https://kubernetes.io/docs/concepts/scheduling-eviction/taint-and-toleration/

ku taint node ip-172-20-104-194.ec2.internal munnabhai=mbbs:NoSchedule

ku taint node ip-172-20-42-52.ec2.internal shankardada=mbbs:NoSchedule

ku taint node ip-172-20-70-121.ec2.internal vasoolraja=mbbs:NoSchedule

ku taint node ip-172-20-104-194.ec2.internal munnabhai=mbbs:NoExecute --overwrite

ku taint node ip-172-20-42-52.ec2.internal shankardada=mbbs:NoExecute --overwrite

ku taint node ip-172-20-70-121.ec2.internal vasoolraja=mbbs:NoExecute --overwrite

ku taint node ip-172-20-104-194.ec2.internal munnabhai=mbbs:NoExecute-

ku taint node ip-172-20-42-52.ec2.internal shankardada=mbbs:NoExecute-

ku taint node ip-172-20-70-121.ec2.internal vasoolraja=mbbs:NoExecute-

ku label node ip-172-20-104-194.ec2.internal disktype-

ku label node ip-172-20-42-52.ec2.internal disktype-

ku run sanju --image=sreeharshav/rollingupdate:v3

ku run chiru --image=sreeharshav/rollingupdate:v3

ku run kamal --image=sreeharshav/rollingupdate:v3

ku taint node ip-172-20-55-14.ec2.internal node-role.kubernetes.io/master:NoSchedule-

ku taint node ip-172-20-55-14.ec2.internal node-role.kubernetes.io/master=:NoSchedule

ku taint node ip-172-20-104-194.ec2.internal env-

ku taint node ip-172-20-42-52.ec2.internal env-

ku taint node ip-172-20-55-14.ec2.internal env-

ku taint node ip-172-20-70-121.ec2.internal env-

###########USING-SIMPLE-NODE-SELECTORS#####################

ku label node ip-172-20-55-14.ec2.internal disktype=hdd

apiVersion: v1

kind: Pod

metadata:

name: nginx-hdd

labels:

env: test

spec:

containers:

- name: nginx

image: nginx

nodeSelector:

disktype: hdd

#################TAINTS-TOLERATIONS################################

for node in $(ku get no --no-headers | cut -d " " -f1); do kubectl describe node $node | grep -i taint; done

PATCHING EXISTING DEPLOYMENT:

Create a file called patch.yml and paste the following:

spec:

template:

spec:

tolerations:

- effect: NoExecute

key: perf

value: high

ku patch deployments.apps nginx001 --patch "$(cat patch.yml)"

NEW DEPLOYMENT WITH TOLERATIONS:

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: chiru

name: chiru

spec:

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

run: chiru

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

run: chiru

spec:

containers:

- image: sreeharshav/rollingupdate:v3

imagePullPolicy: IfNotPresent

name: chiru

resources: {}

tolerations:

- key: "env"

operator: "Equal"

value: "dev"

effect: "NoSchedule"

dnsPolicy: ClusterFirst

restartPolicy: Always

tolerations:

- key: "key"

operator: "Equal"

value: "value"

effect: "NoSchedule"

or

tolerations:

- key: "diskType"

operator: "Exists"

effect: "NoExecute"

#################################---NODE-EFFINITY---#############################

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: chiru

name: chiru

spec:

progressDeadlineSeconds: 600

replicas: 2

revisionHistoryLimit: 10

selector:

matchLabels:

run: chiru

strategy:

rollingUpdate:

maxSurge: 25%

maxUnavailable: 25%

type: RollingUpdate

template:

metadata:

creationTimestamp: null

labels:

run: chiru

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: perf

operator: In

values:

- low

containers:

- image: sreeharshav/rollingupdate:v3

imagePullPolicy: IfNotPresent

name: chiru

resources: {}

tolerations:

- key: "env"

operator: "Equal"

value: "dev"

effect: "NoSchedule"

dnsPolicy: ClusterFirst

restartPolicy: Always

##############################POD-AFFINITY-ANTIAFFINITY##########################

**Note:** Inter-pod affinity and anti-affinity require substantial amount of processing which can slow down scheduling in large clusters significantly. We do not recommend using them in clusters larger than several hundred nodes.

**Note:** Pod anti-affinity requires nodes to be consistently labelled, in other words every node in the cluster must have an appropriate label matching topologyKey. If some or all nodes are missing the specified topologyKey label, it can lead to unintended behavior.

