# KPLABS Course

## Certified Kubernetes Administrator 2022

## Workloads & Scheduling

**ISSUED BY**
Zeal Vora

**REPRESENTATIVE**
instructors@kplabs.in

# Module 1: Labels & Selector

## 1.1 Labels

Labels are key/value pairs that are attached to objects, such as pods


## 1.2 Selectors

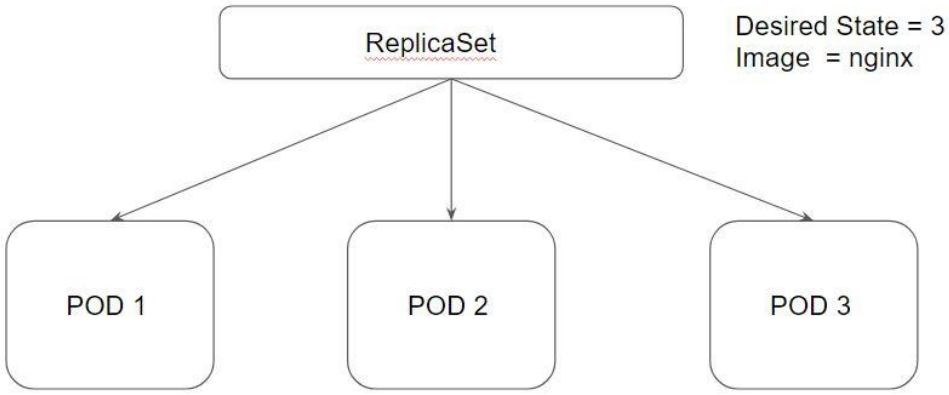
Selectors allow us to filter objects based on labels.

Example:

Show me all the objects which have a label where env: prod

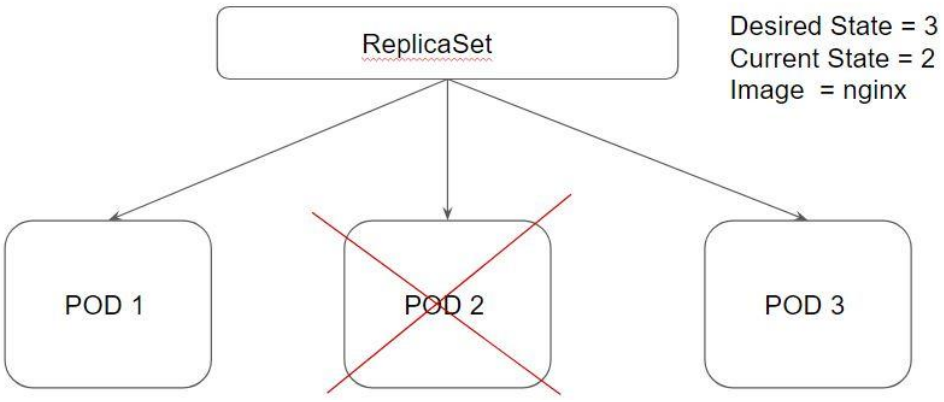
# Module 2: ReplicaSets

A ReplicaSet purpose is to maintain a stable set of replica Pods running at any given time.



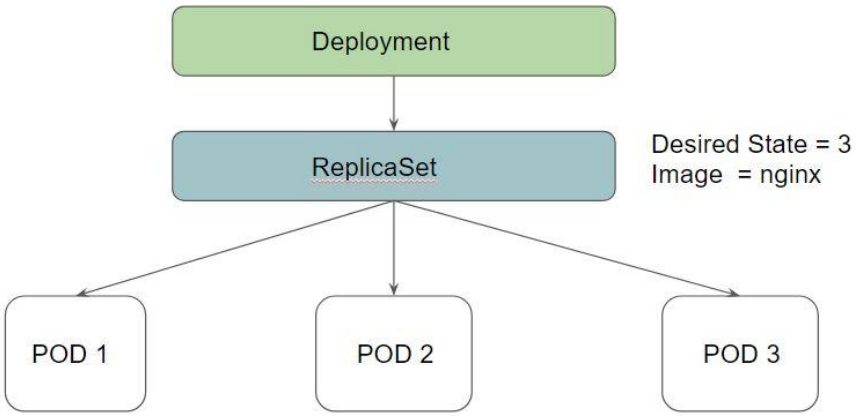

Desired State - The state of pods which is desired.

