CKA by ZEAL vora

Microsoft Windows [Version 10.0.19045.5440]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 14m v1.31.4

C:\Windows\system32>kubectl create deployment apache --image=httpd

deployment.apps/apache created

**#create a container with image httpd inside kubernetes**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-7hrlq 1/1 Running 0 84s

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 33m v1.31.4

deepanshu\_dev@DESKTOP-BA6T73K:~$ ls -l ~/.kube/config

-rw-r--r-- 1 deepanshu\_dev deepanshu\_dev 5636 Feb 10 13:41 /home/deepanshu\_dev/.kube/config

deepanshu\_dev@DESKTOP-BA6T73K:~$

* IN CMD
* C:\Windows\system32>ls -l ~/.kube/config
* 'ls' is not recognized as an internal or external command,
* operable program or batch file.
* C:\Windows\system32>dir %USERPROFILE%\.kube\config
* Volume in drive C has no label.
* Volume Serial Number is BE41-028C
* Directory of C:\Users\user121\.kube
* 21-03-2025 09:48 5,696 config
* 1 File(s) 5,696 bytes
* 0 Dir(s) 628,095,053,824 bytes free
* C:\Windows\system32>type %USERPROFILE%\.kube\config
* apiVersion: v1
* clusters:

C:\Windows\system32>kubectl run nginx1 --image=nginx

**pod/nginx1 created**

C:\Windows\system32>kubectl get deployments

NAME READY UP-TO-DATE AVAILABLE AGE

apache 1/1 1 1 20m

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-7hrlq 1/1 Running 0 20m

nginx1 0/1 ContainerCreating 0 19s

C:\Windows\system32>kubectl logs nginx1

/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration

C:\Windows\system32>kubectl describe pod nginx1

Name: nginx1

Namespace: default

Priority: 0

Service Account: default

Node: docker-desktop/192.168.65.3

C:\Windows\system32>kubectl exec -it nginx1 -- /bin/bash

root@nginx1:/#

exit

C:\Windows\system32>kubectl delete pod nginx1

pod "nginx1" deleted

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-7hrlq 1/1 Running 0 25m

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-7hrlq 1/1 Running 0 25m

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 50m v1.31.4

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

pod.yml - to create pod we need api version – metadata –

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  containers:

  - name: nginx-container

    image: nginx

**### Create Pod Using Manifest File**

```sh

kubectl apply -f pod.yaml

```

**#### List the Running Pods**

```sh

kubectl get pods

```

**#### Delete the Resources Created via pod.yaml file**

```sh

kubectl delete -f pod.yaml

```

pod/nginx created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-km7nr 1/1 Running 0 11m

nginx 1/1 Running 0 19s

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl delete -f pod.yml

pod "nginx" deleted

C:\Windows\system32>kubectl api-resources

NAME SHORTNAMES APIVERSION NAMESPACED KIND

bindings v1 true Binding

componentstatuses cs v1 false ComponentStatus

replicationcontrollers rc v1

# **for pod we have api version v1 and for deplpoyments we have api version apps/v1,**

**# we can see all this using api-resources**

apiVersion: v2

kind: Pod

metadata:

  name: nginx

spec:

  containers:

  - name: nginx-container

    image: nginx

#### **What This Manifest Defines:**

* **apiVersion: v1** → Uses the core Kubernetes API version.
* **kind: Pod** → Specifies that this YAML defines a pod.
* **metadata:** → Defines the pod name as nginx.
* **spec:** → Specifies the pod's behavior.
  + **containers:** → Defines a single container inside the pod.
    - **name: nginx-container** → The name of the container.
    - **image: nginx** → Uses the nginx image from Docker Hub.

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

error: resource mapping not found for name: "nginx" namespace: "" from "pod.yml": no matches for kind "Pod" in version "v2"

ensure CRDs are installed first

PS E:\CKA BY zeal vora\PracticeME\section2>

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  containers:

  - name: nginx-container

    image: nginx

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

pod/nginx created

**# Creating a pod using manifest file**

apiVersion: v1

kind: Pod

metadata:

  name: nginx-2

spec:

  containers:

  - name: nginx-container

    image: nginx

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

pod/nginx-2 created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-km7nr 1/1 Running 0 31m

nginx 1/1 Running 0 70s

nginx-2 1/1 Running 0 35s

C:\Windows\system32>kubectl delete pods apache-7d8d5c5d5-km7nr

pod "apache-7d8d5c5d5-km7nr" deleted

C:\Windows\system32>kubectl run nginx --image=nginx

pod/nginx created

**# creating pod**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-4j9pp 1/1 Running 0 12m

nginx 0/1 ContainerCreating 0 8s

C:\Windows\system32>kubectl run nginx2 --image=nginx --dry-run=client

pod/nginx2 created (dry run)

**$ nginx2 pod never be created , as it just validate resoeurces , in real this pod never be created**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

C:\Windows\system32>kubectl run nginx2 --image=nginx -o yaml

apiVersion: v1

**$ throguht this yaml file a pod is created ,**

apiVersion: v1

kind: Pod

metadata:

  name: multi-container-pod

spec:

  containers:

  - name: nginx-container

    image: nginx

  - name: redis-container

    image: redis

apiVersion: v1

kind: Pod

metadata:

  name: nginx-2

spec:

  containers:

  - name: nginx-container

    image: nginx

C:\Windows\system32>kubectl run nginx4 --image=nginx --dry-run=client -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

$ **pod is not create but generate a manift file to create a pod , we are gerneatiing a mainst file thorugh a cli command**

**$ this pod is never be created in real just see manifst file as an output**

C:\Windows\system32>kubectl run nginx4 --image=nginx --dry-run=client -o yaml > pod-custom.yaml

**# we are stoing manist file into pod-custom.yaml file or foarmt**

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

error: resource mapping not found for name: "nginx" namespace: "" from "pod.yml": no matches for kind "Pod" in version "v2"

ensure CRDs are installed first

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

pod/nginx created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

pod/nginx-2 created

PS E:\CKA BY zeal vora\PracticeME\section2>

\* History restored

See 'kubectl --help' for usage.

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f multi-container-pod.yml

pod/multi-container-pod created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod.yml

pod/nginx-2 created

Microsoft Windows [Version 10.0.19045.5440]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 4h57m v1.31.4

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-wxg7z 1/1 Running 1 (3m38s ago) 179m

C:\Windows\system32>kubectl get deployments

NAME READY UP-TO-DATE AVAILABLE AGE

apache 1/1 1 1 4h36m

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-wxg7z 1/1 Running 1 (32m ago) 3h28m

multi-container-pod 0/2 ContainerCreating 0 8s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-wxg7z 1/1 Running 1 (41m ago) 3h36m

multi-container-pod 2/2 Running 0 8m31s

nginx-2 1/1 Running 0 28s

C:\Windows\system32>kubectl describe pod multi-container-pod

Name: multi-container-pod

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-wxg7z 1/1 Running 1 (44m ago) 3h39m

multi-container-pod 2/2 Running 0 11m

nginx-2 1/1 Running 0 3m19s

C:\Windows\system32>kubectl exec -it nginx-2 -- /bin/bash

root@nginx-2:/#

exit

C:\Windows\system32>kubectl exec -it multi-container-pod -- /bin/bash

Defaulted container "nginx-container" out of: nginx-container, redis-container

root@multi-container-pod:/# ls

bin boot dev docker-entrypoint.d docker-entrypoint.sh etc home lib lib64 media mnt opt proc root run sbin srv sys tmp usr var

root@multi-container-pod:/#

exit

C:\Windows\system32>kubectl exec -it multi-container-pod -c redis-container -- /bin/bash

root@multi-container-pod:/data# ls

root@multi-container-pod:/data# pwd

/data

root@multi-container-pod:/data#

exit

D:\CKA BY zeal vora\kuberntes\_practice\_23 mar>kubectl exec -it mutli-container-pod -c redis-container -- redis-cli

E0323 10:29:22.030133 13484 websocket.go:296] Unknown stream id 1, discarding message

127.0.0.1:6379>

C:\Windows\system32>kubectl delete -f pod.yml

error: the path "pod.yml" does not exist

**# Cmd will execute the containers automatically we don’t need to specify any command here thorugh cmd it will execute. – through cmd it will execute the appn inside the container automatically**

FROM busybox:latest

CMD ["ping","-c","3","google.com"]

PS E:\CKA BY zeal vora\PracticeME\section2\dockerfile> docker build -t myimage .

[+] Building 58.4s (6/6) FINISHED

C:\Windows\system32>docker run myimage

PING google.com (142.250.192.174): 56 data bytes

64 bytes from 142.250.192.174: seq=0 ttl=63 time=37.821 ms

64 bytes from 142.250.192.174: seq=1 ttl=63 time=56.056 ms

64 bytes from 142.250.192.174: seq=2 ttl=63 time=37.054 ms

--- google.com ping statistics ---

3 packets transmitted, 3 packets received, 0% packet loss

round-trip min/avg/max = 37.054/43.643/56.056 ms

Dockerfile

FROM ubuntu

ENTRYPOINT [ "/bin/echo" ]

CMD [ "hello","world" ]

PS E:\CKA BY zeal vora\PracticeME\section2\sile> docker build -t sile .

[+] Building 14.0s (4/6)

C:\Windows\system32>docker run sile

hello world

C:\Windows\system32>docker run sile how are you

how are you

C:\Windows\system32>

**$ cmd command is over written by how are you and is append**

C:\Windows\system32>docker run myimage

PING google.com (142.250.192.174): 56 data bytes

64 bytes from 142.250.192.174: seq=0 ttl=63 time=37.821 ms

64 bytes from 142.250.192.174: seq=1 ttl=63 time=56.056 ms

64 bytes from 142.250.192.174: seq=2 ttl=63 time=37.054 ms

--- google.com ping statistics ---

3 packets transmitted, 3 packets received, 0% packet loss

round-trip min/avg/max = 37.054/43.643/56.056 ms

C:\Windows\system32>docker run sile

hello world

C:\Windows\system32>docker docker run sile how are you

docker: 'docker' is not a docker command.

See 'docker --help'

C:\Windows\system32>docker run sile how are you

how are you

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED

**$ CREATing docker iamge and using that image creating kuberntes pods**

C:\Windows\system32>kubectl run my-echo-pod --image=myimage --command -- echo "Hello from kubeclt runs"

**pod/my-echo-pod created**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-nghtr 1/1 Running 0 138m

my-echo-pod 0/1 ContainerCreating 0 9s

C:\Windows\system32>kubectl logs my-echo-pod

Error from server (BadRequest): container "my-echo-pod" in pod "my-echo-pod" is waiting to start: trying and failing to pull image

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-nghtr 1/1 Running 0 140m

my-echo-pod 0/1 ImagePullBackOff 0 2m36s

C:\Windows\system32>kubectl logs my-echo-pod

**Error from server (BadRequest): container "my-echo-pod" in pod "my-echo-pod" is waiting to start: trying and failing to pull image**

C:\Windows\system32>kubectl delete my-echo-pod

error: the server doesn't have a resource type "my-echo-pod"

C:\Windows\system32>kubectl delete pod my-echo-pod

pod "my-echo-pod" deleted

FROM busybox:latest

CMD ["ping","-c","5","google.com"]

docker build –t busybox .

C:\Windows\system32>docker images

REPOSITORY TAG IMAGE ID CREATED SIZE

#READ YHA TAK

C:\Windows\system32>kubectl run ping-pong --image=busybox --command -- ping "-c" "30" "google.com"

pod/ping-pong created

**# create image using busybox image – created using docker file**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-nghtr 1/1 Running 0 151m

ping-pong 0/1 ContainerCreating 0 17s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-nghtr 1/1 Running 0 152m

ping-pong 1/1 Running 1 (33s ago) 80s

C:\Windows\system32>kubectl logs ping-pong

PING google.com (142.250.183.238): 56 data bytes

C:\Windows\system32>kubectl delete pod ping-pong

pod "ping-pong" deleted

Using cmd and arg we **can specify what things to be done when container starts**

cmd-arg.yml

apiVersion: v1

kind: Pod

metadata:

  name: new-ping-pod

spec:

  containers:

  - name: ping-container

    image: busybox

    command: ["ping"]

    args: ["-c","60","google.com"]

apiversion: v1

kind: pod

metadata:

name: new-ping-pod

spec:

containers:

- name: ping-container

Image: busybox

Command: [“ping”]

args: [“-c”, “30”,“google.com”]

apiversion: v1

kind: pod

metadata:

name:demo-pod

spec:

containers:

- name: nginx-container

Image: nginx

command: [“ping”]

args: [“-c”, “30”,“google.com”]

PS E:\CKA BY zeal vora\PracticeME\section2\dockerfile> kubectl apply -f cmd-arg.yml

error: the path "cmd-arg.yml" does not exist

PS E:\CKA BY zeal vora\PracticeME\section2\dockerfile> cd ..

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f cmd-arg.yml

pod/new-ping-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-nghtr 1/1 Running 0 163m

new-ping-pod 1/1 Running 0 27s

C:\Windows\system32>kubectl logs new-ping-pod

PING google.com (142.250.194.142): 56 data bytes

C:\Windows\system32>kubectl delete pod --all

pod "apache-7d8d5c5d5-nghtr" deleted

pod "new-ping-pod" deleted

cmd-clarity.yml

apiVersion: v1

kind: Pod

metadata:

  name: new-ping-pod

spec:

  containers:

  - name: ping-container

    image: busybox

    command: ["ping","-c","60","google.com"]

apiversion: v1

kind: pod

metadata:

name: new-ping-pod

spec:

containers:

- name: ping-pong-c1

Image: busybox

command: [“ping”,“-c”,“google.com”,]

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f cmd-arg.yml

pod/new-ping-pod created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f cmd-clarity.yml

pod/new-ping-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-4kpff 1/1 Running 0 6m5s

new-ping-pod 1/1 Running 0 43s

C:\Windows\system32>kubectl logs new-ping-pod

PING google.com (142.250.194.142): 56 data bytes

C:\Windows\system32>kubectl delete pods new-ping-pod

pod "new-ping-pod" deleted

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-4kpff 1/1 Running 0 17m

new-ping-pod 1/1 Running 2 (34s ago) 2m9s

C:\Windows\system32>kubectl logs new-ping-pod

PING google.com (142.250.194.142): 56 data bytes

64 bytes from 142.250.194.142: seq=0 ttl=63 time=119.725 ms

64 bytes from 142.250.194.142: seq=1 ttl=63 time=47.671 ms

64 bytes from 142.250.194.142: seq=2 ttl=63 time=23.618 ms

64 bytes from 142.250.194.142: seq=3 ttl=63 time=23.362 ms

64 bytes from 142.250.194.142: seq=18 ttl=63 time=45.525 ms

--- google.com ping statistics ---

30 packets transmitted, 27 packets received, 10% packet loss

round-trip min/avg/max = 12.109/74.389/422.113 ms

apiVersion: v1

kind: Pod

metadata:

  name: new-ping-pod

spec:

  containers:

  - name: ping-container

    image: busybox

    command:

    - "ping"

    - "-c"

    - "30"

    - "google.com"

apiversion: v1

kind: pod

metadata:

name: demo-pod

spec:

containers:

- name: ping-container

image: ping-pong

command:

- “ping”

- “-c”

- “30”

- “google.com”

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f cmd-clarity.yml

pod/new-ping-pod created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-4kpff 1/1 Running 0 15m

new-ping-pod 0/1 ContainerCreating 0 9s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

apache-7d8d5c5d5-4kpff 1/1 Running 0 17m

new-ping-pod 1/1 Running 2 (34s ago) 2m9s

C:\Windows\system32>kubectl logs new-ping-pod

PING google.com (142.250.194.142): 56 data bytes

64 bytes from 142.250.194.142: seq=0 ttl=63 time=119.725 ms

C:\Windows\system32>kubectl run custom-pods --image=busybox --dry-run=client -o yaml --command -- echo "Hello World!"

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: custom-pods

name: custom-pods

spec:

containers:

- command:

- echo

- Hello World!

image: busybox

name: custom-pods

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

# manisft file is generates and

**# cmd is entry point for executing of our container while args refer to arguments or parameters that can be changes and passed to cmd ,arg like meaning or word like “hello”**

**# using dry run pod never be created just basic yaml file we have**

C:\Windows\system32>kubectl explain pod

KIND: Pod

VERSION: v1

DESCRIPTION:

Pod is a collection of containers that can run on a host. This resource is

created by clients and scheduled onto hosts.

C:\Windows\system32>docker run -d --name nginx-c1 nginx

5ead9bc9ecb0dbfa920927b9180d85aff672dedd1f14b69d7a05bc53e4b122a0

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

5ead9bc9ecb0 nginx "/docker-entrypoint.…" 8 seconds ago Up 7 seconds 80/tcp nginx-c1

C:\Windows\system32>docker inspect nginx-c1

C:\Windows\system32>docker run -d --name=nginx2 -p 80:80 nginx

271979be1a9775f3ecf6fb8870ca44cbf39ae8fb9e20cf4f01328b284426844e

# 80 port on left side is host port ,anyone accessing on host port 80 it will be redirect to nginx port 80 on right side automaiclly on nginx container

# we have expose the port of our nginx container

C:\Windows\system32>docker ps -a

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

**# what if don’t expoese port**

**Then anyone don’t idnefty who can access or now where is our appn run**

We can expose the port of out conatienr in our kuberntes

pod-expose.yml

apiVersion: v1

kind: Pod

metadata:

  name: nginx-3

spec:

  containers:

  - name: nginx-container

    image: nginx

    ports:

    - containerPort: 8080

apiversion:v1

kind: pod

metadata:

name:container-pod

spec:

containers:

- name: ngnix-c1

image: nginx

ports:

- containerport: 8080

**apiVersion: v1**

**kind: Service**

**metadata:**

**name: nginx-service**

**spec:**

**selector:**

**app: nginx-3**

**ports:**

**- protocol: TCP**

**port: 80 # Port on which the service will be exposed**

**targetPort: 8080 # Port inside the container**

**type: LoadBalancer # If you want to expose it externally, like in cloud environments**

**IN KUBERNTES EXPOSING THE PORT**

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl apply -f pod-expose.yml

pod/nginx-3 created

PS E:\CKA BY zeal vora\PracticeME\section2> kubectl get pods

NAME READY STATUS RESTARTS AGE

C:\Windows\system32>kubectl describe pods nginx-3

Name: nginx-3

Namespace: default

C:\Windows\system32>kubectl explain pod.spec.

KIND: Pod

VERSION: v1

FIELD: spec <PodSpec>

PodSpec is a description of a pod.

**Lablels and selectors**

**Used to identify information reosurces in aws ,** like instance name through we can access to it is known as labels , using label we can specify the env using pods like which one and what is doing , like dev and production env and using selector we can select the env

Pod1 - env:dev

Pod2 – env:dev using label specify the env

Selector will look into the labels which are associated with the objects and all the objects which do

not have the environment of prod.

Will be removed from the list.

This is very similar here **where you have a pod and you have a label** where N is equal to dev.

So that is what labels are.

And through selectors you will be able to filter the Kubernetes objects depending on the labels which

are associated with them.

So in one of the organizations where I had recently joined what they used to do, they never had any

of the tags or any of the labels which are associated with the servers there.

So it was very difficult to find which server belonged to which environment.

And in one of the email thread, one of the developer requested the team member to go ahead and delete

the development EC2 instance and what the team member did.

He deleted the EC2 instance which belonged to the staging environment and all of the QA team came running

at the DevOps team asking why did they delete the staging environment?

And this was one of the reason, the reason why this mistake happened was because there was no label

which was associated with the resources in AWS.

And this is the reason why having labels is extremely important not only for the Kubernetes objects,

but also for the non Kubernetes environments like AWS.

Microsoft Windows [Version 10.0.19045.5440]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

C:\Windows\system32>kubectl get pods -l env=prod

No resources found in default namespace.

C:\Windows\system32>kubectl describe pod pod\_name

C:\Windows\system32>kubectl run pod-1 --image=nginx

pod/pod-1 created

C:\Windows\system32>kubectl run pod-2 --image=nginx

pod/pod-2 created

C:\Windows\system32>kubectl run pod-3 --image=nginx

pod/pod-3 created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

pod-1 1/1 Running 0 93s

pod-2 1/1 Running 0 54s

pod-3 1/1 Running 0 40s

C:\Windows\system32>kubectl get pods --show=labels

error: unknown flag: --show

See 'kubectl get --help' for usage.

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

pod-1 1/1 Running 0 2m1s run=pod-1

pod-2 1/1 Running 0 82s run=pod-2

pod-3 1/1 Running 0 68s run=pod-3

**# Attaching label to pods in different pod**

C:\Windows\system32>kubectl label pod pod-1 env=dev

pod/pod-1 labeled

C:\Windows\system32>

C:\Windows\system32>kubectl label pod pod-2 env=stage

pod/pod-2 labeled

C:\Windows\system32>kubectl label pod pod-3 env=prod

pod/pod-3 labeled

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

pod-1 1/1 Running 0 5m42s env=dev,run=pod-1

pod-2 1/1 Running 0 5m3s env=stage,run=pod-2

pod-3 1/1 Running 0 4m49s env=prod,run=pod-3

**#selector select a pods based on specify label or filter the data**

C:\Windows\system32>kubectl get pods -l env=dev

NAME READY STATUS RESTARTS AGE

pod-1 1/1 Running 0 10m

# get pods whose env is dev

C:\Windows\system32>kubectl get pods -l env=prod

NAME READY STATUS RESTARTS AGE

pod-3 1/1 Running 0 10m

# accessing pod with label env = dev with selector

C:\Windows\system32>kubectl get pods -l env!=dev

NAME READY STATUS RESTARTS AGE

pod-2 1/1 Running 0 11m

pod-3 1/1 Running 0 11m

C:\Windows\system32>kubectl label --help

Update the labels on a resource.

\* A label key and value must begin with a letter or number, and may contain letters, numbers, hyphens, dots, and

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

pod-1 1/1 Running 0 14m env=dev,run=pod-1

pod-2 1/1 Running 0 14m env=stage,run=pod-2

pod-3 1/1 Running 0 14m env=prod,run=pod-3

C:\Windows\system32>kubectl run nginx --image=nginx --dry-run=client -o yaml

C:\Windows\system32>kubectl run nginx --image=nginx --dry-run=client -o yaml > label-pod.yaml

C:\Windows\system32>notepad label-pod.yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: nginx

name: nginx

spec:

containers:

- image: nginx

name: nginx

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

label-pod.yaml

C:\Windows\system32>kubectl apply -f lable-pod.yaml

error: the path "lable-pod.yaml" does not exist

C:\Windows\system32>kubectl apply -f label-pod.yaml

pod/nginx created

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx 1/1 Running 0 32s env=dev,run=nginx

pod-1 1/1 Running 0 21m env=dev,run=pod-1

pod-2 1/1 Running 0 21m env=stage,run=pod-2

pod-3 1/1 Running 0 21m env=prod,run=pod-3

C:\Windows\system32>kubectl label --help

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx 1/1 Running 0 2m41s env=dev,run=nginx

pod-1 1/1 Running 0 24m env=dev,run=pod-1

pod-2 1/1 Running 0 23m env=stage,run=pod-2

pod-3 1/1 Running 0 23m env=prod,run=pod-3

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx 1/1 Running 0 2m41s env=dev,run=nginx

pod-1 1/1 Running 0 24m env=dev,run=pod-1

pod-2 1/1 Running 0 23m env=stage,run=pod-2

pod-3 1/1 Running 0 23m env=prod,run=pod-3

C:\Windows\system32>kubectl label pods --all status=running

pod/nginx labeled

pod/pod-1 labeled

pod/pod-2 labeled

pod/pod-3 labeled

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

nginx 1/1 Running 0 5m9s env=dev,run=nginx,status=running

pod-1 1/1 Running 0 26m env=dev,run=pod-1,status=running

pod-2 1/1 Running 0 25m env=stage,run=pod-2,status=running

pod-3 1/1 Running 0 25m env=prod,run=pod-3,status=running

C:\Windows\system32>kubectl delete pods --all

pod "nginx" deleted

pod "pod-1" deleted

pod "pod-2" deleted

pod "pod-3" deleted

**# label associate with pods**

C:\Windows\system32>

Label-pod.yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: nginx

env: dev

name: nginx

spec:

containers:

- image: nginx

name: nginx

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

**$ replica set ensure right no of pods are always running , if anyone pod delete by mistake then replica set automraticcly create another pods**

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: frontend-replicaset

spec:

  replicas: 3

  selector:

    matchLabels:

      tier: frontend

  template:

    metadata:

      labels:

        tier: frontend

    spec:

      containers:

      - name: php-redis

        image: us-docker.pkg.dev/google-samples/containers/gke/gb-frontend:v5

apiversion: apps/v1

kind: Replicaset

metadata:

name: frontend-replicaset

spec:

replicas: 3

selector:

**$labels are associated with each of the pods , pod create by replica set will have label tier frontend**

**Label used to identify objects**

**Selector:**

A selector is used by the ReplicaSet to **identify which Pods it should manage**.

yaml

CopyEdit

selector:

matchLabels:

tier: frontend

This tells the ReplicaSet to look for **all Pods that have the label tier=frontend** and ensure the desired number of replicas (3 in this case) are running.

### ****How They Work Together:****

* The **ReplicaSet** creates Pods with the label tier: frontend (defined in template.metadata.labels).
* The **selector** (matchLabels: tier=frontend) ensures the ReplicaSet manages only these Pods.
* If a Pod with this label is deleted, the ReplicaSet will automatically create a new one to maintain the **desired replica count**.

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f replica-set.yml

error: the path "replica-set.yml" does not exist

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f replica-set.yaml

replicaset.apps/frontend-replicaset created

C:\Windows\system32>kubectl get replica-set

error: the server doesn't have a resource type "replica-set"

C:\Windows\system32>kubectl get replicaset

NAME DESIRED CURRENT READY AGE

frontend-replicaset 3 3 0 48s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-qw9gv 0/1 ContainerCreating 0 59s

frontend-replicaset-sh4xc 0/1 ContainerCreating 0 59s

frontend-replicaset-vg8fq 0/1 ContainerCreating 0 59s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-qw9gv 1/1 Running 0 3m29s

frontend-replicaset-sh4xc 1/1 Running 0 3m29s

frontend-replicaset-vg8fq 1/1 Running 0 3m29s

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

frontend-replicaset-qw9gv 1/1 Running 0 3m48s tier=frontend

frontend-replicaset-sh4xc 1/1 Running 0 3m48s tier=frontend

frontend-replicaset-vg8fq 1/1 Running 0 3m48s tier=frontend

$ same label are associated with pods every time

For example, when you run kubectl get pods show labels, you see all of them has this tier is equal

to front end which is this specific label.

And the next part is the selector.

Now as we know replica set manages pods.

It can scale up the pod.

It can scale down the pod.

Now how will replica set know that these are the pods that it is managing.

For example, how will replica set know that maybe out of 1000 pods, these are the three pods that

it is managing.

So it does that using the labels.

And this is the reason why you also have the selectors.

So here what we are seeing is replica set should use this specific selector to identify the pods that are being created through replica set, YAML file and whatever pods that you see with this label replica set can work on that.

Maybe as part of scaling up, maybe as part of scaling down and so on.

C:\Windows\system32>kubectl scale --replicas=5 rs/frontend-replicaset

replicaset.apps/frontend-replicaset scaled

# we are scalind our replica sets , it ensure 5 pods are running eveyr time , rs for replica set and fronetne-rep are name of replic set that need to be scale

C:\Windows\system32>kubectl scale --replicas=5 rd/frontend-replicaset

error: the server doesn't have a resource type "rd"

C:\Windows\system32>kubectl scale --replicas=5 rs/frontend-replicaset

replicaset.apps/frontend-replicaset scaled

C:\Windows\system32>kubectl get replicaset

NAME DESIRED CURRENT READY AGE

frontend-replicaset 5 5 5 11m

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-9fgp5 1/1 Running 0 108s

frontend-replicaset-9gbqd 1/1 Running 0 108s

frontend-replicaset-qw9gv 1/1 Running 0 11m

frontend-replicaset-sh4xc 1/1 Running 0 11m

frontend-replicaset-vg8fq 1/1 Running 0 11m

C:\Windows\system32>kubectl scale --replicas=1 rs/frontend-replicaset

replicaset.apps/frontend-replicaset scaled

# Replica set not update existing pods ,

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: webserver-replicaset

spec:

  replicas: 3

  selector:

    matchLabels:

      app: webserver

  template:

    metadata:

      labels:

        app: webserver

    spec:

      containers:

      - name: nginx-container

        image: nginx

rs.yaml

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f rs.yaml

replicaset.apps/webserver-replicaset created

**# replica set not hande regurlar update and cant handle daily challenges**

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: webserver-replicaset

spec:

  replicas: 3

  selector:

    matchLabels:

      app: webserver

  template:

    metadata:

      labels:

        app: webserver

    spec:

      containers:

      - name: nginx-container

        image: httpd

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f rs.yaml

replicaset.apps/webserver-replicaset configured

PS D:\CKA BY zeal vora\PracticeME\section3>

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 20m

webserver-replicaset-687nr 1/1 Running 0 2m48s

webserver-replicaset-7tdrt 1/1 Running 0 2m48s

webserver-replicaset-vf8fv 1/1 Running 0 2m48s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 21m

webserver-replicaset-687nr 1/1 Running 0 3m53s

webserver-replicaset-7tdrt 1/1 Running 0 3m53s

webserver-replicaset-vf8fv 1/1 Running 0 3m53s

C:\Windows\system32>kubectl describe pod webserver-replicaset.yaml

Error from server (NotFound): pods "webserver-replicaset.yaml" not found

C:\Windows\system32>kubectl describe pod webserver-replicaset-687nr

Name: webserver-replicaset-687nr

Namespace: default

Priority: 0

Service Account: default

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 25m

webserver-replicaset-687nr 1/1 Running 0 7m33s

webserver-replicaset-7tdrt 1/1 Running 0 7m33s

webserver-replicaset-vf8fv 1/1 Running 0 7m33s

C:\Windows\system32>kubect delete pods webserver-replicaset-7tdrt

'kubect' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>kubectl delete pods webserver-replicaset-7tdrt

pod "webserver-replicaset-7tdrt" deleted

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 25m

webserver-replicaset-687nr 1/1 Running 0 8m6s

webserver-replicaset-hn2qs 0/1 ContainerCreating 0 8s

webserver-replicaset-vf8fv 1/1 Running 0 8m6s

C:\Windows\system32>kubectl describe webserver-replicaset-hn2qs

error: the server doesn't have a resource type "webserver-replicaset-hn2qs"

C:\Windows\system32>kubectl describe webserver-replicaset-hn2qs

error: the server doesn't have a resource type "webserver-replicaset-hn2qs"

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 26m

webserver-replicaset-687nr 1/1 Running 0 9m14s

webserver-replicaset-hn2qs 1/1 Running 0 76s

webserver-replicaset-vf8fv 1/1 Running 0 9m14s

C:\Windows\system32>kubectl describe webserver-replicaset-hn2qs

error: the server doesn't have a resource type "webserver-replicaset-hn2qs"

C:\Windows\system32>kubectl describe pod webserver-replicaset-hn2qs

**#new pod based on latest image or newer configrualtuon in yaml file**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 28m

webserver-replicaset-687nr 1/1 Running 0 11m

webserver-replicaset-hn2qs 1/1 Running 0 3m23s

webserver-replicaset-vf8fv 1/1 Running 0 11m

C:\Windows\system32>kubectl scale rs/webserver-replicaset --replicas=0

replicaset.apps/webserver-replicaset scaled

# we only need 0 pod , we don’t need 3 pod

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-vg8fq 1/1 Running 0 30m

C:\Windows\system32>kubectl delete pods frontend-replicaset-vg8fq

pod "frontend-replicaset-vg8fq" deleted

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

frontend-replicaset-jb57v 1/1 Running 0 9s

C:\Windows\system32>kubectl scale rs/frontend-replicaset --replicas=0

replicaset.apps/frontend-replicaset scaled

C:\Windows\system32>kubetctl get pods

'kubetctl' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

C:\Windows\system32>kubectl scale rs/webserver-replicaset --replicas=3

replicaset.apps/webserver-replicaset scaled

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

webserver-replicaset-hg6mm 0/1 ContainerCreating 0 5s

webserver-replicaset-k74vr 0/1 ContainerCreating 0 5s

webserver-replicaset-sqxcx 0/1 ContainerCreating 0 5s

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f rs.yaml

replicaset.apps "webserver-replicaset" deleted

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl get pods

No resources found in default namespace.

**# replica set use selector to specify what pods it need to manage and selector use labels to identify which pod it should manage**

These label assocaieted with pod spec that;w why imp

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: webserver-replicaset

spec:

  replicas: 3

  selector:

    matchLabels:

      app: webserver

  template:

    metadata:

      labels:

        app: webserver

    spec:

      containers:

      - name: nginx-container

        image: httpd

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f rs.yaml

replicaset.apps/webserver-replicaset created

C:\Windows\system32>kubectl run extrnal-pod --image=nginx

pod/extrnal-pod created

C:\Windows\system32>kubectl label pod extrnal-pod app=webserver

pod/extrnal-pod labeled

C:\Windows\system32>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

extrnal-pod 1/1 Running 0 63s app=webserver,run=extrnal-pod

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

extrnal-pod 1/1 Running 0 2m6s

webserver-replicaset-dwshn 1/1 Running 0 26s

webserver-replicaset-jx29b 1/1 Running 0 26s

**# replica set start managing external pod as it have same label like webserver replicaset**

**#those pod which has same label so replica set manage them**

C:\Windows\system32>kubectl get rs

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 43m

webserver-replicaset 3 3 3 2m24s

$ so we have these replica sets ,2 from web server and one is external pod

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f rs.yaml

replicaset.apps "webserver-replicaset" deleted - our extnal pod also get dleted

**# so keep careful while you name label and selector in replica set**

**Deployments**

apiVersion: apps/v1

kind: ReplicaSet

metadata:

  name: webserver-replicaset

spec:

  replicas: 1

  selector:

    matchLabels:

      app: webserver

  template:

    metadata:

      labels:

        app: webserver

    spec:

      containers:

      - name: nginx-container

        image: httpd

rs.yaml

#label are key value pair

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx  # Fixed the typo

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:latest

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f rs.yaml

replicaset.apps/webserver-replicaset created

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f deployment.yaml

The Deployment "nginx-deployment" is invalid: spec.template.metadata.labels: Invalid value: map[string]string{"app":"nginx"}: `selector` does not match template `labels`

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f deployment.yaml

deployment.apps/nginx-deployment created

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-c95765fd4-kr9wk 1/1 Running 0 33s

webserver-replicaset-gq9ph 1/1 Running 0 2m2s

webserver-replicaset-pmmg9 1/1 Running 0 2m2s

webserver-replicaset-wgswd 1/1 Running 0 2m2s

C:\Windows\system32>kubectl get replicaset

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 22h

nginx-deployment-c95765fd4 1 1 1 82s

webserver-replicaset 3 3 3 2m51s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-54f7d6bc76-tsbp9 1/1 Running 0 21s

webserver-replicaset-gq9ph 1/1 Running 0 7m48s

**$ so webserver replica set not updated even after chahingin image**

C:\Windows\system32>kubectl get replicaset

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 22h

nginx-deployment-c95765fd4 1 1 1 82s

webserver-replicaset 3 3 3 2m51s

C:\Windows\system32>kubectl get deployments

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployment 1/1 1 1 6m14s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-54f7d6bc76-tsbp9 1/1 Running 0 21s

webserver-replicaset-gq9ph 1/1 Running 0 7m48s

C:\Windows\system32>kubectl describe pods

Name: nginx-deployment-54f7d6bc76-tsbp9

Namespace: default

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f rs.yaml

replicaset.apps "webserver-replicaset" deleted

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f deployment.yaml

deployment.apps "nginx-deployment" deleted

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx  # Fixed the typo

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: httpd:latest

deploy.yaml

kubectl apply –f deployment.yaml

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx  # Fixed the typo

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:latest

**#change image new pod is careted with new configyrations**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-54f7d6bc76-b4xrs 1/1 Running 0 2m17s

nginx-deployment-c95765fd4-t82nw 0/1 ContainerCreating 0 6s

**$ old pods is terminated and new pod is created that**;s part of rolling update

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-54f7d6bc76-b4xrs 1/1 Running 0 2m17s

nginx-deployment-c95765fd4-t82nw 0/1 ContainerCreating 0 6s

C:\Windows\system32>kubectl get pods -w

NAME READY STATUS RESTARTS AGE

nginx-deployment-c95765fd4-t82nw 1/1 Running 0 2m21s

C:\Windows\system32>kubectl describe pods nginx-deployment-c95765fd4-t82nw

Name: nginx-deployment-c95765fd4-t82nw

Namespace: default

**# pod with ngninx image update successfully witg deployemts**

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

spec:

  replicas: 1

  selector:

    matchLabels:

      app: nginx  # Fixed the typo

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx:1.2.3

deployment.yaml

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f deployment.yaml

deployment.apps/nginx-deployment configured

C:\Windows\system32>kubectl rollout history deployment/nginx-deployment

deployment.apps/nginx-deployment

REVISION CHANGE-CAUSE

1 <none>

2 <none>

3 <none>

**# revision or changes 3 reviesorns are there , using deployemtn if we do any change in manifest file then these changes are reflected in pods as well**

C:\Windows\system32>kubectl get replicaset

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 22h

nginx-deployment-54f7d6bc76 0 0 0 10m

nginx-deployment-66bcc6cdb5 1 1 0 119s

nginx-deployment-c95765fd4 1 1 1 8m2s

**# we can easily rollback to previous changes at any time**

C:\Windows\system32>kubectl create deployemtn nginx-deployment --image=nginx

C:\Windows\system32>kubectl get deployment

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployemnt 1/1 1 1 2m17s

C:\Windows\system32>kubectl get rs

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 23h

nginx-deployemnt-86f574f7f5 1 1 1 2m22s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

s

C:\Windows\system32>kubectl create deployment nginx-deployment --image=nginx --dry-run=client -o yaml > deployment.yaml

C:\Windows\system32>notepad deployment.yaml

# create a deployment using CLI

apiVersion: apps/v1

kind: Deployment

metadata:

labels:

app: nginx-deployment

name: nginx-deployment

spec:

replicas: 1

selector:

matchLabels:

app: nginx-deployment

template:

metadata:

labels:

app: nginx-deployment

spec:

containers:

- image: nginx

name: nginx

deployment.yaml

C:\Windows\system32>kubectl get deployment

NAME READY UP-TO-DATE AVAILABLE AGE

nginx-deployemnt 1/1 1 1 10m

C:\Windows\system32>kubectl rollout history deployment nginx-deployemnt

deployment.apps/nginx-deployemnt

REVISION CHANGE-CAUSE

1. <none>

C:\Windows\system32>kubectl set image --help

Update existing container image(s) of resources.

C:\Windows\system32>kubectl get rs

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 23h

nginx-deployemnt-86f574f7f5 0 0 0 15m

nginx-deployemnt-fc76dc788 1 1 1 2m9s

C:\Windows\system32>kubectl describe rs

C:\Windows\system32>kubectl describe rs nginx-deployemnt-86f574f7f5

Name: nginx-deployemnt-86f574f7f5

Namespace: default

C:\Windows\system32>kubectl get rs

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 23h

nginx-deployemnt-86f574f7f5 0 0 0 18m

nginx-deployemnt-fc76dc788 1 1 1 4m30s

C:\Windows\system32>kubectl rollout undo deployment nginx-deployemnt

deployment.apps/nginx-deployemnt rolled back

C:\Windows\system32>kubectl get rs

NAME DESIRED CURRENT READY AGE

frontend-replicaset 0 0 0 23h

nginx-deployemnt-86f574f7f5 1 1 1 19m

nginx-deployemnt-fc76dc788 0 0 0 5m30s

C:\Windows\system32>kubectl rollout undo --help

Roll back to a previous rollout.

Examples:

# Roll back to the previous deployment

kubectl rollout undo deployment/abc

# Roll back to daemonset revision 3

kubectl rollout undo daemonset/abc --to-revision=3

# Roll back to the previous deployment with dry-run

kubectl rollout undo --dry-run=server deployment/abc

C:\Windows\system32>kubectl rollout history deployment nginx-deployemnt

deployment.apps/nginx-deployemnt

REVISION CHANGE-CAUSE

2 <none>

1. <none>

C:\Windows\system32>kubectl rollout undo deployment nginx-deployemnt --to-revision=2

deployment.apps/nginx-deployemnt rolled back

**$ now we are at version2 that we have done changes**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployemnt-fc76dc788-wdzc2 1/1 Running 0 72s

C:\Windows\system32>kubectl scale --replicas=3 deployment nginx-deployment

error: no objects passed to scale

C:\Windows\system32>kubectl scale --replicas=3 deployment nginx-deployemnt

deployment.apps/nginx-deployemnt scaled

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployemnt-fc76dc788-wdzc2 1/1 Running 0 2m24s

nginx-deployemnt-fc76dc788-zjn4v 0/1 ContainerCreating 0 4s

nginx-deployemnt-fc76dc788-zn2zr 0/1 ContainerCreating 0 4s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployemnt-fc76dc788-wdzc2 1/1 Running 0 2m54s

nginx-deployemnt-fc76dc788-zjn4v 1/1 Running 0 34s

nginx-deployemnt-fc76dc788-zn2zr 1/1 Running 0 34s

C:\Windows\system32>kubectl delete deployment nginx-deployemnt

deployment.apps "nginx-deployemnt" deleted

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

C:\Windows\system32>kubectl get deployement

error: the server doesn't have a resource type "deployement"

#Multiple work nodes – to distribute workloads we can use in prodn even

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

nginx-deployment-5959b5b5c9-4hcz6 1/1 Running 0 16s 10.1.0.45 docker-desktop <none> <none>

nginx-deployment-5959b5b5c9-92mgx 1/1 Running 0 16s 10.1.0.46 docker-desktop <none> <none>

nginx-deployment-5959b5b5c9-g2pz7 1/1 Running 0 16s 10.1.0.47 docker-desktop <none> <none>

**# our pod run inside node and node is run inside docker desktop**

C:\Windows\system32>kubectl delete deployment nginx-deployment

deployment.apps "nginx-deployment" deleted

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

Each pod is scheduled in one worker node.

So in total three pods are scheduled across all of the three worker nodes over here.

Now, the next pointer that we were discussing is that if one worker node fails, then the application

can become unavailable.

# if one workder node fail control panel will recreate pod in another workder node that are avalibel

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 27h v1.31.4

**#one weorker node is running - maybe docker desktop**

**# pod run insidie worker nodes**

**Node selector** –

node for specific port to run we can choose approcaite work node to run our pods

Using node selector

On which node we want to run our pods we can use node selector for it

C:\Windows\system32>kubectl create deployment tmp-deploy --image=nginx --replicas=3

deployment.apps/tmp-deploy created

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

tmp-deploy-568c7959d6-4nsxv 1/1 Running 0 15s 10.1.0.50 docker-desktop <none> <none>

tmp-deploy-568c7959d6-wxmqw 1/1 Running 0 15s 10.1.0.49 docker-desktop <none> <none>

tmp-deploy-568c7959d6-z4mwd 1/1 Running 0 15s 10.1.0.48 docker-desktop <none> <none>

**$ these pods are running in Docker-desktop which is our worker-nodes ,which pod run in which worker node we can use node selector for it**

C:\Windows\system32>kubectl create deployment tmp-deploy --image=nginx --replicas=3

deployment.apps/tmp-deploy created

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

tmp-deploy-568c7959d6-4nsxv 1/1 Running 0 15s 10.1.0.50 docker-desktop <none> <none>

tmp-deploy-568c7959d6-wxmqw 1/1 Running 0 15s 10.1.0.49 docker-desktop <none> <none>

tmp-deploy-568c7959d6-z4mwd 1/1 Running 0 15s 10.1.0.48 docker-desktop <none> <none>

C:\Windows\system32>kubectl delete deployment

C:\Windows\system32>kubectl delete deployment tmp-deploy

deployment.apps "tmp-deploy" deleted

C:\Windows\system32>kubectl get deployment

No resources found in default namespace.

apiVersion: v1

kind: Pod

metadata:

  name: kplabs-pod

spec:

  nodeSelector:

    size: Large

  containers:

  - name: nginx-pod

    image: nginx

nodeSelector-pod.yaml

apiVersion: v1

kind: Pod

metadata:

  name: nginx-deployment

spec:

  nodeSelector:

    size: medium

  containers:

  - name: nginx-pod

    image: nginx

nodeSelector-deployment.yaml

**pod goes in worker node whose size is medium here**

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 0/1 Pending 0 74s <none> <none> <none> <none>

C:\Windows\system32>kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 0/1 Pending 0 107s <none> <none> <none> <none>

C:\Windows\system32>kubectl get pod -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 0/1 Pending 0 2m56s <none> <none> <none> <none>

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

kplabs-pod 0/1 Pending 0 3m33s

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 0/1 Pending 0 4m47s <none> <none> <none> <none>

**# we don’t have any worker nodes whose size is large always**

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeSelector-pod.yaml

pod "kplabs-pod" deleted

apiVersion: apps/v1

kind: Deployment

metadata:

  name: nginx-deployment

spec:

  replicas: 3

  selector:

    matchLabels:

      app: nginx

  template:

    metadata:

      labels:

        app: nginx

    spec:

      nodeSelector:

        size: medium

      containers:

      - name: nginx-pod

        image: nginx

node-selector-pod.yaml

**# we will place the pod in worker node whose size is large**

**# place in worker node whose size is large , place pod in worker node**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

nginx-deployment-5c65fb4686-5w975 0/1 Pending 0 28s

nginx-deployment-5c65fb4686-f8hpm 0/1 Pending 0 28s

nginx-deployment-5c65fb4686-gr5d5 0/1 Pending 0 28s

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

nginx-deployment-5c65fb4686-5w975 0/1 Pending 0 39s <none> <none> <none> <none>

nginx-deployment-5c65fb4686-f8hpm 0/1 Pending 0 39s <none> <none> <none> <none>

nginx-deployment-5c65fb4686-gr5d5 0/1 Pending 0 39s <none> <none> <none> <none>

**$ use label to specify size**

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeSelector-deployment.yaml

deployment.apps "nginx-deployment" deleted

apiVersion: v1

kind: Pod

metadata:

  name: kplabs-pod

spec:

  nodeSelector:

    size: Large

  containers:

  - name: service-pod

    image: nginx

# place in worker node whose size is large , using node selector

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f nodeSelector-pod.yaml

pod/kplabs-pod created

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 28h v1.31.4

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 28h v1.31.4

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 0/1 Pending 0 17s <none> <none> <none> <none>

C:\Windows\system32>kubectl describe pod kplabs-pod

Name: kplabs-pod

Namespace: default

Priority: 0

Service Account: default

Node: <none>

Labels: <none>

Annotations: <none>

Status: Pending

IP:

IPs: <none>

Containers:

service-pod:

Image: nginx

Port: <none>

Host Port: <none>

Environment: <none>

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-58kb7 (ro)

Conditions:

Type Status

PodScheduled False

Volumes:

kube-api-access-58kb7:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

ConfigMapOptional: <nil>

DownwardAPI: true

QoS Class: BestEffort

Node-Selectors: size=Large

Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s

node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Warning FailedScheduling 2m17s default-scheduler 0/1 nodes are available: 1 node(s) didn't match Pod's node affinity/selector. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.

# pod don’t match with the selector large that we have , size == large we have

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 28h v1.31.4

C:\Windows\system32>kubectl label node docker-desktop size=large

node/docker-desktop labeled

**$after labelling worker node now pod is in running state**

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

Roles: control-plane

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

kubernetes.io/arch=amd64

kubernetes.io/hostname=docker-desktop

kubernetes.io/os=linux

node-role.kubernetes.io/control-plane=

node.kubernetes.io/exclude-from-external-load-balancers=

size=large

Annotations: kubeadm.alpha.kubernetes.io/cri-socket: unix:///var/run/cri-dockerd.sock

node.alpha.kubernetes.io/ttl: 0

volumes.kubernetes.io/controller-managed-attach-detach: true

CreationTimestamp: Tue, 11 Feb 2025 19:15:55 +0530

Taints: <none>

Unschedulable: false

Lease:

HolderIdentity: docker-desktop

AcquireTime: <unset>

RenewTime: Thu, 13 Feb 2025 00:10:07 +0530

Conditions:

Type Status LastHeartbeatTime LastTransitionTime Reason Message

---- ------ ----------------- ------------------ ------ -------

MemoryPressure False Thu, 13 Feb 2025 00:10:05 +0530 Tue, 11 Feb 2025 19:15:54 +0530 KubeletHasSufficientMemory kubelet has sufficient memory available

DiskPressure False Thu, 13 Feb 2025 00:10:05 +0530 Tue, 11 Feb 2025 19:15:54 +0530 KubeletHasNoDiskPressure kubelet has no disk pressure

PIDPressure False Thu, 13 Feb 2025 00:10:05 +0530 Tue, 11 Feb 2025 19:15:54 +0530 KubeletHasSufficientPID kubelet has sufficient PID available

Ready True Thu, 13 Feb 2025 00:10:05 +0530 Tue, 11 Feb 2025 19:15:55 +0530 KubeletReady kubelet is posting ready status

Addresses:

InternalIP: 192.168.65.3

Hostname: docker-desktop

Capacity:

cpu: 4

ephemeral-storage: 1055762868Ki

hugepages-1Gi: 0

hugepages-2Mi: 0

memory: 8080248Ki

pods: 110

Allocatable:

cpu: 4

ephemeral-storage: 972991057538

hugepages-1Gi: 0

hugepages-2Mi: 0

memory: 7977848Ki

pods: 110

System Info:

Machine ID: 2b077ff7-f5d8-41b7-8cfa-5d91b59e0f45

System UUID: 2b077ff7-f5d8-41b7-8cfa-5d91b59e0f45

Boot ID: 9947d690-4310-4381-801e-449e9f975ea8

Kernel Version: 5.15.167.4-microsoft-standard-WSL2

OS Image: Docker Desktop

Operating System: linux

Architecture: amd64

Container Runtime Version: docker://27.5.1

Kubelet Version: v1.31.4

Kube-Proxy Version: v1.31.4

Non-terminated Pods: (9 in total)

Namespace Name CPU Requests CPU Limits Memory Requests Memory Limits Age

--------- ---- ------------ ---------- --------------- ------------- ---

kube-system coredns-7c65d6cfc9-gklnm 100m (2%) 0 (0%) 70Mi (0%) 170Mi (2%) 28h

kube-system coredns-7c65d6cfc9-whm2v 100m (2%) 0 (0%) 70Mi (0%) 170Mi (2%) 28h

kube-system etcd-docker-desktop 100m (2%) 0 (0%) 100Mi (1%) 0 (0%) 28h

kube-system kube-apiserver-docker-desktop 250m (6%) 0 (0%) 0 (0%) 0 (0%) 28h

kube-system kube-controller-manager-docker-desktop 200m (5%) 0 (0%) 0 (0%) 0 (0%) 28h

kube-system kube-proxy-zphbm 0 (0%) 0 (0%) 0 (0%) 0 (0%) 28h

kube-system kube-scheduler-docker-desktop 100m (2%) 0 (0%) 0 (0%) 0 (0%) 28h

kube-system storage-provisioner 0 (0%) 0 (0%) 0 (0%) 0 (0%) 28h

kube-system vpnkit-controller 0 (0%) 0 (0%) 0 (0%) 0 (0%) 28h

Allocated resources:

(Total limits may be over 100 percent, i.e., overcommitted.)

Resource Requests Limits

-------- -------- ------

cpu 850m (21%) 0 (0%)

memory 240Mi (3%) 340Mi (4%)

ephemeral-storage 0 (0%) 0 (0%)

hugepages-1Gi 0 (0%) 0 (0%)

hugepages-2Mi 0 (0%) 0 (0%)

Events: <none>

**$ specify the label with our nodes**

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeSelector-pod.yaml

pod "kplabs-pod" deleted

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 28h v1.31.4

C:\Windows\system32>kubectl label node docker-desktop size=large

node/docker-desktop labeled

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

cpu 850m (21%) 0 (0%)

memory 240Mi (3%) 340Mi (4%)

ephemeral-storage 0 (0%) 0 (0%)

hugepages-1Gi 0 (0%) 0 (0%)

hugepages-2Mi 0 (0%) 0 (0%)

Events: <none>

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

kplabs-pod 0/1 Pending 0 6m58s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

kplabs-pod 0/1 Pending 0 7m24s

C:\Windows\system32>kubectl get nodes --show-labels

NAME STATUS ROLES AGE VERSION LABELS

docker-desktop Ready control-plane 28h v1.31.4 beta.kubernetes.io/arch=amd64,beta.kubernetes.io/os=linux,kubernetes.io/arch=amd64,kubernetes.io/hostname=docker-desktop,kubernetes.io/os=linux,node-role.kubernetes.io/control-plane=,node.kubernetes.io/exclude-from-external-load-balancers=,size=large

apiVersion: v1

kind: Pod

metadata:

  name: kplabs-pod

spec:

  nodeSelector:

    size: large  # Changed "Large" to "large"

  containers:

  - name: service-pod

    image: nginx

$ after this pod has succesffuly label and in the running state

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f nodeSelector-pod.yaml

pod/kplabs-pod created

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

kplabs-pod 1/1 Running 0 28s 10.1.0.51 docker-desktop <none> <none>

**# our pod is succeflly attach to large worker node**

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeSelector-pod.yaml

pod "kplabs-pod" deleted

C:\Windows\system32>kubectl label node docker-desktop size-

node/docker-desktop unlabeled

$ node docker has been un labelled

C:\Windows\system32>

**# Daemon sets**

apiVersion: apps/v1

kind: DaemonSet

metadata:

  name: anti-virus

spec:

  selector:

    matchLabels:

      app: nginx

  template:

    metadata:

      labels:

        app: nginx

    spec:

      containers:

      - name: nginx

        image: nginx

daemonset.yaml

So in total now we will have three worker nodes that will be available.

