Kubernetes Research Paper

Matthew Mattozzi

Department of Cybersecurity, Old Dominion University

CYSE 280: Windows System Management and Security

Professor Malik Gladden

November 29, 2022

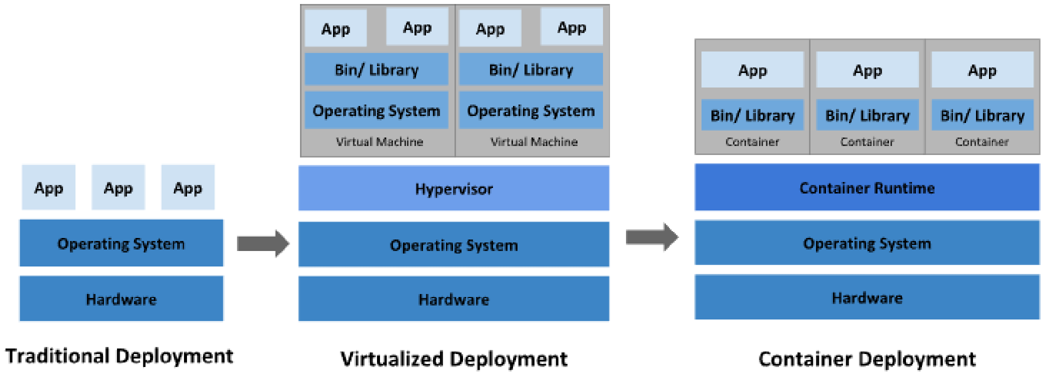
Abstract

This study will explore the inner machinations and building blocks of the container orchestration platform, Kubernetes. Beginning with the basic container, the platform will be expounded upon as the paper moves up the Kubernetes hierarchy. Finally, use cases for Kubernetes in DevOps will be explained, along with alternative considerations for similar platforms.

Kubernetes Research

A relatively young technology, Kubernetes was released in June 2014 and positioned itself to be a force in the industry when it came to container and automation management (Burns, 2022). This paper will discuss the building blocks that make up this orchestration tool by starting at the most basic level—the container—and building up to the complete Kubernetes hierarchy. Once this has been established, real use-cases of the technology will be discussed and how it makes sense for various industries to incorporate this new technology in their respective businesses.

Deploying applications has changed significantly. The following diagram provided by the Kubernetes foundation shows the transitions that have occurred.

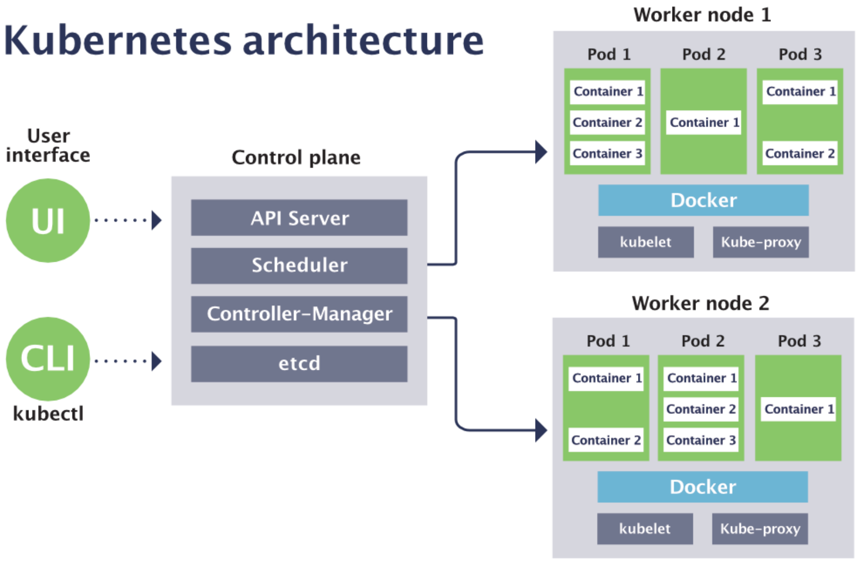


In the first block, applications were deployed in the format of hardware->OS->apps. Virtualized deployments optimized this flow by adding on a hypervisor that could run multiple virtual machines, each containing their own OS and apps. Container deployment streamlines this process by replacing the hypervisor with a Container

Runtime. Now, nodes in the cluster do the legwork. More details into how this works exactly will proceed in the following paragraphs.

