[https://github.com/IntiOps/turno\_tarde](https://github.com/IntiOps/turno_tarde" \t "_blank)

subir archivos a un bucket que se va a crear  
documentar como se va a crear un rol para aws

github action con s3

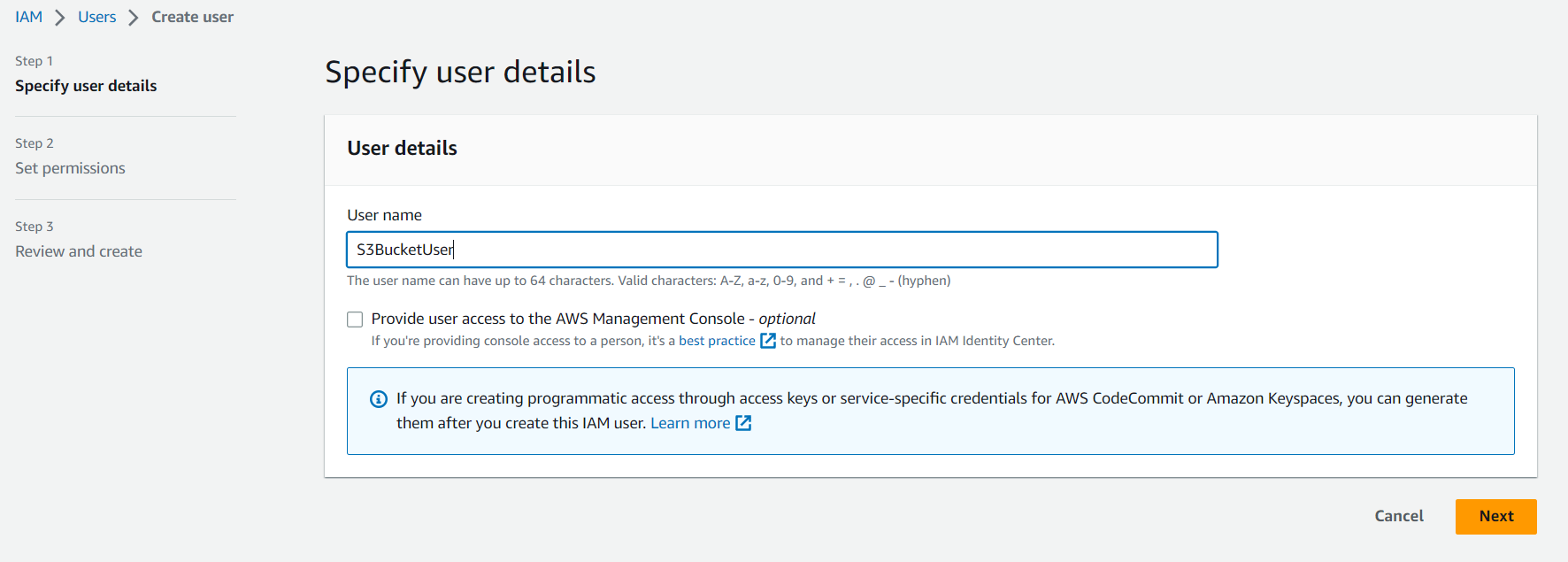
user access y secret-key

mejor manera de como subir la informacion al s3 con github actions

entornos

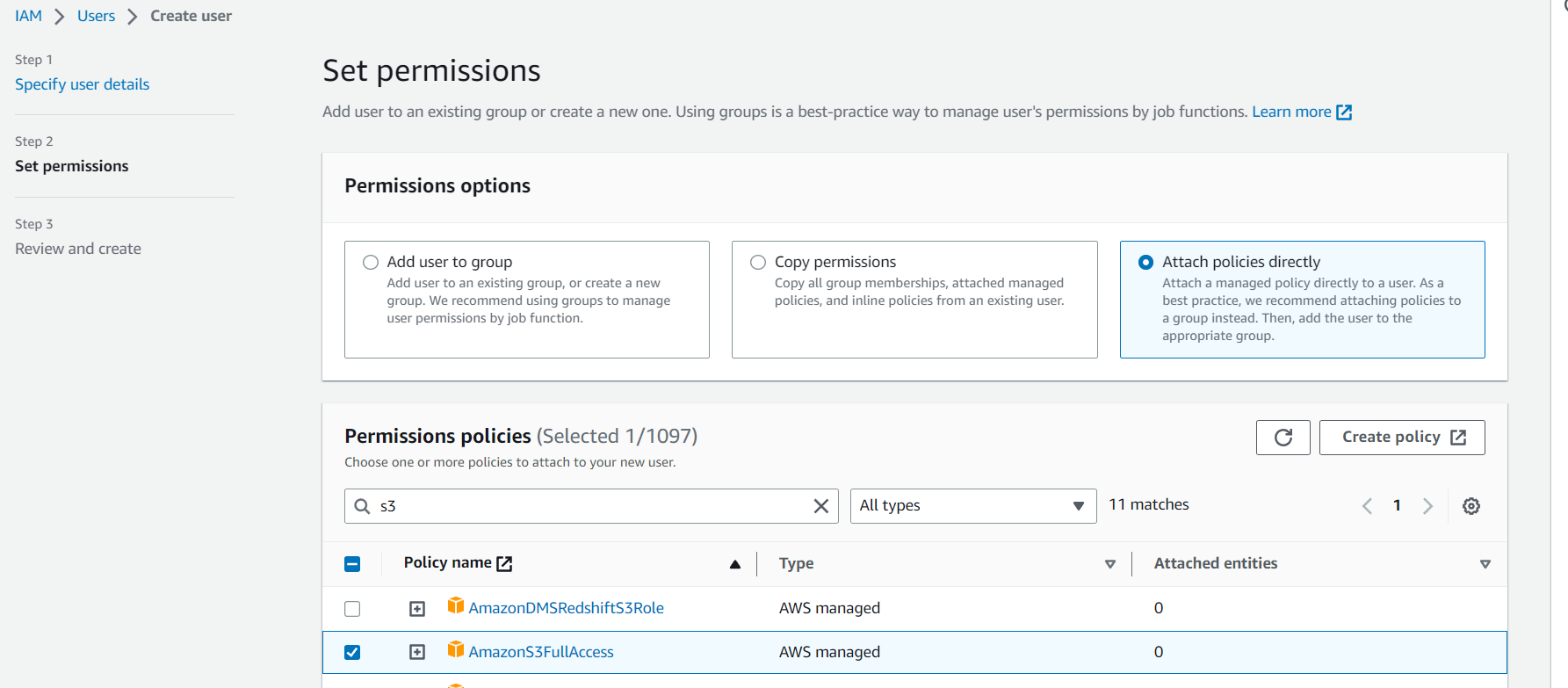
como funciona yaml

integracion de github actions con s3

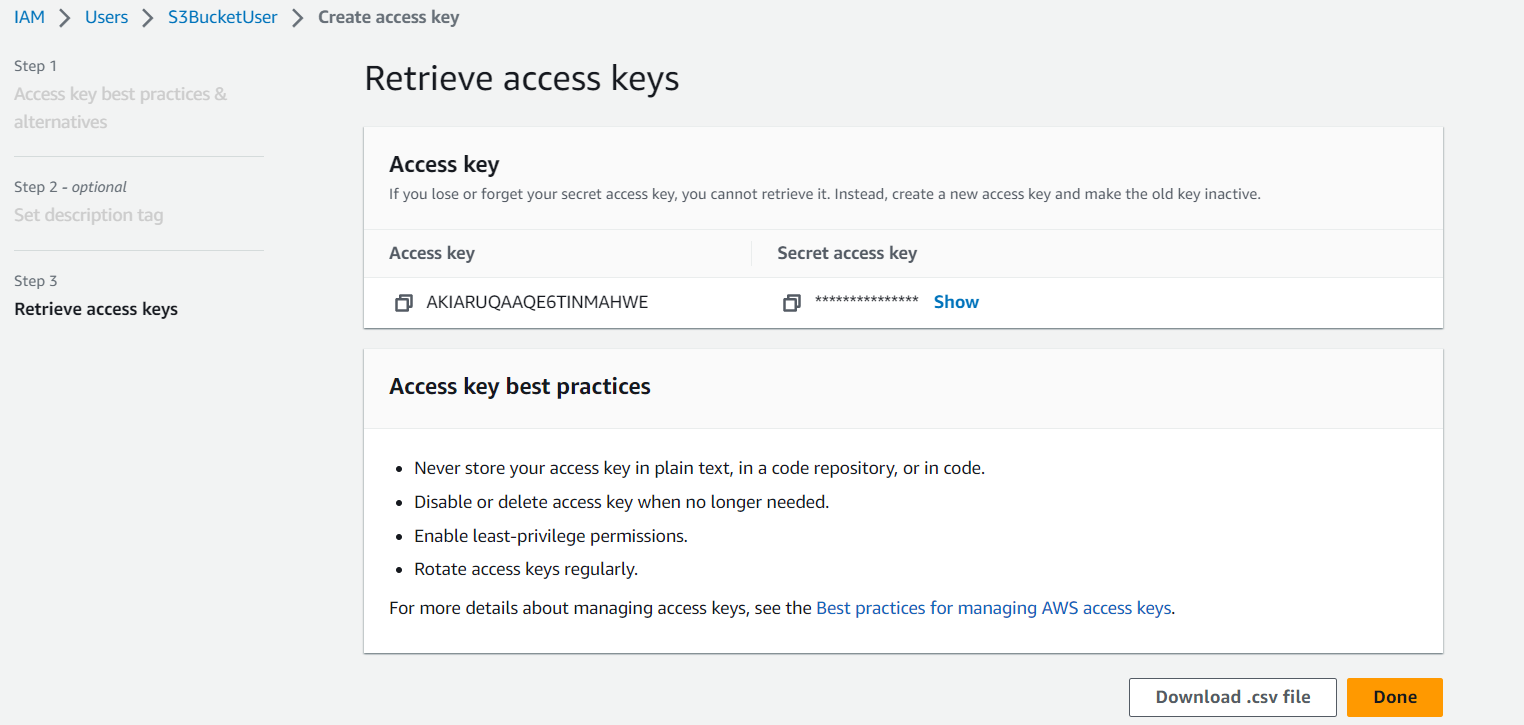
Primero en AWS (IAM) debemos tener un usuario con los permisos necesarios para acceder a S3  


