**Lab Exercise 19- Using NodeAffinity in Kubernetes**

This lab exercise is to demonstrate both required and preferred NodeAffinity using standalone pods instead of deployments.

**Step 1: Label the Nodes**

Identify the nodes in your cluster:

kubectl get nodes

Label a couple of nodes:

Choose two nodes and label them. Replace <node1> and <node2> with the names of your chosen nodes and label1, label2, and their respective values as you see fit:

kubectl label nodes <node1> type=high-performance

kubectl label nodes <node2> type=low-performance

Example:

kubectl label nodes node1 type=high-performance

kubectl label nodes node2 type=low-performance

Verify the node labels:

kubectl get nodes --show-labels

**Step 2: Create a Pod with Required NodeAffinity**

Create a YAML file for the pod with required NodeAffinity:

Create a file named **nginx-pod-required.yaml** with the following content:

apiVersion: v1

kind: Pod



metadata:



name: nginx-pod-required

spec:

affinity:

nodeAffinity:



requiredDuringSchedulingIgnoredDuringExecution:



nodeSelectorTerms:



- matchExpressions:



- key: type



operator: In



values:

- high-performance



containers:

- name: nginx

image: nginx:1.21

ports:

- containerPort: 80

Apply the pod configuration:

kubectl apply -f nginx-pod-required.yaml

Verify that the pod is scheduled on the node with the required label:

kubectl get pod nginx-pod-required -o wide

Check the NODE column to ensure that the pod is scheduled on the node with the label type=high-performance.

**Step 3: Create a Pod with Preferred NodeAffinity**

Create a YAML file for the pod with preferred NodeAffinity:

Create a file named **nginx-pod-preferred.yaml** with the following content:

apiVersion: v1

kind: Pod



metadata:

name: nginx-pod-preferred

spec:

affinity:

nodeAffinity:

preferredDuringSchedulingIgnoredDuringExecution:



- weight: 1

preference:

matchExpressions:

- key: type

operator: In



values:

- avg-performance



containers:

- name: nginx



image: nginx:1.21



ports:

- containerPort: 80

Apply the pod configuration:

kubectl apply -f nginx-pod-preferred.yaml

Verify that the pod is preferably scheduled on the node with the preferred label:

kubectl get pod nginx-pod-preferred -o wide

Check the NODE column to ensure that the pod is preferably scheduled on the node with the label type=high-performance, but may be scheduled on other nodes if necessary.

**Step 4: Clean Up**

Delete the pods:

kubectl delete -f nginx-pod-required.yaml

kubectl delete -f nginx-pod-preferred.yaml

Remove the labels from the nodes (optional):

kubectl label nodes <node1> type-

kubectl label nodes <node2> type-

Example:

kubectl label nodes node1 type-

kubectl label nodes node2 type-