



## Red Hat Enterprise Linux 8

# RHEL 8 での Java のインストール、更新、および構成

RHEL 8 での Java アプリケーション開発の概要



# Red Hat Enterprise Linux 8 RHEL 8 での Java のインストール、更新、および構成

---

RHEL 8 での Java アプリケーション開発の概要

## 法律上の通知

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## 概要

本ガイドは、Java アプリケーション開発に最適なエンタープライズプラットフォームとして、Red Hat Enterprise Linux 8 を活用するさまざまな機能とユーティリティを説明します。

## 目次

|  |    |
|--|----|
| RED HAT ドキュメントへのフィードバック (英語のみ)                               | 3  |
| 第1章 RED HAT OPENJDK の概要                                      | 4  |
| 第2章 RHEL 8 への JAVA のインストール                                   | 5  |
| 2.1. YUM を使用して RHEL に JRE をインストール                            | 5  |
| 手順   | 5  |
| 2.2. アーカイブポータブルバンドルを使用して RHEL へ JRE をインストール                  | 5  |
| 手順   | 6  |
| 2.3. YUM を使用して RHEL 8 に JDK をインストール                          | 7  |
| 手順   | 7  |
| 2.4. アーカイブポータブルバンドルを使用して RHEL 8 に JDK をインストール                | 7  |
| 手順   | 7  |
| 2.5. YUM を使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール           | 8  |
| 前提条件   | 8  |
| 手順   | 8  |
| 2.6. アーカイブポータブルバンドルを使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール | 9  |
| 手順   | 9  |
| 2.7. YUM を使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール           | 9  |
| 手順   | 9  |
| 2.8. アーカイブポータブルバンドルを使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール | 10 |
| 第3章 RHEL 8 での JAVA の更新                                       | 11 |
| 3.1. YUM を使用して RHEL 8 で JAVA の更新                             | 11 |
| 手順   | 11 |
| 3.2. アーカイブを使用して RHEL 8 で JAVA の更新                            | 11 |
| 前提条件   | 11 |
| 手順   | 11 |
| 第4章 RHEL 8 での JAVA の構成                                       | 12 |
| 4.1. RHEL でシステム全体の JAVA バージョンを対話的に選択                         | 12 |
| 前提条件   | 12 |
| 手順   | 12 |
| 4.2. RHEL でシステム全体の JAVA バージョンを非対話的に選択                        | 13 |
| 前提条件   | 13 |
| 手順   | 13 |
| 4.3. 特定のアプリケーション用にインストールされている JAVA バージョンの選択                  | 14 |
| 前提条件   | 14 |
| 手順   | 14 |
| 4.4. システム全体の ZIP バンドルの JAVA バージョンの選択                         | 15 |
| 前提条件   | 15 |
| 手順   | 15 |
| 4.5. RHEL での JAVA_HOME 環境変数の構成                               | 16 |
| 前提条件   | 16 |
| 手順   | 16 |
| 4.6. RHEL での JAVA アプリケーションのヒープサイズの構成                         | 16 |
| 手順   | 16 |



## RED HAT ドキュメントへのフィードバック (英語のみ)

ご意見ご要望をお聞かせください。ドキュメントの改善点はございませんか。改善点を報告する場合は、以下のように行います。

- 特定の文章に簡単なコメントを記入する場合は、以下の手順を行います。
  1. ドキュメントの表示が **Multi-page HTML** 形式になっていて、ドキュメントの右上端に **Feedback** ボタンがあることを確認してください。
  2. マウスカーソルで、コメントを追加する部分を強調表示します。
  3. そのテキストの下に表示される **Add Feedback** ポップアップをクリックします。
  4. 表示される手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
  1. [Bugzilla](#) の Web サイトにアクセスします。
  2. Component で **Documentation** を選択します。
  3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
  4. **Submit Bug** をクリックします。

## 第1章 RED HAT OPENJDK の概要

OpenJDK の Red Hat ビルドは、Java Platform Standard Edition (Java SE) のオープンソース実装です。これは、アップストリームの OpenJDK 8u プロジェクトと 11u プロジェクトをベースとしており、OpenJDK バージョン 8 と 11 の両方に、Shenandoah Garbage Collector が含まれています。

- **マルチプラットフォーム** - OpenJDK が Windows および RHEL でサポートされるようになりました。これは、デスクトップ、データセンター、およびハイブリッドクラウド全体で単一の Java プラットフォームを標準化するのに役立ちます。
- **頻繁なリリース** - Red Hat は、OpenJDK 8 および 11 ディストリビューションに対して、毎年四半期ごとに JRE および JDK の更新を提供します。これは **rpm** ファイル、**msi** ファイル、**zip** ファイル、およびコンテナとして利用できます。
- **長期サポート** - Red Hat は、最近リリースされた OpenJDK 11 だけでなく、OpenJDK 7 および 8 をサポートします。サポートライフサイクルの詳細は、[「OpenJDK のライフサイクルおよびサポートポリシー」](#) を参照してください。
- **Java Web Start** - Red Hat OpenJDK は、Windows および RHEL 用の Java Web Start をサポートします。



## 第2章 RHEL 8 への JAVA のインストール

Java は、モバイルアプリケーションからデスクトップアプリケーション、Web アプリケーション、エンタープライズシステムまで、プラットフォームに依存しない幅広いアプリケーションを開発および実行するための環境です。Java は汎用プログラミング言語です。Red Hat は、OpenJDK と呼ばれる Java Platform SE (Standard Edition) のオープンソース実装を提供します。RHEL リポジトリに含まれています。

アプリケーションは、JDK (Java Development Kit) を使用して開発されます。アプリケーションは、JRE (Java ランタイム環境) および JDK に含まれている JVM (Java 仮想マシン) で実行されます。フットプリントが最小で、ユーザーインターフェースに必要なライブラリーが含まれていないヘッドレスバージョンの Java もあります。ヘッドレスバージョンは、**ヘッドレスサブパッケージ** にパッケージ化されています。



### 注記

JRE と JDK のどちらが必要かわからない場合は、JDK をインストールすることが推奨されます。



### 注記

短期的に対応している Java バージョンを取得するには、Enterprise Linux (EPEL) 用の追加パッケージを使用します。詳細は [EPEL](#) を参照してください。パッケージの名前は **java-latest-openjdk** です。

次のセクションでは、RHEL に Java をインストールする手順を説明します。

### 2.1. YUM を使用して RHEL に JRE をインストール

システムパッケージマネージャーの **yum** を使用して、OpenJDK JRE (Java Runtime Environment) をインストールできます。これには root 権限が必要です。

JRE は OpenJDK の一部です。

#### 手順

1. インストールするパッケージを指定して、**yum** コマンドを実行します。
  - JRE 8 の場合 - **java-1.8.0-openjdk**  
**\$ sudo yum install java-1.8.0-openjdk**
  - JRE 11 の場合 - **java-11-openjdk**  
**\$ sudo yum install java-11-openjdk**
2. インストールが機能することを確認します。

```
$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

### 2.2. アーカイブポータブルバンドルを使用して RHEL へ JRE をインストール

OpenJDK JRE (Java Runtime Environment) は、ZIP バンドルを使用してインストールできます。これは、Java 管理者が root 権限を持っていない場合に役立ちます。



### 注記

JRE を含む親ディレクトリーを作成し、汎用パスを使用して最新の JRE へのシンボリックリンクを作成することが推奨されます。これにより、新しいバージョンへのアップグレードが容易になります。

### 手順

1. [Linux 用の最新バージョンの JRE ZIP バンドルをダウンロードします。](#)
2. ZIP バンドルの内容を任意のディレクトリーに展開します。

```
$ mkdir ~/jres
$ cd ~/jres
$ tar -xf java-1.8.0-openjdk-1.8.0.242.b08-1.static.jre.openjdkportable.x86_64.tar.xz
```

3. アップグレードを容易にするために、JRE へのシンボリックリンクを使用して汎用パスを作成します。

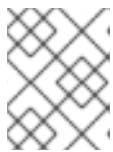
```
$ ln -s ~/jres/java-1.8.0-openjdk-1.8.0.242.b08-1.static.jre.openjdkportable.x86_64
~/jres/java-8
```

4. **JAVA\_HOME** 環境変数を設定します。

```
$ export JAVA_HOME=~/jres/java-8
```

5. **JAVA\_HOME** 環境変数が正しく設定されていることを確認します。

```
$ printenv | grep JAVA_HOME
JAVA_HOME=~/jres/java-8
```



### 注記

この方法でインストールした場合、Java は現在のユーザーのみが使用できません。

6. 一般的な JRE パスの **bin** ディレクトリーを **PATH** 環境変数に追加します。

```
$ export PATH="$JAVA_HOME/bin:$PATH"
```

7. フルパスを指定せずに **java -version** が機能することを確認します。

```
$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```



## 注記

~/**bashrc** に環境変数をエクスポートすることで、**JAVA\_HOME** 環境変数が現在のユーザーに対して持続することを確認できます。

## 2.3. YUM を使用して RHEL 8 に JDK をインストール

システムパッケージマネージャー (**yum**) を使用して、OpenJDK (Open Java Development Kit) をインストールできます。これには **root** 権限が必要です。

### 手順

1. インストールするパッケージを指定して、**yum** コマンドを実行します。
  - JDK 8 の場合 - **java-1.8.0-openjdk-devel**  
\$ **sudo yum install java-1.8.0-openjdk-devel**
  - JDK 11 の場合 - **java-11-openjdk-devel**  
\$ **sudo yum install java-11-openjdk-devel**
2. インストールが機能することを確認します。

```
$ javac -version
javac 1.8.0_242
```

## 2.4. アーカイブポータブルバンドルを使用して RHEL 8 に JDK をインストール

OpenJDK (Open Java Development Kit) は、ZIP バンドルを使用してインストールできます。これは、Java 管理者が **root** 権限を持っていない場合に役立ちます。



## 注記

JDK を含む親ディレクトリーを作成し、汎用パスを使用して最新の JDK へのシンボリックリンクを作成することが推奨されます。これにより、新しいバージョンへのアップグレードが容易になります。

### 手順

1. [Linux 用の最新バージョンの JDK ZIP バンドルをダウンロード](#)します。
2. ZIP バンドルの内容を任意のディレクトリーに展開します。

```
$ mkdir ~/jdk8
$ cd ~/jdk8
$ tar -xf java-1.8.0-openjdk-1.8.0.242.b08-1.static.jdk.openjdkportable.x86_64.tar.xz
```

3. アップグレードを容易にするために、JRE へのシンボリックリンクを使用して汎用パスを作成します。

```
$ ln -s ~/jdk8/java-1.8.0-openjdk-1.8.0.242.b08-1.static.jdk.openjdkportable.x86_64
~/jdk8/java-8
```

4. **JAVA\_HOME** 環境変数を設定します。

```
$ export JAVA_HOME=~/.jdk/java-8
```

5. **JAVA\_HOME** 環境変数が正しく設定されていることを確認します。

```
$ printenv | grep JAVA_HOME
JAVA_HOME=~/.jdk/java-8
```



### 注記

この方法でインストールした場合、Java は現在のユーザーのみが使用できません。

6. 一般的な JDK パスの **bin** ディレクトリーを **PATH** 環境変数に追加します。

```
$ export PATH="$JAVA_HOME/bin:$PATH"
```

7. 完全パスを指定しなくても **javac -version** が機能することを確認します。

```
$ javac -version
javac "1.8.0_242"
```



### 注記

~/.bashrc に環境変数をエクスポートすることで、**JAVA\_HOME** 環境変数が現在のユーザーに対して持続することを確認できます。

## 2.5. YUM を使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール

システムパッケージマネージャー (**yum**) を使用して、openJDK の複数のバージョンをインストールできます。これには root 権限が必要です。

### 前提条件

- インストールする OpenJDK バージョンを提供するリポジトリーへのアクセスを提供するアクティブなサブスクリプションを持つ Red Hat Subscription Manager (RHSM) アカウント。

### 手順

1. インストールするパッケージを指定して、**yum** コマンドを実行します。
  - JRE 8 の場合 - **java-1.8.0-openjdk**  
\$ **sudo yum install java-1.8.0-openjdk**
  - JRE 11 の場合 - **java-11-openjdk**  
\$ **sudo yum install java-11-openjdk**
2. インストールが機能することを確認します。

```
$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

■  
**java --alternatives** を使用して、使用するデフォルトの Java バージョンを構成できます。詳細は「[システムワイドの Java バージョンの選択](#)」を参照してください。

## 2.6. アーカイブポータブルバンドルを使用して RHEL に OPENJDK の複数のメジャーバージョンをインストール

OpenJDK の複数のメジャーバージョンをインストールするには、「[アーカイブポータブルバンドルを使用して RHEL へ JRE をインストール](#)」で同じ手順を使用するか、複数のメジャーバージョンを使用する「[アーカイブポータブルバンドルを使用して RHEL 8 に JDK をインストール](#)」を参照してください。



### 注記

システムのデフォルトの Java バージョンを構成する方法は、「[システム全体の ZIP バンドル Java バージョンの選択](#)」を参照してください。

## 2.7. YUM を使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール

RHEL には、OpenJDK の複数のマイナーバージョンをインストールできます。これは、インストールされているマイナーバージョンが更新されないようにすることで行われます。

### 手順

- `/etc/yum.conf` に **installonlypkgs** オプションを追加して、**yum** がインストールできても更新はできない Java パッケージを指定します。  
**installonlypkgs=java-<version>--openjdk,java-<version>--openjdk-headless,java-<version>--openjdk-devel**

更新は、システムに古いバージョンを残したまま、新しいパッケージをインストールします。

```
$ rpm -qa | grep java-1.8.0-openjdk
java-1.8.0-openjdk-1.8.0.242.b08-0.el8_1.x86_64
java-1.8.0-openjdk-headless-1.8.0.242.b08-0.el8_1.x86_64
```

- OpenJDK のさまざまなマイナーバージョンは、`/usr/lib/jvm/<minor version>` ファイルにあります。  
 たとえば、以下は `/usr/lib/jvm/java-1.8.0-openjdk-1.8.0` の一部を示しています。

```
$ /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-0.el8_1.x86_64/bin/java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)

$ /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.172-7.b11.el7.x86_64/bin/java -version
openjdk version "1.8.0_172"
OpenJDK Runtime Environment (build 1.8.0_172-b11)
OpenJDK 64-Bit Server VM (build 25.172-b11, mixed mode)
```

[システム全体の Java バージョンの選択](#) に従って、システム全体のバージョンを選択できます。

## 2.8. アーカイブポータブルバンドルを使用して RHEL に OPENJDK の複数のマイナーバージョンをインストール

複数のマイナーバージョンをインストールするのは、「[アーカイブポータブルバンドルを使用して RHEL へ JRE をインストール](#)」、または複数のマイナーバージョンを使用する「[アーカイブポータブルバンドルを使用して RHEL 8 に JDK をインストール](#)」するのと同じです。



### 注記

システムのデフォルトのマイナーバージョンを選択する方法は、「[システム全体の zip バンドルの Java バージョンの選択](#)」を参照してください。

## 第3章 RHEL 8 での JAVA の更新

以下のセクションでは、RHEL で Java を更新する方法を説明します。

### 3.1. YUM を使用して RHEL 8 で JAVA の更新

インストールされた Java パッケージは、**yum** システムパッケージマネージャーを使用して更新できます。これには root 権限が必要です。

#### 手順

1. 現在の Java バージョンを確認します。

