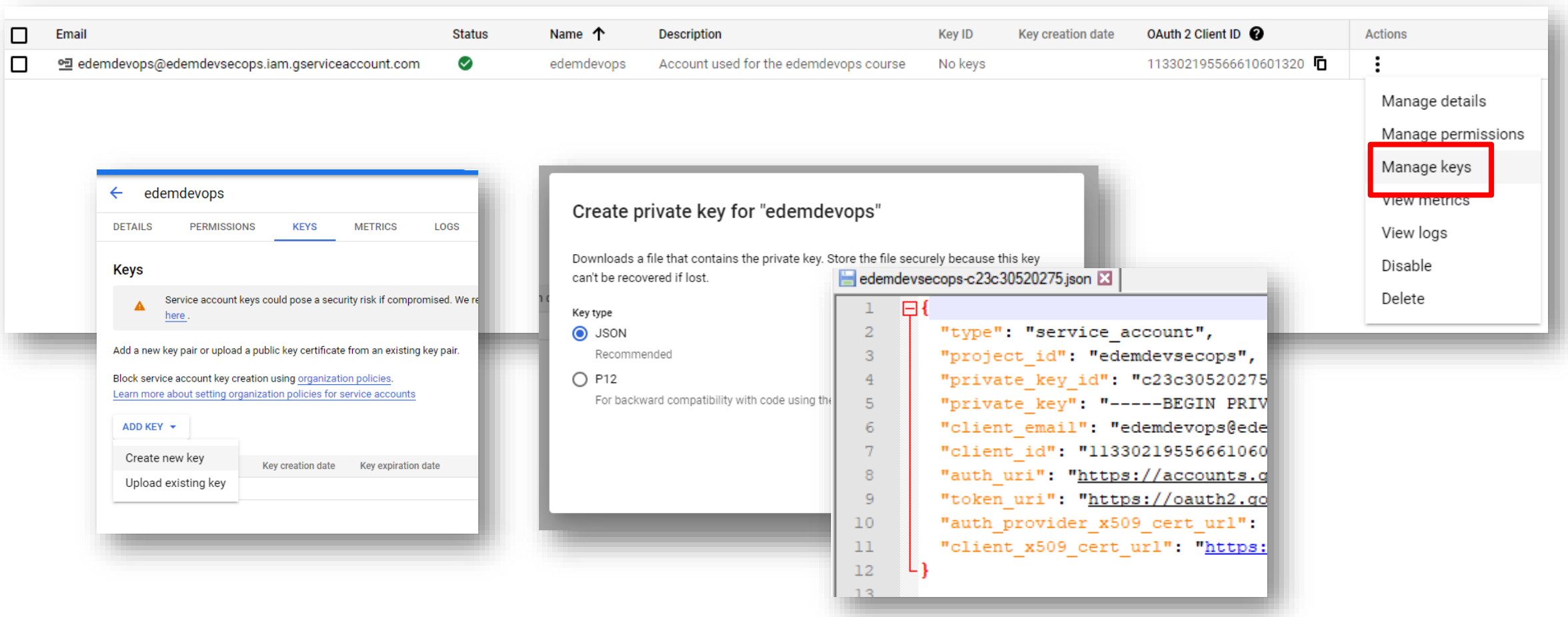






WHAT IS DEVOPS & DEVSECOPS?

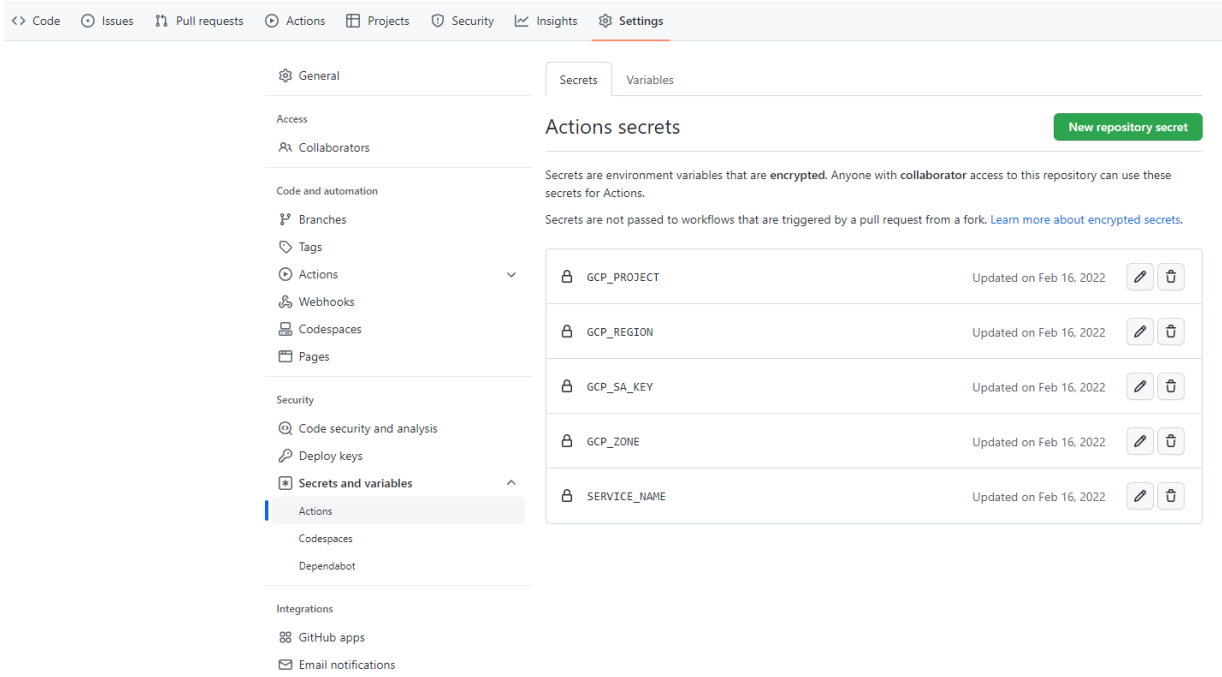
Security Management



Security Management

Add the following secrets to your GitHub project secrets:

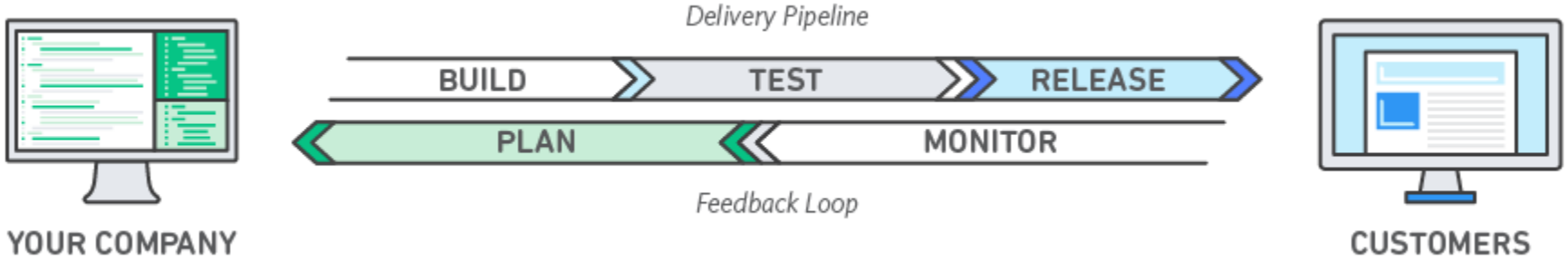
- `GCP_PROJECT` - field project_id of the json file
- `GCP_REGION` - europe-west1
- `GCP_SA_KEY` - copy&paste whole json file
- `GCP_ZONE` - europe-west1-b
- `SERVICE_NAME` - edemdevsecops



After copy-pasting the secrets delete the downloaded json file

WHAT IS DEVOPS & DEVSECOPS?

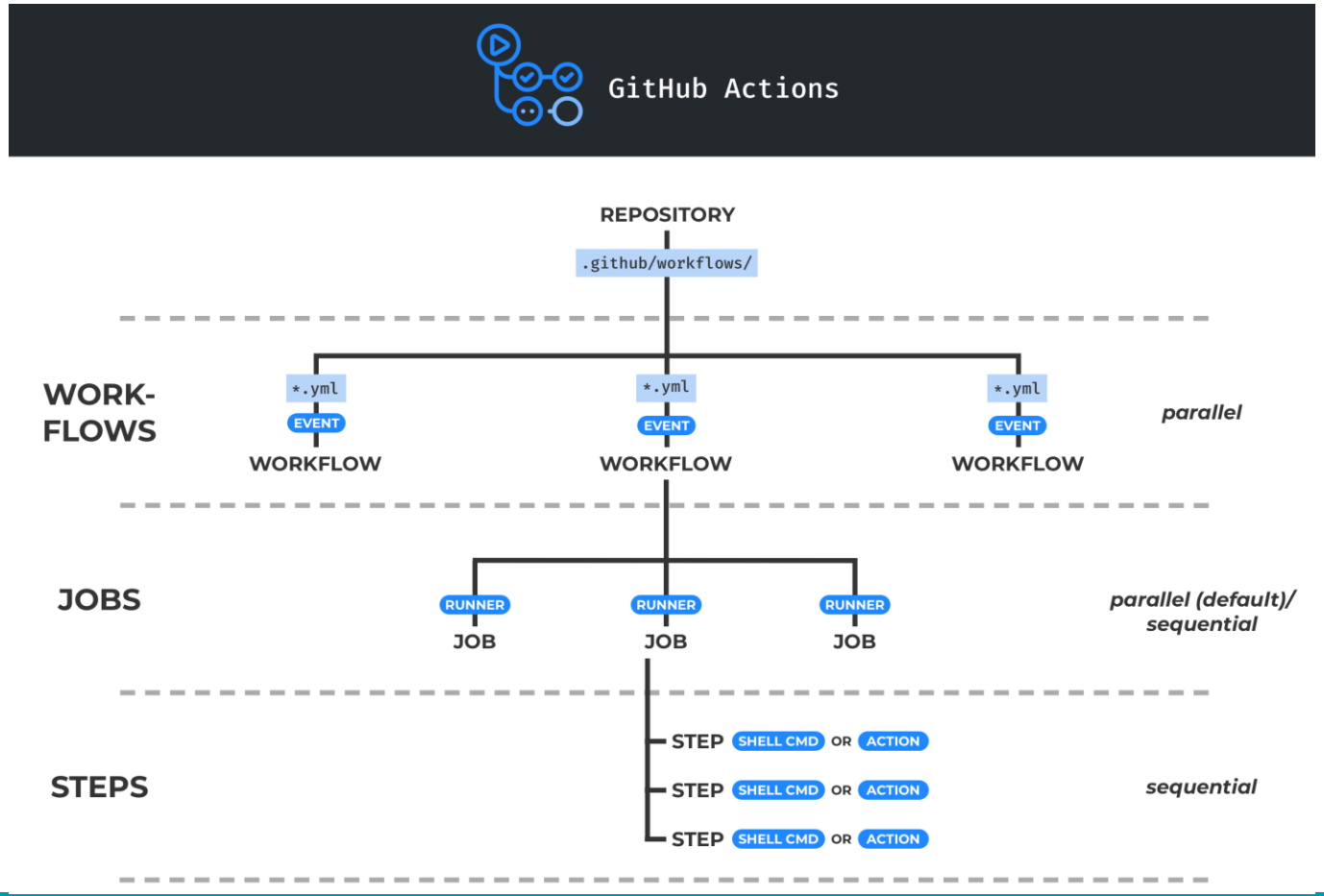
Continuous Integration (CI) & Continuous Delivery (CD)



