## Setting up different tools:

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl cluster-info --context $(kubectl config current-context)

E1001 12:05:42.376179 90899 memcache.go:265] couldn't get current server API group list: Get "https://127.0.0.1:34575/api?timeout=32s": dial tcp 127.0.0.1:34575: connect: connection refused

E1001 12:05:42.376567 90899 memcache.go:265] couldn't get current server API group list: Get "https://127.0.0.1:34575/api?timeout=32s": dial tcp 127.0.0.1:34575: connect: connection refused

E1001 12:05:42.378230 90899 memcache.go:265] couldn't get current server API group list: Get "https://127.0.0.1:34575/api?timeout=32s": dial tcp 127.0.0.1:34575: connect: connection refused

E1001 12:05:42.378652 90899 memcache.go:265] couldn't get current server API group list: Get "https://127.0.0.1:34575/api?timeout=32s": dial tcp 127.0.0.1:34575: connect: connection refused

E1001 12:05:42.380154 90899 memcache.go:265] couldn't get current server API group list: Get "https://127.0.0.1:34575/api?timeout=32s": dial tcp 127.0.0.1:34575: connect: connection refused

To further debug and diagnose cluster problems, use 'kubectl cluster-info dump'.

The connection to the server 127.0.0.1:34575 was refused - did you specify the right host or port?

ibtisam@mint-dell:~/k8s/10-1$ kind get clusters

No kind clusters found.

ibtisam@mint-dell:~/k8s/10-1$

The error message you're encountering indicates that kubectl is trying to connect to the Kubernetes API server at 127.0.0.1:34575, but the connection is being refused. This typically means that the Kubernetes cluster (likely a local one using Kind in your case) is either not running, or the API server is not accessible.

## Steps to troubleshoot and resolve this issue:

kind get clusters

kind create cluster --name kind-ibtisam

docker ps

kubectl config use-context kind-ibtisam

Unfortunately, Kind doesn't have a start command for restarting an existing cluster. If you need to restart (recreate) a cluster that was previously created but isn't running, you should first delete the existing cluster, and then you would need to use the kind create cluster command.

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kind get clusters

No kind clusters found.

ibtisam@mint-dell:~/k8s/10-1$ kind get nodes

No kind nodes found for cluster "kind".

ibtisam@mint-dell:~/k8s/10-1$ kubectl config delete-context kind-ibtisam

warning: this removed your active context, use "kubectl config use-context" to select a different one

deleted context kind-ibtisam from /home/ibtisam/.kube/config

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-clusters

# NAME

kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl config delete-cluster kind-ibtisam

deleted cluster kind-ibtisam from /home/ibtisam/.kube/config

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-clusters

# NAME

ibtisam@mint-dell:~/k8s/10-1$

Cluster: The actual Kubernetes environment where your resources are deployed. Clusters tell kubectl where to send your commands (which environment you're managing).

Context: A combination of cluster, user, and namespace that defines the current environment for kubectl commands. Contexts make it easier to switch between different environments (clusters and users).

ibtisam@mint-dell:~/k8s/10-1$ ls

01.kind-1.yaml

ibtisam@mint-dell:~/k8s/10-1$ kind create cluster --config 01.kind-1.yaml

Creating cluster "ibtisam" ...

✓ Ensuring node image (kindest/node:v1.30.0) 🖼

✓ Preparing nodes 📦 📦 📦 📦

✓ Configuring the external load balancer ⚖️

✓ Writing configuration 📜

✓ Starting control-plane 🕹️

✓ Installing CNI 🔌

✓ Installing StorageClass 💾

✓ Joining more control-plane nodes 🎮

✓ Joining worker nodes 🚜

Set kubectl context to "kind-ibtisam"

## You can now use your cluster with:

kubectl cluster-info --context kind-ibtisam

Have a nice day! 👋

ibtisam@mint-dell:~/k8s/10-1$ pwd

/home/ibtisam/k8s/10-1

ibtisam@mint-dell:~/k8s/10-1$ cat /home/ibtisam/k8s/nectar.yaml

# kind-cluster-config.yaml

apiVersion: kind.x-k8s.io/v1alpha4 # Specifies the Kind API version to use

kind: Cluster # Declares that this is a Kind Cluster configuration

name: ibtisam

nodes: # Declares the nodes section to define multiple nodes in the cluster

- role: control-plane # Defines the first node as a control-plane (master) node

image: kindest/node:v1.30.0 # Specifies the container image for the control-plane node (Kubernetes version v1.24.0)

- role: control-plane # Defines the second node as a control-plane (master) node

image: kindest/node:v1.30.0 # Uses the same Kubernetes version for uniformity

- role: worker # Defines the first worker node

image: kindest/node:v1.30.0 # Specifies the Kubernetes version for the worker node

- role: worker # Defines the second worker node

image: kindest/node:v1.30.0 # Uses the same Kubernetes version for the worker node

ibtisam@mint-dell:~/k8s/10-1$ kubectl get nodes -o wide

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

ibtisam-control-plane Ready control-plane 4m53s v1.30.0 172.18.0.5 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-control-plane2 Ready control-plane 4m16s v1.30.0 172.18.0.6 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-worker Ready <none> 2m49s v1.30.0 172.18.0.4 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-worker2 Ready <none> 2m45s v1.30.0 172.18.0.3 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-clusters

# NAME

kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl config view

apiVersion: v1

## clusters:

## - cluster:

certificate-authority-data: DATA+OMITTED

server: https://127.0.0.1:34821

name: kind-ibtisam

## contexts:

## - context:

cluster: kind-ibtisam

user: kind-ibtisam

name: kind-ibtisam

current-context: kind-ibtisam

kind: Config

preferences: {}

## users:

* name: kind-ibtisam

## user:

client-certificate-data: DATA+OMITTED

client-key-data: DATA+OMITTED

ibtisam@mint-dell:~/k8s/10-1$ kind delete cluster --name ibtisam

Deleting cluster "ibtisam" ...

Deleted nodes: ["ibtisam-control-plane2" "ibtisam-control-plane" "ibtisam-worker2" "ibtisam-worker" "ibtisam-external-load-balancer"]

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-clusters

# NAME

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

ibtisam@mint-dell:~/k8s/10-1$

setting up pod

ibtisam@mint-dell:~/k8s/10-1$ ls

01.kind-1.yaml

ibtisam@mint-dell:~/k8s/10-1$ kind create cluster --config 01.kind-1.yaml

kubectl cluster-info --context kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kind get clusters

ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl get nodes -o wide

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

ibtisam-control-plane Ready control-plane 4m48s v1.30.0 172.18.0.6 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-control-plane2 Ready control-plane 4m9s v1.30.0 172.18.0.3 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-worker NotReady <none> 97s v1.30.0 172.18.0.5 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam-worker2 NotReady <none> 97s v1.30.0 172.18.0.4 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 02.pod.yaml

pod/abc created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

abc 0/1 ContainerCreating 0 19s <none> ibtisam-worker2 <none> <none>

No. of containers

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide -w

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

abc 0/1 ContainerCreating 0 31s <none> ibtisam-worker2 <none> <none>

abc 1/1 Running 0 2m21s 10.244.2.2 ibtisam-worker2 <none> <none>

^Cibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

abc 1/1 Running 0 2m48s 10.244.2.2 ibtisam-worker2 <none> <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po abc

Name: abc

Namespace: default

Priority: 0

Service Account: default

Node: ibtisam-worker2/172.18.0.4

Start Time: Wed, 02 Oct 2024 11:14:41 +0500

Labels: run=abc

Annotations: <none>

Status: Running

# IP: 10.244.2.2

## IPs:

# IP: 10.244.2.2

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs abc

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

/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/

/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh

10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf

10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf

/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh

/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh

/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh

/docker-entrypoint.sh: Configuration complete; ready for start up

2024/10/02 06:16:59 [notice] 1#1: using the "epoll" event method

2024/10/02 06:16:59 [notice] 1#1: nginx/1.27.1

2024/10/02 06:16:59 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)

