Pod Scheduling

CLARUSWAY©
WAY TO REINVENT YOURSELF

Table of Contents

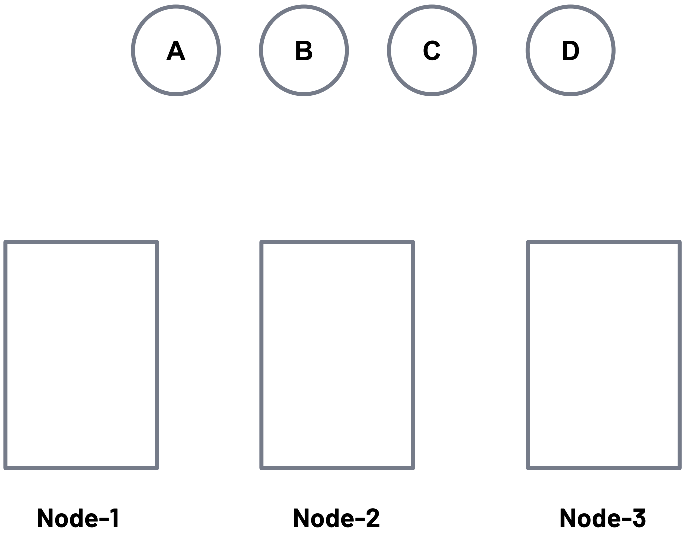
- Taints and Tolerations
- nodeSelector and Node Affinity

CLARUSWAY©
WAY TO REINVENT YOURSELF

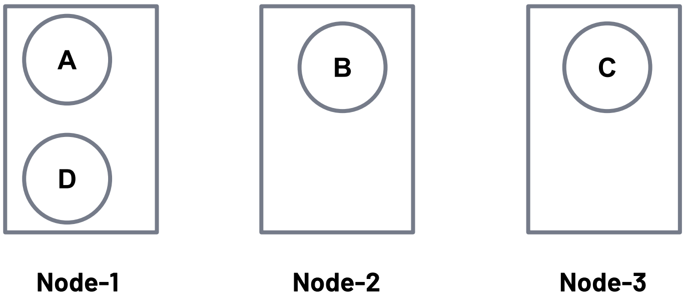
1

Taints and Tolerations

Taints and Tolerations

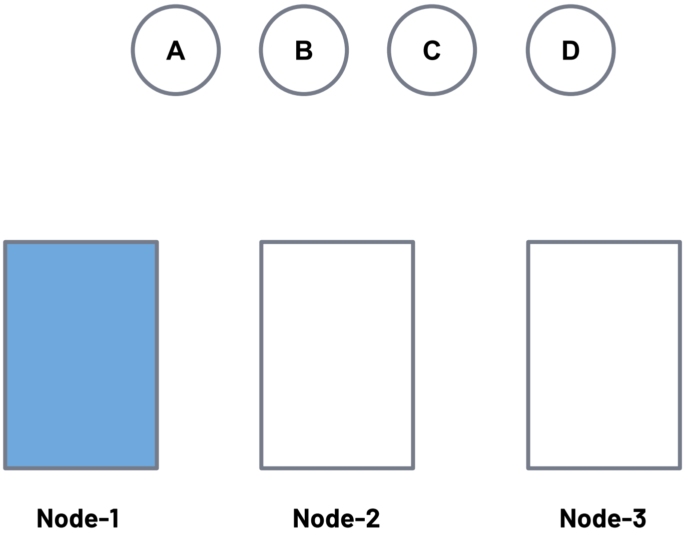


Taints and Tolerations



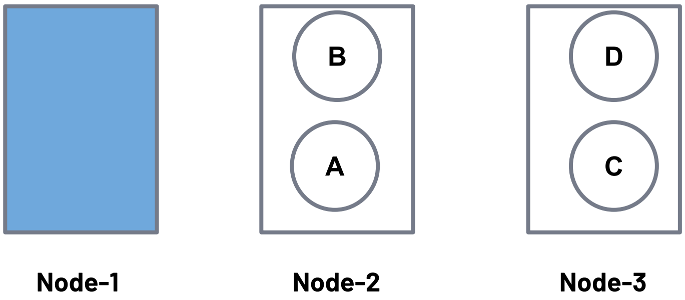
CLARUSWAY©
WAY TO REINVENT YOURSELF

Taints and Tolerations

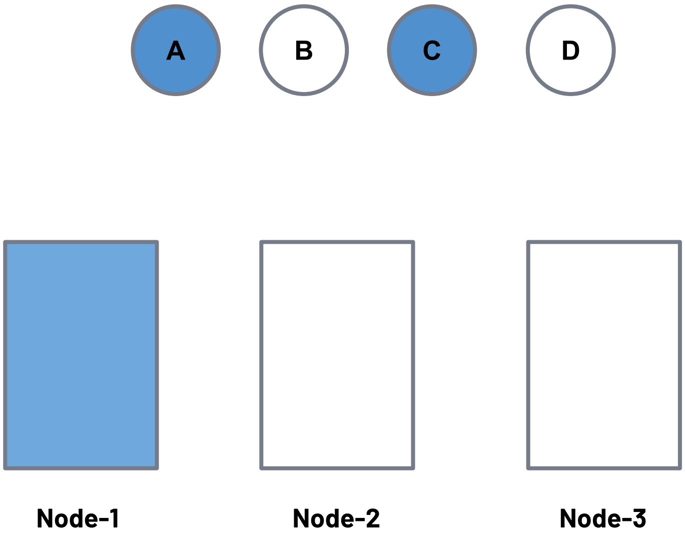


CLARUSWAY©
WAY TO REINVENT YOURSELF

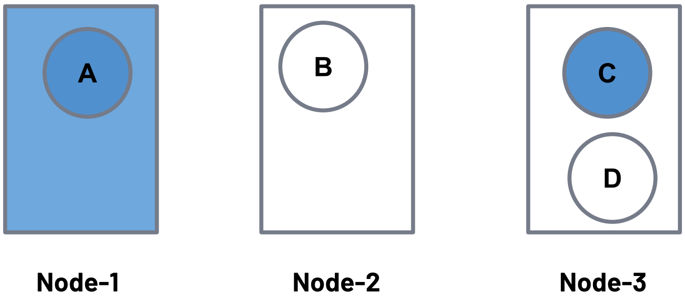
Taints and Tolerations



Taints and Tolerations



Taints and Tolerations



Taints and Tolerations

```
kubectl taint nodes node-name key=value:taint-effect
```

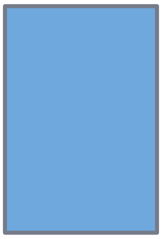
taint-effects:

- **NoExecute:**
  - Pods that do not tolerate the taint are evicted immediately
- **NoSchedule**
  - No new Pods will be scheduled on the tainted node unless they have a matching toleration. Pods currently running on the node are **not** evicted.
- **PreferNoSchedule**
  - `PreferNoSchedule` is a "preference" or "soft" version of `NoSchedule`. The control plane will *try* to avoid placing a Pod that does not tolerate the taint on the node, but it is not guaranteed.

Taints and Tolerations

- The **nodes** are **tainted**.
- **Tolerations** are added to **pods**.
- There is no rule about it; pods with tolerations will always be able to run on tainted nodes. Tolerations simply allow pods to be scheduled on tainted nodes.

```
kubectl taint nodes node-1 color=blue:NoSchedule
```

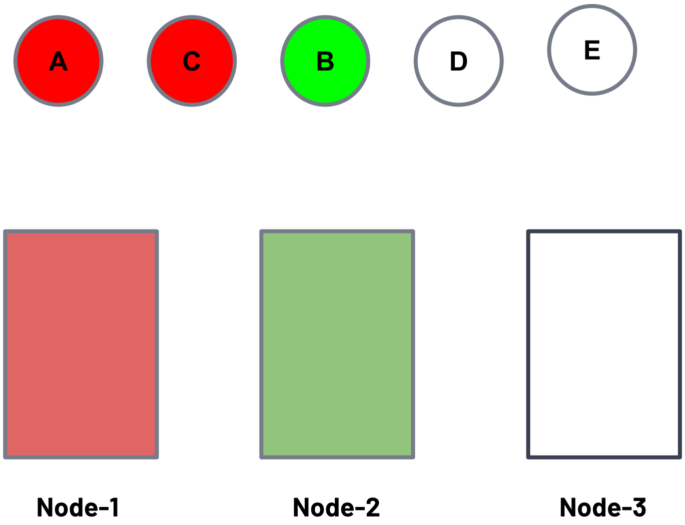


node-1

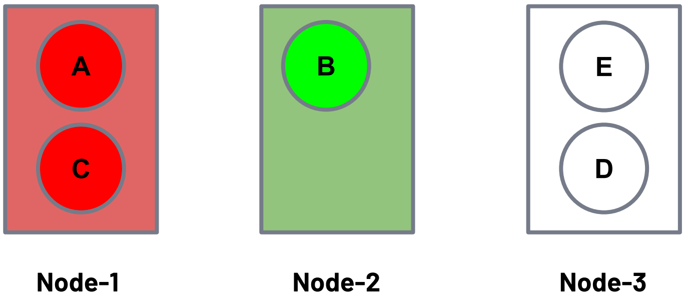
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-con
    image: nginx
  tolerations:
  - key: "color"
    operator: "Exists"
    effect: "NoSchedule"
```

2 nodeSelector and Node Affinity

nodeSelector and Node Affinity



nodeSelector and Node Affinity

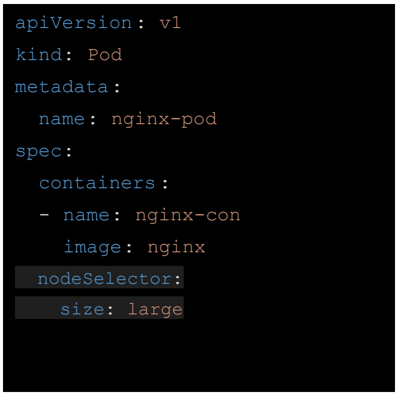


nodeSelector and Node Affinity

- The **nodes** are **labeled**.
- **nodeSelector** and **nodeAffinity** are added to pods.

```
kubectl  label node node-1 size=large
```

size: large
node-1



```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-con
    image: nginx
  nodeSelector:
    size: large
```

nodeSelector and Node Affinity

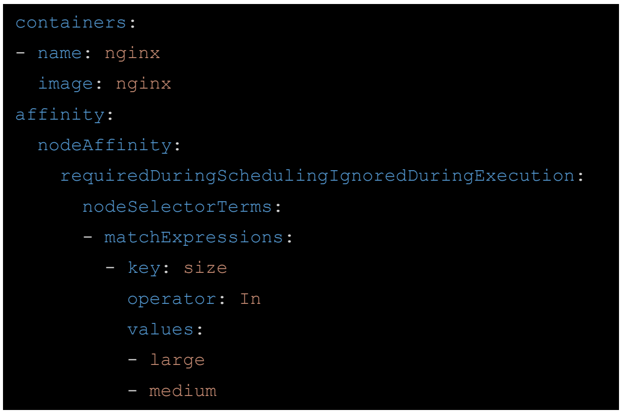
- **required**DuringSchedulingIgnoredDuringExecution:

The scheduler can't schedule the Pod unless the rule is met. This functions like nodeSelector, but with a more expressive syntax.

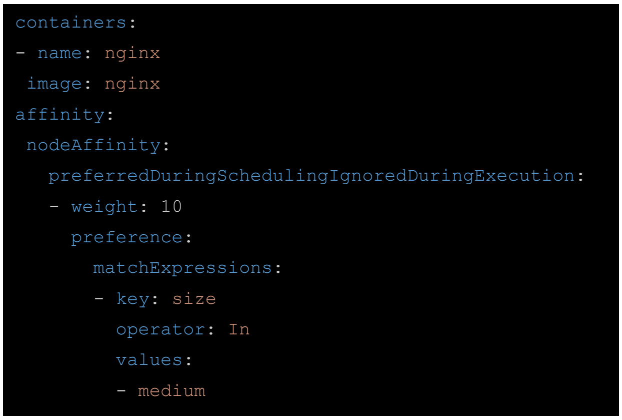
- **preferred**DuringSchedulingIgnoredDuringExecution:

The scheduler tries to find a node that meets the rule. If a matching node is not available, the scheduler still schedules the Pod.

nodeSelector and Node Affinity



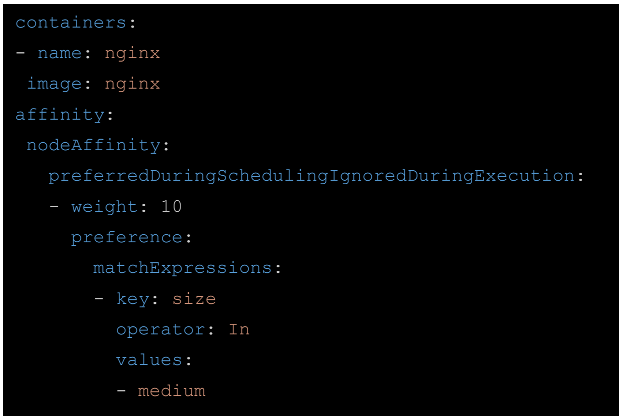
```
containers:
- name: nginx
  image: nginx
affinity:
  nodeAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
      nodeSelectorTerms:
      - matchExpressions:
        - key: size
          operator: In
          values:
          - large
          - medium
```