Current State - The actual state of pods that are running.



# Module 3: Deployments

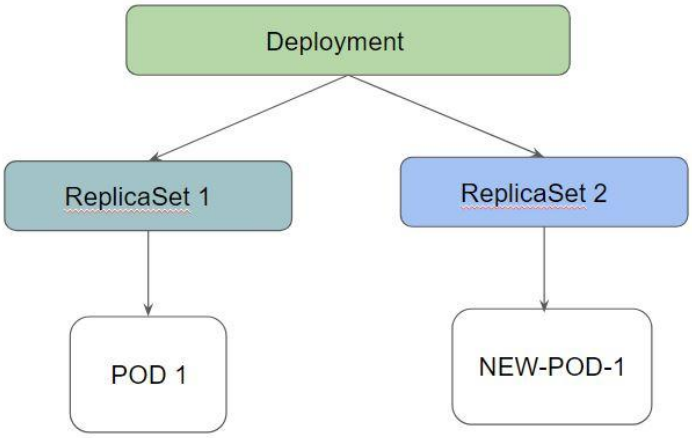
Deployments provide replication functionality with the help of ReplicaSets, along with various additional capability like rolling out of changes, rollback changes if required.




## 3.1 Benefits of Deployment - Rollout Changes

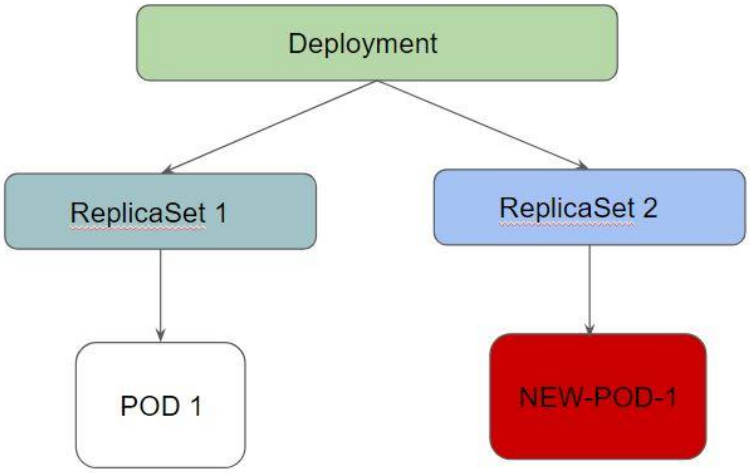
We can easily roll out new updates to our application using deployments.

Deployments will perform an update in a rollout manner to ensure that your app is not down.



## 3.2 Benefits of Deployment - Rollback Changes

Sometimes, you may want to rollback a Deployment; for example, when the Deployment is not stable, such as crash looping




Deployment ensures that only a certain number of Pods are down while they are being updated.

By default, it ensures that at least 25% of the desired number of Pods are up (25% max unavailable).

Deployments keep the history of revision which had been made.

# Module 4: Deployment Configuration

While performing a rolling update, there are two important configurations to know.

| Configuration Parameter | Description |
|---|---|
| maxSurge | Maximum Number of PODS that can be scheduled above original number of pods. |
| maxUnavailable | Maximum number of pods that can be unavailable during the update |

maxUnavailable=0 and maxSurge=20% << Full Capacity is maintained.

maxUnavailable=10% and maxSurge=0 << Update with no extra capacity. In-place updates.

