redhat.

详细信息

直接云迁移开发人员指南

目录

红帽官方微博　红帽官方微信

cn.redhat.com

简介

随着云模型的日益普及，许多企业都在积极探求转向云原生开发的途径。您可以完全借助云模型和服务，通过微服务、自主开发团队、敏捷和持续部署以及容器化和经过编排的云部署来开发新的应用。但遗憾的是，鉴于所需的时间和成本，完全重写所有传统应用基本上是不可行的。

为了保持敏捷性和竞争力，企业必须迁移和现代化现有 Java™ 应用，并采用云原生开发方案。对于大多数公司而言，这个过程涉及到尽可能重用现有的功能和数据，将现有的工作负载迁移到现代部署平台，最后再应用新的流程、产品和技术。

“直接迁移”现代化模型是云原生开发中富有吸引力的第一步，其中涉及：

- 实现现有单体式工作负载的容器化。
- 在平台即服务 (PaaS) 上部署工作负载。
- 保留传统平台上的外部集成和数据。

完成上述步骤后，开发人员就可以开始“扼杀单体式应用”，并逐步用微服务取代选择的应用内功能。

本指南将介绍使用红帽® 软件进行直接迁移的过程，详见下面的图 1，其中涉及：

1. 使用**红帽应用迁移工具包**分析现有的单体式应用，并迁移至开放标准接口。
2. 更新代码和配置，以便在**红帽 JBoss® 企业应用平台上运行**。
3. 在**红帽 OpenShift® 容器平台**上将单体式应用部署为容器，这个平台提供了自动扩展、集成集群和故障转移等功能，可以增强应用性能。

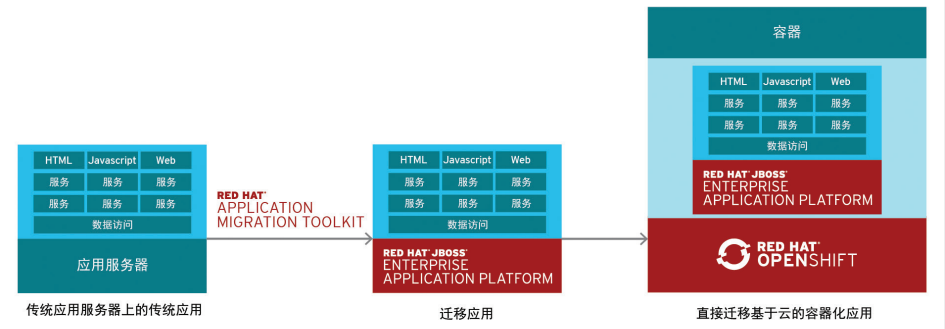

图 1. 使用红帽软件的直接迁移流程

将应用部署到红帽 OpenShift 容器平台后，可以使用其他工具将特定功能转换为微服务。该过程可用于将功能分解为更小的部分，以实现更大程度的并行化和自主管理，并借助以下工具将注意力放在缩短价值实现时间上：

- Thorntail/MicroProfile
- Spring Boot
- Node.js
- Eclipse Vert.x

使用红帽应用迁移工具包简化迁移过程

红帽应用迁移工具包是一个基于规则的可扩展、可自定义工具，它可以帮助您简化 Java 应用的迁移过程。该工具包可用于：

- 规划和工作估算。
- 确定迁移问题并提供解决方案建议。
- 评估详细报告。
- 使用内置规则和迁移路径。
- 实现规则扩展和自定义。
- 分析源代码或应用归档。

在迁移应用时，红帽应用迁移工具包会查找公共资源并重点突出技术和已知难点。该工具包高度介绍了应用所运用的技术并提供一份详细报告，供企业来评估、记录 Java EE 应用，并将其迁移到红帽 JBoss 企业应用平台。

注：红帽应用迁移工具包通常只是大规模应用迁移和现代化计划的一部分。这些计划通常由明确定义且可重复的若干阶段（数周或数月）构成，并且可能涉及来自特定组织的许多人。

要了解有关红帽理念和经过验证的方法的更多信息，请查看红帽应用迁移工具包文档和开发人员主页。

分析 JAVA EE 应用

红帽应用迁移工具包会检查应用工件，包括项目源目录和应用存档。随后，它会生成一份 HTML 报告，上面重点突出需要改变的领域。该工具包可用于从先前版本的红帽 JBoss 企业应用平台 (EAP) 或其他应用服务器平台（如 Oracle WebLogic Server 或 IBM® WebSphere 应用服务器）迁移 Java 应用。

您可以通过几种不同的方式来安装和使用红帽应用迁移工具包：

- **Web 控制台。**红帽应用迁移工具包的 Web 控制台允许开发人员团队对大量应用的迁移和现代化工作进行评估并确定其优先级。它可以将应用分组到项目中进行分析，并提供大量报告以重点突出分析结果。
- **命令行界面 (CLI)。**CLI 是一个命令行工具，它允许开发人员对应用的迁移和现代化工作进行评估并确定其优先级。之后，它会提供大量报告以重点突出分析结果。CLI 的功能简单明了，是分析单个应用的理想之选。
- **Eclipse 插件。**红帽应用迁移工具包的 Eclipse 插件可直接在 Eclipse 和红帽 JBoss 开发人员工作室中提供帮助，协助开发人员进行有关迁移或现代化工作的更改。它将使用红帽应用迁移工具包对您的项目进行分析、在源代码中标记迁移问题、提供有关修复问题的指导，并在可能的情况下提供自动代码替换。

所有这些访问方法都可以从**红帽应用迁移工具包**页面下载。有关安装说明，请参阅每种访问方法的文档。对于多用户场景，Web 控制台是一个不错的选择。为简单起见，本文档以 CLI 为例进行说明。以下步骤总结了红帽应用迁移工具包的相关操作。

1. 验证红帽应用迁移工具包 CLI

安装好红帽应用迁移工具包 CLI 后，运行以下命令以验证该工具是否安装正确：

```
$ ${HOME}/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli --version
```

此时应显示类似于如下所示的内容：

```
Using RHAMT at /root/rhamt-cli-4.0.0.Beta4

> Red Hat Application Migration Toolkit (RHAMT) CLI, version 4.0.0.Beta4.
```

2. 针对您的项目运行红帽应用迁移工具包 CLI

红帽应用迁移工具包 CLI 中有许多选项可用于控制它的运行方式。下面，我们将针对一个名为 monolith 的项目运行 CLI，并生成一份同名报告。在研究期间，请用实际项目的名称替换下面的 `~/projects/monolith`。

```
$ ~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli  \
--sourcemode  \
--input ~/projects/monolith  \
--output ~/rhamt-reports/monolith  \
--overwrite  \
--source weblogic  \
--target eap:7  \
--packages com.redhat weblogic
```

请注意 `--source` 和 `--target` 选项的使用。这些选项允许您明确指定红帽应用迁移工具包所支持的特定迁移路径。迁移路径包括：

- Oracle WebLogic Server
- IBM WebSphere Application Server
- 红帽 JBoss EAP 5/6/7

等待工具包运行完成，然后再继续。完成后，应显示类似于如下所示的内容：

```
Report created: ~/rhamt-reports/monolith/index.html
```

3. 查看结果

要查看报告，请将本地浏览器指向以下文件：

```
~/rhamt-reports/monolith/index.html
```

此时应显示图 2 所示的报告登录页面。主报告登录页面上会列出已处理的应用以及遇到的问题。页面上的每一行中都包含故事点的高级概述、事件数量以及在应用中遇到的技术。

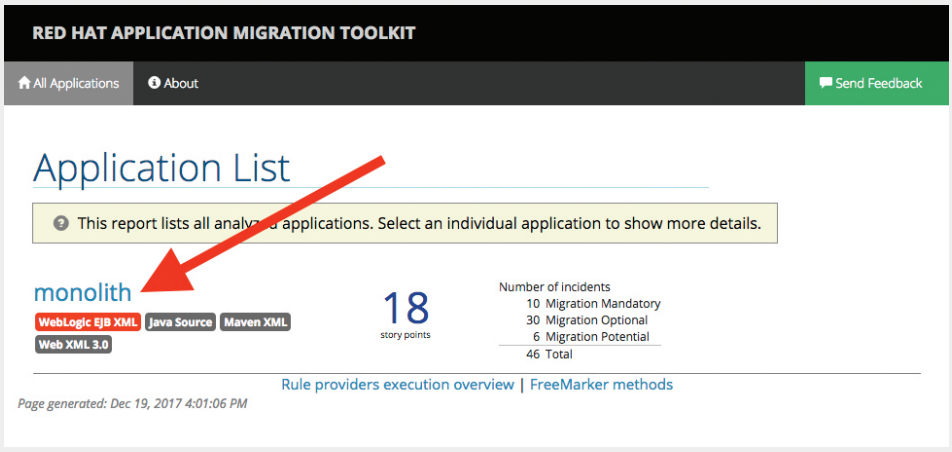

图 2. 红帽应用迁移工具包的报告登录页面

在本例中，单击 `monolith` 链接可访问项目的仪表板，如图 3 所示。

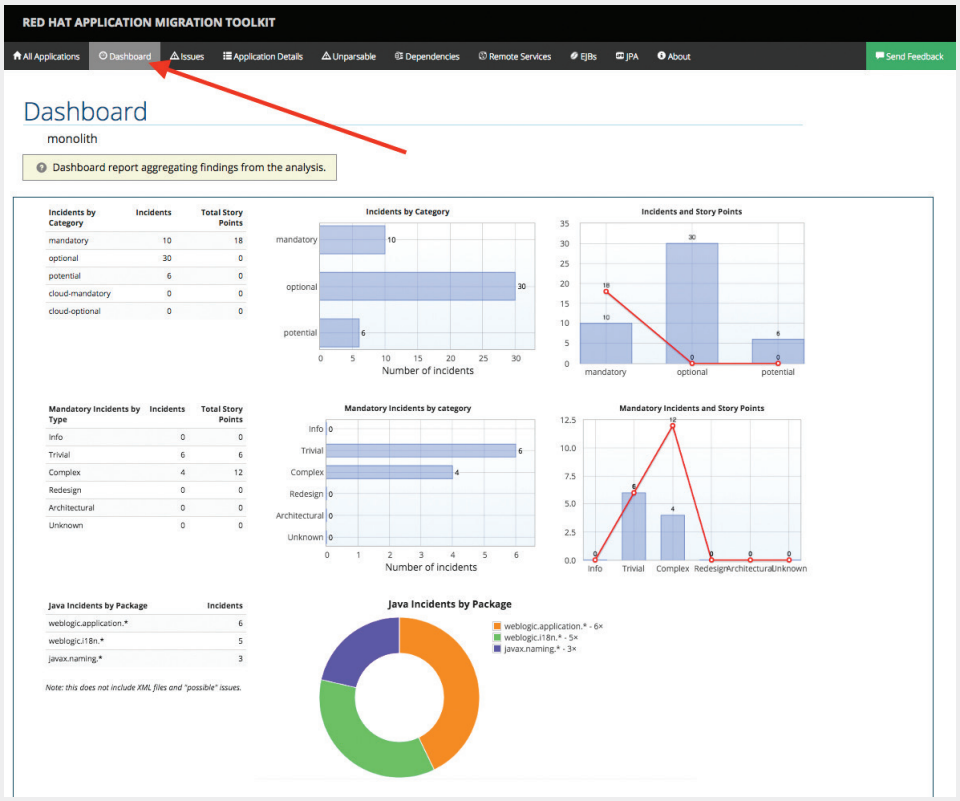

图 3. 红帽应用迁移工具包仪表板

理解报告

仪表板提供了有关整个应用迁移工作的概述。它对以下内容进行了总结：

- 按类别划分的事件和故事点。
- 按所建议的更改的工作量划分的事件和故事点。
- 按软件包划分的事件。

注：故事点是敏捷软件开发中常用的抽象指标，用于估计实现功能或更改所需的相对工作量。红帽应用迁移工具包使用故事点来表示迁移特定应用构建（以及整个应用）所需的工作量。根据所迁移应用的规模和复杂程度，工作量会有很大差异。

通过靠近顶端的菜单可以访问其他几个子页面。

- **全部应用**提供了所有扫描到的应用的列表。
- **仪表板**提供了有关特定应用的概述。
- **问题**总结了所有需要注意的问题。

- **应用详情**详细概述了应用中找到的迁移期间可能需要注意的所有资源。
- **无法解析**显示了红帽应用迁移工具包无法以预期格式进行解析的所有文件。例如，带有 .xml 或 .wsdl 后缀的文件被视为 XML 文件。如果 XML 解析器失败，就会在本部分以及列出相应文件的地方报告此问题。
- **依赖项**显示了在应用中找到的所有 Java 打包依赖项。
- **远程服务**显示了在应用中找到的所有远程服务引用。
- **EJB** 包含了在应用中找到的企业 Java Bean 列表。
- **JBPM** 列出了在分析期间发现的所有与 Java 业务流程管理相关的资源。
- **JPA** 包含了有关在应用中找到的所有 Java Persistence 应用编程接口 (API) 相关资源的详细信息。
- **关于**描述了红帽应用迁移工具包的当前版本，并提供了一些有用的链接以获取进一步帮助。

注：根据项目中检测到的实际问题，上述部分内容可能不会显示。

有了红帽应用迁移工具包报告后，现在即可开始迁移应用。

解决问题

图 4 中的“问题报告”选项卡中提供了所研究应用的所有已知问题的详细列表。它列出了一些有用的链接，可帮助用户更深入地了解问题并提供有关迁移和现代化的指导。

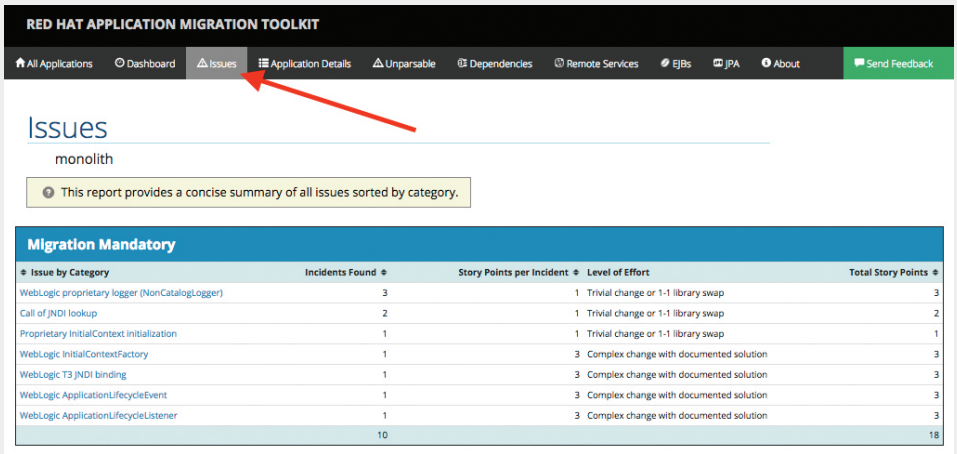

图 4. 红帽应用迁移工具包的问题报告

许多应用中都有一些特定于平台的代码，要想使用标准 Java EE 接口，就需要对这些代码进行更新，其中包括：

- **启动代码 -** 用于在启动或停止应用时执行功能或安排作业。
- **日志记录代码 -** 使用特定于平台的日志记录功能。
- **消息传递代码 -** 采用了一些过时的或特定于平台的机制。

无论是哪种情况，问题报告都会提出可能的解决方案。在实际用例中，其中一些问题可能需要作为迁移的一部分予以额外考虑。但是，实现这些更改会使代码更具可移植性。

完成必要的更改后，可使用 Maven 来构建和打包应用，以确保更改后的代码仍然编译：

```
$ mvn clean package
```

如果代码构建成功（显示 `BUILD SUCCESS`），您可以继续处理问题报告中的下一个问题。

重新测试构建

解决了所有问题之后，再次运行红帽应用迁移工具包，以验证迁移是否成功。

1. 针对项目运行红帽应用迁移工具包 CLI

运行以下命令以清除旧的构建工件，重新执行红帽应用迁移工具包 CLI，然后分析新项目。同样，用您的实际项目来替换 `~/projects/monolith`。

```
$ mvn clean &&  \
~/rhamt-cli-4.0.0.Beta4/bin/rhamt-cli  \
--sourceMode  \
--input ~/projects/monolith  \
--output ~/rhamt-reports/monolith  \
--overwrite  \
--source weblogic  \
--target eap:7  \
--packages com.redhat weblogic
```

完成此过程后，CLI 应显示：

```
Report created: /root/rhamt-reports/monolith/index.html
```

2. 查看结果

重新加载本地报告网页：

```
~/rhamt-reports/monolith/index.html
```

成功迁移后，相应的应用应报告零 (0) 个故事点，表示应用已成功实现现代化，现在可以移至红帽 JBoss EAP 了。

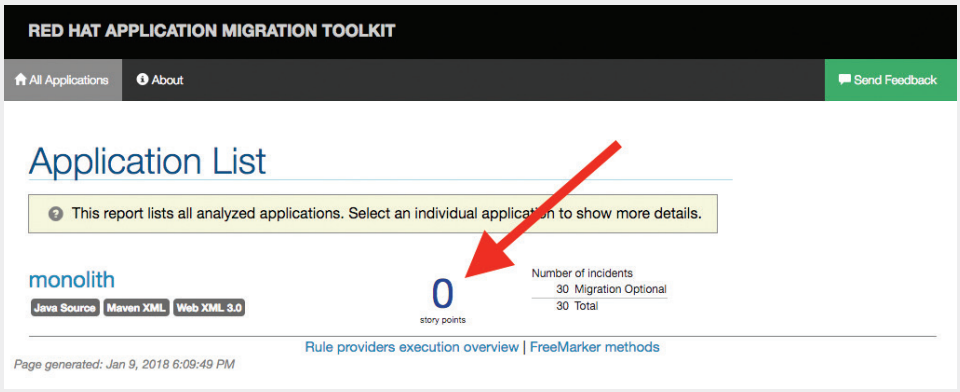

图 5. “0”个故事点意味着迁移成功

在红帽 JBOSS 企业应用平台中运行项目

既然应用已更新为使用更标准的 Java EE 接口，您便可以对其进行部署、测试，并开始探索红帽 JBoss EAP 所提供的一些功能。

安装红帽 JBOSS 企业应用平台

下载最新版的红帽 JBoss EAP。然后，在终端窗口中运行以下命令，从而在本地安装红帽 JBoss EAP。调整文件名以便与下载的版本相匹配。

```
$ unzip -d $HOME $HOME/jboss-eap-7.2.0.zip
```

接着，设置 `JBOSS_HOME` 环境变量：

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0
```

安装红帽 JBoss EAP 就是这么简单。

WILDFLY MAVEN 插件

红帽 JBoss EAP 提供了一个 wildfly-maven-plugin 工具，可直接从 Apache Maven 停止、启动、部署和配置平台。配置完 maven 插件工具后，可以使用它：

- 在完整的 Java EE 或 Java EE Web 配置文件之间进行选择。
- 配置数据库资源。
- 配置服务，如 Java 消息服务主题。

所有这些功能都是在 `pom.xml` 文件中配置的。

配置应用

使用红帽应用迁移工具包后，应用基本上基于标准配置。您可以通过启动红帽 JBoss EAP 对其进行配置，包括启动应用、添加资源及关闭应用，如下所示：

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; \ mvn wildfly:start
wildfly:add-resource wildfly:shutdown
```

当显示 `BUILD SUCCESS` 消息时，标志着本步骤结束。

注：之所以要使用 wildfly:start 和 wildfly:shutdown，原因就是 add-resource 命令需要一个正在运行的应用服务器。添加上述资源后，我们不再需要运行此命令。

部署应用

您现在可以部署应用了：

```
$ export JBOSS_HOME=$HOME/jboss-eap-7.2.0 ; mvn wildfly:run
```

服务器启动后，应显示：

```
Deployed “ROOT.war” (runtime-name: “ROOT.war”)
```

测试应用

通过向浏览器中加载以下 URL 来访问应用：

```
http://localhost:8080
```

如有警告，将会显示在控制台输出中。

关闭应用

在继续之前，通过在终端窗口中键入 CTRL-C 来终止应用。

将单体式应用部署至红帽 OPENSHIFT 平台

在红帽 JBoss EAP 中启动并运行应用后，即可将更新后的单体式应用迁移至红帽 OpenShift 容器平台。以下步骤假定可以访问红帽 OpenShift 容器平台集群。如果您无权访问群集，可以在本地安装红帽 OpenShift 容器平台。

添加红帽 OPENSHIFT 配置文件

打开 pom.xml 文件并添加红帽 OpenShift 配置文件。

```
<profile>
  <id>openshift</id>
  <build>
      <plugins>
          <plugin>
              <artifactId>maven-war-plugin</artifactId>
              <version>2.6</version>
              <configuration>
                  <webResources>
                      <resource>
                          <directory>${basedir}/src/main/webapp/WEB-INF</directory>
                          <filtering>true</filtering>
                          <targetPath>WEB-INF</targetPath>
                      </resource>
                  </webResources>
                  <outputDirectory>deployments</outputDirectory>
                  <warName>ROOT</warName>
              </configuration>
          </plugin>
      </plugins>
  </build>
</profile>
```

图 6. 红帽 OpenShift 配置文件（pom.xml 文件）

创建红帽 OPENSHIFT 项目

首先，打开红帽 OpenShift 容器平台 Web 控制台。

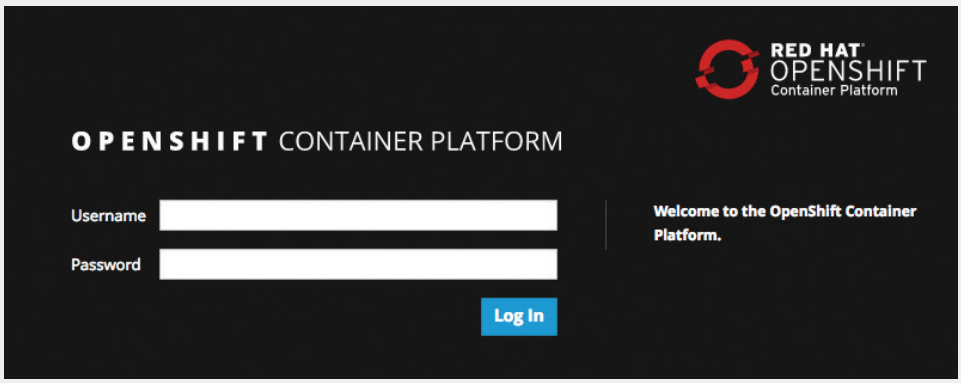

图 7. 红帽 OpenShift 容器平台 Web 控制台

使用您的红帽 OpenShift 凭据登录:

- 用户名: XXXXXX
- 密码: XXXXXX

此时将显示红帽 OpenShift 容器平台登录页面:

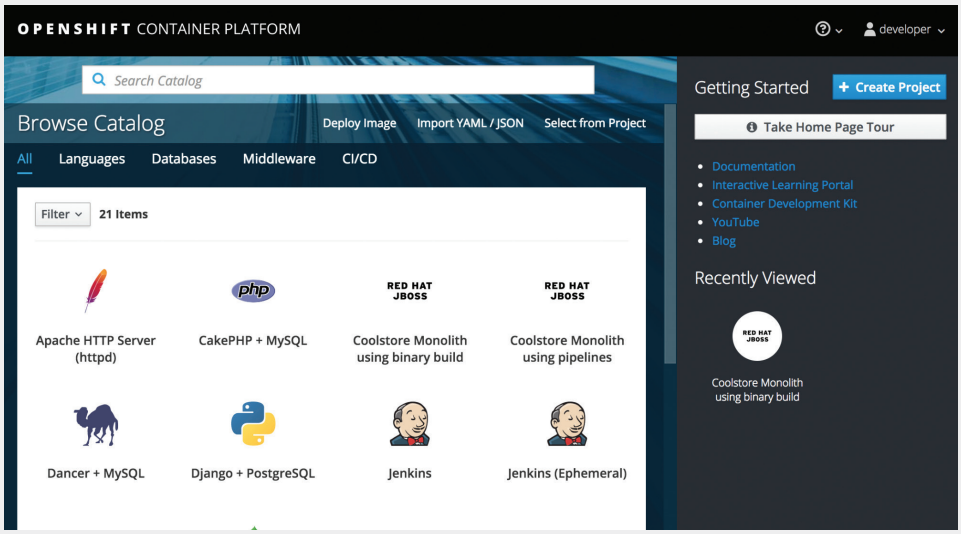

图 8. 红帽 OpenShift 容器平台登录页面

单击“创建项目”，填写字段，然后单击“创建”。在本例中，应用的名称为“coolstore”。

- 名称: `coolstore-dev`
- 显示名称: `Coolstore Monolith - Dev`
- 描述: 请将此字段留空

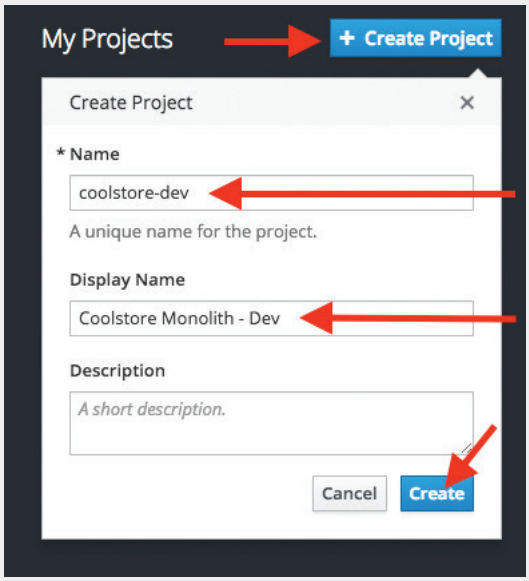

图 9. 红帽 OpenShift 容器平台项目定义的示例

单击新创建的项目的名称：

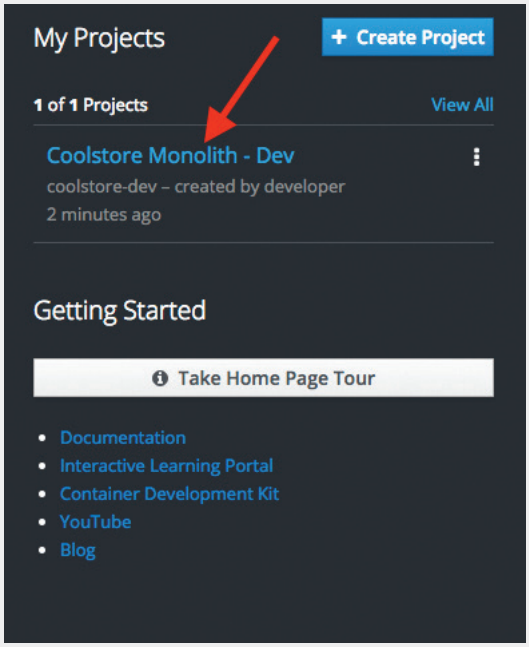

图 10. 红帽 OpenShift 容器平台项目概述

这将带您进入项目概述。目前还没有任何内容，但即将就会发生改变。

部署单体式应用

使用 CLI 部署单体式应用的组件。要使用 CLI 来部署单体式应用模板，请执行以下命令。

首先，切换到之前创建的开发人员项目：

```
$ oc project coolstore-dev
```

最后，部署模板：

```
$ oc new-app coolstore-monolith-binary-build
```

此步骤将部署应用及红帽 JBoss EAP 所需的数据库，但不会开始应用的构建。

接下来，从以下地址打开“单体式应用概述”页面：

