Oracle® Big Data Appliance

Software User's Guide

Release 2 (2.1)

E36963-03

April 2013

Provides an introduction to Oracle Big Data Appliance end-user software and to the administrative tools and procedures.

Oracle Big Data Appliance Software User's Guide, Release 2 (2.1)

E36963-03

Contents

Preface v

Audience v
Documentation Accessibility v
Related Documents v
Conventions vi

1 Introducing Oracle Big Data Appliance

What Is Big Data? 1-1
High Variety 1-1
High Complexity 1-2
High Volume 1-2
High Velocity 1-2
The Oracle Big Data Solution 1-2
Software for Big Data 1-3
Software Component Overview 1-4
Acquiring Data for Analysis 1-4
Hadoop Distributed File System 1-5
Oracle NoSQL Database 1-5
Hive 1-6
Organizing Big Data 1-6
MapReduce 1-6
Oracle R Support for Big Data 1-7
Oracle Big Data Connectors 1-8
Analyzing and Visualizing Big Data 1-9

2 Administering Oracle Big Data Appliance

Monitoring a Cluster Using Oracle Enterprise Manager 2-1
Managing CDH Operations Using Cloudera Manager 2-2
Monitoring the Status of Oracle Big Data Appliance 2-3
Performing Administrative Tasks 2-4
Managing Services With Cloudera Manager 2-4
Using Hadoop Monitoring Utilities 2-5
Monitoring the JobTracker 2-5
Monitoring the TaskTracker 2-5
Using Hue to Interact With Hadoop 2-6

About the Oracle Big Data Appliance Software 2-7
Software Components 2-7
Logical Disk Layout 2-8
About the Software Services 2-9
Monitoring the CDH Services 2-9
Where Do the Services Run? 2-9
Automatic Failover of the NameNode 2-10
Unconfigured Software 2-11
Configuring HBase 2-11
Effects of Hardware on Software Availability 2-12
Critical and Noncritical Nodes 2-12
First NameNode 2-13
Second NameNode 2-13
JobTracker Node 2-13
Noncritical Nodes 2-14
Collecting Diagnostic Information for Oracle Customer Support 2-14
Security on Oracle Big Data Appliance 2-15
About Predefined Users and Groups 2-16
Port Numbers Used on Oracle Big Data Appliance 2-16
About CDH Security Using Kerberos 2-17
About Puppet Security 2-18

3 Supporting User Access to Oracle Big Data Appliance

Providing Remote Client Access to CDH 3-1
Prerequisites 3-1
Installing CDH on Oracle Exadata Database Machine 3-2
Installing a CDH Client on Any Supported Operating System 3-3
Configuring CDH 3-3
Managing User Accounts 3-5
Providing User Login Privileges 3-5
Creating New HDFS Users 3-6
Recovering Deleted Files 3-7
Restoring Files from the Trash 3-7
Changing the Trash Interval 3-8
Disabling the Trash Facility 3-8

4 Configuring Oracle Exadata Database Machine for Use with Oracle Big Data Appliance

About Optimizing Communications 4-1
Prerequisites 4-1
Enabling SDP on Exadata Database Nodes 4-1
Configuring a JDBC Client for SDP 4-2
Creating an SDP Listener on the InfiniBand Network 4-3

Glossary

Index

Preface

The *Oracle Big Data Appliance Software User's Guide* describes how to manage and use the installed software.

Audience

This guide is intended for users of Oracle Big Data Appliance including:

- Application developers
- Data analysts
- Data scientists
- Database administrators
- System administrators

The *Oracle Big Data Appliance Software User's Guide* introduces the terminology and concepts necessary to discuss Oracle Big Data Appliance. However, you must acquire the necessary information about administering Hadoop clusters and writing MapReduce programs from other sources.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc`.

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info` or visit `http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs` if you are hearing impaired.

Related Documents

For more information, see the following documents:

- *Oracle Big Data Appliance Owner's Guide*
- *Oracle Big Data Connectors User's Guide*
- *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Oracle Big Data Appliance*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introducing Oracle Big Data Appliance

This chapter presents an overview of Oracle Big Data Appliance and describes the software installed on the system. This chapter contains the following sections:

- What Is Big Data?
- The Oracle Big Data Solution
- Software for Big Data
- Acquiring Data for Analysis
- Organizing Big Data
- Analyzing and Visualizing Big Data

What Is Big Data?

Using transactional data as the source of business intelligence has been commonplace for many years. As digital technology and the World Wide Web spread into every aspect of modern life, other sources of data can make important contributions to business decision making. Many businesses are looking to these new data sources. They are finding opportunities in analyzing vast amounts of data that until recently was discarded.

Big data is characterized by:

- High Variety
- High Complexity
- High Volume
- High Velocity

These characteristics pinpoint the challenges in deriving value from big data, and the differences between big data and traditional data sources that primarily provide highly structured, transactional data.

High Variety

Big data is derived from a variety of sources, such as:

- Equipment sensors: Medical, manufacturing, transportation, and other machine sensor transmissions
- Machines: Call detail records, web logs, smart meter readings, Global Positioning System (GPS) transmissions, and trading systems records

- Social media: Data streams from social media sites such as Facebook and blogging sites such as Twitter

Analysts can mine this data repeatedly as they devise new ways of extracting meaningful insights. What seems irrelevant today might prove to be highly pertinent to your business tomorrow.

Challenge: Delivering flexible systems to handle this high variety

High Complexity

As the variety of data types increases, the complexity of the system increases. The complexity of data types also increases in big data because of its low structure.

Challenge: Finding solutions that apply across a broad range of data types.

High Volume

Social media can generate terabytes of daily data. Equipment sensors and other machines can generate that much data in less than an hour.

Even traditional data sources for data warehouses, such as customer profiles from customer relationship management (CRM) systems, transactional enterprise resource planning (ERP) data, store transactions, and general ledger data, have increased tenfold in volume over the past decade.

Challenge: Providing scalability and ease in growing the system

High Velocity

Huge numbers of sensors, web logs, and other machine sources generate data continuously and at a much higher speed than traditional sources, such as individuals entering orders into a transactional database.

Challenge: Handling the data at high speed without stressing the structured systems

The Oracle Big Data Solution

Oracle Big Data Appliance is an engineered system comprising both hardware and software components. The hardware is optimized to run the enhanced big data software components.

Oracle Big Data Appliance delivers:

- A complete and optimized solution for big data
- Single-vendor support for both hardware and software
- An easy-to-deploy solution
- Tight integration with Oracle Database and Oracle Exadata Database Machine

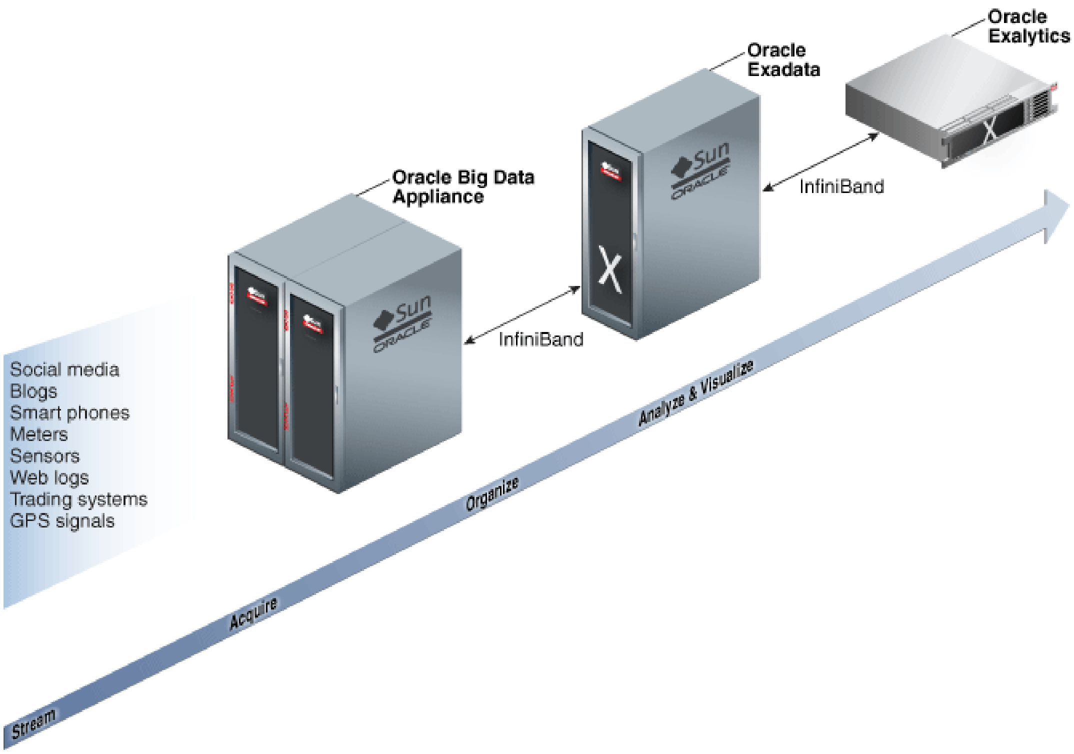
Oracle provides a big data platform that captures, organizes, and supports deep analytics on extremely large, complex data streams flowing into your enterprise from a large number of data sources. You can choose the best storage and processing location for your data depending on its structure, workload characteristics, and end-user requirements.

Oracle Database enables all data to be accessed and analyzed by a large user community using identical methods. By adding Oracle Big Data Appliance in front of Oracle Database, you can bring new sources of information to an existing data warehouse. Oracle Big Data Appliance is the platform for acquiring and organizing big data so that the relevant portions with true business value can be analyzed in Oracle Database.

For maximum speed and efficiency, Oracle Big Data Appliance can be connected to Oracle Exadata Database Machine running Oracle Database. Oracle Exadata Database Machine provides outstanding performance in hosting data warehouses and transaction processing databases. Moreover, Oracle Exadata Database Machine can be connected to Oracle Exalytics In-Memory Machine for the best performance of business intelligence and planning applications. The InfiniBand connections between these engineered systems provide high parallelism, which enables high-speed data transfer for batch or query workloads.

Figure 1–1 shows the relationships among these engineered systems.

Figure 1–1 Oracle Engineered Systems for Big Data



Software for Big Data

The **Oracle Linux** operating system and Cloudera's Distribution including Apache Hadoop (CDH) underlie all other software components installed on Oracle Big Data Appliance. **CDH** is an integrated stack of components that have been tested and packaged to work together.

CDH has a batch processing infrastructure that can store files and distribute work across a set of computers. Data is processed on the same computer where it is stored. In a single Oracle Big Data Appliance rack, CDH distributes the files and workload across 18 servers, which compose a **cluster**. Each server is a node in the cluster.

The software framework consists of these primary components:

- File system: The **Hadoop Distributed File System (HDFS)** is a highly scalable file system that stores large files across multiple servers. It achieves reliability by replicating data across multiple servers without RAID technology. It runs on top of the Linux file system on Oracle Big Data Appliance.

- MapReduce engine: The MapReduce engine provides a platform for the massively parallel execution of algorithms written in Java.
- Administrative framework: Cloudera Manager is a comprehensive administrative tool for CDH.

CDH is written in Java, and Java is the language for applications development. However, several CDH utilities and other software available on Oracle Big Data Appliance provide graphical, web-based, and other language interfaces for ease of use.

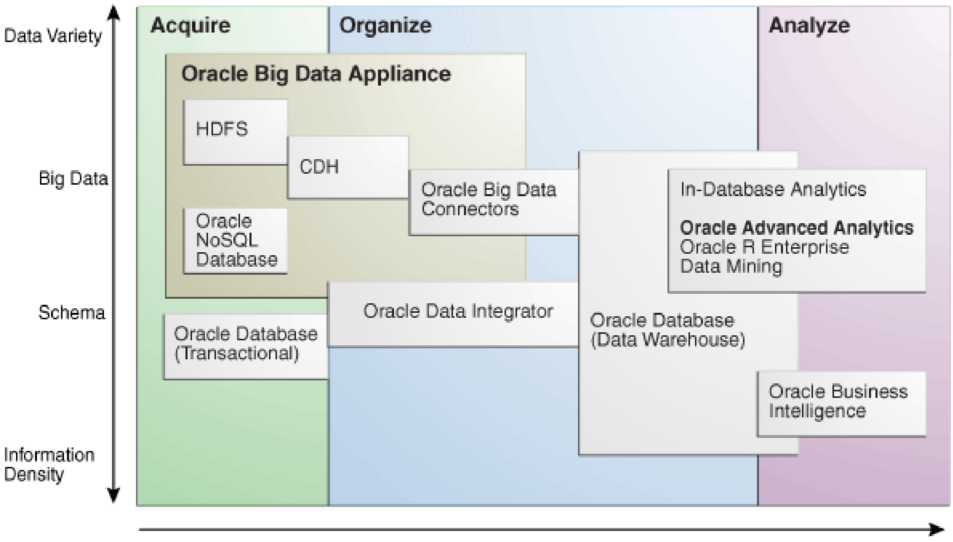
Software Component Overview

The major software components perform three basic tasks:

- Acquire
- Organize
- Analyze and visualize

The best tool for each task depends on the density of the information and the degree of structure. Figure 1–2 shows the relationships among the tools and identifies the tasks that they perform.

Figure 1–2 Oracle Big Data Appliance Software Overview



Acquiring Data for Analysis

Oracle Big Data Appliance provides these facilities for capturing and storing big data:

- Hadoop Distributed File System (HDFS)
- Oracle NoSQL Database
- **Hive**

Databases used for online transaction processing (OLTP) are the traditional data sources for data warehouses. The Oracle solution enables you to analyze traditional data stores with big data in the same Oracle data warehouse. Relational data continues to be an important source of business intelligence, although it runs on separate hardware from Oracle Big Data Appliance.

Hadoop Distributed File System

Cloudera's Distribution including Apache Hadoop (CDH) on Oracle Big Data Appliance uses the Hadoop Distributed File System (HDFS). HDFS stores extremely large files containing record-oriented data. On Oracle Big Data Appliance, HDFS splits large data files into chunks of 256 megabytes (MB), and replicates each chunk across three different nodes in the cluster. The size of the chunks and the number of replications are configurable.

Chunking enables HDFS to store files that are larger than the physical storage of one server. It also allows the data to be processed in parallel across multiple computers with multiple processors, all working on data that is stored locally. Replication ensures the high availability of the data: if a server fails, the other servers automatically take over its work load.

HDFS is typically used to store all types of big data.

See Also:

- For conceptual information about Hadoop technologies, refer to this third-party publication:

  Hadoop: The Definitive Guide, Third Edition by Tom White (O'Reilly Media Inc., 2012., ISBN: 978-1449311520).

- For documentation about Cloudera's Distribution including Apache Hadoop, see the Cloudera library at

  `http://oracle.cloudera.com/`

Oracle NoSQL Database

Oracle NoSQL Database is a distributed key-value database built on the proven storage technology of Berkeley DB Java Edition. Whereas HDFS stores unstructured data in very large files, Oracle NoSQL Database indexes the data and supports transactions. But unlike Oracle Database, which stores highly structured data, Oracle NoSQL Database has relaxed consistency rules, no schema structure, and only modest support for joins, particularly across storage nodes.

NoSQL databases, or "Not Only SQL" databases, have developed over the past decade specifically for storing big data. However, they vary widely in implementation. Oracle NoSQL Database has these characteristics:

- Uses a system-defined, consistent hash index for data distribution
- Supports high availability through replication
- Provides single-record, single-operation transactions with relaxed consistency guarantees
- Provides a Java API

Oracle NoSQL Database is designed to provide highly reliable, scalable, predictable, and available data storage. The key-value pairs are stored in shards or partitions (that is, subsets of data) based on a primary key. Data on each shard is replicated across multiple storage nodes to ensure high availability. Oracle NoSQL Database supports fast querying of the data, typically by key lookup.

An intelligent driver links the NoSQL database with client applications and provides access to the requested key-value on the storage node with the lowest latency.

Oracle NoSQL Database includes hashing and balancing algorithms to ensure proper data distribution and optimal load balancing, replication management components to handle storage node failure and recovery, and an easy-to-use administrative interface to monitor the state of the database.

Oracle NoSQL Database is typically used to store customer profiles and similar data for identifying and analyzing big data. For example, you might log in to a website and see advertisements based on your stored customer profile (a record in Oracle NoSQL Database) and your recent activity on the site (web logs currently streaming into HDFS).

Oracle NoSQL Database is an optional component of Oracle Big Data Appliance. It is always installed, but might not be activated during installation of the software.

See Also:

- *Oracle NoSQL Database Getting Started Guide* at

  `http://docs.oracle.com/cd/NOSQL/html/index.html`
- *Oracle Big Data Appliance Licensing Information*

Hive

Hive is an open-source data warehouse that supports data summarization, ad hoc querying, and data analysis of data stored in HDFS. It uses a SQL-like language called **HiveQL**. An interpreter generates MapReduce code from the HiveQL queries. By storing data in Hive, you can avoid writing MapReduce programs in Java.

Hive is a component of CDH and is always installed on Oracle Big Data Appliance. Most of the Oracle Big Data Connectors can access Hive tables.

Organizing Big Data

Oracle Big Data Appliance provides several ways of organizing, transforming, and reducing big data for analysis:

- MapReduce
- Oracle R Support for Big Data
- Oracle Big Data Connectors

MapReduce

The MapReduce engine provides a platform for the massively parallel execution of algorithms written in Java. MapReduce uses a parallel programming model for processing data on a distributed system. It can process vast amounts of data quickly and can scale linearly. It is particularly effective as a mechanism for batch processing of unstructured and semistructured data. MapReduce abstracts lower-level operations into computations over a set of keys and values.

Although big data is often described as unstructured, incoming data always has some structure. However, it does not have a fixed, predefined structure when written to HDFS. Instead, MapReduce creates the desired structure as it reads the data for a particular job. The same data can have many different structures imposed by different MapReduce jobs.

A simplified description of a MapReduce job is the successive alternation of two phases: the Map phase and the Reduce phase. Each Map phase applies a transform function over each record in the input data to produce a set of records expressed as key-value pairs. The output from the Map phase is input to the Reduce phase. In the Reduce phase, the Map output records are sorted into key-value sets, so that all records in a set have the same key value. A reducer function is applied to all the records in a set, and a set of output records is produced as key-value pairs. The Map phase is logically run in parallel over each record, whereas the Reduce phase is run in parallel over all key values.

> **Note:** Oracle Big Data Appliance does not support the Yet Another Resource Negotiator (YARN) implementation of MapReduce.

Oracle R Support for Big Data

R is an open-source language and environment for statistical analysis and graphing It provides linear and nonlinear modeling, standard statistical methods, time-series analysis, classification, clustering, and graphical data displays. Thousands of open-source packages are available in the Comprehensive R Archive Network (CRAN) for a spectrum of applications, such as bioinformatics, spatial statistics, and financial and marketing analysis. The popularity of R has increased as its functionality matured to rival that of costly proprietary statistical packages.

Analysts typically use R on a PC, which limits the amount of data and the processing power available for analysis. Oracle eliminates this restriction by extending the R platform to directly leverage Oracle Big Data Appliance. Analysts continue to work on their PCs using the familiar R user interface while manipulating huge amounts of data stored in HDFS using massively parallel processing.

The standard R distribution is installed on all nodes of Oracle Big Data Appliance, enabling R programs to run as MapReduce jobs on vast amounts of data. Users can transfer existing R scripts and packages from their PCs to use on Oracle Big Data Appliance.

Oracle R Connector for Hadoop provides R users with high-performance, native access to HDFS and the **MapReduce** programming framework. Oracle R Connector for Hadoop is included in the Oracle Big Data Connectors. See "Oracle R Connector for Hadoop" on page 1-8.

Oracle R Enterprise is a separate package that provides real-time access to Oracle Database. It enables you to store the results of your analysis of big data in an Oracle database, where it can be analyzed further.

These two Oracle R packages make Oracle Database and the Hadoop computational infrastructure available to statistical users without requiring them to learn the native programming languages of either one.

See Also:

- For information about R, go to

  `http://www.r-project.org/`

- For information about Oracle R Enterprise, go to

  `http://docs.oracle.com/cd/E27988_01/welcome.html`

Oracle Big Data Connectors

Oracle Big Data Connectors facilitate data access between data stored in CDH and Oracle Database. The connectors are licensed separately from Oracle Big Data Appliance and include:

- Oracle SQL Connector for Hadoop Distributed File System
- Oracle Loader for Hadoop
- Oracle R Connector for Hadoop
- Oracle Data Integrator Application Adapter for Hadoop

See Also: *Oracle Big Data Connectors User's Guide*

Oracle SQL Connector for Hadoop Distributed File System

Oracle SQL Connector for Hadoop Distributed File System (Oracle SQL Connector for HDFS) provides read access to HDFS from an Oracle database using **external tables**.

An external table is an Oracle Database object that identifies the location of data outside of the database. Oracle Database accesses the data by using the metadata provided when the external table was created. By querying the external tables, users can access data stored in HDFS as if that data were stored in tables in the database. External tables are often used to stage data to be transformed during a database load.

You can use Oracle SQL Connector for HDFS to:

- Access data stored in HDFS files
- Access Hive tables.
- Access comma-separated value (CSV) files generated by Oracle Loader for Hadoop
- Load data extracted and transformed by Oracle Data Integrator

Oracle Loader for Hadoop

Oracle Loader for Hadoop is an efficient and high-performance loader for fast movement of data from CDH into a table in an Oracle database. Oracle Loader for Hadoop partitions the data and transforms it into a database-ready format on CDH. It optionally sorts records by primary key before loading the data or creating output files.

You can use Oracle Loader for Hadoop as either a Java program or a command-line utility. The load runs as a MapReduce job on the CDH cluster.

Oracle Loader for Hadoop also reads from and writes to Oracle Data Pump files.

Oracle R Connector for Hadoop

Oracle R Connector for Hadoop is a collection of R packages that provide:

- Interfaces to work with HIVE tables, Apache Hadoop compute infrastructure, local R environment and database tables
- Predictive analytic techniques written in R or Java as Hadoop MapReduce jobs that can be applied to data in HDFS files

Using simple R functions, you can copy data between R memory, the local file system, HDFS, and Hive. You can schedule R programs to execute as Hadoop MapReduce jobs and return the results to any of those locations.

Oracle Data Integrator Application Adapter for Hadoop

Oracle Data Integrator (ODI) extracts, transforms, and loads data into Oracle Database from a wide range of sources.

In ODI, a knowledge module (KM) is a code template dedicated to a specific task in the data integration process. You use Oracle Data Integrator Studio to load, select, and configure the KMs for your particular application. More than 150 KMs are available to help you acquire data from a wide range of third-party databases and other data repositories. You only need to load a few KMs for any particular job.

Oracle Data Integrator Application Adapter for Hadoop contains the KMs specifically for use with big data.

Analyzing and Visualizing Big Data

After big data is transformed and loaded in Oracle Database, you can use the full spectrum of Oracle business intelligence solutions and decision support products to further analyze and visualize all your data.

See Also:

- Oracle Business Intelligence website at

  `http://www.oracle.com/us/solutions/ent-performance-bi/business-intelligence/index.html`

- Data Warehousing and Business Intelligence in the Oracle Database Documentation Library at

  `http://www.oracle.com/pls/db112/portal.portal_db?selected=6&frame=`

2
Administering Oracle Big Data Appliance

This chapter provides information about the software and services installed on Oracle Big Data Appliance. It contains these sections:

- Monitoring a Cluster Using Oracle Enterprise Manager
- Managing CDH Operations Using Cloudera Manager
- Using Hadoop Monitoring Utilities
- Using Hue to Interact With Hadoop
- About the Oracle Big Data Appliance Software
- About the Software Services
- Configuring HBase
- Effects of Hardware on Software Availability
- Collecting Diagnostic Information for Oracle Customer Support
- Security on Oracle Big Data Appliance

Monitoring a Cluster Using Oracle Enterprise Manager

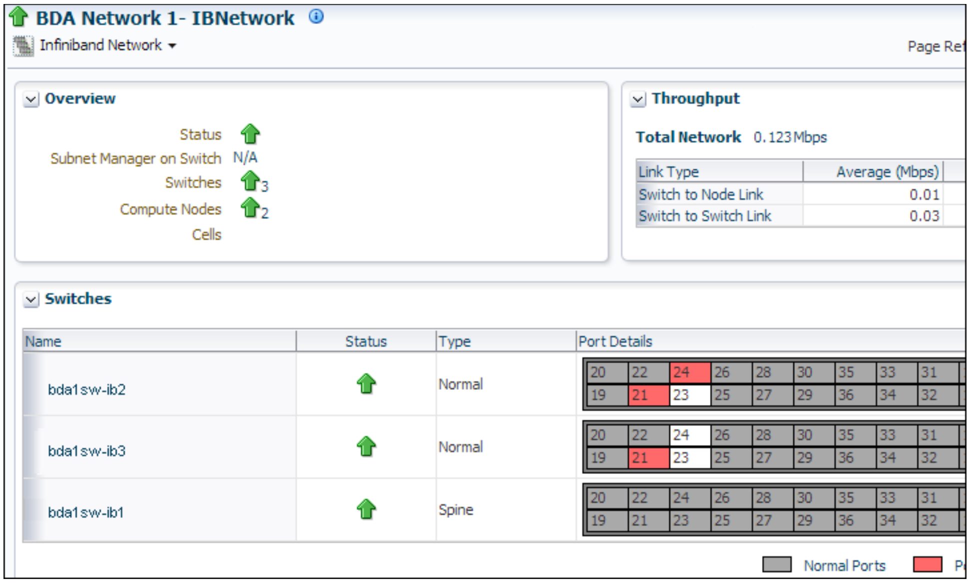
An Oracle Enterprise Manager plug-in enables you to use the same system monitoring tool for Oracle Big Data Appliance as you use for Oracle Exadata Database Machine or any other Oracle Database installation. With the plug-in, you can view the status of the installed software components in tabular or graphic presentations, and start and stop these software services. You can also monitor the health of the network and the rack components.

After selecting a target cluster, you can drill down into these primary areas:

- **InfiniBand network**: Network topology and status for InfiniBand switches and ports. See Figure 2–1.
- **Hadoop cluster**: Software services for HDFS, MapReduce, and ZooKeeper.
- **Oracle Big Data Appliance rack**: Hardware status including server hosts, Oracle Integrated Lights Out Manager (Oracle ILOM) servers, power distribution units (PDUs), and the Ethernet switch.

Figure 2–1 shows some of the information provided about the InfiniBand switches.

Figure 2–1 InfiniBand Home in Oracle Enterprise Manager



BDA Network 1- IBNetwork

Infiniband Network

Overview

Status
Subnet Manager on Switch N/A
Switches 3
Compute Nodes 2
Cells

Throughput

Total Network 0.123 Mbps

Link Type	Average (Mbps)
Switch to Node Link	0.01
Switch to Switch Link	0.03

Switches

Name	Status	Type	Port Details
bda1sw-ib2	⬆	Normal	20 22 24 26 28 30 35 33 31 19 21 23 25 27 29 36 34 32
bda1sw-ib3	⬆	Normal	20 22 24 26 28 30 35 33 31 19 21 23 25 27 29 36 34 32
bda1sw-ib1	⬆	Spine	20 22 24 26 28 30 35 33 31 19 21 23 25 27 29 36 34 32

To monitor Oracle Big Data Appliance using Oracle Enterprise Manager:

1. Download and install the plug-in. See *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Oracle Big Data Appliance*.
2. Log in to Oracle Enterprise Manager as a privileged user.
3. From the Targets menu, choose **Big Data Appliance** to view the Big Data page. You can see the overall status of the targets already discovered by Oracle Enterprise Manager.
4. Select a target cluster to view its detail pages.
5. Expand the target navigation tree to display the components. Information is available at all levels.
6. Select a component in the tree to display its home page.
7. To change the display, choose an item from the drop-down menu at the top left of the main display area.

> **See Also:** *Oracle Enterprise Manager System Monitoring Plug-in Installation Guide for Oracle Big Data Appliance* for installation instructions and use cases.

Managing CDH Operations Using Cloudera Manager

Cloudera Manager is installed on Oracle Big Data Appliance to help you with Cloudera's Distribution including Apache Hadoop (CDH) operations. Cloudera Manager provides a single administrative interface to all Oracle Big Data Appliance servers configured as part of the Hadoop cluster.

Cloudera Manager simplifies the performance of these administrative tasks:

- Monitor jobs and services
- Start and stop services
- Manage security and Kerberos credentials
- Monitor user activity

- Monitor the health of the system
- Monitor performance metrics
- Track hardware use (disk, CPU, and RAM)

Cloudera Manager runs on the JobTracker node (node03) of the primary rack and is available on port 7180.

To use Cloudera Manager:

1. Open a browser and enter a URL like the following:

   ```
   http://bda1node03.example.com:7180
   ```

   In this example, `bda1` is the name of the appliance, `node03` is the name of the server, `example.com` is the domain, and `7180` is the default port number for Cloudera Manager.

2. Log in with a user name and password for Cloudera Manager. Only a user with administrative privileges can change the settings. Other Cloudera Manager users can view the status of Oracle Big Data Appliance.

   > **See Also:** *Cloudera Manager User Guide* at
   >
   > http://ccp.cloudera.com/display/ENT/Cloudera+Manager+User+Guide
   >
   > or click **Help** on the Cloudera Manager Help menu

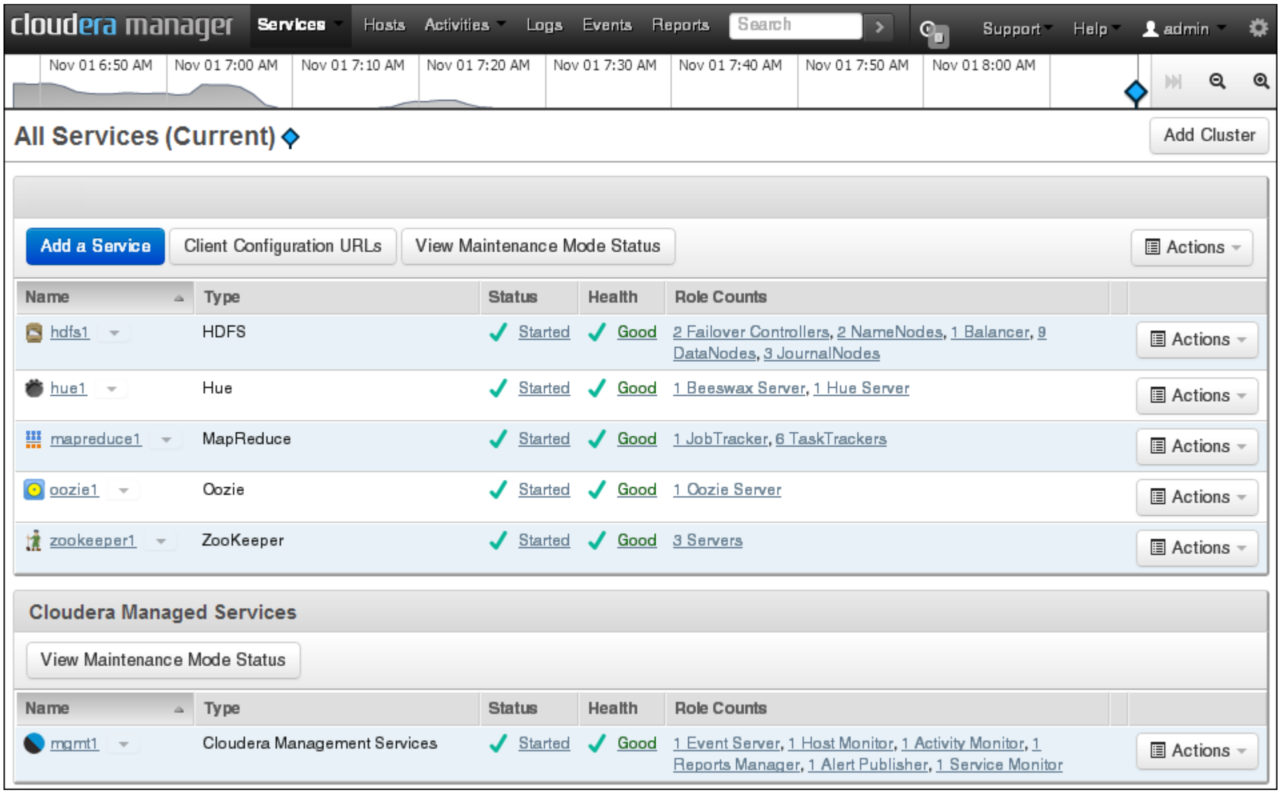
Monitoring the Status of Oracle Big Data Appliance

In Cloudera Manager, you can choose any of the following pages from the menu bar across the top of the display:

- **Services**: Monitors the status and health of services running on Oracle Big Data Appliance. Click the name of a service to drill down to additional information.
- **Hosts**: Monitors the health, disk usage, load, physical memory, swap space, and other statistics for all servers.
- **Activities**: Monitors all MapReduce jobs running in the selected time period.
- **Logs**: Collects historical information about the systems and services. You can search for a particular phrase for a selected server, service, and time period. You can also select the minimum severity level of the logged messages included in the search: TRACE, DEBUG, INFO, WARN, ERROR, or FATAL.
- **Events**: Records a change in state and other noteworthy occurrences. You can search for one or more keywords for a selected server, service, and time period. You can also select the event type: Audit Event, Activity Event, Health Check, or Log Message.
- **Reports**: Generates reports on demand for disk and MapReduce use.

Figure 2–2 shows the opening display of Cloudera Manager, which is the Services page.

Figure 2–2 Cloudera Manager Services Page



All Services (Current)

Name	Type	Status	Health	Role Counts	
hdfs1	HDFS	✓ Started	✓ Good	2 Failover Controllers, 2 NameNodes, 1 Balancer, 9 DataNodes, 3 JournalNodes	Actions
hue1	Hue	✓ Started	✓ Good	1 Beeswax Server, 1 Hue Server	Actions
mapreduce1	MapReduce	✓ Started	✓ Good	1 JobTracker, 6 TaskTrackers	Actions
oozie1	Oozie	✓ Started	✓ Good	1 Oozie Server	Actions
zookeeper1	ZooKeeper	✓ Started	✓ Good	3 Servers	Actions

Cloudera Managed Services

View Maintenance Mode Status

Name	Type	Status	Health	Role Counts	
mgmt1	Cloudera Management Services	✓ Started	✓ Good	1 Event Server, 1 Host Monitor, 1 Activity Monitor, 1 Reports Manager, 1 Alert Publisher, 1 Service Monitor	Actions

Performing Administrative Tasks

As a Cloudera Manager administrator, you can change various properties for monitoring the health and use of Oracle Big Data Appliance, add users, and set up Kerberos security.

To access Cloudera Manager Administration:

1. Log in to Cloudera Manager with administrative privileges.
2. Click **Welcome admin** at the top right of the page.

Managing Services With Cloudera Manager

Cloudera Manager provides the interface for managing these services:

- HDFS
- Hue
- MapReduce
- Oozie
- ZooKeeper

You can use Cloudera Manager to change the configuration of these services, stop, and restart them.

> **Note:** Manual edits to Linux service scripts or Hadoop configuration files do not affect these services. You must manage and configure them using Cloudera Manager.

Using Hadoop Monitoring Utilities

Users can monitor MapReduce jobs without providing a Cloudera Manager user name and password.

Monitoring the JobTracker

Hadoop Map/Reduce Administration monitors the **JobTracker**, which runs on port 50030 of the JobTracker node (node03) on Oracle Big Data Appliance.

To monitor the JobTracker:

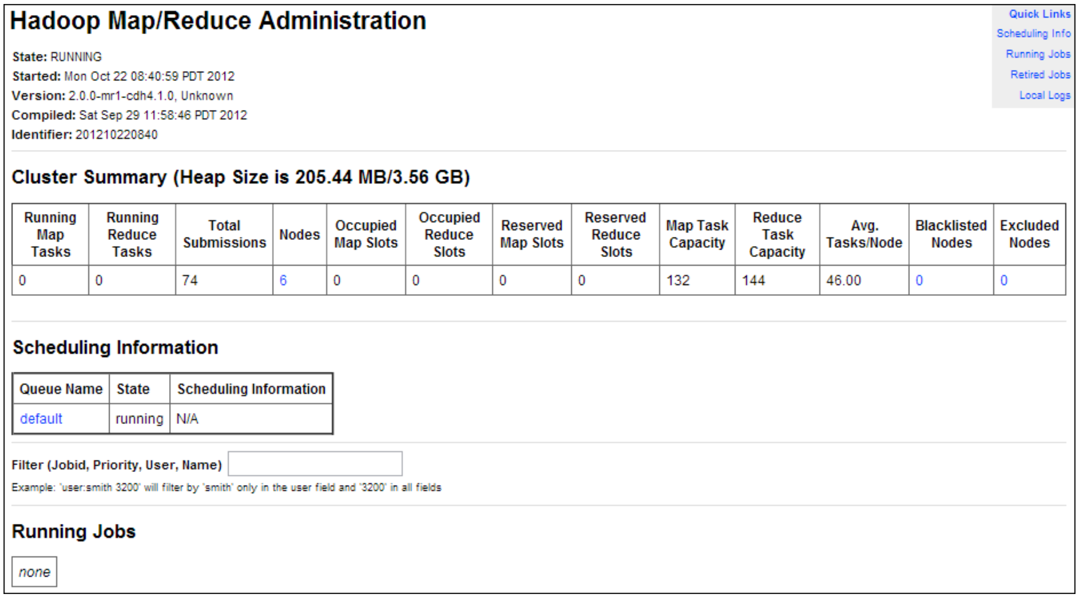
- Open a browser and enter a URL like the following:

  ```
  http://bda1node03.example.com:50030
  ```

  In this example, `bda1` is the name of the appliance, `node03` is the name of the server, and `50030` is the default port number for Hadoop Map/Reduce Administration.

Figure 2–3 shows part of a Hadoop Map/Reduce Administration display.

Figure 2–3 Hadoop Map/Reduce Administration



Hadoop Map/Reduce Administration

State: RUNNING
Started: Mon Oct 22 08:40:59 PDT 2012
Version: 2.0.0-mr1-cdh4.1.0, Unknown
Compiled: Sat Sep 29 11:58:46 PDT 2012
Identifier: 201210220840

Cluster Summary (Heap Size is 205.44 MB/3.56 GB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Excluded Nodes
0	0	74	6	0	0	0	0	132	144	46.00	0	0

Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)

Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Running Jobs

none

Monitoring the TaskTracker

The Task Tracker Status interface monitors the **TaskTracker** on a single node. It is available on port 50060 of all noncritical nodes (node04 to node18) in Oracle Big Data Appliance. On six-node clusters, the TaskTracker also runs on node01 and node02.

To monitor a TaskTracker:

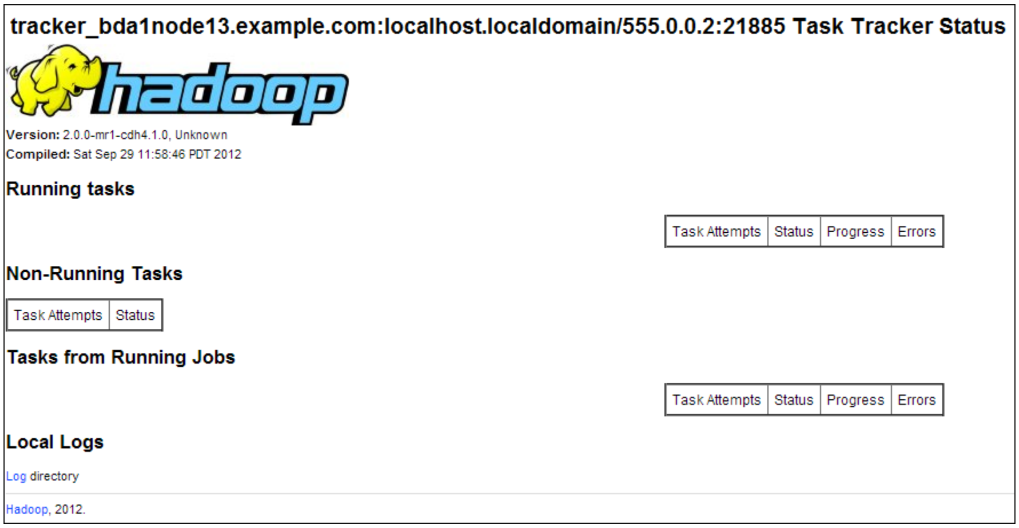
- Open a browser and enter the URL for a particular node like the following:

```
http://bda1node13.example.com:50060
```

In this example, `bda1` is the name of the rack, `node13` is the name of the server, and `50060` is the default port number for the Task Tracker Status interface.

Figure 2–4 shows the Task Tracker Status interface.

Figure 2–4 Task Tracker Status Interface



tracker_bda1node13.example.com:localhost.localdomain/555.0.0.2:21885 Task Tracker Status

Version: 2.0.0-mr1-cdh4.1.0, Unknown
Compiled: Sat Sep 29 11:58:46 PDT 2012

Running tasks

Task Attempts	Status	Progress	Errors

Non-Running Tasks

Task Attempts	Status

Tasks from Running Jobs

Task Attempts	Status	Progress	Errors

Local Logs

Log directory

Hadoop, 2012.

Using Hue to Interact With Hadoop

Hue runs in a browser and provides an easy-to-use interface to several applications to support interaction with Hadoop and HDFS. You can use Hue to perform any of the following tasks:

- Query Hive data stores
- Create, load, and delete Hive tables
- Work with HDFS files and directories
- Create, submit, and monitor MapReduce jobs
- Monitor MapReduce jobs
- Create, edit, and submit workflows using the Oozie dashboard
- Manage users and groups

Hue runs on port 8888 of the JobTracker node (node03).

To use Hue:

1. Open Hue in a browser using an address like the one in this example:

   ```
   http://bda1node03.example.com:8888
   ```

   In this example, `bda1` is the cluster name, `node03` is the server name, and `example.com` is the domain.

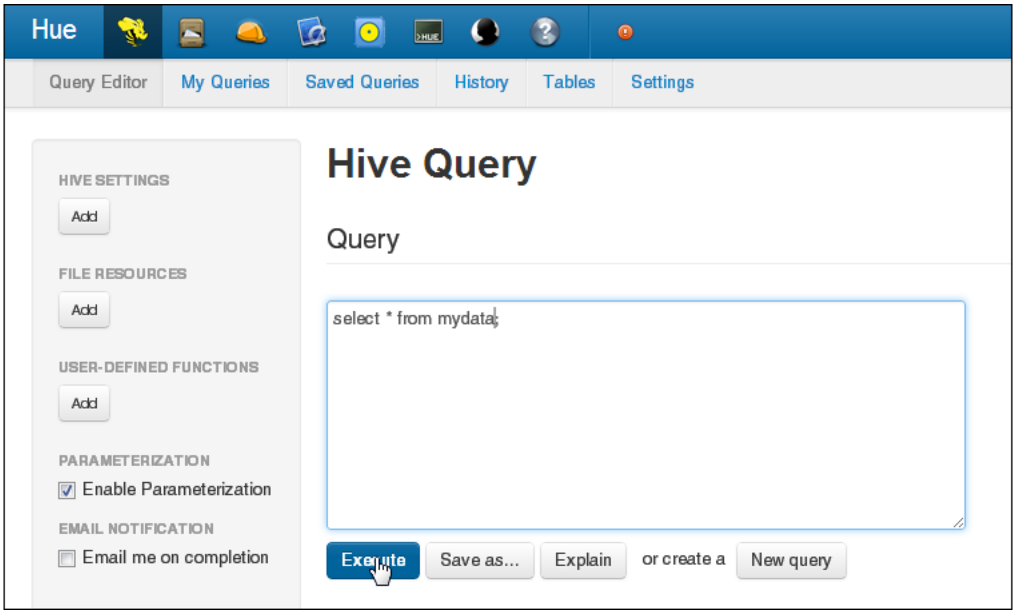
2. Log in with your Hue credentials.

   Oracle Big Data Appliance is not configured initially with any Hue user accounts. The first user who connects to Hue can log in with any user name and password, and automatically becomes an administrator. This user can create other user and administrator accounts.

3. Use the icons across the top to open a utility.

Figure 2–5 shows the Beeswax Query Editor for entering Hive queries.

Figure 2–5 Beeswax Query Editor



☑ Enable Parameterization

□ Email me on completion

```
select * from mydata;
```

See Also: *Hue Installation Guide* for information about using Hue, which is already installed and configured on Oracle Big Data Appliance, at

`http://cloudera.github.com/hue/docs-2.1.0/manual.html`

About the Oracle Big Data Appliance Software

The following sections identify the software installed on Oracle Big Data Appliance and where it runs in the rack. Some components operate with Oracle Database 11.2.0.2 and later releases.

Software Components

These software components are installed on all 18 servers in Oracle Big Data Appliance Rack. Oracle Linux, required drivers, firmware, and hardware verification utilities are factory installed. All other software is installed on site using the Mammoth Utility. The optional software components may not be configured in your installation.

Note: You do not need to install additional software on Oracle Big Data Appliance. Doing so may result in a loss of warranty and support. See the *Oracle Big Data Appliance Owner's Guide*.

Base image software:

- Oracle Linux 5.8

- Java **HotSpot** Virtual Machine 6 Update 37
- **Oracle R Distribution** 2.15.1
- **MySQL Server** 5.5.17 Advanced Edition
- Puppet, firmware, utilities

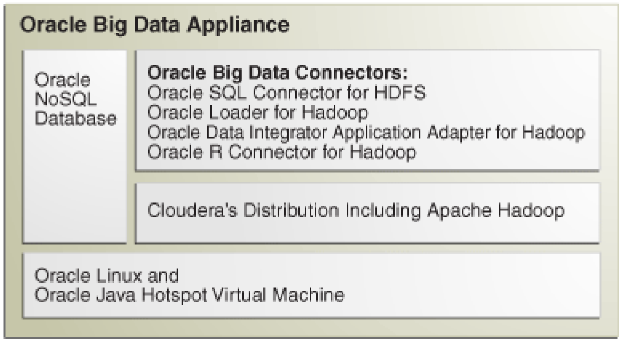
Mammoth installation:

- Cloudera's Distribution including Apache Hadoop Release 4 Update 1.2 (CDH)
- Cloudera Manager Enterprise 4.1.2
- **Oracle Database Instant Client** 11.2.0.3
- Oracle NoSQL Database Community Edition or Enterprise Edition 11*g* Release 2.0 (optional)
- Oracle Big Data Connectors 2.1 (optional):
  - Oracle SQL Connector for Hadoop Distributed File System (HDFS)
  - Oracle Loader for Hadoop
  - Oracle Data Integrator Agent 11.1.1.6.0
  - Oracle R Connector for Hadoop

> **See Also:** *Oracle Big Data Appliance Owner's Guide* for information about the Mammoth Utility

Figure 2–6 shows the relationships among the major components.

Figure 2–6 Major Software Components of Oracle Big Data Appliance


Oracle Big Data Appliance
Oracle NoSQL Database
Oracle Big Data Connectors:
Oracle SQL Connector for HDFS
Oracle Loader for Hadoop
Oracle Data Integrator Application Adapter for Hadoop
Oracle R Connector for Hadoop
Cloudera's Distribution Including Apache Hadoop
Oracle Linux and
Oracle Java Hotspot Virtual Machine

Logical Disk Layout

Each server has 12 disks. The critical operating system is stored on disks 1 and 2.

Table 2–1 describes how the disks are partitioned.

Table 2–1 Logical Disk Layout

Disk	Description
1 to 2	150 gigabytes (GB) physical and logical partition, mirrored to create two copies, with the Linux operating system, all installed software, NameNode data, and MySQL Database data. The NameNode and MySQL Database data are replicated on two servers for a total of four copies. 2.8 terabytes (TB) HDFS data partition

Table 2–1 (Cont.) Logical Disk Layout

Disk	Description
3 to 10	Single HDFS data partition
11 to 12	Single Oracle NoSQL Database partition, if activated during software installation; otherwise, a single HDFS data partition

About the Software Services

This section contains the following topics:

- Monitoring the CDH Services
- Where Do the Services Run?
- Automatic Failover of the NameNode
- Unconfigured Software

Monitoring the CDH Services

You can use Cloudera Manager to monitor the CDH services on Oracle Big Data Appliance.

To monitor the services:

1. In Cloudera Manager, click the **Services** tab at the top of the page to display the Services page.
2. Click the name of a service to see its detail pages. The service opens on the Status page.
3. Click the link to the page that you want to view: Status, Instances, Commands, Configuration, or Audits.

Where Do the Services Run?

All services are installed on all servers, but individual services run only on designated nodes in the Hadoop cluster.

Table 2–2 identifies the nodes where the services run on the primary rack. Services that run on all nodes run on all racks of a multirack installation.

Table 2–2 Software Service Locations

Service Type	Role	Node Name	Initial Node Position
Cloudera Management Services	Cloudera Manager agents	All nodes	Node01 to node18
Cloudera Management Services	Cloudera Manager server	JobTracker node	Node03
HDFS	**Balancer**	First NameNode	Node01
HDFS	**DataNode**	All nodes	Node01 to node18
HDFS	Failover controller	First NameNode and second NameNode	Node01 and node02
HDFS	First **NameNode**	First NameNode	Node01
HDFS	JournalNode	First NameNode, second NameNode, JobTracker node	Node01 to node03

Table 2–2 (Cont.) Software Service Locations

Service Type	Role	Node Name	Initial Node Position
HDFS	Second NameNode	Second NameNode	Node02
Hive	**Hive** server	JobTracker node	Node03
Hue	Beeswax server	JobTracker node	Node03
Hue	**Hue** server	JobTracker node	Node03
MapReduce	**JobTracker**	JobTracker node	Node03
MapReduce	**TaskTracker**	All noncritical nodes	Node04 to node18
MySQL	MySQL Backup Server[1]	Second NameNode	Node02
MySQL	MySQL Primary Server[1]	JobTracker node	Node03
NoSQL	Oracle NoSQL Database Administration[2]	Second NameNode	Node02
NoSQL	Oracle NoSQL Database Server processes[2]	All nodes	Node01 to node18
ODI	Oracle Data Integrator agent[2]	JobTracker node	Node03
Puppet	**Puppet** agents	All nodes	Node01 to node18
Puppet	Puppet master	First NameNode	Node01
ZooKeeper	**ZooKeeper** server	First NameNode, second NameNode, JobTracker node	Node01 to node03

[1] If the software was upgraded from version 1.0, then MySQL Backup remains on node02 and MySQL Primary Server remains on node03.

[2] Started only if requested in the *Oracle Big Data Appliance Configuration Worksheets*

Automatic Failover of the NameNode

The NameNode is the most critical process because it keeps track of the location of all data. Without a healthy NameNode, the entire cluster fails. Apache Hadoop v0.20.2 and earlier are vulnerable to failure because they have a single name node.

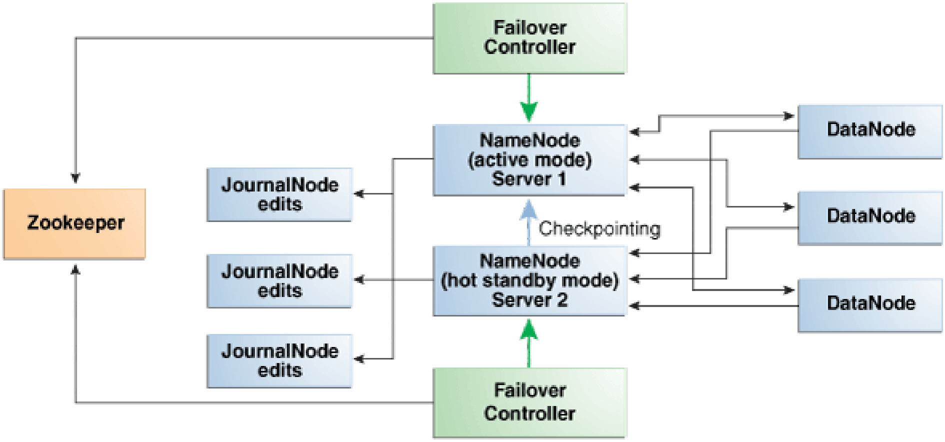
Cloudera's Distribution including Apache Hadoop Version 4 (CDH4) reduces this vulnerability by maintaining redundant NameNodes. The data is replicated during normal operation as follows:

- CDH maintains redundant NameNodes on the first two nodes. One of the NameNodes is in active mode, and the other NameNode is in hot standby mode. If the active NameNode fails, then the role of active NameNode automatically fails over to the standby NameNode.
- The NameNode data is written to a mirrored partition so that the loss of a single disk can be tolerated. This mirroring is done at the factory as part of the operating system installation.
- The active NameNode records all changes in at least two JournalNode processes, which the standby NameNode reads. There are three JournalNodes, which run on node01 to node03.

Note: Oracle Big Data Appliance 2.0 and later releases do not support the use of an external NFS filer for backups and do not use NameNode federation.

Figure 2–7 shows the relationships among the processes that support automatic failover on Oracle Big Data Appliance.

Figure 2–7 Automatic Failover of the NameNode on Oracle Big Data Appliance



Unconfigured Software

The RPM installation files for the following tools are available on Oracle Big Data Appliance. Do not download them from the Cloudera website. However, you must install and configure them.

- Flume
- HBase
- Mahout
- Sqoop
- Whirr

You can find the RPM files on Oracle Big Data Appliance in /opt/oracle/BDAMammoth/bdarepo/RPMS/noarch.

> **See Also:** *CDH4 Installation and Configuration Guide* for configuration procedures at
>
> `http://oracle.cloudera.com`

Configuring HBase

HBase is an open-source, column-oriented database provided with CDH. HBase is not configured automatically on Oracle Big Data Appliance. You must set up and configure HBase before you can access it from an HBase client on another system.

To create an HBase service:

1. Open Cloudera Manager in a browser, using a URL like the following:

   ```
   http://bda1node03.example.com:7180
   ```

   In this example, `bda1` is the name of the appliance, `node03` is the name of the server, `example.com` is the domain, and `7180` is the default port number for Cloudera Manager.

2. On the All Services page, click **Add a Service**.

3. Select HBase from the list of services, then click **Continue**.
4. Select zookeeper, then click **Continue**.
5. Click **Continue** on the host assignments page.
6. Click **Accept** on the review page.

HBase is now ready for you to configure.

To configure HBase on Oracle Big Data Appliance:

1. On the All Services page of Cloudera Manager, click **hbase1**.
2. On the hbase1 page, click **Configuration**.
3. In the Category pane on the left, select **Advanced** under Service-Wide.
4. In the right pane, locate the HBase Service Configuration Safety Valve for hbase-site.xml property and click the Value cell.
5. Enter the following XML property descriptions:

```
<property>
   <name>hbase.master.ipc.address</name>
   <value>0.0.0.0</value>
</property>
<property>
   <name>hbase.regionserver.ipc.address</name>
   <value>0.0.0.0</value>
</property>
```

6. Click the **Save Changes** button.
7. From the Actions menu, select either **Start** or **Restart**, depending on the current status of the HBase server.
8. Log out of Cloudera Manager.

Effects of Hardware on Software Availability

The effects of a server failure vary depending on the server's function within the CDH cluster. Oracle Big Data Appliance servers are more robust than commodity hardware, so you should experience fewer hardware failures. This section highlights the most important services that run on the various servers of the primary rack. For a full list, see Table 2–2.

Critical and Noncritical Nodes

Critical nodes are required for the cluster to operate normally and provide all services to users. In contrast, the cluster continues to operate with no loss of service when a noncritical node fails.

The critical services are installed initially on the first three nodes of the primary rack. Table 2–3 identifies the critical services that run on these nodes. The remaining nodes (initially node04 to node18) only run noncritical services. If a hardware failure occurs on one of the critical nodes, then the services can be moved to another, noncritical server. For example, if node02 fails, its critical services might be moved to node05. Table 2–3 provides names to identify the nodes providing critical services.

Moving a critical node requires that all clients be reconfigured with the address of the new node. The other alternative is to wait for the repair of the failed server. You must weigh the loss of services against the inconvenience of reconfiguring the clients.

Table 2–3 Critical Nodes

Node Name	Initial Node Position	Critical Functions
First NameNode	Node01	ZooKeeper, first NameNode, failover controller, balancer, puppet master
Second NameNode	Node02	ZooKeeper, second NameNode, failover controller, MySQL backup server
JobTracker Node	Node03	ZooKeeper, JobTracker, Cloudera Manager server, Oracle Data Integrator agent, MySQL primary server, Hue, Hive, Oozie

First NameNode

One instance of the NameNode initially runs on node01. If this node fails or goes offline (such as a reboot), then the second NameNode (node02) automatically takes over to maintain the normal activities of the cluster.

Alternatively, if the second NameNode is already active, it continues without a backup. With only one NameNode, the cluster is vulnerable to failure. The cluster has lost the redundancy needed for automatic failover of the active NameNode.

These functions are also disrupted:

- **Balancer**: The balancer runs periodically to ensure that data is distributed evenly across the cluster. Balancing is not performed when the first NameNode is down.
- **Puppet master**: The Mammoth utilities use Puppet, and so you cannot install or reinstall the software if, for example, you must replace a disk drive elsewhere in the rack.

Second NameNode

One instance of the NameNode initially runs on node02. If this node fails, then the function of the NameNode either fails over to the first NameNode (node01) or continues there without a backup. However, the cluster has lost the redundancy needed for automatic failover if the first NameNode also fails.

These services are also disrupted:

- **MySQL Master Database**: Cloudera Manager, Oracle Data Integrator, Hive, and Oozie use MySQL Database. The data is replicated automatically, but you cannot access it when the master database server is down.
- **Oracle NoSQL Database KV Administration**: Oracle NoSQL Database database is an optional component of Oracle Big Data Appliance, so the extent of a disruption due to a node failure depends on whether you are using it and how critical it is to your applications.

JobTracker Node

The JobTracker assigns MapReduce tasks to specific nodes in the CDH cluster. Without the JobTracker node (node03), this critical function is not performed.

These services are also disrupted:

- **Cloudera Manager**: This tool provides central management for the entire CDH cluster. Without this tool, you can still monitor activities using the utilities described in "Using Hadoop Monitoring Utilities" on page 2-5.

- **Oracle Data Integrator**: This service supports Oracle Data Integrator Application Adapter for Hadoop. You cannot use this connector when the JobTracker node is down.
- **Hive**: Hive provides a SQL-like interface to data that is stored in HDFS. Most of the Oracle Big Data Connectors can access Hive tables, which are not available if this node fails.
- **MySQL Backup Database**: MySQL Server continues to run, although there is no backup of the master database.
- **Oozie**: This workflow and coordination service runs on the JobTracker node, and is unavailable when the node is down.

Noncritical Nodes

The noncritical nodes (node04 to node18) are optional in that Oracle Big Data Appliance continues to operate with no loss of service if a failure occurs. The NameNode automatically replicates the lost data to maintain three copies at all times. MapReduce jobs execute on copies of the data stored elsewhere in the cluster. The only loss is in computational power, because there are fewer servers on which to distribute the work.

Collecting Diagnostic Information for Oracle Customer Support

If you need help from Oracle Support to troubleshoot CDH issues, then you should first collect diagnostic information using the `bdadiag` utility with the `cm` option.

To collect diagnostic information:

1. Log in to an Oracle Big Data Appliance server as `root`.
2. Run `bdadiag` with at least the `cm` option. You can include additional options on the command line as appropriate. See the *Oracle Big Data Appliance Owner's Guide* for a complete description of the `bdadiag` syntax.

   ```
   # bdadiag cm
   ```

   The command output identifies the name and the location of the diagnostic file.
3. Go to My Oracle Support at `http://support.oracle.com`.
4. Open a Service Request (SR) if you have not already done so.
5. Upload the bz2 file into the SR. If the file is too large, then upload it to `ftp.oracle.com`, as described in the next procedure.

To upload the diagnostics to ftp.oracle.com:

1. Open an FTP client and connect to `ftp.oracle.com`.

   See Example 2–1 if you are using a command-line FTP client from Oracle Big Data Appliance.
2. Log in as user `anonymous` and leave the password field blank.
3. In the bda/incoming directory, create a directory using the SR number for the name, in the format SR*number*. The resulting directory structure looks like this:

   ```
   bda
      incoming
         SRnumber
   ```

4. Set the binary option to prevent corruption of binary data.
5. Upload the diagnostic bz2 file to the new directory.
6. Update the SR with the full path, which has the form bda/incoming/SR*number*, and the file name.

Example 2–1 shows the commands to upload the diagnostics using the FTP command interface on Oracle Big Data Appliance.

Example 2–1 Uploading Diagnostics Using FTP

```
# ftp
ftp> open ftp.oracle.com
Connected to bigip-ftp.oracle.com.
220-***********************************************************************
220-Oracle FTP Server
         .
         .
         .
220-****************************************************************************
220-
220
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (ftp.oracle.com:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd bda/incoming
250 Directory successfully changed.
ftp> mkdir SR12345
257 "/bda/incoming/SR12345" created
ftp> cd SR12345
250 Directory successfully changed.
ftp> bin
200 Switching to Binary mode.
ftp> put /tmp/bdadiag_bda1node01_1216FM5497_2013_01_18_07_33.tar.bz2
local: bdadiag_bda1node01_1216FM5497_2013_01_18_07_33.tar.bz2
remote: bdadiag_bda1node01_1216FM5497_2013_01_18_07_33.tar.bz2
227 Entering Passive Mode (141,146,44,21,212,32)
150 Ok to send data.
226 File receive OK.
2404836 bytes sent in 1.8 seconds (1.3e+03 Kbytes/s)
```

Security on Oracle Big Data Appliance

You can take precautions to prevent unauthorized use of the software and data on Oracle Big Data Appliance.

This section contains these topics:

- About Predefined Users and Groups
- Port Numbers Used on Oracle Big Data Appliance
- About CDH Security Using Kerberos
- About Puppet Security

About Predefined Users and Groups

Every open-source package installed on Oracle Big Data Appliance creates one or more users and groups. Most of these users do not have login privileges, shells, or home directories. They are used by daemons and are not intended as an interface for individual users. For example, Hadoop operates as the `hdfs` user, MapReduce operates as `mapred`, and Hive operates as `hive`.

You can use the `oracle` identity to run Hadoop and Hive jobs immediately after the Oracle Big Data Appliance software is installed. This user account has login privileges, a shell, and a home directory.

Oracle NoSQL Database and Oracle Data Integrator run as the `oracle` user. Its primary group is `oinstall`.

> **Note:** Do not delete or modify the users created during installation, because they are required for the software to operate.

Table 2–4 identifies the operating system users and groups that are created automatically during installation of Oracle Big Data Appliance software for use by CDH components and other software packages.

Table 2–4 Operating System Users and Groups

User Name	Group	Used By	Login Rights
`flume`	`flume`	**Flume** parent and nodes	No
`hbase`	`hbase`	**HBase** processes	No
`hdfs`	`hadoop`	**NameNode**, **DataNode**	No
`hive`	`hive`	Hive metastore and server processes	No
`hue`	`hue`	**Hue** processes	No
`mapred`	`hadoop`	JobTracker, TaskTracker, **Hive Thrift** daemon	Yes
`mysql`	`mysql`	MySQL server	Yes
`oozie`	`oozie`	**Oozie** server	No
`oracle`	`dba`, `oinstall`	Oracle NoSQL Database, Oracle Loader for Hadoop, Oracle Data Integrator, and the Oracle DBA	Yes
`puppet`	`puppet`	Puppet parent (puppet nodes run as `root`)	No
`sqoop`	`sqoop`	**Sqoop** metastore	No
`svctag`		Auto Service Request	No
`zookeeper`	`zookeeper`	**ZooKeeper** processes	No

Port Numbers Used on Oracle Big Data Appliance

Table 2–5 identifies the port numbers that might be used in addition to those used by CDH. For the full list of CDH port numbers, go to the Cloudera website at

`http://ccp.cloudera.com/display/CDH4DOC/Configuring+Ports+for+CDH4`

To view the ports used on a particular server:

1. In Cloudera Manager, click the **Hosts** tab at the top of the page to display the Hosts page.

2. In the Name column, click a server link to see its detail page.
3. Scroll down to the Ports section.

> **See Also:** The Cloudera website for CDH port numbers:
>
> - *Hadoop Default Ports Quick Reference* at
>
>   http://www.cloudera.com/blog/2009/08/hadoop-default-ports-quick-reference/
> - *Configuring Ports for CDH3* at
>
>   https://ccp.cloudera.com/display/CDHDOC/Configuring+Ports+for+CDH3

Table 2–5 Oracle Big Data Appliance Port Numbers

Service	Port
Automated Service Monitor (ASM)	30920
HBase master service (node01)	60010
MySQL Database	3306
Oracle Data Integrator Agent	20910
Oracle NoSQL Database administration	5001
Oracle NoSQL Database processes	5010 to 5020
Oracle NoSQL Database registration	5000
Port map	111
Puppet master service	8140
Puppet node service	8139
rpc.statd	668
ssh	22
xinetd (service tag)	6481

About CDH Security Using Kerberos

Apache Hadoop is not an inherently secure system. It is protected only by network security. After a connection is established, a client has full access to the system.

Cloudera's Distribution including Apache Hadoop (CDH) supports Kerberos network authentication protocol to prevent malicious impersonation. You must install and configure Kerberos and set up a Kerberos Key Distribution Center and realm. Then you configure various components of CDH to use Kerberos.

CDH provides these securities when configured to use Kerberos:

- The CDH master nodes, NameNode, and JobTracker resolve the group name so that users cannot manipulate their group memberships.
- Map tasks run under the identity of the user who submitted the job.
- Authorization mechanisms in HDFS and MapReduce help control user access to data.

See Also: http://oracle.cloudera.com for these manuals:

- *CDH4 Security Guide*
- *Configuring Hadoop Security with Cloudera Manager*
- *Configuring TLS Security for Cloudera Manager*

About Puppet Security

The puppet node service (puppetd) runs continuously as root on all servers. It listens on port 8139 for "kick" requests, which trigger it to request updates from the puppet master. It does not receive updates on this port.

The puppet master service (puppetmasterd) runs continuously as the puppet user on the first server of the primary Oracle Big Data Appliance rack. It listens on port 8140 for requests to push updates to puppet nodes.

The puppet nodes generate and send certificates to the puppet master to register initially during installation of the software. For updates to the software, the puppet master signals ("kicks") the puppet nodes, which then request all configuration changes from the puppet master node that they are registered with.

The puppet master sends updates only to puppet nodes that have known, valid certificates. Puppet nodes only accept updates from the puppet master host name they initially registered with. Because Oracle Big Data Appliance uses an internal network for communication within the rack, the puppet master host name resolves using /etc/hosts to an internal, private IP address.

3

Supporting User Access to Oracle Big Data Appliance

This chapter describes how you can support users who are running MapReduce jobs on Oracle Big Data Appliance or using Oracle Big Data Connectors. It contains these sections:

- Providing Remote Client Access to CDH
- Managing User Accounts
- Recovering Deleted Files

Providing Remote Client Access to CDH

Oracle Big Data Appliance supports full local access to all commands and utilities in Cloudera's Distribution including Apache Hadoop (CDH).

You can use a browser on any computer that has access to the client network of Oracle Big Data Appliance to access Cloudera Manager, Hadoop Map/Reduce Administration, the Hadoop Task Tracker interface, and other browser-based Hadoop tools.

To issue Hadoop commands remotely, however, you must connect from a system configured as a CDH client with access to the Oracle Big Data Appliance client network. This section explains how to set up a computer so that you can access HDFS and submit MapReduce jobs on Oracle Big Data Appliance.

See Also: My Oracle Support ID 1506203.1

Prerequisites

Ensure that you have met the following prerequisites:

- You must have these access privileges:
  - Root access to the client system
  - Login access to Cloudera Manager

  If you do not have these privileges, then contact your system administrator for help.
- The client system must run an operating system that Cloudera supports for CDH4. For the list of supported operating systems, see "Before You Install CDH4 on a Cluster" in the *Cloudera CDH4 Installation Guide* at

http://ccp.cloudera.com/display/CDH4DOC/Before+You+Install+CDH4+on+a+Cluster

- The client system must run the same version of Oracle JDK as Oracle Big Data Appliance. CDH4 requires Oracle JDK 1.6.

Installing CDH on Oracle Exadata Database Machine

When you use Oracle Exadata Database Machine as the client, you can use the RPM files on Oracle Big Data Appliance, because both engineered systems use the same operating system (Oracle Linux 5.*x*). Copying the files across the local network is faster than downloading them from the Cloudera website.

Note: In the following steps, replace *version_number* with the missing portion of the file name, such as 2.0.0+552-1.cdh4.1.2.p0.2.

To install a CDH client on Oracle Exadata Database Machine:

1. Log into an Exadata database server.
2. Verify that Hadoop is not installed on your Exadata system:

   ```
   rpm -qa | grep hadoop
   ```

3. If the `rpm` command returns a value, then remove the existing Hadoop software:

   ```
   rpm -e hadoop_rpm
   ```

4. Copy the following Linux RPMs to the database server from the first server of Oracle Big Data Appliance. The RPMs are located in the /opt/oracle/BDAMammoth/bdarepo/RPMS/x86_64 directory.
   - ed-*version_number*.x86_64.rpm
   - m4-*version_number*.x86_64.rpm
   - nc-*version_number*.x86_64.rpm
   - redhat-lsb-*version_number*.x86_64.rpm
5. Install the Oracle Linux RPMs from Step 4 on all database nodes. For example:

   ```
   sudo yum --nogpgcheck localinstall ed-0.2-39.el5_2.x86_64.rpm
   sudo yum --nogpgcheck localinstall m4-1.4.5-3.el5.1.x86_64.rpm
   sudo yum --nogpgcheck localinstall nc-1.84-10.fc6.x86_64.rpm
   sudo yum --nogpgcheck localinstall redhat-lsb-4.0-2.1.4.0.2.el5.x86_64.rpm
   ```

   Be sure to install the Oracle Linux RPMs before installing the CDH RPMs.
6. Copy the following CDH RPMs from the /opt/oracle/BDAMammoth/bdarepo/RPMS/noarch directory.
   - bigtop-utils-*version_number*.noarch.rpm
   - zookeeper-*version_number*.noarch.rpm
7. Copy the following CDH RPMs from the /opt/oracle/BDAMammoth/bdarepo/RPMS/x86_64 directory.
   - hadoop-*version_number*.x86_64.rpm
   - bigtop-jsvc-*version_number*.x86_64.rpm
   - hadoop-hdfs-*version_number*.x86_64.rpm

- hadoop-0.20-mapreduce-*version_number*.x86_64.rpm
- hadoop-yarn-*version_number*.x86_64.rpm
- hadoop-mapreduce-*version_number*.x86_64.rpm
- hadoop-client-*version_number*.x86_64.rpm

8. Install the CDH RPMs in the exact order shown in Steps 6 and 7 on all database servers. For example:

```
rpm -ihv bigtop-utils-0.4+359-1.cdh4.1.2.p0.34.el5.noarch.rpm
rpm -ihv zookeeper-3.4.3+28-1.cdh4.1.2.p0.34.el5.noarch.rpm
rpm -ihv hadoop-2.0.0+552-1.cdh4.1.2.p0.27.el5.x86_64.rpm
rpm -ihv bigtop-jsvc-0.4+359-1.cdh4.1.2.p0.43.el5.x86_64.rpm
rpm -ihv hadoop-hdfs-2.0.0+552-1.cdh4.1.2.p0.27.el5.x86_64.rpm
rpm -ihv hadoop-0.20-mapreduce-0.20.2+1265-1.cdh4.1.2.p0.24.el5.x86_64.rpm
rpm -ihv hadoop-yarn-2.0.0+552-1.cdh4.1.2.p0.27.el5.x86_64.rpm
rpm -ihv hadoop-mapreduce-2.0.0+552-1.cdh4.1.2.p0.27.el5.x86_64.rpm
rpm -ihv hadoop-client-2.0.0+552-1.cdh4.1.2.p0.27.el5.x86_64.rpm
```

9. Configure the CDH client. See "Configuring CDH" on page 3-3.

Installing a CDH Client on Any Supported Operating System

To install a CDH client on any operating system identified as supported by Cloudera, follow these instructions.

To install the CDH client software:

1. Follow the installation instructions for your operating system provided in the *Cloudera CDH4 Installation Guide* at

   `http://ccp.cloudera.com/display/CDH4DOC/CDH4+Installation+Guide`

   When you are done installing the Hadoop core and native packages, the system can act as a basic CDH client.

   > **Note:** Be sure to install CDH4 Update 1 (CDH4u1) or a later version.

2. To provide support for other components, such as **Hive**, **Pig**, or **Oozie**, see the component installation instructions.
3. Configure the CDH client. See "Configuring CDH" on page 3-3.

Configuring CDH

After installing CDH, you must configure it for use with Oracle Big Data Appliance.

To configure the Hadoop client:

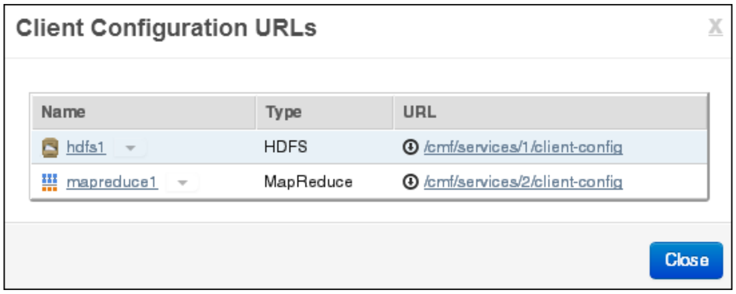
1. Open a browser on your client system and connect to Cloudera Manager. It runs on the JobTracker node (node03) and listens on port 7180, as shown in this example:

   ```
   http://bda1node03.example.com:7180
   ```

2. Log in as `admin`.
3. Cloudera Manager opens on the Services tab. Click the **Client Configuration URLs** button.

4. In the popup window, click the URL for mapreduce1 (/cmf/services/2/client-config) to download mapreduce1-clientconfig.zip.

   The following figure shows the download page for the client configuration.



Client Configuration URLs

Name	Type	URL
hdfs1	HDFS	/cmf/services/1/client-config
mapreduce1	MapReduce	/cmf/services/2/client-config

5. Log out of Cloudera Manager and navigate to the download directory.

6. Unzip mapreduce1-clientconfig.zip into a permanent location on the client system.

```
$ unzip mapreduce-clientconfig.zip
Archive:  mapreduce-clientconfig.zip
  inflating: hadoop-conf/hadoop-env.sh
  inflating: hadoop-conf/core-site.xml
  inflating: hadoop-conf/hdfs-site.xml
  inflating: hadoop-conf/log4j.properties
  inflating: hadoop-conf/mapred-site.xml
```

   All files are stored in a subdirectory named hadoop-config.

7. Open hadoop-env.sh in a text editor and set `JAVA_HOME` to the correct location on your system:

```
export JAVA_HOME=full_directory_path
```

8. Delete the number sign (#) to uncomment the line, and then save the file.

9. Make a backup copy of the Hadoop configuration files:

```
# cp /full_path/hadoop-conf /full_path/hadoop-conf-bak
```

10. Overwrite the existing configuration files with the downloaded configuration files in Step 6.

```
# cd /full_path/hadoop-conf
# cp * /usr/lib/hadoop/conf
```

11. Verify that you can access HDFS on Oracle Big Data Appliance from the client, by entering a simple Hadoop file system command like the following:

```
$ hadoop fs -ls /user
Found 4 items
drwx------   - hdfs    supergroup          0 2013-01-16 13:50 /user/hdfs
drwxr-xr-x   - hive    supergroup          0 2013-01-16 12:58 /user/hive
drwxr-xr-x   - oozie   hadoop              0 2013-01-16 13:01 /user/oozie
drwxr-xr-x   - oracle  hadoop              0 2013-01-29 12:50 /user/oracle
```

   Check the output for HDFS users defined on Oracle Big Data Appliance, and not on the client system. You should see the same results as you would after entering the command directly on Oracle Big Data Appliance.

12. Validate the installation by submitting a MapReduce job. You must be logged in to the host computer under the same user name as your HDFS user name on Oracle Big Data Appliance.

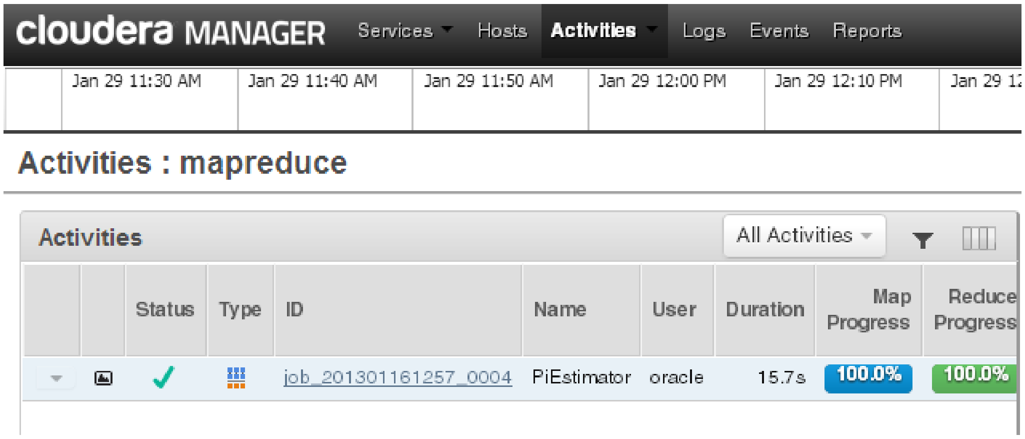
    The following example calculates the value of *pi*:

```
$ hadoop jar
/usr/lib/hadoop-mapreduce/hadoop-mapreduce-examples-2.0.0-cdh4.1.2.jar pi 10
1000000
Number of Maps  = 10
Samples per Map = 1000000
Wrote input for Map #0
Wrote input for Map #1
     .
     .
     .
Job Finished in 17.981 seconds
Estimated value of Pi is 3.14158440000000000000
```

13. Use Cloudera Manager to verify that the job ran on Oracle Big Data Appliance instead of the local system. Select **mapreduce** from the Activities menu for a list of jobs.

Figure 3–1 shows the job created by the previous example.

Figure 3–1 Monitoring a MapReduce Job in Cloudera Manager



		Status	Type	ID	Name	User	Duration	Map Progress	Reduce Progress
		✓		job_201301161257_0004	PiEstimator	oracle	15.7s	100.0%	100.0%

Managing User Accounts

This section describes the users created for use by the software, and explains how to create additional users. It contains the following topics:

- About Predefined Users and Groups
- Creating New HDFS Users

Providing User Login Privileges

Users do not need login privileges on Oracle Big Data Appliance to run MapReduce jobs from a remote client. However, for those who want to log in to Oracle Big Data Appliance, you must set a password. You can set or reset a password the same way.

To set a user password across all Oracle Big Data Appliance servers:

1. Create an HDFS user as described in the previous procedure.

2. Confirm that the user does not have a password:

   ```
   # dcli passwd -S user_name
   bda1node01.example.com: jdoe NP 2013-01-22 0 99999 7 -1 (Empty password.)
   bda1node02.example.com: jdoe NP 2013-01-22 0 99999 7 -1 (Empty password.)
   bda1node03.example.com: jdoe NP 2013-01-22 0 99999 7 -1 (Empty password.)
   ```

   If the output shows either "Empty password" or "Password locked," then you must set a password.

3. Set the password:

   ```
    hash=$(echo 'password' | openssl passwd -1 -stdin); dcli "usermod
   --pass='$hash' user_name"
   ```

4. Confirm that the password is set across all servers:

   ```
   ]# dcli passwd -S user_name
   bda1node01.example.com: jdoe PS 2013-01-24 0 99999 7 -1 (Password set, MD5
   crypt.)
   bda1node02.example.com: jdoe PS 2013-01-24 0 99999 7 -1 (Password set, MD5
   crypt.)
   bda1node03.example.com: jdoe PS 2013-01-24 0 99999 7 -1 (Password set, MD5
   crypt.)
   ```

See Also:

- *Oracle Big Data Appliance Owner's Guide* for information about dcli.
- The Linux man page for the full syntax of the useradd command.

Creating New HDFS Users

When creating additional user accounts, define them as follows:

- To run MapReduce jobs, users must be in the hadoop group.
- To create and modify tables in Hive, users must be in the hive group.
- To create Hue users, open Hue in a browser and click the User Admin icon. See "Using Hue to Interact With Hadoop" on page 2-6.

To create an HDFS user:

1. Open an ssh connection as the root user to a noncritical node (node04 to node18).
2. Create the user's home directory:

   ```
   # sudo -u hdfs hadoop fs -mkdir /user/user_name
   ```

   You use sudo because the HDFS super user is hdfs (not root).

3. Change the ownership of the directory:

   ```
   # sudo -u hdfs hadoop fs -chown user_name:hadoop /user/user_name
   ```

4. Verify that the directory is set up correctly:

   ```
   # hadoop fs -ls /user
   ```

5. Create the operating system user across all nodes in the cluster:

   ```
   # dcli useradd -G hadoop,hive[,group_name...] -m user_name
   ```

In this syntax, replace *group_name* with an existing group and *user_name* with the new name.

6. Verify that the operating system user belongs to the correct groups:

   ```
   # dcli id user_name
   ```

7. Verify that the user's home directory was created on all nodes:

   ```
   # dcli ls /home | grep user_name
   ```

Example 3–1 creates a user named `jdoe` with a primary group of `hadoop` and an addition group of `hive`.

Example 3–1 Creating a Hadoop User

```
]# sudo -u hdfs hadoop fs -mkdir /user/jdoe
# sudo -u hdfs hadoop fs -chown jdoe:hadoop /user/jdoe
# hadoop fs -ls /user
Found 5 items
drwx------   - hdfs     supergroup          0 2013-01-16 13:50 /user/hdfs
drwxr-xr-x   - hive     supergroup          0 2013-01-16 12:58 /user/hive
drwxr-xr-x   - jdoe     jdoe                0 2013-01-18 14:04 /user/jdoe
drwxr-xr-x   - oozie    hadoop              0 2013-01-16 13:01 /user/oozie
drwxr-xr-x   - oracle   hadoop              0 2013-01-16 13:01 /user/oracle
# dcli useradd -G hadoop,hive -m jdoe]
# dcli id jdoe
bda1node01: uid=1001(jdoe) gid=1003(jdoe) groups=1003(jdoe),127(hive),123(hadoop)
bda1node02: uid=1001(jdoe) gid=1003(jdoe) groups=1003(jdoe),123(hadoop),127(hive)
bda1node03: uid=1001(jdoe) gid=1003(jdoe) groups=1003(jdoe),123(hadoop),127(hive)
     .
     .
     .
# dcli ls /home | grep jdoe
bda1node01: jdoe
bda1node02: jdoe
bda1node03: jdoe
```

Recovering Deleted Files

CDH provides an optional trash facility, so that a deleted file or directory is moved to a trash directory for a set period of time instead of being deleted immediately from the system. By default, the trash facility is enabled for HDFS and all HDFS clients.

Restoring Files from the Trash

When the trash facility is enabled, you can easily restore files that were previously deleted.

To restore a file from the trash directory:

1. Check that the deleted file is in the trash. The following example checks for files deleted by the `oracle` user:

   ```
   $ hadoop fs -ls .Trash/Current/user/oracle
   Found 1 items
   -rw-r--r--  3 oracle hadoop  242510990 2012-08-31 11:20
   /user/oracle/.Trash/Current/user/oracle/ontime_s.dat
   ```

2. Move or copy the file to its previous location. The following example moves ontime_s.dat from the trash to the HDFS /user/oracle directory.

```
$ hadoop fs -mv .Trash/Current/user/oracle/ontime_s.dat /user/oracle/ontime_
s.dat
```

Changing the Trash Interval

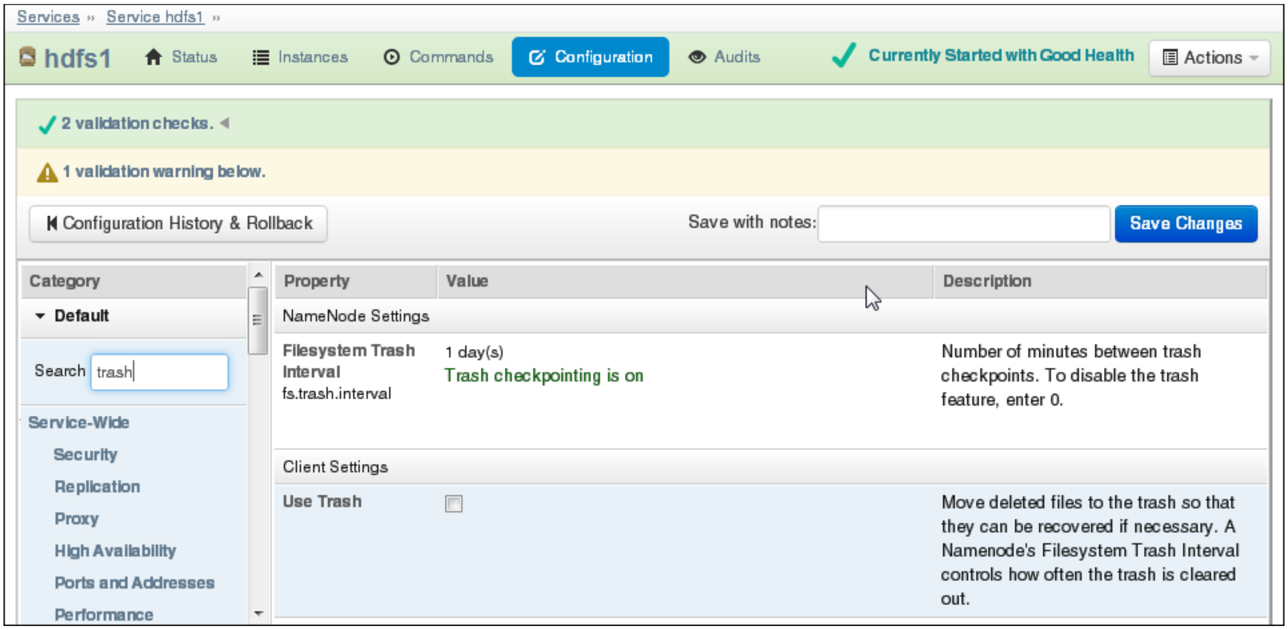
The **trash interval** is the minimum number of minutes that a file remains in the trash directory before being deleted permanently from the system. The default value is 1 day (24 hours).

To change the trash interval:

1. Open Cloudera Manager. See "Managing CDH Operations Using Cloudera Manager" on page 2-2.
2. On the All Services page under Name, click **hdfs1**.
3. On the hdfs1 page, click the **Configuration** subtab.
4. Search for or scroll down to the Filesystem Trash Interval property under NameNode Settings. See Figure 3–2.
5. Click the current value, and enter a new value in the pop-up form.
6. Click **Save Changes**.
7. Expand the Actions menu at the top of the page and choose **Restart**.

Figure 3–2 shows the Filesystem Trash Interval property in Cloudera Manager.

Figure 3–2 HDFS Property Settings in Cloudera Manager



✓ Currently Started with Good Health

✓ 2 validation checks.

⚠ 1 validation warning below.

Search: trash

Property	Value	Description
NameNode Settings		
Filesystem Trash Interval fs.trash.interval	1 day(s) Trash checkpointing is on	Number of minutes between trash checkpoints. To disable the trash feature, enter 0.
Client Settings		
Use Trash	☐	Move deleted files to the trash so that they can be recovered if necessary. A Namenode's Filesystem Trash Interval controls how often the trash is cleared out.

Disabling the Trash Facility

The trash facility on Oracle Big Data Appliance is enabled by default. You can change this configuration at the server or the client level. When the trash facility is disabled, deleted files and directories are not moved to the trash. They are not recoverable.

Completely Disabling the Trash Facility

The following procedure disables the trash facility for HDFS. When the trash facility is completely disabled, the client configuration is irrelevant.

To completely disable the trash facility:

1. Open Cloudera Manager. See "Managing CDH Operations Using Cloudera Manager" on page 2-2.
2. On the All Services page under Name, click **hdfs1**.
3. On the hdfs1 page, click the **Configuration** subtab.
4. Search for or scroll down to the Filesystem Trash Interval property under NameNode Settings. See Figure 3–2.
5. Click the current value, and enter a value of 0 (zero) in the pop-up form.
6. Click **Save Changes**.
7. Expand the Actions menu at the top of the page and choose **Restart**.

Disabling the Trash Facility for Local HDFS Clients

All HDFS clients that are installed on Oracle Big Data Appliance are configured to use the trash facility. An **HDFS client** is any software that connects to HDFS to perform operations such as listing HDFS files, copying files to and from HDFS, and creating directories.

You can use Cloudera Manager to change the local client configuration setting, although the trash facility is still enabled.

> **Note:** If you do not want any clients to use the trash, then you can completely disable the trash facility. See "Completely Disabling the Trash Facility" on page 3-9.

To disable the trash facility for local HDFS clients:

1. Open Cloudera Manager. See "Managing CDH Operations Using Cloudera Manager" on page 2-2.
2. On the All Services page under Name, click **hdfs1**.
3. On the hdfs1 page, click the **Configuration** subtab.
4. Search for or scroll down to the Use Trash property under Client Settings. See Figure 3–2.
5. Deselect the Use Trash check box.
6. Click **Save Changes**. This setting is used to configure all new HDFS clients downloaded to Oracle Big Data Appliance.
7. To deploy the new configuration to existing clients:
   a. Expand the Actions menu and choose **Deploy Client Configuration**.
   b. At the prompt to confirm the action, click **Deploy Client Configuration**.

Disabling the Trash Facility for a Remote HDFS Client

Remote HDFS clients are typically configured by downloading and installing a CDH client, as described in "Providing Remote Client Access to CDH" on page 3-1. Oracle

SQL Connector for HDFS and Oracle R Connector for Hadoop are examples of remote clients.

To disable the trash facility for a remote HDFS client:

1. Open a connection to the system where the CDH client is installed.
2. Open /etc/hadoop/conf/hdfs-site.xml in a text editor.
3. Change the trash interval to zero:

```
<property>
     <name>fs.trash.interval</name>
     <value>0</value>
</property>
```

4. Save the file.

4

Configuring Oracle Exadata Database Machine for Use with Oracle Big Data Appliance

This chapter provides information about optimizing communications between Oracle Exadata Database Machine and Oracle Big Data Appliance. It contains the following sections:

- About Optimizing Communications
- Prerequisites
- Enabling SDP on Exadata Database Nodes
- Configuring a JDBC Client for SDP
- Creating an SDP Listener on the InfiniBand Network

About Optimizing Communications

Sockets Direct Protocol (SDP) is a standard communication protocol for clustered server environments, providing an interface between the network interface card and the application. By using SDP, applications place most of the messaging burden upon the network interface card, which frees the CPU for other tasks. As a result, SDP decreases network latency and CPU utilization, and thereby improves performance.

This chapter describe how you can configure Oracle Exadata Database Machine to use SDP over InfiniBand to communicate with Oracle Big Data Appliance.

Prerequisites

Oracle Big Data Appliance and Oracle Exadata Database Machine racks must be cabled together using InfiniBand cables. The IP addresses must be unique across all racks and use the same subnet for the InfiniBand network.

See Also:

- *Oracle Big Data Appliance Owner's Guide* about multirack cabling
- *Oracle Big Data Appliance Configuration Worksheets* about IP addresses and subnets

Enabling SDP on Exadata Database Nodes

The following procedure describes how to enable SDP on database nodes in an Oracle Exadata Database Machine running Oracle Linux.

To enable SDP on Oracle Exadata Database Machine:

1. Open /etc/infiniband/openib.conf file in a text editor, and add the following line:

   ```
   set: SDP_LOAD=yes
   ```

2. Save these changes and close the file.
3. To enable both SDP and TCP, open /etc/ofed/libsdp.conf in a text editor, and add the `use both` rule:

   ```
   use both server * :
   use both client * :
   ```

4. Save these changes and close the file.
5. Open `/etc/modprobe.conf` file in a text editor, and add this setting:

   ```
   options ib_sdp sdp_zcopy_thresh=0 recv_poll=0
   ```

6. Save these changes and close the file.
7. Replicate these changes across all servers in the Oracle Exadata Database Machine rack.
8. Restart all database nodes for the changes to take effect.
9. If you have multiple Oracle Exadata Database Machine racks, then repeat these steps on all of them.

Configuring a JDBC Client for SDP

The following procedure explains how to configure a JDBC client to use SDP.

To enable SDP support for JDBC:

1. Configure the database to support InfiniBand, as described in the *Oracle Database Net Services Administrator's Guide*. Ensure that you set the protocol to SDP.
2. Set the `LD_PRELOAD` environment variable to libsdp.so before starting the Java virtual machine. This example uses the Bash shell:

   ```
   export LD_PRELOAD="libsdp.so"
   ```

The following steps are an alternative to setting `LD_PRELOAD`:

1. In the JDBC URL, replace `TCP` protocol with `SDP` protocol. For example:

   ```
   jdbc:oracle:thin:@(DESCRIPTION=(ADDRESS=(PROTOCOL=sdp)(HOST=xxx.x.x.x)(PORT=152
   2))(CONNECT_DATA=(SERVICE_NAME=myservice)))
   ```

2. Open *DOMAIN_HOME*/bin/startWebLogic.sh in a text editor and make the following change:

   a. Locate this line in the file:

      ```
      . ${DOMAIN_HOME}/bin/setDomainEnv.sh $*
      ```

   b. Add this property immediately *after* the previous line:

      ```
      JAVA_OPTIONS="${JAVA_OPTIONS} -Djava.net.preferIPv4Stack=true
      -Doracle.net.SDP=true"
      ```

   c. Save and close the file.

Creating an SDP Listener on the InfiniBand Network

To add a listener for the Oracle Big Data Appliance connections coming in on the InfiniBand network, first add a network resource for the InfiniBand network with virtual IP addresses.

> **Note:** This example lists two nodes for an Oracle Exadata Database Machine quarter rack. If you have an Oracle Exadata Database Machine half or full rack, you must repeat node-specific lines for each node in the cluster.

1. Edit `/etc/hosts` on each node in the Exadata rack to add the virtual IP addresses for the InfiniBand network. Make sure that these IP addresses are not in use. For example:

   ```
   # Added for Listener over IB
   192.168.10.21 dm01db01-ibvip.example.com dm01db01-ibvip
   192.168.10.22 dm01db02-ibvip.example.com dm01db02-ibvip
   ```

2. As the `root` user, create a network resource on one database node for the InfiniBand network. For example:

   ```
   # /u01/app/grid/product/11.2.0.2/bin/srvctl add network -k 2 -S
   192.168.10.0/255.255.255.0/bondib0
   ```

3. Verify that the network was added correctly with one of the following commands:

   ```
   # /u01/app/grid/product/11.2.0.2/bin/crsctl stat res -t | grep net
   ora.net1.network
   ora.net2.network -- Output indicating new Network resource
   ```

   or

   ```
   # /u01/app/grid/product/11.2.0.2/bin/srvctl config network -k 2
   Network exists: 2/192.168.10.0/255.255.255.0/bondib0, type static -- Output
   indicating Network resource on the 192.168.10.0 subnet
   ```

4. Add the virtual IP addresses on the network created in Step 2, for each node in the cluster:

   ```
   # srvctl add vip -n dm01db01 -A dm01db01-ibvip/255.255.255.0/bondib0 -k 2
   # srvctl add vip -n dm01db02 -A dm01db02-ibvip/255.255.255.0/bondib0 -k 2
   ```

5. As the `oracle` user, who owns Grid Infrastructure Home, add a listener for the virtual IP addresses created in Step 4.

   ```
   # srvctl add listener -l LISTENER_IB -k 2 -p TCP:1522,/SDP:1522
   ```

6. For each database that will accept connections from the middle tier, modify the `listener_networks init` parameter to allow load balancing and failover across multiple networks (Ethernet and InfiniBand). You can either enter the full `TNSNAMES` syntax in the initialization parameter or create entries in `tnsnames.ora` in the `$ORACLE_HOME/network/admin` directory. The `TNSNAMES.ORA` entries must exist in `GRID_HOME`. The following example first updates `tnsnames.ora`.

   Complete this step on each node in the cluster with the correct IP addresses for that node. `LISTENER_IBREMOTE` should list all other nodes that are in the cluster. `DBM_IB` should list all nodes in the cluster.

> **Note:** The TNSNAMES entry is only read by the database instance on startup, if you modify the entry that is referred to by any init.ora parameter (LISTENER_NETWORKS), you must restart the instance or issue an ALTER SYSTEM SET LISTENER_NETWORKS command for the modifications to take affect by the instance.

```
DBM =
(DESCRIPTION =
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01-scan)(PORT = 1521))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = dbm)
))

DBM_IB =
(DESCRIPTION =
(LOAD_BALANCE=on)
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01db01-ibvip)(PORT = 1522))
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01db02-ibvip)(PORT = 1522))
(CONNECT_DATA =
(SERVER = DEDICATED)
(SERVICE_NAME = dbm)
))

LISTENER_IBREMOTE =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01db02-ibvip.mycompany.com)(PORT = 1522))
))

LISTENER_IBLOCAL =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01db01-ibvip.mycompany.com)(PORT = 1522))
(ADDRESS = (PROTOCOL = SDP)(HOST = dm01db01-ibvip.mycompany.com)(PORT = 1522))
))

LISTENER_IPLOCAL =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = dm0101-vip.mycompany.com)(PORT = 1521))
))

LISTENER_IPREMOTE =
(DESCRIPTION =
(ADDRESS_LIST =
(ADDRESS = (PROTOCOL = TCP)(HOST = dm01-scan.mycompany.com)(PORT = 1521))
))
```

7. Connect to the database instance as sysdba.
8. Modify the listener_networks init parameter:

```
SQL> alter system set listener_networks=
     '((NAME=network2) (LOCAL_LISTENER=LISTENER_IBLOCAL)
        (REMOTE_LISTENER=LISTENER_IBREMOTE))',
     '((NAME=network1)(LOCAL_LISTENER=LISTENER_IPLOCAL)
        (REMOTE_LISTENER=LISTENER_IPREMOTE))' scope=both;
```

9. Restart `LISTENER_IB` to implement the modification in Step 7:

```
# srvctl stop listener -l LISTENER_IB
# srvctl start listener -l LISTENER_IB
```

Glossary

ASR

Oracle Auto Service Request, a software tool that monitors the health of the hardware and automatically generates a service request if it detects a problem.

See also **OASM**.

Balancer

A service that ensures that all nodes in the cluster store about the same amount of data, within a set range. Data is balanced over the nodes in the cluster, not over the disks in a node.

CDH

Cloudera's Distribution including Apache Hadoop, the version of Apache Hadoop and related components installed on Oracle Big Data Appliance.

Cloudera's Distribution including Apache Hadoop (CDH)

See **CDH**.

cluster

A group of servers on a network that are configured to work together. A server is either a master node or a worker node.

All servers in an Oracle Big Data Appliance rack form a cluster. Servers 1, 2, and 3 are master nodes. Servers 4 to 18 are worker nodes.

See **Hadoop**.

DataNode

A server in a CDH cluster that stores data in HDFS. A DataNode performs file system operations assigned by the NameNode.

See also **HDFS**; **NameNode**.

Flume

A distributed service in CDH for collecting and aggregating data from almost any source into a data store such as HDFS or HBase.

See also **HBase**; **HDFS**.

Hadoop

A batch processing infrastructure that stores files and distributes work across a group of servers. Oracle Big Data Appliance uses Cloudera's Distribution including Apache Hadoop (CDH).

Hadoop Distributed File System (HDFS)

See **HDFS**.

Hadoop User Experience (Hue)

See **Hue**.

HBase

An open-source, column-oriented database that provides random, read/write access to large amounts of sparse data stored in a CDH cluster. It provides fast lookup of values by key and can perform thousands of insert, update, and delete operations per second.

HDFS

Hadoop Distributed File System, an open-source file system designed to store extremely large data files (megabytes to petabytes) with streaming data access patterns. HDFS splits these files into data blocks and distributes the blocks across a CDH cluster.

When a data set is larger than the storage capacity of a single computer, then it must be partitioned across several computers. A distributed file system can manage the storage of a data set across a network of computers.

See also **cluster**.

Hive

An open-source data warehouse in CDH that supports data summarization, ad hoc querying, and data analysis of data stored in HDFS. It uses a SQL-like language called HiveQL. An interpreter generates MapReduce code from the HiveQL queries.

By using Hive, you can avoid writing MapReduce programs in Java.

See also **Hive Thrift**; **HiveQL**; **MapReduce**.

Hive Thrift

A remote procedure call (RPC) interface for remote access to CDH for Hive queries.

See also **CDH**; **Hive**.

HiveQL

A SQL-like query language used by Hive.

See also **Hive**.

HotSpot

A Java Virtual Machine (JVM) that is maintained and distributed by Oracle. It automatically optimizes code that executes frequently, leading to high performance. HotSpot is the standard JVM for the other components of the Oracle Big Data Appliance stack.

Hue

Hadoop User Experience, a web user interface in CDH that includes several applications, including a file browser for HDFS, a job browser, an account management tool, a MapReduce job designer, and Hive wizards. Cloudera Manager runs on Hue.

See also **HDFS**; **Hive**.

Java HotSpot Virtual Machine

See **HotSpot**.

JobTracker

A service that assigns MapReduce tasks to specific nodes in the CDH cluster, preferably those nodes storing the data.

See also **Hadoop**; **MapReduce**.

MapReduce

A parallel programming model for processing data on a distributed system.

A MapReduce program contains these functions:

- Mappers: Process the records of the data set.
- Reducers: Merge the output from several mappers.
- Combiners: Optimizes the result sets from the mappers before sending them to the reducers (optional).

MySQL Server

A SQL-based relational database management system. Cloudera Manager, Oracle Data Integrator, Hive, and Oozie use MySQL Server as a metadata repository on Oracle Big Data Appliance.

NameNode

A service that maintains a directory of all files in HDFS and tracks where data is stored in the CDH cluster.

See also **HDFS**.

node

A server in a CDH cluster.

See also **cluster**.

NoSQL Database

See **Oracle NoSQL Database**.

OASM

Oracle Automated Service Manager, a service for monitoring the health of Oracle Sun hardware systems. Formerly named Sun Automatic Service Manager (SASM).

Oozie

An open-source workflow and coordination service for managing data processing jobs in CDH.

Oracle Database Instant Client

A small-footprint client that enables Oracle applications to run without a standard Oracle Database client.

Oracle Linux

An open-source operating system. Oracle Linux 5.6 is the same version used by Exalogic 1.1. It features the Oracle Unbreakable Enterprise Kernel.

Oracle NoSQL Database

A distributed key-value database that supports fast querying of the data, typically by key lookup.

Oracle R Distribution

An Oracle-supported distribution of the R open-source language and environment for statistical analysis and graphing.

Oracle R Enterprise

A component of the Oracle Advanced Analytics Option. It enables R users to run R commands and scripts for statistical and graphical analyses on data stored in an Oracle database.

Pig

An open-source platform for analyzing large data sets that consists of the following:

- Pig Latin scripting language
- Pig interpreter that converts Pig Latin scripts into MapReduce jobs

Pig runs as a client application.

See also **MapReduce**.

Puppet

A configuration management tool for deploying and configuring software components across a cluster. The Oracle Big Data Appliance initial software installation uses Puppet.

The Puppet tool consists of these components: puppet agents, typically just called puppets; the puppet master server; a console; and a cloud provisioner.

See also **puppet agent**; **puppet master**.

puppet agent

A service that primarily pulls configurations from the puppet master and applies them. Puppet agents run on every server in Oracle Big Data Appliance.

See also **Puppet**; **puppet master**

puppet master

A service that primarily serves configurations to the puppet agents.

See also **Puppet**; **puppet agent**.

Sqoop

A command-line tool that imports and exports data between HDFS or Hive and structured databases. The name Sqoop comes from "SQL to Hadoop." Oracle R Connector for Hadoop uses the Sqoop executable to move data between HDFS and Oracle Database.

table

In Hive, all files in a directory stored in HDFS.

See also **HDFS**.

TaskTracker

A service that runs on each node and executes the tasks assigned to it by the JobTracker service.

See also **JobTracker**.

ZooKeeper

A centralized coordination service for CDH distributed processes that maintains configuration information and naming, and provides distributed synchronization and group services.

Index

A

application adapters, 1-9, 2-10
Automated Service Manager
  See OASM

B

balancer, 2-9
  node location, 2-13
Beeswax server, 2-10
Berkeley DB, 1-5
big data description, 1-1
business intelligence, 1-3, 1-5, 1-9

C

CDH
  about, 1-3
  diagnostics, 2-14
  file system, 1-5
  remote client access, 3-1
  security, 2-17
  version, 2-8
chunking files, 1-5
client configuration, 3-1
Cloudera Manager
  about, 2-2
  accessing administrative tools, 2-4
  connecting to, 2-3
  effect of hardware failure on, 2-13
  software dependencies, 2-13
  software services, 2-9
  UI overview, 2-3
  version, 2-8
Cloudera's Distribution including Apache Hadoop
  See CDH
clusters, definition, 1-3
CSV files, 1-8

D

Data Pump files, 1-8
data replication, 1-5
DataNode, 2-9, 2-12
dba group, 2-16
diagnostics, collecting, 2-14
disks, 2-8
duplicating data, 1-5

E

engineered systems, 1-3
Exadata Database Machine, 1-3
Exalytics In-Memory Machine, 1-3
external tables, 1-8

F

files, recovering HDFS, 3-7
first NameNode,
  NameNode
    first, 2-13
Flume, 2-11, 2-16
ftp.oracle.com, uploading to, 2-14

G

groups, 2-16, 3-5

H

Hadoop Distributed File System
  See HDFS
hadoop group, 3-5
Hadoop Map/Reduce Administration, 2-5
Hadoop version, 1-3
HBase, 2-11, 2-16
HBase configuration, 2-11
HDFS
  about, 1-3, 1-5
  user identity, 2-16
HDFS data files, 1-8
Hive, 2-16
  about, 1-6
  node location, 2-14
  software dependencies, 2-13
  software services, 2-10
  tables, 3-6
  user identity, 2-16
hive group, 3-5
HiveQL, 1-6
HotSpot

See Java HotSpot Virtual Machine
Hue
software services, 2-10
user identity, 2-16

I

installing CDH client, 3-1

J

Java HotSpot Virtual Machine, 2-8
JobTracker, 2-10
about, 2-13
monitoring, 2-5
opening, 2-5
security, 2-17
user identity, 2-16
JobTracker node, 2-13

K

Kerberos network authentication, 2-17
key-value database, 1-5, Glossary-4
knowledge modules, 1-9

L

Linux
disk location, 2-8
installation, 2-7
loading data, 1-8

M

Mahout, 2-11
mapred user, 2-16
MapReduce, 1-4, 1-6, 2-17, 3-6
monitoring
JobTracker, 2-5
TaskTracker, 2-5
MySQL Database
about, 2-13
backup location, 2-10, 2-14
port number, 2-17
software services, 2-10
user identity, 2-16
version, 2-8

N

NameNode, 2-9, 2-10, 2-17
NoSQL databases
See also Oracle NoSQL Database

O

OASM, port number, 2-17
ODI
See Oracle Data Integrator
oinstall group, 2-16, 3-5
Oozie
software dependencies, 2-13
software services, 2-16
user identity, 2-16
Oozie service, 2-14
operating system users, 2-16
Oracle Automated Service Manager
See OASM
Oracle Data Integrator, 1-9
about, 1-8
node location, 2-14
software dependencies, 2-13
version, 2-8
Oracle Data Integrator agent, 2-10, 2-17
Oracle Data Pump files, 1-8
Oracle Database Instant Client, 2-8
Oracle Direct Connector for Hadoop Distributed File System, 1-8
Oracle Exadata Database Machine, 1-3
using as a CDH client, 3-2
Oracle Exalytics In-Memory Machine, 1-3
Oracle Linux
about, 1-3
relationship to HDFS, 1-4
version, 2-7
Oracle Loader for Hadoop, 1-8, 2-8
Oracle NoSQL Database
about, 1-5
KV Administration, 2-13
port numbers, 2-17
software services, 2-10
version, 2-8
Oracle R Connector for Hadoop, 1-8, 2-8
Oracle R Enterprise, 1-7
Oracle Support, creating a service request, 2-14
oracle user, 2-16, 3-5

P

partitioning, 2-8
planning applications, 1-3
port map, 2-17
port numbers, 2-16, 2-17
puppet
port numbers, 2-17
security, 2-18
software services, 2-10
user identity, 2-16
puppet master
node location, 2-13

R

R Connector
See Oracle R Connector for Hadoop
R distribution, 2-8
R language support, 1-7
recovering HDFS files, 3-7
remote client access, 3-1
replicating data, 1-5

rpc.statd service, 2-17

S

second NameNode, NameNode
  second, 2-13
Secondary NameNode, 2-10
security, 2-15
service requests, creating for CDH, 2-14
service tags, 2-17
services
  See software services
software components, 2-7
software framework, 1-3
software services
  monitoring, 2-9
  node locations, 2-9
  port numbers, 2-17
Sqoop, 2-11, 2-16
ssh service, 2-17
svctag user, 2-16

T

tables, 1-8, 3-6
Task Tracker Status interface, 2-5
TaskTracker
  monitoring, 2-5
  node location, 2-10
  user identity, 2-16
trash facility, 3-7
troubleshooting CDH, 2-14

U

user groups, 3-5
users
  Cloudera Manager, 2-4
  operating system, 2-16

W

Whirr, 2-11

X

xinetd service, 2-17

Z

ZooKeeper, 2-16