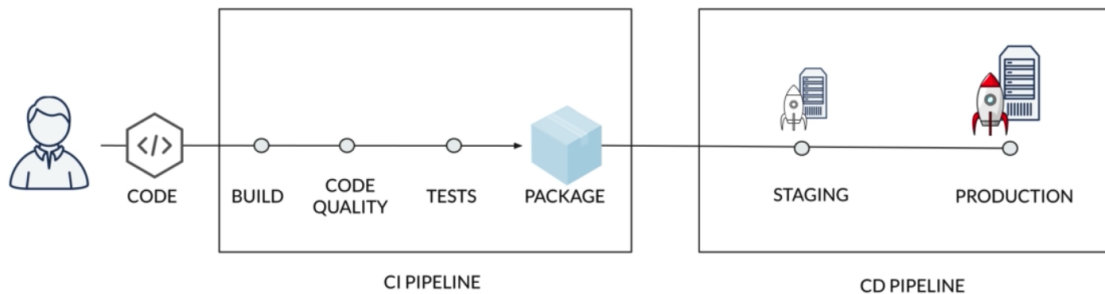


## Ambientes Dinámicos

Parece que ahora podemos trabajar con merge requests y con branches, pero aun así solo tenemos dos entornos, Staging y Production:



La pregunta es, ¿no sería bueno tener para cada branch o para cada feature, una manera de ver los cambios que hemos realizado? En realidad, esto es posible. Podemos crear entornos para cada merge request. Podemos activar automáticamente un entorno dinámico para cada merge request. Esto nos permite revisar los cambios en un sistema y si es necesario, podemos ejecutar algunas pruebas más avanzadas. De esta forma, un no desarrollador podría inspeccionar el trabajo hecho visitando una URL y no teniendo que descargar todo el código, compilar e iniciar servidores. Esto puede ser muy conveniente y es exactamente lo que haremos en esta lección. Vamos a agregar un entorno dinámico para cada merge request.

Para hacer esto, vamos a agregar un nuevo Stage al que llamaremos “review”, ya que estará enfocado a realizar revisiones del sistema:

```
3  stages:
4    - build
5    - test
6    - deploy review
7    - deploy staging
8    - deploy production
9    - production tests
```

Ahora vamos a agregar el Job:

```
56  deploy review:
57    stage: deploy review
58    only:
59      - merge_requests I
60    script:
61      - npm install --global surge
62      - surge --project ./public --domain $STAGING_DOMAIN
```

Esta vez no especificamos que se ejecute para el branch master, ya que solo queremos que se ejecute para los merge request, por eso agregamos solo el valor merge\_request.

Ahora debemos especificar el entorno, lo cual hacemos de la siguiente manera:

```
56  deploy review:
57    stage: deploy review
58    only:
59      - merge_requests
60    environment:
61      name: review/$CI_COMMIT_REF_NAME
62      url: https://instazone-$CI_ENVIRONMENT_SLUG.surge.sh
63    script:
64      - npm install --global surge
65      - surge --project ./public --domain $STAGING_DOMAIN
```

Especificamos el entorno y lo llamamos “review/” seguido de una variable especial que nos proporciona GitLab llamada `CI_COMMIT_REF_NAME`. La dirección que utilizamos es [https://instazone-\\$CI\\_ENVIRONMENT\\_SLUG.surge.sh](https://instazone-$CI_ENVIRONMENT_SLUG.surge.sh), siendo `CI_ENVIRONMENT_SLUG` una variable proporcionada por GitLab.

Ya casi tenemos todo lo que necesitamos. Debemos agregar el dominio correcto para que se despliegue apropiadamente en instazone:

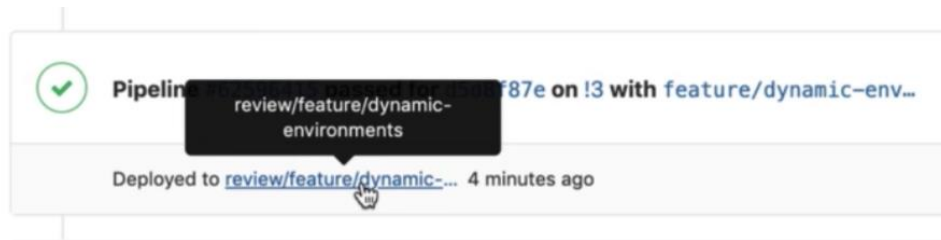
```
56  deploy review:
57    stage: deploy review
58    only:
59      - merge_requests
60    environment:
61      name: review/$CI_COMMIT_REF_NAME
62      url: https://instazone-$CI_ENVIRONMENT_SLUG.surge.sh
63    script:
64      - npm install --global surge
65      - surge --project ./public --domain instazone-$CI_ENVIRONMENT_SLUG.surge.sh
```

Así que vamos a crear un nuevo branch y a hacer push a ese branch para ver cómo funciona:

### Add dynamic environments

The screenshot shows a GitLab merge request interface. At the top, it says "Request to merge feature/dynamic-env... into master" with buttons for "Open in Web IDE", "Check out branch", and a dropdown menu. Below this, a green checkmark indicates "Pipeline #62596415 passed for d5d8f87e on !3 with feature/dynamic-env...". A message states "Deployed to review/feature/dynamic-env... 4 minutes ago" with a "View app" button. At the bottom, there is a "Merge" button and a checked checkbox for "Delete source branch", with the note "Fast-forward merge without a merge commit".

Ahora hemos extendido nuestro pipeline para ejecutarse con branches y merge requests. Lo que podemos ver ahora es que hemos creado un nuevo entorno de forma dinámica:



El nombre de este entorno se llama “review/feature/dynamic-environments” ya que ese fue el nombre que le pusimos a nuestro nuevo branch.