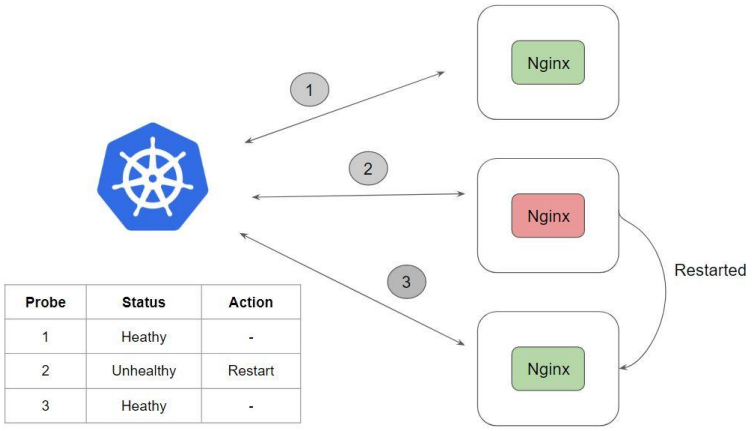
If you want fast rollout, make use of maxSurge.

If there might be a resource quota in place and partial unavailability is acceptable, maxUnavailable can be used.

# Module 5: Liveness Probe

Many applications running for long periods of time eventually transition to broken states, and cannot recover except by being restarted.

Kubernetes provides liveness probes to detect and remedy such situations.




| Probe | Status | Action |
|---|---|---|
| 1 | Heathy | - |
| 2 | Unhealthy | Restart |
| 3 | Heathy | - |

There are 3 types of probes which can be used with Liveness

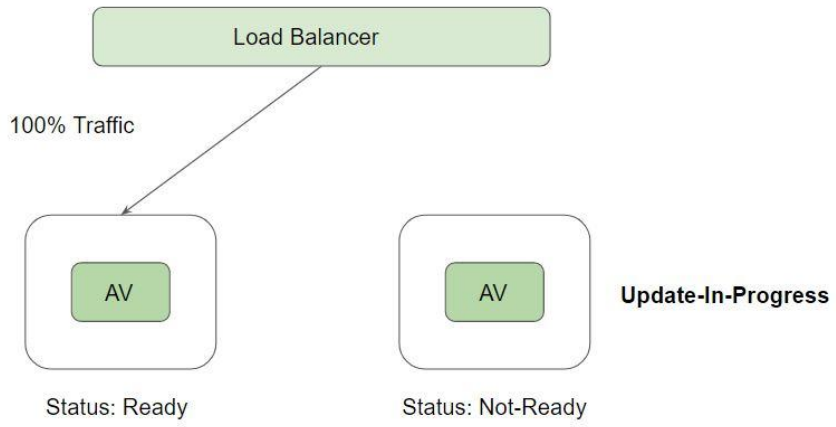
- 
- HTTP
- Command
- TCP

# Module 6:  Readiness Probe

It can happen that an application is running but temporarily unavailable to serve traffic.

For example, the application is running but it is still loading it’s large configuration files from external vendors.

In such a case, we don’t want to kill the container however we also do not want it to serve the traffic.



# Module 7: Restart Policies

By default, Docker containers will not start when they exit or when docker daemon is restarted.

Docker provides restart policies to control whether your containers start automatically when they exit, or when Docker restarts.

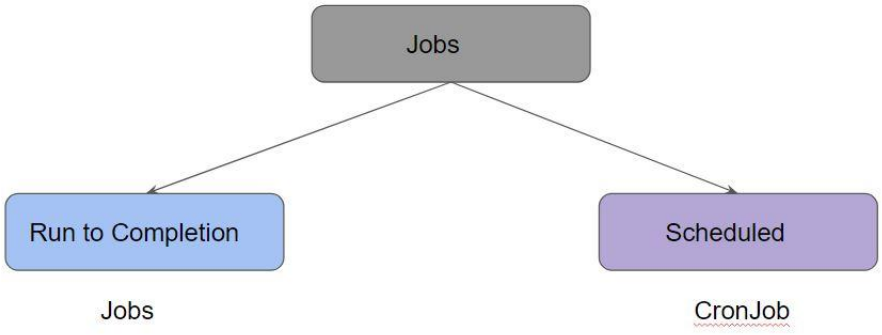

Docker Host

We can specify the restart policy by using the --restart flag with docker run command.

| Flag | Description |
|---|---|
| no | Do not automatically restart the container. (the default) |
| on-failure | Restart the container if it exits due to an error, which manifests as a non-zero exit code. |
| unless-stopped | Restart the container unless it is explicitly stopped or Docker itself is stopped or restarted. |
| always | Always restart the container if it stops. |

# Module 8: Jobs

Job is responsible for creating one or more pods to run the instruction which has been specified.




