CDP Private Cloud Base Reference Architecture

Date published: 2022-12-05
Date modified: 2022-12-05

CLOUDERA

Legal Notice

Contents

Security integration... 29

FAQs...29

References and acknowledgements..30

CDP Private Cloud Base Reference Architecture: Abstract

The Cloudera Data Platform (CDP) Private Cloud Base Reference Architecture is a high-level design and best-practices guide for deploying CDP Private Cloud Base in customer data centers.

CDP Private Cloud Base is the on-premises version of the Cloudera Data Platform. This new product combines the best of Cloudera Enterprise Data Hub and Hortonworks Data Platform Enterprise along with new features and enhancements across the stack. This unified distribution is a scalable and customizable platform where you can securely run many types of workloads.

CDP Private Cloud Base supports a variety of hybrid solutions where compute tasks are separated from data storage and where data can be accessed from remote clusters. This hybrid approach provides a foundation for containerized applications by managing storage, table schema, authentication, authorization, and governance.

CDP Private Cloud Base comprises of a variety of components such as Apache HDFS, Apache Hive 3, Apache HBase, and Apache Impala, along with many other services for specialized workloads. You can select any combination of these services to create clusters that address your business requirements and workloads. Several pre-configured packages of services are also available for common workloads. These include:

- Data Engineering: Ingest, transform, and analyze data

  Services: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive metastore, Hive on Tez, Spark, Oozie, Hue, and Data Analytics Studio (DAS)
- Data Mart: Browse, query, and explore your data in an interactive way

  Services: HDFS, YARN, YARN Queue Manager, Ranger, Atlas, Hive metastore, Impala, and Hue
- Operational Database: Low latency writes, reads, and persistent access to data for Online Transactional Processing (OLTP) use cases

  Services: HDFS, Ranger, Atlas, and HBase

Installing a CDP Private Cloud Base cluster involes installing a parcel called Cloudera Runtime that contains all of the services and installing certain powerful tools to manage, govern, and secure your cluster. For a complete list of the included components, see Cloudera Runtime Component Versions.

> **Note:** Cloudera Reference Architecture guides illustrate sample cluster configurations and certified partner products. The reference architecture guides are not replacements for official statements of supportability. They provide guidance to assist with deployment and sizing options. Statements regarding supported configurations in the reference architecture guides are informational and should be cross-referenced with the latest documentation.
>
> CDP Private Cloud architecture might significantly affect the node and network sizing considerations. This reference architecture is appropriate for aggregated workload clusters running Cloudera Runtime.

Related Information
CDP Private Cloud Base Overview
CDP Private Cloud Base Runtime Documentation
Cloudera Reference Architecture Documentation
CDP Release Notes
Cloudera Services & Support

What's new in CDP Private Cloud Base

CDP Private Cloud Base offers the best of runtime services that provide services such as goverance, access control, compliance management, troubleshooting, and scheduling, to both CDH and HDP users.

For CDH users

Atlas

Apache Atlas provides data governance capabilities for Hadoop. Apache Atlas serves as a common metadata store that is designed to exchange metadata both within and outside of the Hadoop stack. The close integration of Atlas with Apache Ranger enables you to define, administer, and manage security and compliance policies consistently across all components of the Hadoop stack.

Ranger

CDP security components enable you to control access to CDP services and data sets, and also provide access to auditing and reporting.

Data Analytics Studio

Data Analytics Studio (DAS) is an application that provides diagnostic tools and intelligent recommendations to make the business analysts self-sufficient and productive with Hive. DAS helps you to perform operations on Hive tables and provides recommendations for optimizing the performance of your queries.

Phoenix

Apache Phoenix is an add-on for Apache HBase that provides a programmatic ANSI SQL interface. Apache Phoenix implements best-practice optimizations to enable software engineers to develop HBase based next-generation applications that operationalize big data. Using Phoenix, you can create and interact with tables in the form of typical DDL/DML statements using the Phoenix standard JDBC API.

YARN Capacity Scheduler

CDH offered the Fair Scheduler and HDP offered the Capacity Scheduler. After a thorough analysis of the YARN schedulers available in the legacy platforms, for CDP, Cloudera chose the Capacity Scheduler as the supported YARN scheduler. In CDP Capacity Scheduler, functionalities of the two schedulers are merged to minimize the impact to CDH users going through this transition.

For HDP users

Cloudera Manager

Cloudera Manager replaces Apache Ambari. It provides the operational interface to the cluster, allowing administrators to do common installation, configuration, and break/fix activity, such as changing service properties, adding or restarting services, or evaluating overall cluster performance, similar to Ambari Metrics Service and Grafana in HDP.

Impala

Apache Impala provides high-performance, low-latency SQL queries to run on data that is stored in popular Apache Hadoop file formats. The fast response for queries enables interactive exploration and fine-tuning of analytic queries, rather than long batch jobs traditionally associated with SQL-on-Hadoop technologies. Impala integrates with the Apache Hive Metastore (HMS) database, to share databases and tables between both components. The high level of integration with Hive, and compatibility with the HiveQL syntax lets you use either Impala or Hive to create tables, run queries, load data, and so on.

Kudu

Apache Kudu is a columnar storage manager developed for the Hadoop platform. Kudu shares the common technical properties of Hadoop ecosystem applications: Kudu runs on commodity hardware, is horizontally scalable, and supports highly-available operation. Kudu's benefits include:

- Fast processing of OLAP workloads
- Integration with MapReduce, Spark, Flume, and other Hadoop ecosystem components
- Tight integration with Apache Impala, making it a good, mutable alternative to using HDFS with Apache Parquet
- Strong but flexible consistency model, allowing you to choose consistency requirements on a per-request basis, including the option for strict serialized consistency

- Strong performance for running sequential and random workloads simultaneously
- High availability
- Structured Data Model

Hue

Hue is an open source analytics workbench designed for fast data discovery, intelligent query assistance, and seamless collaboration. SQL developers can do analytics on Hive and Impala, browse and load data to HDFS and also build Oozie workflows inside Hue.

For all users

Ranger RMS

Ranger Resource Mapping Service (RMS) enables automatic translation of access policies for Hadoop SQL (Hive and Impala) to HDFS. With the help of Ranger RMS any user with access permissions on a Hive table automatically receives similar HDFS file level access permissions on the table's data files. So, Ranger RMS allows you to authorize access to HDFS directories and files using policies defined for Hive tables. This is similar to the Sentry HDFS-ACL Sync feature that was present in CDH but implemented in a different way. Read more about Ranger RMS in this blog.

Ozone

Ozone is a scalable, redundant, and distributed object store, optimized for big data workloads. Apart from scaling to billions of objects of varying sizes, Ozone can function effectively in containerized environments such as Kubernetes and YARN.

Infrastructure

This section describes Cloudera's recommendations and best practices applicable to Hadoop cluster system architecture.

System Architecture Best Practices

Review the requirements and best practices for various infrastructure sub-components.

Operating system guidelines

Review the best practices and Cloudera's recommendations for installing operating systems (OS) on your clusters during your planning phase.

You must be aware of the following guidelines related to the operating system (OS) versions:

- All runtime hosts in a logical cluster must run on the same major OS release.
- Cloudera supports a temporarily mixed OS configuration during an OS upgrade project.
- Cloudera Manager must run on the same OS release as one of the clusters it manages.

Cloudera recommends running the same minor release on all cluster nodes. However, the risk caused by running different minor OS releases is considered lower than the risk of running different major OS releases.

You must be aware of the following guidelines related to the supported platforms:

- Cloudera does not support runtime cluster deployments in Docker containers.
- CDP Private Cloud Base is supported on platforms with Security-Enhanced Linux (SELinux) enabled and in an enforcing mode. Typical best practice is to implement SELinux in a permissive mode in order to benefit from regular security and feature improvements. Cloudera is not responsible for policy support or policy enforcement. If you experience issues with SELinux, contact your OS provider.

See CDP Private Cloud Base operating systems requirements for the latest information on compatibility and special considerations. See Cloudera Security Reference Architecture for cluster security best practices.

Related Information

Operating System Requirements

CDP Security Reference Architecture

Database guidelines

Review the best practices and Cloudera's recommendations for databases for your clusters during your planning phase.

