



# Migration Toolkit for Runtimes 1.2

## Web 控制台指南

使用 Migration Toolkit for Runtimes web 控制台将您的应用程序分组到项目中进行分析。



## Migration Toolkit for Runtimes 1.2 Web 控制台指南

---

使用 Migration Toolkit for Runtimes web 控制台将您的应用程序分组到项目中进行分析。

## 法律通告

Copyright © 2024 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.

## 摘要

本指南论述了如何使用 Migration Toolkit for Runtimes web 控制台来简化 Java 应用程序的迁移或现代化。

---

# 目录

<b>使开源包含更多</b> .....	<b>3</b>
<b>第 1 章 简介</b> .....	<b>4</b>
1.1. 关于 WEB 控制台	4
1.2. 关于 MIGRATION TOOLKIT FOR RUNTIMES	4
1.3. MTR WEB 控制台	4
<b>第 2 章 安装 WEB 控制台</b> .....	<b>5</b>
2.1. 在 LINUX、WINDOWS 或 MACOS 上安装 WEB 控制台	5
2.2. 在 OPENSIFT CONTAINER PLATFORM 4.11 及之后的版本上安装 WEB 控制台	6
<b>第 3 章 使用 WEB 控制台分析应用程序</b> .....	<b>10</b>
3.1. 创建一个项目	10
3.2. 运行保存分析	12
3.3. 查看分析结果	13
3.4. 检查报告	14
3.5. 更新分析配置	14
3.6. 添加全局自定义规则	15
3.7. 添加全局自定义标签	15
<b>第 4 章 在 LINUX、WINDOWS 或 MACOS 上为 WEB 控制台配置身份验证</b> .....	<b>17</b>
启用身份验证	17
禁用身份验证	18



## 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

# 第 1 章 简介

## 1.1. 关于 WEB 控制台

本指南适用于希望使用 Migration Toolkit for Runtimes (MTR) 迁移 Java 应用程序或其他组件的工程师、顾问和其他组件。它介绍了如何安装和使用 Web 控制台来管理迁移或现代化项目并分析应用程序。

## 1.2. 关于 MIGRATION TOOLKIT FOR RUNTIMES

### 什么是 Migration Toolkit for Runtimes?

Migration Toolkit for Runtimes (MTR) 是一个可扩展、可自定义的基于规则的工具，简化了 Java 应用程序的迁移和现代化。

MTR 检查应用程序工件，包括项目源目录和应用程序存档，然后生成 HTML 报告突出显示需要更改的区域。MTR 支持许多迁移路径，包括以下示例：

- 升级至 Red Hat JBoss Enterprise Application Platform 的最新版本
- 从 Oracle WebLogic 或 IBM WebSphere Application Server 迁移到 Red Hat JBoss Enterprise Application Platform
- 容器化应用程序并使之成为云就绪
- 从 Java Spring Boot 迁移到 Quarkus
- 从 Oracle JDK 更新至 OpenJDK
- 从 OpenJDK 8 升级到 OpenJDK 11
- 从 OpenJDK 11 升级到 OpenJDK 17
- 从 OpenJDK 17 升级到 OpenJDK 21
- 将 EAP Java 应用程序迁移到 Azure
- 将 Spring Boot Java 应用程序迁移到 Azure

有关用例和迁移路径的更多信息，请参阅[开发人员网页的 MTR](#)。

### Migration Toolkit for Runtimes 如何简化迁移？

Migration Toolkit for Runtimes 会查找常见资源和在迁移应用程序时的已知问题。它为应用程序使用的技术提供了高级视图。

MTR 生成详细的报告，评估迁移或现代化路径。此报告可帮助您估算大型项目所需的工作量，并减少涉及的工作。

### 如何了解更多信息？

请参阅 [Migration Toolkit for Runtimes 简介](#)，了解有关 Migration Toolkit for Runtimes 中的功能、支持的配置、系统要求以及可用工具的更多信息。

## 1.3. MTR WEB 控制台

通过将 Web 控制台用于 Migration Toolkit for Runtimes，一组用户可以对大量应用程序的迁移和现代化工作进行评估和优先级。您可以使用 Web 控制台将应用程序分组到项目中进行分析，并提供大量突出显示结果的报告。

## 第 2 章 安装 WEB 控制台

您可以在 Linux、Windows、macOS 或 Red Hat OpenShift Container Platform 上安装 Web 控制台。

### 2.1. 在 LINUX、WINDOWS 或 MACOS 上安装 WEB 控制台

您可以在 Linux、Windows 或 macOS 操作系统上安装 Web 控制台，并在浏览器中访问 Web 控制台。

Web 控制台已使用 Chrome 和 Firefox 进行测试。

#### 先决条件

以下是 Migration Toolkit for Runtimes (MTR) 安装的先决条件：

- 安装了 Java Development Kit (JDK)。MTR 支持以下 JDK：
  - OpenJDK 11
  - OpenJDK 17
  - Oracle JDK 11
  - Oracle JDK 17
  - Eclipse Temurin™ JDK 11
  - Eclipse Temurin™ JDK 17
- 8 GB RAM
- macOS 安装：**maxproc** 的值必须是 **2048** 或更高版本。

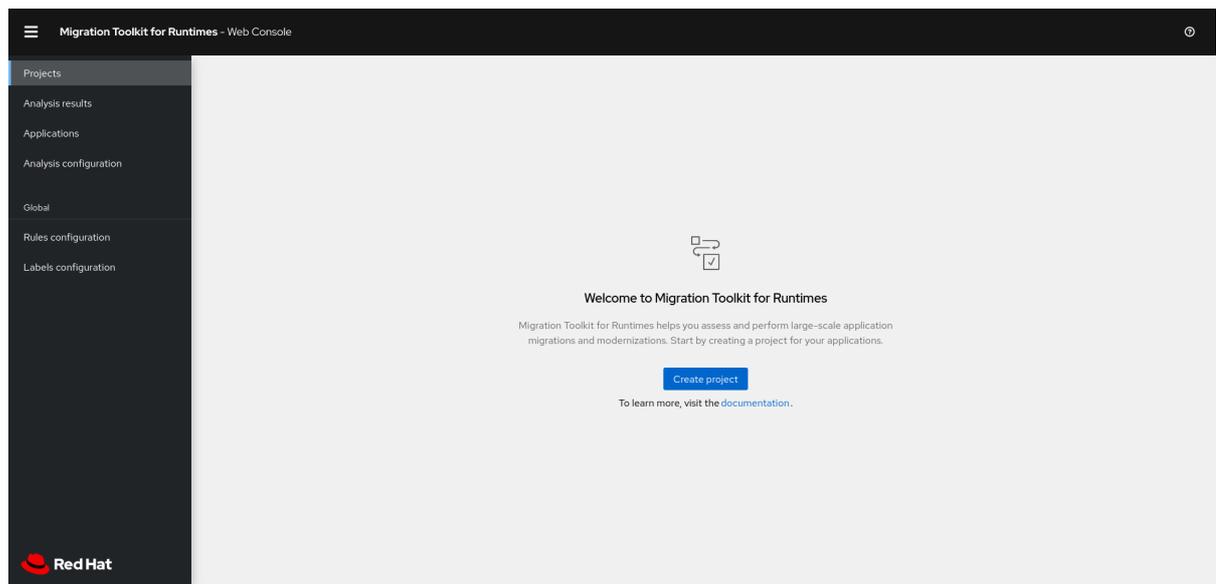
#### 流程

1. 进入到 [MTR Download 页面](#)，再下载 Web 控制台 **Local install & OpenShift** 文件。
2. 将 **.zip** 文件提取到您选择的目录。  
在本指南中遇到 **<MTR\_HOME>** 时，将其替换为 MTR 安装的实际路径。
3. 默认情况下，不需要身份验证。如果要启用身份验证，请在启动 Web 控制台前执行此操作。
4. 启动 Web 控制台：
  - Linux 或 macOS 操作系统：

```
$ <MTR_HOME>/run_windup.sh
```
  - Windows 操作系统：

```
C:\<MTR_HOME>\run_windup.bat
```
5. 打开浏览器并进入到 **http://localhost:8080/windup-ui**。  
如果没有配置身份验证，则您的浏览器中会显示 Projects 屏幕。

图 2.1. MTR Projects 屏幕



## 2.2. 在 OPENSIFT CONTAINER PLATFORM 4.11 及之后的版本上安装 WEB 控制台

您可以使用 Migration Toolkit for Runtimes Operator 在 OpenShift Container Platform 4.11 和更新的版本上安装 Web 控制台。

### 先决条件

- 6 个 vCPU、8 GB RAM 和 40 GB 持久性存储。
- 您可以在其中安装 Web 控制台的一个或多个项目。



### 重要

不要在 default 项目中安装 Web 控制台。

- **cluster-admin** 权限，用来为 Runtimes Operator 安装 Migration Toolkit。
- **project-admin-user** 特权，以便在项目中安装 Web 控制台应用程序。

### 配置 Red Hat Single Sign-on (SSO)

您必须在安装时决定 Web 控制台是否需要身份验证。如果这样做，您必须首先安装和配置 Red Hat SSO，并在实例化 MTR Operator 时输入一些 RH SSO 设置。



### 注意

在安装后无法添加或删除身份验证。

1. 打开 Red Hat SSO 管理控制台。
2. 添加名为 **windup** 的域。
3. 在域中，创建一个名为 **windup-web** 的客户端。

4. 检查 **Access Type** 是否已设置为 **public**。将 **Valid Redirect URI** 和 **Web Origins** 设置为 **\***。点 **\*Save\***。  
请注意，在 MTR 操作器实例化 **Valid Redirect URI** 和 **Web Origins** 字段后，必须将 **secure-mtr-web-console** 路由设置为 **secure-mtr-web-console** 路由。
5. 创建名为 **user** 的角色。
6. 创建一个具有任何名称的用户。
7. 设置用户的凭据，禁用 **Temporary**，并将角色"user"分配给用户。

### 安装 MTR Operator

1. 以具有 **cluster-admin** 权限的用户身份登录 OpenShift Web 控制台。
2. 点 **Operators** → **OperatorHub**。
3. 使用 **Search by keyword** 字段找到 **Migration Toolkit for Runtimes Operator**。
4. 点 **Install**。
5. 从 **Installed Namespace** 列表中选择个项目，再点 **Install**。
6. 点 **Operators** → **Installed Operators** 来验证是否安装了 Operator。

### 安装 Web 控制台应用程序

1. 以具有 **project-admin-user** 特权的用户身份登录 OpenShift Web 控制台。
2. 切换到 **Migration** 视角并点 **+Add**。
3. 在 **Add** 视图中，点 **Operator Backed**。
4. 点 **Migration Toolkit for Runtimes Operator**。
5. 点 **Create**。
6. 检查应用设置。如果 Web 控制台需要身份验证，请输入 RH SSO 设置，然后点 **Create**。
7. 在 **Topology** 视图中，点 **mtr-web-console** 应用，然后点 **Resources** 选项卡。
8. 如果需要身份验证，请将 RH SSO **Valid Redirect URI** 和 **Web Origins** 字段设置为 **secure-mtr-web-console** 路由。
9. 点 **secure-mtr-web-console** 路由，以在新的浏览器窗口中打开 Web 控制台。

## 2.2.1. 对 OpenShift 中的 Web 控制台安装进行故障排除

本节论述了如何在 OpenShift Container Platform 上安装 web 控制台。

### 2.2.1.1. 使用 OpenShift 控制台下载日志

您可以使用 OpenShift 控制台下载 pod 日志。

#### 流程

1. 打开 OpenShift 控制台，再进入到 **Applications → Pods**。
2. 点 **mtr-web-console** pod。
3. 点 **Logs**。
4. 点 **Download** 下载并保存日志。

### 2.2.1.2. 使用 CLI 下载日志

您可以使用 CLI 下载 pod 日志。

#### 流程

1. 获取 pod 名称：

```
$ oc get pods -n <project-name>
```

输出类似于以下：

NAME	READY	STATUS	RESTARTS	AGE
eap-builder-1-build	0/1	Completed	0	1d
mtr-postgresql-1-hfbdn	1/1	Running	0	1d
mtr-sso-1-build	0/1	Completed	0	1d
mtr-web-console-1-build	0/1	Completed	0	1d
mtr-web-console-1-vt7s5	1/1	Running	1	1d
sso-1-wjl2n	1/1	Running	1	1d

2. 使用 **oc logs** 来检查 pod 日志：

```
$ oc logs <pod>
```



#### 注意

您可以重定向输出来获取当前日志的副本：

```
$ oc logs <pod> > ./<pod>.log
```

### 2.2.1.3. No route to host 错误

**mtr-web-console-executor** 日志中的 **No route to host** 错误表示 **mtr-web-console-executor** pod 无法连接到 **mtr-web-console** pod:

```
13:44:03,501 SEVERE [org.jboss.windup.web.messaging.executor.ExecutorBootstrap] (main) Could not start messaging listener due to: Failed to connect to any server. Servers tried: [http-remoting://192.0.2.4:8080 (java.net.NoRouteToHostException: No route to host)]: javax.naming.CommunicationException: Failed to connect to any server. Servers tried: [http-remoting://192.0.2.4:8080 (java.net.NoRouteToHostException: No route to host)]
```

发生此错误的原因是 **mtr-web-console-executor** pod 在 **mtr-web-console** pod 之前开始运行。

在 **mtr-web-console** pod 运行后的几分钟内，检查 **mtr-web-console-executor** 日志。

#### 2.2.1.4. 解决资源不足的问题

以下条件表示资源不足：

- **mtr-web-console** pod 没有运行，OpenShift 控制台中的 **Pod Details** 屏幕的**事件**标签页中会显示以下错误：

```
0/9 nodes are available: 4 Insufficient cpu, 4 MatchNodeSelector, 9 Insufficient memory.
```

- **mtr-web-console-deploy**, **mtr-web-console-executor-deploy**, 和 **mtr-web-console-postgresql-deploy** pod 超时，并在日志中显示以下错误：

```
error: update acceptor rejected mtr-web-console-executor-1: Pods for rc 'mtr/mtr-web-console-executor-1' took longer than 600 seconds to become available
```

#### 流程

1. 安装并运行 [集群容量工具](#) 以确定您可以调度的 pod 数量。
2. 通过执行以下操作之一更改集群资源的负载：
  - 增加 [限制范围](#) 或 [项目的资源配额](#)。
  - 减少项目请求的资源。Web 控制台至少需要 4 个 vCPU 和 8 GB RAM。
  - 运行较少的作业。
3. 重新部署 Web 控制台。

