



Migration Toolkit for Applications 5.0

Migration Toolkit for Applications の概要

Migration Toolkit for Applications を使用して Java アプリケーションおよびコンポーネントを移行およびモダライズする方法

Migration Toolkit for Applications 5.0 Migration Toolkit for Applications の概要

Migration Toolkit for Applications を使用して Java アプリケーションおよびコンポーネントを移行およびモダライズする方法

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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Introduction_to_the_Migration_Toolkit_for_Applications.ent file |.

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[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

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

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] 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[®] 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

本ガイドでは、Migration Toolkit for Applications を始めるのに役立つ情報を提供します。

目次

多様性を受け入れるオープンソースの強化	3
第1章 はじめに	4
1.1. MIGRATION TOOLKIT FOR APPLICATIONS のご紹介	4
第2章 MIGRATION TOOLKIT FOR APPLICATIONS の概要	5
Migration Toolkit for Applications とは	5
Migration Toolkit for Applications で移行を単純化する方法	5
2.1. MTA の機能	5
2.2. MTA ルールについて	6
第3章 サポートされる設定	7
3.1. サポート対象の移行パス	7
3.2. システム要件	7
3.2.1. ソフトウェア	8
3.2.2. ハードウェア	8
第4章 ツールについて	9
4.1. CLI について	9
4.2. WEB コンソールについて	9
4.3. IDE プラグインについて	9
4.4. MAVEN プラグインについて	9
第5章 アプリケーション移行の計画	10
5.1. 移行を評価するゴール	10
5.2. RED HAT のアプリケーション移行アプローチ	10
5.2.1. ベストプラクティス	10
5.2.2. 方法	11
5.2.2.1. 発見フェーズ	11
5.2.2.2. 設計フェーズ	12
5.2.2.3. デプロイフェーズ	13
付録A 参考資料	14
A.1. 関連情報	14
A.1.1. 関わること	14
A.1.2. リソース	14
A.1.3. MTA の問題の報告	15
A.1.3.1. JIRA 問題の作成	15

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#)をご覧ください。

第1章 はじめに

1.1. MIGRATION TOOLKIT FOR APPLICATIONS のご紹介

Java アプリケーションやその他のコンポーネントを移行するために、Migration Toolkit for Applications (MTA) を使用するエンジニア、コンサルタント、およびその他のユーザーを対象としています。Migration Toolkit for Applications の概要と、移行を計画して実行するツールを使用して開始する方法を説明します。

第2章 MIGRATION TOOLKIT FOR APPLICATIONS の概要

Migration Toolkit for Applications とは

Migration Toolkit for Applications (MTA) は、Java アプリケーションの移行を簡素化する拡張可能でカスタマイズ可能なルールベースのツールです。

MTA は、プロジェクトソースディレクトリーやアプリケーションアーカイブを含むアプリケーションアーティファクトを検査し、変更を必要とするエリアを強調表示する HTML レポートを作成します。MTA は、以前のバージョンの Red Hat JBoss Enterprise Application Platform から、または Oracle WebLogic Server や IBM WebSphere Application Server などの他のアプリケーションサーバーから Java アプリケーションを移行できます。

Migration Toolkit for Applications で移行を単純化する方法

Migration Toolkit for Applications は一般的なリソースを探し、アプリケーションを移行する際のテクノロジーと既知の問題点を明らかにします。この目的は、アプリケーションが使用するテクノロジーの概要を提供し、組織がエンタープライズアプリケーションを Java EE および Red Hat JBoss Enterprise Application Platform に推定、文書化、移行するために使用できる詳細なレポートを提供することです。

2.1. MTA の機能

Migration Toolkit for Applications (MTA) は、移行プロジェクトの計画および実行に役立つ多くの機能を提供します。

計画および作業見積

MTA は、作業の種類を詳細に説明し、タスクを完了する作業の見積もりを行うことで、プロジェクトマネージャーを支援します。作業レベルは MTA レポートでストーリーポイントとして表されます。実際の見積もりは、必要なスキルと必要な移行作業の分類に基づいて行われます。

