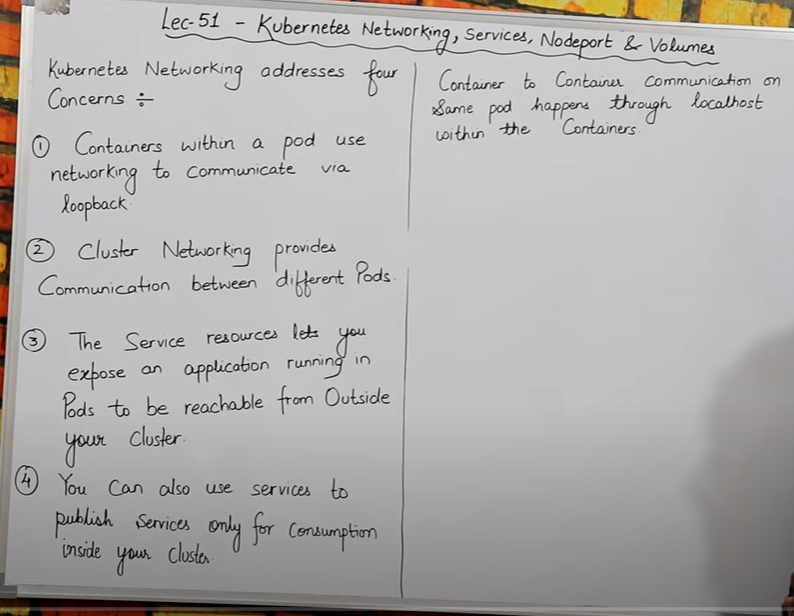
**Kubernetes-Networking, services, Nodeport & Volume**

****

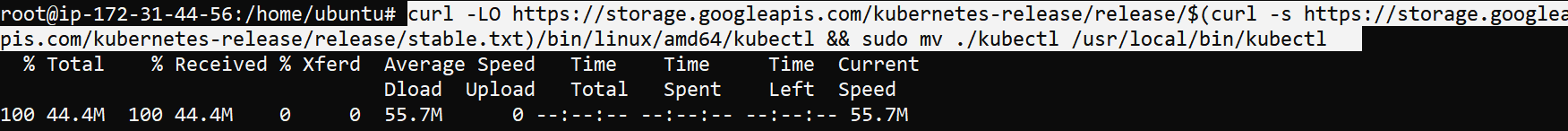
* Agar mere pass kisi eak pod kein andar multiple container hai toh vo container aapas mein eak dusare sein baat kar sakte hai, unhe kisi ki jarurat nahi ha vo simply **localhost** kein through eak dusare sein baat kar sakte hai,bas jis port ko humane allow kar rakha hai vo port number humko dena hoga **for e.x. = localhost:80**
* Humara pass 2 node A & B , JISME sein **node A** mein mere pod kein andar container hai vo chahahta hai ki **node B** kein andar jo pod bana huva hai usase baat kar sake toh ye possible hota hai eak container dusare node kein container sein baat kar sakta hai .
* sudo sus

**command to install docker is:-**

* sudo apt update && apt -y install docker.io

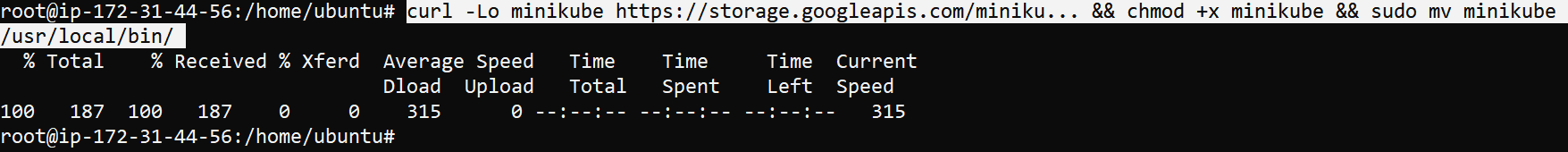
**install Kubectl now with the given link:-**

curl -LO https://storage.googleapis.com/kubernetes-release/release/$(curl -s https://storage.googleapis.com/kubernetes-release/release/stable.txt)/bin/linux/amd64/kubectl && sudo mv ./kubectl /usr/local/bin/kubectl

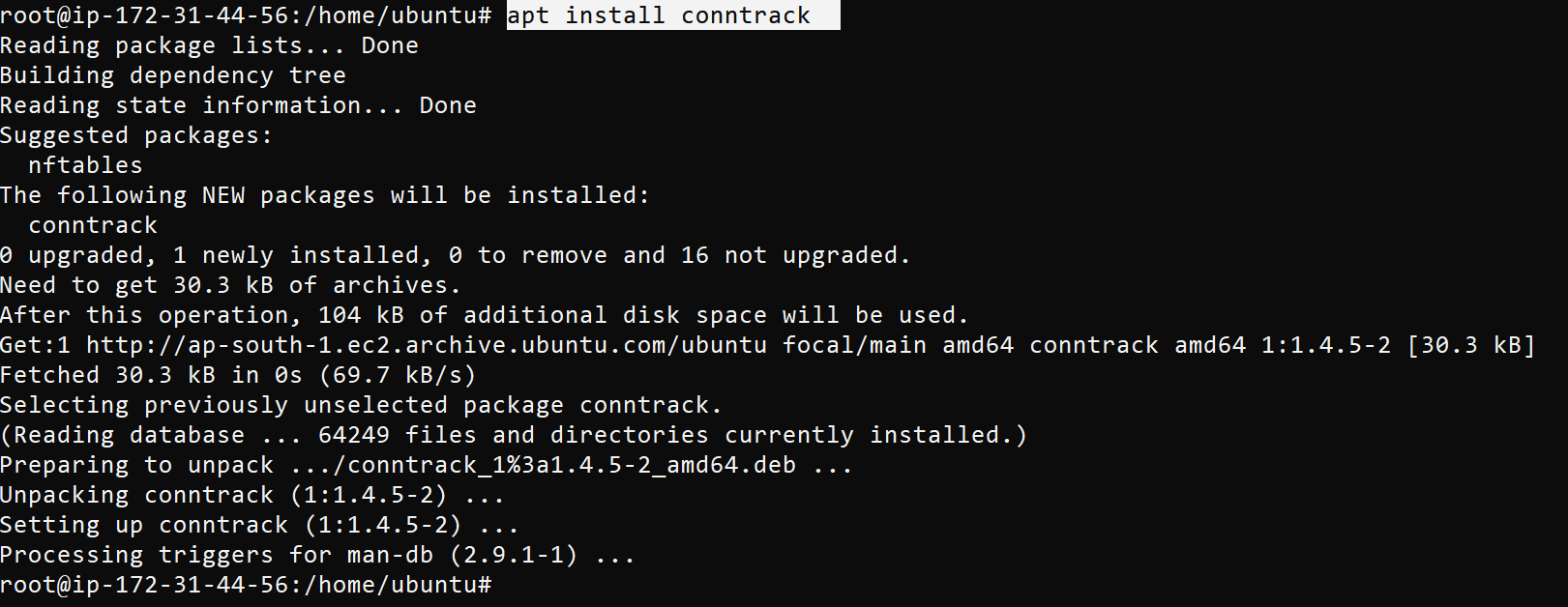


**install Minikube with the given link:-**

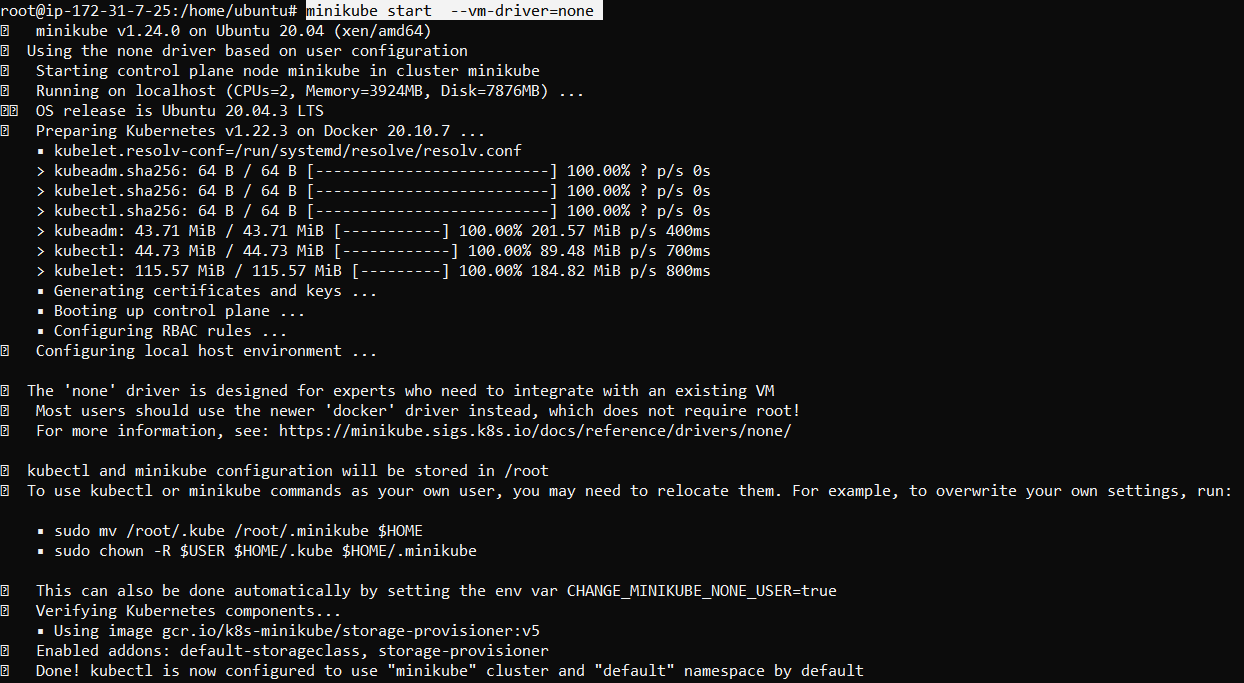
curl -Lo minikube https://storage.googleapis.com/miniku... && chmod +x minikube && sudo mv minikube /usr/local/bin/



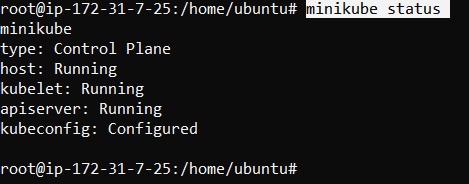
* **apt install conntrack**



* minikube start --vm-driver=none



* minikube status



kubectl version

kubectl get nodes

KUBERNETES NETWORKING **(samepod container communication)**

=======================================================

* nano pod1.yml

kind: Pod

apiVersion: v1

metadata:

name: testpod

spec:

containers:

- name: c00

image: ubuntu

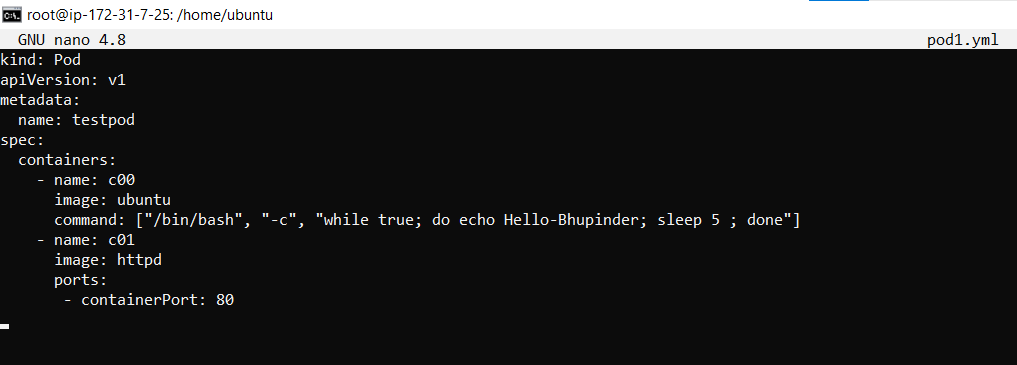
command: ["/bin/bash", "-c", "while true; do echo Hello-Bhupinder; sleep 5 ; done"]

- name: c01

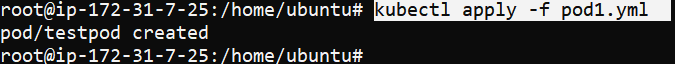
image: httpd

ports:

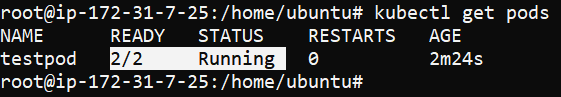
- containerPort: 80



* kubectl apply-f pod1.yml



* kubectl get pods

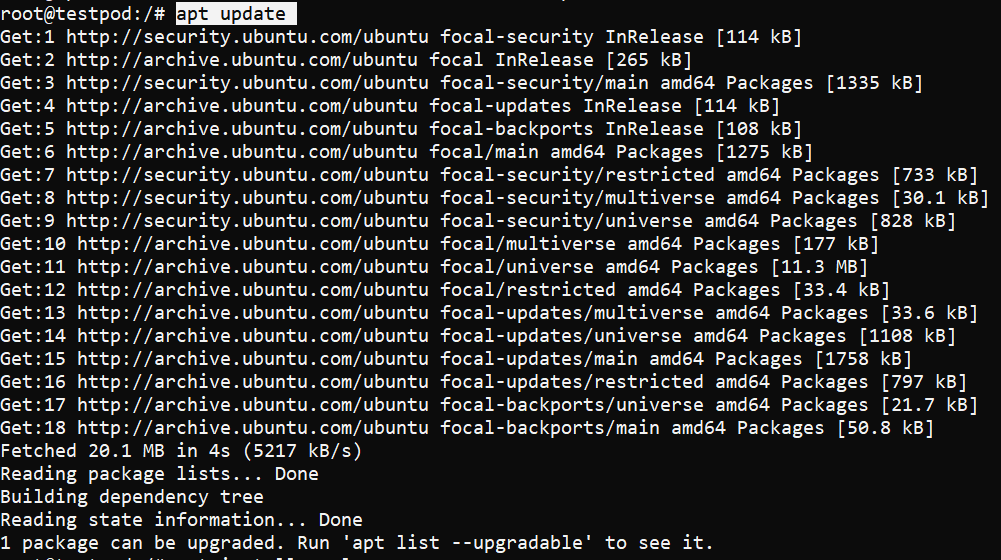


* kubectl exec testpod -it -c c00 -- /bin/bash

****

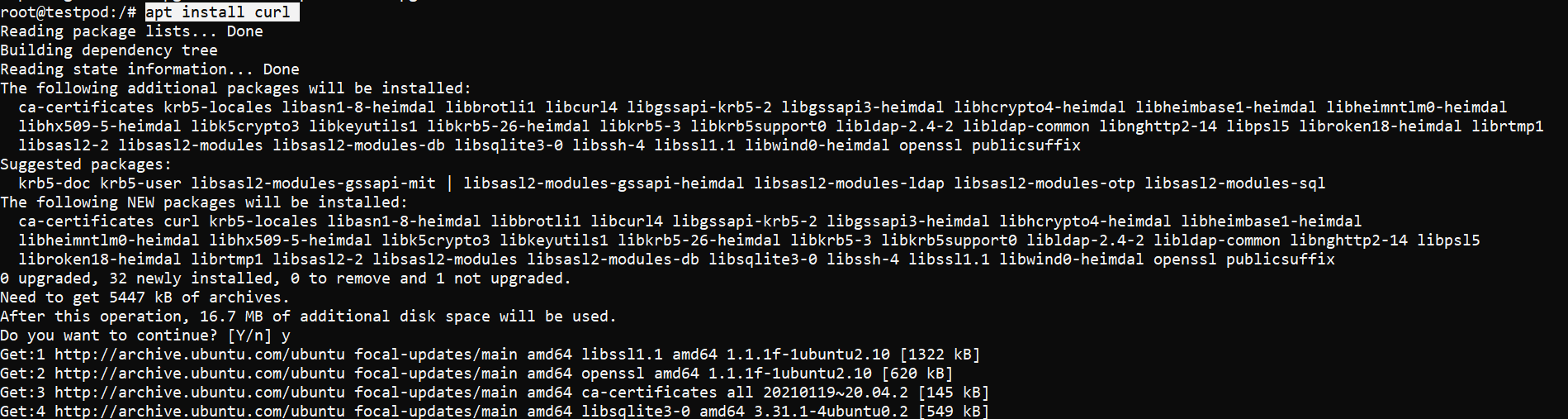
Iss command sein mene jo testpod naam sein **pod** create kiya thaa uske andar ghus jaunga . aur mene **c00** naam sein apane **container** ka name define kiya hai toh mein apane container kein andar bhi ghus jaunga .