2024/10/02 06:16:59 [notice] 1#1: OS: Linux 6.6.32-linuxkit

2024/10/02 06:16:59 [notice] 1#1: getrlimit(RLIMIT\_NOFILE): 1048576:1048576

2024/10/02 06:16:59 [notice] 1#1: start worker processes

2024/10/02 06:16:59 [notice] 1#1: start worker process 33

2024/10/02 06:16:59 [notice] 1#1: start worker process 34

2024/10/02 06:16:59 [notice] 1#1: start worker process 35

2024/10/02 06:16:59 [notice] 1#1: start worker process 36

ibtisam@mint-dell:~/k8s/10-1$ kubectl delete po abc

pod "abc" deleted

ibtisam@mint-dell:~/k8s/10-1$

init container

ibtisam@mint-dell:~/k8s/10-1$ ls

01.kind-1.yaml 02.sin-con.yaml 03.side-con.yaml 04.amb-con.yaml 05.init-con.yaml 06.po.yaml

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 05.init-con.yaml

pod/init created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide -w

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

init 0/1 Init:0/1 0 22s 10.244.3.3 ibtisam-worker <none> <none>

init 0/1 PodInitializing 0 66s 10.244.3.3 ibtisam-worker <none> <none>

init 1/1 Running 0 69s 10.244.3.3 ibtisam-worker <none> <none>

^Cibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

init 1/1 Running 0 116s 10.244.3.3 ibtisam-worker <none> <none>

ibtisam@mint-dell:~/k8s/10-1$

#rc,rs,deploy

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rc,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-55cd5746b5-jhwvb 1/1 Running 0 4m21s 10.244.3.6 ibtisam-worker <none> <none>

pod/dp7xyz-55cd5746b5-lmkv7 1/1 Running 0 4m22s 10.244.2.4 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-pj88s 1/1 Running 0 4m21s 10.244.2.3 ibtisam-worker2 <none> <none>

pod/rc7abc-8gk9l 1/1 Running 0 4m21s 10.244.3.8 ibtisam-worker <none> <none>

pod/rc7abc-kfnqc 1/1 Running 0 4m21s 10.244.3.4 ibtisam-worker <none> <none>

pod/rc7abc-qzl5h 1/1 Running 0 4m22s 10.244.2.6 ibtisam-worker2 <none> <none>

pod/rs7def-gf5d2 1/1 Running 0 4m22s 10.244.3.7 ibtisam-worker <none> <none>

pod/rs7def-phc56 1/1 Running 0 4m22s 10.244.3.5 ibtisam-worker <none> <none>

pod/rs7def-sxt6f 1/1 Running 0 4m22s 10.244.2.5 ibtisam-worker2 <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicationcontroller/rc7abc 3 3 3 4m25s rc7nginx nginx app=frontend

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55cd5746b5 3 3 3 4m23s dp7nginx nginx app=database,pod-template-hash=55cd5746b5

replicaset.apps/rs7def 3 3 3 4m25s rs7nginx nginx app=backend

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 3/3 3 3 4m24s dp7nginx nginx app=database

Number of pods or containers?

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe rc rc7abc

Name: rc7abc

Namespace: default

Selector: app=frontend

Labels: sex=m

Annotations: <none>

Replicas: 3 current / 3 desired

Pods Status: 3 Running / 0 Waiting / 0 Succeeded / 0 Failed

## Pod Template:

Labels: app=frontend

## Containers:

## rc7nginx:

Image: nginx

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Node-Selectors: <none>

Tolerations: <none>

## Events:

Type Reason Age From Message

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

Normal SuccessfulCreate 26m replication-controller Created pod: rc7abc-qzl5h

Normal SuccessfulCreate 26m replication-controller Created pod: rc7abc-kfnqc

Normal SuccessfulCreate 26m replication-controller Created pod: rc7abc-8gk9l

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe rs dp7xyz-55cd5746b5

Name: dp7xyz-55cd5746b5

Namespace: default

Selector: app=database,pod-template-hash=55cd5746b5

Labels: app=database

pod-template-hash=55cd5746b5

Annotations: deployment.kubernetes.io/desired-replicas: 3

deployment.kubernetes.io/max-replicas: 4

deployment.kubernetes.io/revision: 1

Controlled By: Deployment/dp7xyz

Replicas: 3 current / 3 desired

Pods Status: 3 Running / 0 Waiting / 0 Succeeded / 0 Failed

## Pod Template:

Labels: app=database

pod-template-hash=55cd5746b5

## Containers:

## dp7nginx:

Image: nginx

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Node-Selectors: <none>

Tolerations: <none>

## Events:

Type Reason Age From Message

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

Normal SuccessfulCreate 66m replicaset-controller Created pod: dp7xyz-55cd5746b5-lmkv7

Normal SuccessfulCreate 66m replicaset-controller Created pod: dp7xyz-55cd5746b5-jhwvb

Normal SuccessfulCreate 66m replicaset-controller Created pod: dp7xyz-55cd5746b5-pj88s

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe deploy dp7xyz | grep -i strategy

StrategyType: RollingUpdate Recreate

RollingUpdateStrategy: 25% max unavailable, 25% max surge

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl scale deploy dp7xyz --replicas 6

deployment.apps/dp7xyz scaled

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rc,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-55cd5746b5-dbkqh 1/1 Running 0 2m14s 10.244.2.7 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-dt65f 1/1 Running 0 2m14s 10.244.3.10 ibtisam-worker <none> <none>

pod/dp7xyz-55cd5746b5-jhwvb 1/1 Running 0 74m 10.244.3.6 ibtisam-worker <none> <none>

pod/dp7xyz-55cd5746b5-lmkv7 1/1 Running 0 74m 10.244.2.4 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-pj88s 1/1 Running 0 74m 10.244.2.3 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-v86hp 1/1 Running 0 2m14s 10.244.3.9 ibtisam-worker <none> <none>

pod/rc7abc-8gk9l 1/1 Running 0 74m 10.244.3.8 ibtisam-worker <none> <none>

pod/rc7abc-kfnqc 1/1 Running 0 74m 10.244.3.4 ibtisam-worker <none> <none>

pod/rc7abc-qzl5h 1/1 Running 0 74m 10.244.2.6 ibtisam-worker2 <none> <none>

pod/rs7def-gf5d2 1/1 Running 0 74m 10.244.3.7 ibtisam-worker <none> <none>

pod/rs7def-phc56 1/1 Running 0 74m 10.244.3.5 ibtisam-worker <none> <none>

pod/rs7def-sxt6f 1/1 Running 0 74m 10.244.2.5 ibtisam-worker2 <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicationcontroller/rc7abc 3 3 3 74m rc7nginx nginx app=frontend

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55cd5746b5 6 6 6 74m dp7nginx nginx app=database,pod-template-hash=55cd5746b5

replicaset.apps/rs7def 3 3 3 74m rs7nginx nginx app=backend

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 6/6 6 6 74m dp7nginx nginx app=database

ibtisam@mint-dell:~/k8s/10-1$ kubectl scale deploy dp7xyz --replicas 2

deployment.apps/dp7xyz scaled

ibtisam@mint-dell:~/k8s/10-1$ kubectl edit deploy dp7xyz

deployment.apps/dp7xyz edited

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rc,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-55cd5746b5-cj8pj 1/1 Running 0 56s 10.244.3.12 ibtisam-worker <none> <none>

pod/dp7xyz-55cd5746b5-lmkv7 1/1 Running 0 84m 10.244.2.4 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-p6qb8 1/1 Running 0 55s 10.244.3.11 ibtisam-worker <none> <none>

pod/dp7xyz-55cd5746b5-pj88s 1/1 Running 0 84m 10.244.2.3 ibtisam-worker2 <none> <none>

pod/dp7xyz-55cd5746b5-vgj6w 1/1 Running 0 55s 10.244.2.8 ibtisam-worker2 <none> <none>

pod/rc7abc-8gk9l 1/1 Running 0 84m 10.244.3.8 ibtisam-worker <none> <none>

pod/rc7abc-kfnqc 1/1 Running 0 84m 10.244.3.4 ibtisam-worker <none> <none>

pod/rc7abc-qzl5h 1/1 Running 0 84m 10.244.2.6 ibtisam-worker2 <none> <none>

pod/rs7def-gf5d2 1/1 Running 0 84m 10.244.3.7 ibtisam-worker <none> <none>

pod/rs7def-phc56 1/1 Running 0 84m 10.244.3.5 ibtisam-worker <none> <none>

pod/rs7def-sxt6f 1/1 Running 0 84m 10.244.2.5 ibtisam-worker2 <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicationcontroller/rc7abc 3 3 3 84m rc7nginx nginx app=frontend