Cloudera Manager and runtime services come packaged with an embedded PostgreSQL database for use in non-production environments. The embedded PostgreSQL database is not supported in production environments. For production environments, you must configure your cluster to use dedicated external databases. For more information on compatibility and special considerations, see Database Requirements.

Use UTF8 encoding for all custom databases.

> **Note:** Cloudera recommends that for most purposes you use the default versions of databases that correspond to the operating system of your cluster nodes. See the operating system's documentation to verify support if you choose to use a database other than the default. Enabling database backup for these databases should be addressed by the your IT department.

Related Information

Database Requirements

Java guidelines

Review the best practices and Cloudera's recommendations on JDKs for your clusters during your planning phase.

Only 64 bit JDKs are supported on the CDP clusters.

Unless specifically excluded, Cloudera supports later updates to a major JDK release from the release that support was introduced. Cloudera excludes or removes support for select Java updates when security is jeopardized.

Running Cloudera runtime hosts within the same cluster on different JDK releases is not supported. All cluster hosts must use the same JDK update level. See Java Requirements for the latest information in terms of compatibility and special considerations.

Related Information

Java Requirements

Right-size server configuration guidelines

Cloudera recommends deploying up to four machine types into your production environment: master nodes, worker nodes, utility nodes, and edge nodes.

Master nodes

> Runs the Hadoop master daemons such as NameNode, Standby NameNode, YARN Resource Manager and History Server, the HBase Master daemon, Ranger server, Atlas server, and the Impala StateStore server and Catalog server. Master nodes are also the location where Zookeeper and JournalNodes are installed. The daemons can share a single pool of servers. Depending on the cluster size, the roles can be run on a dedicated server. Kudu Master servers should also be deployed on master nodes.

Worker nodes

> Runs the HDFS DataNode, YARN NodeManager, HBase RegionServer, Impala impalad, Search worker daemons and Kudu Tablet Servers on the worker nodes.

Utility nodes

> Runs Cloudera Manager and the Cloudera Management Services. It can also host a MariaDB or another supported database instance, which is used by Cloudera Manager, Hive, Ranger, and other Hadoop-related projects.

Edge nodes

Contains all client-facing configurations and services, including gateway configurations for HDFS, YARN, Impala, Hive, and HBase. The edge node is also a good place for Hue, Oozie, HiveServer2, and Impala HAProxy. HiveServer2 and Impala HAProxy serve as a gateway to external applications such as the business intelligence (BI) tools. Edge nodes are also known as Gateway nodes.

Note: The edge and utility nodes can be combined in smaller clusters.

For more information on sizing clusters and role layout, see Runtime Cluster Hosts and Role Assignments.
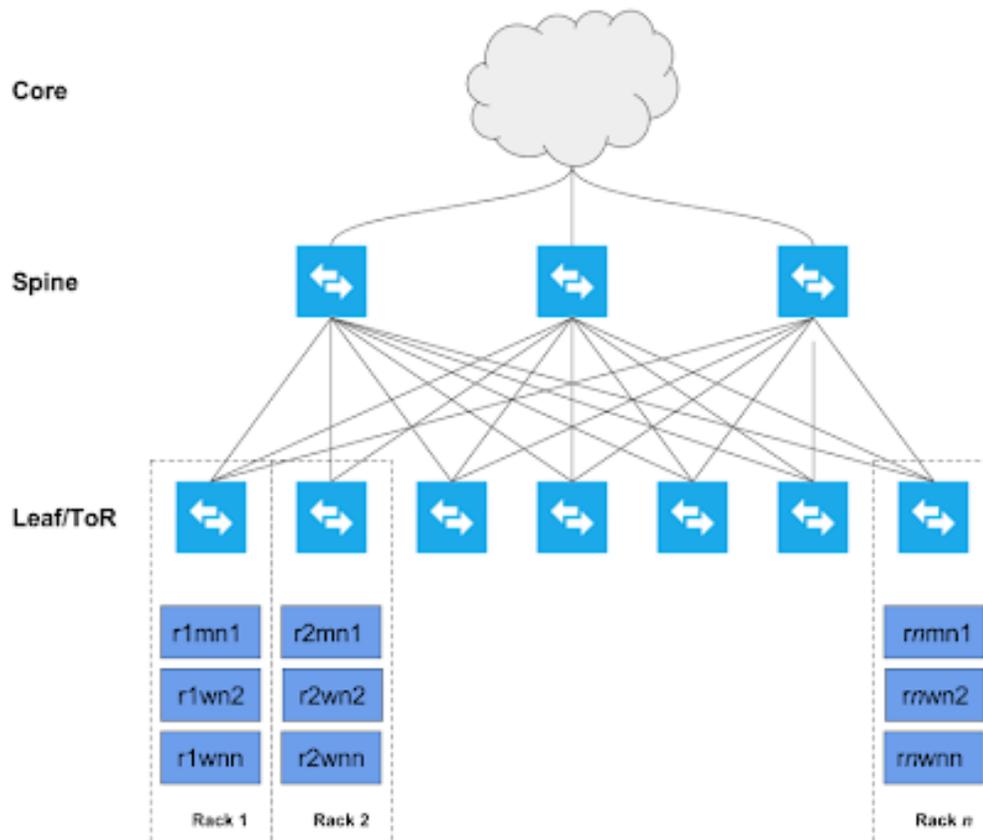
Related Information

Runtime Cluster Hosts and Role Assignments

Deployment topology

Learn about the recommended deployment topology for your CDP Private Cloud Base cluster that allows each host to maximize throughput and minimize latency, while encouraging scalability.

The following graphic shows a cluster deployed across several racks (Rack 1, Rack 2, … Rack n). Each host is connected to two top-of-rack (TOR) switches which are in turn connected to a collection of spine switches which are then connected to the enterprise network. This deployment model allows each host to maximize throughput and minimize latency, while encouraging scalability. The specifics of the network topology are described in the subsequent sections. The nomenclature represents the rack number, the role of a node in the cluster, and its ordinality in the rack. For example: r1mn1 would represent Rack1, Master Node 1, and so on. Every rack need not have a master node. It is a good practice to spread all master nodes of a cluster across different racks to avoid single point of failure. Gateway and utility nodes also reside within these racks (not annotated in the diagram).

Physical component list

Review the minimum configuration of physical components recommended for deploying a CDP Private Cloud Base cluster.

Component	Configuration	Description	Quantity
Physical servers	Two-socket, 8-16 cores per socket, > 2 GHz; minimum 128 GB RAM.	Hosts the various cluster components.	Based on cluster design.
NICs	10 Gbps Ethernet NICs (minimum required).	Provides the data network services for the cluster.	At least one per server, although two NICs can be bonded for additional throughput. **Important:** Cloudera does not support multi-homing. Multiple network adapters can be used, but they must be aggregated to provide a single host IP per node.
Internal HDDs	500 GB HDD or SSD/NVMe recommended for operating system and logs; HDD for data disks (size varies with data volume requirements). SSD/ NVMe can be used for YARN local directories.	Ensures continuity of service on server resets and contains the cluster data.	10-24 disks per physical server. The largest storage density currently supported is 100 TB per DataNode.
Ethernet ToR/leaf switches	Minimum 10 Gbps switches with sufficient port density to accommodate the cluster. These require enough ports to create a realistic spine-leaf topology providing ISL bandwidth above a 1:4 oversubscription ratio (preferably 1:1).	Although most enterprises have mature data network practices, consider building a dedicated data network for the Hadoop cluster.	At least two per rack.
Ethernet spine switches	Minimum 40 Gbps switches with sufficient port density to accommodate incoming ISL links and ensure required throughput over the spine (for inter-rack traffic).	Same considerations as for ToR switches.	Depends on the number of racks.

Network specification

Review the network specifications recommended for deploying a CDP Private Cloud Base cluster.

Dedicated network hardware

Hadoop can consume all available network bandwidth. For this reason, Cloudera recommends that Hadoop be placed in a separate physical network with its own core switches.

Switch per rack

Hadoop supports the concept of rack locality and takes advantage of the network topology to minimize network congestion. Ideally, nodes in one rack should connect to a single physical switch. Two top- of- rack (ToR) switches can be used for high availability. Each ToR switch uplinks to a core switch with a significantly bigger backplane. Cloudera recommends 10 GbE (or faster) connections between the servers and ToR switches. ToR uplink bandwidth to the core switch (two switches in a HA configuration) can often be oversubscribed.

