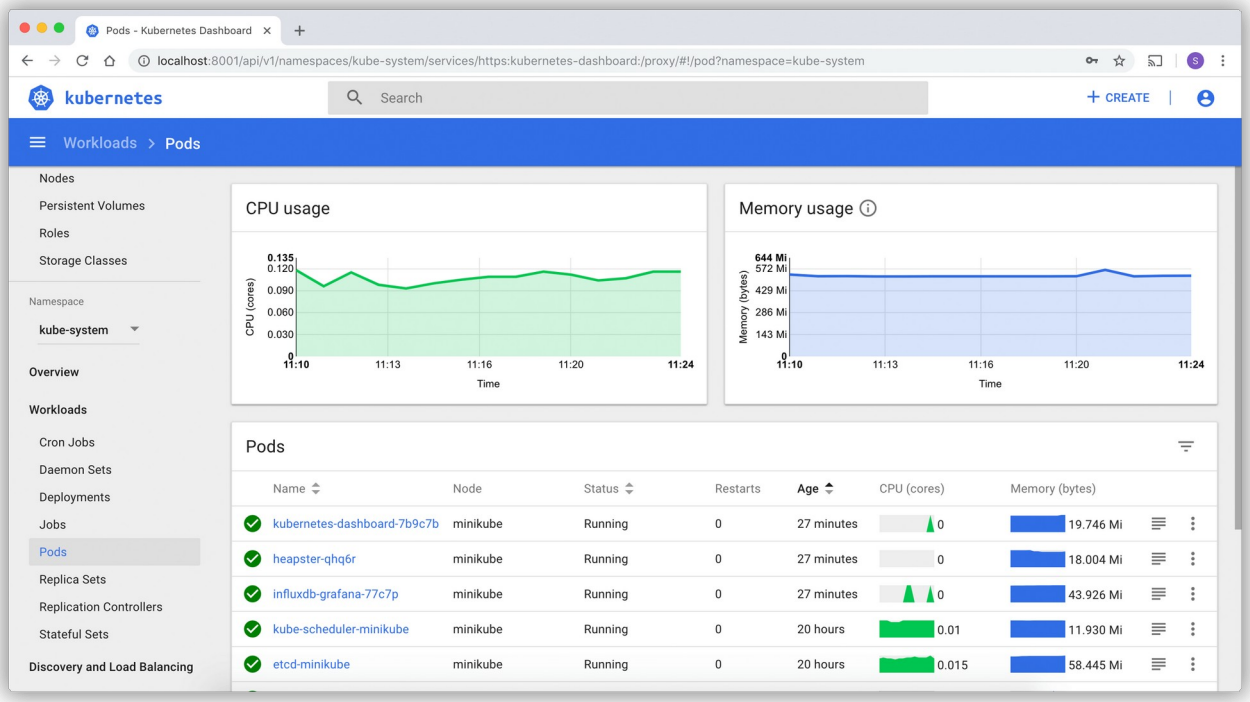
Monitoring

Why Monitoring?

To scale an application and provide a reliable service, you need to understand how the application behaves when it is deployed. You can examine application performance in a Kubernetes cluster by examining the containers, pods, services and the characteristics of the overall cluster. Kubernetes provides detailed information about an application's resource usage at each of these levels. This information allows you to evaluate your application's performance and where bottlenecks can be removed to improve overall performance.

In Kubernetes, application monitoring does not depend on a single monitoring solution. On new clusters, you can use resource metric pipelines to collect monitoring statistics.

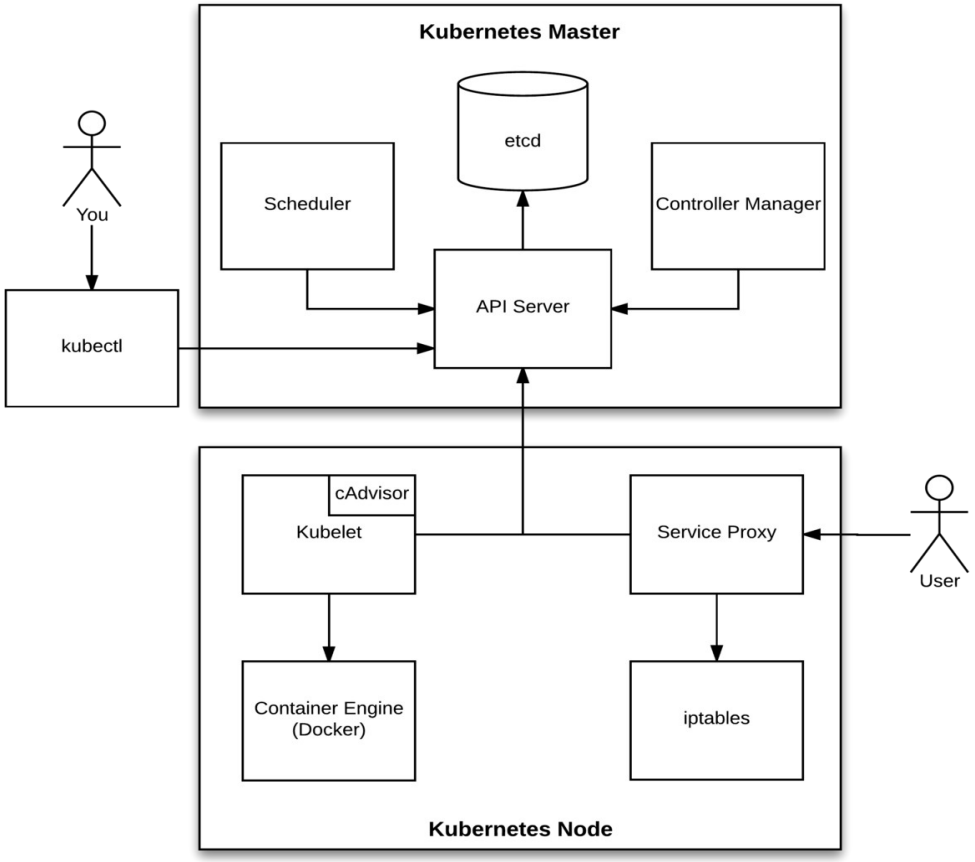
At the infrastructure level, a Kubernetes cluster is a set of physical or virtual machines acting in a specific role. The machines acting in the role of Master act as the brain of all operations and are charged with orchestrating containers that run on all of the Nodes.



Master Components

Master components manage the lifecycle of a pod, the base unit of deployment within a Kubernetes cluster. If a pod dies, the Controller creates a new one. If you scale the number of pod replicas up or down, the Controller creates or destroys pods to satisfy your request. The Master role includes the following components:

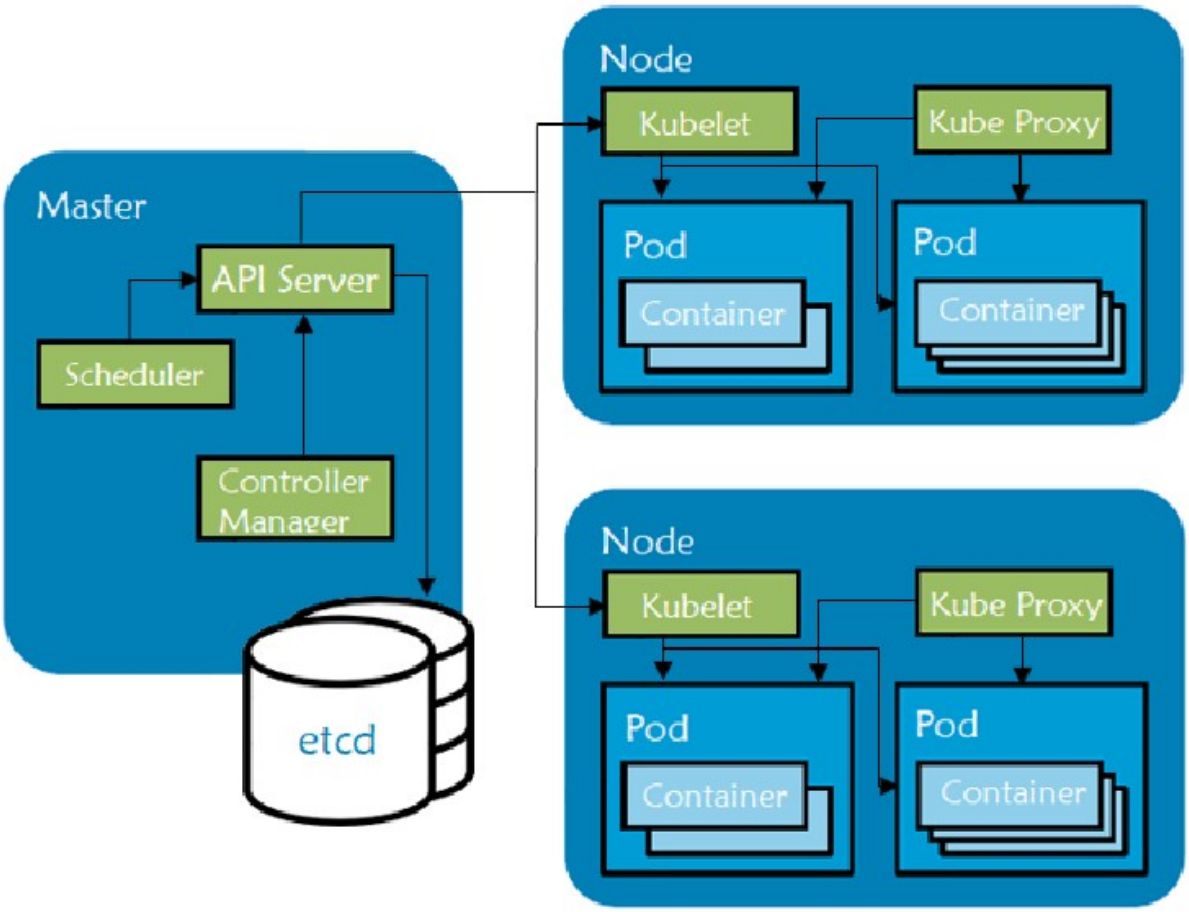
- Kube-apiserver - exposes APIs for the other master components.
- Etcd -a consistent and highly-available key/value store used for storing all internal cluster data.
- Kube-scheduler - uses information in the Pod spec to decide on which Node to run a Pod.
- Kube control manager - responsible for Node management (detecting if a Node fails), pod replication, and endpoint creation.
- Cloud controller manager - runs controllers that interact with the underlying cloud providers.



Node Components

Node components are worker machines in Kubernetes and are managed by the Master. A node may be a virtual machine (VM) or physical machine, and Kubernetes runs equally well on both types of systems. Each node contains the necessary components to run pods:

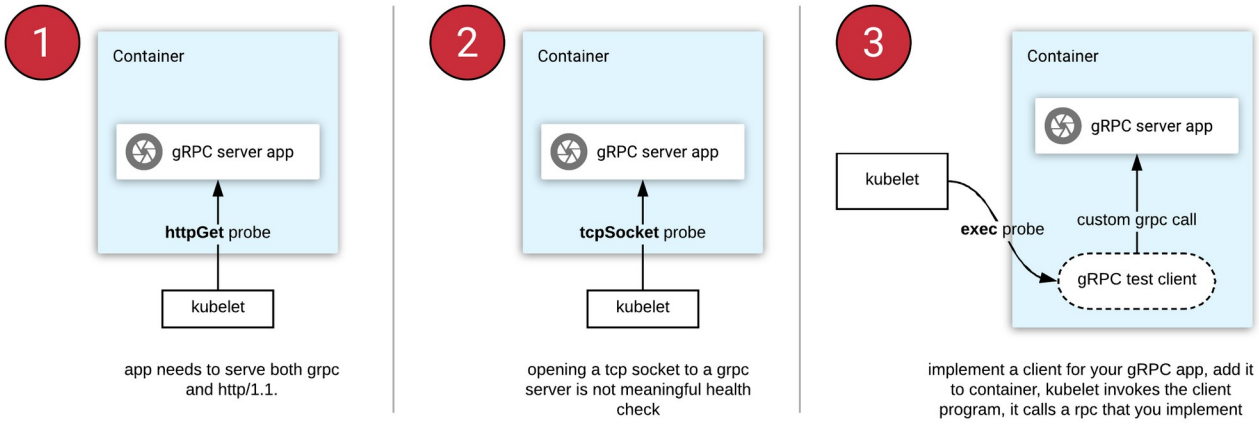
- Kubelet - handles all communication between the Master and the node on which it is running. It interfaces with the container run time to deploy and monitor containers.
- Kube proxy - maintains the network rules on the host and handles transmission of packets between pods, the host, and the outside world.



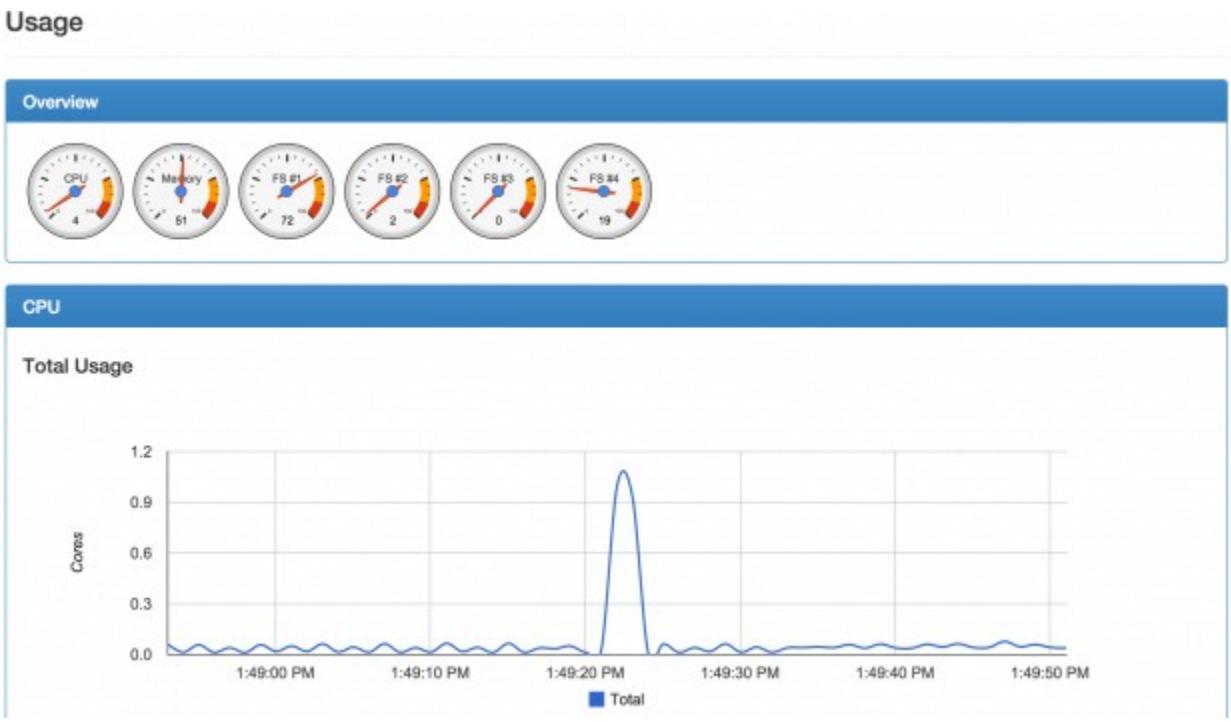
Built in Monitoring

Kubernetes offers built in monitoring features that can then be tied into Prometheus or Grafana for more friendly user interfaces. The main components are as follows:

Probes actively monitor the health of a container. If the probe determines that a container is no longer healthy, the probe will restart it.



cAdvisor is an open source agent that monitors resource usage and analyzes the performance of containers. Originally created by Google, cAdvisor is now integrated with the Kubelet. It collects, aggregates, processes and exports metrics such as CPU, memory, file and network usage for all containers running on a given node.



The Kubernetes Dashboard is an add-on which gives an overview of the resources running on your cluster. It also gives a very basic means of deploying and interacting with those resources.