

IBM SPSS Analytic Server
バージョン 3.1.1

インストールと構成のガイド

IBM

注記

本書および本書で紹介する製品をご使用になる前に、77 ページの『特記事項』に記載されている情報をお読みください。

本書は、IBM SPSS Analytic Server バージョン 3、リリース 1、モディフィケーション 1、および新しい版で明記されていない限り、以降のすべてのリリースおよびモディフィケーションに適用されます。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM SPSS Analytic Server
Version 3.1.1
Installation and Configuration Guide

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

目次

第 1 章 前提条件	1	マイグレーション	48
第 2 章 Ambari のインストールおよび構成	3	Cloudera での Analytic Server のアンインストール	49
Ambari 固有の前提条件	3	第 4 章 MapR のインストールおよび構成	51
IBM SPSS Analytic Server インストールの事前チェック・ツールと事後チェック・ツール	3	MapR の概要	51
Ambari でのインストール	5	MapR での Analytic Server のインストール	51
オフライン・インストール	7	MapR の構成	55
外部管理される MySQL 環境に対する Analytic Server のインストール	11	データベース・プッシュバックの有効化	55
構成	12	Apache Hive の有効化	56
セキュリティー	12	Apache HBase の有効化	57
Essentials for R に対するサポートの有効化	18	Apache Spark の有効化	58
リレーショナル・データベース・ソースの有効化	21	機能フラグの有効化	60
HCatalog データ・ソースの有効化	22	R の有効化	60
Analytic Server で使用するポートの変更	23	LZO の有効化	61
高可用性 Analytic Server	23	MapR の SLM タグの有効化	61
スモールデータ向けの JVM オプションの最適化	25	MapR 用の IBM SPSS Analytic Server クラスターのセットアップ	61
クライアント依存関係の更新	25	MapR のアンインストール	62
Apache Knox の構成	25	MapR での IBM SPSS Analytic Server のマイグレーション	62
Ambari での IBM SPSS Analytic Server のマイグレーション	30	MapR のトラブルシューティング	63
アンインストール	31	第 5 章 Huawei FusionInsight HD のインストールおよび構成	65
Essentials for R のアンインストール	31	FusionInsight HD の概要	65
第 3 章 Cloudera のインストールおよび構成	33	Huawei FusionInsight HD でのインストール	65
Cloudera の概要	33	第 6 章 IBM SPSS Analytic Server で使用するための IBM SPSS Modeler の構成	69
Cloudera 固有の前提条件	33	第 7 章 SLM タグを使用したライセンス交付の追跡	71
Analytic Server 用の MySQL の構成	33	第 8 章 トラブルシューティング	73
Cloudera でのインストール	34	特記事項	77
Cloudera の構成	36	商標	78
セキュリティー	37		
Essentials for R に対するサポートの有効化	42		
リレーショナル・データベース・ソースの有効化	43		
HCatalog データ・ソースの有効化	44		
Apache Impala の構成	45		
Analytic Server で使用するポートの変更	46		
高可用性 Analytic Server	47		
スモールデータ向けの JVM オプションの最適化	48		

第 1 章 前提条件

Analytic Server をインストールする前に、以下の情報を確認してください。

システム要件

最新のシステム要件情報については、IBM Technical Support サイトの <http://publib.boulder.ibm.com/infocenter/prodguid/v1r0/clarity/softwareReqsForProduct.html> にある「Detailed system requirements」レポートを使用してください。このページでは以下を行うことができます。

1. 製品名として SPSS Analytic Server を入力して、「**Search**」をクリックします。
2. 目的のバージョンとレポートの範囲を選択して、「**Submit**」をクリックします。

Power Systems

クラスター内のすべてのホストに IBM XLC コンパイラーおよび XLF コンパイラーがインストールされており、PATH に含まれていることを確認してください。

これらのコンパイラー用のライセンスの取得の詳細については、以下の Web サイトで確認できます。

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

Hortonworks Data Platform (HDP)

Analytic Server をインストールする前に、1 つ以上の HDP クライアントがクラスター環境にデプロイされていることを確認する必要があります。Ambari Manager をホストするノードでは /usr/hdp ディレクトリが想定されるため、HDP クライアントが存在しない場合は Analytic Server に障害が発生します。

Hive/HCatalog

NoSQL データ・ソースを使用する予定の場合、Hive および HCatalog をリモート・アクセス用に構成します。さらに、hive-site.xml に、アクティブな Thrift Hive Metastore サーバーを示す *hive.metastore.uris* プロパティが `thrift://<host_name>:<port>` の形式で設定されていることを確認します。詳しくは、使用している Hadoop ディストリビューションの資料を参照してください。

注: Analytic Server Metastore は、Hive Metastore と同じマシンにインストールすることはできません。

Hive 2.1 を使用する場合は、Ambari コンソールで「**Interactive Query**」設定を有効にして Hive 2.1 を有効化してから、Analytic Server インストール時に `hive.version` プロパティとして 2.x と入力する必要があります。

1. Ambari コンソールを開き、「**Analytic Server Advanced analytics.cfg**」セクションで以下のプロパティを追加します。
 - キー: `hive.version`
 - 値: 適切な Hive バージョン (2.x など) を入力します。
2. 構成を保存します。

注: Hive 2.1 は、HDP 2.5 および 2.6 でサポートされています。

メタデータ・リポジトリ

デフォルトでは、Analytic Server は MySQL データベースをインストールして使用します。あるいは、既存の Db2 インストール済み環境を使用するように Analytic Server を構成することもできます。選択するデータベースのタイプにかかわらず、データベースには UTF-8 のエンコードが必要です。

MySQL

MySQL のデフォルト文字セットはバージョンとオペレーティング・システムによって異なります。ご使用の MySQL インストール済み環境が UTF-8 に設定されているかどうかを確認するには、以下の手順を使用してください。

1. MySQL のバージョンを確認します。

```
mysql -V
```

2. MySQL のコマンド・ライン・インターフェースから以下の照会を実行して、MySQL のデフォルト文字セットを確認します。

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

文字セットが既に UTF-8 に設定されている場合、追加の変更は不要です。

3. MySQL のコマンド・ライン・インターフェースから以下の照会を実行して、MySQL のデフォルト照合を確認します。

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

照合が既に UTF-8 に設定されている場合、追加の変更は不要です。

4. デフォルトの文字セットまたは照合が UTF-8 でない場合、文字セットを UTF-8 に変更するために `/etc/my.cnf` を編集して MySQL デーモンを再始動する方法について、MySQL の資料を参照してください。

Db2 Db2 の構成について詳しくは、Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.5.0/com.ibm.db2.luw.kc.doc/welcome.html) を参照してください。

高可用性クラスター

ロード・バランサー

高可用性クラスターには、セッション・アフィニティー (スティッキー・セッションと呼ばれることもあります) をサポートするロード・バランサーが必要です。Analytic Server は、Cookie 「request-token」 でセッションを識別します。これにより、アプリケーションによって制御されるセッション・アフィニティーで使用するために、ユーザー・ログインの期間にわたってセッションが識別されます。セッション・アフィニティーがどのようにサポートされるかについて詳しくは、ご使用の特定のロード・バランサーの資料を参照してください。

第 2 章 Ambari のインストールおよび構成

Ambari 固有の前提条件

一般的な前提条件に加えて、以下の情報を確認してください。

サービス

Analytic Server は Ambari サービスとしてインストールされます。Analytic Server をインストールする前に、以下のクライアントが Ambari サービスとしてインストールされていることを確認する必要があります。

- HDFS/HDFS_CLIENT
- MAPREDUCE2/MAPREDUCE2_CLIENT
- HIVE/HIVE_CLIENT
- SPARK/SPARK_CLIENT (Spark 1.x が使用される場合)
- SPARK2/SPARK2_CLIENT (Spark 2.x が使用される場合)
- HBASE/HBASE_CLIENT (HBASE が使用される場合)
- YARN
- Zookeeper

パスワード無し SSH

root ユーザー用に Analytic Metastore ホストとクラスター内のすべてのホストの間にパスワード無し SSH をセットアップしてください。

IBM SPSS Analytic Server インストールの事前チェック・ツールと事後チェック・ツール

事前チェック・ツールの概要

Analytic Server インストールの事前チェック・ツールは、Analytic Server インストールの前に潜在的な環境問題を特定することにより、インストール問題と実行時エラーを減らすために役立ちます。

事前チェック・ツールは、以下の項目を検証します。

- ローカル・システム上の OS と Ambari のバージョン
- ローカル・システム上の OS の ulimit 設定
- ローカル・システム上の使用可能なディスク・スペース
- Hadoop バージョン
- Ambari サービスの可用性 (HDFS、HCatalog、Spark、Hive、MapReduce、Yarn、Zookeeper など)
- Analytic Server 固有の Ambari 設定

注: 事前チェック・ツールは、Analytic Server インストールの後でも実行できます。

事後チェック・ツールの概要

Analytic Server インストールの事後チェック・ツールは、Analytic Server インストールの後で、以下を処理するための REST API 要求をサブミットすることにより、構成問題を特定します。

- HDFS 内のデータ
- Hive/HCatalog 内のデータ
- 圧縮データ (deflate、bz2、snappy を含む)
- PySpark でのデータ
- ネイティブ SPSS コンポーネントを使用するデータ (alm、tree、neuralnet、scoring、tascoreing を含む)
- MapReduce でのデータ
- メモリー内の MapReduce でのデータ

ツールの場所と前提条件

事前チェック・ツールと事後チェック・ツールは、以下のディレクトリーにあります。

- **HDP**

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

- **Cloudera**

これらのツールは `AnalyticServer-*.jar` ファイルにパッケージされています (このファイルは `/opt/cloudera/csd` にあります)。

ツールは `root` として実行する必要があるため、Python 2.6.X 以上を必要とします。

Analytic Server をインストールする前に、Analytic Server サービスをホストするすべての Ambari ノード上で事前チェック・ツールを実行する必要があります。別のノード上でツールを実行するには、そのノードに `chktool` ディレクトリー全体をコピーする必要があります。

事前チェック・ツールが失敗を報告した場合は、Analytic Server インストールを続行する前に、それらの失敗に対処する必要があります。

`chktool` ディレクトリーは、Analytic Server 自己解凍型バイナリーの実行 (5 ページの『Ambari でのインストール』セクションのステップ 2) の後で使用可能になります。 7 ページの『オフライン・インストール』の実行を選択した場合、`chktool` ディレクトリーは、メタデータ RPM のインストールの後で使用可能になります。

事前チェック・ツールの実行

以下の事前チェックの例は、Ambari クラスタ `MyCluster` をチェックします。このクラスタは、`myambarihost.ibm.com:8080` で実行され、SSL が有効になっていて、ログイン資格情報 `admin:admin` を使用します。

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

Notes:

- `as_host` 値は、IP アドレスまたは完全修飾ドメイン名のいずれかによって指定する必要があります。
- パスワード引数が省略されると、ツールはパスワードの入力を求めるプロンプトを出します。
- `precheck.py` コマンドに含まれている使用法ヘルプは、`-h` 引数 (`python ./precheck.py -help`) と表示されます。
- `--cluster` 引数はオプションです (`--cluster` が使用されていない場合は、現在のクラスタが指定されます)。

事前チェック・ツールがチェックを実行しているときには、各チェックの状況がコマンド・ウィンドウに表示されます。失敗が発生した場合は、ログ・ファイル内の詳細情報を参照できます (ログ・ファイルの具体的な場所は、コマンド・ウィンドウで指示されます)。追加のサポートが必要な場合は、ログ・ファイルを IBM Technical Support に提供できます。

事後チェック・ツールの実行

事後チェック・ツールは、Analytic Server が適切に実行されていること、および単純なジョブを処理できることを検証します。以下の事後チェックの例は、特定の Analytic Server インスタンスをチェックします。このインスタンスは、myanalyticserverhost.ibm.com:9443 で実行され、SSL が有効になっていて、ログイン資格情報 admin:ibmspss を使用します。

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

Knox が Analytic Server と共に使用される場合、コマンドは以下のとおりです。

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

単一のチェックを実行するには、以下のコマンドを使用します。

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

Notes:

- パスワード引数が省略されると、ツールはパスワードの入力を求めるプロンプトを出します。
- postcheck.py コマンドに含まれている使用法ヘルプは、--h 引数 (python ./postcheck.py --help) と表示されます。

事後チェック・ツールがチェックを実行しているときには、各チェックの状況がコマンド・ウィンドウに表示されます。失敗が発生した場合は、ログ・ファイル内の詳細情報を参照できます (ログ・ファイルの具体的な場所は、コマンド・ウィンドウで指示されます)。追加のサポートが必要な場合は、ログ・ファイルを IBM Technical Support に提供できます。

Ambari でのインストール

基本的なプロセスは、次のとおりです。Analytic Server ファイルを Ambari クラスター内のホストにインストールし、続いて Analytic Server を Ambari サービスとして追加します。さらに詳細なステップは、以下のとおりです。

重要: Analytic Server では、Ambari-Server が root 以外のユーザーとして実行されている環境でのインストールはサポートされません。

- IBM パスポート・アドバンテージ Web サイトに移動し、ご使用のスタック、スタック・バージョン、およびハードウェア・アーキテクチャーに固有の自己解凍型バイナリー・ファイルを Ambari クラスター内のホストにダウンロードします。使用可能な Ambari バイナリーは以下のとおりです。

表 1. Analytic Server 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM® SPSS® Analytic Server 3.1.1 for Hortonworks Data Platform 2.4, 2.5, および 2.6, Linux x86-64, 英語	spss_as-3.1.1-hdp2.4-2.6-lx86_en.bin
IBM SPSS Analytic Server 3.1.1 for Hortonworks Data Platform 2.6, Linux on System p LE, 英語	spss_as-3.1.1-hdp2.6-lppc64_en.bin

2. 自己解凍型バイナリー・ファイルを実行し、指示に従って (オプションで) ライセンスを表示し、ライセンスを受け入れて、オンライン・インストールまたはオフライン・インストールを選択します。

オンライン・インストール

Ambari サーバー・ホストおよびクラスター内のすべてのノードが <https://ibm-open-platform.ibm.com> にアクセス可能な場合は、オンライン・インストールを選択してください。

オフライン・インストール

ご使用の Ambari サーバー・ホストがインターネットにアクセスできない場合は、オフラインを選択します。詳しくは、7 ページの『オフライン・インストール』を参照してください。

3. `var/lib/ambari-server/resources/stacks/<stack-name>/<stack-version>/services/ANALYTICSERVER/package/scripts` ディレクトリーから、`update_clientdeps.sh` スクリプトを適切な引数を指定 (例えば、`--help` 引数を使用) して実行します。

4. Ambari サーバーを再起動します。

```
ambari-server restart
```

5. オフライン・インストールを実行している場合は、`repoinfo.xml` および `IBM-SPSS-AnalyticServer-3.1.1.0.repo` の変更内容がまだ存在することを確認します。`repoinfo.xml` および `IBM-SPSS-AnalyticServer-3.1.1.0.repo` の変更内容が保存されていない場合は、IBM サポート担当員にお問い合わせください。

6. Ambari サーバーにログオンし、Ambari UI を使用して Analytic Server をサービスとしてインストールします。

メタデータ・リポジトリー

デフォルトでは、Analytic Server は MySQL を使用して、データ・ソース、プロジェクト、およびテナントに関する情報を追跡します。インストール時に、Analytic Server と MySQL の間の JDBC 接続で使用されるユーザー名 (**metadata.repository.user.name**) およびパスワード (**metadata.repository.password**) を指定する必要があります。インストーラーは MySQL データベースにそのユーザーを作成しますが、そのユーザーは MySQL データベースに固有であり、既存の Linux ユーザーや Hadoop ユーザーである必要はありません。

メタデータ・リポジトリーを Db2 に変更するには、以下のステップを実行します。

注: インストールの完了後にメタデータのリポジトリーを変更することはできません。

- a. 別のマシンに Db2 がインストールされていることを確認します。詳しくは、トピック 1 ページの『第 1 章 前提条件』の『メタデータ・リポジトリー』セクションを参照してください。
- b. Ambari の「Services」タブで、Analytic Server サービスの「Configs」タブに移動します。
- c. 「**Advanced analytics-env**」セクションを開きます。
- d. **as.database.type** の値を `mysql` から `db2` に変更します。
- e. 「**Advanced analytics-meta**」セクションを開きます。
- f. **metadata.repository.driver** の値を `com.mysql.jdbc.Driver` から `com.ibm.db2.jcc.DB2Driver` に変更します。
- g. **metadata.repository.url** の値を `jdbc:db2://{Db2_HOST}:{PORT}/{DBName}:currentSchema={SchemaName}`; に変更します。ここで、
 - `{Db2_HOST}` は、Db2 がインストールされているサーバーのホスト名です。

- {PORT} は、Db2 が listen しているポートです。
- {SchemaName} は、使用可能な、未使用のスキーマです。

入力する値がわからない場合は、Db2 管理者に協力を求めてください。

- h. **metadata.repository.user.name** および **metadata.repository.password** に、有効な Db2 資格情報を入力します。
- i. 「**Save**」をクリックします。

インストール後に変更してはならない構成設定

インストール後に以下の設定は変更しないでください。変更すると Analytic Server が動作しなくなります。

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

7. これで、Analytic Server のインスタンスが機能するようになりました。追加の構成はオプションです。Analytic Server の構成と管理について詳しくは、12 ページの『構成』のトピックを参照してください。既存構成の新規インストール済み環境へのマイグレーションについては、30 ページの『Ambari での IBM SPSS Analytic Server のマイグレーション』のトピックを参照してください。
8. Web ブラウザーを開き、アドレス `http://<host>:<port>/analyticserver/admin/ibm` を入力します。ここで、<host> は Analytic Server ホストのアドレスであり、<port> は Analytic Server が listen しているポートです。デフォルトではこれは 9080 です。この URL にアクセスすると、Analytic Server コンソールのログイン・ダイアログが開きます。Analytic Server 管理者としてログインします。デフォルトでは、このユーザー ID は admin であり、パスワードは admin です。

オフライン・インストール

オフライン・インストールの一般的なワークフローは、以下のとおりです。

自動による HDP の手順 - 3.1.1

インストール・プロセスが自動化され、この方法によるインストールが推奨されています。

1. IBM パスポート・アドバンテージ Web サイトに移動し、自己解凍型バイナリー・ファイルを Ambari クラスタ内のホストにダウンロードします。

表 2. Analytic Server 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for Hortonworks Data Platform	spss_as-ambari-offlineinstall_en.bin

2. 実行可能バイナリーを実行します。オフライン・インストールでは必要な RPM または DEB ファイルをダウンロードするため、<https://ibm-open-platform.ibm.com> にアクセス可能なコンピューターで実行する必要があります。実行可能バイナリー・ファイルは、使用可能な Ambari `<AS_INSTALLABLE_HOME>` ディストリビューション・ディレクトリーにあります。
3. 該当する `<AS_INSTALLABLE_HOME>` ディレクトリーのすべての内容を、インターネットにアクセスできるマシンから、(ファイアウォールで保護されている) Ambari マネージャー・ノードにコピーします。

- 必要な Analytic Server RPM/DEB ファイルが IBM-SPSS-AnalyticServer/packages ディレクトリーにあることを確認します。必要な RPM ファイルは、ご使用のディストリビューション、バージョン、およびアーキテクチャーによって異なります。

Redhat

IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.0-1.noarch.rpm

IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64.rpm

Ubuntu

IBM-SPSS-AnalyticServer-ambari-2.x_3.1.1.0_amd64.deb

IBM-SPSS-AnalyticServer_1_amd64.deb

- 以下のコマンドを実行してインストールを開始します。

```
apt-get install dpkg-dev
```

インストール中に、Analytic Server バージョン、JDBC ドライバー、Spark バージョン、Hive バージョンなどの入力を求めるプロンプトが出されます。

手動による HDP の手順 - 3.1.1

- IBM パスポート・アドバンテージ Web サイトに移動し、ご使用のスタック、スタック・バージョン、およびハードウェア・アーキテクチャーに固有の自己解凍型バイナリー・ファイルを Ambari クラスタ内のホストにダウンロードします。使用可能な Ambari バイナリーは以下のとおりです。

表 3. Analytic Server 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for Hortonworks Data Platform 2.4、2.5、および 2.6、Linux x86-64、英語	spss_as-3.1.1-hdp2.4-2.6-lx86_en.bin
IBM SPSS Analytic Server 3.1.1 for Hortonworks Data Platform 2.6、Linux on System p LE、英語	spss_as-3.1.1-hdp2.6-lppc64_en.bin

- 実行可能バイナリーを実行し、オフライン・インストールを指定します。オフライン・インストールでは必要な RPM または DEB ファイルをダウンロードするため、<https://ibm-open-platform.ibm.com> にアクセス可能なコンピューターで実行する必要があります。実行可能バイナリー・ファイルは、使用可能な Ambari <AS_INSTALLABLE_HOME> ディストリビューション・ディレクトリーにあります。
- 該当する <AS_INSTALLABLE_HOME> ディレクトリーのすべての内容を、インターネットにアクセスできるマシンから、(ファイアウォールで保護されている) Ambari マネージャー・ノードにコピーします。
- (ファイアウォールで保護されている) 該当する Ambari マネージャー・ノード上で、ご使用の特定のディストリビューション (HDP または Ubuntu) の手順を続行します。

重要: Analytic Server では、Ambari-Server が root 以外のユーザーとして実行されている環境でのインストールはサポートされません。