Uplink oversubscription

How much oversubscription is appropriate depends on the workload. Cloudera's recommendation is that the ratio between the total access port bandwidth and uplink bandwidth be as close to 1:1 as possible. This is especially important for heavy ETL workloads and MapReduce jobs that have a lot of data sent to reducers.

Oversubscription ratios up to 4:1 are generally fine for balanced workloads, but network monitoring is needed to ensure uplink bandwidth is not the bottleneck for Hadoop. The following table provides some examples as a point of reference:

Access Port Bandwidth (In Use)	Uplink Port Bandwidth (Bonded)	Ratio
48 x 1 GbE = 48 Gbit/s	4 x 10 GbE = 40 Gbit/s	1.2:1
24 x 10 GbE = 240 Gbit/s	2 x 40 Gig CFP = 80 Gbit/s	3:1
48 x 10 GbE = 480 Gbit/s	4 x 40 Gig CFP = 160 Gbit/s	3:1

⚠️ **Important:** Do not exceed 4:1 oversubscription ratio. For example, if a ToR has 20 x 10 GbE ports used, the uplink should be at least 50 Gbps. Different switches have dedicated uplink ports of specific bandwidth (often 40 Gbps or 100 Gbps) and therefore careful planning needs to be done to choose the right switch types.

Redundant network switches

Having redundant core switches in a full mesh configuration allows the cluster to continue operating in the event of a core switch failure. Redundant ToR switches prevent the loss of an entire rack of processing and storage capacity in the event of a ToR switch failure. General cluster availability can still be maintained in the event of the loss of a rack, as long as master nodes are distributed across multiple racks.

Accessibility

The accessibility of your CDP Private Cloud Base cluster is defined by the network configuration and depends on the security requirements and the workload. Typically, there are edge/client nodes that have direct access to the cluster. Users go through these edge nodes through the client applications to interact with the cluster and the data residing there. These edge nodes may be running a web application for real-time serving workloads, BI tools, or simply the Hadoop command-line client used to submit or interact with HDFS.

Cloudera recommends allowing access to the CDP Private Cloud Base cluster through edge nodes only. You can configure this in the security groups for the hosts that you provision. The rest of this document describes the various options in detail.

Internet connectivity

Clusters that do not require heavy data transfer between the internet or services outside of the immediate network and HDFS, might need access to services like software repositories for updates or other low-volume data sources located outside of the immediate network.

If you intend to leverage the multi-cloud/hybrid-cloud functionality in CDP, then you must ensure that adequate network bandwidth is present between your data centers and the public cloud vendors' networks. Details on this topic are out of scope of this document. Engage with your Cloud vendor's technical sales team and Cloudera Sales Engineering team to determine the requirements in such scenarios.

If you completely disconnect the cluster from the internet, you block access for software updates which makes maintenance difficult.

Cloudera Manager

Cloudera Manager is an application you use to manage, configure, and monitor CDP Private Cloud Base clusters and Cloudera Runtime services. Cloudera recommends that you install Cloudera Runtime services using Cloudera Manager.

You can install the Runtime services using parcels or native packages. A parcel is a binary distribution format. Parcels offer a number of benefits including consistency, flexible installation location, installation without sudo,

reduced upgrade downtime, rolling upgrades, and easy downgrades. Cloudera recommends using parcels, though using packages is supported.

Cluster sizing best practices

Review the best practices on how to size your physical hardware for setting up a CDP Private Cloud Base cluster.

Each worker node typically has a number of physical disks dedicated to raw storage for Hadoop. This number is used to calculate the total available storage for each cluster. Also, the calculations listed below assume that 10% disk space is allocated for YARN temporary storage. Cloudera recommends allocating 10-25% of the raw disk space for temporary storage as a general guideline. This can be changed within Cloudera Manager and should be adjusted after analyzing production workloads. For example, MapReduce jobs that send less data to reducers allow for adjusting this number percentage down considerably.

The following table contains example calculations for a cluster that contains 17 worker nodes. Each server has twelve 3 TB drives available for use by Hadoop. The table below outlines the available Hadoop storage based upon the number of worker nodes:

Table 1: Default replication factor

Raw Storage	612 TB
HDFS Storage (configurable)	550.8 TB
HDFS Unique Storage (default replication factor)	183.6 TB
MapReduce Intermediate Storage (configurable)	61.2 TB

Table 2: Erasure coding RS-6-3

Raw Storage	612 TB
HDFS Storage (configurable)	550.8 TB
HDFS Unique Storage (EC RS-6-3 -- 1.5x overhead)	367.2 TB
MapReduce Intermediate Storage (configurable)	61.2 TB

Table 3: Erasure Coding RS-10-4

Raw Storage	612 TB
HDFS Storage (configurable)	550.8 TB
HDFS Unique Storage (EC RS-10-4 -- 1.4x overhead)	393.4 TB
MapReduce Intermediate Storage (configurable)	61.2 TB

Note:

HDFS Unique Storage varies depending on the amount of data stored in EC directories and the RS policies. The tables above are examples of how different policies can affect HDFS Unique Storage.

Compressing raw data can effectively increase HDFS storage capacity.

While Cloudera Manager provides tools such as Static Resource Pools which utilize Linux Cgroups to allow multiple components to share hardware, in high volume production clusters, it can be beneficial to allocate dedicated hosts for roles such as Solr and Kafka.

Cluster hardware selection best practices

Read about a high-level overview of how different hardware component selections impact the performance of a Hadoop cluster.

See Hardware Requirements for detailed workload-specific practices.

Related Information
Hardware Requirements

Number of drives

Traditionally, Hadoop has been thought of as a large I/O platform. While there are many new types of workloads being run on Cloudera clusters that may not be as I/O bound as traditional MapReduce applications, it is still useful to consider the I/O performance when designing a Cloudera cluster.

Unlike the number of cores in a CPU and the density of RAM, the speed at which data can be read from a spinning hard drive (spindle) has not changed much in the past 10 years. To counter the limited performance of hard drive read/write operations, Hadoop reads and writes from many drives in parallel. Every additional spindle added to a node increases the overall read/write speed of the cluster.

> **Note:** SSDs have dramatically changed the persistent storage performance landscape, but the price per GB of spinning disks is still significantly less than that of SSD storage. As SSDs come down in cost and technologies such as Intel's Optane<tm> enter the market, workloads may swing back towards being CPU bound. Most Cloudera customers are still deploying clusters that store data on spinning hard disks.

Additional drives also come with the likelihood of more network traffic in the cluster. For the majority of cases, network traffic between nodes is generally limited by how fast data can be written to or read from a node. Therefore, the rule normally follows that, with more drives network speed requirements increase.

Generally speaking, the more drives a node has, the lower the cost per TB. However, the larger the quantity of data stored on one node, the longer the re-replication time if that node goes down. Hadoop clusters are designed to have many nodes. It is generally better to have more average nodes than fewer super nodes. This has a lot to do with both data protection, as well as increased parallelism for distributed computation engines such as MapReduce and Spark.

Lastly, the number of drives per node impacts the number of YARN containers configured for a node. YARN configuration and performance tuning is a complicated topic, but for I/O bound applications, the number of physical drives per host may be a limiting factor in determining the number of container slots configured per node.

Kafka clusters are often run on dedicated servers that do not run HDFS data nodes or processing components such as YARN and Impala. Because Kafka is a message-based system, fast storage and network I/O is critical to performance. Although Kafka does persist messages to disk, it is not generally necessary to store the entire contents of a Kafka topic log on the Kafka cluster indefinitely. Kafka brokers should be configured with dedicated spinning hard drives for the log data directories. Using SSDs instead of spinning disks has not shown to provide a significant performance improvement for Kafka.

Kafka drives should also be configured as RAID 10 because the loss of a single drive on a Kafka broker can cause the broker to experience an outage.

Related Information
Intel Optane Technology for Data Centers
YARN tuning overview

Disk layout

Review the recommended layout for master and worker nodes.

The following layout is recommended for master nodes:

- 2 x Disks (capacity at least 500 GB) in RAID 1 (software or hardware) for operating system (OS) and logs.
- 4 x Disks (>= 1 TB each) in RAID 10 for Database data (see Note).
- 2 x Disks (capacity at least 1 TB) in RAID 1 (software or hardware) for NameNode metadata.

- 1 x Disk JBOD/RAID 0 for ZooKeeper (>= 1TB) (see Note).
- 1 x Disk JBOD/RAID 0 for Quorum JournalNode (>= 1 TB).

⚠ **Important:** Ideally, databases should be run on an external host rather than running on the master node.

📝 **Note:** If a customer has experienced fsync delays and other I/O related issues with ZooKeeper, ZooKeeper's dataDir and dataLogDir can be configured to use separate disks. It is hard to determine ahead of time whether this will be necessary; even a small cluster can result in heavy ZooKeeper activity.

The following layout is recommended for worker nodes:

- 2x Disks (capacity at least 500 GB) in RAID 1 (software or hardware) for OS and logs.
- 12-24 SATA Disks JBOD mode (or as multiple single-drive RAID 0 volumes if using a RAID controller incapable of doing JBOD passthrough) no larger than 4 TB in capacity. If the RAID controller has cache, use it for write caching (preferably with battery backup) and disable read caching. Follow your hardware vendor's best practices where available.
- For a higher performance profile, use 10K RPM SATA or faster SAS drives. These often have lower capacity, but capacity considerations can be offset by adding more data nodes.

RAID controllers should be configured to disable any optimization settings for the RAID 0 volumes.

⚠ **Important:** Do not create multi-disk RAID 0 volumes.

Data density per drive

Hard drives come in many sizes today. Popular drive sizes are 1-4 TB, although larger drives are more common now. When picking a drive size, consider the cost per TB, replication storm in case of drive failure, and cluster performance.

- Lower Cost Per TB: The larger the drive, the cheaper the cost per TB, which makes for lower TCO.
- Replication Storms: Larger drive means drive failures will produce larger re-replication storms, which can take longer and saturate the network while impacting in-flight workloads.
- Cluster Performance: In general, drive size has little impact on cluster performance. The exception is when drives have different read/write speeds and a use case that leverages this gain. MapReduce is designed for long sequential reads and writes, so latency timings are generally not as important. HBase can potentially benefit from faster drives, but that is dependent on a variety of factors, such as HBase access patterns and schema design; this also implies acquisition of more nodes. Impala and Cloudera Search workloads can also potentially benefit from faster drives, but for those applications the ideal architecture is to maintain as much data in memory as possible.

Cloudera does not support more than 100 TB per HDFS data node. You can use 12 x 8 TB spindles or 24 x 4 TB spindles. Cloudera does not support drives larger than 8 TB for HDFS data.

📝 **Note:** Larger disks offer increased capacity but not increased I/O throughput. Clusters with larger disks can easily result in capacities exceeding 100 TB per-worker, contributing to replication storms mentioned above. Clusters with larger disks that observe the 100 TB limit end up having fewer spindles which reduces HDFS throughput.

⚠ **Important:** Running CDP Private Cloud Base on storage platforms other than direct-attached physical disks can provide suboptimal performance. Cloudera Runtime and the majority of the Hadoop platform are optimized to provide high performance by distributing work across a cluster that can utilize data locality and fast local I/O. See the Cloudera Enterprise Storage Device Acceptance Criteria Guide for more information about using non-local storage.

Number of cores and multithreading

Other than cost, there is no negative for buying more and better CPUs. However, the return of investment on additional CPU power must be evaluated carefully.

Note: It might help to consider the latest Cloudera licensing model as different models might fit for different use-cases.

Following are some points to consider:

- Cluster Bottleneck: In general, CPU resources (and lack thereof) are not the bottleneck for MapReduce and HBase. Almost always, the bottleneck is drive and/or network performance. Certainly, there are exceptions to this, such as inefficient Hive queries. Other compute frameworks like Impala, Spark, and Cloudera Search may be CPU-bound depending on the workload.
- Additional Cores/Threads: Within a given MapReduce job, a single task typically uses one thread at a time. With Spark this is different, because a single task might use multiple threads in parallel. As outlined earlier, the number of slots allocated per node may be a function of the number of drives in the node. A more careful evaluation is required in scenarios where the Virtual Private Cluster model is being considered. In such scenarios, if storage and compute are clearly distinct roles, then requirements might vary significantly. As long as there is no huge disparity in the number of cores (threads) and the number of drives, there is no need for additional cores. In addition, a MapReduce task is going to be I/O bound for typical jobs. Thus, a given thread used by the task will have a large amount of idle time while waiting for an I/O response.
- Clock Speed: Because Cloudera clusters often begin with a small number of use cases and associated workloads and grow over time, it makes sense to purchase the fastest CPUs available. Actual CPU usage is use case and workload dependent. For instance, computationally intensive Spark jobs benefit more from faster CPUs than I/O bound MapReduce applications.

Important: Allocate two vCPUs for the operating system and other non-Hadoop use (although this amount may need to be higher if additional non-Hadoop applications are running on the cluster nodes, such as third-party active monitoring/alerting tools). The more services you are running, the more vCPUs are required; you need to use more capable hosts to accommodate these needs.

For worker nodes, a mid-range 12-14 core CPU running at 2.4-2.5 GHz provides a good cost/performance tradeoff. For master nodes, a mid-range 8 core CPU with a faster clock speed (For example 2.6 GHz) can suffice. Where available, Simultaneous Multi-Threading implementations should be enabled (for example Intel's HyperThreading). BIOS settings for CPU and memory should be set to Maximum Performance mode or equivalent.

See Hardware Requirements for detailed workload-specific practices.

Related Information

Hardware Requirements

Virtual Private Clusters and Cloudera SDX

RAM

More memory is always good and it is recommended to purchase as much as the budget allows. Applications such as Impala and Cloudera Search are often configured to use large amounts of heap, and a mixed workload cluster supporting both services should have sufficient RAM to allow all required services.

Important: Allocate at least 4 GB memory for the operating system and other non-Hadoop use (although this amount may need to be higher if additional non-Hadoop applications are running on the cluster nodes, such as third-party active monitoring/alerting tools). The more services you run, the more memory is required; you need to use more capable hosts to accommodate these needs.

It is critical to performance that the total memory allocated to all Hadoop-related processes (including processes such as HBase) is less than the total memory on the node, taking into account the operating system and non-Hadoop processes. Oversubscribing the memory on a system can lead to the Linux kernel's out-of-memory process killer being invoked and important processes being terminated. Performance might be affected by over-allocation of memory to a Hadoop process, as this can lead to long Java garbage collection pauses. For processes such as HBase, you must factor in aspects such as off heap bucket cache configuration.

For optimum performance, confer with your hardware vendor for defining optimal memory configuration layout.

While 128 GB RAM can be accommodated, this typically constrains the amount of memory allocated to services such as YARN and Impala, thereby reducing the query capacity of the cluster. 256 GB RAM is recommended with higher values also possible.

Related Information

Off-heap BucketCache

Power supplies

Hadoop software is designed around the expectation that nodes can fail. Redundant hot-swap power supplies are not necessary for worker nodes, but should be used for master, utility, and edge nodes.

If you are using a single power supply on worker nodes, confer with your data center team to alternate the power feeds for each rack. For example, Rack 1 goes on Feed A, Rack 2 goes on Feed B, Rack 3 goes on Feed C, and continuing the pattern.

Operating system best practices

Read about the operating system best practices.

Cloudera supports running the platform only on Linux. Supported distributions include Red Hat Enterprise LInux, Ubuntu, and Suse. Refer to the Support Matrix for specific OS version details. To receive support from Cloudera, a supported version of the operating system must be in use. The Requirements and Supported Versions guide lists the supported operating systems for each version of Cloudera Manager and CDP Runtime.

Review the planning information before installing CDP Private Cloud Base.

See Runtime cluster hosts and role assignments for service layout guidelines.

Related Information

CDP Private Cloud Base Requirements and Supported Versions

Before You Install

Runtime Cluster Hosts and Role Assignments

Hostname naming convention

Cloudera recommends using a hostname convention that allows for easy recognition of roles and/or physical connectivity. This is especially important for configuring rack awareness within Cloudera Manager.

