

GUIA PASO 1

En este paso vamos a obtener 4 valores de las variables secretas que vamos a crear en nuestro Git.

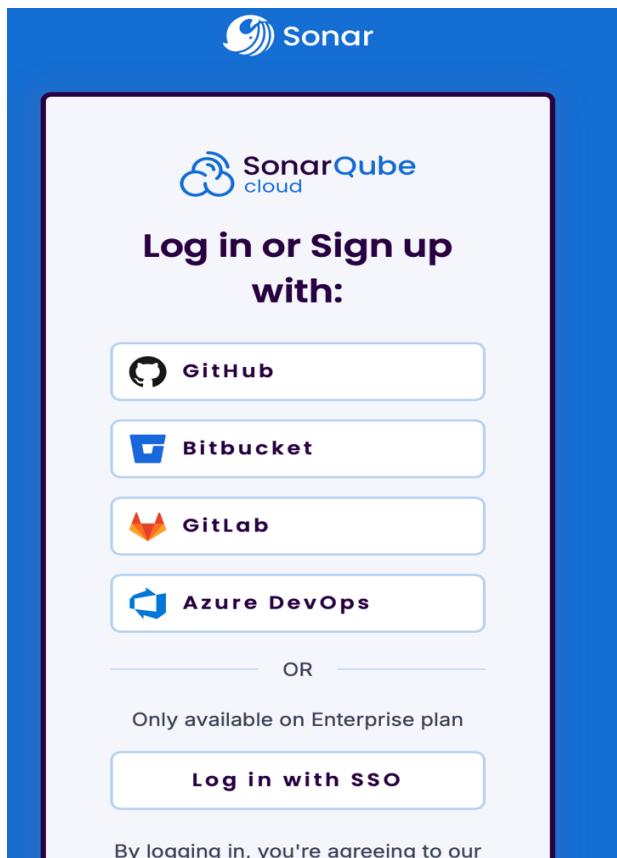
SONAR_HOST_URL

SONAR_ORGANIZATION

SONAR_PROJECT_KEY

SONAR_TOKEN

1. Accedemos al Sonar iniciando sesión desde la cuenta donde vamos a tener nuestro repositorio. Iniciamos sesión con uno de nuestros GIT.



2. Si no tenemos el proyecto creado todavía, tenemos que dar en el '+' que nos aparece en la esquina superior derecha y elegimos la opción "analyze new project". Dentro de esa opción elegimos nuestro repositorio que queremos analizar. Aquí vemos ya el valor de nuestra variable "SONAR_ORGANIZATION".

SonarQube cloud

My Projects My Issues Explore

Analyze projects

Select repositories from one of your GitHub organization.

