tp-dev-ops

NILS JAUDON

devops

Rendu

Fichier CD YAML :

```
name: Deploy to Kubernetes

on:

push:

branches:

- main

jobs:

deploy:

runs-on: ubuntu-latest

steps:

- name: Check out code

uses: actions/checkout@v3

- name: Set up Kubernetes

uses: azure/setup-kubernetes@v1

with:

kubeconfig: ${{ secrets.KUBECONFIG }}

- name: Deploy with Helm
```

```
  run: helm upgrade --install mon-app ./mon-app
```

fichier CI.yaml :

```
  name: CI - Lint & Security & Docker


  on: [push, pull_request]


  jobs:

  flake8-lint:

  runs-on: ubuntu-latest

  name: Lint

  steps:

  - name: Checkout

  uses: actions/checkout@v3


  - name: Setup Python

  uses: actions/setup-python@v4

  with:

  python-version: "3.11"


  - name: flake8 Lint

  uses: py-actions/flake8@v2
```

```
bandit-scan:

runs-on: ubuntu-latest

name: Bandit

steps:

- name: Checkout

uses: actions/checkout@v4



- name: Setup Python

uses: actions/setup-python@v5

with:

python-version: "3.12"



- name: Install Bandit

run: pip install bandit



- name: Run Bandit

run: bandit --severity-level high -r src/app.py



docker:

runs-on: ubuntu-latest

needs: [flake8-lint, bandit-scan]
```

```
name: Docker Build & Push

steps:

- name: Checkout

uses: actions/checkout@v4



- name: Login to Docker Hub

uses: docker/login-action@v3

with:

username: ${{ secrets.DOCKERHUB_USERNAME }}

password: ${{ secrets.DOCKERHUB_TOKEN }}



- name: Set up QEMU

uses: docker/setup-qemu-action@v3



- name: Set up Docker Buildx

uses: docker/setup-buildx-action@v3



- name: Build and push Docker image

uses: docker/build-push-action@v6

with:

context: .
```

```
  push: true
  tags: ${{ secrets.DOCKERHUB_USERNAME }}/mon-app:latest

  - name: Docker Scout analysis
  uses: docker/scout-action@v1
  with:
  command: cves
  image: ${{ secrets.DOCKERHUB_USERNAME }}/mon-app:latest
  only-severities: critical,high
```

le fichier docker build :

```
  FROM python:3.9

  WORKDIR /src

  COPY requirements.txt .

  RUN pip install --no-cache-dir -r requirements.txt

  COPY . .
```

```
CMD ["python", "/src/app.py"]


EXPOSE 8080
```

