Kubenertes -

Derived from Greek Word for "Pilot" / "Helmsman" implying kubernetes container (Maritime Theme of Docker)

orchestration capabilities and administration.

Internal Google Project - Borg (~15 years) ref: https://ai.google/research/pubs/pub43438

Focus : Decoupled, Transient Services

Decoupled meaning everything has been designed to not require anything else in particular

Transient meaning the whole system expects various components to be terminated or replaced

To achieve Flexibility and Scalability

Encourage migration from Monolithic to Micro Services Approach for above mentioned pros

Configuration is stored in JSON format, but written in YAML

Kubenertes agents convert the YAML to JSON prior to persistence to the database

Kubernetes is written in Go Language

Kubernetes to address the following:

- Container Deployments

- Base Infrastructure Management

- Fail Over / Self Heal

- Rolling Updates and Rollbacks

- Tear down unused resources

- Network Management ( Overlay Networks )

- Storage Management

- Supports Chaos Engineering :)

Alternatives to Kubernetes:

- Docker Swarm , based on SwarmKit , Embedded with Docker

- Apache Mesos- Data center scheduler, uses frameworks. Marathon is a container orchestrator

- Nomad from hashicorp, makers of Vagrant and Consul. Nomad schedules tasks in Jobs, has a docker driver to define running container

- Rancher Container Orchestrator-agnostic system ( One stop interface to manage applications)

supports mesos , swarm, k8s

Google contributed cgroups to Linux Kernel in 2007

**cgroups** and **Linux namespaces** are at the heart of the containers

<https://www.slideshare.net/kerneltlv/namespaces-and-cgroups-the-basis-of-linux-containers>

Address 12 Factor Principles

https://aws.amazon.com/blogs/compute/applying-the-twelve-factor-app-methodology-to-serverless-applications/



Every node running a container would have **kubelet** and **kube-proxy**

Kubernetes has a master and worker nodes called minions. Master runs the API server, a scheduler, various controllers and a storage system to keep state of the cluster, container settings and network config

Objects and state of cluster is stored in **etcd**

**Kube API or kubectl – to manage/administer k8s clusters**

**Kube-scheduler** receives API requests for running containers, finds suitable node to run that container

**Kubelet**: receives requests to run the containers, manages any necessary resources and watches over them for fail over/self-heal. Interacts with local container engine, docker by default ( rkt and cri-o are alternatives to docker)

**Kube-proxy**: creates and manages networking rules to expose the container to overlay network

*Containers are not managed individually*, managed as PODS

A pod consists of one or more containers which share an IP address, access to storage and namespace. Typical setup: one container in a pod runs applications while other containers support primary application (Logging, monitoring, tracing etc.,)

Orchestration is managed through series of watch-loops also called **CONTROLLERS.** Each controller communicates with **kube-apiserver** for object state, modifying the object until declared state matches current state. Controllers are compiled into **kube-controller-manager**.

Default,newest, feature filled controller for containers is **DEPLOYMENT**. Ensures resources declared in PodSpec such as IP address, storage are available. Then deploys a **REPLICASET**

**ReplicaSet** controllers: deploys and restarts pods, declared to container engine to spawn or terminate containers matching requested running instances count. **Jobs** or **CronJobs** to handle single or recurring tasks

To tag and manage multiple pods: **labels.** They become part of Object metadata. Nodes can have **taints** to discourage Pod Assignments (similar to ETHOS\_ROLE LIKE in tag we use in DC/OS ). The **taint** config can be overridden with **toleration** tag in Pod Metadata.

Object Metadata also supports **annotations** to be used by third party information (Eg: Adobe Cost Center)

<https://github.com/kubernetes/kubernetes/>