Oracle® Big Data SQL

Installation Guide

Release 4.1.2

F38621-01

June 2022

Oracle Big Data SQL Installation Guide, Release 4.1.2

F38621-01

Primary Author: Drue Swadener, Frederick Kush, Lauran Serhal, Ben Gelernter

Contents

1 Introduction

2 Installing or Upgrading the Hadoop Side of Oracle Big Data SQL

3 Installing or Upgrading the Oracle Database Side of Oracle Big Data SQL

4 Post-Installation Tasks

5 Expanding or Shrinking an Installation

6 Reconfiguring an Installation

7 Uninstalling Oracle Big Data SQL

8 Securing the Oracle Big Data SQL Installation

9 Additional Tools Installed

A Installation Quick Reference

B bds-config.json Configuration Examples

C Oracle Big Data SQL Installation Example

D Downloading the Correct Versions of the Hadoop, Hive, and HBase Clients for a Local Repostory

E Oracle Big Data SQL Licensing

Index

Preface

This guide shows you how to install and configure Oracle Big Data SQL.

Audience

This guide is intended for administrators 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 basic knowledge of Oracle Database single-node and multinode systems, the Hadoop framework, the Linux operating system, and networking concepts.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

See the *Oracle Big Data SQL User’s Guide* for instructions on using this product after installation.

The following are Oracle products that are often used along with Oracle Big Data SQL:

- Oracle Big Data Appliance
- Oracle Big Data Connectors
- Oracle SQL Developer

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 this Oracle Big Data SQL Release

See Changes in This Release in the *Oracle Big Data SQL User's Guide* to learn what's new in this release and in previous releases.

1 Introduction

This guide describes how to install Oracle Big Data SQL, how to reconfigure or extend the installation to accommodate changes in the environment, and, if necessary, how to uninstall the software.

1.1 About Installation

The Oracle Big Data SQL installation is done in phases.

The first two phases are:

- Installation on the node of the Hadoop cluster where the cluster management server is running.
- Installation on each node of the Oracle Database system.
- (Optional) Activate security features if you have chosen to enable them.

The Hadoop cluster and Oracle Database system must be networked together via Ethernet or InfiniBand. (Connectivity to Oracle SuperCluster is InfiniBand only).

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

The installation process starts on the Hadoop system, where you install the software manually on one node only (the node running the cluster management software). Oracle Big Data SQL leverages the administration facilities of the cluster management software to automatically propagate the installation to all DataNodes in the cluster.

The package that you install on the Hadoop side also generates an Oracle Big Data SQL installation package for your Oracle Database system. After the Hadoop-side installation is complete, copy this package to all nodes of the Oracle Database system, unpack it, and install it using the instructions in this guide. If you have enabled Database Authentication or Hadoop Secure Impersonation, you then perform the third installation step.

1.2 Supported System Combinations

Oracle Big Data SQL supports connectivity between a number of Oracle Engineered Systems and commodity servers.

The current release supports Oracle Big Data SQL connectivity for the following Oracle Database platforms/Hadoop system combinations:

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

> **Note:**
>
> The phrase “Oracle Database on commodity systems” refers to Oracle Database hosts that are not the Oracle Exadata Database Machine. Commodity database systems may be either Oracle Linux or RHEL-based. “Commodity Hadoop systems” refers to Hortonworks HDP systems and to Cloudera CDH-based systems other than Oracle Big Data Appliance.

1.3 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.

1.4 Installing on Oracle Big Data Appliance

Each Oracle Big Data Appliance software release already includes a version of Oracle Big Data SQL that is ready to install, using the utilities available on the appliance.

You can download and install the standalone Big Data SQL bundle as described in this guide on all supported Hadoop platforms, including Big Data appliance. But for Big Data Appliance, the recommended method is to install the Big Data SQL package included with your Big Data Appliance software. The instructions for doing this are in the *Oracle Big Data Appliance Owner's Guide*. You can find them in the same location in most versions of the Owner's Guide. For example, Big Data Appliance 5.1 and 5.2 include Big Data SQL 4.0 (not 4.1) and the instructions are here: 10.9.5 Installing Oracle Big Data SQL.

The advantages of installing the version of Big Data SQL included with the appliance are:

- The prerequisites to the installation are already met.
- You can add Big Data SQL to the Big Data Appliance release installation by checking a checkbox in the Big Data Appliance Configuration Generation Utillity. The Mammoth utility will then automatically include Big Data SQL in the installation.
- You can also install Big Data SQL later, using the bdacli utility. This is also a simple procedure. The command is `bdacli enable big_data_sql`.
- When Big Data SQL is installed by the Mammoth utility, then during an upgrade to a newer Big Data Appliance software release, Mammoth will automatically upgrade the Hadoop side of the Big Data SQL installation to the version included in the release bundle.

The limitations of installing the version of Big Data SQL include with Big Data Appliance are:

- The installation is performed for the Hadoop side only. You still need to install the database side of the product using the instructions in this guide. You also must refer to this guide if you want to modify the default installation.
- The Big Data Appliance release may not include the latest available version of Big Data SQL.

> **Note:**
>
> If you choose to download and install a release of Big Data SQL from the Oracle Software Delivery Cloud instead of installing the version included with Big Data Appliance, then first check the Oracle Big Data SQL Master Compatibility Matrix to confirm that your current Big Data Appliance release level supports the version that you want to install.

1.5 Prerequisites for Networking

The Oracle Big Data SQL installation has the following network dependencies.

1.5.1 Port Access Requirements

Oracle Big Data SQL requires that the following ports are open though firewalls protecting the Hadoop cluster and Oracle Database.

Table 1-1 Ports That Must be Open on Both the Hadoop Cluster and Oracle Database Servers

Port	Use
Ephemeral_range, i.e. 9000-65500	UDP communication from the celliniteth.ora IP address
5042	Diskmon

Table 1-2 Additional Ports That Must Be Open on the Hadoop Cluster

Hadoop Cluster Ports	Where	Use
50010	All nodes on unsecured clusters	dfs.datanode.address
1004	All nodes on secured clusters	dfs.datanode.address
50020	All nodes	dfs.datanode.ipc.address
8020	NameNodes	fs.defaultFS
8022	NameNodes	dfs.namenode.servicerpc-address
9083	Hive Metastore & HiveServer2 node.	hive.metastore
10000	Hive Metastore & HiveServer2 node.	hive.server2.thrift.port
88	Kerberos KDC	TCP & UDP
16000	Where HDFS Encryption is enabled	KMS HTTP Port

1.6 Prerequisites for Installation on the Hadoop Cluster

The following installed software package active services, tools, and environment settings 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 are not listed here. See the Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support) for this information.

The Oracle Big Data SQL installer checks all prerequisites before beginning the installation and reports any missing requirements on each node.

> **Tip:**
>
> *Use `bds_node_check.sh` to pre-check whether or not the DataNodes of the cluster are ready for the installation.*
> You can manually check for them, but the easiest way is to run `bds_node_check.sh` on each node. This script returns a complete readiness report. After you download the installation bundle, unzip it, and execute the run file, `bds_node_check.sh` will be available, along with the tools to perform the installation. See Check Hadoop-Side Prerequisites for details.

Note:

- Oracle Big Data SQL 4.1 does not support single user mode for Cloudera clusters.
- The JDK is no longer a prerequisite. JDK 8u171 is included with this release of Oracle Big Data SQL.

1.6.1 Software Package Requirements for all DataNodes

The following packages must be pre-installed on all Hadoop cluster nodes before installing Oracle Big Data SQL. These are already installed on releases of Oracle Big Data Appliance that support Oracle Big Data SQL 4.1.2. Several additional packages are required if Query Server will be installed.

```
libaio
dmidecode
net-snmp
net-snmp-utils
glibc
libgcc
libcgroup-tools (Oracle Linux 7 only)
libstdc++
libuuid
ntp
perl
perl-libwww-perl
perl-libxml-perl
perl-XML-LibXML
perl-Time-HiRes
perl-XML-SAX
perl-Env (Oracle Linux 7 only)
rpm
curl
unzip
zip
tar
uname
```

The following packages are required only if you install Query Server:

```
expect
procmail
```

The yum utility is the recommended method for installing these packages. All of them can be installed with a single yum command. For example (not including expect and procmail):

```
# yum -y install dmidecode net-snmp net-snmp-utils glibc libgcc libcgroup-
tools libstdc++ libuuid ntp perl perl-libs perl-Time-HiRes perl-libwww-perl
```

```
perl-libxml-perl perl-XML-LibXML perl-XML-SAX perl-Env fuse fuse-libs
rpm curl unzip zip tar uname libaio gcc
```

Special Prequisites for the Configuration Management Server

On the node where CM or Ambari runs (usually Node 3 on Oracle Big Data Appliance), you may also need to install a compatible version of Python as well as the Python Cryptography package. See the next section to determine whether or not this is necessary. If you do need to manually install a version of Python, then add openssl-devel to the yum parameter string:

Other Prequisites

- HDFS, YARN, and Hive must be running on the cluster at Oracle Big Data SQL installation time and runtime. They can be installed as parcels or packages on Cloudera CDH and as stacks on Hortonworks HDP.
- On CDH, if you install the Hadoop services required by Oracle Big Data SQL as packages, be sure that they are installed from within CM. Otherwise, CM will not be able to manage them. This is not an issue with parcel-based installation.

1.6.2 Python Requirements for the Cluster Management Node

On the node where the CM or Ambari cluster management service is running, the Oracle Big Data SQL installer requires Python 2.7.5 or greater, but less that 3.0. You must also add the Python Cryptography package to this Python installation if it is not present.

Jaguar, the Oracle Big Data SQL installer, requires Python (>= 2.7.5 <3.0) locally on the node where you run the installer. This is the node where CM or Ambari cluster management service is running. If any installation of Python in this supported version range is already present, you can use it to run Jaguar.

- On Oracle Big Data Appliance or commodity Hadooop clusters running Oracle Linux 6 or 7:
  Do not manually install Python to support the Jaguar installer. There is a compatible Python package already available on the appliance and the Jaguar installer will automatically find and use this package without prompting you.
- On commodity Hadoop clusters running Oracle Linux 6:
  Install a compatible version of Python if not present.
- On Oracle Big Data Appliance or commodity Hadooop clusters running Oracle Linux 5:
  Install a compatible version of Python if not present. On Oracle Big Data Appliance, install it as secondary installation only.

> **Important:**
>
> On Oracle Big Data Appliance do not overwrite the default Python installation with a newer version or switch the default to a newer version. This restriction may also apply other supported Hadoop platforms. Consult the documentation for the CDH or HDP platform you are using.

On Oracle Linux 6 on a commmodity Hadoop platform, the Jaguar installer will prompt you for the path of the compatible Python installation.

Installing the Required Python Cryptography Module

You can use Python's pip utility to install the Python Cryptography module. Use scl if Python (>= 2.7.5 <3.0) is not the default. This example installs pip and then installs and imports the module.

```
# scl enable python27 "pip install -U pip"
# scl enable python27 "pip install cryptography"
# scl enable python27 "python -c 'import cryptography; print \"ok\";'"
```

You can then run the Jaguar installer.

1.6.2.1 Adding Python 2.7.5 or Greater as a Secondary Installation

Below is a procedure for adding the Python 2.7.5 or greater (but less than 3.0) as a secondary installation.

Note:

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

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

```
# 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/2.7.5
# make
# mkdir -p /usr/local/python/2.7.5
# make install
# export PATH=/usr/local/python/2.7.5/bin:$PATH
```

If you create a secondary installation of Python, it is strongly recommended that you apply Python update regularly to include new security fixes.

Important: On Oracle Big Data Appliance, do not update the mammoth-installed Python unless directed to do so by Oracle.

1.6.2.2 When You May Need to Use scl to Invoke the Correct Python Version

If there is more than one Python release on the cluster managerment server, then be sure that Python 2.7.5 or greater (but less than 3.0) is invoked for any operations associated with this release of Oracle Big Data SQL.

If the scl utility is available, you can use to invoke Python 2.7.5 or greater explicitly. This is necessary if a different Python installation is the default. In that case, use scl or another method to invoked the correct Python version for scripts as well as Python-based utilities such as Jaguar, the Oracle Big Data SQL installer,

```
[root@myclusteradminserver:BDSjaguar] # scl enable python27 "./jaguar
install bds-config.json"
```

There is one exception to this requirement. On Oracle Big Data Appliance clusters running Oracle Linux 6 or Oracle Linux 7, it is not necessary to use scl explicitly in order to run the Jaguar installer. In this case, you can invoke Jaguar directly, as in:

```
[root@myclusteradminserver:BDSjaguar] # ./jaguar install bds-
config.json
```

Jaguar itself will silently invoke scl if it is available and if scl is required to invoke a compatible Python release in this environment.

Note that this only applies to Jaguar on Big Data Appliance. To run any other Python scripts required by Oracle Big Data SQL (even on Oracle Big Data Appliance), use scl if Python 2.7.5 is not the default.

For example, to install the required Python Cryptography package, you may need to invoke scl to ensure that you are using the correct version of Python:

```
# scl enable python27 pip install cryptography
```

1.6.3 Environment Settings

The following environment settings are required prior to the installation.

- ntp enabled
- Minimum ratio of `shmmax` to `shmall`:
  `shmmax` = `shmall` * `PAGE_SIZE`
- `shmmax` must be greater that physical memory.
- `swappiness` set between 5 and 25.
- All `*.rp_filter` instances disabled
- Socket buffer size equal to or greater than 4194304
- If the cell is running on a processor with Memory Protection Keys for Userspace (PKU) enabled, `vm.max_map_count` should be set to at least 100,000,000. You can check if running PKU with this command:

  ```
  cat /proc/cpuinfo | grep flags | grep pku | uniq
  ```

1.6.4 Proxy-Related Settings

The installation process requires Internet access in order to download some packages from Cloudera or Hortonworks sites.

If a proxy is required for Internet access, then either ensure that the following are set as Linux environment variables, or, enable the equivalent parameters in the Jaguar configuration file, `bds-config.json`)

- `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.

> **See Also:**
>
> Configuration Parameters for bds-config.json describes the use of `http_proxy` , `https_proxy`, and other parameters in the installer configuration file.

1.6.5 CPU, Memory, and Networking Requirements

Oracle Big Data SQL requires the following.

Minimum CPU and Memory for Each Node

- 8 CPU cores
- 32 GB RAM
- 64 GB RAM depending on the Oracle database versions you intend to support (12.1, 12.2, 18c, 19c). If you set the `database_compatibility` parameter in the Jaguar Configuration Parameter and Command Reference to full, with simultaneous support for 12.1 and 12.2 and above, then 64 GB are required. However, if you only need 1 offloader, 32 GB is sufficient.

Networking

If Hadoop traffic is over VLANs, all DataNodes must be on the same VLAN.

1.7 Prerequisites for Installation on Oracle Database Nodes

Installation prerequisites vary, depending on type of Hadoop system and Oracle Database system where Oracle Big Data SQL will be installed.

Patch Level

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.

Before you begin the installation, review the additional environmental and user access requirements described below.

Packages Required for Kerberos

If you are installing on a Kerberos-enabled Oracle Database System, these package must be pre-installed:

- krb5-workstation
- krb5-libs

rdma-core and ibverbs Packages

rdma-core and ibverbs packages are only required for Exadata. If you have a problem bringing up a non-Exadata database due to a diskmon failure with error messages related to either packages, you should remove the packages.

Required Environment Variables

The following are always required. Be sure that these environment variables are set correctly.

- ORACLE_SID
- ORACLE_HOME

> **Note:**
>
> GI_HOME (which was required in Oracle Big Data SQL 3.1 and earlier) is no longer required.

Required Credentials

- Oracle Database owner credentials (The owner is usually the `oracle` Linux account.)

  Big Data SQL is installed as an add-on to Oracle Database. Tasks related directly to database instance are performed through database owner account (`oracle` or other).
- Grid user credentials

  In some cases where Grid infrastructure is present, it must be restarted. If the system uses Grid then you should have the Grid user credentials on hand in case a restart is required.

The Linux users `grid` and `oracle` (or other database owner) must both be in the same group (usually `oinstall`). This user requires permission to read all files owned by the `grid` user and vice versa.

All Oracle Big Data SQL files and directories are owned by the `oracle:oinstall` user and group.

Required Grid Infrastructure Patches

Check the Grid Infrastructure to make sure all patches required by the Oracle Big Data Installation have been installed. Go to the Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support) to find up-to-date information on patch requirements.

1.8 Downloading Oracle Big Data SQL and Query Server

You can download Oracle Big Data SQL from the Oracle Software Delivery Cloud (also known as eDelivery).

Follow these steps to download Oracle Big Data SQL and prepare for installation:

1. **Download Oracle Big Data SQL**
   Always download and install the latest version of Big Data SQL. This will provide the best installation experience and offer compatibility with all supported Oracle Databases.

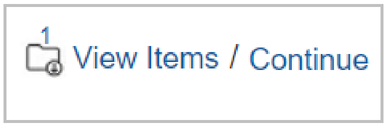
   There are three files to download for Oracle Big Data SQL 4.1.2:

   - The primary BDSJaguar bundle, which contains the Jaguar installer for Oracle Big Data SQL:
     - `Vnnnnnn-01.zip`
   - The two parts of the optional Query Server bundle. If you want to use Query Server, then download the following two parts of the Query Server bundle in addition to the primary BDSJaguar bundle:
     - `Vnnnnnn-01_1of2.zip`
     - `Vnnnnnn-01_2of2.zip`

   a. Sign in to Oracle Software Delivery Cloud.

   b. Search for **Oracle Big Data SQL**.
   A list of Oracle Big Data SQL versions to download appears.

   c. Click **Select** for DLP: Oracle Big Data SQL 4.1.2 (Oracle Big Data SQL) .
   The Download Queue now shows an entry to download. The View Items icon displays the number of items.

   d. Click **Continue**.
   The actual version available may be greater than 4.1.2. *n.n.n*. The same bundle is compatible with all supported Hadoop clusters.

   e. Click **Continue**.
   The Download Queue displays showing your selected downloads.

   f. Click **Continue**.

Oracle Standard Terms and Restrictions displays.

g. Accept the license agreement.
The list of downloadable bundles is displayed.

h. Select all three bundles, and click **Download**.

2. **Prepare for installation**

a. Copy the Installer Bundle to the Hadoop node that hosts the cluster management server (CDH or Ambari). On Oracle Big Data Appliance this is usually Node3. Choose any location. If you intend to use Query Server, copy the Query Server zips to the same location.

b. Log on as root and unzip the Installer Bundle.
You will see that the Release 4.1.2 Installer Bundle contains only the run file.

```
# unzip Vnnnnnn-01.zip
Archive:  Vnnnnnn-01.zip
  inflating: BDSJaguar-4.1.2.run
  inflating: readme
```

c. Before executing the run file, decide if you want to keep the default extraction target for the installation and configuration files. The default is `/opt/oracle`. If not, then you can change it by setting the `JAGUAR_ROOT` environment variable.

```
# export JAGUAR_ROOT=<my_directory>
```

Throughout this guide, the placeholder *`Big Data SQL Install Directory`* refers to the `JAGUAR_ROOT` where you extracted the files.

> **Important:**
>
> This is the permanent working directory from which you configure and install Oracle Big Data SQL. You will also need the tools in this directory post installation. It is strongly recommended that you secure this directory against accidental or unauthorized modification or deletion. The primary file to protect is your installation configuration file (by default, `bds-config.json`). As you customize the configuration to your needs, this file becomes the record of the state of the installation. It is useful for recovery purposes and as a basis for further changes.

d. Execute the run file.

```
# ./BDSJaguar-4.1.2.run
BDSJaguar-4.1.2.run: platform is: Linux
BDSJaguar-4.1.2.run: Jaguar directory created successfull
BDSJaguar-4.1.2.run: Based on features selected in config.json
file, extra bundles could be required
BDSJaguar-4.1.2.run: Please go to /opt/oracle/BDSJaguar
```

3. **(Optional) Include Query Server**

If you want to include Query Server in the installation, you must unzip both Query Server downloads to extract both parts of the bundle, then run `join.sh` to assemble into one Query Server bundle.

> **Note:**
>
> You cannot use or install Query Server separately from Oracle Big Data SQL. It can be included in the Jaguar-driven installation as described below.

a. Unzip `Vnnnnnn-01_1of2.zip` to extract the first part of the bundle.

```
$ unzip -j -o Vnnnnnn-01_1of2.zip
 Archive: Vnnnnnn-01_1of2.zip
 inflating: BDSQLQS82d323d472f5c4666e1a7e48cd2d75b9-00
 inflating: join.sh
 inflating: readme.1st
```

b. Unzip `Vnnnnnn-01_2of2.zip` to extract the second part of the bundle.

```
$ unzip -j -o Vnnnnnn-01_2of2.zip
 Archive: Vnnnnnn-01_2of2.zip
 inflating: BDSQLQS82d323d472f5c4666e1a7e48cd2d75b9-01
 inflating: join.sh
 inflating: readme.1st
```

c. Run the `join.sh` script from either zip file to assemble the bundle.

```
$ ./join.sh
 Re-assembling Big Data SQL Query Server bundle
 Detected files:
 BDSQLQS82d323d472f5c4666e1a7e48cd2d75b9-00
 BDSQLQS82d323d472f5c4666e1a7e48cd2d75b9-01
 Joining 2 files
 BigDataSQL-4.1.2-QueryServer.zip successfully created !!!
```

d. Unzip the newly created bundle to extract the QueryServer run file.

```
# unzip BDSExtra-4.1.2-QueryServer.zip
...
# ./BDSExtra-4.1.2-QueryServer.run
```

e. Execute the `BDSExtra-4.1.2-QueryServer.run` script to include Query Server in the Big Data SQL installation.

> **Note:**
>
> To include Query Server in the Big Data SQL installation, be sure to execute this extra run file before running the Jaguar installer.

1.9 Upgrading From a Prior Release of Oracle Big Data SQL

On the Oracle Database side, Oracle Big Data SQL can now be installed over a previous release with no need to remove the older software. The install script automatically detects and upgrades an older version of the software.

Upgrading the Oracle Database Side of the Installation

On the database side, you need to perform the installation only once to upgrade the database side for any clusters connected to that particular database. This is because the installations on the database side are not entirely separate. They share the same set of Oracle Big Data SQL binaries. This results in a convenience for you – if you upgrade one installation on a database instance then you have effectively upgraded the database side of all installations on that database instance.

Upgrading the Hadoop Cluster Side of the Installation

If existing Oracle Big Data SQL installations on the Hadoop side are not upgraded, these installations will continue to work with the new Oracle Big Data SQL binaries on the database side, but will not have access to the new features in this release.

1.10 Important Terms and Concepts

These are special terms and concepts in the Oracle Big Data SQL installation.

Oracle Big Data SQL Installation Directory

On both the Hadoop side and database side of the installation, the directory where you unpack the installation bundle is not a temporary directory which you can delete after running the installer. These directories are staging areas for any future changes to the configuration. You should not delete them and may want to secure them against accidental deletion.

Database Authentication Keys

Database Authentication uses a key that must be identical on both sides of the installation (the Hadoop cluster and Oracle Database). The first part of the key is created on the cluster side and stored in the `.reqkey` file. This file is consumed only once on the database side, to connect the first Hadoop cluster to the database. Subsequent cluster installations use the configured key and the `.reqkey` file is no longer required. The full key (which is completed on the database side) is stored in an `.ackkey` file. This key is included in the part of the ZIP file created by the database-side installation and must be copied back to the Hadoop cluster by the user.

Request Key

By default, the Database Authentication feature is enabled in the configuration. (You can disable it by setting the parameter `database_auth_enabled` to “false” in the configuration file.) When this setting is true, then the Jaguar `install` and `reconfigure` operations can generate a `request key` (stored in a file with the extension `.reqkey`). This key is part of a unique GUID-key pair used for Database Authorization. This GUID-key pair is generated during the database side of the installation. The Jaguar operation creates a request key if the command line includes the `--requestdb` command line parameter along with a single database name (or a comma separated list of names). In this example, the install operation creates three keys, one for each of three different databases:

```
# ./jaguar --requestdb orcl,testdb,proddb install
```

The operation creates the request keys files in the directory `<Oracle Big Data SQL install directory>/BDSJaguar/dbkeys`. In this example, Jaguar `install` would generate these request key files:

```
orcl.reqkey
testdb.reqkey
proddb.reqkey
```

Prior to the database side of the installation, you copy request key to the database node and into the path of the database-side installer, which at runtime generates the GUID-key pair.

Acknowledge Key

After you copy a request key into the database-side installation directory, then when you run the database-side Oracle Big Data SQL installer it generates a corresponding *acknowledge key* . The acknowledge key is the original request key, paired with a GUID. This key is stored in a file that is included in a ZIP archive along with other information that must be returned to the Hadoop cluster by the user. .

Database Request Operation (databasereq)

The Jaguar `databasereq` operation is “standalone” way to generate a request key. It lets you create one or more request keys without performing an `install` or `reconfigure` operation:

```
# ./jaguar --requestdb <database name list> databasereq {configuration file
| null}
```

Database Acknowledge ZIP File

If Database Authentication, or Hadoop Secure Impersonation is enabled for the configuration, then the database-side installer creates a ZIP bundle configuration information . If Database Authentication is enabled, this bundle includes the acknowledge key file. Information required for Hadoop Secure Impersonation is also included if that option was enabled. Copy this ZIP file back to`/opt/oracle/DM/databases/conf` on the Hadoop cluster management server for processing.

Database Acknowledge is a third phase of the installation and is performed only when any of the three security features cited above are enabled.

Database Acknowledge Operation (databaseack)

If you have opted to enable any or all of three new security features (Database Authentication, or Hadoop Secure Impersonation), then after copying the Database Acknowledge ZIP file back to the Hadoop cluster, run the Jaguar *Database Acknowledge* operation.

The setup process for these features is a “round trip” that starts on the Hadoop cluster management server, where you set the security directives in the configuration file and run Jaguar, to the Oracle Database system where you run the database-side installation, and back to the Hadoop cluster management server where you return a copy of the ZIP file generated by the database-side installation. The last step is when you run `databaseack`, the

Database Acknowledge operation described in the outline below. Database Acknowledge completes the setup of these security features.

Default Cluster

The *default cluster* is the first Oracle Big Data SQL connection installed on an Oracle Database. In this context, the term *default cluster* refers to the installation directory on the database node where the connection to the Hadoop cluster is established. It does not literally refer to the Hadoop cluster itself. Each connection between a Hadoop cluster and a database has its own installation directory on the database node.

An important aspect of the default cluster is that the setting for Hadoop Secure Impersonation in the default cluster determines that setting for all other cluster connections to a given database. If you run a Jaguar `reconfigure` operation some time after installation and use it to turn Hadoop Secure Impersonation in the default cluster on or off, this change is effective for *all* clusters associated with the database.

If you perform installations to add additional clusters, the first cluster remains the default. If the default cluster is uninstalled, then next one (in chronological order of installation) becomes the default.

1.11 Installation Overview

The Oracle Big Data SQL software must be installed on all Hadoop cluster DataNodes 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.

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 Oracle Big Data SQL cell settings on the Grid nodes. See Potential Requirement to Restart Grid Infrastructure for details.

If a Previous Version of Oracle Big Data SQL is Already Installed

On commodity Hadoop systems (those other than Oracle Big Data Appliance) the installer automatically uninstalls any previous release from the Hadoop cluster.

You can install Oracle Big Data SQL on all supported Oracle Database systems without uninstalling a previous version.

Before installing this Oracle Big Data SQL release on Oracle Big Data Appliance, you must use bdacli to manually uninstall the older version if it had been enabled via bdacli or Mammoth. If you are not sure, try `bdacli disable big_data_sql`. If the disable comment fails, then the installation was likely done with the setup-bds installer. In that case, you can install the new version Oracle Big Data SQL without disabling the old version.

How Long Does It Take?

The table below estimates the time required for each phase of the installation. Actual times will vary.

Table 1-3 Installation Time Estimates

Installation on the Hadoop Cluster	Installation on Oracle Database Nodes
Eight minutes to 28 minutes The Hadoop side installation may take eight minute if all resources are locally available. An additional 20 minutes or more may be required if resources must be downloaded from the Internet.	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.

The installation process on Hadoop side includes installation on the Hadoop cluster as well as generation of the bundle for the second phase of the installation on the Oracle Database side. The database bundle includes Hadoop and Hive clients and other software. The Hadoop and Hive client software enable Oracle Database to communicate with HDFS and the Hive Metastore. The client software is specific to the version of the Hadoop distribution (i.e. Cloudera or Hortonworks). As explained later in this guide, you can download these packages prior to the installation, set up an URL or repository within your network, and make that target available to the installation script. If instead you let the installer download them from the Internet, the extra time for the installation depends upon your Internet download speed.

Pre-installation Steps

- Check to be sure that the Hadoop cluster and the Oracle Database system both meet all of the prerequisites for installation. On the database side, this includes confirming that all of the required patches are in installed. Check against these sources:
  - Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support)
  - Sections 2.1 in this guide, which identifies the prerequisites for installing on the Hadoop cluster. Also see Section 3.1, which describes the prerequisites for installing the Oracle Database system component of Oracle Big Data SQL.

  Oracle Big Data Appliance already meets all prerequisites.
- Have these login credentials available:
  - `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.

- On the cluster management server (where CM or Ambari is running), download the Oracle Big Data SQL installation bundle and unzip it into a permanent location of your choice. (See Downloading Oracle Big Data SQL and Query Server.)

Outline of the Installation Steps

This is an overview to familiarize you with the process.

- Phase 1: (Required) Perform the Hadoop cluster-side installation.
- Phase 2: (Required) Perform the database-side installation.
- Phase 3: (Optional) Database Acknowledge phase - only needed if Database Authentication or Hadoop Secure Impersonation is enabled.

For complete instructions, see Installing or Upgrading the Hadoop Side of Oracle Big Data SQL and Installing or Upgrading the Oracle Database Side of Oracle Big Data SQL.

1. **Start the Hadoop-Side Installation**

   Review the installation parameter options described in Chapter 2. The installation on the Hadoop side is where you make all of the decisions about how to configure Oracle Big Data SQL, including those that affect the Oracle Database side of the installation.

2. Edit the `bds-config.json file` provided with the bundle in order to configure the Jaguar installer as appropriate for your environment. You could also create your own configuration file using the same parameters.

3. Run the installer to perform the Hadoop-side installation as described in Installing or Upgrading the Hadoop Side of Oracle Big Data SQL.

   If the Database Authentication feature is enabled, then Jaguar must also output a “request key” (`.reqkey`) file for each database that will connect to the Hadoop cluster. You generate this file by including the `--requestdb` parameter in the Jaguar `install` command (the recommended way). You can also generate the file later with other Jaguar operations that support the `--requestdb`.

   This file contains one half of a GUID-key pair that is used in Database Authentication. The steps to create and install the key are explained in more detail in the installation steps.

4. Copy the database-side installation bundle to any temporary directory on each Oracle Database compute node.

5. If a request key file was generated, copy over that file to the same directory.

6. **Start the Database-Side Installation**

   Log on to the database server as the database owner Unzip bundle and execute the run file it contained. The run file does not install the software. It sets up an installation directory under `$ORACLE_HOME`.

7. As the database owner, perform the Oracle Database server-side installation. (See Installing or Upgrading the Oracle Database Side of Oracle Big Data SQL.)

   In this phase of the installation, you copy the database-side installation bundle to a temporary location on each compute node. If a `.reqkey` file was generated for the database, then copy the file into the installation directory before proceeding. Then run the `bds-database-install.sh` installation program.

   The database-side installer does the following:

- Copies the Oracle Big Data SQL binaries to the database node.
- Creates all database metadata and MTA extprocs (external processes) required to access the Hadoop cluster and configures the communication settings.

> **Important:**
>
> Be sure to install the bundle on each database compute node. The Hadoop-side installation automatically propagates the software to each node 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 a Grid restart may be needed. Be sure that you know the Grid password. If a Grid restart is required, then you will need the Grid credentials to complete the installation.

8. **If Applicable, Perform the “Database Acknowledge” Step**

   If Database Authentication or Hadoop Secure Impersonation were enabled, the database-side installation generates a ZIP file that you must copy back to Hadoop cluster management server. The file is generated in the installation directory under `$ORACLE_HOME` and has the following filename format.

   ```
   <Hadoop cluster name>-<Number nodes in the cluster>-<FQDN of the cluster
   management server node>-<FQDN of this database node>.zip
   ```

   Copy this file back to `/opt/oracle/DM/databases/conf` on the Hadoop cluster management server and then as `root` run the Database Acknowledge command from the BDSJaguar directory:

   ```
   # cd <Big Data SQL install directory>/BDSJaguar
   # ./jaguar databaseack <bds-config.json>
   ```

Workflow Diagrams

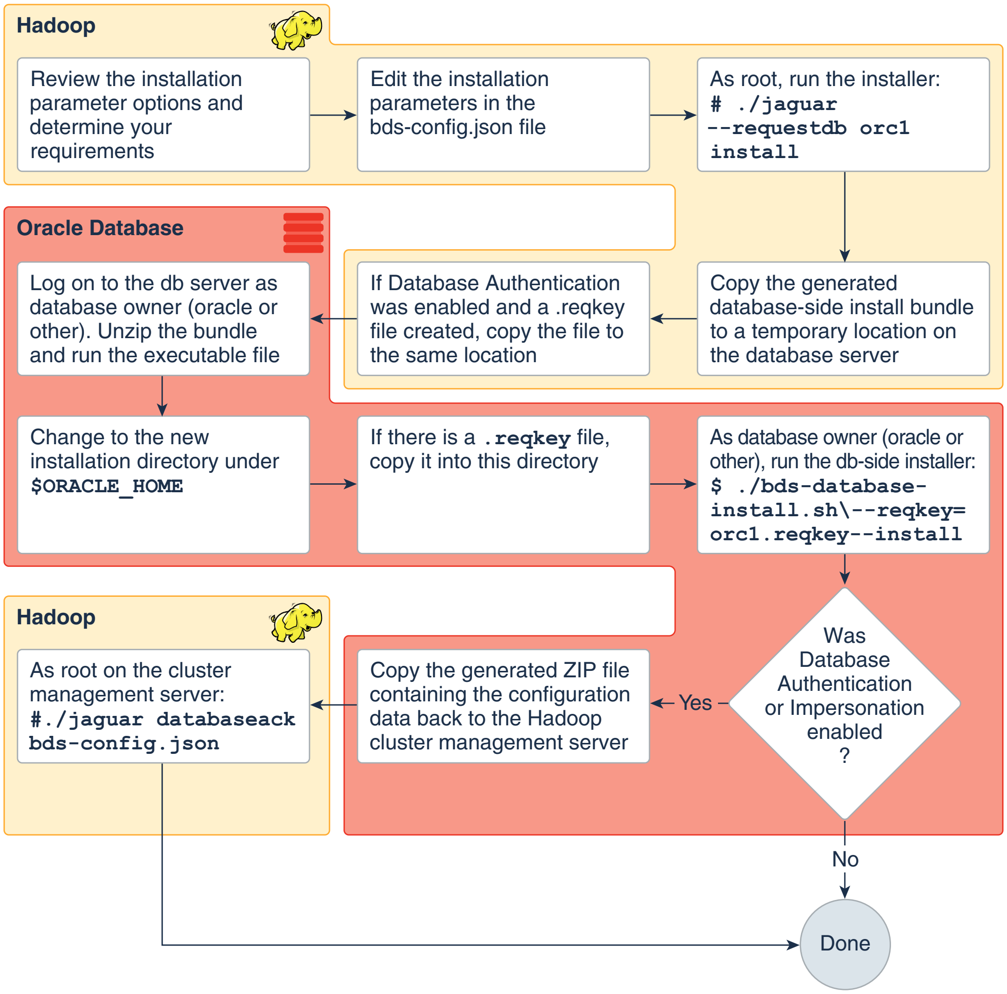
Complete Installation Workflow

The figure below illustrates the complete set of installation steps as described in this overview.

> **Note:**
>
> Before you start the steps shown in the workflow, be sure that both systems meet the installation prerequisites.

Figure 1-1 Installation Workflow

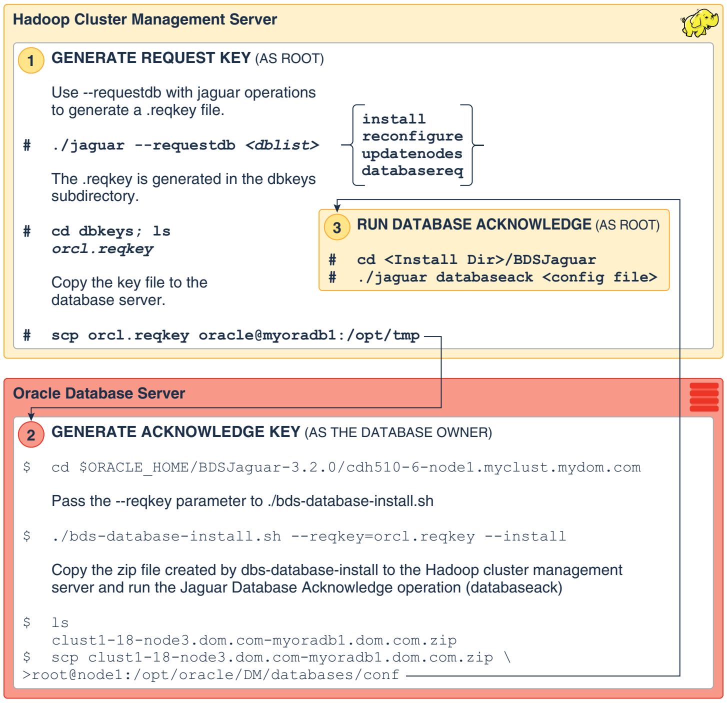
> **Note:**
>
> The --reqkey parameter in this diagram actually requires the full path to the file, as in `/bds-databse-install.sh --reqkey=/opt/tmp/orcl.reqkey`.


Hadoop
Review the installation parameter options and determine your requirements
Edit the installation parameters in the bds-config.json file
As root, run the installer: # ./jaguar --requestdb orc1 install
Oracle Database
Copy the generated database-side install bundle to a temporary location on the database server
If Database Authentication was enabled and a .reqkey file created, copy the file to the same location
Log on to the db server as database owner (oracle or other). Unzip the bundle and run the executable file
Change to the new installation directory under $ORACLE_HOME
If there is a .reqkey file, copy it into this directory
As database owner (oracle or other), run the db-side installer: $ ./bds-database-install.sh\--reqkey=orc1.reqkey--install
Was Database Authentication or Impersonation enabled ?
Yes
No
Copy the generated ZIP file containing the configuration data back to the Hadoop cluster management server
Hadoop
As root on the cluster management server: #./jaguar databaseack bds-config.json
Done

Key Generation and Installation

The figure below focuses on the three steps required to create and installing the GUID-key pair used in Database Authentication. The braces around parameters of the Jaguar command indicate that one of the operations in the list is required. Each of these operations supports use of the `--requestdb` parameter. Note that although `updatenodes` is included in this list, `updatenodes` is deprecated in this release. You should use `reconfigure` instead.

Figure 1-2 Generating and Installing the GUID-Key Pair for Database Authentication


Hadoop Cluster Management Server
1 GENERATE REQUEST KEY (AS ROOT)
3 RUN DATABASE ACKNOWLEDGE (AS ROOT)
Oracle Database Server
2 GENERATE ACKNOWLEDGE KEY (AS THE DATABASE OWNER)

Use --requestdb with jaguar operations to generate a .reqkey file.

```
#   ./jaguar --requestdb <dblist>
```

```
install
reconfigure
updatenodes
databasereq
```

The .reqkey is generated in the dbkeys subdirectory.

```
#   cd dbkeys; ls
    orcl.reqkey
```

Copy the key file to the database server.

```
#   scp orcl.reqkey oracle@myoradb1:/opt/tmp
```

```
$   cd $ORACLE_HOME/BDSJaguar-3.2.0/cdh510-6-node1.myclust.mydom.com
```

Pass the --reqkey parameter to ./bds-database-install.sh

```
$   ./bds-database-install.sh --reqkey=orcl.reqkey --install
```

Copy the zip file created by dbs-database-install to the Hadoop cluster management server and run the Jaguar Database Acknowledge operation (databaseack)

```
$   ls
    clust1-18-node3.dom.com-myoradb1.dom.com.zip
$   scp clust1-18-node3.dom.com-myoradb1.dom.com.zip \
>root@node1:/opt/oracle/DM/databases/conf
```

```
#   cd <Install Dir>/BDSJaguar
#   ./jaguar databaseack <config file>
```

1.12 Using the Installation Quick Reference

Once you are familiar with the functionality of the Jaguar utility on the Hadoop side and bds-database-install.sh on the Oracle Database side, you may find it useful to work from the Installation Quick Reference for subsequent installations. This reference provides an abbreviated description of the installation steps. It does not fully explain each step, so users should already have a working knowledge of the process. Links to relevant details in this and other documents are included.

2 Installing or Upgrading the Hadoop Side of Oracle Big Data SQL

After downloading the Oracle Big Data SQL deployment bundle and extracting the files, the next step is to configure the installer and then run the installation.

The installation of Oracle Big Data SQL on the Hadoop cluster is deployed using the services provided by the cluster management server (Cloudera Manager or Ambari). The Jaguar `install` command uses the management server API to register the BDS service and start the deployment task. From there, the management server controls the process that deploys the software to the nodes of the cluster and installs it.

The Hadoop-side installation also generates the deployment bundle for the database side of the installation.

If a previous version of Oracle Big Data SQL is already installed, Oracle Big Data SQL upgrades the installation to Release 4.1.2.

Users of previous Oracle Big Data SQL releases, please note that there are changes to the BDSJaguar configuration parameters available in this release.

- About Support for Multiple Database Versions (19c, 18c, 12.2, and 12.1)
- Before You Start the Hadoop-Side Installation
- About the Jaguar Utility
- Steps for Installing on the Hadoop Cluster

2.1 About Support for Multiple Database Versions (19c, 18c, 12.2, and 12.1)

Oracle Big Data SQL now supports Oracle Database 19c and also provides backward compatibility for Oracle Database 18c, 12.2, and 12.1.

You can use Oracle Big Data SQL 4.1.2 with any Oracle Database from release 12.1 to 19c. The database-related feature set available to you in Big Data SQL is determined by the Oracle Database version where it is installed. Each release of the database provides some advantages for Big Data SQL that its predecessors do not.

- Oracle Database 19c (first supported in Big Data SQL 4.1) provides the ability to create hybrid partitioned tables that can include data in CSV or Parquet files, and other formats accessible to tools in Spark, Hadoop, and other big data technologies. See Hybrid Partitioned Tables in the *Oracle Database VLDB and Partitioning Guide*.
  Another Oracle Database 19c new feature that is useful to Big Data SQL is In-Memory External Tables.

  In addition, an installation of Big Data SQL on an 19c database system has all of the functionality available to Big Data SQL on 18c databases.

- With Oracle Database 18c (which is supported by Oracle Big Data SQL 4.0 and later), you can access object stores in the cloud through the ORACLE_BIGDATA driver. 18c also enables Big Data SQL to perform aggregation offload, in which processing of aggregations in queries against data in Hadoop is pushed down to the Hadoop cluster.
- Oracle Database 12.1 and 12.2 are fully supported in this release. However, Big Data SQL installations on these databases do not enable you to leverage the newer capabilities that are available with 18c and 19c. With 12.1 and 12.2, Big Data SQL functionality is equivalent to Big Data SQL 3.2.1.1.

This backward compatibility enables you to install and administer release 4.1.2 in a mixed environment that includes both Oracle Database 19c, 18c, and 12c.

> **See Also:**
>
> The Jaguar Configuration Parameter and Command Reference in this chapter shows you how to configure support for Oracle Database versions when you install the Hadoop side of Big Data SQL.

2.2 Before You Start the Hadoop-Side Installation

Check to ensure all DataNodes of the cluster meet prerequisites.

- Check Hadoop-Side Prerequisites
- Check Memory Requirements
- Plan the Configuration and Edit the Jaguar Configuration File

2.2.1 Check Hadoop-Side Prerequisites

You can run `bds_node_check.sh` on all cluster DataNodes prior to installing Oracle Big Data SQL. This is a quick way to check if each node meets the installation criteria. You can see exactly what needs to be updated.

Running `bds_node_check.sh` is not required, but is recommended. The Jaguar installer runs the same pre-checks internally, but when Jaguar runs the pre-checks it also starts and stops the cluster management server. Furthermore, the installation stops in place when it encounters a node that does not meet the prerequisites. Each time this happens, you then need to fix the readiness errors on the node in order to continue. Running `bds_node_check.sh` as a first step contributes to a smoother installation. You can use this same script to check for the prerequisites when you add new nodes to the cluster.

Deploying and Running bds_node_check.sh

The script checks the local node where it is run. It does not check all nodes in the cluster.

1. Find the script on the cluster management server in the install directory created when you executed `./BDSJaguar-4.1.2.run`.

   ```
   $ ls <Big Data SQL Install Directory>
   BDSJaguar
   ```

```
bds_node_check.sh
$ cd <Big Data SQL Install Directory>
```

2. Use your preferred method to copy the script to a node that you want to check.

   ```
   $ scp bds_node_check.sh oracle@<node_IP_address>:/opt/tmp
   ```

3. Log on to the node and run the script.

   ```
   $ ./bds_node_check.sh
   ```

Checking for Missing Prerequisites in the bds_node_check.sh Output

The report returned by `bds_node_check.sh` inspects the node both for Jaguar installer prerequisites and prerequisites for support of communications with Query Server on its edge node. If you do not intend to install Query Server, you can ignore that subset of the prerequisites.

```
bds_node_check.sh: BDS version 4.1.2 (c) 2020 Oracle Corporation
bds_node_check.sh:
bds_node_check.sh: Starting pre-requirements checks for BDS Jaguar
bds_node_check.sh: Total memory 64240 >= 40960 correct
bds_node_check.sh: vm_overcommit_memory=0 correct
bds_node_check.sh: shmmax=4398046511104, shmall=1073741824, PAGE_SIZE=4096
correct
bds_node_check.sh: shmmax=4398046511104 >= total_memory=67360522240 + 1024
correct
bds_node_check.sh: swappiness=10 correct
bds_node_check.sh: Total cores 32 >= 8 correct
bds_node_check.sh: Size of socket buffer rmem_default 4194304 >= 4194304
correct
bds_node_check.sh: Size of socket buffer rmem_max 8388608 >= 4194304 correct
bds_node_check.sh: Size of socket buffer wmem_default 4194304 >= 4194304
correct
bds_node_check.sh: Size of socket buffer wmem_max 8388608 >= 4194304 correct
bds_node_check.sh: dmidecode installed
bds_node_check.sh: net-snmp installed
bds_node_check.sh: net-snmp-utils installed
bds_node_check.sh: perl-XML-SAX installed
bds_node_check.sh: perl-XML-LibXML installed
bds_node_check.sh: perl-libwww-perl installed
bds_node_check.sh: perl-libxml-perl installed
bds_node_check.sh: libaio installed
bds_node_check.sh: glibc installed
bds_node_check.sh: libgcc installed
bds_node_check.sh: libstdc++ installed
bds_node_check.sh: libuuid installed
bds_node_check.sh: perl-Time-HiRes installed
bds_node_check.sh: perl-libs installed
bds_node_check.sh: perl-Env installed
bds_node_check.sh: libcgroup-tools installed
bds_node_check.sh: rpm found
bds_node_check.sh: scp found
bds_node_check.sh: curl found
```

```
bds_node_check.sh: unzip found
bds_node_check.sh: zip found
bds_node_check.sh: tar found
bds_node_check.sh: uname found
bds_node_check.sh: perl found
bds_node_check.sh: cgget found
bds_node_check.sh:
bds_node_check.sh: Optionally, if this node will be running the Jaguar
installer,
bds_node_check.sh: it must have at least python version 2.7.5
bds_node_check.sh: with cryptography module available
bds_node_check.sh: Testing with /usr/bin/python
bds_node_check.sh: Python version 2.7.5, correct
bds_node_check.sh: Python cryptography module available, correct
bds_node_check.sh:
bds_node_check.sh: All pre-requirements were met for BDS Jaguar
bds_node_check.sh:
bds_node_check.sh: Starting pre-requirements checks for BDS Query
Server
bds_node_check.sh: Open files 131072 >= 131072 correct
bds_node_check.sh: expect installed
bds_node_check.sh: procmail not installed
bds_node_check.sh: oracle-database-preinstall-19c not installed
bds_node_check.sh: rpm found
bds_node_check.sh: scp found
bds_node_check.sh: curl found
bds_node_check.sh: unzip found
bds_node_check.sh: zip found
bds_node_check.sh: tar found
bds_node_check.sh: uname found
bds_node_check.sh: perl found
bds_node_check.sh: cgget found
bds_node_check.sh: No database instances running on this node, correct
bds_node_check.sh: /etc/oracle/olr.loc file does not exist
bds_node_check.sh: /etc/oracle/ocr.loc file does not exist
bds_node_check.sh:
bds_node_check.sh: 2 error(s) found for BDS Query Server pre-
requirements
```

2.2.2 Check Memory Requirements

Support for Oracle Database 12.1 may require additional memory.

Oracle Big Data SQL provides backward compatibility with Oracle Database 12.1 and 12.2 . However compatibility with Oracle Database 12.1 incurs an additional cost in memory. Oracle Database 12.1 support requires that the Hadoop nodes run an older offload server (in addtion to the offload server normally present). The overhead of running this additional offload server is a resource expense that you can prevent if you do not need to support Oracle Database 12.1.

If You Need to Support Oracle Database 12.1 for this Cluster:

Be sure that the DataNodes in the Hadoop cluster have enough memory. The minimum memory requirement per Hadoop node for an installation that supports full database compatibility, including 19c, 18c, 12.2, 12.1 is 64 GB.

Also check to be sure that the memory cgroup upper limit is set to allow Oracle Big Data SQL to consume this much memory.

If You do not Need to Support Oracle Database 12.1 for this Cluster:

Be sure to choose the right setting for the `database_compatibility` value in the Jaguar configuration file (`bds-config.json` or other). The options for this parameter are: `"12.1"`, `"12.2"`, `"18"`, `"19"`, and `"full"`. It's important to note that both `"12.1"` and `"full"` trigger the startup of the additional offload server to support 12.1.

> **See Also:**
>
> The Jaguar Configuration Parameter and Command Reference describes the `database_compatibility` parameter.

2.2.3 Plan the Configuration and Edit the Jaguar Configuration File

Before you start, consider the questions below.

Answering these questions will help clarify how you should edit the Jaguar configuration file. (See the Jaguar Configuration Parameter and Command Reference in this chapter.)

- *Do you plan to access data in object stores (S3, Azure, or Oracle Object Store)?*
  If so, then in the Jaguar configuration file you need to enable this access and also define some proxy settings that are specific to object store access.
- *Do you want to install the optional Query Server?*
  If so, several things are required before you run Jaguar in order to install Oracle Big Data SQL:
  - Identify a cluster edge node to host the optional Query Server.
    A dedicated node is strongly recommended. Query Server is resource-intensive and cannot run on node hosting either the DataNode or BDSSERVER roles.
  - Download and unzip the Query Server bundle and then execute the run file.
    The Query Server is in a separate deployment bundle from the Jaguar installer (`BDSExtra-4.1.2-QueryServer.zip).` Before running Jaguar, download and unzip this bundle from https://edelivery.oracle.com/. Then execute the Query Server run file. Also, in the configuration file submitted to Jaguar, define the two related parameters in the edgedb section of the file -- `node` and `enabled`.

    ```
       "edgedb":
    {
            "node"    : "<some server>.<some
    domain>.com",
            "enabled" :
    "true"
        }
    ```

- *Does your Hadoop environment need to connect to more than one Oracle Database version?*
  By default, Big Data SQL allows connections from 12.1, 12.2, 18c, and 19c databases. However, the offloaders that support these connections are resource-intensive, particularly for memory consumption on the nodes of the cluster. If you do not need to

support all three releases, you can save resource by turning off either the 12.1 or 12.2 offloader. (Note that the 12.2 offloader actually supports 12.2, 18c, and 19c.) You set this in the configuration passed to the Jaguar installer. For example, you can enter this string to allow connections from 12.1 databases only:

```
"database_compatibility" : [ "12.1" ]
```

If you specify "12.2", "18c", or "19c", the 12.1 offload is not enabled:

```
"database_compatibility" : [ "12.2" ]
```

- *Do you want to enable Database Authentication in order to validate connection requests from Oracle Database to Big Data SQL server processes on the Hadoop DataNodes?*

  Database Authentication in the network connection between the Oracle Database and Hadoop is set to `“true”` in the configuration by default. You have the option to disable it by setting the `database_auth_enabled` to `“false”`:

  ```
  "database_auth_enabled" : "false",
  ```

- *Do you want to use the Multi-User Authorization feature?*

  Multi-User Authorization enables you to grant users other than `oracle` permissions to run SQL queries against the Hadoop cluster. Multi-User Authorization can be used in conjunction with Sentry's role-based access control to provide improved control over user access.

  The first step in setting up Multi-User Authorization is to set these parameters in the security section of the configuration file:

  ```
  "impersonation_enabled" : "true",
  "impersonation_usehosts" : "true",
  "impersonation_blacklist" : "hdfs,hive"
  ```

  Note that you can add any account to the blacklist.

- *Are the Hadoop cluster and the Oracle Database system going to communicate over Ethernet or over InfiniBand? Also, do the Hadoop nodes have more than one network interface?*

  See the `use_infiniband` and `selection_subnet` parameters. (The `selection_subnet` does not apply to Oracle Big Data Appliance.)

  ```
  "use_infiniband" : "false",
  "selection_subnet" : "5.32.128.10/21"
  ```

  By default `use_infiniband` is set to false. Ethernet is the default protocol.

- *Are you going to pre-download the Hadoop and Hive client tarballs and set up a local repository or directory where the installer can acquire them, or, will you allow Jaguar to download them directly from the Cloudera or Hortonworks repository on the Internet (the default behavior)?*

  For Cloudera releases prior to 6.0, you can use the `url` or `dir` parameters in `BDS-config.json` (the Jaguar installer's configuraton file) to specify an arbitrary download location. If Internet access is via proxies, you can also set the `http_proxy` and `https_proxy` parameters in `BDS-config.json`.

> **Note:**
>
> On Big Data Appliance only, if you use the built-in Mammoth or bdacli utilities to install Big Data SQL the clients are automatically installed for you. However, if you use the Jaguar to install Big Data SQL on Big Data Appliance as well as other supported Hadoop platforms, you do have to provide the path in `BDS-config.json` if the location is other than the public repository.

For Big Data SQL on Cloudera 6.x, the default is also automatic download of the clients from the public repository. However, in these environments you cannot specify a different repository in the Jaguar configuration file. Instead, the CLI for installer on the database side provides the `--alternate-repo` parameter. Use this parameter to pass the client download location to the installer. See `--alternate-repo`in the Command Line Parameter Reference for bds-database-install.sh

- *If the network is Kerberos-secured, do you want the installer to set up automatic Kerberos ticket renewal for the Kerberos principal on the Hadoop side and the Oracle Database side?*

  See the parameters in the `kerberos` section:

  ```
  "principal" : "<oracle or other>/mycluster@MY.<DOMAIN>.COM",
  "keytab" : "/home/oracle/security/oracle.keytab",
  "hdfs-principal" : "hdfs/mycluster@MY.<DOMAIN>.COM",
  "hdfs-keytab" : "/home/hdfs/security/hdfs.keytab"
  ```

  The Kerberos principal and keytab identified here are used on the Hadoop side. They also copied into the database-side installation bundle. You can either use the same principal or a different principal on the database side. See `--alternate-principal` in Command Line Parameter Reference for bds-database-install.sh.

- *Do you want the Oracle Big Data SQL install process to automatically restart services that are in a stale state?*

  By default, stale services are restarted automatically. If you want to suppress this, you can set the `restart_stale` parameter in the configuration file to `"false"`.

- *Is the Hadoop cluster using the default REST API port for CDH or Ambari?*

  If not, set the `ports` parameter.

- *Are the HDFS or Hive daemons in the Hadoop cluster owned by non-default groups and/or users?*
  By default, HDFS daemons are owned by the `hdfs` user in the `hdfs` group. Hive daemons are by default owned by the `hive` user in the `hive` group. If these defaults have been changed, use the parameters in the `hadooop_ids section` of the configuration file to identify the current groups and users for these daemons: `hdfs_user`, `hdfs_group`, `hive_user`, `hive_group`.

> **Note:**
>
> Setting these parameters in the configuration file does not complete the set up for some features. For example, to enable Database Authentication, you must also pass a special `––requestdb` parameter to the Jaguar utility in order to identify the target database or databases. There are also steps required to generate and install the security key used by this feature. To enable Multi-User Authorization, you start by setting the Hadoop Impersonation parameters in the configuration file, but also need to set up the authorization rules. The steps to complete these setups are provided where needed as you work through the instructions in this guide.

2.3 About the Jaguar Utility

Jaguar is a multifunction command line utility that you use to perform all Oracle Big Data SQL operations on the Hadoop cluster.

Jaguar currently supports these operations:

- `install`
  - Deploys Oracle Big Data SQL binaries to each cluster node that is provisioned with the DataNode service.
  - Configures Linux and network settings for `bd_cell` (the Oracle Big Data SQL service) on each of these nodes.
  - Generates the bundle that installs Oracle Big Data SQL on the Oracle Database side. It uses the parameter values that you set in the configuration file in order to configure the Oracle Database connection to the cluster.
- `reconfigure`

  Modifies the current configuration of the installation (according to the settings in the configuration file provided).
- `databasereq`

  Generates a request key file that contains one segment of the GUID-key pair used in Database Authentication. (The `databasereq` operation performs this function only. For `install` and `reconfigure`, request key generation is an option that can be included as part of the larger operation.)
- `databaseack`

  Perform the last step in Database Authentication setup -- install the GUID-key pair on all Hadoop DataNodes in a cluster in order to allow queries from the Oracle Database that provided it.
- `sync_principals`
  Gets a list of principals from a KDC running on a cluster node and use it to create externally-identified database users for Query Server.
- `uninstall`

  Uninstalls Oracle Big Data SQL from all DataNodes of the Hadoop cluster.

See Also:

Jaguar Operations in the next section provides details and examples.

2.3.1 Jaguar Configuration Parameter and Command Reference

This section describes the parameters within the Jaguar configuration file as well as Jaguar command line parameters.

Configuration Parameters

The table below describes all parameters available for use in `bds-config.json` or your own configuration file. Only the cluster `name` parameter is always required. Others are required under certain conditions stated in the description.

Note:

When editing the configuration file, be sure to maintain the JSON format. Square brackets are required around lists, even in the case of a list with a single item.

Table 2-1 Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`cluster`	`name`	String	The name of the cluster. For CDH clusters (Oracle Big Data Appliance or other), this name can be either the physical cluster name or the display name. The installer searches first by physical name and then by display name. The `name` parameter is required only if Cloudera Manager will manage two or more clusters.

`cluster`	database_compatibility	string	Select which Oracle Database versions must be supported. Possible values: `"12.1"`| `"12.2"`| `"18"`|`"19"`| `"full"`
			For example, either of these settings enables support for Oracle Database 12.1, 12.2, 18c, and 19c.
			`"database_compatibility" : [ "12.1" ]`
			`"database_compatibility" : [ "full" ]`
			Either of the following settings enables support for Oracle Database 12.2, and 18c, but disable support for Oracle Database 12.1. By disabling support for Oracle Database 12.1 if it is not needed, you conserve some system resources, particularly memory.
			`"database_compatibility" : [ "12.2" ]`
			`"database_compatibility" : [ "18" ]`
			The default is `"full"`.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`api`	`hostname`	String	Visible hostname for the cluster management server. In some scenarios, the visible hostname for Cloudera Manager or Ambari is not the same to the current hostname, for example, in High Availability environments. Default: the local hostname.
`api`	`skip_health_check`	Boolean	If "`true`", the cluster health check is skipped. The cluster health check verifies that HDFS, Hive and Yarn services are running with good health and are not stale. Additionally, for CDH clusters, management services should be running with good health and not stale. Default: "`false`".
`api`	`port`	Integer	Cloudera Manager or Ambari REST API port. By default, on CDH clusters this port is 7183 for secured and 7180 for unsecured access. For Ambari, is 8443 for secured and 8080 for unsecured. Optional.
`api`	`restart_stale`	Boolean	If “`true`”, then services with stale configurations are restarted at the end of install process. These services are HDFS NodeName, YARN NodeManager and/or Hive (depending upon the settings selected). If “`false`”, the installation will finish but those services will remain on stale state. This is useful for avoiding unwanted service interruptions. You can then restart later when it is more convenient. Optional. The default is “`true`”.
`edgedb`	`enabled`	Boolean	Determines whether or not the Query Server functionality is enabled or not. Default: "`false`"
`edgedb`	`node`	String	Hostname of the node where the Query Server database will be running (if enabled). **Note:** Because Query Server is resource-intensive, it is highly recommended that you install the database on a dedicated node. Query server cannot run on a node that is running the DataNode role, nor the BDSSERVER role.
`object_store_support`	`enabled`	boolean	If "`true`", Oracle Wallet is set up both in the cluster and on the database system in order to allow access to Object Store. Default: "`false`".

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`object_store_support`	`cell_http_proxy`	`string`	If object store access support is enabled, this parameter is required for access to an object store from the Hadoop cluster side, even for empty values. Follows same rules as the Linux http_proxy variable. For example: `http://myproxy.<domain>.com:80`. No default value.
`object_store_support`	`cell_no_proxy`	`string`	Like `cell_http_proxy`, supports access to object stores and is also required if this access is enabled, even for empty values. Follows same syntax rules as the Linux no_proxy environment variable. For example: `localhost,127.0.0.1,.<domain>.com`. No default value.
`object_store_support`	`database_http_proxy`	`string`	Same description as `cell_http_proxy`, except that this parameter supports object store access from the database side, not the Hadoop side.
`object_store_support`	`database_no_proxy`	`string`	Same description as `cell_no_proxy`, except that this parameter supports object store access from the database side, not the Hadoop side.
`network`	`http_proxy` `https_proxy`	String	Specify the proxy settings to enable download of the Hadoop client tarballs and cluster settings files. If both of these strings are empty, the OS environment proxy settings are used. By default, both strings are empty. Using these two parameters in the configuration file is optional. If they are needed, you could instead set them externally as in `export http_proxy=<proxy value>` Not applicable to Oracle Big Data Appliance
`network`	`extra_nodes`	List	List additional nodes where the BDSAgent should be installed The BDSAgent and BDSServer roles are installed on all DataNodes instances. In addition, BDSAgent is installed on cluster nodes running HiveServer2 and HiveMetaStore instances. All remaining nodes are automatically excluded unless you add them here. Default: empty
`network`	`excluded_nodes`	List	Nodes that are not hosting the DataNode role can be excluded by listing them within this parameter.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`security`	`impersonation_enabled`	Boolean	If "`true`", Hadoop impersonation for Multi-user Authorization support is enabled. This sets up the `oracle` OS user as the Hadoop proxy user and propagates the proxy user's black list to the database nodes. If "`false`", this feature is not enabled. Default value: "`true`" in Oracle Big Data SQL 4.x and higher. **Note:** For CDH clusters, if the Sentry service is running, this setting is overidden and impersonation is enabled regardless of the value of this parameter.
`kerberos`	`principal`	String	The fully-qualified Kerberos principal name for a user. Before 4.1 the principal had to be `oracle` user. Starting with 4.1 the principal can be that of the `oracle` user or any other user. The Jaguar installer does not create the principal or the keytab. These must already exist. The principal has three parts: • The User Name: The name of the Linux account associated with the principal. Note that Kerberos principal names are case-sensitive. • Qualifier: "/<*qualifier*>." This is optional information to help you organize and identify principals. • Domain: "`@MY.<DOMAIN>.COM`." This is required information managed by the KDC. The Oracle Big Data SQL installation uses the Kerberos `principal` field (and `keytab` field below) to set up Kerberos authentication and automated Kerberos ticket renewal for the user represented by the principal. It does this on the Hadoop cluster. It also does it on the Oracle Database system automatically if two conditions are true: • The principal to be used on the Hadoop cluster and the Oracle Database system are the same. • The principal for the database owner does not exist. If the database owner principle does exist, it takes precedence and is used for authentication. Required for secured clusters. **Note:** Later, when you perform the database-side installation of Big Data SQL, review the description of the `--alternate-principal` install parameter in the Command Line Parameter Reference for bds-database-install.sh . This reference provides steps to take prior the database-side installation if the principal used on the Hadoop and database sides are not the same, or if the principal required is not that of the database owner.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`kerberos`	`db-service-principal`	String	Specifies a principal on the KDC server for use by Query Server (and only Query Server). It is not used for authentication against an external Oracle Database. Both `db-service-principal` and `db-service-keytab` are used to validate the Kerberos ticket presented by a client . Note that the parameters SQLNET.AUTHENTICATION_KERBEROS5_SERVICE and SQLNET.KERBEROS5_KEYTAB in `sqlnet.ora` will be set accordingly. The qualifier for the principal name must match the fully qualified domain name of the node where the Query Server will be running. Required for secured clusters.
`kerberos`	`db-service-keytab`	String	Fully-qualified location of the keytab file for the principal specificed with `db-service-principal`. Be sure to store the keytab in a location that is accessible to the Jaguar installer.
`kerberos`	`sync_principals`	Boolean	The sync_principals parameter specifies whether or not Jaguar automatically gets a list of principals from a KDC running on a cluster node and then uses the list to create externally-identified database users for Query Server. If set to true, then an automatic synchronization with Kerberos principals occurs during Jaguar install and reconfigure operations. The user can also call this synchronization at any time by invoking the sync_principals operation of Jaguar on the command line. Default: `"true"`.
`kerberos`	`hdfs-keytab`	String	Fully-qualified path to the principal keytab file. A keytab file is created for each principal on the KDC server. It must exist in a location accessible to the Jaguar installer. Required for secured clusters.
`kerberos`	`keytab`	String	Fully-qualified location for the principal’s keytab file name. Copy the keytab file to a location accessible to the Jaguar installer and set the path as the value of this parameter.
`kerberos`	`hdfs-principal`	String	Fully-qualified Kerberos principal name for the "`hdfs`" user. It has three parts: `User name`, `Qualifier`, and `Domain`. The `User name` is the fully-qualified principal name for the `hdfs` user. `Qualifier` is the cluster name prefixed by a forward slash, as in `/mycluster`. `Domain` is specified in the form `@MY.<DOMAIN>.COM`. All three are required. The principal name is defined on the KDC. Required for secured clusters.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`repositories`	`dir`	List	List of directories where the Hadoop clients for deployment on the database side are located. These directories can be on the local file system or on NFS. Directories are searched in the order listed. By default, the list is empty. If the `dir` list has any entries, these are searched before the URL list is searched, since this option should provide the fastest access to the clients. To give the installer the quickest access to the tarballs, you could set up a local repository, download the tarballs separately though a direct Internet connection, copy them into a directory on the same node where the Oracle Big Data SQL installer will run, and list that directory in the `dir` parameter. Optional. Not applicable to Oracle Big Data Appliance, which already includes the required clients. **Important:** The `dir` and `url` parameters are not supported on Cloudera 6.x systems. On these systems, you can specify a repository when you run the installer on the database side.
`repositories`	`url`	List	This is the list of URLs where the Hadoop client tarballs for deployment on the database side are located. If you data center already has repositories set up for access via HTTP, then you may prefer to maintain the Hadoop tarballs in that repository and use URL parameter for Oracle Big Data SQL installations. The URLs can be to the localhost, an internal network, or a site on the Internet (if the node has Internet access). The URLs are tried in the order listed. Note that internal proxy values and/or OS environment proxy settings must be set to allow this access if needed. If access to all listed repositories fails and/or Internet access is blocked, the database installation bundle is not created and a warning message is displayed. After correcting any problems and providing access to a repository, you can re-run the installer using the `reconfigure` and the installer should successfully generate the database-side installation bundle. Note that `reconfigure` detects and implements changes according the current directives in the configuration file. It does not uninstall and reinstall Oracle Big Data SQL on the cluster. Not applicable to Big Data Appliance, where the tarballs are stored in a local repository in the cluster and the location is automatically added to the configuration file.
`network`	`use_infiniband`	Boolean	If “`true`”, the communication will be set through private network interface, if "`false`" by client network interface. Used for Oracle Big Data Appliance clusters only. Default value: “`false`”.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
`network`	`selection_subnet`	String	If Hadoop cluster nodes have several network interfaces, you can use `selection_subnet` to select one. The selected IP address will be the nearest to the selection subnetwork. If the Hadoop cluster nodes have only one network interface, this parameter is ignored. The default value depends upon these conditions: • On non-Oracle commodity Hadoop clusters (CDH or HDP) the default selection is 0.0.0.0/0. (If a cluster node has several IP addresses, the lowest address is selected.) • On Oracle Big Data Appliance, the default is either the private or client IP address, depending upon the setting of the `use_infiniband` parameter. **Note for Oracle Big Data Appliance Users:** It's possible to configure several networks on an Oracle Big Data Appliance. If multiple networks exist, then this parameter must be set in order to select a specific network.
security	`database_auth_enabled`	Boolean	If "`true`", the database authentication through the GUID-key mechanism is enabled. This requires an extra step in the installation process in order to set up the database GUID-key pair on the cluster side. If "`false`", the feature will not be enabled. Default value: "`true`".
security	`impersonation_blacklist`	String	The Hadoop proxy users blacklisted for impersonation. This parameter is used only if Hadoop impersonation is enabled. Since this is a required setting on the Oracle Database side, it is provided with a default value of "`dummy`" in order to avoid extproc errors that can occur if Hadoop Impersonation is not enabled.
security	`impersonation_usehosts`	Boolean	If "`true`", the proxy hosts variable is set to the IP address of the database node. If "`false`", the proxy hosts variable is set to the wildcard: "*". Default value: "`true`".
memory	`min_hard_limit`	Integer	The minimum amount of memory reserved for Big Data SQL, in megabytes. This parameter is used on CDH clusters (Oracle Big Data Appliance and others). It is not used on HDP clusters. By default, the value is 32768 MB (32 GB) . If you set the `database_compatibility`parameter to "`full`", then the value of `min_hard_limit` must be 64 MB.

Table 2-1 (Cont.) Configuration Parameters in bds-config.json (or in Customer-Created Configuration Files)

Section	Parameter	Type	Description
memory	`max_percentage`	Integer	On CDH clusters (Oracle Big Data Appliance and others) this parameter specifies the percentage of memory on each node to reserve for Big Data SQL. This percentage is considered from a total amount of: NodeManager if YARN ResourceManager is enabled for that node. Physical memory if not. If the YARN Resource Manager is enabled for the node, then percentage should be based on the total amount of memory used by the NodeManager. Otherwise it should be a percentage of physical memory. This parameter is ignored on HDP clusters.
hadoop_ids	`hdfs_user`	String	The operating system user that runs the Hadoop HDFS daemons. Default value: `"hdfs"` **Note:** By default, Jaguar assumes Hive and HDFS usernames and groups. But if you used different names in your Hadoop installation, then use the hadoop_ids parameters to identify them.
hadoop_ids	`hdfs_group`	String	The operating system group that runs the Hadoop HDFS daemons. Default value: `"hdfs"`
hadoop_ids	`hive_user`	String	The operating system user that runs the Hadoop Hive daemons. Default value: `"hive"`
hadoop_ids	`hive_group`	String	The operating system group that runs the Hadoop Hive daemons. Default value: `"hive"`

Note:

After Oracle Big Data SQL is installed on the Hadoop cluster management server, you can find configuration file examples that demonstrate various parameter combinations in the `<Big Data SQL Install directory>/BDSjaguar` directory:

```
example-bda-config.json
example-cdh-config.json
example-kerberos-config.json
example-localrepos-config.json
example-subnetwork-config.json
example-unsecure-config.json
```

You can see all possible parameter options in use in `example-cdh-config.json`.

See Also:

See the Appendix Downloading the Correct Versions of the Hadoop, Hive, and HBase Clients for a Local Repostory for suggestions that can help with the setup of client tarball downloads.

Jaguar Operations

The table below lists the full set of operations performed by the Jaguar utility on the Hadoop side of the Oracle Big Data SQL installation.

The general syntax for Jaguar commands is as follows. The `--requestdb` parameter does not apply to all Jaguar commands.

```
# ./jaguar {--requestdb <comma-separated database names> | NULL } <action>
{ bds-config.json | <myfilename>.json | NULL }
```

Examples:

```
# ./jaguar install
# ./jaguar install bds-config.json
# ./jaguar install mycustomconfig.json
# ./jaguar --requestdb orcl,testdb,proddb install
# ./jaguar --requestdb orcl install
# ./jaguar sync_principals
```

You can use the default `bds-config.json` or your own configuration file, or omit the configuration file argument (which defaults to `bds-config.json`).

About --requestdb:

The --requestdb parameter is required for the databasereq command, optional for install, and reconfigure, and non-applicable for other Jaguar commands. The parameter must be passed in to one of these operations in order to enable Database Authentication in the connection between a Hadoop cluster and a database. Unless you prefer to disable Database Authentication, it is recommended that you include --requestdb with the initial install operation. Otherwise, you will need perform an additional step later in order to generate the request key.

This parameter is functional only when Database Authentication (database_auth_enabled) is set to “true” in the configuration. (This setting is a configuration default and does not need to be explicitly set in the configuration file.)

Jaguar needs the database names in order to generate a unique .reqkey (request key) file for each database. When database_auth_enabled is set “true” at installation time, the --requestdb parameter is still optional. Post-installation you have the same option to send the request key using the reconfigure, and databasereq operations. Database Authentication is not implemented until you do all of the following:

1. Ensure that database_auth_enabled is either absent from the configuration file or is set to “"true". (It is "true" by default.)
2. Include --requestdb in a Jaguar command:
   a. Run the Jaguar install or reconfigure and install the updated database-side installation bundle.
   b. Run Jaguar databasereq to generate an acknowledge key from the existing database side installation.
3. Copy the generated ZIP file that contains the .ackkey file from the database-side installation directory to /opt/oracle/DM/databases/conf on the Hadoop cluster management server.
4. Run the Jaguar databaseack command as described in the table below.

The table below shows the available Jaguar commands.

Table 2-2 Jaguar Operations

Jaguar Operation	Supports --requestdb?	Usage and Examples

`install`
The `--requestdb` parameter is not strictly required by the install operation, but you cannot enable Database Authentication if you do not generate a request key for each database.
`--requestdb` *`<comma-separated database list>`*
	Y	Installs Oracle Big Data SQL on the Hadoop cluster identified in the configuration file and creates an installation bundle for the database side based on the parameters included in the configuration file (or default values for parameters not explicitly assigned value in the configuration file). Examples:
		`# ./jaguar --requestdb orcl,testdb,proddb install`
		No configuration file parameter is included in the above example. `bds-config.json` is the implicit default. You can specify a different configuration file as in`./jaguar --requestdb mydb install myconfig.json`
		Note:
		You may need to use the `scl` utility to ensure that the correct Python version is invoked:
		`scl enable python27 "./jaguar install"`
		On Big Data Appliance clusters running Oracle Linux 6 and Oracle Linux 7, scl is not needed in order call the correct Python version for Jaguar.

Table 2-2 (Cont.) Jaguar Operations

Jaguar Operation	Supports --requestdb?	Usage and Examples

`reconfigure`	Y	Modify the current installation by applying changes you have made to the configuration file (`bds-config.json` or other).
		`# ./jaguar reconfigure myconfigfile.json`
		Note that if you run `./jaguar reconfigure <config file>` to reconfigure Oracle Big Data SQL on the Hadoop cluster, a corresponding reconfiguration is required on the Oracle Database side. The two sides cannot communicate if the configurations do not match. The Jaguar utility regenerates the database-side bundle files to incorporate the changes, You must redeploy the bundle on all database servers where it was previously installed.
		The `--requestdb` argument is required if `database_auth_enabled` is set to "`true`" in the updated configuration file. This is so that Jaguar will generate `.reqkey` files that are included in the database-side installation bundle. Note that we let the configuration file parameter default to `bds-config.json`.
		`# ./jaguar --requestdb demodb,testdb,proddb1 reconfigure`

`databasereq`	Y	Use this command to create the `.reqkey` file without repeating the Hadoop-side installation, or doing a `reconfigure` operation. (For example, if you forgot to include the `--requestdb` argument with the Jaguar `install` command), you can create a request key later with `databasereq`. This operation requires that `database_auth_enabled` is set to "true" (the default value) in the configuration. `<br>`# ./jaguar --requestdb demodb,testdb,proddb1 databasereq`

Table 2-2 (Cont.) Jaguar Operations

Jaguar Operation	Supports --requestdb?	Usage and Examples

`databaseack`	N	The “Database Acknowledge” process provides confirmation to the Oracle Big Data SQL installation on the Hadoop cluster that security features you enabled in the configuration file have been successfully implemented in the database-side installation. It then completes implementation of the selected security features on the Hadoop cluster side.
		`./jaguar databaseack bds-config.json`
		Only run `databaseack` if you chose to enable security features by setting either of these parameters in the configuration file to “`true`”:
		• `"impersonation_enabled" : "true"`
		• `"database_auth_enabled" : "true"`
		If a database-side installation bundle is built with any of these features set to “`true`”, then the database-side installation from that bundle generates a ZIP file in the installation directory under `$ORACLE_HOME` on the database server. The format of the ZIP file name is <*Hadoop cluster name*>-<*Number nodes in the cluster*>-<*FQDN of the cluster management server node*>-<*FQDN of this database node*>.zip. For example:
		`$ ls $ORACLE_HOME/BDSJaguar-4.1.2/cdh510-6-node1.my.<domain>.com/*.zip`
		`$ cdh510-6-node1.my.<domain>.com-myoradb1.my<domain>.com.zip`
		Copy this zip archive back to `/opt/oracle/DM/databases/conf` on the Hadoop cluster management server after the database-side installation is complete. Then, to fully enable the security features, run `databaseack`.

`sync_principals`	N/A	Gets a list of principals from a KDC running on a cluster node and use it to create externally-identified database users in Query Server. You can do the same by including the similarly-named sync_principals parameter in a Jaguar configuration file during Jaguar `install` and `reconfigure` operations.
`--object-store-http-proxy`	N/A	Specify a different proxy for Object Store access than the one set in the configuration file.
`--object-store-no-proxy`	N/A	Sets a no-proxy value and overrides the `no_proxy` value that may be set in the configuration file.
`uninstall`	N/A	Uninstall Oracle Big Data SQL from the Hadoop cluster. The uninstall process stops the `bd_cell` process (the Oracle Big Data SQL process) on all Hadoop cluster nodes, removes all instances from Hadoop cluster, and release all related resources.

Note:

When Oracle Big Data SQL is uninstalled on the Hadoop side, any queries against Hadoop data that are in process on the database side will fail. It is strongly recommended that you uninstall Oracle Big Data SQL from all databases systems shortly after uninstalling the Hadoop component of the software.

See Also:

Uninstalling Oracle Big Data SQL.

2.4 Steps for Installing on the Hadoop Cluster

After you have set up the Jaguar configuration file according to your requirements, follow these steps to run the Jaguar installer, which will install Oracle Big Data SQL on the Hadoop cluster and will also generate a database-side installation bundle that you deploy to the Oracle Database system. In these steps, `bds-config.json` is the configuration filename passed to Jaguar. This is the default. Any file name is accepted, therefore you can create separate configuration files for installation on different clusters and save them in different files.

Note:

Jaguar requires Python 2.7 to 3.0. Versions greater than 3.0 are not supported by Oracle Big Data SQL at this time. If necessary, you can add a Jaguar-compatible version of Python as a secondary installation. Revisit the prerequisites section in the Introduction for details. If you are using Oracle Big Data Appliance, do **not** overwrite the Mammoth-installed Python release.

1. Log on to the cluster management server node as `root` and cd to the directory where you extracted the downloaded Oracle Big Data SQL installation bundle.
2. Cd to the `BDSJaguar` subdirectory under the path where you unzipped the bundle.

   ```
   # cd <Big Data SQL Install Directory>/BDSJaguar
   ```

3. Edit the file `bds-config.json`.

   ```
   {
   "cluster": {
               "name": "<Your cluster name>"
               }
   }
   ```

   Add the parameters that you want to use in this installation.

See Also:

The cluster name is the only required parameter, but it is required only in environments where the configuration management service must manage more than one cluster. See the Jaguar Configuration Parameter and Command Reference for a description of all available parameters. You can see an example of a `bds-config.json` file populated with all available parameters in bds-config.json Configuration Examples.

In the BDSJaguar directory, run the Jaguar `install` operation. Pass the `install` parameter and the configuration file name. (`bds-config.json` is the implicit default) as arguments to the Jaguar command. You may or may not need to include the `--requestdb` option.

```
[root@myclusteradminserver:BDSjaguar] #  ./jaguar install <config file
name>
```

Note:

By default, Database Authentication is set to true unless you set `database_auth_enabled` to “false” in the configuration file. If you enable Database Authentication, then either as part of the install operation or later, generate a “request key.” This is half of a GUID/key pair used in the authentication process. To generate this key, include the `--requestdb` parameter in the Jaguar `install` command line:

```
[root@myclusteradminserver:BDSjaguar] # ./jaguar --requestdb mydb
install
```

If the install was run with `database_auth_enabled` is “true”, you can use the Jaguar `databasereq` command to generate the key after the database-side installation. Several other Jaguar commands can also generate the request key if you pass them the `--requestdb` parameter.

Jaguar prompts for the cluster management service administrator credentials and then installs Oracle Big Data SQL throughout the Hadoop cluster. It also generates the database-side installation bundle in the `db-bundles` subdirectory. The following message is returned if the installation completed without error.

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

4. Check for the existence of the database side installation bundle:

```
# ls <Big Data SQL Install Directory>/BDSJaguar/db-bundles
 bds-4.1.2-db-<cluster>-<yymmdd.hhmi>.zip
```

This bundle is for setting up Oracle Big Data SQL connectivity Oracle database and the specific cluster defined in the `bds-config.json` (or other) configuration file. It contains all packages and settings files required except for an optional database request key file.

If you included `--requestdb` in the install command, then the installation also generates one or more database request key files under the `dbkeys` subdirectory. You should check to see that this key exists.

```
# ls <Big Data SQL Install Directory>/BDSJaguar/dbkeys
 cluster1db.reqkey
```

This completes the Oracle Big Data SQL installation on the Hadoop cluster.

See Also:

- Working With Query Server in the *Oracle Big Data SQL User's Guide*. If you chose to install Query Server, you can connect and start working with it now. It is not dependent on completion of the Oracle Database side of the installation.
- Post-Installation Tasks in this guide. Most of the tasks described are performed on the Hadoop system. You may want to complete those tasks before proceeding to the second half of the installation on the Oracle Database system. All of them are optional.

What Next?

After Jaguar has successfully installed Oracle Big Data SQL on the Hadoop cluster, you are done with the first half of the installation. The next step is to install Oracle Big Data SQL on the Oracle Database system that will run queries against the data on the Hadoop cluster.

To do this, copy the database-side installation bundle to any location on the Oracle Database system. Unless you set `database_auth_enabled` to `"false"` in the configuration file, then also copy over the `.reqkey` file generated by Jaguar.

Tip:

You only need to send a request key to a database once. A single request key is valid for all Hadoop cluster connections to the same database. If you have already completed the installation to connect one Hadoop cluster to a specific database, then the database has the key permanently and you do not need to generate it again or copy it over to the database again in subsequent cluster installations.

Go to Installing or Upgrading the Oracle Database Side of Oracle Big Data SQL for instructions on unpacking the bundle and installing the database-side components of the software.

See Also:

An example of the complete standard output from a successful installation is provided in Oracle Big Data SQL Installation Example.

3

Installing or Upgrading the Oracle Database Side of Oracle Big Data SQL

Oracle Big Data SQL must be installed on both the Hadoop cluster management server and the Oracle Database system. This section describes the installation of Oracle Big Data SQL on both single-instance and RAC systems.

3.1 Before You Start the Database-Side Installation

Review the points in this section before starting the installation.

If the current version of Oracle Big Data SQL is already installed on the Oracle Database system and you are making changes to the existing configuration, then you do not need to repeat the entire installation process. See Reconfiguring an Existing Oracle Big Data SQL Installation .

Important:

- If you intend to log in as the database owner user with an anonymous connection to SQL*Plus, first be sure that the ORACLE_HOME and ORACLE_SID environment variables are set properly. GI_HOME should also be set if Grid infrastructure is running on the system and its location is not relative to ORACLE_HOME.
- For multi-node databases (such as Oracle RAC systems), **you must repeat this installation on every compute node of the database**. If this is not done, you will see RPC connection errors when the Oracle Big Data SQL service is started.
- It is recommended that you set up passwordless SSH for root on the database nodes where Grid is running. Otherwise, you will need to supply the credentials during the installation on each node.
- If the `diskmon` process is not already running prior to the installation, a Grid infrastructure restart will be required in order to complete the installation.
- If you set up Oracle Big Data SQL connections from more than one Hadoop cluster to the same database, be sure that the configurations for each connection are the same. Do not set up one connection to use InfiniBand and another to use Ethernet. Likewise, if you enable database authentication in one configuration, then this feature must be enabled in all Oracle Big Data SQL connections between different Hadoop clusters and the same database.
- If the database system is Kerberos-secured, then it is important to note that authentication for the database owner principal can be performed by only one KDC. Oracle Big Data SQL currently does not support multiple Kerberos tickets. If two or more Hadoop cluster connections are installed by the same database owner, all must use the same KDC.

  The `KRB5_CONF` environment variable must point to only one configuration file. The configuration file must exist, and the variable itself must be set for the database owner account at system startup.

  Be sure that you have installed a Kerberos client on each of the database compute nodes.

  Also, if the principal used on the Hadoop side is not the same as the principal of the database owner, then before running the database-side installation, you must copy the correct principal from the KDC to the Oracle Database side. Then when you run the installer, include the `--alternate-principal` parameter in the command line in order to direct the installer to the correct keytab. (See Command Line Parameter Reference for bds-database-install.sh
- For database running under Oracle Grid Infrastructure, if the system has more than one network interfaces of the same type (two or more Ethernet, two or more InfiniBand, or two or more Ethernet-over-InfiniBand interfaces), then the installation always selects the one whose name is first in alphabetical order.

See Also:

Details are provide in Special Considerations When a System Under Grid Infrastructure has Multiple Network Interfaces of the Same Type.

3.1.1 Check for Required Grid Patches

Make sure all patches to the Oracle Grid Infrastructure required by Oracle Big Data SQL have been installed.

Check the Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support) to find up-to-date information on patch requirements.

3.1.2 Potential Requirement to Restart Grid Infrastructure

In certain database environments `bds-database-install.sh` needs to create `cellinit.ora` and/or `celliniteth.ora`. In these cases, the script will attempt to propagate similar changes across all nodes in the Grid Infrastructure. To do this, the script expects that passwordless SSH is set up between `oracle` and `root` or it will prompt for the root 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.

3.1.2.1 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 set up on the cluster, no user interaction is required. If 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.

3.1.3 Special Considerations When a System Under Grid Infrastructure has Multiple Network Interfaces of the Same Type

The Oracle Big Data SQL installation or SmartScan operation may sometimes fail in these environments because the wrong IP address is selected for communication with the cells.

When installing Oracle Big Data SQL within Oracle Grid Infrastructure, you cannot provide the installer with a specific IP address to select for communication with the Oracle Big Data SQL cells on the nodes. Network interface selection is automatically determined in this environment. This determination is not always correct and there are instances where the database-side installation may fail, or, you may later discover that SmartScan is not working.

You can manually correct this problem, but first it is helpful to understand how the installation decides which network interfaces to select.

How the Installation Selects From Among Multiple Interfaces of the Same Type on a System Under Grid Infrastructure

- The `diskmon` process is controlled by Oracle Grid Infrastructure and not by the database. Grid manages communications with the Oracle Big Data SQL cells.
- The Oracle Big Data SQL installer in these cases does not create `cellinit.ora` and `celliniteth.ora`, nor does it update the cell settings stored in these files. In these environments, the task is handled by Grid, because it is a cluster-wide task that must be synchronized across all nodes.
- If there are multiple network interfaces, the Grid-managed update to the cells automatically selects the first network interface of the appropriate type on each node. It selects the interface whose name is first in alphabetical order.

For example, here is a system that is under Grid Infrastructure. It has multiple InfiniBand, Ethernet, and Ethernet over InfiniBand (bondeth*) network interfaces. This is the list of interfaces:

```
[root@mynode ~]# ip -o -f inet addr show
1: lo    inet 127.0.0.1/8
2: eth0    inet 12.17.207.156/21
3: eth1    inet 16.10.145.12/21
19: bondeth0    inet 12.17.128.15/20
20: bondeth2    inet 16.10.230.160/20
21: bondeth4    inet 192.168.1.45/20
30: bondib0    inet 192.168.31.178/21
31: bondib1    inet 192.168.129.2/21
32: bondib2    inet 192.168.199.205/21
33: bondib3    inet 192.168.216.31/21
34: bondib4    inet 192.168.249.129/21
```

When the Oracle Big Data SQL installer runs on this system, the following interfaces would be selected.

- 192.168.31.178/21 is selected for the InfiniBand connection configured in `cellinit.ora`.

  Among the InfiniBand interfaces on this system, the interface name `bondib0` is first in an ascending alphabetical sort.
- 12.17.128.15/20 is selected for an Ethernet-over-InfniBand connection configured in `celliniteth.ora`.

> **Note:**
>
> This example demonstrates an additional selection factor – Ethernet over InfiniBand takes precedence over standard Ethernet.
> The `bondeth0` interface name is first in this case.

How the Installation (or SmartScan) may Fail Under These Conditions

It is possible that `diskmon` cannot connect to the Oracle Big Data SQL cells via the network interface selected according to the logic described above. The correct subnet (one that `diskmon` can reach) may not appear first in an alphabetical sort.

How to Fix the Problem

You can manually change the IP addresses in the`cellinit.ora` and `celliniteth.ora` files. These files are at `/etc/oracle/cell/network-config` on each node.

1. Stop the CRS process. (Be sure to do this before the cell edit. If you do not, `diskmon` may hang.)
2. Edit `cellinit.ora` and/or `celliniteth.ora` . Change the IP addresses as needed.
3. Restart CRS.

3.2 About the Database-Side Installation Directory

You start the database side of the installation by unpacking the database-side installation bundle and executing the run file it contains. The run file creates an installation directory under `$(orabasehome)/BDSJaguar-4.1.2`. For example:

```
$(orabasehome)/BDSJaguar-4.1.2/cdh510-6-node1.mycluster.<domain>.com
```

The installation of Oracle Big Data SQL is not finished when this directory is created. The directory is a staging area that contains all of the files needed to complete the installation on the database node.

There can be Oracle Big Data SQL connections between the database and multiple Hadoop clusters. Each connection is established through a separate database-side installation and therefore creates a separate installation directory. The segments in the name of the installation directory enable you to identify the Hadoop cluster in this specific connection:

```
<Hadoop cluster name>-<Number nodes in the cluster>-<FQDN of the cluster
management server node>
```

You should keep this directory. It captures the latest state of the installation and you can use it to regenerate the installation if necessary. Furthermore, if in the future you need to adjust the database-side of the installation to Hadoop cluster changes, then the updates generated by the Jaguar `reconfigure` command are applied to this directory.

Consider applying permissions that would prevent the installation directory from being modified or deleted by any user other than `oracle` (or other database owner).

3.3 Steps for Installing on Oracle Database Nodes

To install the database side of Oracle Big Data SQL, copy over the zip file containing the database-side installation bundle that was created on the Hadoop cluster management server, unzip it, execute the run file it contains, then run the installer.

Perform the procedure in this section as the database owner (`oracle` or other). You stage the bundle in a temporary directory, but after you unpack the bundle and execute the run file it contains, then the installation package is installed in a subdirectory under `$(orabasehome)`. For example: `$(orabasehome)/BDSJaguar-4.1.2/cdh510-6-node1.mycluster.<`*`domain`*`>.com`.

Before starting, check that `ORACLE_HOME` and `ORACLE_SID` are set correctly. You should also review the script parameters in Command Line Parameter Reference for bds-database-install.sh so that you know which ones you should include in the install command for a given environment.

How Many Times Do I Run the Installation?

You must perform the installation for each instance of each database. For example, if you have a non-CBD database and a CBD database (DBA and DBB respectively, in the example below) on a single two-node RAC, then you would install Oracle Big Data SQL on both nodes as follows:

- On node 1

```
./bds-database-install.sh --db-resource=DBA1 --cdb=false
./bds-database-install.sh --db-resource=DBB1 --cdb=true
```

- On node 2

```
./bds-database-install.sh --db-resource=DBA2 --cdb=false
./bds-database-install.sh --db-resource=DBB2 --cdb=true
```

Copy Over the Components and Prepare for the Installation

Copy over and install the software on each database node in sequence from node 1 to node *n*.

1. If you have not already done so, use your preferred method to copy over the database-side installation bundle from the Hadoop cluster management server to the Oracle Database server. If there are multiple bundles, be sure to select the correct bundle for cluster that you want to connect to Oracle Database. Copy the bundle to any secure directory that you would like to use as a temporary staging area.

In this example the bundle is copied over to `/opt/tmp`.

```
scp root@<hadoop node name>:/opt/oracle/BDSJaguar/db-bundles/bds-4.1.2-db-
<Hadoop
    cluster>-200421.0517.zip /opt/tmp
```

2. If you generated a database request key, then also copy that key over to the Oracle Database server.

```
# scp root@<hadoop node name>:/opt/oracle/BDSJaguar/dbkeys/<database name
or other name>.reqkey oracle@<database_node> /opt/tmp
```

3. Then, `cd` to the directory where you copied the file (or files).
4. Unzip the bundle. You will see that it contains a single, compressed run file.
5. Check to ensure that the prerequisite environment variables are set – `$ORACLE_HOME` and `$ORACLE_SID`.
6. Run the file in order to unpack the bundle into `$(orabasehome)`. For example:

```
$ ./bds-4.1.2-db-cdh510-170309.1918.run
```

Because you can set up independent Oracle Big Data SQL connections between an Oracle Database instance and multiple Hadoop clusters, the `run` command unpacks the bundle to a cluster-specific directory under `$(orabasehome)/BDSJaguar-4.1.2`. For example:

```
$ ls $(orabasehome)/BDSJaguar-4.1.2
  cdh510-6-node1.mycluster.<domain>.com
  test1-3-node1.myothercluster.<domain>.com
```

If the `BDSJaguar-4.1.2` directory does not already exist, it is created as well.

7. If you generated a database request key, then copy it into the newly created installation directory. For example:

```
$ cp /opt/tmp/mydb.reqkey $(orabasehome)/BDSJaguar-4.1.2/cdh510-6-
node1.mycluster.<domain>.com
```

> **Tip:**
>
> You also have the option to leave the key file in the temporary location and use the `--reqkey` parameter in the installation command in order to tell the script where the key file is located. This parameter lets you specify a non-default request key filename and/or path.
>
> However, the install script only detects the key in the installation directory when the key filename is the same as the database name. Otherwise, even if the key is local, if you gave it a different name then you must still use `--reqkey` to identify to the install script.

8. If you plan to authenticate with a Kerberos principal that is not the same as the Kerberos principal used on the Hadoop side, then copy the keytab for that principal into the installation directory.
   Skip this step if you plan to use the same principal on both sides of the installation. A copy of the keytab that was used for authentication in the Hadoop cluster is already in the installation directory.

Now that the installation directory for the cluster is in place, you are ready to install the database side of Oracle Big Data SQL.

Install Oracle Big Data SQL on the Oracle Database Node

> **Important:**
>
> The last 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 this is a re-installation and the IP address for the Hadoop/Oracle Database connection 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. Start the database if it is not running.
2. Log on to the Oracle Database node as the database owner and change directories to the cluster-specific installation directory under `$(orabasehome)/BDSJaguar-4.1.2`. For example:

   ```
   $ cd $(orabasehome)/BDSJaguar-4.1.2/cdh510-6-node1.my.<domain>.com
   ```

3. Run `bds-database-install.sh`, the database-side Oracle Big Data SQL installer. You may need to include some optional parameters.

   ```
   $ ./bds-database-install.sh [options]
   ```

4. Restart the database (optional).

> **See Also :**
>
> The `bds-database-install.sh` installer command supports parameters that are ordinarily optional, but may be required for some configurations. See the Command Line Parameter Reference for bds-database-install.sh

Extra Step If You Enabled Database Authentication or Hadoop Secure Impersonation

- If `database_auth_enabled` or `impersonation_enabled` was set to "true" in the configuration file used to create this installation bundle, copy the ZIP file generated by the database-side installer back to the Hadoop cluster management server and run the Jaguar "Database Acknowledge" operation. This completes the set up of login authentication between Hadoop cluster and Oracle Database.

  Find the zip file that contains the GUID-key pair in the installation directory. The file is named according to the format below.

  ```
  <name of the Hadoop cluster>-<number nodes in the cluster>-<FQDN of the
  node where the cluster management server is running>-<FQDN of this
  database node>.zip
  ```

  For example:

  ```
  $ ls $(orabasehome)/BDSJaguar-4.1.2/cdh510-6-node1.my.<domain>.com/*.zip
  $ mycluster1-18-mycluster1node03.<domain>.com-myoradb1.<domain>.com.zip
  ```

  1. Copy the ZIP file back to `/opt/oracle/DM/databases/conf` on the Hadoop cluster management server.
  2. Log on to the cluster management server as `root`, cd to `/BDSJaguar` under the directory where Oracle Big Data SQL was installed, and run `databaseack` (the Jaguar "database acknowledge" routine). Pass in the configuration file that was used to generate the installation bundle (`bds-config.json` or other).

     ```
     # cd <Big Data SQL Install Directory>/BDSJaguar
     # ./jaguar databaseack bds-config.json
     ```

3.3.1 Command Line Parameter Reference for bds-database-install.sh

The bds-database-install.sh script accepts a number of command line parameters. Each parameter is optional, but the script requires at least one.

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

Parameter	Function

`--install`	Install the Oracle Big Data SQL connection to this cluster.
	Note:
	In mid-operation, this script will pause and prompt you to run a second script as root:
	`bds-database-install: root shell script /u03/app/masha/12.1.0/dbhome_mydb/install/bds-database-install-10657-root-scriptclust1.sh`
	`please run as root:`
	`<enter> to continue checking..`
	`q<enter> to quit`
	`bds-database-install: root`
	As root, open a session in a second terminal and run the script there. When that script is complete, return to the original terminal and press **Enter** to resume the `bds-database-install.sh` session.
	Because Oracle Big Data SQL is installed on the database side as a regular user (not a superuser), tasks that must be done as root and/or the Grid user require the installer to spawn shells to run other scripts under those accounts while `bds-database-install.sh` is paused.
	In some earlier releases, the `--install` parameter was implicit if no other parameters are supplied. You must now explicitly include this parameter in order to do an installation.

`--version`	Show the `bds-database-install.sh` script version.
`--info`	Show information about the cluster, such as the cluster name, cluster management server host, and the web server.
`--grid-home`	Specifies the Oracle Grid home directory.

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

Parameter	Function
`--crs`	Use `crsctl` to set up Oracle Big Data SQL MTA extprocs. Ignored in non-Grid environments. `--crs={true|false}` The installer always checks to verify that Grid is running. If Grid is not running, then the installer assumes that Grid is not installed and that the database is single-instance. It then automatically sets the `crs` flag to `false`. If `--crs=true` is explicitly set and Grid cannot be found, the installer terminates with an error message stating that `GI_HOME` must be set.
`--alternate-principal`	This optional parameter enables you to specify a Kerberos principal that is other than the database owner. It can be used in conjunction with the `--install`, `--reconfigure`, and `--uninstall` parameters. See About Using an Alternate Kerberos Principal on the Database Side below for details.

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

Parameter	Function

`--alternate-repo`	CDH 6 uses yum for the client installations. The `--alternate-repo` parameter enables you to specify an arbitrary yum repository for the installer to acquire the Hadoop, Hive, and HBase RPMs needed to complete the Big Data SQL installation on the database side. (The optional `url` and `dir` parameters in the `bds-config.json` configuration file on the Hadoop side of the installation are also for creating a pointer to a repository, but these parameters are ignored on CDH 6 systems.)
	For CDH 6.x prior to 6.3.3, `--alternate-repo` is optional. If Internet access is open, then by default the Hadoop-side Jaguar installer should be able to download the clients from the public Cloudera repository and integrate them into the database-side installation bundle that it generates. If the Internet is not accessible or if you want to download the clients from a different local or remote repository, use `--alternate-repo`. To enable the installer to download directly from the Cloudera repository (or elsewhere) pass the URL, as in this example for CDH 6.2.1:
	`# ./bds-database-install.sh --alternate-repo="https://archive.cloudera.com/cdh6/6.2.1/<Red Hat OS version>/yum/" <other parameters>`
	For CDH 6.3.3 and later, `--alternate-repo` is required, because you must provide Cloudera login credentials in order to download files for these releases from Cloudera. The Big Data SQL installer cannot automatically download CDH 6.3.3 and later files from the public repository. For these later releases, include you Cloudera user account and password, as in this example:
	`# ./bds-database-install.sh --alternate-repo="https://<Cloudera account name>:<Cloudera password>@archive.cloudera.com/p/cdh6/6.3.3/<Red Hat OS version>/yum/" <other parameters>`
	Note:
	You can set up a local yum repository that includes the correct version of the HDFS, Hive, and HBase clients and then point to that local repo instead of the Cloudera public repository:
	`# ./bds-database-install.sh --alternate-repo="<valid yum repository baseurl>" <other parameters>`
	See Cloudera's instructions on Configuring a Local Package Repository.
	There are several options if HTTP is not available on the database system:

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

Parameter	Function
	• Temporarily install a lightweight HTTP server as described in the Cloudera article at the URL above and then remove it after the Big Data SQL installation. • If HTTP is not an option, then on a machine that has Internet access, create a local mirror of the Cloudera repository (also described in the article) and then move the files into a directory that is accessible from the database server. This could be a local directory on the database machine itself or an NFS share elsewhere. Then, use a tool such as createrepo to set up the directory as a yum repository. This enables you to use the FILE:/// protocol for the baseurl passed to `--alternate-repo`, as in: `# ./bds-database-install.sh --alternate-repo="file:///path/to/local/repo" <other parameters>` The --alternate-repo parameter is required for CDH 6.3.3 and greater, optional for CDH 6.x prior to 6.3.3, unavailable for CDH 5 as well as for HDP systems.
`--cdb`	Create database objects on all PDBs for CDB databases. `--cdb={true|false}`

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

Parameter	Function

`--reqkey`	This parameter tells the installer the name and location of the request key file. Database Authentication is enabled by default in the configuration. Unless you do not intend to enable this feature, then as one of the steps to complete the configuration, you must use `--reqkey`.
	• To provide the key file name and the path if the key file is not in the local directory:
	`$ ./bds-database-install.sh --reqkey=/opt/tmp/some_name.reqkey --install`
	• To provide the key file name if the file is local, but the filename is not the same as the database name:
	`$ ./bds-database-install.sh --reqkey=some_name.reqkey --install`
	If the request key filename is provided without a path, then the key file is presumed to be in the installation directory. The installer will find the key if the filename is the same as the database name.
	For example, in this case the request key file is in the install directory and the name is the same as the database name.
	`$ ./bds-database-install.sh --install`
	This file is consumed only once on the database side in order to connect the first Hadoop cluster to the database. Subsequent cluster installations to connect to the same database use the configured key. You do not need to resubmit the `.reqkey` file.

`--uninstall`	Uninstall the Oracle Big Data SQL connection to this cluster. If you installed using the `--alternate-principal` parameter, then also include this parameter when you uninstall. `./bds-database-install --uninstall --alternateprincipal="<primary>/<instance>@<REALM>"` See `--alternate-principal` in this table for more detail.
`--reconfigure`	Reconfigures bd_cell network parameters, Hadoop configuration files, and Big Data SQL parameters. The Oracle Big Data SQL installation to connect to this cluster must already exist. Run reconfigure when a change has occurred on the Hadoop cluster side, such as a change in the DataNode inventory.
`--databaseack-only`	Create the Database Acknowledge zip file. (You must then copy the zip file back to `/opt/oracle/DM/databases/conf` on the Hadoop cluster management server and run `./jaguar databaseack <configuration file>`.)
`--mta-restart`	MTA extproc restart. (Oracle Big Data SQL must already be installed on the database server.)
`--mta-setup`	Set up MTA extproc with no other changes. (Oracle Big Data SQL must already be installed on the database server.)
`--mta-destroy`	Destroy MTA extproc and make no other changes. (Oracle Big Data SQL must already be installed on the database server.)

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

Parameter	Function
`--aux-run-mode`	Because Oracle Big Data SQL is installed on the database side as a regular user (not a superuser), tasks that must be done as `root` and/or the Grid user require the installer to spawn shells to run other scripts under those accounts while `bds-database-install.sh` is paused. The`--aux-run-mode` parameter specifies a mode for running these auxiliary scripts. `--aux-run-mode=<`*`mode`*`>` Mode options are: • `session` – through a spawned session. • `su` — as a substitute user. • `sudo` — through sudo. • `ssh` — through secure shell.
`--root-script`	Enables or disables the startup root script execution. `--root-script={true|false}`
`--no-root-script`	Skip root script creation and execution.
`--root-script-name`	Set a name for the root script (the default name is based on the PID).
`--pdb-list-to-install`	For container-type databases, Oracle Big Data SQL is by default set up on all open PDBs. This parameter limits the setup to the specified list of PDBs.
`--restart-db`	If a database restart is required, then by default the install script prompts the user to choose between doing the restart now or later. Setting `--restart-db=yes` tells the script in advance to proceed with any needed restart without prompting the user. If `--restart-db=no` then the prompt is displayed and the installation waits for a response. This is useful for unattended executions.

`--skip-db-patches-check`	To skip the patch validation, add this parameter when you run the installer:
	`./bds-database-install.sh`
	`      --skip-db-patches-check`
	By default, database patch requirements are checked when you run `./bds-database-install.sh`. If the patch validation fails, the installer returns a prompt with a warning message, indicating that there are some patches missing. Installation will continue after the warning.

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

Parameter	Function
`--set-default-cluster`	To set the default cluster to the current cluster in a multi-cluster installation, add this parameter when you run the installer: `./bds-database-install.sh --set-default-cluster` In a multi-cluster installation, the first cluster installed is considered the default cluster. If more clusters are installed, all queries use the java clients provided by the default cluster regardless of the versions of the actual cluster receiving the query. In some cases you may want to change the default, for example, to use newer java clients. If passing this parameter the current cluster is set as the default. The required database links and file system links are regenerated accordingly, and the existing multi-threading agents are restarted.
`--force-incompatible`	To allow different versions of Hadoop, set this parameter when you run the installer: `./bds-database-install.sh --force-incompatibble` Without this parameter, the installer prevents the installation of clusters with different major or minor Hadoop versions. For example, if a Hadoop 2.6 cluster is installed and then a Hadoop 3.0 is installed the script does not allow the second install. This behavior avoids possible Hadoop version incompatibilities. However, this parameter can be used to override this behavior. When the force-incompatible parameter is used and two clusters of different Hadoop versions are detected the user simply gets a warning.

The `--root-script-only` parameter from previous releases is obsolete.

3.3.2 About Using an Alternate Kerberos Principal on the Database Side

The Kerberos principal that represents the Oracle Database owner is the default, but you can also choose a different principal for Oracle Big Data SQL authentication on Oracle Database.

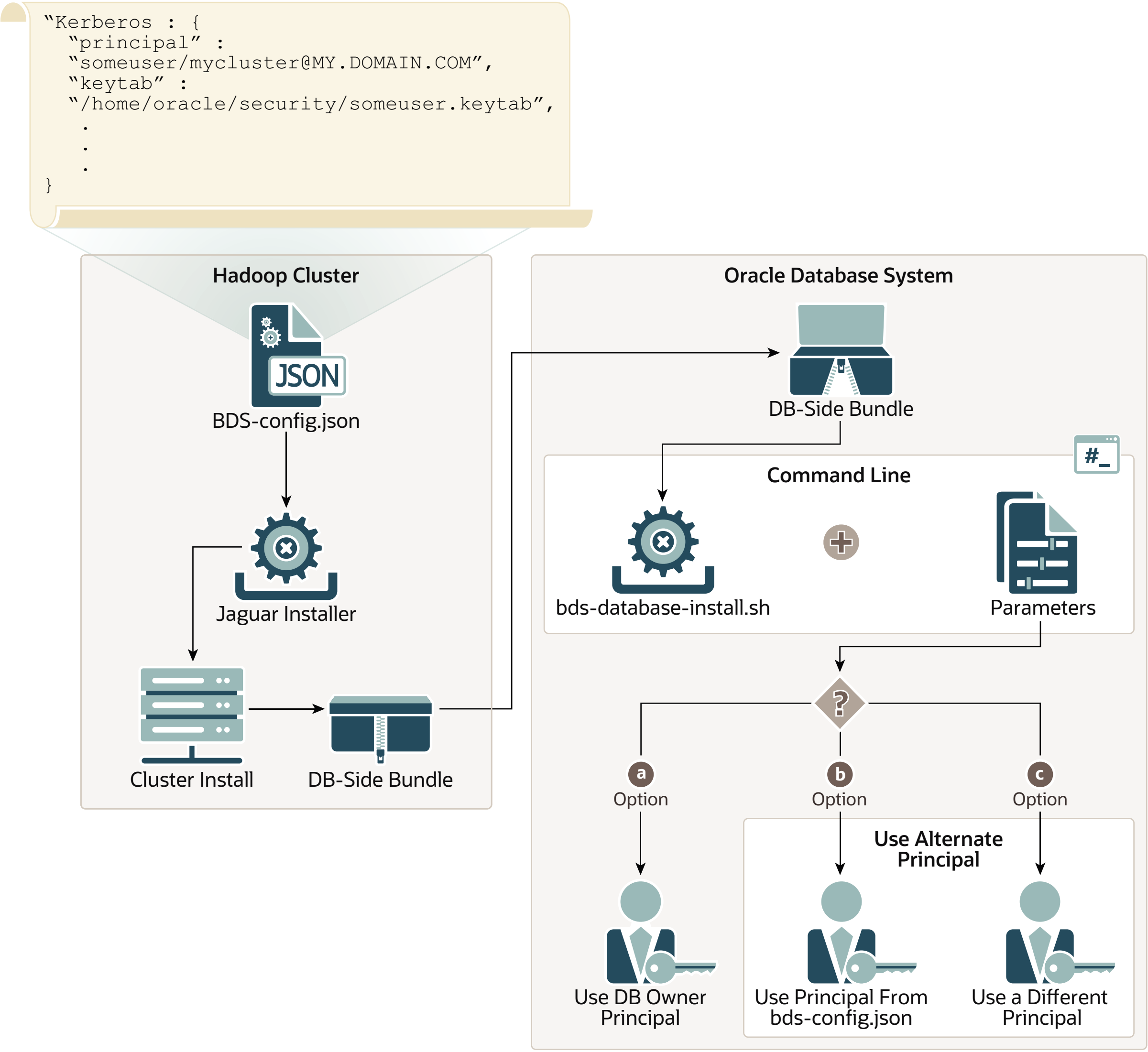
The figure below shows how alternate principals are handled in the installation. On the Hadoop side of the install, use the Kerberos principal and keytab specified in the BDS-config.json file to set up Kerberos authentication for Big Data SQL on the Hadoop cluster. The same keytab is included in the installation bundle that you unzip on the database side. When you unzip the bundle and run the installer on the database side, you have three choices for setting up Kerberos authentication:

a. Ignore the principal and keytab provided with the installation bundle and use the database owner principal.

b. Use the principal and keytab provided with the bundle.

c. Specify a different principal.

Figure 3-1 Using the --alternate-principal Parameter with bds-database-install.sh



```
“Kerberos : {
  “principal” :
  “someuser/mycluster@MY.DOMAIN.COM”,
  “keytab” :
  “/home/oracle/security/someuser.keytab”,
   .
   .
   .
}
```

This is the explanation of options **a**, **b**, and **c** in the figure above.

Option a: Use the database owner principal for Kerberos authentication.

If the keytab of the database owner principal is present in the staging directory and you want to use that principal, then do not include the `--alternate-principal` parameter when you invoke the installer. By default the installer will use the database owner principal.

```
# ./bds-datbase-install.sh --install <other parameters, except --alternate-
principal>
```

Option b: Use the principal included with the installation bundle for authentication.

Include `--alternate-principal` in the install command, but do not supply a value for this parameter. This tells the installer to use the principal that was specified in the Jaguar `bds-config.json` configuration file on the Hadoop side of the installation. This principal is included in the database-side installation bundle and already exists in the staging directory. This is the directory that the installation run file generated. It includes the installation files for connection to a specific Hadoop cluster and node. For example, `$(orabasehome)/BDSJaguar-4.1.2/cdh510-6-node1.mycluster.mydomain.com`.

```
# ./bds-datbase-install.sh --install --alternate-principal <other 
parameters>
```

Option c: Use a different principal for authentication.

Include `--alternate-principal` on the command line and pass the principal as the value.

This last case requires two extra steps before you run `bds-database-install.sh`. When you ran the run file that unpacked the database-side install bundle, it created a database-side installation directory. Copy the alternate keytab file into this directory. Then, rename the file by prefixing "`bds_`" to the principal. The prefix tells the installer to look for the principal in this keytab file. You can then run the command as follows:

```
./bds-database-install.sh --reconfigure --alternate-
principal=<primary>/<instance>@<REALM>
```

> **Note:**
>
> Any `bds-database-install.sh` operation that includes the `--alternate-principal` parameter generates a script that must be run as root. You will be prompted to run this script and then to press `Enter` to continue and complete the operation. For example:
>
> ```
> bds-database-install: root shell script : /home/oracle/
> db_home2/install/bds-database-install-28651-root-script-
> scaj42.sh
> please run as root:
> /home/oracle/db_home2/install/bds-database-install-28651-root-
> script-<cluster name>.sh
> waiting for root script to complete, press <enter> to continue 
> checking.. q<enter> to quit
> ```

Changing the Kerberos Principal Post Release

You can use the reconfigure operation to switch to a different principal.

```
./bds-database-install --reconfigure --alternate-
principal="<other_primary>/<instance>@<REALM>"
```

> **Important:**
>
> If the keytab of the database owner principal is found by the installer in the staging directory (the directory where the run file unpacked the installer files), then the database owner principal is used and the `--alternate-principal` parameter is ignored.

Removing a Kerberos Principal and Keytab When You Uninstall Big Data SQL

If you specified an alternate principle when you installed Big Data SQL, then if you uninstall Big Data SQL later, include the same `--alternate-principle` parameter and value, so that the uninstall operation removes the principal and keytab from the system.

```
./bds-database-install --uninstall --alternate-principal="bds_<primary>/
<instance>@<REALM>"
```

3.4 Granting User Access to Oracle Big Data SQL

In Oracle Big Data SQL releases prior to 3.1, access is granted to the `PUBLIC` user group. In the current release, 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.
- Grant read privileges on the Multi-User Authorization security table to enable impersonation.

For example, to grant access to `user1`:

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

> **See Also:**
>
> Use the DBMS_BDSQL PL/SQL Package in the *Oracle Big Data SQL User's Guide* to indirectly provide users access to Hadoop data. The Multi-User Authorization feature that this package implements uses Hadoop Secure Impersonation to enable the `oracle` account to execute tasks on behalf of other designated users.

3.5 Enabling Object Store Access

If you want to use Oracle Big Data SQL to query object stores, certain database properties and Network ACL values must be set on the Oracle Database side of the installation. The installation provides two SQL scripts you can use to do this.

The ORACLE_BIGDATA driver enables you to create external tables over data within object stores in the cloud. Currently, Oracle Object Store, Microsoft Azure, and Amazon S3 are supported. You can create external tables over Parquet, Avro, and text files in these stores. The first step is set up access to the object stores as follows.

Run set_parameters_cdb.sql and allow_proxy_pdb.sql to Enable Object Store Access

1. After you run `bds-database-install.sh` to execute the database-side installation, find these two SQL script files under $(orabasehome), in the cluster subdirectory under the BDSJaguar directory:

   ```
   set_parameters_cdb.sql
   allow_proxy_pdb.sql
   ```

2. Open and read each of these files. Confirm that the configuration is correct.

   > **Important:**
   >
   > Because there are security implications, carefully check that the HTTP server setting and other settings are correct.

3. In CBD ROOT, run `set_parameters_cdb.sql`.
4. In each PDB that needs access to object stores, run `allow_proxy_pdb.sql`.

In a RAC database, you only need to run these scripts on one instance of the database.

3.6 Enabling Hive Transactional Node Tables

If you want to enable transactional node table support on Hive, you need to manually add parameters to the Cloudera configuration files inside the <configuration> xml tag.

You must add the following entries to the `$ORACLE_HOME/bigdatasql/clusters/<your cluster name>/config/hive-site.xml` file.

```
<property>
  <name>hive.exec.dynamic.partition.mode</name>
  <value>nonstrict</value>
</property>
<property>
  <name>hive.txn.manager</name>
  <value>org.apache.hadoop.hive.ql.lockmgr.DbTxnManager</value>
</property>
```

3.7 Installing and Configuring Oracle SQL Access to Kafka

If you work with Apache Kafka clusters as well as Oracle Database, Oracle SQL Access to Kafka (OSAK) can give you access to Kafka brokers through Oracle SQL. You can then query data in the Kafka records and also join the Kafka data with data from Oracle Database tables.

After completing the Oracle Database side of the Oracle Big Data SQL installation, you can configure OSAK connections to your Kafka cluster.

Installation Requirements

- A cluster running Apache Kafka. Kafka version 2.3.0 is supported, as are all versions of Kafka that run on Oracle Big Data Appliance.
- A version of Oracle Database from 12.2 to 19c.

> **Note:**
>
> Big Data SQL supports Oracle Database 12.1, but OSAK does not.

Steps Performed by the Oracle Big Data SQL Installer

OSAK is included in the database-side installation bundle. As part of the installation, the `bds-database-install.sh` installer described in the previous section of this guide does the following:

- Copies the OSAK kit into the Big Data SQL directory (`$(orabasehome)/bigdatasql/`), unzips it, and verifies the contents of the kit.
- Creates the symlink `$(orabasehome)/bigdatasql/orakafka`, which links to the extracted OSAK directory.
- Sets JAVA_HOME for OSAK to `$(orabasehome)/bigdatasql/jdk`.

You can check for the OSAK installation steps in the `bds-database-install.sh` output. This example is from an Oracle Database 19c system. It is the same on earlier database releases except for the shiphome path.

```
 ...
 bds-database-setup: installing orakafka kit

 Step1: Check for valid JAVA_HOME
 --------------------------------------------------------------
 Found /u03/app/oracle/19.1.0/dbhome_orcl/shiphome/database/
bigdatasql/jdk/bin/java, \
JAVA_HOME path is valid.
 Step1 succeeded.

 Step2: JAVA version check
 --------------------------------------------------------------
 java version "1.8.0_171"
 Java(TM) SE Runtime Environment (build 1.8.0_171-b11)
 Java HotSpot(TM) 64-Bit Server VM (build 25.171-b11, mixed mode)
 Java version >= 1.8
 Step2 succeeded.

 Step3: Creating configure_java.sh script
 --------------------------------------------------------------
 Wrote to /u03/app/oracle/19.1.0/dbhome_orcl/shiphome/database/bigdatasql/
orakafka-1.1.0\
/bin/scripts/configure_java.sh
 Step3 succeeded.

 Successfully configured JAVA_HOME in /u03/app/oracle/19.1.0/dbhome_orcl/
```

```
shiphome/database\
/bigdatasql/orakafka-1.1.0/bin/scripts/configure_java.sh

 The above information is written to /u03/app/oracle/19.1.0/
dbhome_orcl/shiphome/database\
/bigdatasql/orakafka-1.1.0/logs/set_java_home.log.2020.02.20-17.40.23
...
```

Steps You Must Perform to Complete the OSAK Installation and Configuration

To complete the installation and configuration, refer to the instructions in the Oracle SQL Access to Kafka `README_INSTALL` file at `$(orabasehome)/bigdatasql/orakafka/doc`. The `README_INSTALL` file tells you how to use `orakafka.sh`. This script can help you perform the remaining configuration steps:

- Add a new Kafka cluster configuration directory under `$(orabasehome)/bigdatasql/orakafka/clusters/`.
- Allow a database user to access the Kafka cluster.
- Install the ORA_KAFKA PL/SQL packages in a user schema and create the required database directories.

> **Important:**
>
> If you are setting up access to a secured Kafka cluster, then after generating the cluster directory, you must also set the appropriate properties in this properties file.
>
> ```
> $(orabasehome)/bigdatasql/orakafka/clusters/<cluster_name>/
> conf/orakafka.properties
> ```
>
> The properties file includes some tested subsets of possible security configurations. For general information about Kafka security, refer to the Apache Kafka documentation at https://kafka.apache.org/documentation/#security

- Working with Oracle Cloud Infrastructure Streaming Service
  In order to work with Oracle Cloud Infrastructure Streaming Service, the following properties must be uncommented and set in the cluster configuration properties file:

  ```
  <app_data_home>/clusters/<cluster_name>/conf/orakafka.properties
      security.protocol=SASL_SSL

      sasl.mechanism=PLAIN

      #Parameters for JAAS config (Authentication using SASL/PLAIN)
      sasl.plain.username=<tenancyName/<username>/<streamPoolId>
      sasl.plain.password=<user auth_token>

      #Uncomment the following for SASL/PLAIN authentication
  ```

```
sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required username=\"${sasl.plain.username}\" password=\"$
{sasl.plain.password}\";
```

For more information on Kafka compatibility with Oracle Cloud Infrastructure Streaming Service, refer to Using Streaming with Apache Kafka.

When you have completed the configuration, the next step is to register your Kafka cluster. After registering the cluster, you can start using Oracle SQL Access to Kafka to create views over Kafka topics and query the data.

See Also:

Oracle SQL Access to Kafka in the *Oracle Big Data SQL User's Guide* shows you how to register your Kafka cluster and start working with OSAK.

4 Post-Installation Tasks

This section describes the following post-installation tasks.

4.1 Validating the Installation With bdschecksw and Other Tests

The script `bdschecksw.sh` runs automatically as part of the installation, but you can also run it at any time.

This script gathers and analyzes diagnostic information about the Oracle Big Data SQL installation from both the Oracle Database and the Hadoop cluster sides of the installation. You can also run this script as a troubleshooting check at any time after the installation. The script is in `$ORACLE_HOME/bin` on the Oracle Database server.

```
$ bdschecksw --help
```

- See Running Diagnostics With bdachecksw in the *Oracle Big Data SQL User's Guide* for a complete description.
- Also see How to do a Quick Test in the user's guide for some additional functionality tests.

4.2 Checking the Installation Log Files

You can examine these log files after the installation.

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.

4.3 Installing and Configuring Oracle SQL Access to Kafka (OSAK)

You have the option to install and configure OSAK after completing the Hadoop and Oracle Database sides of the Big Data SQL installation.

After performing the Big Data SQL installation, see Installing and Configuring Oracle SQL Access to Kafka

4.4 Perform Network ACL Setup for Object Store Access

This topic applies to Oracle Database 18c and 19c environments only.

Run these scripts generated by the Oracle Big Data SQL installer (`bds-database-install.sh`) to enable access to object stores.

Notice the following messages in the `bds-database-install.sh` output:

```
bds-database-setup: Object Store scripts templates have been created
bds-database-setup: set_parameters_cdb.sql     should be executed in
CDB$ROOT
bds-database-setup: allow_proxy_pdb.sql        should be executed on
required PDBs
```

You can find these scripts in the installer staging directory (for example: `$ORACLE_HOME/BDSJaguar-4.1.2/cdh510-6-node1.`*`<cluster name>`*`.`*`<domain name>`*`.com`).

If you intend to support connects to object stores from Oracle Database 18c or 19c, run the corresponding script on the CDB and the other on PDBs. For example, on the CDB:

```
# @set_parameters_cdb.sql
```

Run these scripts after you perform the installation for each connection to a Hadoop cluster. If you reconfigure the database-side installation (`./bds-database-install --reconfigure`), the reconfigure operation prompts you to run the scripts again:

```
bds-database-reconfigure: Object Store scripts templates have been created
bds-database-reconfigure: set_parameters_cdb.sql     should be executed in
CDB$ROOT
bds-database-reconfigure: allow_proxy_pdb.sql        should be executed on
required PDBs
```

You can edit and the re-run the scripts to customize them for different roles and URLs.

4.5 Run bds_cluster_node_helper.sh to Get Information About the Oracle Big Data SQL Installation on a Node

The script `bds_cluster_node_helper.sh` aggregates information about a Hadoop cluster node that is useful for Oracle Big Data SQL maintenance purposes.

This script provides options to do the following:

- Show Oracle Big Data SQL status information via bdscli, the Oracle Big Data SQL command line interface.
- Collect and archive log data that is pertinent to Oracle Big Data SQL operations. There are three levels to the scope of the data collection.
- Set some parameters that control the level of debug information in logs that are collected.

You can find this script at `<`*`Oracle Big Data SQL installation directory`*`>/BDSJaguar`. It must be run as root.

Usage

```
# bds_cluster_node_helper.sh [OPTIONS]
```

Table 4-1 Parameters for bds_cluster_node_helper.sh

Parameter	Description
`-h, --help`	Show usage information.
`-v, --version`	Show the Oracle Big Data Appliance release version.

`--skip-bdscli-info`	Skip bdscli information gathering.
	Default: `false`.
	Runs the following bdscli commands and returns the output:
	`bdscli -e "list bdsql"`
	`bdscli -e "list bdsql detail"`
	`bdscli -e "list offloadgroup"`
	`bdscli -e "list offloadgroup detail"`
	`bdscli -e "list quarantine"`

Table 4-1 (Cont.) Parameters for bds_cluster_node_helper.sh

Parameter	Description
`--get-logs [--log-level=<1|2|3>] [--bundle-name=<name>] [--wrap, --envelop]`	Generates a gzipped tar file of logs. Default: `false`. Options: • `--log-level=<supported value>` Specifies the log level. • `--bundle-name=<name>` Names the `.tar.gz` created. • `--wrap, --envelop` Prepares the bundle to be sent over email. **Note:** See the table below for more detail on each `-get-logs` sub-option.
`--set-debug=<on|off>`	`--set-debug=<supported value>` Set or remove the `_cell_server_event` parameter from the `cellinit.ora` file. • `--set-debug=on` – In the files `/opt/oracle/bigdatasql/bdcell-12.1/bigdata-log4j.properties` and `/opt/oracle/bigdatasql/bdcell-12.2/bigdata-log4j.properties`, this sets the parameter value `log4j.logger.oracle.hadoop.sql=ALL`.In cellinit.ora, sets `_cell_server_event` as follows: `_cell_server_event="trace[CELLSRV_Disk_Layer] disk=highest, memory=highest"` • `--set-debug=off` – In the file `/opt/oracle/cell/cellsrv/deploy/config/cellinit.ora`, this setting remove the parameter whose prefix is `_cell_server_event="trace[CELLSRV_Disk_Layer]` – In `bigdata-log4j.properties`, sets `log4j.logger.oracle.hadoop.sql=OFF`.

The table below provides full details on `bds_cluster_node_helper.sh --get-logs` sub-options.

Table 4-2 Sub-Parameters for --get-logs Option of bds_cluster_node_helper.sh

bds_cluster_node_helper.sh --get-logs sub-options	Description

`--get-logs --log-level=<1|2|3>`	Specifies the log level.
	Default: `1`.
	The scope of the recovery for each log level is as follows:
	• `--get-logs`, or `--get-logs --log-level=1`:
	`/var/log/bigdatasql`
	`/opt/oracle/cell/.install_log.txt`
	• `--get-logs --log-level=2`:
	Includes level 1 logging, plus:
	`/var/log/oracle`
	`/opt/oracle/cell/cellsrv/deploy/msdomain/servers/msServer/logs`
	• `--get-logs --log-level=3`
	Includes level 1 and 2 logging, plus:
	`/var/run/cloudera-scm-agent/process (on Cloudera clusters)`
	`/var/lib/ambari-agent/data (on Hortonworks HDP clusters)`
	Example: `# bds_cluster_node_helper.sh --get-logs --log-level=2`

`--get-logs --bundle-name=<name>`	Give a name to the created `tar.gz` bundle. Default: `bds-<Oracle Big Data SQL version>-<YYYY-mm-dd-HH-MM-SS>`. For example, `bds_4.1.2_2020-05-01_23-55-03.tar.gz` The customer can use this option to specify a different name. For example: `# bds_cluster_node_helper.sh --get-logs --bundle-name=logs_from_node1.tar.gz`

`--get-logs [--wrap | --envelop]`	Prepares the bundle for email transmission.
	Default: `false`.
	Examples:
	`# bds_cluster_node_helper.sh --get-logs --wrap`
	`# bds_cluster_node_helper.sh --get-logs --envelop`
	These sub-options are equivalent.

4.6 Other Post-Installation Steps to Consider

Some additional steps that are recommended after installation are:

- Read about measures you can take to secure the installation. (See Securing the Oracle Big Data SQL Installation.)
- Learn how to modify the Oracle Big Data SQL configuration when changes occur on the Hadoop cluster and in the Oracle Database installation. (See Expanding or Shrinking an Installation.)
- 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.)

5

Expanding or Shrinking an Installation

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.

- Adding or Removing Oracle Big Data SQL on Hadoop Cluster Nodes
- Adding or Removing Oracle Big Data SQL on Oracle Database Nodes

5.1 Adding or Removing Oracle Big Data SQL on Hadoop Cluster Nodes

On the Hadoop side, Oracle Big Data SQL must be installed on every cluster node where the DataNode role is running and should not be installed on other nodes. Note that when you run `jaguar reconfigure` as described below, you must also reconfigure the Oracle Database side of the installation.

The Jaguar `reconfigure` operaton uses the cluster management server to determine where the DataNode service is running and then installs or uninstalls Oracle Big Data SQL on each cluster node as needed.

Run `reconfigure` when the distribution of DataNode services has changed in the Hadoop cluster.

```
# cd <Big Data SQL install directory>/BDSJaguar
# ./jaguar reconfigure <JSON_configuration_file>
```

When the reconfiguration is done, this message is returned to the console:

```
BigDataSQL: RECONFIGURE workflow completed.
```

The following are situations where you should run `reconfigure`:

- New nodes are added to the Hadoop cluster and some or all of them are provisioned with the DataNode role.
- Services and roles are redistributed in the cluster and some DataNode instances are moved.
- One or more nodes provisioned with the DataNode role are removed from the cluster or deprovisioned.

Important:

Be sure to run `reconfigure` after a cluster expansion. This will enable local processing for data stored on the updated nodes. Also, the new DataNode instances in the cluster provide additional resource for query processing that is not utilized until Oracle Big Data SQL is installed on those nodes.
You must also run `bds-database-install.sh --reconfigure` on the database side after you run `reconfigure` on the Hadoop side.

See Also:

The Jaguar Configuration Parameter and Command Reference provides a full description of the Jaguar `reconfigure` operation.

Communicating DataNode Changes to Oracle Database

The Oracle Database side of the installation maintains an inventory of the DataNodes where Oracle Big Data SQL is installed in the Hadoop cluster. This is a map that tells it which nodes to query. If you run `reconfigure`, this updated DataNode inventory must be provided to the database compute nodes. The `reconfigure` procedure generates a new database installation bundle that includes the new inventory. Install this updated bundle on the database compute nodes using the same method that you use in a full installation. Install the update using `bds-database-install.sh --reconfigure`

1. As the database owner, copy the bundle to temporary location on the database node.
2. Extract the contents of the zip file.
3. Execute the run file extracted from the zip file.
4. Go to the installation directory under `$ORACLE_HOME/BDSJaguar-4.1.2`.
5. Run the installer with the `--reconfigure` command line parameter. For example:

```
$ cd $ORACLE_HOME/BDSJaguar-4.1.2/cdh510-6-node1.<domain>.com
$ ./bds-database-install.sh --reconfigure
```

5.2 Adding or Removing Oracle Big Data SQL on Oracle Database Nodes

Add or remove Oracle Big Data SQL from Oracle Database nodes by manually running the install or uninstall program on individual database nodes as needed.

On the Oracle Database side, there is at this time no equivalent to the Hadoop-side Jaguar `reconfigure` operation, which installs or uninstalls Oracle Big Data SQL on cluster nodes as required.

To Install Oracle Big Data SQL on Additional Nodes of Oracle Database RAC System

If you want to extend Oracle Big Data SQL from a Hadoop cluster to additional Oracle Database nodes at any time after the initial installation of Oracle Big Data SQL, then install the database-side installation bundle for that cluster on these database nodes using the same configuration applied to other nodes of the database.

> **Note:**
>
> The configuration for the connection between a Hadoop cluster and Oracle Database must be the same for all nodes of the database.

- You may find the bundle at <*Big Data SQL Install Directory*>/BDSJaguar/db-bundles or at the temporary location on other database nodes where you copied the bundle in order to stage the installation. The bundle ZIP file should include the Hadoop cluster name and a time stamp: `bds-4.1.2-db-<`*cluster*`>-<`*yymmdd.hhmi*`>.zip`
- If you can no longer locate the original database-side installation bundle, but do have the configuration file that was used for other nodes of the database, then you can recreate the same bundle by running the Jaguar `reconfigure` command.

```
[root@myclusteradminserver:BDSjaguar]# ./jaguar reconfigure 
my_original_configfile.json
```

Deploy and install the bundle on the new database nodes as described in Steps for Installing on Oracle Database Nodes.

To Uninstall Oracle Big Data SQL from Oracle Database Nodes

See Uninstalling From an Oracle Database Server.

6 Reconfiguring an Installation

After the initial Oracle Big Data SQL installation, there may be changes on the Hadoop cluster or the Oracle Database system that require corresponding updates to the Oracle Big Data SQL configuration, including:

- A change in the status of Kerberos — if it was previously disabled and is now enabled, or vice versa.
- Implementation of TLS (Transport Layer Security) or other security measures, such as Database Authentication option now supported by Oracle Big Data SQL.
- Changes to the location of Hive service instances within the Hadoop cluster.
- A switch from Ethernet to InfiniBand (or InfiniBand to Ethernet) for networking between the Hadoop cluster and the Oracle Database system.
- A change to the version of Oracle Big Data SQL running on the Hadoop cluster.

The basic procedure for reconfiguring an installation is:

- Edit the `bds-config.json` file as needed.
- Run the Jaguar `reconfigure` command to push the configuration changes to the Hadoop cluster nodes and to generate an updated database-side installation bundle.
- Reinstall Oracle Big Data Data SQL on Oracle Database compute nodes using the new installation bundle. You can just overwrite the current installation. There is no prerequisite to uninstall the current Oracle Big Data SQL installation on either side.

Details are provided below.

> **See Also:**
>
> The addition or removal of Oracle Big Data SQL from Hadoop cluster nodes or nodes in a multinode Oracle Database system is not a considered configuration change. See Expanding or Shrinking an Installation.

Run “jaguar reconfigure” to Reconfigure the Hadoop Side of the Installation and to Generate an Updated Database Installation Bundle

1. Log on to the cluster management server node as `root` and cd to the directory where you extracted the downloaded Oracle Big Data SQL installation bundle.
2. Go to the `BDSJaguar-4.1.2` subdirectory under the path where you unzipped the Oracle Big Data SQL bundle .

   ```
   # cd <Big Data SQL Install Directory>/BDSJaguar-4.1.2
   ```

3. In the `BDSJaguar` directory, run the Jaguar utility. Pass it the `reconfigure` parameter and the configuration file name (`bds-config.json` or other) as arguments. You may or may not need to include the `--requestdb` option.

   ```
   [root@myclusteradminserver:BDSjJaguar] #  ./jaguar reconfigure bds-
   config.json
   ```

   If `bds-config.json` sets `database_auth_enabled` to `"true"`, then include the `--requestdb` option in the install command:

   ```
   [root@myclusteradminserver:BDSJaguar] # ./jaguar --requestdb testdb
   reconfigure
   ```

   The script prompts for the cluster management service administrator credentials and then reconfigures Oracle Big Data SQL on the management server and the cluster nodes according to the new settings in the configuration file. The script output terminates with the following message if the installation completed without error.

   ```
   BigDataSQL: RECONFIGURE workflow completed.
   ```

Copy the Updated Database Installation Bundle to Each Oracle Database Node and Install It

1. Log on as either `root` or `oracle` on the cluster management server and cd to the directory where the new database bundle was generated.

   ```
   # cd <Big Data SQL Install Directory>/BDSJaguar/db-bundles
   ```

   Check that the new bundle is there.

   ```
   [oracle@myclusteradminserver:db-bundles] # ls
   [oracle@myclusteradminserver:db-bundles] # bds-4.1.2-db-<cluster>-
   <yymmdd.hhmi>.zip
   ```

2. Copy the database installation bundle over to any directory on the Oracle Database node, using the `oracle` account on the Oracle system. If there are multiple bundles, be sure to select the correct bundle for cluster that you want to connect to Oracle Database.

   ```
   [oracle@myclusteradminserver:db-bundles] # scp bds-4.1.2-db-
   <cluster>-<yymmdd.hhmi>.zip oracle@<database_node>:/opt/tmp
   ```

3. Log on to the Oracle Database node as the database owner (`oracle` or other) in order to perform the installation.

4. Cd to the directory where you copied the bundle, unzip the bundle, and then run the compressed executable that was extracted from the zip file. For example:

   ```
   $ unzip bds-4.1.2-db-cdh510-170309.1918.zip
   $ ./bds-4.1.2-db-cdh510-170309.1918.run
   ```

5. Change directories to the new directory under ORACLE_HOME and run `bds-database-install.sh`, the database-side installer.

   ```
   $ cd $ORACLE_HOME/BDSJaguar/cdh510-6-node1.my.<domain>.com
   $ ./bds-database-install.sh
   ```

6.1 Reconfiguring an Existing Oracle Big Data SQL Installation

When Oracle Big Data SQL is installed on the cluster management server, the database-side installation bundle that is generated contains configuration information that is based on the current state of the cluster. When you install the bundle on the Oracle Database side, this configuration information is used to set up the connnection between the Hadoop cluster and the database server. After the installation, changes in the Hadoop cluster may require an update to the Oracle Big Data SQL configuration both sides (Hadoop cluster and database server) in order to maintain connectivity.

Here are some Hadoop-side changes which require you to reconfigure the installation:

- 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.

Note:

The `reconfigure` procedure cannot be used to update the installation of an older version of Oracle Big Data SQL. You should instead overwrite the old installation with the current Oracle Big Data SQL version.

1. As root on the Hadoop cluster management server, make any necessary updates to the configuration file for the connection between the Hadoop cluster and Oracle Database (`bds-config.json` or other).
2. Go to the `BDSJaguar` directory and run Jaguar to invoke the `reconfigure` operation. Pass in the configuration filename.

   ```
   # cd <Big Data SQL install directory>/BDSJaguar
   # ./jaguar reconfigure bds-config.json
   ```

   This updates the Oracle Big Data SQL installation on the Hadoop cluster and generates a new database-side installation bundle, which you can find at `<`*`Oracle Big Data SQL Install Directory`*`>/BDSJaguar/db-bundles`.

> **Note:**
>
> If you are using Database Authentication and already provided a request key to the database-side installer when you did the original installation, then you do not need to do again. That is, you do not need to include `--requestdb` in a `reconfigure` operation unless you want to set up a connection to another database.

3. As in the original installation, copy the new bundle file to a temporary directory each database node. Connect using the `oracle` account on the database node. For example:

   ```
   # scp bds-4.1.2-db-<cluster>-<yymmdd.hhmi>.zip
   oracle@dbnode:/opt/tmp
   ```

4. Log on to the database node as the `oracle` user.
5. Locate the bundle and unzip it.
6. Run the executable file extracted from the bundle.

   ```
   [oracle@mydbserver: $ bds-4.1.2-db-<cluster>-<yymmdd.hhmi>.run
   ```

   This extracts the installation files into a new directory under ORACLE_HOME.

7. Change directories to ORACLE_HOME and list the sub directories. You should see the old installation directory along with the new one. The directory names will include the same cluster name, but will have different timestamps. Delete the old directory.

   ```
   $ cd $ORACLE_HOME
   $ ls
     bds-4.1.2-db-<cluster>-<new timestamp>
     bds-4.1.2-db-<cluster>-<old timestamp>
   $ rm -rf bds-4.1.2-db-<cluster>-<old timestamp>
   ```

8. Change directories to the new directory.

   Run `bds-database-install.sh` with the `--reconfigure` parameter.

   ```
   $ ./bds-database-install.sh --reconfigure
   ```

7

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.

- General Guidelines for Removing the Software
- Uninstalling From an Oracle Database Server
- Uninstalling From a Hadoop Cluster

7.1 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, the first Oracle Big Data SQL connection installed is known as the "default cluster."

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

7.2 Uninstalling From an Oracle Database Server

On any database server where you want to uninstall the current version of Oracle Big Data SQL, run the commands 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.

Note that the `--crs` command line parameter may or may not be required, depending upon conditions related to Grid support. See the *Command Line Parameter Reference for bds-database-install.sh* in this guide.

Uninstalling Oracle Big Data SQL

Run `bds-database-install.sh` with the `uninstall` command.

```
$ /bds-database-install.sh --uninstall --crs=false
```

> **Note:**
>
> If Grid is not running on this database node, or, if the database does not use Grid (CRS/ASM), then include the optional `--crs` parameter and set it to `false`.
>
> Also note that the `--uninstall-as-primary` and `--uninstall-as-secondary` parameters from previous releases are deprecated in this release. It is no longer necessary to differentiate between primary and secondary clusters in the uninstall process. In addition, `--db-resource` has been deprecated and is therefore not used in the example provided here.

7.3 Uninstalling From a Hadoop Cluster

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.

1. On the cluster management server (where CM or Ambari is running), log on as `root`.
2. Change directories to the `BDSjaguar` directory under the location where the Oracle Big Data SQL Installation bundle `BigDataSQL-<`*`Hadoop distribution`*`><`*`version`*`>.zip` was extracted.
3. Run Jaguar with the `uninstall` command and provide the configuration file previously used to install on the cluster (`bds-config.json` or other):

   ```
   [root@myclusteradminserver:BDSjaguar] # ./jaguar uninstall bds-
   config.json
   ```

When the uninstall is complete, the following message is returned:

```
BigDataSQL: UNINSTALL workflow completed.
```

8

Securing the Oracle Big Data SQL Installation

This section describes security features provided by Oracle Big Data SQL. It also explains measures you can take to secure the Oracle Big Data SQL installation, and provides pointers to the information you need in order to configure the Oracle Big Data SQL installation within secured environments.

> **See Also:**
>
> The Security Overview in the *Oracle Big Data SQL User's Guide* provides more detail on security topics related to user authentication and authorization.

- Security Overview
- Big Data SQL Communications and Secure Hadoop Clusters
- Configuring Oracle Big Data SQL in a Kerberos-Secured Environment
- Using Oracle Secure External Password Store to Manage Database access for Oracle Big Data SQL
- About Data Security on Oracle Big Data Appliance
- Authentication Between Oracle Database and Oracle Big Data SQL Offload Cell Server Processes
- Securing Object Store Access

8.1 Security Overview

This section covers installer, network, file, and password security.

Network Security

In Oracle Big Data SQL, network traffic between the database and the Hadoop cluster, is no longer guaranteed to be over a private InfiniBand network, but can occur over a client network. This network traffic is not currently secured. Therefore when operating a secured Hadoop cluster (e.g., Kerberos-enabled, RPC encryption), Oracle Big Data SQL requires the following:

- All members of the client network are trusted, or
- Private network connectivity is used exclusively for communication between the Hadoop nodes and Oracle database instances.

This private network is commonly referred to as the *Big Data SQL interconnect network*. The interconnect network must be a private network with only trusted users, use at least one switch, and 10 Gigabit Ethernet adapters. Ensure that only the nodes in the Hadoop cluster and Oracle RAC cluster can access the interconnect network. Do not use the interconnect network for other user communication.

Installer File Security

The new Jaguar installer incorporates the following best practices for secure Linux installers and applications:

- No persistent or temporary world-writable files are created.
- No setuid or setgid files are used.

In addition, 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

Password Safety

The Jaguar installer provides these password security measures:

- Passwords for the Ambari and Cloudera Manager cluster management servers are not be passed in on the command line and are not saved in any persistent file during the installation or after the installation is complete.
- Passwords are not logged to any log or trace files.

Security of Related Software

Oracle Big Data SQL relies on other installed software, including third party projects. It is the customer’s responsibility to ensure that such software is kept up-to-date with the latest security patches. This software includes (but is not limited to):

- Python
- Perl

8.2 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.

8.2.1 Connecting a Database to Both Secure and Unsecured Clusters

Oracle Big Data SQL can connect a single database to multiple Kerberos secured or unsecured clusters at the same time.

There is limitation on the use of Hive metadata queries where a database has mixed connections to both secured and unsecured clusters. If the default (the first available connection) is secured, then Hive metadata on unsecured clusters is inaccessible. The reverse is also true – if the default cluster is unsecured, then Hive metadata on secured clusters is inaccessible. This is because the database can only support one Hadoop authentication method.

Suppose the database has these connections: cluster_1 (the default cluster, secured), cluster_2 (unsecured), cluster_3 (secured). In this case, a query such as the following succeeds against cluster_1 since is the default. It also succeeds against cluster_3. However it fails against cluster_2, which is not secured.

```
select cluster_id, database_name, table_name from all_hive_tables;
```

If cluster_1 (the default) were not secured, then instead the query would succeed against cluster_2 and fail against cluster_3.

If you already know the metadata (which you can acquire through other tools), then you can successfully create external tables over Hive sources. For example, the following create table command will work against both secured and unsecured clusters, regardless of the status of the default cluster.

```
 create table myhivetable (
     id number(10),
     name varchar2(4000),
     type varchar2(4000))
   organization external (
     type ORACLE_HIVE
     default directory default_dir
     access parameters (
       com.oracle.bigdata.cluster=<cluster name>
       com.oracle.bigdata.tablename=<Hive database name>.<Hive table name>)
   )
   parallel 2 reject limit unlimited;
```

8.3 Configuring Oracle Big Data SQL in a Kerberos-Secured Environment

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 database node where Oracle Big Data SQL is installed. Also, the OS account that owns the database (`oracle` or another account) must be provisioned as a user principal.

When operating a secured Hadoop cluster (e.g., Kerberos enabled, RPC encryption), Oracle Big Data SQL requires either that all members of the client network be trusted, or that private network connectivity is used exclusively for communication between the Hadoop nodes and Oracle database instances. This private network is commonly referred to as the Big Data SQL interconnect network. The interconnect network must be a private network with only trusted users, use at least one switch, and 10 Gigabit Ethernet adapters. Ensure that only the nodes in the Hadoop cluster and Oracle RAC cluster can access the interconnect network. Do not use the interconnect network for other user communication.

8.3.1 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, but not always) the `oracle` Linux user. When Kerberos is enabled on the Hadoop system, the principal of the database owner or that of another Linux user account can be used for authentication. The following is required in order to give the user access.

The Linux user must be able to be authenticated as a principal in the Kerberos database on the Kerberos Key Distribution Center (KDC) server. The principal on the Hadoop side 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:

- And installation of the Kerberos client software.
- A copy of the Kerberos configuration file from the KDC.
- A copy of the Kerberos keytab file generated on the KDC for the Linux user.
- A valid Kerberos ticket for the Linux user.

Installing the Kerberos Client

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

Creating a Kerberos Principal for the User

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

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 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 <user>
   quit
   ```

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

   > **Note:**
   >
   > If you run the xst(ktadd) command without the `-norandkey` option , this will invalidate other keytabs previously created for this user.

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

   ```
   $ chown <user> <user>.keytab
   $ chmod 400 <user>.keytab
   ```

Kerberos Tasks Automated by Oracle Big Data SQL

The following Kerberos tasks are now automated:

- Distributing Keytab and Kerberos configuration file distrbution files.

  The Oracle Big Data SQL installation can now be configured to automatically distribute the keytab and Kerberos configuration files for the principal to the Hadoop DataNodes (and Oracle Database compute nodes). This is done if the principal name and keytab file location parameters are set in the Jaguar configuration file. This automation is performed on the Hadoop and also on Oracle Database side if the principals are the same. If not, you can designate an alternate principal on the database side.

  On Oracle Big Data Appliance, the keytab file distribution is done by default for the `oracle` account and you do not need to add the principal and keytab file path for this account to the configuration file.

- Acquiring a Kerberos Ticket for designated principals.

  The installation acquires a Kerberos ticket on each Hadoop DataNode and Oracle DB compute node.

- Ticket renewal

  The installation automatically sets up cron jobs in the Hadoop cluster and on Oracle Database to kinit for new ticket for each principal in the configuration four times daily.

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 principal 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.

8.3.2 Installing a Kerberos Client on the Oracle Database Nodes

If the Oracle Database system is Kerberos secured, then Oracle Big Data SQL requires a Kerberos client. The client must be installed on each compute 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 appropriate Oracle Linux repository (ol7 or ol6).

1. Check that the correct Oracle epository ID is installed.

   ```
   # yum repolist
   ```

2. Temporarily disable all repository IDs and then enable the Oracle repository only (Oracle Linux 7 in this example).

   ```
   # yum --disablerepo="*" --enablerepo="public-yum-ol7" 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

8.4 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.

8.5 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.

> **See Also:**
>
> Security for Oracle Big Data Appliance in *Oracle Big Data Appliance Software User’s Guide*.

8.6 Authentication Between Oracle Database and Oracle Big Data SQL Offload Cell Server Processes

The Database Authentication feature described in this guide prevents unauthorized and potentially malicious processes (which can originate from anywhere) from connecting to Oracle Big Data SQL cell server processes in the DataNodes of the Hadoop cluster.

When Ethernet is selected for the connection between Oracle Databases and Oracle Big Data SQL, then by default this secured authentication framework is set up automatically during the installation. Database Authentication is also available as a configuration option for InfiniBand connections.

See Also:

Jaguar Configuration Parameter and Command Reference

8.7 Securing Object Store Access

Oracle Big Data SQL enables object store access from Oracle Database and also from the Oracle Big Data SQL offload server, which runs on Hadoop nodes.

The appropriate proxy setting for the Oracle Big Data offload server is required for access to object stores. These is set by including either the `cell_http_proxy` or `cell_no_proxy` parameters in the Jaguar configuration file.

Note:

Outbound HTTP connections from Hadoop nodes to object stores should be safeguarded. It is recommended that you use a firewall to whitelist target object store domain names.

The credential information required for object store access is encrypted before it is sent from Oracle Database to the Oracle Big Data SQL offload server.

See Also:

- The Jaguar Configuration Parameter and Command Reference shows how to apply the proxy parameters for object store access.
- Granting User Access to Oracle Big Data SQL describes serveral permissions that must be granted on the Oracle Database side before object store access can be authorized.

9
Additional Tools Installed

9.1 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

Although Copy to Hadoop and OHSH can be run from a Hadoop cluster node, or an Oracle Database node, it is recommended to run Copy to Hadoop and OHSH from 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 for the installation on a Hadoop cluster, an 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.

9.1.1 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.

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>;
```

> **Note:**
>
> Later, if you employ the “stage” method to copy Oracle tables, 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. 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.

   ii. 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. Clear the `HADOOP_CLASSPATH` environment variable before invoking Oracle Shell for Hadoop Loaders.
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`.

9.1.2 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 from the Oracle Technology Network. Select the client versions that matches the version of the Oracle Database.

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 and $HADOOP_HOME/bin to the PATH variable

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

6. Clear the HADOOP_CLASSPATH environment variable before invoking Oracle Shell for Hadoop Loaders.

7. 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.

9.1.3 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.

9.1.4 Configuring OHSH to Enable Job Monitoring

When OHSH jobs are executed, status and other information about the job is recorded into a back-end database. To access information from the OHSH command line, you must first configure the connection to the database.

Configuration Steps

Configure the following properties in `conf/smartloader-conf.xml` in order to enable a database instance where job history is stored.

- `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.driver`

  Specifies the JDBC driver for the supported back-end database type. Currently, `MYSQL` and `ORACLE` are valid values. If this property is not specified, the job history commands fail.

Additional properties must be set. These differ, depending upon which database type is defined as the supported back-end database

- If `jdc.driver` = `ORACLE`:
  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.oracleConnectId`

    A TNS entry name defined in the `tnsnames.ora` file.
  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.oracleWalletDir`

    The OS directory containing the Oracle Wallet used to connect to an Oracle Database schema through JDBC.
  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.oracleTnsDir`

    The file path to a directory on the node where OHSH runs. This directory contains SQL*Net configuration files such as `sqlnet.ora` and `tnsnames.ora`. Typically, this is `${ORACLE_HOME}/network/admin`.

    > **Note:**
    >
    > If you are running OHSH from a Hadoop client and want to use Oracle Wallet, copy `tnsnames.ora` and the wallet files to any directory on the Hadoop client.
- If `jdbc.driver` = `MYSQL`:

  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.mysqlConnectUrl`

    The URL used to make a JDBC connection to the MySQL database

  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.mysqlUser`

    MySQL user of job history schema

  - `oracle.hadoop.smartloader.diagnostics.jobhistory.jdbc.mysqlPW`

    Password of the MySQL user account.

Commands for Monitoring OHSH Jobs

After this configuration is completed, you will be able to execute the following OHSH commands:

- `ohsh> show job <`*`job_id`*`>`

  Shows the detailed information of the job specified by ID.

- `ohsh> show job summary <`*`job_id`*`>`

  Shows the performance of the completed job specified by ID.

- `ohsh> show job abstract <`*`job_id`*`>`

  Provides a functional description of the job.

- `ohsh>  show jobs [failed|running|completed|finished] [extended] [<`*`integer`*`>]`

  Shows the last *n* jobs of a particular job status.

  - The first parameter specifies job status. If the status is not specified, all jobs are shown, regardless of job status.
  - The second parameter specifies whether to show details.
  - The third parameter specifies that the last *n* jobs of the specified status should be shown. If *n* is not specified, then all jobs of that status are shown.

- `ohsh>  truncate jobs [<`*`integer`*`>]`

  Removes the last *n* jobs from the database. If the integer is not specified, the command removes all jobs

9.2 Oracle SQL Access to Kafka

If you work with Apache Kafka clusters as well as Oracle Database, Oracle SQL Access to Kafka (OSAK) can give you access to Kafka brokers through Oracle SQL. You can then query data in the Kafka records and also join the Kafka data with data from Oracle Database tables.

OSAK is optional. The OSAK kit is copied to the Oracle Database system when you run Oracle Big Data SQL's database-side installer, `bds-database-install.sh`. If you want to use OSAK, complete the setup as described in Installing and Configuring Oracle Access to Kafka

A

Installation Quick Reference

This condensed description of the installation steps may be useful if you have installed the release before, or, feel that you do not need a detailed explanation of the steps. You can refer back to the full instructions in the main body of the guide for more information.

About Restarts

Hadoop Side:

- Cloudera Configuration Manager (or Ambari) may be restarted by the installation. 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.

Database Side:

- In some cases where Grid infrastructure is present, it must be restarted. If the system uses Grid then you should have the Grid user credentials on hand in case a restart is required
- The Linux users grid and oracle (or other database owner) must both in the same group (usually `oinstall`).

Table A-1 Checking Prerequisites and Patches

Step	For More Information
1. Check for the prerequisites on both the Hadoop or Oracle Database systems. (The installation will alert you if any are missing.)	• Prerequisites for Installation on the Hadoop Cluster • Prerequisites for Installation on Oracle Database Nodes
2. Ensure that the latest Database Proactive Bundle Patch is installed on database as well as any one-off patches required for this product. **NOTE:** The Database Proactive Bundle Patch is different than the Patch Set Update. Refer to the Master Compatibility Matrix for complete details on required patches.	Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1 in My Oracle Support
3. On the database server, check that `opatch` is included in `PATH` . Then run `opatch` to see what patches are installed on the database: `$ ./opatch lspatches`.	
4. Make sure database and listener are down before installing any missing patches.	Managing Oracle Software and Applying Patches (*Database 2 Day + Real Application Clusters Guide*)

Table A-1 (Cont.) Checking Prerequisites and Patches

Step	For More Information
6. Unzip the patch archive on the database server. Navigate to the patch and apply it: `$ ./opatch apply` If the database server has both a database and Grid home, then apply a one-off patch to BOTH the grid home and database home.	
7. Navigate to `$ORACLE_HOME/Opatch` and run the `datapatch` command: `$ ./datapatch`	

Table A-2 Installing the Hadoop Side of Oracle Big Data SQL

<table>
<tr><th>Step</th><th>For More Information</th></tr>
<tr><td><b>1.</b> Download the Oracle Big Data SQL installation bundle from OTN or eDelivery (Oracle Software Delivery Cloud).</td><td>- OTN
- Oracle Software Delivery Cloud</td></tr>
<tr><td><b>2.</b> Copy this file on to master node of the Hadoop cluster (where Ambari or CM is running). Unzip it and navigate to extracted directory.</td><td></td></tr>
<tr><td><b>3.</b> Create a configuration file:
<pre>
{
"cluster": {
  "name": <Your cluster name>
  }
}
</pre>
Some important implicit default settings you may want to modify are <code>database_compatibility</code> and <code>database_auth_enabled</code>. See the description of these parameters in the Jaguar Configuration Parameter and Command Reference in this guide.
<pre>
{
"cluster": {
  "name": <Your cluster name>
  "database_compatibility" : ["18"]
  },
"security": {
  "database_auth_enabled": "false"
  }
}
</pre>
Other configuration options are available.</td><td>- Jaguar Configuration Parameter and Command Reference
- Steps for Installing on the Hadoop Cluster</td></tr>
</table>

Table A-2 (Cont.) Installing the Hadoop Side of Oracle Big Data SQL

Step	For More Information

4. As `root`, run : `./jaguar install <config file>`
If you want to generate a key for Database Authentication, then include the `--requestdb` parameter and list the databases that should accept the key:
`./jaguar --requestdb orcl,testdb,proddb`
`install <config file>`	

5. Copy the database-side installation bundle generated in `<Big Data SQL Install directory>/BDSJaguar/db-bundles` to the database server. If you used `--requestdb` to generate a request key, find the `.reqkey` file under `<Big Data SQL Install directory>/BDSJaguar/dbkeys` and copy it to the database server as well.	

Table A-3 Installing the Database Side of Oracle Big Data SQL

Step	For More Information
1. Log on as the database owner, locate the database-side installation bundle, and unzip it.	Steps for Installing on Oracle Database Nodes
2. Execute the `.run` file extracted from the ZIP file.	

3. Navigate to `$ORACLE_HOME/BDSJaguar-4.1.2` and find the installation directory created by the `.run` file . The directory name will indicate the cluster, node, and domain where the installation bundle originated. Cd into the directory and run the installer:
`$ cd cdh510-6-node1.my.<domain>.com`
`$ ./bds-database-install.sh`
You may need to include some parameters with this command.
	Command Line Parameter Reference for bds-database-install.sh

Perform the steps in the table below if `database_auth_enabled` and/or `impersonation_enabled` was set to "`true`" in the configuration.

Table A-4 Extra Steps to Complete the Configuration for Database Authentication and/or Multi-User Authorization

Step	For More Information

1. If Database Authentication and/or Multi-User Authorization are enabled, the database-side installation generates a ZIP file. Find this file in the installation directory and copy it back to the path indicated below on the Hadoop cluster management server:
`$ ls $ORACLE_HOME/BDSJaguar-4.1.2/cdh510-6-`
`node1.my.<domain>.com/*.zip`
`  mycluster1-18-`
`mycluster1node03.my<domain>.com-`
`myoradb1.<domain>.com.zip`
`$ scp mycluster1-18-`
`mycluster1node03.my<domain>.com-`
`myoradb1.my<domain>.com.zip`
`root@node1:/opt/oracle/DM/databases/conf`
2. Log on to the Hadoop cluster management server as `root` and run the Database Acknowledge operation:
`# cd <Big Data SQL Install Directory>/`
`BDSJaguar`
`# ./jaguar databaseack <config file>`
	Steps for Installing on Oracle Database Nodes

Table A-5 Creating Users and Setting Up Mappings for Multi-User Authorization

Action	For More Information

1. To grant a user access:
• Grant the `BDSQL_USER` role.
• Grant read privileges on the `BigDataSQL` configuration directory object.
`SQL> grant BDSQL_USER to user1;`
`SQL> grant read on directory`
`ORACLE_BIGDATA_CONFIG to user1;`	

Table A-5 (Cont.) Creating Users and Setting Up Mappings for Multi-User Authorization

Action	For More Information
2. To grant users Hadoop access through the Multi-User Authorization feature, use the procedures of the DBMS_BDSQL PL/SQL Package: • Grant the `BDSQL_ADMIN` role to designated administrators in order to allow them to invoke these procedures. • Use the `ADD_USER_MAP` procedure to define the rules for a user. This creates a row in the `BDSQL_USER_MAP` table. • Use `REMOVE_USER_MAP` to remove a row from the `BDSQL_USER_MAP` table.	DBMS_BDSQL PL/SQL Package

See Also

These workflow diagrams provide a visual overview of the installation process as a whole and a close-up view of the steps needed to complete the setup of Database Authentication:

- Installation Workflow
- Generating and Installing the GUID-Key Pair for Database Authentication

The next two sections of this guide provide useful code examples:

- bds-config.json Configuration Examples
- Oracle Big Data SQL Installation Example

B

bds-config.json Configuration Examples

These examples demonstrate some of the available configuration parameters and values.

Example Configuration File: Implicit Default Values

The values that this example shows for the parameters in this group are the implicit default values, which do not need to be specified in the configuration file. If you want to set non-default values, then include the `hadoop_ids` parameters in the configuration file.

```
{
    "cluster": {
        "name": "cluster",
         "database_compatibility" : [ "12.1", "19" ]
    },
    "object_store_support" : {
        "enabled" : "true",
        "cell_http_proxy"     : "http://proxy1.server.com:80",
        "cell_no_proxy"       : 
"localhost,127.0.0.1,.my.<domain>.com,my.<domain>.com",
        "database_http_proxy" : "http://proxy2.server.com:80",
        "database_no_proxy"   : 
"localhost,127.0.0.1,.my.<domain>.com,my.<domain>.com"
    },
    "api" : {
        "hostname"            : "admin.server.com",
        "skip_health_check"   : "true",
        "port" : 5050,
        "restart_stale" : "false"
    },
    "repositories" : {
        "url" : [
            "http://repo1.<domain>.com/loc/hadoop",
            "http://repo2.<domain>.com/loc",
            "http://alternate.<domain>.com/backup/repo"
        ],
        "dir" : [ "/root/hadooprepo" ]
    },
    "network" : {
        "use_infiniband"   : "false",
        "selection_subnet" : "5.32.128.10/21",
        "http_proxy"     : "http://my.proxy.server.com:80",
        "https_proxy"    : "http://mysecure.proxy.server.com:80"
     "excluded_nodes" : [ "edge.node1.<domain>.com", 
"edge.node2.<domain>.com" ]
        },
    "security" : {
        "database_auth_enabled"   : "true",
        "impersonation_enabled"   : "true",
        "impersonation_usehosts"  : "true",
        "impersonation_blacklist" : "hdfs,hive"
    },
    "kerberos" : {
        "principal"   : "oracle/mycluster@MY.<DOMAIN>.COM",
        "keytab"      : "/home/oracle/security/oracle.keytab",
        "hdfs-principal" : "hdfs/mycluster@MY.<DOMAIN>.COM",
        "hdfs-keytab"    : "/home/hdfs/security/hdfs.keytab",
        "db-service-principal" : "oracle/
dbnode.<domain>.com@MY.<DOMAIN>.COM",
        "db-service-keytab"    : "/home/edgedb/edgedb.keytab"

    },
    "edgedb": {
        "node"    : "dbnode.<domain>.com",
        "enabled" : "true",
        "sync_hive_db_list" : "my_hive_db_1,my_hive_db2"
    },
    "memory" : {
        "min_hard_limit" : 16384,
        "max_percentage" : 30
    },
    "hadoop_ids" : {
        "hdfs_user"  : "hdfs",
        "hdfs_group" : "hdfs",
        "hive_user"  : "hive",
        "hive_group" : "hive"
    }
}
```

Example Configuration File: Database Authentication with Required databaseack Operation

The following is an example configuration file that uses database authentication and requires a databaseack operation.

```
{
  "security" : {
    "database_auth_enabled" : "true"
  },
  "network" : {
    "use_infiniband" : "false"
  }
}
```

Example Configuration File: Database Authentication with Object Store Access

The following example uses Database Authentication with object store access.

```
{
    "cluster": {
        "database_compatibility" : [ "19" ]
    },
    "network" : {
        "use_infiniband"   : "false"
    },
    "security" : {
        "database_auth_enabled"   : "true",
        "impersonation_enabled"   : "true",
        "impersonation_usehosts"  : "true"
    },
    "kerberos" : {
        "principal" : "oracle/bdshive@BDAVMSERVICE.ORACLE.COM",
        "keytab" : "/root/oracle.keytab",
        "hdfs-principal" : "hdfs/bdshive@BDAVMSERVICE.ORACLE.COM",
        "hdfs-keytab" : "/root/hdfs.keytab"
    },

    "object_store_support" : {
        "enabled"             : "true",
        "cell_http_proxy"     : "http://www-proxy-hqdc.us.oracle.com:80",
        "cell_no_proxy"       : "",
        "database_http_proxy" : "http://www-proxy-hqdc.us.oracle.com:80",
        "database_no_proxy"   : ""
    }
}
```

C

Oracle Big Data SQL Installation Example

The following shows the console output for a typical Oracle Big Data SQL 4.1.2 installation.

This example shows the full circle of the installation workflow, from Hadoop on the Big Data Appliance, to the Oracle Database system, and then back to Hadoop. The Hadoop-side of the installation is similar on an HDP system. There is no difference for the database side where the connection is to an HDP system.

The database-side installation in this example is on a system running Oracle Database 12.2. But note this output indicates support was enabled in the bds-config.json file for 19c, 18c, and 12.2, excluding 12.1.

```
BigDataSQL: DB version support: 19,18,12.2.
```

The output is the same for an installation on Oracle Database 19c, 18c and 12.2 systems, except for minor path changes. The database side of the installation is the same for Exadata Database Machine and all commodity Oracle Database platforms.

Installation on Oracle Big Data Appliance

Here are things to note about the example:

- The Big Data Appliance in this case is running the Oracle Big Data Appliance release 4.10. Output may differ slightly on other releases.
- We do not use `scl` on this system because on Oracle Big Data Appliance systems running Oracle Linux 6 or 7, Jaguar will automatically use `scl` if needed.
- The `--requestdb` parameter takes the name of the database where you will install the other side of Oracle Big Data SQL will be installed after completing the Hadoop-side installation. The database name in this case is cdborcl. When this parameter is included in the command, Jaguar will a generate a "request key" along with the database-side installation bundle. This key is part of the process that authorizes access to the database. See Important Terms and Concepts for more information about `--requestdb` and about the database authorization process.
- The output file names are branded with the cluster name, the fully-qualified domain name of the node where the files are generated, and a time stamp.
- The line `BigDataSQL: Object Store support: Enabled` indicates that the `bds-config.json` configuration file was set up to enable Oracle Big Data SQL access to Amazon S3, Microsoft Azure, or Oracle OCI.
- Placeholders in brackets are used to fill in hostnames and IP addresses in this example
- You can run these commands as the `oracle` user. However, you will be prompted for the `root` password in order to do the secure copy of files over to the database server.

The bds-config.json file for this example is as follows:

```
{
    "cluster": {
        "database_compatibility" : [ "19" ]
    },
    "network" : {
        "use_infiniband"   : "false"
    },
    "security" : {
        "database_auth_enabled"   : "true",
        "impersonation_enabled"   : "true",
        "impersonation_usehosts"  : "true"
    },
    "kerberos" : {
        "principal" : "oracle/mycluster@MY.<DOMAIN>.COM",
        "keytab" : "/root/oracle.keytab",
        "hdfs-principal" : "hdfs/mycluster@MY.<DOMAIN>.COM",
        "hdfs-keytab" : "/root/hdfs.keytab"
    },

    "object_store_support" : {
        "enabled"             : "true",
        "cell_http_proxy"     : "http://my.proxy.server.com:80",
        "cell_no_proxy"       : "",
        "database_http_proxy" : "http://proxy2.server.com:80",
        "database_no_proxy"   : ""
    }
}
```

Start by running jaguar install on the Hadoop side:

```
[root@scaj22bda17 BDSJaguar]# ./jaguar databasereq --requestdb orcl
BigDataSQL:
=======================================================================
.
BigDataSQL: Oracle Big Data SQL 4.1.2.
BigDataSQL: Starting execution on: Wed Oct 20 12:35:29 2021.
BigDataSQL:          in directory: /opt/oracle/BDSJaguar.
BigDataSQL:          and hostname: <Hadoop node>.<domain>.
BigDataSQL:            IP address: <Hadoop node IP Address>.
BigDataSQL:    DB version support: full.
BigDataSQL:  Object Store support: Enabled.
BigDataSQL: API url: https://<Hadoop node>.<domain>:7183.
BigDataSQL: Waiting for management server to respond...
BigDataSQL: Configuration json file name: bds-config.json.
BigDataSQL: WARNING: Cluster health will not be checked !!!.
BigDataSQL: WARNING: If there are services in the stale state, they
will be restarted !!!.

Cloudera Manager user: admin
Password:

BigDataSQL: Authentication succeeded.
BigDataSQL: API version: 19.
BigDataSQL: Retrieving clusters...
BigDataSQL: One cluster found.
BigDataSQL: setting up cluster: <Hadoop cluster>.
```

```
BigDataSQL: Cluster name   : <Hadoop cluster>.
BigDataSQL: Cluster type   : BDA.
BigDataSQL: Cluster version: CDH5.
BigDataSQL:    full version: 5.14.2.
BigDataSQL: Service: BIGDATASQL.
BigDataSQL: Extracting cloudera manager configuration values.
BigDataSQL: verifying service configuration. .
BigDataSQL: Connection type: Ethernet.
BigDataSQL: Initialization completed..
BigDataSQL: Starting deployment server..
BigDataSQL: Checking for Big Data SQL requirements..
BigDataSQL: Requiring database keys only..
BigDataSQL: reqkey file created: /opt/oracle/BDSJaguar/dbkeys/orcl.reqkey.
BigDataSQL: Inspecting workflow on node scaj22bda16.us.oracle.com.
BigDataSQL: Inspecting workflow on node <Hadoop node>.<domain>.
BigDataSQL: Inspecting workflow on node scaj22bda15.us.oracle.com.
BigDataSQL: No error reported on feedback logs.
BigDataSQL: DATABASEREQ workflow completed.
[root@scaj22bda17 BDSJaguar]#
```

Copy the database-side installation bundle to the database server.

```
[root@<Hadoop cluster> BDSJaguar]# scp db-bundles/bds-4.1.2-db-<Hadoop
cluster>-201014.1336.zip oracle@<Hadoop cluster>:/home/oracle/jaguar
The authenticity of host '<Oracle Database server> (<Oracle DB IP Address>)'
can't be established.
RSA key fingerprint is SHA256:<SHA>.
RSA key fingerprint is MD5:<fingerprint>.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '<Oracle Database server>' (RSA) to the list of
known hosts.
oracle@<Oracle Database server>'s password:
bds-4.1.2-db-<Hadoop
cluster>-201014.1336.zip

           100%  199MB 230.1MB/s   00:00
[root@<Hadoop cluster> BDSJaguar]#
```

Log on to the Oracle Database server, unzip the installation bundle, execute the run file (which creates the staging directory). Copy the request key file into the install directory created by the run file. Then, cd to the staging directory and run the installer. Run the generated "root script" as prompted by the installer.

```
[oracle@<Hadoop cluster> jaguar]$ unzip bds-4.1.2-db-<Hadoop
cluster>-201014.1336.zip
Archive:  bds-4.1.2-db-<some dir>-201014.1336.zip
  inflating: bds-4.1.2-db-<some dir>-201014.1336.run
[oracle@<some dir> jaguar]$ ./bds-4.1.2-db-<Hadoop cluster>-201014.1336.run
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: platform is               :
Linux
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: script started at         :
Wed Oct 14 13:55:07 PDT 2021
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: running location          : /
```

```
home/oracle/jaguar
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run:
ORACLE_SID                : orcl
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run:
ORACLE_HOME               : /u03/app/oracle/18.1.0/dbhome_orcl/
shiphome/database
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: install
directory         : /u03/app/oracle/18.1.0/dbhome_orcl/shiphome/
database/BDSJaguar-4.1.2/<Hadoop cluster>-4-node04.<domain>
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: previous installer
exists
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: extracting...
bds-4.1.2-db-<Hadoop cluster>-201014.1336.run: installer extracted
successfully
```

Unzip the bundle, execute the run file, and copy the request key file into the install directory created by the run file.

```
[oracle@<Hadoop node> <Hadoop cluster>-4-<Hadoop cluster>.<domain>]$ ./
bds-database-install.sh --install
bds-database-install: platform is               : Linux
bds-database-install: script started at         : Wed Oct 20 13:56:52
PDT 2021
bds-database-install: running location          : /u03/app/oracle/
18.1.0/dbhome_orcl/shiphome/database/BDSJaguar-4.1.2/<Hadoop name>-4-
<Hadoop node>.<domain>
bds-database-install: ORACLE_SID                : orcl
bds-database-install: ORACLE_HOME               : /u03/app/oracle/
18.1.0/dbhome_orcl/shiphome/database
bds-database-install: Using principal oracle/mycluster@MY.<DOMAIN>.COM
in bds_oracle.keytab
bds-database-install: configuration spfile      : spfileorcl.ora
bds-database-install: Grid home                 : not detected
bds-database-install: db resource               : orcl
bds-database-install: mta extproc sid           :
bds_orcl_bdsqlkerberos
bds-database-install: use of crsctl             : false
bds-database-install: Kerberos variables file   : /u03/app/oracle/
18.1.0/dbhome_orcl/shiphome/database/bigdatasql/clusters/<Hadoop
cluster>/bds_kerberos_variables.env
bds-database-install: root shell script         : /u03/app/oracle/
18.1.0/dbhome_orcl/shiphome/database/install/bds-database-
install-69481-root-script-<Hadoop cluster>.sh
please run as root:

/u03/app/oracle/18.1.0/dbhome_orcl/shiphome/database/install/bds-
database-install-69481-root-script-<Hadoop cluster>.sh

waiting for root script to complete, press <enter> to continue
checking.. q<enter> to quit
bds-database-install: root script seem to have succeeded, continuing...
bds-database-install: cluster type              : bda
bds-database-install: connection type           : Ethernet
bds-database-install: cluster name              : <Hadoop cluster>
```

```
bds-database-install: cluster version           : cdh5
bds-database-install: dictionary cluster name   : <Hadoop cluster>
bds-database-install: Cluster Hadoop version    : 2.6.0-cdh5.14.2
bds-database-install: Hadoop client version     : hadoop-2.6.0-cdh5.14.2
bds-database-install: Hive client version       : hive-1.1.0-cdh5.14.2
bds-database-install: HBase client version      : Not available on the
cluster
bds-database-install: BDSQL version             : 4.1.2
bds-database-install: BDSQL install date        : Wed Oct 20 11:19:11 2021
bds-database-install: bd_cell version           :
bd_cell-12.2.1.1.0_LINUX.X64_211006-1.x86_64
bds-database-install: cp2hadoop kit version     : orahivedp-5.1.2
bds-database-install: ohsh kit version          : ohsh-5.1.2
bds-database-install: orakafka kit version      : orakafka-1.2.3
bds-database-install: kvclient version          : kv-ee-20.2.16
bds-database-install: cell config dir           : /etc/oracle/cell/network-
config
bds-database-install: hybrid cell model         : false
bds-database-install: cell network(s)           : <Cluster node IP
Address>/20
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: listener end point        : EXTPROC1521
bds-database-install: installing JDK            : jdk1.8.0_231
bds-database-install: Installing enterprise kvclient driver
bds-database-install: creating cellinit backup files
bds-database-install: action                    : setup
bds-database-setup: validating <Oracle DB IP Address>/20 against network
interfaces
bds-database-setup: found <Oracle DB IP Address>/20 at bondeth0
bds-database-setup: ping to cluster nodes from <Oracle DB IP Address> with
1000 bytes
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: Network mask and MTU not need to be validated for
Ethernet connection
bds-database-setup: ping to cluster nodes from <Oracle DB IP Address> with
8400 bytes
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: response ok from <Cluster node IP Address>
bds-database-setup: Database current version is 18.0.0.0.0
bds-database-setup: Webhdfs is only available with database 19
bds-database-setup: webhdfs support will be disabled
bds-database-setup: Previous wallet detected, won't be replaced
bds-database-setup: installing cluster configuration files
bds-database-setup: Extracting jars and cofiguration to /u03/app/oracle/
18.1.0/dbhome_orcl/shiphome/database/bigdatasql/clusters/<Hadoop cluster>
bds-database-setup: unpacking hadoop-2.6.0-cdh5.14.2.tar.gz
bds-database-setup: unpacking files in hadoop-2.6.0-cdh5.14.2/hadoop/lib/
native
bds-database-setup: unpacking hadoop-2.6.0-nativelib-ol7.tar.gz
hadooputils.sh: line 1142: cd: /u03/app/oracle/18.1.0/dbhome_orcl/shiphome/
```

```
database/bigdatasql/clusters/<Hadoop cluster>/hadoop-2.6.0-cdh5.14.2/
share/hadoop/mapreduce1: No such file or directory
bds-database-setup: unpacking hive-1.1.0-cdh5.14.2.tar.gz
ln: failed to create symbolic link â/u03/app/oracle/18.1.0/dbhome_orcl/
shiphome/database/bigdatasql/clusters/<Hadoop cluster>/hadoop/lib/
native/libz.so.1â: No such file or directory
bds-database-setup: creating database objects
bds-database-setup: mta will be setup
bds-database-setup: creating /u03/app/oracle/18.1.0/dbhome_orcl/
shiphome/database/hs/admin/initbds_orcl_<Hadoop cluster>.ora
bds-database-setup: mta shutdown bds_orcl_<Hadoop cluster>
bds-database-setup: removing existing mta parameter settings
bds-database-setup: setting performance parameters for mta
bds-database-setup: setting listener_address to EXTPROC1521
bds-database-setup: starting extproc bds_orcl_<Hadoop cluster>
bds-database-setup: waiting upto 100 seconds for extproc to register
with listener
..ok
bds-database-setup: patching view LOADER_DIR_OBJS
bds-database-setup: creating mta dblinks
bds-database-setup: Verifying orakafka toolkit...
Check all files/dirs ownership - passed
Check directory privileges - passed
Check expected executable files - passed
bds-database-setup: Oracle SQL Access to Kafka toolkit is already
installed
bds-database-setup: installing cp2hadoop
bds-database-setup: installing ohsh
bds-database-setup: enabling TLS encryption
bds-database-setup: reconfiguring database instance orcl
bds-database-setup: adding parameter _bigdata_external_table=true
bds-database-setup: setup cell contiguration
bds-database-setup: celliniteth.ora already enabled with UDP protocol
bds-database-setup: cellinit.ora already has
_cell_disable_ipcdat_on_client=true
bds-database-setup: cellaffinity.ora already disabled
database instance needs to be restarted, enter <yes> for restart it
now or <no> for restart it later:
yes
bds-database-setup: restarting database instance
bds-database-setup: mta shutdown bds_orcl_<Hadoop cluster>
bds-database-setup: removing existing mta parameter settings
bds-database-setup: setting performance parameters for mta
bds-database-setup: setting listener_address to EXTPROC1521
bds-database-setup: starting extproc bds_orcl_<Hadoop cluster>
bds-database-setup: waiting upto 100 seconds for extproc to register
with listener
............ok

+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+
_+_+_+_+_+_+
bds-database-setup: Object Store / Webhdfs scripts templates have been
created:
bds-database-setup: set_parameters_cdb.sql     should be executed in
CDB$ROOT
```

```
bds-database-setup: allow_proxy_pdb.sql        should be executed on
required PDBs
+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+_+
_+_+_+

/u03/app/oracle/18.1.0/dbhome_orcl/shiphome/database/BDSJaguar-4.1.2/<Hadoop
cluster>-4-<Hadoop node>.<domain>/set_parameters_cdb.sql
/u03/app/oracle/18.1.0/dbhome_orcl/shiphome/database/BDSJaguar-4.1.2/<Hadoop
cluster>-4-<Hadoop node>.<domain>/allow_proxy_pdb.sql

bds-database-setup: please review and update them if required

bds-database-setup: activation zip file created: <Hadoop cluster>-4-<Hadoop
node>.<domain>-<Oracle DB hostname><domain>.zip

# /u03/app/oracle/18.1.0/dbhome_orcl/shiphome/database/BDSJaguar-4.1.2/
<Hadoop cluster>-4-<Hadoop node>.<domain>/<Hadoop cluster>-4-<Hadoop
node>.<domain>-<Oracle DB hostname>.<domain>.zip

bds-database-setup: copy this file to root@<Hadoop node>.<domain>:/opt/
oracle/DM/databases/conf
bds-database-setup: and run:

./jaguar databaseack

bds-database-setup: configuration successful for cluster <Hadoop cluster>
bds-database-setup:

===================================================================
For correct operation please ensure appropriate database
patches are in:
/u03/app/oracle/18.1.0/dbhome_orcl/shiphome/database

For more information, please check MOS note:
Oracle Big Data SQL Master Compatibility Matrix (Doc ID 2119369.1)

===================================================================


bds-database-setup: log files in /u03/app/oracle/18.1.0/dbhome_orcl/shiphome/
database/install
bds-database-setup: bds-database-install-69481-* files for more detailed info
bds-database-setup: full log in bds-database-install-69481-main-script.log
[oracle@<Oracle DB hostname> <Hadoop cluster>-4-<Hadoop node>.<domain>]$
```

Copy the zip file generated by bds-database-install.sh back to the Hadoop cluster management node (usually node03. This zip file contains the .ackkey file needed to complete the Database Authentication setup.

```
[oracle@<Oracle DB hostname>8 <Hadoop cluster>-4-<Oracle Database
server>.<domain>]$ scp <Hadoop cluster>-4-<Oracle Database server>.<domain>-
<Oracle DB hostname>.<domain>.zip root@<Hadoop cluster>-3-
node03.<domain>:/opt/oracle/DM/databases/conf
```

```
root@<Oracle Database server>.<domain>'s password:
<Hadoop cluster>-4-<Oracle Database server>.<domain>-<Hadoop
cluster><Hadoop node>.<domain>.zip      100%  670     1.1MB/s   00:00
[oracle@scaj51bda08 <Hadoop cluster>-4-<Oracle Database
server>.<domain>]$
```

Log on as root to the Hadoop cluster management node and run the Jaguar database acknowledge operation.

```
[root@<Hadoop cluster> BDSJaguar]# ./jaguar databaseack
BigDataSQL:
=======================================================================
.
BigDataSQL: Oracle Big Data SQL 4.1.2.
BigDataSQL: Starting execution on: Wed Oct 20 14:02:11 2021.
BigDataSQL:          in directory: /opt/oracle/BDSJaguar.
BigDataSQL:          and hostname: <Hadoop cluster>-4-node03.<domain>.
BigDataSQL:            IP address: <Oracle DB IP Address>.
BigDataSQL:    DB version support: full.
BigDataSQL:  Object Store support: Enabled.
BigDataSQL: API url: https://<Oracle Database server>.<domain>:7183.
BigDataSQL: Waiting for management server to respond...
BigDataSQL: Configuration json file name: bds-config.json.
BigDataSQL: WARNING: If there are services in the stale state, they
will be restarted !!!.

Cloudera Manager user: admin
Password:

BigDataSQL: Authentication succeeded.
BigDataSQL: API version: 19.
BigDataSQL: Retrieving clusters...
BigDataSQL: One cluster found.
BigDataSQL: Cluster name   : <Hadoop cluster>.
BigDataSQL: Cluster type   : BDA.
BigDataSQL: Cluster version: CDH5.
BigDataSQL:    full version: 5.14.2.
BigDataSQL: Service: BIGDATASQL.
BigDataSQL: Extracting cloudera manager configuration values.
BigDataSQL: verifying service configuration. .
BigDataSQL: Connection type: Ethernet.
BigDataSQL: Initialization completed..
BigDataSQL: Starting deployment server..
BigDataSQL: Checking for Big Data SQL requirements..
BigDataSQL: Database acknowledge..
BigDataSQL: Stopping bigdatasql service..
BigDataSQL: Refreshing Big Data SQL database connections..
BigDataSQL: Startig bigdatasql service..
BigDataSQL: Setting database key for Big Data SQL..

Big Data SQL setup finished !!!
run bdschecksw tool on database node for sanity verification
```

```
BigDataSQL: No error reported on feedback logs.
BigDataSQL: DATABASEACK workflow completed.
```

D

Downloading the Correct Versions of the Hadoop, Hive, and HBase Clients for a Local Repostory

If you choose to download these dependencies from a local repository, use these instructions to add the correction client versions to the repository.

By default, the Jaguar installer will attempt to download the clients from the Cloudera or HDP repositories on the Internet. If this is access is restricted within your data center, then you can specify a local repository. These can be local directories or NFS paths. They can also be URLs within the local network or on the Internet.

For CDH 5.x :

First, check the content management service (CM or Ambari) and find the version of the Hadoop, Hive, and HBase services running on the Hadoop cluster. The compatible clients are of the same versions. In each case, the client tarball filename includes a version string segment that matches the version of the service installed on the cluster. In the case of CDH, you can then browse the public repository and find the URL to the client that matches the service version.

1. Log on to Cloudera Manager and go to the **Hosts** menu. Select **All Hosts** , then **Inspect All Hosts**.
2. When the inspection is finished, select either **Show Inspector Results** (on the screen) or **Download Result Data** (to a JSON file).
3. In either case, scan the result set and find the service versions.

   In JSON version of the inspector results, there is a `componentInfo` section for each cluster that shows the versions of software installed on that cluster. The format of the data set is as follows:

   ```
   "componentInfo": [
          ...
   {
      "cdhVersion": "CDH5",
      "componentRelease": "1.cdh5.11.1.p0.6",
      "componentVersion": "2.6.0+cdh5.11.1+2400",
      "name": "hadoop"
   },
   ...
   ```

4. Go to https://archive.cloudera.com/cdh5/cdh/5.

> **Note:**
>
> Since February 2021 all Cloudera repos require password authentication, you need to supply your Cloudera credential to access and download both client jars for cdh5 or client rpms for cdh6. If you are running Big Data Appliance please contact oracle support to request a patch with the specific clients you need.

Look in the "hadoop," hive," and "hbase" subdirectories of the CDH5 section of the archive. In the listings, you should find the client tarball packages for the versions of the services installed on the cluster, such as the following:

```
https://archive.cloudera.com/cdh5/cdh/5/hadoop-2.6.0-
cdh5.12.1.tar.gz
https://archive.cloudera.com/cdh5/cdh/5/hbase-1.2.0-cdh5.12.1.tar.gz
https://archive.cloudera.com/cdh5/cdh/5/hive-1.1.0-cdh5.12.1.tar.gz
```

After you identify the correct versions of the clients and download them to the local repository, provide the path in the `repositories` section of the `bds-config.json` file used by the Jaguar installer.

For CDH 6.X:

The `dir` and `url` parameters in the `bds-config.json` configuration file are not supported on Cloudera 6.x systems. For CDH 6.x, set up a local repository prior to running `bds-database-install.sh` (the database-side) installer and include the `--alternate-repo` parameter on the installer command line as described in the Command Line Parameter Reference for bds-database-install.sh.

For HDP:

1. Log on to Ambari. Go to **Admin**, then **Stack and Versions**. On the **Stack** tab, locate the entries for the HDFS, Hive, and HBase services and note down the version number of each as the "`service version`."
2. Click the **Versions** tab. Note down the version of HDP that is running on the cluster as the "`HDP version base`."
3. Click **Show Details** to display a pop-up window that shows the full version string for the installed HDP release. Note this down as the "`HDP full version`"
4. The last piece of information needed is the Linux version ("centos5," "centos6," or "centos7"). Note this down as "`OS version`."

To search though the HDP repository in Amazon S3 storage to find the correct client URLs using this information acquired in this steps, you would need an S3 browser, browser extension, or command line tool. As alternative, you can piece together the correct URLs, using these strings.

For HDP 2.5 and earlier, the URLs pattern is as follows.

```
http://public-repo-1.hortonworks.com/HDP/<OS version>/2.x/updates/<HDP
version base>/tars/{hadoop|apache-hive|hbase}-<service version>.<HDP
full version>.tar.gz
```

Here are some examples. Note that the pattern of the gzip filename is slightly different for Hive. There is an extra "`-bin`" segment in the name.

```
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.3.2.0/tars/
hadoop-2.7.1.2.3.2.0-2950.tar.gz
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.3.2.0/tars/
apache-hive-1.2.1.2.3.2.0-2950-bin.tar.gz
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.3.2.0/tars/
hbase-1.1.2.2.3.2.0-2950.tar.gz
```

For HDP 2.5 and later releases, the pattern is almost the same except that there is an additional `hadoop`, `hive`, or `hbase` directory under the `tar` directory:

```
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.5.6.0/tars/
hadoop/hadoop-2.7.3.2.5.6.0-40.tar.gz
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.5.6.0/tars/
hive/apache-hive-1.2.1000.2.5.6.0-40-bin.tar.gz
http://public-repo-1.hortonworks.com/HDP/centos6/2.x/updates/2.5.6.0/tars/
hbase/hbase-1.1.2.2.5.6.0-40.tar.gz
```

Alternative Method for HDP:

You can get the required software versions from the command line instead of using Ambari.

-
  ```
  # hdp-select versions
  ```

  Copy and save the numbers to the left of the dash as the "`HDP version base`".

-
  ```
  # hadoop version
  # beeline --version
  # hbase version
  ```

  Use the output from these commands to formulate the <*`service version`*>.<*`HDP full version`*> segment for each URL.

E

Oracle Big Data SQL Licensing

E.1 About Oracle Big Data SQL Licensing

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

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

- A separate license must be procured for each Hadoop cluster.
- All data nodes within the Hadoop cluster must be licensed. Partial licensing within a node is not available.
- All disks accessed by Oracle Big Data SQL (either within or outside of the Hadoop cluster) must be licensed. This includes disks used by Kafka or NoSQL stores when Oracle Big Data SQL accesses those sources.
- There is no additional license required for the Oracle Database server side.
- Oracle Copy to Hadoop licensing is included.
- Oracle Big Data SQL includes a Query Server that is used to query data sources including HDFS, Kafka, NoSQL and object stores. Oracle Big Data SQL includes a restricted use license of Oracle Database for storing metadata (i.e. external tables and views) maintained by the Query Server. No other objects may be created in this Oracle Database.

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.

E.2 ANTLR 4.9.1

```
====================================================================
Copyright (c) 2012-2017 The ANTLR Project. All rights reserved.
Use of this file is governed by the BSD 3-clause license that
can be found in the LICENSE.txt file in the project root.
===================================================================
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. Neither the name of the copyright holder nor the names of its
contributors may be
   used to endorse or promote products derived from this software
without specific prior
   written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"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 COPYRIGHT HOLDER OR CONTRIBUTORS 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.
```

E.3 KAFKA Client 2.7.0

```
                                 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.

Fourth Party Dependencies:

com.github.luben » zstd-jni	1.4.4-7
org.lz4 » lz4-java	1.7.1
org.slf4j » slf4j-api	1.7.30
org.xerial.snappy » snappy-java	1.1.7.3

MIT License for slf4j-api-1.7.30:
Copyright 2004-2019 QOS.ch

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF

```
MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO
EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM,
DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR
OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE
USE OR OTHER DEALINGS IN THE SOFTWARE.

    Zstd-jni: JNI bindings to Zstd Library

    Copyright (c) 2015-2016, Luben Karavelov/ All rights reserved.

    BSD License

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

    * Redistributions of source code must retain the above copyright notice,
this
      list of conditions and the following disclaimer.

    * 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.

    THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "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 COPYRIGHT HOLDER OR CONTRIBUTORS 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.

    Copyright for snappy-java-1.1.7.3
    > Copyright 2011 Taro L. Saito

    BSD License (for zstd-jni)
    See above

    "snappy-java" (org.xerial.snappy:snappy-java)
    Copyright 2008,2011 Taro L. Saito
    Apache 2.0

    This product includes software developed by Google
```

```
Snappy: http://code.google.com/p/snappy/ (New BSD License)

This product includes software developed by Apache
PureJavaCrc32C from apache-hadoop-common http://hadoop.apache.org/
(Apache 2.0 license)

== Contributors ==
  * Tatu Saloranta
    * Providing benchmark suite
  * Alec Wysoker
    * Performance and memory usage improvement

"LZ4 and xxHash" (org.lz4:lz4-java)
Copyright 2020 Rei Odaira and the lz4-java contributors.
Copyright 2020 Adrien Grand and the lz4-java contributors.
Apache 2.0
```

E.4 Snappy Compression

```
There are two licenses associated with this technology. The c source
code
that we used for the port contains the license below in the header of
the
module snappy.c (https://github.com/andikleen/snappy-c/blob/master/
snappy.c)
In the GitHub directory for the project, there is a file called
LICENSE
(https://github.com/andikleen/snappy-c/blob/master/LICENSE).
It's contents follow the license from the c code.

Copyright 2011, Google Inc.
All rights reserved.
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions are
met:
    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * 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.
    * Neither the name of Google Inc. nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"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 COPYRIGHT
OWNER OR CONTRIBUTORS 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.
===
Contents of LICENSE file from github directory
The snappy-c code is under the same license as the original snappy source

Copyright 2011 Intel Corporation All Rights Reserved.

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

    * Redistributions of source code must retain the above copyright
notice, this list of conditions and the following disclaimer.
    * 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.
    * Neither the name of Intel Corporation nor the names of its
contributors may be used to endorse or promote products derived from
this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
"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 COPYRIGHT
OWNER OR CONTRIBUTORS 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.
```

Index