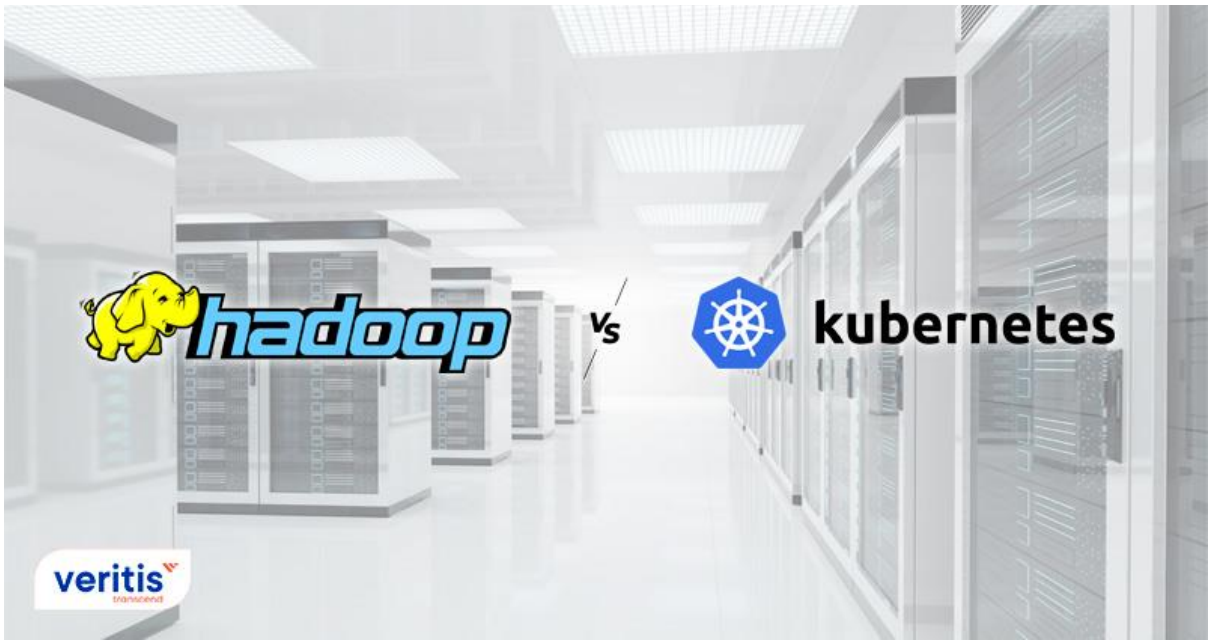
Hadoop Vs. Kubernetes Is K8s invading Hadoop Turf?



Kubernetes's popularity is something we are aware of. In the age of containerization, DevOps teams worldwide rely on containers such as Kubernetes or K8s. There are various reasons for its high adoption rate. Be it due to its open-source origins or malleable compatibility; developers favor it over various other options.

Its extreme usage has allowed Gartner to predict that "by 2022, more than 75% of global organizations will be running containerized applications in production, up from less than 30% today."

While this is a heartening observation, the question often cropping up is whether Kubernetes encroaches on the Hadoop market. The open-source tool, Hadoop, has a base of its own, primarily due to its data processing abilities. However, Hadoop isn't just a data processing tool and a storage utility that Kubernetes is now overshadowing.

While Hadoop is not dead in the water, many users are now shifting to other tools to get the job done. While one of them is Apache's Spark, among many others, Kubernetes is

another tool that DevOps project members rely upon. At the outset, you may wonder how Kubernetes is invading Hadoop's domain as the former is a container, and the latter is a data processing tool.

Useful link: Kubernetes Adoption: The Prime Drivers and Challenges

---

At first glance, both don't have a direct relation apart from being open-source tools. However, Kubernetes is poaching the user base of Hadoop, and in this blog, we shall understand what Hadoop is, what Kubernetes is, and how users are leveraging **Kubernetes over Hadoop**.

Understanding Hadoop

Developed by the Apache foundation, Hadoop is an open-source tool its users employ to crunch and manage vast volumes of data sets that may span various infrastructure clusters. The data processing and storage are made possible by easy-to-use programming models. The tool has made a name for itself as it is capable enough to process mountain loads of data that can even go up to petabytes.

Apache's unique approach enables these cutting-edge data processing abilities, where the Hadoop processes the data on multiple servers instead of running the entire data

operation on one server. These parallel data processing operations allow Hadoop to crunch the data faster than the standard tools which are on the market.

The distributed computational capabilities give Hadoop to execute the data operations over numerous servers, and each one comes along with processing power and computational storage.

This decentralized operational approach makes it easy for the users, and Hadoop brings additional benefits, such as failure management at the application layer level.

This feature allows the open-source tool to detect and mitigate the failures in the data operations it is assigned to take care and thereby, the users are provided high-quality services.

