

PODS, ALLOCATION IN K8S

DATE: 25/10

PAGE: 1

- ① Scheduling:
- ② labels.
- ③ nodeSelector
(Affinity & Anti-Affinity)
- ④ nodeName
- ⑤ DAEMON SETS.
- ⑥ STATIC PODS
- ⑦ mirror pods.
- ⑧ Node Affinity.
- ⑨ Node Anti-Affinity

PODS ALLOCATION in Kubernetes

Section 37. [235]

- 1) What is Scheduling
- 2) Scheduling Process
- 3) nodeSelector
- 4) nodeName
- 5) Hands On Demonstration.

What is Scheduling:

- ① Scheduling is a process to assign the pods to nodes.
So that kubectl can run them.

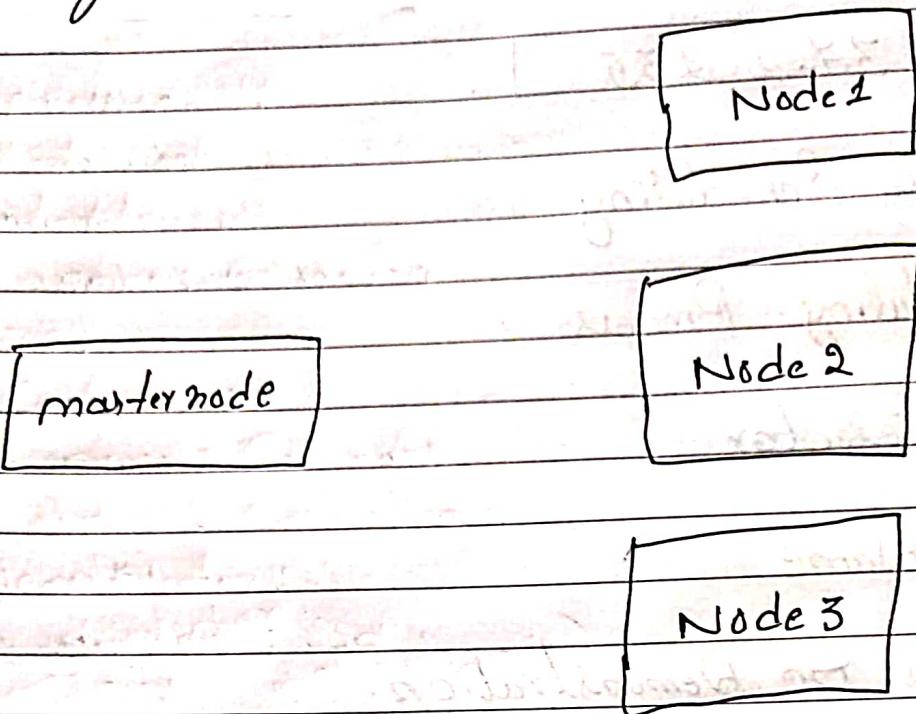
Whenever we are creating the pods in Kubernetes cluster,
the Kubernetes needs to schedule these pods to a
particular worker nodes.

For scheduling, there is a component in the K8s which is
called Scheduler.

Scheduler is a component of the Kubernetes master node
or we can call it Control plane Component Scheduler.

- ② It is responsible to schedule the pods on the nodes.

Suppose you have One master node & 3 worker nodes



- ④ Scheduler in master node is responsible to Schedule on worker nodes.
- ⑤ It is possible that Scheduler Schedule the pods on node 1 or node 2 or node 3.
- ⑥ How does the Schedule happen? ~~it's one of the~~
Scheduling of the pods depend on the multiple factors
~~& one~~
- ① Resource request varies ~~on~~ available node resources.

whenever you are creating the pod resource ~~request~~
it will identify the suitable worker node and you are defining a resource request in that particular pod the Kubernetes will read the pod resource request & it will identify the suitable worker node.

② Configuration like Node Labels.

You can define the labels on your nodes & then you can use the same labels in your Pod Container Specification in manifest(Yaml file). which are attached with the worker nodes and Schedule the pod Once the Condition will be met.

③ nodeSelector, Affinity & Anti-Affinity.

④ nodeSelector :- is define in Pod Specification to limit which Node(s) the pod can be Schedule on.

We need to define some key value pair, which is a "label" & this label key will match on the k8s worker nodes.

it may be possible that you will attach the same label with the multiple worker nodes.

in that case, again Kubernetes can Schedule your pod on any of the worker node, which is matching your label

But if Resource Demand is also defined, then Resource Demand will be made & pod will Schedule on a node which will satisfy the label match plus the resources demand match.

a-2 NodeSelector use Labels to Select the Suitable Node

Sample pod definition / manifest / Yaml file.

apiVersion: v1

kind: Pod

metadata:

- name: Cassandra

Spec:

Containers:

- name: Cassandra

- image: Cassandra

- nodeSelector:

- diskType: SSD

Node NAME :-

(1) User can bypass Scheduling & assign Pod to a specific node using Node Name.

