

Oracle® Big Data Discovery

Installation and Deployment Guide

Version 1.0.0 • Revision B • April 2015

Copyright and disclaimer

Copyright © 2003, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

Table of Contents

Copyright and disclaimer	2
Preface	6
About this guide	6
Who should use this guide	6
Conventions used in this document	6
Contacting Oracle Customer Support	7

Part I: Before You Install

Chapter 1: Introduction	9
The Big Data Discovery software package	9
Integration of Big Data Discovery with Cloudera Distribution for Hadoop	10
Integration of Big Data Discovery with WebLogic	11
Deployment configurations and diagrams	11
Big Data Discovery administration	16
Security in Big Data Discovery	16
A note about component names	16
Chapter 2: Prerequisites	17
CDH requirements	17
Hardware requirements	18
User access requirements	19
Physical memory and disk space requirements	19
Software requirements	21
Required Linux utilities	22
Database requirements	23
Sample commands for a production database	24
Index requirements	24
Supported browsers	25

Part II: Installing and Deploying Big Data Discovery

Chapter 3: Overview	27
The orchestration script	27
Installation and deployment workflow	30
Chapter 4: Installing and Deploying Big Data Discovery	31
Selecting the Admin Server	31
Downloading the BDD media pack	32
Creating the installation source directory	33

Updating the configuration file	34
Running the orchestration script	46
Troubleshooting orchestration script problems	47
Rerunning the orchestration script	47

Part III: After You Install

Chapter 5: Post-Deployment Tasks	49
Navigating the BDD directory structure	49
Verifying your deployment	52
Verifying the deployed services	53
Verifying Data Processing	53
Updating the CLI whitelist and blacklist	53
Signing in to Studio as an administrator	54
Backing up BDD	54
Replacing certificates	54
Increasing Linux file descriptors	55
Customizing the WebLogic JVM heap size	55
Chapter 6: Creating Multiple Studio Instances	56
About multiple Studio instances	56
Setting up multiple Studio instances	57
Installing the Studio instances	57
Configuring synchronized caching for the Studio instances	58
About synchronized caching	58
Updating portal-ext.properties to synchronize caching for Studio instances	58
Customizing the shared cache configuration files	59
Clearing the cache for multiple Studio instances	60
Chapter 7: Using Studio with a Reverse Proxy	62
About reverse proxies	62
What is a reverse proxy?	62
Types of reverse proxies	62
Example sequence for a reverse proxy request	63
Recommendations for reverse proxy configuration	63
Preserving HTTP 1.1 Host: headers	64
Enabling the Apache ProxyPreserveHost directive	64
Reverse proxy configuration options for Studio	65
Simple Studio reverse proxy configuration	65
Studio reverse proxy configuration without preserving Host: headers	65
Configuring Studio to support an SSL-enabled reverse-proxy	66

Part IV: Uninstalling Big Data Discovery

Chapter 8: Uninstalling Big Data Discovery	68
The uninstallation script	68

Running the uninstallation script69

Preface

Oracle Big Data Discovery is a set of end-to-end visual analytic capabilities that leverage the power of Hadoop to transform raw data into business insight in minutes, without the need to learn complex products or rely only on highly skilled resources.

About this guide

This guide describes how to configure, install, and deploy the Oracle Big Data Discovery product. It also provides information on tasks you can perform after deployment and instructions for uninstalling the product.

This guide relates specifically to Big Data Discovery version 1.0. The most up-to-date version of this document is available on the <http://www.oracle.com/technetwork/index.html>.



Note: This guide does *not* describe how to install Big Data Discovery on the Oracle Big Data Appliance. For information on installing on the BDA, please contact Oracle Customer Support.

Who should use this guide

This guide addresses administrators and engineers who need to install and deploy Big Data Discovery within their existing Hadoop environment.

Conventions used in this document

The following conventions are used in this document.

Typographic conventions

The following table describes the typographic conventions used in this document.

Typeface	Meaning
User Interface Elements	This formatting is used for graphical user interface elements such as pages, dialog boxes, buttons, and fields.
Code Sample	This formatting is used for sample code phrases within a paragraph.
<i>Variable</i>	This formatting is used for variable values. For variables within a code sample, the formatting is <i>Variable</i> .
File Path	This formatting is used for file names and paths.

Symbol conventions

The following table describes symbol conventions used in this document.

Symbol	Description	Example	Meaning
>	The right angle bracket, or greater-than sign, indicates menu item selections in a graphic user interface.	File > New > Project	From the File menu, choose New, then from the New submenu, choose Project.

Path variable conventions

This table describes the path variable conventions used in this document.

Path variable	Meaning
\$MW_HOME	Indicates the absolute path to your Oracle Middleware home directory, which is the root directory for your WebLogic installation.
\$DOMAIN_HOME	Indicates the absolute path to your WebLogic domain home directory. For example, if <code>bdd_domain</code> is the domain name, then the <code>\$DOMAIN_HOME</code> value is the <code>\$MW_HOME/user_projects/domains/bdd_domain</code> directory.
\$BDD_HOME	Indicates the absolute path to your Oracle Big Data Discovery home directory. For example, if <code>BDD1.0</code> is the name you specified for the Oracle Big Data Discovery installation, then the <code>\$BDD_HOME</code> value is the <code>\$MW_HOME/BDD1.0</code> directory.
\$DGRAPH_HOME	Indicates the absolute path to your Dgraph home directory. For example, the <code>\$DGRAPH_HOME</code> value might be the <code>\$BDD_HOME/dgraph</code> directory.

Contacting Oracle Customer Support

Oracle Customer Support provides registered users with important information regarding Oracle software, implementation questions, product and solution help, as well as overall news and updates from Oracle.

You can contact Oracle Customer Support through Oracle's Support portal, My Oracle Support at <https://support.oracle.com>.

Part I

Before You Install



Chapter 1

Introduction

The following sections describe Oracle Big Data Discovery and how it integrates with other software products. They also describe some of the different deployment configurations Big Data Discovery supports.

[*The Big Data Discovery software package*](#)

[*Integration of Big Data Discovery with Cloudera Distribution for Hadoop*](#)

[*Integration of Big Data Discovery with WebLogic*](#)

[*Deployment configurations and diagrams*](#)

[*Big Data Discovery administration*](#)

[*Security in Big Data Discovery*](#)

[*A note about component names*](#)

The Big Data Discovery software package

Oracle Big Data Discovery is comprised of a number of separate components, which are installed and deployed simultaneously. These components are described below.

Studio

Studio is Big Data Discovery's front-end web application. It provides tools that enable users to create and manage data sets and projects, as well as administrator tools for managing user access and other settings. Studio stores its project data and the majority of its configuration in a relational database.

Studio is a Java-based application. It runs inside the WebLogic Server, along with the Dgraph Gateway.

Dgraph Gateway

The Dgraph Gateway is a Java-based interface that routes requests to the Dgraph instances and provides caching and business logic. It also uses the Cloudera Distribution for Hadoop (CDH) ZooKeeper package to handle cluster services for the Dgraph instances.

The Dgraph Gateway runs inside WebLogic Server, along with Studio.

Data Processing

Data Processing collectively refers to a set of processes and jobs that perform discovery, sampling, profiling, and enrichment of source data. Many of the processes run within Hadoop, so Data Processing must be deployed to CDH nodes.

Data Processing CLI

The Data Processing Command Line Interface (CLI) provides a way to manually launch Data Processing jobs and invoke the Hive Table Detector (see below). Because the CLI shares configuration information with Studio, it is automatically deployed to all Managed Server nodes. It can later be moved to any node that has access to the Big Data Discovery deployment.

Hive Table Detector

The Hive Table Detector is a Data Processing component that monitors the Hive database for new or deleted tables and launches a Data Processing workflow when it discovers one. If you enable the CLI to run as a cron job, the Big Data Discovery installer starts the Hive Table Detector immediately after deployment.

The Hive Table Detector is invoked by the CLI, either manually by the Hive administrator or via the CLI cron job.

Dgraph

The Dgraph indexes the data sets produced by Data Processing and stores them on a shared NFS. It also responds to requests users make for records in the data sets.

The Dgraph is designed to be stateless, which allows each Dgraph instance to respond to requests independently of others. Queries are routed to the Dgraph instances by the Dgraph Gateway.

The Dgraph can be hosted on any node in the Big Data Discovery deployment, although it is recommended that you dedicate specific nodes to hosting it. The nodes that host Dgraph instances form a Dgraph cluster inside the BDD cluster.

Dgraph HDFS Agent

The Dgraph HDFS Agent acts as a data transport layer between the Dgraph and the HDFS environment. It exports records to HDFS on behalf of the Dgraph, and imports records from HDFS during data ingest operations.

The HDFS Agent is dependent on the Dgraph. It is deployed to the same nodes the Dgraph is deployed to, starts when the Dgraph starts, and shuts down when the Dgraph shuts down.

Integration of Big Data Discovery with Cloudera Distribution for Hadoop

Cloudera Distribution for Hadoop (CDH) provides a number of Hadoop-related components and tools that BDD requires to process and manage data. CDH 5.3.0 must be installed on your system before you install BDD.



Note: CDH does not need to be installed on all nodes that will host BDD components. Some BDD components only require specific CDH components, and others require none at all. For more information, see [CDH requirements](#).

CDH components have a number of functions within BDD. For example:

- Cloudera Manager provides the BDD installer with information about your CDH cluster at runtime. You can also use Cloudera Manager to administer your CDH cluster after installation, although this is not required.

- The Hadoop Distributed File System (HDFS) stores all of your source data.
- ZooKeeper manages the Dgraph instances.
- Spark runs all Data Processing jobs.

Integration of Big Data Discovery with WebLogic

The WebLogic Server provides a J2EE container for hosting and managing the Studio and Dgraph Gateway J2EE applications. Additionally, WebLogic's Admin Server plays an important role in the installation process, as well as BDD administration after deployment.

You download the installation package for WebLogic Server 12c (12.1.3) along with the BDD installation packages. BDD's installer automatically installs WebLogic on all nodes that will run Studio and the Dgraph Gateway, and deploys Studio and the Dgraph Gateway inside WebLogic Server.



Note: BDD does not currently support integration with an existing installation of WebLogic. You must use the version you download with the BDD packages.

How the WebLogic Server is used

The WebLogic Admin Server serves as a central point of control for the BDD cluster. You must select a single node in your cluster to be the Admin Server; this is the node you will download the BDD software modules to and perform the installation and deployment process from. After deployment, you will perform all script-based administrative tasks—such as starting individual components and updating the configuration for the entire cluster—from this server.

You also use the Administration Console and WLST (the WebLogic Server Scripting Tool) for starting and stopping the Managed Servers that host Studio and the Dgraph Gateway.

BDD does *not* use the following WebLogic Server features (this list is not exhaustive):

- The WebLogic Server message catalog and the default Java Logging API. Instead, BDD uses Log4j logging. However, messages from the WebLogic domain itself are logged using WebLogic Server's message catalog and the Java Logging API.
- The WebLogic Server JDBC modules or resources, as BDD does not require them.
- The WebLogic Server clusters are not used for load balancing and request routing. BDD accepts requests on any Studio instance and uses its own routing service to send the requests to Dgraph instances. BDD can also use an external load balancer in front of its Studio instances.

Deployment configurations and diagrams

Big Data Discovery supports many different deployment configurations. Before installing, you can configure your deployment to have one that best supports your needs. This topic describes three types of deployments suitable for demonstration purposes, development, and production.

While this topic illustrates three types of deployments and lists their possible variations, you can deploy BDD into any configuration that meets your data processing needs; you are not limited to the configurations described in this topic.

Consider the following deployment options:

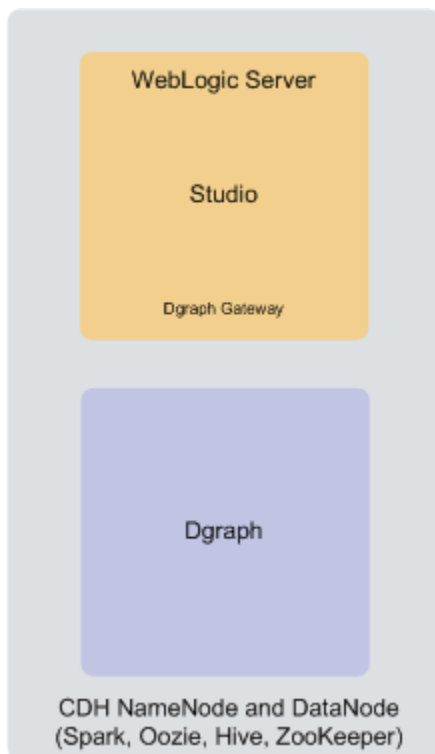
- [Single-node deployment for a demo environment on page 12](#)

- [Two-node deployment for a development environment on page 12](#)
- [Six-node deployment for a production environment on page 13](#)

Single-node deployment for a demo environment

You can deploy BDD to a demo environment running on a single physical or virtual machine. This configuration can only handle a limited amount of data, so it is recommended solely for demonstrating the product's functionality with a small sample index.

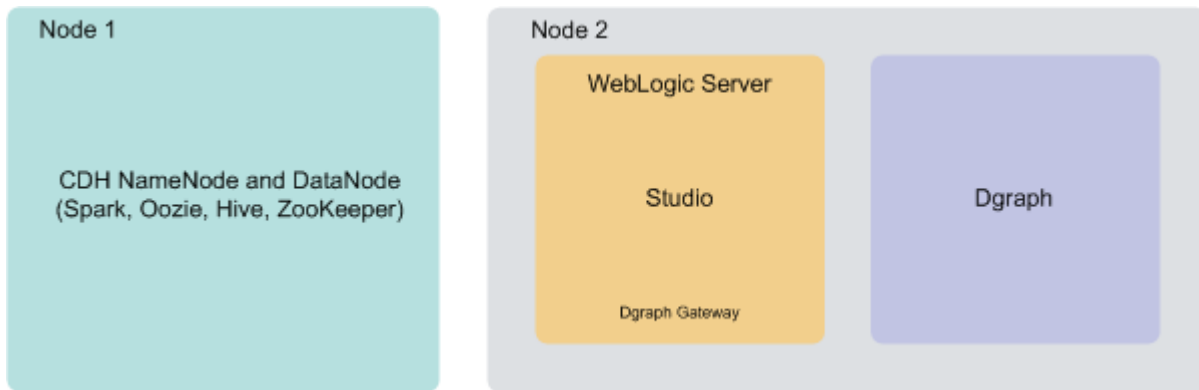
In a single-node deployment, CDH (including the NameNode and one DataNode), the WebLogic Server with Studio and Dgraph Gateway, and the Dgraph instance are all hosted on the same node.



