

# DevSecOps



Ignacio Asín Calamonte

Curso 21-22 – 3ª Edición

Fecha 19/02/2022

# 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.
- Favorite topics: **Automation, Google Cloud, Blockchain, latest Tech trends, ...**
- Project Architect/Developer, Research, Formation courses, Meetups, ...



[@iasinDev](https://twitter.com/iasinDev)



[ignacioasincalamonte](https://www.linkedin.com/in/ignacioasincalamonte)



[iasin.dev@gmail.com](mailto:iasin.dev@gmail.com)

# AGENDA

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

**Questions??**

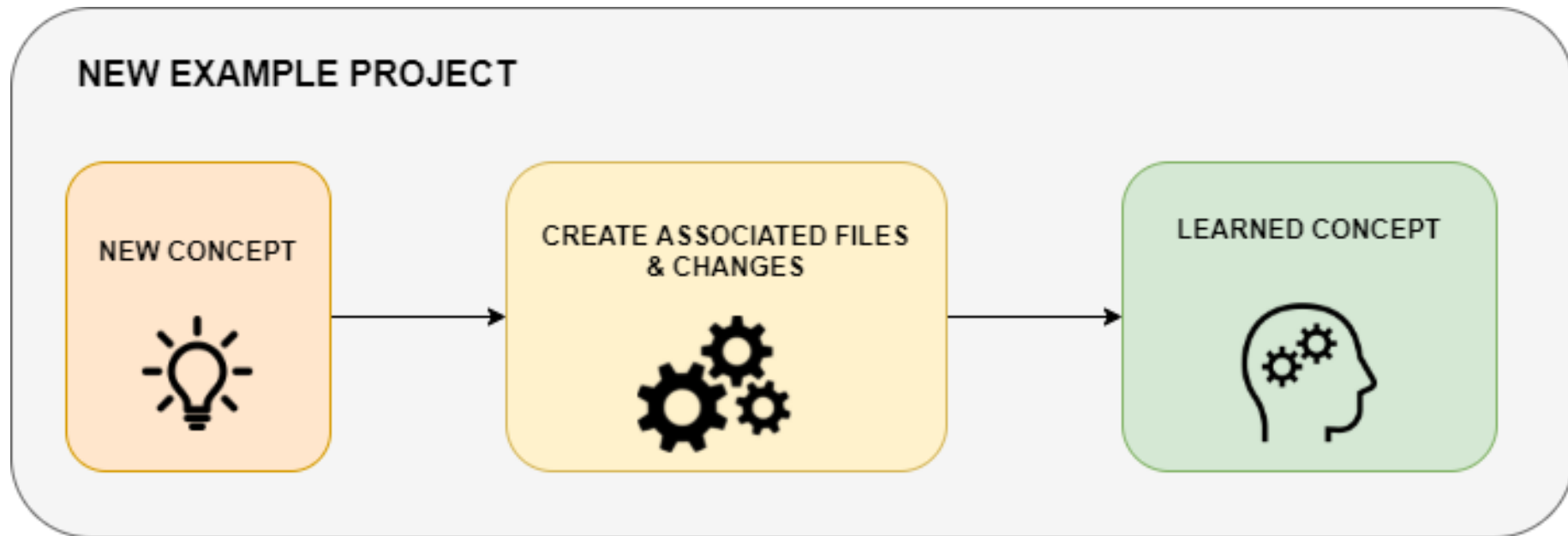
# 1. FORMAT OF THE CLASS

## FORMAT OF THE CLASS

*“Learning by doing”*

# FORMAT OF THE CLASS

*“Learning by doing”*



## 2. LOCAL ENVIRONMENT

# LOCAL ENVIRONMENT

- [Visual Studio Code](#)
- [GitHub Account](#)
- [Google Cloud Platform \(GCP\) account](#)
- [Google Chrome](#) / similar browser
- [Git for Windows](#)
- [Notepad++](#)



## 3. WHAT IS DEVOPS & DEVSECOPS?

## WHAT IS DEVOPS & DEVSECOPS?



What **IS NOT** DevOps?



New  
Superhero



Profession  
(not only)

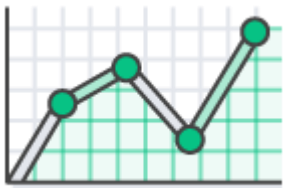


Culture  
(not only)

# WHAT IS DEVOPS & DEVSECOPS?



## What **IS** DevOps?



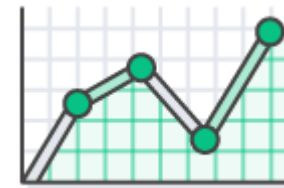
Velocity



Faster  
releases



Reliability



Scale



Improved  
collaboration



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

## WHAT IS DEVOPS & DEVSECOPS?



What **IS** DevOps?



DIFFICULT TO FIND !!

## WHAT IS DEVOPS & DEVSECOPS?



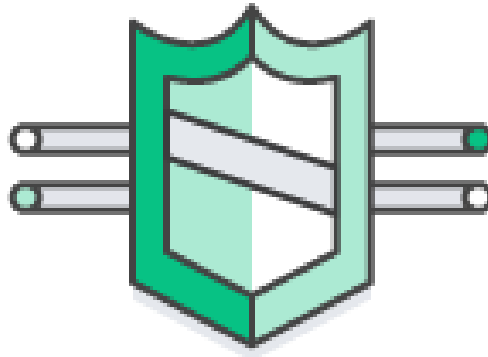
What **IS** DevOps?



## WHAT IS DEVOPS & DEVSECOPS?



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



Security

# WHAT IS DEVOPS & DEVSECOPS?



## Everything as Code



Infrastructure as Code

(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)



2. Clone the new Project in local

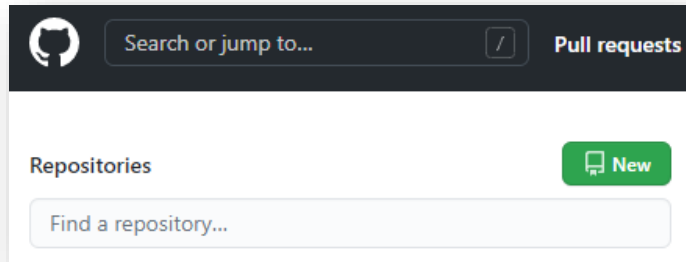


3. Open the new Project with Visual Studio Code



# WHAT IS DEVOPS & DEVSECOPS?

## Everything as Code



### Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner \*  / Repository name \*

Great repository names are short and memorable. Need inspiration? How about [symmetrical-train](#)?

Description (optional)

- ☐ Public  
Anyone on the internet can see this repository. You choose who can commit.
- ☒ Private  
You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

☒ Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

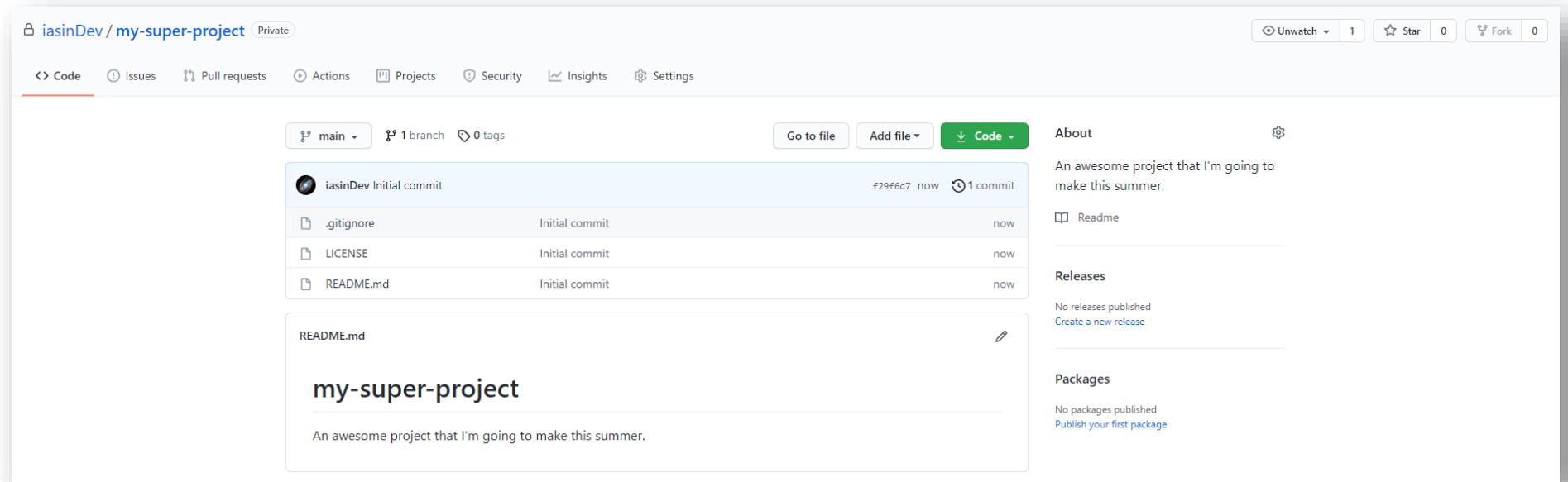
☒ Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  as the default branch. Change the default name in your [settings](#).



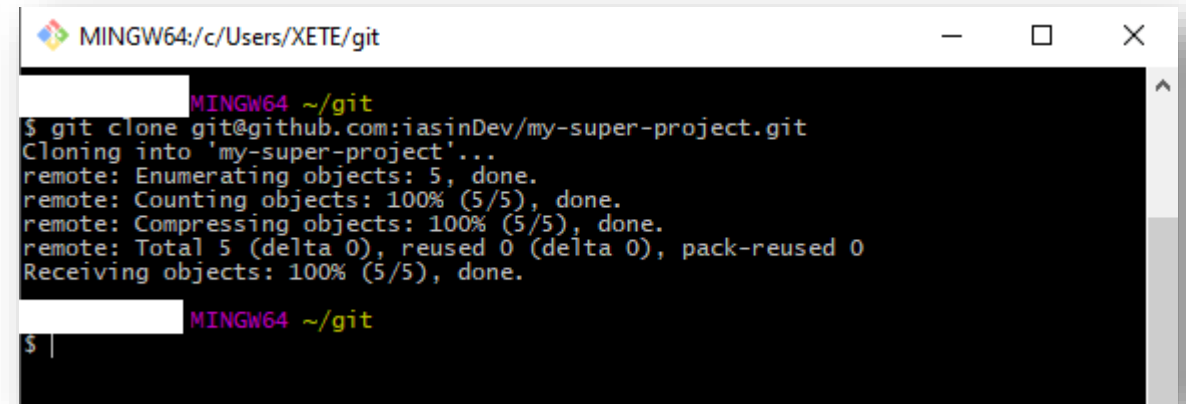
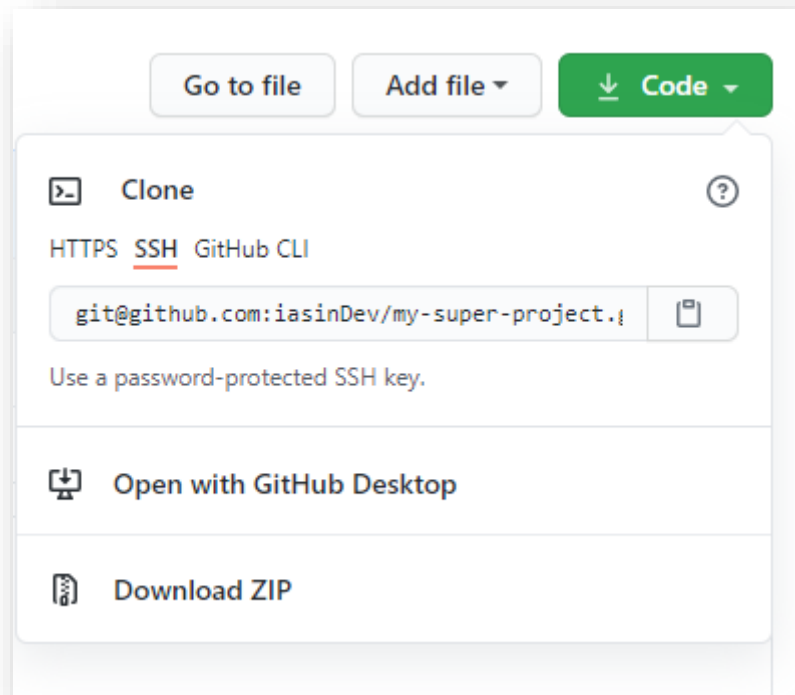
# WHAT IS DEVOPS & DEVSECOPS?