Organization

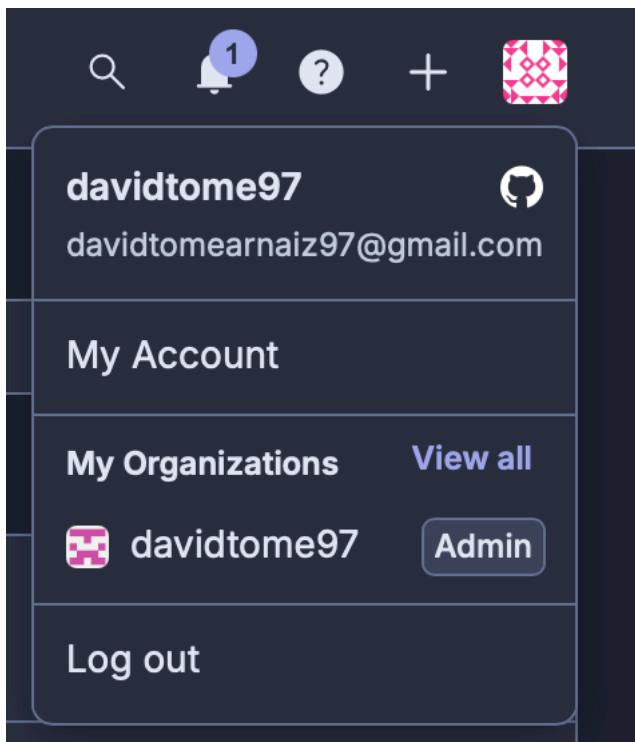
davidtome97

Import another organization

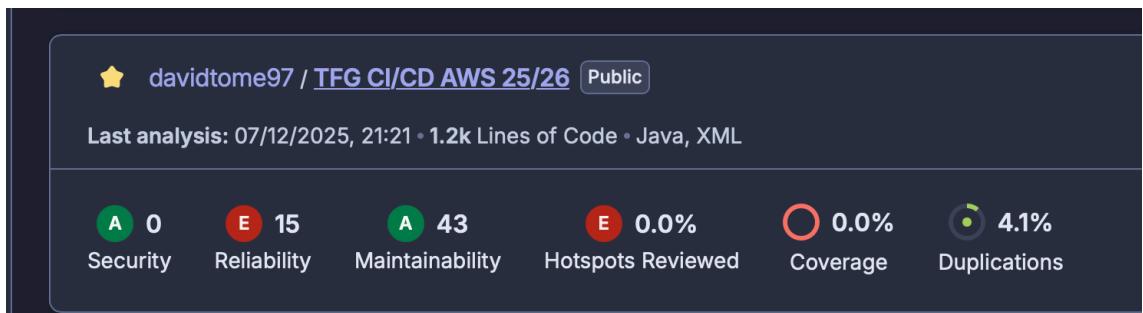
Select all on this page Search for repositories

<input checked="" type="checkbox"/>  tfg-cicd-aws-2526	✓ Already imported
<input type="checkbox"/>  refactoring-fowler-example	
<input type="checkbox"/>  Practica3_PCTR_DependenciasEstados	
<input checked="" type="checkbox"/>  ramas	✓ Already imported
<input type="checkbox"/>  ActividadRepositorio	
<input type="checkbox"/>  ejemploArray	

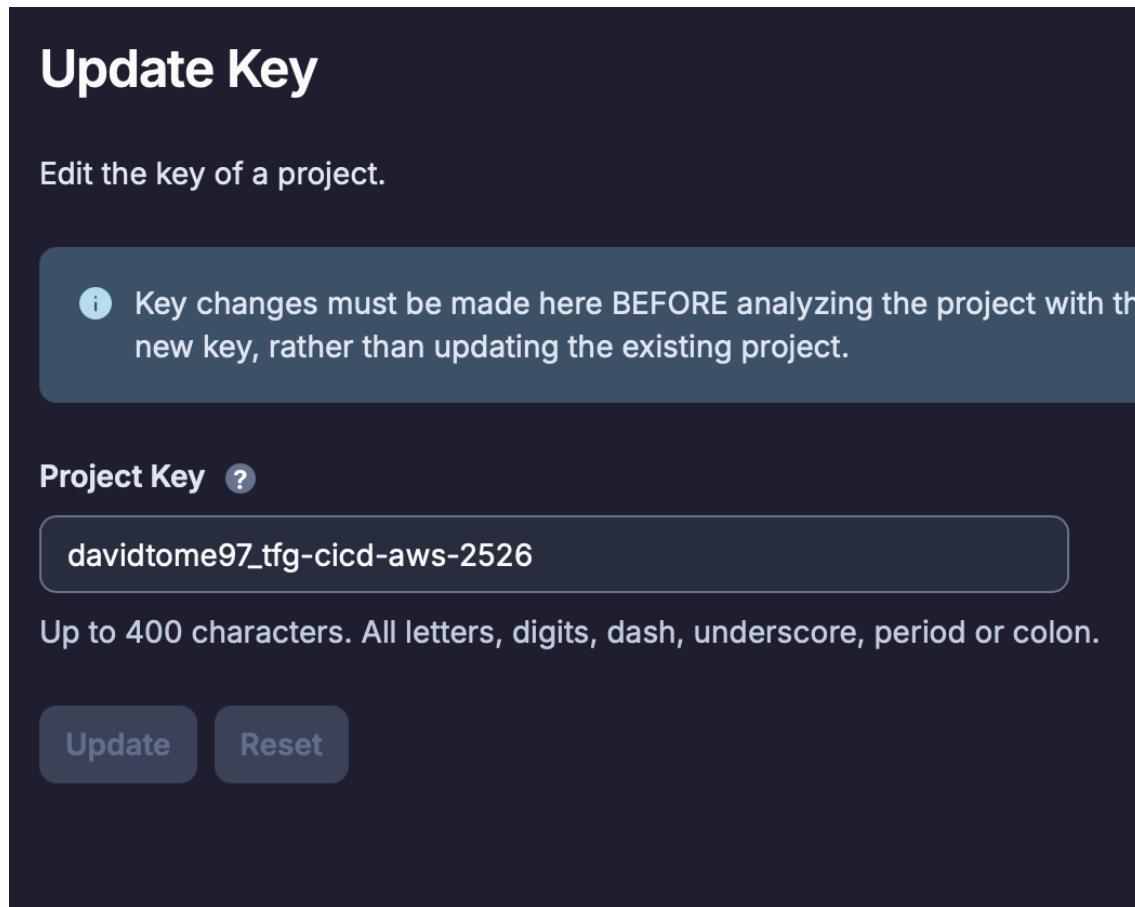
3. En la esquina superior derecha, si damos sobre nuestra foto de perfil, podremos ver una opción que pone: “My Organizations”, ahí también podemos ver el valor de nuestra variable “SONAR_ORGANIZATION”. También lo vamos a poder ver en el inicio de cada proyecto que tenemos



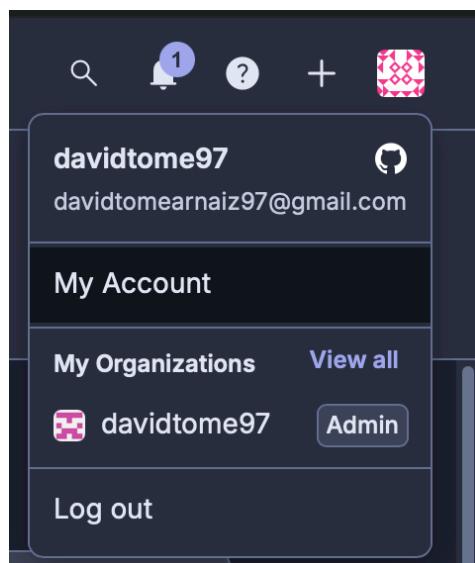
4. Una vez que ya tenemos nuestro proyecto y nuestra organización, vamos a ver el valor de nuestra variable “SONAR_PROJECT_KEY”. Para ello damos en nuestro nombre del proyecto, en este caso “TFG CI/CD AWS 25/26”.



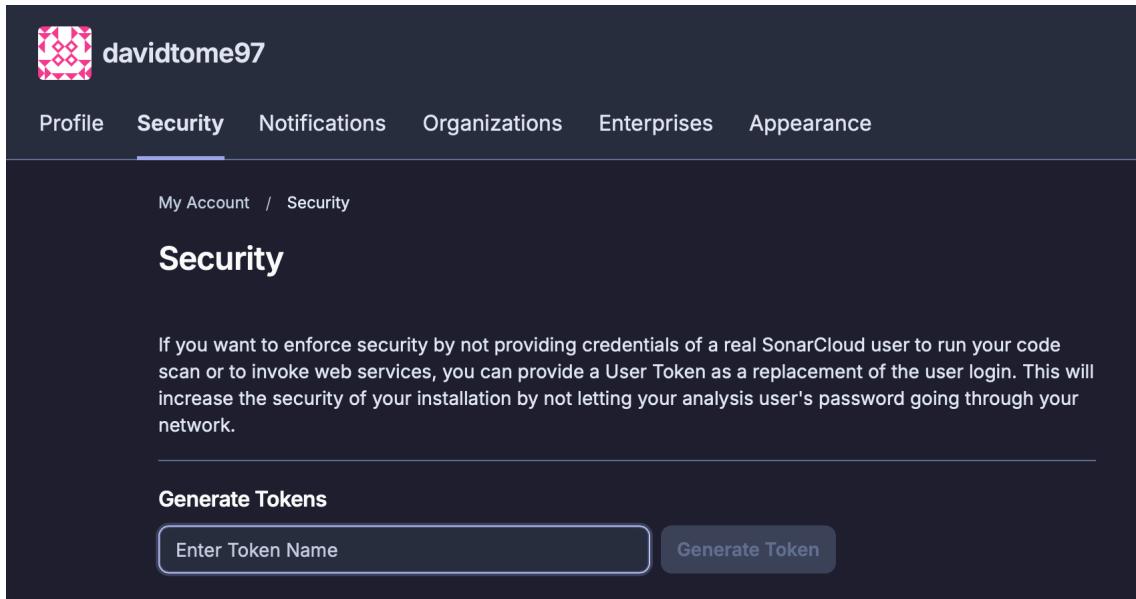
5. Dentro de nuestro proyecto, en la esquina inferior izquierda pulsamos en Administration → Update Key. Se nos abrirá una ventana donde aparece nuestro valor de “SONAR_PROJECT_KEY”.



