

## Lesson 04 Demo 10

### Configuring DaemonSet

**Objective:** To demonstrate the process of configuring the DaemonSets within Kubernetes for efficient application deployment and management

**Tools required:** kubeadm, kubectl, kubelet, and containerd

**Prerequisites:** A Kubernetes cluster should already be set up (refer to the steps provided in Lesson 02, Demo 01 for guidance).

Steps to be followed:

1. Create and verify the DaemonSet

#### Step 1: Create and verify the DaemonSet

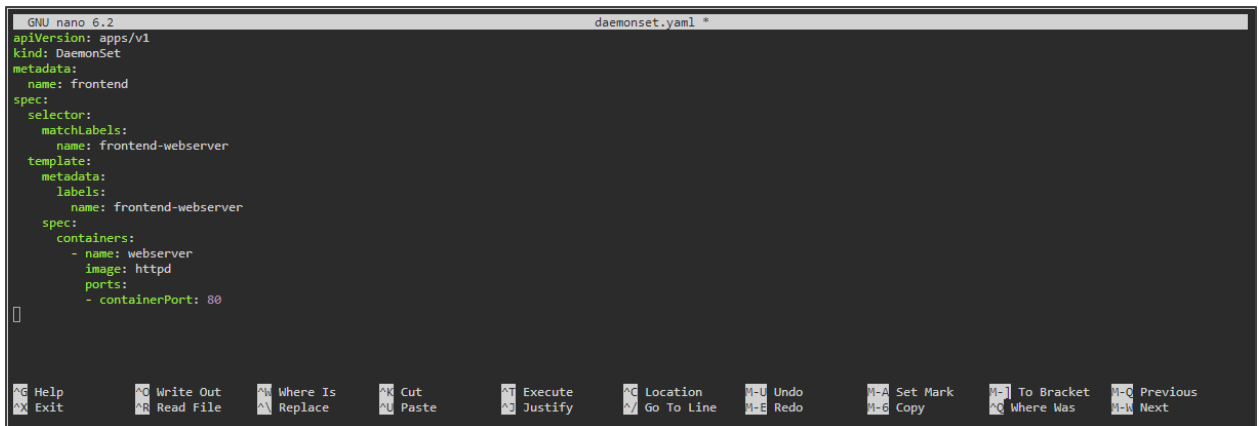
- 1.1 Create the YAML file by using the following command:

**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
```



The screenshot shows a terminal window with the GNU nano 6.2 editor open. The file being edited is daemonset.yaml. The content of the file is as follows:

```
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
```

The terminal window has a dark background with light green text. The nano editor's status bar at the bottom shows various keyboard shortcuts for editing and navigation.

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 **DemonSet** resource by 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 by 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 these steps, you have successfully configured a DaemonSet in Kubernetes to run a specified containerized application across all cluster nodes.