## Everything as Code



# WHAT IS DEVOPS & DEVSECOPS?

## Everything as Code

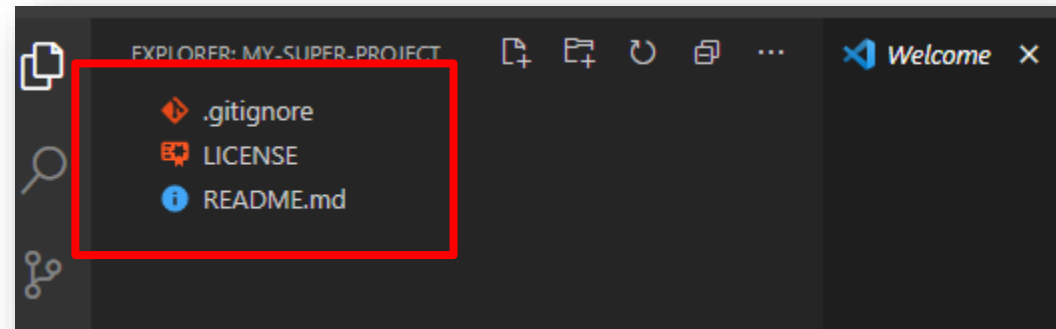


# WHAT IS DEVOPS & DEVSECOPS?

Everything as Code



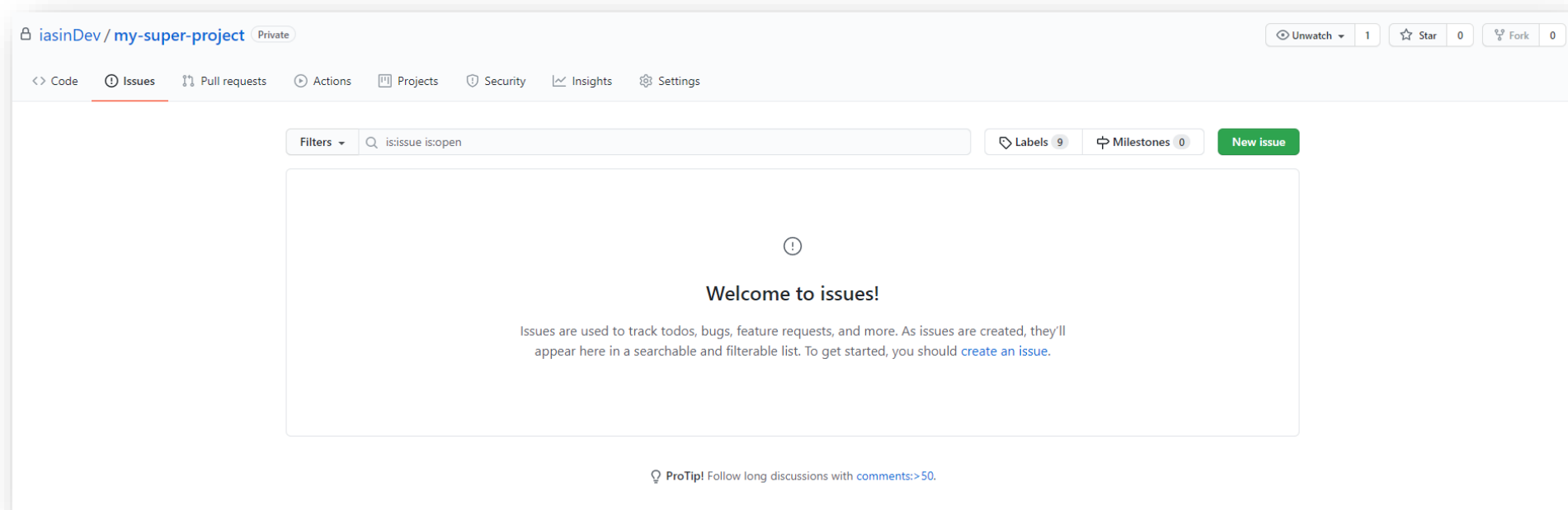
```
MINGW64 ~/git  
$ cd my-super-project/  
MINGW64 ~/git/my-super-project (main)  
$ code .
```



# WHAT IS DEVOPS & DEVSECOPS?



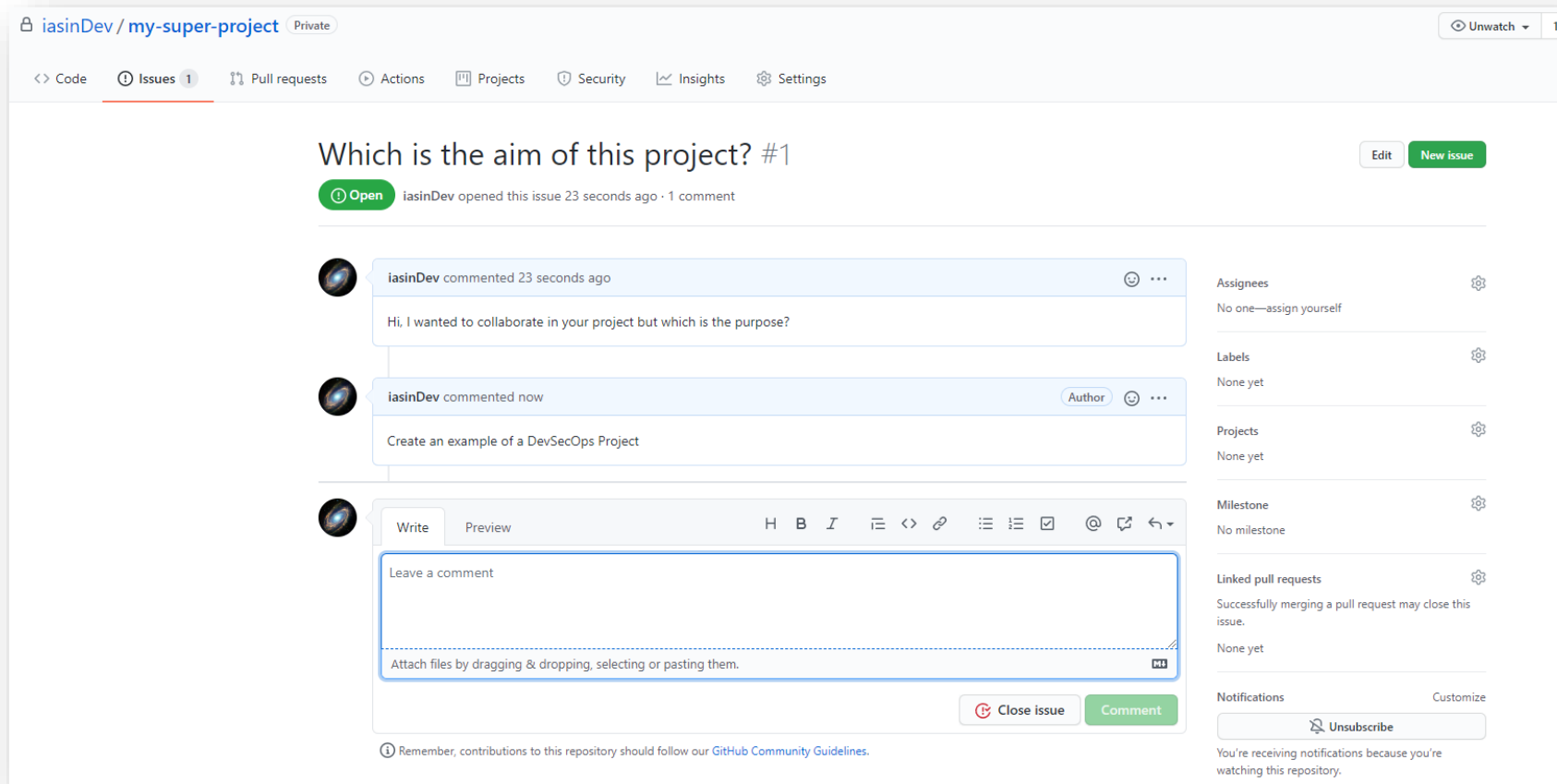
## Communication & Collaboration



# WHAT IS DEVOPS & DEVSECOPS?



## Communication & Collaboration



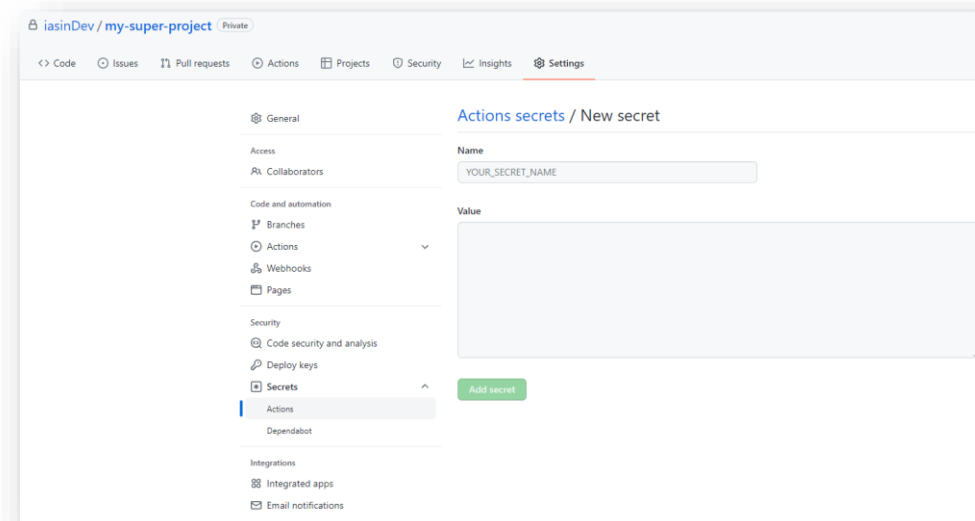
The screenshot shows a GitHub issue page for the repository 'iasinDev / my-super-project'. The issue title is 'Which is the aim of this project? #1'. The issue is marked as 'Open' and was opened by 'iasinDev' 23 seconds ago, with 1 comment. The issue page includes a navigation bar with links to Code, Issues (1), Pull requests, Actions, Projects, Security, Insights, and Settings. The main content area shows two comments from 'iasinDev'. The first comment, posted 23 seconds ago, says 'Hi, I wanted to collaborate in your project but which is the purpose?'. The second comment, posted 'now', says 'Create an example of a DevSecOps Project'. Below the comments is a text input field for leaving a comment, with a 'Write' tab and a 'Preview' tab. The input field has a rich text editor toolbar with options for bold, italic, link, and other formatting. To the right of the comments is a sidebar with various issue settings: Assignees (No one—assign yourself), Labels (None yet), Projects (None yet), Milestone (No milestone), Linked pull requests (Successfully merging a pull request may close this issue. None yet), and Notifications (Unsubscribe). At the bottom of the page, there is a footer note: 'Remember, contributions to this repository should follow our GitHub Community Guidelines.'

# WHAT IS DEVOPS & DEVSECOPS?



## Security Management

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



GitHub Secrets

# WHAT IS DEVOPS & DEVSECOPS?



## 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



**New Project**

You have 24 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name \*  
edemdevsecops

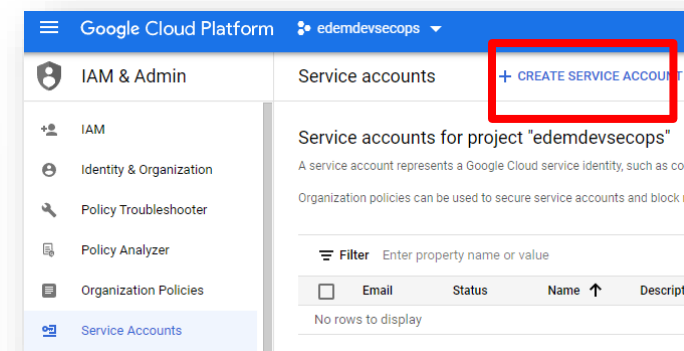
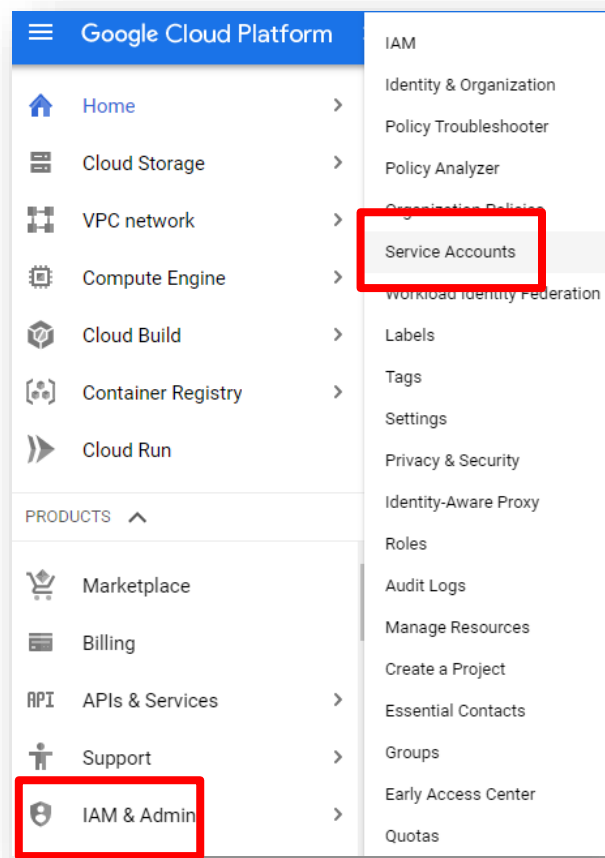
Project ID \*  
edemdevsecops

Project ID can have lowercase letters, digits, or hyphens. It must start with a lowercase letter and end with a letter or number.

Location \*  
 No organization [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)



**Create service account**

**1 Service account details**

Service account name \*  
edemdevsecops

Display name for this service account

Service account ID \*  
edemdevsecops @edemdevsecops.iam.gserviceaccount.com

Service account description  
Account used for the edemdevsecops course [Clear](#)

[CREATE AND CONTINUE](#)

**2 Grant this service account access to project (optional)**

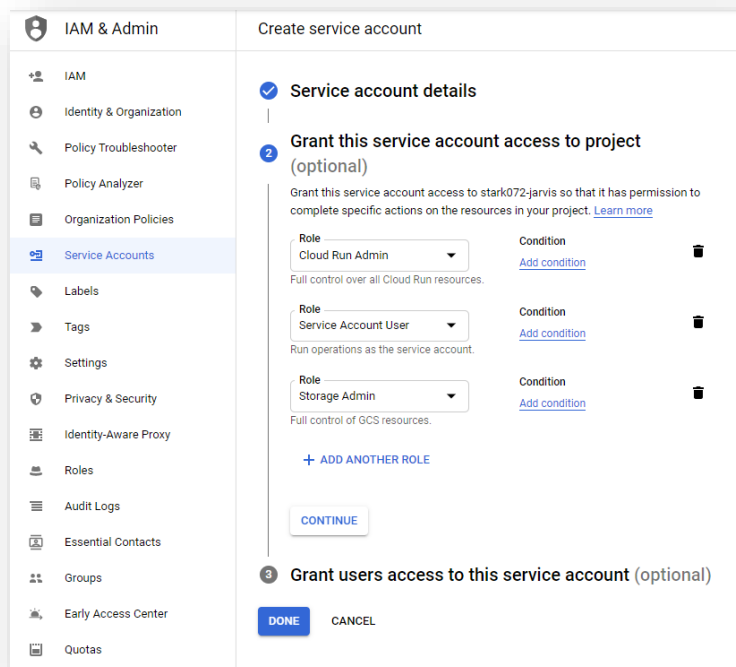
**3 Grant users access to this service account (optional)**

[DONE](#) [CANCEL](#)

# WHAT IS DEVOPS & DEVSECOPS?



## Security Management



**IAM & Admin** | Create service account

**Service account details**

**2 Grant this service account access to project (optional)**

Grant this service account access to stark072-jarvis so that it has permission to complete specific actions on the resources in your project. [Learn more](#)

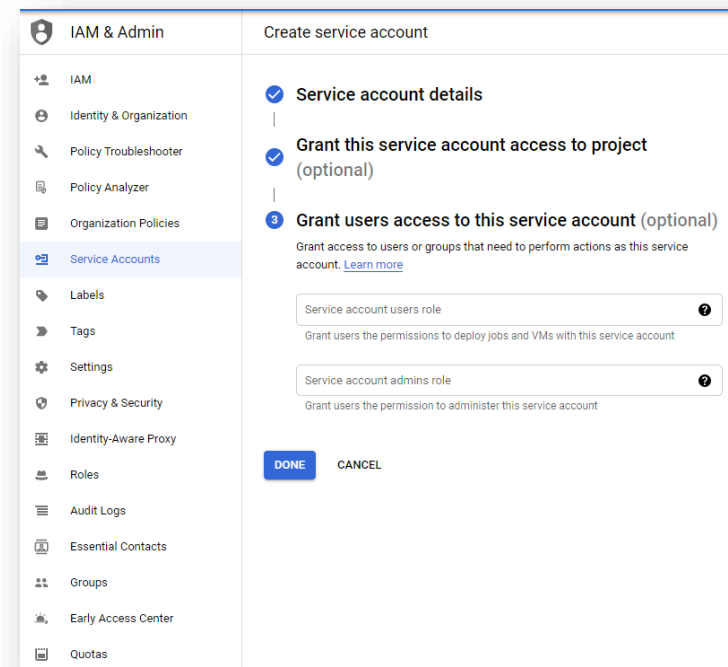
Role	Condition	
Cloud Run Admin	<a href="#">Add condition</a>	
Full control over all Cloud Run resources.		
Service Account User	<a href="#">Add condition</a>	
Run operations as the service account.		
Storage Admin	<a href="#">Add condition</a>	
Full control of GCS resources.		

[+ ADD ANOTHER ROLE](#)

[CONTINUE](#)

**3 Grant users access to this service account (optional)**

[DONE](#) [CANCEL](#)



**IAM & Admin** | Create service account

**Service account details**

**Grant this service account access to project (optional)**

**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

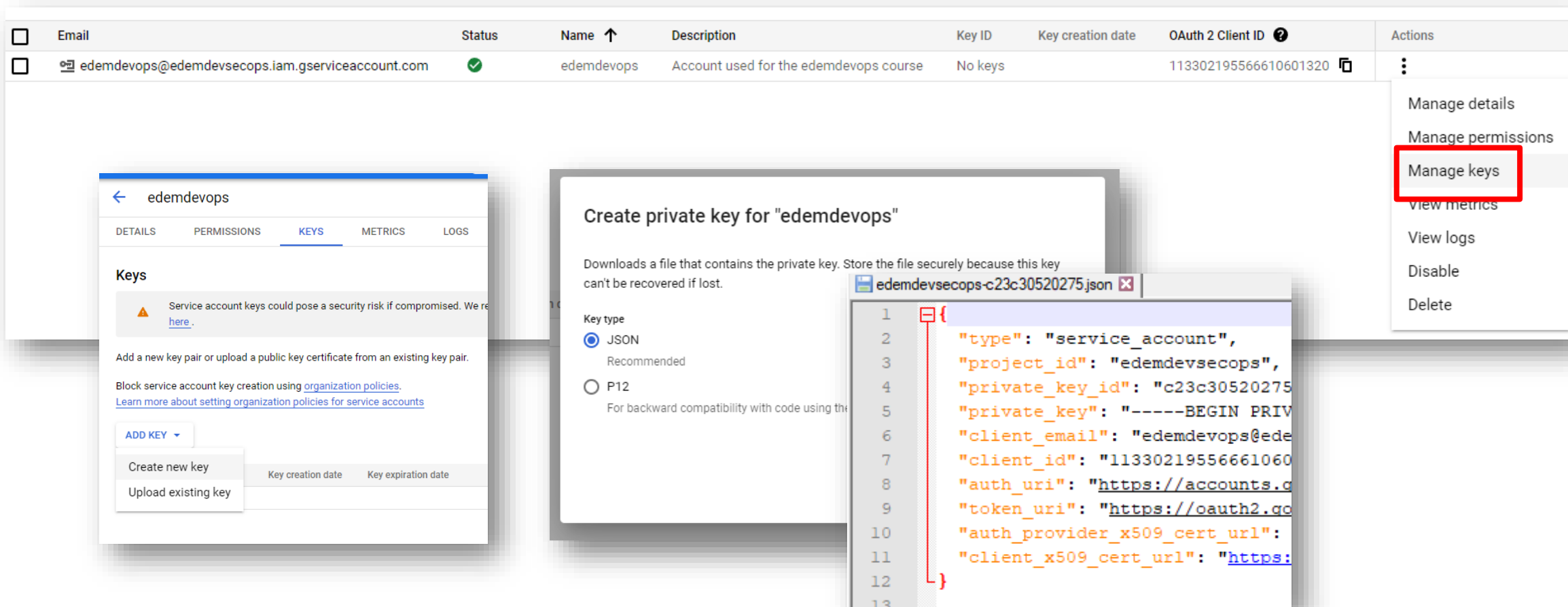
[DONE](#) [CANCEL](#)

<input type="checkbox"/>	Email	Status	Name ↑	Description	Key ID	Key creation date	OAuth 2 Client ID ?	Actions
<input type="checkbox"/>	edemdevops@edemdevsecops.iam.gserviceaccount.com	✓	edemdevops	Account used for the edemdevops course	No keys		113302195566610601320	

# WHAT IS DEVOPS & DEVSECOPS?



## Security Management



The screenshot displays the Google Cloud IAM console interface for managing service account keys. The main table lists service accounts, with 'edemdevops' selected. The 'Actions' menu for this account is open, highlighting 'Manage keys'. Below this, the 'Keys' tab for 'edemdevops' is shown, indicating that no keys are currently present. A 'Create private key for "edemdevops"' dialog is open, showing the 'JSON' key type selected. A preview of the resulting JSON key file is also visible, showing fields like 'type', 'project\_id', 'private\_key\_id', 'private\_key', 'client\_email', 'client\_id', 'auth\_uri', 'token\_uri', 'auth\_provider\_x509\_cert\_url', and 'client\_x509\_cert\_url'.

Email	Status	Name	Description	Key ID	Key creation date	OAuth 2 Client ID	Actions
edemdevops@edemdevsecops.iam.gserviceaccount.com		edemdevops	Account used for the edemdevops course	No keys		113302195566610601320	<ul style="list-style-type: none"><li>Manage details</li><li>Manage permissions</li><li><b>Manage keys</b></li><li>View metrics</li><li>View logs</li><li>Disable</li><li>Delete</li></ul>

**Keys**

Service account keys could pose a security risk if compromised. We recommend you rotate keys regularly. [Learn more](#).

Add a new key pair or upload a public key certificate from an existing key pair.

Block service account key creation using [organization policies](#).  
[Learn more about setting organization policies for service accounts](#)

**ADD KEY**

- Create new key
- Upload existing key

**Create private key for "edemdevops"**

Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost.