6. Ya solo nos falta crear el token de Sonar. Para ello, desde el inicio de sonar. Si damos a nuestra foto de perfil (parte superior derecha) → My Account



7. Se nos abre una ventana con diferentes pestañas, pulsamos en la de “Security”. Escribimos un nombre en el campo “generate tokens” y damos al botón de generar. Copia ese token que te va a dar ¡¡OJO, ESE TOKEN SOLO LO VAS A VER UNA VEZ!! Si se te olvida, tendrás que generar otro nuevo. Aquí ya tendríamos el valor de nuestra variable “SONAR_TOKEN”



8. Por defecto, el valor de nuestra variable “SONAR_HOST_URL” es “<https://sonarcloud.io>”

Con esto dariamos como finalizado la obtención de nuestros 4 valores que vamos a crear ahora en Git:

SONAR_ORGANIZATION (paso 2-3)

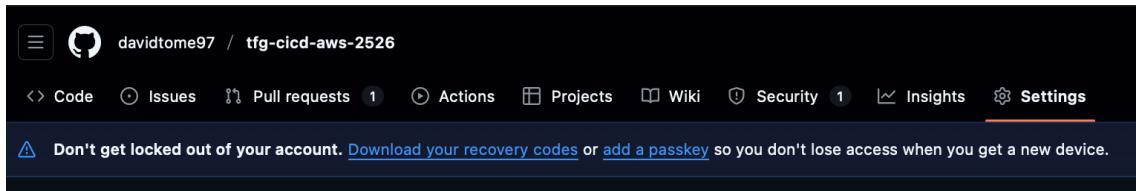
SONAR_PROJECT_KEY (paso 5)

SONAR_TOKEN (paso 7)

SONAR_HOST_URL (paso 8)

¿Cómo creamos nuestras variables secretas en nuestro github?

1. Iniciamos sesión en nuestro github → accedemos a nuestro repositorio con el que queremos trabajar → settings.



2. En el menú lateral, vamos hasta la opción de security → secrets and variables → actions. Ahí en la pestaña de “secrets”, damos al botón “New repository secret”.

A screenshot of the GitHub repository settings page, specifically the 'Actions' section under 'Secrets and variables'. The left sidebar shows various repository settings like General, Access, Collaborators, and Security. The 'Secrets and variables' section is selected. The main content area shows the 'Actions' section with the 'Secrets' tab selected. It displays a table of existing repository secrets and a green 'New repository secret' button. The table columns are 'Name' and 'Last updated'. The secrets listed are: AWS_ACCESS_KEY_ID (2 months ago), AWS_ECR_URL (2 weeks ago), AWS_REGION (2 months ago), and AWS_SECRET_ACCESS_KEY (2 months ago).

Name	Last updated
AWS_ACCESS_KEY_ID	2 months ago
AWS_ECR_URL	2 weeks ago
AWS_REGION	2 months ago
AWS_SECRET_ACCESS_KEY	2 months ago

3. Escribimos el nombre de las variables que te he proporcionado y el valor de cada una de ellas que hemos obtenido antes y damos a “add secret”.
¡¡MUY IMPORTANTE, EL NOMBRE TIENE QUE SER EL MISMO QUE TE HE PROPORCIONADO AL PRINCIPIO DEL MENÚ!!

