# Pod Affinity & AntiAffinity

Pod affinity can be used for advanced scheduling in Kubernetes. Tutorial below is a walkthrough of such usage.

## Concepts

### Pod Affinity & AntiAffinity

* Node affinity allows you to schedule a pod on a set of nodes based on labels present on the nodes. However, in certain scenarios, we might want to schedule certain pods together or we might want to make sure that certain pods are never scheduled together. This can be achieved by PodAffinity and/or PodAntiAffinity respectively.
* Similar to node affinity, there are a couple of variants in pod affinity namely requiredDuringSchedulingIgnoredDuringExecution and preferredDuringSchedulingIgnoredDuringExecution.

## Use cases

* While scheduling workload, when we need to schedule a certain set of pods together, PodAffinity makes sense. Example, a web server and a cache.
* While scheduling workload, when we need to make sure that a certain set of pods are not scheduled together, PodAntiAffinity makes sense.

## Examples:

Follow through guide.

Let's begin with listing nodes.

# kubectl get nodes

You should be able to see the list of nodes available in the cluster,

NAME STATUS ROLES AGE VERSION

node1.compute.infracloud.io Ready <none> 25m v1.9.4

node2.compute.infracloud.io Ready <none> 25m v1.9.4

node3.compute.infracloud.io Ready <none> 28m v1.9.4

#### Pod Affinity example

Let's deploy [deployment-Affinity.yaml](https://github.com/infracloudio/kubernetes-scheduling-examples/blob/master/podAffinity/deployment-Affinity.yaml), which has pod affinity as,

affinity:

podAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- nginx

topologyKey: "kubernetes.io/hostname"

Here we are specifying that all nginx pods should be scheduled together.

# Kubectl apply -f deployment-Affinity.yaml

Check the pods using,

# Kubectl get pods -o wide -w

You should be able to see that all pods are scheduled on the same node.

NAME READY STATUS RESTARTS AGE IP NODE

nginx-deployment-6bc5bb7f45-49dtg 1/1 Running 0 36m 10.20.29.18 node2.compute.infracloud.io

nginx-deployment-6bc5bb7f45-4ngvr 1/1 Running 0 36m 10.20.29.20 node2.compute.infracloud.io

nginx-deployment-6bc5bb7f45-lppkn 1/1 Running 0 36m 10.20.29.19 node2.compute.infracloud.io

To clean up run,

# Kubectl delete -f deployment-Affinity.yaml

#### Pod Anti Affinity example

Let's deploy [deployment-AntiAffinity.yaml](https://github.com/infracloudio/kubernetes-scheduling-examples/blob/master/podAffinity/deployment-AntiAffinity.yaml), which has pod affinity as,

affinity:

podAntiAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- nginx

topologyKey: "kubernetes.io/hostname"

Here we are specifying that no two nginx pods should be scheduled together.

# Kubectl apply -f deployment-AntiAffinity.yaml

Check the pods using,

# Kubectl get pods -o wide -w

You should be able to see that pods are scheduled on different nodes.

NAME READY STATUS RESTARTS AGE IP NODE

nginx-deployment-85d87bccff-4w7tf 1/1 Running 0 27s 10.20.29.16 node3.compute.infracloud.io

nginx-deployment-85d87bccff-7fn47 1/1 Running 0 27s 10.20.42.32 node1.compute.infracloud.io

