



Le génie pour l'industrie

Patrick Bertin TALLA TAMADJE(TALP19077806)
Nadine Carelle MASSUDOM MGNIE(MASN29548301)
MOISE MAHARA MOUKENE (MAHM22089208)

Introduction à l'approche DevOps

Cours: Log680.
Equipe : 07
Laboratoire 3

Travail présenté à
Vincent Boivin

Remis le 28 Juillet 2021
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

1. INTRODUCTION

Le présent document a pour but de décrire ce que nous avons réalisé dans le cadre du laboratoire 3 qui consistait de poursuivre la mise en place d'un pipeline d'intégration et le déploiement continu et le monitoring de données que nous avons commencé au laboratoire 2.

Le but principal de notre travail était de faire le déploiement d'applications en production avec le cluster Kubernetes. Pour atteindre l'objectif de ce laboratoire, nous avons structuré notre travail en trois étapes: Premièrement, nous avons créé des ressources kubernetes pour déployer notre image Docker sur le cluster pour chaque mise à jour. Ensuite nous avons créé une base de données Mysql sur AWS qui sera utilisée par Grafana pour faire le monitoring.

Enfin nous avons apporté des modifications au code source pour envoyer les données de température et les événements à la base de données.

2. Explication du fonctionnement du déploiement sur Kubernetes

Pour faire le déploiement sur kubernetes, nous avons utilisé notre image de container du laboratoire 2. Dans le dossier `manifests-k8s`, nous avons créé nos différentes ressources telles que Kubernetes Secrets, ConfigMap et *Deployment* en ligne de commande. Pour créer nos *secrets* pour nos variables d'environnement, nous avons utilisé la commande suivante:

```
kubectl create secret generic hvac_secret --from-literal=DATABASE=tp3
--from-literal=HOST=ec2-3-237-178-114.compute-1.amazonaws.com
--from-literal=PASSWORD=Tp@3!55.
--from-literal=TOKEN=f0c51c904ed6dd637b2f --from-literal=USER=tp3
--dry-run=client -o yaml > manifests-k8s/hvac-secret.yaml
```

Pour créer la *configMap* pour nos variables d'environnement, nous avons utilisé la commande suivante:

```
kubectl create configmap hvac_configmap --from-literal=LIMITCHAUD=60
--from-literal=LIMITFROID=40 --from-literal=NBTICK=6 --dry-run=client -o yaml >
manifests-k8s/hvac-configmap.yaml.
```

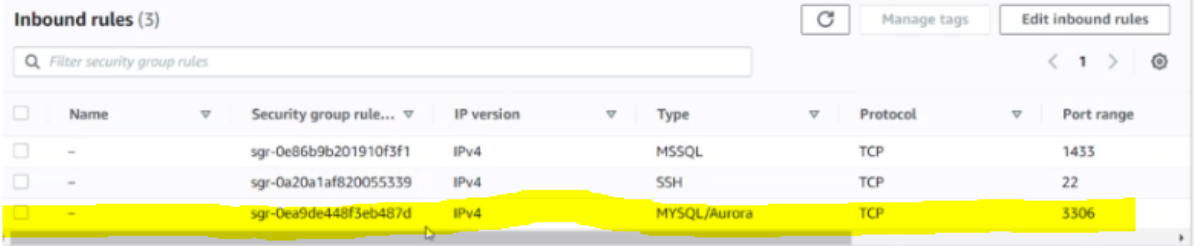
Pour créer le *Deployment* qui doit nous permettre de faire la maintenance de notre container dans le cluster, nous avons utilisé la commande suivante:

```
kubectl create deployment hvac_deployment
--image=etslog680/image_hvac:latest --dry-run=client -o yaml >
.\manifests-k8s\hvac-deployment.yaml
```

3. Construction de l'infrastructure AWS

En se servant de l'image de VM EC2 AWS qui était créée au préalable par l'ETS, nous avons créé notre propre image. Au début, il était impossible de se connecter à la Base de

données de la VM; Le port ssh existant par défaut se limitant seulement à l'accès de la VM, alors pour autoriser les trafics dans la VM, nous avons créé des règles de sécurité. Pour le trafic entrant, une règle de sécurité a été ajoutée dans le Security group, en définissant le nom du protocole, le numéro du port et l'adresse du réseau. Nous avons autorisé le trafic pour le MySQL en direction du port 3306.



