



链滴

云服务器 1 折起，仅 89 元 / 年，限量抢购
， 5 分钟搭建博客

作者：[muyikun007](#)

原文链接：<https://ld246.com/article/1573280092428>

来源网站：[链滴](#)

许可协议：[署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

大家都知道最近阿里云服务器很便宜火爆，于是小编也入手了一台 3 年的，配置：1 核 2G 内存，40G 硬盘。

今年比去年便宜，**1 年 89 元，3 年 229 元**，

限新用户(如不是新用户,可以用家人的支付宝注册),打开以下链接参与: [抢购1折服务器](#)

买了不能闲置着，可以搭建自己的网站、博客、代码仓库等，用处广泛着呢！有很多朋友问怎么搭建网站，怎么用？应小伙伴们的需求，今天来学习一下最简单的网站搭建，只需要 5 分钟，就可以轻松搭！

可以浏览查看[我的博客](#)。



后台管理发布文章界面：



自带管理功能，使用 GitHub 账号登陆即可。Solo 是一款小而美 Java 编写的博客系统，功能丰富，件化，皮肤可选可定制，管理方便，社区活跃。

本文章介绍利用 Solo 开源博客系统在云服务器上搭建自己的博客，让我开始吧。

前提，把需要公网 IP 访问的端口，在阿里云控制台加到安全组里放行。

Docker 搭建

第一步就是安装 Docker

```
yum install docker.x86_64 -y
```

安装完成后启动 Docker

1. `systemctl start docker`

MySQL 镜像安装

Docker 镜像安装 MySQL 的好处：方便，容器间互不干扰，可以在同一主机上安装多个版本的 MySQL，只需要在主机上映射不同的端口即可。

安装 MySQL 镜像

```
docker search mysql #查找MySQL镜像版本
```

```
docker pull mysql:5.7 #安装指定版本的mysql镜像，5.7为版本号
```

启动 MySQL 容器

```
docker run --name test-mysql -e MYSQL_ROOT_PASSWORD=123456 -d -p 3307:3306 mysql:5.7
```

- `--name` 容器的别称
- `-p` 端口映射。格式是 主机的端口:容器的端口。
- `-e` 设置容器的环境变量。`-e MYSQL_ROOT_PASSWORD=123456` 就代表 MySQL 的 root 的密码是 123456
- `-d` 使用镜像包名称,可以通过 Docker images 查看
- `restart=always`, 告诉 Docker, 这个容器要自动启动

查看 MySQL 容器是否建立成功并启动:

```
[root@izbp1c741skofxsqfc2ftz ~]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
c4ef2dee75d1	centos_sshd:7.0	"/bin/bash"	5 days ago	Up 5 days	0.0.0.0:2222->22/tcp	centos_7.0-1
e83b09064c55	b3log/solo	"java -cp WEB-INF/..."	6 days ago	Up 6 days		solo
aafc0b472ea0	mysql:5.7	"docker-entrypoint..."	6 days ago	Up 6 days	33060/tcp, 0.0.0.0:3307->3306/tcp	test-mysql
bb40d63b3e34	nsqio/nsq	"/nsqadmin -looku..."	11 months ago	Exited (2) 11 months ago		nsqadmin
c6036892ca5e	nsqio/nsq	"/nsqd -broadcast..."	11 months ago	Exited (2) 11 months ago		nsqd
cd3b92735037	nsqio/nsq	"/nsqlookupd"	11 months ago	Exited (2) 11 months ago		lookupd
5ef931ab0f7f	centos	"/bin/bash"	11 months ago	Exited (137) 10 months ago		test-centos1
4552239f11c1	ubuntu:15.10	"/bin/bash"	12 months ago	Exited (0) 12 months ago		tender_kare
0ec81fe64690	ubuntu:15.10	"/bin/echo 'Hello ..."	12 months ago	Exited (0) 12 months ago		cranky_saha
1b1d44f7c1a8	linode/lamp	"/bin/bash"	12 months ago	Created		hopeful_saha
9a7ffa0a0c69	ubuntu:15.10	"/bin/bash"	17 months ago	Exited (100) 17 months ago		affectionate_snyder
4211eec5f7e7	httpd	"httpd -foreground"	17 months ago	Exited (0) 16 months ago		confident_swirles
d73128de9ba4	training/webapp	"python app.py"	17 months ago	Exited (137) 16 months ago		quirky_mayer
36e32aaaa364	ubuntu:15.10	"/bin/bash"	17 months ago	Exited (127) 17 months ago		epic_joliot
0ce16396a379	ubuntu:15.10	"/bin/bash"	17 months ago	Exited (100) 17 months ago		focused_albattani
d38f543a5897	ubuntu:15.10	"/bin/bash"	17 months ago	Exited (130) 17 months ago		practical_brown
lcaa4fab7f63	ubuntu:15.10	"/bin/sh -c 'while..."	17 months ago	Exited (137) 17 months ago		distracted_wright
8a052d0ca7c0	ubuntu:15.10	"/bin/bash"	17 months ago	Exited (0) 17 months ago		priceless_liskov
1860cf03b980	ubuntu:15.10	"/bin/echo 'Hello ..."	17 months ago	Exited (0) 17 months ago		stupefied_newton
ac10f90b3698	hello-world	"/hello"	17 months ago	Exited (0) 17 months ago		determined_brahmagupta

从 host 上连接上面启动的容器:

```
ifconfig #查看下docker虚拟出的ip地址
```

```
docker ps -a #查看下容器的运行状态
```

```
mysql -h {ip} -P {port} -u root -p #从host连接docker中的MySQL
```

如:

```
mysql -h172.17.0.1 -P3007 -u root -p123456
```

如果要用远程用 Navicat 连接 MySQL, 继续往下看:

以下配置允许远程用 honey 用户，通过密码 honey 连接数据库，且只允许操作名为 museum_of_art 的数据库。

创建 honey 用户，密码也为 honey:

```
create user'honey'@'% 'identified by'honey';
```

查看用户 honey 的权限:

```
show grants for 'honey'@'% ';
```

创建数据库，honey 用户只有操作 museum_of_art 数据库的权限:

```
create database museum_of_art;
```

允许用户 honey 操作 museum_of_art 表:

```
grant all on museum_of_art.* to'honey'@'% ';
```

刷新权限:

```
FLUSH PRIVILEGES;
```

如果开最大权限:

```
GRANT ALL PRIVILEGES ON *.* TO 'honey'@'localhost' IDENTIFIED BY 'honey';
```

```
GRANT ALL PRIVILEGES ON *.* TO 'honey'@'% ' IDENTIFIED BY 'honey';
```

阿里云服务器如果通过以上设置后远程 Navicat 还是连不上，请继续往下看。

现在如果在公网上用 Navicat 是链接不上的。

原因如下:

首先需要登陆阿里云后台，添加阿里云安全组策略 具体位置 -> 网络和安全 -> 安全组 -> 配置规则

网卡类型：	内网	
规则方向：	入方向	
授权策略：	允许	
协议类型：	MySQL (3306)	
* 端口范围：	3306/3306	?
优先级：	1	?
授权类型：	IPv4地址段访问	
* 授权对象：	0.0.0.0/0	? 教我设置
描述：		

长度为2-256个字符，不能以http://或https://开头。

可以选择多配置需要的端口。

下面我们需要配置阿里云防火墙

查看下防火墙的状态:

```
systemctl status firewalld
```

关闭防火墙:

```
systemctl stop firewalld
```

其实这样就可以使用了，但是这样很不安全，我们可以将 firewall 服务禁用，应用 iptables 服务（上大部分启用端口的资料都是基于 iptables 服务）。

安装 iptables

由于没有防火墙会造成不安全，所以给服务器安装一应用更广的防火墙 iptables，首先要禁用 firewal，通过 yum 安装 iptables:

```
systemctl disable firewalld  
yum install -y iptables-services
```

启动 iptables

```
systemctl start iptables
```

启动后可以通过 `systemctl status iptables` 查看状态。

更改 iptables 规则

将 iptables 文件备份下:

```
cp -a /etc/sysconfig/iptables /etc/sysconfig/iptables.bak
```

设置 INPUT 方向所有的请求都拒绝

```
iptables -P INPUT DROP
```

放开所需端口

```
iptables -I INPUT -p tcp --dport 3307 -m state --state NEW -j ACCEPT
```

保存规则

```
iptables-save > /etc/sysconfig/iptables
```

设置为开机启动并且重启

```
systemctl enable iptables.service  
systemctl reboot
```

好了, 系统到这里我们需要重新去启动 Docker

```
systemctl start docker #运行Docker守护进程
```

那么到这里我们外网就可以正常的去使用阿里云上的 MySQL 服务了。

比如用上面几行命令部署 MySQL, 把容器内 3306 端口映射到宿主机的 3307 端口。到时候就可以用阿里云公网 IP:3307 访问数据库, MySQL 部署好后, 先手动建库 (库名 Solo, 字符集使用 utf8mb, 排序规则 utf8mb4generalci)

启动 Solo 容器

然后启动容器

```
docker run --detach --name solo --network=host  
--env RUNTIME_DB="MYSQL"  
--env JDBC_USERNAME="root"  
--env JDBC_PASSWORD="password"  
--env JDBC_DRIVER="com.mysql.cj.jdbc.Driver"  
--env JDBC_URL="jdbc:mysql://47.91.6.217:3307/solo?useUnicode=yes&characterEncoding=  
TF-8&useSSL=false&serverTimezone=UTC"  
b3log/solo --listen_port=8080  
--server_scheme=http --server_host=47.98.192.142
```

- --detach 即-d 参数指定后台运行,
- --name 指定容器名称,
- --env 指定 Solo 系统运行数据库参数,
- --listen_port: 进程监听端口
- --server_scheme: 最终访问协议, 如果反代服务启用了 HTTPS 这里也需要改为 https

- --server_host: 最终访问域名或公网 IP, 不要带端口
- --server_port: 最终访问端口, 使用浏览器默认的 80 或者 443 的话值留空即可

使用的镜像是 b3log/solo 最新版, 这里比如, 47.98.192.142 是我阿里云公网 IP, 用 47.98.192.142 8080 访问:

