

类别 研发文档

编号 QXSK-D-RAP

版本 4.2

密级 公开

全息时空运营服务平台 运行环境的安装与配置手册

编制单位: 江苏凌比特微电子有限公司

编 制 人: 任 珊 虹 马 琛

编制日期: 2018 年 4 月 20 日

修订记录

| 时间 | 作者 | 版本 | 说明 |
|--------------|-----|------|--|
| 2015. 11. 06 | 任珊虹 | 0. 1 | 初始版本 |
| 2015. 11. 11 | 任珊虹 | 0. 2 | 补充串口驱动部分 |
| 2016. 02. 22 | 任珊虹 | 0. 8 | 添加 CentOS 部分 |
| 2016. 02. 29 | 任珊虹 | 0. 9 | 修正错误；添加问题描述及其解决办法 |
| 2016. 03. 05 | 马琛 | 1. 0 | 详细描述 linux 部署每一步的步骤，确保操作成功 |
| 2016. 08. 01 | 任珊虹 | 2. 0 | 添加 Active MQ 的内容 |
| 2017. 10. 26 | 任珊虹 | 3. 0 | 添加 Mysql 复制库的内容。 删除 Active MQ 的内容（已内嵌，应用发布时自动配置）。 删除 GlassFish 远程管理的内容（不建议远程管理）。 |
| 2017. 11. 06 | 任珊虹 | 3. 1 | 完善 Mysql 库配置的内容。 |
| 2017. 11. 08 | 任珊虹 | 3. 2 | 完善 Mysql 库配置的内容。 |
| 2017. 11. 17 | 任珊虹 | 3. 3 | 添加 EclipseLink 补丁的内容。 |
| 2018. 4. 4 | 任珊虹 | 4. 0 | 添加从库处理事件的步骤。并入安全管理的内容 |
| 2018. 4. 8 | 任珊虹 | 4. 1 | 补充信息 |
| 2018. 4. 20 | 任珊虹 | 4. 2 | 调整章节。添加：SSL 配置、网站管理 |
| | | | |

对于文档中的错误或者描述不清的地方，请联系作者，谢谢！

目 录

| | |
|--------------------------------|----|
| 1 在 Windows 上安装配置..... | 5 |
| 1.1Java 开发工具：JDK..... | 5 |
| 1.1.1 资源地址..... | 5 |
| 1.1.2 安装..... | 5 |
| 1.1.3 设置环境变量..... | 5 |
| 1.1.4 验证安装..... | 5 |
| 1.2java 串口驱动..... | 6 |
| 1.2.1 资源地址..... | 6 |
| 1.2.2 安装..... | 6 |
| 1.3 数据库系统：MySQL..... | 7 |
| 1.3.1 资源地址..... | 7 |
| 1.3.2 安装..... | 7 |
| 1.3.3 设置环境变量..... | 7 |
| 1.3.4 配置主库..... | 7 |
| 1.3.5 导入平台数据库..... | 8 |
| 1.3.6 建立复制库..... | 9 |
| 1.3.7 建立历史库..... | 10 |
| 1.4 Java EE 服务器：GlassFish..... | 11 |
| 1.4.1 资源地址..... | 11 |
| 1.4.2 安装..... | 11 |
| 1.4.3 设置环境变量..... | 11 |
| 1.4.4 启动 GlassFish..... | 11 |
| 1.4.5 管理 GlassFish..... | 12 |
| 1.4.6 停止 GlassFish..... | 13 |
| 1.4.7 添加 JDBC 驱动..... | 13 |
| 1.4.8 配置应用资源..... | 13 |
| 2 在 Linux 上安装配置..... | 14 |
| 2.1Java 开发工具：JDK..... | 14 |
| 2.1.1 资源地址..... | 14 |
| 2.1.2 安装..... | 14 |
| 2.1.3 设置环境变量..... | 14 |
| 2.1.4 验证安装..... | 15 |
| 2.2java 串口驱动..... | 15 |
| 2.2.1 资源地址..... | 15 |
| 2.2.2 安装..... | 15 |
| 2.3 数据库系统：MySQL..... | 16 |
| 2.3.1 资源地址..... | 16 |
| 2.3.2 清理环境..... | 16 |
| 2.3.3 安装..... | 16 |
| 2.3.4 设置账户安全..... | 16 |
| 2.3.5 设置字符集..... | 17 |
| 2.3.6 验证安装..... | 17 |
| 2.3.7 导入平台数据库..... | 17 |
| 2.3.8 创建复制库和历史库..... | 17 |
| 2.3.9 修改配置文件..... | 18 |
| 2.3.10 创建复制账号..... | 20 |
| 2.3.11 配置复制库..... | 20 |
| 2.3.12 建立历史库..... | 21 |
| 2.3.13 备份平台数据库..... | 22 |

| | |
|--|----|
| 2.4 Java EE 服务器: GlassFish..... | 23 |
| 2.4.1 资源地址..... | 23 |
| 2.4.2 安装..... | 23 |
| 2.4.3 设置环境变量..... | 23 |
| 2.4.4 创建域..... | 23 |
| 2.4.5 启动 GlassFish..... | 24 |
| 2.4.6 管理 GlassFish..... | 24 |
| 2.4.7 停止 GlassFish..... | 24 |
| 2.4.8 添加 JDBC 驱动..... | 25 |
| 2.4.9 配置应用资源..... | 25 |
| 3 发布平台应用..... | 26 |
| 3.1 发布应用包..... | 26 |
| 3.2 平台应用的短地址..... | 28 |
| 3.3 部署应用时可能遇到的问题及其解决办法..... | 28 |
| 3.3.1 “NoClassFound”..... | 28 |
| 3.3.2 “Exception starting EMBEDDED broker=Cannot create broker instance”..... | 28 |
| 3.3.3 “Error in linking security policy for QXSKweb -- Inconsistent Module State”..... | 29 |
| 3.4 GlassFish4.1.*的 EclipseLink 补丁..... | 29 |
| 3.5 调整 JVM..... | 29 |
| 4 配置 SSL..... | 30 |
| 4.1 申请证书..... | 30 |
| 4.2 导入证书..... | 31 |
| 4.3 配置 GlassFish..... | 34 |
| 5 管理 www.qxsk9.com 网站..... | 35 |
| 6 CentOS 服务器的安全防护措施..... | 36 |
| 6.1 防火墙 firewalld..... | 36 |
| 6.2 只开放必要的端口和服务..... | 37 |
| 6.3 限制 SSH 的使用..... | 37 |
| 6.3.1 禁止 root 通过 ssh 登录..... | 38 |
| 6.3.2 设置登录白名单..... | 38 |
| 6.3.3 限制登录失败次数..... | 38 |
| 6.4 安装配置 Fail2Ban..... | 38 |
| 6.5 注释掉不必要的账户..... | 39 |
| 6.6 使用 webmin..... | 39 |
| 6.7 禁止 root 运行应用程序..... | 41 |
| 6.8 数据库的安全设置..... | 41 |
| 6.8.1 从官网下载安装..... | 41 |
| 6.8.2 按官网步骤..... | 41 |
| 6.8.3 限制系统账户 mysql..... | 41 |
| 6.8.4 修改 mysql 的 root 密码..... | 41 |
| 6.8.5 清除匿名账户..... | 41 |
| 6.8.6 删除测试数据库..... | 41 |
| 6.9 GlassFish 的安全设置..... | 42 |

1 在 Windows 上安装配置

1.1 Java 开发工具：JDK

1.1.1 资源地址

JDK 的官方地址为：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

云盘地址：（或者更高版本）

软件\java\jre_8u45_windows_x64.exe

1.1.2 安装

直接解包运行。当 java 自动更新程序要求后台运行或者开机子启动时，点击允许。
保持 java 版本自动更新。

1.1.3 设置环境变量

添加以下系统环境变量（替换为实际安装目录）

`JAVA_HOME=C:\Program Files\Java\jre1.8.0_45`

`CLASSPATH=.;C:\Program Files\Java\jre1.8.0_45\lib`

注意：别拉了前面的“.”

1.1.4 验证安装

打开命令窗口，运行“java -version”：

```
Microsoft Windows [版本 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\mara>java -version
java version "1.8.0_51"
Java(TM) SE Runtime Environment (build 1.8.0_51-b16)
Java HotSpot(TM) 64-Bit Server VM (build 25.51-b03, mixed mode)
```

1.2 java 串口驱动

1.2.1 资源地址

官网地址：

http://rxtx.qbang.org/wiki/index.php/Main_Page

64 位下载地址：

<http://fizzed.com/oss/rxtx-for-java>

云盘地址：（或者更高版本）

软件\rxtx\mfz-rxtx-2.2-20081207-win-x64.zip

1.2.2 安装

将文件 RXTXcomm.jar 放到 “JRE 目录\lib\ext\”下。

将文件 rxtxParallel.dll 和 rxtxSerial.dll 放到 “JRE 目录\bin”下。

注意：文件需要放在系统上实际生效的 JRE 目录下。如果系统上有多个 JDK 或者 JRE 目录，请检查确认哪个目录是系统实际生效的。一个保险的做法是把这几个文件分别放到系统上所有 JDK 和 JRE 目录下。

1.3 数据库系统：MySQL

1.3.1 资源地址

官网地址：

<http://dev.mysql.com/downloads/>

网盘地址：（或者更高版本）

软件\mysql\mysql-installer-community-5.6.25.0.msi

1.3.2 安装

安装 mysql，选择“开发者”类型就行了，不要定制，一路缺省就好了。注意不要将 mysql 安装在包含汉字的路径下。

安装过程中需要设置 root 的密码。

安装完毕，mysql 服务将自动启动，MySQL Notifier 也在系统托盘（右下角）开始工作。

1.3.3 设置环境变量

将以下路径加入系统环境变量“PATH”中（替换为实际安装目录）

[C:\Program Files\MySQL\MySQL Server 5.7\bin](#)

1.3.4 配置主库

MySQL 正常安装完毕以后会自动创建和配置好一个缺省的数据库实例，并自动设置为系统服务。我们把这个缺省实例作为主库。

执行以下步骤以配置主库：

1）在系统服务列表中右键 MySQL 服务名选择“停止”。或者以管理员账户运行以下命令（替换为实际服务名）：

```
net stop MySQL57
```

2）找到主库的数据目录，例如“[C:\ProgramData\MySQL\MySQL Server 5.7](#)”。

3）编辑主库数据目录下的配置文件“[my.ini](#)”，添加或者修改以下参数：

```
[client]
```

```
port=3306
```

```
[mysqld]
```

```
datadir=C:/ProgramData/MySQL/MySQL Server 5.7/Data
```

```
port=3306
```

```
server-id=1
```

```
character-set-server=utf8
```

```
collation-server=utf8_unicode_ci
```

```
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"
```

```
log-bin=mysql-bin
```

```
innodb_flush_log_at_trx_commit=1
```

```
sync_binlog=1
```

```
gtid_mode=on
```

```
enforce_gtid_consistency=on
```

```
binlog_format=ROW
```

```
binlog-do-db=beidou
```

```
expire_logs_days=10
```

`binlog_row_image=minimal`

`max_sp_recursion_depth=100`
`event_scheduler=1`

4) 在系统服务列表中右键 Mysql 服务名选择“启动”。或者以管理员账户运行以下命令（替换为实际服务名）：

`net start MySQL57`

5) 登入主库：

`mysql -u root -p`

6) 输入以下命令（xxxxxx 换成实际密码），以创建数据复制账户：

`CREATE USER 'repl'@'localhost' IDENTIFIED BY 'xxxxxx';`
`GRANT REPLICATION SLAVE ON *.* TO 'repl'@'localhost';`
`SHOW MASTER STATUS;`

以下为示意图：

```
C:\Users\mara>mysql -u root -p beidou
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.9-log MySQL Community Server (GPL)
Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'repl'@'localhost' IDENTIFIED BY 'xxxxxx';
Query OK, 0 rows affected (0.05 sec)

mysql> GRANT REPLICATION SLAVE ON *.* TO 'repl'@'localhost';
Query OK, 0 rows affected (0.05 sec)

mysql> SHOW MASTER STATUS;
+-----+-----+-----+-----+-----+
| File          | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000001 | 623 |      |      |      |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

1.3.5 导入平台数据库

按以下顺序导入表结构、存储过程、初始化数据、事件、触发器：

`mysql -u root -p"密码" < bd_schema_sp.sql`
`mysql -u root -p"密码" beidou < bd_adjust_schema.sql`
`mysql -u root -p"密码" beidou < bd_data.sql`
`mysql -u root -p"密码" beidou < bd_events.sql`
`mysql -u root -p"密码" beidou < bd_triggers.sql`

1.3.6 建立复制库

- 1) 停止主库的运行。
- 2) 将主库的数据目录整体复制成一个新的数据目录，如：“C:\ProgramData\MySQL\MySQL Server 5.7 – repl”。
- 3) 删除新数据目录下的“data\auto.cnf”文件。这个文件将由系统自动生成的，用以唯一标识数据库实例。
- 4) 编辑新目录下配置文件“my.ini”，删除或注释参数“log-bin”，然后添加或者修改以下参数：

```
[client]
port=3307

[mysqld]
datadir=C:/ProgramData/MySQL/MySQL Server 5.7 – repl\Data
port=3307
server-id=2

character-set-server=utf8
collation-server = utf8_unicode_ci
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"

gtid_mode=on
enforce_gtid_consistency=on
skip-slave-start=1

log-error="MARA-PC-repl.err"
max_sp_recursion_depth=100
event_scheduler=1
```

- 5) 以系统管理员账户执行以下命令，生成复制库的系统服务“MySQL57-repl”：
mysql --install MySQL57-repl --defaults-file="C:\ProgramData\MySQL\MySQL Server 5.7 – repl\my.ini"
- 6) 在系统服务列表中分别启动主库“MySQL57”和复制库“MySQL57-repl”。或者以管理员账户运行以下命令：
net start MySQL57
net start MySQL57-repl
- 7) 登入复制库：（由于是直接复制主库，因此 root 密码与主库相同）
mysql --port=3307 -u root -p
- 8) 输入以下命令，以建立与主库的数据复制关系：（xxxxx 换成前面建立的复制账户的密码）
change master to master_host='localhost', master_user='repl', master_password='xxxxx', master_auto_position=1;
- 9) 输入命令“show slave status \G”，并检查以下这行的值是否为 1：
Auto_Position: 1
- 10) 输入以下命令，以启动数据复制：
start slave;
- 11) 输入命令“show slave status \G”，并检查以下这些行的值（有的值可能随时间变化），以验证数据复制正在正确进行：

```
Slave_IO_State: Waiting for master to send event
Master_Host: localhost
Master_User: repl
Master_Port: 3306
Master_Log_File: mysql-bin.000004
Read_Master_Log_Pos: 486
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
Master_Server_Id: 1
Master_UUID: 6bd8673c-9bee-11e5-a939-1c872c722172
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
```

- 12) 可以随时输入命令“start slave;”或者“stop slave;”，以启停数据同步。

- 13) 观察 “`show slave status \G`” 的输出，当 “**Seconds_Behind_Master: 0**” 意味着数据已全部同步到复制库中。
- 14) 如果主库中定义了事件，则需要把从库里的相应的事件删除或者让它们失效。输入命令 “`show events\G`” 查看事件，并且对于每一个由主库复制而来的事件运行以下命令。这一步保证从库不会重复运行主库的事件：
`alter event e_add_partition_monthly ON COMPLETION PRESERVE DISABLE;`
- 15) 如果需要从库每次启动后自动开始同步，则在它的配置文件中修改为 “`skip-slave-start=0`”。

1.3.7 建立历史库

- 1) 停止主库的运行。
- 2) 将主库的数据目录整体复制成一个新的数据目录，如： “`C:\ProgramData\MySQL\MySQL Server 5.7 – his`”。
- 3) 删除新数据目录下的 “`data\auto.cnf`” 文件。这个文件将由系统自动生成的，用以唯一标识数据库实例。
- 4) 编辑新目录下配置文件 “`my.ini`”，删除或注释参数 “`log-bin`”，然后添加或者修改以下参数：


```
[client]
port=3308

[mysqld]
datadir=C:/ProgramData/MySQL/MySQL Server 5.7 – his\Data
port=3308
server-id=3

character-set-server=utf8
collation-server = utf8_unicode_ci
sql-mode="STRICT_TRANS_TABLES,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION"

gtid_mode=on
enforce_gtid_consistency=on
skip-slave-start=1

log-error="MARA-PC-his.err"
max_sp_recursion_depth=100
event_scheduler=1
```
- 5) 以系统管理员账户执行以下命令，生成复制库的系统服务 “`MySQL57-his`”：


```
mysqld --install MySQL57-his --defaults-file="C:\ProgramData\MySQL\MySQL Server 5.7 – his\my.ini"
```
- 6) 在系统服务列表中分别启动主库 “`MySQL57`” 和历史库 “`MySQL57-his`”。或者以管理员账户运行以下命令：


```
net start MySQL57
net start MySQL57-his
```
- 7) 登入历史库：（由于是直接复制主库，因此 `root` 密码与主库相同）


```
mysql --port=3308 -u root -p
```
- 8) 至 15) 与上一节复制库的步骤一样。

1.4 Java EE 服务器：GlassFish

1.4.1 资源地址

JDK 的官方地址为：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

云盘地址：（或者更高版本）

软件\java\glassFish\glassfish-4.1.zip

1.4.2 安装

在选定的目录下直接解包即可。

安装目录不要包含汉字。建议安装在目录“c:\glassFish-4.1\”下。

1.4.3 设置环境变量

将以下目录加入系统环境变量“PATH”中（替换为实际安装目录）：

C:\glassfish-4.1\glassfish4\bin

可以创建多个服务器实例。

在安装包里，缺省的域“domain1”已被安装和配置好了。

1.4.4 启动 GlassFish

使用命令“asadmin”来控制和管理工作 GlassFish（启动、停止、配置、发布应用等）。

要启动 GlassFish，运行命令：

asadmin start-domain

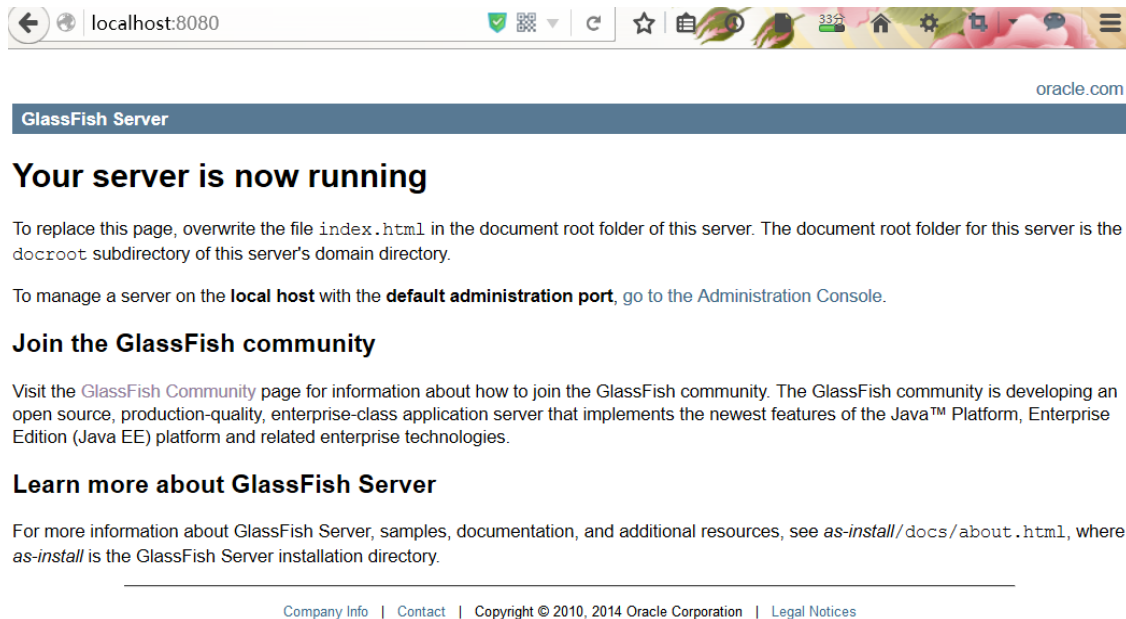
例如：