- 以下のコマンドを使用して、Ambari サーバーが実行していることを確認します。

```
ambari-server status
```

以下のようにして、Ambari サーバー・ノードをシャットダウンします (現在実行中の場合)。

```
ambari-server stop
```

- ローカル yum リポジトリを作成するツールをインストールします。

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

または

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

3. Analytic Server の RPM ファイルのリポジトリとして機能するディレクトリーを作成します。以下の例を参照してください。

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

4. 新規ディレクトリーに、必要な Analytic Server の RPM ファイルをコピーします。必要な RPM ファイルは、ご使用のディストリビューション、バージョン、およびアーキテクチャーによって異なります。

HDP 2.4、2.5、および 2.6 (x86_64)

```
IBM-SPSS-AnalyticServer-ambari-2.x-3.1.1.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64.rpm
```

5. ローカル・リポジトリーの定義を作成します。例えば、IBM-SPSS-AnalyticServer-3.1.1.0.repo というファイルを、/etc/yum.repos.d/ (RHEL、CentOS の場合) または /etc/zypp/repos.d/ (SLES の場合) に、以下の内容を指定して作成します。

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

6. ローカル yum リポジトリーを作成します。

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

7. root ユーザー・コマンド・ウィンドウから、cd を実行して <AS_INSTALLABLE_HOME>/IBM-SPSS-AnalyticServer ディレクトリーに移動し、./offLineInstall.sh を実行します。スクリプトは、以前に実行されたバイナリー実行可能インストール・コマンドに対する永続化された応答を読み取り、(rpm をインストールするための) 該当するプラットフォーム・コマンドを発行します。

注: ステップ 8 とステップ 9 は、外部管理される MySQL 環境を使用する場合にのみ適用されます。

8. AS_MetaStore として使用される MySQL インスタンスがインストールされているノード/ホストで add_mysql_user.sh スクリプトを実行します。
 - a. add_mysql_user.sh スクリプトを、/opt/AS_Installable/IBM-SPSS-AnalyticServer から、AS_MetaStore として使用される MySQL インスタンスがインストールされているノード/ホストにコピーします。例: /opt/AS_InstallTools
 - MySQL ノード/ホスト上で add_mysql_user.sh スクリプトを実行します。例:
./add_mysql_user.sh -u as_user -p spss -d aedb

Notes:

- ユーザー名およびパスワードは、Ambari 構成画面の AS_Metastore で入力されたデータベース・ユーザー名およびパスワードと一致する必要があります。
- コマンドを発行するように add_mysql_user.sh スクリプトを手動で更新できます (希望する場合)。
- セキュアな (root ユーザーによってアクセスされる) MySQL データベースに対して add_mysql_user.sh スクリプトを実行する場合は、-r パラメーターおよび -t パラメーターを使用して、dbuserid および dbuserid_password を渡します。スクリプトは、dbuserid および dbuserid_password を使用して、MySQL 操作を実行します。

注: 「AS_Configuration」画面の metadata.repository.url 設定 (「Advanced analytics-meta」) が MySQL データベース・ホストを指すように変更する必要があります。例えば、JDBC 設定 `mysql://{analytic_metastore_host}/aedb?createDatabaseIfNotExist=true` を `mysql://{MySQL_DB}/aedb?createDatabaseIfNotExist=true` に変更します。

9. ご使用の Ambari リポジトリ・ファイル `repoinfo.xml` (通常は `/var/lib/ambari-server/resources/stacks/$stackName/$stackVersion/repos/` に配置されています) に以下の行を追加して、ローカル yum リポジトリを使用するように更新します。

```
<os type="host_os">
  <repo>
    <baseurl>file:///path to local repository/</baseurl>
    <repopid>IBM-SPSS-AnalyticServer</repopid>
    <reponame>IBM-SPSS-AnalyticServer-3.1.1.0</reponame>
  </repo>
</os>
```

例の中の {path to local repository} は次のようになります。

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

10. 各 Ambari 非サーバー・クラスター・ノードに対して以下の手順を繰り返します。
 - a. 該当する `<AS_INSTALLABLE_HOME>` ディレクトリーのすべての内容を、インターネットにアクセスできるマシンから、Ambari 非サーバー・クラスター・ノードにコピーします。
 - b. ローカル yum リポジトリを作成するツールをインストールします。
`yum install createrepo (RHEL, CentOS)`

または

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

- c. Analytic Server の RPM ファイルのリポジトリとして機能するディレクトリを作成します。以下の例を参照してください。
`mkdir /home/root/repos/IBM-SPSS-AnalyticServer/x86_64`
- d. 新規ディレクトリに、必要な Analytic Server の RPM ファイルをコピーします。必要な RPM ファイルは、ご使用のディストリビューション、バージョン、およびアーキテクチャーによって異なります。

HDP 2.4, 2.5, および 2.6 (x86_64)

```
SPSS-AnalyticServer-ambari-2.x-3.1.1.0-1.noarch.rpm
```

```
IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64.rpm
```

- e. ローカル・リポジトリの定義を作成します。例えば、`IBM-SPSS-AnalyticServer-3.1.1.0.repo` というファイルを、`/etc/yum.repos.d/` (RHEL, CentOS の場合) または `/etc/zypp/repos.d/` (SLES の場合) に、以下の内容を指定して作成します。

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

- f. ローカル yum リポジトリを作成します。

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

11. 5 ページの『Ambari でのインストール』のトピックのステップ 5 に進みます。

手動による Ubuntu の手順 - 3.1.1

1. IBM パスポート・アドバンテージ Web サイトに移動し、ご使用のスタック、スタック・バージョン、およびハードウェア・アーキテクチャーに固有の該当する Ubuntu 自己解凍型バイナリー・ファイルを Ambari クラスタ内のホストにダウンロードします。使用可能なバイナリーは以下のとおりです。

表 4. *Analytic Server* 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for Hortonworks Data Platform 2.4、2.5、および 2.6、Linux x86-64、英語	spss_as-3.1.1-hdp2.4-2.6-lx86_en.bin

2. インターネットにアクセスできるマシンで、実行可能バイナリーを実行し、オフライン・インストールを指定します。オフライン・インストールでは必要な DEB ファイルをダウンロードするため、<https://ibm-open-platform.ibm.com> にアクセス可能なコンピューターで実行する必要があります。実行可能バイナリー・ファイルは、使用可能な Ambari <AS_INSTALLABLE_HOME> ディストリビューション・ディレクトリーにあります。
3. 必要な Analytic Server DEB ファイルは、以下のディレクトリーにあります。
IBM-SPSS-AnalyticServer/packages
4. Analytic Server 3.1.0 をインストールするには、以下のコマンドを使用します。

```
dpkg -i ./IBM-SPSS-AnalyticServer-ambari-2.x_3.1.1.0_amd64.deb  
dpkg -i ./IBM-SPSS-AnalyticServer_1_amd64.deb
```
5. Ambari サーバーを再起動します。

```
ambari-server restart
```
6. Ambari サーバーにログオンし、Ambari UI を使用して Analytic Server をサービスとしてインストールします。

外部管理される MySQL 環境に対する Analytic Server のインストール

外部管理される MySQL 環境に対してインストールする場合、Analytic Server のインストール・プロセスは、通常のインストールとは異なります。

以下のステップでは、外部管理される MySQL 環境に対して Analytic Server をインストールするプロセスについて説明します。

1. IBM パスポート・アドバンテージ Web サイトに移動し、ご使用のスタック、スタック・バージョン、およびハードウェア・アーキテクチャーに固有の自己解凍型バイナリー・ファイルを Ambari クラスタ内のホストにダウンロードします。
2. 自己解凍型バイナリー・ファイルを実行し、指示に従って (オプションで) ライセンスを表示し、ライセンスを受け入れます。
 - a. オンライン・オプションを選択します。
 - b. プロンプトが出されたら、「外部 MySQL データベース (External MySQL Database)」オプションを選択します。
3. `add_mysql_user.sh` スクリプトを、`/opt/AS_Installable/IBM-SPSS-AnalyticServer` から、`AS_MetaStore` として使用される MySQL インスタンスがインストールされているノード/ホストにコピーします。例: `/opt/AS_InstallTools`
 - MySQL ノード/ホスト上で `add_mysql_user.sh` スクリプトを実行します。例:

```
./add_mysql_user.sh -u as_user -p spss -d aedb
```

Notes:

- ユーザー名およびパスワードは、Ambari 構成画面の AS_Metastore で入力されたデータベース・ユーザー名およびパスワードと一致する必要があります。
 - コマンドを発行するように `add_mysql_user.sh` スクリプトを手動で更新できます (希望する場合)。
 - セキュアな (root ユーザーによってアクセスされる) MySQL データベースに対して `add_mysql_user.sh` スクリプトを実行する場合は、`-r` パラメーターおよび `-t` パラメーターを使用して、`dbuserid` および `dbuserid_password` を渡します。スクリプトは、`dbuserid` および `dbuserid_password` を使用して、MySQL 操作を実行します。
4. Ambari サーバーを再起動します。
 5. Ambari コンソールから、AnalyticServer サービスを通常として追加します (ステップ 3 で入力したのと同じデータベース・ユーザー名およびパスワードを入力します)。

注: 「AS_Configuration」画面の `metadata.repository.url` 設定 (「Advanced analytics-meta」) が MySQL データベース・ホストを指すように変更する必要があります。例えば、JDBC 設定 `mysql://{analytic_metastore_host}/aedb?createDatabaseIfNotExist=true` を `mysql://{MySQL_DB}/aedb?createDatabaseIfNotExist=true` に変更します。

構成

インストール後、オプションで Ambari UI を使用して Analytic Server を構成し、管理することができます。

注: Analytic Server ファイル・パスには以下の規則が使用されます。

- `{AS_ROOT}` は、Analytic Server がデプロイされている場所を示します (例えば、`/opt/IBM/SPSS/AnalyticServer/{version}`)。
- `{AS_SERVER_ROOT}` は、構成ファイル、ログ・ファイル、およびサーバー・ファイルのロケーションを示します (例えば、`/opt/IBM/SPSS/AnalyticServer/{version}/ae_wlpserver/usr/servers/aeserver`)。
- `{AS_HOME}` は、Analytic Server がルート・フォルダーとして使用する HDFS 上の場所を示します。

セキュリティ

`security.config` パラメーターは、Analytic Server システムにプリンシパルとして追加できるユーザーとグループのレジストリーを定義します。

デフォルトでは、基本レジストリーは、単一のユーザー `admin` およびパスワード `admin` を指定して定義されています。`security.config` を編集するか、または Kerberos を構成することにより、このレジストリーを変更できます。`security.config` パラメーターは、Analytic Server サービスの「Configs」タブの「Advanced analytics.cfg」セクションにあります。

注: `security.config` パラメーターを編集してレジストリーを変更する場合、新規の各ユーザーをプリンシパルとして Analytic Server システムに追加する必要があります。テナント管理について詳しくは、「IBM SPSS Analytic Server 管理者ガイド」を参照してください。

基本レジストリーの変更

基本レジストリーを使用して、`security.config` パラメーター内にユーザーとグループのデータベースを定義できます。

デフォルトの基本レジストリーは以下のようにになっています。

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

変更後の基本レジストリーの例を以下に示します。

```
<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>
```

securityUtility ツール ({AS_ROOT}/ae_wlpserver/bin にあります) を使用してパスワードをエンコードすることで、パスワードの値を難読化できます。

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

注: securityUtility ツールについて詳しくは、http://www-01.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp_command_securityutil.html を参照してください。

注: 基本レジストリーは、サンドボックス環境では有用ですが、実稼働環境ではお勧めしません。

LDAP レジストリーの構成

LDAP レジストリーは、Active Directory や OpenLDAP などの外部 LDAP サーバーを使用してユーザーを認証できるようにします。

重要: LDAP ユーザーを Ambari 内の Analytic Server 管理者として指定する必要があります。

以下に、OpenLDAP の ldapRegistry の例を示します。

```
<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>

```

以下の例は、Active Directory を使用した Analytic Server の認証を提供します。

```

<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" />
  </customFilters>
</ldapRegistry>

```

注: 多くの場合、LDAP 構成を確認するには、サード・パーティーの LDAP ビューアー・ツールを使用すると便利です。

以下の例は、Active Directory を使用した WebSphere Liberty Profile の認証を提供します。

```

<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:

- Analytic Server での LDAP のサポートは、WebSphere Liberty によって制御されます。詳しくは、『Liberty での LDAP ユーザー・レジストリーの構成』を参照してください。
- LDAP が SSL で保護されている場合は、以下の『Analytic Server から LDAP への Secure Sockets Layer (SSL) 接続の構成』セクションの指示に従ってください。

Analytic Server から LDAP への Secure Sockets Layer (SSL) 接続の構成

1. Analytic Server マシンのそれぞれに Analytic Server ユーザーとしてログインし、SSL 証明書の共通ディレクトリーを作成します。

注: デフォルトでは、Analytic Server ユーザーは `as_user` です。Ambari コンソールの「Admin」タブの下の「**Service accounts**」を参照してください。

2. 鍵ストア・ファイルおよびトラストストア・ファイルを、すべての Analytic Server マシンの共通ディレクトリにコピーします。また、LDAP クライアントの CA 証明書をトラストストアに追加します。以下に、手順例を示します。

```
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
```

注: `JAVA_HOME` は、Analytic Server の起動に使用するのと同じ JRE です。

3. `securityUtility` ツール (`{AS_ROOT}/ae_wlpserver/bin` にあります) を使用してパスワードをエンコードすることで、パスワードの値を難読化できます。次に例を示します。

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

4. Ambari コンソールにログインし、Analytic Server の構成設定 `ssl.keystore.config` を、正しい SSL 構成設定に更新します。次に例を示します。

```
<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="/>
```

注: 鍵ファイルおよびトラストストア・ファイルについては、絶対パスを使用してください。

5. Analytic Server の構成設定 `security.config` を、正しい LDAP 構成設定に更新します。例えば、`LdapRegistry` 要素の場合、`sslEnabled` 属性を `true` に設定し、`sslRef` 属性を `defaultSSLConfig` に設定します。

Kerberos の構成

Analytic Server は、Ambari を使用した Kerberos をサポートします。

注: Kerberos シングル・サインオン (SSO) が Apache Knox と組み合わせて使用される場合、IBM SPSS Analytic Server は Kerberos SSO をサポートしません。

1. Analytic Server へのアクセス権を付与する予定のすべてのユーザーについて、Kerberos ユーザー・リポジトリ内にアカウントを作成します。

注: Analytic Server インストール済み環境で基本レジストリーを使用する場合、このレジストリーには、すべての Kerberos ユーザー・アカウントが、パスワードとして "-" を使用して、指定されていなければなりません。次に例を示します。

```
<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. 前のステップで、Analytic Server および Hadoop の各ノードで作成したそれぞれのユーザーについて、OS ユーザー・アカウントを作成します。
 - これらのユーザーの UID は、すべてのマシンで一致させてください。kinit コマンドを使用して各アカウントにログオンして、これをテストすることができます。
 - UID が、Yarn の「ジョブをサブミットするための最小ユーザー ID (Minimum user ID for submitting job)」設定に従っていることを確認してください。これは、container-executor.cfg 内の **min.user.id** パラメーターです。例えば、**min.user.id** が 1000 の場合、作成される各ユーザー・アカウントの UID は 1000 以上でなければなりません。
3. Analytic Server のすべてのプリンシパルについて、HDFS 上にユーザーのホーム・フォルダーを作成します。例えば、Analytic Server システムに testuser1 を追加した場合、HDFS 上に /user/testuser1 のようなホーム・フォルダーを作成し、testuser1 がこのフォルダーに対する読み取り権限と書き込み権限を持つようにします。
4. [オプション] HCatalog データ・ソースを使用する予定であり、Analytic Server が Hive Metastore とは別のマシンにインストールされている場合、HDFS で Hive クライアント名を使用する必要があります。
 - a. Ambari コンソールで、HDFS サービスの「Configs」タブに移動します。
 - b. **hadoop.proxyuser.hive.groups** パラメーターを編集して値 * を設定するか、すべてのユーザーが Analytic Server へのログインを許可されているグループを指定します。
 - c. **hadoop.proxyuser.hive.hosts** パラメーターを編集して値 * を設定するか、サービスとして Hive Metastore および Analytic Server の各インスタンスがインストールされているホストのリストを指定します。
 - d. HDFS サービスを再起動します。

これらのステップの実行を完了した後、Analytic Server がインストールされていると、Analytic Server がサイレントかつ自動的に Kerberos の構成を行います。

Kerberos を使用したシングル・サインオン (SSO) 用の HAProxy の構成

1. HAProxy の資料 (<http://www.haproxy.org/#docs>) に従って HAProxy を構成して開始します。
2. HAProxy ホスト用の Kerberos プリンシパル (HTTP/<proxyHostname>@<realm>) およびキータブ・ファイルを作成します。ここで、<proxyHostname> は HAProxy ホストの完全な名前、<realm> は Kerberos レalmです。
3. キータブ・ファイルを各 Analytic Server ホストに /etc/security/keytabs/spnego_proxy.service.keytab としてコピーします。
4. このファイルのアクセス許可を各 Analytic Server ホストで更新します。次に例を示します。


```

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

```
5. Amabri コンソールを開き、Analytic Server の「Custom analytics.cfg」セクションで以下のプロパティを更新します。


```

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

```
6. 構成を保存し、Amabri コンソールからすべての Analytic Server サービスを再始動します。

これで、ユーザーは Kerberos SSO を使用して Analytic Server にログインできるようになりました。

Kerberos 偽名の使用の有効化

偽名を使用すると、スレッドを所有しているプロセスのセキュリティー・コンテキストとは異なるセキュリティー・コンテキスト内で、そのスレッドを実行できます。例えば、偽名の使用は、標準 Analytic Server ユーザー (as_user) 以外のユーザーとして Hadoop ジョブを実行する手段を提供します。Kerberos 偽名の使用を有効にするには、以下を行います。

1. Kerberos が有効になっているクラスター内で実行する場合は、HDFS (または Hive サービス構成) に偽名の使用構成属性を追加します。HDFS の場合は、以下のプロパティを HDFS core-site.xml ファイルに追加する必要があります。

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

ここで、<analytic_server_service_principal_name> は、Analytic Server 構成の Analytic_Server_User フィールドで指定されているデフォルトの as_user 値です。

HDFS から Hive/HCatalog を経由してデータにアクセスする場合は、以下のプロパティも HDFS core-site.xml ファイルに追加する必要があります。

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

2. as_user 以外のユーザー名を使用するように Analytic Server が構成されている場合は、そのユーザー名を反映するようにプロパティ名を変更する必要があります (例えば、hadoop.proxyuser.xxxxx.hosts です。ここで、xxxxx は、Analytic Server 構成で指定されている構成済みのユーザー名です)。

Kerberos の無効化

1. Ambari コンソールで Kerberosを無効化します。
2. Analytic Server サービスを停止します。
3. Custom analytics.cfg から、以下のパラメーターを削除します。

```
default.security.provider
hdfs.keytab
hdfs.user
java.security.krb5.conf
as.db.connect.method
web.authentication.kerberos.keytab
web.authentication.kerberos.principal
```

4. 「Save」をクリックし、Analytic Server サービスを再起動します。

Analytic Server コンソールへの Secure Sockets Layer (SSL) 接続の有効化

デフォルトでは、Analytic Server は自己署名証明書を生成して Secure Socket Layer (SSL) を有効にします。自己署名証明書を受け入れることにより、セキュア・ポートを使用して Analytic Server コンソールにアクセスできるようになります。HTTPS によるアクセスの安全性をさらに強化するには、サード・パーティー・ベンダーの証明書をインストールする必要があります。

サード・パーティー・ベンダーの証明書をインストールするには、以下のステップを実行します。

1. サード・パーティー・ベンダーの鍵ストア証明書およびトラストストア証明書を、すべての Analytic Server ノードで、同じディレクトリーにコピーします。例えば、/home/as_user/security です。

注: Analytic Server ユーザーには、このディレクトリーの読み取りアクセス権限が必要です。

2. Ambari の「Services」タブで、Analytic Server サービスの「Configs」タブに移動します。

3. **ssl.keystore.config** パラメーターを編集します。

```
<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>"/>
```

置き換える値:

- <KEYSTORE-LOCATION> に、鍵ストアの絶対位置を指定します。例: /home/as_user/security/mykey.jks
- <TRUSTSTORE-LOCATION> に、トラストストアの絶対位置に指定します。例: /home/as_user/security/mytrust.jks
- <TYPE> に、証明書のタイプを指定します。例: JKS、PKCS12、その他。
- <PASSWORD> に、Base64 暗号化形式の暗号化パスワードを指定します。エンコードには、securityUtility を使用できます。例: /opt/ibm/spss/analyticserver/3.1/ae_wlpserver/bin/securityUtility encode <password>。

自己署名証明書を作成する場合は、securityUtility を使用できます。例: /opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/bin/securityUtility createSSLCertificate --server=myserver --password=mypassword --validity=365 --subject=CN=mycompany,O=myOrg,C=myCountry。

securityUtility およびその他の SSL 設定について詳しくは、WebSphere Liberty Profile の資料を参照してください。

4. 「**Save**」をクリックし、Analytic Server サービスを再起動します。

Essentials for R に対するサポートの有効化

Analytic Server は、R モデルのスコアリング、および R スクリプトの実行をサポートしています。

R に対するサポートを構成するには、Analytic Server が正常にインストールされた後で、以下を行います。

1. Essentials for R のサーバー環境をプロビジョンします。

RedHat Linux x86_64

以下のコマンドを実行します。

```
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

以下のコマンドを実行します。

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