Two-node deployment for a development environment

You can deploy BDD to two nodes for a development environment. This configuration can handle a slightly larger index than a single-node configuration, but is not recommended for production as it does not provide high availability of Dgraph or Studio services and also has limited capacity for processing queries on high volumes of data.

In a two-node configuration, CDH (including the NameNode and one DataNode) is hosted on the first node. The WebLogic Server with Studio and Dgraph Gateway, and the Dgraph instance are hosted on the second node.

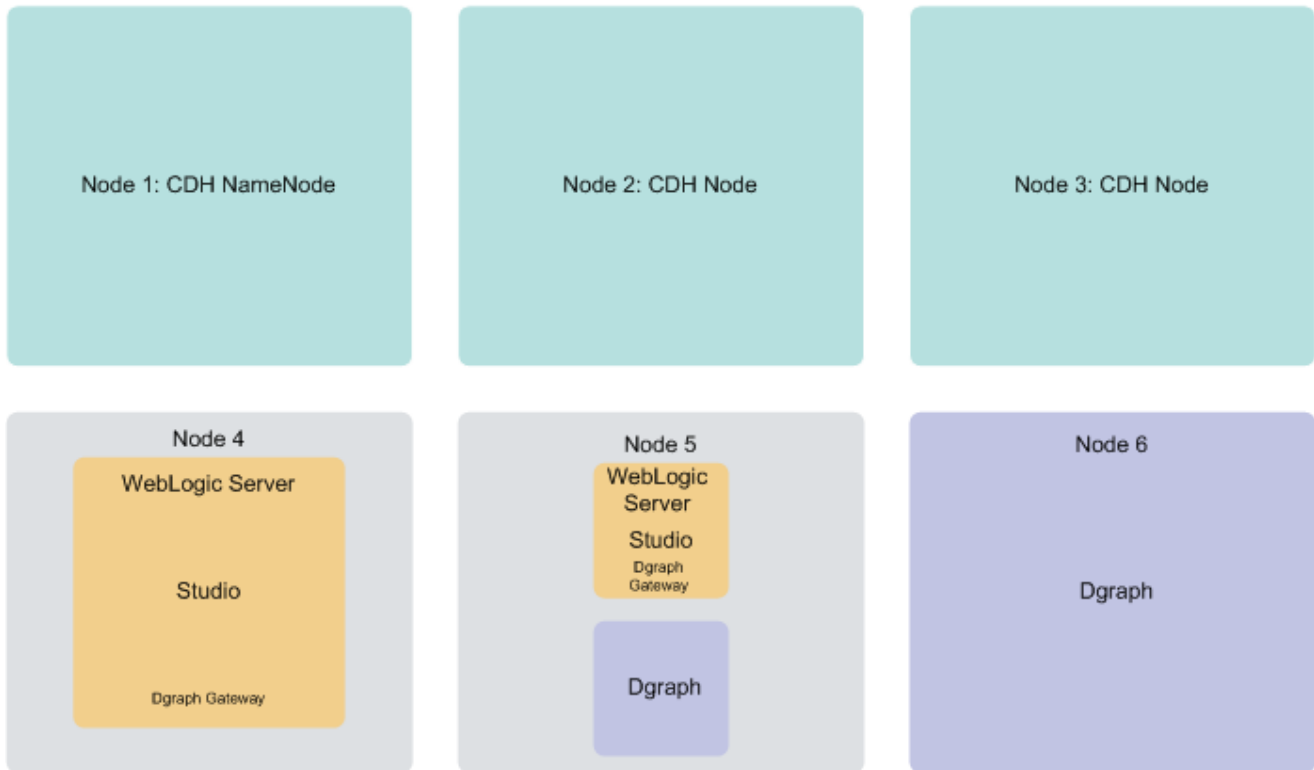


Six-node deployment for a production environment

A production environment can consist of any number of nodes required for scale; however, a cluster of six nodes, with at least three CDH nodes on which Big Data Discovery is deployed, provides minimum availability guarantees.

In this six-node cluster deployment of Big Data Discovery:

- Nodes 1, 2 and 3 are running CDH software. Note that the Big Data Discovery software is also deployed on these nodes. After the installation, Data Processing jobs are launched from these nodes, and run on other BDD nodes. Having three nodes with CDH ensures enhanced availability of services (including query processing performed by the Dgraph), provided by the Big Data Discovery.
- Nodes 4 and 5 are running WebLogic Server with Studio. This ensures minimal redundancy of the Studio instances. (Node 5 is also hosting the Dgraph).
- Nodes 5 and 6 are running the Dgraph instances. This creates a Dgraph cluster within Big Data Discovery cluster, which, in turn, increases the availability of query processing.



Note: You can also set up a multi-node Big Data Discovery cluster in ways that differ from the suggested multi-node layout. For example, at deployment time, you can add more nodes in each category — additional CDH nodes, WebLogic Server nodes, or Dgraph nodes. You can also decide to co-locate CDH with WebLogic Server on some of the nodes, instead of dedicating separate nodes to running WebLogic Server. Similarly, you can decide to host the Dgraph on the same node on which CDH is running. Such decisions may have an impact on overall performance and are dependent on your site's resources and deployment requirements. See section [About co-locating CDH, WebLogic Server, and the Dgraph on page 15](#) in this topic.

About the number of nodes

This documentation does not provide sizing recommendations. To determine an appropriate size for your deployment, use the following guidelines along with your site's specific requirements.



Important: You cannot add nodes after deployment, so you must determine the number of CDH, WebLogic Server, and Dgraph nodes your cluster will include before installing. You should read the following guidelines and configure your cluster according to your requirements to avoid having to reinstall.

The following statements provide high-level guidance on the number of nodes in each category — CDH nodes, WebLogic Server nodes with Studio, and Dgraph nodes:

- **CDH nodes.** The minimum requirement is to have one CDH node in the BDD deployment. For high availability considerations, Oracle recommends having at least three CDH nodes in the BDD deployment. (Note: your pre-existing CDH cluster may have more than three nodes. The CDH nodes that are discussed in this topic are those nodes running CDH on which BDD has also been deployed). The installer will automatically deploy Data Processing to all qualified CDH nodes in the cluster.

- **WebLogic Server nodes.** The minimum requirement is to have one WebLogic Server node running Studio and Dgraph Gateway. There is no recommended number of Studio instances, but if you expect to have a large number of end users generating concurrent query requests to Big Data Discovery, it may be desirable to run two Studio instances (and thus configure two WebLogic Server nodes). If you have more than one WebLogic Server node, Oracle recommends configuring an external load balancer that is connected to the Studio instances running on these nodes. You must specify the number of WebLogic Server nodes in the installer's configuration file before installing.
- **Dgraph nodes.** The minimum requirement is to have one Dgraph instance for each Big Data Discovery cluster deployment. Having more than one Dgraph instance turns the Dgraph instances into a Dgraph cluster, running within the Big Data Discovery cluster. Having a cluster of Dgraphs is desirable as it enhances high availability of query processing. You must specify the number of Dgraph nodes in the installer's configuration file before installing.

About co-locating CDH, WebLogic Server, and the Dgraph

One way to configure your cluster is to co-locate different components on the same nodes. For example, a single node in your BDD cluster deployment can host any combination of CDH, the Weblogic Server, and the Dgraph, including all three components together.

Co-locating enables you to use your hardware more efficiently, since you don't have to devote an entire server to any specific component of Big Data Discovery. However, it also means that the co-located components must compete for memory, which can have a negative impact on performance.

The decision to co-locate different components of Big Data Discovery on the same nodes depends on your site's production requirements and the capacity of the machines running each component.

Here are possible co-location options:

- **Co-location of Dgraph and CDH.** For best performance, Oracle recommends dedicating specific servers to running just the Dgraph process (one Dgraph per machine). You can also co-locate Dgraph instances on the CDH DataNodes, although it is recommended that you use a node that is *not* running Spark. If you decide to co-locate the Dgraph with CDH, you should allocate a specific amount of memory to the Dgraph process using Linux cgroups (control groups) and Dgraph flags; this will prevent it from crashing. Cgroups enable you to control the amount of memory used by the Dgraph at the operating system level, and the Dgraph flags allow you to control it from within the Dgraph. For more information, see the *Administrator's Guide*.
- **Co-location of Dgraph and WebLogic Server instances.** You can co-locate Dgraph instances on the same nodes on which WebLogic Server is deployed. If you use this option, you should configure the WebLogic Server to consume a limited amount of memory to ensure that the Dgraph process has access to sufficient resources for its query processing.
- **Co-location of WebLogic Server and CDH.** You can co-locate WebLogic Server instances on the same nodes on which CDH is deployed. If you use this option, you should configure the WebLogic Server to consume a limited amount of memory to ensure that CDH has access to enough resources for processing.

Big Data Discovery administration

After deployment, you have two options for administering Big Data Discovery: the `bdd-admin` script and Oracle Enterprise Manager.

The `bdd-admin` script

The `bdd-admin` script enables you to perform a number of administrative operations from the command line, such as starting and stopping individual components and updating the configuration of the entire BDD cluster. This script is installed with the rest of the BDD components and can be run from the WebLogic Admin Server. For more information on the `bdd-admin` script, see the *Administrator's Guide*.

Enterprise Manager Plug-in

The Enterprise Manager Plug-in for Big Data Discovery extends Oracle Enterprise Manager Cloud Control to add support for monitoring, diagnosing, and managing BDD components. This is a separate product that enables you to administer your BDD cluster using a graphical user interface. It integrates with BDD but is not included in the installation.

For more information on Enterprise Manager, see the [Oracle Enterprise Manager documentation](#).

Security in Big Data Discovery

BDD does not currently support SSL for the inward-facing ports between its components. Oracle therefore recommends that you deploy Big Data Discovery behind a firewall.

You can, however, enable SSL on Studio's outward-facing ports. You can do this in one or both of the following ways:

- Enable encryption through WebLogic Server. You can do this in the configuration file for the BDD installer. This method activates WebLogic's default demo keystores, which you should replace with your own certificates after deployment. For more information, see [Replacing certificates](#).
- Set up a reverse-proxy server. For instructions on how to do this, see [About reverse proxies](#).



Note: The methods described above do not enable encryption on the inward-facing port on which the Dgraph Gateway listens for requests from Studio.

A note about component names

Some of the installation files and scripts may contain references to the Endeca Server or the Hadoop Exporter. These are legacy names for the Dgraph Gateway and HDFS Agent, respectively. This document refers to the components by their official names, and will make a note of any discrepancies to avoid confusion.



Chapter 2

Prerequisites

The following sections describe the hardware and software requirements your system must meet before you can install BDD.

[CDH requirements](#)

[Hardware requirements](#)

[User access requirements](#)

[Software requirements](#)

[Required Linux utilities](#)

[Database requirements](#)

[Index requirements](#)


[Supported browsers](#)


CDH requirements

CDH 5.3.0 must be installed on your system before you install BDD. The table below describes the specific CDH components BDD requires.

If you are installing on a cluster, the required CDH components do not need to be installed on all nodes that will host BDD components, since some BDD components only require specific CDH components and others don't require CDH at all. The CDH components required on each type of BDD node are specified in [CDH requirements on page 17](#).

If you are installing on a single machine, that machine must have all required CDH components installed.

Component	Description
Cloudera Manager	<p>A web-based user interface that provides administrative capabilities for the CDH cluster. You can use this to perform operations like monitoring the health of the entire cluster, and starting and stopping individual components.</p> <p>When the BDD installer runs, it uses a RESTful API to query Cloudera Manager for information about specific CDH nodes, such as their hostnames and port numbers.</p> <p> Note: Cloudera Manager must be running during installation but is not required once the installation is finished. You can continue using it afterwards, as it provides a number of useful administrative features; however, if you are working in a resource-constrained environment, you may shut it down without affecting BDD's performance.</p>

Component	Description
ZooKeeper	An open source distributed resource coordination package. BDD uses ZooKeeper services to manage the Dgraph instances and ensure high availability of Dgraph query processing.
HDFS	Hadoop's highly fault-tolerant distributed file system. The Hive tables that contain your source data are stored in HDFS.
HCatalog	A metadata abstraction layer that allows you to reference data without using filenames or formats. It insulates users and client programs that need to query data from the data storage. When you create a table in Hive, a table is automatically created in HCatalog. Data Processing's Hive Table Detector monitors HCatalog for new and deleted tables that require processing.
Hive	An open source data warehouse that allows you to query and analyze large amounts of data stored in HDFS. It obtains metadata from HCatalog, enabling you to query your data without knowing its schema or location. All of your source data is stored as Hive tables within HDFS. When BDD discovers a new or modified Hive table, it launches a Data Processing workflow for that table.
Oozie	An open source system for scheduling and managing jobs in Hadoop. BDD relies on Oozie to manage Data Processing workflows.
Spark (Standalone)	An open source parallel data processing framework that compliments Hadoop to make it easy to develop fast, unified big data applications combining batch, streaming, and interactive analytics on all of your data. Spark workers run all Data Processing jobs.  Note: Big Data Discovery requires the Spark (Standalone) service. It does <i>not</i> support Spark on YARN.
Hue	Hadoop User Experience. An open source user interface for a number of Hadoop components.
YARN	An open source data processing framework that provides resource management for distributed applications.

Hardware requirements

The hardware requirements for your specific BDD deployment depend on the amount of data you will process. Oracle recommends the following minimum requirements:

- x86_64 2-core CPU for nodes that will run the Dgraph and HDFS Agent
- x86_64 4-core CPU cores for WebLogic Managed Servers, which will run Studio and the Dgraph Gateway



Note: In this guide, the term "x64" refers to any processor compatible with the AMD64/EM64T architecture. You might need to upgrade your hardware, depending on the data you are processing. All run-time code must fit entirely in RAM. Likewise, hard disk capacity must be sufficient based on the size of your data set. Please contact your Oracle representative if you need more information on sizing your hardware.

User access requirements

The following must be configured for the user who will perform the installation:

- Password-less sudo-to-root enabled on all nodes in the cluster, including CDH nodes
- Bash set as the default shell on all nodes in the cluster, including CDH nodes
- Permission to create the directory in which BDD and the WebLogic Server will be installed on all nodes in the cluster, including CDH nodes (This directory is defined by the `ORACLE_HOME` property in the BDD configuration file.)
- Password-less SSH enabled so that they can log into all other nodes in the cluster (including CDH nodes) from the Admin Server

Additionally, the user who performs the installation must *not* be the root user.

Physical memory and disk space requirements

Each type of node has different physical memory and disk space requirements.



Type of node	Disk space requirements
WebLogic Managed Server and WebLogic Admin Server	<p>All WebLogic Server nodes (both Managed Servers and the Admin Server) have the following requirements:</p> <ul style="list-style-type: none"> • At least 512MB of free swap space • 10GB of available space in the directory defined by the <code>TEMP_FOLDER_PATH</code> property in the BDD configuration file • 6GB of available space in the directory defined by the <code>ORACLE_HOME</code> property in the BDD configuration file <p>Additionally, Managed Servers require at least 5GB of RAM (although your system may require more, depending on the amount of data you will process). The Admin Server requires 6GB of space in the directory defined by the <code>INSTALLER_PATH</code> property in the BDD configuration file.</p>
Dgraph	<p>Dgraph nodes have the following requirements:</p> <ul style="list-style-type: none"> • At least 5GB of RAM (your system may require more, depending on the amount of data you will process) • 10GB of available space in the directory defined by the <code>TEMP_FOLDER_PATH</code> property in the BDD configuration file • 1GB of available space in the directory defined by the <code>ORACLE_HOME</code> property in the BDD configuration file



Type of node	Disk space requirements
CDH	<p>CDH nodes that will run Data Processing have the following requirements:</p> <ul style="list-style-type: none">• 3GB of available space in the directory defined by the <code>TEMP_FOLDER_PATH</code> property in the BDD configuration file• 2GB of available space in the directory defined by the <code>ORACLE_HOME</code> property in the BDD configuration file <p>Additionally, the CDH cluster must have the following YARN properties set to at least 4GB:</p> <ul style="list-style-type: none">• Container Memory (<code>yarn.nodemanager.resource.memory-mb</code>). This is located in the NodeManager Default Group > Resource Management category.• Container Memory Maximum (<code>yarn.scheduler.maximum-allocation-mb</code>). This is located in the ResourceManager Default Group > Resource Management category. <p>To access these properties, open Cloudera Manager, select Clusters > YARN, then click the Configuration tab.</p>

Software requirements

BDD has a number of software requirements. Many of these requirements must be met by all nodes in the cluster, while others only need to be met by node of a specific type. If you are installing on a single node, that node must meet all requirements.

The following table lists the software requirements for each type of node.

Servers	Software requirements
All nodes (including CDH)	<p>The following must be installed on all nodes, including CDH nodes:</p> <ul style="list-style-type: none"> • Oracle Enterprise Linux 6 x86_64 or Red Hat Enterprise Linux 6 • <code>/usr/bin/sudo</code> (this is the default version of <code>sudo</code> on OEL 6) • HotSpot JDK 1.7.0_67 or higher installed in the same location on all nodes. This location is arbitrary, but the BDD installer and this document both assume it is <code>/usr/java/</code>. <p> Note: The JDK 1.7.0_67 is also a prerequisite for CDH 5.3.0, so it should already be installed on all CDH nodes. If the JDK you installed CDH with includes HotSpot, you can copy it from a CDH node to any BDD nodes that do not currently have it installed. Be sure to copy it to the same location on all nodes that will host BDD components.</p> <p>Additionally, all nodes in the cluster must have the following:</p> <ul style="list-style-type: none"> • TTY disabled for <code>sudo</code>, if it is currently enabled. You can do this by changing <code>Defaults requiretty</code> to <code>Defaults !requiretty</code> in the <code>/etc/sudoers</code> file. • The <code>\$JAVA_HOME</code> environment variable set to the same location on all nodes. If the path is set to or contains a symlink, that symlink must be identical on all other nodes. <p> Note: All nodes must also meet a number of access requirements for the user who will perform the installation. For more information, see User access requirements.</p>

Servers	Software requirements
CDH nodes that will run Data Processing once BDD is installed	<p>The following CDH components must be installed on nodes that will host Data Processing:</p> <ul style="list-style-type: none"> • Spark (Standalone) • YARN • HDFS • Hive • Oozie <p> Note: Big Data Discovery requires the Spark (Standalone) service. It does <i>not</i> support Spark on YARN.</p> <p>Data Processing will automatically be installed on CDH nodes that host these components.</p>
WebLogic Admin Server	<p>Perl 5.10 or higher with multi-thread must be installed on the Admin Server.</p> <p>Additionally, you must determine a username and password for the WebLogic Server administrator prior to installing. The password must contain at least 8 characters, one of which must be a number, and cannot start with a number.</p>
WebLogic Managed Servers	<p>Each Managed Server must be able to connect to at least one CDH server to ensure it has access to a ZooKeeper instance. For more information on ZooKeeper and how it affects the cluster deployment's high availability, see the <i>Administrator's Guide</i>.</p>
Dgraph nodes	<p>Dgraph nodes, which the HDFS Agent will also run on, must have read/write access to the shared NFS. Note that at any given time, only one Dgraph instance, the leader, will have write access.</p> <p>If you will be co-locating the Dgraph and CDH, you must enable cgroups and limit the Dgraph's memory consumption.</p> <p> Note: Oracle recommends turning off hyper-threading for nodes hosting the Dgraph. Because of the way the Dgraph works, it is actually detrimental to cache performance to use hyper-threading.</p>

Required Linux utilities

The BDD installer requires several Linux utilities.

The following Linux utilities must be present in the `/bin` directory:

```

basename
cat
chgrp
chown
date
dd
df
mkdir

```

```
more
rm
sed
tar
true
```

The following Linux utilities must be present in the `/usr/bin` directory:

```
awk
cksum
cut
dirname
expr
gzip
head
id
netcat
perl
printf
tail
tr
wc
which
```

If these utilities are not in the specified locations, the installation fails with a message similar to the following:

```
Required dependency is not executable: /bin/df. Aborting.
```

Database requirements

Studio requires a relational database to store configuration and state, including component configuration, user permissions, and system settings. All Studio instances must be able to connect and write to the same database.

Studio supports the following types of databases:

- Oracle 11g or 12c
- MySQL 5.5.3 or higher (this includes CDH's default MySQL database)



Note: BDD does not currently support database migration. If you decide to switch to a different type of database later on, you must uninstall the software and reinstall with a new database instance.

If you are installing BDD in a production environment, you must create the following prior to installation:

- A database of one of the types listed above
- A database username and password
- An empty schema

When creating the database, be aware of the following:

- MySQL databases must use UTF-8 as the default character set.
- The name of the schema is arbitrary, but by default it is `studio`.

Sample commands for creating Oracle and MySQL database users and schemas are available in [Sample commands for a production database](#).

Demo environment database requirements

If you are deploying to a demo environment, you can use one of the databases listed above or a Hypersonic (HSQL) database.

Hypersonic is an embedded database running inside of the JVM. It is useful for getting Studio up and running quickly, but *cannot* be used in a production environment due to performance issues and its inability to support multiple Studio nodes.



Important: If you deploy to a demo environment using a Hypersonic database and later decide to scale up to a production environment, you must reinstall BDD with one of the supported MySQL or Oracle databases listed above.

Sample commands for a production database

The following sections contain sample commands you can use to create users and schemas for Oracle and MySQL databases. You are not required to use these exact commands when setting up your database—these are just examples to help get you started.

Oracle database

You can use the following commands to create a user and schema for an Oracle 11g or 12c database.

```
CREATE USER <username> PROFILE "DEFAULT" IDENTIFIED BY <password> DEFAULT TABLESPACE "USERS"
TEMPORARY TABLESPACE "TEMP" ACCOUNT UNLOCK;
GRANT CREATE PROCEDURE TO <username>;
GRANT CREATE SESSION TO <username>;
GRANT CREATE SYNONYM TO <username>;
GRANT CREATE TABLE TO <username>;
GRANT CREATE VIEW TO <username>;
GRANT UNLIMITED TABLESPACE TO <username>;
GRANT CONNECT TO <username>;
GRANT RESOURCE TO <username>;
```

MySQL database

You can use the following commands to create a user and schema for a MySQL database.



Note: MySQL databases must use UTF-8 as the default character encoding.

```
create user '<username>'@'%' identified by '<password>';
create database <database name> default character set utf8 default collate utf8_general_ci;
grant all on <database name>.* to '<username>'@'%' identified by '<password>' with grant option;
flush privileges;
```

Index requirements

The Dgraph requires an index to store the contents of the data sets it can query. All Dgraph nodes in the cluster must be able to connect to the index.

You can install BDD with an existing BDD-formatted index, if you have one you'd like to use. To do this, you must put the index on the NFS before installing and update installer's configuration file to point to the correct directory. For more information, see [Updating the configuration file](#).

If you don't have an existing index, the installer can create an empty one for you. This is also specified in the installer's configuration file.

Supported browsers

BDD supports the following browsers:

- Firefox ESR
- Internet Explorer 10 and 11 (compatibility mode is not supported)
- Safari on iOS7 for iPad
- Chrome for Business

Part II

Installing and Deploying Big Data Discovery



Chapter 3

Overview

The following sections describe the BDD installer and its behavior, as well as the high-level steps involved in the installation and deployment process.

[The orchestration script](#)

[Installation and deployment workflow](#)

The orchestration script

BDD uses a single script, called the orchestration script, to install and deploy its components all at once. When the script finishes, BDD will be completely installed and your cluster will be up and running.

The orchestration script is contained in one of the BDD installation packages, which you will download to a single directory on the Admin Server. You must perform the entire installation process, including running the orchestration script, from this location.

The same installation package also contains the script's configuration file, `bdd.conf`, which defines the configuration of your cluster and provides the script with information it requires at runtime. You must update this file with information specific to your system and BDD cluster configuration before you run the orchestration script.

Silent installation

Normally, when the orchestration script runs, it prompts you to enter the following information:

- The username and password for Cloudera Manager, which it uses to query Cloudera Manager for information related to your CDH cluster.
- The username and password for the WebLogic Server administrator. The script will create this user when it deploys WebLogic.
- The username and password for your Studio database, which it requires to connect Studio to the database.

You can avoid these steps by running the script in silent mode. To do this, you must add the following environment variables to your system before running the script. When the script runs, it checks for these environment variables and executes silently if it finds them.

This table describes the environment variables required to run the orchestration script in silent mode.

Environment variable	Value
CM_USER	The username for Cloudera Manager.
CM_PASSWORD	The password for Cloudera Manager.

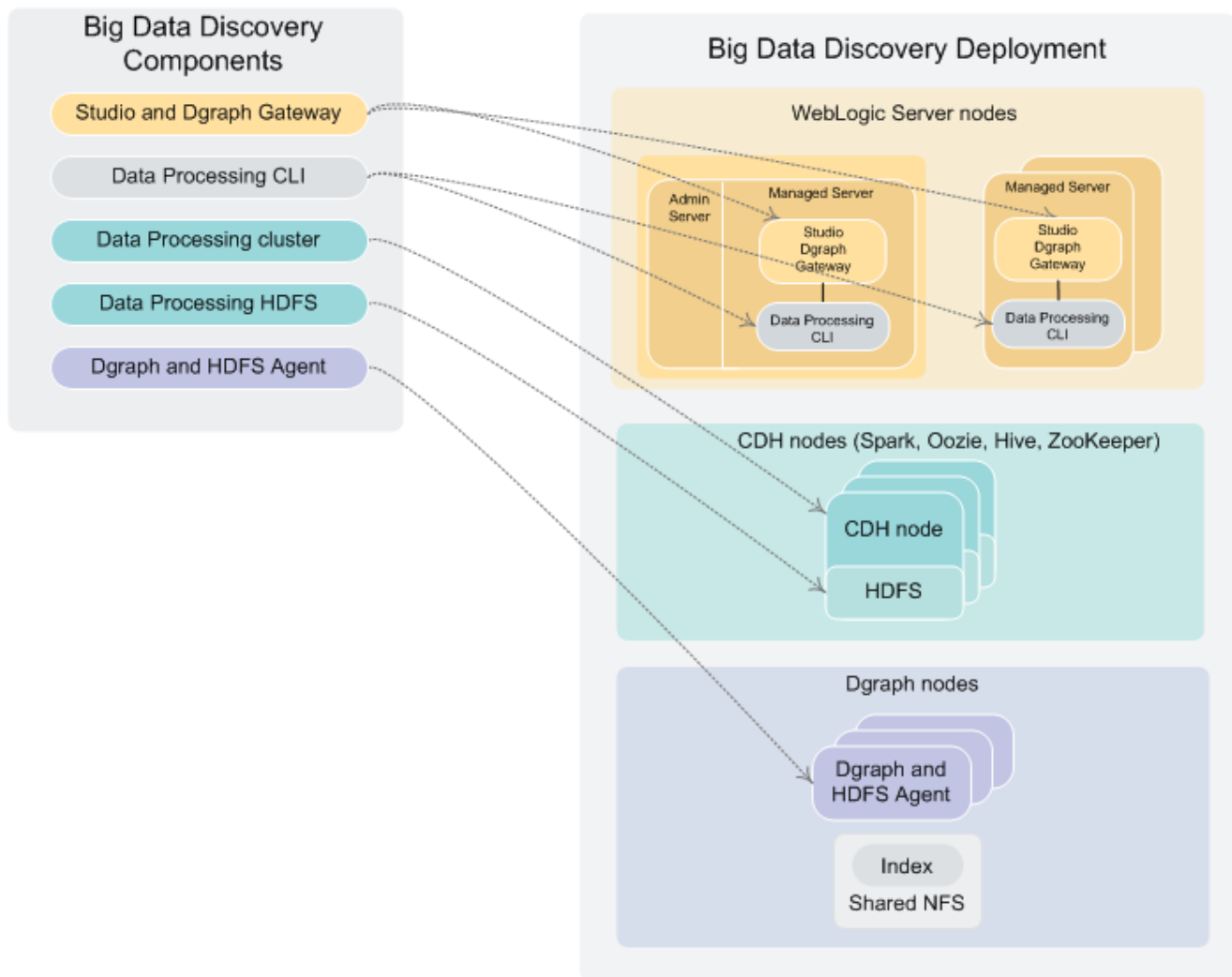
Environment variable	Value
WLS_USERNAME	The username for the WebLogic Server administrator.
WLS_PASSWORD	The password for the WebLogic Server administrator. Remember that this must contain at least 8 characters, one of which must be a number, and cannot start with a number.
STUDIO_JDBC_USERNAME	The username for your Studio database.
STUDIO_JDBC_PASSWORD	The password for your Studio database.

Orchestration script behavior

The following diagram illustrates the behavior of the orchestration script.



Note: This diagram shows how the orchestration script distributes various portions of the BDD installation packages on various nodes in the deployment. This diagram is not intended to show how many nodes you can have in your deployment. For various deployment scenarios, including options for co-locating different parts of the BDD on the same nodes, see [Deployment configurations and diagrams on page 11](#).



When the script runs, it does the following:

1. Reads and validates `bdd.conf`.
2. Prompts you for the user names and passwords for Cloudera Manager, the WebLogic Server administrator, and your database.
3. Queries Cloudera Manager for CDH-related information, including the hostnames and port numbers of specific CDH nodes.
4. Verifies that the Managed Servers nodes and Dgraph nodes meet the minimum CPU and RAM requirements defined in `bdd.conf`.
5. Verifies that the `COORDINATOR_INDEX` defined in `bdd.conf` does not exist.
6. Verifies that the Hive database defined in `bdd.conf` exists.
7. Distributes the installation packages to each node in the cluster according to the configuration defined in `bdd.conf`.
8. Verifies that each node meets all other requirements, including the operating system, and the JDK.

9. If the `FORCE` property in `bdd.conf` is set to `TRUE`, deletes the `ORACLE_HOME` directory from each node.
10. Installs the components:
 1. Installs WebLogic Server (including Studio and the Dgraph Gateway) on the Admin Server node and all Managed Server nodes.
 2. Installs the Dgraph and HDFS Agent on all nodes that will host Dgraph instances.
 3. Installs Data Processing on the HDFS node and all Spark servers.
 4. Installs the Data Processing CLI on all Managed Server nodes.
 5. Installs the `bdd-admin` script on all Managed Server nodes, Dgraph nodes, Spark worker nodes, and YARN node manager servers (not shown in the diagram).
11. Deploys Data Processing:
 1. Deploys Data Processing to the HDFS node and all Spark nodes.
 2. Deploys the CLI to all Managed Server nodes.
 3. If configured to do so, deploys the Hive Table Detector to the specified node and starts it.
12. Deploys WebLogic Server:
 1. Creates the WebLogic domain and the Managed Servers.
 2. Deploys the Dgraph Gateway and Studio as applications within the WebLogic domain.
 3. Deploys WebLogic as a service on all Managed Servers.
 4. Starts all Managed Servers.
13. Deploys the Dgraph and HDFS Agent:
 1. Deploys both components as services to all Dgraph nodes.
 2. If configured to do so, creates the empty Dgraph index files on the NFS.
 3. Starts the Dgraph and HDFS Agent.
14. Verifies that the entire BDD deployment cluster is running.

Installation and deployment workflow

Once you have verified that your cluster satisfies all of BDD's pre-installation requirements, you can begin installing the software. At a high level, this involves the following steps:

1. Select one node in your cluster to be the WebLogic Admin Server and verify that it meets all Admin Server requirements.
2. Download the BDD media pack to the Admin Server.
3. Create the installation source directory (the location from which you will perform the installation) on the Admin Server and move the installation packages there.
4. Update the BDD configuration file with information specific to your cluster.
5. Run the orchestration script.



Chapter 4

Installing and Deploying Big Data Discovery

The following sections describe the full installation and deployment process, from selecting the Admin Server to running the installer. They also provide tips for troubleshooting a failed installation and instructions on rerunning the installer.

[Selecting the Admin Server](#)

[Downloading the BDD media pack](#)

[Creating the installation source directory](#)

[Updating the configuration file](#)

[Running the orchestration script](#)

[Troubleshooting orchestration script problems](#)

[Rerunning the orchestration script](#)

Selecting the Admin Server

The first step in the installation and deployment process is to select a machine to be the Admin Server and verify that it meets all requirements specific to the Admin Server.

You can select any machine you will install BDD on to be the Admin Server—it can be one currently running CDH that will also run BDD components, or one that will run only BDD. If you are installing on a single node, that node will be the Admin Server.



Note: The Admin Server does not require CDH. Although the two can be cohosted on the same machine, WebLogic Server does not depend on CDH and therefore neither does the Admin Server. For more information on cohosting and dependencies, see [Deployment configurations and diagrams](#).

The Admin Server must have the following:

- Oracle Enterprise Linux 6 installed
- Password-less sudo-to-root enabled for the user who will run the orchestration script
- Bash set as the default shell for the user who will run the orchestration script
- JDK 1.7.0_67+ installed
- Perl with multi-thread installed
- Password-less SSH enabled for the user who will run the orchestration script so that they can log in to all other servers in the cluster (including CDH servers)

Once you have selected the Admin Server, you can download the BDD media pack.

Downloading the BDD media pack

Once you have selected the Admin Server, you can download the BDD media pack from the Oracle Software Delivery Cloud.

To download the media pack:

1. On the Admin Server, sign in to the [Oracle Software Delivery Cloud](#) and sign in.
2. Accept the License Agreement and Export Restrictions, then click **Continue**.
3. On the **Media Pack Search** page:
 - (a) From the **Select a Product Pack** drop-down menu, select **Oracle Business Intelligence**.
 - (b) From the **Platform** drop-down list, select **Linux x86-64**.
 - (c) Click **Go**.

The available media packs are displayed in the **Results** table.
 - (d) Select **Oracle Big Data Discovery (1.0.x), Linux x86-64**, then click **Continue**.
4. On the **Oracle Big Data Discovery (1.0.x), Linux x86-64** page, click **Download** next to the following items:

Name	Part Number
Installer for Oracle Big Data Discovery	V75265-01
Documentation for Oracle Big Data Discovery	V75264-01
First of two parts of the Oracle Big Data Discovery binary	V74470-01
Second of two parts of the Oracle Big Data Discovery binary	V74471-01
SDK for Oracle Big Data Discovery	V74473-01
Oracle Fusion Middleware 12c (12.1.3.0.0) WebLogic Server and Coherence	V44413-01



Note: The part number for any component may change if it is updated, but its name will stay the same.

When all of the downloads finish, the following files will be present on your machine:

- V75265-01.zip
- V75264-01.zip
- V74470-01.zip
- V74471-01.zip
- V74473-01.zip
- V44413-01.zip

Once you have downloaded the media pack, you should create the installation source directory and move the installation packages there.

Creating the installation source directory

Once you have downloaded the BDD media pack, you need to create the installation source directory. This is the location from which you will run the orchestration script, as well as the uninstallation script should you choose to uninstall the software.

You must also move the installation packages to the installation source directory and either extract or rename them (depending on the file) so that the orchestration script will recognize them.



Note: The installation source directory must contain at least 6GB of free disk space.

To create the installation source directory:

1. On the Admin Server, choose a directory to be the installation source directory, or create a new one.
The name and location of this directory are arbitrary, but both the configuration file and this document assume it is `/localdisk/BDD_deployer`.
2. Within the installation source directory, create a new directory named `packages`.
3. Move the following files from the download location to the `/packages` directory:
 - `V75265-01.zip`
 - `V74470-01.zip`
 - `V74471-01.zip`
 - `V44413-01.zip`
4. Rename `V74470-01.zip` to `bdd1.zip`, and `V74471-01.zip` to `bdd2.zip`.
This ensures that the orchestration script will recognize the BDD installation packages.
5. Extract `V44413-01.zip`:

```
unzip V44413-01.zip
```

This creates a file called `fmw_12.1.3.0.0_wls.jar`, which contains the WebLogic Server installer.

6. Navigate back to the `/BDD_deployer` directory and extract `V75265-01.zip`:

```
cd ..
unzip packages/V75265-01.zip
```

This creates a new directory within `/BDD_deployer` called `installer`, which contains the orchestration script, `bdd.conf`, and other files required by the orchestration script.

When you are finished, the `/packages` directory will contain the following:

- `bdd1.zip`
- `bdd2.zip`
- `fmw_12.1.3.0.0_wls.jar`

The `/installer` directory will contain the following:

- `bdd.conf`
- `BDD_installer.pl`
- `deploy.sh`
- `install.sh`
- `/linux`
- `setup.sh`
- `undeploy.sh`
- `uninstall.sh`

Once you have moved the installation packages to the installation source directory, you should update the orchestration script's configuration file.

Updating the configuration file

Once you have created the installation source directory, you must configure your deployment by updating the `bdd.conf` file, which is located in the `/BDD_deployer/installer` directory.



Important: The `bdd.conf` file defines the configuration of your BDD cluster and provides the orchestration script with parameters it requires to run. Updating this file is the most important step of the installation and deployment process. If you don't modify the file, or if you modify it incorrectly, the orchestration script could fail or your cluster could be configured differently than you intended.

You can edit the configuration file using any text editor. Be sure to save your changes before closing.


The orchestration script validates the configuration file at runtime and fails if the file contains any invalid values. To avoid this, keep the following in mind when updating the file:

- You must provide a value for all properties *except* `DGRAPH_ADDITIONAL_ARG`, which is only intended for use by Oracle Support.
- The accepted values for some properties are case-sensitive and must be entered exactly as they appear in this document.
- You must provide fully qualified hostnames.
- Any symlinks included in paths must be identical on all nodes. If any are different, or do not exist, the installation may fail.
- Each port setting must have a unique value. You cannot use the same port number more than once.
- Some of the directories defined in the configuration file have location requirements. These are specified in this document.

The following sections describe the properties in the configuration file and any requirements or restrictions they have. The configuration file itself also provides some of this information. Be sure to read the following sections carefully before modifying any properties.

Global settings

The first section in `bdd.conf` configures global settings, which are relevant to all components and the installation and deployment process itself.

Configuration property	Description
INSTALL_TYPE	<p>Sets the installation type according to the hardware you're installing on. This can be set to one of the following:</p> <ul style="list-style-type: none"> • BDA: Use this value if you're installing on the Oracle Big Data Appliance. • GENERIC: Use this value if you're installing on general-purpose hardware. This is the default value. • OPC: Use this value if you're installing on the Oracle Public Cloud. <p>Note that this document does not cover BDA or Cloud installation. For information on installing on either platform, please contact Oracle Customer Support.</p>
CLUSTER_MODE	<p>Determines whether you're deploying to a single machine or a cluster. Use <code>TRUE</code> if you're deploying to a cluster. This is the default value.</p> <p>If you're deploying to a single machine, use <code>FALSE</code>. When deploying to a single machine, you should also be sure that the <code>MANAGED_SERVERS</code>, <code>DGRAPH_SERVERS</code>, and <code>DETECTOR_SERVER</code> properties are set to <code>\${ADMIN_SERVER}</code>, or the orchestration script will fail.</p> <p>Note that this property only accepts UPPERCASE values.</p>
FORCE	<p>Determines whether the orchestration script will remove files and directories left over from previous installations when it runs.</p> <p>When set to <code>TRUE</code>, the orchestration script removes any previous installations from the <code>ORACLE_HOME</code> directory. Use this value if you're rerunning the script after a failed attempt.</p> <p>When set to <code>FALSE</code>, the orchestration script does not remove any previous installations. If one exists, the script will fail. This is the default value.</p> <p>Note that this property only accepts UPPERCASE values.</p>
ORACLE_HOME	<p>The path to the BDD root directory, where BDD will be installed on all nodes in the cluster. This directory will be created by the orchestration script, and therefore should not be an existing one.</p> <p> Important: You must ensure that this directory can be created on all nodes that BDD will be installed on, including CDH nodes that will host Data Processing.</p> <p>On the Admin Server and nodes that will host WebLogic Server, this directory must contain at least 6GB of free space. Nodes that will host the Dgraph require 1GB of free space, and those that will host Data Processing require 2GB.</p> <p>The default value is <code>/localdisk/Oracle/Middleware</code>.</p>

Configuration property	Description
ORACLE_INV_PTR	<p>The path to the Oracle inventory pointer file. This file can't be located in the ORACLE_HOME directory. The default value is /localdisk/Oracle/oraInst.loc.</p> <p>If any other Oracle software products are installed on the machine, this file will already exist. You should update this value to point to that file.</p>
JAVA_HOME	<p>The path to the JDK install directory. This must be the same on all BDD servers. Note that this property is not the same as the JAVA_PATH property. The default value is /usr/java/jdk1.7.0_67.</p>
INSTALLER_PATH	<p>The path to the installation source directory on the Admin Server (the location you moved the installation packages to). This directory must contain at least 6GB of free space. The default value is /localdisk/BDD_deployer/packages.</p>
BDD_HOME	<p>The path to the BDD install directory, which the orchestration script will create on all BDD servers. This directory must be inside ORACLE_HOME. The default value is \${ORACLE_HOME}/BDD1.0.</p>
ENABLE_AUTOSTART	<p>Determines whether the BDD cluster will automatically restart after its servers are rebooted:</p> <ul style="list-style-type: none"> • TRUE: WebLogic (including Studio and the Dgraph Gateway), the Dgraph, and the HDFS Agent will automatically restart after their host servers are rebooted. This is the default value. • FALSE: WebLogic, the Dgraph, and the HDFS Agent must be restarted manually. <p>Note that this property only accepts UPPERCASE values.</p>
TEMP_FOLDER_PATH	<p>The temporary directory used on each node during the installation. The default value is /tmp.</p> <p>On the Admin Server and nodes that will host WebLogic Server or the Dgraph, this directory must contain at least 10GB of free space. Nodes that will host Data Processing require 3GB of free space.</p>

WebLogic settings

The third section in `bdd.conf` configures the WebLogic Server, including the Admin Server and all Managed Servers. It does not configure Studio or the Dgraph Gateway.

Configuration property	Description and possible settings
WLS_START_MODE	<p>Defines the mode WebLogic Server will start in.</p> <p>If set to <code>prod</code>, the WebLogic Server starts in production mode, which requires a username and password when it starts. This is the default value.</p> <p>If set to <code>dev</code>, it starts in development mode, which does not require a username or password. The orchestration script will still prompt you for a username and password at runtime, but these will not be required when starting WebLogic Server.</p> <p>Note that this property only accepts lowercase values.</p>
ADMIN_SERVER	<p>The fully qualified hostname of the machine that will become the WebLogic Admin Server. This should be the machine you are currently working on.</p> <p>There is no default value for this property, so you must provide one. Be sure to provide a value for this property, as the installation script will fail if it is not set.</p>
MANAGED_SERVERS	<p>A comma-separated list of the fully qualified hostnames of the WebLogic Managed Servers (the servers that will run WebLogic, Studio, and the Dgraph Gateway). This list must include the hostname for the Admin Server, and cannot contain duplicate values.</p> <p>If you're installing on a single machine, this property should be set to <code>\${ADMIN_SERVER}</code>, or the orchestration script will fail.</p>
WEBLOGIC_DOMAIN_NAME	<p>The name of the WebLogic domain, which Studio and the Dgraph Gateway run in. The default value is <code>bdd_domain</code>.</p>
ADMIN_SERVER_PORT	<p>The port number used by the Admin Server. This number must be unique. The default value is <code>7001</code>.</p>
MANAGED_SERVER_PORT	<p>The port used by the Managed Server (i.e., Studio). This number must be unique. The default value is <code>7003</code>.</p> <p>This property is still required if you are installing on a single server.</p>

Configuration property	Description and possible settings
WLS_CPU_CORES	<p>This property does not set the number of CPU cores that the WebLogic Server will actually use.</p> <p>Instead, the orchestration script uses the value for this property at runtime to check whether each of the machines has the minimum number of CPU cores required on each Managed Server.</p> <p>The value you enter should be less than or equal to the number of CPU cores available on the node. If you are unsure of how many cores a node has, check its node file.</p> <p>If you enter a value that is greater than the total number of cores available on the node, the script issues a warning but continues to run. If you do not specify a number for this property, the script uses the default value of 4.</p>
WLS_RAM_SIZE	<p>This property does not set the RAM size that the WebLogic Server will actually use.</p> <p>Instead, the orchestration script uses the value for this property at runtime to check whether each of the machines has the minimum amount of RAM available on each Managed Server, in KB.</p> <p>The value you enter (in KB) should be less than or equal to the total amount of RAM available on the node. If you are unsure of how much RAM a node has, check its node file.</p> <p>If you enter a value that is greater than the total amount of RAM available on the node, the script issues a warning but continues to run. If you do not specify a value for this property, the script uses the default value of 2048000 KB.</p>
WLS_SECURE_MODE	<p>Enables and disables SSL for Studio's outward-facing ports.</p> <p>This can be set to <code>TRUE</code> or <code>FALSE</code>. When set to <code>TRUE</code>, the Studio instances on the Admin Server and the Managed Servers listen for requests on the <code>ADMIN_SERVER_SECURE_PORT</code> and <code>MANAGED_SERVER_SECURE_PORT</code>, respectively.</p> <p>The default value is <code>TRUE</code>. Note that this property does not enable SSL for any other BDD components.</p>
ADMIN_SERVER_SECURE_PORT	<p>The secure port on the Admin Server on which Studio listens when <code>WLS_SECURE_MODE</code> is set to <code>TRUE</code>. This number must be unique. The default value is 7002.</p> <p>Note that when SSL is enabled, Studio still listens on the un-secure <code>ADMIN_SERVER_PORT</code> for requests from the Dgraph Gateway.</p>

Configuration property	Description and possible settings
MANAGED_SERVER_SECURE_PORT	<p>The secure port on the Managed Server on which Studio listens when <code>WLS_SECURE_MODE</code> is set to <code>TRUE</code>. This number must be unique. The default value is 7004.</p> <p>Note that when SSL is enabled, Studio still listens on the un-secure <code>MANAGED_SERVER_PORT</code> for requests from the Dgraph Gateway.</p>

CDH settings

The second section in `bdd.conf` contains properties related to Cloudera Manager. The orchestration script uses the values you provide to query Cloudera Manager for information about the other CDH components, such as the URIs and names of their host servers.

Configuration property	Description and possible settings
CM_HOST	The hostname of the server running Cloudera Manager. The default value is <code>\${ADMIN_SERVER}</code> .
CM_PORT	The port number used by the server running Cloudera Manager. The default value is 7180.
CM_CLUSTER_NAME	The name of the CDH cluster, which is listed in the Cloudera Manager. Be sure to replace any spaces in the cluster name with <code>%20</code> . The default value is <code>Cluster%201</code> .


Dgraph Gateway settings

The fourth section in `bdd.conf` configures the Dgraph Gateway.

Configuration Property	Description and possible settings
ENDECA_SERVER_LOG_LEVEL	<p>The log level used by the Dgraph Gateway:</p> <ul style="list-style-type: none"> • DEBUG • INFO • WARN • ERROR • FATAL <p>The default value is <code>ERROR</code>.</p> <p>More information on Dgraph Gateway log levels is available in the <i>Oracle Big Data Discovery Administrator's Guide</i>.</p>

Studio settings



The fifth section in `bdd.conf` configures Studio.


Configuration property	Description and possible settings
SERVER_TIMEOUT	The timeout value (in milliseconds) used when responding to requests sent to all Dgraph Gateway web services except the Data Ingest Web Service. A value of 0 means there is no timeout. The default value is 300000.
SERVER_INGEST_TIMEOUT	The timeout value (in milliseconds) used when responding to requests sent to the Data Ingest Web Service. A value of 0 means there is no timeout. The default value is 1680000.
SERVER_HEALTHCHECK_TIMEOUT	The timeout value (in milliseconds) used when checking data source availability when connections are initialized. A value of 0 means there is no timeout. The default value is 10000.
STUDIO_JDBC_URL	<p>The JDBC URL for the database, which enables Studio to connect to it. There are three templates for this property, but only one can be used. The remaining two must be commented out with a hash symbol (#).</p> <p>The first template is for MySQL 5.5.3 (or later) databases, which use the <code>com.mysql.jdbc.Driver</code> driver. This template is uncommented out by default. If you are using a MySQL database, leave this template uncommented, make sure the other two are commented out, and update the URL as follows:</p> <pre>jdbc:mysql://<database hostname>:<port number>/<database name> ?useUnicode=true&characterEncoding=UTF-8&useFastDateParsing =false</pre> <p>The second template is for Oracle 11g or 12c databases, which use the <code>oracle.jdbc.OracleDriver</code> driver. If you are using an Oracle database, uncomment this template, comment out the other two, and update the URL as follows:</p> <pre>jdbc:oracle:thin: @<database hostname>:<port number>:<database SID></pre> <p>The third template is for Hypersonic databases, which use the <code>org.hsqldb.jdbcDriver</code> driver. Hypersonic is not supported for production environments, so you should only use this instance if you are deploying to a demo environment. If you want the orchestration script to create a Hypersonic database for you, uncomment this template and comment out the other two. The orchestration script will create the database for you in the location defined by the URL.</p> <p> Note: BDD does not currently support database migration. After deployment, the only ways to change to a different database are to reconfigure the database itself or reinstall BDD.</p>

Dgraph and HDFS Agent settings

The sixth section in `bdd.conf` configures the Dgraph and HDFS Agent.

Configuration property	Description and possible settings
DGRAPH_SERVERS	<p>A comma-separated list of the fully qualified hostnames of all Dgraph nodes in the cluster. The orchestration script will install and deploy the Dgraph to these nodes.</p> <p>This list cannot contain duplicate values. Additionally, as Oracle does not recommend cohosting the Dgraph with Spark, this list should not contain hostnames of Spark nodes.</p> <p>If you're installing on a single machine, this property should be set to <code>\${ADMIN_SERVER}</code>, or the orchestration script will fail.</p>
DGRAPH_CPU_CORES	<p>This property does not set the number of cores that the Dgraph will actually use.</p> <p>Instead, the orchestration script uses the value for this property at runtime to check whether each of the machines has the minimum number of CPU cores required for hosting the Dgraph and the HDFS Agent.</p> <p>The value you enter should be less than or equal to the number of cores available on the machine.</p> <p>If you enter a value that is greater than the total number of cores available on the Dgraph nodes, the script issues a warning but continues to run.</p> <p>If you do not specify a number for this property, the orchestration script uses the default value of 2 cores.</p>
DGRAPH_RAM_SIZE	<p>This property does not set the RAM size that the Dgraph will actually use.</p> <p>Instead, the orchestration script uses the value for this property (in KB) at runtime to check whether each of the machines has the minimum amount of RAM required on nodes hosting the Dgraph and the HDFS Agent.</p> <p>The value you enter should be less than or equal to the total amount of RAM available on the node.</p> <p>If you enter a value that is greater than the total amount of RAM available on the Dgraph nodes, the script issues a warning but continues to run.</p> <p>If you do not specify a number for this property, the orchestration script uses the default value of 2048000 KB.</p>
DGRAPH_OUT_FILE	<p>The path to the Dgraph's stdout/stderr file. The default value is <code>\${BDD_HOME}/logs/dgraph.out</code>.</p>

Configuration property	Description and possible settings
DGRAPH_INDEX_DIR	<p>The path to the directory on the shared NFS in which the Dgraph index (defined by DGRAPH_INDEX_DIR) will be located. The orchestration script will create this directory if it does not already exist.</p> <p>The default value is /share/bdd_dgraph_index. If you are installing with an existing index, be sure to change the value of this property to the name of the directory the index is located in.</p> <p> Important: If DGRAPH_INDEX_NAME is set to base, the orchestration script will delete any files in this location and replace them with the empty indexes.</p>
DGRAPH_INDEX_NAME	<p>The name of the Dgraph index, which will be located in the directory defined by DGRAPH_INDEX_DIR. The default value is base.</p> <p> Important: If you do not change this value, the orchestration script will delete all files in the DGRAPH_INDEX_DIR and create an empty index named base. Only use this value if you want to install with an empty index.</p> <p>If you are installing with an existing index, move the index to the directory defined by DGRAPH_INDEX_DIR and change the value of this property to the name of the index you are using. If the index does not exist in the DGRAPH_INDEX_DIR location, the orchestration script will fail.</p> <p>Do not include <code>_indexes</code> in the index's name. For example, if you have an index named <code>product_indexes</code>, you should only specify <code>product</code>.</p>
DGRAPH_THREADS	<p>The number of threads the Dgraph starts with. There is no default value for this property, so you must provide one. Oracle recommends the following:</p> <ul style="list-style-type: none"> • For machines running only the Dgraph, the number of threads should be equal to the number of CPU cores on the machine. • For machines running the Dgraph and other BDD components, the number of threads should be the number of CPU cores minus 2. For example, a machine with 4 cores should have 2 threads. <p>Be sure that the number you use is in compliance with the licensing agreement.</p>
DGRAPH_CACHE	<p>The size of the Dgraph cache, in MB. There is no default value for this property, so you must provide one.</p> <p>You only need to specify the number of MB to allocate to the cache. For example, a value of 50 sets the cache size to 50MB.</p> <p>For enhanced performance, Oracle recommends allocating at least 50% of the node's available RAM to the Dgraph cache. If you later find that queries are getting cancelled because there is not enough available memory to process them, experiment with gradually decreasing this amount.</p>

Configuration property	Description and possible settings
DGRAPH_WS_PORT	The port number the Dgraph web service runs on. This number must be unique. The default value is 7010.
DGRAPH_BULKLOAD_PORT	The port on which the Dgraph listens for bulk load ingest requests. This number must be unique. The default value is 7019.
COORDINATOR_INDEX	The index of the Dgraph cluster in the ZooKeeper ensemble. ZooKeeper uses this value to identify the cluster. The default value is <code>cluster1</code> . Note that this property is not related to the Dgraph index.
DGRAPH_ADDITIONAL_ARG	 Note: This property is only intended for use by Oracle Support. Do not provide a value for this property when installing BDD. Defines one or more flags to start the Dgraph with. More information on Dgraph flags is available in the <i>Oracle Big Data Discovery Administrator's Guide</i> .
AGENT_PORT	The port on which the HDFS Agent listens for HTTP requests. This number must be unique. The default value is 7102.
AGENT_EXPORT_PORT	The port on which the HDFS Agent listens for requests from the Dgraph. This number must be unique. The default value is 7101.
AGENT_OUT_FILE	The path to the HDFS Agent's stdout/stderr file. The default value is <code>\${BDD_HOME}/logs/dgraphHDFSAgent.out</code> .

Data Processing settings

The seventh section in `bdd.conf` configures Data Processing and the Hive Table Detector.

Configuration property	Description and possible settings
HDFS_DP_USER_DIR	The location within the HDFS <code>/user</code> directory that stores the Avro files created when users export data from BDD. The orchestration script will create this directory if it does not already exist. The name of this directory must not include spaces. The default value is <code>bdd</code> .
ENABLE_HIVE_TABLE_DETECTOR	Enables and disables the Hive Table Detector. When set to <code>TRUE</code> , the Hive Table Detector runs automatically on the server defined by <code>DETECTOR_SERVER</code> . When set to <code>FALSE</code> , the Hive Table Detector is not created. The default value is <code>FALSE</code> .

Configuration property	Description and possible settings
DETECTOR_SERVER	<p>The fully qualified hostname of the server the Hive Table Detector runs on. This must be one of the WebLogic Managed Servers. The default value is <code>\${ADMIN_SERVER}</code>.</p> <p>If you are installing on a single machine, this property should be set to <code>\${ADMIN_SERVER}</code>, or the orchestration script will fail.</p>
DETECTOR_HIVE_DATABASE	<p>The name of the Hive database that the Hive Table Detector monitors.</p> <p>The default value is <code>default</code>. This is the same as the default value of <code>HIVE_DATABASE_NAME</code>, which is used by Studio and the CLI. It is possible to use different databases for these properties, but it is recommended that you start with one for a first time installation.</p>
DETECTOR_MAXIMUM_WAIT_TIME	<p>The maximum amount of time (in seconds) that the Hive Table Detector waits between update jobs. The default value is 1800.</p>
DETECTOR_SCHEDULE	<p>A Cron format schedule that specifies how often the Hive Table Detector runs. This must be enclosed in quotes. The default value is <code>"0 0 * * *"</code>, which means the Hive Table Detector runs at midnight, every day of every month.</p>

CLI settings

The final section in `bdd.conf` configures the CLI. These properties are used in both Studio and Data Processing.

Configuration property	Description and possible settings
ENABLE_ENRICHMENTS	<p>Determines whether data enrichments are run during the sampling phase of data processing. This setting controls the Language Detection, Term Extraction, Geocoding Address, Geocoding IP, and Reverse Geotagger modules.</p> <p>When set to <code>true</code>, all of the data enrichments run, and when set to <code>false</code>, none of them run. The default value is <code>true</code>.</p> <p>For more information on data enrichments, see the <i>Data Processing Guide</i>.</p>
JAVA_PATH	<p>The path to the Java binaries within the Java installation, which should be in the same location on each server in the cluster. The default value is <code>\${JAVA_HOME}/bin/java</code>.</p> <p>Note that this property is not the same as <code>JAVA_HOME</code>.</p>

Configuration property	Description and possible settings
MAX_RECORDS	<p>The maximum number of records included in a data set. For example, if a Hive table has 1,000,000 records, you could restrict the total number of sampled records to 100,000.</p> <p>Note that the actual number of records in each data set will sometimes be slightly more than or slightly less than the value of MAX_RECORDS.</p> <p>The default value is 1000000.</p>
SPARK_EXECUTOR_MEMORY	<p>The amount of memory that Data Processing jobs request from the Spark worker nodes. The default value is 48g. You should increase this value if you plan on processing very large Hive tables.</p> <p>The value of this property must be equal to or less than the value of Spark's Total Java Heap Sizes of Worker's Executors in Bytes (executor_total_max_heapsize) property. To access the executor_total_max_heapsize property, open Cloudera Manager and select Clusters > Spark (Standalone), click the Configuration tab, and select the Worker Default Group category.</p>
SANDBOX_PATH	<p>The path to the HDFS directory in which the Avro files created when users export data from BDD are stored. The default value is /user/\${HDFS_DP_USER_DIR}.</p>
LANGUAGE	<p>Specifies either a supported ISO-639 language code (en, de, fr, etc.) or a value of unknown to set the language property for all attributes in the data set. This controls whether Oracle Language Technology (OLT) libraries are invoked during indexing.</p> <p>A language code requires more processing but produces better processing and indexing results by using OLT libraries for the specified language. If the value is unknown, the processing time is faster but the processing and indexing results are more generic and OLT is not invoked.</p> <p>The default value is unknown.</p>
HIVE_DATABASE_NAME	<p>The name of the Hive database that stores the source data for Studio data sets. This is used by Studio as well as the CLI.</p> <p>The default value is default. This is the same as the default value of DETECTOR_HIVE_DATABASE, which is used by the Hive Table Detector. It is possible to use different databases for these properties, but it is recommended that you start with one for a first time installation.</p>

Running the orchestration script

Once you have updated `bdd.conf`, you can install and deploy BDD by running the orchestration script from the Admin Server.

Before you run the orchestration script, you should verify that all of BDD's prerequisites have been satisfied. Specifically, make sure that:

- Your system meets all hardware, software, and disk space requirements described in [Prerequisites](#).
- You meet all of the user access requirements described in [User access requirements](#).
- You are working on the Admin Server, which is properly set up.
- You determined a username and password for the WebLogic Server administrator.
- The Studio database (including its username, password, and schema) is set up.
- If you are installing with an existing Dgraph index, the index files are on the NFS and the `DGRAPH_INDEX_NAME` and `DGRAPH_INDEX_DIR` properties point to the correct location.
- If you want to run the script in silent mode, you set the environment variables described in [Silent installation](#).
- `bdd.conf` is available and properly configured for your deployment.
- The following Hadoop components are running:
 - Cloudera Manager
 - ZooKeeper
 - HDFS
 - Hive
 - Oozie
 - Spark (Standalone) running as a service
 - YARN
 - Hue

The installation procedure is the same for single-node and cluster configurations. The orchestration script uses the configuration file to determine whether it should run in single-node or cluster mode.

To run the orchestration script:

1. On the Admin Server, open a new terminal window and navigate to the installation source directory:

```
cd /localdisk/BDD_deployer/installer
```

2. Run the orchestration script:

```
./setup.sh [bdd.conf]
```

The `bdd.conf` argument is optional, as the script reads this file at runtime by default.

3. If you are not running the script in silent mode, enter the following information when prompted:
 - The username and password for Cloudera Manager.

- The username and password for the WebLogic Server administrator. Remember that the password must contain at least 8 characters, one of which must be a number, and can't begin with a number.
- The JDBC username and password for your database.

The script verifies that your system meets all requirements, then installs and deploys BDD according to the configuration defined in `bdd.conf`. When the script finishes, BDD will be up and running.

Troubleshooting orchestration script problems

If the orchestration script fails, use its console output and log files to determine the cause of the failure.

The orchestration script's console output specifies the steps it performed and whether each passed or failed. For failed steps, the output also indicates the cause of the failure. If a step failed on one or more specific servers, the output will also list the hostnames of those servers. For example:

```
[Installer] Error! Fail to copy Data Processing package to servers: <hostname1, hostname2>
```

You can then check the log files on those servers for more information about the failure. The orchestration script's log files are located in the `/tmp/bdd_tmp/log` directory on each server.

Once you have determined the issue, you can fix it and rerun the orchestration script. For more information, see [Rerunning the orchestration script](#).

Rerunning the orchestration script

After you have fixed the errors that caused the orchestration script to fail, you can reinstall BDD. To do this, you must first run the uninstallation script, set the `FORCE` property in `bdd.conf` to `TRUE`, and rerun the orchestration script.

The uninstallation script removes many of the files created the last time you ran the orchestration script and cleans up your environment. Setting `FORCE` to `TRUE` tells the orchestration script to remove any remaining BDD files before installing again.



Note: If the orchestration script was previously run by a different user, you must delete the `/tmp/tmp_bdd` directory from all nodes beforehand; otherwise, the orchestration script will fail.

To rerun the orchestration script:

1. On the Admin Server, navigate to the installation source directory and run the uninstallation script:

```
./uninstall.sh bdd.conf
```

2. If the orchestration script was previously run by a different user, delete the `/tmp/bdd_tmp` directory from all nodes.
3. Open `bdd.conf` in any text editor and set the value of `FORCE` to `TRUE`.
Be sure to enter `TRUE` in all caps.
4. Rerun the orchestration script.

The orchestration script removes any files created the last time it ran and runs again on the clean system.

Part III

After You Install



Chapter 5

Post-Deployment Tasks

The following sections describe tasks you can perform after you install and deploy BDD, such as verifying your installation and increasing Linux file descriptors.

[Navigating the BDD directory structure](#)

[Verifying your deployment](#)

[Updating the CLI whitelist and blacklist](#)

[Signing in to Studio as an administrator](#)

[Backing up BDD](#)

[Replacing certificates](#)

[Increasing Linux file descriptors](#)

[Customizing the WebLogic JVM heap size](#)

Navigating the BDD directory structure

The BDD installation and deployment process adds a number of new directories to your system. The following sections describe these directories and their contents.


\$BDD_HOME directory structure

\$BDD_HOME is the root directory for the BDD components. The default path to this directory is:

```
$ORACLE_HOME/BDD1.0
```

The following table describes the structure of \$BDD_HOME immediately after deployment.

Directory name	Description
.	Contains the oraInst.loc file.

Directory name	Description
/BDD_manager	<p>Directories related to the BDD Manager utility, including:</p> <ul style="list-style-type: none"> • /bin: Contains the BDD Manager utility script, <code>bdd-admin.sh</code>, which you can use to administer the cluster from the command line. • /conf: Contains <code>bdd.conf</code> and Hadoop configuration files required by the HDFS Agent. • /linux: Additional scripts for administering the BDD cluster. • /log: Log files for the <code>bdd-admin</code> script. • <code>version.txt</code>: Contains version information for the BDD Manager utility. <p>More information on the <code>bdd-admin</code> script is available in the <i>Oracle Big Data Discovery Administrator's Guide</i>.</p> <p> Note: Because this directory is required for updating the cluster configuration after deployment and uninstalling BDD, it is created on all Managed Servers, Dgraph nodes, Spark worker nodes, and YARN node manager servers in the cluster. However, the <code>bdd-admin</code> script can only be run from the Admin Server.</p>
/dgraph	<p>Files and directories related to the Dgraph, including:</p> <ul style="list-style-type: none"> • /bin: Scripts for administering the Dgraph. • /conf: Stylesheets for Dgraph statistics pages and schemas for Dgraph queries and responses. • /dgraph-hdfs-agent: Scripts for administering the HDFS Agent and its libraries. • /empty-indexes: The empty indexes required to start the Dgraph after deployment. • /lib and /lib64: Dgraph libraries. • /msg: Documents defining the format of Dgraph EQL queries. • /nls and /olt: Files related to the OLT. • /ssl: Files for configuring SSL. This directory is not used in BDD 1.0. • <code>version.txt</code>: Contains version information for the Dgraph and HDFS Agent components. • /xquery: XQuery documents for communications between the Dgraph and other services.
/logs	The Dgraph and HDFS Agent log files.

Directory name	Description
/dataprocessing	Contains executables and packages for Data Processing, as well as the following subdirectories: <ul style="list-style-type: none"> • /edp/log: Data Processing log files. • /edp_cli: Contains the CLI and related files. This directory is only available on the Managed Servers.
/security	Files related to BDD security.
/server	Files and directories related to the Dgraph Gateway, including: <ul style="list-style-type: none"> • /apis: Contains APIs used by the BDD web services. • /endeca-cmd: Contains the endeca-cmd command line utility, which you use to administer the Dgraph Gateway from the command line. • /endeca-server: EAR file for the Dgraph Gateway application. • post-install.sh • README.txt • version.txt: Contains version information for the Dgraph Gateway component.
/studio	Scripts and packages for the Studio application.
/crfgtoollogs, /common, /diagnostics, /inventory, /OPatch, /oui	Directories for Oracle-related software.

\$DOMAIN_HOME directory structure

The \$DOMAIN_HOME directory is the root directory of Studio and your WebLogic domain. The default path to this directory is:

```
$ORACLE_HOME/user_projects/domains/bdd_domain
```

The following table describes the structure of \$DOMAIN_HOME immediately after deployment.

Directory name	Description
.	Contains the edit.lnk file.
/autodeploy	Provides a way to quickly deploy applications to a development server. You can place J2EE applications in this directory; these will be automatically deployed to the WebLogic Server when it is started in development mode.

Directory name	Description
/bin	Scripts for migrating servers and services, setting up domain and startup environments, and starting and stopping the WebLogic Server and other components.
/config	Data sources and configuration files for Studio and the Dgraph Gateway.
/console-ext	Console extensions. This directory is only used on the Admin Server.
/EndecaServer	Contains files and libraries used by the Dgraph Gateway, as well as the Dgraph Gateway Command Utility (<code>endeca-cmd</code>).
<code>fileRealm.properties</code>	Configuration file for the file realm.
/init-info	Schemas used by the Dgraph Gateway.
/lib	The domain library. The JAR files in this directory are dynamically added to the end of the Dgraph Gateway's classpath when the Dgraph Gateway is started. You use this directory to add application libraries to the Dgraph Gateway's classpath.
/nodemanager	Files used by the Node Manager. <code>nodemanager.domains</code> lists the locations of directories created by the configuration wizard, and <code>nodemanager.properties</code> configures the Node Manager.
/pending	
/security	
/servers	Log files and security information for each server in the cluster.
<code>shutdown-AdminServer.py</code> , <code>shutdown-<hostname>.py</code>	Python scripts for shutting down individual servers.
<code>startWebLogic.sh</code>	Script for starting the WebLogic Server.
/tmp	Temporary directory.

Verifying your deployment

Once the orchestration script completes, you can verify that each of the major BDD components were installed properly and are running.

[Verifying the deployed services](#)

[Verifying Data Processing](#)

Verifying the deployed services

Use the `bdd-admin` script to verify the statuses of the Dgraph, HDFS Agent, Studio, and Dgraph Gateway services.

More information on the `bdd-admin` script is available in the *Oracle Big Data Discovery Administrator's Guide*.



Note: You must perform this task from the Admin Server.

To verify the deployed services:

1. On the Admin Server, open a command prompt and navigate to the `$BDD_HOME/BDD_manager/bin` directory.
2. Execute the following command:

```
./bdd-admin.sh status --all
```

This command returns a message indicating the current status (running, unresponsive, or stopped) of each service in the cluster, along with its hostname and PID number.

Verifying Data Processing

To verify that Data Processing is running, you must launch a Data Processing workflow. You can do this in two ways:

- Use the CLI to launch a Data Processing workflow. For more information, please refer to the *Oracle Big Data Discovery Data Processing Guide*.
- Create a new Hive table in Studio. For more information, please refer to the *Oracle Big Data Discovery Data Exploration and Analysis Guide*.



Note: If you use the CLI to verify Data Processing, you must first add the table(s) you want processed to the CLI whitelist. For more information, see [Updating the CLI whitelist and blacklist](#).

Updating the CLI whitelist and blacklist

In order to create data sets from existing Hive tables, you must update the CLI white- and blacklists that define which tables are processed by Data Processing.

The CLI whitelist specifies which Hive tables should be processed: tables not included in this list are ignored by the Hive Table Detector and any Data Processing workflows invoked by the CLI. Similarly, the blacklist specifies the Hive tables that should not be processed. You can use one or both of these lists to control which of your Hive tables are processed and which are not.

Once you have updated the whitelist and/or blacklist as needed, you can either wait for the Hive Table Detector to process your tables automatically or use the CLI to start a Data Processing workflow immediately.

For information on the CLI white- and blacklists, see the *Data Processing Guide*.

Signing in to Studio as an administrator

After you complete the BDD installation and deployment, you can sign in to Studio as an administrator, begin to create new users, explore data sets, re-configure Studio settings as necessary, and so on.

To sign in to Studio as an administrator:

1. Ensure the WebLogic Server on the Admin Server node is running.
(This is the WebLogic instance running Studio.)
2. Open a Web browser and load Studio.
By default, the URL is `http://<Admin Server Name>:7003/bdd`.
3. Specify the default login and password and click **Sign In**.

Table 5.1: Sign in Values

Field	Value
Login	admin@oracle.com
Password	Welcome123

You are immediately prompted to change the password. The new password must contain:

- At least 6 characters
- At least one non-alphabetic character

Now you can add additional Studio users. There are several ways to add new Studio Users:

- Integrate Studio with an Oracle Single Sign On (SSO) system. For details, see the *Administrator's Guide*.
- Integrate Studio with an LDAP system. For details, see the *Administrator's Guide*.
- Or, while you are signed in as an administrator, you can create users manually in Studio from the **Control Panel>Users** page.

Backing up BDD

Oracle recommends that you back up BDD after deployment. This procedure must be performed manually.

For information on BDD backups, see the *Administrator's Guide*.

Replacing certificates

Enabling SSL for Studio activates WebLogic Server's default Demo Identity and Demo Trust Keystores. As their names suggest, these keystores are untrusted and meant for demo purposes only. After deployment, you should replace them with your own certificates.

More information on WebLogic's demo keystores is available in section [Configure keystores](#) of WebLogic's *Administration Console Online Help*.

Increasing Linux file descriptors

You should increase the number of file descriptors from the 1024 default.

Having a higher number of file descriptors ensures that the WebLogic Server can open sockets under high load and not abort requests coming in from clients.

To increase the number of file descriptors on Linux:

1. Edit the `/etc/security/limits.conf` file.
2. Modify the **nofile** limit so that **soft** is 4096 and **hard** is 8192. Either edit existing lines or add these two lines to the file:

```
*      soft      nofile      4096
*      hard      nofile      8192
```

The "*" character is a wildcard that identifies all users.

Customizing the WebLogic JVM heap size

You can change the default JVM heap size to fit the needs of your deployment.

The default JVM heap size for WebLogic is 3GB. The size is set in the `setDomainEnv.sh` file, which is in the `$DOMAIN_HOME/bin` directory. The heap size is set with the `-Xmx` option.

To change the WebLogic JVM heap size:

1. Open the `setDomainEnv` file in a text editor.
2. Search for this comment line:

```
# IF USER_MEM_ARGS the environment variable is set, use it to override ALL MEM_ARGS values
```

3. Add the following line immediately after the comment line:

```
export USER_MEM_ARGS="-Xms128m -Xmx3072m ${MEM_DEV_ARGS} ${MEM_MAX_PERM_SIZE}"
```

4. Save and close the file.
5. Re-start WebLogic Server.



Chapter 6

Creating Multiple Studio Instances

For a larger production environment, you may want to configure a number of Studio instances.

[About multiple Studio instances](#)

[Setting up multiple Studio instances](#)

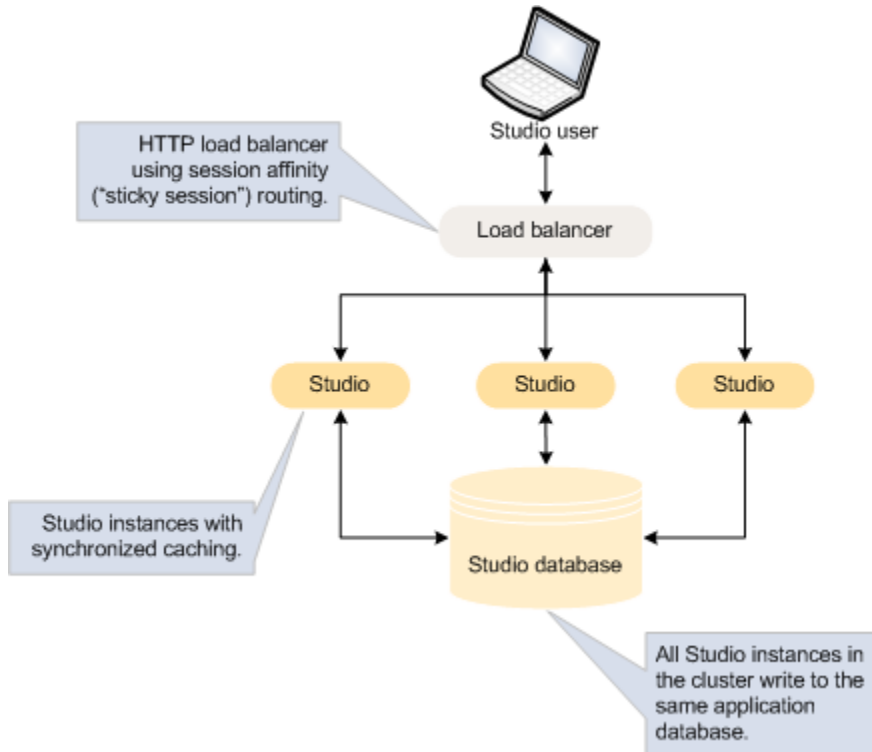
About multiple Studio instances

Studio allows you to create multiple Studio instances. In a cluster of Studio instances, changes made to one instance are automatically made to the other instances. For a large production environment, using clustering provides redundancy and support for higher throughput, allowing for more concurrent users.

A Studio cluster is made up of Studio instances configured to write to the same application database. For a clustered implementation, you cannot use a Hypersonic database.

The Studio instances also must be configured to use synchronized caching, so that information cached on one instance is available to all of the other instances in the cluster. Studio uses Ehcache (www.ehcache.org), which uses RMI (Remote Method Invocation) multicast to notify each member of the cluster when the cache has been updated.

In a Studio cluster, requests are routed to the Studio instances by an HTTP load balancer. The load balancer must use session affinity (also known as "sticky session") load balancing. If a member of the Studio cluster is down, the load balancer routes requests to another instance in the cluster.



Setting up multiple Studio instances

To configure multiple Studio instances, you connect each instance to the same application database, and then configure a shared cache for those instances.

[Installing the Studio instances](#)

[Configuring synchronized caching for the Studio instances](#)

Installing the Studio instances

Each instance in the cluster of Studio nodes is first installed as a standalone instance and then modified to share certain configuration.

Connecting each instance to the same Studio database

Each instance in the Studio cluster must be connected to the same Studio application database.

Optionally, you could use a clustered database configuration. For clustering, Oracle 11g uses RAC and MySQL has MySQL Cluster. For details on setting up a clustered database configuration, see the documentation for your database system.

Using the same configuration for each instance

In a clustered configuration, each instance should have the same configuration, to ensure that users have the same experience no matter which instance in the cluster they are connected to.

Most of the application settings are stored in the database. Because each instance writes to the same database, those settings remain constant among the cluster instances.

Also make sure that each instance has the same settings in `portal-ext.properties`. This includes any framework settings that you set in the file instead of from the **Control Panel** user interface.

Configuring synchronized caching for the Studio instances

Studio instances in a multiple node environment must use synchronized caching.

About synchronized caching

Synchronized caching ensures that the information cached by one Studio instance is available to all of the Studio instances in the environment.

This reduces the number of times each instance needs to query the Studio database, which allows for faster response times and better performance. Studio uses Ehcache (www.ehcache.org) for caching synchronization.

Updating `portal-ext.properties` to synchronize caching for Studio instances

The `portal-ext.properties` file for each instance includes commented-out settings for synchronizing the caches for Studio instances.

For each Studio instance, uncomment the following clustering settings in `portal-ext.properties`. You should be able to use the default values provided.

```
##
## Cluster
##
# Uncomment the following properties to enable clustering
# Note: Clustering will not work with Hypersonic.  Configure a common database for all cluster nodes.

#net.sf.ehcache.configurationResourceName=/ehcache/hibernate-clustered.xml
#ehcache.multi.vm.config.location=/ehcache/liferay-multi-vm-clustered.xml
#org.quartz.jobStore.isClustered=true
```

This table lists settings in `portal-ext.properties` used to enable synchronized caching of Studio instances (in a cluster of Studio instances). For each setting, the table provides a description of the required value.

Setting	Description
<code>net.sf.ehcache.configurationResourceName</code>	<p>The name and location of the XML configuration file for Hibernate caching. Hibernate is used by Studio to read from and write to the Studio application database.</p> <p>In the default <code>portal.properties</code> file, the configuration file is set to <code>hibernate.xml</code>, to implement caching in a non-clustered Studio implementation.</p> <p>When you uncomment this property in <code>portal-ext.properties</code>, which changes the configuration file to <code>hibernate-clustered.xml</code>, then Hibernate synchronizes the cache with the other Studio instances in the Studio cluster.</p>
<code>ehcache.multi.vm.config.location</code>	<p>The name and location of the XML configuration file for Ehcache.</p> <p>In the default <code>portal.properties</code> file, the file is set to <code>liferay-multi-vm.xml</code>, to implement caching in a non-clustered Studio implementation.</p> <p>When you uncomment this property in <code>portal-ext.properties</code>, which changes the configuration file to <code>liferay-multi-vm-clustered.xml</code>, then the cache is synchronized with the other Studio instances in the Studio cluster.</p>
<code>org.quartz.jobStore.isClustered</code>	<p>Enables clustering of Studio instances on the built-in Quartz job scheduling engine.</p>

These configuration files are configured to automatically detect the other Studio instances in the Studio cluster, and to use IP address 233.0.0.1 and port 4446 to send the updated cache information.

Customizing the shared cache configuration files

The default versions of the shared cache configuration files should work in most cases. However, you can if needed create and deploy customized versions.

The most likely customization that might be needed would be to the IP address and port number configured near the top of each file:

```
<cacheManagerPeerProviderFactory
  class="net.sf.ehcache.distribution.RMICacheManagerPeerProviderFactory"
  properties="peerDiscovery=automatic,multicastGroupAddress=230.0.0.1,multicastGroupPort
=4446,timeToLive=1"
  propertySeparator=","
/>
```

If you make any changes to these configuration files, make sure to make the same changes for all of the instances in the cluster.

To customize the clustered cache configuration files:

1. Extract the default files from the ehcache directory in `portal-impl.jar`.
The file is in the `WEB-INF/lib` directory, which is located in `endeca-portal.war`, which is in `bdd-studio.ear`.
2. Make the necessary updates to the files.
To ensure that Studio uses the correct files, you may want to rename the customized files to something like:
 - `hibernate-clustered-custom.xml`
 - `liferay-multi-vm-clustered-custom.xml`
3. To deploy the customized files:
 - (a) Undeploy `bdd-studio.ear`.
Use the appropriate method to undeploy the file based on whether you auto-deployed the `.ear` file or installed it.
 - (b) Update `bdd-studio.ear` to add a subdirectory `APP-INF/classes/ehcache/` that contains the customized XML files.
 - (c) Redeploy the updated `.ear` file.
4. If needed, update `portal-ext.properties` to reflect the customized file names:

```
net.sf.ehcache.configurationResourceName=/ehcache/hibernate-clustered-custom.xml
ehcache.multi.vm.config.location=/ehcache/liferay-multi-vm-clustered-custom.xml
```

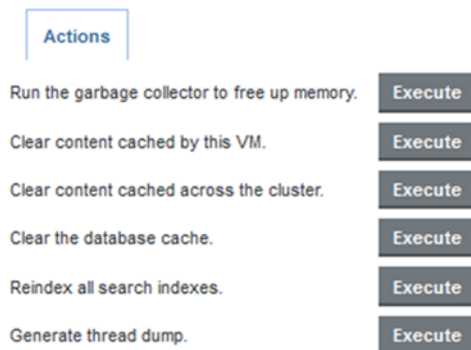
Clearing the cache for multiple Studio instances

As part of troubleshooting issues with a multi-instance Studio implementation, you can clear the cache for Studio instances. From the Studio **Control Panel**, you can clear the cache for either the current instance or for the entire Studio cluster.

