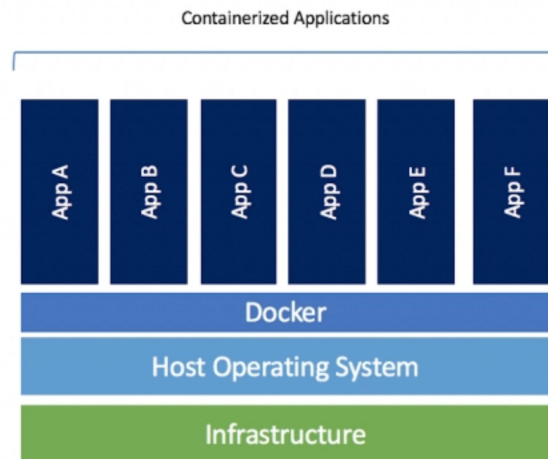


Kubernetes 101:

Container: Unit of software that packages up code and all its dependencies (Applications jar file, Lib, tools, etc.) into isolated environments (own processes or services, their own networking interface) with access to an operating system kernel that it shares with other containers and little or no access between them.

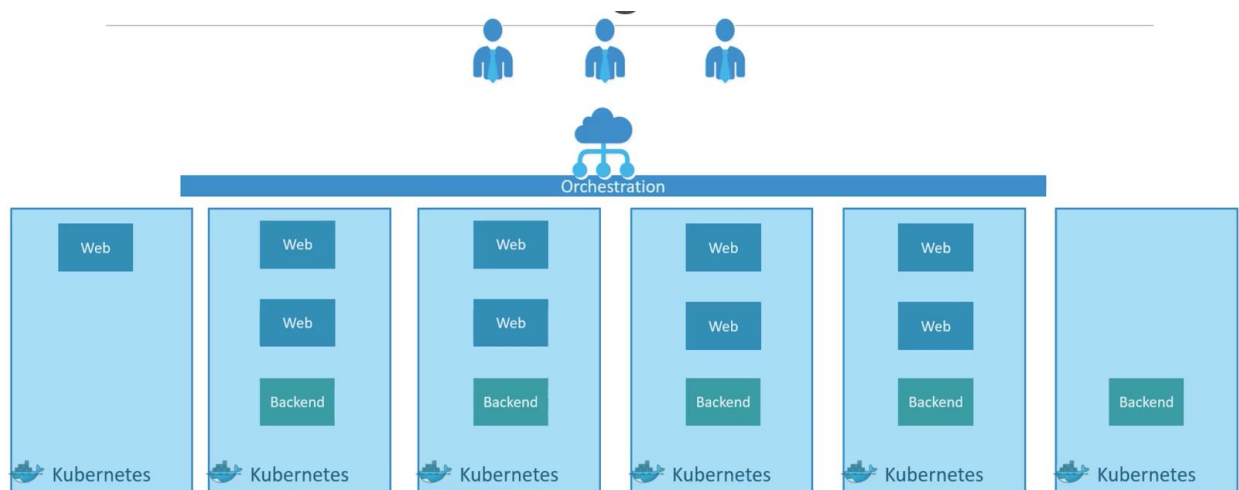


Kubernetes is a container **orchestration tool**.

Container orchestration:

The process of automatically deploying and managing containers is known as container orchestration.

- a) Managing means maintaining a connection between containers such as DB, messaging services or any backend services.
- b) Automatically scale up or down based on the load (number of users or any other processes).
- c) Restart failed containers and many more.

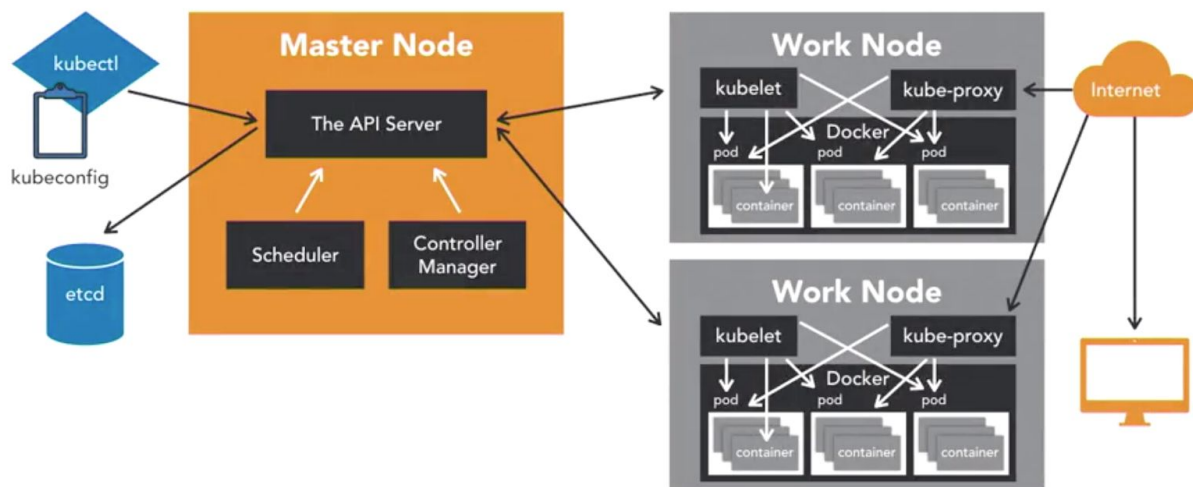


keywords in Kubernetes:

A **Node** is a machine in Kubernetes. It might be a laptop, for instance running on a cloud.

The collection of Nodes is nothing but a **Cluster**. Each cluster consists of at least one cluster **master node**.

A **Node** can have multiple **Pods**(Pods are nothing but a set of containers)



Above image is Kubernetes cluster architecture.

Components of Kubernetes:-

- 1) **API Server**: Acts as a front end to the service. Users talk to the API server. via CLI, UI.
- 2) **Etcd**: Key-value store to save all the data needed for managing clusters.
- 3) **Scheduler**: It is responsible for distributing work, such as it looks for newly created containers and assign them to nodes.
- 4) **Controller Manager**: They are responsible for noticing and responding when a node or container goes down. This controller makes decisions on whether to bring up a node back again or not.

- 5) **Container runtime:** Underlying software that runs container software (Docker, rkt).
- 6) **kublet:** It is the agent that runs in each node in the cluster. It is responsible for making sure that the containers are running on the node as expected or not.