# 官网

http://wildfly.org/

# 概念

JBoss是全世界开发者共同努力的成果，一个基于[J2EE](http://baike.baidu.com/view/1507.htm)的开放源代码的应用服务器。 因为JBoss代码遵循LGPL许可，你可以在任何商业应用中免费使用它，而不用支付费用。

JBoss是一个基于J2EE的[开放源代码](https://baike.baidu.com/item/%E5%BC%80%E6%94%BE%E6%BA%90%E4%BB%A3%E7%A0%81)的[应用服务器](https://baike.baidu.com/item/%E5%BA%94%E7%94%A8%E6%9C%8D%E5%8A%A1%E5%99%A8)，JBoss是一个管理EJB的容器和服务器，支持EJB 1.1、EJB 2.0和EJB3的规范，但JBoss核心服务不包括支持servlet/JSP的WEB容器，一般与Tomcat或Jetty绑定使用。

# 特点

1、JBoss是免费的，开放源代码的J2EE实现，它通过LGPL许可证进行发布。

2、JBoss需要的内存和硬盘空间比较小。

3、具有统一的类装载器，从而能够实现应用的[热部署](https://baike.baidu.com/item/%E7%83%AD%E9%83%A8%E7%BD%B2)和热卸载能力。

4、高度模块化和松耦合，JBoss应用服务器是健壮的、高质量的，而且还具有良好的性能。

5、安装便捷：解压后，只需配置一些[环境变量](https://baike.baidu.com/item/%E7%8E%AF%E5%A2%83%E5%8F%98%E9%87%8F)即可。

6、JBoss支持"热部署"，部署Bean时，只拷贝Bean的[JAR](https://baike.baidu.com/item/JAR)文件到部署路径下即可自动加载，如果有改动也会自动更新。

7、JBoss与Web服务器在同一个Java虚拟机中运行，Servlet调用EJB不经过网络，从而大大提高运行效率，提升安全性能。

8、用户可以直接实施J2EE-EAR，而不是以前分别实施EJB-JAR和Web-WAR，非常方便。

# jboss中重要的文件

## jboss-service.xml

启动时运行的jboss服务的定义（类加载器、JNDI、deployer等）

## log4j.xml

Log4j的配置

## login-config.xml

jboss的安全配置

## standalone.xml（jboss6.3以上的配置文件）

模块是用于类加载和依赖关系管理的类的逻辑分组。Jboss EAP 6识别两种不同类型的模块，有时称为静态和动态模块。

### 静态模块

静态模块被预定义在EAP\_HOME/modules/这个目录下面，所有应用程序服务器提供的API都作为静态模块提供，包括Java EE API以及JBoss Logging等其他API。

### 动态模块

我们部署的jar包和war包都被加载为动态模块，所有这些module只有需要的时候再加载。

standalone.xml可以指定server需要加载哪些model，并且对每个model都可以定义一个subsystem元素来指定其属性。另外 standalone.xml中也可以定义socket-binding-group元素，指定端口对应关系。

例如如果我们已经定义了

<http-interface security-realm="ManagementRealm">

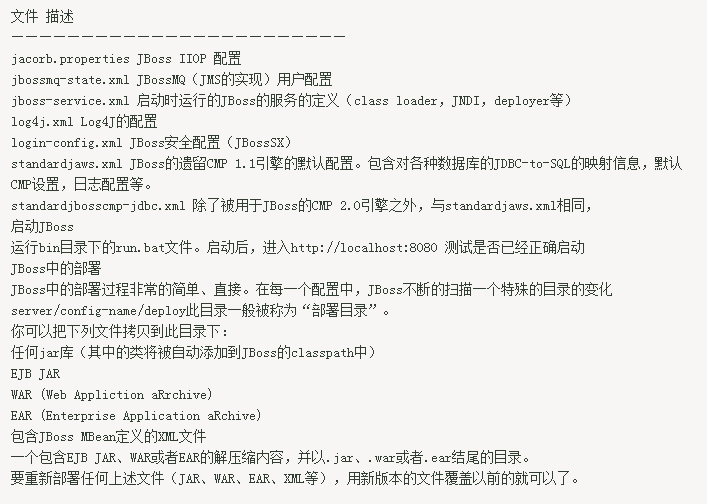
                <socket-binding http="management-http"/>（绑定management-http）

</http-interface>

那么我们在socket-binding-group中具体定义management-http绑定哪个端口。

<socket-binding name="management-http" interface="management" port="${jboss.management.http.port:9990}"/>

默认的话jboss web console的端口为9990



# Jboss目录结构说明（这是4.x版本的）



bin目录：启动和关闭 JBoss 的脚本（ run.bat 为 windows 系统下的启动脚本，shutdown.bat 为 windows 系统下的关闭脚本）

client：客户端与 JBoss 通信所需的 Java 库（ JARs ）

docs：配置的样本文件（数据库配置等）

docs/dtd：在 JBoss 中使用的各种 XML 文件的 DTD

lib：一些 JAR ， JBoss 启动时加载，且被所有 JBoss 配置共享。（不要把你的库放在这里）

**server**：各种 JBoss 配置。每个配置必须放在不同的子目录，子目录的名字表示配置的名字。 JBoss 包含 3 个默认的配置： minimial ， default 和 all ，在你安装时可以进行选择。

**server/all**：JBoss 的完全配置，启动所有服务，包括集群和 IIOP（互联网内部对象请求代理协议）

**server/default**：JBoss 的默认配置，没有在 JBoss 命令行中指定配置名称时使用。 ( 我们下载的 JBOSS5.0 Beta4 版本默认采用此配置 )  。

# 环境搭建（Jboss4.x以上版本）

准备：JBoss 5.0下载和JDK1.6下载

## 第一步：JDK1.6安装

不说了

## 第二步：JBoss 5.0安装

使用JBoss-5.0.1.GA.zip稳定版本，设置环境变量JBoss\_HOME=D:\JBoss\JBoss-5.0.1.GA\JBoss-5.0.1.GA

## 第三步：启动

在D:\JBoss\JBoss-5.0.1.GA\JBoss-5.0.1.GA\bin目录下，修改run.bat文件

在第一行加上如下内容：

”set JAVA\_HOME=JDK的安装目录“（如set JAVA\_HOME=D:\Program Files\Java\jdk1.6.0\_14）

然后双击run.bat文件就启动jboss服务器了

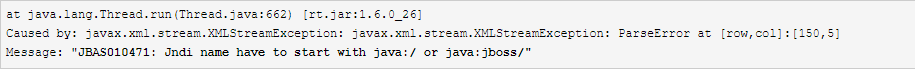
# jboss6.3基本知识

1、jboss6.3使用的是模块化加载jar包的方式，并且也包含了jboss之后版本的domain和standalone两种启动方式。此外jboss6.3的配置是standalone.xml文件，避免了jboss5中混乱的配置结构。

2、数据源的配置

  在jboss5中配置数据源的方式比较简单，在segment（如server下的default节点）下的lib文件夹中加入数据库驱动jar，然后在segment的deploy文件夹中添加name-ds.xml文件，jboss便可以自动配置数据源。而jboss6.3中进行了模块化设计，所以与jboss5的配置有所不同。

在配置的过程中，原本jboss5中的-ds.xml文件中的数据源为datasource1.我一开始使用的方式为全部照抄。结果出现异常：



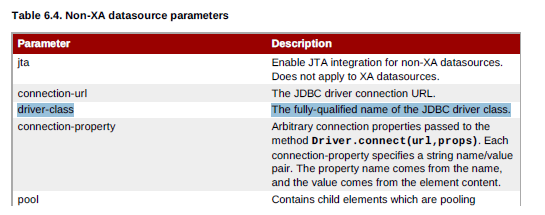
即所有的jndi都要以java:/或java:jboss/开头，这也意味着你的工程中的所有的数据源都将失效。

3、数据库驱动问题

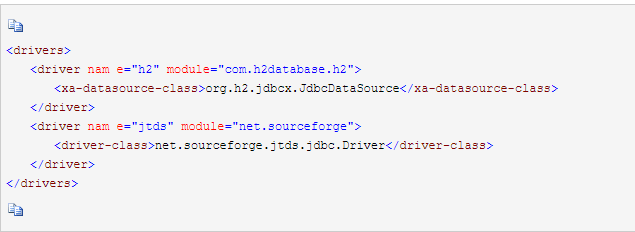
下面是在standalone.xml中添加的驱动：



上面的就是没有改过的，第一个驱动是jboss自带的h2，第二个是我自己写的jtds，由于我之前做过jboss8配置mysql的数据源，也是使用驱动做的配置，并且连接是成功的，所以下意识的认为这也是可以的，后来在一篇国外的博客中发现，在驱动的配置中，除了<xa-datasource-class>这个标签之外，还有<driver-class>标签，专门用于Driver类型的一个标签，这个标签我也查了一下jboss6的开发文档，有写，但只是在一个表格中提了一下，没有示例容易忽略。



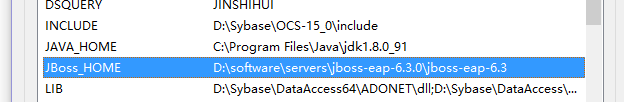
将其改为：



这样配置就可以连接成功了。

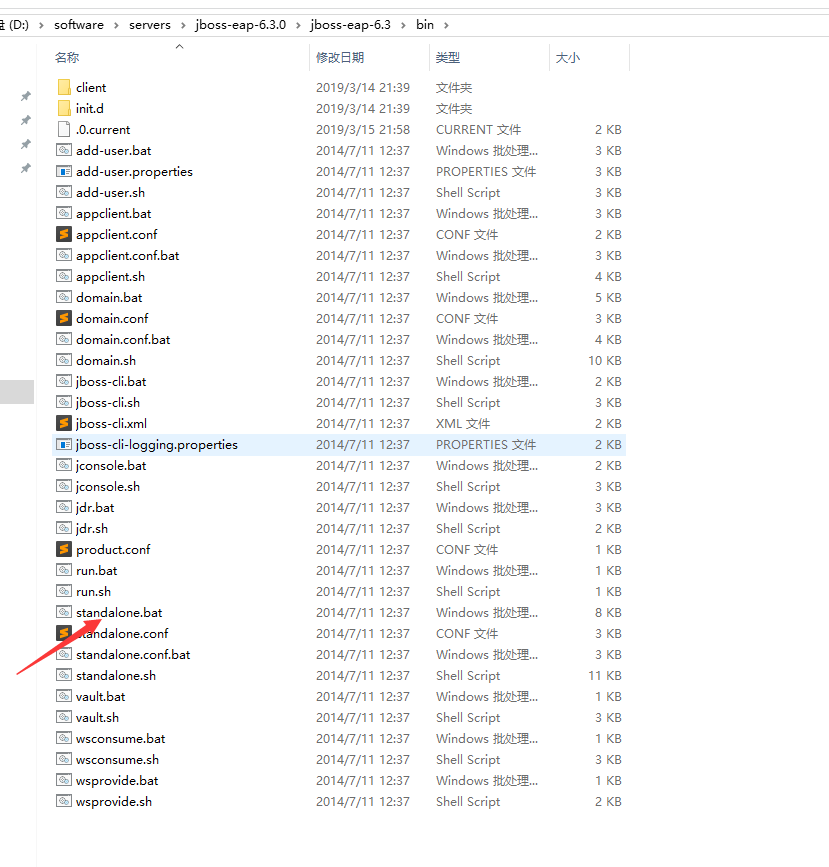
# 环境搭建（jboss6.3版本）

## 1、配置环境变量



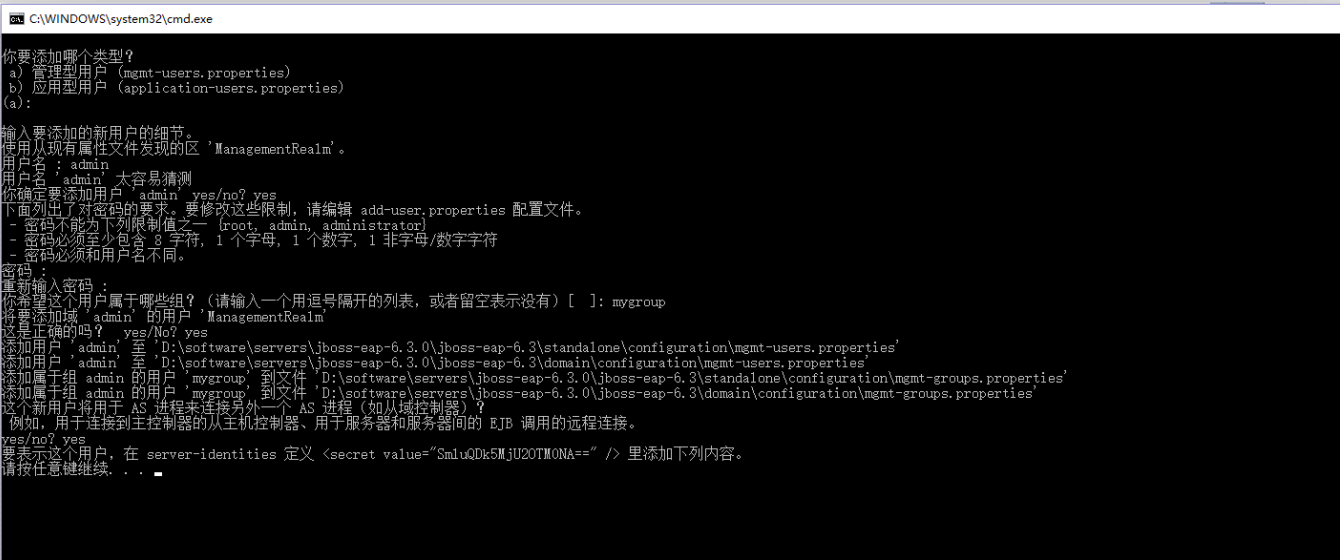
## 2、启动Jboss服务器

双击bin目录下面的standalone.bat文件



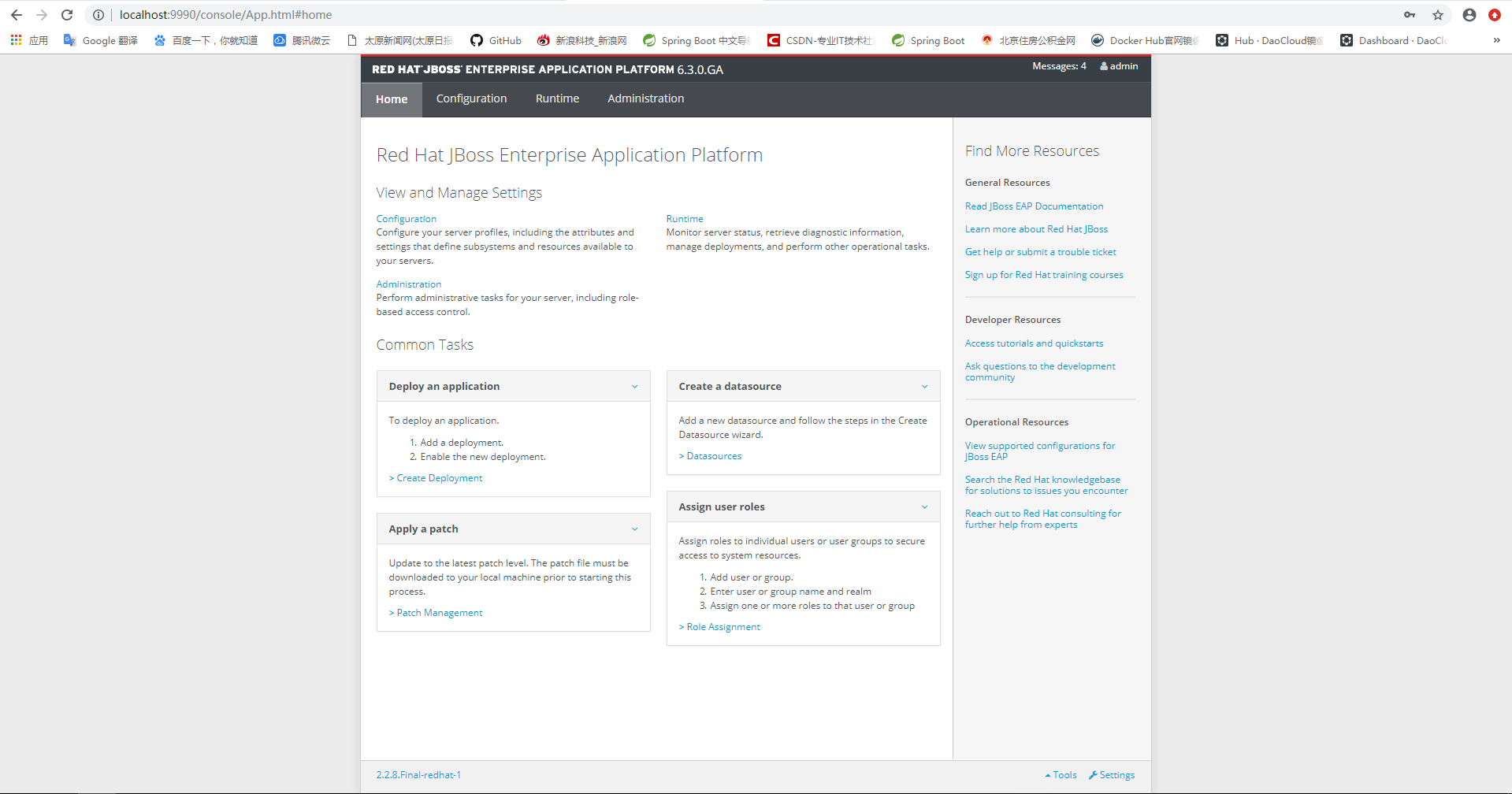
## 3、添加管理用户

双击bin目录下面的add-user.bat文件，操作流程如下



## 4、访问控制台

到此jboss的管理用户就添加成功了，可以在IE地址栏里输入http://127.0.0.1:9990/console/进行访问了，如下图：



用户名：admin

密码：Jin@992569344

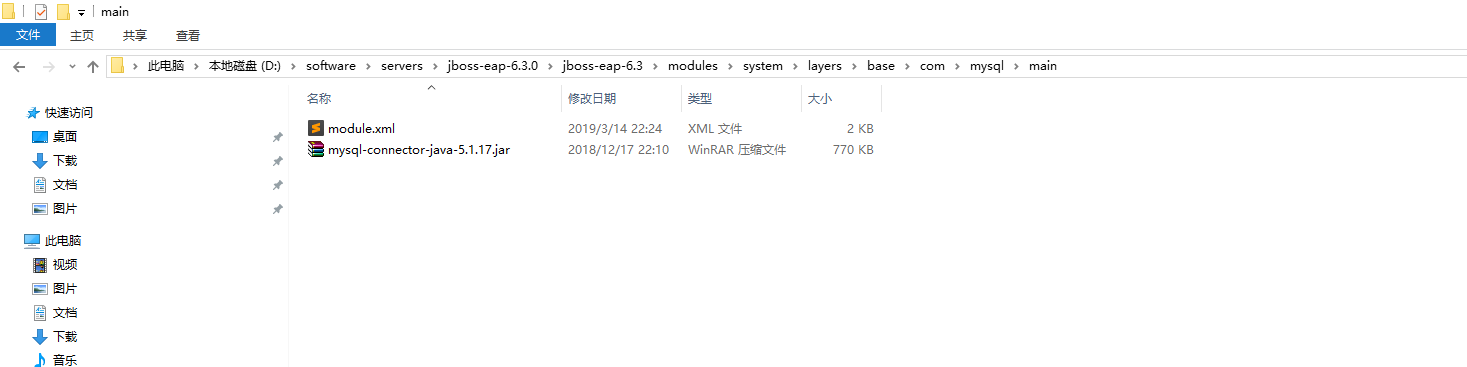
## 5、配置数据源(重点)

### 通过配置文件配置数据源

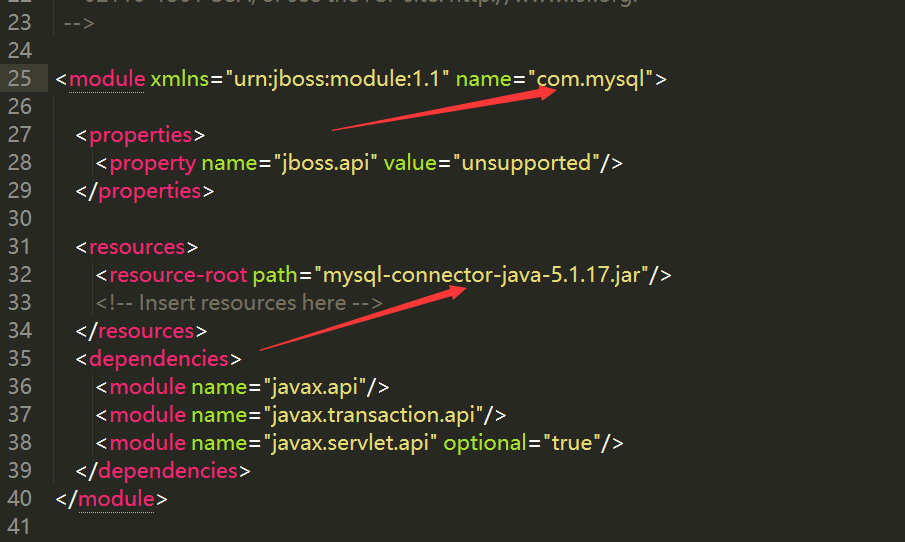
（1）在D:\jboss\jboss-eap-6.3\modules\system\layers\base\com目录下创建mysql\main目录

（2）拷贝mysql-connector-java-5.1.2.jar文件到这个目录下

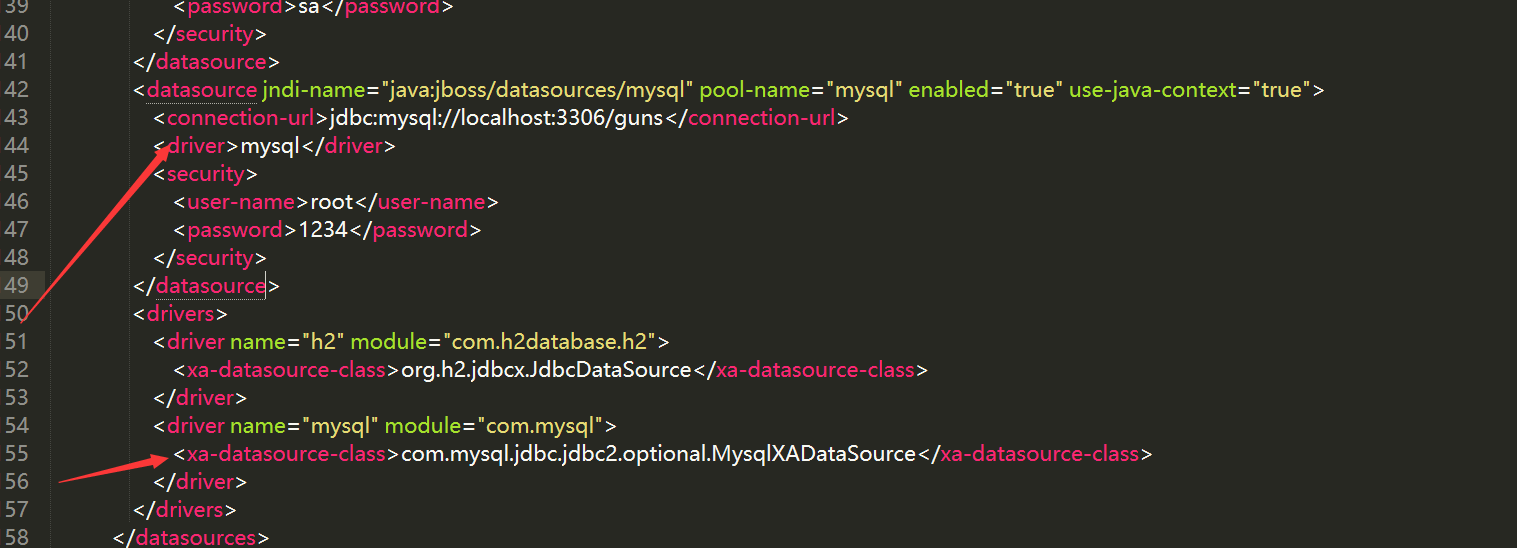
（3）到D:\jboss\jboss-eap-6.3\modules\system\layers\base\com\h2database\h2\main目录下拷贝module.xml文件到D:\jboss\jboss-eap-6.3\modules\system\layers\base\com\mysql\main目录下，如下图所示：



