

# DAW - Lab 5

Roger Sancho Martí

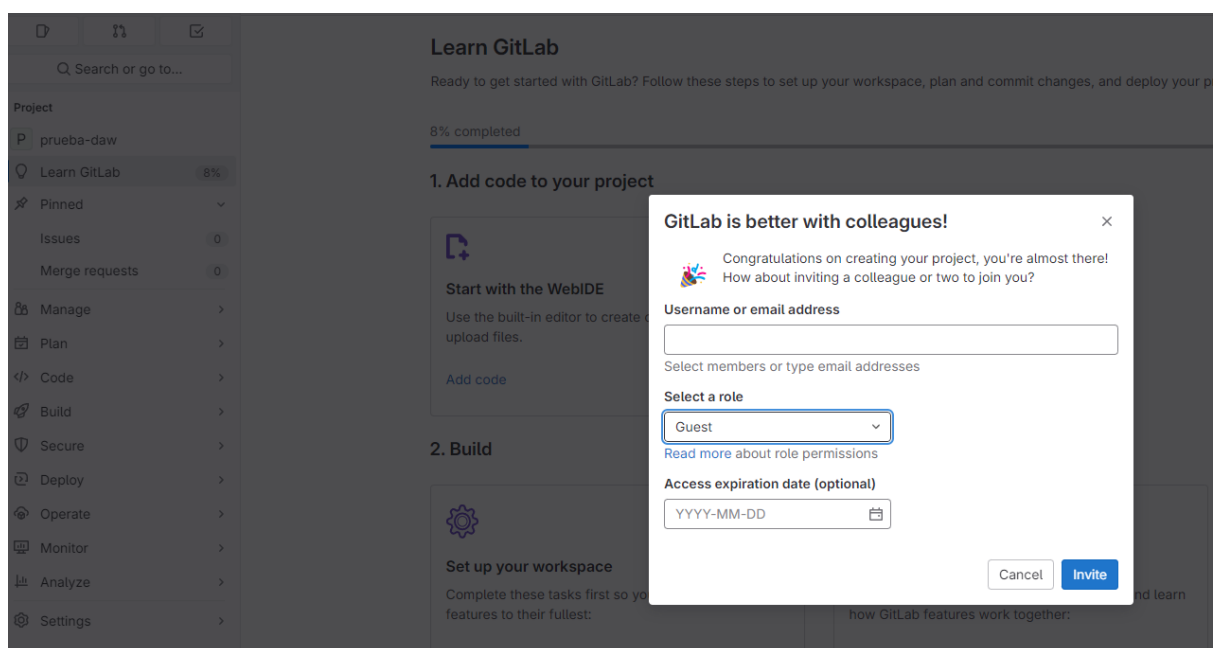
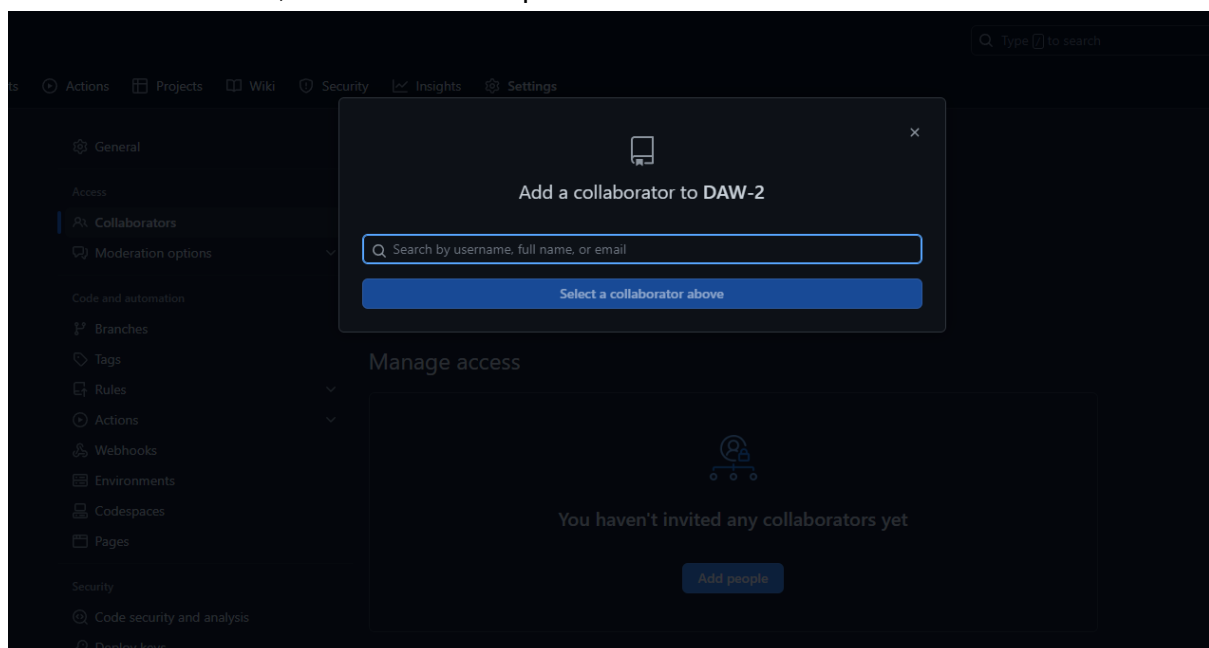
## Exercise 2

I signed up in gitlab and created a public repository:

<https://gitlab.com/rogersanchom/prueba-daw>

Differences:

GitLab encourages you to invite collaborators from the creation start of a project and lets you select a role for them, and an access expiration date for them.
















From the start, it shows GitLab is an all-in-one DevOps platform that includes not only Git repository hosting, but also project management, IDE integration (integrated development environment), CI/CD pipelines, issue tracking, and more.





On the other hand, GitHub has historically been focused more on code hosting and open-source collaboration but has recently added features such as GitHub actions, CI/CD workflows, an online IDE integration such as VSCode, and project management tools.

The image displays two web interfaces side-by-side. The top interface is the GitLab 'Learn GitLab' onboarding page. It features a sidebar on the left with a search bar and a list of project management tools: Project, Pinned, Issues, Merge requests, Manage, Plan, Code, Build, Secure, Deploy, Operate, Monitor, Analyze, and Settings. The main content area is titled 'Learn GitLab' and includes a progress bar showing '8% completed'. It lists two steps: '1. Add code to your project' with a 'Start with the WebIDE' button, and '2. Build' with three sub-sections: 'Set up your workspace' (listing tasks like 'Invite your colleagues', 'Create a repository', 'Set up your first project's CI/CD', 'Start a free trial of GitLab Ultimate', 'Add code owners', and 'Enable require merge approvals'), 'Plan and execute' (listing 'Create an issue' and 'Submit a merge request (MR)'), and 'Deploy' (listing 'Scan dependencies for licenses', 'Scan dependencies for vulnerabilities', and 'Analyze your application for vulnerabilities with DAST').

The bottom interface is the GitHub repository page for 'prueba-daw'. It shows the repository name, a 'Public' badge, and a 'Pin' button. Below this are two cards: 'Set up GitHub Copilot' and 'Add collaborators to this repository'. A 'Quick setup' section provides instructions for cloning the repository using 'Set up in Desktop', 'HTTPS', or 'SSH'. It also includes a 'Quick setup — if you've done this kind of thing before' section with a command line example for creating a new repository and a section for pushing an existing repository from the command line.

Project

-  prueba-daw
-  Learn GitLab 8%
-  Pinned >
- Issues 0
- Merge requests 0
-  Manage >
-  Plan >
-  Code >
-  Build >
-  Secure >
-  Deploy >
-  Operate >
-  Monitor >
-  Analyze >
-  Settings >

-  Build >
  - Pipelines
  - Jobs
  - Pipeline editor
  - Pipeline schedules
  - Artifacts
-  Secure >
  - Security capabilities
  - Audit events
  - Security configuration
-  Deploy >
  - Releases
  - Feature flags
  - Package Registry
  - Container Registry
  - Pages
-  Operate >
  - Environments
  - Kubernetes clusters
  - Terraform states
  - Terraform modules

# Get started with GitHub Actions

Build, test, and deploy your code. Make code reviews, branch management, and issue triaging work the way you want. Select a workflow to get started.

