

Hortonworks Data Platform

Quick Start Guide

1.0 (Nov 30, 2012)

Hortonworks Data Platform : Quick Start Guide

1.0 (2012-11-30)

Copyright © 2012 Hortonworks, Inc. All rights reserved.

Hortonworks Data Platform (HDP) is a 100% open source data management platform based on Apache Hadoop. It allows you to load, store, process and manage data in virtually any format and at any scale. As the foundation for the next generation enterprise data architecture, HDP includes all of the necessary components to begin uncovering business insights from the quickly growing streams of data flowing into and throughout your business – TODO - verify

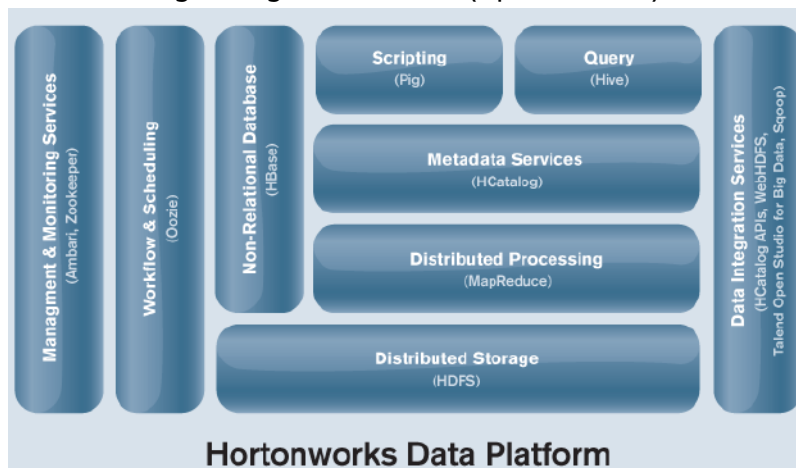
Table of Contents

- 1. About Hortonworks Data Platform 1
 - 1.1. Understanding Hadoop Ecosystem 2
 - 1.2. Typical Hadoop Cluster 5

1. About Hortonworks Data Platform

Hortonworks Data Platform (HDP) is an open source distribution powered by Apache Hadoop. HDP provides you with the actual Apache-released versions of the components with all the necessary bug fixes to make all the components interoperable in your production environments. It is packaged with an easy to use installer (HDP Installer) that deploys the complete Apache Hadoop stack to your entire cluster and provides the necessary monitoring capabilities using Ganglia and Nagios. The HDP distribution consists of the following components:

1. Core Hadoop platform (Hadoop HDFS and Hadoop MapReduce)
2. Non-relational database (Apache HBase)
3. Metadata services (Apache HCatalog)
4. Scripting platform (Apache Pig)
5. Data access and query (Apache Hive)
6. Workflow scheduler (Apache Oozie)
7. Cluster coordination (Apache Zookeeper)
8. Management and monitoring (Apache Ambari)
9. Data integration services (HCatalog APIs, WebHDFS, Talend Open Studio for Big Data, and Apache Sqoop)
10. Distributed log management services (Apache Flume)



To learn more about the distribution details and the component versions, see the [Release Notes](#). All components are official Apache releases of the most recent stable versions available. Hortonworks' philosophy is to do patches only when absolutely necessary to assure interoperability of the components. Consequently, there are very few patches in the HDP, and they are all fully documented. Each of the HDP components have been tested rigorously prior to the actual Apache release. To learn more about the testing strategy adopted at Hortonworks, Inc., see: [Delivering high-quality Apache Hadoop releases](#).

1.1. Understanding Hadoop Ecosystem

This section provides information on the various components of the Apache Hadoop ecosystem.

Apache Hadoop core components

Apache Hadoop is a framework that allows for the distributed processing of large data sets across clusters of commodity computers using a simple programming model. It is designed to scale up from single servers to thousands of machines, each providing computation and storage. Rather than rely on hardware to deliver high-availability, the framework itself is designed to detect and handle failures at the application layer, thus delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures.

HDFS (storage) and **MapReduce** (processing) are the two core components of Apache Hadoop. The most important aspect of Hadoop is that both HDFS and MapReduce are designed with each other in mind and each are co-deployed such that there is a single cluster and thus provides the ability to move computation to the data not the other way around. Thus, the storage system is not physically separate from a processing system.

Hadoop Distributed File System (HDFS)

HDFS is a distributed file system that provides high-throughput access to data. It provides a limited interface for managing the file system to allow it to scale and provide high throughput. HDFS creates multiple replicas of each data block and distributes them on computers throughout a cluster to enable reliable and rapid access.



Note

A file consists of many blocks (large blocks of 64MB and above).

The main components of HDFS are as described below:

- NameNode is the master of the system. It maintains the name system (directories and files) and manages the blocks which are present on the DataNodes.
- DataNodes are the slaves which are deployed on each machine and provide the actual storage. They are responsible for serving read and write requests for the clients.
- Secondary NameNode is responsible for performing periodic checkpoints. In the event of NameNode failure, you can restart the NameNode using the checkpoint.

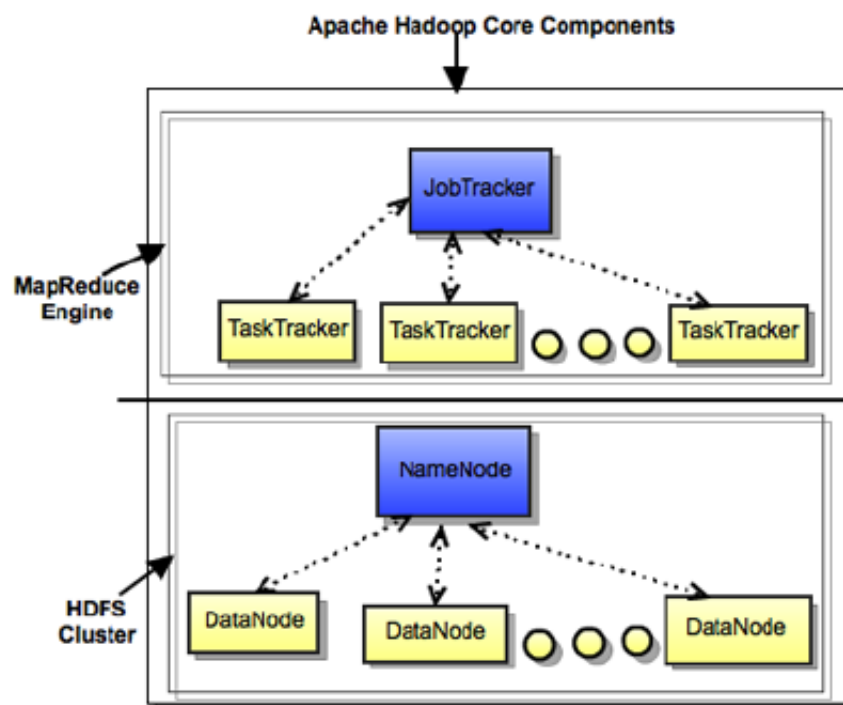
MapReduce

MapReduce is a framework for performing distributed data processing using the MapReduce programming paradigm. In the MapReduce paradigm, each job has a user-defined map phase (which is a parallel, share-nothing processing of input; followed by a user-defined reduce phase where the output of the map phase is aggregated). Typically, HDFS is the storage system for both input and output of the MapReduce jobs.

The main components of MapReduce are as described below:

- JobTracker is the master of the system which manages the jobs and resources in the cluster (TaskTrackers). The JobTracker tries to schedule each map as close to the actual data being processed i.e. on the TaskTracker which is running on the same DataNode as the underlying block.
- TaskTrackers are the slaves which are deployed on each machine. They are responsible for running the map and reduce tasks as instructed by the JobTracker.
- JobHistoryServer is a daemon that serves historical information about completed applications. Typically, JobHistory server can be co-deployed with JobTracker, but we recommend to run it as a separate daemon.

The following illustration provides details of the core components for the Hadoop stack.



Apache Pig

Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs, coupled with infrastructure for evaluating these programs. The salient property of Pig programs is that their structure is amenable to substantial parallelization, which in turns enables them to handle very large data sets. At the present time,

Pig's infrastructure layer consists of a compiler that produces sequences of Map-Reduce programs. Pig's language layer currently consists of a textual language called Pig Latin, which is easy to use, optimized, and extensible.

Apache Hive

Hive is a data warehouse system for Hadoop that facilitates easy data summarization, ad-hoc queries, and the analysis of large datasets stored in Hadoop compatible file systems.

It provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL. Hive also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.

Apache HCatalog

HCatalog is a metadata abstraction layer for referencing data without using the underlying filenames or formats. It insulates users and scripts from how and where the data is physically stored.

