Stateful Applications

» Hadoop StatefulSet

Overview

Create a StatefulSet manifest file for an a Hadoop cluster (datanode and namenode).

```
cat >> hadoop-stateful-set.yaml <<EOF
# A headless service to create DNS records.
apiVersion: v1
kind: Service
metadata:
  name: hdfs-namenode
  labels:
    type: big-data-example
    app: hdfs-namenode
spec:
  ports:
  - port: 8020
    name: fs
  clusterIP: None
  selector:
    app: hdfs-namenode
---
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: hdfs-namenode
spec:
  serviceName: "hdfs-namenode"
  replicas: 1
  template:
    metadata:
      labels:
        type: big-data-example
        app: hdfs-namenode
    spec:
      terminationGracePeriodSeconds: 0
      containers:
        - name: hdfs-namenode
          image: uhopper/hadoop-namenode:2.7.2
          env:
            - name: CLUSTER_NAME
              value: hdfs-k8s
          ports:
          - containerPort: 8020
            name: fs
      restartPolicy: Always
---
# A headless service to create DNS records.
apiVersion: v1
```

```
apiVersion: v1
kind: Service
metadata:
  name: hdfs-datanode
  labels:
    type: big-data-example
    app: hdfs-datanode
spec:
  ports:
  - port: 50010
    name: fs
  clusterIP: None
  selector:
    app: hdfs-datanode
---
apiVersion: apps/v1beta1
kind: StatefulSet
metadata:
  name: hdfs-datanode
spec:
  serviceName: "hdfs-datanode"
  replicas: 1
  template:
    metadata:
      labels:
        type: big-data-example
        app: hdfs-datanode
    spec:
      containers:
        - name: datanode
          image: uhopper/hadoop-datanode:2.7.2
          env:
            - name: CORE_CONF_fs_defaultFS
              value: hdfs://hdfs-namenode-0.hdfs-namenode.default.svc.cluster.local:8020
          ports:
          - containerPort: 50010
            name: fs
      restartPolicy: Always
EOF
```

Create the StatefulSet.

```
kubectl create -f hadoop-stateful-set.yaml
```

Verify the StatefulSet was successfully created.

```
kubectl get pods -l type=big-data-example
```

Access the Hadoop dashboard by using the `port-forward` command.

```
kubectl port-forward hdfs-namenode-0 50070:50070
```

You will need to create an SSH tunnel to your local machine when using a bastion node. Remember to replace `labXX` with your actual lab number.

```
ssh -i labXX-key -L 50070:localhost:50070 lab@labXX-bastion.coreostrain.me
```

Navigate to the datanodes page. You should currently see one datanode.

```
http://localhost:50070/dfshealth.html#tab-datanode
```

Scale the number of datanode replicas to 3!

```
kubectl scale statefulset hdfs-datanode --replicas=3
```

Refresh your browser. You should now see additional datanodes. Take note of available capacity of the nodes. This reflects storage available storage on our worker nodes.

```
http://localhost:50070/dfshealth.html#tab-datanode
```

Use the Hadoop filesystem user client to create a new Hadoop directory and copy the ping binary to it.

```
kubectl exec -ti hdfs-namenode-0 /bin/sh
hadoop fs -mkdir /tmp
hadoop fs -put /bin/ping /tmp/
hadoop fs -ls /tmp
```

From the Hadoop GUI, view your file in the datanode cluster.

```
http://localhost:50070/explorer.html
```

Leave this StatefulSet running, as we will use it in the next exercise.