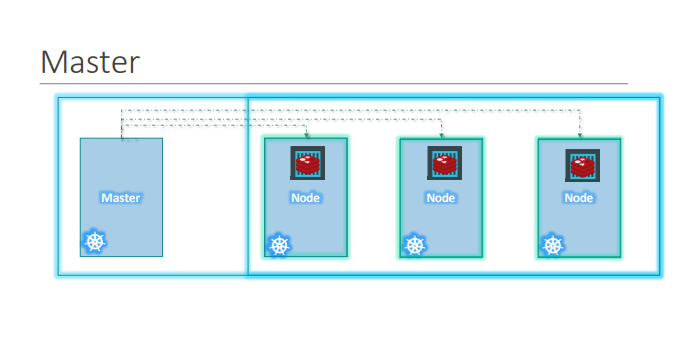
**Kubernetes Architecture**

**A node is a machine – physical or virtual – on which kubernetesis installed.**

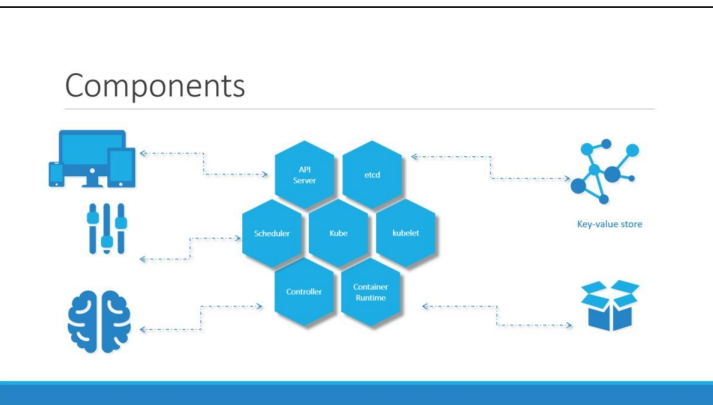
**It was also known as Minions in the past.**

**A cluster is a set of nodes grouped together**

****

**Now we have a cluster, but who is responsible for managing the cluster? Were is the information about the members of the cluster stored? How are the nodes monitored? When a node fails how do you move the workload of the failed node to another worker node? That’s were the Master comes in.**

**The master is another node with Kubernetes installed in it, and is configured as a Master. The master watches over the nodes in the cluster and is responsible for the actual orchestration of containers on the worker nodes.**

****

**When you install Kubernetes on a System, you are actually installing the following components. An API Server. An ETCD service. A kubelet service. A Container Runtime, Controllers and Schedulers**

**The API server acts as the front-end for kubernetes. The users, management devices, Command line interfaces all talk to the API server to interact with the kubernetes cluster**

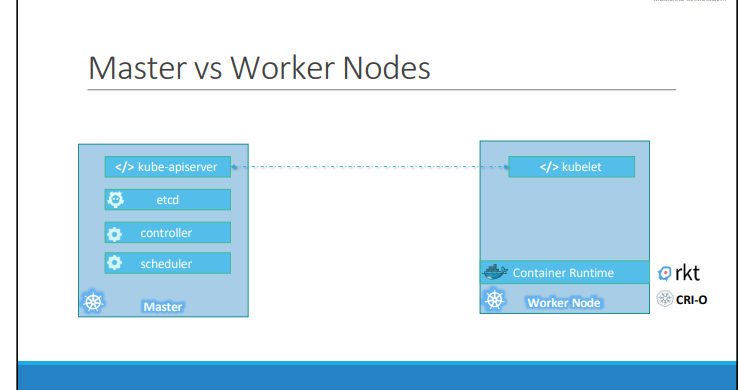
**Next is the ETCD key store. ETCD is a distributed reliable key-value store used by Kubernetes, store all data used to manage the cluster.**

**Think of it this way, when you have multiple nodes and multiple masters in your cluster, etcd stores all that information on all the nodes in the cluster in a distributed manner. ETCD is responsible for implementing locks within the cluster to ensure there are no conflicts between the Masters.**

**The scheduler is responsible for distributing work or containers across multiple nodes. It looks for newly created containers and assigns them to Nodes**

**The controllers are the brain behind orchestration. They are responsible for noticing and responding when nodes, containers or endpoints goes down. The controllers makes decisions to bring up new containers in such cases. The container runtime is the underlying software that is used to run containers. In our case it happens to be Docker.**

**And finally kubelet is the agent that runs on each node in the cluster. The agent is responsible for making sure that the containers are running on the nodes as expected**

****

**So far we saw two types of servers – Master and Worker and a set of components that make up Kubernetes. But how are these components distributed across different types of servers.**

**In other words, how does one server become a master and the other slave? The worker node is were the containers are hosted.**

**For example Docker containers, and to run docker containers on a system, we need a container runtime installed. And that’s were the container runtime falls. In this case it happens to be Docker. This doesn’t HAVE to be docker, there are other container runtime alternatives available such as Rocket or CRIO.**

**The master server has the kube-apiserver and that is what makes it a master. Similarly the worker nodes have the kubelet agent that is responsible for interacting with the master to provide health information of the worker node and carry out actions requested by the master on the worker nodes.**

**All the information gathered are stored in a key-value store on the Master. The key value store is based on the popular etcd framework . The master also has the controller manager and the scheduler. There are other components as well.**

**The command line utilities known as the kube command line tool or kubectl or kube control as it is also called.**

**The kubectl run command is used to deploy an application on the cluster.**

**The kubectl cluster-info command is used to view information about the cluster**

**kubectl get pod command is used to list all the nodes part of the cluster**

**kubectl get nodes**