So once you have three worker node we will see on how Daemonset automatically launches one pod in each worker node.

So in total two more additional pods for two additional worker nodes that are getting created.

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f daemonset.yaml

daemonset.apps/anti-virus created

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 29h v1.31.4

C:\Windows\system32>kubectl get daemonset

No resources found in default namespace.

C:\Windows\system32>kubectl get deamonset

error: the server doesn't have a resource type "deamonset"

C:\Windows\system32>kubectl get daemonset

NAME DESIRED CURRENT READY UP-TO-DATE AVAILABLE NODE SELECTOR AGE

anti-virus 1 1 1 1 1 <none> 32s

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-wks8s 1/1 Running 0 43s

**#dameon set autmatilly create our pod in different worker node**

# daemon-set use cases learning

**$ if we have there worker node and we create daemon set , so it will launch in eveyr worker node by default , we create it for only one worker node but it load in three worker node use in prodn environment**

**Node affinity**

@using node selector **we specify label and pod is sehecluede in those nodes**

In prodn we use node affinity

To run pod in which node we can decide using node affinity

**Require all cond need to satfiyt only will execute** and preffred only specific part will be executed

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 29h v1.31.4

C:\Windows\system32>kubectl describe docker-desktop

error: the server doesn't have a resource type "docker-desktop"

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

Roles: control-plane

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  affinity:

    nodeAffinity:

      requiredDuringSchedulingIgnoredDuringExecution:

        nodeSelectorTerms:

        - matchExpressions:

          - key: disktype

            operator: In

            values:

            - ssd

  containers:

  - name: nginx

    image: nginx

nodeaffintiy-require.yaml

**@worker nodes these things to be part as label**

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 29h v1.31.4

C:\Windows\system32>kubectl describe docker-desktop

error: the server doesn't have a resource type "docker-desktop"

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

Roles: control-plane

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f nodeAffnity-required.yaml

pod/nginx created

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-wks8s 1/1 Running 0 23m

nginx 0/1 Pending 0 10s

PS D:\CKA BY zeal vora\PracticeME\section3>

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-wks8s 1/1 Running 0 24m

nginx 0/1 Pending 0 51s

C:\Windows\system32>kubectl describe pod nginx

Name: nginx

Namespace: default

Priority: 0

Service Account: default

Node: <none>

Labels: <none>

Annotations: <none>

Status: Pending

IP:

IPs: <none>

**Warning FailedScheduling 58s default-scheduler 0/1 nodes are available: 1 node(s) didn't match Pod's node affinity/selector. preemption: 0/1 nodes are available: 1 Preemption is not helpful for scheduling.**

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete nodeAffnity-required.yaml

error: the server doesn't have a resource type "nodeAffnity-required"

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  affinity:

    nodeAffinity:

      preferredDuringSchedulingIgnoredDuringExecution:

      - weight: 1

        preference:

          matchExpressions:

          - key: disktype

            operator: In

            values:

            - ssd

  containers:

  - name: nginx

    image: nginx

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeAffnity-required.yaml

pod "nginx" deleted

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f nodeAffnity-preffred.yaml

pod/nginx created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-wks8s 1/1 Running 0 30m

nginx 1/1 Running 0 13s

# NOW pod is in running state after starify all codntion – in prefer – pod will run

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeAffnity-preffred.yaml

pod "nginx" deleted

apiVersion: v1

kind: Pod

metadata:

  name: nginx

spec:

  affinity:

    nodeAffinity:

      requiredDuringSchedulingIgnoredDuringExecution:

        nodeSelectorTerms:

        - matchExpressions:

          - key: disktype

            operator: In

            values:

            - ssd

  containers:

  - name: nginx

    image: nginx

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f nodeAffnity-required.yaml

pod/nginx created

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 29h v1.31.4

C:\Windows\system32>kubectl label node docker-desktop disktype=ssd

node/docker-desktop labeled

#setting the dikstype of docker-dekstop so that cond meet

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-wks8s 1/1 Running 0 34m

nginx 1/1 Running 0 2m4s

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f nodeAffnity-required.yaml

pod "nginx" deleted

**Requrired and limits**

Requyest menas guarantee chahiye hi chaiye itni mujhe , agar itni nahi mili toh pod run anhi hoga

Maximum - itne reuse kar sakte hai max memour isse jda nahi to control pod how much it consume memoru

Limit – max itni use kar sakte hai memory , isse jyda pod crahs , or request – itni memory chahiye run karne ka liye pod koo

apiVersion: v1

kind: Pod

metadata:

  name: kplabs-pod

spec:

  containers:

  - name: kplabs-container

    image: nginx

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f request-limit.yaml

pod/kplabs-pod created

**$ create a pod using manist file**

**$ limit and reques used in prodn eve so that they not affetct pod**

C:\Windows\system32>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 29h v1.31.4

C:\Windows\system32>kubectl describe docker-desktop

error: the server doesn't have a resource type "docker-desktop"

C:\Windows\system32>kubectl describe node docker-desktop

Name: docker-desktop

Roles: control-plane

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

apiVersion: v1

kind: Pod

metadata:

  name: kplabs-pod

spec:

  containers:

  - name: kplabs-container

    image: nginx

    resources:

      requests:

        memory: "128Mi"

        cpu: "0.1"

      limits:

        memory: "500Mi"

        cpu: "1"

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f reuqest-limit.yaml

error: the path "reuqest-limit.yaml" does not exist

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f request-limit.yaml

pod "kplabs-pod" deleted

PS D:\CKA BY zeal vora\PracticeME\section3> kubectl apply -f request-limit.yaml

pod/kplabs-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

anti-virus-vbqph 1/1 Running 0 5m39s

kplabs-pod 1/1 Running 0 11s

S D:\CKA BY zeal vora\PracticeME\section3> kubectl delete -f request-limit.yaml

pod "kplabs-pod" deleted

**# Service**

**Pod can communicate with each other**

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

C:\Windows\system32>kubectl get pods -o wide

No resources found in default namespace.

**# frontend pod communicate with bakcend pod**

C:\Windows\system32>kubectl exec -it frontend-pod -- bash

Error from server (NotFound): pods "frontend-pod" not found

$ BOTH POD CAN COMumucante with each other

**Service can distribute traffic to wide varity of pod s**

C:\Windows\system32>kubectl get services

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 41m

**$ throught service we can distribute traffic , we are sending traffic from frenotend pod to service and service sending traffic to bakcedn pod**

# service automatically send traffic to backend pod , from frontned we are making request to service

kubectl run backend-pod-3 --image=nginx --labels=type=webserver

C:\Windows\system32>kubectl run backend-pod --image=nginx

pod/backend-pod created

C:\Windows\system32>kubectl run frontend-pod --image=ubuntu --command -- sleep 36000

pod/frontend-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod 1/1 Running 0 69s

frontend-pod 1/1 Running 0 25s

C:\Windows\system32>kubectl exec -it frontend-pod -- bash

root@frontend-pod:/#

# conncet to the fronetne pod here

C:\Windows\system32>kubectl exec -it frontend-pod -- bash

root@frontend-pod:/# apt-get update && apt-get install curl -y

Terminal 2

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod 1/1 Running 0 6m38s

frontend-pod 1/1 Running 0 5m54s

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

backend-pod 1/1 Running 0 6m46s 10.244.0.3 minikube <none> <none>

frontend-pod 1/1 Running 0 6m2s 10.244.0.4 minikube <none> <none>

T-1

root@frontend-pod:/# curl 10.244.0.3

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

html { color-scheme: light dark; }

body { width: 35em; margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif; }

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.</p>

<p>For online documentation and support please refer to

<a href="http://nginx.org/">nginx.org</a>.<br/>

Commercial support is available at

<a href="http://nginx.com/">nginx.com</a>.</p>

**# frontend pod communicate with backend pod and we are making request from**

apiVersion: v1

kind: Service

metadata:

  name: simple-service

spec:

  ports:

  - port: 80

    targetPort: 80

service.yaml

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod 1/1 Running 0 20m

frontend-pod 1/1 Running 0 19m

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f service.yaml

service/simple-service created

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 73m

simple-service ClusterIP 10.100.246.207 <none> 80/TCP 23s

PS D:\CKA BY zeal vora\PracticeME\section4>

**# wehenever frontend pod want to communicate with backend pod it need to communicate through that ip address only**

$ through service we can setup communication between frontend and backend pod

C:\Windows\system32>kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 75m

simple-service ClusterIP 10.100.246.207 <none> 80/TCP 119s

C:\Windows\system32>kubectl describe service simple-service

Name: simple-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: <none>

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.246.207

IPs: 10.100.246.207

Port: <unset> 80/TCP

TargetPort: 80/TCP

Endpoints: <none>

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

**# backend are nothing but endpoints , endpoints receieve request from service , endpoints have ip address of backendpod**

So **that communication can setup easily between frontend pod to backend pod**

# we are sending request to service . and tehn service sending request to backend pod ,service has ip add of backedn pod .

C:\Windows\system32>kubectl exec -it frontend-pod -- bash

root@frontend-pod:/# curl 10.100.246.207

curl: (7) Failed to connect to 10.100.246.207 port 80 after 0 ms: Couldn't connect to server

$ fornend point connect to simple serveice but fails , we need to add enpoint manully

apiVersion: v1

kind: Endpoints

metadata:

  name: simple-service

subsets:

  - addresses:

      - ip: 10.244.0.4  # IP address of backend pod

    ports:

      - port: 80

Endpoints.yaml

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f endpoints.yaml

endpoints/simple-service created

C:\Windows\system32>kubectl describe service simple-service

Name: simple-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: <none>

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.246.207

IPs: 10.100.246.207

Port: <unset> 80/TCP

TargetPort: 80/TCP

Endpoints: 10.244.0.4:80

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

**$ now we have endpoints ip , if service get request then it forward to enpoints ip – in this way we can setup commn between frontend to backend pod**

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod 1/1 Running 0 38m

frontend-pod 1/1 Running 0 37m

C:\Windows\system32>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

backend-pod 1/1 Running 0 38m 10.244.0.3 minikube <none> <none>

frontend-pod 1/1 Running 0 37m 10.244.0.4 minikube <none> <none>

curl: (7) Failed to connect to 10.100.246.207 port 80 after 0 ms: Couldn't connect to server

root@frontend-pod:/# curl 10.244.0.4:80

curl: (7) Failed to connect to 10.244.0.4 port 80 after 0 ms: Couldn't connect to server

apiVersion: v1

kind: Endpoints

metadata:

  name: simple-service

subsets:

  - addresses:

      - ip: 10.244.0.3  # IP address of backend pod

    ports:

      - port: 80

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl delete endpoints simple-service

endpoints "simple-service" deleted

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl apply -f endpoints.yaml

endpoints/simple-service created

#if sending request

PS D:\CKA BY zeal vora\PracticeME\section4>

root@frontend-pod:/# curl 10.244.0.3:80

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

html { color-scheme: light dark; }

body { width: 35em; margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif; }

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.</p>

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl delete -f service.yaml

service "simple-service" deleted

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl get endpoints

NAME ENDPOINTS AGE

kubernetes 192.168.49.2:8443 95m

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl delete -f endpoints.yaml

Error from server (NotFound): error when deleting "endpoints.yaml": endpoints "simple-service" not found

apiVersion: v1

kind: Service

metadata:

  name: simple-service

spec:

  ports:

  - port: 8080

    targetPort: 80

service.yaml

#### port: 8080

* This is the **port on which the service is exposed**.
* Any external application or pod trying to access this service will do so via 8080.

#### targetPort: 80

* This is the **port on the container (inside the pod) where the application is running**.
* It means that traffic arriving at 8080 on the service will be forwarded to port 80 inside the container.

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f service.yaml

service/simple-service created

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f endpoints.yaml

endpoints/simple-service created

apiVersion: v1

kind: Endpoints

metadata:

  name: simple-service

subsets:

  - addresses:

      - ip: 10.244.0.3  # IP address of backend pod

    ports:

      - port: 80

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f service.yaml

service/simple-service created

PS D:\CKA BY zeal vora\PracticeME\section4> kubectl create -f endpoints.yaml

endpoints/simple-service created

PS D:\CKA BY zeal vora\PracticeME\section4>

C:\Windows\system32>kubectl describe service simple-service

Name: simple-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: <none>

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.97.84.215

IPs: 10.97.84.215

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints: 10.244.0.3:80

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

**# when forentne pod make request on port 8080 of service that request send on port 80 of backend port 80**

**Service become port and backend become target port**

root@frontend-pod:/# curl 10.97.84.215

# making request on ip addr of serive ip

root@frontend-pod:/# curl 10.97.84.215:8080

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

**$ sending requests on service port on 8080 and this service send request to endpoints on port 80**

# using selector

# all those pod who have appropriate label , the ip add of pods register as service endpoints

apiVersion: v1

kind: Service

metadata:

  name: simple-service

spec:

  ports:

  - port: 8080

    targetPort: 80

service1.yaml

$ DONE YHA TAK DONE

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 3h42m

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f service1.yaml

service/simple1-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service service1.yaml

Error from server (NotFound): services "service1.yaml" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple1-service

Name: simple1-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: <none>

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.96.149.26

IPs: 10.96.149.26

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints: <none>

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete service simple1-service

service "simple1-service" deleted

apiVersion: v1

kind: Service

metadata:

  name: simple1-service

spec:

  selector:

    app: backend

  ports:

  - port: 8080

    targetPort: 80

# thre is no selector that match this label

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f service1.yaml

service/simple1-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service service1.yaml

Error from server (NotFound): services "service1.yaml" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple1-service

Name: simple1-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.111.211.170

IPs: 10.111.211.170

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints:

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

**$No pod with these label that match with the selector , so no endpoint assingned**

D:\CKA BY zeal vora\PracticeME\section4>kubectl run backend-pod1 --image=nginx

pod/backend-pod1 created

D:\CKA BY zeal vora\PracticeME\section4>kubectl run backend-pod2 --image=nginx

pod/backend-pod2 created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple-service1

Error from server (NotFound): services "simple-service1" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple-service1

Error from server (NotFound): services "simple-service1" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple1-service

Name: simple1-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.111.211.170

IPs: 10.111.211.170

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints:

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\PracticeME\section4>kubectl label pod backend-pod-1 app=backend

Error from server (NotFound): pods "backend-pod-1" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl label pod backend-pod1 app=backend

pod/backend-pod1 labeled

D:\CKA BY zeal vora\PracticeME\section4>kubectl label pod backend-pod2 app=backend

pod/backend-pod2 labeled

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple1-service

Name: simple1-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.111.211.170

IPs: 10.111.211.170

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints: 10.244.0.5:80,10.244.0.6:80

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

# two end points have benn added in the service after specifying label in the pods

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

backend-pod1 1/1 Running 0 6m26s 10.244.0.5 minikube <none> <none>

backend-pod2 1/1 Running 0 6m13s 10.244.0.6 minikube <none> <none>

$ whose pod label is app=backend they are registred as end points of service

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

backend-pod1 1/1 Running 0 9m11s app=backend,run=backend-pod1

backend-pod2 1/1 Running 0 8m58s app=backend,run=backend-pod2

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods --show-labels

NAME READY STATUS RESTARTS AGE LABELS

backend-pod1 1/1 Running 0 10m app=backend,run=backend-pod1

backend-pod2 1/1 Running 0 10m run=backend-pod2

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service simple1-service

Name: simple1-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.111.211.170

IPs: 10.111.211.170

Port: <unset> 8080/TCP

TargetPort: 80/TCP

Endpoints: 10.244.0.5:80

Session Affinity: None

Internal Traffic Policy: Cluster

Events: <none>

**$when replica set env used extensively we can use selector, dynamically add pods within service ,**

Selector match with specific label

Kubectl delete service simple1-service

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h8m

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f service.yaml

service/simple-service created

D:\CKA BY zeal vora\PracticeME\section4>kubecl get svc

'kubecl' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section4>kubectl get sv

error: the server doesn't have a resource type "sv"

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h8m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 16s

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service --help

Create a service using a specified subcommand.

Aliases:

service, svc

Available Commands:

clusterip Create a ClusterIP service

externalname Create an ExternalName service

loadbalancer Create a LoadBalancer service

nodeport Create a NodePort service

Usage:

kubectl create service [flags] [options]

Use "kubectl create service <command> --help" for more information about a given command.

Use "kubectl options" for a list of global command-line options (applies to all commands).

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service --help

Create a service using a specified subcommand.

$ Create service using CLI

Aliases:

service, svc

**# DONE YHA TAK**

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service clusterip test-service --tcp=80:80

service/test-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h13m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 4m49s

test-service ClusterIP 10.105.131.68 <none> 80/TCP 6s

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service clusterip test-service --tcp=80:80 --dry-run=client -o yaml

apiVersion: v1

kind: Service

metadata:

creationTimestamp: null

labels:

app: test-service

name: test-service

spec:

ports:

- name: 80-80

port: 80

protocol: TCP

targetPort: 80

selector:

app: test-service

type: ClusterIP

status:

loadBalancer: {}

Nodeport service – so that internet connect to service or any external entity , we expose our port and traffic redirectred through node-port

All for worker node , nodeport is open

apiVersion: v1

kind: Service

metadata:

  name: backend-nodeport-service

spec:

  type: NodePort

  selector:

    app: backend  # Ensure this matches the backend pod label

  ports:

    - protocol: TCP

      port: 80       # The service port

      targetPort: 80 # The backend pod's port

      nodePort: 30008 # NodePort (must be in range 30000-32767)

node-port.yaml

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service clusterip test-service --tcp=80:80

service/test-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h13m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 4m49s

test-service ClusterIP 10.105.131.68 <none> 80/TCP 6s

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service clusterip test-service --tcp=80:80 --dry-run=client -o yaml

apiVersion: v1

kind: Service

metadata:

creationTimestamp: null

labels:

app: test-service

name: test-service

spec:

ports:

- name: 80-80

port: 80

protocol: TCP

targetPort: 80

selector:

app: test-service

type: ClusterIP

status:

loadBalancer: {}

D:\CKA BY zeal vora\PracticeME\section4>kubectl apply -f node-port.yaml

service/backend-nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

backend-nodeport-service NodePort 10.102.230.169 <none> 80:30008/TCP 7s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h26m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 18m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 13m

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service backend-nodeport-service

Name: backend-nodeport-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.102.230.169

IPs: 10.102.230.169

Port: <unset> 80/TCP

TargetPort: 80/TCP

NodePort: <unset> 30008/TCP

Endpoints: 10.244.0.5:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods -o wide

NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

backend-pod1 1/1 Running 0 95m 10.244.0.5 minikube <none> <none>

backend-pod2 1/1 Running 0 95m 10.244.0.6 minikube <none> <none>

D:\CKA BY zeal vora\PracticeME\section4>kubectl get nodes

NAME STATUS ROLES AGE VERSION

minikube Ready control-plane 5h29m v1.32.0

D:\CKA BY zeal vora\PracticeME\section4>kubectl get nodes -o wide

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

minikube Ready control-plane 5h29m v1.32.0 192.168.49.2 <none> Ubuntu 22.04.5 LTS 5.15.167.4-microsoft-standard-WSL2 docker://27.4.1

D:\CKA BY zeal vora\PracticeME\section4>curl 192.168.49.2

curl: (28) Failed to connect to 192.168.49.2 port 80 after 21047 ms: Could not connect to server

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

backend-nodeport-service NodePort 10.102.230.169 <none> 80:30008/TCP 5m3s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h31m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 23m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 18m

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod1 1/1 Running 0 99m

backend-pod2 1/1 Running 0 99m

D:\CKA BY zeal vora\PracticeME\section4>kubectl exec -it backend-pod1 -- bash

root@backend-pod1:/# curl ^C

root@backend-pod1:/# curl 10.102.230.169

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

<style>

html { color-scheme: light dark; }

body { width: 35em; margin: 0 auto;

font-family: Tahoma, Verdana, Arial, sans-serif; }

</style>

</head>

<body>

<h1>Welcome to nginx!</h1>

<p>If you see this page, the nginx web server is successfully installed and

working. Further configuration is required.</p>

<p>For online documentation and support please refer to

<a href="http://nginx.org/">nginx.org</a>.<br/>

Commercial support is available at

<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>

</body>

</html>

root@backend-pod1:/#

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete service backend-nodeport-service

service "backend-nodeport-service" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f node-ports.yaml

service/nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h45m

nodeport-service NodePort 10.100.69.146 <none> 8080:30940/TCP 29s

**$ Node port automatically assigned to services**

apiVersion: v1

kind: Service

metadata:

  name: nodeport-service

spec:

  selector:

    app: backend

  type: NodePort

  ports:

  - port: 8080

    targetPort: 80

nodeport.yaml

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f node-ports.yaml

service/nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h50m

nodeport-service NodePort 10.100.12.113 <none> 8080:30975/TCP 36s

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 42m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 37m

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service nodeport-service

Name: nodeport-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.12.113

IPs: 10.100.12.113

Port: <unset> 8080/TCP

TargetPort: 80/TCP

NodePort: <unset> 30975/TCP

Endpoints: 10.244.0.5:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

#End points are automatically connected if we are using seelcter and it matches with port label , they are connect as end points

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod1 1/1 Running 0 118m

backend-pod2 1/1 Running 0 118m

D:\CKA BY zeal vora\PracticeME\section4>kubectl run backend-pod --image=nginx

pod/backend-pod created

D:\CKA BY zeal vora\PracticeME\section4>kubectl label pod backend-pod app=backend

pod/backend-pod labeled

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service nodeport-service

Name: nodeport-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.12.113

IPs: 10.100.12.113

Port: <unset> 8080/TCP

TargetPort: 80/TCP

NodePort: <unset> 30975/TCP

Endpoints: 10.244.0.5:80,10.244.0.7:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

$ 30975 is node port we have

D:\CKA BY zeal vora\PracticeME\section4>kubectl apply -f nodeport.yaml

error: the path "nodeport.yaml" does not exist

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f node-ports.yaml

service/nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h50m

nodeport-service NodePort 10.100.12.113 <none> 8080:30975/TCP 36s

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 42m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 37m

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service nodeport-service

Name: nodeport-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.100.12.113

IPs: 10.100.12.113

Port: <unset> 8080/TCP

TargetPort: 80/TCP

NodePort: <unset> 30975/TCP

Endpoints: 10.244.0.5:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods

NAME READY STATUS RESTARTS AGE

backend-pod1 1/1 Running 0 118m

backend-pod2 1/1 Running 0 118m

D:\CKA BY zeal vora\PracticeME\section4>kubectl get nodes -o wide

NAME STATUS ROLES AGE VERSION INTERNAL-IP EXTERNAL-IP OS-IMAGE KERNEL-VERSION CONTAINER-RUNTIME

minikube Ready control-plane 5h55m v1.32.0 192.168.49.2 <none> Ubuntu 22.04.5 LTS 5.15.167.4-microsoft-standard-WSL2 docker://27.4.1

D:\CKA BY zeal vora\PracticeME\section4>minikube ip

192.168.49.2

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h55m

nodeport-service NodePort 10.100.12.113 <none> 8080:30975/TCP 5m56s

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 47m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 42m

D:\CKA BY zeal vora\PracticeME\section4>kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 5h55m

The system cannot find the file specified.

D:\CKA BY zeal vora\PracticeME\section4>nodeport-service NodePort 10.100.12.113 <none> 8080:30975/TCP 5m56s

D:\CKA BY zeal vora\PracticeME\section4>minikube service backend-nodeport-service --url

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service --help

Create a service using a specified subcommand.

Aliases:

service, svc

Available Commands:

**clusterip Create a ClusterIP service**

externalname Create an ExternalName service

loadbalancer Create a LoadBalancer service

nodeport Create a NodePort service

Usage:

kubectl create service [flags] [options]

Use "kubectl create service <command> --help" for more information about a given command.

Use "kubectl options" for a list of global command-line options (applies to all commands).

D:\CKA BY zeal vora\PracticeME\section4>kubectl create service nodeport test-nodeport --tcp=80:80 --dry-run=client -o yaml

apiVersion: v1

kind: Service

metadata:

creationTimestamp: null

labels:

app: test-nodeport

name: test-nodeport

spec:

ports:

- name: 80-80

port: 80

protocol: TCP

targetPort: 80

selector:

app: test-nodeport

type: NodePort

status:

loadBalancer: {}

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6h4m

nodeport-service NodePort 10.100.12.113 <none> 8080:30975/TCP 14m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 56m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 51m

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete -f node-ports.yaml

service "nodeport-service" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f node-ports.yaml

service/nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectlg get svc

'kubectlg' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6h6m

nodeport-service NodePort 10.106.157.9 <none> 8080:32081/TCP 15s

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 57m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 52m

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete service nodeport-service

service "nodeport-service" deleted

D:\CKA BY zeal vora\PracticeME\section4>

* + DONE

apiVersion: v1

kind: Service

metadata:

  name: nodeport-service

spec:

  selector:

    app: backend

  type: NodePort

  ports:

  - port: 8080

    targetPort: 80

    nodePort: 30556

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f node-ports.yaml

service/nodeport-service created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6h8m

nodeport-service NodePort 10.111.240.172 <none> 8080:30556/TCP 7s

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 59m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 54m

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete svc nodeport-service

service "nodeport-service" deleted

Service type load balancer

**Nodeport expose out envirnomet to external world** ,nodeprot not used for esxtrnal env prod

Load balancer intermediate between user and node port , load balancer send traffic to node port

**DONE YAHA TAK**

apiVersion: v1

kind: Service

metadata:

  name: backend-loadbalancer

spec:

  selector:

    app: backend

  type: LoadBalancer

  ports:

    - protocol: TCP

      port: 80

      targetPort: 80

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

D:\CKA BY zeal vora\PracticeME\section4>kubectl apply -f loadbalancer.yaml

service/backend-loadbalancer created

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

backend-loadbalancer LoadBalancer 10.106.229.211 <pending> 80:31280/TCP 7s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6h45m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 96m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 92m

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service backend-loadbalancer

Name: backend-loadbalancer

Namespace: default

Labels: <none>

Annotations: <none>

D:\CKA BY zeal vora\PracticeME\section4>minikube tunnel

\* Tunnel successfully started

\* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#access-to-ports-1024-on-windows-requires-root-permission

\* Starting tunnel for service backend-loadbalancer.

# minikube tunnel for extranal ip to service , ip add of load balancer

D:\CKA BY zeal vora\PracticeME\section4>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

backend-loadbalancer LoadBalancer 10.106.229.211 127.0.0.1 80:31280/TCP 3m12s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 6h48m

simple-service ClusterIP 10.99.4.158 <none> 8080/TCP 100m

test-service ClusterIP 10.105.131.68 <none> 80/TCP 95m

D:\CKA BY zeal vora\PracticeME\section4>curl 127.0.0.1

<!DOCTYPE html>

<html>

<head>

<title>Welcome to nginx!</title>

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe service backend-loadbalancer

Name: backend-loadbalancer

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=backend

Type: LoadBalancer

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.106.229.211

IPs: 10.106.229.211

LoadBalancer Ingress: 127.0.0.1 (VIP)

Port: <unset> 80/TCP

TargetPort: 80/TCP

NodePort: <unset> 31280/TCP

Endpoints: 10.244.0.5:80,10.244.0.7:80

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

**$ Bydfeilult node port automatically create , load balancer forward traffic to node port of worker node , and then it directly reach the traffic to pod**

apiVersion: v1

kind: Service

metadata:

  name: backend-loadbalancer

spec:

  selector:

    app: backend

  type: LoadBalancer

  ports:

      port: 80

      targetPort: 80

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete -f loadbalancer.yaml

service "backend-loadbalancer" deleted

Ingrees

**Whenve a load balancer receive request for any service , like ex.internal - it will go to ingrees component , ingress comp through it logic will forware to ex service based on rules**

**We define rule in ingress that route traffic**

**Load balacenr sned traffic to ingrees controller ,controller recive reqeuset from laod balancer and cotroller based on rules route traffic to approacpirate service**

**Controller based on rules send traffic to services**

$ ingress route traffic to service

D:\CKA BY zeal vora\PracticeME\section4>kubectl get ingress

No resources found in default namespace.

D:\CKA BY zeal vora\PracticeME\section4>kubectl create ingress --help

Create an ingress with the specified name.

Aliases:

ingress, ing

Examples:

# Create a single ingress called 'simple' that directs requests to foo.com/bar to svc

# svc1:8080 with a TLS secret "my-cert"

kubectl create ingress simple --rule="foo.com/bar=svc1:8080,tls=my-cert"

# request come for this website then it forward to this services

C:\>kubectl create ingress first-ingress --rule="example.internal/\*-example-service: 80"

ingress.networking.k8s.io/first-ingress created

C:\Windows\system32>kubectl create ingress first-ingress --rule="example.internal/\*=example-service:80"

ingress.networking.k8s.io/first-ingress created

D:\CKA BY zeal vora\kuberntes\_practice\_23 mar\part-15-ingress>kubectl create ingress first-ingress-rule --rule="example.internal/\*=example-service:80"

ingress.networking.k8s.io/first-ingress-rule created

C:\>kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kubernetes ClusterIP 10.109.0.1 <none> 443/TCP 26d

C:\>kubectl describe ingress first-ingress

Name: first-ingress

Labels: <none>

Namespace: default

Address:

Ingress Class: <none>

Default backend: <default>

Rules:

Host Path Backends

example.internal

example-service: 80 (<error: endpoints "example-service" not found>)

Annotations: <none>

Events: <none>

C:\>

**$ If any traffic come for ex.internal then it redirect to ex-service**

C:\>kubectl create ingress first-ingress --rule="example.internal/\*-example-service: 80" --rule="kplabs.internal/\*-kplabs-serv

ice:80"

C:\>kubectl describe ingress second-ingress

Name: second-ingress

Labels: <none>

Namespace: default

Address:

Ingress Class: <none>

Default backend: <default>

Rules:

Host Path Backends

----

example.internal

/ example-service: 80 (<error: endpoints "example-service" not found>)

\*plabs.internal

kplabs-service: 80 (<error: endpoints "kplabs-service" not found>)

Annotations: <none>

Events: <none>

C:\>

D:\CKA BY zeal vora\PracticeME\section4>kubectl create ingress second-ingress --rule="example.internal/\*=example-service:80" --rule="kplabs.internal/\*=kplabs-service:80"

ingress.networking.k8s.io/second-ingress created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress second-ingress

Name: second-ingress

Labels: <none>

Namespace: default

Address:

Ingress Class: <none>

Default backend: <default>

Rules:

Host Path Backends

---- ---- --------

example.internal

/ example-service:80 (<error: services "example-service" not found>)

kplabs.internal

/ kplabs-service:80 (<error: services "kplabs-service" not found>)

Annotations: <none>

Events: <none>

C:\>kubectl create ingress second-ingress --rule="example.internal/\*-example-service: 80" --rule="kplabs.internal/\* =kplabs-service:80" --dry-run-client -o yam

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

creationTimestamp: null

name: second-ingress

spec:

rules:

- host: example.internal

http:

paths:

- backend:

service:

name: example-service

port:

number: 80

path: /

pathType: Prefix

host: kplabs.internal

http:

paths:

- backend:

service:

name: kplabs-service

port:

number:180

path: /

pathType: Prefix

status:

Balancer: {}

**$we have 2 rules here one is intenal host and the other is**

MY

D:\CKA BY zeal vora\PracticeME\section4>kubectl create ingress second-ingress --rule="example.internal/\*=example-service:80" --rule="kplabs.internal/\*=kplabs-service:80" --dry-run=client -o yaml

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

creationTimestamp: null

name: second-ingress

spec:

rules:

- host: example.internal

http:

paths:

- backend:

service:

name: example-service

port:

number: 80

path: /

pathType: Prefix

- host: kplabs.internal

http:

paths:

- backend:

service:

name: kplabs-service

port:

number: 80

path: /

pathType: Prefix

status:

loadBalancer: {}

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  name: path-based-ingress

spec:

  rules:

    - http:

        paths:

          - path: /app-1

            pathType: Exact

            backend:

              service:

                name: app-1-service

                port:

                  number: 80

          - path: /app-2

            pathType: Exact

            backend:

              service:

                name: app-2-service

                port:

                  number: 80

path.yaml

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f path-based.yaml

ingress.networking.k8s.io/path-based-ingress created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress path-based

Name: path-based-ingress

Labels: <none>

Namespace: default

Address:

Ingress Class: <none>

Default backend: <default>

Rules:

Host Path Backends

---- ---- --------

\*

/app-1 app-1-service:80 (<error: services "app-1-service" not found>)

/app-2 app-2-service:80 (<error: services "app-2-service" not found>)

Annotations: <none>

Events: <none>

**$ If any request come on this path /app-1 then it redirect to app1-service**

D:\CKA BY zeal vora\PracticeME\section4>kubectl get ingress

NAME CLASS HOSTS ADDRESS PORTS AGE

first-ingress <none> example.internal 80 34m

path-based-ingress <none> \* 80 83s

second-ingress <none> example.internal,kplabs.internal 80 27m

D:\CKA BY zeal vora\PracticeME\section4>kubectl create ingress named-path --rule=example.internal/=app-1:80 --dry-run=client -o yaml

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

creationTimestamp: null

name: named-path

spec:

rules:

- host: example.internal

http:

paths:

- backend:

service:

name: app-1

port:

number: 80

path: /

pathType: Exact

status:

loadBalancer: {}

apiVersion: networking.k8s.io/v1

kind: Ingress

metadata:

  creationTimestamp: null

  name: named-path

spec:

  rules:

  - host: example.internal

    http:

      paths:

      - backend:

          service:

            name: app-1-service

            port:

              number: 80

        path: /app-1

        pathType: Exact

      - backend:

          service:

            name: app-2-service

            port:

              number: 80

        path: /app-2

        pathType: Exact

name-path.yaml

**1. host: example.internal**

yaml

CopyEdit

rules:

- host: example.internal

* The host specifies the **domain name** that this Ingress rule applies to.
* When a user accesses **http://example.internal**, this Ingress rule determines **which backend service** will handle the request.
* If host is omitted, the rule applies to **all hosts (\*)**.

✅ **Example Request Matching**

* A request to http://example.internal/app-1 → Matches the rule and routes traffic to **app-1-service**.
* A request to http://example.internal/app-2 → Routes traffic to **app-2-service**.
* A request to http://different-domain.com/app-1 → ❌ **Does not match** this Ingress rule.

**2. path: /app-1 and /app-2**

yaml

CopyEdit

- backend:

service:

name: app-1-service

port:

number: 80

path: /app-1

pathType: Exact

* The path defines the **specific URL path** that must be requested for this rule to apply.
* It works **with the host** to determine where traffic should be sent.

**Example Request Matching**

| **Request URL** | **Matched Path** | **Routed To** |
| --- | --- | --- |
| http://example.internal/app-1 | /app-1 | app-1-service:80 |
| http://example.internal/app-2 | /app-2 | app-2-service:80 |
| http://example.internal/app-3 | ❌ No match | ❌ 404 Not Found |

**pathType: Exact**

yaml

CopyEdit

pathType: Exact

* This means the path **must match exactly** (/app-1 or /app-2).
* A request to http://example.internal/app-1/extra **will not match** /app-1 because it's not an exact match.
* If you want to match **any subpath**, use pathType: Prefix instead.

### ****Final Example****

🔹 If a user visits **http://example.internal/app-1**, it gets routed to app-1-service.  
🔹 If they visit **http://example.internal/app-2**, it gets routed to app-2-service.  
🔹 If they visit **http://example.internal/app-3**, they get a **404 Not Found**.

D:\CKA BY zeal vora\PracticeME\section4>kubectl create -f name-path.yaml

ingress.networking.k8s.io/named-path created

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress name-path.yaml

Error from server (NotFound): ingresses.networking.k8s.io "name-path.yaml" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress name-path

Error from server (NotFound): ingresses.networking.k8s.io "name-path" not found

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress named-path

Name: named-path

Labels: <none>

Namespace: default

Address:

Ingress Class: <none>

Default backend: <default>

Rules:

Host Path Backends

---- ---- --------

example.internal

/app-1 app-1-service:80 (<error: services "app-1-service" not found>)

/app-2 app-2-service:80 (<error: services "app-2-service" not found>)

Annotations: <none>

**$ we have host and afater that we are doing routing , path based routing if any traffic come on /app1 it direct to service app-1-service**

C: \>kubectl create ingress main-ingress -- class=nginx -- rule="example. internal/ \*= example-service: 80" -- rule="kplabs.internal/\*= kplabs-service : 80"

ingress.networking.k8s.io/main-ingress created

D:\CKA BY zeal vora\PracticeME\section4>kubectl create ingress main-ingress --class=nginx --rule="example.internal/\*=example-service:80" --rule="kplabs.internal/\*=kplabs-service:80"

ingress.networking.k8s.io/main-ingress created

**# we have host example.intenal – if traffic come on example.int/app-1 or app-2 \* means any – redirect to traffic example-service**

D:\CKA BY zeal vora\PracticeME\section4>kubectl describe ingress main-ingress

Name: main-ingress

Labels: <none>

Namespace: default

Address:

Ingress Class: nginx

Default backend: <default>

Rules:

Host Path Backends

---- ---- --------

example.internal

/ example-service:80 (10.244.0.9:80)

kplabs.internal

/ kplabs-service:80 (10.244.0.10:80)

Annotations: <none>

Events: <none>

C: \>kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.12.0/deploy/static/provider/cloud/deploy.yaml

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods

NAME READY STATUS RESTARTS AGE

example-pod 1/1 Running 0 19m

kplabs-pod 1/1 Running 0 18m

D:\CKA BY zeal vora\PracticeME\section4>kubectl get namespace

NAME STATUS AGE

default Active 19h

ingress-nginx Active 96s

kube-node-lease Active 19h

kube-public Active 19h

kube-system Active 19h

D:\CKA BY zeal vora\PracticeME\section4>kubectl get pods -n ingress-nginx

NAME READY STATUS RESTARTS AGE

ingress-nginx-admission-create-b4npm 0/1 ContainerCreating 0 2m5s

ingress-nginx-admission-patch-pxqks 0/1 ContainerCreating 0 2m5s

ingress-nginx-controller-d8c96cf68-q57s2 0/1 ContainerCreating 0 2m5s

D:\CKA BY zeal vora\PracticeME\section4>kubectl get service -n ingress-nginx

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

ingress-nginx-controller LoadBalancer 10.107.244.211 <pending> 80:32551/TCP,443:31710/TCP 5m49s

ingress-nginx-controller-admission ClusterIP 10.108.170.102 <none> 443/TCP 5m49s

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete ingress main-ingress

ingress.networking.k8s.io "main-ingress" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete ingress --all

No resources found

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete ingress --all

No resources found

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete pods --all

pod "example-pod" deleted

pod "kplabs-pod" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete service example-service kplabs-service simple-service test-service

service "example-service" deleted

service "kplabs-service" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete svc --all

service "kubernetes" deleted

D:\CKA BY zeal vora\PracticeME\section4>kubectl delete -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v1.12.0/deploy/static/provider/cloud/deploy.yaml

namespace "ingress-nginx" deleted

**HELM**

Deploy worddpress chart in kuberntes cluster using helm

Package your whole appn In helm charts

Deploy helm chart in wordpress in kuberntes cluster

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>helm install demo-release bitnami/wordpress

'helm' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>choco install kubernetes-helm -y

Chocolatey v2.4.2

Installing the following packages:

kubernetes-helm

By installing, you accept licenses for the packages.

Downloading package from source 'https://community.chocolatey.org/api/v2/'

Progress: Downloading kubernetes-helm 3.17.0... 100%

kubernetes-helm v3.17.0 [Approved]

kubernetes-helm package files install completed. Performing other installation steps.

Downloading kubernetes-helm 64 bit

from 'https://get.helm.sh/helm-v3.17.0-windows-amd64.zip'

Progress: 100% - Completed download of C:\Users\user121\AppData\Local\Temp\chocolatey\kubernetes-helm\3.17.0\helm-v3.17.0-windows-amd64.zip (17.08 MB).

Download of helm-v3.17.0-windows-amd64.zip (17.08 MB) completed.

Hashes match.

Extracting C:\Users\user121\AppData\Local\Temp\chocolatey\kubernetes-helm\3.17.0\helm-v3.17.0-windows-amd64.zip to C:\ProgramData\chocolatey\lib\kubernetes-helm\tools...

C:\ProgramData\chocolatey\lib\kubernetes-helm\tools

ShimGen has successfully created a shim for helm.exe

The install of kubernetes-helm was successful.

Deployed to 'C:\ProgramData\chocolatey\lib\kubernetes-helm\tools'

Chocolatey installed 1/1 packages.

See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).

C:\Windows\system32>helm version

version.BuildInfo{Version:"v3.17.0", GitCommit:"301108edc7ac2a8ba79e4ebf5701b0b6ce6a31e4", GitTreeState:"clean", GoVersion:"go1.23.4"}

C:\Windows\system32>helm repo add bitnami https://charts.bitnami.com/bitnami

"bitnami" has been added to your repositories

C:\Windows\system32>helm repo update

Hang tight while we grab the latest from your chart repositories...

...Successfully got an update from the "bitnami" chart repository

Update Complete. ⎈Happy Helming!⎈

C:\Windows\system32>helm repo list

NAME URL

bitnami https://charts.bitnami.com/bitnami

C:\Windows\system32>helm install demo-release bitnami/wordpress

NAME: demo-release

LAST DEPLOYED: Sat Feb 15 13:23:21 2025

NAMESPACE: default

STATUS: deployed

REVISION: 1

TEST SUITE: None

NOTES:

CHART NAME: wordpress

CHART VERSION: 24.1.11

APP VERSION: 6.7.2

Did you know there are enterprise versions of the Bitnami catalog? For enhanced secure software supply chain features, unlimited pulls from Docker, LTS support, or application customization, see Bitnami Premium or Tanzu Application Catalog. See https://www.arrow.com/globalecs/na/vendors/bitnami for more information.

\*\* Please be patient while the chart is being deployed \*\*

Your WordPress site can be accessed through the following DNS name from within your cluster:

demo-release-wordpress.default.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:

NOTE: It may take a few minutes for the LoadBalancer IP to be available.

Watch the status with: 'kubectl get svc --namespace default -w demo-release-wordpress'

export SERVICE\_IP=$(kubectl get svc --namespace default demo-release-wordpress --template "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{ end }}")

echo "WordPress URL: http://$SERVICE\_IP/"

echo "WordPress Admin URL: http://$SERVICE\_IP/admin"

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

echo Username: user

echo Password: $(kubectl get secret --namespace default demo-release-wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d)

WARNING: There are "resources" sections in the chart not set. Using "resourcesPreset" is not recommended for production. For production installations, please set the following values according to your workload needs:

- resources

+info https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

demo-release-mariadb-0 0/1 Init:0/1 0 117s

demo-release-wordpress-74c6c9b6f7-twrk9 0/1 Init:0/1 0 117s

C:\Windows\system32>helm uninstall demo-release

release "demo-release" uninstalled

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

demo-release-mariadb-0 0/1 Terminating 0 4m22s

demo-release-wordpress-74c6c9b6f7-twrk9 0/1 Terminating 0 4m22s

**Installing Helm Charts**

Installing Jenkins charts

C:\Windows\system32>helm repo add jenkins https://charts.jenkins.io

"jenkins" has been added to your repositories

# adding Jenkins reposoitry

C:\Windows\system32>helm repo update

Hang tight while we grab the latest from your chart repositories...

...Successfully got an update from the "jenkins" chart repository

...Successfully got an update from the "bitnami" chart repository

Update Complete. ⎈Happy Helming!⎈

C:\Windows\system32>helm install my-jenkins jenkins/jenkins

NAME: my-jenkins

LAST DEPLOYED: Sat Feb 15 14:18:18 2025

NAMESPACE: default

STATUS: deployed

REVISION: 1

NOTES:

1. Get your 'admin' user password by running:

kubectl exec --namespace default -it svc/my-jenkins -c jenkins -- /bin/cat /run/secrets/additional/chart-admin-password && echo

2. Get the Jenkins URL to visit by running these commands in the same shell:

echo http://127.0.0.1:8080

kubectl --namespace default port-forward svc/my-jenkins 8080:8080

3. Login with the password from step 1 and the username: admin

4. Configure security realm and authorization strategy

5. Use Jenkins Configuration as Code by specifying configScripts in your values.yaml file, see documentation: http://127.0.0.1:8080/configuration-as-code and examples: https://github.com/jenkinsci/configuration-as-code-plugin/tree/master/demos

For more information on running Jenkins on Kubernetes, visit:

https://cloud.google.com/solutions/jenkins-on-container-engine

For more information about Jenkins Configuration as Code, visit:

https://jenkins.io/projects/jcasc/

NOTE: Consider using a custom image with pre-installed plugins

C:\Windows\system32>helm list

NAME NAMESPACE REVISION UPDATED STATUS CHART

APP VERSION

my-jenkins default 1 2025-02-15 14:18:18.760885 +0530 IST deployed jenkins-5.8.11

2.492.1

C:\Windows\system32>helm uninstall my-jenkins

release "my-jenkins" uninstalled

C:\Windows\system32>

**$ Cluster must have sufficient CPU to deployed helm charts**

**Namespaces**

Used to islolate resources in kuberntes

C:\Windows\system32>kubectl get pods

No resources found in default namespace.

C:\Windows\system32>kubectl get pods --all namespace

error: unknown flag: --all

See 'kubectl get --help' for usage.

C:\Windows\system32>kubectl get pods --all-namespaces

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system coredns-668d6bf9bc-qw7vf 1/1 Running 1 (3h35m ago) 21h

kube-system etcd-minikube 1/1 Running 1 (3h35m ago) 21h

kube-system kube-apiserver-minikube 1/1 Running 5 (3h35m ago) 21h

kube-system kube-controller-manager-minikube 1/1 Running 2 (3h35m ago) 21h

kube-system kube-proxy-mhg2f 1/1 Running 1 (3h35m ago) 21h

kube-system kube-scheduler-minikube 1/1 Running 1 (3h35m ago) 21h

kube-system storage-provisioner 1/1 Running 4 (3h34m ago) 21h

$ No pod in default namespace but in other namespace pod is running , isolating resource using namespace , pod are running in kubentstem namepsac’e ,never touch kube-system namespacs

C:\Windows\system32>kubectl get configmaps

NAME DATA AGE

kube-root-ca.crt 1 21h

C:\Windows\system32>kubectl get configmaps --all-namespaces

NAMESPACE NAME DATA AGE

default kube-root-ca.crt 1 22h

kube-node-lease kube-root-ca.crt 1 22h

kube-public cluster-info 2 22h

kube-public kube-root-ca.crt 1 22h

kube-system coredns 1 22h

kube-system extension-apiserver-authentication 6 22h

kube-system kube-apiserver-legacy-service-account-token-tracking 1 22h

kube-system kube-proxy 2 22h

kube-system kube-root-ca.crt 1 22h

kube-system kubeadm-config 1 22h

kube-system kubelet-config 1 22h

# **never tuch kube system otherwise functionality of kuberntes cluster might be impactes**

C:\Windows\system32>kubectl get namespaces

NAME STATUS AGE

default Active 22h

kube-node-lease Active 22h

kube-public Active 22h

kube-system Active 22h

C:\Windows\system32>kubectl run test-pod --image=nginx

pod/test-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

test-pod 1/1 Running 0 7s

C:\Windows\system32>

**$ by default pod is created inside default namespace**

C:\Windows\system32>kubectl get pods -n kube-system

NAME READY STATUS RESTARTS AGE

coredns-668d6bf9bc-qw7vf 1/1 Running 1 (3h44m ago) 22h

etcd-minikube 1/1 Running 1 (3h44m ago) 22h

kube-apiserver-minikube 1/1 Running 5 (3h44m ago) 22h

kube-controller-manager-minikube 1/1 Running 2 (3h44m ago) 22h

kube-proxy-mhg2f 1/1 Running 1 (3h44m ago) 22h

kube-scheduler-minikube 1/1 Running 1 (3h44m ago) 22h

storage-provisioner 1/1 Running 4 (3h43m ago) 22h

$ pod that are running in kube system name spaces

C:\Windows\system32>kubectl create namespace development

namespace/development created

C:\Windows\system32>kubectl create namespace qa

namespace/qa created

C:\Windows\system32>kubectl get namespace

NAME STATUS AGE

default Active 22h

development Active 2m52s

kube-node-lease Active 22h

kube-public Active 22h

kube-system Active 22h

qa Active 2m19s

C:\Windows\system32>kubectl run development-pod --image=nginx -n development

pod/development-pod created

#in this namespace development , createing a pod

C:\Windows\system32>kubectl run qa-pod --image=nginx -n qa

pod/qa-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

test-pod 1/1 Running 0 9m26s

**# create pod in development anemspace and qa namespace**

C:\Windows\system32>kubectl get pods -n development

NAME READY STATUS RESTARTS AGE

development-pod 1/1 Running 0 2m3s

C:\Windows\system32>kubectl get pods -n qa

NAME READY STATUS RESTARTS AGE

qa-pod 1/1 Running 0 116s

#pods in qa and development namespace

C:\Windows\system32>kubectl get pods --all-namespaces

NAMESPACE NAME READY STATUS RESTARTS AGE

default test-pod 1/1 Running 0 11m

development development-pod 1/1 Running 0 3m21s

kube-system coredns-668d6bf9bc-qw7vf 1/1 Running 1 (3h54m ago) 22h

kube-system etcd-minikube 1/1 Running 1 (3h54m ago) 22h

kube-system kube-apiserver-minikube 1/1 Running 5 (3h54m ago) 22h

kube-system kube-controller-manager-minikube 1/1 Running 2 (3h54m ago) 22h

kube-system kube-proxy-mhg2f 1/1 Running 1 (3h54m ago) 22h

kube-system kube-scheduler-minikube 1/1 Running 1 (3h54m ago) 22h

kube-system storage-provisioner 1/1 Running 4 (3h54m ago) 22h

qa qa-pod 1/1 Running 0 3m6s

**#prodn env is seprate from kubernets cluter like seprate namespace in diff env , not mixed with dev and staging env**

C:\Windows\system32>kubectl run test-pod --image=nginx -n qa --dry-run=client -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: test-pod

name: test-pod

namespace: qa

spec:

containers:

- image: nginx

name: test-pod

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

**#cretaing pod in the qa namespaces**

**service accounts**

**tokens we have in config file so that we can connect to kuberntes clutser**

s

C:\Windows\system32>kubectl get serviceaccounts --all-namespaces

NAMESPACE NAME SECRETS AGE

default default 0 22h

development default 0 14m

kube-node-lease default 0 22h

**#when a service account asoocaaite with pod automatically , then pods get tokens to intercat with kubetnes clustes**

C:\Windows\system32>kubectl run app-pod --image=nginx

pod/app-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

app-pod 1/1 Running 0 9s

test-pod 1/1 Running 0 24m

C:\Windows\system32>kubectl describe pod app-pod

Name: app-pod

Namespace: default

Priority: 0

Service Account: default

Node: minikube/192.168.49.2

C:\Windows\system32>kubectl exec -it app-pod -- bash

root@app-pod:/# cd /var/run/secrets/kubernetes.io/serviceaccount/

root@app-pod:/var/run/secrets/kubernetes.io/serviceaccount# cat token

eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc1aGNuLVdFY2FvMWVLbjh4clVab0Zpb0ZCd1FEZWs4UGZ2dWtyZ0ZDSDgifQ..Ji2Duhh4QWT5f0HdPK8bjn1poQg3ShlKNaWRbhxt\_G1ZJqw7SbkpCqyPatS2D12rXQGRmzx7RXBm3JufP56FlQr7H6fNro49AIlHPZz3epM0wKs2\_zVY-tt30utTw95YDkWV7oPHe2kQ93-tGQEidnnAgdvHEWR3ldh9VSM9f4MkFVKu3YIkX2ryIIkUU8d8icuyKiwh0X1n-AWHF9jce4gahoUCG\_Q6jgo-9C8AlLjHTLo6a1dmw1wi5zGvwSUp4Kuut-g7h3iAEvl-r5P97S8SiCtacDLvL9daOVFa1IhhhZ18W84pmcYzva20EycA-LIWJDfBNUXhmTvVKaZ\_wgroot@app-pod:/var/run/secrets/kubernetes.io/serviceaccount#

