1. Understand Kubernetes cluster upgrade process

Upgrade from v1.14.1 to 1.14.4

in short

```
kubeadm upgrade plan
apt instal kubeadm=1.14.4-00
```

show

Follow instructions from Kubernetes document.

The upgrade workflow at high level is the following:

- Upgrade the primary control plane node.
- Upgrade additional control plane nodes.
- Upgrade worker nodes.

Find the latest stable 1.14 version:

```
sudo -i
apt update
apt-cache policy kubeadm
# find the latest 1.14 version in the list
# it should look like 1.14.x-00, where x is the latest patch
```

Upgrade the first control plane node (master):

```
apt-mark unhold kubeadm kubelet && apt-get update && apt-get install -y kubeadm=1.14.4-00 && apt-mark hold kubeadm
kubeadm upgrade plan
kubeadm upgrade apply v1.14.4
```

Upgrade the kubelet and kubectl on the control plane node:

```
apt-mark unhold kubelet kubectl && apt-get update && apt-get install -y kubelet=1.14.4-00 kubectl=1.14.4-00 && apt-mark hold kubelet kubectl
```

Restart kubelet:

```
systemctl restart kubelet
```

Upgrade additional control plane nodes:

```
sudo kubeadm upgrade node experimental-control-plane
```

Upgrade worker nodes:

Upgrade kubeadm on all worker nodes:

```
apt-mark unhold kubeadm kubelet && apt-get update && apt-get install -y kubeadm=1.14.4-00 && apt-mark hold kubeadm
```

Going back to master node and cordon the worker node. Prepare the node for maintenance by marking it unschedulable and evicting the workloads:

```
kubectl drain worker1 --ignore-daemonsets --delete-local-data
```

Enter the worker node and Upgrade the kubelet config:

```
sudo -i
kubeadm upgrade node config --kubelet-version v1.14.4
```

Upgrade kubelet and kubectl, and restart kubectl

```
apt-mark unhold kubelet kubectl && apt-get update && apt-get install -y kubelet=1.14.4-00 kubectl=1.14.4-00 && apt-mark hold kubelet kubectl
```

```
systemctl restart kubelet
```

Uncordon the worker node and verify the status of the cluster:

```
kubectl uncordon worker1
kubectl get nodes
```

2. Facilitate operating system upgrades

show

During maintenance, when we need to take down a node, it is important to keep the service running by evicting the pods on that node, moving it to another node. After maintenance, you can continue to schedule pods on that node.

```
kubectl drain worker1 --ignore-daemonsets --delete-local-data
kubectl get nodes
kubectl uncordon worker1
kubectl get nodes
```

In case of failure, the node needs to be deleted, you can just as easily remove the node and replace it with a new one, joining it to the cluster.

```
kubectl drain worker1 --ignore-daemonsets --delete-local-data
kubectl get nodes
kubectl delete node worker1
kubeadm token create --print-join-command
```

3. Implement backup and restore methodologies

Here, here, here, and here.

- create a pod name family with image nginx
- create a deployment name nginx with image nginx, scale the deployment to replicas=3
- take a snapshot name snapshot.db with etcdctl
- delete all the pod and deployment (simulate the disaster happens that delete all pods and deployments in the cluster)
- restore to the previous state of the cluster with the snapshot.db

show

```
kubectl run family --generator=run-pod/v1 --image=nginx
kubectl create deployment nginx --image=nginx
kubectl scale deployment nginx --replicas=3
sudo ETCDCTL_API=3 etcdctl snapshot save snapshot.db --endpoints=https://[127.0.0.1]:2379 --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key
kubectl delete pod family
kubectl delete deployment nginx
sudo ETCDCTL_API=3 etcdctl snapshot restore snapshot.db --endpoints=https://[127.0.0.1]:2379 --cacert=/etc/kubernetes/pki/etcd/ca.crt --cert=/etc/kubernetes/pki/etcd/server.crt --key=/etc/kubernetes/pki/etcd/server.key --data-dir=/var/lib/etcd-from-backup --name=master --initial-advertise-peer-urls="http://localhost:2380" --initial-cluster="master=http://localhost:2380" --initial-cluster-token="etcd-cluster-1"
```

Modify /etc/kubernetes/manifests/etcd.yaml: --initial-cluster-token=etcd-cluster-1; --data-dir=/var/lib/etcd-from-backup, update the volumeMounts & hostPath to new path /var/lib/etcd-from-backup