To clear the Studio cache:

1. Click the **Control Panel** icon.
2. On the **Control Panel** menu, in the **Server** section, click **Server Administration**.

3. At the bottom of the page, on the **Actions** tab:



- To clear the cache for the current instance only, click the **Execute** button next to **Clear content cached by this VM**.
- To clear the cache for the entire Studio cluster, click the **Execute** button next to **Clear content cached across the cluster**.



Chapter 7

Using Studio with a Reverse Proxy

Studio can be configured to use a reverse proxy.

[About reverse proxies](#)

[Example sequence for a reverse proxy request](#)

[Recommendations for reverse proxy configuration](#)

[Reverse proxy configuration options for Studio](#)

About reverse proxies

A reverse proxy provides a more secure way for users to get access to application servers.

[What is a reverse proxy?](#)

[Types of reverse proxies](#)

What is a reverse proxy?

A reverse proxy retrieves resources on behalf of a client from one or more servers, and then returns these resources to the client as though they came from the server itself.

A reverse proxy is located between the client and the proxied server(s). Clients access content through the proxy server. The reverse proxy server assumes the public hostname of the proxied server. The hostname(s) of the actual/proxied servers are often internal and unknown to the client browser.

Some common reasons for implementing a reverse proxy include:

- Security or firewalling
- SSL termination
- Load balancing and failover
- Resource caching/acceleration
- URL partitioning

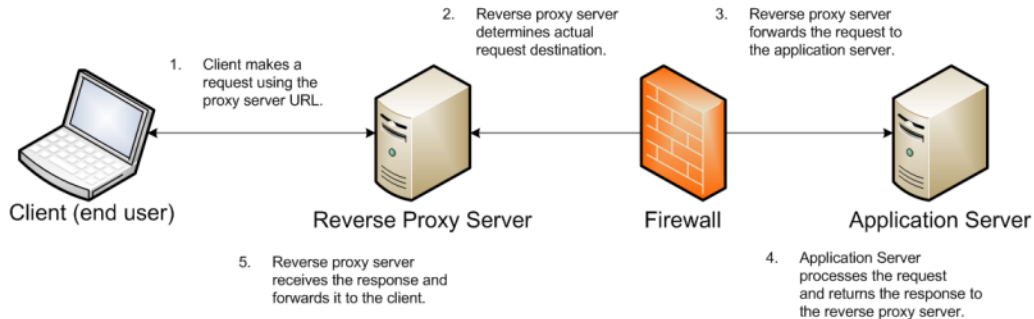
Types of reverse proxies

Reverse proxies may be either devices/appliances or specially configured web servers.

A very popular software-based reverse proxy is the Apache HTTP Server configured with the `mod_proxy` module. Many commercial web servers and reverse proxy solutions are built on top of Apache HTTP Server, including Oracle HTTP Server.

Example sequence for a reverse proxy request

Here is an example of the typical sequence for a request processed using a reverse proxy server.



1. The client makes a request to the public URL.

For this example, for a Studio project, the request URL might be something like `http://mybdd/bdd/web/myproject`, using the default port 80.

The hostname resolves to the address of the reverse proxy server. The reverse proxy is listening on this address and receives the request.

2. The reverse proxy server analyzes the URL to determine where the request needs to be proxied to.

A reverse proxy might use any part of the URL to route the request, such as the protocol, host, port, path, or query-string. Typically the path is the main data used for routing.

The reverse proxy configuration rules determine the outbound URL to send the request to. This destination is usually the end server responsible for serving the content. The reverse proxy server may also rewrite parts of the request. For example, it may change or make additions to path segments.

Reverse proxies can also add standard or custom headers to the request.

For example, the URL `http://mybdd/web/myproject` might be proxied to `http://bddserver1:8080/bdd/web/myproject`. In this case:

- The hostname of the target server is `bddserver1`
- The port is changed to 8080
- The context path `/bdd/` is added

3. The reverse proxy server sends the request to the target server.
4. The target server sends the response to the reverse proxy server.
5. The reverse proxy server reads the request and returns it to the client.

Recommendations for reverse proxy configuration

Here are some general configuration recommendations for setting up a reverse proxy.

[Preserving HTTP 1.1 Host: headers](#)

[Enabling the Apache ProxyPreserveHost directive](#)

Preserving HTTP 1.1 Host: headers

HTTP 1.1 requests often include a `Host:` header, which contains the hostname from the client request. This is because a server may use a single IP address or interface to accept requests for multiple DNS hostnames.

The `Host:` header identifies the server requested by the client. When a reverse proxy proxies an HTTP 1.1 request between a client and a target server, when it makes the request, it must add the `Host:` header to the outbound request. The `Host:` header it sends to the target server should be the same as the `Host:` header it received from the client. It should not be the `Host:` header that would be sent if accessing the target server directly.

When the application server needs to create an absolute, fully-qualified URL, such as for a redirect URL or an absolute path to an image or CSS file, it must provide the correct hostname to the client to use in a subsequent request.

For example, a Java application server sends a client-side redirect to a browser (HTTP 302 Moved). It uses the `ServletRequest.getServerName()` method to fetch the hostname in the request, then constructs a `Host:` header.

The URL sent by the client is `http://mystudio/web/myapp`. The actual internal target URL generated by the reverse proxy will be `http://studioserver1:8080/bdd/web/myapp`.

If there is no specific configuration for the target server, then if the reverse proxy retains the `Host:` header, the header is:

```
Host: http://mystudio
```

If the reverse proxy does not retain the `Host:` header, the result is:

```
Host: http://studioserver1:8080
```

In the latter case, where the header uses the actual target server hostname, the client may not have access to `studioserver1`, or may not be able to resolve the hostname. It also will bypass the reverse proxy on the next request, which may cause security issues.

If the `Host:` header cannot be relied on as correct for the client, then it must be configured specifically for the web or application server, so that it can render correct absolute URLs.

Most reverse proxy solutions should have a configuration option to allow the `Host:` header to be preserved.

Enabling the Apache ProxyPreserveHost directive

The `ProxyPreserveHost` directive is used to instruct Apache `mod_proxy`, when acting as a reverse proxy, to preserve and retain the original `Host:` header from the client browser when constructing the proxied request to send to the target server.

The default setting for this configuration directive is `Off`, indicating to not preserve the `Host:` header and instead generate a `Host:` header based on the target server's hostname.

Because this is often not what is wanted, you should add the `ProxyPreserveHost On` directive to the Apache HTTPD configuration, either in `httpd.conf` or related/equivalent configuration files.

Reverse proxy configuration options for Studio

Here are some options for configuring reverse proxy for Studio.

[Simple Studio reverse proxy configuration](#)

[Studio reverse proxy configuration without preserving Host: headers](#)

[Configuring Studio to support an SSL-enabled reverse-proxy](#)

Simple Studio reverse proxy configuration

Here is a brief overview of a simple reverse proxy configuration for Studio. The configuration preserves the `Host:` header, and does not use SSL or path remapping. Studio only supports matching context paths.

In this simple configuration:

- A reverse proxy server is in front of a single Studio application server.
- The reverse proxy server is configured to preserve the `Host:` header.
- The context paths match.
- Neither the reverse proxy nor the application server is configured for SSL.

With this setup, Studio should be able to be accessed correctly using the reverse proxy without additional configuration.

Studio reverse proxy configuration without preserving Host: headers

If a reverse proxy used by Studio does not preserve the `Host:` header, and instead makes a request with a `Host:` header referring to the target application server, Studio and the application server receive an incorrect hostname. This causes Studio to generate absolute URLs that refer to the proxied application server instead of to the reverse proxy server.

If the reverse proxy cannot be configured to preserve the `Host:` header, you must configure a fixed hostname and port. To do this, you can either:

- Configure the application server to have a fixed hostname and port
- Use `portal-ext.properties` to configure Studio with a fixed hostname and port

Configuring a fixed hostname for the application server

In WebLogic, set up a virtual host with the fixed hostname and port.

Configuring Studio with a fixed hostname

To configure Studio with a fixed hostname and port, add the following properties to `portal-ext.properties`:

```
web.server.host=<reverseProxyHostName>
web.server.http.port=<reverseProxyPort>
```

Configuring Studio to support an SSL-enabled reverse-proxy

If Studio is installed behind a reverse proxy that has SSL capabilities, and the client SSL is terminated on the reverse proxy, you must configure Studio to set the preferred protocol to HTTPS, and provide the host and port for the reverse proxy server.

To do this, add the following settings to `portal-ext.properties`:

```
web.server.protocol=https
web.server.host=<reverseProxyHostName>
web.server.https.port=<reverseProxyPort>
```

Where:

- *reverseProxyHostName* is the host name of the reverse proxy server.
- *reverseProxyPort* is the port number for the reverse proxy server.

Part IV

Uninstalling Big Data Discovery



Chapter 8

Uninstalling Big Data Discovery

The following sections describe how to uninstall BDD using the uninstallation script.

The uninstallation script

Running the uninstallation script

The uninstallation script

You can uninstall BDD by running the `uninstall.sh` script. This script deletes all BDD components, including all services, the Data Processing libraries, and the contents of the `$ORACLE_HOME` directory.

The uninstallation script is located in the installation source directory on the Admin Server. Like the orchestration script, it must be run from this location. The script must also have access to the `bdd.conf` file, which it reads at runtime to determine which components it needs to remove from each node.

Note that the uninstallation script does not delete everything the orchestration script created on your system. Specifically:

- Although the script deletes the contents of the BDD-specific directories from your system, it does not delete the directories themselves. For example, it removes everything inside of `$ORACLE_HOME`, but leaves the empty directory on your system. If you do not want these on your system, you can delete them manually when the script finishes running.



Note: You do not need to delete the empty directories if you plan on reinstalling BDD.

- The script does not delete the Dgraph index from the shared NFS, even if you're using the base index created by the orchestration script. If you plan on reinstalling BDD, you can leave the index on the NFS and reuse it.

Uninstallation script behavior

When the `uninstall.sh` script runs, it does the following:

1. Reads `bdd.conf`.
2. Terminates all currently running processes.
3. Deletes the WebLogic domain.
4. Cleans up the Hive Table Detector cron job.
5. Deletes the Data Processing CLI.
6. Deletes all Data Processing libraries from the Spark servers.
7. Deletes the `/userx/bdd` directory from HDFS.

8. Deletes the contents of the `$ORACLE_HOME` directory, including WebLogic Server and all BDD components, and the WebLogic domain.
9. Deletes the znode for the Dgraph cluster from the ZooKeeper namespace.

Running the uninstallation script

You can uninstall BDD by running the `uninstall.sh` from the Admin Server.

Before you run the uninstallation script, you should verify that `bdd.conf` is available in the installation source directory.

To run the uninstallation script:

1. On the Admin Server, navigate to the installation source directory.
2. Run the uninstallation script:

```
./uninstall.sh [bdd.conf]
```

The script reads `bdd.conf` at runtime by default, so this argument is optional.

3. Enter `yes` or `y` when asked if you're sure you want to uninstall BDD.

When the script finishes running, all BDD components are removed from your system.

Index

A

- administration
 - bdd-admin script 16
 - Enterprise Manager Plug-in 16
- Admin Server
 - about 11
 - selecting 31

B

- Big Data Discovery
 - about 9
 - administration 16
 - backup 54
 - configuration options 11
 - integration with CDH 10
 - integration with WebLogic 11
 - security 16
 - uninstalling 68

C

- CLI whitelist and blacklist, updating 53
- Cloudera Distribution for Hadoop, about 10
- clustering, Studio
 - about 56
 - enabling synchronized caching 58
 - installing instances 57
- Command Line Interface, about 10

D

- database, creating 24
- Data Processing, about 9
- Data Processing CLI, about 10
- Dgraph, about 10
- Dgraph Gateway, about 9
- Dgraph HDFS Agent, about 10
- directory structure
 - \$BDD_HOME 49
 - \$DOMAIN_HOME 51

E

- Endeca Server 16

F

- file descriptors, increasing 55

H

- Hadoop Exporter 16

- Hive Table Detector, about 10

I

- installation and deployment
 - about 27
 - configuring 34
 - creating the installation source directory 33
 - downloading the media pack 32
 - installation 46
 - rerunning the orchestration script 47
 - selecting the Admin Server 31
 - silent installation, about 27
 - troubleshooting 47
 - workflow 30

J

- JVM heap size, setting 55

M

- multi-node, Studio
 - clearing the cache 60
 - customizing the cache configuration 59

O

- orchestration script
 - about 27
 - configuring 34
 - rerunning 47
 - running 46
 - silent installation, about 27
 - troubleshooting 47

P

- prerequisites
 - CDH components 17
 - database 23
 - database commands 24
 - hardware 18
 - index 24
 - physical memory and disk space 19
 - software 21
 - supported browsers 25
 - user access 19

R

- reverse proxy, using with Studio 62

S

- security
 - about 16

- replacing certificates 54
 - reverse proxy 62
 - Studio
 - about 9
 - clearing the cache 60
 - clustering, about 56
 - clustering, about synchronized caching 58
 - clustering, enabling synchronized caching 58
 - clustering, installing instances 57
 - customizing the shared cache configuration 59
 - signing in 54
 - system requirements
 - CDH components 17
 - database 23
 - database commands 24
 - hardware 18
 - index 24
 - Linux utilities 22
 - physical memory and disk space 19
 - software 21
 - supported browsers 25
 - user access 19
- U**
- uninstallation
 - about 68
 - running the uninstallation script 69
- V**
- verification
 - Data Processing 53
 - deployed services 53
- W**
- WebLogic Server
 - about 11
 - setting JVM heap size 55