

RestCloud-部署安装手册V2.0

安装准备

JDK包

Tomcat

MongoDB

Linux环境部署

1. 安装步骤
2. 本地文件上传工具包安装
3. JDK安装
4. Tomcat安装
 - 4.1 安装步骤
 - 4.2 设置tomcat内存
 - 4.3 设置tomcat开机自启
 - 4.4 配置停止tomcat的同时杀掉进程
- 5 MONGO安装
 - 5.1 安装步骤
 - 5.2 错误解决
 - 场景1：启动mongo时报错
 - 场景2：启动mongo提示权限不足
6. 安装RestCloud War工程包
7. 配置application.properties文件
8. 启动RestCloud服务
 - 8.1. 普通方式启动
 - 8.2. 服务方式启动（开机启动）
9. 更换序列号
 - 9.1. 平台界面替换
 - 9.2. MONGODB数据库替换
 - 9.3. 后端代码配置文件替换
 - 9.4. 自动初始化数据库并启动系统
10. 安装后启动异常
 - 10.1. java.net.UnknownHostException
 - 10.2. The system has not been initialized yet序列号过期

Windows环境部署

1. 安装步骤
2. JDK1.8安装
3. Tomcat8.5版本安装
4. RestCloud链接MongoDB
- 5.启动tomcat测试运行

文档编号	V2.0
文档密级	---

RestCloud部署 安装手册V2.0



谷云科技(广州)有限责任公司
<http://www.restcloud.cn>

修订历史记录

版本	日期	AMD	修订者	说明
2.0	2020-05-20	A	RestCloud	新增文档

A-添加, M-修改, D-删除)

目 录

安装准备

Linux环境部署

Windows环境部署

安装准备

JDK包

jdk下载地址：

<https://www.oracle.com/java/technologies/javase/javase8-archive-downloads.html>

注意根据系统实际情况选择 以下是操作系统均为X64为例

Java SE Development Kit 8u201		
This software is licensed under the Oracle Binary Code License Agreement for Java SE Platform Products		
Product / File Description	File Size	Download
Linux ARM v6/v7 Soft Float ABI	72.98 MB	jdk-8u201-linux-arm32-vfp-hflt.tar.gz
Linux ARM v6/v7 Soft Float ABI	69.92 MB	jdk-8u201-linux-arm64-vfp-hflt.tar.gz
Linux x86	170.98 MB	jdk-8u201-linux-i586.rpm
Linux x86	185.77 MB	jdk-8u201-linux-i586.tar.gz
Linux x64	168.05 MB	jdk-8u201-linux-x64.rpm
Linux x64	182.93 MB	jdk-8u201-linux-x64.tar.gz
Mac OS X x64	245.92 MB	jdk-8u201-macosx-x64.dmg
Solaris SPARC 64-bit (SVR4 package)	125.33 MB	jdk-8u201-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	88.31 MB	jdk-8u201-solaris-sparcv9.tar.gz
Solaris x64 (SVR4 package)	133.99 MB	jdk-8u201-solaris-x64.tar.Z
Solaris x64	92.16 MB	jdk-8u201-solaris-x64.tar.gz
Windows x86	197.66 MB	jdk-8u201-windows-i586.exe
Windows x64	207.46 MB	jdk-8u201-windows-x64.exe

Tomcat

Tomcat下载：<https://tomcat.apache.org/download-80.cgi>

Apache Tomcat®

Search... GO

APACHECON

Save the date!

Apache Tomcat

- Home
- Taglibs
- Maven Plugin

Download

- Which version?
- Tomcat 10.0 (alpha)
- Tomcat 9
- Tomcat 8
- Tomcat 7
- Tomcat Connectors
- Tomcat Native
- Taglibs
- Archives

Documentation

- Tomcat 10.0 (alpha)
- Tomcat 9.0
- Tomcat 8.5
- Tomcat 7.0
- Tomcat Connectors
- Tomcat Native
- Wiki
- Migration Guide
- Presentations

Problems?

- Security Reports
- Find help
- FAQ
- Mailing Lists
- Bug Database
- IRC

Get Involved

- Overview
- Source code
- Buildbot
- Translations
- Tools

Media

- Twitter
- YouTube
- Blog

Tomcat 8 Software Downloads

Welcome to the Apache Tomcat® 8.x software download page. This page provides download links for obtaining the latest versions of Tomcat 8.x software, as well as information on previous versions. Unsure which version you need? Specification versions implemented, minimum Java version required and lots more useful information may be found on the [Tomcat 8.x page](#).

Quick Navigation

[KEYS](#) | [8.5.59](#) | [Browse](#) | [Archives](#)

Release Integrity

You **must** [verify](#) the integrity of the downloaded files. We provide OpenPGP signatures for every release file. This signature should be matched against the [KEYS](#) file. SHA-512 checksums for every release file. After you download the file, you should calculate a checksum for your download, and make sure it is the same as ours.

Mirrors

You are currently using <https://downloads.apache.org/>. If you encounter a problem with this mirror, please select another mirror. If all mirrors are failing, then please contact the [Apache Tomcat Project](#).

Other mirrors: <https://downloads.apache.org/> | [Change](#)

8.5.59

Please see the [README](#) file for packaging information. It explains what every distribution contains.

