# KUBERNETES APPLICATION DEPLOYMENT

Version 5.0

AUGUST 10, 2020

AVANSEUS TECHNOLOGIES PVT. LTD.

**REVISION HISTORY**

| Version | Date | Change description | Created by | Updated by | Reviewed by |
|---|---|---|---|---|---|
| V 5.0 | Aug, 2020 | - | Naveen | Sandeep | Chiranjib |

**Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.**

## TABLE OF CONTENTS

Copyright © 2020 Avanseus Holdings Pte. Ltd.  All rights reserved.

## 1. Focus

This document focuses on building a docker image as a kubernetes pod and exposing them as a service. The kubernetes configuration files for applications written in a YAML file are tested in both single and multi-node kubernetes cluster setup.

The docker images wrapped into a kubernetes pod are as follows:

1. MongoDB
2. LDAP image
3. VBI
4. CAN (Master)
5. CAN (Slave)

## 2. Pre-Requisites

A few prerequisites which need to be configured before deploying kubernetes applications are as follows:

1. Docker must be installed
2. Kubernetes must be installed
3. SWAP must be disabled
4. Enable/Allow IPV4 & IPV6 bridge network
5. Install flannel network for multi-node cluster communication
6. Make sure the necessary docker images are loaded on all the machines which are part of kubernetes multi-node cluster.
7. Open all traffic protocols between the master and worker node machines.

## 3. Steps to build pods

### 1. MongoDB

Pod configuration

MongoDB needs to be installed with an admin user. This is possible by writing a stateful set configuration. Create a file with name “statefulsetmongo.yaml” and add the following lines of commands.

```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: mongodb-standalone
spec:
  serviceName: database
  replicas: 1
  selector:
    matchLabels:
      app: database
  template:
    metadata:
      labels:
        app: database
        selector: mongodb-standalone
    spec:
      volumes:
      - name: mongodb-mount
        hostPath:
         path: <path_on_physical_system>
         type: DirectoryOrCreate
      containers:
      - name: mongodb-standalone
        image: mongo:3.4.6
```

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.

```
        env:
          - name: MONGO_INITDB_ROOT_USERNAME
            value: <admin_user>
          - name: MONGO_INITDB_ROOT_PASSWORD
            value: <admin_password>
        volumeMounts:
        - mountPath: /data/db
          name: mongodb-mount
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the stateful set, use the following command:

```
kubectl create -f statefulsetmongo.yaml
```

This will create a MongoDB pod.

Service configuration

To expose the application as a service on a physical port, a YAML file for service needs to be written. Please write the following set of commands in a file named “mongoservice.yaml”.

```
apiVersion: v1
kind: Service
metadata:
  name: database
  labels:
    app: database
spec:
  selector:
    app: database
  ports:
  - port: 27017
    protocol: TCP
    targetPort: 27017
    nodePort: 30001
  type: NodePort
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the MongoDB as a service, use the following command:

```
kubectl create -f statefulsetmongo.yaml
```

This will allow users to connect to MongoDB on Nodeport 30001 as mentioned in the service configuration from the master node.

Copyright © 2020 Avanseus Holdings Pte. Ltd.  All rights reserved.

## 2. OpenDS LDAP

Pod configuration

OpenDS pod configuration file “ldap_pod.yaml” is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: avanseus-ldap
  labels:
   app: ldap-container
spec:
 containers:
 - name: ldap-container
   image: ldapimage:1
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the LDAP pod, use the following command:

```
kubectl create -f ldap_pod.yaml
```

This will create a LDAP pod.

Service configuration

To expose the application as a service on a physical port, a YAML file for service needs to be written. Please write the following set of commands in a file named “ldap_service.yaml”.

```
apiVersion: v1
kind: Service
metadata:
  name: ldap-container
  labels:
    app: ldap-container
spec:
  selector:
    app: ldap-container
  ports:
  - port: 1389
    name: ldap-port
    protocol: TCP
    targetPort: 1389
    nodePort: 30002
  - port: 4444
    name: ldap-admin-port
    protocol: TCP
    targetPort: 4444
    nodePort: 30003
  type: NodePort
```

Please make sure the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the LDAP as a service, use the following command:

```
kubectl create -f ldap_service.yaml
```

This will allow users to connect to LDAP on Nodeport 30002 for user access & 30003 for admin operation as mentioned in the service configuration from master node.

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.

## 3. VBI Module

Pod configuration

VBI module configuration file “vbi_pod.yaml” is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: avanseus-vbi
  labels:
   app: vbi-container
spec:
 containers:
 - name: vbi-container
   image: pyvbi:v1
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the VBI pod, use the following command:

```
kubectl create -f vbi_pod.yaml
```

This will create a VBI pod.

