

Kubernetes üzerine MySQL kurulumu

MySQL Kurulumu

Manifest Dosyalarını kullanarak yükleme:

Yaml dosyalarını kullanarak MySQL'i kurarken özelleştirmişte olacağız.

MySQL Operator for Kubernetes tarafından kullanılan Custom Resource Definition (CRD) kurun:

CRD, Kubernetes API'sini genişleten veya kendi API'nizi bir projeye veya kümeye tanıtmaya olanak tanıyan bir nesnedir. Özel bir kaynak tanımlama (CRD) dosyası, kendi nesne türlerinizi tanımlar ve API Sunucusunun tüm yaşam döngüsünü yönetmesini sağlar.

```
kubectl apply -f https://raw.githubusercontent.com/mysql/mysql-operator/trunk/deploy/deploy-crds.yaml
```

RBAC tanımlarını da içeren Kubernetes için MySQL Operator'ü dağıtın:

Rol tabanlı erişim denetimi (RBAC), kuruluşunuzdaki bireysel kullanıcıların rollerine dayalı olarak bilgisayar veya ağ kaynaklarına erişimi düzenleme yöntemidir.

```
kubectl apply -f https://raw.githubusercontent.com/mysql/mysql-operator/trunk/deploy/deploy-operator.yaml
```

deploymentı oluşup oluşmadığını görmek için get kullandık.

```
kubectl get deployment mysql-operator --namespace mysql-operator
```

MySQL InnoDB Cluster

veritabanında oluşan yükü birden fazla sunucuya dağıtarak çok daha yüksek performanslar elde etmenizi sağlayan bir veritabanı çeşitidir.

InnoDB , hata toleransı yüksek sistemleri oluşturmak için kullanılmaktadır.

kubectl ile bir InnoDB Kümesi oluşturmak için, önce yeni bir MySQL kök kullanıcısı için kimlik bilgilerini içeren bir secret oluşturun.

```
kubectl create secret generic mypwds \
  --from-literal=rootUser=root \
  --from-literal=rootHost=% \
  --from-literal=rootPassword="parlak"
```

Şimdi istediğimiz gibi yaml dosyası oluşturabiliriz.

```
vim mycluster.yaml
```

InnoDBCluster tanımı, üç MySQL sunucu örneği ve bir MySQL Router örneği oluşturur:

```
apiVersion: mysql.oracle.com/v2
kind: InnoDBCluster
metadata:
  name: mycluster
spec:
  secretName: mypwds
  tlsUseSelfSigned: true
  instances: 3
  router:
    instances: 1
```

```
kubectl apply -f mycluster.yaml # kendi özel clusterımızı çalıştırdık.
```

Clusterin durumunu kontrol etmek için

```
kubectl get innodbcluster --watch
```

```
root@mysqlserver:/home/selman# kubectl get innodbcluster --watch
NAME          STATUS    ONLINE INSTANCES  ROUTERS  AGE
mycluster     PENDING  0        3          1        19s
mycluster     INITIALIZING  0        3          1        118s
mycluster     INITIALIZING  0        3          1        2m8s
mycluster     INITIALIZING  0        3          1        2m14s
mycluster     INITIALIZING  0        3          1        2m18s
mycluster     INITIALIZING  0        3          1        2m19s
mycluster     INITIALIZING  0        3          1        2m19s
mycluster     INITIALIZING  0        3          1        2m19s
mycluster     INITIALIZING  0        3          1        2m19s
mycluster     INITIALIZING  0        3          1        2m25s
mycluster     ONLINE    1        3          1        2m26s
mycluster     ONLINE    1        3          1        2m30s
mycluster     ONLINE    2        3          1        2m40s
mycluster     ONLINE    2        3          1        2m46s
mycluster     ONLINE    3        3          1        2m59s
```

artık MySQL'e bağlanabiliriz.

```
kubectl run --rm -it myshell --image=mysql/mysql-operator -- mysqlsh root@mycluster --
sql
```

MySQL InnoDB Kümesini yönetmek için MySQL Shell ile yeni bir kapsayıcı oluşturduk.

oluşturduğumuz secret dosyasındaki parola isteniyor. Benim şifrem “parlak”

```
If you don't see a command prompt, try pressing enter.
Please provide the password for 'root@mycluster': *****
Save password for 'root@mycluster'? [Y]es/[N]o/[e]ver (default No): Y
MySQL Shell 8.0.30

Copyright (c) 2016, 2022, 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 '\?' for help; '\quit' to exit.
Creating a session to 'root@mycluster'
Fetching schema names for autocompletion... Press ^C to stop.
Your MySQL connection id is 867 (X protocol)
Server version: 8.0.30 MySQL Community Server - GPL
No default schema selected; type \use <schema> to set one.
MySQL mycluster:33060+ ssl SQL >
```

Helm kullanarak WordPress kurulumu:

Kubernetes üzerine wordpress kurulumu için helm'i tercih ettim. Helm bizim için Kubernetes ortamında çalışan uygulamalarımızın kaynaklarını (deployment, statefulset, service, ingress vb.) kolayca yönetebilmemizi ve karmaşıklıklardan kurtulmamızı sağlar. Paket yöneticisi gibi düşünebiliriz. Ubuntu'daki apt gibi.

