COUNTER PYTHON APP

Deploy App

Date : 18/03/2024

Zitouni Iptissem, Léa Perin

Table of contents :

I. Introduction

1.Project objectives

Our project aims to automate the deployment of our web application. To do this, we will use different technologies : gitlab and docker.

So we will see step by step how to do this.

1.2 Context of our infrastructure

3 virtuals machines :

- gitlab-instance : were we create our Ci/CD pipeline
- gitlab-runner : were this run
- dev-srv : were the application will run

II. **Environment**: Dockerfile, git and runner configuration

1.DockerFile

Dockerfile

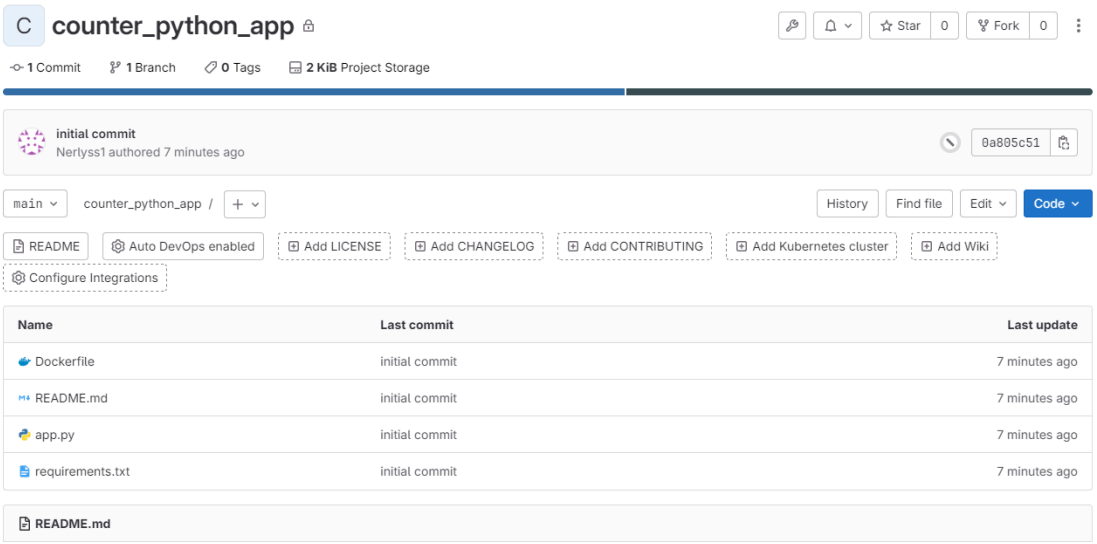
```
FROM python:3.10-alpine
WORKDIR /code
ENV FLASK_APP=app.py
ENV FLASK_RUN_HOST=0.0.0.0
RUN apk add --no-cache gcc musl-dev linux-headers
# copy the requirements.txt into the container with the same name
COPY requirements.txt ./requirements.txt
RUN pip install -r requirements.txt
EXPOSE 5000
# TODO: copy all from the current local folder to the containner
COPY . .
CMD ["flask", "run"]
```

2. Push our project gitlab instance

Once the files of our application are up to date, we will push them in gitlab :

