## Exercise 7.2: Working with DaemonSets

A `DaemonSet` is a watch loop object like a `Deployment` which we have been working with in the rest of the labs. The `DaemonSet` ensures that when a node is added to a cluster a pods will be created on that node. A `Deployment` would only ensure a particular number of pods are created in general, several could be on a single node. Using a `DaemonSet` can be helpful to ensure applications are on each node, helpful for things like metrics and logging especially in large clusters where hardware may be swapped out often. Should a node be removed from a cluster the `DaemonSet` would ensure the Pods are garbage collected before removal. Starting with Kubernetes v1.12 the scheduler handles `DaemonSet` deployment which means we can now configure certain nodes to not have a particular `DaemonSet` pods.

This extra step of automation can be useful for using with products like **ceph** where storage is often added or removed, but perhaps among a subset of hardware. They allow for complex deployments when used with declared resources like memory, CPU or volumes.

1. We begin by creating a yaml file. In this case the `kind` would be set to `DaemonSet`. For ease of use we will copy the previously created `rs.yaml` file and make a couple edits. Remove the `Replicas:    2` line.

   student@cp:~$ cp rs.yaml ds.yaml

   student@cp:~$ vim ds.yaml

   **ds.yaml**
   ```
   1  ....
   2  kind: DaemonSet
   3  ....
   4    name: ds-one
   5  ....
   6    replicas: 2 #<<<----Remove this line
   7  ....
   8         system: DaemonSetOne #<<-- Edit both references
   9  ....
   ```

2. Create and verify the newly formed DaemonSet. There should be one Pod per node in the cluster.

   student@cp:~$ kubectl create -f ds.yaml
   ```
   daemonset.apps/ds-one created
   ```

   student@cp:~$ kubectl get ds
   ```
   NAME       DESIRED    CURRENT    READY      UP-TO-DATE    AVAILABLE    NODE-SELECTOR    AGE
   ds-one     2          2          2          2             2            <none>           1m
   ```

   student@cp:~$ kubectl get pod
   ```
   NAME                READY      STATUS      RESTARTS    AGE
   ds-one-b1dcv        1/1        Running     0           2m
   ds-one-z31r4        1/1        Running     0           2m
   ```

3. Verify the image running inside the Pods. We will use this information in the next section.

   student@cp:~$ kubectl describe pod ds-one-b1dcv | grep Image:
   ```
   Image:              nginx:1.15.1
   ```