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55cd5746b5 5 5 5 84m dp7nginx nginx app=database,pod-template-hash=55cd5746b5

replicaset.apps/rs7def 3 3 3 84m rs7nginx nginx app=backend

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 5/5 5 5 84m dp7nginx nginx app=database

ibtisam@mint-dell:~/k8s/10-1$

#rollout

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy

No resources found in default namespace.

ibtisam@mint-dell:~/k8s/10-1$ kubectl create deploy dp7xyz --image nginx --replicas 3 --port 80 --dry-run=client -o yaml | kubectl apply --record -f-

Flag --record has been deprecated, --record will be removed in the future

deployment.apps/dp7xyz created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy -o wide

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

dp7xyz 3/3 3 3 34s nginx nginx app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$ kubectl edit deploy dp7xyz --record

Flag --record has been deprecated, --record will be removed in the future

deployment.apps/dp7xyz edited

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout status deploy dp7xyz

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 old replicas are pending termination...

Waiting for deployment "dp7xyz" rollout to finish: 1 old replicas are pending termination...

deployment "dp7xyz" successfully rolled out

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy -o wide

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

dp7xyz 3/3 3 3 8m13s nginx nginx:1.26 app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$ kubectl set image deploy dp7xyz nginx=nginx:1.26.2 --record

Flag --record has been deprecated, --record will be removed in the future

deployment.apps/dp7xyz image updated

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout status deploy dp7xyz

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 2 out of 3 new replicas have been updated...

Waiting for deployment "dp7xyz" rollout to finish: 1 old replicas are pending termination...

Waiting for deployment "dp7xyz" rollout to finish: 1 old replicas are pending termination...

deployment "dp7xyz" successfully rolled out

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout history deploy dp7xyz

deployment.apps/dp7xyz

# REVISION CHANGE-CAUSE

1 kubectl apply --record=true --filename=-

2 kubectl edit deploy dp7xyz --record=true

3 kubectl set image deploy dp7xyz nginx=nginx:1.26.2 --record=true

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy -o wide

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

dp7xyz 3/3 3 3 19m nginx nginx:1.26.2 app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-7fd9556898-bhg86 1/1 Running 0 11m 10.244.3.18 ibtisam-worker <none> <none>

pod/dp7xyz-7fd9556898-grsm9 1/1 Running 0 11m 10.244.3.19 ibtisam-worker <none> <none>

pod/dp7xyz-7fd9556898-w5z4h 1/1 Running 0 11m 10.244.2.16 ibtisam-worker2 <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55b54fc4b 0 0 0 19m nginx nginx:1.26 app=dp7xyz,pod-template-hash=55b54fc4b

replicaset.apps/dp7xyz-665d8bb6c9 0 0 0 23m nginx nginx app=dp7xyz,pod-template-hash=665d8bb6c9

replicaset.apps/dp7xyz-7fd9556898 3 3 3 11m nginx nginx:1.26.2 app=dp7xyz,pod-template-hash=7fd9556898

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 3/3 3 3 23m nginx nginx:1.26.2 app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout history deploy dp7xyz --revision 2

deployment.apps/dp7xyz with revision #2

## Pod Template:

Labels: app=dp7xyz

pod-template-hash=55b54fc4b

Annotations: kubernetes.io/change-cause: kubectl edit deploy dp7xyz --record=true

## Containers:

## nginx:

Image: nginx:1.26

Port: 80/TCP

Host Port: 0/TCP

Environment: <none>

Mounts: <none>

Volumes: <none>

Node-Selectors: <none>

Tolerations: <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout undo deploy dp7xyz

deployment.apps/dp7xyz rolled back

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-55b54fc4b-mczhp 0/1 ContainerCreating 0 14s <none> ibtisam-worker2 <none> <none>

pod/dp7xyz-7fd9556898-bhg86 1/1 Running 0 38m 10.244.3.18 ibtisam-worker <none> <none>

pod/dp7xyz-7fd9556898-grsm9 1/1 Running 0 38m 10.244.3.19 ibtisam-worker <none> <none>

pod/dp7xyz-7fd9556898-w5z4h 1/1 Running 0 38m 10.244.2.16 ibtisam-worker2 <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55b54fc4b 1 1 0 46m nginx nginx:1.26 app=dp7xyz,pod-template-hash=55b54fc4b

replicaset.apps/dp7xyz-665d8bb6c9 0 0 0 50m nginx nginx app=dp7xyz,pod-template-hash=665d8bb6c9

replicaset.apps/dp7xyz-7fd9556898 3 3 3 38m nginx nginx:1.26.2 app=dp7xyz,pod-template-hash=7fd9556898

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 3/3 1 3 50m nginx nginx:1.26 app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout history deploy dp7xyz

deployment.apps/dp7xyz

# REVISION CHANGE-CAUSE

1 kubectl apply --record=true --filename=-

3 kubectl set image deploy dp7xyz nginx=nginx:1.26.2 --record=true

4 kubectl edit deploy dp7xyz --record=true

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout undo deploy dp7xyz --to-revision 1

deployment.apps/dp7xyz rolled back

ibtisam@mint-dell:~/k8s/10-1$ kubectl rollout history deploy dp7xyz

deployment.apps/dp7xyz

# REVISION CHANGE-CAUSE

3 kubectl set image deploy dp7xyz nginx=nginx:1.26.2 --record=true

4 kubectl edit deploy dp7xyz --record=true

5 kubectl apply --record=true --filename=-

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,rs,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7xyz-55b54fc4b-mczhp 1/1 Running 0 2m16s 10.244.2.17 ibtisam-worker2 <none> <none>

pod/dp7xyz-665d8bb6c9-j4468 1/1 Running 0 13s 10.244.2.19 ibtisam-worker2 <none> <none>

pod/dp7xyz-665d8bb6c9-ns4s6 1/1 Running 0 22s 10.244.3.21 ibtisam-worker <none> <none>

pod/dp7xyz-665d8bb6c9-prn4r 0/1 ContainerCreating 0 7s <none> ibtisam-worker <none> <none>

# NAME DESIRED CURRENT READY AGE CONTAINERS IMAGES SELECTOR

replicaset.apps/dp7xyz-55b54fc4b 1 1 1 48m nginx nginx:1.26 app=dp7xyz,pod-template-hash=55b54fc4b

replicaset.apps/dp7xyz-665d8bb6c9 3 3 2 52m nginx nginx app=dp7xyz,pod-template-hash=665d8bb6c9

replicaset.apps/dp7xyz-7fd9556898 0 0 0 40m nginx nginx:1.26.2 app=dp7xyz,pod-template-hash=7fd9556898

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7xyz 3/3 3 3 52m nginx nginx app=dp7xyz

ibtisam@mint-dell:~/k8s/10-1$

#blue-green deployment (port forwarding is optional)

ibtisam@mint-dell:~/k8s/10-1$ ls

01.kind-1.yaml 02.sin-con.yaml 03.side-con.yaml 04.amb-con.yaml 05.init-con.yaml 06.po.yaml 071.b-g-dep.yaml 072.can.dep.yaml 07.rc-rs-dep.yaml

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 071.b-g-dep.yaml

deployment.apps/dp71xyz created

service/b-g-svc created

deployment.apps/dp711xyz created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy,po,svc -o wide

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp711xyz 3/3 3 3 5m34s dp711nginx linuxacademycontent/ckad-nginx:green app=frontend,color=blue

deployment.apps/dp71xyz 3/3 3 3 5m38s dp71nginx linuxacademycontent/ckad-nginx:blue app=frontend,color=blue