**Key type**

- ☒ JSON  
Recommended
- ☐ P12  
For backward compatibility with code using the P12 key type

**edemdevsecops-c23c30520275.json**

```
1 {  
2   "type": "service_account",  
3   "project_id": "edemdevsecops",  
4   "private_key_id": "c23c30520275",  
5   "private_key": "-----BEGIN PRIVATE KEY-----",  
6   "client_email": "edemdevops@edemdevsecops.iam.gserviceaccount.com",  
7   "client_id": "113302195566610601320",  
8   "auth_uri": "https://accounts.google.com/o/oauth2/auth",  
9   "token_uri": "https://oauth2.googleapis.com/token",  
10  "auth_provider_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",  
11  "client_x509_cert_url": "https://www.googleapis.com/oauth2/v1/certs",  
12 }  
13
```

# WHAT IS DEVOPS & DEVSECOPS?

## Security Management

Add the following secrets to your GitHub project secrets:

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








### 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.

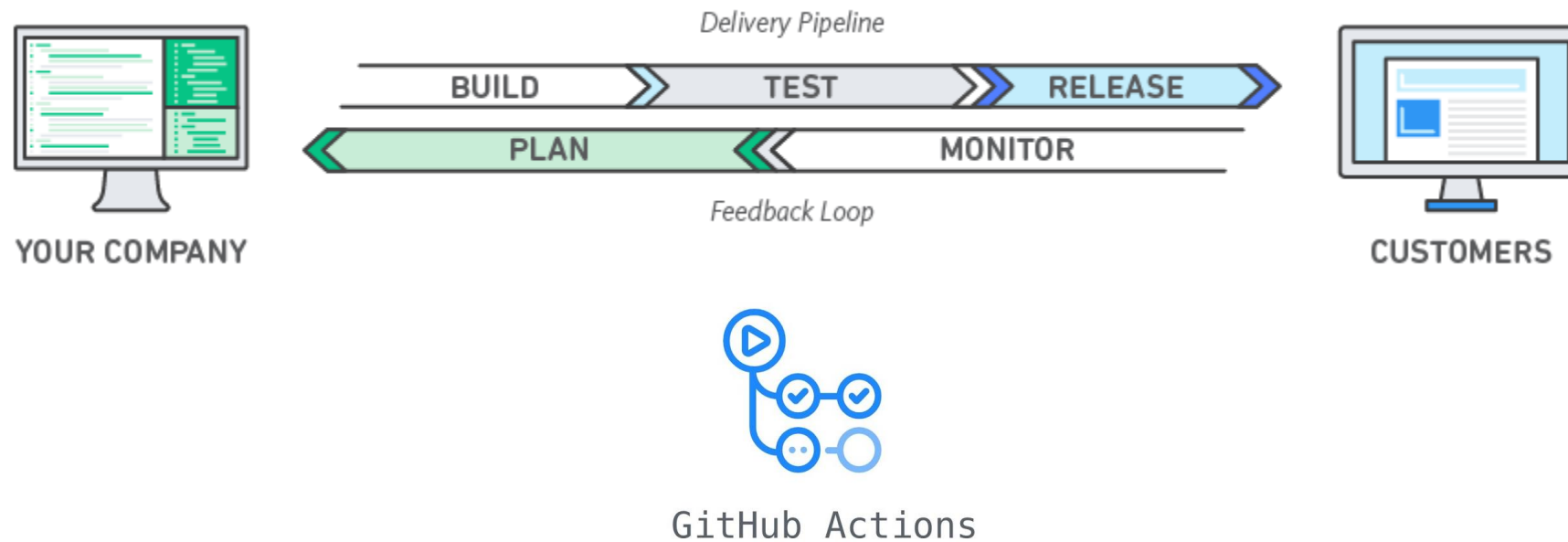
Secrets are not passed to workflows that are triggered by a pull request from a fork. [Learn more](#).

 <code>GCP_PROJECT</code>	Updated 1 minute ago	<a href="#">Update</a>	<a href="#">Remove</a>
 <code>GCP_REGION</code>	Updated 31 seconds ago	<a href="#">Update</a>	<a href="#">Remove</a>
 <code>GCP_SA_KEY</code>	Updated 1 minute ago	<a href="#">Update</a>	<a href="#">Remove</a>
 <code>GCP_ZONE</code>	Updated 12 seconds ago	<a href="#">Update</a>	<a href="#">Remove</a>
 <code>SERVICE_NAME</code>	Updated now	<a href="#">Update</a>	<a href="#">Remove</a>

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



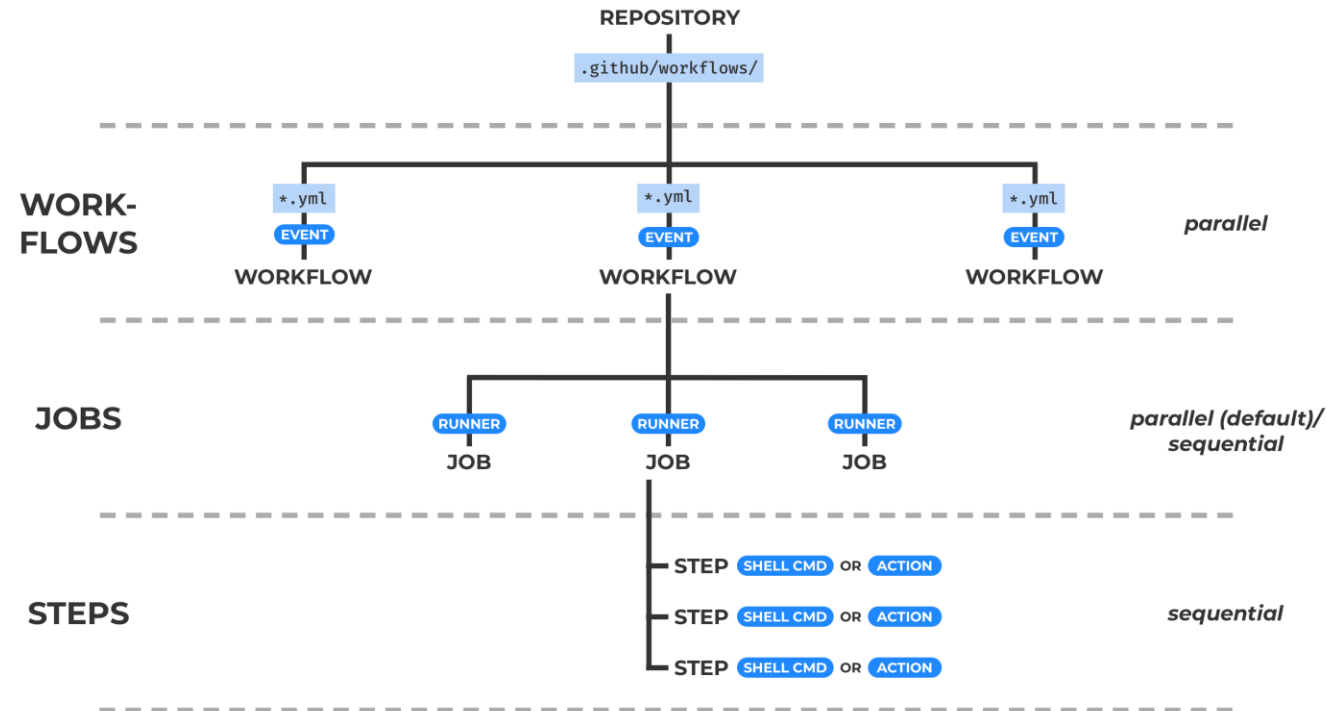
# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



GitHub Actions



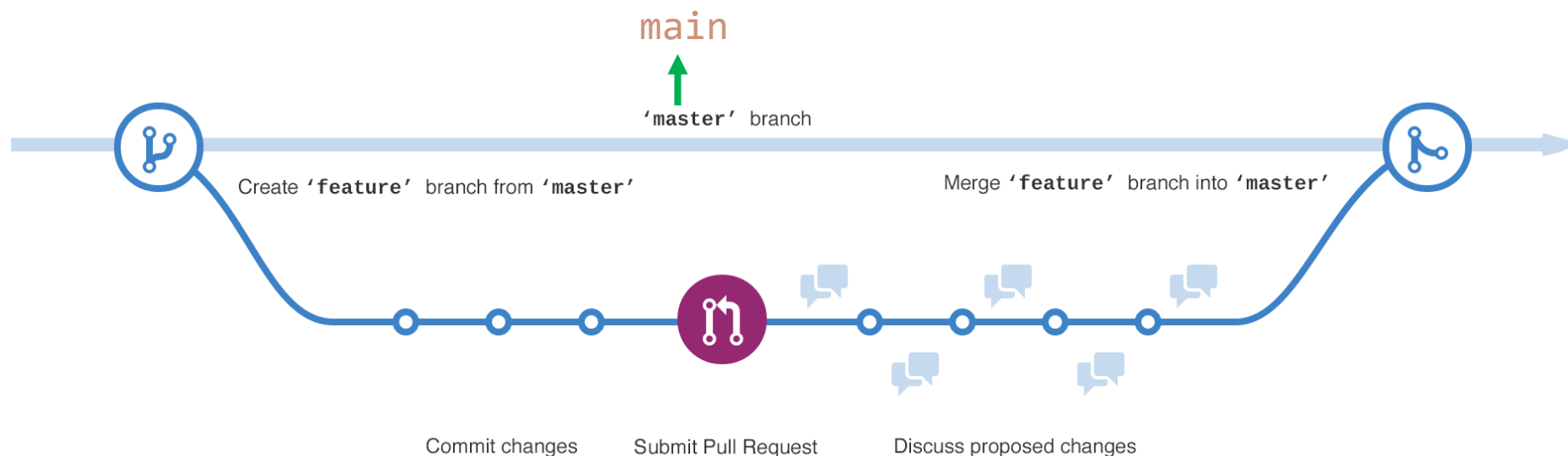
# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



### GitHub Flow



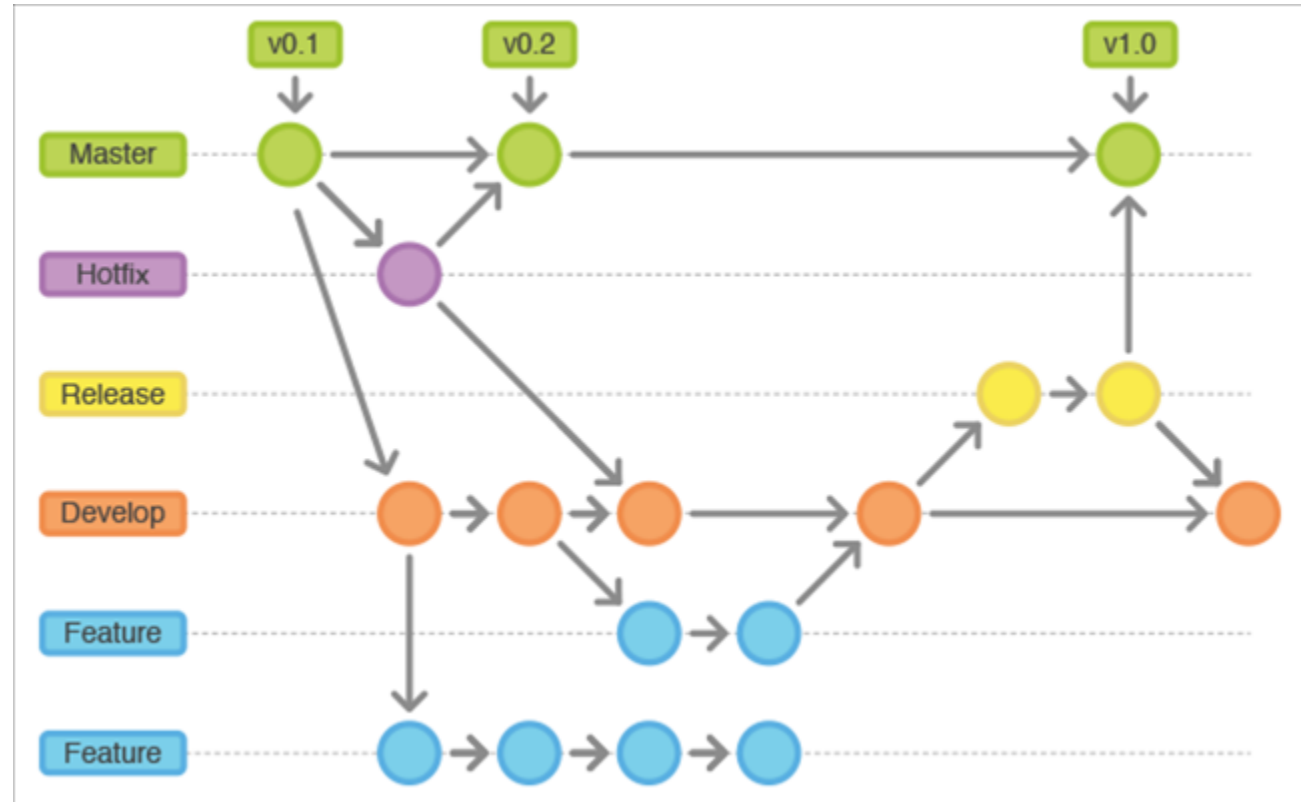
# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