There are two types of Jobs available:

- Jobs (Run to completion)
- CronJobs

Once the task is completed, the pods are not terminated.

It is up to the user to delete old jobs after taking note of the status.

When you delete the job, all the pods associated with the job will be deleted.

In a multiple-worker node cluster, if one node fails where the job is running, that job will start the pod in the available node.

# Module 9: CronJobs

CronJob allows us to run jobs based on a time schedule.

Cron jobs are useful for creating periodic and recurring tasks, like running backups or sending emails.

You can specify cron schedule similar to the cron in Linux.

# Module 10: Generating Deployment Manifests via CLI

Till now, we have been referencing the documentation to get the manifests for objects like Pods.

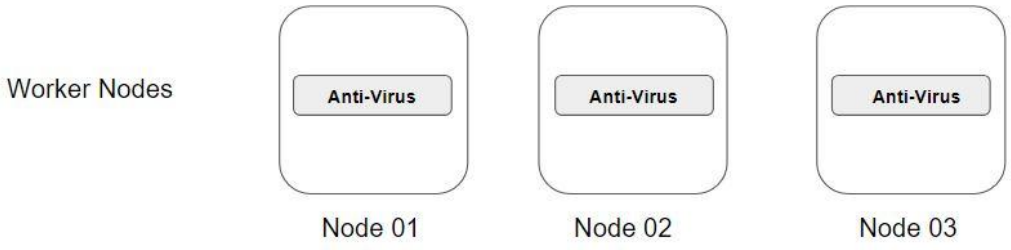
This can be a tedious process and can consume time.

```
C:\Users\Zeal Vora>kubectl run nginx --image=nginx --dry-run=client -o yaml
apiVersion: v1
kind: Pod
metadata:
  creationTimestamp: null
  labels:
    run: nginx
  name: nginx
spec:
  containers:
  - image: nginx
    name: nginx
```

# Module 11: Daemonsets

A DaemonSet can ensure that all Nodes run a copy of a Pod.

As nodes are added to the cluster, Pods are added to them.

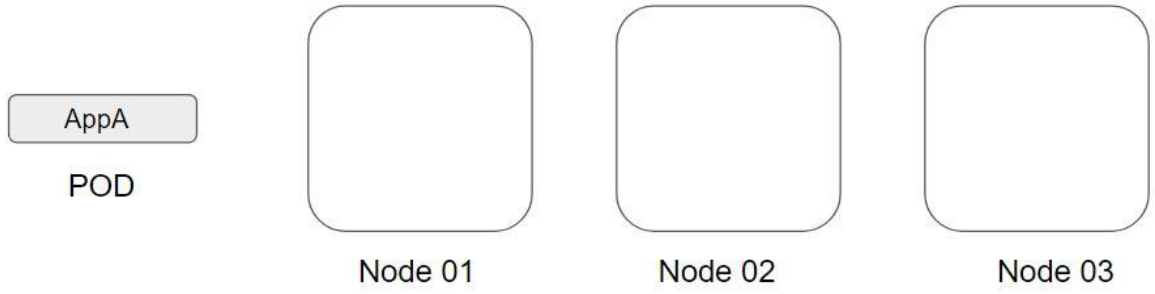


# Module 12: NodeSelector

nodeSelector allows us to add a constraint about running a pod in a specific worker node.