Podemos

* crear un grupo, dar los permisos al grupo y asociarlo al usuario
* asociar directamente los permisos al usuario

Por temas de rapidez asociaremos directamente todos los permisos de S3 al usuario  


Ademas necesitamos que posea un access key y un secret access key

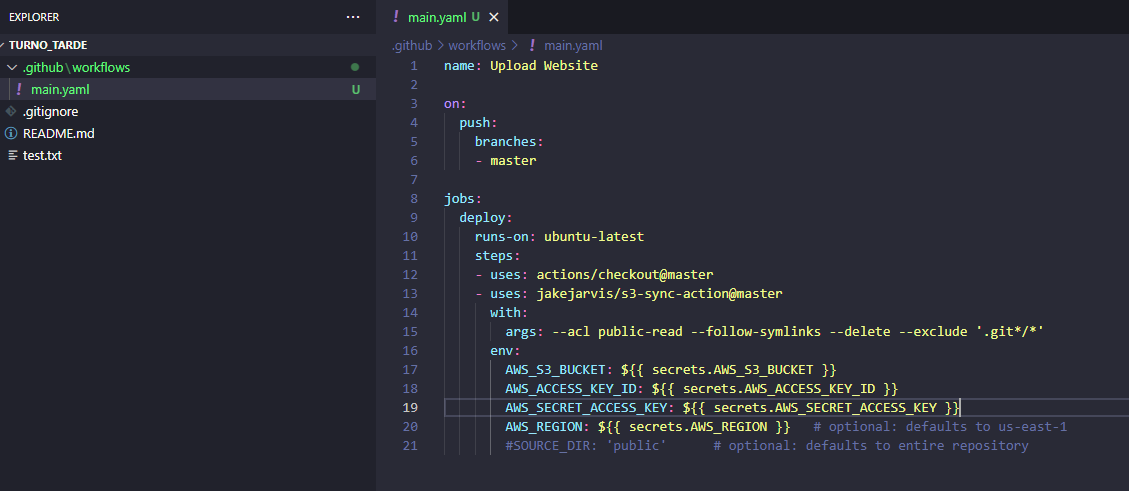


Para realizar una sincronizacion con Amazon S3 podemos usar la siguiente documentacion <https://github.com/marketplace/actions/s3-sync>

PASOS A REALIZAR EN EL REPOSITORIO

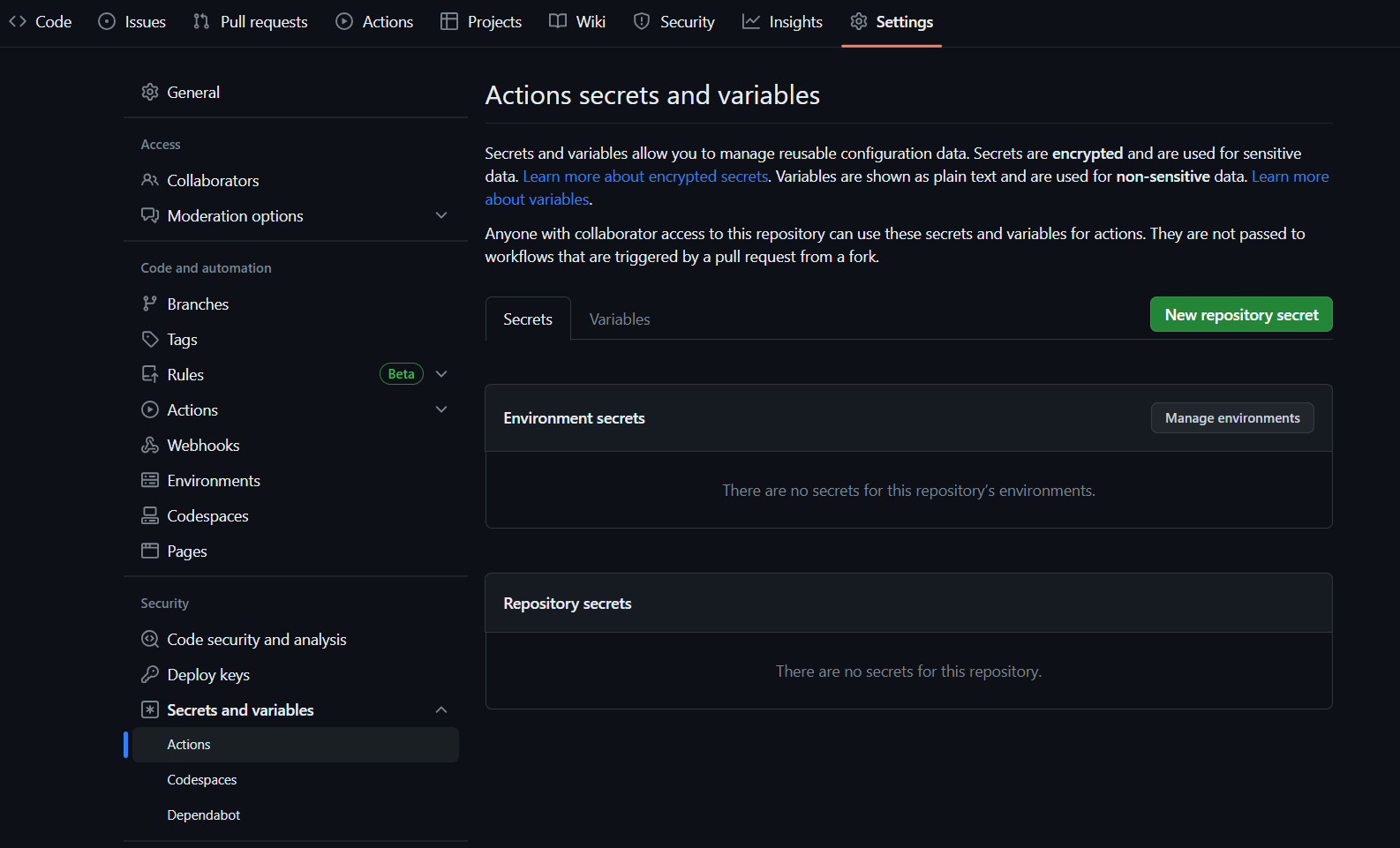
Paso 1: Crear una carpeta “.github/workflows” (notar el ‘.’ antes de github)