（4）D:\jboss\jboss-eap-6.3\modules\system\layers\base\com\mysql\main目录下的module.xml文件，修改下图中的两个地方。

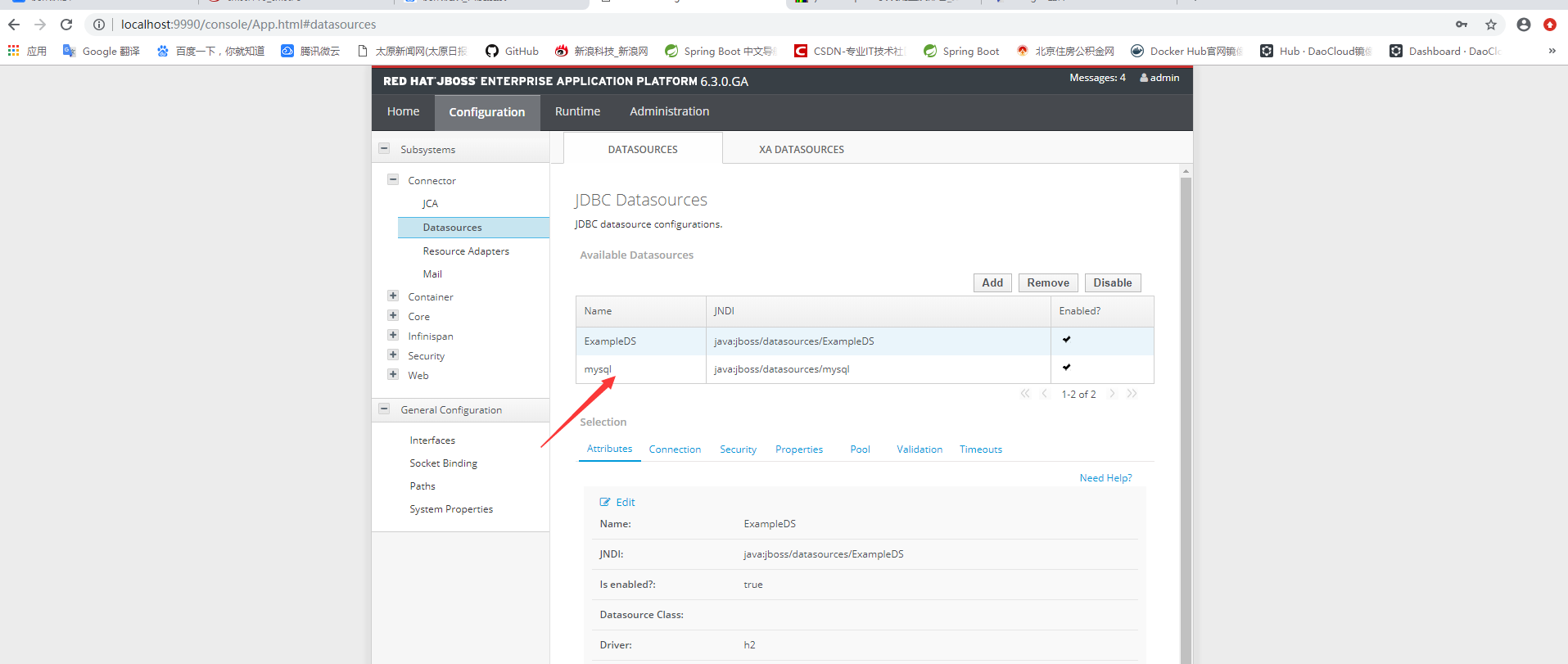


（5）修改D:\jboss\jboss-eap-6.3\standalone\configuration目录下的standalone.xml文件内容，修改这个文件来添加外部数据源，修改下图中的两个地方



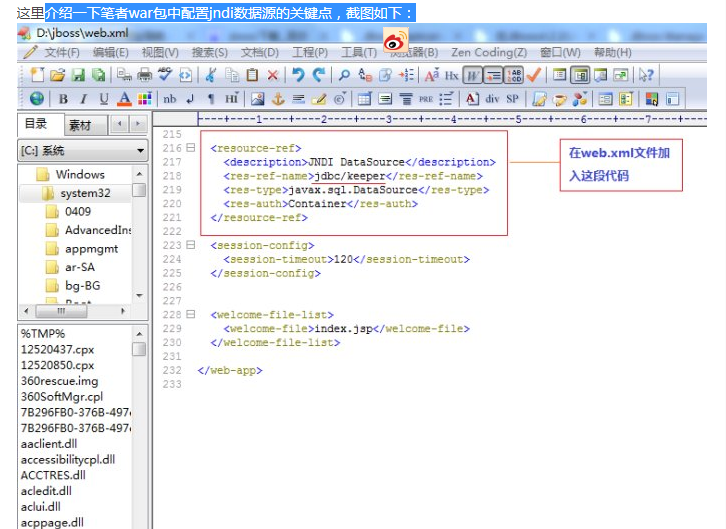
注意：其中xa-datasource-class属性的值为com.mysql.jdbc.jdbc2.optional.MysqlXADataSource

