CENTRE ESPECÍFIC D´EDUCACIÓ A DISTÀNCIA DE LA COMUNITAT VALENCIANA

UNIT 7.
ACTIVITY: CI/CD USING GITLAB ON HEROKU

Web Applications Deployment
CFGS DAW

Important: this activity is not mandatory and does not compute for the final grade.

Importante: esta actividad no es obligatoria y no cuenta para la nota final.

Author: Carlos Cacho López

Reviewed by: Lionel Tarazón Alcocer

Reviewed by: Pau Miñana

2020/2021

License

Nomenclature

During this unit we are going to use special symbols to distinct some important elements.

This symbols are:

Important

Attention

Interesting

INDEX

UT07. CONTINUOUS INTEGRATION
ACTIVITY: CI/CD USING GITLAB ON HEROKU

1. INTRODUCTION

In this activity we are going to practice how to deploy a Node.js application using CI/CD (continuous integration/continuous deployment) with GitLab on Heroku.

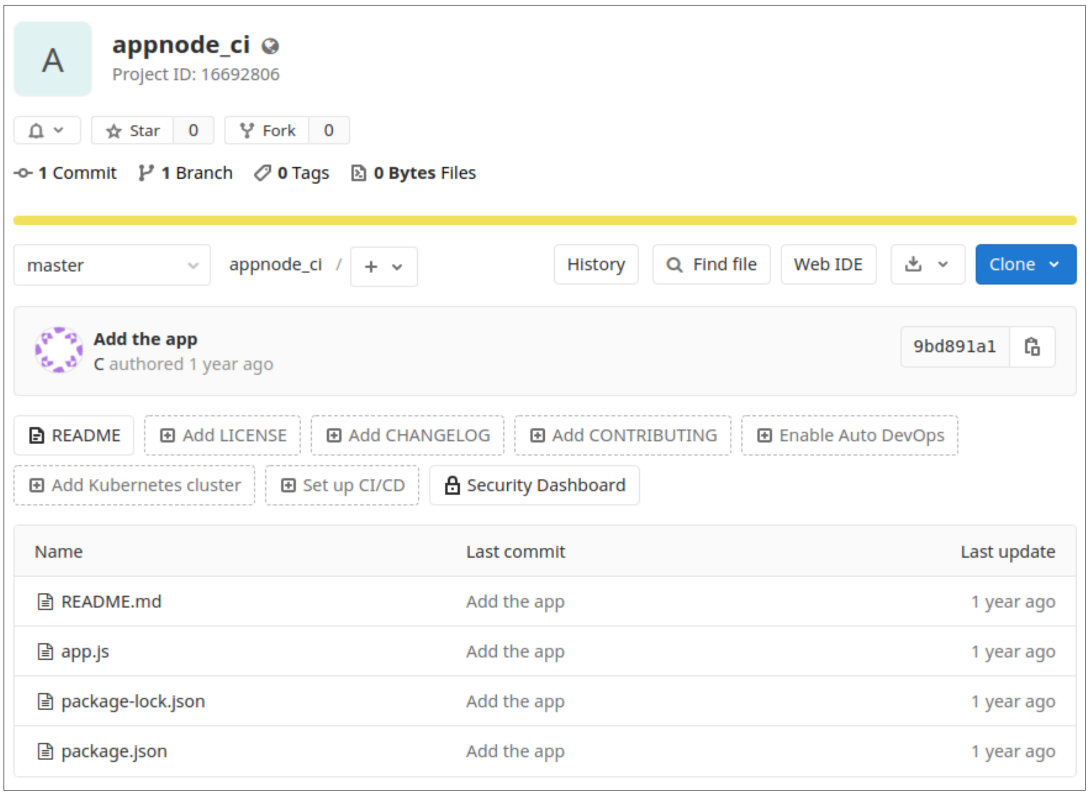
The goal is to configure a GitLab repository with a web application deployed in Heroku, so that every time we do changes to the repository (via git push) a pipeline of jobs will automatically start running (tipically code compilation and tests). If those jobs validate everything works properly, the changes will be deployed straight away to the Heroku app.

You can use your own physical machine or in a virtual machine. I will do it using Ubuntu. Remember you need an Internet connection.

2. CI/CD USING GITLAB ON HEROKU

First of all we need a GitLab repository with the Node.js application we want to use. For this activity you can use this repository: https://gitlab.com/lionel_ceedcv/appnode_ci

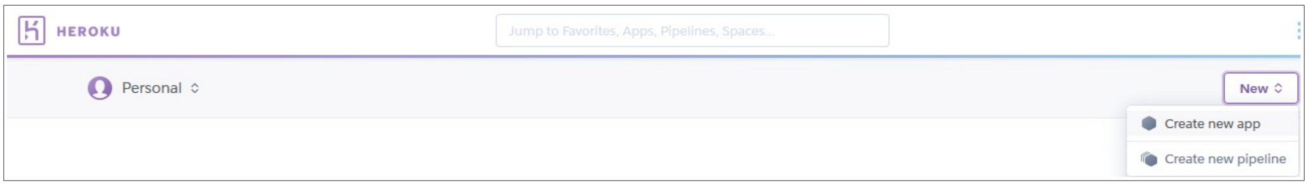
You will need your own repository (to do CI/CD you need privileged access). So you should download the repository above, create you own Gitlab repository and upload/push it there. Another option would be to create a fork of the repository, but as Lionel is not the working here this year I recommend against it.



