

---

---

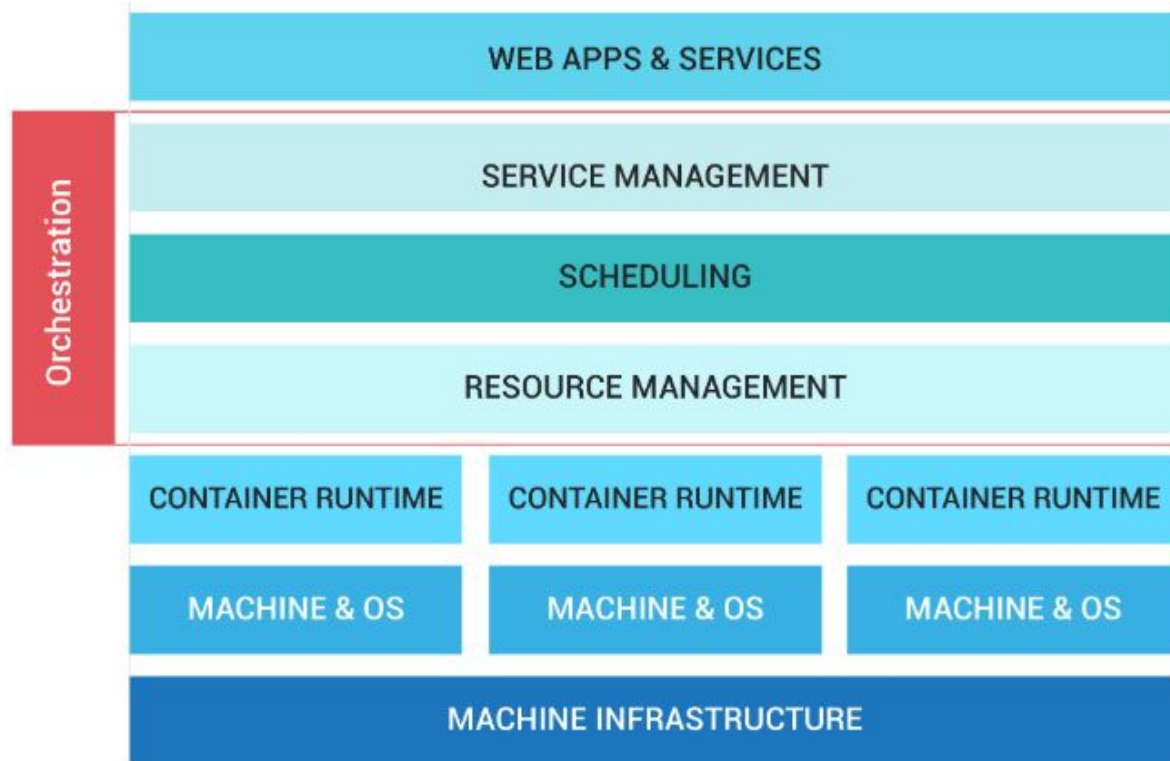
# Kubernetes

Ritesh Goyal

---

---

# Container Orchestration



# What is Kubernetes?

1. Deployment of containerized application.
2. Scaling of containerized application.
3. Management of containerized application.

Kubernetes enables to:

1. Run multiple containers on a single machine.
2. Schedule containers on cluster of machines.
3. Run long running services such as web applications.



**kubernetes**

# Features Of Kubernetes

## Automatic bin-packing

Automatically places containers based on their resource requirements and other constraints, while not sacrificing the availability.

## Self-Healing

Automatically restarts containers that fail, replaces and reschedules containers when nodes die, kills containers that don't respond.

## Storage-Orchestration

Automatically mount the storage system of your choice, whether from local storage or a public cloud provider such as GCP, AWS.

01



03



05



02



04



## Horizontal Scaling

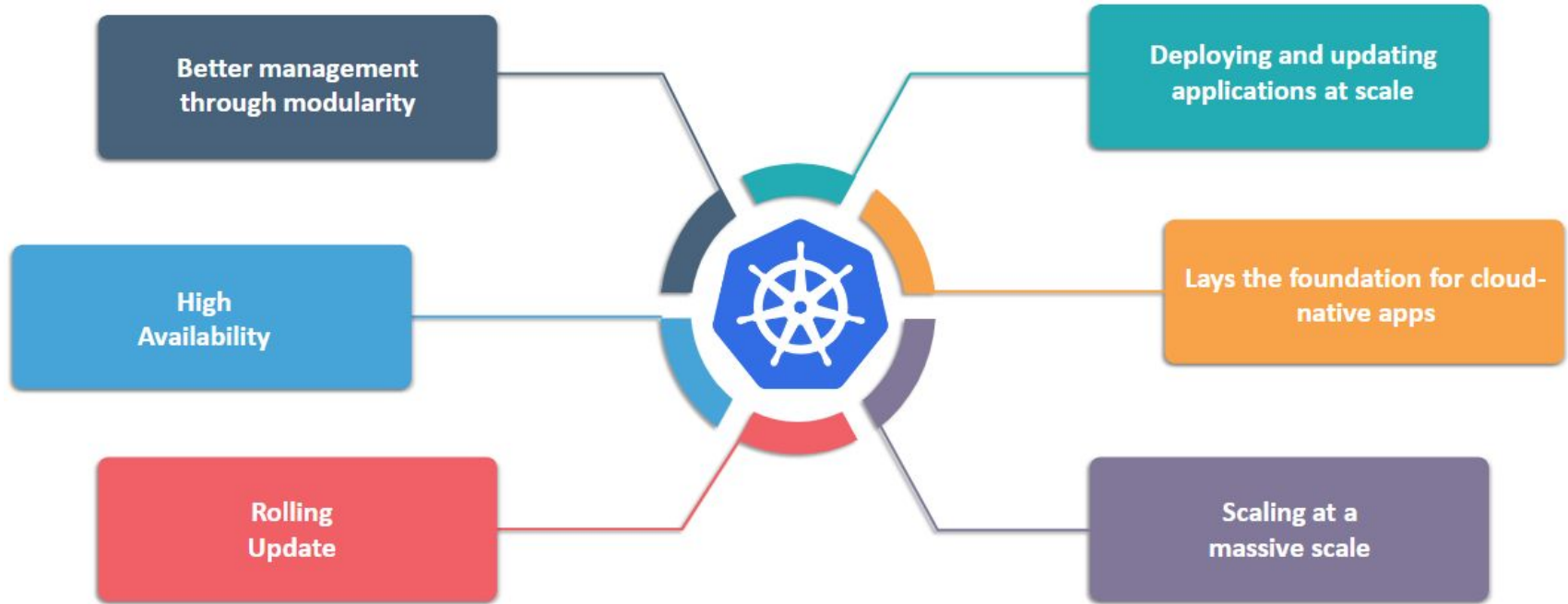
Scale your application up and down with a simple command, with a UI, or automatically based on CPU usage.

