Oracle® Big Data SQL

Installation Guide

Release 3 (3.1)

E77165-10

May 2018

Oracle Big Data SQL Installation Guide, Release 3 (3.1)

E77165-10

Contents

Preface

1 Introduction

2 Installing the Hadoop Side of Oracle Big Data SQL

3 Installing the Oracle Database Side of Oracle Big Data SQL

4 Uninstalling Oracle Big Data SQL

5 Adapting to Hadoop Cluster and Oracle Database Changes

6 Securing Big Data SQL

7 Additional Tools Installed

A Oracle Big Data SQL Install/Uninstall/Reconfigure Examples

B Oracle Big Data SQL Licensing

Index

Preface

The *Oracle Big Data SQL User's Guide* describes how to install and manage the Oracle Big Data SQL product.

Audience

This guide is intended for installers and users of Oracle Big Data SQL. including:

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

The guide assumes that the reader has sufficient background knowledge about the database server and the particular Hadoop platform which will host the software in order to follow the instructions successfully.

Related Documents

See the *Oracle Big Data SQL User's Guide* for instructions on using the product.

The following publications provide information about the use of Oracle Big Data SQL with the Oracle Big Data Appliance and Oracle Big Data Connectors:

- *Oracle Big Data Appliance Owner's Guide*
- *Oracle Big Data Appliance Software User's Guide*
- *Oracle Big Data Connectors User's Guide*

You can find more information about Oracle's Big Data solutions and Oracle Database at the Oracle Help Center

For more information on Hortonworks HDP and Ambari, refer to the Hortonworks documentation site at http://docs.hortonworks.com/index.html.

For more information on Cloudera CDH and Configuration Manager, see http://www.cloudera.com/documentation.html

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.
# prompt	The pound (#) prompt indicates a command that is run as the Linux `root` user.

Backus-Naur Form Syntax

The syntax in this reference is presented in a simple variation of Backus-Naur Form (BNF) that uses the following symbols and conventions:

Symbol or Convention	Description
[ ]	Brackets enclose optional items.
{ }	Braces enclose a choice of items, only one of which is required.
|	A vertical bar separates alternatives within brackets or braces.
...	Ellipses indicate that the preceding syntactic element can be repeated.
delimiters	Delimiters other than brackets, braces, and vertical bars must be entered as shown.
boldface	Words appearing in boldface are keywords. They must be typed as shown. (Keywords are case-sensitive in some, but not all, operating systems.) Words that are not in boldface are placeholders for which you must substitute a name or value.

Changes in Oracle Big Data SQL 3.1

The following are changes in Oracle Big Data SQL Release 3.1.

Expanded Deployments

Release 3.1 broadens support for Oracle Big Data SQL connectivity between Oracle Engineered Systems and commodity servers.

In earlier Oracle Big Data SQL releases, the following Oracle Database/Hadoop connections are possible:

- Oracle Exadata Database Machine to Oracle Big Data Appliance.
- Oracle Database on commodity servers to commodity Hadoop systems.

As of Release 3.1, Oracle Big Data SQL supports all of the following Oracle Database/ Hadoop system connections:

- Oracle Database on commodity servers to Oracle Big Data Appliance.
- Oracle Database on commodity servers to commodity Hadoop systems.
- Oracle Exadata Database Machine to Oracle Big Data Appliance.

- Oracle Exadata Database Machine to commodity Hadoop systems.

The phrase “Oracle Database on commodity servers” refers to non-Exadata Linux systems that are officially-supported as Oracle Database platforms. “Commodity Hadoop systems” refers to Hortonworks HDP systems or to Cloudera CDH-based systems other than Oracle Big Data Appliance. In all cases, Oracle Database servers and Hadoop systems must meet the prerequisites identified in the Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support).

Oracle SPARC SuperCluster Support

Release 3.1 provides support for Oracle SPARC SuperCluster, with certain limitations:

- Ethernet connections between BDA (and commodity Hadoop systems) and SPARC SuperCluster are not supported.
- Oracle Database Tablepaces in HDFS (with Smart Scan technology) is not supported on this platform.

Ethernet Option for Connections to the Exadata Database Machine

The preferred method of connecting Oracle Big Data Appliance and Oracle Exadata Database for any purpose is through InfiniBand. Previous releases of Oracle Big Data SQL have required InfiniBand for these connections. In Release 3.1, Ethernet networking between the Exadata Database Machine and Oracle Big Data Appliance is now supported. This enables you to use Oracle Big Data SQL with these two Engineered Systems in environments where InfiniBand is not feasible, such as when the two systems are geographically distant from each other.

Release 3.1 also enables Ethernet connections between commodity Hadoop systems and the Oracle Exadata Database Machine.

Oracle Big Data SQL connectivity between commodity Hadoop systems and commodity Oracle Database servers has been Ethernet-based throughout previous releases.

Note that Ethernet connections between Oracle Big Data Appliance (or commodity Hadoop systems) and Oracle SPARC SuperCluster are not supported at this time.

Simplified Deployment on Oracle Database Servers – Oracle Grid Infrastructure is now Optional

In previous releases of Oracle Big Data SQL, Oracle Grid Infrastructure is a prerequisite of the installation for all Oracle Database servers, including standalone servers that are not part of an Oracle RAC system. In Release 3.1, you have the option to install Oracle Big Data SQL on servers where Oracle Grid Infrastructure is not present. Note that in these cases, the installer makes some configuration file changes that require a restart of Oracle Database.

Unified Platform Support in the Oracle Big Data SQL Installer

Previous Oracle Big Data SQL releases included two separate installation procedures – one for Oracle Engineered Systems and another for commodity servers. In Release 3.1, you use the same installation process for both Oracle and non-Oracle platforms. This is also true for maintenance. For all of the supported Hadoop/Oracle Database combinations there is a uniform set of steps to update the Oracle Big Data SQL configuration when there are changes to the Hadoop cluster or Oracle Database server.

New Features to Simplify ILM – Oracle Database Tablespaces in HDFS (With Smart Scan Technology)

Oracle Database ILM (Information Lifecycle Management) can now be extended to use Hadoop to store read-only Oracle Database tablespaces. When you move tablespaces from Oracle Database to HDFS, the tables, partitions, and data retain their original Oracle Database internal format, remain accessible to queries, and support the full range of Oracle Database performance optimizations and security features, including the following:

- Smart Scan for HDFS, which enables off-load of query processing to Oracle Big Data SQL on the Hadoop cluster. Smart Scan also provides filtering of query results in Hadoop prior to the return of the data to Oracle Database. In most circumstances, this can be a significant performance optimization. Indexing, Hybrid Columnar Compression, Partition Pruning, and Oracle Database In-Memory are also supported.
- Oracle Advanced Security Option (ASO) Transparent Encryption and Data Redaction.

Tablespaces stored in HDFS are read-only, therefore this storage is best suited to data archiving.

See Section 3.2 in the *Oracle Big Data SQL User's Guide* for details.

Enhancements in Oracle Shell for Hadoop Loaders 1.2

Oracle Shell for Hadoop Loaders (OHSH) is an intuitive command line tool for data migration. You can set up resources to connect to Hive, HDFS, or Oracle Database, and then access each of these data sources through the uniform OHSH interface. OHSH is one of the ways to use Copy to Hadoop. Copy to Hadoop users can download OHSH from OTN.

OHSH 1.2 includes the following changes:

- On-disk logging of load operations in the `$HOME/.ohsh` shadow directory.
- The ability to minimize output when doing load commands. (See the help command for `set outputlevel`.)
- Loading Hive tables from Oracle tables not living in the `oracle` user's schema.
- Wallet and TNS usage by OHSH relies on the setting of environmental variables WALLET_LOCATION and TNS_ADMIN. The `set tnsadmin` and `set walletlocation` commands are no longer supported.

In addition, you no longer set `HIVE0_URL` to the fully-qualified URL of remote HiveServer2 in order to create a `%hive0` resource. In OHSH 1.2, set the environmental variable `HS2_HOST_PORT` in `bin/ohsh`, which is the `<hostname>:<port>` pair of HiveServer2.

Enhancements to Copy To Hadoop

The new method `directcopy` is added to Copy to Hadoop.

This is a direct, single-step method of copying data from Oracle Database to HDFS. See *Using Copy to Hadoop to do Direct Copies* in the *Oracle Big Data SQL User's Guide* for more information.

Granting Access – Users Now Require the BDSQL_USER Role

In prior Oracle Big Data SQL releases, all users are granted Big Data SQL access implicitly. Release 3.1 introduces the `BDSQL_USER` role. Users requiring Oracle Big Data SQL access must be granted this role explicitly.

You must also now grant read privileges on the BigDataSQL configuration directory object.

For example, to grant access to `user1`:

```
SQL> grant BDSQL_USER to user1;
SQL> grant read on directory ORACLE_BIGDATA_CONFIG to user1;
```

Installation Instructions Moved to Oracle Big Data SQL Installation Guide

The *Oracle Big Data SQL Installation Guide* provides instructions how to install and uninstall the software. In releases prior to Oracle Big Data Appliance 3.1, installation instructions are in the user’s guide.

1 Introduction

Oracle Big Data SQL 3.1 can connect Oracle Database to the Hadoop environment on Oracle Big Data Appliance, or on other Hadoop systems based on CDH (Cloudera's Distribution including Apache Hadoop), HDP (Hortonworks Data Platform).

In previous releases of Big Data SQL, the installations on different combinations of Hadoop server and Oracle Database server included some differences. In the current release, there is a common installation process for all supported Hadoop systems and Oracle Database systems.

> **Note:**
>
> Oracle Big Data SQL 3.0.1 is the prior release and is included in the Oracle Big Data Appliance 4.7 installation bundle. Oracle Big Data SQL 3.0.1 can be enabled either in the Oracle Big Data Appliance 4.3 or higher Mammoth installation or post installation. If you choose to install Oracle Big Data SQL 3.0.1, do not use the instructions in this version of the guide. In that case, refer to the Installing On Oracle Big Data Appliance and the Oracle Exadata Database Machine in the *Oracle Big Data SQL User’s Guide for Release 3.0.1* .

1.1 Oracle Big Data SQL Master Compatibility Matrix

See the *Oracle Big Data SQL Master Compatibility Matrix* (Doc ID 2119369.1) in My Oracle Support for up-to-date information on Big Data SQL compatibility with the following:

- Oracle Engineered Systems.
- Other systems.
- Linux OS distributions and versions.
- Hadoop distributions.
- Oracle Database releases, including required patches.

1.2 Installation Overview

The Oracle Big Data SQL software must be installed on all Hadoop cluster nodes and all Oracle Database compute nodes.

Important: About Service Restarts

On the Hadoop-side installation, the following restarts may occur.

- Cloudera Configuration Manager (or Ambari) may be restarted. This in itself does not interrupt any services.

- Hive, YARN , and any other services that have a dependency on Hive or YARN (such as Impala) are restarted.

  The Hive libraries parameter is updated in order to include Oracle Big Data SQL JARs. On Cloudera installations, if the YARN Resource Manager is enabled, then it is restarted in order to set cgroup memory limit for Oracle Big Data SQL and the other Hadoop services. On Oracle Big Data Appliance, the YARN Resource Manager is always enabled and therefore always restarted.

  > **Note:**
  >
  > The Oracle Big Data SQL installation script includes this message in its output:
  >
  > ```
  > BigDataSQL: Restarting cluster...
  > ```
  >
  > The restart is limited to “stale” components of the cluster, which can include Hive, YARN, and their dependent services.

On the Oracle Database server(s), the installation may require a database and/or Oracle Grid infrastructure restart in environments where updates are required to cell settings on the Grid nodes. See Potential Requirement to Restart Grid Infrastructure for details.

How Long Does It Take?

The installation on the Hadoop cluster may take approximately 30 minutes. The generation of the installation bundle for the database side requires a download Hadoop and Hive clients and other software. This may take 20 additional minutes, depending on Internet download speed. The average installation time for the database side can be estimated as follows:

- 15 minutes for a single node database if a restart is not required. If a restart is required, the time will vary, depending on the size of the database.
- On a RAC database, multiply the factors above by the number of nodes.
- If an Oracle Grid restart is required, factor that in as well.

Outline of the Installation Steps

This is the sequence of tasks for installing Oracle Big Data SQL.

1. Before you start:
   - Check the *Oracle Big Data SQL Master Compatibility Matrix* (Document 2119369.1 in My Oracle Support) for general platform compatibility.
   - Check Installation Prerequisites in this guide for required software.
   - Login credentials that you will need:
     - `root` credentials for both the Hadoop cluster and all Oracle Grid nodes.

       On the grid nodes you have the option of using passwordless SSH with the root user instead.
     - `oracle` Linux user (or other, if the database owner is not `oracle`)
     - The Oracle Grid user (if this is not the same as the database owner).

   - The Hadoop configuration management service (CM or Amabari) admin password.

2. On the cluster management server (where CM or Ambari is running), download the software from Oracle. (See Downloading Oracle Big Data SQL.)
3. On the cluster management server, check to see if Python 2.7 is installed, as described in Installation Prerequisites.
4. Perform the Hadoop-side installation described in Installing the Hadoop Side of Oracle Big Data SQL.

   The Hadoop-side phase of the installation does the following:

   - Deploys Oracle Big Data SQL binaries along the cluster.
   - Configures Linux and network settings for the service on each cluster node.
   - Configures the BDS service on the cluster management server.
   - Acquires information needed to configure Oracle Database connections to the cluster.
5. On the Hadoop cluster management server, run the scripts to create the bundle that installs Oracle Big Data SQL on the Oracle Database side. (Described in Creating the Database-Side Installation Bundle.)
6. Perform the Oracle Database server-side installation. (See Installing the Oracle Database Side of Oracle Big Data SQL.)

   In this phase of the installation, you deploy the database-side installation bundle that you generated to the database nodes. Extract it from the zip file and run it.

