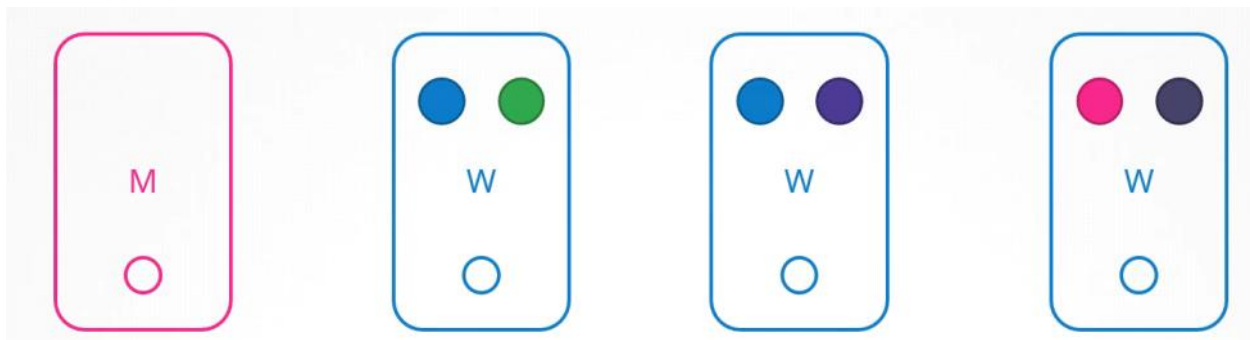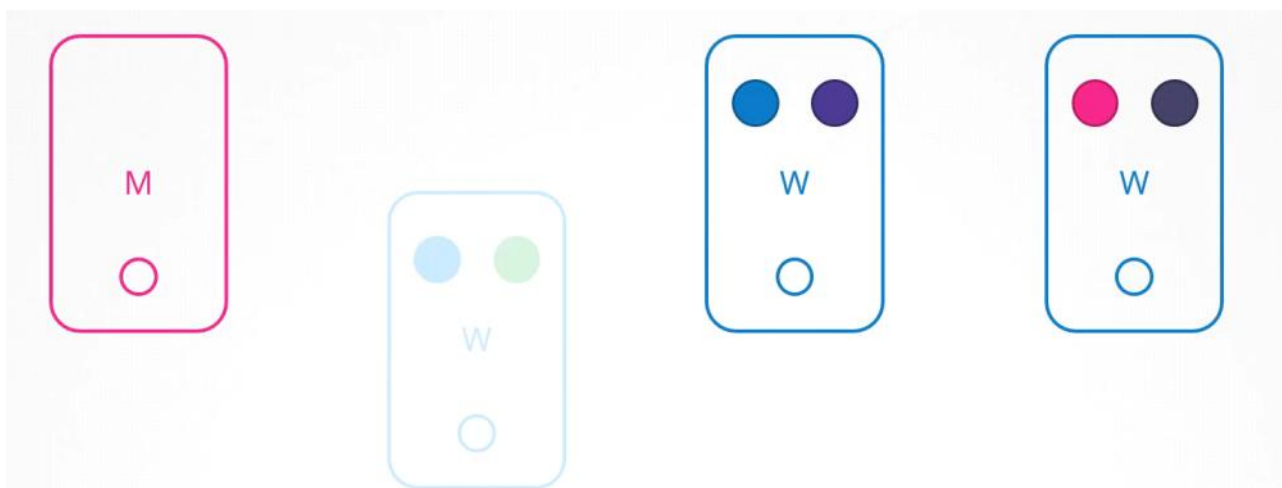So you have a cluster with a few notes and pods serving applications.
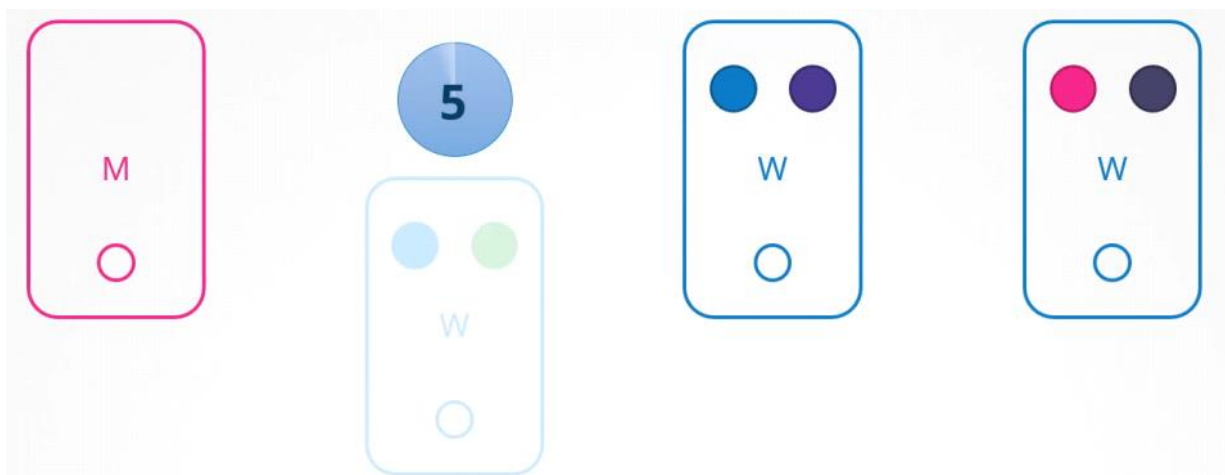
M   W   W   W