```

apt-get install -y libcurl4-openssl-dev
apt-get install -y liblzma-dev
apt-get install -y libpcre3 libpcre3-dev
apt-get install -y libcurl4-openssl-dev

```

SUSE Linux

SUSE での Essentials for R のインストールには、構成された ZYPPER リポジトリでは通常は使用できない、互換性のある FORTRAN が必要です (SUSE SDK メディアからのみ使用可能です)。結果として、SUSE サーバーで Essentials for R の Ambari インストールを実行すると、FORTRAN をインストールできないため、失敗します。SUSE でプロビジョンするには、以下のステップを使用します。

- a. GCC C++ をインストールします。

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

- b. GCC FORTRAN をインストールします。必要な RPM ファイルを SUSE SDK メディアからコピーできますが、以下の順序でインストールする必要があります。

```

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. 以下のコマンドを実行して、Essentials for R ライブラリーをインストールします。

```

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-1.0.2l.tar.gz --no-check-certificate
tar xzvf openssl-1.0.2l.tar.gz
cd openssl-1.0.2l/
./config shared
make
make install
echo '/usr/local/ssl/lib' >> /etc/ld.so.conf
ldconfig
cd $R_PREFIX/r_libs
wget --no-check-certificate https://curl.haxx.se/download/curl-7.50.1.tar.gz
tar xzvf curl-7.50.1.tar.gz
cd curl-7.50.1
./configure --with-ssl
make -j3
make install
cd $R_PREFIX/r_libs
wget ftp://rpmfind.net/linux/opensuse/distribution/12.3/repo/oss/suse/x86_64/libgomp1-4.7.2_20130108-2.1.6.x86_64.rpm --no-check-certificate
rpm -ivh libgomp1-4.7.2_20130108-2.1.6.x86_64.rpm

```

2. IBM SPSS Modeler Essentials for R の RPM または DEB 用の自己解凍型アーカイブ (BIN) をダウンロードします。Essentials for R は、<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp> からダウンロードできます。ご使用のスタック、スタックのバージョン、およびハードウェア・アーキテクチャーに固有のファイルを選択します。

3. 自己解凍型バイナリー・ファイルを実行し、指示に従って (オプションで) ライセンスを表示し、ライセンスを受け入れて、オンライン・インストールまたはオフライン・インストールを選択します。

オンライン・インストール

Ambari サーバー・ホストおよびクラスター内のすべてのノードが <https://ibm-open-platform.ibm.com> にアクセス可能な場合は、オンライン・インストールを選択してください。

オフライン・インストール

ご使用の Ambari サーバー・ホストがインターネットにアクセスできない場合は、オフラインを選択します。オフライン・インストールでは必要な RPM ファイルをダウンロードするため、<https://ibm-open-platform.ibm.com> にアクセス可能なマシンで実行する必要があります。その後、RPM ファイルを Ambari サーバー・ホストにコピーできます。

- a. 必要な Essentials for R の RPM または DEB ファイルを Ambari サーバー・ホスト上の任意の場所にコピーします。必要な RPM/DEB ファイルは、ご使用のディストリビューション、バージョン、およびアーキテクチャーによって以下のように異なります。

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-3.1.1.0_3.1.1.0_amd64.deb

- b. RPM または DEB をインストールします。以下の例では、コマンドは Essentials for R を HDP 2.6 (x86_64) にインストールします。

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

以下の例では、コマンドは Essentials for R を HDP 2.4 (Ubuntu) にインストールします。

```
dpkg -i IBM-SPSS-ModelerEssentialsR-ambari-3.1.1.0_3.1.1.0_amd64.deb
```

4. Ambari サーバーを再起動します。

```
ambari-server restart
```

5. Ambari サーバーにログオンし、Ambari コンソールを使用して SPSS Essentials for R をサービスとしてインストールします。SPSS Essentials for R は、Analytic Server および Analytic Metastore がインストールされているすべてのホストにインストールする必要があります。

注: Ambari は R をインストールする前に gcc-c++ および gcc-gfortran (RHEL)、および gcc-fortran (SUSE) のインストールを試行します。これらのパッケージは、R の Ambari サービス定義で依存関係として宣言されています。R のインストールおよび実行場所となるサーバーが、gcc-c++ および gcc-[g]fortran の RPM をダウンロードするように構成されていること、あるいは、そのサーバーに GCC コンパイラーおよび FORTRAN コンパイラーがインストールされていることを確認してください。Essentials for R のインストールが失敗する場合は、Essentials for R をインストールする前にこれらのパッケージを手動でインストールしてください。

6. Analytic Server サービスをリフレッシュします。
7. 25 ページの『クライアント依存関係の更新』の手順に従って `update_clientdeps` スクリプトを実行します。
8. SPSS Modeler Server をホストするマシンに Essentials for R をインストールすることも必要です。詳しくは、SPSS Modeler の資料を参照してください。

リレーショナル・データベース・ソースの有効化

各 Analytic Server ホストの共有ディレクトリー内に JDBC ドライバーを配置すると、Analytic Server でリレーショナル・データベース・ソースを使用できます。デフォルトでは、このディレクトリーは /usr/share/jdbc です。

共有ディレクトリーを変更するには、以下のステップを実行します。

1. Ambari の「Services」タブで、Analytic Server サービスの「Configs」タブに移動します。
2. 「**Advanced analytics.cfg**」セクションを開きます。
3. **jdbc.drivers.location** で、JDBC ドライバーの共有ディレクトリーを指定します。
4. 「**Save**」をクリックします。
5. Analytic Server サービスを停止します。
6. 「**Refresh**」をクリックします。
7. Analytic Server サービスを開始します。

表 5. サポート対象データベース

データベース	サポート対象バージョン	JDBC ドライバー jar	ベンダー
Amazon Redshift	8.0.2 以降	RedshiftJDBC41-1.1.6.1006.jar 以降	Amazon
Apache Impala	2.8	hive_metastore.jar、hive_service.jar、commons-codec-1.3.jar、commons-logging-1.1.1.jar、httpclient-4.1.3.jar、httpcore-4.1.3.jar、ImpalaJDBC4.jar、log4j-1.2.14.jar、libthrift-0.9.0.jar、libfb303-0.9.0.jar、slf4j-api-1.5.11.jar、ql.jar、zookeeper-3.4.6.jar、TCLIServiceClient.jar	Apache
BigSQL	4.1.0.0 以降	db2jcc.jar	IBM
dashDB	Bluemix サービス	db2jcc.jar	IBM
Db2 for Linux、UNIX、および 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, 2.1	hive-jdbc-*.jar	Apache
MySQL	5.6, 5.7	mysql-connector-java-commercial-5.1.25-bin.jar	MySQL
Netezza	7, 6.x	nzjdbc.jar	IBM
Oracle	12c, 11g R2 (11.2)	ojdbc6.jar、ora118n.jar	Oracle
SQL Server	2014, 2012, 2008 R2	sqljdbc4.jar	Microsoft

表 5. サポート対象データベース (続き)

データベース	サポート対象バージョン	JDBC ドライバー jar	ベンダー
Teradata	15, 15.1	tdgssconfig.jar、 terajdbc4.jar	Teradata

Notes

- Analytic Server をインストールする前に Redshift データ・ソースを作成した場合、Redshift データ・ソースを使用するには以下のステップを実行する必要があります。
 1. Analytic Server コンソールで Redshift データ・ソースを開きます。
 2. Redshift データベース・データ・ソースを選択します。
 3. Redshift のサーバー・アドレスを入力します。
 4. データベース名とユーザー名を入力します。パスワードは自動的に入力されます。
 5. データベース表を選択します。
- BigSQL は、Apache Hadoop 環境の IBM SQL インターフェースです。BigSQL はリレーショナル・データベースではありませんが、Analytic Server は、JDBC を経由した BigSQL へのアクセスをサポートします (JDBC jar ファイルは、Db2 で使用される jar ファイルと同じです)。

Analytic Server での BigSQL の一般的な使用法の 1 つは、HCatalog データ・ソースを経由した BigSQL Hadoop/HBase 表へのアクセスです。

HCatalog データ・ソースの有効化

Analytic Server は、Hive/HCatalog を介して複数のデータ・ソースをサポートしています。一部のソースでは、手動での構成ステップが必要です。

1. データ・ソースを有効にするために必要な JAR ファイルを収集します。詳しくは、下のセクションを参照してください。
2. これらの JAR ファイルを、各 Analytic Server ノードの {HIVE_HOME}/auxlib ディレクトリーおよび /usr/share/hive ディレクトリーに追加します。
3. Hive Metastore サービスを再起動します。
4. Analytic Metastore サービスをリフレッシュします。
5. Analytic Server サービスの各インスタンスを再起動します。

注: Analytic Server Metastore は、Hive Metastore と同じマシンにインストールすることはできません。

NoSQL データベース

Analytic Server は、ベンダーから Hive ストレージ・ハンドラーが提供されている任意の NoSQL データベースをサポートします。

Apache HBase および Apache Accumulo のサポートを有効にするために、追加のステップは必要ありません。

その他の NoSQL データベースについては、データベース・ベンダーに連絡して、該当するストレージ・ハンドラーおよび関連する jar を取得してください。

ファイル・ベース Hive 表

Analytic Server は、組み込みまたはカスタムの Hive SerDe (serializer-deserializer) が利用可能な任意のファイル・ベース Hive 表をサポートします。

XML ファイルを処理するための Hive XML SerDe は Maven の Central Repository (<http://search.maven.org/#search%7Cga%7C1%7Cchivexmlserde>) にあります。

Apache Spark

Spark (バージョン 1.5 以降) を使用する場合は、Analytic Server インストール時に `spark.version` プロパティを手動で追加する必要があります。

1. Amabri コンソールを開き、Analytic Server の「**Advanced analytics.cfg**」セクションで以下のプロパティを追加します。
 - キー: `spark.version`
 - 値: 適切な Spark バージョン番号 (例えば、1.x、2.x、または None) を入力します。
2. 構成を保存します。

注: `Custom analytics.cfg` 設定を使用して、HCatalog が Spark を使用しないように強制できます。

1. Amabri コンソールを開き、Analytic Server の「**Custom analytics.cfg**」セクションで以下のプロパティを追加します。
 - キー: `spark.hive.compatible`
 - 値: `false`

Analytic Server で使用するポートの変更

デフォルトでは、Analytic Server はポート 9080 (HTTP 用) および 9443 (HTTPS 用) を使用します。ポートの設定を変更するには、以下のステップを実行します。

1. Ambari の「Services」タブで、Analytic Server サービスの「Configs」タブに移動します。
2. 「**Advanced analytics.cfg**」セクションを開きます。
3. 使用するポートを、`http.port` (HTTP ポート) および `https.port` (HTTPS ポート) に指定します。
4. 「**Save**」をクリックします。
5. Analytic Server サービスを再始動します。

高可用性 Analytic Server

クラスター内の複数のノードに Analytic Server をサービスとして追加することにより、高可用性構成にすることができます。

1. Ambari コンソールで、「Hosts」タブに移動します。
2. Analytic Server をまだサービスとして実行していないホストを選択します。
3. 「Summary」タブで、「**Add**」をクリックし、Analytic Serverを選択します。
4. 「追加の確認 (**Confirm Add**)」をクリックします。

複数クラスタのサポート

複数クラスタ機能は、IBM SPSS Analytic Server の高可用性機能の拡張であり、複数テナント環境での独立性を強化します。デフォルトでは、(Ambari または ClouderaManager のいずれかで) Analytic Server サービスをインストールすると、結果として、単一の Analytic Server クラスタが定義されます。

クラスタ仕様では、Analytic Server クラスタ・メンバーシップが定義されます。クラスタ仕様の変更は、(Ambari Analytic Server 構成の `analytics-cluster` フィールドで、または Cloudera Manager の `configuration/analytics-cluster.xml` ファイルを手動で編集して) XML コンテンツを使用して実行されます。複数の Analytic Server クラスタを構成する際は、それぞれの Analytic Server クラスタに独自のロード・バランサーを提供する必要があります。

複数クラスタ機能を使用することで、あるテナントに対する作業が、別のテナントのクラスタで実行されている作業にマイナスの影響を与えることがなくなります。高可能性ジョブについては、ジョブのフェイルオーバーは、タスクが開始された Analytic Server クラスタの範囲内のみで発生します。以下の例は、複数クラスタ XML 仕様を提供します。

注: クラスタ内の複数のノードに Analytic Server をサービスとして追加することにより、それを高可用性にすることができます。

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

上記の例では、2 つのロード・バランサーが必要です。一方のロード・バランサーは `cluster1` のメンバー (`one.cluster` および `two.cluster`) に要求を送信し、もう一方のロード・バランサーは `cluster2` のメンバー (`three.cluster` および `four.cluster`) に要求を送信します。

以下の例は、単一クラスタ XML 仕様 (デフォルト構成) を提供します。

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

上記の例では、構成されたクラスタ・メンバーが複数ある場合に対応するために、1 つのロード・バランサーが必要です。

注

- シングルトン・クラスタのみが、**memberName** 要素でのワイルドカードの使用をサポートしています (例えば、クラスタ・カーディナリティー = "1")。カーディナリティー要素の有効な値は、1 および 1+ です。
- **memberName** は、Analytic Server 役割の割り当て先のホスト名と同じように指定する必要があります。
- クラスタ構成の変更が適用された後は、すべてのクラスタ内のすべてのサーバーを再起動する必要があります。
- Cloudera Manager では、すべての Analytic Server ノードの `analytics-cluster.xml` ファイルを変更して維持する必要があります。すべてのノードが同じ内容を含むように維持する必要があります。

スモールデータ向けの JVM オプションの最適化

小規模な (M3R) ジョブの実行時にご使用のシステムを最適化するために、JVM プロパティを編集できます。

Ambari コンソールで、Analytic Server サービスの「Configs」タブの Advanced analytics-jvm-options セクションを参照します。以下のパラメーターを変更して、Analytic Server (Hadoop ではなく) をホストするサーバーで実行されるジョブのヒープ・サイズを設定します。これは小規模な (M3R) ジョブを実行する場合に重要です。システムを最適化するために、これらの値を調整する必要がある場合があります。

```
-Xms512M  
-Xmx2048M
```

クライアント依存関係の更新

このセクションでは、update_clientdeps スクリプトを使用して Analytic Server サービスの依存関係を更新する方法を説明します。

1. Ambari サーバー・ホストに root としてログインします。
2. ディレクトリーを /var/lib/ambari-server/resources/stacks/<stack-name>/<stack-version>/services/ANALYTICSERVER/package/scripts に変更します。例を示します。

```
cd "/var/lib/ambari-server/resources/stacks/HDP/2.4/services/ANALYTICSERVER/package/scripts"
```
3. 以下の引数を指定して、update_clientdeps スクリプトを実行します。

```
-u <ambari-user>  
    Ambari アカウント・ユーザー名。  
  
-p <ambari-password>  
    Ambari アカウント・ユーザーのパスワード。  
  
-h <ambari-host>  
    Ambari サーバーのホスト名。  
  
-x <ambari-port>  
    Ambari が listen しているポート。
```

以下の例を参照してください。

```
./update_clientdeps.sh -u admin -p admin -h host.domain -x 8080
```

4. 以下のコマンドを使用して Ambari サーバーを再始動します。

```
ambari-server restart
```

Apache Knox の構成

Apache Knox Gateway は、Apache Hadoop サービスにセキュア・アクセスの単一ポイントを提供するシステムです。このシステムにより、ユーザー (クラスター・データにアクセスし、ジョブを実行する人) およびオペレーター (アクセスを制御し、クラスターを管理する人) の両者の Hadoop セキュリティーが簡素化されます。この Gateway は、1 つ以上の Hadoop クラスターに機能やサービスを提供するサーバー (またはサーバーのクラスター) として実行されます。

注: Apache Knox が Kerberos シングル・サインオン (SSO) と組み合わせて使用される場合、IBM SPSS Analytic Server は Apache Knox をサポートしません。

Apache Knox Gateway は、Hadoop クラスター・トポロジーの詳細を効果的に非表示にし、エンタープライズ LDAP および Kerberos と統合されます。以下のセクションでは、Apache Knox および Analytic Server の必要な構成タスクについての情報を提供します。

前提条件

- Analytic Server ノードは、パスワードなしの SSH 接続を使用して Knox サーバーと接続する必要があります。パスワードなしの SSH 接続は、Analytic Server から Knox に移動します (「**Analytic Server**」 > 「**Knox**」)。
- Analytic Server は、Knox サービスがインストールされた後でインストールされる必要があります。

場合によっては、予期しない問題の結果、構成ファイルが自動的にコピーされないことがあります。このような場合には、以下の構成ファイルを手動でコピーする必要があります。

- com.ibm.spss.knox_0.6-3.1.1.0.jar: このファイルは、Analytic Server の以下の場所からコピーする必要があります。

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

Knox サーバー・ノードの以下の場所にコピーします。

```
/KnoxServicePath/ext
```

例: /usr/iop/4.1.0.0/knox/ext

- rewrite.xml および service.xml: これらのファイルは、Analytic Server の以下の場所からコピーする必要があります。

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

Knox サーバー・ノードの以下の場所にコピーします。

```
/KnoxServicePath/data/services
```

例: /usr/iop/4.1.0.0/knox/data/services

Ambari の構成

Analytic Server サービスは、Ambari ユーザー・インターフェースで構成する必要があります。

1. Ambari ユーザー・インターフェースで、「**Knox**」 > 「**Configs**」 > 「**Advanced topology**」に移動します。現在の Knox 構成設定が「**content**」ウィンドウに表示されます。
2. 以下の <service> を Knox 構成に追加します。

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

{analyticserver-host} および {analyticserver-port} は、Analytic Server の対応するサーバー名およびポート番号に置き換える必要があります。

- {analyticserver-host} URL は、Ambari ユーザー・インターフェース (「**SPSS Analytic Server**」 > 「**Summary**」 > 「**Analytic Server**」) にあります。
- {analyticserver-port} 番号は、Ambari ユーザー・インターフェース (「**SPSS Analytic Server**」 > 「**Configs**」 > 「**Advanced analytics.cfg**」 > 「**http.port**」) にあります。

注: Analytic Server が複数のノードにデプロイされていて、LoadBalancer が使用される場合、{analyticserver-host} および {analyticserver-port} は LoadBalancer の URL およびポート番号に対応する必要があります。

3. Knox サービスを再起動します。

LDAP が使用される場合、Knox のデフォルトは、指定された「Demo」LDAP になります。エンタープライズ LDAP サーバー (Microsoft LDAP や OpenLDAP など) に変更できます。

Analytic Server の構成

Analytic Server に LDAP を使用するには、Apache Knox で使用されたものと同じ LDAP サーバーを使用するように、Analytic Server を構成する必要があります。以下の Ambari 設定の <value> 項目は、対応する Knox LDAP サーバー設定を反映するように更新する必要があります。

- main.ldapRealm.userDnTemplate
- main.ldapRealm.contextFactory.url

その値は、Ambari ユーザー・インターフェース (「Knox」 > 「Configs」 > 「Advanced topology」) で使用可能です。以下に例を示します。

```
<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>
```

Knox LDAP 設定を更新した後で Knox サービスを再起動します。

重要: Analytic Server の管理者パスワードは、Knox の管理者パスワードと同じである必要があります。

Apache Knox の構成

1. Knox サーバーで、サブディレクトリー <knox_server>/data/service/analyticserver/3.1.1 を作成し、service.xml ファイルおよび rewrite.xml ファイルを新規ディレクトリーにアップロードします。これらの 2 つのファイルは、Analytic Server 上の <analytic_server>/configuration/knox/analyticserver/3.1.1 (例えば、/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/knox/analyticserver/3.1.1/*.xml) にあります。
2. <knox_server>/bin で、スクリプト ./knoxcli.sh redeploy --cluster default を実行します。
3. com.ibm.spss.knoxservice_0.6-*.jar ファイルを <knox_server>/ext にアップロードします。このファイルは、Analytic Server 上の <analytic_server>/apps/AE_BOOT.war/WEB-INF/lib/com.ibm.spss.knox_0.6-3.1.1.0.jar (例えば、/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/com.ibm.spss.knox_0.6-3.1.1.0.jar) にあります。
4. Ambari ユーザー・インターフェースで、「Knox」 > 「Configs」 > 「Advanced topology」から以下の要素を追加します。

```
<service>
  <role>ANALYTICSERVER</role>
  <url>http://{AS-Host}:{AS-port}/analyticserver</url>
</service>
```

5. Ambari ユーザー・インターフェースで、「Knox」 > 「Configs」 > 「Advanced users-ldif」からユーザーを追加または更新します (例えば、admin、qauser1、qauser2)。
6. 「Knox」 > 「Service Actions」 > 「Start Demo LDAP」から LDAP を再起動します。

7. Knox サービスを再起動します。

Hortonworks Data Platform (HDP) での Apache Knox のインストール

以下のステップでは、HDP クラスタで Apache Knox をインストールするプロセスについて簡単に説明します。

1. Knox ユーザーが HDP クラスタ上に存在するかどうかを確認します。Knox ユーザーが存在しない場合は、作成する必要があります。
2. Apache Knox をダウンロードして /home/knox の下のフォルダーに解凍します。
3. HDP で、Knox ユーザーに切り替え、knox フォルダーに移動します。Knox ユーザーは、すべての knox サブフォルダーに対する permission(RWX) を持っている必要があります。
4. Analytic Server のために Apache Knox を構成します。詳しくは、『**Apache Knox** の構成』セクションを参照してください。
 - a. {knox}/data/services の下に analyticserver/3.1.1 フォルダー階層を作成します。
 - b. rewrite.xml ファイルおよび service.xml ファイルを Analytic Server の以下の場所からコピーします。

```
/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/knox/  
analyticserver/3.1.1
```

