Oracle® Cloud

Using Oracle Big Data Cloud Service

Release 19.3.3
E62152-45
September 2019

Oracle Cloud Using Oracle Big Data Cloud Service, Release 19.3.3

E62152-45

Contents

9 Connect to Oracle Database with Oracle Big Data Connectors

10 Use Oracle Big Data SQL Cloud Service with Oracle Big Data Cloud Service

Preface

Using Oracle Big Data Cloud Service describes how to use Oracle Big Data Cloud Service to store and manage large amounts of data of various data types in the cloud, by using Hadoop HDFS file system and associated services and tools.

Topics

Audience

Using Oracle Big Data Cloud Service is intended for administrators and users who want to provision a Hadoop cluster in the cloud and use it to manage big data.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more information, see these Oracle resources:

- About Oracle Cloud in *Getting Started with Oracle Cloud*.
- Getting Started with Oracle Storage Cloud Service in *Using Oracle Cloud Infrastructure Object Storage Classic*

Conventions

The following text conventions are used in this document:

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

1

Get Started with Oracle Big Data Cloud Service

This section describes how to get started with Oracle Big Data Cloud Service.

Topics

About Oracle Big Data Cloud Service

An entitlement to Oracle Big Data Cloud Service gives you access to the resources of a preconfigured Oracle Big Data environment, including a complete installation of the Cloudera Distribution Including Apache Hadoop (CDH) and Apache Spark. Use Oracle Big Data Cloud Service to capture and analyze the massive volumes of data generated by social media feeds, e-mail, web logs, photographs, smart meters, sensors, and similar devices.

> **Note:**
>
> Oracle Big Data Cloud Service is offered on Oracle Cloud, using state-of-the-art Oracle-managed data centers. You can also choose Oracle Big Data Cloud at Customer, which provides Oracle Big Data Cloud Service hosted in your data center.

When you set up your Oracle Big Data Cloud Service, you can create a cluster of 3 to 60 nodes, consisting of Oracle Compute Units (OCPU), memory, and storage. All clusters must start with the 3–node starter pack and can have up to 57 additional permanent nodes. Those nodes can be added when creating the cluster or later. You can also temporarily extend the processing power (OCPUs) and memory of the cluster by adding cluster compute nodes (“bursting”). Oracle manages the whole hardware and networking infrastructure as well as the initial setup, while you have complete administrator’s control of the software.

All nodes in an Oracle Big Data Cloud Service instance form a cluster.

Software

An Oracle Big Data Cloud Service entitlement includes the following:

- Oracle Linux operating system
- Cloudera Distribution Including Apache Hadoop (CDH)

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

CDH includes:

- File system: The Hadoop Distributed File System (HDFS) is a highly scalable file system that stores large files across multiple servers. It achieves reliability by replicating data across multiple servers without RAID technology. It runs on top of the Linux file system.
- MapReduce engine: The MapReduce engine provides a platform for the massively parallel execution of algorithms written in Java. Oracle Big Data Cloud Service runs YARN by default.
- Administrative framework: Cloudera Manager is a comprehensive administrative tool for CDH.
- Apache projects: CDH includes Apache projects for MapReduce and HDFS, such as Hive, Pig, Oozie, ZooKeeper, HBase, Sqoop, and Spark.
- Cloudera applications: Oracle Big Data Cloud Service includes all products included in Cloudera Enterprise Data Hub Edition, including Impala, Search, and Navigator.

Several CDH utilities and other software available on Oracle Big Data Cloud Service provide graphical, web-based, and other language interfaces for ease of use.

- Built-in utilities for managing data and resources.
- Oracle Big Data Connectors, which facilitate access to data stored in an Apache Hadoop cluster.

  Included are:

  - Oracle SQL Connector for Hadoop Distributed File System
  - Oracle Loader for Hadoop
  - Oracle XQuery for Hadoop
  - Oracle R Advanced Analytics for Hadoop
  - Oracle Data Integrator Enterprise Edition
- Oracle Big Data Spatial and Graph, which provides advanced spatial and graph analytic capabilities to supported Apache Hadoop and NoSQL Database Big Data platforms.

Oracle Big Data SQL Cloud Service (Optional)

You can optionally integrate Oracle Big Data SQL Cloud Service. As a prerequisite, you must have an entitlement for the Oracle Big Data SQL Cloud Service add-on to Oracle Big Data Cloud Service and an entitlement for Oracle Database Exadata Cloud Service. For more information, contact an Oracle Sales Representative.

Oracle Cloud Infrastructure Object Storage Classic Integration (Optional)

If you have an entitlement for Oracle Cloud Infrastructure Object Storage Classic, you can integrate the storage with Oracle Big Data Cloud Service. For information about

the storage service, see Oracle Cloud Infrastructure Object Storage Classic Get Started.

Before You Begin with Oracle Big Data Cloud Service

Before you begin using Oracle Big Data Cloud Service, you should be familiar with the following:

- Oracle Cloud

  See Welcome to Oracle Cloud.

- Cloudera Distribution Including Apache Hadoop (CDH)

  CDH provides the software infrastructure for working with big data. See the Cloudera documentation for the Cloudera release supported by this release of Oracle Big Data Cloud Service. (See *What's New for Oracle Big Data Cloud Service* to find the release numbers.)

- Oracle Cloud Infrastructure Object Storage Classic (optional)

  See About Oracle Cloud Infrastructure Object Storage Classic.

Before you create your first Oracle Big Data Cloud Service cluster:

- Obtain an entitlement to Oracle Big Data Cloud Service. See How to Begin with Oracle Big Data Cloud Service.
- Optionally, create a Secure Shell (SSH) public/private key pair so you can provide the public key when you create the cluster. See Generating a Secure Shell (SSH) Public/Private Key Pair. Alternatively, you can create a new key pair from the wizard when creating the cluster.
- Optionally, have access to and credentials for an Oracle Cloud Infrastructure Object Storage Classic instance. See Oracle Cloud Infrastructure Object Storage Classic Get Started.
- Optionally, have an entitlement to Oracle Big Data SQL Cloud Service. (You must also have an entitlement to Oracle Database Cloud Exadata Service in order to use. Oracle Big Data SQL Cloud Service.)

How to Begin with Oracle Big Data Cloud Service

Here’s how to get started with Oracle Big Data Cloud Service .

1. Sign up for an Oracle Cloud account and purchase a subscription. See Welcome to Oracle Cloud and Buy an Oracle Cloud Subscription in *Getting Started with Oracle Cloud*.
2. Learn aboutOracle Big Data Cloud Service users and roles. See About Oracle Big Data Cloud Service Users and Roles.
3. Create accounts for your users and assign them appropriate privileges and roles.

   See Managing User Accounts and Managing User Roles in *Managing and Monitoring Oracle Cloud*.
4. Be sure to review the prerequisites described in Before You Begin with Oracle Big Data Cloud Service before you create your first Oracle Big Data Cloud Service cluster.

Generating a Secure Shell (SSH) Public/Private Key Pair

Several tools exist to generate SSH public/private key pairs. The following sections show how to generate an SSH key pair on UNIX, UNIX-like and Windows platforms.

Generating an SSH Key Pair on UNIX and UNIX-Like Platforms Using the ssh-keygen Utility

UNIX and UNIX-like platforms (including Solaris and Linux) include the ssh-keygen utility to generate SSH key pairs.

To generate an SSH key pair on UNIX and UNIX-like platforms using the ssh-keygen utility:

1. Navigate to your home directory:

   ```
   $ cd $HOME
   ```

2. Run the ssh-keygen utility, providing as ***filename*** your choice of file name for the private key:

   ```
   $ ssh-keygen -b 2048 -t rsa -f filename
   ```

   The ssh-keygen utility prompts you for a passphrase for the private key.

3. Enter a passphrase for the private key, or press **Enter** to create a private key without a passphrase:

   ```
   Enter passphrase (empty for no passphrase): passphrase
   ```

   > **Note:**
   >
   > While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

   The ssh-keygen utility prompts you to enter the passphrase again.

4. Enter the passphrase again, or press **Enter** again to continue creating a private key without a passphrase:

   ```
   Enter the same passphrase again: passphrase
   ```

5. The ssh-keygen utility displays a message indicating that the private key has been saved as *filename* and the public key has been saved as *filename*`.pub`. It also displays information about the key fingerprint and randomart image.

Generating an SSH Key Pair on Windows Using the PuTTYgen Program

The PuTTYgen program is part of PuTTY, an open source networking client for the Windows platform.

To generate an SSH key pair on Windows using the PuTTYgen program:

1. Download and install PuTTY or PuTTYgen.

To download PuTTY or PuTTYgen, go to http://www.putty.org/ and click the **You can download PuTTY here** link.

2. Run the PuTTYgen program.

   The PuTTY Key Generator window is displayed.

3. Set the **Type of key to generate** option to **SSH-2 RSA**.
4. In the **Number of bits in a generated key** box, enter **2048**.
5. Click Generate to generate a public/private key pair.

   As the key is being generated, move the mouse around the blank area as directed.

6. (Optional) Enter a passphrase for the private key in the **Key passphrase** box and reenter it in the **Confirm passphrase** box.

   > **Note:**
   >
   > While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use. When you specify a passphrase, a user must enter the passphrase every time the private key is used.

7. Click **Save private key** to save the private key to a file. To adhere to file-naming conventions, you should give the private key file an extension of `.ppk` (PuTTY private key).
8. Select all of the characters in the **Public key for pasting into OpenSSH authorized_keys file** box.

   Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.

9. Right click somewhere in the selected text and select **Copy** from the menu.
10. Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.
11. Save the text file in the same folder where you saved the private key, using the `.pub` extension to indicate that the file contains a public key.
12. If you or others are going to use an SSH client that requires the OpenSSH format for private keys (such as the `ssh` utility on Linux), export the private key:
    a. On the **Conversions** menu, choose **Export OpenSSH key** .
    b. Save the private key in OpenSSH format in the same folder where you saved the private key in `.ppk` format, using an extension such as `.openssh` to indicate the file's content.

About Oracle Big Data Cloud Service Users and Roles

Oracle Big Data Cloud Service supports the following service roles and operating system roles.

Cloud Service Users and Roles

In addition to the roles and privileges described in Oracle Cloud User Roles and Privileges in *Getting Started with Oracle Cloud*, the following roles are created for Oracle Big Data Cloud Service:

- Big Data Administrator

  A user assigned this role has complete administrative control over the service.

- Viewer

  A user assigned this role has read-only access to the service.

When the Oracle Big Data Cloud Service account is first set up, the service administrator is given the Big Data Administrator role. User accounts must be added and assigned one of the above roles before anyone else can access and use Oracle Big Data Cloud Service.

Only the identity domain administrator is allowed to create user accounts and assign roles. See Adding Users and Assigning Roles in *Managing and Monitoring Oracle Cloud*.

The predefined Oracle Big Data Cloud Service roles are associated with specific clusters. That is, if you have two clusters called **test123** and **production123**, four predefined roles are available to assign to users:

- test123 Big Data Administrator
- test123 Viewer
- production123 Big Data Administrator
- production123 Viewer

Users have access only to those clusters associated with the roles assigned them. For example, in the above case, a user might be assigned the role Big Data Administrator for test123 and the role Viewer for production123.

Big Data Manager Users and Roles.

By default, the `bigdatamgr` user is created and granted the administrator role. This user should be used to grant roles and register providers. The `bigdatamgr` user has the same password as the Cloudera Manager administrator that was defined in Create Cluster wizard when creating the cluster.

Operating System Users and Roles

Every Oracle Big Data Cloud Service cluster node is provisioned with the following operating system user accounts.

- **`opc`**

  The system administrator account you use in conjunction with the `sudo` command to gain `root` user access to your nodes. By default, `opc` doesn't allow connection using a password; however, you may choose to connect using a password by

assigning a known password to `opc` or by creating another user with a known password. See Managing User Accounts in *Oracle Big Data Appliance Software User's Guide*.

- **`root`**

  The root administrator for the system. You do not have direct access to this account. To perform operations that require root user access, execute `sudo -s` as the `opc` user. By default, `root` doesn't require a password.

- **`oracle`**

  An operating system and cluster user account that is used to run jobs on the cluster during the validation of the cluster. This account is used by the system and has a randomly generated password.

Enabling IPsec VPN Access to Oracle Big Data Cloud Service

Oracle Network Cloud Service — VPN for Engineered Systems is an add-on service available at an additional subscription fee. Using this service, you can create a secure virtual private network (VPN) tunnel over the Internet that connects your corporate network to Oracle Public Cloud services, such as Oracle Big Data Cloud Service. The service uses IPsec, which is a suite of protocols designed to authenticate and encrypt all IP traffic between two locations.

> **Note:**
>
> For information on IPSec standards, see the Internet Engineering Task Force (IETF) Request for Comments (RFC) 6071: *IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap*.

Before you request VPN, ensure these requirements are met at your site:

- **VPN device requirements.** You need a VPN gateway device that uses current IPSec standards to establish a secure tunnel between your network and the Oracle Public Cloud. You will provide the details of your device to Oracle. The device must support:
  - IPv4 traffic with support for ICMP, TCP and UDP. Multicast traffic is not supported.
  - Tunnel mode sessions: Tunnel mode is used to create a virtual private network between your network and the Oracle Public Cloud, rather than between a specific set of hosts. It is used to protect all communications between both networks.
  - Authentication with pre-shared keys. The same pre-shared key is configured on each IPSec VPN gateway device.
  - Dynamic rekeying: IPsec uses dynamic rekeying to control how often a new key is generated during communication. Communication is sent in blocks and each block of data is secured with a different key.
- **Network requirements for an IPSec VPN connection.** Both sides must provide subnets:
  - On your side, dedicate subnets in your network for this VPN connection. You will indicate these subnets to Oracle. You will give the necessary information

about these subnets to Oracle. To prevent an IP address conflict in the end-to-end network connection, mask your internal systems with a public or non-RFC 1918 address range.

  - On the Oracle side, the network engineers from the Oracle Cloud Operations will provide the destination subnets in a way that avoids IP address conflicts.

To request a VPN provisioning by Oracle Support:

1. Contact your sales representative and ask them to place an order for Oracle Network Cloud Service — VPN for Engineered Systems — Non-metered. This can be a separate order, or it can be made in conjunction with an order for Oracle Big Data Cloud Service.
2. Once you have an active subscription to Oracle Network Cloud Service — VPN for Engineered Systems, go to the My Oracle Support Note 2056914.1 and follow its instructions.

Oracle engineers will receive your information and check that all prerequisites are met. Next, during an agreed maintenance window, Oracle together with your network engineers will provision the VPN service and run through a post-configuration checklist to ensure that the VPN connection is working and that the setup is completed.

Where Do Services Run On a Cluster?

Services are installed on all nodes of an Oracle Big Data Cloud Service cluster, but individual services run only on designated nodes. A cluster can have from 3 to 60 permanent nodes, and the services are distributed differently, depending on the number of nodes:

- Services on a three-node cluster are distributed as shown in Where Do the Services Run on a Three-Node, Development-Only Cluster?.
- Services on a cluster of four or more nodes are distributed as shown in Where Do the Services Run on a Cluster of Four or More Nodes?.

Consider the following:

- If you have a three-node cluster and then expand it to four or more nodes, the services are automatically redistributed, as shown in Where Do the Services Run on a Cluster of Four or More Nodes?.
- Three- and four-node clusters are recommended for development purposes only.
- The minimum recommended size for a production cluster is five nodes.
- If you plan to install Oracle Big Data Discovery on Oracle Big Data Cloud Service, a minimum of five nodes is required.

Where Do the Services Run on a Three-Node, Development-Only Cluster?

The following table shows how services are distributed on the nodes of Oracle Big Data Cloud Service three-node clusters.

Service	Node1	Node2	Node3
Active Navigator Key Trustee Server (if secure cluster is enabled)	**Yes**	No	No
Big Data Manager	No	No	**Yes**
Cloudera Manager and Cloudera Manager roles	No	No	**Yes**
DataNode	**Yes**	**Yes**	**Yes**
Failover Controller	**Yes**	**Yes**	No
Hive	No	**Yes**	No
Hive Metastore	No	**Yes**	No
HttpFS	No	**Yes**	No
Hue	No	**Yes**	No
JobHistory	No	No	**Yes**
JournalNode	**Yes**	**Yes**	**Yes**
Kerberos Master KDC (if secure cluster is enabled.)	**Yes**	No	No
Kerberos Slave KDC (if secure cluster is enabled)	No	**Yes**	No
MySQL Backup	No	**Yes**	No
MySQL Primary	No	No	**Yes**
NameNode	**Yes**	**Yes**	No
NodeManager	**Yes**	**Yes**	**Yes**
ODI	No	**Yes**	No
Oozie	No	**Yes**	No
Passive Navigator Key Trustee Server (if secure cluster is enabled)	No	**Yes**	No
ResourceManager	**Yes**	No	**Yes**
Sentry	**Yes**	No	No
Spark History	No	No	**Yes**
WebHCat	No	**Yes**	No
ZooKeeper	**Yes**	**Yes**	**Yes**

Where Do the Services Run on a Cluster of Four or More Nodes?

The following table shows how services are distributed on the nodes of Oracle Big Data Cloud Service clusters with four, five, or more nodes.

> **Note:**
>
> The minimum recommended size for a production cluster is five nodes.

Service	Node1	Node2	Node3	Node4	Node5 to Node*nn*
Active Navigator Key Trustee Server (if secure cluster is enabled)	**Yes**	No	No	No	No
Balancer	**Yes**	No	No	No	No
Big Data Manager	No	No	**Yes**	No	No
Cloudera Manager Server	No	No	**Yes**	No	No
Cloudera Manager and Cloudera Manager roles	No	No	**Yes**	No	No
DataNode	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Failover Controller	**Yes**	**Yes**	No	No	No
Hive	No	No	No	**Yes**	No
Hive Metastore	No	**Yes**	No	No	No
HttpFS	No	**Yes**	No	No	No
Hue	No	No	No	**Yes**	No
JobHistory	No	No	**Yes**	No	No
JournalNode	**Yes**	**Yes**	**Yes**	No	No
Kerberos KDC (if secure cluster is enabled.)	**Yes**	**Yes**	No	No	No
MySQL Backup	No	**Yes**	No	No	No
MySQL Primary	No	No	**Yes**	No	No
NameNode	**Yes**	**Yes**	No	No	No
Navigator Audit Server and Navigator Metadata Server	No	No	**Yes**	No	No
NodeManager	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Oozie	No	No	No	**Yes**	No
Oracle Data Integrator Agent	No	No	No	**Yes**	No

Service	Node1	Node2	Node3	Node4	Node5 to Node*nn*
Passive Navigator Key Trustee Server (if secure cluster is enabled)	No	**Yes**	No	No	No
ResourceManager	No	No	**Yes**	**Yes**	No
Sentry Server (if enabled)	**Yes**	No	No	No	No
Solr	No	No	No	**Yes**	No
Spark History	No	No	**Yes**	No	No
ZooKeeper	**Yes**	**Yes**	**Yes**	No	No

> **Note:**
>
> If Oracle Big Data Discovery is installed, the NodeManager and DataNode on Node05 of the cluster are decomissioned.

Typical Workflow for Using Oracle Big Data Cloud Service

To start using Oracle Big Data Cloud Service, refer to the following tasks as a guide:

Task	Description	More Information
Purchase a subscription.	Purchase a subscription through the `cloud.oracle.com` website or by talking to an Oracle Sales Representative..	How to Begin with Oracle Big Data Cloud Service
Add and manage users and roles	Create accounts for your users and assign them appropriate privileges. Assign the necessary Oracle Big Data Cloud Service roles.	About Oracle Big Data Cloud Service Users and Roles
Create a service instance	Use the wizard to create a service instance, which allocates resources for a cluster.	Create an Oracle Big Data Cloud Service Instance
Create an SSH key pair	Create an SSH public/private key pair for use when creating or accessing clusters.	Generating a Secure Shell (SSH) Public/Private Key Pair
Create a cluster	Use a wizard to create a cluster, using the resources allocated to your service instance.	Create a Cluster
Control network access to services	Configure whitelists or the firewall to open or close the ports used by services such as Cloudera Manager and Hue.	Control Network Access to Services

Task	Description	More Information
Access and work with your cluster.	Access and work with your cluster using Secure Shell and by using graphical tools such as Oracle Big Data Manager, Cloudera Manager, and Hue.	Access Your Oracle Big Data Cloud Service
Add permanent nodes to a cluster	Add nodes in one-node increments up to a total of 60 nodes in the cluster.	Adding Permanent Nodes To a Cluster
Add temporary compute nodes to a cluster (bursting)	Temporarily add and remove cluster compute nodes to handle increased loads.	Adding and Removing Cluster Compute Nodes (Bursting)
Patch a cluster	Apply a patch or roll back a patch.	Using Mammoth to Install a One-Off Patch
Manage storage providers and copy data.	Register storage providers and users, and copy data between HDFS and storage.	Copy Data With Oracle Big Data Cloud Service Tools

2

Manage the Life Cycle of Oracle Big Data Cloud Service

This section describes tasks to manage the life cycle of your service.

Topics

Understand the Life Cycle of Oracle Big Data Cloud Service

For each Oracle Big Data Cloud Service cluster you want to create, you must have a separate subscription.

The process is as follows:

1. **Purchase a subscription.** Go to cloud.oracle.com or contact an Oracle Sales Representative to buy a subscription. See How to Begin with Oracle Big Data Cloud Service.

   Every cluster must have at least three permanent Hadoop nodes (a starter pack) and can have an additional 57 permanent nodes, which can be a combination of permanent Hadoop nodes and edge nodes. When you purchase your subscription, make sure to include enough resources to create a cluster or clusters with the number of nodes you want to have initially.

   After you've created a cluster, you can extend your subscription with additional permanent nodes, up to the maximum 60.

2. **If this is a new Oracle Cloud account, provide account details,** such as account name and administrator details.

3. **Create a service instance.** This step allocates the resources for a cluster. A cluster must have one and only one starter pack of 3 permanent nodes and can have up to 57 additional nodes (for a total of 60 permanent nodes), so you must create an instance with the number of nodes you initially want for your cluster. The minimum is the three –node starter pack, but at least five nodes are recommended for production environments. Creating an instance doesn't create the cluster itself. See Create an Oracle Big Data Cloud Service Instance.
4. **Create the cluster.** You can have only one cluster per service instance. See Create a Cluster.
5. After the cluster is created, you can **extend** it in two ways:
   - **Add permanent nodes**, at additional cost. These nodes, which can be *permanent Hadoop nodes* or *edge nodes*. These nodes include OCPUs, memory, and storage. See About Oracle Big Data Cloud Service Nodes and Adding Permanent Nodes To a Cluster.
6. If needed, you can delete the cluster and create a new one, using the resources specified in your subscription and allocated to the service instance. In this case, you don't have to create a new service instance before creating the cluster. Because the resources are already allocated, you can create the cluster directly.

The above steps are similar to the steps for other Oracle Cloud services, but each service has its own variations. For generic instructions on subscribing to a service and instantiating it, see Getting Started with Oracle Cloud.

> **Note:**
>
> If you've included the Oracle Big Data SQL Cloud Service add-on in your subscriptions, there are a few additional steps. See Use Oracle Big Data SQL Cloud Service with Oracle Big Data Cloud Service

About Oracle Big Data Cloud Service Nodes

Every cluster must have at least three permanent Hadoop nodes (a starter pack) and can have an additional 57 permanent nodes, which can be a combination of permanent Hadoop nodes and edge nodes. In addition, a cluster can have up to 15 cluster compute nodes (480 OCPUs).

Permanent Hadoop Nodes

Permanent Hadoop nodes last for the lifetime of the cluster. Each node has:

- 32 Oracle Compute Units (OCPUs)
- 248 GB of available RAM
- 48 TB storage
- Full use of the Cloudera Enterprise Data Hub Edition software stack, including licenses and support

When planning the number of nodes you want for a cluster, be aware of the following:

- Three-node clusters are recommended for development only. A production cluster should have five or more nodes. This is to ensure that, if a node fails, you can

migrate the node responsibilities to a different node and retain quorums in the high availability setup of the cluster.

- Services are distributed differently on three-node clusters than they are on clusters of four or more nodes. See Where Do the Services Run on a Three-Node, Development-Only Cluster?
- You must have at least four permanent Hadoop nodes before you can add edge nodes.
- Installing Oracle Big Data Discovery on a Oracle Big Data Cloud Service cluster requires at least five nodes.

Edge Nodes

Edge nodes provide an interface between the Hadoop cluster and the outside network. They are commonly used to run client applications and cluster administration tools, keeping them separate from the nodes of the cluster that run Hadoop services. Like permanent Hadoop nodes, edge nodes last for the lifetime of the cluster. They have the same characteristics as permanent Hadoop nodes:

- 32 Oracle Compute Units (OCPUs)
- 248 GB of available RAM
- 48 TB storage
- Full use of the Cloudera Enterprise Data Hub Edition software stack, including licenses and support

When you create a cluster or expand a cluster, you can specify how many of the nodes will be edge nodes, as long as the first four nodes are permanent Hadoop nodes.

Cluster Compute Nodes

Cluster compute nodes have only OCPUs and memory (no storage), and you can add and remove them at will, a process known as “bursting.” Bursting provides the elasticity of growing and shrinking the cluster as processing needs fluctuate.

Clusters can be extended by up to 15 cluster compute nodes. However, when you work with cluster compute nodes in the service console, you identify them by their number of OCPUs, The following are supported.

- 32 OCPUs = 1 node
- 64 OCPUs = 2 nodes
- 96 OCPUs = 3 nodes
- 128 OCPUs = 4 nodes
- 160 OCPUs = 5 nodes
- 192 OCPUs = 6 nodes
- 224 OCPUs = 7 nodes
- 256 OCPUs = 8 nodes
- 288 OCPUs = 9 nodes
- 320 OCPUs = 10 nodes
- 352 OCPUs = 11 nodes

- 384 OCPUs = 12 nodes
- 416 OCPUs = 13 nodes
- 448 OCPUs = 14 nodes
- 480 OCPUs = 15 nodes

Because cluster compute nodes don't include the Hadoop Distributed File System (HDFS), you don't store data on these nodes. Therefore, when you remove cluster compute nodes from the cluster, there is no impact on any data stored in the cluster.

Create an Oracle Big Data Cloud Service Instance

When you create an Oracle Big Data Cloud Service *instance*, you initiate a process that allocates resources for a cluster. You have to create a service instance before you can create a cluster.

To create an Oracle Big Data Cloud Service instance:

1. Sign in to Oracle Cloud and enter your account credentials.
2. From the navigation menu at the top left, select Big Data Cloud Service.
3. Click **Create Instance** to display the **Create Instance** dialog box.
4. In the **BigData** section of the **All Services** tab, click the **Create** button to launch the **Create New Oracle Big Data Cloud Service Instance** wizard.
5. On the **Instance Details** page of the wizard, provide the following details for the service instance and then click **Next**:

   Instance Details

   - **Name** —Enter a name for the service instance. You'll see this name later in the **Create Service** wizard, where it will appear on a list of instances available for creating a cluster.
   - **Region**—Select the region to host your Oracle Big Data Cloud Service instance. (Not available on Oracle Big Data Cloud at Customer)

     If you choose a region that supports Oracle Cloud Infrastructure,, the **Availability Domain** and **Subnet** fields are displayed, and the instance will be hosted in an availability domain in an Oracle Cloud Infrastructure environment. Otherwise, the deployment will be hosted in a region in an Oracle Cloud Infrastructure Classic environment.

     > **Note:**
     >
     > – A classic *region* is a single data center.
     >
     > – An *availability domain* is one of possibly multiple data centers in a region, where the region is a localized geographic area.
     >
     > See Regions and Availability Domains.
     >
     > In some environments, the **Region** option is not available, and a region is assigned automatically.

Choose **No Preference** to let Oracle Big Data Cloud Service choose an Oracle Cloud Infrastructure Classic region for you.

- **Availability Domain**—The availability domain (within the region) where the instance will be provisioned. (Available only on Oracle Cloud Infrastructure)
- **Subnet**—The subnet (within the availability domain) that will determine network access to the instance. (Available only on Oracle Cloud Infrastructure)
- **Plan**—There's currently only one plan for Oracle Big Data Cloud Service, and it's selected by default.
- **Starter Pack** – Every service instance must include one (and only one) starter pack of three nodes.

> **Note:**
>
> If you have a metered service, it may look like you can add additional starter packs, but you shouldn't try to do so. If you want additional nodes, add them as individual nodes.

- **Additional Nodes**

  For a metered subscription, this field shows the number of additional permanent nodes in your subscription; that is, in addition to the three-node starter pack. The field can't be edited because you can only enable an instance on the full entitlement, and it will always include all nodes present.

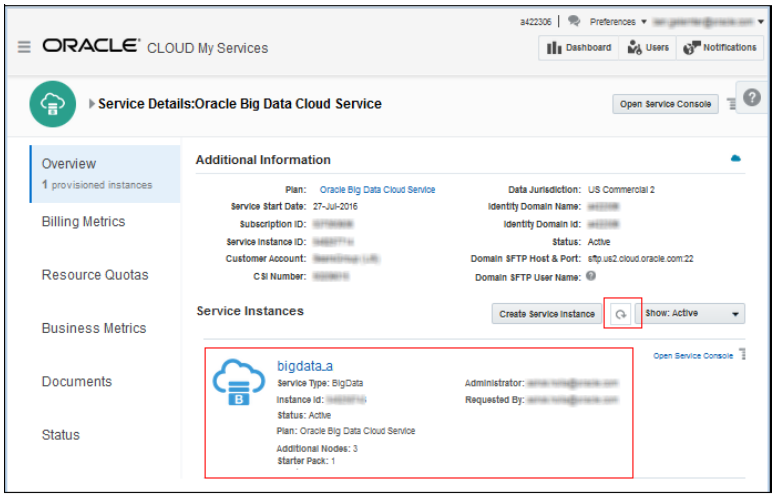
  For an entitlement purchased with Universal Credits,, enter the number of additional nodes you want. Make sure that you have enough credits to pay for the additional nodes.

Administrator Details

- **E-mail** —The e-mail address of the administrator for this service instance.
- **Use e-mail as user name** – Select this to use the administrator's e-mail address, above, as the user name for this service instance.
- **User Name**—Enter a user name for the user, if you didn't select **Use e-mail as user name**.
- **First/Last Name** —Enter the administrator's first and last name.

6. On the Confirmation page, if you're satisfied with the details, click **Create Service Instance** to initiate the allocation of resources.

   When the new service instance appears under **Service Instances** on the Service Details page, the instance's resources are available for creating a cluster.



You'll also receive an "Action Required" e-mail announcing that the service instance is ready and you can create a cluster using those resources.

7. When the instance is ready, you can proceed to create a cluster. See Create a Cluster.

Create a Cluster

You can create a single cluster from the resources allocated in an Oracle Big Data Cloud Service instance.

Before You Begin

Before you can create a cluster,

- You must have an Oracle Big Data Cloud Service instance that hasn't yet been used for a cluster. See Understand the Life Cycle of Oracle Big Data Cloud Service and Create an Oracle Big Data Cloud Service Instance.
- Optionally, you can create a secure shell (SSH) key pair. See Generating a Secure Shell (SSH) Public/Private Key Pair. Alternatively, you can create it when creating the cluster. See Create a Cluster.
- Optionally, you can use Oracle Cloud Infrastructure Object Storage Classic for storing data. To a associate an Oracle Cloud Infrastructure Object Storage Classic instance with the cluster, you'll need the credentials for accessing the storage. See Register Storage Credentials with the Cluster.

Procedure

To create an Oracle Big Data Cloud Service cluster:

1. On the **Services** page of your console, click the **Big Data Cloud Service** link at the top of the page.
2. Click the **Create Service** button to display the **Create Service** wizard.
3. On the **Services** page of the **Create Service** wizard, configure details for your cluster, as described below, and then click **Next**:
    - **Service Name**—Enter a name for the cluster that is unique in your identity domain.

- **Notification E-mail** —Enter an e-mail address to be used for notifications about this cluster.
- **Description**—Enter a description for the cluster.

4. On the **Cluster Details** page of the **Create Service** wizard, configure details for your cluster, as described below, and then click **Next**:

   Service Type Subscription Configuration

   - **Big Data Appliance System**—Select the resource configuration to use for this cluster. The items on this list show the Oracle Big Data Cloud Service instances that are available for creating clusters.
   - **Node Roles**—If you have five or more nodes available in your instance, you can specify how many are to be configured as permanent Hadoop nodes and how many as edge nodes. The first four must always be permanent Hadoop nodes. If you don't set these options, all the available nodes will be used as permanent Hadoop nodes.

   Cluster Parameters

   - **SSH Key**—The SSH public key is used for authenticating the `opc` user when using an SSH client to connect to a node that is associated with this cluster. Click **Edit** to open a dialog box for entering SSH key information. Use any of the following options and then click **Enter**.
     - **Key File Name**—Click the button to open a dialog box to select an existing SSH public key file.
     - **Key Value**—Paste in the value of an SSH public key. Make sure the value doesn't contain extra spaces or line breaks and doesn't have a line break at the end.
     - **Create a New Key**—Select this option to create a new key. You will be prompted to download the key after it's created.
   - **Cloudera Administrator Password** —Enter a string for the password for accessing Cloudera Manager and other Cloudera tools. Don't use a dollar sign ($) in your password. It may be accepted, but it can cause problems later on. *Be sure to record the password and store it securely.*
   - **Secure Setup** — Select **Enabled** to activate security services for the cluster or `Disabled` to leave them inactive. When you enable secure setup, you enable:
     - MIT Kerberos
     - Apache Sentry
     - HDFS Encryption—Key Trustee Servers are installed on the master nodes
     - Network Encryption
     - Auditing

> **Note:**
>
> If you don't enable security now, when creating the cluster, you cannot enable security later.
>
> When you create a cluster with security enabled, you can't disable any of these features for the lifetime of the cluster.

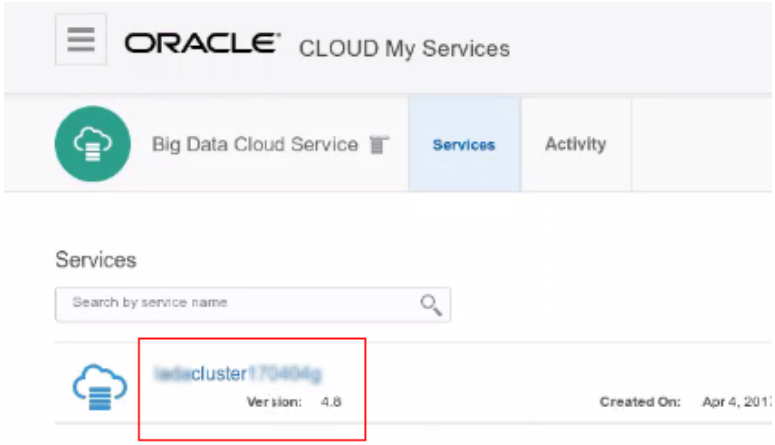
Oracle Cloud Infrastructure Object Storage Classic Configuration

- **Storage URL** — Enter the URL for a container in your storage service instance. This is the URL for a container in your storage service instance, for example, `https://storage-a123456.storage.oraclecloud.com/v1/storaage-a123456/myContainer4`. It is not the storage authentication URL. Storage service URLs are constructed in a variety of ways. See Accessing Oracle Cloud Infrastructure Object Storage Classic.
- **User Name** — Enter the user name that is associated with the storage service, for example the administrators e-mail (if that was specified as the user name for the service).
- **Password** — Enter the password associated with the storage user name.

> **Note:**
>
> When you specify storage details on this page, the storage provider name "BDCS" is created by default.

If you've entered incorrect credentials or if the storage service is unavailable, you will receive an error when you click **Next** to move to the next page in the wizard.

5. On the Confirm page of the **Create Service** wizard, check your configuration details. If you are satisfied, click **Create** to create the cluster.
6. When the cluster is ready, the new cluster is shown on your **Services** page. (You may have to refresh the page to see the status of the new cluster.)
7. To see details about your new cluster, click the name of the cluster.



8. Review the details on the **Service Overview** page.

Add Nodes to a Cluster

You can extend a cluster by adding *permanent Hadoop nodes*, *edge nodes*, and *cluster compute* nodes.

See About Oracle Big Data Cloud Service Nodes for information about the different kinds of nodes.

Topics

- Adding Permanent Nodes To a Cluster
- Adding and Removing Cluster Compute Nodes (Bursting)

Adding Permanent Nodes To a Cluster

You can add additional *permanent* nodes to a cluster after it was created and started. Permanent nodes include *permanent Hadoop nodes* and *edge nodes*.

To add permanent nodes to a cluster:

1. The process is slightly different for metered and nonmetered subscriptions:
   - If you have a nonmetered subscription, contact an Oracle Sales Representative to extend your subscription by the number of permanent nodes you want to add. You don't have to say how many will be used as permanent Hadoop nodes and how many as edge nodes yet. You'll make that choice in the steps below.

     You'll receive an e-mail notification when the nodes have been added to your subscription.
   - If you have a metered subscription, continue with the steps below. You cloud credits will be applied according to what you add.
2. Go to your Big Data Cloud Service **Services** page
3. Click the name of the cluster that you want to extend.
4. On the **Service Overview** page for that cluster, click  at the top of the page, and select **Extend/Shrink**.
5. In the **Extend/Shrink Service** dialog box, enter your Cloudera administrator password, select how many of the extra nodes will be permanent Hadoop nodes and how many will be edge nodes, and then and click **Extend**.

   Four permanent Hadoop nodes are required before you can assign any additional nodes as edge nodes. For example, if your cluster has three permanent Hadoop nodes and you've purchased an entitlement to two more, you must allocate one as a permanent Hadoop node. You can then allocate the fifth node either as a permanent Hadoop node or an edge node. If your cluster already has four nodes and you purchase an entitlement to more nodes, you can allocate them as any combination of permanent Hadoop nodes and edge nodes.

When successful, the new status appears on the **Service Overview** page for the cluster..

Adding and Removing Cluster Compute Nodes (Bursting)

You can add and remove *cluster compute* nodes at any time. This is sometimes called “bursting.”

Cluster compute nodes have processing resources (OCPUs) and memory only, with no storage. That makes it possible to add them temporarily, when your processing needs increase, and delete them when you no longer need them.

You don’t have to contact an Oracle representative and you don’t have to purchase additional. entitlements. You’re charged only for the number of nodes you allocate and only for the time they exist in the cluster. See About Oracle Big Data Cloud Service Nodes.

To add cluster compute nodes to a cluster:

1. On the **Services** page of your console, click the **Big Data Cloud Service** link at the top of the page, and then select the service you want to update,..
2. From the , select **Modify**.
3. On the Modify Oracle Big Data Cloud Service page, find the **Additional Number of OCPU (Cores)** field. A message under the field shows the number of OCPUs currently assigned to this instance. This message shows **0** when you have no cluster compute nodes in use.

   An Oracle Compute Unit (OCPU) is a single core in the virtual CPU. There are 32 Oracle OCPUs in a single cluster compute node. When you make a choice to add or remove cluster compute nodes, you identify them by their number of OCPUs, The following are supported.

   - 32 OCPUs = 1 node
   - 64 OCPUs = 2 nodes
   - 96 OCPUs = 3 nodes
   - 128 OCPUs = 4 nodes
   - 160 OCPUs = 5 nodes
   - 192 OCPUs = 6 nodes
   - 224 OCPUs = 7 nodes
   - 256 OCPUs = 8 nodes
   - 288 OCPUs = 9 nodes
   - 320 OCPUs = 10 nodes
   - 352 OCPUs = 11 nodes
   - 384 OCPUs = 12 nodes
   - 416 OCPUs = 13 nodes
   - 448 OCPUs = 14 nodes
   - 480 OCPUs = 15 nodes

   a. To add OCPUs, enter a positive number, in increments of 32, up to a total of 480 OCPUs for the instance. Then click **Next**.

b. To remove OCPUs, enter a negative number, in increments of 32, up to the number assigned to the instance. Then click **Next**.

4. On the Modify Instance Confirm page, review your changes and then click **Modify** to initiate the changes.

   After you've initiated this process is complete, you'll receive an e-mail that says the nodes have been added to the instance.

   If you're only adding cluster compute nodes, they're added automatically to the cluster. If you're also adding permanent nodes, you must select **Extend/Shrink** from the top menu on the cluster's **Service Overview** page to add them all at the same time.

Control Network Access to Services

An administrator must configure an Oracle Big Data Cloud Service cluster to control access from incoming network traffic. The configuration specifies whether requests from specified clients will be accepted or denied for services at specified ports. In a new cluster, all ports on all nodes are closed by default, except for port 22 (for SSH access), which is open on all nodes. Other nodes must be opened explicitly.

The following topics tell how to configure a cluster to control network access:

Topics:

How Network Access Control is Determined by the Host Region

Oracle Big Data Cloud Service provides different ways to control network access to a cluster, depending on what kind of region it's in.

The region that hosts a service instance is selected or assigned when the service instance is created. It can be either of the following:

- An Oracle Cloud Infrastructure region, which contains one or more *availability domains*. Each availability domain contains one or more data centers. See Regions and Availability Domains.
- An Oracle Cloud Infrastructure Classic region, which contains one or more data centers that are *not* part of an availability domain. See About Oracle Data Regions.

In either case, network access is controlled by a firewall that uses the Linux iptables utility to filter network traffic. The firewall can be configured to accept or deny network requests from specified clients for services at specified ports.

In a new cluster, all ports on all nodes are closed by default, except port 22 (for SSH access), which is open on all nodes. You must open any other ports by using one of the following methods, depending on your region:

- If your service is hosted in data center in an availability domain, you can control access on any port. You can manage the configuration by using `bdacli firewall`

commands at a command prompt or by using graphical tools in Oracle Big Data Manager. See:

  - Controlling Network Access for Services in Availability Domains
  - Configuring the Firewall Through Oracle Big Data Manager in *Using Oracle Big Data Manager*.

- If your service is hosted in a data center that *isn't* part of an availability domain, you can control incoming access on three ports only: `22` (SSH), `7183` (Cloudera Manager) and `8888` (Hue and Oracle Big Data Manager). See Controlling Network Access for Services That Are Not in Availability Domains.

Controlling Network Access for Services in Availability Domains

When a cluster is hosted in a data center in an availability domain, you can use `bdacli firewall` commands to control network access through the Oracle Big Data Cloud Service firewall. Any port on any node can be opened or closed.

Managing the Firewall With bdacli firewall Commands

You must have administrative privileges to configure the firewall.

To configure the firewall, use SSH to connect to the third node of the cluster as the `root` user, and use `bdacli firewall` commands.

> **Note:**
>
> You can also configure the firewall by using graphical tools in Oracle Big Data Manager. See Configuring the Firewall Through Oracle Big Data Manager.

Syntax

```
bdacli firewall command [arguments]
```

Commands

`list`
Lists all rules in the cluster. The `–json` flag returns the output formatted in JavaScript Object Notation (JSON).
Syntax:

```
bdacli firewall list [--json]
```

```
# bdacli firewall list
BDCS Network Firewall.
Using nodes: scaj53bda06 scaj53bda07 scaj53bda08 scaj53bda09
ID | Source         | Dest         |Port    | Protocol | Comments
--------------------------------------------------------------------------------
-
```

```
1  | 192.168.0.0/24  | scaj53bda06   | cm      | all   | Cloudera manager
rule
2  | 192.168.0.0/24 | scaj53bda08   | cm      | all   | Cloudera manager
rule
3  | 192.168.0.0/24  | scaj53bda08   | hue    | all   | Hue rule
4  | 192.168.0.0/24  | scaj53bda06   | 8080  | all   | Webserver rule
```

add | addbulk
Adds a rule or rules to the cluster IP rules. Use addto add a single rule and addbulk to add multiple rules in a single command.
Syntax::

```
bdacli firewall add
    src-ipaddr-1 [/mask]
    node-name
    dest-port
    protocol
    --comment comment


bdacli firewall addbulk
    --rule
        src-ipaddr-1 [/mask]
        node-name
        dest-port
        protocol
    --comment comment
    --rule
        src-ipaddr-2 [/mask]
        node-name-2
        dest-port-2
        protocol-2
    --comment comment
```

Examples:

```
# bdacli firewall add 192.168.0.0/24 scaj53bda08 cm all
--comment "Cloudera manager rule"
BDCS Network Firewall.
Using nodes: scaj53bda06 scaj53bda07 scaj53bda08 scaj53bda09
Adding single rule.
File doesn't exists, creating file...
Rule added.
Distributing configuration, please wait.
Configuration reloaded.


# bdacli firewall addbulk --rule 192.168.0.0/24
scaj53bda08 cm all --comment "Cloudera manager rule" --rule 1
92.168.0.0/24 scaj53bda08 hue all --comment "Hue rule" --rule
192.168.0.0/24 scaj53bda06 8080 all --comment "Webserver rule"
BDCS Network Firewall.
```

```
Using nodes: scaj53bda06 scaj53bda07 scaj53bda08 scaj53bda09
Adding multiple rules.
Rule added.
Rule added.
Rule added.
Distributing configuration, please wait.
Configuration reloaded.
```

replace
Deletes all rules on firewall and adds the rules to the cluster ip rules. Multiple lines can be added.
Syntax:

```
bdacli firewall replace
    rule-id
        src-ipaddr [/mask]
        node-name
    dest-port?
        protocol
    --comment comment


bdacli firewall replaceall
    --rule
        src-ipaddr-1 [/mask]
        dest-ipaddr-1|node-name-1
        dest-port-1
        protocol-1
    --comment comment
    --rule
        src-ipaddr-2 [/mask]
        dest-ipaddr-2|node-name-2
        dest-port-2
        protocol-2
    --rule
        src-ipaddr-3 [/mask]
        dest-ipaddr-3|node-name-3
        dest-port-3
        protocol-3
    --comment comment
```

Example:

```
# bdacli firewall replace 1 192.168.0.0/24 scaj53bda06
10001 all --comment "Custom service rule"
BDCS Network Firewall.
Using nodes: scaj53bda06 scaj53bda07 scaj53bda08 scaj53bda09
Replacing a single rule.
Rule deleted.
Rule added.
```

```
Distributing configuration, please wait.
Configuration reloaded.
```

Example:

```
# bdacli firewall replaceall --rule 192.168.0.0/24
scaj53bda08 cm all --comment "Cloudera manager rule" --rule
192.168.0.0/24 scaj53bda08 hue all --comment "Hue rule" --rule
192.168.0.0/24 scaj53bda06 8080 all --comment "Webserver rule"
BDCS Network Firewall.
Using nodes: scaj53bda06 scaj53bda07 scaj53bda08 scaj53bda09
Firewall configuration back to factory settings.
Adding new rules.
File doesn't exists, creating file...
Rule added.
Rule added.
Rule added.
Distributing configuration, please wait.
Configuration reloaded.
```

`delete`
Deletes a single rule on the cluster.
Syntax:

```
bdacli firewall delete rule-id
```

`reset`
Deletes all customer-made rules on the cluster.
Syntax:

```
bdacli firewall reset
```

`reload`
Reprocesses `firewall json` to generate iptables script and runs new script.
Syntax:

```
bdacli firewall reload
```

Arguments

The following table describe the arguments to the above commands.

Argument	Description
`src-ipaddr[/mask]`	The IP address of the incoming connection, in the format `AA.BB.CC.DD` in dot decimal notation. A whole network can be added, by adding the prefix mask with the notation `AA.BB.CC.DD/EE`.

Argument	Description
`node-name`	The hostname (of the node) that will receive the incoming connection, or the keyword `all` to use client interface of all nodes. Hostname can be a full domain or just the name of the node.
`dest-port`	The destination port that will receive the connection. The number can be from 1 to 65535. A range can be defined in format `a:b` or you can use the keyword `all` to use all ports. Keywords `hue` and `cm` are valid for ports 8888 and 7183 respectively.
`protocol`	The protocol of the connection. It can be one of the following: `tcp`, `udp`, `icmp`, or the keyword `all` as a wildcard for any protocol.
`comment`	Comments are added in IP tables and in the `firewall.json` file. They have no functional value. It is just in case the administrator wants to add a description. Comments are optionally added in replace and add operations with the flag `–comment`.

Controlling Network Access for Services That Are Not in Availability Domains

When a cluster is hosted in a data center that is *not* in an availability domain, network access to services is controlled through *whitelists*. Access is allowed only on ports 22 (SSH), 7183 (Cloudera Manager), and 8888 (Hue and Oracle Big Data Manager).

A whitelist configuration specifies whether network requests from specified clients will be accepted or denied for services at specified ports. When a connection tries to establish itself, `iptables` looks for a matching client IP address or range of IP addresses in the whitelist. If it doesn't find one, it uses the default action.

You must have administrative privileges to configure whitelists.

Services That Can Be Added to the Whitelist

The following table shows the services that can be configured through the whitelist, their default port numbers, and the default access enforced through the whitelist:

Service	Port	Default Access
Cloudera Manager	7183	Deny access
Hue	8888	Deny access
Oracle Big Data Manager	8888 (same port as used Hue)	Deny access
Secure Shell (SSH)	22	Allow access

Managing the Whitelist With the bdacli bdcs_whitelist Command

Use the `bdacli bdcs_whitelist` command to manage the whitelist configuration for a cluster.

Run the command as the `root` user on the primary host of the cluster. To find out what your primary host is, connect to any node and enter `bdacli getinfo cluster_primary_host`. For example:

```
# bdacli getinfo cluster_primary_host
host1891
```

bdacli bdcs_whitelist Usage

Syntax

```
bdacli bdcs_whitelist parameters
```

Parameters

The following table describe the parameters for the `bdacli bdcs_whitelist` command.

Parameter	Description
`reset_default_config`	Resets the files to empty and applies the default configuration. Denies all incoming traffic to the server except SSH, which allows all traffic. This only affects ports controlled by the whitelist. All other `iptables` configurations aren't touched.
`reload_config`	Deletes all `iptables` rules on ports controlled by the whitelist files and reprocesses what is in the whitelist files. If the files are empty, the default configuration is applied. If there are one or more entries, then all traffic is denied except the whitelist in the files.
`allow` *`service ip/range`*	Adds an IP address or a range of IP addresses to the whitelist of the named service and runs an `iptables` command to allow access to that service from those IP addresses. See the **Variable** table, below, for descriptions of the *`service`* and *`ip/range`* variables.
`deny` *`service ip/range`*	Removes an IP address or a range of IP addresses to the whitelist of a specific service and runs an `iptables` command to deny access to that service from those IP addresses. See the **Variables** table, below, for descriptions of the *`service`* and *`ip/range`* variables.

Variables

The following table describe the variables for the `bdacli bdcs_whitelist allow` and `bdacli bdcs_whitelist deny` commands.

Variable	Description
service	One of the following: • `cloudera_manager` • `hue` • `ssh` • `all` — where the command applies to all the above services. We recommend that you do not use `all`, but rather run the command for just the services you need to open. This is recommended because it leaves the critical SSH settings alone (which default to open) unless you explicitly change them. Changing SSH settings from the default can lock you permanently out of the cluster.
`ip/range`	One of the following: • Specify a single IP address by simply giving the address, for example: **`192.0.2.48`** • Specify a range of IP addresses by using either of the following: **`192.0.2.0/24`** **`192.0.2.0/255.255.255.0`** Both of the above ranges mean `192.0.2.0` to `192.0.2.255` inclusive

Example

```
# bdacli bdcs_whitelist allow cloudera_manager 198.51.100.48
BDCS Network Services Firewall & Whitelist
host1891.us.example.com
host1892.us.example.com
host1893.us.example.com
host1894.us.example.com
host1895.us.example.com
```

In the above example:

- `bdacli bdcs_whitelist allow cloudera_manager` specifies that Cloudera Manager will accept requests from the specified client IP addresses.
- `198.51.100.48` specifies that the client with that IP address will be allowed access to the service.
- `host1891.us.example.com` through `host1895.us.example.com` are the host names of the nodes of the cluster.

Opening the Port for Big Data Manager

Big Data Manager uses the same port as Hue (8888). Therefore, opening the port for Hue also opens it for Big Data Manager. For example, either of the following commands enables Big Data Manager as well as Hue:

```
bdcs_whitelist allow hue 198.51.100.48
```

```
bdcs_whitelist allow all 198.51.100.48
```

View All Clusters

To view all your clusters, open the Oracle Big Data Cloud Service **Services** page.

Services Page

The Oracle Big Data Cloud Service **Services** page lists all the clusters in the current identity domain.

The following table describes each item on the page.

> **Note:**
>
> Depending on your role, you may not have access to all the options on this page. Only a user with Administrator privileges has access to all.

Element	Description
Services	The **Clusters** link at the top of the page displays the current page, the Oracle Big Data Cloud Service **Services** page.
Activity	Click the **Activity** link at the top of the page to display the **Activity** page, where you can search and display logs of past operations.
Create Service button	Click this button to create a new cluster, using resources allocated for a service instance. See Create a Cluster.
cluster_name	Click the name of a cluster to display the **Service Overview** page for the cluster, where you can review and manage aspects of the cluster. .
Version *n.n.n*	The version of the Oracle Big Data Cloud Service software running on this cluster.

View Details About a Cluster

To view detailed information for a cluster:

1. Open the Oracle Big Data Cloud Service Services page. See Open the Oracle Big Data Cloud Service Console.
2. Find the cluster you want, and click its name to display the **Service Overview** page for the cluster.

Service Overview Page

The Oracle Big Data Cloud Service **Service Overview** page displays details about the cluster.

The band at the top of the page includes the following:

Item	Description
Big Data Cloud Service	Click this link to return to the Oracle Big Data Cloud Service **Services** page, for an overview of all your clusters.
cluster_name	The name of this cluster.
 (for the cluster)	Click the icon to open a menu with these options: • **Cloudera Manager Console**—Open the Cloudera Manager console to manage the cluster. • **Big Data Manager Console**—Open the Cloudera Manager console to manage copy jobs. • **Start**—Start the cluster if it's not running. • **Stop**—Stop the cluster. • **Restart**—Restart the stopped cluster. • **Extend/Shrink**—Extend and/or shrink the cluster, based on the configuration in the service instance used for this cluster. In other words, if you have modified your subscription to add or remove nodes, selecting this command will initiate the updates on the cluster. • **Update**—Edit the description of the cluster. • **Add Association**—If you've subscribed to the Big Data SQL add-on to Oracle Big Data Cloud Service and you haven't associated the services yet, select this items to make the association. • **Service Credentials**—Add an SSH public key for secure access. • **View Activity**—Search and display logs of operations on this cluster.

Cluster Details

The information listed under **Service Overview** shows details about the cluster.

Item	Description
Status	The operational status of the cluster, including Creating, Updating, Restarting, Terminating. On-boarding, Ready, etc.
BDA Version	Version of Oracle Big Data Cloud Service software running on this cluster.
Description	A description of the cluster.
Big Data Appliance System	The name of the resource configuration used for this cluster.

Node Details

Each row under **Resources** shows details about a single node in the cluster.

Item	Description
Host Name	The name of the host for this node.
Public IP	The public IP address of this node.
Role	The type of node, for example, Hadoop permanent node or edge node.

Item	Description
(for the node)	Click the icon to open a menu with these options: • **Cloudera Manager Console**—Open the Cloudera Manager console, where you can manage the cluster. • **Big Data Manager Console**—Open the Cloudera Manager console, where you can manage copy jobs. • **Start**—Start the cluster if it's not running. • **Stop**—Stop the cluster. • **Restart**—Restart the stopped cluster. • **Extend/Shrink**—Extend and/or shrink the cluster, based on the configuration in the service instance used for this cluster. In other words, if you have modified your subscription to add or remove nodes, selecting this command will initiate the updates on the cluster. • **Update**—Edit the description of the cluster. • **Add Association**—If you've subscribed to the Big Data SQL add-on to Oracle Big Data Cloud Service and you haven't associated the services yet, select this items to make the association. • **Service Credentials**—Display a dialog box, where you can associate an SSH public key with the cluster. The options are: – **Key File Name**—Provide the location and name of a file containing the key. – **Key Value**—Paste in a key value. – **Create a New Key**—Generate a new SSH key pair. • **View Activity**—Display logs of operations on this cluster.

Use HDFS Transparent Encryption

HDFS Transparent Encryption protects Hadoop data that's at rest on disk. When the encryption is enabled for a cluster, data write and read operations on encrypted *zones* (HDFS directories) on the disk are automatically encrypted and decrypted. This process is "transparent" because it's invisible to the application working with the data. HDFS Transparent Encryption does not affect user access to Hadoop data, although it can have a minor impact on performance.

Prerequisite

The cluster where you want to use HDFS Transparent Encryption must have Kerberos enabled.

> **Important:**
>
> Security Setup must be enabled when creating the cluster. The person creating the cluster must choose the **Security Setup: Enabled** option on the Security page of the Create Cluster wizard, as described in Create a Cluster. You can't enable Kerberos for a cluster after it's been created.

When you create a cluster with Security Setup enabled, the following takes place:

- HDFS Transparent Encryption is enabled on the cluster. You can verify this by entering the following at the command line:

  ```
  bdacli getinfo cluster_hdfs_transparent_encryption_enabled
  ```

- MIT Kerberos, Sentry, Network Firewall, Network Encryption, and Auditing are also enabled on the cluster.
- Two principals are created as part of the Kerberos configuration:
  - `hdfs/clustername@BDACLOUDSERVICE.ORACLE.COM` — The password for authenticating this principal is your Cloudera admin password.
  - `oracle/clustername@BDACLOUDSERVICE.ORACLE.COM` — The password for authenticating this principal is your Oracle operating system password.

  In both cases, *clustername* is the name of your cluster and `BDACLOUDSERVICE.ORACLE.COM` is the Kerberos realm for Oracle Big Data Cloud Service.
- A Key Trustee Server is installed and configured on the cluster. This server is used for managing keys and certificates for HDFS Transparent Encryption. See Cloudera Navigator Key Trustee Server for more information about this server. (You should back up Key Trustee Server databases and configuration files on a regular schedule. See the Cloudera documentation topic, Backing Up and Restoring Key Trustee Server.)

Creating Encryption Zones on HDFS

An encryption zone is an HDFS directory in which the contents are encrypted on a write operation and decrypted on a read operation.

> **See Also:**
>
> Cloudera documentation Managing Encryption Keys and Zones.

Prerequisites:

1. Make sure services are healthy in Cloudera Manager. Especially make sure the Key Trustee service is healthy.
2. Make sure the two KMS hosts are in sync.

   On each KMS host run the commands below as the `root` user. The output should be the same on each host. If not, open a service request (SR) with Oracle Support, because that would indicate a problem synchronizing the two Key Management Servers.

   ```
   # ls -l /var/lib/kms-keytrustee/keytrustee/.keytrustee
   # cksum /var/lib/kms-keytrustee/keytrustee/.keytrustee/*
   # gpg --homedir /var/lib/kms-keytrustee/keytrustee/.keytrustee --
   fingerprint;
   ```

Perform the following steps on any node of the cluster as the `root` user, unless otherwise specified.

To create an encryption zone:

1. Create an encryption key for the zone:

a. Authenticate the `hdfs/clustername@BDACLOUDSERVICE.ORACLE.COM` principal using your Cloudera password, for example:

```
# kinit -p hdfs@BDACLOUDSERVICE.ORACLE.COM
Password for hdfs@BDACLOUDSERVICE.ORACLE.COM: ****
```

b. Create the encryption key, using the following command::

```
hadoop key create keyname
```

For example:

```
# hadoop key create bdakey
bdakey has been successfully created with options
Options{cipher='AES/CTR/NoPadding', bitLength=128,
description='null',
attributes=null}.
org.apache.hadoop.crypto.key.kms.LoadBalancingKMSClientProvider@4145
bad8 has been updated.
```

2. Create a new empty directory and make it an encryption zone using the key generated above with the following two commands:

```
# hadoop fs -mkdir path
# hdfs crypto -createZone -keyName keyname -path path
```

For example:

```
# hadoop fs -mkdir /zone
# hdfs crypto -createZone -keyName bdakey -path /zone
Added encryption zone /zone
```

> **Note:**
>
> Encryption zones must be created as the super user, but after that access to encrypted file data and metadata is controlled by normal HDFS file system permissions.

3. Verify creation of the new encryption zone by running the `-listZones` command; for example:

```
# hdfs crypto -listZones
/zone bdakey
```

Adding Files to Encryption Zones

Use the `hadoop fs -put` command to add files to the encryption zone.

For example:

```
# hadoop fs -put helloWorld /zone
```

Viewing Keys in Encryption Zones

Use the `hadoop key list` command to view keys in an encryption zone.

For example:

```
# hadoop key list
Listing keys for KeyProvider:
org.apache.hadoop.crypto.key.kms.LoadBalancingKMSClientProvider@xxxxxx
MYKEY1
MYKEY2
```

Upgrade Oracle Big Data Cloud Service Software Through the Console

You can upgrade cluster software by using the interactive tools in the Oracle Big Data Cloud Service console.

You may have to apply a patch before you can use the console for subsequent upgrades, as shown in the following table:

To upgrade a cluster that was...	Do this...
Created on release 18.3.1 or later	You can use the console to upgrade to subsequent releases.
Created on release 18.2.5	Manually apply patch 22405911. Then you can use the console for updates to subsequent releases.
Created before release 18.2.5	It isn't currently possible to use the console to patch or upgrade a cluster created before 18.2.5. However, it will be available in a future release.

Upgrading Cluster Software Through the Console

To upgrade cluster software through the Oracle Big Data Cloud Service console:

1. Open the console.
2. If necessary, navigate to the Instances page.
3. If a patch is available for a cluster, the row that shows the details for that cluster includes a link, **One or more patches are available**. Click the link to display the Patching page for the cluster.
4. Click the **PSU** menu to the right of the patch information, and select **Precheck**.
5. When the precheck is complete and successful, click the PSU menu again and select **Patch**.

Patching a Release 18.2.5 Cluster

Clusters created with Oracle Big Data Cloud Service release 18.2.5 must be patched before you can use the console tools to upgrade to subsequent releases.

> **Note:**
>
> When you run the patch precheck on a cluster that hasn't been patched, a message is displayed: **This cluster needs a patch in order to proceed with upgrade. Please contact support to download and install the patch.**

To patch a cluster that was created on release 18.2.5, contact Oracle Support to download patch 22405911 and apply it to the cluster (domU nodes only). Apply the patch individually to each of the nodes in the cluster.

To apply the patch, do the following on each node:

1. Copy the `bda-4.11.4-1.el6.x86_64.rpm` file to the node.
2. Upgrade to the new `bda rpm` package by running either of the following:

```
# rpm -e bda
# rpm -Uvh bda-4.11.4-1.el6.x86_64.rpm
```

or

```
# rpm -e bda
# rpm -Uvh --force bda-4.11.4-1.el6.x86_64.rpm
```

3. Run the following command:.

```
# service bda-monitor restart
```

Restart a Cluster

To restart a cluster:

1. Go to the **Service Overview** page of the cluster you want to restart.
2. Click the icon at the top of the page, and select **Restart**.

Restart a Cluster Node

To restart a cluster node:

1. Go to the **Service Overview** page of the cluster with the node you want to restart.
2. Click the icon on the row of the node you want to restart, and select **Restart**.

Update the SSH Public Key for a Cluster

A cluster must have a Secure Shell (SSH) key pair associated with it to permit secure access for the opc user. When you create a cluster, you must specify the public key. After the cluster has been created, you can replace that key (or any subsequently assigned key) with a new one.

To replace the SSH public key:

1. Go to the **Service Overview** page for the cluster whose SSH public key you want to change.
2. From the menu at the top of the page, select **Service Credentials.**.

   The **SSH Key for VM Access** dialog is displayed.
3. Use any of the following to specify the new public key:
   a. Select **Key File Name** and then click **Select File** to select a file containing the new public key.
   b. Select **Key Value** and delete the current key value shown in the text area; then paste in a new one. Make sure the value does not contain extra spaces or line breaks and does not have extra line breaks at the end.

      Note: Some tools generate public SSH keys with a line break at the end, and that is allowed here. However, you shouldn't add any additional line breaks.
   c. Select **Create New Key**, and a new key pair will be generated for you.

Support Multiple Key Pairs for Secure Shell (SSH) Access

By default, the Oracle Big Data Cloud Service opc user has Secure Shell (SSH) access to all the nodes of the cluster when using the SSH key pair that was provided when the cluster was provisioned. You can also provide SSH access from different clients and for other users. For example, you may want to provide opc access to an Oracle Big Data Discovery administrator who accesses the cluster from a different computer, or you may want to create other users with different access rights.

Adding SSH Support for the opc User Using a Different Key Pair

When anOracle Big Data Cloud Service cluster is provisioned, /home/opc/.ssh/authorized_keys files are created on all the nodes of the cluster. The authorized_keys files contain the SSH public key that was provided when the cluster was provisioned.

To add an additional public key for the opc user,

1. Obtain the new SSH public key.

   The user who needs access to the cluster can create the SSH key pair, retain the private key, and transfer the public key to the Oracle Big Data Cloud Service administrator. Or, the administrator can create the new key pair and transfer the private key to the other user. See Generating a Secure Shell (SSH) Public/Private Key Pair.
2. Connect as the opc user to a node to which you want to add the key. See Connect to a Cluster Node Through Secure Shell (SSH).

3. On a new line of the `/home/opc/.ssh/authorized_keys` file, paste the contents of the new SSH public key file. Do not add extra lines or line breaks.
4. Repeat the process on every node to which you want to provide access by using the new key pair.

Adding SSH Support for Other User Accounts

To add an SSH key pair for a user other than `opc`:

1. Obtain the new SSH public key.

   The user who needs access to the cluster can create the SSH key pair, retain the private key, and transfer the public key to the Oracle Big Data Cloud Service administrator. Or, the administrator can create the new key pair and transfer the private key to the other user. See Generating a Secure Shell (SSH) Public/Private Key Pair.
2. Connect as the `opc` user to a node to which you want to add the key. See Connect to a Cluster Node Through Secure Shell (SSH).
3. Create a `/home/`*`user`*`/.ssh/authorized_keys` file, where `user` is the name of the user who will have SSH access.
4. Paste the contents of the new SSH public key file into`/home/`*`user`*`/.ssh/authorized_keys` file. Do not add extra lines or line breaks.
5. Repeat the process on every node to which you want to provide SSH access for the user.

Delete a Cluster

To delete a cluster:

1. Go to your Oracle Big Data Cloud Service **Services** page.
2. Click the  icon on the row of the cluster you want to delete, and select **Delete**.

   The cluster is deleted.

3

Manage Oracle Big Data Cloud Service System Software at the Command Line

The Oracle Linux operating system and Cloudera's Distribution including Apache Hadoop (CDH) underlie all other software components installed on Oracle Big Data Cloud Service .

The following sections describe and tell how to use command line utilities for interact with the system software.

Topics

Command Line Utilities for Managing Oracle Big Data Cloud Service Software

The following command line utilities are available for managing the software on anOracle Big Data Cloud Service cluster::

- The `bdacli` utility can be used to upgrade or patch the software and to query various configuration files to return information about the cluster, and nodes. See Use bdacli to Patch Software and to Display Configuration Information.
- The `dcli` utility executes commands across a group of nodes on a cluster and returns the output. See Executing Commands Across a Cluster Using the dcli Utility.

Patch and Upgrade Oracle Big Data Cloud Service Software

This chapter explains how to patch, update, and reconfigure software through the command line utilities on an Oracle Big Data Cloud Service cluster.

> **Note:**
>
> Ensure that you know the current passwords for the operating system `root` and `oracle` users and the the Cloudera Manager `admin` user.

Topics

- Using the Mammoth Command-Line Utility to Upgrade Software on a Cluster
- Using Mammoth to Install a One-Off Patch

Also see Upgrade Oracle Big Data Cloud Service Software Through the Console

Using the Mammoth Command-Line Utility to Upgrade Software on a Cluster

The procedure for upgrading the software is the same whether you are upgrading from one major release to another or just applying a patch set.

> **Note:**
>
> The easiest way to patch and upgrade a cluster is through the Oracle Big Data Cloud Service console. See

The process upgrades all components of the software stack including the firmware, Oracle Linux Unbreakable Enterprise Kernel (UEK), CDH, JDK, and Oracle Big Data Connectors.

To upgrade only Oracle Big Data Connectors, and no other components of the software stack, contact Oracle Support for assistance.

Software downgrades are not supported.

Upgrading the Software

Follow these procedures to upgrade the software on an Oracle Big Data Cloud Service cluster to the current version.

Prerequisites

You must know the passwords currently in effect for the cluster, which the Mammoth utility will prompt you for:

- `oracle`
- `root`
- Cloudera Manager `admin`

Upgrading to the Current Software Version

Making sure cluster services are healthy before upgrade, and especially after reboot is very important. Manual steps will be needed to resume.

Upgrade the Oracle Big Data Cloud Service software to the current software version as follows. This is a summary. Refer to My Oracle Support (MOS) Doc ID 2101906.1 for details, including prerequisites, further information on the steps below, and known issues.

> **Note:**
>
> All Oozie jobs should be stopped before the upgrade. Failure to do this may cause the upgrade to fail.

1. Download and unzip the Mammoth bundle, as described in Downloading the Mammoth Software Deployment Bundle.. Mammoth is a command-line utility for installing and configuring the Oracle Big Data Cloud Service software.

   You must be logged in as root to node 1 of the cluster.

2. Change to the `BDAMammoth` directory.

   ```
   # cd /opt/oracle/BDAMammoth
   ```

3. Run the `mammoth` command with the -p option:

   ```
   # ./mammoth -p
   ```

   Mammoth automatically upgrades the base image if necessary.

4. After all nodes reboot, perform the following checks.

   a. Check uptime.

      ```
      # dcli -C uptime
      ```

   b. Check /root/BDA_REBOOT_SUCCEEDED.

      ```
      # dcli -C ls -ltr /root/BDA_REBOOT_SUCCEEDED
      ```

      Note: Note: If there is no `BDA_REBOOT_SUCCEEDED`, check for `/root/BDA_REBOOT_*` and `/root/bda_reboot_status`.

   c. Verify that the kernel and JDK are upgraded.

      ```
      # dcli -C uname -a
      # dcli -C java -version
      ```

   d. Check that all Cloudera Configuration Manager services are healthy. You may need to manually restart some services.

      > **Note:**
      >
      > During any upgrade, it is **crucial that all services in Cloudera Manager are healthy after the reboot before continuing**. Failure to do so will result in upgrade failures.

5. After the reboot and the post reboot checks, log on to node 1 of the cluster and rerun `mammoth -p` in order to resume the upgrade.

```
# cd /opt/oracle/BDAMammoth
# ./mammoth -p
```

6. When the upgrade is complete, perform the post-upgrade validation steps described in the MOS document (Doc ID 2101906.1).

Using Mammoth to Install a One-Off Patch

One-off patch bundles provide a fix to specific bugs in one or more releases. You use Mammoth to apply the patch to your cluster.

To install a one-off patch bundle:

1. Download the patch bundle from the Automated Release Update (ARU) system to a directory such as `/tmp` on the first node of the Oracle Big Data Cloud Service cluster.

   The file is named `BDA-patch-release-patch.zip`. The examples in this procedure use the name `BDA-patch-4.3.1-123456.zip`.

2. Unzip the file. For example:

```
# unzip BDA-patch-4.3.0-123456.zip
Archive:  BDA-patch-4.3.0-123456.zip
   creating: BDA-patch-4.3.0-123456/
  inflating: BDA-patch-4.3.0-123456/BDA-patch-4.3.0-123456.run
  inflating: BDA-patch-4.3.0-123456/README.txt
```

3. Change to the patch directory created in Step 2. For example:

```
$ cd BDA-patch-4.3.0-123456
```

4. Extract the contents of the run file. For example:

```
$ ./BDA-patch-4.3.0-123456.run
Big Data Appliance one-off patch 123456 for v4.3.0 Self-extraction

Removing existing temporary files

Generating /tmp/BDA-patch-4.3.0-123456.tar
Verifying MD5 sum of /tmp/BDA-patch-4.3.0-123456.tar
/tmp/BDA-patch-4.3.0-123456.tar MD5 checksum matches

Extracting /tmp/BDA-patch-4.3.0-123456.tar to /opt/oracle/BDAMammoth/
patches/123456
Removing temporary files

Please "cd /opt/oracle/BDAMammoth" before running "./mammoth -p 123456"
```

5. Change to the BDAMammoth directory:

```
$ cd /opt/oracle/BDAMammoth
```

6. Install the patch. For example:

```
$ ./mammoth -p 123456
```

Alternatively, you can use the `bdacli` command. See Use bdacli to Patch Software and to Display Configuration Information.

Use bdacli to Patch Software and to Display Configuration Information

The Oracle Big Data Cloud Service command-line interface (bdacli) queries various configuration files to return information about the cluster, nodes, and software patches.

Syntax

`bdacli` *action* [*parameters*]

Actions

`help`
Displays general usage information for `bdacli`, a list of actions, and a list of supported parameters for the `getinfo` action.

`{add | remove} patch` *`patch_number`*
Adds or removes a software patch on Oracle Big Data Cloud Service that matches *patch_number*. You must log in as `root` to use `add` or `remove`.

`admin_cluster` *`parameter node_name`*
Enables you to administer the nodes in a cluster in response to a failing node. The following table describes the parameters.

Parameter	Description
`decommission`	Removes the specified node from the cluster and decommissions the node in Cloudera Manager. It also updates the Mammoth files. You can decommission a failing, noncritical node. Note that critical services on the node must be moved first.
`recommission`	Removes the node from the list of decommissioned nodes, and recommissions the node in Cloudera Manager. Use this command after decommissioning and repairing a failing node.
`migrate`	Moves the services from a critical node to a noncritical node, and decommissions the failing node in Cloudera Manager. You specify the name of the failing critical node, and the utility selects the noncritical node for the migration. When migration is complete, the new node has all of the functionality of the original critical node. You can only migrate a critical node, and should do so only when it is failing.

Parameter	Description
`reprovision`	Restores a node to the cluster as a noncritical node, and recommissions the node in Cloudera Manager. Use this command after migrating the services of a critical node and repairing the failing node.

`{start | stop | restart | status} service`
Starts, stops, restarts, or returns the current status of a service on a cluster or a specific node.
The following table describes the service parameters:

Parameter	Description
`big_data_sql_cluster`	Oracle Big Data SQL on all nodes of the cluster
`big_data_sql_server` *`node_name`*	Oracle Big Data SQL on a specified node of the cluster. Use `bdacli` with this parameter only from the first node of the cluster, where the current `config.json` file is stored.

`getinfo [parameter]`
Returns a list of `getinfo` parameters. If you include a parameter name in the command, then `getinfo` returns information about that system component:

- **Cluster parameters**: Describes a logical Oracle Big Data Cloud Service cluster. The bdacli command queries the current config.json file for the Hadoop cluster where the command executes. See Cluster Parameters.
- **Node parameters**: Describes a node. The bdacli command queries the operating system of the node where the bdacli command executes. See Node Parameters.
- **One-off patch parameters**: Provides information about one-off patches. See One-Off Patch Parameters.

Parameter names that end with an "s" return lists. Boolean parameters return a string of either `true` or `false`.

Cluster Parameters

The following tables describe the cluster parameters.

- General Cluster Parameters
- Oracle Big Data Connectors Status Parameters
- Cluster Network Parameters
- Cluster Security Parameters

The following table describes the general cluster parameters for `bdacli getinfo`.

General Cluster Parameters

Parameter	Returns
`cluster_asr_installed`	`true` if Auto Service Request is configured for this cluster; `false` otherwise

Parameter	Returns
`cluster_big_data_sql_enabled`	`true` if Oracle Big Data SQL is enabled for this cluster; `false` otherwise.
`cluster_cdh_version`	The version of Cloudera's Distribution including Apache Hadoop installed on this cluster, such as `4.5.0-ol6`.
`cluster_cm_server`	The Cloudera Manager address, including the node name and port number, such as `bda1node03.example.com:7180`.
`cluster_cm_version`	The version of Cloudera Manager running on this cluster, such as `4.8.0-ol6`.
`cluster_name`	The name of the cluster, such as `cluster-c`.
`cluster_primary_host`	The unqualified host name of the node that hosts the puppet master. The Mammoth utility was deployed from this host, and any reconfiguration of the cluster must be done while logged in to that node.
`cluster_type`	The type of cluster (default: `Hadoop`).
`cluster_version`	The software version installed on this cluster by the Mammoth utility, such as 3.1.0.

The following table describes describes the cluster parameters related to Oracle Big Data Connectors for `bdacli getinfo`.

Oracle Big Data Connectors Status Parameters

Parameter	Returns
`cluster_bdc_installed`	true if Oracle Big Data Connectors is installed; false otherwise
`cluster_odi_enabled`	true if the Oracle Data Integrator agent is enabled; false otherwise.
`cluster_odi_version`	The version of Oracle Data Integrator agent installed on the cluster.
`cluster_oraah_version`	The version of Oracle R Advanced Analytics for Hadoop installed on the cluster
`cluster_oraloader_version`	The version of Oracle Loader for Hadoop installed on the cluster
`cluster_osch_version`	The version of Oracle SQL Connector for HDFS installed on the cluster
`cluster_oxh_version`	The version of Oracle XQuery for Hadoop installed on the cluster

The following table describes the cluster network parameters for `bdacli getinfo`.

Cluster Network Parameters

Parameter	Returns
`cluster_hosts_entries`	A list of `/etc/hosts` entries that you can append to the `/etc/hosts` file on any device on the same InfiniBand fabric as the Oracle Big Data Cloud Service cluster, to ensure that Hadoop traffic uses the InfiniBand network. Do not add these entries to a device on a different fabric. Each entry is on a separate line with three parts: the InfiniBand IP address, the full client host name, and the short client host name.
`cluster_ilom_ips`	An ordered list of IP addresses for the Oracle ILOMs in the nodes, starting with the first node in the cluster
`cluster_ilom_names`	A list of unqualified host names on the administrative network for the Oracle ILOMs in the nodes, in order starting with the first node in the cluster
`cluster_node_ips`	The IP addresses on the client network of all nodes in this cluster
`cluster_node_names`	The host names on the client network of all nodes in the cluster, such as `bda1node01`

The following table describes the cluster security parameters for `bdacli getinfo`.

Cluster Security Parameters

Parameter	Returns
`cluster_av_admin`	The name of the Oracle Audit Vault and Database Firewall administration user. Returns an error if Audit Vault is not configured for this cluster.
`cluster_av_enabled`	`true` if Oracle Audit Vault and Database Firewall auditing is enabled; `false` otherwise
`cluster_av_port`	The port number that the Audit Vault node listens on. Returns an error if Oracle Audit Vault and Database Firewall is not configured on this cluster.
`cluster_av_server`	The IP address of the Audit Vault node. Returns an error if Oracle Audit Vault and Database Firewall is not configured on this cluster.
`cluster_av_service`	The database service name for the Audit Vault node. Returns an error if Oracle Audit Vault and Database Firewall is not configured on this cluster.
`cluster_hdfs_encryption_enabled`	`true` if network encryption of Hadoop data is enabled for this cluster; false otherwise
`cluster_hdfs_transparent_encryption_enabled`	`true` if HDFS Transparent Encryption of Hadoop data at rest is enabled for this cluster; `false` otherwise

Parameter	Returns
`cluster_kerberos_enabled`	`true` if Kerberos security is enabled; `false` otherwise.
`cluster_kerberos_kdc_hosts`	A list of key distribution center (KDC) hosts external to Oracle Big Data Appliance. Returns an error if Kerberos is not enabled.
`cluster_kerberos_kdc_on_bda`	`true` if the Kerberos KDC is on Oracle Big Data Appliance; `false` otherwise. Returns an error if Kerberos is not enabled.
`cluster_kerberos_realm`	The Kerberos realm for the cluster. Returns an error if Kerberos is not enabled.
`cluster_sentry_enabled`	`true` if Sentry is configured on the cluster; `false` otherwise.

Node Parameters

The following table describes the node parameters for `bdacli getinfo`.

Parameter	Returns
`server_mammoth_installed`	`true` if the Mammoth utility has deployed the Oracle Big Data Appliance software on this node; `false` otherwise.
`server_name`	The name of this node on the client network, such as bda1node01.
`server_os_version`	The version of Oracle Linux on this node, such as 6.4.

One-Off Patch Parameters

The following table describes the one-off patch parameters for `bdacli getinfo`.

Parameter	Returns
`available_patches`	A list of valid patches available for installation. A valid patch has a directory under `/opt/oracle/bda/patches` or `/opt/oracle/BDAMammoth/patches` that contains a file named `inventory`.
`installed_patches`	A list of patches already installed. An installed patch has a directory in both `/opt/oracle/bda/patches` and `/opt/oracle/BDAMammoth/patches`.

Examples

The following commands provide information about the optional software on the cluster:

```
# bdacli getinfo cluster_bdc_installed
true
# bdacli getinfo cluster_odi_version
```

```
11.1.1.7.0

# bdacli getinfo cluster_hdfs_transparent_encryption_enabled
true
```

The following command lists all switches on the current InfiniBand fabric. In this example, three Oracle Big Data Cloud Service racks are on the fabric with the standard hardware configuration of one spine switch and two gateway switches each.

```
$ bdacli getinfo ib_switches
bda1sw-iba0 00:21:28:6c:c8:af:a0:a0 36P
bda1sw-ibb0 00:21:28:46:9e:3b:a0:a0 36P
bda1sw-ibs0 00:21:28:6c:c8:ae:a0:a0 36P
bda2sw-ib1 00:21:28:46:98:d3:a0:a0 36P
bda2sw-ib2 00:21:28:de:ae:4a:c0:a0 GTW
bda2sw-ib3 00:21:28:c3:70:9a:c0:a0 GTW
bda3sw-ib1 00:21:28:46:90:ee:a0:a0 36P
bda3sw-ib2 00:21:28:df:34:8a:c0:a0 GTW
bda3sw-ib3 00:21:28:df:0f:0a:c0:a0 GTW
bda4sw-ib1 00:21:28:e8:af:23:a0:a0 36P
bda4sw-ib2 00:10:e0:0c:48:a0:c0:a0 GTW
bda4sw-ib3 00:21:28:f4:82:ce:c0:a0 GTW
```

This example installs patch 1234:

```
$ bdacli add patch 1234
```

Using dcli to Execute Commands Across a Cluster

The dcli utility executes commands across a group of nodes on Oracle Big Data Cloud Service and returns the output.

This chapter contains the following sections:

Overview of the dcli Utility

The `dcli` utility executes commands across a group of nodes in an Oracle Big Data Cloud Service cluster and returns the output. You use `dcli` to reinstall or reconfigure software on a cluster.

Basic Use of dcli

Getting Help

To see the `dcli` help page, enter the `dcli` command with the `-h` or `--help` options. You can see a description of the commands by entering the `dcli` command with no options.

Identifying the Target Nodes

You can identify the nodes where you want the commands to run either in the command line or in a file. For a list of default target nodes, use the -t option. To change the target nodes for the current command, use the `-c` or `-g` options described in the table below..

You can manually create files with groups of nodes to manage together. For example, you might manage nodes 5 to 18 together, because they have no special functions like nodes 1 to 4.

Specifying the Commands

You typically specify a command for execution on the target nodes on the command line. However, you can also create a command file for a series of commands that you often use together or for commands with complex syntax. See the `-x` option in the table below.

You can also copy files to the target nodes without executing them by using the `-f` option.

Controlling the Output Levels

You can request more information with the `-v` option or less information with the `-n` option. You can also limit the number of returned lines with the `--maxlines` option, or replace matching strings with the `-r` option.

Following are examples of various output levels using a simple example: the Linux date command.

> **Note:**
>
> The output from only one node (node07) is shown. The syntax in these examples executes the `date` command on all nodes.

This is the default output, which lists the node followed by the date.

```
# dcli date
bda1node07-adm.example.com: Tue Feb 14 10:22:31 PST 2016
```

The minimal output returns `OK` for completed execution:

```
# dcli -n date
OK: ['bda1node07.example.com']
```

Verbose output provides extensive information about the settings under which the command ran:

```
dcli -v dateoptions.nodes: Noneoptions.destfile: Noneoptions.file:
Noneoptions.group: dcserversoptions.maxLines: 100000options.listNegatives:
```

```
Falseoptions.pushKey: Falseoptions.regexp: Noneoptions.sshOptions:
Noneoptions.scpOptions: Noneoptions.dropKey: Falseoptions.serializeOps:
Falseoptions.userID: rootoptions.verbosity 1options.vmstatOps
Noneoptions.execfile: Noneargv: ['/opt/oracle/bda/bin/dcli', '-g',
'dcservers', '-v', 'date']Success connecting to nodes:
['bda1node07.example.com']...entering thread for
bda1node07.example.com:execute: /usr/bin/ssh -l root
bda1node07.example.com ' date' ...exiting thread for
bda1node07.example.com status: 0bda1node07.example.com: Tue Feb 14
10:24:43 PST 2012]
```

dcli Syntax

```
dcli [option] [command]
```

Parameters

`option`
An option described in the table below. You can omit all options to run a command on all nodes in the cluster.

`command`
Any command that runs from the operating system prompt. If the command contains punctuation marks or special characters, then enclose the command in double quotation marks.
The backslash (\) is the escape character. Precede the following special characters with a backslash on the command line to prevent interpretation by the shell. The backslash is not needed in a command file. See the `-x` option for information about command files.

- $ (dollar sign)
- ' (quotation mark)
- < (less than)
- > (greater than)
- ( ) (parentheses)

dcli Options

Option	Description
`-c` *`nodes`*	Specifies a comma-separated list of Oracle Big Data Cloud Service nodes where the command is executed
`-C`	Uses the list of nodes in `/opt/oracle/bda/cluster-rack-infiniband` as the target. See Identifying the Target Nodes.
`-d` *`destfile`*	Specifies a target directory or file name for the `-f` option
`-f` *`file`*	Specifies files to be copied to the user's home directory on the target nodes. The files are not executed. See the `-l` option.

Option	Description
`-g groupfile`	Specifies a file containing a list of Oracle Big Data Cloud Service nodes where the command is executed. You can use either node names or IP addresses in the file.
`-h, --help`	Displays a description of the commands
`-k`	Pushes the `ssh` key to each node's `/root/.ssh/authorized_keys` file.
`-l userid`	Identifies the user ID for logging in to another node. The default ID is `root`.
`--maxlines=maxlines`	Identifies the maximum lines of output displayed from a command executed on multiple nodes. The default is 10,000 lines.
`-n`	Abbreviates the output for non-error messages. Only the node name is displayed when a node returns normal output (return code `0`). You cannot use the `-n` and `-r` options together.
`-r regexp`	Replaces the output with the node name for lines that match the specified regular expression
`-s sshoptions`	Specifies a string of options that are passed to SSH
`--scp=scpoptions`	Specifies a string of options that are passed to Secure Copy (SCP), when these options are different from *`sshoptions`*
`--serial`	Serializes execution over the nodes. The default is parallel execution.
`-t`	Lists the target nodes
`--unkey`	Drops the keys from the `authorized_key` files of the target nodes
`-v`	Displays the verbose version of all messages
`--version`	Displays the `dcli` version number
`--vmstat=VMSTATOPS`	Displays the syntax of the Linux Virtual Memory Statistics utility (`vmstat`). This command returns process, virtual memory, disk, trap, and CPU activity information. To enter a `vmstat` command, enclose its options in quotation marks. For example: `--vmstat="-a 3 5"` See your Linux documentation for more information about `vmstat`.
`-x execfile`	Specifies a command file to be copied to the user's home directory and executed on the target nodes. See the `-l` option.

dcli Return Values

- `0`: The command ran successfully on all nodes.
- `1`: One or more nodes were inaccessible or remote execution returned a nonzero value. A message lists the unresponsive nodes. Execution continues on the other nodes.
- `2`: A local error prevented the command from executing.

If you interrupt the local dcli process, then the remote commands may continue without returning their output or status.

dcli Examples

Following are examples of the dcli utility.

This example returns the default list of target nodes:

```
# dcli -t
Target nodes: ['bda1node01-adm.example.com', 'bda1node02-adm.example.com',
'bda1node03-adm.example.com', 'bda1node04-adm.example.com', 'bda1node05-
adm.example.com', 'bda1node06-adm.example.com', 'bda1node07-
adm.example.com', 'bda1node08-adm.example.com', 'bda1node09-
adm.example.com']
```

The next example checks the temperature of all nodes:

```
# dcli 'ipmitool sunoem cli "show /SYS/T_AMB" | grep value'

bda1node01-adm.example.com: value = 22.000 degree C
bda1node02-adm.example.com: value = 22.000 degree C
bda1node03-adm.example.com: value = 22.000 degree C
bda1node04-adm.example.com: value = 23.000 degree C
          .
          .
          .
```

4

Access Your Oracle Big Data Cloud Service

This section describes how to access clusters and the tools, utilities and interfaces available in a cluster.

> **Note:**
>
> By default, the port used for accessing the cluster through SSH, port 22, is open and other ports used for accessing other services are closed. You can control access to the ports by configuring the network whitelist (for clusters hosted in *regions*) or by configuring the firewall (for cluster hosted in *availability domains*). The configuration controls whether network requests from specified IP addresses (or ranges of addresses) will be accepted or denied at specified ports. See Control Network Access to Services.

Topics

Connect to a Cluster Node Through Secure Shell (SSH)

To gain local access to the tools, utilities and other resources on anOracle Big Data Cloud Service cluster node, use Secure Shell (SSH) client software to establish a secure connection and log in.

> **Note:**
>
> By default, the port used for accessing the cluster through SSH, port 22, is open. You can control access to that and other ports by configuring the network whitelist (for clusters hosted in regions) or by configuring the firewall (for cluster hosted in availability domains). The configuration controls whether network requests from specified IP addresses (or ranges of addresses) will be accepted or denied at specified ports. See Control Network Access to Services.

Several SSH clients are freely available. The following sections show how to use SSH clients on UNIX, UNIX-like, and Windows platforms to connect to a node.

The following instructions describe how to connect as the `opc` user and then use the `sudo` command to open a `root` shell. After you do this the first time, you may choose instead to connect using a password by assigning a known password to `opc` or creating another user with a known password.

Connecting to a Node By Using PuTTY on Windows

PuTTY is a freely available SSH client program for Windows.

Before You Begin

Before you use the PuTTY program to connect to a node, you need the following:

- The IP address of the node

  The IP address of the node is listed on the Cluster Details page for the cluster containing the node. To display this page, see View Details About a Cluster.

- The SSH private key file that pairs with the public key associated with the cluster

  The public key was associated with your cluster when it was created. See Create a Cluster. If you don't have the private key that's paired with the public key, contact your administrator.

  The private key file must of the PuTTY `.ppk` format. If the private key file was originally created on the Linux platform, you can use the PuTTYgen program to convert it to the `.ppk` format.

  For instructions on creating an SSH key pair, see Generating a Secure Shell (SSH) Public/Private Key Pair.

Procedure

To connect to a node using the PuTTY program on Windows:

1. Download and install PuTTY.

   To download PuTTY, go to `http://www.putty.org/` and click the **You can download PuTTY here** link.

2. Run the PuTTY program.

   The PuTTY Configuration window is displayed, showing the Session panel.

3. In **Host Name (or IP address)** box, enter the IP address of the node.

4. Confirm that the **Connection type** option is set to **SSH**.

5. In the Category tree, expand **Connection** if necessary and then click **Data**.

   The Data panel is displayed.

6. In the **Auto-login username** box, enter **opc**. As the `opc` user, you can use the `sudo` command to gain root access to the node, as described in the last step, below.

7. Confirm that the **When username is not specified** option is set to **Prompt**.

8. In the Category tree, expand **SSH** and then click **Auth**.

   The **Auth** panel is displayed.

9. Click the **Browse** button next to the **Private key file for authentication** box. Then, in the **Select private key file** window, navigate to and open the private key file that matches the public key that is associated with the cluster.
10. In the **Category** tree, click **Session**.

    The **Session** panel is displayed.
11. In the **Saved Sessions** box, enter a name for this connection configuration. Then, click **Save**.
12. Click **Open** to open the connection.

    The PuTTY Configuration window is closed and the PuTTY window is displayed.

    If this is the first time you are connecting to the VM, the PuTTY **Security Alert** window is displayed, prompting you to confirm the public key. Click **Yes** to continue connecting.
13. To perform operations that require root access to the node—such as issuing `bda–ossadmin` commands—open a root command shell. Enter `sudo -s` at the command prompt:

    ```
    $ sudo -s
    # whoami
    # root
    ```

Connecting to a Node By Using SSH on UNIX

UNIX and UNIX-like platforms (including Solaris and Linux) include the `ssh` utility, an SSH client.

Before You Begin

Before you use the `ssh` utility to connect to a node, you need the following:

- The IP address of the node

  The IP address of the node is listed on the Cluster Details page of the cluster containing the node. To display this page, see View Details About a Cluster.
- The SSH private key file that pairs with the public key associated with the cluster

  The public key was associated with your cluster when it was created. See Create a Cluster. If you don't have the private key that's paired with the public key, contact your administrator.

Procedure

To connect to a node using the ssh utility on UNIX and UNIX-like platforms:

1. In a command shell, set the file permissions of the private key file so that only you have access to it:

   ```
   $ chmod 600 private-key-file
   ```

   private-key-file is the path to the SSH private key file that matches the public key that is associated with the cluster.
2. Run the ssh utility:

   ```
   $ ssh -i private-key-file opc@node-ip-address
   ```

where:

- ***private-key-file*** is the path to the SSH private key file.
- **opc** is the `opc` operating system user. As `opc`, you can use the `sudo` command to gain root access to the node, as described in the next step.
- ***node-ip-address*** is the IP address of the node in *x.x.x.x* format.

If this is the first time you are connecting to the node, the ssh utility prompts you to confirm the public key. In response to the prompt, enter **yes**.

3. To perform operations that require root access to the node—such as issuing `bda-oss-admin` commands—open a root command shell. Enter `sudo -s` at the command prompt:

```
$ sudo -s
# whoami
# root
```

Open the Oracle Big Data Cloud Service Console

Use the Oracle Big Data Cloud Service Services page to create and review Hadoop clusters.

1. Sign in to your Cloud Account.

   See Signing in to Your Cloud Account in *Getting Started with Oracle Cloud*.

2. Click the navigation menu in the top corner, then find and click **Big Data Cloud Service**.
3. If a Welcome page is displayed, go to the Instances page by clicking the**Instances** tab near the top of the page.

Access Cloudera Manager to Work with Hadoop Data and Services

You can access Cloudera Manager from the Oracle Big Data Cloud Service console, or you can access it directly from a browser.

Open Cloudera Manager from the Oracle Big Data Cloud Service Console

1. Open the Oracle Big Data Cloud Service console. See Opening the Oracle Big Data Cloud Service Console.
2. Click the icon on the row of the cluster you want to manage, and select **Open Cloudera Manager**.

   The Cloudera Manager application is displayed.

Open Cloudera Manager from a Web Browser

> **Note:**
>
> By default, the port used for accessing Cloudera Manager, port 7183, is blocked. To control access to that port, you must configure the network whitelist for the service. That configuration controls whether network requests from specified IP addresses (or ranges of addresses) will be accepted or denied at that port. See Control Network Access to Services.

To open Cloudera Manager from a web browser:

1. Open the Oracle Big Data Cloud Service console. See Opening the Oracle Big Data Cloud Service Console.
2. Click the icon on the row of the cluster you want to manage, and select **View Details**.

   The Cluster Overview page is displayed.
3. Find the URL for Cloudera Manager listed at the top of the page.
4. Open a browser and navigate to that URL.
5. Enter your credentials for logging in, as prompted.

Access Cloudera Hue to Manage Hadoop Data and Resources

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

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

Hue is automatically installed and configured on Oracle Big Data Cloud Service clusters. Hue runs on port 8888 of the ResourceManager node (node03).

> **Note:**
>
> By default, port 8888 is blocked. To control access to that port, you must configure the network whitelist for the service. That configuration controls whether network requests from specified IP addresses (or ranges of addresses) will be accepted or denied at that port. See Control Network Access to Services.

To use Hue:

1. Log in to Cloudera Manager and click the **hue** service on the Home page.
2. On the hue page under Quick Links, click `Hue Web UI`.
3. Bookmark the Hue URL, so that you can open Hue directly in your browser. The following URL is an example:

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

4. Log in with your Hue credentials.

   If Hue accounts haven't been created yet, you can log into the default Hue administrator account, by using the following credentials:

   - **Username**: `admin`
   - **Password:** *`cm-admin-password`*

   where *`cm-admin-password`* is the password that was specified when the cluster for the Cloudera Manager `admin` user was activated. You can then create other user and administrator accounts.

> **See Also:**
>
> *`Hue User Guide`*.

5

Copy Data With Oracle Big Data Cloud Service Tools

Oracle Big Data Cloud Service provides a number of tools and features to facilitate data management:

You can use one or a combination of the following to fit your desired workflow.

- Oracle Distributed Copy (odcp)
- Oracle Distributed Diff (odiff)
- Big Data Management Command Line Utility (bda-oss-admin)
- Oracle Big Data Manager
- Oracle Big Data Manager Command Line Interface (bdm-cli)
- Oracle Big Data Manager SDKs

Oracle Distributed Copy (odcp)

`odcp` is a distributed command line interface (CLI) for copying data sets to and from various storage providers:

- Oracle Cloud Infrastructure Object Storage Classic (formerly known as Oracle Storage Cloud Service)
- Hadoop Distributed File System (HDFS)
- Amazon Simple Storage Service (S3)
- WebHDFS and Secure WebHDFS (SWebHDF)
- Oracle Cloud Infrastructure Object Storage (formerly known as Oracle Bare Metal Cloud Object Storage Service)
- Hypertext Transfer Protocol (HTTP) and HTTP Secure (HTTPS) — Used for sources only.

`odcp` is a Spark application that can read its configuration from the command line or from the `core-site.xml` file on the cluster. The configuration can include the storage provider URL, user credentials, login certificate, and provider-specific configuration details.

Use `bda-oss-admin` commands to update the configuration in the `core-site.xml` file (see below).

See Use the odcp Command Line Utility to Copy Data

Big Data Management Command Line Utility (bda-oss-admin)

Use the `bda-oss-admin` command line utility to add and update storage providers' configurations. The configurations are saved in the `core-site.xml` file on the cluster. With bda-oss-admin, you can perform actions such as add, list, remove, and update credentials.

See Use bda-oss-admin to Manage Storage Resources.

Oracle Distributed Diff (odiff)

`odiff` is a command line utility for comparing large data sets stored in HDFS and Oracle Cloud Infrastructure Object Storage Classic. The computation runs as a distributed Spark application.

See Use odiff to Compare Large Data Sets.

Oracle Big Data Manager

The Oracle Big Data Manager web application provides visual tools for creating and managing data sources, creating and scheduling data transfer jobs, displaying logs, and a performing a number of other data transfer tasks.

Oracle Big Data Manager uses `odiff` and `odcp` for data management tasks.

See About Oracle Big Data Manager.

Oracle Big Data Manager Command Line Interface (bdm-cli)

`bdm-cli` is command line utility for creating and managing data sources, creating and scheduling data transfer jobs, displaying logs, and a performing a number of other data transfer tasks.

You can install `bdm-cli` in a remote operating system. That means you can create and schedule data transfer jobs from any remote server. You don't have to use SSH to connect to a cluster node to execute any of these commands (although you can).

See Oracle Big Data Manager Command Line Interface (bdm-cli)

Oracle Big Data Manager SDKs

You can use the Oracle Big Data Manager SDKs to use Oracle Big Data Manager from within applications. See Manage Data and Copy Jobs With the Oracle Big Data Manager SDKs.

6

Use bda-oss-admin to Manage Storage Resources

Use the `bda-oss-admin` command line utility to add and manage storage providers, user credentials for storage access, and other resources for use with an Oracle Big Data Cloud Service cluster. Configuration details are stored in the `core.site.xml` file on the cluster.

Topics

Register Storage Providers with Your Cluster

Use `bda-oss-admin` commands to configure command shell access to your Hadoop cluster and access to storage providers. You can then use `odcp` and `bdm-cli` commands to copy data between HDFS and storage.

Topics

Set bda-oss-admin Environment Variables

Some `bda-oss-admin` options can be set as environment variables so you don't have to specify the values every time you run the commands.

The following tables show the environment variables that correspond to (and can be substituted for) those `bda-oss-admin` options.

Options and Environment Variables for All bda-oss-admin Commands

The following values must be set for all `bda-oss-admin` commands, either on the command line as options to the command or as shell environment variables.

Command Option	Environment Variable	Description
`--cm-admin`	`CM_ADMIN`	Cloudera Manager administrator user name.
`--cm-passwd`	`CM_PASSWORD`	Cloudera Manager administrator password. The command will prompt for the password if it is not provided.
`--cm-url`	`CM_URL`	Cloudera Manager URL (e.g. `https://servername.bigdata.examplecloud.com:7183`)

Options and Environment Variables for bda-oss-admin Storage Credentials Commands

The `bda-oss-admin` storage credentials commands that require some or all of the following options are:

- `bda-oss-admin add_swift_cred`
- `bda-oss-admin change_swift_cred_passwd`
- `bda-oss-admin delete_swift_cred`

See Register Storage Credentials with the Cluster for more complete descriptions of the options.

Command Option	Environment Variable	Description
`--swift-username`	`ST_USER`	Oracle Cloud Infrastructure Object Storage Classic account administrator's user name in the form *`Service-IdentityDomain:username`*. User names must conform to the rules described in Naming Requirements.
`--swift-password`	`ST_KEY`	The administrator's password.
`--swift-storageurl`	`ST_AUTH`	The Oracle Cloud Infrastructure Object Storage Classic (Swift) authentication URL from which to obtain authentication tokens, for example, `https://storage.a123456.examplecloud.com/auth/v1.0`.
`--swift-provider`	`ST_PROVIDER`	A user-supplied name for the credentials. This name is provided when using `add_swift_cred` to add credentials. The name is used as an alias for the credentials in Hadoop. Provider names must conform to the rules described in Naming Requirements. **Note:** When you create a Hadoop instance by using the Create Instance wizard, a default provider name, BDCS, is created. If you do not provide a storage provider name using the `--swift-provider` *`name`* option, you can use BDCS whenever the storage provider name is needed; for example, when using a command like: `hdfs dfs –cat mycontainer.BDCS/file`

Setting Environment Variables

To set these as environment variables, you can create and run a shell script.

In the following example, a a Linux bash shell script named `bdcsvars.sh` sets the Cloudera Manager credentials required by all `bda-oss-admin` commands:

```
#!/bin/bash
export CM_ADMIN="my_CM_admin_username"
export CM_PASSWORD="my_CM_admin_password"
export CM_URL="https://my_CM_hostname_:7183"
```

In the following example, a Linux bash shell script named `storvars.sh` sets the storage credentials :

```
#!/bin/bash
export ST_USER="MyServiceName-MyIdentityDomain:MyUserName"
export ST_KEY="Wel_123"
export ST_AUTH="http://storage.a123456.examplecloud.com/auth/v1.0"
export ST_PROVIDER="MyProviderName"
```

If you are working with multiple storage providers, it may be convenient to create shell scripts for all of them. Then you only have to run a script to set storage credentials for whichever provider you are using.

To run the above scripts from the directory in which they reside:

```
# source ./bdacsvars.sh
# source ./storvars.sh
```

Reviewing the Configuration

Configurations that you set with the `bda-oss-admin` commands, parameters, and environment variables are stored in the Hadoop `/etc/hadoop/conf/core-site.xml` configuration file. For example:

```
<configuration>
...
  <property>
    <name>fs.swift.service.storageservice3991-bdaoss.username</name>
    <value>john.smith@example.com</value>
  </property>
  <property>
    <name>fs.swift.service.storageservice3991-bdaoss.tenant</name>
    <value>storageservice3991-bdaoss</value>
  </property>
  <property>
    <name>fs.swift.service.storageservice3991-bdaoss.password</name>
    <value>A_password</value>
  </property>
  <property>
    <name>fs.swift.service.storageservice3991-bdaoss.auth.url</name>
    <value>https://storageservice3991-bdaoss.storage.examplecloud.com/auth/
```

```
v2.0/tokens</value>
  </property>
  <property>
    <name>fs.swift.service.storageservice3991-bdaoss.public</name>
    <value>true</value>
  </property>
</configuration>
```

You can look in the `core-site.xml` file to see the current configuration, but you shouldn't edit it directly. Use `bda-oss-admin` instead.

Register Storage Credentials with the Cluster

When you create a cluster, you can register with it the credentials of a specific user of a specific instance of Oracle Cloud Infrastructure Object Storage Classic (formerly known as Oracle Storage Cloud Service). That allows that user to copy data to and from that storage instance, without having to reestablish credentials. After the cluster is created, you can register other users and other storage service instances, so those users can copy data to and from those storage service instances.

Prerequisite

You must have access to an Oracle Cloud Infrastructure Object Storage Classic instance.

About Storage Credentials

When you create a cluster using the Create Cluster wizard and you choose to associate it with an Oracle Cloud Infrastructure Object Storage Classic instance, you're prompted for a user name and a password. (See Create a Cluster.) The user name identifies the storage service instance, its identity domain, and a user with administrator rights to the instance. When the cluster is created, a default storage service *provider* , named *BDCS*, is also created for that storage server instance. The provider is an alias for all the credentials required for using the storage service instance. Those credentials are: the user name (service instance name, administrrator user name, and administrrator user's password) and the URL for a server that provides authentication tokens for accessing the service. See Storage Credential Details, below.

If this is the only storage service instance that will be used with the Hadoop cluster, you don't have to register any other credentials. However, if you need to access other storage service instances and if you want to register other credentials for them, you can do so by using `bda-oss-admin` commands. You set the values for the credentials either by entering them as parameters to `bda-oss-admin` commands. See Using odcp to Copy Data.

Storage Credential Details

The components of the credentials are described in the following table.

> **Note:**
>
> Oracle Cloud Infrastructure Object Storage Classic is based on *Swift*, the open-source OpenStack Object Store project. As a consequence, the bda-oss-admin command line utility contains some subcommands and options that say “swift.”

Storage Credentials, Syntax, and Usage	Description

STORAGE USER NAME
Option Syntax
`--swift-username` *`Service-IdentityDomain:username`*`[.role]`
Environment Variable
`ST_USER`
Required for
`add_swift_cred`
	Oracle Cloud Infrastructure Object Storage Classic account administrator’s user name in the form *`Service-IdentityDomain:username`*, where
	• *`Service`*
	is the customer-supplied name of the Oracle Cloud Infrastructure Object Storage Classic instance. This name was given when the storage service was provisioned.
	• *`IdentityDomain`*
	is the identity domain in which the service instance is provisioned. This name was given when the storage service was provisioned. (In Swift, this is called the tenant ID.)
	• *`username`*
	is the user name for a user with administrative privileges on the account.
	• *`role`*
	is a role defined in an Access Control List (ACL) assigned to a container. See Setting Container Metadata in *Using Oracle Cloud Infrastructure Object Storage Classic* for more information. If a role is not defined, this option is not required.
	You can find the storage details described above on the Service Details page of your Oracle Cloud Infrastructure Object Storage Classic account. See Accessing Oracle Cloud Infrastructure Object Storage Classic in *Getting Started with Oracle Cloud*.
	User names must conform to the rules described below in Naming Requirements.
	Example:
	`WestRegion-nloracle12345:pstuyves`

STORAGE PASSWORD
Option Syntax
`--swift-password` *`password`*
Environment Variable
`ST_KEY`
Required for
`add_swift_cred`
`change_swift_cred_passwd`
	Password associated with *`username`*, described above..
	If the password is Base64–encoded, use the `-b` flag, for example, `--swift-password` *`AnEncodedPassword`* `–b`.

Storage Credentials, Syntax, and Usage	Description

STORAGE LOCATION
Option Syntax
`--swift-storageurl` *`url`*
Environment Variable
`ST_AUTH`
Required for
`add_swift_cred`
	The Oracle Cloud Infrastructure Object Storage Classic (Swift) \ authentication URL for your Oracle Cloud Infrastructure Object Storage Classic data center/region, for example, `https://storage.us2.oraclecloud.com/auth/v1.0`. For an explanation of how those URLs are constructed, see Authenticating Access When Using REST API.

STORAGE PROVIDER NAME
Option Syntax
`--swift-provider` *`name`*
Environment Variable
`ST_PROVIDER`
Required for
`add_swift_cred`
`change_swift_cred_passwd`
`delete_swift_cred`
	A user-supplied name for the credentials. This name is provided when using `add_swift_cred` to add credentials. The name is used as an alias for the credentials in Hadoop.
	Provider names must conform to the rules described below in Naming Requirements.
	Note: When you create a Hadoop cluster by using the Create Cluster wizard, a default provider name, `BDCS`, is created. If you don't provide a storage provider name using the `--swift-provider` *`name`* option, you can use `BDCS` whenever the storage provider name is needed; for example, when using a command like:
	`hdfs dfs –cat mycontainer.BDCS/file`

Naming Requirements

Provider names and container names must conform to RFC952 DOD Internet Host Table Specification, for example:

- A name is a text string of 2 to 24 characters and can contain only:
  - Letters *A-Z* and *a-z* (case is not considered)
  - Digits *0–9*
  - Hyphen (-)
- The first character must be an alpha character.
- The last character cannot be a period or hyphen.
- Spaces are not allowed.
- Periods are allowed only to delimit components of domain style names.

bda-oss-admin Command Reference

Use the Oracle Big Data Cloud Service command line utility `bda-oss-admin` to manage users and resources of your cluster.

To issue `bda-oss-admin` commands, you must connect to a node as the `opc` user and then use the `sudo` command to switch to the `root` user. See Connect to a Cluster Node Through Secure Shell (SSH).

Syntax

```
bda-oss-admin [options] subcommand [arguments]...
```

Options

Option	Description
`--version`	Show the `bda-oss-admin` version
`--cm-admin` *user_name*	Cloudera Manager administrator user name
`--cm-passwd` *password*	Cloudera Manager administrator password. The command will prompt for the password if it is not provided.
`--b64–cm-passwd` *password*	The Cloudera Manager password is Base64 encoded. It will be decoded before uploading.
`--cm-url` *url*	Cloudera Manager URL; for example, `https://servername.bigdata.oraclecloud.com:7183`
`-b` `--b64-encoded-pwds`	The password is Base64 encoded. It will be decoded before uploading.
`-h` `--help`	Show help for this command.

When you specify any of the above options on the command line, the options must be placed immediately after the `bda-oss-admin` command and before any of its subcommands. For example, this command is legal:

```
# bda-oss-admin --cm-passwd password list_swift_creds
```

However, the following command is not legal, because the `--cm-passwd` option is placed after the `list_swift_creds` subcommand:

```
# bda-oss-admin list_swift_creds --cm-passwd password
```

Environment Variables

Instead of setting Cloudera Manager credentials as options on the command line, you can set them as environment variables.

This environment variable...	Is equivalent to this option...
`CM_ADMIN`	`--cm-admin`
`CM_PASSWORD`	`--cm-passwd`
`CM_URL`	`--cm-url`

See also Setting bda-oss-admin Environment Variables.

Subcommands

Command	Task
add_bdcs_cp_extensions_mr	Add Oracle Big Data Cloud Service `classpath` extensions to the MapReduce configuration.
add_swift_cred	Add credentials for an Oracle Cloud Infrastructure Object Storage Classic user with administrative privileges.
change_swift_cred_passwd	Change the Oracle Cloud Storage password for a provider stored in Hadoop.
delete_swift_cred	Delete Oracle Cloud Infrastructure Object Storage Classic credentials.
export swift_creds	Write Oracle Cloud Infrastructure Object Storage Classic credentials to a file in JSON format.
import_swift_creds	Add one or more Oracle Cloud Infrastructure Object Storage Classic credentials from a JSON file.
list_swift_creds	List Oracle Cloud Infrastructure Object Storage Classic (Swift) credentials.
print_yarn_mapred_cp	Display the YARN MapReduce default classpath.
remove_bdcs_cp_extensions_mr	Remove Oracle Big Data Cloud Service `classpath` extensions from the MapReduce configuration.
restart_cluster	Restart the cluster (only stale services).

bda-oss-admin add_bdcs_cp_extensions_mr

Adds the Oracle Big Data Cloud Service `classpath` extensions to the MapReduce configuration file, `mapred-site.xml`.

Syntax

```
bda-oss-admin add_bdcs_cp_extensions_mr [options]
```

Options

Option	Description
`-h` `--help`	Show help for this command

Example

```
# bda-oss-admin add_bdcs_cp_extensions_mr
Changes will not affect the cluster until services get restarted. See the
restart_cluster command
```

bda-oss-admin add_swift_cred

Add a single Oracle Big Data Cloud Service credential.

Syntax

```
bda-oss-admin add_swift_cred options
```

Options

Option	Description

`--swift-username` *`Service-IdentityDomain:username`*	Oracle Storage Cloud Service account administrator's user name in the form *`Service-IdentityDomain:UserName`*, where
	• *`Service`* is the customer-supplied name of the Oracle Storage Cloud Service instance. This name was given when the storage service was provisioned.
	• *`IdentityDomain`* is the identity domain in which the service instance is provisioned. This name was given when the storage service was provisioned. (In Swift, this is called the tenant ID.)
	• *`UserName`* is the user name for a user with administrative privileges on the account. Conventionally, it is an e-mail address, but it can be any name.
	User names must conform to the rules described in Naming Requirements.
	You can find the storage details described above on the Service Details page of your Oracle Storage Cloud Service account.
	Examples:
	`MyServiceName-MyIdenityDomain:MyUserName`
	`bigdatastorage1-ProdDomain:my.name@example.com`

`--swift-password` *`password`*	Password associated with *`username`*, described above.. If the password is Base64–encoded, use the `-b`flag, for example, `--swift-password` *`AnEncodedPassword`* –b.
`--swift-storageurl` *`url`*	The Oracle Storage Cloud Service (Swift) \ authentication URL from which to obtain authentication tokens, for example, `https://storage.us2.oraclecloud.com/auth/v1.0`.

Option	Description
`--swift-provider` *provider_name*	Give a name for the credentials being added, such as `bigdatastorage1`. The name is used as an alias for the credentials in Hadoop. Provider names must conform to the rules described in Naming Requirements. **Note:** If you don't specify a storage provider name, the name "BDCS" is used by default. For more information about the provider name, see Register Storage Credentials with the Cluster.
`-N` `--no-verify`	Do not verify accounts against actual storage service before adding.
`-h` `--help`	Show help for this command.

Environment Variables

Instead of providing storage credentials as options on the command line, you can set them as environment variables.

This environment variable...	Is equivalent to this option...
`ST_USER`	`--swift-username`
`ST_KEY`	`--swift-password`
`ST_AUTH`	`--swift-storageurl`
`ST_PROVIDER`	`--swift-provider`

Example

```
# bda-oss-admin add_swift_cred --swift-username bigdatastorage1-
ProdDomain:my.name@example.com --swift-password Welcome_1 --swift-
storageurl https://storage.us2.oraclecloud.com/auth/v1.0 --swift-provider
bigdatastorage1
#
```

See Also

- Register Storage Credentials with the Cluster
- Set bda-oss-admin Environment Variables

bda-oss-admin change_swift_cred_passwd

Change an Oracle Storage Cloud service provider stored password.

Syntax

```
bda-oss-admin change_swift_cred_passwd [options]
```

Options

Option	Description
`--swift–password` *password*	New storage password. If it is Base64 encoded, use the –b flag, that is, `--swift–password –b` *password*. If `--swift-password` is not specified, the command will prompt for it.
`--swift–provider` *provider*	Name of the storage provider associated with password to change.
`-h` `--help`	Show help for this command.

Example

```
# bda-oss-admin change_swift_cred_passwd --swift-password Welcome_123
```

bda-oss-admin delete_swift_cred

Delete one or more Oracle Storage Cloud Service (Swift) credentials, by provider name.

Syntax

```
bda-oss-admin delete_swift_cred [options]
```

Options

Option	Description
`-p` `--swift–provider` *provider*	The name of the provider associated with the credentials to delete. The `provider` is the provider name (alias) defined when adding Swift credentials to the Hadoop configuration. See Register Storage Credentials with the Cluster and bda-oss-admin add_swift_cred. You can list several providers to delete mulitple credentials, for example, `bda-oss-admin delete_swift_cred --swift-provider bdcsscred1 --swift-provider bdcsscred2`.

Option	Description
`-h` `--help`	Show help for this command.

Example

```
# bda-oss-admin delete_swift_cred --swift-provider bdcsscred
#
```

bda-oss-admin export_swift_creds

Exports Oracle Storage Cloud Service credentials to a JSON-formatted file. Use a `-` (hyphen) in place of file name to display the JSON-formatted credentials on screen.

Usage

```
bda-oss-admin export_swift_creds [options] filename
```

Options

Option	Description
`-h` `--help`	Show help text for this command and exit

Example

```
# bda-oss-admin export_swift_creds -
[{"swift_password": "Welcome_123", "swift_username": "MyServiceName-
MyDomainName:MyUserName", "swift_provider": "mybdcss", "swift_storageurl":
"http://storage.us2.oraclecloud.com/auth/v1.0"}, {"swift_password":
"Welcome_1", "swift_username": "bigdatastorage1-
ProdDomain:my.name@example.com", "swift_provider": "ProdDom",
"swift_storageurl": "https://storage.us2.oraclecloud.com/auth/v1.0"}]
```

bda-oss-admin import_swift_creds

Import Oracle Storage Cloud Service credentials from a JSON-formatted file. Use `-` in place of filename to import using standard input (`stdin`).

Syntax

```
bda-oss-admin import_swift_creds [options] filename
```

Usage

The JSON schema is a list of one or more objects with Oracle Cloud Storage credentials::

```
    [
      {
          "swift_provider" : "bdcs",
          "swift_username" : "MyServiceName-MyDomainName:MyUserName",
          "swift_password" " "password",
          "swift_storageurl" : "https://storage.us2.examplecloud.com/auth/
v1.0"
      }
      ...
  ]
```

Options

Option	Description
`-N` `–no-verify`	Do not verify accounts against the actual storage service before importing
`-h` `--help`	Show help text for this command.

Example

```
# bda-oss-admin import_swift_creds /user/company/work/swift_creds
#
```

bda-oss-admin list_swift_creds

List the Oracle Storage Cloud Service (Swift) credentials stored in this Hadoop configuration.

Syntax

```
bda-oss-admin list_swift_creds [options]
```

Options

Option	Description
`-t` `--as-table`	List credentials in table format.
`-h` `--help`	Show help for this command.

Example

```
$ bda-oss-admin list_swift_creds -t

PROVIDER  USERNAME                                        STORAGE URL
test2     testtenant:testusername                         https://storage-
us2.oraclecloud.com/auth/v1.0
main2     Storage-bdcs:bdcs.Storageadmin                  https://
storage.us2.oraclecloud.com/auth/v1.0
main      MyServiceName-MyDomainName:MyUserName https://
storage.us2.oraclecloud.com/auth/v1.0
```

bda-oss-admin print_yarn_mapred_cp

Display the YARN MapReduce default classpath.

Syntax

```
bda-oss-admin print_yarn_mapred_cp [options]
```

Options

Option	Description
`-h` `--help`	Show help for this command.

Example

```
# bda-oss-admin print_yarn_mapred_cp

YARN Mapreduce Conf mapreduce_application_classpath: $HADOOP_MAPRED_HOME/
*,$HADOOP_MAPRED_HOME/lib/*,$MR2_CLASSPATH,/opt/oracle/orabalancer-2.4.0-
h2/jlib/orabalancer-2.4.0.jar,/opt/oracle/orabalancer-2.4.0-h2/jlib/
commons-math-2.2.jar,/opt/oracle/bda/bdcs/bdcs-rest-api-app/current/lib-
hadoop/*
```

bda-oss-admin remove_bdcs_cp_extensions_mr

Removes the Oracle Big Data Cloud Service `classpath` extensions from the MapReduce configuration file, `mapred-site.xml`.

Syntax

```
bda-oss-admin remove_bdcs_cp_extensions_mr [options]
```

Options

Option	Description
`-h` `--help`	Show help for this command

Example

```
# bda-oss-admin remove_bdcs_cp_extensions_mr
Changes will not affect the cluster until services get restarted. See the
restart_cluster command
```

bda-oss-admin restart_cluster

Restarts the Hadoop Cluster by interacting with Cloudera Manager.

Syntax

```
bda-oss-admin restart_cluster [options]
```

Options

Option	Description
`-h` `--help`	Show help for this command.

Example

```
# bda-oss-admin restart_cluster
Restarting the cluster...
********************
Cluster restarted successfully
```

7

Use the odcp Command Line Utility to Copy Data

Use the `odcp` command line utility to manage copy jobs data between HDFS on your cluster and remote storage providers.

Topics

What Is odcp?

`odcp` is a command line interface for copying very large files in a distributed environment.

`odcp` uses Spark to provide parallel transfer of one or more files. It takes the input file and splits it into chunks, which are then transferred in parallel to the destination. By default, transferred chunks are then merged back to one output file.

`odcp` is compatible with Cloudera Distributed Hadoop 5.7.x and supports copying files when using the following:

- Apache Hadoop Distributed File Service (HDFS)
- Apache WebHDFS and Secure WebHDFS (SWebHDFS)
- Oracle Cloud Infrastructure Object Storage Classic
- Amazon Simple Storage Service (S3)
- Oracle Cloud Infrastructure Object Storage
- Hypertext Transfer Protocol (HTTP) and HTTP Secure (HTTPS) — Used for sources only.

odcp Reference

The odcp command-line utility has the single command `odcp`, with parameters and options as described below.

Syntax

```
odcp [options] source1 [source2 ...] destination
```

Parameters

Parameter	Description
source1 [*source2* ...]	The source can be any of the following: • One or more individual files. Wildcard characters are allowed (glob patterns). • One or more HDFS directories. • One or more storage containers. If you specify multiple sources, list them one after the other: `odcp source1 source2 source3 destination` If two or more source files have the same name, nothing is copied and `odcp` throws an exception. Regular expressions are supported through these parameters: • `--srcPattern pattern` Files with matching names are copied. This parameter is ignored if the `--groupBy` parameter is set. • `--groupBy pattern` Files with matching names are copied and are then concatenated into one output file. Set a name for the concatenated file name by using the parameter `--groupName output_file_name`. When the `--groupBy` parameter is used, the `--srcPattern` parameter is ignored.
destination	The destination can be any of the following: • A specified file in an HDFS directory or a storage container If you don't specify a file name, the name of the source file is used for the copied file at the destination. But you can specify a different filename at the destination, to prevent overwriting a file with the same name. • An HDFS directory • A storage container

Use the following formats:

- For HDFS:

  `hdfs:///path/[file]`

  For example: `hdfs:///user/company/data.raw`

  or

  `hdfs://[host:port]/path/[file]`

  For example: `hdfs://192.0.2.0:22/user/company/data.raw`

- For Oracle Storage Cloud Service::

  `swift://container.provider/[file]`

  where

  – *container* is the name of a container in the Oracle Storage Cloud Service instance.

  – `provider` is the provider name that serves as an alias for the credentials for accessing the instance. See Register Storage Credentials with the Cluster.

  For example: `swift://feeds.BDCS/stream-061016-1827-534`

For examples showing other storage types, see odcp Supported Storage Sources and Targets

Options

Option	Description
`-b` `--block-size`	Destination file part size in bytes. • Default = `134217728` • Minimum = `1048576` • Maximum = `2147483647` The remainder after dividing `partSize` by `blockSize` must be equal to zero.
`-c` `--concat`	Concatenate the file chunks (default).
`--executor-cores`	Specify the number of executor cores. The default value is 5.
`--executor-memory`	Specify the executors memory limit in gigabytes. The default value is `40 GB`.
`--extra-conf`	Specify extra configuration options. For example: `--extra-conf spark.kryoserializer.buffer.max=128m`
`--groupBy`	Specify files to concatenate to a *`destination`* file by matching source file names with a regular expression.
`-h` `--help`	Show help for this command.
`--krb-keytab`	The full path to the keytab file of the Kerberos principal. (Use in a Kerberos-enabled Spark environment only.)
`--krb-principal`	The Kerberos principal. (Use in a Kerberos-enabled Spark environment only.)
`-n` `--no-clobber`	Don't overwrite an existing file.
`--non-recursive`	Don't copy files recursively.
`--num-executors`	Specify the number of executors. The default value is 3 executors.
`--progress`	Show the progress of the data transfer.
`--retry`	Retry if the previous transfer failed or was interrupted.
`--partSize`	Destination file part size in bytes. • Default = `536870912` • Minimum = `1048576` • Maximum = `2147483647` The remainder after dividing `partSize` by `blockSize` must be equal to zero.
`--spark-home`	The path to a directory containing an Apache Spark installation. If nothing is specified, `odcp` tries to find it in `/opt/cloudera directory`.

Option	Description
`--srcPattern`	Filters sources by matching the source name with a regular expression. `--srcPattern` is ignored when the `--groupBy` parameter is used.
`--sync`	Synchronize the *`destrination`* with the *`source`*.
`-V`	Enable verbose mode for debugging.

Copying Data With odcp

Use the Oracle Big Data Cloud Service distributed-copy utility odcp at the command line to copy data between HDFS on your cluster and various other supported storage providers.

See odcp Reference for the odcp syntax, parameters, and options.

Topics

- Prerequisites
- Operations Allowed When Using odcp to Copy Data, by Storage Type
- odcp Supported Storage Sources and Targets
- Filter and Copy Files
- Filter, Copy, and Group Files
- Copy Files from an HTTP Server
- Use odcp to Copy Data on a Secure Cluster
- Synchronize the Destination with Source
- Retry a Failed Copy Job

Prerequisites

For all the operations shown on this page, every cluster node must have

- Access to all running storage services
- All required credentials established, for example Oracle Storage Cloud Service accounts

Operations Allowed When Using odcp to Copy Data, by Storage Type

You can use odcp to copy data between a number of different kinds of storage. The operations available for each possible combination are shown below.

For examples of each scenario, see odcp Supported Storage Sources and Targets.

The following table summarizes what you can do with odcp when copying data between various types of storage. The first two columns show the scenario (from/to) and the remaining columns show the operations.

When Transferring Data From...	To...	Copy	Filter and Copy	Filter, Copy, and Group	Move	Sync	Retry
HDFS	HDFS	**yes**	**yes**	**yes**	no	**yes**	**yes**
HDFS	WebHDFS	**yes**	**yes**	**yes**	no	no	no
HDFS	Secure WebHDFS	**yes**	**yes**	**yes**	no	no	no
HDFS	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	**yes**	**yes**
HDFS	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage Classic	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	**yes**	**yes**
Oracle Cloud Infrastructure Object Storage Classic	HDFS	**yes**	**yes**	**yes**	no	**yes**	**yes**
Oracle Cloud Infrastructure Object Storage Classic	WebHDFS	**yes**	**yes**	**yes**	no	no	no
Oracle Cloud Infrastructure Object Storage Classic	Secure WebHDFS	**yes**	**yes**	**yes**	no	no	no
Oracle Cloud Infrastructure Object Storage Classic	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage Classic	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no
WebHDFS	WebHDFS	**yes**	**yes**	**yes**	no	no	no
WebHDFS	HDFS	**yes**	**yes**	**yes**	no	no	no
WebHDFS	HDFS	**yes**	**yes**	**yes**	no	no	no
WebHDFS	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	no	no

When Transferring Data From...	To...	Copy	Filter and Copy	Filter, Copy, and Group	Move	Sync	Retry
WebHDFS	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
WebHDFS	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no
Secure WebHDFS	WebHDFS	**yes**	**yes**	**yes**	no	no	no
Secure WebHDFS	HDFS	**yes**	**yes**	**yes**	no	no	no
Secure WebHDFS	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	no	no
Secure WebHDFS	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
Secure WebHDFS	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	HDFS	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	WebHDFS	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	Secure WebHDFS	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
Amazon Simple Storage Service (S3)	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no

When Transferring Data From...	To...	Copy	Filter and Copy	Filter, Copy, and Group	Move	Sync	Retry
Oracle Cloud Infrastructure Object Storage	HDFS	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage	WebHDFS	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage	Secure WebHDFS	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no
Oracle Cloud Infrastructure Object Storage	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no
HTTP	HDFS	**yes**	**yes**	**yes**	no	no	no
HTTP	WebHDFS	**yes**	**yes**	**yes**	no	no	no
HTTP	Secure WebHDFS	**yes**	**yes**	**yes**	no	no	no
HTTP	Oracle Cloud Infrastructure Object Storage Classic	**yes**	**yes**	no	no	no	no
HTTP	Oracle Cloud Infrastructure Object Storage	**yes**	**yes**	no	no	no	no
HTTP	Amazon Simple Storage Service (S3)	**yes**	**yes**	no	no	no	no

odcp Supported Storage Sources and Targets

The following examples show different scenarios for copying data between the various storage systems and services supported by `odcp`.

Copy From...	To...	Example
HDFS	HDFS	`[oracle@cfclbv2491 ~]$ odcp hdfs:///user/example/bigdata.file hdfs:///user/example/bigdata.file.copy`
HDFS	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp hdfs:///user/example/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
HDFS	Secure WebHDFS	`[oracle@cfclbv2491 ~]$ odcp hdfs:///user/example/bigdata.file swebhdfs://webhdfs-host:50470/user/example/bigdata.file.copy`
HDFS	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp hdfs:///user/example/bigdata.file swift://aserver.a424392/bigdata.file.copy`
HDFS	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp hdfs:///user/example/bigdata.file s3a://aserver/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file swift://aserver.a424392/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic	HDFS	`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file hdfs:///user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic	Secure WebHDFS	`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file swebhdfs://webhdfs-host:50470/user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file s3a://aserver/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage Classic		`[oracle@cfclbv2491 ~]$ odcp swift://aserver.a424392/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`

Copy From...	To...	Example
WebHDFS	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
WebHDFS	HDFS	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file hdfs:///user/example/bigdata.file.copy`
WebHDFS	HDFS	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file hdfs:///user/example/bigdata.file.copy`
WebHDFS	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file swift://aserver.a424392/bigdata.file.copy`
WebHDFS	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file s3a://aserver/bigdata.file.copy`
WebHDFS	Oracle Cloud Infrastructure Object Storage	`[oracle@cfclbv2491 ~]$ odcp webhdfs://webhdfs-host:50070/user/example/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`
Secure WebHDFS	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp swebhdfs://webhdfs-host:50470/user/example/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
Secure WebHDFS	HDFS	`[oracle@cfclbv2491 ~]$ odcp swebhdfs://webhdfs-host:50470/user/example/bigdata.file hdfs:///user/example/bigdata.file.copy`
Secure WebHDFS	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp swebhdfs://webhdfs-host:50470/user/example/bigdata.file swift://aserver.a424392/bigdata.file.copy`
Secure WebHDFS	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp swebhdfs://webhdfs-host:50470/user/example/bigdata.file s3a://aserver/bigdata.file.copy`
Secure WebHDFS	Oracle Cloud Infrastructure Object Storage	`[oracle@cfclbv2491 ~]$ odcp swebhdfs://webhdfs-host:50070/user/example/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`
Amazon Simple Storage Service (S3)	HDFS	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file hdfs:///user/example/bigdata.file.copy`
Amazon Simple Storage Service (S3)	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file hdfs:///user/example/bigdata.file.copy`
Amazon Simple Storage Service (S3)	Secure WebHDFS	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file swebhdfs://webhdfs-host:50470/user/example/bigdata.file.copy`

Copy From...	To...	Example
Amazon Simple Storage Service (S3)	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file swift://aserver.a424392/bigdata.file.copy`
Amazon Simple Storage Service (S3)	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file s3a://aserver/bigdata.file.copy`
Amazon Simple Storage Service (S3)	Oracle Cloud Infrastructure Object Storage	`[oracle@cfclbv2491 ~]$ odcp s3a://aserver/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	HDFS	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace/bigdata.file hdfs:///user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	Secure WebHDFS	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace/bigdata.file swebhdfs://webhdfs-host:50470/user/example/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace//bigdata.file swift://aserver.a424392/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace/bigdata.file s3a://aserver/bigdata.file.copy`
Oracle Cloud Infrastructure Object Storage	Oracle Cloud Infrastructure Object Storage	`[oracle@cfclbv2491 ~]$ odcp examplebmc://bucket@namespace/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`
HTTP file	HDFS	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/bigdata.file hdfs:///user/example/bigdata.file.copy`
HTTP file	WebHDFS	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/bigdata.file webhdfs://webhdfs-host:50070/user/example/bigdata.file.copy`
HTTP file	Secure WebHDFS	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/my.file swebhdfs://webhdfs-host:50470/user/example/bigdata.file.copy`
HTTP file	Oracle Cloud Infrastructure Object Storage Classic	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/bigdata.file swift://aserver.a424392/bigdata.file.copy`

Copy From...	To...	Example
HTTP file	Oracle Cloud Infrastructure Object Storage	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/bigdata.file examplebmc://bucket@namespace/bigdata.file.copy`
HTTP file	Amazon Simple Storage Service (S3)	`[oracle@cfclbv2491 ~]$ odcp http://exampleserver.com/my.file s3a://aserver/bigdata.file.copy`

Use bda-oss-admin with odcp

Use `bda-oss-admin` commands to configure the cluster for use with storage providers. This makes it easier and faster to use odcp with the storage provider.

Any user with access privileges to the cluster can run odcp.

> **Note:**
>
> To copy data between HDFS and Oracle Cloud Infrastructure Object Storage Classic, you must have an Oracle Cloud Infrastructure Object Storage Classic account, which isn't included with a Oracle Big Data Cloud Service account. To obtain an Oracle Cloud Infrastructure Object Storage Classic account, go to https://cloud.oracle.com/storage, or contact an Oracle Sales Representative.

Procedure

To copy data between HDFS and a storage provider:

1. Open a command shell and connect to the cluster. You can connect to any node for which you have HDFS access rights. (Oracle Cloud Infrastructure Object Storage Classic is accessible from all nodes.) See Connect to a Cluster Node Through Secure Shell (SSH).
2. Set shell environment variable values for Cloudera Manager access. See Setting bda-oss-admin Environment Variables

   Set these environment variables

   - `CM_ADMIN` — Cloudera Manager administrator user name
   - `CM_PASSWORD` — Cloudera Manager administrator password
   - `CM_URL` — Cloudera Manager URL
3. You must also have access privileges to the storage provider you want to use. If you're using the Oracle Cloud Infrastructure Object Storage Classic instance that was registered when the Hadoop cluster was created, you don't have to set any environment variables to access it. By default, that storage service instance has the provider name `BDCS`, and you can use that name to access it; for example, `swift://MyContainer.BDCS/data.raw`. See Create a Cluster and Register Storage Credentials with the Cluster.

You can also use providers (credentials) other than BDCS, to access different storage service instances or to access them as different users:

- Any providers besides BDCS must be added to the Hadoop configuration.
  - To see what credentials are already added, use the `bda-oss-admin list_swift_creds` command. You can also look at the `/etc/hadoop/conf/core-site.xml` file. See Reviewing the Configuration.
  - To add a single credential, use the `bda-oss-admin add_swift_cred` command.
  - To add multiple credentials, use the `bda-oss-admin import_swift_creds` command.
- Set the PROVIDER_NAME environment variable to refer to the provider you want to use. For example, if you have a provider named rssfeeds-admin2, use SSH to connect to the cluster and enter:

  ```
  # PROVIDER_NAME="rssfeeds-admin2"
  ```

  Or, in a shell script:

  ```
  export PROVIDER_NAME="rssfeeds-admin2"
  ```

  Then, in your odcp commands, use that provider name; for example; `swift://aContainer.rssfeeds-admin2/data.raw`.

See Register Storage Credentials with the Cluster.

4. You can use the `hadoop fs -ls` command to browse your HDFS and storage data.
   - To browse HDFS, use:
     `hadoop fs -ls`
   - To browse storage, use:
     `hadoop fs -ls swift://container.provider/`

     You can also browse storage in Big Data Manager.
5. Use the odcp command to copy files between Oracle Cloud Infrastructure Object Storage Classic and HDFS, as shown in the following examples.

Examples

- Copy a file from HDFS to an Oracle Cloud Infrastructure Object Storage Classic container:

  ```
  # usr/bin/odcp hdfs:///user/example/data.raw swift://
  myContainer.myProvider/data.raw
  ```

- Copy a file from an Oracle Cloud Infrastructure Object Storage Classic container to HDFS:

  ```
  # usr/bin/odcp swift://myContainer.myProvider/data.raw hdfs:///user/
  example/data.raw
  ```

- Copy a directory from HDFS to an Oracle Cloud Infrastructure Object Storage Classic container:

  ```
  # usr/bin/odcp hdfs:///user/data/ swift://myContainer.myProvider/backup
  ```

- If you have more than three nodes, you can increase transfer speed by specifying a higher number of executors. For example, if you have six nodes, use a command such as:

  ```
  # usr/bin/odcp --num-executors=6 hdfs:///user/company/data.raw swift://
  myContainer.myProvider/data.raw
  ```

Filter and Copy Files

Use the `odcp` command with the `--srcPattern` option to filter and copy files, as shown in the following example.

sdf`kjnlasn`

```
# list source directory
[oracle@cfclbv2491 ~]$ hadoop fs -ls swift://rstrejc.a424392/logs/
Found 3 items
-rw-rw-rw-   1    3499940 2016-10-18 09:58 swift://rstrejc.a424392/logs/
spark.log
-rw-rw-rw-   1    7525772 2016-10-18 10:00 swift://rstrejc.a424392/logs/
hadoop.log
-rw-rw-rw-   1          8 2016-10-18 10:13 swift://rstrejc.a424392/logs/
report.txt

# filter and copy files
[oracle@cfclbv2491 ~]$ odcp -V --srcPattern ".*log" swift://
rstrejc.a424392/logs/ hdfs:///user/oracle/filtered/

# list destination directory
[oracle@cfclbv2491 ~]$ hadoop fs -ls hdfs:///user/oracle/filtered
Found 2 items
-rw-r--r--   3 oracle hadoop    3499940 2016-10-18 10:29 hdfs:///user/
oracle/filtered/spark.log
-rw-r--r--   3 oracle hadoop    7525772 2016-10-18 10:30 hdfs:///user/
oracle/filtered/hadoop.log
```

Filter, Copy, and Group Files

Use the `odcp` command with the `-–groupBy` and `-–groupName` options to filter, copy, and group files, as shown in the following example:

```
# list source directory
[oracle@cfclbv2491 ~]$ hadoop fs -ls swift://rstrejc.a424392/logs/
Found 3 items
-rw-rw-rw-   1    3499940 2016-10-18 09:58 swift://rstrejc.a424392/logs/
spark.log
-rw-rw-rw-   1    7525772 2016-10-18 10:00 swift://rstrejc.a424392/logs/
hadoop.log
```

```
-rw-rw-rw-   1          8 2016-10-18 10:13 swift://rstrejc.a424392/logs/
report.txt

# copy and group files
[oracle@cfclbv2491 ~]$ odcp --groupBy ".*log" --groupName "summary.log"
swift://rstrejc.a424392/logs/ hdfs:///user/oracle/logs/

# list destination directory
[oracle@cfclbv2491 ~]$ hadoop fs -ls hdfs:///user/oracle/logs
Found 1 items
-rw-r--r--   3 oracle hadoop   11025712 2016-10-18 10:00 hdfs:///user/
oracle/logs/summary.log
```

Copy Files from an HTTP Server

You can use ODCP to copy files from an HTTP Server in a number of ways, as described below.

Copy Files From an HTTP Server

There are two ways to download files via the HTTP protocol:

1. Specify each of files as a source file:

   ```
   [oracle@cfclbv2491 ~]$ odcp http://example.com/fileA http://example.com/
   fileB swift://rstrejc.a424392/dstDirectory
   ```

2. Create a list of files to download:

   ```
   [oracle@cfclbv2491 ~]$ odcp --file-list hdfs:///files_to_download --
   file-list http://example.com/logs_to_download swift://rstrejc.a424392/
   dstDirectory
   ```

Use a File List to Specify Files

A file list is a comma-separated value (CSV) list file with following schema:

link_to_file[,http headers encoded in Base64]

For example:

```
http://172.16.253.111/public/big.file
https://172.16.253.111/public/small.file
http://172.16.253.111/private/
secret.file,QXV0aG9yaXphdGlvbjogQmFzaWMgYjNKaFkyeGxPa2cwY0hCNVJqQjQK
https://oracle:H4ppyF0x@172.16.253.111/private/small.file
```

where *QXV0aG9yaXphdGlvbjogQmFzaWMgYjNKaFkyeGxPa2cwY0hCNVJqQjQK* is a Base64 encoded string (HTTP headers):

- `Authorization: Basic b3JhY2xlOkg0cHB5RjB4`

For example:

```
[oracle@cfclbv2491 ~]$ odcp --file-list hdfs:///files_to_download --file-
list http://example.com/logs_to_download swift://rstrejc.a424392/
dstDirectory
```

Copy Files from an HTTP Server By Using the Proxy

When using the HTTP proxy, download files as shown in the following example:

```
[oracle@cfclbv2491 ~]$ odcp --http-proxy-host www-proxy.example.com http://
example.com/fileA swift://rstrejc.a424392/dstDirectory
```

Copy a List of Files from an HTTP Server By Using a File with Predefined HTTP Headers

If you need to specify HTTP headers but you don't want to specify them in a file list, you can create a separate file with HTTP headers and pass the file to ODCP, as shown in the example below:

```
[oracle@cfclbv2491 ~]$ odcp --http-headers hdfs:///file_with_http_headers
http://example.com/logs_to_download swift://rstrejc.a424392/dstDirectory
```

The structure of the file with HTTP headers is:

```
regex_pattern,http_headers
```

For example, the following file will apply specific HTTP headers for files which contain "image" or "log" in their path or name:

```
.*image.*,QXV0aG9yaXphdGlvbjogQmFzaWMgYjNKaFkyeGxPa2cwY0hCNVJqQjQK
.*log.*,QXV0aG9yaXphdGlvbjogQmFzaWMgYjNKaFkyeGxPa2cwY0hCNVJqQjQK
```

Use odcp to Copy Data on a Secure Cluster

Using odcp to copy data on a Kerberos-enabled cluster requires some additional steps.

> **Note:**
>
> In Oracle Big Data Cloud Service, Oracle Big Data Cloud Service, a cluster is Kerberos-enabled when it's created with "Secure Setup."

If you want to execute a long running job or run odcp from an automated shell script or from a workflow service such as Apache Oozie, then you must pass to the `odcp` command a Kerberos principal and the full path to the principal's `keytab` file, as described below:

1. Use SSH to connect to any node on the cluster.

2. Choose the principal to be used for running the `odcp` command. In the example below it's `odcp@BDACLOUDSERVICE.EXAMPLE.COM`.
3. Generate a `keytab` file for the principal, as shown below:

```
$ kutil
ktutil:  addent -password -p odcp@BDACLOUDSERVICE.EXAMPLE.COM -k 1 -e
rc4-hmac
Password for odcp@BDACLOUDSERVICE.EXAMPLE.COM: [enter your password]
ktutil:  addent -password -p odcp@BDACLOUDSERVICE.EXAMPLE.COM -k 1 -e
aes256-cts
Password for odcp@BDACLOUDSERVICE.EXAMPLE.COM: [enter your password]
ktutil:  wkt /home/odcp/odcp.keytab
ktutil:  quit
```

4. Pass the principal and the full path to the `keytab` file to the `odcp` command, for example:

```
odcp --krb-principal odcp@BDACLOUDSERVICE.EXAMPLE.COM --krb-keytab /
home/odcp/odcp.keytab source destination
```

If you just want to execute a short-running ODCP job from the console, you don't have to generate a `keytab` file or specify the principal. You just have to have an active Kerberos token (created using the `kinit` command).

Synchronize the Destination with Source

You can synchronize the destination with the source at the level of the file parts. When syncing HDFS with Oracle Storage Service, use the HDFS `partSize` equal to the file `partSize` on Oracle Storage Service.

What You Can Do When Synchronizing

The following list shows what you can do when synchronizing HDFS and Oracle Cloud Infrastructure Object Storage Classic sources and destinations:

- HDFS to Oracle Cloud Infrastructure Object Storage Classic
  - Retrieve a list of already uploaded segments on Oracle Cloud Infrastructure Object Storage Classic.
  - Read file parts from HDFS.
  - Compare each file part checksum on HDFS with a checksum on Oracle Cloud Infrastructure Object Storage Classic that is already uploaded. If they're the same, you can skip the transfer. Otherwise you can upload a part from HDFS to Oracle Cloud Infrastructure Object Storage Classic.
- Oracle Cloud Infrastructure Object Storage Classic to HDFS
  - Retrieve list of already downloaded parts on HDFS.
  - Split already concatenated files into parts on HDFS and calculate checksums.
  - Before downloading a segment from Oracle Cloud Infrastructure Object Storage Classic, compare its checksum with an already downloaded part checksum on HDFS. If they're the same, skip the transfer. Otherwise, you can download the segment from Oracle Cloud Infrastructure Object Storage Classic and store it as a file part on HDFS.

- Oracle Cloud Infrastructure Object Storage Classic to Oracle Cloud Infrastructure Object Storage Classic
  - Retrieve a list of already uploaded segments on Oracle Cloud Infrastructure Object Storage Classic.
  - Retrieve a list of source segments on Oracle Cloud Infrastructure Object Storage Classic.
  - Before downloading a segment from Oracle Cloud Infrastructure Object Storage Classic, compare its checksum with already uploaded segment checksum on Oracle Cloud Infrastructure Object Storage Classic. If they're the same, you can skip the transfer. Otherwise, you can download the segment from one Oracle Cloud Infrastructure Object Storage Classic and upload it to another Oracle Cloud Infrastructure Object Storage Classic.
- HDFS to HDFS
  - Retrieve a list of already downloaded parts on HDFS.
  - Split already concatenated files into parts on HDFS and calculate checksums.
  - Read file parts from HDFS.
  - Compare each file part checksum on HDFS with already stored part checksums on HDFS. If they're the same, you can skip the transfer. Otherwise you can store the part from HDFS to HDFS.

Examples

```
# sync file hdfs:///user/oracle/bigdata.file with swift://rstrejc.a42439/
bigdata.file
odcp --sync hdfs:///user/oracle/bigdata.file swift://rstrejc.a42439

# sync file hdfs:///user/oracle/bigdata.file with swift://rstrejc.a42439/
bigdata.file.new
odcp --sync hdfs:///user/oracle/bigdata.file swift://rstrejc.a42439/
bigdata.file.new

# sync directory hdfs:///user/oracle/directoryWithBigData with swift://
rstrejc.a42439/directoryWithBigData
odcp --sync hdfs:///user/oracle/directoryWithBigData swift://
rstrejc.a42439/

# sync directory hdfs:///user/oracle/directoryWithBigData with swift://
rstrejc.a42439/someDirectory/directoryWithBigData
odcp --sync hdfs:///user/oracle/directoryWithBigData swift://
rstrejc.a42439/someDirectory
```

Retry a Failed Copy Job

If a copy job fails, you can retry it. When retrying the job, the source and destination are automatically synchronized. Therefore odcp doesn't transfer successfully transferred file parts from source to destination.

The retry mechanism works as follows:

1. Before transferring files, odcp retrieves the destination file status and stores it to HDFS.

2. When a retry operation is required,
   a. odcp reads the destination file status stored on HDFS.
   b. Input and output files are re-indexed with same result as in the failed execution.
   c. The re-indexed files are synchronized.
3. If the copying operation is successful, odcp deletes the stored file status from HDFS.

Example:

```
# Run odcp and let's assume it is going to fail
odcp hdfs:///user/oracle/bigdata.file swift://rstrejc.a424392/
# Run same command with --retry option
odcp --retry hdfs:///user/oracle/bigdata.file swift://rstrejc.a424392/
```

Debugging odcp

You must configure the cluster to enable debugging for odcp.

To configure the cluster:

1. As the `root` user, add following lines to `/etc/hadoop/conf/log4j.properties` on each node of the cluster:

   ```
   log4j.logger.oracle.paas.bdcs.conductor=DEBUG
   log4j.logger.org.apche.hadoop.fs.swift.http=DEBUG
   ```

   Or, to configure all nodes:

   ```
   $ dcli -c $NODES "echo 'log4j.logger.oracle.paas.bdcs.conductor=DEBUG'
   >> /etc/hadoop/conf/log4j.properties"
   $ dcli -c $NODES "echo
   'log4j.logger.org.apche.hadoop.fs.swift.http=DEBUG' >> /etc/hadoop/conf/
   log4j.properties"
   ```

2. As the `oracle` user, find the logs in following HDFS directory:

   `hdfs:///tmp/logs/`*`username`*`/logs/application_`*`application_ID/`*

   For example:

   ```
   $ hadoop fs -ls /tmp/logs/oracle/logs/
   Found 15 items
   drwxrwx---   - oracle hadoop       0 2016-08-23 07:29 /tmp/logs/oracle/
   logs/application_14789235086687_0029
   drwxrwx---   - oracle hadoop       0 2016-08-23 08:07 /tmp/logs/oracle/
   logs/application_14789235086687_0030
   drwxrwx---   - oracle hadoop       0 2016-08-23 08:20 /tmp/logs/oracle/
   logs/application_14789235086687_0001
   drwxrwx---   - oracle hadoop       0 2016-08-23 10:19 /tmp/logs/oracle/
   logs/application_14789235086687_0002
   drwxrwx---   - oracle hadoop       0 2016-08-23 10:20 /tmp/logs/oracle/
   ```

```
logs/application_14789235086687_0003
drwxrwx---   - oracle hadoop       0 2016-08-23 10:40 /tmp/logs/oracle/
logs/application_14789235086687_0004
...

# cat logs as:
hadoop fs -cat /tmp/logs/oracle/logs/application_1469028504906_0032/
slclbv0036.em3.oraclecloud.com_8041

# copy to local FShadoop fs -copyToLocal /tmp/logs/oracle/logs/
application_1469028504906_0032/
slclbv0036.em3.oraclecloud.com_8041 /tmp/log/
slclbv0036.em3.oraclecloud.com_8041
```

Collecting Transfer Rates

You can collect the transfer rates when debugging is enabled. Transfer rates are reported after every:

- Read chunk operation
- Write/upload chunk operation

The summary throughput is reported after a chunk transfer is completed. The summary throughput includeds all:

- Read operations
- Write/upload operations
- Spark framework operations (task distribution, task management, etc.)

Output Example:

```
./get-transfer-rates.sh application_1476272395108_0054 2>/dev/null
Action,Speed [MBps],Start time,End time,Duration [s],Size [B]
Download from OSS,2.5855451864420473,2016-10-31 11:34:48,2016-10-31
11:38:06,198.024,536870912
Download from OSS,2.548912231791706,2016-10-31 11:34:47,2016-10-31
11:38:08,200.87,536870912
Download from OSS,2.53447780846872,2016-10-31 11:34:47,2016-10-31
11:38:09,202.014,536870912
Download from OSS,2.5130931169717226,2016-10-31 11:34:48,2016-10-31
11:38:11,203.733,536870912
Write to HDFS,208.04550995530275,2016-10-31 14:00:30,2016-10-31
14:00:33,2.46099999999999967435,536870912
Write to HDFS,271.76220806794055,2016-10-31 14:00:38,2016-10-31
14:00:40,1.88400000000000001398,536870912
Write to HDFS,277.5067750677507,2016-10-31 14:00:43,2016-10-31
14:00:45,1.84499999999999985045,536870912
Write to HDFS,218.0579216354344,2016-10-31 14:00:44,2016-10-31
14:00:46,2.34800000000000013207,536870912
Write to HDFS,195.56913674560735,2016-10-31 14:00:44,2016-10-31
14:00:47,2.61799999999999978370,536870912
```

Use the following command to collect output rates:

```
get-transfer-rates.sh application_ID
```

8
Use odiff to Compare Large Data Sets

odiff (Oracle Distributed Diff) is a utility that compares large data sets stored in various locations.

odiff runs as a distributed Spark application. It is compatible with Cloudera Distributed Hadoop 5.7.x.

- Only HDFS and Oracle Storage Cloud Service are supported.
- When odiff compares two objects, no data is downloaded. Only segment checksums are compared. If objects are equal but have segments with different sizes, then they're evaluated as different objects
- The default size of a block to compare is 128 MB.
- The minimum block size to compare is 5 MB. The maximum is 2 GB.

odiff Reference

Use `odiff` at the command line, as described below.

Syntax

```
/usr/bin/odiff [OPTIONS] directory_or_file_1 directory_or_file_2
```

where

`directory_or_file` is a directory or a file, qualified by its path, for example, `file:///tmp/diff/originalFiles` (directory) or `file:///tmp/diff/originalFiles/file-b.txt` (file).

Environment Variable (Optional)

By default, `odiff` uses the first provider configured in `core-site.xml`. If `core-site.xml` contains more than one provider, you can specify which one to use by declaring the *`PROVIDER_NAME`* environment variable.

You can export the environment variable...

```
# export PROVIDER_NAME="some_value" /usr/bin/odiff [OPTIONS] path/file_1
path/file_2
```

...or inject it:

```
# PROVIDER_NAME="some_value" /usr/bin/odiff [OPTIONS] path/file_1 path/
file_2
```

Options

Option	Use
-b --diffBlockSize	Diff the file block sizes in bytes. Not used when comparing files stored in Oracle Storage Cloud Service.
-d --showDetails	Shows detailed output.
--executor-cores	Specify the count of executors cores. Default value is 5.
--executor-memory	Specify the executors memory limit in GB. Default value is 40.
--extra-conf	Specify extra configuration options; for example, `--extra-conf spark.kryoserializer.buffer.max=128m`
-h --help	Display this help text.
--krb-keytab	Specify the full path to the Kerberos keytab of the principal. Use in a Kerberos-enabled Spark environment only
--krb-principal	Specify the Kerberos principal. Use in a Kerberos-enabled Spark environment only.
--num-executors	Specify the count of executors, The default value is 3.
-O --output	Specify an output file.
--spark-home	Specify the path to the directory containing the Spark installation. If this option isn't specified, `odiff` tries to find it in the `/opt/cloudera directory`.
-V	Enable verbose mode for debugging.

Usage Examples

```
/usr/bin/odiff hdfs:///user/oracle/data.raw swift://myContainer.myProvider/
data.raw
```

```
/usr/bin/odiff swift://jmyContainer.myProvider/data.raw hdfs:///user/
oracle/odcp-data.raw
```

If you have more then three nodes you can increase transfer speed by specifying a higher number of executors. For example, for six nodes use following command:

```
/usr/bin/odiff --num-executors=6 hdfs:///user/oracle/data.raw swift://
myContainer.myProvider/data.raw
```

For debugging you can enable verbose mode using switch `-V`:

```
/usr/bin/odiff -V swift://jmyContainer.myProvider/data.raw hdfs:///user/
oracle/odcp-data.raw
```

Limitations:

`/usr/bin/odiff` consumes a lot of resources of your cluster. If you want to execute in parallel other Spark/MapReduce jobs, you need to decrease the number of executors, the executors memory, or the number of executors cores by using the `--num-executors`, `--executor-memory`, and `--executor-cores` parameters.

odiff Examples

This topic examines a more extended example of using `odiff` to compare data structures.

The Data

Consider the following situation:

- Files from a directory `originalFiles` were copied (by using the odcp distributed-copy tool) to a directory named `copiedFiles`.
- After copying, the user
  - **added** a few lines of text to `originalFiles/file-b.txt`
  - **deleted** a few lines of text in the `originalFiles/file-c.txt`
  - **modified** one byte of text in the `originalFiles/file-d.txt`
  - **created** `.hiddenFile` in the `copiedFiles` directory

As a result, the directories and files look like this:

```
originalFiles
    +-- file-a.txt
    +-- file-b.txt
    +-- file-c.txt
    `-- file-d.txt

copiedFiles
    +-- file-a.txt      (same as originalFiles/file-a.txt)
    +-- file-b.txt      (added few lines)
    +-- file-c.txt      (deleted few lines)
    +-- file-d.txt      (modified one byte)
    `-- .hiddenFile     (added)
```

The remaining sections show different `odiff` operations performed on the above data.

Compare Two Files (Original and Copied)

odiff Parameters	Output	Return Code
file:///tmp/diff/originalFiles/file-a.txt file:///tmp/diff/copiedFiles/file-a.txt --diffBlockSize 104857600	.Files file:///tmp/diff/originalFiles/file-a.txt and file:///tmp/diff/copiedFiles/file-a.txt are same.	0
file:///tmp/diff/originalFiles/file-a.txt file:///tmp/diff/copiedFiles/file-a.txt --diffBlockSize 104857600 --showDetails	Files file:///tmp/diff/originalFiles/file-a.txt and file:///tmp/diff/copiedFiles/file-a.txt are same.	0

Compare Two Directories (One With Original Files and the Other With Copied Files)

odiff Parameters	Output	Return Code
file:///tmp/diff/originalFiles file:///tmp/diff/copiedFiles --diffBlockSize 104857600	Directories file:///tmp/diff/originalFiles and file:///tmp/diff/originalFiles are same.	0
file:///tmp/diff/originalFiles file:///tmp/diff/copiedFiles --diffBlockSize 104857600 --showDetails	Directories file:///tmp/diff/originalFiles and file:///tmp/diff/originalFiles are same.	0

Compare Two Different Files

odiff Parameters	Output	Return Code
file:///tmp/diff/originalFiles/file-b.txt file:///tmp/diff/copiedFiles/file-b.txt --diffBlockSize 104857600	Files file:/tmp/diff/originalFiles/file-b.txt and file:/tmp/diff/copiedFiles/file-b.txt are different	1
file:///tmp/diff/originalFiles/file-b.txt file:///tmp/diff/copiedFiles/file-b.txt --diffBlockSize 104857600 --showDetails	Files file:/tmp/diff/originalFiles/file-b.txt and file:/tmp/diff/copiedFiles/file-b.txt are different. Block#00000001: equals Block#00000002: equals Block#00000003: equals Block#00000004: missing in revised file	1

Compare Two Different Directories

odiff Parameters	Output	Return Code
file:///tmp/diff/originalFiles file:///tmp/diff/copiedFiles --diffBlockSize 1048576	Found 1 same file(s), found 4 different file(s): Files file:/tmp/diff/originalFiles/file-b.txt and file:/tmp/diff/copiedFiles/file-b.txt are different Files file:/tmp/diff/originalFiles/file-c.txt and file:/tmp/diff/copiedFiles/file-c.txt are different Files file:/tmp/diff/originalFiles/file-a.txt and file:/tmp/diff/copiedFiles/file-a.txt are same Files file:/tmp/diff/originalFiles/file-d.txt and file:/tmp/diff/copiedFiles/file-d.txt are different Files file:/tmp/diff/originalFiles/.hiddenFile and file:/tmp/diff/copiedFiles/.hiddenFile are different: The original file file:/tmp/diff/originalFiles/.hiddenFile is missing	1

odiff Parameters	Output	Return Code
file:///tmp/diff/originalFiles file:///tmp/diff/copiedFiles --diffBlockSize 104857600 --showDetails	Found 1 same file(s), found 4 different file(s): Files file:/tmp/diff/originalFiles/file-a.txt and file:/tmp/diff/copiedFiles/file-a.txt are same Files file:/tmp/diff/originalFiles/file-d.txt and file:/tmp/diff/copiedFiles/file-d.txt are different Block#00000001: different Block#00000002: equals Block#00000003: equals Block#00000004: equals Block#00000005: equals Block#00000006: equals Files file:/tmp/diff/originalFiles/file-b.txt and file:/tmp/diff/copiedFiles/file-b.txt are different Block#00000001: equals Block#00000002: equals Block#00000003: equals Block#00000004: missing in revised file Files file:/tmp/diff/originalFiles/file-c.txt and file:/tmp/diff/copiedFiles/file-c.txt are different Block#00000001: equals Block#00000002: equals Block#00000003: different Files file:/tmp/diff/originalFiles/.hiddenFile and file:/tmp/diff/copiedFiles/.hiddenFile are different: Original file file:/tmp/diff/originalFiles/.hiddenFile is missing	1

9

Connect to Oracle Database with Oracle Big Data Connectors

This section describes how to connect to Oracle Database from Oracle Big Data Cloud Service using the Oracle Loader for Hadoop and Copy to Hadoop database connectors. These connectors are preinstalled and preconfigured on all cluster nodes in Oracle Big Data Cloud Service.