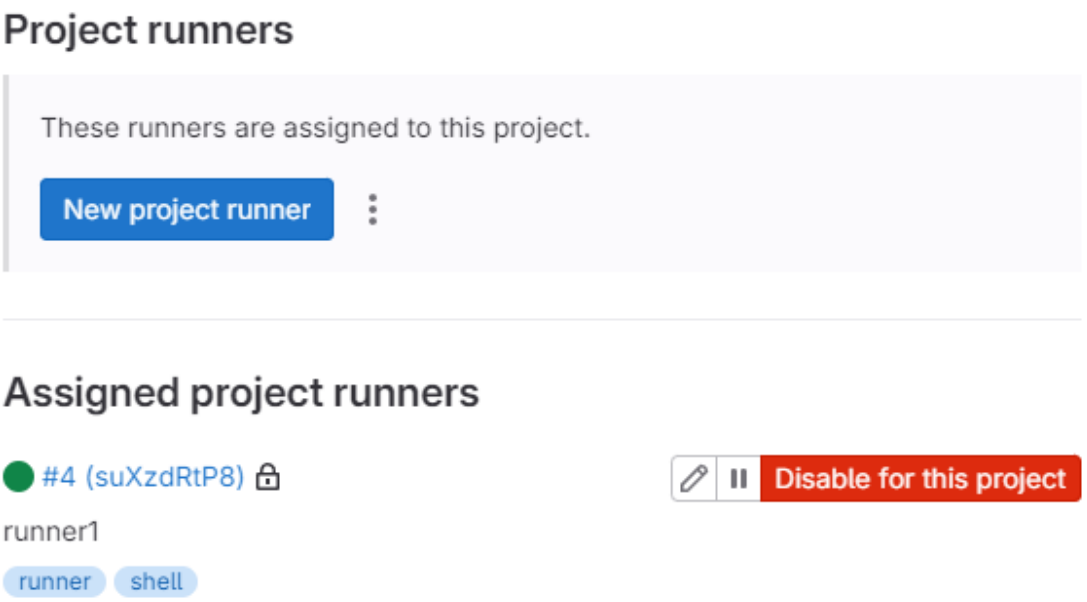
```
PS C:\Ynov\Devops\counter_pyhton_app> git init --initial-branch=main
Initialized empty Git repository in C:/Ynov/Devops/counter_pyhton_app/.git/
PS C:\Ynov\Devops\counter_pyhton_app> git remote add origin http://192.168.198.142/root/counter_python_app.git
PS C:\Ynov\Devops\counter_pyhton_app> git add .
PS C:\Ynov\Devops\counter_pyhton_app> git commit -m "initial commit"
[main (root-commit) 0a805c5] initial commit
 create mode 100644 Dockerfile
 create mode 100644 README.md
 create mode 100644 app.py
 create mode 100644 requirements.txt
PS C:\Ynov\Devops\counter_pyhton_app> git push --set-upstream origin main
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (6/6), 1.50 KiB | 1.50 MiB/s, done.
Total 6 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote:
remote:
remote: To configure the remote, run:
remote:   git remote add origin http://192.168.198.142/root/counter_python_app.git
remote:
remote: To view the project, visit:
remote:   http://192.168.198.142/root/counter_python_app
remote:
remote:
remote:
To http://192.168.198.142/root/counter_python_app.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.
PS C:\Ynov\Devops\counter_pyhton_app>
```

Result :



3.Add runner to our instance :

admin area > ci cd > runner > assign the runner to our project



III. **Automation**: CI/CD file

1. Build-image step :

This will create a docker images

```
build-image-job:
  stage: build-image
  script:
    - docker build -t ynov_lea_ipti_counter_app_img .
    - docker images
```

When we run our pipeline, the image is well created :

```
$ docker images
REPOSITORY                          TAG       IMAGE ID       CREATED          SIZE
ynov_lea_ipti_counter_app_img       latest    964a575f17b8   1 second ago     224MB
```

2. Publish-image step :

This step will push our image into dockerhub:

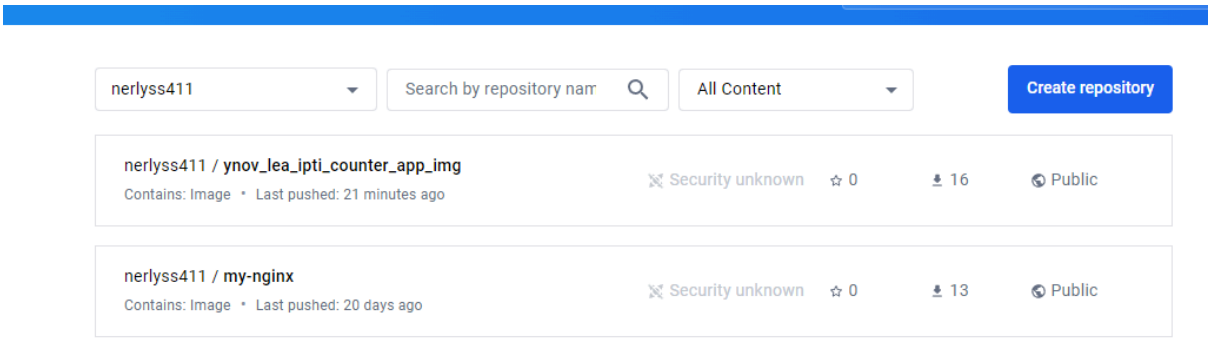
We create variables in order not to disclose our personal information : settings > cicd > variable

```
publish-image-job:
  stage: publish-image
  script:
    - docker tag ynov_lea_ipti_counter_app_img nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA
    - echo "$DOCKERHUB_PASSWORD" | docker login --username "$DOCKERHUB_LOGIN" --password-stdin
    - docker push nerlyss411/ynov_lea_ipti_counter_app_img:$
```