(2) it means within your manifest file you can define the node name & Kubernetes will schedule your pod on a node which name is basically defined within the node name selector.

apiVersion: v1

kind: Pod

metadata:

- name: nginx

Spec:

Containers:

- name: nginx

- image: nginx

- nodeName: KubeNode-01

Q1 Typically, we should avoid these kind of practices:-

- (a) we should not use the node name because if we are executing the k8s in the Cloud environment, then the infrastructure will be dynamic & it may possible that the worker nodes are basically going down dynamically & new worker nodes will be spun up dynamically.

Lab:- 23

① Go to the directory where Scripts are stored

cd ~~scripts~~ kuberfiles

② vi pod-Scheduling :

apiVersion: v1

kind: Pod

metadata :

name: nginx-nodeSelector

spec :

Containers :

- name : nginx

image : nginx

nodeSelector :

diskType : ssd

apiVersion: v1

kind: Pod

metadata:

name: nginx-nodeName

spec:

Containers:

- name: nginx

image: nginx

nodeName: kube-01

Separate C

apiVersion: v1

kind: Pod

metadata:

name: frontend-app

spec:

Containers:

- name: app

image: alpine

Command:

- Sleep

- '3600'

resources:

requests:

memory: 64mi

CPU: 1000m

nodeSelector:

disktype: SSD

apiVersion: v1

kind: Pod

metadata:

name: frontend-app-2

spec:

Containers:

- name: app

image: alpine

Command:

- sleep

- '3600'

resources:

requests:

memory: 64Mi

CPU: 1000m

③ Execute Only first manifest very disktype = ssd.

④ Kubectl apply -f <filename.yaml>

⑤ Kubectl get pod -o wide

We notice that pod is in pending, as, we don't have SSD in your Cluster.

⑥ To Check the Label present in your Cluster.

Kubectl get nodes --show-labels

We notice the label is not present disktype = ssd.

Let us Create the Label.

- ⑦ To Create Labels to respective nodes Execute following Command:

`kubectl get label nodes <node-name> disktype=ssd`

Ex:- `kubectl label nodes k8s-worker-02 disktype=sqd`

- ⑧ `kubectl get nodes -show-labels`

- ⑨ `kubectl get pods -o wide`

Using node name :-

Same steps as mentioned for previous Example.

- ⑩ Copy second manifest & Save the file.

- ⑪

Section 37 [237]

DAEMON SETS in Kubernetes.

- ① what is DaemonSets
- ② DaemonSets working
- ③ DaemonSets and Scheduling
- ④ Hands On Demonstration

DaemonSet:-

- ① DaemonSet in K8s is an object like Pods, which will automatically run a copy of a pod on each node.

So whenever you want to Create ~~that~~ pod, a copy of a particular pod will execute on each node present in your Kubernetes Cluster

- ② DaemonSet Run a Copy of pod on new node as they add it to a Cluster

So whenever we are creating a daemon set, it will execute a copy of defined pod on each available node in K8s.

- ③ So daemon set will make sure that whatever the nodes are present in your Kubernetes Cluster at a point of time, a copy of a particular pod which is defined in daemon set will execute on each & every node which is a part of K8s Cluster.

(4)

DaemonSet will be helpful in case of monitoring, log collection, proxy configuration etc.

If you want to put some kind of monitoring on your k8s worker nodes, you can execute the monitoring application.

If you want to execute some kind of log collection, some kind of data collection.

You can execute that particular thing from a daemon set so that that log collection or data collection will be done from all of the pods.

If you want to configure the proxy setting by which you can k8s over the internet or some external n/w, then you can set that particular thing with the daemon set as well.

Scheduling and DaemonSets

(1)

DaemonSets follows normal scheduling rules around node labels, taints & tolerations.

So whatever the rules are applicable for the overall scheduling the all of the rules are basically applicable with the daemon set as well.

(2)

If pods normally not scheduled on a node, daemonset will also not create copy of pod on that node.

(as soon as the schedule will be available on that particular node, daemonset will create a copy of a pod on that node).

Lab [238] :- DaemonSets in K8s.

① Go To directory where Scripts are stored.

cd k8sdirectory

② vi daemonset.yaml

apiVersion : apps/v1

kind : DaemonSet

metadata :

name : logging

Spec :

Selectors :

matchLabels :

app : httpd-logging

Template :

metadata :

labels :

app : httpd-logging

Spec :

Containers :

- name : webserver

image : httpd

ports :

- ContainerPort : 80

Save & exit .

- ③ Kubectl apply -p daemonsets.yaml
- ④ Kubectl get daemonsets
- ⑤ Kubectl get pods -o wide
- ⑥ Kubectl describe daemonsets [logging] ^{none of the pod}

Section 37 [239]

~~REVIEW~~

STATIC Pods in Kubernetes

- ① what is static pod.
- ② mirror pod.
- ③ Hands On Demo

Static Pod :-

In Kubernetes, there is a typical architecture, master slave architecture & all the operations which are performing in a k8s cluster master node is basically managing or master node is basically managing all the operations in your k8s node.

