SCHEDULING

# Scheduling

## Taints

A node with a particular taint will repel Pods without tolerations for that taint. A taint is expressed as **`key=value:effect`**. The key and the value are created by the administrator.

The key and value used can be any legal string, and this allows flexibility to prevent Pods from running on nodes based off of any need. If a Pod does not have an existing toleration, the scheduler will not consider the tainted node.

***There are three effects, or ways to handle Pod scheduling. Click on each box to learn more about these effects.***

### Ways to Handle Pod Scheduling

NoSchedule — Close

The scheduler will not schedule a Pod on this node, unless the Pod has this toleration. Existing Pods continue to run, regardless of toleration.

PreferNoSchedule — Close

The scheduler will avoid using this node, unless there are no untainted nodes for the Pods toleration. Existing Pods are unaffected.

NoExecute — Close

This taint will cause existing Pods to be evacuated and no future Pods scheduled. Should an existing Pod have a toleration, it will continue to run. If the Pod **`tolerationSeconds`** is set, they will remain for that many seconds, then be evicted. Certain node issues will cause the kubelet to add 300 second tolerations to avoid unnecessary evictions.

If a node has multiple taints, the scheduler ignores those with matching tolerations. The remaining unignored taints have their typical effect.

The use of **`TaintBasedEvictions`** is still an alpha feature. The kubelet uses taints to rate-limit evictions when the node has problems.