**pod use token to authenticate all resources**

root@app-pod:/var/run/secrets/kubernetes.io/serviceaccount# cat token

eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc1aGNuLVdFY2FvMWVLbjh4clVab0Zpb0ZCd1FEZWs4UGZ2dWtyZ0ZDSDgifQ..Ji2Duhh4QWT5f0HdPK8bjn1poQg3ShlKNaWRbhxt\_G1ZJqw7SbkpCqyPatS2D12rXQGRmzx7RXBm3JufP56FlQr7H6fNro49AIlHPZz3epM0wKs2\_zVY-tt30utTw95YDkWV7oPHe2kQ93-tGQEidnnAgdvHEWR3ldh9VSM9f4MkFVKu3YIkX2ryIIkUU8d8icuyKiwh0X1n-AWHF9jce4gahoUCG\_Q6jgo-9C8AlLjHTLo6a1dmw1wi5zGvwSUp4Kuut-g7h3iAEvl-r5P97S8SiCtacDLvL9daOVFa1IhhhZ18W84pmcYzva20EycA-LIWJDfBNUXhmTvVKaZ\_wgroot@app-pod:/var/run/secrets/kubernetes.io/serviceaccount# token=$(cat token)

root@app-pod:/var/run/secrets/kubernetes.io/serviceaccount# echo $token

C:\Users\user121>kubectl cluster-info

Kubernetes control plane is running at https://127.0.0.1:59601

CoreDNS is running at <https://127.0.0.1:59601/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy>

**$ control plane running on this url**

t@app-pod:/var/run/secrets/kubernetes.io/serviceaccount# curl -k -H "Authorization: Bearer Stoken" https://0f3570d8-03b7-4r-aaf4-4c1b90504be3.k8s.ondigitalocean.com/api/v1

C:\Windows\system32>kubectl get serviceaccount

NAME SECRETS AGE

default 0 22h

C:\Windows\system32>kubectl get sa

NAME SECRETS AGE

default 0 22h

C:\Windows\system32>kubectl get sa -n kube-system

NAME SECRETS AGE

attachdetach-controller 0 22h

bootstrap-signer 0 22h

certificate-controller 0 22h

C:\Windows\system32>kubectl create namespace kplabs-test

namespace/kplabs-test created

C:\Windows\system32>kubectl get sa -n kplabs-test

NAME SECRETS AGE

default 0 19s

**Pod has default service account create after creteing a pod**

C:\Windows\system32>kubectl get sa

NAME SECRETS AGE

default 0 22h

C:\Windows\system32>kubectl create sa custom-sa

serviceaccount/custom-sa created

C:\Windows\system32>kubectl get sa

NAME SECRETS AGE

custom-sa 0 7s

default 0 22h

**$created a default service account**

C:\Windows\system32>kubectl run custom-pod --image=nginx --dry-run=client -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: custom-pod

name: custom-pod

spec:

containers:

- image: nginx

name: custom-pod

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

apiVersion: v1

kind: Pod

metadata:

labels:

run: custom-pod

name: custom-pod

spec:

serviceAccountName: custom-sa

containers:

- image: nginx

name: custom-pod

c

C:\Windows\system32>kubectl run custom-pod --image=nginx --dry-run=client -o yaml > custom-pod-sa.yaml

C:\Windows\system32>notepad custom-pod-sa.yaml

C:\Windows\system32>kubectl apply -f custom-pod-sa.yaml

pod/custom-pod created

C:\Windows\system32>notepad custom-pod-sa.yaml

apiVersion: v1

kind: Pod

metadata:

name: custom-pod

labels:

run: custom-pod

spec:

serviceAccountName: custom-sa # Assigns the custom service account

containers:

- name: custom-pod

image: nginx

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

C:\Windows\system32>kubectl apply -f custom-pod-sa.yaml

pod/custom-pod created

C:\Windows\system32>kubectl describe pod custom-pod

Name: custom-pod

Namespace: default

Priority: 0

Service Account: custom-sa

Node: minikube/192.168.49.2

**$ assign service account to pod**

**Namedport**

**We can specify name in the cotnaiern refer to name port**

C:\Windows\system32>kubectl run nginx --image=nginx --port=80 --dry-run=client -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: nginx

name: nginx

spec:

containers:

- image: nginx

name: nginx

ports:

- containerPort: 80

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

C:\Windows\system32>kubectl run nginx --image=nginx --port=80

pod/nginx created

C:\Windows\system32>kubectl expose pod nginx --name first-svc --port=80 --target-port=http --type=NodePort

service/first-svc exposed

#by exposing port we are creating a service

C:\Windows\system32>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

first-svc NodePort 10.98.203.74 <none> 80:32052/TCP 14s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 170m

# we had container port but has no names in this

#extranal ip public ip of worker node

C:\Windows\system32>kubectl run nginx --image=nginx --port=80 --dry-run=client -o yaml > pod\_named.yaml

C:\Windows\system32>notepad pod\_named.yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: nginx

name: nginx

spec:

containers:

- image: nginx

name: nginx

ports:

- containerPort: 80

name: custom-http

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

**# name=custom http is our naemsd port**

**We can use it directly , don’t need to give the port nounber**

C:\Windows\system32>kubectl apply -f pod\_named.yaml

Warning: resource pods/nginx is missing the kubectl.ku

C:\Windows\system32>kubectl expose pod nginx2 --name named-svc first-svc --port=80 --target-port=http --type=NodePort

Error from server (NotFound): pods "nginx2" not found

AUTHETNICITY

apiVersion: v1

kind: Pod

metadata:

  creationTimestamp: null

  labels:

    run: external-pod

  name: external-pod

spec:

  serviceAccountName: external

  containers:

  - image: nginx

    name: external-pod

    resources: {}

  dnsPolicy: ClusterFirst

  restartPolicy: Always

status: {}

C:\Windows\system32>kubectl create sa external

serviceaccount/external created

C:\Windows\system32>kubectl run external-pod --image=nginx --dry-run=client -o yaml

apiVersion: v1

kind: Pod

metadata:

creationTimestamp: null

labels:

run: external-pod

name: external-pod

spec:

containers:

- image: nginx

name: external-pod

resources: {}

dnsPolicy: ClusterFirst

restartPolicy: Always

status: {}

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl apply -f sa-pod.yaml

pod/external-pod created

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl get pods

NAME READY STATUS RESTARTS AGE

external-pod 1/1 Running 0 11s

nginx 1/1 Running 1 (20m ago) 9h

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl get pods external-pod -o yaml

apiVersion: v1

kind: Pod

metadata:

**$ poD RUNNING in external serviceaccount name**

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl exec -it external-pod -- bash

root@external-pod:/# ls -l /run/secrets/kubernetes.io/serviceaccount/

total 0

lrwxrwxrwx 1 root root 13 Feb 15 19:14 ca.crt -> ..data/ca.crt

lrwxrwxrwx 1 root root 16 Feb 15 19:14 namespace -> ..data/namespace

lrwxrwxrwx 1 root root 12 Feb 15 19:14 token -> ..data/token

root@external-pod:/# cat /run/secrets/kubernetes.io/serviceaccount/token

eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc1aGNuLVdFY2FvMWVLbjh4clVab0Zpb0ZCd1FEZWs4UGZ2dWtyZ0ZDSDgifQ.

#DONE YAHA TAK

**AUTHORIZATION**

First we define role like read or write service and then role binding which user can perfume this opn

apiVersion: rbac.authorization.k8s.io/v1

kind: Role

metadata:

  namespace: default

  name: pod-reader

rules:

  - apiGroups: [""]  # "" indicates the core API group

    resources: ["pods"]

    verbs: ["list"]

**pod have only list permission ,action is list and**

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl apply -f role.yaml

role.rbac.authorization.k8s.io/pod-reader created

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl get role

NAME CREATED AT

pod-reader 2025-02-15T19:44:55Z

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl describe role pod-reader

Name: pod-reader

Labels: <none>

Annotations: <none>

PolicyRule:

Resources Non-Resource URLs Resource Names Verbs

--------- ----------------- -------------- -----

pods [] [] [list]

$ pod have only list persmissions

$ **Created role successfully now assocate role with user s , first we create a role then we associated with pods**

apiVersion: rbac.authorization.k8s.io/v1

kind: RoleBinding

metadata:

  name: read-pods

  namespace: default

subjects:

  - kind: User

    name: jane

    apiGroup: rbac.authorization.k8s.io  # Corrected typo (was "kas.io")

roleRef:

  kind: Role

  name: pod-reader

  apiGroup: rbac.authorization.k8s.io

**pod-reader role associate with role user jane ,in subject define users and in roleref define permsiions ,pod readaer role associated with user jane**

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl apply -f rolebinding.yaml

rolebinding.rbac.authorization.k8s.io/read-pods created

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl get rolebinding

NAME ROLE AGE

read-pods Role/pod-reader 39s

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl describe rolebinding read-pods

Name: read-pods

Labels: <none>

Annotations: <none>

Role:

Kind: Role

Name: pod-reader

Subjects:

Kind Name Namespace

---- ---- ---------

User jane

**$ this role associated with User Jane**

**$ Pod reader associated with user jane**

:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl delete -f role.yaml

role.rbac.authorization.k8s.io "pod-reader" deleted

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl delete -f rolebinding.yaml

rolebinding.rbac.authorization.k8s.io "read-pods" deleted

**CLUSTER ROLE and ClusterRole Binding**

Cluster role is global define across all namespaces

apiVersion: rbac.authorization.k8s.io/v1

kind: ClusterRole

metadata:

  name: pod-reader

rules:

  - apiGroups: [""]

    resources: ["pods"]

    verbs: ["list"]

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl apply -f cluster-role.yaml

clusterrole.rbac.authorization.k8s.io/pod-reader created

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl get clusterrole

NAME CREATED AT

admin 2025-02-14T11:09:08Z

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl describe clusterrole pod-reader

Name: pod-reader

Labels: <none>

Annotations: <none>

PolicyRule:

Resources Non-Resource URLs Resource Names Verbs

--------- ----------------- -------------- -----

pods [] [] [list]

**$ pod have only list permissions**

apiVersion: rbac.authorization.k8s.io/v1

kind: clusterRoleBinding

metadata:

  name: list-pods-global

subjects:

  - kind: User

    name: system:serviveaccount:default:external

    apiGroup: rbac.authorization.k8s.io

roleRef:

  kind: ClusterRole

  name: pod-reader

  apiGroup: rbac.authorization.k8s.io

**$ cluser binding for cluster role for user pod-reader**

#YHA TAK DONE

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl apply -f cluster-role-binding.yaml

clusterrolebinding.rbac.authorization.k8s.io/list-pods-global created

D:\CKA BY zeal vora\PracticeME\section5 Domain and securtiy>kubectl describe clusterrolebinding list-pods-global

Name: list-pods-global

Labels: <none>

Annotations: <none>

Role:

Kind: ClusterRole

Name: pod-reader

Subjects:

Kind Name Namespace

---- ---- ---------

ServiceAccount external default

**$ pod read permission associate waiht external user**

**Through cluster permission are attach to every or default namespace , thes permission are valid for the user in every namespace** – check karlene dfault or evey namespace

**Authentication**

To get public and private key

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>ssh-keygen

Generating public/private ed25519 key pair.

Enter file in which to save the key (C:\Users\user121/.ssh/id\_ed25519): deep

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in deep

Your public key has been saved in deep.pub

The key fingerprint is:

SHA256:WpkPzUHlaJeut/P7ZsVLjgWaTjpjRbW9e8XqNsb3mGE user121@DESKTOP-JFJ21I5

The key's randomart image is:

+--[ED25519 256]--+

| ... |

| . o o |

| + = o |

| \* = o . |

| S + + .o.|

| o o \* +=|

| . B ..E.=|

| = o.+\*B=|

| . o .\*\*B=|

+----[SHA256]-----+

C:\Windows\system32>cat id\_rsa.pub

'cat' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>cat id\_rsa

'cat' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>

**$green with https signal we can define**

**$c Ceertificate based authentications**

Kubectconfig FILES

C:\Windows\system32>kubectl config view

apiVersion: v1

clusters:

- cluster:

certificate-authority-data: DATA+OMITTED

server: https://kubernetes.docker.internal:6443

name: docker-desktop

- cluster:

certificate-authority: C:\Users\user121\.minikube\ca.crt

extensions:

- extension:

last-update: Sun, 16 Feb 2025 00:24:20 IST

provider: minikube.sigs.k8s.io

version: v1.35.0

name: cluster\_info

server: https://127.0.0.1:51990

name: minikube

contexts:

- context:

cluster: docker-desktop

user: docker-desktop

name: docker-desktop

- context:

cluster: minikube

extensions:

- extension:

last-update: Sun, 16 Feb 2025 00:24:20 IST

provider: minikube.sigs.k8s.io

version: v1.35.0

name: context\_info

namespace: default

user: minikube

name: minikube

current-context: minikube

kind: Config

preferences: {}

users:

- name: docker-desktop

user:

client-certificate-data: DATA+OMITTED

client-key-data: DATA+OMITTED

- name: minikube

user:

client-certificate: C:\Users\user121\.minikube\profiles\minikube\client.crt

client-key: C:\Users\user121\.minikube\profiles\minikube\client.key

C:\Windows\system32>

**$ GENERATE KUBECONFIG FILE FROM SCRATCH**

**Document - Kubeconfig from Scratch**

Create KubeConfig from scratch

**1. Add cluster details.**

kubectl config --kubeconfig=base-config set-cluster development --server=https://1.2.3.4

**2. Add user details**

kubectl config --kubeconfig=base-config set-credentials experimenter --username=dev --password=some-password

**3. Setting Contexts**

kubectl config --kubeconfig=base-config set-context dev-frontend --cluster=development --namespace=frontend --user=experimenter

**4. Repeating above steps for second cluster**

kubectl config --kubeconfig=base-config set-cluster production --server=https://4.5.6.7

kubectl config --kubeconfig=base-config set-context prod-frontend --cluster=production --namespace=frontend --user=experimenter

**Next Steps:**

**1. View Kubeconfig**

kubectl config --kubeconfig=base-config view

**2. Get current conext information:**

kubectl config --kubeconfig=base-config get-contexts

**3. Switch Conexts:**

kubectl config --kubeconfig=base-config use-context dev-frontend

Kubernetes Volumes

apiVersion: v1

kind: Pod

metadata:

name: demopod-volume

spec:

containers:

- image: nginx

name: test-container

volumeMounts:

- mountPath: /data

name: first-volume

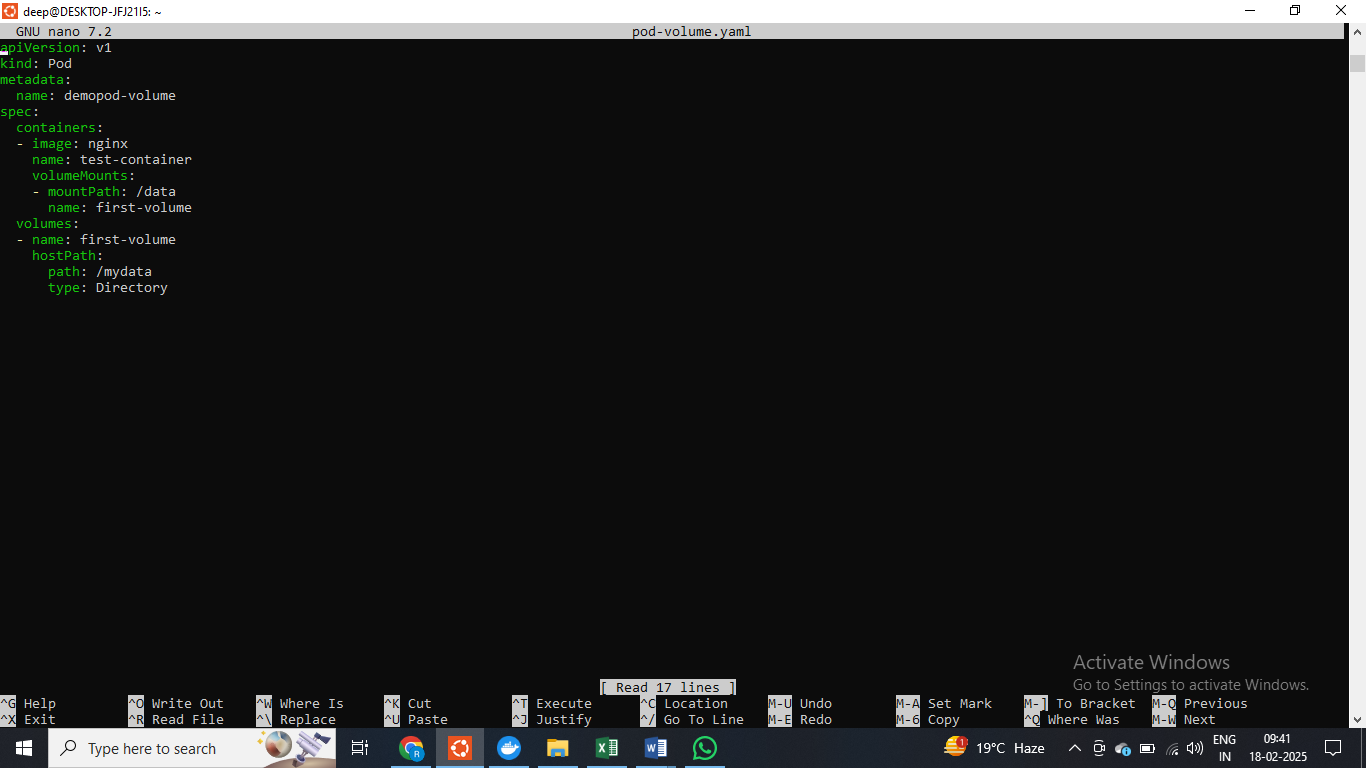
volumes:

- name: first-volume

hostPath:

path: /mydata

type: Directory



nano-podVolume.yaml

**Host directory /mydata mount with /data , munting the director of host /mydata within the container /data**

[root@demo-docker video]# mkdir /mydata

etes demo-docker video]# kubectl apply -f pod-volume.yaml

pod/demopod-volume created

[root@demo-docker video]# kubectl get pods

NAMEREADYSTATUS

demopod-volume1/1Running

[root@demo-docker video]# kubectl exec -it demopod-volume bash

root@demopod-volume: /# cd /data

root@demopod-volume:/data# touch kplabs.txt

root@demopod-volume:/data# echo "Hi" > kplabs.txt

root@demopod-volume:/data# 1s

kplabs. txt

root@demopod-volume:/data# \_

root@demopod-volume:/data# df -h

etesystemSizeUsed Avail Use% Mounted on

overlay25G3.1G

tmpfs3.9GO

tmpfs3.9G0

/dev/vdal25G3.1G20

shm64M

tmpfs3.9G12K

tmpfs3.9G0

tmpfs3.9G

tmpfs3.9G0 3.9G

root@demopod-volume:/data# -

[root@demo-docker video]# cd /mydata/

etes demo-docker mydata]# 1s

kplabs. txt

[root@demo-docker mydata]# cat kplabs. txt

Hi

[root@demo-docker mydata]# \_

**$ Persisitent Volume and PVC**

deep@DESKTOP-JFJ21I5:~$ nano pv.yaml

deep@DESKTOP-JFJ21I5:~$ mkdir /tmp/data

deep@DESKTOP-JFJ21I5:~$ kubectl apply pv.yaml

error: Unexpected args: [pv.yaml]

See 'kubectl apply -h' for help and examples

deep@DESKTOP-JFJ21I5:~$ kubectl apply -f pv.yaml

persistentvolume/block-pv created

deep@DESKTOP-JFJ21I5:~$ kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE

block-pv 10Gi RWO Retain Available manual <unset> 24s

deep@DESKTOP-JFJ21I5:~$ nano pvc.yaml

deep@DESKTOP-JFJ21I5:~$ kubectl apply -f pvc.yaml

persistentvolumeclaim/pvc created

deep@DESKTOP-JFJ21I5:~$ kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE

pvc Bound block-pv 10Gi RWO manual <unset> 15s

deep@DESKTOP-JFJ21I5:~$ nano pvc.yaml

deep@DESKTOP-JFJ21I5:~$ nano pvc.yaml

deep@DESKTOP-JFJ21I5:~$ kubectl apply -f pod-pvc.yaml

error: the path "pod-pvc.yaml" does not exist

deep@DESKTOP-JFJ21I5:~$ kubectl delete pvc pvc

persistentvolumeclaim "pvc" deleted

deep@DESKTOP-JFJ21I5:~$ kubectl apply -f pvc.yaml

pod/kplabs-pvc created

deep@DESKTOP-JFJ21I5:~$ kubectl get pods

NAME READY STATUS RESTARTS AGE

demopod-volume 0/1 ContainerCreating 0 34m

kplabs-pvc 0/1 Pending 0 12s

deep@DESKTOP-JFJ21I5:~$

**$ Static Vs Dynamic PVS**

deep@DESKTOP-JFJ21I5:~$ kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE

block-pv 10Gi RWO Retain Released default/pvc manual <unset> 51m

deep@DESKTOP-JFJ21I5:~$ ls

cloud day5 devops. hello.txt ldf.zip minikube-linux-amd64 newfile.txt pod-volume.yaml pvc.yaml

cloud.tar.gz demo-file.txt file-sample\_100kB.doc kplabs.txt linux\_for\_devops my-file.txt nohup.out pv.yaml softlink-file

deep@DESKTOP-JFJ21I5:~$ nano pvc.yaml % STATIC PVC

GNU nano 7.2 pvc.yaml apiVersion: v1

kind: Pod

metadata:

name: kplabs-pvc

spec:

containers:

- name: my-frontend

image: nginx

volumeMounts:

- mountPath: "/data"

name: my-volume

volumes:

- name: my-volume

persistentVolumeClaim:

claimName: pvc

DYNAMIC PVC

deep@DESKTOP-JFJ21I5:~$ kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE

block-pv 10Gi RWO Retain Released default/pvc manual <unset> 51m

deep@DESKTOP-JFJ21I5:~$ ls

cloud day5 devops. hello.txt ldf.zip minikube-linux-amd64 newfile.txt pod-volume.yaml pvc.yaml

cloud.tar.gz demo-file.txt file-sample\_100kB.doc kplabs.txt linux\_for\_devops my-file.txt nohup.out pv.yaml softlink-file

deep@DESKTOP-JFJ21I5:~$ nano pvc.yaml

deep@DESKTOP-JFJ21I5:~$ nano dynamic-pvc.yaml

deep@DESKTOP-JFJ21I5:~$ kubectl apply -f dynamic-pvc.yaml

persistentvolumeclaim/dynamicpvc created

deep@DESKTOP-JFJ21I5:~$ kubectl get pvs

error: the server doesn't have a resource type "pvs"

deep@DESKTOP-JFJ21I5:~$ kubectl get pvc

NAME STATUS VOLUME CAPACITY ACCESS MODES STORAGECLASS VOLUMEATTRIBUTESCLASS AGE

dynamicpvc Bound pvc-af34b8eb-8f0e-4d11-83b1-4fe0a86746d4 50Gi RWO standard <unset> 20s

deep@DESKTOP-JFJ21I5:~$ kubectl get pv

NAME CAPACITY ACCESS MODES RECLAIM POLICY STATUS CLAIM STORAGECLASS VOLUMEATTRIBUTESCLASS REASON AGE

block-pv 10Gi RWO Retain Released default/pvc manual <unset> 55m

pvc-af34b8eb-8f0e-4d11-83b1-4fe0a86746d4 50Gi RWO Delete Bound default/dynamicpvc standard <unset> 28s

deep@DESKTOP-JFJ21I5:~$ kubectl describe pv pvc-af34b8eb-8f0e-4d11-83b1-4fe0a86746d4

Name: pvc-af34b8eb-8f0e-4d11-83b1-4fe0a86746d4

Labels: <none>

Annotations: hostPathProvisionerIdentity: 392f1f57-760e-4845-83d3-e0b4d4cac389

pv.kubernetes.io/provisioned-by: k8s.io/minikube-hostpath

Finalizers: [kubernetes.io/pv-protection]

StorageClass: standard

Status: Bound

Claim: default/dynamicpvc

Reclaim Policy: Delete

Access Modes: RWO

VolumeMode: Filesystem

Capacity: 50Gi

Node Affinity: <none>

Message:

Source:

Type: HostPath (bare host directory volume)

Path: /tmp/hostpath-provisioner/default/dynamicpvc

HostPathType:

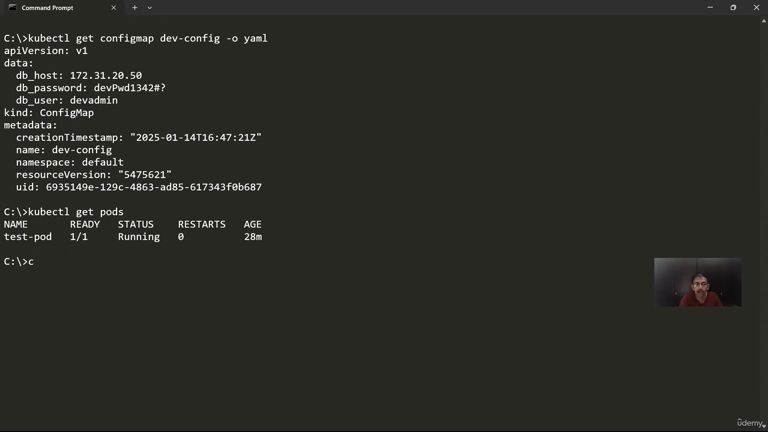
Events: <none>

deep@DESKTOP-JFJ21I5:~$

# DONE YHA TAK

Digital Ocean Documentation

<https://www.digitalocean.com/docs/kubernetes/how-to/add-volumes/>

KUBECONFIG

C:\Windows\system32>kubectl get configmaps

NAME DATA AGE

kube-root-ca.crt 1 2d12h

C:\Windows\system32>kubectl create configmap first-configmap --from-literal=key1=value1 --from-literal=key2=value2

configmap/first-configmap created

C:\Windows\system32>kubectl get configmap

NAME DATA AGE

first-configmap 2 16s

kube-root-ca.crt 1 2d12h

C:\Windows\system32>kubectl describe configmap first-configmap

Name: first-configmap

Namespace: default

Labels: <none>

Annotations: <none>

Data

====

key1:

----

value1

key2:

----

value2

BinaryData

====

Events: <none>

C:\Windows\system32>kubectl get configmap first-configmap -o yaml

apiVersion: v1

data:

key1: value1

key2: value2

kind: ConfigMap

metadata:

creationTimestamp: "2025-02-18T06:54:18Z"

name: first-configmap

namespace: default

resourceVersion: "76250"

uid: 3b980762-38b3-42e5-a0c0-211c9616e8fc

C:\Windows\system32>

Microsoft Windows [Version 10.0.19045.5487]

(c) Microsoft Corporation. All rights reserved.

