

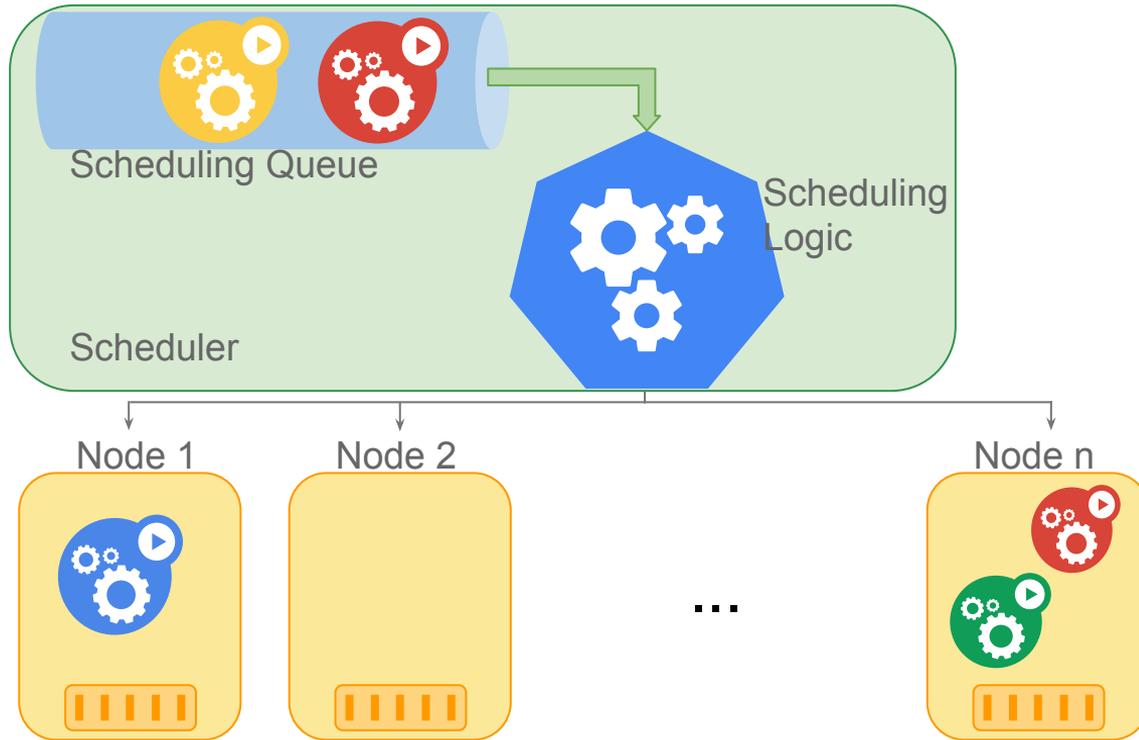
Kubernetes SIG Scheduling Deep Dive

Bobby (Babak) Salamat - Google
Jonathan Basseri - Google

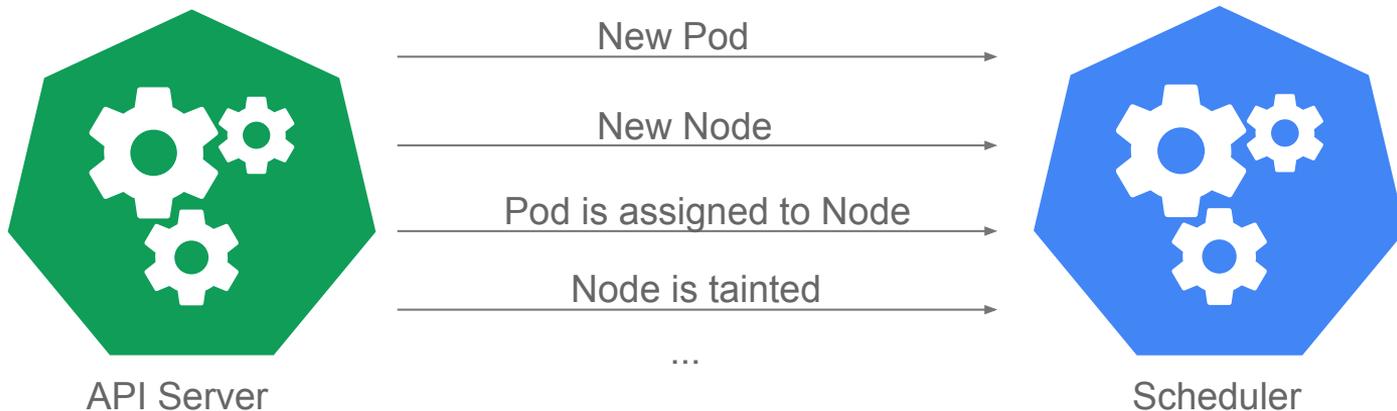
KubeCon Europe 2018

Introduction to the Scheduler

Scheduler places Pods on Nodes

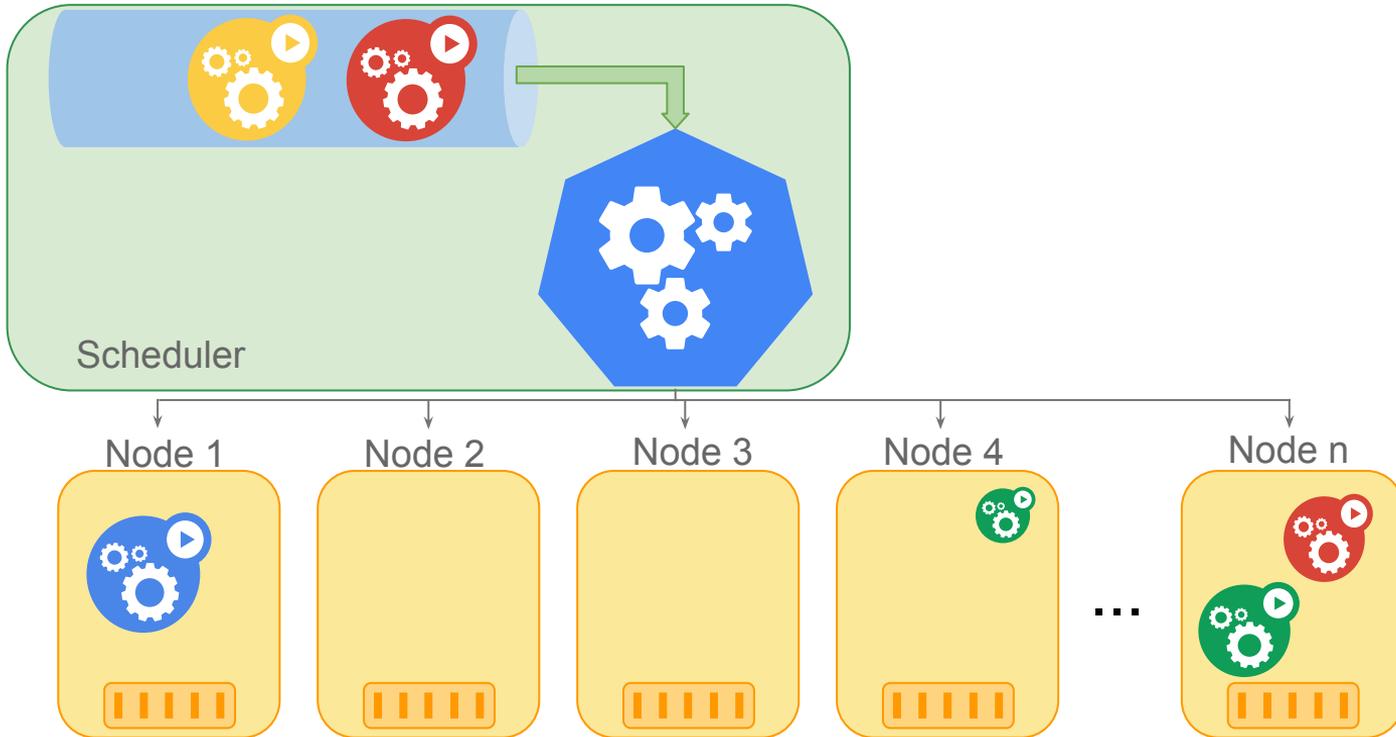


Scheduler caches the state of the cluster

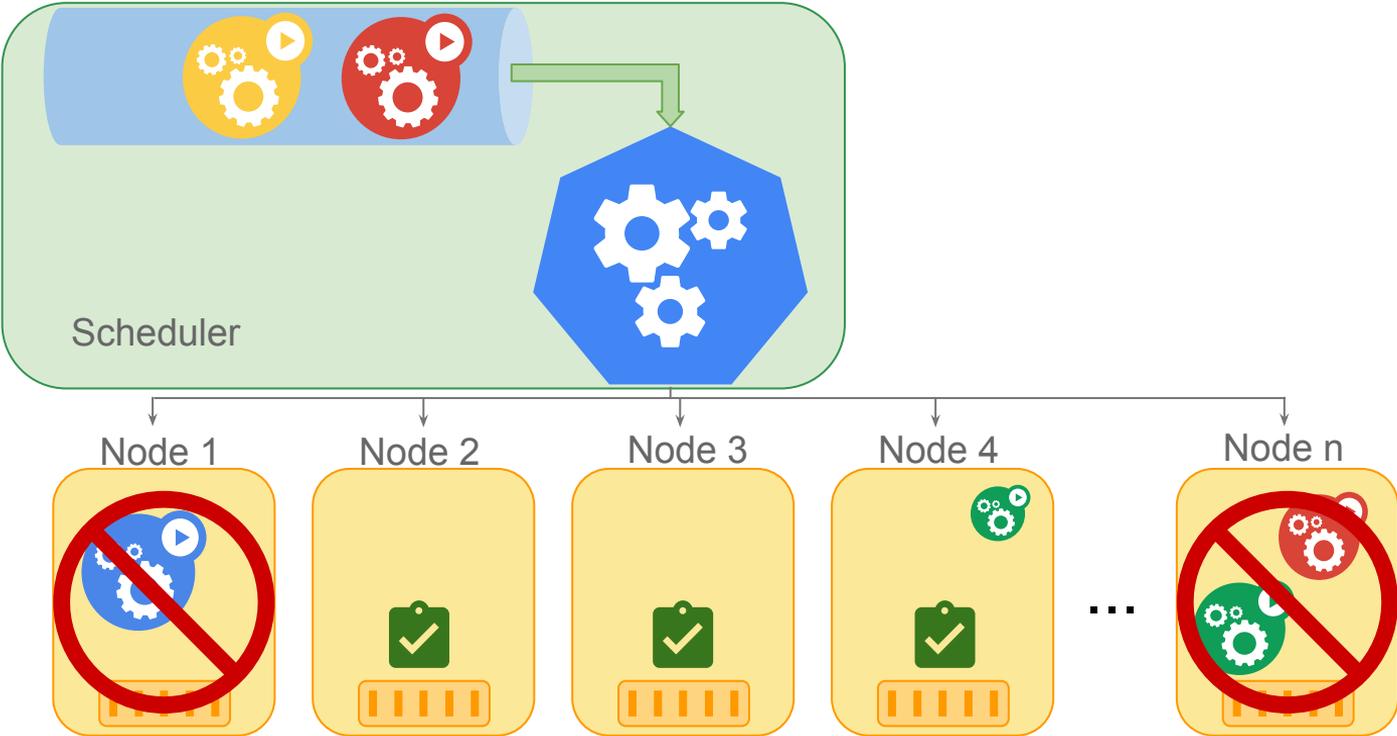


