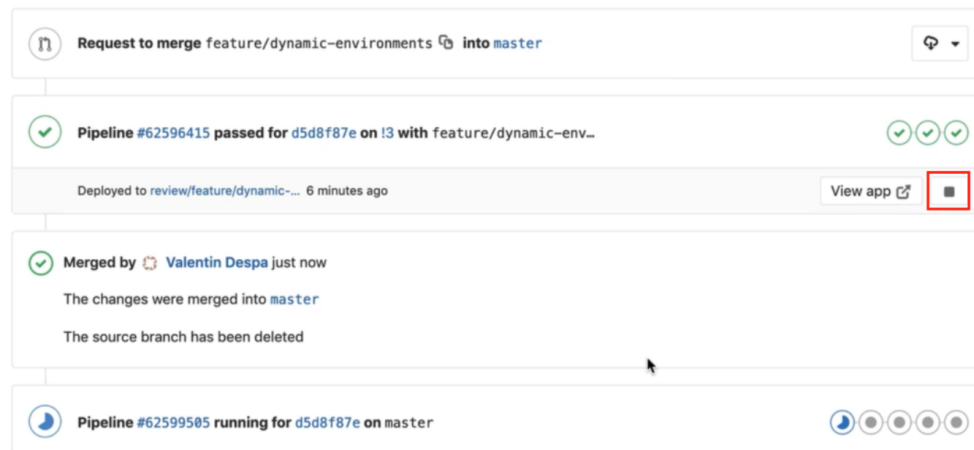


Destruyendo Entornos

Con esta nueva característica de crear y usar entornos dinámicamente, hemos creado también un problema. Una vez que el branch se ha mergeado, en realidad no hay necesidad de mantener ese entorno en ejecución. Por defecto, éste seguirá ejecutándose, peor tenemos la posibilidad de detenerlo:

Add dynamic environments



Pero este botón en realidad no funciona ya que no le hemos indicado a GitLab cómo detener el entorno. Esto es lo que haremos en esta lección, vamos a detener o destruir el entorno tan pronto como se haga el merge de nuestro branch.

Desde un punto de vista técnico, el problema que tenemos es que le hemos pedido a Surge que cree nuestro dominio, y hemos desplegado los archivos allí, pero ahora necesitamos una manera de decirle a Surge que ya no lo necesitamos. Para ello, solo tenemos que especificar el comando “surge teardown domainname.com”. Así que vamos a nuestro pipeline para crear un Job que nos permita destruir ese dominio:

Comencemos con la siguiente configuración:

```
67 | stop review:
68 |   stage: deploy review
69 |   variables:
70 |     GIT_STRATEGY: none
71 |   script:
72 |     - echo "Remove review app"
73 |   when: manual
74 |   environment:
75 |     name: review/$CI_COMMIT_REF_NAME
76 |     action: stop
```

Nombramos a nuestro nuevo Job “stop review”, el cual forma parte de nuestro Stage “deploy review”.

Ahora, queremos ejecutar este Job sol cuando el branch haya sido mergeado y posiblemente eliminado, por esta razón necesitamos “añadir” una variable preexistente llamada GIT_STRATEGY. Esta variable es necesaria para que este Job no intente clonar este branch, que puede ya no existir. Asignamos el valor “none” a nuestra variable:

```
67 | stop review:
68 |   stage: deploy review
69 |   variables:
70 |     GIT_STRATEGY: none
71 |   script:
72 |     - echo "Remove review app"
73 |   when: manual
74 |   environment:
75 |     name: review/$CI_COMMIT_REF_NAME
76 |   action: stop
```

Esta Variable la estamos definiendo dentro del Job “stop review”, por lo que solo tiene efecto dentro de este Job. También la variable GIT_STRATEGY es una variable preexistente que nos proporciona GitLab, y especificando este valor, GitLab entiende que no debe intentar clonar este branch.

Ahora, en nuestros scripts hay un par de cosas que tenemos que hacer. Primero, tenemos que instalar Surge:

```
67 | stop review:
68 |   stage: deploy review
69 |   variables:
70 |     GIT_STRATEGY: none
71 |   script:
72 |     - npm install --global surge
73 |   when: manual
74 |   environment:
75 |     name: review/$CI_COMMIT_REF_NAME
76 |   action: stop
```

A continuación, podemos añadir el comando para destruir el dominio:

```
67 | stop review:
68 |   stage: deploy review
69 |   variables:
70 |     GIT_STRATEGY: none
71 |   script:
72 |     - npm install --global surge
73 |     - surge teardown instazone-$CI_ENVIRONMENT_SLUG.surge.sh
74 |   when: manual
75 |   environment:
76 |     name: review/$CI_COMMIT_REF_NAME
77 |   action: stop
```

