CKA Support

1 Static Pods

Static Pods

- **Static Pods** are managed directly **by the kubelet** daemon on a specific node, without the API server observing them.

- Unlike Pods that are managed by the control plane (for example, a Deployment); instead, the kubelet watches each static Pod (and restarts it if it fails).

- As default, kubelet check **/etc/kubernetes/manifests** folder and if there is a pod manifest file,it creates the pod.

- For Example, all controlplane components are static pods.

Static Pods

- **/etc/kubernetes/manifests** folder defined in the **/var/lib/kubelet/config.yaml** file.

```
apiVersion: kubelet.config.k8s.io/v1beta1
…
…
kind: KubeletConfiguration
staticPodPath: /etc/kubernetes/manifests
…
```

Static Pods

- Static Pods are always bound to one Kubelet on a specific node.

- The kubelet automatically tries to **create a mirror Pod** on the Kubernetes API server for each static Pod. This means that the Pods running on a node are visible on the API server, but cannot be controlled from there.

Static Pods

- The Pod names will be suffixed with the node hostname with a leading hyphen.

```
controlplane $ kubectl get node
NAME            STATUS      ROLES           AGE         VERSION
controlplane    Ready       control-plane   17d         v1.29.0
node01          Ready       <none>          17d         v1.29.0

controlplane $ kubectl get po -n kube-system
NAME                                READY       STATUS      RESTARTS        AGE
etcd-controlplane                   1/1         Running     2 (14m ago)     17d
kube-apiserver-controlplane         1/1         Running     2 (14m ago)     17d
kube-controller-manager-controlplane 1/1        Running     2 (14m ago)     17d
kube-scheduler-controlplane

controlplane $ kubectl get po
NAME            READY       STATUS      RESTARTS        AGE
static-web-node01  1/1      Running     0               22s
```

2 Cluster Update

Static Pods

- https://kubernetes.io/docs/tasks/administer-cluster/kubeadm/kubeadm-upgrade/
- https://killercoda.com/killer-shell-cka/scenario/cluster-upgrade

3 etcd backup and restore

---

etcd backup and restore

- https://kubernetes.io/docs/tasks/administer-cluster/configure-upgrade-etcd/
- https://killercoda.com/chadmcrowell/course/cka/kubernetes-backup-etcd

4 Apiserver Crash Logs

Log Locations

Log locations to check:

- /var/log/pods
- /var/log/containers
- crictl ps + crictl logs
- docker ps + docker logs (in case when Docker is used)
- kubelet logs: /var/log/syslog or **journalctl -u kubelet**
- sudo systemctl status kubelet
- sudo systemctl start kubelet

Certificate Management

View and renew the Certificates

Certificates Folders as Default

- control plane component certificates
  - /etc/kubernetes/pki/ folder
- Kubelet certificates

  - /var/lib/kubelet/pki folder

View the certificate manually:

- ```
  openssl x509  -noout -text -in ./server.crt
  ```

Certificate Management with kubeadm

- **Check certificate expiration**

You can use the `check-expiration` subcommand to check when certificates expire:

```
kubeadm certs check-expiration
```

- **Manual certificate renewal**

`kubeadm certs renew` can renew any specific certificate or, with the subcommand `all`, it can renew all of them, as shown below:

```
kubeadm certs renew all
kubeadm certs renew apiserve r
```

- **Automatic certificate renewal**

kubeadm renews all the certificates during control plane upgrade.

kubeadm upgrade node

THANKS!

Any questions?