Helm Kurulumu:

```
curl https://baltocdn.com/helm/signing.asc | gpg --dearmor | sudo tee /usr/share/keyrings/helm.gpg > /dev/null
sudo apt-get install apt-transport-https --yes
echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/helm.gpg] https://baltocdn.com/helm/stable/debian/ all main" | sudo tee /etc/apt/sources.list.d/helm-stable-debian.list
sudo apt-get update
sudo apt-get install helm
```

Helm'i kurduktan sonra sunucu veya sisteminizde docker ve k3s olması gerekiyor. Yoksa aşağıdaki gibi bir hata alabilirsiniz.

```
root@wordpress:/home/selman# helm install my-release bitnami/wordpress
Error: INSTALLATION FAILED: Kubernetes cluster unreachable: Get "http://localhost:8080/version": dial tcp 127.0.0.1:8080: connect: connection refused
root@wordpress:/home/selman#
```

yine hata alıyorsanız şu kodu deneyebilirsiniz.

```
export KUBECONFIG=/etc/rancher/k3s/k3s.yaml
```

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

```
helm install my-release \
  --set auth.rootPassword=secretpassword,auth.database=app_database \
  --set auth.username=selman \
  --set auth.password=parlak \
  bitnami/mysql
```

yaml dosyasıyla da kurulum yapabiliriz.

```
helm install my-release -f values.yaml bitnami/mysql
```

Helm parametreleri :

`global.storageClass` = kalıcı depolama birimleri için StorageClass.

`image.registry` = image dosyası kaynağı (docker.io).

`image.repository` = image dosyası (bitnami/mysql).

`auth.database` = oluşturulacak veritabanının adı.

`auth.username` = oluşturulacak kullanıcının adı.

`auth.password` = oluşturulan kullanıcı için şifre.rootpassword sağlanmışsa yok sayılabilir.

`primary.persistence.storageClass` = birincil kalıcı deploma birimi için StorageClass.

`primary.persistence.accessModes` =birincil kalıcı depolama birimi için erişim Modları.
varsayılan(ReadWriteOnce).

`primary.persistence.size` = birincil kalıcı depolama birimi boyutu.

```
NAME: my-releases
LAST DEPLOYED: Tue Sep  6 13:48:19 2022
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: mysql
CHART VERSION: 9.3.1
APP VERSION: 8.0.30

** Please be patient while the chart is being deployed **

Tip:
  Watch the deployment status using the command: kubectl get pods -w --namespace default

Services:

  echo Primary: my-releases-mysql.default.svc.cluster.local:3306

Execute the following to get the administrator credentials:

  echo Username: root
  MYSQL_ROOT_PASSWORD=$(kubectl get secret --namespace default my-releases-mysql -o jsonpath="{.data.mysql-root-password}" | base64 -d)

To connect to your database:

  1. Run a pod that you can use as a client:

      kubectl run my-releases-mysql-client --rm --tty -i --restart='Never' --image docker.io/bitnami/mysql:8.0.30-debian-11-r6 --namespace default --env MYSQL_ROOT_PASSWORD=$MYSQL_ROOT_PASSWORD --command -- bash

  2. To connect to primary service (read/write):

      mysql -h my-releases-mysql.default.svc.cluster.local -uroot -p"$MYSQL_ROOT_PASSWORD"
```

1. Run a pod that you can use as a client:

```
kubectl run my-releases-mysql-client --rm --tty -i --restart='Never' --image do
cker.io/bitnami/mysql:8.0.30-debian-11-r6 --namespace default --env MYSQL_ROOT_PASSWOR
D=$MYSQL_ROOT_PASSWORD --command -- bash
```

```
I have no name!@my-releases-mysql-clienttt:/$
```

2. To connect to primary service (read/write):

```
mysql -h my-releases-mysql.default.svc.cluster.local -uroot -p"$MYSQL_ROOT_PASSWORD"
```

```
I have no name!@my-releases-mysql-clientt:/$ mysql -h my-releases-mysql.default.svc.cluster.local -uroot -p"$MYSQL_ROOT_PASSWORD"
Enter password:
```

`auth.rootPassword=secretpassword` = secretpassword

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 414
Server version: 8.0.30 Source distribution

Copyright (c) 2000, 2022, 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> select *
      -> SELECT *
      -> \c
mysql>
```

```
mysql> SHOW DATABASES;
+-----+
| Database |
+-----+
| app_database |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.29 sec)
```