Adicionalmente, necesitamos establecer un enlace entre nuestros dos Jobs, el que despliega y el que destruye el dominio. Esto lo hacemos configurando el comando `on_stop: stop review` dentro de nuestro Job “deploy review”, de esta forma indicamos a nuestro Job de despliegue cuál será el Job de destrucción del entorno:

```
56  deploy review:
57    stage: deploy review
58    only:
59      - merge_requests
60    environment:
61      name: review/$CI_COMMIT_REF_NAME
62      url: http://instazone-$CI_ENVIRONMENT_SLUG.surge.sh
63      on_stop: stop review
64    script:
65      - npm install --global surge
66      - surge --project ./public --domain instazone-$CI_ENVIRONMENT_SLUG.surge.sh
67
68  stop review:
69    stage: deploy review
70    variables:
```

Ahora vamos a crear un nuevo branch para ver cómo funciona esto.

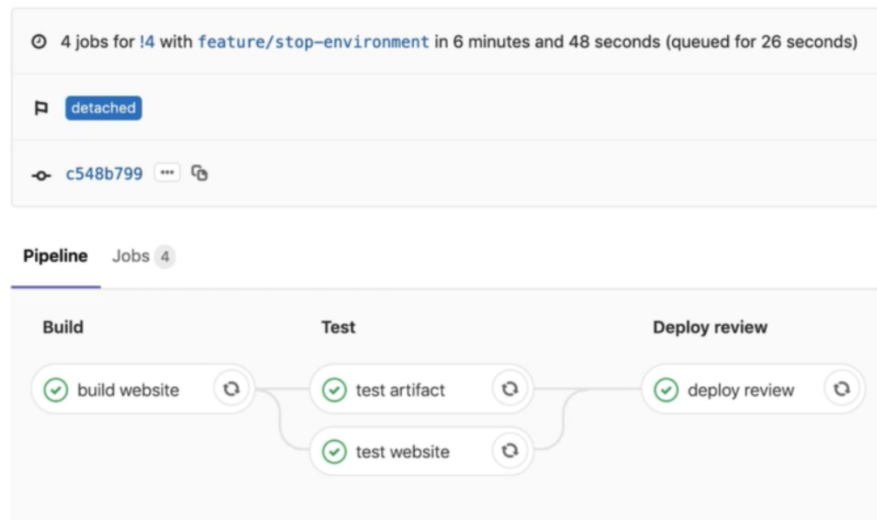
Ahora se ha ejecutado nuestro pipeline para nuestro merge request:

Stop environment

The screenshot shows a merge request interface. At the top, it says "Request to merge feature/stop-enviro... into master" with buttons for "Open in Web IDE", "Check out branch", and a dropdown menu. Below this, a pipeline status bar shows "Pipeline #62601919 passed for c548b799 on !4 with feature/stop-enviro..." with three green checkmarks. A progress bar indicates "Deploying to review/feature/stop-envi...". At the bottom, there are two main options: "Merge" (highlighted in green) and "Delete source branch" (checked). Below these options, it says "Fast-forward merge without a merge commit" and "You can merge this merge request manually using the command line".

Demos un vistazo a nuestros Jobs:

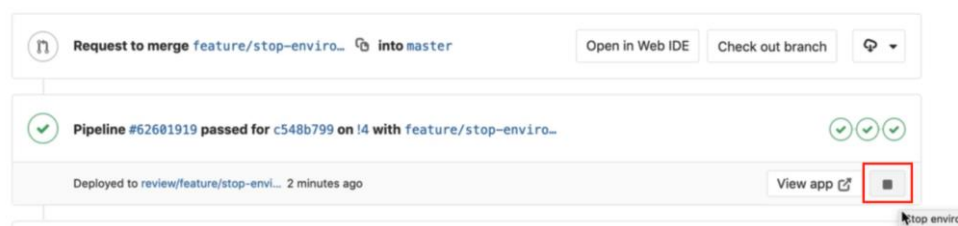
Stop environment



Podemos notar que no ha cambiado mucho. Todavía tenemos nuestros tres Stages, y además este Job “stop review” que se suponía que iba a destroy el entorno dinámico en realidad no aparece en nuestro pipeline.

Aun así, podríamos seguir adelante y destruir manualmente el entorno dinámico desde el botón:

Stop environment



Pero esto sería un paso manual adicional que realmente no queremos hacer. Queremos dejar que GitLab lo haga por nosotros, así que vamos a dar click en “Merge”:



Stop environment

This screenshot shows the history of the 'Stop environment' workflow. It includes a merge request, a successful pipeline run (#62601919) that was deployed to a review environment, and a merge by Valentin Despa. A new pipeline run (#62603986) is shown as running on the master branch.

