# Multi Container Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: multi-container
spec:
  containers:
  - image: nginx
    name: nginx
    ports:
    - containerPort: 80
  - name: tomcat
    image: tomcat
    ports:
    - containerPort: 8080
```

Verify that pod is running

```
kubectl get pods -o wide
```

Access applications inside the pods:

```
curl <pod-IP>
curl <pod-IP>:8080
```

# Labels and Selectors

## Labels

### Apply Labels

Apply the label “environment=production” label to worker 1 Node
Apply the label “environment=staging” label to worker 2 Node
Apply the label “location=india” label to all Nodes

Reference commands:

```
kubectl label nodes <one of the nodes’ name> environment=production
kubectl label nodes <the other nodes’ name> location=india
```

### Get nodes with label information

```
kubectl get nodes --show-labels
```

### Delete a label

```
kubectl label node master environment-
```

### Update a label

```
kubectl label node master --overwrite location=usa
```

## Selector

Select all the nodes with environment set to production

```
kubectl get nodes -l environment=production
```

# Scheduling

## nodeName

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    app: myapp
  name: pod
spec:
  nodeName: worker2  #Desired Node Name
  containers:
  - image: nginx
    name: pod
    ports:
    - containerPort: 80
```

## nodeSelector

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: node-selector-pod
spec:
  nodeSelector:
     color: green  # Node labels
  containers:
  - image: nginx
    name: pod
    ports:
    - containerPort: 80
```

# nodeAffinity

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: required-affinity-pod
spec:
  containers:
  - image: nginx
    name: pod
    ports:
    - containerPort: 80
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
          - key: color
            operator: In  # In, NotIn, Exists, DoesNotExist, Gt, Lt
            values:
            - red
            - green
```

```yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  name: preferred-affinity-pod
spec:
  containers:
  - image: nginx
    name: pod
    ports:
    - containerPort: 80
  affinity:
    nodeAffinity:
      preferredDuringSchedulingIgnoredDuringExecution:
      - weight: 100
        preference:
          matchExpressions:
          - key: color
            operator: In
            values:
            - purple
```

# Taints and Tolerations

## Effects:

1. NoSchedule
2. PreferNoSchedule
3. No execute

NoExecute

## Taint a node:

```
kubectl taint node ip-172-31-19-129 type=gpu:NoSchedule
```

## Tolerate the taint in a Pod

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: test-taint-pod
spec:
  containers:
  - name: nginxcontainer
    image: nginx
  tolerations:
  - key: type
    operator: Equal
    value: gpu
```

## Untaint a node:

```
kubectl taint node ip-172-31-19-129 type=gpu:NoSchedule-
```

# Logs

Print logs of specific containers in a pod:

```
kubectl logs [-f] <Podname> [containername]
kubectl logs -f multi-container <containername>
```

Print logs of all containers in a Pod

```
kubectl logs -f --all-containers multi-container
```

# Exec

## single container pod

```
kubectl exec declarative-pod -- printenv
```

## interactively into a single container pod

```
kubectl exec -it declarative-pod -- /bin/sh
```

## multi container pod (defaults to the first container)

```
kubectl exec multi-container-pod -- printenv
```

## specific container in a multi container pod

```
kubectl exec multi-container-pod -c c2 -- printenv
```