

# **KUBERNETES - COMPUTING AND SCHEDULING**

**Github repo..**

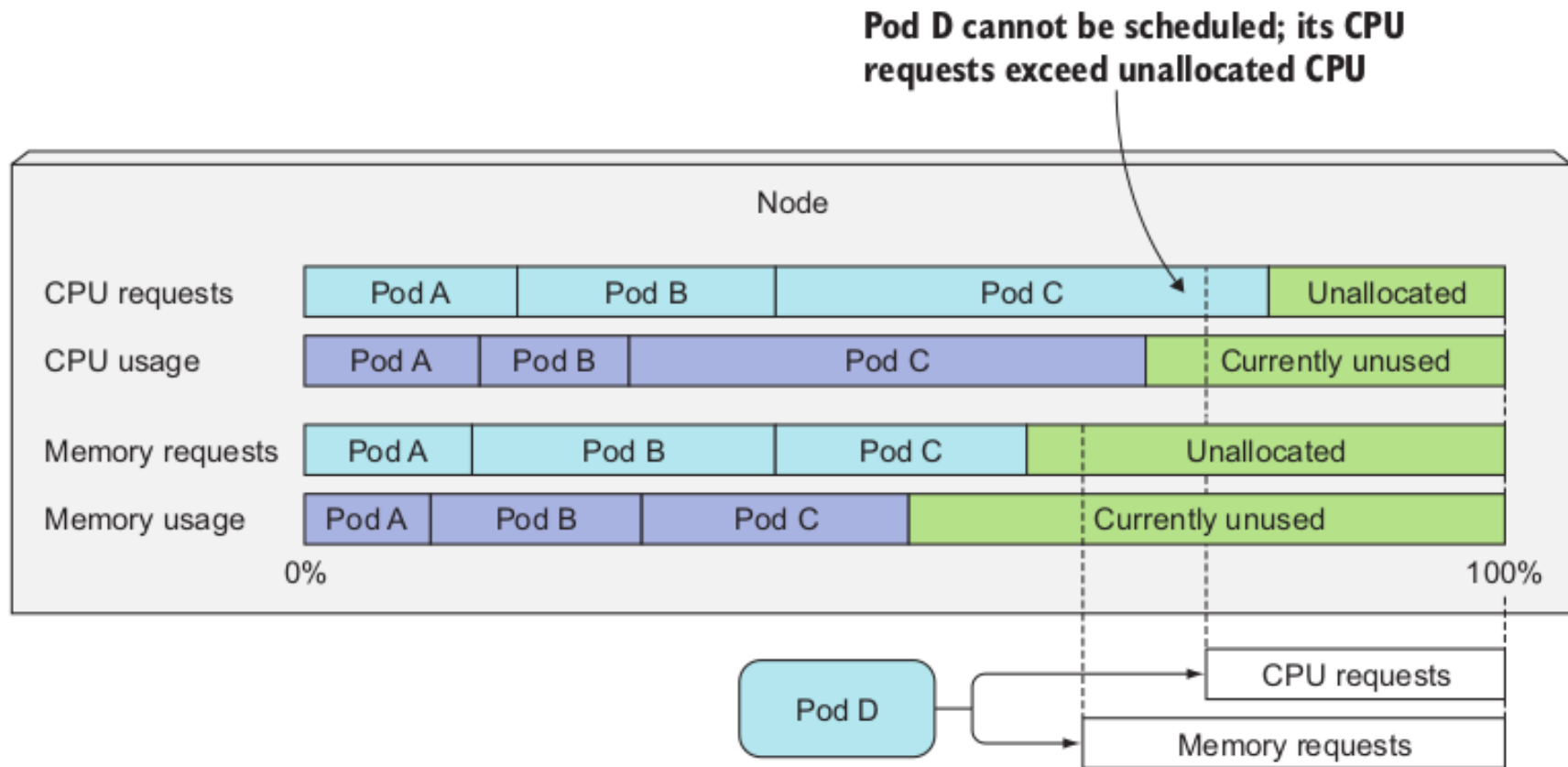
**<https://github.com/vsaini44/KubernetesRepo.git>**

# Requesting Resources for pod

**When creating a pod, you can specify the amount of CPU and memory that a container needs (these are called requests) and a hard limit on what it may consume known as limits. They're specified for each container individually, not for the pod as a whole.**

**The pod's resource requests and limits are the sum of the requests and limits of all its containers.**

# Scheduler with Requests



# Limits for containers

**Setting resource requests for containers in a pod ensures each container gets the minimum amount of resources it needs. Limits let's you manage the other side of the coin :**

**The maximum amount the container will be allowed to consume.**

# Understanding limit range

**The limits specified in a LimitRange resource apply to each individual pod/container or other kind of object created in the same namespace as the LimitRange Object.**