Request to merge feature/stop-environment into master

Pipeline #62601919 passed for c548b799 on !4 with feature/stop-enviro... ✓✓✓

Deployed to review/feature/stop-envi... 4 minutes ago [View app](#)

Merged by Valentin Despa 1 minute ago

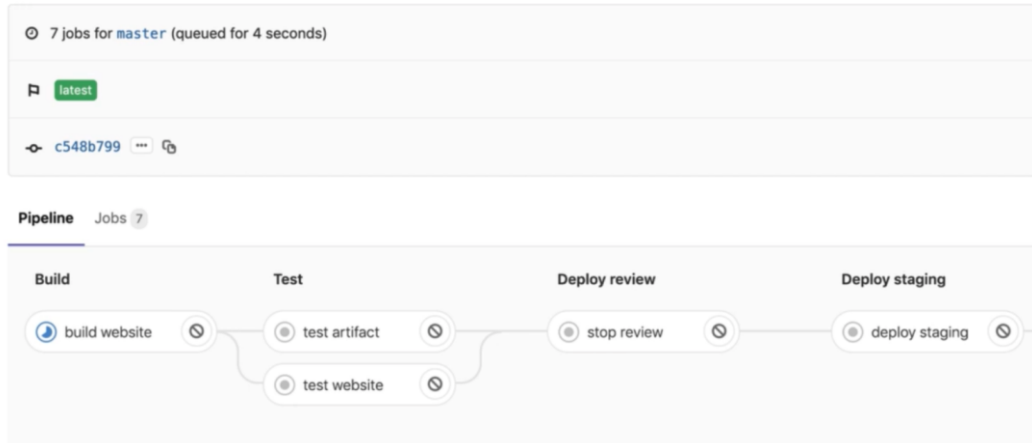
The changes were merged into master

The source branch has been deleted

Pipeline #62603986 running for c548b799 on master 🔄

Inicialmente no ha pasado mucho, pero ahora vamos a inspeccionar el pipeline en nuestro branch master:

Stop environment



Vemos ahora que nuestro pipeline tiene los Jobs “build” y “test” que ya conocemos, y tiene erróneamente el Job “deploy review”, esto ha sucedido porque olvidamos excluir la ejecución de este Job del branch master, así que vamos a corregirlo antes de continuar:

Así que lo que necesitamos hacer es agregar el flag “only” a nuestro Job “deploy review” para que solo se ejecute en los merge requests:

```

68   stop review:
69     stage: deploy review
70     only:
71     - merge_requests
72     variables:
73       GIT_STRATEGY: none
74     script:
75       - npm install --global surge
76       - surge teardown instazone-$CI_ENVIRONMENT_SLUG.surge.sh
77     when: manual
78     environment:
79       name: review/$CI_COMMIT_REF_NAME
80     action: stop

```

El pipeline del merge request se ha finalizado correctamente:

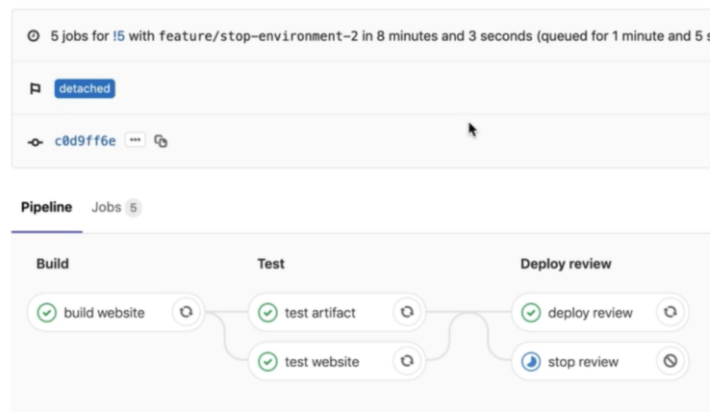
Fix stop environment

Ahora hagamos click en Merge:

Fix stop environment

Ahora vemos que, tan pronto como hemos hecho merge, el pipeline para este branch se ha iniciado de nuevo, y si revisamos a detalle los Jobs de este pipeline en ejecución, podemos ver que, el Job “deploy review” sigue ahí, finalizado, desde la ejecución anterior, pero se ha comenzado a ejecutar el nuevo Job “stop review”:

Fix stop environment



Así que, si ahora vamos a nuestro sitio web, en nuestro entorno dinámico, podremos ver que ya no está disponible.