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl create configmap second-configmap --from-file=large-file.txt

configmap/second-configmap created

D:\CKA BY zeal vora\PracticeME\sec6storage>

**$ Create a kube config file using previous file large.txt**

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get configmap second-configmap -o yaml

apiVersion: v1

data:

large-file.txt: "key1=value1\r\nkey2=value2\r\nkey3=value3\r\n"

kind: ConfigMap

metadata:

creationTimestamp: "2025-02-18T07:01:45Z"

name: second-configmap

namespace: default

resourceVersion: "76845"

uid: 550a325b-d87e-46d8-9dce-264b749d5bcb

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl delete configmap second-configmap

configmap "second-configmap" deleted

apiVersion: v1

kind: ConfigMap

metadata:

  name: manifest-configmap

data:

  key1: "value1"

  key2: "value2"

  big-data: |

    This is Line 1

    This is Line 2

    This is LineE

Config.yaml

C:\Windows\system32>kubectl get configmaps

NAME DATA AGE

kube-root-ca.crt 1 2d12h

C:\Windows\system32>kubectl create configmap first-configmap --from-literal=key1=value1 --from-literal=key2=value2

configmap/first-configmap created

C:\Windows\system32>kubectl get configmap

NAME DATA AGE

first-configmap 2 16s

kube-root-ca.crt 1 2d12h

C:\Windows\system32>kubectl describe configmap first-configmap

Name: first-configmap

Namespace: default

Labels: <none>

Annotations: <none>

Data

====

key1:

----

value1

key2:

----

value2

BinaryData

====

Events: <none>

C:\Windows\system32>kubectl get configmap first-configmap -o yaml

apiVersion: v1

data:

key1: value1

key2: value2

kind: ConfigMap

metadata:

creationTimestamp: "2025-02-18T06:54:18Z"

name: first-configmap

namespace: default

resourceVersion: "76250"

uid: 3b980762-38b3-42e5-a0c0-211c9616e8fc

C:\Windows\system32>

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl delete configmap manifest-configmap

configmap "manifest-configmap" deleted

**$ HOW POD will access the data of config maps**

apiVersion: v1

kind: ConfigMap

metadata:

  name: demo-configmap

data:

  DB\_HOST: "172.31.10.30:3306"

  DB\_USER: "dbadmin"

  DB\_PASS: "db!2312$#"

  APP\_MODE: "production"

  APP\_CAPACITY: "100%"

  large-data: |

    This is Line 1

    This is Line 2

    This is Line 3

Config.yaml

**We need to define volume inside pod , we need to specify config inside volume , so that pod can connect to config maps**

apiVersion: v1

kind: ConfigMap

metadata:

  name: demo-configmap

data:

  DB\_HOST: "172.31.10.30:3306"

  DB\_USER: "dbadmin"

  DB\_PASS: "db!2312$#"

  APP\_MODE: "production"

  APP\_CAPACITY: "100%"

  large-data: |

    This is Line 1

    This is Line 2

    This is Line 3

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl create -f configmap.yaml

configmap/demo-configmap created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get configmap demo-configmap -o yaml

apiVersion: v1

data:

APP\_CAPACITY: 100%

APP\_MODE: production

DB\_HOST: 172.31.10.30:3306

DB\_PASS: db!2312$#

DB\_USER: dbadmin

large-data: |

This is Line 1

This is Line 2

This is Line 3

kind: ConfigMap

metadata:

creationTimestamp: "2025-02-18T07:50:47Z"

name: demo-configmap

namespace: default

resourceVersion: "80746"

uid: 5f1b48da-63a5-4756-8271-6abbdaabb671

D:\CKA BY zeal vora\PracticeME\sec6storage>

**$ Each key-value pair appear in the config file**

apiVersion: v1

kind: ConfigMap

metadata:

  name: demo-configmap

data:

  DB\_HOST: "172.31.10.30:3306"

  DB\_USER: "dbadmin"

  DB\_PASS: "db!2312$#"

  APP\_MODE: "production"

  APP\_CAPACITY: "100%"

  large-data: |

    This is Line 1

    This is Line 2

    This is Line 3

apiVersion: v1

kind: Pod

metadata:

  name: configmap-volume-pod

spec:

  containers:

    - name: nginx-container

      image: nginx

      volumeMounts:

        - name: config-volume

          mountPath: /etc/config

  volumes:

    - name: config-volume

      configMap:

        name: demo-configmap

pod.yaml

**$ In this mount path key-value pair resides**

D:\CKA BY zeal vora\PracticeME\sec6storage> kubectl apply -f pod.yaml

pod/configmap-volume-pod created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

configmap-volume-pod 1/1 Running 0 3h

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it configmap-volume-pod

error: you must specify at least one command for the container

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it configmap-volume-pod -- bash

root@configmap-volume-pod:/# cd /etc/config

root@configmap-volume-pod:/etc/config# ls

APP\_CAPACITY APP\_MODE DB\_HOST DB\_PASS DB\_USER large-data

root@configmap-volume-pod:/etc/config# cat DB\_HOST

172.31.10.30:3306root@configmap-volume-pod:/etc/config# cat large-data

This is Line 1

This is Line 2

This is Line 3

root@configmap-volume-pod:/etc/config#

**$ here large data is key and db\_host is key and – this is line1 are value , this is line2 are value**

apiVersion: v1

kind: Pod

metadata:

  name: configmap-env-pod

spec:

  containers:

    - name: nginx-container

      image: nginx

      env:

        - name: MODE\_APP

          valueFrom:

            configMapKeyRef:

              name: demo-configmap

              key: APP\_MODE

**$ value of this env taken from this key mode ,Mode\_APP is actually contianeing value of env variable**

#start for here

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod.yaml

pod/configmap-volume-pod created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

configmap-volume-pod 1/1 Running 0 3h

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it configmap-volume-pod

error: you must specify at least one command for the container

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it configmap-volume-pod -- bash

root@configmap-volume-pod:/# cd /etc/config

root@configmap-volume-pod:/etc/config# ls

APP\_CAPACITY APP\_MODE DB\_HOST DB\_PASS DB\_USER large-data

root@configmap-volume-pod:/etc/config# cat DB\_HOST

172.31.10.30:3306root@configmap-volume-pod:/etc/config# cat large-data

This is Line 1

This is Line 2

This is Line 3

root@configmap-volume-pod:/etc/config# exit

exit

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod-env.yaml

pod/configmap-env-pod created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

configmap-env-pod 1/1 Running 0 14s

configmap-volume-pod 1/1 Running 0 3h14m

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it configmap-env-pod -- bash

root@configmap-env-pod:/# echo $MODE\_APP

production

root@configmap-env-pod:/#

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl delete pod configmap-env-pod

pod "configmap-env-pod" deleted

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get config

error: the server doesn't have a resource type "config"

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

configmap-volume-pod 1/1 Running 0 3h34m

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl delete pod configmap-volume-pod

pod "configmap-volume-pod" deleted

apiVersion: v1

kind: Pod

metadata:

  name: insecure-pod

spec:

  containers:

    - name: demo-container

      image: busybox:latest

      command: ["sleep", "36000"]

      volumeMounts:

        - name: host-root

          mountPath: /host

  volumes:

    - name: host-root

      hostPath:

        path: /

pod.yaml

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod1.yaml

pod/insecure-pod created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

insecure-pod 1/1 Running 0 32s

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it insecure-pod -- sh

/ # id

uid=0(root) gid=0(root) groups=0(root),10(wheel)

/ # ls

bin dev etc home host lib lib64 proc root sys tmp usr var

/ #

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl delete pod insecure-pod

pod "insecure-pod" deleted

**$ Using security-context run contianers with non- root user privliges**

apiVersion: v1

kind: Pod

metadata:

  name: controlled-pod

spec:

  securityContext:

   runAsUser: 1000

   runAsGroup: 2000

   fsGroup: 3000

  containers:

    - name: demo-container

      image: busybox:latest

      command: ["sleep", "36000"]

      volumeMounts:

        - name: host-root

          mountPath: /host

  volumes:

    - name: host-root

      hostPath:

        path: /

**$ Give access of host files to appropriate pds**

**$on host path root is mounted**

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

controlled-pod 1/1 Running 0 11m

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it controlled-pod

error: you must specify at least one command for the container

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it controlled-pod -- sh

~ $ id

uid=1000 gid=2000 groups=2000,3000

~ $ cd /host

/host $ ls

A H O Users b etc init mnt r tmp z

Applications I P V bin f j n root u

B J Q Volumes boot g k o run usr

C K R W c h l opt s v

D L S X cores home lib p sbin var

E Library System Y d host\_mnt lib64 private srv w

F M T Z dev i m proc sys x

G N U a e image media q t y

/host $ cd /boot

sh: cd: can't cd to /boot: No such file or directory

/host $ cd boot

/host/boot $ cd grub/

sh: cd: can't cd to grub/: No such file or directory

/host/boot $ vi grub.cfg

$ we cant edit this file no modifn allow

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

controlled-pod 1/1 Running 0 11m

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it controlled-pod

error: you must specify at least one command for the container

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it controlled-pod -- sh

~ $ id

uid=1000 gid=2000 groups=2000,3000

~ $ cd /host

/host $ ls

A H O Users b etc init mnt r tmp z

Applications I P V bin f j n root u

B J Q Volumes boot g k o run usr

C K R W c h l opt s v

D L S X cores home lib p sbin var

E Library System Y d host\_mnt lib64 private srv w

F M T Z dev i m proc sys x

G N U a e image media q t y

/host $ cd /boot

sh: cd: can't cd to /boot: No such file or directory

/host $ cd boot

/host/boot $ cd grub/

sh: cd: can't cd to grub/: No such file or directory

/host/boot $ cd ..

/host $ ls

A H O Users b etc init mnt r tmp z

Applications I P V bin f j n root u

B J Q Volumes boot g k o run usr

C K R W c h l opt s v

D L S X cores home lib p sbin var

E Library System Y d host\_mnt lib64 private srv w

F M T Z dev i m proc sys x

G N U a e image media q t y

/host $ cd tmp

/host/tmp $ touch test.txt

/host/tmp $ ls -ls

total 0

0 drwxr-xr-x 4 root root 80 Feb 18 02:54 01-docker

0 -rw-r--r-- 1 1000 2000 0 Feb 18 12:04 test.txt

/host/tmp $

1k , 2k ,3k are basicall premiissons related to file associated with it

**Kubertntes secrets**

Secretes are or data are stored in etcd in plain text

We can use rbac

We can encode this sercrets they are not in plain text

**Create secret in kubernetes**

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl create secret generic auth-secret --from-literal=admin=password

secret/auth-secret created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get secret

NAME TYPE DATA AGE

auth-secret Opaque 1 14s

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl describe secret auth-secret

Name: auth-secret

Namespace: default

Labels: <none>

Annotations: <none>

Type: Opaque

Data

====

admin: 8 bytes

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get secret auth-secret -o yaml

apiVersion: v1

data:

admin: cGFzc3dvcmQ=

kind: Secret

metadata:

creationTimestamp: "2025-02-18T15:14:09Z"

name: auth-secret

namespace: default

resourceVersion: "95899"

uid: 23d598b6-3617-4516-9397-a29151665471

type: Opaque

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl create secret generic auth-secret --dry-run=client -o yaml --from-literal=admin=password

apiVersion: v1

data:

admin: cGFzc3dvcmQ=

kind: Secret

metadata:

creationTimestamp: null

name: auth-secret

D:\CKA BY zeal vora\PracticeME\sec6storage>

**$ after creating secrets mount it to the pod , first create volume then mount it to the path /etc/secrets**

**$ mount it to the specific mount path,inside volume specify secret**

apiVersion: v1

kind: Pod

metadata:

  name: demo-pod

spec:

  volumes:

    - name: secret-volume

      secret:

        secretName: auth-secret # Replace with your actual secret name

  containers:

    - name: test-container

      image: nginx

      volumeMounts:

        - name: secret-volume

          mountPath: "/etc/secret-volume"

pod-secrets.yaml

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod-secrets.yaml

pod/demo-pod created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

demo-pod 0/1 ContainerCreating 0 5s

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl get pods

NAME READY STATUS RESTARTS AGE

demo-pod 1/1 Running 0 30s

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it demo-pod -- bash

root@demo-pod:/# cd /etc/secret-volume

root@demo-pod:/etc/secret-volume# ls

admin

root@demo-pod:/etc/secret-volume# cat admin

passwordroot@demo-pod:/etc/secret-volume#

**$ Here we are connecting the pod**

apiVersion: v1

kind: Pod

metadata:

  name: demo-pod-env

spec:

  containers:

    - name: test-container

      image: nginx

      env:

        - name: DB\_PASSWORD

          valueFrom:

            secretKeyRef:

              name: auth-secret

              key: admin

pod-env

**$ Here we are using env variable , to authenticate secret**

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod-secret-env.yaml

pod/demo-pod-env created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it demo-pod-env -- bash

root@demo-pod-env:/# echo $DB\_PASSWORD

password

root@demo-pod-env:/#

**USING ENV TO STORE SECTESTS**

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl apply -f pod-secret-env.yaml

pod/demo-pod-env created

D:\CKA BY zeal vora\PracticeME\sec6storage>kubectl exec -it demo-pod-env -- bash

root@demo-pod-env:/# echo $DB\_PASSWORD

password

root@demo-pod-env:/#

**section 7 – Cluster architecture , configurations , Installations – optional**

**worker node where the pod is actually running**

**container run on pods , and pod is run inside worker node**

worker node

**kubeadm used to run the master node and worker node**

* Seen it but not VIP

Kubeconfig file is veery important and very useful

SECTION – 8 – LOGGING OR MONITORING

Microsoft Windows [Version 10.0.19045.5608]

(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>kubectlg get events

'kubectlg' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>kubectl get events

LAST SEEN TYPE REASON OBJECT MESSAGE

40m Warning PossibleMemoryBackedVolumesOnDisk node/docker-desktop The tmpfs noswap option is not supported. Memory-backed volumes (e.g. secrets, emptyDirs, etc.) might be swapped to disk and should no longer be considered secure.

40m Normal Starting node/docker-desktop Starting kubelet.

40m Warning CgroupV1 node/docker-desktop Cgroup v1 support is in maintenance mode, please migrate to Cgroup v2.

40m Normal NodeHasSufficientMemory node/docker-desktop Node docker-desktop status is now: NodeHasSufficientMemory

40m Normal NodeHasNoDiskPressure node/docker-desktop Node docker-desktop status is now: NodeHasNoDiskPressure

40m Normal NodeHasSufficientPID node/docker-desktop Node docker-desktop status is now: NodeHasSufficientPID

40m Normal NodeAllocatableEnforced node/docker-desktop Updated Node Allocatable limit across pods

40m Normal RegisteredNode node/docker-desktop Node docker-desktop event: Registered Node docker-desktop in Controller

39m Normal Starting node/docker-desktop

C:\Windows\system32>kubectl run event-pod --image=nginx

pod/event-pod created

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 69s

C:\Windows\system32>kubectl get events

LAST SEEN TYPE REASON OBJECT MESSAGE

42m Warning PossibleMemoryBackedVolumesOnDisk node/docker-desktop The tmpfs noswap option is not supported. Memory-backed volumes (e.g. secrets, emptyDirs, etc.) might be swapped to disk and should no longer be considered secure.

42m Normal Starting node/docker-desktop Starting kubelet.

42m Warning CgroupV1 node/docker-desktop Cgroup v1 support is in maintenance mode, please migrate to Cgroup v2.

42m Normal NodeHasSufficientMemory node/docker-desktop Node docker-desktop status is now: NodeHasSufficientMemory

42m Normal NodeHasNoDiskPressure node/docker-desktop Node docker-desktop status is now: NodeHasNoDiskPressure

42m Normal NodeHasSufficientPID node/docker-desktop Node docker-desktop status is now: NodeHasSufficientPID

42m Normal NodeAllocatableEnforced node/docker-desktop Updated Node Allocatable limit across pods

42m Normal RegisteredNode node/docker-desktop Node docker-desktop event: Registered Node docker-desktop in Controller

41m Normal Starting node/docker-desktop

91s Normal Scheduled pod/event-pod Successfully assigned default/event-pod to docker-desktop

89s Normal Pulling pod/event-pod Pulling image "nginx"

84s Normal Pulled pod/event-pod Successfully pulled image "nginx" in 4.929s (4.929s including waiting). Image size: 72180980 bytes.

81s Normal Created pod/event-pod Created container event-pod

78s Normal Started pod/event-pod Started container event-pod

C:\Windows\system32>kubectl get deployment

No resources found in default namespace.

C:\Windows\system32>kubectl get namespace

NAME STATUS AGE

default Active 3d1h

kube-node-lease Active 3d1h

kube-public Active 3d1h

kube-system Active 3d1h

C:\Windows\system32>kubectl get events -n kube-system

LAST SEEN TYPE REASON OBJECT MESSAGE

43m Normal SandboxChanged pod/coredns-7c65d6cfc9-6cwcs Pod sandbox changed, it will be killed and re-created.

43m Normal Pulled pod/coredns-7c65d6cfc9-6cwcs Container image "registry.k8s.io/coredns/coredns:v1.11.3" already present on machine

43m Normal Created pod/coredns-7c65d6cfc9-6cwcs Created container coredns

43m Normal Started pod/coredns-7c65d6cfc9-6cwcs Started container coredns

43m Warning Unhealthy pod/coredns-7c65d6cfc9-6cwcs Readiness probe failed: HTTP probe failed with statuscode: 503

43m Normal SandboxChanged pod/coredns-7c65d6cfc9-mdg9r Pod sandbox changed, it will be killed and re-created.

43m Normal Pulled pod/coredns-7c65d6cfc9-mdg9r Container image "registry.k8s.io/coredns/coredns:v1.11.3" already present on machine

43m Normal Created pod/coredns-7c65d6cfc9-mdg9r Created container coredns

43m Normal Started pod/coredns-7c65d6cfc9-mdg9r Started container coredns

43m Warning Unhealthy pod/coredns-7c65d6cfc9-mdg9r Readiness probe failed: HTTP probe failed with statuscode: 503

42m Normal LeaderElection endpoints/docker.io-hostpath storage-provisioner\_0a832d4b-aecd-418d-bfc6-4daf1ca0015e became leader

44m Normal SandboxChanged pod/etcd-docker-desktop Pod sandbox changed, it will be killed and re-created.

44m Normal Pulled pod/etcd-docker-desktop Container image "registry.k8s.io/etcd:3.5.15-0" already present on machine

44m Normal Created pod/etcd-docker-desktop Created container etcd

44m Normal Started pod/etcd-docker-desktop Started container etcd

44m Normal SandboxChanged pod/kube-apiserver-docker-desktop Pod sandbox changed, it will be killed and re-created.

44m Normal Pulled pod/kube-apiserver-docker-desktop Container image "registry.k8s.io/kube-apiserver:v1.31.4" already present on machine

44m Normal Created pod/kube-apiserver-docker-desktop Created container kube-apiserver

44m Normal Started pod/kube-apiserver-docker-desktop Started container kube-apiserver

44m Warning Unhealthy pod/kube-apiserver-docker-desktop Startup probe failed: HTTP probe failed with statuscode: 500

36m Warning Unhealthy pod/kube-apiserver-docker-desktop Readiness probe failed: HTTP probe failed with statuscode: 500

44m Normal SandboxChanged pod/kube-controller-manager-docker-desktop Pod sandbox changed, it will be killed and re-created.

44m Normal Pulled pod/kube-controller-manager-docker-desktop Container image "registry.k8s.io/kube-controller-manager:v1.31.4" already present on machine

44m Normal Created pod/kube-controller-manager-docker-desktop Created container kube-controller-manager

44m Normal Started pod/kube-controller-manager-docker-desktop Started container kube-controller-manager

43m Normal SandboxChanged pod/kube-proxy-vvxm8 Pod sandbox changed, it will be killed and re-created.

43m Normal Pulled pod/kube-proxy-vvxm8 Container image "registry.k8s.io/kube-proxy:v1.31.4" already present on machine

43m Normal Created pod/kube-proxy-vvxm8 Created container kube-proxy

43m Normal Started pod/kube-proxy-vvxm8 Started container kube-proxy

44m Normal SandboxChanged pod/kube-scheduler-docker-desktop Pod sandbox changed, it will be killed and re-created.

44m Normal Pulled pod/kube-scheduler-docker-desktop Container image "registry.k8s.io/kube-scheduler:v1.31.4" already present on machine

44m Normal Created pod/kube-scheduler-docker-desktop Created container kube-scheduler

44m Normal Started pod/kube-scheduler-docker-desktop Started container kube-scheduler

43m Normal LeaderElection lease/kube-scheduler docker-desktop\_ff128047-89c1-4058-b578-e4462ec8a452 became leader

43m Normal SandboxChanged pod/storage-provisioner Pod sandbox changed, it will be killed and re-created.

43m Normal Pulled pod/storage-provisioner Container image "docker/desktop-storage-provisioner:v2.0" already present on machine

43m Normal Created pod/storage-provisioner Created container storage-provisioner

43m Normal Started pod/storage-provisioner Started container storage-provisioner

43m Warning BackOff pod/storage-provisioner Back-off restarting failed container storage-provisioner in pod storage-provisioner\_kube-system(24cca0ee-20f7-4060-9d9c-00a96bb1df4d)

43m Normal SandboxChanged pod/vpnkit-controller Pod sandbox changed, it will be killed and re-created.

43m Normal Pulled pod/vpnkit-controller Container image "docker/desktop-vpnkit-controller:dc331cb22850be0cdd97c84a9cfecaf44a1afb6e" already present on machine

43m Normal Created pod/vpnkit-controller Created container vpnkit-controller

43m Normal Started pod/vpnkit-controller Started container vpnkit-controller

C:\Windows\system32>kubectl get pods --all-namespaces

NAMESPACE NAME READY STATUS RESTARTS AGE

default event-pod 1/1 Running 0 5m56s

kube-system coredns-7c65d6cfc9-6cwcs 1/1 Running 3 (47m ago) 3d1h

kube-system coredns-7c65d6cfc9-mdg9r 1/1 Running 3 (47m ago) 3d1h

kube-system etcd-docker-desktop 1/1 Running 3 (47m ago) 3d1h

kube-system kube-apiserver-docker-desktop 1/1 Running 3 (47m ago) 3d1h

kube-system kube-controller-manager-docker-desktop 1/1 Running 3 (47m ago) 3d1h

kube-system kube-proxy-vvxm8 1/1 Running 3 (47m ago) 3d1h

kube-system kube-scheduler-docker-desktop 1/1 Running 3 (47m ago) 3d1h

kube-system storage-provisioner 1/1 Running 6 (45m ago) 3d1h

kube-system vpnkit-controller 1/1 Running 3 (47m ago) 3d1h

C:\Windows\system32>kubectl get pods --all-namespaces --field-selector metadata.namespace=default

NAMESPACE NAME READY STATUS RESTARTS AGE

default event-pod 1/1 Running 0 7m51s

C:\Windows\system32>kubectl get pods --all-namespaces --field-selector metadata.namespace!=default

NAMESPACE NAME READY STATUS RESTARTS AGE

kube-system coredns-7c65d6cfc9-6cwcs 1/1 Running 3 (49m ago) 3d1h

kube-system coredns-7c65d6cfc9-mdg9r 1/1 Running 3 (49m ago) 3d1h

kube-system etcd-docker-desktop 1/1 Running 3 (49m ago) 3d1h

kube-system kube-apiserver-docker-desktop 1/1 Running 3 (49m ago) 3d1h

kube-system kube-controller-manager-docker-desktop 1/1 Running 3 (49m ago) 3d1h

kube-system kube-proxy-vvxm8 1/1 Running 3 (49m ago) 3d1h

kube-system kube-scheduler-docker-desktop 1/1 Running 3 (49m ago) 3d1h

kube-system storage-provisioner 1/1 Running 6 (48m ago) 3d1h

kube-system vpnkit-controller 1/1 Running 3 (49m ago) 3d1h

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 9m17s

C:\Windows\system32>kubectl create deployment test-deployment --replicas 3 --image=nginx

deployment.apps/test-deployment created

C:\Windows\system32>kubectl get events

LAST SEEN TYPE REASON OBJECT MESSAGE

51m Warning PossibleMemoryBackedVolumesOnDisk node/docker-desktop The tmpfs noswap option is not supported. Memory-backed volumes (e.g. secrets, emptyDirs, etc.) might be swapped to disk and should no longer be considered secure.

51m Normal Starting node/docker-desktop Starting kubelet.

51m Warning CgroupV1 node/docker-desktop Cgroup v1 support is in maintenance mode, please migrate to Cgroup v2.

51m Normal NodeHasSufficientMemory node/docker-desktop Node docker-desktop status is now: NodeHasSufficientMemory

51m Normal NodeHasNoDiskPressure node/docker-desktop Node docker-desktop status is now: NodeHasNoDiskPressure

51m Normal NodeHasSufficientPID node/docker-desktop Node docker-desktop status is now: NodeHasSufficientPID

51m Normal NodeAllocatableEnforced node/docker-desktop Updated Node Allocatable limit across pods

50m Normal RegisteredNode node/docker-desktop Node docker-desktop event: Registered Node docker-desktop in Controller

50m Normal Starting node/docker-desktop

10m Normal Scheduled pod/event-pod Successfully assigned default/event-pod to docker-desktop

10m Normal Pulling pod/event-pod Pulling image "nginx"

10m Normal Pulled pod/event-pod Successfully pulled image "nginx" in 4.929s (4.929s including waiting). Image size: 72180980 bytes.

