JBoss Enterprise Application Platform Common Criteria Certification 7.2.3

Common Criteria 設定ガイド

ガイド

Last Updated: 2021-11-25

JBoss Enterprise Application Platform Common Criteria Certification 7.2.3 Common Criteria 設定ガイド

ガイド

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

目次

第1章 はじめに

1.1. 本書の目的

本ガイドでは、Red Hat JBoss Enterprise Application Platform 7 を認定された Common Criteria 準拠のセキュアな設定で使用する管理者およびアプリケーション開発者へのガイダンスを提供します。

本ガイドは、高レベルで最も重要な問題に対処するための自己完結を目的としており、詳細情報が必要な既存のドキュメントを参照します。本書のリーダーには、共通基準に関する知識は必要ありません。

JBoss EAP 7 は、Common Criteria 認定の Target of Evaluation (TOE) として本書の対象となります。JBoss EAP 7 は、Assurance EAL4 レベルの Common Criteria バージョン 3.1 で評価されています。これは、この製品が徹底的にテストされたことを保証します。

本章では、Common Criteria の認定と本書の構造の概要を説明します。

評価済み設定の 要件には、認定製品をデプロイするための要件が含まれます。

パッケージのダウンロードおよび検証には、JBoss EAP 7 の認定バージョンを使用していることを確認するために必要な手順が含まれています。

JBoss EAP 7 の起動および停止には、サーバーの起動および停止方法と、さまざまな操作モードの手順が記載されています。

Common Criteria Certified System の開発ガイド には、JBoss EAP 7 のアプリケーションを作成する開発者向けのガイドラインが含まれています。

セキュリティー機能の概要には、JBoss EAP 7 のセキュリティー実装および使用方法の制限の詳細が含まれています。

本ガイドの情報と他の製品ドキュメントとの間に不一致がある場合、本ガイドは優先されます。JBoss EAP 7 の評価される設定の要件に対応します。

1.2. JBOSS EAP 固有の規則

本ガイドの **EAP_HOME** すべてのインスタンスは、JBoss EAP のルートインストールディレクトリーを参照します。たとえば、ZIP インストールパッケージを使用して JBoss EAP バイナリーを Linux **/home/USER** ディレクトリーに抽出した場合、EAP_HOME は **/home/USER/jboss-eap-[version]/** ディレクトリーを参照します。

詳細は、**インストールガイド** の**EAP_HOME の使用**のトピックを参照してください。

1.3. COMMON CRITERIA コンプライアントシステムとは?

Common Criteria または CC として知られる Common Criteria for Information Technology Security Evaluation は、IT 製品のセキュリティープロパティーに対する独立した評価の基盤として使用される国際認識された規格 (ISO/IEC 15408) です。

共通基準は、事前定義されたレベルに対する製品の不適切なセキュリティー保証をコンシューマーに提供します。これらのレベルは EAL1 から EAL7 まであります。各レベルは、テストの証明に対して開発者への要求が高まり、さらにコンシューマー向け製品の保証が高まります。

Common Criteria Recognition Arrangement (CCRA) 下で、メンバーは、CCRA に記載されている用語に従って、参加者を承認する証明書によって生成された共通基準証明書を認識することに同意します。現在、CCRA は 20 人以上のメンバーで構成されています。メンバーは、Australian、Sustralian、

Sustralian、Sustral、Setron、Setron、Hare、Hare、Han、Han、Sustral、Sustral、Sustral、米国、その他多くのメンバーが構成されています。新しいメンバーが、まもなく参加することが予想されます。

システムは、評価され認定された設定と一致する場合は、**CC に準拠** していると見なされます。これは、ハードウェアおよびソフトウェアに関するさまざまな要件、および運用環境、ユーザー、および継続中の操作手順に関する要件を意味します。

Common Criteria の詳細は Common Criteria Portal を参照してください。

1.4. 認定ドキュメント

Common Criteria で評価される設定で JBoss EAP 7 をインストール、設定、および操作する場合は、この Common Criteria の認定での使用を承認された製品ドキュメントのみを参照する必要があります。

製品ドキュメントバンドルは、Red Hat カスタマーポータルから 2 つの認定形式から入手できます。

- PDF ドキュメントバンドル
- オンライン

本ガイドの JBoss EAP ドキュメントへの参照はすべて、認定形式に含まれるガイドを参照してください。

警告

評価される設定を実行している場合は、JBoss EAP 7 のドキュメント Common Criteria バージョン ***のみ*** を参照する必要があります。標準の製品ドキュメントバージョンには、設定認定の違反を招く可能性のある情報が含まれている場合があります。

第2章 評価済み設定の要件

2.1. ソフトウェア要件

2.1.1. オペレーティングシステム

以下のテスト済みの構成で幅広いプラットフォームテストを実施しています。

オペレーティングシステム	Oracle JDK 1.8.0	IBMJDK 1.8.0	OpenJDK 1.8.0	Oracle JDK 11	OpenJDK 11
Red Hat Enterprise Linux 6 x86_64	X	X	X	X	
Red Hat Enterprise Linux 7 x86_64	X	X	X	X	X
Solaris 11 x86_64	X				
Solaris 11 SPARC64	X			X	
Windows Server 2012 R2 Standard x86_64	X		X	X	X
Windows Server 2016 x86_64	X		X	X	X

2.1.2. データベース JDBC ドライバー

JBoss EAP 7 は以下のリレーショナルデータベースシステムと評価されます。特定のドライバーバージョンを持つこれらのデータベースシステムのみが JBoss EAP 7 での使用が許可されます。

表2.1 以下のデータベースおよびデータベースドライバーのみがサポートされます。

データベース	JDBC ドライバーバージョン
IBM DB2 Enterprise e11.1 (FP1 11.1.1.1)	IBM DB2 JDBC Universal Driver Architecture 4.24.92
Oracle 12cR1 RAC (12.1.0.2.0)	Oracle JDBC Driver v12.2.0.1 (ojdbc8.jar)
MySQL 5.7 (5.7.17)	MySQL Connector/J 8.0.12
MariaDB 10.1.19	MariaDB Connector/J 2.2.4
MariaDB Galera Cluster 10.1.19	MariaDB Connector/J 2.2.4
Microsoft SQL Server 2016 SP1	Microsoft JDBC Drivers 6.4.0
PostgreSQL 10.1	JDBC4 Postgresql Driver、Version 42.2.2

データベース	JDBC ドライバーバージョン
Enterprise DB Postgres Plus Advanced Server 10.1 (10.1.5)	Postgres Plus Advanced Server Driver 10.1 (10.0.0.1)
Sybase ASE 16.0 (SP02)	JDBC™/16.0 GA (Build 27008)/P/EBF22326

JBoss EAP 7 で各データベースを設定する方法は、データベースの設定を参照してください。

2.1.3. LDAP サーバー

JBoss EAP 7 は以下の LDAP サーバーと評価されます。これらの LDAP サーバーのみが JBoss EAP 7 での使用が許可されます。

- Red Hat Directory Server 10.1
- Red Hat Directory Server 10.0
- Microsoft Active Directory 2012 R2
- Microsoft Active Directory 2016

2.2. 物理的な要件

JBoss EAP 7 を実行するハードウェアおよびソフトウェア、およびセキュリティーポリシーの適用に不可欠なソフトウェアは、内部または外部に関係なく、承認者が変更から保護する必要があります。JBoss EAP 7 ソフトウェアを実行しているハードウェアに物理的にアクセスしないように、承認されていないユーザーが JBoss EAP 7 ソフトウェアを実行するハードウェアに物理的にアクセスしないようにする、物理的なセキュリティー対策です。

2.3. 人事要件

JBoss EAP 7 の管理に割り当てられている 1 つ以上の有能な個人とその環境、および情報のセキュリティーが必要です。システム管理者の担当者は、不注意で、または悪用されていなかったり、管理者のドキュメントに記載されている指示に従って従ってください。

web サーバーアプリケーションやエンタープライズ Bean を含む JBoss EAP 7 のユーザーアプリケーションの開発者は、信頼でき、JBoss EAP 7 のユーザーガイダンスおよび評価されるすべての命令に準拠する必要があります。

2.4. 接続の要件

オペレーティングシステムと Java 仮想マシンの仕様に従って動作します。このような外部システムは、このガイダンスに従って設定されます。

JBoss EAP 7 が通信する他のシステムは同じ管理制御下にあり、JBoss EAP 7 と同じセキュリティーポリシー制約下で動作していると見なされます。

2.4.1. クラスターの接続性の要件

JBoss EAP 7 インスタンスは、パケットフィルタリングメカニズムを使用して、他のネットワークセグメントと論理的に分離されるネットワークセグメントで動作します。このパケットフィルターは、以下の条件の両方を満たす着信通信のみを許可する必要があります。

- ネットワークプロトコルが TCP である。
- 宛先ポートは 8080 または 8443 です。

JBoss EAP 7 インスタンスのいずれかからの送信通信はすべて許可する必要があります。

注記

信頼できないネットワークトラフィック (public、cluster、および internal) の 3 つの定義済みインターフェースがあります。詳細は、ネットワークインターフェース を参照してください。

各クラスターノードは、標準的なネットワークソケットを介して他のノードと通信します。各接続のクライアント側には、クライアントソケット用に予約されるポート範囲からホストのオペレーティングシステムにより割り当てられるポート番号が割り当てられます。これらのポートは動的ポートまたは一時ポートと呼ばれます。これらは、閉じられるまで接続によってのみ使用されます。接続が閉じられると、他の新しいクライアント接続がポートで使用できるようになります。このポート範囲を設定する必要がある場合は、オペレーティングシステムのドキュメントを参照してください。