`https://$OPENSHIFT_MASTER/console/project/coolstore-dev/`，然后验证是否已创建单体式应用模板项目。

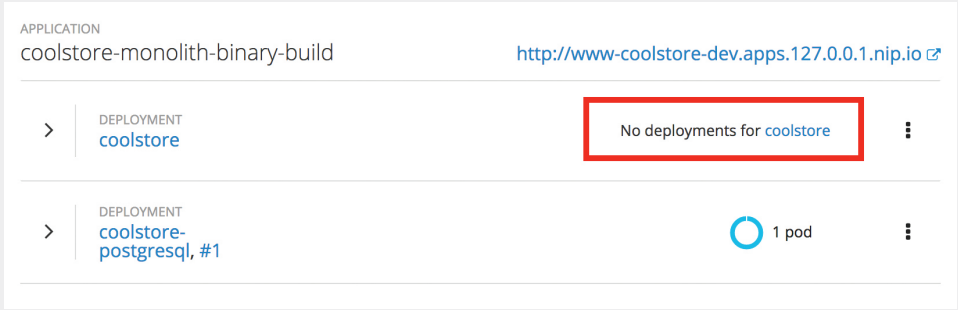

图 11. 红帽 OpenShift 容器平台单体式应用概述页面

您可以在“项目概述”中看到正在部署的组件，但请注意，目前还没有针对 coolstore 的部署。在之前步骤中构建的容器映像尚部署，但接下来会执行此操作。

使用二进制构建来部署应用

在本开发项目中，我们选择使用一个名为“二进制构建”的进程。此进程将在本地完成构建，且只上传工件（例如 `.war` 文件），而不是指向公共 Git 存储库并让源至镜像 (S2I) 构建进程下载、构建并为我们创建容器镜像。二进制部署可以显著加快构建过程。

首先，使用 `openshift` Maven 配置文件再次构建项目，这样将创建一个适用于红帽 OpenShift 容器平台的二进制文件（这还不是容器镜像，只是 `.war` 文件）。我们将使用 oc 命令行来构建项目。

构建项目：

```
$ mvn clean package -Popenshift
```

等待构建完成并生成 `BUILD SUCCESS` 消息。

最后，启动构建进程，该进程将获取 `.war` 文件，将其与红帽 JBoss EAP 进行组合，然后生成 Linux® 容器映像。借助根据模板创建的 *DeploymentConfig* 对象，会自动将此映像部署到项目中：

```
$ oc start-build coolstore --from-file=deployments/ROOT.war
```

检查红帽 OpenShift Web 控制台，应该能看到正在构建的应用。在这种情况下，说明已部署 PostgreSQL 数据库。

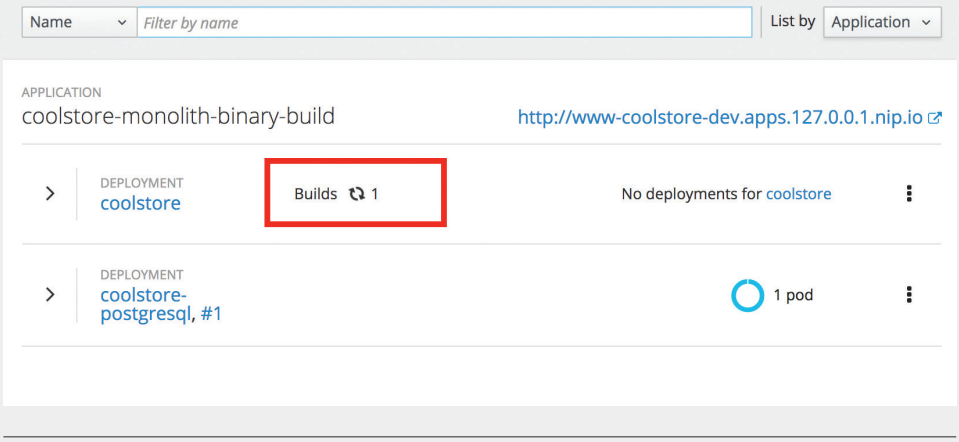

图 12. 单体式应用概述页面显示正在进行构建，但尚未部署

等待构建和部署完成：

```
$ oc rollout status -w dc/coolstore
```

此命令经常用于等待部署完成。使用此命令时，务必确保它返回成功消息。最后应显示：

```
replication controller “coolstore-1” successfully rolled out.
```

注：如果上述命令报告来自服务器的错误 (ServerTimeout)，则只需重新运行该命令，直到报告成功为止。

当该进程完成时，应显示已成功部署应用，并且数据库和单体式应用上都带有蓝圈：

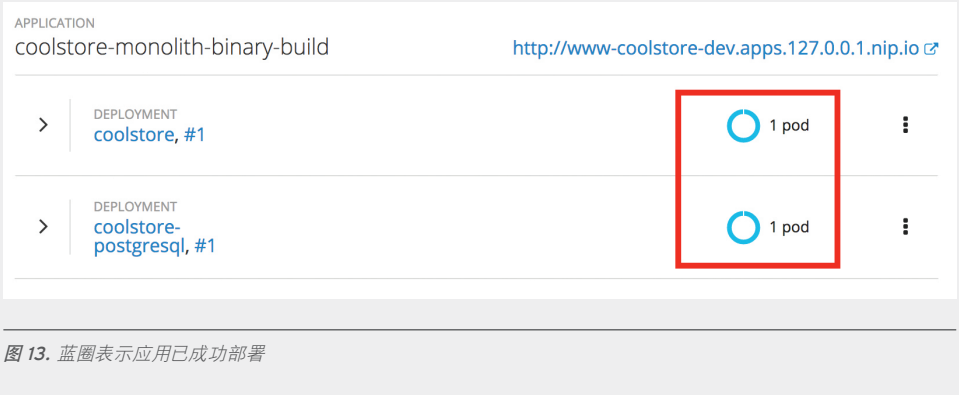

图 13. 蓝圈表示应用已成功部署

通过单击以下路由链接，对应用进行测试：

`http://www-coolstore-dev.$ROUTE_SUFFIX`，此时将在浏览器中打开 coolstore 单体式应用，但这一次是在红帽 OpenShift 容器平台上运行。

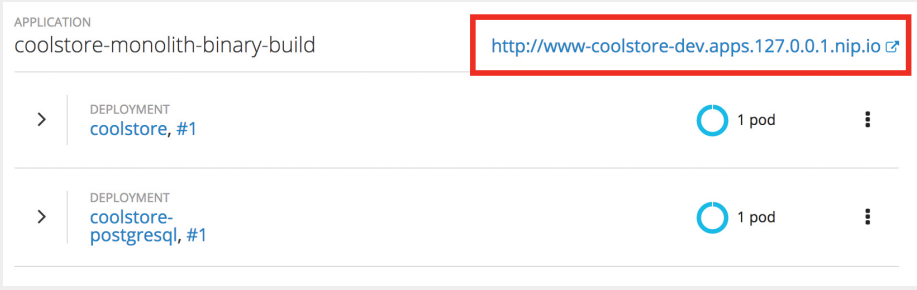

图 14. 单击链接以在浏览器中启动应用 - 该应用现在运行在红帽 OpenShift 容器平台上

综述

本过程演示了如何使用红帽 JBoss 企业应用平台和红帽 OpenShift 容器平台将现有的 Java EE 应用迁移到云端。现在，您已准备好对应用进行现代化，方法是：采取循序渐进的步骤将单体式应用分解为较小的微服务，同时使用现代技术确保应用能在分布式、容器化环境中正常运行。

销售及技术支持

800 810 2100
400 890 2100

红帽北京办公地址

北京市朝阳区东大桥路 9 号侨福芳草地大厦 A 座 8 层　邮编：100020
8610 6533 9300

红帽官方微博　红帽官方微信

cn.redhat.com
#F17126_0419