What happens when one of these nodes go down.

M   W   W   W

USERS CAN ACCESS BLUE POD IF THERE IS A REPLICA SET but THEY CAN NOT ACCESS

GREEN POD

If the node came back online immediately, then the kubectl process starts and the pods come back
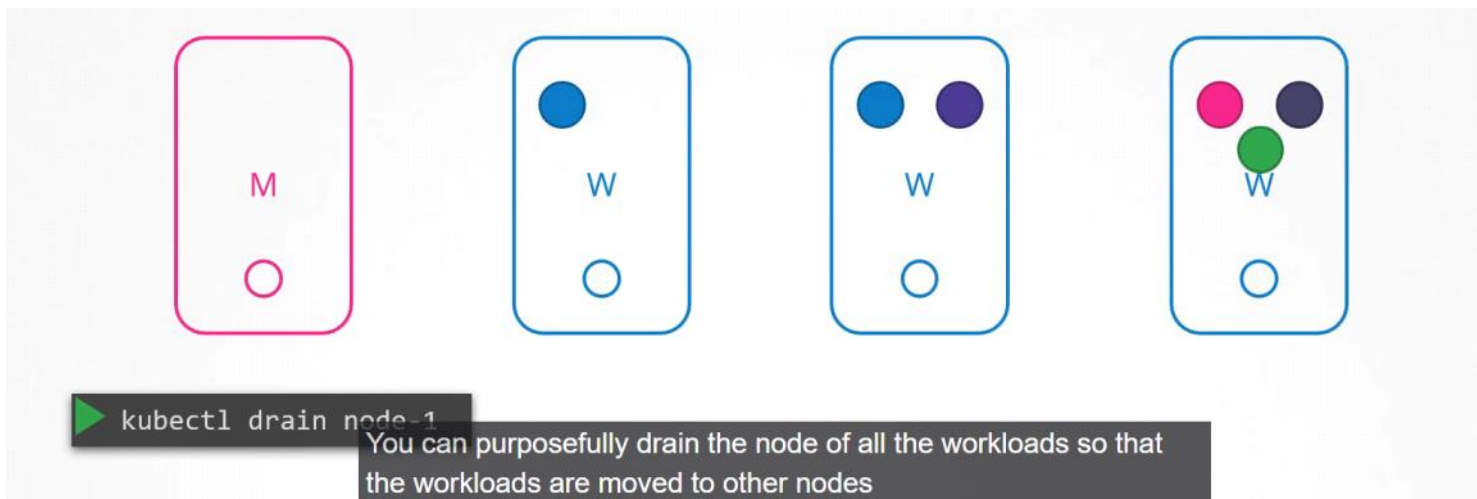
M   5 W   W   W

However, if the node was down for more than 5 minutes, then the pods are terminated from that node.

Well, kubernetes considers them as dead.