---

apiVersion: apps/v1

kind: Deployment

metadata:

name: redis-cache

spec:

selector:

matchLabels:

app: store

replicas:

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: sreeharshav/rollingupdate:v3

apiVersion: apps/v1

kind: Deployment

metadata:

name: redis-cache

spec:

selector:

matchLabels:

app: store

replicas: 3

template:

metadata:

labels:

app: store

spec:

affinity:

podAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

- labelSelector:

matchExpressions:

- key: app

operator: In

values:

- store

topologyKey: "kubernetes.io/hostname"

containers:

- name: redis-server

image: sreeharshav/rollingupdate:v3

the operator is Exists (in which case no value should be specified), or

the operator is Equal and the values are equal.

##################TAINT-TOLERATONS-NODEAFINITY################

kubectl taint <node> env=prod:NoExecute

apiVersion: apps/v1

kind: Deployment

metadata:

name: prod-deployment

labels:

app: nginx

spec:

replicas: 4

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: db

operator: In

values:

- mysql

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

tolerations:

- key: "env"

operator: "Equal"

value: "prod"

effect: "NoExecute"

============WITH-NO-NODESELECTOR=====================================

apiVersion: apps/v1

kind: Deployment

metadata:

name: test-deployment

labels:

app: nginx

spec:

replicas: 2

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: db

operator: In

values:

- mysql

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

tolerations:

- key: "diskType"

operator: "Equal"

value: "hdd"

effect: "NoSchedule"

=================NODE-AFFINITY==================================

============WITH-NO-NODESELECTOR=====================================

apiVersion: apps/v1

kind: Deployment

metadata:

name: test-deployment

labels:

app: nginx

spec:

replicas: 3

selector:

matchLabels:

app: nginx

template:

metadata:

labels:

app: nginx

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: env

operator: In

values:

- dev

containers:

- name: nginx

image: nginx:1.14.2

ports:

- containerPort: 80

========================================================================

#Using NodeName to schedule the POD.

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: deploy01

name: deploy01

spec:

replicas: 3

selector:

matchLabels:

run: deploy01

template:

metadata:

labels:

run: deploy01

spec:

containers:

- image: nginx

name: deploy01

nodeName: ip-172-20-72-92.ec2.internal

---

#Using NodeSelectors to schedule the POD

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: nodeselector-deploy

name: nodeselector-deploy

spec:

replicas: 3

selector:

matchLabels:

run: nodeselector-deploy

template:

metadata:

labels:

run: nodeselector-deploy

spec:

containers:

- image: nginx

name: deploy01

nodeSelector:

env: testing

---

#Using NodeSelectors to schedule the POD

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: nodeselector-prod

name: nodeselector-prod

spec:

replicas: 3

selector:

matchLabels:

run: nodeselector-prod

template:

metadata:

labels:

run: nodeselector-prod

spec:

containers:

- image: nginx

name: deploy01

nodeSelector:

env: prod

---

#Using NodeSelectors to schedule the POD

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: taint-env-testing

name: taint-env-testing

spec:

replicas: 3

selector:

matchLabels:

run: taint-env-testing

template:

metadata:

labels:

run: taint-env-testing

spec:

containers:

- image: nginx

name: taint-testing

tolerations:

- key: "env"

operator: "Equal"

value: "testing"

effect: "NoSchedule"

---

#Using NodeSelectors to schedule the POD

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: no-taint

name: no-taint

spec:

replicas: 4

selector:

matchLabels:

run: no-taint

template:

metadata:

labels:

run: no-taint

spec:

containers:

- image: nginx

name: no-taint-prod

tolerations:

- key: "env"

operator: "Equal"

value: "dev"

effect: "NoExecute"

---

#Using NodeSelectors to schedule the POD

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: ebs-high

name: ebs-high

spec:

replicas: 4

selector:

matchLabels:

run: ebs-high

template:

metadata:

labels:

run: ebs-high

spec:

affinity:

nodeAffinity:

requiredDuringSchedulingIgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:

- key: "EBS"

operator: In

values:

# - IO1

# - GP2

- SC1

containers:

- image: nginx

name: ebs-high-nginx

---

#Using PREFERED NODE AFFINITY

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

run: ebs-low

name: ebs-low

spec:

replicas: 6

selector:

matchLabels:

run: ebs-low

template:

metadata:

labels:

run: ebs-low

spec:

affinity:

nodeAffinity:

preferredDuringSchedulingIgnoredDuringExecution:

- weight: 1

preference:

matchExpressions:

- key: "EBS"

operator: In

values:

- SC1

containers:

- image: nginx

name: ebs-low-nginx