DigitalHouse>

Certified Tech Developer

The Ultimate Degree

Infraestructura II

Ejercitación GitLab + AWS S3

OBJETIVO

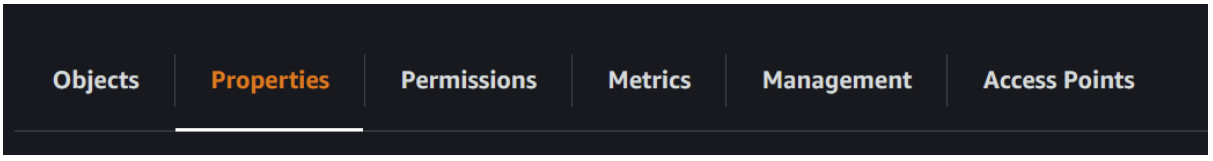
Generar un pipeline en GitLab que me permita pushear el repositorio a un bucket s3 configurando el bucket como *static website hosting* para poder visualizar la página públicamente.

RESOLUCIÓN

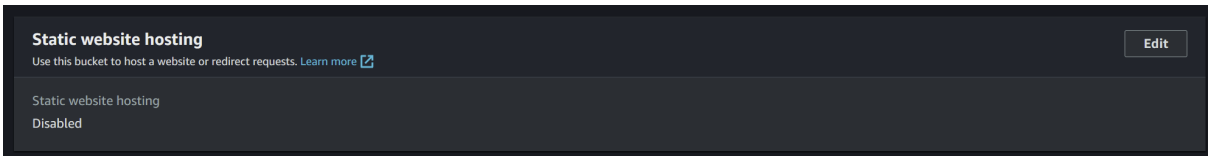
Continuamos integrando herramientas. En nuestro ambiente de GitLab debemos subir a un nuevo proyecto un template web. Podemos obtenerlo desde esta url: https://plantillashtmlgratis.com/en/home/. Tendremos que configurar un nuevo runner.

Antes de realizar todo esto, debemos crear nuestro bucket S3 en AWS y setearlo para hospedar páginas web estáticas.

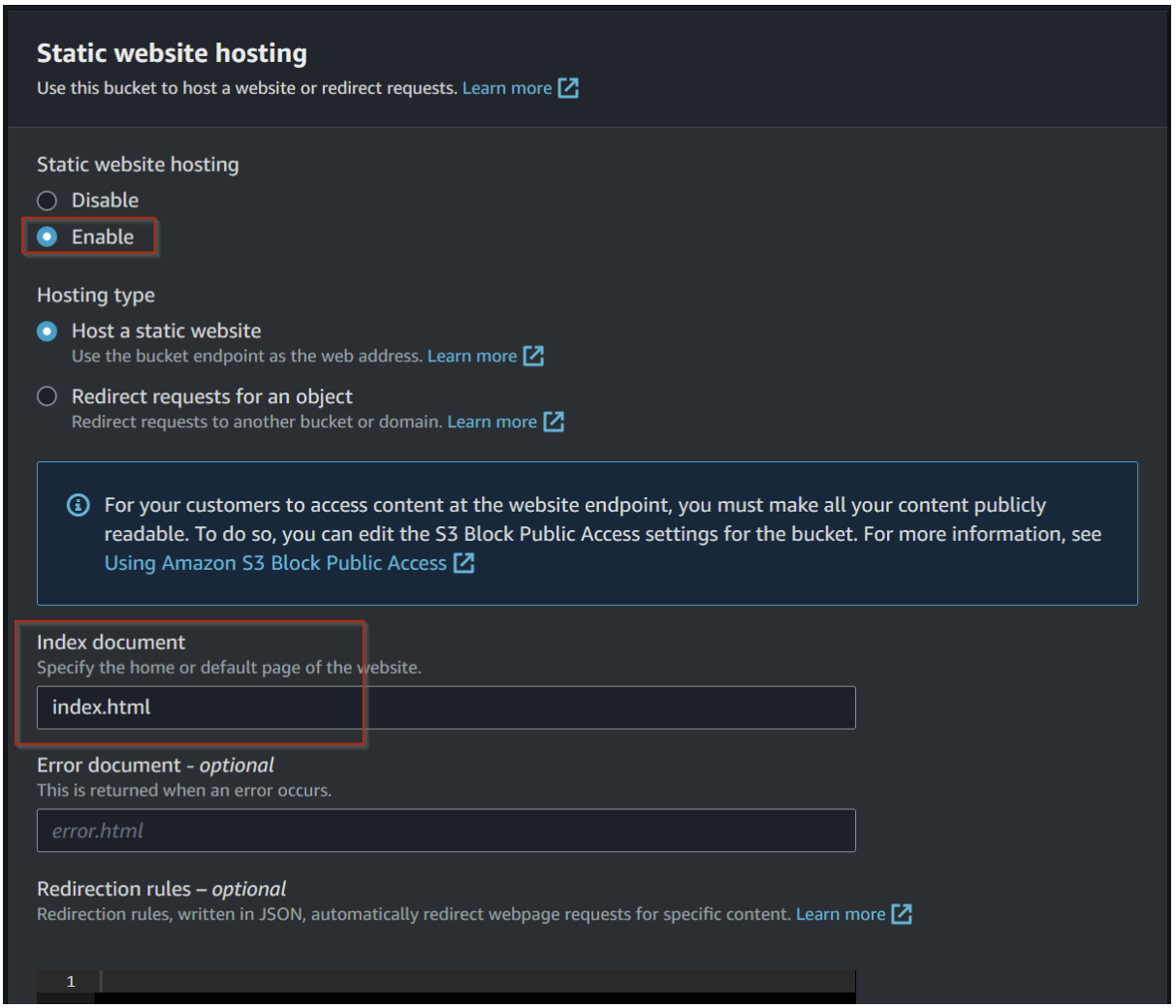
Esto lo hacemos desde la solapa propiedades del bucket