Under the hood, Hadoop is propped up by four modules which are:

- **HDFS:** Hadoop Distributed Files System, abbreviated as **HDFS**, buttresses Hadoop's primary principle to execute data operations. The USP of this module is that it can be executed even on low-specs hardware infrastructures. It comes with the capacity to take on large volumes of datasets while utilizing its built-in fault tolerance to deliver better results.

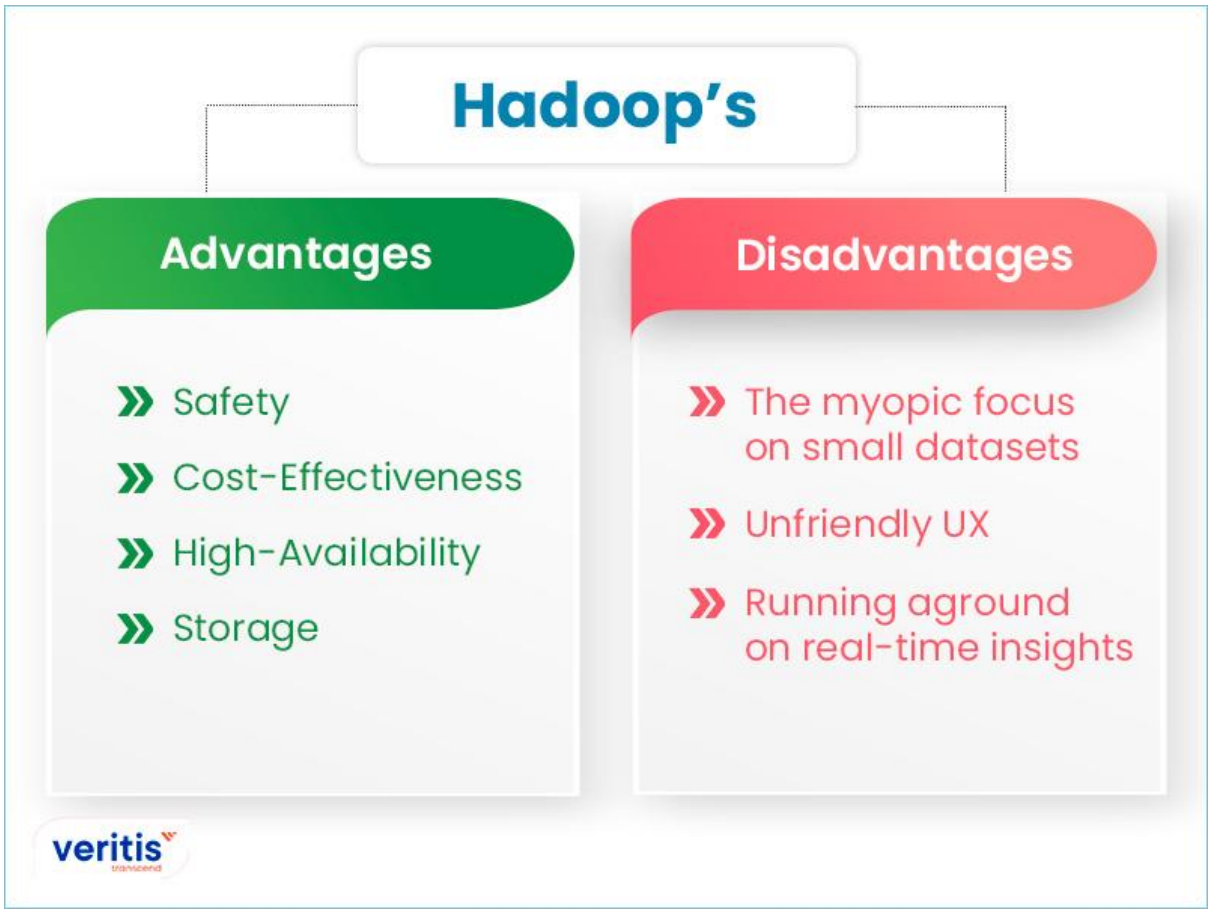
- **YARN:** It appears Marvel Comics may have influenced YARN's acronym as it bears semblance to Just A Really Very Intelligent System, who is popularly known as Iron Man's AI assistant, JARVIS. **YARN** is Yet Another Resource Negotiator, and while it sounds tiresome, it is a useful tool that allows one to manage tasks, schedules, and other priorities.

- **MapReduce:** Massive amounts of data may be processed in parallel using the big data processing engine **MapReduce**. It is the Hadoop processing engine by default. But, Hadoop also accommodates other systems like Apache Tez and Apache Spark.

- **Hadoop Common:** A set of frameworks that all the other Hadoop modules may utilize is provided by **Hadoop Common**.

As you may have caught the scent, the blog is about to indulge you in the benefits of Hadoop.