```
C:\Users\mara>cd C:\glassfish-4.1\glassfish4\bin
C:\glassfish-4.1\glassfish4\bin>asadmin start-domain
Waiting for domain1 to start .....
Successfully started the domain : domain1
domain Location: C:\glassfish-4.1\glassfish4\glassfish\domains\domain1
Log File: C:\glassfish-4.1\glassfish4\glassfish\domains\domain1\logs\server.log
Admin Port: 4848
Command start-domain executed successfully.
C:\glassfish-4.1\glassfish4\bin>
```

等几秒，GlassFish 将启动并准备接受请求。

缺省的域 “domain1”在端口 8080 监听，打开浏览器，输入以下地址来访问缺省的页面：

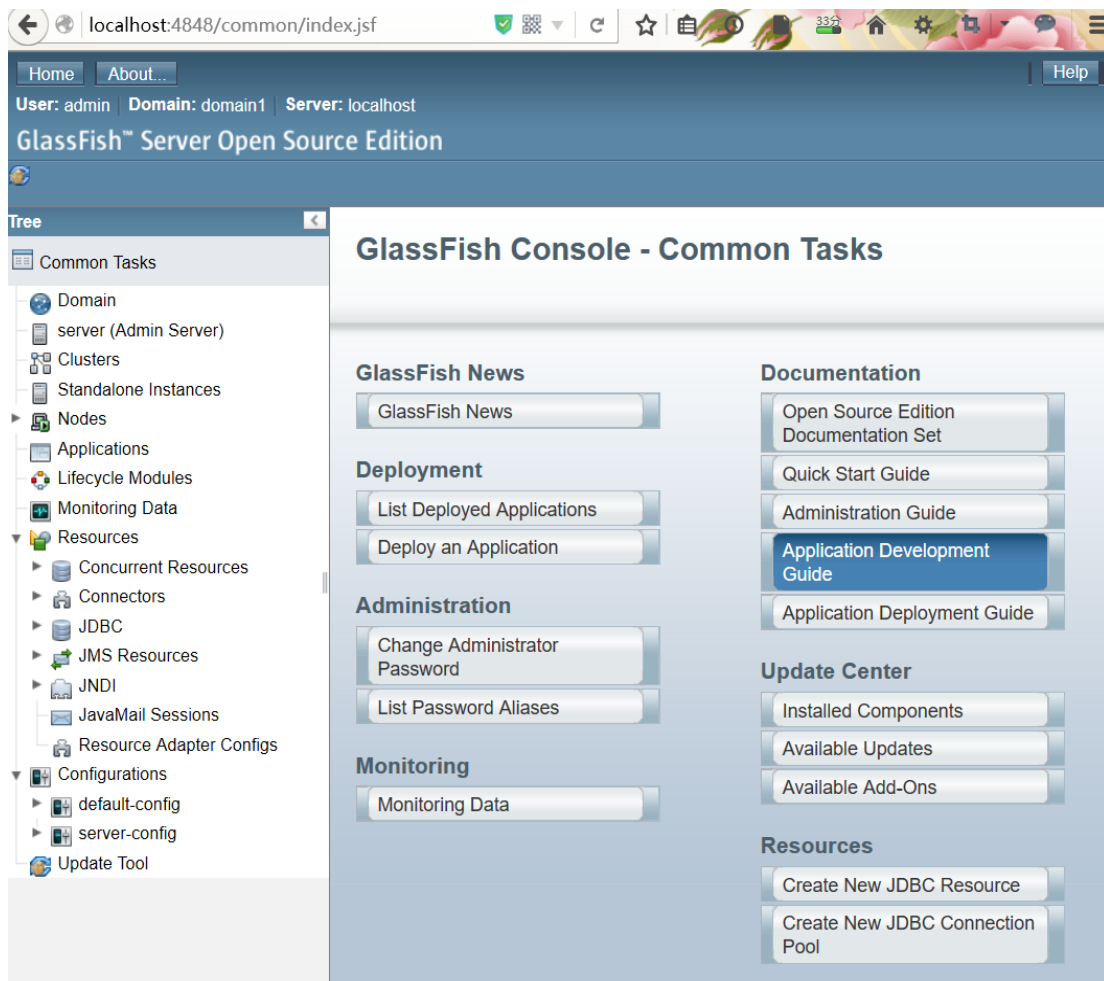
<http://localhost:8080>



1.4.5 管理 GlassFish

要管理 GlassFish，访问以下网址：

<http://localhost:4848>



1.4.6 停止 GlassFish

要停止 GlassFish，运行以下命令：

`asadmin stop-domain`

```
C:\glassfish-4.1\glassfish4\bin>asadmin stop-domain
Waiting for the domain to stop .
Command stop-domain executed successfully.
C:\glassfish-4.1\glassfish4\bin>
```

1.4.7 添加 JDBC 驱动

网盘地址：（或者更高版本）

软件\mysql\mysql-connector-java-5.1.35-bin.jar

把这个 jar 包放入以下目录：（替换成实际的 domain 目录）

`C:\glassfish-4.1\glassfish4\glassfish\domains\domain1\lib\ext\`

必须重启 GlassFish 才能使 JDBC 驱动生效。

1.4.8 配置应用资源

用文本编辑器打开文件 “sun-resources.xml”，替换 xxxxxx 为 mysql 账户 root 的实际密码：

`<property name="User" value="root"/>`

`<property name="Password" value="xxxxxx"/>`

然后运行以下命令导入应用资源：

`asadmin add-resources sun-resources.xml 的路径`

```
C:\glassfish-4.1\glassfish4\bin>asadmin add-resources D:\BeiDou\sun-resources.xml
JDBC connection pool beidouPool created successfully.
JDBC resource jdbc/beidou created successfully.
Mail Resource mail/QXSKSession created.
Administered object jms/BD/bytes/received created.
Administered object jms/BD/statement/received created.
Administered object jms/BD/statement/sending created.
Administered object jms/BD/location/new created.
Administered object jms/BD/sos/new created.
Administered object jms/BD/message/new created.
Command add-resources executed successfully.
C:\glassfish-4.1\glassfish4\bin>
```

2 在 Linux 上安装配置

2.1 Java 开发工具：JDK

2.1.1 资源地址

JDK 的官方地址为：

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

云盘地址：（或者更高版本）

QXSK 环境资源\linux64\jdk-8u74-linux-x64.rpm

2.1.2 安装

在安装 jdk 时，首先检验系统原版本，查看是否已经安装了 openjdk:

```
[root@localhost ~]# java -version
java version "1.7.0_75"
OpenJDK Runtime Environment (rhel-2.5.4.2.el7_0-x86_64 u75-b13)
OpenJDK 64-Bit Server VM (build 24.75-b04, mixed mode)
[root@localhost ~]#
```

如果已经安装了 jdk 将其卸载掉，然后安装系统需要的 jdk 版本。

```
[root@localhost ~]# rpm -qa|grep java
java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64
python-javapackages-3.4.1-6.el7_0.noarch
tzdata-java-2015a-1.el7.noarch
java-1.7.0-openjdk-headless-1.7.0.75-2.5.4.2.el7_0.x86_64
javapackages-tools-3.4.1-6.el7_0.noarch
[root@localhost ~]#
```

卸载命令如下所示：

```
[root@localhost ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64

[root@localhost ~]#
[root@localhost ~]# rpm -e java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64
error: package java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64 is not installed
[root@localhost ~]# rpm -e --nodeps java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64
error: package java-1.7.0-openjdk-1.7.0.75-2.5.4.2.el7_0.x86_64 is not installed
[root@localhost ~]# rpm -e --nodeps python-javapackages-3.4.1-6.el7_0.noarch

[root@localhost ~]#
[root@localhost ~]# rpm -e --nodeps python-javapackages-3.4.1-6.el7_0.noarch
error: package python-javapackages-3.4.1-6.el7_0.noarch is not installed
[root@localhost ~]# rpm -e --nodeps tzdata-java-2015a-1.el7.noarch
[root@localhost ~]# rpm -e --nodeps java-1.7.0-openjdk-headless-1.7.0.75-2.5.4.2.el7_0.x86_64
[root@localhost ~]#
[root@localhost ~]# rpm -e --nodeps javapackages-tools-3.4.1-6.el7_0.noarch
[root@localhost ~]#
```

运行命令：

`rpm -ihv jdk-8u74-linux-x64.rpm`

2.1.3 设置环境变量

编辑 “/etc/profile”，添加或者修改以下行：

```
LANG="en_US.UTF-8"  
JAVA_HOME=/usr/java/jdk1.8.0_74  
CLASSPATH=.:$JAVA_HOME/lib  
PATH=$JAVA_HOME/bin:$PATH  
export LANG PATH JAVA_HOME CLASSPATH
```

2.1.4 验证安装

运行以下命令：

```
source /etc/profile  
java -version
```

2.2 java 串口驱动

2.2.1 资源地址

官网地址：

<http://rxtx.qbang.org/pub/rxtx/rxtx-2.2pre2-bins.zip>

注意：在 Linux64 上写串口只有这个版本不会引起系统崩溃。

云盘地址：（或者更高版本）

QXSK 环境资源\linux64\rxtx\

2.2.2 安装

运行以下命令以复制驱动文件到 JAVA 目录下：