Use-Case:

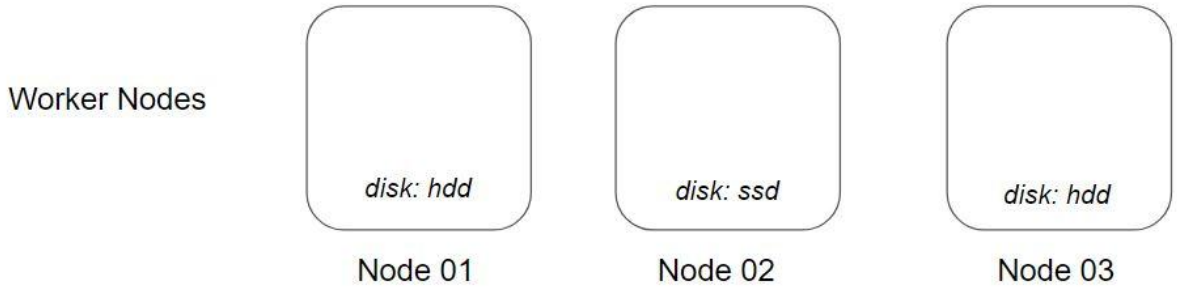
- AppA requires faster disk in order to be able to run effectively.
- Run AppA in nodes which has SSD.




Step 1: Adding a Label to the Node

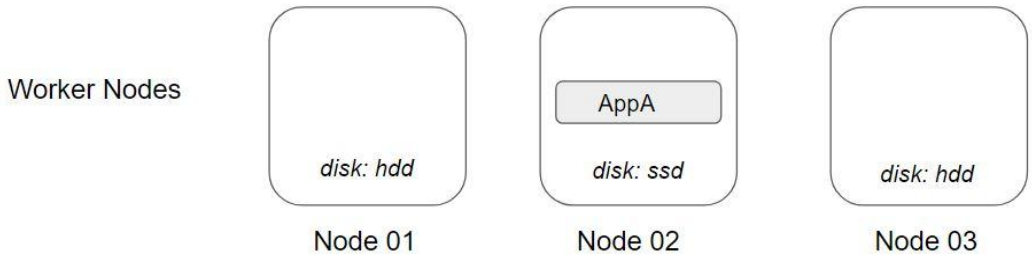
Step 1:  Add a Label to your nodes depending on their disk types.

- disk: hdd
- disk:ssd



Step 2: Using nodeSelector Configuration

Create a nodeSelector configuration to run pods only on nodes which has a label of disk=ssd




# Module 13: Node Affinity

For multiple reasons, there can be a need to run a pod on a specific worker node.

There can be multiple reasons, node hardware being the common one.

Node affinity is a set of rules used by the scheduler to determine where a pod can be placed

In Kubernetes terms, it is referred as nodeSelector, and nodeAffinity/podAffinity fields under PodSpec.

In Kubernetes, we can achieve nodeAffinity with the help of:

- nodeSelector
- nodeAffinity (more flexibility)

Node affinity is conceptually similar to nodeSelector – it allows you to constrain which nodes your pod is eligible to be scheduled on, based on labels on the node.

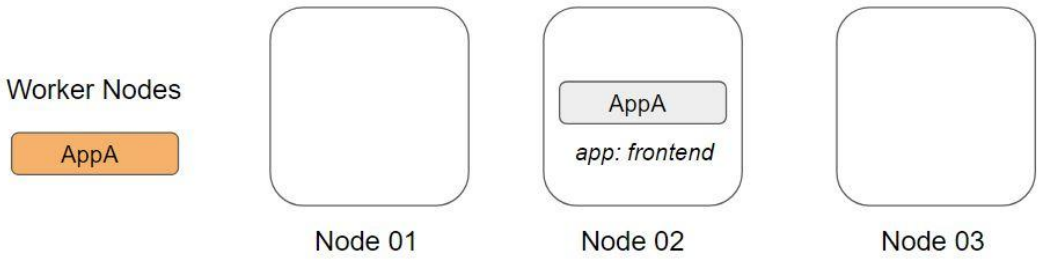
| Sr | Types | Description |
|---|---|---|
| 1 | requiredDuringSchedulingIgnoredDuringExecution | Hard Preference |
| 2 | preferredDuringSchedulingIgnoredDuringExecution | Soft Preference |

# Module 14: Pod Affinity and Pod Anti-Affinity

First Question: Where should I be running this pod?

With Node Affinity, the question became: Should I be running my pod in this node?

The considerations are still about the node. No outside information is considered apart from node.




## Step 1: Pod Selector

The first thing to figure out: “What other POD are we referring to”

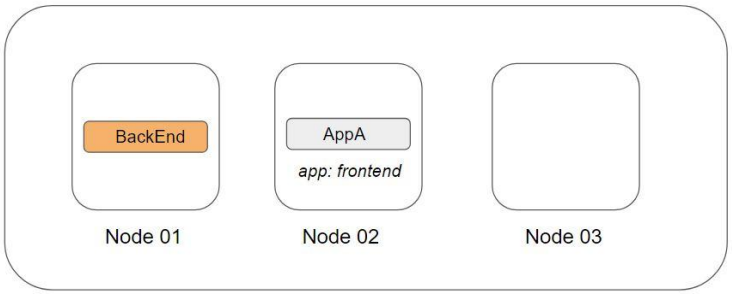
In this case, we are referring to other POD as AppA which has a label of app:frontend

Answer: I want BackEnd Pod to be running in the same place as AppA Pod.

## Step 2: Topology

Topology refers to “what does the same place mean”?

It can mean the same place if we look at the zone or region level (same AZ / same region)
It can mean a different place if we look at the host level.



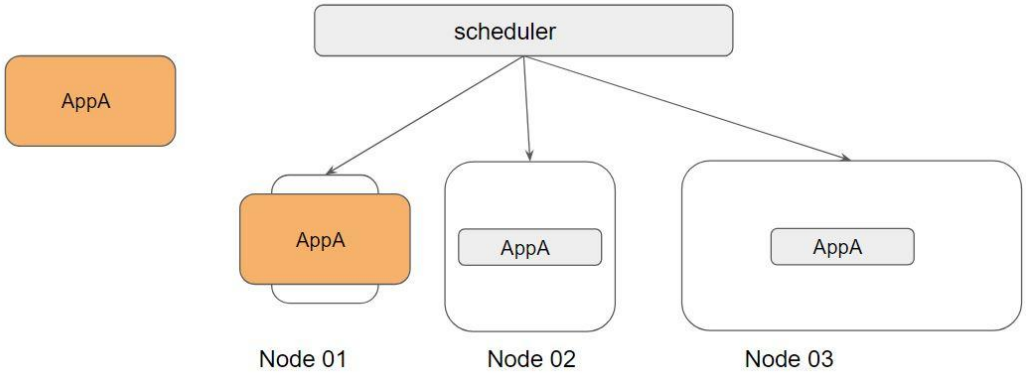

Step 3: Yes/No

Should I run my pod in the same place as the other POD?  (Yes or No)

Yes:  Pod Affinity
No:  Pod Anti-Affinity

# Module 15:  Resource Requests and Limits

If you schedule a large application in a node which has limited resource, then it will soon lead to OOM or others and will lead to downtime.




Requests and Limits are two ways in which we can control the amount of resource that can be assigned to a pod (resource like CPU and Memory)

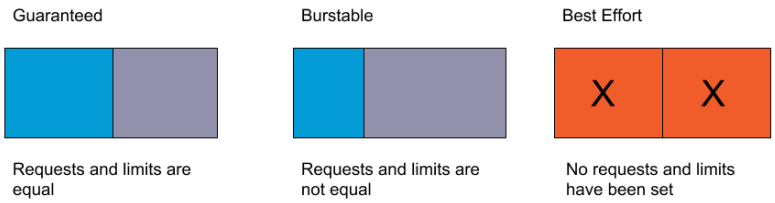
Requests:  Guaranteed to get.
Limits:  Makes sure that the container does not take node resources above a specific value.


Kubernetes Scheduler decides the ideal node to run the pod depending on the requests and limits.

If your POD requires 8GB of RAM, however, there are no nodes within your cluster which has 8GB RAM, then your pod will never get scheduled.

Kubernetes Scheduler decides the ideal node to run the pod depending on the requests and limits.

If your POD requires 8GB of RAM, however, there are no nodes within your cluster which has 8GB RAM, then your pod will never get scheduled.




---

# Module 16: Static Pods

You can directly inform the kubelet that it needs to run a specific pod.