GitHub Actions

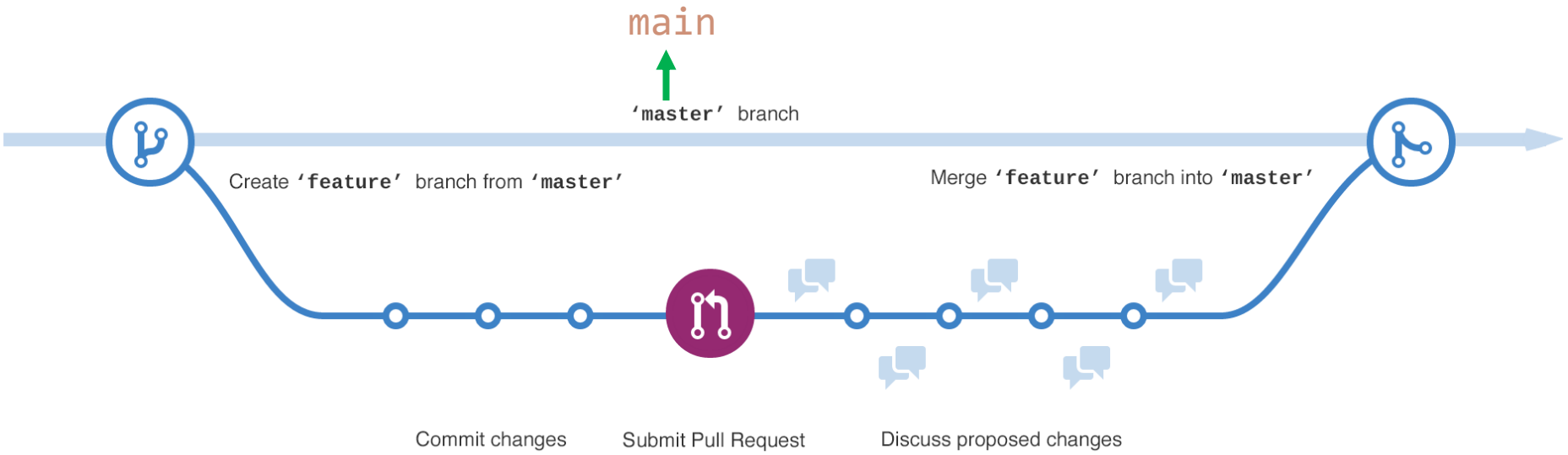
WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)



Continuous Integration (CI) & Continuous Delivery (CD)

GitHub Flow



Continuous Integration (CI) & Continuous Delivery (CD)

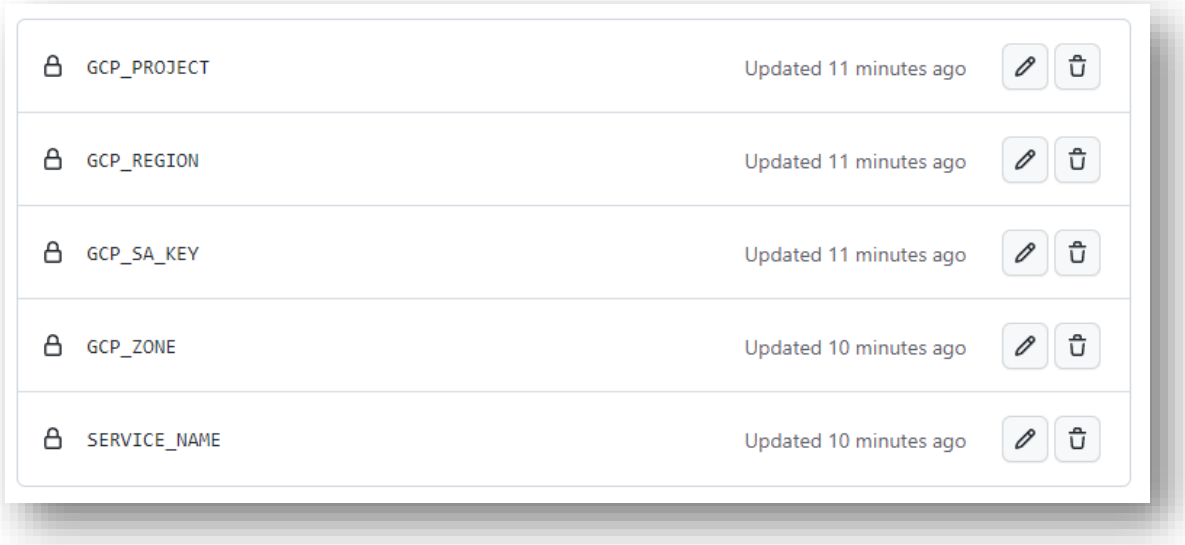
Git Flow

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```
# Copyright 2021 Ignacio Asin (iasinDev)
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

on:
  push:
    branches:
    - main
  pull_request:
  workflow_dispatch:
```

Continuous Integration (CI) & Continuous Delivery (CD)



Continuous Integration (CI) & Continuous Delivery (CD)

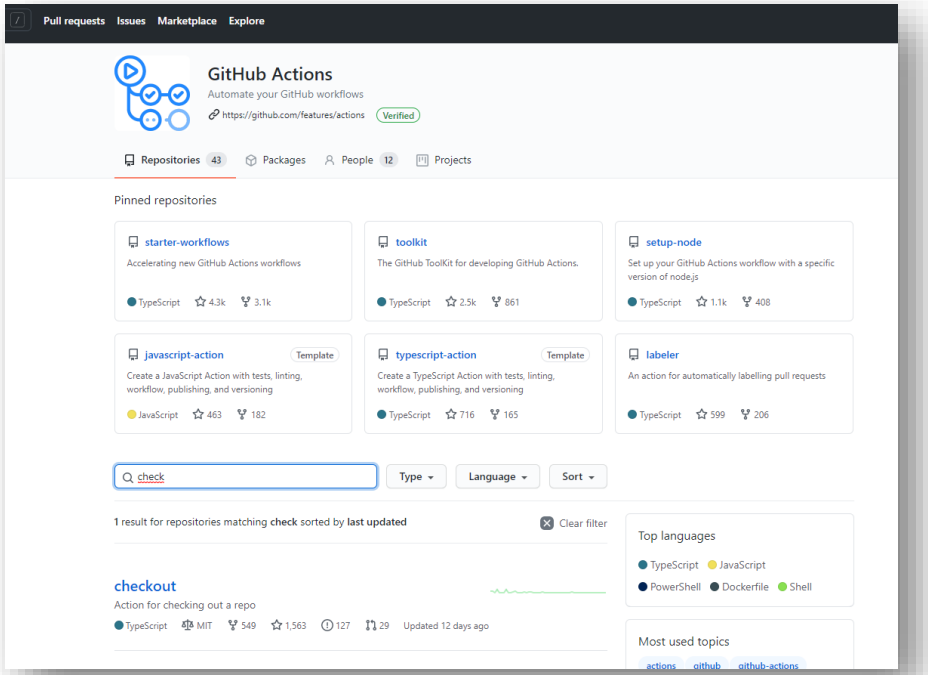
.github/workflows/pipeline.yml

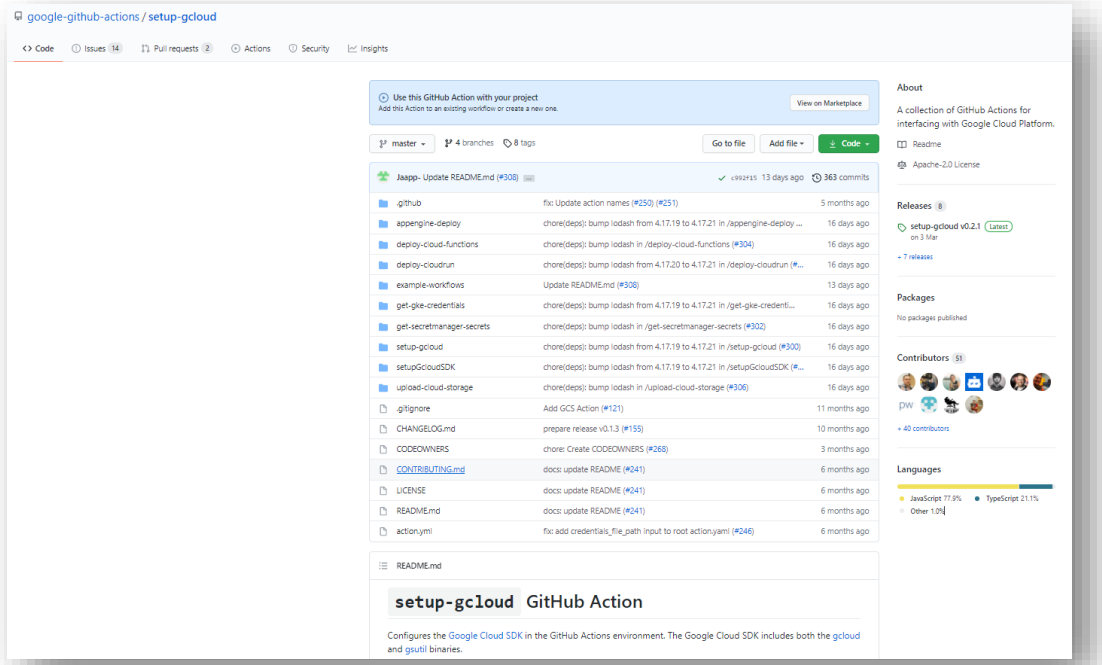
```yaml
name: Build, Check and Deploy to Cloud Run on GCP
env:
  PROJECT_ID: ${{ secrets.GCP_PROJECT }}
  SERVICE: ${{ secrets.SERVICE_NAME }}
  REGION: ${{ secrets.GCP_REGION }}
  ZONE: ${{ secrets.GCP_ZONE }}