#### 2.2.1.5. 报告问题

MTR 使用 JIRA 作为其问题跟踪系统。如果您在执行 MTR 时遇到问题，请提交 [JIRA 问题](#)。

## 第 3 章 使用 WEB 控制台分析应用程序

您可以在 Web 控制台中创建项目来分析应用程序。

每个项目对应用程序进行分组以特定分析，您可以使用自定义规则和标签进行配置。

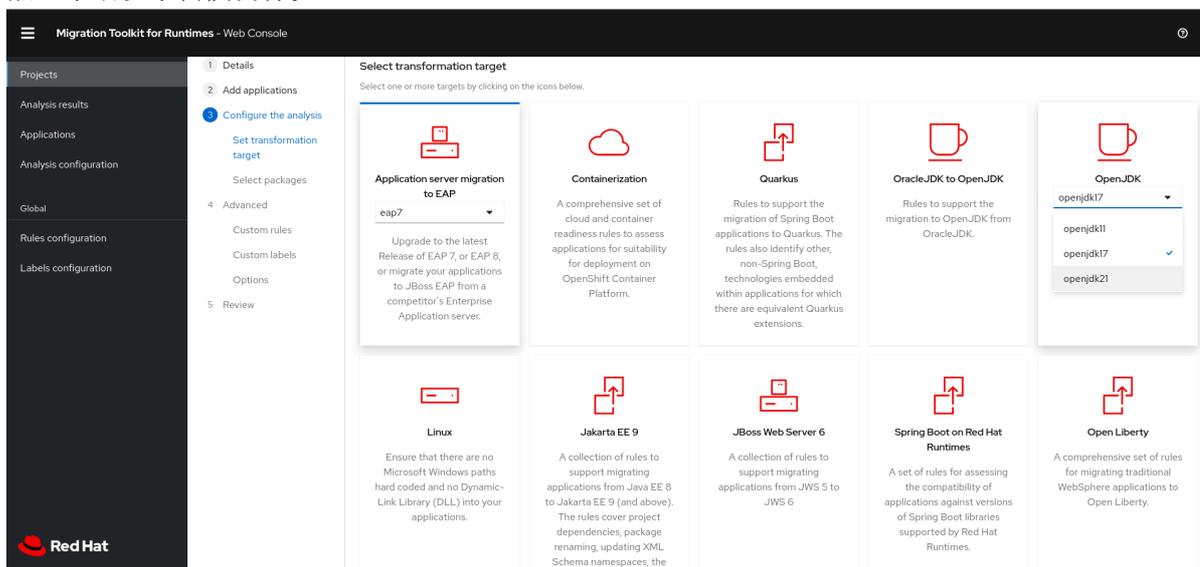
分析过程会生成报告，用于描述应用程序对迁移或现代化的准备情况。

### 3.1. 创建一个项目

您可以使用 **Create project** 向导在 web 控制台中创建项目。

#### 流程

1. 在 Web 控制台中，点 **Projects**。
2. 点 **Create project**。
3. 输入项目的唯一名称，是可选的描述，然后点 **Next**。
4. 要上传应用程序，点 **Upload** 选项卡，点 **Browse**，选择您要上传的应用程序文件，然后点 **Close**。  
上传应用程序直接将其存储在 MTR 服务器上。
5. 要注册服务器路径，点 **Server path** 选项卡，然后在字段中输入应用程序的 **服务器端路径**。  
注册应用程序的服务器路径可确保 MTR 始终使用最新版本。
6. 点击 **Next**。
7. 点一个或多个转换目标。



8. 点击 **Next**。
9. 选择软件包并点 **>** 在分析中包含它们。
10. 点 **Next**。
11. 如果要添加自定义规则，点 **Add rule**。  
[如需更多信息](#)，请参阅规则开发指南。

- 要上传规则文件，点 **Upload** 选项卡，点 **Browse**，选择一个或多个文件，然后点 **Close**。ruleset 文件必须具有 **.windup.xml** 扩展。上传的文件存储在 MTR 服务器上。
- 要注册规则集文件的服务器路径，请点 **Server path** 选项卡，输入 **Rules** 路径，然后点 **Save**。  
注册服务器路径可确保 MTR 服务器始终使用规则集文件的最新版本。

12. 点击 **Next**。

13. 如果要添加自定义标签，点 **Add label**。

- 要上传标签集文件，点 **Upload** 选项卡，点 **Browse**，选择一个或多个文件，然后点 **Close**。labelset 文件必须具有 **.windup.label.xml** 扩展。上传的文件存储在 MTR 服务器上。
- 要注册服务器路径，点 **Server path** 选项卡，在字段中输入标签文件的 **Labels** 路径，然后点 **Save**。  
注册服务器路径可确保 MTR 服务器始终使用标记设置文件的最新版本。

14. 点击 **Next**。

15. 查看以下 **Advanced** 选项 并进行必要的更改：

- **目标**
- **源**
- **排除标签**：带有这些标签的规则不会被处理。
- **其他类路径**：输入以空格分隔的额外 **.jar** 文件或目录列表，以便它们可用于编译或其他分析。
- **应用程序名称**
- **Mavenize 组 ID**
- **忽略路径**：输入要从分析中排除的文件的路径。
- **导出 CSV**：将报告数据导出为 CSV 文件。
- **Export Summary**: 在输出目录中生成 **analysisSummary.json** 导出文件。文件包含了对每个被分析的应用程序的分析摘要信息，包括按类别划分的事件和故事点数，以及与分析的应用程序关联的所有技术标签。
- **Export reports zip**: 在输出文件夹中创建一个 report **.zip** 文件。该文件包含分析输出，通常是报告。如果请求，它也可以包含 CSV 导出文件。
- **禁用 Tattletale**：禁用为每个应用程序生成 Tattletale 报告。
- **Class Not Found 分析**：启用对类路径上不可用的 Java 文件的分析。



