# A Report on Kubernetes

Srikanth Ramanam[1]

[1] School of Informatics and Computing, Bloomington, IN 47408, U.S.A.
[*] Corresponding authors: srikrama@iu.edu

**Kubernetes is cluster management software. This paper explores the architecture, functioning and competition of Kubernetes.**

https://github.com/cloudmesh/sp17-i524/raw/master/paper1/S17-IR-2028/report.pdf

## 1. INTRODUCTION

Google Kubernetes is a cluster management platform developed by Google. Kubernetes is an open source system for "automating deployment, scaling and management of containerized applications" [1]. It primarily manages clusters through containers as they decouple applications from the host operating system dependencies and allowing their quick and seamless deployment, maintenance and scaling.

Docker tool, that can package applications and their dependencies into containers, was released in 2013, paving way for flexible and portable applications that can run on premises, public and private clouds. Google also decided to release key components of its orchestration, scheduling and load balancing software, called Borg as Open Source. This led to its release in 2015 as Kubernetes [2].

## 2. COMPONENTS

Kubernetes components are designed to be extensible primarily through Kubernetes API.

### 2.1. Pods

A pod is a basic unit of Kubernetes scheduling and provides a higher level of abstraction over containers. Pods are collections on containers and resources they are dependent on bundled together. Pods are not durable. They are easily created and deleted as needed.

### 2.2. Labels and selectors

Labels are Key-Value pairs associated with Pods that are queryable. This allows to manage a subset of nodes in a cluster selected through querying on the labels.

### 2.3. Services

A Kubernetes Service is a set of pods usually constitute a tier of an application. Such set of pods are identified by querying their labels. A service usually has an external IP address, a port and a label selector.

### 2.4. Controllers

These are used for managing for managing the pods. Replication controller replicates specified number of pods to maintain availability. DaemonSet Controller is used for running one pod on each node. And Job Controller is used for running Pods to completion of a task like a batch job.

## 3. ARCHITECTURE

Kubernetes has a Master-Worker architecture. The worker agents were earlier called minions but are now known as Nodes. The Kubernetes Master is the controlling unit of the cluster and is responsible for scheduling, managing the workload and communications [2].
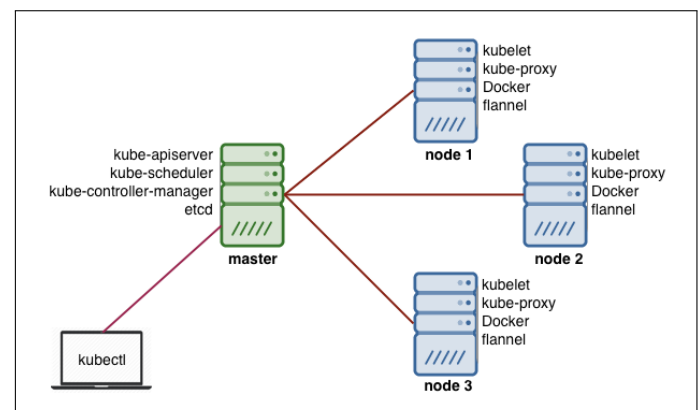


**Fig. 1.** Kubernetes Architecture [3]

### 3.1. Master

#### 3.1.1. API server

All the tasks by master and workers are accomplished through API calls, which are handled by API server. This serves the Kubernetes API which provides external and internal interface to Kubernetes API [4].

- Get API version: /api/

- Get list of pods: /api/v1/namespaces/default/pods

- Get list of nodes: /localhost:8080//api/v1/nodes

New configurations for pods can be uploaded using JSON or yaml files.

#### 3.1.2. Etcd

This is a persistent key-value data store used for storing configuration and overall state of the cluster. Etcd was developed by the CoreOS team.

#### 3.1.3. Scheduler and Controller Manager

Controller manager communicates with the API server and creates, manages or deletes pods according to the API calls and/or preset configurations [5].

### 3.2. Worker Node

#### 3.2.1. Kubelet

This is a special background process meant for implementing commands from master. The common commands include create, destroy and monitor its containers.

#### 3.2.2. Proxy

This is a network proxy and a load balancer at the node level and is responsible for redirecting the incoming requests to the appropriate containers of the node.

#### 3.2.3. cAdvisor

This is an agent that monitors resource usage and performance metrics of containers in nodes.

## 4. FUNCTIONING

Kubernetes applications are deployed over clusters as Services which comprise of a collection of pods. Pods are groupings of containers and their resources. These services are accessed through the endpoints they expose. Usually the target is to ensure high availability and scalability of services by managing the pods through controllers- creating, allocating and deleting when needed.

## 5. USER INTERFACE

Along with REST API, Kubernetes also provides a command line interface, kubectl [6], and a web based dashboard. This dashboard can be utilized to deploy, troubleshoot and manage containerized applications on Kubernetes. It also displays key metrics on resource usage and errors. Figure 2 shows a kubernetes dashboard.