Service configuration

To expose the application as a service on a physical port, a YAML file for service needs to be written. Please write the following set of commands in a file named “vbi_service.yaml”.

```
apiVersion: v1
kind: Service
metadata:
  name: vbi-container
  labels:
    app: vbi-container
spec:
  selector:
    app: vbi-container
  ports:
  - port: 12001
    protocol: TCP
    targetPort: 12001
    nodePort: 30004
  type: NodePort
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the LDAP as a service, use the following command:

```
kubectl create -f vbi_service.yaml
```

This will allow users to connect to the VBI module on Nodeport 30004.

**Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.**

## 4. CAN Master

Pod configuration

CAN master configuration file “can_pod.yaml” is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: avanseus-can-master
  labels:
   app: can-master-container
spec:
 volumes:
 - name: can-logs-dir
   hostPath:
    path: <path_to_master_logs_directory>
    type: DirectoryOrCreate
 - name: can-app-config
   hostPath:
    path: <path_to_config_properties_file>
    type: File
 - name: catalina-properties
   hostPath:
    path: <path_to_catalina_properties>
    type: File
 - name: java-security-env
   hostPath:
    path: <path_to_setenv_script>
    type: File
 containers:
 - name: can-master-container
   image: canapp:1
   volumeMounts:
   - mountPath: /data/workspace/logs/
     name: can-logs-dir
   - mountPath: /data/workspace/tomcatCAN/config.properties
     name: can-app-config
   - mountPath: /data/workspace/tomcatCAS/config.properties
     name: can-app-config
   - mountPath: /data/workspace/tomcatCAN/conf/catalina.properties
     name: catalina-properties
   - mountPath: /data/workspace/tomcatCAS/conf/catalina.properties
     name: catalina-properties
   - mountPath: /data/workspace/tomcatCAN/bin/setenv.sh
     name: java-security-env
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the VBI pod, use the following command:

```
kubectl create -f can_pod.yaml
```

This will create a CAN master pod.

Service configuration

To expose the application as a service on a physical port, a YAML file for service needs to be written. Please write the following set of commands in a file named “can_service.yaml”.

```
apiVersion: v1
kind: Service
metadata:
  name: can-master-container
  labels:
```

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.

```
    app: can-master-container
spec:
  selector:
    app: can-master-container
  ports:
  - port: 2000
    protocol: TCP
    targetPort: 2000
    name: can-port
    nodePort: 32000
  - port: 2002
    protocol: TCP
    targetPort: 2002
    name: can-ajp-port
    nodePort: 32002
  - port: 2003
    protocol: TCP
    targetPort: 2003
    name: cas-port
    nodePort: 32003
  - port: 2005
    protocol: TCP
    targetPort: 2005
    name: cas-ajp-port
    nodePort: 32005
  - port: 31900
    protocol: TCP
    targetPort: 31900
    name: cas-master-port
    nodePort: 31900
  type: NodePort
```

Make sure that the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the LDAP as a service, use the following command:

```
kubectl create -f can_service.yaml
```

The service exposes many ports following purpose:

32000 - CAN application port

32002 - CAN AJP port

32003 - CAS application port

32005 - CAS AJP port

31900 - CAN master port for prediction distribution

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.

## 5. CAN Slave

Pod configuration

CAN master configuration file “can_slave_pod_nodeX.yaml” is as follows:

```
apiVersion: v1
kind: Pod
metadata:
  name: avanseus-can-slave-node-x
  labels:
   app: can-slave-pod-node-x
spec:
 volumes:
 - name: can-logs-dir-node-x
   hostPath:
    path: <path_to_nodex_log_directory>
    type: DirectoryOrCreate
 - name: can-app-config-node-b
   hostPath:
    path: <path_to_nodex_config_properties>
    type: File
 - name: catalina-properties
   hostPath:
    path:<path_to_catalina_properties>
    type: File
 - name: java-security-env
   hostPath:
    path: <path_to_setenv_script>
    type: File
 containers:
 - name: can-slave-node-b
   image: canslaveapp:1
   volumeMounts:
   - mountPath: /data/workspace/logs/
     name: can-logs-dir-node-b
   - mountPath: /data/workspace/tomcatCAN/config.properties
     name: can-app-config-node-b
   - mountPath: /data/workspace/tomcatCAN/conf/catalina.properties
     name: catalina-properties
   - mountPath: /data/workspace/tomcatCAN/bin/setenv.sh
     name: java-security-env
```

Please make sure the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the VBI pod, use the following command:

```
kubectl create -f can_slave_pod_nodeX.yaml
```

This will create a CAN slave pod.

Service configuration

To expose the application as a service on a physical port, a YAML file for service needs to be written. Please write the following set of commands in a file named “can_slave_service_nodeX.yaml”.

```
apiVersion: v1
kind: Service
metadata:
  name: can-slave-service-node-x
  labels:
    app: can-slave-pod-node-x
spec:
```

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.

```
  selector:
    app: can-slave-pod-node-x
  ports:
  - port: 31901
    protocol: TCP
    targetPort: 31901
    name: can-slave-port-node-x
    nodePort: 31901
  type: NodePort
```

Please make sure the indentations are exactly the same as above. Replace the places enclosed in angular brackets with necessary information.

To run the LDAP as a service, use the following command:

```
kubectl create -f can_slave_service_nodeX.yaml
```

The service exposes 31901 for the master to communicate with it. If slave workers need to be replicated more than 1 node, then a new pod and service file needs to be written. Also, if it is noticed, there is a placeholder "x" which depicts the node name/number. This can be modified for replication. Also, in the service file, the port numbers need to be changed across different slave nodes.

Copyright © 2020 Avanseus Holdings Pte. Ltd. All rights reserved.