Binary Distributions

- Core:
 - [tar.gz \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
 - [32-bit Windows.zip \(pgp, sha512\)](#)
 - [64-bit Windows.zip \(pgp, sha512\)](#)
 - [32-bit/64-bit Windows Service Installer \(pgp, sha512\)](#)
- Full documentation:
 - [tar.gz \(pgp, sha512\)](#)
- Deployer:
 - [zip \(pgp, sha512\)](#)
 - [tar.gz \(pgp, sha512\)](#)
- Extras:
 - [JMX Remote jar \(pgp, sha512\)](#)
 - [Web services jar \(pgp, sha512\)](#)
- Embedded:
 - [tar.gz \(pgp, sha512\)](#)
 - [zip \(pgp, sha512\)](#)

Source Code Distributions

- [tar.gz \(pgp, sha512\)](#)
- [zip \(pgp, sha512\)](#)

MongoDB

下载地址：<https://www.mongodb.com/try/download/community>

← → 🔍 mongodb.com/try/download/community

Join us for the MongoDB live series beginning November 10!

mongodb Cloud Software Pricing Learn Solutions Docs

Contact Sign In Try Free

Choose which type of deployment is best for you

- Cloud
The easiest way to run MongoDB
- On-Premises
Download on your own infrastructure
- Tools
Do more with your database

MongoDB Enterprise Server

MongoDB Community Server

MongoDB offers both an Enterprise and Community version of its powerful distributed document database. The community version offers the flexible document model along with ad hoc queries, indexing, and real time aggregation to provide powerful ways to access and analyze your data. As a distributed system you get high availability through built-in replication and failover along with horizontal scalability with native sharding.

The MongoDB Enterprise Server gives you all of this and more. Review the Enterprise Server tab to learn what else is available.

Available Downloads

Version: 3.6.0

Platform: Redhat / CentOS 7.0

Package: rpm

[Download](#) [Copy Link](#)

Current releases & packages
Development releases
Archived releases
Changelog
Release Notes

Linux环境部署

1. 安装步骤

1. 安装JDK
2. 安装tomcat
3. 安装ROOT.war工程包
4. 更换序列号
5. 启动tomcat
6. 访问 <http://localhost:8080/restcloud/admin>

2. 本地文件上传工具包安装

安装lrzsz包用来上传本地安装包文件

```
1 sudo yum install lrzsz -y
2 //创建一个目录用于存放所有安装文件,上传jdk, tomcat, 以及Restcloud工程war包
3 mkdir -p /usr/restcloud
```

Bash 复制代码

3. JDK安装

注意：JDK 必须是1.8.X 以上版本。以下与 JAVA 相关的配置项中版本为jdk1.8.0_201，JDK版本如果使用的不是该版本，需注意修改对应内容

```
1 //创建 jdk 目录用来安装jdk
2 mkdir -p /usr/jdk
3 //进入存放安装文件的restcloud目录
4 cd /usr/restcloud
5 //解压jdk,文件名字根据上传的jdk版本号修改,可输入jdk之后按tab自动补充
6 tar -zxvf jdk-8u201-linux-x64.tar.gz
7 //查看解压后的jdk文件名
8 ls
9 //移动解压后的jdk到/usr/jdk目录下
10 mv jdk1.8.0_201/ /usr/jdk/
11 解压后配置jdk的环境变量
12 sudo vi /etc/profile
```

Bash 复制代码

在文本增加以下变量配置，文件编辑界面中按 **a** 进入编辑模式

```
1 JAVA_HOME=/usr/jdk/jdk1.8.0_201
2 PATH=$JAVA_HOME/bin:$PATH
3 CLASSPATH=.:$JAVA_HOME/lib/dt.jar:$JAVA_HOME/lib/tools.jar
4 export JAVA_HOME
5 export PATH
6 export CLASSPATH
```

Plain Text 复制代码

按ESC键 跳到命令模式，然后输入 **:wq** 保存并退出，输入 **:q!** 不保存退出

保存退出后输入以下命令，使环境变量立即生效

`source /etc/profile`

再输入 `java -version` 如果能显示版本号说明配置成功

```
[root@px-restcloud1 jdk1.8.0_201]# vi /etc/profile
[root@px-restcloud1 jdk1.8.0_201]# source /etc/profile
[root@px-restcloud1 jdk1.8.0_201]# java -version
java version "1.8.0_201"
Java(TM) SE Runtime Environment (build 1.8.0_201-b09)
Java HotSpot(TM) 64-Bit Server VM (build 25.201-b09, mixed mode)
[root@px-restcloud1 jdk1.8.0_201]# vi /etc/profile
[root@px-restcloud1 jdk1.8.0_201]#
```

4. Tomcat安装

注意：tomcat要8.5.X以上版本才可以，以下配置内容中版本号为8.5.37，tomcat的版本如果不是该版本，文档中与版本号相关的配置内容需注意修改对应内容

4.1 安装步骤

```

1 //进入存放安装文件的restcloud目录
2 cd /usr/restcloud
3 //解压tomcat,文件名字根据上传的tomcat版本号修改,可输入apache-tomcat之后按tab键自动补充
4 tar -zxvf apache-tomcat-8.5.37.tar.gz
5 //查看解压后的tomcat文件名
6 ls
7 //移动解压后的tomcat到/usr/tomcat目录下
8 mv apache-tomcat-8.5.37 /usr/tomcat
9 //配置tomcat环境
10 cd /usr/tomcat/bin/
11 vi startup.sh

```

在文本中增加以下配置，文件编辑界面中按a进入编辑模式

```

1 #set java environment
2 export JAVA_HOME=/usr/jdk/jdk1.8.0_201
3 export JRE_HOME=${JAVA_HOME}/jre
4 export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib
5 export PATH=${JAVA_HOME}/bin:$PATH
6
7 #tomcat
8 export TOMCAT_HOME=/usr/tomcat

