# 二十分钟搭建一个属于自己的个人博客

作者：K
原文链接：https://ld246.com/article/1601272304078
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)

# 正文

不拖拉，直接正文。

# 服务器

阿里云

"疼"讯云

十一将近，两大运营商会出活动。发这篇文章的时候 腾讯云 的活动 更优惠。

这里不费话，推荐去秒杀"疼"讯云

如果你是新用户购买的时候非常推荐你直接买个 3年 或者更长时间 的 ~ 因为新用户优惠非常力度非大，后面续费就没有这么大优惠了会比你买的时候贵上好几倍。

## 配置说明

如果只是用作个人博客，且用户量不是很大 1核 2G 足矣，如果你还想搭点什么的其它网站，可以酌升高配置与带宽。

- CPU 1核
- 内存 1G/2G
- 系统 CentOS(Linux)

# 准备

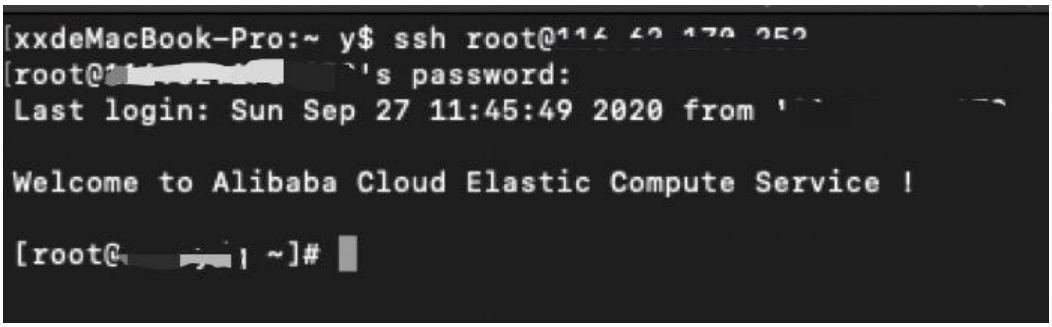
有了服务器就可以开始准备安装所需的软件了

- FTP 工具 例: FileZilla
- SSH 连接工具 例: putty

我这里是使用的 mac，自带了 ssh 命令，不过连接方式都是一样的。

# 安装过程

先使用ssh连接到服务器




ssh root@00.00.00.00 // @之后改为你服务器的ip
// 输入密码登录，密码为你开通 服务器实例时的密码，如果忘记了可以点重置

## 安装 docker

一键安装命令：

curl -sSL https://get.daocloud.io/docker | sh

等待安装完成，基本不会遇到什么问题, 安装完成之后 启动 docker 服务

sudo systemctl start docker

如果想了解更多可以 看这里

## 安装 mysql

启动之后 docker 命令就可以运行了

docker pull mysql:5.5.62

```
[[root@ ~]# docker pull mysql:5.5.62
5.5.62: Pulling from library/mysql
743f2d6c1f65: Already exists
3f0c413ee255: Already exists
aef1ef8f1aac: Already exists
f9ee573e34cb: Already exists
3f237e01f153: Already exists
03da1e065b16: Pull complete
04087a801070: Pull complete
7efd5395ab31: Pull complete
1b5cc03aaac8: Pull complete
2b7adaec9998: Pull complete
385b8f96a9ba: Pull complete
Digest: sha256:12da85ab88aedfdf39455872fb044f607c32fdc233cd59f1d26769fbf439b045
Status: Downloaded newer image for mysql:5.5.62
[root@r ~]#
```

docker images // 查看镜像
mysql 5.5.62 d404d78aa797 16 months ago 205MB // 会看到这个镜像

使用 ftp 工具上传 这个启动脚本

目录可以按自己喜好，推荐专门建一个目录来管理 docker容器
这里我的目录是 /docker/mysql

start.sh

```
#!/bin/bash

# mysql 启动脚本 start.sh

_pwd=$(cd `dirname $0`; pwd)

docker run \
    -p 3306:3306 --name mysql \
    -e innodb_buffer_pool_size=64M \
    -v $_pwd/conf:/etc/mysql/conf.d \
    -v $_pwd/logs:/logs \
    -v $_pwd/data:/var/lib/mysql \
    -e MYSQL_ROOT_PASSWORD=password \
    -d mysql:5.5.62
```