Hadoop’s edge



Hadoop’s distinctive trait is its data processing capabilities. But, there are more advantages; let’s dig in.

- **Safety:** Backups are created and managed automatically using Hadoop. Therefore, in the event of a breakdown, you may quickly restore your data from a backup.

- **Cost-Effectiveness:** Organizations can quickly create a data analytics platform using Hadoop since it can operate on a common infrastructure. Additionally, it does away with the need for pricey and specialized gear.

Headquarters: Veritis Group, Inc , 1231 Greenway Drive, Suite 1040, Irving, TX 75038

Phone: 972-753-0022 | **Email:** connect@veritis.com

- **High-Availability:** Hadoop offers high availability without relying on hardware since it is built to manage errors at the application layer.

- **Storage:** You may use Hadoop to retain the raw data indefinitely once a sizable data set has been gathered, and the necessary data has been extracted. Users may now readily correspond to earlier data, and as Hadoop runs on affordable technology, storage costs are also low.

But, as is with everything, there are certain disadvantages to Hadoop. Let's look at them.

Disadvantages of Hadoop

All of the characteristics above helped Apache Hadoop clusters become more popular. But as technology has developed, other choices have appeared that compete with Hadoop and sometimes even outperform it.

So, without further ado, let's understand the drawbacks of Hadoop, which later leads to the crux of the blog.

- **The myopic focus on small datasets:** Hadoop is made for processing extensive data, which consists of enormous data collections. Smaller data sets are processed very inefficiently. Therefore, when it comes to fast analytics of smaller data sets, Hadoop is unsuitable and too expensive. Another argument is that Hadoop does not offer an easy mechanism to produce the required data, even though it can aggregate, process, and transform data. As a result, business intelligence teams have fewer alternatives for visualizing and reporting on the processed data sets.

- **Unfriendly UX:** Java, one of the top programming languages with a sizable developer community, was used to construct Hadoop. Java is not the most significant language for data analytics, and it might be challenging for beginners. This may complicate setups and usage; to effectively utilize and troubleshoot the cluster, the user needs a solid understanding of both Java and Hadoop. Additionally, it is an open-source tool, and let's face it, open-source tools don't

come with the best user interfaces as self-sustaining foundations design them, in this case, the Apache foundation.

- **Running aground on real-time insights**: Hadoop's architecture well supports system analysis and design. However, Hadoop is not a good choice for speedy real-time analytics due to its limits in processing smaller data volumes and lack of native support.

How are K8s poaching the Hadoop Userbase?

Hadoop currently restricts users from using its tools and technologies, such as HDFS and YARN with Java-based tools, even with newer and faster data processing engines.

But what if you need to combine several platforms and technologies to obtain the best results for your unique data storage and analytics requirements? The answer is to manage your cluster with Kubernetes as the orchestration engine.

There are various benefits that come along with Kubernetes. Without digressing, Kubernetes's ease of use and its various other advantages is various enticing users to jump the ship. While cost-effectiveness is one of the advantages of Hadoop, it oddly restricts its users from moving to affordable data warehouses such as Amazon Redshift or Google BigQuery.

K8s outshines Hadoop on this aspect. Meanwhile, Kubernetes can quickly integrate them into Kubernetes clusters so that the containers can access them. Like cloud providers handle all daily maintenance and data availability, Kubernetes clusters offer unlimited storage with few maintenance requirements.

K8s also allow users to execute Big Data tools such as Apache Spark (another contender of Hadoop) and any tool gelling well with Kubernetes clusters. This freedom doesn't restrict you to Java-based tools.

The mobility of Kubernetes is another strength. Kubernetes is simple to set up so that it may operate in various cloud settings and spread across several locations. Users of containerized apps may quickly switch between development and production systems, enabling data analytics to take place wherever they choose without requiring significant changes.

With the capability for serverless computing, Kubernetes has further reduced the need to manage infrastructure independently. In the serverless model, which is in its infancy, the cloud platform grows and controls the hardware resources based on the application's demands.

Serverless applications may be executed on several container-native, open-source, and function-as-a-service computing platforms without requiring tools like Hadoop.

Final Thoughts

It is quite apparent that K8s are overshadowing Hadoop. However, Hadoop still packs a punch, especially while handling large datasets better than others. If Hadoop had become irrelevant, it wouldn't even be discussed in this blog. But, Hadoop's user base

will actively consider other options to get the job done at some point. And one of these options is Kubernetes.

The developer community widely uses these open-source tools but mitigating both requires an experienced helmsman. Veritis, the Stevie Award winner, has enough experience in Big Data and Kubernetes to churn out a solution based on your requirements.

So, approach us with your unique needs, and we shall fire on all cylinders to bring a customized solution.

Services

Headquarters: Veritis Group, Inc , 1231 Greenway Drive, Suite 1040, Irving, TX 75038

Phone: 972-753-0022 | **Email:** connect@veritis.com