* **apt update**



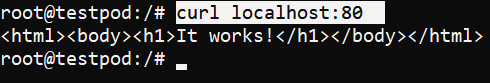
mene apane container ko apt update kiya ,

* apt install curl



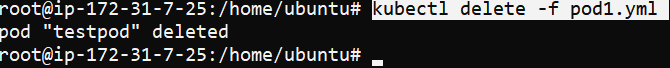
mein karna ye chahata hu ki mera eak container dusare pod kein container sein baat kar sake. Isliye mujhe apane container kein andar curl ko install karna hoga.

* curl localhost:80



Agar localhost:80 chalane sein output mein **it works** likh kar aa raha iska matlab hum apane dusare container sein baat kar sakte hai same pod kein andar humara jo humara dusara container hai hum usase aapas mein baat kar sakte hai aur , jo pod humare localhost:80 sein connect hai vo pod aapas mein baat kar sakte hai .

* Kubectl delete –f pod1.yml

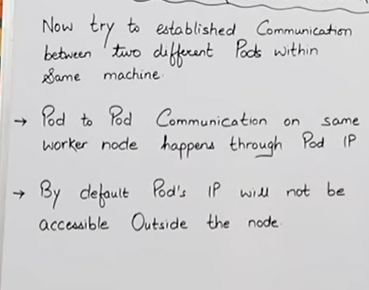


**Different pod communication :-**

* **(In same node different pod container communicate to each other .**

Same node kein andar multiple pod bane hai aur multiple pod kein andar jo container hai un container ko hum eak dusare sein communicate karana chahate hai.)

* Node kein bahar sein hum pod ki ip by default nahi jaan sakte.
* Same node kein andar Pod sein pod kein container ko agar aapas mein baat karana ho toh hum pod ki IP use karke container sein communicate kar sakte hai .

****

* nano pod1.yml

kind: Pod

apiVersion: v1

metadata:

name: testpod1

spec:

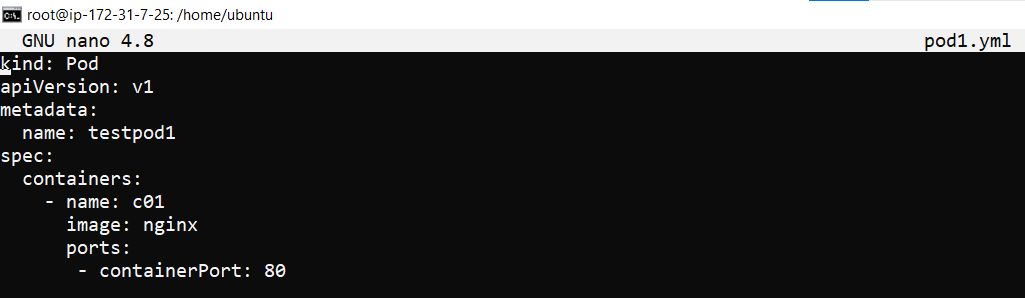
containers:

- name: c01

image: nginx

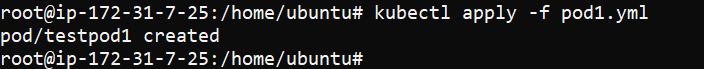
ports:

- containerPort: 80



Mene eak pod banaya jiska naam **testpod1** hai aur us pod kein andar mene 1 ubuntu operating system kein naam sein container banaya hai .

* kubectl apply -f pod1.yml



* nano pod2.yml

kind: Pod

apiVersion: v1

metadata:

name: testpod2

spec:

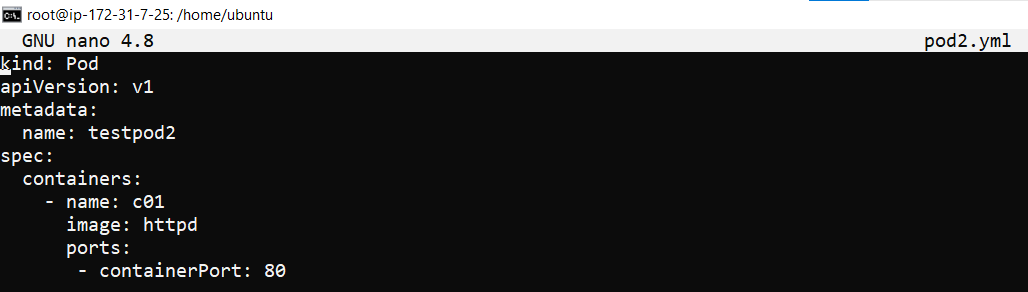
containers:

- name: c01

image: httpd

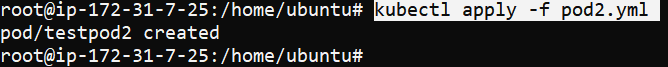
ports:

- containerPort: 80

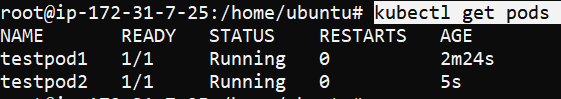


Mene yaha same naya pod banaya hai jiske name mene testpod2 banaya hai aur iske andar mene image ka naam **httpd** rakha hai aur iske andar bhi mene eak container bana rakhe hai, aur container ka naam  **co2** hai isme.

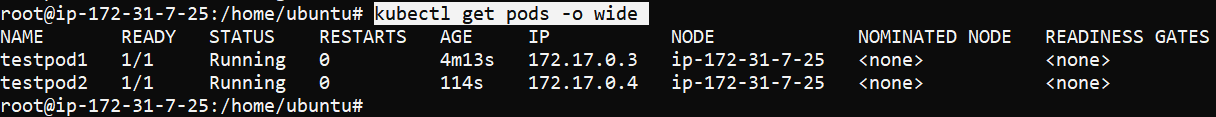
* kubectl apply -f pod2.yml



* Kubectl get pods

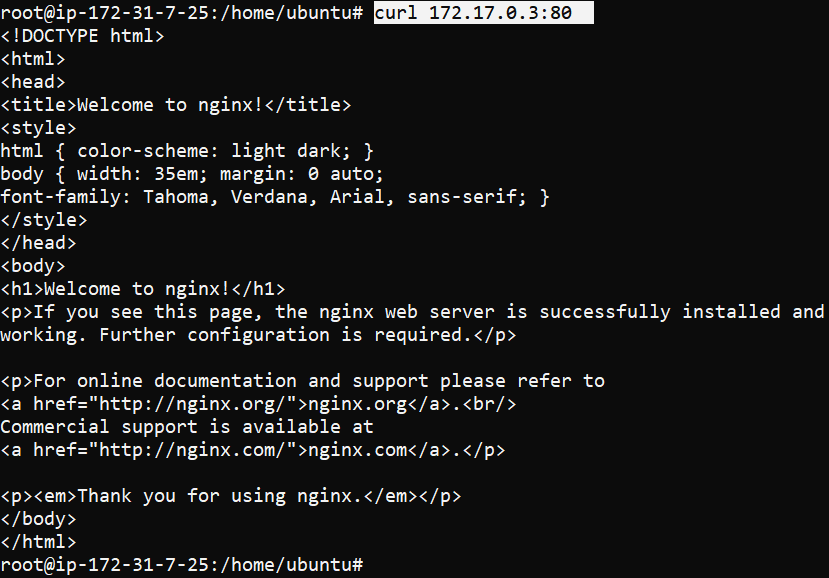


* kubectl get pods -o wide



Humare 2 pod running hai aur 2 kein 2 no pod ki ip mere pass hai , hum jis bhi pod ko chahe access kar sakte hai .

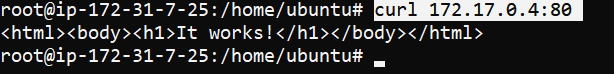
* curl 172.17.0.3:80
* (ye mene testpod 1 ki ip dali hai curl kein aage aur usko port 80 sein connect kar diya hai, isase humari image mein jo container mene banaya hai usme jo bhi chal raha hai vo sab dikh jayega, aur mene us container kein andar nginx image daal rakhi hai , aur output mein mere nginx likh kar aaraha matlab sab sah hai ).