重新双击standalone.bat脚本启动jboss就可以看到刚刚配置好的mysql的jndi数据源了，截图如下

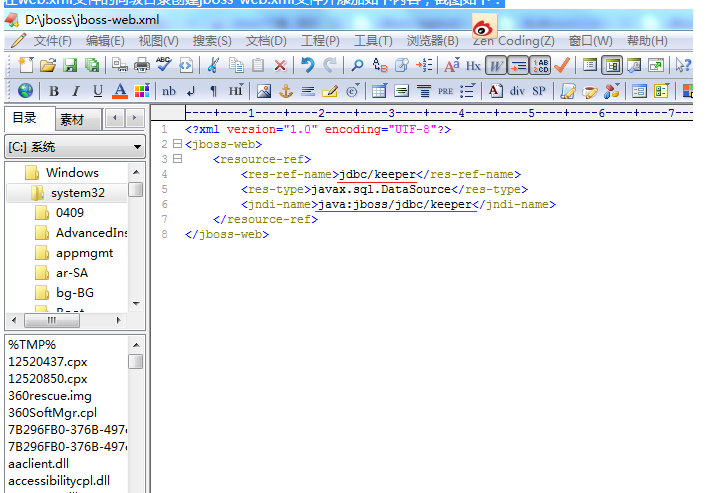


### war包中配置JNDI数据源

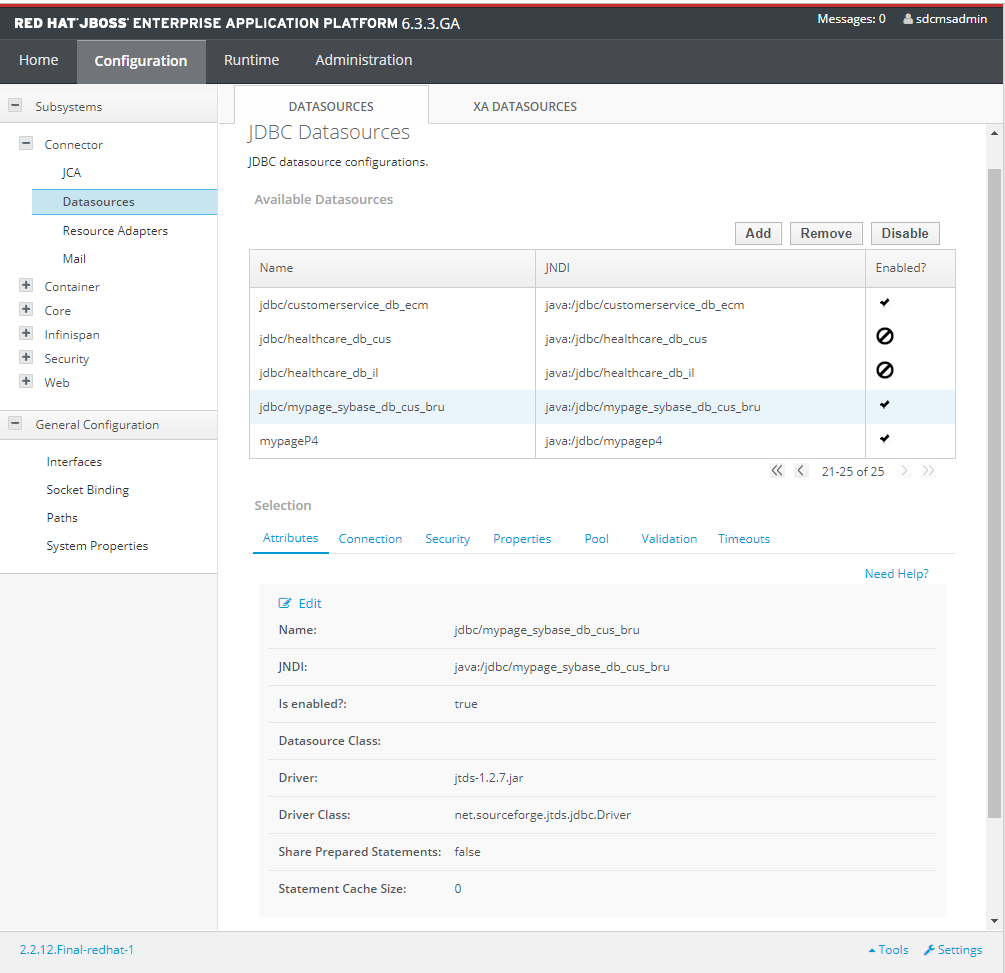
介绍一下war包中配置jndi数据源的关键点，截图如下：



在web.xml文件的同级目录创建jboss-web.xml文件并添加如下内容，截图如下：

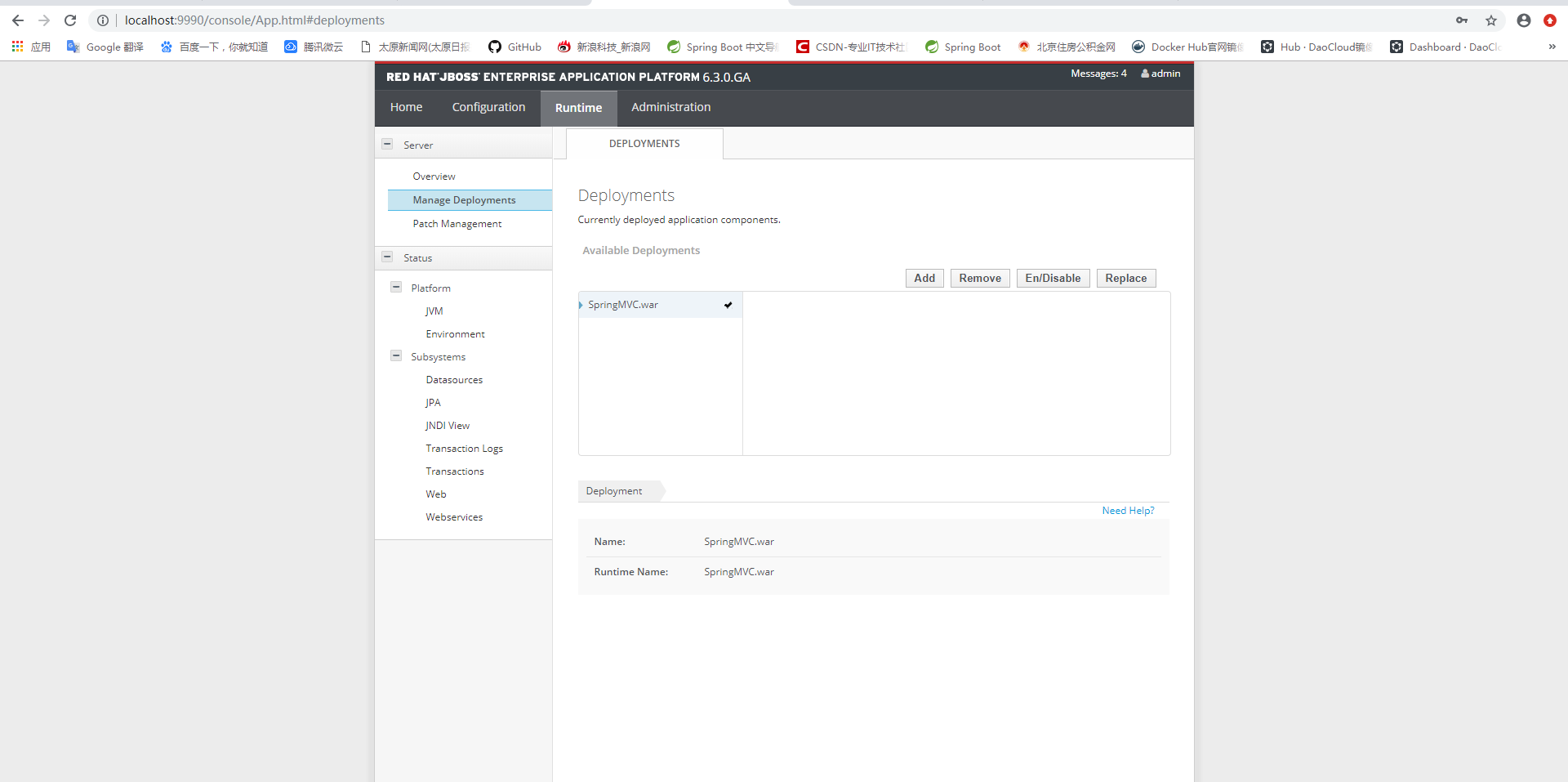


### 通过jboss控制台去配置数据源

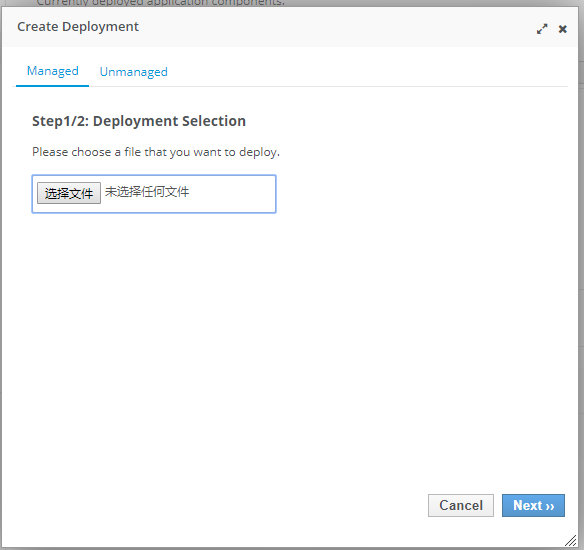


## 6、部署war包到jboss控制台中

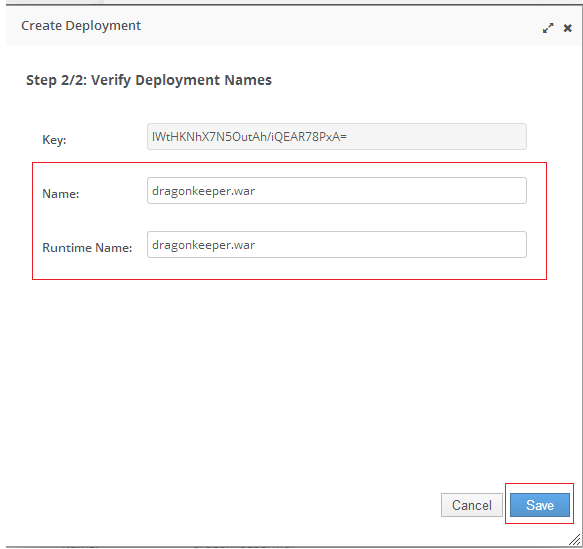
打开jboss控制台，点击runtime—manage deployments，



点击add，添加war包



选择本地的war包去添加，点击next，



点击save即可。

## 7、日志文件路径

jboss-eap-6.3\standalone\log\server.xml

# Jboss 7.2版本

## 基本说明

RedHat发布了[JBoss 7.2](https://blog.csdn.net/weixin_34362790/article/details/%5C)，完全支持Java EE 8规范。值得注意的改进包括新的Servlet 4.0规范，它增加了对跨浏览器HTTP/2连接的支持，以及对Java微服务和Eclipse MicroProfile框架的支持。

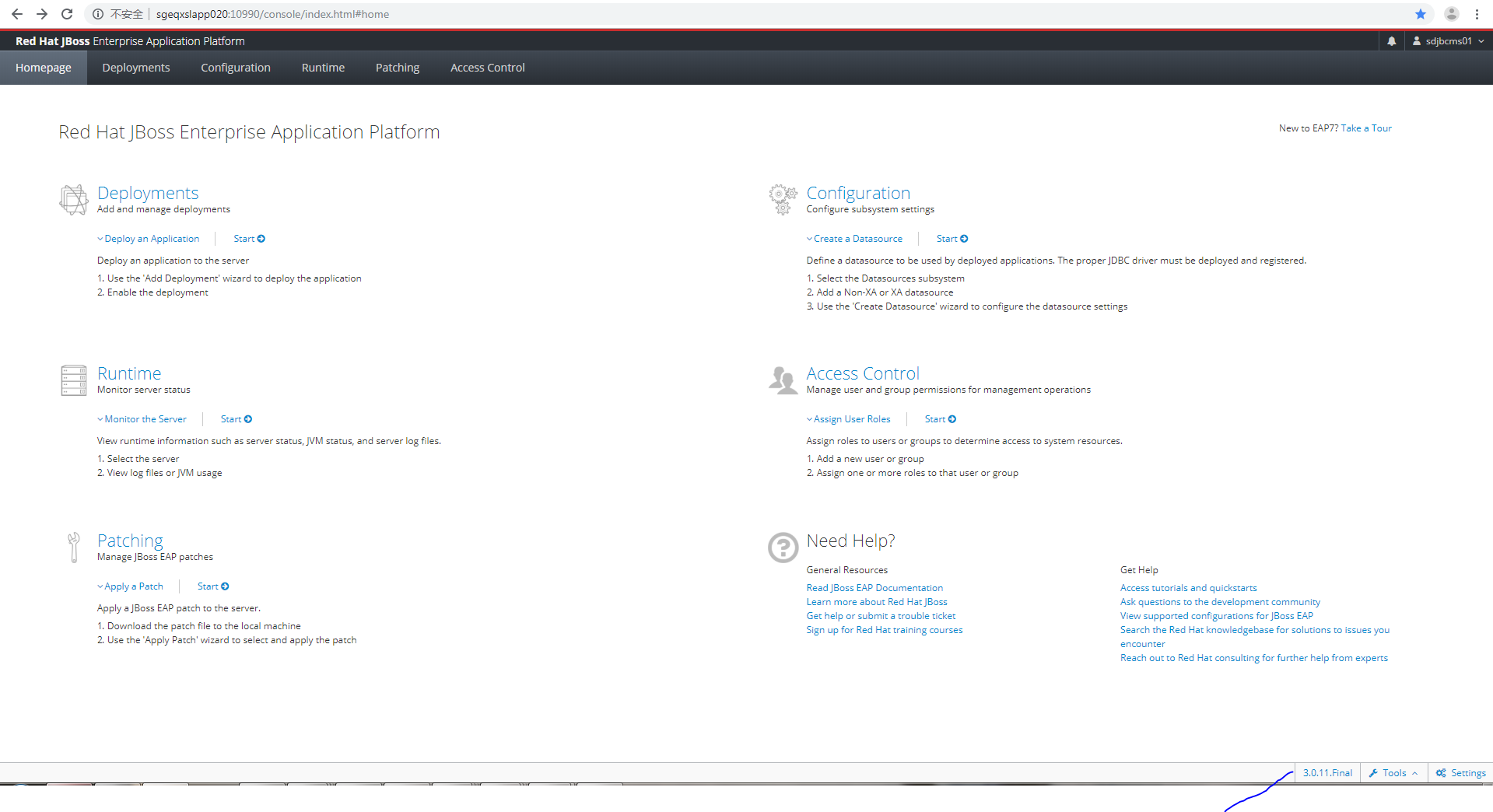
## [Java EE 8规范](https://blog.csdn.net/weixin_34362790/article/details/%5C)

## 学习文档的地址

<https://access.redhat.com/documentation/en-us/red_hat_jboss_enterprise_application_platform/7.2/>