Scheduler keeps its cache updated by receiving events from the API server.

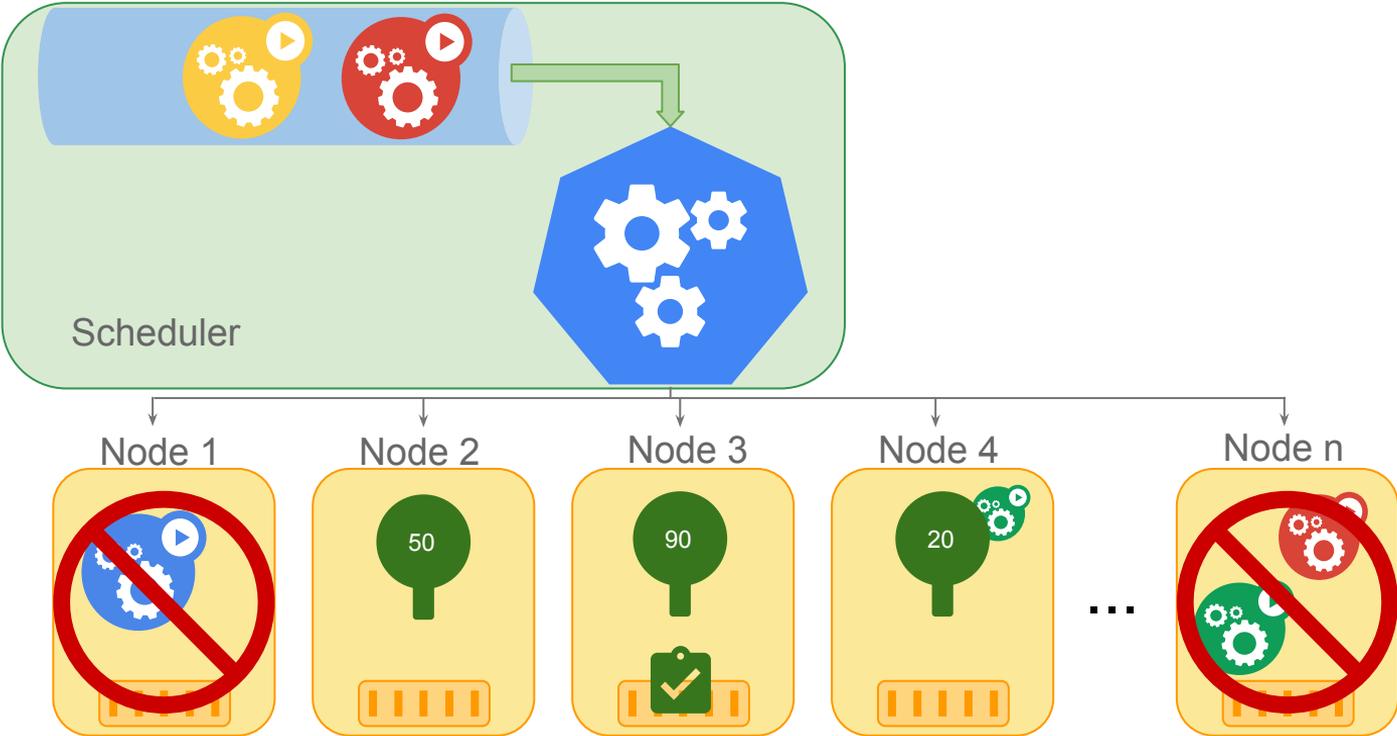
Scheduler schedules one Pod at a time



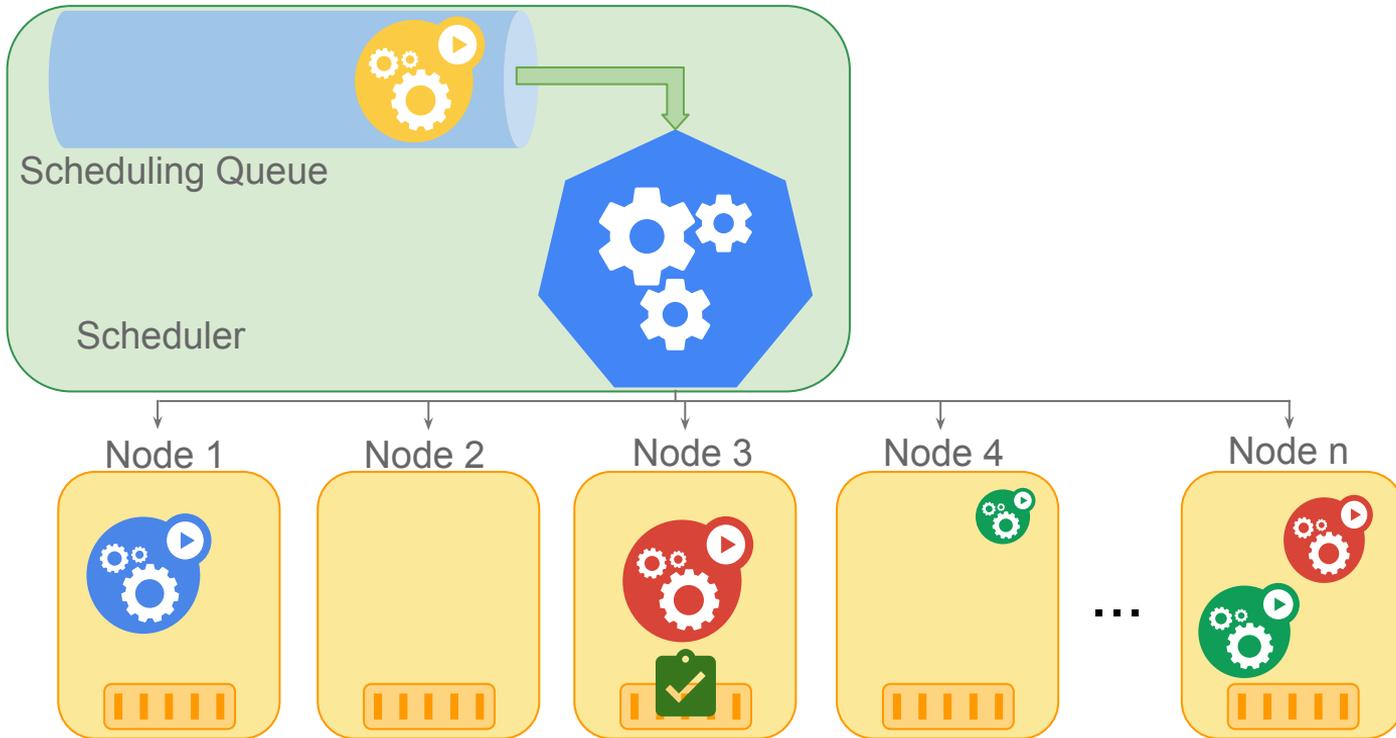
Predicate functions filter out Nodes



Priority functions rank the remaining Nodes



When Pod is bound the Kubelet is notified



Scheduling Scenarios

How can I spread my service in different zones?