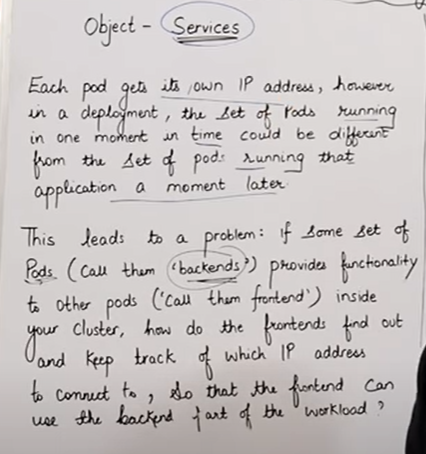
Mene **testpod1** ki ip dali hai aur ip dalne kein baad mene **:80** lagaya hai matlab port define kiya hai ki agar mera port 80 hai toh aap isase connect kar do .

Mene nginx image choose kiya thaa iss wajah sein jab mene apane port ko sconnect kiya toh mujhe ye pata chal gaya .

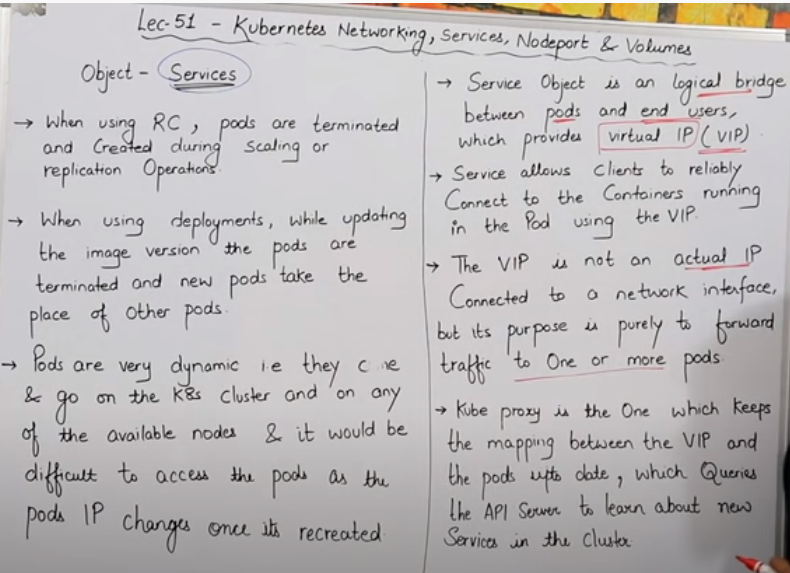
* curl 172.17.0.4:80

(ye mene testpod 2 ki ip dali hai curl kein aage aur usko port **80** sein connect kar diya hai, isase humari image mein jo container mene banaya hai usme jo bhi chal raha hai vo sab dikh jayega, aur mene us container kein andar httpd image daal rakhi hai, aur isme **it works** likh kar aa raha hai, matlab sab sahi hai ).

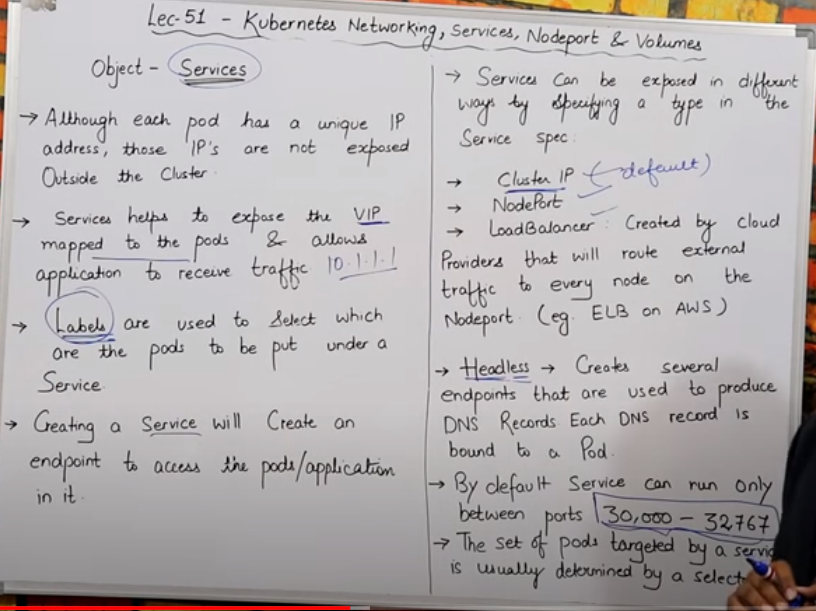
**Services**



* Jab hum **replication control** use karte hai toh us case mein hardum new pod create bhi hote hai aur delete bhi hote hai toh iski wajah sein humare pod ki **ip** bhi change hoti rehati hai baar-baar jiski wajah sein aage humko pod ko access karne kein liye baar baar pod ki **ip** dalni padti hai . agar mere pass 500 pod hai aur vo fal hokar phir apane aap create ho rahe toh har baar pod ki ip change hoti jayegi , to hum agar baar baar uski ip note karke batayaenge toh vo achii baat nahi hai, isliye hum uske. Liye service use karte hai.



