

IBM SPSS Analytic Server  
Version 3.1.0

*Installation and Configuration Guide*



**Note**

Before using this information and the product it supports, read the information in "Notices" on page 67.

**Product Information**

This edition applies to version 3, release 1, modification 0 of IBM SPSS Analytic Server and to all subsequent releases and modifications until otherwise indicated in new editions.

---

# Contents

<b>Chapter 1. Prerequisites . . . . .</b>	<b>1</b>	Migration . . . . .	41
<b>Chapter 2. Ambari Installation and Configuration . . . . .</b>	<b>3</b>	Uninstalling Analytic Server on Cloudera . . . . .	42
Ambari-specific prerequisites . . . . .	3	<b>Chapter 4. MapR Installation and Configuration . . . . .</b>	<b>43</b>
IBM SPSS Analytic Server installation precheck and postcheck tools . . . . .	3	MapR Overview . . . . .	43
Installation on Ambari . . . . .	5	Installing Analytic Server on MapR . . . . .	43
Offline installation . . . . .	7	Configuring MapR . . . . .	46
Installing Analytic Server against an externally managed MySQL environment . . . . .	10	Enabling database pushback. . . . .	46
Configuration. . . . .	11	Enabling Apache Hive. . . . .	47
Security. . . . .	11	Enabling Apache HBase . . . . .	48
Enabling Support for Essentials for R. . . . .	16	Enabling Apache Spark . . . . .	49
Enabling relational database sources . . . . .	18	Enabling feature flags . . . . .	50
Enabling HCatalog data sources . . . . .	19	Enabling R . . . . .	51
Changing ports used by Analytic Server. . . . .	20	Enabling LZO . . . . .	51
High availability Analytic Server . . . . .	21	Enabling SLM tags for MapR . . . . .	52
Optimizing JVM options for small data . . . . .	21	Setting up an IBM SPSS Analytic Server cluster for MapR . . . . .	52
Updating client dependencies . . . . .	21	Uninstalling MapR . . . . .	52
Configuring Apache Knox . . . . .	21	Migrating IBM SPSS Analytic Server on MapR . . . . .	53
Migrating IBM SPSS Analytic Server on Ambari . . . . .	25	MapR troubleshooting. . . . .	54
Uninstalling . . . . .	26	<b>Chapter 5. Huawei FusionInsight HD Installation and Configuration. . . . .</b>	<b>55</b>
Uninstalling Essentials for R. . . . .	26	FusionInsight HD overview . . . . .	55
<b>Chapter 3. Cloudera Installation and Configuration . . . . .</b>	<b>29</b>	Installation on Huawei FusionInsight HD . . . . .	55
Cloudera overview . . . . .	29	<b>Chapter 6. Configuring IBM SPSS Modeler for use with IBM SPSS Analytic Server . . . . .</b>	<b>59</b>
Cloudera-specific prerequisites . . . . .	29	<b>Chapter 7. Using SLM tags to track licensing . . . . .</b>	<b>61</b>
Configuring MySQL for Analytic Server. . . . .	29	<b>Chapter 8. Troubleshooting . . . . .</b>	<b>63</b>
Installation on Cloudera . . . . .	30	<b>Notices . . . . .</b>	<b>67</b>
Configuring Cloudera . . . . .	32	Trademarks . . . . .	69
Security . . . . .	32		
Enabling support for Essentials for R. . . . .	36		
Enabling relational database sources . . . . .	37		
Enabling HCatalog data sources . . . . .	38		
Configuring Apache Impala . . . . .	39		
Changing ports used by Analytic Server. . . . .	40		
High availability Analytic Server . . . . .	40		
Optimizing JVM options for small data . . . . .	40		



---

# 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**.

## 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>

## 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.

## 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 daemon to 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](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.

---

## 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 run after the Analytic Server installation.

### 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, cmx)

**Note:** cmx is supported only in BigInsights.

- Data with PySpark

- Data that uses native SPSS components (including alm, tree, neuralnet, scoring, tascoreing)
- Data with MapReduce
- Data with in-memory MapReduce

## Tool location and prerequisites

The precheck and postcheck tools are located in the following directories:

- **BigInsights**

`/var/lib/ambari-server/resources/stacks/BigInsights/4.X/services/ANALYTICSERVER/package/chktool`

- **HDP**

`/var/lib/ambari-server/resources/stacks/HDP/2.X/services/ANALYTICSERVER/package/chktool`

- **Cloudera**

The tools are packaged in the `AnalyticServer-*.jar` file (the file is located at: `/opt/cloudera/csd`).

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 Ambari 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 7, 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 Ambari -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. More detailed steps follow.

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

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. The available Ambari binaries are:

Table 1. Analytic Server self-extracting binary files

Description	Binary filename
IBM® SPSS® Analytic Server 3.1 for BigInsights 4.1 and 4.2.5 Linux on System p LE English	spss_as-3.1-bi4.1-4.2.5-lppc64.bin
IBM SPSS Analytic Server 3.1 for BigInsights 4.1, 4.2, and 4.2.5 Linux x86-64 English	spss_as-3.1-bi4.1-4.2-4.2.5-lx86.bin
IBM SPSS Analytic Server 3.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	spss_as-3.1-hdp2.4-2.6-lx86.bin
IBM SPSS Analytic Server 3.1 for Hortonworks Data Platform 2.6 Linux on System p LE English	spss_as-3.1-hdp2.6-lppc64.bin

2. 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>.

