Steps:

**First Backup the cluster**

1. install the etcd-client

apt install etcd-client

2. run the below command

sudo ETCDCTL\_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key --cacert=/etc/kubernetes/pki/etcd/ca.crt snapshot save /opt/etcd-backup.db

We can see status in the place of save we can add that

**To restore**

service kube-apiserver stop

sudo ETCDCTL\_API=3 etcdctl --endpoints=https://127.0.0.1:2379 --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key --cacert=/etc/kubernetes/pki/etcd/ca.crt snapshot restore /opt/etcd-backup.db --data-dir /var/lib/etcd-from-backup

After that we need to change the data-dir in configuration file in /etc/kubernetes/manifests/etcd.yaml as --data-dir /var/lib/etcd-from-backup

--listen-client-urls=https://127.0.0.1:2379,https://192.5.86.6:2379

https://192.5.86.6:2379

Backup the all resources in server

kubectl get all --all-namespaces -o yaml > all-deploy-services.yaml

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

--data-directory = /var/lib/etcd

**Add Below repository for each version when upgrade**

curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg --dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg

echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

**Next steps:**

apt update

apt-cache madison kubeadm

# find the latest 1.24 version in the list

# it should look like 1.24.x-\*, where x is the latest patch

**Upgrading control plane nodes**

apt-mark unhold kubeadm && \

apt-get install -y kubeadm=1.28.9-2.1&& \

apt-mark hold kubeadm

Verify that the download works and has the expected version:

kubeadm version

Verify the upgrade plan:

kubeadm upgrade plan

sudo kubeadm upgrade apply v1.28.9

For the other control plane nodes

Same as the first control plane node but use:

kubeadm upgrade node

### Drain the node

Prepare the node for maintenance by marking it unschedulable and evicting the workloads:

# replace <node-to-drain> with the name of your node you are draining

kubectl drain --ignore-daemonsets --force

Upgrade the kubelet and kubectl:

# replace x in 1.24.x-\* with the latest patch version

apt-mark unhold kubelet kubectl && \

apt-get install -y kubelet=1.28.9-2.1 kubectl=1.28.9-2.1&& \

apt-mark hold kubelet kubectl

sudo systemctl daemon-reload

sudo systemctl restart kubelet

kubectl uncordon <node-to-drain>

kubectl -n kube-system get cm kubeadm-config -o yaml

**Upgrade worker nodes**

The upgrade procedure on worker nodes should be executed one node at a time or few nodes at a time, without compromising the minimum required capacity for running your workloads.

apt update

apt-cache madison kubeadm

Upgrade kubeadm:

# replace x in 1.24.x-00 with the latest patch version

apt-mark unhold kubeadm && \

apt-get install -y kubeadm=1.28.9-2.1 && \

apt-mark hold kubeadm

For worker nodes this upgrades the local kubelet configuration:

sudo kubeadm upgrade node

Prepare the node for maintenance by marking it unschedulable and evicting the workloads:

# replace <node-to-drain> with the name of your node you are draining

kubectl drain <node-to-drain> --ignore-daemonsets

kubectl drain prodk8sworker4 --ignore-daemonsets --delete-emptydir-data --force

# replace x in 1.24.x-00 with the latest patch version

apt-mark unhold kubelet kubectl && \

apt-get install -y kubelet=1.28.9-2.1 kubectl=1.28.9-2.1&& \

apt-mark hold kubelet kubectl

sudo systemctl daemon-reload

sudo systemctl restart kubelet

Bring the node back online by marking it schedulable:

# replace <node-to-drain> with the name of your node

kubectl uncordon <node-to-drain>

Check the status of the cluster with the below command

kubectl get nodes

**if kubectl not start in master follow below steps**

kubeadm init phase kubelet-start

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

# echo """

runtime-endpoint: unix:///run/containerd/containerd.sock

image-endpoint: unix:///run/containerd/containerd.sock

""" > /etc/crictl.yaml

# systemctl restart containerd

**If kubectl not works in worker node**

Remove –network-plugin in /var/lib/kubelet/kubeadm-flags.env file

**To install lower version when we installed Higher version**

apt-get install -y kubeadm= 1.25.16-1.1 --allow-downgrades

**Weave net version upgrade**

Weave net version upgrade for 1.28 version -- latest weave 2.8.7

kubectl apply -f <https://reweave.azurewebsites.net/k8s/v1.28/net.yaml>

Weave net present version ------ 2.8.1

kubectl apply -f https://github.com/weaveworks/weave/releases/download/v2.8.1/weave-daemonset-k8s.yaml