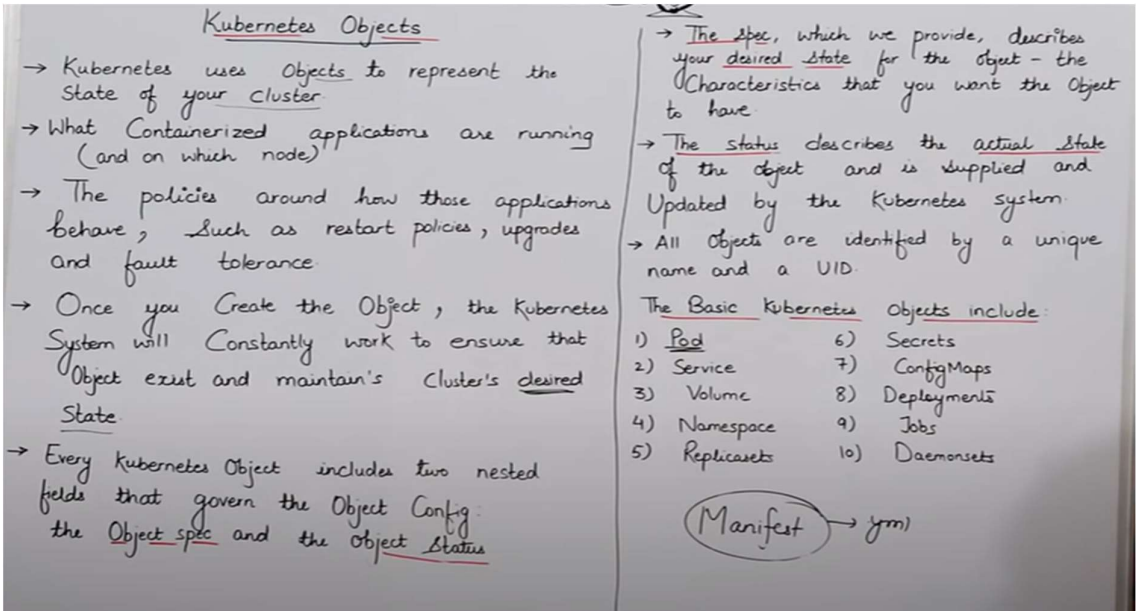
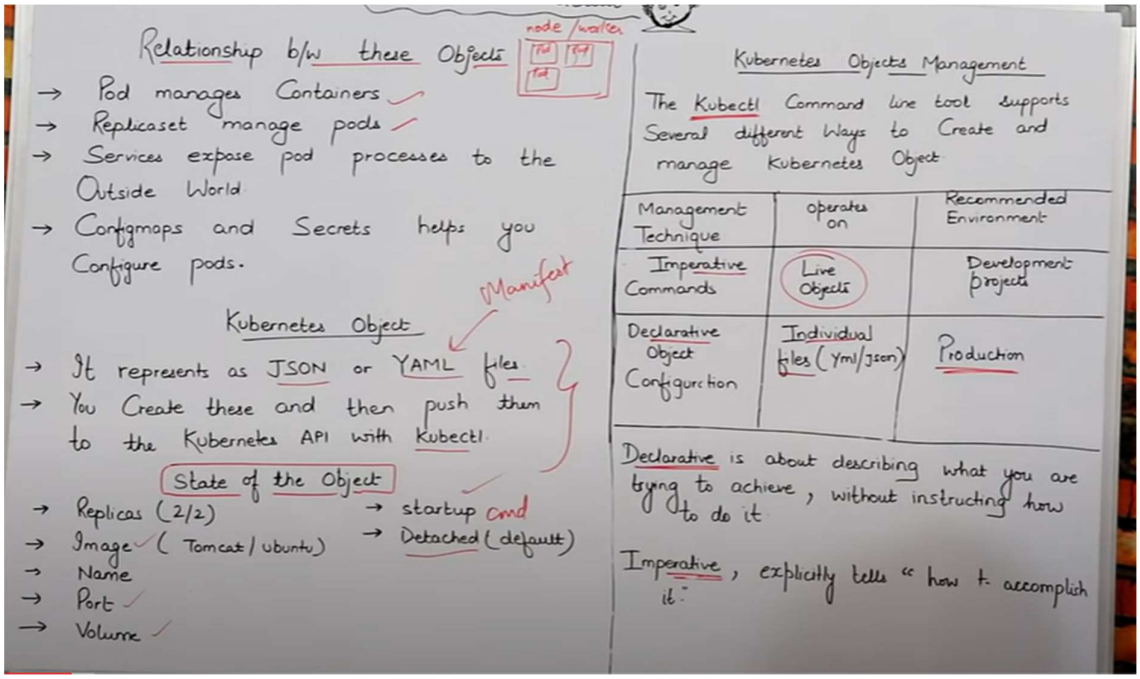
Installation of Minikube and Detailed



➔ Hum manifest file mein jo bhi likh kar denge kubernetes puri koshish karega ki us desired state ko pura kare matlab jo bhi kuch humare manifest file mein likha hai usko ye pura kar de.

➔ Har eak object ka alag unique name aur aur alag unique id hota hai.

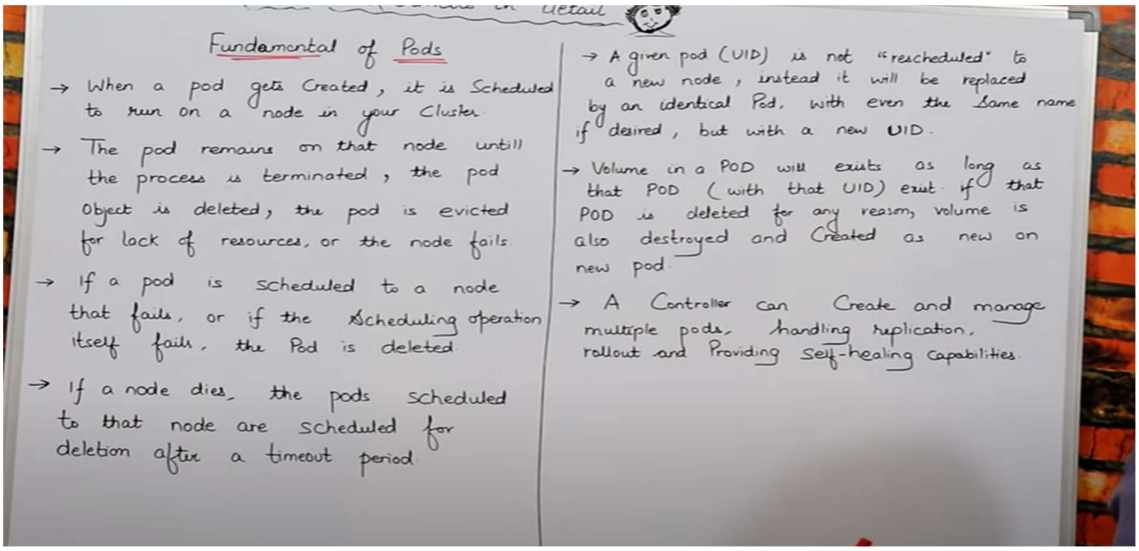
➔ Hum manifest file ko json aur yml mein lkhate hai.



➔ Kubectl eak command line tool hota hai, Kubectl kein through hi hum sariii command likhenge, starting mein kubectl then uske baad jo command dena chahhenge vo.

➔ **Declarative** = declarative ka matlab hai ki agar hum kuch chahate hai toh usko eak file kein andar describe kar dete hai ki mujhe ye ye chahiye aur vo file jab execute hoti hai toh sab apane aap run ho jata hai.

➔ **Imperative** = Imperative matlab hai ki humkp jo bhi cheej chahihe har eak ko command mein baar baar likho then execute karao.



==========

➔ **Create 1 ec2 machine :-** 2vcpu, t2medium

➔ ubuntu@ip-172-31-12-153:~$ sudo su
➔ root@ip-172-31-12-153:/home/ubuntu# apt update
➔ root@ip-172-31-12-153:/home/ubuntu# apt install docker

➔ root@ip-172-31-12-153:/home/ubuntu# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl

```
Select root@ip-172-31-12-153: /home/ubuntu
root@ip-172-31-12-153:/home/ubuntu# curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl && chmod +x ./kubectl && sudo mv ./kubectl /usr/local/bin/kubectl
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 44.4M  100 44.4M    0     0  54.9M      0 --:--:-- --:--:-- --:--:-- 54.8M
root@ip-172-31-12-153:/home/ubuntu#
```

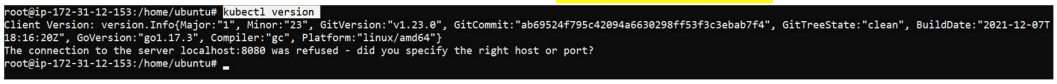
Iss command sein hum kubectl ko download karke install karenge, aise karane sein humara command line interface start ho jayega aisa isliye kyuki aage hum jo bhi kaam karenge kubectl command ki help sein karenge.

➔ root@ip-172-31-12-153:/home/ubuntu# which kubectl

```
root@ip-172-31-12-153:/home/ubuntu# which kubectl
/usr/local/bin/kubectl
root@ip-172-31-12-153:/home/ubuntu#
```

(hum dekh sakte hai ki humara kubectl kaha install huva hai)

➔ root@ip-172-31-12-153:/home/ubuntu# kubectl version



(kubectl ka version check kar sakte hai hum)

==========

➔ **Example =** agar mere pass **1 master** , aur **3 worker node** hai, aur mujhe pod banana hai , toh mujhe manifest file kein andar batana padega ki **Node2** kein andar jaakar humara pod create karo , aur agar hum ye nahi bataye ki kis node par banana hai pod , toh kubernetes apane aap jis node kein andar usko samjh aayega vo apane aap us node kein andar jaakar pod create kar dega .

➔ Pod humara tab tak fail nahi hota jab tak koi process fail na ho jaye , koi pod delete na ho jaye , aur agar humara node hi band ho gaya toh humara pod bhi easily sein cancel ho jayega.

➔ Agara humane kisi node par pod create kiya hai aur vo node hi fail hogaya toh us case mein mera pod bhi fail ho jayega .

➔ Agar humane koi pod bana rakha hai node kein andar aur vo node fail hogaya toh humara pod eak time period tak active rahega agar uss time period tak node apane aap ko active nahi kar paya toh humara pod bhi cancel ho jayega.

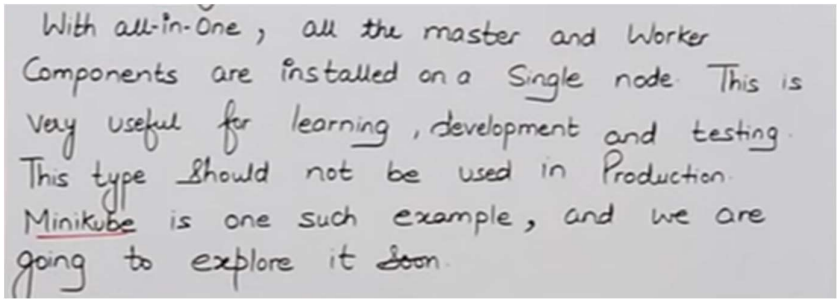
➔ Agar humara pod fail hoajata hai toh vahi same pod restart nahi hoga kubernetes eak naya same pod bana dega aur jo pod fail huva hoga uski sari dependencies new pod mein aajyegi aur pod naya bana dega, aur humko pata kaise chelega ki naya pod kubernetes ne create kiya hai ya vahi purana vala hi restart kar diya toh us case mein hum dekhenge pura jo pod hoga uski **UID(unique id)** alag hogi aur naye pod ki **UID(unique id)** different hogi.

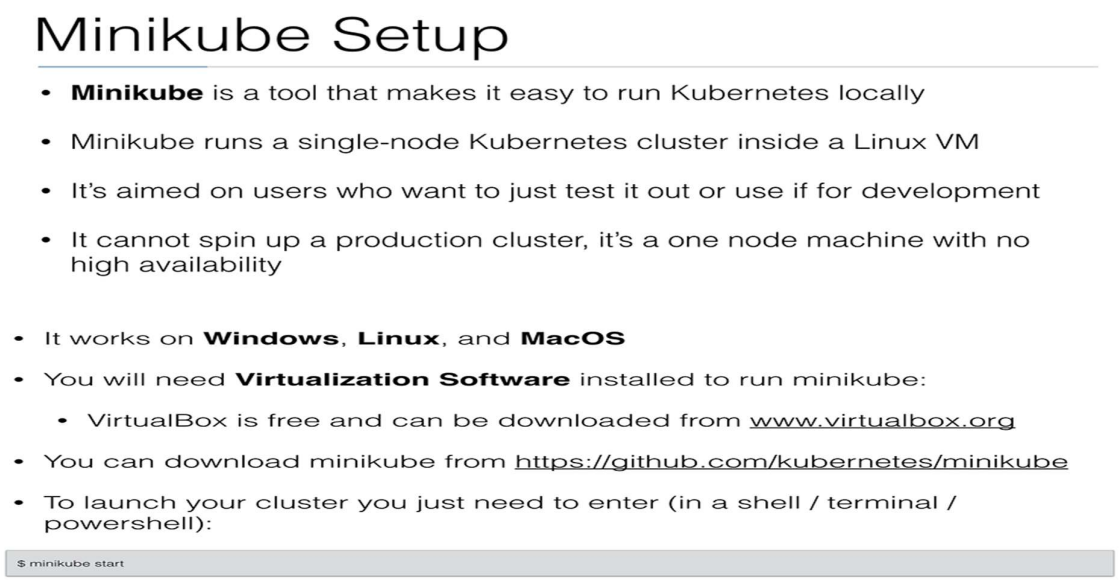
➔ Agar humane pod ko delete kar diya toh usase juda volume apane aap delete ho jayega.

➔ Kubernetes mein handling replica, rollout, self-healing ye feature nahi hote hai isko use karne kein liye humko alag sein **API** aur kuch alag sein **controler** add karne hote hai, uske baad hum iss feature ko kubernetes mein use kar sakte hai.

3 types of Kubernetes configuration are:-

1 = All in one single node installation (use only for practice)



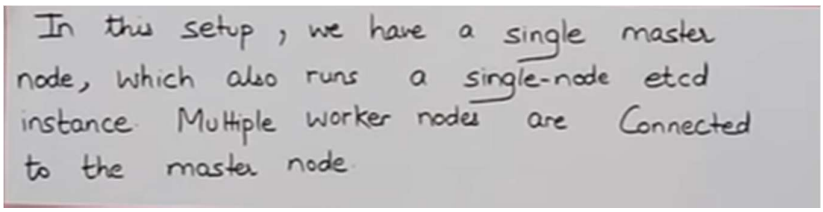


- Eak hi instance par master par bhi kaam kar raha hoga aur eak hi instance par worker bhi kaam kar raha hoga, as a practice karne kein liye ye bas badhiya hai as a production kaam nahi aayega ye

- Minikube practice karane kein liye theek hai production par kaam karane kein liye minikube better nahi hai

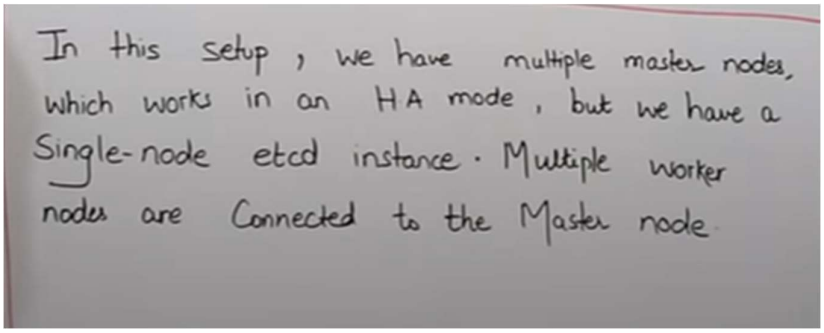
- Minikube mein master kewal eak ho sakta hai aur worker node hum kitane bhi bana sakte hai .