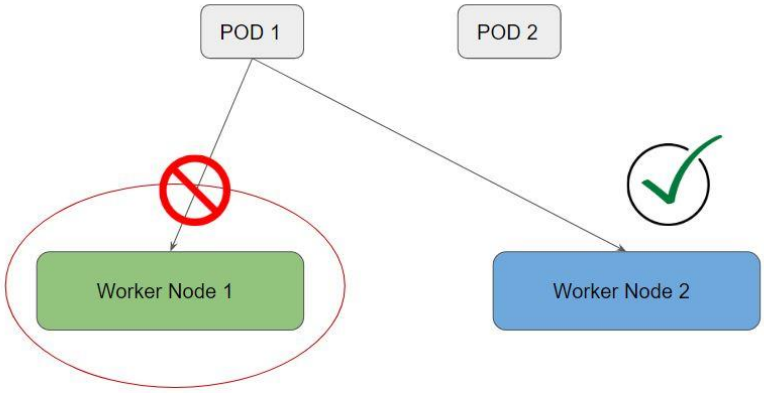
There are multiple ways in which you can tell kubelet to run a pod.

Pod created directly without schedulers are also referred as Static Pods.

# Module 17:  Taints and Toleration

## 17.1 Understanding Taints:

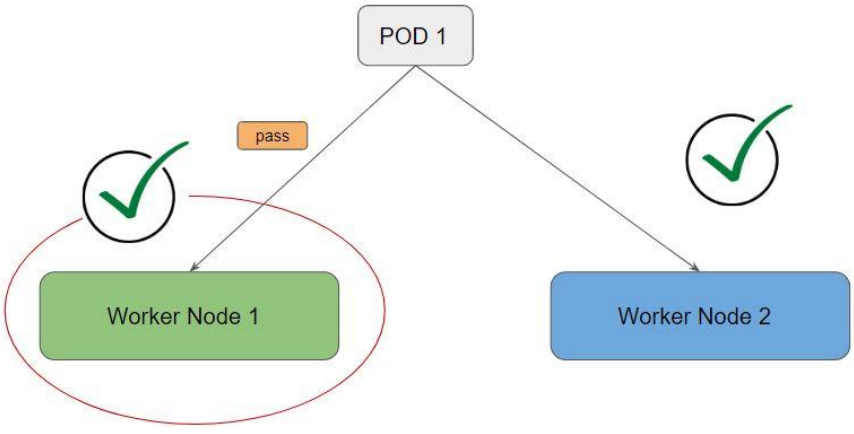
Taints are used to repel the pods from a specific node.




## 17.2 Understanding Toleration:

In order to enter the taint worker node, you need a special pass.

This pass is called toleration.



# Module 17: Components of Taints and Tolerations

A taint allows a node to refuse pod to be scheduled unless that pod has matching toleration.

We can apply toleration to a pod within the PodSpec

| Parameter | Description |
|---|---|
| key | A key is any string upto 253 characters. |
| value | The value is any string, up to 63 characters. |
| effect | • NoSchedule<br>• PreferNoSchedule<br>• NoExecute |
| operator | • Equal<br>• Exist |

```
kubectl taint nodes kubadm-worker-01 key=value:NoSchedule
```

The following table shows the effects:

| Effects | Description |
|---|---|
| NoSchedule | New pods that do not match the taint are not scheduled onto that node.<br><br>Existing pods on the node remain. |
| PreferNoSchedule | New pods that do not match the taint might be scheduled onto that node, but the scheduler tries not to.<br><br>Existing pods on the node remain. |
| NoExecute | New pods that do not match the taint cannot be scheduled onto that node.<br><br>Existing pods on the node that do not have a matching toleration are removed. |

The following are the two primary operators.

| Operator | Description |
| --- | --- |
| Equal | The key/value/effect parameters must match. This is the default. |
| Exists | The key/effect parameters must match. You must leave a blank value parameter, which matches any. |

## Join Our Discord Community

We invite you to join our Discord community, where you can interact with our support team for any course-based technical queries and connect with other students who are doing the same course.

Joining URL:

http://kplabs.in/chat