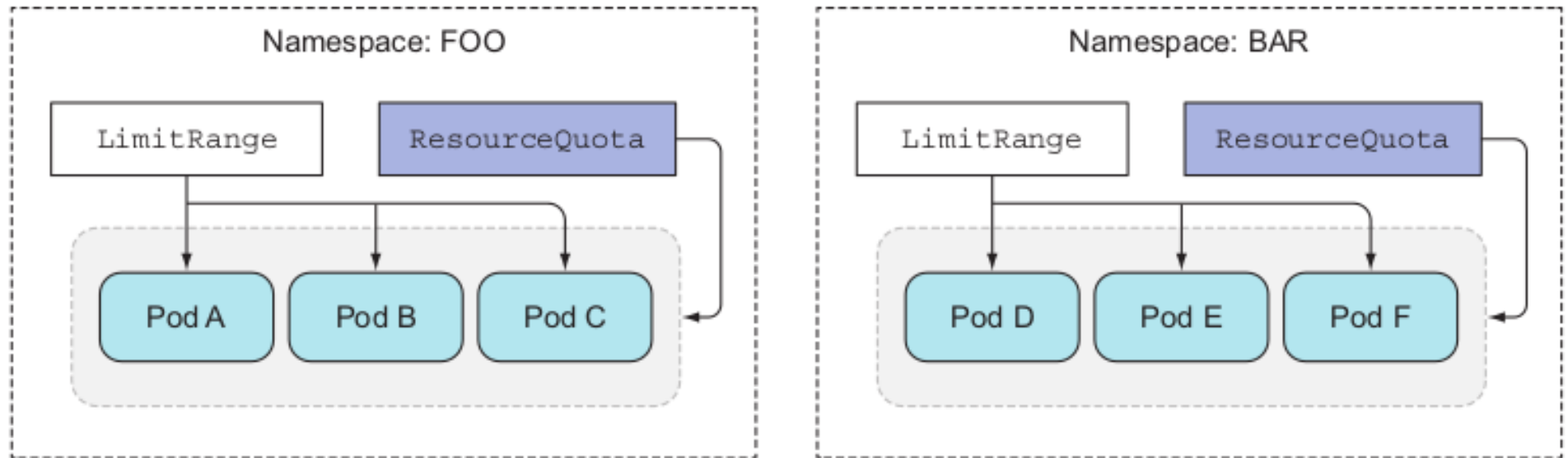
**They don't limit the total amount of resources available across all the pods in the namespace**

# Understanding Quota

**A ResourceQuota limits the amount of computational resources the pods and the amount of storage PersistentVolumeClaims in a namespace can consume.**

**It can also limit the number of pods, claims, and other API objects users are allowed to create inside the namespace.**