```
cp RXTXcomm.jar $JAVA_HOME/jre/lib/ext/  
cp librxtxSerial.so $JAVA_HOME/jre/lib/amd64/
```

必须重启 GlassFish 才能使串口驱动生效。

2.3 数据库系统：MySQL

2.3.1 资源地址

官网地址：

<http://dev.mysql.com/downloads/>

云盘地址：（或者更高版本）

QXSK 环境资源\linux64\mysql\mysql57-community-release-el7-7.noarch.rpm

2.3.2 清理环境

运行以下命令来检查并卸载系统中已有的 mysql 版本：

```
yum remove mariadb, mariadb-devel, mariadb-libs, mariadb-server
yum-config-manager --disable mariadb
yum repolist all | grep mysql
rpm -qa | grep mysql
rpm -e <mysql 包名>
```

2.3.3 安装

运行以下命令以安装 mysql 资源库：

```
rpm -ihv mysql57-community-release-el7-7.noarch.rpm
```

然后运行以下命令以安装 mysql 服务器和客户端：（该步骤需要等待下载 mysql 各个包，时间较长）

```
yum install mysql
```

应确保 MySQL 版本在 5.7 以上。

接着运行以下命令以升级系统中已有的 mysql 数据：

```
mysql_upgrade -u root -p --force
```

最后运行以下命令以启动或者终止 mysql 服务器：

```
service mysqld start
service mysqld status
```

//请参考以下官网资料安装配置 mysql：

<http://dev.mysql.com/doc/refman/5.7/en/linux-installation.html>

<http://dev.mysql.com/doc/refman/5.7/en/linux-installation-yum-repo.html>

若碰到如下错误，则是因为系统中还残留旧版本 mysql，安装之前应当彻底清除已存在版本：

mysql-community-release conflicts with mysql57-community-release-el7-7.noarch

2.3.4 设置账户安全

运行以下命令以查看 mysql 安装时自动生成的 root 密码：

```
cat /var/log/mysqld.log | grep 'temporary password'
```

（在/var/log/mysqld.log 中找到 temporary password 这一行，双引号中的既是密码）

进入 mysql，#mysql -uroot -p，密码即上一步中的密码。

运行以下 mysql 命令来修改 root 密码：

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'new_password';
```

然后参考以下网页设置账号安全（如删除测试账户和测试数据库）：

<http://dev.mysql.com/doc/refman/5.7/en/postinstallation.html>

<http://dev.mysql.com/doc/refman/5.7/en/default-privileges.html>

2.3.5 设置字符集

编辑 “/etc/my.cnf”，添加或者修改以下行：

```
character-set-server=utf8
collation-server=utf8_unicode_ci
```

然后重启 mysql。

运行以下 mysql 命令以验证字符集已正确设置为 utf8：

```
mysql> show variables like 'character_set%';
```

| Variable_name | Value |
|--------------------------|----------------------------|
| character_set_client | utf8 |
| character_set_connection | utf8 |
| character_set_database | utf8 |
| character_set_filesystem | binary |
| character_set_results | utf8 |
| character_set_server | utf8 |
| character_set_system | utf8 |
| character_sets_dir | /usr/share/mysql/charsets/ |

8 rows in set (0.00 sec)

2.3.6 验证安装

运行以下命令：

```
mysql -u root -p
```

应该能够用修改后的 root 密码登录进去。

运行以下 mysql 命令：（用实际的密码）

```
SET PASSWORD = PASSWORD('XXXXXXX');
```

若不执行 set password 命令则——mysql 命令返回以下错误，（用户修改未同步）：

ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

2.3.7 导入平台数据库

按以下顺序导入表结构、存储过程、初始化数据、事件、触发器：

```
mysql -uroot -p 密码 < bd_schema_sp.sql
mysql -uroot -p 密码 beidou < bd_adjust_schema.sql
mysql -uroot -p 密码 beidou < bd_data.sql
mysql -uroot -p 密码 beidou < bd_events.sql
mysql -uroot -p 密码 beidou < bd_triggers.sql
```

2.3.8 创建复制库和历史库

Mysql 正常安装完毕以后会自动创建和配置好一个缺省的数据库实例，我们把这个缺省实例作为主库。在

CentOS 7 上，主库的数据目录是 “/var/lib/mysql”。

以 root 账号执行以下步骤：

- 1) 停止主库：

```
systemctl stop mysqld
```

- 2) 复制主库数据目录：

```
cd /var/lib
cp -R mysql mysql-repl
rm -f mysql-repl/auto.cnf
rm -rf mysql-repl/mysql-bin*
chmod -R 750 mysql-repl
chown -R mysql:mysql mysql-repl
cp -R mysql mysql-his
rm -f mysql-his/auto.cnf
rm -rf mysql-his/mysql-bin*
chmod -R 750 mysql-his
chown -R mysql:mysql mysql-his
```

- 3) 若 SELinux 正在启用（命令 “getenforce” 返回 “Enforcing”），则必须执行以下命令：

```
chcon -R --reference=/var/lib/mysql /var/lib/mysql-repl/
semanage fcontext -a -t mysqld_db_t "/var/lib/mysql-repl(/.*)?"
restorecon -R /var/lib/mysql-repl
chcon -R --reference=/var/lib/mysql /var/lib/mysql-his/
semanage fcontext -a -t mysqld_db_t "/var/lib/mysql-his(/.*)?"
restorecon -R /var/lib/mysql-his
semanage port -a -t mysqld_port_t -p tcp 3307
semanage port -a -t mysqld_port_t -p tcp 3308
```

- 4) 检查数据目录的属性，应当如下：

```
[root@localhost lib]# cd /var/lib
[root@localhost lib]# ls -ladZ mysql*
drwxr-x--x. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql
drwxr-x---. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql-files
drwxr-x---. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql-his
drwxr-x---. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql-keyring
drwxr-x---. mysql mysql system_u:object_r:mysqld_db_t:s0 mysql-repl
```

2.3.9 修改配置文件

以 root 账号执行以下步骤：

- 1) 停止主库：

```
systemctl stop mysqld
```

- 2) 编辑配置文件 “/etc/my.cnf”，添加或者修改以下参数：

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
symbolic-links=0
log-error=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

port=3306
server-id=1

log-bin=mysql-bin
sync_binlog=1
gtid_mode=on
enforce_gtid_consistency=on
```

```

binlog_format=ROW
binlog-do-db=beidou
expire_logs_days=10
binlog_row_image=minimal

max_sp_recursion_depth=100

character-set-server=utf8
collation-server=utf8_general_ci
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

event_scheduler=1

[mysqld@repl]
datadir=/var/lib/mysql-repl
socket=/var/lib/mysql-repl/mysql.sock
symbolic-links=0
log-error=/var/log/mysqld-repl.log
pid-file=/var/run/mysqld/mysqld-repl.pid

port=3307
server-id=2

gtid_mode=on
enforce_gtid_consistency=on
skip-slave-start=1

max_sp_recursion_depth=100

character-set-server=utf8
collation-server=utf8_general_ci
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

event_scheduler=1

[mysqld@his]
datadir=/var/lib/mysql-his
socket=/var/lib/mysql-his/mysql.sock
symbolic-links=0
log-error=/var/log/mysqld-his.log
pid-file=/var/run/mysqld/mysqld-his.pid

port=3308
server-id=3

gtid_mode=on
enforce_gtid_consistency=on
skip-slave-start=1

max_sp_recursion_depth=100

character-set-server=utf8
collation-server=utf8_general_ci
sql_mode=STRICT_TRANS_TABLES,NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_AUTO_CREATE_USER,NO_ENGINE_SUBSTITUTION

event_scheduler=1

```

2.3.10 创建复制账号

以 root 账号执行以下步骤：

- 1) 启动主库：

```
systemctl start mysqld
```

- 2) 登入主库：

```
mysql -u root -p
```

- 3) 输入以下命令（xxxxxx 换成实际密码），以创建数据复制账户：

```
CREATE USER 'repl'@'localhost' IDENTIFIED BY 'xxxxxx';
GRANT REPLICATION SLAVE ON *.* TO 'repl'@'localhost';
SHOW MASTER STATUS;
```

2.3.11 配置复制库

以 root 账号执行以下步骤：

- 1) 启动复制库：

```
systemctl start mysqld@repl
```

此命令正常应当无返回。

用命令 “ps -ef mysql | grep mysql” 检查复制库的运行状态：

```
[root@localhost ~]# ps -ef | grep mysql
mysql      410      1  0 16:55 ?        00:00:01 /usr/sbin/mysqld --daemonize --pid-file=/var/run
/mysqld/mysqld.pid
mysql      991      1  0 17:02 ?        00:00:00 /usr/sbin/mysqld --defaults-group-suffix=@repl -
-daemonize --pid-file=/var/run/mysqld/mysqld-repl.pid
root       1040    2613  0 17:03 pts/1    00:00:00 mysql --port=3307 -u root -p
root       1539    1175  0 17:09 pts/4    00:00:00 grep --color=auto mysql
[root@localhost ~]#
```

- 2) 登入复制库：（由于是直接复制主库，因此 root 密码与主库相同）

```
mysql --host=127.0.0.1 --port=3307 -u root -p
```

或

```
mysql -S /var/lib/mysql-repl/mysql.sock -u root -p
```

- 3) 输入以下命令，以检查当前数据库实例的唯一标识：

```
show variables like '%server_uuid%';
show variables like 'server_id';
```

```
mysql> show variables like '%server_uuid%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_uuid   | def01a04-b96b-11e7-ac40-68f728cafa36 |
+-----+-----+
1 row in set (0.00 sec)

mysql> show variables like 'server_id';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id     | 2     |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