```
containers:
- name: nginx
 image: nginx
affinity:
 nodeAffinity:
   preferredDuringSchedulingIgnoredDuringExecution:
   - weight: 10
     preference:
       matchExpressions:
       - key: size
         operator: In
         values:
         - medium
```

nodeSelector and Node Affinity

- You can specify a **weight** between 1 and 100 for each instance of the **preferredDuringSchedulingIgnoredDuringExecution** affinity type.
- When the scheduler finds nodes that meet all the other scheduling requirements of the Pod, the scheduler iterates through every preferred rule that the node satisfies and adds the value of the weight for that expression to a sum.
- The final sum is added to the score of other priority functions for the node. Nodes with the highest total score are prioritized when the scheduler makes a scheduling decision for the Pod.

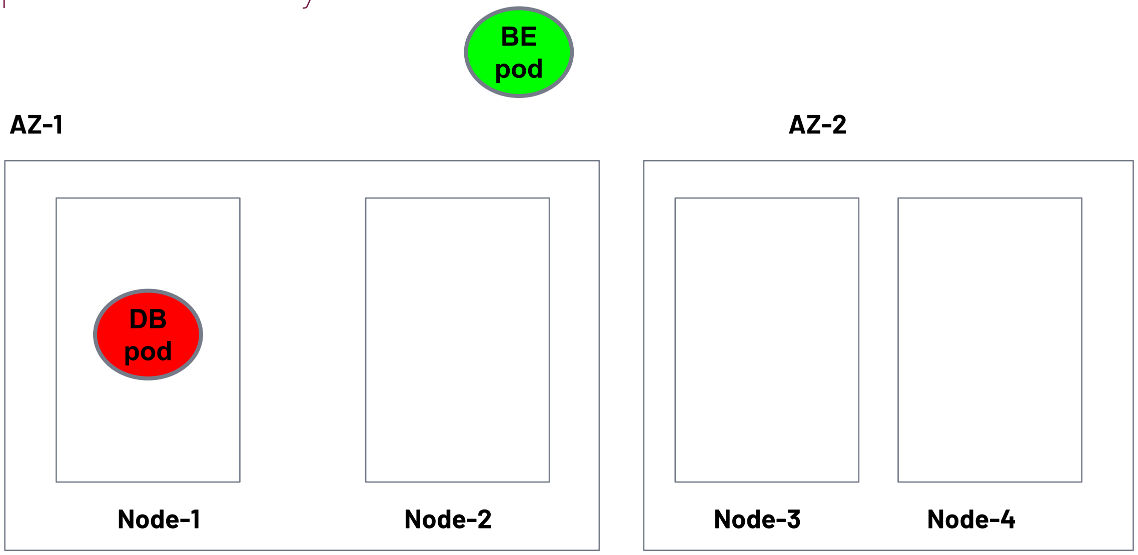