9m57s Normal Created pod/event-pod Created container event-pod

9m54s Normal Started pod/event-pod Started container event-pod

19s Normal Scheduled pod/test-deployment-8687dc8bff-b57xd Successfully assigned default/test-deployment-8687dc8bff-b57xd to docker-desktop

15s Normal Pulling pod/test-deployment-8687dc8bff-b57xd Pulling image "nginx"

9s Normal Pulled pod/test-deployment-8687dc8bff-b57xd Successfully pulled image "nginx" in 5.341s (5.342s including waiting). Image size: 72180980 bytes.

8s Normal Created pod/test-deployment-8687dc8bff-b57xd Created container nginx

7s Normal Started pod/test-deployment-8687dc8bff-b57xd Started container nginx

19s Normal Scheduled pod/test-deployment-8687dc8bff-gcx5m Successfully assigned default/test-deployment-8687dc8bff-gcx5m to docker-desktop

13s Normal Pulling pod/test-deployment-8687dc8bff-gcx5m Pulling image "nginx"

5s Normal Pulled pod/test-deployment-8687dc8bff-gcx5m Successfully pulled image "nginx" in 4.452s (8.47s including waiting). Image size: 72180980 bytes.

4s Normal Created pod/test-deployment-8687dc8bff-gcx5m Created container nginx

4s Normal Started pod/test-deployment-8687dc8bff-gcx5m Started container nginx

19s Normal Scheduled pod/test-deployment-8687dc8bff-j24cc Successfully assigned default/test-deployment-8687dc8bff-j24cc to docker-desktop

13s Normal Pulling pod/test-deployment-8687dc8bff-j24cc Pulling image "nginx"

0s Normal Pulled pod/test-deployment-8687dc8bff-j24cc Successfully pulled image "nginx" in 4.503s (12.758s including waiting). Image size: 72180980 bytes.

19s Normal SuccessfulCreate replicaset/test-deployment-8687dc8bff Created pod: test-deployment-8687dc8bff-b57xd

19s Normal SuccessfulCreate replicaset/test-deployment-8687dc8bff Created pod: test-deployment-8687dc8bff-j24cc

19s Normal SuccessfulCreate replicaset/test-deployment-8687dc8bff Created pod: test-deployment-8687dc8bff-gcx5m

19s Normal ScalingReplicaSet deployment/test-deployment Scaled up replica set test-deployment-8687dc8bff to 3

C:\Windows\system32>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 11m

test-deployment-8687dc8bff-b57xd 1/1 Running 0 87s

test-deployment-8687dc8bff-gcx5m 1/1 Running 0 87s

test-deployment-8687dc8bff-j24cc 1/1 Running 0 87s

C:\Windows\system32>kubectl get events -o json

{

"apiVersion": "v1",

"items": [

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:31Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:31Z",

"message": "The tmpfs noswap option is not supported. Memory-backed volumes (e.g. secrets, emptyDirs, etc.) might be swapped to disk and should no longer be considered secure.",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e83199398e88e",

"namespace": "default",

"resourceVersion": "29602",

"uid": "daf8d275-0efc-4b39-a1af-70b2ccc7c9f1"

},

"reason": "PossibleMemoryBackedVolumesOnDisk",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Warning"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:31Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:31Z",

"message": "Starting kubelet.",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e83199842141a",

"namespace": "default",

"resourceVersion": "29603",

"uid": "ac4e9099-1a08-4be9-a7a0-c1d8db9f50e5"

},

"reason": "Starting",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:31Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:31Z",

"message": "Cgroup v1 support is in maintenance mode, please migrate to Cgroup v2.",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e831999942850",

"namespace": "default",

"resourceVersion": "29604",

"uid": "d9646e3f-cb30-449f-a0a7-f4bea58e3c27"

},

"reason": "CgroupV1",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Warning"

},

{

"apiVersion": "v1",

"count": 8,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:32Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:38Z",

"message": "Node docker-desktop status is now: NodeHasSufficientMemory",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e8319b80b7096",

"namespace": "default",

"resourceVersion": "29625",

"uid": "b12ffe85-0484-4c18-a9b4-10fc71dd1c86"

},

"reason": "NodeHasSufficientMemory",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 7,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:32Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:38Z",

"message": "Node docker-desktop status is now: NodeHasNoDiskPressure",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e8319b80b9486",

"namespace": "default",

"resourceVersion": "29626",

"uid": "051668f3-4ae3-40af-8eb6-d2789e48d3ee"

},

"reason": "NodeHasNoDiskPressure",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 6,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:32Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:38Z",

"message": "Node docker-desktop status is now: NodeHasSufficientPID",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e8319b80babf6",

"namespace": "default",

"resourceVersion": "29623",

"uid": "1cbae735-6747-4277-8655-08cb58e3f516"

},

"reason": "NodeHasSufficientPID",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:37:35Z",

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:37:35Z",

"message": "Updated Node Allocatable limit across pods",

"metadata": {

"creationTimestamp": "2025-03-20T12:37:57Z",

"name": "docker-desktop.182e831a8b0be35e",

"namespace": "default",

"resourceVersion": "29608",

"uid": "4970b958-5eae-4a0d-affc-ddc1aa8d6a4b"

},

"reason": "NodeAllocatableEnforced",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T12:38:01Z",

"involvedObject": {

"apiVersion": "v1",

"kind": "Node",

"name": "docker-desktop",

"uid": "dc8d134e-5920-4674-94a3-e2e4685a020a"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T12:38:01Z",

"message": "Node docker-desktop event: Registered Node docker-desktop in Controller",

"metadata": {

"creationTimestamp": "2025-03-20T12:38:01Z",

"name": "docker-desktop.182e8320816a36ce",

"namespace": "default",

"resourceVersion": "29645",

"uid": "7a9d9125-66c6-4e7b-bf40-3c8327609916"

},

"reason": "RegisteredNode",

"reportingComponent": "node-controller",

"reportingInstance": "",

"source": {

"component": "node-controller"

},

"type": "Normal"

},

{

"action": "StartKubeProxy",

"apiVersion": "v1",

"eventTime": "2025-03-20T12:38:27.297600Z",

"firstTimestamp": null,

"involvedObject": {

"kind": "Node",

"name": "docker-desktop",

"uid": "docker-desktop"

},

"kind": "Event",

"lastTimestamp": null,

"metadata": {

"creationTimestamp": "2025-03-20T12:38:27Z",

"name": "docker-desktop.182e832684b7e9f7",

"namespace": "default",

"resourceVersion": "29701",

"uid": "8f2e0120-05ba-41a1-8037-098d2f45b970"

},

"reason": "Starting",

"reportingComponent": "kube-proxy",

"reportingInstance": "kube-proxy-docker-desktop",

"source": {},

"type": "Normal"

},

{

"action": "Binding",

"apiVersion": "v1",

"eventTime": "2025-03-20T13:18:35.478352Z",

"firstTimestamp": null,

"involvedObject": {

"apiVersion": "v1",

"kind": "Pod",

"name": "event-pod",

"namespace": "default",

"resourceVersion": "32731",

"uid": "42c861c4-4fa5-4f37-8d6b-82a0f338bb7e"

},

"kind": "Event",

"lastTimestamp": null,

"message": "Successfully assigned default/event-pod to docker-desktop",

"metadata": {

"creationTimestamp": "2025-03-20T13:18:35Z",

"name": "event-pod.182e8557377d3857",

"namespace": "default",

"resourceVersion": "32734",

"uid": "a6db824f-8bdd-4e50-8e05-28be751e7642"

},

"reason": "Scheduled",

"reportingComponent": "default-scheduler",

"reportingInstance": "default-scheduler-docker-desktop",

"source": {},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:18:37Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{event-pod}",

"kind": "Pod",

"name": "event-pod",

"namespace": "default",

"resourceVersion": "32732",

"uid": "42c861c4-4fa5-4f37-8d6b-82a0f338bb7e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:18:37Z",

"message": "Pulling image \"nginx\"",

"metadata": {

"creationTimestamp": "2025-03-20T13:18:37Z",

"name": "event-pod.182e8557c34adad5",

"namespace": "default",

"resourceVersion": "32738",

"uid": "b7bfa4c0-963a-4fd0-8bb8-f3e78ed7a0be"

},

"reason": "Pulling",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:18:42Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{event-pod}",

"kind": "Pod",

"name": "event-pod",

"namespace": "default",

"resourceVersion": "32732",

"uid": "42c861c4-4fa5-4f37-8d6b-82a0f338bb7e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:18:42Z",

"message": "Successfully pulled image \"nginx\" in 4.929s (4.929s including waiting). Image size: 72180980 bytes.",

"metadata": {

"creationTimestamp": "2025-03-20T13:18:42Z",

"name": "event-pod.182e8558e91c648a",

"namespace": "default",

"resourceVersion": "32746",

"uid": "b81270c1-18b6-4532-a811-1262d59b440c"

},

"reason": "Pulled",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:18:45Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{event-pod}",

"kind": "Pod",

"name": "event-pod",

"namespace": "default",

"resourceVersion": "32732",

"uid": "42c861c4-4fa5-4f37-8d6b-82a0f338bb7e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:18:45Z",

"message": "Created container event-pod",

"metadata": {

"creationTimestamp": "2025-03-20T13:18:45Z",

"name": "event-pod.182e8559a7569ceb",

"namespace": "default",

"resourceVersion": "32750",

"uid": "d9fe7d2c-c589-4069-a63d-ec3379c37695"

},

"reason": "Created",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:18:48Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{event-pod}",

"kind": "Pod",

"name": "event-pod",

"namespace": "default",

"resourceVersion": "32732",

"uid": "42c861c4-4fa5-4f37-8d6b-82a0f338bb7e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:18:48Z",

"message": "Started container event-pod",

"metadata": {

"creationTimestamp": "2025-03-20T13:18:48Z",

"name": "event-pod.182e855a2e6d297d",

"namespace": "default",

"resourceVersion": "32753",

"uid": "11a7c036-86b9-461f-be68-e3dcb6283adc"

},

"reason": "Started",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"action": "Binding",

"apiVersion": "v1",

"eventTime": "2025-03-20T13:28:23.611286Z",

"firstTimestamp": null,

"involvedObject": {

"apiVersion": "v1",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-b57xd",

"namespace": "default",

"resourceVersion": "33469",

"uid": "6fdd7f48-db20-4b96-989d-3ff1623ce16c"

},

"kind": "Event",

"lastTimestamp": null,

"message": "Successfully assigned default/test-deployment-8687dc8bff-b57xd to docker-desktop",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff-b57xd.182e85e026ee7e4b",

"namespace": "default",

"resourceVersion": "33481",

"uid": "f22c8f27-5afa-423b-bb72-c12d67b0596a"

},

"reason": "Scheduled",

"reportingComponent": "default-scheduler",

"reportingInstance": "default-scheduler-docker-desktop",

"source": {},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:27Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-b57xd",

"namespace": "default",

"resourceVersion": "33474",

"uid": "6fdd7f48-db20-4b96-989d-3ff1623ce16c"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:27Z",

"message": "Pulling image \"nginx\"",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:27Z",

"name": "test-deployment-8687dc8bff-b57xd.182e85e11c995eac",

"namespace": "default",

"resourceVersion": "33494",

"uid": "10c68f6a-0732-400f-b7ae-fb832ec92df4"

},

"reason": "Pulling",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:33Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-b57xd",

"namespace": "default",

"resourceVersion": "33474",

"uid": "6fdd7f48-db20-4b96-989d-3ff1623ce16c"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:33Z",

"message": "Successfully pulled image \"nginx\" in 5.341s (5.342s including waiting). Image size: 72180980 bytes.",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:33Z",

"name": "test-deployment-8687dc8bff-b57xd.182e85e25b04bfd2",

"namespace": "default",

"resourceVersion": "33504",

"uid": "95436073-0770-491d-831b-f90631d455b5"

},

"reason": "Pulled",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:34Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-b57xd",

"namespace": "default",

"resourceVersion": "33474",

"uid": "6fdd7f48-db20-4b96-989d-3ff1623ce16c"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:34Z",

"message": "Created container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:34Z",

"name": "test-deployment-8687dc8bff-b57xd.182e85e2a378cfe8",

"namespace": "default",

"resourceVersion": "33506",

"uid": "c4bb6f46-6193-4d64-82d5-96fd1e718af4"

},

"reason": "Created",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:35Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-b57xd",

"namespace": "default",

"resourceVersion": "33474",

"uid": "6fdd7f48-db20-4b96-989d-3ff1623ce16c"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:35Z",

"message": "Started container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:35Z",

"name": "test-deployment-8687dc8bff-b57xd.182e85e2cfc449b2",

"namespace": "default",

"resourceVersion": "33508",

"uid": "8d95a22d-b70e-4e43-8eac-7ff57a4375b3"

},

"reason": "Started",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"action": "Binding",

"apiVersion": "v1",

"eventTime": "2025-03-20T13:28:23.652973Z",

"firstTimestamp": null,

"involvedObject": {

"apiVersion": "v1",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-gcx5m",

"namespace": "default",

"resourceVersion": "33472",

"uid": "5d8a3bec-ac8c-4995-b146-23af701a5282"

},

"kind": "Event",

"lastTimestamp": null,

"message": "Successfully assigned default/test-deployment-8687dc8bff-gcx5m to docker-desktop",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff-gcx5m.182e85e0296aa094",

"namespace": "default",

"resourceVersion": "33485",

"uid": "0d0c172b-1246-45f8-984f-d4747cb31381"

},

"reason": "Scheduled",

"reportingComponent": "default-scheduler",

"reportingInstance": "default-scheduler-docker-desktop",

"source": {},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:29Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-gcx5m",

"namespace": "default",

"resourceVersion": "33478",

"uid": "5d8a3bec-ac8c-4995-b146-23af701a5282"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:29Z",

"message": "Pulling image \"nginx\"",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:29Z",

"name": "test-deployment-8687dc8bff-gcx5m.182e85e16b8a32f7",

"namespace": "default",

"resourceVersion": "33497",

"uid": "8dd7a6a5-f24a-41c9-a067-369a1c00cb68"

},

"reason": "Pulling",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:37Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-gcx5m",

"namespace": "default",

"resourceVersion": "33478",

"uid": "5d8a3bec-ac8c-4995-b146-23af701a5282"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:37Z",

"message": "Successfully pulled image \"nginx\" in 4.452s (8.47s including waiting). Image size: 72180980 bytes.",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:37Z",

"name": "test-deployment-8687dc8bff-gcx5m.182e85e36465c31e",

"namespace": "default",

"resourceVersion": "33513",

"uid": "55c024a4-8087-4df2-85a8-63bb70165df0"

},

"reason": "Pulled",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:38Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-gcx5m",

"namespace": "default",

"resourceVersion": "33478",

"uid": "5d8a3bec-ac8c-4995-b146-23af701a5282"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:38Z",

"message": "Created container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:38Z",

"name": "test-deployment-8687dc8bff-gcx5m.182e85e396eeefcf",

"namespace": "default",

"resourceVersion": "33517",

"uid": "cb899497-5826-45bd-9050-3132e621c556"

},

"reason": "Created",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:38Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-gcx5m",

"namespace": "default",

"resourceVersion": "33478",

"uid": "5d8a3bec-ac8c-4995-b146-23af701a5282"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:38Z",

"message": "Started container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:38Z",

"name": "test-deployment-8687dc8bff-gcx5m.182e85e3b69bb45e",

"namespace": "default",

"resourceVersion": "33519",

"uid": "f998c5cc-e9cc-41c4-960b-a54ad0ba91cc"

},

"reason": "Started",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"action": "Binding",

"apiVersion": "v1",

"eventTime": "2025-03-20T13:28:23.650201Z",

"firstTimestamp": null,

"involvedObject": {

"apiVersion": "v1",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-j24cc",

"namespace": "default",

"resourceVersion": "33473",

"uid": "2c1c72e1-b7f5-473c-a980-0e399852d509"

},

"kind": "Event",

"lastTimestamp": null,

"message": "Successfully assigned default/test-deployment-8687dc8bff-j24cc to docker-desktop",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff-j24cc.182e85e029405022",

"namespace": "default",

"resourceVersion": "33484",

"uid": "eee29815-18c2-435b-a64b-6631c12fc4ab"

},

"reason": "Scheduled",

"reportingComponent": "default-scheduler",

"reportingInstance": "default-scheduler-docker-desktop",

"source": {},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:29Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-j24cc",

"namespace": "default",

"resourceVersion": "33479",

"uid": "2c1c72e1-b7f5-473c-a980-0e399852d509"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:29Z",

"message": "Pulling image \"nginx\"",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:29Z",

"name": "test-deployment-8687dc8bff-j24cc.182e85e17859e687",

"namespace": "default",

"resourceVersion": "33499",

"uid": "2e6862c4-79f4-433b-90f0-bbb0818ad2da"

},

"reason": "Pulling",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:42Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-j24cc",

"namespace": "default",

"resourceVersion": "33479",

"uid": "2c1c72e1-b7f5-473c-a980-0e399852d509"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:42Z",

"message": "Successfully pulled image \"nginx\" in 4.503s (12.758s including waiting). Image size: 72180980 bytes.",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:42Z",

"name": "test-deployment-8687dc8bff-j24cc.182e85e470ce043e",

"namespace": "default",

"resourceVersion": "33525",

"uid": "c32b5fc1-64d7-4ae3-87e4-19a01f318c0e"

},

"reason": "Pulled",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:43Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-j24cc",

"namespace": "default",

"resourceVersion": "33479",

"uid": "2c1c72e1-b7f5-473c-a980-0e399852d509"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:43Z",

"message": "Created container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:43Z",

"name": "test-deployment-8687dc8bff-j24cc.182e85e4c2c23a56",

"namespace": "default",

"resourceVersion": "33531",

"uid": "48d9db84-ab20-4d23-9f32-8a695a0f8ce2"

},

"reason": "Created",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:44Z",

"involvedObject": {

"apiVersion": "v1",

"fieldPath": "spec.containers{nginx}",

"kind": "Pod",

"name": "test-deployment-8687dc8bff-j24cc",

"namespace": "default",

"resourceVersion": "33479",

"uid": "2c1c72e1-b7f5-473c-a980-0e399852d509"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:44Z",

"message": "Started container nginx",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:44Z",

"name": "test-deployment-8687dc8bff-j24cc.182e85e4ef7b8395",

"namespace": "default",

"resourceVersion": "33532",

"uid": "e021f93c-384f-4de9-b2eb-05903a8902bf"

},

"reason": "Started",

"reportingComponent": "kubelet",

"reportingInstance": "docker-desktop",

"source": {

"component": "kubelet",

"host": "docker-desktop"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:23Z",

"involvedObject": {

"apiVersion": "apps/v1",

"kind": "ReplicaSet",

"name": "test-deployment-8687dc8bff",

"namespace": "default",

"resourceVersion": "33467",

"uid": "5d23aa41-e11a-419e-a364-095d132c131e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:23Z",

"message": "Created pod: test-deployment-8687dc8bff-b57xd",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff.182e85e0243855a7",

"namespace": "default",

"resourceVersion": "33471",

"uid": "ae5e51ed-96f0-403d-9ca5-89e26645e3f0"

},

"reason": "SuccessfulCreate",

"reportingComponent": "replicaset-controller",

"reportingInstance": "",

"source": {

"component": "replicaset-controller"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:23Z",

"involvedObject": {

"apiVersion": "apps/v1",

"kind": "ReplicaSet",

"name": "test-deployment-8687dc8bff",

"namespace": "default",

"resourceVersion": "33467",

"uid": "5d23aa41-e11a-419e-a364-095d132c131e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:23Z",

"message": "Created pod: test-deployment-8687dc8bff-j24cc",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff.182e85e025b34b0a",

"namespace": "default",

"resourceVersion": "33476",

"uid": "eb316ccd-2968-4b2b-9119-aab7daaab6c1"

},

"reason": "SuccessfulCreate",

"reportingComponent": "replicaset-controller",

"reportingInstance": "",

"source": {

"component": "replicaset-controller"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:23Z",

"involvedObject": {

"apiVersion": "apps/v1",

"kind": "ReplicaSet",

"name": "test-deployment-8687dc8bff",

"namespace": "default",

"resourceVersion": "33467",

"uid": "5d23aa41-e11a-419e-a364-095d132c131e"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:23Z",

"message": "Created pod: test-deployment-8687dc8bff-gcx5m",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment-8687dc8bff.182e85e025b9bc3f",

"namespace": "default",

"resourceVersion": "33480",

"uid": "a2203d02-52d2-46db-b0f5-ea6baaee5a42"

},

"reason": "SuccessfulCreate",

"reportingComponent": "replicaset-controller",

"reportingInstance": "",

"source": {

"component": "replicaset-controller"

},

"type": "Normal"

},

{

"apiVersion": "v1",

"count": 1,

"eventTime": null,

"firstTimestamp": "2025-03-20T13:28:23Z",

"involvedObject": {

"apiVersion": "apps/v1",

"kind": "Deployment",

"name": "test-deployment",

"namespace": "default",

"resourceVersion": "33466",

"uid": "aa19fae8-52d1-43f9-bcda-2d836a9b1e87"

},

"kind": "Event",

"lastTimestamp": "2025-03-20T13:28:23Z",

"message": "Scaled up replica set test-deployment-8687dc8bff to 3",

"metadata": {

"creationTimestamp": "2025-03-20T13:28:23Z",

"name": "test-deployment.182e85e0221b89c3",

"namespace": "default",

"resourceVersion": "33470",

"uid": "421fbaba-147a-4b8e-b8f5-5a838afc5c75"

},

"reason": "ScalingReplicaSet",

"reportingComponent": "deployment-controller",

"reportingInstance": "",

"source": {

"component": "deployment-controller"

},

"type": "Normal"

}

],

"kind": "List",

"metadata": {

"resourceVersion": ""

}

}

C:\Windows\system32>kubectl get events --field-selector involvedObject.name=event-pod

LAST SEEN TYPE REASON OBJECT MESSAGE

14m Normal Scheduled pod/event-pod Successfully assigned default/event-pod to docker-desktop

14m Normal Pulling pod/event-pod Pulling image "nginx"

14m Normal Pulled pod/event-pod Successfully pulled image "nginx" in 4.929s (4.929s including waiting). Image size: 72180980 bytes.

14m Normal Created pod/event-pod Created container event-pod

14m Normal Started pod/event-pod Started container event-pod

C:\Windows\system32>kubectl top pods

error: Metrics API not available

C:\Windows\system32>kubectl top pods --all-namespaces

error: Metrics API not available

C:\Windows\system32>kubectl apply -f https://github.com/kubernetes-sigs/metrics-server/releases/latest/download/components.yaml

serviceaccount/metrics-server created

clusterrole.rbac.authorization.k8s.io/system:aggregated-metrics-reader created

clusterrole.rbac.authorization.k8s.io/system:metrics-server created

rolebinding.rbac.authorization.k8s.io/metrics-server-auth-reader created

clusterrolebinding.rbac.authorization.k8s.io/metrics-server:system:auth-delegator created

clusterrolebinding.rbac.authorization.k8s.io/system:metrics-server created

service/metrics-server created

deployment.apps/metrics-server created

apiservice.apiregistration.k8s.io/v1beta1.metrics.k8s.io created

C:\Windows\system32>kubectl top pods

error: Metrics API not available

C:\Windows\system32>kubectl get pods -n kube-system | grep metrics-server

'grep' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>kubectl get pods -n kube-system | grep metrics-server

'grep' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>C:\Windows\system32>kubectl get pods -n kube-system | grep metrics-server

'C:\Windows\system32' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>'grep' is not recognized as an internal or external command,

''grep'' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>operable program or batch file.

'operable' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>

C:\Windows\system32>C:\Windows\system32>kubectl get pods -n kube-system | grep metrics-server

'C:\Windows\system32' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>'grep' is not recognized as an internal or external command,

''grep'' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>operable program or batch file.

'operable' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>

C:\Windows\system32>C:\Windows\system32>kubectl get pods -n kube-system | findstr "metrics-server"

'C:\Windows\system32' is not recognized as an internal or external command,

operable program or batch file.

