**Lesson 4 Demo 5**

**nodeName and nodeSelector Affinity**

**Objective:** To assign Pods to nodes using nodeName and nodeSelector fields

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

**Prerequisites:** A Kubernetes cluster must be set up (follow steps of Lesson 2 Demo 1)

Steps to be followed:

1. Creating a Pod with the field nodeName
2. Creating a Pod with the field nodeSelector
3. Assigning labels to nodes
4. Creating a Pod with the NotIn operator

**Step 1: Creating a Pod with the field nodeName**

1. Write the following code in the **nodename.yaml** file.

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: nginx**

**labels:**

**env: test**

**spec:**

**containers:**

**- name: nginx**

**image: httpd**

**imagePullPolicy: IfNotPresent**

**nodeName: worker-node2.example.com**

Text

Description automatically generated

1. Run the following command to create Pod:

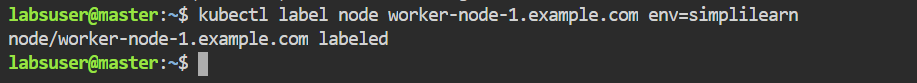
**kubectl create -f nodename.yaml**

A screenshot of a computer

Description automatically generated

1. Run the following command to label the node:

**kubectl label node worker-node-1.example.com env=simplilearn**

****

1. Verify the label.

**kubectl get nodes --show-labels**

**A picture containing text

Description automatically generated**

**Step 2: Creating a Pod with the field nodeSelector**

1. Write the following code in the **nodeselector.yaml** file.

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: nginx-labels**

**labels:**

**env: test**

**spec:**

**containers:**

**- name: nginx**

**image: nginx**

**imagePullPolicy: IfNotPresent**

**nodeSelector:**

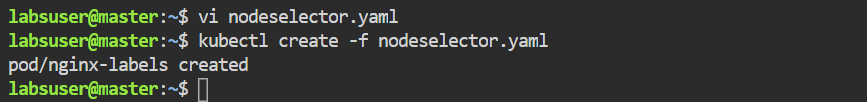
**env: simplilearn**

Text

Description automatically generated

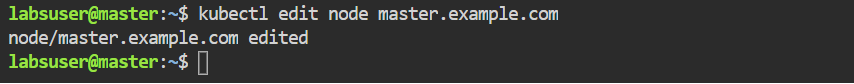
1. Run the following command to create Pod:

**kubectl create -f nodeselector.yaml**

****

1. Remove the **taints** field from **master.example.com**.

**kubectl edit node master.example.com**

****

**Text

Description automatically generated**

**Step 3: Assigning labels to nodes**

1. Assign labels to **worker 1** and **worker 2** nodes for Pod assignment using the following commands:

**kubectl label node worker-node-1.example.com color=blue**

**kubectl label node worker-node-2.example.com color=red**

**kubectl get nodes --show-labels**

**Graphical user interface, text

Description automatically generated**

**Step 4: Creating a Pod with the NotIn operator**

1. Write the following code in the **notin.yaml** file.

**apiVersion: v1**

**kind: Pod**

**metadata:**

**name: with-node-affinity**

**spec:**

**affinity:**

**nodeAffinity:**

**preferredDuringSchedulingIgnoredDuringExecution:**

**- weight: 1**

**preference:**

**matchExpressions:**

**- key: color**

**operator: NotIn**

**values:**

**- blue**

**containers:**

**- name: httpd**

**image: docker.io/httpd**Text

Description automatically generated

1. Run the following command to create Pod:

**kubectl create -f notin.yaml**

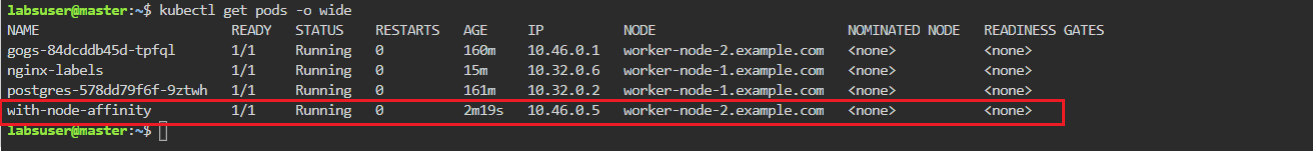
**Text

Description automatically generated**

|  |
| --- |
| Note: Pod will not execute on nodes with the **color=blue** label. |

1. Use the following command to verify which node the Pod is executing on:

**kubectl get pods -o wide**

****

The **node affinity** Pod is running on **worker node 2** since it has label **color=red** as seen in the above screenshot.