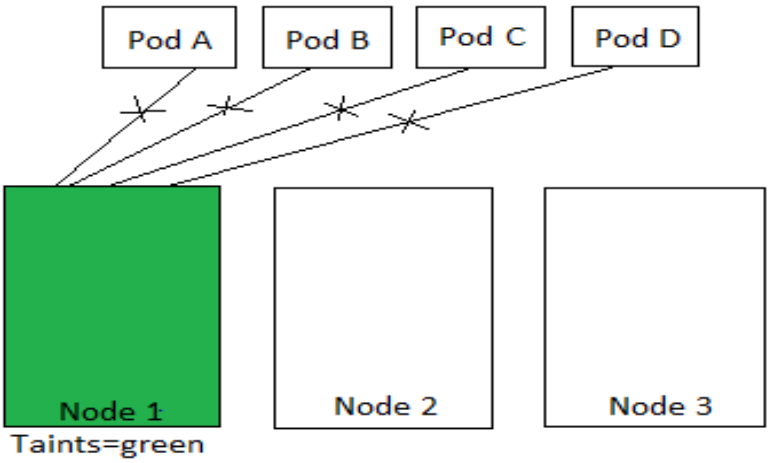
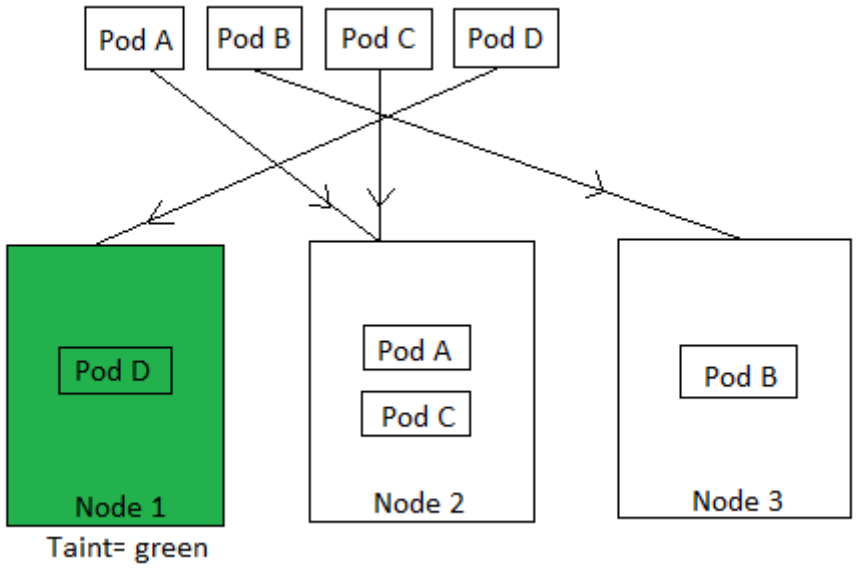
Kubernetes Taints and Toleration

Taints: -



- Placing a taint on node 1. (taint=green)
- By default, pods have no toleration which means unless specified otherwise none of the pods can tolerate any taints (Node 1 have taint=green but pods have not toleration that's why pods cannot goes through to Node 1). So, in this case none of the pod can be placed on Node1.

Taints and Tolerations: -



- Placing a taint on node 1. (taint=green)
- In this case we would like to allow only Pod D to be placed on node 1. so, we add a toleration to Pod D. Pod D is now tolerant to green taint. Node 1 can now only accept the pods that can tolerate the taints.
- The scheduler tries to place Pod A on node 1 but due to taint is thrown off and it goes to node 2. After that the scheduler then tries to place Pod B on node 1 but again due to the taint it is thrown off and placed on node 3. (because of free node) Then scheduler tries to place Pod C to the node 1it is thrown off again and it goes to node 2. So finally, when the scheduler tries to place this Pod D on node 1 since the pod is tolerant to node. The pod D is set on node 1.

