

### Basics

**Define** :it is a container management technology developed by GOOGLE (later made open source in 2015) to manage containerized application(orchestration).

### Why

- |  |                                  |
|--|----------------------------------|
| 1.Service discovery and load balancing | 2.Automated rollbacks            |
| 1.Service discovery and load balancing | 3.Self-healing                   |
| 3.Self-healing                         | 4.Auto Scaling                   |
| 5.Canary updates and Rolling updates   | 6.Open source & Community driven |
| 7.High Availability                    |                                  |

### Kubernetes Concepts

<b>Node</b>	machine in the cluster
<b>Docker</b>	helps in creation of containers that includes apps and its binaries.
<b>Pods</b>	A Pod is the basic building block of Kubernetes—the smallest and simplest unit in the Kubernetes object model that you create or deploy, is also a group of containers (1 or more).Only containers of same pod can share shared storage.
<b>Service</b>	is an abstraction which defines a logical set of Pods and a policy by which to access them.
<b>Jobs</b>	Creates pod(s) and ensures that a specified number successfully completed.When a specified number of successful run of pods is completed, then the job is considered complete.
<b>Cronjob</b>	job scheduler in K8s
<b>Repl-casets</b>	ensures how many replica of pod should be running.

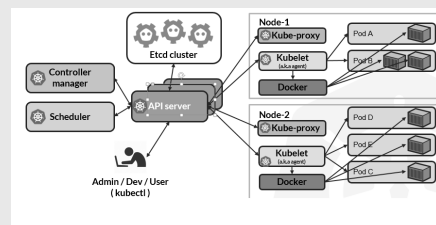
### Kubernetes Concepts (cont)

<b>Namespaces</b>	Logical separation between teams and their environments.It allows various teams(Dev,Prod) to share k8s cluster by providing isolated workspace.
<b>Deployment</b>	Desired state of pods for declarative updates
<b>daemonset</b>	ensures a particular pod to be run on some or all nodes
<b>persistent-volume</b>	Persistent storage in the cluster with an independent lifecycle.
<b>persistent-volum-claim</b>	Request for storage (for a PersistentVolume) by a user
<b>ingress</b>	An Ingress is a collection of rules that allow inbound connections to reach the cluster services.

### Namespaces

<code>kubectl get all --all-namespaces</code>	shows all objects of all namespaces
<code>kubectl get pods --all-namespaces</code>	shows pods of all namespaces
<code>kubectl get pods -n &lt;name&gt;</code>	shows pods of a namespace eg.kubectl get all -n kube-system :shows objects of system namespace
<code>kubectl get namespaces</code>	(show all namespaces)
<b>default</b>	:default name spaces of user
<b>kube-public</b>	:Namespace for resources that are publicly available/readable by all
<b>kube-system</b>	:Namespace for objects/resources created by Kubernetes systems

### Components Architecture Diagram



By **Gaurav Pandey**  
(gauravpandey44)

Published 30th September, 2019.  
Last updated 1st October, 2019.  
Page 1 of 2.

Sponsored by **Readable.com**  
Measure your website readability!  
<https://readable.com>

### Components Basic

#### Components in Manager Nodes:

**Controller manager:** Runs various controllers to help move running state to desired state.

*Node Controller:* Responsible for noticing and responding when nodes go down.

*Replication Controller:* Responsible for maintaining the correct number of pods for every replication controller object in the system.

*Endpoints Controller:* Populates the Endpoints object (i.e, joins Services & Pods).

*Service Account & Token Controllers:* Create default accounts and API access tokens for new namespaces.

**Scheduler:** Watches newly created pods that have no node assigned, and selects a node for them to run on.

**Api Server:** The front-end for the Kubernetes control plane. It is designed to scale horizontally. Every other component contact with this to communicate.

**Etcd Cluster:** key/value backing store for cluster data, it stores state of the cluster (what nodes exist in the cluster, what pods should be running, which nodes they are running on, and a whole lot more) at any given point of time.

#### Components in Worker Nodes:

**Kubelet:** Agent that continuously watches API server. It makes sure that containers are running in a pod.

**Kube-proxy:** a proxy service that runs on each worker node to deal with individual host subnetting and expose services to the external world. It performs request forwarding to the correct pods/containers across the various isolated networks in a cluster.

### Various Ways of installing K8s

minikube single node cluster

kubeadm multinode cluster (doesn't support kubenet, hence require CNI [container network interface] plugin eg. flannel).

GKE multi node

Important: The network must be deployed before any applications. Also, CoreDNS will not start up before a network is installed. kubeadm only supports Container Network Interface (CNI) based networks (and does not support kubenet).

### Output format in get

-o wide width wise details output

-o yaml details output in yaml format

-o json details output in json format



By **Gaurav Pandey**  
(gauravpandey44)

Published 30th September, 2019.

Last updated 1st October, 2019.

Page 2 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>