In order to understand why Kubernetes is emerging as a prominent player in the world of orchestration, it is pertinent to first give a brief overview of what they help orchestrate: containers. A container is a type of special packaging that can house multiple components into a single entity (or container) that can perform various types of microservices/applications. The reason this is so useful is because any single application is made up of a plethora of components. Developers then need to concern themselves with how to package up these components, how to configure them with their various dependencies and conflicts, and how to best deliver them into a runtime environment (Hohn, 2022, p. 7). Containers take the guesswork out of this conundrum. In addition, they give much necessary scalability, because it will never just be a single container in use. Large companies could be running thousands of variously configured containers simultaneously. Imagining the headache of configuring each of containers separately is immediately apparent, so efficient orchestration for configuration is a major boon.

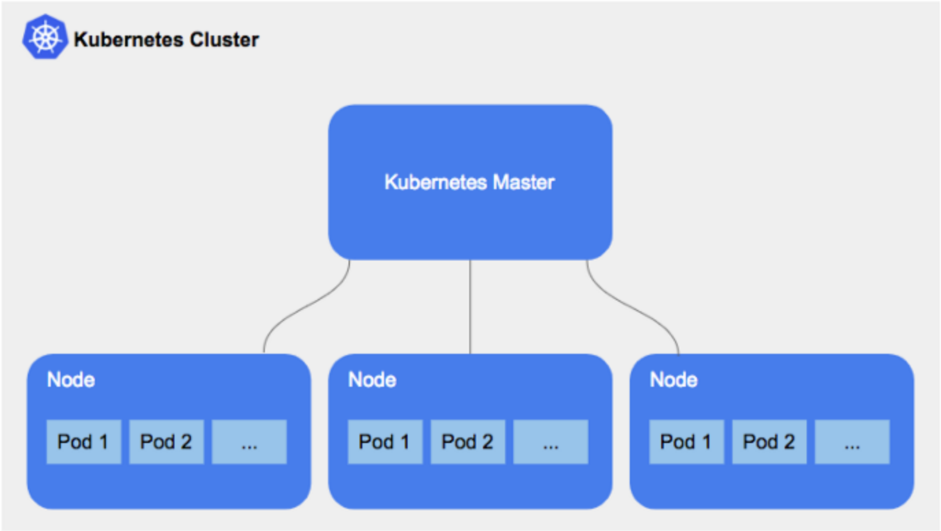
These containers need to meet certain requirements for the various microservices they may be used for. These include packaging, versioning, isolation, fast startup, and low overhead. And for these containers to be used in orchestration, the idea behind Kubernetes, clustering, discovery, and configuration is also required (Hohn, 2022, p. 8). The term ‘microservices’ is interchangeable with ‘fleets of containers’ (MirantisIT, 2022)



By Sensu, Jef Spaleta, 2016, located at https://sensu.io/blog/how-kubernetes-works

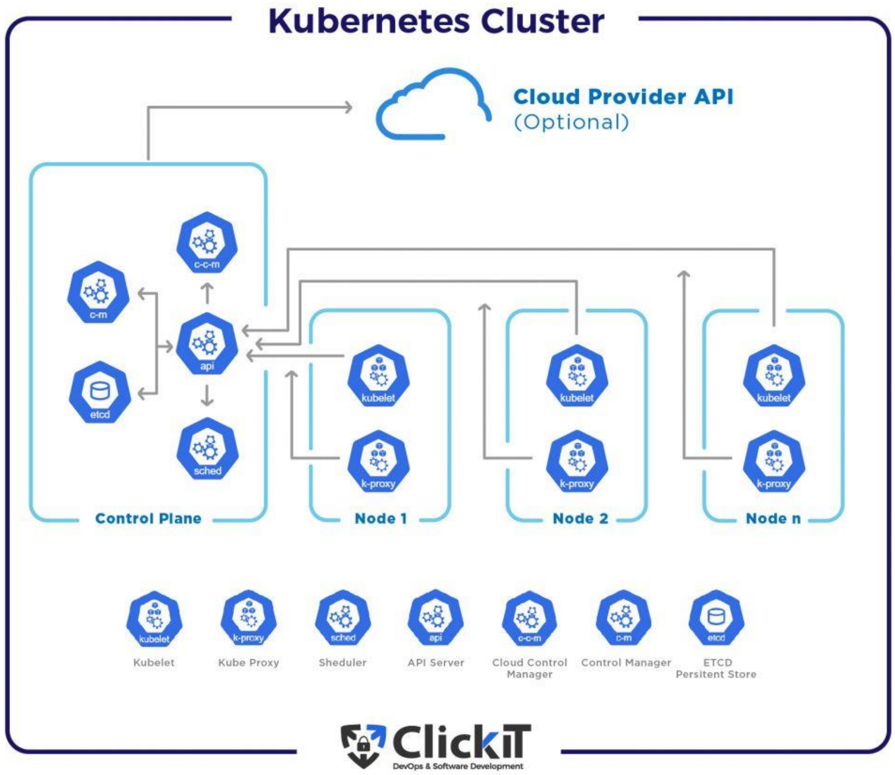
The above diagram gives a more tangible overview of where containers fit in the grand scheme of things and will help segway into the next section: pods. As one can see above, multiple containers fit into what is known as a pod. Fundamentally, pods are the smallest unit used in Kubernetes orchestration. A pod supplies a single IP address to every container within the pod, allowing all the collective containers to be treated as a singular unit (Spaleta, 2016).

The next level in the hierarchy is the *node*, which corrals the pods that function together into a single entity. These nodes can be a physical machine or a virtual machine. These nodes will be managed via the Control Plane. It is also important to mention here that the various nodes will be further organized by being grouped into *clusters*.



By Medium, Tomer Froumin, 2019, located at https://medium.com/@tomerf/so-you-want-to-configure-the-perfect-db-cluster-inside-a-kubernetes-cluster-a4d2c26aca7a

This diagram provides a succinct example of the overall Kubernetes hierarchy which we are constructing. Each of the pods contains several containers, which are then contained collectively in a pod. These pods are stored in a node, which are then contained in a cluster that makes up the necessary number of nodes. This level, the *cluster*, contains a master node which is charged with managing the state of the cluster. This master node is vital because it performs various important functions such as acting like a scheduler, API server, Kubelet (ensuring containers are acting optimally within their respective pod), controller manager, and Kube-proxy (maintains all networking rules and like across the network) (ARMO, 2022).

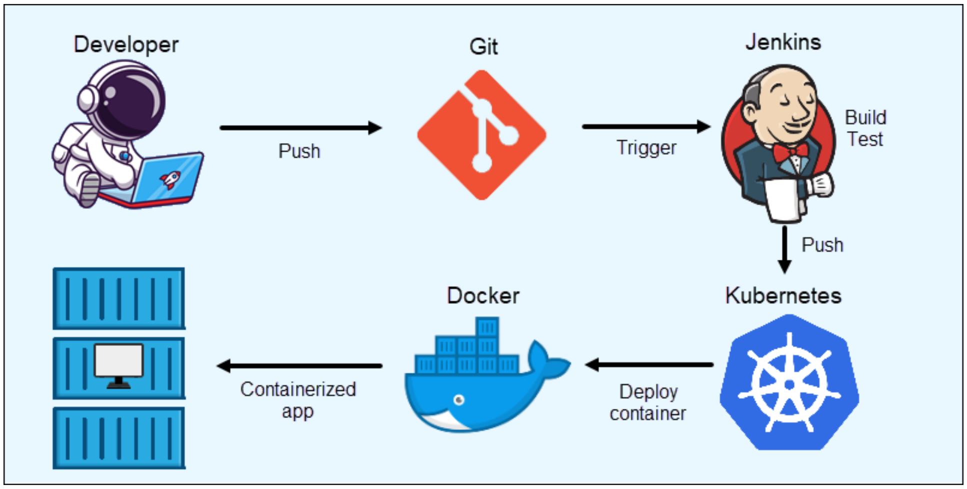


The diagram above provides the most comprehensive view of a Kubernetes cluster so far. It is important to note that it is almost never just one cluster operating independently, but instead a multitude of them (the exception being in a home-lab, for instance).

