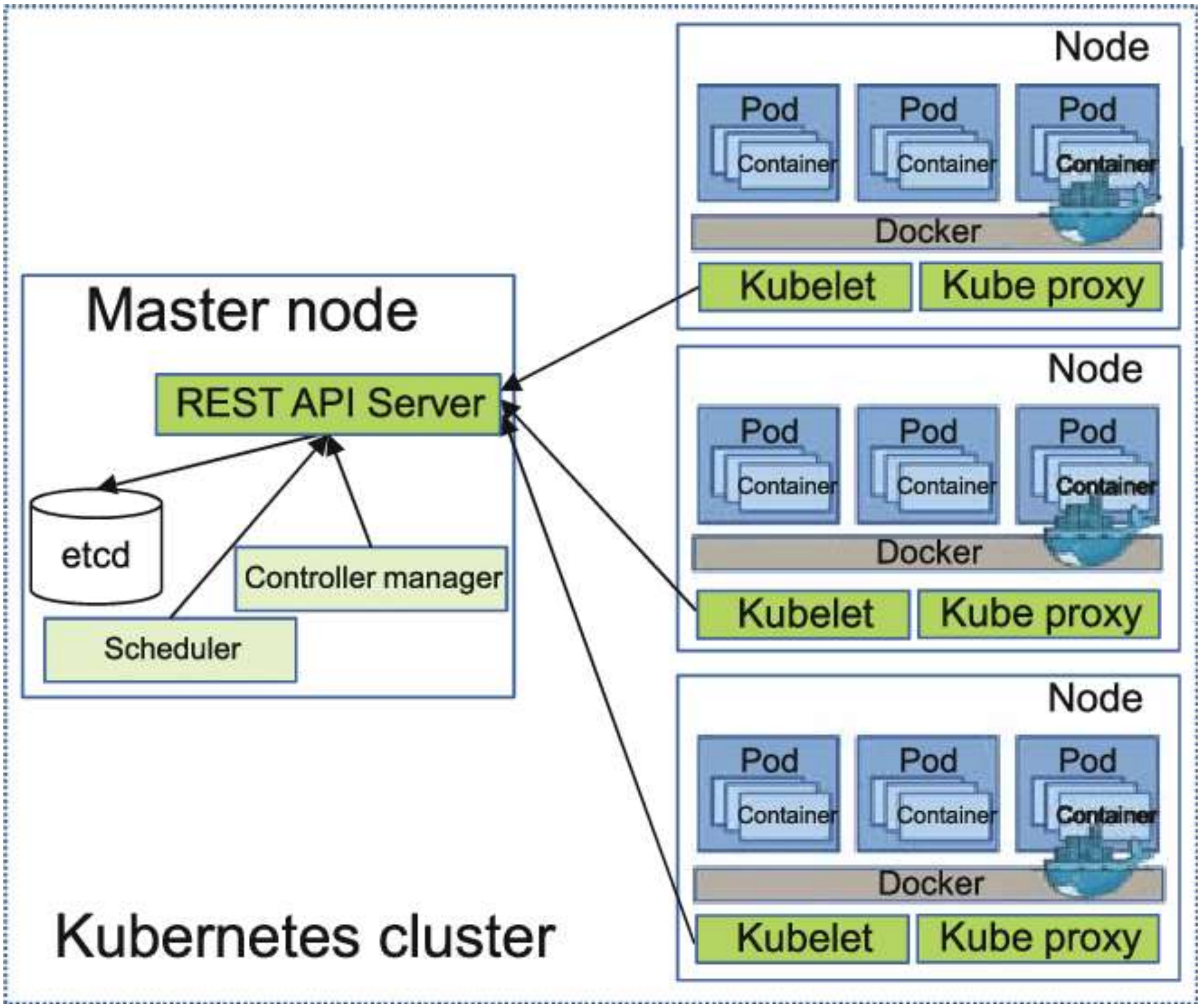
Kubernetes architecture



Kubernetes, often abbreviated as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. The architecture of Kubernetes is designed to provide a flexible and scalable framework for deploying and managing containerized workloads across a cluster of machines. Here are the key components of Kubernetes architecture:

1. **Master Node:**

- **API Server:** Acts as the entry point for the Kubernetes cluster. It exposes the Kubernetes API, which is used to manage the cluster.

- **Controller Manager:** Ensures the desired state of the cluster by managing different controllers responsible for maintaining the correct number of pods, deployments, and other resources.

- **Scheduler:** Assigns workloads (pods) to worker nodes based on resource availability and constraints.

2. **Worker Node (Minion):**

- **Kubelet:** Ensures that containers are running in a Pod. It communicates with the master node and manages the containers on the node.

- **Container Runtime:** The software responsible for running containers, such as Docker or containerd.

- **Kube Proxy:** Maintains network rules on nodes, enabling communication between different pods and external traffic.

3. **etcd:**

- A distributed key-value store that stores the configuration data of the Kubernetes cluster. The data includes information about cluster state, configurations, and metadata.

4. **Pod:**

- The smallest deployable unit in Kubernetes. A Pod represents a single instance of a running process in a cluster, and it can contain one or more containers that share the same network namespace.

5. **Controller:**

- Controllers manage the lifecycle of pods and other resources, ensuring that the desired state is maintained. Examples include ReplicaSet, Deployment, StatefulSet, and DaemonSet.

6. **Service:**

- Defines a set of pods and a policy for accessing them. Services enable communication between different sets of pods and provide a stable endpoint for external access.

7. **Namespace:**

- A way to divide cluster resources among multiple users or projects. It provides a scope for names, and each namespace can have its own set of resources without interfering with others.

8. **Label and Selector:**

- Labels are key-value pairs attached to resources (e.g., pods). Selectors are used to identify a group of resources based on their labels, allowing for easy grouping and management.

9. **ConfigMap and Secret:**

- ConfigMaps hold configuration data, while Secrets store sensitive information. Both can be mounted into pods as volumes or used as environment variables.

Understanding these components helps in grasping the overall structure of a Kubernetes cluster and how it manages containerized applications efficiently. Kubernetes, often abbreviated as K8s, is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. The architecture of Kubernetes is designed to provide a flexible and scalable framework for deploying and managing containerized workloads across a cluster of machines. Here are the key components of Kubernetes architecture: