KUBERNETES ADMINISTRATION

The Need for Orchestration Systems

While Docker provided an open standard for packaging and distributing containerized applications, there arose a new problem.

> How would all of these containers be coordinated and scheduled?

>How do you seamlessly upgrade an application without any interruption of service?

> How do you monitor the health of an application, know when something goes wrong and seamlessly restart it?

Life of an application

Container Orchestration tools ?

Apache
MESOS™

kubernetes

Kubernetes ?

Kubernetes is the container orchestrator that was developed at Google which has been donated to the CNCF and is now open source.

It has the advantage of leveraging Google’s years of expertise in container management.

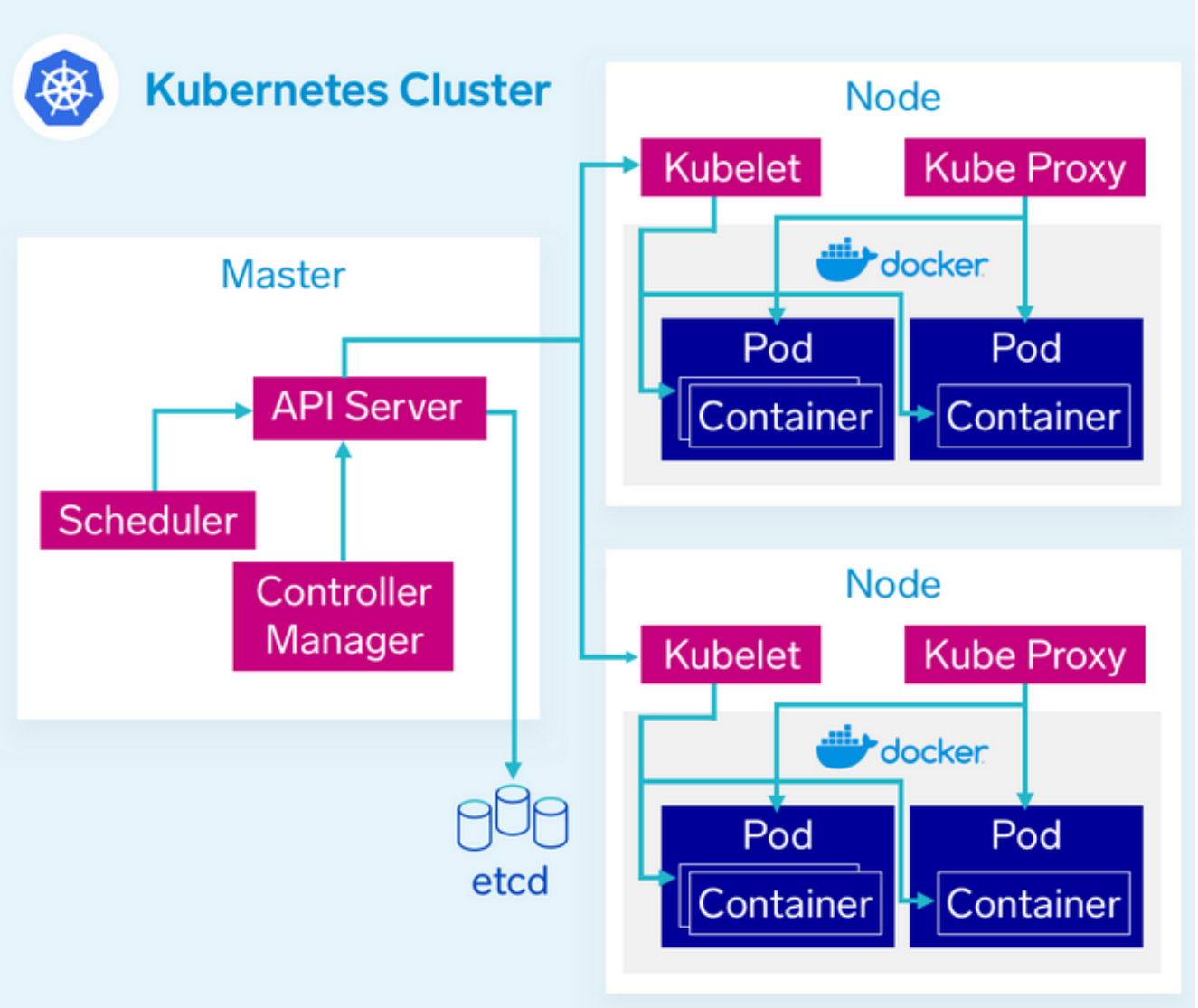
It is a comprehensive system for automating deployment, scheduling and scaling of containerized applications, and supports many containerization tools such as Docker.

Setting up Kubernetes Cluster

Different Kubernetes solutions meet different requirements: ease of maintenance, security, control, available resources, and expertise required to operate and manage a cluster.

You can deploy a Kubernetes cluster on a local machine, cloud, on-prem datacenter, or choose a managed Kubernetes cluster. You can also create custom solutions across a wide range of cloud providers, or bare metal environments.

Architecture of Kubernetes



Hardware Requirements

- **Master node’s minimal required memory is 2GB and the worker node needs minimum is 1GB**
- **The master node needs at least 1.5 and the worker node need at least 0.7 cores.**

Installation Step on all nodes

Follow the steps mentioned on

https://github.com/vsaini44/KubernetesRepo.git

> You can clone the repo or copy paste the contents of the files.