Mise en situation professionnelle	**EasyLinux**
	Cours : Kubernetes
Sujet : Deployer une application	Numéro : 5 à 10 Version : 1.0

Objectifs :

Préparer votre poste de travail

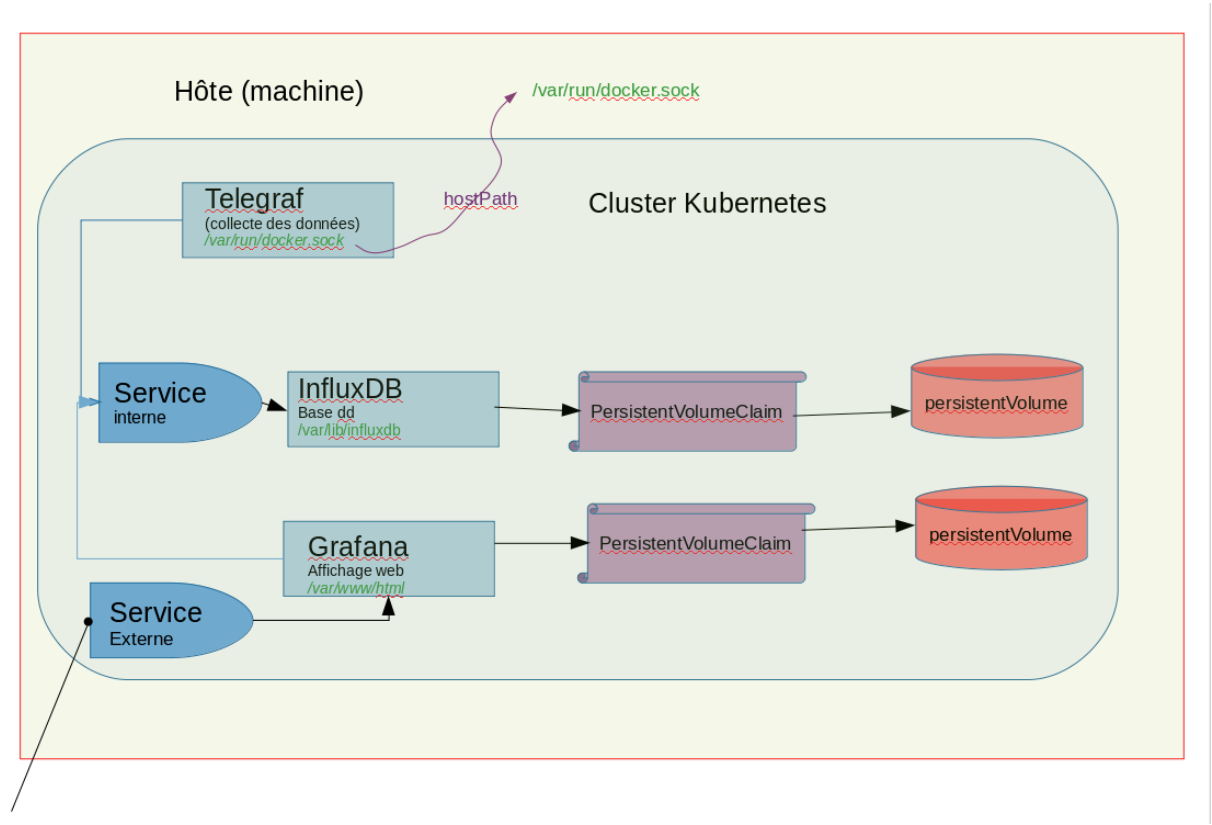
Prérequis :

aucun

Principales tâches à réaliser :

5 Déployer un monitoring du cluster

Nous allons travailler sur l'application suivante



5.1 Créer un service

Nous allons maintenant mettre en production notre application

NB : Il est conseillé de définir les services **avant** de créer les pod / deployment, ...