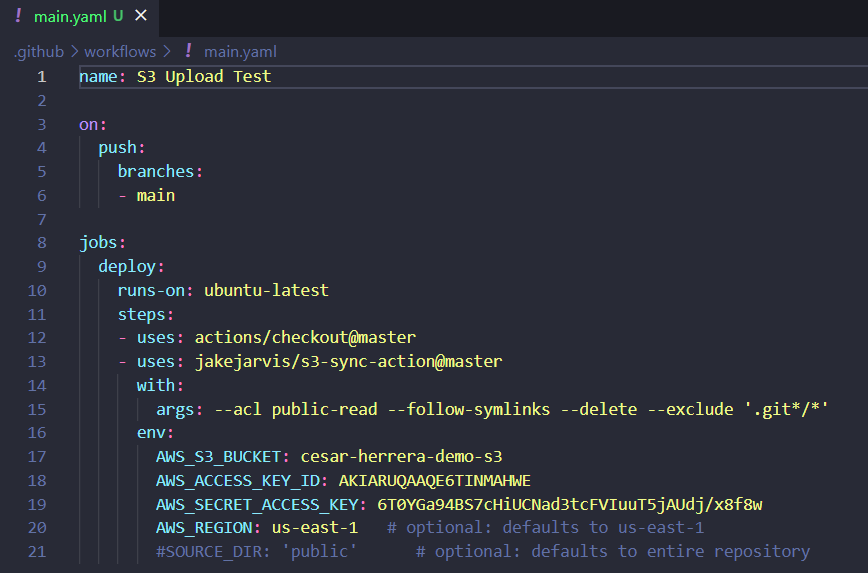
Paso 2: Crear un archivo .yaml (nosotros hemos creado main.yaml) y nos guiamos de la estructura que nos da la documentacion anterior



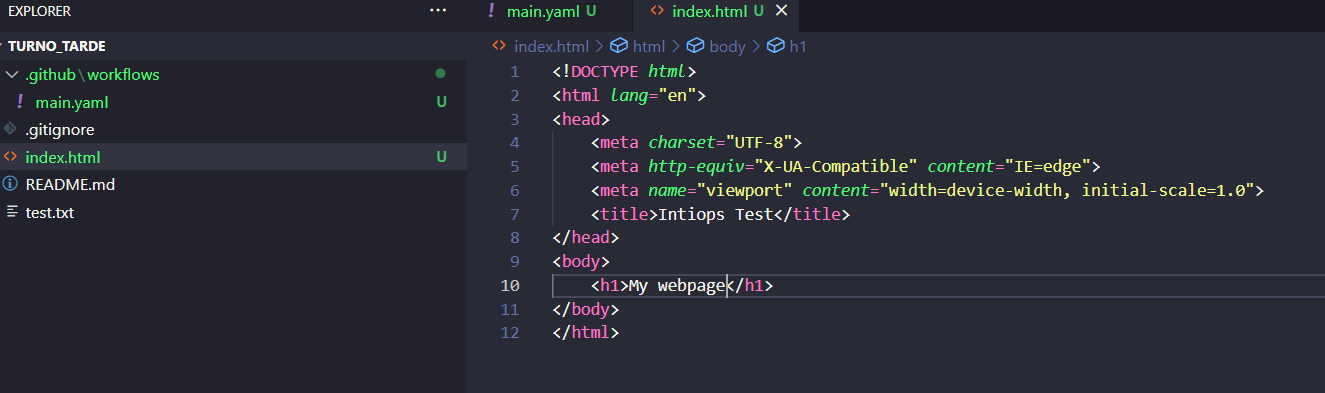
Algunas notas:

* Vemos que la accion se realiza en cada PUSH a la rama MASTER. En nuestro caso puntual, el repositorio posee una rama llamada MAIN, no MASTER. Es necesario realizar este cambio.
* Vemos que la accion a realizar es un DEPLOY, los STEPS son los pasos a realizar, y al parecer USES utiliza un repositorio de jakejarvis en la rama master (haremos una investigacion mas profunda sobre estos USES).
* En args hemos agregado --exclude '.git\*/\*' para evitar que cualquier folder u archivo que empiece con .git se sincronice a S3. Esto puede ser util si queremos que algunos archivos no sean subidos a S3.
* Env son variables de entorno que se configuran en github

Estas variables de entorno se configuran en el repositorio, sin embargo no tenemos acceso asi que lo haremos directamente en el archivo yaml (hardcoded values)  
Aqui un ejemplo de donde deberiamos configurar estas variables  


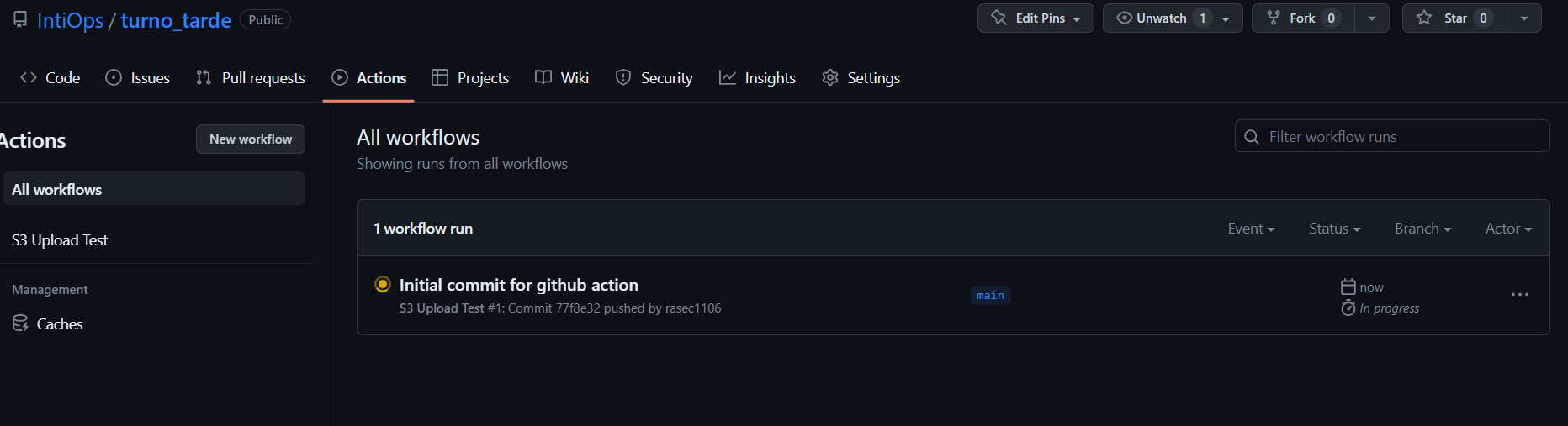
Una vez cambiado el archivo yaml  


Agregamos un archivo index.html



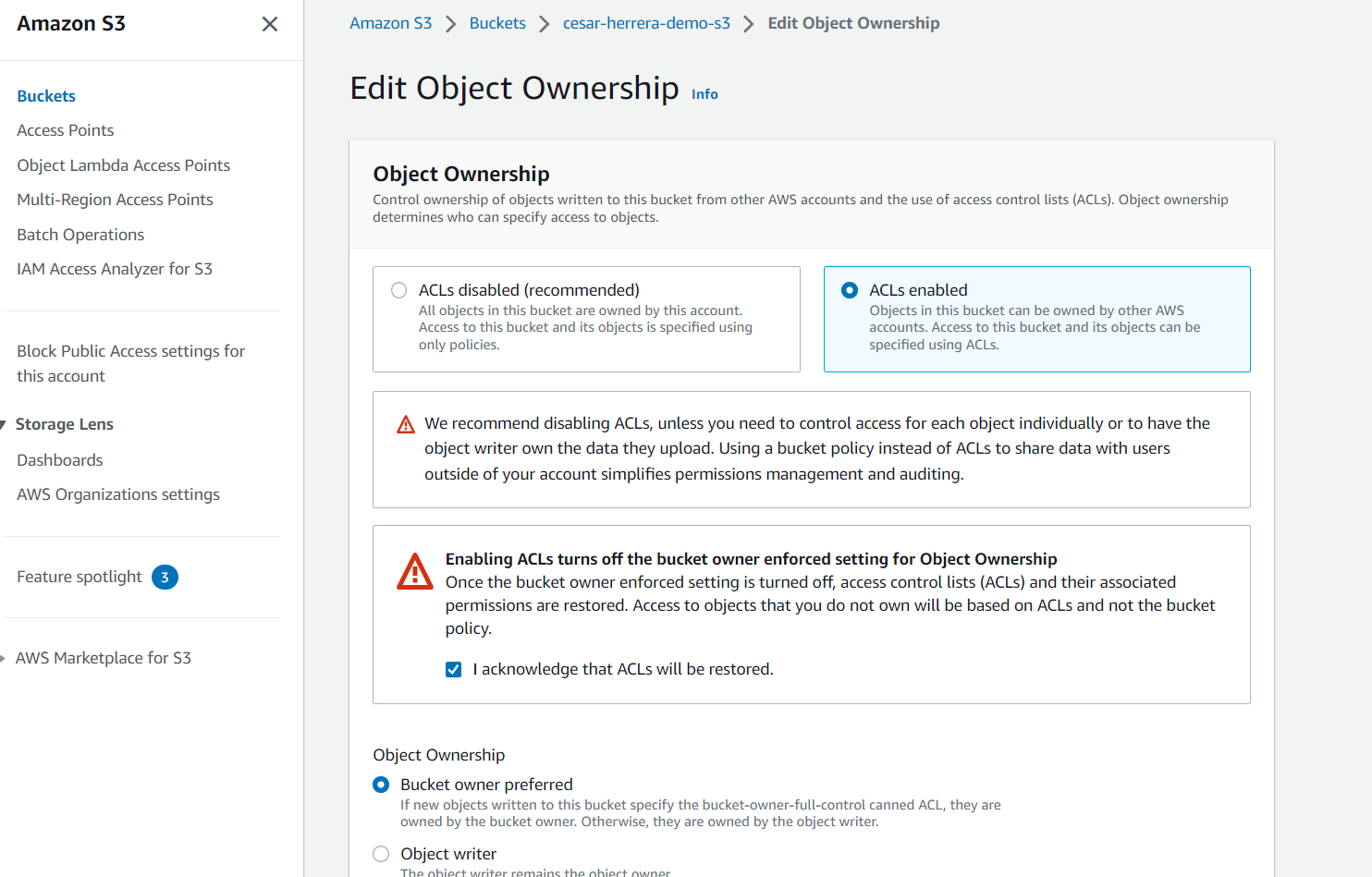
Y corremos los clasicos comandos de git (add, commit, push).

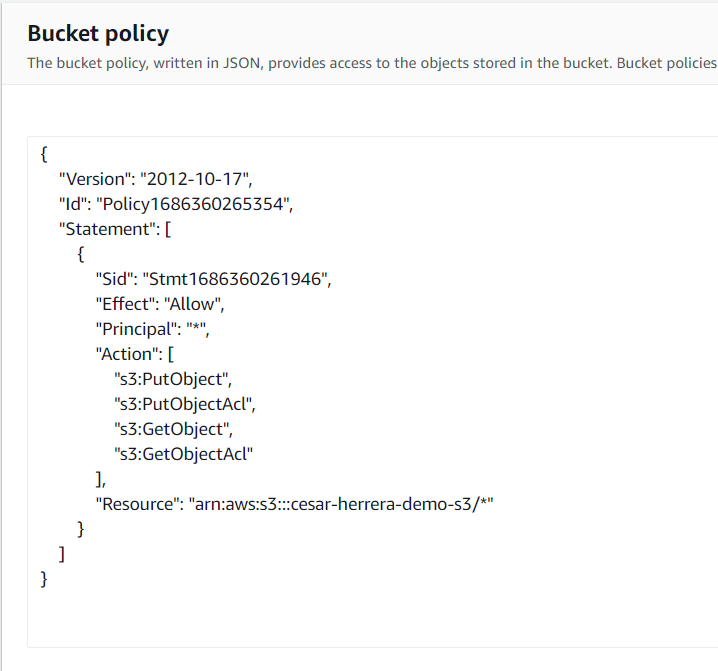
Luego vemos que el github action esta corriendo como esperabamos

