# Cluster Administration

Verify Cluster setup

Kubernetes API

Deep-dive into Master setup

Registering Working Nodes

Deploying the first pod and accessing it

Working with Kubeadm

Kubernetes Dashboard

ETCD - Backup & Restore

Upgrading Kubernetes Cluster

# Verify Cluster Setup

In the Master node,

kubectl get nodes

kubectl config view

# Kubernetes API

Proxy port 8080

kubectl proxy --port=8080

Note: Do, CTRL+C to close the proxy. After accessing API Server in the Desktop

Click on the master tab on the lab, and then click on the desktop option.   
Open Firefox browser to access the API server

<http://localhost:8080>

# Deep-dive into Master setup

kubectl cluster-info

kubectl cluster-info dump > cluster-dump

kubectl get node worker-node-1.example.com

kubectl describe node worker-node1.example.com | less

# Look at Status(should be FALSE), Address, Capacity, and Events

kubectl get namespaces

kubectl get pods -A

kubectl get pods -n kube-system

# Look into /etc/kubernetes/ - Config, manifests & pki

kubectl get pods -n kube-system -o wide | grep proxy

service kubelet status

# Registering Working Nodes

kubectl get nodes

kubectl describe node worker-node1.example.com

kubectl delete node worker-node1.example.com

kubectl get nodes

Create a new file with Node info,

vi nodereg.json

{

"kind": "Node",

"apiVersion": "v1",

"metadata": {

"name": "worker-node-1.example.com",

"labels": {

"name": "firstnode"

}

}

}

kubectl create -f nodereg.json

# kubectl get nodes

# Deploying the first pod and accessing it

kubectl run nginxpod --image=nginx --port 80

kubectl get pods

kubectl describe pod nginxpod

kubectl exec -it nginxpod /bin/sh

# Working with Kubeadm

**Viewing the configuration details**

kubeadm config print init-defaults

**Kubernetes certificates**

sudo kubeadm certs check-expiration

# 

# Kubernetes Dashboard

**Deploying the dashboard**

kubectl apply -f <https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml>

**Verifying the Dashboard resources**

kubectl get pods -n kubernetes-dashboard -o wide

kubectl get deployment -n kubernetes-dashboard -o wide

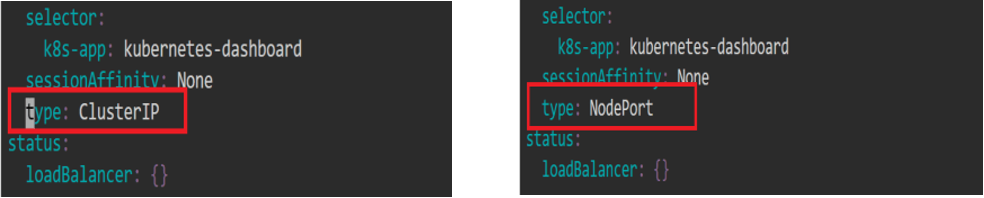
kubectl get svc -n kubernetes-dashboard -o wide

**Editing the Service type of the dashboard**

kubectl edit svc -n kubernetes-dashboard kubernetes-dashboard

**Note:** Change the attribute after entering the deployment

type: ClusterIP (image 1) to NodePort (image 2)



**Verifying the changes**

kubectl get svc -n kubernetes-dashboard -o wide

Note down the service(node-port) port number , here it is 31851****

**Checking where the Pod is running**

kubectl get pods -n kubernetes-dashboard -o wide

kubectl get svc -n kubernetes-dashboard -o wide

kubectl get nodes -o wide

**Accessing Kubernetes Dashboard**

Click on the master tab on the lab, and then click on the desktop option.   
Open Firefox browser to access the API server

[**https://localhost**](https://localhost)**:<<NodePort>>**

Example: https://localhost:31851

Click on Advanced -> Accept Risk and Continue

On the Kubernetes Dashboard,

Select **Token** from the given options and enter the token

**Note:** To get the token, navigate to the master node and use the command:

kubectl -n kube-system describe secret $(kubectl -n kube-system get secret | awk '/^deployment-controller-token-/{print $1}') | awk '$1=="token:"{print $2}'

**Cleanup:**

kubectl delete -f <https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml>

# ETCD - Backup

Step 1: Get URLs and keys

kubectl describe pod etcd-master -n kube-system

Get client-URL, cert, key, and trusted-ca location

Step 2: Command

sudo snap install etcd

sudo apt install etcd-client

sudo chmod a+rw -R /etc/kubernetes/pki

sudo ETCDCTL\_API=3 etcdctl snapshot save etcd\_backup.db \

--endpoints https://172.31.49.128:2379 \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key \

--cacert=/etc/kubernetes/pki/etcd/ca.crt

Step 3: Verify

sudo ETCDCTL\_API=3 etcdctl --write-out=table snapshot status etcd\_backup.db \

--endpoints https://<cluster-ip>:2379 \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key \

--cacert=/etc/kubernetes/pki/etcd/ca.crt

# ETCD - Restore

Works only on multi-master etcd node, Don’t try in single master setup.

This command is only for Reference.

ETCDCTL\_API=3 etcdctl --write-out=table snapshot restore etcd\_backup.db \

--endpoints https://<cluster-ip>:2379 \

--cert=/etc/kubernetes/pki/etcd/server.crt \

--key=/etc/kubernetes/pki/etcd/server.key \

--cacert=/etc/kubernetes/pki/etcd/ca.crt

# Upgrading Kubernetes Cluster

**Finding the latest release of Kubernetes**

sudo apt update

sudo apt-cache madison kubeadm

sudo apt-cache madison kubectl

**Verifying the current version of Kubernetes**

kubeadm version

kubectl get nodes

**Upgrading the repositories**

sudo apt update

sudo apt upgrade

**Holding the Kubernetes versions**

sudo apt-mark hold kubeadm

sudo apt-mark hold kubelet kubectl

**Upgrading the control plane(Master)**

sudo apt-get install -y kubeadm=1.23.12-00 --allow-change-held-packages

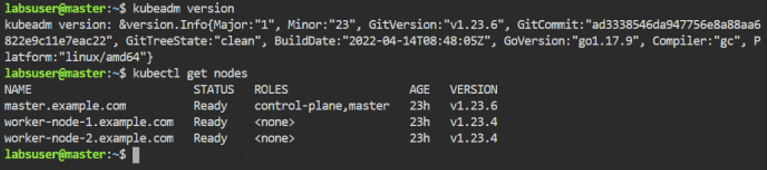
sudo apt-get install -y kubelet=1.23.12-00 kubectl=1.23.12-00 --allow-change-held-packages

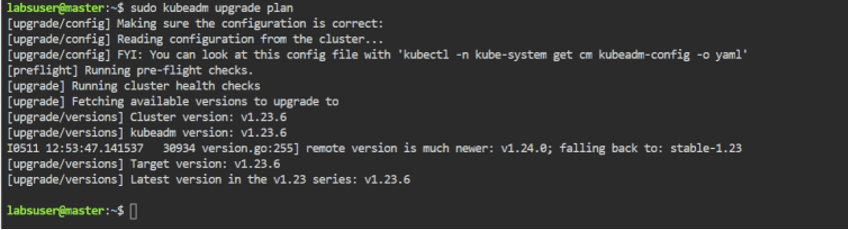
**Verifying the updated version of Kubernetes**

kubeadm version

kubectl get nodes

sudo kubeadm upgrade pla**n**

**

**