How can I save my special hardware for a specific workload?

How could we prevent our pods from landing on unhealthy nodes?

How can I run a webserver with a memcached instance on the same Node?

How do I ensure a certain number of Pods of my service will always run?

How should I run my cluster more efficiently to save money?



Labels

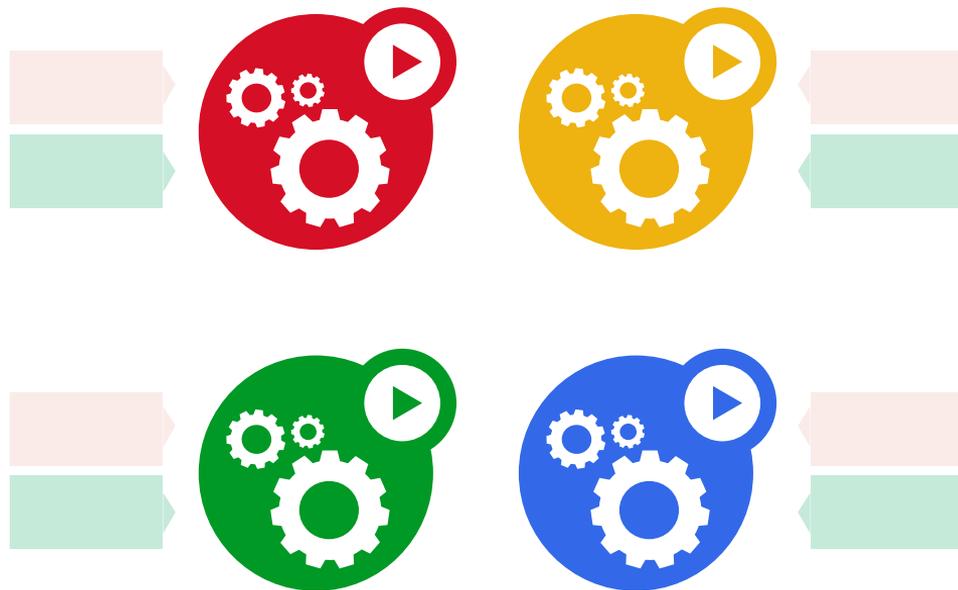
Arbitrary metadata

Attached to **any API object**

Generally represent **identity**

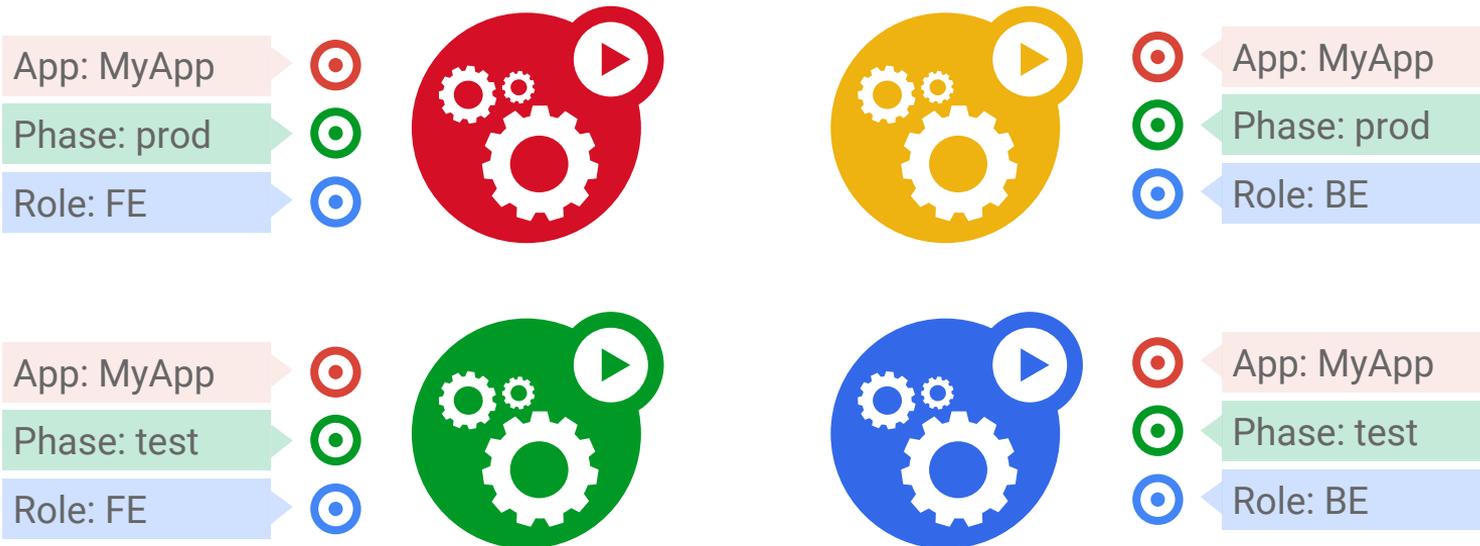
Queryable by **selectors**

- think SQL *'select ... where ...'*



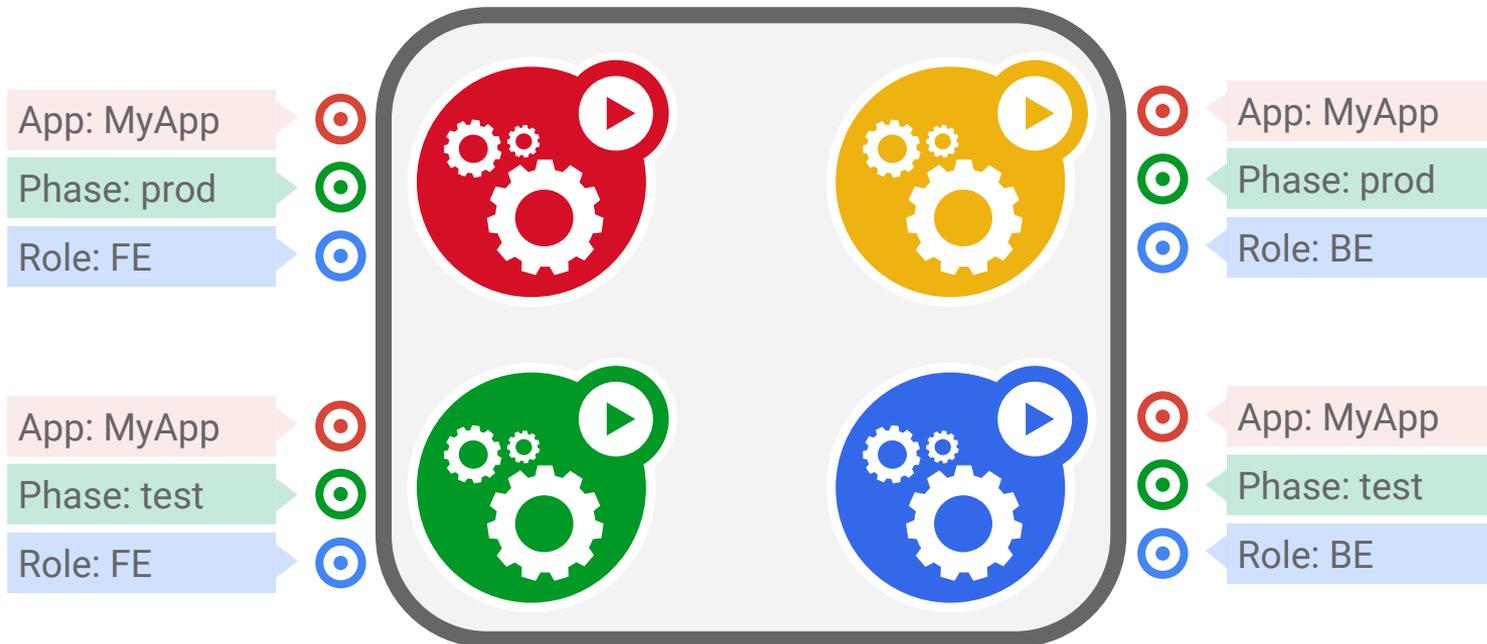


Selectors





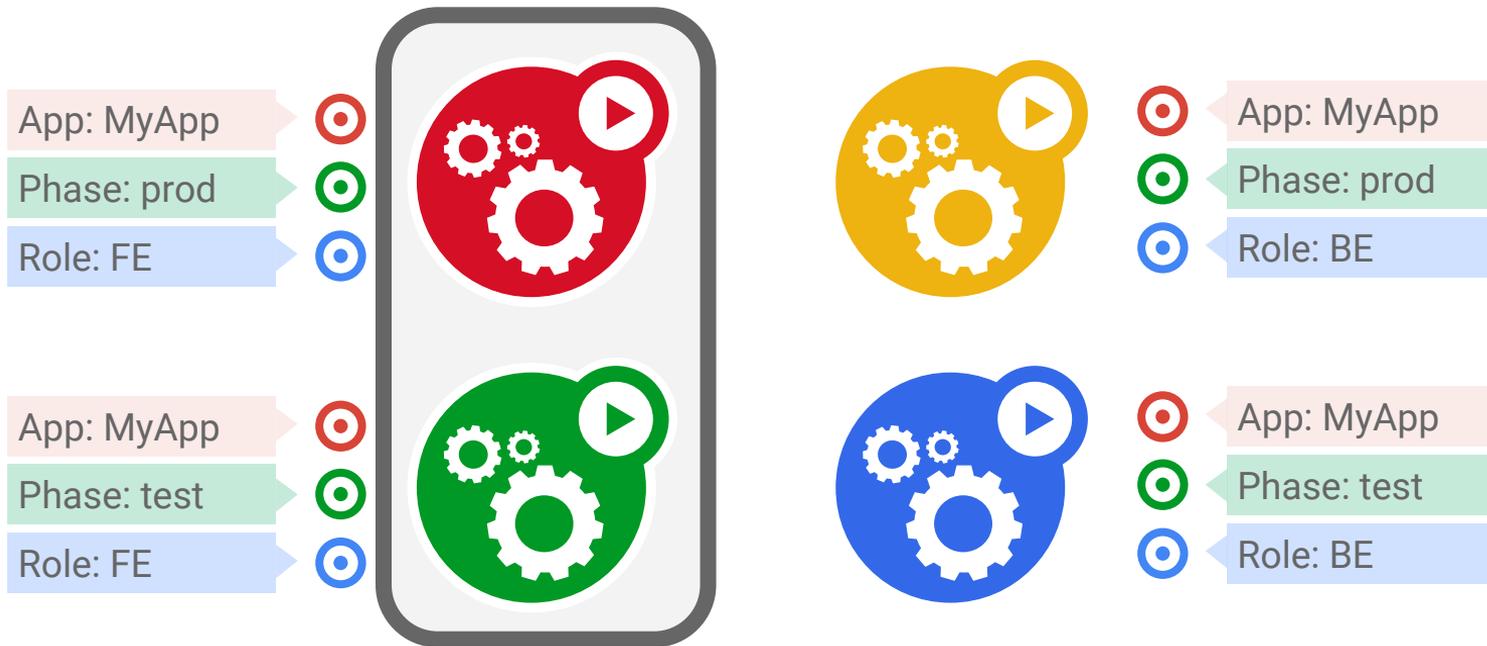
Selectors



App = MyApp



Selectors



App = MyApp, Role = FE



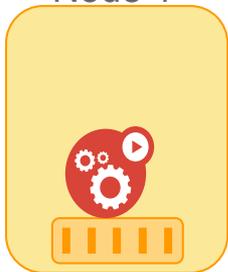
Run my Pods on a specific group of Nodes

Pod Spec

```
nodeAffinity:  
  labelSelector: "zone" In  
  {"central"}
```



Node 1