Using a project name identifier, followed by the rack ID, the machine class, and a machine ID is an easy way to encode useful information about the cluster. For example, "acme-test-r01m01".

This hostname represents the ACME customer's test project, rack #1, and master node #1.

Hostname resolution

Cloudera recommends using DNS for hostname resolution. The usage of /etc/hosts becomes cumbersome quickly, and is routinely the source of hard-to-diagnose problems.

/etc/hosts should contain an entry for 127.0.0.1, and localhost should be the only name that resolves to it. The file should also contain an entry for the system's IP address, FQDN, and shortname. The machine name must not resolve to the 127.0.0.1 address. All hosts in the cluster must have forward and reverse lookups to be the inverse of each other. An easy test to perform on the hosts to ensure proper DNS resolution is to run the following commands:

```
dig [***HOSTNAME***]
dig -x [***IP-ADDRESS-RETURNED-FROM-HOSTNAME-LOOKUP***])
```

For example:

```
dig themis.apache.org
themis.apache.org. 1758 IN A 140.211.11.105

dig -x 140.211.11.105
105.11.211.140.in-addr.arpa. 3513 IN  PTR themis.apache.org.
```

This is the acceptable behavior for every host in the cluster.

Also, enable nscd with only hostname caching enabled for a 30-60 second period. This further reduces heavy DNS impact. This is a mitigation technique for preventing the overload of AD DNS, for example, which can fail over during high DNS load.

```
enable-cache passwd no
enable-cache group no
enable-cache hosts yes
positive-time-to-live hosts 60
negative-time-to-live hosts 20
```

Functional accounts

Cloudera Manager and CDP Private Cloud Base use dedicated functional accounts for the associated daemon processes. By default, these accounts are created as local accounts on every machine in the cluster that needs them if they do not already exist (locally or from a directory service, such as LDAP).

Kerberos deployment models (including identity integration with Active Directory) are covered in detail within the Authentication documentation. Refer to the CDP Security Reference Architecture for more details about Kerberos and related security best practices.

Related Information
Hadoop Users (user:group) and Kerberos Principals

Authentication Overview

CDP Security Reference Architecture

Time and date

All machines in the cluster need to have the same time and date settings, including time zones. Use of the Network Time Protocol (NTP) is recommended. Many cluster services are sensitive to time (For example HBase, Kudu, and ZooKeeper) and troubleshooting is easier if time is consistent across all hosts.

You can enable the NTP daemon by running the following commands on RHEL and CentOS 7 operating systems:

```
systemctl start ntpd.service
systemctl enable ntpd.service
```

Note: Chrony may be preferred on newer operating systems.

Name service caching

Cloudera recommends that you enable name service caching, particularly for clusters that use non-local Hadoop functional accounts, such as the hdfs and yarn users. This becomes critical in the case where the latter is combined with using Kerberos. Many difficult-to-diagnose problems can arise when name service lookups time out or fail during heavy cluster utilization.

Enable the Name Service Cache Daemon (nscd) by running the following commands on RHEL and CentOS 7 operating systems:

```
systemctl start nscd.service
systemctl enable nscd.service
```

If you are running Red Hat SSSD, then you must modify the nscd configuration to not cache the passwd, group, or netgroup information.

Related Information
Using NSCD with SSSD

SELinux

CDP Private Cloud Base is supported on platforms with Security-Enhanced Linux (SELinux) enabled in permissive mode. However Cloudera recommends SELinux be disabled on all machines in the Hadoop cluster until the cluster is up and running.

The Linux command getenforce returns the status of SELinux.

SELinux can be disabled on RHEL or CentOS by editing /etc/selinux/config and setting SELINUX=disabled. This change must be done as root (or with proper sudo access), and requires a reboot.

> **Note:**
>
> Cloudera does not provide SELinux policies to enable enforcing mode. This is beyond the scope of what Cloudera tests and validates.
>
> Customers requiring this must carefully evaluate the audit details from permissive mode and create customer-specific policies accordingly. If issues arise in the platform due to SELinux enforcing policies, you may be requested to disable them by Cloudera Support when opening support cases.

IPv6

Hadoop and other components in the platform do not support IPv6. IPv6 configurations should be removed, and IPv6-related services must be stopped.

Host-based firewalls

Cloudera recommends disabling host-based firewalls on the cluster, until the cluster is up and running. Many problems that are difficult to diagnose result from incorrect or conflicting host-based firewall entries that interfere with normal cluster communication.

Run the following commands to disable host-based firewalls for both IPv4 and IPv6 on RHEL and CentOS operating systems:

```
systemctl stop firewalld.service
systemctl disable firewalld.service
```

For those who must restrict access using host-based firewalls, see the list of ports used by Cloudera Manager, Cloudera Runtime components, managed services, and third-party components.

Related Information
Ports

Startup services

As with any production server, unused services that have been enabled by default during startup should be removed or disabled.

Some example services that are enabled by default and not needed by the CDP Private Cloud Base are:

• bluetooth
• cups
• iptables
• ip6tables
• postfix

> **Note:** While not needed by CDP Private Cloud Base, postfix (or other MTA) may be required by other services to deliver generated notices/alerts from the system.

This list is not exhaustive. To view the list of services that are configured to start during system startup, run the following command on RHEL and CentOS operating systems:

```
systemctl list-unit-files --type service | grep enabled
```

Process memory

The memory on each node is allocated to the various Hadoop processes. This predictability reduces the chance of Hadoop processes inadvertently running out of memory and paging to disk, which in turn leads to severe degradation in performance.

A minimum of 4 GB of memory should be reserved on all nodes for operating system and other non-Hadoop use. This amount may need to be higher if additional non-Hadoop applications are running on the cluster nodes, such as third-party active monitoring and alerting tools.

Memory requirements and allocation for Hadoop components are discussed in further detail in other sections of this document. See Kernel and OS Tuning for more information.

Kernel and OS tuning

The Cloudera Runtime platform depends on a tuned underlying host operating system (OS) for optimal performance. Cloudera suggests setting the vm.swappiness and transparent hugepage compaction kernel parameters.

For additional background information and suggested settings, see Optimizing Performance in Cloudera Runtime.

Entropy

Cryptographic operations require entropy to ensure randomness. The Cloudera Security guide explains how to check available entropy and how to ensure sufficient entropy is available: Entropy Requirements.

Networking parameters

The Transmit and Receive ring buffer sizes on the ethernet interfaces of every node of the cluster should be adjusted to ensure higher throughput rates. Check existing ring buffer sizes by running the following command:

```
ethtool -g eth0
Ring parameters for eth0:
Pre-set maximums:
RX:  4096
RX Mini: 0
RX Jumbo: 0
TX:  4096
Current hardware settings:
RX:  256
RX Mini: 0
RX Jumbo: 0
TX:  256
```

After checking the preset maximum values and the current hardware settings, run the following commands to resize the ring buffers:

```
ethtool -G [***INTERFACE***] rx [***NEWSIZE***]
```

or

```
ethtool -G [***INTERFACE***] tx [***NEWSIZE***]
```

Most modern enterprise-grade network adapters have several performance optimization features, such as offload capabilities and large segmentation offload, which reduce load on host CPU by handling these functions at the network interface level which can be explored as part of performance optimization initiatives. An iterative approach is recommended while applying a standard load generator such as the Terasuite benchmarks, to test the effect of

enabling said features. Performance optimization parameters should not be applied indiscriminately, without thorough testing, and should only be applied based on genuine need.

> **Note:** The performance tuning guidelines provided in this document are meant to be applied in an iterative manner, along with sufficient testing. Not all parameters specified may be applicable. These are general best practices but details may vary based on infrastructure being used as well as application workload patterns. When in doubt, see your equipment vendor documentation.