5.2 Services

1. Réaliser la connexion entre telegraf (composant météo) et InfluxDB via un Service (interne).
2. Créer un service externe (NodePort) pour accéder à grafana.
3. Pousser vos modifications sur le dépôt

6 Deployment

Nous allons convertir nos PODs de l'exercice précédent en deployment. S'aider du document deployment fourni par le formateur.

1. Modifer les trois Pods en conséquence
2. Pousser vos modifications sur le dépôt

Si vous voulez voir le résultat :

NB : avec minikube pour accéder au service, vous devez entrer la commande.

```
$ minikube service <nom du service>
```

7 hostPath

Nos applications stockent leur données dans un répertoire sur une couche RW de AUFS, nous allons pérenniser les données dans un premier temps en local.

Créer l'arborescence sur Minikube

7.1 Telegraf

Copier le fichier telegraf.conf

```
$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___   (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/''`\____)

$ su -
# mkdir /data/telegraf /data/influxdb /data/grafana
# cd /data/telegraf
# wget http://registry.formation.local/Scripts/telegraf.conf
```

Modifier le déploiement pour utiliser 2 hostpath :

- /etc/telegraf vers /data/telegraf
- /var/run/docker.sock vers /var/run/docker.sock

7.2 InfluxDb

Faire pointer la base de données (/var/lib/influxdb) dans /data/influxdb

7.3 Grafana

Faire pointer (/var/lib/grafana) vers /data/grafana

8 Persistent Volume

Le souci avec la configuration précédente, est que en fonctionnement réél, la base de donnée et grafana peuvent être schedulé sur un hôte différent, il vaut mieux définir des volumes persistents

1. Créer 2 persitentVolumeClaim de classe standard
2. Créer et attacher des volumes persistents pour Grafana et InfluxDB

Confirmer après plusieurs suppression du POD influxdb ou grafana que vos données sont toujours présentes.

9 Gestion des secrets et de la configuration

Nous allons modifier notre projet pour stocker les identifiants d'InfluxDB dans des Secret.

Créer un Secret contenant le mot de passe, le compte et le nom de la base.

Utiliser ce Secret pour injecter les valeurs dans les variables d'environnement des PODs InfluxDB, météo et Telegraf.

Résumé

Mettre en œuvre WordPress avec les images suivantes :

registry.formation.local:5000/easylinux/mariadb

registry.formation.local:5000/easylinux/wordpress

10 Solution

10.1 Services

a 01-SvcInflux.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: svc-influx
  labels:
    app: influxdb
spec:
  ports:
  - port: 8086
    protocol: TCP
    targetPort: 8086
  selector:
    app: dep-influxdb
```

b 02-SvcGrafana.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: svc-grafana
  labels:
    app: grafana
spec:
  ports:
  - port: 3000
    protocol: TCP
    targetPort: 3000
  type: NodePort
  selector:
    app: dep-grafana
```

10.2 Deployment

a 03-DepInfluxDb.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dep-influxdb
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dep-influxdb
  template:
    metadata:
      labels:
        app: dep-influxdb
    spec:
      containers:
        - name: influx
          image: registry.formation.local:5000/easylinux/influxdb:1.7
          env:
            - name: INFLUXDB_DB
              value: "db0"
            - name: INFLUXDB_USER
              value: "grafana"
            - name: INFLUXDB_PASSWORD
              value: "Secr3t"
          ports:
            - name: http
              protocol: TCP
              containerPort: 8086
```

b 04-DepTelegraf.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dep-telegraf
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dep-telegraf
  template:
    metadata:
      labels:
        app: dep-telegraf
    spec:
      containers:
        - name: telegraf
          image: registry.formation.local:5000/easylinux/telegraf
          imagePullPolicy: IfNotPresent
          env:
          - name: INFLUXDB_DB
            value: "db0"
          - name: INFLUXDB_USER
            value: "telegraf"
          - name: INFLUXDB_PASSWORD
            value: "Secr3t"
          - name: INFLUXDB_HOST
            value: "svc-influxdb"
          - name: INFLUXDB_PORT
            value: "8086"
```

c 05-DepGrafana.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: dep-grafana
spec:
  replicas: 1
  selector:
    matchLabels:
      app: dep-grafana
  template:
    metadata:
      labels:
        app: dep-grafana
    spec:
      containers:
        - name: grafana
          image: registry.formation.local:5000/easylinux/grafana
          ports:
            - name: http
              containerPort: 3000
```

Penser à git commit …

10.3 hostPath

Créer les répertoires de destination