jobs:
```

WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)

Examples here, here and here

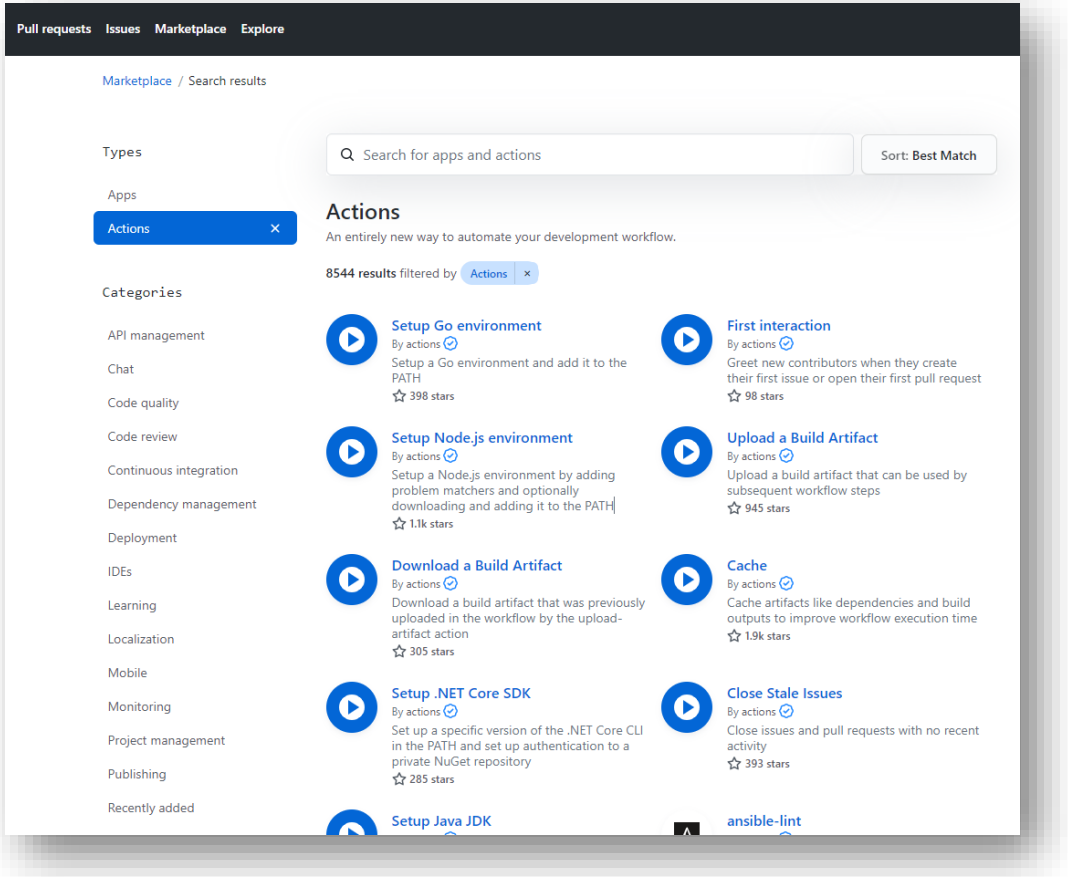




WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)

https://github.com/marketplace?type=actions



Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
build-and-push-container-image:

  runs-on: ubuntu-latest

  steps:

  - name: Enable GitHub Actions
    uses: actions/checkout@v3.3.0

  - id: auth
    uses: google-github-actions/auth@v1
    with:
      credentials_json: ${{ secrets.GCP_SA_KEY }}

  - name: Set up Cloud SDK
    uses: google-github-actions/setup-gcloud@v1
    with:
      project_id: ${{ env.PROJECT_ID }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

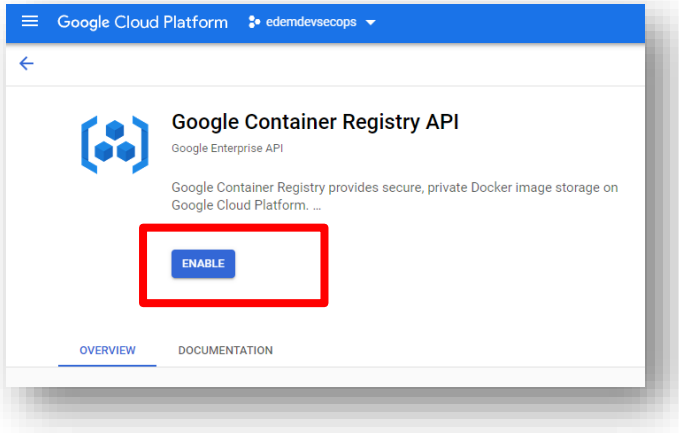
.github/workflows/pipeline.yml

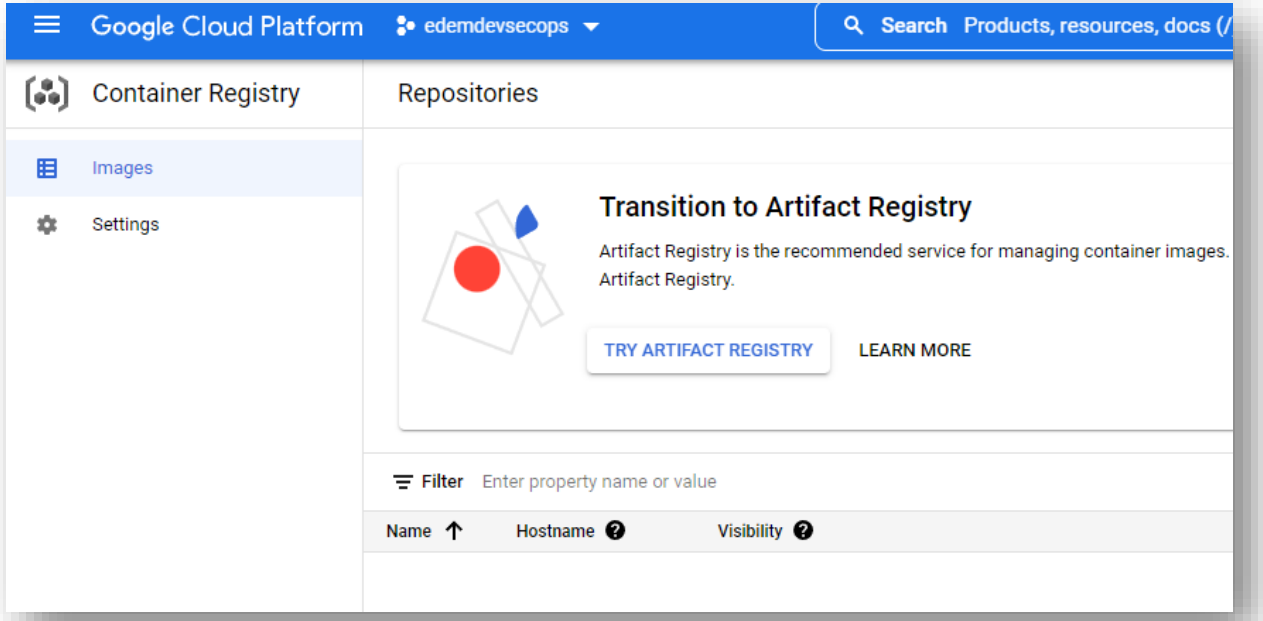
```
- name: Check GCP account details
  run: gcloud config list

- name: Authorize Docker push
  run: gcloud auth configure-docker
```

WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)

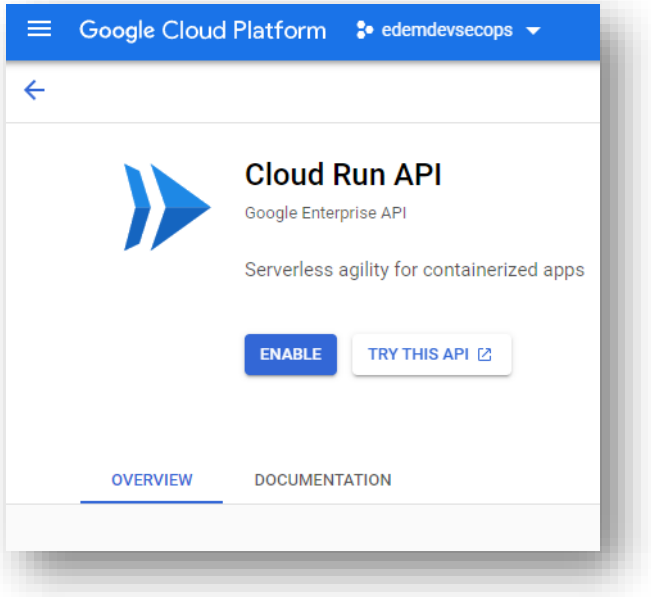
Enable the Container Registry API
https://console.cloud.google.com/marketplace/product/google/containerregistry.googleapis.com





Continuous Integration (CI) & Continuous Delivery (CD)

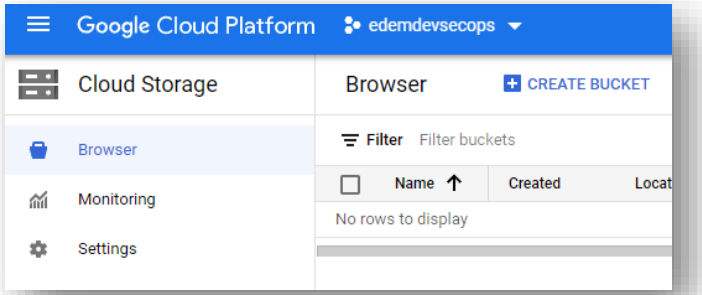
Enable also the Cloud Run API
https://console.cloud.google.com/apis/library/run.googleapis.com

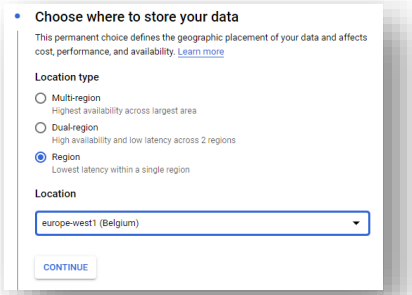


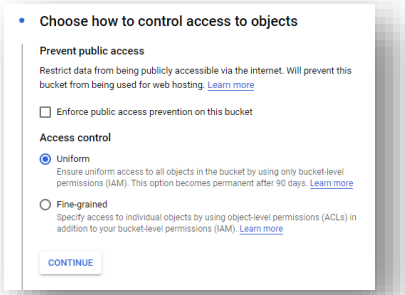
WHAT IS DEVOPS & DEVSECOPS?

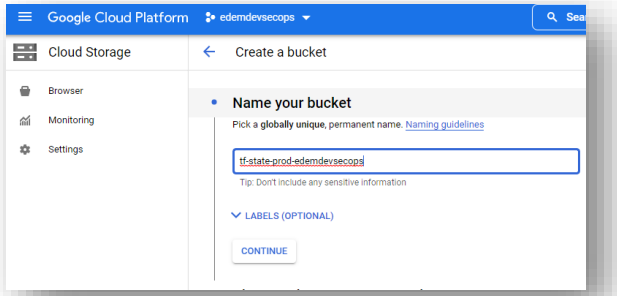
Continuous Integration (CI) & Continuous Delivery (CD)

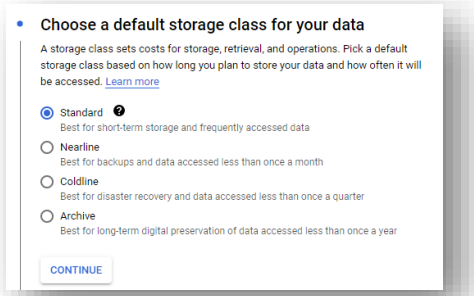
In Google Cloud create a bucket called "tf-state-prod-edemdevsecops" or a free name

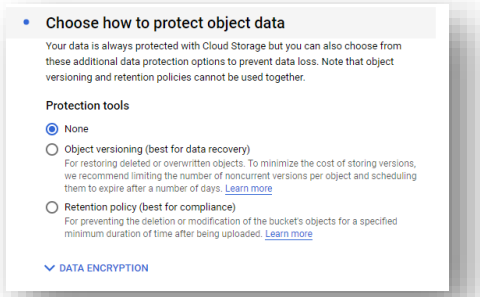






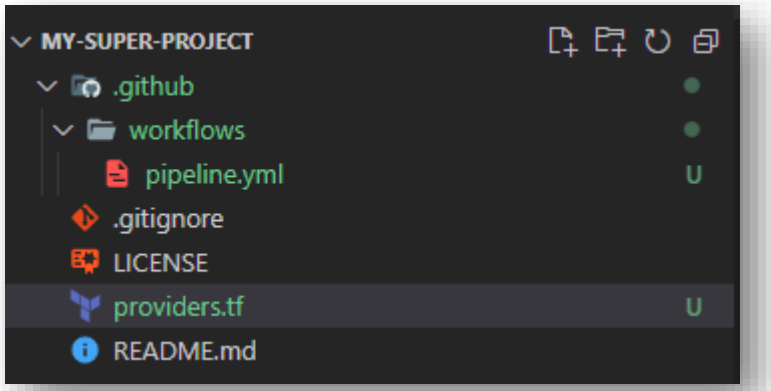






Continuous Integration (CI) & Continuous Delivery (CD)

In Google Cloud create a bucket called "tf-state-prod-edemdevsecop" or a free name