[GPFS (Spectrum Scale) only] Download the file <https://ibm-open-platform.ibm.com/repos/IBM-SPSS-AnalyticServer/3.1.0.0/rpms/IBM-SPSS-AnalyticServer-3.1.0.0.repo> (x86 and ppc64le) or <https://ibm-open-platform.ibm.com/repos/IBM-SPSS-AnalyticServer/3.1.0.0/Ubuntu/IBM-SPSS-AnalyticServer.list> (Ubuntu) and move it to the folder /etc/yum.repos.d (RHEL, CentOS), /etc/zypp/repos.d (SLES), or /etc/apt/sources.list.d (Ubuntu) on every node on which you add the Analytic Server Metastore as a service.

#### Offline installation

Choose offline if your Ambari server host does not have internet access. For details see “Offline installation” on page 7.

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. If you are performing an offline installation, verify that the `repoinfo.xml` and `IBM-SPSS-AnalyticServer-3.1.0.0.repo` changes are still present. Contact your IBM support representative if you find that the `repoinfo.xml` and `IBM-SPSS-AnalyticServer-3.1.0.0.repo` changes are not persisting.
6. 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**.

### 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`

7. You now have a functioning instance of Analytic Server. Further configuration is optional. For more information on configuring and administering Analytic Server, see the topic: “Configuration” on page 11. For information on migrating an existing configuration to a new installation, see the topic: “Migrating IBM SPSS Analytic Server on Ambari” on page 25.

- 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

The general workflow for an offline installation is as follows:

- 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. The available Ambari binaries are:

Table 2. Analytic Server self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1 for BigInsights 4.1 and 4.2.5 Linux on System p LE English	spss_as-3.1-bi4.1-4.2.5-lppc64.bin
IBM SPSS Analytic Server 3.1 for BigInsights 4.1, 4.2, and 4.2.5 Linux x86-64 English	spss_as-3.1-bi4.1-4.2-4.2.5-lx86.bin
IBM SPSS Analytic Server 3.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English	spss_as-3.1-hdp2.4-2.6-lx86.bin
IBM SPSS Analytic Server 3.1 for Hortonworks Data Platform 2.6 Linux on System p LE English	spss_as-3.1-hdp2.6-lppc64.bin

- Run the executable binary and specify an offline installation. An offline installation downloads the necessary RPM or DEB files, and should be run on a computer that can access `https://ibm-open-platform.ibm.com`. The executable binary file is located in the available Ambari `<AS_INSTALLABLE_HOME>` distribution directories.
- Copy the entire contents of the appropriate `<AS_INSTALLABLE_HOME>` directory from the machine with internet access to the Ambari manager node (behind the firewall).
- Continue with the instruction steps for your particular distribution (BigInsights/HDP or Ubuntu) on the appropriate Ambari manager nodes (behind the firewall).

## BigInsights and HDP instructions - 3.1.0

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

- Use the following command to check if the Ambari server is running:

```
ambari-server status
```

Shut down the Ambari server node (if it is currently running):

```
ambari-server stop
```

- Install the tool that creates a local yum repository.

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

or

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

- 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
```

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

**BigInsights 4.1, 4.2, and 4.2.5 (x86\_64)**

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm
```

IBM-SPSS-AnalyticServer-3.1.0.0-1.x86\_64.rpm

### BigInsights 4.1 and 4.2.5 (PPC64LE)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.0.0-1.ppc64le.rpm

### HDP 2.4, 2.5, and 2.6 (x86\_64)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.0.0-1.x86\_64.rpm

### HDP 2.6 (PPC64LE)

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.0.0-1.ppc64le.rpm

5. Create the local repository definition. For example, create a file that is named IBM-SPSS-AnalyticServer-3.1.0.0.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
```

6. Create the local yum repository.  
createrepo /home/root/repos/IBM-SPSS-AnalyticServer/x86\_64 (RHEL, CentOS, SLES)
7. 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:** Steps 8 and 9 apply only if you use an externally managed MySQL environment.

8. 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 /opt/AS\_Installable/IBM-SPSS-AnalyticServer to the node/host where the MySQL instance, that will be used as the AS\_MetaStore, is installed. For example, /opt/AS\_InstallTools.
    - 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

9. 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>
```

```

    <repo>IBM-SPSS-AnalyticServer</repo>
    <reponame>IBM-SPSS-AnalyticServer-3.1.0.0</reponame>
  </repo>
</os>

```

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

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

10. 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.

**BigInsights 4.1, 4.2, and 4.2.5 (x86\_64)**

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.0.0-1.x86_64.rpm
```

**BigInsights 4.1 and 4.2.5 (PPC64LE)**

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.0.0-1.ppc64le.rpm
```

**HDP 2.4, 2.5, and 2.6 (x86\_64)**

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.0.0-1.x86_64.rpm
```

**HDP 2.6 (PPC64LE)**

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.0.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.0.0-1.ppc64le.rpm
```

e. Create the local repository definition. For example, create a file that is named IBM-SPSS-AnalyticServer-3.1.0.0.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)
```

11. Continue with step 5 in the “Installation on Ambari” on page 5 topic.

## Ubuntu instructions - 3.1.0

1. Navigate to the [IBM Passport Advantage® Web Site](#) and download the appropriate Ubuntu self-extracting binary file specific to your stack, stack version, and hardware architecture to a host within the Ambari cluster. The available binaries are:

Description	Binary filename
Table 3. Analytic Server self-extracting binary files	spss_as-3.1-hdp2.4-2.6-1x86.bin

IBM SPSS Analytic Server 3.1 for Hortonworks Data Platform 2.4, 2.5, and 2.6 Linux x86-64 English

2. On a machine with internet access, run the executable binary and specify an offline installation. An offline installation downloads the necessary DEB files, and should be run on a computer that can access <https://ibm-open-platform.ibm.com>. The executable binary file is in the available Ambari <AS\_INSTALLABLE\_HOME> distribution directory.
3. The required Analytic Server DEB files are located in the following directory.  
IBM-SPSS-AnalyticServer/packages
4. Use the following commands to install Analytic Server 3.1.0:  