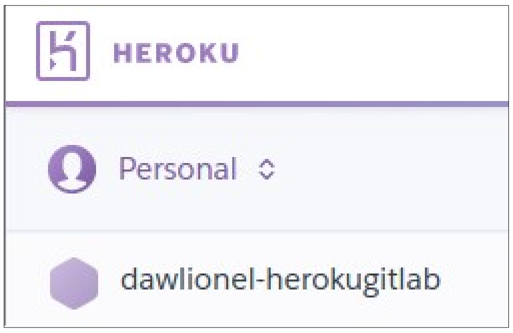
Add the app
C authored 1 year ago
9bd891a1

Name	Last commit	Last update
README.md	Add the app	1 year ago
app.js	Add the app	1 year ago
package-lock.json	Add the app	1 year ago
package.json	Add the app	1 year ago

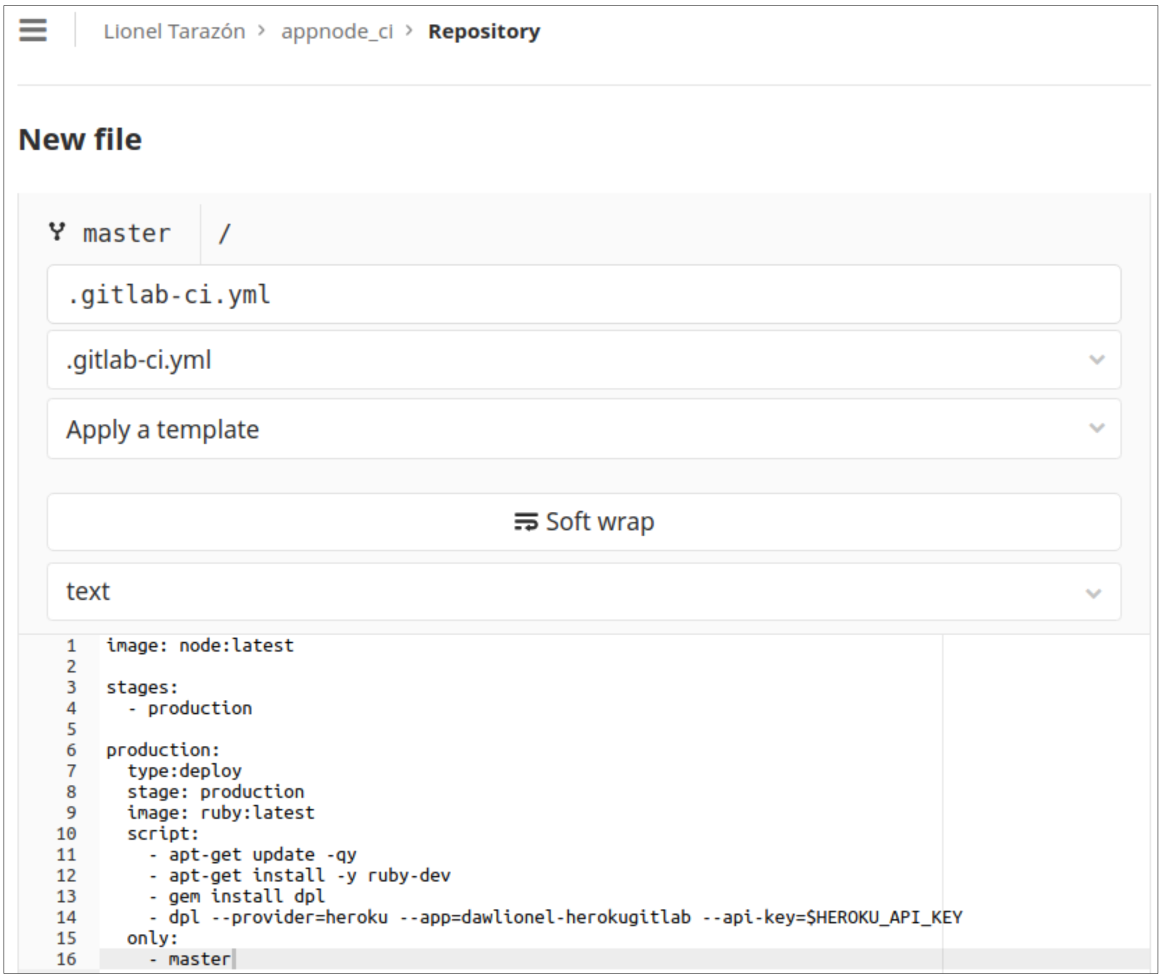
Now we have to create a new application in Heroku clicking on ***New > create new app***:



For instance, we will call it *dawlionel-herokugitlab*:



Now, in our GitLab repository, we need to create and commit a new file called *.gitlab-ci.yml* where we will configure the production stage of our CI pipeline:



```
image: node:latest

stages:
  - production

production:
  type:deploy
  stage: production
  image: ruby:latest
  script:
    - apt-get update -qy
    - apt-get install -y ruby-dev
    - gem install dpl
    - dpl --provider=heroku --app=dawlionel-herokugitlab --api-key=$HEROKU_API_KEY
  only:
    - master
```

A *YAML* (*Ain't Markup Language*) file is a human-readable data serialization language. It is commonly used for configuration files, but could be used in many applications where data is being stored (e.g. debugging output) or transmitted. Its extensions are *.yaml* or *.yml*.

