Lab 1 Pod Affinity

Lab Description:
In this lab, you will gain hands-on experience with advanced Kubernetes features, including pod affinity, pod anti-affinity, node selectors, taints, and tolerations. You will learn how to schedule and manage pods based on specific requirements and constraints within a Kubernetes cluster.

Step 1: Create Node Labels and Taints

Add labels to specific nodes in the cluster to define their characteristics or roles using the kubectl label nodes command.
Apply taints to certain nodes to mark them with specific restrictions or preferences using the kubectl taint nodes command.

Step 2: Define Pod Affinity and Anti-Affinity

Create a sample deployment manifest that specifies pod affinity rules, ensuring that related pods are scheduled close to each other.
Create another deployment manifest that includes pod anti-affinity rules, ensuring that related pods are scheduled on different nodes.

Step 3: Use Node Selectors

Create a deployment manifest that includes a node selector to schedule pods on nodes with specific labels.
Apply the deployment to the cluster and observe the pod scheduling based on the node selector.

Step 4: Apply Tolerations

Create a pod manifest that includes tolerations to allow pods to tolerate specific taints on nodes.
Apply the pod to the cluster and verify that it can be scheduled on nodes with the corresponding taints.

Step 5: Test and Validate

Test the functionality and behavior of the pods, verifying that they are scheduled according to the defined rules.
Use commands like kubectl get pods, kubectl describe pods, and kubectl describe nodes to inspect the pod scheduling, affinity, anti-affinity, node selectors, taints, and tolerations.

Lab Conclusion:

In this lab, you explored advanced Kubernetes features such as pod affinity, pod anti-affinity, node selectors, taints, and tolerations. You learned how to define and use these features to influence pod scheduling within a Kubernetes cluster. By understanding and leveraging these advanced features, you can optimize the deployment and management of your applications within Kubernetes environments.