The following parameters should be added to /etc/sysctl.conf to optimize various network behaviors:

- Disable TCP timestamps to improve CPU utilization (this is an optional parameter and depends on your NIC vendor):

```
net.ipv4.tcp_timestamps=0
```

- Enable TCP sacks to improve throughput:

```
net.ipv4.tcp_sack=1
```

- Increase the maximum length of processor input queues:

```
net.core.netdev_max_backlog=250000
```

- Increase the TCP max and default buffer sizes using setsockopt():

```
net.core.rmem_max=4194304
net.core.wmem_max=4194304
net.core.rmem_default=4194304
net.core.wmem_default=4194304
net.core.optmem_max=4194304
```

- Increase memory thresholds to prevent packet dropping:

```
net.ipv4.tcp_rmem=4096 87380 4194304
net.ipv4.tcp_wmem=4096 65536 4194304
```

> **Note:** If you want to run this from the command line, then quote the values being set. For example:
>
> ```
> sysctl -w net.ipv4.tcp_rmem="4096 87380 4194304"
> ```

Related Information
Optimizing Performance in Cloudera Runtime
Entropy
Entropy Requirements
Terasuite benchmarks

Swapiness parameter configuration

The kernel parameter that determines the tendency of the kernel to swap pages in and out of the main memory is vm.swappiness. By default, on RHEL 7 and variants, the value is set to 60 (range 0-99). This is not appropriate for Hadoop clusters. This needs to be tuned down to 1 to avoid unnecessary page outs and long garbage collection pauses.

Run the following command to set the vm.swapiness parameter to 1:

```
sudo sysctl -w vm.swappiness=1
```

To make the vm.swapiness parameter persist across reboots, add an entry in the /etc/sysctl.conf file as follows:

```
vm.swappiness=1
```

Filesystems

In Linux, there are several choices for formatting and organizing drives. However, only a few choices are optimal for Hadoop.

In RHEL and CentOS, the Logical Volume Manager (LVM) should not be used for data drives. It is not optimal and can lead to combining multiple drives into one logical disk, which is in complete contrast to how Hadoop manages fault tolerance across HDFS. It is beneficial to keep LVM enabled on the OS drives. Any performance impact that may occur is countered by the improvement of system manageability. Using LVM on the OS drives enables the admin to avoid over-allocating space on partitions. Space needs can change over time and the ability to dynamically grow a filesystem is better than having to rebuild a system. Do not use LVM to stripe or span logical volumes across multiple physical volumes to mimic RAID.

Cloudera recommends using an extent-based filesystem. This includes ext3, ext4, and xfs. Most new Hadoop clusters use the ext4 filesystem by default. RHEL 7 uses xfs as its default filesystem.

Note: If using Kudu, ensure that filesystem hole punching is a capability of the filesystem. Hole punching is the use of the fallocate() system call with the FALLOC_FL_PUNCH_HOLE option. Newer versions of ext4 and xfs support Hole Punching. ext3 does not support hole punching and unpatched RHEL prior to 6.4 does not support this facility. Older versions of ext4 and xfs that do not support hole punching cause Kudu to fail to start because Kudu provides a pre-start test for this facility. Without the hole punching support, the block manager is unsafe to use because claimed blocks are released and more disk space is consumed.

Filesystem creation options

When creating ext4 filesystems for use with Hadoop data volumes, Cloudera recommends reducing the superuser block reservation from 5% to 1% for root (using the -m1 option) as well as setting the following options:

- use one inode per 1 MB (largefile)
- minimize the number of super block backups (sparse_super)
- enable journaling (has_journal)
- use b-tree indexes for directory trees (dir_index)
- use extent-based allocations (extent)

Run the following command for creating an ext4 filesystem:

```
mkfs –t ext4 –m 1 –O -T largefile sparse_super,dir_index,extent,has_journal
/dev/sdb1
```

Run the following command for creating an xfs filesystem:

```
mkfs –t xfs /dev/sdb1
```

Note: You need not specify any options when creating an xfs filesystem.

Disk mount options

By design, HDFS is a fault-tolerant filesystem. All drives used by DataNode machines for data need to be mounted without the use of RAID. Drives should be mounted in the /etc/fstab filesystem table using the noatime option (which also implies nodiratime). In case of SSD or flash, turn on TRIM by specifying the discard option when mounting. This reduces premature SSD wear and device failures, while primarily avoiding long garbage collection pauses.

In the /etc/fstab filesystem table, ensure that the appropriate filesystems have the noatime mount option specified:

```
/dev/sda1 /              ext4    noatime           0 0
```

To enable TRIM, edit the /etc/fstab filesystem table and also set the discard mount option:

```
/dev/sdb1 /data          ext4      noatime,discard          0 0
```

Disk mount naming convention

For ease of administration, it is recommended to mount all of the disks on the DataNode machines with a naming pattern, such as the following:

```
/data1
/data2
/data3
/data4
/data5
/data6
```

Cluster configuration

Learn about configuring the cluster, including Cloudera recommendations specific to the customer hardware being used, best practices, and general recommendations for each Cloudera Runtime service. This section is not an exhaustive description of every configuration, but rather focuses on important configurations that have been changed from the default setting.

TeraGen and TeraSort performance baseline

The TeraGen and TeraSort benchmarking tools are part of the standard Apache Hadoop distribution and are included with the Cloudera distribution.

In the course of a cluster installation or certification, Cloudera recommends running several TeraGen and TeraSort jobs to obtain a performance baseline for the cluster. The intention is not to demonstrate the maximum performance possible for the hardware or to compare with externally published results, because tuning the cluster for this may be at odds with actual customer operational workloads. Rather the intention is to run a real workload through YARN to functionally test the cluster as well as obtain baseline numbers that can be used for future comparison, such as in evaluating the performance overhead of encryption features or in evaluating whether operational workload performance is limited by the I/O hardware. Running the benchmarks provides an indication of cluster performance and may also identify and help diagnose hardware or software configuration problems by isolating hardware components, such as disks and network, and subjecting them to a higher than normal load.

The TeraGen job generates an arbitrary amount of data, formatted as 100-byte records of random data, and stores the result in HDFS. Each record has a random key and value. The TeraSort job sorts the data generated by TeraGen and writes the output to HDFS.

During the first iteration of the TeraGen job, the goal is to obtain a performance baseline on the disk I/O subsystem. The HDFS replication factor should be overridden from the default value 3 and set to 1 so that the data generated by the TeraGen job is not replicated to additional data nodes. Replicating the data over the network obscures the raw disk performance with potential network bandwidth constraints.

Once the first TeraGen job has been run, a second iteration should be run with the HDFS replication factor set to the default value. This applies a high load on the network, and deltas between the first run and second run can provide an indication of network bottlenecks in the cluster.

While the TeraGen application can generate any amount of data, 1 TB is standard. For larger clusters, it may be useful to also run 10 TB or even 100 TB, because the time to write 1 TB may be negligible compared to the startup overhead of the YARN job. Another TeraGen job should be run to generate a dataset that is 3 times the RAM size of the entire cluster. This ensures you are not seeing page cache effects and are exercising the disk I/O subsystem.

The number of mappers for the TeraGen and TeraSort jobs should be set to the maximum number of disks in the cluster. This is less than the total number of YARN vcores available, so it is advisable to temporarily lower the vcor es available per YARN worker node to the number of disk spindles to ensure an even distribution of the workload. An additional vcore is needed for the YARN ApplicationMaster.

The TeraSort job should also be run with the HDFS replication factor set to 1 as well as with the default replication factor. The TeraSort job hardcodes a DFS replication factor of 1, but it can be overridden or set explicitly by specifying the mapreduce.terasort.output.replication parameter as follows:

TeraGen command to generate 1 TB of data with HDFS replication set to 1

The following sample command generates 1 TB of data with an HDFS replication factor of 1, using 360 mappers. This command is appropriate for a cluster with 360 disks:

```
EXAMPLES_PATH=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce

yarn jar ${EXAMPLES_PATH}/hadoop-mapreduce-examples.jar \
   teragen -Ddfs.replication=1 -Dmapreduce.job.maps=360 \
   10000000000 TS_input1
```

TeraGen command to generate 1 TB of data with HDFS default replication

The following sample command generates 1 TB of data with the default HDFS replication factor (usually 3), using 360 mappers. This command is appropriate for a cluster with 360 disks:

```
EXAMPLES_PATH=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce

yarn jar ${EXAMPLES_PATH}/hadoop-mapreduce-examples.jar \
   teragen -Dmapreduce.job.maps=360 \
   10000000000 TS_input2
```

TeraSort command to sort data with HDFS replication set to 1

The following sample command sorts the data generated by TeraSort using 360 mappers and writes the sorted output to HDFS with a replication factor of 1. This is appropriate for a cluster with 360 disks:

```
EXAMPLES_PATH=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce

yarn jar ${EXAMPLES_PATH}/hadoop-mapreduce-examples.jar \
   terasort -Ddfs.replication=1 \
   -Dmapreduce.job.maps=360 \
   TS_input1 TS_output1
```

TeraSort command to sort data with HDFS replication set to 3

The following sample command sorts the data generated by TeraSort using 360 mappers and writes the sorted output to HDFS with a replication factor of 3 (a typical default). This is appropriate for a cluster with 360 disks:

```
EXAMPLES_PATH=/opt/cloudera/parcels/CDH/lib/hadoop-mapreduce

yarn jar ${EXAMPLES_PATH}/hadoop-mapreduce-examples.jar \
   terasort -Dmapreduce.job.maps=360 -Ddfs.replication=3 \
   TS_input2 TS_output2
```

Cloudera recommends that you record the results of TeraSort and TeraGen (number of mappers and run time) for future comparisons in the following table:

Table 4: TeraGen and TeraSort results

Command	HDFS replication	Number of mappers	Run time
Teragen for 3x cluster RAM data set	1		
Terasort for 1 TB data set	1		
Terasort for 3x cluster RAM data set	1		
Teragen for 1 TB data set	3		
Teragen for 3x cluster RAM data set	3		
Terasort for 1 TB data set	3		
Terasort for 3x cluster RAM data set	3		
Teragen for 1 TB data set	1		

Cluster configuration best practices

Review the cluster configuration best practices.

ZooKeeper

Learn why it is recommended to install ZooKeeper on a node where it can have unobstructed access to the disk.

ZooKeeper is sensitive to disk latency. While it only uses a modest amount of resources, having ZooKeeper swap out or wait for a disk operation can result in that ZooKeeper node being considered 'dead' by its quorum peers. For this reason, Cloudera recommends against deploying ZooKeeper on worker nodes where loads are unpredictable and are prone to spikes. It is acceptable to deploy Zookeeper on master nodes where load is more uniform and predictable (or on any node where it can have unobstructed access to disk).

HDFS

Learn about the various considerations and bottlenecks when planning cluster configuration for the HDFS service.

Java heap sizes

The NameNode memory should be increased over time as HDFS has more files and blocks stored. Cloudera Manager can monitor and alert on memory usage. Roughly, the NameNode needs 1 GB of memory for every 1 million files. Setting the heap size too large when it is not needed leads to inefficient Java garbage collection, which can lead to erratic behavior that is hard to diagnose. NameNode and Standby NameNode heap sizes must always be the same, and must be adjusted together.

NameNode metadata locations

When a quorum-based high availability HDFS configuration is used, JournalNodes handle the storage of metadata writes. The NameNode daemons require a local location to store metadata. Cloudera recommends that only a single directory be used if the underlying disks are configured as RAID, or two directories on different disks if the disks are mounted as JBOD.

Block size

HDFS stores files in blocks that are distributed over the cluster. A block is typically stored contiguously on disk to provide high read throughput. The choice of block size influences how long these high throughput reads run, and over how many nodes a file is distributed. When reading the many blocks of a single file, a small block size spends

more overall time in slow disk seek, and a large block size has reduced parallelism. Data processing that is I/O heavy benefits from larger block sizes, and data processing that is CPU heavy benefits from smaller block sizes.

The default provided by Cloudera Manager is 128 MB. The block size can also be specified by an HDFS client on a per-file basis.

Replication factor

Bottlenecks can occur on a small number of nodes when only small subsets of files on HDFS are being heavily accessed. Increasing the replication factor of the files so that their blocks are replicated over more nodes can alleviate this. This is done at the expense of storage capacity on the cluster. This can be set on individual files, or recursively on directories with the -R parameter, by using the Hadoop shell command hadoop fs -setrep. By default, the replication factor is 3.

Erasure Coding

Erasure Coding (EC) is an alternative to the 3x replication scheme. EC levies additional demands on the number of nodes or racks required to achieve fault tolerance:

- node-level: number of DataNodes needed to equal or exceed data stripe width
- rack-level: number of racks needed to equal or exceed data stripe width

For example, for a RS-10-4 policy to be rack-failure tolerant, you need at least 14 racks (10 for data blocks, 4 for parity blocks); for host-failure tolerance you need at least 14 nodes. EC observes rack topology, but the resulting block placement policy (BPP) differs from replication. With EC, the BPP tries to place all blocks as evenly as possible on all racks. Cloudera recommends that racks have a consistent number of nodes. Racks with fewer DataNodes are busier and fill faster than racks with more DataNodes.

> ⚠ **Important:** Impala and HBase queries fail if they attempt to access Erasure Coded data.

Rack awareness

Hadoop optimizes performance and redundancy when rack awareness is configured for clusters that span across multiple racks, and Cloudera recommends doing so. You can assign racks for nodes using Cloudera Manager.

When setting up a multi-rack environment, place each master node on a different rack. In the event of a rack failure, the cluster continues to operate using the remaining master(s).

DataNode failed volumes tolerated

By default, Cloudera Manager sets the HDFS DataNode failed volume threshold to half of the data drives in a DataNode. This is configured using the dfs_datanode_failed_volumes_tolerated HDFS property in Cloudera Manager. If each DataNode has eight drives dedicated to data storage, this threshold is set to four, meaning that HDFS marks the DataNode dead on the fifth drive failure. This number may need to be adjusted up or down depending on internal policies regarding hard drive replacements, or because of evaluating what behavior is actually seen on the cluster under normal operating conditions. Setting the value too high has a negative impact on the Hadoop cluster. Specifically for YARN, the number of total containers available on the node with many drive failures is the same as nodes without drive failures, meaning data locality is less likely on the former, leading to more network traffic and slower performance.

> ⚠ **Important:** Multiple drive failures in a short amount of time can indicate a larger problem with the machine, such as a failed disk controller.

DataNode max transfer threads count

This parameter replaces the deprecated dfs.datanode.max.xcievers parameter, and needs to be adjusted for workloads like HBase to ensure that the DataNodes serve adequate number of files at any one time. Failure to do so can result in error messages about exceeding the number of transfer threads or missing blocks.

Balancing

HDFS spreads data evenly across the cluster to optimize read access, MapReduce performance, and node utilization. Over time, it is possible that the data distribution in the cluster can go out of balance due to various reasons. Hadoop can help mitigate this by rebalancing data across the cluster using the balancer tool. You can run the balancer tool manually using Cloudera Manager or from the command line. By default, Cloudera Manager configures the balancer to rebalance a DataNode when its utilization is 10% more or less from the average utilization across the cluster. Individual DataNode utilization can be viewed from Cloudera Manager.

By default, the maximum bandwidth a DataNode uses for rebalancing is set to 1 MB/second (8 Mbit/second). This can be increased but network bandwidth used by rebalancing could potentially impact production cluster application performance. Changing the balancer bandwidth setting within Cloudera Manager requires a restart of the HDFS service: however, this setting can also be made instantly across all nodes without a configuration change by running the following command as an HDFS superuser:

```
hdfs dfsadmin -setBalancerBandwidth [***BYTES-PER-SECOND***]
```

This is a convenient way to change the setting without restarting the cluster, but since it is a dynamic change, it does not persist if the cluster is restarted. See Recommended configurations for the Balancer for more insights into scenarios and suggested values for tuning.

> **Note:**
>
> Cloudera does not recommend running the balancer on an HBase cluster as it affects data locality for the RegionServers, which can reduce performance. Unfortunately, when HBase and YARN services are colocated and heavy usage is expected on both, there is no good way to ensure that the cluster is optimally balanced.
>
> If HDFS rebalancing is required on HBase clusters, then you may also need to run an HBase major compaction operation when the rebalancing is completed. This can help restore region locality.

You can configure HDFS to distribute writes on each DataNode in a manner that balances out available storage among that DataNode's disk volumes. By default, a DataNode writes new block replicas to disk volumes solely on a round-robin basis. You can configure a volume-choosing policy that causes the DataNode to take into account how much space is available on each volume when deciding where to place a new replica.