```
provider "google" {
  project = var.project_id
  region  = var.region
  zone    = var.zone
}

terraform {
    backend "gcs" {
        bucket  = "tf-state-prod-edemdevsecops"
        prefix  = "terraform/state"
    }
}
```

Same name

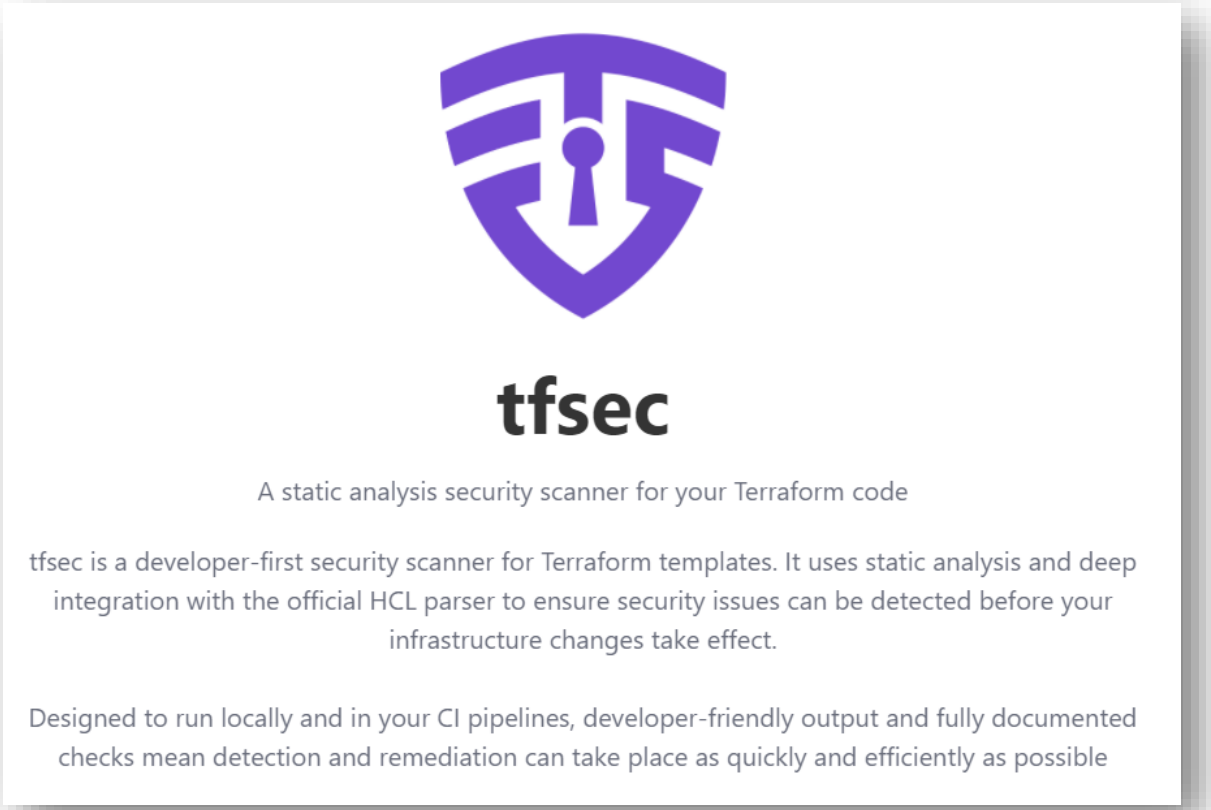
Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
- name: Build and Push Container
  run: |-
    docker build -t gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }}:${{  github.sha }} .
    docker push gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }}:${{  github.sha }}
    docker build -t gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }}:latest .
    docker push gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }}:latest
```

WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)



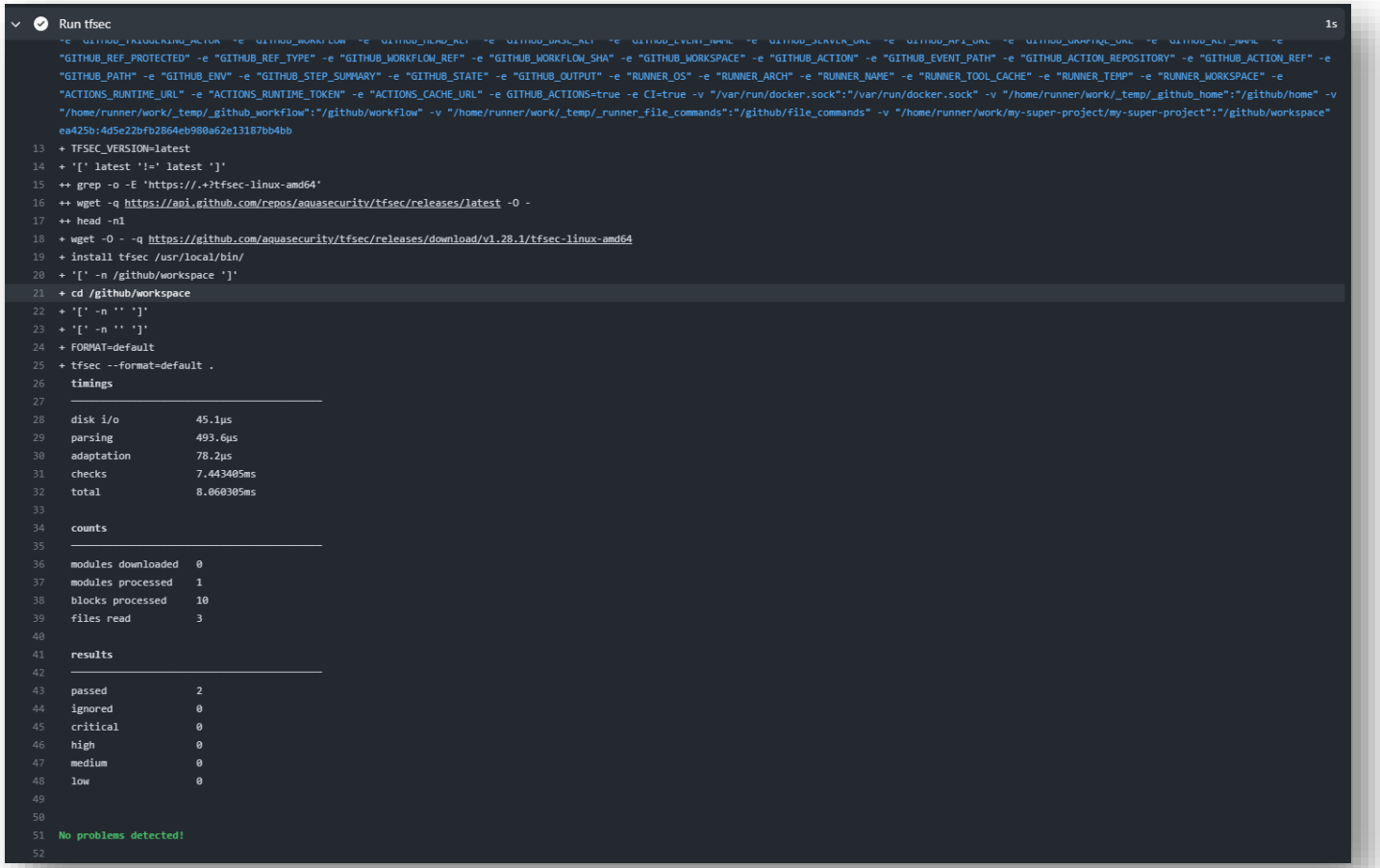
Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
check-terraform-security:

  needs: build-and-push-container-image

  runs-on: ubuntu-latest

  steps:

  - name: Clone repo
    uses: actions/checkout@v3.3.0

  - name: Run tfsec
    uses: aquasecurity/tfsec-action@v1.0.0
```

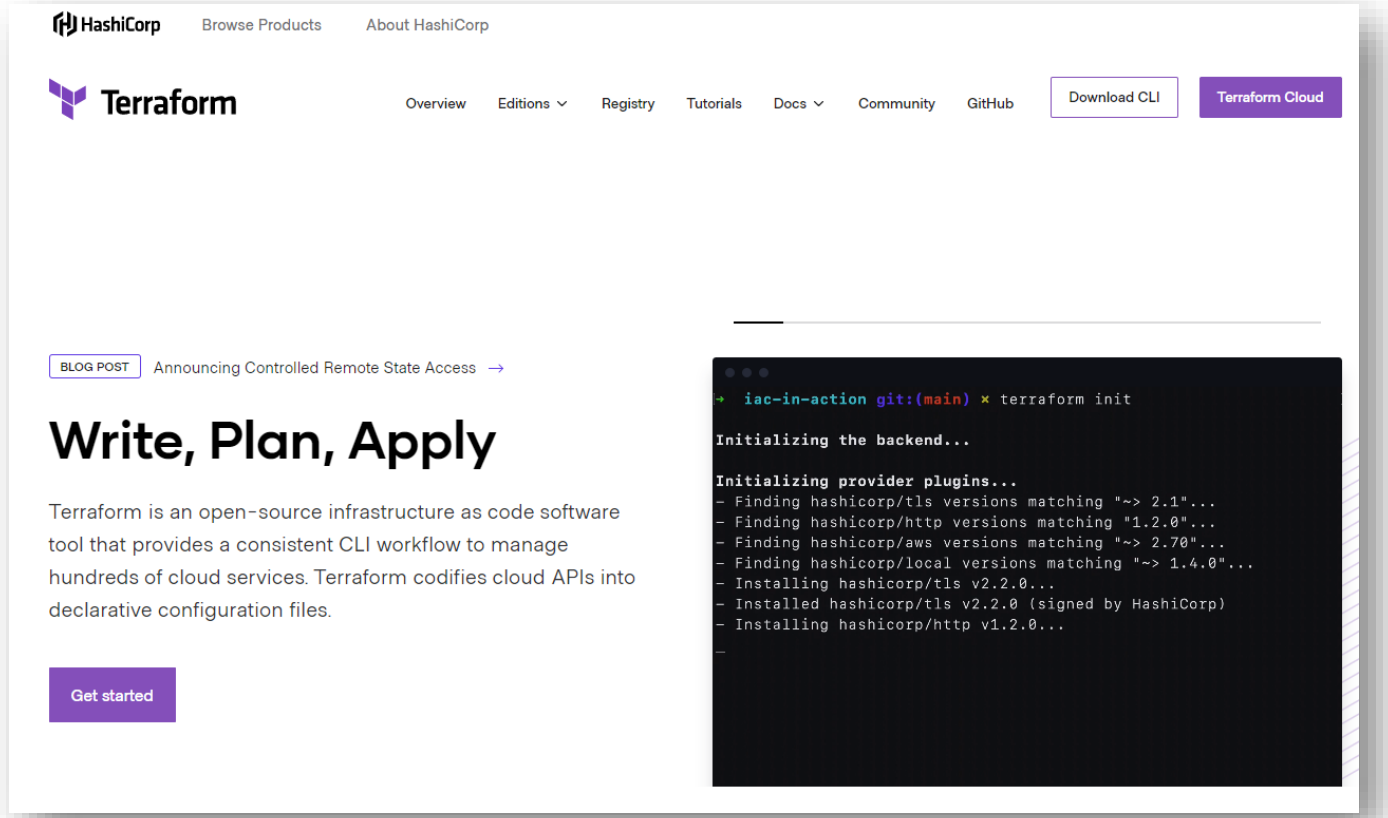
WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)



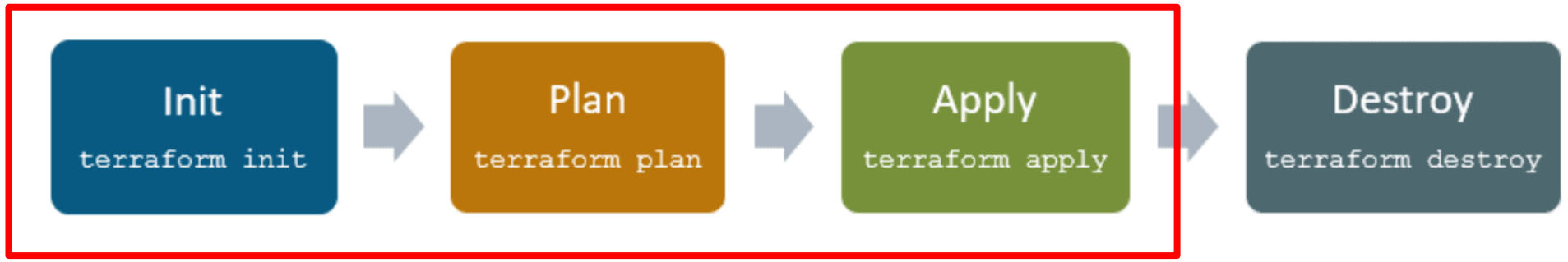
WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)



WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)



Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
deploy-in-cloud-run:

  needs: [build-and-push-container-image,check-terraform-security]

  runs-on: ubuntu-latest

  steps:

  - name: Enable GitHub Actions
    uses: actions/checkout@v3.3.0

  - id: auth
    uses: google-github-actions/auth@v1
    with:
      credentials_json: ${{ secrets.GCP_SA_KEY }}

  - name: Set up Cloud SDK
    uses: google-github-actions/setup-gcloud@v1
    with:
      project_id: ${{ env.PROJECT_ID }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

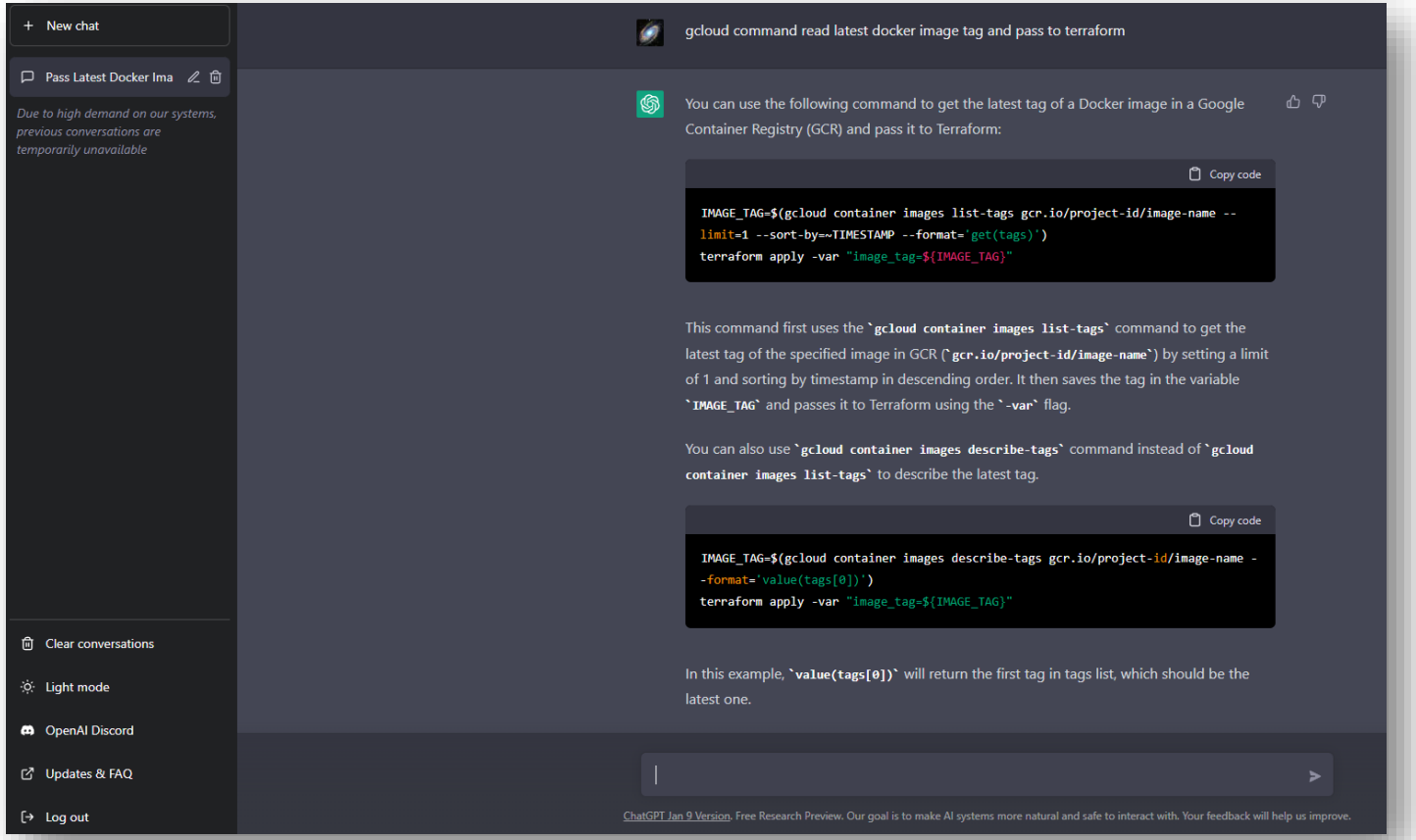
```
- name: Getting tag of the built image
  run: |-
    image_tag=$(gcloud container images list-tags gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }} --limit=1 --sort-by=~TIMESTAMP --format='get(tags)')
    IFS=';' read -ra image_tag_ARRAY <<< "$image_tag"
    image_tag=${image_tag_ARRAY[0]}
    echo "image_tag: $image_tag"
    echo "image_tag=$image_tag" >> $GITHUB_ENV

