Copyright © 2012 NTT DATA INTRAMART CORPORATION

目次

改訂情報

変更年月日	変更内容
2012-10-01	初版
2012-11-01	第2版 下記を追加・変更しました ▪ 「*時刻の調整*」の追加
2013-04-01	第3版 下記を追加・変更しました ▪ 「*キースペースの削除方法*」の追加
2013-07-01	第4版 下記を追加・変更しました ▪ 「*Cassandra への接続認証設定*」の追加
2013-10-01	第5版 下記を追加・変更しました ▪ 「*Windows サービスへの登録・削除*」の動作検証済み Commons Daemon のバージョンを 1.0.10 から 1.0.15 に更新 ▪ 「*Cassandraのクラスタ構築*」に利用するポートについて追加 ▪ 「*Cassandraのクラスタ構築*」の**replication-factor**に関する説明を更新
2013-10-11	第6版 下記を追加・変更しました ▪ 「*Windows サービスへの登録・削除*」の動作検証済み Commons Daemon のバージョンを 1.0.15 から 1.0.14 に修正 ▪ 「*Windows サービスへの登録・削除*」に Commons Daemon のダウングレード手順を追加
2014-01-01	第7版 下記を追加・変更しました ▪ Cassandraのバージョンを Cassandra 1.1.4 から Cassandra 1.1.12 に変更 ▪ 「*Cassandra のバージョンアップ*」を追加 ▪ 「*Cassandra の参考情報*」にdatastaxのApache Cassandra 1.1 Documentationを追加 ▪ 「*ノードの追加*」の「以下のコマンドを実行し、以下のように各ノードが表示されることを確認できれば完了です。」を「以下のコマンドを実行し次のように各ノードが表示され、Effective-Ownershipの合計値が「RF * 100%」の値となっていることを確認できれば完了です。」に変更 ▪ 「*Cassandra の運用*」に記載されていた内容を「*Cassandra の操作*」に記載し、「*Cassandra の運用*」の内容を変更 ▪ 「*スナップショット*」にWindows環境における注意事項を追加

変更年月日	変更内容
2014-04-01	第8版 下記を追加・変更しました ■ 「*Cassandra のセットアップ*」にセットアップガイドへの注意事項を追加 ■ 「*Cassandra への接続認証設定*」にintra-mart Accel Platform 2014 Spring(Granada)以降のバージョン利用に関する注意事項を追加 ■ 「*Cassandra への接続認証設定*」に認証設定に関する注意事項を追加 ■ 「*Cassandraのクラスタ構築*」にクライアント接続ポートに関する注意事項を追加 ■ 「*Cassandraのクラスタ構築*」にセットアップガイドへの注意事項を追加 ■ 「*Cassandra の操作*」にキースペースの作成方法、キースペースの作成方法（認証設定ありの場合）を追加 ■ 「*Cassandra の参考情報*」の「cassandra.yaml 主な項目一覧」に項目を追加
2014-05-01	第9版 下記を追加・変更しました ■ 「*タイムアウト*」を追加 ■ 「*Cassandra のバージョンアップ*」の「data_file_directories」プロパティを流用する手順をコピーする手順に修正 ■ 「*キースペースの削除方法*」にノートを追加
2014-05-30	第10版 下記を追加・変更しました ■ 「*Cassandra の参考情報*」cassandra.yaml 主な項目一覧のauthenticator、authorityの説明を認証とのユーザ承認についての記載に修正 ■ 「*Cassandra のバージョンアップ*」にJDKのバージョンアップ手順を追加
2014-09-01	第11版 下記を追加・変更しました ■ 「*Apache commons daemon の取得*」に取得する実行ファイルの記載を追加 ■ 「*Cassandraの起動 for Linux*」の一部誤記を修正
2015-08-01	第12版 下記を追加・変更しました ■ 「*クラスタの構成*」に通信用アドレスを指定することについての記載を追加 ■ 「*Cassandra への接続認証設定*」-「*アクセス権設定 (access.properties)*」の注意事項に、キースペースごとにアクセス権限を設定する必要がある記載を追加
2015-12-01	第13版 下記を追加・変更しました ■ 「*Cassandra への接続認証設定*」-「*認証設定の追加*」のユーザ名、および、パスワードの設定に関する注意事項を修正 ■ 「*Cassandra の操作*」-「*スナップショットデータによる復旧（リストア）*」のシステム情報に関する記載を追加

変更年月日	変更内容
2016-04-01	第14版 下記を追加・変更しました ■ 「*Cassandra のセットアップ*」-「*Linuxデーモンへの登録、削除*」のデーモンスクリプトを修正 ■ 「*Cassandraのクラスタ構築*」-「*ノードの追加*」のnodetoolコマンドをringに修正
2018-12-01	第15版 下記を追加・変更しました ■ 「*Cassandra のセットアップ*」-「*Apache Cassandra の取得*」に Java 11 対応版 Cassandra 1.1.12 の取得方法について追記 ■ 「*Cassandra のセットアップ*」-「*Windows サービスへの登録・削除*」に Java 11 対応版 Apache commons daemon 1.0.14 の取得方法について追記 ■ 「*Cassandra のセットアップ*」-「*Windows サービスへの登録・削除*」から「Apache commons daemon のダウングレード方法」を削除 ■ 「*Cassandra のバージョンアップ*」-「*Java 11 対応版 Cassandra 1.1.12 への移行*」を追記
2020-12-01	第16版 下記を追加・変更しました ■ 「*Cassandra のセットアップ*」-「*Windows サービスへの登録・削除*」の Java 11 対応版 Apache commons daemon 1.0.14 の取得方法のコラムを修正 ■ 「*Cassandra のセットアップ*」-「*Windows サービスへの登録・削除*」の Java 11 対応版 Apache commons daemon 1.0.14 の取得方法に注意書きを追加

はじめに

本書の内容

本書では Apache Cassandra の詳細について記載されています。
説明範囲は以下のとおりです。

- Apache Cassandra の概要
- Apache Cassandra のインストール方法
- Apache Cassandra の利用方法

対象読者

以下の利用者を対象としています。

- IMBoxをモジュール構成に含めて intra-mart Accel Platformをセットアップする方
- intra-mart Accel Platformのデータベース（Cassandra）を管理する運用担当者の方

本書に記載されている外部サイトのURL

本書内で記載されている外部URLは、 2018-12 現在のものです。

Cassandra の概要

ここでは Apache Cassandra の概要や、基本的な知識を紹介します。

項目

Apache Cassandra とは

Facebook社が開発したオープンソースのKey-Value形式でのデータベース管理システムです。
Facebook内でのユーザメッセージの検索機能で利用され、その後、2009年3月にApache Software Foundationに寄贈され、2010年2月にApacheトップレベルプロジェクトに昇格しました。

Cassandra の特徴

Cassandra は、ブリュワーのCAP定理のうち、AP（可用性：Availability とネットワーク分断耐性：Partition Tolerance）を
重視しているため、以下のような特徴を備えています。

- 高いパフォーマンス
- 分散型で伸縮自在なスケーラビリティ
- 単一故障点（SPOF）がないアーキテクチャ
- 高可用性

Cassandra の一貫性保証

このように、可用性を高めるかわりに、ブリューワのCAP定理での一貫性：Consistencyを犠牲にしてます。
しかし、遅延とのトレードオフで一貫性のレベル設定することが可能で、Cassandra で設定できるレベルは、

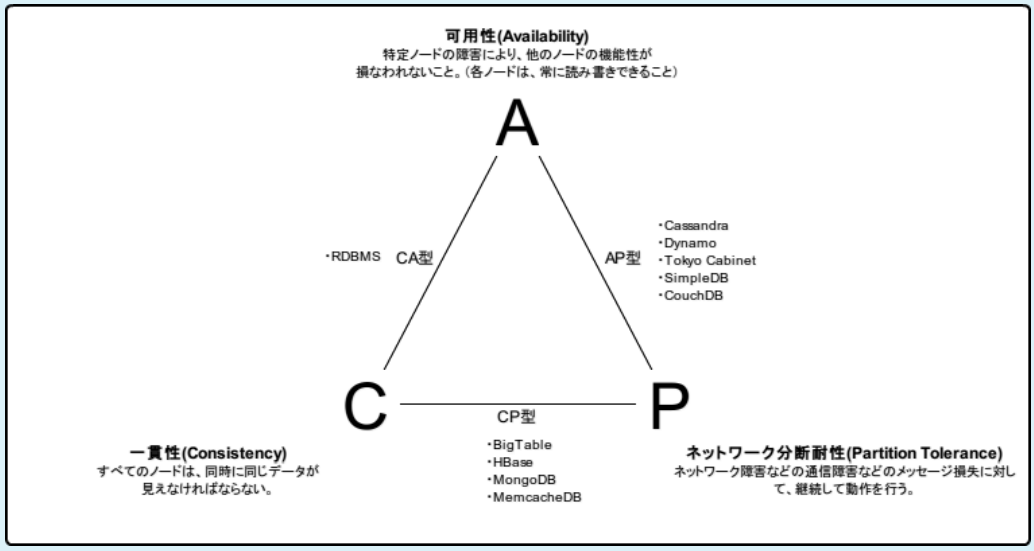
- Zero（一貫性の保証なし）
- One（１つのみの一貫性を保証）
- Quorum（(ノード数+1)/2の数分の一貫性を保証）
- ALL（全ノード数分の一貫性を保証）

があり、IMBoxでは、**Quorum** を採用しています。

コラム

ブリュワーのCAP定理

分散コンピュータシステムでのデータ複製において、同時に次の３つの保証を提供することはできないという定理。



- 一貫性（Consistency）

  すべてのノードは、同時に同じデータが見えなければならない。

- 可用性（Availability）

  特定ノードの障害により、他のノードの機能性が損なわれないこと。

- ネットワーク分断耐性（Partition Tolerance）

  ネットワーク障害などの通信障害などのメッセージ損失に対して、継続して動作を行う。

一般的にRDBは、一貫性（Consistency）と可用性（Availability）を満たしたCA型です。

Cassandra のセットアップ

ここでは Cassandra のセットアップ方法や、各種設定について説明をします。

Apache Cassandra の取得

2018 年 12 月現在、intra-mart Accel Platform で利用可能な Apache Cassandra のバージョンは 1.1.12 です。

> **注意**
>
> 2018 年 12 月現在の Apache Cassandra の最新版は、3.11.3 です。
> 1.2 ～ 3.11 は、 intra-mart Accel Platform で利用している Cassandra のライブラリがサポートしていないため、 1.1 台の最新の 1.1.12 を利用してください。
> Cassandra のバージョンアップに関しては、 *Cassandra のバージョンアップ* を参照してください。