### 注意

如果某些类无法进行分析，则不应使用这个选项。

- **兼容文件报告**：生成兼容的文件报告可能需要很长时间才能进行大型应用程序。
- **展开式应用**：输入目录包含应用程序的未打包源文件。

- **保留工作目录**：保留临时文件，例如：图形数据库或提取的存档文件（用于调试）。
- **跳过报告**：不会生成 HTML 报告。如果启用了 **导出 CSV**，则必须启用。
- **允许网络访问**：根据其 schema 验证分析的应用程序中的任何 XML 文件。



#### 注意

这个选项可能会降低性能。

- **Mavenize**：基于应用的结构和内容创建 Maven 项目目录结构。
- **源模式**：指示应用程序文件是原始源文件，而不是编译的二进制文件。sourceMode 参数已弃用。现在，不再需要指定它。MTR 可以直观地处理向其呈现的任何输入。另外，可以使用同一分析执行中的二进制输入来分析项目源文件夹。
- **分析已知库**：分析嵌入在应用程序中的已知软件工件。默认情况下，MTR 仅分析应用程序代码。



#### 注意

这个选项可能会导致较长的执行时间和报告大量迁移问题。

- **事务分析**：[技术预览] 生成一个交易报告，该报告显示调用堆栈，该堆栈对关系数据库表执行操作。Enable Transaction Analysis 功能支持 Spring Data JPA，以及用于 SQL 语句执行的传统的 **preparedStatement()** 方法。它不支持 ORM 框架，如 Hibernate。



#### 注意

事务分析只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

- **Skip source code reports**: 添加此选项会在生成应用程序分析报告时跳过生成 **源代码报告**。当需要关注应用程序分析中的源代码信息时，请使用此高级选项。

16. 点击 **Next**。

17. 检查您的项目，然后点 **Save** 或 **Save and run**。  
该项目显示在 **Projects** 屏幕中。

## 3.2. 运行保存分析

您可以运行已保存的分析。

### 流程

1. 在 Web 控制台中，点 **Analysis 结果**。
2. 选择一个项目。
3. 点 **Run analysis**。

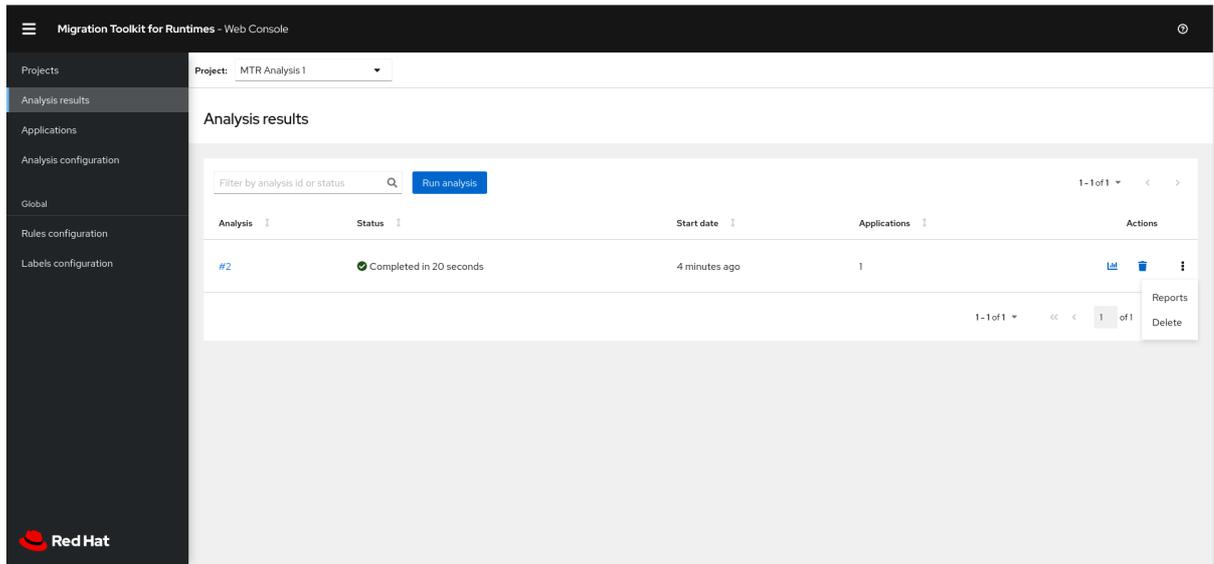
进度条显示您分析的进度。

### 3.3. 查看分析结果

所有分析结果都会在 **分析结果** 屏幕上按项目分组和列出。

#### 流程

1. 在 Web 控制台中，点 **Analysis 结果**。
2. 从列表中选择个项目。

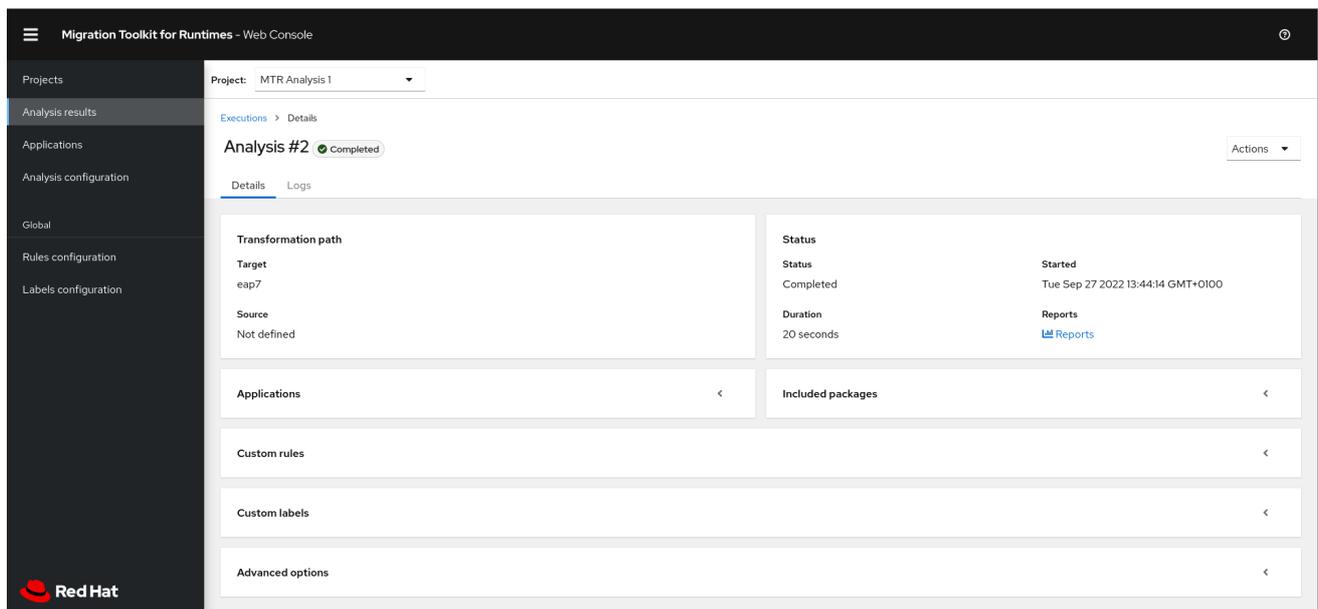


分析报告可以通过报告条图表图标或屏幕右侧的 **Reports** 操作来访问。

3. 点您要检查的分析数量，以查看分析配置设置和分析执行统计信息的详情。  
结果会显示在结果屏幕中，其中包含两个标签页：**Details** 和 **Logs**。

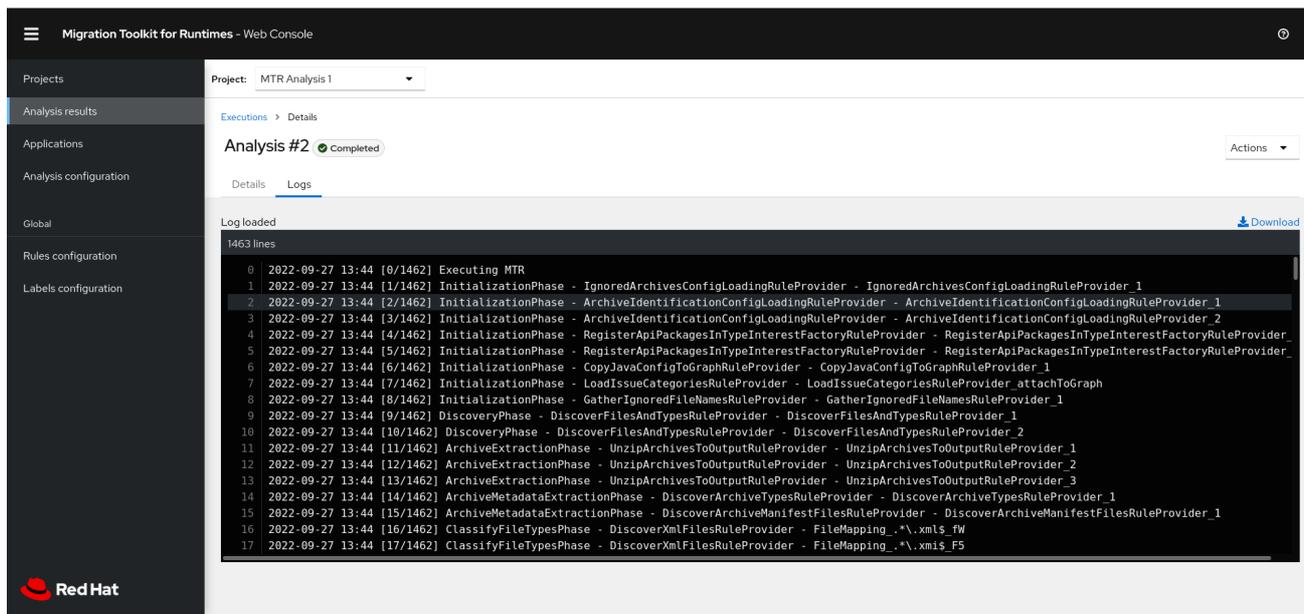
**Details** 选项卡显示分析的重要详情，如状态、开始日期、持续时间和配置设置。