数据库服务器实例的 UUID 也可以用命令 “cat /var/lib/mysql-repl/auto.cnf” 来查看验证。

- 4) 输入以下命令，以建立与主库的数据复制关系：（xxxxx 换成前面建立的复制账户的密码）
`change master to master_host='localhost', master_user='repl', master_password='xxxxx', master_auto_position=1;`
- 5) 输入命令 “`show slave status \G`”，并检查以下这行的值是否为 1：
`Auto_Position: 1`
- 6) 输入以下命令，以启动数据同步复制：
`start slave;`
- 7) 输入命令 “`show slave status \G`”，并检查以下这些行的值（有的值可能随时间变化），以验证数据复制正在正确进行：

```
Slave_IO_State: Waiting for master to send event
Master_Host: localhost
Master_User: repl
Master_Port: 3306
Master_Log_File: mysql-bin.000004
Read_Master_Log_Pos: 486
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
Seconds_Behind_Master: 0
Master_Server_Id: 1
Master_UUID: 6bd8673c-9bee-11e5-a939-1c872c722172
Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates
```

- 8) 可以随时输入命令 “`start slave;`”或者 “`stop slave;`”，以启停数据同步。
- 9) 观察 “`show slave status \G`”的输出，当 “**Seconds_Behind_Master: 0**”意味着数据已全部同步到复制库中。
- 10) 如果主库中定义了事件，则需要把从库里的相应的事件删除或者让它们失效。输入命令 “`show events \G`”查看事件，并且对于每一个由主库复制而来的事件运行以下命令。这一步保证从库不会重复运行主库的事件：
`alter event e add partition monthly ON COMPLETION PRESERVE DISABLE;`
- 11) 如果需要从库每次启动后自动开始同步，则在它的配置文件中修改为 “`skip-slave-start=0`”。

2.3.12 建立历史库

以 root 账号执行以下步骤：

- 1) 启动历史库：
`systemctl start mysqld@his`
 此命令正常应当无返回。
 用命令 “`ps -ef | grep mysql`”检查复制库的运行状态：

```
[root@localhost lib]# ps -ef | grep mysql
mysql      12502      1  0 18:04 ?          00:00:01 /usr/sbin/mysqld --defaults-group-suffix=@repl --daemonize --pid-file=/var/run/mysqld/mysqld-repl.pid
mysql      15385      1  0 18:39 ?          00:00:00 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid
root       15421 14603  0 18:39 pts/4      00:00:00 mysql -u root -p
mysql      15693      1  0 18:42 ?          00:00:00 /usr/sbin/mysqld --defaults-group-suffix=@his --daemonize --pid-file=/var/run/mysqld/mysqld-his.pid
root       15751  2613  0 18:43 pts/1      00:00:00 grep --color=auto mysql
[root@localhost lib]#
```

- 2) 登入历史库：（由于是直接复制主库，因此 root 密码与主库相同）
`mysql --host=127.0.0.1 --port=3308 -u root -p`
 或
`mysql -S /var/lib/mysql-his/mysql.sock -u root -p`
- 3) 输入以下命令，以检查当前数据库实例的唯一标识：
`show variables like '%server_uuid%';`
`show variables like 'server_id';`

```
mysql> show variables like '%server_uuid%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_uuid   | 3fd16809-b971-11e7-8a61-68f728cafa36 |
+-----+-----+
1 row in set (0.00 sec)

mysql> show variables like 'server_id';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| server_id     | 3     |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

数据库服务器实例的 UUID 也可以用命令 “`cat /var/lib/mysql-his/auto.cnf`”来查看验证。

4) 输入以下命令，以建立与主库的数据复制关系：（xxxxx 换成前面建立的复制账户的密码）

`change master to master_host='localhost', master_user='repl', master_password='xxxxx', master_auto_position=1;`

5) 输入命令 “`show slave status \G`”，并检查以下这行的值是否为 1:

Auto_Position: 1

6) 输入以下命令，以启动数据同步复制:

`start slave;`

7) 输入命令 “`show slave status \G`”，并检查以下这些行的值（有的值可能随时间变化），以验证数据复制正在正确进行:

Slave_IO_State: Waiting for master to send event

Master_Host: localhost

Master_User: repl

Master_Port: 3306

Master_Log_File: mysql-bin.000004

Read_Master_Log_Pos: 486

Slave_IO_Running: Yes

Slave_SQL_Running: Yes

Seconds_Behind_Master: 0

Master_Server_Id: 1

Master_UUID: dfeae97b-a3cc-11e5-987b-68f728cafa36:1-2

Slave_SQL_Running_State: Slave has read all relay log; waiting for more updates

8) 可以随时输入命令 “`start slave;`”或者 “`stop slave;`”，以启停数据同步。

9) 观察 “`show slave status \G`”的输出，当 “**Seconds_Behind_Master: 0**”意味着数据已全部同步到复制库中。

10) 如果主库中定义了事件，则需要把从库里的相应的事件删除或者让它们失效。输入命令 “`show events \G`”查看事件，并且对于每一个由主库复制而来的事件运行以下命令。这一步保证从库不会重复运行主库的事件:

`alter event e_add_partition_monthly ON COMPLETION PRESERVE DISABLE;`

11) 如果需要从库每次启动后自动开始同步，则在它的配置文件中修改为 “`skip-slave-start=0`”。

2.3.13 备份平台数据库

输入以下命令以备份平台数据库:

`mysqldump -u root -p beidou > 平台数据库文件的路径`

此命令保存所有平台数据表定义以及数据。

2.4 Java EE 服务器：GlassFish

2.4.1 资源地址

官方地址为：

<https://glassfish.java.net>

网盘地址：（或者更高版本）

QXSK 环境资源\linux64\glassfish-4.1.zip

2.4.2 安装

在选定的目录下直接解包即可。一般安装在 “/usr/local”下：

```
cd /usr/local
unzip glassfish-4.1.zip
mv glassfish4 glassfish41
```

2.4.3 设置环境变量

编辑 “/etc/profile”，添加或者修改以下行：

```
PATH=/usr/java/jdk1.8.0_74/bin:/usr/local/glassfish41/bin:/usr/local/glassfish41/glassfish/bin:$PATH
export PATH JAVA_HOME CLASSPATH
```

2.4.4 创建域

使用命令 “asadmin”来控制和管理 GlassFish（启动、停止、配置、发布应用等）。

要启动 GlassFish，运行命令：（要先 cd 到 bin 目录，所有命令要加 “./”）

```
asadmin start-domain
```

例如：

```
[root@localhost glassfish4]# cd bin
[root@localhost bin]# ll
total 56
-rwxr-xr-x. 1 root root 2274 Aug 21 2014 asadmin
-rwxr-xr-x. 1 root root 2543 Aug 21 2014 asadmin.bat
-rwxr-xr-x. 1 root root 8285 Aug 21 2014 pkg
-rwxr-xr-x. 1 root root 9599 Aug 21 2014 pkg.bat
-rwxr-xr-x. 1 root root 8285 Aug 21 2014 updatetool
-rwxr-xr-x. 1 root root 9599 Aug 21 2014 updatetool.bat
[root@localhost bin]# ./asadmin start-domain domain1
Waiting for domain1 to start ....
Successfully started the domain : domain1
domain Location: /usr/glassfish4/glassfish/domains/domain1
Log File: /usr/glassfish4/glassfish/domains/domain1/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
[root@localhost bin]#
```

运行以下命令：

```
[root@qxsk99 local]# ./asadmin list-domains
domain1 not running
Command list-domains executed successfully.
```

```
[root@qxsk99 local]# ./asadmin delete-domain domain1
Domain domain1 deleted.
Command delete-domain executed successfully.

[root@qxsk99 local]# ./asadmin create-domain qxsk
Enter admin user name [Enter to accept default "admin" / no password]>
Using default port 4848 for Admin.
Using default port 8080 for HTTP Instance.
Using default port 7676 for JMS.
Using default port 3700 for IIOP.
Using default port 8181 for HTTP_SSL.
Using default port 3820 for IIOP_SSL.
Using default port 3920 for IIOP_MUTUALAUTH.
Using default port 8686 for JMX_ADMIN.
Using default port 6666 for OSGI_SHELL.
Using default port 9009 for JAVA_DEBUGGER.
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=qxsk99,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Distinguished Name of the self-signed X.509 Server Certificate is:
[CN=qxsk99-instance,OU=GlassFish,O=Oracle Corporation,L=Santa Clara,ST=California,C=US]
Domain qxsk created.
Domain qxsk admin port is 4848.
Domain qxsk allows admin login as user "admin" with no password.
Command create-domain executed successfully.
```

如果系统报告端口 4848 已被占据，请用以下命令检查进程并且杀掉占据端口的进程：

```
netstat -lntp
```

当主机名设置不正确时，也会虚报端口被占据，请用以下命令设置主机名：

```
hostnamectl set-hostname <主机名>
```

2.4.5 启动 GlassFish

使用命令 “asadmin” 来控制和管理工作 GlassFish（启动、停止、配置、发布应用等）。

要启动 GlassFish，运行命令：

```
[root@qxsk99 local]# ./asadmin start-domain qxsk
Waiting for qxsk to start ...
Successfully started the domain : qxsk
domain Location: /usr/local/glassfish41/glassfish/domains/qxsk
Log File: /usr/local/glassfish41/glassfish/domains/qxsk/logs/server.log
Admin Port: 4848
Command start-domain executed successfully.
```

2.4.6 管理 GlassFish

要管理 GlassFish，访问以下网址：

```
http://localhost:4848
```

2.4.7 停止 GlassFish

要停止 GlassFish，运行以下命令：

```
asadmin stop-domain qxsk
```


2.4.8 添加 JDBC 驱动

云盘地址：（或者更高版本）

QXSK 环境资源\linux64\mysql\mysql-connector-java-5.1.35-bin.jar

把这个 jar 包放入以下目录：

[/usr/local/glassfish41/glassfish/domains/qxsk/lib/ext/](#)

必须重启 GlassFish 才能使 JDBC 驱动生效。

2.4.9 配置应用资源

用文本编辑器打开文件 “sun-resources.xml”，把 mysql 连接池的密码改为实际值：