## Batch Execution

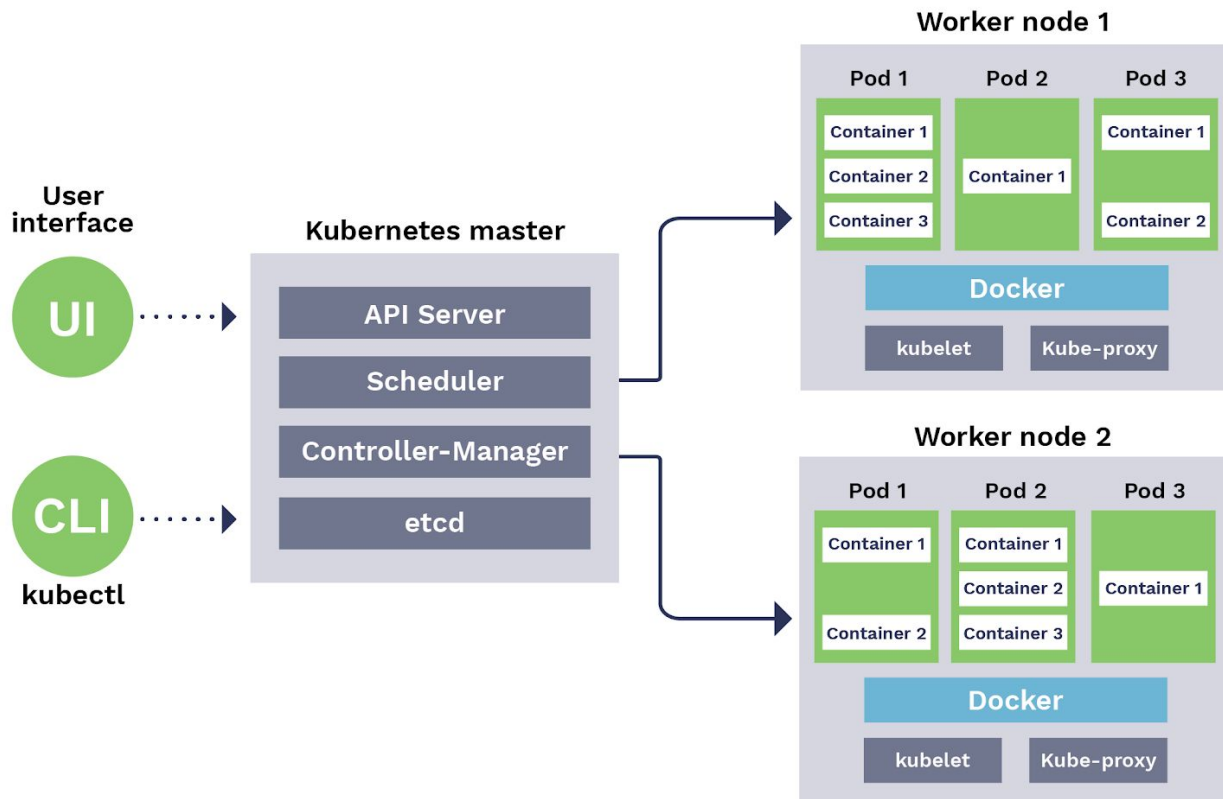
In addition to services, Kubernetes can manage your batch and CI workloads as well as replacing containers that fail, if needed.



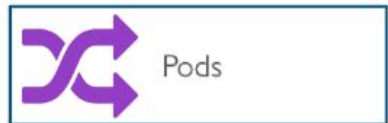
# Advantages of Kubernetes



# Kubernetes Cluster Architecture



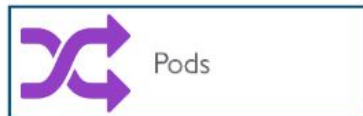
# Kubernetes Master



Kubernetes master is a collection of three processes:

- **Kube-apiserver** : It validates and configures all the data for the API objects which include pods, services, replica controllers, and others
- **Kube-controller-manager** : It's a daemon that includes the non terminating loops (that regulates the state of the system) shipped with Kubernetes
- **Etcd** : It is a distributed key-value store designed to reliably and quickly preserve and provide access to critical data
- **Kube-scheduler** : The Kubernetes scheduler is a workload-specific function that significantly impacts availability, performance, and capacity. Workload-specific requirements will be exposed through the API as required

# Kubernetes Worker Node



Worker node in the cluster runs two processes:

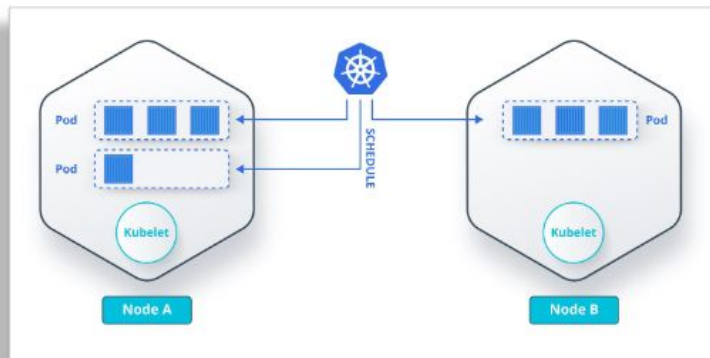
- **Kubelet** : It's a foremost node agent running on each node works under the terms of PodSec. A PodSpec is a YAML or JSON object that describes a pod
- **Kube-Proxy** : Kubernetes network proxy runs on each node
- **Container Runtime** : The container runtime is the software that is responsible for running containers. Kubernetes supports several runtimes: Docker, rkt, runc and any OCI runtime-spec implementation.



# Kubernetes Pods



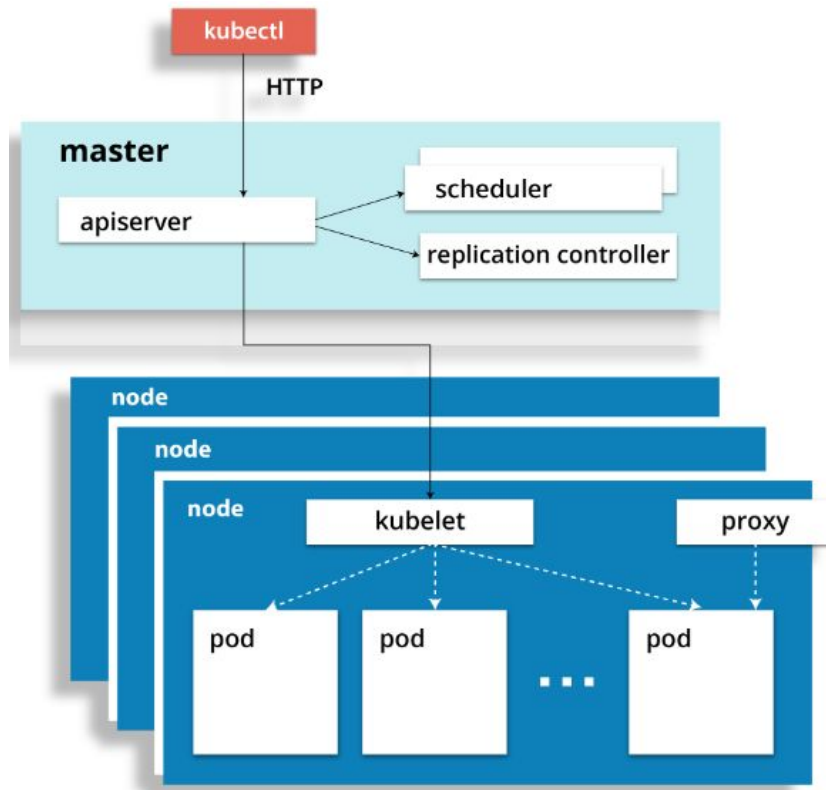
- Containers are deployed and scheduled through Kubernetes in a group called pods
- These are tightly coupled containers i.e the applications running on them are dependent on each other
- 1 to 5 tightly coupled containers can be stored in a pod that collaborate to provide a service