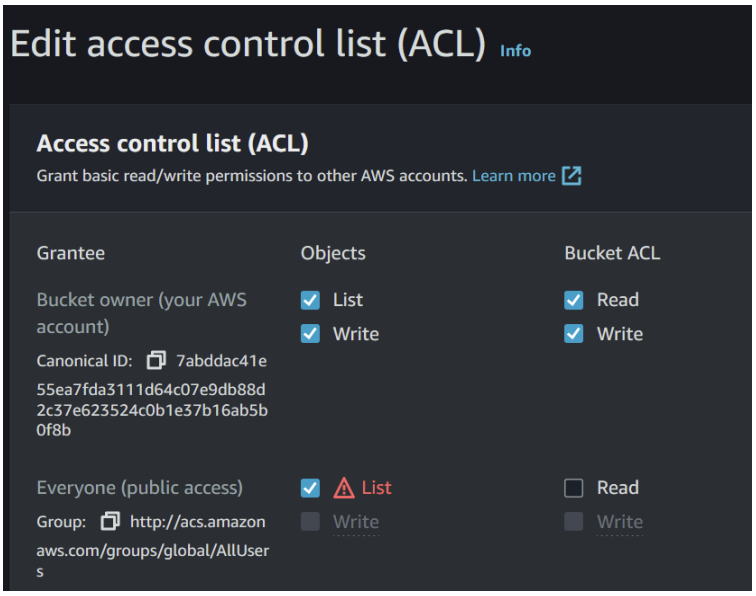
Y vamos hasta el final hasta encontrar la siguiente opción:



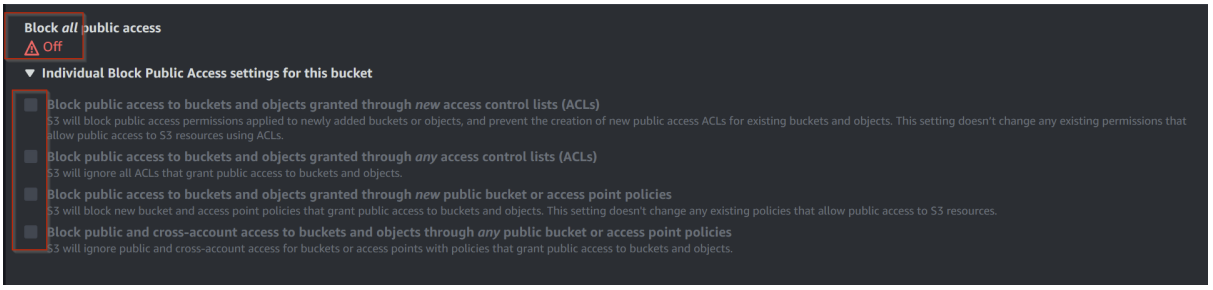
Una vez que habilitamos la opción, indicamos en la documentación del index el archivo principal de nuestra web que sería el index.html como se muestra en la imagen siguiente:



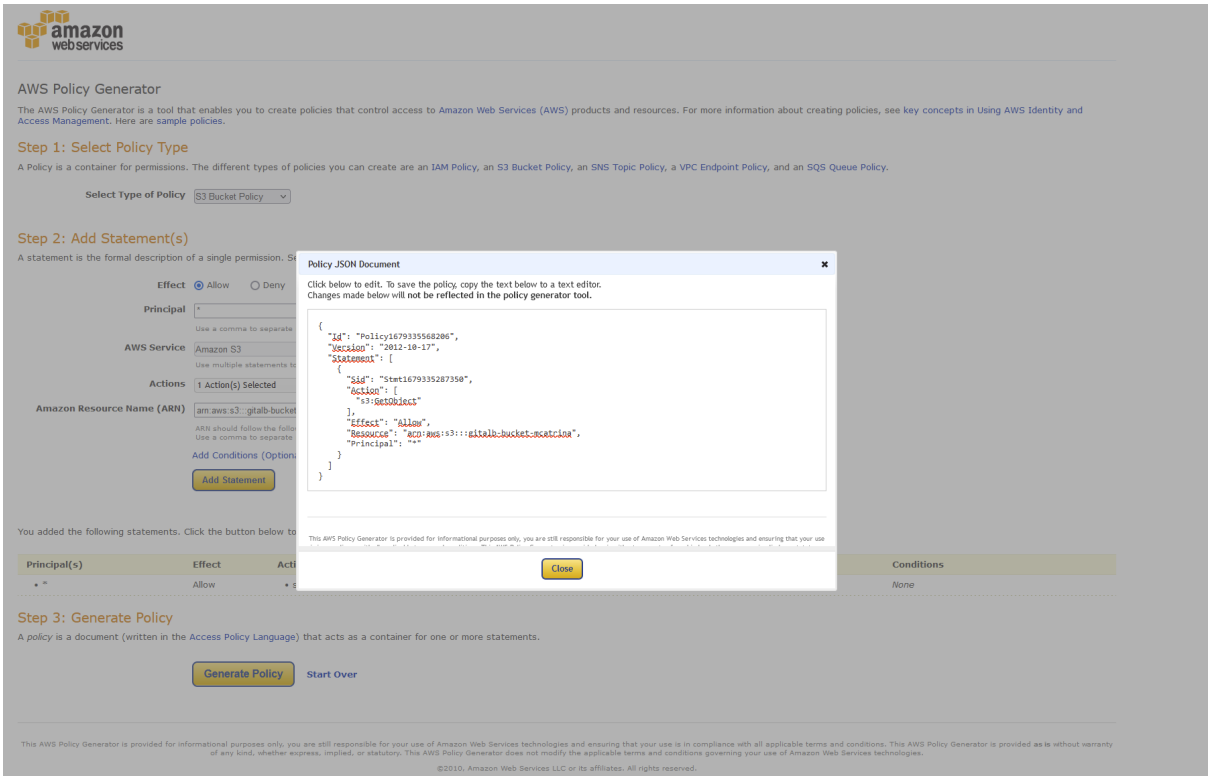
Verificamos que nuestro bucket posea acceso al público.



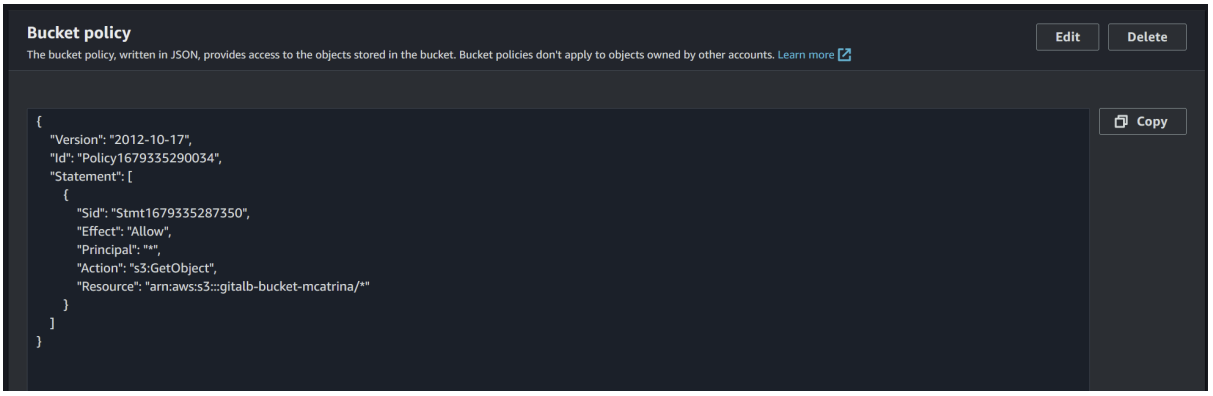
Y que no tenga bloqueos:



Generamos nuestra policy para el bucket del generador de policy de aws



Y nos quedaría de la siguiente manera:



Una vez ya teniendo nuestro bucket configurado procedemos a levantar un runner y configurarlo.

```
docker run -d --name gitlab-runner \
-v /srv/gitlab-runner/config:/etc/gitlab-runner \
-v /var/run/docker.sock:/var/run/docker.sock \
gitlab/gitlab-runner:latest
```

Ingresamos al contenedor y registramos el Runner pero configuramos el executor como **docker** y default image **python:3.6**

Sino recordamos con que comando registrar el runner es el siguientes:

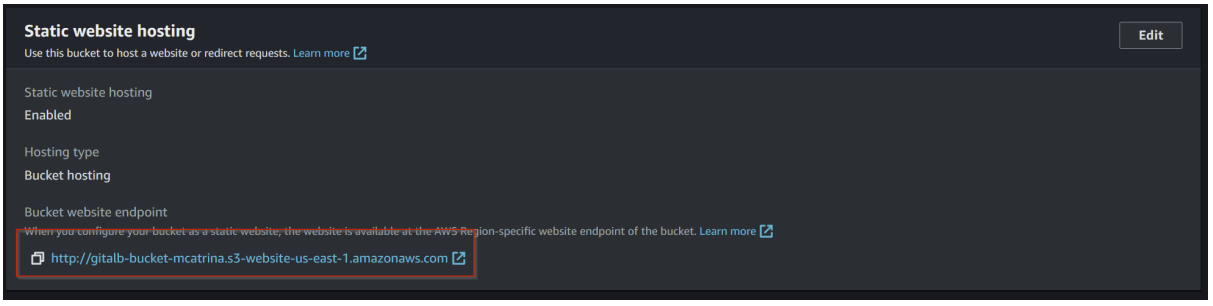
gitlab-runner register

Luego generamos nuestro archivo .gitlab-ci.yml dentro de nuestro repositorio de gitlab con los siguientes datos:

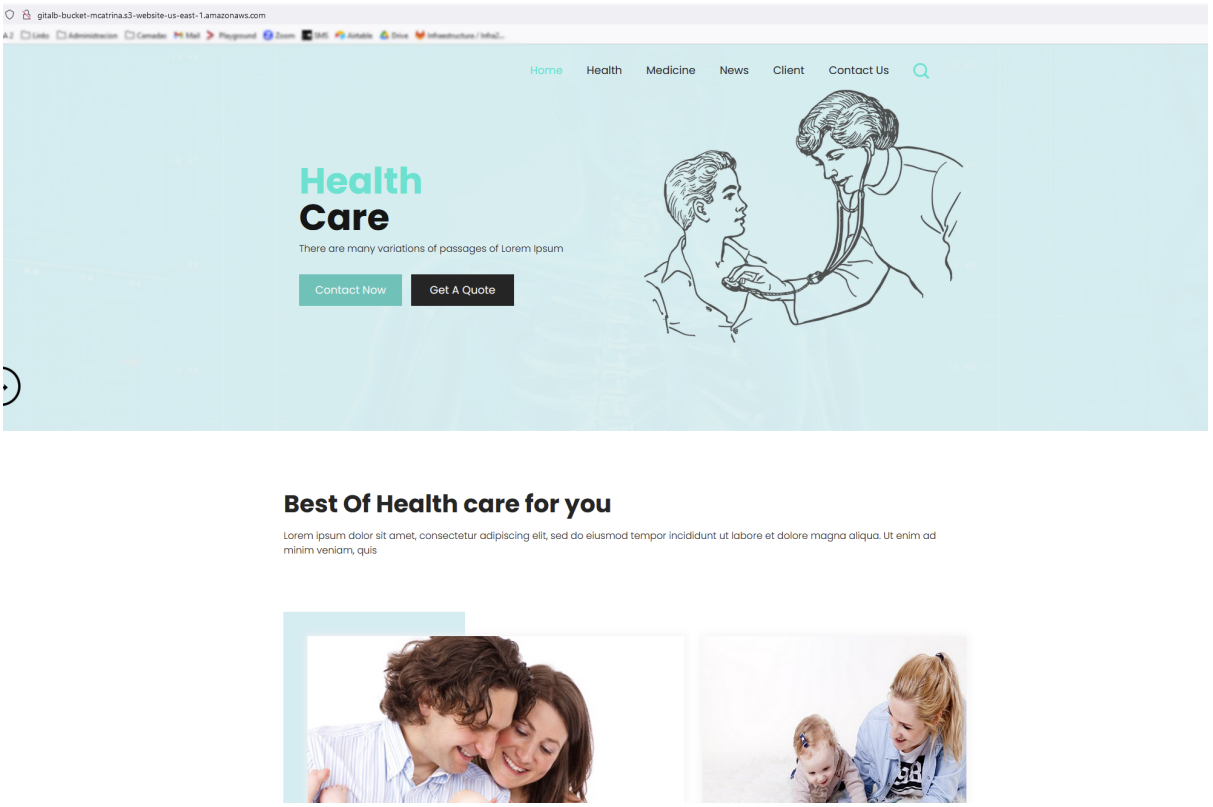
```
stages:
  - despliegue
deploy:
  stage: despliegue
  image: python:3.6
  script:
    - echo "Installing aws command line"
    - pip install awscli
    - aws --version
    - echo "Extracting the production build"
    - echo "Uploading files to AWS Bucket"
    - aws s3 cp --recursive . s3://<nuestro_bucket>
```