Knox サーバー・ノードの以下の場所にコピーします。

```
{knox}/data/services/analyticserver/3.1.1
```

- c. Knox *.jar ファイルを Analytic Server ホストの以下の場所からコピーします。

```
/opt/ibm/spss/analyticserver/3.1.1/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/  
WEB-INF/lib/com.ibm.spss.knox_0.6-*.jar
```

以下の Knox ext ディレクトリーにコピーします。

```
{knox}/ext
```

- d. 以下の例と一致するように {knox}/conf/topologies 内の default.xml ファイルを更新します。

注: ファイルが存在しない場合は、作成する必要があります。

```
<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>
  <provider>
    <role>identity-assertion</role>
    <name>Default</name>
    <enabled>true</enabled>
  </provider>
  <provider>
    <role>authorization</role>
    <name>AclsAuthz</name>
    <enabled>true</enabled>
  </provider>
</gateway>

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

```

5. {knox}/bin/knoxccli.sh を実行します。
6. {knox}/bin/ldap.sh start を実行します。

注: スクリプトはポート 33389 を使用します。このポートが現在使用中でないことを確認してください。

7. {knox}/bin/gateway.sh start を実行します。

注: スクリプトはポート 8443 を使用します。このポートが現在使用中でないことを確認してください。

8. インストールを検証します。
 - a. 以下のように、Knox URL 上の Analytic Server に対して curl コマンドを実行します。

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

トラブルシューティング

問題: インストール後に Analytic Server が Knox で機能しない。

解決策: Knox を停止し、{knox}/data/deployments/* の下のすべてのファイルを削除し、Knox を再起動します。

問題: Knox を経由して Analytic Server にログインできない。

解決策: {knox}/conf/users.ldif 内のユーザーを確認します。既存のユーザーを更新するか、新規 Analytic Server ユーザーを追加します。Knox ユーザーのプリンシパルおよび資格情報は、Analytic Server ユーザーと一致する必要があります。

Apache Knox 対応の Analytic Server の URL 構造

Knox 対応の Analytic Server のユーザー・インターフェース URL は、https://{knox-host}:{knox-port}/gateway/default/analyticserver/admin です。

- https プロトコル - ユーザーは証明書を受け入れて、Web ブラウザーに進む必要があります。
- knox-host は、Knox のホストです。
- knox-port は、Knox のポート番号です。
- URI は、gateway/default/analyticserver です。

Ambari での IBM SPSS Analytic Server のマイグレーション

Analytic Server は、既存の Analytic Server インストール済み環境から新規インストール済み環境にデータおよび構成設定をマイグレーションすることができます。

Analytic Server 3.1 から 3.1.1 へのマイグレーション

Analytic Server 3.1 の既存のインストール済み環境がある場合、3.1 の構成設定を 3.1.1 インストール済み環境にマイグレーションできます。

- 5 ページの『Ambari でのインストール』の指示に従って、Analytic Server の新規バージョンをインストールします。
- 以前のインストール済み環境から新規インストール済み環境に Analytic ルートをコピーします。
 - Analytic ルートの場所が不明な場合は、`hadoop -fs ls` を実行します。Analytic ルートのパスは `/user/aeuser/analytic-root` と類似しています。ここで、`aeuser` は、Analytic ルートを所有するユーザー ID です。
 - 新規サーバーの Analytic ルートを削除します。
 - `hadoop fs -copyToLocal` および `hadoop fs -copyFromLocal` を使用して、以前のサーバーの Analytic ルートを新規サーバーの `/user/as_user/` フォルダにコピーします (所有者が `as_user` として設定されていることを確認します)。
- 新規サーバーで、Ambari/Cloudera Manager コンソールを開き、Analytic Server サービスを停止します (Ambari では、Analytic Metastore サービスが実行されていることを確認します)。
- 古いインストール済み環境から構成設定を収集します。
 - 新規インストール済み環境の `configcollector.zip` アーカイブを、古いインストール済み環境の `{AS_ROOT}¥tools` にコピーします。
 - コピーした `configcollector.zip` を解凍します。これにより、以前のインストール済み環境内に新規の `configcollector` サブディレクトリーが作成されます。
 - `{AS_ROOT}¥tools¥configcollector` 内の **configcollector** スクリプトを実行して、以前のインストール済み環境内の構成収集ツールを実行します。その結果生成された圧縮ファイル (ZIP) を、新規インストール済み環境をホストするサーバーにコピーします。

重要: 指定された **configcollector** スクリプトは、最新バージョンの Analytic Server と互換性がない場合があります。**configcollector** スクリプトに関する問題が発生した場合は、IBM 技術サポート担当員にお問い合わせください。

- Zookeeper の状態をクリアします。Zookeeper の `bin` ディレクトリー (Hortonworks 上の `/usr/hdp/current/zookeeper-client` など) で、以下のコマンドを実行します。

```
./zkCli.sh rmr /AnalyticServer
```
- migrationtool** スクリプトを実行し、構成収集ツールによって作成された圧縮ファイルのパスを引数として渡すことで、マイグレーション・ツールを実行します。次に例を示します。

```
migrationtool.sh /opt/ibm/spss/analyticserver/3.1.1/ASConfiguration_3.1.0.0.xxx.zip
```
- すべての Analytic Server ノードの `ae_wlpserver/user/servers/aeserver/configuration/config.properties` ファイルを更新します。このファイルに `as_user` のエントリーを追加します。以下に例を示します。

```
hdfs.user=as_user/host@REALM
```

host は、config.properties ファイルが存在する Analytic Server ノードのホスト名と一致する必要があります。各ノードには、異なる hdfs.user 値があります。各 host 値は、それが存在する Analytic Server ホストと一致する必要があります。

8. Ambari/Cloudera Manager コンソールで、Analytic Server サービスを開始します。

注: 既存の Analytic Server インストール済み環境で使用するよう R を構成している場合、新規 Analytic Server インストール済み環境で R を構成するステップに従います。

アンインストール

重要: Essentials for R がインストールされている場合、まず remove_R.sh スクリプトを実行する必要があります。Analytic Server をアンインストールする前に、Essentials for R のアンインストールに失敗すると、後から Essentials for R をアンインストールできなくなります。Analytic Server がアンインストールされると、remove_R.sh スクリプトは削除されます。Essentials for R の アンインストールについては、『Essentials for R のアンインストール』を参照してください。

1. Analytic Metastore ホストで、{AS_ROOT}/bin ディレクトリーにある remove_as.sh スクリプトを、以下のパラメーターを指定して実行します。

- u** 必須。Ambari サーバー管理者のユーザー ID。
- p** 必須。Ambari サーバー管理者のパスワード。
- h** 必須。Ambari サーバー・ホスト名。
- x** 必須。Ambari サーバー・ポート。
- l** オプション。セキュア・モードを有効にします。

以下に例を示します。

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

クラスター内の Ambari ホスト one.cluster から Analytic Server を削除します。

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

クラスター内の Ambari ホスト one.cluster から Analytic Server をセキュア・モードで削除します。

注: この操作により、HDFS 上の Analytic Server フォルダーが削除されます。

注: この操作では、Analytic Server に関連付けられた Db2 スキーマは一切削除されません。スキーマを手動で削除する方法については、Db2 の資料を参照してください。

Essentials for R のアンインストール

1. Essentials for R ホストで、{AS_ROOT}/bin ディレクトリーにある remove_R.sh スクリプトを、以下のパラメーターを指定して実行します。

- u** 必須。Ambari サーバー管理者のユーザー ID。
- p** 必須。Ambari サーバー管理者のパスワード。
- h** 必須。Ambari サーバー・ホスト名。
- x** 必須。Ambari サーバー・ポート。
- l** オプション。セキュア・モードを有効にします。

以下に例を示します。

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

クラスター内の Ambari ホスト `one.cluster` から Essentials for R を削除します。

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

クラスター内の Ambari ホスト `one.cluster` から Essentials for R をセキュア・モードで削除します。

2. Ambari サーバー・サービス・ディレクトリーから R サービス・ディレクトリーを削除します。例えば、HDP 2.6 の場合、ESSENTIALR ディレクトリーは `/var/lib/ambari-server/resources/stacks/HDP/2.6/services` に配置されています。
3. Ambari コンソールで、Essentials for R サービスが存在していないことを確認します。

第 3 章 Cloudera のインストールおよび構成

Cloudera の概要

Cloudera は、オープン・ソースの Apache Hadoop ディストリビューションです。Cloudera Distribution Including Apache Hadoop (CDH) は、当該テクノロジーのエンタープライズ・クラスのデプロイメントを対象としています。

Analytic Server は CDH プラットフォームで実行できます。CDH には、大規模なデータ・セット (主に MapReduce および HDFS) の確実かつスケーラブルな分散データ処理を実現する Hadoop の主要なコア要素が含まれています。また、セキュリティ、高可用性、およびハードウェアや他のソフトウェアとの統合を実現するその他のエンタープライズ向けのコンポーネントも含まれています。

Cloudera 固有の前提条件

一般的な前提条件に加えて、以下の情報を確認してください。

サービス

各 Analytic Server ホストに以下のインスタンスがインストールされていることを確認してください。

- HDFS: Gateway、DataNode または NameNode
- Hive: Gateway、Hive Metastore Server または HiveServer2
- Yarn: Gateway、ResourceManager または NodeManager

以下のインスタンスは、それらの機能が使用される場合にのみ必要です。

- Accumulo: Gateway
- HBase: Gateway、Master または RegionServer

メタデータ・リポジトリ

MySQL を Analytic Server のメタデータ・リポジトリとして使用する場合は、『Analytic Server 用の MySQL の構成』の手順に従ってください。

Analytic Server 用の MySQL の構成

Cloudera Manager で IBM SPSS Analytic Server を構成するには、MySQL サーバー・データベースをインストールして構成する必要があります。

1. MySQL データベースが格納されているノードのコマンド・ウィンドウから以下のコマンドを実行します。

```
yum install mysql-server
```

注: SuSE Linux の場合は `zypper install mysql` を使用してください。

2. 各 Cloudera クラスター・ノードのコマンド・ウィンドウから以下のコマンドを実行します。

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

注: SUSE Linux の場合は `sudo zypper install mysql-connector-java` を使用してください。

3. Analytic Server が MySQL データベースへのアクセス時に使用する Analytic Server のデータベース名、データベースのユーザー名、およびデータベースのパスワードを決定し、メモを取ります。

- 『Cloudera でのインストール』の手順に従って Analytic Server をインストールします。
- Cloudera によって管理されているいずれかのサーバーから、MySQL データベースがインストールされているノードに `/opt/cloudera/parcels/AnalyticServer/bin/add_mysql_user.sh` スクリプトをコピーします。ユーザー固有の構成に適したパラメーターを指定してそのスクリプトを実行します。以下に例を示します。

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

Notes: データベースがセキュア・モード (root ユーザー・パスワードが設定される) で実行される場合は `a -r <dbRootPassword>` パラメーターが必須になります。

root 以外のユーザー名を使用してデータベースがセキュア・モードで実行されている場合は `-r <dbUserPassword>` パラメーターおよび `-t <dbUserName>` パラメーターが必須になります。

Cloudera でのインストール

以下のステップでは、Cloudera Manager で IBM SPSS Analytic Server を手動でインストールするプロセスについて説明します。

Analytic Server 3.1.1

オンライン・インストール

- IBM パスポート・アドバンテージ Web サイトに移動し、ご使用のスタック、スタック・バージョン、およびハードウェア・アーキテクチャーに固有の自己解凍型バイナリー・ファイルを Cloudera クラスター内のホストにダウンロードします。使用可能な Cloudera バイナリーは以下のとおりです。

表 6. *Analytic Server* 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for Cloudera 5.10、5.11、および 5.12、Ubuntu、英語	spss_as-3.1.1-cdh5.10-5.12-ubun_en.bin
IBM SPSS Analytic Server 3.1.1 for Cloudera 5.10、5.11、および 5.12、Linux x86-64、英語	spss_as-3.1.1-cdh5.10-5.12-lx86-en.bin

- Cloudera の自己解凍型 `*.bin` インストーラーを Cloudera Manager マスター・クラスター・ノードで実行します。ご使用条件に同意し、デフォルトの CSD インストール・ディレクトリーを維持してインストールのプロンプトに従ってください。

注: CSD ディレクトリーをデフォルトの場所から変更した場合は、別の CSD ディレクトリーを指定する必要があります。

- インストールが完了した後に Cloudera Manager を再始動します。
- Cloudera Manager インターフェース (例えば、`http://${CM_HOST}:7180/cmf/login`) を、デフォルトのログイン資格情報 `admin/admin` を使用して開き、「リモート・パーセル・リポジトリーの URL (Remote Parcel Repository URLs)」をリフレッシュし、URL が正しいことを確認します。以下に例を示します。

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

注: 「パーセルの更新頻度 (Parcel Update Frequency) および「リモート・パーセル・リポジトリーの URL (Remote Parcel Repository URLs)」は、ユーザーの固有のニーズに合わせて更新できます。

5. Cloudera Manager がパーセル・ファイルをリフレッシュした後で (「新しいパーセルの確認 (Check for New Parcels)」をクリックすることでパーセル・ファイルを手動でリフレッシュできます)、 「AnalyticServer」パーセルの状況が「リモートで使用可能 (Available Remotely)」に設定されていることが分かります。
6. 「ダウンロード (Download)」 > 「配布 (Distribute)」 > 「アクティブ化 (Activate)」を選択します。「AnalyticServer」パーセルの状況が「配布済み、アクティブ化済み (Distributed, Activated)」に更新されます。
7. Analytic Server の MySQL を構成します。
8. Cloudera Manager で Analytic Server をサービスとして追加し、Analytic Server を配置する場所を決定します。以下の情報を「サービスの追加ウィザード (Add Service Wizard)」に指定する必要があります。
 - Analytic Server metastore ホスト名
 - Analytic Server metastore データベース名
 - Analytic Server metastore ユーザー名
 - Analytic Server metastore パスワード

「サービスの追加ウィザード (Add Service Wizard)」には、サービス作成プロセスの各フェーズにおける全体の進行状況が表示されます。また、クラスターでサービスが正常にインストールおよび構成されたときに最終確認メッセージが表示されます。

注: Analytic Server が正常にインストールされた後に、Cloudera Manager の Analytic Server サービス・ページの「アクション (Actions)」リストで「**Analytic Server Metastore の作成 (Create Analytic Server Metastore)**」をクリックしないでください。Metastore を作成すると、既存のメタデータ・リポジトリが上書きされます。

オフライン・インストール

オフライン・インストールのステップは、特定のオペレーティング・システムに適したパーセル・ファイルとメタデータをユーザーが手動でダウンロードする必要がある点を除いて、オンラインのステップと同じです。

RedHat Linux では、以下のファイルが必要です。

- AnalyticServer-3.1.1.0-el7.parcel
- AnalyticServer-3.1.1.0-el7.parcel.sha
- manifest.json

SuSE Linux では、以下のファイルが必要です。

- AnalyticServer-3.1.1.0-sles11.parcel
- AnalyticServer-3.1.1.0-sles11.parcel.sha
- manifest.json

または

- AnalyticServer-3.1.1.0-sles12.parcel
- AnalyticServer-3.1.1.0-sles12.parcel.sha

Ubuntu Linux 14.04 では、以下のファイルが必要です。

- AnalyticServer-3.1.1.0-trusty.parcel

- AnalyticServer-3.1.1.0-trusty.parcel.sha

Ubuntu Linux 16.04 では、以下のファイルが必要です。

- AnalyticServer-3.1.1.0-xenial.parcel
- AnalyticServer-3.1.1.0-xenial.parcel.sha

1. Cloudera の自己解凍型 *.bin インストーラーをダウンロードして Cloudera Manager マスター・クラスター・ノードで実行します。ご使用条件に同意し、デフォルトの CSD インストール・ディレクトリーを維持してインストールのプロンプトに従ってください。

注: CSD ディレクトリーがデフォルトの場所とは異なる場合は、別の CSD ディレクトリーを指定する必要があります。

2. 必要なパーセル・ファイルとメタデータ・ファイルを、Cloudera Manager マスター・クラスター・ノード上のローカル Cloudera repo パスにコピーします。デフォルトのパスは /opt/cloudera/parcel-repo です (このパスは Cloudera Manager ユーザー・インターフェースで構成可能です)。

Cloudera Manager が「**AnalyticServer**」パーセルをリフレッシュした後でそのパーセルが「ダウンロード済み (**downloaded**)」と表示されます。「新しいパーセルの確認 (**Check for New Parcels**)」をクリックすると強制的にリフレッシュできます。

3. 「配布 (**Distribute**)」 > 「アクティブ化 (**Activate**)」をクリックします。

「**AnalyticServer**」パーセルが「配布済み」および「アクティブ化済み」と表示されます。

Cloudera での Analytic Server 3.1.1 へのアップグレード

Analytic Server 3.1 の既存のインストール済み環境がある場合、そのインストール済み環境をインプレースでバージョン 3.1.1 にアップグレードできます。

1. Cloudera Manager で、Analytic Server サービスを停止し、削除します。
2. Cloudera Manager で、Analytic Server の以前のバージョンを非アクティブ化します。
3. Analytic Server 3.1.1 のインストールの手順については、30 ページの『Ambari での IBM SPSS Analytic Server のマイグレーション』の『オンライン』または『オフライン』のセクションを参照してください。
4. Analytic Server サービスが Cloudera Manager にインストールおよび追加された後で、「**Analytic Server** バイナリーのリフレッシュ (**Refresh Analytic Server Binaries**)」を実行します。Analytic Server 3.1.1 が使用できる状態になりました。

Cloudera の構成

インストール後、オプションで Cloudera Manager を使用して Analytic Server を構成し、管理することができます。

注: Analytic Server ファイル・パスには以下の規則が使用されます。

- {AS_ROOT} は、Analytic Server がデプロイされている場所を示します (例えば、/opt/cloudera/parcels/AnalyticServer)。
- {AS_SERVER_ROOT} は、構成ファイル、ログ・ファイル、およびサーバー・ファイルの場所を示します (例えば、/opt/cloudera/parcels/AnalyticServer/ae_wlpserver/usr/servers/aeserver)。
- {AS_HOME} は、Analytic Server がルート・フォルダーとして使用する HDFS 上の場所を示します (例えば、/user/as_user/analytic-root)。

セキュリティー

security_cfg パラメーターは、Analytic Server システムにプリンシパルとして追加できるユーザーとグループのレジストリーを定義します。

デフォルトでは、基本レジストリーは、単一のユーザー `admin` およびパスワード `admin` を指定して定義されています。**security_cfg** を編集するか、またはセキュリティー・プロバイダーとして Kerberos を構成することにより、このレジストリーを変更できます。**security_cfg** パラメーターは、Analytic Server サービスの「構成 (Configuration)」タブの「**Analytic Server Advanced Configuration Snippet**」セクションにあります。

注: **security_cfg** パラメーターを編集してレジストリーを変更する場合、新規の各ユーザーをプリンシパルとして Analytic Server システムに追加する必要があります。テナント管理について詳しくは、「*IBM SPSS Analytic Server 管理者ガイド*」を参照してください。

基本レジストリーの変更

基本レジストリーを使用して、**security_cfg** パラメーター内にユーザーとグループのデータベースを定義できます。

デフォルトの基本レジストリーは以下のようになっています。

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

変更後の基本レジストリーの例を以下に示します。

```
<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>
```

`securityUtility` ツール (`{AS_ROOT}/ae_wlpserver/bin` にあります) を使用してパスワードをエンコードすることで、パスワードの値を難読化できます。

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

注: `securityUtility` ツールについて詳しくは、http://www-01.ibm.com/support/knowledgecenter/SSD28V_8.5.5/com.ibm.websphere.wlp.core.doc/ae/rwlp_command_securityutil.html を参照してください。

注: 基本レジストリーは、サンドボックス環境では有用ですが、実稼働環境ではお勧めしません。

LDAP レジストリーの構成

LDAP レジストリーは、Active Directory や OpenLDAP などの外部 LDAP サーバーを使用してユーザーを認証できるようにします。

以下に、OpenLDAP の ldapRegistry の例を示します。

```
<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>
```

追加の構成例については、テンプレート・フォルダー {AS_ROOT}/ae_wlpserver/templates/config を参照してください。

注: Analytic Server での LDAP のサポートは、WebSphere Liberty によって制御されます。詳しくは、『Liberty での LDAP ユーザー・レジストリーの構成』を参照してください。

Analytic Server から LDAP への Secure Sockets Layer (SSL) 接続の構成

1. Analytic Server マシンのそれぞれに Analytic Server ユーザーとしてログインし、SSL 証明書の共通ディレクトリーを作成します。

注: Cloudera では、Analytic Server ユーザーは常に as_user になり、これは変更できません。

2. 鍵ストア・ファイルおよびトラストストア・ファイルを、すべての Analytic Server マシンの共通ディレクトリーにコピーします。また、LDAP クライアントの CA 証明書をトラストストアに追加します。以下に、手順例を示します。

```
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
```

注: JAVA_HOME は、Analytic Server の起動に使用するのと同じ JRE です。

3. securityUtility ツール ({AS_ROOT}/ae_wlpserver/bin にあります) を使用してパスワードをエンコードすることで、パスワードの値を難読化できます。次に例を示します。

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

4. Cloudera Manager にログインし、Analytic Server の構成設定 **ssl_cfg** を、正しい SSL 構成設定で更新します。次に例を示します。

```
<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="/>
```

注: 鍵ファイルおよびトラストストア・ファイルについては、絶対パスを使用してください。

5. Analytic Server の構成設定 **security_cfg** を、正しい LDAP 構成設定で更新します。例えば、**ldapRegistry** 要素の場合、**sslEnabled** 属性を true に設定し、**sslRef** 属性を defaultSSLConfig に設定します。

Kerberos の構成

Analytic Server は、Cloudera での Kerberos をサポートします。

1. Analytic Server へのアクセス権を付与する予定のすべてのユーザーについて、Kerberos ユーザー・リポジトリ内にアカウントを作成します。

注: Analytic Server インストール済み環境で基本レジストリーを使用する場合、このレジストリーには、すべての Kerberos ユーザー・アカウントが、パスワードとして "-" を使用して、指定されていなければなりません。次に例を示します。

```
<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. 前のステップで、Analytic Server および Hadoop の各ノードで作成したそれぞれのユーザーについて、OS ユーザー・アカウントを作成します。
 - これらのユーザーの UID は、すべてのマシンで一貫させてください。kinit コマンドを使用して各アカウントにログオンして、これをテストすることができます。
 - UID が、Yarn の「ジョブをサブミットするための最小ユーザー ID (Minimum user ID for submitting job)」設定に従っていることを確認してください。これは、container-executor.cfg 内の **min.user.id** パラメーターです。例えば、**min.user.id** が 1000 の場合、作成される各ユーザー・アカウントの UID は 1000 以上でなければなりません。
3. Analytic Server のすべてのプリンシパルについて、HDFS 上にユーザーのホーム・フォルダーを作成します。例えば、Analytic Server システムに testuser1 を追加した場合、HDFS 上に /user/testuser1 のようなホーム・フォルダーを作成し、testuser1 がこのフォルダーに対する読み取り権限と書き込み権限を持つようにします。
4. HCatalog データ・ソースを使用する予定であり、Analytic Server が Hive Metastore とは別のマシンにインストールされている場合、HDFS で Hive クライアント名を使用する必要があります。
 - a. Cloudera Manager で、HDFS サービスの「構成 (Configuration)」タブに移動します。

注: 以下のパラメーターがまだ設定されていない場合、「構成 (Configuration)」タブにそれらのパラメーターが表示されない可能性があります。その場合は、検索を実行して見つけてください。

- b. **hadoop.proxyuser.hive.groups** パラメーターを編集して値 * を設定するか、すべてのユーザーが Analytic Server へのログインを許可されているグループを指定します。
- c. **hadoop.proxyuser.hive.hosts** パラメーターを編集して値 * を設定するか、サービスとして Hive Metastore および Analytic Server の各インスタンスがインストールされているホストのリストを指定します。

- d. HDFS サービスを再起動します。

これらのステップの実行を完了した後、Analytic Server がインストールされていると、Analytic Server がサイレントかつ自動的に Kerberos の構成を行います。

Kerberos を使用したシングル・サインオン (SSO) 用の HAProxy の構成

1. HAProxy の資料 (<http://www.haproxy.org/#docs>) に従って HAProxy を構成して開始します。
2. HAProxy ホスト用の Kerberos プリンシパル (HTTP/<proxyHostname>@<realm>) およびキータブ・ファイルを作成します。ここで、<proxyHostname> は HAProxy ホストの完全な名前、<realm> は Kerberos レalmです。
3. キータブ・ファイルを各 Analytic Server ホストに /etc/security/keytabs/spnego_proxy.service.keytab としてコピーします。
4. このファイルのアクセス許可を各 Analytic Server ホストで更新します。次に例を示します。

```
chown root:hadoop /etc/security/keytabs/spnego_proxy.service.keytab
chmod 440 /etc/security/keytabs/spnego_proxy.service.keytab
```
5. Cloudera Manager を開き、Analytic Server の「**Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties**」領域で以下のプロパティを追加するか、更新します。

```
web.authentication.kerberos.keytab=/etc/security/keytabs/spnego_proxy.service.keytab
web.authentication.kerberos.principal=HTTP/<proxy machine full name>@<realm>
```
6. 構成を保存し、Cloudera Manager からすべての Analytic Server サービスを再起動します。
7. Kerberos を使用するようにブラウザを構成することをユーザーに指示します。

これで、ユーザーは Kerberos SSO を使用して Analytic Server にログインできるようになりました。

Kerberos 偽名の使用の有効化

偽名を使用すると、スレッドを所有しているプロセスのセキュリティー・コンテキストとは異なるセキュリティー・コンテキスト内で、そのスレッドを実行できます。例えば、偽名の使用は、標準 Analytic Server ユーザー (as_user) 以外のユーザーとして Hadoop ジョブを実行する手段を提供します。Kerberos 偽名の使用を有効にするには、以下を行います。

1. Cloudera Manager を開き、Analytic Server の「**Analytic Server Advanced Configuration Snippet (Safety Valve) for core-site.xml**」領域 (「**HDFS (サービス全体) (HDFS (Service-Wide))**」 > 「**構成 (Configuration)**」タブにあります) で以下のプロパティを追加するか、更新します。

```
- hadoop.proxyuser.as_user.hosts = *
- hadoop.proxyuser.as_user.groups = *
```
2. as_user 以外のユーザー名を使用するように Analytic Server が構成されている場合は、そのユーザー名を反映するようにプロパティ名を変更する必要があります (例えば、hadoop.proxyuser.xxxxx.hosts です。ここで、xxxxx は、Analytic Server 構成で指定されている構成済みのユーザー名です)。

注: プロパティは、(Analytic Server 構成内の値に基づいて) Ambari に自動的に追加されます。

Kerberos の無効化

1. Ambari コンソールで Kerberosを無効化します。
2. Analytic Server サービスを停止します。
3. 「**Analytic Server Advanced Configuration Snippet (Safety Valve) for analyticserver-conf/config.properties**」領域から以下のパラメーターを削除します。

```
default.security.provider
hdfs.keytab
hdfs.user
java.security.krb5.conf
as.db.connect.method
web.authentication.kerberos.keytab
web.authentication.kerberos.principal
```

4. 「変更の保存 (Save Changes)」をクリックし、Analytic Server サービスを再始動します。

Analytic Server コンソールへの Secure Sockets Layer (SSL) 接続の有効化

デフォルトでは、Analytic Server は自己署名証明書を生成して Secure Socket Layer (SSL) を有効にします。自己署名証明書を受け入れることにより、セキュア・ポートを使用して Analytic Server コンソールにアクセスできるようになります。HTTPS によるアクセスの安全性をさらに強化するには、サード・パーティー・ベンダーの証明書をインストールする必要があります。

サード・パーティー・ベンダーの証明書をインストールするには、以下のステップを実行します。

1. サード・パーティー・ベンダーの鍵ストア証明書およびトラストストア証明書を、すべての Analytic Server ノードで、同じディレクトリにコピーします。例えば、/home/as_user/security です。

注: Analytic Server ユーザーには、このディレクトリの読み取りアクセス権限が必要です。

2. Cloudera Manager で、Analytic Server サービスの「構成 (Configuration)」タブに移動します。
3. **ssl_cfg** パラメーターを編集します。

```
<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>"/>
```

置き換える値:

- <KEYSTORE-LOCATION> に、鍵ストアの絶対位置を指定します。例: /home/as_user/security/mykey.jks
- <TRUSTSTORE-LOCATION> に、トラストストアの絶対位置に指定します。例: /home/as_user/security/mytrust.jks
- <TYPE> に、証明書のタイプを指定します。例: JKS、PKCS12、その他。
- <PASSWORD> に、Base64 暗号化形式の暗号化パスワードを指定します。エンコードには、securityUtility を使用できます。例: {AS_ROOT}/ae_wlpserver/bin/securityUtility encode <password>

自己署名証明書を生成する場合は、securityUtility を使用できます。例: {AS_ROOT}/ae_wlpserver/bin/securityUtility createSSLCertificate --server=myserver --password=myspassword --validity=365 --subject=CN=mycompany,O=myOrg,C=myCountry。securityUtility およびその他の SSL 設定について詳しくは、WebSphere Liberty Profile の資料を参照してください。

4. 「変更の保存 (Save Changes)」をクリックし、Analytic Server サービスを再始動します。

Essentials for R に対するサポートの有効化

Analytic Server は、R モデルのスコアリング、および R スクリプトの実行をサポートしています。

Cloudera Manager で Analytic Server が正常にインストールされた後で Essentials for R をインストールするには、以下を行います。

- Essentials for R のサーバー環境をプロビジョンします。詳しくは、18 ページの『Essentials for R に対するサポートの有効化』のステップ 1 を参照してください。
- IBM SPSS Modeler Essentials for R の RPM 用の自己解凍型アーカイブ (BIN) をダウンロードします。Essentials for R は、<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp> からダウンロードできます。ご使用のスタック、スタックのバージョン、およびハードウェア・アーキテクチャーに固有のファイルを選択します。
- Cloudera Manager サーバー・ホストで root ユーザーまたは sudo ユーザーとして自己解凍型アーカイブを実行します。以下のパッケージがインストール済みであるか、構成済みのリポジトリから使用可能である必要があります。
 - Red Hat Linux: gcc-gfortran、zip、gcc-c++
 - SUSE Linux: gcc-fortran、zip、gcc-c++
 - Ubuntu Linux: gcc-fortran、zip、gcc-c++
- この自己解凍型インストーラーは、以下のタスクを行います。
 - 必要なライセンスを表示し、それに同意するようにインストーラーでプロンプトを出します。
 - R のソースの場所を入力するか、またはデフォルトの場所で続行するようにインストーラーでプロンプトを出します。インストールされているデフォルトの R バージョンは 3.3.2 です。別のバージョンをインストールするには、以下を行います。
 - オンライン・インストール: 必要な R バージョン・アーカイブの URL を指定します。例えば、R 2.15.3 の場合は <https://cran.r-project.org/src/base/R-2/R-2.15.3.tar.gz> です。
 - オフライン・インストール: 必要な R バージョン・アーカイブをダウンロードし、Cloudera Manager サーバー・ホストにコピーします。アーカイブの名前を変更しないでください (デフォルトの名前は R-x.x.x.tar.gz です)。コピーした R アーカイブの URL を `file://<R_archive_directory>/R-x.x.x.tar.gz` のように指定します。R-2.15.3.tar.gz アーカイブをダウンロードして /root にコピーした場合、その URL は `file:///root/R-2.15.3.tar.gz` になります。

注: その他の R バージョンは <https://cran.r-project.org/src/base/> にあります。

 - R が必要とするパッケージをインストールします。
 - R および Essentials for R プラグインをダウンロードしてインストールします。
 - パーセルおよび `parcel.sha` ファイルを作成し、それらを `/opt/cloudera/parcel-repo` にコピーします。この場所が変更されている場合は、正しい場所を入力します。
- インストールが完了したら、Cloudera Manager で「Essentials for R」パーセルを配布およびアクティブ化します (「新しいパーセルの確認 (Check for New Parcels)」をクリックしてパーセル・リストをリフレッシュします)。
- Analytic Server サービスが既にインストールされている場合は、以下を行います。
 - サービスを停止します。
 - Analytic Server バイナリーをリフレッシュします。
 - サービスを開始して Essentials for R のインストールを完了します。
- Analytic Server サービスがインストールされていない場合は、そのインストールを進めます。

注: すべての Analytic Server ホストに適切なアーカイブ (zip および unzip) パッケージがインストールされている必要があります。

リレーショナル・データベース・ソースの有効化

各 Analytic Server ホストの共有ディレクトリー内に JDBC ドライバーを配置すると、Analytic Server でリレーショナル・データベース・ソースを使用できます。デフォルトでは、このディレクトリーは /usr/share/jdbc です。

共有ディレクトリーを変更するには、以下のステップを実行します。

1. Cloudera Manager で、Analytic Server サービスの「構成 (Configuration)」タブに移動します。
2. **jdbc.drivers.location** で、JDBC ドライバーの共有ディレクトリーを指定します。
3. 「変更の保存 (Save Changes)」をクリックします。
4. 「アクション (Action)」ドロップダウンから「停止 (Stop)」を選択して Analytic Server サービスを停止します。
5. 「アクション (Action)」ドロップダウンから「Analytic Server バイナリーのリフレッシュ (Refresh Analytic Server Binaries)」を選択します。
6. 「アクション (Action)」ドロップダウンから「開始 (Start)」を選択して Analytic Server サービスを開始します。

表 7. サポート対象データベース

データベース	サポート対象バージョン	JDBC ドライバー jar	ベンダー
Amazon Redshift	8.0.2 以降	RedshiftJDBC41-1.1.6.1006.jar 以降	Amazon
Apache Impala	2.8	hive_metastore.jar、hive_service.jar、commons-codec-1.3.jar、commons-logging-1.1.1.jar、httpClient-4.1.3.jar、httpcore-4.1.3.jar、ImpalaJDBC4.jar、log4j-1.2.14.jar、libthrift-0.9.0.jar、libfb303-0.9.0.jar、slf4j-api-1.5.11.jar、ql.jar、zookeeper-3.4.6.jar、TCLIServiceClient.jar	Apache
dashDB	Bluemix サービス	db2jcc.jar	IBM
Db2 for Linux、UNIX、および Windows	11.1, 10.5, 10.1, 9.7	db2jcc.jar	IBM
Db2 z/OS	11, 10	db2jcc.jar、db2_license_cisuz.jar	IBM
Greenplum	5.x	postgresql.jar	Greenplum
Hive	1.1, 1.2, 2.1	hive-jdbc-*.jar	Apache
MySQL	5.6, 5.7	mysql-connector-java-commercial-5.1.25-bin.jar	MySQL

表 7. サポート対象データベース (続き)

データベース	サポート対象バージョン	JDBC ドライバー jar	ベンダー
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

- Analytic Server をインストールする前に Redshift データ・ソースを作成した場合、Redshift データ・ソースを使用するには以下のステップを実行する必要があります。
 1. Analytic Server コンソールで Redshift データ・ソースを開きます。
 2. Redshift データベース・データ・ソースを選択します。
 3. Redshift のサーバー・アドレスを入力します。
 4. データベース名とユーザー名を入力します。パスワードは自動的に入力されます。
 5. データベース表を選択します。

HCatalog データ・ソースの有効化

Analytic Server は、Hive/HCatalog を介して複数のデータ・ソースをサポートしています。一部のソースでは、手動での構成ステップが必要です。

1. データ・ソースを有効にするために必要な JAR ファイルを収集します。詳しくは、下のセクションを参照してください。
2. これらの JAR ファイルを、各 Analytic Server ノードの {HIVE_HOME}/auxlib ディレクトリーおよび /usr/share/hive ディレクトリーに追加します。
3. Hive Metastore サービスを再起動します。
4. Analytic Server サービスの各インスタンスをすべて再起動します。

NoSQL データベース

Analytic Server は、ベンダーから Hive ストレージ・ハンドラーが提供されている任意の NoSQL データベースをサポートします。

Apache HBase および Apache Accumulo のサポートを有効にするために、追加のステップは必要ありません。

その他の NoSQL データベースについては、データベース・ベンダーに連絡して、該当するストレージ・ハンドラーおよび関連する jar を取得してください。

ファイル・ベース Hive 表

Analytic Server は、組み込みまたはカスタムの Hive SerDe (serializer-deserializer) が利用可能な任意のファイル・ベース Hive 表をサポートします。

XML ファイルを処理するための Hive XML SerDe は Maven の Central Repository (<http://search.maven.org/#search%7Cga%7C1%7Chivexmlserde>) にあります。

Apache Spark

Spark (バージョン 1.5 以降) を使用する場合は、Analytic Server インストール時に `spark.version` を選択する必要があります。

1. Cloudera Manager を開き、「**Analytic Server Spark Version**」領域で適切な `spark.version` (None、1.x、2.x など) を選択します。
2. 構成を保存します。

Apache Impala の構成

Apache Impala は、(Impala が SSL 対応であるかどうかにかかわらず) Analytic Server データベース・データ・ソースまたは HCatalog データ・ソースに対して Cloudera 上で実行する場合にサポートされます。

Apache Impala データのデータベース・データ・ソースの作成

1. メインの Analytic Server 「データ・ソース」ページで、「新規」をクリックして新規データ・ソースを作成します。「新規データ・ソース (New data source)」ダイアログが表示されます。
2. 「新規データ・ソース (New data source)」フィールドに適切な名前を入力し、「データベース」を「内容タイプ (Content type)」値として選択して、「Ok」をクリックします。
3. 「データベース選択 (Database Selections)」セクションを開き、以下の情報を入力します。

データベース：

ドロップダウン・メニューから「**Impala**」を選択します。

サーバー・アドレス (Server address):

Impala デモンをホストするサーバーの URL を入力します。Analytic Server に対して Kerberos が有効になっている場合、完全修飾ドメイン名が必要です。

サーバー・ポート:

Impala データベースが listen するポートの番号を入力します。

データベース名:

接続先データベースの名前を入力します。

ユーザー名:

Impala データベースにログインする権限を持つユーザー名を入力します。

パスワード:

適切なユーザー名のパスワードを入力します。

表名: 使用するデータベースの表の名前を入力します。「選択」をクリックしてファイルを手動で選択します。

最大同時読み取り数 (Maximum concurrent reads):

データ・ソースで指定された表からデータを読み込むために、Analytic Server からデータベースに送信することができる同時クエリー数の制限を入力します。

4. 必要な情報を入力した後、「保存」をクリックします。

Apache Impala データの HCatalog データ・ソースの作成

1. メインの Analytic Server 「データ・ソース」ページで、「新規」をクリックして新規データ・ソースを作成します。「新規データ・ソース (New data source)」ダイアログが表示されます。
2. 「新規データ・ソース (New data source)」フィールドに適切な名前を入力し、「HCatalog」を「内容タイプ (Content type)」値として選択して、「Ok」をクリックします。

3. 「データベース選択 (**Database Selections**)」セクションを開き、以下の情報を入力します。

データベース:

ドロップダウン・メニューから「デフォルト」を選択します。

表名: 使用するデータベースの表の名前を入力します。

HCatalog スキーマ

「**HCatalog** の要素 (**HCatalog Element**)」オプションを選択し、適切な「**HCatalog** のフィールドのマッピング (**HCatalog Field Mappings**)」オプションを選択します。

4. 必要な情報を入力した後、「保存」をクリックします。

Apache Impala 有効データへの接続

1. Analytic Server コンソールで以下の Impala SSL 設定を定義します。

Impala の TLS/SSL を有効にする (**client_services_ssl_enabled**)

「**Impala** (サービス全体) (**Impala (Service-Wide)**)」オプションを選択します。

Impala TLS/SSL サーバー証明書ファイル (PEM 形式) (Impala TLS/SSL Server Certificate File (PEM Format)) (ssl_server_certificate)

PEM 形式の自己署名証明書の場所とファイル名を入力します (例: /tmp/<user_name>/ssl/114200v21.crt)。

Impala TLS/SSL サーバー秘密鍵ファイル (PEM 形式) (Impala TLS/SSL Server Private Key File (PEM Format)) (ssl_private_key)

PEM 形式の秘密鍵の場所とファイル名を入力します (例: /tmp/<user_name>/ssl/114200v21.key)。

2. Analytic Server ホストで、*.crf ファイル (Impala SSL を有効にするために使用される) を *.jks ファイルにインポートします。このファイルは、cacerts ファイル (/etc/pki/java/cacerts など) であっても他の *.jks ファイルであっても構いません。
3. Analytic Server ホストで、以下の jdbcurl キー値を付加することにより、Impala 構成ファイル (impala.properties) を更新します。

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

注: *.jks ファイル (cacerts 以外) を使用した場合、以下を指定する必要もあります。