# Install the latest version of Terraform CLI and configure the Terraform CLI configuration file with a Terraform Cloud user API token
- name: Setup Terraform
  uses: hashicorp/setup-terraform@v2.0.3
```

WHAT IS DEVOPS & DEVSECOPS?

Continuous Integration (CI) & Continuous Delivery (CD)

`.github/workflows/pipeline.yml`

[ChatGPT-3]



Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
# Initialize a new or existing Terraform working directory by creating initial files, loading any remote state, downloading modules, etc.
- name: Terraform Init
  run: |
    echo "image_tag: ${{ env.image_tag }}"
    terraform init
  env:
    GOOGLE_CREDENTIALS: ${{ secrets.GCP_SA_KEY }}
    TF_VAR_project_id: ${{ env.PROJECT_ID }}
    TF_VAR_region: ${{ env.REGION }}
    TF_VAR_zone: ${{ secrets.GCP_ZONE }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
# Generates an execution plan for Terraform
- name: Terraform Plan
  run: terraform plan -lock=false
  env:
    GOOGLE_CREDENTIALS: ${{ secrets.GCP_SA_KEY }}
    TF_VAR_project_id: ${{ env.PROJECT_ID }}
    TF_VAR_region: ${{ env.REGION }}
    TF_VAR_zone: ${{ env.ZONE }}
    TF_VAR_service: ${{ env.SERVICE }}
    TF_VAR_image_tag: ${{ env.image_tag }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/pipeline.yml

```yaml
# Apply the execution plan
- name: Terraform Apply
  if: github.ref == 'refs/heads/main'
  run: terraform apply -lock=false -auto-approve
  env:
    GOOGLE_CREDENTIALS: ${{ secrets.GCP_SA_KEY }}
    TF_VAR_project_id: ${{ env.PROJECT_ID }}
    TF_VAR_region: ${{ env.REGION }}
    TF_VAR_zone: ${{ env.ZONE }}
    TF_VAR_service: ${{ env.SERVICE }}
    TF_VAR_image_tag: ${{ env.image_tag }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/destroy.yml

```
# Copyright 2021 Ignacio Asin (iasinDev)
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#    http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to in writing, software
# distributed under the License is distributed on an "AS IS" BASIS,
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
# See the License for the specific language governing permissions and
# limitations under the License.

on:
  workflow_dispatch:
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/destroy.yml

```yaml
name: Destroy GCP resources
env:
  PROJECT_ID: ${{ secrets.GCP_PROJECT }}
  SERVICE: ${{ secrets.SERVICE_NAME }}
  REGION: ${{ secrets.GCP_REGION }}
  ZONE: ${{ secrets.GCP_ZONE }}

jobs:
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/destroy.yml

```yaml
destroy-terraform-resources:

  runs-on: ubuntu-latest

  steps:

  - name: Enable GitHub Actions
    uses: actions/checkout@v3.3.0

  - id: auth
    uses: google-github-actions/auth@v1
    with:
      credentials_json: ${{ secrets.GCP_SA_KEY }}

  - name: Set up Cloud SDK
    uses: google-github-actions/setup-gcloud@v1
    with:
      project_id: ${{ env.PROJECT_ID }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/destroy.yml

```yaml
- name: Getting tag of the built image
  run: |-
    image_tag=$(gcloud container images list-tags gcr.io/${{ env.PROJECT_ID }}/${{ env.SERVICE }} --limit=1 --sort-by=~TIMESTAMP --format='get(tags)')
    IFS=';' read -ra image_tag_ARRAY <<< "$image_tag"
    image_tag=${image_tag_ARRAY[0]}
    echo "image_tag: $image_tag"
    echo "image_tag=$image_tag" >> $GITHUB_ENV

# Install the latest version of Terraform CLI and configure the Terraform CLI configuration file with a Terraform Cloud user API token
- name: Setup Terraform
  uses: hashicorp/setup-terraform@v2.0.3
```

Continuous Integration (CI) & Continuous Delivery (CD)

`.github/workflows/destroy.yml`

```yaml
# Initialize a new or existing Terraform working directory by creating initial files, loading any remote state, downloading modules, etc.
- name: Terraform Init
  run: terraform init
  env:
    GOOGLE_CREDENTIALS: ${{ secrets.GCP_SA_KEY }}
    TF_VAR_project_id: ${{ env.PROJECT_ID }}
    TF_VAR_region: ${{ env.REGION }}
    TF_VAR_zone: ${{ secrets.GCP_ZONE }}
```

Continuous Integration (CI) & Continuous Delivery (CD)

.github/workflows/destroy.yml

```yaml
# Destroy the resources
- name: Terraform Destroy
  run: terraform destroy -lock=false -auto-approve
  env:
    GOOGLE_CREDENTIALS: ${{ secrets.GCP_SA_KEY }}
    TF_VAR_project_id: ${{ env.PROJECT_ID }}
    TF_VAR_region: ${{ env.REGION }}
    TF_VAR_zone: ${{ env.ZONE }}
    TF_VAR_service: ${{ env.SERVICE }}
    TF_VAR_image_tag: ${{ env.image_tag }}
```

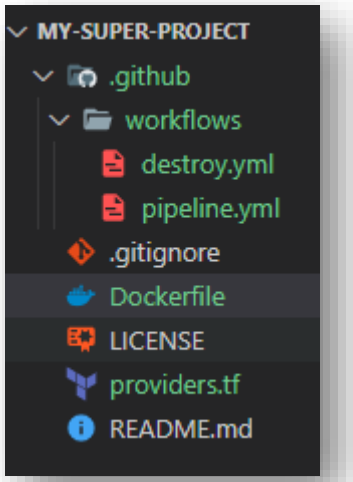
WHAT IS DEVOPS & DEVSECOPS?

“Application as Code” … “Platform as Code”

docker

Containerization

Dockerfile



Dockerfile Multistage

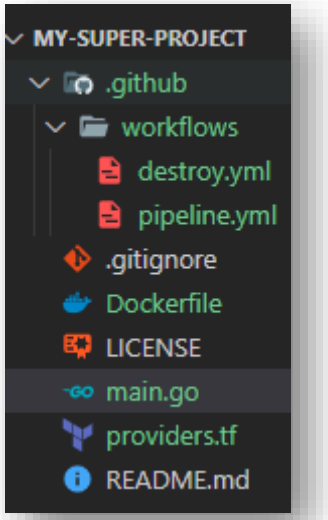
```
# Use the official Golang image to create a build artifact.
# This is based on Debian and sets the GOPATH to /go.
# https://hub.docker.com/_/golang
FROM golang:1.13 as builder

# Create and change to the app directory.
WORKDIR /app

# Copy local code to the container image.
COPY ./main.go ./main.go

# Build the binary.
RUN CGO_ENABLED=0 GOOS=linux go build -v -o server

# Use the official Alpine image for a lean production container.
# https://hub.docker.com/_/alpine
# https://docs.docker.com/develop/develop-images/multistage-build/#use-multi-stage-builds
FROM alpine:3
RUN apk add --no-cache ca-certificates

# Copy the binary to the production image from the builder stage.
COPY --from=builder /app/server /server
COPY ./snake.html ./snake.html

# Run the web service on container startup.
CMD ["/server"]
```

WHAT IS DEVOPS & DEVSECOPS?

“Application as Code” … “Platform as Code”

docker

main.go




```go
package main

import (
    "fmt"
    "log"
    "net/http"
    "io/ioutil"
)

func handler(w http.ResponseWriter, r *http.Request) {
    file, err := ioutil.ReadFile("./snake.html")
    if err != nil {
      fmt.Print(err)
    }
    log.Print("Serving the body")
    log.Print(string(file))
    body := string(file)
    fmt.Fprintf(w, body)
}

func main() {
    log.Print("Hello world webapp started.")

    http.HandleFunc("/", handler)

    port := "8080"

    log.Fatal(http.ListenAndServe(fmt.Sprintf(":%s", port), nil))
}
```

WHAT IS DEVOPS & DEVSECOPS?

“Application as Code” … “Platform as Code”

snake.html

Copy from the solution

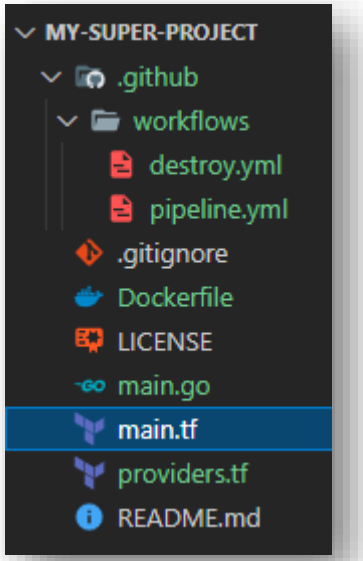
```
<!DOCTYPE html>
<html>
<head>
  <title></title>
  <style>
  html, body {
    height: 100%;
    margin: 0;
  }

  body {
    background: black;
    display: flex;
    align-items: center;
    justify-content: center;
  }
  canvas {
}

…
```

WHAT IS DEVOPS & DEVSECOPS?

Infrastructure as Code

Terraform

main.tf




```
resource "google_cloud_run_service" "edemdevsecops" {
  name     = "edemdevsecops"
  location = var.region

  template {
    spec {
      containers {
        image = "gcr.io/${var.project_id}/${var.service}:${var.image_tag}"
      }
    }
  }

  traffic {
    percent         = 100
    latest_revision = true
  }
}
```

WHAT IS DEVOPS & DEVSECOPS?

Infrastructure as Code

Terraform

main.tf

```
data "google_iam_policy" "noauth" {
  binding {
    role = "roles/run.invoker"
    members = [
      "allUsers",
    ]
  }
}

resource "google_cloud_run_service_iam_policy" "noauth" {

  depends_on = [
    google_cloud_run_service.edemdevsecops,
  ]

  location    = var.region
  project     = var.project_id
  service     = var.service

  policy_data = data.google_iam_policy.noauth.policy_data
}
```

Infrastructure as Code

providers.tf

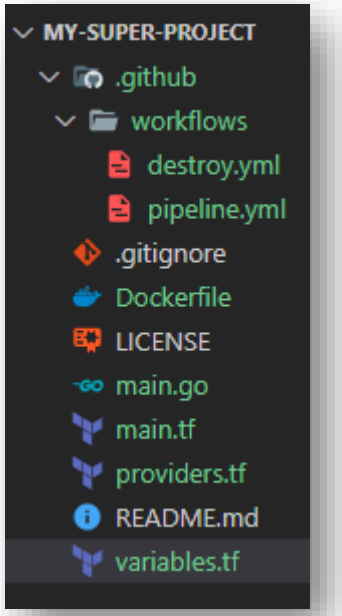
```
provider "google" {
  project = var.project_id
  region  = var.region
  zone    = var.zone
}

terraform {
  backend "gcs" {
    bucket = "tf-state-prod-edemdevsecops-2023"
    prefix = "terraform/state"
  }
}
```

WHAT IS DEVOPS & DEVSECOPS?

Infrastructure as Code

Terraform

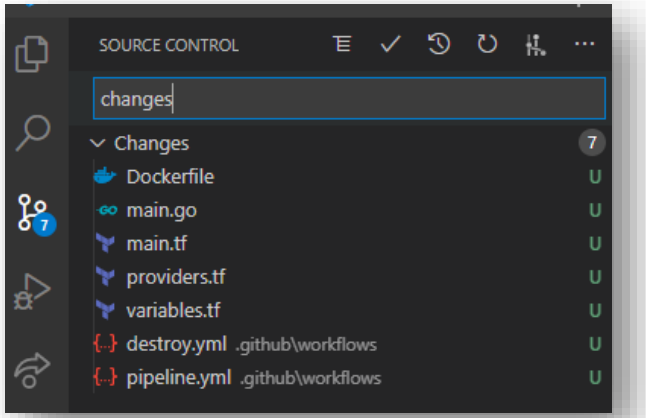
variables.tf

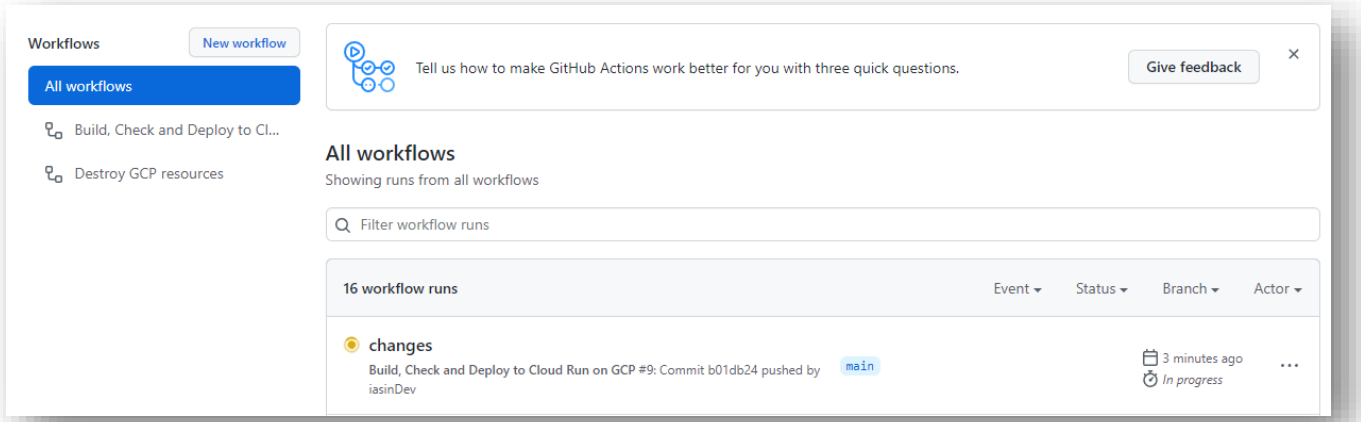