```

按ESC键 跳到命令模式，然后输入 :wq 保存并退出，输入 :q! 不保存退出

保存退出后可以启动tomcat

sudo ./startup.sh //启动Tomcat

查看是否启动成功

sudo netstat -naptl

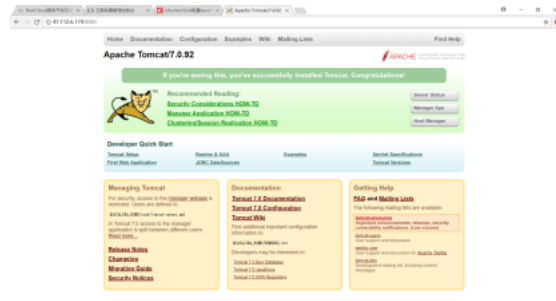
```

Active Internet connections (servers and established)
tcp        0      0 0.0.0.0:8080->0.0.0.0:* LISTEN 1565/java
tcp        0      0 0.0.0.0:8080->0.0.0.0:* LISTEN 1565/java
tcp        0      0 0.0.0.0:8080->0.0.0.0:* LISTEN 1565/java
tcp        0      0 0.0.0.0:8080->0.0.0.0:* LISTEN 1565/java
tcp        0      0 0.0.0.0:8080->0.0.0.0:* LISTEN 1565/java

```

表示在8080端口启动成功

输入 http://192.168.X.X:8080/可以显示tomcat的页面



如果不能访问可尝试开启端口

临时开启: iptables -I INPUT -p tcp --dport 8080 -j ACCEPT

永久开启:

firewall-cmd --zone=public --add-port=8080/tcp --permanent

firewall-cmd --reload

4.2 设置tomcat内存

编辑vim /usr/tomcat/bin/catalina.sh文件，加入如下配置

```

JAVA_OPTS="-server -Xms10240M -Xmx10240M -XX:PermSize=256M -
XX:MaxPermSize=256M"

```

注: -Xms: 初始化堆的内存大小(10G)

-Xmx: 最大堆内存大小(10G)

内存大小配置为机器内存的80%

一般只修改这两个值的大小，另外两个配置默认

查看机器内存

free

```
[root@px-restcloud1 ~]# free
              total        used        free       shared    buff/cache   available
Mem:         1767108      358708      1091284         16880         317116      1241956
Swap:          4124668           0         4124668
```

4.3 设置tomcat开机自启

Bash | 复制代码

```
1 1.修改脚本文件rc.local,这个脚本是使用者自定的开机启动程序,可以在里面添加想在系统启动之后执行的
2 脚本或者脚本执行命令
3
4 vim /etc/rc.d/rc.local
5
6 2.添加如下内容:
7
8 export JAVA_HOME=/usr/jdk/jdk1.8.0_201
9 /usr/tomcat/bin/startup.sh start
10
11 3.保存修改并退出
12
13 esc
14 :wq
15
16 4.将rc.local修改为可执行
17
18 chmod 777 /etc/rc.d/rc.local
19
20 5.重启机器测试
21
22 reboot
23
24 6.机器重启完成,重新连接服务器并查看tomcat是否自启,参考安装过程中的检查步骤
```

4.4 配置停止tomcat的同时杀掉进程

因项目中有非守护线程的存在,导致执行shutdown命令停止tomcat之后进程仍然存在

解决方案:

Bash | 复制代码

```
1 1、修改tomcat下bin/catalina.sh文件,找到PRGDIR=`dirname "$PRG"`这一行,在下面添加以下内容
2
3 if [ -z "$CATALINA_PID" ]; then
4     CATALINA_PID=$PRGDIR/CATALINA_PID
5 fi
6
7 2、修改tomcat的shutdown.sh文件,修改最后一行
8 修改前:
9 exec "$PRGDIR"/"$EXECUTABLE" stop "$@"
10 修改后:
11 exec "$PRGDIR"/"$EXECUTABLE" stop -force "$@"
```

5 MONGO安装

5.1 安装步骤

注意: 如果服务器的系统为CentOS 7.x, mongo推荐使用3.x版本, 如果服务器的系统为CentOS 8.x, mongo推荐使用4.x版本, 不然可能出现5.2中场景1的错误。

```

1 //创建文件夹/usr/restcloud目录统一存放mongo安装文件以及备份相关文件
2 mkdir -p /usr/restcloud
3 //上传mongo安装文件
4 cd /usr/restcloud
5 rz
6 //解压文件
7 tar -zxvf mongodb-linux-x86_64-rhel70-3.6.20.tgz
8 //查看解压后的文件
9 ls
10 //创建data目录。注意，mongodb的安装目录为/data/mongodb，/usr/restcloud只是用于存放安
    装文件
11 mkdir /data
12 //移动mongo安装目录到data下
13 mv /usr/restcloud/mongodb-linux-x86_64-rhel70-3.6.20 /data/mongodb
14 cd /data/mongodb/
15 //创建mongo数据文件目录
16 mkdir -p /data/mongodb/data/db
17 //创建mongo日志文件目录
18 mkdir -p /data/mongodb/logs
19 //创建mongo日志文件
20 cd /data/mongodb/logs/
21 touch mongodb.log
22 //创建mongo配置文件
23 cd /data/mongodb/bin/
24 touch mongodb.conf
25 //编辑配置文件
26 vi mongodb.conf

```

将以下设置复制到mongo配置文件中，文件编辑界面中按 **a** 进入编辑模式

```

1 bind_ip=0.0.0.0
2 port=27017
3 logappend=true
4 logpath=/data/mongodb/logs/mongodb.log
5 dbpath=/data/mongodb/data/db
6 fork=true
7 #auth=true
8 #其中auth=true表示启动后需要鉴权登陆，注释掉这行就不用鉴权登陆

```

按ESC键 跳到命令模式，然后输入 **:wq** 保存并退出，输入 **:q!** 不保存退出

启动与停止

```

1 //在MONGODB/BIN目录下
2 cd /data/mongodb/bin
3
4 //启动命令
5 ./mongod --config mongodb.conf
6
7 //链接命令
8 ///无认证
9 ./mongo
10
11 //开启认证之后
12 ./mongo -u admin -p
13
14 停止命令
15 ./mongod --shutdown --config mongodb.conf

```

创建用户

链接上mongo之后

切换到admin库

`use admin;`

创建用户，admin用户读写数据库，root用户管理数据库，运维工作中的备份等操作需使用root账号，注意密码复杂度


```

1 db.createUser(
2   {
3     user: "admin",
4     pwd: "pass",
5     roles:['readWriteAnyDatabase','dbAdminAnyDatabase']
6   }
7 );
8 db.createUser(
9   {
10    user: "root",
11    pwd: "root",
12    roles:['root']
13  }
14 );

```

创建后出现以下提示则创建成功

```

1 Successfully added user: {
2   "user" : "admin",
3   "roles" : [
4     "readWriteAnyDatabase",
5     "dbAdminAnyDatabase"
6   ]
7 }

```

验证用户

```
db.auth("admin","pass");
```

退出连接

```
exit
```

开启认证模式并重启mongo

在mongodb.conf中，修改配置文件，将auth一项的注释去掉

```

1 bind_ip=0.0.0.0
2 port=27017
3 logappend=true
4 logpath=/data/mongodb/logs/mongodb.log
5 dbpath=/data/mongodb/data/db
6 fork=true
7 auth=true
8 #其中auth=true表示启动后需要鉴权登陆，注释掉这行就不用鉴权登陆

```

重启mongo

在MONGODB/BIN目录下

```
cd /data/mongodb/bin
```

停止mongo

```
./mongod --shutdown --config mongodb.conf
```

启动mongo

```
./mongod --config mongodb.conf
```

查看是否启动成功

开启端口

查看防火墙状态

```
systemctl status firewalld
```

```

[root@px-restcloud1 ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2021-06-11 21:57:07 EDT; 39min ago
     Docs: man:firewalld(1)
    Main PID: 881 (firewalld)
      Tasks: 2 (limit: 10796)
     Memory: 30.0M
    CGroup: /system.slice/firewalld.service
            └─881 /usr/libexec/platform-python -s /usr/sbin/firewalld --nofork --nopid

Jun 11 21:57:05 px-restcloud1 systemd[1]: Starting firewalld - dynamic firewall daemon...
Jun 11 21:57:07 px-restcloud1 systemd[1]: Started firewalld - dynamic firewall daemon.
Jun 11 21:57:08 px-restcloud1 firewalld[881]: WARNING: AllowZoneDrifting is enabled. This is considered an insecure configuration option. It will be removed in a future release. Please consider disabling it now.

```

如果防火墙不是图中的启动状态，执行启动命令

```
systemctl start firewalld
```

开启端口

```
firewall-cmd --zone=public --add-port=27017/tcp --permanent
```

重新加载配置

firewall-cmd --reload

5.2 错误解决

场景1: 启动mongo时报错

```
[root@epishujv-cs bin]# ./mongod --config mongod.conf
./mongod: error while loading shared libraries: libcrypto.so.1.1: cannot open shared object file: No such file or directory
```

先检查缺失的共享库

ldd /data/mongodb/bin/mongod

```
[root@epishujv-cs bin]# ldd /usr/mongodb/bin/mongod
/usr/mongodb/bin/mongod: /lib64/libc.so.6: version 'GLIBC_2.18' not found (required by /usr/mongodb/bin/mongod)
linux-vdso.so.1 => (0x00007ffdf7680000)
libcurl.so.4 => /lib64/libcurl.so.4 (0x00007f5a136d3000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f5a133e9000)
libcrypto.so.1.1 => not found
libssl.so.1.1 => not found
libdat.so.2 => /lib64/libdat.so.2 (0x00007f5a131e5000)
librt.so.1 => /lib64/librt.so.1 (0x00007f5a12fd9000)
libm.so.6 => /lib64/libm.so.6 (0x00007f5a12cd0000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f5a12ac5000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f5a128a9000)
libc.so.6 => /lib64/libc.so.6 (0x00007f5a124db000)
/lib64/ld-linux-x86-64.so.2 (0x00007f5a1751a000)
libldm.so.11 => /lib64/libldm.so.11 (0x00007f5a122a8000)
libssh2.so.1 => /lib64/libssh2.so.1 (0x00007f5a1207b000)
libssl3.so => /lib64/libssl3.so (0x00007f5a11e22000)
libsmime3.so => /lib64/libsmime3.so (0x00007f5a11bfa000)
libnss3.so => /lib64/libnss3.so (0x00007f5a118cb000)
libnssutil3.so => /lib64/libnssutil3.so (0x00007f5a1169e000)
libldb4.so => /lib64/libldb4.so (0x00007f5a11407000)
libplc4.so => /lib64/libplc4.so (0x00007f5a11292000)
libnspr4.so => /lib64/libnspr4.so (0x00007f5a11054000)
libgssapi_krb5.so.2 => /lib64/libgssapi_krb5.so.2 (0x00007f5a10e07000)
libkrb5.so.3 => /lib64/libkrb5.so.3 (0x00007f5a10b1e000)
libk5crypto.so.3 => /lib64/libk5crypto.so.3 (0x00007f5a108eb000)
libcom_err.so.2 => /lib64/libcom_err.so.2 (0x00007f5a106e7000)
liblber-2.4.so.2 => /lib64/liblber-2.4.so.2 (0x00007f5a104d0000)
libldap-2.4.so.2 => /lib64/libldap-2.4.so.2 (0x00007f5a10283000)
libz.so.1 => /lib64/libz.so.1 (0x00007f5a1006d000)
```

安装GLIBC_2.18

进入/usr/lib目录

cd /usr/lib

下载: wget <http://ftp.gnu.org/gnu/glibc/glibc-2.18.tar.gz>

1. 下载完压缩包之后, 解压压缩包: tar -xzf glibc-2.18.tar.gz
2. 进入目录: cd glibc-2.18
3. 创建build目录并进入该目录: mkdir -p build; cd build
4. 运行configure命令: ./configure --prefix=/usr --disable-profile --enable-add-ons --with-headers=/usr/include --with-binutils=/usr/bin
5. 运行make
6. 运行make install
7. 运行strings /lib64/libc.so.6 | grep GLIBC

```
[root@epishujv-cs bin]# strings /lib64/libc.so.6 | grep GLIBC
GLIBC_2.2.5
GLIBC_2.2.6
GLIBC_2.3
GLIBC_2.3.2
GLIBC_2.3.3
GLIBC_2.3.4
GLIBC_2.4
GLIBC_2.5
GLIBC_2.6
GLIBC_2.7
GLIBC_2.8
GLIBC_2.9
GLIBC_2.10
GLIBC_2.11
GLIBC_2.12
GLIBC_2.13
GLIBC_2.14
GLIBC_2.15
GLIBC_2.16
GLIBC_2.17
GLIBC_2.18
GLIBC_PRIVATE
```

解决libcrypto.so.1.1和libssl.so.1.1的not found问题

先检查系统中是否存在这两个依赖库

find / -name libcrypto*

find / -name libssl*

如果存在, 执行以下命令建立软链接

ln -s /usr/local/lib64/libcrypto.so.1.1 libcrypto.so.1.1在系统中的路径

ln -s /usr/local/lib64/libssl.so.1.1 libssl.so.1.1在系统中的路径

如果不存在, 下载安装

```

1 #从官网下载
2 wget https://www.openssl.org/source/openssl-1.1.1g.tar.gz
3 #腾讯云提供的镜像
4 wget https://mirrors.cloud.tencent.com/openssl/source/openssl-1.1.1g.tar.gz
5
6
7 #编译openssl
8 tar -xvf openssl-1.1.1g.tar.gz
9 cd openssl-1.1.1g
10 ./config shared --openssldir=/usr/local/openssl --prefix=/usr/local/openssl
11 make && make install
12 #执行完成以后用openssl version查看当前版本号, 如果不是1.1, 需配置
13 echo "/usr/local/lib64/" >> /etc/ld.so.conf
14 ldconfig
15 #再次只用openssl version验证版本, 仍然不显示1.1, 尝试依次执行以下命令
16 mv /usr/bin/openssl /usr/bin/openssl.old
17 mv /usr/lib/openssl /usr/lib/openssl.old
18 ln -s /usr/local/openssl/bin/openssl /usr/bin/openssl
19 ln -s /usr/local/openssl/include/openssl /usr/include/openssl
20 echo "/usr/local/openssl/lib" >> /etc/ld.so.conf
21 ldconfig -v

```

再次执行ldd /data/mongodb/bin/mongod, 没有缺少的共享库, 说明安装成功

```

[root@pishujv-cs bin]# ldd /usr/mongodb/bin/mongod
linux-vdso.so.1 (0x00007f6d73f60000)
libcurl.so.4 => /lib64/libcurl.so.4 (0x00007f6b7af42000)
libresolv.so.2 => /lib64/libresolv.so.2 (0x00007f6b7adb6000)
libcrypto.so.1.1 => /usr/local/openssl/lib/libcrypto.so.1.1 (0x00007f6b7ad20000)
libssl.so.1.1 => /usr/local/openssl/lib/libssl.so.1.1 (0x00007f6b7ad40000)
libdl.so.2 => /lib64/libdl.so.2 (0x00007f6b7a43c000)
librt.so.1 => /lib64/librt.so.1 (0x00007f6b7a234000)
libm.so.6 => /lib64/libm.so.6 (0x00007f6b79f22000)
libgcc_s.so.1 => /lib64/libgcc_s.so.1 (0x00007f6b79d1c000)
libpthread.so.0 => /lib64/libpthread.so.0 (0x00007f6b79afe000)
libc.so.6 => /lib64/libc.so.6 (0x00007f6b79752000)
/lib64/ld-linux-x86_64.so.2 (0x00007f6b792c0000)
libidn.so.11 => /lib64/libidn.so.11 (0x00007f6b7951f000)
libssh2.so.1 => /lib64/libssh2.so.1 (0x00007f6b792f2000)
libssl12.so => /lib64/libssl12.so (0x00007f6b79059000)
libsqlite3.so => /lib64/libsqlite3.so (0x00007f6b78e21000)
libnss3.so => /lib64/libnss3.so (0x00007f6b78b42000)
libnssutil3.so => /lib64/libnssutil3.so (0x00007f6b78912000)
libpld4.so => /lib64/libpld4.so (0x00007f6b7870e000)
libplc4.so => /lib64/libplc4.so (0x00007f6b78509000)
libnspr4.so => /lib64/libnspr4.so (0x00007f6b782cb000)
libgssapi_krb5.so.2 => /lib64/libgssapi_krb5.so.2 (0x00007f6b7807e000)
libkrb5.so.3 => /lib64/libkrb5.so.3 (0x00007f6b77d00000)
libk5crypto.so.3 => /lib64/libk5crypto.so.3 (0x00007f6b77b62000)
libcom_err.so.2 => /lib64/libcom_err.so.2 (0x00007f6b7795e000)
liblber-2.4.so.2 => /lib64/liblber-2.4.so.2 (0x00007f6b7774f000)
libldap-2.4.so.2 => /lib64/libldap-2.4.so.2 (0x00007f6b774fa000)
libz.so.1 => /lib64/libz.so.1 (0x00007f6b772e4000)
libssl10.so => /lib64/libssl10.so.10 (0x00007f6b77072000)
libcrypto.so.10 => /lib64/libcrypto.so.10 (0x00007f6b76c8f000)
libkrb5support.so.0 => /lib64/libkrb5support.so.0 (0x00007f6b769ff000)
libkeyutils.so.1 => /lib64/libkeyutils.so.1 (0x00007f6b76770000)

```

场景2：启动mongo提示权限不足

```

[root@px-mongo2 bin]# ./mongod --shutdown --config mongodb.conf
-bash: ./mongod: 权限不够
[root@px-mongo2 bin]# ./mongod --config mongodb.conf
-bash: ./mongod: 权限不够
[root@px-mongo2 bin]# sudo ./mongod --config mongodb.conf
sudo: ./mongod: 找不到命令
[root@px-mongo2 bin]# ll
总用量 277048
-rw-r--r--. 1 root root 10359081 6月 17 03:53 bsondump
-rw-r--r--. 1 root root 29860072 6月 17 03:53 mongo
-rw-r--r--. 1 root root 54387648 6月 17 03:53 mongod
-rw-r--r--. 1 root root 276 6月 17 03:55 mongodb.conf
-rw-r--r--. 1 root root 12696783 6月 17 03:53 mongodump
-rw-r--r--. 1 root root 10711297 6月 17 03:53 mongoexport
-rw-r--r--. 1 root root 10593233 6月 17 03:53 mongofiles
-rw-r--r--. 1 root root 10867956 6月 17 03:53 mongoimport
-rw-r--r--. 1 root root 10361065 6月 17 03:53 mongooplog
-rw-r--r--. 1 root root 53756680 6月 17 03:53 mongoperf
-rw-r--r--. 1 root root 14000016 6月 17 03:53 mongoreplay
-rw-r--r--. 1 root root 14054073 6月 17 03:53 mongorestore
-rw-r--r--. 1 root root 30523368 6月 17 03:53 mongos
-rw-r--r--. 1 root root 10931198 6月 17 03:53 mongostat
-rw-r--r--. 1 root root 10557955 6月 17 03:53 mongotop

```

解决方案：将文件赋予可执行权限

```
1 chmod 777 *
```

6. 安装RestCloud War工程包

```

1 //如果tomcat启动着, 先停止tomcat
2 cd /usr/tomcat/bin/
3 ./shutdown.sh
4 //清空tomcat工程目录
5 cd /usr/tomcat/webapps
6
7 //从/usr/restcloud目录中移动RestCloud工程包到tomcat工程目录下,使用cp命令和mv命令都一样, cp是复制, mv是移动
8 //工程名需改为ROOT.war
9 cp /usr/restcloud/restcloud-V4.5-20210531.war /usr/tomcat/webapps/ROOT.war
10 mv /usr/restcloud/restcloud-V4.5-20210531.war /usr/tomcat/webapps/ROOT.war
11
12 //进入tomcat工程目录文件
13 cd /usr/tomcat/webapps/
14 unzip -o ROOT.war -d ROOT
15 //进入工程包修改配置文件,配置内容参考 7. 配置application.properties文件
16 cd ROOT/WEB-INF/classes/
17 vim application.properties

```

7. 配置application.properties文件

进入到tomcat\webapps\ROOT\WEB-INF\classes目录下

修改application.properties文件

[#这个是mongodb认证库的名称、安装后注册用户时创建的](#)

spring.data.mongodb.authentication-database=admin

[#指定本系统配置数据存储的数据库名](#)

spring.data.mongodb.database=RestCloud_V45PUB

[#LOG库可以按日期进行存储分库,按月分库: ApiLog_{yyyy-MM},按周分库:ApiLot_{WEEK},按天分库:ApiLog_{yyyy-MM-dd}。部署时需确认分库的策略并进行响应的配置](#)

spring.data.mongodb.log.database=RC_ApiLog

[#这个是组织用户数据库, 多机器时需要关联用户信息就需要配置相同数据库\) \(集群配置\)](#)

spring.data.mongodb.org.database=RestCloud_Org

[#集群服务器之间同步用的公共配置数据库,空表示和配置库一起 \(集群配置\)](#)

spring.data.mongodb.public.database=RC_PublicConfig

[#数据库连接配置](#)

spring.data.mongodb.host=127.0.0.1:27017

spring.data.mongodb.repositories.enabled=true

spring.data.mongodb.username=admin

spring.data.mongodb.password=pass@2021

[#生产环境标识改为true时性能更优, 极大影响性能](#)

restcloud.production.environment=false

[#配置API日志文件存储的方式,mongo表示直接存储到mongo中,file表示先存储到本地log文件中, maxtime表示多久时间写入一次0表示实时。正式环境中设为file, 写入时间设为30秒, 减少与mongo的交互, 可以大幅提升性能](#)

restcloud.apilog.writer.type=file

restcloud.apilog.writer.cache.maxtime=30

restcloud.apilog.writer.debug=true

[#本服务在注册中心中出现的ServiceName,多个集群服务器应具有相同的ServiceName \(集群配置\)](#)

restcloud.discovery.current.servicename=API01

[#指定当前服务器的IP地址, 如果不指定则系统会自动计算出一个IP](#)

restcloud.discovery.current.server.ip=127.0.0.1

[#服务器的集群标识\(同一集群的服务器集群标识要一样\), 比如网关的标识统一叫做GATEWAY](#)

restcloud.CurrentServerClusterFlag=API01

[#当前服务器的唯一Id,每个服务必须要唯一,由字母或数字组成 \(集群配置\)](#)

restcloud.CurrentServerId=API01

[#SN序列号必须指定一个SN序列号, 如果此处为空请联系我们如果已经有值不用再替换](#)

restcloud.sn=

8. 启动RestCloud服务

8.1. 普通方式启动

配置完application.properties后就可以启动tomcat了

注意:系统第一次启动时会自动初始化一个数据库到mongodb中, 在tomcat控制台日记中可以看到数据初始化的提示信息。如果初始化失败可以进入mongo里将相关的库删除, 重新启动会自动创建数据库

```
1 //配置完成, 进入tomcat的bin目录启动tomcat
2 cd /usr/tomcat/bin
3 //启动tomcat并打印启动日志
4 ./startup.sh & tail -f ../logs/catalina.out
```

Bash | 复制代码

输入<http://192.168.X.X:8080/restcloud/admin> 地址

出现登录界面输入用户名:admin 密码:pass 即可以进入系统界面



启动成功后的界面

8.2. 服务方式启动（开机启动）

<https://blog.csdn.net/wangli61289/article/details/37924785>

在/etc/init.d目录下新建文件, 命名为tomcat

对tomcat文件进行编辑, 执行

```
# cd /etc/init.d/
# vi tomcat
```

将下面代码粘上去

```

1  #!/bin/bash
2  # description: Tomcat7 Start Stop Restart
3  # processname: tomcat7
4  # chkconfig: 234 20 80
5
6  export JAVA_HOME=/usr/jdk/jdk1.8.0_201
7  export JRE_HOME=/usr/jdk/jdk1.8.0_201
8  export CATALINA_BASE=/usr/tomcat
9  export CATALINA_HOME=/usr/tomcat
10
11 case $1 in
12     start)
13         sh $CATALINA_HOME/bin/startup.sh
14         ;;
15     stop)
16         sh $CATALINA_HOME/bin/shutdown.sh
17         ;;
18     restart)
19         sh $CATALINA_HOME/bin/shutdown.sh
20         sh $CATALINA_HOME/bin/startup.sh
21         ;;
22     esac
23     exit 0

```

Bash 复制代码

3. 按ESC退出，并#: wq
4. 设置tomcat的文件属性，把tomcat 修改为可运行的文件，命令参考如下
#chmod a+x tomcat

5. 设置服务运行级别
#chkconfig --add tomcat

6. 服务就添加成功了
然后用 chkconfig --list 查看，在服务列表里就会出现自定义的服务了
chkconfig --list

7. 测试

```

service tomcat start
service tomcat stop
service tomcat restart
service tomcat status

```

9. 更换序列号

注意:如果安装包中的application.properties文件中的最后一行的SN已经有配置序列号则说明已经正带SN，如果没有请联系我们获取一个SN并填写才能启动。

9.1. 平台界面替换

系统恢复后都自带有序列号，如果需要列换序列号及版本信息需要进入到系统设置=>序列号填写界面

序列号填写

序列号

4f5c3a6e120e08f15c10b12206676d03832aee1b815824b183a46b548f916a9b5a225184159161908b1882267f66a0935ba7c9b0a6c2b078a371a05c3a9b0e025a052532a09f7a5a19a2c037776b2a7a7a09416716a77a6966c2c096a09107d09b095358a355e37c2709b74343b7a791954c38822b0a67795043c0b4174447837b0834c1f1914ac705a42c708a7964cc983b2231a07d18a37f1d516d8f190a6755642b08a7791b4d78a0f87b135a6b71a0a63242b53220aee55421aee0915b54b050908b04011a531a060ca0b2a012390583020441a0d070cb081d594643374b4b1a0b0d471722887c033a2205d049477a2c271a57a16541794a0a79b0a0a708a19b2a0c17a0777b01a0f00a02a0e0e4001234b0d0a0e0c2a0f

请输入本系统的序列号，请严格按照格式

属性名称: /C:/java/apache-tomcat-8.5.37/webapps/ROOT/WEB-INF/classes/application.properties

请指定要修改的属性文件(application.properties)路径

保存

即使是序列号已经过期也是可以登录到界面进行更换序列号的

更换为新的序列号即可，一般序列号可以在安装包中找到，如果没有请联系RestCloud获取，
同时要注意如果application.properties文件中最后一行的SN有序列号的情况下，需要同时替换，如果不替换则tomcat重启后会被初始化application.properties文件中的SN

