### **Lab Exercise 13- Taints and Tolerations in Kubernetes Cluster**

Taints and tolerations in Kubernetes are used to ensure that Pods are only scheduled onto appropriate nodes. Taints are applied to nodes and indicate that only Pods with a matching toleration can be scheduled on those nodes. This mechanism helps control where Pods can run, offering a way to set aside specific nodes for certain types of workloads.

**Step 1: Set Up Kubernetes Cluster**

Ensure you have access to a Kubernetes cluster. You can use a local setup with Minikube, kind, or use a cloud-based Kubernetes service.

**Step 2: Apply Taint to a Node**

List all nodes in your cluster to identify which node you want to taint:

kubectl get nodes

Apply a taint to a node. Replace <node-name> with the name of your chosen node:

kubectl taint nodes <node-name> key=value:NoSchedule

This command taints the node with a key-value pair and a taint effect of NoSchedule, meaning no Pods will be scheduled on this node unless they have a matching toleration.

**Step 3: Create a Pod Without a Toleration**

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

apiVersion: v1

kind: Pod

metadata:

name: nginx-pod

spec:

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

Apply the manifest to create the Pod:

kubectl apply -f nginx-pod.yaml

Check the status of the Pod:

kubectl get pods

The Pod should be in a Pending state because it cannot be scheduled on the tainted node.

**Describe the Pod to see why it is pending:**

kubectl describe pod nginx-pod

You should see a message indicating that no nodes are available to schedule the Pod due to the taint.

**Step 4: Create a Pod with a Matching Toleration**

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

apiVersion: v1

kind: Pod

metadata:

name: toleration-pod

spec:

tolerations:

- key: "key"

operator: "Equal"

value: "value"

effect: "NoSchedule"

containers:

- name: nginx

image: nginx

ports:

- containerPort: 80

Apply the manifest to create the Pod:

kubectl apply -f toleration-pod.yaml

**Check the status of the Pod:**

kubectl get pods

The Pod should be in a Running state as it has a toleration matching the node's taint.

**Step 5: Verify the Pod's Node**

Check which node the Pod is running on:

kubectl get pod toleration-pod -o wide

You should see that the Pod is running on the tainted node.

**Step 6: Clean Up**

After completing the exercise, clean up the resources created and remove the taint from the node:

kubectl delete pod nginx-pod

kubectl delete pod toleration-pod

kubectl taint nodes <node-name> key=value:NoSchedule-