* Certified Kubernetes Administrators Certification with live practice tests right in your browser – CKA
* [Mumshad Mannambeth](https://www.udemy.com/user/mumshad-mannambeth/)
* UDEMY

Etcd: Database that stores information in a key-value format. Control client is a command line for etcd. Port number is 2379. Every changes updates on the etcd server.

Additional information about ETCDCTL Utility  
  
ETCDCTL is the CLI tool used to interact with ETCD.  
  
ETCDCTL can interact with ETCD Server using 2 API versions - Version 2 and Version 3.  By default its set to use Version 2. Each version has different sets of commands.

For example ETCDCTL version 2 supports the following commands:

etcdctl backup

etcdctl cluster-health

etcdctl mk

etcdctl mkdir

etcdctl set

Whereas the commands are different in version 3

etcdctl snapshot save

etcdctl endpoint health

etcdctl get

etcdctl put

To set the right version of API set the environment variable ETCDCTL\_API command

export ETCDCTL\_API=3

When API version is not set, it is assumed to be set to version 2. And version 3 commands listed above don't work. When API version is set to version 3, version 2 commands listed above don't work.

Apart from that, you must also specify path to certificate files so that ETCDCTL can authenticate to the ETCD API Server. The certificate files are available in the etcd-master at the following path. We discuss more about certificates in the security section of this course. So don't worry if this looks complex:

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

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

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

So for the commands I showed in the previous video to work you must specify the ETCDCTL API version and path to certificate files. Below is the final form:

kubectl exec etcd-master -n kube-system -- sh -c "ETCDCTL\_API=3 etcdctl get

Kube-apiserver: responsible orchestrating all operations within the cluster. Is the primary management component in Kubernetes. The kube-api server fisrt authenticates the request and validates it then retrieves the data from the etcd cluster and responds back with the request information.

To summarize the kube-api server is responsible for authenticating and validating requests, retrieving and updating data in etcd data store in fact kube api server is the only component that interacts directly with the etcd data store. The other components such as the scheduler, kube-controller-manager and kubelet uses the api server to perform updates in the cluster in their respective areas.

**Kube Controller Manager:** Manages various controller in Kubernetes. A controller is a process that continuously monitors the state of various components within the system and works towards bringing the whole system to the desired functioning state. Node controller is responsible for monitoring the status of the nodes and taking necessary actions to keep the application running. The node controller checks the status of the nodes every 5 seconds. It waits for 40 secons before marking it unreachable. After he delete the nodes and assigned to that node and provisions them on the healthy ones. Replication controller responsible of replica sets and ensuring that the desired number of pods are available at all times within the set. If a pod dies it creates another one. All of the controller is in kube-controller-manager.

Replication Controller: Older techology. Replication Set new techonology. Big difference is selector section. Selector section helps the replica set identify what pods fall under it.

**Kube Scheduler:** Identifies the right note to place a container on based on the containers. Responsible for deciding which pods goes on which node. Kubelet or the captain on the ship is who creates the pod on the ship. The scheduler only decides which pod goes where.

**Kubelet:** A kubelet is an agent each node in a cluster. Listens for instructions from kube-api server and manages containers. The kubelt in the kubernetes worker node, registers the node with the kubernetes cluster. When it receives intructions to load a containe or a pod on the node, it requests the container run time engine which may be Docker to pull the required image and run an instance. The kubelet then continuous to monitor the state of the pod and the containers in it and reports to the kube-api on a timely basis.

­

**Kube-proxy:** Commucation between workes nodes are enabled kube-proxy.

**Pods:** A pod is a single instance of an application. A pod is the smallest object that you can create in Kubernetes.