

# Lab: Working with Kubernetes DaemonSets

## Introduction

A Kubernetes **DaemonSet** is a container tool that ensures that all nodes (or a specific subset of them, but we'll get to that later) are running exactly **one copy of a pod**. DaemonSets will even create the pod on new nodes that are added to your cluster!

When using Kubernetes, most of the time you don't care where your pods are running, but sometimes you want to run a single pod on all your nodes

In this Lab, you will learn below items:

### Objective:

- **Create** DaemonSet
- **Verify** Daemonset
- **Cleanup**

**Note:** Ensure you have running cluster deployed

1. Ensure that you have logged-in as **root** user with password as **linux** on **kube-master** node.

1.1 Let us clone the git repository which contains manifests required for this exercise, by executing the below command.

```
# git clone https://github.com/EyesOnCloud/k8s-daemonset.git
```

### Output:

```
Cloning into 'k8s-daemonset'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
```

1.2 Let us view the manifest file.

```
# cat -n ~/k8s-daemonset/daemonset.yaml
```

Output:

```
1 apiVersion: apps/v1
2 kind: DaemonSet
3 metadata:
4   name: webserver
5 spec:
6   selector:
7     matchLabels:
8       name: webserver
9   template:
10    metadata:
11     labels:
12       name: webserver
13    spec:
14     tolerations:
15       # this toleration is to have the daemonset
16       # remove it if your masters can't run pods
17       - key: node-role.kubernetes.io/master
18         effect: NoSchedule
19     containers:
20       - name: nginx-webserver
21         image: nginx
```

1.3 Let us create the daemonset, by executing the below command.

```
# kubectl create -f ~/k8s-daemonset/daemonset.yaml
```

Output:

```
[root@kube-master ~]# kubectl create -f ~/k8s-daemonset/daemonset.yaml
daemonset.apps/webserver created
```

1.4 Let's see verify the DaemonSet is created and ready, by executing below command.

```
# kubectl get ds webserver
```

Output:

```
[root@kube-master ~]# kubectl get ds webserver
```

NAME	DESIRED	CURRENT	READY	UP-TO-DATE	AVAILABLE	NODE SELECTOR	AGE
webserver	3	3	3	3	3	<none>	45s

**1.5** Let's verify the pods are created on all the nodes, by executing the below command.

```
# kubectl get pods -l name=webserver -o wide
```

**Output:**

```
[root@kube-master ~]# kubectl get pods -l name=webserver -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED	NODE	READINESS	GATES
webserver-9w5hf	1/1	Running	0	89s	172.16.221.205	kube-master	<none>		<none>	
webserver-cr44g	1/1	Running	0	89s	172.16.9.82	kube-node1	<none>		<none>	
webserver-w9mq1	1/1	Running	0	89s	172.16.233.209	kube-node2	<none>		<none>	

**Note:** Daemonset by design schedules at least one pod running on each node of the cluster. If we add another node to the cluster, then a pod will be scheduled in the new node. Due to limitation of this demo cluster, we won't be able to show this functionality.

**1.6** Let's clean up by deleting the DaemonSet, by executing the below command.

```
# kubectl delete -f ~/k8s-daemonset
```

**Output:**

```
[root@kube-master ~]# kubectl delete -f ~/k8s-daemonset/daemonset.apps "fluentd-elasticsearch" deleted
```