```
SSLTrustStore=<your_pks_file>;SSLTrustStorePwd=<password_for_pks_file>;
```

4. Cloudera Manager コンソールで Analytic Server を再始動します。

Analytic Server で使用するポートの変更

デフォルトでは、Analytic Server はポート 9080 (HTTP 用) および 9443 (HTTPS 用) を使用します。ポートの設定を変更するには、以下のステップを実行します。

1. Cloudera Manager で、Analytic Server サービスの「構成 (Configuration)」タブに移動します。
2. 使用する HTTP ポートおよび HTTPS ポートを、それぞれ **http.port** パラメーターおよび **https.port** パラメーターに指定します。

注: これらのパラメーターを表示するには、「フィルター (Filters)」セクションで「ポートおよびアドレス (Ports and Addresses)」カテゴリを選択する必要がある場合があります。

3. 「変更の保存 (**Save Changes**)」をクリックします。
4. Analytic Server サービスを再始動します。

高可用性 Analytic Server

クラスター内の複数のノードに Analytic Server をサービスとして追加することにより、高可用性構成にすることができます。

1. Cloudera Manager で、Analytic Server サービスの「インスタンス (Instances)」タブに移動します。
2. 「役割インスタンスの追加 (Add Role Instances)」をクリックし、Analytic Server をサービスとして追加するホストを選択します。

複数クラスターのサポート

複数クラスター機能は、IBM SPSS Analytic Server の高可用性機能の拡張であり、複数テナント環境での独立性を強化します。デフォルトでは、(Ambari または ClouderaManager のいずれかで) Analytic Server サービスをインストールすると、結果として、単一の Analytic Server クラスターが定義されます。

クラスター仕様では、Analytic Server クラスター・メンバーシップが定義されます。クラスター仕様の変更は、(Ambari Analytic Server 構成の `analytics-cluster` フィールドで、または Cloudera Manager の `configuration/analytics-cluster.xml` ファイルを手動で編集して) XML コンテンツを使用して実行されます。複数の Analytic Server クラスターを構成する際は、それぞれの Analytic Server クラスターに独自のロード・バランサーを提供する必要があります。

複数クラスター機能を使用することで、あるテナントに対する作業が、別のテナントのクラスターで実行されている作業にマイナスの影響を与えることがなくなります。高可能性ジョブについては、ジョブのフェイルオーバーは、タスクが開始された Analytic Server クラスターの範囲内のみで発生します。以下の例は、複数クラスター XML 仕様を提供します。

注: クラスター内の複数のノードに Analytic Server をサービスとして追加することにより、それを高可用性にすることができます。

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

上記の例では、2 つのロード・バランサーが必要です。一方のロード・バランサーは `cluster1` のメンバー (`one.cluster` および `two.cluster`) に要求を送信し、もう一方のロード・バランサーは `cluster2` のメンバー (`three.cluster` および `four.cluster`) に要求を送信します。

以下の例は、単一クラスター XML 仕様 (デフォルト構成) を提供します。

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

上記の例では、構成されたクラスター・メンバーが複数ある場合に対応するために、1 つのロード・バランサーが必要です。

注

- シングルトン・クラスターのみが、**memberName** 要素でのワイルドカードの使用をサポートしています (例えば、クラスター・カーディナリティー = "1"). カーディナリティー要素の有効な値は、1 および 1+ です。
- **memberName** は、Analytic Server 役割の割り当て先のホスト名と同じように指定する必要があります。
- クラスター構成の変更が適用された後は、すべてのクラスター内のすべてのサーバーを再起動する必要があります。
- Cloudera Manager では、すべての Analytic Server ノードの `analytics-cluster.xml` ファイルを変更して維持する必要があります。すべてのノードが同じ内容を含むように維持する必要があります。

スモールデータ向けの JVM オプションの最適化

小規模な (M3R) ジョブの実行時にご使用のシステムを最適化するために、JVM プロパティを編集できます。

Cloudera Manager で、Analytic Server サービスの「構成 (Configuration)」タブの「**Jvm オプション (jvm.options)**」コントロールを確認します。以下のパラメーターを変更して、Analytic Server (Hadoop ではなく) をホストするサーバーで実行されるジョブのヒープ・サイズを設定します。これは小規模な (M3R) ジョブを実行する場合に重要です。システムを最適化するために、これらの値を調整する必要がある場合があります。

```
-Xms512M  
-Xmx2048M
```

マイグレーション

Analytic Server では、既存の Analytic Server インストール済み環境から新規インストール済み環境へのデータおよび構成設定のマイグレーションが可能です。

Analytic Server の新規バージョンへのアップグレード

Analytic Server 3.1 の既存のインストール済み環境がある状態で新規バージョンを購入した場合、3.1 の構成設定を新規インストール済み環境にマイグレーションできます。

制約事項: 3.1 インストール済み環境と新規インストール済み環境は、同じ Hadoop クラスター内には共存できません。3.1 インストール済み環境と同じ Hadoop クラスターを使用するように新規インストール済み環境を構成すると、3.1 インストール済み環境は動作しなくなります。

3.1 から新規バージョンへのマイグレーション・ステップ

1. 34 ページの『Cloudera でのインストール』 の手順に従って、Analytic Server の新規インストールを実行します。
2. 以前のインストール済み環境から新規インストール済み環境に Analytic ルートをコピーします。
 - a. Analytic のルートの場合が不明な場合は、`hadoop -fs ls` を実行します。Analytic のルートのパスの形式は `/user/aeuser/analytic-root` です。ここで、`aeuser` は、Analytic ルートを所有しているユーザー ID です。
 - b. Analytic ルートの所有権を `aeuser` から `as_user` に変更します。

```
hadoop dfs -chown -R {as_user:{group}} {path to 3.1 analytic-root}
```

注: マイグレーション後に既存の Analytic Server インストール済み環境を使用する予定の場合は、`analytic-root` ディレクトリーのコピーを HDFS 内に作成してから、そのディレクトリーのコピーで所有権を変更します。

- c. Analytic Server の新規インストール済み環境のホストに `as_user` としてログインします。 `/user/as_user/analytic-root` ディレクトリーが存在する場合は削除します。
- d. 以下のコピー・スクリプトを実行します。

```
hadoop distcp hftp://{host of 3.1 namenode}:50070/{path to 3.1 analytic-root}
hdfs://{host of 3.1 namenode}/user/as_user/analytic-root
```

3. Cloudera Manager で、Analytic Server サービスを停止します。
4. 古いインストール済み環境から構成設定を収集します。
 - a. 新規インストール済み環境の `configcollector.zip` アーカイブを、古いインストール済み環境の `{AS_ROOT}¥tools` にコピーします。
 - b. コピーした `configcollector.zip` を解凍します。これにより、古いインストール済み環境内に新規の `configcollector` サブディレクトリーが作成されます。
 - c. `{AS_ROOT}¥tools¥configcollector` 内の **configcollector** スクリプトを実行して、古いインストール済み環境内の構成収集ツールを実行します。その結果生成された圧縮ファイル (ZIP) を、新規インストール済み環境をホストするサーバーにコピーします。

重要: 指定された **configcollector** スクリプトは、最新バージョンの Analytic Server と互換性がない場合があります。**configcollector** スクリプトに関する問題が発生した場合は、IBM 技術サポート担当員にお問い合わせください。

5. Zookeeper の状態をクリアします。Zookeeper の `bin` ディレクトリー (例えば、Cloudera 上の `/opt/cloudera/parcels/CDH-5.4...../lib/zookeeper/bin`) で、以下のコマンドを実行します。

```
./zkCli.sh rmr /AnalyticServer
```

6. **migrationtool** スクリプトを実行し、構成収集ツールによって作成された圧縮ファイルのパスを引数として渡すことで、マイグレーション・ツールを実行します。次に例を示します。

```
migrationtool.sh /opt/ibm/spss/analyticserver/3.1.1/ASConfiguration_3.1.0.0.xxx.zip
```

7. Cloudera Manager で、Analytic Server サービスを開始します。

注: 既存の Analytic Server インストール済み環境で使用するように R を構成している場合、新規 Analytic Server インストール済み環境で R を構成するステップに従う必要があります。

Cloudera での Analytic Server のアンインストール

Cloudera は、Analytic Server のサービスおよびパーセルをアンインストールするために必要なステップの大部分を自動的に処理します。

Cloudera 環境から Analytic Server をクリーンアップするには、以下のステップを実行する必要があります。

1. Analytic Server サービスを停止してから削除します。
2. Analytic Server パーセルの非アクティブ化、ホストからの削除、および削除を行います。
3. HDFS の Analytic Server ユーザー・ディレクトリーを削除します。デフォルトの場所は `/user/as_user/analytic-root` です。
4. Analytic Server が使用するデータベース (スキーマ) を削除します。

第 4 章 MapR のインストールおよび構成

MapR の概要

MapR は Apache Hadoop 向けの完全なディストリビューションであり、Hadoop エコシステムから 10 以上のプロジェクトがパッケージ化され、ビッグデータの処理に幅広く対応しています。

サーバー・クラスターの外部から MapR ファイル・システムにアクセスすることはできません。結果として、IBM SPSS Analytic Server は MapR クラスター・ノードにデプロイする必要があります。このデプロイメント・シナリオでは、Analytic Server は、MapR ファイル・システムへのアクセス権を持ち、かつ Analytic Server にデプロイするために YARN にジョブをサブミットする権限を持つユーザーが (<as_user> として) 実行する必要があります。

MapR での Analytic Server のインストール

以下のステップでは、IBM SPSS Analytic Server を手動で MapR クラスターにインストールするプロセスについて詳しく説明します。

MapR 5.1 での Analytic Server 3.1.1 のインストール

1. IBM パスポート・アドバンテージ Web サイトに移動し、MapR 自己解凍型バイナリー・ファイルをダウンロードします。

表 8. MapR 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for MapR 5.1、Linux x86-64、英語	spss_as-3.1.1-mapr5.1-5.2-lx86_en.bin

2. root ユーザーまたは sudo ユーザーで Analytic Server インストーラーを実行します。インストールのプロンプトに従い、ライセンスに同意し、Analytic Server をオンラインでインストールするかオフラインでインストールするかを選択します。
 - a. Analytic Server をホストするサーバーに <https://ibm-open-platform.ibm.com> へのインターネット接続がある場合はオンライン・オプションを選択します。インストーラーが自動的に Analytic Server をインストールします。
 - b. Analytic Server をホストするサーバーに <https://ibm-open-platform.ibm.com> へのインターネット接続がない場合はオフライン・オプションを選択します。その URL にアクセス可能な別のサーバーでインストーラーを実行し、Analytic Server をオフラインでインストールすることを選択します。インストーラーが自動的に RPM または DEB パッケージをダウンロードします。
3. 以下のように、Analytic Server の RPM または DEB を検索して実行します。
 - RedHat または SUSE Linux:

```
rpm -ivh IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64.rpm
```
 - Ubuntu Linux:

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

オンラインおよびオフラインのいずれのインストール・モードでも、Analytic Server は /opt/ibm/spss/analyticserver/3.1.1 に (<as_installation_path> として) インストールされます。

4. インストール・パス内のすべてのファイルを、以下のように、Analytic Server を実行するユーザーに変更します。

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

ユーザーを <as_user> に切り替えます。以降のすべてのステップでは <as_user> を使用します。

5. HTTP プロパティを構成します。http_endpoint.xml という名前のファイルをパス <as_installation_path>/ae_wlpserver/usr/servers/aeserver に作成し、そのファイルに以下の行を追加します。

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

<http_port> と <https_port> は、HTTP プロトコルと HTTPS プロトコルを介して Analytic Server が使用するポートです。使用可能な任意のポートでそれらを置き換えます。

6. ユーザーおよびグループを追加します。security_cfg.xml という名前のファイルをパス <as_installation_path>/ae_wlpserver/usr/servers/aeserver に作成し、そのファイルに以下の行を追加します。

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

デフォルトの状態では、XML ファイルには admin ユーザーのみが含まれています。他のユーザーおよびグループを手動で <basicRegistry> 設定に追加するか、その設定を ldapRegistry に変更する必要があります。

7. メタデータ・データベースをセットアップします。Analytic Server は、Db2 データベースおよび MySQL データベースをサポートします。
 - a. データベース・ユーザーを構成します。MySQL データベースを使用する場合は、MySQL シェルで以下の SQL スクリプトを実行します。

```
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. パスワードを暗号化します。データベース・ユーザーのパスワードは、Analytic Server に渡す前に暗号化する必要があります。以下のコマンドを実行します。

```
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>
```

注: このコマンドを直接 Linux シェルで実行する場合は、文字 * を ¥* としてエスケープする必要が生じる可能性があります。

このコマンドの出力は、The encrypted password is '<encrypted_db_password>' と表示されます。暗号化されたデータベース・パスワードを記録してください。

- c. <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties ファイルが存在する場合はそのファイルを削除し、同じ名前で新規ファイルを作成します。Db2 データベースを使用する場合は、以下のプロパティを変更します。

```
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>
```


<db_schema_name> スキーマが存在しない場合、ユーザー <db_username> には、スキーマを作成するための暗黙的な権限が付与されている必要があります。MySQL データベースを使用する場合は、以下のプロパティーを変更します。

```
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. MySQL データベースを使用する場合は、MySQL JDBC ドライバーをインストールする必要があります。以下のコマンドを実行します。

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

- e. 以下のコマンドを実行して、必要な表を作成します。

```
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
```

<db_type> は、どちらのデータベースを使用するかに応じて、db2 または mysql のいずれかです。

注: MYISAM エンジンと共に MySQL を使用する場合、2 番目のコマンドで以下のエラー・メッセージが報告されますが、無視しても問題ありません。

```
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. 以下のコマンドを実行して、cf ライブラリーをアンパックします。

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

9. private_library.xml という名前のファイルをパス <as_installation_path>/ae_wlpserver/usr/servers/aeserver に作成することにより JAAS ログイン・モジュールのクラスパスを構成し、そのファイルに以下の情報を入力します。

```
<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>
```

注: 上記の例は、hadoop_simple login モジュールを構成するためのものです。MapR で他のログイン・モジュールを使用する場合は、この構成を変更する必要があります。

10. ファイル ASModules.xml がパス <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/ に存在することを確認します。そのファイルが存在しない場合は、ファイル ASModules.xml.template (同じパス内) を ASModules.xml に名前変更します。

11. 以下のプロパティを `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルに追加することにより、クラスター情報を構成します。

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

`ae.cluster.zookeeper.connect.string` プロパティは、コンマ区切りの `zookeeper` ノード・リストです。このプロパティは、MapR が使用する `zookeeper` クラスターを共有できます。
`ae.cluster.member.name` は、Analytic Server をホストするノードのホスト名です。

以下の例は、`ae.cluster.zookeeper.connect.string` 形式を示しています。

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

Analytic Server と MapR が同じ `zookeeper` クラスターを共有している場合、
`ae.cluster.zookeeper.connect.string` 値は、MapR の `warden.conf` ファイル (このファイルのデフォルトの場所は `/opt/mapr/conf` です) 内の `zookeeper.servers` プロパティと同じである必要があります。

12. `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` ファイルを開き、そのファイルに以下の行を追加します。

```
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
```

`<as_installation_path>` および `<java_home>` を、実際のインストール・パスおよび Java ホーム・パスで置き換えます。

13. `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルを開き、以下の行を追加することにより、Analytic ルートを編集します。

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

`<analytic_root>` は、必須の Analytic Server リモート・ファイルをホストする MapR ファイル・システムのパスです。推奨パスは `/user/<as_user>/analytic-root` です。

14. `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルを開き、以下の行を追加することにより、管理者ユーザーを設定します。

```
admin.username=admin
```

この値は、Analytic Server 管理者ユーザー名にする必要があります、かつ `security_cfg.xml` ファイルで構成されたいずれかのユーザーでなければなりません。

15. 以下の行を `<as_installation_path>/bin/hdfsUpdate.sh` ファイルの行 69 に追加することにより、Analytic Server の依存関係を MapR ファイル・システムにアップロードします。

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

以下のコマンドを実行して、`<analytic_root>` を作成します。

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

`<as_user>` には、`<analytic_root>` 親ディレクトリーに対する書き込み権限が付与されている必要があります。

16. Analytic Server を開始してから停止します。
 - a. 以下のコマンドを実行し、Analytic Server を開始します。

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

- b. 以下のコマンドを実行し、Analytic Server を停止します。

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

MapR 5.2 での Analytic Server 3.1.1 のインストール

1. IBM パスポート・アドバンテージ Web サイトに移動し、MapR 自己解凍型バイナリー・ファイルをダウンロードします。

表 9. MapR 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for MapR 5.2、Linux x86-64、英語	spss_as-3.1.1-mapr5.1-5.2-lx86_en.bin

2. Analytic Server のインストールの残りのステップは、MapR 5.1 での Analytic Server 3.1.1 のインストールのステップとほとんど同じです。ただし、57 ページの『Apache HBase の有効化』および 58 ページの『Apache Spark の有効化』の情報は、MapR 5.1 と 5.2 で異なります。MapR 5.2 でのインストールについては、それらのトピックを参照してください。

MapR の構成

インストール後、オプションで Analytic Server MapR の機能を構成し、管理することができます。

データベース・プッシュバックの有効化

データベース・プッシュバックとは、データベースからデータを読み取り、直接そのデータで処理を行う操作です。

IBM SPSS Analytic Server は、以下のデータベースのプッシュバックをサポートします。

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

データベース・プッシュバックを有効にするには、以下のステップを使用します。

1. 適切な JDBC ドライバー JAR ファイルを <as_installation_path>/jdbc にコピーします。
2. <as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml ファイルを開き、ID が maprLib のタグ・ライブラリーを見つけ、以下の行をタグに追加します。

```
<fileset dir="<as_installation_path>/jdbc" includes="*.jar"/>
```

3. 以下のコマンドを実行します。

```
cd <as_installation_path>/jdbc
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

4. Analytic Server を再始動します。

Apache Hive の有効化

Apache Hive は、データの要約、照会、および分析を提供するために Hadoop 上に構築されたデータウェアハウス・インフラストラクチャーです。

注: Hive は、metastore として MySQL を使用するように構成する必要があります。IBM SPSS Analytic Server をホストするノードに存在する hive-site.xml ファイルは、Hive Metastore を実行するノードのファイルと同じでなければなりません。

MapR が正常にインストールされた後で Apache Hive のサポートを有効にするには、以下を行います。

1. 以下のコマンドを実行して、Hive および hcatalog の依存関係を MapR ファイル・システムにアップロードします。

```
cd /opt/mapr/hive/hive-2.1/lib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
cd /opt/mapr/hive/hive-2.1/hcatalog/share/hcatalog
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

<as_analytic_root> は、51 ページの『MapR での Analytic Server のインストール』で定義した Analytic のルート・パスです。

2. <as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml ファイルを開き、ID が maprLib のタグ・ライブラリーを見つけ、以下の行をタグに追加します。

```
<fileset dir="/opt/mapr/hive/hive-2.1/lib" includes="*.jar"/>
<fileset dir="/opt/mapr/hive/hive-2.1/hcatalog/share/hcatalog" includes="*.jar"/>
```

3. 以下のコマンドを実行して、Hive および hcatalog の構成ファイル・リンクを作成します。

```
mkdir <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hive-conf
ln -s /opt/mapr/hive/hive-2.1/conf/* <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/hive-conf
```

4. Hive の auxlib に追加の JAR ファイルがある場合は、以下の行を private_library.xml ファイルに追加します。

```
<fileset dir="/opt/mapr/hive/hive-2.1/auxlib" includes="*.jar"/>
```

前述の行を追加した後に、以下のコマンドを実行します。

```
cd /opt/mapr/hive/hive-2.1/auxlib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

5. Analytic Server を再始動します。

HTTP モードでの Hive の実行

デフォルトでは、Hive はバイナリー・モード (TCP モード) で実行されます。Hive を HTTP モードで実行するには、以下の Hive 構成プロパティ (特に hive.server2.transport.mode プロパティ) を更新する必要があります。

注: 各プロパティについて詳しくは、『Hive Configuration Properties』を参照してください。

表 10. HTTP モードの Hive プロパティ

プロパティ名	デフォルト値	説明
hive.server2.transport.mode	binary	サーバー・トランスポート・モード。値は binary または http にすることができます。HTTP トランスポート・モードを有効にするには、http に設定します。

表 10. HTTP モードの Hive プロパティ (続き)

プロパティ名	デフォルト値	説明
hive.server2.thrift.http.port	10001	HTTP モードの場合のポート番号。
hive.server2.thrift.http.path	cliservice	HTTP モードの場合の URL エンドポイントのパス・コンポーネント。
hive.server2.thrift.http.min.worker.threads	5	HTTP モードの場合のサーバー・プール内のワーカー・スレッドの最小数。
hive.server2.thrift.http.max.worker.threads	500	HTTP モードの場合のサーバー・プール内のワーカー・スレッドの最大数。

注: プロパティが更新された後で、Hive を再始動する必要があります。

Apache HBase の有効化

Apache HBase は、Java で記述されたオープン・ソースの非リレーショナル分散データベースです。これは Apache Software Foundation の Apache Hadoop プロジェクトの一環として開発され、HDFS (Hadoop Distributed Filesystem) 上で実行されます。

MapR が正常にインストールされた後で Apache HBase のサポートを有効にするには、以下を行います。

MapR 5.1 での IBM SPSS Analytic Server 3.1.1 のインストール

1. HBase の依存関係を MapR ファイル・システムにアップロードします。以下のコマンドを実行します。

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

<as_analytic_root> は、51 ページの『MapR での Analytic Server のインストール』で定義した Analytic のルート・パスです。

2. <as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml ファイルを開き、ID が maprLib のタグ・ライブラリーを見つけ、以下の行をタグに追加します。

```
<fileset dir="/opt/mapr/hbase/hbase-0.98.12/lib" includes="*.jar"/>
```

3. 以下のコマンドを実行して、HBase および hcatalog の構成ファイル・リンクを作成します。

```
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. IBM SPSS Analytic Server を再始動します。

MapR 5.2 での IBM SPSS Analytic Server 3.1.1 のインストール

1. 以下のコマンドを実行して、HBase の依存関係を MapR ファイル・システムにアップロードします。

```
cd /opt/mapr/hbase/hbase-1.1.1/lib
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

<as_analytic_root> は、51 ページの『MapR での Analytic Server のインストール』のステップ 12 で設定されたパスです。

2. <as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml を開き、ID が maprLib のタグ・ライブラリーを見つけてみます。以下の行をタグに追加します。

```
<fileset dir="/opt/mapr/hbase/hbase-1.1.1/lib" includes="*.jar"/>
```

3. 以下のコマンドを実行して、Hive および HCatalog の構成ファイルのリンクを作成します。

```
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
```

- 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` に追加します。

```
spark.executor.extraClassPath=/opt/mapr/hbase/hbase-1.1.1/lib/*
```

- Analytic Server を再始動します。

Apache Spark の有効化

Apache Spark は、バッチ処理、リアルタイム処理、および高度な分析を、柔軟にメモリー内でデータ処理するオープン・スタンダードです。

MapR が正常にインストールされた後で Apache Spark のサポートを有効にするには、以下を行います。

MapR 5.1 での IBM SPSS Analytic Server 3.1.1 のインストール

- `spark-assembly-1.4.1-hadoop2.5.1-mapr-1501.jar` ファイルを `/opt/mapr/spark/spark-1.4.1/lib` から `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/` にコピーします。
- Spark の依存関係を MapR ファイル・システムにアップロードします。以下のコマンドを実行します。

```
cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

`<as_analytic_root>` は、51 ページの『MapR での Analytic Server のインストール』で定義した Analytic のルート・パスです。

- `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/private_library.xml` ファイルを開き、ID が `maprLib` のタグ・ライブラリーを見つけ、以下の行をタグに追加します。

```
<fileset dir="/opt/mapr/spark/spark-1.4.1/lib" includes="spark-assembly-*.jar"/>
```

- 以下のコマンドを実行して、Spark の構成ファイル・リンクを作成します。

```
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
```

- 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` ファイルに追加します。

```
SPARK_HOME=/opt/mapr/spark/spark-1.4.1
```

- 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルに追加します。

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

- IBM SPSS Analytic Server を再始動します。
- PySpark 機能を有効にするには、以下の行を `yarn-env.sh` ファイルに追加し、ResourceManager および NodeManager を再始動します。

```
export SPARK_HOME=/opt/mapr/spark/spark-1.4.1
```

MapR 5.2 での IBM SPSS Analytic Server 3.1.1 のインストール

ステップは、Spark バージョンによって異なります。

Spark 1.x

- ファイル `spark-assembly-1.4.1-hadoop2.5.1-mapr-1501.jar` を以下の場所からコピーします。

```
/opt/mapr/spark/spark-1.4.1/lib
```

以下の場所にコピーします。

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

2. 以下のファイルを開きます。

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

ID が `maprLib` のタグ・ライブラリーを見つけます。以下の行をタグに追加します。

```
<fileset dir="/opt/mapr/spark/spark-1.4.1/lib" includes="spark-assembly-*.jar"/>
```

注: 新規に追加する行は、その他の `maprLib` 子タグより上位に配置する必要があります。

3. 以下のファイルを削除します。

```
<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/com.ibm.spss.sparkmapreduce_2-3.1.0.0.jar
```

Spark 2.x

1. 以下のファイルを削除します。

```
<as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/com.ibm.spss.sparkmapreduce-3.1.0.0.jar
```

2. 以下のファイルを開きます。

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

ID が `maprLib` のタグ・ライブラリーを見つけます。以下の行をタグに追加します。

```
<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"/>
```

注: 新規に追加する行は、他の `maprLib` 子タグより上位に配置する必要があります。

3. 行 `spark.version=2.0` を以下のファイルに追加します。

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

以下のステップは、Spark 1.x と 2.x の両方に共通であり、上記の Spark 1.x または 2.x のステップの後で実行する必要があります。

注: 参照変数 `<spark_version>` はすべて、実際の `spark` バージョン (例えば、1.4.1 または 2.0.1) に置き換える必要があります。

1. 以下のコマンドを実行して、Spark の依存関係を MapR ファイル・システムにアップロードします。

```
cd <as_installation_path>/ae_wlpserver/usr/servers/aeserver/modules/spark/  
hadoop fs -put *.jar <as_analytic_root>/cluster1/classpath
```

`<as_analytic_root>` は、51 ページの『MapR での Analytic Server のインストール』のステップ 12 で設定されたパスです。

2. 以下のコマンドを実行して、Spark の構成ファイルのリンクを作成します。

```
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. 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` ファイルに追加します。

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

4. 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルに追加します。

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

5. Analytic Server を再始動します。

6. PySpark 機能を有効にする必要がある場合は、以下の行を `yarn-env.sh` ファイルに追加します。

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

ResourceManager および NodeManager を再始動します。

機能フラグの有効化

機能フラグを使用すると、特定のアプリケーションの機能を有効にしたり無効にしたりすることができます。

MapR が正常にインストールされた後で機能フラグのサポートを有効にするには、以下を行います。

1. 以下の行を <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties ファイルに追加します。

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

2. IBM SPSS Analytic Server を再始動します。

R の有効化

R は、統計的な計算およびグラフィックスのための言語であり、環境です。

MapR が正常にインストールされた後で R のサポートを有効にするには、以下を行います。

注: Node Manager および IBM SPSS Analytic Server をホストするすべてのクラスター・ノードでインストーラーを実行するには、事前に以下のパッケージをインストールする必要があります。

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

1. Essentials for R のサーバー環境をプロビジョンします。詳しくは、18 ページの『Essentials for R に対するサポートの有効化』のステップ 1 を参照してください。
2. Node Manager および Analytic Server をホストするすべてのクラスター・ノードでインストーラー spss_er-8.4.0.0-mapr5-1x86_64_en.bin を実行します。このインストーラーを実行するユーザーには、R および Analytic Server インストール・パスに対する書き込み権限が付与されている必要があります。
3. ご使用条件に同意してインストール手順に従い、必要な情報を入力してください。Analytic Server がインストール・サーバーにインストールされている場合は、プロンプトが出されたら Yes を選択し、<as_installation_path> を入力します。Analytic Server がインストール・サーバーにインストールされていない場合は、プロンプトが出されたら No を選択します。
4. Analytic Server がインストールされている場合、Essentials for R は自動的に Analytic Server インストール・パスにインストールされます。

- Analytic Server がインストールされていない場合、Essentials for R はパス <installer_path>/IBM_SPSS_ModelerEssentialsR/linux にインストールされます。
- Analytic Server を後でインストールした場合は、以下のコマンドを使用して、Analytic Server がインストールされている Analytic Server 構成パスに Essentials for R をコピーします。

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

5. パス <as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration の cf.zip ファイルを削除し、以下のコマンドを使用して新規ファイルを生成します。

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

6. 以下のコマンドを実行します。

```
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. Analytic Server を再始動します。

LZO の有効化

LZO は、無損失データ圧縮ライブラリーであり、圧縮率よりも速度を優先します。LZO のサポートを提供するには MapR を手動で構成する必要があります。

LZO のインストールと構成の手順はこのサイト (<https://github.com/twitter/hadoop-lzo>) に記載されています。

以下のステップでは、LZO ライブラリーを MapR にインポートするためのプロセスについて詳しく説明します。

1. `hadoop-lzo-<version>.jar` ファイルを Hadoop クラスパスにコピーします。推奨パスは `/opt/mapr/hadoop/hadoop-2.7.0/share/hadoop/mapreduce/lib` です。
2. ネイティブ・ファイルの `libgplcompression.so` および `liblzo2.so.2` を `/opt/mapr/hadoop/hadoop-2.7.0/lib/native` にコピーし、以下のプロパティーを `core-site.xml` ファイルに追加します。

```
<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. `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/server.env` ファイルを開き、`<lzo_native_path>` を `LD_LIBRARY_PATH` パラメーターに追加します。 `<lzo_native_path>` は、Hadoop-LZO ネイティブ・ライブラリーが含まれるフォルダーです。

```
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. IBM SPSS Analytic Server を再始動します。

MapR の SLM タグの有効化

SLM タグは、Resource Utilization Measurement の ISO/IEC 19770-4 規格のドラフトに基づいています。SLM タグは、製品がライセンス・メトリック (ソフトウェア資産の使用に関連するリソース) の使用量を報告するための標準化機能を提供します。製品の SLM を有効にすると、そのライセンス使用を自己報告するためのランタイム XML ファイルが生成されます。

MapR で SLM タグ を有効にするには、`<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration` フォルダーに `SlmTagOutput.properties` ファイルを作成する必要があります。ファイルには、以下の内容が含まれている必要があります。

```
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
```

MapR 用の IBM SPSS Analytic Server クラスターのセットアップ

MapR のサポート用に IBM SPSS Analytic Server のクラスター環境をセットアップするには、以下のステップを使用します。

1. 以下の行を `<as_installation_path>/ae_wlpserver/usr/servers/aeserver/configuration/config.properties` ファイルに追加します。

```
enable.resume=true
```

2. このインストール・パスを他のクラスター・ノードにコピーし、`config.properties` ファイルの `ae.cluster.member.name` プロパティを正しいホスト名に変更します。
3. すべてのクラスター・ノードを開始します。

MapR のアンインストール

以下のステップでは、MapR をアンインストールするプロセスについて説明します。

1. IBM SPSS Analytic Server を停止します。
2. メタデータ・データベースを削除します。
 - a. 以下のコマンドを実行します。

```
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. 以下の SQL ステートメントを実行して、データベースを除去します。

```
drop database <db_name>
```

3. 次のように、RPM パッケージをアンインストールします。

```
rpm -e IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64
```

4. 次のように、インストール・パスを削除します。

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

5. 次のように、Analytic ルートを削除します。

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

6. 次のように、zookeeper のデータを削除します。

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

MapR での IBM SPSS Analytic Server のマイグレーション

IBM SPSS Analytic Server は MapR でマイグレーションできます。

MapR で IBM SPSS Analytic Server 3.1 からバージョン 3.1.1 へのマイグレーションを行うには、以下のステップを使用します。

1. 51 ページの『MapR での Analytic Server のインストール』のインストール手順に従って、Analytic Server 3.1.1 を MapR クラスターにインストールします。
2. analytic ルートをコピーします。

注: Analytic ルートが変更されていない場合、このステップは省略可能です。

- Analytic Server バージョン 3.1 と 3.1.1 の両方の Analytic ルートが同じ MapR クラスターにある場合は、いずれかのデータ・ノードで以下のコマンドを実行します。

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

- Analytic Server バージョン 3.1 と 3.1.1 の Analytic ルートが別々の MapR クラスターにある場合、インストールされた WEBHDFS サービスまたは NFS サービスによって処理が決まります。WEBHDFS または NFS では、Analytic ルートのデータをコピーする必要があります。これは、クラスターの外部から直接 MapR ファイル・システムにアクセスできないためです。

- a. 以前の Analytic Server 3.1 クラスターに WEBHDFS サービスが含まれている場合、新規の Analytic Server 3.1.1 クラスター・ノードのいずれかで以下のコマンドを実行します。

```
hadoop distcp webhdfs://<webhdfs_server>:<webhdfs_port>/<old_analytic_root>/analytic-workspace/*
maprfs://<new_analytic_root>/analytic-workspace
```

- b. 新規の Analytic Server 3.1.1 クラスターに WEBHDFS サービスが含まれている場合、以前の Analytic Server 3.1 クラスター・ノードのいずれかで以下のコマンドを実行します。

```
hadoop distcp maprfs://<old_analytic_root>/analytic-workspace/*  
webhdfs://<webhdfs_server>:<webhdfs_port>/<new_analytic_root>/analytic-workspace
```

- c. 以前のクラスターに NFS が含まれており、かつ NFS が新規の Analytic Server 3.1.1 クラスター・ノードのいずれかにマウントされている場合、以前の Analytic Server 3.1 クラスター・ノードのいずれかで以下のコマンドを実行します。

```
hadoop distcp file:///<mount_path>/<old_analytic_root>/analytic-workspace/* maprfs://<new_analytic_root>/analytic-workspace
```

- d. 新規のクラスターに NFS が含まれており、かつ NFS が以前の Analytic Server 3.1 クラスター・ノードのいずれかにマウントされている場合、新規の Analytic Server 3.1.1 クラスター・ノードのいずれかで以下のコマンドを実行します。

```
hadoop discp maprfs://<old_analytic_root>/analytic-workspace/* file:///<mount_path>/<new_analytic_root>/analytic-workspace
```

異なる MapR クラスター間でデータのマイグレーションを行う方法については、MapR の「Data Migration」サイトを参照してください。

3. 以下のコマンドを実行して、新規の Analytic ルートの所有者とアクセス権を変更します。

```
hadoop fs -chown -R <as_user> <analytic_root>  
hadoop fs -chmod -R 755 <>
```

4. Analytic Server 3.1.1 を停止します。ただし、メタデータ・データベースは引き続き実行中であることを確認してください。

5. 以前の Analytic Server 3.1 クラスター・インストール済み環境から構成設定を収集します。

- a. 新規の Analytic Server 3.1.1 クラスター・インストール済み環境から以前の Analytic Server 3.1 クラスター・インストール済み環境上の `<old_as_installation_path>/tools` に `configcollector.zip` アーカイブをコピーします。
- b. 以前の Analytic Server 3.1 クラスター・インストール済み環境で `configcollector.zip` の内容を解凍します。以前の Analytic Server 3.1 クラスター・インストール済み環境に新規の `configcollector` サブディレクトリーが作成されます。
- c. `<old_as_installation_path>/tools/configcollector` から `configcollector` スクリプトを実行して、以前の Analytic Server 3.1 クラスター・インストール済み環境で構成収集ツールを実行します。結果として生成される圧縮 (ZIP) ファイルを新規の Analytic Server 3.1.1 クラスター・インストール済み環境にコピーします。

重要: 指定された `configcollector` スクリプトは、最新バージョンの Analytic Server と互換性がない場合があります。`configcollector` スクリプトに関する問題が発生した場合は、IBM 技術サポート担当員にお問い合わせください。

6. `migrationtool` スクリプトを実行し、構成収集ツールによって作成された圧縮ファイルのパスを引数として渡すことで、新規の Analytic Server 3.1.1 クラスターでマイグレーション・ツールを実行します。以下に例を示します。

```
migrationtool.sh /opt/ibm/spss/analyticserver/3.1.1/ASConfiguration_3.1.0.0.xxx.zip
```

7. Analytic Server 3.1.1 を開始します。

MapR のトラブルシューティング

このセクションでは、MapR のインストールおよび構成の一般的な問題とその解決方法を説明します。

hdfsUpdate.sh スクリプトに関する問題

hdfsUpdate.sh スクリプトは、新規ファイルをアップロードする前に analytic-root 内のすべてのファイルを削除するため、このスクリプトの実行は 1 回のみにする必要があります。このスクリプトを 2 回以上実行した場合、データベース・プッシュバック、Hive、HBase、および Spark の依存関係を再アップロードする必要があります。必要な依存関係を再アップロードするには、以下のコマンドを実行します。

```
cd <as_installation_path>/jdbc

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

cd /opt/mapr/hive/hive-2.1/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
```

MapR と Spark の間のバージョン競合の結果、Spark ジョブ実行が失敗する

MapR バージョンが 5.1 以降である場合は、MapR と Spark (1.6.1) の間でクラス競合問題が発生します。競合の結果、Spark ジョブ実行が失敗します。<as_installation_path>/ae_wlpserver/usr/servers/aeserver 内の private_library.xml ファイルを変更することにより、この問題を解決できます。以下の例は、必要な変更を示しています。

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

第 5 章 Huawei FusionInsight HD のインストールおよび構成

FusionInsight HD の概要

Huawei FusionInsight HD は、オープン・ソースの Hadoop と Spark のテクノロジーを使用するバッチおよびリアルタイムの分析のための包括的なビッグデータ・ソフトウェア・プラットフォームを提供します。このシステムは、高速リアルタイム分析と対話式照会のための Apache Spark とともに、Hadoop クラスタリング用の HDFS、HBase、MapReduce、および YARN/Zookeeper を使用します。

Analytic Server は FusionInsight HD プラットフォームで実行できます。FusionInsight HD には、大規模なデータ・セット (主に MapReduce および HDFS) の確実かつスケーラブルな分散データ処理を実現する Hadoop の主要なコア要素が含まれています。また、セキュリティ、高可用性、およびハードウェアや他のソフトウェアとの統合を実現するその他のエンタープライズ向けのコンポーネントも含まれています。

Huawei FusionInsight HD でのインストール

以下のステップでは、Huawei FusionInsight HD で IBM SPSS Analytic Server を手動でインストールするプロセスについて説明します。

Analytic Server 3.1.1

1. [IBM パスポート・アドバンテージ Web サイト](#)に移動し、以下の自己解凍型バイナリー・ファイルを FusionInsight HD クラスタ内のホストにダウンロードします。

表 11. Analytic Server 自己解凍型バイナリー・ファイル

説明	バイナリー・ファイル名
IBM SPSS Analytic Server 3.1.1 for FusionInsight HD 2.6、Linux x86-64、英語	spss_as-3.1.1-fhd2.6-1x86_en.bin

2. 自己解凍型 *.bin インストーラーを FusionInsight Manager マスター・クラスター・ノードで実行します。ご使用条件に同意し、デフォルトのインストール・ディレクトリーを維持してインストールのプロンプトに従ってください。インストーラーは、必要な RPM ファイルをダウンロードするため、<https://ibm-open-platform.ibm.com> にアクセス可能なコンピューターで実行する必要があります。実行可能バイナリー・ファイルは、使用可能な FusionInsight HD <AS_INSTALLABLE_HOME> ディストリビューション・ディレクトリーにあります。

3. Analytic Server 3.1.1 をインストールするには、以下のコマンドを使用します。

```
# yum install -y IBM-SPSS-AnalyticServer-3.1.1.0-1.x86_64.rpm
```

4. 以下のようにして、omm でログインして analyticserver.keytab を作成します。

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

デフォルトの kadmin パスワードは Admin@123 です。このパスワードは、初めて使用するときに変更する必要があります。以下のコマンドの _HOST を、ご使用のホスト名で置き換えます。

```
kadmin > addprinc -randkey omm/_HOST@HADOOP.COM
kadmin > ktadd -k /opt/ibm/spss/analyticserver/3.1.1/analyticserver.keytab HTTP/_HOST@HADOOP.COM
kadmin > ktadd -k /opt/ibm/AnalyticServer/analyticserver.keytab omm/_HOST@HADOOP.COM
```

5. MYSQL をインストールし、手動で aedb を作成します。以下に例を示します。

```
# 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
```

以下のようにして、MYSQL root ユーザーのパスワードを取得します。

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

6. IBM スキーマを作成します。

```
# /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/sql/mysql
  java -Xmx128m -Xms128m -cp <as_installation_path>/ae_wlpserver/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib/*:/usr/share/
  java/* com.spss.ae.dbscript.ScriptRunner ../../configuration/config.properties schema.sql true
```

7. ファイル所有者を omm に設定します。

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

8. FusionInsight 管理者ユーザー・インターフェースから HDFS クライアントおよび Spark クライアントをダウンロードします。例えば、Analytic Server /tmp/FusionInsight-Client フォルダーにクライアントをダウンロードし、ダウンロードした *.tar ファイルを解凍します。

9. その *.tar ファイルを解凍した後に、HDFS クライアントをインストールします。

10. Analytic Server で、/opt/ibm/spss/analyticsserver/3.1.1 内に hadoop フォルダーを作成し、Hadoop *.jar ファイルを hadoop フォルダーにコピーします。Hadoop *.jar ファイルは、ステップ 8 で *.tar ファイルを解凍したときに作成した FusionInsight_V100R002C60U20_Spark_ClientConfig/Spark/FusionInsight-Spark-1.5.1/lib フォルダー内にあります。

11. Analytic Server で、/opt/ibm/spss/analyticsserver/3.1.1 内に zookeeper フォルダーを作成し、zookeeper-3.5.1.jar を zookeeper フォルダーにコピーします。

12. cf.zip の内容を /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration フォルダーに解凍します。

13. ネイティブ Hadoop ファイルを /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64 フォルダーにコピーします。

ネイティブ Hadoop ファイルは、ステップ 8 で解凍した HDFS クライアント *.tar の内容にあります。

14. 対応するコード例に記載されているとおりに、以下の構成ファイルを変更します。

- /opt/ibm/spss/analyticsserver/3.1.1/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/aedb
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=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAGk3b1uya2BzXyEYFc0rxo=
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/3.1.1/analyticsserver.keytab
hdfs.keytab=/opt/ibm/spss/analyticsserver/3.1.1/analyticsserver.keytab
ae.db.connect.method=Basic
kdcrealm=HADOOP.COM
kdcserver=172.16.155.212:21732
encryption.keystore.password=FEFFUy9FQ0IvUEtDUzVQYWRkaW5nAMDJul7PVsvdIyLlZjeS8ws=
encryption.keystore.base64=zs70zgAAAAIAAAABAAAAAwA6Y29tLnNwc3MuYUWUzW5jcn1wdG1vbi5wcm92aWR1ci51bmNyeXB0aW9ucHJvdm1kZXJpbXBs
LmF1c2AAUUtG2Ahyr00ABXNyAB1qYXZheC5jcn1wdG8uU2VhbGVkT2JqZWNoY29pY3p0Y3VHACAARbAA11bmNvZGVkUGFyYW1zdAACW0JbABB1bmNyeXB0ZW50c25
OZW50cQBAAAFMAA1wYXJhbXB0ABJmamF2YS9sYW5nL1N0cm1uZzttMAAdzWfSQWxncQB+AAJ4cHYAAJbQqzF/gGCFtGAgAAeHAAAAAPMA0EEnr6ybtX0
1mAgEudXEafgAEAAAAcGbnRpiJe0xkAuiMpwPjhzFuWCD20eK7YZ4pwutRbgEcx4u13SfPDAQcMZDTH+Ze03p8p1m7Kb/yY7SK6xvaaFYvCC9IWNUG6pkz/FXsw
nVgb1G/Jsve7mYEX+8R2FUC+t2CEuzioKdTCiUzsnzz0x00AAANQkV0ABZQkVXaXRoTUQ1QW5kVHJpcGx1REVtqmaA1K/MuEHB/yIaqSe9NgA2JsY=
jdbc.drivers.location=/usr/share/jdbc
default.security.provider=WebSphere
load.feature.flags.on.msg=true
spark.serializer=org.apache.spark.serializer.JavaSerializer
spark.executor.extraLibraryPath=/opt/ibm/spss/analyticsserver/3.1.1/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/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/jaas.conf
zookeeper.server.jaas.conf=/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/jaas.conf
krb5.conf=/home/omm/kerberos/var/krb5kdc/krb5.conf

```

- /opt/ibm/spss/analyticsserver/3.1.1/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. hadoop-conf フォルダを /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration ディレクトリー内に作成します。

