What is ETCD?

etcd is a consistent and highly-available key value store used as Kubernetes' backing store for all cluster data. **It stores k8s state data, meta data and configuration data.[json based]**



What is a Key-Value Store?

A key-value store, or key-value database is a simple database that uses an associative array (think of a map or dictionary) as the fundamental data model where each key is associated with one and only one value in a collection. This relationship is referred to as a key-value pair.

What is watch function?

Throughout the time watch function keep monitoring desire state and actual state, and if any deviation found then it informs to API server to take corrective action.



Important point about ETCD

* Open-source key value data source
* Core component of K8s
* Single source of truth at any point of time
* Fully replicated
* Reliable
* Using RAFT algorithm for internal management

How ETCD works?

ETCD uses RAFT algorithm for internal management.

Simple leader election with Kubernetes and Docker

Typically, in leader election, a set of candidates for becoming leader is identified. These candidates all race to declare themselves the leader. One of the candidates wins and becomes the leader. Once the election is won, the leader continually "heartbeats" to renew their position as the leader, and the other candidates periodically make new attempts to become the leader. This ensures that a new leader is identified quickly, if the current leader fails for some reason.

Implementing leader election usually requires either deploying software such as ZooKeeper, etcd or Consul and using it for consensus, or alternately, implementing a consensus algorithm on your own. We will see below that Kubernetes makes the process of using leader election in your application significantly easier.

[Source] <https://kubernetes.io/blog/2016/01/simple-leader-election-with-kubernetes/>

How values are getting updated

Lets, say we have Key-value stored as 7, and now we want to update it to 17. So below steps need to perform

* Request to update 7 to 17 to leader
* Leader will update his most of worker first
* Once most of the worked are updated leader would be updating himself, if there is any request on ungraded worked node then worked node will forward that request to leader for further processing

