

Kubernetes

A Container Management Platform

I. INTRODUCTION

The ubiquity of wireless networks and increasing number of electronic devices promote development of Internet of Things (IoTs), which highlights the high cost and difficulty of application deployment. Unlike the traditional deployment way such that run applications on a physical server directly, the container concept is introduced. It has some similarity of using virtual memory to address the memory limitation and program conflict problems. The containers behaved like virtual machines, but it is more lightweight. It is isolated from the underlying infrastructure so it is portable to deploy through cloud.

Kubernetes is an open-source platform used to bundle and run the applications in a production environment [1]. As IaaS (Infrastructure as a Service), PaaS (Platform as a Service) and SaaS (Software as a Service) are often used with cloud, *Kubernetes* served a bit like PaaS, with offering deployment, scaling, load balancing and etc [2].

II. BASIC ARCHITECTURE

First to understand its mechanism, start with its architecture [3].

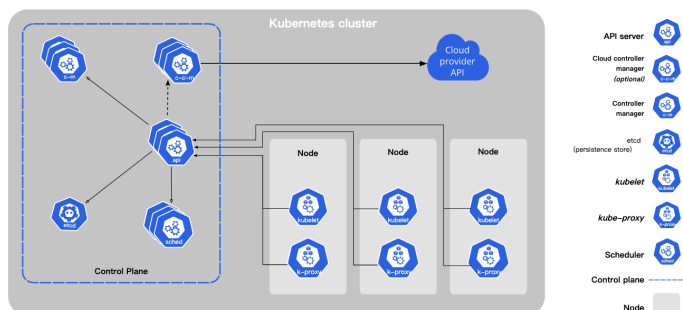


Figure 1. Kubernetes Cluster [4]

- A Kubernetes cluster consists of a master and various nodes, the control plane could be considered as utilized through the master.
- API server is used as the system's external interface, such that the control plane "communicates" with the nodes through API server. Other components could call from it.
- Scheduler is the whole system's scheduling part. As a cluster, there would exist problem for control plane such that which node to call, in which node deploy new application (contained by a pod).

- Controller manager is used to control the controller, including node controller, replication controller, endpoint controller, service account and token controller.
- ETCD is used to store the whole cluster's status.
- Cloud controller manager and cloud provider API is used for cloud management and could be omitted.
- Kubelet watch on each pod sent to its node, including pod creation, modification and deletion.
- K-proxy is used to offer proxy for the node. The network communication within cluster is different, such that there will be virtual IP address for master to communicate with nodes and among different nodes. Calico and Flannel are popular network communication methods for Kubernetes cluster.
- Pod is the atomic unit for deployment. Each time the application could be contained by a pod and deployed on one or more specific nodes, controlled by scheduler. Each pod could have various containerized app and some shared-resources for these containers, such as the shared storage, as volume. The IP address of this pod, called Pod IP address, is used for all containers with this pod. The pod will also store the information of how to run each container so the node could utilize the containers within the pod.
- Service is used to combine many pods, and a service in whole could be used. DNS could be used to address the problem that the services visiting each other, through the service name to visit the virtual IP address. And for user, ingress is used to visit on master.

III. SPECIAL PROPERTIES

Kubernetes is already widely used in industry. For example, Ant Financial has huge scale of data processing, it is really challenging to do real-time compute, storage, and processing capability as well as ensure the security [5]. They begun to use containers in 2014, and migrated the production workload to the *Kubernetes* platform. There are some special properties that might cause *Kubernetes* to be so popular.

- DaemonSet. It could be configured for different pods. The when a new node join in the cluster, this pod with DaemonSet could be automatically deployed on the new node. When a node is deleted from the cluster, the pod will be safely deleted as well [6]. We could see the efficiency of the cluster with such mechanism, as in industry a huge cluster is highly likely to be used. An automatically set up and deletion is very useful.
- ReplicaSet. It is used to ensure at each time, the number of pods running is a certain fixed number. So even when

a node is shuttled down accidentally, the service provided will not be influenced.

- Deployment. It could realize the function of replicaset but with more properties. Or it is same to view replicaset is implemented through Deployment. The pod could be deployed through the deployment. It provides a simple way to manage resources. New deployment could have all the resources from an old deployment.
- Customization. The *Kubernetes* cluster could be customized through changing scheduler, which default is Kube-Scheduler. It is commonly appeared in Federated Learning areas. Because for Federated Learning, its critical difference with Machine Learning is how the resources or models are allocated to different clients (nodes). So the default scheduler of *Kubernetes* could be replaced by a special customized scheduler using configuration file [7]. This might be more useful for research areas.

IV. POTENTIAL LIMITATIONS

Although it is widely used in industry, it still exists some limitations. As a new and developing open-source platform, it's still upgrading and developing new features. Because we do not need to do much coding for building such a platform, it is highly correlated to potential issues during running. Although suggested in [7] that *Kubernetes* could be used for Machine Learning, there might be some problems even though *Kubernetes* has install instruction and quick setup.

- Version. Because of the continuous upgrading, various version might support different devices, and the components version is specified for one specific version.
- Inconsistent default setting. Different components for *Kubernetes* have inconsistent settings, and the modification of the default setting might fail. So the customization might not be well supported.
- Communication within cluster. As introduced before, it uses different network communication method. It needs to be decided before the initialization, but due to the real world devices limit, the network policy, or the container network plugin might be changed.
- Various standard. The Raspbery Pi could not join the cluster, although the official documents claim the support for Raspbery Pi. Because if installed the GUI, it will not be supported anymore.

The troubleshooting for *Kubernetes* is not so understandable and the solutions often needs a long time searching as well as reading the logs. But the varying situation make it difficult to find the right solution. Integrated with a problem diagnosis using log might be a future developing direction [8]. A precise log model is provided and the experimental results show their approach can find out the real-world *Kubernetes* problems. So the potential development might be a lower application threshold, as for now, one corporation might still need advanced engineers to apply and maintain this system, which limits a more extensive usage.

REFERENCE

- [1] A. Kubernetes. (2020) What is kubernetes. [Online]. Available: <https://kubernetes.io/docs/concepts/overview/what-is-kubernetes/>
- [2] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 27–51, 2015.
- [3] D. K. Rensin, *Kubernetes - Scheduling the Future at Cloud Scale*, 1005 Gravenstein Highway North Sebastopol, CA 95472, 2015. [Online]. Available: <http://www.oreilly.com/webops-perf/free/kubernetes.csp>
- [4] A. Kubernetes. (2020) Kubernetes components. [Online]. Available: <https://kubernetes.io/docs/concepts/overview/components/>
- [5] —. (2020) Ant financial's hypergrowth strategy using kubernetes. [Online]. Available: <https://kubernetes.io/case-studies/ant-financial/>
- [6] —. (2020) Kubernetes components. [Online]. Available: <https://kubernetes.io/docs/concepts/workloads/controllers/daemonset/>
- [7] Y. Bao, Y. Peng, C. Wu, and Z. Li, "Online job scheduling in distributed machine learning clusters," in *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*, 2018, pp. 495–503.
- [8] J. Xu, P. Chen, L. Yang, F. Meng, and P. Wang, "Logdc: Problem diagnosis for declaratively-deployed cloud applications with log," in *2017 IEEE 14th International Conference on e-Business Engineering (ICEBE)*, 2017, pp. 282–287.