   The database-side installer does the following:

   - Copies the Oracle Big Data SQL binaries to the database node.
   - Configures network settings for the service.
   - Inserts cluster metadata into Oracle Database.

   > **Important:**
   >
   > Be sure to install the bundle on each compute node. The Hadoop-side installation automatically propagates the Oracle Big Data SQL software to each DataNode of the Hadoop cluster. However, the database-side installation does not work this way. You must copy the software to each database compute node and install it directly.
   >
   > In Oracle Grid environments, if cell settings need to be updated then the installer will alert you if a grid restart will be needed. (Be sure to have the grid password because if the grid restart is required, then the installation will not complete without these credentials.)

This completes the basic installation.

> **Tip:**
>
> See How to do a Quick Test in the *Oracle Big Data SQL User's Guide* for some simple functionality tests.

Installation Log Files

On the Hadoop cluster side:

```
/var/log/bigdatasql
/var/log/oracle
```

On the Oracle Database side:

```
$ORACLE_HOME/install/bds* (This is a set of files, not a directory)
$ORACLE_HOME/bigdatasql/logs
/var/log/bigdatasql
```

> **Tip:**
>
> If you make a support request, create a zip archive that includes all of these logs and include it in your email to Oracle Support.

> **See Also:**
>
> - Read about measures you can take to secure the installation. (See Securing Big Data SQL.)
> - Learn how to modify the Oracle Big Data SQL configuration when changes occur on the Hadoop cluster and in the Oracle Database installation. (See Adapting to Hadoop Cluster and Oracle Database Changes.)
> - If you have used Copy to Hadoop in earlier Oracle Big Data SQL releases, learn how Oracle Shell for Hadoop Loaders can simplify Copy to Hadoop tasks. (See Additional Tools Installed.)

1.3 Downloading Oracle Big Data SQL

You can download Oracle Big Data SQL from the Oracle Software Delivery Cloud

1. On the cluster management server, create a new directory or choose an existing one to be the installation source directory.
2. Log in to the Oracle Software Delivery Cloud.
3. Search for Oracle Big Data SQL.
4. Select `Oracle Big Data SQL 3.1.0.0.0 for Linux x86-64`.
5. Read and agree to the Oracle Standard Terms and Restrictions.
6. From the list, select the zip file that is appropriate for your Hadoop system:
   - `Oracle Big Data SQL 3.1.0 installer for Hortonworks Data Platform`
   - `Oracle Big Data SQL 3.1.0 installer for Cloudera Enterprise`

   Each zip file contains the complete installation package.

7. Download the appropriate package for your Hadoop system. Currently packages are provided for Hortonworks HDP and for Cloudera Enterprise systems (including CDH-based Oracle Big Data Appliance and other CDH systems identified in the Oracle Big Data SQL Master Compatibility Matrix).

   If your Hadoop system is a supported version of Oracle Big Data Appliance, download the installer for Cloudera Enterprise.

Your product bundle should include the significant files listed the table below, as well as other supporting files.

Table 1-1 Oracle Big Data SQL Product Bundle Inventory

File or Directory	Description
`BDSSetup/setup-bds`	Cluster-side installation script
`BDSSetup/bds-config.json`	Configuration template file
`BDSSetup/deployment_manager/*`	Deployment manager package
`BDSSetup/db/hadoop-*-nativelib-*.tar.gz`	Hadoop native libraries
`BDSSetup/db/bds-database-create-bundle.sh`	Script to create the database-side installation bundle
`BDSSetup/db/database-install.zip`	Database-side pre-bundle

1.4 Upgrading From a Prior Release of Oracle Big Data SQL

As a prerequisite to upgrading to Release 3.1, you must remove earlier installations.

1. First, completely uninstall the earlier version of Oracle Big Data SQL from the Hadoop system as well as the Oracle Database system.

   See Chapter 4 in this guide for instructions.

   The methods for uninstalling will differ, depending on whether the installation is on Oracle Engineered Systems or commodity servers.

2. Then, proceed with the Release 3.1 installation on the Hadoop system and the Oracle Database system as described in Chapter 2 and Chapter 3 of this guide.

2 Installing the Hadoop Side of Oracle Big Data SQL

Oracle Big Data SQL is deployed using the services provides by the cluster management server. The installer program uses the management server API to register the service and start the deployment task. From there, the management server controls the process and deploys the software to the nodes of the cluster and installs it.

After installing Big Data SQL on the cluster management server, use the tools provided in the bundle to generate an installation package for the database server side.

> **Important Notes for Oracle Big Data Appliance Installations Prior to Release 4.8:**
>
> - Oracle Big Data SQL 3.1 is decoupled from the `bdacli` utility on Oracle Big Data Appliance systems prior to Release 4.8. On these systems, you cannot use the following `bdacli` commands with Release 3.1 on Oracle Big Data Appliance 4.7 and earlier:
>
>   ```
>   bdacli {enable|disable} big_data_sql
>   bdacli getinfo cluster_big_data_sql_enabled
>   bdacli {start | stop | restart | status}  {big_data_sql_cluster |
>   big_data_sql_server node_name}
>   ```
>
> - The bdacli commands listed above *do* work for previous versions of Oracle Big Data SQL installed on the Oracle Big Data Appliance releases that they support.
>
> On all supported Hadoop systems (listed in the *Oracle Big Data SQL Master Compatibility Matrix* (Doc ID 2119369.1) in My Oracle Support) ,you can use the `setup-bds` script as described in this guide to install, extend, reconfigure, and uninstall Oracle Big Data SQL 3.1 on a Hadoop cluster. You can also use the cluster management server interface to stop or stop Oracle Big Data SQL processes.
>
> On Oracle Big Data Appliance 4.8 and greater, because Oracle Big Data SQL integration with `bdacli` has been restored, you have the additional option of using `bdacli` to install and uninstall (enable and disable) Oracle Big Data SQL. You also have access to all other `bdacli` commands for administering Oracle Big Data SQL.

See Also:

olink:BIGOG-GUID-685D1923-EC2A-42B6-8D97-1DFB8239D57C in the Oracle Big Data Appliance Owner’s Guide.

2.1 Installation Prerequisites

The following active services, installed packages, and available system tools are prerequisites to the Oracle Big Data SQL installation.

Platform requirements, such as supported Linux distributions and versions, as well as supported Oracle Database releases and required patches and are not listed here. See the Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1) in My Oracle Support for this information.

Services Running

These Apache Hadoop services must be running on the cluster.

- HDFS
- YARN
- Hive

You do not need to take any extra steps to ensure that the correct HDFS and Hive clients URLs are specified in the database-side installation bundle.

Important:

The Apache Hadoop services listed above must be installed as parcels on Cloudera CDH and as stacks on Hortonworks HDP. Installation of these services via RPM is not supported in either case.

About HBase:

HBase is not a prerequisite for the installation of Oracle Big SQL on the Hadoop cluster. However, the installation on the Oracle Database nodes does require an HBase client. If HBase is present on the cluster during the Hadoop-side installation, then no action on your part is required. In this case, when you create the package that is installed on the Oracle Database nodes, the installation package builder automatically includes the URL to the download site of the compatible HBase client JAR.

If HBase is not installed on the Hadoop cluster, then when you create the database-side installation package, use the `--hbase-client-ws` parameter to add the URL to the the installation package. If an HBase service is installed on the Hadoop cluster, then the URL should point to the download site for a compatible client. When you install the database side of Oracle Big Data SQL, the installer will download the HBase client from the URL provided.

Packages

The following packages must be pre-installed on all Hadoop cluster nodes before installing Oracle Big Data SQL.

- Oracle JDK version 1.7 or later
- dmidecode
- net-snmp, net-snmp-utils
- perl

  PERL LibXML – 1.7.0 or higher, e.g. perl-XML-LibXML-1.70-5.el6.x86_64.rpm

  perl-libwww-perl, perl-libxml-perl, perl-Time-HiRes, perl-libs, perl-XML-SAX, perl-Env

The Java JDK is available for download on the Oracle Technology Network.

The `yum` utility is the recommended method for installing these packages:

```
yum -y install dmidecode
yum -y install net-snmp net-snmp-utils
yum -y install perl perl-libs
yum -y install perl-Time-HiRes perl-libwww-perl
yum -y install perl-libxml-perl perl-XML-LibXML perl-XML-SAX
yum -y install perl-Env
yum -y localinstall <JDK RPM that you downloaded>
```

System Tools

- curl
- gcc
- libaio
- rpm
- scp
- tar
- unzip
- wget
- yum
- zip

The libaio libraries must be installed on each Hadoop cluster node:

```
yum install -y libaio gcc
```

Environment Settings

The following environment settings are required prior to the installation.

- NTP enabled
- Ensure that `/usr/java/default` exists and is linked to the appropriate Java version (if $JAVA_HOME does not exist).
- The path to the Java binaries must exist in `/usr/java/latest`.

- The installation process requires Internet access in order to download some packages from Cloudera or Hortonworks sites. If a proxy is needed for this access, ensure that the following Linux environment variables are properly set:
  - `http_proxy` and `https_proxy`
  - `no_proxy`

    Set `no_proxy` to include the following: "`localhost,127.0.0.1,`*`<Comma–separated list of the hostnames in the cluster (in FQDN format).>`*".
- On Cloudera CDH, clear any proxy settings in Cloudera Manager administration before running the installation. You can restore them after running the script that creates the database-side installation bundle (`bds-database-create-bundle.sh`).

Python 2.7 (for the Oracle Big Data SQL Installer)

The Oracle Big Data SQL installer requires Python 2.7 locally on the node where you run the installer. This should be the same node where the cluster management service (CM or Ambari) is running.

If any version of Python 2.7.x is already installed, you can use it to run the Oracle Big Data installer.

If an earlier version than Python 2.7.0 is already installed on the cluster management server and you need to avoid overwriting this existing installation, you can add Python 2.7.x as a secondary installation.

> **NOT_SUPPORTED:**
>
> On Oracle Big Data Appliance do not overwrite or update the pre-installed Python release. This restriction may also apply other supported Hadoop platforms. Consult the documentation for the CDH or HDP platform you are using.

On Oracle Linux 5, add Python 2.7 as a secondary installation. On Oracle Linux 6, both Python 2.6 and 2.7 are pre-installed and you should use the provided version of Python 2.7 for the installer. Check whether the default interpreter is Python 2.6 or 2.7. To run the Oracle Big Data SQL installer, you may need to invoke Python 2.7 explicitly. On Oracle Big Data Appliance, SCL is installed so you can use to enable version 2.7 for the shell as in this example:

```
# scl enable python27 "./setup-bds install bds-config.json"
```

Below is a procedure for adding the Python 2.7.5 as a secondary installation.

Tip:

If you manually install Python, first ensure that the `openssl-devel` package is installed:

```
# yum install -y openssl-devel
```

If you create a secondary installation of Python, it is strongly recommended that you apply Python update regularly to include new security fixes. Do not update the mammoth-installed Python unless directed to do so by Oracle.

```
# pyversion=2.7.5
# cd /tmp/
# mkdir py_install
# cd py_install
# wget https://www.python.org/static/files/pubkeys.txt
# gpg --import pubkeys.txt
# wget https://www.python.org/ftp/python/$pyversion/Python-$pyversion.tgz.asc
# wget https://www.python.org/ftp/python/$pyversion/Python-$pyversion.tgz
# gpg --verify Python-$pyversion.tgz.asc Python-$pyversion.tgz
# tar xfzv Python-$pyversion.tgz
# cd Python-$pyversion
# ./configure --prefix=/usr/local/python/2.7
# make
# mkdir -p /usr/local/python/2.7
# make install
# export PATH=/usr/local/python/2.7/bin:$PATH
```

If Oracle Big Data SQL is Already Installed

If Oracle Big Data SQL is already installed, please read Upgrading From a Prior Release of Oracle Big Data SQL before proceeding.

Important:

If Oracle Big Data SQL 3.0.1 or earlier is already enabled on the Hadoop cluster, it must be disabled or removed before installing Release 3.1.

For Release 3.0.1 or earlier on Oracle Big Data Appliance, run `bdacli disable big_data_sql`. This will disable Oracle Big Data SQL on all nodes of the cluster.

The uninstall on other Hadoop platforms (HDP or non-Oracle CDH-based systems) differs for Oracle Big Data SQL 3.0.1 and 3.0:

- For Release 3.0.1, run `setup-bds` from the original installation with the `--uninstall` parameter, as in `./setup-bds --uninstall bds-config.json`.
- For Release 3.0 there is no programmatic uninstall. You can find the manual procedure for uninstalling Release 3.0 in the Oracle Big Data SQL 3.0 User's Guide.

  Pre-3.0 releases of Oracle Big Data SQL were not supported on systems other than Oracle Big Data Appliance.

2.1.1 Checking for Prerequisites on the Hadoop DataNodes

You can check the DataNodes of the cluster for Oracle Big Data SQL installation prerequisites as follows.

As `root`, run the following checks on each DataNode:

```
# yum -y install dmidecode
# yum -y install net-snmp net-snmp-utils
# yum -y install perl perl-libs
# yum -y install perl-Time-HiRes perl-libwww-perl
# yum -y install perl-libxml-perl perl-XML-LibXML perl-XML-SAX
# ls -l /usr/java
```

2.2 Installing on the Hadoop Cluster Management Server

The first step of the Oracle Big Data SQL installation is to run the installer on the Hadoop cluster management server (where Cloudera Manager runs on a CDH system or Ambari runs on an HDP system). As post-installation task on the management server, you then run the script that prepares the installation bundle for the database server.

On the Hadoop side, you manually install the software on the cluster management server only. The installer uses CDH or Ambari to analyze the cluster configuration and automatically deploys and installs Oracle Big Data SQL on all nodes where it is required.

> **About Installer Security**
>
> - Passwords for Ambari and Cloudera Manager are not be passed in on the command line and are not be saved in any persistent files (including log or trace files) during the installation or after the installation is complete.
> - No temporary or persistent world-writable files are created.
> - No setuid or setgid files are used.
> - The installer works with hardened Oracle Database environments as well as hardened CDH and HDP clusters as described in the Cloudera CDH and Hortonworks HDP security documentation.

This is a summary of the three tasks to perform on the cluster manager server. Details follow this summary.

> **Temporary Workaround may be Required:**
>
> The `setup-bds` installation script sets some Hive auxiliary parameters, which may overwrite existing custom settings. For example, to enable Hive operation logging for the installation, the script sets `hive.server2.logging.operation.enabled=true`. After `setup-bds` has finished (and before you run the next script in the process, `bds-database-create-bundle.sh`), check your Hive auxiliary parameter settings and, if needed, restore any that may have been overwritten. This workaround will be unnecessary in a subsequent release of Oracle Big Data SQL.
>
> In an Oracle Big Data SQL uninstall (which also uses `setup-bds`) the uninstall completely removes the `hive.server2.logging.operation.enabled parameter`, which effectively sets it to `true`, the default value.

- Extract the files from `BigDataSQL-<`*`Hadoop_distribution`*`>-<`*`version`*`>.zip` archive. Then, configure `bds-config.json` and run `setup-bds` with the `install` argument. This installs Oracle Big Data SQL on the cluster management server as well as on all Hadoop DataNodes in the cluster.
- Run the database bundle creation script, `bds-database-create-bundle` . This generates the database bundle file that you will run on the Oracle Database server in order to install Oracle Big Data SQL on the Oracle Database server.
- Check the parameters in the database bundle file and adjust as needed.

After you have checked and (if necessary) edited the database bundle file, copy it over to the Oracle Database server and run it as described in Installing on the Oracle Database Server

Install Oracle Big Data SQL on the Cluster Management Server

> **Important: Patch 25796576 Required For Some Systems:**
>
> Install this patch if your system meets the following criteria:
>
> - For systems other than Oracle Big Data Appliance: using CDH, and running on Oracle Linux 6, Oracle Linux 7, Red Hat 6 or Red Hat 7.
> - For Oracle Big Data Appliance: when Oracle Big Data SQL is installed with the installer described in this document (in other words, when it is not installed by Mammoth or the bdacli utility).
>
> If your system meets these criteria, download Patch 25796576 to the same node and same directory where you download the zip file in the steps below. Install the patch after unpacking the zip file.
> For instructions on installing the patch and more detail on when the patch is required, see the following document in My Oracle Support: *One-Off Patch 25796576 On An Oracle Big Data Appliance CDH Cluster OL6 with Big Data SQL 3.1 (Doc ID 2269180.1)*. This does not apply to Oracle Linux 5.

To install Big Data SQL on the cluster management server:

1. Copy the appropriate zip file (`BigDataSQL-<Hadoop_distribution>-<version>.zip`) to a temporary location on the cluster management server (the node where Cloudera Manager or Ambari is running).
2. Unzip file.
    1. For CDH systems, run Patch 25796576 at this point and then proceed with the Oracle Big Data SQL installation.
3. Change directories to `BDSSetup` .
4. Edit the configuration file.

    Table 2–4 below describes the use of each configuration parameter.

    - For CDH, edit `bds-config.json` , as shown in this example. Any unused port will work as the web server port.

      ```
      {
      "cluster": {
         "name": "cluster1",
         "display_name": "Cluster 1"
      },
      "database":{
         "ip": "10.11.12.13/14"
      },
      "memory": {
         "min_hard_limit": 8192
      },
      "webserver": {
         "port": 80
      }
      }
      ```

    - For HDP, edit `bds-config.json` as in this example. Notice that the HDP configuration file does not include the `display_name` or `min_hard_limit` parameters.

      ```
      {
      "cluster": {
         "name": "cluster1",
      },
      "database":{
         "ip": "10.11.12.13/14"
      },
      "webserver": {
         "port": 80
      }
      }
      ```

    `DATABASE_IP` must be the correct network interface address for the database node where you will perform the installation. You can confirm this by running `/sbin/ip -o -f inet addr show` on the database node.

    > **Note:**
    >
    > The next step requires the cluster administrator user ID and password.

5. In the BDSSetup directory, become `root` and run `setup-bds`. Pass it the `install` parameter and the configuration file name (`bds-config.json`) as arguments. Note that the cluster management service is restarted in this process.

```
[root@myclusteradminserver:BDSSetup] # ./setup-bds install bds-config.json
```

The script prompts for the cluster management service administrator credentials and then installs Oracle Big Data SQL on the management server and the cluster nodes. The script output terminates with the following message if the installation completed without error.

```
BigDataSQL: INSTALL workflow completed.
```

> **See Also:**
>
> This is a condensed example. An example of the complete standard output from a successful installation is provided in Oracle Big Data SQL Install/Uninstall/Reconfigure Examples.

Parameters in bds-config.json

The table below explains edits you need to make to `bds-config.json` for CDH and HDP platforms.

Table 2-1 Configuration Parameters in bds-config.json

Configuration Parameter	Description	Applies To	Required/ Optional
cluster:{*name*}	The name of the cluster. See the description of `display_name` below for behavior that applies to both of these parameters.	CDH, HDP	Required for CDH. Optional for HDP.
cluster: {*display_name*}	The `display_name` is an optional identifier for locating the target cluster in Cloudera Manager. You can use it as an alternative to the required name parameter. Oracle Big Data SQL attempts to use it as a fallback if *cluster*:{*name*} cannot be validated.	CDH only	Optional
database:{*ip*}	The IP address of the Oracle Database server that will make connection requests. This must be configured on one interface on the database node. The address must include the prefix length (as in 100.112.10.36/24). Although only one IP address is specified in the configuration file, it is possible to install the database-side software on multiple database servers (as in a RAC environment) by using a command line parameter to override `ip` at installation time. (See the description of `--ip-cell` in Table 3-1.)	CDH, HDP	Required
memory: {*min_hard_limit*}	The minimum amount of memory required for Oracle Big Data SQL.	CDH only	Optional

Table 2-1 (Cont.) Configuration Parameters in bds-config.json

Configuration Parameter	Description	Applies To	Required/ Optional
webserver:{*port*}	Port for the temporary repository used for deploying tasks and gather responses from the DataNodes during installation. This can be any port that does not conflict with current cluster operation.	CDH, HDP	Required
api:{*port*}	Port for the CDH or Ambari REST API in cases where this is different from the default port.	CDH, HDP	Optional

Operations Performed by setup-bds

The table below lists the full set of operations performed by `setup-bds` .

The syntax for all options is:

```
# ./setup-bds <option> bds-config.json
```

For example:

```
# ./setup-bds install bds-config.json
```

Table 2-2 Command Line Options for setup-bds.sh

setup-bds Option	Use
`install`	Install the Oracle Big Data SQL software on the cluster management server.
`extend`	Extend Oracle Big Data SQL to any new DataNodes and update the cells inventory if the cluster has grown since the last Oracle Big data SQL installation.
`remove`	Remove Oracle Big Data SQL components from any nodes where the DataNode service no longer exists. This must be done if a DataNode service is moved or removed. A possible scenario where this would be necessary is if a DataNode service has been moved another node for better cluster load balancing.
`reconfigure`	Modify the current installation by applying changes you have made to the configuration in `bds-config.json`. Note that if you run `setup-bds reconfigure bds-config.json` to reconfigure the Hadoop side Oracle Big Data SQL, a corresponding reconfiguration is required on the Oracle Database side. The two sides cannot communicate if the configurations do not match. In this case you must also regenerate the database-side bundle files to incorporate the changes, and then redeploy the bundle on all database servers where it was previously installed. For reconfigurations, a lightweight database bundle is provided so that the changes can be deployed relatively quickly. This bundle does not need and does not include the tarballs that are required by the initial installation.

Table 2-2 (Cont.) Command Line Options for setup-bds.sh

setup-bds Option	Use
`uninstall`	Uninstall Oracle Big Data SQL from the Hadoop cluster management server.

> **See Also:**
>
> Uninstalling Oracle Big Data SQL.

2.3 Creating the Database-Side Installation Bundle

After installing Oracle Big Data SQL on the cluster management server, run the script `BDSSetup/db/bds-database-create-bundle.sh`. This script creates the corresponding Oracle Big Data SQL installation bundle for any Oracle Database servers that will query data on the Hadoop system.

In addition to running `bds-database-create-bundle` at installation time, you also need to run it when you have made configuration changes to the cluster management server that must be communicated to the database server, such as:

- A change cluster configuration, such as from unsecure to secure HTTP.
- Migration of the Hive Metastore from one node to another.
- Change of the CM or Ambari port on the cluster management server.
- Change of IP address used for installation on the database side.

External Resources Required for the Database-Side Bundle

`bds-database-create-bundle` attempts to download the following external resources.

- Hadoop and Hive client tarballs from Cloudera or Hortonworks repository web site.
- Configuration files for Yarn and Hive from the cluster management server, via Cloudera Manager (for the CDH versions) or Ambari (for the HDP versions).
- For HDP only, HDFS and MapReduce configuration files from Ambari.

If some of these resources are not accessible from the management server, you can add them manually. You can also use the command line switches described in Table 2-3 to manually turn off selected resource downloads so that these resources are not added to the bundle.

Running bds-database-create-bundle

Run this script as `root`. Note that you are prompted for the cluster management service administrator credentials.

1. Change directories to `BDSSetup/db`.

2. Run the BDS database bundle creation script. See Table 2-3 below for optional parameters that you can pass to the script in order to override any of the default settings.

   ```
   [root@myclusteradminserver: db] # ./bds-database-create-bundle.sh <optional
   parameters>
   ```

   This message is returned if the operation is successful:

   ```
         bds-database-create-bundle: database bundle creation script completed all
   steps
   ```

The `bds-database-create-bundle` script generates two different database bundles in the BDSSetup directory (not in the BDSSetup/db directory):

- `bds-database-install.zip`

  Deploy this bundle to the database servers for the initial installation of Oracle Big Data SQL (or if a full re-installation is required for other reasons). It contains all of the files needed to install the software, including the resources that were downloaded from Cloudera or Hortonworks (tarballs and configuration files).

- `bds-database-install-config.zip`

  Deploy this bundle to the database servers instead of `bds-database-install.zip` when you change the existing Oracle Big Data SQL configuration on the cluster management server. This package makes the corresponding changes to the database-side configuration so that the two sides (Hadoop and Oracle Database) are aligned. A reconfiguration of Oracle Big Data SQL does not require the external resources needed in the full installation (such as the client tarballs). Therefore this bundle is smaller and can be deployed faster than the full installation bundle.

The database bundle file includes a number of parameters. When you run `bds-database-create-bundle.sh`, you can use the switches in the table below to override any of these parameters as necessary. Any URLs specified must be accessible from the cluster management server at the time you run `bds-database-create-bundle.sh`.

Table 2-3 Command Line Switches for bds-database-create-bundle.sh

Parameter	Value
`--hadoop-client-ws`	Specifies an URL for the Hadoop client tarball download.
`--no-hadoop-client-ws`	Exclude this download.
`--hive-client-ws`	Specifies an URL for the Hive client tarball download.
`--no-hive-client-ws`	Exclude this download.
`--yarn-conf-ws`	Specifies an URL for the YARN configuration zip file download.
`--no-yarn-conf-ws`	Exclude this download.
`--hive-conf-ws`	Specifies an URL for the Hive configuration zip file download.
`--no-hive-conf-ws`	Exclude this download.
`--ignore-missing-files`	Create the bundle file even if some files are missing.

Table 2-3 (Cont.) Command Line Switches for bds-database-create-bundle.sh

Parameter	Value
`--jdk-tar-path`	Override the default JDK path. Do not specify a relative path, use `--jdk-tar-path=<`*`jdk tarfile absolute path`*`>`.
`--clean-previous`	Deletes previous bundle files and directories from `bds-database-install/` . If cluster management server the cluster settings have changed (for example, because of an extension, service node migration, or adding/ removing security) then it necessary to redo the installation on the database server. As part of this re-installation, you must run `--clean-previous` to purge the cluster information left the database server side from the previous installation.
`--script-only`	This is useful for re-installations on the database side when there are no cluster configuration changes to communicate to the database server and where there is no need to refresh files (such as client tarballs) on the database side. With this switch, `bds-database-create-bundle.sh` generates a zip file that contains only the database installation script and does not bundle in other components, such as the tarballs. If these already exist on the database server, you can use `--script-only` to bypass the downloading and packaging of these large files. Do not include `--clean-previous` in this case.
`--hbase-client-ws`	This is parameter is required only if HBase is not installed in the Hadoop cluster. It specifies the URL where the HBase tarball can be downloaded from the Cloudera or Ambari website. The URL should point to the specific version of HBase that is supported by the CDH or HDP installation you are using. If HBase is not installed, include this parameter when you run `bds-database-create-bundle.sh`. `$ bds-database-create-bundle.sh --hbase-client-ws <URL>` If HBase is not installed and you do not supply this parameter when you run `bds-database-create-bundle.sh`, then you will be prompted for the URL.
`--hdfs-conf-ws`	Specify an URL for the HDFS configuration zip file download.
`--no-hdfs-conf-ws`	Exclude this download (HDP only).
`--mapreduce-conf-ws`	Specify an URL for the MapReduce configuration zip file download (HDP only).
`--no-mapreduce-conf-ws`	Exclude this download (HDP only).

Table 2-3 (Cont.) Command Line Switches for bds-database-create-bundle.sh

Parameter	Value
`--reconfigure`	Creates the bundles with the existing files, but force downloading new configuration.