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp711xyz-6645cc5974-hz7n2 1/1 Running 0 5m31s 10.244.2.4 ibtisam-worker <none> <none>

pod/dp711xyz-6645cc5974-jf7bj 1/1 Running 0 5m31s 10.244.3.3 ibtisam-worker2 <none> <none>

pod/dp711xyz-6645cc5974-xf7sb 1/1 Running 0 5m31s 10.244.2.3 ibtisam-worker <none> <none>

pod/dp71xyz-554984dbb4-czchg 1/1 Running 0 5m31s 10.244.2.5 ibtisam-worker <none> <none>

pod/dp71xyz-554984dbb4-jxxn5 1/1 Running 0 5m31s 10.244.3.4 ibtisam-worker2 <none> <none>

pod/dp71xyz-554984dbb4-qcr7t 1/1 Running 0 5m31s 10.244.3.2 ibtisam-worker2 <none> <none>

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

service/b-g-svc ClusterIP 10.96.210.98 <none> 80/TCP 5m37s app=frontend,color=blue

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 120m <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/b-g-svc 7070:80

Forwarding from 127.0.0.1:7070 -> 80

Forwarding from [::1]:7070 -> 80

Handling connection for 7070

^Cibtisam@mint-dell:~/k8s/10-1$ kubectl edit svc b-g-svc

service/b-g-svc edited

ibtisam@mint-dell:~/k8s/10-1$ kubectl get svc -o wide

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

service/b-g-svc ClusterIP 10.96.210.98 <none> 80/TCP 17m app=frontend,color=green

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 131m <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/b-g-svc 7060:80

error: timed out waiting for the condition

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/b-g-svc 8000:80

error: timed out waiting for the condition

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl delete deploy dp71xyz dp711xyz

deployment.apps "dp71xyz" deleted

deployment.apps "dp711xyz" deleted

ibtisam@mint-dell:~/k8s/10-1$ kubectl delete svc b-g-svc

service "b-g-svc" deleted

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 071.b-g-dep.yaml

deployment.apps/dp7blue created

service/b-g-svc created

deployment.apps/dp7green created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,deploy,svc -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7blue-5d5547458-bhs6l 1/1 Running 0 5m10s 10.244.2.9 ibtisam-worker <none> <none>

pod/dp7blue-5d5547458-nznj8 1/1 Running 0 5m10s 10.244.2.10 ibtisam-worker <none> <none>

pod/dp7blue-5d5547458-rl2ts 1/1 Running 0 5m11s 10.244.3.9 ibtisam-worker2 <none> <none>

pod/dp7green-6d4dc468cc-7j97l 1/1 Running 0 5m10s 10.244.3.8 ibtisam-worker2 <none> <none>

pod/dp7green-6d4dc468cc-8z2ks 1/1 Running 0 5m11s 10.244.2.11 ibtisam-worker <none> <none>

pod/dp7green-6d4dc468cc-hnqhj 1/1 Running 0 5m10s 10.244.3.10 ibtisam-worker2 <none> <none>

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7blue 3/3 3 3 5m16s dp7bl-nginx linuxacademycontent/ckad-nginx:blue app=frontend,color=blue

deployment.apps/dp7green 3/3 3 3 5m14s dp7gr-nginx linuxacademycontent/ckad-nginx:green app=frontend,color=green

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

service/b-g-svc ClusterIP 10.96.188.206 <none> 3741/TCP 5m15s app=frontend,color=blue

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 9h <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe svc b-g-svc

Name: b-g-svc

Namespace: default

Labels: <none>

Annotations: <none>

Selector: app=frontend,color=blue

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

# IP: 10.96.188.206

IPs: 10.96.188.206

Port: <unset> 3741/TCP

TargetPort: 80/TCP

Endpoints: 10.244.2.10:80,10.244.2.9:80,10.244.3.9:80

Session Affinity: None

Events: <none>

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ docker exec -it ibtisam-worker bash

root@ibtisam-worker:/# curl 10.96.188.206:3741

I'm Blue!

root@ibtisam-worker:/# curl 10.244.2.9 # pod IP, accessible even the service is not exposed yet.

I'm Blue!

root@ibtisam-worker:/# curl 10.244.3.8 # accessing even its node is not executed. # this is how green env is tested before deploy

I'm green!

root@ibtisam-worker:/# exit

exit

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward service/b-g-svc 7070:3741 > /dev/null 2>&1 & extra

[1] 1029983

ibtisam@mint-dell:~/k8s/10-1$ kubectl edit svc b-g-svc

service/b-g-svc edited

ibtisam@mint-dell:~/k8s/10-1$ kubectl get svc -o wide

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

b-g-svc ClusterIP 10.96.188.206 <none> 3741/TCP 3h35m app=frontend,color=green

kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 8h <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward service/b-g-svc 7080:3741 > /dev/null 2>&1 & extra

[2] 1031007

ibtisam@mint-dell:~/k8s/10-1$

setting up canary deployment (no port forwarding)

ibtisam@mint-dell:~/k8s/10-1$ ls

01.kind-1.yaml 02.sin-con.yaml 03.side-con.yaml 04.amb-con.yaml 05.init-con.yaml 06.po.yaml 071.b-g-dep.yaml 072.can.dep.yaml 07.rc-rs-dep.yaml 08.svc.yaml

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 072.can.dep.yaml

deployment.apps/dp7v1 created

service/can-svc created

The Deployment "dp7can" is invalid: spec.template.metadata.labels: Invalid value: map[string]string{"env":"canary", "tier":"frontend"}: `selector` does not match template `labels`

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 072.can.dep.yaml

deployment.apps/dp7v1 unchanged

service/can-svc unchanged

deployment.apps/dp7can created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,svc,deploy -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dp7can-5d6646fbc5-tt4q8 1/1 Running 0 8m3s 10.244.3.12 ibtisam-worker2 <none> <none>

pod/dp7v1-5ff55c58b9-9qhzg 1/1 Running 0 14m 10.244.3.11 ibtisam-worker2 <none> <none>

pod/dp7v1-5ff55c58b9-btqhp 1/1 Running 0 14m 10.244.2.13 ibtisam-worker <none> <none>

pod/dp7v1-5ff55c58b9-vw7fn 1/1 Running 0 14m 10.244.2.12 ibtisam-worker <none> <none>

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

service/can-svc ClusterIP 10.96.3.3 <none> 4741/TCP 14m tier=frontend

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 11h <none>

# NAME READY UP-TO-DATE AVAILABLE AGE CONTAINERS IMAGES SELECTOR

deployment.apps/dp7can 1/1 1 1 8m4s dp7can-nginx linuxacademycontent/ckad-nginx:canary env=canary,tier=frontend

deployment.apps/dp7v1 3/3 3 3 14m dp7v1nginx linuxacademycontent/ckad-nginx:1.0.0 env=main,tier=frontend

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe svc can-svc

Name: can-svc

Namespace: default

Labels: <none>

Annotations: <none>

Selector: tier=frontend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

# IP: 10.96.3.3

IPs: 10.96.3.3

Port: <unset> 4741/TCP

TargetPort: 80/TCP

Endpoints: 10.244.2.12:80,10.244.2.13:80,10.244.3.11:80 + 1 more... 1+3=4

Session Affinity: None

Events: <none>

ibtisam@mint-dell:~/k8s/10-1$ docker exec -it ibtisam-worker bash

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the main production environment!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# exit

exit

ibtisam@mint-dell:~/k8s/10-1$ kubectl edit svc can-svc

service/can-svc edited

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe svc can-svc

Name: can-svc

Namespace: default

Labels: <none>

Annotations: <none>

Selector: env=canary,tier=frontend

Type: ClusterIP

IP Family Policy: SingleStack

IP Families: IPv4

# IP: 10.96.3.3

IPs: 10.96.3.3

Port: <unset> 4741/TCP

TargetPort: 80/TCP

Endpoints: 10.244.3.12:80 #reduced to one only

Session Affinity: None

Events: <none>

ibtisam@mint-dell:~/k8s/10-1$ docker exec -it ibtisam-worker bash

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# curl 10.96.3.3:4741

I'm the canary!

root@ibtisam-worker:/# exit

exit

ibtisam@mint-dell:~/k8s/10-1$

# namespace, resource quota

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 09.ns.yaml --force

namespace "ibtisam" deleted

resourcequota "abc-ns-rq" deleted

namespace/ibtisam replaced

resourcequota/abc-ns-rq replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get resourcequota -n ibtisam