5

W

W

W

```
kube-controller-manager --pod-eviction-timeout=5m0s   ...
```

M

W

W

W
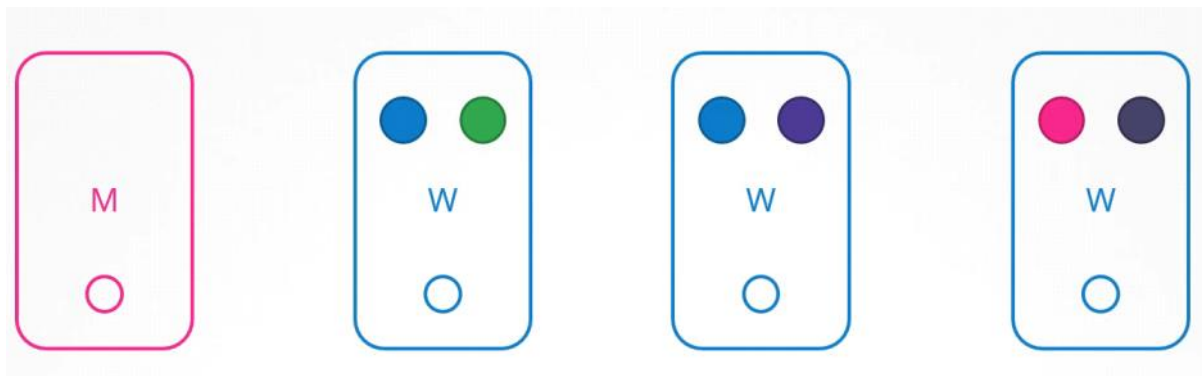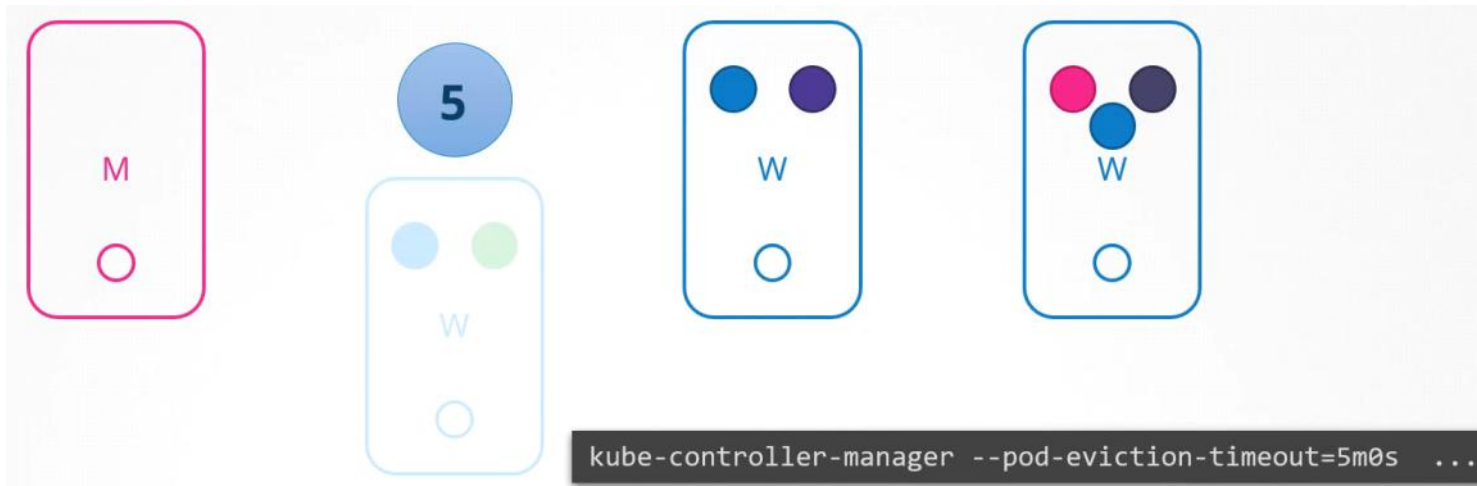
When the node comes back on line after the pod eviction timeout it comes up blank without any pods scheduled

M

W

W

W

```
▶ kubectl drain node-1
```

You can purposefully drain the node of all the workloads so that the workloads are moved to other nodes

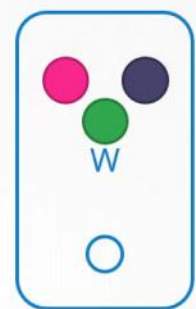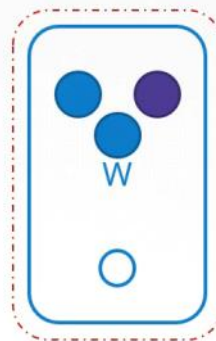The node is also cordoned or marked as unschedulable. Meaning no pods can be scheduled on this node until

you specifically remove the restriction.

You then need to uncordon it, so that pods can be scheduled on it again.
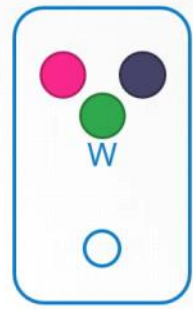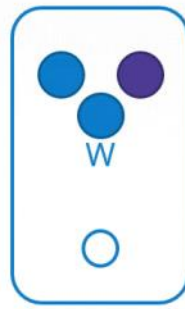
```
kubectl drain node-1
```

```
kubectl uncordon node-1
```



```
kubectl drain node-1
```

```
kubectl cordon node-2
```

```
kubectl uncordon node-1
```

Cordon means a node schedulable. Unlike drain it does not terminate or move the pods on an existing node.