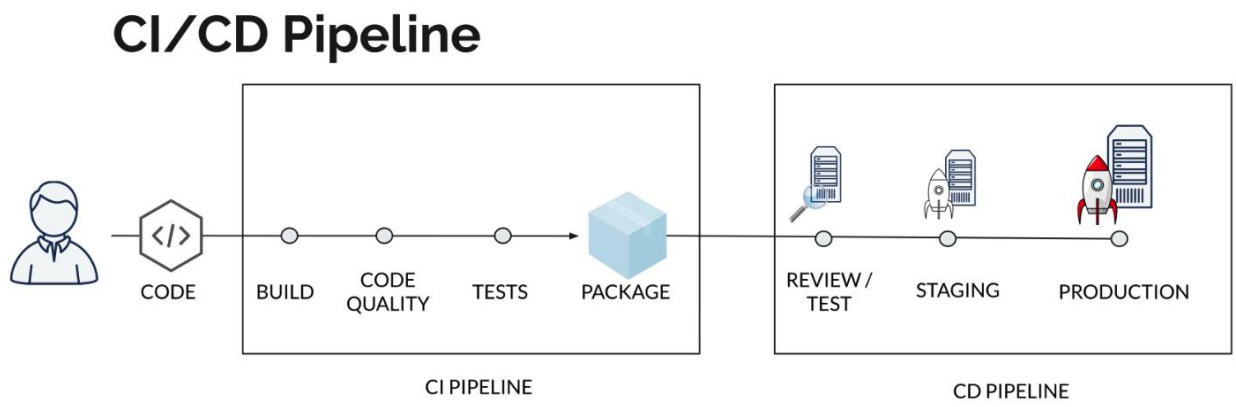
Variables de Despliegue

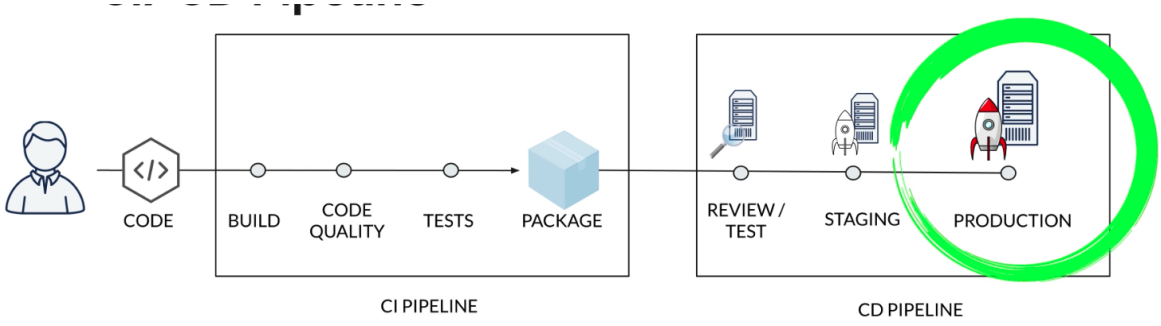
En esta lección, veremos como agregar otro entorno a nuestro proceso de CD.

Si volvemos a nuestra idea original, entre el CI y el CD, podemos ver que la línea de CI está casi completa, ya que tenemos nuestro Job de construcción, otro Job con algunos tests y otro más en el que empaquetamos:

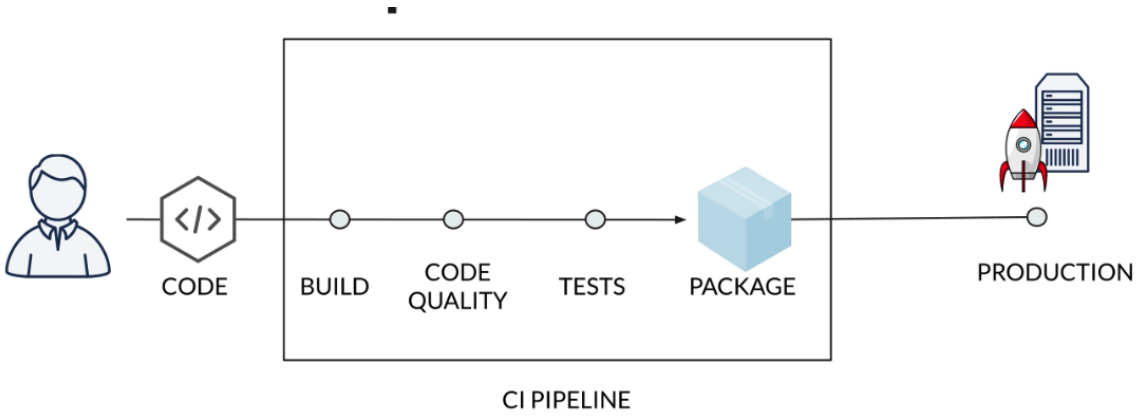


Claro que por el momento nuestros tests son fake, ya que no estamos testeando nada.