```
$ minikube ssh
                         _             _
            _         _ ( )           ( )
  ___ ___  (_)  ___  (_)| |/')  _   _ | |_      __
/' _ ` _ `\| |/' _ `\| || , <  ( ) ( )| '_`\  /'__`\
| ( ) ( ) || || ( ) || || |\`\ | (_) || |_) )(  ___/
(_) (_) (_)(_)(_) (_)(_)(_) (_)`\___/'(_,__/''`\____)

$ su -
# mkdir /data/telegraf /data/influxdb /data/grafana
# cd /data/telegraf
# wget http://registry.formation.local/Scripts/telegraf.conf
```

a Modification de 04-DepTelegraf.yaml

```
          volumeMounts:
            - mountPath: /var/run/docker.sock
              name: docker-sock
            - mountPath: /etc/telegraf
              name: conf-telegraf

      volumes:
        - name: docker-sock
          hostPath:
            path: /var/run/docker.sock
            type: Socket
        - name: conf-telegraf
          hostPath:
            path: /data/telegraf
            type: Directory
```

b Modification de 03-DepInflux.yaml

```
          volumeMounts:
            - mountPath: /var/lib/influxdb
              name: influx-db

      volumes:
        - name: influx-db
          hostPath:
            path: /data/influxdb
            type : Directory
```

c Modification de 05-DepGrafana.yaml

```
          volumeMounts:
            - mountPath: /var/lib/grafana
              name: grafana-hp

      volumes:
        - name: grafana-hp
          hostPath:
            path: /data/grafana
            type: Directory
```

10.4 PersistentVolume

Commencer par créer les persistentVolumeClaim

a 00-VolInfluxGrafana.yaml

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-influx
  labels:
    app: dep-influxdb
spec:
  storageClassName: standard
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 1Gi
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: claim-grafana
  labels:
    app: dep-grafana
spec:
  storageClassName: standard
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 1Gi
```

b 03-DepInfluxDb.yaml

```
        volumeMounts:
          - mountPath: /var/lib/influxdb
            name: influx-claim

      volumes:
        - name: influx-claim
          persistentVolumeClaim:
            claimName: claim-influxdb
```

c 05-DepGrafana.yaml

```
        volumeMounts:
          - mountPath: /var/lib/grafana
            name: grafana-claim

      volumes:
        - name: grafana-claim
          persistentVolumeClaim:
            claimName: claim-grafana
```

10.5 Gestion des secrets

Créer un fichier 06-Secrets.yaml

```
apiVersion: v1
kind: Secret
metadata:
  name: influx-secret
type: Opaque
data:
  influxdb_password: PASSWD
  influxdb_db: DBNAME
  influxdb_user: USER
```

Remplacer les données par les bonnes valeurs en base 64

```
$ sed -i "s/PASSWD/$(echo -n Secr3t| base64)/" 06-Secrets.yaml
$ sed -i "s/DBNAME/$(echo -n db0| base64)/" 06-Secrets.yaml
$ sed -i "s/USER/$(echo -n grafana| base64)/" 06-Secrets.yaml
```

Les enregistrer par Kubernetes

```
kubectl create -f 06-Secrets.yaml
```

a 03-DepInfluxDb.yaml

```
            - name: INFLUXDB_DB
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_db
            - name: INFLUXDB_USER
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_user
            - name: INFLUXDB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_password
```

b 04-DepTelegraf.yaml

```
            - name: INFLUXDB_DB
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_db
            - name: INFLUXDB_USER
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_user
            - name: INFLUXDB_PASSWORD
              valueFrom:
                secretKeyRef:
                  name: influx-secret
                  key: influxdb_password
```