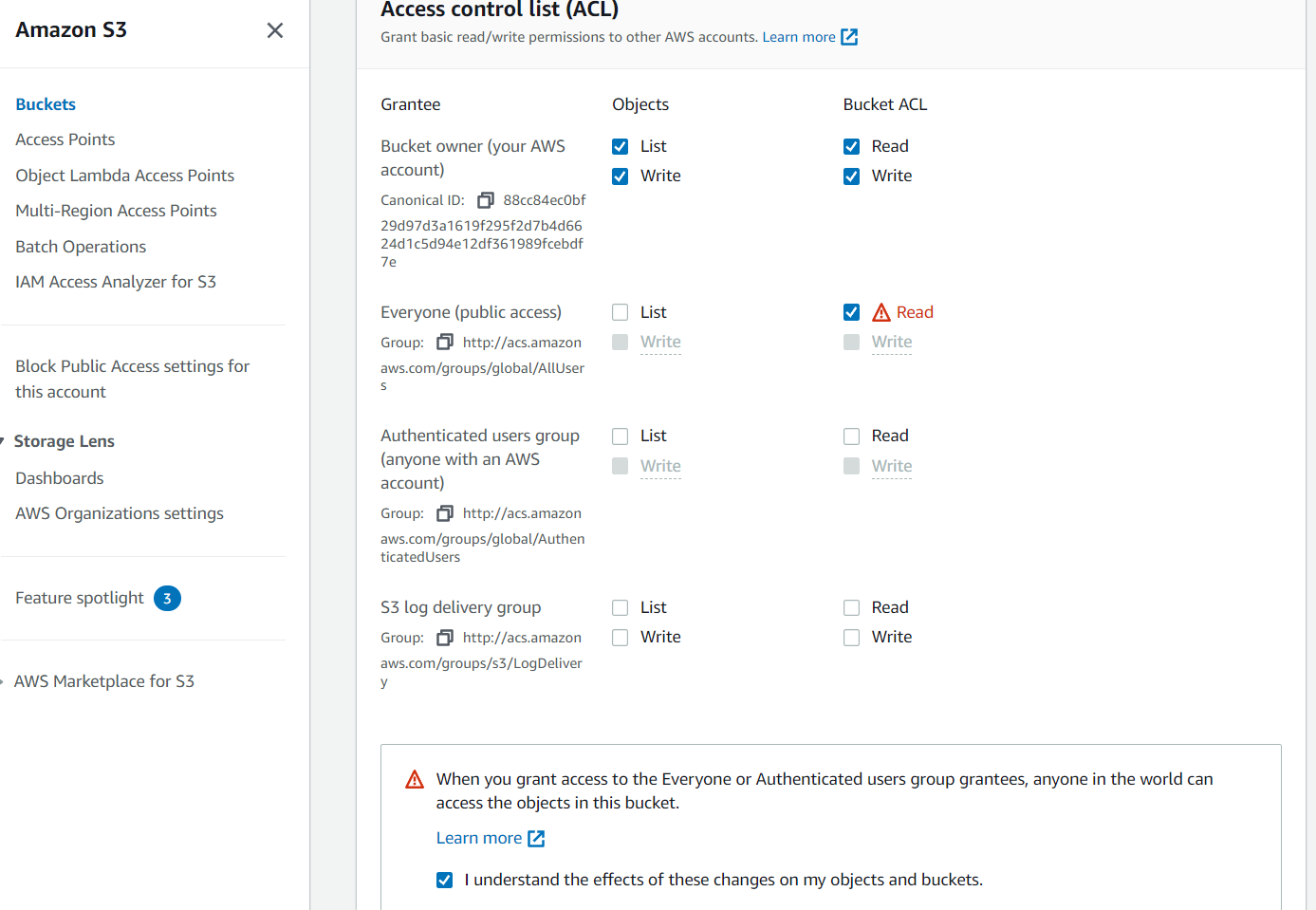


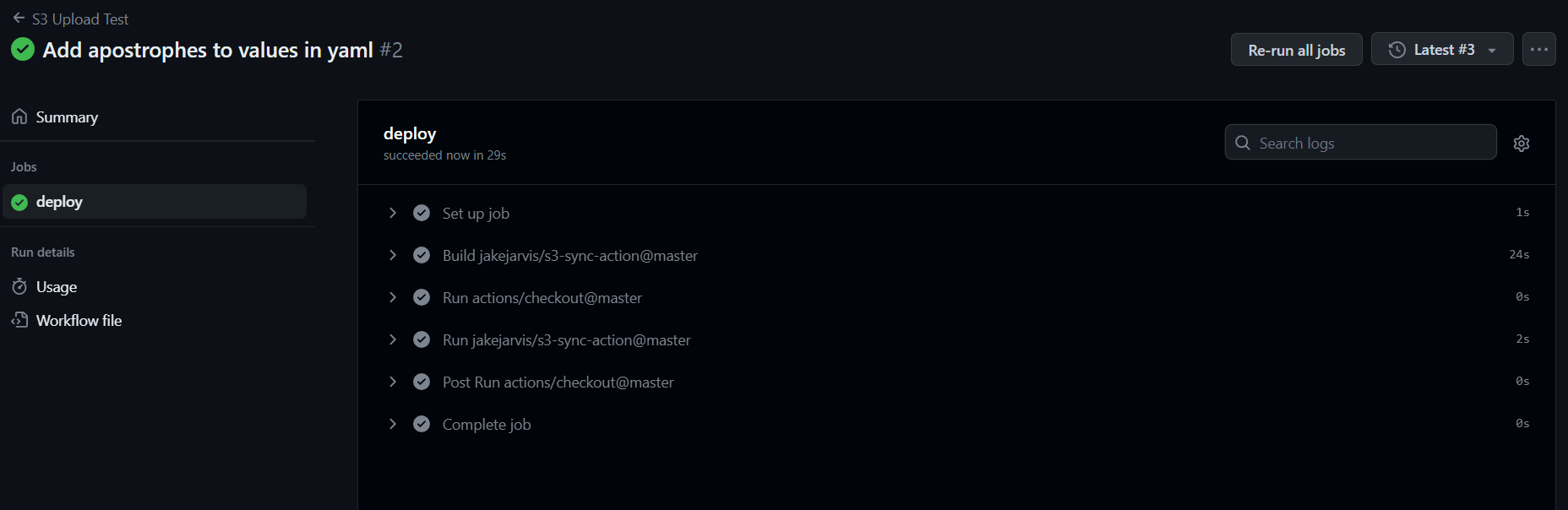
Aqui podemos observar un error  

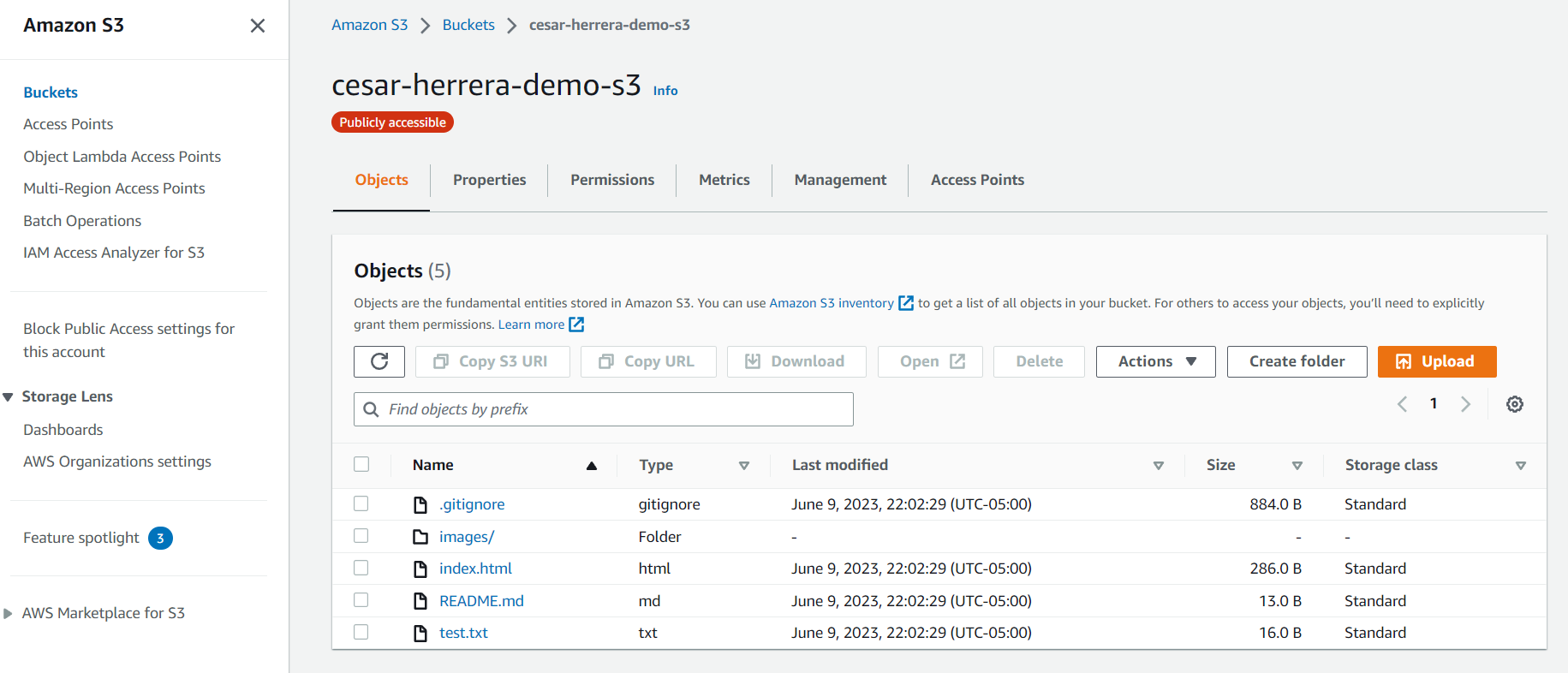

Si lo leemos observamos que dice “The bucket does not allow ACLs”

Necesitamos habilitar el ACL  


Ademas de dar politicas para escribir en el bucket   


Y habilitar un par de accesos mas en AWS  


Despues de esto observamos que el github action fue satisfactorio  


Si vemos en el bucket de AWS se subieron todos los archivos, aunque tambien el .gitignore (por revisar cual seria la correcta configuracion)  


Y tambien podemos acceder directamente a la url <https://cesar-herrera-demo-s3.s3.amazonaws.com/index.html>   