脚本内容需要按自己的需求配置 主要是 password

运行 start.sh 脚本

```
cd /docker/mysql
sh ./start.sh
```

如果你看到如下结果则运行下文命令即可

```
[root@all-ydq mysql]# sh start.sh
docker: Error response from daemon: Conflict. The container name "/mysql" is already in use by container "10f23eef7160c667e646d97956ffab8e296c974a4f5a6588
0448451a7e1ca2ec". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
```

```
docker start mysql
```

正常的启动结果是只输出一个

```
mysql
```

到这里mysql就安装并启动成功了

**创建 solo 数据库**

使用命令行，或者 mysql管理工具创建 solo库，字符集为 utf8mb4 utf8mb4_general_ci

## 安装 solo

依然是直接用 sh 脚本, 将这两个脚本上传到服务器

我的目录是 /docker/solo

拉取 solo镜像 repull.sh

```
#!/bin/bash

#
# solo docker 更新重启脚本
#
# 可将该脚本加入 crontab，每日凌晨运行来实现自动更新
#

docker pull b3log/solo
docker stop solo
docker rm solo

_pwd=$(cd `dirname $0`; pwd)

# 重新执行启动
sh $_work/start.sh
```

启动脚本 start.sh

```
#!/bin/bash

#
# 1. 请注意修改参数
#

_pwd=$(cd `dirname $0`; pwd)

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://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=
TF-8&useSSL=false&serverTimezone=UTC" \
    b3log/solo \
    --listen_port=8654 \
    --server_scheme=https \
    --server_host=blog.pilipala.co \
```

```
    --server_port=
```

注意修改 start.sh 当中的参数

执行

```
sh repull.sh
```

过后即可看到提示，让你去浏览器初始化数据 ~
打开你的 server_host 就可以登录 滴涟 进入管理了。

# 附件

附加几个自己用到的脚本

**注意 mysql 一定要常备份**

<details>
<summary>MySql start.sh</summary>

```
#!/bin/bash

# mysql 启动脚本 start.sh

_pwd=$(cd `dirname $0`; pwd)

docker run \
    -p 3306:3306 --name mysql \
    -e innodb_buffer_pool_size=64M \
    -v $_pwd/conf:/etc/mysql/conf.d \
    -v $_pwd/logs:/logs \
    -v $_pwd/data:/var/lib/mysql \
    -e MYSQL_ROOT_PASSWORD=password \
    -d mysql:5.5.62
```

</details>

<details>
<summary>solo 更新拉取脚本 repull.sh</summary>

```
#!/bin/bash

#
# solo docker 更新重启脚本
#
# 可将该脚本加入 crontab，每日凌晨运行来实现自动更新
#

docker pull b3log/solo
docker stop solo
docker rm solo
```

```
_pwd=$(cd `dirname $0`; pwd)

# 重新执行启动
sh $_work/start.sh
```

</details>

<details>
<summary>solo  运行脚本 start.sh</summary>

```
#!/bin/bash

#
# 1. 请注意修改参数
#

_pwd=$(cd `dirname $0`; pwd)

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://127.0.0.1:3306/solo?useUnicode=yes&characterEncoding=
TF-8&useSSL=false&serverTimezone=UTC" \
    b3log/solo \
    --listen_port=8654 \
    --server_scheme=https \
    --server_host=blog.pilipala.co \
    --server_port=
```

</details>

<details>
<summary>solo 重启脚本 restart.sh</summary>

```
#!/bin/bash

#
# docker 更新重启脚本
#

_pwd=$(cd `dirname $0`; pwd)

docker stop solo
docker rm solo
sh $_pwd/start.sh
```

</details>

<details>
<summary>清空无用的docker pull下来的 none 静像</summary>

#!/bin/bash

# 清除 docker 中更新镜像时，会多生成其它的 <none> 版本的镜像.

docker rmi $(docker images -f "dangling=true" -q)

</details