Interview Assignment For DevOps Engineer

• Cluster MariaBD – Kubernetes

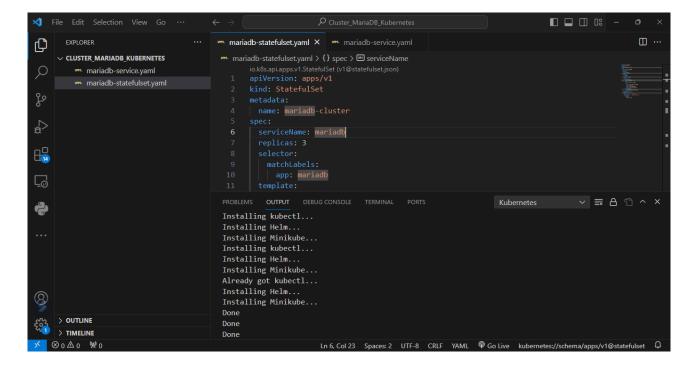
Requirements:

- Docker CE
- Minikube ou Microk8s
- Linux Ubuntu 20.04 (Preferably higher than version 18)
- GitHub (https://github.com/wagnersoftwareengineer/Cluster_MariaDB_Kubernetes)
- VsCode

I initialy installed the softwares, below evidence about Microk8s or Minikube:

```
agner@Wagner:~$ sudo chown
agner@Wagner:~$ su - $USER
                                                                                                                                  -f -R $USER ~/.kube
Password:
wagner@Wagner:~$ microk8s status --wait-ready
microk8s is running
high-availability: no
datastore master nodes: 127.0.0.1:19001
datastore standby nodes: none
  addons:
enabled:
dns
                                                                                                             # (core) CoreDNS
# (core) Configure high availability on the current node
# (core) Helm - the package manager for Kubernetes
# (core) Helm 3 - the package manager for Kubernetes
                  ha-cluster
                                                                                                         # (core) HeLM 3 - The package manager for Kubernetes

# (core) Cloud native certificate management
# (core) Apply CIS k8s hardening
# (core) The community addons repository
# (core) The Kubernetes dashboard
# (core) Allow Pods connecting to Host services smoothly
# (core) Allow Pods connecting to Host services smoothly
# (core) Storage class; allocates storage from host directory
# (core) Ingress controller for external access
# (core) An advanced network fabric for Kubernetes
# (core) OpenEBS MayaStor
# (core) Loadbalancer for your Kubernetes cluster
# (core) K8s Metrics Server for API access to service metrics
# (core) MinIO object storage
# (core) A lightweight observability stack for logs, traces and metrics
# (core) Prometheus operator for monitoring and logging
# (core) Prometheus operator for monitoring and logging
# (core) Pole-Eased Access Control for authorisation
# (core) Private image registry exposed on localhost:32000
# (core) Distributed Ceph storage using Rook
# (core) Alias to hostpath-storage add-on, deprecated
       disabled:
                cert-manager
cis-hardening
                community
dashboard
                gpu
host-access
                  hostpath-storage
                mayastor
metallb
metrics-server
                  minio
                minio
observability
prometheus
                registry
rook-ceph
                  storage
                                               ner:~$
```



Then, i create new files about MariaDB configurations inside Yaml and by terminal we can use some commands to start. Attention, you need stay inside the folder to run commands.

microk8s kubectl apply -f mariadb-statefulset.yaml microk8s kubectl apply -f mariadb-service.yaml

Here, created MariaDB Server:

```
wagner@Wagner:/mnt/c/Users/Wagner Santos/Documents/Cluster_MariaDB_Kubernetes$ microk8s kubectl apply -f mariadb-statefulset.yaml statefulset.apps/mariadb-cluster created wagner@Wagner:/mnt/c/Users/Wagner Santos/Documents/Cluster_MariaDB_Kubernetes$ microk8s kubectl apply -f mariadb-service.yaml service/mariadb created wagner@Wagner:/mnt/c/Users/Wagner Santos/Documents/Cluster_MariaDB_Kubernetes$ |
```

A change was made to Yaml to expose the IP for external access, so I modified it to run again.

Command delete:

Server created:

Next page I share the Yamls:

```
EXPLORER
                                              mariadb-statefulset.yaml
                                                                                mariadb-service.yaml X

∨ CLUSTER_MARIADB_KUBERNETES

    · mariadb-service.yaml
                                                       metadata:
    mariadb-statefulset.yaml
                                                        name: mariadb
                                                         selector:
                                                           app: mariadb
                                                          ports:
                                                               port: 3306
                                                               targetPort: 3306
                                                 12
                                                         type: NodePort
    mariadb-statefulset.yaml > {} spec > [ ] volumeClaimTemplates > {} 0 > {} metadata
          io.k8s.api.apps.v1.StatefulSet (v1@statefulset.json)
apiVersion: apps/v1
           name: mariadb-cluster
              metadata:
                 app: mariadb
               - name: mariadb
                 image: mariadb:latest
                - containerPort: 3306
env:
- name: MYSQL_ROOT_PASSWORD
value: "master"
                 - name: MYSQL_DATABASE value: "DevopsDatabase"
                  volumeMounts:
                  - name: mariadb-storage
                   mountPath: /var/lib/mysql
            - metadata:
     30
               name: mariadb-storage
               accessModes: [ "ReadWriteOnce" ]
```

To test access:

- 1 List the created pods
- 2 Access via kubectl exec command
- 3 After that, I can use the MariaDB client, which is MySQL
- 4 -Now connected, I can run any command on the MariaDB Database Client that was applied in Yaml.

Note: I installed the Client to connect to the Bank. In my case the "DevOpsDatabase" database.

```
root@mariadb-cluster-0:/# mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 11.1.2-MariaDB-1:11.1.2+maria~ubu2204 mariadb.org binary distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```