**KUBERNETES**

Configuration management: Maintain all the change in infrastructure with the help of software.

Before Vms: Mutable

What happened earlier- We had virtual machine, immutable, wastage of memory,

Previously there were no **package managers,** these packages all the requirements for the project,

Later it is solved by using containers.

One application on one VM, this is not scalable and good so user Docker.

**Now Managing containers,**

*Monolithic architecture: Does monolithic applications means that the frontend, backend and DB are made a single container and not like we find in compose files.??*

* Monolithic applications: Frontend , backend and DB all together
* Micro services applications: Frontend , backend and DB all separate.

How to make different containers communicate with each other as they are collectively a part of single application?

**Orchestrators:** Help in deploying and managing applications dynamically. **Kubernetes** help us to do this. Managing all the microserves

* Deploy
* Zero downtime updates
* Scale
* Heal containers

These are collectively application of **cloud native applications**. These meet the modern nosiness demand that follow above points.

Kubernetes provides us more features additional to orchestrators.

**Need of Kubernetes:**

AWS offered cloud,

Open stack is open source alternative to AWS

Google were internally running **Borg and Omega**, then google made **Kubernetes or K8s**

Open source in 2014 and donated to CNCF, cloud native computing foundation.

Docker is containerizes our application it’s a CRI, Container Runtime Interface

Kubernetes allow us to use multiple CRI,

**Architecture of Kubernetes:**

**Cluster:** Collection of Control Plane (master node) + Worker Nodes (Vms or can be assumed as servers )

**KubCTL**: Comminicated with Control Plane, it is a CLI tool.

**Kubernetes Command line tool**

Worker Node

Container Runtime

Kubernetes Pod

Container

**Objects:**

**Pods:** Scheduling unit in kuberneters.

**Steps to running applcaiton in KUberneted:**

* Create microservice of your application
* Containize the microservices.
* Every container in PODs
* Deploy these pods in controllers in Kubernneeter