

## Definir Variables

En esta lección veremos cómo podemos definir nuestras propias variables en GitLab CI.

Podemos ver que tenemos un nombre de dominio en varias partes de nuestro archivo:

```
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.s
50
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
```

Ya hemos utilizado variables en lecciones anteriores:

**Variables** ?

Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. Additionally, they will be masked by default so they are hidden in job logs, though they must match certain regexp requirements to do so. You can use environment variables for passwords, secret keys, or whatever you want. You may also add variables that are made available to the running application by prepending the variable key with `KBS_SECRET_`. [More information](#)

Type	Key	Value	State	Masked	Scope
Variable	SURGE_LOGIN	*****	Protected <input type="checkbox"/>	Masked <input type="checkbox"/>	<input type="radio"/>
Variable	SURGE_TOKEN	*****	Protected <input type="checkbox"/>	Masked <input type="checkbox"/>	<input type="radio"/>
Variable	Input variable ke	Input variable	Protected <input type="checkbox"/>	Masked <input checked="" type="checkbox"/>	<input type="radio"/>

Podemos configurar nuestras variables desde la interfaz de GitLab pero también desde dentro de nuestro pipeline.

Ahora, como muchas cosas, podemos definir variables para un solo Job, por ejemplo, si queremos reutilizar algo solamente en un Job. Pero el dominio se utiliza en varias partes, incluso en el Stage de producción, entonces en este caso es mejor definirlo a un nivel global.

Así que definamos dos variables fuera de los Job:

```
10  cache:
11    key: ${CI_COMMIT_REF_SLUG}
12    paths:
13      - node_modules/
14
15  variables:
16    STAGING_DOMAIN: instazone-staging.surge.sh
17    PRODUCTION_DOMAIN: instazone-staging.surge.sh
18
```

Ahora vamos a modificar nuestros Jobs para implementar estas variables:

```
46  deploy staging:
47    stage: deploy staging
48    environment:
49      name: staging
50      url: http://$STAGING_DOMAIN
51    script:
52      - npm install --global surge
53      - surge --project ./public --domain $STAGING_DOMAIN
```

```
55  deploy production:
56    stage: deploy production
57    environment:
58      name: production
59      url: $PRODUCTION_DOMAIN
60    script:
61      - npm install --global surge
62      - surge --project ./public --domain $PRODUCTION_DOMAIN
```

Así que hagamos commit de nuestros cambios y veamos si nuestro pipeline aún funciona:

#### Use variables for domain names