```
variable "project_id" {
    type  = string
}

variable "region" {
    type  = string
}

variable "zone" {
    type  = string
}

variable "service" {
    type  = string
}

variable "image_tag" {
    type  = string
}
```

Testing

```
commit & push your changes to GitHub
```

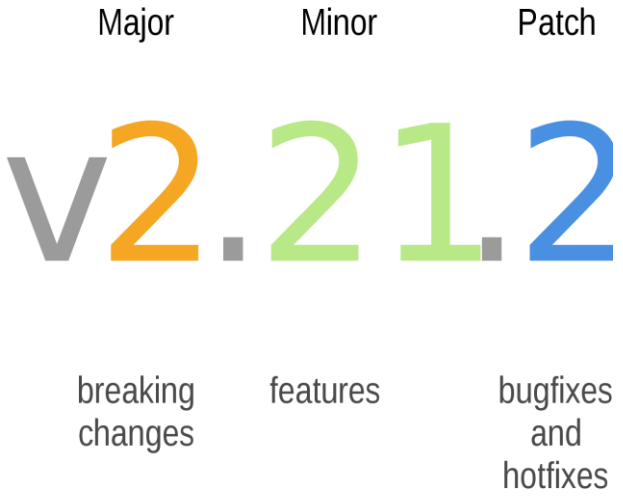


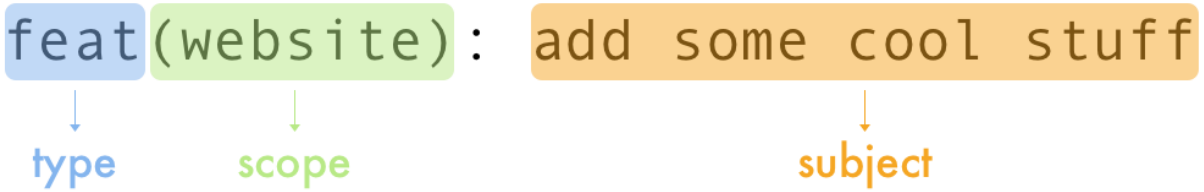


Testing

PLUS

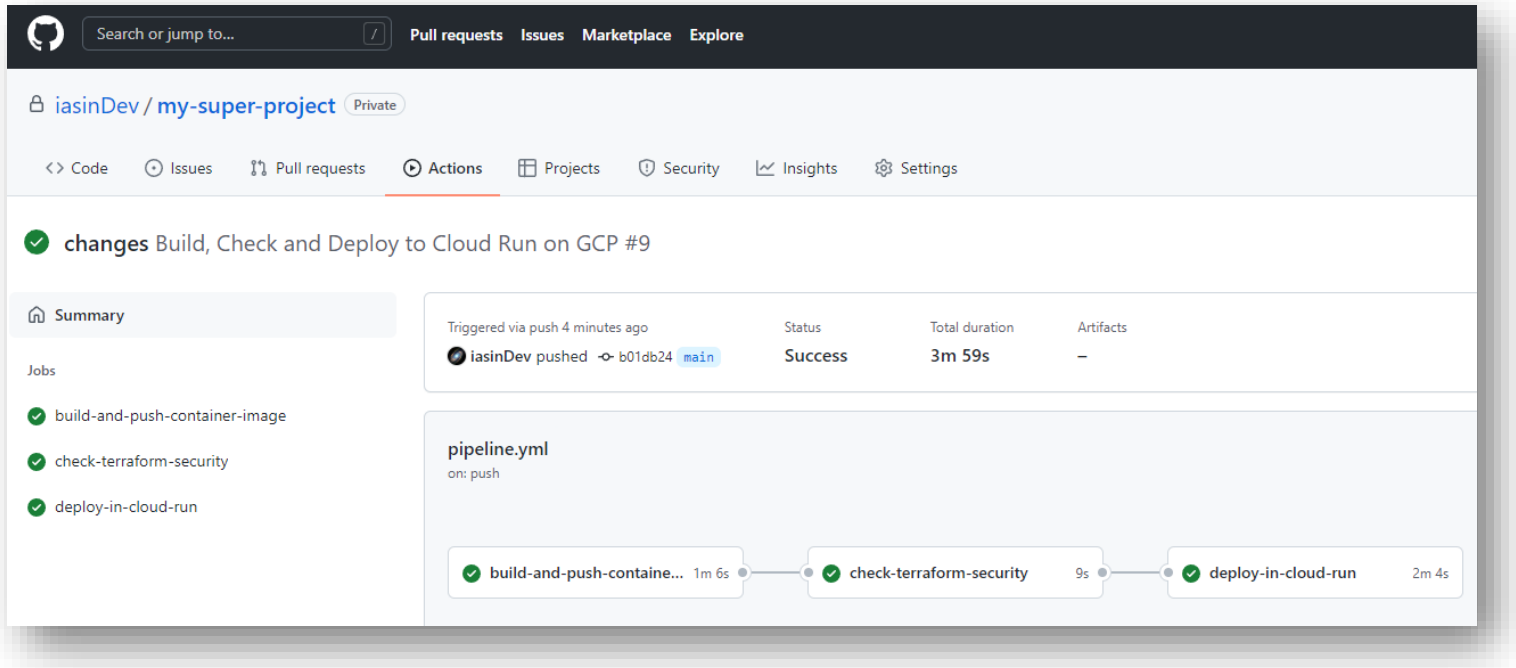
Semantic Versioning & Semantic Commit Messages





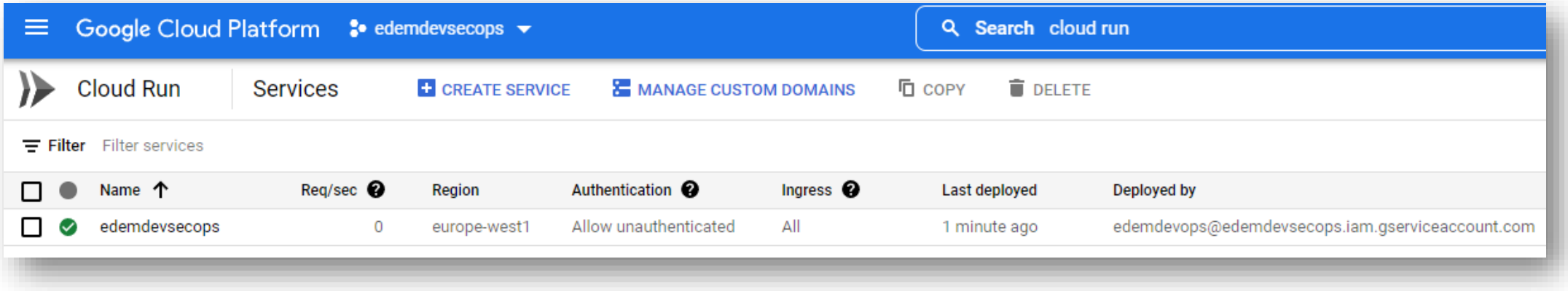
Testing

See the pipeline execution



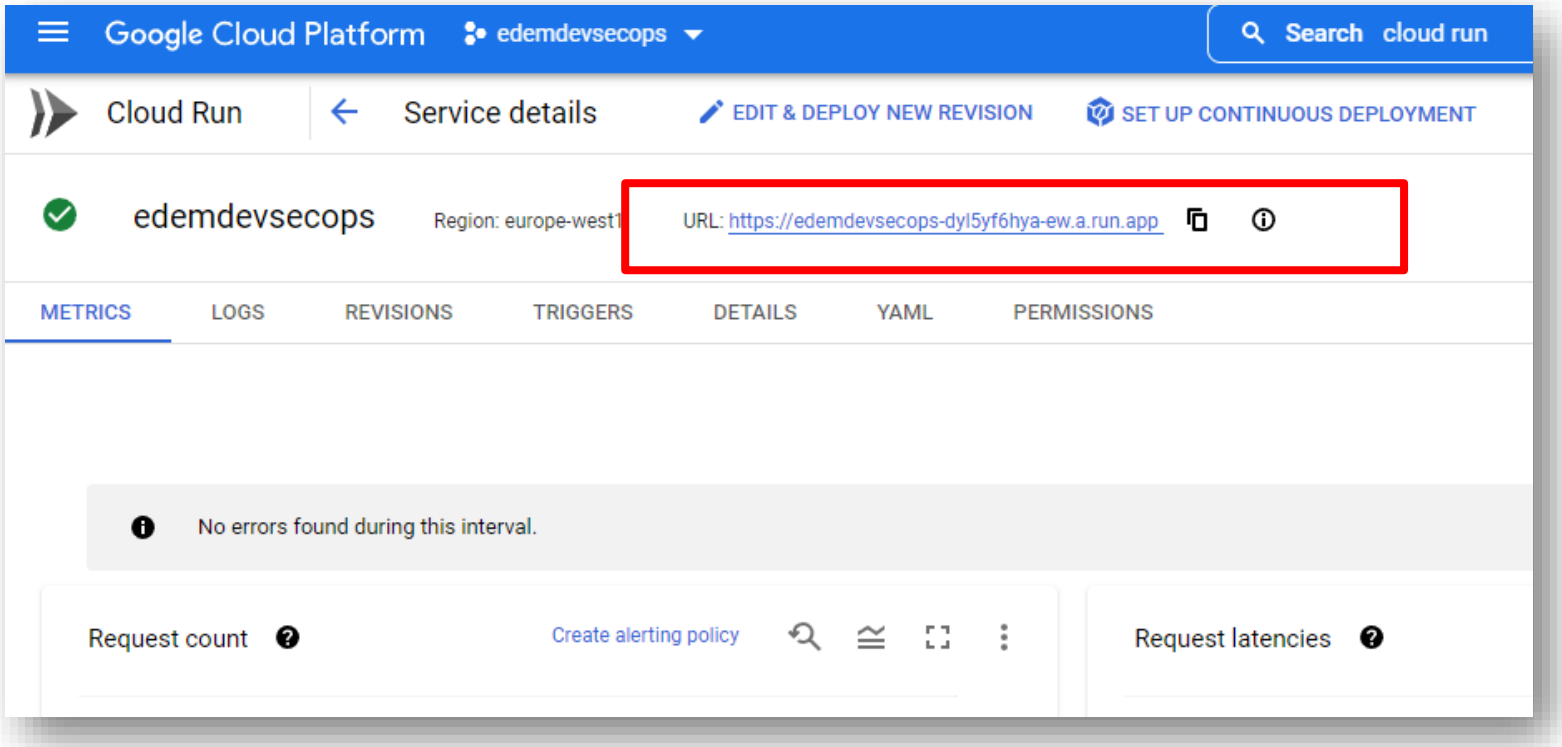
Testing

See the pipeline execution



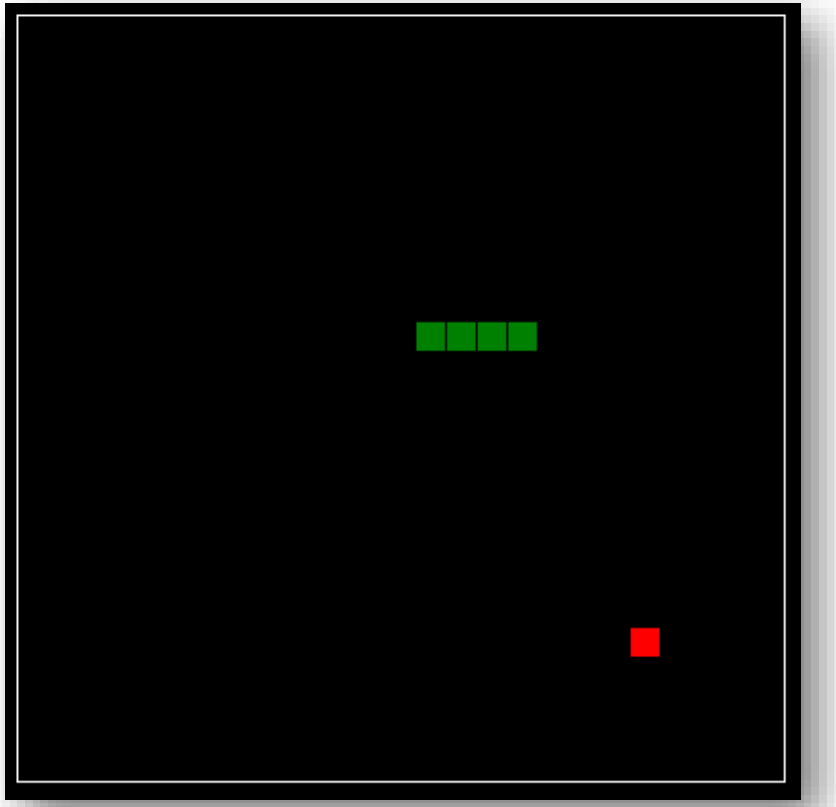
Testing

See the pipeline execution



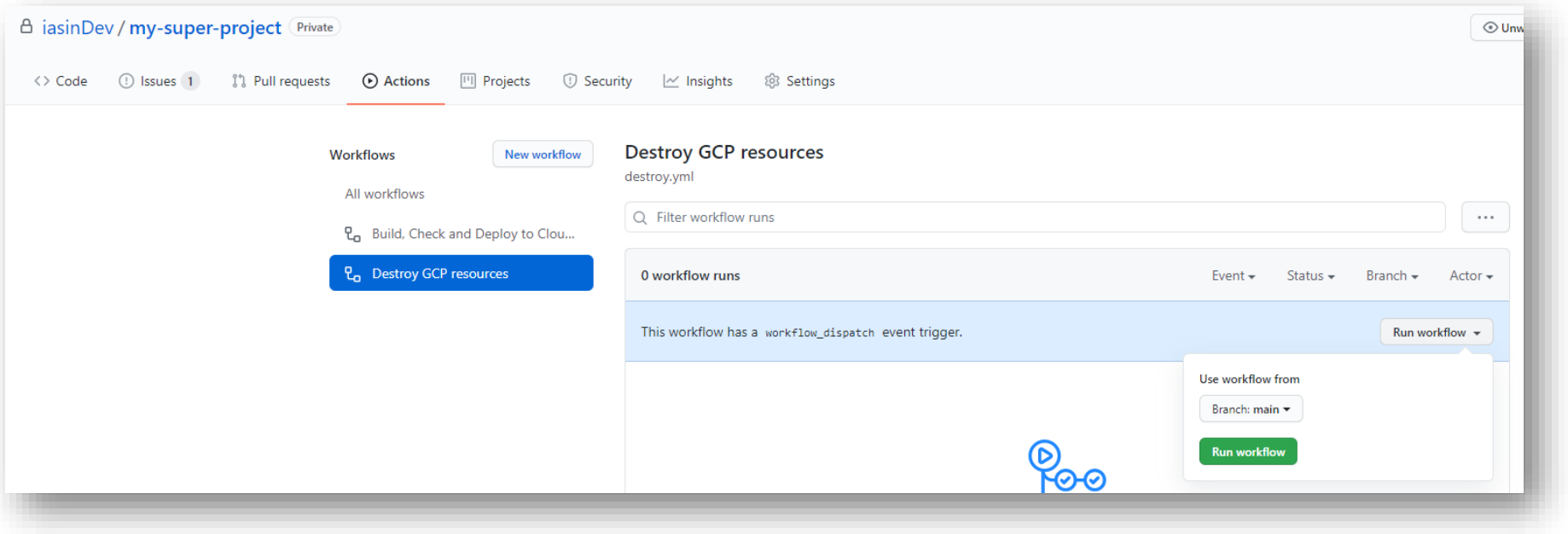
Testing

See the pipeline execution



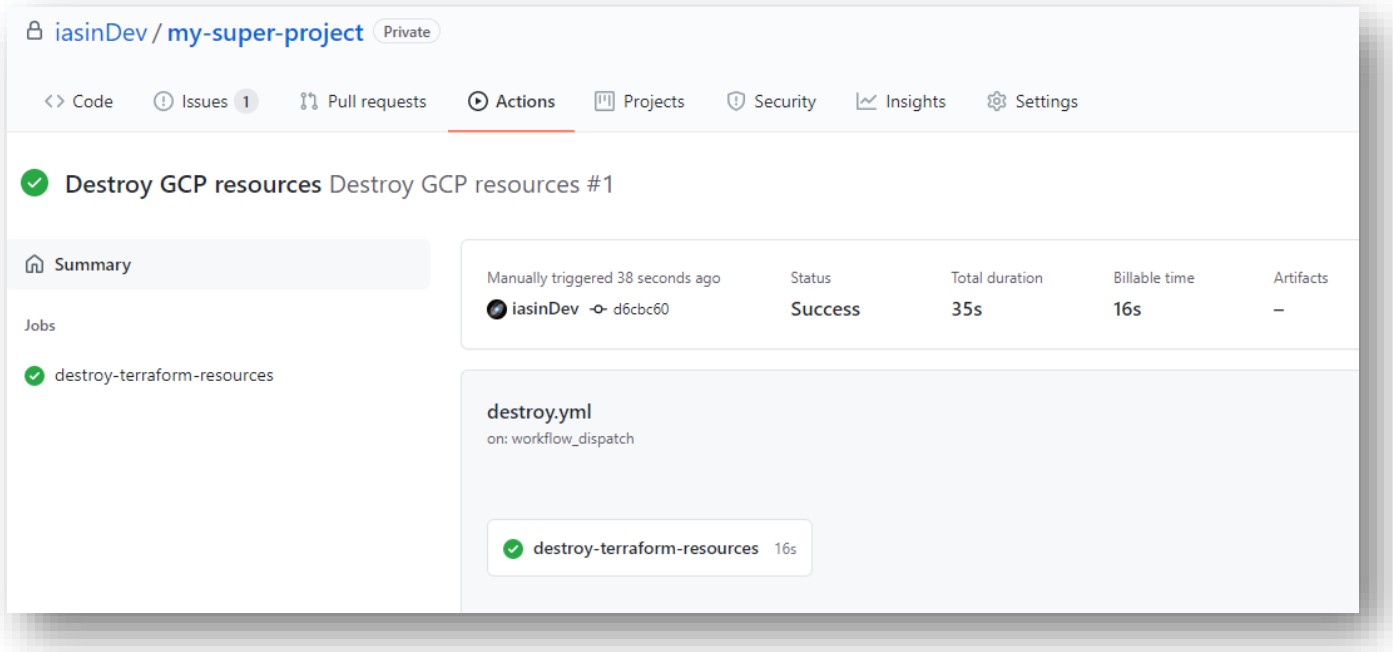
Destroy All

Manually launch the destroy pipeline



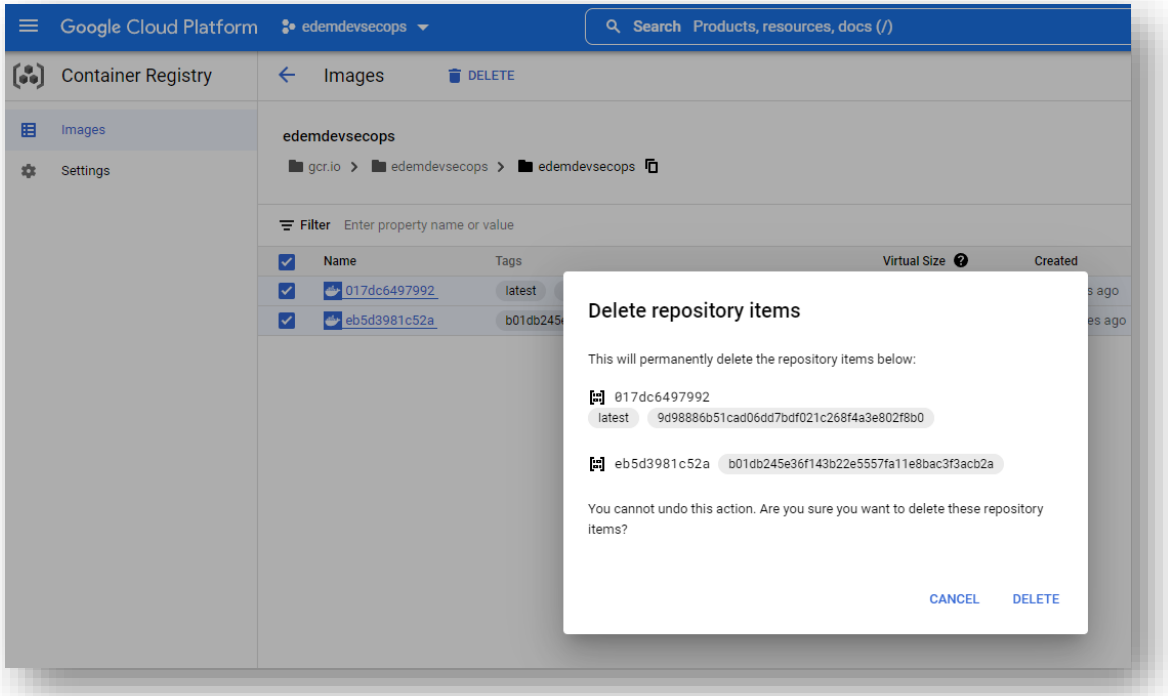
Destroy All

Manually launch the destroy pipeline



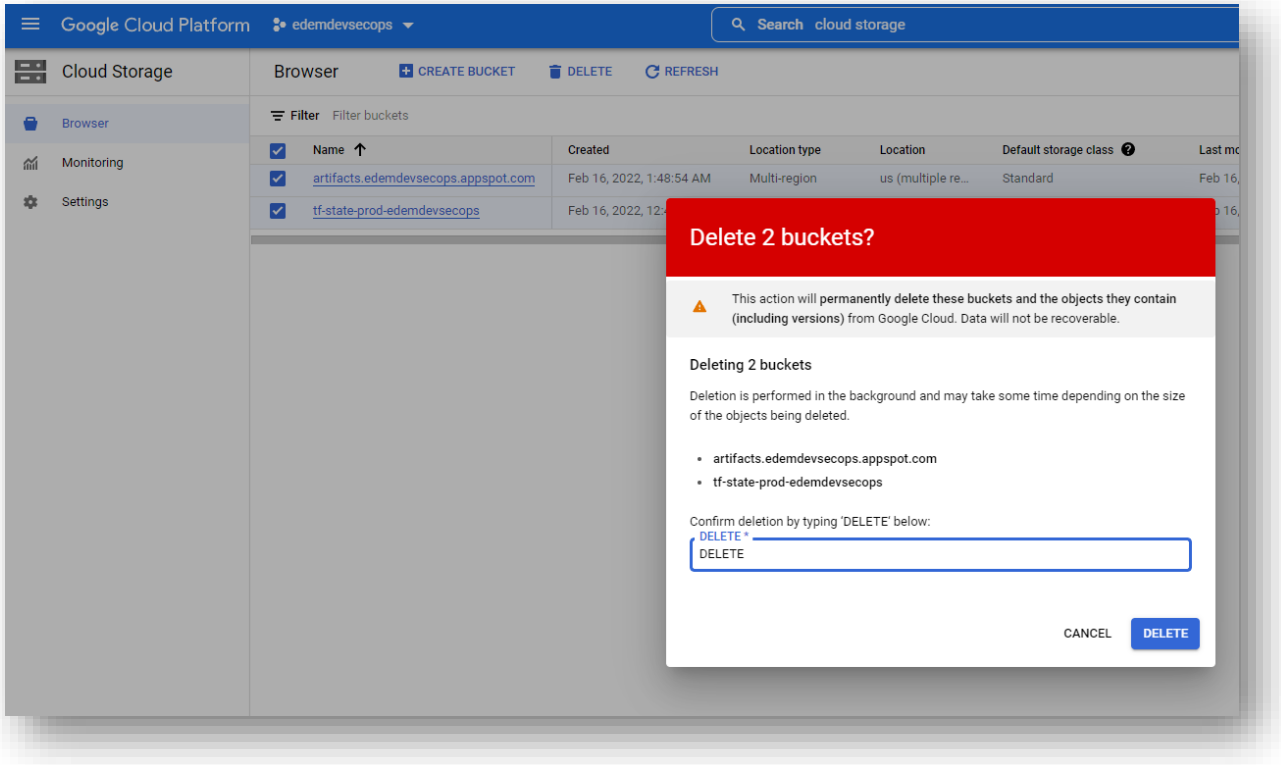
Destroy All