Taints-Node

"Kubectl taint nodes <node-name> key=value:taint-effect "

Three types of taint-effect

1) NoSchedule: Pods will not be schedule on the node

2) PreferNoSchedule: The system will try to avoid placing a pod on the node but that is not guaranteed.

3) NoExecute: New pods will not be schedule on the node and existing pods on the node. If they do not tolerate the taint the pod will be killed.

Example:

1) First add taint to node 1

"Kubectl taint nodes node1 app=green:NoSchedule"

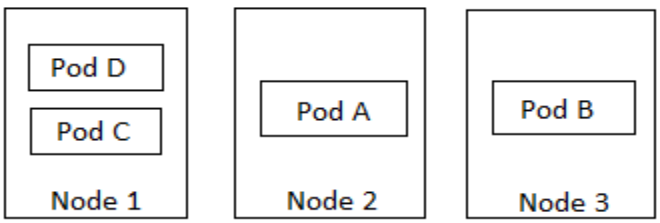
2) Create a pod with toleration or add toleration to the pod

Pod-definition.yml

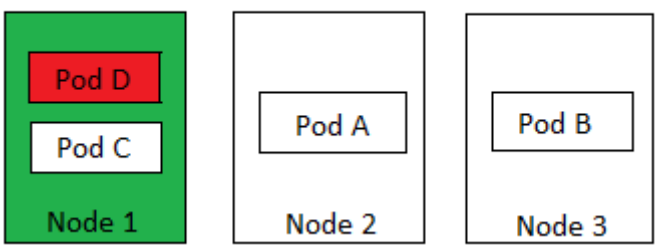
```
apiVersion: v1
kind: Pod
metadata:
    name: myapp-pod
spec:
  containers:
  - name: nginx-container
    image: nginx
  tolerations:
   - key: "app"
     operator: "Equal"
     value: "green"
     effect: "NoSchedule"
```

Taint-NoExecute:

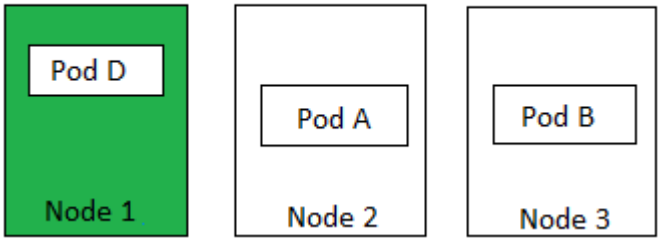
I) So, we do not have any taints and toleration at this point.



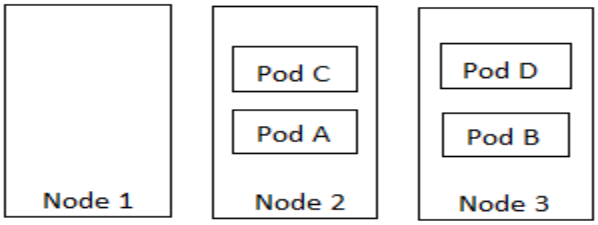
II) we then decide to dedicate node 1 for a special application and as such we obtained the node with the application name and toleration to pod that belongs to application which happens to be Pod D.



III) In this case while tending the node we set the taint effect to NoExecute and as such once detained on the node takes effect it kills Pod C from the node. The Pod D continues to run on the node 1.



- Taints and toleration are only meant to restrict nodes from accepting certain pods.
- Taints and toleration do not tell the pod to go to a particular node. Instead, it tells the node to only accept pods with certain toleration.



It does not guarantee that Pod D will always be placed on Node 1. Since there is no taints and toleration applied on the other two nodes. Pod D may be placed on any of the other two nodes.

Master node:

Master node is technically just another node that has all the capabilities of hosting a pod and it runs all the management software. Scheduler does not schedule any pod on the master node. Taints is set on the master node automatically that prevent any pods from being schedule on the master node.

"Kubectl describe node kubemaster | grep Taint"