IBM SPSS Analytic Server
Version 3.1.1.1

Installation and Configuration Guide

IBM

Note

Before using this information and the product it supports, read the information in "Notices" on page 85.

Product Information

This edition applies to version 3, release 1, modification 1 of IBM SPSS Analytic Server and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Chapter 1. Prerequisites

Before installing Analytic Server, review the following information.

System requirements

For the most up-to-date system requirements information, use the Detailed system requirements reports at the IBM Technical Support site: http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/softwareReqsForProduct.html. On this page:

1. Type SPSS Analytic Server as the product name and click **Search**.
2. Select the wanted version and scope of report, then click **Submit**.

WebSocket traffic

You must ensure that WebSocket traffic between clients and the Analytic Server is not blocked by firewalls, VPN's, or other port blocking methods. The WebSocket port is the same as the general Analytic Server port.

SuSE Linux (SLES) 12

Perform the following tasks prior to installing Analytic Server on SuSE Linux 12:

1. Download a public key to your host from the following URL:
   ```
   https://ibm-open-platform.ibm.com/repos/IBM-SPSS-AnalyticServer/3.1.1.1/IBM-SPSS-ANALYTICSERVER-GPG-KEY.public
   ```
2. Import the public key by running the following command on your host:
   ```
   rpm --import IBM-SPSS-ANALYTICSERVER-GPG-KEY.public
   ```

Power systems

Ensure that the IBM XLC and XLF compilers are installed and included in the PATH on all hosts in the cluster.

You can find more information about getting a license for these compilers at the following web sites:

- XL C for Linux: http://www-03.ibm.com/software/products/en/xlcpp-linux
- XL Fortran for Linux: http://www-03.ibm.com/software/products/en/xlfortran-linux

Hortonworks Data Platform (HDP)

Before installing Analytic Server, you must ensure that at least one HDP client has been deployed in your clustered environment. Because the node that hosts Ambari Manager expects the `/usr/hdp` directory, the Analytic Server will fail in the absence of an HDP client.

Hive/HCatalog

If you plan to use NoSQL data sources, then configure Hive and HCatalog for remote access. Also ensure that `hive-site.xml` contains a *hive.metastore.uris* property in the form `thrift://<host_name>:<port>` that points to the active Thrift Hive Metastore server. Refer to your Hadoop distribution documentation for details.

Note: The Analytic Server Metastore cannot be installed on the same machine as the Hive Metastore.

If you want to use Hive 2.1, you must enable Hive 2.1 by enabling the **Interactive Query** setting in the Ambari console, and then enter `2.x` as the `hive.version` property during Analytic Server installation.

1. Open the Ambari console and add the following property in the **Analytic Server Advanced analytics.cfg** section.
   - Key: `hive.version`
   - Value: Enter the appropriate Hive version (for example, 2.x)
2. Save the configuration.

Note: Hive 2.1 is supported on HDP 2.5 and 2.6 with Spark 2.x.

Metadata repository

By default, Analytic Server installs and uses a MySQL database. Alternatively, you can configure Analytic Server to use an existing Db2 installation. Regardless of which type of database you choose, it must have an encoding of UTF-8.

MySQL

The default character set for MySQL is dependent upon the version and operating system. Use the following steps to determine whether your installation of MySQL is set to UTF-8.

1. Determine the version of MySQL.

   ```
   mysql -V
   ```

2. Determine the default character set for MySQL by running the following query from the MySQL command line interface.

   ```
   mysql>show variables like 'char%';
   ```

   If the character sets already set to UTF-8 no further changes are needed.

3. Determine the default collation for MySQL by running the following query from the MySQL command line interface.

   ```
   mysql>show variables like 'coll%';
   ```

   If the collation is already set to UTF-8 no further changes are needed.

4. If the default character set or collation is not UTF-8 refer to the MySQL documentation for details on how to edit /etc/my.cnf and restart the MySQL daemonto change the character set to UTF-8.

Db2 For more information on configuring Db2, see the Knowledge Center http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.kc.doc/welcome.html.

High-availability clusters

Load balancer

Your high availability cluster should have a load balancer that supports session affinity, sometimes also known as sticky sessions. Analytic Server identifies sessions with the cookie "request-token". This identifies a session for the duration of a user login for use in application-controlled session affinity. Please consult the documentation of your particular load balancer for the details of how it supports session affinity.

Analytic Server job failure

When an Analytic Server job fails because a cluster member fails, the job is normally restarted automatically on another cluster member. If the job does not resume, check to ensure there are at least 4 cluster members in the High Availability cluster.

Chapter 2. Ambari Installation and Configuration

Ambari-specific prerequisites

In addition to the general prerequisites, review the following information.

Services

Analytic Server is installed as an Ambari service. Prior to installing Analytic Server, you must ensure that the following clients are installed as Ambari services:

- HDFS/HDFS_CLIENT
- MAPREDUCE2/MAPREDUCE2_CLIENT
- HIVE/HIVE_CLIENT
- SPARK/SPARK_CLIENT (when Spark 1.x is used)
- SPARK2/SPARK2_CLIENT (when Spark 2.x is used)
- HBASE/HBASE_CLIENT (when HBASE is used)
- YARN
- Zookeeper

Password-less SSH

Set up password-less SSH for the root user between the Analytic Metastore host and all hosts in the cluster.

IBM SPSS Analytic Server installation precheck and postcheck tools

Precheck tool overview

The Analytic Server installation precheck tool helps reduce installation issues and runtime errors by identifying potential environment issues before Analytic Server installation.

The precheck tool verifies:

- OS and Ambari versions on the local system
- OS ulimit settings on the local system
- Available disk space on the local system
- Hadoop version
- Ambari service availability (HDFS, HCatalog, Spark, Hive, MapReduce, Yarn, Zookeeper, and so on)
- Analytic Server specific Ambari settings

Note: The precheck tool can be used after the running the self-extracting Analytic Server binary file.

Postcheck tool overview

The Analytic Server installation postcheck tool identifies configuration issues, after Analytic Server installation, by submitting REST API requests for processing:

- Data in HDFS
- Data in Hive/HCatalog
- Compressed data (including deflate, bz2, snappy)
- Data with PySpark
- Data that uses native SPSS components (including alm, tree, neuralnet, scoring, tascoring)
- Data with MapReduce

- Data with in-memory MapReduce

Tool location and prerequisites

After running the self-extracting Analytic Server binary file, the precheck tool is located in the following directories:

- **HDP**

  ```
  /opt/ibm/spss/analyticserver-ambari/3.1/ANALYTICSERVER/package/chktool/precheck.py
  ```

  ```
  [root@servername chktool]# cd /opt/ibm/spss/analyticserver-ambari/3.1/ANALYTICSERVER/package/chktool
  [root@servername chktool]# ls
  checkers data lib logs postcheck.py precheck.py readme.txt
  ```

- **Cloudera**

  ```
  /opt/cloudera/parcels/AnalyticServer-3.1.1.1/tools/com.spss.ibm.checker.zip
  ```

  ```
  [root@servername ~]# cd /opt/cloudera/parcels/AnalyticServer-3.1.1.1/tools/
  [root@servername tools]# ls
  com.spss.ibm.checker.zip configcollector.zip regex-files
  ```

  Note: The precheck tool is not available in the tools directory until you run the executable binary file and then distribute (**Download** > **Distribute**) and activate Analytic Server in the Cloudera Manager's **Parcels** page.

After installing Analytic Server, the postcheck tool is located in the following directories:

- **HDP**

  ```
  /opt/ibm/spss/analyticserver/3.1/tools/com.spss.ibm.checker.zip
  ```

- **Cloudera**

  ```
  /opt/cloudera/parcels/AnalyticServer-3.1.1.1/tools/com.spss.ibm.checker.zip
  ```

The tools must be run as root and require Python 2.6.X (or greater).

Before you install Analytic Server, the precheck tool should be run on all of the nodes that will host the Analytic Server service. Running the tool on a different node requires copying the entire chktool directory to the node.

If the precheck tool reports any failures, the failures must be addressed before you continue with the Analytic Server installation.

The chktool directory is available after the Analytic Server self-extracting binary is run (step 2 in the "Installation on Ambari" on page 5 section). If you choose to run an "Offline installation" on page 9, the chktool directory is available after the metadata RPM is installed.

Running the precheck tool

The following precheck example checks the Ambari cluster MyCluster that is running on myambarihost.ibm.com:8080, with SSL enabled, and uses the login credentials admin:admin:

```
python ./precheck.py --target B --cluster MyCluster --username admin
--password admin --host myambarihost.ibm.com --port 8080 --as_host myashost.ibm.com --ssl
```

Notes:

- The as_host value must be to be provided by either IP address or by a fully qualified domain name.
- The tool prompts for a password when the password argument is omitted.
- The precheck.py command includes usage help, which is displayed with the --h argument (python ./precheck.py --help).
- The --cluster argument is optional (the current cluster is identified when --cluster is not used).

As the precheck tool runs its checks, the status of each check displays in the command window. When a failure occurs, detailed information is available in the log file (the exact log file location is provided in the command window). The log file can be provided to IBM technical support when more support is required.

Running the postcheck tool

The postcheck tool verifies that Analytic Server is running properly and is able to process simple jobs. The following postcheck example checks an Analytic Server instance that is running on `myanalyticserverhost.ibm.com:9443`, with SSL enabled, and uses the login credentials `admin:ibmspss`:

```
python ./postcheck.py --host myanalyticserverhost.ibm.com --port 9443
--username admin --password ibmspss --ssl
```

When Knox is used with Analytic Server, the command is as the follows:

```
python ./postcheck.py --host myknoxserverhost.ibm.com --port 8443
--username admin --password ibmspss --ssl --gateway_url /gateway/default
```

To perform a single check, use the following command:

```
python ./postcheck.py --host myknoxserverhost.ibm.com --port 8443
--username admin --password ibmspss --ssl --gateway_url /gateway/default --check AS_PYSPARK_BUILDMODEL
```

Notes:

- The tool prompts for a password when the password argument is omitted.
- The `postcheck.py` command includes usage help, which is displayed with the `--h` argument (`python ./postcheck.py --help`).

As the postcheck tool runs its checks, the status of each check displays in the command window. When a failure occurs, detailed information is available in the log file (the exact log file location is provided in the command window). The log file can be provided to IBM technical support if more support is required.

Installation on Ambari

The basic process is to install the Analytic Server files on a host within the Ambari cluster, then add Analytic Server as an Ambari service.

"Online installation"
Choose online installation if your Ambari server host, and all nodes in the cluster, are able to access https://ibm-open-platform.ibm.com.

"Offline installation" on page 9
Choose offline if your Ambari server host does not have internet access.

Important: Analytic Server does not support installation in an environment where the Ambari-Server is running as a non-root user.

Online installation

Choose online installation if your Ambari server host, and all nodes in the cluster, are able to access https://ibm-open-platform.ibm.com.

1. Navigate to the IBM Passport Advantage® Web Site and download the self-extracting binary file specific to your stack, stack version, and hardware architecture to the Ambari Manager node. The available Ambari binaries are:

Table 1. Analytic Server self-extracting binary files

Description	Binary filename
IBM® SPSS® Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	`spss_as-3.1.1.1-hdp2.4-2.6-lx86.bin`

Table 1. Analytic Server self-extracting binary files (continued)

Description	**Binary filename**
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.6 Linux on System p LE English	`spss_as-3.1.1.1-hdp2.6-lppc64.bin`

2. Execute the self-extracting binary file and follow the instructions to view the license, accept the license, choose online installation, and select the installation process for the database type that Analytic Server uses. You are provided the following database type options:
   - New MySQL instance
   - Preexisting MySQL or Db2 instance
3. From the `/var/lib/ambari-server/resources/stacks/<stack-name>/<stack-version>/services/ANALYTICSERVER/package/scripts` directory, run the `update_clientdeps.sh` script with the appropriate arguments (use the `--help` argument for examples).
4. Restart your Ambari server.

   ```
   ambari-server restart
   ```
5. Log on to your Ambari server and install Analytic Server as a service via the Ambari UI.

   Metadata repository

   Analytic Server uses MySQL by default to track information about data sources, projects, and tenants. During installation you need to provide a username (**`metadata.repository.user.name`**) and password **`metadata.repository.password`** used in the JDBC connection between Analytic Server and MySQL. The installer creates the user in the MySQL database but that user is specific to the MySQL database and does not need to be an existing Linux or Hadoop user.

   To change the metadata repository to Db2, follow these steps.

   Note: You cannot change the metadata repository after installation is complete.

   a. Ensure that Db2 is installed on another machine. For more information, see the metadata repository section of the topic Chapter 1, "Prerequisites," on page 1.
   b. In the Ambari Services tab, navigate to the Configs tab of the Analytic Server service.
   c. Open the **Advanced analytics-env** section.
   d. Change the value of **`as.database.type`** from `mysql` to `db2`.
   e. Open the **Advanced analytics-meta** section.
   f. Change the value of **`metadata.repository.driver`** from `com.mysql.jdbc.Driver` to `com.ibm.db2.jcc.DB2Driver`.
   g. Change the value of **`metadata.repository.url`** to `jdbc:db2://{Db2_HOST}:{PORT}/{DBName}:currentSchema={SchemaName};`, where
      - {Db2_HOST} is the hostname of the server where Db2 is installed
      - {PORT} is the port on which Db2 is listening
      - {SchemaName} is an available, unused schema.

      If you are unsure of what values to enter, work with your Db2 administrator.
   h. Supply valid Db2 credentials in **`metadata.repository.user.name`** and **`metadata.repository.password`**.
   i. Click **Save**.

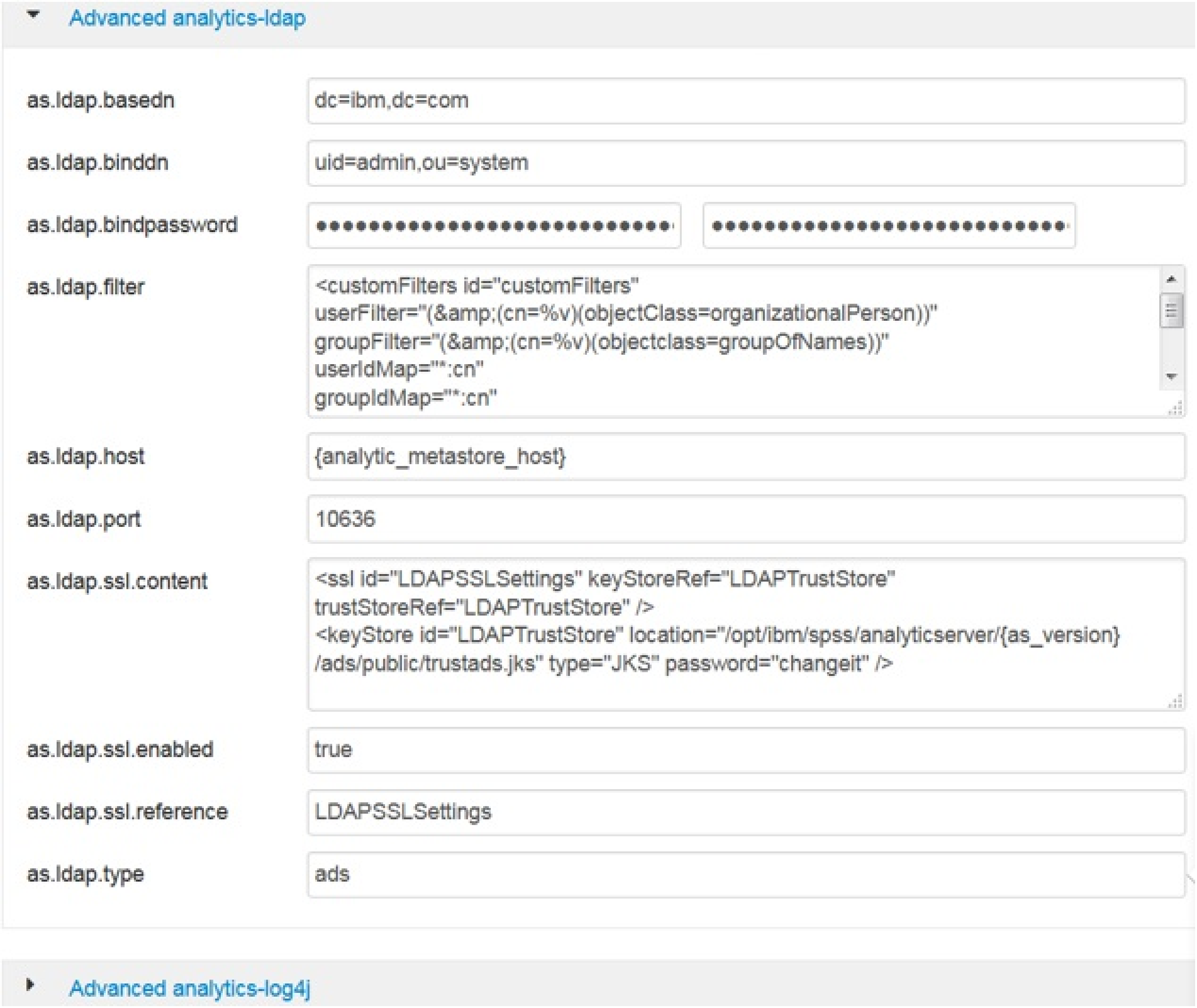
   LDAP configuration

   Analytic Server uses an LDAP server to store and authenticate users and groups. You provide the required LDAP configuration information during Analytic Server installation.

Table 2. LDAP configuration settings

LDAP setting	Description
as.ldap.type	LDAP type. The value can be ads, ad, or openladp. • ads - Apache Directory Server (default setting) • ad - Microsoft Active Directory • openladp - OpenLDAP
as.ldap.host	LDAP host
as.ldap.port	LDAP port number
as.ldap.binddn	LDAP bind DN
as.ldap.bindpassword	LDAP bind DN password
as.ldap.basedn	LDAP base DN
as.ldap.filter	LDAP user and group filter rule
as.ldap.ssl.enabled	Specifies whether to use SSL to communicate between Analytic Server and LDAP. The value can be true or false.
as.ldap.ssl.reference	LDAP SSL reference ID
as.ldap.ssl.content	LDAP SSL configuration

- By default, as.ldap.type is set to ads and the other related settings contain default settings. The exception is you must provide a password for the as.ldap.bindpassword setting. Analytic Server uses the configuration settings to install an Apache Directory Server (ADS) and run the server initialization. The default ADS profile includes the user admin with a password of admin. You can conduct user management through the Analytic Server Console or import user and group information from an XML file via the importUser.sh script that is located in the <Analytic Root>/bin folder.
- If you plan to use an external LDAP server, such as Microsoft Active Directory or OpenLDAP, you must define the configuration settings according to the actual LDAP values. For more information, see Configuring LDAP user registries in Liberty.
- You can change the LDAP configuration after Analytic Server is installed (for example, changing from Apache Directory Server to OpenLDAP). However, if you initially start with Microsoft Active Directory or OpenLDAP, and decide to later switch to Apache Directory Server, Analytic Server will not install an Apache Directory Server during installation. The Apache Directory Server is only installed when it is selected during the initial Analytic Server installation.



▾ Advanced analytics-ldap

as.ldap.basedn	dc=ibm,dc=com
as.ldap.binddn	uid=admin,ou=system
as.ldap.bindpassword	•••••••••••••••••••••••••••• ••••••••••••••••••••••••••••

as.ldap.filter	<customFilters id="customFilters"
	userFilter="(&amp;(cn=%v)(objectClass=organizationalPerson))"
	groupFilter="(&amp;(cn=%v)(objectclass=groupOfNames))"
	userIdMap="*:cn"
	groupIdMap="*:cn"

as.ldap.host	{analytic_metastore_host}
as.ldap.port	10636

as.ldap.ssl.content	<ssl id="LDAPSSLSettings" keyStoreRef="LDAPTrustStore"
	trustStoreRef="LDAPTrustStore" />
	<keyStore id="LDAPTrustStore" location="/opt/ibm/spss/analyticserver/{as_version}
	/ads/public/trustads.jks" type="JKS" password="changeit" />

as.ldap.ssl.enabled	true
as.ldap.ssl.reference	LDAPSSLSettings
as.ldap.type	ads

▸ Advanced analytics-log4j

Figure 1. Example LDAP configuration settings

Configuration settings that should not be changed after installation

Do not change the following settings after installation, or Analytic Server will fail to work.

- Analytic_Server_User
- Analytic_Server_UserID
- as.database.type
- metadata.repository.driver
- distrib.fs.root

6. You now have a functioning instance of Analytic Server. Further configuration is optional. For more information on configuring and administrating Analytic Server, see the topic: "Configuration" on page 14. For information on migrating an existing configuration to a new installation, see the topic: "Migrating IBM SPSS Analytic Server on Ambari" on page 30.
7. Open a web browser and enter the address `http://<host>:<port>/analyticserver/admin/ibm`, where <host> is the address of the Analytic Server host, and <port> is the port that Analytic Server is listening on. By default this is 9080. This URL opens the login dialog for the Analytic Server console. Log in as the Analytic Server administrator. By default this userid is admin and has password admin.

Offline installation

An IBM SPSS Analytic Server offline installation can be done automatically, or manually.

"Automatic Installation on HDP"

The automatic installation process utilizes the Ambari REST API, and is the preferred method for installation.

"Manual installation on HDP (RHEL, SLES)" on page 10

For manually installing Analytic Server on Hortonworks Data Platform

"Manual installation on HDP (Ubuntu)" on page 12

For manually installing Analytic Server on Ubuntu Linux.

Automatic Installation on HDP

The automatic installation process utilizes the Ambari REST API, and is the preferred method for installation.

Important:

- The offline automatic installation procedure installs an embedded Apache Directory Server (ADS). If you want to use a 3rd party LDAP server, you can configure your LDAP settings after the IBM SPSS Analytic Server installation is completed.
- The offline automatic installation procedure can only install a single Analytic Server service instance. You can add more instances after the initial installation is completed.
- The offline automatic installation procedure does not support installing Analytic Server on a Kerberos enabled cluster.

These limitations does not apply to manual HDP or Ubuntu installations.

1. Navigate to the IBM Passport Advantage® Web Site and download the self-extracting binary file to a computer that can access https://ibm-open-platform.ibm.com.

Table 3. Analytic Server self-extracting binary file

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	`spss_as-3.1.1.1-hdp2.4-2.6-lx86.bin`
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.6 Linux on System p LE English	`spss_as-3.1.1.1-hdp2.6-lppc64.bin`

   Important: Analytic Server does not support installation in an environment where the Ambari-Server is running as a non-root user.

2. Run the executable binary that you downloaded in step 1 and specify an offline installation. An offline installation downloads the RPM or DEB files that are required later in the installation process, and should be run on a computer that can access https://ibm-open-platform.ibm.com. The downloaded files are located in the current executable binary directory `./IBM-SPSS-AnalyticServer`.
3. Copy the entire contents of the executable binary directory `./IBM-SPSS-AnalyticServer` from the machine with internet access to the Ambari Manager node (located behind the firewall).
4. On the Ambari Manager node, use the following command to check if the Ambari server is running:

   ```
   ambari-server status
   ```

5. On the Ambari Manager node, and all other nodes on which you want to deploy Analytic Server, install the tool that creates a local `yum` repository.

   ```
   yum install createrepo (RHEL, CentOS)
   ```

   or

   ```
   apt-get install dpkg-dev (Ubuntu)
   ```

6. On the Ambari Manager node, run the executable binary file `./IBM-SPSS-AnalyticServer/packages/spss_as-ambari-offlineinstall.bin`. During the installation, the executable binary verifies that the necessary Analytic Server RPM/DEB files are located in the packages directory. The RPM files that you need depend on your distribution, version, and architecture.

   HDP 2.4, 2.5, and 2.6 (x86_64)

   IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

   IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64.rpm

   HDP 2.6 (PPC64LE)

   IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

   IBM-SPSS-AnalyticServer-3.1.1.1-1.ppc64le.rpm

   HDP 2.4, 2.5, and 2.6 (Ubuntu)

   IBM-SPSS-AnalyticServer-ambari-2.x_3.1.1.1_amd64.deb

   IBM-SPSS-AnalyticServer_1_amd64.deb

During the installation you are prompted to enter the Analytic Server version, JDBC driver, Spark version, Hive version, and so on.

Manual installation on HDP (RHEL, SLES)

The general workflow for a manual offline installation on HDP (RHEL, SLES) is as follows:

1. Navigate to the IBM Passport Advantage® Web Site and download the self-extracting binary file to a computer that can access https://ibm-open-platform.ibm.com.

Table 4. Analytic Server self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	`spss_as-3.1.1.1-hdp2.4-2.6-lx86.bin`
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.6 Linux on System p LE English	`spss_as-3.1.1.1-hdp2.6-lppc64.bin`

2. Run the executable binary that you downloaded in step 1 and specify an offline installation. An offline installation downloads the RPM files that are required later in the installation process, and should be run on a computer that can access https://ibm-open-platform.ibm.com. The downloaded files are located in the current executable binary directory `./IBM-SPSS-AnalyticServer`.
3. Copy the entire contents of the executable binary directory `./IBM-SPSS-AnalyticServer` from the machine with internet access to the Ambari Manager node's `<AS_INSTALLABLE_HOME>` directory (the Ambari Manager node is located behind the firewall).
4. On the Ambari Manager node, use the following command to check if the Ambari server is running:

   ```
   ambari-server status
   ```
5. Install the tool that creates a local `yum` repository.

   ```
   yum install createrepo (RHEL, CentOS)
   ```
   or
   ```
   zypper install createrepo (SLES)
   ```
6. Create a directory that serves as the repository for the Analytic Server RPM files. See the following example.

   ```
   mkdir /home/root/repos/IBM-SPSS-AnalyticServer/x86_64
   ```
7. Copy the necessary Analytic Server RPM files to the new directory. The RPM files that you need depend on your distribution, version, and architecture.

   HDP 2.4, 2.5, and 2.6 (x86_64)

   IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64.rpm

HDP 2.6 (PPC64LE)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.1.1-1.ppc64le.rpm

8. Create the local repository definition. For example, create a file that is named `IBM-SPSS-AnalyticServer-3.1.1.1.repo` in `/etc/yum.repos.d/` (for RHEL, CentOS) or `/etc/zypp/repos.d/` (for SLES) with the following contents.

```
[IBM-SPSS-AnalyticServer]
name=IBM-SPSS-AnalyticServer
baseurl=file:///{path to local repository}
enabled=1
gpgcheck=0
protect=1
```

9. Create the local `yum` repository.

```
createrepo /home/root/repos/IBM-SPSS-AnalyticServer/x86_64 (RHEL, CentOS, SLES)
```

10. From a root user command window, `cd` to the `<AS_INSTALLABLE_HOME>/IBM-SPSS-AnalyticServer` directory, and run `./offLineInstall.sh`. The script reads persisted responses to the binary executable installation command that was previously run, and issues the appropriate platform command (to install the rpm).

Note: Step 11 applies only if you use an externally managed MySQL environment.

11. Run the `add_mysql_user.sh` script on the node/host where the MySQL instance, that will be used as the `AS_MetaStore`, is installed.

    a. Copy the `add_mysql_user.sh` script from `<AS_INSTALLABLE_HOME>/IBM-SPSS-AnalyticServer` to the node/host where the MySQL instance, that will be used as the `AS_MetaStore`.

       - Run the `add_mysql_user.sh` script on the MySQL node/host. For example, `./add_mysql_user.sh -u as_user -p spss -d aedb`

    Notes:

    - The username and password must match the database username and password that was entered for the `AS_Metastore` on the Ambari configuration screen.
    - The `add_mysql_user.sh` script can be manually updated to issue commands (if so desired).
    - When running the `add_mysql_user.sh` script against a secured (root user access) MySQL database, use the `-r` and `-t` parameters to pass in the `dbuserid` and `dbuserid_password`. The script uses `dbuserid` and `dbuserid_password` to perform MySQL operations.

Note: The `metadata.repository.url` setting on the **AS_Configuration** screen (**Advanced analytics-meta**) must be modified to point to the MySQL database host. For example, change the JDBC setting `mysql://{analytic_metastore_host}/aedb?createDatabaseIfNotExist=true` to `mysql://{MySQL_DB}/aedb?createDatabaseIfNotExist=true`

12. Update your Ambari repository file `repoinfo.xml`, typically located in `/var/lib/ambari-server/resources/stacks/$stackName/$stackVersion/repos/`, to use the local `yum` repository, by adding the following lines.

```
<os type="host_os">
   <repo>
        <baseurl>file:///{path to local repository}/</baseurl>
        <repoid>IBM-SPSS-AnalyticServer</repoid>
        <reponame>IBM-SPSS-AnalyticServer-3.1.1.1</reponame>
   </repo>
</os>
```

An example `{path to local repository}` would resemble the following:

```
/home/root/repos/IBM-SPSS-AnalyticServer/x86_64/
```

13. Repeat the following steps for each Ambari non-server cluster node.

a. Copy the entire contents of the appropriate <AS_INSTALLABLE_HOME> directory from the machine with internet access to the Ambari non-server cluster node.

b. Install the tool that creates a local yum repository.

```
yum install createrepo (RHEL, CentOS)
```

or

```
zypper install createrepo (SLES)
```

c. Create a directory that serves as the repository for the Analytic Server RPM files. See the following example.

```
mkdir /home/root/repos/IBM-SPSS-AnalyticServer/x86_64
```

d. Copy the necessary Analytic Server RPM files to the new directory. The RPM files that you need depend on your distribution, version, and architecture.

HDP 2.4, 2.5, and 2.6 (x86_64)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64.rpm

HDP 2.6 (PPC64LE)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.1-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.1.1-1.ppc64le.rpm

e. Create the local repository definition. For example, create a file that is named `IBM-SPSS-AnalyticServer-3.1.1.1.repo` in `/etc/yum.repos.d/` (for RHEL, CentOS) or `/etc/zypp/repos.d/` (for SLES) with the following contents.

```
[IBM-SPSS-AnalyticServer]
name=IBM-SPSS-AnalyticServer
baseurl=file:///{path to local repository}
enabled=1
gpgcheck=0
protect=1
```

f. Create the local yum repository.

```
createrepo /home/root/repos/IBM-SPSS-AnalyticServer/x86_64 (RHEL, CentOS, SLES)
```

14. Continue with step 3 in the "Online installation" on page 5 section.

Manual installation on HDP (Ubuntu)

The general workflow for a manual offline installation on HDP (Ubuntu) is as follows:

1. Navigate to the IBM Passport Advantage® Web Site and download the appropriate Ubuntu self-extracting binary file to a computer that can access https://ibm-open-platform.ibm.com.

Table 5. Analytic Server self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	`spss_as-3.1.1.1-hdp2.4-2.6-lx86.bin`

2. Run the executable binary that you downloaded in step 1 and specify an offline installation. An offline installation downloads the DEB files that are required later in the installation process, and should be run on a computer that can access https://ibm-open-platform.ibm.com. The downloaded files are located in the current executable binary directory `./IBM-SPSS-AnalyticServer`.
3. Copy the entire contents of the executable binary directory `./IBM-SPSS-AnalyticServer` from the machine with internet access to the Ambari Manager node's <AS_INSTALLABLE_HOME> directory (the Ambari Manager node is located behind the firewall).
4. On the Ambari Manager node, use the following command to check if the Ambari server is running:

```
ambari-server status
```

5. Create a `<local_repo>` directory that serves as the repository for the Analytic Server DEB files. For example:

```
mkdir –p /usr/local/mydebs
```

6. Copy the required Analytic Server DEB files to the `<local_repo>` directory.
   - `IBM-SPSS-AnalyticServer-ambari-2.x_3.1.1.1_amd64.deb`
   - `IBM-SPSS-AnalyticServer_1_amd64.deb`
7. Create the local repository.
   a. Install the tool that creates a local repository:
      ```
      apt-get install dpkg-dev
      ```
   b. Generate the source package file:
      ```
      cd <local_repo>
      dpkg-scanpackages . /dev/null | gzip -9c > Packages.gz
      ```
   c. Create the component (main) and architecture (for example, `binary-i386`, `binary-amd64`) of your local repository:
      ```
      mkdir -p <local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-amd64/
      mkdir -p <local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-i386/
      ```
   d. Copy the source package:
      ```
      cp -fr <local_repo>/Packages.gz <local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-amd64/Packages
      cp -fr <local_repo>/Packages.gz <local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-i386/Packages
      ```
8. Create the local repository definition. For example, create a file that is named `IBM-SPSS-AnalyticServer-3.1.1.1.list` in `/etc/apt/sources.list.d` with the following content.
   ```
   deb file:/usr/local/mydebs ./
   ```
9. Run the following command to update the repository list:
   ```
   apt-get update
   ```
10. Run the following command to install Analytic Server 3.1.1.1:
    ```
    apt-get install ./IBM-SPSS-AnalyticServer-ambari-2.x
    ```

    Note: To verify that your local repository is setup correctly, do not run the previous command in your `<local_repo>` directory. If the installation cannot find the package, it means your local repository is not setup correctly (in which case you must verify all previous steps).
11. Repeat the following steps for each Ambari non-server cluster node.
    a. Create a `<local_repo>` directory that serves as the repository for the Analytic Server DEB files. For example:
       ```
       mkdir –p /usr/local/mydebs
       ```
    b. Copy the entire contents of the `<local_repo>` directory from the Ambari Manager node machine to the Ambari non-server cluster node's `<local_repo>` directory. The directory should contain the following files:
       - `<local_repo>/IBM-SPSS-AnalyticServer-ambari-2.x_3.1.1.1_amd64.deb`
       - `<local_repo>/IBM-SPSS-AnalyticServer_1_amd64.deb`
       - `<local_repo>/Packages.gz`
       - `<local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-amd64/Packages`
       - `<local_repo>/dists/IBM-SPSS-AnalyticServer/main/binary-i386/Packages`
    c. Create the local repository definition. For example, create a file that is named `IBM-SPSS-AnalyticServer-3.1.1.1.list` in `/etc/apt/sources.list.d` with the following content.
       ```
       deb file:/usr/local/mydebs ./
       ```
12. Continue with step 3 in the "Online installation" on page 5 section.

Installing Analytic Server against an externally managed MySQL environment

The Analytic Server installation process differs from a normal installation when installing against an externally managed MySQL environment.

The following steps explain the process of installing Analytic Server against an externally managed MySQL environment.

1. Navigate to the IBM Passport Advantage® Web Site and download the self-extracting binary file specific to your stack, stack version, and hardware architecture to a host within the Ambari cluster.
2. Execute the self-extracting binary file and follow the instructions to (optionally) view the license, accept the license.
   a. Choose the online option.
   b. Select the **External MySQL Database** option when prompted.
3. Copy the `add_mysql_user.sh` script from `<AS_INSTALLABLE_HOME>/IBM-SPSS-AnalyticServer` to the node/host where the MySQL instance, that will be used as the `AS_MetaStore`.
   - Run the `add_mysql_user.sh` script on the MySQL node/host. For example, `./add_mysql_user.sh -u as_user -p spss -d aedb`

   Notes:

   - The username and password must match the database username and password that was entered for the `AS_Metastore` on the Ambari configuration screen.
   - The `add_mysql_user.sh` script can be manually updated to issue commands (if so desired).
   - When running the `add_mysql_user.sh` script against a secured (root user access) MySQL database, use the `-r` and `-t` parameters to pass in the `dbuserid` and `dbuserid_password`. The script uses `dbuserid` and `dbuserid_password` to perform MySQL operations.
4. Restart your Ambari server.
   ```
   ambari-server restart
   ```
5. From the Ambari console, add the `AnalyticServer` service as normal (enter the same database username and password as entered in step 3).

   Note: The `metadata.repository.url` setting on the **AS_Configuration** screen (**Advanced analytics-meta**) must be modified to point to the MySQL database host. For example, change the JDBC setting `mysql://{analytic_metastore_host}/aedb?createDatabaseIfNotExist=true` to `mysql://{MySQL_DB}/aedb?createDatabaseIfNotExist=true`

Configuration

After installation, you can optionally configure and administer Analytic Server through the Ambari UI.

Note: The following conventions are used for Analytic Server file paths.

- {AS_ROOT} refers to the location where Analytic Server is deployed; for example, `/opt/IBM/SPSS/AnalyticServer/3.1`.
- {AS_SERVER_ROOT} refers to the location of the configuration, log, and server files; for example, `/opt/IBM/SPSS/AnalyticServer/3.1/ae_wlpserver/usr/servers/aeserver`.
- {AS_HOME} refers to the location on HDFS that is used by Analytic Server as a root folder.

Security

Configuring an LDAP registry

The LDAP registry allows you to authenticate users with an external LDAP server such as Active Directory or OpenLDAP.

Important: An LDAP user must be designated as an Analytic Server administrator in Ambari.

Here is an example of an ldapRegistry for OpenLDAP.

```
<ldapRegistry
    baseDN="ou=people,dc=aeldap,dc=org"
    ldapType="Custom"
    port="389"
    host="server"
    id="OpenLDAP"
    bindDN="cn=admin,dc=aeldap,dc=org"
    bindPassword="{xor}Dz4sLG5tbGs="
    searchTimeout="300000m"
    recursiveSearch="true">
    <customFilters
        id="customFilters"
        userFilter="(&amp;(uid=%v)(objectClass=inetOrgPerson))"
        groupFilter="(&amp;(cn=%v)(|(objectclass=organizationalUnit)))"
        groupMemberIdMap="posixGroup:memberUid"/>
</ldapRegistry>
```

The following example provides Analytic Server authentication with Active Directory:

```
<ldapRegistry id="Microsoft Active Directory" realm="ibm"
  host="host"
  port="389"
  baseDN="cn=users,dc=adtest,dc=mycompany,dc=com"
  bindDN="cn=administrator,cn=users,dc=adtest,dc=mycompany,dc=com"
  bindPassword ="adminpassword"
  ldapType="Custom"
    <customFilters
    userFilter="(&amp;(sAMAccountName=%v)(objectcategory=user))"
    groupFilter="(&amp;(cn=%v)(objectcategory=group))"
    userIdMap="user:sAMAccountName"
    groupIdMap="*:cn"
    groupMemberIdMap="memberOf:member" />
</ldapRegistry>
```

Note: It is often helpful to use an LDAP viewer third party tool to verify the LDAP configuration.

The following example provides WebSphere Liberty profile authentication with Active Directory:

```
<ldapRegistry id="ldap" realm="SampleLdapADRealm"
    host="ldapserver.mycity.mycompany.com" port="389" ignoreCase="true"
    baseDN="cn=users,dc=adtest,dc=mycity,dc=mycompany,dc=com"
    bindDN="cn=testuser,cn=users,dc=adtest,dc=mycity,dc=mycompany,dc=com"
    bindPassword="testuserpwd"
    ldapType="Microsoft Active Directory"
    sslEnabled="true"
    sslRef="LDAPSSLSettings">
    <activedFilters
       userFilter="(&amp;(sAMAccountName=%v)(objectcategory=user))"
       groupFilter="(&amp;(cn=%v)(objectcategory=group))"
       userIdMap="user:sAMAccountName"
       groupIdMap="*:cn"
       groupMemberIdMap="memberOf:member" >
    </activedFilters>
</ldapRegistry>

<ssl id="LDAPSSLSettings" keyStoreRef="LDAPKeyStore" trustStoreRef="LDAPTrustStore" />

<keyStore id="LDAPKeyStore" location="${server.config.dir}/LdapSSLKeyStore.jks"
          type="JKS" password="{xor}CDo9Hgw=" />

<keyStore id="LDAPTrustStore" location="${server.config.dir}/LdapSSLTrustStore.jks"
          type="JKS" password="{xor}CDo9Hgw=" />
```

Notes:

- Support for LDAP in Analytic Server is controlled by WebSphere Liberty. For more information, see Configuring LDAP user registries in Liberty.

- When LDAP is secured with SSL, follow the instructions in the following "Configure a secure socket layer (SSL) connection from Analytic Server to LDAP" section.

Configuring a secure socket layer (SSL) connection from Analytic Server to LDAP

If you select the Apache Directory Server (`ads`) LDAP option during Analytic Server installation, and use the default configuration, the Apache Directory Server ins installed with SSL configured and enabled (Analytic Server will automatically use SSL to communicate with the Apache Directory Server).

Configure SSL using the following steps when one of the other LDAP options is selected during Analytic Server installation (for example, when using an external LDAP server).

1. Login to each of the Analytic Server machines as the Analytic Server user and create a common directory for SSL certificates.

   Note: By default, as_user is the Analytic Server user; see **Service accounts** under the Admin tab in the Ambari console.
2. Copy the key store and trust store files to some common directory on all Analytic Server machines. Also add the LDAP client CA certificate to the trust store. Below are some sample instructions.

   ```
   mkdir /home/as_user/security
   cd /home/as_user/security
   openssl s_client -connect <ldap-hostname>:636 -showcerts > client.cert
   $JAVA_HOME/bin/keytool -import -file ./client.cert -alias ldapCA -keystore mytrust.jks
   password : changeit
   ```

   Note: JAVA_HOME is the same JRE used for Analytic Server startup.
3. Passwords can be encoded to obfuscate their values with the securityUtility tool, which is in `{AS_ROOT}/ae_wlpserver/bin`. An example follows.

   ```
   securityUtility encode changeit
            {xor}PDc+MTg6Nis=
   ```
4. Login to the Ambari console and update the Analytic Server configuration setting **`ssl.keystore.config`** with the correct SSL configuration settings. An example follows.

   ```
   <ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" trustStoreRef="defaultTrustStore"
         clientAuthenticationSupported="true"/>
             <keyStore id="defaultKeyStore" location="/home/as_user/security/mykey.jks" type="JKS"
                       password="{xor}Ozo5PiozKxYdEgwPDAweDG1uDz4sLCg7"/>
             <keyStore id="defaultTrustStore" location="/home/as_user/security/mytrust.jks" type="JKS"
                       password="{xor}PDc+MTg6Nis="/>
   ```

   Note: Use the absolute path for key and trust store files.
5. Update the Analytic Server configuration setting **`security.config`** with the correct LDAP configuration settings. For example, in the **`ldapRegistry`** element, set the **`sslEnabled`** attribute to `true` and the **`sslRef`** attribute to `defaultSSLConfig`.

Configuring Kerberos

Analytic Server supports Kerberos using Ambari.

Note: IBM SPSS Analytic Server does not support Kerberos Single-Sign-On (SSO) when it is used in conjunction with Apache Knox.

1. Create accounts in the Kerberos user repository for all users you plan to give access to Analytic Server.
2. Create the same accounts (from the previous step) on the LDAP server.
3. Create an OS user account for each of the users created in the previous step on each and every Analytic Server node and Hadoop node.
   - Make sure that the UID for these users matches on all machines. You can test this using the kinit command to log in to each of the accounts.

- Make sure that the UID adheres to the "Minimum user ID for submitting job" Yarn setting. This is the **min.user.id** parameter in container-executor.cfg. For example, if **min.user.id** is 1000, then each user account created must have a UID greater than or equal to 1000.

4. Create a user home folder on HDFS for all principals in Analytic Server. For example, if you add testuser1 to the Analytic Server system, then create a home folder like `/user/testuser1` on HDFS and ensure that testuser1 has read and write permissions to this folder.
5. [Optional] If you plan to use HCatalog data sources and Analytic Server is installed on a different machine from the Hive metastore, you need to impersonate the Hive client on HDFS.
   a. Navigate to the Configs tab of the HDFS service in the Ambari console.
   b. Edit the **hadoop.proxyuser.hive.groups** parameter to have the value `*`, or a group that contains all of the users allowed to log in to Analytic Server.
   c. Edit the **hadoop.proxyuser.hive.hosts** parameter to have the value `*`, or the list of hosts on which the Hive metastore and every instance of Analytic Server are installed as services.
   d. Restart the HDFS service.

After these steps have been performed and Analytic Server is installed, Analytic Server silently and automatically configures Kerberos.

Configuring HAProxy for Single Sign On (SSO) using Kerberos

1. Configure and start HAProxy per the HAProxy documentation guide: http://www.haproxy.org/#docs
2. Create the Kerberos principle (HTTP/<proxyHostname>@<realm>) and keytab file for the HAProxy host, where <proxyHostname> is the full name of the HAProxy host, and <realm> is the Kerberos realm.
3. Copy the keytab file to each of the Analytic Server hosts as `/etc/security/keytabs/spnego_proxy.service.keytab`
4. Update permissions to this file on each of the Analytic Server hosts. An example follows.

   ```
   chown root:hadoop /etc/security/keytabs/spnego_proxy.service.keytab
   chmod 440 /etc/security/keytabs/spnego_proxy.service.keytab
   ```

5. Open the Amabri console and update the following properties in the Analytic Server 'Custom analytics.cfg' section.

   ```
   web.authentication.kerberos.keytab=/etc/security/keytabs/spnego_proxy.service.keytab
   web.authentication.kerberos.principal=HTTP/<proxy machine full name>@<realm>
   ```

6. Save the configuration and restart all Analytic Server services from the Amabri console.

Now users are able to log in to Analytic Server using Kerberos SSO.

Enabling Kerberos impersonation

Impersonation allows a thread to execute in a security context that differs from the security context of the process that owns the thread. For example, impersonation provides a means for Hadoop jobs to run as users other than the standard Analytic Server user (`as_user`). To enable Kerberos impersonation:

1. Add impersonation configuration attributes to HDFS (or the Hive service configurations) when running in a Kerberos enabled cluster. In the case of HDFS, the following properties must be added to the HDFS `core-site.xml` file:

   ```
   hadoop.proxyuser.<analytic_server_service_principal_name>.hosts = *
   hadoop.proxyuser.<analytic_server_service_principal_name>.groups = *
   ```

   where `<analytic_server_service_principal_name>` is the default `as_user` value that is specified in the Analytic Server configuration's `Analytic_Server_User` field.

   The following properties must also be added to the HDFS `core-site.xml` file in cases where data is accessed from HDFS via Hive/HCatalog:

```
hadoop.proxyuser.hive.hosts = *
hadoop.proxyuser.hive.groups = *
```

2. If Analytic Server is configured to use a user name other than `as_user`, you must modify the property names to reflect the other user name (for example, `hadoop.proxyuser.xxxxx.hosts`, where `xxxxx` is the configured user name that is specified in the Analytic Server configuration).
3. Run the following command from a command shell on the Analytic Server node:

   ```
   hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
   ```

Disabling Kerberos

1. Disable Kerberos in the Ambari console.
2. Stop the Analytic Server service.
3. Remove the following parameters from Custom `analytics.cfg`.

   ```
   default.security.provider
   hdfs.keytab
   hdfs.user
   java.security.krb5.conf
   as.db.connect.method
   web.authentication.kerberos.keytab
   web.authentication.kerberos.principal
   ```
4. Click **Save** and restart the Analytic Server service.

Enabling Secure Socket Layer (SSL) connections to the Analytic Server console

By default, Analytic Server generates self-signed certificates to enable Secure Socket Layer (SSL), so you can access the Analytic Server console through the secure port by accepting self signed certificates. In order to make HTTPS access more secure, you need to install 3rd party vendor certificates.

To install 3rd party vendor certificates, follow these steps.

1. Copy the 3rd party vendor key store and trust store certificates to the same directory in all Analytic Server nodes; for example, `/home/as_user/security`.

   Note: The Analytic Server User must have read access to this directory.
2. In the Ambari Services tab, navigate to the Configs tab of the Analytic Server service.
3. Edit the **`ssl.keystore.config`** parameter.

   ```
   <ssl id="defaultSSLConfig"
        keyStoreRef="defaultKeyStore"
        trustStoreRef="defaultTrustStore"
        clientAuthenticationSupported="true"/>
   <keyStore id="defaultKeyStore"
             location="<KEYSTORE-LOCATION>"
             type="<TYPE>"
             password="<PASSWORD>"/>
   <keyStore id="defaultTrustStore"
             location="<TRUSTSTORE-LOCATION>"
             type="<TYPE>"
             password="<PASSWORD>"/>
   ```

   Replace

   - <KEYSTORE-LOCATION> with the absolute location of the key store; for example: `/home/as_user/security/mykey.jks`
   - <TRUSTSTORE-LOCATION> with the absolute location of the trust store; for example: `/home/as_user/security/mytrust.jks`
   - <TYPE> with the type of the certificate; for example: JKS, PKCS12 etc.
   - <PASSWORD> with the encrypted password in Base64 encryption format. For encoding you can use the securityUtility; for example: `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/bin/securityUtility encode <password>`

If you want to generate a self-signed certificate, you can use securityUtility; for example: `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/bin/securityUtility createSSLCertificate --server=myserver --password=mypassword --validity=365 --subject=CN=mycompany,O=myOrg,C=myCountry`.

Note: You must provide an appropriate host domain name for the `CN` value.

For more information on securityUtility and other SSL settings, refer to the WebSphere Liberty Profile documentation.

4. Click **Save** and restart the Analytic Server service.

Enabling Support for Essentials for R

Analytic Server supports scoring R models and running R scripts.

To configure support for R after a successful Analytic Server installation:

1. Provision the server environment for Essentials for R.

RedHat Linux x86_64

Run the following commands:

```
yum update
yum install -y zlib zlib-devel
yum install -y bzip2 bzip2-devel
yum install -y xz xz-devel
yum install -y pcre pcre-devel
yum install -y libcurl libcurl-devel
```

Ubuntu Linux

Run the following commands:

```
apt-get update
apt-get install -y zlib1g-dev
apt-get install -y libreadline-dev
apt-get install -y libxt-dev
apt-get install -y bzip2
apt-get install -y libbz2-dev
apt-get install -y liblzma-dev
apt-get install -y libpcre3 libpcre3-dev
apt-get install -y libcurl4-openssl-dev
apt-get install -y liblzma-dev
apt-get install -y libpcre3 libpcre3-dev
apt-get install -y libcurl4-openssl-dev
```

SUSE Linux

Essentials for R installation on SUSE requires compatible FORTRAN which is not normally available in the configured ZYPPER repositories (it is available only from the SUSE SDK media). As a result, running an Ambari installation for Essentials for R on SUSE server will fail as it will not be able to install FORTRAN. Use the following steps to provision on SUSE:

a. Install GCC C++.

```
zypper install gcc-c++
```

b. Install GCC FORTRAN. The required RPM files can be copied from the SUSE SDK media and must be installed in the following order.

```
zypper install libquadmath0-4.7.2_20130108-0.19.3.x86_64.rpm
zypper install libgfortran3-4.7.2_20130108-0.19.3.x86_64.rpm
zypper install gcc43-fortran-4.3.4_20091019-0.37.30.x86_64.rpm
zypper install gcc-fortran-4.3-62.200.2.x86_64.rpm
```

c. Run the following command to install the Essentials for R libraries.

```
R_PREFIX=/opt/ibm/spss/R
cd $R_PREFIX
rm -fr $R_PREFIX/r_libs
mkdir -p $R_PREFIX/r_libs
cd $R_PREFIX/r_libs
wget https://zlib.net/fossils/zlib-1.2.11.tar.gz --no-check-certificate
tar zxvf zlib-1.2.11.tar.gz
cd zlib-1.2.11/
./configure
make && make install
cd $R_PREFIX/r_libs
wget http://www.bzip.org/1.0.6/bzip2-1.0.6.tar.gz
tar xzvf bzip2-1.0.6.tar.gz
cd bzip2-1.0.6
sed "s|^CC=gcc|CC=gcc -fPIC|" -i ./Makefile
make -f Makefile-libbz2_so
make clean
make
make install
cd $R_PREFIX/r_libs
wget https://tukaani.org/xz/xz-5.2.3.tar.gz
tar xzvf xz-5.2.3.tar.gz
cd xz-5.2.3
./configure
make -j3
make install
cd $R_PREFIX/r_libs
wget http://ftp.pcre.org/pub/pcre/pcre-8.38.tar.gz
tar xzvf pcre-8.38.tar.gz
cd pcre-8.38
./configure --enable-utf8
make
make install
cd $R_PREFIX/r_libs
wget https://www.openssl.org/source/openssl-1.0.2l.tar.gz --no-check-certificate
tar zxvf openssl-1.0.2l.tar.gz
cd openssl-1.0.2l/
./config shared
make
make install
echo '/usr/local/ssl/lib' >> /etc/ld.so.conf
ldconfig
cd $R_PREFIX/r_libs
wget --no-check-certificate https://curl.haxx.se/download/curl-7.50.1.tar.gz
tar xzvf curl-7.50.1.tar.gz
cd curl-7.50.1
./configure --with-ssl
make -j3
make install
cd $R_PREFIX/r_libs
wget ftp://rpmfind.net/linux/opensuse/distribution/12.3/repo/oss/suse/x86_64/libgomp1-4.7.2_20130108-2.1.6.x86_64.rpm --no-check-certificate
rpm -ivh libgomp1-4.7.2_20130108-2.1.6.x86_64.rpm
```

2. Download the self-extracting archive (BIN) for IBM SPSS Modeler Essentials for R RPM or DEB. Essentials for R is available for download (https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp). Choose the file specific to your stack, stack version, and hardware architecture.
3. Execute the self-extracting binary file and follow the instructions to (optionally) view the license, accept the license, and choose online or offline installation.

   Online installation

   Choose online installation if your Ambari server host, and all nodes in the cluster, are able to access https://ibm-open-platform.ibm.com.

   Offline installation

   Choose offline if your Ambari server host does not have internet access. Offline installation will download the necessary RPM files, and should be run on a machine that can access https://ibm-open-platform.ibm.com. The RPM files can then be copied to the Ambari server host.

   a. Copy the necessary Essentials for R RPM or DEB files to any location on your Ambari server host. The RPM/DEB files you need depend on your distribution, version, and architecture, shown below.

      HDP 2.4, 2.5, and 2.6 (x86_64)

      IBM-SPSS-ModelerEssentialsR-ambari-2.1-HDP-2.6-8.5.0.0-1.x86_64.rpm

      HDP 2.6 (PPC64LE)

      IBM-SPSS-ModelerEssentialsR-ambari-2.1-HDP-2.6-8.5.0.0-1.ppc64le.rpm

      HDP 2.4, 2.5, and 2.6 (Ubuntu)

      IBM-SPSS-ModelerEssentialsR-ambari-3.1.1.1_3.1.1.1_amd64.deb

   b. Install the RPM or DEB. In the following example, the command installs Essentials for R on HDP 2.6 (x86_64).

      ```
      rpm -i IBM-SPSS-ModelerEssentialsR-ambari-2.1-HDP-2.6-8.5.0.0-1.x86_64.rpm
      ```

      In the following example, the command installs Essentials for R on HDP 2.4 (Ubuntu).

      ```
      dpkg -i IBM-SPSS-ModelerEssentialsR-ambari-3.1.1.1_3.1.1.1_amd64.deb
      ```

4. Restart your Ambari server.

   ```
   ambari-server restart
   ```

5. Log on to your Ambari server and install SPSS Essentials for R as a service via the Ambari console. SPSS Essentials for R should be installed on every host where Analytic Server and the Analytic Metastore is installed.

   Note: Ambari will attempt to install gcc-c++ and gcc-gfortran (RHEL) and gcc-fortran (SUSE) prior to installing R. These packages are declared as dependencies on R's Ambari service definition. Ensure that the servers where R is to be installed and executed are configured to download gcc-c++ and gcc-[g]fortran RPMs or have GCC and FORTRAN compilers installed. If the installation of Essentials for R fails, install these packages manually prior to installing Essentials for R.
6. Refresh the Analytic Server service.
7. Run the `update_clientdeps` script using the instructions in "Updating client dependencies" on page 25.
8. You must also install Essentials for R on the machine that hosts SPSS Modeler Server. See the SPSS Modeler documentation for details.

Enabling relational database sources

Analytic Server can use relational database sources if you supply the JDBC drivers in a shared directory on each Analytic Server host. By default, this directory is `/usr/share/jdbc`.

To change the shared directory, follow these steps.
1. In the Ambari Services tab, navigate to the Configs tab of the Analytic Server service.
2. Open the **Advanced analytics.cfg** section.
3. Specify the path of the shared directory of JDBC drivers in **`jdbc.drivers.location`**.
4. Click **Save**.
5. Stop the Analytic Server service.
6. Click **Refresh**.
7. Start the Analytic Server service.

Table 6. Supported databases

Database	Supported versions	JDBC driver jars	Vendor
Amazon Redshift	8.0.2 or later	`RedshiftJDBC41-1.1.6.1006.jar` or later	Amazon
BigSQL	4.1.0.0 or later	db2jcc.jar	IBM
DashDB	Bluemix Service	`db2jcc.jar`	IBM
Db2 for Linux, UNIX, and Windows	11.1, 10.5, 10.1, 9.7	`db2jcc.jar`	IBM
Db2 z/OS	11, 10	`db2jcc.jar`, `db2_license_cisuz.jar`	IBM
Greenplum	5	postgresql.jar	Greenplum
Hive	1.2, 2.1	`hive-jdbc-*.jar`	Apache
MySQL	5.6, 5.7	mysql-connector-java-commercial-5.1.25-bin.jar	MySQL
Netezza	7, 6.x	`nzjdbc.jar`	IBM
Oracle	12c, 11g R2 (11.2)	`ojdbc6.jar`, `orai18n.jar`	Oracle
SQL Server	2014, 2012, 2008 R2	`sqljdbc4.jar`	Microsoft
Teradata	15, 15.1	`tdgssconfig.jar`, `terajdbc4.jar`	Teradata

Notes

- If you created a Redshift data source prior to installing Analytic Server, you need perform the following steps in order to use the Redshift data source.
  1. In the Analytic Server console, open the Redshift data source.
  2. Select the `Redshift` database data source.
  3. Enter the Redshift server address.
  4. Enter the database name and username. The password should automatically populate.
  5. Select the database table.
- BigSQL is the IBM SQL interface for the Apache Hadoop environment. BigSQL is not a relational database, but Analytic Server support access to it via JDBC (the JDBC jar file is the same as what is used for Db2).

  A common usage for BigSQL with Analytic Server is accessing BigSQL Hadoop/HBase tables via an HCatalog data source.

Enabling HCatalog data sources

Analytic Server provides support for a number of data sources through Hive/HCatalog. Some sources require manual configuration steps.

1. Collect the necessary JAR files to enable the data source. No additional steps are necessary to enable support for Apache HBase and Apache Accumulo. For other NoSQL data sources, contact the database vendor and obtain the storage handler and related jars. For information on supported HCatalog data sources, see the "Using HCatalog data sources" section in the IBM SPSS Analytic Server 3.1.1.1 User's guide.
2. Add these JAR files to the `{HIVE_HOME}/auxlib` directory and to the `/usr/share/hive` directory on each Analytic Server node.
3. Restart the Hive Metastore service.
4. Refresh the Analytic Metastore service.
5. Restart each and every instance of the Analytic Server service.

Notes:

- The Analytic Server Metastore cannot be installed on the same machine as the Hive Metastore.
- When accessing HBase data via an Analytic Server HCatalog data source, the accessing user must have read permission for the HBase tables.
  - In non-kerberos environments, Analytic Server accesses HBase using `as_user` (`as_user` must have read permission for HBase).
  - In kerberos environments, both `as_user` and the login user must have read permission for HBase tables.

NoSQL databases

Analytic Server supports any NoSQL database for which a Hive storage handler is available from the vendor.

No additional steps are necessary to enable support for Apache HBase and Apache Accumulo.

For other NoSQL databases, contact the database vendor and obtain the storage handler and related jars.

File-based Hive tables

Analytic Server supports any file-based Hive tables for which a built-in or custom Hive SerDe (serializer-deserializer) is available.

The Hive XML SerDe for processing XML files is located in the Maven Central Repository at http://search.maven.org/#search%7Cga%7C1%7Chivexmlserde.

MapReduce v2 jobs

Use the `preferred.mapreduce` setting in the Analytic Server **Custom analytic.cfg** section to control how MapReduce jobs are handled:

Table 7. Custom analytics.cfg properties

Property	Description
`preferred.mapreduce`	Controls the method in which MapReduce jobs are run. Valid values include: • `spark` • `m3r` • `hadoop` For example: `preferred.mapreduce=spark`

Apache Spark

If you want to use Spark (version 1.5 or later), you must manually add the `spark.version` property during Analytic Server installation.

1. Open the Amabri console and add the following property in the Analytic Server **Advanced analytics.cfg** section.
   - **Key**: `spark.version`
   - **Value**: Enter the appropriate Spark version number (for example, `1.x`, `2.x`, or `None`).
2. Save the configuration.

Note: You can force HCatalog to never use Spark via a custom `analytics.cfg` setting.

1. Open the Amabri console and add the following property in the Analytic Server **Custom analytic.cfg** section.
   - **Key**: `spark.hive.compatible`
   - **Value**: `false`

Changing ports used by Analytic Server

Analytic Server uses the 9080 port for HTTP and the 9443 port for HTTPS by default. To change the port settings, follow these steps.

1. In the Ambari Services tab, navigate to the Configs tab of the Analytic Server service.
2. Open the **Advanced analytics.cfg** section.
3. Specify the desired HTTP and HTTPS ports in **`http.port`** and **`https.port`**, respectively.
4. Click **Save**.
5. Restart the Analytic Server service.

High availability Analytic Server

You can make Analytic Server highly available by adding it as a service to multiple nodes in your cluster.

1. In the Ambari console, navigate to the Hosts tab.
2. Select a host that is not already running Analytic Server as a service.
3. On the Summary tab, click **Add** and select Analytic Server.
4. Click **Confirm Add**

Multiple-cluster support

The multiple-cluster feature is an enhancement to the High-Availability capability of IBM SPSS Analytic Server, and provides improved isolation in multiple-tenant environments. By default, installation of the Analytic Server service (in either Ambari or ClouderaManager) results in the definition of a single analytic server cluster.

The cluster specification defines the Analytic Server cluster membership. Modifying the cluster specification, is accomplished with XML content (in the Ambari Analytic Server configuration's `analytics-cluster` field or by manually editing the Cloudera Manager's `configuration/analytics-cluster.xml` file). When configuring multiple Analytic Server clusters, it is necessary to feed requests to each Analytic Server cluster with its own load balancer.

Using the multiple-cluster feature assures that work for one tenant cannot negatively impact work being performed in another tenant's cluster. With respect to highly available jobs, job failover occurs only within the scope of the Analytic Server cluster upon which the work was initiated. The following example provides a multiple-cluster XML specification.

Note: Analytic Server can be made highly available by adding it as a service to multiple nodes in your cluster.

```
<analayticServerClusterSpec>
    <cardinality>1+</cardinality>
    <cluster name="cluster1">
        <memberName>one.cluster</memberName>
        <memberName>two.cluster</memberName>
    </cluster>
    <cluster name="cluster2">
        <memberName>three.cluster</memberName>
        <memberName>four.cluster</memberName>
    </cluster>
</analayticServerClusterSpec>
```

In the previous example, two load balancers are required. One load balancer sends requests to the `cluster1` members (`one.cluster` and `two.cluster`) and the other sends requests to `cluster2` members (`three.cluster` and `four.cluster`).

The following example provides a single cluster XML specification (the default configuration).

```
<analayticServerClusterSpec>
    <cardinality>1</cardinality>
    <cluster name="cluster1">
        <memberName>*</memberName>
    </cluster>
</analayticServerClusterSpec>
```

In the previous example, a single load balancer is required to handle cases where there is more than one configured cluster member.

Notes

- Only singleton clusters support the use of wildcards in the **`memberName`** element (for example, cluster cardinality = "1"). Valid values for the cardinality element are `1` and `1+`.
- The **`memberName`** must be specified in the same manner as the host name to which the Analytic Server role is assigned.
- All servers in all clusters must be restarted after the cluster configuration changes are applied.
- In Cloudera Manager, you must modify and maintain the `analytics-cluster.xml` file on all Analytic Server nodes. All nodes must be maintained to ensure that they contain the same content.

Optimizing JVM options for small data

You can edit JVM properties in order to optimize your system when running small (M3R) jobs.

In the Ambari console, see the Advanced analytics-jvm-options section of the Configs tab in the Analytic Server service. Modifying the following parameters sets the heap size for jobs run on the server that hosts Analytic Server; that is, not Hadoop. This is important if running small (M3R) jobs, and you may need to experiment with these values to optimize your system.

```
-Xms512M
-Xmx2048M
```

Updating client dependencies

This section describes how to update the Analytic Server service's dependencies using the `update_clientdeps` script.

1. Login to Ambari server host as root.
2. Change directory to `/var/lib/ambari-server/resources/stacks/<stack-name>/<stack-version>/services/ANALYTICSERVER/package/scripts`; see the following example.

   ```
   cd "/var/lib/ambari-server/resources/stacks/HDP/2.4/services/ANALYTICSERVER/package/scripts"
   ```
3. Run the `update_clientdeps` script with the following arguments.

   -u <ambari-user>
   The Ambari account username

   -p <ambari-password>
   The password for the Ambari account user.

   -h <ambari-host>
   The hostname of the Ambari server.

   -x <ambari-port>
   The port on which Ambari is listening.

   See the following example.

   ```
   ./update_clientdeps.sh -u admin -p admin -h host.domain -x 8080
   ```
4. Restart the Ambari server using the following command.

   ```
   ambari-server restart
   ```

Configuring Apache Knox

The Apache Knox Gateway is a system that provides a single point of secure access for Apache Hadoop services. The system simplifies Hadoop security for both users (who access the cluster data and run jobs) and operators (who control access and manage the cluster). The Gateway runs as a server (or cluster of servers) that serve one or more Hadoop clusters.

Note: IBM SPSS Analytic Server does not support Apache Knox when it is used in conjunction with Kerberos Single-Sign-On (SSO).

The Apache Knox Gateway effectively hides the Hadoop cluster topology details and integrates with Enterprise LDAP and Kerberos. The following sections provide information on the required Apache Knox and Analytic Server configuration tasks.

Prerequisites

- A known Apache Knox issue does not propagate the security information that is contained in HTTP cookies and headers (for more information, see https://issues.apache.org/jira/browse/KNOX-895). The issue is resolved in Knox 0.14.0 (or later). You must obtain an updated Hortonworks distribution, that includes Knox 0.14.0 (or later), before Knox with work with Analytic Server. Contact your Hortonworks provider for more information.
- The Analytic Server nodes must connect with the Knox server with a passwordless SSH connection. The passwordless SSH connection moves from Analytic Server to Knox (**Analytic Server** > **Knox**).
- Analytic Server must be installed after the Knox service is installed.

In some cases, unexpected issues result in the configuration files not being automatically copied. In these cases you must manually copy the following configuration files:

- `com.ibm.spss.knox_0.6-3.1.1.1.jar`: The file must be copied from the Analytic Server location:

  `<Analytic_Server_Installation_Path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib`

  to the Knox server node:

  `/KnoxServicePath/ext`

  For example: `/usr/iop/4.1.0.0/knox/ext`
- `rewrite.xml` and `service.xml`: The files must be copied from the Analytic Server location:.

  `<Analytic_Server_Installation_Path>/ae_wlpserver/usr/servers/aeserver/configuration/knox`

  to the Knox server node:

  `/KnoxServicePath/data/services`

  For example: `/usr/iop/4.1.0.0/knox/data/services`

  Note: There are two sets of `rewrite.xml` and `service.xml` files (one set for `http://rest` traffic and one set for `ws://websocket` traffic). Copy all of the `rewrite.xml` and `service.xml` files for both `analyticserver` and `analyticserver_ws` to the Knox server node.

Configuring Ambari

The Analytic Server service must be configured in the Ambari user interface:

1. In the Ambari user interface, navigate to **Knox** > **Configs** > **Advanced topology**. The current Knox configuration settings display in the **content** window.
2. Add the following two services to the **Advanced topology** section in the Knox configuration:

```
<service>
    <role>ANALYTICSERVER</role>
    <url>http://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
 </service>
<service>
    <role>ANALYTICSERVER_WS</role>
    <url>ws://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
 </service>
```

   `{analyticserver-host}` and `{analyticserver-port}` must be replaced with the appropriate Analytic Server server name and port number:

   - The `{analyticserver-host}` URL can be found in the Ambari user interface (**SPSS Analytic Server** > **Summary** > **Analytic Server**).
   - The `{analyticserver-port}` number can be found in the Ambari user interface (**SPSS Analytic Server** > **Configs** > **Advanced analytics.cfg** > **http.port**).

   Note: When Analytic Server is deployed to multiple nodes, and LoadBalancer is used, the `{analyticserver-host}` and `{analyticserver-port}` must correspond to the LoadBalancer URL and port number.
3. Restart the Knox service.

When LDAP is used, Knox defaults to the provided "Demo" LDAP. You can change to an enterprise LDAP server (such as Microsoft LDAP or OpenLDAP).

Configuring Analytic Server

To use LDAP for Analytic Server, the Analytic Server must be configured to use the same LDAP server that is used by Apache Knox. The `<value>` entries for the following Ambari settings must be updated to reflect the appropriate Knox LDAP server settings:

- `main.ldapRealm.userDnTemplate`

- main.ldapRealm.contextFactory.url

The values are available in the Ambari user interface at: **Knox** > **Configs** > **Advanced topology**. For example:

```
<param>
   <name>main.ldapRealm.userDnTemplate</name>
   <value>uid={0},ou=people,dc=hadoop,dc=apache,dc=org</value>
</param>
<param>
   <name>main.ldapRealm.contextFactory.url</name>
   <value>ldap://{{knox_host_name}}:33389</value>
</param>
```

Restart the Knox service after updating the Knox LDAP settings.

Important: The Analytic Server administrator password must be the same as the Knox administrator password.

Configuring Apache Knox

1. Refresh the Knox gateway.jks file:
   a. On the Knox server, stop the Knox service.
   b. Delete the gateway.jks from /var/lib/knox/data-2.6.2.0-205/security/keystores.
   c. Restart the Knox service.
2. On the Knox server, create the sub directory <knox_server>/data/service/analyticserver/3.1.1.1, then upload the service.xml and rewrite.xml files to the new directory. The two files are on the Analytic Server at <analytic_server>/configuration/knox/analyticserver/3.1.1.1 (for example,/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/knox/analyticserver/3.1.1.1/*.xml)
3. In <knox_server>/bin, run the script ./knoxcli.sh redeploy --cluster default
4. Upload the com.ibm.spss.knoxservice_0.6-*.jar file to <knox_server>/ext. The file is on the Analytic Server at <analytic_server>/apps/AE_BOOT.war/WEB-INF/lib/com.ibm.spss.knox_0.6-3.1.1.1.jar (for example,/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/com.ibm.spss.knox_0.6-3.1.1.1.jar).
5. In the Ambari user interface, add the following element in **Knox** > **Configs** > **Advanced topology**:

   ```
   <service>
      <role>ANALYTICSERVER</role>
      <url>http://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
      <role>ANALYTICSERVER_WS</role>
      <url>ws://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
   </service>
   ```

   Note: By default WebSocket functionality is disabled. It can be enabled by changing the gateway.websocket.feature.enabled property to true in the /conf/gateway-site.xml file.
6. In the Ambari user interface, add or update the users in **Knox** > **Configs** > **Advanced users-ldif** (for example admin, qauser1, qauser2).
7. Restart LDAP from **Knox** > **Service Actions** > **Start Demo LDAP**.
8. Restart the Knox service.

Installing Apache Knox on Hortonworks Data Platform (HDP)

The following steps outline the process of installing Apache Knox in an HDP cluster.

1. Verify whether a Knox user exists on the HDP cluster. If a Knox user does not exist, you must create one.
2. Download and extract Apache Knox to a folder under /home/knox.
3. In HDP, switch to the Knox user and go to the knox folder. The Knox user must have permission(RWX) on all knox subfolders.

4. Configure Apache Knox for Analytic Server. For more information, refer to the **Configuring Apache Knox** section.
   a. Create an `analyticserver/3.1.1.1` folder hierarchy under `{knox}/data/services`.
   b. Copy the `rewrite.xml` and `service.xml` files from the Analytic Server location:.

      `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/knox/analyticserver/3.1.1.1`

      to the Knox server node:

      `{knox}/data/services/analyticserver/3.1.1.1`
   c. Copy the Knox *.jar file from the Analytic Serverhost:

      `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/com.ibm.spss.knox_0.6-*.jar`

      to the Knox `ext` directory:

      `{knox}/ext`
   d. Update the `default.xml` file in `{knox}/conf/topologies` to match the following example:

      Note: You must create the file if it does not exist.

```
<topology>
   <gateway>
      <provider>
         <role>authentication</role>
         <name>ShiroProvider</name>
         <enabled>true</enabled>
         <param>
            <name>sessionTimeout</name>
            <value>30</value>
         </param>
         <param>
            <name>main.ldapRealm</name>
            <value>org.apache.hadoop.gateway.shirorealm.KnoxLdapRealm</value>
         </param>
         <param>
            <name>main.ldapRealm.userDnTemplate</name>
            <value>uid={0},ou=people,dc=hadoop,dc=apache,dc=org</value>
         </param>
         <param>
            <name>main.ldapRealm.contextFactory.url</name>
            <value>ldap://localhost:33389</value>
         </param>
         <param>
            <name>main.ldapRealm.contextFactory.authenticationMechanism</name>
            <value>simple</value>
         </param>
         <param>
            <name>urls./**</name>
            <value>authcBasic</value>
         </param>
      </provider>
      <provider>
         <role>identity-assertion</role>
         <name>Default</name>
         <enabled>true</enabled>
      </provider>
      <provider>
         <role>authorization</role>
         <name>AclsAuthz</name>
         <enabled>true</enabled>
      </provider>

   </gateway>

   <!--other service-->
   <service>
      <role>ANALYTICSERVER</role>
      <!--replace the {analyticserver-host} and {analyticserver-port} with real value-->
      <url>http://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
      <role>ANALYTICSERVER_WS</role>
      <url>ws://{analyticserver-host}:{analyticserver-port}/analyticserver</url>
   </service>

</topology>
```

      Note: By default WebSocket functionality is disabled. It can be enabled by changing the `gateway.websocket.feature.enabled` property to `true` in the `/conf/gateway-site.xml` file.

5. Run `{knox}/bin/knoxcli.sh`.
6. Run `{knox}/bin/ldap.sh start`.

   Note: The script uses port 33389. Ensure that the port is not currently in use.
7. Run `{knox}/bin/gateway.sh start`.

   Note: The script uses port 8443. Ensure that the port is not currently in use.
8. Verify the installation.
   a. Run the `curl` command against the Analytic Server on Knox URL:

   ```
   curl -ikvu {username}:{password} https://{knox-host}:8443/gateway/default/analyticserver/admin
   ```

Troubleshooting

Problem: Analytic Server does not work in Knox after installation.

Solution: Stop Knox, remove all files under `{knox}/data/deployments/*`, and then restart Knox.

Problem: Cannot login to Analytic Server through Knox.

Solution: Verify the users in `{knox}/conf/users.ldif`. Update existing users, or add new Analytic Server users. The Knox user principals and credentials must match the Analytic Server users.

URL structure for the Apache Knox enabled Analytic Server

The Knox enabled Analytic Server user interface URL is `https://{knox-host}:{knox-port}/gateway/default/analyticserver/admin`

- https protocol - users must accept a certificate to proceed in the web browser.
- `knox-host` is the Knox host.
- `knox-port` is the Knox port number.
- The URI is `gateway/default/analyticserver`.

Configuring separate YARN queues for each IBM SPSS Analytic Server tenant - HDP

Configuring Yarn queues is accomplished through the use of Spark Dynamic Resource Allocation technicals.

Hortonworks Data Platform 2.x

1. In the Ambari user interface, navigate to **SPSS Analytic Server service** > **Configs** > **Advanced analytics.cfg** tab.
2. Change the **resource.pool.enabled** value to `true`.
3. Add the following properties on the **Custom analytics.cfg** tab:

   ```
   config.folder.path=/etc/spark2/conf
   resource.pool.mapping=tenant1:test,tenant2:production
   resource.pool.default=default
   spark.scheduler.mode=FAIR
   spark.yarn.queue=default
   ```

Table 8. Custom analytics.cfg properties

Property	**Description**
`config.folder.path`	The directory contains the `fairscheduler.xml` file that contains the Spark pool properties information. The file is required and must be created manually. For more information, see the **fairscheduler.xml example** section.

Table 8. Custom analytics.cfg properties (continued)

Property	Description
resource.pool.mapping	**Spark:** Maps the tenants to the pools that are defined in the fairscheduler.xml file. Tenant pairs must be separated by commas (for example, tenant1:test,tenant2:production. Before specifying a pool, ensure that the pool is configured in the fairscheduler.xml file. **MapReduce:** Maps tenants to the queue that is defined in the YARN Queue Manager. Tenant pairs must be separated by commas (for example, tenant1:test,tenant2:production). Before specifying a queue, ensure that the system is configured with the queue, and access is allowed for submitting jobs to the queue. **Note:** If you want to run the Spark and MapReduce jobs together, the tenant map values must be the same name in the fairscheduler.xml file and the YARN Queue Manager.
resource.pool.default	**Spark:** Defines the default resource pool. The value can be default or a pool name that is defined in the fairscheduler.xml file. Use the default setting when tenants are not configured (or configured incorrectly). **MapReduce:** Defines the default queue to which jobs are submitted.
spark.scheduler.mode=FAIR	**Spark:** Enables the fair scheduler. The property should not be changed.
spark.yarn.queue	**Spark:** The name of the YARN queue to which the application is submitted. You can specify a customized YARN queue name in the YARN Queue Manager.

4. Save the configuration and restart the Analytic Server service.

fairscheduler.xml example

The fairscheduler.xml file contains the Spark pool properties information. The file is required and must be created manually.

```
<?xml version="1.0"?>
<allocations>
  <pool name="production">
    <schedulingMode>FAIR</schedulingMode>
    <weight>1</weight>
    <minShare>2</minShare>
  </pool>
  <pool name="test">
    <schedulingMode>FIFO</schedulingMode>
    <weight>2</weight>
    <minShare>3</minShare>
  </pool>
</allocations>
```

Reference

Refer to the following sites for more information:

- https://spark.apache.org/docs/latest/job-scheduling.html#dynamic-resource-allocation
- https://spark.apache.org/docs/latest/running-on-yarn.html

Migrating IBM SPSS Analytic Server on Ambari

Analytic Server can migrate data and configuration settings from an existing Analytic Server installation to a new installation. The migration can occur on the same cluster environment, or on a new cluster environment.

Migrating from Analytic Server 3.1.1 to 3.1.1.1 on the same server cluster

If you have an existing installation of Analytic Server 3.1.1, you can migrate your 3.1.1 configuration settings to your 3.1.1.1 installation on the same server cluster.

1. Collect the configuration settings from the old Analytic Server version (Analytic Server 3.1).
   a. Expand the `{AS_ROOT}\tools\unzip configcollector.zip` archive (it will create a new folder named `configcollector`).
   b. Run the `configcollector.sh` script in the `configcollector` folder. Copy the resulting compressed (ZIP) `ASConfiguration_3.1.1.0.xxx.zip` file to a different folder location (as a backup).
2. Backup the analytic root from your old Analytic Server 3.1.1 version installation to a new location.
   a. If you are unsure of the location of the analytic root, run the **`hadoop fs -ls`** command. The path to the analytic root is similar to `/user/as_user/analytic-root/analytic-workspace`, where `as_user` is the user ID that owns the analytic root.
   b. Use the **`hadoop fs -copyToLocal`** and **`hadoop fs -copyFromLocal`** commands to copy the old Analytic Server version `analytic-workspace` folder to the new location (for example, `/user/as_user/analytic-root/AS31Location`).
3. If you use the embedded Apache Directory Server , backup the current user/group configuration with a 3rd-party LDAP client tool. After Analytic Server 3.1.1.1 is installed import the backup user/group configuration to the Apache Directory Server.

   Note: This step can be skipped if you use an external LDAP server.
4. Open the Ambari console and stop the **Analytic Server service**.
5. Uninstall the old Analytic Server version (Analytic Server 3.1.1), and then install Analytic Server 3.1.1.1. For installation instructions, see Chapter 2, "Ambari Installation and Configuration," on page 3.
6. Open the Ambari console and stop the **Analytic Server service** (in Ambari, ensure that the **Analytic Metastore service** is running).
7. Copy the backed-up Analytic Server 3.1.1 analytic root, from step 2, to the new Analytic Server version location.
   a. Remove the `analytic-workspace` from the newly installed Analytic Server version.
   b. Copy the backed-up Analytic Server 3.1.1 analytic workspace folder (`/user/as_user/analytic-root/AS31Location`) to the new version location (for example, `/user/as_user/analytic-root/analytic-workspace`). You must ensure that the analytic workspace owner is defined as `as_user`.
8. Clear the Zookeeper state. In the Zookeeper bin directory (for example, `/usr/hdp/current/zookeeper-client` on Hortonworks), run the following command:
   ```
   ./zkCli.sh rmr /AnalyticServer
   ```
9. Copy the backup archive `ASConfiguration_3.1.1.0.xxx.zip` from step 1 to the new Analytic Server version location (for example, `/opt/ibm/spss/analyticserver/3.1/`).
10. Run the migration tool by running the **`migrationtool.sh`** script and passing the path of the `ASConfiguration_3.1.1.0.xxx.zip` archive file (that was created by the configuration collector) as an argument. For example:
    ```
    migrationtool.sh /opt/ibm/spss/analyticserver/3.1/ASConfiguration_3.1.1.0.xxx.zip
    ```
11. Update the `ae_wlpserver/usr/servers/aeserver/configuration/config.properties` file on every Analytic Server cluster node.
    - Add an entry for `as_user` to the file. For example:
      ```
      hdfs.user=as_user/host@REALM
      ```
      `host` must match the Analytic Server node host name upon which the `config.properties` file resides.

      Each node has a different `hdfs.user` value; each host value must match the Analytic Server host upon which it resides.
12. Run the following command from a command shell on the Analytic Server node:

```
hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
```

13. In the Ambari console, start the **Analytic Server service**.

Migrating from Analytic Server 3.1.1 to 3.1.1.1 on a new server cluster

If you have an existing installation of Analytic Server 3.1.1, you can migrate your 3.1.1 configuration settings to your 3.1.1.1 installation on a new server cluster.

1. Install the new Analytic Server version according to the instructions in "Installation on Ambari" on page 5.
2. Copy the analytic workspace from your old installation to your new one.
   a. If you are unsure of the location of the analytic workspace, run `hadoop fs -ls`. The path to the analytic workspace is similar to `/user/as_user/analytic-root/analytic-workspace`, where `as_user` is the user ID that owns the analytic workspace.
   b. Remove the `analytic-workspace` on the new server.
   c. Use `hadoop fs -copyToLocal` and `hadoop fs -copyFromLocal` to copy the old server's analytic workspace to the new server's `/user/as_user/analytic-root/analytic-workspace` folder (ensure that the owner is set as `as_user`).
3. If you use the embedded Apache Directory Server, backup the current user/group configuration with a 3rd-party LDAP client tool. After Analytic Server 3.1.1.1 is installed import the backup user/group configuration to the Apache Directory Server.

   Note: This step can be skipped if you use an external LDAP server.
4. On new server, open the Ambari console, and stop the Analytic Server service (on Ambari, ensure that the Analytic Metastore service is running).
5. Collect the configuration settings from the old installation.
   a. Copy the `configcollector.zip` archive in your new installation to `{AS_ROOT}\tools` in your old installation.
   b. Extract the copy of `configcollector.zip`, which creates a new `configcollector` subdirectory in your old installation.
   c. Run the configuration collector tool in your old installation by running the **`configcollector`** script in `{AS_ROOT}\tools\configcollector`. Copy the resulting compressed (ZIP) file to the server that hosts your new installation.

   Important: The provided **`configcollector`** script may not be compatible with the most recent Analytic Server version. Contact you IBM technical support representative if you encounter problems with the **`configcollector`** script.
6. Clear the Zookeeper state. In the Zookeeper bin directory (for example, `/usr/hdp/current/zookeeper-client` on Hortonworks), run the following command.
   ```
   ./zkCli.sh rmr /AnalyticServer
   ```
7. Run the migration tool by running the **`migrationtool`** script and passing the path of the compressed file that was created by the configuration collector as an argument. An example follows.
   ```
   migrationtool.sh /opt/ibm/spss/analyticserver/3.1/ASConfiguration_3.1.1.0.xxx.zip
   ```
8. Update the `ae_wlpserver/usr/servers/aeserver/configuration/config.properties` file on every Analytic Server node. Add an entry for `as_user` to the file. For example:
   ```
   hdfs.user=as_user/host@REALM
   ```

   `host` must match the Analytic Server node host name upon which the `config.properties` file resides. Each node has a different `hdfs.user` value; each `host` value must match the Analytic Server host upon which it resides.
9. Run the following command from a command shell on the Analytic Server node:
   ```
   hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
   ```

10. In the Ambari console, start the Analytic Server service.

    Note: If you configured R for use with the existing Analytic Server installation, follow the steps to configure it with the new Analytic Server installation.

Uninstalling

Important: When Essentials for R is installed, you must first run the `remove_R.sh` script. Failure to uninstall Essentials for R, prior to uninstalling Analytic Server, results in the inability to uninstall Essentials for R at a later time. The `remove_R.sh` script is removed when Analytic Server is uninstalled. For information on uninstalling Essentials for R, see "Uninstalling Essentials for R."

1. On the Analytic Metastore host, run the `remove_as.sh` script in the `{AS_ROOT}/bin` directory with the following parameters.

   u Required. The Ambari Server administrator's user ID.

   p Required. The Ambari Server administrator's password.

   h Required. The Ambari Server host name.

   x Required. The Ambari Server port.

   l Optional. Enables secure mode.

   Examples follow.

   ```
   remove_as.sh -u admin -p admin -h one.cluster -x 8081
   ```

   Removes Analytic Server from a cluster with Ambari host one.cluster.

   ```
   remove_as.sh -u admin -p admin -h one.cluster -x 8081 -l
   ```

   Removes Analytic Server from a cluster with Ambari host one.cluster, in secure mode.

Note: This operation removes the Analytic Server folder on HDFS.

Note: This operation does not remove any Db2 schemas associated with Analytic Server. Consult the Db2 documentation for information on manually removing schemas

Uninstalling Essentials for R

1. On the Essentials for R host, run the `remove_R.sh` script in the `{AS_ROOT}/bin` directory with the following parameters.

   u Required. The Ambari Server administrator's user ID.

   p Required. The Ambari Server administrator's password.

   h Required. The Ambari Server host name.

   x Required. The Ambari Server port.

   l Optional. Enables secure mode.

   Examples follow.

   ```
   remove_R.sh -u admin -p admin -h one.cluster -x 8081
   ```

   Removes Essentials for R from a cluster with Ambari host one.cluster.

   ```
   remove_R.sh -u admin -p admin -h one.cluster -x 8081 -l
   ```

   Removes Essentials for R from a cluster with Ambari host one.cluster, in secure mode.

2. Remove the R services directory from the Ambari server services directory. For example, in HDP 2.6, the `ESSENTIALR` directory is located in `/var/lib/ambari-server/resources/stacks/HDP/2.6/services`.
3. In the Ambari console, verify that the Essentials for R service no longer exists.

Chapter 3. Cloudera Installation and Configuration

Cloudera overview

Cloudera is an open source Apache Hadoop distribution. The Cloudera Distribution Including Apache Hadoop (CDH), targets enterprise-class deployments of that technology.

Analytic Server can run on the CDH platform. CDH contains the main, core elements of Hadoop that provide reliable, scalable distributed data processing of large data sets (chiefly MapReduce and HDFS), as well as other enterprise-oriented components that provide security, high availability, and integration with hardware and other software.

Cloudera-specific prerequisites

In addition to the general prerequisites, review the following information.

Services

Ensure that the following instances are installed on each Analytic Server host.

- HDFS: Gateway, DataNode or NameNode
- Hive: Gateway, Hive Metastore Server or HiveServer2
- Yarn: Gateway, ResourceManager or NodeManager

The following instances are required only when their features are used.

- Accumulo: Gateway
- HBase: Gateway, Master or RegionServer
- Spark: Gateway
- Spark 2: Gateway

Metadata repository

You can use Db2 and MySQL as the Analytic Server metadata repository. If you plan to use MySQL as Analytic Server metadata repository, follow the instructions for "Configuring MySQL for Analytic Server" on page 37.

Kerberos enabled Cloudera environments

If you plan to install Analytic Server in a Kerberos enabled Cloudera environment, you must verify that Kerberos is properly configured in a manner that is compatible with Analytic Server.

The following sections apply to Cloudera environments where Kerberos is already installed. The following sections must be followed prior to installing Analytic Server in Cloudera. Is is assumed that you have basic Kerberos authentication knowledge as the sections include Kerberos-specific terminology (for example, **`kinit`**, **`kadmin`**, and so on).

Note: Analytic Server inspects the HDFS configuration for Kerberos related values to use for authentication.

Kerberos authentication

Verify that the Kerberos authentication is configured on each Cloudera cluster node prior to installing Analytic Server. For more information, see Configuring Authentication in Cloudera Manager in the Cloudera product documentation.

Note: After configuring Kerberos authentication on each Cloudera cluster node, the **cloudera-scm-server** and **cloudera-scm-agent** services must be restarted prior to installing Analytic Server. The **cloudera-scm-agent** service must be restarted on all cluster nodes.

Creating the required accounts in Kerberos

1. Create accounts in the Kerberos user repository for all users you plan to give access to Analytic Server.
2. Create the same accounts (from the previous step) on the LDAP server.
3. Create an OS user account for each of the users created in the previous step on each and every Analytic Server node and Hadoop node.
   - Make sure that the UID for these users matches on all machines. You can test this using the `kinit` command to log in to each of the accounts.
   - Make sure that the UID adheres to the **Minimum user ID for submitting job** Yarn setting. This is the **`min.user.id`** setting in `container-executor.cfg`. For example, if **`min.user.id`** is 1000, then each user account created must have a UID greater than or equal to 1000.
4. Create a user home folder on HDFS for the Analytic Server administrator user. The folder permission must be set to `777`, the owner must be defined as `admin`, and the user group must be set as `hdfs`. See the following, bolded example:

```
[root@xxxxx configuration]# hadoop fs -ls /user

Found 9 items

drwxrwxrwx   - hdfs     supergroup      0 2017-07-26 03:41 /user/AE
drwxrwxrwx   - admin    hdfs            0 2017-06-08 01:33 /user/admin
drwxr-x--x   - as_user  hdfs            0 2017-06-06 01:00 /user/as_user
drwx------   - hdfs     supergroup      0 2017-07-31 00:17 /user/hdfs
drwxrwxrwx   - mapred   hadoop          0 2017-06-05 00:28 /user/history
drwxrwxr-t   - hive     hive            0 2017-06-05 00:30 /user/hive
drwxrwxr-x   - hue      hue             0 2017-06-05 00:30 /user/hue
drwxrwxr-x   - impala   impala          0 2017-07-19 00:52 /user/impala
drwxr-x--x   - spark    spark           0 2017-06-05 01:34 /user/spark
```

5. If you plan to use HCatalog data sources and Analytic Server is installed on a different machine from the Hive metastore, you need to impersonate the Hive client on HDFS.
   a. Navigate to the Configuration tab of the HDFS service in Cloudera Manager.

      Note: The following settings may not appear on the **Configuration** tab if they have not already been set. In this case, run a search to find them.
   b. Edit the **`hadoop.proxyuser.hive.groups`** setting to have the value `*`, or a group that contains all of the users allowed to log in to Analytic Server.
   c. Edit the **`hadoop.proxyuser.hive.hosts`** setting to have the value `*`, or the list of hosts on which the Hive metastore and every instance of Analytic Server are installed as services.
   d. Restart the HDFS service.

After these steps have been performed and Analytic Server is installed, Analytic Server silently and automatically configures Kerberos.

Enabling Kerberos impersonation

Impersonation allows a thread to execute in a security context that differs from the security context of the process that owns the thread. For example, impersonation provides a means for Hadoop jobs to run as users other than the standard Analytic Server user (`as_user`). To enable Kerberos impersonation:

1. Open Cloudera Manager and add or update the following properties in the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** area (located on the **HDFS (Service-Wide)** > **Configuration** tab).
   - **Name:** `hadoop.proxyuser.as_user.hosts`
   - **Value:** `*`

- **Name:** hadoop.proxyuser.as_user.groups
- **Value:** *

Note: The **core-site.xml** settings apply to the Hadoop configuration (not Analytic Server).

2. Run the following command from a command shell on the Analytic Server node:

```
hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
```

Configuring MySQL for Analytic Server

Configuring the IBM SPSS Analytic Server in Cloudera Manager requires the installation and configuration of a MySQL server database.

1. Run the following command from a command window on the node on which the MySQL database is stored:

```
yum install mysql-server
```

Note: Use zypper install mysql for SuSE Linux.

2. Run the following command from a command window on each Cloudera cluster node:

```
yum install mysql-connector-java
```

Note: Use sudo zypper install mysql-connector-java for SuSE Linux.

3. Decide upon, and take note of, the Analytic Server database name, database user name, and database password that Analytic Server uses when it accesses the MySQL database.
4. Install Analytic Server according to the instructions in "Installation on Cloudera."
5. Copy the /opt/cloudera/parcels/AnalyticServer/bin/add_mysql_user.sh script from one of the servers managed by Cloudera to the node where the MySQL database is installed. Run the script with parameters that are appropriate for your particular configuration. For example:

```
./add_mysql_user.sh -u <database_user_name> -p <database_password> -d
<database_name>
```

Notes: The a -r <dbRootPassword) parameter is required when the database runs in secured mode (the root user password is set).

The -r <dbUserPassword> and -t <dbUserName> parameters are required when the database is running in secured mode with a user name other than root.

Installation on Cloudera

The following steps explain the process of manually installing IBM SPSS Analytic Server in Cloudera Manager.

Analytic Server 3.1.1.1

Online installation

1. Navigate to the IBM Passport Advantage® Web Site and download the self-extracting binary file specific to your stack, stack version, and hardware architecture to a host within the Cloudera cluster. The available Cloudera binaries are:

Table 9. Analytic Server self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for Cloudera 5.10, 5.11, 5.12, and 5.13 Ubuntu English	spss_as-3.1.1.1-cdh5.10-5.13-ubun.bin
IBM SPSS Analytic Server 3.1.1.1 for Cloudera 5.10, 5.11, 5.12, and 5.13 Linux x86-64 English	spss_as-3.1.1.1-cdh5.10-5.13-lx86.bin

2. Run the Cloudera self-extracting `*.bin` installer on the Cloudera Manager master cluster node. Follow the installation prompts by accepting the license agreement and keeping the default CSD installation directory.

   Note: You must specify a different CSD directory if it is altered from the default location.
3. Use the following command to restart Cloudera Manager after the installation is complete:

   ```
   service cloudera-scm-server restart
   ```
4. Open the Cloudera Manager interface (for example, `http://${CM_HOST}:7180/cmf/login` with the default login credentials of `admin/admin`), refresh the **Remote Parcel Repository URLs** (located in **Host** > **Parcels** > **click Configuration**), and verify that the URL is correct. For example:

   ```
   https://ibm-open-platform.ibm.com
   ```

   Note: The **Parcel Update Frequency** and **Remote Parcel Repository URLs** can be updated to suit your specific needs.
5. After Cloudera Manager refreshes the parcel files (you can manually refresh the parcel files by clicking **Check for New Parcels**), you will see that the **AnalyticServer** parcel status is set to **Available Remotely**.
6. Select **Download** > **Distribute** > **Activate**. The **AnalyticServer** parcel status is updated to **Distributed, Activated**.
7. In Cloudera Manager, add Analytic Server as a service and decide where to place the Analytic Server. You need to provide the following information in the Add Service Wizard:

   Note: The Add Service Wizard shows the overall progress during each phase of the service creation process, and provides a final confirmation message when the service is successfully installed and configured on the cluster.
   - Analytic Server metastore host name
   - Analytic Server metastore database name
   - Analytic Server metastore user name
   - Analytic Server metastore password

   MySQL as the Analytic Server metadata repository
   - Analytic Server metastore driver class: `com.mysql.jdbc.Driver`
   - Analytic Server metastore repository URL: `jdbc:mysql://${MySQL_DB}/{DBName}?createDatabaseIfNotExist=true`

     `{MySQL_DB}` is the hostname of the server where MySQL is installed

   Db2 as the Analytic Server metadata repository
   - Analytic Server metastore driver class: `com.ibm.db2.jcc.DB2Driver`
   - Analytic Server metastore repository URL: `jdbc:db2://{Db2_HOST}:{PORT}/{DBName}:currentSchema={SchemaName};`

     `{Db2_HOST}` is the hostname of the server where Db2 is installed.

     `{PORT}` is the port on which Db2 is listening.

     `{SchemaName}` is an available, unused schema.

     Work with your Db2 administrator if you are unsure of what values to enter.

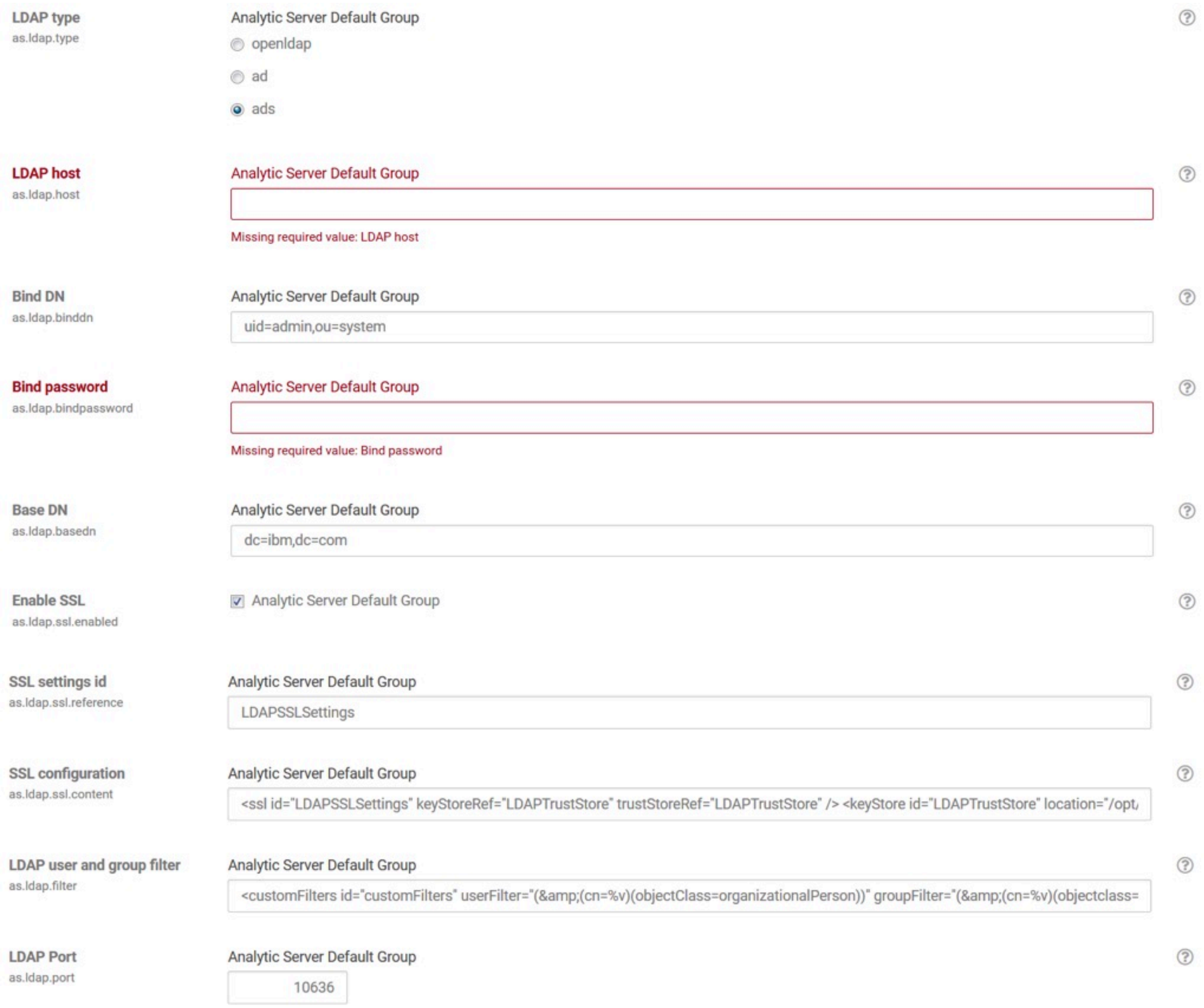
   LDAP configuration

   Analytic Server uses an LDAP server to store and authenticate users and groups. You provide the required LDAP configuration information during Analytic Server installation.

Table 10. LDAP configuration settings

LDAP setting	Description
`as.ldap.type`	LDAP type. The value can be `ads`, `ad`, or `openladp`. • `ads` - Apache Directory Server (default setting) • `ad` - Microsoft Active Directory • `openladp` - OpenLDAP
`as.ldap.host`	LDAP host
`as.ldap.port`	LDAP port number
`as.ldap.binddn`	LDAP bind DN
`as.ldap.bindpassword`	LDAP bind DN password
`as.ldap.basedn`	LDAP base DN
`as.ldap.filter`	LDAP user and group filter rule **Note:** When this value contains vertical bar `|` characters, the characters must be escaped with backslash characters (for example, `\\|`).
`as.ldap.ssl.enabled`	Specifies whether to use SSL to communicate between Analytic Server and LDAP. The value can be `true` or `false`.
`as.ldap.ssl.reference`	LDAP SSL reference ID
`as.ldap.ssl.content`	LDAP SSL configuration

- By default, `as.ldap.type` is set to `ads` and the other related settings contain default settings. The exception is you must provide a password for the `as.ldap.bindpassword` setting. Analytic Server uses the configuration settings to install an Apache Directory Server (ADS) and run the server initialization. The default ADS profile includes the user `admin` with a password of `admin`. You can conduct user management through the Analytic Server Console or import user and group information from an XML file via the `importUser.sh` script that is located in the `<Analytic Root>/bin` folder.
- If you plan to use an external LDAP server, such as Microsoft Active Directory or OpenLDAP, you must define the configuration settings according to the actual LDAP values. For more information, see Configuring LDAP user registries in Liberty.
- You can change the LDAP configuration after Analytic Server is installed (for example, changing from Apache Directory Server to OpenLDAP). However, if you initially start with Microsoft Active Directory or OpenLDAP, and decide to later switch to Apache Directory Server, Analytic Server will not install an Apache Directory Server during installation. The Apache Directory Server is only installed when it is selected during the initial Analytic Server installation.



LDAP type
as.ldap.type
Analytic Server Default Group
○ openldap
○ ad
● ads

LDAP host
as.ldap.host
Analytic Server Default Group
Missing required value: LDAP host

Bind DN
as.ldap.binddn
Analytic Server Default Group
uid=admin,ou=system

Bind password
as.ldap.bindpassword
Analytic Server Default Group
Missing required value: Bind password

Base DN
as.ldap.basedn
Analytic Server Default Group
dc=ibm,dc=com

Enable SSL
as.ldap.ssl.enabled
☑ Analytic Server Default Group

SSL settings id
as.ldap.ssl.reference
Analytic Server Default Group
LDAPSSLSettings

SSL configuration
as.ldap.ssl.content
Analytic Server Default Group
<ssl id="LDAPSSLSettings" keyStoreRef="LDAPTrustStore" trustStoreRef="LDAPTrustStore" /> <keyStore id="LDAPTrustStore" location="/opt

LDAP user and group filter
as.ldap.filter
Analytic Server Default Group
<customFilters id="customFilters" userFilter="(&amp;(cn=%v)(objectClass=organizationalPerson))" groupFilter="(&amp;(cn=%v)(objectclass=

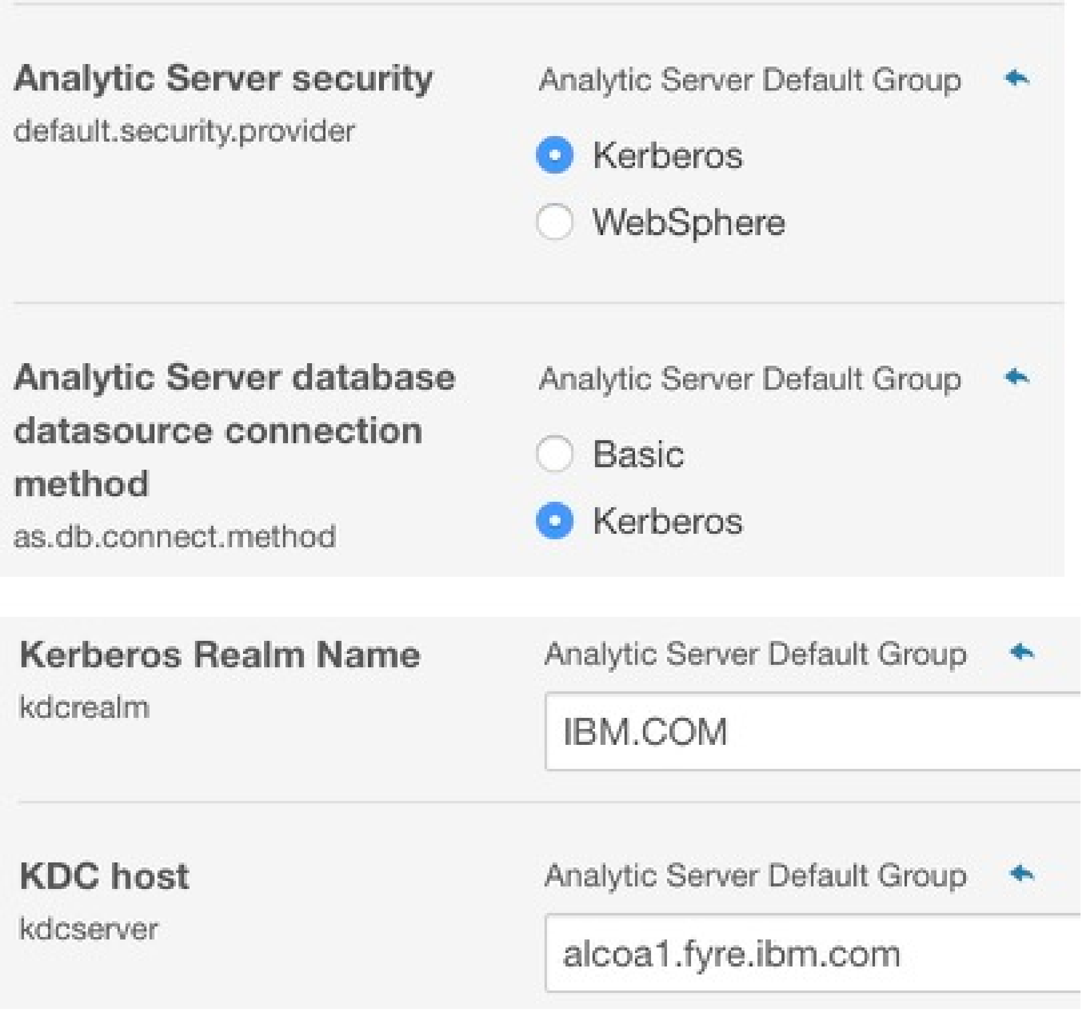
LDAP Port
as.ldap.port
Analytic Server Default Group
10636

Figure 2. Example LDAP configuration settings

8. When installing Analytic Server in a Kerberos enabled Cloudera environment, the following settings must also be configured in the Add Service Wizard:

   Note: Analytic Server inspects the HDFS configuration for Kerberos related values to use for authentication.

   - Select `Kerberos` as the **Analytic Server security** setting if you want to enable Kerberos authentication when logging into the Analytic Server console. When **Kerberos** is selected as the **Analytic Server security** setting, the Analytic Server console defaults to the Kerberos login mode.
   - Select `Kerberos` as the **Analytic Server database data source connection method** setting when you want to connect to Kerberos enabled databases. When **Kerberos** is selected as the **Analytic Server database data source connection method** setting, the Analytic Server console uses Kerberos mode when connecting to a database
   - The **Kerberos Realm Name** and **KDC host** settings are required. The **Kerberos Realm Name** and **KDC host** values are located in the `krb5.conf` file on the Kerberos Key Distribution Center (KDC) server.



Analytic Server security
default.security.provider
Analytic Server Default Group
● Kerberos
○ WebSphere

Analytic Server database datasource connection method
as.db.connect.method
Analytic Server Default Group
○ Basic
● Kerberos

Kerberos Realm Name
kdcrealm
Analytic Server Default Group
IBM.COM

KDC host
kdcserver
Analytic Server Default Group
alcoa1.fyre.ibm.com

Figure 3. Example Kerberos settings

Notes:

- The **Analytic Server security** and **Analytic Server database data source connection method** settings are applicable to IBM SPSS Modeler client and Analytic Server console authentication.
- When **Analytic Server database data source connection method** is set to `Kerberos`, you must ensure that the target databases are also Kerberos enabled.
- The **Analytic Server security** and **Analytic Server database data source connection method** settings do not configure Kerberos authentication on the Hadoop cluster. For more information, see the "Enabling Kerberos impersonation" section.
- If you want Kerberos authentication to be enabled at login, you must deploy the IBM SPSS Modeler client as a valid Kerberos client. This is accomplished by using the **`addprinc`** command in the Kerberos Key Distribution Center (KDC) server. For more information, refer to your IBM SPSS Modeler documentation.

When installing Analytic Server in a Kerberos enabled Cloudera environment you must also create the required accounts in Kerberos and enable Kerberos impersonation. For more information, see “Configuring Kerberos” on page 43.

Note: After successfully installing Analytic Server, do not click **Create Analytic Server Metastore** in the Actions list on the Analytic Server services page in Cloudera Manager. Creating a metastore overwrites the existing metadata repository.

Offline installation

The offline installation steps are the same as the online steps except you must manually download the parcel files and metadata that are appropriate for your particular operating system.

RedHat Linux requires the following files:

- AnalyticServer-3.1.1.1-el7.parcel
- AnalyticServer-3.1.1.1-el7.parcel.sha
- manifest.json

SuSE Linux requires the following files:

- AnalyticServer-3.1.1.1-sles11.parcel
- AnalyticServer-3.1.1.1-sles11.parcel.sha
- manifest.json

  or
- AnalyticServer-3.1.1.1-sles12.parcel
- AnalyticServer-3.1.1.1-sles12.parcel.sha

Ubuntu Linux 14.04 requires the following files:

- AnalyticServer-3.1.1.1-trusty.parcel
- AnalyticServer-3.1.1.1-trusty.parcel.sha

Ubuntu Linux 16.04 requires the following files:

- AnalyticServer-3.1.1.1-xenial.parcel
- AnalyticServer-3.1.1.1-xenial.parcel.sha

1. Download and run the Cloudera self-extracting `*.bin` installer on the Cloudera Manager master cluster node. Follow the installation prompts by accepting the license agreement and keeping the default `CSD` installation directory.

   Note: You must specify a different `CSD` directory if it differs from the default location.
2. Copy the required parcel and metadata files to your local Cloudera `repo` path on the Cloudera Manager master cluster node. The default path is `/opt/cloudera/parcel-repo` (the path is configurable in the Cloudera Manager user interface).
3. Use the following command to restart Cloudera Manager:

   ```
   service cloudera-scm-server restart
   ```

   The **AnalyticServer** parcel shows as **downloaded** after Cloudera Manager refreshes the parcel. You can click **Check for New Parcels** to force a refresh.
4. Click **Distribute** > **Activate**.

   The **AnalyticServer** parcel shows as distributed and activated.
5. In Cloudera Manager, add Analytic Server as a service. Refer to steps 7 and 8 in the "Online installation" section for more information.

Configuring Cloudera

After installation, you can optionally configure and administer Analytic Server through the Cloudera Manager.

Note: The following conventions are used for Analytic Server file paths.

- {AS_ROOT} refers to the location where Analytic Server is deployed; for example, `/opt/cloudera/parcels/AnalyticServer`.

- {AS_SERVER_ROOT} refers to the location of the configuration, log, and server files; for example, `/opt/cloudera/parcels/AnalyticServer/ae_wlpserver/usr/servers/aeserver`.
- {AS_HOME} refers to the location on HDFS that is used by Analytic Server as a root folder; for example, `/user/as_user/analytic-root`.

Security

The default **`tenant_id`** value in the IBM SPSS Modeler `options.cfg` file is **`ibm`**. You can view Tenants in the Analytic Server console. See the *IBM SPSS Analytic Server Administrator's Guide* for details on tenant management.

Configure an LDAP registry

LDAP is configured during Analytic Server installation. You can change to another LDAP server method after Analytic Server installation.

Note: Support for LDAP in Analytic Server is controlled by WebSphere Liberty. For more information, see Configuring LDAP user registries in Liberty.

Configure a secure socket layer (SSL) connection from Analytic Server to LDAP

1. Login to each of the Analytic Server machines as the Analytic Server user and create a common directory for SSL certificates.

   Note: On Cloudera, the Analytic Server user is always as_user, and this cannot be changed.
2. Copy the key store and trust store files to some common directory on all Analytic Server machines. Also add the LDAP client CA certificate to the trust store. Below are some sample instructions.
   ```
   mkdir /home/as_user/security
   cd /home/as_user/security
   openssl s_client -connect <ldap-hostname>:636 -showcerts > client.cert
   $JAVA_HOME/bin/keytool -import -file ./client.cert -alias ldapCA -keystore mytrust.jks
   password : changeit
   ```

   Note: JAVA_HOME is the same JRE used for Analytic Server startup.
3. Passwords can be encoded to obfuscate their values with the securityUtility tool, which is in `{AS_ROOT}/ae_wlpserver/bin`. An example follows.
   ```
   securityUtility encode changeit
           {xor}PDc+MTg6Nis=
   ```
4. Login to Cloudera Manager and update the Analytic Server configuration setting **`ssl_cfg`** with the correct SSL configuration settings. An example follows.
   ```
   <ssl id="defaultSSLConfig" keyStoreRef="defaultKeyStore" trustStoreRef="defaultTrustStore"
         clientAuthenticationSupported="true"/>
             <keyStore id="defaultKeyStore" location="/home/as_user/security/mykey.jks" type="JKS"
                       password="{xor}Ozo5PiozKxYdEgwPDAweDG1uDz4sLCg7"/>
             <keyStore id="defaultTrustStore" location="/home/as_user/security/mytrust.jks" type="JKS"
                       password="{xor}PDc+MTg6Nis="/>
   ```

   Note: Use the absolute path for key and trust store files.
5. Update the Analytic Server configuration setting **`security_cfg`** with the correct LDAP configuration settings. For example, in the **`ldapRegistry`** element, set the **`sslEnabled`** attribute to `true` and the **`sslRef`** attribute to `defaultSSLConfig`.

Configuring Kerberos

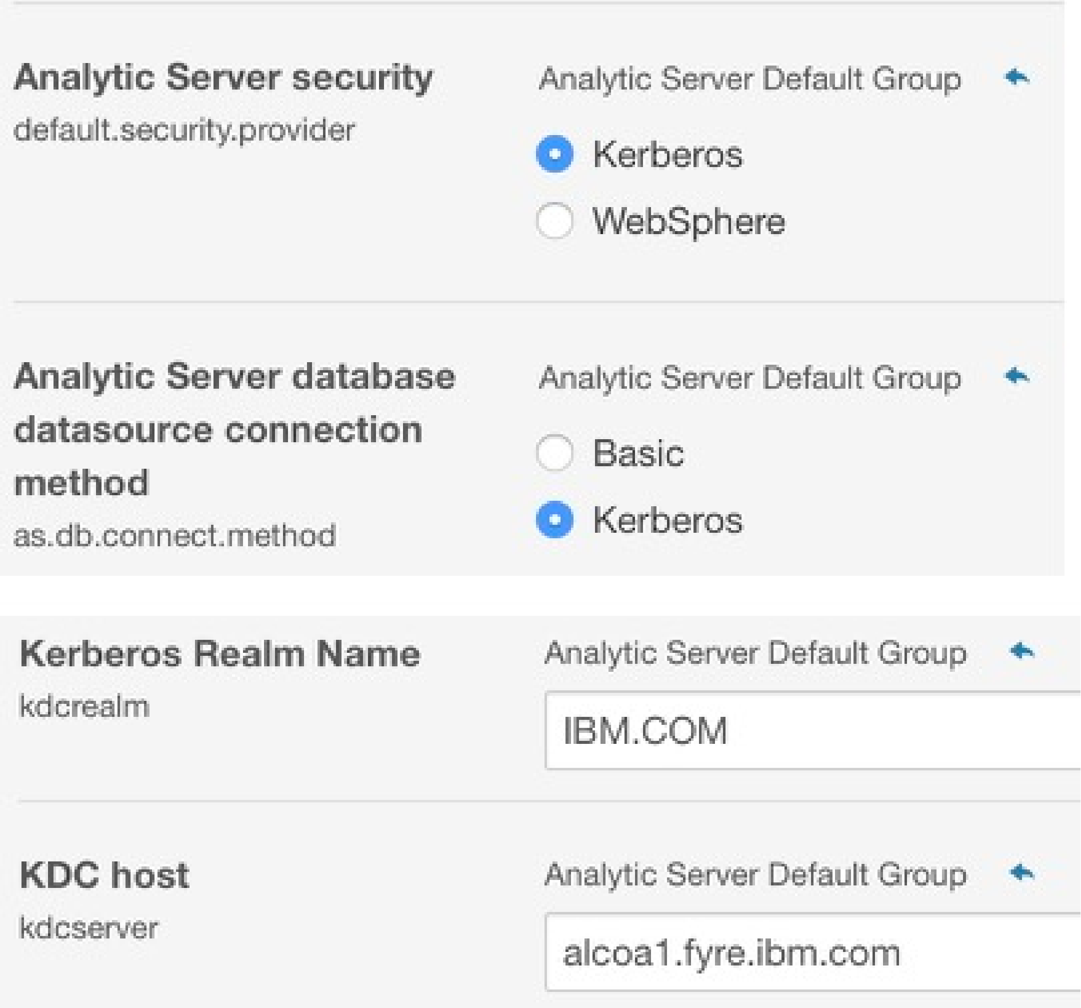
Analytic Server supports Kerberos in Cloudera. The following sections provide the configuration settings to ensure that Kerberos is properly configured in a manner that is compatible with Analytic Server.

Note: Analytic Server inspects the HDFS configuration for Kerberos related values to use for authentication.

Analytic Server and Kerberos settings

Keep the following settings in mind when installing Analytic Server in a Kerberos enabled Cloudera environment.

- Select `Kerberos` as the **Analytic Server security** setting if you want to enable Kerberos authentication when logging into the Analytic Server console. When **Kerberos** is selected as the **Analytic Server security** setting, the Analytic Server console defaults to the Kerberos login mode.
- Select `Kerberos` as the **Analytic Server database data source connection method** setting when you want to connect to Kerberos enabled databases. When **Kerberos** is selected as the **Analytic Server database data source connection method** setting, the Analytic Server console uses Kerberos mode when connecting to a database
- The **Kerberos Realm Name** and **KDC host** settings are required. The **Kerberos Realm Name** and **KDC host** values are located in the `krb5.conf` file on the Kerberos Key Distribution Center (KDC) server.



Analytic Server security
default.security.provider
Analytic Server Default Group
● Kerberos
○ WebSphere

Analytic Server database datasource connection method
as.db.connect.method
Analytic Server Default Group
○ Basic
● Kerberos

Kerberos Realm Name
kdcrealm
Analytic Server Default Group
IBM.COM

KDC host
kdcserver
Analytic Server Default Group
alcoa1.fyre.ibm.com

Figure 4. Example Kerberos settings

Notes:

- The **Analytic Server security** and **Analytic Server database data source connection method** settings are applicable to IBM SPSS Modeler client and Analytic Server console authentication.
- When **Analytic Server database data source connection method** is set to `Kerberos`, you must ensure that the target databases are also Kerberos enabled.

- The **Analytic Server security** and **Analytic Server database data source connection method** settings do not configure Kerberos authentication on the Hadoop cluster. For more information, see the "Enabling Kerberos impersonation" section.
- If you want Kerberos authentication to be enabled at login, you must deploy the IBM SPSS Modeler client as a valid Kerberos client. This is accomplished by using the **addprinc** command in the Kerberos Key Distribution Center (KDC) server. For more information, refer to your IBM SPSS Modeler documentation.

Creating the required accounts in Kerberos

1. Create accounts in the Kerberos user repository for all users you plan to give access to Analytic Server.
2. Create the same accounts (from the previous step) on the LDAP server.
3. Create an OS user account for each of the users created in the previous step on each and every Analytic Server node and Hadoop node.
   - Make sure that the UID for these users matches on all machines. You can test this using the `kinit` command to log in to each of the accounts.
   - Make sure that the UID adheres to the **Minimum user ID for submitting job** Yarn setting. This is the **`min.user.id`** setting in `container-executor.cfg`. For example, if **`min.user.id`** is 1000, then each user account created must have a UID greater than or equal to 1000.
4. Create a user home folder on HDFS for the Analytic Server administrator user. The folder permission must be set to `777`, the owner must be defined as `admin`, and the user group must be set as `hdfs`. See the following, bolded example:

```
[root@xxxxx configuration]# hadoop fs -ls /user

Found 9 items

drwxrwxrwx   - hdfs     supergroup      0 2017-07-26 03:41 /user/AE
drwxrwxrwx   - admin    hdfs            0 2017-06-08 01:33 /user/admin
drwxr-x--x   - as_user  hdfs            0 2017-06-06 01:00 /user/as_user
drwx------   - hdfs     supergroup      0 2017-07-31 00:17 /user/hdfs
drwxrwxrwx   - mapred   hadoop          0 2017-06-05 00:28 /user/history
drwxrwxr-t   - hive     hive            0 2017-06-05 00:30 /user/hive
drwxrwxr-x   - hue      hue             0 2017-06-05 00:30 /user/hue
drwxrwxr-x   - impala   impala          0 2017-07-19 00:52 /user/impala
drwxr-x--x   - spark    spark           0 2017-06-05 01:34 /user/spark
```

5. If you plan to use HCatalog data sources and Analytic Server is installed on a different machine from the Hive metastore, you need to impersonate the Hive client on HDFS.
   a. Navigate to the Configuration tab of the HDFS service in Cloudera Manager.

      Note: The following settings may not appear on the **Configuration** tab if they have not already been set. In this case, run a search to find them.
   b. Edit the **`hadoop.proxyuser.hive.groups`** setting to have the value `*`, or a group that contains all of the users allowed to log in to Analytic Server.
   c. Edit the **`hadoop.proxyuser.hive.hosts`** setting to have the value `*`, or the list of hosts on which the Hive metastore and every instance of Analytic Server are installed as services.
   d. Restart the HDFS service.

After these steps have been performed and Analytic Server is installed, Analytic Server silently and automatically configures Kerberos.

Enabling Kerberos impersonation

Impersonation allows a thread to execute in a security context that differs from the security context of the process that owns the thread. For example, impersonation provides a means for Hadoop jobs to run as users other than the standard Analytic Server user (`as_user`). To enable Kerberos impersonation:

1. Open Cloudera Manager and add or update the following properties in the **Cluster-wide Advanced Configuration Snippet (Safety Valve) for core-site.xml** area (located on the **HDFS (Service-Wide)** > **Configuration** tab).
   - **Name:** `hadoop.proxyuser.as_user.hosts`
   - **Value:** `*`
   - **Name:** `hadoop.proxyuser.as_user.groups`
   - **Value:** `*`

   Note: The **core-site.xml** settings apply to the Hadoop configuration (not Analytic Server).
2. Run the following command from a command shell on the Analytic Server node:
   ```
   hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
   ```

Configuring HAProxy for Single Sign On (SSO) using Kerberos

1. Configure and start HAProxy per the HAProxy documentation guide: http://www.haproxy.org/#docs
2. Create the Kerberos principle (HTTP/<proxyHostname>@<realm>) and keytab file for the HAProxy host, where <proxyHostname> is the full name of the HAProxy host, and <realm> is the Kerberos realm.
3. Copy the keytab file to each of the Analytic Server hosts as `/etc/security/keytabs/spnego_proxy.service.keytab`
4. Update permissions to this file on each of the Analytic Server hosts. An example follows.
   ```
   chown root:hadoop /etc/security/keytabs/spnego_proxy.service.keytab
   chmod 440 /etc/security/keytabs/spnego_proxy.service.keytab
   ```
5. Open Cloudera Manager and add or update the following properties in the Analytic Server **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area.
   ```
   web.authentication.kerberos.keytab=/etc/security/keytabs/spnego_proxy.service.keytab
   web.authentication.kerberos.principal=HTTP/<proxy machine full name>@<realm>
   ```
6. Save the configuration and restart all Analytic Server services from Cloudera Manager.
7. Instruct users to configure their browser to use Kerberos.

Now users are able to log in to Analytic Server using Kerberos SSO.

Disabling Kerberos

1. Disable Kerberos in the Cloudera Manager console.
2. Stop the Analytic Server service.
3. Remove the following settings from the **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area.
   ```
   default.security.provider
   hdfs.keytab
   hdfs.user
   java.security.krb5.conf
   as.db.connect.method
   web.authentication.kerberos.keytab
   web.authentication.kerberos.principal
   ```
4. Click **Save Changes** and restart the Analytic Server service.

Enabling Secure Socket Layer (SSL) connections to the Analytic Server console

By default, Analytic Server generates self-signed certificates to enable Secure Socket Layer (SSL), so you can access the Analytic Server console through the secure port by accepting self signed certificates. In order to make HTTPS access more secure, you need to install 3rd party vendor certificates.

To install 3rd party vendor certificates, follow these steps.

1. Copy the 3rd party vendor key store and trust store certificates to the same directory in all Analytic Server nodes; for example, `/home/as_user/security`.

   Note: The Analytic Server User must have read access to this directory.
2. In Cloudera Manager, navigate to the Configuration tab of the Analytic Server service.
3. Edit the **`ssl_cfg`** parameter.

   ```
   <ssl id="defaultSSLConfig"
        keyStoreRef="defaultKeyStore"
        trustStoreRef="defaultTrustStore"
        clientAuthenticationSupported="true"/>
   <keyStore id="defaultKeyStore"
             location="<KEYSTORE-LOCATION>"
             type="<TYPE>"
             password="<PASSWORD>"/>
   <keyStore id="defaultTrustStore"
             location="<TRUSTSTORE-LOCATION>"
             type="<TYPE>"
             password="<PASSWORD>"/>
   ```

   Replace
   - <KEYSTORE-LOCATION> with the absolute location of the key store; for example: `/home/as_user/security/mykey.jks`
   - <TRUSTSTORE-LOCATION> with the absolute location of the trust store; for example: `/home/as_user/security/mytrust.jks`
   - <TYPE> with the type of the certificate; for example: JKS, PKCS12 etc.
   - <PASSWORD> with the encrypted password in Base64 encryption format. For encoding you can use the securityUtility; for example: `{AS_ROOT}/ae_wlpserver/bin/securityUtility encode <password>`

   If you want to generate a self-signed certificate, you can use securityUtility; for example: `{AS_ROOT}/ae_wlpserver/bin/securityUtility createSSLCertificate --server=myserver --password=mypassword --validity=365 --subject=CN=mycompany,O=myOrg,C=myCountry`. For more information on securityUtility and other SSL settings, refer to the WebSphere Liberty Profile documentation.

   Note: You must provide an appropriate host domain name for the `CN` value.
4. Click **Save Changes** and restart the Analytic Server service.

Communicating with Apache Hive over SSL

You must update the `hive.properties` file in order to communicate with Apache Hive over an SSL connection. Alternatively, if your Apache Hive environment is enabled for high availability, you can select the high availability parameters on the main Analytic Server Data sources page.

Updating the hive.properties file

1. Open the `hive.properties` file. The file is located at: `/opt/cloudera/parcels/AnalyticServer-3.1.1.1/ae_wlpserver/usr/servers/aeserver/configuration/database`
2. Locate the following line:

   ```
   jdbcurl = jdbc:hive2://{db.servername}:{db.serverport}/{db.databasename};user={db.username};password={db.password}
   ```
3. Update the line by adding the **bold** information below:

   ```
   jdbcurl = jdbc:hive2://{db.servername}:{db.serverport}/{db.databasename};user={db.username};password={db.password};
   ssl=true;sslTrustStore=pathtotheirtruststorefile;
   ```
4. Save the `hive.properties` file.

Enabling support for Essentials for R

Analytic Server supports scoring R models and running R scripts.

To install Essentials for R after a successful Analytic Server installation in Cloudera Manager:

1. Provision the server environment for Essentials for R. For more information, see step 1 in "Enabling Support for Essentials for R" on page 19.
2. Download the self-extracting archive (BIN) for IBM SPSS Modeler Essentials for R RPM. Essentials for R is available for download (https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp). Choose the file specific to your stack, stack version, and hardware architecture.
3. Run the self-extracting archive as a root or sudo user on the Cloudera Manager server host. The following packages must be installed or available from the configured repositories:
   - Red Hat Linux: gcc-gfortran, zip, gcc-c++
   - SUSE Linux: gcc-fortran, zip, gcc-c++
   - Ubuntu Linux: gcc-fortran, zip, gcc-c++
4. The self-extracting installer does the following tasks:
   a. Displays the required licenses and prompts the installer to accept them.
   b. Prompts the installer to input the R source location, or continue with the default location. The default R version that is installed is 3.3.2. To install a different version:
      - Online installation: Provide the URL to the required R version archive. For example, https://cran.r-project.org/src/base/R-2/R-2.15.3.tar.gz for R 2.15.3.
      - Offline installation: Download and then copy the required R version archive to the Cloudera Manager server host. Do not rename the archive (by default, it is named `R-x.x.x.tar.gz`). Provide the URL to the copied R archive as follows: `file://<R_archive_directory>/R-x.x.x.tar.gz`. If the `R-2.15.3.tar.gz` archive was downloaded and then copied to `/root`, the URL is `file:///root/R-2.15.3.tar.gz`.

      Note: Other R versions can be found at https://cran.r-project.org/src/base/.
   c. Installs the packages that R requires.
   d. Downloads and installs R plus the Essentials for R plugin.
   e. Creates the parcel and `parcel.sha` file and copies them to `/opt/cloudera/parcel-repo`. Enter the correct location if the location has changed.
5. After the installation is complete, distribute and activate the **Essentials for R** parcel in Cloudera Manager (click **Check for New Parcels** to refresh the parcel list).
6. If the Analytic Server service is already installed:
   a. Stop the service.
   b. Refresh the Analytic Server binaries.
   c. Start the service to finish the Essentials for R installation.
7. If the Analytic Server service is not installed, then proceed with its installation.

Note: All Analytic Server hosts must have the appropriate archive (`zip` and `unzip`) packages installed.

Enabling relational database sources

Analytic Server can use relational database sources if you supply the JDBC drivers in a shared directory on each Analytic Server host. By default, this directory is `/usr/share/jdbc`.

To change the shared directory, follow these steps.

1. In Cloudera Manager, navigate to the Configuration tab of the Analytic Server service.
2. Specify the path of the shared directory of JDBC drivers in **`jdbc.drivers.location`**.

3. Click **Save Changes**.
4. Select **Stop** from the **Actions** dropdown to stop the Analytic Server service.
5. Select **Refresh Analytic Server Binaries** from the **Actions** dropdown.
6. Select **Start** from the **Actions** dropdown to start the Analytic Server service.

Table 11. Supported databases

Database	Supported versions	JDBC driver jars	Vendor
Amazon Redshift	8.0.2 or later	`RedshiftJDBC41-1.1.6.1006.jar` or later	Amazon
Apache Impala	JDBC 4 with 2.5.5 or later	`ImpalaJDBC4.jar`, `commons-codec-*.jar`, `commons-logging-*.jar`, `httpclient-*.jar`, `httpcore-*.jar`, `log4j-*.jar`, `libthrift-*.jar`, `libfb303-*.jar`, `slf4j-api-*.jar`, `ql.jar`, `zookeeper-*.jar`, `TCLIServiceClient.jar`	Apache
DashDB	Bluemix Service	`db2jcc.jar`	IBM
Db2 for Linux, UNIX, and Windows	11.1, 10.5, 10.1, 9.7	`db2jcc.jar`	IBM
Db2 z/OS	11, 10	`db2jcc.jar`, `db2_license_cisuz.jar`	IBM
Greenplum	5.x	postgresql.jar	Greenplum
Hive	1.1	`hive-jdbc-*.jar`	Apache
MySQL	5.6, 5.7	mysql-connector-java-commercial-5.1.25-bin.jar	MySQL
Netezza	7, 6.x	`nzjdbc.jar`	IBM
Oracle	12c, 11g R2 (11.2)	`ojdbc6.jar`, `orai18n.jar`	Oracle
SQL Server	2014, 2012, 2008 R2	`sqljdbc4.jar`	Microsoft
Teradata	15, 15.1	`tdgssconfig.jar`, `terajdbc4.jar`	Teradata

Notes

- If you created a Redshift data source prior to installing Analytic Server, you need perform the following steps in order to use the Redshift data source.
  1. In the Analytic Server console, open the Redshift data source.
  2. Select the `Redshift` database data source.
  3. Enter the Redshift server address.
  4. Enter the database name and username. The password should automatically populate.
  5. Select the database table.

Enabling HCatalog data sources

Analytic Server provides support for a number of data sources through Hive/HCatalog. Some sources require manual configuration steps.

1. Collect the necessary JAR files to enable the data source. See the sections below for details.

2. Add these JAR files to the `{HIVE_HOME}/auxlib` directory and to the `/usr/share/hive` directory on each Analytic Server node.
3. Restart the Hive Metastore service.
4. Restart each and every instance of the Analytic Server service.

Note:

When accessing HBase data via an Analytic Server HCatalog data source, the accessing user must have read permission for the HBase tables.

- In non-kerberos environments, Analytic Server accesses HBase using `as_user` (`as_user` must have read permission for HBase).
- In kerberos environments, both `as_user` and the login user must have read permission for HBase tables.

NoSQL databases

Analytic Server supports any NoSQL database for which a Hive storage handler is available from the vendor.

No additional steps are necessary to enable support for Apache HBase and Apache Accumulo.

For other NoSQL databases, contact the database vendor and obtain the storage handler and related jars.

File-based Hive tables

Analytic Server supports any file-based Hive tables for which a built-in or custom Hive SerDe (serializer-deserializer) is available.

The Hive XML SerDe for processing XML files is located in the Maven Central Repository at http://search.maven.org/#search%7Cga%7C1%7Chivexmlserde.

MapReduce v2 jobs

Use the **`preferred.mapreduce`** setting in the **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area to control how MapReduce jobs are handled:

Table 12. Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties

Property	Description
`preferred.mapreduce`	Controls the method in which MapReduce jobs are run. Valid values include: • `spark` • `m3r` • `hadoop` For example: `preferred.mapreduce=spark`

Apache Spark

If you want to use Spark (version 1.5 or later), you must select the `spark.version` during Analytic Server installation.

1. Open Cloudera Manager and select the appropriate `spark.version` (for example, `None`, `1.x`, or `2.x`) in the **Analytic Server Spark Version** area.

   Note: When using Spark 1.x, you must also add the following line in the **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area.

```
spark.extraListeners=org.apache.spark.JavaSparkListener
```

2. Save the configuration.

Configuring Apache Impala

Apache Impala is supported when running on Cloudera against an Analytic Server database data source or an HCatalog data source (regardless of whether Impala is SSL enabled).

Creating a database data source for Apache Impala data

1. On the main Analytic Server **Data sources** page, click **New** to create a new data source. the New data source dialog displays.
2. Enter an appropriate name in the **New data source** field, select `Database` as the **Content type** value, and then click **Ok**.
3. Open the **Database Selections** section and enter the following information.

   Database:
   Select **Impala** from the drop-down menu.

   Server address:
   Enter the URL of the server that hosts the Impala daemon. A fully qualified domain name is required when Kerberos is enabled for Analytic Server.

   Server port:
   Enter the port number that the Impala database listens on.

   Database name:
   Enter the name of the database to which you want to connect.

   Username:
   Enter a user name with authority to log into the Impala database.

   Password:
   Enter the appropriate user name password.

   Table name:
   Enter the name of a table from the database that you want to use. Click **Select** to manually select a file.

   Maximum concurrent reads:
   Enter the limit on the number of parallel queries that can be sent from Analytic Server to the database to read from the table specified in the data source.
4. Click **Save** after entering the required information.

Creating an HCatalog data source for Apache Impala data

1. On the main Analytic Server **Data sources** page, click **New** to create a new data source. the New data source dialog displays.
2. Enter an appropriate name in the **New data source** field, select `HCatalog` as the **Content type** value, and then click **Ok**.
3. Open the **Database Selections** section and enter the following information.

   Database:
   Select **default** from the drop-down menu.

   Table name:
   Enter the name of a table from the database that you want to use.

   HCatalog Schema
   Select the **HCatalog Element** option, and then select the appropriate **HCatalog Field Mappings** options.
4. Click **Save** after entering the required information.

Connecting to Apache Impala enabled data

1. Define the following Impala SSL settings in the Cloudera Manager console.

   Enable TLS/SSL for Impala (client_services_ssl_enabled)
   Select the **Impala (Service-Wide)** option.

   Impala TLS/SSL Server Certificate File (PEM Format) (ssl_server_certificate)
   Enter the self-signed, PEM format certificate location and file name (for example: `/tmp/<user_name>/ssl/l14200v21.crt`).

   Impala TLS/SSL Server Private Key File (PEM Format) (ssl_private_key)
   Enter the private key, in PEM format, location and file name (for example: `/tmp/<user_name>/ssl/l14200v21.key`).
2. On the Analytic Server host, import the `*.crf` file (that is used to enable Impala SSL) into a `*.jks` file. The file can be a cacerts file (for example, `/etc/pki/java/cacerts`) or any other `*.jks` file.
3. On the Analytic Server host, update the Impala configuration file (`impala.properties`) by appending the following `jdbcurl` key value:

   ```
   SSL=1;AllowSelfSignedCerts=1;CAIssuedCertNamesMismatch=1;
   ```

   Note: When a `*.jks` file (other than cacerts) is used, you need to also specify the following:

   ```
   SSLTrustStore=<your_pks_file>;SSLTrustStorePwd=<password_for_pks_file>;
   ```
4. Restart Analytic Server in the Cloudera Manager console.

Changing ports used by Analytic Server

Analytic Server uses the 9080 port for HTTP and the 9443 port for HTTPS by default. To change the port settings, follow these steps.

1. In Cloudera Manager, navigate to the Configuration tab of the Analytic Server service.
2. Specify the desired HTTP and HTTPS ports in the **`http.port`** and **`https.port`** parameters, respectively.

   Note: You may need to select the **Ports and Addresses** category in the Filters section in order to see these parameters.
3. Click **Save Changes**.
4. Restart the Analytic Server service.

High availability Analytic Server

You can make Analytic Server highly available by adding it as a service to multiple nodes in your cluster.

1. In Cloudera Manager, navigate to the Instances tab of the Analytic Server service.
2. Click **Add Role Instances** and select the hosts on which to add Analytic Server as a service.

Multiple-cluster support

The multiple-cluster feature is an enhancement to the High-Availability capability of IBM SPSS Analytic Server, and provides improved isolation in multiple-tenant environments. By default, installation of the Analytic Server service (in either Ambari or ClouderaManager) results in the definition of a single analytic server cluster.

The cluster specification defines the Analytic Server cluster membership. Modifying the cluster specification, is accomplished with XML content (in the Ambari Analytic Server configuration's `analytics-cluster` field or by manually editing the Cloudera Manager's `configuration/analytics-cluster.xml` file). When configuring multiple Analytic Server clusters, it is necessary to feed requests to each Analytic Server cluster with its own load balancer.

Using the multiple-cluster feature assures that work for one tenant cannot negatively impact work being performed in another tenant's cluster. With respect to highly available jobs, job failover occurs only within the scope of the Analytic Server cluster upon which the work was initiated. The following example provides a multiple-cluster XML specification.

Note: Analytic Server can be made highly available by adding it as a service to multiple nodes in your cluster.

```
<analayticServerClusterSpec>
    <cardinality>1+</cardinality>
    <cluster name="cluster1">
        <memberName>one.cluster</memberName>
        <memberName>two.cluster</memberName>
    </cluster>
    <cluster name="cluster2">
        <memberName>three.cluster</memberName>
        <memberName>four.cluster</memberName>
    </cluster>
</analayticServerClusterSpec>
```

In the previous example, two load balancers are required. One load balancer sends requests to the `cluster1` members (`one.cluster` and `two.cluster`) and the other sends requests to `cluster2` members (`three.cluster` and `four.cluster`).

The following example provides a single cluster XML specification (the default configuration).

```
<analayticServerClusterSpec>
    <cardinality>1</cardinality>
    <cluster name="cluster1">
        <memberName>*</memberName>
    </cluster>
</analayticServerClusterSpec>
```

In the previous example, a single load balancer is required to handle cases where there is more than one configured cluster member.

Notes

- Only singleton clusters support the use of wildcards in the **`memberName`** element (for example, cluster cardinality = "1"). Valid values for the cardinality element are 1 and 1+.
- The **`memberName`** must be specified in the same manner as the host name to which the Analytic Server role is assigned.
- All servers in all clusters must be restarted after the cluster configuration changes are applied.
- In Cloudera Manager, you must modify and maintain the `analytics-cluster.xml` file on all Analytic Server nodes. All nodes must be maintained to ensure that they contain the same content.

Optimizing JVM options for small data

You can edit JVM properties in order to optimize your system when running small (M3R) jobs.

In Cloudera Manager, see the **Jvm Options (jvm.options)** control on the Configuration tab in the Analytic Server service. Modifying the following parameters sets the heap size for jobs run on the server that hosts Analytic Server; that is, not Hadoop. This is important if running small (M3R) jobs, and you may need to experiment with these values to optimize your system.

```
-Xms512M
-Xmx2048M
```

Configuring separate YARN queues for each IBM SPSS Analytic Server tenant - Cloudera

Configuring Yarn queues is accomplished through the use of Spark Dynamic Resource Allocation technicals.

Cloudera 5.x

Follow these steps when adding the SPSS Analytic Server Service to an existing cluster.

1. In Cloudera Manager, navigate to **SPSS Analytic Server Service** > **Configuration**.
2. Change the **Resource Pool Enable: resource.pool.enabled** value to `true`.
3. Add the following properties to **Analytic Server Advanced Configuration Snippet (Safety Valve)** > **analyticserver-conf.config.properties**:

```
config.folder.path=/etc/spark2/conf
resource.pool.mapping=tenant1:test,tenant2:production
resource.pool.default=default
spark.scheduler.mode=FAIR
spark.yarn.queue=default
```

Table 13. analyticserver-conf.config.properties settings

Property	Description
`config.folder.path`	The directory contains the `fairscheduler.xml` file that contains the Spark pool properties information. The file is required and must be created manually. For more information, see the **fairscheduler.xml example** section.
`resource.pool.mapping`	**Spark:** Maps the tenants to the pools that are defined in the `fairscheduler.xml` file. Tenant pairs must be separated by commas (for example, `tenant1:test,tenant2:production`. Before specifying a pool, ensure that the pool is configured in the `fairscheduler.xml` file. **MapReduce:** Maps tenants to the queue that is defined in the Dynamic Resource Pool Configuration. Tenant pairs must be separated by commas (for example, tenant1:test,tenant2:production). Before specifying a queue, ensure that the system is configured with the queue, and access is allowed for submitting jobs to the queue. **Note:** If you want to run the Spark and MapReduce jobs together, the tenant map values must be the same name in the `fairscheduler.xml` file and the Dynamic Resource Pool Configuration.
`resource.pool.default`	**Spark:** Defines the default resource pool. The value can be `default` or a pool name that is defined in the `fairscheduler.xml` file. Use the `default` setting when tenants are not configured (or configured incorrectly). **MapReduce:** Defines the default queue to which jobs are submitted.
`spark.scheduler.mode=FAIR`	**Spark:** Enables the fair scheduler. The property should not be changed.
`spark.yarn.queue`	**Spark:** The name of the YARN queue to which the application is submitted. You can specify a customized YARN queue name in the Dynamic Resource Pool Configuration.

4. Save the configuration and restart the Analytic Server service.

fairscheduler.xml example

The `fairscheduler.xml` file contains the Spark pool properties information. The file is required and must be created manually.

```
<?xml version="1.0"?>
<allocations>
  <pool name="production">
    <schedulingMode>FAIR</schedulingMode>
    <weight>1</weight>
    <minShare>2</minShare>
  </pool>
  <pool name="test">
    <schedulingMode>FIFO</schedulingMode>
    <weight>2</weight>
    <minShare>3</minShare>
  </pool>
</allocations>
```

Reference

Refer to the following sites for more information:

- https://spark.apache.org/docs/latest/job-scheduling.html#dynamic-resource-allocation
- https://spark.apache.org/docs/latest/running-on-yarn.html

Migration

Analytic Server allows you to migrate data and configuration settings from an existing Analytic Server installation to a new installation.

Upgrade to a new version of Analytic Server
If you have an existing installation of Analytic Server 3.1 and have purchased a newer version, then you can migrate your 3.1 configuration settings to your new installation.

Restriction: Your 3.1 and new installations cannot coexist on the same Hadoop cluster. If you configure your new installation to use the same Hadoop cluster as your 3.1 installation, the 3.1 installation will no longer function.

Migration steps, 3.1.1 to newer version

1. Install the new installation of Analytic Server according to the instructions in "Installation on Cloudera" on page 37.
2. Copy the analytic workspace from your old installation to your new one.
   a. If you are unsure of the location of the analytic workspace, run `hadoop -fs ls`. The path to the analytic workspace will be of form `/user/as_user/analytic-root/analytic-workspace`, where `as_user` is the user ID that owns the analytic workspace.
   b. Log in to the host of the new Analytic Server installation as `as_user`. Delete the `/user/as_user/analytic-root/analytic-workspace` directory, if it exists.
   c. Run the following copy script.
   ```
   hadoop distcp hftp://{host of 3.1.1 namenode}:50070/{path to 3.1.1 analytic-workspace}
   hdfs://{host of 3.1.1.1 namenode}/user/as_user/analytic-root/analytic-workspace
   ```
3. If you use the embedded Apache Directory Server, backup the current user/group configuration with a 3rd-party LDAP client tool. After Analytic Server 3.1.1.1 is installed import the backup user/group configuration to the Apache Directory Server.

   Note: This step can be skipped if you use an external LDAP server.
4. In Cloudera Manager, stop the Analytic Server service.
5. Collect the configuration settings from the old installation.
   a. Copy the `configcollector.zip` archive in your new installation to `{AS_ROOT}\tools` in your old installation.
   b. Extract the copy of `configcollector.zip`. This creates a new `configcollector` subdirectory in your old installation.
   c. Run the configuration collector tool in your old installation by executing the **`configcollector`** script in `{AS_ROOT}\tools\configcollector`. Copy the resulting compressed (ZIP) file to the server that hosts your new installation.

   Important: The provided **`configcollector`** script may not be compatible with the most recent Analytic Server version. Contact you IBM technical support representative if you encounter problems with the **`configcollector`** script.
6. Clear the Zookeeper state. In the Zookeper bin directory (for example, `/opt/cloudera/parcels/CDH-5.4...../lib/zookeeper/bin` on Cloudera), run the following command.
   ```
   ./zkCli.sh rmr /AnalyticServer
   ```

7. Run the migration tool by executing the **migrationtool** script and passing the path of the compressed file created by the configuration collector as an argument. An example follows.

   ```
   migrationtool.sh /opt/ibm/spss/analyticserver/3.1/ASConfiguration_3.1.1.0.xxx.zip
   ```

8. Run the following command from a command shell on the Analytic Server node:

   ```
   hadoop fs -chmod -R 755 /user/as_user/analytic-root/analytic-workspace
   ```

9. In Cloudera Manager, start the Analytic Server service.

   Note: If you configured R for use with the existing Analytic Server installation, you will need to follow the steps to configure it with the new Analytic Server installation.

Uninstalling Analytic Server on Cloudera

Cloudera automatically handles most of the steps that are required to uninstall the Analytic Server service and parcel.

The following steps are required to cleanup Analytic Server from the Cloudera environment:

1. Stop and delete the Analytic Server Service.
2. **Deactivate**, **Remove From Hosts**, and **Delete** the Analytic Server parcels.
3. Delete the Analytic Server user directory in HDFS. The default location is `/user/as_user/analytic-root`.
4. Delete the database, or schema, that is used by Analytic Server.
5. Cleanup any remnants of the Analytic Server installation package. This is accomplished by deleting the following:
   - `csd` folder
   - Any existing 3.1.1.1 files located in the `parcels`, `parcel-cache`, and `parcel-repo` folders.

Chapter 4. MapR Installation and Configuration

MapR Overview

MapR is a complete distribution for Apache Hadoop that packages more than a dozen projects from the Hadoop ecosystem to provide a broad set of big data capabilities.

The MapR file system cannot be accessed outside of the server cluster. As a consequence, IBM SPSS Analytic Server must be deployed in the MapR cluster nodes. In this deployment scenario, Analytic Server must be run by a user who has authority to access the MapR file system and submit jobs to yarn to deploy to Analytic Server (as <as_user>).

Installing Analytic Server on MapR

The following steps detail the process of manually installing IBM SPSS Analytic Server on a MapR cluster.

Installing Analytic Server 3.1.1.1 on MapR 5.1

1. Navigate to the IBM Passport Advantage® Web Site and download the MapR self-extracting binary file.

Table 14. MapR self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for MapR 5.1 Linux x86-64 English	spss_as-3.1.1.1-mapr5.1-5.2-lx86.bin

2. Run the Analytic Server installer with a root or sudo user. Follow the installation prompts to accept the license and choose to install Analytic Server online or offline.
   a. Select the online option when the server that hosts Analytic Server has an internet connection to https://ibm-open-platform.ibm.com. The installer automatically installs Analytic Server.
   b. Select the offline option when the server that hosts Analytic Server does not have an internet connection to https://ibm-open-platform.ibm.com. Run the installer on another server that has access the URL and choose to install Analytic Server offline. The installer automatically downloads the RPM or DEB package.
3. Find and run the RPM or DEB for Analytic Server:
   - RedHat or SuSe Linux:

     ```
     rpm -ivh IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64.rpm
     ```

   - Ubuntu Linux:

     ```
     dpkg -i IBM-SPSS-AnalyticServer_1_amd64.deb
     ```

   For both online and offline installation modes, Analytic Server is installed to /opt/ibm/spss/analyticserver/3.1 (as <as_installation_path>).
4. Add the Analytic Server service user **as_user** to all your cluster nodes. Ensure that **as_user** has the same uid on all cluster nodes.
   a. Run the following command on a cluster node:

      ```
      useradd as_user
      ```

   b. Check the uid of **as_user** with the following command:

      ```
      id as_user
      ```

      You will receive details similar to the following:

      ```
      uid=5001(as_user) gid=5002(as_user) groups=5002(as_user)
      ```

c. Use the following command to add **as_user**, with the uid information from the previous step, to the other cluster nodes:

```
useradd -u 5001 as_user
```

5. Change all of the files in the installation path to the user who runs Analytic Server:

```
chown -R <as_user> <as_installation_path>
```

Switch the user to <as_user>; all of the proceeding steps use <as_user>.

6. Configure the HTTP property. Create a file that is named http_endpoint.xml in the path <as_installation_path>/ae_wlpserver/usr/servers/aeserver and add the following lines to the file:

```
<server>
    <httpEndpoint host="*" id="defaultHttpEndpoint" httpPort="<http_port>" httpsPort="<https_port>" onError="FAIL"/>
</server>
```

<http_port> and <https_port> are the ports that are used by Analytic Server via the HTTP and HTTPS protocols. Replace them with any available ports.

7. Configure the LDAP properties by creating a file named security_cfg.xml in the following path:

```
<as_installation_path>/ae_wlpserver/usr/servers/aeserver
```

Add the following lines to the file (select the option according to your LDAP type):

Option A: Use an external Microsoft Active Directory as the user/group repository

```
<server>
     <ldapRegistry
     id="Microsoft Active Directory"
     ldapType="Custom"
     realm="ibm"
     host="<ldap_host>"
     port="<ldap_port>"
     bindDN="cn=administrator,cn=users,dc=ldapad1,dc=com"
     bindPassword ="<password>"
     baseDN="ou=ASUsers,ou=QATest,dc=ldapad1,dc=com">
          <customFilters
          userFilter="(&amp;(uid=%v)(objectClass=inetOrgPerson))"
          groupFilter="(&amp;(cn=%v)(objectClass=groupOfNames))"
          userIdMap="*:uid"
          groupIdMap="*:cn"
          groupMemberIdMap="memberOf:member" />
     </ldapRegistry>
</server>
```

Attention:

- Replace the following attribute values with your actual system values: realm, host, port, bindDN, bindPassword, baseDN, userFilter, groupFilter, userIdMap, groupIdMap, groupMemberIdMap
- Refer to your Liberty documentation for configuration information when you use SSL between Active Directory and Analytic Server.

Option B: Use an external OpenLDAP as the user/group repository

```
<server>
     <ldapRegistry
     id="OpenLDAP"
     ldapType="Custom"
     host="<ldap_host>"
     port="<ldap_port>"
     bindDN="cn=admin,dc=ibm,dc=com"
     bindPassword="<password>"
     baseDN="dc=QATest,dc=ibm,dc=com"
     searchTimeout="300000m"
     recursiveSearch="true">
          <customFilters
          id="customFilters"
          userFilter="(&amp;(uid=%v)(objectClass=inetOrgPerson))"
          groupFilter="(&amp;(cn=%v)(objectClass=groupOfNames))"
          userIdMap="*:uid"
          groupIdMap="*:cn"
          groupMemberIdMap="groupOfNames:member"/>
     </ldapRegistry>
</server>
```

Attention:

- Replace the following attribute values with your actual system values: host, port, bindDN, bindPassword, baseDN, userFilter, groupFilter, userIdMap, groupIdMap, groupMemberIdMap

- Refer to your Liberty documentation for configuration information when you use SSL between Active Directory and Analytic Server.

Option C: Install a default Apache Directory Server to store user/group configuration. By default, Analytic Server includes an Apache Directory Server installer, but you need to install and configure it before it can be used. The configuration steps are as follows:

a. Run the following commands:

```
cd <as_installation_path>
cp ./tools/ads.zip ./
unzip ads.zip
rm -f ads.zip
chmod 777 ./ads/bin/apacheds.sh
./ads/bin/apacheds.sh start
java -Duser.language=en -cp <as_installation_path>/ads/lib/*.jar com.spss.ae.embeddedADS.AdsOper <hostname> initData
java -Duser.language=en -cp <as_installation_path>/ads/lib/*.jar com.spss.ae.embeddedADS.AdsOper <hostname> disableAnonymous
```

Note: When a java command is run directly in a Linux shell, the character * might need to be escaped as *.

b. For security considerations, Apache Directory Server and Analytic Server communicate via SSL (by default). Run the following commands:

```
cd <as_installation_path>/ads
mkdir private
mkdir public
cd private
keytool -genkey -keyalg "RSA" -dname "cn=asads, ou=SPSS, o=IBM, c=US" -alias asads -keystore adsserver.ks -storepass spssas -keypass spssas -validity 36
keytool -export -alias asads -file ../public/asads.cer -keystore adsserver.ks -storepass spssas
cd ../public
keytool -import -file asads.cer -alias asads -keystore trustads.jks -storepass changeit -noprompt
cd ..
java -Duser.language=en -cp <as_installation_path>/ads/lib/*.jar com.spss.ae.embeddedADS.AdsOper <hostname> selfCer <as_installation_path>/tools
java -Duser.language=en -cp <as_installation_path>/ads/lib/*.jar com.spss.ae.embeddedADS.AdsOper <hostname> onlyLDAPs
java -Duser.language=en -cp <as_installation_path>/ads/lib/*.jar com.spss.ae.embeddedADS.AdsOper <hostname> changeAdminPwd <your_password>
<as_installation_path>/ads/bin/apacheds.sh stop
<as_installation_path>/ads/bin/apacheds.sh start
```

c. Update the `security_cfg.xml` configuration file.

```
<server>
     <ldapRegistry
     id="ads"
     ldapType="Custom"
     host="<hostname>"
     port="10636"
     bindDN="uid=admin,ou=system"
     bindPassword="<password>"
     baseDN="dc=ibm,dc=com"
     searchTimeout="300000m"
     recursiveSearch="true"
     sslEnabled="True"
     sslRef="LDAPSSLSettings">
          <customFilters id="customFilters"
          userFilter="(&amp;(cn=%v)(objectClass=organizationalPerson))"
          groupFilter="(&amp;(cn=%v)(objectclass=groupOfNames))"
          userIdMap="*:cn"
          groupIdMap="*:cn"
          groupMemberIdMap="groupOfNames:member"/>

          <ldapEntityType name="PersonAccount">
               <objectClass>organizationalPerson</objectClass>
          </ldapEntityType>
     </ldapRegistry>
     <ssl id="LDAPSSLSettings" keyStoreRef="LDAPTrustStore" trustStoreRef="LDAPTrustStore" />
     <keyStore id="LDAPTrustStore" location="<as_installation_path>/ads/public/trustads.jks" type="JKS" password="changeit" />
</server>
```

Attention: Replace the following attribute values with your actual system values: `host`, `bindPassword`. `bindPassword` is the value you set in step b. You can set plain value, or use the Liberty encryption utility to encrypt it. The Liberty encryption utility can be called with the following command:

`<as_installation_path>/ae_wlpserver/bin/securityUtility encode <your_password>`

d. Append the following properties in the `config.properties` file (`<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`):

```
embeddedads.type=ads
embeddedads.host=<hostname>
embeddedads.port=10636
embeddedads.password=<encrypted_password>
```

Notes:

- `embeddedads.host` is the LDAP hostname
- `embeddedads.port` is the LDAP server port. If the default Apache Directory Server configuration is used the port number is 10636.
- `embeddedads.password` is the encrypted password you set in step b. The encryption value can be retrieved with the following utility (this utility is different than the utility from step c):

  ```
  java -Duser.language=en -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/* com.spss.ae.encryption.provider.EncryptK
  ```

8. Setup the metadata database. Analytic Server supports the Db2 and MySQL databases.
   a. Configure the database users. When the MySQL database is used, run the following SQL script in the MySQL shell:

   ```
   DROP DATABASE IF EXISTS <db_name>;
   CREATE DATABASE <db_name> DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;
   CREATE USER '<db_username>'@'%' IDENTIFIED BY '<db_password>';
   CREATE USER '<db_username>'@'localhost' IDENTIFIED BY '<db_password>';
   GRANT ALL PRIVILEGES ON *.* TO '<db_username>'@'%';
   GRANT ALL PRIVILEGES ON *.* TO '<db_username>'@'localhost';
   ```

   b. Encrypt the password. The database users' passwords must be encrypted before it can be passed to Analytic Server. Run the following command:

   ```
   java -Duser.language=en -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/*
   com.spss.ae.encryption.provider.EncryptKeystorePassword <db_password>
   ```

   Note: When the command is run directly in a Linux shell, the character `*` might need to be escaped as `\*`.

   The command output reads as: `The encrypted password is '<encrypted_db_password>'`. Record the encrypted database password.

   c. Delete the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`, if it exists, and create a new file with the same name. Change the following properties when the Db2 database is used:

   ```
   jndi.aedb=jdbc/aeds
   jndi.aedb.url=jdbc:db2://<db_host>:<db_port>/<db_name>:currentSchema=<db_schema_name>;
   jndi.aedb.driver=com.ibm.db2.jcc.DB2Driver
   jndi.aedb.username=<db_username>
   jndi.aedb.password=<encrypted_db_password>
   ```

   If the `<db_schema_name>` schema does not exist, the user `<db_username>` must have implicit permission to create the schema. Change the following properties when the MySQL database is used:

   ```
   jndi.aedb=jdbc/aeds
   jndi.aedb.url=jdbc:mysql://<db_host>:<db_port>/<db_name>?createDatabaseIfNotExist=true
   jndi.aedb.driver=com.mysql.jdbc.Driver
   jndi.aedb.username=<db_username>
   jndi.aedb.password=<encrypted_db_password>
   ```

   d. The MySQL JDBC driver must be installed when the MySQL database is used. Run the following command:

   ```
   yum install mysql-connector-java
   ```

   e. Run the following command the create the required tables:

   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/sql/<db_type>
   java -Xmx128m -Xms128m -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/*:/usr/share/java/*
   com.spss.ae.dbscript.ScriptRunner ../../configuration/config.properties schema.sql true
   ```

   The `<db_type>` is either `db2` or `mysql`, depending on which database is used.

   Note: When MySQL with the MYISAM engine is used, the second command reports the following error messages, which can be safely ignored:

   ```
   Error executing: set global innodb_large_prefix=ON
   java.sql.SQLException: Unknown system variable 'innodb_large_prefix'
   Error executing: set global innodb_file_format=BARRACUDA
   java.sql.SQLException: Unknown system variable 'innodb_file_format'
   Error executing: set global innodb_file_format_max=BARRACUDA
   java.sql.SQLException: Unknown system variable 'innodb_file_format_max'
   Error executing: set global innodb_file_per_table=TRUE
   java.sql.SQLException: Variable 'innodb_file_per_table' is a read only variable
   ```

9. Run the following command to unpack the `cf` library.

```
cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration
unzip cf.zip
```

10. Configure the JAAS login modules class path by creating a file that is named `private_library.xml` in the path`<as_installation_path>/ae_wlpserver/usr/servers/aeserver` and enter the following information in the file:

    ```
    <server>
     <library id="maprLib">
      <fileset dir="${wlp.install.dir}/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib" includes="*.jar"/>
      <fileset dir="/usr/share/java" includes="*.jar"/>
      <folder dir="/opt/mapr/hadoop/hadoop-2.7.0/etc/hadoop"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/lib" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/hdfs" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/hdfs/lib" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/yarn" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/yarn/lib" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce" includes="*.jar"/>
      <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce/lib" includes="*.jar"/>
     </library>
     <jaasLoginModule id="maprLoginModule1" className="org.apache.hadoop.security.login.GenericOSLoginModule"
      controlFlag="REQUIRED" libraryRef="maprLib"></jaasLoginModule>
     <jaasLoginModule id="maprLoginModule2" className="org.apache.hadoop.security.login.HadoopLoginModule"
      controlFlag="REQUIRED" libraryRef="maprLib"></jaasLoginModule>
     <jaasLoginContextEntry id="hadoop_simple" name="hadoop_simple" loginModuleRef="maprLoginModule1,maprLoginModule2" />
     <application context-root="/analyticserver" id="AS_BOOT" location="AE_BOOT.war" name="AS_BOOT" type="war">
       <classloader commonLibraryRef="maprLib"></classloader>
     </application>
     <application id="help" location="help.war" name="help" type="war" context-root="/analyticserver/help"/>
    </server>
    ```

    Note: The previous example is for configuring the `hadoop_simple login` module. The configuration must be changed when MapR uses other login modules.

11. Verify if the file `ASModules.xml` exists in the path `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/`. If the file does not exist, rename the file `ASModules.xml.template` (in the same path) to `ASModules.xml`

12. Configure the cluster information by adding the following properties in the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`.

    ```
    ae.cluster.zookeeper.connect.string=
    ae.cluster.member.name=
    ae.cluster.collective.name=mapr_5.1
    ```

    The `ae.cluster.zookeeper.connect.string` property is the comma-separated zookeeper node list. The property can share the zookeeper cluster that is used by MapR. `ae.cluster.member.name` is the host name of the node that hosts Analytic Server.

    The following example demonstrates the `ae.cluster.zookeeper.connect.string` format:

    ```
    ae.cluster.zookeeper.connect.string=<zookeeper host 1>:<zookeeper port 1>,
    <zookeeper host 2>:<zookeeper port 2>,<zookeeper host 3>:<zookeeper port 3>...
    ```

    When Analytic Server shares the same zookeeper cluster with MapR, the `ae.cluster.zookeeper.connect.string` value must be the same as the `zookeeper.servers` property in the MapR `warden.conf` file (the file's default location is `/opt/mapr/conf`).

13. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` and add the following lines to the file:

    ```
    JAVA_HOME=<java_home>

    PATH=<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:<java_home>/jre/lib/amd64:/usr/sbin:/usr/bin:/sbin:/bin

    IBM_SPSS_AS_NATIVE_PATH=<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64

    LD_LIBRARY_PATH=<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:<java_home>/jre/lib/amd64:/opt/mapr/hadoop/hadoop-2.7.0/lib/native
    ```

    Replace `<as_installation_path>` and `<java_home>` with the actual installation path and Java home path.

14. Edit the analytic root by opening the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` and adding the following line:

    ```
    distrib.fs.root=<analytic_root>
    ```

    `<analytic_root>` is a path in the MapR file system that hosts the essential Analytic Server remote files. The recommended path is `/user/<as_user>/analytic-root`.

15. Set the administrator user by opening the file <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties and adding the following line:

    ```
    admin.username=admin
    ```

    The value must be an Analytic Server administrator user name and must be one of the users that is configured in the security_cfg.xml file.

16. Upload Analytic Server dependencies to the MapR file system by adding the following line at line 69 in the file <as_installation_path>/bin/hdfsUpdate.sh:

    ```
    JAVA_CLASS_PATH=`hadoop classpath`:$JAVA_CLASS_PATH
    ```

    Run the following commands to create the <analytic_root>:

    ```
    cd <as_installation_path>/bin
    ./hdfsUpdate.sh
    ```

    <as_user> must have write permission to the <analytic_root> parent directory.

17. Start and stop Analytic Server.

    a. Run the following command to start Analytic Server:

    ```
    cd <as_installation_path>/ae_wlpserver/bin
    ./server start aeserver
    ```

    b. Run the following command to stop Analytic Server:

    ```
    cd <as_installation_path>/ae_wlpserver/bin
    ./server stop aeserver
    ```

Installing Analytic Server 3.1.1.1 on MapR 5.2

1. Navigate to the IBM Passport Advantage® Web Site and download the MapR self-extracting binary file.

Table 15. MapR self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1.1.1 for MapR 5.2 Linux x86-64 English	spss_as-3.1.1.1-mapr5.1-5.2-lx86.bin

2. The remaining steps for installing Analytic Server are mostly the same as those for installing Analytic Server 3.1.1.1 on MapR 5.1. However, the "Enabling Apache HBase" on page 64 and "Enabling Apache Spark" on page 65 information differs between MapR 5.1 and 5.2. Refer to those topics for information regarding installing on MapR 5.2.

Configuring MapR

After installation, you can optionally configure and administer Analytic Server MapR features.

Enabling database pushback

Database pushback is the practice of reading data from a database and processing directly on the data.

IBM SPSS Analytic Server supports pushback for the following databases:

- DashDB
- Db2
- Db2 for Z
- Hive
- MySQL
- Netezza
- Oracle
- PostgreSQL
- Redshift
- SQL Server

- Terradata

Use the following steps to enable database pushback.

1. Copy the appropriate JDBC driver JAR files to `<as_installation_path>/jdbc`.
2. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml`, locate the tag library with the `maprLib` ID, and add the following line in the tag:
   ```
   <fileset dir="<as_installation_path>/jdbc" includes="*.jar"/>
   ```
3. Run the following commands:
   ```
   cd <as_installation_path>/jdbc
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```
4. Restart Analytic Server.

Enabling Apache Hive

Apache Hive is a data warehouse infrastructure that is built on top of Hadoop for providing data summarization, query, and analysis.

Note: Hive must be configured to use MySQL as a metastore. The `hive-site.xml` file that exists in the node that hosts IBM SPSS Analytic Server should be same as the file in the node that runs the Hive metastore.

To enable Apache Hive support after a successful MapR installation:

1. Upload the Hive and hcatalog dependencies to the MapR file system, by running the following commands:
   ```
   cd /opt/mapr/hive/hive-2.1/lib
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   cd /opt/mapr/hive/hive-2.1/hcatalog/share/hcatalog
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```
   `<as_analytic_root>` is the analytic root path that is defined in "Installing Analytic Server on MapR" on page 57.
2. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml`, locate the tag library with the ID `maprLib`, and add the following lines in the tag:
   ```
   <fileset dir="/opt/mapr/hive/hive-2.1/lib" includes="*.jar"/>
   <fileset dir="/opt/mapr/hive/hive-2.1/hcatalog/share/hcatalog" includes="*.jar"/>
   ```
3. Run the following commands to create Hive and hcatalog configuration file links:
   ```
   mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hive-conf
   ln -s /opt/mapr/hive/hive-2.1/conf/* <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hive-conf
   ```
4. Add the following line to the `private_library.xml` file when there are extra jar files in the auxlib for Hive:
   ```
   <fileset dir="/opt/mapr/hive/hive-2.1/auxlib" includes="*.jar"/>
   ```
   Run the following commands after adding the previous line:
   ```
   cd /opt/mapr/hive/hive-2.1/auxlib
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```
5. Restart Analytic Server.

Running Hive in HTTP mode

By default, Hive runs in binary mode (TCP mode). To run Hive in HTTP mode, you must update the following Hive configuration properties (in particular the `hive.server2.transport.mode` property).

Note: For more information regarding each property, see Hive Configuration Properties.

Table 16. Hive properties for HTTP mode

Property name	Default value	Description
hive.server2.transport.mode	binary	The server transport mode. The value can be `binary` or `http`. Set to `http` to enable HTTP transport mode.
hive.server2.thrift.http.port	10001	The port number when in HTTP mode.
hive.server2.thrift.http.path	cliservice	The path component of the URL endpoint when in HTTP mode.
hive.server2.thrift.http.min.worker.threads	5	The minimum number of worker threads, in the server pool, when in HTTP mode.
hive.server2.thrift.http.max.worker.threads	500	The maximum number of worker threads, in the server pool, when in HTTP mode.

Note: Hive must be restarted after the properties are updated.

Enabling Apache HBase

Apache HBase is an open source, non-relational, distributed database that is written in Java. It is developed as part of Apache Software Foundation's Apache Hadoop project and runs on top of HDFS (Hadoop Distributed Filesystem).

To enable Apache HBase support after a successful MapR installation:

IBM SPSS Analytic Server 3.1.1 on MapR 5.1

1. Upload the HBase dependencies to the MapR file system and run the following commands:

   ```
   cd /opt/mapr/hbase/hbase-0.98.12/lib
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```

   `<as_analytic_root>` is the analytic root path that is defined in “Installing Analytic Server on MapR” on page 57.
2. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml`, locate the tag library with the ID `maprLib`, and add the following line in the tag:

   ```
   <fileset dir="/opt/mapr/hbase/hbase-0.98.12/lib" includes="*.jar"/>
   ```
3. Run the following commands to create HBase and hcatalog configuration file links:

   ```
   mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hbase-conf
   ln -s /opt/mapr/hbase/hbase-0.98.12/conf/* <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hbase-conf
   ```
4. Restart IBM SPSS Analytic Server.

IBM SPSS Analytic Server 3.1.1 on MapR 5.2

1. Upload the HBase dependencies to the MapR file system by executing the following commands:

   ```
   cd /opt/mapr/hbase/hbase-1.1.1/lib
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```

   The `<as_analytic_root>` is the path that is set in the 12th step in “Installing Analytic Server on MapR” on page 57.
2. Open `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml` and locate the tag library with the ID `maprLib`. Add the following line to the tag:

   ```
   <fileset dir="/opt/mapr/hbase/hbase-1.1.1/lib" includes="*.jar"/>
   ```
3. Execute the following commands to create links for the Hive and HCatalog configuration files:

   ```
   mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hbase-conf
   ln -s /opt/mapr/hbase/hbase-1.1.1/conf/* <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hbase-conf
   ```
4. Add the following line to `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`

   ```
   spark.executor.extraClassPath=/opt/mapr/hbase/hbase-1.1.1/lib/*
   ```
5. Restart Analytic Server.

Enabling Apache Spark

Apache Spark is an open standard for flexible in-memory data processing for batch, real-time, and advanced analytics.

To enable Apache Spark support after a successful MapR installation:

IBM SPSS Analytic Server 3.1.1.1 on MapR 5.1

1. Copy the file `spark-assembly-1.4.1-hadoop2.5.1-mapr-1501.jar` from `/opt/mapr/spark/spark-1.4.1/lib` to `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/`.
2. Upload the Spark dependencies to the MapR file system and run the following commands:
   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```
   `<as_analytic_root>` is the analytic root path that is defined in "Installing Analytic Server on MapR" on page 57.
3. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml`, locate the tag library with the ID `maprLib`, and add the following line in the tag:
   ```
   <fileset dir="/opt/mapr/spark/spark-1.4.1/lib" includes="spark-assembly-*.jar"/>
   ```
4. Run the following commands to create Spark configuration file links:
   ```
   mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/spark-conf
   ln -s /opt/mapr/spark/spark-1.4.1/conf/* <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/spark-conf
   ```
5. Add the following line in the file `<as_installation_path/ae_wlpserver/usr/servers/aeserver/server.env`:
   ```
   SPARK_HOME=/opt/mapr/spark/spark-1.4.1
   ```
6. Add the following line in the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`:
   ```
   spark.executor.extraLibraryPath=/opt/mapr/hadoop/hadoop-2.7.0/lib/native
   ```
7. Restart IBM SPSS Analytic Server.
8. To enable the PySpark feature, add the following line in the `yarn-env.sh` file, and then restart ResourceManagers and NodeManagers:
   ```
   export SPARK_HOME=/opt/mapr/spark/spark-1.4.1
   ```

IBM SPSS Analytic Server 3.1.1.1 on MapR 5.2

The steps differ depending on the Spark version.

Spark 1.x

1. Copy the file spark-assembly-1.4.1-hadoop2.5.1-mapr-1501.jar from:
   ```
   /opt/mapr/spark/spark-1.4.1/lib
   ```
   to
   ```
   <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/
   ```
2. Open the following file:
   ```
   <as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml
   ```
   and locate the tag library with the ID `maprLib`. Add the following line to the tag:
   ```
   <fileset dir="/opt/mapr/spark/spark-1.4.1/lib" includes="spark-assembly-*.jar"/>
   ```
   Note: The newly added line must be placed above the other `maprLib` child tags.
3. Delete the following file:
   ```
   <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/com.ibm.spss.sparkmapreduce_2-3.1.1.1.jar
   ```

Spark 2.x

1. Delete the following file:

```
<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/com.ibm.spss.sparkmapreduce-3.1.1.1.jar
```

2. Open the following file:

   `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml`

   a. Locate the tag library with the ID `maprLib`. Add the following lines to the library:

   ```
   <fileset dir="/opt/mapr/spark/spark-2.0.1/jars" includes="*.jar"/>
   <fileset dir="/opt/mapr/spark/spark-2.0.1/scala/lib" includes="*.jar"/>
   <fileset dir="<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark" includes="*.jar"/>
   <fileset dir="${wlp.install.dir}/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib" includes="*.jar"/>
   ```

   Note: The newly added lines must be placed above the other `maprLib` child tags and the last line must be under the other three lines.

   b. Locate the tag `<fileset dir="/opt/mapr/hive/hive-1.2/lib" includes="*.jar"/>` and update it to:

   ```
   <fileset dir="/opt/mapr/hive/hive-1.2/lib" includes="*.jar" excludes="jackson-*.jar,derby-*.jar"/>
   ```

   c. Locate the tag `<fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/lib" includes="*.jar"/>` and update it to:

   ```
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/lib" includes="*.jar" excludes="jackson-*.jar"/>
   ```

   Note: After updating the tags, the fileset tags in `private_library.xml` should resemble the following:

   ```
   <fileset dir="/opt/mapr/spark/spark-2.0.1/jars" includes="*.jar"/>
   <fileset dir="/opt/mapr/spark/spark-2.0.1/scala/lib" includes="*.jar"/>
   <fileset dir="/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/modules/spark" includes="*.jar"/>
   <fileset dir="${wlp.install.dir}/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib" includes="*.jar"/>
   <fileset dir="/opt/ibm/spss/analyticserver/3.1/jdbc" includes="*.jar"/>
   <fileset dir="/opt/mapr/hive/hive-1.2/lib" includes="*.jar" excludes="jackson-*.jar,derby-*.jar"/>
   <fileset dir="/opt/mapr/hive/hive-1.2/hcatalog/share/hcatalog" includes="*.jar"/>
   <fileset dir="/opt/mapr/hive/hive-1.2/auxlib" includes="*.jar"/>
   <fileset dir="/opt/mapr/hbase/hbase-1.1.8/lib" includes="*.jar"/>
   <fileset dir="/usr/share/java" includes="*.jar"/>
   <folder dir="/opt/mapr/hadoop/hadoop-2.7.0/etc/hadoop"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/lib" includes="*.jar" excludes="jackson-*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/hdfs" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/hdfs/lib" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/yarn" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/yarn/lib" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce" includes="*.jar"/>
   <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce/lib" includes="*.jar"/>
   ```

3. Run the following command to move the file `xgboost4j-spark-0.7-jar-with-dependencies.jar` from `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib` to`<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark`:

   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark
   mv <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/xgboost4j-spark-0.7-jar-with-dependencies.jar
   ```

4. Add the line `spark.version=2.0` to the following file:

   `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`

The following steps are common to both Spark 1.x and 2.x, and must be completed after the previous Spark 1.x or 2.x steps.

Note: All references to `<spark_version>` must be replaced with the actual spark version (for example `1.4.1` or `2.0.1`).

1. Upload the Spark dependencies to the MapR file system by executing the following commands:

   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/
   hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
   ```

   The `<as_analytic_root>` is the path that is set in the 12th step in "Installing Analytic Server on MapR" on page 57.

2. Execute the following commands to create links for the Spark configure file:

   ```
   mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/spark-conf
   ln -s /opt/mapr/spark/spark-<spark_version>/conf/*
   <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/spark-conf
   ```

3. Add the following line to the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env`

```
SPARK_HOME=/opt/mapr/spark/spark-<spark_version>
```

4. Add the following line in the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`

   ```
   spark.executor.extraLibraryPath=/opt/mapr/hadoop/hadoop-2.7.0/lib/native
   ```

5. Restart Analytic Server.
6. If need to enable the PySpark feature, add the following lines in the `yarn-env.sh` file:

   ```
   export SPARK_HOME=/opt/mapr/spark/spark-<spark_version>
   ```

   Restart ResourceManagers and NodeManagers.

Enabling feature flags

Feature flags provide the capability of enabling and disabling specific application features.

To enable feature flag support after a successful MapR installation:

1. Add the following line in the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`:

   ```
   load.feature.flags.on.msg=true
   ```

2. Restart IBM SPSS Analytic Server.

Enabling R

R is a language and environment for statistical computing and graphics.

To enable R support after a successful MapR installation:

Note: The following package must be installed before you can run the installer on all cluster nodes that host Node Manager and IBM SPSS Analytic Server:

```
gcc-gfortran
libgfortran
gcc-c++
```

1. Provision the server environment for Essentials for R. For more information, see step 1 in “Enabling Support for Essentials for R” on page 19.
2. Run the installer `spss_er-8.5.0.0-mapr5-lx86_64.bin` on all the cluster nodes that host Node Manager and Analytic Server. The user that runs the installer must have write permission to the R and Analytic Server installation paths.
3. Follow the installation instructions by accepting the license agreement and enter the required information. If Analytic Server is installed on the installation server, choose `Yes` when prompted and input `<as_installation_path>`. If Analytic Server is not installed on the installation server, choose `No` when prompted.
4. When Analytic Server is installed, Essentials for R is automatically installed in the Analytic Server installation path.
   - If Analytic Server is not installed, Essentials for R is installed to the path `<installer_path>/IBM_SPSS_ModelerEssentialsR/linux`.
   - If Analytic Server is installed later, use the following command to copy Essentials for R to the Analytic Server configuration path where Analytic Server is installed.

     ```
     cp -r <installer_path>/IBM_SPSS_ModelerEssentialsR/linux <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration
     ```

5. Delete the `cf.zip` file in the path `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration` and generate a new file with the following commands:

   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration
   zip -r cf.zip linux
   ```

6. Run the following commands:

   ```
   cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration
   hadoop fs -rm <as_analytic_root>/defaultASsubpath/configuration/cf.zip
   hadoop fs -put cf.zip <as_analytic_root>/defaultASsubpath/configuration/
   ```

7. Restart Analytic Server.

Enabling LZO

LZO is a lossless data compression library that favors speed over compression ratio. MapR must be manually configured to provide LZO support.

The following site provides LZO installation and configuration instructions: https://github.com/twitter/hadoop-lzo.

The following steps detail the process of importing an LZO library into MapR.

1. Copy the `hadoop-lzo-<version>.jar` file to the Hadoop class path. The recommended path is `/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce/lib`.
2. Copy the native files `libgplcompression.so` and `liblzo2.so.2` to `/opt/mapr/hadoop/hadoop-2.7.0/lib/native`, and add the following properties to the `core-site.xml` file:

   ```
   <property>
      <name>io.compression.codecs</name>
      <value>org.apache.hadoop.io.compress.GzipCodec,org.apache.hadoop.io.compress.DefaultCodec,com.hadoop.compression.lzo.
       LzoCodec,com.hadoop.compression.lzo.LzopCodec,org.apache.hadoop.io.compress.BZip2Codec</value>
   </property>
   <property>
       <name>io.compression.codec.lzo.class</name>
       <value>com.hadoop.compression.lzo.LzoCodec</value>
   </property>
   ```
3. Open the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` and add `<lzo_native_path>` to the `LD_LIBRARY_PATH` parameter. `<lzo_native_path>` is the folder that contains the Hadoop-LZO native library.

   ```
   LD_LIBRARY_PATH=<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:<java_home>/jre/lib/amd64:
   /opt/mapr/hadoop/hadoop-2.7.0/lib/native:<lzo_native_path>
   ```
4. Restart IBM SPSS Analytic Server.

Enabling SLM tags for MapR

SLM tags are based on the ISO/IEC 19770-4 standard draft for Resource Utilization Measurement. SLM tags provide a standardized capability for a product to report its consumption of license metrics (resources related to the use of a software asset). After enabling SLM in a product, an runtime XML file is generated to self-report its license use.

To enable SLM tags in MapR, you must create a `SlmTagOutput.properties` file in the `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration` folder. The file must contain the following content:

```
license.metric.logger.output.enabled=true
license.metric.logger.softwareid=5d2b4d9dae05494cbfaf676add5f4d30
license.metric.logger.output.dir=slmtag
license.metric.logger.output.SLMLogFrequency=43200000
license.metric.logger.file.size=2048000
license.metric.logger.file.number=10
```

Configuring MapReduce v2 jobs

Feature flags provide the capability of enabling and disabling specific application features.

Use the **`preferred.mapreduce`** setting in the `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` file to control how MapReduce jobs are handled:

Table 17. preferred.mapreduce settings

Property	Description
preferred.mapreduce	Controls the method in which MapReduce jobs are run. Valid values include: • spark • m3r • hadoop For example: preferred.mapreduce=spark

Setting up an IBM SPSS Analytic Server cluster for MapR

Use the following steps to setup an IBM SPSS Analytic Server cluster environment for MapR support.

1. Add the following line in the file `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties`.

   ```
   enable.resume=true
   ```

2. Copy the installation path to the other cluster nodes and change the `ae.cluster.member.name` property in the `config.properties` file to the correct host name.
3. Start all of the cluster nodes.

Uninstalling MapR

The following steps explain the process of uninstalling MapR:

1. Stop IBM SPSS Analytic Server.
2. Delete the metadata database.
   a. Run the following commands:

      ```
      cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/sql/<db_type>
      java -Xmx128m -Xms128m -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/*:/usr/share/java/*
      com.spss.ae.dbscript.ScriptRunner ../../configuration/config.properties drop.sql true
      ```

   b. Run the following SQL statement to drop the database:

      ```
      drop database <db_name>
      ```

3. Uninstall the RPM package:

   ```
   rpm -e IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64
   ```

4. Delete the installation path:

   ```
   rm -r <as_installation_path>
   ```

5. Delete the analytic root:

   ```
   hadoop fs -rm -r <analytic-root>
   ```

6. Delete the zookeeper data:

   ```
   /opt/mapr/zookeeper/zookeeper-3.4.5/bin/zkCli.sh -server <zookeeper_host>:<zookeeper_port>
   rmr /AnalyticServer
   ```

Migrating IBM SPSS Analytic Server on MapR

IBM SPSS Analytic Server can be migrated on MapR.

Use the following steps to migrate IBM SPSS Analytic Server 3.1 to version 3.1.1.1 on MapR.

1. Install Analytic Server 3.1.1.1 on a MapR cluster by following the installation instructions in "Installing Analytic Server on MapR" on page 57.
2. Copy the analytic root.

   Note: This step can be ignored if the analytic root is not changed.

   - Run the following command on one of the data nodes if the analytic root for both Analytic Server versions 3.1 and 3.1.1.1 are on the same MapR cluster:

```
hadoop fs -cp <old_analytic_root>/analytic-workspace/* <new_analytic_root>/analytic-workspace
```

- The installed WEBHDFS or NFS services dictate when the analytic root for Analytic Server versions 3.1 and 3.1.1.1 are on different MapR clusters. WEBHDFS or NFS are required to copy the analytic root data because the MapR file system cannot be accessed directly outside the cluster.
  a. Run the following command on one of the new Analytic Server 3.1.1.1 cluster nodes when the old Analytic Server 3.1 cluster includes the WEBHDFS service:
  ```
  hadoop distcp webhdfs://<webhdfs_server>:<webhdfs_port>/<old_analytic_root>/analytic-workspace/*
  maprfs://<new_analytic_root>/analytic-workspace
  ```
  b. Run the following command on one of the old Analytic Server 3.1 cluster nodes when the new Analytic Server 3.1.1.1 cluster includes the WEBHDFS service:
  ```
  hadoop distcp maprfs://<old_analytic_root>/analytic-workspace/*
  webhdfs://<webhdfs_server>:<webhdfs_port>/<new_analytic_root>/analytic-workspace
  ```
  c. Run the following command on one of the old Analytic Server 3.1 cluster nodes when the old cluster includes NFS, and NFS is also mounted on one of the new Analytic Server 3.1.1.1 cluster nodes:
  ```
  hadoop distcp file:///<mount_path>/<old_analytic_root>/analytic-workspace/* maprfs://<new_analytic_root>/analytic-workspace
  ```
  d. Run the following command on one of the new Analytic Server 3.1.1.1 cluster nodes when the new cluster includes NFS, and NFS is also mounted on one of the old Analytic Server 3.1 cluster nodes:
  ```
  hadoop discp maprfs://<old_analytic_root>/analytic-workspace/* file:///<mount_path>/<new_analytic_root>/analytic-workspace
  ```

Review the MapR Data Migration site for information on migrating data between different MapR clusters.

3. Run the following commands to change the new analytic root's owner and permissions:
```
hadoop fs -chown -R <as_user> <analytic_root>
hadoop fs -chmod -R 755 <>
```
4. Stop Analytic Server 3.1.1.1, but ensure that the metadata database is still running.
5. Collect the configuration settings from the old Analytic Server 3.1 cluster installation.
   a. Copy the `configcollector.zip` archive from the new Analytic Server 3.1.1.1 cluster installation to `<old_as_installation_path>/tools` on the old Analytic Server 3.1 cluster installation.
   b. Extract the `configcollector.zip` contents on the old Analytic Server 3.1 cluster installation. A new `configcollector` subdirectory is created in the old Analytic Server 3.1 cluster installation.
   c. Run the configuration collector tool in the old Analytic Server 3.1 cluster installation by running the `configcollector` script from `<old_as_installation_path>/tools/configcollector`. Copy the resulting compressed (ZIP) file to the new Analytic Server 3.1.1.1 cluster installation.

   Important: The provided **`configcollector`** script may not be compatible with the most recent Analytic Server version. Contact you IBM technical support representative if you encounter problems with the **`configcollector`** script.
6. Run the migration tool on the new Analytic Server 3.1.1.1 cluster by running the `migrationtool` script and passing the path of the compressed file, that was created by the configuration collector, as an argument. For example:
```
migrationtool.sh /opt/ibm/spss/analyticserver/3.1/ASConfiguration_3.1.1.1.xxx.zip
```
7. Start Analytic Server 3.1.1.1.

MapR troubleshooting

This section describes some common MapR installation and configuration issues and how you can fix them.

Issues with the hdfsUpdate.sh script

The hdfsUpdate.sh script must be run only one time because the script removes all files in the analytic-root before it uploads new files. When the script is run more than one time, you must re-upload the dependencies for database push-back, Hive, HBase, and Spark. Run the following commands to re-upload the required dependencies:

```
cd <as_installation_path>/jdbc

hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath

cd /opt/mapr/hive/hive-2.1/lib
hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
cd /opt/mapr/hive/hive-1.2/hcatalog/share/hcatalog
hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath

cd /opt/mapr/hbase/hbase-0.98.12/lib
hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath

cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/

hadoop fs -put *.jar <as_analytic_root>/defaultASsubpath/classpath
```

A conflict between MapR and Spark versions results in failed Spark job execution

A class conflict issue occurs between MapR and Spark (1.6.1) when the MapR version is 5.1 or later. The conflict results in failed Spark job execution. You can resolve the issue by modifying the private_library.xml file in <as_installation_path>/ae_wlpserver/usr/servers/aeserver. The following example identifies the required change:

```
  ......
  <fileset dir="/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/common/lib" includes="*.jar" excludes="jackson-databind-*.jar" />
  ......
```

Chapter 5. Huawei FusionInsight HD Installation and Configuration

FusionInsight HD overview

Huawei FusionInsight HD provides a comprehensive Big Data software platform for batch and real-time analytics using open-source Hadoop and Spark technologies. The system leverages HDFS, HBase, MapReduce, and YARN/Zookeeper for Hadoop clustering, along with Apache Spark for faster real-time analytics and interactive queries.

Analytic Server can run on the FusionInsight HD platform. FusionInsight HD contains the main, core elements of Hadoop that provide reliable, scalable distributed data processing of large data sets (chiefly MapReduce and HDFS), as well as other enterprise-oriented components that provide security, high availability, and integration with hardware and other software.

Installation on Huawei FusionInsight HD

The following steps explain the process of manually installing IBM SPSS Analytic Server in Huawei FusionInsight HD.

Analytic Server 3.1.1.1

1. Navigate to the IBM Passport Advantage® Web Site and download the following self-extracting binary file to a host within the FusionInsight HD cluster.

Table 18. Analytic Server self-extracting binary file

Description	Binary file name
IBM SPSS Analytic Server 3.1.1.1 for FusionInsight HD 2.6 Linux x86-64 English	spss_as-3.1.1.1-fhd2.6-lx86.bin

2. Run the self-extracting *.bin installer on the FusionInsight Manager master cluster node. Follow the installation prompts by accepting the license agreement and keeping the default installation directory. The installer downloads the necessary RPM files, and must be run on a computer that can access https://ibm-open-platform.ibm.com. The executable binary file is located in the available FusionInsight HD <AS_INSTALLABLE_HOME> distribution directory.
3. Use the following command to install Analytic Server 3.1.1.1:

```
# yum install -y IBM-SPSS-AnalyticServer-3.1.1.1-1.x86_64.rpm
```

4. Log in with omm and create the analyticserver.keytab:

```
# su omm
# source /opt/huawei/Bigdata/om-0.0.1/meta-0.0.1-SNAPSHOT/kerberos/scripts/component_env
# kadmin -p kadmin/admin
```

The default kadmin password is Admin@123. You must change the password upon first use. In the following commands, replace _HOST with your host name.

```
kadmin > addprinc -randkey omm/_HOST@HADOOP.COM
          kadmin > ktadd -k  /opt/ibm/spss/analyticserver/3.1/analyticserver.keytab HTTP/_HOST@HADOOP.COM
          kadmin > ktadd -k /opt/ibm/AnalyticServer/analyticserver.keytab omm/_HOST@HADOOP.COM
```

5. Install MYSQL and manually create the aedb. For example:

```
# cd /etc/yum.repos.d
# wget http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
# yum -y install mysql57-community-release-el7-9.noarch.rpm
# yum repolist all | grep mysql
# yum -y install mysql-community-server
# yum install -y mysql-connector-java
# systemctl enable  mysqld.service
# systemctl start  mysqld.service
```

Retrieve the MYSQL root user password:

```
# grep 'temporary password' /var/log/mysqld.log
   # mysql -uroot -p
   # MySQL> set global validate_password_policy=0;
         # MySQL> DROP DATABASE IF EXISTS aedb;
              # MySQL> CREATE DATABASE aedb  DEFAULT CHARACTER SET utf8 DEFAULT COLLATE utf8_bin;
                  # MySQL> CREATE USER 'aeuser'@'%' IDENTIFIED BY 'Pass1234';
                  # MySQL> CREATE USER 'aeuser'@'localhost' IDENTIFIED BY 'Pass1234';
                  # MySQL> GRANT ALL PRIVILEGES ON *.* TO 'aeuser'@'%';
                  # MySQL>  GRANT ALL PRIVILEGES ON *.* TO 'aeuser'@'localhost';
```

6. Create the IBM schema:

   ```
   # /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/sql/mysql
       java -Xmx128m -Xms128m -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/*:/usr/share/
   java/* com.spss.ae.dbscript.ScriptRunner ../../configuration/config.properties schema.sql true
   ```

7. Set the file owner to omm:

   ```
   # chown -R omm:wheel /opt/ibm/*
   ```

8. Download the HDFS and Spark clients from the FusionInsight administrator user interface. For example, download the clients to the Analytic Server /tmp/FusionInsight-Client folder and extract the downloaded *.tar files.
9. Install the HDFS client after extracting its *.tar file.
10. On the Analytic Server, create a hadoop folder in /opt/ibm/spss/analyticserver/3.1 and copy the Hadoop *.jar files to the hadoop folder. The Hadoop *.jar files are located in the FusionInsight_V100R002C60U20_Spark_ClientConfig/Spark/FusionInsight-Spark-1.5.1/lib folder that was created when you extracted the *.tar files in step 8.
11. On the Analytic Server, create a zookeeper folder in /opt/ibm/spss/analyticserver/3.1 and copy the zookeeper-3.5.1.jar to the zookeeper folder.
12. Extract the contents of the cf.zip to the following folder: /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration.
13. Copy the native Hadoop files to the following folder: /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64

    The native Hadoop files are located in the extracted HDFS client *.tar contents from step 8.
14. Modify the following configuration files as described in corresponding code examples.
    - /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/config.properties

      ```
      spark.version=1.x
      http.port=9080
      https.port=9443
      ae.cluster.zookeeper.connect.string=172.16.155.123:24002,172.16.155.212:24002,172.16.186.208:24002
      ae.cluster.member.name=huawei-1
      ae.cluster.collective.name=Test_01
      jndi.aedb=jdbc/aeds
      jndi.aedb.url=jdbc:mysql://huawei-1/aedb?createDatabaseIfNotExist=true
      jndi.aedb.username=aeuser
      jndi.aedb.driver=com.mysql.jdbc.Driver
      distrib.fs.root=/user/as_user/analytic-root
      admin.username=admin
      enable.resume=true
      load.feature.flags.on.msg=true
      jndi.aedb.password=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAGk3bIuya2BzXYeXyFcOrxo=
      ae.kerberos.principal=omm/huawei-1@HADOOP.COM
      hdfs.user=omm/huawei-1@HADOOP.COM
      web.authentication.kerberos.principal=HTTP/huawei-1@HADOOP.COM
      java.security.krb5.conf=/home/omm/kerberos/var/krb5kdc/krb5.conf
      web.authentication.kerberos.keytab=/opt/ibm/spss/analyticserver/3.1.1/analyticserver.keytab
      hdfs.keytab=/opt/ibm/spss/analyticserver/3.1.1/analyticserver.keytab
      ae.db.connect.method=Basic
      kdcrealm=HADOOP.COM
      kdcserver=172.16.155.212:21732
      encryption.keystore.password=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAMDJul7PVsvdIyLlzjeS8ws=
      encryption.keystore.base64=zs7OzgAAAAIAAAABAAAAAwA6Y29tLnNwc3MuYWUuZW5jcnlwdGlvbi5wcm92aWRlci5lbmNyeXB0aW9ucHJvdmlkZXJpbXBs
      LmFlcwAAAUTg2AhyrO0ABXNyABlqYXZheC5jcnlwdG8uU2VhbGVkT2JqZWN0PjY9psO3VHACAARbAA1lbmNvZGVkUGFyYW1zdAACW0JbABBlbmNyeXB0ZWRDb25
      0ZW50cQB+AAFMAAlwYXJhbXNBbGd0ABJMamF2YS9sYW5nL1N0cmluZztMAAdzZWFsQWxncQB+AAJ4cHVyAAJbQqzzF/gGCFTgAgAAeHAAAAAPMA0ECEnr6ybTxO
      1mAgEUdXEAfgAEAAAAcGbNRpiJe0xkAuiMpWPjhzFuWCD2OeK7YZ4pwutRbgEcx4ul3SfPDAQcMZDTH+Ze03p8p1m7Kb/yY7SK6xvaaFYvCC9IWNgU6pkz/FXsw
      nVgb1G/Jsve7mYEX+8R2FUC+t2CEuzioKdTChUZsnzz0xB0AANQQkV0ABZQQkVXaXRoTUQ1QW5kVHJpcGxlREVTqmaA1K/MuEHB/yIaqSe9NgA2JsY=
      jdbc.drivers.location=/usr/share/jdbc
      default.security.provider=Websphere
      load.feature.flags.on.msg=true
      spark.serializer=org.apache.spark.serializer.JavaSerializer
      spark.executor.extraLibraryPath=/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
      ```

```
zookeeper.server.prinicipal=zookeeper/hadoop.hadoop.com@HADOOP.COM
zookeeper.server.keytab=/opt/huawei/Bigdata/FusionInsight_V100R002C60U20/FusionInsight-Zookeeper-3.5.1/zookeeper/conf/zookeeper.keytab
zookeeper.server.jaas.conf=/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/jaas.conf
krb5.conf=/home/omm/kerberos/var/krb5kdc/krb5.conf
```

- /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/jass.conf

```
Client {
com.sun.security.auth.module.Krb5LoginModule required
useKeyTab=true
keyTab="/opt/huawei/Bigdata/FusionInsight/FusionInsight-Zookeeper-3.5.1/zookeeper/conf/zookeeper.keytab"
storeKey=true
principal="zkcli/hadoop.hadoop.com@HADOOP.COM"
useTicketCache=false
debug=true;
};
```

15. Create a hadoop-conf folder in the following directory: /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration

    Copy the Spark *.xml files to the hadoop-conf folder. The Spark *.xml files are located in the extracted Spark client *.tar contents from step 8.

16. Modify the /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/server.env file as follows (you must modify the path value to match your actual server path):

```
JAVA_HOME=/opt/huawei/Bigdata/jdk/jre
PATH=/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/Bigdata/jdk/jre/
lib/amd64:/usr/sbin:/usr/bin:/sbin:/bin
IBM_SPSS_AS_NATIVE_PATH=/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
LD_LIBRARY_PATH=/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/
Bigdata/jdk/jre/lib/amd64=/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/native
SPARK_HOME=/opt/ibm/spss/analyticserver/3.1/spark-client
HADOOP_HOME="/opt/client_hdfs/HDFS/hadoop"
```

17. Modify the /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/server.xml file as follows:

```
<server description="new server">
<!-- Enable features -->
  <featureManager>
    <feature>servlet-3.1.1</feature>
    <feature>jsp-2.3</feature>
    <feature>jdbc-4.0</feature>
    <feature>jndi-1.0</feature>
    <feature>localConnector-1.0</feature>
    <feature>jaxrs-2.0</feature>
    <feature>json-1.0</feature>
    <feature>appSecurity-2.0</feature>
    <feature>ldapRegistry-3.0</feature>
    <feature>restConnector-1.0</feature>
    <feature>monitor-1.0</feature>
    <feature>ssl-1.0</feature>
  </featureManager>
  <applicationManager startTimeout="120s" />
  <executor name="LargeThreadPool" id="default" coreThreads="100" keepAlive="60s" stealPolicy="STRICT" rejectedWorkPolicy="CALLER_RUNS" />
  <webContainer deferServletLoad="false" disallowAllFileServing="false" fileServingEnabled="true" trusted="false" directoryBrowsingEnabled=
"false" asyncTimeoutDefault="300000"/>
  <classloading useJarUrls="true"/>
  <applicationMonitor updateTrigger="mbean" />
  <mimeTypes>
    <type>svg=image/svg+xml</type>
  </mimeTypes>
  <variable name="AE_DATABASE" value="${wlp.install.dir}/usr/servers/aeserver/aedb" />
  <administrator-role>
    <user>admin</user>
  </administrator-role>
  <include optional="true" location="${server.config.dir}/private_library.xml"/>
  <include optional="true" location="${server.config.dir}/http_endpoint.xml"/>
  <include optional="true" location="${server.config.dir}/security_cfg.xml"/>
  <include optional="true" location="${server.config.dir}/ssl_cfg.xml"/>
  <include optional="true" location="${server.config.dir}/configuration/key.xml" />
</server>
```

18. Create a private_library.xml file in the /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/ folder. The file contents should resemble the following:

```
<server>
  <application context-root="/analyticserver" id="AS_BOOT" location="AE_BOOT.war" name="AS_BOOT" type="war">
    <classloader>
      <privateLibrary>
        <fileset dir="${wlp.install.dir}/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib" includes="*.jar"/>
        <fileset dir="/usr/share/java" includes="*.jar"/>
        <fileset dir="${wlp.install.dir}/../lib" includes="*.jar"/>
        <fileset dir="${wlp.install.dir}/../spark-client/lib" includes="spark-assembly-*.jar"/>
        <folder  dir="${wlp.install.dir}/usr/servers/aeserver/configuration/hadoop-conf"/>
        <fileset dir="${wlp.install.dir}/../jdbc" includes="postgresql-*.jar"/>
        <fileset dir="${wlp.install.dir}/../jdbc" includes="*.jar"/>
        <fileset dir="${wlp.install.dir}/../hive" includes="*.jar"/>
        <fileset dir="${wlp.install.dir}/../zookeeper" includes="*.jar"/>
        <fileset dir="${wlp.install.dir}/../hadoop" includes="*.jar"/>
      </privateLibrary>
    </classloader>
  </application>
  <application id="help" location="help.war" name="help" type="war" context-root="/analyticserver/help"/>
</server>
```

19. Create a `security_cfg.xml` file in the `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/` folder. The file contents should resemble the following:

```
<server>
  <basicRegistry id="basic" realm="ibm">
    <user name="admin" password="admin"/>
  </basicRegistry>
</server>
```

20. Create a `http_endpoint.xml` file in the `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/` folder. The file contents should resemble the following:

```
<server>
  <httpEndpoint host="*" id="defaultHttpEndpoint" httpPort="9080" httpsPort="9443" onError="FAIL"/>
</server>
```

21. Modify the `jvm_option` file as follows:

```
-Xms512M
-Xmx2048M
-Dclient.encoding.override=UTF-8
-XX:+UseParNewGC
-Dconfig.folder.path=/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration

# -server
# posible values for config.profile are "local" or "hadoop". The values must be specified without quotes
# Temporary disable profiles due to installer changes
#-Dconfig.profile=hadoop
```

22. Run the `hdfsUpdate.sh` script (`/opt/ibm/spss/analyticserver/3.1/bin/hdfsUpdate.sh`).
23. Run the `start.sh` script (`/opt/ibm/spss/analyticserver/3.1/bin/start.sh`) to start the Analytic Server service; run the `stop.sh` script (`/opt/ibm/spss/analyticserver/3.1/bin/stop.sh`) to stop the Analytic Server service.
24. You can access the Analytic Server console via the following URL: `http://<servername>:9080/analyticserver/admin/ibm`

Configuring MapReduce v2 jobs

Feature flags provide the capability of enabling and disabling specific application features.

Use the **`preferred.mapreduce`** setting in the `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` file to control how MapReduce jobs are handled:

Table 19. preferred.mapreduce settings

Property	**Description**
`preferred.mapreduce`	Controls the method in which MapReduce jobs are run. Valid values include: • `spark` • `m3r` • `hadoop` For example: `preferred.mapreduce=spark`

Chapter 6. Configuring IBM SPSS Modeler for use with IBM SPSS Analytic Server

In order to enable SPSS Modeler for use with Analytic Server, you need to make some updates to the SPSS Modeler Server installation.

1. Configure SPSS Modeler Server to associate it with an Analytic Server installation.
   a. Edit the `options.cfg` file in the `config` subdirectory of the main server installation directory, and add or edit the following lines:

      ```
      as_ssl_enabled, {Y|N}
      as_host, "{AS_SERVER}"
      as_port, PORT
      as_context_root, "{CONTEXT-ROOT}"
      as_tenant, "{TENANT}"
      as_prompt_for_password, {Y|N}
      as_kerberos_auth_mode, {Y|N}
      as_kerberos_krb5_conf, {CONF-PATH}
      as_kerberos_krb5_spn, {AS-SPN}
      ```

      as_ssl_enabled
      Specify Y if secure communication is configured on Analytic Server; otherwise, N.

      as_host
      The IP address/host name of the server that hosts Analytic Server.

      Note: You must provide an appropriate IP address/host domain name when SSL is enabled for Analytic Server.

      as_port
      The port on which Analytic Server is listening (by default this is 8080).

      as_context_root
      The Analytic Server context root (by default this is analyticserver).

      as_tenant
      The tenant the SPSS Modeler Server installation is a member of (the default tenant is ibm).

      as_prompt_for_password
      Specify N if the SPSS Modeler Server is configured with the same authentication system for users and passwords as that used on Analytic Server; for example, when using Kerberos authentication. Otherwise, specify Y.

      When running SPSS Modeler in batch mode, you add `-analytic_server_username {ASusername} -analytic_server_password {ASpassword}` as arguments to the `clemb` command.

      as_kerberos_auth_mode
      Specify Y to enable Kerberos SSO from SPSS Modeler.

      as_kerberos_krb5_conf
      Specify the path to the Kerberos configuration file that Analytic Server should use; for example, `\etc\krb5.conf`.

      as_kerberos_krb5_spn
      Specify the Analytic Server Kerberos SPN; for example, HTTP/ashost.mydomain.com@MYDOMAIN.COM.

   b. Restart the SPSS Modeler Server service.

   In order to connect to an Analytic Server installation that has SSL/TLS enabled, there are some further steps to configuring your SPSS Modeler Server and client installations.

a. Navigate to `http{s}://{HOST}:{PORT}/{CONTEXT-ROOT}/admin/{TENANT}` and log on to the Analytic Server console.

b. Download the certification file from the browser and save it to your file system.

c. Add the certification file to the JRE of both your SPSS Modeler Server and SPSS Modeler Client installations. The location to update can be found under the `/jre/lib/security/cacerts` subdirectory of the SPSS Modeler installation path.

   1) Make sure the `cacerts` file is not read-only.

   2) Use the keytool program Modeler ships with – this can be found in the `/jre/bin/keytool` subdirectory of the SPSS Modeler installation path.

      Run the following command

      ```
      keytool -import -alias <as-alias> -file <cert-file> -keystore "<cacerts-file>"
      ```

      Note that <as-alias> is an alias for the `cacerts` file. You can use any name you like as long as it is unique to the `cacerts` file.

      So an example command would look like the following.

      ```
      keytool -import -alias MySSLCertAlias -file C:\Download\as.cer
              -keystore "c:\Program Files\IBM\SPSS\Modeler\{ModelerVersion}\jre\lib\security\cacerts"
      ```

d. Restart your SPSS Modeler Server and SPSS Modeler Client .

2. [optional] Install IBM SPSS Modeler - Essentials for R , if you plan to score R models in streams with Analytic Server data sources. IBM SPSS Modeler - Essentials for R is available for download (https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp).

Chapter 7. Using SLM tags to track licensing

SLM tags are based on the ISO/IEC 19770-4 standard draft for Resource Utilization Measurement. SLM tags provide a standardized capability for a product to report its consumption of license metrics (resources related to the use of a software asset). After enabling SLM in a product, an runtime XML file is generated to self-report its license use.

When Analytic Server is started, `slmtag` files are created in the `<as_installation_path>/logs/slmtag` folder.

Because there are two license types, two different metrics are periodically recorded:

- For the current Analytic Server version, licensing is based on the total number of data nodes in the Hadoop cluster (based on Virtual Server). The number of nodes is recorded in the following `slmtag` file section.

  ```
  <Type>VIRTUAL_SERVER</Type>
     <SubType>Number of Data Nodes in Hadoop</SubType>
     <Value>2</Value>
     ...
  ```

- For Analytic Server versions prior to 3.1, licensing was based on the HDFS storage size in the Hadoop cluster (based on RVU). For example, the storage size (in tegabytes) is recorded in the following `slmtag` file section.

  ```
  <Type>RESOURCE_VALUE_UNIT</Type>
     <SubType>HDFS storage (Unit: Tega byte)</SubType>
     <Value>0.21</Value>
  ```

The SLM tag output is started in a thread and it is affected by the properties that are defined in the `SlmTagOutput.properties` file. The file is located in the`<as_installation_path>/configuration` folder.

Table 20. SLM Tag Properties

Properties	**Description**
license.metric.logger.output.enabled	Controls the SLM log file generation. The default value is `False`.
license.metric.logger.output.dir	The relative path to the directory that stores the SLM tag files. The default directory is `<as_installation_path>/ logs`.
license.metric.logger.output.SLMLogFrequency	The time interval (unit:milliseconds) for collecting SLM logs.
icense.metric.logger.file.size	The maximum SML tag file size, in bytes.
license.metric.logger.file.number	The maximum number of SLM tag files for one software identity instance.

Chapter 8. Troubleshooting

This section describes some common installation and configuration issues and how you can fix them.

General issues

Installation succeeds with warnings, but users are unable to create data sources with error "Unable to complete the request. Reason: Permission denied"

Setting the **`distrib.fs.root`** parameter to a directory that the Analytic Server user (by default, as_user) doesn't have access to will result in errors. Make certain that the Analytic Server user is authorized to read, write, and execute the **`distrib.fs.root`** directory.

Analytic Server performance is progressively getting worse.

When the Analytic Server performance does not meet expectations, remove all of the `*.war` files from the Knox service deployment path: `/<KnoxServicePath>/data/ deployments`. For example: `/usr/iop/4.1.0.0/knox/data/deployments`.

Uninstalling Analytic Server or Essentials for R on Ambari

In some cases, the uninstall process hangs when uninstalling Analytic Server or Essentials for R on Ambari. When the issue occurs, you must manually stop the Ambari server's process ID.

Issues when Analytic Server is installed on POWER System that uses OpenJDK

When Analytic Server is running on a POWER System that uses OpenJDK, you must manually perform the following configuration steps to ensure that the coordinate system API works as expected

Note: You can disregard the configuration requirement if you do not use the coordinate system API.

1. In the Ambari console, navigate to **Analytic Server service** > **Configs tab** > **Advanced analytics-jvm-options** and add the following line to the content area:
   ```
   -XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCSHorizon.*
   ```
2. In the Ambari console, navigate to the **Custom analytics.cfg** section and add the following 3 configurations:

   spark.executor.extraJavaOptions

   Set the value to: `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCSHorizon.*`

   spark.driver.extraJavaOptions

   Set the value to: `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCSHorizon.*`

   mapred.child.java.opts

   Set the value to: `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCSHorizon.*`

Error when installing Analytic Server on SuSE Linux 12

You may encounter the following error when installing Analytic Server on SuSE Linux 12:

```
Signature verification failed [4-Signatures public key is not available]
```

The issue can be resolved by performing the following tasks prior to installing Analytic Server on SuSE Linux 12:

1. Download a public key to your host from the following URL:
   ```
   https://ibm-open-platform.ibm.com/repos/IBM-SPSS-AnalyticServer/3.1.1.1/IBM-SPSS-ANALYTICSERVER-GPG-KEY.public
   ```
2. Import the public key by running the following command on your host:
   ```
   rpm --import IBM-SPSS-ANALYTICSERVER-GPG-KEY.public
   ```

Issues with specific Hadoop distributions

Refresh action for Analytic Server service is disabled on Hortonworks 2.3-2.6

To manually refresh Analytic Server libraries on Hortonworks 2.3-2.6 use the following steps.

1. Log on to the host running the Analytic Metastore as the Analytic Server user (by default as_user).

   Note: You can find this host name from the Ambari console.

2. Run the **`refresh`** script in the directory `{AS_ROOT}/bin`; for example:

   ```
   cd /opt/ibm/spss/analyticserver/3.1/bin
   ./refresh
   ```

3. Restart the Analytic Server service in the Ambari console.

Packages that are downloaded from an external site fail the hash check in Cloudera Manager

The hash verification error displays in the parcels list. The problem can be resolved by allowing the download process to finish and then restart Cloudera via the `cloudera-scm-server` service. The error does not occur after the service restarts.

HDFS supergroup properties

Analytic Server will log an exception during start-up if the `as_user` is not a member of the following HDFS group properties: **`dfs.permissions.supergroup/dfs.permissions.superusergroup`**. For example:

```
[11/15/17 7:32:35:510 PST] 000000bf SystemOut
O 2017-11-15 07:32:35,510 | : |  |  |  |  | ERROR | slmtagoutput.SlmOuputAgent | SLM Logger => Error in performing callback function when calculating number
of nodes in kerberos environment: org.apache.hadoop.ipc.RemoteException(org.apache.hadoop.security.AccessControlException): Access denied for user as_user.
Superuser privilege is required
at org.apache.hadoop.hdfs.server.namenode.FSPermissionChecker.checkSuperuserPrivilege(FSPermissionChecker.java:93)
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.checkSuperuserPrivilege(FSNamesystem.java:6606)
at org.apache.hadoop.hdfs.server.namenode.FSNamesystem.datanodeReport(FSNamesystem.java:5595)
at org.apache.hadoop.hdfs.server.namenode.NameNodeRpcServer.getDatanodeReport(NameNodeRpcServer.java:928)
at org.apache.hadoop.hdfs.server.namenode.AuthorizationProviderProxyClientProtocol.getDatanodeReport(AuthorizationProviderProxyClientProtocol.java:390)
at org.apache.hadoop.hdfs.protocolPB.ClientNamenodeProtocolServerSideTranslatorPB.getDatanodeReport(ClientNamenodeProtocolServerSideTranslatorPB.java:694)
at org.apache.hadoop.hdfs.protocol.proto.ClientNamenodeProtocolProtos$ClientNamenodeProtocol$2.callBlockingMethod(ClientNamenodeProtocolProtos.java)
at org.apache.hadoop.ipc.ProtobufRpcEngine$Server$ProtoBufRpcInvoker.call(ProtobufRpcEngine.java:617)
at org.apache.hadoop.ipc.RPC$Server.call(RPC.java:1073)
at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2141)
at org.apache.hadoop.ipc.Server$Handler$1.run(Server.java:2137)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:415)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1912)
at org.apache.hadoop.ipc.Server$Handler.run(Server.java:2135)
```

You must manually add `as_user` to the OS group that is defined in the `hdfs-site` configuration properties: **`dfs.permissions.supergroup/dfs.permissions.superusergroup`**.

- For Cloudera, the default property value is **`supergroup`** and must be changed to an OS group that actually exists. For information on the supergroup setting in Cloudera, refer to the Cloudera documentation.
- For Ambari, the default property value is **`hdfs`**. By default, during an Ambari installation Analytic Server adds `as_user` to the HDFS and Hadoop groups.

On Linux use the **`usermod`** command to add `as_user` to the HDFS **`superusergroup`** (if it does not already exist).

For general information regarding HDFS permissions, see the HDFS Permissions Guide.

Issues with the metadata repository

Operation CREATE USER fails when running the add_mysql_user script

Before running the **`add_mysql_user`** script, you need to first manually remove the user that you are attempting to add from the mysql database. You can remove the users via the MySQL Workbench UI or via MySQL commands. For example:

```
mysql -u root -e "DROP USER '$AEDB_USERNAME_VALUE'@'localhost';"
mysql -u root -e "DROP USER '$AEDB_USERNAME_VALUE'@'$METASTORE_HOST';"
mysql -u root -e "DROP USER '$AEDB_USERNAME_VALUE'@'%';"
```

In the above commands, replace $AEDB_USERNAME_VALUE with the username you want removed, and replace $METASTORE_HOST with the host name the database is installed on.

Issues with Apache Spark

Issues with streams that are run within a Spark process

SPSS Modeler streams fail to complete when forced to run within a Spark process. The SPSS Modeler streams that fail are built with an Analytic Server source node (HDFS file), that is linked to a `Sort` node, and then set to export to another Analytic Server data source. After the stream is run, the Resource Manager user interface indicates that the new application is running, but the stream never completes and remains in a `Running` state. There are no messages that indicate why the stream fails to complete in the Analytic Server logs, YARN logs, or Spark logs.

The issue can be resolved by adding the `spark.executor.memory` setting to the custom `analytics.cfg` file in the Analytic Server configuration. Setting the memory value to 4GB allows the previously failed SPSS Modeler streams to complete in less than 2 minutes (in a single node cluster environment).

Failure to run Spark jobs with Cloudera 5.x and Spark 1.x

You may encounter the following exception when using Cloudera 5.x with Spark 1.x:

```
org.apache.spark.SparkException: Exception when registering SparkListener
```

The exception is caused by `java.lang.ClassCastException: com.cloudera.spark.lineage.ClouderaNavigatorListener` not being able to cast `org.apache.spark.scheduler.SparkListener`.

To avoid the exception, you must add the following line in the **Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area.

```
spark.extraListeners=org.apache.spark.JavaSparkListener
```

High availability clusters

Analytic Server cannot be added to more hosts due to changes in dependencies

Run the `update_clientdeps` script using the instructions in "Updating client dependencies" on page 25.

"The Analytic Cluster Service has unexpectedly lost contact with Zookeeper, this JVM is being terminated to maintain cluster integrity."

One thing that may cause this is if the amount of data being written to Zookeeper is too large. If, in the Zookeeper logs are exceptions like:

```
java.io.IOException: Unreasonable length = 2054758
```

or in the Analytic Server logs are messages like:

```
Caused by: java.io.UTFDataFormatException: encoded string too long: 2054758 bytes
  at java.io.DataOutputStream.writeUTF(DataOutputStream.java:375)
```

1. In the Ambari console, navigate to the Zookeeper service Configs tab and add the following line to the env-template, then restart the Zookeeper service.

   ```
   export JVMFLAGS="-Xmx2048m -Djute.maxbuffer=2097152"
   ```

2. In the Ambari console, navigate to the Analytic Server service Configs tab and add the following in the Advanced analytics-jvm-options, then restart the Analytic Cluster service.

   ```
   -Djute.maxbuffer=2097152
   ```

The number to specify for the jute.maxbuffer setting should be higher than the number indicated in the exception messages.

Zookeeper transaction data becomes unmanageable

Set the **autopurge.purgeInterval** parameter in zoo.cfg to 1 to enable automatic purges of the Zookeeper transaction log.

Analytic cluster service loses contact with Zookeeper

Review and modify the **tickTime**, **initLimit**, and **syncLimit** parameters in zoo.cfg. For example:

```
# The number of milliseconds of each tick
tickTime=2000
# The number of ticks that the initial
# synchronization phase can take
initLimit=30
# The number of ticks that can pass between
# sending a request and getting an acknowledgement
syncLimit=15
```

See the Zookeeper documentation for details: https://zookeeper.apache.org/doc/r3.3.3/zookeeperAdmin.html

Analytic Server jobs do not resume

There is a common situation in which Analytic Server jobs do not resume.

- When an Analytic Server job fails because a cluster member fails, the job is normally restarted automatically on another cluster member. If the job does not resume, check to ensure there are at least 4 cluster members in the High Availability cluster.

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBMproducts. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Printed in USA