KUBERNETES NODESELECTOR & AFFINITY

The NodeSelector field in a deployment YAML is used to schedule pods onto nodes whose labels match the specified selector.

Lets Explore Node Selector Practically.

Firstly we have to apply labels to the following nodes:

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get nodes
NAME                          STATUS   ROLES           AGE    VERSION
rayeez-cluster-control-plane   Ready    control-plane   3h3m   v1.31.4
rayeez-cluster-worker          Ready    <none>          3h3m   v1.31.4
rayeez-cluster-worker2         Ready    <none>          3h3m   v1.31.4
```

Lets apply labels to the nodes:

```
kubectl label nodes rayeez-cluster-worker storage=ssd
kubectl label nodes rayeez-cluster-worker2 storage=hdd
```

Verify;

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker | head -10
Name:               rayeez-cluster-worker
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    env=prod
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker
                    kubernetes.io/os=linux
                    storage=ssd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker2 | head -10
Name:               rayeez-cluster-worker2
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker2
                    kubernetes.io/os=linux
                    storage=hdd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
```

Lets write a deployment.yaml using nodeSelector:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ns-deploy
spec:
  replicas: 3
  selector:
    matchLabels:
```

```
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      nodeSelector:
        storage: ssd
      containers:
        - name: nginx-container
          image: nginx
```

Apply;

```
kubectl apply -f ns.yaml
```

Verify on which nodes these pods are running;

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                         READY   STATUS    RESTARTS   AGE   IP            NODE                    NOMINATED NODE   READINESS GATES
ns-deploy-7bc788d6c9-2vjbx   1/1     Running   0          55s   10.244.1.53   rayeez-cluster-worker   <none>           <none>
ns-deploy-7bc788d6c9-drxj5   1/1     Running   0          55s   10.244.1.51   rayeez-cluster-worker   <none>           <none>
ns-deploy-7bc788d6c9-wn4sm   1/1     Running   0          55s   10.244.1.52   rayeez-cluster-worker   <none>           <none>
```

Pods are running on Worker node-1

Lets remove this label from worker node-1;

```
kubectl label nodes rayeez-cluster-worker storage-
```

Verify;

```
kubectl get nodes rayeez-cluster-worker --show-labels
```

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get nodes rayeez-cluster-worker --show-labels
NAME                  STATUS   ROLES    AGE    VERSION   LABELS
rayeez-cluster-worker   Ready    <none>   3h20m   v1.31.4   beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,env=prod,kubernetes.io/arch=amd64,kube
rnetes.io/hostname=rayeez-cluster-worker,kubernetes.io/os=linux
```

→ Even if the label is removed from the node-1, existing pods will continue to run on it.

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                         READY   STATUS    RESTARTS   AGE   IP            NODE                    NOMINATED NODE   READINESS GATES
ns-deploy-7bc788d6c9-2vjbx   1/1     Running   0          17m   10.244.1.53   rayeez-cluster-worker   <none>           <none>
ns-deploy-7bc788d6c9-drxj5   1/1     Running   0          17m   10.244.1.51   rayeez-cluster-worker   <none>           <none>
ns-deploy-7bc788d6c9-wn4sm   1/1     Running   0          17m   10.244.1.52   rayeez-cluster-worker   <none>           <none>
```

Lets add one more key under nodeSelector field of deployment yaml:

```
spec:
      nodeSelector:
        storage: ssd
        env: prod
```

Remove labels from both nodes and verify;

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker | head -10
Name:               rayeez-cluster-worker
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    env=prod
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker2 | head -10
Name:               rayeez-cluster-worker2
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker2
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
                    volumes.kubernetes.io/controller-managed-attach-detach: true
```

Both Nodes are unlabelled.

Now re-apply the deployment yaml and observe the Pods status:

```
kubectl apply -f ns.yaml
```

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                          READY   STATUS    RESTARTS   AGE   IP       NODE     NOMINATED NODE   READINESS GATES
ns-deploy-ff97d6dfc-l5lh8     0/1     Pending   0          2s    <none>   <none>   <none>           <none>
ns-deploy-ff97d6dfc-mn7tf     0/1     Pending   0          2s    <none>   <none>   <none>           <none>
ns-deploy-ff97d6dfc-pnxs9     0/1     Pending   0          2s    <none>   <none>   <none>           <none>
```

Since none of the nodes have labels that match the nodeSelector requirements, the pods remain unscheduled and stay in a Pending state.