The explanation of the lines is:

image: node:latest

Indicates the Docker image to use.

```
stages:
  - production
```

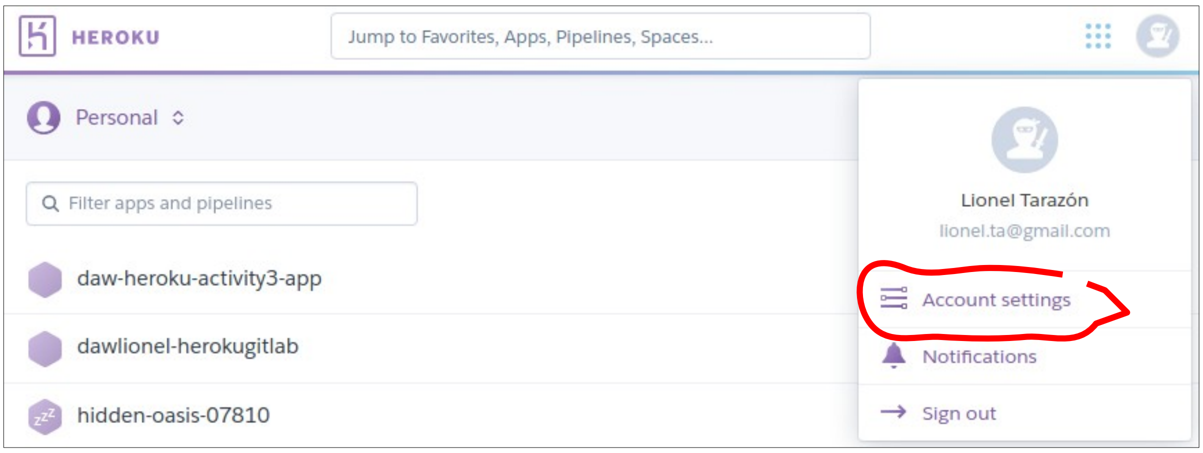
Define stages that the jobs will use them. In this case we define one stage called *production*.

```
production:
  type: deploy
  stage: production
  image: ruby:latest
  script:
    - apt-get update -qy
    - apt-get install -y ruby-dev
    - gem install dpl
    - dpl --provider=heroku --app=dawlionel-herokugitlab --api-key=$HEROKU_API_KEY
  only:
    - master
```

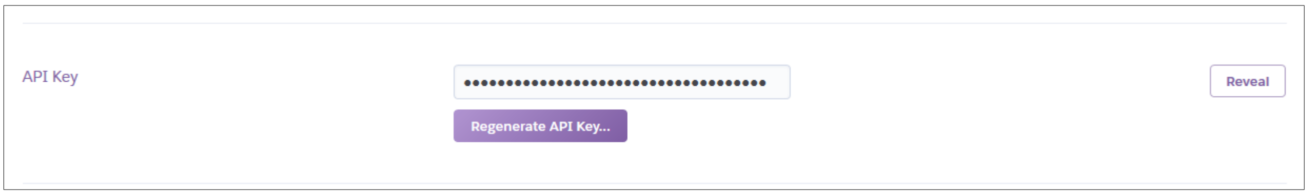
The *stage* production used to deploy uses the Docker image *ruby* and runs this four *scripts* to deploy the app to Heroku. It is important to see that in the provider option we have to write heroku, in app our app name (*dawlionel-herokugitlab* in this case) and in the api-key the environment variable we will create later. Finally we specify that we *only* work with the *master* branch.

You can find more information about *.gitlab-ci.yml* in this official documentation: https://docs.gitlab.com/ee/ci/yaml/

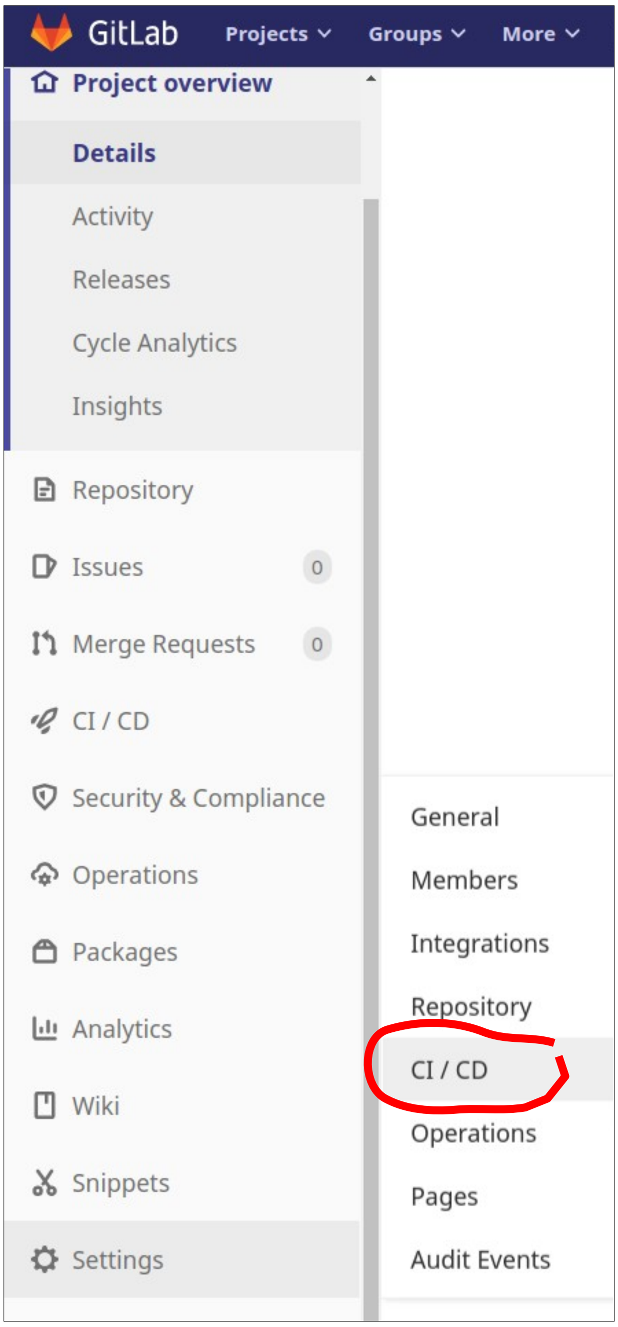
Now we have to store our Heroku API key in GitLab. We can find the key in ***Profile > Account settings***:



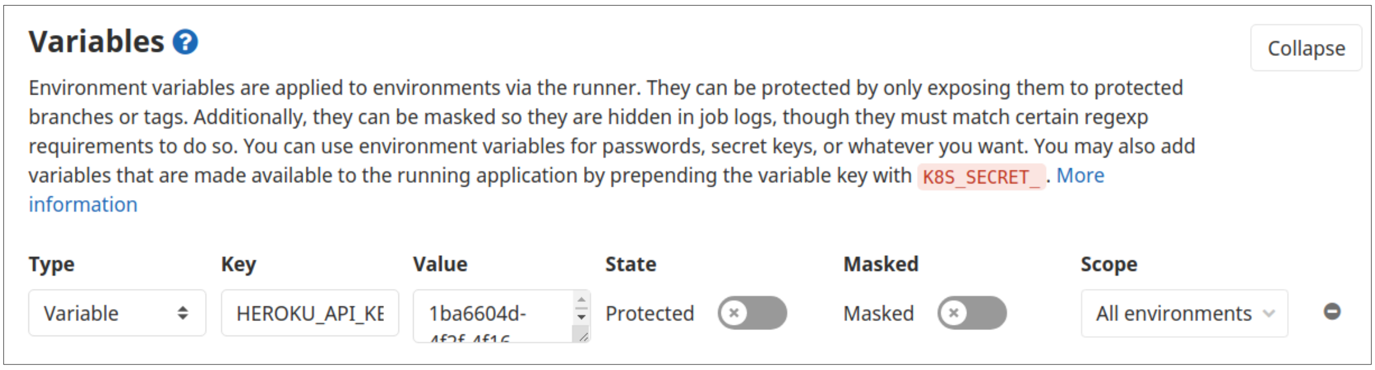
And in *API Key* section we can *Reveal* and copy it:



Now in the GitLab repository go to ***Settings > CI/CD***:



And in the *Variables* section create a new variable called **HEROKU_API_KEY** and paste the HerokuAPI Key as its value. Then click on ***Save variables***.



Variables

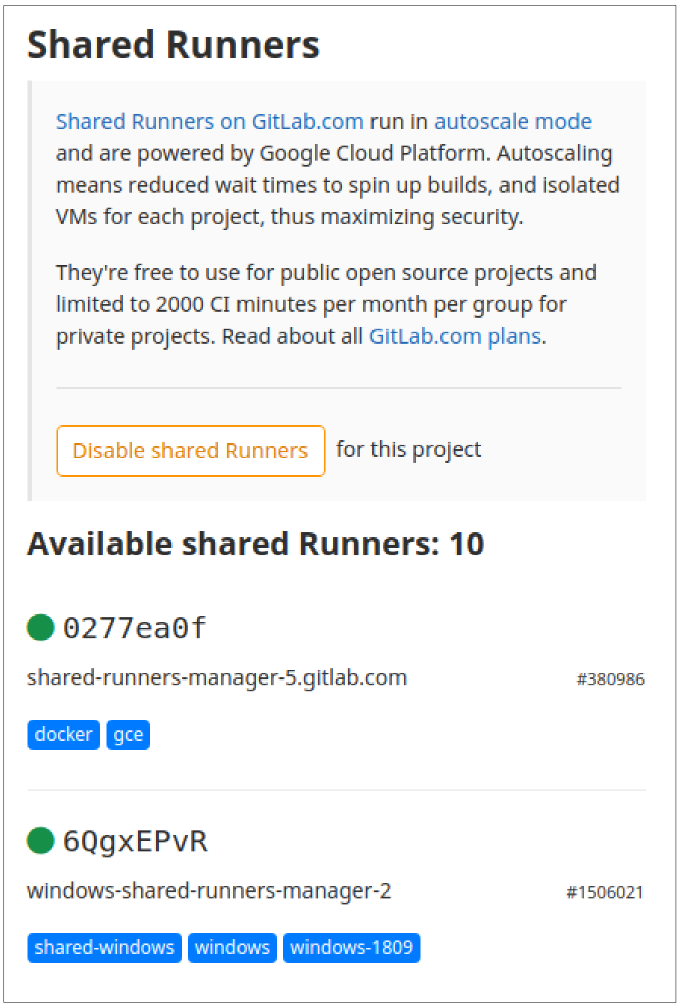
Environment variables are applied to environments via the runner. They can be protected by only exposing them to protected branches or tags. Additionally, they can be masked so they are hidden in job logs, though they must match certain regexp requirements to do so. You can use environment variables 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 `K8S_SECRET_`. More information

Type	Key	Value	State	Masked	Scope
Variable	HEROKU_API_KE	1ba6604d-4f2f-4f16	Protected	Masked	All environments

Now lets go to ***Settings > CI/CD*** (as before) and take a look at the ***Runners*** section. Runners are the machines that run the code in a CD/CI pipeline. In our case, they will run the *.gitlab-ci.yml* file script.

We are going to use Shared Runners. These are virtual machines provided by Google Cloud Platform that we can use for free (up to a maximum of 2000 minutes of CI/CD per month).

Shared Runners should already be activated.



Shared Runners

Shared Runners on GitLab.com run in autoscale mode and are powered by Google Cloud Platform. Autoscaling means reduced wait times to spin up builds, and isolated VMs for each project, thus maximizing security.

They're free to use for public open source projects and limited to 2000 CI minutes per month per group for private projects. Read about all GitLab.com plans.

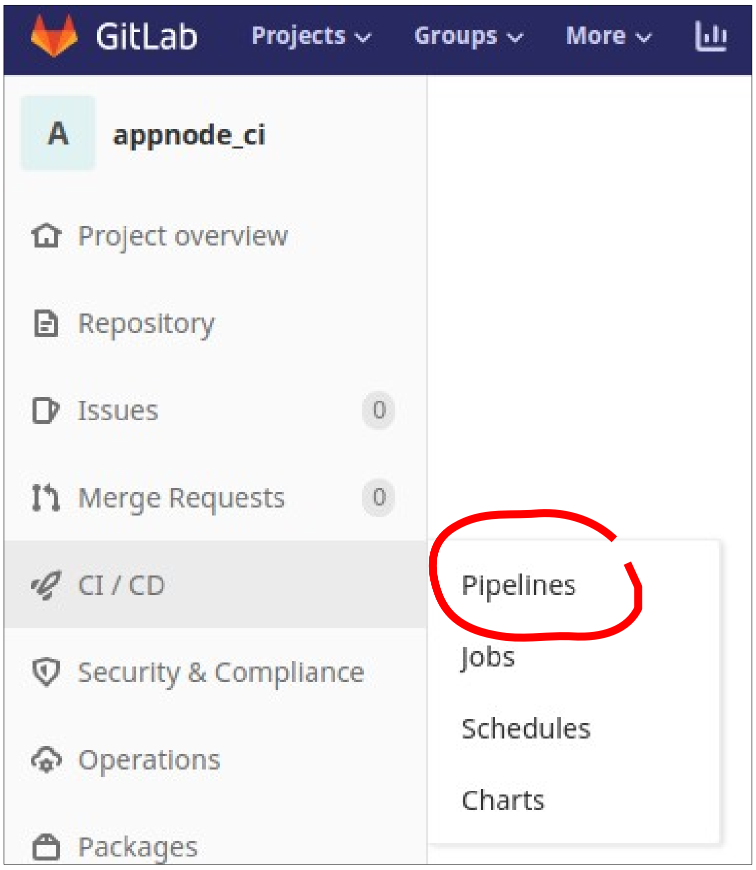
Disable shared Runners for this project

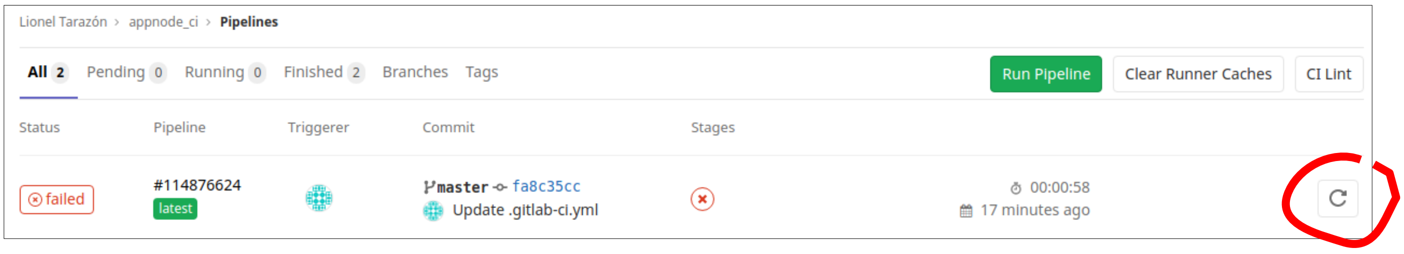
Available shared Runners: 10

● `0277ea0f`
shared-runners-manager-5.gitlab.com #380986
docker gce

● `6QgxEPvR`
windows-shared-runners-manager-2 #1506021
shared-windows windows windows-1809

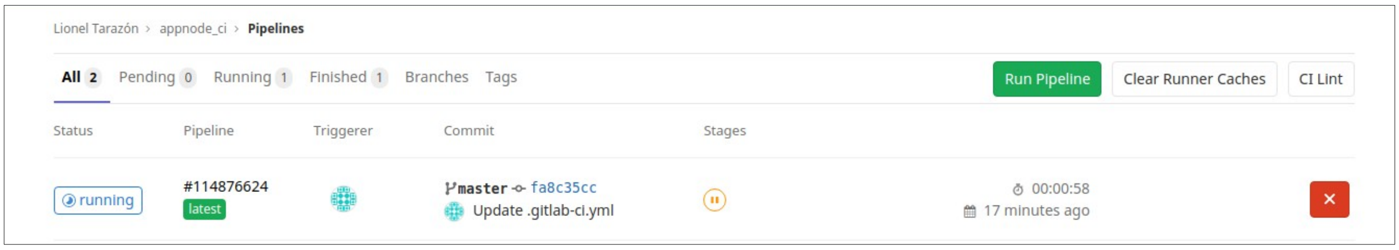
Now in ***CI/CD -> Pipelines*** we can see that it failed because the HEROKU_API_KEY was not set:




Lionel Tarazón › appnode_ci › Pipelines
All 2 Pending 0 Running 0 Finished 2 Branches Tags
Run Pipeline
Clear Runner Caches
CI Lint

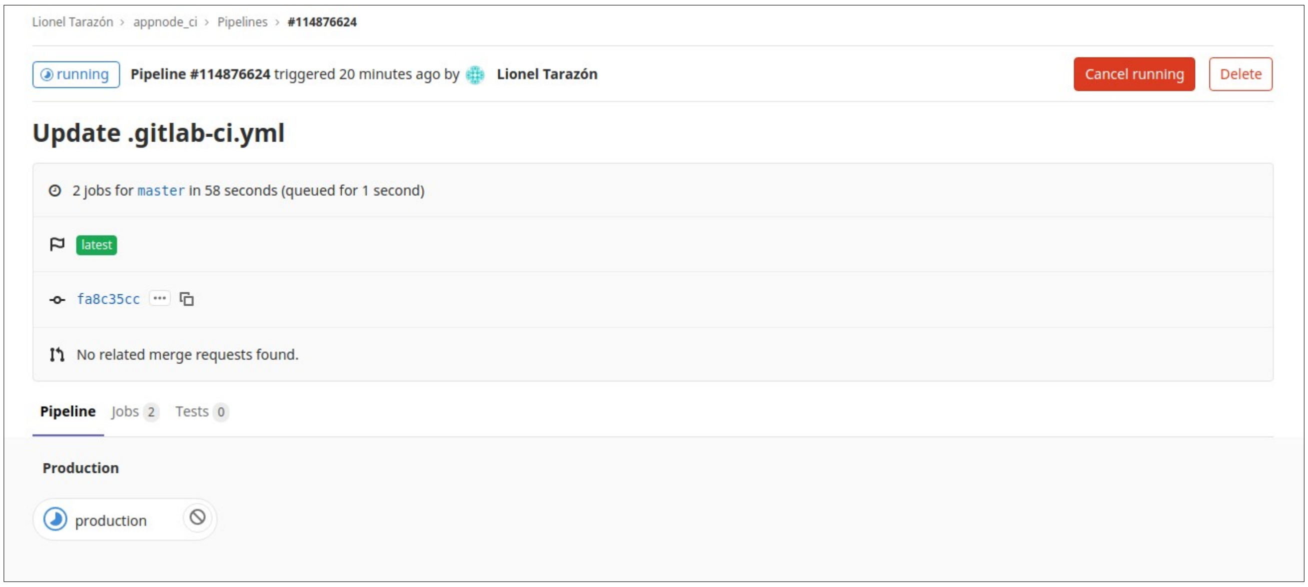
Status	Pipeline	Triggerer	Commit	Stages		
failed	#114876624 latest		master fa8c35cc Update .gitlab-ci.yml		00:00:58 17 minutes ago	

So we have to click on the refresh icon and it will start running.


Lionel Tarazón › appnode_ci › Pipelines
All 2 Pending 0 Running 1 Finished 1 Branches Tags
Run Pipeline
Clear Runner Caches
CI Lint

Status	Pipeline	Triggerer	Commit	Stages		
running	#114876624 latest		master fa8c35cc Update .gitlab-ci.yml		00:00:58 17 minutes ago	

If we click on 'running' we can see the Pipeline:


Lionel Tarazón > appnode_ci > Pipelines > **#114876624**

running **Pipeline #114876624** triggered 20 minutes ago by **Lionel Tarazón**

Update .gitlab-ci.yml

2 jobs for `master` in 58 seconds (queued for 1 second)