移行の問題の特定および解決策の提供

MTA は移行の問題を特定し、問題が発生したコードの特定ポイントを強調表示します。MTA は、コードの変更を提案し、エンジニアが特定の問題を解決するのに役立つ追加のリソースを提供します。

詳細なレポート

MTA は、移行作業の概要と特定の移行タスクの詳細の両方を示すために、多数のレポートを作成します。すべてのアプリケーション、チャート、およびアプリケーションの問題に関するサマリー情報、アプリケーションのモジュールの問題の内訳、使用される技術のレポート、他のアプリケーションおよびサービスの依存関係などを表示できます。ソースファイルを検査して、問題が発生したコードの行を確認することもできます。利用可能な MTA レポートの詳細は、[CLI ガイド](#) を参照してください。

組み込みルールおよび移行パス

MTA には、いくつかの一般的な移行パスに移行を支援するためのコアルールセットが付属しています。これらのルールは、他のアプリケーションサーバーからのプロプライエタリ機能の使用、または以前のバージョンの JBoss EAP で非推奨のサブシステムの使用を特定します。MTA には、ハードコーディングされた IP アドレスや JNDI ルックアップなどの一般的な移行の問題を識別するルールも含まれています。

ルールの拡張性とカスタマイズ

MTA は、強力で複雑なルールを作成する機能を提供します。MTA が提供するルールのコアセットに拡張し、ルールを作成して、移行プロジェクトにとって重要な追加の問題を特定することができます。コアルールを上書きし、カスタムルールカテゴリーを作成することも可能です。MTA ルールのカスタマイズに関する詳細は、[ルール開発ガイド](#) を参照してください。

ソースコードまたはアプリケーションアーカイブを分析する機能

MTA は、アプリケーションアーカイブまたはソースコードを評価でき、複数のアプリケーションをまとめて評価できます。複数のアプリケーション間で共有されるアーカイブを特定できるため、より正確な作業量の見積りに役立ちます。

2.2. MTA ルールについて

Migration Toolkit for Applications (MTA) には、移行予定のアプリケーションが使用する API、テクノロジー、アーキテクチャーを分析するルールベースの移行ツールが含まれています。実際、MTA 分析プロセスは MTA ルールを使用して実装されます。MTA は内部でルールを使用して、アーカイブからのファイルの抽出、ファイルの逆コンパイル、ファイルタイプのスキャンと分類、XML およびその他のファイルコンテンツの分析、アプリケーションコードの分析、レポートの作成を行います。

MTA は、ルール実行結果に基づいてデータモデルを構築し、グラフデータベースにコンポーネントデータと関係を格納します。これにより、移行ルールやレポート目的に応じてクエリーおよび更新が可能になります。

MTA ルールでは、以下のルールパターンを使用します。

```
when(condition)
  perform(action)
otherwise(action)
```

MTA は、標準の包括的な移行ルールの包括的なセットをそのまま使用できます。アプリケーションにはカスタムライブラリーまたはコンポーネントが含まれる可能性があるため、MTA では独自のルールを作成して、既存のルールセットで対応していない可能性のあるコンポーネントやソフトウェアの使用を特定することができます。

独自のカスタムルールを作成する場合は、詳細な手順について、[ルール開発ガイド](#) を参照してください。

第3章 サポートされる設定

3.1. サポート対象の移行パス

Migration Toolkit for Applications (MTA) は、Oracle WebLogic Server などのサードパーティーのエンタープライズアプリケーションサーバーから JBoss Enterprise Application Platform (JBoss EAP) への移行と、JBoss EAP の最新リリースへのアップグレードをサポートしています。

MTA は、Red Hat OpenShift Container Platform でのコンテナ化およびデプロイメントのアプリケーションの適合性を評価するための包括的なルールセットを提供します。MTA 分析を実行して、複数のターゲットプラットフォームに移行するアプリケーションの適切性を評価することができます。

以下の表は、最も一般的な移行パスを示しています。

表3.1 サポート対象の移行パス: ソースプラットフォーム ⇒ ターゲット

ソースプラットフォーム ⇒	JBoss EAP 6	JBoss EAP 7	OpenShift Container Platform	OpenJDK 8 & 11	Apache Camel 3	Spring Boot on Red Hat Runtimes
Oracle WebLogic Server	✓	✓	✓	✓	-	-
IBM WebSphere Application Server	✓	✓	✓	✓	-	-
JBoss EAP 4	✓	✗ [a]	✓	✓	-	-
JBoss EAP 5	✓	✓	✓	✓	-	-
JBoss EAP 6	該当なし	✓	✓	✓	-	-
JBoss EAP 7	該当なし	✓	✓	✓	-	-
Oracle JDK	-	-	✓	✓	-	-
Apache Camel 2	-	-	✓	✓	✓	-
Spring Boot	-	-	✓	✓	✓	-
すべての Java アプリケーション	-	-	✓	✓	-	-
[a] 現在 MTA はこの移行パスのルールを提供していませんが、Red Hat コンサルティングはソースプラットフォームから JBoss EAP 7 への移行を支援します。						

3.2. システム要件

3.2.1. ソフトウェア

- OpenJDK 1.8、OpenJDK 11、Oracle JDK 1.8、Oracle JDK 11
- MTA は Linux、Windows、および macOS でテストされています。Java 8+ サポートのあるその他のオペレーティングシステムは適切に動作するはずです。

3.2.2. ハードウェア

以下のメモリーおよびディスク領域の要件は、MTA の実行に必要な最小要件です。アプリケーションが非常に大きい場合や複数のアプリケーションを評価する必要がある場合には、これらの値を増やしてパフォーマンスを向上させることができます。

パフォーマンスを最適化する方法は、MTA の [CLI ガイド](#) の [MTA パフォーマンスの最適化](#) セクションを参照してください。

- 最小 4 GB の RAM。パフォーマンスを向上させるために、RAM が 8 GB のクアッドコアプロセッサを推奨します。これにより、3 ~ 4 GB の RAM が JVM で使用できるようになります。
- 最小 4 GB の空きディスク領域高速ディスク (特にソリッドステートドライブ (SSD)) はパフォーマンスを向上する必要があります。

第4章 ツールについて

Migration Toolkit for Applications (MTA) は、移行およびモダライゼーションのさまざまな段階を支援するツールを提供します。各ツールの詳細を確認し、プロジェクトに適したものを特定します。

- [CLI](#)
- [Web コンソール](#)
- [IDE プラグイン](#)
- [Maven プラグイン](#)

4.1. CLI について

CLI は、Migration Toolkit for Applications のコマンドラインツールです。これにより、ユーザーはアプリケーションに対する移行およびモダライゼーションの作業を評価および優先順位付けできます。他のツールのオーバーヘッドなしに分析を強調表示する多数のレポートが提供されます。CLI にはさまざまなカスタマイズオプションが含まれており、MTA 分析オプションを細かく調整したり、外部の自動化ツールと統合したりできます。

CLI の使用に関する詳細は、MTA の [CLI ガイド](#) を参照してください。

4.2. WEB コンソールについて

Migration Toolkit for Applications の Web コンソールを使用すると、多数のアプリケーションに対する移行およびモダライゼーションの作業を評価および優先順位付けできます。これにより、分析のためにアプリケーションをプロジェクトにグループ化し、結果を強調表示する多数のレポートを提供します。

Web コンソールの使用方法は、MTA の [Web コンソールガイド](#) を参照してください。

4.3. IDE プラグインについて

Migration Toolkit for Applications の IDE プラグインは、移行またはモダライゼーションの作業に変更を加える開発者向けに、Eclipse と Red Hat CodeReady Studio を直接サポートを提供します。MTA を使用してプロジェクトを分析し、ソースコード内の移行問題をマークし、問題を修正するためのガイダンスを提供して、可能な場合は自動コードの置き換えを提供します。

Visual Studio Code および Eclipse Che には MTA 拡張があります。この拡張機能は Minimum Viable Product としてのみ利用可能であり、今後の MTA リリースで完全に文書化され、サポートされます。[拡張機能](#) の詳細は、Visual Studio Marketplace にアクセスしてください。

IDE プラグインの使用方法は、MTA の [IDE プラグインガイド](#) を参照してください。

4.4. MAVEN プラグインについて

Migration Toolkit for Applications の Maven プラグインは Maven ビルドプロセスに統合されるため、開発者はソースコードの反復ごとに移行およびモダライゼーション作業を継続的に評価できます。分析結果を強調表示するレポートが多数提供され、ビルドごとに更新が必要な開発者向けに設計されています。

Maven プラグインの使用方法は、MTA の [Maven プラグインガイド](#) を参照してください。

第5章 アプリケーション移行の計画

5.1. 移行を評価するゴール

多くの組織は、既存のビジネスオペレーションの実行を継続しつつ、イノベーションを試みるという課題に直面します。新しい機能をより迅速に提供し、コストを削減することへの期待が高まっています。

移行またはモダナイゼーションの評価時に、組織固有の課題に対応することが重要です。組織が直面する一般的な課題は、限られた予算、社内スキル、認識されたリスク、既知の予測可能なプロセス、およびタイムリーな方法での正確な作業の見積もりの欠如です。

一般的には、大規模なミドルウェアの移行またはモダナイゼーションの作業を評価する際のゴールは以下のとおりです。

- 作業レベルとコストの予測
- アプリケーション移行のスケジュールと競合の処理
- コード、インフラストラクチャー、プロセス、またはナレッジレベルで潜在的なリスクを特定
- ビジネスケースを作成するための投資回収の予測
- ビジネスへのリスクの特定と軽減
- 既存のビジネスオペレーションの中断を最小限に抑える

Red Hat は、リスクを最小限に抑え、アプリケーションの移行やモダナライゼーションをより効率的にする戦略を開発しました。

5.2. RED HAT のアプリケーション移行アプローチ

Red Hat は、Red Hat JBoss Middleware の定量化が可能で、コストが低く、リスクを低減し、完了しやすくなるように、大規模なアプリケーションを Red Hat JBoss Middleware に移行する戦略を定義しました。

プロプライエタリーまたは以前のミドルウェアプラットフォームから Red Hat JBoss ミドルウェアに移行すると、軽量でモジュール化されたクラウド対応のミドルウェアに対応したアプリケーションインフラストラクチャーを利用でき、チームの生産性が向上します。JBoss Middleware およびその他の Red Hat テクノロジーは、アプリケーション開発とデリバリーを最新化するために非常に大きな利点を提供するため、より迅速にイノベーションが可能になります。

5.2.1. ベストプラクティス

Red Hat は、移行またはモダナイゼーションのプランニングおよび実行時に、以下のプラクティスを推奨します。

- 情報を共有する集中型コラボレーションプラットフォームを作成します。
包括的でアクセス可能なドキュメントは、知識を簡単に共有し、同じ問題を 2 回解決することで労力が重複しないようにするために重要です。以下の項目を文書化することを推奨します。
 - アプリケーションをゼロから移行したり、最新の状態にするためのステップごとのガイドです。
 - 既知の問題および発生した問題の解決策のコレクション。

- 新しいプラットフォームに関する情報
- 特定のスケジュールプロジェクトに加えられた変更の記録。
- 可能な限りの再利用、自動化、および標準化を行います。
新規コンポーネントを作成する代わりに、既存コンポーネントを再利用することを検討してください。ビルド、設定、デプロイメント、テストなどのアプリケーションライフサイクルに関連するプロセスを自動化します。ソフトウェアパッケージ形式、設定管理、ライブラリー、および依存関係を標準化し、文書化します。
- 実証済みの反復可能な方法論を使用します。
これは、実用的なアプローチに従い、機能的に同一アプリケーションを取得するためにできるだけ変更を加えることが推奨されます。
- 選択した Red Hat Middleware コンポーネントについて早期に Red Hat の技術専門知識が必要になります。
これは、移行とモダナイゼーションを低リスク、予測可能、かつ効率的にするために重要です。Red Hat コンサルティングにご連絡ください。

5.2.2. 方法

Red Hat が推奨する方法は、段階的に計画し、移行またはモダナイゼーションを行うのに役立つ、スケーラブルなアプローチです。

図5.1 Red Hat 移行の方法



このアプローチは、以下のフェーズで設定されます。

発見

テクノロジーを調査し、懸念事項、組織の要件、課題を特定します。オプションと考えられるアプローチについて話し合います。

設計

アプリケーション、インフラストラクチャー、プロセス、およびナレッジの観点から最も重要なリスクを特定し、軽減します。ソリューションのストラテジーを確立します。テクノロジーを証明し、ビジネスケースを構築します。

デプロイ

明確に定義された配信モデルに従って、以前に定義したストラテジーを実行し、モダナイゼーションまたは移行を繰り返しスケーリングします。

5.2.2.1. 発見フェーズ

図5.2 Red Hat 移行の方法: 発見フェーズ



発見 フェーズでは、すべての利害関係者と意思決定者を集めて、現在の状態とビジネスドライバーを理解し、移行またはモダナイゼーションのニーズを決定します。

このフェーズでは、テクノロジーを調査し、潜在的なアプローチについて話し合います。既存の問題点、懸念事項、要件、および潜在的な課題を特定します。ビジネス優先順位の概要を定義し、評価の範囲を定めます。アプリケーションの開発と配信を最新にして、より迅速に革新できるようにする方法を決定します。

通常、これは Red Hat エキスパートとのワークショップで1日で完了します。

5.2.2.2. 設計フェーズ

図5.3 Red Hat 移行の方法: 設計フェーズ



設計 フェーズでは、すべてのリスクを特定し、ストラテジーとターゲットアーキテクチャーを定義して、テクノロジーを証明し、ビジネスケースを構築します。このフェーズは、以下の手順で設定されます。

評価

既存のインフラストラクチャー、アーキテクチャー、テクノロジー、およびアプリケーションを確認します。依存関係、知識、プロセス、ライフサイクルを特定します。必要なインフラストラクチャー、アーキテクチャー、テクノロジー、およびアプリケーションを定義します。可用性と潜在的なリスクを決定します。計画を作成し、大まかな見積もりを提供します。

Migration Toolkit for Applications Web コンソールまたは CLI を使用してアプリケーションを分析すると、依存関係、潜在的なリスク、および相対作業の判断に役立ちます。これらのツールの使用方法は、[Web コンソールガイド](#) および [CLI ガイド](#) を参照してください。

証明

特定された技術的リスク、たとえば、高リスクの項目や未知の努力による問題を解決して文書化します。新しいプラットフォームインフラストラクチャーを構築して形成します。結果に基づいて見積もりを改良します。

パイロット

変換を試行するための小規模のアプリケーションを選択します。必要に応じて、ターゲットプラットフォームのインフラストラクチャーを完成させます。プロセスが微調整され、ドキュメントを更新します。このパイロットは、移行フェーズの1つまたは複数の反復で設定され、デプロイフェーズ中にスケーリングされます。

Migration Toolkit for Applications IDE プラグインを使用してコードの移行を加速します。IDE プラグインの使用方法は、[IDE プラグインガイド](#) を参照してください。

計画

前の手順の結果に基づいて、見積もりをより正確にし、プロジェクト計画を改善します。移行を完了するために、デプロイフェーズで使用するロールアウトストラテジーを定義します。関連する [テクノロジー支援コース](#) を準備してスケジュールします。

5.2.2.3. デプロイフェーズ

図5.4 Red Hat 移行の方法: デプロイフェーズ



デプロイ フェーズは、設計フェーズで作成したプランを実行する場合です。このフェーズでは、全体的な変換プロセスをスケーリングして計画を完成させ、すべてのアプリケーションを新しい本番環境に導入します。

計画は反復的に実行され、価値を段階的に提供します。反復ごとに、計画に対して継続的に検証し、次のスプリントを改善するために調査結果を文書化します。

Migration Toolkit for Applications の IDE プラグインに使用すると、反復ごとに速度が向上します。Eclipse または Red Hat CodeReady Studio で使用でき、移行の問題をソースコードで直接マークし、オンラインのヒントを提供して、コード変更の提案を提供します。IDE プラグインの使用方法は、[IDE プラグインガイド](#) を参照してください。

付録A 参考資料

A.1. 関連情報

A.1.1. 関わること

Migration Toolkit for Applications が、自分のものを含むほとんどのアプリケーション設定とサーバー設定に対応できるように、以下の項目のいずれかを支援できます。

- jboss-migration-feedback@redhat.com にメールを送信し、MTA 移行ルールが対象とすべき内容をご連絡ください。
- 移行ルールをテストするためのアプリケーションの例を指定します。
- 移行が困難なアプリケーションコンポーネントおよび問題の領域を特定してください。
 - これらの問題がある移行領域について簡単な説明を記入する。
 - 問題の移行領域を解決する方法を説明する簡単な概要を記述します。
- アプリケーションで Migration Toolkit for Applications を試行します。発生している問題を必ず報告してください。
- Migration Toolkit for Applications ルールリポジトリへの貢献。
 - Migration Toolkit for Applications ルールを記述して、移行プロセスを識別または自動化します。
 - 新規ルールのテストを作成します。
 - 詳細は、[ルール開発ガイド](#) を参照してください。
- プロジェクトのソースコードへの貢献。
 - コアルールを作成します。
 - MTA のパフォーマンスまたは効率が向上します。
 - 環境の設定およびプロジェクトの設定方法に関する詳細は、[Core Development Guide](#) を参照してください。

あらゆるレベルの貢献が大きく評価されます。

A.1.2. リソース

- MTA フォーク: <https://developer.jboss.org/en/windup>
- MTA JIRA 問題トラッカー
 - コア MTA: <https://issues.jboss.org/browse/WINDUP>
 - MTA ルール: <https://issues.jboss.org/browse/WINDUPRULE>
- MTA メーリングリスト: jboss-migration-feedback@redhat.com
- MTA IRC チャンネル: Server FreeNode (irc.freenode.net) (チャンネル [#windup](#) ([transcripts](#)))

A.1.3. MTA の問題の報告

Migration Toolkit for Applications は、問題追跡システムとして JIRA を使用します。MTA の実行で問題が発生した場合は、JIRA Issue を作成してください。



注記

JIRA ユーザーアカウントがない場合は、JIRA の課題を作成するためにアカウントを作成する必要があります。

A.1.3.1. JIRA 問題の作成

1. ブラウザーを開き、JIRA [Create issue](#) ページに移動します。
ログインしていない場合は、ページの右上にある **Log In** リンクをクリックし、認証情報を入力します。
2. 以下のオプションを選択し、**Next** ボタンをクリックします。
 - **プロジェクト**
コア MTA の問題の場合は、**Migration Toolkit for Applications (WINDUP)**を選択します。

MTA ルールの問題は、**Migration Toolkit for Applications rules (WINDUPRULE)**を選択します。
 - **問題のタイプ: バグ**
3. 次の画面では、以下のフィールドを入力します。
 - **Summary:** 問題またはその問題の簡単な説明を入力します。
 - **Environment:** オペレーティングシステム、Java のバージョン、およびその他の関連情報の詳細を指定します。
 - **Description:** 問題の詳細情報を指定します。ログと例外トレースを必ず含めてください。
 - **Attachment:** 問題を生じるアプリケーションまたはアーカイブに機密情報が含まれておらず、MTA 開発チームと共有する場合は、**browse** ボタンを使用して問題を添付します。
4. **Create** ボタンをクリックして JIRA の問題を作成します。

改訂日時: 2022-12-03 13:21:01 +1000