2 = Single – Node etcd, Single –Master and Multi-worker installation



- isme humara single etcd hoga matlab single database, single master hoga aur worker-node hum multiple bana sakte hai.
- For example eak master ka instance bana dunga aur 3 worker node ka instance bana dunga

3= Single - Node etcd ,Multi –Master and multi –worker node installation

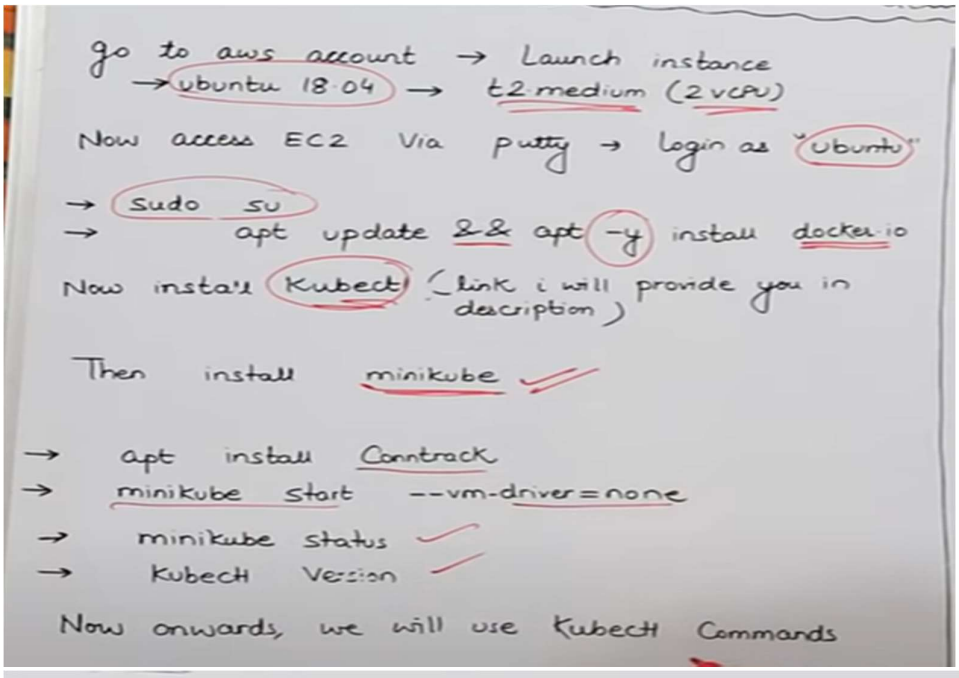


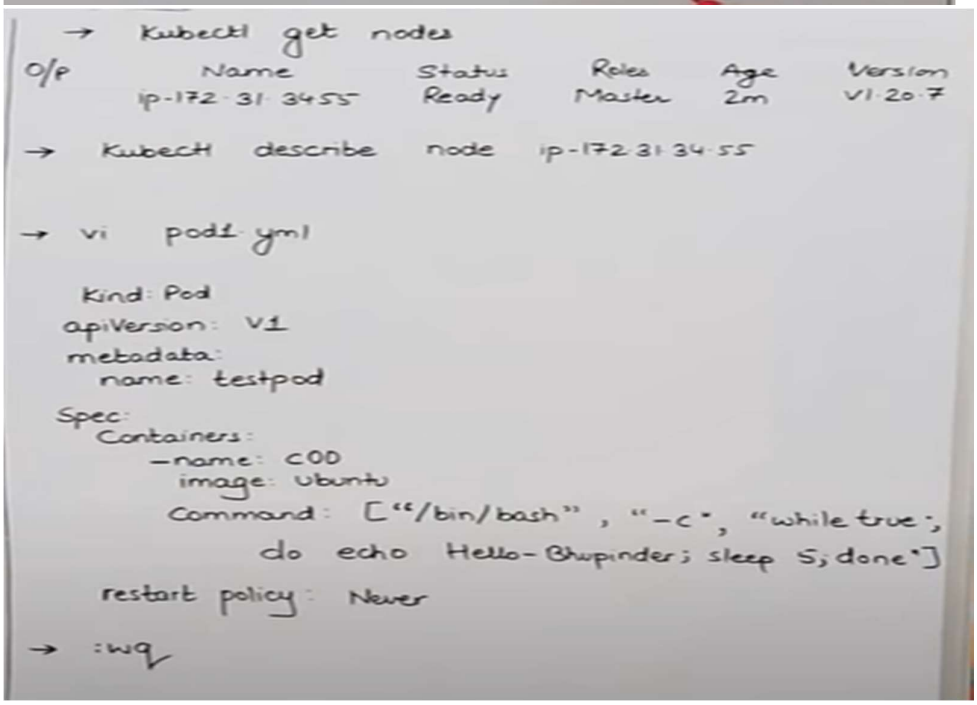
➔ Etcd single node hi rahega

➔ eak sein jada master rahenge iska fayada ye rahega ki agar koi eak master bhi koi fail huva toh uski jagah dusara master worker node ko sambhall laega

➔ Worker node multiple ho sakte hai

Install minikube in AWS account





➔ **Minikube install link :-**

curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube /usr/local/bin/

```
root@ip-172-31-12-153:/home/ubuntu# curl -Lo minikube https://storage.googleapis.com/minikube/releases/latest/minikube-linux-amd64 && chmod +x minikube && sudo mv minikube
/usr/local/bin/
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 66.3M  100 66.3M    0     0  68.4M      0 --:--:-- --:--:-- --:--:-- 68.3M
root@ip-172-31-12-153:/home/ubuntu#
```

(Is link sein hum minikube install kar denge hum dekhenge ki minikube jo install huva hai uski size 68.3 mb hai)

➔ **apt install Conntrack**

Minikube install karne kein baad contrack command chalana hoga otherwise humara minikube bhi kaam nahi karega kayade sein.

```
root@ip-172-31-12-153:/home/ubuntu# apt install conntrack
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  nftables
The following NEW packages will be installed:
  conntrack
0 upgraded, 1 newly installed, 0 to remove and 40 not upgraded.
Need to get 30.3 kB of archives.
After this operation, 104 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 conntrack amd64 1:1.4.5-2 [30.3 kB]
Fetched 30.3 kB in 4s (7688 B/s)
Selecting previously unselected package conntrack.
(Reading database ... 63764 files and directories currently installed.)
Preparing to unpack .../conntrack_1%3a1.4.5-2_amd64.deb ...
Unpacking conntrack (1:1.4.5-2) ...
Setting up conntrack (1:1.4.5-2) ...
Processing triggers for man-db (2.9.1-1) ...
root@ip-172-31-12-153:/home/ubuntu#
```

➔ **minikube start --vm-driver=none**

Iska matlab ki hum minikube ko start kar rahe hai aur abhi iske andar koi bhi vm driver nahi install kar rahe isliye hum driver ki value ko none kar rakhe hai .

Agar hum apane instance ko stop karke phir sein start kar rahe toh humko ye command phir sein run karni hogi

```
root@ip-172-31-12-153:/home/ubuntu# minikube start  --vm-driver=none
😄  minikube v1.24.0 on Ubuntu 20.04 (xen/amd64)
✨  Using the none driver based on user configuration
👍  Starting control plane node minikube in cluster minikube
🤹  Running on localhost (CPUs=2, Memory=3928MB, Disk=7876MB) ...
ℹ️  OS release is Ubuntu 20.04.3 LTS
🐳  Preparing Kubernetes v1.22.3 on Docker 20.10.7 ...
    ▪ kubelet.resolv-conf=/run/systemd/resolve/resolv.conf
    > kubeadm.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubectl.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubelet.sha256: 64 B / 64 B [--------------------------] 100.00% ? p/s 0s
    > kubectl: 44.73 MiB / 44.73 MiB [------------] 100.00% 72.74 MiB p/s 800ms
    > kubeadm: 43.71 MiB / 43.71 MiB [------------] 100.00% 69.17 MiB p/s 800ms
    > kubelet: 115.57 MiB / 115.57 MiB [-----------] 100.00% 68.08 MiB p/s 1.9s
    ▪ Generating certificates and keys ...
    ▪ Booting up control plane ...
    ▪ Configuring RBAC rules ...
🤹  Configuring local host environment ...

⚠️  The 'none' driver is designed for experts who need to integrate with an existing VM
💡  Most users should use the newer 'docker' driver instead, which does not require root!
📘  For more information, see: https://minikube.sigs.k8s.io/docs/reference/drivers/none/

❗  kubectl and minikube configuration will be stored in /root
❗  To use kubectl or minikube commands as your own user, you may need to relocate them. For example, to overwrite your own settings, run:

    ▪ sudo mv /root/.kube /root/.minikube $HOME
    ▪ sudo chown -R $USER $HOME/.kube $HOME/.minikube

💡  This can also be done automatically by setting the env var CHANGE_MINIKUBE_NONE_USER=true
🔎  Verifying Kubernetes components...
    ▪ Using image gcr.io/k8s-minikube/storage-provisioner:v5
🌟  Enabled addons: storage-provisioner, default-storageclass
🏄  Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
root@ip-172-31-12-153:/home/ubuntu#
```

Mera command line bhi configure ho gaya hai aur mera minikube cluster ready hogaya hai.

➔ **Minikube status**

Hum minikube ka status check kar sakte hai ki ye ready hai ya nahi, hum dekhenege ki humara ubectl bhi running hai humara apiserver bhi running hai .

```
root@ip-172-31-12-153:/home/ubuntu# minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured

root@ip-172-31-12-153:/home/ubuntu#
```

➔ Kubectl version

Humko kubectl check karna hoga ki kubectl humara run ho raha ya nahi kyuki aage jtani bhi command hai hum sabme kubectl use karenge.

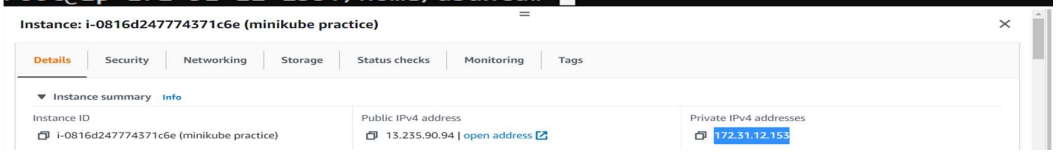
```
root@ip-172-31-12-153:/home/ubuntu# kubectl version
Client Version: version.Info{Major:"1", Minor:"23", GitVersion:"v1.23.0", GitCommit:"ab69524f795c42094a6630298ff53f3c3ebab7f4", GitTreeState:"clean", BuildDate:"2021-12-07T
18:16:20Z", GoVersion:"go1.17.3", Compiler:"gc", Platform:"linux/amd64"}
Server Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.3", GitCommit:"c92036820499fedefec0f847e2054d824aea6cd1", GitTreeState:"clean", BuildDate:"2021-10-27T
18:35:25Z", GoVersion:"go1.16.9", Compiler:"gc", Platform:"linux/amd64"}
root@ip-172-31-12-153:/home/ubuntu#
```

➔ Kubectl get nodes

Is command sein humko ye pata chal jayega ki humare pass kitane worker nodes hai joki master mein attached hai , hum jab ye command chalayenge toh usme roles mein ye bhi dekh sakte hai ki humari jo node hai vo kisase connect hai ,

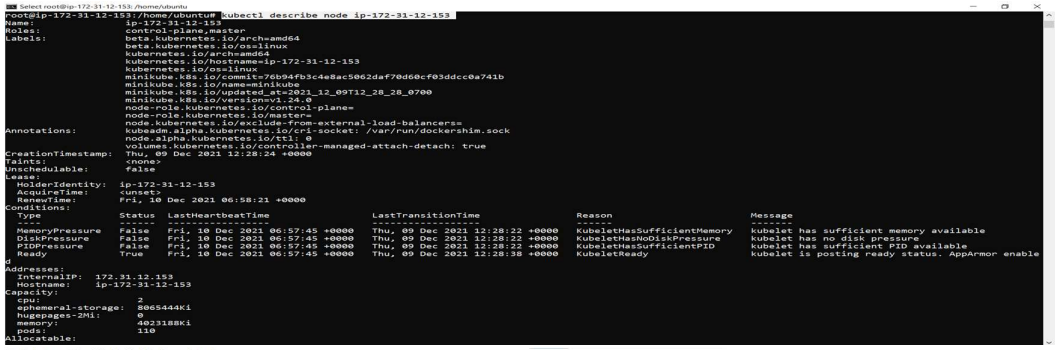
- uske andar eak ip hogi jo ki humare instance ki ip hogi.
- Status pata chalega ki vo ready hai ya nahi
- Mere instance ki private ip aur worker node ki private ip same hai kyu ki mein eak hi instance kein andar master node aur private node dono bana rakha hu.

```
root@ip-172-31-12-153:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES                  AGE   VERSION
ip-172-31-12-153    Ready    control-plane,master   18h   v1.22.3
root@ip-172-31-12-153:/home/ubuntu#
```



➔ kubectl describe node ip-172-31-12-153

- Agar mujhe node ki aur jankaari chahiye toh uske liye humko ye command chalani hogi mein node ki private ip daal kar us node ki full details le sakta hu.



```
root@ip-172-31-12-153:/home/ubuntu# kubectl describe node ip-172-31-12-153
Name:               ip-172-31-12-153
Roles:              control-plane,master
Labels:             beta.kubernetes.io/arch=amd64
                    beta.kubernetes.io/os=linux
                    kubernetes.io/arch=amd64
                    kubernetes.io/hostname=ip-172-31-12-153
                    kubernetes.io/os=linux
                    minikube.k8s.io/commit=76b94fb3c4e8ac5062daf70d60cf03ddcc0a741b
                    minikube.k8s.io/name=minikube
                    minikube.k8s.io/updated_at=2021_12_09T12_28_28_0700
                    minikube.k8s.io/version=v1.24.0
                    node-role.kubernetes.io/control-plane=
                    node-role.kubernetes.io/master=
                    node.kubernetes.io/exclude-from-external-load-balancers=
Annotations:        kubeadm.alpha.kubernetes.io/cri-socket: /var/run/dockershim.sock
                    node.alpha.kubernetes.io/ttl: 0
                    volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp:  Thu, 09 Dec 2021 12:28:24 +0000
Taints:             <none>
Unschedulable:      false
Lease:
  HolderIdentity:  ip-172-31-12-153
  AcquireTime:     <unset>
  RenewTime:       Fri, 10 Dec 2021 06:58:21 +0000
Conditions:
  Type             Status  LastHeartbeatTime                 LastTransitionTime                Reason                       Message
  ----             ------  -----------------                 ------------------                ------                       -------
  MemoryPressure   False   Fri, 10 Dec 2021 06:57:45 +0000   Thu, 09 Dec 2021 12:28:22 +0000   KubeletHasSufficientMemory   kubelet has sufficient memory available
  DiskPressure     False   Fri, 10 Dec 2021 06:57:45 +0000   Thu, 09 Dec 2021 12:28:22 +0000   KubeletHasNoDiskPressure     kubelet has no disk pressure
  PIDPressure      False   Fri, 10 Dec 2021 06:57:45 +0000   Thu, 09 Dec 2021 12:28:22 +0000   KubeletHasSufficientPID      kubelet has sufficient PID available
  Ready            True    Fri, 10 Dec 2021 06:57:45 +0000   Thu, 09 Dec 2021 12:28:38 +0000   KubeletReady                 kubelet is posting ready status. AppArmor enable
d
Addresses:
  InternalIP:  172.31.12.153
  Hostname:    ip-172-31-12-153
Capacity:
  cpu:                2
  ephemeral-storage:  8065444Ki
  hugepages-2Mi:      0
  memory:             4023188Ki
  pods:               110
Allocatable:
```

```
Allocatable:
  cpu:                2
  ephemeral-storage:  8065444Ki
  hugepages-2Mi:      0
  memory:             4023188Ki
  pods:               110
System Info:
  Machine ID:                 9f27305a4865481c940ab93923b2ad27
  System UUID:                ec2dc785-ae09-cddf-c46b-a8b33eb90d02
  Boot ID:                    c9b766bf-55ee-43a4-9831-96144418daed
  Kernel Version:             5.11.0-1020-aws
  OS Image:                   Ubuntu 20.04.3 LTS
  Operating System:           linux
  Architecture:               amd64
  Container Runtime Version:  docker://20.10.7
  Kubelet Version:            v1.22.3
  Kube-Proxy Version:         v1.22.3
PodCIDR:                      10.244.0.0/24
PodCIDRs:                     10.244.0.0/24
Non-terminated Pods:          (7 in total)
  Namespace                   Name                                       CPU Requests  CPU Limits  Memory Requests  Memory Limits  Age
  ---------                   ----                                       ------------  ----------  ---------------  -------------  ---
  kube-system                 coredns-78fcd69978-vcdqx                   100m (5%)     0 (0%)      70Mi (1%)        170Mi (4%)     18h
  kube-system                 etcd-ip-172-31-12-153                      100m (5%)     0 (0%)      100Mi (2%)       0 (0%)         18h
  kube-system                 kube-apiserver-ip-172-31-12-153            250m (12%)    0 (0%)      0 (0%)           0 (0%)         18h
  kube-system                 kube-controller-manager-ip-172-31-12-153   200m (10%)    0 (0%)      0 (0%)           0 (0%)         18h
  kube-system                 kube-proxy-bdhrg                           0 (0%)        0 (0%)      0 (0%)           0 (0%)         18h
  kube-system                 kube-scheduler-ip-172-31-12-153            100m (5%)     0 (0%)      0 (0%)           0 (0%)         18h
  kube-system                 storage-provisioner                        0 (0%)        0 (0%)      0 (0%)           0 (0%)         18h
Allocated resources:
  (Total limits may be over 100 percent, i.e., overcommitted.)
  Resource           Requests    Limits
  --------           --------    ------
  cpu                750m (37%)  0 (0%)
  memory             170Mi (4%)  170Mi (4%)
  ephemeral-storage  0 (0%)      0 (0%)
  hugepages-2Mi      0 (0%)      0 (0%)
Events:              <none>
root@ip-172-31-12-153:/home/ubuntu#
```

==========

➔ Now we create a manifest file

➔ vi pod1.yml

```
kind: Pod
apiVersion: v1
metadata:
  name: testpod
spec:
  containers:
    - name: c00
      image: ubuntu
      command: ["/bin/bash", "-c", "while true; do echo Hello-Satyam; sleep 5 ; done"]
  restartPolicy: Never      # Defaults to Always
```

➢ Hum yml file likhate samaye humko indentation ka bahut dhayan dena padta hai.

➢ **Kind : pod**

(iska matlab humara object pod hai hum pod create kar rahe)

➢ **apiVersion: v1**

humare pod ki api version mene de rakha hai version 1 jadatar hum likhate hai by default.

➢ **metadata:**

name: testpod

metadata mein agar hum koi naam dena chahahte hai apane pod ka toh hum uska naam de akte hai jaise ki mene apane pod ka naam testpod diya hai.

- **spec:**
  containers:
  - name: c00
  image: ubuntu
  command: ["/bin/bash", "-c", "while true; do echo Hello-Satyam; sleep 5 ; done"]

    - spec matlab specification aur specification kein andar mene container banaya hai

    - container kein andar mene eak group banaya hai jiske andar mene container kein baare mein bataya hai

    - mene container ko start karane sein pehale hi **– lagaya hai ,** aur agar mujhe eak pod kein andar multiple container banana hai toh hum **- laga kar eak naya group bana** kar naye container ki detail daal sakte hai .

    - **- lagane** sein ye clear ho jayega ki ab dusare container ko banana ki baat ho rahi hai , aise hum kitane bhi – laga kar container bana sakte hai

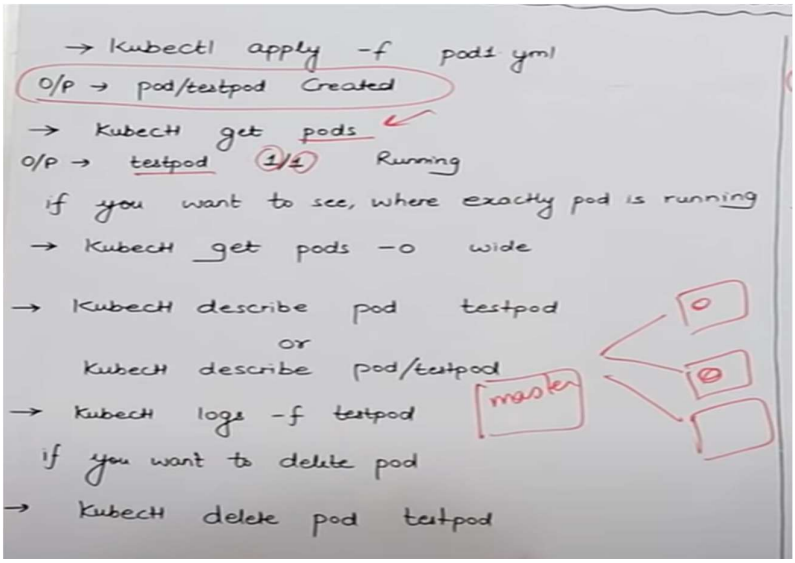
    - **- name: c00** mene apane container ka naam c00 diya hai

    - **command: ["/bin/bash", "-c", "while true; do echo Hello-Satyam; sleep 5 ; done"]**

      agar mera container chal jata hai matlab true hota hai toh hello satyam run kardena

    - uske baad hum wq likh kar file ko save kar denge.

==========



➔ kubectl apply –f pod1.yml

output :- pod/testpod created

(hum jaise hi kubectl vali command chalayenge vaise hi humari pod1.yml vali file chal jayegi aur run bhi ho jayegi, aur vo eak testpod naam sein pod create kar dega.)

```
root@ip-172-31-12-153:/home/ubuntu# kubectl apply -f pod1.yml
pod/testpod created
root@ip-172-31-12-153:/home/ubuntu#
```

➔ kubectl get pods

output :- testpod 1/1 running

- (humare pass node kein andar kitane pod hai vo dikh jayega, aur us pod kein andar kitane container hai vo bhi dikh jayega aur kitane container run ho rahe vo bhi dikh jayega.)
- Hum dekh sakte hai humare pass total eak hi container aur jo eak container humare pass hai vo run bhi ho raha, kyuki ststus running bata raha .

```
root@ip-172-31-12-153:/home/ubuntu# kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
testpod   1/1     Running   0          18m
root@ip-172-31-12-153:/home/ubuntu#
```

➔ kubectl get pods –o wide

(suppose humare pass eak master hai aur us master sein 3 worker node connected hai, humane eak pod banaya aur aur hum ye check karna chahate hai ki humare pod kaun sein worker node kein andar jaakar bana hai toh uske liye hum ye command use karenge. (wide sein humko humare pod ki exact detail pata chal jayegi. Ki vo kis instace par hai aur uss instance ki ip kya hai toh usase humko pata chal jayega ki vo kaun sein insatace par bana huva hai.)

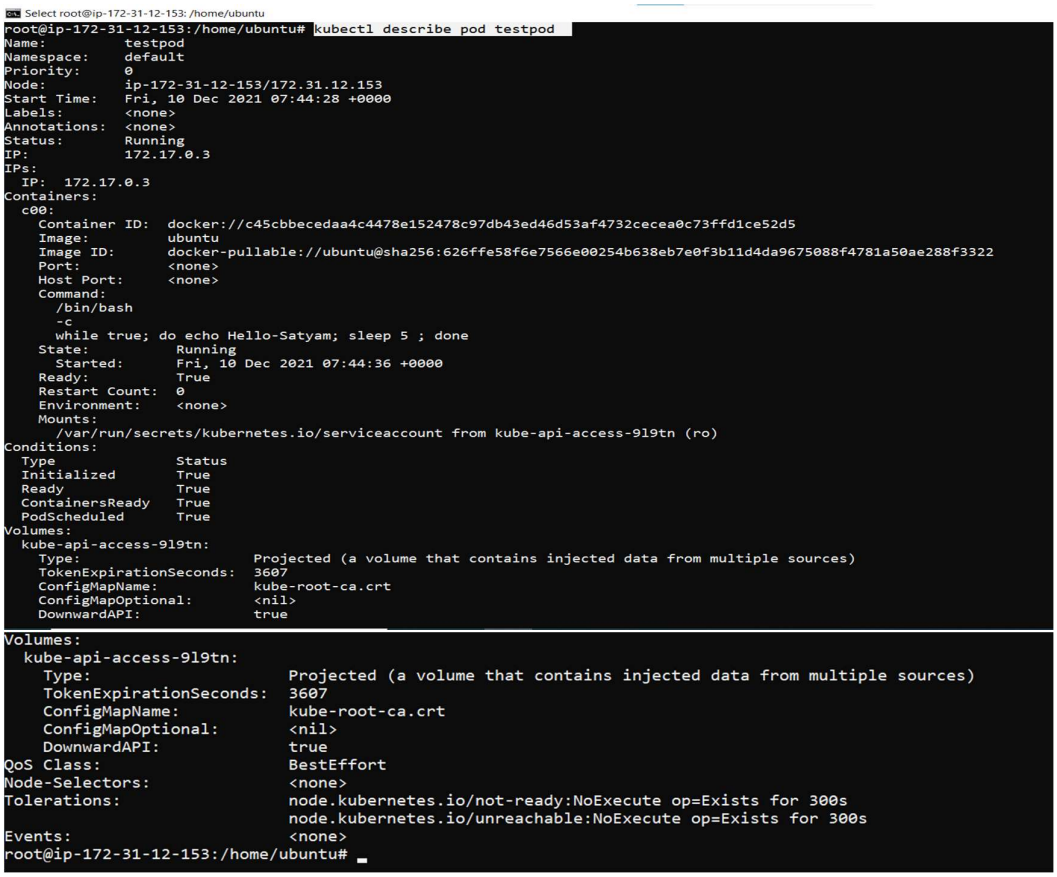
```
root@ip-172-31-12-153:/home/ubuntu# kubectl get pods -o wide
NAME      READY   STATUS    RESTARTS   AGE   IP           NODE               NOMINATED NODE   READINESS GATES
testpod   1/1     Running   0          36m   172.17.0.3   ip-172-31-12-153   <none>           <none>
root@ip-172-31-12-153:/home/ubuntu#
```

- isme 2 ip aa rahi hai toh isme sein jo node kein neeche vali ip diya gaya hai vo mere ec2 instance ki private ip hai aur jo dusari IP hai vo meri pod ki ip hai .
- aur ye dono ip private ip hoti hai , matlab aisa ki jo humare worker node kein instance ki private ip hai vahi mere pod kein prvate ip sein jaake baat karega
- aur humare instance ki private ip humare pod ko bhi eak private ip dedega jisase instance ki private ip aur pod ki private ip dono aapas mein baat kar sake .
- Kabhi bhi container ki ip nahi hoti ip hardum pod ki hoti hai .

➔ **kubectl describe pod testpod**

➔ **kubectl describe pod/testpod**

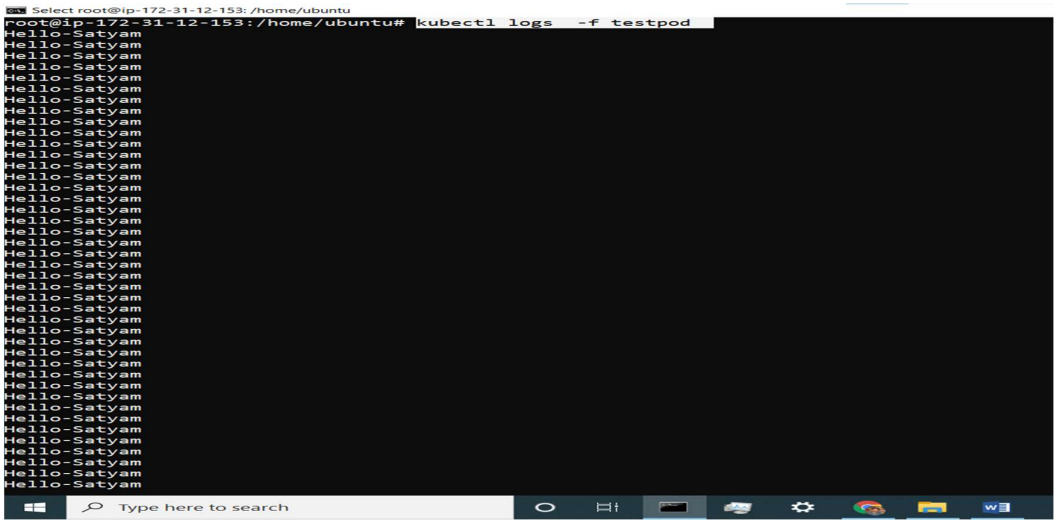
(agar hum apane pod sein related sab details dekhana chahahate hai ki kya kya pod mein huva hai kab kya kya cheej install huva, pehale kya install huva uske baad kya install huva ye sab cheeje humko is command sein pata chal jayega., dono mein sein koi bhi command use kar sakte hai same matlab hai .)



➔ **kubectl logs -f testpod**

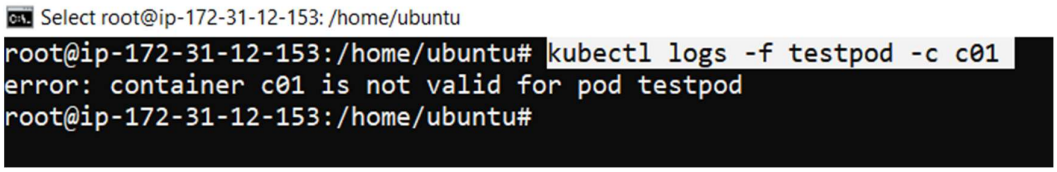
(is command sein humko ye pata chal jayega ki humare pod kein andar jo container hai usme kya chal raha, matlab ki container meien kya kya run ho raha hai aur ye bhi bata dega ki kaun kaun sa container running hai. Aur agar maanle ki mere pass multiple container hai aur mujhe kewal kisi eak container ki details chahiye ya phir uske logs dekhana ha toh hum ye command use karenge

Ctrl + z press karke hum bahar aa sakte hai



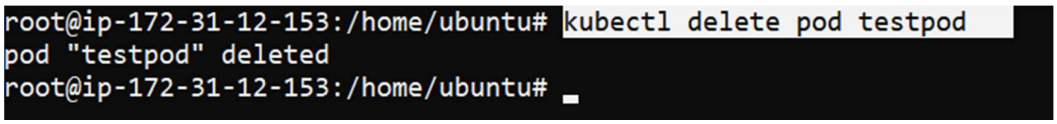
➔ kubectl logs -f testpod -c c01

- agar humare pass eak sein jaad container hai eak hi pod mein aur humko kisi eak particular container ki details ya log check karne ho toh uske liye hum ye command use karenge. Hum pod kein baad us container ka naam likh denge toh humko us container kein log details pata chal jayega
- mene niche pics mein container ka naam galat bataya hai container ka isliye details nahi aayi agar container ka naam c00 likhata toh **hello satyam** print kar deta kyu ki c00 naam sein jo container hai uske andar hello satyam hi details likh rakhi hai.

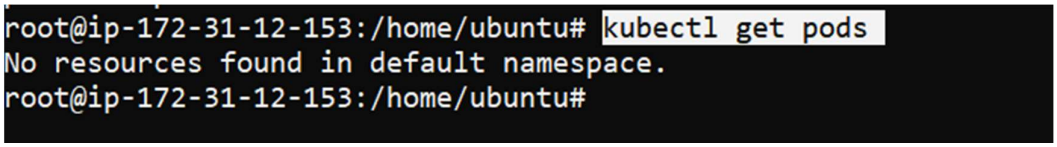


➔ **kubectl delete pod testpod**

(agar mujhe apane pod delete karna hai toh hum ye command use karenge, pod kein baad pod ka name)



Pod deleted



No pods available

➔ **kubelet delete –f pod1.yml**

(agar mujhe apani yml file hi delete karni ho jisase mene pod banaya thaa toh uske liye hum ye command use karenge)

==========

Annotation

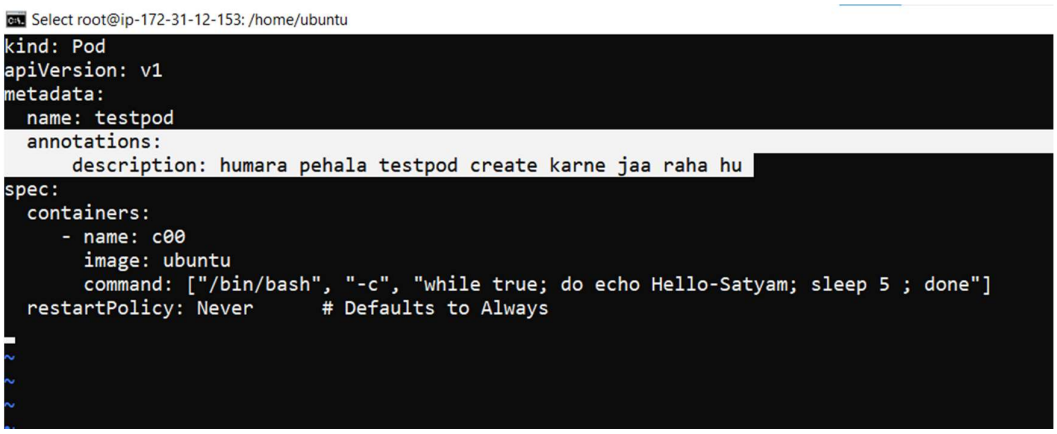
➔ **nano pod1.yml**

Annotations :-

(Annotation hum jaha metadata likhate hai apane yml file mein uske neeche annotations kein andar description kein anadar kuch bhi as a comment likh sakte hai agar kisi ko mein apani yml file de raha toh vo annotationpdh kar sakjh jayega us file kein baare mein.)