latest

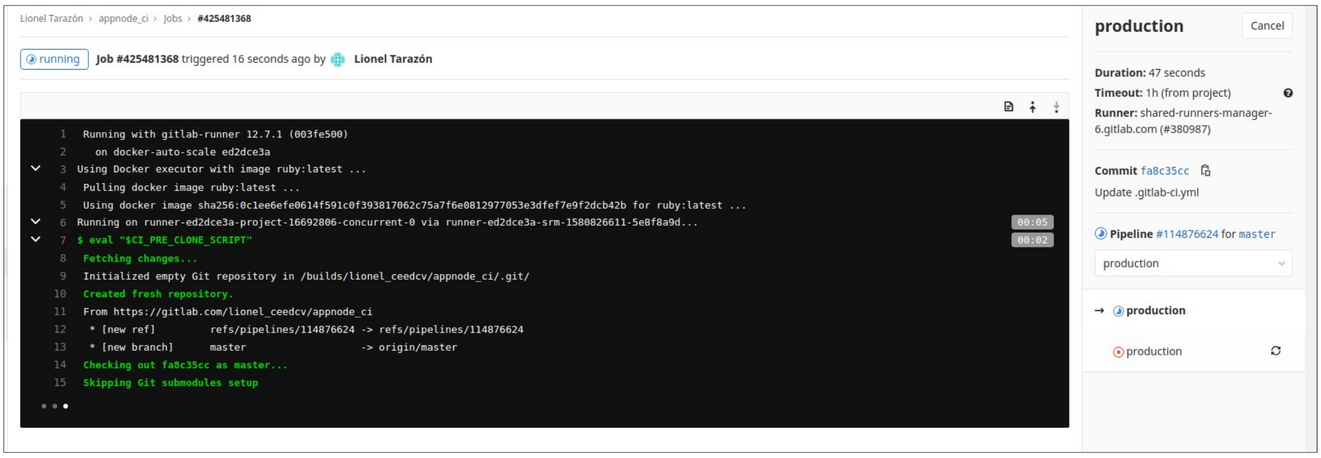
`fa8c35cc`

No related merge requests found.

Production

production

And if we click on 'production' we can see what is happening:


Lionel Tarazón > appnode_ci > Jobs > **#425481368**

running **Job #425481368** triggered 16 seconds ago by **Lionel Tarazón**

```
Running with gitlab-runner 12.7.1 (003fe500)
  on docker-auto-scale ed2dce3a
Using Docker executor with image ruby:latest ...
Pulling docker image ruby:latest ...
Using docker image sha256:0c1ee6efe0614f591c0f393817062c75a7f6e0812977053e3dfef7e9f2dcb42b for ruby:latest ...
Running on runner-ed2dce3a-project-16692806-concurrent-0 via runner-ed2dce3a-srm-1580826611-5e8f8a9d...      00:05
$ eval "$CI_PRE_CLONE_SCRIPT"                                                                                00:02
Fetching changes...
Initialized empty Git repository in /builds/lionel_ceedcv/appnode_ci/.git/
Created fresh repository.
From https://gitlab.com/lionel_ceedcv/appnode_ci
 * [new ref]         refs/pipelines/114876624 -> refs/pipelines/114876624
 * [new branch]      master                   -> origin/master
Checking out fa8c35cc as master...
Skipping Git submodules setup
```

production

Duration: 47 seconds
Timeout: 1h (from project)
Runner: shared-runners-manager-6.gitlab.com (#380987)

Commit `fa8c35cc`
Update .gitlab-ci.yml

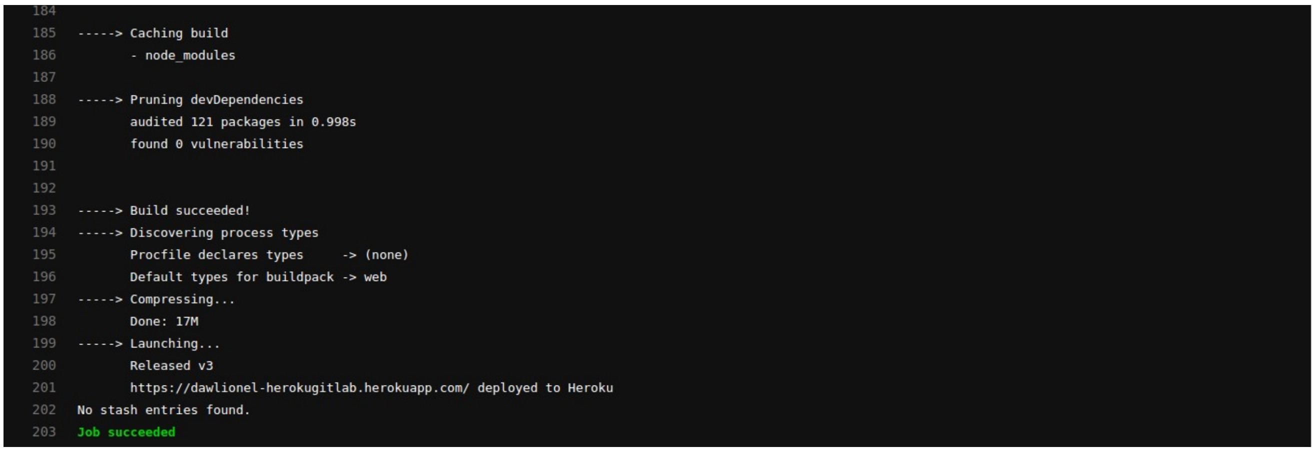
Pipeline #114876624 for `master`

production

→ **production**

production

If everything works correctly it will show the message "Job succeeded". The app hass been deployed to Heroku!

