**Create a pod**

kubectl run myfirstpod --image=httpd --port 80

kubectl get pods -o wide

**Create a pod Declaratively:**

pod.yaml

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: mypod  **labels**:  **env**: prod  **spec**:  **containers**:  **- image**: nginx  **name**: mynginxcontainer  **ports**:  **- containerPort**: 80 |
| --- |

kubectl apply -f pod.yaml

**Explore objects/resources:**

kubectl describe

**Labels and selectors**

**NodeSelector:**

Apply labels to nodes

kubectl label node <worker\_node\_name> env=prod

Create pod with node selector

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: mypodwithnodeselector  **labels**:  **env**: prod  **spec**:  **containers**:  **- image**: nginx  **name**: mynginxcontainer  **ports**:  **- containerPort**: 80  **nodeSelector**:  **env**: prod |
| --- |

**Node Affinity:**

pod-affinity.yaml

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: pod-with-enforced-affinity  **spec**:  **containers**:  **- name**: nginx  **image**: nginx  **affinity**:  **nodeAffinity**:  **requiredDuringSchedulingIgnoredDuringExecution**:  **nodeSelectorTerms**:  **- matchExpressions**:  **- key**: site  **operator**: In  **values**:  - india  - usa |
| --- |

kubectl apply -f pod-affinity.yaml

Check the status of pod (it should be pending)

kubectl label node <worker\_node\_name> site=usa

Check status of pod again (should be running)

**Taints and tolerations:**

kubectl taint node <worker\_node1\_name> site=india:NoSchedule

kubectl taint node <worker\_node2\_name> site=usa:NoSchedule

| **apiVersion**: v1  **kind**: Pod  **metadata**:  **name**: pod-with-toleration  **labels**:  **env**: prod  **spec**:  **containers**:  **- image**: nginx  **name**: mynginxcontainer  **ports**:  **- containerPort**: 80  **tolerations**:  **- key**: site  **operator**: Equal  **value**: india  **effect**: NoSchedule |
| --- |

kubectl apply pod.yaml