Le helm Chart

```
# Default values for mon-app.

# This is a YAML-formatted file.

# Declare variables to be passed into your templates.


# This will set the replicaset count more information can be found here:
https://kubernetes.io/docs/concepts/workloads/controllers/replicaset/

replicaCount: 1


# This sets the container image more information can be found here:
https://kubernetes.io/docs/concepts/containers/images/

image:

repository: z3ph7r/mon-app


# This sets the pull policy for images.

pullPolicy: IfNotPresent

# Overrides the image tag whose default is the chart appVersion.

tag: "latest"
```

```
# This is for the secrets for pulling an image from a private repository
more information can be found here:
https://kubernetes.io/docs/tasks/configure-pod-container/pull-image-private-
registry/

imagePullSecrets: []

# This is to override the chart name.

nameOverride: ""

fullnameOverride: ""



# This section builds out the service account more information can be found
here: https://kubernetes.io/docs/concepts/security/service-accounts/

serviceAccount:

# Specifies whether a service account should be created

create: true

# Automatically mount a ServiceAccount's API credentials?

automount: true

# Annotations to add to the service account

annotations: {}

# The name of the service account to use.

# If not set and create is true, a name is generated using the fullname
template

name: ""
```

```
# This is for setting Kubernetes Annotations to a Pod.

# For more information checkout:
https://kubernetes.io/docs/concepts/overview/working-with-
objects/annotations/

podAnnotations: {}

# This is for setting Kubernetes Labels to a Pod.

# For more information checkout:
https://kubernetes.io/docs/concepts/overview/working-with-objects/labels/

podLabels: {}



podSecurityContext: {}

# fsGroup: 2000



securityContext: {}

# capabilities:

# drop:

# - ALL

# readOnlyRootFilesystem: true

# runAsNonRoot: true

# runAsUser: 1000



# This is for setting up a service more information can be found here:
https://kubernetes.io/docs/concepts/services-networking/service/
```

```
service:

# This sets the service type more information can be found here:
https://kubernetes.io/docs/concepts/services-networking/service/#publishing-
services-service-types

type: ClusterIP

# This sets the ports more information can be found here:
https://kubernetes.io/docs/concepts/services-networking/service/#field-spec-
ports

port: 80



# This block is for setting up the ingress for more information can be found
here: https://kubernetes.io/docs/concepts/services-networking/ingress/

ingress:

enabled: true

className: "nginx"

hosts:

- host: mon-app.local

paths:

- path: /

pathType: Prefix

tls: []

# - secretName: chart-example-tls

# hosts:

# - chart-example.local
```

```
resources: {}

# We usually recommend not to specify default resources and to leave this as
a conscious

# choice for the user. This also increases chances charts run on
environments with little

# resources, such as Minikube. If you do want to specify resources,
uncomment the following

# lines, adjust them as necessary, and remove the curly braces after
'resources:'.

# limits:

# cpu: 100m

# memory: 128Mi

# requests:

# cpu: 100m

# memory: 128Mi



# This is to setup the liveness and readiness probes more information can be
found here: https://kubernetes.io/docs/tasks/configure-pod-
container/configure-liveness-readiness-startup-probes/

livenessProbe:

httpGet:

path: /

port: http

readinessProbe:
```

```
httpGet:

path: /

port: http



# This section is for setting up autoscaling more information can be found
here: https://kubernetes.io/docs/concepts/workloads/autoscaling/

autoscaling:

enabled: false

minReplicas: 1

maxReplicas: 100

targetCPUUtilizationPercentage: 80

# targetMemoryUtilizationPercentage: 80



# Additional volumes on the output Deployment definition.

volumes: []

# - name: foo

# secret:

# secretName: mysecret

# optional: false



# Additional volumeMounts on the output Deployment definition.

volumeMounts: []
```

```
# - name: foo

# mountPath: "/etc/foo"

# readOnly: true



nodeSelector: {}



tolerations: []



affinity: {}

resources:

requests:

cpu: "100m" # 100 milli-Cores de CPU

memory: "128Mi" # 128 MiB de mémoire

limits:

cpu: "500m" # 500 milli-Cores de CPU

memory: "512Mi" # 512 MiB de mémoire
```

- Les lignes de commandes que vous avez tapez pour installer l'ingress controller et kubecost dans un fichier `helm.md`

Ajouter le dépôt Helm

Déploiement sur Kubernetes avec Helm

Installation du Contrôleur Ingress NGINX

```
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
```

```
helm install nginx-ingress ingress-nginx/ingress-nginx --version 4.10.0 --create-namespace --namespace ingress-nginx
```

```
kubectl get svc -n ingress-nginx
```

Installation de Kubecost

```
helm repo add kubecost https://kubecost.github.io/cost-analyzer/
```

```
helm repo update
```

```
kubectl create namespace kubecost
```

```
helm install kubecost kubecost/cost-analyzer --namespace kubecost
```

```
kubectl get pods -n kubecost
```

Déploiement de notre application

```
helm upgrade --install mon-app ./mon-app
```

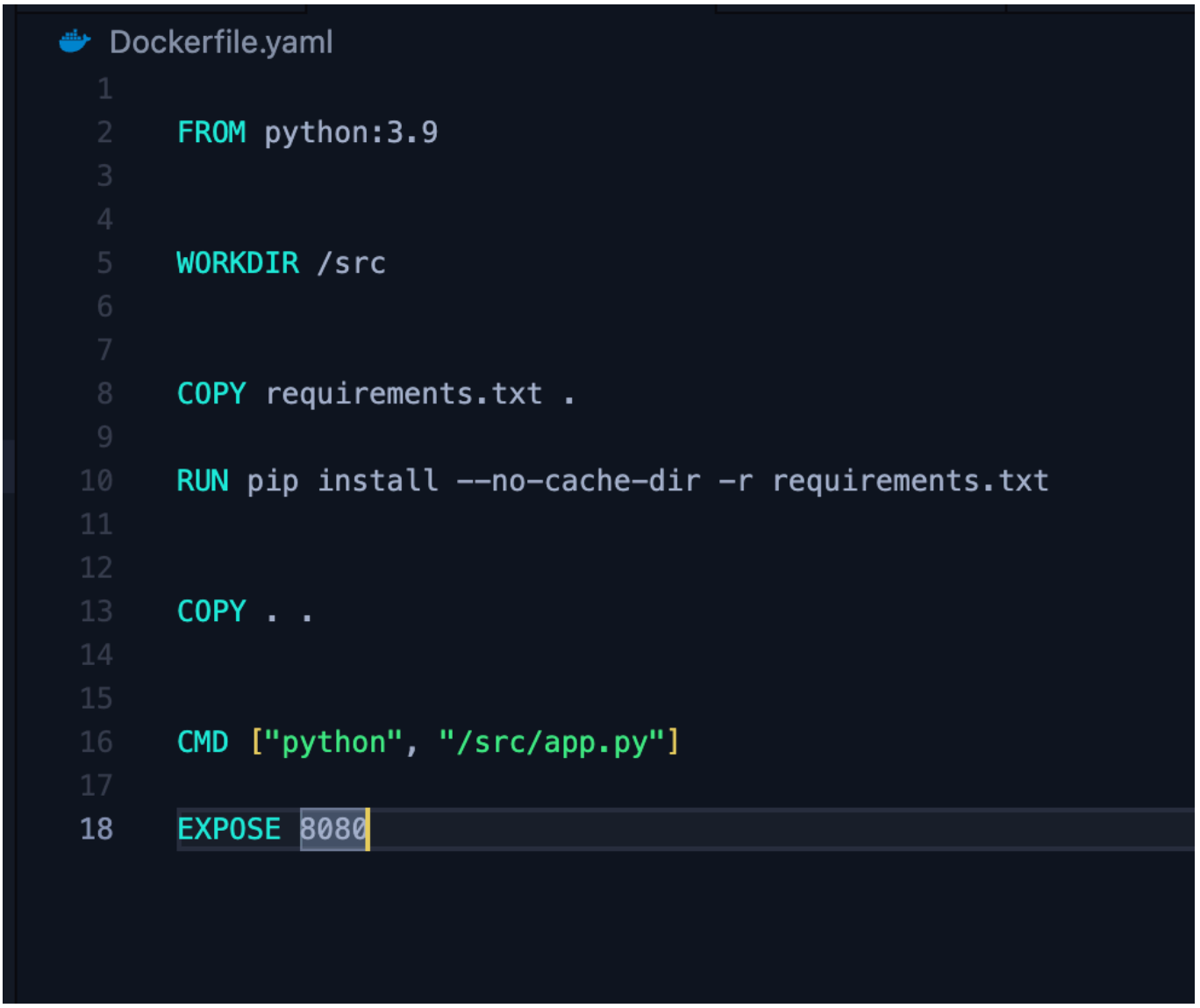
```
kubectl get ingress
```

Détail du tp :

Création d'un repo :
déplacer les fichiers du tp vers le bon repo
création d'un fichier .gitignore