图 3.1. 分析详情屏幕



**Logs** 选项卡显示分析期间生成的日志。

图 3.2. 分析日志屏幕



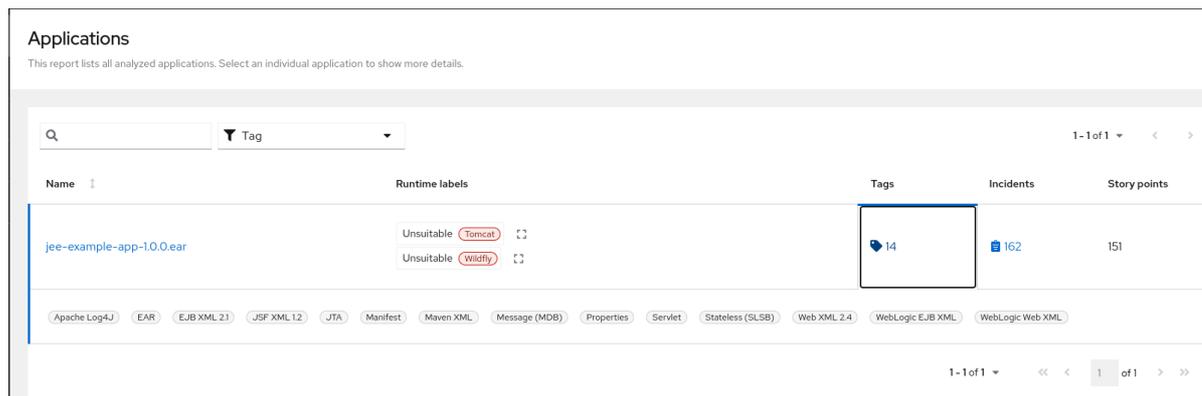
### 3.4. 检查报告

MTR web 控制台提供了一组详细的报告，可帮助您决定是否需要对您的应用程序进行任何更改。您可以从分析结果屏幕中访问这些报告。

该报告在 *CLI 指南* 中的 [检查报告](#) 中详细介绍。

#### 流程

1. 在 Web 控制台中，点 **Analysis 结果**。
2. 点您要调查的分析  旁的 **Reports** 图标。此时会显示 **所有报告的应用程序** 屏幕。

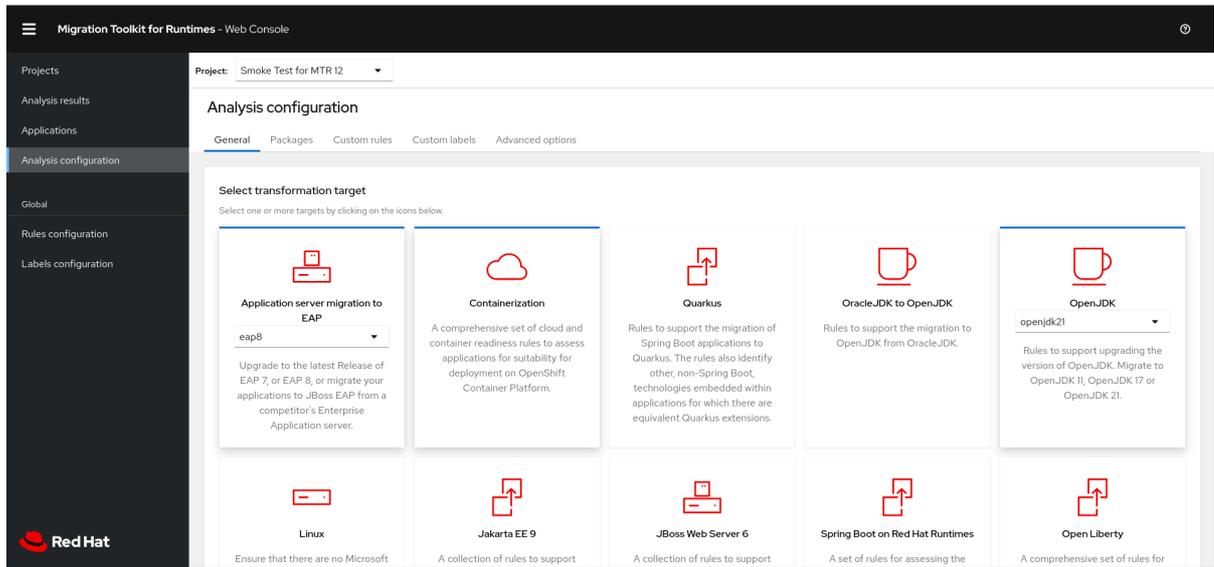


### 3.5. 更新分析配置

您可以更新分析配置，例如使用不同的转换目标、高级选项或自定义规则。然后，您可以在项目中运行更新的分析。

#### 流程

1. 在 Web 控制台中，点 **Analysis 配置**。



2. 选择 项目。
3. 点适当的选项卡并进行更改。
4. 点 **Save** 或 **Save and run**。  
该项目显示在 **Projects** 屏幕中。