9.2. MONGODB数据库替换

在mongoDB安装路径bin下登陆数据库

```
./mongo
```

```
use admin
```

```
db.auth('admin','pass@2021')
```

```
db.P_CoreSNConfig.update({"sn":"旧序列号"},{$set:{"sn":"新序列号"}})
```

update方法中第一参数为旧序列号，第二参数为新序列号，前面必须加\$set方法

9.3. 后端代码配置文件替换

修改\classes\ application.properties配置文件中restcloud.sn的参数

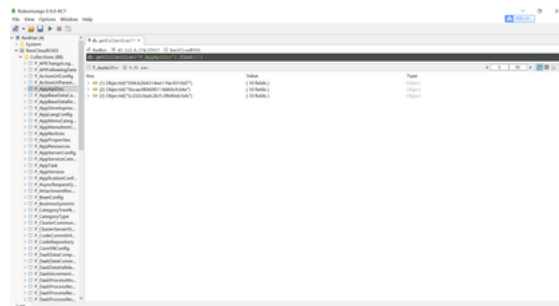
9.4. 自动初始化数据库并启动系统

首先确认解压的ROOT目录的web-inf/classes目录下是否有initdata/all目录，如果有可以用此方法初始化数据包，如果没有请使用第9步的数据库恢复方式恢复数据库。

如果有：initdata目录的情况下，可以直接启动tomcat即可完成系统数据库的初始化并同时启动成功。

注意：如果有此目录但是系统没有提示进行初始化则需要修改 application.proerties 文件中的 .
restcloud.StopInitInstallData=true 为false

一旦数据完成初始化后需要把application.proerties中的配置项restcloud.StopInitInstallData=false改为true,否则每次记动都会进行数据初始化。



恢复成功后用客户端工具可以看到RestCloud的数据库并能看到相应的数据说明恢复成功。

10. 安装后启动异常

10.1. java.net.UnknownHostException

问题原因：

在linux下启动项目 报出 java.net.UnknownHostException

查看log异常引发的代码是InetAddress.getLocalHost() 方法

修复：

修改 vi /etc/hosts 下的文件加入一行

本机ip LUNIX主机名

解析：

主机名: 机器部署后会生成一个主机名，登录后就可以看到，hostname可以查到，在你java抛出的异常中也报告了

10.2. The system has not been initialized yet序列号过期

原因：

The system has not been initialized yet.

解决：

在application.properties文件中更新序列号restcloud.sn=，没有SN请联系管理员

Windows环境部署

1. 安装步骤

RestCloud的安装主要步骤如下：

- Ø 首先安装JDK1.8版本
- Ø 安装tomcat部署产品war文件
- Ø 配置application.properties中的mongodb选项链接mongodb数据库
- Ø 启动tomcat系统自动进行数据初始化并启动系统

2. JDK1.8安装

Windows系统需要至少安装jdk-8u201以上版本可以在sun公司的网站上下载

安装成功后在cmd运行窗口中输入 `java -version` 可以看到安装成功的版本表示安装成功，否则表示jdk安装失败或者没有配置javahome的环境变量。

3. Tomcat8.5版本安装

Tomcat 需要安装 8.5.37 以上版本的 tomcat 服务器，tomcat 可以下载 `apache-tomcat-8.5.37-windows-x64.zip` 然后直接解压后找到 `webapps`目录下，进去后删除掉现有文件夹，然后解压 RestCloud.war包，解压后文件夹名称一定要用ROOT目录，不支持其他目录。

解压后可以看到：`/usr/tomcat/webapps/ROOT/WEB-INF` 目录表示正确

设置tomcat内存

修改bin/catalina.bat文件，加入如下配置

```
set JAVA_OPTS=-server -Xms10240m -Xmx10240m -XX:MaxNewSize=256m -XX:MaxPermSize=256m
```

注：-Xms：初始化堆的内存大小(10G)

-Xmx：最大堆内存大小(10G)

一般只修改这两个值的大小，另外两个配置默认

4. RestCloud链接MongoDB

进入到`/usr/tomcat/webapps/ROOT/WEB-INF/classes`目录下

修改application.properties文件

#这个是mongodb认证库的名称，安装后注册用户时创建的

`spring.data.mongodb.authentication-database=admin`

#这个是mongodb中系统表存储的库名

`spring.data.mongodb.database=RestCloudV4`

#LOG库可以按日期进行存储分库,按月分库: ApiLog_{yyyy-MM},按周分库:ApiLot_{WEEK},按天分库:ApiLog_{yyyy-MM-dd}，正式环境需要推荐按天分库

`spring.data.mongodb.log.database=RC_ApiLog`

#这个是组织用户数据库，多机器时需要关联用户信息就需要配置相同数据库

`spring.data.mongodb.org.database=`

`spring.data.mongodb.host=127.0.0.1:27017`

`spring.data.mongodb.repositories.enabled=true`

`spring.data.mongodb.username=userid`

`spring.data.mongodb.password=password`

#本服务在注册中心中出现的ServiceName,多个集群服务器应具有相同的ServiceName
restcloud.discovery.current.servicename=RC4.0
#指定当前服务器的IP地址, 如果不指定则系统会自动计算出一个IP
restcloud.discovery.current.server.ip=127.1.1.1
#当前服务器的唯一Id,每个服务必须要唯一,由字母或数字组成
restcloud.CurrentServerId=RC4.0_123
#SN序列号必须指定一个SN序列号, 如果此处为空请联系我们如果已经有值不用再替换
restcloud.sn=

5.启动tomcat测试运行

配置完application.properties后就可以启动tomcat了

输入<http://192.168.X.X:8080/restcloud/admin> 地址
出现登录界面输入用户名:admin 密码:pass 即可以进入系统界面



启动成功后的界面