Annotations hota hai na ki annotation, **s** likhana jaruri hai

```
kind: Pod
apiVersion: v1
metadata:
  name: testpod
  annotations:
     description: humara pehala testpod create karne jaa raha hu
spec:
  containers:
    - name: c00
      image: ubuntu
      command: ["/bin/bash", "-c", "while true; do echo Hello-Satyam; sleep 5 ; done"]
  restartPolicy: Never      # Defaults to Always
```



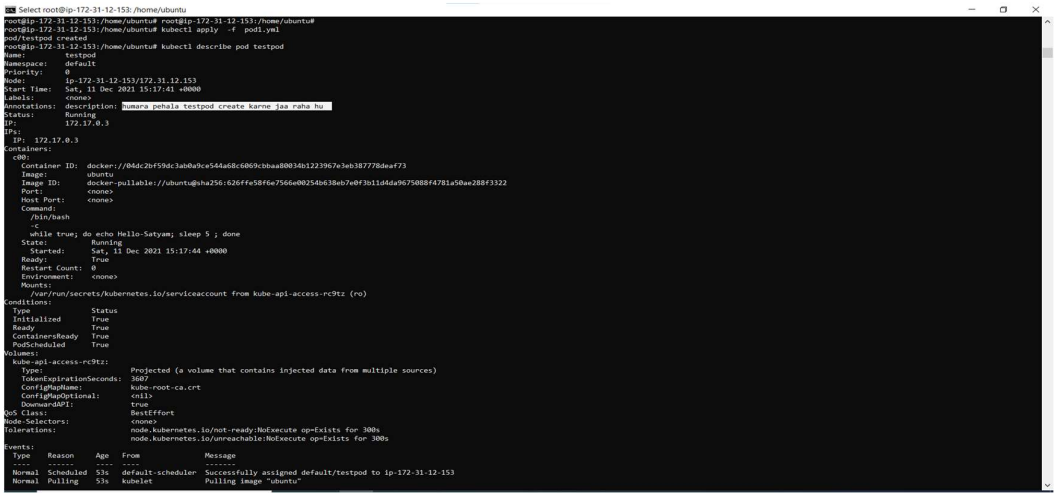
➔ **kubectl apply -f pod1.yml**

(hum jab bhi apane yml file mein kuch edit karenge toh us edit ko apply karne kein liye humko ye command chalani paegi.)

```
root@ip-172-31-12-153:/home/ubuntu# kubectl apply -f pod1.yml
pod/testpod created
root@ip-172-31-12-153:/home/ubuntu#
```

➔ kubectl describe pod testpod

(hum jab ye command chalayenge toh humare pod kein andar jo bhi annotation mene diya hoga vo mujhe pata chal jayega ki ye pod mene kisliye banaya thaa, ya toh phir iss pod kya kaam hai ab aage is pod ka use kaha hum karenge sab pata chal jayega annotation sein jo bhi hum apane yml file kein andr annotation kein andar likhe honge likhe honge iske liye humko yml file kholane ki jarurat nahi padegi.)



MULTI CONTAINER POD ENVIRONMENT

- **nano pod2.yml**

```
kind: Pod
apiVersion: v1
metadata:
  name: testpod3
spec:
  containers:
    - name: c00
      image: ubuntu
      command: ["/bin/bash", "-c", "while true; do echo Satyam - India; sleep 5 ; done"]
    - name: c01
      image: ubuntu
      command: ["/bin/bash", "-c", "while true; do echo Hello - india; sleep 5 ; done"]
```

- Hum 1 pod kein andar multiple container banayenge aur 1st container ka name mere **c00** hai aur 2nd container ka name **c01** hai .
- Hum yml mein containers kein andar parent mein – laga kar new container ki details daal ssakte hai .

- **kubectl apply -f pod2.yml**

```
root@ip-172-31-12-153:/home/ubuntu# kubectl apply -f pod2.yml
pod/testpod3 created
root@ip-172-31-12-153:/home/ubuntu#
```

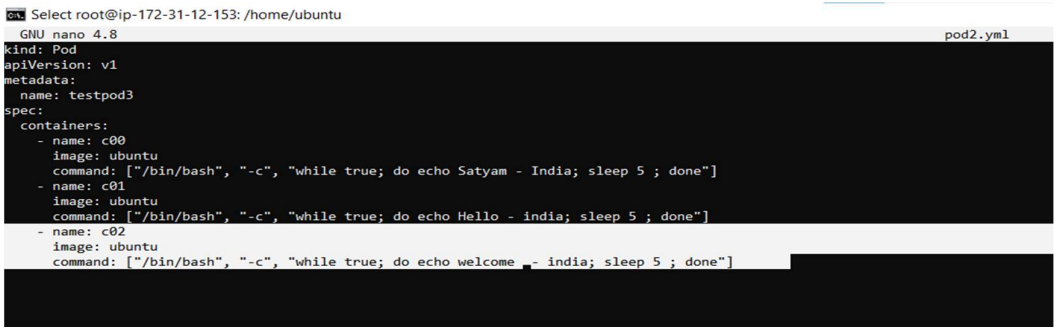
- **kubectl get pods**

```
root@ip-172-31-12-153:/home/ubuntu# kubectl get pods
NAME       READY   STATUS    RESTARTS   AGE
testpod    1/1     Running   0          42h
testpod3   2/2     Running   0          70s
root@ip-172-31-12-153:/home/ubuntu#
```

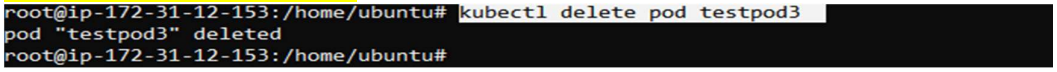
mene testpod 3 name sein pod banaya aur uske andar mene 2 container banaaye thee toh hum dekh sakte hai ki humare dono container running mein hai aur dono container idhar dikh rahe hai.

➔ Agar mujhe apane pod mein 2 ki jagah 3 container banana ho toh mein again usi yml file mein jaunga aur eak container aur bana dunga

For example :-



➢ **kubectl delete pod testpod3**



Hum pehale testpod3 name vale pod ko delete karenge jo pehale bana thaa, kyu ki running pod kein andar hum yml mein agar naya container bana rahe toh vo error dega toh naya container add karne kein liye humko pod delete karne padega uske baad apani pod2.yml vali file meiin container add karenge and save karke yml phir run karenge matlab apply karadenge.

➢ **kubectl apply -f pod2.yml**

➢ **kubectl get pods**

Hum dekhenge jab mene apani yml file mein edit karke new container banaya toh mere pod kein andar 3 running container hogaye hai ab .

➢ **kubectl describe pod testpod3**

```
Select root@ip-172-31-12-153: /home/ubuntu
    Container ID:  docker://6c2b8f62a18712020d8cd2df87853dd87c45d6bb3b22413d12909fa8e81140b8
    Image:         ubuntu
    Image ID:      docker-pullable://ubuntu@sha256:626ffe58f6e7566e00254b638eb7e0f3b11d4da9675088f4781a50ae288f3322
    Port:          <none>
    Host Port:     <none>
    Command:
      /bin/bash
      -c
      while true; do echo welcome  - india; sleep 5 ; done
    State:          Running
      Started:      Mon, 13 Dec 2021 10:40:10 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-s49j5 (ro)
Conditions:
  Type              Status
  Initialized       True
  Ready             True
  ContainersReady   True
  PodScheduled      True
Volumes:
  kube-api-access-s49j5:
    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  11m   default-scheduler  Successfully assigned default/testpod3 to ip-172-31-12-153
  Normal  Pulling    11m   kubelet            Pulling image "ubuntu"
  Normal  Pulled     11m   kubelet            Successfully pulled image "ubuntu" in 2.30772449s
  Normal  Created    11m   kubelet            Created container c00
  Normal  Started    11m   kubelet            Started container c00
  Normal  Pulling    11m   kubelet            Pulling image "ubuntu"
  Normal  Pulled     11m   kubelet            Successfully pulled image "ubuntu" in 2.322410142s
  Normal  Created    11m   kubelet            Created container c01
  Normal  Started    11m   kubelet            Started container c01
  Normal  Pulling    11m   kubelet            Pulling image "ubuntu"
  Normal  Pulled     11m   kubelet            Successfully pulled image "ubuntu" in 2.294017277s
  Normal  Created    11m   kubelet            Created container c02
  Normal  Started    11m   kubelet            Started container c02
root@ip-172-31-12-153:/home/ubuntu#
```

Hum events mein dekh sakte hai ki pehale kaise –kaise kya kya isane banaya hai matlab pehale kaun sa container phir uske baad kaun sa container.