When pipeline is running :

```
Running with gitlab-runner 16.8.0 (c72a09b6)
  on runner1 suXzdRtP, system ID: s_6210be06e6c2
Preparing the "shell" executor
Using Shell (bash) executor...
Preparing environment
Running on pcdely...
Getting source from Git repository
Fetching changes with git depth set to 20...
Reinitialized existing Git repository in /home/gitlab-runner/builds/suXzdRtP/0/root/counter_python_app/.git/
Checking out 4d522f7f as detached HEAD (ref is main)...
Skipping Git submodules setup
Executing "step_script" stage of the job script
$ echo "$DOCKERHUB_PASSWORD" | docker login --username "$DOCKERHUB_LOGIN" --password-stdin
WARNING! Your password will be stored unencrypted in /home/gitlab-runner/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store
Login Succeeded
$ docker tag ynov_lea_ipti_counter_app_img nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA
$ docker push nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA
The push refers to repository [docker.io/nerlyss411/ynov_lea_ipti_counter_app_img]
5d571e7ab619: Preparing
f370f404a3ea: Preparing
3866c8c7b9e6: Preparing
cfbe553f9f6b: Preparing
5dc6cd483cfe: Preparing
```

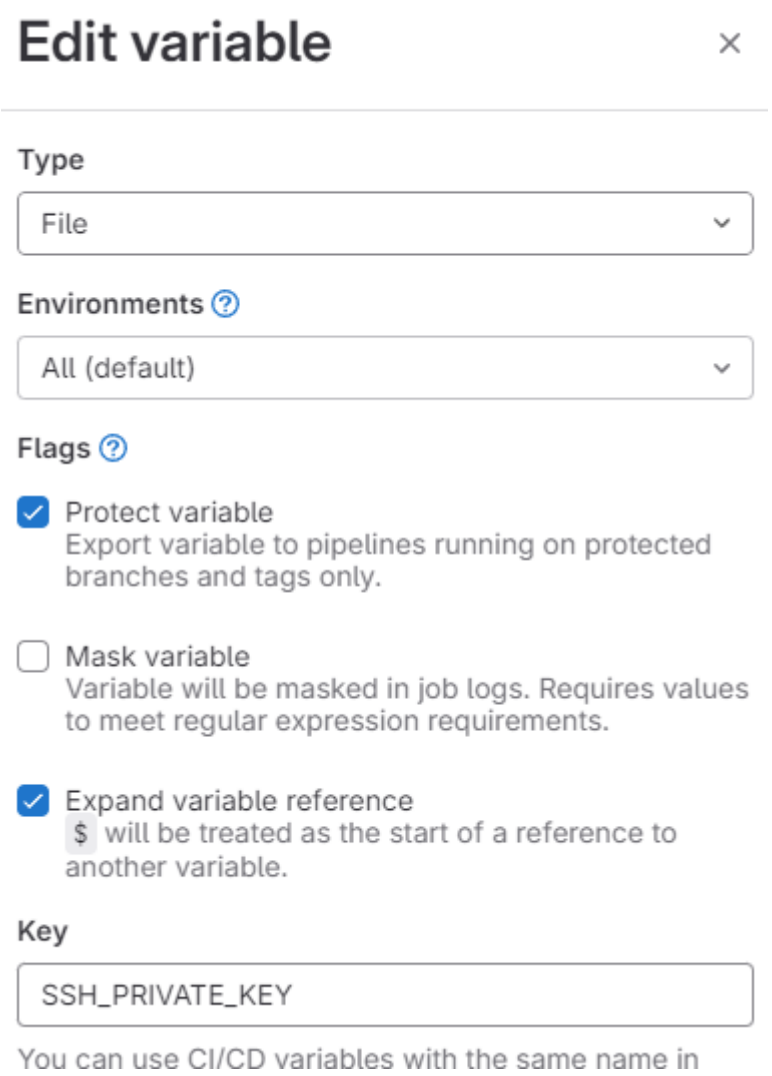
```
Login Succeeded
$ docker tag ynov_lea_ipti_counter_app_img nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA
$ docker push nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA
The push refers to repository [docker.io/nerlyss411/ynov_lea_ipti_counter_app_img]
5d571e7ab619: Preparing
f370f404a3ea: Preparing
3866c8c7b9e6: Preparing
cfbe553f9f6b: Preparing
5dc6cd483cfe: Preparing
8bf30651cf83: Preparing
9da37f712481: Preparing
9869e4387cac: Preparing
4c9c2b9681ab: Preparing
```



3. Deploy-app step :

- SSH Connexion : gitlab-runner should access to dev-srv.

First of all, we create a file in variable who contains our key.

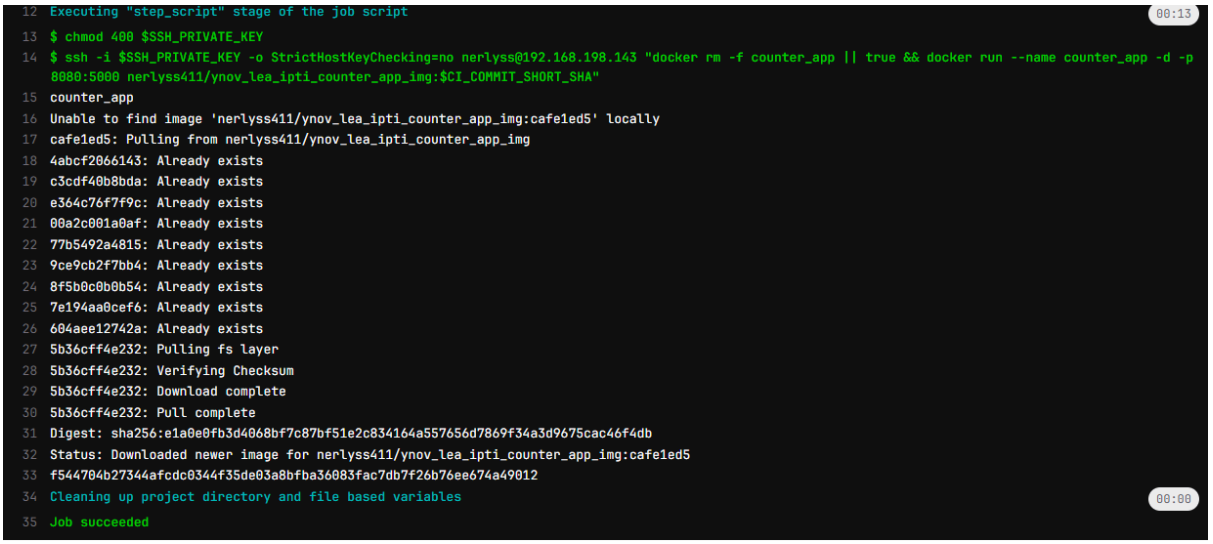


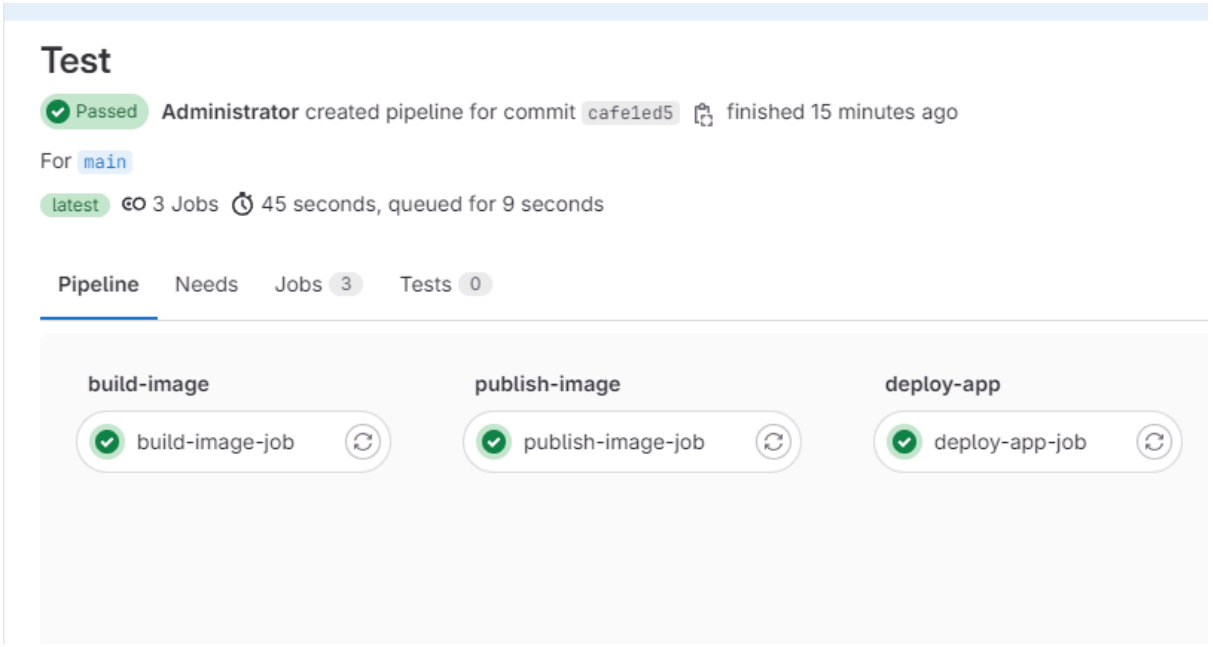
This step will allow to deploy our site: we will connect in SSH using the create variable previously. Then we delete the container if it exists then we run our new container.

```
deploy-app-job:
  stage: deploy-app
  script:
```

```
    - chmod 400 $SSH_PRIVATE_KEY
    - ssh -i $SSH_PRIVATE_KEY -o StrictHostKeyChecking=no
nerlyss@192.168.198.143 "docker rm -f counter_app || true && docker run
--name counter_app -d -p 8080:5000
nerlyss411/ynov_lea_ipti_counter_app_img:$CI_COMMIT_SHORT_SHA"
```

Once the pipeline is launched :





Interface web :