tp-dev-ops

TP DevOps Infrastructure

NILS JAUDON

devops

Rendu

Fichier CD YAML

```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

```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
```

Dockerfile

```
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
```

Helm Chart (values.yaml)

```
# Default values for mon-app
# This is a YAML-formatted file
# Declare variables to be passed into your templates

# Replicaset count
replicaCount: 1

# Container image configuration
image:
  repository: z3ph7r/mon-app
  pullPolicy: IfNotPresent
  tag: "latest"

# Private repository configuration
imagePullSecrets: []
nameOverride: ""
fullnameOverride: ""
```

```yaml
# Service account configuration
serviceAccount:
  create: true
  automount: true
  annotations: {}
  name: ""

# Pod annotations and labels
podAnnotations: {}
podLabels: {}

podSecurityContext: {}
securityContext: {}

# Service configuration
service:
  type: ClusterIP
  port: 80

# Ingress configuration
ingress:
  enabled: true
  className: "nginx"
  hosts:
    - host: mon-app.local
      paths:
        - path: /
          pathType: Prefix
  tls: []

# Resource limits and requests
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

# Health checks
livenessProbe:
  httpGet:
    path: /
    port: http
readinessProbe:
  httpGet:
    path: /
    port: http

# Autoscaling configuration
```

```
autoscaling:
  enabled: false
  minReplicas: 1
  maxReplicas: 100
  targetCPUUtilizationPercentage: 80

# Volumes and mounts
volumes: []
volumeMounts: []

nodeSelector: {}
tolerations: []
affinity: {}
```

Commandes d'installation (helm.md)

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

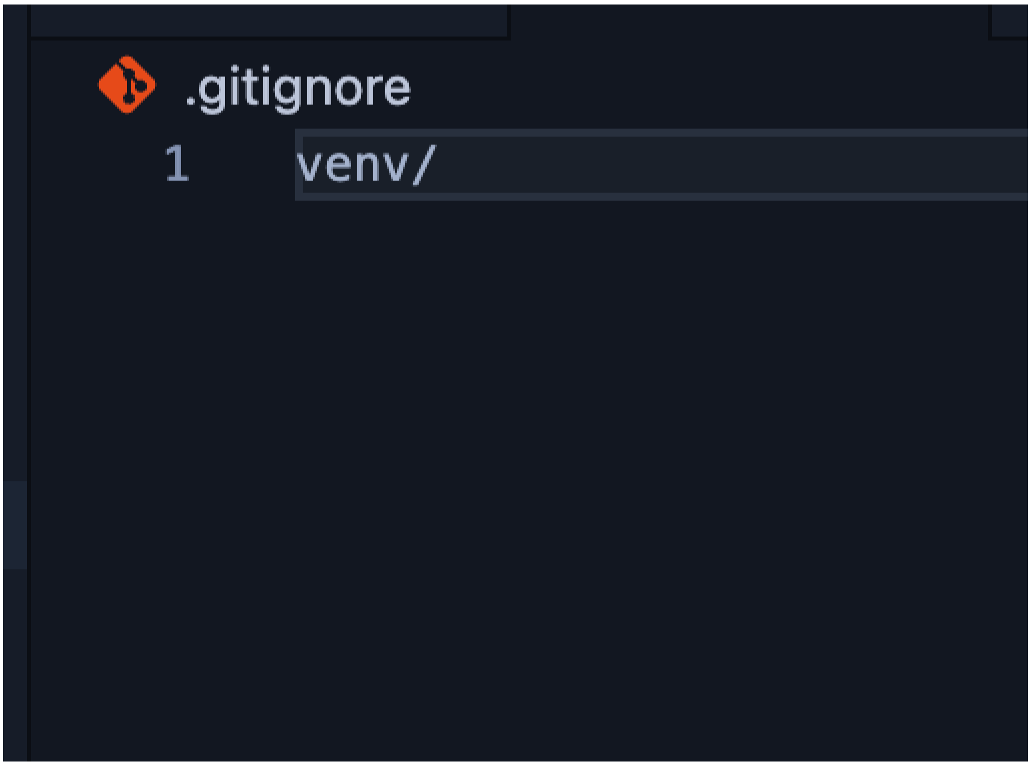
```
helm upgrade --install mon-app ./mon-app
```

```
kubectl get ingress
```

Détail du TP

Création du repository

- Déplacement des fichiers du TP vers le bon repo
- Création d'un fichier .gitignore

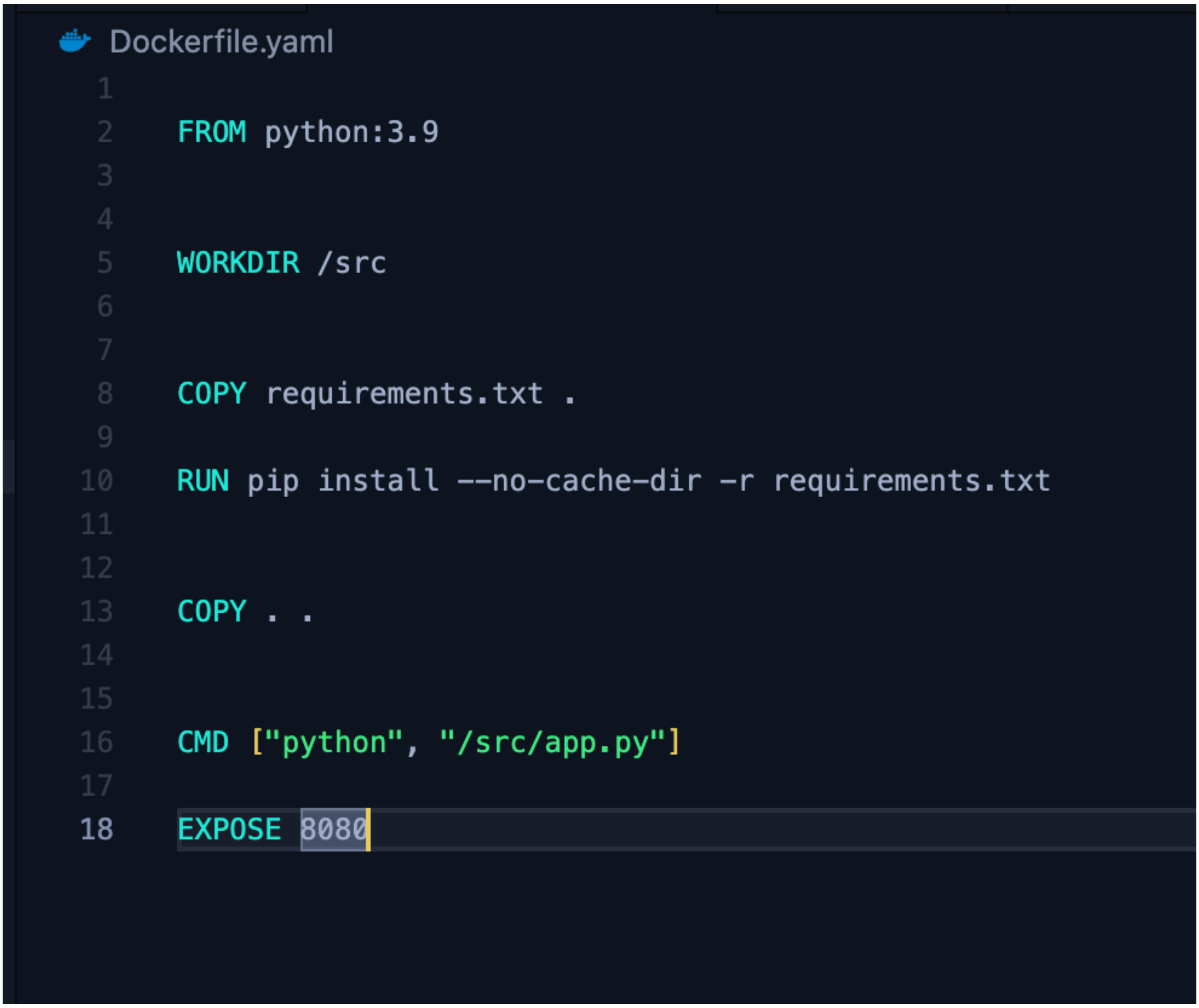


.gitignore

```
venv/
```

Configuration Docker

- Création d'un fichier .dockerignore
- Création d'un fichier Dockerfile

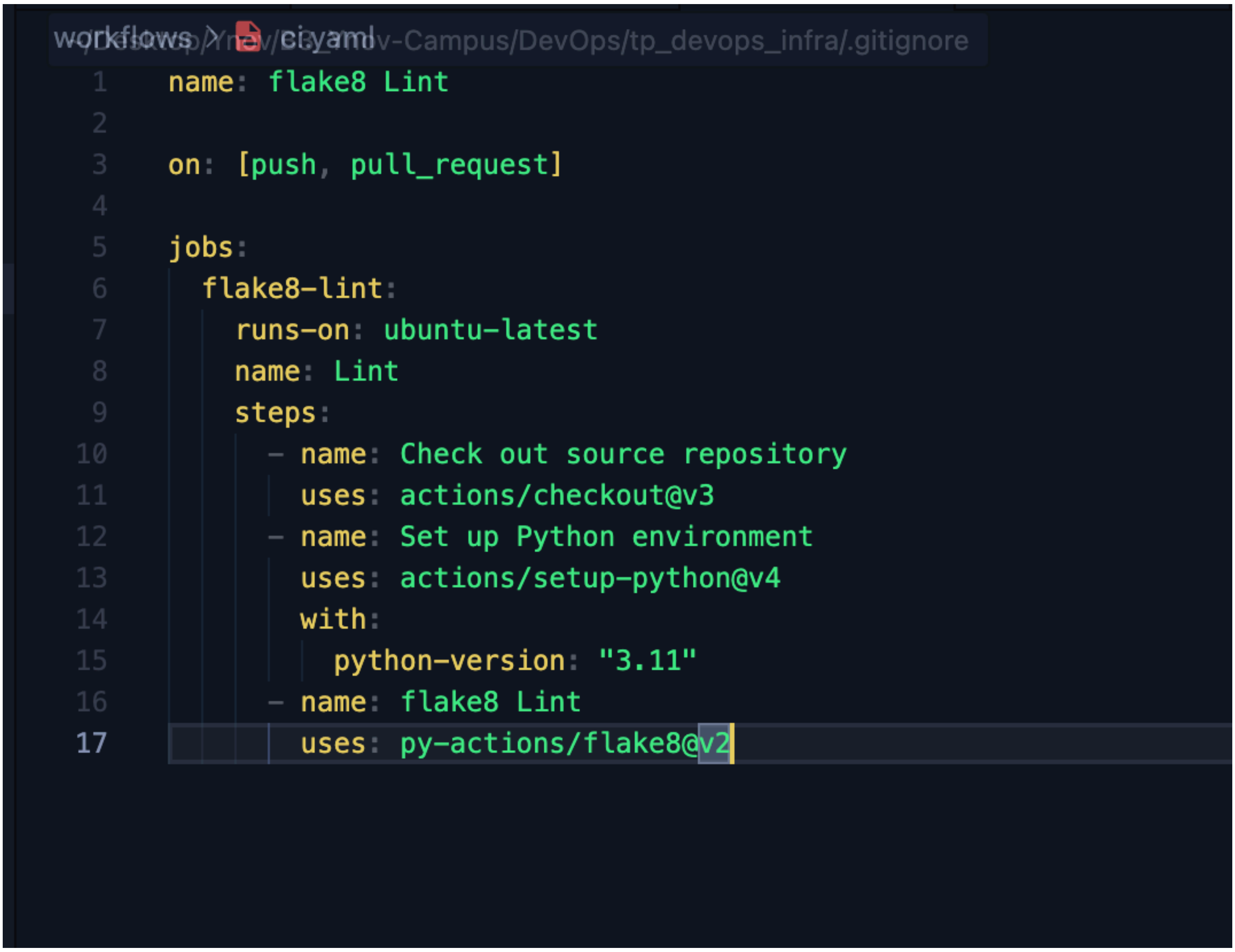


Dockerfile.yaml

```

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
```

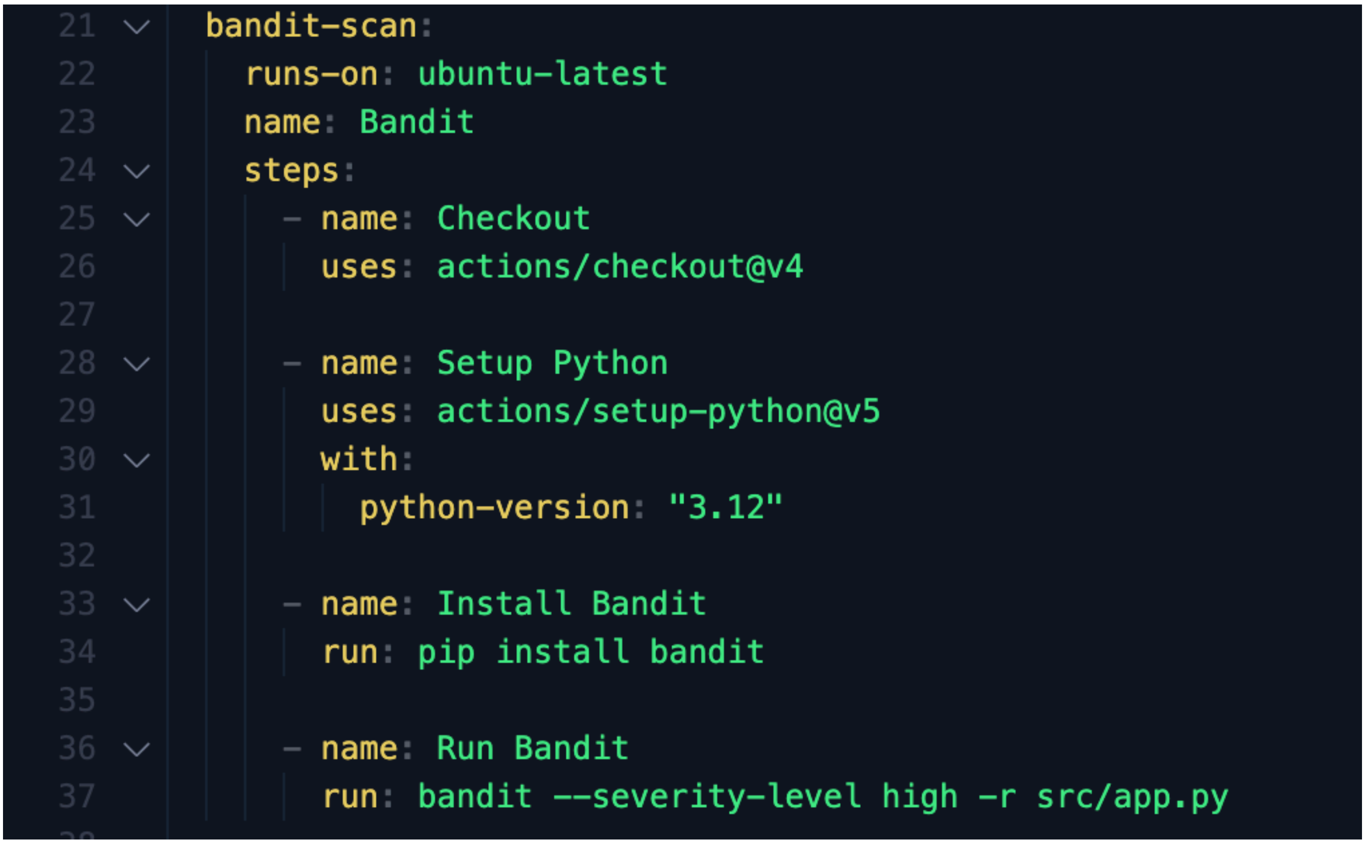
Mise en place du workflow CI/CD

- Création d'un dossier workflow avec ci.yaml
- Ajout d'un workflow pour flask: https://github.com/py-actions/flake8



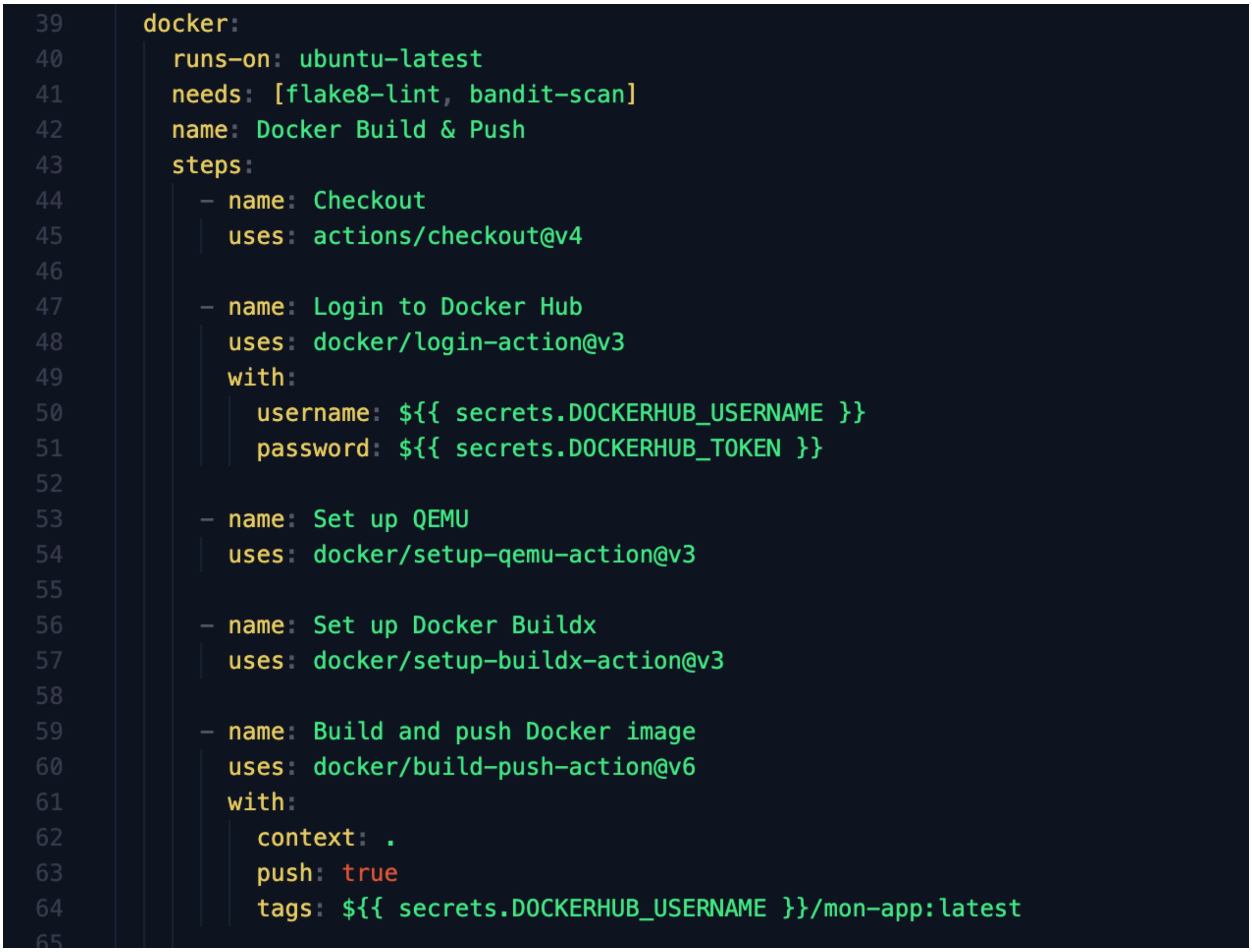
```yaml
name: flake8 Lint

on: [push, pull_request]

jobs:
  flake8-lint:
    runs-on: ubuntu-latest
    name: Lint
    steps:
      - name: Check out source repository
        uses: actions/checkout@v3
      - name: Set up Python environment
        uses: actions/setup-python@v4
        with:
          python-version: "3.11"
      - name: flake8 Lint
        uses: py-actions/flake8@v2
```

- Ajout d'un workflow pour bandit: https://github.com/CICDToolbox/bandit



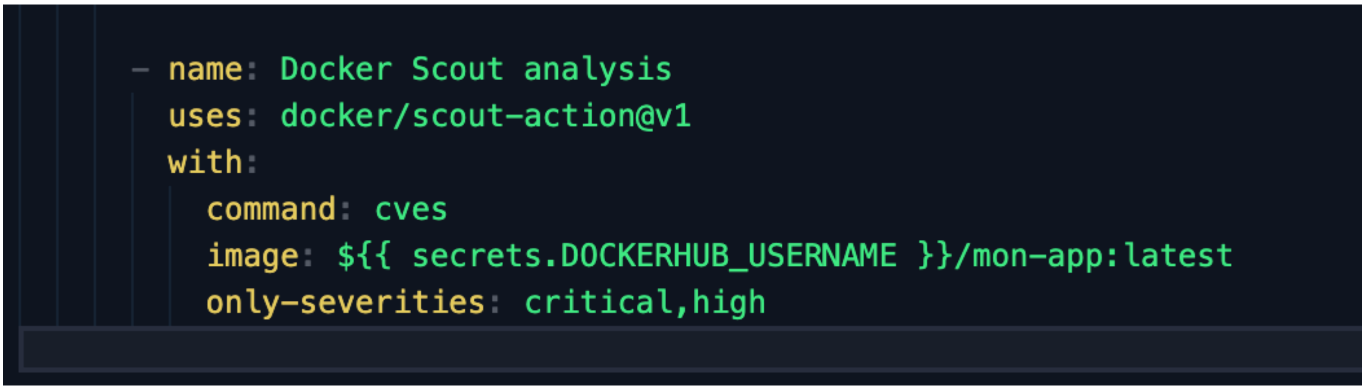
```yaml
  bandit-scan:
    runs-on: ubuntu-latest
    name: Bandit
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - 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 de docker-build: https://github.com/marketplace/actions/build-and-push-docker-images



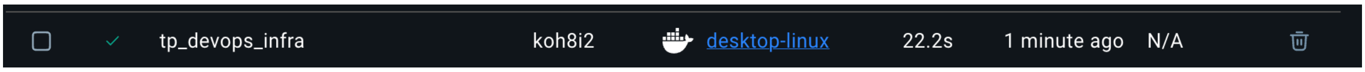
```
  docker:
    runs-on: ubuntu-latest
    needs: [flake8-lint, bandit-scan]
    name: Docker Build & Push
    steps:
      - name: Checkout
        uses: actions/checkout@v4

      - 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



```
      - 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:



```
□   ✓   tp_devops_infra          koh8i2     desktop-linux     22.2s     1 minute ago     N/A
```

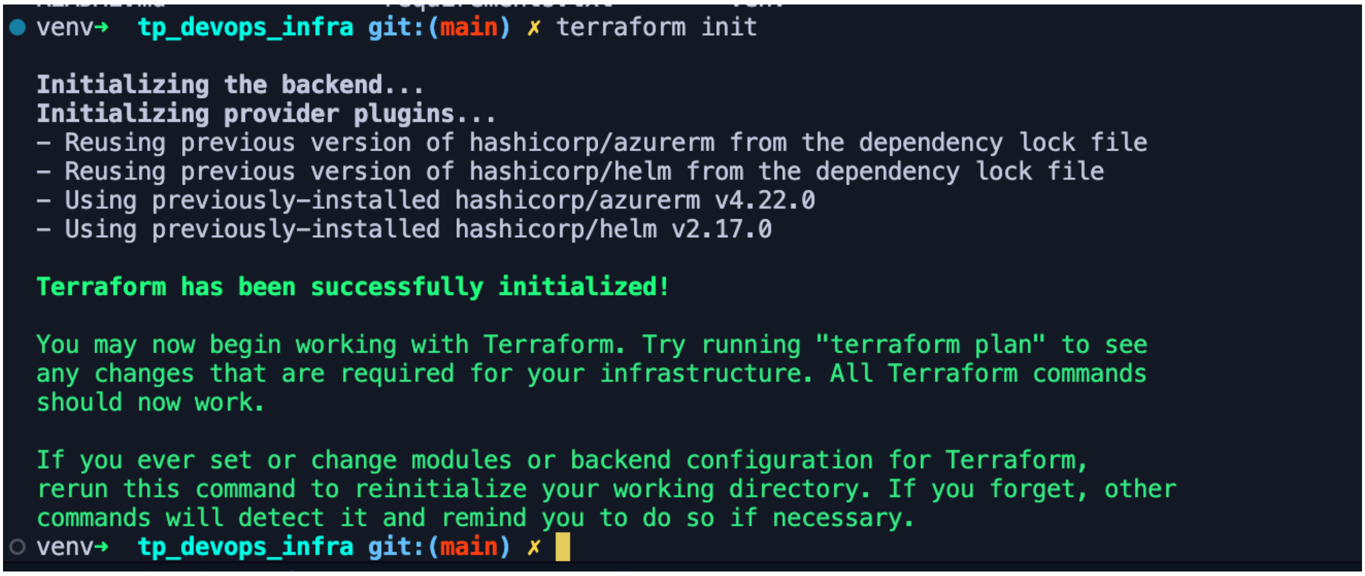
Déploiement Kubernetes

Préparation de l'environnement

- Connexion au Docker Hub depuis le terminal VSCode

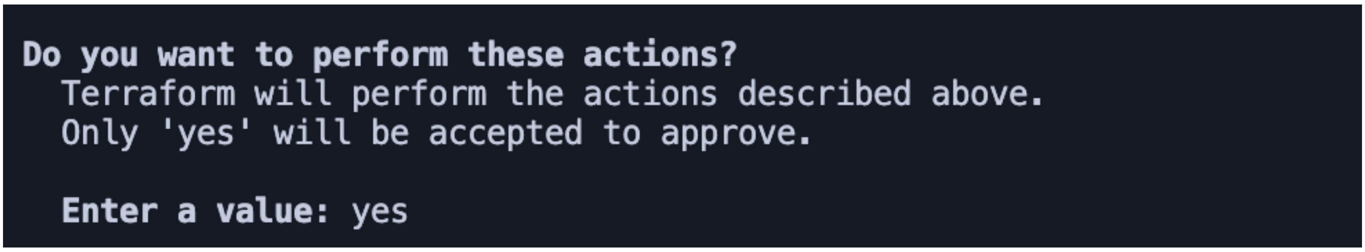
Terraform

- Récupération de la config de l'ancien TP
- Initialisation avec Terraform:

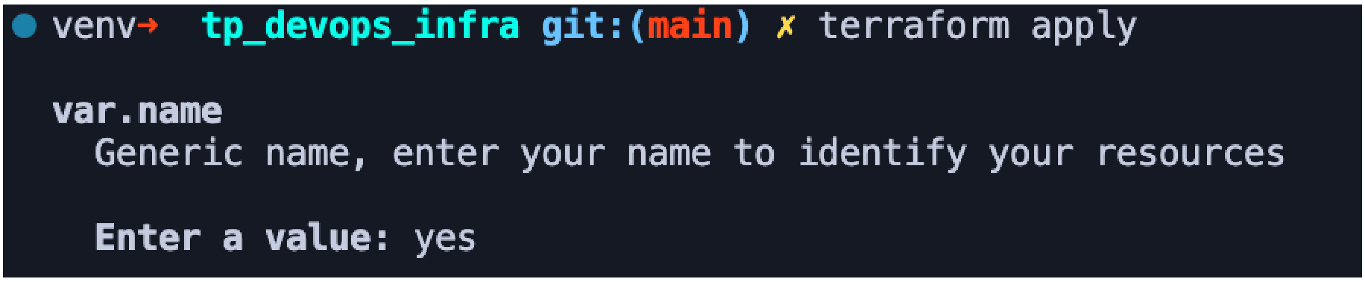


```
● venv→  tp_devops_infra git:(main) ✗ terraform init

  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) ✗ █
```

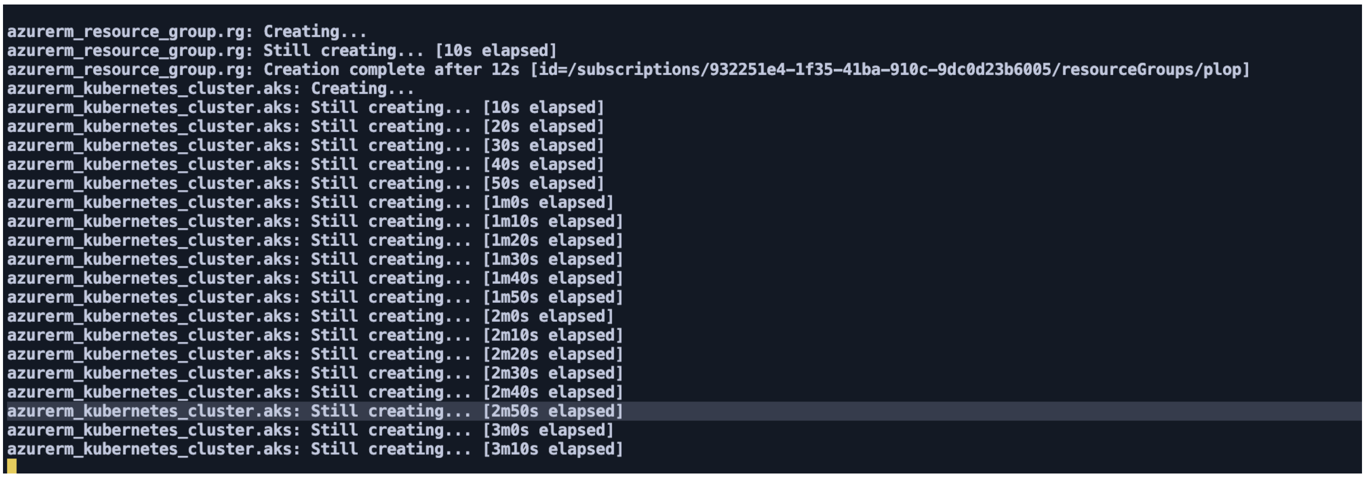
- Application de la configuration:



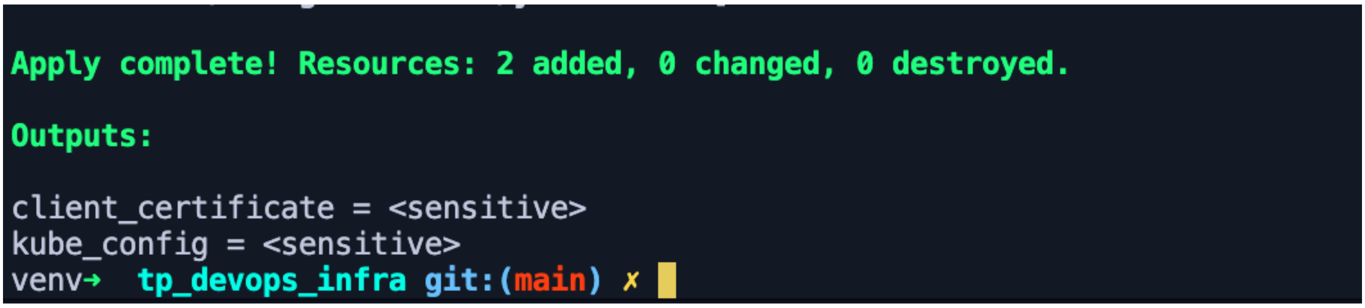
```
Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  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
```

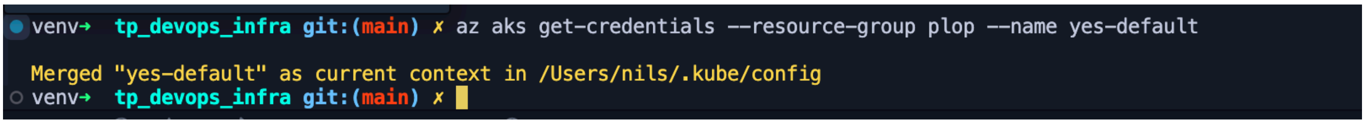


```
azurerm_resource_group.rg: Creating...
azurerm_resource_group.rg: Still creating... [10s elapsed]
azurerm_resource_group.rg: Creation complete after 12s [id=/subscriptions/932251e4-1f35-41ba-910c-9dc0d23b6005/resourceGroups/plop]
azurerm_kubernetes_cluster.aks: Creating...
azurerm_kubernetes_cluster.aks: 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]
```



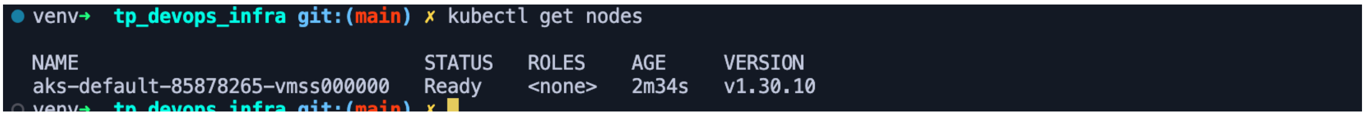
```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.

Outputs:

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

- Connexion 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) ✗
```

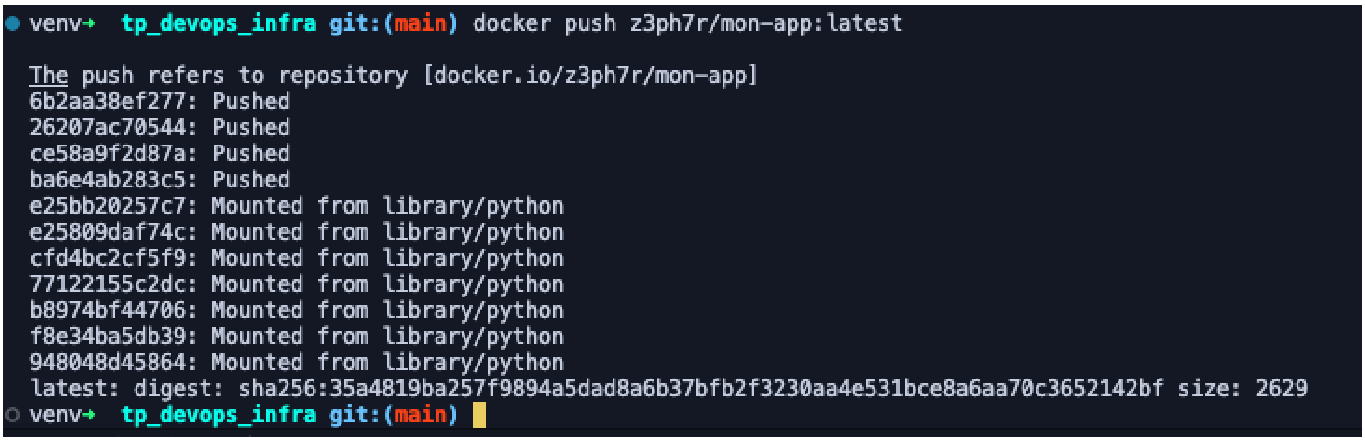
- Vérification de la connexion:



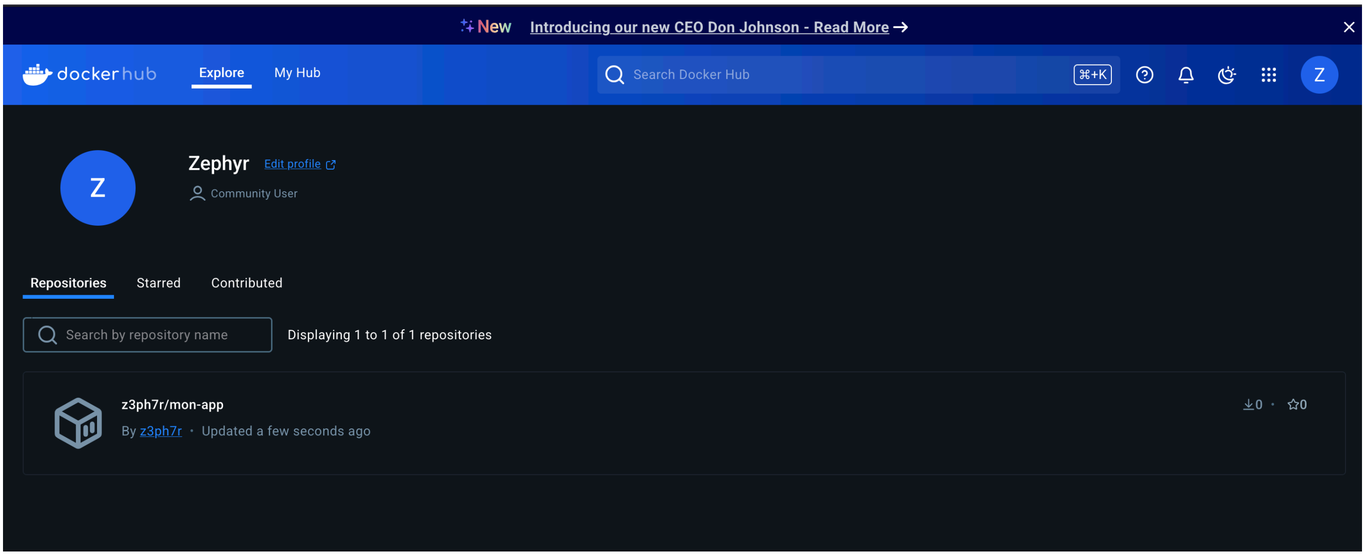
```
● venv→  tp_devops_infra git:(main) ✗ kubectl get nodes

  NAME                              STATUS   ROLES    AGE     VERSION
  aks-default-85878265-vmss000000   Ready    <none>   2m34s   v1.30.10
○ venv→  tp_devops_infra git:(main) ✗
```

Publication de l'image Docker

- Tag de l'image: `docker tag mon-app:latest z3ph7r/mon-app:latest`
- Publication: `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) 
```



Zephyr Edit profile

Community User

Repositories · Starred · Contributed

Search by repository name

Displaying 1 to 1 of 1 repositories

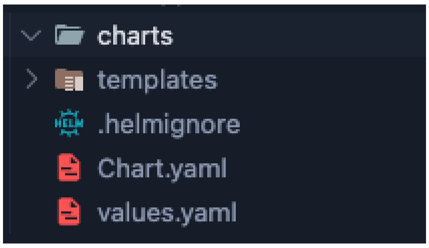
z3ph7r/mon-app

By z3ph7r · Updated a few seconds ago

0 · 0

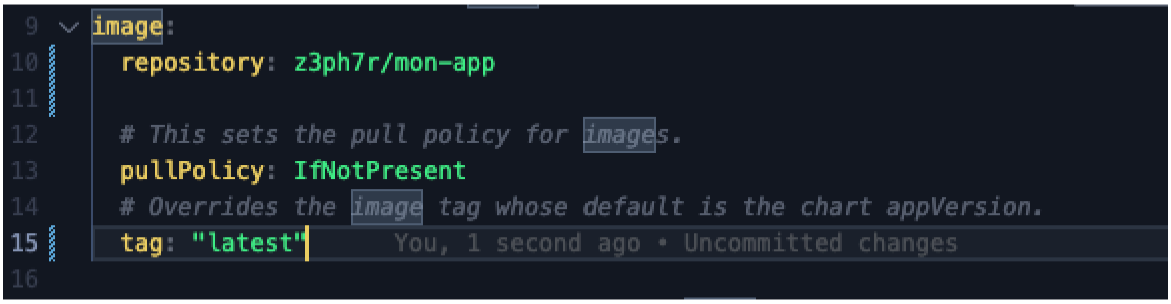
Configuration Helm

- Création du chart: `helm create mon-app`



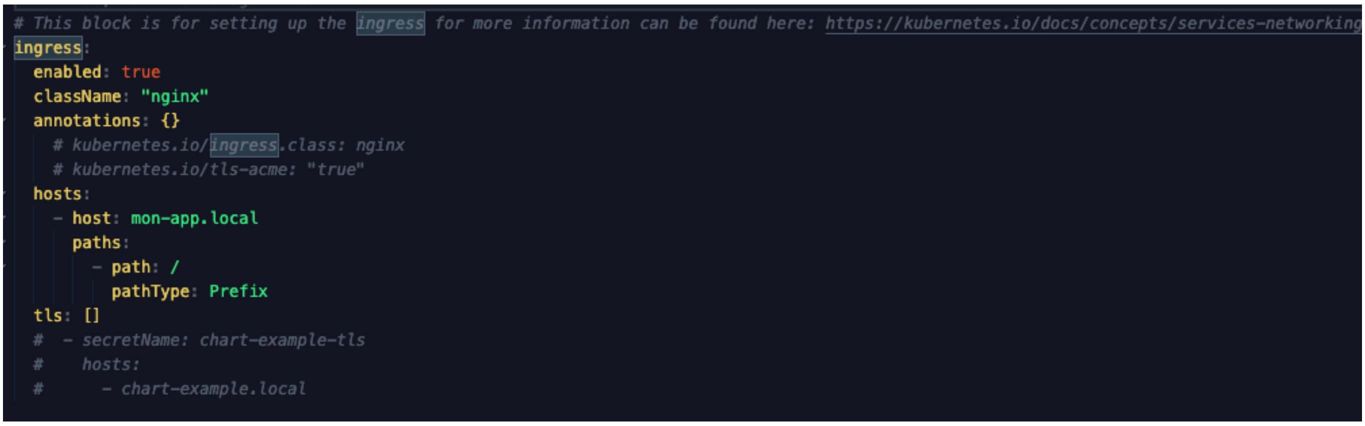
```
charts
templates
.helmignore
Chart.yaml
values.yaml
```

- Modification de la configuration de l'image:



```
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"        You, 1 second ago • Uncommitted changes
```

- Configuration de l'ingress:



```
# 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
```

Installation de l'Ingress Controller

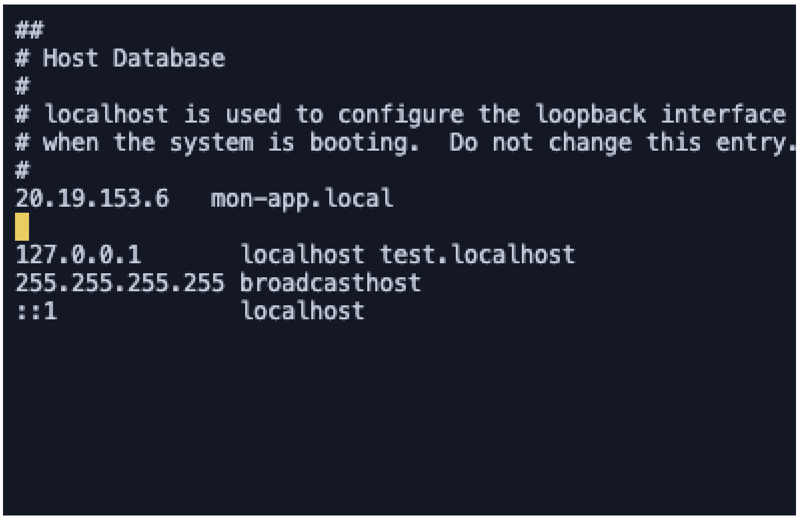
- Ajout du repo et mise à jour:

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

- Installation du contrôleur:

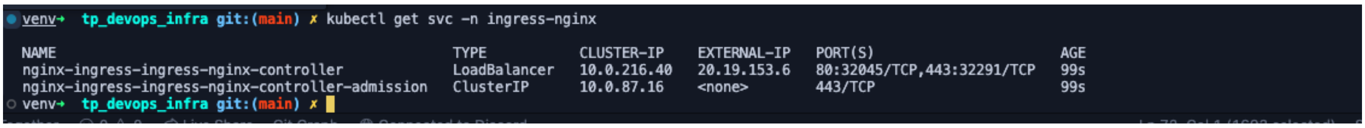
```
helm install nginx-ingress ingress-nginx/ingress-nginx --create-namespace --
namespace ingress-nginx
```

- Vérification de l'IP du service:



```
##
# 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
```

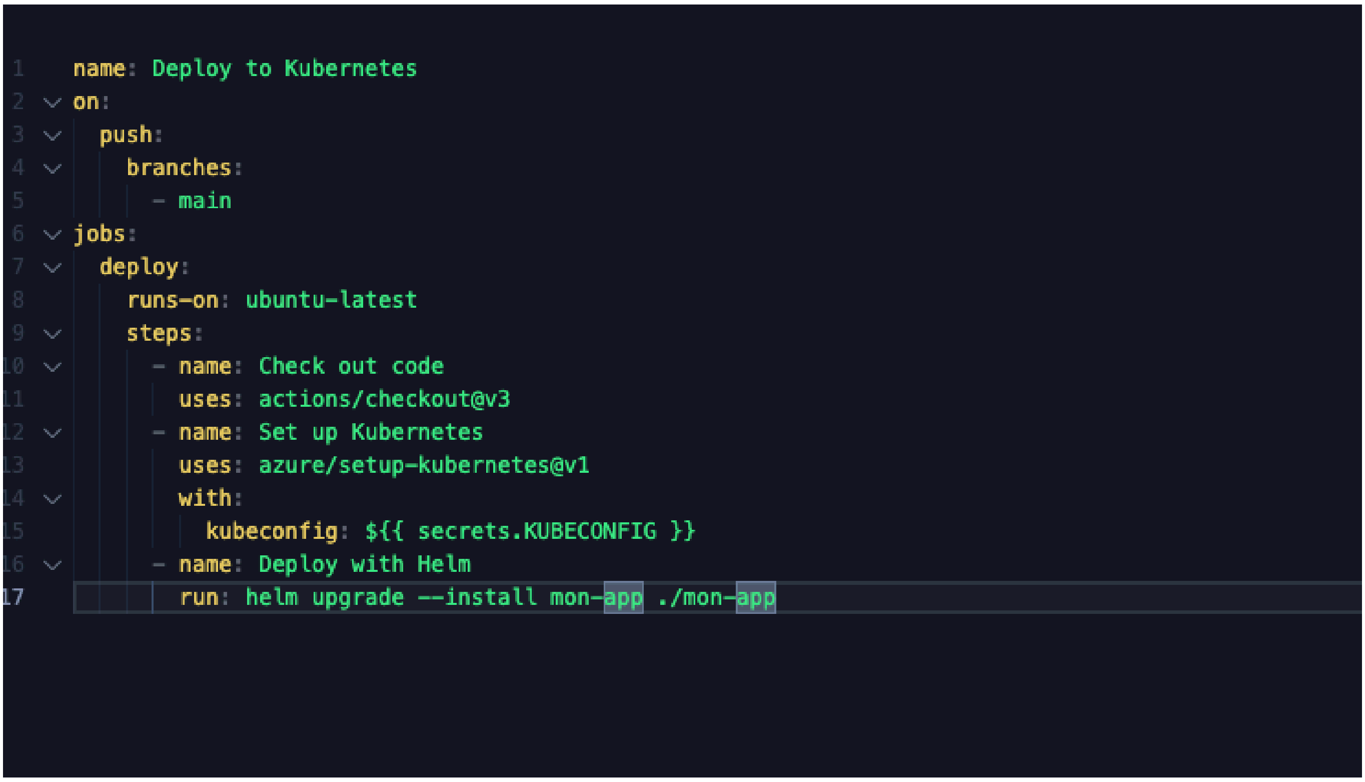
- Ajout au fichier hosts:



```
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) ✗
```

Configuration du CD

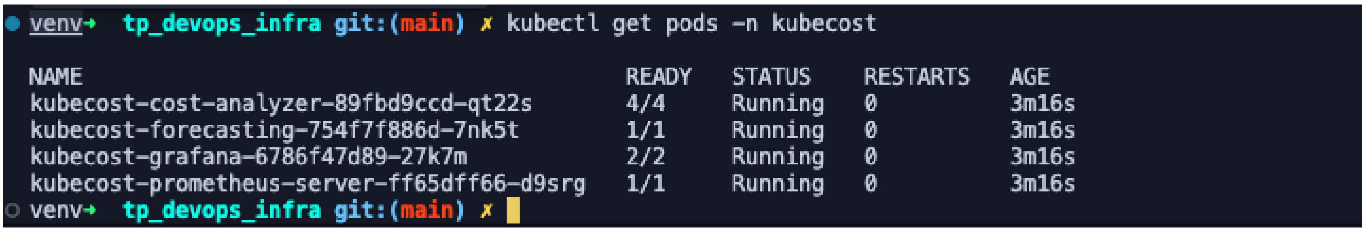
- Ajout du repo Kubecost: `helm repo add kubecost https://kubecost.github.io/cost-analyzer/`
- Configuration du fichier CD pour le déploiement avec Helm:



```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
```

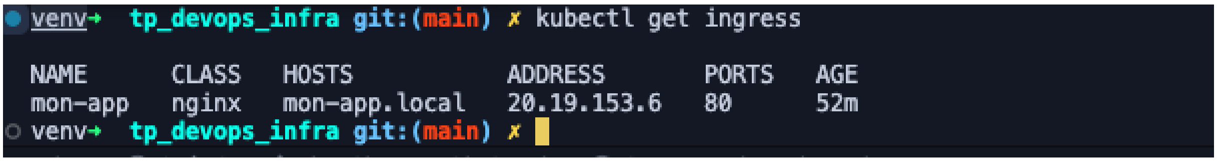
Tests de validation

- Vérification des pods Kubecost:



```
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) ✗
```

- Vérification de l'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) ✗
```