Node Affinity in Kubernetes

Node Affinity in K8s

Node Affinity

➔ **What is NodeAffinity**

➔ **Example of Node Affinity**

➔ **Hands On Demonstration**

Node Affinity

➢ Node Affinity is enhanced version of NodeSelector.

➢ Node Affinity is used for Pods Allocation on Worker Nodes.

➢ Not to Schedule Pod on Nodes is achieve via Node Anti-Affinity.

➢ Anti-Affinity is Opposite of Affinity and NodeSelector Concept.

Node Affinity

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

```yaml
...
spec:
  containers:
    - name: nginx
      image: nginx
  affinity:
    nodeAffinity:
      requiredDuringSchedulingIgnoredDuringExecution:
        nodeSelectorTerms:
          - matchExpressions:
              - key: disktype
                operator: In
                values:
                  - ssd
```

Node Affinity

➢ requiredDuringSchedulingIgnoredDuringExecution :

Must fulfil the condition at the time of Pod Creation.

Also Called Hard Affinity.

IgnoredDuringExecution - Pod will still run if labels on a node change and affinity rules are no longer met.

Node Affinity

➢ preferredDuringSchedulingIgnoredDuringExecution :

Prefer Node which will fulfil the condition but will not guarantee.

Also Called Soft Affinity.

➢ requiredDuringSchedulingRequiredDuringExecution : Not added but will be available in future.

Node Anti-Affinity

```
...
spec:
 containers:
   - name: nginx
     image: nginx
 affinity:
  nodeAffinity:
   requiredDuringSchedulingIgnoredDuringExecution:
    nodeSelectorTerms:
     - matchExpressions:
       - key: disktype
         operator: In
         values:
           - ssd
```

```
...
spec:
 containers:
   - name: nginx
     image: nginx
 affinity:
  nodeAffinity:
   requiredDuringSchedulingIgnoredDuringExecution:
    nodeSelectorTerms:
     - matchExpressions:
       - key: disktype
         operator: NotIn
         values:
           - ssd
```

Hands On Demonstration

Thank You...

Don't be the Same! Be Better!!!