### Git Flow





# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

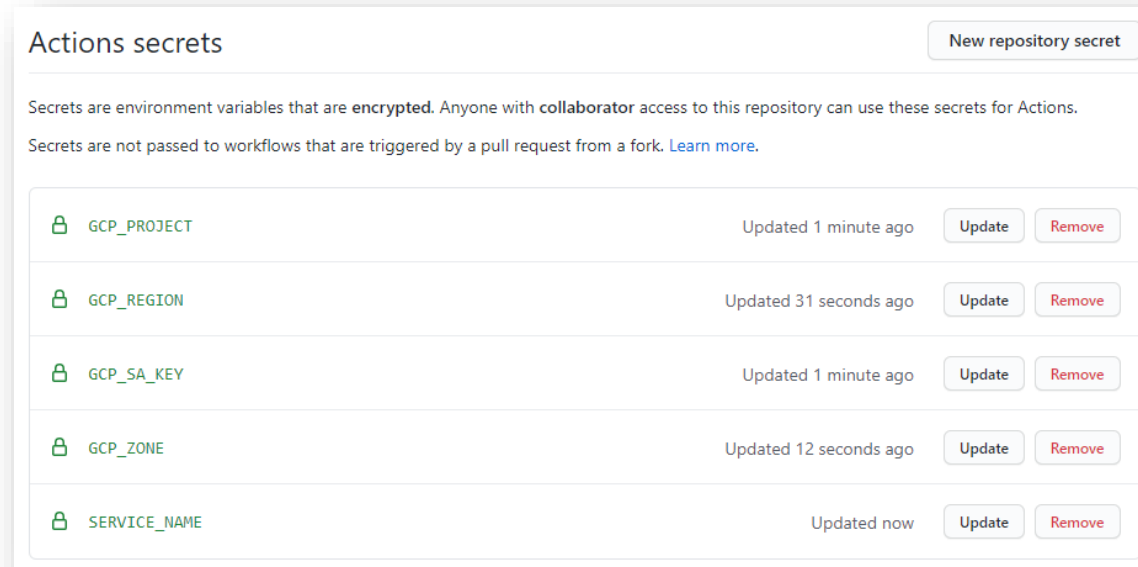
```
# Copyright 2021 Ignacio Asin (iasinDev)
#
# Licensed under the Apache License, Version 2.0 (the "License");
# you may not use this file except in compliance with the License.
# You may obtain a copy of the License at
#
#     http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable 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:
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
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 }

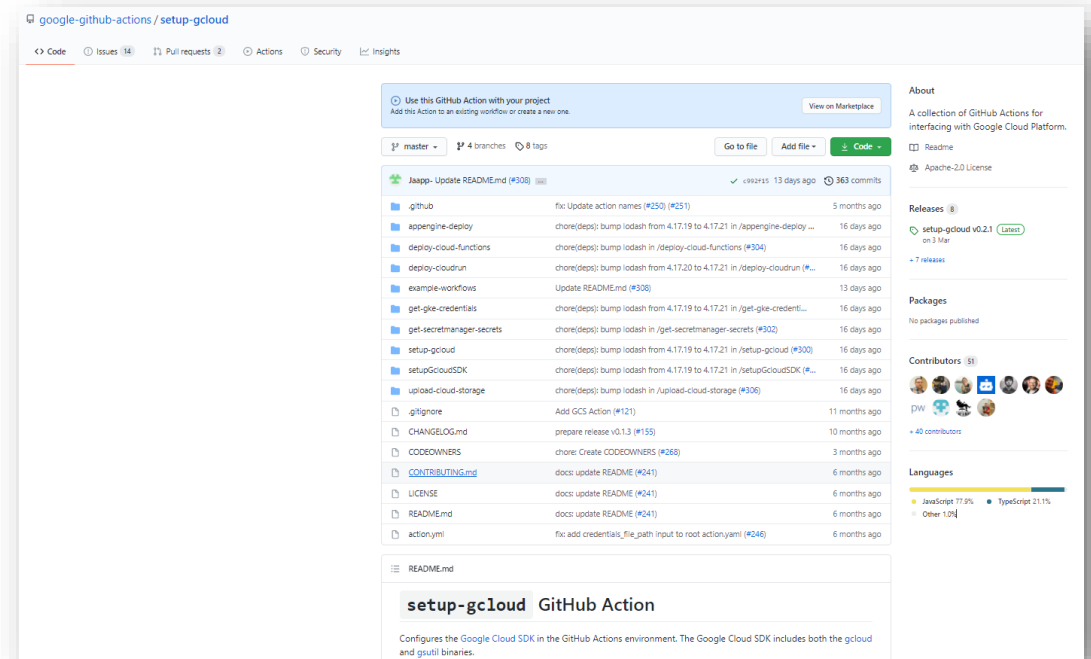
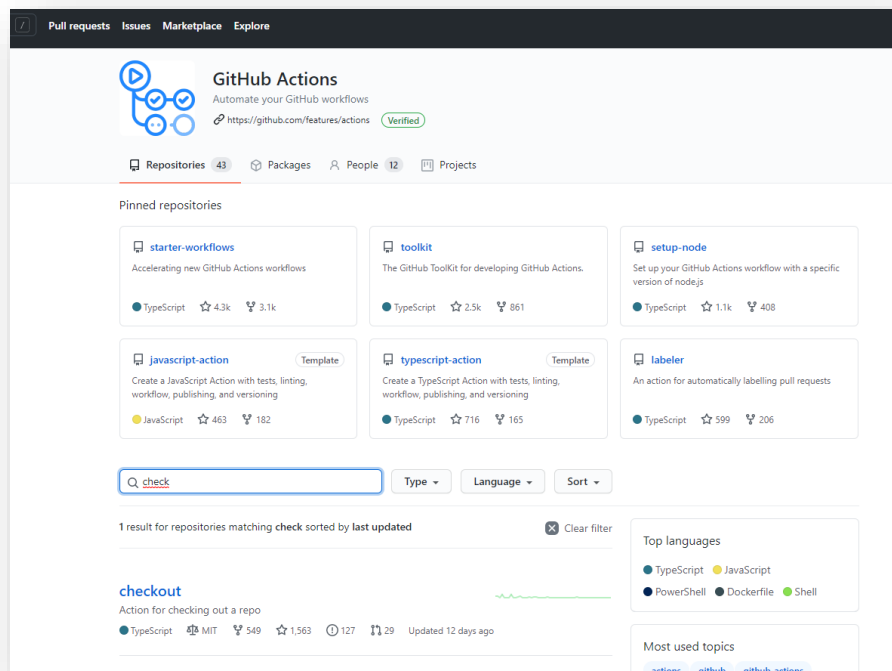
jobs:
```

# WHAT IS DEVOPS & DEVSECOPS?

## Continuous Integration (CI) & Continuous Delivery (CD)



### Examples

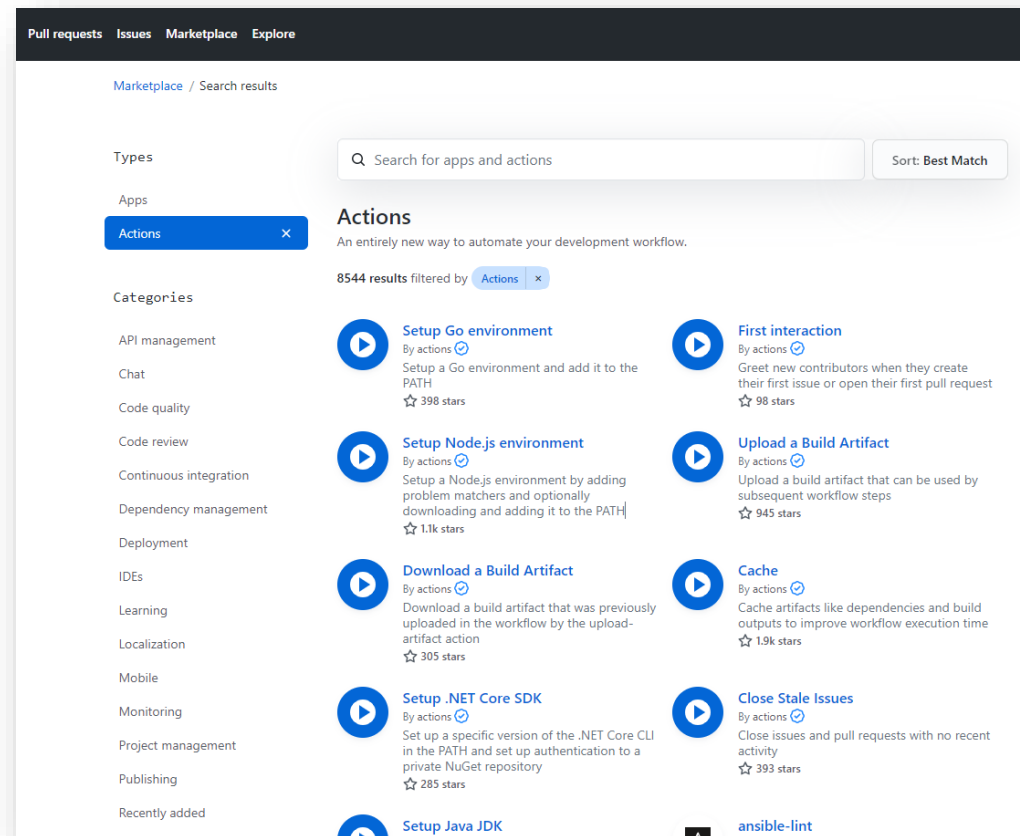


# WHAT IS DEVOPS & DEVSECOPS?

## Continuous Integration (CI) & Continuous Delivery (CD)



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



# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

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

  runs-on: ubuntu-latest

  steps:
    - name: Enable GitHub Actions
      uses: actions/checkout@v2

    - name: Setup Cloud SDK
      uses: google-github-actions/setup-gcloud@v0.2.0
      with:
        project_id: ${{ env.PROJECT_ID }}
        service_account_key: ${{ secrets.GCP_SA_KEY }}

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

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

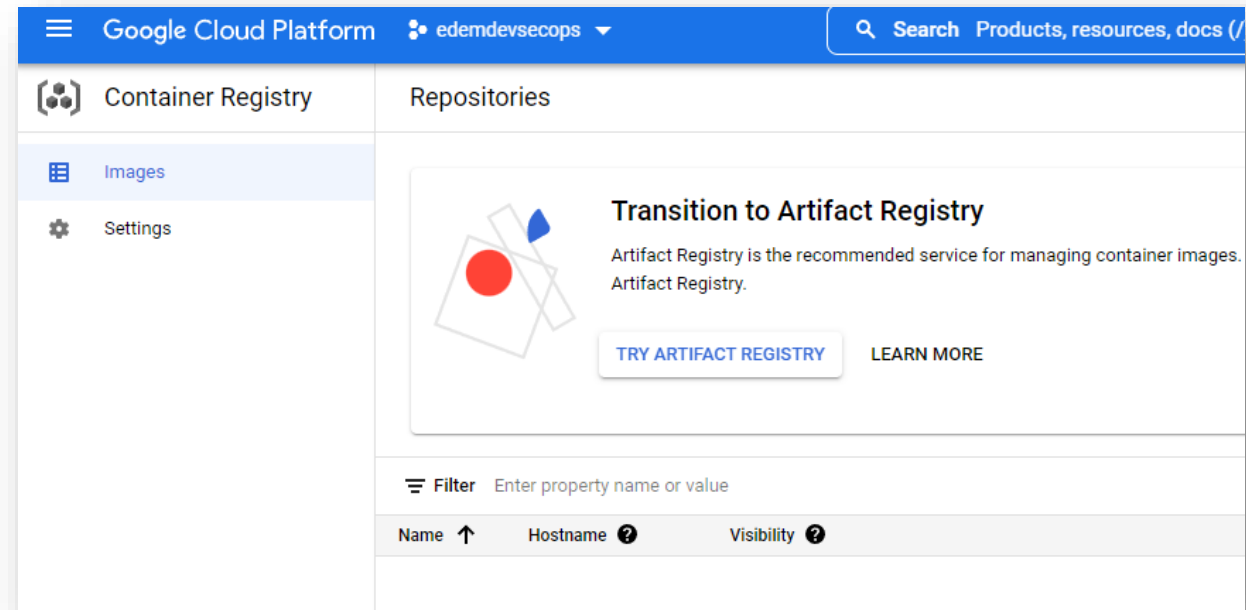
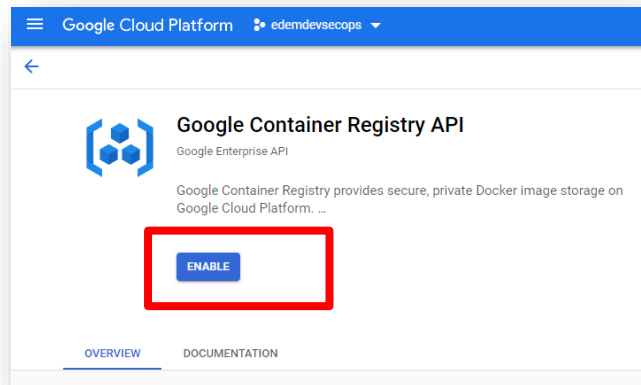
# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



Enable the Container Registry API



# WHAT IS DEVOPS & DEVSECOPS?

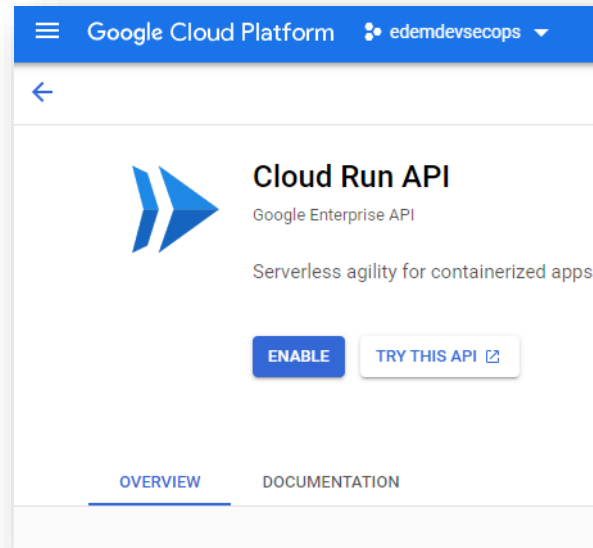


## Continuous Integration (CI) & Continuous Delivery (CD)



Enable also the Cloud Run API -

<https://console.cloud.google.com/apis/library/run.googleapis.com>





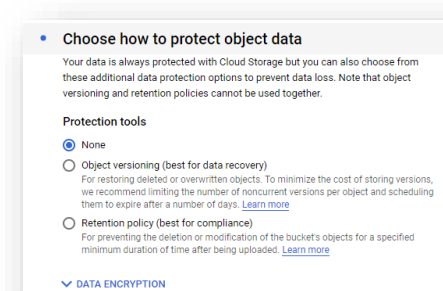
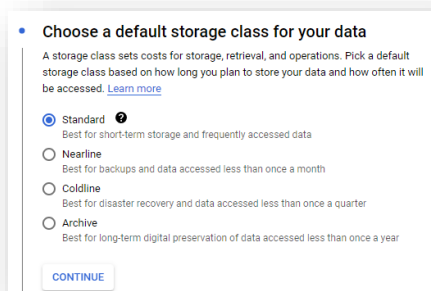
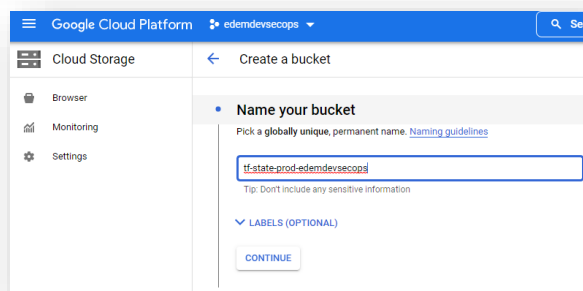
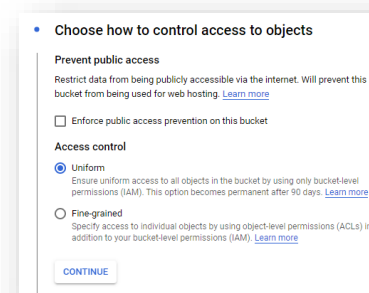
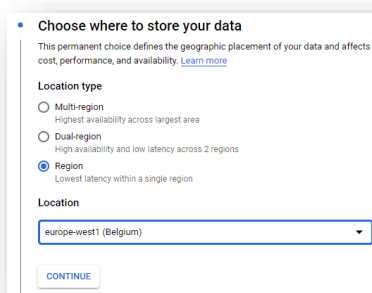
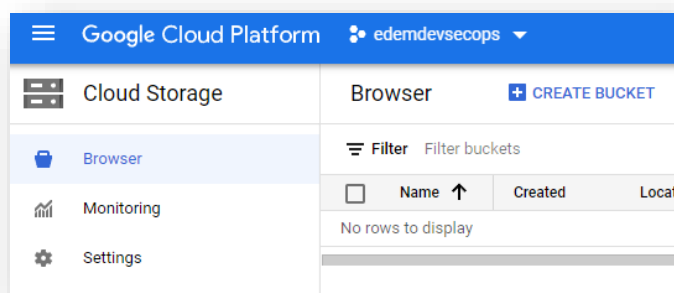
# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



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



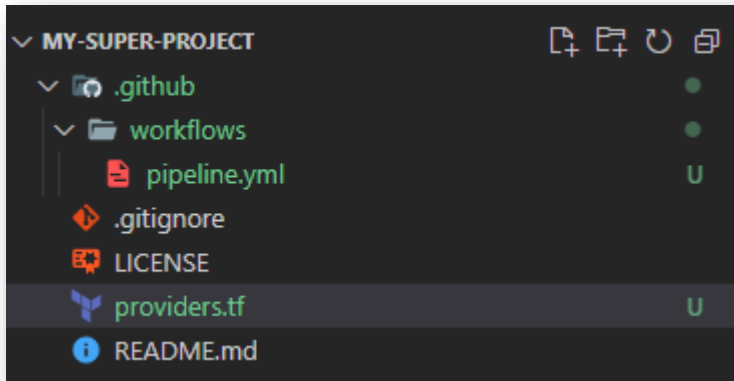
# WHAT IS DEVOPS & DEVSECOPS?



## 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**

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
- 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)



### tfsec

A static analysis security scanner for your Terraform code

tfsec is a developer-first security scanner for Terraform templates. It uses static analysis and deep integration with the official HCL parser to ensure security issues can be detected before your infrastructure changes take effect.

Designed to run locally and in your CI pipelines, developer-friendly output and fully documented checks mean detection and remediation can take place as quickly and efficiently as possible

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
check-terraform-security:

  needs: build-and-push-container-image

  runs-on: ubuntu-latest

  steps:

    - name: Enable GitHub Actions
      uses: actions/checkout@master

    - name: tfsec
      uses: tfsec/tfsec-sarif-action@master
      with:
        sarif_file: tfsec.sarif
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



The screenshot shows the HashiCorp Terraform website. The top navigation bar includes the HashiCorp logo, 'Browse Products', and 'About HashiCorp'. Below this is the 'Terraform' logo and a navigation menu with links to 'Overview', 'Editions', 'Registry', 'Tutorials', 'Docs', 'Community', and 'GitHub'. There are also buttons for 'Download CLI' and 'Terraform Cloud'. The main content area features a 'BLOG POST' link titled 'Announcing Controlled Remote State Access' and a large heading 'Write, Plan, Apply'. Below the heading, a paragraph describes Terraform as an open-source infrastructure as code software tool. A 'Get started' button is visible. On the right side, there is a terminal window showing the output of the 'terraform init' command, including the initialization of the backend and provider plugins.

HashiCorp Browse Products About HashiCorp

Terraform Overview Editions Registry Tutorials Docs Community GitHub Download CLI Terraform Cloud

BLOG POST Announcing Controlled Remote State Access →

### Write, Plan, Apply

Terraform is an open-source infrastructure as code software tool that provides a consistent CLI workflow to manage hundreds of cloud services. Terraform codifies cloud APIs into declarative configuration files.

Get started

```
→ iac-in-action git:(main) × terraform init

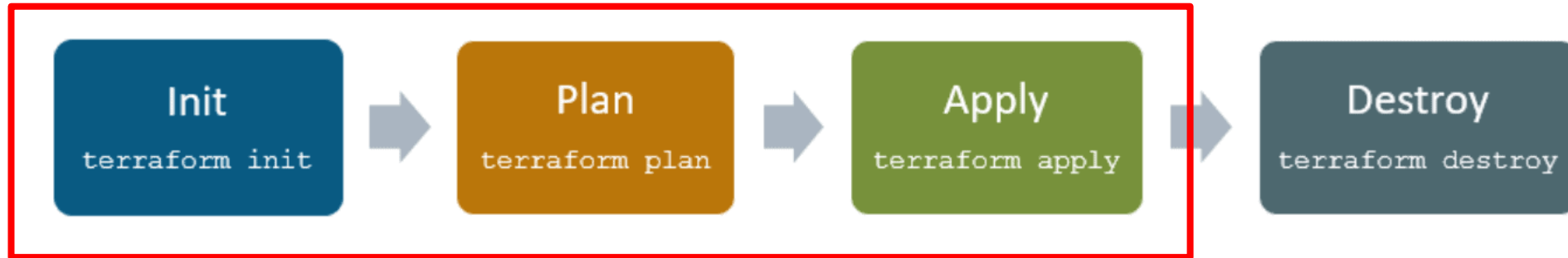
Initializing the backend...

Initializing provider plugins...
- Finding hashicorp/tls versions matching "~> 2.1"...
- Finding hashicorp/http versions matching "1.2.0"...
- Finding hashicorp/aws versions matching "~> 2.70"...
- Finding hashicorp/local versions matching "~> 1.4.0"...
- Installing hashicorp/tls v2.2.0...
- Installed hashicorp/tls v2.2.0 (signed by HashiCorp)
- Installing hashicorp/http v1.2.0...
```

## WHAT IS DEVOPS & DEVSECOPS?



Continuous Integration (CI) & Continuous Delivery (CD)



# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
deploy-in-cloud-run:

  needs: build-and-push-container-image

  runs-on: ubuntu-latest

  steps:

    - name: Enable GitHub Actions
      uses: actions/checkout@v2

      # 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@v1
```



# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
# 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 }
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/pipeline.yml`

```
# Generates an execution plan for Terraform
- name: Terraform Plan
  run: terraform plan
  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 }
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