項目

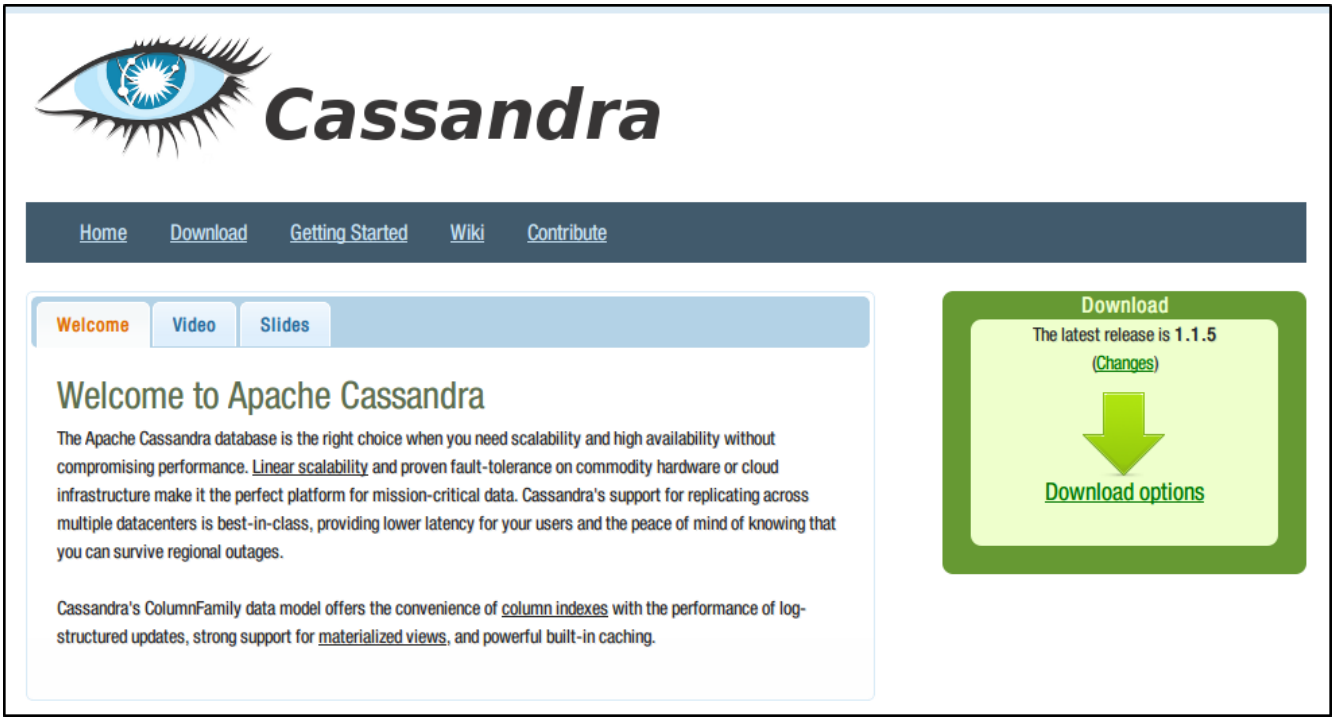
Java 11 対応版 Cassandra 1.1.12 の取得

Apache Cassandra のホームページより取得した Cassandra 1.1.12 は、Java 11 以降で起動が行えません。
弊社にて、Java 11 以降で起動可能に設定した Cassandra 1.1.12 をプロダクトファイルダウンロードにて配布しています。
Java 11 以降で Cassandra の運用を行う場合は「 プロダクトファイルダウンロード 」より<apache-cassandra-1.1.12-bin.tar.gz>をダウンロードしてください。

> **注意**
>
> 上記で取得した Cassandra 1.1.12 は、Java 8 で起動できません。Java 8 で運用を行う場合は *Cassandra 公式サイトからの Cassandra 1.1.12 の取得* を行ってください。

Cassandra 公式サイトからの Cassandra 1.1.12 の取得

Apache Cassandra のホームページより、リリースノートに記載のバージョンの Cassandra をダウンロードしてください。



コラム

URL（2018 年 12 月現在）

- Apache Cassandra プロジェクトページ：http://cassandra.apache.org/ (English)
- リリースアーカイブ：http://archive.apache.org/dist/cassandra/ (English)
- Apache Cassandra 1.1.12：http://archive.apache.org/dist/cassandra/1.1.12/apache-cassandra-1.1.12-bin.tar.gz

Cassandraのセットアップ for Windows

ここでは Windows環境へのCassandraのセットアップ方法を説明します。

項目

ファイルの展開

「*Apache Cassandra の取得*」でダウンロードした、<apache-cassandra-1.1.12-bin.tar.gz>ファイルを任意のパスに展開します。
※ファイルの展開には、TAR-GZ形式の圧縮ファイルが展開できる解凍ツールを利用してください。

> **コラム**
>
> 本書では、例として次のディレクトリを指定します。
>
> 「C:/cassandra/apache-cassandra-1.1.12」
>
> Cassandraを展開したディレクトリを以後、%CASSANDRA_HOME% と略します。

各種設定

- Cassandraを起動する前に、以下の設定が必要になります。

Cassandraの設定

- Cassandraの設定が記載されている、<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルをエディタで開き、編集を行います。
- cassandra.yamlファイルに関する詳細は、「*Cassandra の参考情報*」-「*cassandra.yaml 主な項目一覧*」を参照してください。

1. クラスタ名の指定
   「cluster_name」 プロパティに任意のクラスタの名称を指定します。

   ```
   cluster_name: 'IMBox Cluster'
   ```

   > **注意**
   >
   > Cassandraサーバが複数存在する場合、クラスタ名で同一クラスタのCassandraであるか判断され、クラスタが組まれてしまう場合があります。
   > 新規に intra-mart Accel Platform を構築する場合、初期値を変更することを推奨します。

2. データの保存場所の指定
   「data_file_directories」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

   ```
   data_file_directories:
     - C:/cassandra/data
   ```

3. コミットログの保存場所の指定
   「commitlog_directory」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

   ```
   # commit log
   commitlog_directory: %CASSANDRA_HOME%/commit_log
   ```

4. キャッシュデータの保存場所の指定
   「saved_caches_directory」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

   ```
   # saved caches
   saved_caches_directory: %CASSANDRA_HOME%/saved_caches
   ```

> **コラム**
>
> Cassandraをバージョンアップする際には、旧バージョンのデータを新バージョンのデータとして引き継いで利用します。
> バージョンアップについての詳細は、*Cassandra のバージョンアップ* を参照してください。

5. 他の Cassandra との通信用アドレスの指定
   Cassandraでは、JavaのInetAddress.getLocalHost()で取得されたIPアドレスで他のCassandraとの通信を受け付けますが、複数のアドレスがある場合、明示的に通信に利用するIPアドレスを指定する必要があります。

```
listen_address: 192.168.xxx.xxx(任意のIPアドレス)

rpc_address: 192.168.xxx.xxx(任意のIPアドレス)
```

6. システムログの指定
   <%CASSANDRA_HOME%/conf/log4j-server.properties>ファイルをエディタで開き、
   「log4j.appender.R.File」プロパティに任意のパスを指定します。

```
log4j.appender.R.File=%CASSANDRA_HOME%/system.log
```

メモリの設定

インストール環境に応じたメモリ値を指定します。
<%CASSANDRA_HOME%/bin/cassandra.bat>ファイルをエディタで開きます。
「Xms」プロパティに最小ヒープサイズ、「Xmx」プロパティに最大ヒープサイズを指定します。指定する値は、必ず下記の記載例以上にしてください。

```
set JAVA_OPTS=-ea^
 -javaagent:"%CASSANDRA_HOME%lib\jamm-0.2.5.jar"^
 -Xms512M^
 -Xmx512M^
```

> **注意**
>
> 設定する値は「512M」以上にしてください。
> 「512M」より小さい値を設定した場合、正常に動作しない恐れがあります。
> Cassandraとしては1GB以上に設定することを推奨されています。

環境変数の設定

Windows環境変数に JDKをインストールしたホームディレクトリを追加します。
WindowsOSのマニュアルに従い、次のように設定します。

変数	JAVA_HOME
値	JDK をインストールしたホームディレクトリ

> **コラム**
>
> - 事前にJDKがインストールされている必要があります。
>   JDKのバージョンは「 リリースノート 」-「 Apache Cassandra システム要件 」に準拠します。

Windows サービスへの登録・削除

ここでは Cassandra の Windows サービスへの登録・削除方法を説明します。

項目

Windows サービスへの登録

Apache commons daemon 1.0.14 を利用して Cassandra を Windows サービスで起動可能にします。

Apache commons daemon の取得

Java 11 以降で運用を行う場合の Apache commons daemon の取得

Apache commons daemon のホームページより取得した Apache commons daemon 1.0.14 は、Java 11 以降での Cassandra のサービス起動が行えません。
弊社にて、Java 11 以降で起動可能に設定した Apache commons daemon 1.0.14 をプロダクトファイルダウンロードにて配布しています。
Java 11 以降で Cassandra の運用を行う場合は「プロダクトファイルダウンロード」より<commons-daemon-1.0.14-bin-windows.zip>をダウンロードしてください。

> **コラム**
> 上記で取得した commons-daemon を利用する場合、初期状態では *Windows サービスへの登録* 実行時の環境変数 JAVA_HOME で指定している Java を利用して Cassandra サービスが起動します。

> **コラム**
> 上記で取得した commons-daemon を利用する場合、Visual Studio 2015、2017、および 2019 用 Visual C++ 再頒布可能パッケージをインストールしてください。
>
> - https://support.microsoft.com/ja-jp/help/2977003/the-latest-supported-visual-c-downloads （日本語）
> - https://support.microsoft.com/en-us/help/2977003/the-latest-supported-visual-c-downloads （English）
> - https://support.microsoft.com/zh-cn/help/2977003/the-latest-supported-visual-c-downloads （中文）

注意

intra-mart Accel Platform 2020 Summer(Zephirine) 以前で提供していた <commons-daemon-1.0.14-bin-windows.zip> 使用してる場合、以下をインストールしても利用できない場合があります。

- Visual Studio 2015、2017、および 2019 用 Visual C++ 再頒布可能パッケージ

プロダクトファイルダウンロード より <commons-daemon-1.0.14-bin-windows.zip> をダウンロードしなおしてください。

Java 8 で運用を行う場合の Apache commons daemon の取得

下記のURLより、<commons-daemon-1.0.14-bin-windows.zip>をダウンロードしてください。

- http://archive.apache.org/dist/commons/daemon/binaries/windows/ (English)

コラム

URL（2018 年 12 月現在）

- Apache commons daemon プロジェクトページ：http://commons.apache.org/daemon/ (English)
- ダウンロードページ： http://commons.apache.org/daemon/download_daemon.cgi (English)
- commons-daemon-1.0.14：http://archive.apache.org/dist/commons/daemon/binaries/windows/commons-daemon-1.0.14-bin-windows.zip

コラム

上記で取得した commons-daemon を利用する場合、初期状態では 以下の Windows Registry で設定されている Java を利用して Cassandra サービスが起動します。

- HKEY_LOCAL_MACHINE\SOFTWARE\JavaSoft\Java Runtime Environment

Apache commons daemon の配置

<%CASSANDRA_HOME%/bin/daemon>フォルダを作成し、別フォルダ等に <commons-daemon-1.0.14-bin-windows.zip>を展開します。
prunsrv.exe を<%CASSANDRA_HOME%/bin/daemon>フォルダに配置してください。
prunsrv.exe は、稼働させているサーバの CPU により、実行ファイルが異なるため、以下の表を参考に実行ファイルを選択してください。

プロセッサ	利用する **prunsrv.exe**
x86	解凍した zip 直下
x64（AMD64 系互換命令セット）	amd64 ディレクトリ配下
IA-64（Itanium）	ia64 ディレクトリ配下

コラム

プロダクトファイルダウンロード よりダウンロードを行った場合は、x64 （AMD64 系互換命令セット）用の prunsrv.exe のみ同梱されています。

Windows サービスへの登録

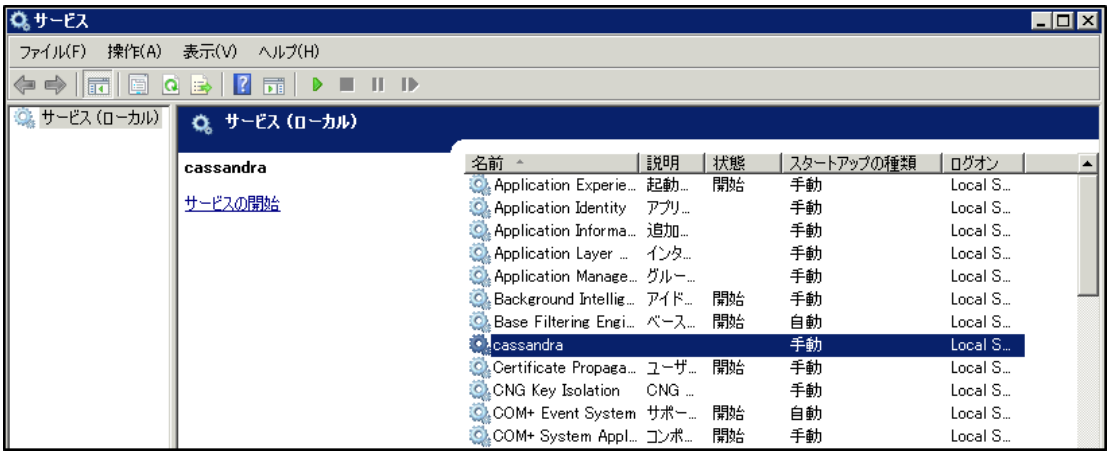
「コマンドプロンプト」を管理者として実行し、

```
%CASSANDRA_HOME%/bin/cassandra install
```

を実行してください。

```
Setting the parameters for "cassandra"
Installation of "cassandra" is complete
```

と表示され、「ローカルサービスの表示」 で以下のように表示されれば、登録は完了です。
必要に応じて、スタートアップの種類を「自動」等に変更してください。



Windows サービスの削除

「コマンドプロンプト」を管理者として実行し、

```
%CASSANDRA_HOME%/bin/cassandra uninstall
```

を実行してください。

```
trying to delete service if it has been created already
```

と表示され、「ローカルサービスの表示」から「cassandra」サービスが削除されていれば、削除は完了です。

Cassandraのセットアップ for Linux

ここでは Linux環境へのCassandraのセットアップ方法を説明します。

項目

ファイルの展開

「*Apache Cassandra の取得*」でダウンロードした、<apache-cassandra-1.1.12-bin.tar.gz>ファイルを任意のパスに展開します。

> **コラム**
>
> 本書では、例として次のディレクトリを指定します。
>
> 「/usr/local/cassandra」

> **コラム**
>
> /usr/local/apache-cassandra-1.1.12として展開されたものを
> /usr/local/cassandraにシンボリックリンクしておくと、バージョンアップの際などにアクセスが容易になります。
>
> ```
> # ln -s /usr/local/apache-cassandra-1.1.12 /usr/local/cassandra
> ```

各種設定

- Cassandraを起動する前に、以下の設定が必要になります。

Cassandraの設定

- Cassandraの設定が記載されている、</usr/local/cassandra/conf/cassandra.yaml>ファイルをエディタで開き、編集を行います。
- cassandra.yamlファイルに関する詳細は、「*Cassandra の参考情報*」-「*cassandra.yaml 主な項目一覧*」を参照してください。

1. クラスタ名の指定
   「cluster_name」プロパティに任意のクラスタの名称を指定します。

   ```
   cluster_name: 'IMBox Cluster'
   ```

   > **注意**
   >
   > Cassandraサーバが複数存在する場合、クラスタ名で同一クラスタのCassandraであるか判断され、クラスタが組まれてしまう場合があります。
   > 新規に intra-mart Accel Platform を構築する場合、初期値を変更することを推奨します。

2. データの保存場所の指定
   「data_file_directories」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

   ```
   data_file_directories:
     - /var/lib/cassandra/data
   ```

3. コミットログの保存場所の指定
   「commitlog_directory」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

```
# commit log
commitlog_directory: /var/lib/cassandra/commit_log/1.1.12
```

4. キャッシュデータの保存場所の指定
   「saved_caches_directory」プロパティに任意のパスを指定します。
   指定したディレクトリが無い場合、起動時に自動で作成されます。

```
# saved caches
saved_caches_directory: /var/lib/cassandra/saved_caches/1.1.12
```

5. 他のCassandraとの通信用アドレスの指定
   Cassandraでは、JavaのInetAddress.getLocalHost()で取得されたIPアドレスで他のCassandraとの通信を受け付けますが、複数のアドレスがある場合、明示的に通信に利用するIPアドレスを指定する必要があります。

```
listen_address: 192.168.xxx.xxx(任意のIPアドレス)

rpc_address: 192.168.xxx.xxx(任意のIPアドレス)
```

6. システムログの指定
   </usr/local/cassandra/conf/log4j-server.properties>ファイルをエディタで開き、
   「log4j.appender.R.File」プロパティに任意のパスを指定します

```
log4j.appender.R.File=/var/log/cassandra/1.1.12/system.log
```

注意

設定上の注意

OSとJDKバージョンの組み合わせにより、JVMパラメータのXssが不足し、Cassandraの起動に失敗する場合があります。
以下の組み合わせにおいて発生することを確認しています。

- OS: Linux系OS
- JDK: Oracle JDK 1.7u40 , Oracle JDK 1.7u45

回避方法

上記の事象は、${cassandra}/conf/cassandra-env.shに設定されているXssの値を変更することで回避可能です。

- Cassandraのバージョンが、1.1.4の場合

```
  if [ "`uname`" = "Linux" ] ; then
  # reduce the per-thread stack size to minimize the impact of Thrift
  # thread-per-client.  (Best practice is for client connections to
  # be pooled anyway.) Only do so on Linux where it is known to be
  # supported.
    if startswith "$JVM_VERSION" '1.7.'
    then
#         JVM_OPTS="$JVM_OPTS -Xss160k"
      JVM_OPTS="$JVM_OPTS -Xss228k"
    else
      JVM_OPTS="$JVM_OPTS -Xss128k"
    fi
 fi
```

- Cassandraのバージョンが、1.1.12の場合

```
 if [ "`uname`" = "Linux" ] ; then
  # reduce the per-thread stack size to minimize the impact of Thrift
  # thread-per-client.  (Best practice is for client connections to
  # be pooled anyway.) Only do so on Linux where it is known to be
  # supported.
  # u34 and greater need 180k

  #     JVM_OPTS="$JVM_OPTS -Xss180k"
  JVM_OPTS="$JVM_OPTS -Xss228k"
fi
```

ファイルディスクリプタ数の変更

OSの設定により、運用中にIOException, FileNotFoundException 等が発生し、Cassandraが正常に動作しなくなる場合があります。
原因は、Cassandraのプロセスが利用（オープン）できるファイル数の上限がOSにより制限されている上限を超える為に発生します。
つまり、/etc/system/limits.conf または/etc/security/limits.confで設定されるOSのファイルディスクリプタ数を環境に合わせた
以下の数を追加することにより、回避することができますので、適切な値に変更してください。

```
* soft nofile 32768
* hard nofile 32768
root soft nofile 32768
root hard nofile 32768
```

※ ユーザ、および、値はサンプルです、環境に合わせて適切な値を設定してください。

コラム

ファイルディスクリプタの現在の設定値は、

```
ulimit -n
```

で確認できます。

メモリの設定

Linux環境の場合、最大ヒープサイズ（Xmx）は</usr/local/cassandra/conf/cassandra-env.sh>により起動時に以下の値で自動設定されます。

- 搭載メモリ量が、2048MB以下の場合、搭載メモリの半分
- 搭載メモリ量が、2049MB以上、4099MB以下の場合、1024MB
- 搭載メモリ量が、4100MB以上、32768MB以下の場合、搭載メモリ量の1/4
- 搭載メモリ量が、32769MB以上の場合、8196MB

任意に設定したい場合は、 </usr/local/cassandra/conf/cassandra-env.sh>の以下の項目のコメントアウトを外し、任意の値を設定してください。

```
#MAX_HEAP_SIZE="4G"
#HEAP_NEWSIZE="800M"
```

注意

MAX_HEAP_SIZEに設定する値は「512M」以上にしてください。
「512M」より小さい値を設定した場合、正常に動作しない恐れがあります。
Cassandraとしては1GB以上に設定することを推奨されています。
HEAP_NEWSIZEはMAX_HEAP_SIZEの設定値に応じて適切な値を設定してください。

環境変数の設定

環境変数に JDKをインストールしたホームディレクトリを追加します。
Cassandraを実行するユーザの環境変数に、次のように設定します。

変数	JAVA_HOME
値	JDK をインストールしたホームディレクトリ

コラム

- 事前にJDKがインストールされている必要があります。
  JDKのバージョンは「 リリースノート 」-「 Apache Cassandra システム要件 」に準拠します。

Linuxデーモンへの登録、削除

ここでは Cassandra のLinuxデーモンへの登録、削除方法を説明します。

項目

Linuxデーモンへの登録

以下の内容を/etc/init.d/cassandraとして、作成して実行権限を付与してください。

```
#!/bin/sh
# chkconfig: 345 99 1
# description: cassandra
# processname: cassandra

CASSANDRA_BIN=/usr/local/cassandra/bin/cassandra
CASSANDRA_PID=/var/run/cassandra.pid

case "$1" in
  start)
    $CASSANDRA_BIN -p $CASSANDRA_PID
    echo "Running Cassandra"
    ;;
  stop)
    kill `cat $CASSANDRA_PID`
    rm -f $CASSANDRA_PID
    echo "Stopped Cassandra"
    ;;
  *)
    echo "Usage: $0 {start|stop}"
    exit 1
esac
exit 0
```

以下のchkconfigコマンドを実行して、起動時に自動するようにしてください。

```
# chkconfig --add cassandra
```

以下のchkconfigコマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list cassandra
cassandra      0:off   1:off   2:off   3:on    4:on    5:on    6:off
```

Linuxデーモンからの削除

以下のchkconfigコマンドを実行して、削除してください。

```
# chkconfig --del cassandra
```

以下のchkconfigコマンドを実行して、以下のように表示されれば成功です。

```
# chkconfig --list cassandra
サービス cassandra は chkconfig をサポートしますが実行レベルで参照されていません (run 'chkconfig --add cassandra')
```

必要に応じて、/etc/init.d/cassandra を削除してください。

コラム

intra-mart Accel PlatformのCassandra向けの設定に関しては別途、「intra-mart Accel Platform セットアップガイド」-「Apache Cassandra接続情報」を参照してください。
Cassandra のセットアップ後や、設定の変更を行った場合は必ず参照してください。

Cassandraの起動、停止方法

ここではCassandraの起動、停止方法を説明します。

項目

注意

Cassandraを運用する各ノードの時刻はかならず、NTP等で、同期してください。
各ノードでの時刻がずれているとCassandra自体が停止することがあります。

Cassandraの起動 for Windows

<%CASSANDRA_HOME%/bin/cassandra.bat> をダブルクリックします。

コマンドプロンプト上に次のメッセージの表示されたら起動は完了です。

```
INFO  02:33:54 Binding thrift service to localhost/127.0.0.1:9160
INFO  02:33:54 Listening for thrift clients...
```

「*Windows サービスへの登録・削除*」でWindowsサービス化している場合は、「ローカルサービスの表示」から起動してください。

コラム

Apache Cassandraの起動時、以下のエラーが発生する場合、ポート番号の設定を変更してください。

```
エラー: エージェントが例外をスローしました。 : java.rmi.server.ExportException: Port already in use: 7199;
nested exception is: java.net.BindException: Address already in use: JVM_Bind
```

<%CASSANDRA_HOME%/bin/cassandra.bat>ファイルをエディタで開きます。

-Dcom.sun.management.jmxremote.portプロパティに設定されている「7199」を、「7198」等の使用されていないポート番号に変更してください。

```
-Dcom.sun.management.jmxremote.port=7198^
```

変更が完了したら、再度Apache Cassandraを起動してください。

Cassandraの停止 for Windows

起動時に立ち上がったコマンドプロンプト上において **「Ctrl」+「C」** コマンドを実行し、プロセスを停止します。

注意

コマンドプロンプトの右上の×ボタンで終了した場合、終了処理が正しく行われず、データが破損する可能性があります。
必ず **「Ctrl」+「C」** コマンドで終了してください。

「*Windows サービスへの登録・削除*」でWindowsサービス化している場合は、「ローカルサービスの表示」から停止してください。

Cassandraの起動 for Linux

以下のコマンドでバックグランドで実行されます。

```
/usr/local/cassandra/bin/cassandra -p /var/run/cassandra.pid
```

コンソール上に次のメッセージの表示されたら起動は完了です。

```
INFO  02:35:04 Binding thrift service to /192.168.0.1:9160
INFO  02:35:04 Listening for thrift clients...
```

以下のコマンドで、Cassandraがデーモン化するのを防ぎ、フォアグラウンドで起動するよう強制できます。

```
/usr/local/cassandra/bin/cassandra -f
```

「*Linuxデーモンへの登録、削除*」で、Linuxデーモン化している場合は、以下のコマンドでバックグランドで実行されます。

```
/etc/init.d/cassandra start
```

コラム

Apache Cassandraの起動時、以下のエラーが発生する場合、ポート番号の設定を変更してください。

```
エラー: エージェントが例外をスローしました。 : java.rmi.server.ExportException: Port already in use: 7199;
nested exception is: java.net.BindException: Address already in use: JVM_Bind
```

</usr/local/cassandra/conf/cassandra-env.sh>ファイルをエディタで開きます。

JMX_PORTプロパティに設定されている「7199」を、「7198」等の使用されていないポート番号に変更してください。

```
JMX_PORT="7198"
```

変更が完了したら、再度Apache Cassandraを起動してください。

Cassandraの停止 for Linux

以下のコマンドで停止できます。

```
kill `cat /var/run/cassandra.pid`
```

下記のコマンドで起動した場合は、 **「Ctrl」+「C」** コマンドを実行し、プロセスを停止します。

```
/usr/local/cassandra/bin/cassandra -f
```

「*Linuxデーモンへの登録、削除*」で、Linuxデーモン化している場合は、以下のコマンドで停止できます。

```
/etc/init.d/cassandra stop
```

Cassandraのクラスタ構築

ここではCassandraのクラスタ構築方法を紹介します。
本書では以下の説明は【表1. Cassandra接続情報設定例】を元に記載しています。

- 【表1. Cassandra接続情報設定例】

1台目のノード（シードノード）	192.168.0.1
2台目のノード	192.168.0.2
3台目のノード	192.168.0.3
キースペース	default

注意

クラスタ構築のため、replication-factorやhostの変更を行った場合、intra-mart Accel Platform のセットアップ時に設定が必要となります。
詳しくは、「 intra-mart Accel Platform セットアップガイド 」-「 Apache Cassandra接続情報 」を参照してください。

項目

クラスタの構成

Cassandraは、1台のみでの運用も可能ですが、大量のメッセージを扱う場合での負荷分散や冗長化を実現するため、クラスタ構成を構築することが可能です。
「*Cassandra の一貫性保証*」での通り、IMBoxでは「**Quorum（(ノード数+1)/2の数分の一貫性を保証）**」を採用しているため、クラスタのノード数は **3台以上** にすることを強く推奨します。
クラスタ構成でCassandraを構築する場合、「*Cassandraの設定* - 他のCassandraとの通信用アドレスの指定」を参照して **listen_address**と**rpc_address**を設定してください。

ノードの追加

- **replication-factor** とhostの変更
  replication-factor （以下、**RF**と表記）とは、クラスタ内部でデータのレプリカを何個持つかという設定です。
  標準の1のままだと、該当のノードを持っているノードがダウンした場合にデータが参照できなくなります。
  IMBoxでは先述の通り一貫性の保証に**Quorum**を採用しているため、一貫性を保証するために下記表にある**RF**の値を推奨します。

ノード数	RF値（推奨値）	説明
1	1	
3	3	RF値を2とした場合、どれか1ノードが停止した時点で一貫性が保証できなくなるため、IMBoxが動作しなくなります。

ノード数	RF値（推奨値）	説明
4 以上	3 以上でノード数以下	対障害性とノード毎のデータサイズを考慮のうえ、任意の値を設定してください。

- シードノードを指定します。

各ノードの<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルをエディタで開き、「seeds」プロパティに1台目のノード(192.168.0.1)をシードノードとして指定します。

```
seed_provider:
    # Addresses of hosts that are deemed contact points.
    # Cassandra nodes use this list of hosts to find each other and learn
    # the topology of the ring.  You must change this if you are running
    # multiple nodes!
    - class_name: org.apache.cassandra.locator.SimpleSeedProvider
      parameters:
          # seeds is actually a comma-delimited list of addresses.
          # Ex: "<ip1>,<ip2>,<ip3>"
          - seeds: "192.168.0.1"
```

各ノードで設定を行った後に、Cassandraを起動します。

以下のコマンドを実行すると各ノードの情報が表示されます。**Owns(Effective-Ownership)**の合計値が「**RF** * 100%」の値となっていることを確認できれば完了です。

```
# /usr/local/cassandra/bin/nodetool ring
 （windowsの場合：C:/ > %CASSANDRA_HOME%/bin/nodetool ring）

Datacenter: datacenter1
==========
Replicas: 1
|/ State=Normal/Leaving/Joining/Moving
Address     DC          Rack     Status   State    Load       Effective-Ownership   Token
192.168.0.1  datacenter1 rack1     Up      Normal   550.14 KB    100.0%
82389416039062679366789129285745111778
192.168.0.2  datacenter1 rack1     Up      Normal   522.27 KB    100.0%
12435848436441025856657894230405668892
192.168.0.3  datacenter1 rack1     Up      Normal   455.95 KB    100.0%
59208578228297542987320156056687160295
```

コラム

Cassandraでは、デフォルトで以下のポートを利用して他のCassandraと通信を行います。
Linuxの場合、iptablesやSElinux、Windowsの場合、ファイアーウォールの設定を必要に応じて設定変更してください。

- 9160 : クライアント接続ポート（rpc_port） [1]
- 7000 : ノード間通信ポート（storage_port）
- 7001 : ノード間通信ポート（ssl_storage_port） [2]
- 7199 : JMX接続ポート（JMX_PORT）

クライアント接続ポート および **ノード間通信ポート** は、
<%CASSANDRA_HOME%/conf/cassandra.yaml> で設定されています。
JMX接続ポート は、Linuxの場合には <%CASSANDRA_HOME%/conf/cassandra-env.sh> に、Windowsの場合には <%CASSANDRA_HOME%/bin/cassandra.bat> で設定されています。

[1] 変更不可

[2] ノード間通信にSSLを利用した場合

注意

クライアント接続ポートに9160以外を指定した場合、新規ノードの検出機能にてエラーが発生します。
初期値の9160から値を変更せずにご利用ください。

コラム

「シードノード」とは？

Cassandraクラスタ上で中心（ハブ）のように機能し、新規にクラスタに参加したノードは、 シードノードから、他のノードの状態情報を取得したり、クラスタ全体の情報を取得します。 シードノードがなくても、動作自体には問題ありませんので、シードノード自体が単一障害点にはなりませんが、 クラスタ内のノードの状態の変化の検出に時間がかかるようになります。

既にIMBox環境を構築している場合

テナント環境セットアップを既に実行して、IMBox環境を構築している場合は以下の手順を行ってください。

1. 以下のディレクトリでcassandra-cliを実行します。

```
# %CASSANDRA_HOME%/bin/cassandra-cli.sh（または、cassandra-cli.bat）
```

2. 以下のコマンドでCassandraに接続します。

```
[default@unknown] connect 127.0.0.1/9160;
```

3. 以下のコマンドで IMBoxで使用しているKeyspaceを指定します。

```
[default@unknown] use default;
```

4. 以下のコマンドでreplication_factorを変更します。

```
[default@default] update keyspace default with strategy_options={replication_factor:2};
```

5. 以下のメッセ―ジが表示されれば成功です。

```
Waiting for schema agreement...
... schemas agree across the cluster
```

6. 以下のコマンドでcassandra-cliを終了し、Cassandraを再起動してください。

```
[default@default] exit;
```

7. 再起動後、以下のコマンドを実行してください。

```
# /usr/local/cassandra/bin/nodetool repair
（windowsの場合：C:/ > %CASSANDRA_HOME%/bin/nodetool repair）
```

ノードの削除

クラスタから任意のノードを削除したい場合、削除したいノードで、decommissionを実行します。
削除するノードのデータを他のノードに転送を行うため、終了するまでしばらく時間がかかります。

```
# /usr/local/cassandra/bin/nodetool -h [削除したいノードのIPアドレス] decommission
（windowsの場合：C:/ > %CASSANDRA_HOME%/bin/nodetool -h [削除したいノードのIPアドレス] decommission）
```

Cassandra への接続認証設定

ここではCassandraへの接続において、接続ユーザ名およびパスワードによる認証設定を行う方法を紹介します。

> **注意**
>
> - 接続認証の設定を行った場合、 intra-mart Accel Platform のセットアップ時に認証情報の設定が必要です。
>   詳しくは、「 intra-mart Accel Platform セットアップガイド 」-「 Apache Cassandra接続情報 」を参照してください。
> - 複数台の Cassandra によるクラスタ構成を構築する場合には、すべてのノードに対して設定を行う必要があります。

項目

認証ライブラリの取得と展開

Cassandra 用の認証ライブラリー式を入手します。
認証ライブラリは、「 プロダクトファイルダウンロード 」よりダウンロードできます。
入手したファイルを任意のパスに展開します。

ライブラリと設定ファイルの配置

認証ライブラリの圧縮ファイルを展開すると以下のファイルが展開されます。

1. **./lib/cassandra_simple_auth-1.0.0.jar**
   接続認証およびアクセス制御の実装クラスが格納されたライブラリファイルです。
2. **./conf/passwd.properties**
   接続認証に必要な接続ユーザ名とパスワードを記述するための設定ファイルです。
3. **./conf/access.properties**
   接続ユーザごとのアクセス権の制御設定を記述するための設定ファイルです。

それぞれのファイルを、%CASSANDRA_HOME%内の相対パスで対応するディレクトリにコピーします。

./lib/cassandra_simple_auth-1.0.0.jar → <%CASSANDRA_HOME%/lib/> にコピー
./conf/passwd.properties → <%CASSANDRA_HOME%/conf/> にコピー
./conf/access.properties → <%CASSANDRA_HOME%/conf/> にコピー

使用する認証クラスの変更

<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルをエディタで開きます。
Cassandra の標準設定では、下記の接続認証およびアクセス制御クラスが設定されています。

```
# authentication backend, implementing IAuthenticator; used to identify users
authenticator: org.apache.cassandra.auth.AllowAllAuthenticator

# authorization backend, implementing IAuthority; used to limit access/provide permissions
authority: org.apache.cassandra.auth.AllowAllAuthority
```

標準の設定では Allow All という名前の通り、接続認証やアクセス制御を行わずにすべての接続を許可します。
これを下記の様に、ダウンロードした認証ライブラリ内のクラスを指定するように修正します。

```
# authentication backend, implementing IAuthenticator; used to identify users
# authenticator: org.apache.cassandra.auth.AllowAllAuthenticator
authenticator: org.apache.cassandra.auth.SimpleAuthenticator

# authorization backend, implementing IAuthority; used to limit access/provide permissions
# authority: org.apache.cassandra.auth.AllowAllAuthority
authority: org.apache.cassandra.auth.SimpleAuthority
```

これで接続認証およびアクセス制御クラスの実装が差し替わるようになりました。

注意

- 弊社サイトにて提供している認証ライブラリ内の **SimpleAuthority** は、 Cassandra が提供しているサンプル実装に対して、 **access.properties** でキースペース内の全カラムファミリへのアクセス権を一括で設定可能にするカスタマイズを行っております。

認証設定の追加

前段にて差し替えた実装クラスは、設定ファイル **access.properties** と **passwd.properties** の認証設定ファイルパスを起動パラメータに指定する必要があります。
ここでは、利用するOSごとの起動パラメータの設定方法を説明します。

for Windows

<%CASSANDRA_HOME%/bin/cassandra.bat>ファイルをエディタで開きます。
下記のサンプルを参考に、 **access.properties** と **passwd.properties** プロパティを追記します。

追記前

```
...
-XX:+UseCMSInitiatingOccupancyOnly^
-Dcom.sun.management.jmxremote.port=7199^
...
```

追記後

```
...
-XX:+UseCMSInitiatingOccupancyOnly^
-Daccess.properties=%CASSANDRA_HOME%/conf/access.properties^
-Dpasswd.properties=%CASSANDRA_HOME%/conf/passwd.properties^
-Dcom.sun.management.jmxremote.port=7199^
...
```

for Linux

</usr/local/cassandra/conf/cassandra-env.sh>ファイルをエディタで開きます。
下記のサンプルを参考に、 **access.properties** と **passwd.properties** を追記します。

追記前

```
...
# Prefer binding to IPv4 network intefaces (when net.ipv6.bindv6only=1). See
# http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6342561 (short version:
# comment out this entry to enable IPv6 support).
JVM_OPTS="$JVM_OPTS -Djava.net.preferIPv4Stack=true"

# jmx: metrics and administration interface
...
```

追記後

```
...
# Prefer binding to IPv4 network intefaces (when net.ipv6.bindv6only=1). See
# http://bugs.sun.com/bugdatabase/view_bug.do?bug_id=6342561 (short version:
# comment out this entry to enable IPv6 support).
JVM_OPTS="$JVM_OPTS -Djava.net.preferIPv4Stack=true"

# set authentication parameter properties path.
JVM_OPTS="$JVM_OPTS -Daccess.properties=$CASSANDRA_HOME/conf/access.properties"
JVM_OPTS="$JVM_OPTS -Dpasswd.properties=$CASSANDRA_HOME/conf/passwd.properties"

# jmx: metrics and administration interface
...
```

注意

- クラスタを構築し複数のCassandraで認証設定を有効にする場合、クラスタ内の全ノードに同一の設定が必要です。
- 認証設定の有効/無効はクラスタ毎に適用されるため、同一クラスタ内のすべてのキースペースが認証設定の対象です。
- 認証設定を有効にした場合の接続ユーザ名、および、パスワードは、クラスタ全体で同一の値を設定する必要があります。
  キースペース毎に変更することはできません。

設定ファイルの解説

ここでは設定ファイルの設定値について説明します。

接続認証設定 (passwd.properties)

標準の **passwd.properties** には、管理者を想定した **admin** と参照のみユーザを想定した **user** を設定してあります。

以下にコメント行を省いた設定値を記載します。

```
admin=admin_pwd
user=user_pwd
```

設定値のフォーマットは、「%接続ユーザ名%=%パスワード%」です。

アクセス権設定 (access.properties)

設定値のフォーマットは、「%アクセス権%=%接続ユーザ名%」です。
アクセス権には、以下の３つが定義されています。

1. キースペースの編集権限
   <modify-keyspaces>：キースペースの作成・削除が可能な権限です。

   > **注意**
   >
   > Apache Cassandra 1.1.12では、modify-keyspacesによるキースペース編集権限は設定できなくなりました。
   > 代替の設定方法として、キースペースごとに下記 **キースペースへのアクセス権限** を設定してください。

2. キースペースへのアクセス権限
   キースペース名.<rw>：キースペース内のカラムファミリを閲覧および変更が可能な権限です。
   キースペース名.<ro>：キースペース内のカラムファミリの閲覧のみが可能な権限です。

3. キースペース内のカラムファミリへのアクセス権限
   キースペース名.カラムファミリ名.<rw>：キースペース内の特定カラムファミリ内のデータを閲覧および変更が可能な権限です。
   キースペース名.カラムファミリ名.<ro>：キースペース内の特定カラムファミリ内のデータの閲覧のみが可能な権限です。

   カラムファミリ名 に * を指定することで、全カラムファミリを指定することが可能です。
   今後の機能拡張にてカラムファミリが追加される可能性があるため、cassandra-cnofig.xmlに記述する接続ユーザには全カラムファミリの変更権限を設定してください。

標準の **access.properties** では、 **passwd.properties** で指定されたそれぞれのユーザに以下の権限が設定されています。

admin

- キースペース編集権限
- defaultキースペースのRead&Write権限
- defaultキースペースの全カラムファミリのRead&Write権限

user

- defaultキースペースのReadOnly権限
- defaultキースペース内の全カラムファミリのReadOnly権限

以下に一部のコメント行を省いた設定値を記載します。

```
# The magical '<modify-keyspaces>' property lists users who can modify the
# list of keyspaces: all users will be able to view the list of keyspaces.
<modify-keyspaces>=admin