zone: west

Node 2



zone: central

Node 3



zone: central



Run Pods of different services together

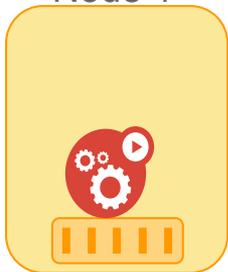
Pod Spec

```
podAffinity:  
  labelSelector: "service" In {"B"}  
  topologyKey: "zone"
```



service: A

Node 1



zone: west

Node 2



zone: central

service: B

Node 3



zone: central



Run Pods of different services together

Pod Spec

```
podAffinity:  
  labelSelector: "service" In {"B"}  
  topologyKey: "hostname"
```

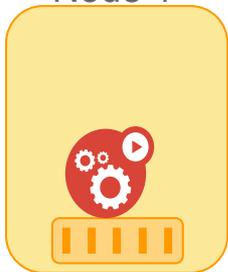


service: A



Pod is not schedulable

Node 1



zone: west

Node 2



service: B

zone: central

Node 3



zone: central



Run Pods of different services together

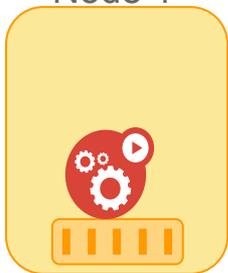
Pod Spec

```
podAffinity:  
{  
  labelSelector: "service" In {"B"}  
  topologyKey: "zone"  
},  
{  
  labelSelector: "service" In {"B"}  
  topologyKey: "hostname"  
  (preferred)  
}
```



service: A

Node 1



zone: west

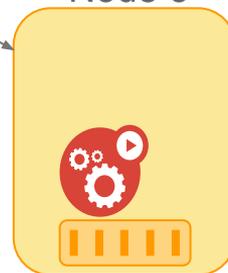
Node 2



service: B

zone: central

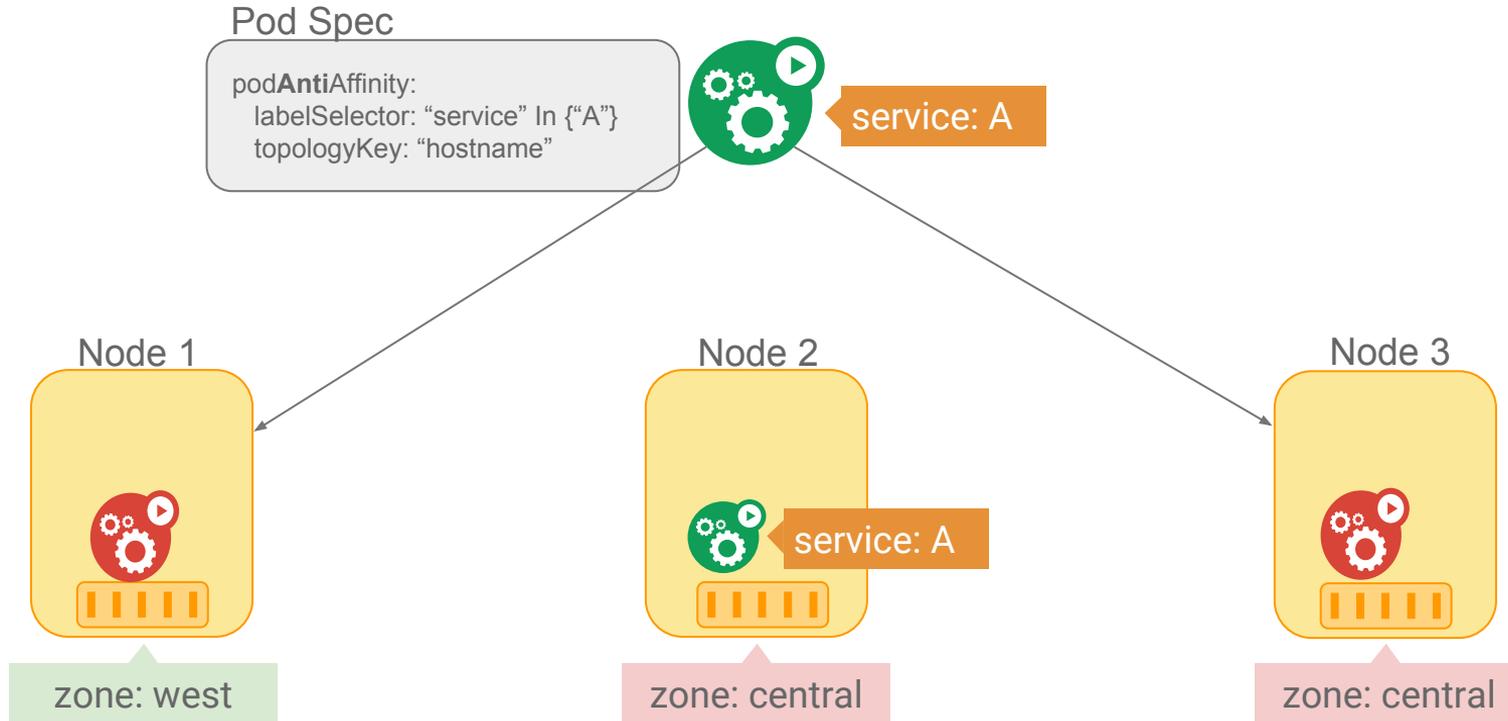
Node 3



zone: central

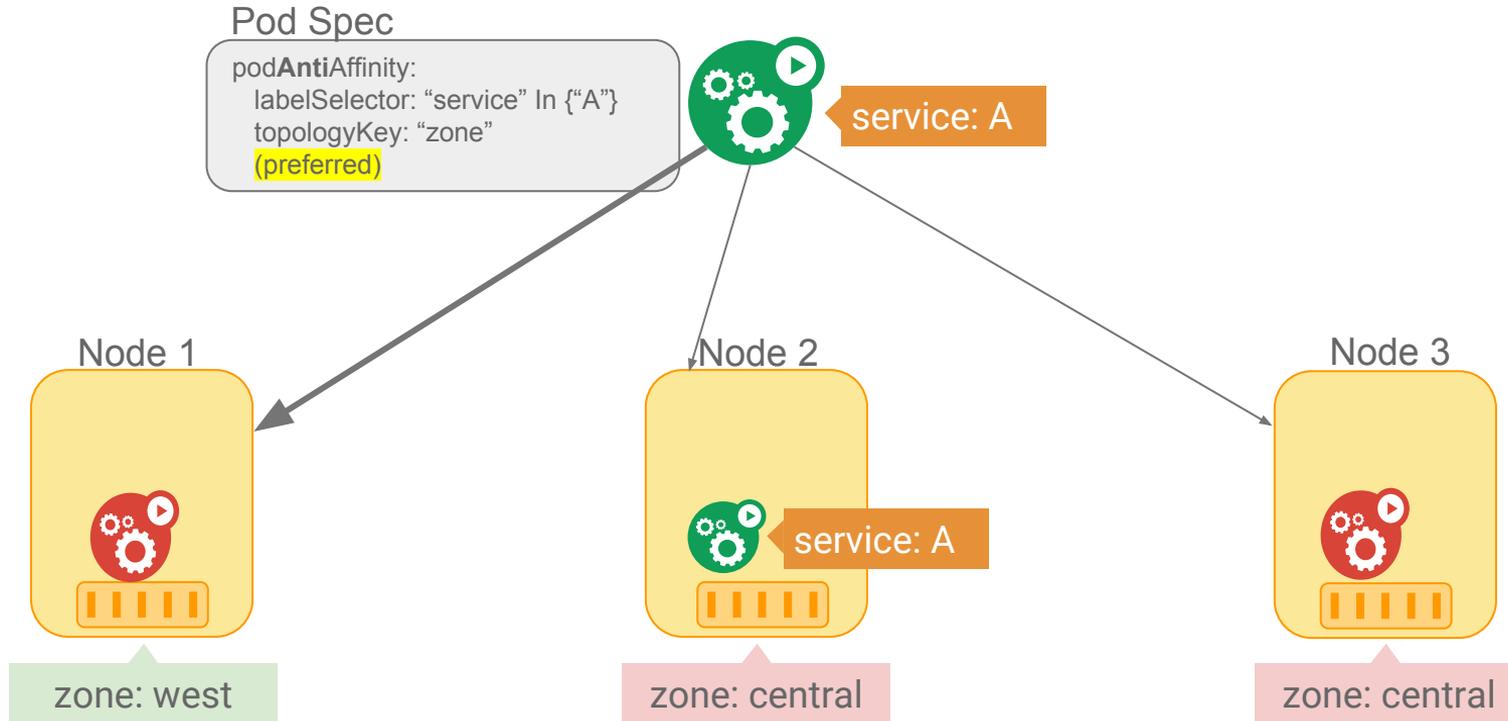


Spread Pods of a service to different Nodes



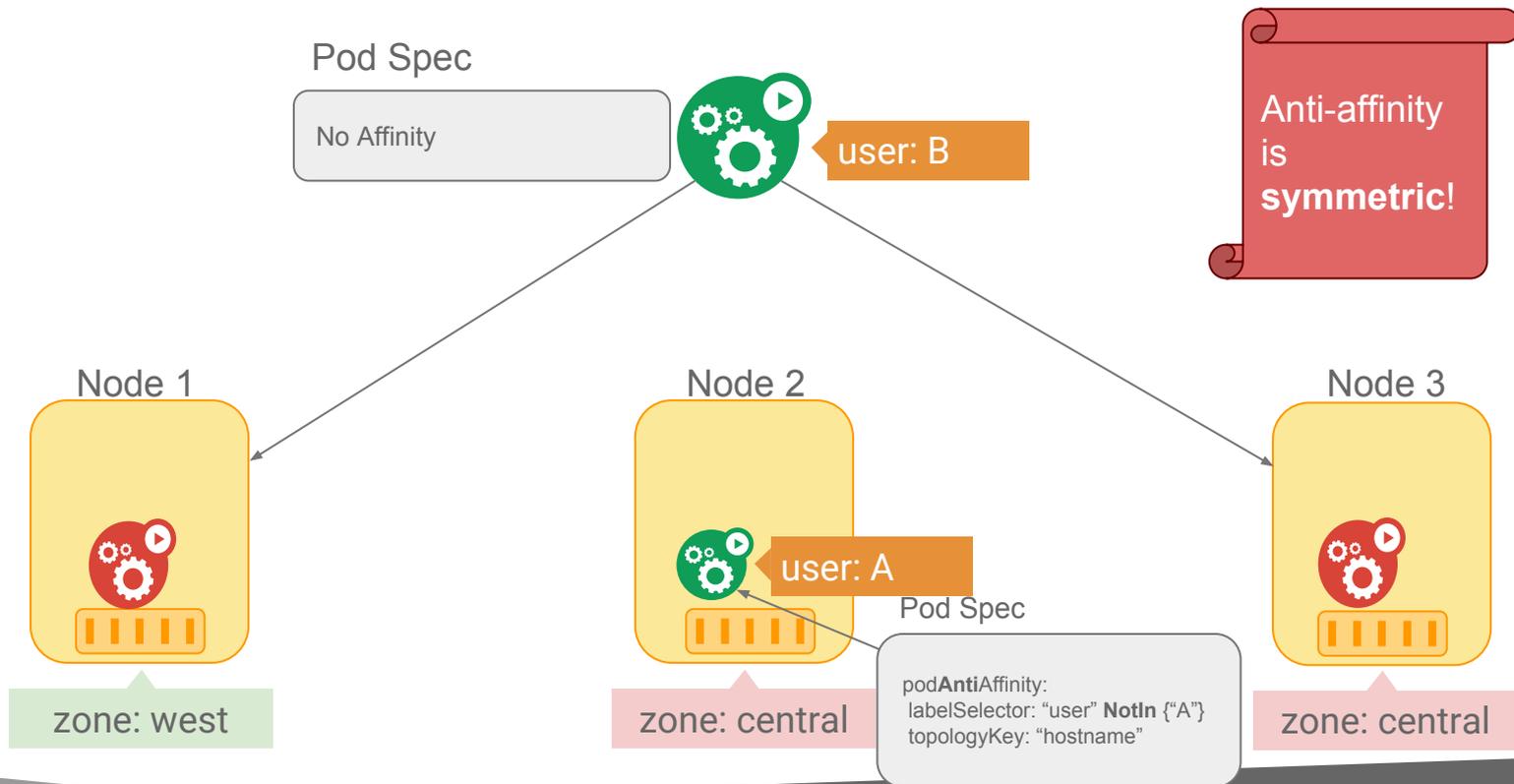


Spread Pods of a service to different Nodes



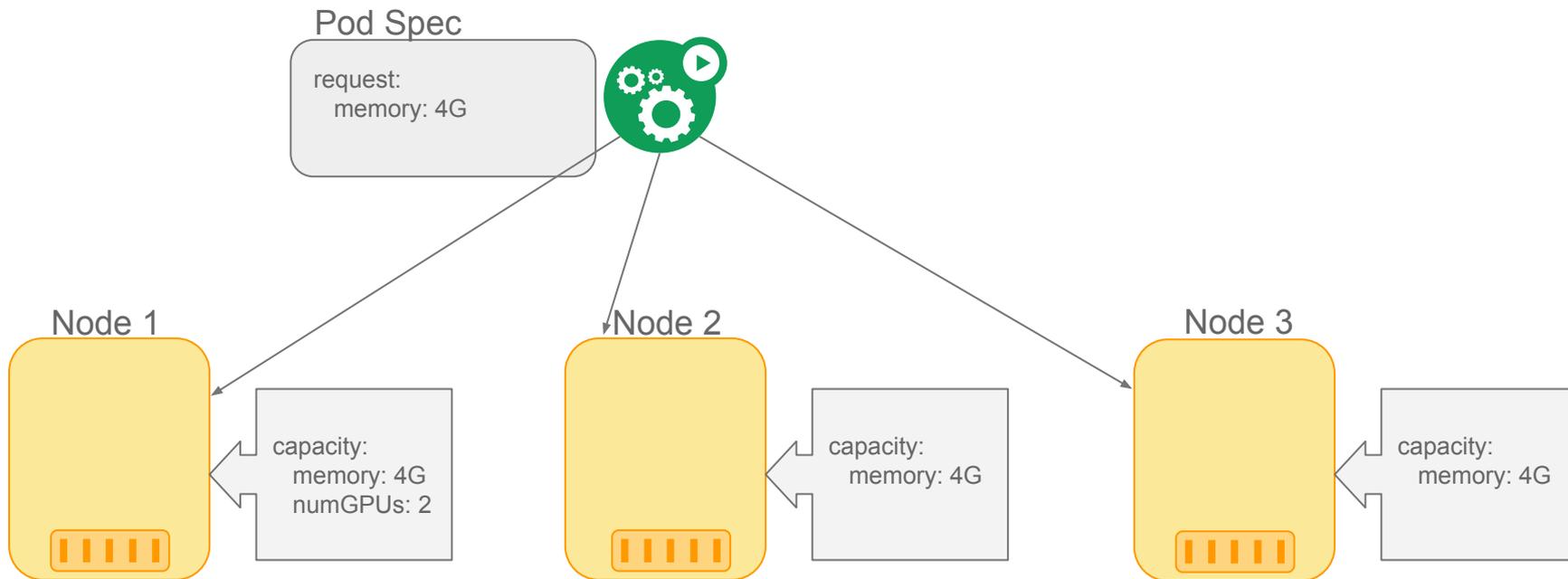


