```
**********************************************************************
STEP:1 AWS UBUNTU INSTANCE
**********************************************************************

FOLLOW BELOW STEPS TO PREPARE UBUNTU INSTANCE
1. PREPARE UBUNTU SERVER 22.04 LTS(HVM) SSD VOLUME TYPE INSTANCE
2. CUSTOM PROTOCOL FOR PORT : 8080
3. DOWNLOAD .PEM KEY TO DESIRE DIRECTORY
4. CONNECT USING MOBA X-TERM

**********************************************************************
STEP: DOCKER INSTALLATION
**********************************************************************
>sudo apt-get update
>sudo apt-get install ca-certificates curl gnupg lsb-release

>sudo mkdir -p /etc/apt/keyrings
>curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

>echo "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null

>sudo apt-get update
>sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-
plugin

------TO VERIFY THE INSTALLTION------

> sudo docker -v
      output: Docker version 20.10.18, build b40c2f6


**********************************************************************
STEP:3 KUBERNATE INSTALLTION
**********************************************************************

GOTO> GOOGLE> SERACH FOR KUBERNATE INSTALLTION ON UBUNTU

LINK:https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/
install-kubeadm/

> sudo apt-get update
> sudo apt-get install -y apt-transport-https ca-certificates curl
> sudo curl -fsSLo /usr/share/keyrings/kubernetes-archive-keyring.gpg
https://packages.cloud.google.com/apt/doc/apt-key.gpg
> echo "deb [signed-by=/usr/share/keyrings/kubernetes-archive-keyring.gpg]
https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee
/etc/apt/sources.list.d/kubernetes.list

>sudo apt-get update
>sudo apt-get install -y kubelet=1.22.8-00 kubeadm=1.22.8-00 kubectl=1.22.8-00
>sudo apt-mark hold kubelet kubeadm kubectl


1. TO CHECK VERSION
> kubectl version
> kubelet version
> kubeadm version


**********************************************************************
```

STEP:4 DISABLING SWAPP
****

> sudo swapoff -a
> sudo sed -i '/ swap / s/^/#/' /etc/fstab

****
STEP:5 ADDING C GROUP
****

> sudo vi /etc/docker/daemon.json

press 'i' for insert

```
{
      "exec-opts":["native.cgroupdriver=systemd"]
}
```

press 'esc' to escape

> :wq!   //to save the file

once the file updated we need to restart the service

> sudo systemctl daemon-reload && sudo systemctl restart docker && sudo systemctl restart kubelet

> sudo docker info | grep -i cgroup

you will get message like:

```
 Cgroup Driver: systemd
 Cgroup Version: 2
  cgroupns
```

****
STEP:6 INITIALIZE KUBERNATE CLUSTER
****

> sudo kubeadm init

or

> sudo kubeadm init --ignore-preflight-errors=all

if all will be ok you will get message that :

your kubernate control-plane has been initiated successfully
also you will get some of the commands in CLI as below

```
      >mkdir -p $HOME/.kube
      >sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
      >sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Then you can join any number of worker nodes by running the following on each as root:

```
kubeadm join 172.31.82.150:6443 --token t1002s.jyols10e5bmpdzkx \
        --discovery-token-ca-cert-hash
sha256:64d33fe991c1700abb7e8bddbb56f51ec659feb8b76db6faee970ddea1a009a2
[note: to prepare worker node follow step-1 to step:5 and then copy paste the
token  on worker node with prefix 'sudo your_token']

--------------------------------------------------------------------------------
------------------------------

execute all the commands line by line
--------------------------------------

1. to get node details

> kubectl get node

2. get cluster details

> kubectl get svc

3. deploy any app /code on cluster (this will work only if cluster is started)

let's deploy weve network to cluster

> export kubever=$(kubectl version | base64 | tr -d '\n' )
> sudo kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$kubever"


4. validate running pods

> kubectl get pods --all-namespaces


********************************************************************************
*****
STEP:7 HOST A SPRINGBOOT APP IN KUBERNATE PODS
********************************************************************************
*****

FOR SPRINGBOOT APP

1. MAVEN INSTALLTION

2. JDK INSTALLTION

3. CLONE YOUR APP FROM GITHUB



1. MAVEN INSTALLTION
---------------------

> sudo apt-get  update
> sudo apt install maven -y


2. JDK INSTALLTION
--------------------

> sudo apt-get  update
> sudo apt install default-jdk -y
```

3. CLONE YOUR APP

> git clone https://github.com/Nikunj-Java/SpringBootDockerApp.git

4. CHANGE DIRECTORY

>cd SpringBootDockerApp

5. CREATE .JAR FILE

> mvn clean install

6. BUILD DOCKER IMAGE

> sudo docker build -t springboot .
> sudo docker images

7. CUBE PROXY

> kubectl run springbootapp --image=springboot --port=8082

you will get msg that: pod/springbootapp created

> kubectl get pods

8. CREATE DEPLOYMENT SERVICE

> kubectl expose pod/springbootapp --port=8082 --target-port=8082 --
type=LoadBalancer

you will get msg: service/springbootapp exposed

1. TO GET LIST OF SERVICES RUNNING

> kubectl get service

2. TO GET DETAILED DESCRIPTION OF APP RUNNING ON POD

> kubectl describe svc springbootapp