

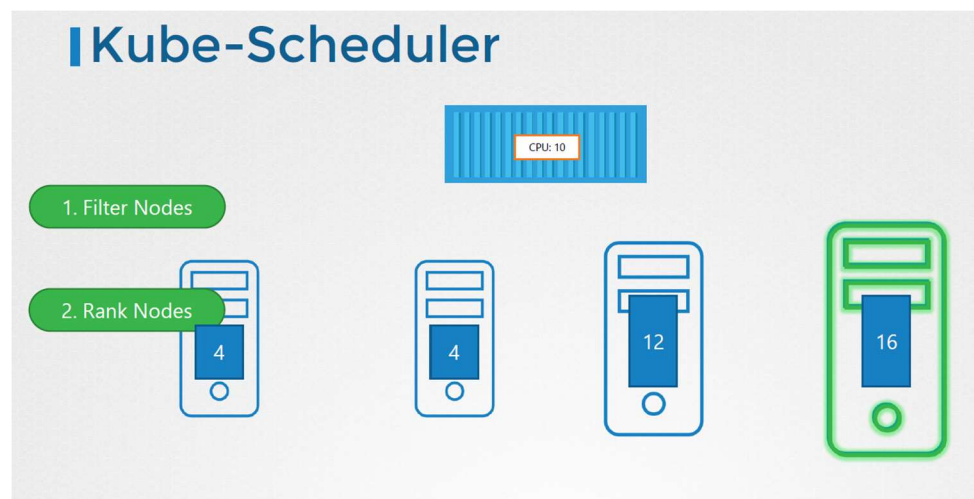
Kube Scheduler

In this lecture, we will talk about kube-scheduler. Earlier, we discussed that the Kubernetes scheduler is responsible for scheduling pods on nodes. Now don't let the graphic mislead you. Remember, the scheduler is only responsible for deciding which pod goes on which node. It doesn't actually place the pod on the nodes. That's the job of the kubelet. The kubelet, or the captain on the ship, is who creates the pod on the ships. The scheduler only decides which pod goes where. Let's look at how the scheduler does that in a bit more detail.

In Kubernetes, the scheduler decides which nodes the pods are placed on depending on certain criteria. You may have pods with different resource requirements. You can have nodes in the cluster dedicated to certain applications. So how does the scheduler assign these pods? The scheduler looks at each pod and tries to find the best node for it. For example, let's take one of these pods, the big blue one. It has a set of CPU and memory requirements. The scheduler goes through two phases to identify the best node for the pod.

In the first phase, the scheduler tries to filter out the nodes that do not fit the profile for this pod. For example, the nodes that do not have sufficient CPU and memory resources requested by the pod. So the first two small nodes are filtered out. We are now left with the two nodes on which the pod can be placed.

Now, how does the scheduler pick one from the two? The scheduler ranks the nodes to identify the best fit for the pod. It uses a priority function to assign a score to the nodes on a scale of zero to 10. For example, the scheduler calculates the amount of resources that would be free on the nodes after placing the pod on them. In this case, the one on the right would have six CPUs free if the pod was placed on it, which is four more than the other one. So it gets a better rank, and so it wins. That's how a scheduler works at a high level.



And, of course, this can be customized, and you can write your own scheduler as well. There are many more topics to look at, such as resource requirements, limits, taints and tolerations, node selectors, affinity rules, etc.,

So how do you install the kube-scheduler? Download the kube-scheduler binary from the Kubernetes release page, extract it, and run it as a service. When you run it as a service, you specify the scheduler configuration file.

```
wget https://storage.googleapis.com/kubernetes-release/release/v1.13.0/bin/linux/amd64/kube-scheduler

kube-scheduler.service
ExecStart=/usr/local/bin/kube-scheduler \\\
--config=/etc/kubernetes/config/kube-scheduler.yaml \\\
--v=2
```

So how do you view the kube-scheduler server options? Again, if you set it up with the kubeadm tool, the kubeadm tool deploys the kube-scheduler as a pod in the kube system namespace on the master node. You can see the options within the pod definition file located at `/etc/kubernetes/manifests/folder`. You can also see the running process and the effective options by listing the process on the master node and searching for kube-scheduler.

```
cat /etc/kubernetes/manifests/kube-scheduler.yaml

spec:
  containers:
  - command:
    - kube-scheduler
    - --address=127.0.0.1
    - --kubeconfig=/etc/kubernetes/scheduler.conf
    - --leader-elect=true
```

```
ps -aux | grep kube-scheduler

root      2477  0.8  1.6 48524 34044 ?        Ssl  17:31   0:08 kube-scheduler --
address=127.0.0.1 --kubeconfig=/etc/kubernetes/scheduler.conf --leader-elect=true
```