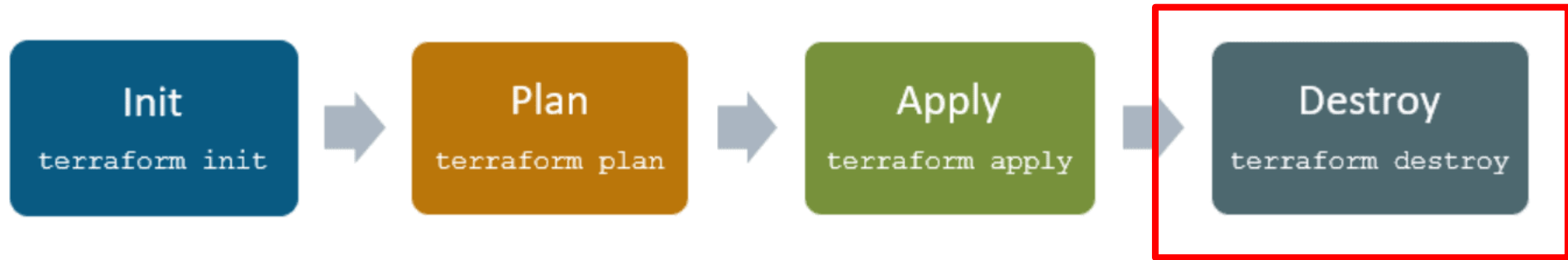
`.github/workflows/pipeline.yml`

```
# Apply the execution plan
- name: Terraform Apply
  if: github.ref == 'refs/heads/main'
  run: terraform apply -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 }
```

## WHAT IS DEVOPS & DEVSECOPS?



Continuous Integration (CI) & Continuous Delivery (CD)



# WHAT IS DEVOPS & DEVSECOPS?



## 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:
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/destroy.yml`

```
name: Destroy GCP resources
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)



`.github/workflows/destroy.yml`

```
destroy-terraform-resources:

  runs-on: ubuntu-latest

  steps:

    - name: Enable GitHub Actions
      uses: actions/checkout@v2

    # 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@v1
```

# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/destroy.yml`

```
# 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 }
```



# WHAT IS DEVOPS & DEVSECOPS?



## Continuous Integration (CI) & Continuous Delivery (CD)



`.github/workflows/destroy.yml`

```
# Destroy the resources
- name: Terraform Destroy
  run: terraform destroy -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 }
```

# WHAT IS DEVOPS & DEVSECOPS?

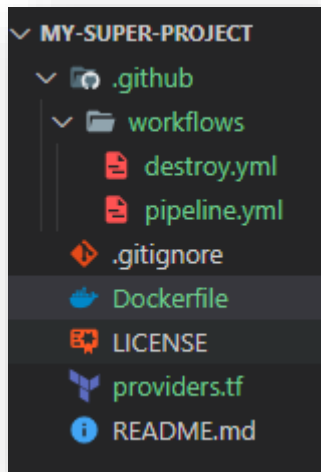


“Application as Code” ... “Platform as Code”



## 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
```

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

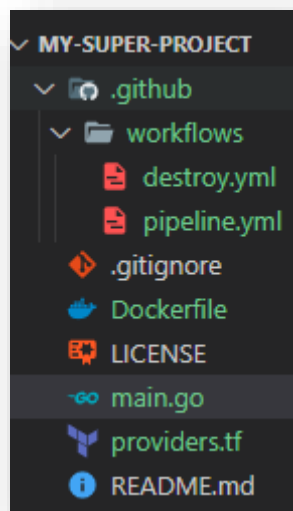
# WHAT IS DEVOPS & DEVSECOPS?



“Application as Code” ... “Platform as Code”



main.go



```
package main

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

func handler(w http.ResponseWriter, r *http.Request) {
    body := `

# Golang webapp running in a Docker container</h1>` 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)) }


```

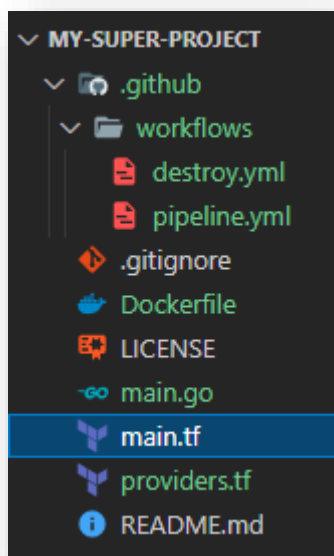
# WHAT IS DEVOPS & DEVSECOPS?



## Infrastructure as Code



main.tf



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

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

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

# WHAT IS DEVOPS & DEVSECOPS?



## Infrastructure as Code



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
}
```

# WHAT IS DEVOPS & DEVSECOPS?



## 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"  
    prefix = "terraform/state"  
  }  
}
```

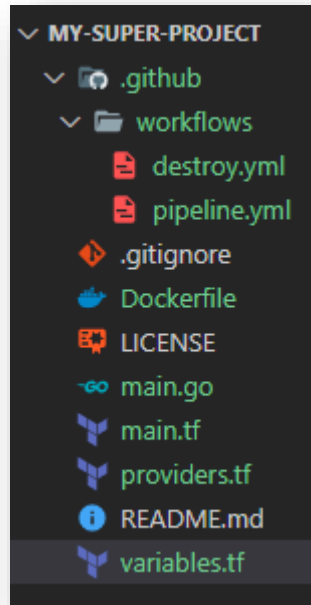
# WHAT IS DEVOPS & DEVSECOPS?



## Infrastructure as Code



variables.tf



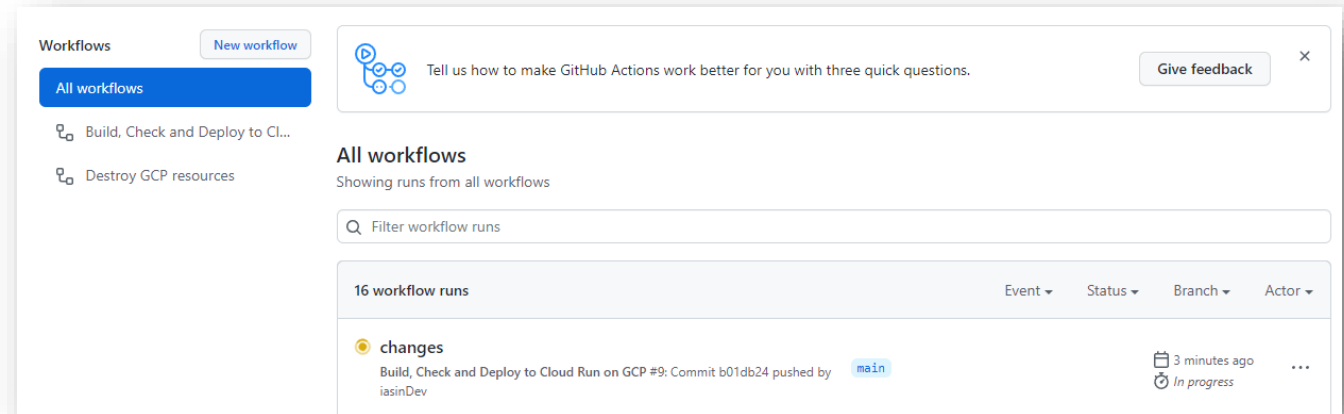
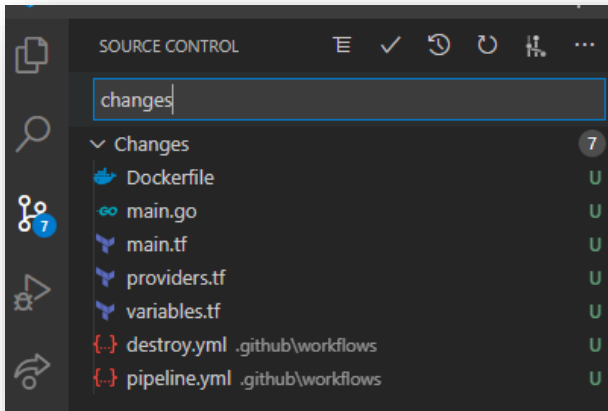
```
variable "project_id" {  
    type = string  
}  
  
variable "region" {  
    type = string  
}  
  
variable "zone" {  
    type = string  
}  
  
variable "service" {  
    type = string  
}
```

# WHAT IS DEVOPS & DEVSECOPS?



## Testing

commit & push your changes to GitHub





# WHAT IS DEVOPS & DEVSECOPS?



## Testing

See the pipeline execution

The screenshot shows the GitHub Actions interface for a repository named 'iasinDev / my-super-project'. The 'Actions' tab is selected, displaying a workflow named 'changes Build, Check and Deploy to Cloud Run on GCP #9'. The workflow status is 'Success', triggered by a push to the 'main' branch. The total duration is 3m 59s. The workflow consists of three jobs: 'build-and-push-container-image' (1m 6s), 'check-terraform-security' (9s), and 'deploy-in-cloud-run' (2m 4s). The 'pipeline.yml' file is shown with the following content:

```
on: push
```

```
jobs:
```

- build-and-push-containe... 1m 6s
- check-terraform-security 9s
- deploy-in-cloud-run 2m 4s

# WHAT IS DEVOPS & DEVSECOPS?



## Testing

See the pipeline execution

Google Cloud Platform

edemdevsecops

Search

cloud run

Cloud Run

Services

+

CREATE SERVICE

MANAGE CUSTOM DOMAINS

COPY

DELETE

Filter

Filter services

<input type="checkbox"/>	<input type="radio"/>	Name ↑	Req/sec ?	Region	Authentication ?	Ingress ?	Last deployed	Deployed by
<input type="checkbox"/>	<input checked="" type="radio"/>	edemdevsecops	0	europa-west1	Allow unauthenticated	All	1 minute ago	edemdevops@edemdevsecops.iam.gserviceaccount.com

# WHAT IS DEVOPS & DEVSECOPS?



## Testing

See the pipeline execution

A screenshot of the Google Cloud Platform (GCP) console interface. The top navigation bar is blue and contains the text 'Google Cloud Platform', a dropdown menu with 'edemdevsecops', and a search bar with the text 'cloud run'. Below this, a breadcrumb trail shows 'Cloud Run' followed by 'Service details'. To the right of the breadcrumb are two links: 'EDIT &amp; DEPLOY NEW REVISION' and 'SET UP CONTINUOUS DEPLOYMENT'. The main content area shows a green checkmark icon, the service name 'edemdevsecops', and the region 'Region: europe-west1'. A red rectangular box highlights the 'URL: https://edemdevsecops-dyl5yf6hya-ew.a.run.app' and its associated copy and info icons. Below this is a horizontal tab bar with 'METRICS' (selected), 'LOGS', 'REVISIONS', 'TRIGGERS', 'DETAILS', 'YAML', and 'PERMISSIONS'. The 'METRICS' section displays a message: 'No errors found during this interval.' At the bottom, there are two panels: 'Request count' with a 'Create alerting policy' link and 'Request latencies'.

# WHAT IS DEVOPS & DEVSECOPS?



## Testing

See the pipeline execution

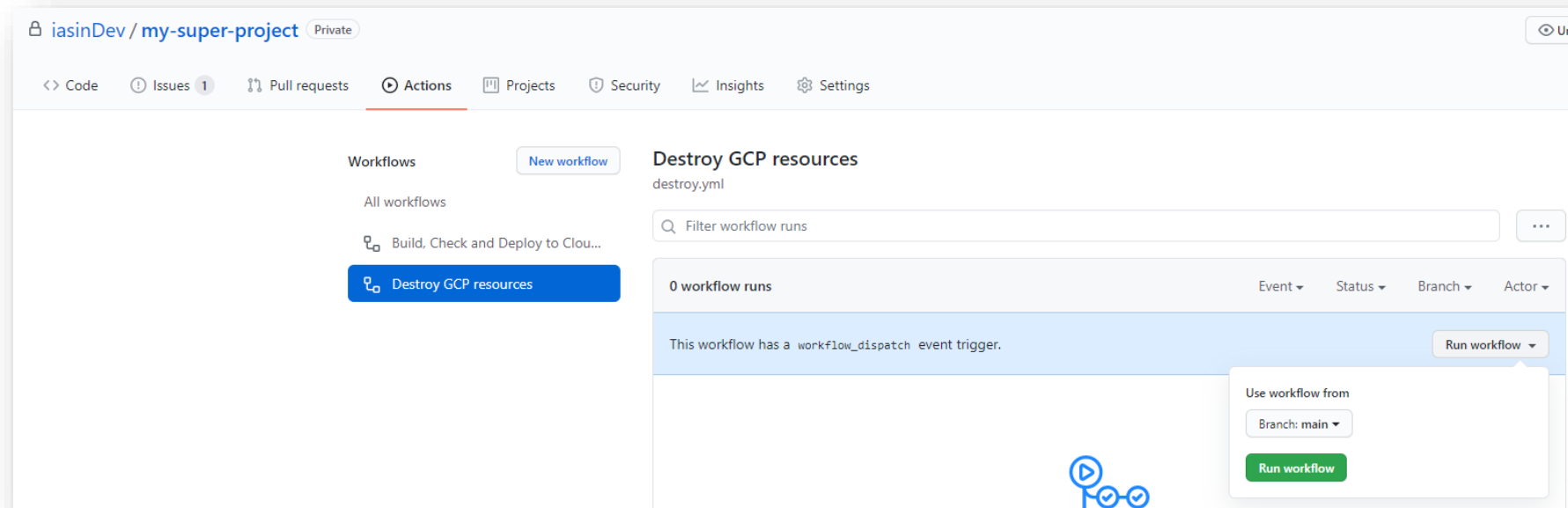
**Golang webapp running in a Docker container for the EDEM DevSecOps course**

# WHAT IS DEVOPS & DEVSECOPS?



## Destroy All

Manually launch the destroy pipeline



# WHAT IS DEVOPS & DEVSECOPS?



## Destroy All

Manually launch the destroy pipeline

The screenshot shows the GitHub Actions interface for a repository named 'iasinDev / my-super-project'. The 'Actions' tab is selected, displaying a workflow titled 'Destroy GCP resources' with a green checkmark icon. Below the title, the workflow is categorized as 'Destroy GCP resources #1'. A 'Summary' section on the left lists the job 'destroy-terraform-resources' with a green checkmark. The main area shows a table with details for the workflow run:

Manually triggered 38 seconds ago	Status	Total duration	Billable time	Artifacts
iasinDev d6cbc60	Success	35s	16s	—

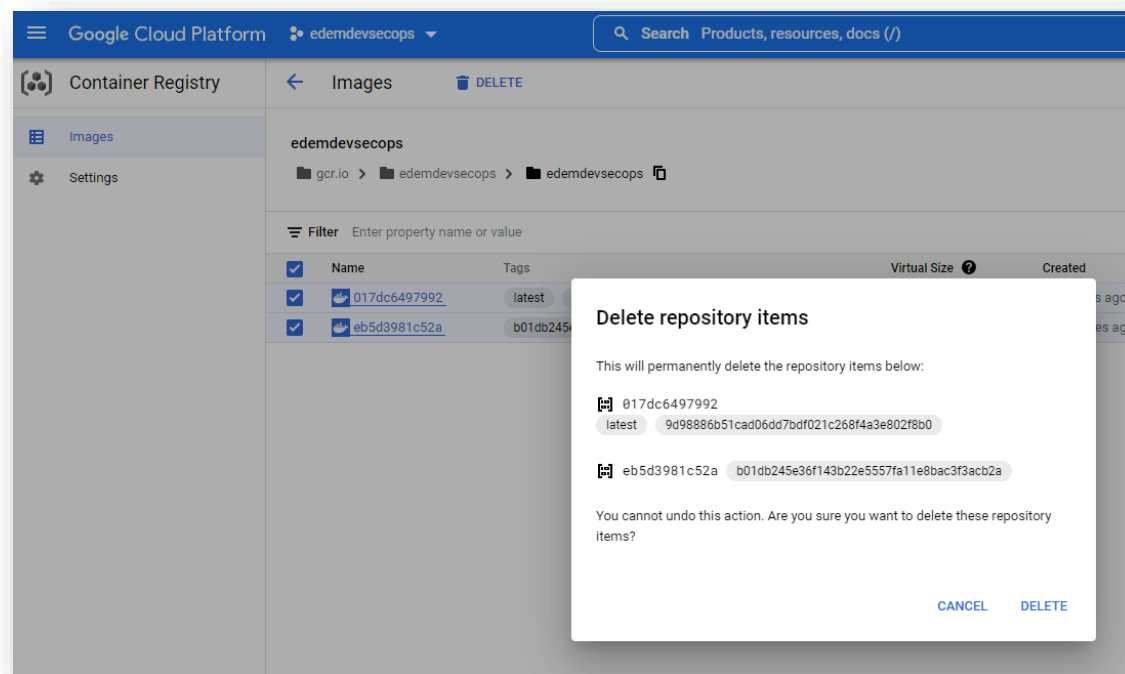
Below the table, the workflow file 'destroy.yml' is shown, triggered by 'on: workflow\_dispatch'. A summary box at the bottom indicates the job 'destroy-terraform-resources' completed in 16s.

# WHAT IS DEVOPS & DEVSECOPS?



## Destroy All

Delete all Container Registry images

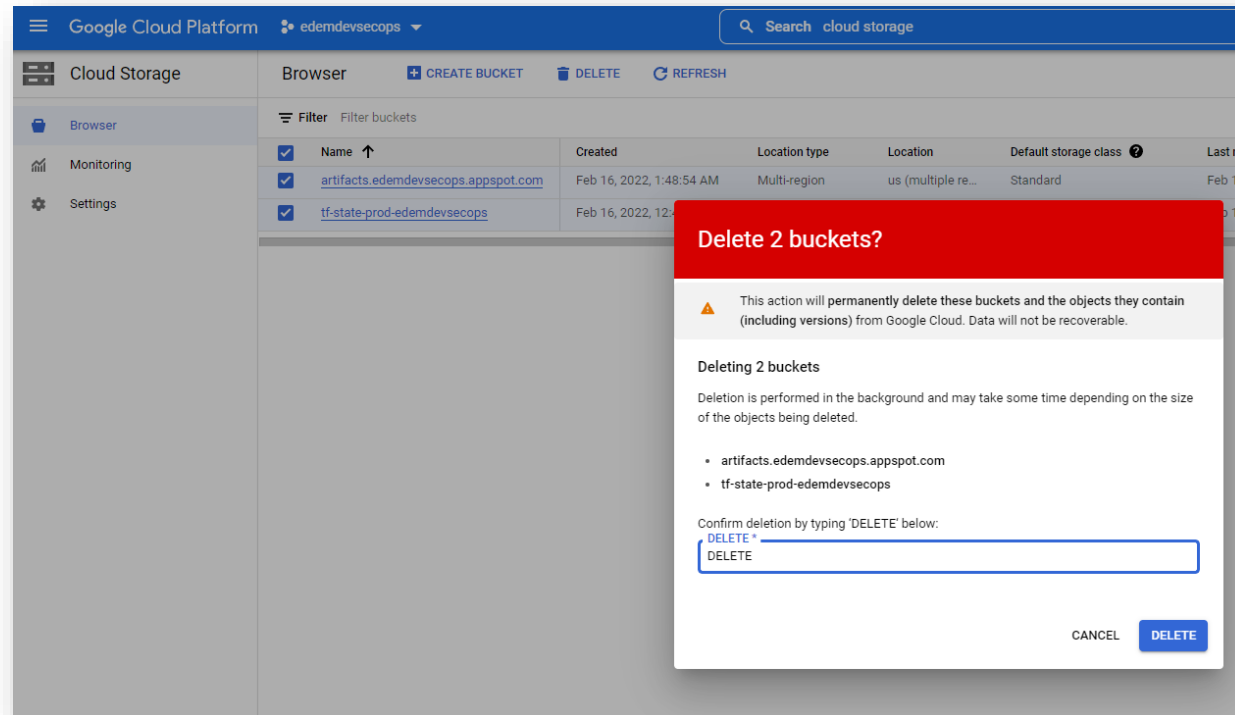


# WHAT IS DEVOPS & DEVSECOPS?



## Destroy All

Delete the buckets created

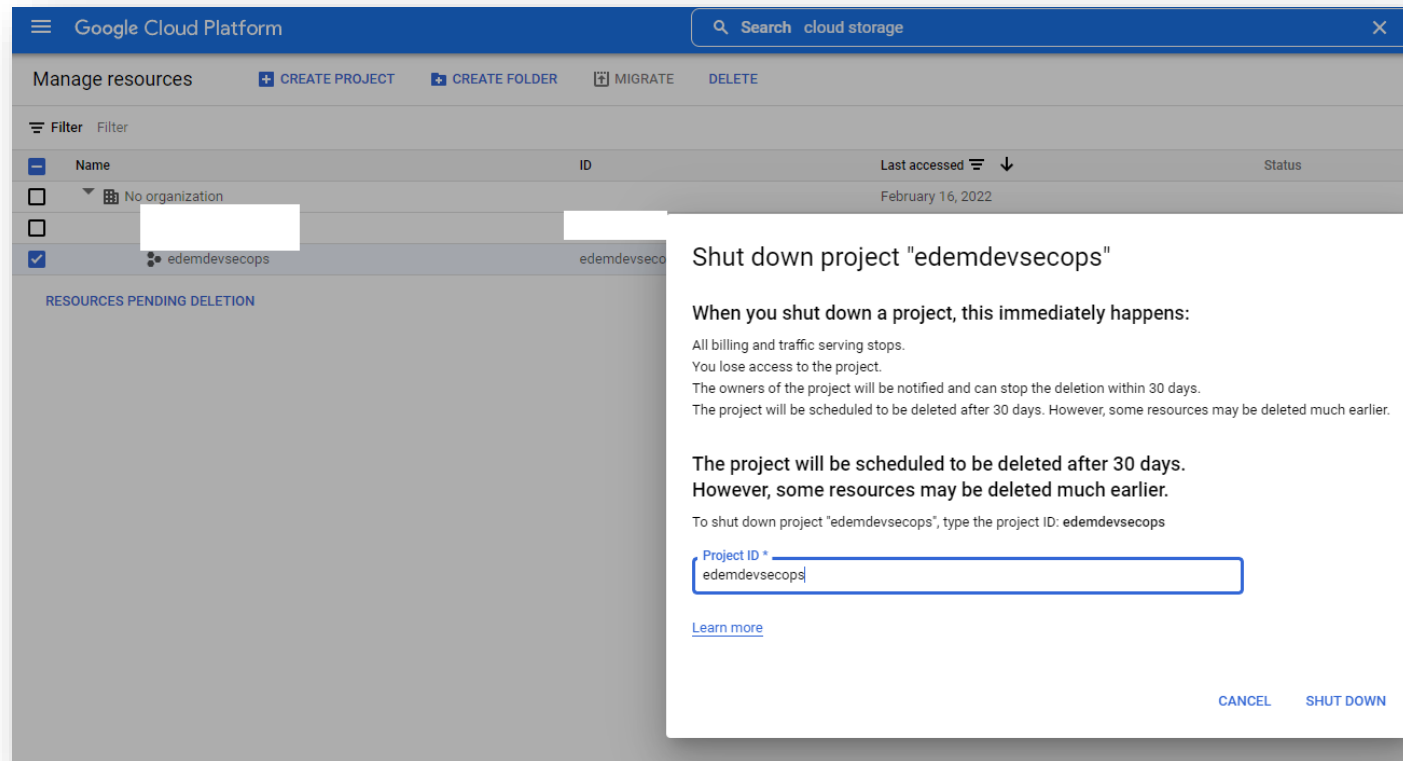




# WHAT IS DEVOPS & DEVSECOPS?

## Destroy All

Delete the GCP project in [resource manager](#)



# WHAT IS DEVOPS & DEVSECOPS?



## Destroy All

Delete the GitHub Project

Are you absolutely sure?

×

Unexpected bad things will happen if you don't read this!

This action **cannot** be undone. This will permanently delete the **iasinDev/my-super-project** repository, wiki, issues, comments, packages, secrets, workflow runs, and remove all collaborator associations.

This will not change your billing plan. If you want to downgrade, you can do so in your Billing Settings.

Please type **iasinDev/my-super-project** to confirm.

iasinDev/my-super-project

I understand the consequences, delete this repository

# AGENDA

**QUESTIONS??**