```
<property name="User" value="root"/>
```

```
<property name="Password" value="mysql 的 root 实际密码"/>
```

然后运行以下命令导入应用资源：（注：需要在在 glassfish 开启状态下才能导入）

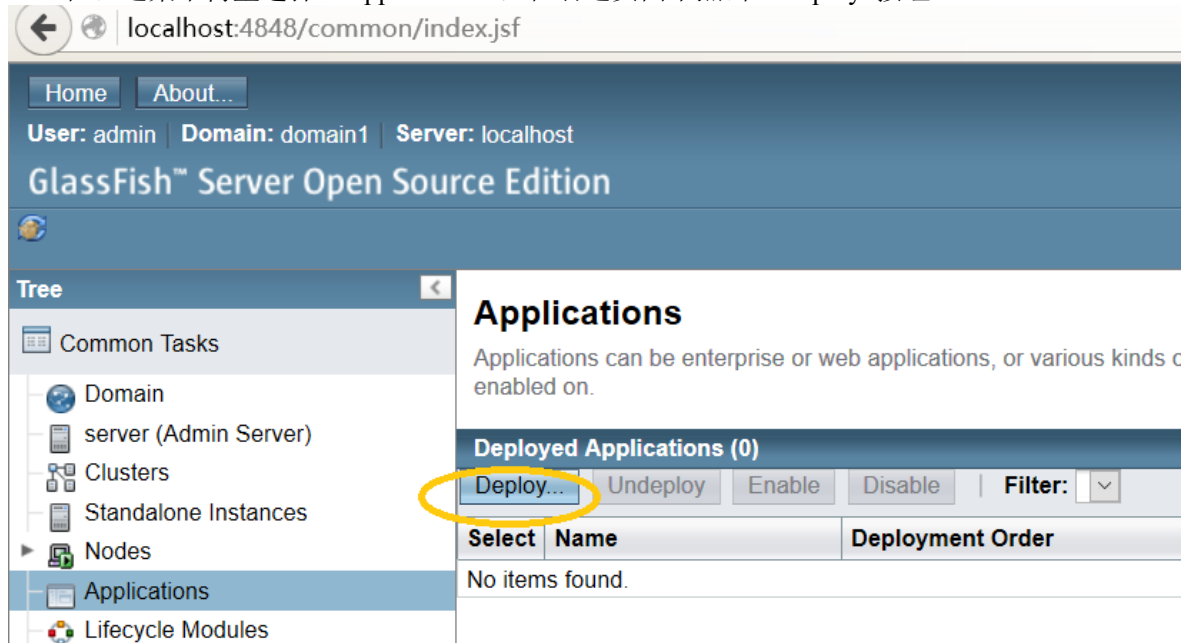
```
[root@qxsk99 20160222]# ./asadmin add-resources sun-resources.xml 文件路径
Enter admin user name> admin
Enter admin password for user "admin">
JDBC connection pool beidouPool created successfully.
JDBC resource jdbc/beidou created successfully.
Mail Resource mail/QXSKSession created.
Administered object jms/email/sending created.
Administered object jms/GSM/message/sending created.
Administered object jms/BD/message/sending created.
Administered object jms/BD/statement/sending created.
Administered object jms/BD/statement4/received created.
Administered object jms/BD/statement21/received created.
Command add-resources executed successfully.
```

3 发布平台应用

3.1 发布应用包

登陆 GlassFish 管理页面：<http://localhost:4848> 或者 <http://<机器 IP 或者域名>:4848>，发布平台应用：

- 1) 在左边菜单树里选择 “Applications”，在右边页面中点击 “Deploy”按钮



- 2) 选择本地或者服务器上的平台 war 包上传：

Deploy Applications or Modules

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

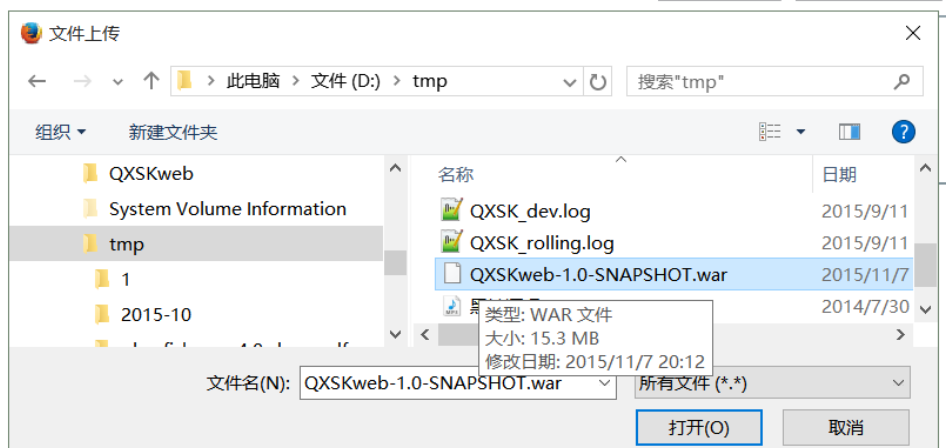
Location: ☒ Packaged File to Be Uploaded to the Server

[浏览...](#) 未选择文件。

☐ Local Packaged File or Directory That Is Accessible from GlassFish Server

[Browse Files...](#)

[Browse Folders...](#)



- 3) “Type”选择 “Web Application”，“Context Root”改成 “QXSKweb”，“Application Name”改成 “QXSKweb”，“Virtual Servers”选择 “server”，其它保持缺省值，点击 “OK”按钮：

[Logout](#)
[Help](#)

213.65.99

Source Edition

Deploy Applications or Modules

[OK](#)
[Cancel](#)

Specify the location of the application or module to deploy. An application can be in a packaged file or specified as a directory.

* Indicates required field

Location: ☐ **Packaged File to Be Uploaded to the Server**

[浏览...](#)
[未选择文件。](#)

☒ **Local Packaged File or Directory That Is Accessible from GlassFish Server**

[Browse Files...](#)
[Browse Folders...](#)

Type: *

Context Root:
Path relative to server's base URL.

Application Name: *

Virtual Servers:

4) 平台应用被成功发布到 GlassFish 服务器上:

Applications

Applications can be enterprise or web applications, or various kinds of modules. Restart an application or module by clicking on the reload link, this action will apply only to the targets that the application or module is enabled on.

| Deployed Applications (1) | | | | | |
|-------------------------------------|---------|------------------------------|-------------------------------------|------------------------|--|
| | | Deploy... | Undeploy | Enable | Disable |
| | | Filter: <input type="text"/> | | | |
| Select | Name | Deployment Order | Enabled | Engines | Action |
| <input checked="" type="checkbox"/> | QXSKweb | 100 | <input checked="" type="checkbox"/> | ejb, web | Launch Redeploy Reload |

5) 选择应用名, 点击各个按钮以开启、关闭、发布、重新发布、重载、访问此应用。

6) 点击 “Launch”按钮以打开浏览器访问平台应用, 地址是:

<http://机器名或者 IP 地址:8080/QXSKweb/faces/login.xhtml>



| | |
|-------------------------------------|--------------------------|
| 用户名 * | <input type="text"/> |
| 密码 * | <input type="password"/> |
| <input type="button" value="登录"/> | |
| <input type="button" value="+ 注册"/> | |

测试版本的管理员账户是“admin”，密码“admin”。

3.2 平台应用的短地址

可以把平台地址设置为 glassfish 域的缺省入口。把安装包里文件“index.html”替换以下目录下的文件：

</usr/local/glassfish41/glassfish/domains/qxsk/docroot/index.html>

然后用户就可以直接用以下地址访问平台了：

<http://服务器地址>

3.3 部署应用时可能遇到的问题及其解决办法

3.3.1 “NoClassFound”

若碰到 **NoClassFound 错误**，则请检查 CLASSPATH 的定义是否已生效，并且检查串口驱动和 JDBC 库是否已复制到正确的目录下。

3.3.2 “Exception starting EMBEDDED broker=Cannot create broker instance”

若碰到以下错误，则说明上次失败的部署未被清理干净：

MQJMSRA_RA4001: start:Aborting:Exception starting EMBEDDED broker=Cannot create broker instance.

解决办法是删除以下文件，然后重启 glassfish：

rm glassfish/domains/qxsk/imq/instances/imqbroker/lock

3.3.3 “Error in linking security policy for QXSKweb -- Inconsistent Module State”

若碰到以下错误，则说明上次失败的部署未被清理干净：

Exception during lifecycle processing org.glassfish.deployment.common.DeploymentException: Error in linking security policy for QXSKweb -- Inconsistent Module State.

解决办法是删除以下目录，然后重启 glassfish：