```
Delete all Container Registry images
```



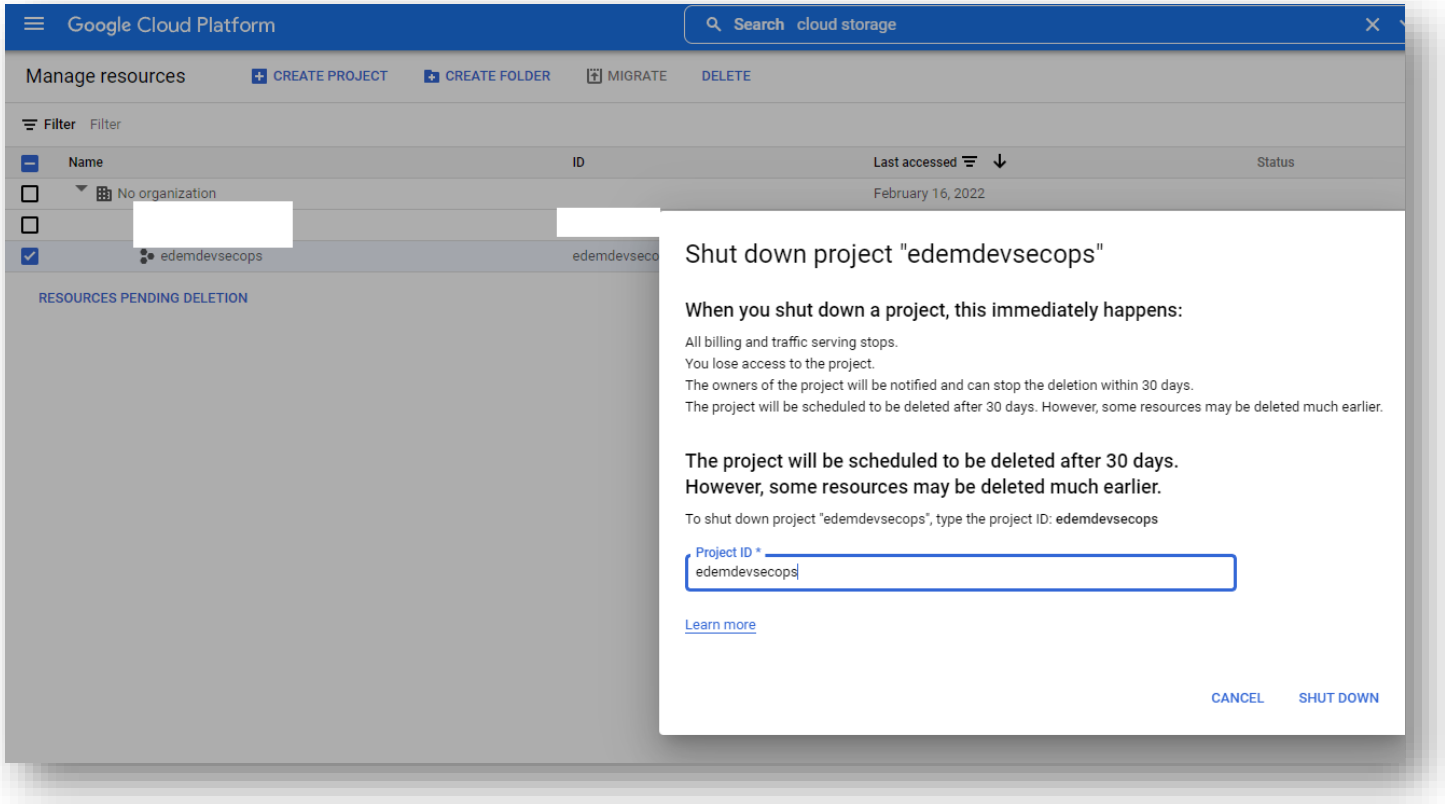
Destroy All

```
Delete the buckets created
```



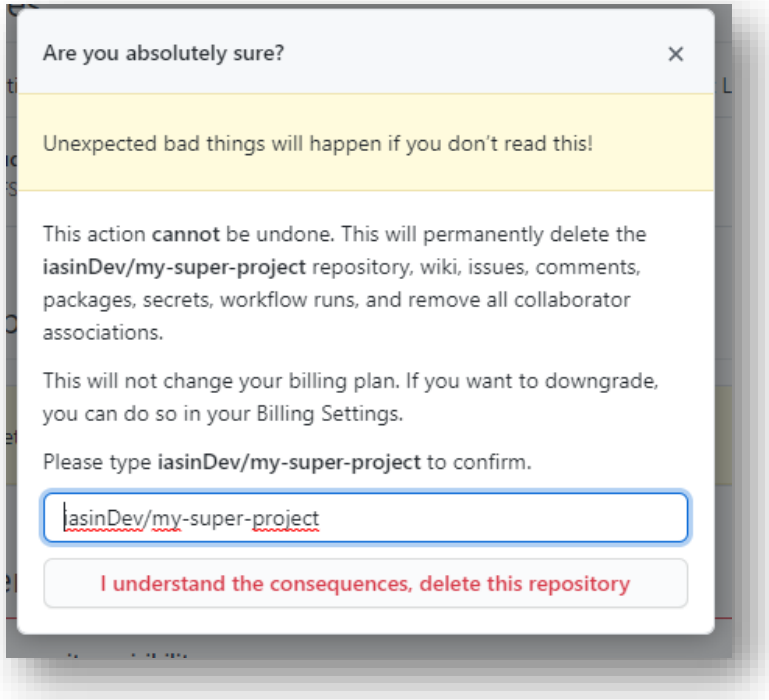
Destroy All

Delete the GCP project in resource manager



Destroy All

`Delete the GitHub Project`



AGENDA

EDEM
Escuela de Empresarios

QUESTIONS??