Generar las variables de AWS en la solapa de Variables en GITLAB.

Una vez ya nuestro pipeline se ejecutará de manera automática y para poder visualizar nuestra web desde el bucket tenemos que buscar el link desde el siguiente lugar:



Y ya tenemos nuestra web funcionando:



AYUDA

No nos olvidemos de que tenemos que configurar la red de nuestro runner y del contenedor. Dentro del contenedor del runner modificamos el archivo /etc/gitlab-runner/config.toml agregar el “network_mode = <nuestra red creada del contenedor>” al final de la línea.

Tanto el contenedor de Gitlab y el del Runner, tienen que estar dentro de la misma red.

Crear la red con "**docker network create <nombre>**"

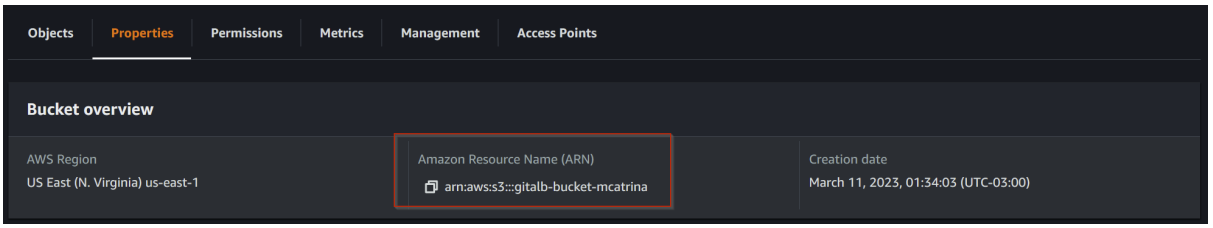
Conectarlo con "**docker network connect <nombre de la red creada> <contenedor id**>"

Pueden verificarlo con el comando "**docker network inspect <contenedor id>**"

Las políticas que utilice dentro del bucket S3 fueron la siguientes:

```
{
        "Version": "2012-10-17",
        "Id": "Policy1679335290034",
        "Statement": [
        {
        "Sid": "Stmt1679335287350",
        "Effect": "Allow",
        "Principal": "*",
        "Action": "s3:GetObject",
        "Resource": "arn:aws:s3:::gitalb-bucket-mcatrina/*"
        }
        ]
}
```

Les tocaría cambiar en la parte del Resource el nombre del bucket que ustedes tienen. En caso de que quieran averiguar de donde se obtiene, es desde las propiedades del bucket como lo muestro en la siguiente imagen:


Objects
Properties
Permissions
Metrics
Management
Access Points
Bucket overview
AWS Region
US East (N. Virginia) us-east-1
Amazon Resource Name (ARN)
arn:aws:s3:::gitalb-bucket-mcatrina
Creation date
March 11, 2023, 01:34:03 (UTC-03:00)