

Lesson 03 Demo 10

Configuring a DaemonSet

Objective: To configure a DaemonSet within Kubernetes for efficient application deployment and management

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. Create and configure a DaemonSet

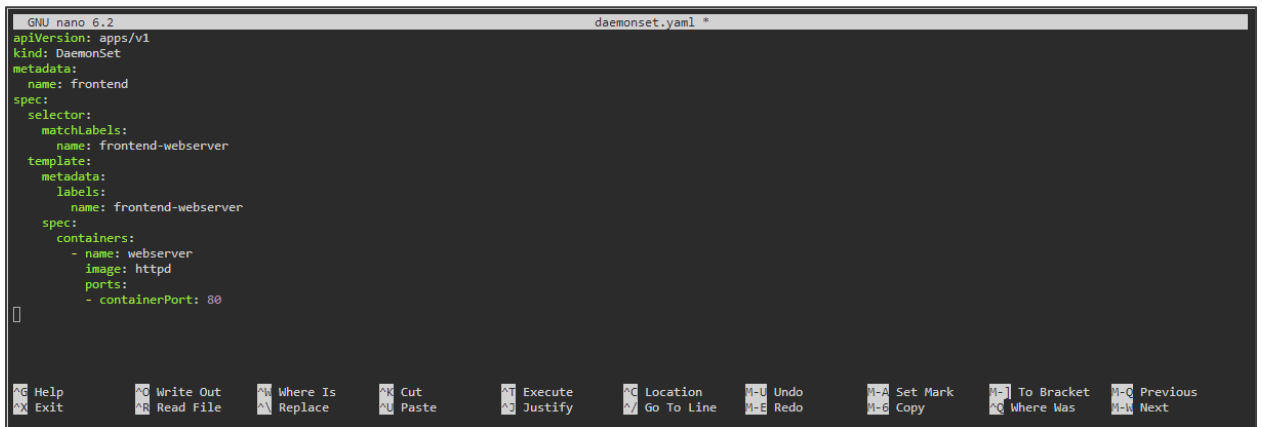
Step 1: Create and configure a DaemonSet

- 1.1 Create a YAML file using the following command:
nano daemonset.yaml

```
labsuser@master:~$ nano daemonset.yaml
```

1.2 Add the following code to the **daemonset.yaml** file:

```
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: frontend
spec:
  selector:
    matchLabels:
      name: frontend-webserver
  template:
    metadata:
      labels:
        name: frontend-webserver
    spec:
      containers:
        - name: webserver
          image: httpd
          ports:
            - containerPort: 80
```



```
GNU nano 6.2 daemonset.yaml *
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: frontend
spec:
  selector:
    matchLabels:
      name: frontend-webserver
  template:
    metadata:
      labels:
        name: frontend-webserver
    spec:
      containers:
        - name: webserver
          image: httpd
          ports:
            - containerPort: 80
[]

Help  Write Out  Where Is  Cut  Execute  Location  Undo  Set Mark  To Bracket  Previous
Exit  Read File  Replace  Paste  Justify  Go To Line  Redo  Copy  Where Was  Next
```

1.3 Use the **cat** command to validate the content of the **daemonset.yaml** file

```
labsuser@master:~$ cat daemonset.yaml
apiVersion: apps/v1
kind: DaemonSet
metadata:
  name: frontend
spec:
  selector:
    matchLabels:
      name: frontend-webserver
  template:
    metadata:
      labels:
        name: frontend-webserver
    spec:
      containers:
        - name: webserver
          image: httpd
          ports:
            - containerPort: 80
labsuser@master:~$
```

1.4 Create the DaemonSet resource using the following command:
kubectl create -f daemonset.yaml

```
spec:
  containers:
    - name: webserver
      image: httpd
      ports:
        - containerPort: 80
labsuser@master:~$ kubectl create -f daemonset.yaml
daemonset.apps/frontend created
labsuser@master:~$
```

1.5 Verify the DaemonSet state using the following command:
kubectl get ds

```
labsuser@master:~$ kubectl create -f daemonset.yaml
daemonset.apps/frontend created
labsuser@master:~$ kubectl get ds
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
frontend	2	2	0	2	0	<none>	3m35s

```
labsuser@master:~$
```

By following the above steps, you have successfully configured a DaemonSet in Kubernetes to ensure that the specified containerized application is deployed and running consistently across all nodes in the cluster, providing high availability and efficient workload distribution.