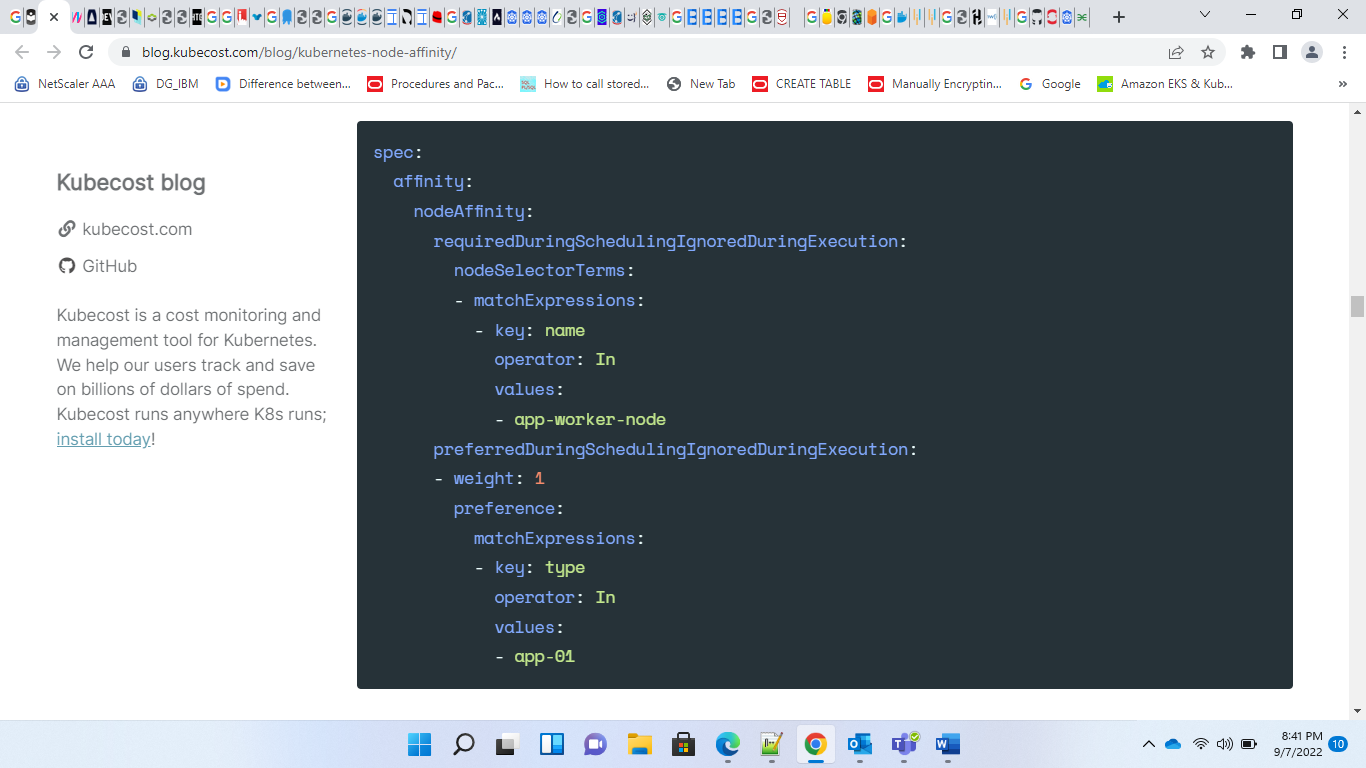
nginx-deployment-85d87bccff-sd4lp 1/1 Running 0 27s 10.20.13.17 node2.compute.infracloud.io

**Note**: In above example, if number of replicas are more than number of nodes then some pods will remain in pending state.

