Step 1: Create a new multipass ubuntu server

```
multipass launch 22.04 --name k3s-master --cpus 2 --memory 2G --disk 20G
```

Step 2: Install k3s on vmware

First ssh into it

```
multipass shell k3s-master
```

Then: curl -sfL https://get.k3s.io | sh -

This installs:

Kubernetes API
containerd runtime
Traefik Ingress Controller
Flannel networking

2.1: Check if everything is running

```
sudo kubectl get nodes
sudo kubectl get pods -A
```

Step 3: Create a namespace for app

Inside VM:

```
sudo kubectl create namespace practise
```

Why?: to keep your practise app isolated, to match real-work kubernetes structure.

Step 4: Optional : Change kubernetes context on your macmini, Why?

So that you can do kubectl from your macOs terminal, no headache of opening docker desktop and eating your 4Gb ram out of nowhere, and multipass has native support for ARM based macs.

4.1:Get Kubeconfig from new Virtual Machine that we just created

a. ssh into the new vmware : multipass shell new vmware(for us its k3s-master): multipass shell k3s-master

copy the kubeconfig of that vmware : sudo cat /etc/rancher/k3s/k3s.yaml , you will see full kubeconfig file

still inside vm copy config to mac: sudo cat /etc/rancher/k3s/k3s.yaml: copy all of it from start to end

now on your mac:

```
mkdir -p ~/.kube
nano ~/.kube/config-kpractise
```

paste the content inside and save and exit!**Important step**

Inside you mac after you copy whole file

change ip into your virtual map ip so that macos can reach to it

On your mac:

export KUBECONFIG=~/.kube/config-kpractise : kpractise is name of vmware k3s-master if you made it k3s-master

cat ~/.kube/config –if you want to see or delete kubeconfig files that was configured before on the system in kube/config/ before

Optional

If you want to configure vs code to permanently use same kubeconfig as your main terminal for kubectl commands use the code:

```
echo 'export KUBECONFIG=/Users/sandesh/.kube/config-kpractise' >> ~/.zshrc
```

ie: kube/config-kpractise was the location in my mac so it differs from person to person but just the location of config files.

```
source ~/.zshrc
```

Now VSCode terminals will ALWAYS use the same kubeconfig as your main terminal.
Congrats now you can kubectl from your mac lol!

The Deployment part now:
This is specific to my monolith application everyone has
different: This part is optional but teaches how to connect to cloudflare tunnel and tunnel traffic from your main domain to your home kubernetes cluster

Step 1: Create a namespace on cluster
kubectl create namespace explorage/name of your cluster
kubectl get ns –check if it worked**Deployment part**
example explorage as namespace
1. Apply the secrets: Considering you have already created the namespaces
kubectl apply -f secret.yaml -n explorage
2. Apply the deployments:
kubectl apply -f deployment.yaml -n explorage
3.Apply the service
kubectl apply -f service.yaml -n explorage
4. Apply ingress
kubectl apply -f ingress.yaml -n explorage
verify: kubectl get ingress -n explorage

if you wanna edit local ingress connection on mac
sudo nano /etc/hosts
change or put the ip of virtual machine/and ingress eg:
192.168.2.16 practise.explorage
After this you should be able to view it in your macos browser
Note that we have k3s cluster running on our mac's ubuntu server which we made with help of multipass, gives us native performance.

Cloudflare Tunnel

1. Install cloudflare on your multipass Vmware

```
curl -fsSL
https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-linux-amd64.deb
```

if arm chips
cloudflared.deb
curl -fsSL

```
sudo dpkg -i cloudflared.deb
```

check version: cloudflared --version

2. Login into cloudfare: cloudflared tunnel login: it asks for a domain / this documentation is strictly based on having a domain/ cloudfare nomore gives free domains with their subnames.

3. Create a cloudfare tunnel:

cloudflared tunnel create practise-explorage-tunnel –name can be any tunnel

4. Create a directory and configure cloudfare tunnel

```
sudo mkdir -p /etc/cloudflared
sudo nano /etc/cloudflared/config.yml
```

fill the etc/dloudflared/config.yml
Helpful link: official tunnel documentation cloudfare

you just need to fill up the config file: this is really important see your kubernetes main port is running on not targetPort but actualport on your service: suppose i have nodeport so ,not the port of service and not port of target, actually nodeport .

5. Install system service

```
sudo cloudflared service install
```

Enable on startup

```
sudo systemctl enable cloudflared
```

start

sudo systemctl start cloudflared**watchlive**
sudo journalctl -u cloudflared -f

if tunnel is not there on cloudfare dashboard don't worry i've got your back
just make sure that you know that cloudfare exposes each tunnel as
<TUNNEL-ID>
.cfargotunnel.com
Go to your main vm or ubuntu and then type: cloudflared tunnel list and expose the like if id is
1234
1234.cfargotunnel.com
Go to dashboard and add a record
and do write /type:
TYPE: CNAME
NAME: hello123 –just this no dot com cause if you bough domain as joe.co.uk it becomes
hello123.joe.co.uk
TARGET: 1234.cfargotunnel.com –or whatever you configured it with