# NAME AGE REQUEST LIMIT

abc-ns-rq 18s cpu: 0/3, memory: 0/5Gi, persistentvolumeclaims: 0/2, pods: 0/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 09.ns.yaml --force

namespace "ibtisam" deleted

resourcequota "abc-ns-rq" deleted

namespace/ibtisam replaced

resourcequota/abc-ns-rq replaced

Error from server (Forbidden): pods "abc" is forbidden: exceeded quota: abc-ns-rq, requested: memory=8Gi, used: memory=0, limited: memory=5Gi

ibtisam@mint-dell:~/k8s/10-1$ kubectl get ns,resourcequota -n ibtisam

# NAME STATUS AGE

namespace/default Active 21h

namespace/ibtisam Active 60s

namespace/kube-node-lease Active 21h

namespace/kube-public Active 21h

namespace/kube-system Active 21h

namespace/local-path-storage Active 21h

# NAME AGE REQUEST LIMIT

resourcequota/abc-ns-rq 59s cpu: 0/2, memory: 0/5Gi, persistentvolumeclaims: 0/2, pods: 0/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe ns ibtisam

Name: ibtisam

Labels: kubernetes.io/metadata.name=ibtisam

Annotations: <none>

Status: Active

Resource Quotas

Name: abc-ns-rq

Resource Used Hard

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

cpu 0 2

memory 0 5Gi

persistentvolumeclaims 0 2

pods 0 10

services 0 5

No LimitRange resource.

ibtisam@mint-dell:~/k8s/10-1$ kubectl get quota abc-ns-rq -n ibtisam

# NAME AGE REQUEST LIMIT

abc-ns-rq 4m53s cpu: 0/2, memory: 0/5Gi, persistentvolumeclaims: 0/2, pods: 0/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl get resourcequota abc-ns-rq -n ibtisam

# NAME AGE REQUEST LIMIT

abc-ns-rq 5m17s cpu: 0/2, memory: 0/5Gi, persistentvolumeclaims: 0/2, pods: 0/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 09.ns.yaml --force

namespace "ibtisam" deleted

resourcequota "abc-ns-rq" deleted

namespace/ibtisam replaced

resourcequota/abc-ns-rq replaced

pod/abc replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get resourcequota abc-ns-rq -n ibtisam

# NAME AGE REQUEST LIMIT

abc-ns-rq 37s cpu: 1/2, memory: 1536Mi/5Gi, persistentvolumeclaims: 0/2, pods: 1/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe resourcequota abc-ns-rq -n ibtisam

Name: abc-ns-rq

Namespace: ibtisam

Resource Used Hard

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

cpu 1 2

memory 1536Mi 5Gi

persistentvolumeclaims 0 2

pods 1 10

services 0 5

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl config view --minify --output yaml | grep namespace:

ibtisam@mint-dell:~/k8s/10-1$ kubectl config set-context --current --namespace ibtisam

Context "kind-ibtisam" modified.

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl config view --minify --output yaml | grep namespace:

namespace: ibtisam

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,resourcequota

# NAME READY STATUS RESTARTS AGE

pod/abc 1/1 Running 0 23m

# NAME AGE REQUEST LIMIT

resourcequota/abc-ns-rq 24m cpu: 1/2, memory: 1536Mi/5Gi, persistentvolumeclaims: 0/2, pods: 1/10, services: 0/5

ibtisam@mint-dell:~/k8s/10-1$ kubectl config set-context --current --namespace default

Context "kind-ibtisam" modified.

ibtisam@mint-dell:~/k8s/10-1$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

\* kind-ibtisam kind-ibtisam kind-ibtisam default

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,resourcequota

No resources found in default namespace.

ibtisam@mint-dell:~/k8s/10-1$ kubectl delete ns ibtisam # deleted everything created in that namespace

namespace "ibtisam" deleted

#services, ClusterIP & NodePort

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 08.svc.yaml

service/c-ip-svc created

pod/c-ip-pod created

service/np-svc created

pod/np-pod created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get nodes,po,svc -o wide

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

node/ibtisam-control-plane Ready control-plane 23h v1.30.0 172.18.0.5 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

node/ibtisam-control-plane2 Ready control-plane 23h v1.30.0 172.18.0.6 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

node/ibtisam-worker Ready <none> 23h v1.30.0 172.18.0.3 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

node/ibtisam-worker2 Ready <none> 23h v1.30.0 172.18.0.4 <none> Debian GNU/Linux 12 (bookworm) 6.6.32-linuxkit containerd://1.7.15

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/c-ip-pod 1/1 Running 0 19m 10.244.3.19 ibtisam-worker2 <none> <none>

pod/np-pod 1/1 Running 0 19m 10.244.2.20 ibtisam-worker <none> <none>

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

service/c-ip-svc ClusterIP 10.96.42.49 <none> 80/TCP 19m app=frontend

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 23h <none>

service/np-svc NodePort 10.96.49.7 <none> 8080:30080/TCP 19m flavor=blue

ibtisam@mint-dell:~/k8s/10-1$ docker exec -it ibtisam-worker bash

root@ibtisam-worker:/# curl 10.244.3.19:80 pod\_ip:service\_port

<h1>Welcome to nginx!</h1>

root@ibtisam-worker:/# curl 10.96.42.49:80 cluster\_ip:service\_port

<h1>Welcome to nginx!</h1>

root@ibtisam-worker:/# curl 10.244.2.20:8080 pod\_ip:service\_port

<!DOCTYPE html>

</html>root@ibtisam-worker:/# curl 10.96.49.7:8080 cluster\_ip:service\_port

<!DOCTYPE html>

ibtisam@mint-dell:~/k8s/10-1$ curl ifconfig.me

139.135.46.208ibtisam@mint-delip r l

default via 192.168.100.1 dev wlp6s0 proto dhcp src 192.168.100.10 metric 600

172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown

192.168.100.0/24 dev wlp6s0 proto kernel scope link src 192.168.100.10 metric 600

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward service c-ip-svc 3121:80 &

[1] 1311338

ibtisam@mint-dell:~/k8s/10-1$ Error from server (NotFound): pods "service" not found

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward service/c-ip-svc 3121:80 & ClusterIP, host:svc\_port

[1] 1312055

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward service/c-ip-svc 3121:80 > /dev/null 2>&1 &

[2] 1313603

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/np-svc 7575:8080 NodePort, host:svc\_port

Forwarding from 127.0.0.1:7575 -> 8080

Forwarding from [::1]:7575 -> 8080

Handling connection for 7575

Jobs & cron jobs

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 10.job.yaml

job.batch/test-job created

cronjob.batch/test-cronjob configured

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,job

# NAME READY STATUS RESTARTS AGE

pod/test-cronjob-28800910-f98gx 0/1 Completed 0 6m40s

pod/test-cronjob-28800915-4q8vp 0/1 Completed 0 100s

pod/test-job-9pk5p 1/1 Running 0 29s completion=1, pods generated=1

# NAME STATUS COMPLETIONS DURATION AGE

job.batch/test-cronjob-28800910 Complete 1/1 76s 6m40s

job.batch/test-cronjob-28800915 Complete 1/1 38s 100s

job.batch/test-job Running 0/1 29s 29s

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-9pk5p

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe job test-job

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 10.job.yaml

cronjob.batch/test-cronjob unchanged

The Job "test-job" is invalid: spec.completions: Invalid value: 5: field is immutable completion=5

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 10.job.yaml --force

job.batch "test-job" deleted

cronjob.batch "test-cronjob" deleted

job.batch/test-job replaced

cronjob.batch/test-cronjob replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,job job directly triggers pod to create.