# Kubecttl

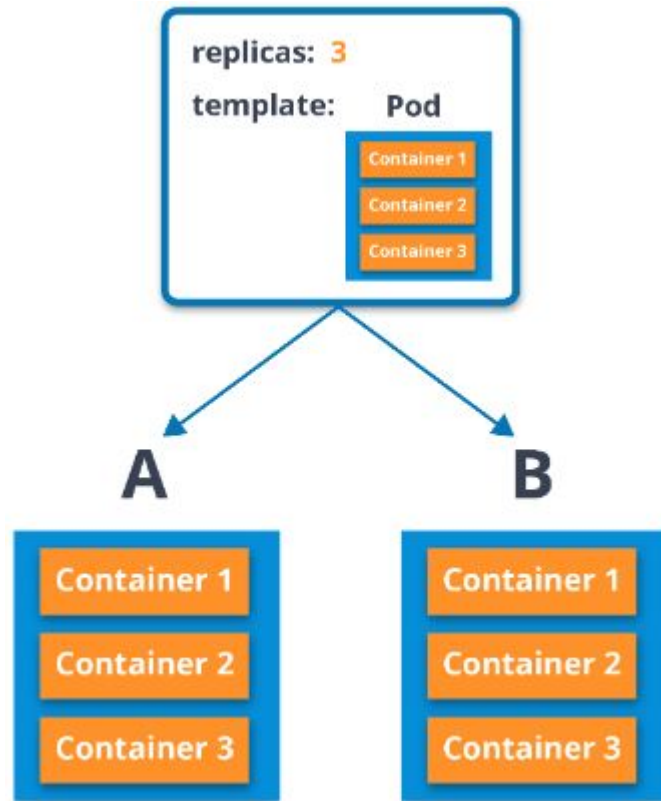
It is a Kubernetes command-line tool which is used to deploy and manage applications on Kubernetes. It helps inspecting the Kubernetes cluster resources.

By using Kubecttl, we can create, delete and update commands on Kubernetes cluster.



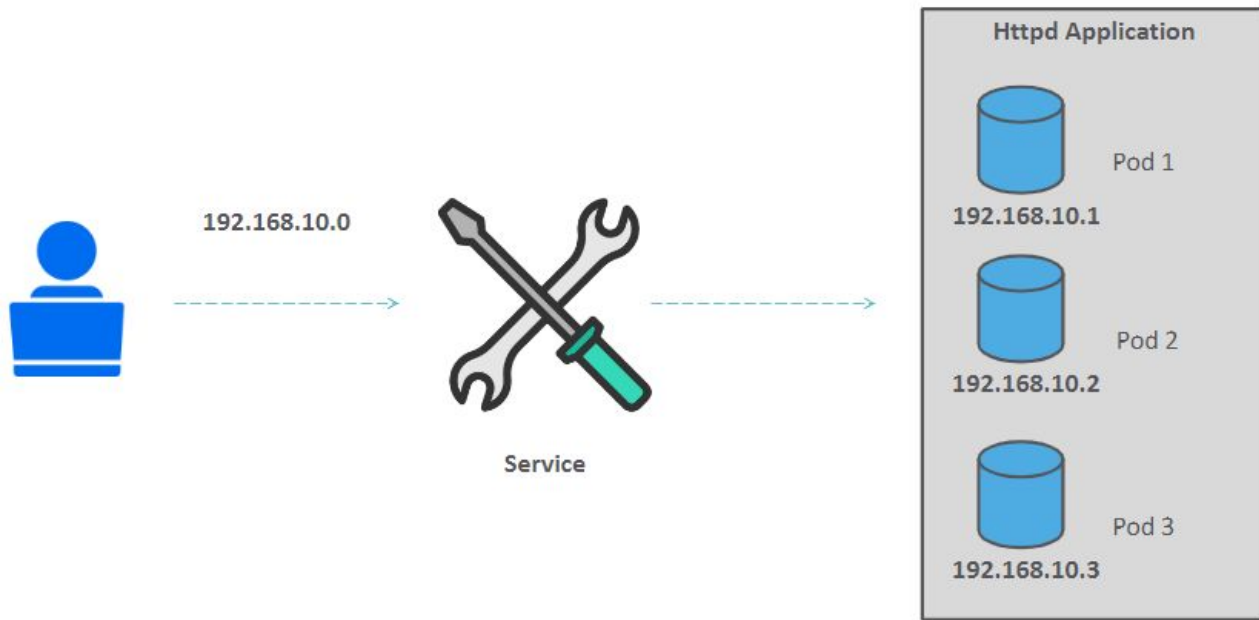
# Replica Set

A Replica-Set makes sure that started number of pod replicas are running at any instant of time. It can be scaled up or down by just updating the `.spec.replicas` field



# Services in Kubernetes

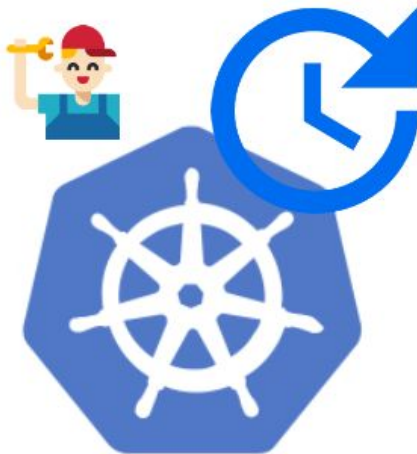
Services act like load balancer in Kubernetes, they also have an IP address. This IP address automatically routes to healthy pod. In case, the pod becomes unhealthy the service automatically routes to next healthy pod. Hence, with this user will interact with only one IP address.



# Rolling Updates in Kubernetes

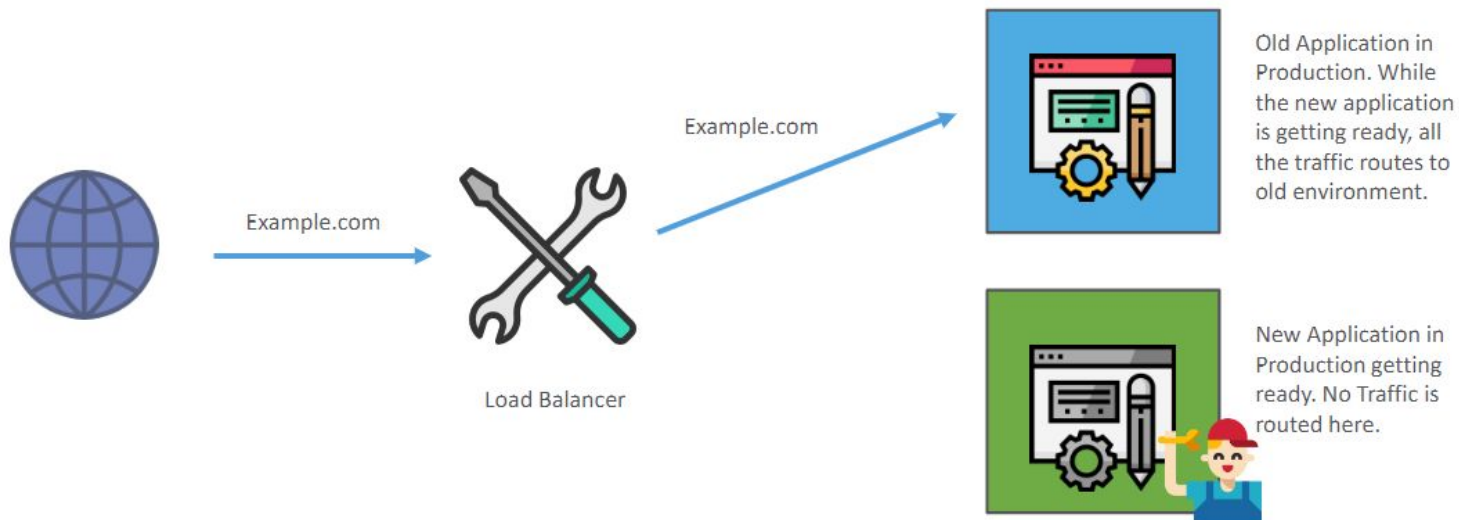
A rolling update is the process of updating an application - whether it is a newer version or just updated configuration - in a serial fashion. By updating one instance at a time, you are able to keep the application up and running.

Rolling update in Kubernetes follows the Blue Green Deployment Model.



# Blue Green Deployment Model

Blue Green deployment is a technique that reduces downtime and risk by running two identical production environments called Blue and Green.



# Blue Green Deployment Model (Cont..)

As soon as the new application is ready, the traffic is routed to the new application in the green environment, hence no downtime!