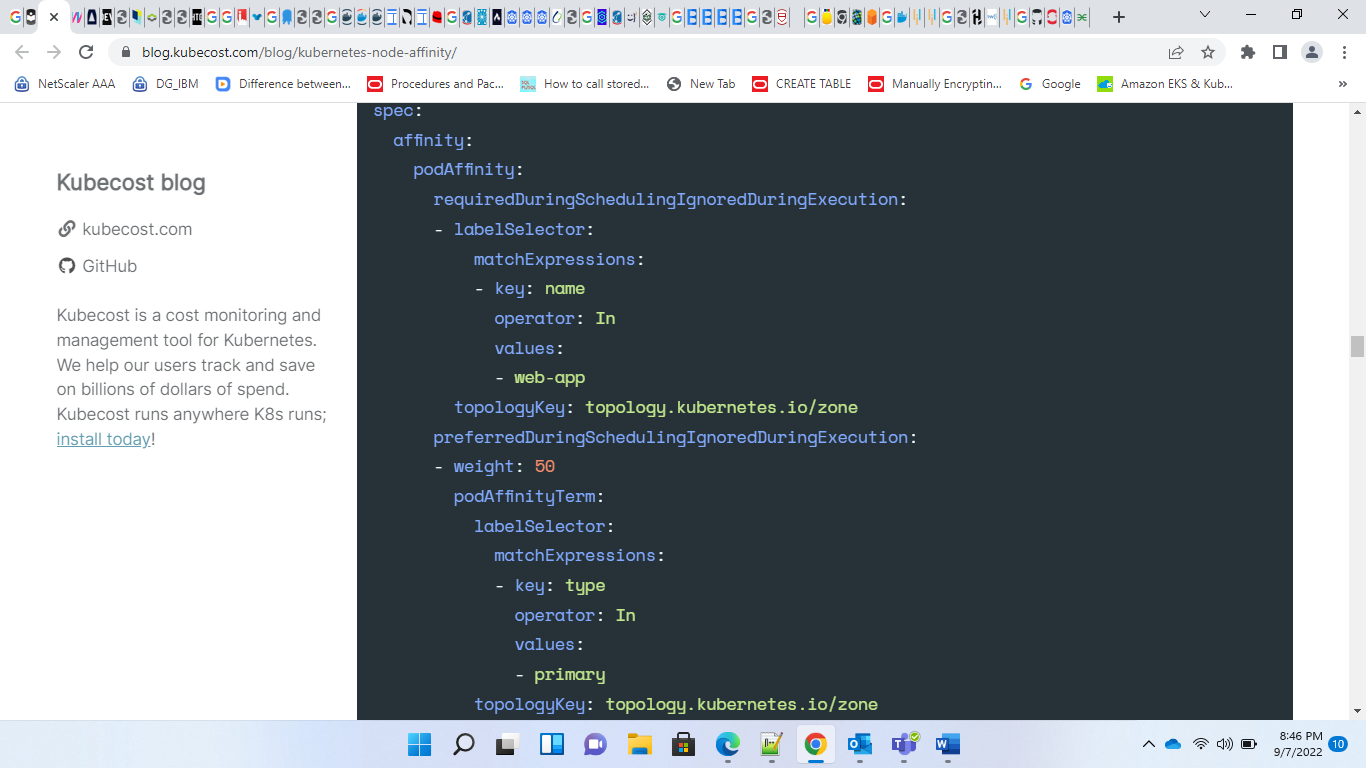
To clean up run,

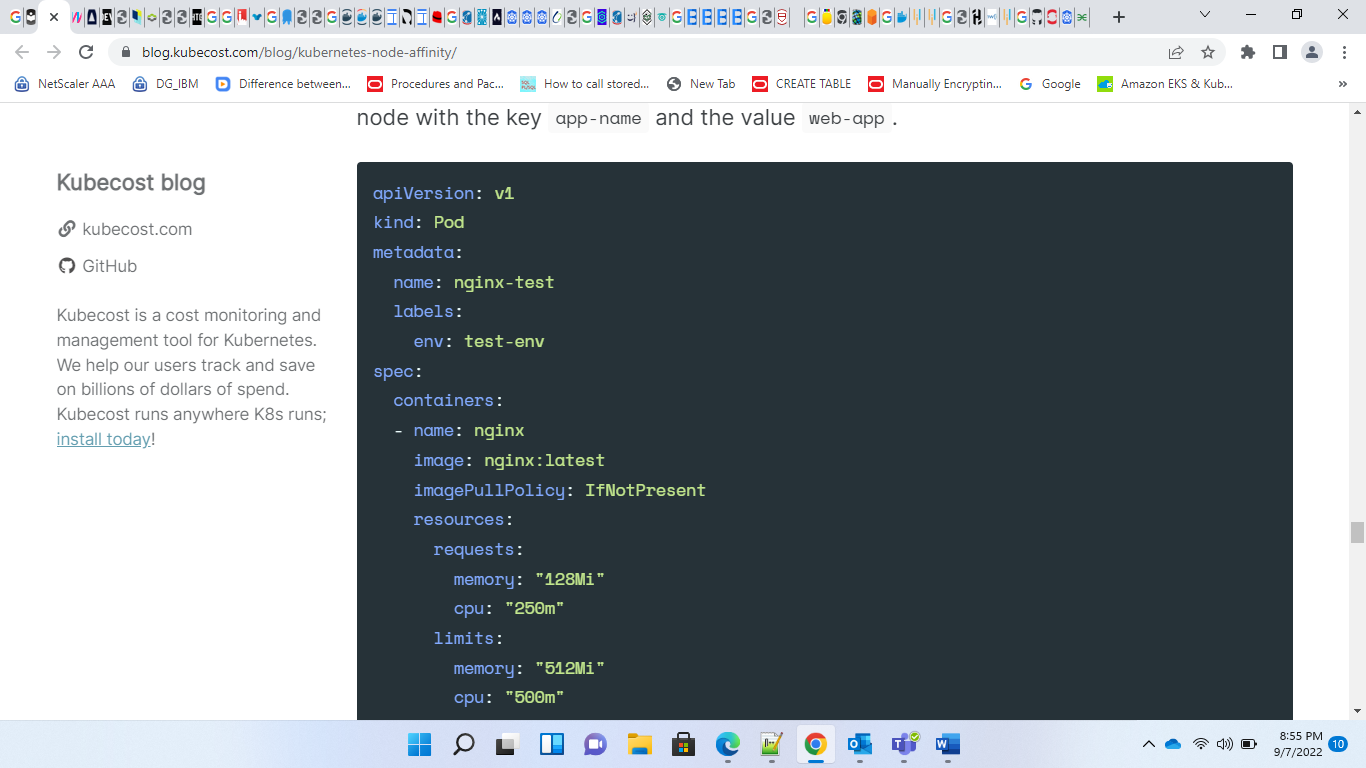
# Kubectl delete -f deployment-AntiAffinity.yaml