Sole tenancy





Avoid placing Pods on Nodes with special hardware





Avoid placing Pods on Nodes with special hardware

Pod Spec

```
request:  
memory: 4G  
numGPUs: 1
```



Pod is not schedulable

Node 1



```
capacity:  
memory: 4G  
numGPUs: 2
```

Node 2



```
capacity:  
memory: 4G
```

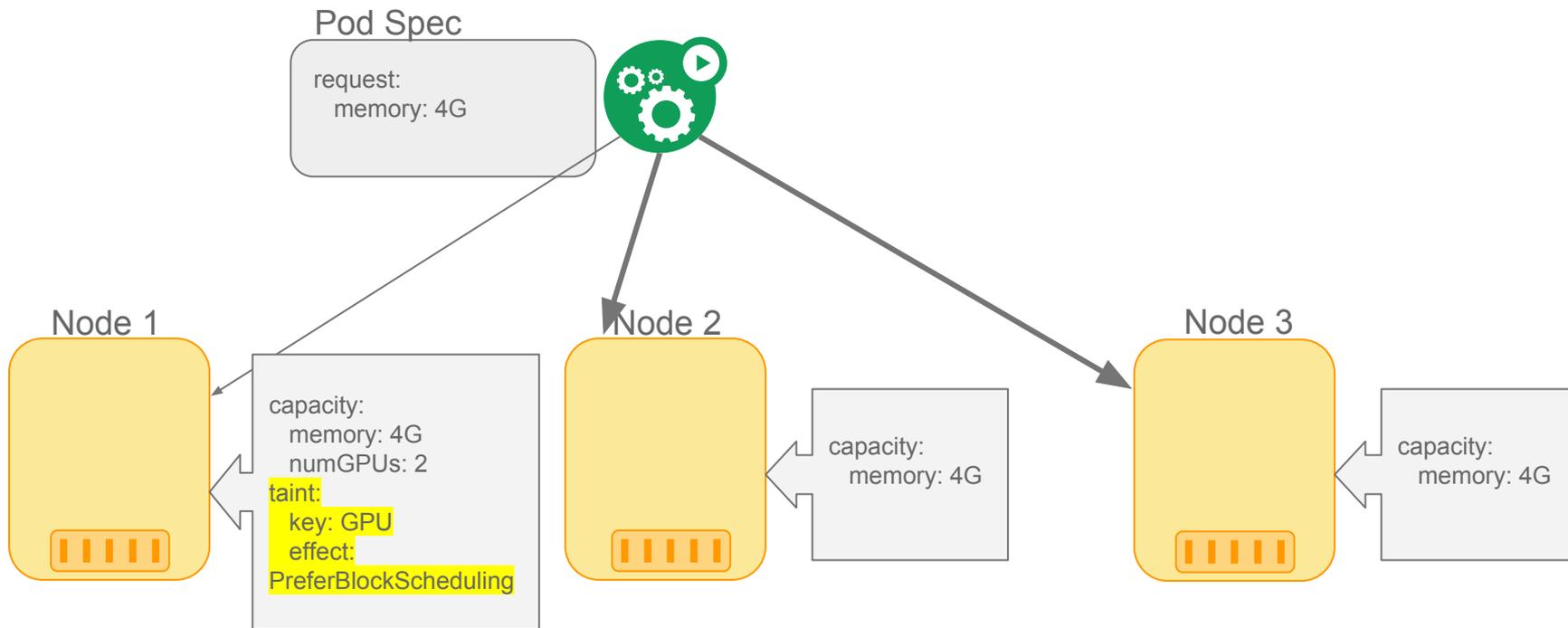
Node 3



```
capacity:  
memory: 4G
```

