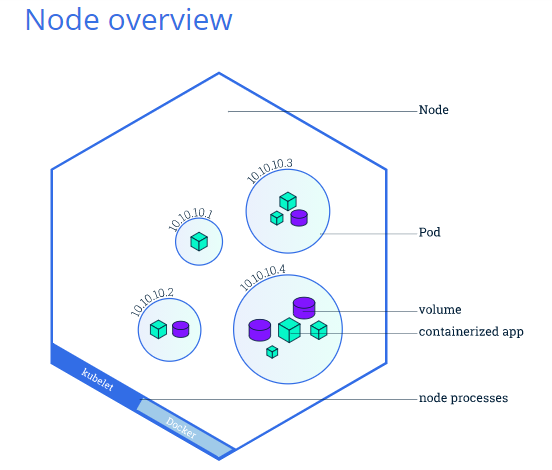
Kubernetes:

* Pods - Similar to an ECS task - its a logical grouping of containers, with shared context such as networking, storage & info on how to run each container within it.
* Node - instance/server where the Pods are launched. Each node is maintained by the master and contains two things:

1. Kubelet: kind of a controller that runs on the node and manages communication between the master and the node, managing the pods and containers running on the node.
2. A container runtime – like Docker, that is responsible for pulling images, unpacking and running the containerized application.



The most common operations can be done with the following kubectl commands:

* **kubectl get** - list resources
* **kubectl describe** - show detailed information about a resource
* **kubectl logs** - print the logs from a container in a pod
* **kubectl exec** - execute a command on a container in a pod

Overview of Kubernetes Services

A Service in Kubernetes is an abstraction which defines a logical set of Pods and a policy by which to access them. Services enable a loose coupling between dependent Pods. A Service is defined using YAML [(preferred)](https://kubernetes.io/docs/concepts/configuration/overview/#general-configuration-tips) or JSON, like all Kubernetes objects.

Although each Pod has a unique IP address, those IPs are not exposed outside the cluster without a Service. Services allow your applications to receive traffic. Services can be exposed in different ways by specifying a type in the ServiceSpec:

* ***ClusterIP*** (default) - Exposes the Service on an internal IP in the cluster. This type makes the Service only reachable from within the cluster.
* ***NodePort*** - Exposes the Service on the same port of each selected Node in the cluster using NAT. Makes a Service accessible from outside the cluster using <NodeIP>:<NodePort>. Superset of ClusterIP.
* ***LoadBalancer*** - Creates an external load balancer in the current cloud (if supported) and assigns a fixed, external IP to the Service. Superset of NodePort.
* ***ExternalName*** - Exposes the Service using an arbitrary name (specified by externalName in the spec) by returning a CNAME record with the name. No proxy is used. This type requires v1.7 or higher of kube-dns.

### Services and Labels.

Services match a set of Pods using [labels and selectors](https://kubernetes.io/docs/concepts/overview/working-with-objects/labels), a grouping primitive that allows logical operation on objects in Kubernetes. Labels are key/value pairs attached to objects.