	Name	Security group rule...	IP version	Type	Protocol	Port range
<input type="checkbox"/>	-	sgr-0e86b9b201910f3f1	IPv4	MSSQL	TCP	1433
<input type="checkbox"/>	-	sgr-0a20a1af820055339	IPv4	SSH	TCP	22
<input type="checkbox"/>	-	sgr-0ea9de448f3eb487d	IPv4	MYSQL/Aurora	TCP	3306

Après la configuration du VM, nous avons accédé à la machine en saisissant la commande suivante dans notre terminal:

```
ssh -i "log680.pem" ubuntu@ec2-3-237-178-114.compute-1.amazonaws.com
```

Après cet étape, nous avons accédé à la base de données à l'aide des accès ci après:

```
Vous avez accès à votre base de données avec
user: 'tp3'
password: 'Tp@3!55.'
db: 'tp3'
```

Nous avons utilisé la base de données nommée TP3 pour créer nos deux tables log680_EVENS et log680_tp3 avec MySQL. La table log680_EVENS permet de stocker les données des événements liés à l'activation ou la désactivation de la climatisation. La table log680_tp3 enregistre les données concernant la variation de la température en fonction de l'heure.

3. Modifications dans le code

- Pour se connecter à la base de données nous avons apporté des changements de la ligne 27 à 36

```
23     self.NBTICK = nbTick
24     self.LIMITCHAUD = limitChaud
25     self.LIMITFROID = limitFroid
26     self.TOKEN = mytoken
27     self.USER = user
28     self.PASSWORD = password
29     self.HOST = host
30     self.DATABASE = database
31     # connection to the DB
32     self.mydb = mysql.connect(
33         user = user,
34         password = password,
35         host = host,
```

- De la ligne 32 à 35 nous avons fait des modifications afin de tester si la connexion est effective à la base de donnée, dans le cas contraire, affiche un message d'erreur.

```

32         if self.mydb.cursor:
33             print("Connection etablie avec la DB")
34         else :
35             print("connection failed")
36
37     def __del__(self):

```

- De la ligne 117 à 122 , nous avons fait les modifications en fin de Créé une fonction qui permet d'envoyer les données à la table log680_EVENS avec les valeurs de l'événement et d'heure recueillies en ligne

```

117     # send events data to mysql
118     def sendEventsToMysql(self,date, ev):
119
120         self._extracted_from_sendEventsToMysql_3(
121             "INSERT INTO log680_EVENS(heure,eventement) VALUES(%s,%s)", date, ev
122         )
123

```

- De la ligne 98 à 109 , nous avons fait les modifications afin de créer une fonction qui permet d'envoyer les données à la table log680_tp3 avec les valeurs de température et d'heure recueillies en ligne .

```

# send temperature data to mysql
def sendDataToMysql(self,date, temp):

    self._extracted_from_sendEventsToMysql_3(
        "INSERT INTO log680_tp3(heure,temperature) VALUES(%s,%s)",
        date,
        temp,
    )

```

- La ligne 85 a été modifiée pour appeler la fonction qui envoie la température.

```

84         #- envois des données( on envoi la temperature
85         self.sendDataToMysql(date,dp)
86

```

- La ligne 105, nous avons fait l'appel de la fonction pour envoyer les événements à la table log680_EVENS

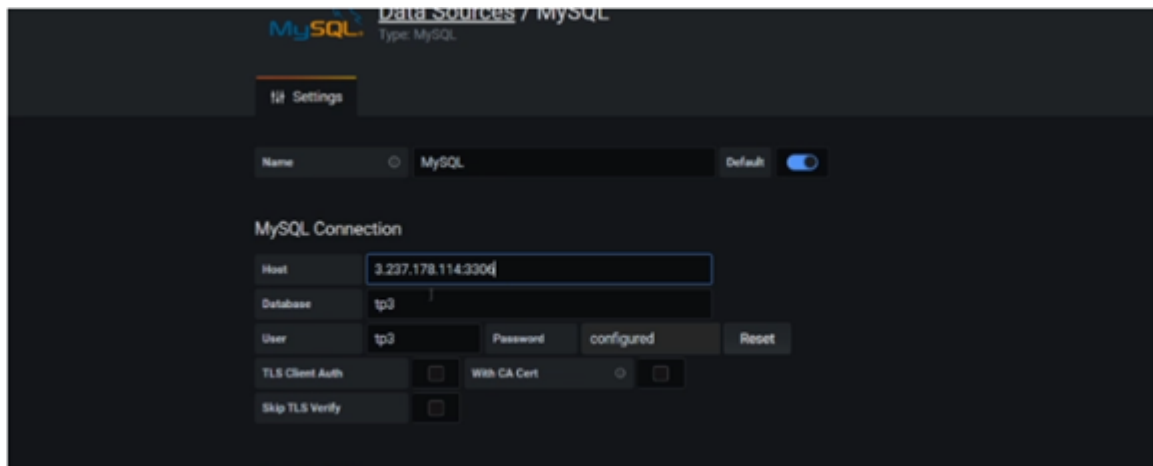
```

104         # envoi des evenements vers la table log680_EVENS
105         self.sendEventsToMysql(date, str(details))
106