Use Cases

The machinations behind the service are important to understand, but what are use cases in which Kubernetes can be employed? The first is the management of microservices by optimizing communication between the different application components (Aleksic, 2022). Another is CI/CD or Continuous Integration – Continuous

Delivery/Deployment, which is a workflow that is optimized by minimizing downtime while developers continually work. Below is an example of this case:



By phoenixmap. 2022, located at https://phoenixnap.com/kb/kubernetes-use-cases

But there are several more use cases, as well. Kubernetes allows serverless computing. This technology is nothing new, however Kubernetes is unique in that it enables the creation of an independent serverless platform to give developers higher control of all backend processes (Aleksic, 2022). Big data processing, computation, and machine learning provide greater flexibility for deployment and use by developers using the Kubernetes framework. It would also be remiss not to mentioned that Kubernetes interfaces excellently with other container engines, notably Docker (Gupta, 2021).

Conclusion

Kubernetes is one of many container orchestration platforms available for use. Competitors include, but are not limited to, AWS Fargate, Azure Container Instance, Google Cloud Run, Docker Swarm, Nomad, and Rancher (Aqua, 2022). And because of high costs, potentially being overkill for the needs of some, and learning curve to employ, it may not always be the best fit for a company. Still, this relatively new platform shows much promise for its use as a container orchestration platform in the DevOps landscape. Not without its faults, it gives powerful control for necessary tools such as load balancing, storage orchestration, automated rollouts and rollbacks, secret and configuration management (Burns, 2022). These features, coupled with a burgeoning open-source community, make it an attractive option in a field of competitors.

References

10 kubernetes alternatives and why you need them. Aqua. (2022, August 10). Retrieved November 21, 2022, from https://www.aquasec.com/cloud-native-academy/kubernetes-101/kubernetes-alternatives/

Aleksic, M. (2022, July 28). *Kubernetes use cases {8 real life examples}*. Knowledge Base by phoenixNAP. Retrieved November 21, 2022, from https://phoenixnap.com/kb/kubernetes-use-cases

Burns, B. (2018, July 20). The history of kubernetes & the community behind it. Kubernetes. Retrieved November 21, 2022, from https://kubernetes.io/blog/2018/07/20/the-history-of-kubernetes-the-community-behind-it/

Froumin, T. (2019, August 7). *So you want to configure the perfect DB cluster inside a kubernetes cluster*. Medium. Retrieved November 21, 2022, from https://medium.com/@tomerf/so-you-want-to-configure-the-perfect-db-cluster-inside-a-kubernetes-cluster-a4d2c26aca7a

Gupta, H. (2021, March 13). Research how Kubernetes is used in industries and what all use cases are solved by Kubernetes. LinkedIn. Retrieved September 15, 2022, from https://www.linkedin.com/pulse/research-how-kubernetes-used-industries what-all-use-cases-gupta

Hohn, A. (2022). The book of Kubernetes: A complete guide to container orchestration. No Starch Press.

MirantisIT. (n.d.). What is kubernetes?: Why Use Kubernetes & More. Mirantis. Retrieved September 15, 2022, from https://www.mirantis.com/cloud-native concepts/getting-started-with-kubernetes/what-is-kubernetes/

Overview. Kubernetes. (n.d.). Retrieved November 21, 2022, from https://kubernetes.io/docs/concepts/overview/

Spaleta, J. (2019, August 8). *Jef Spaleta*. Sensu. Retrieved November 21, 2022, from https://sensu.io/blog/how-kubernetes-works

What is a kubernetes cluster? ARMO. (2022, October 13). Retrieved November 21, 2022, from https://www.armosec.io/glossary/kubernetes-cluster/