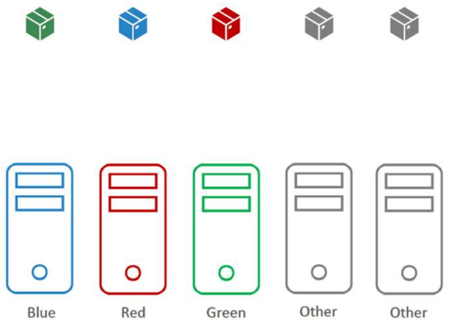
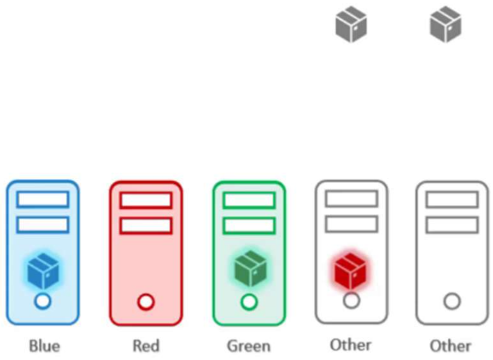
Taints and Tolerations Vs Node Affinity

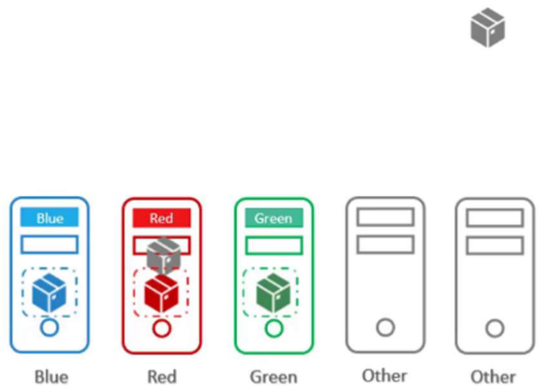
Now that we have learned about taints, and tolerations, and node affinity, let us tie together the two concepts through a fun exercise. We have three nodes and three pods each in three colors, blue, red, and green. The ultimate aim is to place the blue pod in the blue node, the red pod in the red node, and likewise for green. We are sharing the same Kubernetes cluster with other teams. So there are other pods in the cluster as well as other nodes. We do not want any other pod to be placed on our node. Neither do we want our pods to be placed on their nodes



Let us first try to solve this problem using taints and tolerations. We apply a taint to the nodes marking them with their colors, blue, red, and green, and we then set a toleration on the pods to tolerate the respective colors. When the pods are now created, the nodes ensure they only accept the pods with the right toleration. So the green pod ends up on the green node, and the blue pod ends up on the blue node. However, taints and tolerations does not guarantee that the pods will only prefer these nodes. So the red node ends up on one of the other nodes that do not have a taint or toleration set. This is not desired.



Let us try to solve the same problem with node affinity. With node affinity, we first label the nodes with their respective colors, blue, red, and green. We then set node selectors on the pods to tie the pods to the nodes. As such, the pods end up on the right nodes. However, that does not guarantee that other pods are not placed on these nodes. In this case, there is a chance that one of the other pods may end up on our nodes. This is not something we desire.



As such, a combination of taints, and tolerations, and node affinity rules can be used together to completely dedicate nodes for specific pods. We first use taints and tolerations to prevent other pods from being placed on our nodes, and then we use node affinity to prevent our pods from being placed on their nodes.