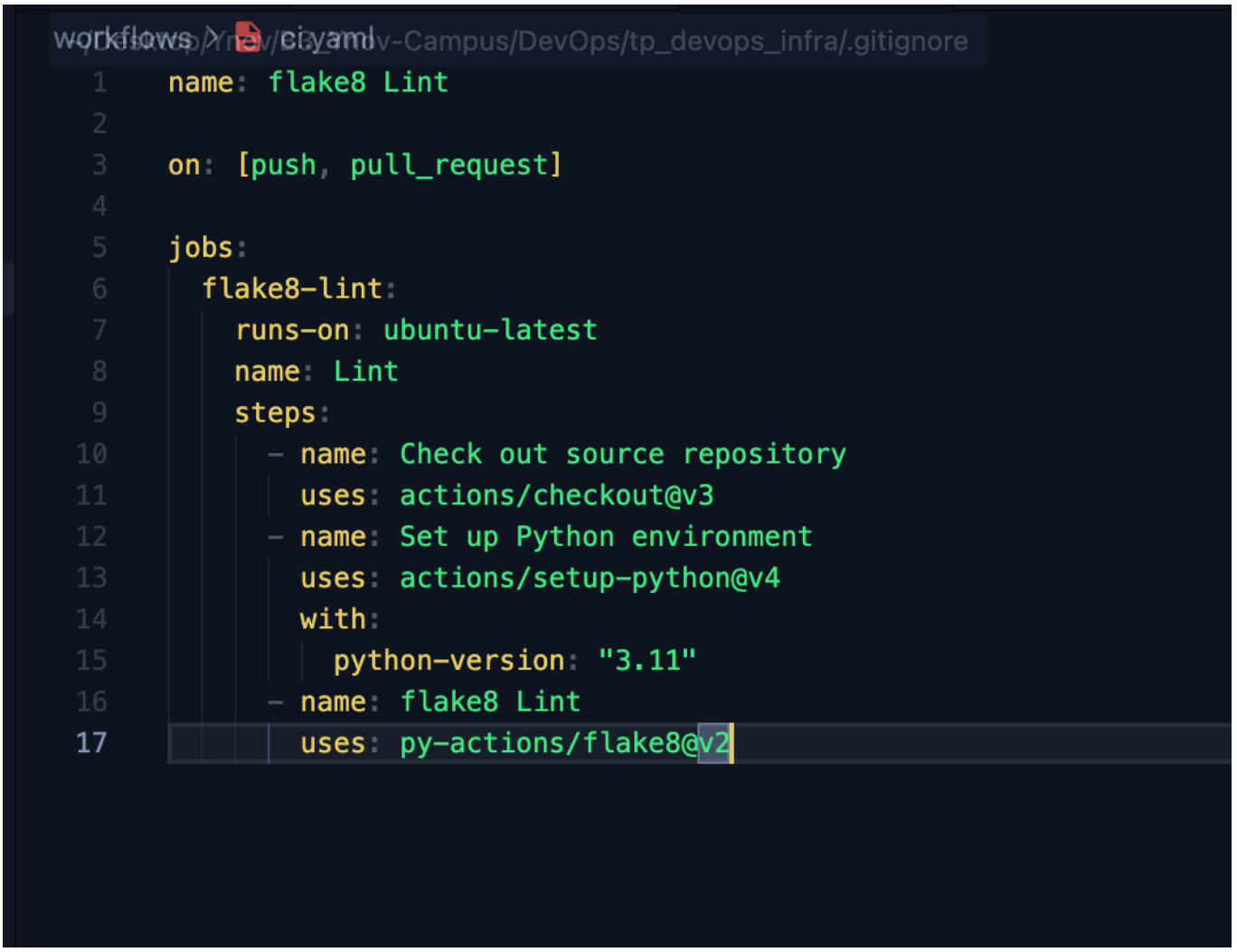
Création d'un fichier .dockerignore

création d'un fichier dockerfiles



création d'un dossier workflow avec ci.yaml

ajout d'un workflow pour flask
https://github.com/py-actions/flake8



ajoute d'un workflow pour bandit :

https://github.com/CICDToolbox/bandit!

```yaml
  bandit-scan:
    runs-on: ubuntu-latest
    name: Bandit
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Setup Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.12"

      - name: Install Bandit
        run: pip install bandit

      - name: Run Bandit
        run: bandit --severity-level high -r src/app.py
```