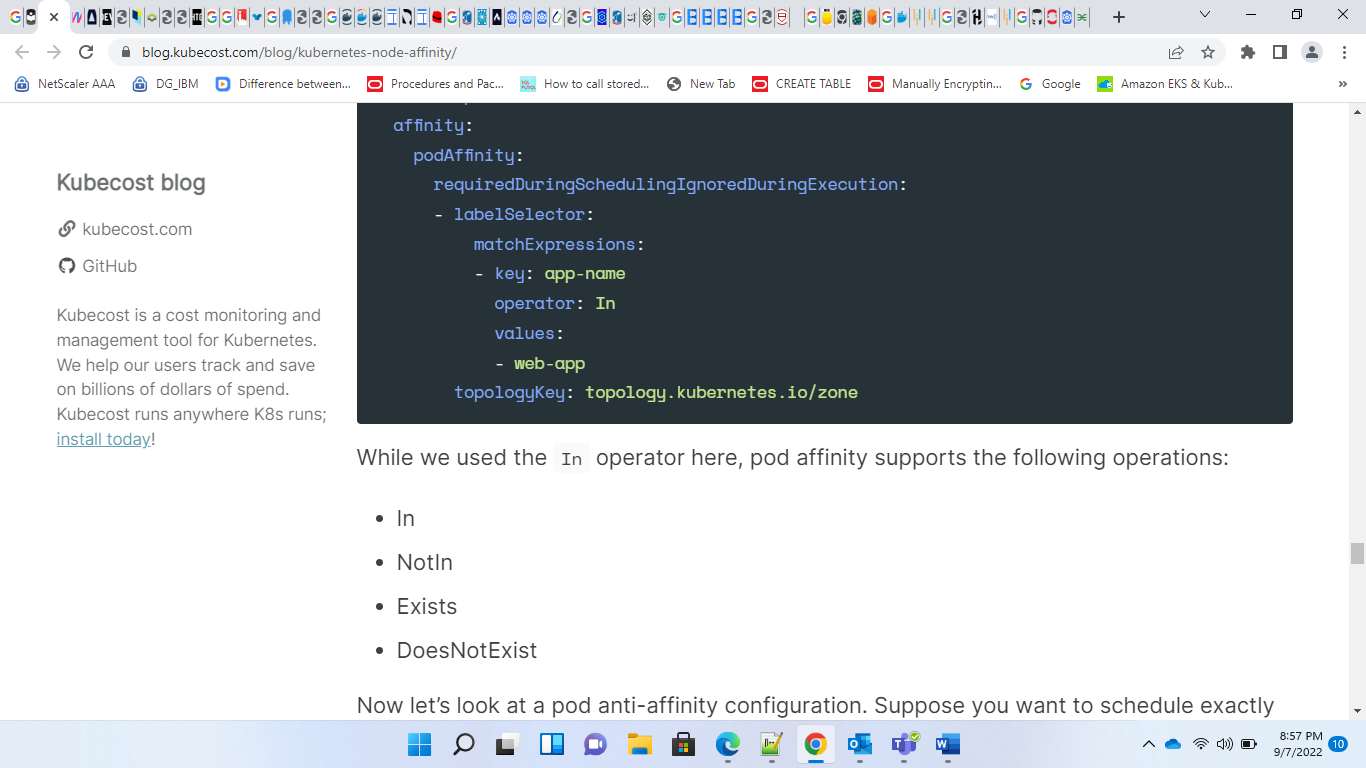
#### **Kubernetes node affinity rule example**



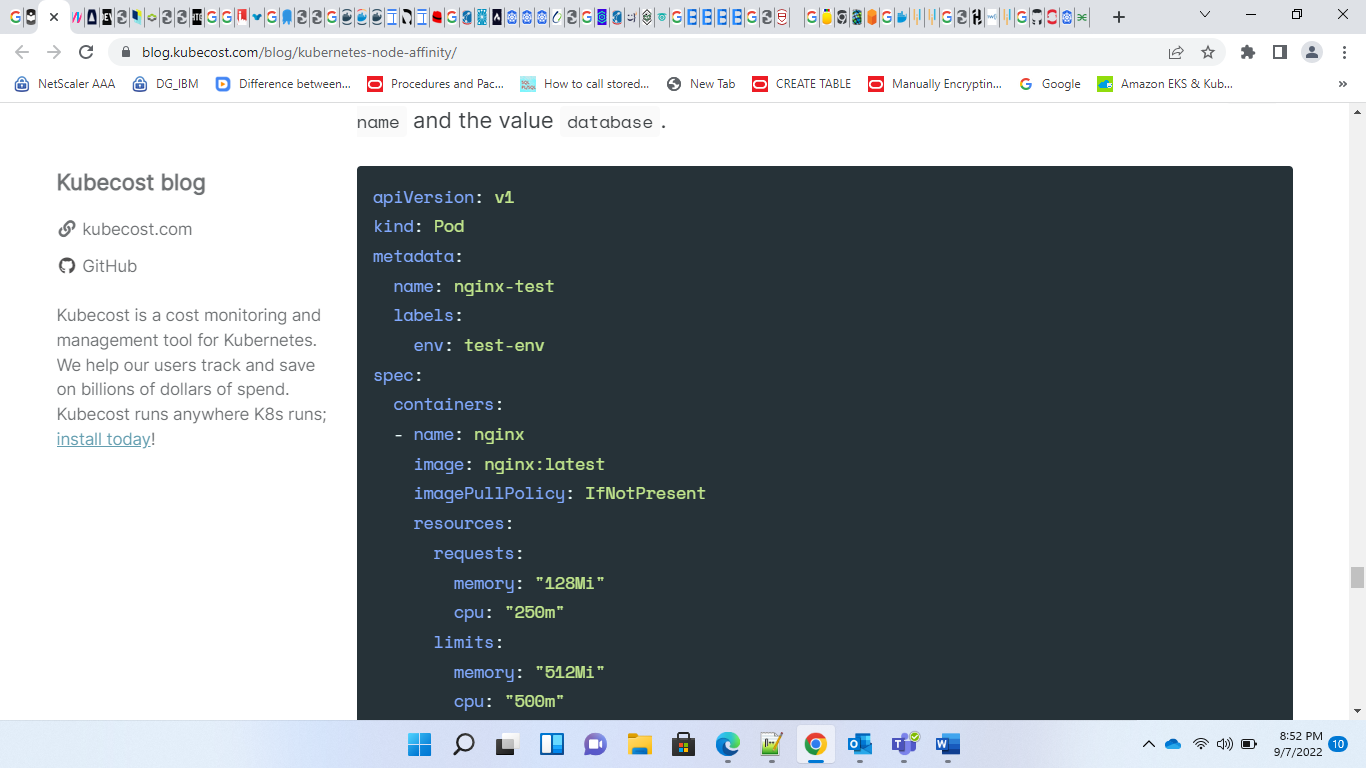
#### **Kubernetes pod affinity rule example**