Templeton provides a REST-like web API for HCatalog and related Hadoop components. Application developers make HTTP requests to access the Hadoop MapReduce, Pig, Hive, and HCatalog DDL from within the applications. Data and code used by Templeton is maintained in HDFS. HCatalog DDL commands are executed directly when requested. MapReduce, Pig, and Hive jobs are placed in queue by Templeton and can be monitored for progress or stopped as required. Developers also specify a location in HDFS into which Templeton should place Pig, Hive, and MapReduce results.

Apache HBase

HBase (Hadoop DataBase) is a distributed, column oriented database. HBase uses HDFS for the underlying storage. It supports both batch style computations using MapReduce and point queries (random reads).

The main components of HBase are as described below:

- HBase Master is responsible for negotiating load balancing across all Region Servers and maintain the state of the cluster. It is not part of the actual data storage or retrieval path.
- RegionServer is deployed on each machine and hosts data and processes I/O requests.

Apache Zookeeper

ZooKeeper is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services which are very useful for a variety of distributed systems. HBase is not operational without ZooKeeper.

Apache Oozie

Apache Oozie is a workflow/coordination system to manage Hadoop jobs.

Apache Sqoop

Apache Sqoop is a tool designed for efficiently transferring bulk data between Hadoop and structured datastores such as relational databases.

Apache Flume

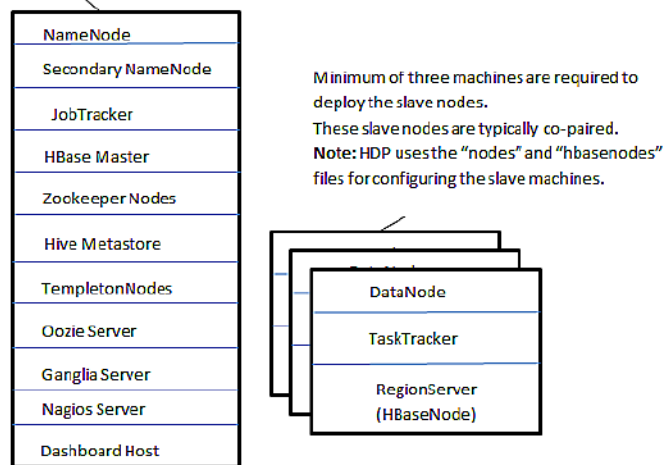
Flume is a top level project at the Apache Software Foundation. While it can function as a general purpose event queue manager, in the context of Hadoop it is most often used as a log aggregator, collecting log data from many diverse sources and moving them to a centralized data store.

1.2. Typical Hadoop Cluster

A typical Hadoop cluster comprises of machines based on the various machine roles (master, slaves, and clients). For more details, see: [Machine Roles In A Typical Hadoop Cluster](#). The following illustration provides information about a typical Hadoop cluster. A smallest cluster can be deployed using a minimum of four machines (for evaluation purpose only). One machine can be used to deploy all the master processes (NameNode, Secondary NameNode, JobTracker, HBase Master), Hive Metastore, Templeton Server, and Zookeeper nodes. The other three machines can be used to co-deploy the slave nodes (TaskTrackers, DataNodes, and RegionServers).

For evaluation purpose only -

The master nodes, Hive Metastore, ZookeeperNodes, TempletonNodes, Oozie Server, Ganglia Server, Nagios Server, and Dashboard Host can be co-deployed on same machine.



A minimum of three machines is required for the slave nodes in order to maintain the replication factor of three. For more details, see: [Data Replication in Apache Hadoop](#)

Machine roles in a typical Hadoop cluster:

In Hadoop and HBase, the following two types of machines are available:

- Masters (HDFS NameNode, Secondary NameNode, MapReduce JobTracker, and the HBase Master)



Note

It is recommended to add only limited number of disks to the master nodes, because the master nodes do not have high storage demands.

- Slaves (HDFS DataNodes, MapReduce TaskTrackers, and HBase RegionServers)

Additionally, we strongly recommend that you use separate client machines for performing the following tasks:

- Load data in the HDFS cluster

- Submit MapReduce jobs (describing how to process the data)
- Retrieve or view the results of the job after its completion
- Submit Pig or Hive queries

Based on the recommended settings for the client machines, the following illustration provides details of a Hadoop cluster:

