



Red Hat JBoss Enterprise Application Platform 7.4

在 Amazon Web Services 上部署 JBoss EAP

在 Amazon Web Services 上使用红帽 JBoss 企业应用平台的说明，包括配置高可用性。

Red Hat JBoss Enterprise Application Platform 7.4 在 Amazon Web Services 上部署 JBoss EAP

在 Amazon Web Services 上使用红帽 JBoss 企业应用平台的说明，包括配置高可用性。

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[®] 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.

摘要

本文档提供有关在 Amazon EC2 上部署红帽 JBoss 企业应用平台的信息。

目录

提供有关 JBOSS EAP 文档的反馈	3
使开源包含更多	4
第 1 章 关于 AMAZON EC2	5
JBoss EAP Amazon Machine Images 的类型	5
关于红帽云访问	5
红帽云访问功能	5
支持的 Amazon EC2 实例类型	5
支持的红帽 AMI	6
第 2 章 启动 JBOSS EAP 实例	7
2.1. 从 AWS 市场启动 JBOSS EAP 实例	7
2.2. 使用 AWS EC2 控制台从私有 AMI 启动 JBOSS EAP 实例	7
第 3 章 启动非集群 JBOSS EAP 实例	9
先决条件	9
启动非集群的 JBoss EAP 实例	9
第 4 章 启动非集群受管域	11
4.1. 启动要作为域控制器的实例	11
4.2. 启动一个或多个实例以作为主机控制器	12
第 5 章 启动集群 JBOSS EAP	14
5.1. 启动集群的 JBOSS EAP AMI (无需 MOD_CLUSTER 和 VPC)	14
5.2. 启动集群的 JBOSS EAP AMI (使用 MOD_CLUSTER 和 VPC)	16
第 6 章 故障排除	18
6.1. 关于 AMAZON EC2 故障排除	18
6.2. 诊断信息	18
附录 A. 参考资料	19
A.1. 用于红帽云访问计划的 AMAZON EC2 AMI	19
A.2. 配置文件和部署示例	19
A.3. 系统路径	19
A.4. 使用脚本在 AMAZON EC2 上启动 JBOSS EAP	20
A.5. 配置 JBOSS EAP 子系统以在云平台上工作	20
A.6. 集群 JBOSS EAP 实例的用户数据示例	21

提供有关 JBOSS EAP 文档的反馈

要报告错误或改进文档，请登录到 Red Hat JIRA 帐户并提交问题。如果您没有 Red Hat Jira 帐户，则会提示您创建一个帐户。

流程

1. 单击以下链接 [以创建 ticket](#)。
2. 请包含 **文档 URL**、**章节编号** 并**描述问题**。
3. 在 **Summary** 中输入问题的简短描述。
4. 在 **Description** 中提供问题或功能增强的详细描述。包括一个指向文档中问题的 URL。
5. 点 **Submit** 创建问题，并将问题路由到适当的文档团队。

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始：master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。详情请查看 [CTO Chris Wright 信息](#)。

第 1 章 关于 AMAZON EC2

Amazon Elastic Compute Cloud(Amazon EC2)是由 amazon.com 运营的服务，为客户提供了可自定义的虚拟计算环境。通过此服务，可以引导 Amazon Machine Image(AMI)以创建虚拟机或实例。用户可以在实例上安装所需的软件，并根据所使用的容量收取费用。Amazon EC2 的设计具有灵活性，允许用户快速扩展其部署的应用。

如需更多信息，请参阅 [Amazon Web Services](#) 网站。

关于 Amazon 机器镜像

Amazon Machine Image(AMI)是 EC2 虚拟机实例的模板。用户通过选择适当的 AMI 从中创建实例来创建 EC2 实例。AMI 的主要组件是一个只读文件系统，它包含已安装的操作系统以及其他软件。每个 AMI 都会为不同的用例安装不同的软件。Amazon EC2 包括 [Amazon Web Services](#) 和第三方提供的许多 AMI。用户也可以创建自己的自定义 AMI。

JBoss EAP Amazon Machine Images 的类型

通过部署公共或私有 Amazon Machine Image (AMI)，在 Amazon Elastic Compute Cloud (Amazon EC2) 上使用 JBoss EAP。



重要

红帽目前不支持单机实例或受管域中 **full-ha** 配置文件。

JBoss EAP 公共 AMI

通过 AWS 市场 <https://aws.amazon.com/marketplace> 访问 JBoss EAP 公共 AMI。公共 AMI 由 pay-as-you-go (PAYG)模型提供。通过 PAYG 模型，您只能根据您使用的计算资源数量付费。

JBoss EAP 私有 AMI

您可以使用现有订阅通过红帽云访问访问 JBoss EAP 私有 AMI。有关 Red Hat Cloud Access 的详情，请参阅[关于 Red Hat Cloud Access](#)。

关于红帽云访问

如果您有现有的红帽订阅，Red Hat Cloud Access 在红帽认证的云基础架构供应商（如 Amazon EC2 和 Microsoft Azure）上提供对 JBoss EAP 的支持。Red Hat Cloud Access 可让您在传统服务器和公共云的资源之间经济高效地迁移订阅。

您可以在 [客户门户中找到有关红帽云访问](#) 的更多信息。

红帽云访问功能

成为红帽云访问计划的成员可以访问红帽创建的受支持的私有 Amazon 机器镜像(AMI)。

Red Hat AMI 预安装并完全支持以下软件：

- Red Hat Enterprise Linux
- JBoss EAP
- 使用 Red Hat Update Infrastructure 对 RPM 进行产品更新

每个红帽 AMI 都只是一个起点，需要根据应用程序的要求进行进一步配置。

支持的 Amazon EC2 实例类型

红帽云访问支持以下 Amazon EC2 实例类型：有关每个 [实例的更多信息](#)，请参阅 [Linux 实例的 Amazon Elastic Compute Cloud User 指南](#)。

部署 JBoss EAP 的 AMI 的最低虚拟硬件要求如下：

- 虚拟 CPU : 2
- 内存 : 4 GB

但是，根据您在 JBoss EAP 上部署的应用，您可能需要额外的处理器和内存。

支持的红帽 AMI

支持的 Red Hat AMI 可以通过其名称来标识，如下例所示：

私有镜像示例

```
RHEL-7-JBEAP-7.4.0_HVM_GA-20210909-x86_64-0-Access2-GP2
```

公共镜像示例

```
RHEL-7-JBEAP-7.4.0_HVM_GA-20220804-x86_64-0-Marketplace-GP2
```

- **RHEL-x** 是 AMI 中安装的 Red Hat Enterprise Linux 的版本号。示例 **7**。
- **JBEAP-x.y.z** 是 AMI 中安装的 JBoss EAP 的版本号。示例 **7.4.0**。
- **20220804** 是 AMI 以 YYYYMMDD 的格式创建的日期。
- **x86_64** 是 AMI 的架构。这可以是 **x86_64** 或 **i386**。
- **Access2** 或 **Marketplace** 指示 AMI 是私有还是公共，如下所示：
 - 私有镜像包含 **Access2**。
 - 公共镜像包含 **Marketplace**。

第 2 章 启动 JBOSS EAP 实例

以下过程演示了从 Amazon Web Services (AWS) 市场启动公共 JBoss EAP 实例，并在 Amazon EC2 控制台中启动 JBoss EAP 实例。

2.1. 从 AWS 市场启动 JBOSS EAP 实例

公共 JBoss EAP Amazon Machine Image (AMI) 随即用即用 (PAYG) 模式提供，可在 Amazon Web Services (AWS) 市场提供。

先决条件

- 您有一个 AWS 帐户。
- 安装并配置了 Amazon Web Services CLI，并配置了您的帐户凭证。

流程

1. 使用 URL 进入 AWS 市场：<https://aws.amazon.com/marketplace>
2. 在搜索栏中搜索 "JBoss EAP"。通过 **发布程序** 过滤结果，选择 **Red Hat Limited** 和 **Red Hat**。
3. 点您要启动的镜像。



注意

如果您位于欧盟，请从发布者 "Red Hat Limited" 中选择该镜像，否则从发布者 "Red Hat" 中选择该镜像。

您将被重定向到软件订阅页面。

4. 选择订阅设置并点击 **Continue to Subscribe**。
5. 单击 **Accept Terms** 以接受条款，单击 **Continue to Configuration**。
您将被重定向到配置页面。
6. 选择配置选项并点击 **Continue to Launch**。
您会被定向到启动软件页面。
7. 检查启动配置详情，再单击 **Launch** 启动实例。

2.2. 使用 AWS EC2 控制台从私有 AMI 启动 JBOSS EAP 实例

您可以使用 AWS EC2 控制台从私有 AMI 启动 JBoss EAP 实例。

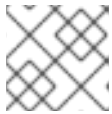
您还可以使用 AWS 命令行界面启动实例。如需有关 [AWS CLI](#) 的更多信息，请参阅 [AWS CLI](#)。

先决条件

- 您有一个红帽订阅。
- 您有一个 AWS 帐户。
- 安装并配置了 Amazon Web Services CLI，并配置了您的帐户凭证。

流程

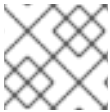
1. 打开 [Amazon EC2 控制台](#)。
2. 从 Amazon EC2 控制台中，单击 **AMI**。
3. 在位于 **Amazon Machine Images (AMI)** 面板中的 **Private images** 中搜索 **jbeap AMI**，然后选择 AMI。
例如，**RHEL-7-JBEAP-7.4.0_HVM_GA-20210909-x86_64-0-Access2-GP2**。
4. 选择实例类型。如需有关 [支持的 Amazon EC2 实例类型](#) 的更多信息，请参阅支持的 Amazon EC2 实例类型。
5. 在 **Configure Instance Details** 部分中，配置实例设置。
6. 在 **Advanced Details** 部分中 **User data** box，您可以粘贴示例脚本，以在实例启动时运行 JBoss EAP。如需了解 [有关示例脚本的信息](#)，请参阅 [使用脚本在 Amazon EC2 上启动 JBoss EAP](#)。



注意

如果需要，指定存储，标记实例并配置安全组详细信息。

7. 点 **Review and Launch**。这会直接进入 **Review Instance Launch** 页面。
8. 单击 **Launch** 以选择密钥对并启动实例。



注意

如果您没有选择密钥对，请在启动实例前指定密钥对。

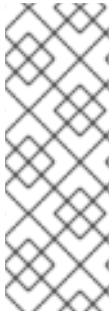
第 3 章 启动非集群 JBOSS EAP 实例

本章列出了在通过私有 AMI 或公共 Marketplace 列表创建的 Red Hat Amazon Machine Image (AMI) 上启动 JBoss EAP 非集群实例的步骤。

先决条件

- 合适的红帽 AMI。如需更多信息，请参阅支持的 [Red Hat AMI](#)。
- 预配置的安全组，允许至少在端口 22、8080 和 9990 上的传入请求。

启动非集群的 JBoss EAP 实例



注意

您可以以 **ec2-user** 用户身份通过 **ssh** 连接到 **EC2** 实例。如果需要管理特权，稍后可以更改为 **root** 用户。例如，

```
$ ssh -l ec2-user ${INSTANCE_PUBLIC_IP}
...
$ sudo su -
```

- 启动红帽 AMI 实例。
红帽 AMI 上已配置并启动了非集群 JBoss EAP 实例。
- 若要配置 JBoss EAP，您可以直接向服务传递参数。有些参数可能无法以这种方式处理。服务配置文件的位置为：
 - RHEL 6： **/etc/sysconfig/eap7-standalone**
 - RHEL 7： **/etc/opt/rh/eap7/wildfly/eap7-standalone.conf**

有关 [系统路径的详情](#)，请参阅 [系统路径](#)。



注意

- 对于复杂的配置，您可以使用 JBoss EAP **bin** 目录中的 **standalone.conf** 文件：
/opt/rh/eap7/root/usr/share/wildfly/bin/，或者您可以使用 CLI 启动 JBoss EAP 服务和配置服务器。脚本可以在 **bin** 目录中找到。然后重新加载配置。
 - 您必须定期运行 **yum -y 更新** 以应用安全修复和增强。
- 要在 RHEL 6 中启动 JBoss EAP，请运行以下命令：


```
$ service eap7-standalone start
```

要在 RHEL 7 中启动 JBoss EAP，请运行以下命令：

```
$ systemctl start eap7-standalone
```
 - 要在 RHEL 6 中停止 JBoss EAP，请运行以下命令：


```
$ service eap7-standalone stop
```

要在 RHEL 7 中停止 JBoss EAP，请运行以下命令：

```
$ systemctl stop eap7-standalone
```



注意

systemctl 命令仅与 RHEL 7 相关。



注意

如果要将 JBoss EAP 绑定到其他 IP 地址，请在 RHEL 7 上的 **/etc/opt/rh/eap7/wildfly/eap7-standalone.conf** 文件中添加以下行。内部 IP 地址由 EC2 转换为公共 IP 地址。

```
WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS
```

第 4 章 启动非集群受管域

4.1. 启动要作为域控制器的实例

本主题列出了在通过私有 AMI 或公共 Marketplace 列表创建的 Red Hat Amazon Machine Image (AMI) 上启动非集群 JBoss EAP 受管域的步骤。

前提条件

- 合适的红帽 AMI。如需更多信息，请参阅支持的 [Red Hat AMI](#)。

启动非集群的 JBoss EAP 实例



注意

您可以以 **ec2-user** 用户身份通过 **ssh** 连接到 **EC2** 实例。如果需要管理特权，稍后可以更改为 **root** 用户。例如，

```
$ ssh -l ec2-user ${INSTANCE_PUBLIC_IP}
...
$ sudo su -
```

- 启动红帽 AMI 实例。
红帽 AMI 上已配置并启动了非集群 JBoss EAP 实例。
- 若要配置 JBoss EAP，您可以直接向服务传递参数。有些参数可能无法以这种方式处理。服务配置文件的位置为：
 - RHEL 6：**/etc/sysconfig/eap7-domain**
 - RHEL 7：**/etc/opt/rh/eap7/wildfly/eap7-domain.conf**

有关 [系统路径](#) 的详情，请参阅 [系统路径](#)。

如需了解有关为 Amazon EC2 [配置 JBoss EAP 子系统](#) 的信息，请参阅 [将 JBoss EAP 子系统配置为在云平台上工作](#)。



注意

- 对于复杂的配置，您可以使用 JBoss EAP **bin** 目录中的 **domain.conf** 文件：
/opt/rh/eap7/root/usr/share/wildfly/bin/，或者您可以使用管理 CLI 启动 JBoss EAP 服务和配置服务器。脚本可以在 **bin** 目录中找到。然后重新加载配置。
 - 您必须定期运行 **yum -y 更新** 以应用安全修复和增强。
- 要在 RHEL 6 中启动 JBoss EAP，请运行以下命令：

```
$ service eap7-domain start
```

要在 RHEL 7 中启动 JBoss EAP，请运行以下命令：

```
$ systemctl start eap7-domain
```

- 要在 RHEL 6 中停止 JBoss EAP，请运行以下命令：

```
$ service eap7-domain stop
```

要在 RHEL 7 中停止 JBoss EAP，请运行以下命令：

```
$ systemctl stop eap7-domain
```



注意

systemctl 命令仅与 RHEL 7 相关。



注意

如果要将 JBoss EAP 绑定到其他 IP 地址，请在 RHEL 7 上的 `/etc/opt/rh/eap7/wildfly/eap7-domain.conf` 文件中添加以下行。内部 IP 地址由 EC2 转换为公共 IP 地址。

```
WILDFLY_BIND=$YOUR_PRIVATE_IP_ADDRESS
```

4.2. 启动一个或多个实例以作为主机控制器

本主题列出了启动一个或多个 JBoss EAP 实例的步骤，充当通过私有 AMI 或公共 Marketplace 列表创建的红帽 AMI 上的非集群主机控制器。

配置和启动非集群域控制器。请参阅 [启动实例以作为域控制器](#)。

有关 [系统路径](#) 的详情，请参阅 [系统路径](#)。

如需了解有关为 Amazon EC2 [配置 JBoss EAP 子系统的信息](#)，请参阅 [将 JBoss EAP 子系统配置为在云平台上工作](#)。

对于域控制器实例

对于在 Amazon EC2 上运行的受管域，除了静态域控制器发现外，主机控制器还可利用 Amazon Simple Storage (Amazon S3) 系统动态发现域控制器。特别是，主机控制器和域控制器可以配置访问 Amazon S3 存储桶所需的信息。

使用此配置时，域控制器启动后，它会将其联系信息写入 bucket 中的 S3 文件。每当主机控制器尝试联系域控制器时，它都会从 **S3** 文件获取域控制器的联系信息。

例如，Amazon EC2 实例的 IP 地址通常会在停止并启动时改变。在这种情况下，如果域控制器的联系信息有变化，则不需要重新配置主机控制器。主机控制器能够从 **S3** 文件获取域控制器的新联系信息。

如需有关 [用户数据示例的更多信息](#)，请参阅 [集群 JBoss EAP 实例的用户数据示例](#)。

手动域控制器发现配置使用以下属性指定：

- **access-key** : Amazon AWS 用户帐户访问密钥。
- **secret-access-key** : Amazon AWS 用户帐户 secret 访问密钥。
- **location** : 要使用的 Amazon S3 存储桶。

1. 将 **domain-ec2.xml** 文件从 `/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs` 复制到 JBoss EAP 配置目录中。

文档地址: [JBoss EAP 配置指南](#)

2. 在适当的服务配置文件中设置以下变量：

```
WILDFLY_SERVER_CONFIG=domain-ec2.xml  
WILDFLY_HOST_CONFIG=host-master.xml
```

3. 将 S3 域控制器发现配置添加到 **domain-ec2.xml** 文件中：

```
<local>  
  <discovery-options>  
    <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"  
code="org.jboss.as.host.controller.discovery.S3Discovery">  
      <property name="access-key" value="S3_ACCESS_KEY"/>  
      <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>  
      <property name="location" value="S3_BUCKET_NAME"/>  
    </discovery-option>  
  </discovery-options>  
</local>
```

第 5 章 启动集群 JBOSS EAP

5.1. 启动集群的 JBOSS EAP AMI（无需 MOD_CLUSTER 和 VPC）

本主题列出了在没有 `mod_cluster` 和 VPC 的情况下启动集群 JBoss EAP AMI 的步骤。



注意

- 您可以使用镜像提供的示例配置脚本。

有关 [系统路径](#) 的详情，请参阅 [系统路径](#)。

如需了解有关为 Amazon EC2 [配置 JBoss EAP 子系统的信息](#)，请参阅 [将 JBoss EAP 子系统配置为在云平台上工作](#)。

要在单机服务器实例上启动集群的 JBoss EAP AMI，您可以使用示例 `/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml` 文件，其中包含预配置的 `S3_PING` JGroups 堆栈。如需更多信息，请参阅 [与 JGroups 可靠的组通信](#) 中的 `S3_PING`。此 `standalone-ec2-ha.xml` 配置文件必须从 `/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/` 复制到 JBoss EAP 配置目录 `/opt/rh/eap7/root/usr/share/wildfly/standalone/configuration/`。然后，您必须将以下行添加到 JBoss EAP 服务配置文件中：

```
WILDFLY_SERVER_CONFIG=standalone-ec2-ha.xml
```

需要为 `undertow` 子系统中的每个单机服务器实例设置唯一的 `instance-id`。可以通过编辑 `standalone-ec2-ha.xml` 文件或使用管理 CLI 手动设置 `instance-id` 的值。例如，您可以使用管理 CLI 设置 `instance-id`，如下所示：

```
/subsystem=undertow:write-attribute(name=instance-id,value=${jboss.jvmRoute})
```

然后，可以使用 `JAVA_OPTS` 变量在 `standalone.conf` 中指定 `jboss.jvmRoute` 的值。

EC2 配置文件中的 `jgroups` 子系统需要一些 `S3_PING` 特定属性来发现群集成员。您必须指定要用于发现的 `S3`、`secret access key` 和 `S3 存储桶的访问密钥`。这些属性可以指定为 Java 选项，或者通过编辑或 CLI 直接放入 XML 文件中。

您需要创建一个 `S3 存储桶` 来发现。如需更多信息，请参阅 [Amazon Simple Storage Service 文档](#)。您可能还必须配置所需的权限。JGroups 堆栈需要绑定到 IP 地址，用于与其他节点通信。这可以通过添加 Java 选项以及 `S3 Java 选项` 到 `/opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf` 文件来完成。例如，如果专用 IP 地址为 `10.10.10.10`，则需要在 `standalone.conf` 文件中添加以下行：

```
JAVA_OPTS="$JAVA_OPTS -Djboss.bind.address.private=10.10.10.10"
```

您可以部署一个示例应用：`/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war` 并观察 `/opt/rh/eap7/root/usr/share/wildfly/standalone/log/server.log` 中的日志，查看 JBoss EAP 服务器是否已创建了集群。

对于域控制器实例

1. 将 `domain-ec2.xml` 文件从 `/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs` 复制到 JBoss EAP 配置目录中。
2. 在适当的服务配置文件中设置以下变量：

■

```
WILDFLY_SERVER_CONFIG=domain-ec2.xml
WILDFLY_HOST_CONFIG=host-master.xml
```

3. 在 **host-master.xml** 文件中添加 S3 域控制器发现配置：

```
<local>
  <discovery-options>
    <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
code="org.jboss.as.host.controller.discovery.S3Discovery">
      <property name="access-key" value="S3_ACCESS_KEY"/>
      <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>
      <property name="location" value="S3_BUCKET_NAME"/>
    </discovery-option>
  </discovery-options>
</local>
```

4. 配置用户，并将用户的机密值添加到主机控制器实例。如需更多信息，请参阅《JBoss EAP 配置指南》中的[在两台机器上创建受管域](#)。

对于主机控制器实例

1. 在适当的服务配置文件中设置以下变量：

```
WILDFLY_HOST_CONFIG=host-slave.xml
```

2. 在 **host-slave.xml** 文件中添加 S3 域控制器发现配置：

```
<remote security-realm="ManagementRealm">
  <discovery-options>
    <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
code="org.jboss.as.host.controller.discovery.S3Discovery">
      <property name="access-key" value="S3_ACCESS_KEY"/>
      <property name="secret-access-key" value="S3_SECRET_ACCESS_KEY"/>
      <property name="location" value="S3_BUCKET_NAME"/>
    </discovery-option>
  </discovery-options>
</remote>
```



注意

有关 S3 域控制器发现的详情，请参阅 [启动 One 或 More Instances to Serve as Host Controller](#)。



警告

在网络掩码小于 24 位或跨多个子网的子网中运行 JBoss EAP 群集会使每个群集成员获取唯一的服务器对等 ID。



重要

自动缩放 Amazon EC2 功能可用于 JBoss EAP 群集节点。不过，请确保它在部署前经过测试。您应该确保您的特定工作负载扩展到所需的节点数量，并确保性能满足您计划使用的实例类型的需求，不同的实例类型会收到 EC2 云资源的不同共享。

此外，实例本地性和当前网络/存储/主机计算机/RDS 利用率可能会影响群集性能。使用您预期的实时负载测试，并尝试考虑意外状况。



警告

Amazon EC2 *缩减* 操作会在没有任何机会正常关闭节点的情况下终止节点，因为可能会中断一些事务，其他集群节点和负载均衡器则需要时间来故障转移。这可能会影响您的应用程序用户体验。

建议您通过从 `mod_cluster` 管理界面禁用服务器，直到完成的会话完成，手动缩减应用群集，或者通过 SSH 访问实例或红帽 JBoss 运营网络安全地关闭 JBoss EAP 实例。

测试您的缩减流程不会对用户体验造成负面影响。特定的工作负载、负载均衡器和设置可能需要额外的测量结果。

5.2. 启动集群的 JBOSS EAP AMI（使用 MOD_CLUSTER 和 VPC）

本主题列出了启动 Apache HTTP 服务器实例以充当 `mod_cluster` 代理的步骤和虚拟私有云(VPC)的 NAT 实例。

有关 [系统路径](#) 的详情，请参见 [系统路径](#)。

如需了解有关为 Amazon EC2 [配置 JBoss EAP 子系统的信息](#)，请参见 [将 JBoss EAP 子系统配置为在云平台上工作](#)。



注意

- 您可以使用镜像提供的示例配置脚本。

Amazon Virtual Private Cloud(Amazon VPC)是 Amazon Web Services(AWS)的一项功能，允许您隔离专用网络中的一组 AWS 资源。此专用网络的拓扑和配置可以根据您的需求进行自定义。

有关 [Amazon VPC 的更多信息](#)，请参见 [Amazon Virtual Private Cloud](#)。



注意

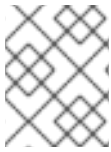
如果您在 VPC 中使用 `mod_cluster` 负载均衡器启动群集，则 JBoss EAP 服务器无法被公共访问。`mod_cluster` 负载均衡器可以是连接到互联网的唯一端点。

如需 [设置域控制器实例](#)，请参见 [启动实例到 Serve 作为 域控制器](#)。

如需设置 [主机控制器实例](#)，请参见 [启动一个或多个实例作为 主机控制器](#)。

如需有关 S3 域控制器发现的信息，请参阅启动 one 或 More Instances to Serve 作为主机控制器。

5.2.1. 使用 VPC 和 mod_cluster 启动集群 AMI



注意

配置 VPC 是可选的。如需更多信息，请参阅《Amazon VPC 用户指南》中的检测您支持的平台和是否具有默认 VPC 部分。

1. 安装 `jboss-httpd24-mod_cluster-native` 软件包及其所有依赖项。`mod_cluster` 配置文件安装在 `/opt/rh/jboss-httpd24/root/etc/httpd/conf.d/mod_cluster.conf` 中。

有关 [安装 Red Hat JBoss Core Services Apache HTTP 服务器的详情](#)，请查看 [Apache HTTP 服务器安装指南](#)。

1. 为 `mod_cluster` 禁用广播。在 `/opt/rh/jboss-httpd24/root/etc/httpd/conf.d/mod_cluster.conf` 配置文件中将以下内容添加到 `VirtualHost`：

```
ServerAdvertise Off
EnableMCPMReceive
# AdvertiseFrequency # comment out AdvertiseFrequency if present
```

2. 允许 SELinux 中的端口。如果需要，配置 `iptables`。可以使用 `semanage port -a -t http_port_t -p tcp $PORT_NR` 命令，在 SELinux 中允许端口。
3. 配置 JBoss EAP，以在 `mod_cluster` 侦听的地址上查找 `mod_cluster` 代理。



注意

提供了

`/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml` 示例配置文件。您需要在 `modcluster` 子系统中配置代理列表。

您可以使用以下方法之一定义代理列表：

- 定义带有适当主机和端口的 `outbound-socket-binding`，名为 `mod-cluster-proxy1`：

```
<outbound-socket-binding name="mod-cluster-proxy1">
  <remote-destination host="{jboss.modcluster.proxy1.host}"
port="{jboss.modcluster.proxy1.port}"/>
</outbound-socket-binding>
```

- 将 `modcluster` 子系统中的 `proxies` 属性设置为具有适当主机和端口的 `mod-cluster-proxy1`：

```
/socket-binding-group=standard-sockets/remote-destination-outbound-
socket-binding=mod-cluster-proxy1:add(host=
{jboss.modcluster.proxy1.host}, port={jboss.modcluster.proxy1.port})
```

第 6 章 故障排除

6.1. 关于 AMAZON EC2 故障排除

EC2 为每个实例提供警报状态，这表示严重实例出现故障，但没有这样的警报不能保证实例已正确启动，服务也正常运行。可以将 Amazon CloudWatch 与自定义指标功能搭配使用来监控实例服务的健康状态，但建议使用企业管理解决方案。

6.2. 诊断信息

如果 JBoss 运营网络、Amazon CloudWatch 或手动检查检测到问题，常见的诊断信息来源如下：

- `/var/log` 还包含从计算机启动、JBoss EAP、httpd 和大多数其他服务收集的所有日志。

JBoss EAP 日志文件可在 `/opt/rh/eap7/root/usr/share/wildfly/` 中找到。

仅可通过 SSH 会话访问这些文件。

如需有关如何配置和建立 Amazon EC2 实例的 SSH 会话的更多信息，请参阅 [Amazon EC2 Linux 实例入门](#)。

附录 A. 参考资料

A.1. 用于红帽云访问计划的 AMAZON EC2 AMI

AMIS 是红帽企业 Linux 镜像中的 JBoss EAP + JDK 的基本 RPM 安装，可能采用 Amazon EC2 示例配置。高级脚本不再可用，但可以使用常规 bash 脚本。

用于平台/JDK 组合的 AMIS :

- RHEL 6 + Open JDK 8 (1 镜像)
- RHEL 7 + Open JDK 8 (1 镜像)



注意

两种平台都应采用 64 位架构。

AMI 维护

yum update 应定期运行，以在 EC2 上应用 z 发行版本（补丁）红帽将提供 y 发行版本（次发行版本）的新 AMI。

场景 1（支持）

1. 注册 EC2。
2. 注册以获取红帽云访问。
3. 从可用 AMI 列表中选择 Red Hat AMI。
4. （可选）使用用户脚本或 **ssh** 自定义 JBoss EAP 配置。
5. Maintenance : **z 发行版本的 yum update**, y 发行版本的新 AMI。

A.2. 配置文件和部署示例

以下软件包为 RHEL 7 AMI 版本添加了示例部署 :

```
$ rpm -ql eap7-jboss-ec2-eap-samples
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/hello.war
/opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/jboss-as-helloworld-mdb-7.0.0.ER5-redhat-1.war
```

JBoss EAP 示例配置文件包含为 **S3_PING** 协议设置的 JGroups 堆栈，可用于跨 EC2 创建群集。有关示例配置文件的确切位置，请参阅 [系统路径](#)。

两个配置文件都包含 **modcluster** 子系统，以使用代理 mod_cluster 发现而不是广播，因为 EC2 上已禁用多播。

A.3. 系统路径

服务配置文件 :

- RHEL 6 : /etc/sysconfig/*

- RHEL 7 : /etc/opt/rh/eap7/wildfly/*

JBoss EAP Home:

- /opt/rh/eap7/root/usr/share/wildfly/

JBoss EAP 配置位置 :

单机实例

- /opt/rh/eap7/root/usr/share/wildfly/standalone/configuration
- /opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf

受管域

- /opt/rh/eap7/root/usr/share/wildfly/bin/domain.conf
- /opt/rh/eap7/root/usr/share/wildfly/domain/configuration

JBoss EAP 配置示例位置 :

- /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml
- /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-full-ha.xml

A.4. 使用脚本在 AMAZON EC2 上启动 JBOSS EAP

当您在 Amazon EC2 上启动 JBoss EAP 实例时，以下示例脚本可用于启动绑定到公共 IP 地址的 JBoss EAP。

```
#!/bin/bash

# platform dependent variables
if [[ "`cat /etc/redhat-release`" = *"release 7"* ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-standalone.conf
    START_COMMAND="systemctl start eap7-standalone"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-standalone
    START_COMMAND="service eap7-standalone start"
fi

# set up addresses
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' |
sed "s-/24--g" | cut -d/ -f1`
echo "JAVA_OPTS=\"$JAVA_OPTS -Djboss.bind.address=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.private=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.management=$INTERNAL_IP_ADDRESS\"" >>
/opt/rh/eap7/root/usr/share/wildfly/bin/standalone.conf

# start EAP
$START_COMMAND
```

A.5. 配置 JBOSS EAP 子系统以在云平台上工作

些 JBoss EAP 子系统必须配置为在云平台上正常工作，如 Amazon EC2 和 Microsoft Azure。这是必要的，因为 JBoss EAP 服务器通常绑定到云虚拟机的专用 IP 地址，例如：**10.x.x.x**，后者仅在云平台中可见。对于某些子系统，此地址还必须映射到服务器的公共 IP 地址，该地址可从云外部可见。

A.5.1. Web 服务

当客户端使用 `Service.create(wsdlURL, serviceName)` 发出 Web 服务请求时，用户将连接到服务器公共 IP 地址，但随后会被重定向到 `webservices` 子系统中服务器配置文件中定义的地址。默认情况下，此地址为 `/${jboss.bind.address:127.0.0.1}`，这意味着在云平台上调用者将重定向到服务器的专用 IP 地址，并且无法解析请求。使用以下命令，必须在 `wsdl-host` 元素中配置服务器的公共 IP 地址：

```
/subsystem=webservices:write-attribute(name=wsdl-host,value=PUBLIC_IP_ADDRESS)
```

A.5.2. 消息传递

在云平台上使用消息传递时，客户端使用的连接工厂必须具有指向服务器公共 IP 地址的连接器。

因此，必须为运行完整配置文件的 JBoss EAP 服务器创建新的连接器和套接字绑定。

1. 引用的 `http-public` 套接字绑定必须在 `socket-binding-group` 中创建：

```
/socket-binding-group=standard-sockets/remote-destination-outbound-socket-binding=http-public:add(host=PUBLIC_IP_ADDRESS,port=${jboss.http.port:8080})
```

2. 在 `messaging` 子系统中创建新的 `http-connector` 元素：

```
/subsystem=messaging-activemq/server=default/http-connector=http-public-connector:add(endpoint=http-acceptor, socket-binding=http-public)
```

3. 在 `connection-factory` 中设置连接器，供客户端使用。例如，将 `RemoteConnectionFactory` 配置为默认连接：

```
/subsystem=messaging-activemq/server=default/connection-factory=RemoteConnectionFactory:write-attribute(name=connectors, value=["http-public-connector"])
```

A.5.3. 远程配置高可用性

如果您在云平台上将 JBoss EAP HA 功能与集群 Jakarta 企业 Bean 搭配使用，则需要对远程子系统进行一些额外的配置，以确保 Jakarta 企业 Beans 客户端能够接收群集视图更新。

这可以通过为远程子系统套接字绑定配置客户端映射实现：

```
/socket-binding-group=standard-sockets/socket-binding=http:write-attribute(name=client-mappings,value=[{"destination-address" => "PUBLIC_IP_ADDRESS", "destination-port" => "8080"}])
```

A.6. 集群 JBOSS EAP 实例的用户数据示例

以下示例显示了为多个不同服务器配置的用户数据。

示例：RHEL6/7 上独立模式的文件

```
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a
# standalone instance using the ec2-ha profile
# This file is for RHEL 6/7, standalone mode only
#### This script makes use of the following four Bash variables for clustering setup,
#### be sure to add in your own values for these variables here when copy/pasting this
#### script into the EC2 user data field

ACCESS_KEY_ID=<your AWS access key>
SECRET_ACCESS_KEY=<your AWS secret access key>
S3_PING_BUCKET=<your bucket name>
NODE_NAME=<your node name>

#### No further modifications should be needed below to run this example ####
# Set the location of JBoss EAP
JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly

# Set the internal IP address of this EC2 instance which is mapped to a public address
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' |
sed "s-/24--g" | cut -d'/' -f1`

# Set the location of the standalone.conf file and set the command used to start EAP in standalone
mode
if [[ "`cat /etc/redhat-release`" = *"release 7"* ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-standalone.conf
    START_COMMAND="systemctl start eap7-standalone"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-standalone
    START_COMMAND="service eap7-standalone start"
fi

# Configure JBoss EAP to use the ec2-ha profile
cp /opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/standalone-ec2-ha.xml
$JBOSS_HOME/standalone/configuration/standalone-ec2-ha.xml
echo "WILDFLY_SERVER_CONFIG=standalone-ec2-ha.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_BIND=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS=\"\$JAVA_OPTS -Djboss.jgroups.s3_ping.access_key='$ACCESS_KEY_ID' -
Djboss.jgroups.s3_ping.secret_access_key='$SECRET_ACCESS_KEY' -
Djboss.jgroups.s3_ping.bucket='$S3_PING_BUCKET' -Djboss.jvmRoute=$NODE_NAME\"" >>
$JBOSS_HOME/bin/standalone.conf
echo "JAVA_OPTS=\"\$JAVA_OPTS -Djboss.bind.address=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.private=$INTERNAL_IP_ADDRESS\"" >> $JBOSS_HOME/bin/standalone.conf

# Deploy the sample application from the local filesystem
cp /opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war
$JBOSS_HOME/standalone/deployments/

# Start JBoss EAP, note that RHEL 7 does not wait for JBoss EAP to start before returning from the
service start. In some cases, there could be a delay of more than 90 seconds.

$START_COMMAND
```

示例：用于启动集群域实例的文件(Domain Controller)

```
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a domain
controller with clustering enabled
# This file is for RHEL 6/7, domain controller, domain mode only
### This script makes use of the following Bash variables for clustering and domain
### controller discovery setup, be sure to add in your own values for these variables here
### when copy/pasting this script into the EC2 user data field

ACCESS_KEY_ID=<your access key id>
SECRET_ACCESS_KEY=<your secret access key>
S3_PING_BUCKET=<your s3 ping bucket>

#### No further modifications should be needed below to run this example ####
# Set the location of JBoss EAP
JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly
CONF_FILE=/opt/rh/eap7/root/usr/share/wildfly/docs/examples/configs/domain-ec2.xml

# Set the internal IP address of this EC2 instance which is mapped to a public address

INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' |
sed "s-/24--g" | cut -d/ -f1`

# Set the location of the domain.conf file and set the command used to start EAP in domain mode
if [[ "`cat /etc/redhat-release`" = *"release 7"* ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-domain.conf
    START_COMMAND="systemctl start eap7-domain"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-domain
    START_COMMAND="service eap7-domain start"
fi

# Configure JBoss EAP to use the domain-ec2.xml and host-master.xml configuration files
cp ${CONF_FILE} $JBOSS_HOME/domain/configuration/domain-ec2.xml

echo "WILDFLY_SERVER_CONFIG=domain-ec2.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_HOST_CONFIG=host-master.xml" >> $SERVICE_CONF_FILE
echo "WILDFLY_BIND=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS=\"\$JAVA_OPTS -Djboss.jgroups.s3_ping.access_key='$ACCESS_KEY_ID' -
Djboss.jgroups.s3_ping.secret_access_key='$SECRET_ACCESS_KEY' -
Djboss.jgroups.s3_ping.bucket='$S3_PING_BUCKET'\"" >> $JBOSS_HOME/bin/domain.conf

echo "JAVA_OPTS=\"\$JAVA_OPTS -Djboss.bind.address=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.private=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.management=$INTERNAL_IP_ADDRESS\"" >>
$JBOSS_HOME/bin/domain.conf

echo 'HOST_CONTROLLER_JAVA_OPTS="$HOST_CONTROLLER_JAVA_OPTS $JAVA_OPTS"'
>> $JBOSS_HOME/bin/domain.conf

# Add a management user with the following credentials:
# User name: admin
# Password: secret_Passw0rd
$JBOSS_HOME/bin/add-user.sh -u admin -p secret_Passw0rd -e -g Management
```

```

# Update the main-server-group in domain-ec2.xml to use the ec2-ha profile
$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --domain-config=domain-ec2.xml, /server-group=main-server-group:write-attribute(name=profile, value=ha)"

# Need to modify permissions since this script is executed as the root user
chgrp jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chgrp jboss $JBOSS_HOME/domain/log/audit.log
chgrp jboss $JBOSS_HOME/domain/log/host-controller.log
chown jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chown jboss $JBOSS_HOME/domain/log/audit.log
chown jboss $JBOSS_HOME/domain/log/host-controller.log

# Configure S3 domain controller discovery
yum install patch -y
cd $JBOSS_HOME/domain/configuration
echo "--- host-master.xml 2016-03-18 17:34:26.000000000 -0400
+++ host-master2.xml 2016-04-11 08:28:02.771000191 -0400
@@ -54,7 +54,15 @@
     </management-interfaces>
     </management>
     <domain-controller>
-    <local/>
+<local>
+  <discovery-options>
+    <discovery-option name=\"s3-discovery\" module=\"org.jboss.as.host-controller\"
code=\"org.jboss.as.host.controller.discovery.S3Discovery\">
+      <property name=\"access-key\" value=\"$ACCESS_KEY_ID\"/>
+      <property name=\"secret-access-key\" value=\"$SECRET_ACCESS_KEY\"/>
+      <property name=\"location\" value=\"$S3_PING_BUCKET\"/>
+    </discovery-option>
+  </discovery-options>
+</local>
     </domain-controller>
     <interfaces>
       <interface name=\"management\">
" | patch host-master.xml

cd -

# Start JBoss EAP, do not forget that RHEL 7 does not wait for JBoss EAP to start before returning
from the service start. In some cases, there could be a delay of more than 90 seconds.

$START_COMMAND
sleep 20
# Set up EC2 HA socket bindings for main server group
$JBOSS_HOME/bin/jboss-cli.sh -c --controller=$INTERNAL_IP_ADDRESS:9990 --timeout=120000 -
-command='/server-group=main-server-group:write-attribute(name=socket-binding-group,value=ha-
sockets)'

# Deploy the sample application from the local filesystem to the main-server-group

```

```
$JBOSS_HOME/bin/jboss-cli.sh -c --controller=$INTERNAL_IP_ADDRESS:9990 --timeout=120000 -
-command='deploy /opt/rh/eap7/root/usr/share/java/eap7-jboss-ec2-eap-samples/cluster-demo.war --
server-groups=main-server-group'
```

示例：用于启动集群域实例的文件（主机控制器）

```
#!/usr/bin/env bash

# This is a sample script for the user data field for EC2, which demonstrates how to launch a host
controller with clustering enabled
# This file is for RHEL 6/7, host controller, domain mode only
### This script makes use of the following Bash variables for clustering and domain
### controller discovery setup, be sure to add in your own values for these variables here
### when copy/pasting this script into the EC2 user data field

ACCESS_KEY_ID=<your access key id>
SECRET_ACCESS_KEY=<your secret access key>
S3_PING_BUCKET=<your s3 ping bucket>

#### No further modifications should be needed below to run this example ####
# Set the location of EAP
JBOSS_HOME=/opt/rh/eap7/root/usr/share/wildfly

# Set the internal IP address of this EC2 instance which is mapped to a public address
INTERNAL_IP_ADDRESS=`ip addr show | grep eth0 -A 2 | head -n 3 | tail -n 1 | awk '{ print $2 }' |
sed "s-/24--g" | cut -d'/' -f1`

# Set the location of the domain.conf file and set the command used to start EAP in domain mode
if [[ "`cat /etc/redhat-release`" = *"release 7"* ]]; then
    SERVICE_CONF_FILE=/etc/opt/rh/eap7/wildfly/eap7-domain.conf
    START_COMMAND="systemctl start eap7-domain"
else
    SERVICE_CONF_FILE=/etc/sysconfig/eap7-domain
    START_COMMAND="service eap7-domain start"
fi

# Configure variables needed by JBoss EAP
echo "WILDFLY_BIND=$INTERNAL_IP_ADDRESS" >> $SERVICE_CONF_FILE
echo "WILDFLY_HOST_CONFIG=host-slave.xml" >> $SERVICE_CONF_FILE
echo "JAVA_OPTS=\"$JAVA_OPTS -Djboss.jgroups.s3_ping.access_key='$ACCESS_KEY_ID' -
Djboss.jgroups.s3_ping.secret_access_key='$SECRET_ACCESS_KEY' -
Djboss.jgroups.s3_ping.bucket='$S3_PING_BUCKET'\"" >> $JBOSS_HOME/bin/domain.conf
echo "JAVA_OPTS=\"$JAVA_OPTS -Djboss.bind.address=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.private=$INTERNAL_IP_ADDRESS -
Djboss.bind.address.management=$INTERNAL_IP_ADDRESS\"" >>
$JBOSS_HOME/bin/domain.conf
echo 'HOST_CONTROLLER_JAVA_OPTS="$HOST_CONTROLLER_JAVA_OPTS $JAVA_OPTS"'
>> $JBOSS_HOME/bin/domain.conf

# Configure S3 domain controller discovery
yum install patch -y
cd $JBOSS_HOME/domain/configuration

echo "--- host-slave.xml.orig 2016-06-07 09:55:27.183390617 +0200
+++ host-slave.xml 2016-06-07 09:56:52.540170784 +0200
@@ -57,7 +57,11 @@
```

```

<domain-controller>
  <remote security-realm="ManagementRealm">
    <discovery-options>
-     <static-discovery name="primary" protocol="\${jboss.domain.master.protocol:remote}"
host="\${jboss.domain.master.address}" port="\${jboss.domain.master.port:9990}"/>
+     <discovery-option name="s3-discovery" module="org.jboss.as.host-controller"
code="org.jboss.as.host.controller.discovery.S3Discovery">
+       <property name="access-key" value="\${ACCESS_KEY_ID}"/>
+       <property name="secret-access-key" value="\${SECRET_ACCESS_KEY}"/>
+       <property name="location" value="\${S3_PING_BUCKET}"/>
+     </discovery-option>
    </discovery-options>
  </remote>
</domain-controller>
" | patch host-slave.xml

sed -i 's/<!--.*-->/g' host-slave.xml # remove nasty '!' signs which break bash
sed -i '/^[ ]*$d' host-slave.xml # remove nasty lines with ' ' whitespaces which break the patch

EAP_HOST_NAME=`$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --host-
config=host-slave.xml, :read-resource" | grep "host" | cut -d" " -f4`
$JBOSS_HOME/bin/jboss-cli.sh --commands="embed-host-controller --host-config=host-slave.xml,
/host=$EAP_HOST_NAME/core-service=management/security-realm=ManagementRealm/server-
identity=secret:write-attribute(name=value,value=c2VjcmV0X1Bhc3N3MHJk)"

sed -i 's/<host xmlns="urn:jboss:domain:8.0"/><host xmlns="urn:jboss:domain:8.0" name="admin"/>/'
host-slave.xml
sed -i 's/other-server-group/main-server-group/' host-slave.xml

cd -

# Need to modify permissions since this script is executed as the root user
chgrp jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chgrp jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chgrp jboss $JBOSS_HOME/domain/log/audit.log
chgrp jboss $JBOSS_HOME/domain/log/host-controller.log
chown jboss $JBOSS_HOME/domain/configuration/domain_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/host_xml_history/
chown jboss $JBOSS_HOME/domain/configuration/domain-ec2.xml
chown jboss $JBOSS_HOME/domain/log/audit.log
chown jboss $JBOSS_HOME/domain/log/host-controller.log

# Start JBoss EAP, do not forget that RHEL 7 does not wait for JBoss EAP to start before returning
from the service start. In some cases, there could be a delay of more than 90 seconds.
$START_COMMAND

```

更新于 2024-02-08