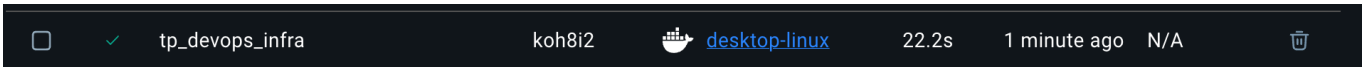
Ajout d'un : docker-build:
https://github.com/marketplace/actions/build-and-push-docker-images

```yaml
  docker:
    runs-on: ubuntu-latest
    needs: [flake8-lint, bandit-scan]
    name: Docker Build & Push
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - name: Login to Docker Hub
        uses: docker/login-action@v3
        with:
          username: ${{ secrets.DOCKERHUB_USERNAME }}
          password: ${{ secrets.DOCKERHUB_TOKEN }}

      - name: Set up QEMU
        uses: docker/setup-qemu-action@v3

      - name: Set up Docker Buildx
        uses: docker/setup-buildx-action@v3

      - name: Build and push Docker image
        uses: docker/build-push-action@v6
        with:
          context: .
          push: true
          tags: ${{ secrets.DOCKERHUB_USERNAME }}/mon-app:latest
```

Ajout de docker-scout :
https://github.com/docker/scout-action

```yaml
- name: Docker Scout analysis
  uses: docker/scout-action@v1
  with:
    command: cves
    image: ${{ secrets.DOCKERHUB_USERNAME }}/mon-app:latest
    only-severities: critical,high
```

test d'un build en local :



kubernetes

dans terminal de vscode docker Login

pour se connecter au docker

Terraform :

On recupère la config de l'ancien tp et on copie colle dans notre repertoire git. puis on éffectue un : Terraform init

```
venv→  tp_devops_infra git:(main) ✗ terraform init

Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/azurerm from the dependency lock file
- Reusing previous version of hashicorp/helm from the dependency lock file
- Using previously-installed hashicorp/azurerm v4.22.0
- Using previously-installed hashicorp/helm v2.17.0

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
venv→  tp_devops_infra git:(main) ✗
```

- on fait ensuite un terraform apply

```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes
```

```
venv➜  tp_devops_infra git:(main) ✗ terraform apply

  var.name
    Generic name, enter your name to identify your resources

    Enter a value: yes
```

```
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Still creating... [10s elapsed]
azurerm_resource_group.rg: Creation complete after 12s [id=/subscriptions/932251e4-1f35-41ba-910c-9dc0d23b6005/resourceGroups/plop]
azurerm_kubernetes_cluster.aks: Creating...
azurerm_kubernetes_cluster.aks: Still creating... [10s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [20s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [30s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [40s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [50s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m0s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m10s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m20s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m30s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m40s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [1m50s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m0s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m10s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m20s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m30s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m40s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [2m50s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [3m0s elapsed]
azurerm_kubernetes_cluster.aks: Still creating... [3m10s elapsed]
```

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

client_certificate = <sensitive>
kube_config = <sensitive>
venv➜  tp_devops_infra git:(main) ✗
```

on se connecte au cluster :

```
venv➜  tp_devops_infra git:(main) ✗ az aks get-credentials --resource-group plop --name yes-default