# Difference



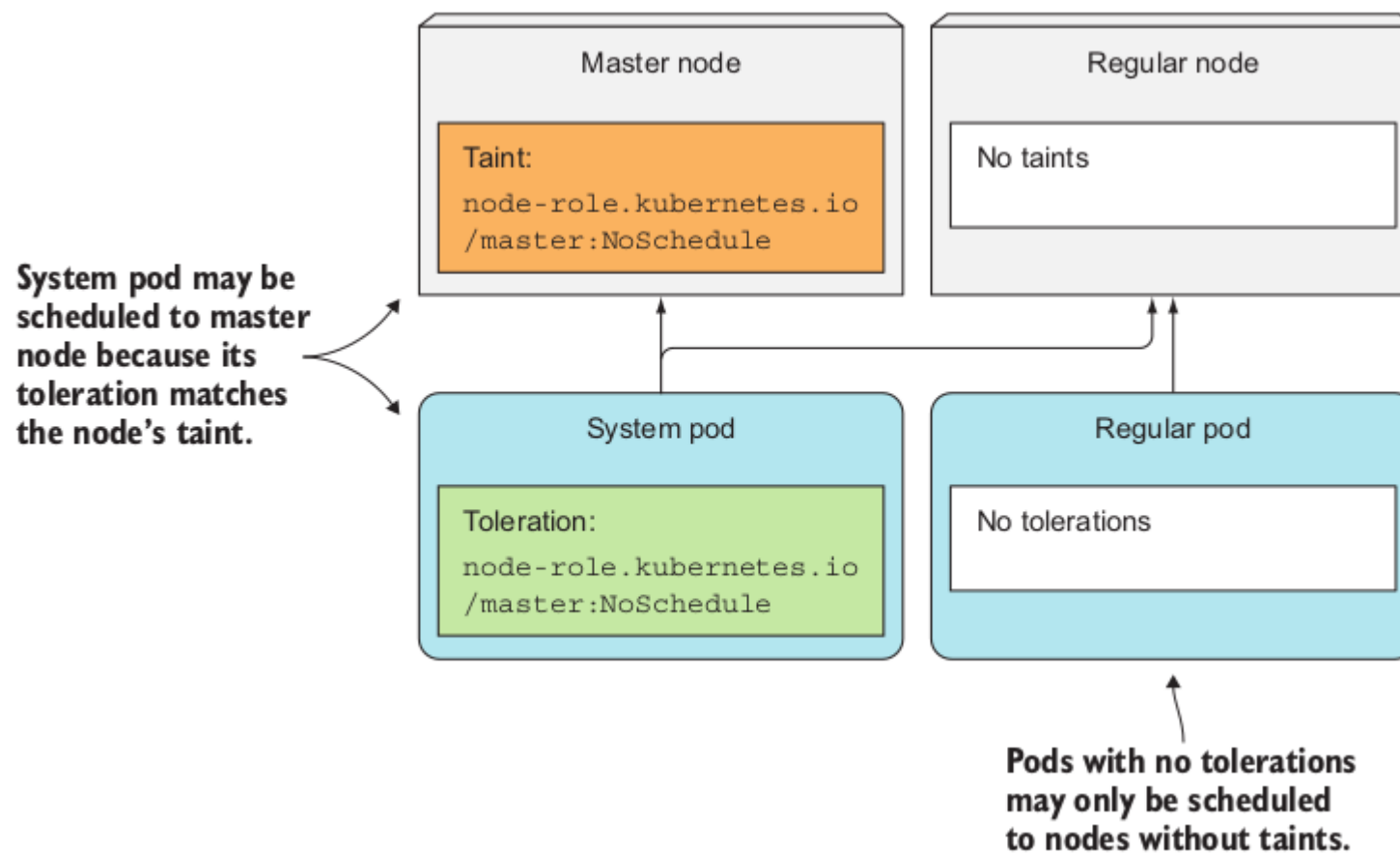


# Taint and Tolerations

**Taints and tolerations allow the node to control which pods should (or should not) be scheduled on them.**

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

# Taint and Tolerations




# Node Affinity

**To get pods to be scheduled to specific nodes Kubernetes provides nodeSelectors and nodeAffinity.**

**With node affinity we can tell Kubernetes which nodes to schedule to a pod using the labels on each node.**

# Node Affinity

 **requiredDuringScheduling** - means the rules defined under this field specify the labels the node must have for the pod to be scheduled to the node.

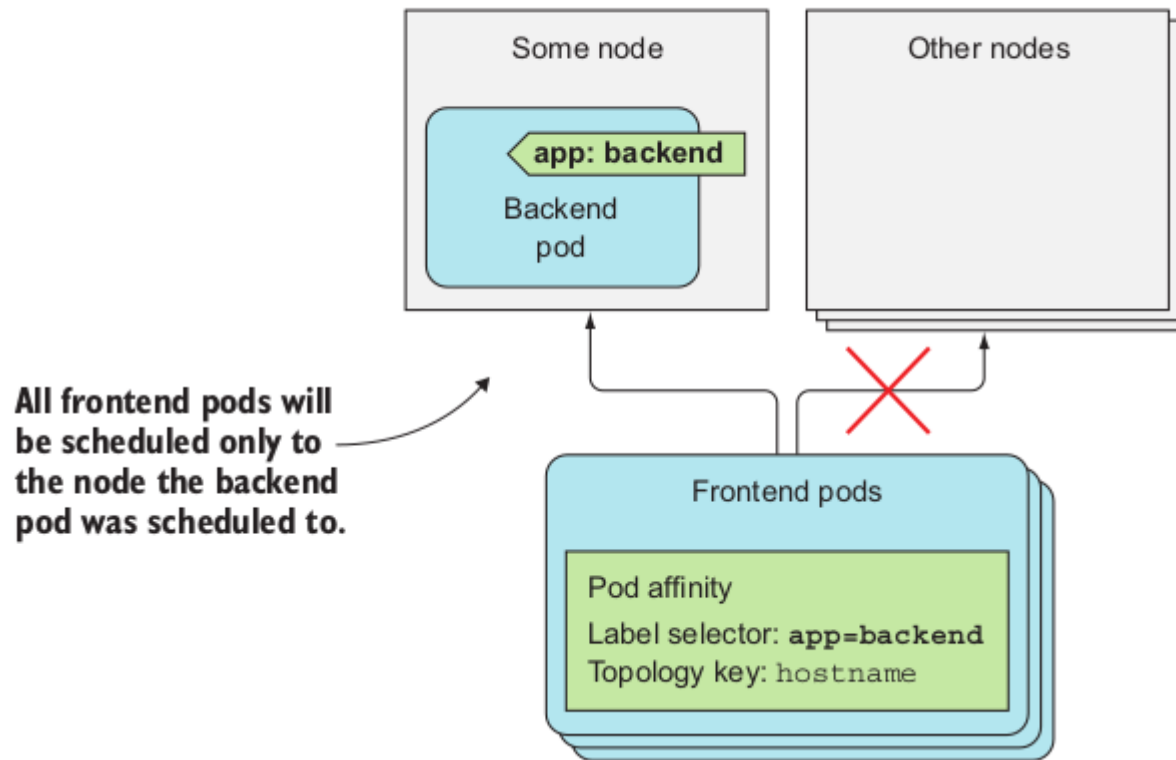
 **IgnoredDuringExecution** - means the rules defined under the field don't affect pods already executing on the node

# Pod Affinity and Anti affinity

**Pod affinity and anti-affinity allows placing pods to nodes as a function of the labels of other pods.**

**These Kubernetes features are useful in scenarios like: an application that consists of multiple services, some of which may require that they be co-located on the same node for performance reasons; replicas of critical services shouldn't be placed onto the same node to avoid loss in the event of node failure.**

# Pod Affinity



# Pod Antiaffinity