# Access to Keyspace 'default'
default.<rw>=admin
default.<ro>=user

# Access to all ColumnFamily
default.*.<rw>=admin
default.*.<ro>=user
```

注意

- intra-mart Accel Platform 2014 Spring(Granada)以前のバージョンをお使いの場合
  - cassandra-config.xmlに設定する接続ユーザは、すべての変更権限を保持している必要があります。
  - cassandra-config.xmlを編集してキースペース名を変更した場合や、複数キースペースでの運用を行う場合には、キースペースおよびキースペース内のカラムファミリへのアクセス権の設定値を変更してください。
- intra-mart Accel Platform 2014 Spring(Granada)以降のバージョンをお使いの場合
  - テナント環境セットアップ時に設定するCassandra接続情報の認証ユーザ名は、すべての変更権限を保持している必要があります。
  - Cassandra接続情報を編集してキースペース名を変更した場合や、複数キースペースでの運用を行う場合には、キースペースおよびキースペース内のカラムファミリへのアクセス権の設定値を変更してください。

接続認証の確認方法

Cassandra への接続時に認証が必要となっていることを確認します。
ここでは、標準の設定ファイルにて認証設定を行うことを想定しています。

1. Cassandra を起動します。
2. Cassandra が起動していることを確認し、cassandra-cliを起動します。
   - Windowsの場合、<%CASSANDRA_HOME%/bin/cassandra-cli.bat>をダブルクリックします。
   - Linuxの場合、以下のコマンドで起動します。

     ```
     /usr/local/cassandra/bin/cassandra-cli
     ```
3. コンソールが立ち上がったら、以下のコマンドを入力ください。

   ```
   connect localhost/9160;
   ```

   認証情報を指定していないため、以下のメッセージが出力されて接続に失敗します。

   ```
   Login failure. Did you specify 'keyspace', 'username' and 'password'?
   ```

   ローカル環境の Cassandra に接続した場合、cassandra-cliの起動時にも上記メッセージが出力されます。

4. 以下のコマンドを入力し、cassandra-cliを一度終了します。

```
quit;
```

5. 次に、認証情報を指定しcassandra-cliを起動します。
   - Windowsの場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

```
%CASSANDRA_HOME%/bin/cassandra-cli -u admin -pw admin_pwd
```

   - Linuxの場合、以下のコマンドで起動します。

```
/usr/local/cassandra/bin/cassandra-cli -u admin -pw admin_pwd
```

> **コラム**
>
> **-u** は、接続ユーザ名を指定するパラメータです。
> **-pw** は、パスワードを指定するパラメータです。

> **注意**
>
> 認証クラスに PasswordAuthenticator を利用している場合は、デフォルトの管理ユーザとして **cassandra** ユーザが自動で指定されているため、 確認するコマンドは以下です。
>
> Windows
>
> ```
> %CASSANDRA_HOME%/bin/cassandra-cli -u cassandra -pw cassandra
> ```
>
> Linux
>
> ```
> /usr/local/cassandra/bin/cassandra-cli -u cassandra -pw cassandra
> ```

6. コンソールが立ち上がったら、以下のコマンドを入力してください。

```
connect localhost/9160;
```

以下のメッセージが表示され、接続に成功したことが確認できます。

```
Connected to: "IMBox Cluster" on localhost/9160
```

> **注意**
>
> intra-mart Accel Platform 2014 Spring(Granada)より、接続認証を利用する場合、あらかじめキースペースを作成する必要があります。
> キースペースの作成、追加に関しては「*Cassandra の操作*」の「*キースペースの作成方法*」、「*キースペースの作成方法（認証設定ありの場合）*」を参照してください。

Cassandra の操作

ここでは Cassandra の操作方法を説明します。
本書では以下の説明は【表1. Cassandra接続情報設定例】を元に記載しています。

- 【表1. Cassandra接続情報設定例】

クラスタ名	IMBox Cluster
ホストのIPアドレス	localhost
ホストのポート番号	9160
認証設定の有効化	false
キースペース	default

キースペースの作成方法

1. Cassandra を起動します。
2. Cassandra が起動していることを確認し、cassandra-cliを起動します。
   - Windowsの場合、<%CASSANDRA_HOME%/bin/cassandra-cli.bat>をダブルクリックします。
   - Linuxの場合、以下のコマンドで起動します。

```
/usr/local/cassandra/bin/cassandra-cli
```

3. コンソールが立ち上がったら、以下のコマンドを入力してください。

```
connect localhost/9160;
```

4. 接続に成功したことを確認し、以下のコマンドを入力してください。

```
create keyspace default;
```

5. 以下のメッセージが表示され、キースペース「default」の作成に成功したことが確認できます。

```
... schemas agree across the cluster
```

キースペースの作成方法（認証設定ありの場合）

Cassandra に以下の条件でキースペースを作成する手順を紹介します。

- 接続ユーザ名：aoyagi
- パスワード：aoyagi_pwd
- キースペース名：imbox_keyspace

1. 接続認証設定 (passwd.properties)にパスワードを追加します。

```
admin=admin_pwd
user=user_pwd
aoyagi=aoyagi_pwd
```

2. アクセス権設定 (access.properties)にaoyagiがアクセスできるキースペースを追加します。

```
# The magical '<modify-keyspaces>' property lists users who can modify the
# list of keyspaces: all users will be able to view the list of keyspaces.
<modify-keyspaces>=admin

# Access to Keyspace 'default'
default.<rw>=admin
default.<ro>=user
# Access to all ColumnFamily
default.*.<rw>=admin
default.*.<ro>=user

# Access to Keyspace 'imbox_keyspace'
imbox_keyspace.<rw>=aoyagi
imbox_keyspace.<ro>=user
# Access to all ColumnFamily
imbox_keyspace.*.<rw>=aoyagi
imbox_keyspace.*.<ro>=user
```

> **注意**
>
> 認証設定を使用している場合は、access.propertiesで設定したキースペース以外のキースペースは作成することができません。

3. Cassandra を起動します。
4. 次に、認証情報を指定しcassandra-cliを起動します。
   - Windowsの場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

```
%CASSANDRA_HOME%/bin/cassandra-cli -u aoyagi -pw aoyagi_pwd
```

   - Linuxの場合、以下のコマンドで起動します。

```
/usr/local/cassandra/bin/cassandra-cli -u aoyagi -pw aoyagi_pwd
```

> **コラム**
>
> 認証情報の指定方法は、「*接続認証の確認方法*」を参照してください。

5. 接続に成功したことを確認し、以下のコマンドを入力し、「imbox_keyspace」を作成します。

```
create keyspace imbox_keyspace;
```

6. 以下のメッセージが表示され、キースペース「imbox_keyspace」が追加されたことが確認できます。

```
... schemas agree across the cluster
```

> **コラム**
>
> クラスタを構築し複数のCassandraにキースペースを作成したい場合、シードノードのCassandraにキースペースを作成後にnodetool repairコマンドを実行することで他のノードにもキースペースが作成されるため、すべてのノードでキースペースを作成する必要はありません。

キースペースの削除方法

1. Cassandra を起動します。
2. Cassandra が起動していることを確認し、cassandra-cliを起動します。
   cassandra-cliの起動方法は以下の通りです。
   - Windowsの場合、<%CASSANDRA_HOME%/bin/cassandra-cli.bat>をダブルクリックします。
   - Linuxの場合、以下のコマンドで起動します。

```
/usr/local/cassandra/bin/cassandra-cli
```

3. コンソールが立ち上がったら、以下のコマンドを入力してください。

```
connect localhost/9160;
```

4. コンソールに「Connected to: “IMBox Cluster” on localhost/9160」と表示されるので、以下のコマンドを入力してください。

```
drop keyspace default;
```

5. コンソールに「... schemas agree across the cluster」が表示されたら削除完了です。

> **コラム**
>
> 上記の手順にてキースペースの削除を行っても、保存されたデータは残った状態となります。
> 削除したキースペースのデータを完全に削除する場合は、「data_file_directories」プロパティで指定した保存先ディレクトリ内にある、該当キースペース名のディレクトリを削除する必要があります。
>
> - Windowsの場合
>
> ```
> C:/cassandra/data/default
> ```
>
> - Linuxの場合
>
> ```
> /var/lib/cassandra/data/default
> ```

フラッシュ

Cassandra では、データは、memtableというメモリ上に保存されています。

cassandra.yamlの「data_file_directories」プロパティで設定したファイルシステム上のSSTableへデータを書き込むには、nodetoolのflushコマンドを実行する必要があります。
冗長化していない場合などにmemtableのデータが、SSTableに書かれる前にサーバが停止した場合にデータが損失する恐れがありますので、定期的にフラッシュすることを推奨します。

- Windowsの場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

```
%CASSANDRA_HOME%/bin/nodetool -h [実行したいノードのIPアドレス] flush
```

- Linuxの場合、以下のコマンドを実行します。

```
# /usr/local/cassandra/bin/nodetool -h [実行したいノードのIPアドレス] flush
```

実行すると、Cassandra 上で、以下のようなログが出力されます。

```
INFO 15:12:11,104 Enqueuing flush of Memtable-Versions@2895088(83/103 serialized/live bytes, 3 ops)
INFO 15:12:11,105 Writing Memtable-Versions@2895088(83/103 serialized/live bytes, 3 ops)
INFO 15:12:11,258 Completed flushing %CASSANDRA_HOME%\data\system\Versions\system-Versions-he-3-Data.db (247 bytes) for commitlog position ReplayPosition(segmentId=42805933509718, position=544)
```

スナップショット

Cassandra のデータをバックアップするには、nodetoolのsnapshotコマンドでスナップショットを取得することが必要です。
ただし、 Cassandra のスナップショットは、SSTableというファイルのみをコピーするだけですので、スナップショットを実行する前に、フラッシュコマンドを実行することが必須です。
スナップショットの手順は以下の通りです。

- Windowsの場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

```
%CASSANDRA_HOME%/bin/nodetool -h [実行したいノードのIPアドレス] flush
%CASSANDRA_HOME%/bin/nodetool -h [実行したいノードのIPアドレス] snapshot -t （任意のスナップショット名）
```

- Linuxの場合、以下のコマンドを実行します。

```
# /usr/local/cassandra/bin/nodetool -h [実行したいノードのIPアドレス] flush
# /usr/local/cassandra/bin/nodetool -h [実行したいノードのIPアドレス] snapshot -t （任意のスナップショット名）
```

コラム

スナップショットされたデータは、conf/cassandra.yamlの「data_file_directories」プロパティで指定されたディレクトリの配下に格納されます。詳細は以下を参照してください。
以下、%CASSANDRA_HOME%/conf/cassandra.yamlの「data_file_directories」が/var/lib/cassandra/data/ の場合

Cassandra のシステム情報	/var/lib/cassandra/data/system/（カラムファミリー名）/snapshot/（スナップショット名）
IMBoxのデータ	/var/lib/cassandra/data/default/（カラムファミリー名）/snapshot/（スナップショット名）

コラム

スナップショットの削除

- snapshot ディレクトリすべてを削除します。

```
# /usr/local/cassandra/bin/nodetool -h [実行したいノードのIPアドレス] clearsnapshot
```

- 任意のsnapshotを削除します。

```
# /usr/local/cassandra/bin/nodetool -h [実行したいノードのIPアドレス] clearsnapshot -t （任意のスナップショット名）
```

注意

Windows環境にて clearsnapshot コマンドを実行した場合、削除対象のファイルがCassandraのプロセスに握られているため削除処理に失敗します。
以下のコマンドを実行することでSSTableが再生成され、その過程でファイルへの参照が外れます。

```
nodetool upgradesstables
```

その後 clearsnapshot を実行することで、スナップショットが正常に削除されます。

なお、Linux環境の場合には上記の問題は発生しません。

スナップショットデータによる復旧（リストア）

スナップショットによりデータを退避していた場合、以下の手順で、スナップショット時点に復旧（リストア）することが可能です。

- Windowsの場合
  1. Cassandra を停止
  2. commitlog（例：%CASSANDRA_HOME%/commit_log）のファイルを削除
  3. SSTableを削除
     - 実データ（例：C:/cassandra/data/default/（カラムファミリー名）/ の各dbファイル）を削除
     - システム情報（例：C:/cassandra/data/system/（カラムファミリー名）/ の各dbファイル）を削除
  4. スナップショットの各dbファイルをコピー
     - スナップショットで取得した実データをC:/cassandra/data/default/（カラムファミリー名）にコピー
     - スナップショットで取得したシステム情報をC:/cassandra/data/system/（カラムファミリー名）にコピー
  5. Cassandra を起動
- Linuxの場合
  1. Cassandra を停止
  2. commitlog（例：/var/lib/cassandra/commit_log）のファイルを削除

3. SSTableを削除
   - 実データ（例：/var/lib/cassandra/data/default/（カラムファミリー名）/ の各dbファイル）を削除
   - システム情報（例：/var/lib/cassandra/data/system/（カラムファミリー名）/ の各dbファイル）を削除
4. スナップショットの各dbファイルをコピー
   - スナップショットで取得した実データを/var/lib/cassandra/data/default/（カラムファミリー名）にコピー
   - スナップショットで取得したシステム情報を/var/lib/cassandra/data/system/（カラムファミリー名）にコピー
5. Cassandra を起動

Cassandra の運用

ここでは Cassandra の運用方法を説明します。

項目

状態変化のタイミングとリスク

Cassandra の運用状態が変化するタイミングと、その時のリスクについて考察します。

計画停止

手動やサービスによって、意図したタイミングで停止した場合を想定します。
また前提として、Cassandraを停止するタイミングではApplicationServerからの更新要求がないものとします。

コラム

対応例

- 停止前にApplicationServerを停止する。
- nodetool drain コマンド を実行し、書き込み要求を受け付けなくする。

drain の実行時には、同時にフラッシュも実行されます。

- フラッシュに成功した場合
  通常期待される状態です。フラッシュが実行されているためデータ損失はなく、当然ながら対策を検討する必要はありません。
- フラッシュに失敗した場合
  通常は計画停止した場合はフラッシュが実行され停止時のデータが担保されますが、停止時にフラッシュが失敗する場合があります。
  フラッシュが正常に実行されていないため、メモリ上に存在していたデータが失われます。

異常停止

Cassandraが何らかの原因により、意図せず停止した場合を想定します。

> **コラム**
>
> **以下のような状況が想定されます。**
>
> - Cassandra内でサービスの継続が不可能な例外（OutOfMemoryError）が発生した場合
> - Cassandraのプロセスが強制停止された場合
> - Cassandra起動中にOSが停止した場合

- フラッシュが実行されていた場合
  異常停止前にフラッシュが実行される場合があります。
  フラッシュが実行されているためデータ損失はなく、当然ながら対策を検討する必要はありません。
- フラッシュが実行されない、または失敗した場合
  異常停止前には通常フラッシュが実行されません。また、実行されても失敗する可能性があります。
  フラッシュが正常に実行されていないため、メモリ上に存在していたデータが失われます。

> **コラム**
>
> 計画停止、異常停止に関わらず、フラッシュ失敗によるデータ損失のリスクが存在します。

ネットワークの分断

Cassandraのノードがネットワークから分断された場合は、データの損失が発生する事はありません。
ただし、構成の状態によってはサービスが利用できなくなります。

- 単一ノードにて構築されている場合
  Cassandraが復旧するまでIMBoxは利用できなくなります。

> **コラム**
>
> 単一ノードにて運用する場合は、ネットワークが分断した時点で読み込み/書き込みが行えなくなるため、データ損失は発生しません。

- クラスタ構成で構築されている場合
  ネットワークから分断されたノード数により状況が変化します。
  - サービス継続が可能な台数の場合
    例えば「ノード数が３台（レプリケーションファクタが３）」で運用中に１台が分断された場合、サービスは他の２台のノードで正常に行われます。
    分断されたノードが再度ネットワークに接続されたタイミングで不足分のデータがレプリケート（コピー）され、通常の運用状態に復旧します。
  - サービス継続が不可能な台数の場合
    Cassandraが復旧するまでIMBoxは利用できなくなります。
    例えば「ノード数が３台、レプリケーションファクタが３」の場合に２台が分断された場合、サービスが利用不可となります。
    サービスが利用不可になり読み込み/書き込みが行えなくなるため、データ損失は発生しません。

リスク対策

前項より、基本的にフラッシュの失敗または未実施の場合について対策を取る必要があります。
対応策について以下にて幾つかの案を考察します。

計画停止前のフラッシュ実行

計画停止前に１度フラッシュ処理を実施します。
これにより、計画停止時に実行されるフラッシュの失敗に対するリスクを回避可能です。

定期的なフラッシュ実行

Cassandra起動時にcronなどで定期的なフラッシュを実行し、異常停止などによってデータが損失した場合でも、最後にフラッシュを実行した時点までの内容を担保します。

> **コラム**
>
> この対策はあくまでもデータ損失の確率や損失量を軽減する対策になります。
> フラッシュ に関してはこちらを参照してください。

> **コラム**
>
> フラッシュ実行の頻度については、運用設計にて要求される条件との調整が必要です。
> 検討が必要な項目として、以下が考えられます。
>
> - データ損失の可能性が許容される最大の時間
> - フラッシュの定期実行によるI/O負荷増加によるサービスのレスポンスやサーバ負荷への影響

クラスタ構成の構築

Cassandraのクラスタを構築して運用することで、データ損失およびサービス停止の可能性を大きく軽減する事が可能です。
このとき、クラスタを構成するノード数は３ノード以上を強く推奨します。

- ノード数が２台の場合
  クラスタを構成するノードが２ノードであった場合には、いずれか１ノードが停止した時点でサービスの継続（データの読み込みおよび書き込み）が不可能となります。
  ただし、サービス継続不可能となった時点でデータの書き込みは行われないため、データの損失は発生しません。
- ノード数が３台（またはそれ以上）の場合
  クラスタを構成するノードが３ノード（およびレプリケーションファクタが３）であった場合、いずれか１ノードが停止してもサービスは継続可能です。
  停止したノードが復旧したタイミングで、停止していたノードに対して不足しているデータがレプリケート（コピー）され、レプリケート終了時に通常の運用状態に復旧します。
  ３ノードのうち２ノードが同時に停止した場合にはサービスが継続不可能となりますので、１ノードが停止した段階で復旧処置を行う必要があります。
  ただし、サービス継続不可能となった時点でデータの書き込みは行われないため、データの損失は発生しません。

まとめ

構成毎に有効な対策についてまとめます。

Cassandraを単一ノードにて運用する場合

Cassandraを単一ノードにて運用する場合には、異常停止の発生時にデータ損失を完全に防ぐ事は不可能です。
異常停止が発生場合の対策として、以下が有効だと考えられます。

- 停止前のフラッシュ実行
- 定期的なフラッシュ実行

> **コラム**
>
> **参考情報**
>
> 弊社の社内システムにおけるCassandraは単一ノードで運用されており、15分間隔でフラッシュを実行しています。
> ２年ほど運用しておりますが、過去に４回ほど直近15分（最大）の投稿データが損失する事象が発生しております。
> ※ 弊社システムは運用試験環境を兼ねているためリリース前（RC版）を利用しており、通常よりもエラー発生確率が高くなっております。

Cassandraをクラスタ構成にて運用する場合

クラスタ構成にて運用する場合には、それだけでデータ損失の発生率を大きく軽減する事が可能です。
また、メンテナンスなどでクラスタ全体を停止させる必要がある場合、以下の対策を行うことでデータ損失のリスクを軽減可能です。

- 停止前のフラッシュ実行

データの復元

データの損失が発生してしまった場合に復元可能なデータは以下になります。

復元可能なデータ

IM共通マスタから同期して書き込まれた「ユーザ」や「組織」の情報

> **コラム**
>
> 専用のジョブ（ジョブネット）を実行することで復元可能です。
> 詳細については、「 IMBox 仕様書 」-「 ジョブスケジューラ 」を参照してください。

復元不可能なデータ

IMBox上で更新された「投稿内容」や「新規に作成したグループ」などはCassandra上のみに保存されております。
そのため、データ損失が発生した場合には復元することはできません。

> **コラム**
>
> 添付ファイルについてはファイルシステム（PublicStorage）上に保存されているため、システム管理者による復旧が可能です。
> 添付ファイルの保存先については、「 IMBox 仕様書 」-「 添付ファイル保存先 」を参照してください。

時刻の調整

Cassandra に格納される全てのデータは時刻を持ちます、この時刻は主にクラスタリング環境においてデータの矛盾を解決する為に利用されます。
その為、 Cassandra を運用しているサーバ上の時計を変更する等の操作をした場合データの不整合が発生しデータが破損する可能性があります。
また、 Cassandra に格納するデータの時刻は Cassandra が実行されているサーバ上ではなく、 Cassandra へ接続するアプリケーション側にて設定する時刻が利用されます。
その為、 intra-mart Accel Platform が動作する環境、および、 Cassandra が実行するサーバ上の時計は全て時刻が合っている必要があります。
時刻の統一を行うため、ntpサーバの利用を検討してください。

タイムアウト

CassandraサーバのDisk I/Oの速度によっては、データの読み込み時、および、書き込み時にタイムアウトが発生する可能性があります。
タイムアウトの発生頻度を下げるためには、cassandra.yamlのrpc_timeout_in_msの設定値を調整してください。
デフォルトでは10000(10秒)となっています。
間隔が長いと応答速度が遅くなる可能性がございます。
お客様の環境や運用に合わせ、必要に応じて間隔を設定してください。

Cassandra のバージョンアップ

ここでは Cassandra 、および、 Cassandra が利用する JDK のバージョンアップの方法を紹介します。
また、Cassandra 1.1.12 で運用を行っている環境において Java 11 対応版 Cassandra 1.1.12 への移行を行う方法について紹介します。

> **コラム**
>
> Cassandra 1.1.12 にていくつかの不具合（運用中にデータが破損して再起動に失敗する恐れのある不具合など）が修正されているため、 Cassandra 1.1.4 で既に運用している場合はバージョンアップを推奨します。
> 2013 Winter(Felicia)以前にリリースされた intra-mart Accel Platform においても、 Cassandra 1.1.12 での動作を確認しております。

項目

Cassandra のバージョンアップ手順

Cassandra 1.1.4 から Cassandra 1.1.12 にバージョンアップを行う場合、以下の手順が必要です。
バージョンアップ前には、 *スナップショット* を取得しておくことを推奨します。

1. Cassandra 1.1.4 を起動します。
2. Cassandra 1.1.4 を nodetool コマンドの drain で書き込み操作の受け入れを停止します。
   - Windows の場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

     ```
     %CASSANDRA_HOME%/bin/nodetool drain
     ```

   - Linux の場合、以下のコマンドを実行します。

     ```
     # /usr/local/cassandra/bin/nodetool drain
     ```

3. 起動中のCassandra に「DRAINED」が表示され、 Cassandra が停止します。
4. Cassandra 1.1.12 の<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルにある、以下のプロパティを Cassandra 1.1.4 で設定しているディレクトリとは異なるディレクトリに設定します。
   - 「data_file_directories」プロパティ
   - 「commitlog_directory」プロパティ
   - 「saved_caches_directory」プロパティ

   > **注意**
   >
   > Cassandra 1.1.12 のcommitlog_directory、saved_caches_directoryを Cassandra 1.1.4 と同一ディレクトリに設定した場合、データ不整合が発生する恐れがあるため、commitlog_directory、saved_caches_directoryは別ディレクトリを設定してください。

5. Cassandra 1.1.12 で新しく設定した「data_file_directories」プロパティの保存先ディレクトリ内に、

Cassandra 1.1.4 で設定していた「data_file_directories」プロパティの保存先ディレクトリ内をコピーします。

> **注意**
>
> Cassandra 1.1.4 のスナップショットは、Cassandra 1.1.12 で設定する「data_file_directories」プロパティのディレクトリ内にコピーする必要はありません。
> スナップショットの格納場所、および、削除方法についての詳細は「*スナップショット*」を参照してください。

6. Cassandra 1.1.12 を起動し、nodetool コマンドの upgradesstables でデータの再構築を行います。
   - Windowsの場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

     ```
     %CASSANDRA_HOME%/bin/nodetool upgradesstables
     ```

   - Linuxの場合、以下のコマンドを実行します。

     ```
     # /usr/local/cassandra/bin/nodetool upgradesstables
     ```

7. Cassandra の再構築が正常終了した時点で、バージョンアップの手順は完了です。

> **コラム**
>
> - Cassandra をクラスタ構築している場合には、各ノード毎に再構築を行ってください。
> - Cassandra 1.1.4 で *Cassandra への接続認証設定*、*Windows サービスへの登録・削除* を行っている場合、バージョンアップ後に Cassandra 1.1.12 で再度設定を行う必要があります。

JDK のバージョンアップ手順

Cassandra が利用する JDK のバージョンアップを行う場合、以下の手順が必要です。
バージョンアップ前には、 *スナップショット* を取得しておくことを推奨します。

1. Cassandra を起動します。
2. Cassandra を nodetool コマンドの drain で書き込み操作の受け入れを停止します。
   - Windows の場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

     ```
     %CASSANDRA_HOME%/bin/nodetool drain
     ```

   - Linux の場合、以下のコマンドを実行します。

     ```
     # /usr/local/cassandra/bin/nodetool drain
     ```

3. 起動中の Cassandra に「DRAINED」が表示され、Cassandra が停止します。
4. JDK のバージョンアップを行います。
5. 再度 Cassandra を起動して、バージョンアップの手順は完了です。

コラム

- Cassandra をクラスタ構築している場合には、各ノード毎にバージョンアップを行ってください。

Java 11 対応版 Cassandra 1.1.12 への移行

Java 8 を利用して Cassandra 1.1.12 を利用しており、Java 11 での運用に切り替える場合、以下の手順が必要です。
Java 11 対応版 Cassandra 1.1.12 の取得方法については *Java 11 対応版 Cassandra 1.1.12 の取得* を参照してください。
バージョンアップ前には、 *スナップショット* を取得しておくことを推奨します。

1. Java 8 で運用中の Cassandra 1.1.12 を起動します。
2. Java 8 で運用中の Cassandra 1.1.12 を nodetool コマンドの drain で書き込み操作の受け入れを停止します。
   - Windows の場合、コマンドプロンプトを起動し、以下のコマンドを実行します。

     ```
     %CASSANDRA_HOME%/bin/nodetool drain
     ```

   - Linux の場合、以下のコマンドを実行します。

     ```
     # /usr/local/cassandra/bin/nodetool drain
     ```

3. 起動中の Cassandra に「DRAINED」が表示され、Cassandra が停止します。
4. Java 8 で運用中の <%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルにある、以下のプロパティが指し示すディレクトリの内容を、Java 11 対応版の<%CASSANDRA_HOME%/conf/cassandra.yaml>ファイルにある、以下のプロパティが指し示すディレクトリにそれぞれコピーします。または、同じディレクトリを指し示すようにプロパティを変更します。
   - 「data_file_directories」プロパティ
   - 「commitlog_directory」プロパティ
   - 「saved_caches_directory」プロパティ
5. JDK のバージョンアップを行います。
6. Java 11 対応版 Cassandra 1.1.12 を起動して、バージョンアップの手順は完了です。

コラム

- Cassandra をクラスタ構築している場合には、各ノード毎に再構築を行ってください。
- Java 8 で運用していた Cassandra 1.1.12 で *Cassandra への接続認証設定* 、*Windows サービスへの登録・削除* を行っている場合、Java 11 対応版 Cassandra 1.1.12 への移行後に再度設定を行う必要があります。

Cassandra の参考情報

ここでは Cassandra についての参考情報をご紹介します。

cassandra.yaml 主な項目一覧

項目名	初期値	説明
cluster_name	Test Cluster	クラスタの名称。
initial_token	<無記述>	Tokenを指定し、クラスタに参加させる場合はここに記述を行う。 ここが空の場合、最も負荷の高いノードが担当する範囲を分担 する様なTokenが割り当てられる。 完全新規ノードなど負荷情報がない場合はランダムに割り当てられる。
hinted_handoff_enabled	true	詳細はCassandra公式wikiのヒンテッドハンドオフを参照。
authenticator	org.apache.cassandra.auth.AllowAllAuthenticator	Cassandraへアクセスする際の認証についての設定項目。 初期設定では認証を行わない。 詳細は 「*Cassandra への接続認証設定*」を参照。
authority	org.apache.cassandra.auth.AllowAllAuthority	Cassandraへアクセスする際のユーザ承認についての設定項目。 初期設定ではすべての接続を許可する。 詳細は 「*Cassandra への接続認証設定*」を参照。

partitioner	org.apache.cassandra.dht.RandomPartitioner	クラスタ内のノードに対してどの様にデータを配置するかの設定。
data_file_directories	/var/lib/cassandra/data	実データ保存場所。
commitlog_directory	/var/lib/cassandra/commitlog	CommitLog保存場所。
saved_caches_directory	/var/lib/cassandra/saved_caches	キャッシュデータ保存場所。
commitlog_sync	periodic	CommitLogの保存の方法。一括の書き込むか少しずつ書き込むかを「periodic/batch」より選択を行う。
commitlog_sync_period_in_ms	10000	ミリ秒にて指定。指定した期間毎にCommitLogが書き込まれる
seeds	127.0.0.1	クラスタに参加させる場合、自身のIPとクラスタ内の他ノードの 指定を行う。
disk_access_mode	auto	mmaped I/Oの利用設定。64bitのJVMで利用する際に有効。 初期値のautoでの利用が推奨。 （Cassandra1.1.4まで）
concurrent_reads	32	同時に処理する読込処理の数。1CPUコア数x2が推奨。
concurrent_writes	32	同時に処理する読込処理の数。
storage_port	7000	ノード間でデータをやり取りする際に利用するポート番号を指定。

listen_address	localhost	他ノードと通信する際に利用するアドレス。 ノードが複数のネットワークに参加している利用させたいネットワーク側のアドレスの記述を行う。
rpc_address	localhost	Thriftを使って外部との接続を許可するIPアドレス。
rpc_port	9160	Thriftを使って外部との接続に利用するポート番号。
rpc_keepalive	true	RPC接続においてKeepAlive属性を利用するか否かの選択を行う
thrift_framed_transport_size_in_mb	15	Thriftで使うフレームバッファのサイズをMB単位で指定。
snapshot_before_compaction	false	Compaction前にスナップショットをとるか否か。
in_memory_compaction_limit_in_mb	64	メモリに圧縮されているカラムのサイズ制限。 推奨値は、使用可能なJavaヒープ·サイズの5から10パーセント
rpc_timeout_in_ms	10000	コマンドのエラーを返すまでのタイムアウト値。
dynamic_snitch	true	Compaction実施中等の理由により応答速度が低下したノード 以外からデータを読み込ませる。 （Cassandra1.1.4まで）
request_scheduler	org.apache.cassandra.scheduler.NoScheduler	リクエスト発行回数に制限を設ける際に利用するアルゴリズムを指定。

外部サイト

サイト名	URL
Apache Cassandra	https://cwiki.apache.org/confluence/display/CASSANDRA2 (English)
Cassandra Wiki	https://cwiki.apache.org/confluence/display/CASSANDRA2/FrontPage+JP （日本語） https://cwiki.apache.org/confluence/display/CASSANDRA2 (English) https://cwiki.apache.org/confluence/display/CASSANDRA2/FrontPage+ZH （中文）
Cassandraのハードウェアサイジング	https://cwiki.apache.org/confluence/display/CASSANDRA2/CassandraHardware+JP （日本語） https://cwiki.apache.org/confluence/display/CASSANDRA2/CassandraHardware (English) https://cwiki.apache.org/confluence/display/CASSANDRA2/CassandraHardware+ZH （中文）
datastaxのApache Cassandra 1.1 ドキュメント	http://docs.datastax.com/en/archived/cassandra/1.1/docs/ (English)
Planet Cassandra	http://www.planetcassandra.org/ (English) http://www.planetcassandra.org/what-is-apache-cassandra-jp/ （日本語）