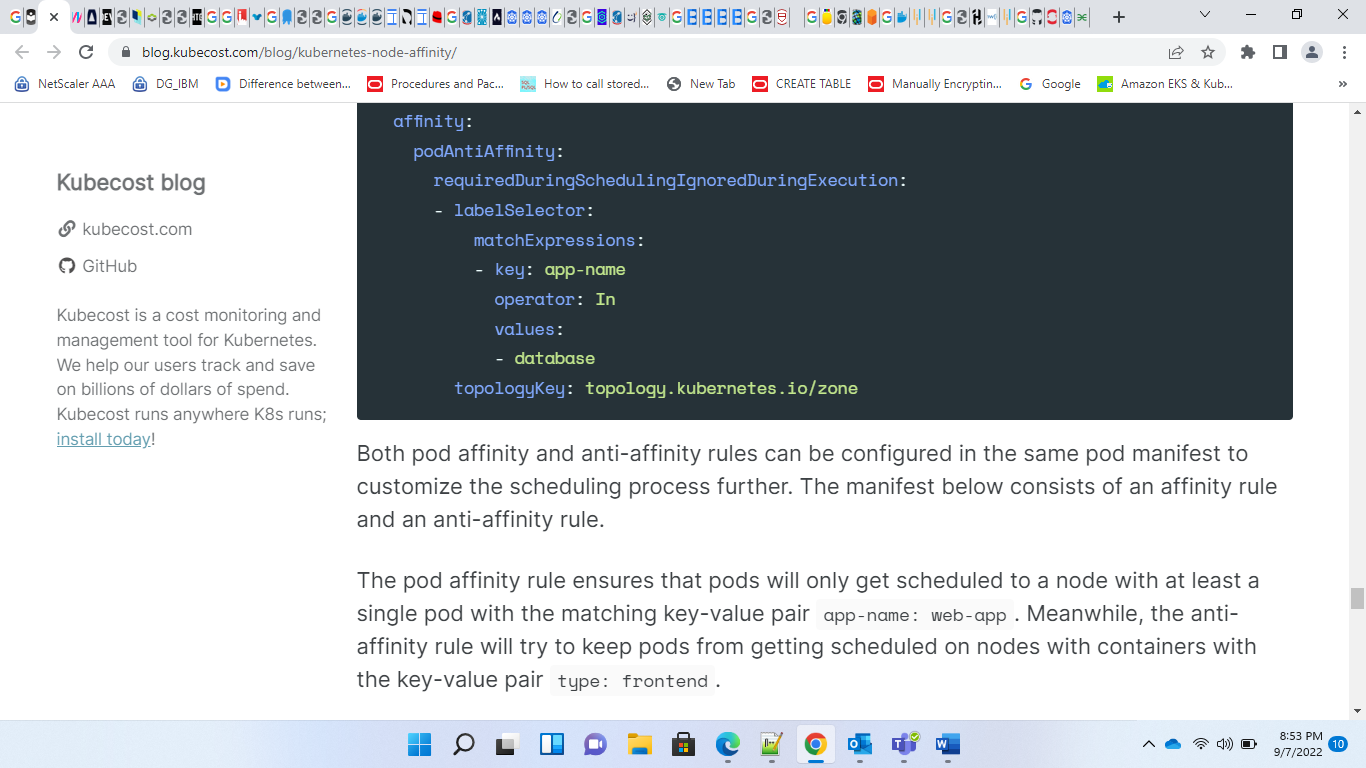


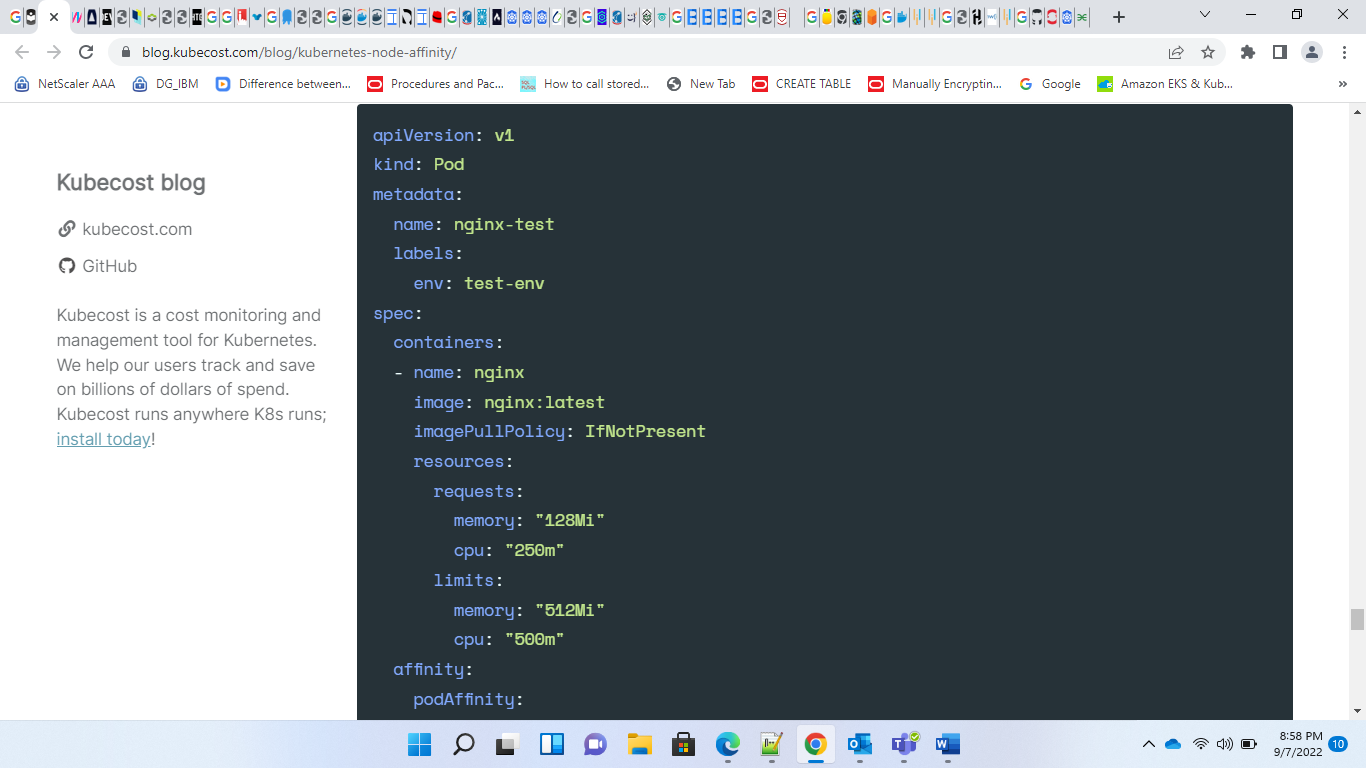


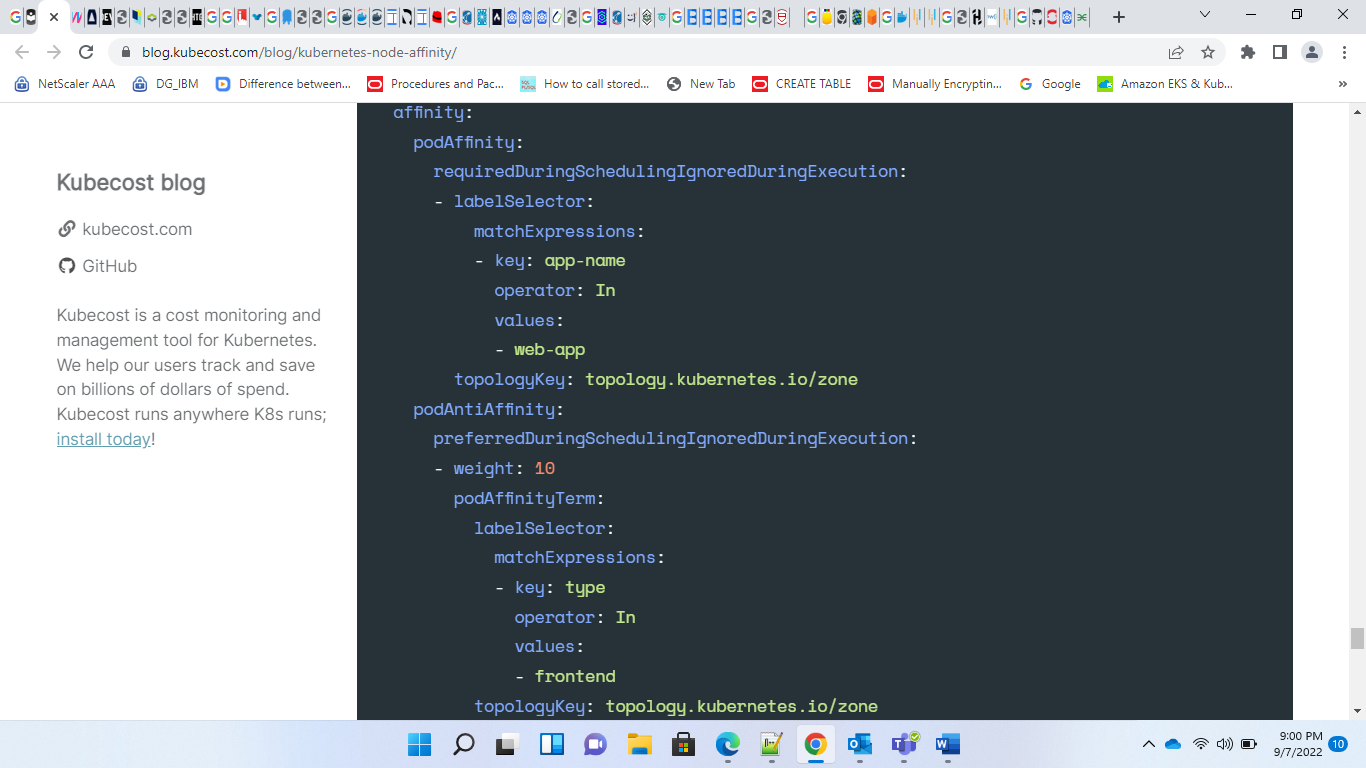


## Anti-affinity vs. taints and tolerations

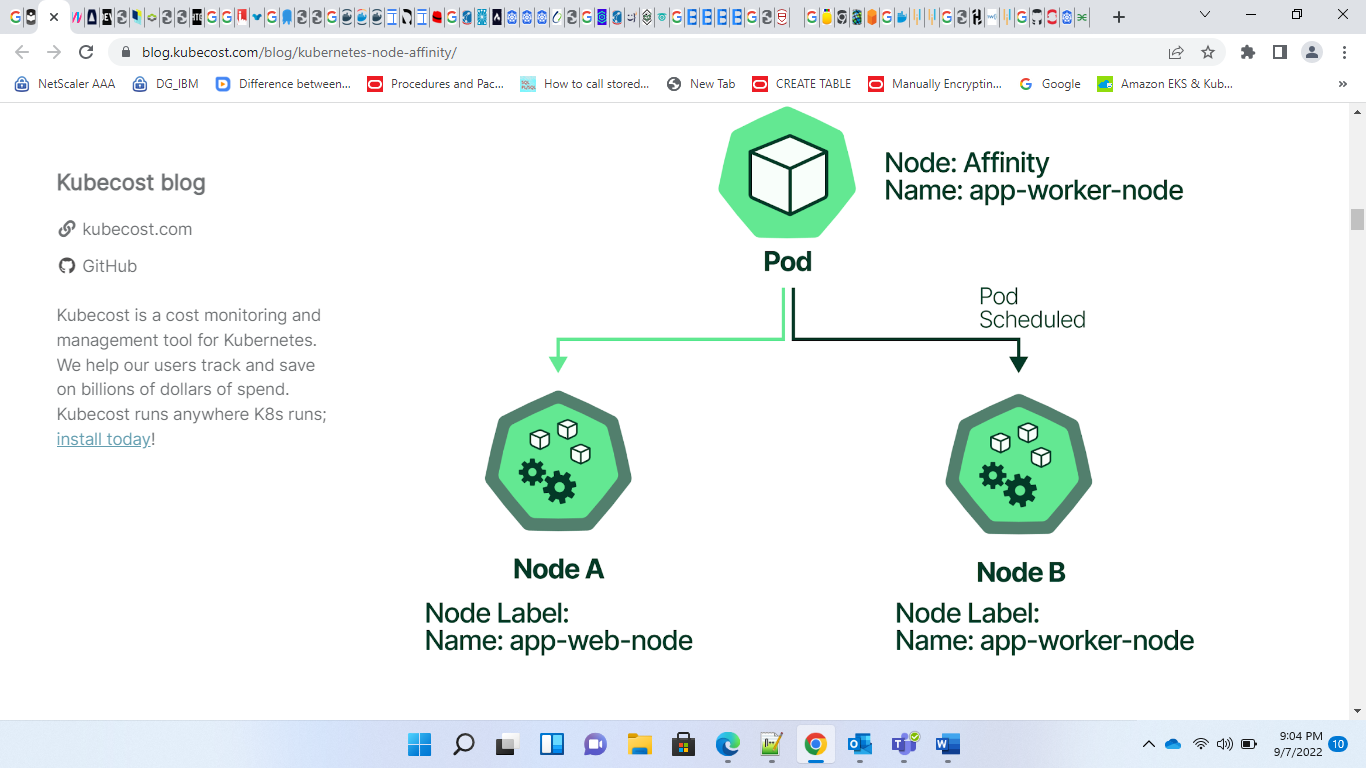




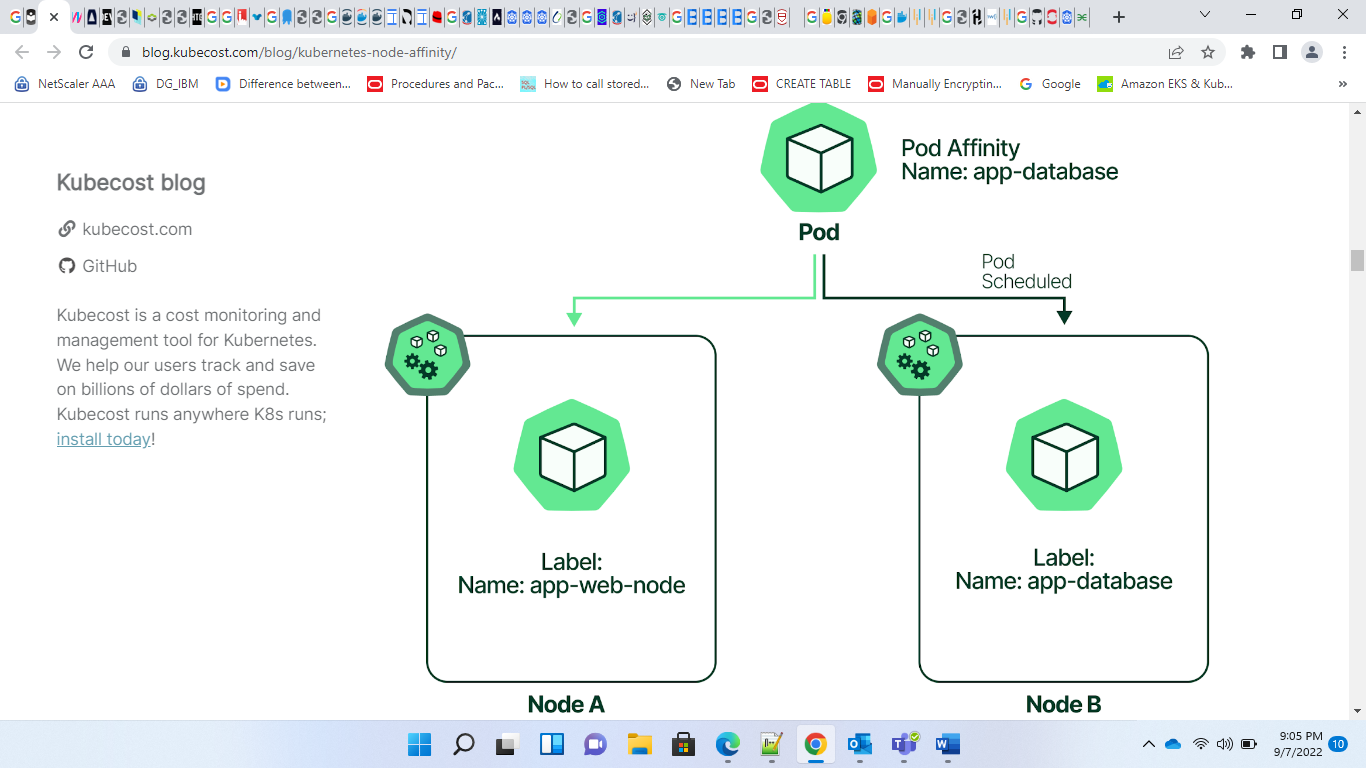




**NODE AFFINITY:**



**POD AFFINITY:**



We used the In operator in this manifest, but node affinity supports the following operators:

* In
* NotIn
* Exists
* DoesNotExist
* Gt
* Lt

**Node labels:**

* Kubernetes.io/arch
* Kubernetes.io/os
* Kubernetes.io/hostname
* Failure-domain.beta.kubernetes.io/zone
* Failure-domain.beta.kubernetes.io/region

<https://blog.kubecost.com/blog/kubernetes-node-affinity/>

<https://github.com/infracloudio/kubernetes-scheduling-examples/blob/master/podAffinity/README.md>

<https://docs.openshift.com/container-platform/3.11/admin_guide/scheduling/pod_affinity.html>

<https://kubernetes.io/docs/concepts/scheduling-eviction/assign-pod-node/>

<https://thenewstack.io/implement-node-and-pod-affinity-anti-affinity-in-kubernetes-a-practical-example/>

<https://www.armosec.io/blog/advanced-kubernetes-pod-to-node-scheduling/>

<https://medium.com/kokster/scheduling-in-kubernetes-part-2-pod-affinity-c2b217312ae1>

<https://vocon-it.com/2019/09/20/cka-14-kubernetes-affinity-and-anti-affinity/>

<https://medium.com/kubernetes-tutorials/learn-how-to-assign-pods-to-nodes-in-kubernetes-using-nodeselector-and-affinity-features-e62c437f3cf8>

<https://banzaicloud.com/blog/k8s-taints-tolerations-affinities/>