Oracle Loader for Hadoop and Copy to Hadoop are high speed connectors used to load data into and copy data from Oracle Database. The interface for these connectors is the Oracle Shell for Hadoop Loaders (OHSH) command line interface.

Topics

- Using the Oracle Shell for Hadoop Loaders Interface (OHSH)
- Using Oracle Loader for Hadoop
- Using Copy to Hadoop

Use the Oracle Shell for Hadoop Loaders Interface (OHSH)

The following sections describe how to use the Oracle Shell for Hadoop Loaders (OHSH) interface.

Oracle Shell for Hadoop Loaders is the preferred way to use the Oracle Loader for Hadoop and Copy to Hadoop database connectors. It includes a command line interface (whose simple command syntax can also be scripted) for moving data between Hadoop and Oracle Database using the database connectors.

About Oracle Shell for Hadoop Loaders

Oracle Shell for Hadoop Loaders is a helper shell that provides an easy-to-use command line interface to Oracle Loader for Hadoop and Copy to Hadoop. It has basic shell features such as command line recall, history, inheriting environment variables from the parent process, setting new or existing environment variables, and performing environmental substitution in the command line.

The core functionality of OHSH includes the following:

- Defining named external resources with which OHSH interacts to perform loading tasks.
- Setting default values for load operations.
- Running load commands.
- Delegating simple pre and post load tasks to the Operating System, HDFS, Hive, and Oracle. These tasks include viewing the data to be loaded, and viewing the data in the target table after loading.

Configure Oracle Big Data Cloud Service for Oracle Shell for Hadoop Loaders

To get started with OHSH in Oracle Big Data Cloud Service:

1. SSH to a node on Oracle Big Data Cloud Service and log in, then execute the following:

   ```
   sudo su oracle
   ```

2. Add `/opt/oracle/ohsh-<version>` to your `PATH` variable. The OHSH executable is at this location.
3. Start OHSH with the following command:

   ```
   ohsh
   ```

You're now ready to run OHSH commands to move data between Oracle Big Data Cloud Service and Oracle Database.

Get Started with Oracle Shell for Hadoop Loaders

Starting an OHSH Interactive Session

To start an interactive session, enter `ohsh` on the command line. This brings you to the OHSH shell (if you have ohsh in your path):

```
$ ohsh
ohsh>
```

You can execute OHSH commands in this shell (using the OHSH syntax). You can also execute commands for Beeline/Hive, Hadoop, Bash, and SQL*Plus. For non-OHSH commands, you add a delegation operator prefix ("%") to the name of the resource used to execute the command. For example:

```
ohsh> %bash0 ls –l
```

Scripting OHSH

You can also script the same commands that work in the CLI. The `ohsh` command provides three parameters for working with scripts.

- `ohsh –i <filename>.ohsh`

  The `–i` parameter tells OHSH to initialize an interactive session with the commands in the script before the prompt appears. This is a useful way to set up the required session resources and automate other preliminary tasks before you start working within the shell.

  ```
  $ ohsh –i initresources.ohsh
  ```

- `ohsh –f <filename>.ohsh`

The `ohsh` command with the `–f` parameter starts a non-interactive session and runs the commands in the script.

```
$ ohsh –f myunattendedjobs.ohsh
```

- `ohsh –i –f <filename>.ohsh`

  You can use `–i` and `–f` together to initialize a non-interactive session and then run another script in the session.

  ```
  $ ohsh -i mysetup.ohsh –f myunattendedjobs.ohsh
  ```

- `ohsh –c`

  This command dumps all Hadoop configuration properties that an OHSH session inherits at start up.

Working With OHSH Resources

A resource is some named entity that OHSH interacts with. For example: a Hadoop cluster is a resource, as is a JDBC connection to an Oracle database, a Hive database, a SQL*Plus session with an Oracle database, and a Bash shell on the local OS.

OHSH provides two default resources at start up: `hive0` (to connect to the `default` Hive database) and `hadoop0`.

- Using `hive0` resource to execute a Hive command:

  ```
  ohsh> %hive0 show tables;
  ```

  You can create additional Hive resources to connect to other Hive databases.

- Using the `hadoop0` resource to execute a Hadoop command:

  ```
  ohsh> %hadoop0 fs -ls
  ```

Within an interactive or scripted session, you can create instances of additional resources, such as SQL*Plus and JDBC. You need to create these two resources in order to connect to Oracle Database through OHSH.

- Creating an SQL*Plus resource:

  ```
  ohsh> create sqlplus resource sql0 connectid="bigdatalite.localdomain:
  1521/orcl"
  ```

- Creating a JDBC resource:

  ```
  ohsh> create jdbc resource jdbc0 connectid=<database connection URL>
  ```

- Showing resources:

  ```
  ohsh> show resources
  ```

This command lists default resources and any additional resources created within the session.

Getting Help

The OHSH shell provides online help for all commands.

To get a list of all OHSH commands:

```
ohsh> help
```

To get help on a specific command, enter `help`, followed by the command:

```
ohsh> help show
```

The table below describes the help categories available.

Help Command	Description
`help load`	Describes load commands for Oracle and Hadoop tables.
`help set`	Shows help for setting defaults for load operations. It also describes what load methods are impacted by a particular setting.
`help show`	Shows help for inspecting default settings.
`help shell`	Shows shell-like commands.
`help resource`	Show commands for creating and dropping named resources.

Use Oracle Loader for Hadoop

The following sections describe how to use Oracle Loader for Hadoop to load data from Hadoop into tables in Oracle Database.

About Oracle Loader for Hadoop

Oracle Loader for Hadoop (OLH) is an efficient and high-performance loader for fast loading of data from a Hadoop cluster into a table in an Oracle database.

Oracle Loader for Hadoop prepartitions the data if necessary and transforms it into a database-ready format. It can also sort records by primary key or user-specified columns before loading the data or creating output files. Oracle Loader for Hadoop uses the parallel processing framework of Hadoop to perform these preprocessing operations, which other loaders typically perform on the database server as part of the load process. Off-loading these operations to Hadoop reduces the CPU requirements on the database server, thereby lessening the performance impact on other database tasks.

Oracle Shell for Hadoop Loaders (OHSH) is the preferred way to use Oracle Loader for Hadoop. It includes a command line interface (whose simple command syntax can also be scripted) for moving data between Hadoop and Oracle Database using various resources, including Oracle Loader for Hadoop. See Use the Oracle Shell for Hadoop Loaders Interface (OHSH).

Get Started With Oracle Loader for Hadoop

These instructions show how to use Oracle Loader for Hadoop through OHSH.

Before You Start

This is what you need to know before using OLH to load an Oracle Database table with data stored in Hadoop:

- The password of the database schema you are connecting to (which is implied by the database connection URL).
- The name of the Oracle Database table.
- The source of the data living in Hadoop (either a path to an HDFS directory or the name of a Hive table).
- The preferred method for loading. Choose either JDBC or direct path. Direct path load is faster, but requires partitioning of the target table. JDBC does not.

About Resources

In OHSH, the term *resources* refers to the interfaces that OHSH presents for defining the data source, destination, and command language. Four types of resources are available:

- Hadoop resources – for executing HDFS commands to navigate HDFS and use HDFS as a source or destination.
- Hive resources – for executing Hive commands and specifying Hive as a source or destination.
- JDBC resources – for making JDBC connections to a database.
- SQL*Plus resources – for executing SQL commands in a database schema.

Two resources are created upon OHSH startup:

- `hive0` – enables access to the Hive database default.
- `hadoop0` – enables access to HDFS.

You can create SQL*Plus and JDBC resources with a session, as well as additional Hive resources (for example, to connect to other Hive databases). Assign a resource any name that is meaningful to you. In the examples below, we use the names `ora_mydatabase` and `sql0` .

Where resources are invoked in the commands below, the percent sign (%) prefix identifies a resource name.

Loading an Oracle Database Table

1. Start an OHSH session.

   ```
   $ ohsh
   ohsh>
   ```

2. Create the following resources:

- SQL*Plus resource

  ```
  ohsh> create sqlplus resource sql0 connectid="<database connection
  url>"
  ```

  At prompt, enter the database password.

- JDBC resource.

  You can provide any name. A name that indicates the target schema is recommended.

  ```
  ohsh> create jdbc resource ora_mydatabase connectid="<database
  connection ur1>"
  ```

  At the prompt, enter the database password.

- Additional Hive resources (if required). The default Hive resource `hive0` connects to the default database in Hive. If you want to connect to another Hive database, create another resource:

  ```
  ohsh> create hive resource hive_mydatabase
  connectionurl="jdbc:hive2:///<Hive database name>
  ```

3. Use the `load` command to load files from HDFS into a target table in the Oracle database.

   The following command loads data from a delimited text file in HDFS *<HDFS path>* into the target table in Oracle Database using the direct path option.

   ```
   ohsh> load oracle table ora_mydatabase:<target table in the Oracle
   database> from path hadoop0:/user/<HDFS path> using directpath
   ```

   > **Note:**
   >
   > The default direct path method is the fastest way to load a table. However, it requires partitioned target table. Direct path is always recommended for use with partition tables. Use the JDBC option to load into a non-partitioned target table.
   >
   > If the command does not explicitly state the load method, then OHSH automatically uses the appropriate method. If the target Oracle table is partitioned, then by default, OHSH uses direct path (i.e. Oracle OCI). If the Oracle table is not partitioned, it uses JDBC.

4. After loading, check the number of rows.

   You can do this conveniently from the OHSH command line:

   ```
   ohsh> %sql0 select count(*) from <target table in Oracle Database>
   ```

Loading a Hive Table Into an Oracle Database Table

You can use OHSH to load a Hive table into a target table in an Oracle database. The command below shows how to do this using the direct path method.

```
ohsh> load oracle table ora_mydatabase:<target table in Oracle Database>
from hive table hive0:<Hive table name>
```

Note that if the target table is partitioned, then OHSH uses direct path automatically. You do not need to enter `using directpath` explicitly in the command.

If the target table is non-partitioned, then specify the JDBC method instead:

```
ohsh> load oracle table ora_mydatabase:<target table in Oracle Database>
from hive table hive0:<Hive table name> using jdbc
```

> **Note:**
>
> The `load` command assumes that the column names in the Hive table and in the Oracle Database table are identically matched. If they do not match, then use OHSH `loadermap`.

Using OHSH Loadermaps

The simple load examples in this section assume the following:

- Where we load data from a text file in Hadoop into an Oracle Database table, the declared order of columns in the target table maps correctly to the physical ordering of the delimited text fields in the file.
- Where we load Hive tables in to Oracle Database tables, the Hive and Oracle Database column names are identically matched.

However, in less straightforward cases where the column names (or the order of column names and delimited text fields) do not match, use the OHSH `loadermap` construct to correct these mismatches.

You can also use a loadermap to specify a subset of target columns to load into table or in the case of a load from a text file, specify the format of a field in the load.

Loadermaps are not covered in this introduction.

Performance Tuning Oracle Loader for Hadoop in OHSH

Aside from network bandwidth, two factors can have significant impact on Oracle Loader for Hadoop performance. You can tune both in OHSH.

- Degree of parallelism

  The degree of parallelism affects performance when Oracle Loader for Hadoop runs in Hadoop. For the default method (direct path), parallelism is determined by

the number of reducer tasks. The higher the number of reducer tasks, the faster the performance. The default value is 4. To set the number of tasks:

```
ohsh> set reducetasks 18
```

For the JDBC option, parallelism is determined by the number of map tasks and the optimal number is determined automatically. However, remember that if the target table is partitioned, direct path is faster than JDBC.

- Load balancing

  Performance is best when the load is balanced evenly across reduce tasks. The load is detected by sampling. Sampling is always enabled by default for loads using the JDBC and the default copy method.

Debugging in OHSH

Several OHSH settings control the availability of debugging information:

- `outputlevel`

  The `outputlevel` is set to `minimal` by default. Set it to `verbose` in order to return a stack trace when a command fails:

  ```
  ohsh> set outputlevel verbose
  ```

- `logbadrecords`

  ```
  ohsh> set logbadrecords true
  ```

  This is set to `true` by default.

These log files are informative for debugging:

- Oracle Loader for Hadoop log files.

  ```
  /user/<username>/smartloader/jobhistory/oracle/<target table
  schema>/<target table name>/<OHSH job ID>/_olh
  ```

- Log files generated by the map and reduce tasks.

Other OHSH Properties That are Useful for Oracle Loader for Hadoop

You can set these properties on the OHSH command line or in a script.

- `dateformat`

  ```
  ohsh> set dateformat “yyyy-MM-dd HH:mm:ss”
  ```

  The syntax for this command is dictated by the Java date format.

- `rejectlimit`

  The number of rows that can be rejected before the load of a delimited text file fails.

- `fieldterminator`

The field terminator in loads of delimited text files.

- `hadooptnsadmin`

  Location of an Oracle TNS admin directory in the Hadoop cluster

- `hadoopwalletlocation`

  Location of the Oracle Wallet directory in the Hadoop cluster.

Use Copy to Hadoop

The following sections describe how to use Copy to Hadoop to copy Oracle Database tables to Hadoop.

About Copy to Hadoop

Copy to Hadoop makes it simple to identify and copy Oracle data to the Hadoop Distributed File System (HDFS).

Data exported to the Hadoop cluster by Copy to Hadoop is stored in Oracle Data Pump format. The Oracle Data Pump files can be queried by Hive. When the Oracle table changes, you can refresh the copy in Hadoop. Copy to Hadoop is primarily useful for Oracle tables that are relatively static, and thus do not require frequent refreshes.

Oracle Shell for Hadoop Loaders (OHSH) is the preferred way to use Copy to Hadoop. It includes a command line interface (whose simple command syntax can also be scripted) for moving data between Hadoop and Oracle Database using various resources, including Copy to Hadoop. See Use the Oracle Shell for Hadoop Loaders Interface (OHSH).

First Look: Loading an Oracle Table Into Hive and Storing the Data in Hadoop

This set of examples shows how to use Copy to Hadoop to load data from an Oracle table, store the data in Hadooop, and perform related operations within the OHSH shell. It assumes that OHSH and Copy to Hadoop are already installed and configured.

What's Demonstrated in The Examples

These examples demonstrate the following tasks:

- Starting an OHSH session and creating the resources you'll need for Copy to Hadoop.
- Using Copy to Hadoop to copy the data from the selected Oracle Database table to a new Hive table in Hadoop (using the resources that you created).
- Using the `load` operation to add more data to the Hive table created in the first example.
- Using the `create or replace` operation to drop the Hive table and replace it with a new one that has a different record set.
- Querying the data in the Hive table and in the Oracle Database table.
- Converting the data into other formats

> **Tip:**
>
> You may want to create select or create a small table in Oracle Database and work through these steps.

Starting OHSH, Creating Resources, and Running Copy to Hadoop

1. Start OHSH. (The startup command below assumes that you've added the OHSH path to your PATH variable as recommended.)

   ```
   $ ohsh
   ohsh>
   ```

2. Create the following resources.

   - SQL*Plus resource.

     ```
     ohsh> create sqlplus resource sql0
     connectid="<database_connection_url>"
     ```

   - JDBC resource.

     ```
     ohsh> create jdbc resource jdbc0
     connectid="<database_connection_url>"
     ```

   > **Note:**
   >
   > For the Hive access shown in this example, only the default `hive0` resource is needed. This resource is already configured to connect to the default Hive database. If additional Hive resources were required, you would create them as follows:
   >
   > ```
   > ohsh> create hive resource hive_mydatabase
   > connectionurl="jdbc:hive2:///<Hive_database_name>"
   > ```

3. Include the Oracle Database table name in the `create hive table` command below and run the command below. This command uses the Copy to Hadoop `directcopy` method. Note that `directcopy` is the default mode and you do not actually need to name it explicitly.

   ```
   ohsh> create hive table hive0:<new_Hive_table_name> from oracle table
   jdbc0:<Oracle_Database_table_name> from oracle table
   jdbc0:<Oracle_Database_table_name> using directcopy
   ```

   The Oracle Table data is now stored in Hadoop as a Hive table.

Adding More Data to the Hive Table

Use the OHSH `load` method to add data to an existing Hive table.

Let's assume that the original Oracle table includes a time field in the format DD-MM-YY and that a number of daily records were added after the Copy to Hadoop operation that created the corresponding Hive table.

Use `load` to add these new records to the existing Hive table:

```
ohsh> load hive table hive0:<Hive_table_name> from oracle table
jdbc0:<Oracle_Database_table_name> where "(time >= '01-FEB-18')"
```

Using OHSH *create or replace*

The OHSH `create or replace` operation does the following:

1. Drops the named Hive table (and the associated Data Pump files) if a table by this name already exists.

> **Note:**
>
> Unlike `create or replace`, a `create` operation fails and returns an error if the Hive table and the related Data Pump files already exist.

2. Creates a new Hive table using the name provided.

Suppose some records were deleted from the original Oracle Database table and you want to realign the Hive table with the new state of the Oracle Database table. Hive does not support update or delete operations on records, but the `create or replace` operation in OHSH can achieve the same end result:

```
ohsh> create or replace hive table hive0:<new_hive_table_name> from oracle
table jdbc0:<Oracle_Database_table_name>
```

> **Note:**
>
> Data copied to Hadoop by Copy to Hadoop can be queried through Hive, but the data itself is actually stored as Oracle Data Pump files. Hive only points to the Data Pump files.

Querying the Hive Table

You can invoke a Hive resource in OHSH in order to run HiveQL commands. Likewise, you can invoke an SQL*Plus resource to run SQL commands. For example, these two queries compare the original Oracle Database table with the derivative Hive table:

```
ohsh> %sql0 select count(*) from <Oracle_Database_table_name>
ohsh> %hive0 select count(*) from <Hive_table_name>
```

Storing Data in Other Formats, Such as Parquet or ORC

By default, Copy to Hadoop outputs Data Pump files. In a `create` operation, you can use the "`stored as`" syntax to change the destination format to Parquet or ORC:

```
ohsh> %hive0 create table <Hive_table_name_parquet> stored as parquet as
select * from <Hive_table_name>
```

This example creates the Data Pump files, but then immediately copies them to Parquet format. (The original Data Pump files are not deleted.)

10

Use Oracle Big Data SQL Cloud Service with Oracle Big Data Cloud Service

With Oracle Big Data SQL, you can use SQL queries to access data in your Hadoop cluster and in an Oracle Database on Exadata Cloud Service.

Oracle Big Data SQL Cloud Service is an add-on to Oracle Big Data Cloud Service. The add-on includes an entitlement to an Oracle Exadata Cloud Service instance, which will be associated with the Oracle Big Data Cloud Service instance.

Topics

Add Oracle Big Data SQL

Oracle Big Data SQL Cloud Service can be used with Oracle Big Data Cloud Service.

You can do any of the following:

- Use universal cloud credits to buy entitlements Oracle Big Data Cloud Service. and Oracle Big Data SQL Cloud Service (plus the prerequisite Oracle Database Cloud Exadata Service).
- Purchase traditional nonmetered or metered subscriptions to Oracle Big Data Cloud Service. and Oracle Big Data SQL Cloud Service .
- Add nonmetered Oracle Big Data SQL Cloud Service to an existing nonmetered Oracle Big Data Cloud Service. subscription.

Purchase subscriptions through `cloud.oracle.com` or contact an Oracle Sales Representative.

The steps for setting up your services will vary slightly, depending on which you do. See Create Oracle Big Data Cloud Service Instances with Oracle Big Data SQL Cloud Service.

Create Oracle Big Data Cloud Service Instances with Oracle Big Data SQL Cloud Service

The first step in setting up Oracle Big Data Cloud Service with Oracle Big Data SQL Cloud Service is to create instances of both services. (Oracle Database Cloud

Exadata Service is required for running Oracle Big Data SQL Cloud Service. An entitlement to the Oracle Database Cloud Exadata Service is included in your subscription to Oracle Big Data SQL Cloud Service.)

Creating these instances allocates resources for the services and creates an association between them so they can communicate with each other. You create the Oracle Big Data Cloud Service instance first and then the Oracle Database Cloud Exadata Service instance.

> **Note:**
>
> You can't substitute an existing Oracle Database Cloud Exadata Service instance for the one that's included with the Oracle Big Data SQL Cloud Service subscription. You can only use an Oracle Database Cloud Exadata Service and an Oracle Big Data Cloud Service that are part of the same subscription.

Next Steps

1. Associate the Oracle Big Data SQL Cloud Service instance with your Oracle Big Data Cloud Service instance. See Associate anOracle Big Data SQL Cloud Service Instance with an Oracle Big Data Cloud Service Instance.
2. For either of the above scenarios, you can create a cluster any time after you've created the Oracle Big Data Cloud Service instance, either before or after creating the Oracle Database Cloud Exadata Service instance. See Create a Cluster.
3. Create a database deployment in your Oracle Database Cloud Exadata Service instance. See Creating a Database Deployment in *Administering Oracle Database Exadata Cloud Service*.
4. Associate the services. See Associate anOracle Big Data SQL Cloud Service Instance with an Oracle Big Data Cloud Service Instance.

Associate anOracle Big Data SQL Cloud Service Instance with an Oracle Big Data Cloud Service Instance

Use Oracle Big Data SQL Cloud Service to query data in your associated Oracle Big Data Cloud Service and Oracle Database Cloud Exadata Service instances.

To associate your Oracle Big Data SQL Cloud Service instance with your Oracle Big Data Cloud Service Instance:

1. Go to the **Service Overview** page for the Oracle Big Data Cloud Service cluster.
2. From the menu at the top of the page, select **Add Association**.
3. For the **Association Name**, there's only one choice: **Big Data Appliance — Exadata association**.
4. Enter a name and description for the association, and enter your Cloudera Manager password.

Use Oracle Big Data SQL

See Oracle Big Data SQL Online Documentation Library.