**Namespaces**

* Kubectl get namespace ( there is no namespace )
* Vi devns.yml

( eg:- apiVersion: v1

kind: Namespace

metadata:

name: dev

labels:

name: dev )

* Kubectl apply -f devns.yml
* Kubectl get namespace ( your namespace created )
* Vi pod.yml

( eg:- kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Technical Guftgu; sleep 5 ; done"]

* Kubectl apply -f pod.yml -n dev ( your pod will create in dev namespace )
* Kubectl get pods
* Kubectl get pods -n dev ( to see pod in dev namespace )
* Kubectl delete -f pod.yml -n dev
* Kubectl apply -f pod.yml -n dev
* kubectl config set-context $(kubectl config current-context) --namespace=dev (to set dev name space )
* kubectl get pods ( now you can see pods, because you are in dev namespace )
* kubectl config view | grep namespace : ( you will see the dev namespace )

**Resource Quota**

* vi podresources.yml

(eg:- apiVersion: v1

kind: Pod

metadata:

name: resources

spec:

containers:

- name: resource

image: centos

command: ["/bin/bash", "-c", "while true; do echo Technical-Guftgu; sleep 5 ; done"]

resources:

requests:

memory: "64Mi"

cpu: "100m"

limits:

memory: "128Mi"

cpu: "200m" )

* kubectl apply -f podresources.yml ( all pod will be create in dev namespace )
* kubectl get pods
* kubectl describe pod resources (to see all information about pod )
* kubectl delete -f podresources.yml
* vi resourcequota.yml

( eg:- apiVersion: v1

kind: ResourceQuota

metadata:

name: myquota

spec:

hard:

limits.cpu: "400m"

limits.memory: "400Mi"

requests.cpu: "200m"

requests.memory: "200Mi" )

* kubectl apply -f resourcequota.yml
* vi testpod.yml

( eg:- kind: Deployment

apiVersion: apps/v1

metadata:

name: deployments

spec:

replicas: 3

selector:

matchLabels:

objtype: deployment

template:

metadata:

name: testpod8

labels:

objtype: deployment

spec:

containers:

- name: c00

image: ubuntu

command: ["/bin/bash", "-c", "while true; do echo Technical-Guftgu; sleep 5 ; done"]

resources:

requests:

cpu: "200m" )

* kubectl apply -f testpod.yml ( your pod will not be create coz you have only 400 mi cpu so 3 pod cannot create )
* Kubectl get pods ( o/p:- 0 pod running )
* kubectl get rs
* kubectl describe rs deployment -fuhfefw ( o/p:- failed quota )
* kubectl delete -f resourcequota.yml
* kubectl delete -f testpod.yml
* vi cpudefault.yml

( eg :- apiVersion: v1

kind: LimitRange

metadata:

name: cpu-limit-range

spec:

limits:

- default:

cpu: 1

defaultRequest:

cpu: 0.5

type: Container )

* kubectl apply -f cpudefault.yml
* kubectl apply -f pod.yml
* kubectl get pods
* kubectl describe pod testpod
* kubectl delete -f pod.yml
* vi cpu2.yml

( eg:- apiVersion: v1

kind: Pod

metadata:

name: default-cpu-demo-2

spec:

containers:

- name: default-cpu-demo-2-ctr

image: nginx

resources:

limits:

cpu: "1" )

* kubectl apply -f cpu2.yml
* kubectl get pods
* kubectl describe pod default-pod ( your request will be 1 cpu )
* kubectl delete -f cpu2.yml
* vi cpu3.yml

( eg:- apiVersion: v1

kind: Pod

metadata:

name: default-cpu-demo-3

spec:

containers:

- name: default-cpu-demo-3-ctr

image: nginx

resources:

requests:

cpu: "0.75" )

* kubectl apply -f cpu3.yml
* kubectl describe pod mypod.yml ( when define request and does not define limit than you will get default limit like-1cpu )
* kubectl delete -f cpu3.yml