Manually Adding Resources if Download Sites are not Accessible to the BDS Database Bundle Creation Script

If one or more of the default download sites is inaccessible from the cluster management server, there are two ways around this problem:

- Download the files from another server first and then provide `bds-database-create-bundle.sh` with the alternate path as an argument. For example:

  ```
  $ ./bds-database-create-bundle.sh --yarn-conf-ws='http://nodexample:1234/config/
  yarn'
  ```

- Because the script will first search locally in `/bds-database-install` for resources, you can download the files to another server, move the files into `/bds-database-install` on the cluster management server and then run the bundle creation script with no additional argument. For example:

  ```
  $ cp hadoop-xxxx.tar.gz bds-database-install/
  $ cp hive-xxxx.tar.gz bds-database-install/
  $ cp yarn-conf.zip bds-database-install/
  $ cp hive-conf.zip bds-database-install/
  $ cd db
  $ ./bds-database-create-bundle.sh
  ```

Copying the Database Bundle to the Oracle Database Server

Use `scp` to copy the database bundle you created to the Oracle Database server. In the example below, `dbnode` is the database server. The Linux account and target directory here are arbitrary. Use any account authorized to `scp` to the specified path.

For a first-time installation of the current release of Oracle Big Data SQL, copy `bds-database-install.zip` (the full installation bundle) to each database node.

```
$ scp bds-database-install.zip oracle@dbnode:/home/oracle
```

If you are updating the configuration in the existing Oracle Big Data SQL installation on the database servers, copy the smaller configuration update bundle to the database nodes (`bds-database-install-config.zip`) instead of (`bds-database-install.zip`).

```
$ scp bds-database-install-config.zip oracle@dbnode:/home/oracle
```

The next step is to log on to the Oracle Database server and install the bundle.

3

Installing the Oracle Database Side of Oracle Big Data SQL

3.1 Installing on the Oracle Database Server

Oracle Big Data SQL must be installed on both the Hadoop cluster management server and the Oracle Database server nodes. This section describes the full installation of Oracle Big Data SQL 3.1 on Oracle Database.

If Oracle Big Data SQL is already installed on a database node and you are updating the existing configuration, see Reconfiguring an Existing Oracle Big Data SQL Installation on Database Nodes.

> **Important:**
>
> - For multi-node databases (such as Oracle RAC systems), **you must repeat this installation on every node of the database**. If this is not done, you will see RPC connection errors when the Oracle Big Data SQL service is started. You can reuse the same installation bundle on each database node by using a parameter to pass in the correct network interface address when you run the installation script. This is described in the section, *If You Need to Change the Configured Database_IP Address*, below.
> - On Oracle Big Appliance, Oracle Big Data SQL can connect to Oracle Database on either Exadata Database Machine or an non-Exadata server, but not to both at the same time.

Prerequisites for Installing on an Oracle Database Server

Required Software

See the *Oracle Big Data SQL Master Compatibility Matrix* (Doc ID 2119369.1) in My Oracle Support for supported Linux distributions, Oracle Database release levels, and required patches.

> **Note:**
>
> Be sure that the correct Bundle Patch and any one-off patches identified in the Compatibility Matrix have been pre-applied before starting this installation.

Recommended Network Connections to the Hadoop Cluster

For Ethernet connections between Oracle Database and the Hadoop cluster, Oracle recommends 10 Gb/s Ethernet.

Environment Variables

The following are required:

- ORACLE_SID
- ORACLE_HOME

GI_HOME (Grid Infrastructure home directory) is optional if it is located on a relative path accessible from ORACLE_HOME. Otherwise, if Grid home exists then GI_HOME must be set. If Grid home does not exist on a node, then you must use the `crs=false` option when you run the `bds-database-install.sh` script as described in the steps below

Copy, Extract, and Run the Big Data SQL Installation Script on all Database Nodes

Perform the procedure in this section as the `oracle` user, except where `sudo` is indicated.

The latter part of the installation may require a single restart of Oracle Database under either or both of these conditions:

- If Oracle Database does not include the Oracle Grid Infrastructure. In this case, the installation script makes a change to the `pfile` or `spfile` configuration file in order to support standalone operation of `diskmon`.
- If there are changes to the IP address and the communication protocol recorded in `cellinit.ora`. These parameters define the connection to the cells on the Hadoop cluster. For example, if the IP address changes from an Ethernet address to an InfiniBand address and/or the protocol changes (between TCP and UDP), then a database restart is required.

1. If you haven’t done so already as part of the instructions in Creating the Database-Side Installation Bundle, copy the installation bundle from the `BDSSetup` directory on the Hadoop cluster management server to `/home/oracle` on the database node.

   ```
   $ scp bds-database-install.zip oracle@<database_node>:/home/oracle
   ```

2. Log on to the database node as `oracle` in order perform the installation.
3. Unzip the bundle and then `cd` into the `bds-database-install` directory that was extracted from the zip file.

   ```
   $ unzip bds-database-install.zip
   $ cd bds-database-install
   ```

4. Run `bds-database-install.sh`.

   Note the optional `bds-database-install` parameters listed in Table 3-1.

   ```
   [oracle@mydbserver bds-database-install]$ ./bds-database-install.sh
   ```

Special Conditions:

- If the database does not use Oracle Grid, then use the `--crs=false` option with the `bds-database-install.sh` command. For example:

  ```
  [oracle@mydbserver bds-database-install]$ ./bds-database-
  install.sh --crs=false --dbname=<db-name>
  ```

  If Oracle Grid is not running, you must also start MTA extproc manually, as in `ORACLE_HOME/bin/mtactl start <MTA_SID>`. In the default installation, `MTA_SID` is formulated as `bds_<ORACLE_SID>_<Hadoop_cluster_name>`.

If `bds-database-install.sh` finds and updates `/etc/oracle/cell/network-config/cellinit.ora`, then the installation is complete and you will see this message:

```
bds-database-install: installation successful for cluster <cluster name>
```

Table 3-1 Optional Parameters for bds-database-install.sh

Parameter	Function
`--crs`	Specifies use of Oracle Grid support. If Grid is not running on this database node, or, if this database does not use Grid (CRS/ASM) in any way, then apply the `--crs=false` option with the `bds-database-install.sh` script.
`--cdb`	Specifies the use of multitier databases.
`--db-resource`	Specify the Oracle Database SID (Oracle Database instance name). The parameter `--db-name` is synonym from older releases and is still supported.
`--debug`	Activate shell trace mode. If you report a problem, Oracle Support may want to see this output.
`--grid-home`	Specifies the Oracle Grid home directory.
`--info`	Show information about the cluster.
`--ip-cell`	Set a particular IP address for db_cell process. See *If You Need to Change the Configured Database_IP Address* below.
`--install-as-secondary`	Specifies a secondary installation. See *Primary and Secondary Installations* below.
`--jdk-home`	Specifies the JDK home directory.
`--reconfigure`	Modify the parameters of the connection of this node with the Hadoop cluster.
`--root-script-only`	Generate the root script only.

Table 3-1 (Cont.) Optional Parameters for bds-database-install.sh

Parameter	Function
`--uninstall-as-primary`	Removes information about the primary cluster from the database. **Important:** Note that `--uninstall-as-primary` also removes key information that is also used by any secondary clusters. To avoid this, use `--uninstall-as-secondary` to uninstall the primary unless you do intend to unregister all Hadoop clusters from the database. Otherwise to restore the connection to earch secondary Hadoop cluster, you would need to reinstall the software from the database bundle that was generated for that cluster.
`--uninstall-as-secondary`	Removes the metadata about a Hadoop cluster from Oracle Database, but does not remove key information that is common to all clusters.
`--version`	Show the `bds-database-install.sh` script version.

Primary and Secondary Installations

Oracle Big Data SQL on the Oracle Database side can provide access to a single Hadoop cluster or multiple Hadoop clusters (which can be on the same rack or on different on racks in the network).

Each installation registers a Hadoop cluster with the database. The first cluster registered with the database becomes the default/primary Hadoop cluster for the database. This first installation also installs the JDK, Hadoop client, and Hive client needed to access data on a Hadoop cluster.

If you want the database to connect to an additional Hadoop cluster, deploy the database-side installation bundle generated on that cluster to the database server. Unpack it, and run `bds-database-install.sh` with the `--install-as-secondary` option. This “secondary installation” adds the metadata about the additional Hadoop cluster to the Oracle Database without unnecessarily repeating the installation of the clients and the JDK.

When there are multiple clusters (one primary and any number of secondaries) registered with the database, an Oracle Big Data SQL user can query each Hadoop cluster separately or do JOINs to combine data from different clusters.

If You Need to Change the Configured Database_IP Address

The `DATABASE_IP` parameter in the `bds-config.json` file identifies the network interface of the database node. If you run `bds-database-install.sh` with no parameter passed in, it will search for that IP address (with that length, specifically) among the available network interfaces. You can pass the `--ip-cell` parameter to `bds-database-install.sh` in order to override the configured `DATABASE_IP` setting:

```
$ ./bds-database-install.sh --ip-cell=10.20.30.40/24
```

Possible reasons for doing this are:

- `bds-database-install.sh` terminates with an error. The configured IP address may be wrong.
- There is an additional database node in the cluster and the defined `DATABASE_IP` address is not a network interface of the current node.
- The connection is to a multi-node database. In this case, perform the installation on each database node. On each node, use the `--ip-cell` parameter to set the correct `DATABASE_IP` value.

To determine the correct value for `ip-cell`, you can use list all network interfaces on a node as follows:

```
/sbin/ip -o -f inet addr show
```

3.2 Potential Requirement to Restart Grid Infrastructure

In certain database environments `bds-database-install.sh` needs to create `cellinit.ora` and/or `celliniteth.ora`. If so, then similar changes need to be propagated across all nodes in the grid infrastructure. In those cases, the script expects passwordless SSH setup between `oracle` and `root` or it will prompt for the password for each node during execution. If the nature of the changes requires a restart of the grid infrastructure, the script will also display messages indicating that grid infrastructure needs to be restarted manually. Because the installlation cannnot complete without grid credentials if a restart is necessary, be sure that you have the grid password at hand.

3.2.1 Understanding When Grid or Database Restart is Required

On the database side of Oracle Big Data SQL, the diskmon process is the agent in charge of communications with the Hadoop cluster. This is similar to its function on the

Oracle Exadata Database Machine, where it manages communications between compute nodes and storage nodes.

In Grid environments, diskmon is owned by the Grid user. In a non-Grid environment, it is owned by the Oracle Database owner.

In Oracle Big Data SQL, diskmon settings are stored on `cellinit.ora` and `celliniteth.ora` files in the `/etc/oracle/cell/network-config/` directory. The installer updates these files in accordance with the cluster connection requirements.

This is how the installer determines when Grid or Oracle Database needs to be restarted:

- If the installer detects that no previous `cellinit.ora` or `celliniteth.ora` file exists, this means that no diskmon process is running. In this case, if the environment includes Oracle Grid then you must restart the Grid. If the environment does not include Grid, then you must restart Oracle Database.
- If previous `cellinit.ora` and/or `celliniteth.ora` file exist, this indicates that diskmon process is running. In this case, if the installer needs to make a change to these files, then only the database needs to be restarted.
- In multi-node Grid environments, diskmon works on all nodes as a single component and `cellinit.ora` and `celliniteth.ora` must be synchronized on all nodes. This task is done through SSH. If passwordless SSH is setup on the cluster, no user interaction is required. If not passwordless SSH is not set up, then the script will pause for you to input the root credentials for all nodes. When the cellinit.ora and celliniteth.ora files across all nodes are synchronized, then the script will continue. Then the script finishes and in this case, you must restart the Grid infrastructure.

3.3 Granting Oracle Big Data Access to Users

In Oracle Big Data SQL releases prior to 3.1, access is granted to the `PUBLIC` user group. In 3.1 (and later releases), you must do the following for each user who needs access to Oracle Big Data SQL:

- Grant the `BDSQL_USER` role.
- Grant read privileges on the BigDataSQL configuration directory object.

For example, to grant access to `user1`:

```
SQL> grant BDSQL_USER to user1;
SQL> grant read on directory ORACLE_BIGDATA_CONFIG to user1;
```

4 Uninstalling Oracle Big Data SQL

If you want to completely uninstall Oracle Big Data SQL, the software must be removed from the Hadoop cluster management server and the Oracle Database nodes where it is currently installed.

If the Hadoop DataNode service is removed from a subset of the nodes, then Oracle Big Data SQL must be removed from those nodes only.

4.1 General Guidelines for Removing the Software

Oracle Big Data SQL can be uninstalled from the Hadoop cluster management server or from any Oracle Database servers connected to the cluster management server. The procedure is the same for all Hadoop platforms.

Guidelines for uninstalling Oracle Big Data SQL are as follows:

- To perform a complete uninstall of Oracle Big Data SQL, remove the software from the cluster management server and from each Oracle Database server connected to the BDS service.

  This is a single script execution on each server. Not other manual steps are needed.
- You can uninstall from the cluster management server first or from the database servers first.

  > **Tip:**
  >
  > If you are removing the software from both sides of the installation, uninstall it from the database side first. If you uninstall from the cluster management server first, queries in process will fail.
- On the database side, uninstall from any secondary nodes before uninstalling from the primary node.

  This is not critical to the uninstall process, but active queries from secondary nodes will fail if the primary node is disconnected from the service.
- You can uninstall from one or more secondary database nodes without impacting the operation of the Big Data SQL service on the Hadoop cluster management server.

> **Note:**
>
> In a Oracle RAC database, be sure to run the uninstall script on all nodes of the database.

4.2 Uninstalling From an Oracle Database Server

On any database server where you want to uninstall Oracle Big Data SQL, run the appropriate command below as the database owner (usually the `oracle` user). In the case of a RAC database, be sure to perform the uninstall on each instance of the RAC.

- `# ./bds-database-install.sh --uninstall-as-primary`

  or

- `# ./bds-database-install.sh --uninstall-as-secondary`

If you use the `--uninstall-as-secondary` switch to uninstall the software from the primary Hadoop cluster configuration, cleanup of database objects will be incomplete. This can be remedied by running the uninstall again. Error messages may appear if your run a second uninstall for cleanup purposes, but the cleanup should complete successfully.

See Using bds-database-install.sh to Uninstall Oracle Big Data SQL for an example.

The `bds-database-install.sh` script is for installing/uninstalling the database side for Oracle Big Data SQL 3.1 or greater. Note that if you are uninstalling Release 3.0.1 or earlier on Oracle Big Data Appliance, use the `bds-exa-install.sh` script from your existing Oracle Big Data SQL installation instead.

4.3 Removing Oracle Big Data SQL From Hadoop Cluster Nodes

Oracle Big Data SQL works with the Hadoop DataNode service. When you run the Oracle Big Data SQL installer on the configuration management server, it automatically propagates the Oracle Big Data SQL components to all Hadoop cluster nodes where the DataNode service is running. If the DataNode service is removed from one or more nodes, then you must also remove the Oracle Big Data SQL components from those same nodes.

> **Note:**
>
> This task does not apply to Oracle Big Data Appliance.

On the cluster management server (where CM or Ambari is running), log on as `root`, cd to the Oracle Big Data SQL installer directory (where the download bundle BigDataSQL-<*Hadoop distribution*><*version*>.zip was extracted), and enter the following command:

```
# ./setup-bds remove bds-config.json
```

The `setup-bds` script with the `remove` option analyzes the cluster to discover any nodes where Oracle Big Data SQL was installed, but where the DataNode service had been subsequently removed. `setup-bds remove` removes the Oracle Big Data SQL components from those nodes. It also updates the `bigdata.hosts` file on each remaining Oracle Big Data SQL-enabled node in order to provide an up-to-date inventory of which nodes within the cluster are running Oracle Big Data SQL.

`setup-bds uninstall` has a different purpose. The uninstall option is for a complete de-installation of Oracle Big Data SQL.

4.4 Uninstalling From the Hadoop Cluster Management Server

Log on to the Hadoop cluster management server as `root` and use `setup-bds uninstall` to uninstall the software.

> **Before You Start:**
>
> If the Hadoop system is Oracle Big Data Appliance and you uninstalling Oracle Big Data SQL 3.0.1 in order to upgrade to Release 3.1, do not use the `setup-bds uninstall` command show below. Instead, use `bdacli disable big_data_sql` as described in the *Oracle Big Data Appliance Owner's Guide*.

To uninstall Oracle Big Data SQL 3.1 or later, from the `bds-database-install` directory run the following command as `root`.

```
# ./setup-bds uninstall bds-config.json
```

The script will return the following to standard output. This example shows the output of the uninstall on an HDP cluster management server. The output is similar for CDH. Italicized annotations in brackets are included for the purposes of the example.

```
[root@mycluster22bda16-adm BDSSetup]# ./setup-bds uninstall bds-config.json
BigDataSQL: FQDN: mycluster22bda16-adm.us.oracle.com.
BigDataSQL: Hostname: mycluster22bda16-adm.us.oracle.com.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/hortonworks.
BigDataSQL: Verifying configuration parameters...
BigDataSQL: Starting server...
BigDataSQL: API protocol: http.
BigDataSQL: API port: 8080.
BigDataSQL: Waiting for server to respond...

Admin User: admin
Password:

BigDataSQL: Authentication succeeded.
BigDataSQL: API version: 1.
BigDataSQL: Cluster type: HDP.
BigDataSQL: Retrieving cluster...
BigDataSQL: Cluster name: HDP24.
BigDataSQL: Cluster version: HDP-2.4.
BigDataSQL: Setting target cluster...
BigDataSQL: Service: ORACLESERVICE.
BigDataSQL: Looking for configured port on the JSON config file...
BigDataSQL: Testing port...
```

```
BigDataSQL: Using port: 81.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/hortonworks/services/bigdatasql.
BigDataSQL: Verifying service configuration...
BigDataSQL: Service TCP connection: False.
BigDataSQL: Uninstalling service...
BigDataSQL: Restoring cluster...
BigDataSQL: Removing service distribution...
BigDataSQL: Cleaning up service...
BigDataSQL: UNINSTALL workflow completed.
```

5
Adapting to Hadoop Cluster and Oracle Database Changes

Changes that occur to the Hadoop cluster over time, such as network reconfiguration or DataNode services added, moved, or removed, as well as changes on the Oracle Database server, can require corresponding changes to the Oracle Big Data SQL installation.

5.1 Installing on New Hadoop DataNodes After a Cluster Extension

If nodes running the DataNode service are added to the Hadoop cluster, then Oracle Big Data SQL must be installed on these nodes.

Run `setup-bds` with the `extend` argument. This operation installs the Oracle Big Data SQL components on nodes where they are required and updates the cells inventory.

```
# ./setup-bds extend bds-config.json
```

5.2 Reconfiguring an Existing Oracle Big Data SQL Installation on Database Nodes

When Oracle Big Data SQL is initially installed on the cluster management server, the database-side installation bundles you create contain cluster configuration information that enables Oracle Big Data SQL on the database server to connect to the Hadoop cluster. After a configuration change in the Hadoop cluster, the set of changes must be propagated to the database nodes in order to maintain connectivity with the cluster. To do this, run `setup-bds reconfigure bds-config.json` on the the cluster management server to capture the changes. Then, `bds-database-bundle-creation.sh --reconfigure` to create an updated database bundle. And finally, copy the bundle over to each of the database nodes, unpack it, and run `bds-database-install.sh --reconfigure`.

Perform this procedure if Oracle Big Data SQL is already installed on a Oracle Database node. (To perform a full installation of the software on a database node, do not use this procedure. In that case, see Installing on the Oracle Database Server instead.)

Here are some Hadoop-side changes which require you to reconfigure Oracle Big Data SQL on the database nodes:

- Changes to the DataNode inventory – DataNodes have be added, moved, or removed.
- Network security changes, such as a switch from HTTP to HTTPS or vice versa.
- Port reassignments in the cluster management service (CM or Ambari).

- Hive changes, such as migration of the service to different node or nodes or a change to the authentication method.
- Changes to other related Hadoop services, such as HDFS, MapReduce2, and YARN, may also affect connectivity.

> **Important:**
>
> You should perform this task as soon as possible after a cluster-side configuration change, because new or reconfigured DataNodes may be invisible to Oracle Big Data SQL on the database server.
> If the DataNodes inventory changes, skip Step 1 below. Instead, make any needed adjustments to `bsd-config` and run `setup-bds extend bds-config.json` or `setup-bds remove bds-config.json` as appropriate. This is a necessary prerequisite and both of these commands implicitly include `setup-bds reconfigure bds-config.json`.

1. On the cluster management server, cd to the `/BDSSetup` directory and run `setup-bds` as `root`, using the `reconfigure` option.

   ```
   [root@myclustermgmtserver: BDSSetup] # ./setup-bds reconfigure bds-config.json
   ```

   This updates the cluster configuration files that will be incorporated into the new database bundle.
2. Cd to `/BDSSetup/db` and recreate the bundle files by executing the `bds-database-create-bundle.sh` script. Use the `--reconfigure` switch.

   ```
   [root@myclustermgmtserver: db] # ./bds-database-bundle-creation.sh --reconfigure
   ```

   This command deletes the old database bundle files and recreates them, using the up-to-date configuration files generated by the previous command.
3. Copy the new `bds-database-install-config.zip` bundle file to the `oracle` home on the database node. Connect using the `oracle` account on the database node.

   ```
   [root@myclustermgmtserver: db] scp bds-database-install-config.zip
   oracle@dbnode:/home/oracle
   ```
4. Log on to the database node as the `oracle` user.
5. Locate the `bds-database-install-config.zip` file that you copied over from the cluster management server (as described in Creating the Database-Side Installation Bundle).
6. Unzip the bundle into the directory where Oracle Big Data SQL is already installed. Use the `-o` switch in order to overwrite the existing installation:

   ```
   [oracle@mydbserver: bds-database-install] $ unzip -o bds-database-install-
   config.zip -d /home/oracle/bds-database-install
   ```
7. Run `bds-database-install.sh` with the `--reconfigure` switch. This updates the local configuration on the database node.

   ```
   [oracle@mydbserver: bds-database-install] $ ./bds-database-install.sh --
   reconfigure
   ```

5.3 Extending an Existing Installation to New Database Nodes

If you want to install Oracle Big Data SQL on additional Oracle Database nodes at any time after the initial installation of Oracle Big Data SQL, install the full database-side installation bundle (`bds-database-install.zip`) on these nodes.

> **Note:**
>
> The full bundle contains the required Hadoop client tarballs as well as the Hadoop cluster configuration files. Do not use the `bds-database-install-config.zip` bundle for a full installation on a database node. `bds-database-install-config.zip` is for reconfiguring existing Oracle Big Data SQL installations on the database side and does not contain all of files required for a complete installation.

1. Even if you still have a copy of `bds-database-install.zip` from a previous installation, you should first regenerate the database bundle as described in Creating the Database-Side Installation Bundle and use the new version. This ensures that the bundle contains the current configuration data.
2. Deploy and install the bundle on the new database nodes as described in Installing on the Oracle Database Server.

6 Securing Big Data SQL

This section describes measures you can take to secure Big Data SQL and to configure the software within secured environments.

6.1 Big Data SQL Communications and Secure Hadoop Clusters

It is generally a good security practice to ensure that HDFS file access permissions are minimized in order to prevent unauthorized write/read access. This is true regardless of whether or not the Hadoop cluster is secured by Kerberos.

Please refer to MOS Document 2123125.1 at My Oracle Support for detailed guidelines on securing Hadoop clusters for use with Oracle Big Data SQL.

6.2 Installing a Kerberos Client on the Database Server

If Kerberos is enabled on the Hadoop system, you must configure Oracle Big Data SQL on the database server to work with Kerberos. This requires a Kerberos client on each node of the database.

For commodity servers, download the Kerberos client software from a repository of your choice. If the database server is an Oracle Exadata Database Machine, download and install the software from the Oracle repository as shown below. The process should be similar for downloads from non-Oracle repositories.

Log on to the database server as root and use yum to install the `krb5-libs` and `krb5-workstation` packages. Download from the Oracle Linux 6 or Oracle Linux 5 repository as appropriate.

1. Check that the Oracle `public-yum-ol6` or `public-yum-ol5` repository ID is installed.

   ```
   # yum repolist
   ```

2. Temporarily disable all repository IDs and then enable the Oracle repository only (Oracle Linux 6 in this example).

   ```
   # yum --disablerepo="*" --enablerepo="public-yum-ol6" list available
   ```

3. Install the Kerberos packages.

   ```
   # yum install krb5-libs krb5-workstation
   ```

4. Copy the `/etc/krb5.conf` file from the Key Distribution Center (KDC) to the same path on the database server.

These steps must be performed for each Oracle Database node.

You must also register the `oracle` Linux user (or other Linux user) and password in the KDC for the cluster as described in Enabling Oracle Big Data SQL Access to a Kerberized Cluster

6.3 Enabling Oracle Big Data SQL Access to a Kerberized Cluster

You must configure Oracle Big Data SQL to use Kerberos in environments where user access is Kerberos-controlled.

There are two situations when this is required:

- When enabling Oracle Big Data SQL on a Kerberos-enabled cluster.
- When enabling Kerberos on a cluster where Oracle Big Data SQL is already installed.

Oracle Big Data SQL processes run on the nodes of the Hadoop cluster as the `oracle` Linux user. On the Oracle Database server, the owner of the Oracle Database process is also (usually) the `oracle` Linux user. When Kerberos is enabled on the Hadoop system, the following is required in order to give the user access to HDFS.

- The `oracle` Linux user needs to be able to authenticate as a principal in the Kerberos database on the Kerberos Key Distribution Center (KDC) server. The principal name in Kerberos does not have to be 'oracle'. However, the principal must have access to the underlying Hadoop data being requested by Oracle Big Data SQL.
- The following are required on all Oracle Database nodes and all Hadoop cluster nodes running Oracle Big Data SQL:
  - Kerberos client software installed.
  - A copy of the Kerberos configuration file from the KDC.
  - A copy of the Kerberos keytab file generated on the KDC for the `oracle` user.
  - A valid Kerberos ticket for the `oracle` Linux user.

Installing the Kerberos Client

If the Kerberos client is not installed, see Installing a Kerberos Client on the Database Server for instructions on installing the Kerberos client.

Creating a Kerberos Principal for the oracle User

On the Kerberos Key Distribution Center (KDC) server, become `root` and use `kadmin.local` to add a principal for the `oracle` user.

1. `# kadmin.local`
2. Within kadmin.local, type:

   ```
   add_principal <user>@<realm>
   quit
   ```

   You have the option to include the password, as in:

   ```
   add_principal <user>@<realm> -pw <password>
   quit
   ```

Creating a Kerberos Keytab for the oracle User

1. On the KDC, become `root` and run the following:

```
# kadmin.local
```

2. Within `kadmin.local`, type:

   ```
   xst –norandkey -k /home/oracle/oracle.keytab oracle
   quit
   ```

   This creates the `oracle.keytab` file for the Kerberos `oracle` user in the `/home/oracle` directory.

3. Ensure that `oracle.keytab` is owned by the `oracle` Linux user and is readable by that user only.

   ```
   $ chown oracle oracle.keytab
   $ chmod 400 oracle.keytab
   ```

Distributing the Keytab and Kerberos Configuration Files

Log on to the KDC and copy these local files to the same path on each Hadoop cluster node and all Oracle Database compute nodes that use Oracle Big Data SQL.

- Become the `oracle` user and copy `/home/oracle/oracle.keytab` to `/home/oracle/oracle.keytab` on each node.
- Become `root` and copy the Kerberos configuration file `/etc/krb5.conf` to `/etc/krb5.conf` on each node.

Be sure you have retained the original permissions on the files.

Acquiring a Kerberos Ticket for oracle on Each Node

The `oracle` user on each Hadoop DataNode and the database owner on each Oracle Database compute node (usually the `oracle` Linux user as well) need a valid ticket to connect through Kerberos.

After the Kerberos client is installed and the `krb5.conf` and `oracle.keytab` files are in place, log onto each node as the `oracle` user and obtain a ticket for the user.

1. Log on as the `oracle` user.
2. Run `kinit` on the `oracle` account.

   ```
    $ /usr/bin/kinit oracle -k -t /home/oracle/oracle.keytab
   ```

   A password is not required if a keytab for the user is defined as shown in this example. You can also use `/usr/bin/kinit oracle` without passing in a keytab file, but in that case you are prompted for the password defined for the principal.

Checking for a Valid Ticket

After you acquire a ticket for each node (or if you are unsure about the validity of a ticket), you can check for a valid ticket using `klist`.

```
$ /usr/bin/klist

Kerberos Utilities for Linux: Version 12.1.0.2.0 - Production on 19-SEP-2016 07:45:20
Copyright (c) 1996, 2014 Oracle. All rights reserved.
Ticket cache: /u01/app/oracle/product/12.1.0/dbhome_1/network/admin/krbcache
Default principal: <userID>@<realm>
```

If there is no valid ticket you will see a message similar to this:

```
klist: No credentials cache found
```

Cleaning up After Ticket Expirations

When the bd_cell process is running on the nodes of a secured Hadoop cluster but the Kerberos ticket is not valid, then the cell goes to quarantine status. You should drop all such quarantines.

1. Check that the `oracle` user has a valid Kerberos ticket on all Hadoop cluster nodes.
2. On each cluster node, become `oracle` and run the following:

   ```
   $ /opt/oracle/bd_cell/cellsrv/bin/bdscli
   ```

3. In the bdscli shell, type:

   ```
   list quarantine
   ```

4. While still in bdscli, drop each quarantine on the list:

   ```
   drop quarantine <id>
   ```

5. Type `exit` to exit bdscli.

Automating Kerberos Ticket Renewal to Avoid Expirations

The `oracle` user needs a valid Kerberos ticket on every Oracle Database instance that is accessing the Hadoop cluster. A valid ticket is also required for the `oracle` user on the Hadoop nodes, since this user owns each Big Data SQL process.

It is best to automate the ticket renewal. Use `cron` or a similar utility to run `kinit` to acquire a new ticket for the user on a schedule that is prior to the ticket expiration date. (The ticket lifetime is recorded in `/etc/krb5.conf`.)

For example, on all nodes you could add a job to the `oracle` user's crontab as follows.

1. Log on as the user (`oracle` or other).
2. Use `crontab -e` to edit the user's crontab.

   ```
   $ crontab -e
   ```

3. Add a line with a schedule and the command to renew the ticket before the ticket expiry. For example, if the ticket expires every two weeks then you may want to renew it every 13 days, as in this example.

   ```
   15 1,13 * * * /usr/bin/kinit oracle -k -t /home/oracle/oracle.keytab
   ```

6.4 Using Oracle Secure External Password Store to Manage Database access for Oracle Big Data SQL

On the Oracle Database server, you can use the Oracle Secure External Password Store (SEPS) to manage database access credentials for Oracle Big Data SQL.

This is done by creating an Oracle wallet for the `oracle` Linux user (or other database owner). An Oracle wallet is a password-protected container used to store authentication and signing credentials, including private keys, certificates, and trusted certificates needed by SSL.

See MOS Document 2126903.1 at My Oracle Support for information on using SEPS with Oracle Big Data SQL.

6.5 About Data Security on Oracle Big Data Appliance

If your Hadoop system is an Oracle Big Data Appliance, the following tools to strengthen security are already available.

- **Kerberos authentication**: Requires users and client software to provide credentials before accessing the cluster.
- **Apache Sentry authorization**: Provides fine-grained, role-based authorization to data and metadata.
- **HDFS Transparent Encryption**: Protects the data on disk and at rest. Data encryption and decryption is transparent to applications using the data.
- **HTTPS/ Network Encryption**

  : Provides HTTPS for Cloudera Manager, Hue, Oozie, and Hadoop Web UIs. Also Enables network encryption for other internal Hadoop data transfers, such as those made through YARN shuffle and RPC.
- **Oracle Audit Vault and Database Firewall monitoring**: The Audit Vault plug-in on Oracle Big Data Appliance collects audit and logging data from MapReduce, HDFS, and Oozie services. You can then use Audit Vault Server to monitor these services on Oracle Big Data Appliance

See Also:

The *Oracle Big Data Appliance Guide* provides details on available security features. You can find this guide and other documentation for your release of the Oracle Big Data Appliance software in the “Big Data” section of the Oracle Help Center.

7

Additional Tools Installed

7.1 Copy to Hadoop and OHSH

Copy to Hadoop

Copy to Hadoop makes it easy to identify and copy data from an Oracle Database table to the Apache Hadoop Distributed File System (HDFS) and create an Apache Hive external table over that data. In HDFS, the data can be saved in Oracle Data Pump files (which store data as Oracle data types). This provides an exact copy of the data in the database, without the potential for data type conversion errors. You can then query the data via Hive (and Hadoop technologies that access data through Hive) and by Oracle Big Data SQL. Oracle Data Pump data can be converted to Parquet or ORC using Hive

Oracle Shell for Hadoop Loaders (OHSH)

OHSH is an intuitive command line tool to move data between Hadoop and Oracle Database. It provides a set of declarative commands you can use to copy contents from an Oracle Database table to a Hive table.

OHSH is bundled with both Oracle Big Data SQL and Oracle Big Data Connectors. With Oracle Big Data SQL, OHSH works with Copy to Hadoop. With Oracle Big Data Connectors, OHSH provides an interface to copy contents from Hadoop and Hive to Oracle tables using Oracle Loader for Hadoop (OLH) and Oracle SQL Connector for Hadoop Distributed File System (OSCH).

Although OHSH can serve as a front end for OLH and OSCH as well as for Copy to Hadoop, OLH and OSCH are part of the Oracle Big Data Connectors product and are not installed by Oracle Big Data SQL.

Installation and Configuration of Copy to Hadoop and OHSH

You can use Copy to Hadoop and OHSH from any of these environments:

- A Hadoop cluster node
- An Oracle Database node
- An edge node.

Oracle Big Data SQL installs the kits for Copy to Hadoop and Oracle Shell for Hadoop Loaders on the Hadoop cluster management server and on any Oracle Database server node where you install Oracle Big Data SQL.

You can also manually copy the kits for both tools from the Hadoop cluster management server to an edge node and configure them for use from the edge node. The kits are stored at `/opt/oracle/orahivedp-<version>`.. See `/opt/oracle/ohsh-<version>/doc/OHSH_EDGE_NODE_README.txt` on the Hadoop cluster management server for instructions.

> **Note:**
>
> In the Oracle Big Data SQL installation, the Copy to Hadoop is fully automated except for the setting of the `CP2HADOOP_HOME` listed in the table below. You must set this manually.
> You must complete configuration of Oracle Shell for Hadoop Loaders manually, as described in this section.

Environment Variables for OHSH

On the Hadoop cluster management server, the OHSH configuration file is`/opt/oracle/ohsh-<version>/bin/ohsh_config.sh`. On an Oracle Database node, it is `$ORACLE_HOME/ohsh/bin/ohsh_config.sh`. This file contains environment required by OHSH. The set of variables needed the installation on a Hadoop cluster, Oracle Database Node or an edge node differ. The table below shows which variables you need to set for each environment. Some of the required variables are set for you automatically by the Oracle Big Data SQL installer.

7.1.1 Completing the OHSH Configuration on Oracle Database Nodes

The Oracle Big Data SQL installation automatically copies the OHSH installation kit to each node where Oracle Big Data SQL is installed. There are some manual steps you must perform to complete the installation.

> **Note:**
>
> The README file at `$ORACLE_HOME/ohsh/doc/OHSH_DB_NODE_README.txt` on the Hadoop cluster management server where you installed Oracle Big Data SQL describes the full set of steps ordinarily needed to enable OHSH on an Oracle Database node. However, when OHSH is installed by Oracle Big Data SQL, some of the setup is done for you by the installer. To complete the configuration in this context, use this shortened procedure instead of the full procedure documented in the README.
> Also notice below that patch 23237037 is required for 12.1.0.2 (or earlier Oracle Database releases) only. Do not install this patch if you are running a later release of Oracle Database.

1. Log on to the Oracle Database server as the `oracle` user.
2. Set up OHSH to work with Copy to Hadoop.
   a. Create a staging directory in the file system on the Oracle Database node:

      ```
      $ mkdir <OS_STAGE_DIR>
      ```

   b. Create an Oracle directory on this staging directory and grant read/write privileges to the Oracle Database user (`oracle` or other). In this example, *ORACLE_STAGE_DIR* refers to the name of the directory object in Oracle Database. *OS_STAGE_DIR* refers to the path on the server.

```
SQL> CREATE DIRECTORY <ORACLE_STAGE_DIR> AS '<OS_STAGE_DIR>';
SQL> GRANT READ,WRITE ON DIRECTORY <ORACLE_STAGE_DIR> TO <ORACLE_USER>;
```

> **NOT_SUPPORTED:**
>
> When you use OHSH later to copy Oracle tables to Hive using the "stage" method, use the OHSH `set locationdirectory` command to point to the Oracle directory object name (the name of the directory you created above in this step).

c. If you are connecting to a CDH-based Hadoop system (Oracle Big Data Appliance or other), perform the substeps below to enable the OHSH “FUSE” method, in which you copy files to Hive tables over an HDFS Mountable FUSE directory. Note that HDP does not support Fuse, therefore you cannot create a mount point to HDFS on HDP using these steps.

   i. If you are using **Oracle Database 12.1.0.2 or earlier**, install patch 23237037 (available on My Oracle Support) on all database nodes.

   ii. On the Oracle Database host, create a FUSE mount to the root of the HDFS system on the Hadoop cluster. Refer to Cloudera’s documentation on installing FUSE. You may find current instructions at Configuring Mountable HDFS on Cloudera’s website.

   iii. Create the Oracle directory on the FUSE mount and grant privileges. In this example, *ORACLE_FUSE_DIR* refers to the name of the directory object in Oracle Database. *OS_FUSE_MOUNT* refers to the path to the mount point on the server.

```
      SQL> CREATE DIRECTORY <ORACLE_FUSE_DIR> AS '<OS_FUSE_MOUNT>';
      SQL> GRANT READ,WRITE ON DIRECTORY <ORACLE_FUSE_DIR> TO
<ORACLE_USER>;
```

Note that when you use OHSH later to copy Oracle tables to Hive using the "FUSE" method, use the OHSH `set locationdirectory` command to point to the same `<ORACLE_FUSE_DIR>`

3. Edit `$ORACLE_HOME/ohsh/bin/ohsh_config.sh` in order to configure the home directories of dependencies. When OHSH is installed by Oracle Big Data SQL, most of the required variables are already set for you. Set `HS2_HOST_PORT` and (if needed) `HIVE_SESS_VAR_LIST` as follows.

```
export HS2_HOST_PORT=<HS2_HOST>:<HS2_PORT>
```

Also set `HIVE_SESS_VAR_LIST` to any required Hive session variables. For example:

- To connect to HiveServer2 with Kerberos authentication: `export HIVE_SESS_VAR_LIST="principal=<The server principal of HiveServer2>"`

  The Hive principal is specified by the configuration property `hive.server2.authentication.kerberos.principal` in `hive-site.xml`.

- To connect to HiveServer2 running in HTTP mode: `export HIVE_SESS_VAR_LIST="transportMode=http;httpPath=<The HTTP endpoint>"`

- To connect to HiveServer2 when SSL is enabled: `export HIVE_SESS_VAR_LIST="ssl=true;sslTrustStore=<Trust Store path>;trustStorePassword=<Trust Store password>"`

4. If you want to configure OHSH to use Oracle Wallet, set environment the variables TNS_ADMIN and WALLET_LOCATION to the directory path where TNS and Oracle Wallet are stored.

   ```
   export WALLET_LOCATION="<wallet_location>"
   export TNS_ADMIN="<tns_admin_location>"
   ```

   > **Note:**
   >
   > Later, when you start an OHSH session, if the TNS_ADMIN and WALLET_LOCATION are different on the Hadoop cluster then set hadooptnsadmin and hadoopwalletlocation:
   >
   > ```
   > set hadooptnsadmin <tns_admin_hadoop_node>
   > set hadoopwalletlocation <wallet_location_hadoop_node>
   > ```

5. Start an OHSH session and check for resources. This check indicates whether or not the configuration is correct.

   ```
   $ ohsh
   ohsh> show resources
   ```

   You should see the resources hadoop0, hive0, and bash0.

7.1.2 Completing the OHSH Configuration on the Hadoop Cluster

On the Hadoop cluster management server, Oracle Big Data SQL installs OHSH from an RPM to /opt/oracle/ohsh. Perform the steps below as root in order to complete the Hadoop side of the configuration.

> **Note:**
>
> The installation instructions at $ORACLE_HOME/ohsh/doc/OHSH_HADOOP_NODE_README.txt on the Hadoop cluster management server contain extra information that is not applicable to the installation of Copy to Hadoop and OHSH in the context of the Oracle Big Data SQL installation. Use the instructions below instead.

1. If the Hadoop system is not an Oracle Big Data Appliance, then download the SQL*Plus Instant Client and Oracle Instant Client for Linux Instant Client Downloads for Linux x86.

   Check to be sure that the version of the packages you select match the version of the Oracle Database. If the Oracle Database is 12.1.0.2, you can use the following downloads:

   - http://download.oracle.com/otn/linux/instantclient/121020/instantclient-basic-linux.x64-12.1.0.2.0.zip
   - http://download.oracle.com/otn/linux/instantclient/121020/instantclient-sqlplus-linux.x64-12.1.0.2.0.zip

   Extract both packages into the same directory.

Add the directory to PATH and LD_LIBRARY_PATH.

2. Create a local directory for TNS and from TNS_ADMIN on the Oracle Database host, copy the tnsnames.ora and sqlnet.ora file into this directory.

   If you create an Oracle Wallet for the Oracle Database host, then also copy over the Oracle Wallet file from the database server. Also check that sqlnet.ora points to the correct Oracle Wallet location.

   ```
   WALLET_LOCATION=
         (SOURCE=(METHOD=FILE)(METHOD_DATA=
           (DIRECTORY=<your wallet directory>)))
      sqlnet.wallet_override=true
   ```

3. As root, edit /opt/oracle/ohsh/bin/ohsh_config.sh. Set the following environment variables.

   ```
   export CP2HADOOP_HOME=/opt/oracle/orahivedp-<version
   export HS2_HOST_PORT=<HiveServer2 hostname>:<Port number>
   ```

   Also set HIVE_SESS_VAR_LIST to any required Hive session variables. For example:

   - To connect to HiveServer2 with Kerberos authentication: export HIVE_SESS_VAR_LIST="principal=<The server principal of HiveServer2>"
   - To connect to HiveServer2 running in HTTP mode: export HIVE_SESS_VAR_LIST="transportMode=http;httpPath=<The HTTP endpoint>"
   - To connect to HiveServer2 when SSL is enabled: export HIVE_SESS_VAR_LIST="ssl=true;sslTrustStore=<Trust Store path>;trustStorePassword=<Trust Store password>"

4. If TNS and Oracle Wallet are enabled, also add the following variables to the file and export them:

   ```
   export WALLET_LOCATION="<your wallet directory>"
   export TNS_ADMIN="<your TNS directory>"
   ```

5. Add /opt/oracle/ohsh to the PATH variable

   ```
   PATH=/opt/oracle/ohsh-<version>/bin:$PATH
   ```

6. Start an OHSH session and check for resources. This check indicates whether or not the configuration is correct.

   ```
   $ ohsh
   ohsh> show resources
   ```

   You should see the resources hadoop0, hive0, and bash0.

7.1.3 Getting Started Using Copy to Hadoop and OHSH

See $ORACLE_HOME/ohsh/doc and $ORACLE_HOME/ohsh/examples for examples that show how to run the OHSH load methods.

See the Oracle Big Data SQL User’s Guide more information on Copy to Hadoop and Oracle Shell for Hadoop Loaders.

A

Oracle Big Data SQL Install/Uninstall/Reconfigure Examples

The following are samples of the console output for each of the phases of a successful Oracle Big Data SQL install and uninstall. A sample that shows use of the `setup-bds reconfigure` option is also included. These examples are intended to provide a general overview of the operations. The actual output on your system may contain differences.

Using setup-bds to Install on the Hadoop Cluster

Run `setup-bds install bds-config.json` as root from the `BDSSetup` directory on the Hadoop cluster management server.

Command:

```
[root@myclustermgmtserver: BDSSetup] # ./setup-bds install bds-config.json
```

In the example below, note the use of SCL to specify Python 2.7. SCL is available on Oracle Big Data Appliance. It may or may not be available on other supported CDH systems or on HDP.

Example:

The following shows the Oracle Big Data SQL installer output on a CDH system. The output on HDP is similar. Note that you are prompted to enter the cluster management service administrator credentials and that the service is restarted. After unpacking the download bundle (`BigDataSQL-CDH-3.1.0.zip` in this CDH example), `cd` down into the `BDSSetup` directory, make the required `bds-config.json` edits, and then run `./setup-bds install bds-config.json`.

```
[root@myclusternode18 bds]# unzip -q BigDataSQL-CDH-3.1.0.zip

[root@myclusternode18 bds]# cd BDSSetup/
[root@myclusternode18 BDSSetup]# vim bds-config.json
[root@myclusternode18 BDSSetup]# scl enable python27 "./setup-bds install bds-
config.json"
BigDataSQL: FQDN: myclusternode18.us.mydomain.com.
BigDataSQL: Hostname: myclusternode18.us.mydomain.com.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/cloudera.
BigDataSQL: Verifying configuration parameters...
BigDataSQL: Starting server...
BigDataSQL: API protocol: https.
BigDataSQL: API port: 7183.
BigDataSQL: Waiting for server to respond...

Admin User: admin
Password:

BigDataSQL: Authentication succeeded.
BigDataSQL: API version: 15.
BigDataSQL: Cluster type: BDA.   // or other...
BigDataSQL: Retrieving clusters...
```

```
BigDataSQL: One cluster detected.
BigDataSQL: Cluster name: mycluster.
BigDataSQL: Cluster version: CDH5.
BigDataSQL: Setting target cluster...
BigDataSQL: Service: BIGDATASQL.
BigDataSQL: Looking for configured port on the JSON config file...
BigDataSQL: Testing port...
BigDataSQL: Using port: 81.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/cloudera/services/bigdatasql.
BigDataSQL: Verifying service configuration...
BigDataSQL: Service TCP connection: True.
BigDataSQL: Setting up related services...
BigDataSQL: Setting up database...
BigDataSQL: Setting up memory...
BigDataSQL: Loading service state file...
BigDataSQL: Distributing service packages...
BigDataSQL: Copying CSD to Cloudera Manager.
BigDataSQL: Restarting server...
BigDataSQL: Waiting for agents to respond...
BigDataSQL: Copying parcel repository to Cloudera Manager...
BigDataSQL: Starting webserver...
BigDataSQL: Waiting for server to respond...
BigDataSQL: Waiting for parcel to respond
BigDataSQL: Downloading parcel...
Progress: 100 / 100
BigDataSQL: Distributing parcel...
Progress: 300 / 300
BigDataSQL: Activating parcel...
Progress: 0 / 0
BigDataSQL: Preparing service...
BigDataSQL: Saving service state file.
BigDataSQL: Setting up Hive auxiliary path...
BigDataSQL: Adding bigdatasql service...
BigDataSQL: Installing bigdatasql service...
BigDataSQL: Starting bigdatasql service...
BigDataSQL: Restarting cluster...
BigDataSQL: Restarting Cloudera Management service...
BigDataSQL: Creating database configuration...
BigDataSQL: Downloading file from https://myclusternode18.us.mydomain.com:7183/cmf/
command/1689/download.
Progress [5424Kb]
BigDataSQL: Download complete, file: /var/log/bigdatasql/DM/inspection.json.
BigDataSQL: Cleaning up service...
BigDataSQL: INSTALL workflow completed.
```

Using bds-database-create-bundle.sh to Generate a Database-Side Installation Bundle

Command:

```
[root@myclustermgmtserver: db] # ./bds-database-create-bundle.sh
```

Run this script as root from the `BDSSetup/db` directory on the Hadoop cluster management server.

Example:

The following is the output on a CDH system. The output on HDP is similar. Note that you are prompted for the cluster management service administrator credentials.

Tip:

Users sometimes forget to set the proxies and then `bds-database-create-bundle.sh` fails to connect to the Internet . Set the proxies as shown below before running the script.

```
[root@myclusternode18 BDSSetup]# cd db
[root@myclusternode18 db]# export http_proxy=<your proxy>
[root@myclusternode18 db]# export https_proxy=<your proxy>
[root@myclusternode18 db]# ./bds-database-create-bundle.sh
bds-database-create-bundle: database bundle creation script started at : Tue Feb 28
22:04:28 PST 2017
bds-database-create-bundle: working directory         : /root/bds/BDSSetup
bds-database-create-bundle: Hadoop client download url : http://archive-
primary.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.10.0.tar.gz
bds-database-create-bundle: Hive client download url  : http://archive-
primary.cloudera.com/cdh5/cdh/5/hive-1.1.0-cdh5.10.0.tar.gz
bds-database-create-bundle: HBase client download url : http://archive-
primary.cloudera.com/cdh5/cdh/5/hbase-1.2.0-cdh5.10.0.tar.gz
mkdir: created directory `/root/bds/BDSSetup/bds-database-install'
bds-database-create-bundle: copying required files

Management server credentials for download configuration files
Userid  : admin
Password:

bds-database-create-bundle: downloading yarn configuration file
bds-database-create-bundle: downloading hive configuration file
bds-database-create-bundle: downloading hadoop client tarball
bds-database-create-bundle: downloading hive client tarball
bds-database-create-bundle: downloading hbase client tarball
bds-database-create-bundle: building bundles files
bds-database-create-bundle: database bundle creation script completed all steps
```

Using bds-database-install to Install the Software on the Oracle Database Server

Command:

```
[oracle@mydbserver: bds-database-install] $ ./bds-database-install.sh
```

Copy the `bds-database-install.zip` bundle from the cluster management server to the database server. Then, unpack it and run `bds-database-install.sh` as the `oracle` user.

This script installs Oracle Big Data SQL on the primary cluster. If this script modifies the cellinit*.ora files, then the database instance must be restarted.

Important:

Run the script on each node of the Oracle Database

Example:

The following is the output on a HDP system. The output on CDH is similar. Note that under some conditions (described in Installing on the Oracle Database Server) you may be prompted to restart the Oracle Database.

```
[oracle@mydomain21bda09 hdp24]$ unzip -q bds-database-install.zip
[oracle@mydomain21bda09 hdp24]$ cd bds-database-install
[mydomain@mydomain21bda09 bds-database-install]$ ./bds-database-install.sh
bds-database-install: platform is               : Linux
bds-database-install: script started at         : Tue Feb 28 22:22:05 PST 2017
bds-database-install: running location          : /home/oracle/hdp24/bds-database-
install
bds-database-install: ORACLE_SID                : orcl
bds-database-install: ORACLE_HOME               : /u03/app/oracle/product/12.1.0/
dbhome_1
bds-database-install: Grid home                 : /u03/app/oracle/product/12.1.0/grid
bds-database-install: configuration spfile      : spfileorcl.ora
bds-database-install: cluster type              : hdp
bds-database-install: running mode              : software only
bds-database-install: cluster name              : HDP24
bds-database-install: dictionary cluster name   : hdp24
bds-database-install: hadoop version            : hadoop-2.7.1.2.4.0.0-169
bds-database-install: hive version              : apache-hive-1.2.1000.2.4.0.0-169-
bin
bds-database-install: hbase version             : hbase-1.1.2.2.4.0.0-169
bds-database-install: bds version               : Big Data SQL 3.1.0
bds-database-install: bds install date          : Wed Mar  1 01:00:47 2017
bds-database-install: bd_cell version           :
bd_cell-12.1.2.0.100_LINUX.X64_170227.1054-1.x86_64
bds-database-install: cell config dir           : /etc/oracle/cell/network-config
bds-database-install: hybrid cell model         : false
bds-database-install: cell network(s)           : 180.168.41.9/21
bds-database-install: cellaffinity.ora file     : missing
bds-database-install: configured DB network     : 11.122.17.217/21
bds-database-install: listener end point        : EXTPROC1521
bds-database-install: db resource               : orcl
bds-database-install: database type             : SINGLE
bds-database-install: cardinality               : 1
bds-database-install: mta extproc sid           : bds_orcl_hdp24
bds-database-install: crs                       : true
bds-database-install: cdb                       : true
bds-database-install: action                    : setup
bds-database-install: validating 11.122.17.217/21 against network interfaces
bds-database-install: ping succeeded on all nodes, using 11.122.17.217/20 at
bondeth0 instead
/bin/mkdir: created directory `/u03/app/oracle/product/12.1.0/dbhome_1/bigdatasql'
bds-database-install: installing JDK tarball
bds-database-install: copying JCE policy jars
bds-database-install: creating big data sql directories
/bin/mkdir: created directory `bigdata_config_orcl'
/bin/mkdir: created directory `hdp24'
/bin/mkdir: created directory `default_dir_orcl'
/bin/mkdir: created directory `log'
/bin/mkdir: created directory `jlib'
bds-database-install: removing old oracle bds jars if any
bds-database-install: installing oracle bds jars
bds-database-install: validating hadoop-2.7.1.2.4.0.0-169.tar.gz
bds-database-install: validating apache-hive-1.2.1000.2.4.0.0-169-bin.tar.gz
bds-database-install: validating hbase-1.1.2.2.4.0.0-169.tar.gz
bds-database-install: unpacking hadoop-2.7.1.2.4.0.0-169.tar.gz
bds-database-install: unpacking apache-hive-1.2.1000.2.4.0.0-169-bin.tar.gz
bds-database-install: unpacking hbase-1.1.2.2.4.0.0-169.tar.gz
```

```
bds-database-install: unpacking hadoop-2.7.1-nativelib-ol6.tar.gz
bds-database-install: installing cluster configuration files
bds-database-install: created bigdata.hosts
bds-database-install: creating /u03/app/oracle/product/12.1.0/dbhome_1/bigdatasql/
hadoop_hdp24.env for hdfs/mapred client access
bds-database-install: creating bds property files
bds-database-install: created bigdata.properties
bds-database-install: created bigdata-log4j.properties
bds-database-install: creating default and cluster directories needed by big data
external tables
catcon: ALL catcon-related output will be written to bds-database-install-3617-
directory-creation_catcon_4435.lst
catcon: See bds-database-install-3617-directory-creation*.log files for output
generated by scripts
catcon: See bds-database-install-3617-directory-creation_*.lst files for spool
files, if any
catcon.pl: completed successfully
bds-database-install: granted default and cluster directories to BDSQL_USER role
bds-database-install: mta will be setup
bds-database-install: creating /u03/app/oracle/product/12.1.0/dbhome_1/hs/admin/
initbds_orcl_hdp24.ora
bds-database-install: mta shutdown bds_orcl_hdp24
ORA-28593: agent control utility: command terminated with error
bds-database-install: removing existing mta parameter settings
bds-database-install: stopping crs resource bds_orcl_hdp24
bds-database-install: deleting crs resource bds_orcl_hdp24
bds-database-install: registering crs resource bds_orcl_hdp24
bds-database-install: using dependency db resource of orcl
bds-database-install: starting crs resource bds_orcl_hdp24
bds-database-install: patching view LOADER_DIR_OBJS
catcon: ALL catcon-related output will be written to bds-database-install-3617-patch-
view_catcon_5016.lst
catcon: See bds-database-install-3617-patch-view*.log files for output generated by
scripts
catcon: See bds-database-install-3617-patch-view_*.lst files for spool files, if any
catcon.pl: completed successfully
bds-database-install: creating mta dblinks
catcon: ALL catcon-related output will be written to bds-database-install-3617-
create-links_catcon_5042.lst
catcon: See bds-database-install-3617-create-links*.log files for output generated
by scripts
catcon: See bds-database-install-3617-create-links_*.lst files for spool files, if
any
catcon.pl: completed successfully
catcon: ALL catcon-related output will be written to bds-database-install-3617-alter-
system_catcon_5064.lst
catcon: See bds-database-install-3617-alter-system*.log files for output generated
by scripts
catcon: See bds-database-install-3617-alter-system_*.lst files for spool files, if
any
catcon.pl: completed successfully
bds-database-install: setup cell contiguration
bds-database-install: updating cellinit.ora ip address
bds-database-install: enabling subnetwork searching for cluster connection in
cellinit.ora
bds-database-install: enabling UDP protocol for cluster connection in cellinit.ora
bds-database-install: creating celliniteth.ora
bds-database-install: cellaffinity.ora already disabled
database instance needs to be restarted, enter <yes> for restart it now or <no> for
restart it later:
bds-database-install: restarting database instance
```

```
bds-database-install: waiting upto 100 seconds for new diskmon process to startup
..ok
bds-database-install: stoping crs resource bds_orcl_hdp24
bds-database-install: starting crs resource bds_orcl_hdp24
bds-database-install: installation successful for cluster HDP24

bds-database-install: log files in /u03/app/oracle/product/12.1.0/dbhome_1/install
bds-database-install: bds-database-install-3617-* files for more detailed info
bds-database-install: full log in bds-database-install-3617-main-script.log
```

Using bds-database-install.sh to Uninstall Oracle Big Data SQL

Commands:

```
./bds-database-install.sh --uninstall-as-primary // or --uninstall-as-secondary
```

See Primary and Secondary Installations for guidance on when to use `--uninstall-as-primary` or `--uninstall-as-secondary`.

Example:

```
[oracle@mydomain21bda09 bds-database-install]$ ./bds-database-install.sh --uninstall-
as-primary
bds-database-install: platform is               : Linux
bds-database-install: script started at         : Tue Feb 28 22:26:07 PST 2017
bds-database-install: running location          : /home/oracle/hdp24/bds-database-
install
bds-database-install: ORACLE_SID                : orcl
bds-database-install: ORACLE_HOME               : /u03/app/oracle/product/12.1.0/
dbhome_1
bds-database-install: Grid home                 : /u03/app/oracle/product/12.1.0/grid
bds-database-install: configuration spfile      : spfileorcl.ora
bds-database-install: cluster type              : hdp
bds-database-install: running mode              : software only
bds-database-install: cluster name              : HDP24
bds-database-install: dictionary cluster name   : hdp24
bds-database-install: hadoop version            : hadoop-2.7.1.2.4.0.0-169
bds-database-install: hive version              : apache-hive-1.2.1000.2.4.0.0-169-
bin
bds-database-install: hbase version             : hbase-1.1.2.2.4.0.0-169
bds-database-install: bds version               : Big Data SQL 3.1.0
bds-database-install: bds install date          : Wed Mar  1 01:00:47 2017
bds-database-install: bd_cell version           :
bd_cell-12.1.2.0.100_LINUX.X64_170227.1054-1.x86_64
bds-database-install: cell config dir           : /etc/oracle/cell/network-config
bds-database-install: hybrid cell model         : false
bds-database-install: cell network(s)           : 180.168.41.9/21
bds-database-install: allow multiple subnets    : _skgxp_ant_options=1
bds-database-install: use UDP protocol          : _skgxp_dynamic_protocol=2
bds-database-install: cellaffinity.ora file     : missing
bds-database-install: configured DB network     : 11.122.17.217/21
bds-database-install: listener end point        : EXTPROC1521
bds-database-install: db resource               : orcl
bds-database-install: database type             : SINGLE
bds-database-install: cardinality               : 1
bds-database-install: mta extproc sid           : bds_orcl_hdp24
bds-database-install: crs                       : true
bds-database-install: cdb                       : true
bds-database-install: action                    : uninstall primary
bds-database-uninstall: mta shutdown bds_orcl_hdp24
bds-database-uninstall: removing existing mta parameter settings
bds-database-uninstall: stopping crs resource bds_orcl_hdp24
```

```
bds-database-uninstall: deleting crs resource bds_orcl_hdp24
bds-database-uninstall: removing /u03/app/oracle/product/12.1.0/dbhome_1/hs/admin/
initbds_orcl_hdp24.ora
bds-database-uninstall: dropping mta related db links
catcon: ALL catcon-related output will be written to bds-database-install-6467-
destroymtalink_catcon_6955.lst
catcon: See bds-database-install-6467-destroymtalink*.log files for output generated
by scripts
catcon: See bds-database-install-6467-destroymtalink_*.lst files for spool files, if
any
catcon.pl: completed successfully
bds-database-uninstall: dropping mta related db directory
catcon: ALL catcon-related output will be written to bds-database-install-6467-
destroymtadirectory_catcon_6981.lst
catcon: See bds-database-install-6467-destroymtadirectory*.log files for output
generated by scripts
catcon: See bds-database-install-6467-destroymtadirectory_*.lst files for spool
files, if any
catcon.pl: completed successfully
bds-database-uninstall: dropping mta default db link and directory
catcon: ALL catcon-related output will be written to bds-database-install-6467-
uninstall_catcon_7006.lst
catcon: See bds-database-install-6467-uninstall*.log files for output generated by
scripts
catcon: See bds-database-install-6467-uninstall_*.lst files for spool files, if any
catcon.pl: completed successfully
bds-database-uninstall: removing bds jars
bds-database-uninstall: removing hadoop client hadoop-2.7.1.2.4.0.0-169
bds-database-uninstall: removing hive client apache-hive-1.2.1000.2.4.0.0-169-bin
bds-database-uninstall: removing hbase client hbase-1.1.2.2.4.0.0-169
bds-database-uninstall: removing bds configuration files
bds-database-uninstall: removing default_dir_orcl
bds-database-uninstall: removing jdk
bds-database-uninstall: removing client configuration files
bds-database-uninstall: uninstall successful for cluster HDP24

bds-database-uninstall: log files in /u03/app/oracle/product/12.1.0/dbhome_1/install
bds-database-uninstall: bds-database-install-6467-* files for more detailed info
bds-database-uninstall: full log in bds-database-install-6467-main-script.log
```

Using setup-bds reconfigure to Reconfigure Oracle Big Data SQL on an Oracle Big Data Appliance Cluster

This example shows a changeover from an InfiniBand connection to an Ethernet connection, but the method is the same for any configuration change. On the Hadoop cluster management server, make the desired changes to `bds-config.json`. Then run `/setup-bds reconfigure bds-config.json` to reconfigure the Hadoop side. Then use `bds-database-create-bundle.sh --reconfigure` to regenerate the database bundle. When finished, copy the new database bundle to each node of the database, unpack it and run the uninstall script. The script execution on the database side is not included in this example.

Commands:

```
# ./setup-bds reconfigure bds-config.json
# ./bds-database-create-bundle.sh --reconfigure
```

Example:

Note the use of SCL to specify Python 2.7 in place of the default Python. SCL is available on Oracle Big Data Appliance and may or may not be available on other Hadoop systems.

You need the password to the admin account on the cluster management server.

```
[root@mydomain21mynode18 BDSSetup]# scl enable python27 "./setup-bds reconfigure bds-
config.json"
BigDataSQL: FQDN: mydomain21mynode18.us.mycompany.com.
BigDataSQL: Hostname: mydomain21mynode18.us.mycompany.com.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/cloudera.
BigDataSQL: Verifying configuration parameters...
BigDataSQL: Starting server...
BigDataSQL: API protocol: https.
BigDataSQL: API port: 7183.
BigDataSQL: Waiting for server to respond...

Admin User: admin
Password:

BigDataSQL: Authentication succeeded.
BigDataSQL: API version: 15.
BigDataSQL: Cluster type: BDA.
BigDataSQL: Retrieving clusters...
BigDataSQL: One cluster detected.
BigDataSQL: Cluster name: mycluster.
BigDataSQL: Cluster version: CDH5.
BigDataSQL: Setting target cluster...
BigDataSQL: Service: BIGDATASQL.
BigDataSQL: Looking for configured port on the JSON config file...
BigDataSQL: Testing port...
BigDataSQL: Using port: 81.
BigDataSQL: Local directories at /root/bds/BDSSetup/deployment_manager/
bigdata_platforms/cloudera/services/bigdatasql.
BigDataSQL: Verifying service configuration...
BigDataSQL: Service TCP connection: False.
BigDataSQL: Setting up related services...
BigDataSQL: Setting up database...
BigDataSQL: Setting up memory...
BigDataSQL: Loading service state file...
BigDataSQL: Reconfigure service...
BigDataSQL: Reconfigure bigdatasql service...
BigDataSQL: Restarting bigdatasql service...
BigDataSQL: Creating database configuration...
BigDataSQL: Downloading file from https://mydomain21mynode18.us.mycompany.com:
7183/cmf/command/1717/download.
Progress [5424Kb]
BigDataSQL: Download complete, file: /var/log/bigdatasql/DM/inspection.json.
BigDataSQL: Cleaning up service...
BigDataSQL: RECONFIGURE workflow completed.
```

```
[root@mydomain21mynode18 BDSSetup]# cd db
[root@mydomain21mynode18 db]# ./bds-database-create-bundle.sh --reconfigure
bds-database-create-bundle: database bundle creation script started at : Tue Feb 28
22:41:13 PST 2017
```

```
bds-database-create-bundle: working directory         : /root/bds/BDSSetup
bds-database-create-bundle: Hadoop client download url : http://archive-
primary.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-cdh5.10.0.tar.gz
bds-database-create-bundle: Hive client download url   : http://archive-
primary.cloudera.com/cdh5/cdh/5/hive-1.1.0-cdh5.10.0.tar.gz
bds-database-create-bundle: HBase client download url : http://archive-
primary.cloudera.com/cdh5/cdh/5/hbase-1.2.0-cdh5.10.0.tar.gz
bds-database-create-bundle: copying required files

Management server credentials for download configuration files
Userid  : admin
Password:

bds-database-create-bundle: downloading yarn configuration file
bds-database-create-bundle: downloading hive configuration file
bds-database-create-bundle: building bundles files
bds-database-create-bundle: database bundle creation script completed all steps
```

B

Oracle Big Data SQL Licensing

The licensing for Oracle Big Data SQL is separate from the licensing for other Oracle products.

When you purchase a license for Oracle Big Data SQL, note the following:

- A separate license must be procured per disk per Hadoop cluster.
- All nodes within the Hadoop cluster that runs Oracle Big Data SQL must be licensed.
- All disks within every node that is part of a cluster running Oracle Big Data SQL must be licensed. Partial licensing within a node is not available. All nodes in the cluster are included.
- Only the Hadoop cluster side (Oracle Big Data Appliance, or other) of an Oracle Big Data SQL installation is licensed and no additional license is required for the database server side.
- Oracle Copy to Hadoop licensing is included.
- Oracle Super Cluster is not included.

Third Party Licensing for Oracle Shell for Hadoop Loaders

Oracle Shell for Hadoop Loaders is included in the Oracle Big Data SQL installation bundle. The following are third-party projects contained in Oracle Shell for Hadoop Loaders.

- ANTLR 4.5.3
- Apache Commons Exec 1.3

Unless otherwise specifically noted, or as required under the terms of the third party license (e.g., LGPL), the licenses and statements herein, including all statements regarding Apache-licensed code, are intended as notices only.

B.1 ANTLR 4.5.3

Copyright (c) 2015 Terence Parr, Sam Harwell

All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

B.2 Apache Commons Exec 1.3

Include the following License ONLY ONCE in the documentation even if there are multiple products licensed under the license.

The following applies to all products licensed under the Apache 2.0 License:

You may not use the identified files except in compliance with the Apache License, Version 2.0 (the "License.").

You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. A copy of the license is also reproduced below.

B.3 Apache Licensed Code

The following is included as a notice in compliance with the terms of the Apache 2.0 License, and applies to all programs licensed under the Apache 2.0 license:

You may not use the identified files except in compliance with the Apache License, Version 2.0 (the "License.")

You may obtain a copy of the License at

`http://www.apache.org/licenses/LICENSE-2.0`

A copy of the license is also reproduced below.

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

B.4 Apache License

Version 2.0, January 2004

`http://www.apache.org/licenses/`

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions**

   "License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

   "Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

   "Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

   "You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

   "Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

   "Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

   "Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

   "Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

   "Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

   "Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-

charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution**. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   a. You must give any other recipients of the Work or Derivative Works a copy of this License; and

   b. You must cause any modified files to carry prominent notices stating that You changed the files; and

   c. You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   d. If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions**. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks**. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. **Disclaimer of Warranty**. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on

an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability**. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.
9. **Accepting Warranty or Additional Liability**. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

`http://www.apache.org/licenses/LICENSE-2.0`

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

This product includes software developed by The Apache Software Foundation (**http://www.apache.org/**) (listed below):

Index