Jboss运用

1. 下载：

地址:https://developers.redhat.com/products/eap/download

1. 解压

Unzip

在Linux系统中，解压缩ZIP格式文件的常用命令是unzip。

解压一个名为example.zip的文件到当前目录，可以使用如下命令：

unzip example.zip

如果想要将文件解压到指定的目录，可以使用-d参数 followed by the destination directory:

unzip example.zip -d /path/to/destination

1. 配置环境变量

vi /etc/profile，添加下列内容：

#set jboss environment

JBOSS\_HOME=/usr/local/Jboss/jboss

PATH=$PATH:$JBOSS\_HOME/bin

CLASSPATH=.:$CLASSPATH:$JBOSS\_HOME/lib

export JBOSS\_HOME PATH CLASSPATH

退出vi，然后source /etc/profile，使配置生效！

1. 启动Jboss

启动JBoss的命令通常为./standalone.sh或nohup ./standalone.sh &，具体取决于您希望如何运行JBoss。

前台启动：如果您希望JBoss在当前命令行窗口中运行，并能看到输出的日志信息，可以直接在bin目录下执行./standalone.sh。这种方式的缺点是，如果关闭了命令行窗口，JBoss服务也会随之停止。

后台启动：如果您希望JBoss在后台运行，即使关闭命令行窗口也不影响服务，可以使用nohup ./standalone.sh &命令。这样，JBoss将在后台运行，所有输出将重定向到nohup.out文件中。

此外，如果您使用的是较旧版本的JBoss，如JBoss 4.x，则启动命令可能是run，并且可以通过指定-c参数来选择不同的配置文件集启动服务器。

在实际操作中，请确保您位于JBoss的bin目录下执行上述命令，并根据需要选择适当的启动方式。

查看占用的进程：

ps aux | grep java | grep -v grep

netstat -ano | grep :8080

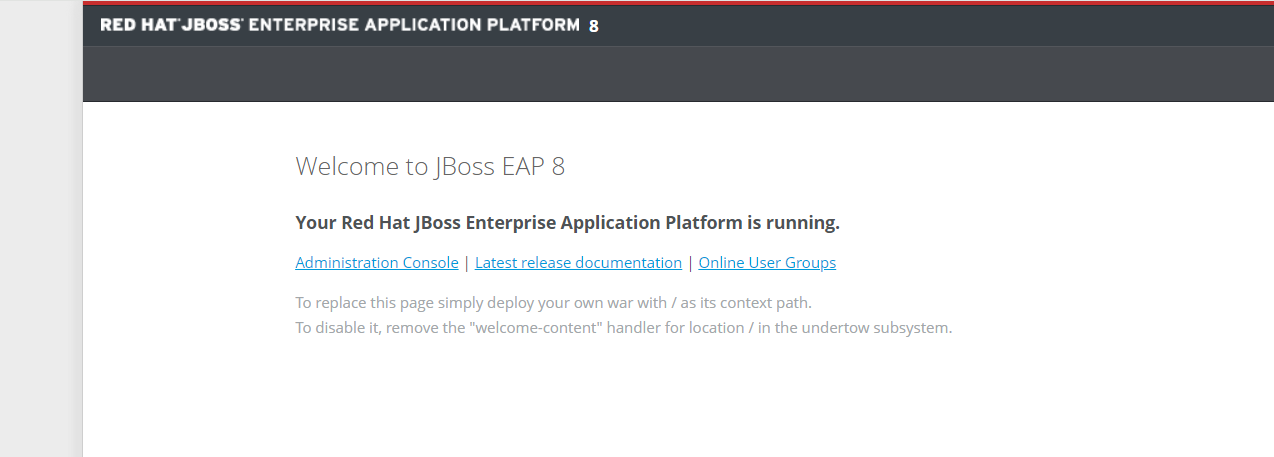
sudo lsof -i :8080

关闭占用的进程：

kill -9 PID

1. 端口占用的情况 先将之前的8080端口改成8090
2. 再启动Jboss
3. 验证成功

浏览器输入：<http://8.138.26.188:8080/> 出现jboss页面



在Linux环境下安装JBoss的基本步骤如下：

系统更新：确保您的Linux系统是最新的，这可以通过系统的包管理器如apt或yum来完成。

Java环境：安装Java Development Kit（JDK），因为JBoss运行需要Java环境。您可以使用java -version来检查是否已经安装了Java，并且版本是否符合JBoss的要求。

下载JBoss：从JBoss官方网站或者镜像站点下载最新版本的JBoss应用服务器。

解压文件：将下载的JBoss压缩包解压到您选择的目录中，通常可以使用tar命令来完成这一步骤。

配置环境变量：设置环境变量，例如JBOSS\_HOME指向JBoss的安装目录，以及更新PATH变量以包含JBoss的bin目录。

启动JBoss：通过运行位于$JBOSS\_HOME/bin目录下的启动脚本（如standalone.sh）来启动JBoss服务器。

验证安装：启动后，通过访问默认的HTTP端口（通常是8080）在Web浏览器中验证JBoss是否成功安装并正在运行。

停止JBoss：如果需要停止JBoss服务，可以执行相应的关闭脚本，通常是$JBOSS\_HOME/bin/jboss-cli.sh --connect command=:shutdown。

配置应用程序：部署您的企业应用程序到JBoss上，根据需要调整配置文件和部署描述符。

维护和监控：定期检查JBoss的日志文件以监控系统状态，并进行必要的维护工作。

请注意，这些步骤是一个大致的指南，具体操作可能会根据您的Linux发行版和JBoss版本有所不同。在实际操作过程中，建议参考JBoss的官方文档和社区资源，以获取更详细的指导信息。此外，如果您在安装过程中遇到任何问题，可以查阅相关错误信息，或在JBoss社区论坛中寻求帮助。

举例1：

Jboss是一个开源的Java EE应用服务器，用于运行和托管Java Web应用程序。它提供了丰富的功能，包括Web容器、EJB容器、消息队列、安全认证等，可以帮助开发者快速构建和部署企业级应用程序。

举个例子，假设你正在开发一个在线银行系统，需要处理大量的用户请求和交易操作。为了提高系统的可靠性和性能，你可以使用Jboss作为你的应用服务器。

首先，你需要将你的银行系统代码打包成一个WAR（Web Application Archive）文件，然后将其部署到Jboss服务器上。部署完成后，当用户在浏览器中输入你的银行系统地址时，Jboss会接收到这个HTTP请求，并根据请求的URL找到对应的Servlet或JSP文件进行处理。同时，Jboss还提供了EJB容器，可以方便地实现业务逻辑的分布式处理和事务管理。

例如，当用户进行转账操作时，Jboss会找到对应的Servlet或JSP文件，执行其中的Java代码，然后将结果返回给用户的浏览器显示。同时，Jboss还可以提供安全管理功能，确保用户的交易操作安全可靠。

总之，Jboss是一个非常强大的工具，可以帮助Java开发者轻松地搭建和管理企业级应用程序，提供稳定可靠的服务给用户。

举例2：

假设你正在开发一个在线商城网站，需要展示商品信息和处理用户的订单操作。为了提高网站的响应速度和稳定性，你可以使用Jboss作为你的应用服务器。

首先，你需要将你的商城网站代码打包成一个WAR（Web Application Archive）文件，然后将其部署到Jboss服务器上。部署完成后，当用户在浏览器中输入你的商城网站地址时，Jboss会接收到这个HTTP请求，并根据请求的URL找到对应的Servlet或JSP文件进行处理。同时，Jboss还提供了EJB容器，可以方便地实现业务逻辑的处理和事务管理。

例如，当用户浏览商品列表时，Jboss会找到对应的Servlet或JSP文件，执行其中的Java代码，然后将结果返回给用户的浏览器显示。同时，Jboss还可以提供安全管理功能，确保用户的交易操作安全可靠。

总之，Jboss是一个非常强大的工具，可以帮助Java开发者轻松地搭建和管理企业级应用程序，提供稳定可靠的服务给用户。

JBoss作为一种应用服务器，通常会使用一些特定的端口来提供服务和管理功能。以下是JBoss常用的一些端口：

8080：这是JBoss用于HTTP服务的默认端口。

8009：用于AJP13（Apache/Tomcat Connector）协议，与Apache服务器集成时使用。

8443：用于HTTPS服务，如果你配置了JBoss支持SSL加密通信，那么这个端口会被用到。

1099：用于JNDI（Java Naming and Directory Interface）服务。

1098：用于JBoss的远程管理服务。

4444、4445：用于JRMP（Java Remote Method Protocol）通信。

需要注意的是，这些端口都可以根据实际需要进行调整。如果遇到端口冲突或者需要改变默认设置，可以在JBoss的配置文件中进行修改。例如，如果你需要修改HTTP服务的端口，可以编辑jboss-4.2.3.GA/server/default/deploy/jboss-web.deployer/server.xml文件，将默认的8080端口改为其他未被占用的端口号。

在实际操作中，记得在修改端口后重启JBoss服务以使新设置生效，并确保新的端口不会与其他服务发生冲突，以免影响系统的正常运行。同时，出于安全考虑，建议仅在必要时开放服务器的端口，并使用防火墙等安全措施保护这些开放的端口。

应用：

JBoss是一款开源的应用服务器，它支持EJB规范和集群，能够集成Tomcat等Servlet容器。

以下是JBoss的主要应用场景：

企业级应用：JBoss被用作企业级应用平台的基础服务器，适用于部署和管理大型的企业应用。

服务架构：由于JBoss本身就是面向服务架构（Service-Oriented Architecture, SOA），它适用于需要SOA解决方案的系统开发和部署。

统一类加载器：JBoss具有统一的类加载器，可以实现应用的热部署和热卸载，适合需要快速更新和维护的动态环境。

中小型公司：JBoss主要应用于需要EJB服务的中小型公司，因为它是开源且免费的，适合预算有限但需要可靠服务的企业。

后台管理界面：JBoss提供了后台管理界面，方便管理员进行配置和管理。

开源社区支持：由于JBoss是由开源社区开发的，因此有着广泛的用户基础和资源，适合喜欢开源文化的开发者和技术团队。

安全性考虑：虽然JBoss作为开源软件在安全性方面可能不如商业软件那么成熟，但它仍然适用于那些能够自行处理安全问题或不需要高级安全功能的应用场景。

与数据库的结合：JBoss可以与多种数据库紧密结合，适用于需要数据库支持的复杂应用系统。

Red Hat收购：2006年，JBoss被红帽公司收购，并更名为WildFly，这标志着JBoss在企业级应用服务器领域的进一步发展和成熟。

综上所述，JBoss作为一个开源的应用服务器，虽然在某些方面可能存在局限性，但其灵活性、开源特性和成本效益使其成为许多项目和企业的理想选择。随着技术的不断进步和社区的支持，JBoss及其后续版本WildFly继续在企业应用开发和部署中发挥着重要作用。