Merged "yes-default" as current context in /Users/nils/.kube/config
venv➜  tp_devops_infra git:(main) ✗
```

on vérifie la connexion au cluster :

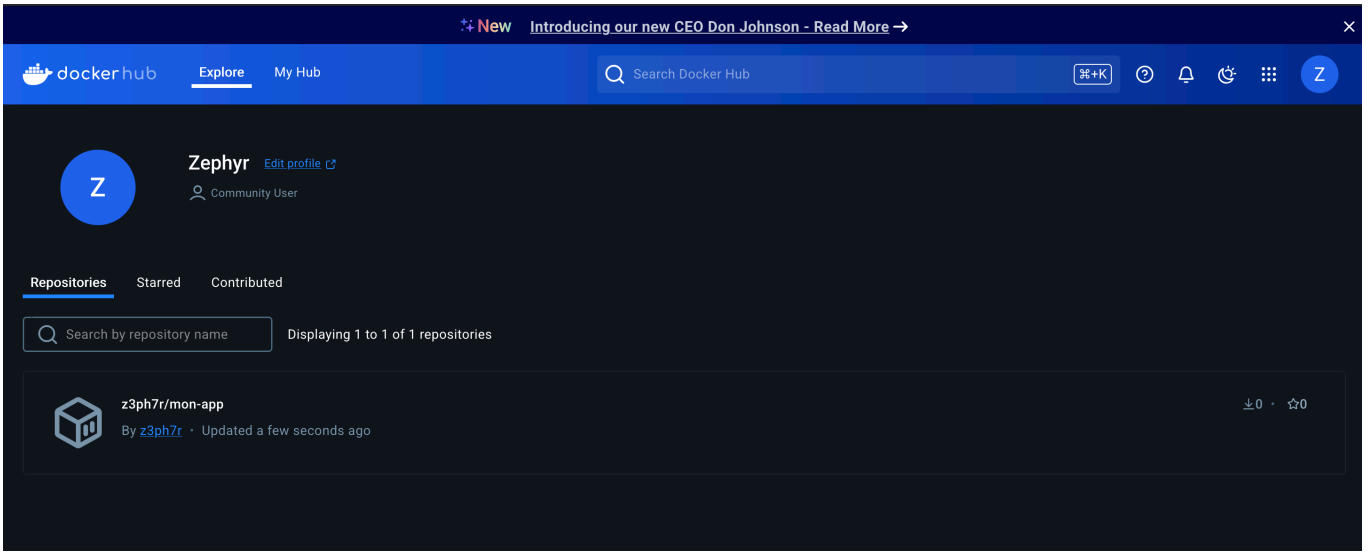
```
venv➜  tp_devops_infra git:(main) ✗ kubectl get nodes

NAME                                STATUS   ROLES    AGE     VERSION
aks-default-85878265-vmss000000   Ready    <none>   2m34s   v1.30.10
```

push docker

on tag l'image : docker tag mon-app:latest z3ph7r/mon-app:latest

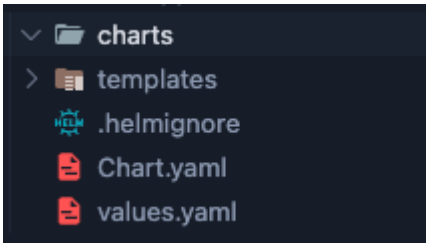
on push :
docker push z3ph7r/mon-app:latest

```
venv→  tp_devops_infra git:(main) docker push z3ph7r/mon-app:latest

The push refers to repository [docker.io/z3ph7r/mon-app]
6b2aa38ef277: Pushed
26207ac70544: Pushed
ce58a9f2d87a: Pushed
ba6e4ab283c5: Pushed
e25bb20257c7: Mounted from library/python
e25809daf74c: Mounted from library/python
cfd4bc2cf5f9: Mounted from library/python
77122155c2dc: Mounted from library/python
b8974bf44706: Mounted from library/python
f8e34ba5db39: Mounted from library/python
948048d45864: Mounted from library/python
latest: digest: sha256:35a4819ba257f9894a5dad8a6b37bfb2f3230aa4e531bce8a6aa70c3652142bf size: 2629
venv→  tp_devops_infra git:(main)
```

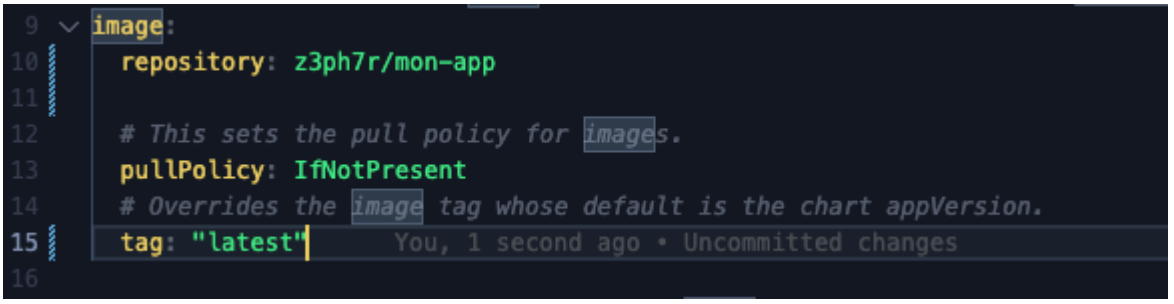


helm

helm create mon-app



on modifie ensuite l'image de configuration :



on setup up ingress :