## 查看版本

点击下列按钮



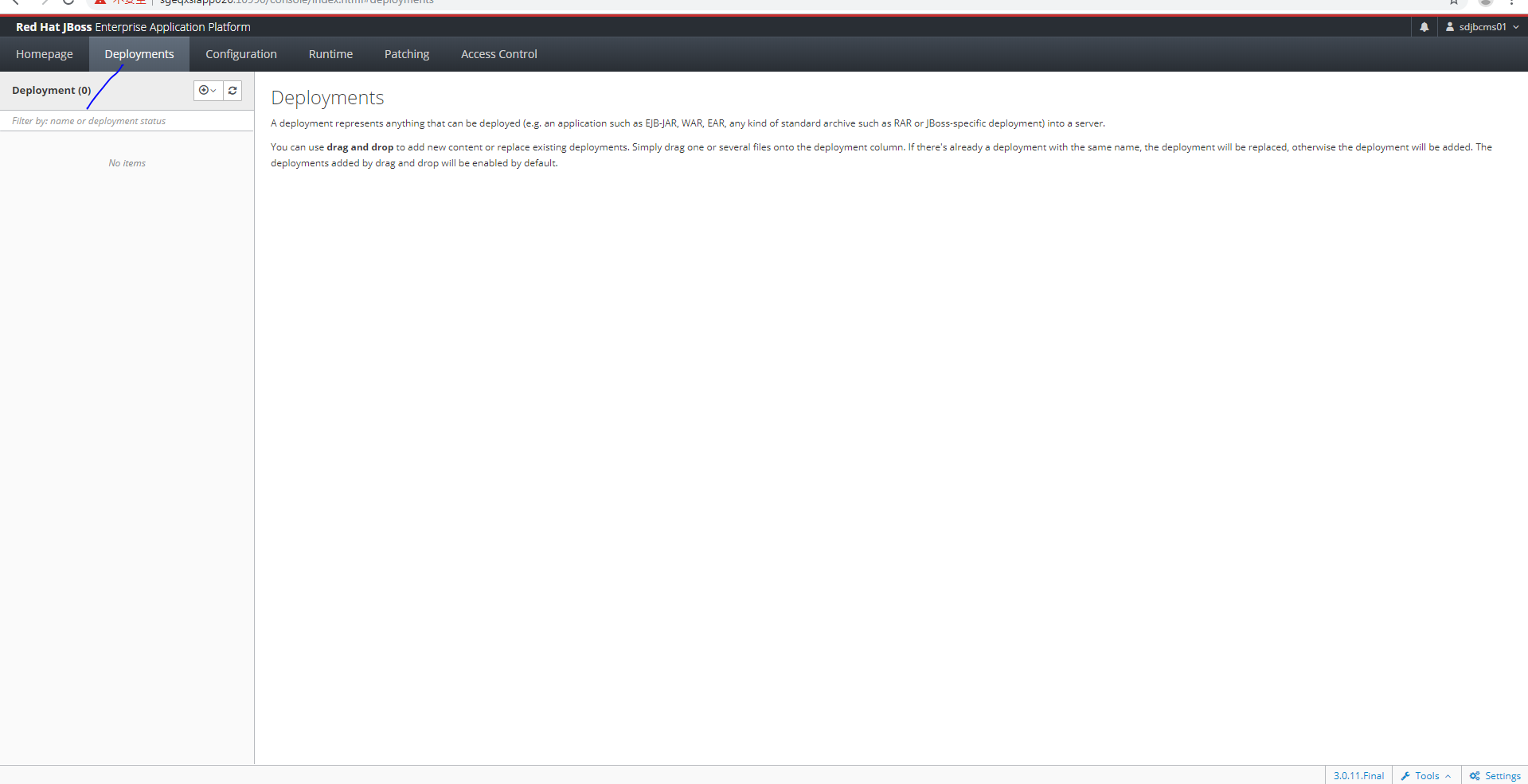


## 部署

部署表示可以部署的任何内容（例如，诸如EJB-JAR，WAR，EAR之类的应用程序，诸如RAR或JBoss特定部署的任何类型的标准存档）到服务器中。

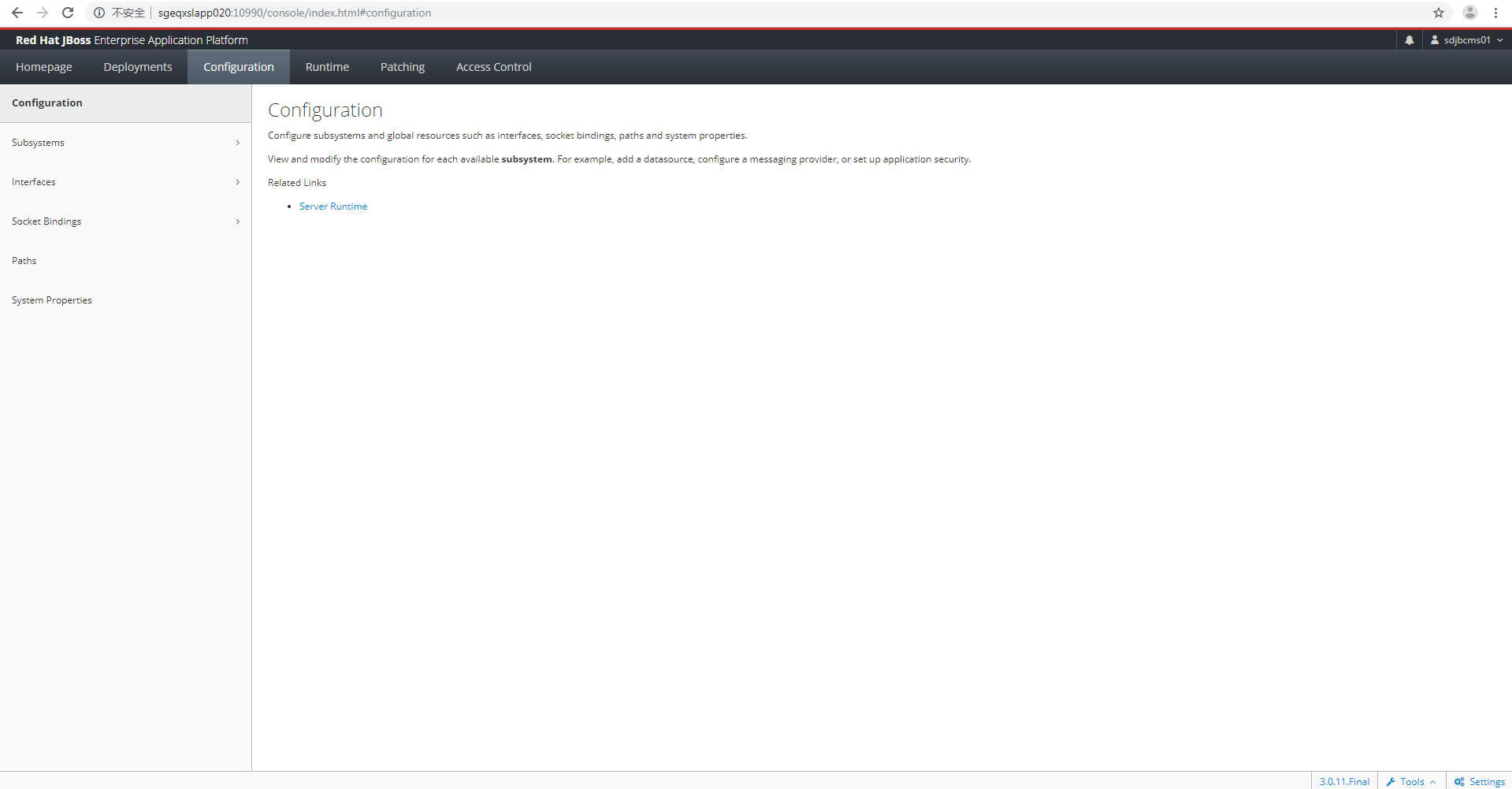
您可以使用拖放来添加新内容或替换现有部署，只需将一个或多个文件拖到部署列上即可。如果已经有相同名称的部署，则将替换部署，否则将添加部署。 默认情况下将启用通过拖放添加的部署。

操作页面如下：



## 配置

配置子系统和全局资源，例如接口，套接字绑定，路径和系统属性。查看和修改每个可用子系统的配置。例如添加数据源，配置消息传递提供程序或设置应用程序安全性。



### 配置JDBC驱动程序

在应用程序可以连接到数据源之前，需要安装数据源供应商的JDBC驱动程序，您可以选择两种不同的方式来安装JDBC驱动程序。

1、模块

要将JDBC驱动程序作为模块安装，需要在EAP\_HOME / modules下创建文件路径结构，将JDBC驱动程序JAR复制到main /子目录中并创建module.xml文件。

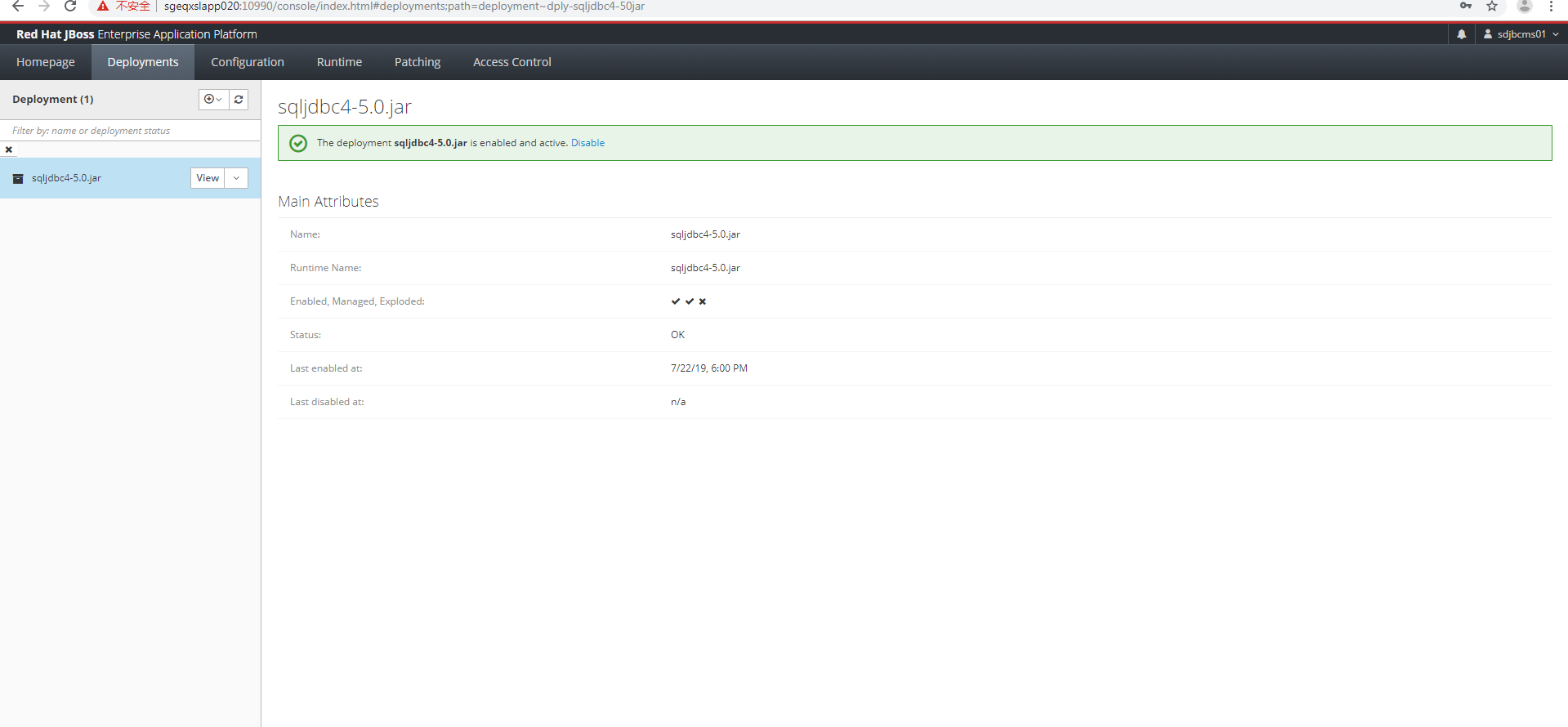
JDBC驱动程序作为模块可用后，您可以使用此部分添加，修改和删除驱动程序配置。

2、部署

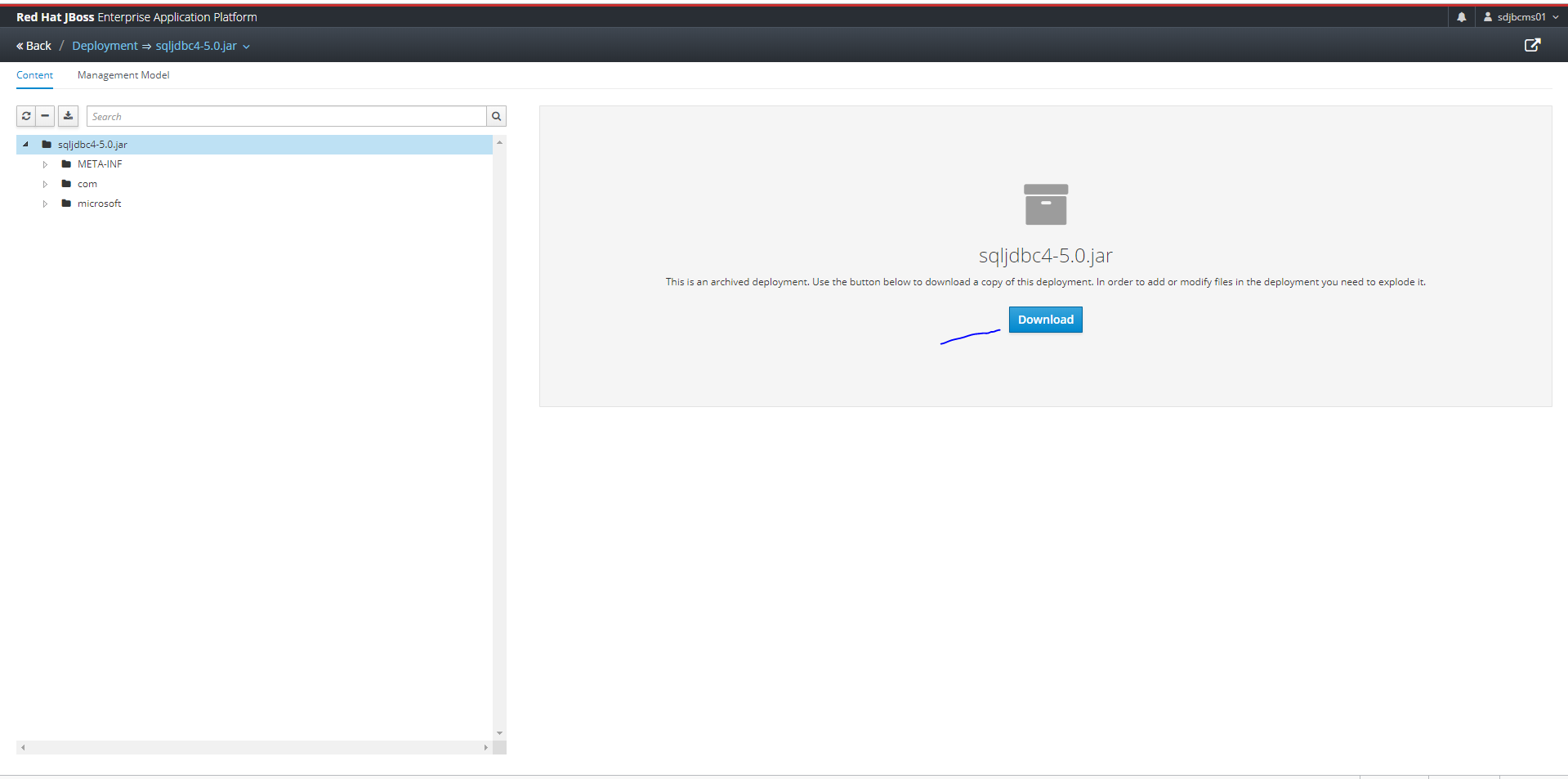
您可以像任何其他部署一样部署JDBC驱动程序。这意味着如果使用托管域，则可以在服务器组中的多个服务器上部署它们。任何兼容JDBC 4的驱动程序都将自动识别并按名称和版本安装到系统中。

在域模式下，作为应用程序部署的驱动程序只有在运行的服务器与所选配置文件匹配时才会显示在此部分中。

如下图所示：

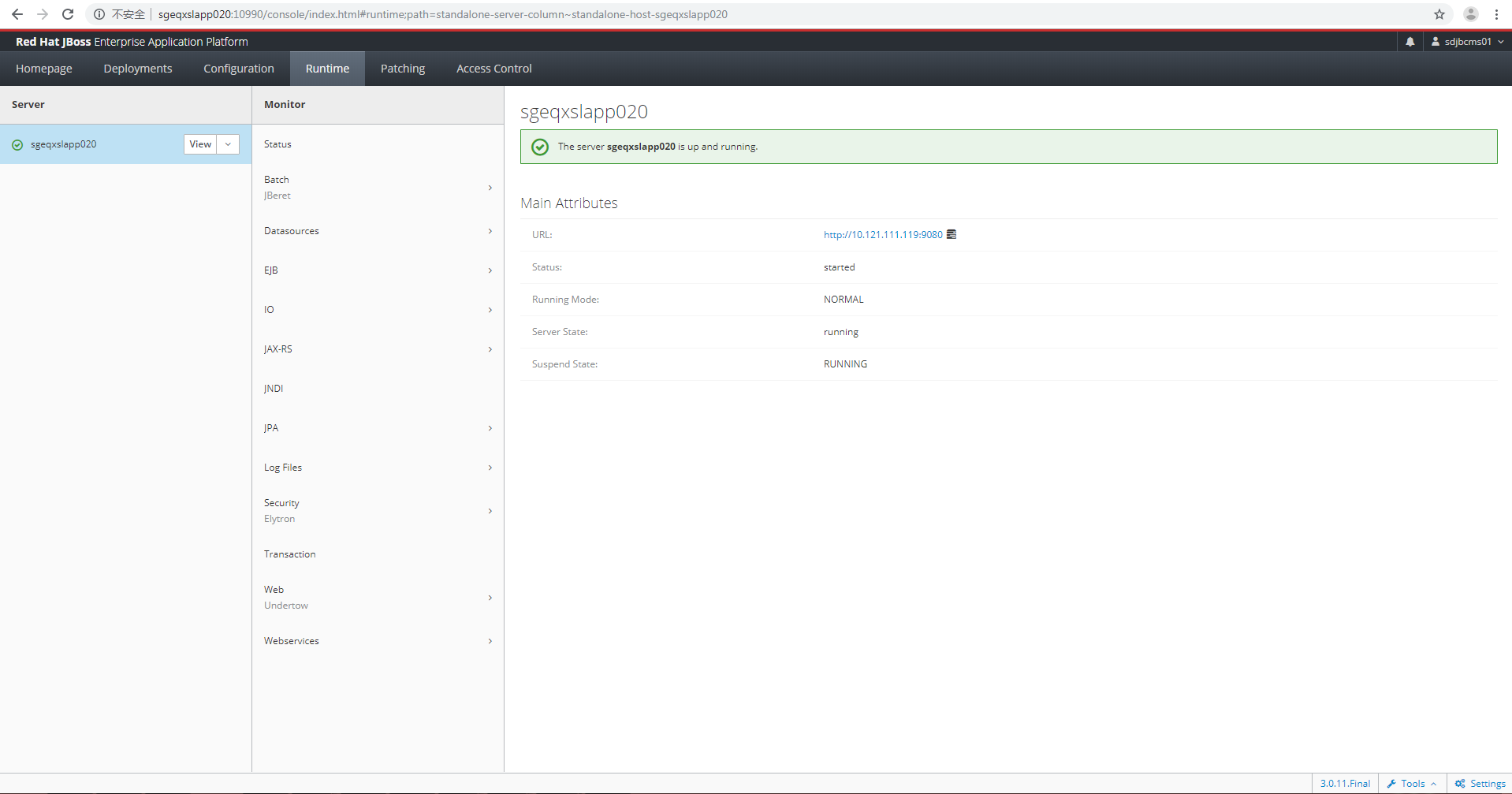


我们可以对部署的jar包进行下载，如下图所示：



## 运行环境

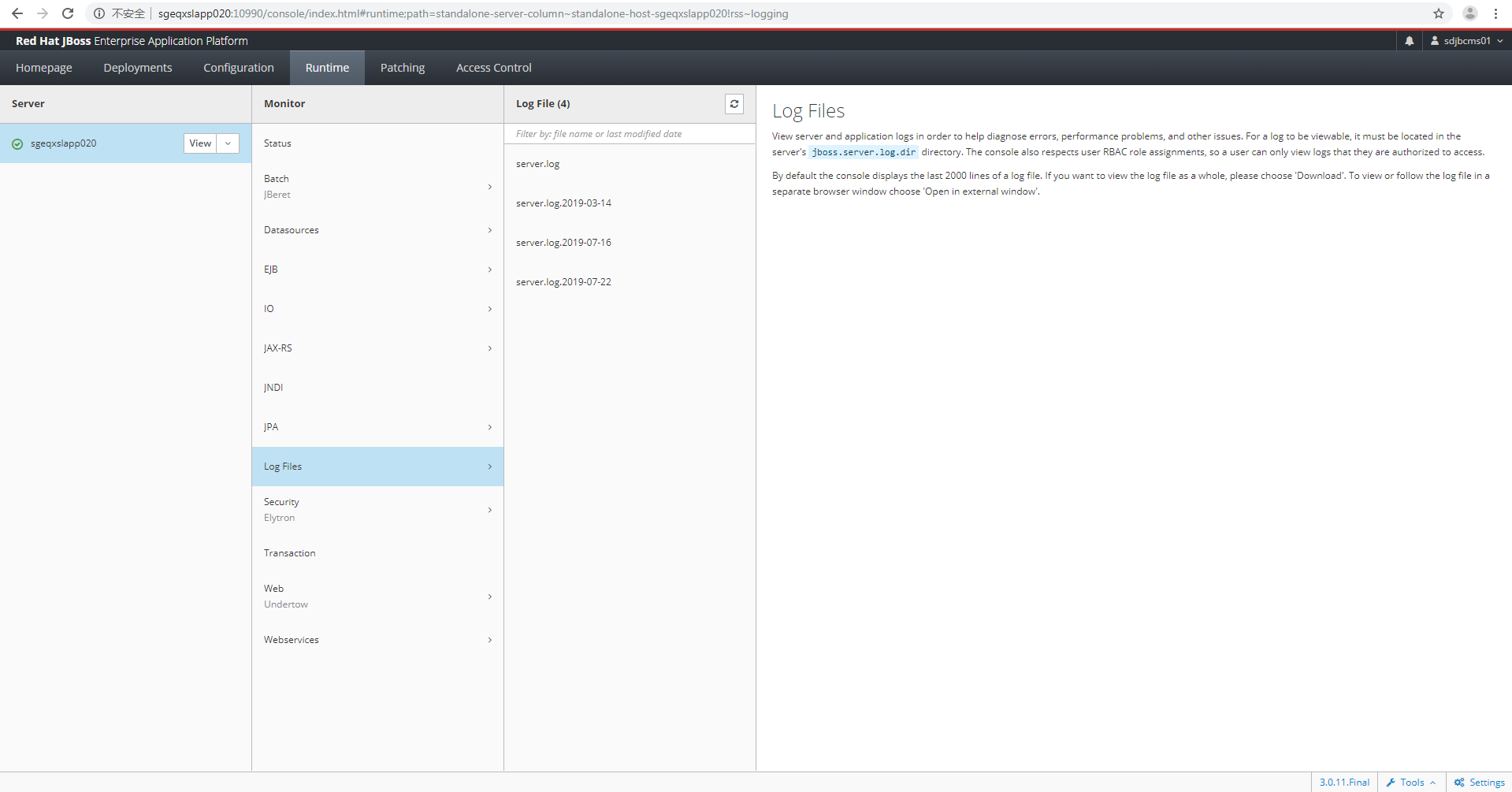
查看和监视运行时服务，如日志文件，JVM指标和子系统特定的运行时数据。



### 查看日志文件

查看服务器和应用程序日志，以帮助诊断错误，性能问题和其他问题。 要使日志可见，它必须位于服务器的jboss.server.log.dir目录中。 控制台还会考虑用户RBAC角色分配，因此用户只能查看他们有权访问的日志。

默认情况下，控制台显示日志文件的最后2000行。 如果要整体查看日志文件，请选择“下载”。 要在单独的浏览器窗口中查看或关注日志文件，请选择“在外部窗口中打开”。如下图所示：



### JNDI

提供本地JNDI名称空间的概述。 Java EE平台规范定义了以下JNDI上下文：

（1）java:comp命名空间的范围限定为当前组件（即EJB）

（2）java:module 适应当前模块

（3）java:app适应当前的应用程序

（4）java:global范围到应用程序服务器

除了标准名称空间，JBoss EAP还提供以下两个全局名称空间：

（1）java:jboss

（2）java:/

请注意，只能通过远程JNDI访问java:jboss/exported上下文中的条目。 对于Web部署，java:comp别名为java:module，因此在war中部署的EJB没有自己的comp命名空间。

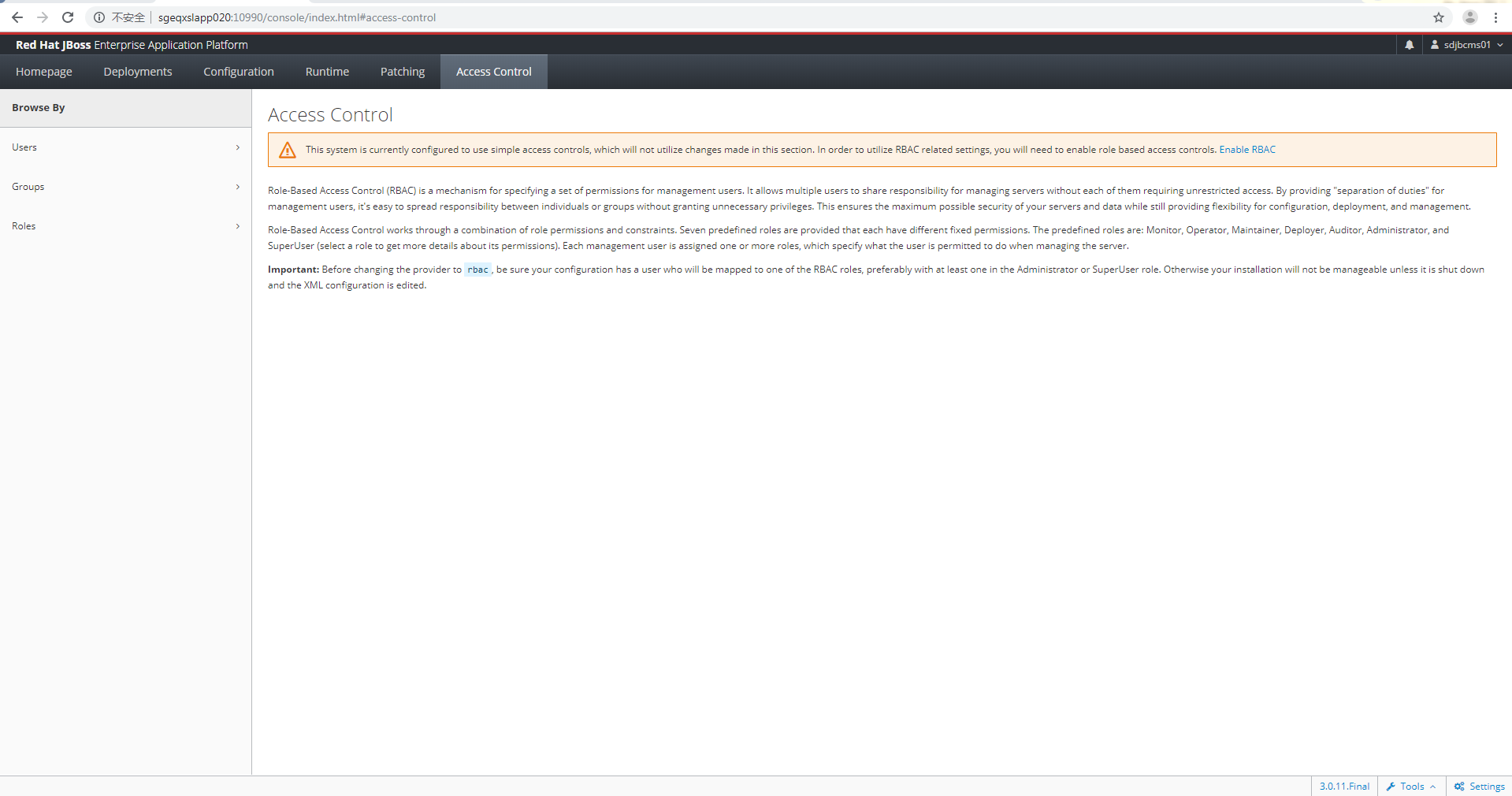
## 访问控制

此系统当前配置为使用简单的访问控制，不会使用本节中所做的更改。 要使用RBAC相关设置，您需要启用基于角色的访问控制。

基于角色的访问控制（RBAC）是一种为管理用户指定一组权限的机制。 它允许多个用户共享管理服务器的责任，而不需要每个用户都需要不受限制的访问。 通过为管理用户提供“职责分离”，可以轻松地在个人或团体之间分配责任，而无需授予不必要的权限。 这可确保服务器和数据的最大可能安全性，同时仍提供配置，部署和管理的灵活性。

基于角色的访问控制通过角色权限和约束的组合来工作。 提供七个预定义角色，每个角色具有不同的固定权限。 预定义角色包括：Monitor，Operator，Maintainer，Deployer，Auditor，Administrator和SuperUser（选择角色以获取有关其权限的更多详细信息）。 为每个管理用户分配一个或多个角色，这些角色指定管理服务器时允许用户执行的操作。

要点：在将提供程序更改为rbac之前，请确保您的配置具有将映射到其中一个RBAC角色的用户，最好至少具有一个Administrator或SuperUser角色。 否则，除非关闭并且编辑了XML配置，否则您的安装将无法管理。



# 启动与关闭JBOSS应用服务器

1、启动JBOSS应用服务器

双击run.bat文件或者

打开命令行窗口，进入 %JBOSS\_HOME%\bin，键入 run ,回车

2、关闭JBOSS应用服务器

直接关闭命令行窗口就可以了

# 修改jboss端口（jboss5.0版本）

根据 jboss-5.1.0.GA\server\default\deploy\jbossweb.sar\server.xml 找到对应的”server.xml“文件，打开此文件，找到端口号，为了避免与其他server冲突，这里把端口号改为”8888“



保存之后，重新启动”jboss-5.1.0.GA\bin\run.bat“文件，启动完成之后，在浏览器中输入”http://localhost:8888/“如果看到Jboss的欢迎界面，说明Jboss环境变量配置完成。

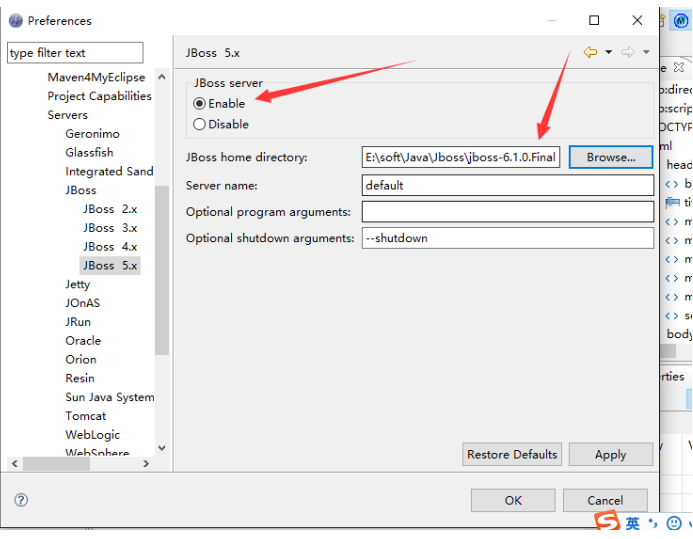
说明：不能的jboss版本可能这个文件的路径不一样。jboss4.x版本中是修改jboss-web.deployer/server.xml文件

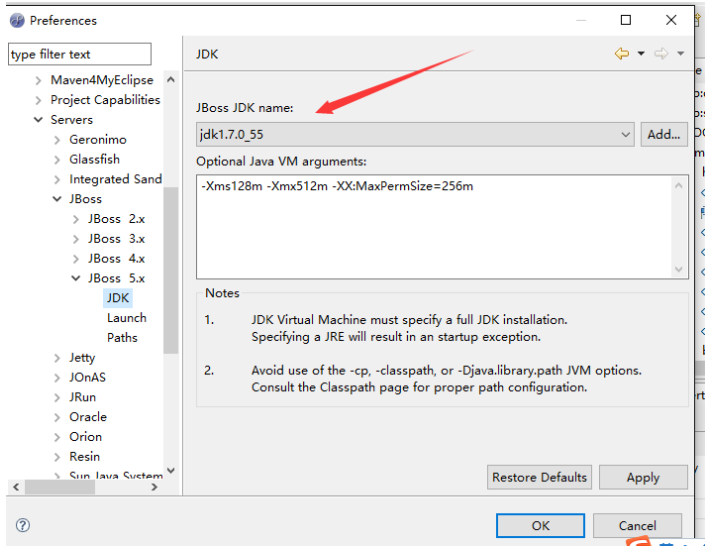
# Jboss在eclipse中的配置

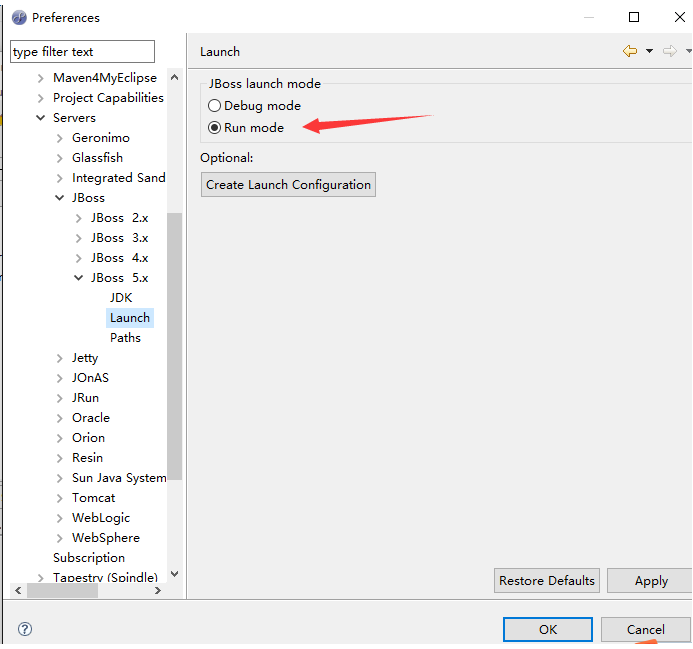
## 配置Jboss环境变量

在这里就不说了

## 配置eclipse







## jboss端口修改

1、进入 jboss-6.1.0.Final\server\default\deploy\jbossweb.sar

2、编辑 server.xml

3、搜索 “8080″ ，格式如下:



address默认为127.0.0.1，局域网不能访问，或指定-b 0.0.0.0，或在这里修改为0.0.0.0就能访问了。

# Jboss配置多个实例

## 基本概念

默认情况下，jboss的端口定义在内置Tomcat的server.xml里面。如果开启多个实例，那么就要依次更新这几个文件的端口，相当麻烦。jboss提供了一个很方便的mbean来统一管理多实例配置 --ServiceBindingManager。

基本思想：

jboss提供一个服务器端口绑定文件sample-bindings.xml，里面定义了多个以server-name为标志的端口，它就作为统一的端口绑定配置文件。然后在每个服务器实例的jboss-service.xml中注册一个端口绑定服务ServiceBindingManager,该服务用参数

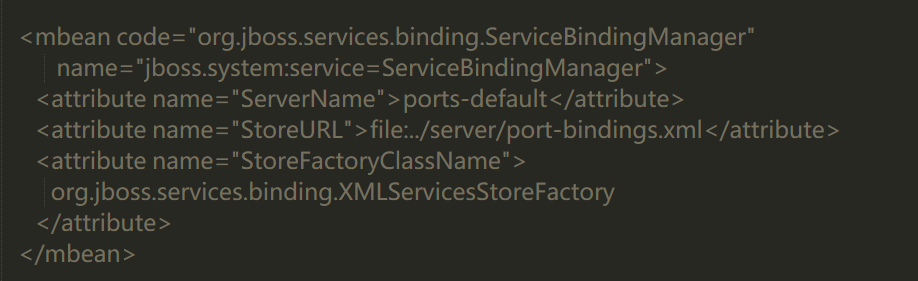
ServerName：指定xxx-bindings.xml对应的端口配置

StoreURL： 指定xx-bindings.xml文件的位置

一旦该服务注册，则Tomcat下server.xml的端口设置被忽略，而启用新的端口配置。

绑定服务的示例配置，它定义在同一台计算机上并行运行多个JBoss实例的不同端口配置（ports-default，ports-01，ports-02）。可以通过ServiceBindingManager属性ServerName在jboss-service.xml文件中选择实际端口配置。

以下样本，例如选择jboss-default端口配置



要运行第二个服务器实例，您必须通过在第二个服务器的jboss-service.xml中指定备用端口绑定配置来更改该实例的端口绑定，例如：

<attribute name="ServerName">ports-01</attribute>

## 操作步骤

1、打开default中conf文件夹jboss-service.xml，搜索“service binding”,这一段代码是注释掉的，把注释取消。

2、现在只需要将server目录中的default复制一份，粘贴到server目录中，重命名为test，将jboss-service.xml中的<attribute name="ServerName">ports-01</attribute> 中的ports-01改为sample-binding.xml中的server的名称（ports-02 or ports-03），这样就可以了。

3、启动各个服务实例。run.bat -c xxx。假如你有2个实例default, test，则运行：   
run.bat -c default   
run.bat -c test

## 注意事项

1、上述配置的前提是每个实例的server.xml中HTTP协议的端口配置均为默认的8080，若该端口修改过，则需要修改sample-bindings.xml中port-default中的默认8080端口为server.xml中HTTP协议端口。

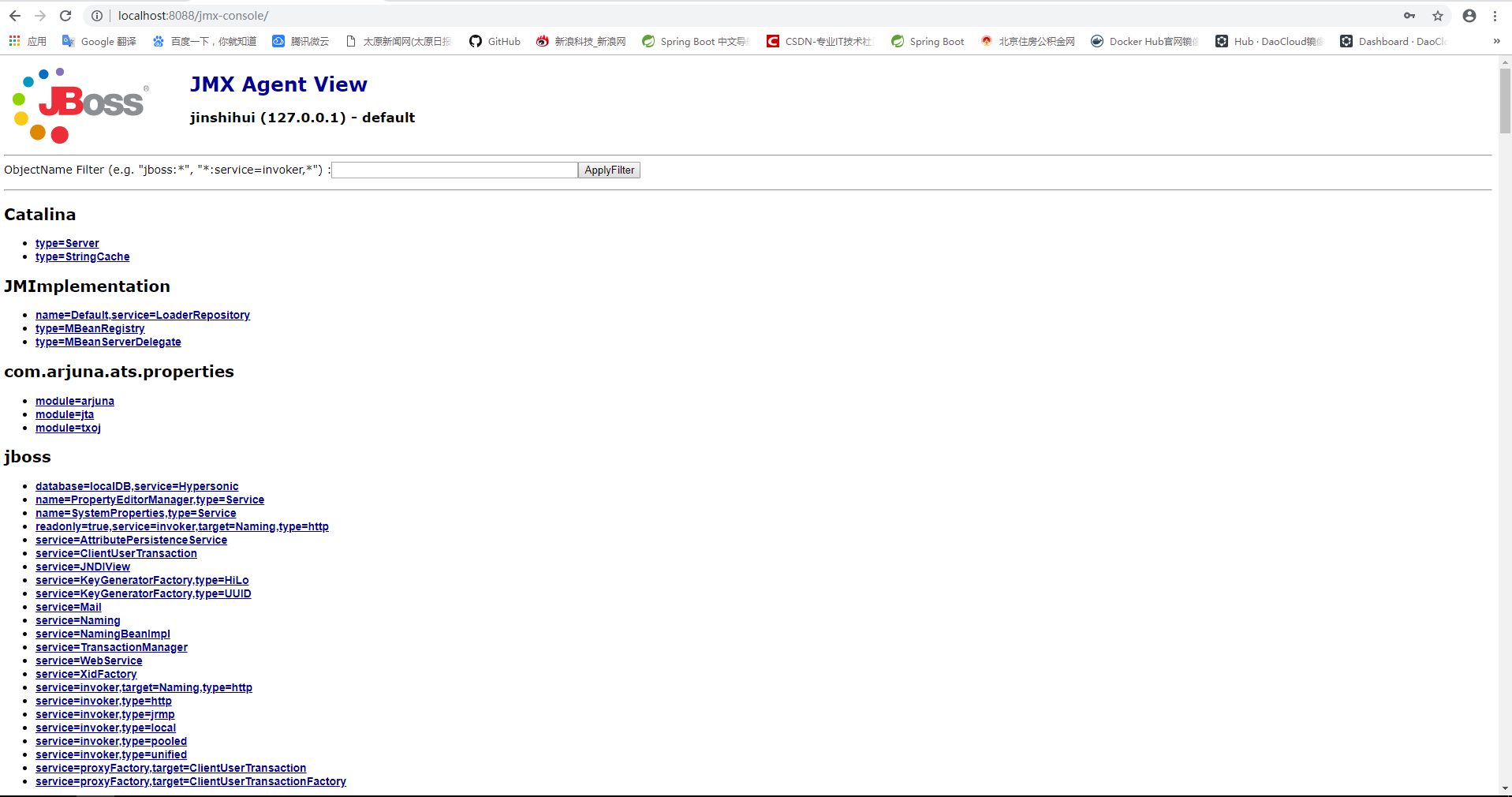
2、上述方法也同样适用于同一台机器上面部署多个JBoss的情况而不是同一个JBoss下多个配置。

3、一般情况下默认的4组端口足够使用，若仍不够，可自行添加其他port组。

4、若可能，仍是推荐不同的实例绑定到不同的IP地址上。

# Jboss JMX控制台

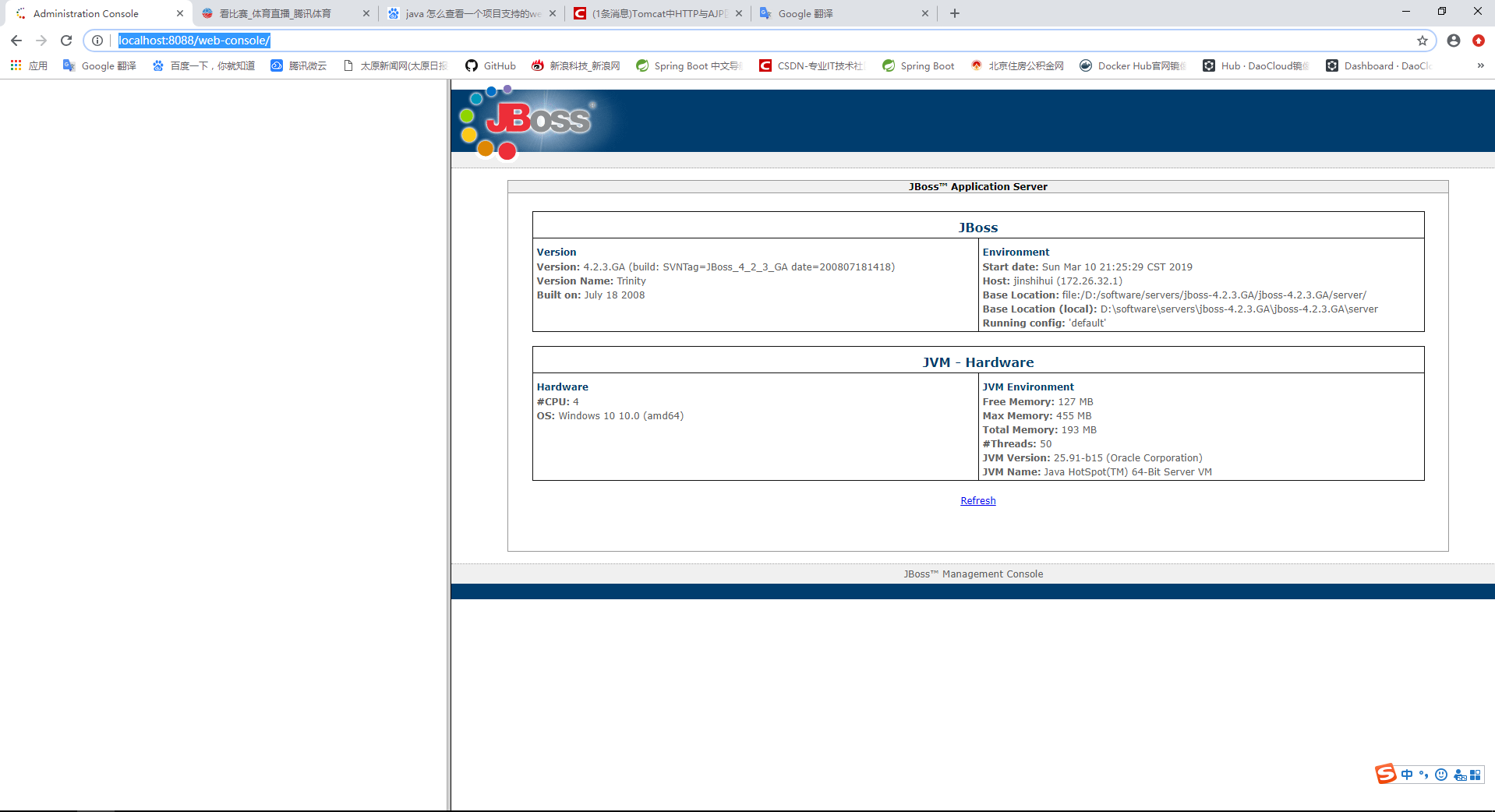
地址：http://localhost:8088/jmx-console/



# Jboss web控制台地址

这是我本地的地址

http://localhost:8088/web-console/



# 给控制台登录窗口设置用户名和密码

设置用户名和密码才能进入控制台地址。

步骤：

1、进入jmx-console.war，通常此目录在{jboss\_home}\server\default\deploy目录下。

2、编辑该目录下的WEB-INF\web.xml，把<security-constraint>部分注释去掉 。

3、 编辑该目录下的WEB-INF\classes的roles.properties和users.properties；

users.properties文件里定义用户名和密码，格式为username=password。当然，可以依照此格式添加多个用户。

roles.propertie文件里定义用户名和角色，格式为username=role。这里的role必须和web.xml文件里的<role-name>相对应。

4、编辑WEB-INF\jboss-web.xml文件，打开<security-domain>部分注释；<security-domain>内容对应的security-domain对应的是{jboss\_home}\server\default\conf\login-config.xml这个jaas配置文件里的某个application-policy。   
如:

<security-domain>java:/jaas/jmx-console</security-domain>

在login-config.xml里对应有：

<application-policy name = "jmx-console">   
             <authentication>   
                   <login-module code="org.jboss.security.auth.spi.UsersRolesLoginModule"   
                           flag = "required" />   
             </authentication>   
</application-policy>

5、重启jboss，就会出现你想要的输入username和password的窗口。

# Jboss部署项目

跟apache一样，jboss也有一个部署目录，那就是jboss/server目录，其中有三个目录：all，default，minimal。代表了jboss提供的三种部署方式，all表示jboss提供的服务全部打开，default表示默认的 jboss服务，minimal表示只打开最基本的，这里面可以增加自己的部署，我们只使用default。

要部署我们的应用，还需要做以下几步：

1、把jboss/docs/examples/jca/mysql-ds.xml拷贝到jboss/server/default/deploy目录下，并做如下修改：  
jdbc:mysql://你要连接的数据库的ip:3306/数据库名?zeroDateTimeBehavior=convertToNull  
com.mysql.jdbc.Driver  
用户名

密码

2、下载jdbc驱动，拷贝到部署目录的lib中(server/default/lib)

3、接下来就是把项目部署到jboss服务器下面就可以了。

JBoss中的部署过程非常的简单、直接并且支持热部署。在每一个配置中，JBoss不断的扫描一个特殊的目录的变化：$JBOSS\_HOME/server/config-name/deploy

你可以把下列文件拷贝到此目录下：   
       任何jar库（其中的类将被自动添加到JBoss的classpath中）   
       EJB JAR

WAR (Web Appliction Archive) 注意默认情况下context为war名称.

EAR (Enterprise Application aRchive)   
       包含JBoss MBean定义的XML文件   
       一个包含EJB JAR、WAR或者EAR的解压缩内容，并以.jar、.war或者.ear结尾的目录。

说明：JBoss支持热部署，也就是war文件部署到服务器上后不需要重新启动JBoss（Tomcat不支这种特性）。war文件的部署很简单，直接将war文件拷贝到JBoss\server\default\deploy目录下即可。

# 虚拟目录配置

编辑

%jboss-4.0.2%\server\default\deploy\jbossweb-tomcat55.sar\server.xml 在</Host>之前增加：   
<Context path="/test" docBase="F:\\test" debug="1" reloadable="true" override="true" crossContext="true"/>   
其中path 为虚拟目录名，docBase为硬盘路径（可为绝对路径）reloadable为类变更时是否重新加载。

# 遇到的错误

## 1、[spring mvc:Error scanning entry module-info.class from jar错误](https://www.cnblogs.com/liaochong/p/moduleinfo.html)

解决方法：

经排查，是引入的包版本不对，slf4j-api-1.8.0-alpha0.jar必须要jdk9才能支持（module-模块化是9的特性），所以删除这个无用的包就可以了。