Let’s apply the label storage=ssd to node-1 and env=prod to node-2, and then check the pod status.

```
kubectl label nodes rayeez-cluster-worker
storage=ssd
kubectl label nodes rayeez-cluster-worker2
env=prod
```

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker | head -10
Name:               rayeez-cluster-worker
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker
                    kubernetes.io/os=linux
                    storage=ssd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker2 | head -10
Name:               rayeez-cluster-worker2
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    env=prod
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker2
                    kubernetes.io/os=linux
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
```

Now check the pod status:

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                          READY   STATUS    RESTARTS   AGE   IP       NODE     NOMINATED NODE   READINESS GATES
ns-deploy-ff97d6dfc-hqk9t     0/1     Pending   0          1s    <none>   <none>   <none>           <none>
ns-deploy-ff97d6dfc-npqzv     0/1     Pending   0          1s    <none>   <none>   <none>           <none>
ns-deploy-ff97d6dfc-sfhf2     0/1     Pending   0          1s    <none>   <none>   <none>           <none>
```

→ The pods are still in a Pending state because no node has an exact match for both nodeSelector labels (storage=ssd and env=prod). Since no node meets all requirements, the pods cannot be scheduled.

Node-1 has only the storage=ssd label, while node-2 has only the env=prod label.

Let's add the env=prod label to node-1 so it fully matches the nodeSelector requirements in the YAML.

```
kubectl label nodes rayeez-cluster-worker env=prod
```

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                          READY   STATUS    RESTARTS   AGE     IP            NODE                  NOMINATED NODE   READINESS GATES
ns-deploy-ff97d6dfc-hqk9t     1/1     Running   0          5m11s   10.244.1.57   rayeez-cluster-worker   <none>           <none>
ns-deploy-ff97d6dfc-npqzv     1/1     Running   0          5m11s   10.244.1.59   rayeez-cluster-worker   <none>           <none>
ns-deploy-ff97d6dfc-sfhf2     1/1     Running   0          5m11s   10.244.1.58   rayeez-cluster-worker   <none>           <none>
```

Now pods are scheduled on Worker node-1 because of exact match with nodeSelector specification.

→ Limitations of Node Selector:

1. **Strict Placement:** If no node matches the label, The pod remains in the pending state.
2. **No Preference:** It does not allow "Soft" preferences-either a node matches or it does not. This is similar to set equality based selectors.

3. **No OR condition:** You cannot specify schedule on nodes with storage=ssd OR storage=hdd

Lets Explore **Node Affinity Rules:**

Node Affinity is the Advance version of Node Selector.

Following is the deployment yaml with affinity rules:

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: na-deploy
spec:
  replicas: 10
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuring
Execution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: storage
                    operator: In
                    values:
                      - ssd
                      - hdd
      containers:
        - name: myapp
          image: nginx
```

requiredDuringScheduling: It means schedule pods only when nodes labels are matched.

IgnoredDuringExecution: This means existing pods will continue running on the node even after the node label is removed.

```
nodeSelectorTerms:
                - matchExpressions:
                    - key: storage
                      operator: In
                      values:
                        - ssd
                        - hdd
```

Here, the key storage can have either ssd or hdd as its value. Since both values belong to the same key, it works as an OR condition.

Let's apply the label storage=ssd to node-1 and storage=hdd to node-2.

```
kubectl label nodes rayeez-cluster-worker storage=ssd
kubectl label nodes rayeez-cluster-worker2 storage=hdd
```

