Installation of Hadoop using Hortonworks Sandbox and Virtual box

Hortonworks

Hortonworks was a company founded in 2011 that specialised in providing open-source Apache Hadoop-based solutions for big data processing and analytics. Hadoop is designed to scale from single servers to thousands of machines, and it is widely used for processing and analysing large volumes of data in a distributed computing environment. Hortonworks contributed to the Hadoop ecosystem by providing its own distribution of Hadoop, along with additional tools and services to make it easier for organisations to implement and manage big data solutions.

In 2019, Hortonworks merged with Cloudera, another prominent player in the big data and Hadoop space. The merger created a unified company called Cloudera, combining the strengths and technologies of both companies to offer a comprehensive enterprise data management and analytics platform. The merged entity continued to provide solutions for data management, analytics, and machine learning on large-scale data sets.

Hortonworks Sandbox

The Hortonworks Sandbox refers to a Docker container provided by Hortonworks for learning, testing, and experimenting with big data technologies, particularly those based on the Apache Hadoop ecosystem. The Sandbox is essentially a pre-configured and pre-installed virtual machine that runs on a user's local system, allowing them to explore various Hadoop-related tools and components without the need to set up a full-scale Hadoop cluster.

Key features of the Hortonworks Sandbox include:

Ease of Use: The Sandbox is designed to be user-friendly and easily deployable, providing a quick way for users to get hands-on experience with Hadoop and related technologies.

Pre-installed Components: The Sandbox typically comes with a variety of Hadoop ecosystem components and tools pre-installed, such as Hadoop Distributed File System (HDFS), Apache Hive, Apache HBase, Apache Pig, Apache Spark, and more.

Tutorials and Examples: It often includes tutorials, sample datasets, and examples to help users learn and practice various big data concepts and workflows.

Docker container

A Docker container is a lightweight, standalone, and executable package that includes everything needed to run a piece of software, including the code, runtime, libraries, and system tools. Containers provide a consistent and reproducible environment, ensuring that an application runs consistently across different environments.

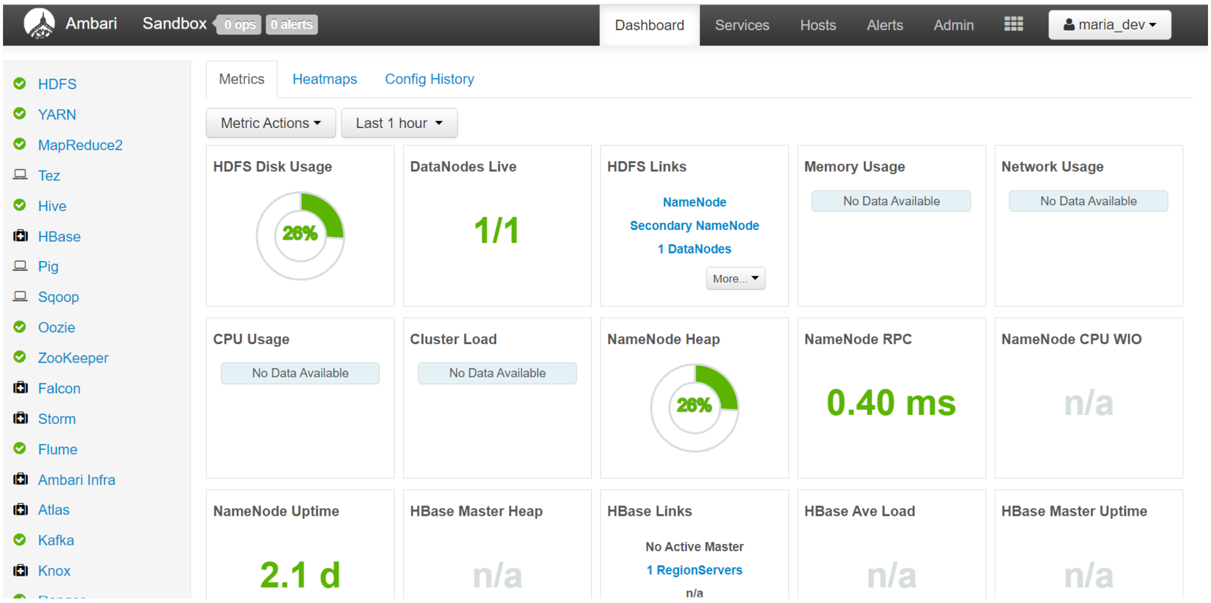
Key characteristics of Docker containers:

Isolation: Containers encapsulate an application and its dependencies, ensuring that it runs consistently across various environments without conflicts with the underlying system or other containers.

Portability: Containers are highly portable, allowing developers to package an application and its dependencies into a container image. This image can be shared and run on any system that supports Docker, providing consistency from development to production.

Resource Efficiency: Containers share the host operating system's kernel, making them lightweight and efficient in terms of system resources compared to traditional virtual machines.

Versioning: Container images can be versioned, allowing for easy rollback to previous versions or updates to newer versions.



HDFS Disk Usage: 26%
DataNodes Live: 1/1
HDFS Links: NameNode, Secondary NameNode, 1 DataNodes
Memory Usage: No Data Available
Network Usage: No Data Available
CPU Usage: No Data Available
Cluster Load: No Data Available
NameNode Heap: 26%
NameNode RPC: 0.40 ms
NameNode CPU WIO: n/a
NameNode Uptime: 2.1 d
HBase Master Heap: n/a
HBase Links: No Active Master, 1 RegionServers, n/a
HBase Ave Load: n/a
HBase Master Uptime: n/a