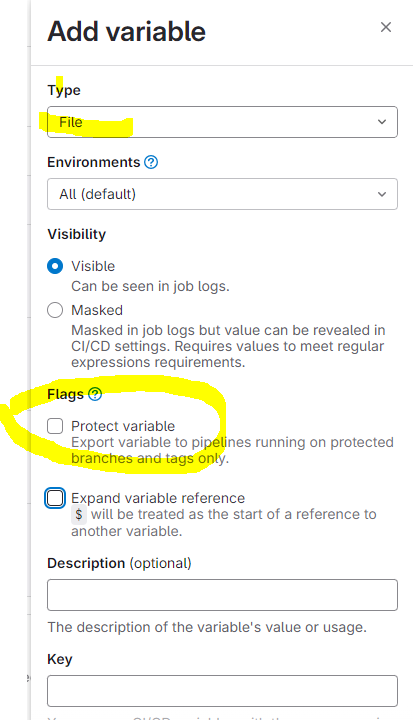
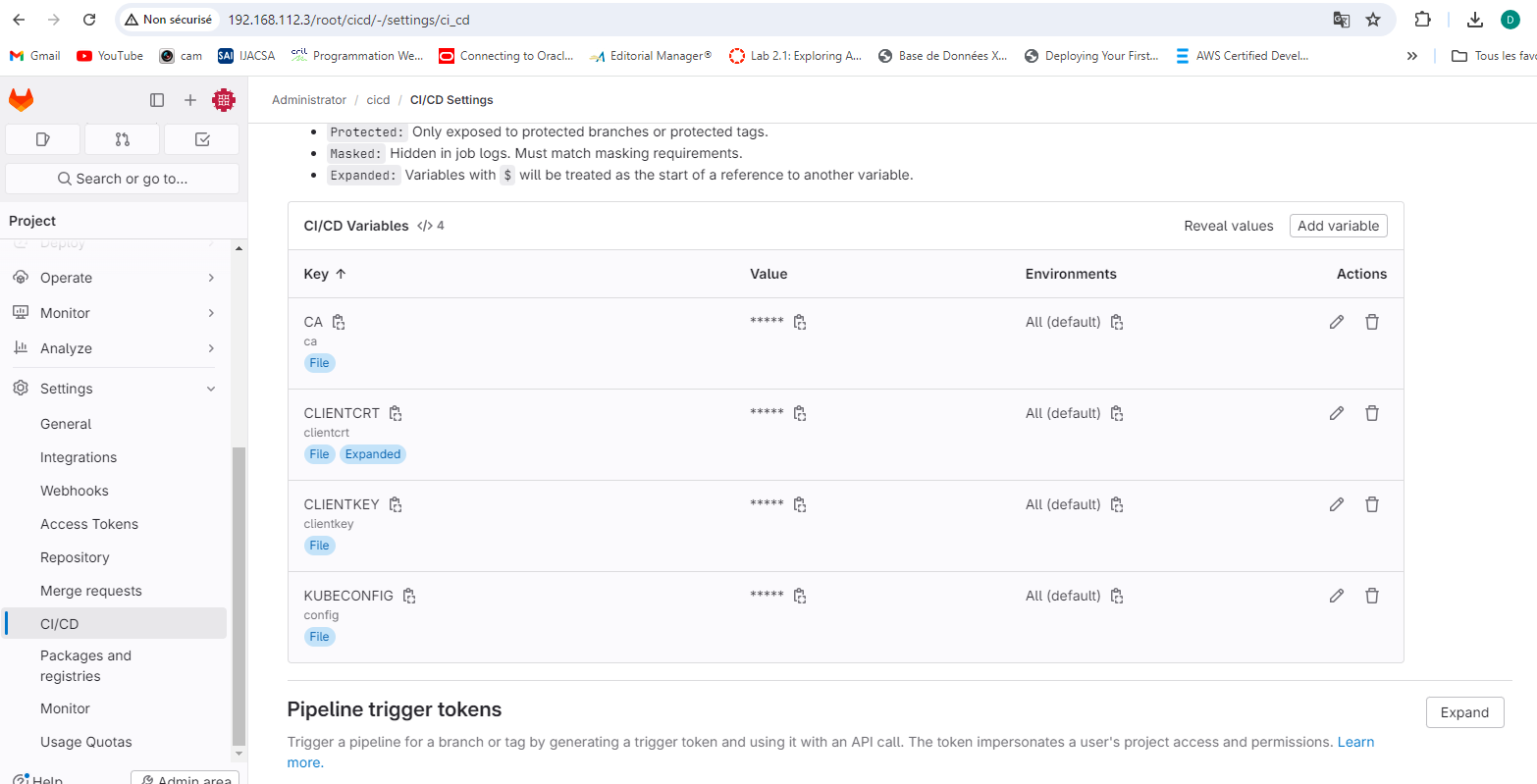
1. Créer des variables de type **file** et décochez « Protect Variable et expand variable reference :



1. **Les variables CA , KUBECONFIG, CLIENTCRT et CLIENTKEY on va les utiliser dans .gitlab-ci.yml**



1. **Les valeurs de chaque variable**

* **La valeur de KUBECONFIG : modifier l ip de minikube**

apiVersion: v1

clusters:

- cluster:

server: https://192.168.49.2:8443

name: minikube

contexts:

- context:

cluster: minikube

namespace: default

user: minikube

name: minikube

current-context: minikube

kind: Config

preferences: {}

users:

- name: minikube

* **La valeur de de la variable CA :**

Mettez le contenu du fichier : /home/osboxes/.minikube/ca.crt

Par exemple :   
  
 -----BEGIN CERTIFICATE-----

MIIDBjCCAe6gAwIBAgIBATANBgkqhkiG9w0BAQsFADAVMRMwEQYDVQQDEwptaW5p

a3ViZUNBMB4XDTI0MDcyMTIxMTIzMFoXDTM0MDcyMDIxMTIzMFowFTETMBEGA1UE

AxMKbWluaWt1YmVDQTCCASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAKyL

Ye44Ep8lKTR33EJ/1+yxfQpCKxf8n5RJJGQ2Y2kLoycCA6M9lq8ON9MnOnab7jlj

k2Nw2UnYR/m/hDens30i3xL2qPtVp+l+iOCPbUQueTUv3JfqyCE84NV75VN0iMv+

BVdqiM/DrYSWBCF1pbU7X9u4YfNJycVpEmn9qeY/tmsdNuEU6nT7yEDz2ubkd5OG

7b+ks2E5FHuqgOxkfkrwTKiBbXG8ZJ94im6MBhU6MQO9qzre+FKOgqSFpzXcUeTM

sO7lo2DLEuStg1P1o2pc6F2+3e0NBK2CANXwXGCjoxng8Gfm9brKCOhCULmF1Hwr

XbtTOdDo8JVs3t8PZG8CAwEAAaNhMF8wDgYDVR0PAQH/BAQDAgKkMB0GA1UdJQQW

MBQGCCsGAQUFBwMCBggrBgEFBQcDATAPBgNVHRMBAf8EBTADAQH/MB0GA1UdDgQW

BBScBqZDW/JIvtmE4w2IaT93Tk5phzANBgkqhkiG9w0BAQsFAAOCAQEAfFei0x7q

Nyr0j5oYEIGNm+gQjDJ0vluHcTcsShfSfwwh5oFZ5GWjzcHxFt7pDm5JGx0KsS9C

+4TqBeV5KxUKQpPD//q/xF0WJNbKvk91fccSEEs/snxzfO87OuMDiRtif3sltkAQ

vcCFDGPdqeEGjsQMlwTM/Y54a7RxHz1eJkQqzb5mP84WDqJNqqr2VNpB4HI9fi89

V9ZAmc0/qMeneFRk0GaxqINZ3fUFkutwEfgOu2I5dmuIXiqT7+i5O+nupORa7IXc

dODX/eIbqHcCeEaKsvCM4aiugXeLf2qT4QHfHoF/8sMUByds43fiThbSKClyZoxn

F9ujgSOHjH62QA==

-----END CERTIFICATE-----

* **La valeur de de la variable CLIENTCRT :**

Mettez le contenu du fichier : /home/osboxes/.minikube/profiles/minikube/client.crt

* **La valeur de de la variable CLIENTKEY :**

Mettez le contenu du fichier : /home/osboxes/.minikube/profiles/minikube/client.key

1. Démarrer minikube : **minikube start --insecure-registry="192.168.112.3:5000”**
2. Changer la config du runner ajouter network\_mode = "host"

Contenu du fichier /etc/gitlab-runner/config.toml :

concurrent = 1

check\_interval = 0

connection\_max\_age = "15m0s"

shutdown\_timeout = 0

[session\_server]

session\_timeout = 1800

[[runners]]

name = "docker"

url = "http://192.168.112.3/"

id = 3

token = "qCdW6F-EjxYfuMnwVddn"

token\_obtained\_at = 2024-07-23T09:29:49Z

token\_expires\_at = 0001-01-01T00:00:00Z

executor = "docker"

clone\_url = "http://192.168.112.3/"

[runners.custom\_build\_dir]

[runners.cache]

MaxUploadedArchiveSize = 0

[runners.cache.s3]

[runners.cache.gcs]

[runners.cache.azure]

[runners.docker]

tls\_verify = false

image = "docker:19.03.12"

privileged = true

network\_mode = "host"

disable\_entrypoint\_overwrite = false

oom\_kill\_disable = false

disable\_cache = false

volumes = ["/cache", "/var/run/docker.sock:/var/run/docker.sock"]

shm\_size = 0

network\_mtu = 0

**Voila le contenu de .gitlab-ci.yml**

image: docker:24.0.5

services:

  - docker:24.0.5-dind

variables:

  DOCKER\_DRIVER: overlay2

  DOCKER\_TLS\_CERTDIR: ""

  DOCKER\_HOST: "tcp://192.168.112.3:2375"

  DOCKER\_REGISTRY: 192.168.112.3:5000

  DOCKER\_IMAGE: $DOCKER\_REGISTRY/myapp:$CI\_COMMIT\_SHA

  KUBE\_NAMESPACE: "default"

cache:

  key: ${CI\_COMMIT\_REF\_SLUG}

  paths:

    - node\_modules/

stages:

  - build

  - test

  - deploy

before\_script:

 - chmod +x node\_modules/.bin/mocha

build\_job:

 stage: build

 script:

   - docker build -t $DOCKER\_IMAGE .

   - docker push $DOCKER\_IMAGE

test\_job:

 stage: test

 script:

   - docker run --rm $DOCKER\_IMAGE npm test

deploy\_to\_Docker:

  stage: deploy

  script:

    - |

      # Nom du conteneur à arrêter et supprimer

      CONTAINER\_NAME=myapp

      # Forcer l'arrêt du conteneur s'il est en cours d'exécution

      if [ "$(docker ps -q -f name=$CONTAINER\_NAME)" ]; then

          echo "Stopping container $CONTAINER\_NAME..."

          docker stop $CONTAINER\_NAME

      fi

      # Supprimer le conteneur s'il existe

      if [ "$(docker ps -aq -f name=$CONTAINER\_NAME)" ]; then

          echo "Removing container $CONTAINER\_NAME..."

          docker rm $CONTAINER\_NAME

      fi

      # Exécuter un nouveau conteneur avec l'image spécifiée

      echo "Running new container with image $DOCKER\_IMAGE..."

      docker run -d --name $CONTAINER\_NAME -p 3000:3000 $DOCKER\_IMAGE

deploy\_to\_minikube:

  image: buildpack-deps:bullseye-curl

  stage: deploy

  script:

    - curl -LO "https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl"

    - chmod +x ./kubectl

    - mv ./kubectl /usr/local/bin/kubectl

    - mkdir -p /root/.kube

    - cat "$KUBECONFIG" > /root/.kube/config

    - cat "$CA"  > /root/.kube/ca.crt

    - cat "$CLIENTCRT"  > /root/.kube/client.crt

    - cat "$CLIENTKEY"  > /root/.kube/client.key

    - kubectl config set-cluster minikube --server=https://192.168.49.2:8443 --certificate-authority=/root/.kube/ca.crt

    - kubectl config set-credentials minikube --client-certificate=/root/.kube/client.crt --client-key=/root/.kube/client.key

    - kubectl config use-context minikube

    - kubectl get nodes

    - |

      if kubectl get deployment myapp >/dev/null 2>&1; then

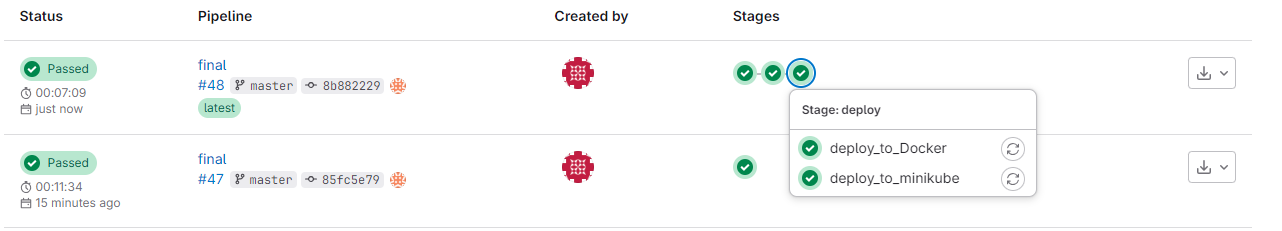
        echo "Deployment exists. Updating..."

        kubectl delete deployment myapp

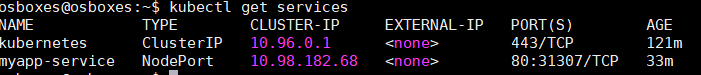
      fi

    - kubectl create deployment myapp --image=$DOCKER\_IMAGE --port=3000

    - kubectl apply -f kube\_manifests/service.yaml







Accedez a l application : <http://votre-ip:nodeport>