### 3.6. 添加全局自定义规则

MTR 包括一组预配置的全局规则，适用于所有项目。

您可以自行定义自定义全局规则。

有关编写自定义 MTR 规则的详情，请参考 MTR [规则开发指南](#)。

#### 流程

1. 在 Web 控制台中，点 **Rules 配置**。
2. 点 **Add rules**。
3. 要上传规则文件，点 **Upload** 选项卡，点 **Browse**，选择一个或多个文件，然后点 **Close**。  
ruleset 文件必须具有 **.windup.xml** 扩展。上传的文件存储在 MTR 服务器上。
4. 要注册规则集文件的服务器路径，请点 **Server path** 选项卡，输入 **Rules** 路径，然后点 **Save**。  
注册服务器路径可确保 MTR 服务器始终使用规则集文件的最新版本。

**Custom rules** 列表显示规则。

### 3.7. 添加全局自定义标签

MTR 包含一组预配置的全局标签，适用于所有项目。

您可以定义自己的自定义全局标签。

#### 流程

1. 在 Web 控制台中，点 **Labels 配置**。

2. 点 **Add label**。
3. 要上传标签集文件，点 **Upload** 选项卡，点 **Browse**，选择一个或多个文件，然后点 **Close**。  
labelset 文件必须具有 **.windup.label.xml** 扩展。上传的文件存储在 MTR 服务器上。
4. 要注册标签集文件的服务器路径，请点 **Server path** 选项卡，输入 **Labels** 路径，然后点 **Save**。  
注册服务器路径可确保 MTR 服务器始终使用标记设置文件的最新版本。

**Custom labels** 列表显示标签。

## 第 4 章 在 LINUX、WINDOWS 或 MACOS 上为 WEB 控制台配置身份验证

您可以将 Web 控制台配置为需要身份验证才能访问。要启用身份验证，您必须安装 Red Hat Single Sign-On (SSO)。

### 启用身份验证

#### 流程

1. 通过输入以下内容来调整 Red Hat SSO 服务器打开的端口号，以避免与 Web 控制台使用的端口冲突：
  - 对于 Linux 和 macOS:

```
┌ $ ./standalone.sh -Djboss.socket.binding.port-offset=<offset_value>
```
  - 对于 Windows:

```
┌ > ...\.bin\standalone.bat -Djboss.socket.binding.port-offset=<offset_value>
```
2. 从 <http://localhost:8180> 打开 Red Hat SSO 管理控制台：
  - 用户名：**admin**
  - Password: **admin**
3. 添加名为 **mtr** 的域。
4. 在域中，创建一个名为 **mtr-web** 的客户端。
5. 检查 **Access Type** 是否已设置为 **public**。
6. 将 **Valid Redirect URI** 设置为 **http://localhost:8080/windup-ui/\***。
7. 将 **Web Origins** 设置为 **\***，然后点 **Save**。
8. 创建名为 **user** 的角色。
9. 创建一个具有任何名称的用户。
10. 设置用户的凭据，禁用 **Temporary**，并将角色 "user" 分配给用户。
11. 通过执行以下操作将 Web 控制台切换到 **Authentication** 所需的模式：
  - a. 导出以下 ENV 变量：
    - 对于 Linux 和 macOS:

```
┌ export SSO_AUTH_SERVER_URL=http://localhost:8180/auth
┌ export SSO_REALM=windup
┌ export SSO_SSL_REQUIRED=EXTERNAL
┌ export SSO_CLIENT_ID=windup-web
```
    - 对于 Windows:

```
set SSO_AUTH_SERVER_URL=http://localhost:8180/auth
set SSO_REALM=windup
set SSO_SSL_REQUIRED=EXTERNAL
set SSO_CLIENT_ID=windup-web
```



### 注意

在 CMD 中由 **set** 命令设置的环境变量是本地的，仅适用于当前 CMD 会话。使用 Windows Control Panel 永久设置环境变量。

b. 运行以下脚本：

- 对于 Linux 和 macOS:

```
$ <MTR_HOME>/switch_to_authentication_required.sh
```

- 对于 Windows:

```
C:\<MTR_HOME>\switch_to_authentication_required.bat
```

12. 输入以下内容启动 Web 控制台：

- 对于 Linux 和 macOS:

```
$ <MTR_HOME>/run_windup.sh
```

- 对于 Windows:

```
C:\<MTR_HOME>\run_windup.bat
```

13. 打开位于 **http://localhost:8080/windup-ui** 的浏览器。

## 禁用身份验证

### 流程

1. 运行以下脚本：

- 对于 Linux 和 macOS:

```
$ <MTR_HOME>/switch_to_automatic_authentication.sh
```

- 对于 Windows:

```
C:\<MTR_HOME>\switch_to_automatic_authentication.bat
```

更新于 2024-04-18