## 6. LICENSE AND PRICING

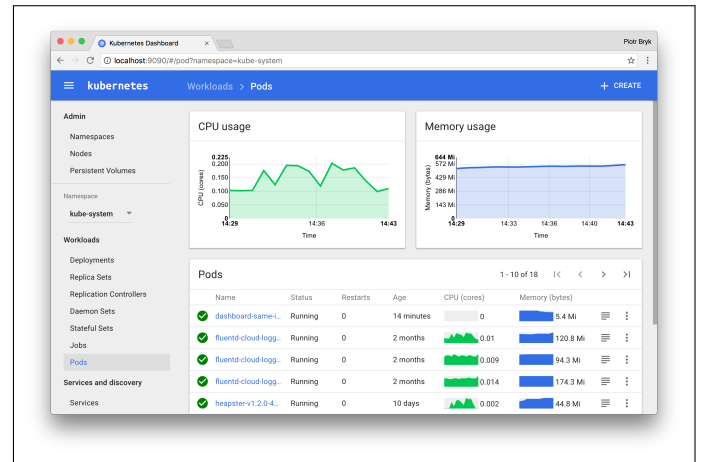Kubernetes is licensed under Apache License 2.0. [8].



**Fig. 2.** UI-dashboard [7]

### 6.1. Pricing

While Kubernetes itself is free and open source, a paid kubernetes implementation is being offered by Google Cloud Platform. Nodes are priced between 0.010 USD and 0.240 USD per hour depending on the type of the machine [9]. 0 to 5 node clusters are free when you are paying for the nodes. 6 plus node clusters are being charged at 0.15 USD per hour on top of the price for nodes [10].

## 7. COMPARISON

Some of the competitors of Kubernetes that offer orchestration, clustering and cluster management are [11]:

### 7.1. Docker Swarm

Swarm uses the standard Docker interface. Though this makes it easy to integrate with existing work flows, this may make it difficult to implement complex scheduling with custom interfaces.

### 7.2. Fleet

Fleet is a low level orchestration tool that can be used run other high level orchestration systems such as Kubernetes.

### 7.3. Mesos

Apache Mesos is also a low level, but more stable, scheduler that supports several frameworks for container orchestrations such as Kubernetes, Marathon and Swarm. Usually large applications and in Big Data centers.

### 7.4. Kontena

Kontena a relatively newcomer to the field is aimed at simplifying the whole process and is targeting smaller companies like startups and quicker deployment [12].

## 8. USERS

Some of the Current users of Kubernetes are Pearson, ebay, wikimedia, box, SAP and ComCast [13].

## 9. CONCLUSION

Kubernetes is an open source cluster management software. Kubernetes comes in with the advantage of being the result of 15

years of research and development by Google and is considered stable. It also has an active community of developers. While embracing Kubernetes might involve some redesigning of existing applications, when utilized efficiently it offers a fault-tolerant and efficient system.

## ACKNOWLEDGEMENTS

## REFERENCES

[1] Kubernetes, "Kubernetes site," Web Page, accessed: 2017-02-26. [Online]. Available: https://kubernetes.io/docs/whatisk8s/

[2] D. Rensin, *Kubernetes*, 1st ed. O'Reilly Media, Inc., Jun. 2015, accessed: 2017-02-27.

[3] S. Braun, "Kubernetes architecture," Web Page, 2016, accessed: 2017-04-04. [Online]. Available: http://www.yet.org/2016/06/kubernetes/

[4] Kubernetes, "Kuberenetes api," Web Page, accessed: 2017-04-04. [Online]. Available: https://kubernetes.io/docs/tasks/access-kubernetes-api/http-proxy-access-api/

[5] Wikipedia, "Kubernetes wiki," Web Page, accessed: 2017-02-26. [Online]. Available: https://en.wikipedia.org/wiki/Kubernetes

[6] Kubernetes, "kubectl documentation," Web Page, accessed: 2017-02-26. [Online]. Available: https://kubernetes.io/docs/user-guide/kubectl-overview/

[7] Kubernetes, "Kubernetes ui dashboard," Web Page, accessed: 2017-02-26. [Online]. Available: https://kubernetes.io/docs/user-guide/ui/

[8] Kubernetes, "Kubernetes license," Git hub file, 2015, accessed: 2017-02-26. [Online]. Available: https://github.com/kubernetes/kubernetes/blob/master/third_party/swagger-ui/LICENSE

[9] Google, "Google container engine pricing," Web Page, Feb. 2017, accessed: 2017-02-26. [Online]. Available: https://cloud.google.com/compute/pricing

[10] G. C. Platform, "Google cloud platform pricing," Web Page, Feb. 2017, accessed: 2017-02-26. [Online]. Available: https://cloud.google.com/container-engine/pricing

[11] A. Mouat, "Kuberntes competition," Web Page, Oct. 2015, accessed: 2017-02-26. [Online]. Available: https://www.oreilly.com/ideas/swarm-v-fleet-v-kubernetes-v-mesos

[12] Kontena, "Kontena website," Web Page. [Online]. Available: https://www.kontena.io/

[13] Kubernetes, "Kubernetes users," Web Page, accessed: 2017-02-26. [Online]. Available: https://kubernetes.io/case-studies/