Kubernetes Architecture

Purpose

Kubernetes eases the burden of configuring, deploying, managing, and monitoring even the largest-scale containerized applications. It also helps IT pros manage container lifecycles and related application lifecycles, and issues including high availability and load balancing.

Worker machine

3 components necessary for every node:

Container runtime (docker): must be installed on every node.

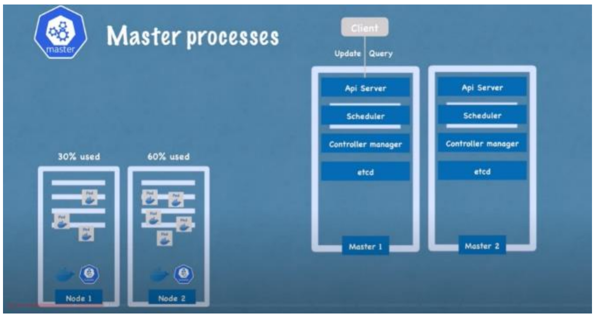
Kubelet: used to configure all the pods – interacts with both the container and node – responsible or taking a configuration and running/starting a pod with a container inside and distributes resources (e.g storage, cpu, ram).

Kube Proxy: forwards the request of services to pods – intelligent forwarding process which makes sure overhead is reduced – sends request to pod inside same node instead of any random replica pod (reduce network overhead of sending request to another machine).

Master processes

How do we Schedule pods/Monitor pod death/Reschedule/restart pod/Join a new node? Master nodes are responsible for these processes

>Clusters are often made up of multiple masters – each master node runs its master processes where the Api server is load balanced and there is distributed storage across all master nodes (etcd)


Master processes

4 processes run on every master node:

API server (cluster gateway): when new API deploy we interact with API server through client (using UI/command line/API) to communicate updates or Querys (cluster gateway). The Api server acts as a gatekeeper for authentication.

Some request -> API server -> validates request -> other processes -> pod

Scheduler: Scheduler decides which node new pods should be scheduled. It observes how much resources a pod you want to schedule will need (cpu, ram, storage) it will then Identify how much resources are available on the workers nodes and schedule pod on node with most available resources.

Schedule new pod -> API server -> scheduler -> Kubelet

Controller Manager: When pods die there must be a way to detect its inactivity and reschedule them as soon as possible. Controller managers detect cluster state changes and tries to restore the cluster as soon as possible.

Controller manager -> scheduler-> Kubelet

Etcd: etcd is known as the clusters brain – cluster changes get stored in the key value store. It tells the schedular what resources are available, if states have changed within a cluster, in a query if the cluster is healthy

>application data is not stored on the etcd