```yaml
# This block is for setting up the ingress for more information can be found here: https://kubernetes.io/docs/concepts/services-networking
ingress:
  enabled: true
  className: "nginx"
  annotations: {}
    # kubernetes.io/ingress.class: nginx
    # kubernetes.io/tls-acme: "true"
  hosts:
    - host: mon-app.local
      paths:
        - path: /
          pathType: Prefix
  tls: []
  #  - secretName: chart-example-tls
  #    hosts:
  #      - chart-example.local
```

on ajoute le repo :
helm repo add ingress-nginx https://kubernetes.github.io/ingress-nginx
helm repo update

& on installe le controlleur :
helm install nginx-ingress ingress-nginx/ingress-nginx --create-namespace --namespace ingress-nginx

on recharge la commande :

on regarde l'ip : : kubectl get svc -n ingress-nginx

```
##
# Host Database
#
# localhost is used to configure the loopback interface
# when the system is booting.  Do not change this entry.
#
20.19.153.6   mon-app.local

127.0.0.1       localhost test.localhost
255.255.255.255 broadcasthost
::1             localhost
```

ajout au fichier host

```
venv→  tp_devops_infra git:(main) ✗ kubectl get svc -n ingress-nginx
NAME                                       TYPE           CLUSTER-IP     EXTERNAL-IP   PORT(S)                      AGE
nginx-ingress-ingress-nginx-controller             LoadBalancer   10.0.216.40   20.19.153.6   80:32045/TCP,443:32291/TCP   99s
nginx-ingress-ingress-nginx-controller-admission   ClusterIP      10.0.87.16    <none>        443/TCP                      99s
venv→  tp_devops_infra git:(main) ✗
```

helm

cd.yaml

on ajoute le repo : helm repo add kubecost https://kubecost.github.io/cost-analyzer/

on configure le fichier cd yaml pour déployer helm

```yaml
name: Deploy to Kubernetes
on:
  push:
    branches:
      - main
jobs:
  deploy:
    runs-on: ubuntu-latest
    steps:
      - name: Check out code
        uses: actions/checkout@v3
      - name: Set up Kubernetes
        uses: azure/setup-kubernetes@v1
        with:
          kubeconfig: ${{ secrets.KUBECONFIG }}
      - name: Deploy with Helm
        run: helm upgrade --install mon-app ./mon-app
```

TEST

```
venv→  tp_devops_infra git:(main) ✗ kubectl get pods -n kubecost

NAME                                          READY   STATUS    RESTARTS   AGE
kubecost-cost-analyzer-89fbd9ccd-qt22s        4/4     Running   0          3m16s
kubecost-forecasting-754f7f886d-7nk5t         1/1     Running   0          3m16s
kubecost-grafana-6786f47d89-27k7m             2/2     Running   0          3m16s
kubecost-prometheus-server-ff65dff66-d9srg    1/1     Running   0          3m16s
venv→  tp_devops_infra git:(main) ✗
```

Test INgress

```
venv→  tp_devops_infra git:(main) ✗ kubectl get ingress

NAME      CLASS   HOSTS           ADDRESS       PORTS   AGE
mon-app   nginx   mon-app.local   20.19.153.6   80      52m
venv→  tp_devops_infra git:(main) ✗
```