NAME READY STATUS RESTARTS AGE deployment first generates rs, and rs then generates pods.

pod/test-cronjob-28800930-cgw8p 0/1 Completed 0 5m34s

pod/test-cronjob-28800935-g24d6 1/1 Running 0 34s

pod/test-job-2xhnq 0/1 Completed 0 8m22s 5 different successful pods are created.

pod/test-job-6f2hd 0/1 Completed 0 6m8s

pod/test-job-g6w8p 0/1 Completed 0 7m18s

pod/test-job-p2898 0/1 Completed 0 3m40s

pod/test-job-sbr65 0/1 Completed 0 5m9s

# NAME STATUS COMPLETIONS DURATION AGE

job.batch/test-cronjob-28800930 Complete 1/1 111s 5m34s

job.batch/test-cronjob-28800935 Running 0/1 34s 34s

job.batch/test-job Complete 5/5 5m28s 8m23s

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-2xhnq

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-6f2hd

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-g6w8p

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-p2898

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs test-job-sbr65

Hello from Kubernetes Job!

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe job test-job

Name: test-job

Namespace: default

Selector: batch.kubernetes.io/controller-uid=e746f3f2-9be8-4e3a-a5d8-40ffd2fb62c6

Labels: app=test-job

batch.kubernetes.io/controller-uid=e746f3f2-9be8-4e3a-a5d8-40ffd2fb62c6

batch.kubernetes.io/job-name=test-job

controller-uid=e746f3f2-9be8-4e3a-a5d8-40ffd2fb62c6

job-name=test-job

Annotations: <none>

Parallelism: 1

Completions: 5

Completion Mode: NonIndexed

Suspend: false

Backoff Limit: 4

Start Time: Fri, 04 Oct 2024 20:27:12 +0500

Completed At: Fri, 04 Oct 2024 20:32:40 +0500

Duration: 5m28s

Active Deadline Seconds: 600s

Pods Statuses: 0 Active (0 Ready) / 5 Succeeded / 0 Failed

## Pod Template:

Labels: app=test-job

batch.kubernetes.io/controller-uid=e746f3f2-9be8-4e3a-a5d8-40ffd2fb62c6

batch.kubernetes.io/job-name=test-job

controller-uid=e746f3f2-9be8-4e3a-a5d8-40ffd2fb62c6

job-name=test-job

## Containers:

## example-job:

Image: busybox

Port: <none>

Host Port: <none>

## Command:

sh

-c

echo Hello from Kubernetes Job! && sleep 30

Environment: <none>

Mounts: <none>

Volumes: <none>

Node-Selectors: <none>

Tolerations: <none>

## Events:

Type Reason Age From Message

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

Normal SuccessfulCreate 24m job-controller Created pod: test-job-2xhnq

Normal SuccessfulCreate 22m job-controller Created pod: test-job-g6w8p

Normal SuccessfulCreate 21m job-controller Created pod: test-job-6f2hd

Normal SuccessfulCreate 20m job-controller Created pod: test-job-sbr65

Normal SuccessfulCreate 19m job-controller Created pod: test-job-p2898

Normal Completed 18m job-controller Job completed

ibtisam@mint-dell:~/k8s/10-1$

# command & args

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f test.po.yaml

pod/test-pod created

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po test-pod | grep -i -A 1 args

## Args:

10

ibtisam@mint-dell:~/k8s/10-1$ kubectl edit po test-pod

error: pods "test-pod" is invalid

A copy of your changes has been stored to "/tmp/kubectl-edit-1961005729.yaml"

error: Edit cancelled, no valid changes were saved. Unlike others, change made in Args don’t save on editing.

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f /tmp/kubectl-edit-1961005729.yaml --force

pod "test-pod" deleted pod is deleted & then newly created.

pod/test-pod replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po

# NAME READY STATUS RESTARTS AGE

test-pod 1/1 Running 0 23s

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po test-pod | grep -i -A 1 args

## Args:

100

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl run abcd --image kodekloud/webapp-color -- --color red

pod/abcd created

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po abcd | grep -i -A 3 args

## Args:

--color

red

State: Waiting

ibtisam@mint-dell:~/k8s/10-1$ kubectl expose po abcd --port 8080 --name abcd-svc --type NodePort

service/abcd-svc exposed

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/abcd-svc 6969:8080 > /dev/null 2>&1 &

[1] 1828999

ibtisam@mint-dell:~/k8s/10-1$ curl localhost:6969

<!doctype html>

<title>Hello from Flask</title>

<body style="background: #e74c3c;"></body>

<div style="color: #e4e4e4;

text-align: center;

height: 90px;

vertical-align: middle;">

<h1>Hello from abcd!</h1>

</div>ibtisam@mint-dell:~/k8s/10-1$ kubectl edit po abcd

error: pods "abcd" is invalid

A copy of your changes has been stored to "/tmp/kubectl-edit-2125647988.yaml"

error: Edit cancelled, no valid changes were saved.

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f /tmp/kubectl-edit-2125647988.yaml --force

pod "abcd" deleted

pod/abcd replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po abcd | grep -i -A 3 args

## Args:

--color

blue

State: Running

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/abcd-svc 6969:8080 > /dev/null 2>&1 &

[2] 1833512

[1] Exit 1 kubectl port-forward svc/abcd-svc 6969:8080 > /dev/null 2>&1

ibtisam@mint-dell:~/k8s/10-1$ curl localhost:6969

<!doctype html>

<title>Hello from Flask</title>

<body style="background: #2980b9;"></body>

<div style="color: #e4e4e4;

text-align: center;

height: 90px;

vertical-align: middle;">

<h1>Hello from abcd!</h1>

</div>ibtisam@mint-dell:~/k8s/10-1$

#env, config map, secret

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 11.cm-sec.yaml

configmap/abc-cm unchanged

pod/pod-without-configmap-env unchanged

pod/pod-using-configmap-env created

pod/pod-using-configmap-specific-env unchanged

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs pod-without-configmap-env

red

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs pod-using-configmap-env

red

/opt

ibtisam@mint-dell:~/k8s/10-1$ kubectl logs pod-using-configmap-specific-env

red

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po pod-without-configmap-env | grep -i -A 3 environment

## Environment:

APP\_COLOR: red

## Mounts:

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

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po pod-using-configmap-env | grep -i -A 3 environment

## Environment Variables from:

abc-cm ConfigMap Optional: false

Environment: <none>

## Mounts:

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

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po pod-using-configmap-specific-env | grep -i -A 3 environment

## Environment:

APP\_COLOR: <set to the key 'APP\_COLOR' of config map 'abc-cm'> Optional: false

## Mounts:

/var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-nm7kt (ro)

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec -it pod-using-configmap-volume-specific-env -- /bin/sh

/ # ls

bin etc lib proc product\_uuid sys usr

dev home lib64 product\_name root tmp var

/ # env

KUBERNETES\_PORT=tcp://10.96.0.1:443

# KUBERNETES\_SERVICE\_PORT=443

HOSTNAME=pod-using-configmap-volume-specific-env

# SHLVL=1

HOME=/root

TERM=xterm

# KUBERNETES\_PORT\_443\_TCP\_ADDR=10.96.0.1

PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin

# KUBERNETES\_PORT\_443\_TCP\_PORT=443

KUBERNETES\_PORT\_443\_TCP\_PROTO=tcp

APP\_COLOR=red

KUBERNETES\_PORT\_443\_TCP=tcp://10.96.0.1:443

# KUBERNETES\_SERVICE\_PORT\_HTTPS=443

# KUBERNETES\_SERVICE\_HOST=10.96.0.1

# PWD=/

/ # cd /etc/config

/etc/config # ls

# APP\_COLOR

/etc/config # cat APP\_COLOR

red/etc/config #

# YAML sh -c

command: ['sh', '-c', 'echo $APP\_COLOR', 'ls -l /etc/config', 'cat /etc/config/APP\_COLOR', 'sleep 3600 && echo $PWD'] so this is also wrong?

## ChatGPT said:

ChatGPT

Yes, the structure in this command is incorrect. In Kubernetes, the command field expects each part of the command to be a separate string within the list. However, when you use multiple commands like echo, ls, cat, and sleep, they all need to be part of a single string when using sh -c.

In this configuration, Kubernetes interprets each part (e.g., 'echo $APP\_COLOR', 'ls -l /etc/config', etc.) as separate commands, which results in errors because sh -c can only execute a single string of commands.

## Solution:

You need to combine all the commands into a single string, separated by && or line breaks within the -c section of the sh shell. This ensures they are run sequentially in the same shell.

## Correct YAML:

## command:

- sh

- -c

- |

echo $APP\_COLOR

ls -l /etc/config

cat /etc/config/APP\_COLOR

sleep 3600 && echo $PWD

## Explanation:

sh -c: The -c flag tells the shell to execute the string of commands.

Multi-line commands: Using | (a multi-line string in YAML), all commands are treated as a single string. The commands will be executed in order:

echo $APP\_COLOR: prints the value of the APP\_COLOR environment variable.

ls -l /etc/config: lists files in the /etc/config directory.

cat /etc/config/APP\_COLOR: outputs the content of the APP\_COLOR file.

sleep 3600 && echo $PWD: pauses for 3600 seconds and then prints the current working directory.

command: ['sh', '-c', 'echo $APP\_COLOR && ls -l /etc/config && cat /etc/config/APP\_COLOR && sleep 3600'] so this is also corrected?

## ChatGPT said:

ChatGPT

Yes, your updated command is now correct. By combining the commands into a single string and connecting them with &&, you are ensuring they execute sequentially in the same shell instance. Here's how it looks:

command: ['sh', '-c', 'echo $APP\_COLOR && ls -l /etc/config && cat /etc/config/APP\_COLOR && sleep 3600']

## Explanation:

sh -c: The -c flag tells the shell to interpret the rest of the string as a command.

Chaining with &&: The && operator ensures that each command only runs if the previous one succeeds.

Warning: resource configmaps/abc-cm is missing the kubectl.kubernetes.io/last-applied-configuration annotation which is required by kubectl apply. kubectl apply should only be used on resources created declaratively by either kubectl create --save-config or kubectl apply. The missing annotation will be patched automatically.

Why use --save-config with kubectl create?

If you initially create a resource using kubectl create without --save-config, then later use kubectl apply to modify it, you might get a warning like the one you saw. This happens because Kubernetes doesn't know the previous configuration since no annotation was added when the resource was created.

If you use --save-config when creating the resource, Kubernetes will store the original configuration, allowing kubectl apply to work correctly in the future without warnings.

## When to Use Each Command:

Use kubectl apply when you plan to manage your resources using YAML files, because it tracks configuration changes automatically.

Use kubectl create --save-config when you prefer to use create but still want to track the configuration for future changes using kubectl apply.

## Example of Applying and Creating Resources:

kubectl apply -f resource.yaml: Applies the resource from the YAML file and tracks changes automatically.

kubectl create -f resource.yaml --save-config: Creates the resource and saves its configuration for future updates using apply.

# security contexts

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 06.po.sec-con.yaml

pod/sec-con-po-0 created

pod/sec-con-po-1 created

pod/sec-con-po-2 created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

sec-con-po-0 1/1 Running 0 4m51s 10.244.3.85 ibtisam-worker <none> <none>

sec-con-po-1 1/1 Running 0 4m50s 10.244.2.80 ibtisam-worker2 <none> <none>

sec-con-po-2 2/2 Running 0 4m50s 10.244.3.86 ibtisam-worker <none> <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec sec-con-po-0 -- whoami

root

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec -it sec-con-po-0 -- /bin/sh

/ # whoami

root

/ # exit

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec sec-con-po-1 -- whoami

whoami: unknown uid 1001

command terminated with exit code 1

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec -it sec-con-po-1 -- /bin/sh

~ $ whoami

whoami: unknown uid 1001

~ $ exit

command terminated with exit code 1

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec sec-con-po-2 -- whoami

Defaulted container "sec-con-cont-1" out of: sec-con-cont-1, sec-con-cont-2

whoami: unknown uid 1002

command terminated with exit code 1

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec sec-con-po-2 -c sec-con-cont-1 -- whoami

whoami: unknown uid 1002

command terminated with exit code 1

ibtisam@mint-dell:~/k8s/10-1$ kubectl exec sec-con-po-2 -c sec-con-cont-2 -- whoami

whoami: unknown uid 1000

command terminated with exit code 1

Service account

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 12.sa-r-rb.yaml

pod/dashboard-app created

service/dashboard-app-svc created

serviceaccount/dashboard-app-sa created

role.rbac.authorization.k8s.io/dashboard-app-po-reader created

rolebinding.rbac.authorization.k8s.io/dashboard-app-po-reader-binding created

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po,svc,sa,role,rolebinding -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

pod/dashboard-app 1/1 Running 0 2m28s 10.244.2.81 ibtisam-worker2 <none> <none>

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

service/dashboard-app-svc NodePort 10.96.74.67 <none> 7575:30240/TCP 2m27s app=dashboard

service/kubernetes ClusterIP 10.96.0.1 <none> 443/TCP 2d20h <none>

# NAME SECRETS AGE

serviceaccount/dashboard-app-sa 0 2m21s

serviceaccount/default 0 2d20h

# NAME CREATED AT

role.rbac.authorization.k8s.io/dashboard-app-po-reader 2024-10-08T04:01:09Z

# NAME ROLE AGE USERS GROUPS SERVICEACCOUNTS

rolebinding.rbac.authorization.k8s.io/dashboard-app-po-reader-binding Role/dashboard-app-po-reader 2m19s ibtisam dev-team default/dashboard-app-sa

ibtisam@mint-dell:~/k8s/10-1$ kubectl port-forward svc/dashboard-app-svc 6565:7575 > /dev/null 2>&1 &

[1] 3326587

ibtisam@mint-dell:~/k8s/10-1$ kubectl create token dashboard-app-sa

eyJhbGciOiJSUzI1NiIsImtpZCI6IlBxRFpHa1dTbHVzTWQzcnhoN20wRWhuSjJZTjZHV1E5T2MwVWwzT2R2Q00ifQ..E9VU87FtyU02Gg-Pbo2\_9etX1BR6NrcJR79uXXHTC0wP1YXneVtZyGcDoPRdwJBcQlqpd95FPfyXMNXKSa-\_4HiN0o-sWR3hn7KIJm9V2T6BVQKYCa3ZC9v4t10ymLt7t0IeY6q7PTSoTljUhqWLuF-8g86cuuxG9578bQLZgfreezAmRVd2B6nH-\_Ab0HDPRfuIUOCTffLvSydhAW78Hv4JiVwm7UXVf4qNgIpF\_0rW0xfEvd6C4BDOQpbPHa\_ccoPrPW7SvtW255eiedT8OxldFyAmW7IEeEIwppZ-SQ-PgYNganul0TYFghckNnxsny0YSzLv3-TnSh3deyFNyw

ibtisam@mint-dell:~/k8s/10-1$ kubectl get sa

# NAME SECRETS AGE

dashboard-app-sa 0 10m

default 0 2d21h

ibtisam@mint-dell:~/k8s/10-1$

# taints & toleration # node selector # node affinity

ibtisam@mint-dell:~/k8s/10-1$ kubectl get nodes

# NAME STATUS ROLES AGE VERSION

ibtisam-control-plane Ready control-plane 3d v1.30.0

ibtisam-control-plane2 Ready control-plane 3d v1.30.0

ibtisam-worker Ready <none> 3d v1.30.0

ibtisam-worker2 Ready <none> 3d v1.30.0

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-control-plane | grep -i taint

Taints: node-role.kubernetes.io/control-plane:NoSchedule

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-control-plane2 | grep -i taint

Taints: node-role.kubernetes.io/control-plane:NoSchedule

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i taint

Taints: <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker2 | grep -i taint

Taints: <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl taint node ibtisam-worker flower=rose:NoSchedule

node/ibtisam-worker tainted

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i taint

Taints: flower=rose:NoSchedule

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker2 | grep -i taint

Taints: <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl apply -f 06.po.tai-tol.yaml

pod/test-pod created

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe po test-pod | grep -i toleration

Tolerations: flower=rose:NoSchedule

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

test-pod 1/1 Running 0 2m34s 10.244.2.82 ibtisam-worker2 <none> <none>

ibtisam@mint-dell:~/k8s/10-1$ kubectl taint node ibtisam-worker flower=rose:NoSchedule-

node/ibtisam-worker untainted

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i taint

Taints: <none>

The node is tainted, but the desired pod is scheduled on undesired node.