* **service** kein help sein hum eak **private ip** define kar dete hai , jo ki kewai humare replica set jo define kiya hai usase baat kar payega . isase jab bhi replica set ki wajah sein jab bhi koi naya pod create ya start hoga toh service ko apane aap replica set kein through **ip** mil jayega sabhi pod ki jiski wajah sein humko alag sein ip likhane ki jarurat nahi hai . hm kewal service ki ip dalenge aur usase hum apane cluster kein sab pod kein ip ko acess kar sakte bhalle vo chahe kitani bhi baar new pod create kare ya fail ho jaye.
* **Services** humko eak bridge provide karata hai jisko hum logical bridge bhi kehate hai aur yein bridge humare pod aur user kein beech mein eak end-user provide karta hai, jiski help sein user, pod kein data ko access kar sakte hai , service humko eak **private virtual ip** de deta hai jisko change nahi kiya jaa sakta hai. Service mujhe jo ip deta hai us ip keein help sein hum kisi bhi node kein pod ki ip bina jane access kaar sakte hai, bas vo node usi service kein andar aana chahihye .
* Kubeproxy hi humari help karega humare **virtual ip** aur pod ki ip ko update rakhane mein, kube proxy hi kaam karega **for example**:- jaise mera koi pod fail hogaya aur recreate ho raha hai mera koi pod toh uski ip change hogi toh kube proxy k help sein hum humare **private virtual ip** ko connect karenge humare pod kein new ip sein. aur humako **private virtual ip** kaha sein millegi humko **service** dega private virtual ip .
* Service jab hum use karte hai toh humko eak **ip** dedeti ha isase hu apane pod ko access kar sakte hai bhale humara pod kitani bhi baar fail ho a create ho.
* Agar humane kuch image mein change kiya hai aur phir usko update kiya hai toh phir naya updated pod create ho jayega uski jagah aur uski bhi ip bhi change ho jayegi



* Har eak **pod** ko eak **ip** milli huyi hai but hum us pod ko cluster kein bahar sein access nahi kar sakte, hum internally cluster kein andar eak dusare pod ko access kar sakte hai ,but agar cluster kein bahar sein hum apane pod ko access karna chahate hai toh uske liye humko alag si kuch cheeje use karni hogi pod ko cluster sein bahar sein access karne kein liye .
* Service ki help sein humko jo **private virtual ip** milati hai aur us ip ko hum jo pod kein sath map karte hai , aur service expose karta hai pod ko ki jo bahar sein traffic aa raha humare pod mein chal rahe application par vo aaram sein aasake .
* Humko **service** use karte samaye label lagana chahiye isase humko pata rahega ki kis-kis pod par service lagani hai . humko **service** mein bhi label define karna hoga aur hum yml file kein andar bhi same label define karna hoga, jisase hum pehachan sake ki kis pod par label lagana hai kis par nahi.
* Jab bhi hum service use karenge toh eak endpoint create hoga aur us **endpoint** kein through hum apane pod ya apane application ko access kar payeneg.
* There are four types of service are:
* Cluster ip
* Nodeport
* Loadbalancer
* Headless
* Cluster ip jab bhi hum service ka use karta hai by default cluster ip ko hi leta hai vo
* But agar hum node port define karte hai th vo nodeport par bhi kaam karega.
* Hum jab bhi bahar kein through ya internet kein through apane pod ko access karenge toh humko eak port number milata hai service ke through toh by default vo port iske andar hi hota hai 30,000-32767.iska matlab aap jab bhi bhr sein pod k access karna chah rahe toh bhai iss port number kein through humare pod ko access karna.
* LABEL aur SELECTOR dono imp cheej hai service kein andar aur yml kein andar batana jisase hum target kar sake ki humko kis-kis pod par service lagani hai.