

ondia

Pod Scheduling



ondia

AGENDA

- ▶ Taints and Tolerations
- ▶ nodeSelector and Node Affinity

1

Taints and Tolerations

Taints and Tolerations



Node-1

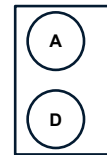


Node-2

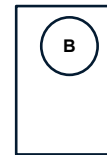


Node-3

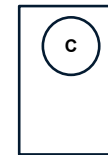
Taints and Tolerations



Node-1

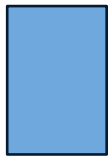


Node-2



Node-3

Taints and Tolerations



Node-1



Node-2

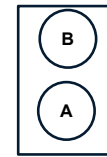


Node-3

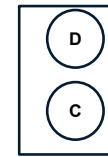
Taints and Tolerations



Node-1

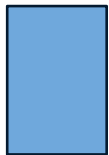


Node-2



Node-3

Taints and Tolerations



Node-1

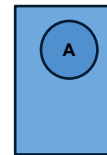


Node-2

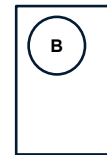


Node-3

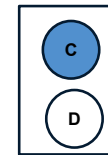
Taints and Tolerations



Node-1



Node-2



Node-3

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

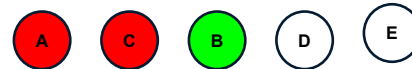
```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
spec:
  containers:
  - name: nginx-con
    image: nginx
  tolerations:
  - key: "color"
    operator: "Exists"
    effect: "NoSchedule"
```

2

nodeSelector and Node Affinity



nodeSelector and Node Affinity



Node-1



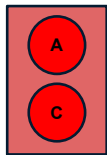
Node-2



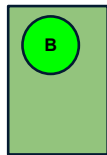
Node-3



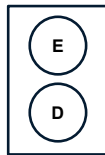
nodeSelector and Node Affinity



Node-1



Node-2



Node-3

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

- **requiredDuringSchedulingIgnoredDuringExecution:**

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

- **preferredDuringSchedulingIgnoredDuringExecution:**

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
```

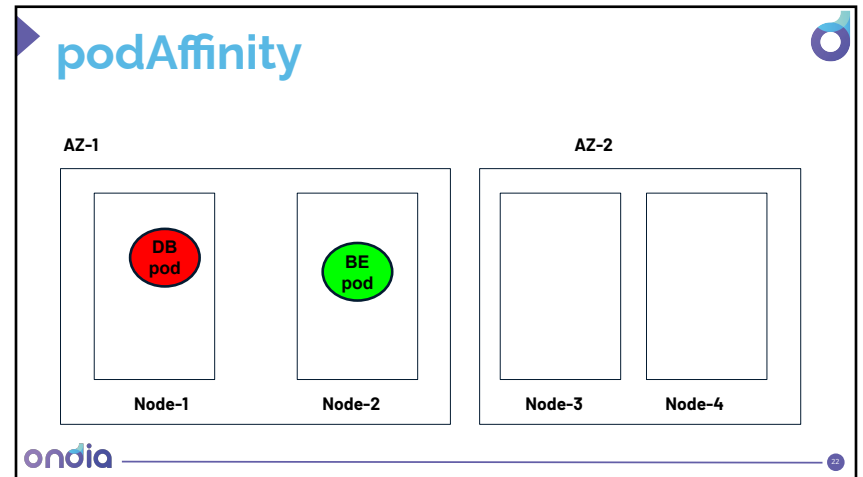
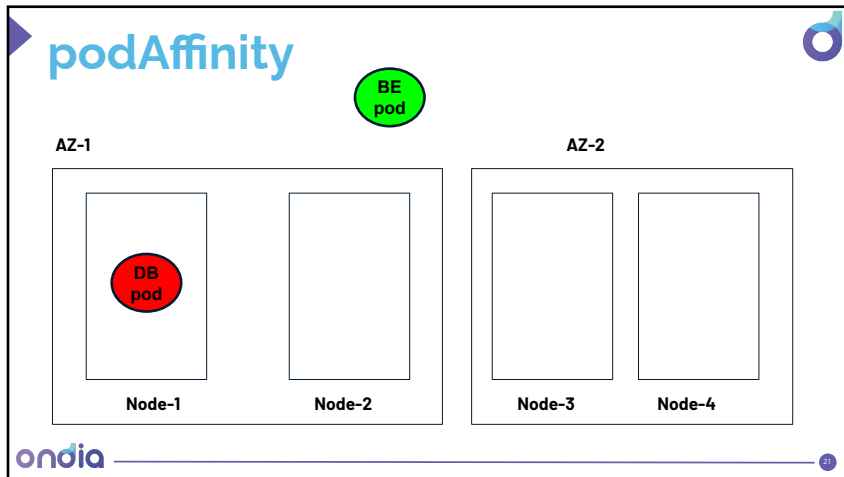
```
containers:
- name: nginx
  image: nginx
affinity:
  nodeAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 10
      preference:
        matchExpressions:
        - key: size
          operator: In
          values:
            - medium
```

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.

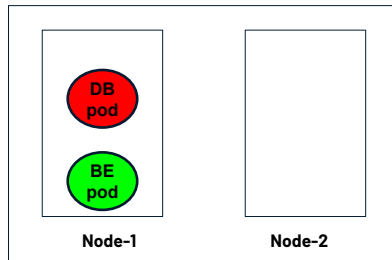
```
containers:
- name: nginx
  image: nginx
affinity:
  nodeAffinity:
    preferredDuringSchedulingIgnoredDuringExecution:
    - weight: 10
      preference:
        matchExpressions:
        - key: size
          operator: In
          values:
          - medium
```

3 podAffinity

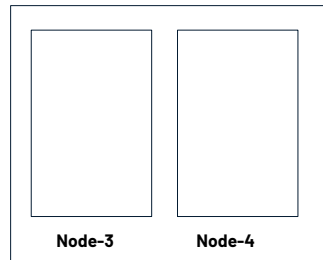


podAffinity

AZ-1



AZ-2



podAffinity

```
containers:
- name: clarussshop
  image: clarusway/clarussshop
  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

ondia



THANKS!
Any questions?



Powered by
Clarusway
Way to Reinvent yourself

Assigning Pods to Nodes

nodeName

nodeAffinity

podAffinity

podAntiAffinity

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
  nodeName: kube-01
```

nodeSelector

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx
  nodeSelector:
    disktype: ssd
```

```
apiVersion: v1
kind: Pod
metadata:
  name: with-node-affinity
spec:
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
        - matchExpressions:
            - key: topology.kubernetes.io/zone
              operator: In
              values:
                - antarctica-east1
                - antarctica-west1
      preferredDuringSchedulingIgnoredDuringExecution:
        - weight: 1
          preference:
            - key: another-node-label-key
              operator: In
              values:
                - another-node-label-value
  containers:
  - name: with-node-affinity
    image: registry.k8s.io/pause:2.8
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-server
spec:
  selector:
    matchLabels:
      app: web-store
  replicas: 3
  template:
    metadata:
      labels:
        app: web-store
    spec:
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - web-store
              topologyKey: "kubernetes.io/hostname"
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - store
              topologyKey: "kubernetes.io/hostname"
      containers:
      - name: web-app
        image: nginx:1.16-alpine
```

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-cache
spec:
  selector:
    matchLabels:
      app: store
  replicas: 3
  template:
    metadata:
      labels:
        app: store
    spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - store
              topologyKey: "kubernetes.io/hostname"
      containers:
      - name: redis-server
        image: redis:3.2-alpine
```

CLARUSWAY®
WAY TO REINVENT YOURSELF

BR 12.1

Assigning Pods to Nodes

podAffinity

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: web-server
spec:
  selector:
    matchLabels:
      app: web-store
  replicas: 3
  template:
    metadata:
      labels:
        app: web-store
    spec:
      affinity:
        podAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - web-store
              topologyKey: "kubernetes.io/hostname"
      podAffinity:
        requiredDuringSchedulingIgnoredDuringExecution:
          - labelSelector:
              matchExpressions:
                - key: app
                  operator: In
                  values:
                    - store
              topologyKey: "kubernetes.io/hostname"
      containers:
        - name: web-app
          image: nginx:1.16-alpine
```

podAntiAffinity

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: redis-cache
spec:
  selector:
    matchLabels:
      app: store
  replicas: 3
  template:
    metadata:
      labels:
        app: store
    spec:
      affinity:
        podAntiAffinity:
          requiredDuringSchedulingIgnoredDuringExecution:
            - labelSelector:
                matchExpressions:
                  - key: app
                    operator: In
                    values:
                      - store
              topologyKey: "kubernetes.io/hostname"
      containers:
        - name: redis-server
          image: redis:3.2-alpine
```

node-1	node-2	node-3
webserver-1	webserver-2	webserver-3
cache-1	cache-2	cache-3

Assigning Pods to Nodes

tolerations:

- key: "gpu"
- operator: "Equal"
- value: "true"
- effect: "NoSchedule"



Pod 1



Pod 2

tolerations:

- key: "ubuntu"
- operator: "Equal"
- value: "true"
- effect: "NoSchedule"



Pod Repelled



Worker Node A

gpu=true:NoSchedule

Task

- Nginx Deployment with 3 Pods
- Service with internet connection (port: 30033)
- Create a configMap mounts “/special” folder and files under it in Pods.

podAffinity