```
dpkg -i ./IBM-SPSS-AnalyticServer-ambari-2.x_3.1.0.0_amd64.deb
dpkg -i ./IBM-SPSS-AnalyticServer_1_amd64.deb
```
5. Restart your Ambari server.  

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

## 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 `/opt/AS_Installable/IBM-SPSS-AnalyticServer` to the node/host where the MySQL instance, that will be used as the `AS_MetaStore`, is installed. For example, `/opt/AS_InstallTools`.
  - 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.
  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/{version}.
- {AS\_SERVER\_ROOT} refers to the location of the configuration, log, and server files; for example, /opt/IBM/SPSS/AnalyticServer/{version}/ae\_wlpserver/usr/servers/aeserver.
- {AS\_HOME} refers to the location on HDFS that is used by Analytic Server as a root folder.

## Security

The **security.config** parameter defines the registry of users and groups that can be added as principals to the Analytic Server system.

By default, a basic registry is defined with a single user, admin, with password admin. You can change the registry by editing **security.config** or configuring Kerberos. The **security.config** parameter can be found in the **Advanced analytics.cfg** section of the Configs tab of the Analytic Server service.

**Note:** If you edit the **security.config** parameter to alter the registry, you then need to add any new users as principals to the Analytic Server system. See the *IBM SPSS Analytic Server Administrator's Guide* for details on tenant management.

### Making changes to the basic registry

The basic registry allows you to define a database of users and groups within the **security.config** parameter.

The default basic registry looks like the following.

```
<basicRegistry id="basic" realm="ibm">
  <user name="admin" password="admin"/>
</basicRegistry>
```

An example of an altered basic registry follows.

```
<basicRegistry id="basic" realm="ibm">
  <user name="user1" password="{xor}Dz4sLG5tbGs="/>
  <user name="user2" password="Pass"/>
  <user name="user3" password="Pass"/>
  <user name="user4" password="Pass"/>
  <user name="admin" password="{xor}KzosKw="/>
  <group name="Development">
    <member name="user1"/>
    <member name="user2"/>
  </group>
  <group name="QA">
    <member name="user3"/>
    <member name="user4"/>
  </group>
  <group name="ADMIN">
    <member name="user1"/>
    <member name="admin"/>
  </group>
</basicRegistry>
```

Passwords can be encoded to obfuscate their values with the securityUtility tool, which is in {AS\_ROOT}/ae\_wlpserver/bin.

```
securityUtility encode changeit
  {xor}PDC+MTg6Nis=
```

**Note:** See [http://www-01.ibm.com/support/knowledgecenter/SSD28V\\_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp\\_command\\_securityutil.html](http://www-01.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp_command_securityutil.html) for details of the securityUtility tool.

**Note:** The basic registry is useful in a sandbox environment, but is not recommended for a production environment.

## Configure 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="( & (uid=%v) (objectClass=inetOrgPerson)) "
    groupFilter="( & (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="( & (sAMAccountName=%v) (objectcategory=user)) "
    groupFilter="( & (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="ldapserverserver.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">
  <activatedFilters
    userFilter="( & (sAMAccountName=%v) (objectcategory=user)) "
    groupFilter="( & (cn=%v) (objectcategory=group)) "
    userIdMap="user:sAMAccountName"
```



```

        groupIdMap="*:cn"
        groupMemberIdMap="memberOf:member" >
    </activatedFilters>
</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.

### 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:** 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.

**Note:** If the Analytic Server installation uses a basic registry, it must include the Kerberos user accounts, using "-" as the password. An example follows.

```
<basicRegistry id="basic" realm="ibm">
  <user name="admin" password="-"/>
  <user name="user1" password="-"/>
  <user name="user2" password="-"/>
  <group name="group1">
    <member name="admin"/>
    <member name="user1"/>
    <member name="user2"/>
  </group>
  <group name="group2">
    <member name="admin"/>
    <member name="user1"/>
  </group>
</basicRegistry>
```

2. 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.
3. 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.
4. [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 Ambari 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).

## 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/analyticsserver/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/analyticsserver/3.1/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.

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 libx-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 ZYPPEP 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 xzvf 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-1.0.21.tar.gz --no-check-certificate
tar xzvf openssl-1.0.21.tar.gz
cd openssl-1.0.21/
./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>.

[GPFs (Spectrum Scale) only] Download the file [https://ibm-open-platform.ibm.com/repos/IBM-SPSS-ModelerEssentialsR/3.1.0.0/x86\\_64/IBM-SPSS-AnalyticServer-3.1.0.0.repo](https://ibm-open-platform.ibm.com/repos/IBM-SPSS-ModelerEssentialsR/3.1.0.0/x86_64/IBM-SPSS-AnalyticServer-3.1.0.0.repo) (x86), <https://ibm-open-platform.ibm.com/repos/IBM-SPSS-ModelerEssentialsR/3.1.0.0/ppc64le/IBM-SPSS-AnalyticServer-3.1.0.0.repo> (ppc64le), or <https://ibm-open-platform.ibm.com/repos/IBM-SPSS-ModelerEssentialsR/3.1.0.0/Ubuntu/IBM-SPSS-ModelerEssentialsR-3.1.0.0.list>

(Ubuntu) and move it to the folder `/etc/yum.repos.d` (RHEL, CentOS), `/etc/zypp/repos.d` (SLES), or `/etc/apt/sources.list.d` (Ubuntu) on every node on which you add the Analytic Server Metastore as a service.

### 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.

#### BigInsights 4.1, 4.2, and 4.2.5 (x86\_64)

`IBM-SPSS-ModelerEssentialsR-ambari-2.1-BI-4.1-8.4.0.0-1.x86_64.rpm`

#### BigInsights 4.1 and 4.2.5 (PPC64LE)

`IBM-SPSS-ModelerEssentialsR-ambari-2.1-BI-4.1-8.4.0.0-1.ppc64le.rpm`

#### HDP 2.4, 2.5, and 2.6 (x86\_64)

`IBM-SPSS-ModelerEssentialsR-ambari-2.1-HDP-2.3-8.4.0.0-1.x86_64.rpm`

#### HDP 2.6 (PPC64LE)

`IBM-SPSS-ModelerEssentialsR-ambari-2.1-HDP-2.3-8.4.0.0-1.ppc64le.rpm`

#### HDP 2.4, 2.5, and 2.6 (Ubuntu)

`IBM-SPSS-ModelerEssentialsR-ambari-3.1.0.0_3.1.0.0_amd64.deb`

- b. Install the RPM or DEB. In the following example, the command installs Essentials for R on BigInsights 4.2.