Skip this and [set up a workflow yourself](#) →

## Suggested for this repository

### Simple workflow

By GitHub

Start with a file with the minimum necessary structure.

[Configure](#)

## Deployment

[View all](#)

### Deploy Node.js to Azure Web App

By Microsoft Azure

Build a Node.js project and deploy it to an Azure Web App.

[Configure](#)

Deployment

### Deploy to Amazon ECS

By Amazon Web Services

Deploy a container to an Amazon ECS service powered by AWS Fargate or Amazon EC2.

[Configure](#)

Deployment

### Build and Deploy to GKE

By Google Cloud

Build a docker container, publish it to Google Container Registry, and deploy to GKE.

[Configure](#)

Deployment

### Terraform

By HashiCorp

Set up Terraform CLI in your GitHub Actions workflow.

[Configure](#)

Deployment

### Deploy to Alibaba Cloud ACK

By Alibaba Cloud

Deploy a container to Alibaba Cloud Container Service for Kubernetes (ACK).

[Configure](#)

Deployment

### Deploy to IBM Cloud Kubernetes Service

By IBM

Build a docker container, publish it to IBM Cloud Container Registry, and deploy to IBM Cloud Kubernetes Service.

[Configure](#)

Deployment

### Tencent Kubernetes Engine

By Tencent Cloud

This workflow will build a docker container, publish and deploy it to Tencent Kubernetes Engine (TKT).

[Configure](#)

Deployment

### OpenShift

By Red Hat

Build a Docker-based project and deploy it to OpenShift.

[Configure](#)

Deployment

## Security

[View all](#)

### CodeQL Analysis

By GitHub

Security analysis from GitHub for C, C++, C#, Go, Java, JavaScript, TypeScript, Python, Ruby, Kotlin and Swift developers.

[Configure](#)

Code scanning

### tfsec

By tfsec

A static analysis security scanner for your terraform code. Discover problems with your infrastructure before hackers do.

[Configure](#)

Code scanning

### njsscan

By NodeJSscan

nodejscan is a static security code scanner that finds insecure code patterns in your Node.js applications.

[Configure](#)

Code scanning

### SonarQube

By Sonar

Static analysis of code for vulnerability detection, covering 26+ languages. Start cleaning your code in minutes!

[Configure](#)

Code scanning

## Continuous integration

[View all](#)

### Gulp

By GitHub Actions

Build a Node.js project with npm and gulp.

[Configure](#)

JavaScript

### Publish Node.js Package to GitHub Packages

By GitHub Actions

Publishes a Node.js package to GitHub Packages.

[Configure](#)

JavaScript

### Python application

By GitHub Actions

Create and test a Python application.

[Configure](#)

Python

### Publish Python Package

By GitHub Actions

Publish a Python Package to PyPI on release.

[Configure](#)

Python

## Automation

### Stale

By GitHub Actions

Checks for stale issues and pull requests.

[Configure](#)

Automation

### Greetings

By GitHub Actions

Greets users who are first time contributors to the repo.

[Configure](#)

Automation

### Labeler

By GitHub Actions

Labels pull requests based on the files changed.

[Configure](#)

Automation

### Manual workflow

By GitHub Actions

Simple workflow that is manually triggered.

[Configure](#)

Automation

## Exercise 3

Things i found out about GitHub by reading the articles:

- In an article from 3 months ago, it says free private repositories can only have a maximum of three collaborators, while github documentation says they are unlimited.
- Some articles say free plans only provide 500 MB of storage and 1GB for Git LFS (Large File System) users, but again, the official documentation says that you can increase this amount to 2GB thanks to Git LFS.
- Also, additional storage and bandwidth are available for purchase in a single data pack, which costs \$5/month. It will provide you with 50 GB of bandwidth and storage for one month.