C:\Windows\system32>kubectl get pods -n kube-system | findstr "metrics-server"

metrics-server-54bf7cdd6-q5t4g 0/1 ContainerCreating 0 2m38s

C:\Windows\system32>kubectl top pods

error: Metrics API not available

C:\Windows\system32>kubectl top nodes

error: Metrics API not available

C:\Windows\system32>kubectl top pods --all-namespaces

error: Metrics API not available

C:\Windows\system32>docker run --name b-1 -d busybox ping google.com

Unable to find image 'busybox:latest' locally

latest: Pulling from library/busybox

97e70d161e81: Pull complete

Digest: sha256:37f7b378a29ceb4c551b1b5582e27747b855bbfaa73fa11914fe0df028dc581f

Status: Downloaded newer image for busybox:latest

1acbb377edfec0eecb122b3f80514796e32c461b3e5cf149199ee5e2be58a90f

C:\Windows\system32>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1acbb377edfe busybox "ping google.com" 43 seconds ago Up 40 seconds b-1

53c7cea44b4f 124b44bfc9cc "/docker-entrypoint.…" 22 minutes ago Up 22 minutes k8s\_nginx\_test-deployment-8687dc8bff-j24cc\_default\_2c1c72e1-b7f5-473c-a980-0e399852d509\_0

77fc7c44fc83 124b44bfc9cc "/docker-entrypoint.…" 22 minutes ago Up 22 minutes k8s\_nginx\_test-deployment-8687dc8bff-gcx5m\_default\_5d8a3bec-ac8c-4995-b146-23af701a5282\_0

2ab3b04567bf 124b44bfc9cc "/docker-entrypoint.…" 22 minutes ago Up 22 minutes k8s\_nginx\_test-deployment-8687dc8bff-b57xd\_default\_6fdd7f48-db20-4b96-989d-3ff1623ce16c\_0

96fa262f792f 124b44bfc9cc "/docker-entrypoint.…" 32 minutes ago Up 32 minutes k8s\_event-pod\_event-pod\_default\_42c861c4-4fa5-4f37-8d6b-82a0f338bb7e\_0

C:\Windows\system32>docker logs b-1

PING google.com (142.250.192.142): 56 data bytes

64 bytes from 142.250.192.142: seq=0 ttl=63 time=118.092 ms

64 bytes from 142.250.192.142: seq=1 ttl=63 time=84.946 ms

64 bytes from 142.250.192.142: seq=2 ttl=63 time=62.591 ms

64 bytes from 142.250.192.142: seq=3 ttl=63 time=28.638 ms

64 bytes from 142.250.192.142: seq=4 ttl=63 time=252.012 ms

64 bytes from 142.250.192.142: seq=5 ttl=63 time=118.400 ms

64 bytes from 142.250.192.142: seq=6 ttl=63 time=79.522 ms

64 bytes from 142.250.192.142: seq=7 ttl=63 time=35.756 ms

64 bytes from 142.250.192.142: seq=8 ttl=63 time=59.298 ms

64 bytes from 142.250.192.142: seq=9 ttl=63 time=106.098 ms

64 bytes from 142.250.192.142: seq=10 ttl=63 time=287.212 ms

64 bytes from 142.250.192.142: seq=11 ttl=63 time=85.612 ms

64 bytes from 142.250.192.142: seq=12 ttl=63 time=43.883 ms

64 bytes from 142.250.192.142: seq=13 ttl=63 time=217.932 ms

64 bytes from 142.250.192.142: seq=14 ttl=63 time=144.665 ms

64 bytes from 142.250.192.142: seq=15 ttl=63 time=432.618 ms

64 bytes from 142.250.192.142: seq=16 ttl=63 time=87.856 ms

64 bytes from 142.250.192.142: seq=17 ttl=63 time=758.362 ms

64 bytes from 142.250.192.142: seq=18 ttl=63 time=469.344 ms

64 bytes from 142.250.192.142: seq=19 ttl=63 time=451.219 ms

64 bytes from 142.250.192.142: seq=20 ttl=63 time=116.380 ms

64 bytes from 142.250.192.142: seq=21 ttl=63 time=110.207 ms

64 bytes from 142.250.192.142: seq=22 ttl=63 time=38.872 ms

64 bytes from 142.250.192.142: seq=23 ttl=63 time=188.850 ms

64 bytes from 142.250.192.142: seq=24 ttl=63 time=31.828 ms

64 bytes from 142.250.192.142: seq=25 ttl=63 time=76.807 ms

64 bytes from 142.250.192.142: seq=26 ttl=63 time=31.249 ms

64 bytes from 142.250.192.142: seq=27 ttl=63 time=20.168 ms

64 bytes from 142.250.192.142: seq=28 ttl=63 time=145.604 ms

64 bytes from 142.250.192.142: seq=29 ttl=63 time=58.261 ms

64 bytes from 142.250.192.142: seq=30 ttl=63 time=151.351 ms

64 bytes from 142.250.192.142: seq=31 ttl=63 time=355.440 ms

64 bytes from 142.250.192.142: seq=32 ttl=63 time=190.561 ms

64 bytes from 142.250.192.142: seq=34 ttl=63 time=111.734 ms

64 bytes from 142.250.192.142: seq=35 ttl=63 time=105.104 ms

64 bytes from 142.250.192.142: seq=36 ttl=63 time=27.676 ms

64 bytes from 142.250.192.142: seq=37 ttl=63 time=52.213 ms

64 bytes from 142.250.192.142: seq=38 ttl=63 time=269.169 ms

64 bytes from 142.250.192.142: seq=39 ttl=63 time=23.685 ms

64 bytes from 142.250.192.142: seq=40 ttl=63 time=54.539 ms

64 bytes from 142.250.192.142: seq=41 ttl=63 time=35.542 ms

64 bytes from 142.250.192.142: seq=42 ttl=63 time=37.031 ms

64 bytes from 142.250.192.142: seq=43 ttl=63 time=56.752 ms

64 bytes from 142.250.192.142: seq=44 ttl=63 time=213.522 ms

64 bytes from 142.250.192.142: seq=45 ttl=63 time=154.438 ms

64 bytes from 142.250.192.142: seq=46 ttl=63 time=64.467 ms

64 bytes from 142.250.192.142: seq=47 ttl=63 time=137.249 ms

64 bytes from 142.250.192.142: seq=48 ttl=63 time=41.728 ms

64 bytes from 142.250.192.142: seq=49 ttl=63 time=68.035 ms

64 bytes from 142.250.192.142: seq=50 ttl=63 time=17.623 ms

64 bytes from 142.250.192.142: seq=51 ttl=63 time=114.105 ms

64 bytes from 142.250.192.142: seq=52 ttl=63 time=91.071 ms

64 bytes from 142.250.192.142: seq=53 ttl=63 time=48.994 ms

64 bytes from 142.250.192.142: seq=54 ttl=63 time=18.937 ms

64 bytes from 142.250.192.142: seq=55 ttl=63 time=82.049 ms

64 bytes from 142.250.192.142: seq=56 ttl=63 time=134.597 ms

64 bytes from 142.250.192.142: seq=57 ttl=63 time=98.799 ms

C:\Windows\system32>docker info

Client:

Version: 28.0.1

Context: desktop-linux

Debug Mode: false

Plugins:

ai: Docker AI Agent - Ask Gordon (Docker Inc.)

Version: v0.9.8

Path: C:\Users\user121\.docker\cli-plugins\docker-ai.exe

buildx: Docker Buildx (Docker Inc.)

Version: v0.21.1-desktop.2

Path: C:\Users\user121\.docker\cli-plugins\docker-buildx.exe

compose: Docker Compose (Docker Inc.)

Version: v2.33.1-desktop.1

Path: C:\Users\user121\.docker\cli-plugins\docker-compose.exe

debug: Get a shell into any image or container (Docker Inc.)

Version: 0.0.38

Path: C:\Users\user121\.docker\cli-plugins\docker-debug.exe

desktop: Docker Desktop commands (Beta) (Docker Inc.)

Version: v0.1.5

Path: C:\Users\user121\.docker\cli-plugins\docker-desktop.exe

dev: Docker Dev Environments (Docker Inc.)

Version: v0.1.2

Path: C:\Users\user121\.docker\cli-plugins\docker-dev.exe

extension: Manages Docker extensions (Docker Inc.)

Version: v0.2.27

Path: C:\Users\user121\.docker\cli-plugins\docker-extension.exe

feedback: Provide feedback, right in your terminal! (Docker Inc.)

Version: v1.0.5

Path: C:\Users\user121\.docker\cli-plugins\docker-feedback.exe

init: Creates Docker-related starter files for your project (Docker Inc.)

Version: v1.4.0

Path: C:\Users\user121\.docker\cli-plugins\docker-init.exe

sbom: View the packaged-based Software Bill Of Materials (SBOM) for an image (Anchore Inc.)

Version: 0.6.0

Path: C:\Users\user121\.docker\cli-plugins\docker-sbom.exe

scout: Docker Scout (Docker Inc.)

Version: v1.16.3

Path: C:\Users\user121\.docker\cli-plugins\docker-scout.exe

Server:

Containers: 47

Running: 45

Paused: 0

Stopped: 2

Images: 25

Server Version: 28.0.1

Storage Driver: overlayfs

driver-type: io.containerd.snapshotter.v1

Logging Driver: json-file

Cgroup Driver: cgroupfs

Cgroup Version: 1

Plugins:

Volume: local

Network: bridge host ipvlan macvlan null overlay

Log: awslogs fluentd gcplogs gelf journald json-file local splunk syslog

CDI spec directories:

/etc/cdi

/var/run/cdi

Swarm: inactive

Runtimes: nvidia runc io.containerd.runc.v2

Default Runtime: runc

Init Binary: docker-init

containerd version: bcc810d6b9066471b0b6fa75f557a15a1cbf31bb

runc version: v1.2.4-0-g6c52b3f

init version: de40ad0

Security Options:

seccomp

Profile: unconfined

Kernel Version: 5.15.167.4-microsoft-standard-WSL2

Operating System: Docker Desktop

OSType: linux

Architecture: x86\_64

CPUs: 4

Total Memory: 7.706GiB

Name: docker-desktop

ID: 2ce7b3fc-2d3d-41ee-bc67-45eb7f00f3b7

Docker Root Dir: /var/lib/docker

Debug Mode: false

HTTP Proxy: http.docker.internal:3128

HTTPS Proxy: http.docker.internal:3128

No Proxy: hubproxy.docker.internal

Labels:

com.docker.desktop.address=npipe://\\.\pipe\docker\_cli

Experimental: false

Insecure Registries:

hubproxy.docker.internal:5555

::1/128

127.0.0.0/8

Live Restore Enabled: false

WARNING: No blkio throttle.read\_bps\_device support

WARNING: No blkio throttle.write\_bps\_device support

WARNING: No blkio throttle.read\_iops\_device support

WARNING: No blkio throttle.write\_iops\_device support

WARNING: daemon is not using the default seccomp profile

C:\Windows\system32>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1acbb377edfe busybox "ping google.com" 3 minutes ago Up 3 minutes b-1

53c7cea44b4f 124b44bfc9cc "/docker-entrypoint.…" 25 minutes ago Up 24 minutes k8s\_nginx\_test-deployment-8687dc8bff-j24cc\_default\_2c1c72e1-b7f5-473c-a980-0e399852d509\_0

77fc7c44fc83 124b44bfc9cc "/docker-entrypoint.…" 25 minutes ago Up 25 minutes k8s\_nginx\_test-deployment-8687dc8bff-gcx5m\_default\_5d8a3bec-ac8c-4995-b146-23af701a5282\_0

2ab3b04567bf 124b44bfc9cc "/docker-entrypoint.…" 25 minutes ago Up 25 minutes k8s\_nginx\_test-deployment-8687dc8bff-b57xd\_default\_6fdd7f48-db20-4b96-989d-3ff1623ce16c\_0

96fa262f792f 124b44bfc9cc "/docker-entrypoint.…" 35 minutes ago Up 34 minutes k8s\_event-pod\_event-pod\_default\_42c861c4-4fa5-4f37-8d6b-82a0f338bb7e\_0

C:\Windows\system32>docker run --name my-custom --log-driver none busybox ping google.com

PING google.com (142.250.192.142): 56 data bytes

64 bytes from 142.250.192.142: seq=0 ttl=63 time=65.206 ms

64 bytes from 142.250.192.142: seq=2 ttl=63 time=88.671 ms

64 bytes from 142.250.192.142: seq=3 ttl=63 time=46.147 ms

64 bytes from 142.250.192.142: seq=4 ttl=63 time=103.139 ms

64 bytes from 142.250.192.142: seq=5 ttl=63 time=70.635 ms

64 bytes from 142.250.192.142: seq=6 ttl=63 time=151.395 ms

64 bytes from 142.250.192.142: seq=7 ttl=63 time=120.237 ms

64 bytes from 142.250.192.142: seq=8 ttl=63 time=57.285 ms

64 bytes from 142.250.192.142: seq=9 ttl=63 time=73.326 ms

64 bytes from 142.250.192.142: seq=10 ttl=63 time=30.945 ms

--- google.com ping statistics ---

11 packets transmitted, 10 packets received, 9% packet loss

round-trip min/avg/max = 30.945/80.698/151.395 ms

C:\Windows\system32>docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

1acbb377edfe busybox "ping google.com" 55 minutes ago Up 55 minutes b-1

53c7cea44b4f 124b44bfc9cc "/docker-entrypoint.…" About an hour ago Up About an hour k8s\_nginx\_test-deployment-8687dc8bff-j24cc\_default\_2c1c72e1-b7f5-473c-a980-0e399852d509\_0

77fc7c44fc83 124b44bfc9cc "/docker-entrypoint.…" About an hour ago Up About an hour k8s\_nginx\_test-deployment-8687dc8bff-gcx5m\_default\_5d8a3bec-ac8c-4995-b146-23af701a5282\_0

2ab3b04567bf 124b44bfc9cc "/docker-entrypoint.…" About an hour ago Up About an hour k8s\_nginx\_test-deployment-8687dc8bff-b57xd\_default\_6fdd7f48-db20-4b96-989d-3ff1623ce16c\_0

96fa262f792f 124b44bfc9cc "/docker-entrypoint.…" About an hour ago Up About an hour k8s\_event-pod\_event-pod\_default\_42c861c4-4fa5-4f37-8d6b-82a0f338bb7e\_0

C:\Windows\system32>docker logs my-custom

Error response from daemon: configured logging driver does not support reading

C:\Windows\system32>

apiVersion: v1

kind: Pod

metadata:

  name: pod01

  labels:

    name: myapp

spec:

  containers:

  - name: ping-container

    image: busybox

    command: [ "ping" ]

    args: ["google.com"]

D:\CKA BY zeal vora\PracticeME\section-8>kubectl apply -f pod-logging.yaml

pod/pod01 created

D:\CKA BY zeal vora\PracticeME\section-8>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 97m

pod01 1/1 Running 0 43s

test-deployment-8687dc8bff-b57xd 1/1 Running 0 87m

test-deployment-8687dc8bff-gcx5m 1/1 Running 0 87m

test-deployment-8687dc8bff-j24cc 1/1 Running 0 87m

D:\CKA BY zeal vora\PracticeME\section-8>kubectl logs pod01

PING google.com (142.250.192.142): 56 data bytes

D:\CKA BY zeal vora\PracticeME\section-8>ping google.com

Pinging google.com [142.250.192.142] with 32 bytes of data:

Reply from 142.250.192.142: bytes=32 time=89ms TTL=116

Request timed out.

D:\CKA BY zeal vora\PracticeME\section-8>kubectl get pods --namespace kube-system

NAME READY STATUS RESTARTS AGE

coredns-7c65d6cfc9-6cwcs 1/1 Running 3 (140m ago) 3d3h

coredns-7c65d6cfc9-mdg9r 1/1 Running 3 (140m ago) 3d3h

etcd-docker-desktop 1/1 Running 3 (140m ago) 3d3h

kube-apiserver-docker-desktop 1/1 Running 3 (140m ago) 3d3h

kube-controller-manager-docker-desktop 1/1 Running 3 (140m ago) 3d3h

kube-proxy-vvxm8 1/1 Running 3 (140m ago) 3d3h

kube-scheduler-docker-desktop 1/1 Running 3 (140m ago) 3d3h

metrics-server-54bf7cdd6-q5t4g 0/1 Running 0 73m

storage-provisioner 1/1 Running 6 (138m ago) 3d3h

vpnkit-controller 1/1 Running 3 (140m ago) 3d3h

D:\CKA BY zeal vora\PracticeME\section-8>kubectl logs metrics-server-54bf7cdd6-q5t4g --namespace kube-system

I0320 13:47:23.311020 1 serving.go:374] Generated self-signed cert (/tmp/apiserver.crt, /tmp/apiserver.key)

I0320 13:47:25.785956 1 handler.go:275] Adding GroupVersion metrics.k8s.io v1beta1 to ResourceManager

E0320 13:47:25.915357 1 scraper.go:149] "Failed to scrape node" err="Get \"https://192.168.65.3:10250/metrics/resource\": tls

apiVersion: v1

kind: Pod

metadata:

  name: pod02

  labels:

    name: myapp

spec:

  containers:

  - name: ping-container

    image: busybox

    command: [ "ping" ]

    args: ["google.com"]

  - name: ping-container-ip

    image: busybox

    command: [ "ping" ]

    args: ["8.8.8.8"]

D:\CKA BY zeal vora\PracticeME\section-8>kubectl apply -f multi-container-logging.yml

pod/pod02 created

D:\CKA BY zeal vora\PracticeME\section-8>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 104m

pod01 1/1 Running 0 7m2s

pod02 0/2 ContainerCreating 0 26s

test-deployment-8687dc8bff-b57xd 1/1 Running 0 94m

test-deployment-8687dc8bff-gcx5m 1/1 Running 0 94m

test-deployment-8687dc8bff-j24cc 1/1 Running 0 94m

D:\CKA BY zeal vora\PracticeME\section-8>kubectl get pods

NAME READY STATUS RESTARTS AGE

event-pod 1/1 Running 0 104m

pod01 1/1 Running 0 7m36s

pod02 2/2 Running 0 60s

test-deployment-8687dc8bff-b57xd 1/1 Running 0 94m

test-deployment-8687dc8bff-gcx5m 1/1 Running 0 94m

test-deployment-8687dc8bff-j24cc 1/1 Running 0 94m

D:\CKA BY zeal vora\PracticeME\section-8>kubectl logs pod02

Defaulted container "ping-container" out of: ping-container, ping-container-ip

PING google.com (142.250.192.142): 56 data bytes

D:\CKA BY zeal vora\PracticeME\section-8>kubectl logs pod02 ping-container-ip

PING 8.8.8.8 (8.8.8.8): 56 data bytes

64 bytes from 142.250.192.142: seq=8 ttl=63

D:\CKA BY zeal vora\PracticeME\section-8>kubectl logs pod02 ping-container

PING google.com (142.250.192.142): 56 data bytes

64 bytes from 142.250.192.142: seq=0 ttl=63 time=104.414 ms

64 bytes from 142.250.192.142: seq=1 ttl=63 time=102.609 ms

64 bytes from 142.250.192.142: seq=2 ttl=63 time=47.996 ms

D:\CKA BY zeal vora\PracticeME\section-8>nano /etc/systemd/system/kube-apiserver.service

'nano' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section-8>journatcl -u kube-apiserver

'journatcl' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section-8>journatcl -f -u kube-apiserver

'journatcl' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section-8>date

The current date is: 20-03-2025

Enter the new date: (dd-mm-yy)

D:\CKA BY zeal vora\PracticeME\section-8>journalctl --since "2019-09-2017" -u kube-apiserver

'journalctl' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\PracticeME\section-8>

# we can use since for specific time frame

**#using sicne we can see logs for specifc time frame**

#section – 9 – TROUBLESHOOTING APPLICATION FAILIURE

---

apiVersion: v1

kind: Service

metadata:

   name: kplabs-service

spec:

   selector:

     run: nginx

   type: NodePort

   ports:

   - port: 8089

     targetPort: 80

---

apiVersion: v1

kind: Namespace

metadata:

  name: teama

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-first

  labels:

      env: prod

  namespace: teama

spec:

  containers:

  - name: first-pod

    image: nginx

    ports:

        - containerPort: 80

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-second

  labels:

      env: pord

  namespace: teama

spec:

  containers:

  - name: second-pod

    image: nginx

    ports:

        - containerPort: 80

Microsoft Windows [Version 10.0.19045.5608]

(c) Microsoft Corporation. All rights reserved.

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl apply -f application-failure.yaml

service/kplabs-service created

namespace/teama created

pod/nginx-pod-first created

pod/nginx-pod-second created

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kplabs-service NodePort 10.108.142.188 <none> 8089:31166/TCP 55s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 18m

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get nodes

NAME STATUS ROLES AGE VERSION

docker-desktop Ready control-plane 19m v1.32.2

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get service

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kplabs-service NodePort 10.108.142.188 <none> 8089:31166/TCP 5m57s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 23m

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl describe service kplabs-service

Name: kplabs-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: run=nginx

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.108.142.188

IPs: 10.108.142.188

Port: <unset> 8089/TCP

TargetPort: 80/TCP

NodePort: <unset> 31166/TCP

Endpoints:

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get pods

No resources found in default namespace.

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get pods -n teama

NAME READY STATUS RESTARTS AGE

nginx-pod-first 1/1 Running 0 7m39s

nginx-pod-second 1/1 Running 0 7m39s

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl describe pods nginx-pod-first --namespace teama

Name: nginx-pod-first

Namespace: teama

Priority: 0

Service Account: default

Node: docker-desktop/192.168.65.3

Start Time: Fri, 21 Mar 2025 10:06:19 +0530

Labels: env=prod

Annotations: <none>

Status: Running

IP: 10.1.0.6

IPs:

IP: 10.1.0.6

Containers:

first-pod:

Container ID: docker://ea345538528559d2e8ff6d81e262793903a84a5535544ce2ffb89b81926ccf76

Image: nginx

Image ID: docker-pullable://nginx@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19

Port: 80/TCP

Host Port: 0/TCP

State: Running

Started: Fri, 21 Mar 2025 10:08:42 +0530

Ready: True

Restart Count: 0

Environment: <none>

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-fqs4x (ro)

Conditions:

Type Status

PodReadyToStartContainers True

Initialized True

Ready True

ContainersReady True

PodScheduled True

Volumes:

kube-api-access-fqs4x:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

ConfigMapOptional: <nil>

DownwardAPI: true

QoS Class: BestEffort

Node-Selectors: <none>

Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s

node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal Scheduled 9m17s default-scheduler Successfully assigned teama/nginx-pod-first to docker-desktop

Normal Pulling 9m15s kubelet Pulling image "nginx"

Normal Pulled 7m kubelet Successfully pulled image "nginx" in 2m3.955s (2m3.955s including waiting). Image size: 192004242 bytes.

Normal Created 6m54s kubelet Created container: first-pod

Normal Started 6m54s kubelet Started container first-pod

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>

---

apiVersion: v1

kind: Service

metadata:

   name: kplabs-service

   namespace: teama

spec:

   selector:

     run: nginx

   type: NodePort

   ports:

   - port: 8089

     targetPort: 80

---

apiVersion: v1

kind: Namespace

metadata:

  name: teama

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-first

  labels:

      env: prod

      run: nginx  #attaching labels with pod

  namespace: teama

spec:

  containers:

  - name: first-pod

    image: nginx

    ports:

        - containerPort: 80

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-second

  labels:

      env: pord

      run: nginx #attaching label with pod so that they match with service

  namespace: teama

spec:

  containers:

  - name: second-pod

    image: nginx

    ports:

        - containerPort: 80

Microsoft Windows [Version 10.0.19045.5608]

(c) Microsoft Corporation. All rights reserved.

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl apply -f application-failure-solution.yaml

namespace/teama created

pod/nginx-pod-first created

pod/nginx-pod-second created

Error from server (NotFound): error when creating "application-failure-solution.yaml": namespaces "teama" not found

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl delete -f application-failure-solution.yaml

namespace "teama" deleted

pod "nginx-pod-first" deleted

pod "nginx-pod-second" deleted

Error from server (NotFound): error when deleting "application-failure-solution.yaml": services "kplabs-service" not found

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl apply -f application-failure.yaml

service/kplabs-service created

namespace/teama created

pod/nginx-pod-first created

pod/nginx-pod-second created

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl apply -f application-failure-solution.yaml

service/kplabs-service created

namespace/teama unchanged

pod/nginx-pod-first configured

pod/nginx-pod-second configured

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl describe pod nginx-pod-first --namespace teama

Name: nginx-pod-first

Namespace: teama

Priority: 0

Service Account: default

Node: docker-desktop/192.168.65.3

Start Time: Fri, 21 Mar 2025 10:29:14 +0530

Labels: env=prod

run=nginx

Annotations: <none>

Status: Running

IP: 10.1.0.12

IPs:

IP: 10.1.0.12

Containers:

first-pod:

Container ID: docker://3d2a1fa052659b23e4ba68fb54d94e8888822de08e00de6ed19d41438abfdbe6

Image: nginx

Image ID: docker-pullable://nginx@sha256:124b44bfc9ccd1f3cedf4b592d4d1e8bddb78b51ec2ed5056c52d3692baebc19

Port: 80/TCP

Host Port: 0/TCP

State: Running

Started: Fri, 21 Mar 2025 10:29:19 +0530

Ready: True

Restart Count: 0

Environment: <none>

Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-z6n6d (ro)

Conditions:

Type Status

PodReadyToStartContainers True

Initialized True

Ready True

ContainersReady True

PodScheduled True

Volumes:

kube-api-access-z6n6d:

Type: Projected (a volume that contains injected data from multiple sources)

TokenExpirationSeconds: 3607

ConfigMapName: kube-root-ca.crt

ConfigMapOptional: <nil>

DownwardAPI: true

QoS Class: BestEffort

Node-Selectors: <none>

Tolerations: node.kubernetes.io/not-ready:NoExecute op=Exists for 300s

node.kubernetes.io/unreachable:NoExecute op=Exists for 300s

Events:

Type Reason Age From Message

---- ------ ---- ---- -------

Normal Scheduled 90s default-scheduler Successfully assigned teama/nginx-pod-first to docker-desktop

Normal Pulling 89s kubelet Pulling image "nginx"

Normal Pulled 86s kubelet Successfully pulled image "nginx" in 3.109s (3.109s including waiting). Image size: 192004242 bytes.

Normal Created 85s kubelet Created container: first-pod

Normal Started 85s kubelet Started container first-pod

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kplabs-service NodePort 10.102.254.160 <none> 8089:31908/TCP 2m45s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 42m

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectld describe service kplabs-service

'kubectld' is not recognized as an internal or external command,

operable program or batch file.

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl describe service kplabs-service

Name: kplabs-service

Namespace: default

Labels: <none>

Annotations: <none>

Selector: run=nginx

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.102.254.160

IPs: 10.102.254.160

Port: <unset> 8089/TCP

TargetPort: 80/TCP

NodePort: <unset> 31908/TCP

Endpoints:

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>

---

apiVersion: v1

kind: Service

metadata:

   name: kplabs-service

   namespace: teama

spec:

   selector:

     run: nginx

     env: prod

   type: NodePort

   ports:

   - port: 8089

     targetPort: 80

---

apiVersion: v1

kind: Namespace

metadata:

  name: teama

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-first

  labels:

      env: prod

  namespace: teama

spec:

  containers:

  - name: first-pod

    image: nginx

    ports:

        - containerPort: 80

---

apiVersion: v1

kind: Pod

metadata:

  name: nginx-pod-second

  labels:

      env: pord

  namespace: teama

spec:

  containers:

  - name: second-pod

    image: nginx

    ports:

        - containerPort: 80

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl apply -f application-failure.yaml

service/kplabs-service unchanged

namespace/teama unchanged

pod/nginx-pod-first configured

pod/nginx-pod-second configured

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get svc

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kplabs-service NodePort 10.102.254.160 <none> 8089:31908/TCP 7m8s

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 47m

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl get svc --namespace teama

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kplabs-service NodePort 10.106.29.17 <none> 8089:31598/TCP 7m4s

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl describe service kplabs-service --namespace teama

Name: kplabs-service

Namespace: teama

Labels: <none>

Annotations: <none>

Selector: run=nginx

Type: NodePort

IP Family Policy: SingleStack

IP Families: IPv4

IP: 10.106.29.17

IPs: 10.106.29.17

Port: <unset> 8089/TCP

TargetPort: 80/TCP

NodePort: <unset> 31598/TCP

Endpoints:

Session Affinity: None

External Traffic Policy: Cluster

Internal Traffic Policy: Cluster

Events: <none>

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl cluster info

error: unknown command "cluster" for "kubectl"

Did you mean this?

cluster-info

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl cluster-info

Kubernetes control plane is running at https://kubernetes.docker.internal:6443

CoreDNS is running at https://kubernetes.docker.internal:6443/api/v1/namespaces/kube-system/services/kube-dns:dns/proxy

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl cluster-info dump

{

"kind": "NodeList",

D:\CKA BY zeal vora\certified-kubernetes-administrator-master\Domain 8 - Troubleshooting>kubectl version

Client Version: v1.32.2

Kustomize Version: v5.5.0

Server Version: v1.32.2