```
rpm -i IBM-SPSS-ModelerEssentialsR-ambari-2.1-BI-4.1-8.4.0.0-1.x86_64.rpm
```

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

```
dpkg -i IBM-SPSS-ModelerEssentialsR-ambari-3.1.0.0_3.1.0.0_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 21.
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 4. 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	1.1, 1.2	hive-jdbc-*.jar	Apache
MySQL	5.1, 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).  
BigSQL is a BigInsights value-add service. As such its version is identical to the BigInsights version. 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. 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. Refresh the Analytic Metastore service.
5. Restart each and every instance of the Analytic Server service.

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

## 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>.

## Apache Spark

If you want to use Spark (version 1.5 or later) with an HCatalog input data source, you must manually add the `spark.version` property to the custom `analytics.cfg` file.

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 and restart all Analytic Server services from the Amabri console.

**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**

## 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

- 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_0.6-3.1.0.0.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`

## 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 `<service>` to the Knox configuration:

```
<service>
  <role>ANALYTICSERVER</role>
  <uri>http://{analyticserver-host}:{analyticserver-port}/analyticserver</uri>
</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. On the Knox server, create the sub directory `<knox_server>/data/service/analyticserver/3.1`, 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` (for example, `/opt/ibm/spss/analyticserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/knox/analyticserver/3.1/*.xml`)
2. In `<knox_server>/bin`, run the script `./knoxcli.sh redeploy --cluster default`
3. 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_0.6-3.1.0.0.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_0.6-3.1.0.0.jar`).
4. In the Ambari user interface, add the following element in **Knox > Configs > Advanced topology**:

```
<service>
  <role>ANALYTICSERVER</role>
  <url>http://{AS-Host}:{AS-port}/analyticserver</url>
</service>
```
5. In the Ambari user interface, add or update the users in **Knox > Configs > Advanced users-ldif** (for example `admin`, `qauser1`, `qauser2`).
6. Restart LDAP from **Knox > Service Actions > Start Demo LDAP**.
7. 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` 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  
 to the Knox server node:  
 {knox}/data/services/analyticserver/3.1
- 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>AcIsAuthz</name>
      <enabled>true</enabled>
    </provider>
  </gateway>

  <!--other service-->
  <service>
    <role>ANALYTICSERVER</role>
    <!--replace the {AS-host}nas {AS-port} with real value-->
    <url>http://{AS-host}:{AS-port}/analyticserver</url>
  </service>
</topology>
```

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`.

---

## Migrating IBM SPSS Analytic Server on Ambari

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

### Migrating from Analytic Server 3.0/3.0.1 to 3.1

If you have an existing installation of Analytic Server 3.0 or 3.0.1 and have purchased 3.1.0, then you can migrate your 3.0/3.0.1 configuration settings to your 3.1.0 installation.

1. Install the new Analytic Server version according to the instructions in “Installation on Ambari” on page 5.
2. Copy the analytic root from your old installation to your new one.
  - a. If you are unsure of the location of the analytic root, run `hadoop -fs ls`. The path to the analytic root will be of form `/user/aeuser/analytic-root`, where `aeuser` is the userid that owns the analytic root.
  - b. Remove the analytic root on the new server.
  - c. Use `hadoop fs -copyToLocal` and `hadoop fs -copyFromLocal` to copy the old server's analytic root to the new server's `/user/as_user/` folder (ensure the owner is set as `as_user`).
3. On new server, open the Ambari/Cloudera Manager console, and stop the Analytic Server service (on Ambari, ensure that the Analytic Metastore service is running).
4. 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.

5. Clear the Zookeeper state. In the Zookeeper bin directory (for example, `/usr/hdp/current/zookeeper-client` on Hortonworks or `/usr/iop/current/zookeeper-server` on BigInsights), run the following command.  

```
./zkCli.sh rmr /AnalyticServer
```
6. 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_3.0.1.0.xxx.zip
```
7. In the Ambari/Cloudera Manager console, 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

**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.

1 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 BigInsights 4.2, the ESSENTIALR directory is located in `/var/lib/ambari-server/resources/stacks/BigInsights/4.2/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

If you plan to use MySQL as Analytic Server's metadata repository, follow the instructions for "Configuring MySQL for Analytic Server."

### 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" on page 30.
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 `-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 3.1.0

#### 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 5. Analytic Server self-extracting binary files*

Description	Binary filename
IBM SPSS Analytic Server 3.1 for Cloudera 5.8, 5.9, and 5.10 Ubuntu English	spss_as-3.1-cdh5.8-5.10-ubun.bin
IBM SPSS Analytic Server 3.1 for Cloudera 5.8, 5.9, and 5.10 Linux x86-64 English	spss_as-3.1-cdh5.8-5.10-1x86.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. Restart Cloudera Manager after the installation is complete.
4. Open the Cloudera Manager interface (for example, `http://${CM_HOST}:7180/cm/` with the default login credentials of `admin/admin`), refresh the **Remote Parcel Repository URLs**, 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. Configure MySQL for Analytic Server.
8. 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:
  - Analytic Server metastore host name
  - Analytic Server metastore database name
  - Analytic Server metastore user name
  - Analytic Server metastore password

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.

**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-3.1.0.0-el7.parcel
- AnalyticServer-3.1.0.0-el7.parcel.sha
- manifest.json

SuSE Linux requires the following files:

- AnalyticServer-3.1.0.0-sles11.parcel
  - AnalyticServer-3.1.0.0-sles11.parcel.sha
  - manifest.json
- or
- AnalyticServer-3.1.0.0-sles12.parcel
  - AnalyticServer-3.1.0.0-sles12.parcel.sha

Ubuntu Linux requires the following files:

- AnalyticServer-3.1.0.0-trusty.parcel
  - AnalyticServer-3.1.0.0-trusty.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).

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

3. Click **Distribute > Activate**.

The **AnalyticServer** parcel shows as distributed and activated.

## Upgrading to Analytic Server 3.1.0 on Cloudera

If you have an existing installation of Analytic Server 3.0/3.0.1, you can upgrade the installation in place to version 3.1.0.

1. In the Cloudera Manager, stop then delete Analytic Server service.
2. In the Cloudera Manager, deactivate the previous version of Analytic Server.
3. Refer to the "Online" or "Offline" sections in "Migrating IBM SPSS Analytic Server on Ambari" on page 25 for instructions on installing Analytic Server 3.1.0.

4. After Analytic Server service is installed and added in Cloudera Manager, run **Refresh Analytic Server Binaries**. Analytic Server 3.1.0 is now ready for use.

---

## 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 **security\_cfg** parameter defines the registry of users and groups that can be added as principals to the Analytic Server system.

By default, a basic registry is defined with a single user, admin, with password admin. You can change the registry by editing **security\_cfg** or configuring Kerberos as the security provider. The **security\_cfg** parameter can be found in the **Analytic Server Advanced Configuration Snippet** section of the Configuration tab of the Analytic Server service.

**Note:** If you edit the **security\_cfg** parameter to alter the registry, you then need to add any new users as principals to the Analytic Server system. See the *IBM SPSS Analytic Server Administrator's Guide* for details on tenant management.

### Making changes to the basic registry

The basic registry allows you to define a database of users and groups within the **security\_cfg** parameter.

The default basic registry looks like the following.

```
<basicRegistry id="basic" realm="ibm">
  <user name="admin" password="admin"/>
</basicRegistry>
```

An example of an altered basic registry follows.

```
<basicRegistry id="basic" realm="ibm">
  <user name="user1" password="{xor}Dz4sLG5tbGs="/>
  <user name="user2" password="Pass"/>
  <user name="user3" password="Pass"/>
  <user name="user4" password="Pass"/>
  <user name="admin" password="{xor}KzosKw="/>
  <group name="Development">
    <member name="user1"/>
    <member name="user2"/>
  </group>
  <group name="QA">
    <member name="user3"/>
    <member name="user4"/>
  </group>
  <group name="ADMIN">
    <member name="user1"/>
    <member name="admin"/>
  </group>
</basicRegistry>
```

Passwords can be encoded to obfuscate their values with the securityUtility tool, which is in {AS\_ROOT}/ae\_wlpserver/bin.

```
securityUtility encode changeit  
  {xor}Pdc+MTg6Nis=
```

**Note:** See [http://www-01.ibm.com/support/knowledgecenter/SSD28V\\_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp\\_command\\_securityutil.html](http://www-01.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp_command_securityutil.html) for details of the securityUtility tool.

**Note:** The basic registry is useful in a sandbox environment, but is not recommended for a production environment.

## Configure an LDAP registry

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

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="(&(uid=%v)(objectClass=inetOrgPerson))"  
    groupFilter="(&(cn=%v)(|(objectclass=organizationalUnit)))"  
    groupMemberIdMap="posixGroup:memberUid"/>  
</ldapRegistry>
```

For more examples of configurations, see the templates folder {AS\_ROOT}/ae\_wlpserver/templates/config.

**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}0zo5PiozKxYdEgwPDaWeDG1uDz4sLCg7"/>
  <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.

1. Create accounts in the Kerberos user repository for all users you plan to give access to Analytic Server.

**Note:** If the Analytic Server installation uses a basic registry, it must include the Kerberos user accounts, using "-" as the password. An example follows.

```
<basicRegistry id="basic" realm="ibm">
  <user name="admin" password="-"/>
  <user name="user1" password="-"/>
  <user name="user2" password="-"/>
  <group name="group1">
    <member name="admin"/>
    <member name="user1"/>
    <member name="user2"/>
  </group>
  <group name="group2">
    <member name="admin"/>
    <member name="user1"/>
  </group>
</basicRegistry>
```

2. 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.
3. 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.
4. 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 parameters 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** 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 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.

## 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 Analytic Server **Analytic Server Advanced Configuration Snippet (Safety Valve) for core-site.xml** area.  

```
- hadoop.proxyuser.as_user.hosts = *
- hadoop.proxyuser.as_user.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).

**Note:** The properties are automatically added in Ambari (based on values in the Analytic Server configuration).

## Disabling Kerberos

1. Disable Kerberos in the Ambari console.
2. Stop the Analytic Server service.
3. Remove the following parameters 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.

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

## 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 16.
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 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
DashDB	Bluemix Service	db2jcc.jar	IBM
DB2 for Linux, UNIX, and Windows	11.1, 10.5, 10.1, 9.7	db2jcc.jar	IBM

Table 6. Supported databases (continued)

Database	Supported versions	JDBC driver jars	Vendor
DB2 z/OS	11, 10	db2jcc.jar, db2_license_cisuz.jar	IBM
Greenplum	5.x	postgresql.jar	Greenplum
Hive	1.1, 1.2	hive-jdbc-*.jar	Apache
MySQL	5.1, 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.

## 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>.

## Apache Spark

If you want to use Spark (version 1.5 or later) with an HCatalog input data source, you must manually add the `spark.version=X.X.0` property (for example, `spark.version=2.0.0`)

1. Open Cloudera Manager and add or update the following properties in the Analytic Server **Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties** area.

`spark.version=2.0.0`

2. Save the configuration and restart all Analytic Server services from Cloudera Manager.

## 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 Analytic Server 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/114200v21.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/114200v21.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.

## 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
```

---

## 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 3.0/3.0.1 and have purchased a newer version, then you can migrate your 3.0/3.0.1 configuration settings to your new installation.

**Restriction:** Your 3.0/3.0.1 and new installations cannot coexist on the same Hadoop cluster. If you configure your new installation to use the same Hadoop cluster as your 3.0/3.0.1 installation, the 3.0/3.0.1 installation will no longer function.

### Migration steps, 3.0/3.0.1 to newer version

1. Install the new installation of Analytic Server according to the instructions in “Installation on Cloudera” on page 30.
2. Copy the analytic root from your old installation to your new one.
  - a. If you are unsure of the location of the analytic root, run `hadoop -fs ls`. The path to the analytic root will be of form `/user/aeuser/analytic-root`, where `aeuser` is the userid that owns the analytic root.
  - b. Change ownership of the analytic root from `aeuser` to `as_user`

```
hadoop dfs -chown -R {as_user:{group}} {path to 3.0/3.0.1 analytic-root}
```

**Note:** If you plan to use the existing Analytic Server installation after migration, make a copy of the `analytic-root` directory in HDFS and then change the ownership on the copy of the directory.

  - c. Log in to the host of the new Analytic Server installation as `as_user`. Delete the `/user/as_user/analytic-root` directory, if it exists.
  - d. Run the following copy script.

```
hadoop distcp hftp://{host of 3.0/3.0.1 namenode}:50070/{path to 3.0/3.0.1 analytic-root}  
hdfs://{host of 3.0/3.0.1 namenode}/user/as_user/analytic-root
```
3. In Cloudera Manager, stop the Analytic Server service.
4. 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 your IBM technical support representative if you encounter problems with the **configcollector** script.

5. Clear the Zookeeper state. In the Zookeeper bin directory (for example, `/opt/cloudera/parcels/CDH-5.4....../lib/zookeeper/bin` on Cloudera), run the following command.

```
./zkCli.sh rmr /AnalyticServer
```
6. 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/analyticsserver/3.1/ASConfiguration_3.0.0.0.xxx.zip
```

7. 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.

---

## 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 on MapR 5.0 or 5.1

1. Navigate to the [IBM Passport Advantage® Web Site](#) and download the MapR self-extracting binary file.

Table 7. MapR self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1 for MapR 5.0, and 5.1 Linux x86-64 English	spss_as-3.1.0-mapr5.0-5.2-1x86.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-3.1.0.0-1.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. 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>.

5. 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.

6. Add users and groups. Create a file that is named `security_cfg.xml` in the path `<as_installation_path>/ae_wlpserver/usr/servers/aeserver` and add the following lines to the file:

```
<server>
  <basicRegistry id="basic" realm="ibm">
    <user name="admin" password="test"/>
  </basicRegistry>
</server>
```

In the default state, the XML file contains only the admin user. You must manually add other users and groups in the `<basicRegistry>` setting, or change the setting to `ldapRegistry`.

7. 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 to 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
```

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



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

- 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.

- 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`
- 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`).

- 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.

- 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`.

- 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.

- 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.

- Start and stop Analytic Server.
  - Run the following command to start Analytic Server:

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

- Run the following command to stop Analytic Server:

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

## Installing Analytic Server 3.1 on MapR 5.2

- Navigate to the [IBM Passport Advantage® Web Site](#) and download the MapR self-extracting binary file.

Table 8. MapR self-extracting binary files

Description	Binary filename
IBM SPSS Analytic Server 3.1 for MapR 5.0, 5.1 and 5.2 Linux x86-64 English	spss_as-3.1.0-mapr5.0-5.2-1x86.bin

- The remaining steps for installing Analytic Server are mostly the same as those for installing Analytic Server 3.1 on MapR 5.0 or 5.1. However, the “Enabling Apache HBase” on page 48 and “Enabling Apache Spark” on page 49 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>/cluster1/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-1.2/lib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
cd /opt/mapr/hive/hive-1.2/hcatalog/share/hcatalog
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

`<as_analytic_root>` is the analytic root path that is defined in “Installing Analytic Server on MapR” on page 43.

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-1.2/lib" includes="*.jar"/>
<fileset dir="/opt/mapr/hive/hive-1.2/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-1.2/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-1.2/auxlib" includes="*.jar"/>
```

Run the following commands after adding the previous line:

```
cd /opt/mapr/hive/hive-1.2/auxlib
hadoop fs -put *.jar <as_analytic_root>/cluster1/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 9. 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 3.1.0 on MapR 5.0/5.1

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>/cluster1/classpath
```

<as\_analytic\_root> is the analytic root path that is defined in "Installing Analytic Server on MapR" on page 43.

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 3.1.0 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>/cluster1/classpath
```

The <as\_analytic\_root> is the path that is set in the 12th step in "Installing Analytic Server on MapR" on page 43.

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 3.1.0 on MapR 5.0/5.1

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>/cluster1/classpath
```

`<as_analytic_root>` is the analytic root path that is defined in “Installing Analytic Server on MapR” on page 43.

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 3.1.0 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_2-3.1.0.0.jar
```

#### Spark 2.x

1. Delete the following file:  
<as\_installation\_path>/ae\_wlpserver/usr/servers/aeserver/modules/spark/com.ibm.spss.sparkmapreduce-3.1.0.0.jar

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 lines to the tag:

```
<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"/>
```

**Note:** The newly added lines must be placed above the other maprLib child tags.

3. 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>/cluster1/classpath

The <as\_analytic\_root> is the path that is set in the 12th step in “Installing Analytic Server on MapR” on page 43.

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 16.
2. Run the installer `spss_er-8.4.0.0-mapr5-1x86_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>/cluster1/configuration/cf.zip
hadoop fs -put cf.zip <as_analytic_root>/cluster1/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, a 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
```

## 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-3.1.0.0-1.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 3.0 or 3.0.1 to version 3.1.0 on MapR.

1. Install Analytic Server 3.1.0 on a MapR cluster by following the installation instructions in “Installing Analytic Server on MapR” on page 43.
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 3.0/3.0.1 and 3.1.0 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 3.0/3.0.1 and 3.1.0 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 3.1.0 cluster nodes when the old Analytic Server 3.0/3.0.1 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 3.0/3.0.1 cluster nodes when the new Analytic Server 3.1.0 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 3.0/3.0.1 cluster nodes when the old cluster includes NFS, and NFS is also mounted on one of the new Analytic Server 3.1.0 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 3.1.0 cluster nodes when the new cluster includes NFS, and NFS is also mounted on one of the old Analytic Server 3.0/3.0.1 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 3.1.0, but ensure that the metadata database is still running.
5. Collect the configuration settings from the old Analytic Server 3.0/3.0.1 cluster installation.
  - a. Copy the configcollector.zip archive from the new Analytic Server 3.1.0 cluster installation to <old\_as\_installation\_path>/tools on the old Analytic Server 3.0/3.0.1 cluster installation.
  - b. Extract the configcollector.zip contents on the old Analytic Server 3.0/3.0.1 cluster installation. A new configcollector subdirectory is created in the old Analytic Server 3.0/3.0.1 cluster installation.
  - c. Run the configuration collector tool in the old Analytic Server 3.0/3.0.1 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 3.0/3.0.1 cluster installation.

**Important:** The provided **configcollector** script may not be compatible with the most recent Analytic Server version. Contact your IBM technical support representative if you encounter problems with the **configcollector** script.

6. Run the migration tool on the new Analytic Server 3.1.0 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_3.0.0.0.xxx.zip
```

7. Start Analytic Server 3.1.0.

---

## 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>/cluster1/classpath

cd /opt/mapr/hive/hive-1.2/lib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
cd /opt/mapr/hive/hive-1.2/hcatalog/share/hcatalog
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath

cd /opt/mapr/hbase/hbase-0.98.12/lib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath

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

hadoop fs -put *.jar <as_analytic_root>/cluster1/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 3.1.0

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 10. Analytic Server self-extracting binary file

Description	Binary file name
IBM SPSS Analytic Server 3.1 for FusionInsight HD 2.6 Linux x86-64 English	spss_as-3.1-fhd2.6-1x86.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 3.1.0:

```
# yum install -y IBM-SPSS-AnalyticServer-3.1.0.0-1.x86_64.rpm
```

4. Log in with omm and create the analyticsserver.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/analyticsserver/3.1/analyticsserver.keytab HTTP/_HOST@HADOOP.COM
kadmin > ktadd -k /opt/ibm/AnalyticServer/analyticsserver.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-e17-9.noarch.rpm
# yum -y install mysql57-community-release-e17-9.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/mysql.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/analyticsserver/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.dbscr
```

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/analyticsserver/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/analyticsserver/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/analyticsserver/3.1/ae\_wlpserver/usr/servers/aeserver/configuration.

13. Copy the native Hadoop files to the following folder: /opt/ibm/spss/analyticsserver/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/analyticsserver/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=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAGk3bIuya2BzXYeYfC0rxo=
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/analyticsserver/3.1/analyticsserver.keytab
hdfs.keytab=/opt/ibm/spss/analyticsserver/3.1/analyticsserver.keytab
ae.db.connect.method=Basic
kdcrealm=HADOOP.COM
kdcservice=172.16.155.212:21732
encryption.keystore.password=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAMDJl7PVsvdIyL1zjeS8w=
encryption.keystore.base64=zs70zgAAAAIAAAABAAAAAwA6Y29tLnNwc3MuYUWuZW5jcnlwdG1vbi5wcm92aWR1ci5lbnNyeXB0aW9ucHJvdmlkZXJpbXBsLmF1c2AAUg2Ahyr00ABXNyAB1qYXZh
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/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
zookeeper.server.principal=zookeeper/hadoop.hadoop.com@HADOOP.COM
zookeeper.server.keytab=/opt/ibm/spss/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64/zookeeper.conf/zookeeper.keytab
zookeeper.server.jaas.conf=/opt/ibm/spss/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/jaas.conf
krb5.conf=/home/omm/kerberos/var/krb5kdc/krb5.conf
```

- /opt/ibm/spss/analyticsserver/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/analyticsserver/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/analyticsserver/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/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/Bigdata/jdk/jre/lib/amd64:/usr/sbin:/usr
IBM_SPSS_AS_NATIVE_PATH=/opt/ibm/spss/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
LD_LIBRARY_PATH=/opt/ibm/spss/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/Bigdata/jdk/jre/lib/amd64:/op
SPARK_HOME=/opt/ibm/spss/analyticsserver/3.1/spark-client
HADOOP_HOME="/opt/client_hdfs/HDFS/hadoop"

```

17. Modify the /opt/ibm/spss/analyticsserver/3.1/ae\_wlpserver/usr/servers/aeserver/server.xml file as follows:

```

<server description="new server">
<!-- Enable features -->
<featureManager>
<feature>servlet-3.1</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" async
<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/analyticsserver/3.1/ae\_wlpserver/usr/servers/aeserver/ folder. The file contents should resemble the following:

```

<server>
<application context-root="/analyticsserver" id="AS_B00T" location="AE_B00T.war" name="AS_B00T" type="war">
<classloader>
<privateLibrary>
<fileset dir="${wlp.install.dir}/usr/servers/aeserver/apps/AE_B00T.war/WEB-INF/lib" includes="*.jar"/>
<fileset dir="/usr/share/java" includes="*.jar"/>
<fileset dir="${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="/analyticsserver/help"/>
</server>

```

19. Create a `security_cfg.xml` file in the `/opt/ibm/spss/analyticsserver/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/analyticsserver/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/analyticsserver/3.1/ae_wlpserver/usr/servers/aeserver/configuration

# -server
# possible 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/analyticsserver/3.1/bin/hdfsUpdate.sh`).
23. Run the `start.sh` script (`/opt/ibm/spss/analyticsserver/3.1/bin/start.sh`) to start the Analytic Server service; run the `stop.sh` script (`/opt/ibm/spss/analyticsserver/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/analyticsserver/admin/ibm`

---

## 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 of the server that hosts 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 `clem` 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, a 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 11. SLM Tag Properties

Properties	Description
<code>license.metric.logger.output.enabled</code>	Controls the SLM log file generation. The default value is <code>False</code> .
<code>license.metric.logger.output.dir</code>	The relative path to the directory that stores the SLM tag files. The default directory is <code>&lt;as_installation_path&gt;/logs</code> .
<code>license.metric.logger.output.SLMLogFrequency</code>	The time interval (unit: milliseconds) for collecting SLM logs.
<code>license.metric.logger.file.size</code>	The maximum SML tag file size, in bytes.
<code>license.metric.logger.file.number</code>	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.*`

### 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/analyticsserver/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.

### LZO Compressor does not work on BigInsights 4.2.X

You must execute `yum install lzo* hadoop-lzo*` to install the LZO components in the BigInsights 4.2.X. Because the installation behavior changed in the BigInsights 4.2 environment, the LZO components now install under the `/usr/lib/hadoop-lzo/lib` directory (this directory does not include the Hadoop execution environment). Use the following steps to manually configure LZO components on BigInsights 4.2.X.

1. Log on to the host that is running Analytic Server as the Analytic Server user (by default `as_user`).

**Note:** You can use the Ambari console to determine the host name.

2. Copy the Hadoop LZO jar file into the Analytic Server libraries directory. For example:

```
cp /usr/lib/hadoop-lzo/lib/hadoop-lzo*.jar {AS_ROOT}/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/
```

3. Copy the LZO native libraries into the Analytic Server native libraries directory. For example:

```
cp /usr/lib/hadoop-lzo/lib/native/* {AS_ROOT}/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
```

4. Run the **refresh** script that is located in the `{AS_ROOT}/bin` directory. For example:

```
cd /opt/ibm/spss/analyticsserver/3.1/bin
./refresh
```

5. Restart the Analytic Server service in the Ambari console.

## 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 IBM SPSS Modeler 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).

## 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 21.

### java.net.SocketTimeoutException: Read timed out

Change the Liberty ND timeout environment variable as follows:

```
export LIBERTYND_READ_TIMEOUT=<milliseconds>
```

where `<milliseconds>` is the number of seconds to use for the JMX read timeout.

### java.io.IOException: CWWKX7202E: The timeout value 60 (seconds) for command ./server start expired

Add the following to the Controller Server `server.xml`

```
<!-- Increase start and stop server timeout to accommodate slow hardware -->
<serverCommands startServerTimeout="120" stopServerTimeout="120"/>
```

### java.lang.OutOfMemoryError: Java heap space

Add the following lines to `jvm.options` on every member of the HA cluster.

```
-Xms512M
-Xmx2048M
```

### "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.

**Analytic Server servers hang occasionally upon server shutdown**

Kill the server manually.

---

## 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-IBM products. 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](http://www.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](http://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