ibtisam@mint-dell:~/k8s/10-1$ kubectl taint node ibtisam-worker flower=rose:NoSchedule

node/ibtisam-worker tainted

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker disktype=ssd

node/ibtisam-worker labeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker disktype=hdd

error: 'disktype' already has a value (ssd), and --overwrite is false

# NAME STATUS ROLES AGE VERSION

ibtisam-control-plane Ready control-plane 3d7h v1.30.0

ibtisam-control-plane2 Ready control-plane 3d7h v1.30.0

ibtisam-worker Ready <none> 3d7h v1.30.0

ibtisam-worker2 Ready <none> 3d7h v1.30.0

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i label -5

Name: ibtisam-worker

Roles: <none>

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

disktype=ssd

kubernetes.io/arch=amd64

kubernetes.io/hostname=ibtisam-worker

kubernetes.io/os=linux

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i taint

Taints: flower=rose:NoSchedule

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 06.po.tai-tol.yaml --force

pod "plain-po" deleted

pod "tol-po" deleted

pod "ha-po" deleted

pod "sa-po" deleted

pod "tol-ha-po" deleted

pod/plain-po replaced

pod/tol-po replaced

pod/nodeselector-po replaced

pod/ha-po replaced

pod/sa-po replaced

pod/tol-ha-po replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

NAME READY STATUS RESTARTS AGE IP NODE ibtisam-worker is tainted & labeled.

ha-po 0/1 Pending 0 23m <none> <none> po; hard affinity added only but no toleration

nodeselector-po 0/1 Pending 0 23m <none> <none> one node is tainted, other one lacks of desired label

plain-po 1/1 Running 0 23m 10.244.2.8 ibtisam-worker2 one node is tainted, other one not tainted, so here it is

sa-po 1/1 Running 0 23m 10.244.2.9 ibtisam-worker2 one node is tainted, other one not tainted, so here it is

tol-ha-po 1/1 Running 0 23m 10.244.3.8 ibtisam-worker both ha & tol, target achieved.

tol-po 1/1 Running 0 23m 10.244.3.7 ibtisam-worker luckily scheduled on desired node, only tol

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker cpu=large

node/ibtisam-worker labeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl describe node ibtisam-worker | grep -i label -5

Name: ibtisam-worker

Roles: <none>

Labels: beta.kubernetes.io/arch=amd64

beta.kubernetes.io/os=linux

cpu=large

disktype=ssd

kubernetes.io/arch=amd64

kubernetes.io/hostname=ibtisam-worker

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE

ha-po 0/1 Pending 0 34m <none> <none>

nodeselector-po 0/1 Pending 0 34m <none> <none> same issue

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker2 cpu=large

node/ibtisam-worker2 labeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE

ha-po 0/1 Pending 0 40m <none> <none>

nodeselector-po 1/1 Running 0 40m 10.244.2.10 ibtisam-worker2 one is tainted, 2nd has got the desired labels.

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker2 cpu-

node/ibtisam-worker2 unlabeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker cpu-

node/ibtisam-worker unlabeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl label node ibtisam-worker disktype-

node/ibtisam-worker unlabeled

ibtisam@mint-dell:~/k8s/10-1$ kubectl taint node ibtisam-worker flower=rose:NoSchedule-

node/ibtisam-worker untainted

ibtisam@mint-dell:~/k8s/10-1$

ibtisam@mint-dell:~/k8s/10-1$ kubectl replace -f 06.po.tai-tol.yaml --force

pod "plain-po" deleted

pod "tol-po" deleted

pod "nodeselector-po" deleted

pod "ha-po" deleted

pod "sa-po" deleted

pod "tol-ha-po" deleted

pod/plain-po replaced

pod/tol-po replaced

pod/nodeselector-po replaced

pod/ha-po replaced

pod/sa-po replaced

pod/tol-ha-po replaced

ibtisam@mint-dell:~/k8s/10-1$ kubectl get po -o wide

# NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES

ha-po 0/1 Pending 0 42s <none> <none> <none> <none>

nodeselector-po 0/1 Pending 0 42s <none> <none> <none> <none>

plain-po 1/1 Running 0 42s 10.244.2.15 ibtisam-worker2 <none> <none>

sa-po 1/1 Running 0 41s 10.244.3.10 ibtisam-worker <none> <none>

tol-ha-po 0/1 Pending 0 41s <none> <none> <none> <none>

tol-po 1/1 Running 0 42s 10.244.2.14 ibtisam-worker2 <none> <none>

# Monitoring & Debugging

ibtisam@mint-dell:~/k8s/10-1$ 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

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy -A

# NAMESPACE NAME READY UP-TO-DATE AVAILABLE AGE

kube-system coredns 2/2 2 2 57m

kube-system metrics-server 0/1 1 0 111s

local-path-storage local-path-provisioner 1/1 1 1 57m

ibtisam@mint-dell:~/k8s/10-1$ kubectl patch -n kube-system deployment metrics-server --type=json -p '[{"op":"add","path":"/spec/template/spec/containers/0/args/-","value":"--kubelet-insecure-tls"}]'

deployment.apps/metrics-server patched

ibtisam@mint-dell:~/k8s/10-1$ kubectl get deploy -A

# NAMESPACE NAME READY UP-TO-DATE AVAILABLE AGE

kube-system coredns 2/2 2 2 71m

kube-system metrics-server 1/1 1 1 14m

local-path-storage local-path-provisioner 1/1 1 1 70m

ibtisam@mint-dell:~/k8s/10-1$ kubectl top pod

No resources found in default namespace.

ibtisam@mint-dell:~/k8s/10-1$ kubectl top node

NAME CPU(cores) CPU% MEMORY(bytes) MEMORY%

ibtisam-control-plane 484m 12% 546Mi 29%

ibtisam-control-plane2 383m 9% 404Mi 21%

ibtisam-worker 95m 2% 112Mi 5%

ibtisam-worker2 114m 2% 184Mi 9%

ibtisam@mint-dell:~/k8s/10-1$

# ingress

ibtisam@mint-dell:~$ kubectl config get-contexts

# CURRENT NAME CLUSTER AUTHINFO NAMESPACE

kind-ibtisam kind-ibtisam kind-ibtisam

\* minikube minikube minikube default

ibtisam@mint-dell:~$ kubectl config get-clusters

# NAME

kind-ibtisam

minikube

ibtisam@mint-dell:~$ minikube addons enable ingress

💡 ingress is an addon maintained by Kubernetes. For any concerns contact minikube on GitHub.

You can view the list of minikube maintainers at: https://github.com/kubernetes/minikube/blob/master/OWNERS

▪ Using image registry.k8s.io/ingress-nginx/controller:v1.10.1

▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1

▪ Using image registry.k8s.io/ingress-nginx/kube-webhook-certgen:v1.4.1

🔎 Verifying ingress addon...

🌟 The 'ingress' addon is enabled

ibtisam@mint-dell:~$ kubectl get ns | grep -i ingress

ingress-nginx Active 14m

ibtisam@mint-dell:~$ kubectl get all -A | grep -i ingress

ingress-nginx pod/ingress-nginx-admission-create-f647d 0/1 Completed 0 14m

ingress-nginx pod/ingress-nginx-admission-patch-n5xxv 0/1 Completed 0 14m

ingress-nginx pod/ingress-nginx-controller-768f948f8f-2t2rf 1/1 Running 0 14m

ingress-nginx service/ingress-nginx-controller NodePort 10.106.99.122 <none> 80:30648/TCP,443:31784/TCP 14m

ingress-nginx service/ingress-nginx-controller-admission ClusterIP 10.103.167.108 <none> 443/TCP 14m

ingress-nginx deployment.apps/ingress-nginx-controller 1/1 1 1 14m

ingress-nginx replicaset.apps/ingress-nginx-controller-768f948f8f 1 1 1 14m

ingress-nginx job.batch/ingress-nginx-admission-create Complete 1/1 69s 14m

ingress-nginx job.batch/ingress-nginx-admission-patch Complete 1/1 70s 14m

ibtisam@mint-dell:~$