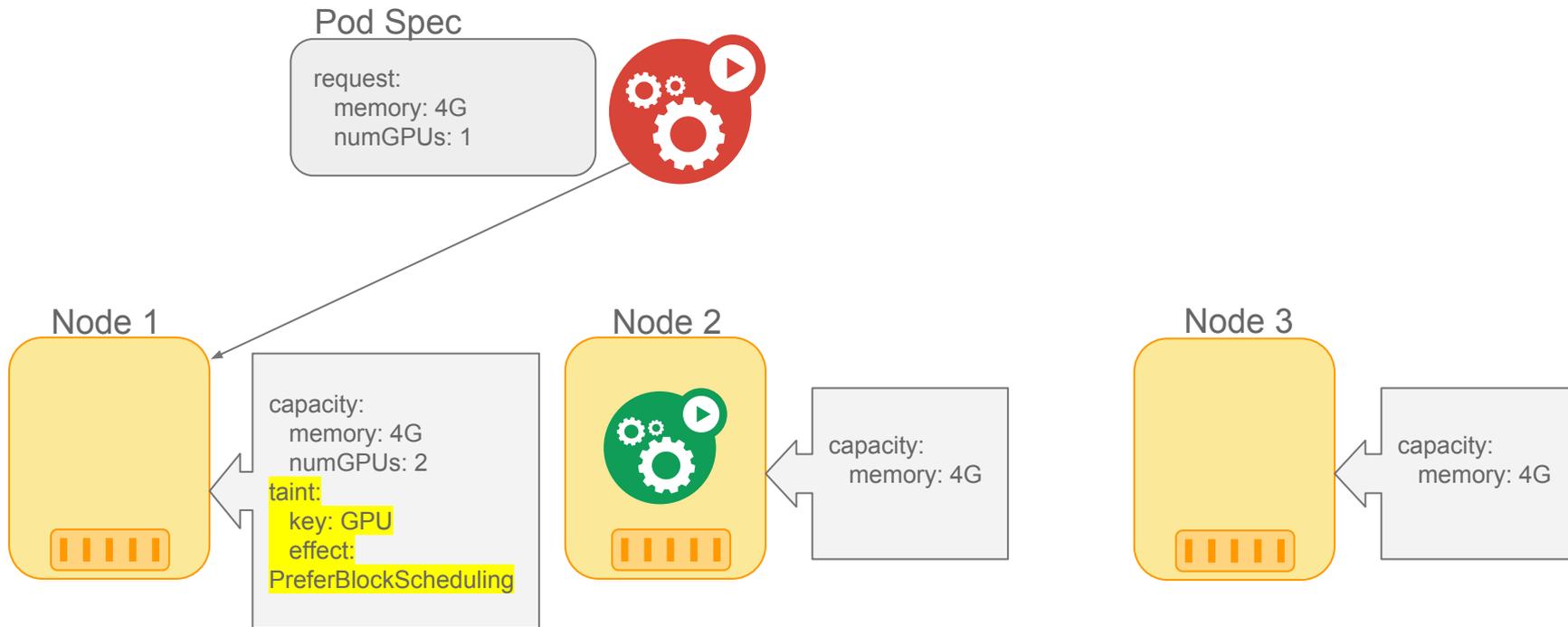


Avoid placing Pods on Nodes with special hardware



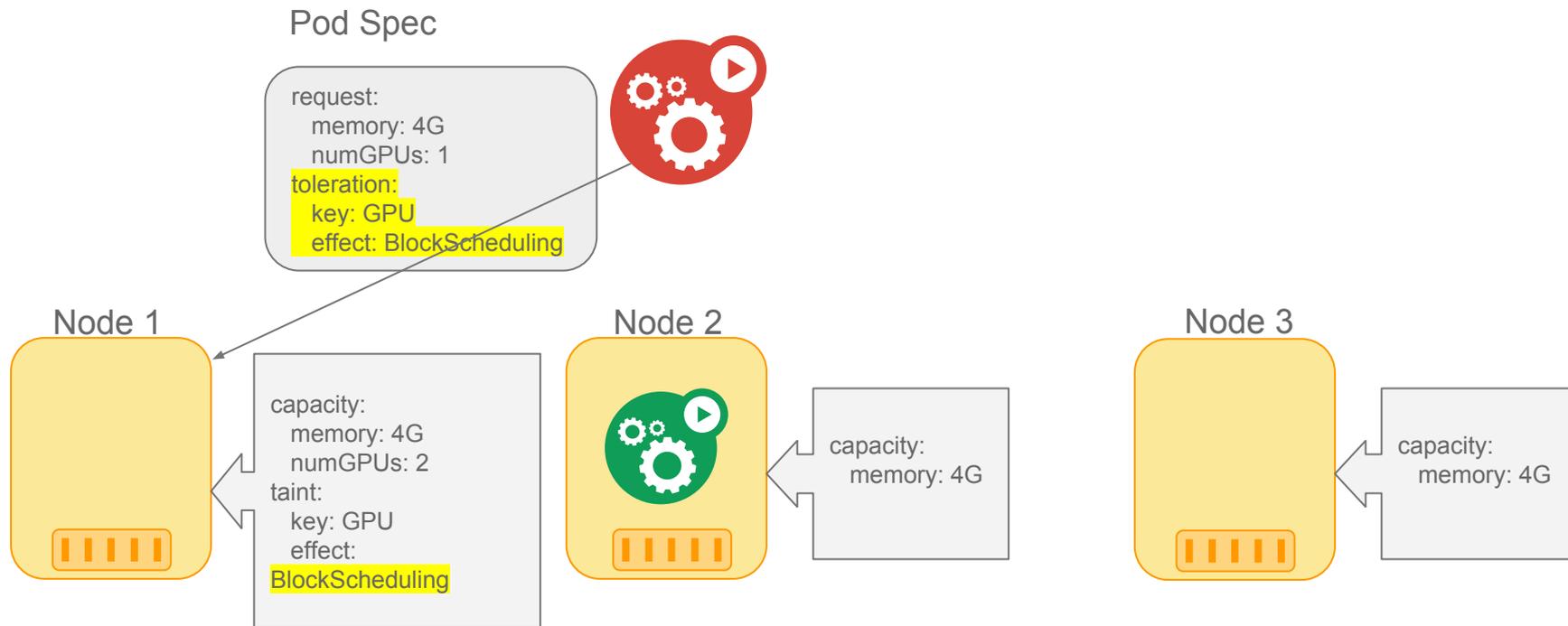


Avoid placing Pods on Nodes with special hardware



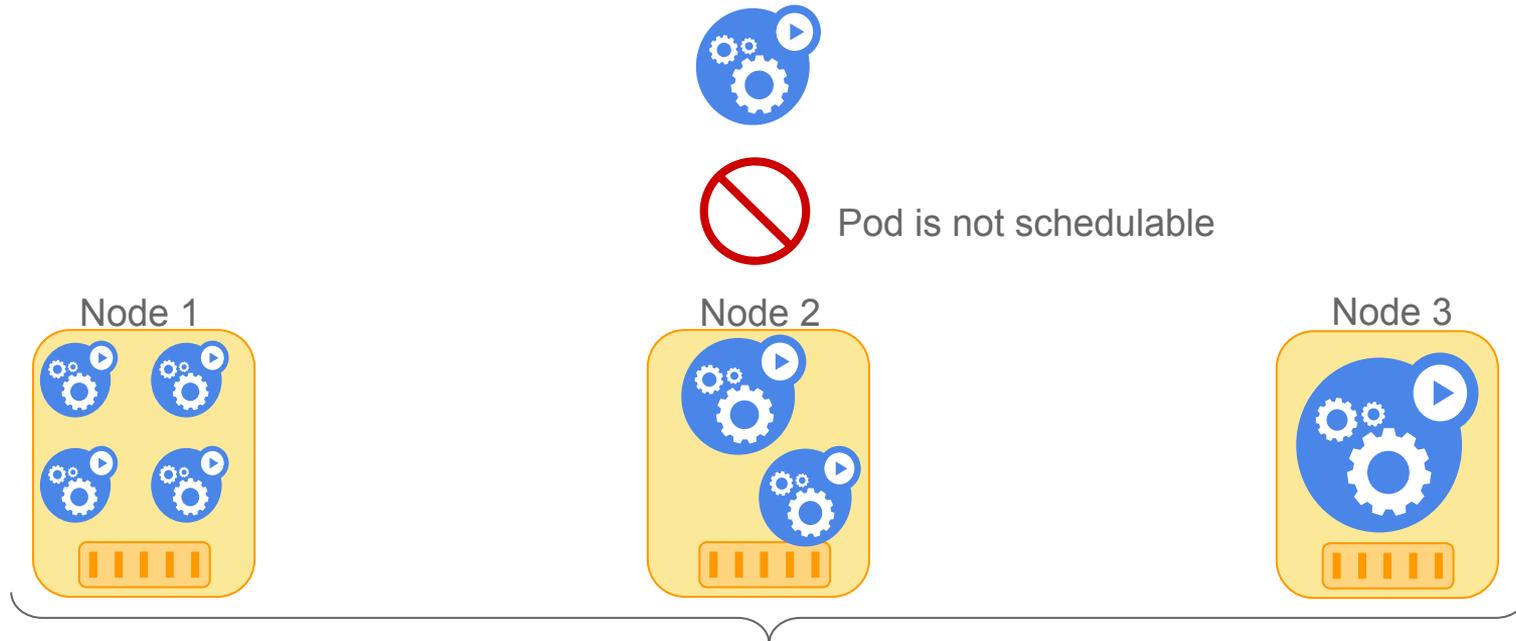


Avoid placing Pods on Nodes with special hardware





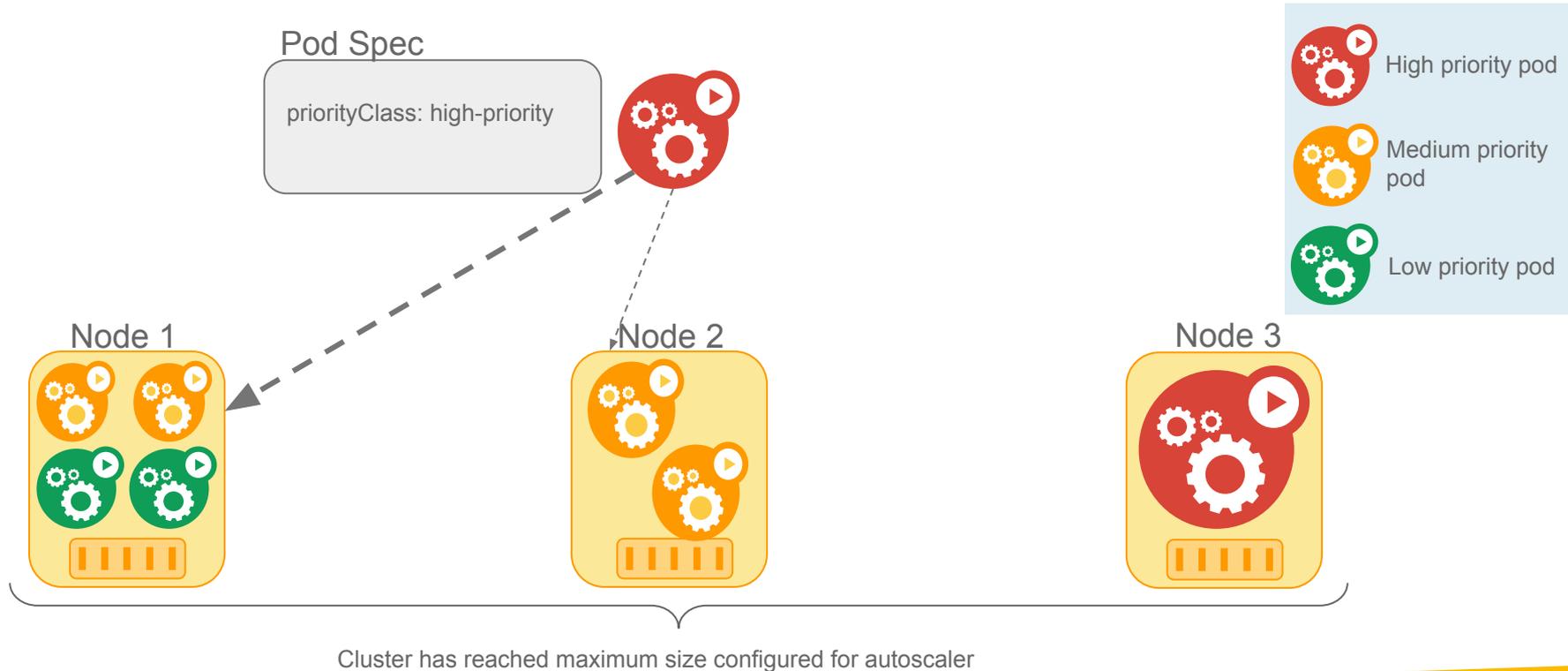
Save money by running multiple types of workloads