Actions secrets / New secret

Name *

Secret *

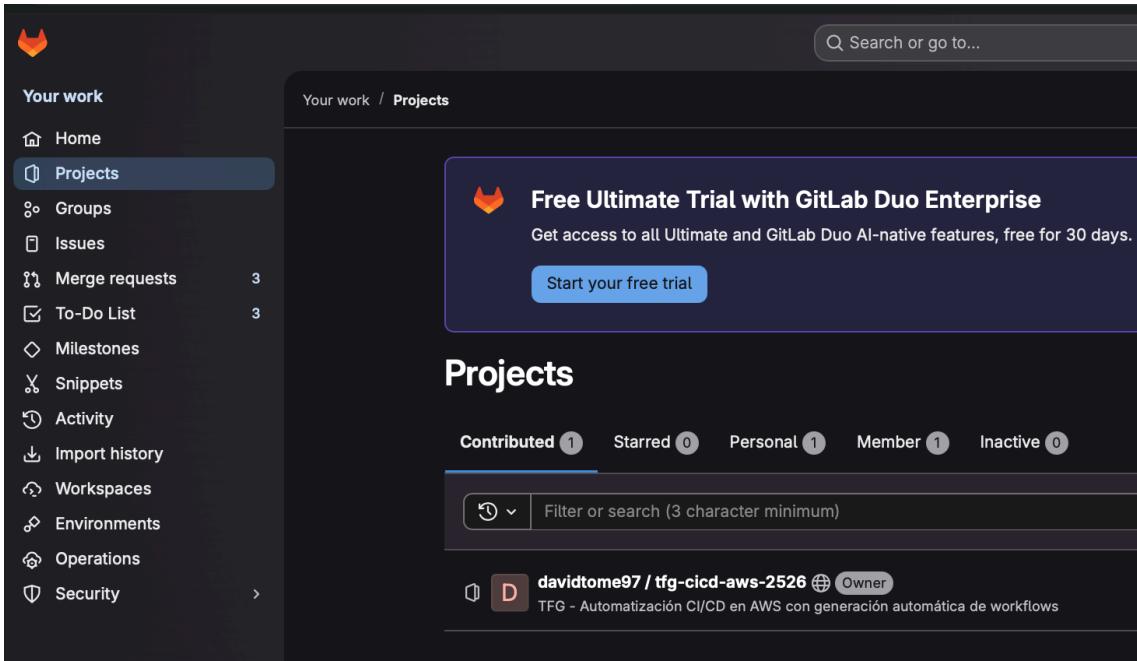
Add secret

Una vez que creamos todas las variables de este paso hay que hacer un commit para que coja los cambios realizados y no nos falle el siguiente paso:

```
git commit --allow-empty -m "creación de variables"  
git push
```

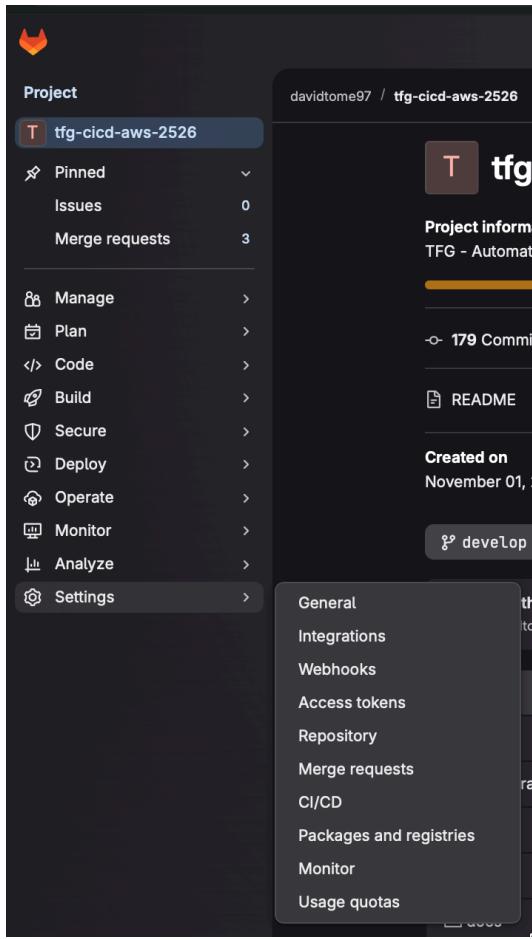
¿Cómo creamos nuestras variables secretas en nuestro gitlab?