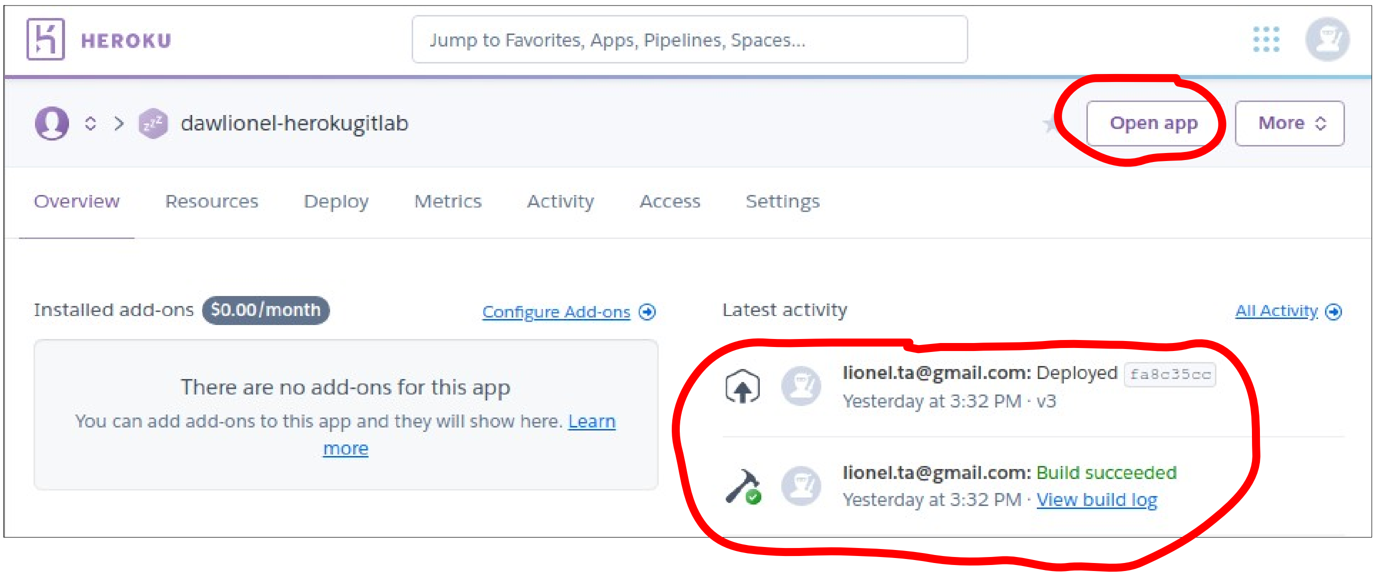

```

-----> Caching build
       - node_modules

-----> Pruning devDependencies
       audited 121 packages in 0.998s
       found 0 vulnerabilities


-----> Build succeeded!
-----> Discovering process types
       Procfile declares types     -> (none)
       Default types for buildpack -> web
-----> Compressing...
       Done: 17M
-----> Launching...
       Released v3
       https://dawlionel-herokugitlab.herokuapp.com/ deployed to Heroku
No stash entries found.
Job succeeded
```

Go to Heroku and select the app. Notice the deployment will appear under “Latest Activity”. Click on ***Open app*** to test it.

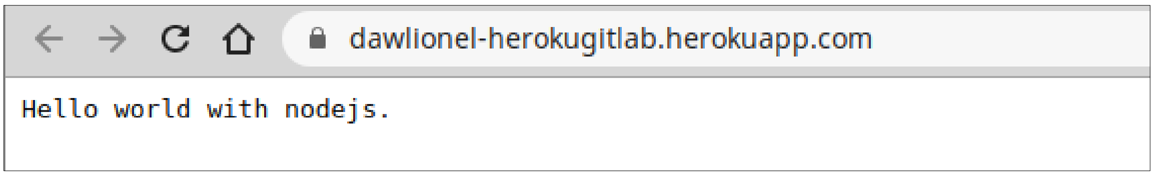


There are no add-ons for this app
You can add add-ons to this app and they will show here. Learn more

lionel.ta@gmail.com: Deployed fa8c35cc
Yesterday at 3:32 PM · v3

lionel.ta@gmail.com: Build succeeded
Yesterday at 3:32 PM · View build log

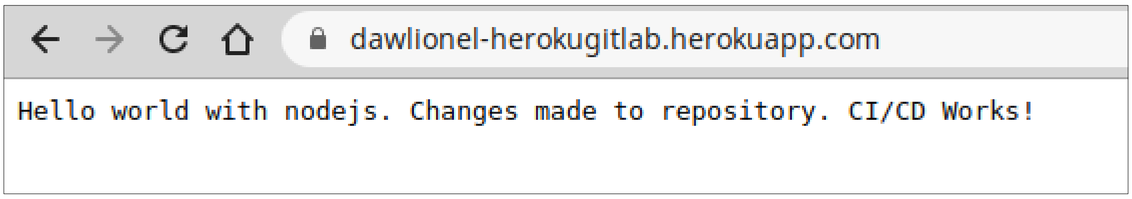
And we will see our app online :)



```
Hello world with nodejs.
```

If we make changes to the code repository and commit-push to GitLab, the CI/CD pipeline will start and automatically deploy it to Heroku!

```
lionel@lenovo-mint ~/appnode_ci $ gedit app.js
lionel@lenovo-mint ~/appnode_ci $ git add app.js
lionel@lenovo-mint ~/appnode_ci $ git commit -m "message changed"
[master 27faf46] message changed
 1 file changed, 1 insertion(+), 1 deletion(-)
lionel@lenovo-mint ~/appnode_ci $ git push origin master
Username for 'https://gitlab.com': lionel_ceedcv
Password for 'https://lionel_ceedcv@gitlab.com':
Counting objects: 3, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 326 bytes | 0 bytes/s, done.
Total 3 (delta 2), reused 0 (delta 0)
To https://gitlab.com/lionel_ceedcv/appnode_ci.git
   fa8c35c..27faf46  master -> master
```



```
Hello world with nodejs. Changes made to repository. CI/CD Works!
```