```

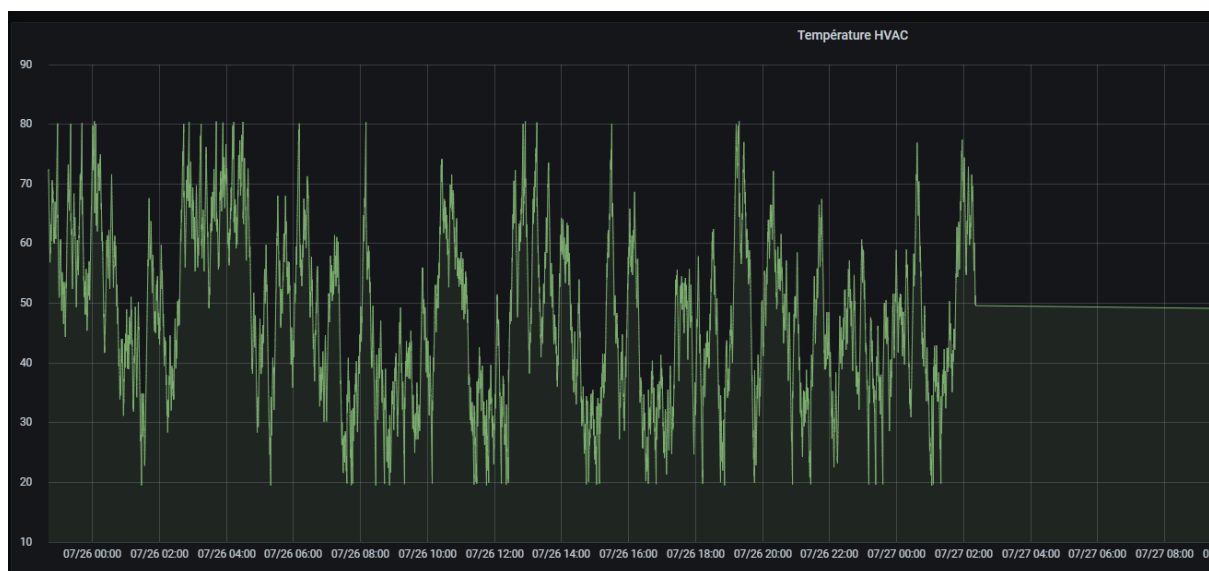
4. Utilisation de Grafana

Dans grafana, nous avons ajouté une source de donnée et avons établi la connexion avec la base de données en effectuant la configuration suivante:



Une fois la connexion établie , nous avons configuré notre tableau de bord Grafana pour nous permettre d'interroger, de visualiser, d'alerter et d'explorer les données de manière globale.

Le premier graphique présenté ci dessous, l'axe des ordonnées Y représente la température et l'axe des abscisses X l'heure. Ce graphique nous permet de visualiser facilement et rapidement les valeurs maximales et minimales de la température. Avec ce graphique , il est plus aisé d'explorer nos métriques car le temps est graduel et les variations de température se distinguent aisément.



Par contre avec le deuxième tableau de bord ci-dessous, il est plus difficile d'explorer les données car cela demande de parcourir tout le tableau afin de déterminer les valeurs maximales et minimales. Nous avons opté pour le tableau à deux dimensions pour les événements, car cela s'affiche comme des fichiers logs.

Evenements	
time	evenement
2021-07-25 22:57:41	{'Response': 'Activating AC for 4 ticks'}
2021-07-25 23:21:07	{'Response': 'Activating AC for 4 ticks'}
2021-07-25 23:41:22	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 00:03:37	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 00:06:22	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 01:27:58	{'Response': 'Activating Heater for 4 ticks'}
2021-07-26 02:43:33	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 02:53:09	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 03:14:39	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 03:41:49	{'Response': 'Activating AC for 4 ticks'}
2021-07-26 03:53:44	{'Response': 'Activating AC for 4 ticks'}

5. CONCLUSION

En somme , pour ce dernier laboratoire nous avons eu pour mandat de faire du déploiement continue sur kubernetes à l'aide d' AWS et aussi de faire le monitoring de données avec Grafana. La tâche n'a pas été évidente, car nous n'étions pas familiers à la plateforme AWS, Kubernetes et grafana. Avec un peu d'effort et de travail, nous avons complété le mandat qui nous avait été assigné. En outre, le laboratoire nous a permis de nous familiariser avec les différents environnements de déploiement et d'intégration continue.