1. Iniciamos sesión en nuestro gitlab → projects → nuestro proyecto.



The screenshot shows the GitLab web interface. On the left, a dark sidebar titled 'Your work' contains a list of navigation items: Home, Projects (which is selected and highlighted in blue), Groups, Issues, Merge requests (3), To-Do List (3), Milestones, Snippets, Activity, Import history, Workspaces, Environments, Operations, and Security. At the bottom of the sidebar is a '...' button. The main content area is titled 'Your work / Projects'. It features a promotional banner for 'Free Ultimate Trial with GitLab Duo Enterprise' with a 'Start your free trial' button. Below the banner, the word 'Projects' is displayed in a large, bold font. Underneath, there are filters for 'Contributed 1', 'Starred 0', 'Personal 1', 'Member 1', and 'Inactive 0'. A search bar with a placeholder 'Filter or search (3 character minimum)' is also present. At the bottom of the main content, a project card is shown for 'davidtome97 / tfg-cicd-aws-2526' (Owner), which is described as 'TFG - Automatización CI/CD en AWS con generación automática de workflows'.

2. En nuestro menú lateral, seleccionamos Settings → CI/CD. Se nos abrirá una nueva ventana.



3. Desplegamos la opción “Variables” y vemos una opción que pone “add variable”.

A job artifact is an archive of files and directories saved by a job when it finishes.

Variables

Variables store information that you can use in job scripts. Each project can define a maximum of 8000 variables. [Learn more](#).

Minimum role to use pipeline variables

Select the minimum role that is allowed to run a new pipeline with pipeline variables. [What are pipeline variables?](#)

No one allowed
Pipeline variables cannot be used.

Owner

Maintainer

Developer

Save changes

Access protected resources in merge request pipelines

Make protected CI/CD variables and runners available in merge request pipelines. Protected resources will only be available in merge request pipelines if both the source and target branches of the merge request are protected. [Learn more](#).

Allow merge request pipelines to access protected variables and runners

Save changes

Display manually-defined pipeline variables

Display all manually-defined variables in the pipeline details page after running a pipeline manually. [Learn more](#).

Display pipeline variables

All manually-defined CI/CD variables and their values are visible to maintainers, which is a security risk if including credentials or other secrets in variables. Do not enable this feature if variables could contain sensitive data. Developers can only view manually-defined variables in their own manual pipelines.

Save changes

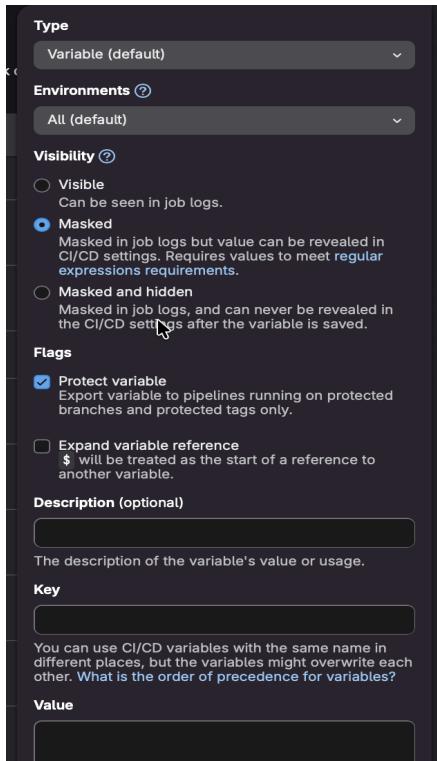
Project variables

Variables can be accidentally exposed in a job log, or maliciously sent to a third party server. The masked variable feature can help reduce the risk of accidentally exposing variable values, but is not a guaranteed method to prevent malicious users from accessing variables. [How can I make my variables more secure?](#)

CICD Variables	Reveal values	Add variable
20		

4. Elegimos el tipo que queremos, la visibility que queremos para esta variable, en “key” ponemos el nombre de la variable y en “value” su valor que hemos obtenido anteriormente.

¡¡MUY IMPORTANTE, EL NOMBRE TIENE QUE SER EL MISMO QUE TE HE PROPORCIONADO AL PRINCIPIO DEL MENÚ!!



Una vez que creamos todas las variables de este paso hay que hacer un commit para que coja los cambios realizados y no nos falle el siguiente paso:

```
git commit --allow-empty -m "creación de variables CI/CD"  
git push
```