



链滴

[博客] 在阿里云上部署 solo 博客

作者: [lin772662623](https://github.com/lin772662623)

原文链接: <https://ld246.com/article/1530110628374>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)



我的博客之前世今生

在给我爸做完卖单车的网站后，我爸说：“单车不好卖了，我决定干点别的。”

然后，我做的网站就荒废了。

偶然的的机会，看到用上 solo 的博客，开源还漂亮，而且还是java写的，好，就你了。

1.服务器

之前就用的阿里的服务器，那我的solo也直接和卖单车的部署在一块好了。

2.环境

apache-tomcat-7.0.73 - 需要升级到tomcat9.0.7

nginx-1.10.2

mysql5.1 -需要升级到5.6，不然不支持utf8mb4字符集

2.1 升级tomcat 9.0.7

这个就很简单了，直接从官网下载就可以了，sftp上传到服务器，解压，重新部署。贴下我用的部署本

```
echo "=====进入git项目bikemall目录======"
cd /developer/git-repository/bikemall
```

```
echo "=====git切换分之到v1.0======"
git checkout v1.0
```

```

echo "=====git fetch=====
git fetch

echo "=====git pull=====
git pull

echo "=====编译并跳过单元测试=====
mvn clean package -Dmaven.test.skip=true

echo "=====删除旧的ROOT.war=====
#rm /developer/apache-tomcat-7.0.73/webapps/ROOT.war
rm /developer/apache-tomcat-9.0.7/webapps/ROOT.war

echo "=====拷贝编译出来的war包到tomcat下-ROOT.war=====
#cp /developer/git-repository/bikemall/target/BikeMall.war /developer/apache-tomcat-7.0.7/
/webapps/ROOT.war
cp /developer/git-repository/bikemall/target/BikeMall.war /developer/apache-tomcat-9.0.7/
ebapps/ROOT.war

echo "=====删除tomcat下旧的ROOT文件夹=====
#rm -rf /developer/apache-tomcat-7.0.73/webapps/ROOT
rm -rf /developer/apache-tomcat-9.0.7/webapps/ROOT

echo "=====关闭tomcat=====
#/developer/apache-tomcat-7.0.73/bin/shutdown.sh
/developer/apache-tomcat-9.0.7/bin/shutdown.sh

echo "=====sleep 10s=====
for i in {1..10}
do
    echo "$i"s"
    sleep 1s
done

echo "=====启动tomcat=====
#/developer/apache-tomcat-7.0.73/bin/startup.sh
/developer/apache-tomcat-9.0.7/bin/startup.sh

```

因为这个脚本，我只需要在同样的目录解压这个tomcat9.0.7，再改一下脚本就可以了。这样卖单车不会受到影响了。☺

2.2 升级mysql

2.2.1 备份数据

因为我的网站刚做完就荒废了，什么数据都没有，所以也无需要备份了，但是mysql的配置需要备份下。[/etc/my.cnf](#)

直接开干咯。☺laughing

2.2.2 停止服务

```
sudo service mysqld stop
```

2.2.3 卸载原版本

```
sudo yum remove mysql mysql-*
```

执行卸载之后，检查一下是否卸载完成

```
sudo yum list installed | grep mysql
```

好了，检查没有任何mysql相关的东西了。

2.2.4 安装

在网上找到一种快速的方式，如果使用安装包安装，会提示需要依赖glibc更高版本，当你安装了更高本之后，会提示依赖其它，比较麻烦，直接用下面方法即可快速安装：

```
rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -Uvh http://mirrors.neusoft.edu.cn/epel/6/i386/epel-release-6-8.noarch.rpm
rpm -Uvh http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.x86_64.rpm
rpm -Uvh http://dl.iuscommunity.org/pub/ius/stable/Redhat/6/x86_64/epel-release-6-5.noar
h.rpm
rpm -Uvh http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
```

安装依赖

```
yum --enablerepo=remi,remi-test info mysql mysql-server
```

安装mysql

```
yum --enablerepo=remi,remi-test install mysql mysql-server
```

配置执行升级

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

把之前的my.cnf替换掉现在/etc/my.cnf

2.2.5 启动检查

```
sudo service mysqld start
```

第一次启动比较慢，因为它会自己做初始化，耐心等待即可。

如果启动失败，提示 “MySQL Daemon failed to start” ，可以尝试先初始化mysql，输入命令：

```
mysqld --initialize。
```

如果初始化提示：

```
[Warning] TIMESTAMP with implicit DEFAULT value is deprecated. Please use --explicit_defaul
s_for_timestamp server option (see documentation for more details).
```

```
[ERROR] --initialize specified but the data directory has files in it. Aborting.
```

```
[ERROR] Aborting
```

说明数据文件夹已被创建，直接删除，让mysql自己创建即可，输入命令：

```
rm -rf /var/lib/mysql
```

删除后，注意：然后不需要再初始化了，直接启动mysql，它会自己初始化，输入命令：

```
sudo service mysqld start
```

2.2.6 设置开机自动启动

将mysql设置为随系统自动启动

```
sudo chkconfig mysqld on
```

```
sudo chkconfig --list mysqld
```

第二到第五个都是on则代表配置完成。

等级0表示：表示关机

等级1表示：单用户模式

等级2表示：无网络连接的多用户命令行模式

等级3表示：有网络连接的多用户命令行模式

等级4表示：不可用

等级5表示：带图形界面的多用户模式

等级6表示：重新启动

2.2.7 建用户、建库

用root登陆mysql

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

新建用户solo

```
insert into mysql.user(Host, User, Password) values("localhost","solo",password("自己定密码"));
```

创建数据库

```
create database `solo` default character set utf8mb4 collate utf8mb4_general_ci;
```

进入数据库

```
use solo;
```

查看用户权限

```
select * from mysql.user \G
```

赋予权限，给solo用户赋予solo数据库所有表的所有权限

```
grant all privileges on solo.* to solo@'%' identified by '自己定的密码' with grant option;
```

```
select * from mysql.user \G
```

```
flush privileges;
```

3 开始部署solo

因为匆忙，也是因为懒，并不打算再修改solo代码了，所以直接从百度云下载war包下来，[war包](#)

3.1 上传solo

直接用sftp上传到阿里云就好了，我直接用的SecureCRT上传的，也可以用其它软件协议，只要能传OK。

3.2 tomcat配置

修改`apache-tomcat-9.0.7/conf/server.xml`，在Server标签中添加一个Service，如下：

```
<Service name="Catalina_blog">
  <Connector port="8081" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" />
  <!-- <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />-->

  <Engine name="Catalina_blog" defaultHost="localhost">

    <Realm className="org.apache.catalina.realm.LockOutRealm">
      <Realm className="org.apache.catalina.realm.UserDatabaseRealm"
        resourceName="UserDatabase"/>
    </Realm>

    <Host name="localhost" appBase="webapps_blog"
      unpackWARs="true" autoDeploy="true">

      <!--<Context path="" docBase="solo" reloadable="true" />-->

      <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
        prefix="localhost_access_log" suffix=".txt"
        pattern="%h %l %u %t &quot;%r&quot; %s %b" />

    </Host>
  </Engine>
</Service>
```

1. Service 标签的 name 属性值为 **Catalina_blog**
2. Connector 的 port 属性值为 **8081** (该端口号可以自由拟定)
3. Engine 的 name 属性值为 **Catalina_blog**
4. Host 的 appBase 属性值为 **webapps_blog**
5. Conext的docBase为solo

保存退出后，我们在apache-tomcat-9.0.7根目录下，新建`webapps_blog`目录。如下：

```
apache-tomcat-9.0.7/webapps_blog
```

把solo的war包放入这个目录中即可。

3.3 solo配置

修改solo配置有很多种方式，

- 一种是直接windows打开war包，在里面修改更新war包；
- 一种是直接运行tomcat，tomcat解压了solo的war包后，直接在解压的配置上修改；
- 一种是从github上下载源码，修改后再打成war包。

这里就不展开了。爱怎么弄怎么弄。反正是自己的服务器，是吧。☺
aughing

这里我们需要配置三个配置文件：

3.3.1 Latke配置剖析

/WEB-INF/classes/latke.properties (详细介绍: [Latke配置剖析](#))

主要看Server里的三个配置项

```
##### Server #####
# Browser visit protocol
serverScheme=http
# Browser visit domain name
serverHost=blog.linliqiang.top
# Browser visit port, 80 as usual, THIS IS NOT SERVER LISTEN PORT!
serverPort=
```

serverHost用域名

serverPort用http默认80端口即可，这里注意，如果你要用80端口的话，这一项是可以不填的。

3.3.2 local.properties

/WEB-INF/classes/local.properties

这个配置文件用于配置数据库，如果要使用MySQL的话请先手动建库（字符集使用utf8mb4，排列规则utf8mb4_general_ci），并注释掉H2相关属性同时打开MySQL相关属性。

```
##### H2 runtime #####
#runtimeDatabase=H2
#jdbc.username=root
#jdbc.password=
#jdbc.driver=org.h2.Driver
#jdbc.URL=jdbc:h2:~/solo_h2/db
#jdbc.pool=h2
```

```
##### MySQL runtime #####
runtimeDatabase=MYSQL
jdbc.username=solo
jdbc.password=自己定义的密码
jdbc.driver=com.mysql.jdbc.Driver
jdbc.URL=jdbc:mysql://localhost:3306/solo?useUnicode=yes&characterEncoding=utf8
jdbc.pool=druid
```

因为我用的是mysql，所以把H2相关的注释掉，打开mysql，并填自己的实际内容

3.3.3 mail.properties

这个我还没调过，我用的是公司的邮箱服务器，反正我没收到过邮件，不知道是哪里配置错了，这个正不要求的，这里就不介绍了，自己按意思填完就好了。

3.4 nginx配置

首先是在阿里云的域名管理里，添加blog.linliqiang.top的映射。

然后开始配置nginx,因为我之前已经在卖单车那里用过nginx了，这里就在这基础上加个配置就好了。

首先，我在/usr/local/nginx/conf/nginx.conf里加了以下内容，

```
include vhost/*.conf;
```

这样我每配置一项我就只需要在vhost文件夹里加个配置文件就行了。

因为我在前面的tomcat配置中，把solo的端口配置为8081了，所以nginx这里也需要配置为8081

```
upstream backend {
    server localhost:8081; # Tomcat/Jetty 监听端口
}

server {
    listen 80;
    server_name blog.linliqiang.top;

    access_log off;

    location / {
        proxy_pass http://backend$request_uri;
        proxy_set_header Host $host:$server_port;
        proxy_set_header X-Real-IP $remote_addr;
        client_max_body_size 10m;
    }
}
```

配置好之后，重启nginx就好了。

```
cd /usr/local/nginx/sbin
```

```
sudo ./nginx -t
```

```
sudo ./nginx -s reload
```

附录

1.solo用户指南

2.solo开发指南