```
rm -rf /usr/local/glassfish4/glassfish/domains/qxsk/generated/
```

3.4 GlassFish4.1.*的 EclipseLink 补丁

GlassFish5 在解析 WebService 数据处理方面有 bug 直接影响客户端解析数据，因此目前我们仍然用 GlassFish4:

<https://github.com/javaee/glassfish/issues/22378>

GlassFish 4.1.*有一个 bug，需要打上 EclipseLink 补丁，才能正确提供 RESTful 服务：

1) 下载最新版的 OSGI bundles （需要大于 2.6.2），例如 “eclipseLink-plugins-2.6.3.v20160428-59c81c5.zip”

<http://www.eclipse.org/eclipseLink/downloads/>

2) 下载后解包，取出并重命名以下三个文件（截掉版本字符串）：

org.eclipse.persistence.moxy.jar

org.eclipse.persistence.asm.jar

org.eclipse.persistence.core.jar

3) 将这 3 个文件覆盖目录下 glassfish-4.1.*\glassfish\modules 的同名文件。

4) 删除域缓冲目录，比如：glassfish-4.1.*\glassfish\domains\<域名>\osgi-cache\

5) 重启 GlassFish

3.5 调整 JVM

1) 登录 glassfish 控制台： <http://localhost:4848>

2) 进入 glassfish 控制台，在-配置->server-config->JVM 设置->JVM 选项中可以看到：

-Xmx512m JVM 最大允许分配的堆内存，按需分配

-XX:PermSize=64M JVM 初始分配的非堆内存

-XX:MaxPermSize=128M JVM 最大允许分配的非堆内存，按需分配

JVM 初始分配的堆内存由-Xms 指定，默认是物理内存的 1/64；

JVM 最大分配的堆内存由-Xmx 指定，默认是物理内存的 1/4。

JVM 使用 -XX:PermSize 设置非堆内存初始值，默认是物理内存的 1/64；

由 XX:MaxPermSize 设置最大非堆内存的大小，默认是物理内存的 1/4。

3) 更改这些参数，重启 glassfish 即可。

4 配置 SSL

目前本系统没有合法的 SSL 证书。

曾经在 2017 年 1 月申请过 StartSSL 免费证书，本节以其为例说明配置 GlassFish 安全证书的过程。

4.1 申请证书

1) 打开命令窗口，在一个临时目录下，执行以下命令：

```
D:\tmp\sslcert>keytool -genkey -alias qxsk9 -keyalg RSA -keystore qxsk9keystore.jks -keysize 2048
```

输入密钥库口令：

再次输入新口令：

您的名字与姓氏是什么？

[Unknown]: www.qxsk9.com

您的组织单位名称是什么？

[Unknown]: [development](#)

您的组织名称是什么？

[Unknown]: [linkbit](#)

您所在的城市或区域名称是什么？

[Unknown]: [NanJing](#)

您所在的省/市/自治区名称是什么？

[Unknown]: [JiangSu](#)

该单位的双字母国家/地区代码是什么？

[Unknown]: [cn](#)

CN=www.qxsk9.com, OU=[development](#), O=[linkbit](#), L=[NanJing](#), ST=[JiangSu](#), C=[cn](#) 是否正确？

[否]: [是](#)

输入 <[qxsk9](#)> 的密钥口令

(如果和密钥库口令相同, 按回车):

```
D:\tmp\sslcert>keytool -certreq -alias qxsk9 -file qxsk9.csr -keystore qxsk9keystore.jks
```

输入密钥库口令：

2) 把生成的文件 “qxsk9keystore.jks”和 “qxsk9.csr”发送给 CA 认证机构。

3) 收到的 StartSSL 证书包括以下文件:

| 名称 | 修改日期 | 类型 | 大小 |
|-------------------|-----------------|------|------|
| Intermediate.crt | 2017/1/13 20:59 | 安全证书 | 3 KB |
| root.crt | 2017/1/13 20:59 | 安全证书 | 3 KB |
| www.qxsk9.com.crt | 2017/1/13 20:59 | 安全证书 | 3 KB |

4.2 导入证书

1) 把以下文件都复制到 GlassFish 域的 config 目录 (如 D:\glassfish-4.1.2\glassfish\domains\qxsk9\config) 下:

qxsk9keystore.jks qxsk9.csr root.crt Intermediate.crt www.qxsk9.com.crt

2) 备份该目录下的文件 keystore.jks , cacerts.jks、和 domain.xml, 当配置不成功导致系统无法正常工作时, 恢复这三个文件即可。

3) 打开命令窗口, 进入以上目录, 执行以下命令:

```
D:\glassfish41\glassfish\domains\mara\config>keytool -list -keystore keystore.jks
```

输入密钥库口令:

密钥库类型: JKS

密钥库提供方: SUN

您的密钥库包含 2 个条目

glassfish-instance, 2016-11-16, PrivateKeyEntry,

证书指纹 (SHA1):

s1as, 2016-11-16, PrivateKeyEntry,

证书指纹 (SHA1):

```
D:\glassfish41\glassfish\domains\mara\config>keytool -list -keystore qxsk9keystore.jks
```

输入密钥库口令:

密钥库类型: JKS

密钥库提供方: SUN

您的密钥库包含 1 个条目

qxsk9, 2017-1-13, PrivateKeyEntry,

证书指纹 (SHA1):

```
D:\glassfish41\glassfish\domains\mara\config>keytool -importkeystore -srckeystore qxsk9keystore.jks -destkeystore keystore.jks
```

输入目标密钥库口令:

输入源密钥库口令:

已成功导入别名 qxsk9 的条目。

已完成导入命令: 1 个条目成功导入, 0 个条目失败或取消

```
D:\glassfish41\glassfish\domains\mara\config>keytool -list -keystore keystore.jks
```

输入密钥库口令:

密钥库类型: JKS

密钥库提供方: SUN

您的密钥库包含 3 个条目

qxsk9, 2017-2-6, PrivateKeyEntry,

证书指纹 (SHA1):

glassfish-instance, 2016-11-16, PrivateKeyEntry,

证书指纹 (SHA1):

slas, 2016-11-16, PrivateKeyEntry,

证书指纹 (SHA1):

```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -v -trustcacerts -alias root -file root.crt -keystore keystore.jks
```

输入密钥库口令:

所有者: CN=StartCom Certification Authority, OU=Secure Digital Certificate Signing, O=StartCom Ltd., C=IL

发布者: CN=StartCom Certification Authority, OU=Secure Digital Certificate Signing, O=StartCom Ltd., C=IL

序列号: 1

有效期开始日期: Mon Sep 18 03:46:36 CST 2006, 截止日期: Thu Sep 18 03:46:36 CST 2036

证书指纹:

MD5: 22:4D:8F:8A:FC:F7:35:C2:BB:57:34:90:7B:8B:22:16

。 。 。 。 。

是否信任此证书? [否]: 是

证书已添加到密钥库中

[正在存储 keystore.jks]


```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -v -trustcacerts -alias root -file root.crt -keystore cacerts.jks
```

输入密钥库口令:

所有者: CN=StartCom Certification Authority, OU=Secure Digital Certificate Signing, O=StartCom Ltd., C=IL

发布者: CN=StartCom Certification Authority, OU=Secure Digital Certificate Signing, O=StartCom Ltd., C=IL

序列号: 1

有效期开始日期: Mon Sep 18 03:46:36 CST 2006, 截止日期: Thu Sep 18 03:46:36 CST 2036

证书指纹:

MD5: 22:4D:8F:8A:FC:F7:35:C2:BB:57:34:90:7B:8B:22:16

。 。 。 。 。 。

是否信任此证书? [否]: 是

证书已添加到密钥库中

[正在存储 cacerts.jks]

```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -v -trustcacerts -alias intermediate -file Intermediate.crt -keystore keystore.jks
```

输入密钥库口令:

证书已添加到密钥库中

[正在存储 keystore.jks]

```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -v -trustcacerts -alias intermediate -file Intermediate.crt -keystore cacerts.jks
```

输入密钥库口令:

证书已添加到密钥库中

[正在存储 cacerts.jks]

```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -alias qxsk9 -trustcacerts -file www.qxsk9.com.crt -keystore keystore.jks
```

输入密钥库口令:

输入 <qxsk9> 的密钥口令

证书回复已安装在密钥库中 <<<-- 注意这句

```
D:\glassfish41\glassfish\domains\mara\config>keytool -import -alias qxsk9 -trustcacerts -file www.qxsk9.com.crt -keystore cacerts.jks
```

输入密钥库口令:

证书已添加到密钥库中

D:\glassfish41\glassfish\domains\mara\config>keytool -keypasswd -keystore keystore.jks -alias qxsk9

输入密钥库口令:

输入 <qxsk9> 的密钥口令

新<qxsk9> 的密钥口令:

重新输入新<qxsk9> 的密钥口令:

4.3 配置 GlassFish

1) 停止 GlassFish 服务器

2) 编辑文件 D:\glassfish-4.1.2\glassfish\domains\qxsk9\config\domain.xml，将其中所有字符串“slas”替换为“qxsk9”

3) 启动 GlassFish 服务器

实际配置过程可能会有波折，目前没有条件再次确认步骤。。。

5 管理 **www.qxsk9.com** 网站

本系统服务器的域名是 **www.qxsk9.com**，并且 GlassFish 的域 **qxsk9** 占用了 80 端口，因此把网站文件放在以下目录中，即可用此域名直接访问系统的网站了：

</usr/local/glassfish41/glassfish/domains/qxsk/docroot/>

6 CentOS 服务器的安全防护措施

6.1 防火墙 firewalld

在 CentOS 7 上缺省的防火墙是 firewalld 而不是 iptables 了，这两个只能选其一。应当保持 firewalld 的运行。

常用的 firewalld 命令：

显示防火墙状态

```
[root@58 ~]# firewall-cmd --state
```

```
Running
```

```
[root@58 ~]# systemctl status firewalld
```

```
firewalld.service - firewalld - dynamic firewall daemon
```

```
Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled)
```

```
Active: active (running) since Fri 2015-12-04 21:02:38 CST; 16h ago
```

```
Main PID: 9195 (firewalld)
```

```
CGroup: /system.slice/firewalld.service
```

```
    9195 /usr/bin/python -Es /usr/sbin/firewalld --nofork --nopid
```

```
Dec 04 21:02:38 xxxxx systemd[1]: Started firewalld - dynamic firewall daemon.
```

```
[root@58 ~]#
```

显示防火墙版本

```
[root@58 ~]# firewall-cmd --version
```

```
0.3.9
```

开放端口

```
[root@localhost ~]# firewall-cmd --zone=public --add-port=4848/tcp --permanent
```