- **kubectl logs -f testpod3 -c c00**

```
Select root@ip-172-31-12-153: /home/ubuntu
root@ip-172-31-12-153:/home/ubuntu# kubectl logs -f testpod3 -c c00
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
Satyam - India
```

Agar mujhe apane kisi containers kein log check karne hai toh uske liye hum ye command use karenge -c kein baad container name likh denge toh mujhe us container kein andar jo bhi logs honge vo sab dikh jayenge

- **Container 2 logs**

```
Select root@ip-172-31-12-153: /home/ubuntu
root@ip-172-31-12-153:/home/ubuntu# kubectl logs -f testpod3 -c c01
Hello - india
Hello - india
Hello - india
Hello - india
Hello - india
Hello - india
Hello - india
Hello - india
Hello - india
```

- **Container3 logs**

```
Select root@ip-172-31-12-153: /home/ubuntu
root@ip-172-31-12-153:/home/ubuntu# kubectl logs -f testpod3 -c c02
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
welcome - india
```

==========

- **kubectl exec testpod3 -c c00 -- hostname –i**

  agar mujhe ye pata karna ho ki mera container jis pod kein andar rakha hai us pod ki Ip kya hai toh hum ye command likh kar pata kar sakte hai, aur dusari baat ye hai ki kabhi bhi humare **container ki Ip nahi hoti hardum pod ki ip hoti hai** isliye mene neeche 3 command chalayi hai sab mein container name change hai but **sab mein same ip hai** aisa isliye kyuki jitane bhi container hai ye sab eak hi pod mein rakhe hai .

```
root@ip-172-31-12-153:/home/ubuntu# kubectl exec testpod3 -c c00 -- hostname -i
172.17.0.4
```

```
root@ip-172-31-12-153:/home/ubuntu# kubectl exec testpod3 -c c01 -- hostname -i
172.17.0.4
```

```
root@ip-172-31-12-153:/home/ubuntu# kubectl exec testpod3 -c c02 -- hostname -i
172.17.0.4
```

- **kubectl exec testpod3 -it -c c00 -- /bin/bash**

  > agar mujhe pod kein andar jaake kisi container kein andar kaam karna ho toh uske liye hum ye command use karenge.

  > (-it matlab iteractive mode) kehate hai, aur phir uske baad mene –c matlab container vale directory mein jao aur c00 naam ka jo container hai uske andar mujhe entry kara do

```
Select root@testpod3: /
root@ip-172-31-12-153:/home/ubuntu# kubectl exec testpod3 -it -c c00 -- /bin/bash
root@testpod3:/#
```

- **ps –ef**

  > Ab mujhe container kein andar ki details dekhani hai ki us container kein andar kaun sein logs chal rahe toh hum is command sein dekh sakte hai.

  > running container sein bahar ane kein liye hum exit command chalayenge .

```
root@testpod3:/# ps -ef
UID          PID    PPID  C STIME TTY          TIME CMD
root           1       0  0 Dec13 ?        00:00:05 /bin/bash -c while true; do echo Satyam - India; sleep 5 ; done
root       14410       0  0 06:39 pts/0    00:00:00 /bin/bash
root       14819       1  0 07:13 ?        00:00:00 sleep 5
root       14820   14410  0 07:13 pts/0    00:00:00 ps -ef
root@testpod3:/#
```

- **kubectl delete -f pod2.yml**

  > hum dekh sakte hai mene pod2.yml chalaya hai but humara testpod3 delete huva aisa isliye kyuki humane yml file mein pod aka naam testpod3 de rakha hai.

```
root@ip-172-31-12-153:/home/ubuntu# kubectl delete -f pod2.yml
pod "testpod3" deleted
root@ip-172-31-12-153:/home/ubuntu#
```

==========

Environmental variable in pod

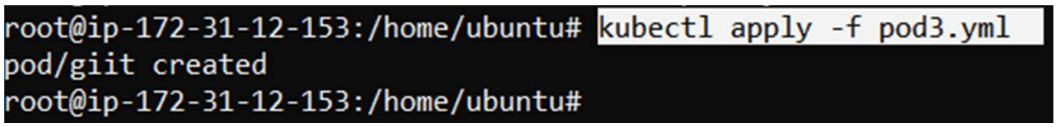
➔ **nano pod3.yml**

```
kind: Pod
apiVersion: v1
metadata:
  name: giit
spec:
  containers:
    - name: c00
      image: ubuntu
      command: ["/bin/bash", "-c", "while true; do echo Hello-kubernetes; sleep 5 ; done"]
      env:                    # List of environment variables to be used inside the pod
    - name: MYNAME
      value: Satyam
```

- Agar apne pod kein andar kuch predefined **key value** pair define karna chahate hai toh usko hum environment variable kein andar key pair value mein bata sakte hai.
- For example :- mene eak pod ka naam satyam de rakha hai key pair value mein aur jab-jab mein satyam kahi call kar raha hunga toh vo pod hi automatically call hoga , jis pod ki value mene satyam de rakhi hai.
- Agar env karke kuch bhi hum apane yml file mein likhnge toh iska matlab ye hai ki vo environment declaraition ki baat kar raha kisi cheej ka.
- Meri env key=MYNAME hai aur iski value=Satyam hai .
- Env variable ko search kaarne kein liye humko $ ka sign lagana padta hai.

➔ **Kubectl apply –f pod3.yml**

- Isko chalane sein mera pod create ho jayega mere container create ho jayega jo jo cheej ehumane yml mein likhi hogi sab start ho jayegi.



➔ **Kubectl get pods**

```
root@ip-172-31-12-153:/home/ubuntu# kubectl get pods
NAME      READY   STATUS    RESTARTS   AGE
giit      1/1     Running   0          39s
testpod   1/1     Running   0          2d19h
root@ip-172-31-12-153:/home/ubuntu#
```

➔ **kubectl exec giit -it -- /bin/bash**

```
root@ip-172-31-12-153:/home/ubuntu# kubectl exec giit -it -- /bin/bash
root@giit:/#
```

➔ **root@giit:/# env**

mujhe iss command sein dikh jayega ki mera kubernetes ka aport kaun sa hai jaise ki mera port hai 443 aur mujhe private ip bhi dikh rahi , baki environment sein related bahut cheeje dikh rahi hai .

➔ **echo $MYNAME**

Mene jaise hi apane key dalli key dalne kein baad mene jo us key ki ko value dali thii vo mujhe mil gayi.

Key hardum dalte samaye pehale $ lagate hai .

```
root@giit:/# echo $MYNAME
Satyam
root@giit:/#
```

➔ **kubectl delete -f pod3.yml**

```
root@ip-172-31-12-153:/home/ubuntu# kubectl delete -f pod3.yml
pod "giit" deleted
root@ip-172-31-12-153:/home/ubuntu#
```

==========

POD WITH PORTS

➔ Nano pod4.yml

Mene iss baar container image httpd choose kiya hai yml file mein, aur **httpd port 80** par chalta hai ., isliye mene container **port 80** bhi expose kar rakha hai .

```
root@ip-172-31-12-153: /home/ubuntu
  GNU nano 4.8
kind: Pod
apiVersion: v1
metadata:
  name: testpod4
spec:
  containers:
    - name: c00
      image: httpd
      ports:
       - containerPort: 80
```

kind: Pod
apiVersion: v1
metadata:
name: testpod4
spec:
containers:
- name: c00
image: httpd
ports:
- containerPort: 80

➔ kubectl apply -f pod4.yml

```
root@ip-172-31-12-153:/home/ubuntu# kubectl apply -f pod4.yml
pod/testpod4 created
root@ip-172-31-12-153:/home/ubuntu#
```

➔ kubectl get pods -o wide

```
root@ip-172-31-12-153:/home/ubuntu# kubectl get pods -o wide
NAME       READY   STATUS    RESTARTS   AGE     IP           NODE               NOMINATED NODE   READINESS GATES
testpod    1/1     Running   0          2d20h   172.17.0.3   ip-172-31-12-153   <none>           <none>
testpod4   1/1     Running   0          46m     172.17.0.4   ip-172-31-12-153   <none>           <none>
root@ip-172-31-12-153:/home/ubuntu#
```

Iss command sein humko pod ki aur instance ki dono ki ip dikh jayegi

➔ curl 172.17.0.4:80

- Mene apane pod ki ip kein dalane kein baad :80 likha hu dekhana chahata hu ki mera pod jo hai vo port 80 sein connect hai ya nahi
- Aur hum dekh sakte hai image mein output mein likh kar aa raha **IT works** ! iska matlab humara pod port 80 sein connect hai .
- Hum yml kein andar jis bhi port ko allow karenge vahi port jaakar connect hoga
- Curl command sein hum check karte hai ki humara pod connected hai ya nahi port sein .

```
root@ip-172-31-12-153:/home/ubuntu# curl 172.17.0.4:80
<html><body><h1>It works!</h1></body></html>
root@ip-172-31-12-153:/home/ubuntu#
```

==========