```
$ sudo yum list installed "java*"
```

インストールされている Java パッケージの一覧が表示されます。

```
Installed Packages
java-1.8.0-openjdk.x86_64      1:1.8.0.242.b08-0.el8_1 @rhel-8-appstream-rpms
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.242.b08-0.el8_1 @rhel-8-appstream-rpms
```

2. 特定のパッケージを更新します。以下に例を示します。

```
$ sudo yum update java-1.8.0-openjdk
```

3. 現在の Java バージョンをチェックして、更新が機能していることを確認します。

```
# java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

### 3.2. アーカイブを使用して RHEL 8 で JAVA の更新

**ZIP** バンドルを使用して Java を更新できます。これは、Java 管理者が root 権限を持っていない場合に役立ちます。

#### 前提条件

- Java Development Kit (JDK) または Java Runtime Environment (JRE) のインストールを示す一般的なパスを把握している。たとえば、`~/jdk/java-11` になります。

#### 手順

1. JDK または JRE への汎用パスの既存のシンボリックリンクを削除します。以下に例を示します。

```
$ unlink ~/jdk/java-11
```

2. インストール場所に最新バージョンの JDK または JRE をインストールします。

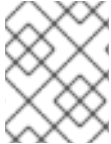
- JRE のインストール手順は、[「RHEL 8 への JRE のインストール」](#) を参照してください。
- JDK のインストール手順は、[「RHEL 8 への JDK のインストール」](#) を参照してください。

## 第4章 RHEL 8 での JAVA の構成

以下のセクションでは、RHEL で Java を構成する手順を説明します。

### 4.1. RHEL でシステム全体の JAVA バージョンを対話的に選択

RHEL に複数のバージョンの Java がインストールされている場合は、システム全体で使用するデフォルトの Java バージョンを対話形式で選択できます。これには root 権限が必要です。



#### 注記

root 権限がない場合は、「[JAVA\\_HOME 環境変数の設定](#)」で Java バージョンを選択できます。

#### 前提条件

- システムに対する root 権限がある。
- 複数のバージョンの Java が、**yum** パッケージマネージャーを使用してインストールされている。

#### 手順

1. システムにインストールされている Java のバージョンを表示します。  
**\$ yum list installed "java\*\*"**

インストールされている Java パッケージの一覧が表示されます。

```

Installed Packages
java-1.8.0-openjdk.x86_64           1:1.8.0.242.b08-0.el8_1           @rhel-8-
appstream-rpms
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.242.b08-0.el8_1           @rhel-8-
appstream-rpms
java-11-openjdk.x86_64             1:11.0.6.10-0.el8_1              @rhel-8-
appstream-rpms
java-11-openjdk-headless.x86_64    1:11.0.6.10-0.el8_1              @rhel-8-
appstream-rpms
javapackages-filessystem.noarch     5.3.0-1.module+el8+2447+6f56d9a6 @rhel-8-
8-appstream-rpms

```

2. 特定の Java コマンドに使用できる Java バージョンを表示し、使用するものを選択します。

```

$ sudo alternatives --config java
There are 2 programs which provide 'java'.

Selection    Command
-----
*+ 1         java-1.8.0-openjdk.x86_64 (/usr/lib/jvm/java-1.8.0-openjdk-1.8.0.242.b08-
0.el8_1.x86_64/jre/bin/java)
  2          java-11-openjdk.x86_64 (/usr/lib/jvm/java-11-openjdk-11.0.6.10-
0.el8_1.x86_64/bin/java)

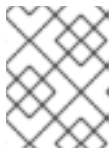
Enter to keep the current selection[+], or type selection number: 1

```



- 現在のシステム全体の Java バージョンにはアスタリスクが付いています。
  - 指定された Java コマンドの現在の Java バージョンには、プラス記号が付いています。
3. **Enter** キーを押して現在の選択を保持するか、選択する Java バージョンの **Selection** 番号を入力し、**Enter** キーを押します。  
システムのデフォルトの Java バージョンは、選択したバージョンです。
  4. 選択したバイナリーが選択されていることを確認します。

```
$ java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```



### 注記

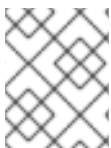
この手順では、**java** コマンドを構成します。次に、**javac** コマンドを同様の方法で設定できますが、独立して動作します。

JDK がインストールされている場合、**alternatives** はより多くの選択肢を提供します。特に、**javac** マスターの代替は、**-devel** サブパッケージにより提供される多くのバイナリーを切り替えます。JDK がインストールされている場合でも、**java** (およびその他の JRE マスター) と **javac** (およびその他の JDK マスター) は別々に動作するため、JRE と JDK に異なる選択をすることができます。

**alternatives --config java** コマンドは、**jre** とそれに関連付けられたスレーブに影響します。**JDK** を変更する場合は、**javac alternatives** コマンドを使用します。**--config javac** ユーティリティーは、**SDK** および関連するスレーブを構成します。可能なすべてのマスターを表示するには、**alternatives --list** を使用し、すべての **java** マスター、**javac** マスター、**jre** マスター、および **sdk** マスターを確認します。

## 4.2. RHEL でシステム全体の JAVA バージョンを非対話的に選択

RHEL に複数のバージョンの Java がインストールされている場合は、デフォルトの Java バージョンを選択して、非インタラクティブな方法でシステム全体で使用できます。これは、Red Hat Enterprise Linux システムで root 権限を持ち、自動化された方法で多くのシステムのデフォルト Java を切り替える必要がある管理者に役立ちます。



### 注記

root 権限がない場合は、「[JAVA\\_HOME 環境変数の設定](#)」で Java バージョンを選択できます。

### 前提条件

- システムに対する root 権限がある。
- 複数のバージョンの Java が、**yum** パッケージマネージャーを使用してインストールされている。

### 手順

1. 切り替える Java のメジャーバージョンを選択します。たとえば、Java 11 の場合は、`java-11-openjdk` を使用します。

```
# PKG_NAME=java-11-openjdk`
# JAVA_TO_SELECT=$(alternatives --display java | grep "family $PKG_NAME" | cut -d' ' -
f1)`
# alternatives --set java $JAVA_TO_SELECT`
```

2. アクティブな Java バージョンが指定したバージョンであることを確認します。

```
$ java -version
openjdk version "11.0.3" 2019-04-16 LTS
OpenJDK Runtime Environment 18.9 (build 11.0.3+7-LTS)
OpenJDK 64-Bit Server VM 18.9 (build 11.0.3+7-LTS, mixed mode, sharing)
```



#### 注記

Java 8 の場合は、**PKG\_NAME** を `java-1.8.0-openjdk` に設定します。



#### 注記

**javac** についても同様のアプローチが可能です。

### 4.3. 特定のアプリケーション用にインストールされている JAVA バージョンの選択

一部のアプリケーションでは、実行に特定の Java バージョンが必要です。**yum** パッケージマネージャーまたはポータブルバンドルを使用して、複数のバージョンの Java がシステムにインストールされている場合は、**JAVA\_HOME** 環境変数の値を設定するか、ラッパースクリプトを使用して、必要に応じて各アプリケーションの Java バージョンを選択できます。

#### 前提条件

- Java の複数のバージョンがマシンにインストールされている。
- 実行するアプリケーションがインストールされている。

#### 手順

1. **JAVA\_HOME** 環境変数を設定します。たとえば、**yum** を使用して `openjdk-11` がインストールされた場合は以下を使用します。

```
$ JAVA_HOME=/usr/lib/jvm/java-11-openjdk
```



#### 注記

シンボリックリンク `java-11-openjdk` は、**alternatives** コマンドにより制御されます。

2. 次のいずれかを行います。

- システム全体のデフォルト設定を使用してアプリケーションを起動します。

```
$ mvn --version
```

```

Apache Maven 3.5.4 (Red Hat 3.5.4-5)
Maven home: /usr/share/maven
Java version: 1.8.0_242, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-1.8.0-
openjdk-1.8.0.242.b08-0.el8_1.x86_64/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "4.18.0-147.3.1.el8_1.x86_64", arch: "amd64", family: "unix"

```

- **JAVA\_HOME** 変数を指定してアプリケーションを起動します。

```
$ JAVA_HOME=/usr/lib/jvm/java-11-openjdk mvn --version
```

```

Apache Maven 3.5.4 (Red Hat 3.5.4-5)
Maven home: /usr/share/maven
Java version: 11.0.5, vendor: Oracle Corporation, runtime: /usr/lib/jvm/java-11-openjdk-
11.0.5.10-0.el8_1.x86_64
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "5.4.12-200.el8_1.x86_64", arch: "amd64", family: "unix"

```

## 4.4. システム全体の ZIP バンドルの JAVA バージョンの選択

ZIP バンドルを使用して RHEL 8 に複数のバージョンの Java がインストールされている場合は、システム全体で使用する特定の Java バージョンを選択できます。

### 前提条件

- ZIP バンドルを使用してインストールされた Java バージョンの場所を把握している。

### 手順

1つのセッションに使用する Java バージョンを指定するには、以下を行います。

1. システム全体で使用する Java バージョンへのパスを使用して、**JAVA\_HOME** を構成します。  
**\$ export JAVA\_HOME=/opt/jdk/jdk-11.0.3**
2. **\$JAVA\_HOME/bin** を **PATH** 環境変数に追加します。  
**\$ export PATH="\$JAVA\_HOME/bin:\$PATH"**

シングルユーザーが永続的に使用する Java バージョンを指定するには、以下のコマンドを **~/.bashrc** に追加します。

```
export JAVA_HOME=/opt/jdk/jdk-11.0.3
export PATH="$JAVA_HOME/bin:$PATH"
```

すべてのユーザーに永続的に使用する Java バージョンを指定するには、次のコマンドを **/etc/bashrc** に追加します。

```
export JAVA_HOME=/opt/jdk/jdk-11.0.3
export PATH="$JAVA_HOME/bin:$PATH"
```



### 注記

**JAVA\_HOME** の正確な意味に注意してください。詳細は「[システム設定を java コマンドの設定から変更/分離](#)」を参照してください。**JAVA\_HOME** を再定義しない場合は、Java バイナリーへのパスを指定して、PATH コマンドのみを **bashrc** に追加します。たとえば、**export PATH="/opt/jdk/jdk-11.0.3/bin:\$PATH"** を指定します。

## 4.5. RHEL での JAVA\_HOME 環境変数の構成

一部のアプリケーションでは、Java インストールを検出できるように、**JAVA\_HOME** 環境変数を設定する必要があります。

### 前提条件

- システムで Java がインストールされている場所を把握している。たとえば、`/opt/jdk/11` です。

### 手順

- JAVA\_HOME** の値を設定します。  
`$ export JAVA_HOME=/opt/jdk/11`
- JAVA\_HOME** が正しく設定されていることを確認します。

```
$ printenv | grep JAVA_HOME
JAVA_HOME=/opt/jdk/11
```



#### 注記

シングルユーザーの場合は `~/.bashrc` に、システム全体の設定の場合は `/etc/bashrc` に環境変数をエクスポートして、**JAVA\_HOME** の値を永続的にすることができます。



#### 注記

**JAVA\_HOME** の正確な意味に注意してください。詳細は「[システム設定を java コマンドの設定から変更/分離](#)」を参照してください。

## 4.6. RHEL での JAVA アプリケーションのヒープサイズの構成

OpenJDK は、カスタマイズされたヒープサイズを使用するように構成できます。

### 手順

- アプリケーションの実行時に Java コマンドに最大ヒープサイズオプションを追加します。たとえば、最大ヒープサイズを 100 メガバイトに設定するには、`-Xmx100m` オプションを使用します。

OpenJDK 8 の場合:

```
$ java -Xmx100m <your-application>
```

OpenJDK 11 の場合は、新しい方法を使用することが推奨されます。

```
$ java -Xmn100m <your-application>
```

`Xmx` オプションの詳細は、[Java ドキュメント](#) の `-Xmxsize` を参照してください。