```
success
```

重启防火墙

```
[root@localhost ~]# firewall-cmd --reload
```

```
Success
```

获取支持的区域列表

```
[root@58 ~]# firewall-cmd --get-zones
```

```
block dmz drop external home internal public trusted work
```

获取所有支持的服务

```
[root@58 ~]# firewall-cmd --get-services
```

```
RH-Satellite-6 amanda-client bacula bacula-client dhcp dhcpv6 dhcpv6-client dns ftp high-availability http https
imaps ipp ipp-client ipsec kerberos kpasswd ldap ldaps libvirt libvirt-tls mdns mountd ms-wbt mysql nfs ntp openvpn pmcd
pmproxy pmwebapi pmwebapis pop3s postgresql proxy-dhcp radius rpc-bind samba samba-client smtp ssh telnet tftp tftp-
client transmission-client vnc-server wbem-https
```

获取所有支持的 ICMP 类型

```
[root@58 ~]# firewall-cmd --get-icmp-types
```

```
destination-unreachable echo-reply echo-request parameter-problem redirect router-advertisement router-
solicitation source-quench time-exceeded
```

列出全部启用的区域的特性

```
[root@58 ~]# firewall-cmd --list-all-zones
```

```
.....
```

获取默认区域的网络设置

```
[root@58 ~]# firewall-cmd --get-default-zone
public
```

输出区域 <zone> 全部启用的特性。

```
[root@58 ~]# firewall-cmd --zone=public --list-all
public (default, active)
  interfaces: eno1
  sources:
  services: dhcpv6-client ssh
  ports: 8181/tcp 80/tcp 8080/tcp 4848/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

列举区域中启用的服务

```
[root@58 ~]# firewall-cmd --zone=public --list-services
dhcpv6-client ssh
```

启用应急模式阻断所有网络连接，以防出现紧急状况

```
firewall-cmd --panic-on
```

禁用应急模式

```
firewall-cmd --panic-off
```

查询应急模式

```
[root@58 ~]# firewall-cmd --query-panic
no
```

此命令返回应急模式的状态，没有输出。可以使用以下方式获得状态输出：

```
firewall-cmd --query-panic && echo "On" || echo "Off"
```

6.2 只开放必要的端口和服务

查看服务和端口：

```
[root@58 ~]# firewall-cmd --zone=public --list-all
public (default, active)
  interfaces: eno1
  sources:
  services: dhcpv6-client ssh
  ports: 8848/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

应保持对外只开放最少的资源。

6.3 限制 SSH 的使用

服务器刚上线，就遭到了大量 ssh 和 ssh-dos 扫描。以下措施可以减少或者避免网络安全问题。

注意，修改配置完毕，需重启 ssh 服务：

```
systemctl restart sshd.service
```

6.3.1 禁止 root 通过 ssh 登录

编辑文件 “/etc/ssh/sshd_config”，设置以下参数：

```
PermitRootLogin no
IgnoreRhosts yes
IgnoreUserKnownHosts yes
PrintMotd no
StrictModes yes
PubkeyAuthentication yes
RSAAuthentication yes
PermitEmptyPasswords no
```

6.3.2 设置登录白名单

编辑文件 “/etc/ssh/sshd_config”，设置以下参数：

```
AllowUsers hello
```

定义允许用 ssh 登录的用户名列表。

6.3.3 限制登录失败次数

编辑文件 “/etc/ssh/sshd_config”，设置以下参数：

```
MaxAuthTries 6
MaxSessions 10
LoginGraceTime 2m
```

即超过 6 次验证错误断开连接。如果用户不能成功登录，服务器等待 2 分钟后切断连接。

6.4 安装配置 Fail2Ban

Fail2Ban 是老牌的防范口令攻击工具。官方地址为：

<http://www.fail2ban.org/>

1) 在 CentOS 上可以用 yum 安装此工具：

```
yum install fail2ban
```

2) 复制配置文件：

```
cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

3) 编辑配置文件 “/etc/fail2ban/jail.conf”，设置以下参数：

```
bantime = -1
findtime = 600
port = 0:65535
banaction = firewallcmd-new
```

```
[sshd]
port = ssh
logpath = /var/log/secure
enabled = true
filter = sshd
action = firewallcmd-new[]
bantime = -1
```

```
[sshd-ddos]
port = ssh
logpath = /var/log/messages
enabled = true
filter = sshd-ddos
action = firewallcmd-new[]
bantime = -1
```

4) 使 fail2ban 生效:

```
systemctl enable fail2ban
systemctl start fail2ban
```

5) 查看 fail2ban 的日志 “/var/log/fail2ban.log”

6.5 注释掉不必要的账户

1) 备份密码文件:

```
cp /etc/passwd /etc/passwd.bak
```

2) 编辑密码文件 “/etc/passwd”，注释掉以下用户:

```
#adm:x:3:4:adm:/var/adm:/sbin/nologin
#lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
#sync:x:5:0:sync:/sbin:/bin/sync
#shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
#halt:x:7:0:halt:/sbin:/sbin/halt
#uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
#operator:x:11:0:operator:/root:/sbin/nologin
#games:x:12:100:games:/usr/games:/sbin/nologin
#gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
#ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin #注释掉 ftp 匿名账号
```

3) 备份组文件

```
cp /etc/group /etc/group.bak
```

4) 编辑组文件 “/etc/group”，注释掉以下组:

```
#adm:x:4:root,adm,daemon
#lp:x:7:daemon,lp
#uucp:x:14:uucp
#games:x:20:
#dip:x:40:
```

6.6 使用 webmin

webmin 是 unix 网管的利器。官网是:

<http://webmin.com/>

<http://sourceforge.net/projects/webadmin/>

通过浏览器远程管理 unix/linux 服务器。

Login: mara

- Webmin
- 系统
- 服务器
- 其它
- 网络
 - Bandwidth Monitoring
 - Fail2Ban Intrusion Detector
 - FirewallD
 - idmappd daemon
 - Kerberos5
 - Linux Firewall
 - Network Configuration
 - NFS Exports
 - NIS Client and Server
 - PPP Dialup Client
 - TCP Wrappers
- 硬件
- 群集
- Un-used Modules

Search:

[View Module's Logs](#)

[System Information](#)

[Refresh Modules](#)

[退出](#)



System Information

System hostname qxsk9 (192.168.1.3)

Operating system CentOS Linux 7.4.1708

Webmin version 1.881

Time on system Sun Apr 8 10:26:27 2018

Kernel and CPU Linux 3.10.0-693.5.2.el7.x86_64 on x86_64

Processor information Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz, 24 cores

Drive temperatures sda: 32°C sdb: 32°C sdc: 30°C (2 errors!)

System uptime 163 days, 23 hours, 43 minutes

Running processes 419

CPU load averages 0.27 (1 min) 0.20 (5 mins) 0.16 (15 mins)

CPU usage 0% user, 0% kernel, 0% IO, 100% idle

Real memory 6.09 GB used, 15.41 GB total

Virtual memory 26.73 MB used, 7.81 GB total

Local disk space 16.15 GB used, 1.72 TB total

Package updates All installed packages are up to date

Recent Webmin logins

| IP address | Last active at | State |
|---------------|------------------|------------|
| 49.80.119.217 | 08/四 /2018 10:26 | This login |
| 49.80.119.217 | 06/四 /2018 19:20 | Logged in |

Login: mara

- Webmin
- 系统
- 服务器
 - MySQL Database Server
 - Postfix Mail Server
 - PostgreSQL Database Server
 - Read User Mail
 - SSH Server
- 其它
- 网络
- 硬件
- 群集
- Un-used Modules

Search:

[View Module's Logs](#)

[System Information](#)

[Refresh Modules](#)

[退出](#)

[帮助..](#)
[模块配置](#)

SSH服务器

OpenSSH_7.4

[搜索文档..](#)

验证

网络

访问控制

杂项选项

客户端主机选项

Host SSH Keys

Edit Config Files

[应用变更](#)

单击此按钮，通过向正在运行的 SSHd 进程发送一个 SIGHUP 信号以应用当前的配置。

[Stop Server](#)

Click this button to stop the running SSH server. Once it is stopped, no users will be able to login via SSH, but existing connections will remain active.

6.7 禁止 root 运行应用程序

不应用 root 账户直接运行应用程序，而是应当创建专门的账户来运行各个应用程序，如 mysql 和 GlassFish。并且这些专门运行应用程序的账户只允许本地执行程序、不允许远程登陆。

6.8 数据库的安全设置

6.8.1 从官网下载安装

应当用 yum 下载安装，它会自动创建 mysql 的账户和组：

```
rpm -ihv mysql57-community-release-el7-7.noarch.rpm
yum install mysql-community-server
```

6.8.2 按官网步骤

应当按照 mysql 官网的步骤进行安装和配置：

<http://dev.mysql.com/doc/refman/5.7/en/linux-installation-yum-repo.html>

6.8.3 限制系统账户 mysql

系统账户 mysql 是 yum 安装过程自动创建的，注意禁止这个账户远程登陆。

6.8.4 修改 mysql 的 root 密码

Mysql 安装完毕 root 没有密码！应立即执行以下步骤：

```
[root@58 ~]# mysql -u root -p
mysql> use mysql;
mysql> update user set password=password("xxxxxxx") where user='root';
mysql> update user set authentication_string=password('xxxxxxx*') where user='root';
mysql> SELECT User, Host, HEX(authentication_string) FROM mysql.user;
mysql> SELECT User, Host, Password FROM mysql.user;
mysql> flush privileges;
```

6.8.5 清除匿名账户

Mysql 安装完毕存在空用户，应删除：

```
mysql> drop user "@'localhost';
mysql> drop user "@'x.x.x.x';
```

6.8.6 删除测试数据库

Mysql 安装完毕存在测试数据库，应删除：

```
mysql> DELETE FROM mysql.db WHERE Db LIKE 'test%';
mysql> FLUSH PRIVILEGES;
mysql> DROP DATABASE test;
```

6.9 GlassFish 的安全设置

从官网下载介质。

禁止开启远程管理。 GlassFish4*远程管理有严重的安全漏洞，尽管 4.2 声称解决了，但是经过实践发现漏洞依然存在：

http://www.nxca.gov.cn/user/user_find.action?artical.id=945

<文档结束>