- 1) But Static pod pod directly managed by the Kubelet On K8s nodes.
- 2) So Kubelet is an agent which works On the worker nodes, & static pods are basically directly managed by the ~~K8s master~~ - Kubelet's.

They are not managed by the Control plane or Kubernetes API Server.

- ⑤ Kubelet watch it's static pod & if restarts if it fails.
if that fails or that exit with the error code the Kubelet will automatically restart the pod.

- ⑥ To Create the Static Pod, you need to define a YAML manifest file.

- ⑦ That YAML manifest file is similar to the pod manifest file, but you need to create that particular file at a specific manifest location and that location Kubelet will automatically identify the YAML file & create a static pod.

- ⑧ what is the use of Static Pod :-

Suppose you want to put some kind of monitoring or you want to put some kind of configuration.

that you want to execute without the API Server or without the Controlplane.

- ⑨ Static pod will execute on each & every node, which is the part of your k8s cluster.

Mirror Pod :-

- (1) mirror pods are nothing but a replica of the Static pod as the static pod is being managed by the kubelet so we don't have any way to see the pod status or to get the pod information.
- (2) So that kubelet will create a mirror pod for each static pod so that user can access the static pod by a Kubernetes API servers or Kubernetes Control plane.
- (3) Although you can monitor the static pod, but you cannot change or update the static pod by a mirror pod.
- (4) If you want to perform some kind of operations, changing the state of your static pod or doing some kind of update in your static pod, you cannot execute that kind of thing by a mirror pod.
- (5) For that change, you need to update the YAML file on the node itself & kubelet will update the status.

But By a mirror pod it is not possible to change the state of your static pod.

Static Pod Lab [240]

Go to worker node (Any worker node) \rightarrow SSH to worker node.

② cd /etc/kubernetes/manifests/

\Rightarrow :- ls

③ vi staticpods.yaml

apiVersion: V1

kind: Pod

metadata:

name: nginx-~~andreas~~ Static-Pod

Spec:

Containers:

- name: nginx

image: nginx

④ Save & close

⑤ Wait for sometime Kubelet will Create a pod

⑥ If you don't want to wait, Restart the Kubelet

⑦ sudo systemctl restart kubelet

⑧ Go to master node & check if Static-Pod is created

Kubectl -o get pod -o wide

We see that static pod is running

(10) kubectl describe pod <podname>

(11) Delete and other command don't work.

SECTION 37 [241]

NODE AFFINITY

(1) What is Node Affinity

(2) Examples of Node Affinity & ANTI-Affinity

(3) Hands On Demo

Node Affinity :-

(1) Node Affinity is enhanced version of Node Selector

(2) We have already learned about the node selector where we need to define the key-value pair, which is a label on node and on the basis of that label node selector work.

(2) In node affinity is enhanced version of the node selector. Because:-

In node affinity we can put condition in node selector & that is introduced in K8s 1.4

④ Node affinity is also a pod allocation mechanism in K8s.

By node-affinity and node-selector, we can schedule a pod on a particular node or set of nodes which have a specific tag or a specific label.

⑤ If you don't want to schedule pod on a particular node? We can use Node Anti-Affinity.

⑥ Anti-Affinity is Opposite of Affinity & Node Selector Concept.

Difference with Node-Affinity & node selector manifest:

Node Selector manifest | Node Affinity (from Spec)

apiVersion: v1

kind: Pod

metadata:

name: nginx-nodeSelector

Spec:

Containers:

- name: nginx

- image: nginx

affinity:

Spec:

Containers:

- name: nginx

- image: nginx

nodeSelector:

- diskType: ssd

nodeAffinity:

Required During Scheduling

Ignored During Execution:

node Selector Terms:

- match Expressions:

- Key: diskType

Operator: In

Values:

- ssd

In manifest we just clean the Condition :-

* "Searched During Scheduling Ignored During Execution" :-

Searched During Scheduling :-

- 1) must fulfill the Condition at the time of Pod Creation

Scheduling

Condition Should met before Scheduling On Node.

- 2) also called HARD Affinity

Ignored During Execution :-

- 1) Pod will still run if labels on a node change & affinity rules are no longer met.

means. After pod Creation, if label is changed on node, it does not affects. Pod will be running or affected.

* ~~Prefered~~ Preferred During Scheduling Ignored During Execution" :-

Preferred During Scheduling :-

- ① Prefer Node which will fulfil The Condition but will not guarantee

- ② also called Soft Affinity.

Node Anti-Affinity : manifest

Spec

Containers:

- name: nginx

image: nginx

affinity:

nodeAffinity:

Gleamed During Scheduling Ignored During Execution:

nodeSelector Terms:

- matchExpressions:

- key: disktype

operator: NotIn \rightsquigarrow

values:

- ssd

Only Operator
change in
Node Anti-
Affinity
manifest.

For demo & Lab Head: Section 37. [249]