

## Edit Pods

kubectl edit pod <pod-name>

kubectl get pod <pod-name> -o yaml > pod-def.yaml

↳ Extract definition to a file

--dry-run=client → This will not create resource

## Imperative commands

### Service

kubectl expose pod redis --port=6379 --name=redis

service → by default type will be clusterIP

--type=NodePort

## Deployment

kubectl create deployment nginx --image=nginx

kubectl scale deployment nginx --replicas=4

## Pod

kubectl run nginx --image=nginx

## Commands & Arguments

In spec, under containers we can mention custom entrypoint & arguments for the containers.

Spec:

- containers:

- name:

- image:

- command: ["~~co~~nsleep"]

- args: ["10"]

→ command  
(sleep 10)

↓  
Entrypoint

## Env variables

- env:

- name:

- value:

Types

Plain  
Key  
value

or

- env:

- name:

- valueFrom:

- configMapKeyRef:

ConfigMap

or

- env:

- name:

- valueFrom:

- secretKeyRef:

Secrets

# Create ConfigMaps

Imperative

use keyword  
generic for  
secret

Declarative

Should be in  
encoded format  
for secrets.

base64

Replace configmaps  
with secrets

Same for Secrets

i) kubectl create configmap <config-name> --from-literal-  
<key>=<values>

ex: kubectl create configmap app-config  
--from-literal=APP-COLOR=blue  
--from-literal=APP-MODE=prod

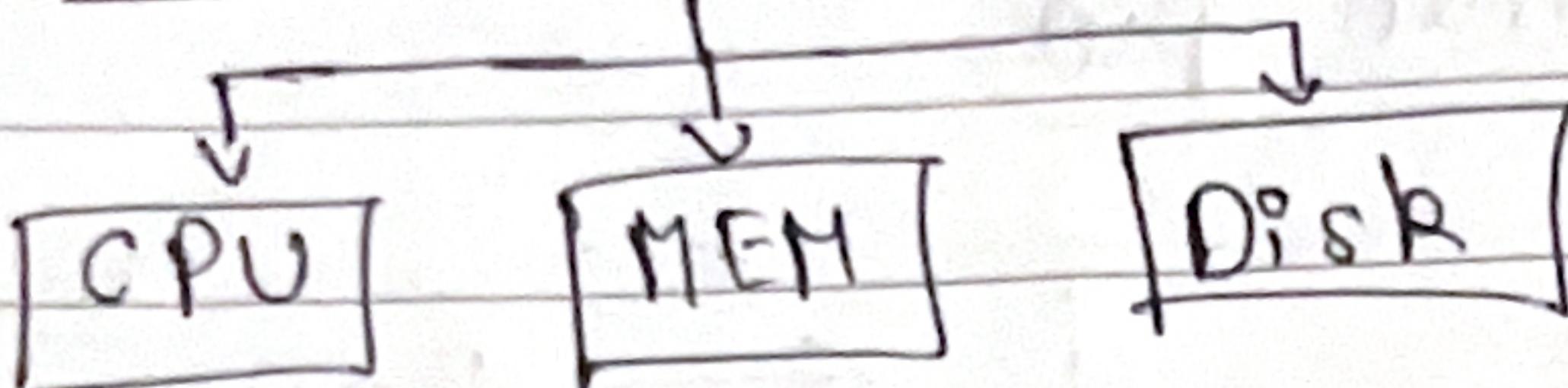
ii) kubectl create configmap <config-name>  
--from-file=<path-to-files>

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: app-config
data:
  APP-COLOR: blue
  APP-MODE: prod
```

← Config-map.yaml

kubectl create -f config-map.yaml

## Resource Requests



mention  
under  
containers

resources:

requests:

memory: "1Gi"

cpu: 1

Default

1 vCPU

512 Mi

## Resource Limits

limits:

memory: "2Gi"

cpu: 2

Under resource

## Accounts

User

Admin

Developer

Service

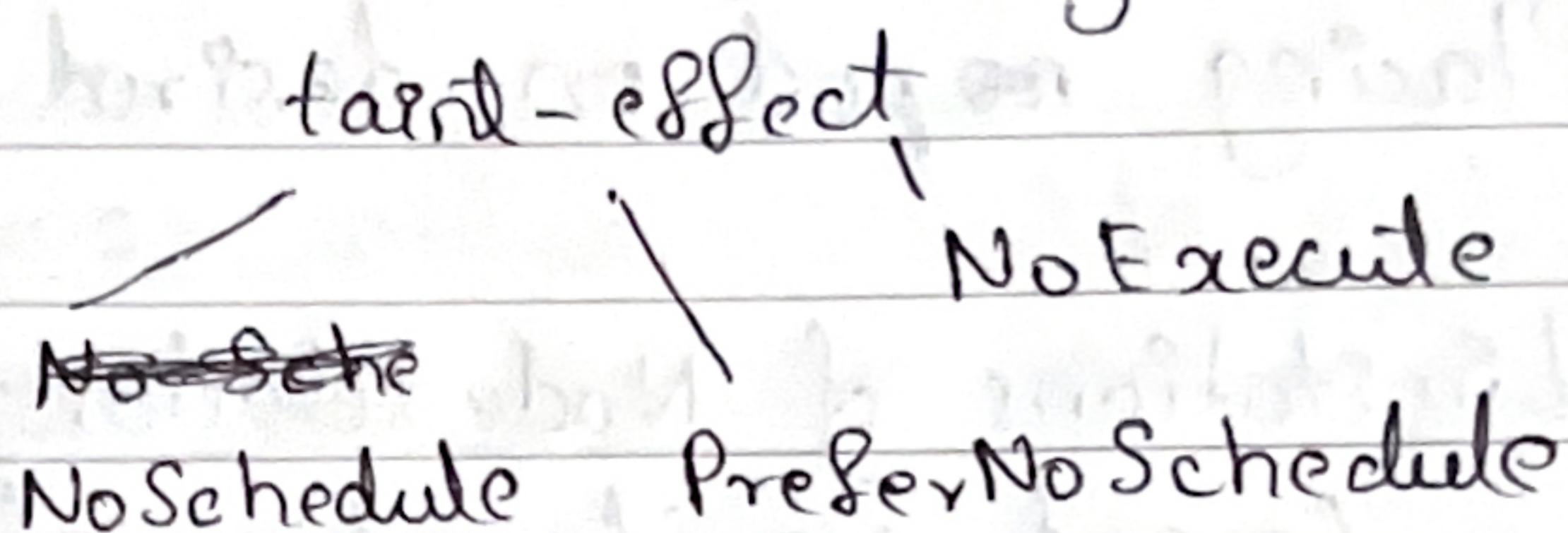
Ex: Prometheus

Jenkins

## Taints - Node

command:

kubectl taint nodes <node-names> <key>=<value>:



Ex: kubectl taint nodes node1 app=blue:NoSchedule

## Tolerations - Pods

tolerations:

- key: "app"
- operator: "Equal"
- value: "blue"
- effect: "NoSchedule"

placed under  
spec in pod-def  
file

Note: Taints & tolerations does not tell pod to go to a particular node. Instead, it tells the node to only accept pods with certain tolerations

## Node Selectors

First label node -

kubectl label nodes <node-name> <label-key>=<label-value>

Ex: kubectl label nodes node-1 size=large

nodeSelector  
size: Large

Place under  
spec in pod-def file

\* Placing ~~no~~ pod on desired node.

\* Limitations of Node selectors:

Cannot provide advanced expressions like OR & NOT. ex: Large or medium not small

solution

~ Node Affinity

placed under  
spec

affinity:

nodeAffinity :

requiredDuringScheduling IgnoredDuringExecution:

nodeSelectorTerms:

- matchExpressions:
  - key : size

operator : In

// can be NotIn

values:

- Large
- Medium

// For only key used

// operator Exists

Types

- requiredDuringScheduling IgnoredDuringExecution
- preferredDuringScheduling IgnoredDuringExecution