Ejecutando Jobs en Paralelo

Ahora continuemos con nuestras pruebas y hagamos algunas optimizaciones.

Hemos visto anteriormente que nuestro Job “test artifact” está utilizando la imagen por defecto que proporciona GitLab, que es la que contiene Ruby instalado, vamos a cambiarla por otra imagen más ligera de node, la versión “alpine”:

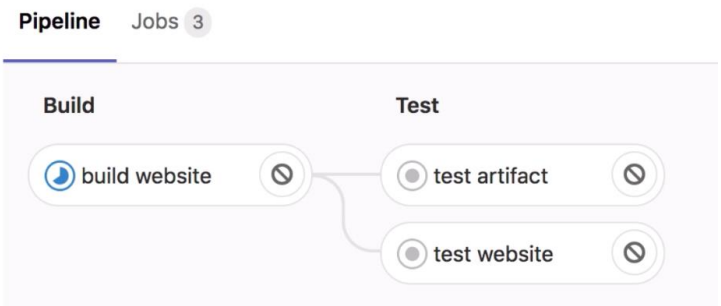
```
16  test artifact:
17    image: alpine
18    stage: test
19    script:
20      - grep -q "Gatsby" ./public/index.html
```

Sigamos adelante y añadamos otro Job.

Gatsby nos ofrece la posibilidad de iniciar un Server, y luego usar HTTP para acceder al sitio web que hemos creado. Hagamos esto asignando nuestro nuevo Job al Stage “test”:

```
22  test website:
23    image: node
24    stage: test
25    script:
26      - npm install
27      - npm install -g gatsby-cli
28      - gatsby serve
29      - curl "http://localhost:9000" | grep -q "Gatsby"
```

Después de esto, veamos como luce nuestro pipeline:



Una de las primeras cosas que podemos notar es que ahora tenemos dos Jobs ejecutándose en paralelo. Esto es porque hemos creado el Job “test artifact” y el Job “test website” y los hemos asignado al mismo Stage “test”, y cuando hacemos eso, le estamos diciendo a GitLab que debe ejecutar esos Jobs en paralelo.

Ahora, en este caso específico, la ejecución del Job en paralelo probablemente no hará que se ejecute más rápido, ya que tendrá que descargar aun así la imagen Docker, pero hay escenarios en los que si tiene sentido ejecutar Jobs en paralelo. Si estamos interesados en aplicar optimización a nuestro proceso, esta es una buena forma de hacerlo.

Hay que aclarar que no todos los Jobs pueden ser paralelizados, especialmente si hay dependencias entre los Jobs. Por ejemplo, no podemos ejecutar el paso “build” y el paso “test” en paralelo porque, para probar el artifact, primero tenemos que construirlo. Por lo tanto, si hay alguna dependencia entre los Jobs, entonces no podemos paralelizarlos.

Después de un rato, podemos ver ahora que GitLab está ejecutando ambos Jobs en el Stage “test” en paralelo: