K8s 09

1) Execute all the steps shown in video.

--check minikube version

```
naren@narendar MINGW64 /c/minikube
$ minikube version
minikube version: v1.35.0
commit: dd5d320e41b5451cdf3c01891bc4e13d189586ed-dirty
```

--start minikube and check minikube status

```
naren@narendar MINGW64 /c/minikube
$ minikube start
* minikube v1.35.0 on Microsoft Windows 11 Home Single Language 10.0.26100.3775 Build 26100.3775
* Using the docker driver based on existing profile
* Starting "minikube" primary control-plane node in "minikube" cluster
* Pulling base image v0.0.46 ...
* Updating the running docker "minikube" container ...
! Failing to connect to https://registry.k8s.io/ from inside the minikube container
* To pull new external images, you may need to configure a proxy: https://minikube.sigs.k8s.io/docs/reference/networking/proxy/
* Preparing Kubernetes v1.32.0 on Docker 27.4.1 ...
* 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

naren@narendar MINGW64 /c/minikube
$ minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

--create deployment.yaml

MINGW64:/c/minikube

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:latest
        ports:
        - containerPort: 80
```

Run yaml then pod got created

```
naren@narendar MINGW64 /c/minikube
$ kubectl apply -f deployment.yaml
deployment.apps/nginx-deployment unchanged

naren@narendar MINGW64 /c/minikube
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-96b9d695-9pr9k     1/1     Running   0          49s
```

--create service.yaml

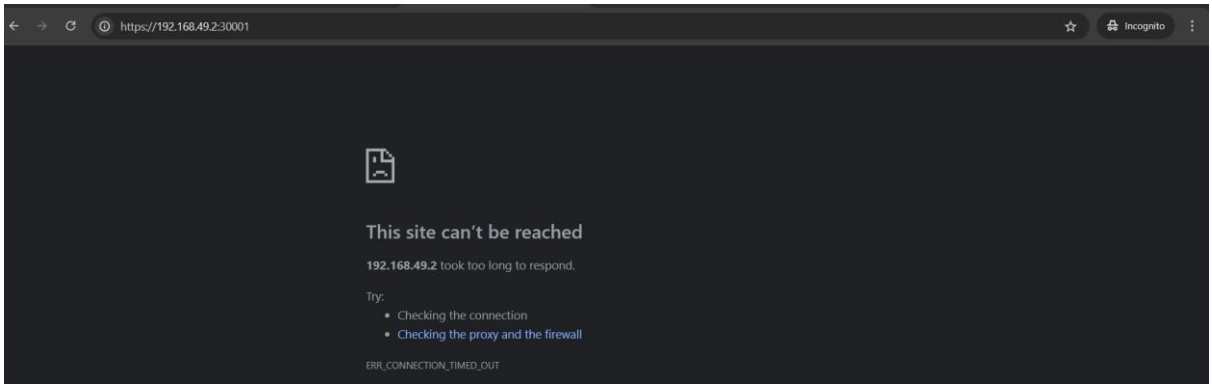
MINGW64:/c/minikube

```
apiVersion: v1
kind: Service
metadata:
  name: nginx-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 80
      nodePort: 30001
  type: NodePort
```

Run yaml file

```
naren@narendar MINGW64 /c/minikube
$ vi service.yaml

naren@narendar MINGW64 /c/minikube
$ kubectl apply -f service.yaml
service/nginx-service created

naren@narendar MINGW64 /c/minikube
$ kubectl get services
NAME            TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP   10.96.0.1        <none>        443/TCP          12m
nginx-service   NodePort    10.110.199.154   <none>        80:30001/TCP     20s
```

-- Access it with ip



Not got so generate url

--create url for service

minikube service nginx-service –url

access now on browser with above url

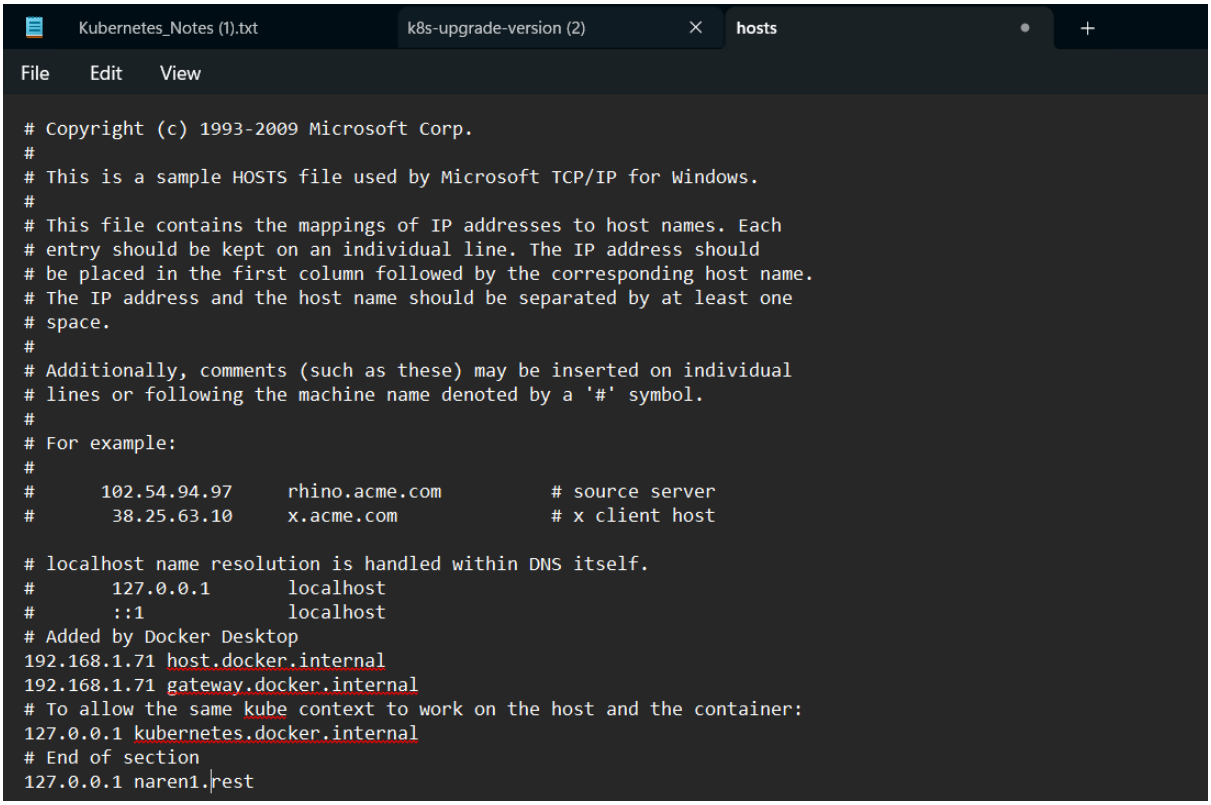
----create ingress

--to create ingress controller run this command

to enable ingress on minikube:

minikube addons enable ingress

--add dns in hostfile



--Yaml file to create ingress

MINGW64:/c/minikube

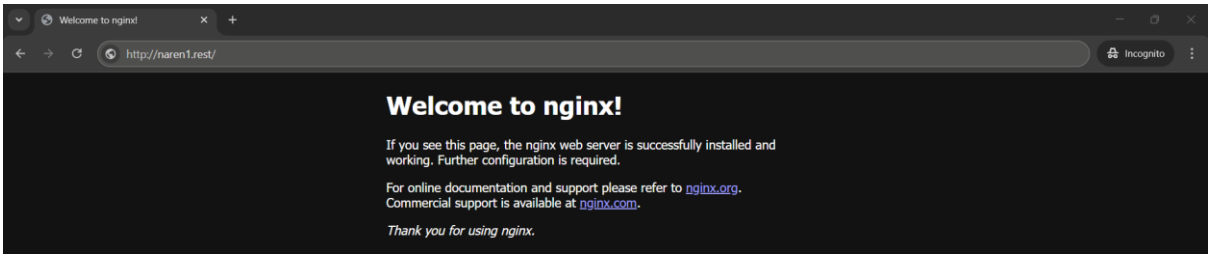
```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: nginx-ingress
spec:
  rules:
  - host: naren1.rest
    http:
      paths:
      - path: /
        pathType: Prefix
        backend:
          service:
            name: nginx-service
            port:
              number: 80
```

Run yaml

```
naren@narendar MINGW64 /c/minikube
$ vi ingress.yaml

naren@narendar MINGW64 /c/minikube
$ kubectl apply -f ingress.yaml
ingress.networking.k8s.io/nginx-ingress created

naren@narendar MINGW64 /c/minikube
$ minikube ip
192.168.49.2

naren@narendar MINGW64 /c/minikube
$ minikube tunnel
* Tunnel successfully started

* NOTE: Please do not close this terminal as this process must stay alive for the tunnel to be accessible ...

! Access to ports below 1024 may fail on Windows with OpenSSH clients older than v8.1. For more information, see: https://minikube.sigs.k8s.io/docs/handbook/accessing/#acce
ss-to-ports-1024-on-windows-requires-root-permission
* Starting tunnel for service nginx-ingress.
* Stopping tunnel for service nginx-ingress.
```

```
naren@narendar MINGW64 /c/minikube
$ kubectl get ingress
NAME            CLASS   HOSTS          ADDRESS        PORTS   AGE
nginx-ingress   nginx   naren1.rest    192.168.49.2   80      99m
```

Access it with hostname on browser



2) Upgrade k8s cluster from 1.26 to 1.28 Version.

--check nodes

root@master: ~

```
root@master:~# kubectl get nodes
NAME                STATUS   ROLES           AGE   VERSION
ip-172-31-13-158    Ready    <none>          9d    v1.29.15
ip-172-31-4-112     Ready    <none>          9d    v1.29.15
master              Ready    control-plane   9d    v1.29.15
root@master:~#
```

--upgrade the master node

--Check the existing Kubeadm version

```
root@master:~# kubeadm version -o json
{
  "clientVersion": {
    "major": "1",
    "minor": "29",
    "gitVersion": "v1.29.15",
    "gitCommit": "0d0f172cdf9fd42d6feee3467374b58d3e168df0",
    "gitTreeState": "clean",
    "buildDate": "2025-03-11T17:46:36Z",
    "goVersion": "go1.23.6",
    "compiler": "gc",
    "platform": "linux/amd64"
  }
}
```

--unhold kubeadm and Install the latest version

```
root@master:~# sudo apt-mark unhold kubeadm
Canceled hold on kubeadm.
root@master:~# sudo apt-cache madison kubeadm | tac
   kubeadm | 1.29.0-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.1-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.2-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.3-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.4-2.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.5-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.6-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.7-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.8-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.9-1.1  | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.10-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.11-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.12-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.13-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.14-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
   kubeadm | 1.29.15-1.1 | https://pkgs.k8s.io/core:/stable:/v1.29/deb  Packages
```

```
root@master:~# sudo apt-get install -y kubeadm=1.29.15-1.1
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubeadm is already the newest version (1.29.15-1.1).
0 upgraded, 0 newly installed, 0 to remove and 30 not upgraded.
root@master:~# sudo apt-mark hold kubeadm
kubeadm set on hold.
```

--Decide on the upgrade version

```
root@master:~# sudo kubeadm upgrade plan
[upgrade/config] Making sure the configuration is correct:
[upgrade/config] Reading configuration from the cluster...
[upgrade/config] FYI: You can look at this config file with 'kubectl -n kube-system get cm kubeadm-config -o yaml'
[preflight] Running pre-flight checks.
[upgrade] Running cluster health checks
[upgrade] Fetching available versions to upgrade to
[upgrade/versions] Cluster version: v1.29.15
[upgrade/versions] kubeadm version: v1.29.15
I0423 14:47:54.928536    9291 version.go:256] remote version is much newer: v1.32.4; falling back to: stable-1.29
[upgrade/versions] Target version: v1.29.15
[upgrade/versions] Latest version in the v1.29 series: v1.29.15
```

--Apply Kubeadm upgrade

```
[upgrade/staticpods] Current and new manifests of kube-controller-manager are equal, skipping upgrade
[upgrade/staticpods] Preparing for "kube-scheduler" upgrade
[upgrade/staticpods] Current and new manifests of kube-scheduler are equal, skipping upgrade
[upload-config] Storing the configuration used in ConfigMap "kubeadm-config" in the "kube-system" Namespace
[kubelet] Creating a ConfigMap "kubelet-config" in namespace kube-system with the configuration for the kubelets in the cluster
[upgrade] Backing up kubelet config file to /etc/kubernetes/tmp/kubeadm-kubelet-config3328898735/config.yaml
[kubelet-start] Writing kubelet configuration to file "/var/lib/kubelet/config.yaml"
[kubeconfig] Writing "admin.conf" kubeconfig file
[kubeconfig] Writing "super-admin.conf" kubeconfig file
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to get nodes
[bootstrap-token] Configured RBAC rules to allow Node Bootstrap tokens to post CSRs in order for nodes to get long term certificate credenti
[bootstrap-token] Configured RBAC rules to allow the csrapprover controller automatically approve CSRs from a Node Bootstrap Token
[bootstrap-token] Configured RBAC rules to allow certificate rotation for all node client certificates in the cluster
[addons] Applied essential addon: CoreDNS
[addons] Applied essential addon: kube-proxy

[upgrade/successful] SUCCESS! Your cluster was upgraded to "v1.29.15". Enjoy!

[upgrade/kubelet] Now that your control plane is upgraded, please proceed with upgrading your kubelets if you haven't already done so.
```

--Drain the Node to evict all workloads.

```
root@master:~# kubectl drain master --ignore-daemonsets
node/master cordoned
Warning: ignoring DaemonSet-managed Pods: kube-system/calico-node-cp9gc, kube-system/ebs-csi-node-wvcf9, kube-syst
ode-gb4kv, kube-system/kube-proxy-cfj8v
node/master drained
```

--Uncordon the Node and Verify the Node Status

kubectl uncordon master

```
root@master:~# kubectl uncordon master
node/master already uncordoned
```

--Check nodes status of master

```
root@master:~# kubectl get nodes
NAME                STATUS   ROLES           AGE   VERSION
ip-172-31-13-158    Ready    <none>          9d    v1.29.15
ip-172-31-4-112     Ready    <none>          9d    v1.29.15
master              Ready    control-plane   9d    v1.30.0
```

Master upgraded to v1.30.0

--now Upgrade Worker Node

--update the packages and install all in worker node

```
root@ip-172-31-4-112:~# sudo apt-mark unhold kubeadm && \
sudo apt-get update && sudo apt-get install -y kubeadm=1.30.1-1.1 && \
sudo apt-mark hold kubeadm
Canceled hold on kubeadm.
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Get:5 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.29/deb  InRelease
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [1026 kB]
Get:7 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [223 kB]
Get:8 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1057 kB]
Get:9 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [267 kB]
Get:10 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [782 kB]
Get:11 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [147 kB]
Get:12 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [833 kB]
Get:13 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [181 kB]
Fetched 4769 kB in 2s (2509 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

--upgrade the kubectl and kubelet in worker node

```
sudo apt-get update
sudo apt-get install -y kubelet=1.30.0-1.1 kubectl=1.30.0-1.1
sudo apt-mark hold kubelet kubectl
sudo systemctl daemon-reexec
sudo systemctl restart kubelet
Canceled hold on kubelet.
Canceled hold on kubectl.
Hit:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble InRelease
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]
Hit:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 https://prod-cdn.packages.k8s.io/repositories/isv:/kubernetes:/core:/stable:/v1.30/deb  InRelease
Hit:5 http://security.ubuntu.com/ubuntu noble-security InRelease
Fetched 126 kB in 0s (266 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
kubelet is already the newest version (1.30.0-1.1).
kubectl is already the newest version (1.30.0-1.1).
0 upgraded, 0 newly installed, 0 to remove and 37 not upgraded.
kubelet set on hold.
```

-- then hold the kubelet and kubectl

```
kubelet set on hold.
kubectl set on hold.
root@ip-172-31-4-112:~# client_loop: se
```

--check version of kubectl and kubelet

```
root@ip-172-31-4-112:~# kubelet --version
Kubernetes v1.30.0
```

--now check the version for worker node

```
root@master:~# kubectl get nodes
NAME              STATUS   ROLES           AGE   VERSION
ip-172-31-4-112   Ready    <none>          9d    v1.30.0
master            Ready    control-plane   9d    v1.30.0
```

We can see both the master and worker node has been successfully upgraded