# Troubleshooting

## Kubernetes cluster

### **Troubleshooting the control plane**

### **Listing the nodes in a cluster**

### First thing to check if your cluster is working fine or not is to list the nodes associated with your cluster.

### kubectl **get** nodes

### 

### Make sure that all nodes are in Ready state.

### **List the control plane pods**

### If your nodes are up and running, the next thing to check is the status of Kubernetes components. Run,

### kubectl **get** pods -n kube-system

### 

### If any of the pods are *restarting or crashing*, look into the issue. This can be done by getting the pod's description. For example, in my cluster *kube-dns* is crashing. In order to fix this first check the deployment for errors.

### kubectl **describe** deployment -n kube-**system** kube-dns

### 

### Log files

### Master

### If your deployment is good, the next thing to look for is log files. The locations of log files are given below...

### /var/log/containers/kube-apiserver.log - For API Server logs

### /var/log/containers/kube-scheduler.log - For Scheduler logs

### /var/log/containers/kube-controller-manager.log - For Controller logs

### 

### 

## Worker Nodes

### In your worker, you will need to check for errors in kubelet's log...

### sudo journalctl -u kubelet

### sudo service kubelet status

### If not running,

### sudo service kubelet restart

### 

### 

### 

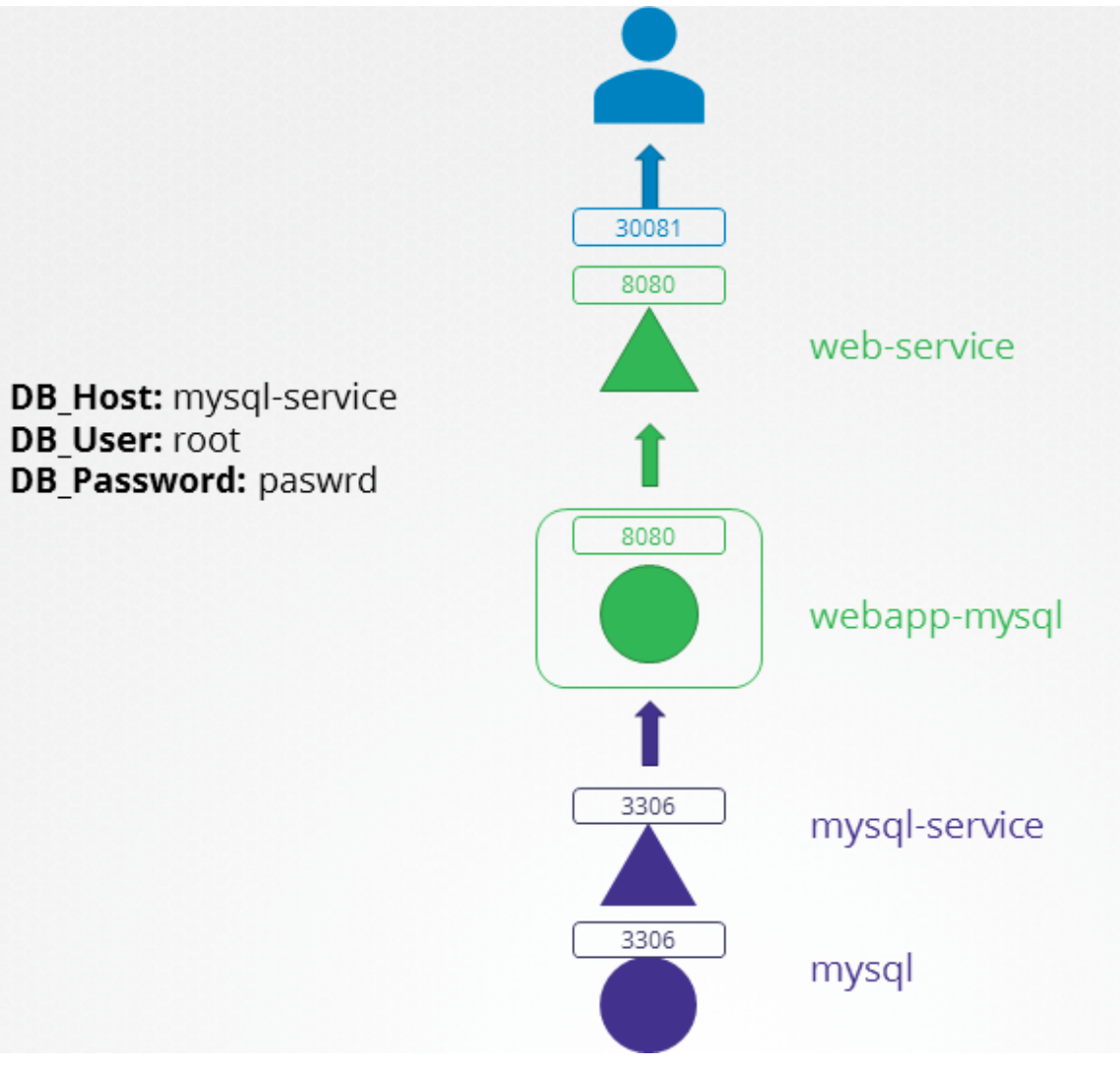
### 

### 

### 

### 

### 2-Tier App



Access the app url - http://localhost:30081 for troubleshooting

k create ns alpha

k apply -f ex1

k get all -n alpha

curl http://localhost:30081

Ex 1:

apiVersion: v1

kind: Pod

metadata:

labels:

name: mysql

name: mysql

namespace: alpha

spec:

containers:

- env:

- name: MYSQL\_ROOT\_PASSWORD

value: paswrd

image: mysql:5.6

imagePullPolicy: IfNotPresent

name: mysql

ports:

- containerPort: 3306

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

**name: mysql**

namespace: alpha

spec:

ports:

- port: 3306

targetPort: 3306

selector:

name: mysql

---

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

namespace: alpha

spec:

replicas: 1

selector:

matchLabels:

name: webapp-mysql

template:

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

spec:

containers:

- env:

- name: DB\_Host

value: mysql-service

- name: DB\_User

value: root

- name: DB\_Password

value: paswrd

image: mmumshad/simple-webapp-mysql

imagePullPolicy: Always

name: webapp-mysql

ports:

- containerPort: 8080

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: web-service

namespace: alpha

spec:

ports:

- port: 8080

targetPort: 8080

nodePort: 30081

selector:

name: webapp-mysql

type: NodePort

Ex2:

apiVersion: v1

kind: Pod

metadata:

labels:

name: mysql

name: mysql

namespace: alpha

spec:

containers:

- env:

- name: MYSQL\_ROOT\_PASSWORD

value: paswrd

image: mysql:5.6

imagePullPolicy: IfNotPresent

name: mysql

ports:

- containerPort: 3306

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: alpha

spec:

ports:

- port: 3306

targetPort: 8080

selector:

name: mysql-db

---

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

namespace: alpha

spec:

replicas: 1

selector:

matchLabels:

name: webapp-mysql

template:

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

spec:

containers:

- env:

- name: DB\_Host

value: mysql-service

- name: DB\_User

value: root

- name: DB\_Password

value: paswrd

image: mmumshad/simple-webapp-mysql

imagePullPolicy: Always

name: webapp-mysql

ports:

- containerPort: 8080

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: web-service

namespace: alpha

spec:

ports:

- port: 8080

targetPort: 8080

nodePort: 30081

selector:

name: webapp-mysql

type: NodePort

Ex3:

apiVersion: v1

kind: Pod

metadata:

labels:

name: mysql

name: mysql

namespace: alpha

spec:

containers:

- env:

- name: MYSQL\_ROOT\_PASSWORD

value: password

image: mysql:5.6

imagePullPolicy: IfNotPresent

name: mysql

ports:

- containerPort: 3306

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: alpha

spec:

ports:

- port: 3306

targetPort: 3306

selector:

name: mysql

---

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

namespace: alpha

spec:

replicas: 1

selector:

matchLabels:

name: webapp-mysql

template:

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

spec:

containers:

- env:

- name: DB\_Host

value: mysql-service

- name: DB\_User

value: sql-user

- name: DB\_Password

value: paswrd

image: mmumshad/simple-webapp-mysql

imagePullPolicy: Always

name: webapp-mysql

ports:

- containerPort: 8080

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: web-service

namespace: alpha

spec:

ports:

- port: 8080

targetPort: 8080

nodePort: 30081

selector:

name: webapp-mysql

type: NodePort

Ex4:

apiVersion: v1

kind: Pod

metadata:

labels:

name: mysql

name: mysql

namespace: ex2

spec:

containers:

- env:

- name: MYSQL\_ROOT\_PASSWORD

value: paswrd

image: mysql:5.6

imagePullPolicy: IfNotPresent

name: mysql

ports:

- containerPort: 3306

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: mysql-service

namespace: ex2

spec:

ports:

- port: 3306

targetPort: 8080

selector:

name: mysql-db

---

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

namespace: ex2

spec:

replicas: 1

selector:

matchLabels:

name: webapp-mysql

template:

metadata:

labels:

name: webapp-mysql

name: webapp-mysql

spec:

containers:

- env:

- name: DB\_Host

value: mysql-svc

- name: DB\_User

value: root

- name: DB\_Password

value: paswrd

image: mmumshad/simple-webapp-mysql

imagePullPolicy: Always

name: webapp-mysql

ports:

- containerPort: 8080

protocol: TCP

---

apiVersion: v1

kind: Service

metadata:

name: web-service

namespace: ex2

spec:

ports:

- port: 8080

targetPort: 8088

nodePort: 30081

selector:

name: webapp-mysql

type: NodePort