Spark *.xml ファイルを hadoop-conf フォルダにコピーします。Spark *.xml ファイルは、ステップ 8 で解凍した Spark クライアント *.tar の内容にあります。

16. /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/server.env ファイルを以下のように変更します (パス値を実際のサーバー・パスと一致するように変更する必要があります)。

```

JAVA_HOME=/opt/huawei/Bigdata/jdk/jre
PATH=/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/Bigdata/jdk/jre/
lib/amd64:/usr/sbin:/usr/bin:/sbin:/bin
IBM_SPSS_AS_NATIVE_PATH=/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64
LD_LIBRARY_PATH=/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/linux/lib_64:/opt/huawei/
Bigdata/jdk/jre/lib/amd64:/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/configuration/native
SPARK_HOME=/opt/ibm/spss/analyticsserver/3.1.1/spark-client
HADOOP_HOME="/opt/client_hdfs/HDFS/hadoop"

```

17. /opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/server.xml ファイルを以下のように変更します。

```

<server description="new server">
<!-- Enable features -->
<featureManager>
<feature>servlet-3.1.1</feature>
<feature>jsp-2.3</feature>
<feature>jdbc-4.0</feature>
<feature>jndi-1.0</feature>
<feature>localConnector-1.0</feature>
<feature>jaxrs-2.0</feature>
<feature>json-1.0</feature>
<feature>appSecurity-2.0</feature>
<feature>ldapRegistry-3.0</feature>
<feature>restConnector-1.0</feature>
<feature>monitor-1.0</feature>
<feature>ssl-1.0</feature>
</featureManager>
<applicationManager startTimeout="120s" />
<executor name="LargeThreadPool" id="default" coreThreads="100" keepAlive="60s" stealPolicy="STRICT" rejectedWorkPolicy="CALLER_RUNS" />
<webContainer deferServletLoad="false" disallowAllFileServing="false" fileServingEnabled="true" trusted="false" directoryBrowsingEnabled="
false" asyncTimeoutDefault="300000"/>

```

```

<classloading useJarUrls="true"/>
<applicationMonitor updateTrigger="mbean" />
<mimeTypes>
  <type>svg=image/svg+xml</type>
</mimeTypes>
<variable name="AE_DATABASE" value="{wlp.install.dir}/usr/servers/aeserver/aedb" />
<administrator-role>
  <user>admin</user>
</administrator-role>
<include optional="true" location="{server.config.dir}/private_library.xml"/>
<include optional="true" location="{server.config.dir}/http_endpoint.xml"/>
<include optional="true" location="{server.config.dir}/security_cfg.xml"/>
<include optional="true" location="{server.config.dir}/ssl_cfg.xml"/>
<include optional="true" location="{server.config.dir}/configuration/key.xml" />
</server>

```

18. `private_library.xml` ファイルを `/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/` フォルダー内に作成します。ファイル内容は、以下のようになります。

```

<server>
  <application context-root="/analyticsserver" id="AS_BOOT" location="AE_BOOT.war" name="AS_BOOT" type="war">
    <classloader>
      <privateLibrary>
        <fileset dir="{wlp.install.dir}/usr/servers/aeserver/apps/AE_BOOT.war/WEB-INF/lib" includes="*.jar"/>
        <fileset dir="/usr/share/java" includes="*.jar"/>
        <fileset dir="{wlp.install.dir}/../lib" includes="*.jar"/>
        <fileset dir="{wlp.install.dir}/../spark-client/lib" includes="spark-assembly-*.jar"/>
        <folder dir="{wlp.install.dir}/usr/servers/aeserver/configuration/hadoop-conf"/>
        <fileset dir="{wlp.install.dir}/../jdbc" includes="postgresql-*.jar"/>
        <fileset dir="{wlp.install.dir}/../jdbc" includes="*.jar"/>
        <fileset dir="{wlp.install.dir}/../hive" includes="*.jar"/>
        <fileset dir="{wlp.install.dir}/../zookeeper" includes="*.jar"/>
        <fileset dir="{wlp.install.dir}/../hadoop" includes="*.jar"/>
      </privateLibrary>
    </classloader>
  </application>
  <application id="help" location="help.war" name="help" type="war" context-root="/analyticsserver/help"/>
</server>

```

19. `security_cfg.xml` ファイルを `/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/` フォルダー内に作成します。ファイル内容は、以下のようになります。

```

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

```

20. `http_endpoint.xml` ファイルを `/opt/ibm/spss/analyticsserver/3.1.1/ae_wlpserver/usr/servers/aeserver/` フォルダー内に作成します。ファイル内容は、以下のようになります。

```

<server>
  <httpEndpoint host="*" id="defaultHttpEndpoint" httpPort="9080" httpsPort="9443" onError="FAIL"/>
</server>

```

21. `jvm_option` ファイルを以下のように変更します。

```

-Xms512M
-Xmx2048M
-Dclient.encoding.override=UTF-8
-XX:+UseParNewGC
-Dconfig.folder.path=/opt/ibm/spss/analyticsserver/3.1.1/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. `hdfsUpdate.sh` スクリプト (`/opt/ibm/spss/analyticsserver/3.1.1/bin/hdfsUpdate.sh`) を実行します。

23. `start.sh` スクリプト (`/opt/ibm/spss/analyticsserver/3.1.1/bin/start.sh`) を実行して Analytic Server サービスを開始します。`stop.sh` スクリプト (`/opt/ibm/spss/analyticsserver/3.1.1/bin/stop.sh`) を実行して Analytic Server サービスを停止します。

24. Analytic Server コンソールには、URL `http://<servername>:9080/analyticsserver/admin/ibm` を介してアクセスできます。

第 6 章 IBM SPSS Analytic Server で使用するための IBM SPSS Modeler の構成

SPSS Modeler を Analytic Server で使用できるようにするには、SPSS Modeler Server インストール済み環境に対する更新をいくつか行う必要があります。

1. SPSS Modeler Server を構成して、Analytic Server インストール済み環境と関連付けます。
 - a. メインサーバーのインストール・ディレクトリーの config サブディレクトリーにある options.cfg ファイルを編集して、以下の行を追加または編集します。

```
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

Analytic Server でセキュア通信が構成されている場合は Y を指定して、それ以外の場合には N を指定してください。

as_host

Analytic Server をホストするサーバーの IP アドレス。

as_port

Analytic Server が listen するポート (デフォルトは 8080)。

as_context_root

Analytic Server コンテキスト・ルート (デフォルトは analyticserver)。

as_tenant

SPSS Modeler Server インストール済み環境がメンバーになっているテナント (デフォルトのテナントは ibm)。

as_prompt_for_password

SPSS Modeler Server が、Analytic Server で使用されているユーザーおよびパスワードの認証システムと同じ認証システムを使用して構成されている場合 (例えば、Kerberos 認証を使用している場合) は N を指定します。そうでない場合は、Y を指定します。

SPSS Modeler をバッチ・モードで実行している場合、clemb コマンドの引数として `-analytic_server_username {ASusername} -analytic_server_password {ASpassword}` を追加します。

as_kerberos_auth_mode

SPSS Modeler からの Kerberos SSO を有効にする場合は Y を指定します。

as_kerberos_krb5_conf

Analytic Server で使用する Kerberos 構成ファイルへのパスを指定します (例: `¥etc¥krb5.conf`)。

as_kerberos_krb5_spn

Analytic Server Kerberos SPN を指定します (例: HTTP/ashost.mydomain.com@MYDOMAIN.COM)。

- b. SPSS Modeler Server サービスを再開します。

SSL/TLS が有効になっている Analytic Server インストール済み環境に接続するには、SPSS Modeler Server とクライアントのインストール済み環境を構成するための追加のステップがいくつかあります。

- a. `http{s}://{HOST}:{PORT}/{CONTEXT-ROOT}/admin/{TENANT}` にナビゲートして、Analytic Server コンソールにログオンします。
- b. ブラウザーから認証ファイルをダウンロードして、ファイル・システムに保存します。
- c. 認証ファイルを SPSS Modeler Server と SPSS Modeler Client の両方のインストール済み環境の JRE に追加します。更新する場所は、SPSS Modeler インストール・パスの `/jre/lib/security/cacerts` サブディレクトリーで見つかります。
 - 1) `cacerts` ファイルが読み取り専用でないことを確認します。
 - 2) Modeler に付属の `keytool` プログラムを使用します。これは、SPSS Modeler インストール・パスの `/jre/bin/keytool` サブディレクトリーにあります。

次のコマンドを実行します。

```
keytool -import -alias <as-alias> -file <cert-file> -keystore "<cacerts-file>"
```

<as-alias> は `cacerts` ファイルの別名であることに注意してください。 `cacerts` ファイルに固有のものである限り、任意の名前を使用できます。

以下にコマンドの例を示します。

```
keytool -import -alias MySSLCertAlias -file C:%Download%as.cer  
-keystore "c:%Program Files%IBM%SPSS%Modeler%{ModelerVersion}%jre%lib%security%cacerts"
```

- d. SPSS Modeler Server および SPSS Modeler Client を再起動します。
2. [オプション] Analytic Server データ・ソースを使用してストリーム内の R モデルのスコアリングを行う予定の場合、IBM SPSS Modeler - Essentials for R をインストールします。IBM SPSS Modeler - Essentials for R は、<https://www14.software.ibm.com/webapp/iwm/web/preLogin.do?source=swg-tspssp> からダウンロードできます。

第 7 章 SLM タグを使用したライセンス交付の追跡

SLM タグは、Resource Utilization Measurement の ISO/IEC 19770-4 規格のドラフトに基づいています。SLM タグは、製品がライセンス・メトリック (ソフトウェア資産の使用に関連するリソース) の使用量を報告するための標準化機能を提供します。製品の SLM を有効にすると、そのライセンス使用を自己報告するためのランタイム XML ファイルが生成されます。

Analytic Server が開始されると、slmtag ファイルが `<as_installation_path>/logs/slmtag` フォルダ内に作成されます。

2 つのライセンス・タイプがあるため、異なる 2 つのメトリックが定期的に記録されます。

- Analytic Server の現行バージョンの場合、ライセンス交付は、(仮想サーバーに基づいた) Hadoop クラスター内のデータ・ノードの総数によって決まります。ノード数は以下の slmtag ファイル・セクションに記録されます。

```
<Type>VIRTUAL_SERVER</Type>
<SubType>Number of Data Nodes in Hadoop</SubType>
<Value>2</Value>
...
```

- Analytic Server の 3.1 より前のバージョンの場合、ライセンス交付は、(RVU に基づいた) Hadoop クラスター内の HDFS ストレージのサイズによって決まります。例えば、ストレージ・サイズ (単位: テガバイト) は以下の slmtag ファイル・セクションに記録されます。

```
<Type>RESOURCE_VALUE_UNIT</Type>
<SubType>HDFS storage (Unit: Tega byte)</SubType>
<Value>0.21</Value>
```

SLM タグ出力がスレッド内で開始されますが、それは `SlmTagOutput.properties` ファイル内で定義されたプロパティの影響を受けます。このファイルは、`<as_installation_path>/configuration` フォルダ内にあります。

表 12. SLM タグのプロパティ

プロパティ	説明
<code>license.metric.logger.output.enabled</code>	SLM ログ・ファイル生成を制御します。デフォルト値は <code>False</code> です。
<code>license.metric.logger.output.dir</code>	SLM タグ・ファイルを格納するディレクトリーの相対パス。デフォルト・ディレクトリーは <code><as_installation_path>/logs</code> です。
<code>license.metric.logger.output.SLMLogFrequency</code>	SLM ログ収集の時間間隔 (単位: ミリ秒)。
<code>license.metric.logger.file.size</code>	SLM タグ・ファイルの最大サイズ (単位: バイト)。
<code>license.metric.logger.file.number</code>	1 つのソフトウェア ID インスタンスに対する SLM タグ・ファイルの最大数。

第 8 章 トラブルシューティング

このセクションでは、インストールおよび構成の一般的な問題とその解決方法を説明します。

一般的な問題

インストールが警告付きで成功するが、「要求を完了できません。理由: 権限が拒否されました (Permission denied)」のエラーが発生してユーザーがデータ・ソースを作成できない

distrib.fs.root パラメーターを Analytic Server ユーザー (デフォルトでは `as_user`) がアクセス権限を持たないディレクトリーに設定すると、エラーが発生します。Analytic Server ユーザーが **distrib.fs.root** ディレクトリーに対して読み取り、書き込み、および実行を許可されるようにしてください。

Analytic Server のパフォーマンスが徐々に低下している

Analytic Server のパフォーマンスが予期されるレベルに到達しない場合は、Knox サービス・デプロイメント・パス `<KnoxServicePath>/data/deployments` からすべての `*.war` ファイルを削除します。例: `/usr/iop/4.1.0.0/knox/data/deployments`

Ambari での **Analytic Server** または **Essentials for R** のアンインストール

場合によっては、Ambari で Analytic Server または Essentials for R をアンインストールするときに、アンインストール・プロセスがハングすることがあります。この問題が発生した場合は、Ambari サーバーのプロセス ID を手動で停止する必要があります。

OpenJDK を使用する **POWER Systems** に **Analytic Server** をインストールする際の問題

OpenJDK を使用する POWER Systems で Analytic Server を実行する際は、以下の構成ステップを手動で実行して、座標系 API が予期されたとおりに機能することを確認する必要があります。

注: 座標系 API を使用しない場合は、構成要件を無視して構いません。

1. Ambari コンソールで、「**Analytic Server service**」 > 「**Configs tab**」 > 「**Advanced analytics-jvm-options**」に移動し、以下の行をコンテンツ領域に追加します。

```
-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCShorizon.*
```

2. Ambari コンソールで、「**Custom analytics.cfg**」セクションに移動し、以下の 3 つの構成を追加します。

spark.executor.extraJavaOptions

値を `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCShorizon.*` に設定します。

spark.driver.extraJavaOptions

値を `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCShorizon.*` に設定します。

mapred.child.java.opts

値を `-XX:CompileCommand=exclude,com/esri/sde/sdk/pe/engine/PeHznTwoPointEquidistant$GCShorizon.*` に設定します。

特定の Hadoop ディストリビューションに関する問題

Analytic Server サービスに対するリフレッシュ・アクションが **Hortonworks 2.3 - 2.6** で無効になっている Hortonworks 2.3 - 2.6 上の Analytic Server ライブラリーを手動でリフレッシュするには、以下のステップを使用します。

1. Analytic Metastore を実行しているホストに Analytic Server ユーザー (デフォルトでは `as_user`) としてログオンします。

注: このホスト名は Ambari コンソールから確認できます。

2. `{AS_ROOT}/bin` ディレクトリーにある **refresh** スクリプトを実行します。例えば、次のようにします。

```
cd /opt/ibm/spss/analyticserver/3.1.1/bin
./refresh
```

3. Ambari コンソールで Analytic Server サービスを再始動します。

外部サイトからダウンロードされたパッケージが、**Cloudera Manager** 内のハッシュ検査で不合格になるハッシュ検査エラーがパーセル・リストに表示されます。ダウンロード・プロセスが完了するまで待機してから、`cloudera-scm-server` サービスを使用して Cloudera を再始動することにより、この問題を解決できます。サービスが再始動されると、エラーは発生しなくなります。

メタデータ・リポジトリーに関する問題

add_mysql_user スクリプトの実行時に **CREATE USER** 操作が失敗する

add_mysql_user スクリプトを実行する前に、まず、追加しようとしているユーザーを mysql データベースから手動で削除する必要があります。MySQL Workbench UI または MySQL コマンドによってユーザーを削除できます。以下に例を示します。

```
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'@'%';"
```

上記のコマンドで、削除するユーザー名で `$AEDB_USERNAME_VALUE` を置換し、データベースがインストールされているホスト名で `$METASTORE_HOST` を置換してください。

Spark プロセス内で実行される IBM SPSS Modeler ストリームに関する問題

Spark プロセス内で強制的に実行された SPSS Modeler ストリームを完了できませんでした。失敗した SPSS Modeler ストリームは、Analytic Server ソース・ノード (HDFS ファイル) を使用して作成されていました。このソース・ノードは、Sort ノードにリンクされていて、さらにその後には別の Analytic Server データ・ソースにエクスポートするように設定されています。ストリームが実行された後で、リソース・マネージャー・ユーザー・インターフェースは、新規アプリケーションが実行されているが、ストリームはいつまでも完了せず Running 状態のままであることを示します。Analytic Server ログ、YARN ログ、および Spark ログには、ストリームを完了できなかった理由を示すメッセージはありません。

Analytic Server 構成内の `Custom analytics.cfg` ファイルに `spark.executor.memory` 設定を追加することにより、この問題を解決できます。メモリー値を 4GB に設定すると、以前に失敗した SPSS Modeler ストリームを 2 分未満で完了できます (単一ノード・クラスター環境の場合)。

高可用性クラスター

依存関係の変更が原因で **Analytic Server** を追加ホストに追加できない

25 ページの『クライアント依存関係の更新』の手順に従って `update_clientdeps` スクリプトを実行します。

java.net.SocketTimeoutException: Read timed out

Liberty ND のタイムアウト環境変数を以下のように変更します。

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

ここで、<milliseconds> は JMX の読み取りタイムアウトに使用する秒数です。

java.io.IOException: CWWKX7202E: コマンド `./server start` のタイムアウト値 **60 (秒)** が経過しました

コントローラー・サーバーの `server.xml` に以下を追加します。

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

java.lang.OutOfMemoryError: Java heap space

以下の行を HA クラスターのすべてのメンバーの `jvm.options` に追加します。

```
-Xms512M
-Xmx2048M
```

「分析クラスター・サービスと **Zookeeper** との接続が予期せず切断されました。クラスターの整合性を保つため、この **JVM** を終了しています。」

この状態が発生する可能性のある原因の 1 つとして、**Zookeeper** に書き込まれるデータの量が多すぎることがあります。 **Zookeeper** のログに以下のような例外がある場合:

```
java.io.IOException: Unreasonable length = 2054758
```

あるいは、**Analytic Server** のログに以下のようなメッセージがある場合:

```
Caused by: java.io.UTFDataFormatException: encoded string too long: 2054758 bytes
at java.io.DataOutputStream.writeUTF(DataOutputStream.java:375)
```

1. Ambari コンソールで、**Zookeeper** サービスの「Configs」タブにナビゲートし、以下の行を `env-template` に追加してから、**Zookeeper** サービスを再始動します。

```
export JVMFLAGS="-Xmx2048m -Djute.maxbuffer=2097152"
```

2. Ambari コンソールで、**Analytic Server** サービスの「Configs」タブに移動し、以下の行を `Advanced analytics-jvm-options` に追加してから、**Analytic** クラスター・サービスを再始動します。

```
-Djute.maxbuffer=2097152
```

`jute.maxbuffer` 設定に対して指定する数値は、例外メッセージで示されている数値よりも大きくする必要があります。

Zookeeper のトランザクション・データが管理不能になる

`zoo.cfg` の `autopurge.purgeInterval` パラメーターを 1 に設定して、**Zookeeper** トランザクション・ログの自動消去を有効にします。

Analytic クラスター・サービスが **Zookeeper** との接続を失う

`zoo.cfg` の `tickTime`、`initLimit`、および `syncLimit` の各パラメーターを確認して変更します。以下に例を示します。

```
# 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
```

詳細については、Zookeeper の資料 (<https://zookeeper.apache.org/doc/r3.3.3/zookeeperAdmin.html>) を参照してください。

Analytic Server ジョブが再開されない

Analytic Server ジョブが再開されない一般的な状態があります。

- クラスタ・メンバーの障害が原因で **Analytic Server** ジョブが失敗した場合、通常そのジョブは他のクラスタ・メンバー上で自動的に再開されます。ジョブが再開されない場合、高可用性クラスタ内に少なくとも 4 つのクラスタ・メンバーが存在することを確認してください。

サーバーのシャットダウン時に **Analytic Server** サーバーが時折ハングする
手作業でサーバーを kill してください。

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。この資料の他の言語版を IBM から入手できる場合があります。ただし、これを入手するには、本製品または当該言語版製品を所有している必要がある場合があります。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510

東京都中央区日本橋箱崎町19番21号

日本アイ・ビー・エム株式会社

法務・知的財産

知的財産権ライセンス渉外

IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Director of Licensing

IBM Corporation

North Castle Drive, MD-NC119

Armonk, NY 10504-1785

US

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

記載されている性能データとお客様事例は、例として示す目的でのみ提供されています。実際の結果は特定の構成や稼働条件によって異なります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願います。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があります、単に目標を示しているものです。

表示されている IBM の価格は IBM が小売り価格として提示しているもので、現行価格であり、通知なしに変更されるものです。卸価格は、異なる場合があります。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名前はすべて架空のものであり、名前や住所が類似する個人や企業が実在しているとしても、それは偶然にすぎません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_. All rights reserved.

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

IT Infrastructure Library は AXELOS Limited の登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、および Pentium は、Intel Corporation または子会社の米国およびその他の国における商標または登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における登録商標です。

Microsoft、Windows、Windows NT および Windows ロゴは、Microsoft Corporation の米国およびその他の国における商標です。

ITIL は AXELOS Limited の登録商標です。

UNIX は The Open Group の米国およびその他の国における登録商標です。

Cell Broadband Engine は、Sony Computer Entertainment, Inc. の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

Linear Tape-Open、LTO、LTO ロゴ、Ultrium および Ultrium ロゴは、HP、IBM Corp. および Quantum の米国およびその他の国における商標です。



Printed in Japan

日本アイ・ビー・エム株式会社

〒103-8510 東京都中央区日本橋箱崎町19-21