For more information, see Configure storage balancing for DataNodes using Cloudera Manager.

DataNode Disks/Data Directories

Use the disks in non-RAID/JBOD (pass-through) mode.

- Do not use RAID/LVM/ZFS to combine multiple disks into one volume.
- Combining multiple disks to one volume causes DNs to store more data onto the disks but during boot up time, it sends one massive block report instead of one per storage disk and it is not multi-threaded. This delays boot up time of the DNs.
- Block verification (checking blocks against bit-rot) is single threaded.

> **Note:** Cloudera recommends setting the data volume mount points to be immutable (chattr +i). This will prevent the DataNode from writing to the root filesystem in the event a data volume fails to mount.

Related Information

Data durability and Erasure coding overview

Specifying Racks for Hosts

Using dfs.datanode.max.transfer.threads with HBase

Recommended configurations for the Balancer

Configure storage balancing for DataNodes using Cloudera Manager

YARN

The YARN service manages MapReduce and Spark tasks. Applications run in YARN containers, which use Linux Cgroups for resource management and process isolation.

See YARN tuning overview for more details.

Related Information

YARN tuning overview

Fine-Tune Fair to Capacity Scheduler in Weight Mode

Fine-Tune Fair to Capacity Scheduler in Relative Mode

Managing and Allocating Cluster Resources using Capacity Scheduler

Impala

The Impala service is a distributed, MPP database engine for interactive performance of SQL queries over large data sets. Impala performs best when it can operate on data in memory. Therefore, Impala is often configured with a very large heap size.

Impala daemon processes must be colocated with HDFS data nodes to use HDFS local reads, which also improve performance.

Impala does not provide any built-in load balancing, so a production Impala deployment should be deployed behind a load balancer for performance and high availability. For more information on configuring Impala with a load balancer, see Configuring Load Balancer for Impala.

Before deploying Impala in a production environment, review the Impala performance best practices.

Related Information

Configuring Load Balancer for Impala

Impala Performance Best Practices

Spark

Cloudera supports Spark on YARN-managed deployments for a more flexible and consistent resource management approach.

When running under Spark, the number of executors (YARN containers) can be specified when submitting the Spark job. By default, dynamic allocation is enabled but can be disabled by setting the spark.dynamicAllocation.enabled parameter in Cloudera Manager to false. If you specify the --num-executors option in the job, then the dynamic allocation is disabled implicitly. For more information on Spark configuration and management, see Configuring Apache Spark.

Deploying Spark in a standalone mode is not supported.

Related Information

Configuring Apache Spark

Spark 3 Properties in Cloudera Runtime 7.1.8

HBase

By default, major compactions happen every 7 days. The next major compaction happens 7 days after the last one has finished. This means that the actual time that major compaction happens can impact production processes, which is not ideal if it is desired to run compactions at a specific known off-peak hour, such as at 3 AM.

Cloudera recommends that you disable automatic major compaction by setting the interval to zero in Cloudera Manager (hbase.hregion.majorcompaction = 0). Major compactions should then be run using cron jobs by calling the HBase admin tool.

Related Information

HBase HRegion Major Compaction property in in Cloudera Runtime 7.1.8

Search

Cloudera Search is a service based on Apache Solr. It provides a distributed search engine service. Search engines are often expected to provide fast, interactive performance so it is important to allocate sufficient RAM to the Search service

If other resource intensive applications, such as Impala, are deployed on the same cluster, then use the resource management facilities available in Cloudera Manager for Search. In some cases, it may also be preferable to avoid colocating the Search service with other services.

Oozie

Writing Oozie XML configuration files can be tedious and error-prone. Cloudera recommends that you use the Oozie editor that is embedded in Hue for creating, scheduling, and executing Oozie workflows.

Kafka

Kafka's default configuration with Cloudera Manager is suited to start development quickly. Several default settings should be changed before deploying a Cloudera Kafka cluster in production.

The default ZooKeeper Kafka root is /. However, Cloudera recommends changing this to /kafka. This is the location within ZooKeeper where the znodes for the Kafka cluster are stored. As long as a single Kafka cluster is using the ZooKeeper service, using /kafka is recommended. If multiple Kafka clusters, for example, the development, test, and QA teams are sharing a ZooKeeper service, then each Kafka instance should have a unique ZooKeeper root. For example, /kafka-dev, /kafka-test, /kafka-qa.

Cloudera Manager enables automatic creation of Kafka topics by default. If some data is written to a Kafka topic, then this operation creates that topic, if that topic does not exist already. While this may be convenient in prototyping and development, auto-creation of Kafka topics should not be used in production environments. Auto-creation can lead to creation of arbitrary topics and data can be written to the wrong topic, in case the application is configured incorrectly.

Cloudera Manager sets the default minimum number of in-sync replicas (ISR) to 1. This should generally be increased to a minimum of 2 in a production cluster to prevent data loss.

The Kafka Maximum Process File Descriptors setting may need to be increased in certain production deployments. This value can be monitored in Cloudera Manager and increased if usage requires a larger value than the default 64 k limit.

The default data retention time for Kafka is acceptable for production deployments, but it should be reviewed based on the use case.

Kudu

Review the partitioning guidelines and limitations before deploying the Kudu service on your cluster.

Partitioning guidelines

Kudu supports partitioning tables by RANGE and HASH partitions. The RANGE and HASH partitions can be combined to create effective partitioning strategies. It is also possible to utilize non-covering RANGE partitions.

For large tables, such as fact tables, aim for as many tablets as the cores in the cluster.

For small tables such as dimension tables, aim for a large number of tablets, and ensure each tablet is at least 1 GB.

> **Note:** In general, be mindful of the number of tablets and limit the parallelism of reads in the current implementation. Increasing the number of tablets significantly beyond the number of cores is likely to have diminishing returns.

Limitations

Current versions of Kudu come with a number of usage limitations. Cloudera recommends that you review the usage limitations of Kudu with respect to schema design, scaling, server management, cluster management, replication and backup, security, and integration with Spark and Impala.

Currently, Kudu does not support multi-row transactions. Operations that affect multiple rows do not roll back if the operation fails half way through. This should be mitigated by exploiting the primary key uniqueness constraint to make operations idempotent.

Related Information

Kudu introduction

Apache Kudu usage limitations

Security integration

System administrators who want to secure a cluster using data encryption, user authentication, and authorization techniques must review security-related information in Cloudera documentation.

Cloudera documentation provides conceptual overviews and how-to information about setting up various Hadoop components for optimal security, including how to set up a gateway to restrict access. It is assumed that you have basic knowledge of Linux and systems administration practices.

Related Information

CDP Security Overview

CDP Security How-tos

CDP Security Reference Architecture

FAQs

Review the information about some of the common questions related to having multiple data centers, operating systems, storage options, and RDBMS.

Operating systems

Review the list of supported operating systems for Cloudera Runtime and Cloudera Manager.

Storage options and configuration

The HDFS data directories should use local storage, which provides all the benefits of keeping compute resources close to the storage and not reading remotely over the network.

The root device size for CDP Private Cloud Base clusters should be at least 500 GB to allow parcels and logs to be stored.

To guard against data center failures, you can set up a scheduled distcp operation to persist data to a supported cloud object storage platform. For examples, see Using DistCp to copy files. You can also leverage Cloudera Manager's Replication Manager feature to backup data to another running cluster.

Relational databases

CDP Private Cloud Base deployments require relational databases for Cloudera Manager, Hive Metastore, Hue, Ranger, Atlas, Oozie, and so on. The database credentials are required during CDP Private Cloud Base installation. See Required Databases for more information.

> **Note:** The embedded PostgreSQL database is not supported for use in production environments.

References and acknowledgements

The CDP Private Cloud Base Reference Architecture document includes several links and references to understand cluster configuration and deployment best practices.

References

Following is a short list of key references:

- Product documentation
    - Hardware requirements
    - CDP Private Cloud Base Requirements and Supported Versions
- Cloudera Services and Support

Acknowledgements

Thanks to Travis Campbell, Kiran Anand, Jay Desai, Suraj Karuvel, J P Player, Ali Bajwa, David Fowler, and Andreas Nemeth for their help with review, feedback, and creation of this document. And thanks to members of the Cloudera Content Development team for their editorial support.