第3章 パッケージのダウンロードおよび検証

JBoss EAP 7 は ZIP および RPM 形式で使用できます。RPM を使用する zip ファイルおよび ISO は、Red Hat カスタマーポータル (https://access.redhat.com) から入手できます。

ダウンロードしたソフトウェアの信頼性を保証するには、ファイルとそのソースの信頼性を確認します。

注記

OpenShift と使用するように設計されたインストーラーによるインストールやコンテナー化イメージなどの他の形式は、CC 準拠のシステムではサポートされません。

重要

特に説明がない限り、このセクションで説明するスクリーンショットやその他のサンプルは例のみになります。ダウンロードした Web サイトの実際のプレゼンテーションは、時間の経過とともに変わる可能性があります。

3.1. RED HAT カスタマーポータル

Red Hat カスタマーポータルは、Red Hat のナレッジリソースやサブスクリプションリソースを管理する集中プラットフォームです。以下を行うにあ、**Red Hat カスタマーポータル** を使用します。

- Red Hat エンタイトルメントやサポート契約の管理および維持
- 正式サポートされたソフトウェアのダウンロード
- 製品ドキュメントや Red Hat ナレッジベースの利用
- Red Hat サポートへのお問い合わせ
- Red Hat 製品のバグの登録

カスタマーポータルは https://access.redhat.com からアクセスできます。

3.2. ダウンロードサイトの認証の確認

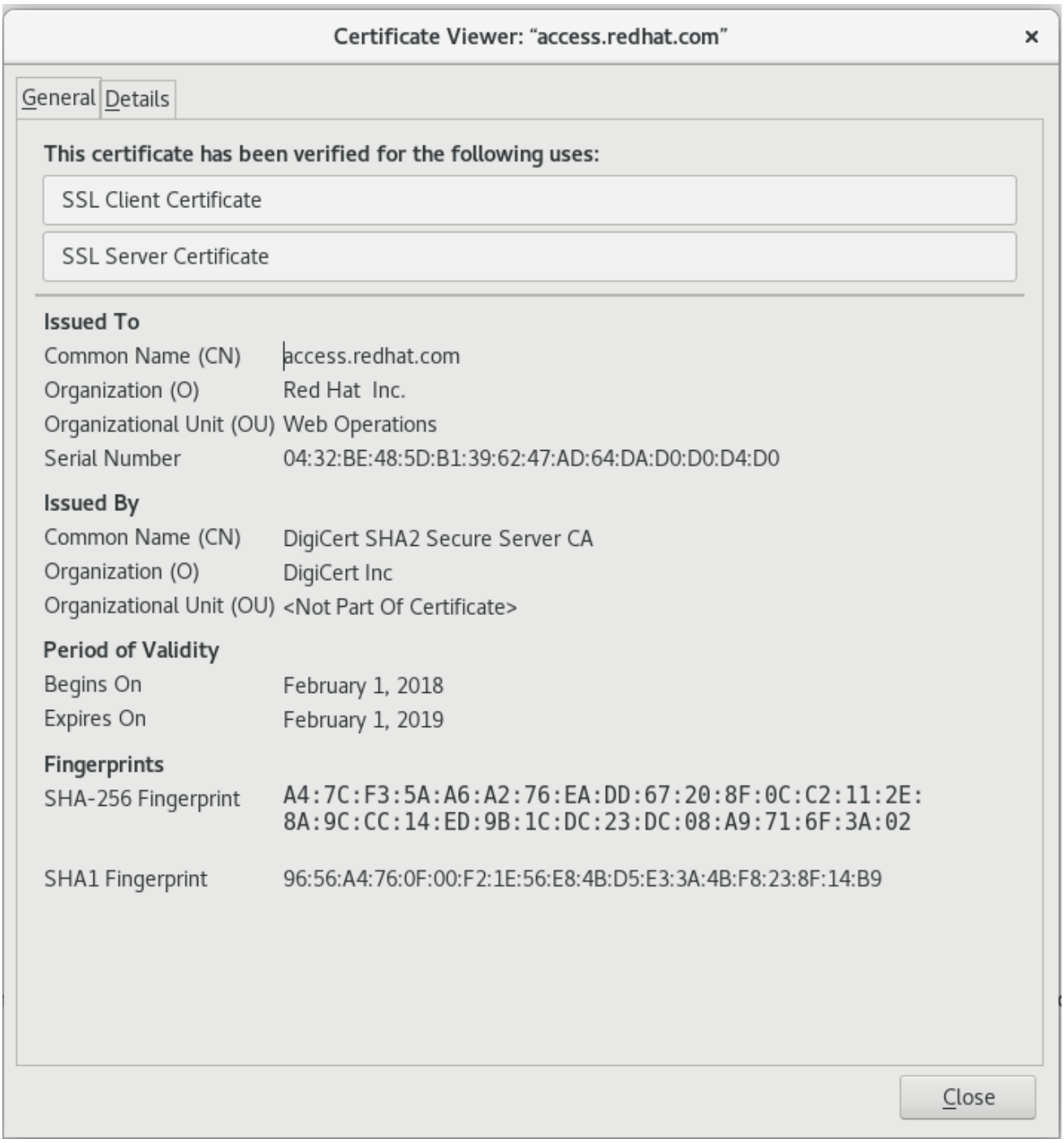
Red Hat カスタマーポータルと Red Hat Network は両方とも安全なサイトです。これは、ブラウザーステータスバーの「security padlock」アイコンで示されます。

「セキュリティーパディングロック」が表示されない場合は、アイデンティティー証明書を参照して、サイトの信頼性を確認してください。

Firefox でのサイトセキュリティーの確認

1. アドレスバーで、パディングロックアイコンをクリックします。
2. ポップアップボックスから **詳細情報**をクリックします。
3. Page Info ウィンドウで **Security** をクリックします。
4. 証明書には、証明書の発行時に発行元のサイトを所有するユーザーなどの詳細が表示されます。これは、期限が切れ、フィンガープリント検証文字列になります。

図3.1 Red Hat Network SSL 証明書の例



ブラウザーにロックアイコンがなく、検証済みの証明書が見つからない場合は、正しいサイトに接続されていない可能性があります。セキュアな Red Hat カスタマーポータルサイトにアクセスできない場合は、Red Hat サポートに連絡し、この問題を報告してください。

3.3. ダウンロードしたファイルの確認

ダウンロードした各ファイルを確認して、認定バージョンの JBoss EAP **PROD_VER** を確認する必要があります。Red Hat カスタマーポータルでは、各ファイルの SHA-256 ハッシュの合計を一覧表示しています。ダウンロードしたファイルの SHA-256 ハッシュの合計が、Red Hat カスタマーポータルの引用符と一致する場合は、検証されていることを前提とします。

JBoss ダウンロードでは、**Download File** リストのファイル名をクリックして、各ファイルの **Software Details** ページで **SHA-256** ハッシュ合計を表示できます。Red Hat Enterprise Linux ダウンロードでは、ISO ダウンロードリンクの横に **SHA-256** ハッシュ合計が表示されます。

Apple OSX では、**sha256** コマンドを **shasum -a 256** に置き換える必要があります。Microsoft Windows では、ネイティブユーティリティーがないため、サードパーティーの SHA256 hash sum ユーティリティーが必要になります。

Linux または Unix での sha256sum ツールの使用

1. ターミナルを開き、ファイルがダウンロードされたディレクトリーに移動します。
2. ファイルで **sha256sum** コマンド (または同等) を実行します。

例:

```
$ sha256sum jboss-eap-7.2.0.zip
682d2e7168c9f09cc019dce8f5a70e61169e2dc438dc44ba7352aba4e0634e20 *jboss-eap-7.2.0.zip
```

sha256sum ユーティリティーで生成された値は、ファイルについて Red Hat カスタマーポータルに表示される値と一致している必要があります。これらが同じでない場合は、ダウンロードが不完全または破損しているため、ファイルを再度ダウンロードする必要があります。ダウンロードを試みると、チェックサムが正常に検証されない場合は、Red Hat サポートにお問い合わせください。

注記

ダウンロードしたファイルのチェックサムを生成するには、ほとんどの Linux および Unix のオペレーティングシステムで **sha256sum** コマンドを使用できます。Mac OS X には、同等のコマンド **shasum -a 256** が含まれています。

Microsoft Windows を使用している場合は、Microsoft Windows に **SHA-256** チェックサムツールが含まれないため、サードパーティーユーティリティーをダウンロードして次の手順を実行する必要があります。

3.4. ZIP インストール

JBoss EAP 7 ZIP ファイルは カスタマーポータル から入手できます。ZIP ファイルのインストールはプラットフォームに依存しており、サポートされるすべてのプラットフォームに JBoss EAP 7 をインストールすることが推奨されます。

3.4.1. JBoss EAP のダウンロード

本トピックでは、必要なアーカイブをダウンロードするための手順を説明します。

ZIP ファイルをダウンロードします。

1. Red Hat カスタマーポータル (https://access.redhat.com) にログインします。
2. **ダウンロード**をクリックします。
3. **製品のダウンロード**リストの **Red Hat JBoss Enterprise Application Platform** をクリックします。
4. **Version** ドロップダウンメニューで 7.2 を選択します。
5. Releases のリストから Red Hat JBoss Enterprise Application Platform 7.2.0 を選択し、**Download** をクリックします。

6. Patches のリストから Red Hat JBoss Enterprise Application Platform 7.2 Update 03 を選択し、**Download** をクリックします。

7. ダウンロードが完了したら、ダウンロードしたファイルのチェックサムが、カスタマーポータルに記載されているチェックサムと一致することを確認します。ダウンロードしたファイルの検証を参照してください。

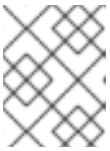
JBoss EAP 7 は正常にターゲットマシンにダウンロードされ、インストールの準備が整いました。

3.4.2. JBoss EAP のインストール (ZIP インストール)

ここでは、ダウンロードした ZIP ファイルを使用して JBoss EAP 7 をインストールする手順を説明します。

JBoss EAP の ZIP インストールファイルをダウンロードしたら、パッケージの内容を展開してインストールできます。

1. 必要な場合は、JBoss EAP をインストールするサーバーおよび場所に ZIP ファイルを移動します。



注記

JBoss EAP を実行するユーザーは、このディレクトリーへの読み書きアクセスが必要になります。

2. ZIP アーカイブを展開します。

```
$ unzip JBossEAPZipName
```

注記

Windows Server の場合は ZIP ファイルを右クリックし、**すべて展開** を選択します。

ZIP アーカイブを展開して作成したディレクトリーは、JBoss EAP インストールの最上位ディレクトリーとなります。このディレクトリーを **EAP_HOME** と呼びます。

3. 必要なパッチを適用します。
Unix ベースのシステムで CLI を使用してこの更新を適用するには、**JBOSS_HOME** から以下のコマンドを実行します。

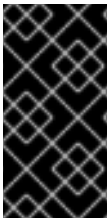
```
bin/jboss-cli.sh "patch apply path/to/jboss-eap-7.2.3-patch.zip"
```

Windows ベースのシステムで CLI を使用してこの更新を適用するには、**JBOSS_HOME** から以下のコマンドを実行します。

```
bin\jboss-cli.bat "patch apply path\to\jboss-eap-7.2.3-patch.zip"
```

3.5. ISO からの RPM インストール

Red Hat カスタマーポータルから JBoss EAP 7 の ISO インストールファイルをダウンロードします。すべてのセキュリティーが含まれ、エラータにパッチを適用します。



重要

認定された設定については、RPM メソッドを使用して JBoss EAP 7 をインストールするには ISO ファイルから行う必要があります。Red Hat Network から直接 RPM を使用して JBoss EAP 7 をインストールすることは、認定された設定に対して有効ではありません。

要件

- Red Hat Network でサーバーを登録します。
- ご自分の Red Hat Enterprise Linux バージョンに適した **Red Hat Enterprise Linux Server**ベースソフトウェアチャンネルにサブスクライブします。
- **JBoss Enterprise Platform グループ** の **JBoss Application Platform for Server**サブチャネルにはサブスクライブしないでください。

3.5.1. JBoss EAP 7 のダウンロード

JBoss EAP 7 の ISO のダウンロード

ISO イメージにアクセスするためのエンタイトルメントが必要です。手順を完了できない場合は、Red Hat サポートにサブスクリプション管理およびカスタマーサポートにお問い合わせください。

1. Red Hat カスタマーポータル (https://access.redhat.com) にログインします。
2. https://access.redhat.com/downloads/content/183/ver=7.2/ のリンクを開きます。
3. **Product Variant** ドロップダウンメニューで JBoss Enterprise Application Platform を選択します。
4. **Version** ドロップダウンメニューから 7.2 for RHEL 6 または RHEL 7 を選択します。
5. RHEL 6 の Red Hat JBoss Enterprise Application Platform 7.2.3 (RHEL 6) ISO イメージファイル、または RHEL 7 の Red Hat JBoss Enterprise Application Platform 7.2.3 (RHEL 7) ISO イメージファイルをクリックしてダウンロードを開始します。
   お使いの Red Hat Enterprise Linux のバージョンに適した ISO をダウンロードしてください。
6. ダウンロードが完了したら、ダウンロードした ISO のチェックサムがカスタマーポータルに記載されているチェックサムと一致することを確認します。手順は、ダウンロードしたファイルの確認 を参照してください。

3.5.2. ISO から JBoss EAP 7 をインストール

この手順は、Red Hat Enterprise Linux にのみ適用されます。

すべての ISO イメージには、評価される設定に関連するセキュリティーエラータとパッチが含まれています。ISO インストール方法を選択する場合は、その他のエラータをインストールする必要はありません。

重要

ISO イメージをインストールするには、スーパーユーザー権限をアクティベートする必要があります。

Red Hat Enterprise Linux の ISO から JBoss EAP 7 をインストールします。

1. ISO イメージをマウントする
   JBoss EAP 7 の ISO のダウンロード でダウンロードした ISO イメージを **/mnt/jboss** にマウントします。

```
[root ~]# mkdir /mnt/jboss
[root ~]# mount -o loop PATH_TO_ISO_IMAGE /mnt/jboss
```

2. リポジトリーの作成
   /etc/yum.repos.d/ に **jbosslocal.repo** という名前のファイルを作成します。

```
[root ~]# cat << EOF > /etc/yum.repos.d/jbosslocal.repo
[jbosslocal]
name=jbosslocal
baseurl=file:///mnt/jboss
enabled=1
gpgcheck=0
EOF
```

3. JBoss EAP 7 のインストール
   以下のコマンドを実行します。

```
[root ~]# yum groupinstall jboss-eap7
```

RPM インストールのデフォルトの **EAP_HOME** パスは **/usr/share/jbossas** です。

3.6. JBOSS EAP 7 インストールバージョンの確認

JBoss EAP 7 インストールのバージョン番号を確認する方法は 3 つあります。

- 起動スクリプトでの **-V** の使用
- 管理コンソールの使用
- コンソール出力または server.log ファイルの表示

起動スクリプトでの -V の使用
-V スイッチのみでサーバーを起動するのと同じスクリプトを実行して、JBoss EAP 7 インストールのバージョンに関する情報を取得します。インストールがスタンドアロンまたは管理対象ドメインの場合、Red Hat Enterprise Linux および Solaris の場合は、このスクリプトは **standalone.sh** または **domain.sh** で 、Microsoft Windows Server の場合は同等の **.bat** スクリプトになります。起動スクリプトは、**EAP_HOME/bin** にあります。

-V スイッチのみで起動スクリプトを実行しても、サーバーは起動せず、サーバーを実行する必要はありません。JBoss EAP のバージョンと、設定された Java 環境に関する情報を表示します。以下は、Red Hat Enterprise Linux における JBoss EAP 7 のインストールで使用する例です。出力の最後の行として表示されるバージョン番号 **JBoss EAP PROD_VER.GA** に注意してください。

```
$ ./standalone.sh -V
=========================================================================

  JBoss Bootstrap Environment
```

```
  JBOSS_HOME: /home/user/EAP-7.2.GA

  JAVA: java

  JAVA_OPTS:  -server -verbose:gc -Xloggc:"/home/user/EAP-7.2.GA/standalone/log/gc.log" -
XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+UseGCLogFileRotation -
XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=3M -XX:-TraceClassUnloading -Xms1303m -
Xmx1303m -XX:MetaspaceSize=96M -XX:MaxMetaspaceSize=256m -
Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -
Djava.awt.headless=true

=========================================================================

17:57:11,912 INFO  [org.jboss.modules] (main) JBoss Modules version 1.6.0.Final-redhat-1
JBoss EAP 7.2.0.GA (WildFly Core 3.0.10.Final-redhat-1)
```

管理コンソールの使用
JBoss EAP 7 サーバーの実行中に、バージョン情報は http://localhost:9990/console/ にある Web コンソールのホームページの上部に表示されます。

コンソール出力または server.log ファイルの表示
サーバーが起動すると、バージョンはコンソールにエコーされ、サーバーログに書き込まれます。スタンドアロン設定の場合、サーバーログは **EAP_HOME/standalone/log/server.log** にあり、管理対象ドメインサーバーでは **EAP_HOME/domain/servers/SERVER_NAME/log/server.log** になります。

```
08:29:23,756 INFO [org.jboss.as] (Controller Boot Thread) WFLYSRV0025: JBoss EAP 7.2.0.GA
(WildFly Core 3.0.10.Final-redhat-1) started in 3494ms - Started 299 of 560 services (348 services
are lazy, passive or on-demand)
```

3.7. COMMON CRITERIA コンプライアント JBOSS EAP 7 インストールの更新

JBoss EAP の更新は定期的にリリースされ、これらの更新には重要なセキュリティー問題に対する修正が含まれることがあります。しかし、JBoss EAP 7 ZIP インストールを除き、Common Criteria 準拠の JBoss EAP インストールへの更新により、そのインストールの Common Criteria 認定が無効になります。

警告

セキュリティー更新を適用しないと、重大なリスクが生じる可能性があります。セキュリティー更新がリリースされると、JBoss EAP インストールを Common Criteria Compliant に適用するかどうかが判断されます。セキュリティー問題を解決する更新を適用することが推奨されます。

3.7.1. ZIP インストールへのパッチの適用

重要

Common Criteria 準拠のインストールにパッチを適用すると、そのインストールの Common Criteria 認定が無効になります。

patch コマンドを使用して、ZIP インストールにパッチを適用します。手順は **インストールガイド** を参照してください。

3.7.2. ISO から RPM インストールへのパッチの適用

重要

Common Criteria 準拠のインストールにパッチを適用すると、そのインストールの Common Criteria 認定が無効になります。

1. ローカルリポジトリーの削除
   jbosslocal.repo という名前のファイルを **/etc/yum.repos.d/** から削除します。

   ```
   [root ~]# rm /etc/yum.repos.d/jbosslocal.repo
   ```

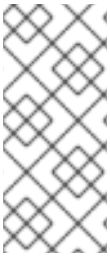
2. インストールを通常の RPM インストールとして更新します。手順は **インストールガイド** を参照してください。

第4章 JBOSS EAP 7 の開始および停止

4.1. JBOSS EAP 7 を起動します。

以下のいずれかの方法で JBoss EAP 7 を起動します。

- JBoss EAP 7 をスタンドアロンサーバーとして起動
- 管理対象ドメインとしての JBoss EAP 7 の起動



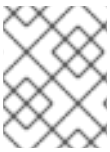
注記

EAL4 CC については、Java Security Manager を有効にする必要があります。Java Security Manager は、Java Virtual Machine (JVM) サンドボックスの外部境界を管理するクラスで、JVM 内で実行されるコードが JVM 外のリソースと対話する方法を制御します。Java Security Manager のサーバー設定の変更に関する詳細は、JBoss EAP **サーバーセキュリティーの設定方法**を参照してください。

4.2. JBOSS EAP 7 をスタンドアロンサーバーとして起動

このトピックでは、JBoss EAP 7 をスタンドアロンサーバーとして起動する手順を説明します。

- Red Hat Enterprise Linux の場合は、**EAP_HOME/bin/standalone.sh** コマンドを実行します。
- Microsoft Windows Server の場合は、**EAP_HOME\bin\standalone.bat** のコマンドを実行します。



注記

起動スクリプトに渡す追加パラメーターの一覧を出力するには、**-h** パラメーターを使用します。

4.3. 管理対象ドメインとしての JBOSS EAP 7 の起動

操作の順序

ドメイン内の他のホストコントローラーの前にドメインコントローラーを起動する必要があります。最初にドメインコントローラーでこの手順を実行し、次にドメイン内の相互ホストコントローラーでこの手順を使用します。

プラットフォームサービスを管理対象ドメインとして起動

1. Red Hat Enterprise Linux の場合は、**EAP_HOME/bin/domain.sh** コマンドを実行します。
2. Microsoft Windows Server の場合は、**EAP_HOME\bin\domain.bat** のコマンドを実行します。

注記

起動スクリプトに渡すことのできるパラメーターの一覧は、**-h** パラメーターを使用します。

4.4. 代替設定での JBOSS EAP 7 の起動

設定ファイルを指定しない場合、サーバーはデフォルトファイルで起動します。ただし、サーバーを起動する際には、設定を手動で指定できます。プロセスは、管理対象ドメインまたはスタンドアロンサーバーを使用しているか、または使用しているオペレーティングシステムによって異なります。

前提条件

代替の設定ファイルを使用する前に、デフォルト設定をテンプレートとして使用して準備します。管理対象ドメインの場合は、設定ファイルを **EAP_HOME/domain/configuration/** ディレクトリーに配置する必要があります。スタンドアロンサーバーの場合、設定ファイルは **EAP_HOME/standalone/configuration/** ディレクトリーに置く必要があります。

注記

複数の設定例が **EAP_HOME/docs/examples/configs/** ディレクトリーに含まれます。これらの例を使用して、クラスタリングやトランザクション XTS API などの追加機能を有効にします。

代替設定でインスタンスの起動

1. スタンドアロンサーバー
   スタンドアロンサーバーの場合、設定ファイルのファイル名を **--server-config** パラメーターにオプションとして指定します。設定ファイルは **EAP_HOME/standalone/configuration/** ディレクトリーに置く必要があり、そのディレクトリーへの相対パスを指定する必要があります。

   例: Red Hat Enterprise Linux でスタンドアロンサーバーの代替設定ファイルの使用

   ```
   [user@host bin]$ ./standalone.sh --server-config=standalone-alternative.xml
   ```

   この例は、**EAP_HOME/standalone/configuration/standalone-alternative.xml** 設定ファイルを使用します。

   例: Microsoft Windows Server でのスタンドアロンサーバーの代替設定ファイルの使用

   ```
   C:\EAP_HOME\bin> standalone.bat --server-config=standalone-alternative.xml
   ```

   この例では、**EAP_HOME\standalone\configuration\standalone-alternative.xml** 設定ファイルを使用します。

2. 管理対象ドメイン
   管理対象ドメインの場合は、**--domain-config** パラメーターにオプションとして設定ファイルのファイル名を指定します。このファイルはディレクトリーに存在する必要があり、その **EAP_HOME/domain/configuration/** ディレクトリーへの相対パスを指定する必要があります。

   例: Red Hat Enterprise Linux で管理対象ドメインの代替設定ファイルの使用

   ```
   [user@host bin]$ ./domain.sh --domain-config=domain-alternative.xml
   ```

   この例では、**EAP_HOME/domain/configuration/domain-alternative.xml** 設定ファイルを使用します。

   例: Microsoft Windows Server の管理対象ドメインの代替設定ファイルの使用

   ```
   C:\EAP_HOME\bin> domain.bat --domain-config=domain-alternative.xml
   ```

この例では、**EAP_HOME\domain\configuration\domain-alternative.xml** 設定ファイルを使用します。

代替の設定ファイルを使用して JBoss EAP が実行されているはずです。

4.5. SERVER RUNTIME で渡されるスイッチおよび引数の参照

アプリケーションサーバーの起動スクリプトは、実行時に引数の追加とスイッチを受け入れます。これらのパラメーターを使用すると、**standalone.xml**、**domain.xml**、および **host.xml** 設定ファイルに定義した別の設定でサーバーを起動できます。これには、ソケットバインディングまたはセカンダリー設定の代替セットを持つサーバーの起動が含まれます。起動時に help スイッチを渡すと、利用可能なパラメーターの一覧にアクセスできます。

例

以下の例は、**-h** または **--help** を加えた状態で、JBoss EAP 7 をスタンドアロンサーバーとして起動と管理対象ドメインとしての JBoss EAP 7 の起動で説明しているサーバーセットアップと似ています。ヘルプスイッチの結果は、以下の表で説明されています。

スタンドアロンサーバー:

```
[localhost bin]$ standalone.sh -h
```

管理対象ドメインの場合:

```
[localhost bin]$ domain.sh -h
```

表4.1 ランタイムスイッチおよび引数の表

引数またはスイッチ	モード	説明
--admin-only	Standalone	サーバーの実行タイプを **ADMIN_ONLY** に設定します。これにより管理インターフェースが開かれ、管理リクエストが許可されますが、他のランタイムサービスは起動されず、エンドユーザーのリクエストは許可されません。
--admin-only	ドメイン	ホストコントローラーの実行タイプを **ADMIN_ONLY** に設定します。 これにより管理インターフェースが開かれ、管理リクエストが許可されますが、サーバーは起動しません。 ホストコントローラーがドメインのマスターである場合はスレーブホストコントローラーからの受信接続が許可されます。
-b <value>, **-b=<value>**	Standalone m、Domain	システムプロパティー **jboss.bind.address** を指定の値に設定します。
-b<interface>=<value>	Standalone m、Domain	システムプロパティー **jboss.bind.address.<interface>** を指定の値に設定します。
--backup	ドメイン	このホストがドメインコントローラーではない場合でも永続ドメイン設定のコピーを保持します。

引数またはスイッチ	モード	説明
-c <config>, -c=<config>	Standalone	使用するサーバー設定ファイルの名前。デフォルトは **standalone.xml** です。
-c <config>, -c=<config>	ドメイン	使用するサーバー設定ファイルの名前。デフォルトは **domain.xml** です。
--cached-dc	ドメイン	ホストがドメインコントローラーではなく、起動時にドメインコントローラーに接続できない場合、ローカルでキャッシュされたドメイン設定のコピーを使用してブートします。
--debug [<port>]	Standalone	オプションの引数を用いてデバッグモードを有効にし、ポートを指定します。起動スクリプトがサポートする場合のみ動作します。
-D<name>[=<value>]	Standalone m、Domain	システムプロパティーを設定します。
--domain-config=<config>	ドメイン	使用するサーバー設定ファイルの名前。デフォルトは **domain.xml** です。
-h、--help	Standalone m、Domain	ヘルプメッセージを表示し、終了します。
--host-config=<config>	ドメイン	使用するホスト設定ファイルの名前。デフォルトは **host.xml** です。
--interprocess-hc-address=<address>	ドメイン	ホストコントローラーがプロセスコントローラーからの通信をリッスンしなければならないアドレス。
--interprocess-hc-port=<port>	ドメイン	ホストコントローラーがプロセスコントローラーからの通信をリッスンしなければならないポート。
--master-address=<address>	ドメイン	システムプロパティー **jboss.domain.master.address** を指定の値に設定します。デフォルトのスレーブホストコントローラー設定では、マスターホストコントローラーのアドレスを設定するために使用されます。
--master-port=<port>	ドメイン	システムプロパティー **jboss.domain.master.port** を指定の値に設定します。デフォルトのスレーブホストコントローラー設定では、マスターホストコントローラーによるネイティブ管理の通信で使用されるポートを設定するために使用されます。
--read-only-server-config=<config>	Standalone	使用するサーバー設定ファイルの名前。元のファイルは上書きされないため、**--server-config** および **-c** とは異なります。

引数またはスイッチ	モード	説明
--read-only-domain-config=<config>	ドメイン	使用するドメイン設定ファイルの名前。最初のファイルは上書きされないため、**--domain-config** および **-c** とは異なります。
--read-only-host-config=<config>	ドメイン	使用するホスト設定ファイルの名前。最初のファイルは上書きされないため、**--host-config** とは異なります。
-P <url>, **-P=<url>**, **--properties=<url>**	Standalone m、Domain	該当する URL からシステムプロパティーをロードします。
--pc-address=<address>	ドメイン	プロセスコントローラーが制御するプロセスからの通信をリッスンするアドレス。
--pc-port=<port>	ドメイン	プロセスコントローラーが制御するプロセスからの通信をリッスンするポート。
-S<name>[=<value>]	Standalone	セキュリティープロパティーを設定します。
--server-config=<config>	Standalone	使用するサーバー設定ファイルの名前。デフォルトは **standalone.xml** です。
-u <value>, **-u=<value>**	Standalone m、Domain	システムプロパティー **jboss.default.multicast.address** を指定の値に設定します。
-v、**-V**、**--version**	Standalone m、Domain	アプリケーションサーバーのバージョンを表示し、終了します。
-secmgr	Standalone m、Domain	セキュリティーマネージャーがインストールされた状態でサーバーを実行します。
--start-mode=<mode>	Standalone	サーバーの起動モードを設定します。このオプションは、 **--admin-only** と併用できません。有効な値は以下のとおりです。 ● **normal**: サーバーは通常どおりに起動します。 ● **admin-only**: サーバーは管理インターフェースのみを開き、管理リクエストを許可しますが、他のランタイムサービスは起動せず、エンドユーザーのリクエストを許可しません。 ● **suspend**: サーバーは中断モードで起動され、再開するまでリクエストに対処しません。

第5章 設定要件

以下のセクションでは、CC 要件に準拠するサーバー設定への変更を説明します。管理コンソールまたは管理 CLI を使用して変更を行うと、既存の設定が自動的にバックアップされます。これらのバックアップは参照に使用したり、必要に応じて以前の設定に戻すことができます。この機能の詳細は、**設定ガイド** を参照してください。

5.1. ネットワーク設定

5.1.1. ネットワークインターフェース

以下のネットワークインターフェースが定義され、作成され、信頼できないネットワークトラフィックは分離されます。

public

外部からの通信、信頼できない可能性があります。

cluster

クラスターノード間の通信信頼できない人からはアクセスできません。これは、ネットワーク/ファイアウォール設定の一部として実施する必要があります。

internal

内部ネットワーク経由で信頼されたサーバーまたはユーザー (管理者など) との通信。システムの信頼できない当事者または一般ユーザーからはアクセスできません。

5.1.2. ネットワークインターフェースの設定

共通基準要件に準拠するには、関連するネットワーク設定を適用します。

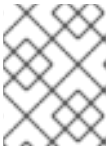
ネットワークインターフェースの定義および設定

1. **internal** および **cluster** のネットワークインターフェースを作成します。

```
[standalone@localhost:9990 /] /interface=cluster:add(inet-address=expression
"${jboss.bind.address.cluster:127.0.0.1}")
[standalone@localhost:9990 /] /interface=internal:add(inet-address=expression
"${jboss.bind.address.internal:127.0.0.1}")
```

2. 各ソケットを指定されたネットワークインターフェースにバインドします。
ネットワークバインディングの各行について、以下のコマンドを使用してソケットを指定されたネットワークインターフェースにバインドします。

```
[standalone@localhost:9990 /] /socket-binding-group=standard-sockets/socket-
binding=BINDING_NAME:write-attribute(name=interface,value=NETWORK_INTERFACE)
```



注記

すべてのソケットバインディングが利用可能な設定に関連するわけではありません。製品設定の一部のみを使用できます。

たとえば、以下のコマンドは **internal** ネットワークインターフェースに **management-http** をバインドします。

```
[standalone@localhost:9990 /] /socket-binding-group=standard-sockets/socket-binding=management-http:write-attribute(name=interface,value=internal)
```

3. **unsecure** ネットワークインターフェースを、設定ファイル **standalone-full.xml** および **standalone-full-ha.xml** から削除します。

```
[standalone@localhost:9990 /] /interface=unsecure:remove
```

4. JBoss EAP を再起動します。
ネットワークバインディングを有効にするために JBoss EAP を再起動します。

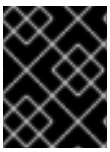
表5.1 ネットワークバインディング

バインディング名	ネットワークインターフェース	ポート番号
ajp	internal	8009
http	public	8080
https	public	8443
iiop	internal	3528
iiop-ssl	internal	3529
jgroups-mping	cluster	0 - multicast: 45700
jgroups-tcp	cluster	7600
jgroups-udp	cluster	55200 - multicast: 45688
management-http	internal	9990
management-https	internal	9993
messaging	internal	5445
messaging-throughput	internal	5455
modcluster	public/internal	0 - multicast: 23364
txn-recovery-environment	internal	4712
txn-status-manager	internal	4713

5.2. セキュリティー設定

共通基準要件のセキュリティーコンプライアンスを確保するには、以下の設定手順を実行する必要があります。

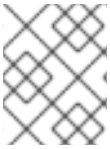
サーバー全体での Elytron セキュリティーの有効化



重要

レガシーセキュリティーサブシステムは、Common Criteria に準拠していません。レガシーセキュリティーサブシステムを無効にし、サブシステム **elytron** を有効にします。

Elytron は、シンプルな方法でサーバー全体に有効にできます。JBoss EAP 7.2 では、Elytron をセキュリティープロバイダーとして有効にするサンプル設定スクリプトが導入されました。このスクリプトは、サーバーインストールの **EAP_HOME**/docs/examples ディレクトリーにあります。



注記

設定スクリプトの例には、サポートされなくなったネイティブインターフェースへの参照が含まれます。このスクリプトから以下の行を削除します。

```
/host=master/core-service=management/management-interface=native-interface:write-
attribute(name=sasl-authentication-factory,value=management-sasl-authentication)
/host=master/core-service=management/management-interface=native-interface:undefine-
attribute(name=security-realm)
```

以下のコマンドを実行して、サーバー全体で Elytron セキュリティーを有効にします。コマンドを実行する前に、サーバーを完全に停止する必要があります。

```
$ EAP_HOME/bin/jboss-cli.sh --file=EAP_HOME/docs/examples/enable-elytron.cli
```

5.2.1. 認可について

JBoss EAP 7 では、アイデンティティーを読み込むために **SecurityDomain** は 1 つ以上の **SecurityRealms** を参照します。これらのアイデンティティーは認証に使用されます。また、承認決定のアイデンティティーをマップするためのロールデコーダーおよびマッパー参照も含まれます。

承認は認証とは異なり、通常は認証後に行われます。

注記

Common Criteria Certified 設定では XACML は許可されません。

5.2.2. Java Security Manager

Java Security Manager は、Java Virtual Machine (JVM) サンドボックスの外部境界を管理するクラスで、JVM 内で実行されるコードが JVM 外のリソースと対話する方法を制御します。Java Security Manager を有効にすると、Java API はさまざまな潜在的に危険な操作を実行する前に、セキュリティーマネージャーで承認を確認します。Java Security Manager はセキュリティーポリシーを使用して、特定のアクションが許可または拒否されるかどうかを判断します。

重要

JBoss EAP 7 では、**security-manager** サブシステムと、個別のデプロイメントで XML ファイルを使用するという 2 つの方法で Java セキュリティーポリシーを定義します。**Security-manager** サブシステムは、**ALL** ディプロイメントの最小および最大パーミッションを定義します。一方、XML ファイルは、個別のディプロイメントによって要求されたパーミッションを指定します。Java Security Manager を有効にして JBoss EAP を起動する前に、すべてのセキュリティーポリシーが **security-manager** サブシステムで定義されていることを確認する必要があります。

security-manager サブシステムでポリシーを定義する方法は、JBoss EAP **How to Configure Server Security** の Java Security Manager を参照してください。

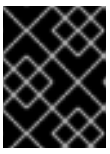
Java Security Manager を使用した JBoss EAP の実行
Java Security Manager で JBoss EAP を実行するには、起動時に **secmgr** オプションを使用する必要があります。これには 2 つの方法があります。

- 起動スクリプトでフラグを使用します。起動スクリプトで **-secmgr** フラグを使用するには、JBoss EAP インスタンスの起動時に、このフラグを含めます。

  例: 起動スクリプト

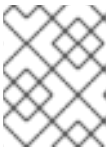
  ```
  ./standalone.sh -secmgr
  ```

- 起動設定ファイルの使用

  重要

  設定ファイルを編集する前に、ドメインまたはスタンドアロンサーバーを完全に停止する必要があります。

  注記

  管理対象ドメインで JBoss EAP を使用している場合は、ドメインの各物理ホストまたはインスタンスで以下の手順を実行する必要があります。

  起動設定ファイルを使用して Java Security Manager を有効にするには、スタンドアロンインスタンスまたは管理対象ドメインのいずれかを実行しているかに応じて、**standalone.conf** または **domain.conf** ファイルのいずれかを編集する必要があります。Windows で実行している場合は、代わりに **standalone.conf.bat** または **domain.conf.bat** ファイルが使用されます。

  設定ファイルの **secmgr="true"** 行のコメントを解除します。

  例: standalone.conf または domain.conf

  ```
  # Uncomment this to run with a security manager enabled
  SECMGR="true"
  ```

  例: standalone.conf.bat または domain.conf.bat

  ```
  rem # Uncomment this to run with a security manager enabled
  set "SECMGR=true"
  ```

5.2.3. EJB 承認ポリシー

アプリケーションは、JACC (Java Authorization Contract for Containers) の承認モジュールを使用して、カスタム認証および承認検証を実装できます。JBoss EAP 7 では、JACC 承認モジュールは JAAS セキュリティードメインの一部を形成します。

5.2.3.1. elytron サブシステムを使用した JACC の有効化

レガシーセキュリティーサブシステムでの JACC の無効化

アプリケーションサーバーはデフォルトでレガシー **security** サブシステムを使用して、JACC ポリシープロバイダーおよびファクトリーを設定します。デフォルト設定は PicketBox から実装へマップします。

Elytron またはアプリケーションサーバーにインストールするその他のポリシーを使用して JACC 設定を管理するには、最初にレガシー **security** サブシステムで JACC を無効にする必要があります。これには、以下の管理 CLI コマンドを使用できます。

```
/subsystem=security:write-attribute(name=initialize-jacc, value=false)
```

この作業を怠ると、次のエラーメッセージがサーバーログに出力されます: **MSC000004: Failure during stop of service org.wildfly.security.policy: java.lang.StackOverflowError**

JACC ポリシープロバイダーの定義

elytron サブシステムは、JACC 仕様を基にして組み込みのポリシープロバイダーを提供します。ポリシープロバイダーを作成するには、以下の管理 CLI コマンドを実行します。

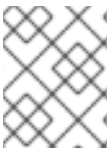
```
/subsystem=elytron/policy=jacc:add(jacc-policy={})

reload
```

Web デプロイメントに対する JACC の有効化

JACC ポリシープロバイダーを定義したら、以下のコマンドを実行して、web デプロイメントに対して JACC を有効にできます。

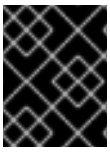
```
/subsystem=undertow/application-security-domain=other:write-attribute(name=enable-
jacc,value=true)
```

上記のコマンドは、Undertow サブシステムの「その他の」application-security-domain の JACC を有効にします。



注記

「その他の」application-security-domain は、サーバー全体で Elytron を有効にするスクリプトを実行する際に追加されます。セキュリティー設定を参照してください。



重要

サーバー全体で Elytron を有効にするスクリプトを実行していない場合は、最初に以下のコマンドを実行する必要があります。

```
/subsystem=undertow/application-security-domain=other:add(security-domain=ApplicationDomain)
```

EJB デプロイメントに対する JACC の有効化

JACC ポリシープロバイダーを定義したら、以下のコマンドを実行して、EJB デプロイメントに対して JACC を有効にできます。

```
/subsystem=ejb3/application-security-domain=other:write-attribute(name=enable-jacc,value=true)
```

上記のコマンドは、EJB サブシステムの「その他の」application-security-domain の JACC を有効にします。

注記

「その他の」application-security-domain は、サーバー全体で Elytron を有効にするスクリプトを実行する際に追加されます。セキュリティー設定を参照してください。

重要

サーバー全体で Elytron を有効にするスクリプトを実行していない場合は、最初に以下のコマンドを実行する必要があります。

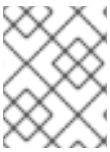
```
/subsystem=ejb3/application-security-domain=other:add(security-domain=ApplicationDomain)
```

カスタム Elytron ポリシープロバイダーの作成

パーミッションをチェックするために一部の外部承認サービスと統合したい場合など、カスタム **java.security.Policy** が必要なときにカスタムのポリシープロバイダーが使用されます。カスタムポリシープロバイダーを作成するには、**java.security.Policy** を実装し、実装でカスタムモジュールを作成およびプラグし、**elytron** サブシステムのモジュールから実装を使用します。

```
/subsystem=elytron/policy=policy-provider-a:add(custom-policy={class-name=MyPolicyProviderA,
module=x.y.z})
```

詳細は、開発ガイドの**ポリシープロバイダープロパティー**を参照してください。



注記

ほとんどの場合で、JACC ポリシープロバイダーを Java EE 対応のアプリケーションサーバーの一部として想定どおりに使用できます。

5.3. ELYTRON サブシステム

elytron サブシステムは JBoss EAP 7.2 に新たに追加されました。これは、アプリケーションサーバー全体でのセキュリティーの統一に使用されるセキュリティーフレームワークである WildFly Elytron プロジェクトをベースにしています。**elytron** サブシステムにより、単一の設定場所でアプリケーションと管理インタフェースの両方をセキュアにできます。WildFly Elytron は、機能のカスタム実装を提供し、**elytron** サブシステムと統合するために複数の API と SPI も提供します。

注記

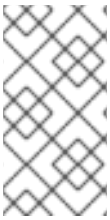
elytron サブシステムは CC 準拠システムのセキュリティーを行います。レガシーセキュリティーサブシステムとレガシーセキュリティーレルムはサポートされません。

注記

ボールトは CC 準拠のシステムでサポートされません。代わりにクレデンシャルストアが推奨されます。

さまざまな Elytron コンポーネントに関する背景情報は、セキュリティーアーキテクチャーを参照してください。

5.3.1. Elytron サブシステムの追加および削除



注記

JBoss EAP 7.2 以降では、**elytron** サブシステムがすでに存在し、追加の設定は必要ありません。

これは、古い JBoss EAP インストールを使用している場合にのみ必要です。

elytron サブシステムに必要な **elytron** 拡張機能を追加する方法:

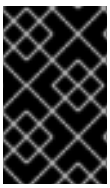
```
/extension=org.wildfly.extension.elytron:add()
```

JBoss EAP で **elytron** サブシステムを追加するには、以下を実行します。

```
/subsystem=elytron:add

reload
```

JBoss EAP で **elytron** サブシステムを削除するには、以下を実行します。

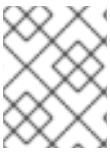
```
/subsystem=elytron:remove

reload
```



重要

JBoss EAP 内の他のサブシステムには **elytron** サブシステムの依存関係があることがあります。これらの依存関係を削除しても問題が解決されない場合、JBoss EAP の起動時にエラーが発生します。

5.4. データベースの設定



注記

起動サーバーの動作を改善するには、JDBC ドライバーをコアモジュールとしてインストールする方法が推奨されます。

JDBC ドライバーのセキュリティーパーミッション

1. デプロイメントでの JDBC ドライバーのセキュリティーパーミッション

JBoss EAP 7 では、JSR 342 に含まれ、Java EE 仕様の一部である **META-INF/permissions.xml** をデプロイメントに追加できます。このファイルでは、デプロイメントに必要なパーミッションを指定できます。

デプロイメントに必要なすべての権限は、それぞれのドライバーのドキュメントに記載されています。

2. モジュールにおける JDBC ドライバーのセキュリティーパーミッション
モジュールとしてインストールされるすべての JDBC ドライバーでは、セキュリティーマネージャーのパーミッションに追加の設定は必要ありません。

詳細は、**サーバーセキュリティーの設定方法** の Java Security Manager を参照してください。

5.5. JAVA セキュリティーパーミッションの設定に関するガイダンス

認定されたシステムの操作を行うシステム管理者は、認証されたシステムがセキュリティーマネージャーの有効化モードで実行された場合に、認定システムにデプロイされたすべてのエンタープライズアプリケーションのセキュリティーパーミッションを設定することが想定されます。

警告

認定された設定を維持するために、一般制限 の他に、以下のパーミッション **をアプリケーションに付与することはできません。**

- アプリケーション専用のファイルを除くファイルパーミッション
- ネットワークパーミッション
- ネイティブコードを読み込むパーミッション

詳細は、**サーバーセキュリティーの設定方法** ガイドの Java セキュリティーポリシーの定義を参照してください。

第6章 COMMON CRITERIA CERTIFIED SYSTEM の開発ガイド

6.1. エンタープライズアプリケーション

JBoss EAP 7 は、以下を含む Java EE 7 の Full Platform および Web Profile 標準を実装します。

- Batch 1.0
- JSON-P 1.0
- Concurrency 1.0
- WebSocket 1.1
- JMS 2.0
- JPA 2.1
- JCA 1.7
- JAX-RS 2.0
- JAX-WS 2.2
- Servlet 3.1
- JSF 2.2
- JSP 2.3
- EL 3.0
- CDI 1.2
- JTA 1.2
- Interceptors 1.2
- Common Annotations 1.1
- Managed Beans 1.0
- EJB 3.2
- Bean Validation 1.1

通常、アプリケーションはクライアントからのリクエストを受け入れ、一部の処理を行い、結果と共に応答します。信頼できる開発者が開発したエンタープライズアプリケーションは、ユーザーアプリケーションと呼ばれます。

6.2. 一般的な制限

認定済みのシステムに対して安全なソフトウェアを開発する場合は、信頼できるソフトウェア開発者は以下の制限に従う必要があります。

1. 該当する製品ドキュメントに記載されていない API (Application Programming Interface) は**使用しないでください**。

2. Enterprise JavaBeans Specification v3.2 が関係するプログラミング制限は、厳密にフォローする必要があります。詳細は、JSR 345: Enterprise JavaBeans 3.2 仕様 を参照してください。

Enterprise Java Beans 仕様開発者の制約
制限は次のとおりです。

- エンタープライズ Bean は、読み取り/書き込み静的フィールドは使用できません。読み取り専用静的フィールドの使用が許可されます。そのため、エンタープライズ bean クラスの静的フィールドはすべて final として宣言されることが推奨されます。

- エンタープライズ Bean は、スレッド同期のプリミティブを使用して、複数のインスタンスの実行を同期することはできません。

- エンタープライズ Bean は、表示への情報の出力またはキーボードから情報の入力を試みるために AWT 機能を使用しないでください。

- エンタープライズ Bean は、ファイルシステムのファイルおよびディレクトリーへのアクセスを試みるために **java.io** パッケージを使用しないでください。

- エンタープライズ Bean は、ソケットをリッスンしたり、ソケット上での接続を許可したり、マルチキャストにソケットを使用したりしないでください。

- Enterprise Bean は、Java 言語のセキュリティールールが原因で、エンタープライズ Bean がアクセスできない宣言されたメンバーに関する情報を取得するためにクラスのクエリーを試行しない必要があります。エンタープライズ Bean は、Java プログラミング言語のセキュリティールールが利用できなくなる情報にアクセスするために Reflection API を使用しないでください。

- エンタープライズ Bean は以下を試行しないでください。
  - クラスローダーの作成
  - 現在のクラスローダーの取得
  - コンテキストクラスローダーの設定
  - Java Security Manager
  - 新しいセキュリティーマネージャーの作成
  - JVM の停止
  - 入力、出力、およびエラーストリームの変更

- エンタープライズ Bean は、URL によって使用される ServerSocket、Socket、またはストリームハンドラーファクトリーによって使用されるソケットファクトリーを設定しないでください。

- エンタープライズ Bean はスレッドの管理を試行しないでください。エンタープライズ Bean は、スレッドの開始、停止、一時停止、再開を試みたり、スレッドの優先順位または名前を変更したりしないでください。エンタープライズ Bean はスレッドグループの管理を試行しないでください。

- エンタープライズ Bean は、特定のコードソースのセキュリティーポリシー情報を取得しないでください。

- エンタープライズ Bean はネイティブライブラリーのロードを試行しないでください。
- エンタープライズ Bean は、Java プログラミング言語の通常のルールによりエンタープライズ Bean が利用できなくなるパッケージおよびクラスへのアクセスを試みないでください。
- エンタープライズ Bean は、パッケージでクラスの定義を試行しないでください。
- エンタープライズ Bean は、セキュリティー設定オブジェクト（Policy、セキュリティー、プロバイダー、署名、およびアイデンティティー）にアクセスまたは変更しないでください。
- エンタープライズ Bean は、Java Serialization Protocol のサブクラスおよびオブジェクトの置換機能の使用を試みないでください。
- エンタープライズ Bean は、これを引数またはメソッドの結果として渡すことはできません。エンタープライズ Bean は、代わりに **SessionContext.getEJBObject**、**SessionContext.getEJBLocalObject**、**EntityContext.getEJBObject**、または **EntityContext.getEJBLocalObject** の結果を渡す必要があります。
- エンタープライズ Bean は、SPNEGO (Simple and Protected GSSAPI Negotiation Mechanism) を使用しないでください。
- エンタープライズ Bean は PicketBox からのアノテーションを使用できません。JAAS モジュールの動作を変更する以下のアノテーションは使用しないでください。
  - **@AuthenticationMechanism**
  - **@SecurityMapping**
  - **@Authentication**
  - **@Authorization**
  - **@SecurityConfig**
  - **@SecurityAudit**

これらの制限は、認定システムがセキュリティーマネージャーの有効モードで実行されている場合に Java Security Manager によって適用されます。認定されたシステムのセキュリティーおよび安定性の観点から保護するには、以下の方法も実行する必要があります。

- 認定システムのシステム管理者は、上記の制限を緩和するユーザーアプリケーションのセキュリティーパーミッションを提供しないようにする必要があります。
- バンドル jar 内に含まれるコードを含むデプロイメント内に含まれるコードが、クラス 'org.wildfly.security.manager.WildFlySecurityManager' によって提供される API に呼び出しを実行しないように、デプロイメントの前にアプリケーションを監査する必要があります。

6.3. ユーザー認証情報の開発者アドバイス

アウトバウンド呼び出しの認証を設定するには、**AuthenticationContext** を使用できます。このアウトバウンド呼び出しでは、SASL 認証が使用されます。

例: AuthenticationContextの設定

```
final AuthenticationContext authenticationContext = AuthenticationContext.empty()
.with(
```

```
MatchRule.ALL,
AuthenticationConfiguration.empty()
.useName("username")
.useRealm(null)
.usePassword("password"));
```

次に、**AuthenticationContext** は、このポリシーが使用されるようにアウトバウンド呼び出しをラップできます。

```
authenticationContext.runCallable(() -> {
    // Make Remote Call
}
```

第7章 セキュリティー機能の概要

7.1. ACCESS CONTROL (アクセス制御)

JBoss EAP 7 には以下のリクエストタイプへのアクセスを制限するアクセス制御メカニズムがあります。

- HTTP: URL で提供される URL およびパス、サーブレットおよびセッション Bean としてデプロイされた Plain Old Java Objects (POJO) はサブジェクトによるアクセスから保護できます。
  - URL へのアクセスを許可されたロールの名前を取得します。ロール名は、呼び出された URL に定義される **security-constraint** 要素と、任意で HTTP デプロイメント記述子または **@ServletSecurity** アノテーションの一部として HTTP リクエストメソッドによって決定されます。

    注記

    JBoss EAP は、RFC およびカスタムメソッドで指定されたすべての HTTP リクエストメソッドをサポートします。

    アクセス制御メカニズムでは、URL および HTTP リクエストメソッドの指定に加え、任意でユーザーの接続の暗号化保護が必要にります。これは、none、**integrity-protected**、または **confidentiality-protected** のいずれかです。

- EJB: EJB および関連するメソッド名がサブジェクトによって呼び出されないように保護できます。
  - EJB コンテナーから EJB メソッドへのアクセスを許可されたロールの名前を取得します。このロール名は、EJB デプロイメント記述子またはアノテーションに定義されているように呼び出しされたメソッドが含まれるすべての **method-permission** 要素の **role-name** 要素によって決定されます。

  - ロールが割り当てられていない場合や、exclude-list 要素にメソッドが指定されている場合は、メソッドへのアクセスは拒否されます。それ以外の場合は、呼び出し元にロール名のいずれかが割り当てられているかどうかを確認するために **doesUserHaveRole** メソッドが呼び出されます。このメソッドはロール名を繰り返し処理し、認証されたユーザーの **Subject Roles** グループに指定の名前のロールが含まれているかどうかを確認します。ロール名が **Roles** グループのメンバーである場合は、アクセスが許可されます。ロール名がメンバーでない場合、アクセスは拒否されます。

- JMS: メッセージキューの宛先およびトピック宛先は、サブジェクトによるアクセスから保護できます。
  - メッセージキューの宛先またはトピック宛先にアクセス可能なロールの名前を取得します。ロール名は、JBoss EAP 設定ファイルのメッセージキューの宛先またはトピック宛先に定義される **security-setting** 要素によって決定されます。

  - メッセージキューおよびトピック宛先へのアクセスは、以下のパーミッションを使用して制御されます。
    - **create-durable-queue** は、一致するアドレスの永続キューの作成をロールに許可します。

    - **delete-durable-queue** は、一致するアドレスの永続キューの削除をロールに許可します。

- **create-non-durable-queue** は、一致するアドレスの非永続キューの作成をロールに許可します。
- **delete-non-durable-queue** は、一致するアドレスの非永続キューの削除をロールに許可します。
- **Send** は、一致するアドレスへのメッセージの送信をロールに許可します。
- **consume** は、一致するアドレスにバインドされたキューからのメッセージの消費をロールに許可します。
- **manage** は、管理アドレスに管理メッセージを送信して管理操作を起動することをロールに許可します。

- TSF では、個々の宛先にアクセス制御ルールが指定されていない場合に、宛先へのアクセスを制御するグローバルのデフォルトアクセス制御ルールを指定できます。ロールが割り当てられていないか、宛先がグローバルアクセス制御ルールなど、アクセス制御ルールに対応していない場合は、メソッドへのアクセスは拒否されます。それ以外の場合は、呼び出し元にロール名のいずれかが割り当てられているかどうかを確認するために **doesUserHaveRole** メソッドが呼び出されます。このメソッドはロール名を繰り返し処理し、認証されたユーザーの **Subject Roles** グループに必要なロール名が含まれているかどうかを確認します。ロール名が **Roles** グループのメンバーである場合は、アクセスが許可されます。ロール名がメンバーでない場合、アクセスは拒否されます。

詳細は **設定ガイド**を参照してください。

7.2. 管理インターフェースのロールベースのアクセス制御

JBoss EAP 7 の管理インターフェース、管理 CLI、Web ベースの管理コンソールでは、JBoss EAP 7 の設定可能な側面をすべて管理するために JBoss EAP システム設定へのアクセスを許可しています。管理者は、ネットワークポート設定やコンテナー設定などの一般的なシステム側面にアクセスできます。さらに、コンテナーが提供するサービスの設定要素も管理されます。

アプリケーションアクセス制御などのアプリケーションの設定は、アプリケーションに含まれるデプロイメント記述子でアドレスが付けられます。そのため、通常は管理インターフェースからアプリケーション設定にアクセスできません。

管理インターフェースは特定のネットワークインターフェースにバインドできます。これにより、信頼できないユーザーが管理インターフェースにアクセスできないように管理インターフェースを管理 LAN に制限できます。管理者は管理インターフェースと対話できるようにするには、ログインする必要があります。管理者アカウントは他のユーザーアカウントとは別に管理されます。

管理ユーザーが実行できるオブジェクトに対する各アクションは、ロールベースのアクセス制御メカニズムに依存します。アクションは以下に分類されます。

- モデルから主な機能を持つモデル操作は、関連するランタイムサービスが結果として開始/停止されることがよくありますが、モデルから読み書きすることがよくあります。
- RPC 操作: ランタイム状態のみに影響するランタイムを呼び出します。これは、ランタイム状態を読み取るか、または変更される可能性があります。このモデルは影響を受けません。

オブジェクトアクション機能のセットは管理ロールにマップされます。このマッピングは、管理ロールの許可されるアクセスを定義します。事前定義済みの管理ロールのセットは JBoss EAP **PROD_VER** に含まれ、インストール後に利用できます。事前設定されたロールの詳細を以下に示します。

ロールベースのアクセス制御が事前設定された管理ロール

表7.1 ロールベースのアクセス制御

ロール	説明
Monitor	Monitor ロールには最も少ないパーミッションが設定され、ユーザーが設定と現在の状態を表示するように制限されます。monitor ロールには、サーバー設定を変更したり、機密データや機密操作にアクセスするパーミッションがありません。
Operator	Operator ロールは Monitor ロールを拡張し、サーバーのランタイム状態を変更する機能を追加しますが、永続設定はできません。たとえば、オペレーターはサーバーを起動または停止し、JMS 宛先を一時停止および再開できます。Operator ロールには、サーバー設定を変更するパーミッションや機密データや機密操作にアクセスするパーミッションがありません。
Maintainer	Maintainer ロールは、ランタイム状態と、機密データおよび機密操作を除くすべての設定を表示および変更できます。Maintainer ロールを持つユーザーには、パスワードやその他の機密情報へのアクセスを許可せずに、サーバー管理に必要なほぼ完全なアクセス権利を付与することができます。
Administrator	Administrator ロールは、監査ロギングシステムを除くサーバーのすべてのリソースおよび操作に無制限にアクセスできます。Administrator ロールは機密データおよび機密操作にアクセスでき、アクセス制御システムを設定することもできます。
SuperUser	SuperUser ロールには制限がなく、監査ロギングシステムと機密データを含むサーバーのすべてのリソースと操作に完全アクセスできます。RBAC が無効の場合、すべての管理ユーザーは SuperUser ロールと同等のパーミッションを持ちます。
Deployer	Deployer ロールは Monitor ロールと同じパーミッションを持ちますが、デプロイメントと、アプリケーションリソースとして有効になっている他のリソースタイプの設定および状態を変更できます。
Auditor	Auditor ロールは Monitor ロールのパーミッションをすべて持ちます。機密データも閲覧できますが、変更はできません。Auditor ロールは監査ロギングシステムにフルアクセスできます。

ロールは、パーミッションの名前付きセットです。これらのパーミッションには制約が含まれます。たとえば、Monitor ロールの読み取りパーミッションは機密でないアクションおよびターゲットに制限されます。上記の標準ロールに関連するパーミッションおよび制約の再定義は許可されません。

新しいロールの作成には限定的な形式があります。これらの新規ロールは標準的なロールと同じですが、すべてのパーミッションに追加の制約が適用されます。たとえば、ターゲットは特定のホストまたはサーバーグループに関連する必要があります。

すべての管理操作は設定ファイルに保存されます (スタートアップモードによって **domain.xml** または **standalone.xml** のいずれかになります)。管理インターフェースは、設定ファイルに保存されるデータのインメモリーイメージです。インメモリーイメージが変更されると、変更された設定ファイルが保存されます。

ロールベースのアクセス制御メカニズムは、管理 CLI または管理コンソールを使用して JBoss EAP

PROD_VER システム設定に管理者がアクセスする場合のみ強制できます。管理者がホストにシェルアクセスがある場合、基礎となるオペレーティングシステムは JBoss EAP システム設定ファイルへの直接読み取りまたは書き込みアクセスを付与する可能性があります。このようなアクセスにより、ロールベースのアクセス制御メカニズムは強制されないことになります。JBoss EAP システム設定ファイルに直接アクセスできない場合は、ホストが保護環境にあることを前提とします。

7.3. AUDIT

JBoss EAP 7 は、アクセス制御イベントの監査レコードを生成できます。Web リソースへのアクセス試行、EJB メソッドの呼び出し、承認されていないメッセージ宛先、および通常の web サービス関連のアクセス制御はすべてログに記録できます。管理者は、監査するイベントのレベルを選択できます。

監査機能は統合された log4j メカニズムに基づいています。log4j には **loggers**、**appenders**、および **layouts** の 3 つの主要コンポーネントがあります。これらの 3 種類のコンポーネントは連携し、開発者はメッセージの種別とレベルに応じてメッセージをログに記録でき、これらのメッセージのフォーマット方法とレポート先をランタイム時に制御できます。

監査情報はテキストファイルに記録されます。このファイルは、ページャーやエディターなどの基礎となるオペレーティングシステムのツールを使用して確認できます。監査レコードは、追加の監査制御のために syslog サーバーに転送することもできます。

ユーザー情報のプリンシパル名は、認証リクエストを記録する最初のログに ***のみ*** 表示され、認証に失敗した場合に生成される ERROR ログにも表示されます。後続のログイベントは、メソッドを実行するユーザーを明示的に記録しません。

ユーザー情報は、各監査ログに記録され、ユーザーセッションの有効期間中に保持されるコンテナーおよびスレッド ID を使用して取得できます。

7.4. クラスタリング

クラスターはノードセットです。JBoss EAP 7 クラスターでは、ノードは JBoss EAP 7 サーバーインスタンスです。クラスターを構築するには、複数の JBoss EAP 7 インスタンスをグループ化する必要があります (**partition** とも呼ばれます)。

クラスタリングは、複数の並列ノードでのアプリケーションの実行を可能にします。JBoss EAP 7 では、フェールオーバークラスターと負荷分散クラスターという 2 つのクラスターの概念が可能になります。どちらの場合も、サーバーの状態は異なるサーバー間で分散され、サーバーが失敗すると、他のクラスターノードからアプリケーションにアクセスできます。

クラスター通信は、異なるクラスターノード間のデータの一貫性を確立します。Jgroups および Infinispan は、JBoss EAP 7 クラスターの基盤となる通信、ノードのレプリケーション、およびキャッシュサービスを提供します。これらのサービスは MBean として設定されます。クラスタリングアプリケーションの各タイプの Infinispan および JGroups MBean のセットがあります (例: Stateful Session EJB、分散エンティティー EJB など)。

Infinispan は、JBoss EAP 7 クラスターに分散キャッシュおよび状態のレプリケーションサービスを提供します。JBoss EAP 7 クラスターには複数の Infinispan MBean を設定できます。1 つは HTTP セッションレプリケーション、ステートフルセッション Bean、キャッシュされたエンティティー bean 用などです。

以下の情報は、クラスター通信の一部として複製されます。

- ノードの状態のレプリケーションには、HTTP セッション、EJB 3.0 セッション Bean、EJB 3.0 エンティティー bean、および Hibernate 永続オブジェクト (Infinispan を使用して状態レプリケーションサービス) のレプリケーションが含まれます。

- HTTP セッションのレプリケーションと EJB 2.x セッション bean に対応するノードの状態レプリケーション。
- JBoss Messaging キューのレプリケーション。あるノードで分散キューまたはトピックに送信されたメッセージは、他のノードで消費できます。

JBoss EAP 7 は JNDI 状態の自動レプリケーションを実行しません。JNDI リソースを定義するアプリケーションが異なるクラスターノードにレプリケートされる場合、それらをノードに新たにデプロイします。このデプロイメントでは、JNDI リソースは通常のデプロイメントと同様に作成されます。JNDI リソースの変更が含まれるシステム設定の変更はクラスターノードにレプリケートされ、ローカルの再設定と同様に適用されます。JNDI マッピングを維持する JNDI レジストリーは、異なるクラスターノード間で一貫して管理されます。JNDI は JNDI レジストリー以外の状態を維持しないため、JNDI サービスのクラスター全体の一貫性を確保するために十分なものです。

7.5. 識別と認証

ユーザーは、アクセス制御の決定および監査のベースとして使用される一意のユーザー ID が割り当てられます。

JBoss EAP 7 は、ユーザーがさらなるセキュリティー調整されたアクションを実行できるようにする前に、ユーザーのアイデンティティーを認証します。

JBoss EAP 7 は、認証に成功した後にユーザーに対して作成されたスレッドに関連付けられた識別子を内部で維持します。

JBoss EAP 7 は、リクエストタイプごとに異なる識別および認証メカニズムを提供します。

- HTTP および web サービス: HTTP-basic 認証、HTTP-digest 認証、フォームベースの認証、クライアント証明書ベースの認証。
- EJB: ユーザー名およびパスワードベースの認証、クライアント証明書ベースの認証
- JMS: ユーザー名およびパスワードベースの認証。

7.6. トランザクションのロールバック

JBoss EAP 7 は操作のトランザクションへの集約をサポートします。これは一貫して適用およびロールバックできます。

トランザクションは、アトミック性、一貫性、分離、永続性 (ACID) プロパティーを持つ 1 つ以上の共有リソースが含まれる 1 つ以上の操作が含まれる作業単位です。これは、トランザクションの重要な 4 つのプロパティーです。

- atomicity: トランザクションはアトミックである必要があります。つまり、トランザクションで実行されたすべての作業を実行する必要があるか、またはそのどちらも実行する必要はありません。トランザクションの一部のみを実行することは許可されません。
- 一貫性: トランザクションが完了すると、システムが安定し一貫性のある状態である必要があります。
- 分離: 異なるトランザクションを相互に分離する必要があります。つまり、トランザクションがコミットされるまでは、1 つのトランザクションで部分的な作業が認識されず、マルチユーザーシステム内の各プロセスは、システムにアクセスする唯一のプロセスであるかのようにプログラムできます。

- 永続性: トランザクション中に追加された変更は、コミット時に永続化されます。トランザクションがコミットされると、サーバーがクラッシュしても変更は失われません。

JBoss EAP 7 のデフォルトのトランザクションマネージャーは JBoss Transactions です。

従来の ACID トランザクションシステムは、以下の特性を共有しています。

- トランザクションは短くなります。
- リソース (データベースなど) はトランザクションの期間ロックされます。
- 参加者は相互に信頼度が高い

インターネットと Web サービスの増加は、互いに認識していない参加者間で分散トランザクションを引き起こしました。JBoss Transactions は、最低限の作業で相互運用可能で信頼できるマルチユーザー、Web サービスベースのアプリケーションを構築するのに必要なコンポーネントを提供することにより、Web サービストランザクションのネイティブサポートを追加します。

プログラミングインターフェースは Java API for XML Transactions (JAXTX) をベースとし、WS-AtomicTransaction および WS-BusinessActivity 仕様のプロトコルサポートが含まれます。JBoss は、複数のコーディネーションプロトコルをサポートするように設計されています。

JBoss EAP 7 は、ローカルトランザクションと分散トランザクションの両方をサポートします。複数のプロセスインスタンス (仮想マシン) にまたがる場合は、トランザクションが分散されるとみなされます。通常、分散トランザクションには複数の仮想マシンに配置される参加者が含まれますが、トランザクションは別の仮想マシンで調整されるか、または参加者の 1 つと共存します。デプロイメントに分散トランザクションが必要な場合、Web サービストランザクションコンポーネント (SOAP/HTTP を使用する) を使用できます。