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker | head -10
Name:               rayeez-cluster-worker
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker
                    kubernetes.io/os=linux
                    storage=ssd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker2 | head -10
Name:               rayeez-cluster-worker2
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker2
                    kubernetes.io/os=linux
                    storage=hdd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
                    node.alpha.kubernetes.io/ttl: 0
```

Lets deploy 10 pods and observe the pods distribution across nodes:

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                         READY   STATUS    RESTARTS   AGE   IP            NODE                     NOMINATED NODE   READINESS GATES
na-deploy-dd99cdcb8-54h85    1/1     Running   0          34s   10.244.2.24   rayeez-cluster-worker2   <none>           <none>
na-deploy-dd99cdcb8-8dtj2    1/1     Running   0          34s   10.244.1.64   rayeez-cluster-worker    <none>           <none>
na-deploy-dd99cdcb8-f9kx4    1/1     Running   0          34s   10.244.2.20   rayeez-cluster-worker2   <none>           <none>
na-deploy-dd99cdcb8-fvmng    1/1     Running   0          34s   10.244.1.63   rayeez-cluster-worker    <none>           <none>
na-deploy-dd99cdcb8-htftn    1/1     Running   0          34s   10.244.2.23   rayeez-cluster-worker2   <none>           <none>
na-deploy-dd99cdcb8-jt6fb    1/1     Running   0          34s   10.244.1.62   rayeez-cluster-worker    <none>           <none>
na-deploy-dd99cdcb8-kdjsb    1/1     Running   0          34s   10.244.2.22   rayeez-cluster-worker2   <none>           <none>
na-deploy-dd99cdcb8-m84xq    1/1     Running   0          34s   10.244.2.19   rayeez-cluster-worker2   <none>           <none>
na-deploy-dd99cdcb8-pwkc6    1/1     Running   0          34s   10.244.1.61   rayeez-cluster-worker    <none>           <none>
na-deploy-dd99cdcb8-rrhkm    1/1     Running   0          34s   10.244.1.60   rayeez-cluster-worker    <none>           <none>
```

Pods are uniformly distributed across both nodes.

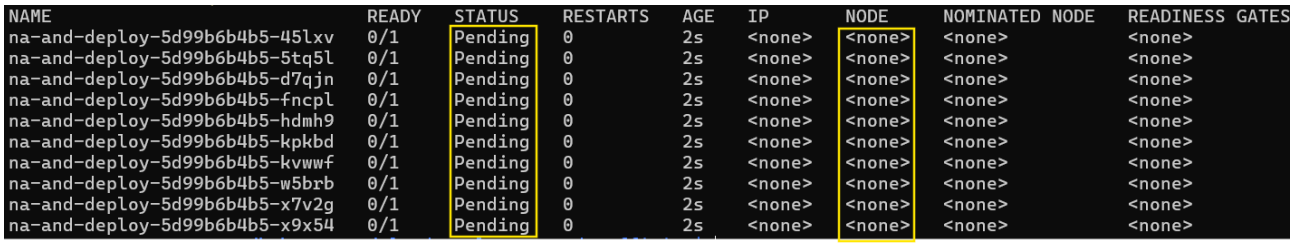
Let’s delete this deployment and apply a new one that uses affinity rules with an **AND** condition.

```
spec:
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuring
Execution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: storage
                    operator: In
                    values:
                      - ssd
                      - hdd
                  - key: env
                    operator: In
                    values:
                      - prod
```

→ This means the pods will be scheduled on nodes that have env=prod **AND** either storage=ssd **OR** storage=hdd.

Lets apply it :

```
kubectl apply -f na-and.yaml
```



NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
na-and-deploy-5d99b6b4b5-45lxv	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-5tq5l	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-d7qjn	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-fncpl	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-hdmh9	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-kpkbd	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-kvwwf	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-w5brb	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-x7v2g	0/1	Pending	0	2s	<none>	<none>	<none>	<none>
na-and-deploy-5d99b6b4b5-x9x54	0/1	Pending	0	2s	<none>	<none>	<none>	<none>

This is because none of the nodes have label **env=prod**.

Lets apply this label to worker node-2 and observe the pods

```
kubectl label nodes rayeez-cluster-worker2
env=prod
kubectl get pods -o wide
```

```
NAME                                READY   STATUS    RESTARTS   AGE     IP            NODE                      NOMINATED NODE   READINESS GATES
na-and-deploy-5d99b6b4b5-45lxv      1/1     Running   0          3m59s   10.244.2.34   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-5tq5l      1/1     Running   0          3m59s   10.244.2.32   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-d7qjn      1/1     Running   0          3m59s   10.244.2.26   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-fncpl      1/1     Running   0          3m59s   10.244.2.30   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-hdmh9      1/1     Running   0          3m59s   10.244.2.27   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-kpkbd      1/1     Running   0          3m59s   10.244.2.28   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-kvwwf      1/1     Running   0          3m59s   10.244.2.33   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-w5brb      1/1     Running   0          3m59s   10.244.2.31   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-x7v2g      1/1     Running   0          3m59s   10.244.2.25   rayeez-cluster-worker2    <none>           <none>
na-and-deploy-5d99b6b4b5-x9x54      1/1     Running   0          3m59s   10.244.2.29   rayeez-cluster-worker2    <none>           <none>
```

All the pods are scheduled on worker node-2 because labels are matched with affinity rules.

```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl describe nodes rayeez-cluster-worker2 | head -10
Name:               rayeez-cluster-worker2
Roles:              <none>
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    env=prod
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=rayeez-cluster-worker2
                    kubernetes.io/os=linux
                    storage=hdd
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: unix:///run/containerd/containerd.sock
```

Lets explore one more affinity rule with foolowing condition:

spec:

```
      affinity:
        nodeAffinity:
          requiredDuringSchedulingIgnoredDuring
Execution:
            nodeSelectorTerms:
              - matchExpressions:
                  - key: storage
                    operator: In
                    values:
                      - ssd
                      - hdd
          preferredDuringSchedulingIgnoredDurin
gExecution:
            - weight: 10
              preference:
                matchExpressions:
                  - key: storage
                    operator: In
                    values:
                      - ssd
            - weight: 5
              preference:
                matchExpressions:
```

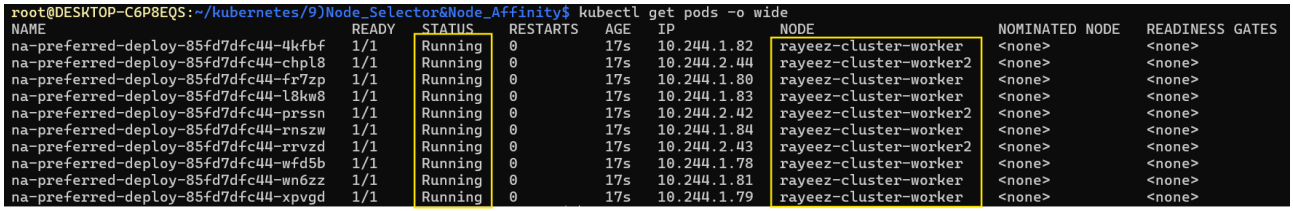
```
                - key: storage
                  operator: In
                  values:
                    - hdd
```

→ This indicates that nodes labeled storage=ssd are preferred over those labeled storage=hdd for scheduling.

Weight depends on lot of other factors such as taints & tolerations, resource availability etc.

Lets clear all resources and apply this deployment yaml;

```
kubectl apply -f na-prefered.yaml
```



```
root@DESKTOP-C6P8EQS:~/kubernetes/9)Node_Selector&Node_Affinity$ kubectl get pods -o wide
NAME                                        READY   STATUS    RESTARTS   AGE   IP            NODE                       NOMINATED NODE   READINESS GATES
na-preferred-deploy-85fd7dfc44-4kfbf        1/1     Running   0          17s   10.244.1.82   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-chpl8        1/1     Running   0          17s   10.244.2.44   rayeez-cluster-worker2     <none>           <none>
na-preferred-deploy-85fd7dfc44-fr7zp        1/1     Running   0          17s   10.244.1.80   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-l8kw8        1/1     Running   0          17s   10.244.1.83   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-prssn        1/1     Running   0          17s   10.244.2.42   rayeez-cluster-worker2     <none>           <none>
na-preferred-deploy-85fd7dfc44-rnszw        1/1     Running   0          17s   10.244.1.84   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-rrvzd        1/1     Running   0          17s   10.244.2.43   rayeez-cluster-worker2     <none>           <none>
na-preferred-deploy-85fd7dfc44-wfd5b        1/1     Running   0          17s   10.244.1.78   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-wn6zz        1/1     Running   0          17s   10.244.1.81   rayeez-cluster-worker      <none>           <none>
na-preferred-deploy-85fd7dfc44-xpvgd        1/1     Running   0          17s   10.244.1.79   rayeez-cluster-worker      <none>           <none>
```

Most number of pods are scheduled on worker node-1 when compared to worker node-2.