Ahora, en la parte del CD, sólo hemos añadido un entorno de producción:

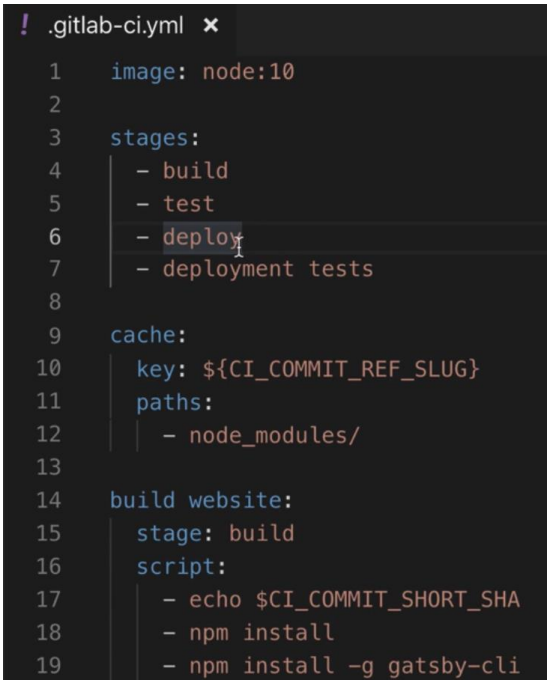


Esto hace que el CD se vuelva difícil, ya que no estamos probando nuestro proceso de despliegue y estamos desplegando directamente en producción:



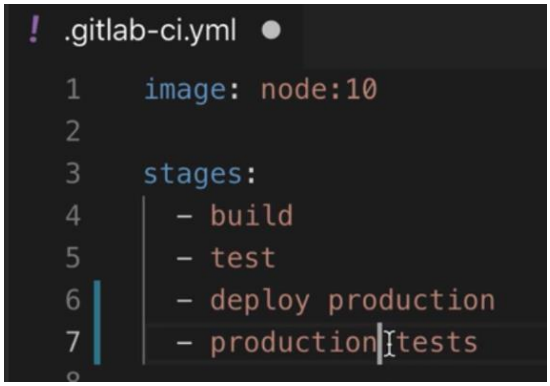
Otro problema es que no nos permite revisar los cambios que hemos realizado en nuestra aplicación antes de desplegar en producción. Por esta razón, agregaremos un entorno de Staging a nuestro pipeline.

Antes que nada, veamos los Stages que tenemos hasta el momento:

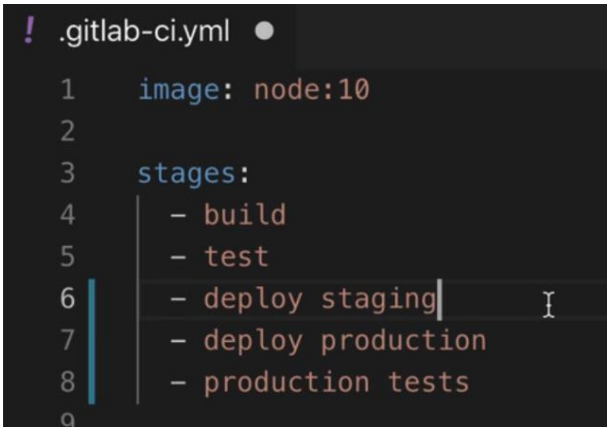


Hagamos algunos cambios menores en los Stages. Primero que nada, el Stage “deploy” ahora se llamará “deploy production”:

Y podemos nombrar el Stage “deployment tests” como “production tests”:

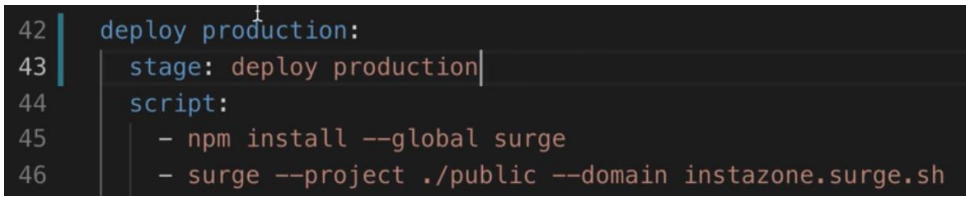


Ahora, agregaremos un nuevo Stage llamado “deploy staging”:

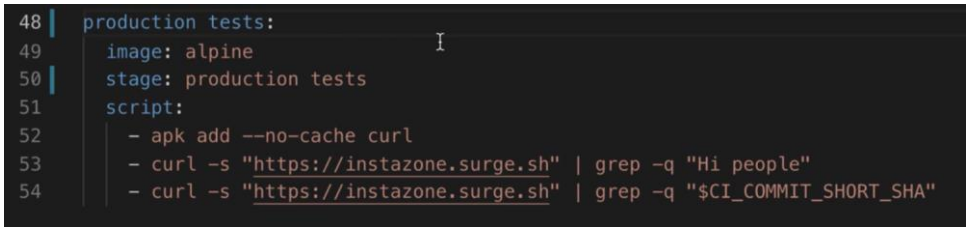


Entonces, nuestro pipeline primero construirá, testeará y luego desplegará en “Staging” y al final desplegará en producción:

Así que cambiemos los Jobs que tenemos hasta ahora. Nombraremos los Jobs igual que el Stage que le corresponde:



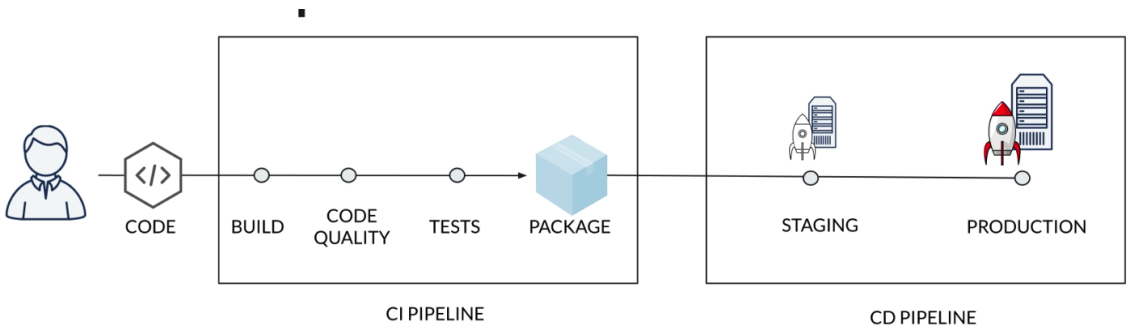
Lo mismo con el Job “test deployment”:



Agreguemos otro Job para el despliegue en Staging:

```
42  deploy staging:
43    stage: deploy staging
44    script:
45      - npm install --global surge
46      - surge --project ./public --domain instazone-staging.surge.sh
```

Ahora tenemos un nuevo entorno de despliegue llamado Staging:



Esto es muy bueno porque cuando hacemos CD, rara vez queremos desplegar directamente en producción. Agregar esta etapa de Staging o cualquier otra etapa antes del entorno de producción es una buena práctica porque nos permite ejecutar algunas pruebas adicionales, diferentes tipos de pruebas que podemos ejecutar llamadas "Pruebas de integración" o Pruebas de aceptación".

Por suerte para nosotros, GitLab tiene el concepto de "environments". Los "environments" nos permiten controlar el proceso de CD de nuestro software. Nos permite realizar un seguimiento de nuestros despliegues y lo que es más importante, podemos agregar "tags" a nuestros Jobs. De esta manera, podemos decirle a GitLab lo que estamos haciendo y luego GitLab podrá realizar un seguimiento de lo que hacemos. Así sabremos qué tenemos instalado actualmente, en qué entorno y qué versión, entre otras cosas.

Para usar lso "environments", hay una pequeña configuración que debemos realizar para etiquetar nuestros Jobs. Esto se hace utilizando la instrucción "environment":

```
42  deploy staging:
43    stage: deploy staging
44    environment:
45
46    script:
47      - npm install --global surge
48      - surge --project ./public --domain instazone-staging.surge.sh
```

Hay dos propiedades que debemos definir. Primero que nada, tenemos que darle un nombre en este caso será “stagging”:

```
42  deploy staging:
43    stage: deploy staging
44    environment:
45      name: staging
46
```

Además, podemos agregar una URL:

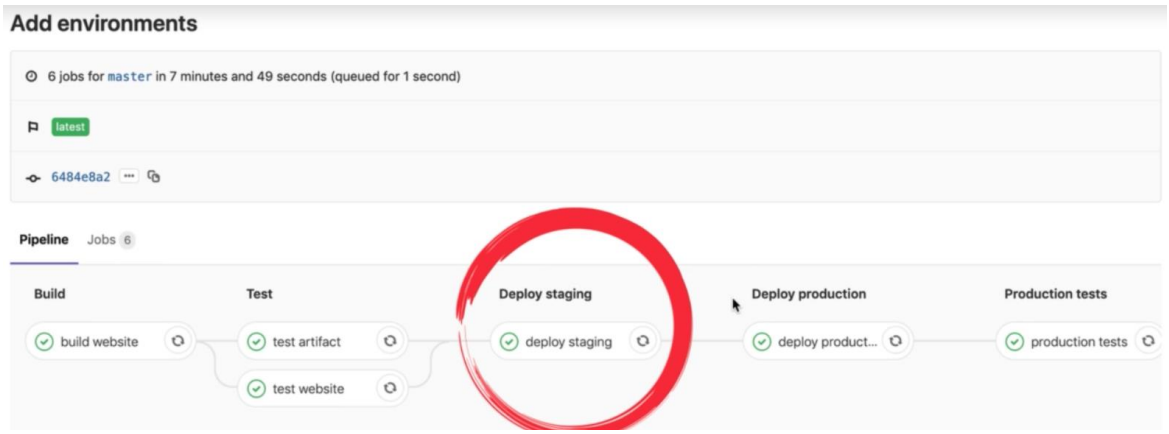
```
42  deploy staging:
43    stage: deploy staging
44    environment:
45      name: staging
46      url: http://instazone-staging.surge.sh
47    script:
48      - npm install --global surge
49      - surge --project ./public --domain instazone-staging.surge.sh
```

Ahora hemos etiquetado nuestro Job y GitLab sabrá como darle seguimiento.

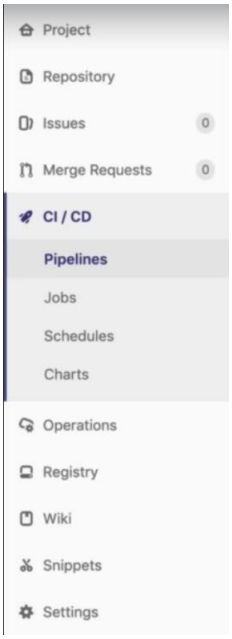
Podemos agregar la misma configuración a nuestro Job “deploy production”:

```
51  deploy production:
52    stage: deploy production
53    environment:
54      name: production
55      url: http://instazone.surge.sh
56    script:
57      - npm install --global surge
58      - surge --project ./public --domain instazone.surge.sh
```

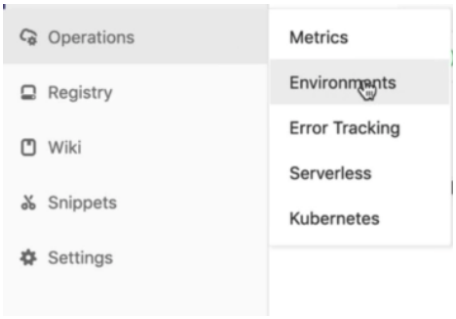
Probémoslo y veamos cómo funciona. Si volvemos a nuestro pipeline y hechamos un vistazo a lo que sucede, veremos que en nuestro preview del pipeline no hay nada nuevo, a excepción de nuestro Job “Deploy Staging”:



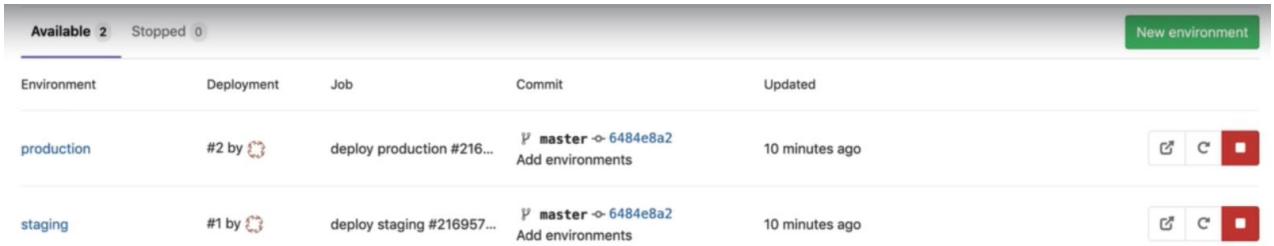
Parece que lo que hemos hecho con los environments realmente no ha tenido efecto aquí. Y para ver exactamente cómo podemos rastrear los environments, debemos ir al panel izquierdo y seleccionar la opción “Operations”:



Luego seleccionamos la opción “Environments”:



Ahora podemos ver lo que hicimos:



Ahora podemos ver los environments que hemos creado, junto con el commit que desplegamos en ese environment.