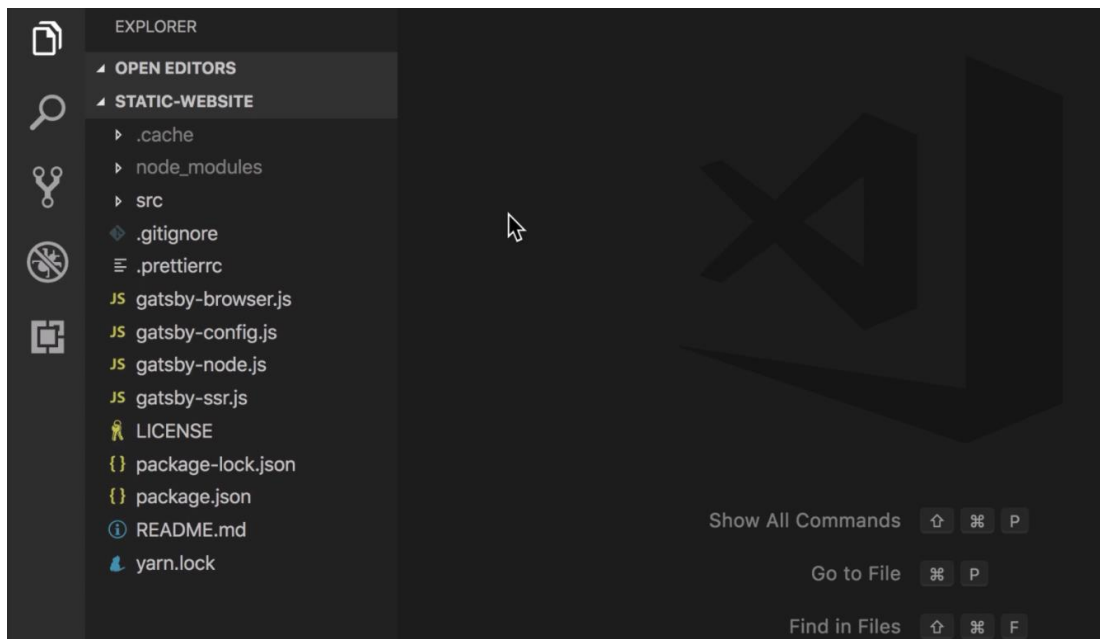


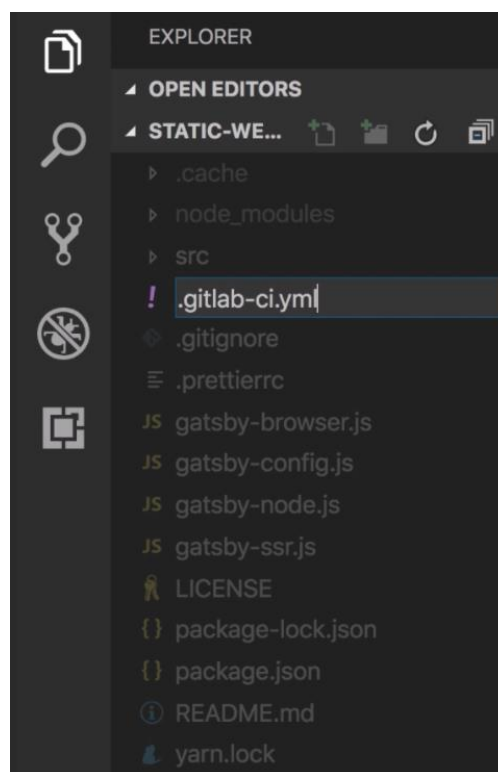
Construcción de un proyecto usando GitLab CI

Vamos ahora a echar un vistazo a cómo podemos utilizar GitLab CI para construir nuestro proyecto. Vamos a ejecutar los mismos pasos que hacemos en nuestra terminal para construir nuestro proyecto, pero desde un pipeline.

Tenemos abierto nuestro proyecto en VSCode, una simple página web hecha con Gatsby CLI:



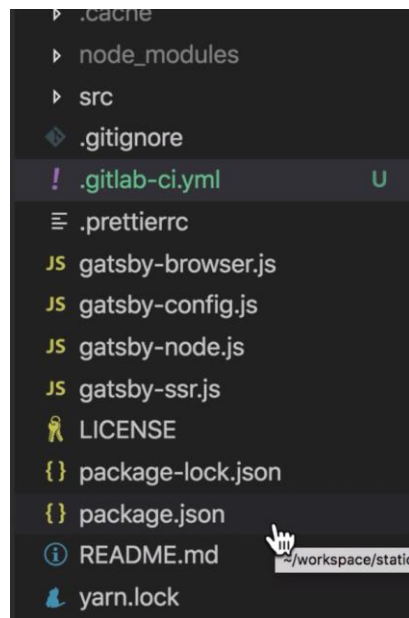
Primero, agregamos nuestro archivo de pipeline:



El primero paso es construir nuestro sitio web, así que agregamos el Job “build website”:

```
! .gitlab-ci.yml ●
1  build:website:
2  |
```

E nuestro script tenemos un par de cosas por hacer. En primer lugar, este es un proyecto JavaScript que utiliza NPM, y el archivo package.json define un conjunto de dependencias que deben ser instaladas, de lo contrario el proyecto no funcionará:



La instalación de las dependencias es un paso que GitLab también debe ejecutar, de lo contrario no podrá construir el proyecto. Para instalar las dependencias, todo lo que tenemos que hacer es ejecutar “npm install”:

```
! .gitlab-ci.yml ●
1  build:website:
2  |   script:
3  |     - npm install
4  |     -
```


Ahora, nosotros estamos utilizando “gatsby” para generar este sitio web, y lo necesitamos para compilar nuestra aplicación. Por supuesto, por defecto gatsby no está instalado, así que primero ejecutamos el comando de instalación:


```
! .gitlab-ci.yml ●  
1  build:website:  
2    · script:  
3    · · - npm install  
4    · · - npm install -g gatsby-cli
```

A continuación, debemos ejecutar el comando “gatsby build” para compilar nuestra aplicación:

```
! .gitlab-ci.yml ●  
1  build:website:  
2    · script:  
3    · · - npm install  
4    · · - npm install -g gatsby-cli  
5    · · - gatsby build
```

Ahora vamos a seguir adelante y hagamos commit de estos cambios y veamos como luce nuestro pipeline:

 Added pipeline
vdespa authored 43 seconds ago

 95dca211 

Added pipeline

parent `da2cee4d` `master`

test: running

Pipeline #56028721 running with stage `build website`

Changes 1 Pipelines 1

Desafortunadamente, tenemos un problema con nuestro pipeline. Tenemos un error que es muy claro, pues el comando `npm` nos está generando un error:

running Job #193784444 triggered 40 seconds ago by Valentin Despa

```
Running with gitlab-runner 11.9.0-rc2 (227934c0)
  on docker-auto-scale fa6cab46
Using Docker executor with image ruby:2.5 ...
Pulling docker image ruby:2.5 ...
Using docker image sha256:e86557c9a8ab97ec7e9ba705f3e3411428b50d19d38643a0f52cb49faf735d3d for ruby:2.5 ...
Running on runner-fa6cab46-project-11731229-concurrent-0 via runner-fa6cab46-srm-1554858638-e3b63874...
Initialized empty Git repository in /builds/vdespa/my-static-website/.git/
Fetching changes...
Created fresh repository.
From https://gitlab.com/vdespa/my-static-website
* [new branch]      master    -> origin/master
Checking out 95dca211 as master...

Skipping Git submodules setup
$ npm install
/bin/bash: line 74: npm: command not found
ERROR: Job failed: exit code 1
```

El error es realmente claro: *"npm command not found"*. ¿Qué significa este error? Por default, estamos usando una imagen Docker que está integrado en GitLab, por lo que nuestro Runner está usando esa imagen por default que contiene ruby en la versión 2.5:

```
Running with gitlab-runner 11.9.0-rc2 (227934c0)
  on docker-auto-scale fa6cab46
Using Docker executor with image ruby:2.5 ...
Pulling docker image ruby:2.5 ...
Using docker image sha256:e86557c9a8ab97ec7e9ba705f3e3411428b50d19d38643a0f52cb49faf735d3d for
ruby:2.5 ...
Running on runner-fa6cab46-project-11731229-concurrent-0 via runner-fa6cab46-srm-1554858638-
e3b63874...
Initialized empty Git repository in /builds/vdespa/my-static-website/.git/
Fetching changes...
Created fresh repository.
From https://gitlab.com/vdespa/my-static-website
* [new branch]      master    -> origin/master
Checking out 95dca211 as master...

Skipping Git submodules setup
$ npm install
/bin/bash: line 74: npm: command not found
ERROR: Job failed: exit code 1
```

Así que, esta imagen docker que estamos utilizando no tiene instalado Node ni npm. Y debido a que tuvimos que instalar localmente node y npm, con el fin de construir nuestra aplicación, tenemos que hacer lo mismo en GitLab.

Lo que tenemos que hacer es utilizar una imagen Docker que contenga Node y npm instalados. Así que vamos adelante y cambiemos nuestro pipeline un poco.

Debemos agregar la instrucción "image: docker:its":

```
! .gitlab-ci.yml x
1   build:website:
2   |   image: node
3   |   script:
4   |     - npm install
5   |     - npm install -g gatsby-cli
6   |     - gatsby build
```

Ahora vemos que el pipeline está iniciando y, en lugar de descargar la imagen Ruby, está descargando la imagen Docker node:

```
Running with gitlab-runner 11.9.0-rc2 (227934c0)
  on docker-auto-scale 0277ea0f
Using Docker executor with image node ...
Pulling docker image node ...

... ..
```

Ahora, probablemente se hará dado cuenta de que GitLab CI no es tan rápido, especialmente si ha trabajado antes con un servidor de CI como Jenkins. La diferencia con GitLab es cómo GitLab está trabajando, y especialmente en el uso de Docker, ya que todo lo que se está generando aquí se desecha después de finalizar la ejecución.

Ahora, nuestro Job finalizó su ejecución de forma exitosa y el sitio web estático ha sido generado:

```
success delete html and css files from previous builds - 0.007 s
success initialize cache - 0.008 s
success copy gatsby files - 0.015 s
success onPreBootstrap - 0.336 s
success source and transform nodes - 0.415 s
success building schema - 0.303 s
success createPages - 0.000 s
success createPagesStatefully - 0.056 s
success onPreExtractQueries - 0.002 s
success update schema - 0.030 s
success extract queries from components - 0.112 s
success run graphql queries - 0.460 s - 7/7 15.30 queries/second
success write out page data - 0.030 s
success write out redirect data - 0.007 s

success onPostBootstrap - 0.416 s

info bootstrap finished - 6.185493832 s

success Building production JavaScript and CSS bundles - 13.006 s
success Building static HTML for pages - 1.030 s - 4/4 16.58 pages/second
info Done building in 20.227351605 sec
Job succeeded
```

Desafortunadamente, una vez que este Job ha terminado, no tenemos ningún archivo de resultado disponible de nuestro sitio web. Así que, lo que tenemos que hacer con el fin de obtener nuestro artifact construido es especificar la ruta en donde se generó ese artifact:

```
! .gitlab-ci.yml ●
1  build:website:
2    image: node
3    script:
4      - npm install
5      - npm install -g gatsby-cli
6      - gatsby build
7    artifacts:
8      paths:
9        - ./public
```

Cuando finaliza la ejecución del Job, veremos que nuestros *artifacts* está disponibles en la parte derecha de la pantalla:

```
$ gatsby build
success open and validate gatsby-configs - 0.006 s
success load plugins - 0.318 s
success onPreInit - 0.005 s
success delete html and css files from previous builds - 0.006 s
success initialize cache - 0.008 s
success copy gatsby files - 0.015 s
success onPreBootstrap - 0.306 s
success source and transform nodes - 0.389 s
success building schema - 0.250 s
success createPages - 0.000 s
success createPagesStatefully - 0.053 s
success onPreExtractQueries - 0.002 s
success update schema - 0.022 s
success extract queries from components - 0.108 s
success run graphql queries - 0.400 s - 7/7 17.56 queries/second
success write out page data - 0.051 s
success write out redirect data - 0.004 s

success onPostBootstrap - 0.421 s

info bootstrap finished - 5.766224641 s

success Building production JavaScript and CSS bundles - 11.856 s
success Building static HTML for pages - 0.989 s - 4/4 18.29 pages/second
info Done building in 18.617327144 sec
Uploading artifacts...
./public: found 62 matching files
Uploading artifacts to coordinator... ok      id=193787103 responseStatus=201 Created
token=21cG7RG-
Job succeeded
```

build website

Duration: 2 minutes 46 seconds
Timeout: 1h (from project)
Runner: shared-runners-man...
3.gitlab.com (#44028)

Job artifacts
Download Browse

Commit faa15b55

Add artifacts

Pipeline #56029470 from

test

→ build website

Si hacemos click en el botón “browser” podremos navegar en los archivos que generamos:

passed **Job #193787103** in pipeline **#56029470** for **faa15b55** from **master** by **Valentin Despa** 3 minutes

Artifacts

Name	Size
public	