Cluster has reached maximum size configured for autoscaler

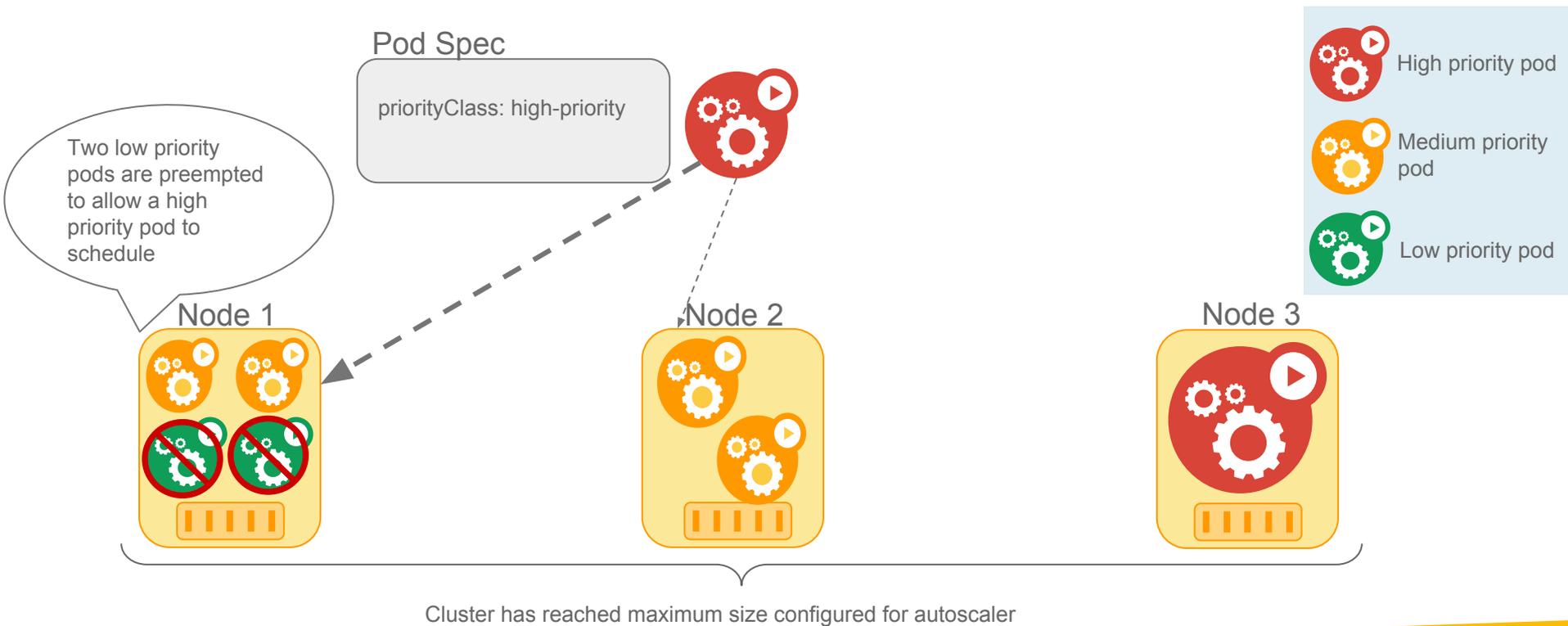


Save money by running multiple types of workloads



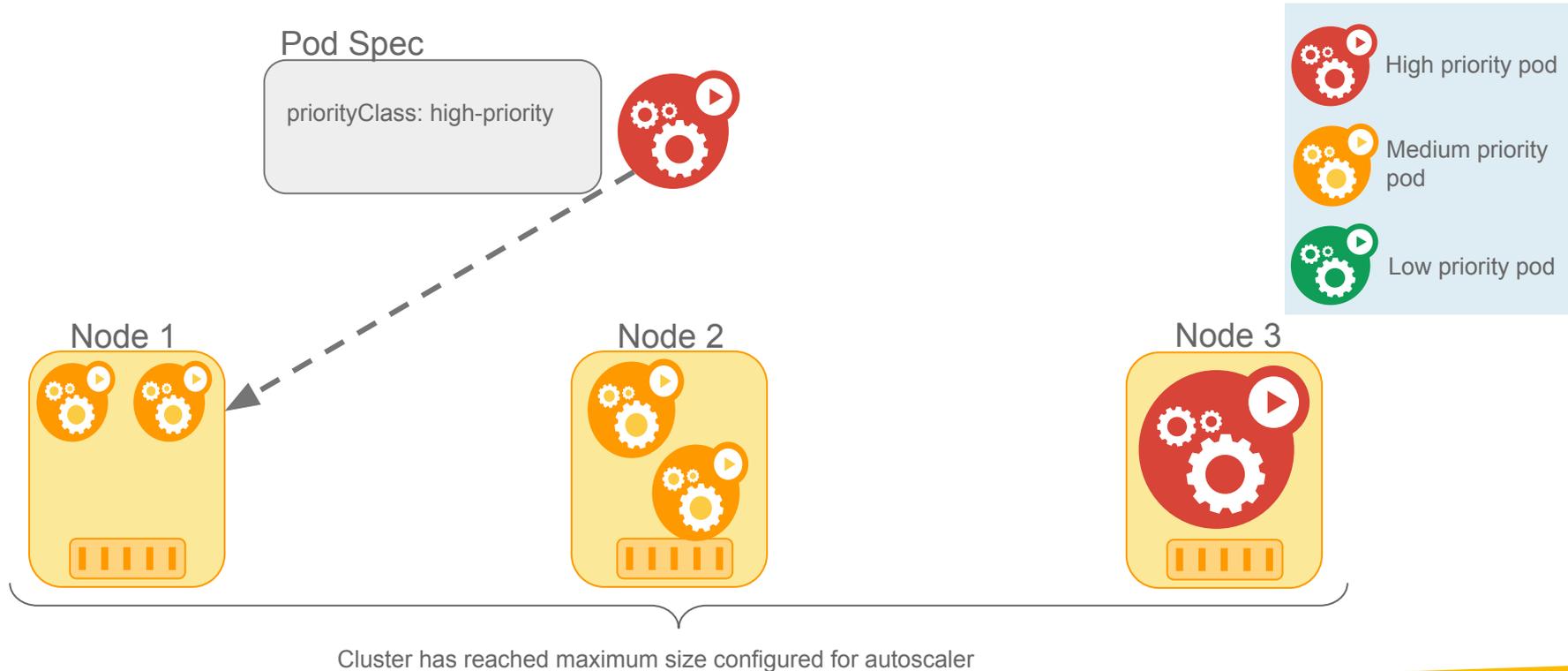


Save money by running multiple types of workloads





Save money by running multiple types of workloads

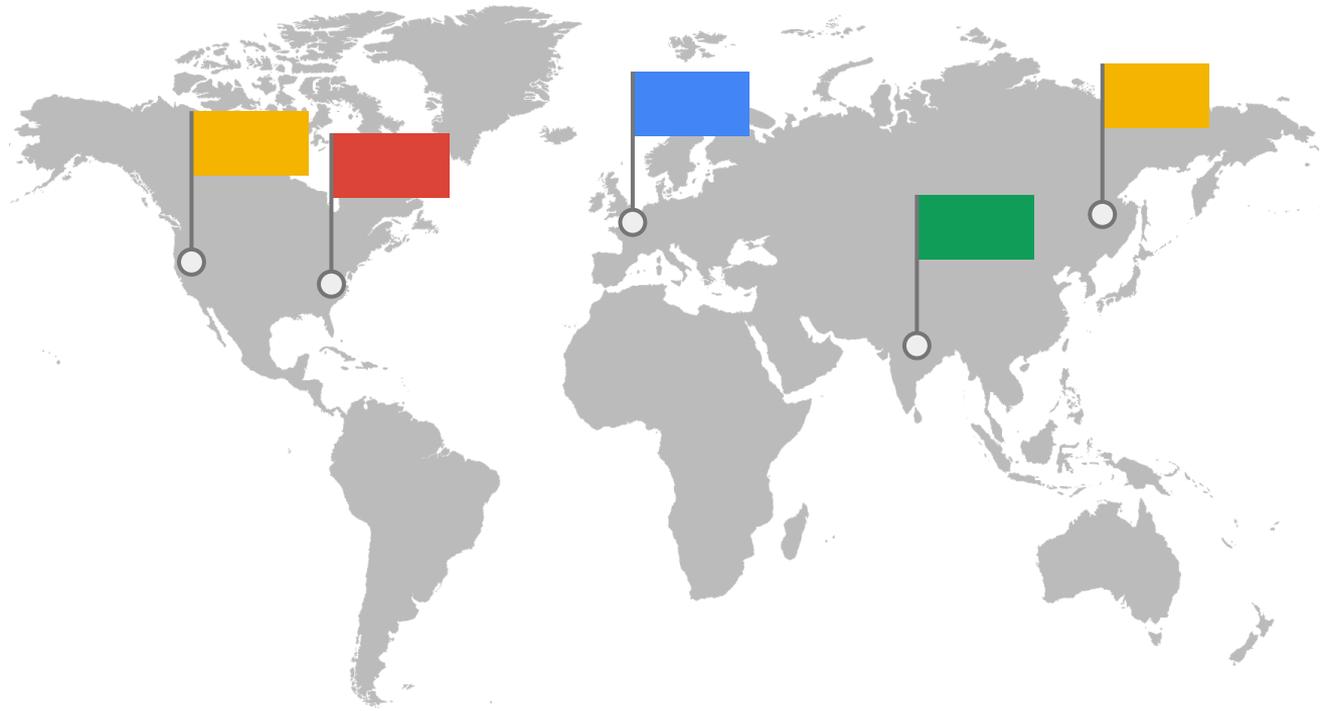


SIG Scheduling Roadmap

Roadmap

		1.9	1.10	1.11	1.12	1.13
1	Priority and Preemption	α	α	β	✓	✓
2	Gang Scheduling				α	$\alpha \beta$
3	Equivalence Cache, Affinity/Anti-affinity, Taint node by cond.	α	α	β	$\beta \checkmark$	✓
4	Scheduling Framework				α	α

Thanks a lot to our contributors!



A scenic view of the Golden Gate Bridge in San Francisco, California. The bridge's red-orange towers and suspension cables are visible on the left side of the frame. The bridge spans across a vast expanse of white fog that fills the middle ground, obscuring the water and the city skyline in the distance. The city skyline, including the Transamerica Pyramid, is visible on the horizon. The foreground shows a rocky, brownish hillside with some green trees and a small beach area on the left. The sky is a pale, overcast blue.

Scheduler,

1. Assigns pods to nodes
2. Solves complex deployment patterns
3. Is under active development



Useful links

[SIG Scheduling Community Page](#)

[Assigning Pods to Nodes](#)

[Taints and Tolerations](#)

[Pod Priority and Preemption](#)



Backup slides



Run a minimum number of instances

```
podDisruptionBudget:  
  minAvailable: 2  
  Selector: "service" In {"etcd"}
```



PDB is respected in:
Node eviction
Pod preemption
Node upgrades
...

Node 1



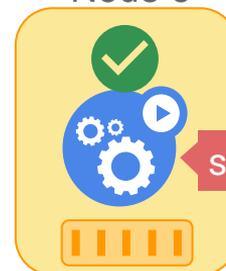
service: etcd

Node 2



service: etcd

Node 3



service: etcd

Assume phase updates scheduler cache