```
containers:
- name: nginx
 image: nginx
affinity:
 nodeAffinity:
   preferredDuringSchedulingIgnoredDuringExecution:
   - weight: 10
     preference:
       matchExpressions:
       - key: size
         operator: In
         values:
         - medium
```

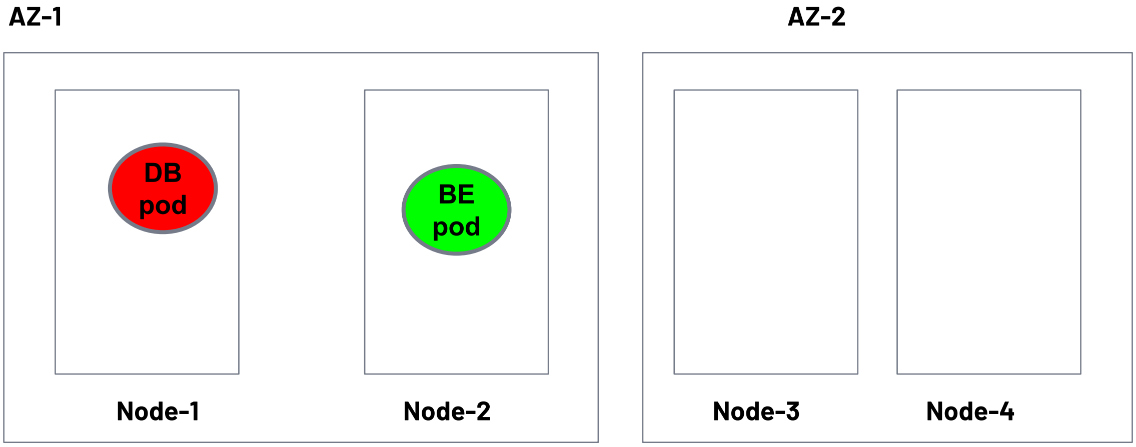
3

podAffinity

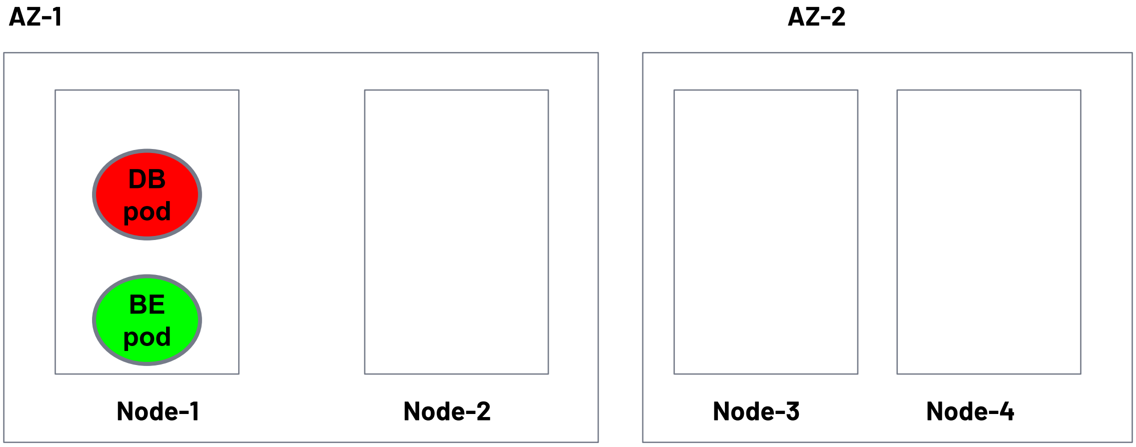
podAffinity



podAffinity



pod Affinity



pod Affinity

```
containers:
- name: clarusshop
  image: clarusway/clarusshop
  ports:
  - containerPort: 80
affinity:
  podAffinity:
    requiredDuringSchedulingIgnoredDuringEx
    - labelSelector:
        matchExpressions:
        - key: tier
          operator: In
          values:
          - db
      topologyKey: "kubernetes.io/hostname"
```

topologyKeys:

- kubernetes.io/hostname
- topology.kubernetes.io/zone
- topology.kubernetes.io/region

THANKS!

Any questions?