

# 开源软件安装部署 教程·集合

无字天书

([WuZiTianShu.com](http://WuZiTianShu.com))

# 目 录

致谢

[ThingsBoard安装教程 \(CentOS7.6安装/JDK1.8/ThingsBoard CE\)](#)

[Homeland Docker安装教程](#)

[Bookstack Linux\(Ubuntu\)下安装和部署](#)

[Ubuntu 18.0.4 安装 Docker CE 和 Docker compose](#)

# 致谢

当前文档《开源软件安装部署教程·集合》由扫地僧使用无字天书(wuzitianshu.com)进行构建，生成于2019-12-27。

无字天书网仅提供文档编写、整理、归类等功能，以及对文档内容的生成和导出工具。

文档内容由网友们编写和整理，无字天书网难以确认文档内容知识点是否错漏。如果您在阅读文档获取知识的时候，发现文档内容有不恰当的地方，请向我们反馈，让我们共同携手，将知识准确、高效且有效地传递给每一个人。

同时，如果您在日常工作、生活和学习中遇到有价值有营养的知识文档，欢迎分享到无字天书网，为知识的传承献上您的一份力量！

如果当前文档生成时间太久，请到无字天书网获取最新的文档，以跟上知识更新换代的步伐。

文档地址：<http://www.wuzitianshu.com/books/install>

**分享，让知识传承更久远！**感谢知识的创造者，感谢知识的分享者，也感谢每一位阅读到此处的读者，因为我们都将成为知识的传承者。

## 先决条件

1. `yum install -y nano wget`  
`yum install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm`

## 1. 安装JDK8

ThingsBoard服务正在Java 8上运行

1. `yum install java-1.8.0-openjdk`

请不要忘记将操作系统配置为默认使用OpenJDK 8。您可以使用以下命令配置哪个版本是默认版本：

1. `update-alternatives --config java`

使用以下命令检查安装：

1. `java -version`

```
[root@localhost ~]# java -version
openjdk version "1.8.0_222"
OpenJDK Runtime Environment (build 1.8.0_222-b10)
OpenJDK 64-Bit Server VM (build 25.222-b10, mixed mode)
```

## 2. ThingsBoard服务安装

下载安装包

1. `wget`  
`https://github.com/thingsboard/thingsboard/releases/download/v2.4.1/thingsboard-2.4.1.rpm`

```
[root@localhost ~]# wget https://github.com/thingsboard/thingsboard/releases/download/v2.4.1/thingsboard-2.4.1.rpm
--2019-09-24 07:59:23-- https://github.com/thingsboard/thingsboard/releases/download/v2.4.1/thingsboard-2.4.1.rpm
正在解析主机 github.com (github.com)... 13.229.188.59
正在连接 github.com (github.com)|13.229.188.59|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 302 Found
位置: https://github-production-release-asset-2e65be.s3.amazonaws.com/75277003/221da880-d644-11e9-919f-38b1ff0e0481?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWONJYAX4CSVEH53A%2F20190923%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190923T155813Z&X-Amz-Expires=300&X-Amz-Signature=d0e80e8def245705f7255d4dc9f4bf08611074f4d3f01cb149b5cd3e8c0226cd&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dthingsboard-2.4.1.rpm&response-content-type=application%2Foctet-stream [跟随至新的 URL]
--2019-09-24 07:59:24-- https://github-production-release-asset-2e65be.s3.amazonaws.com/75277003/221da880-d644-11e9-919f-38b1ff0e0481?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWONJYAX4CSVEH53A%2F20190923%2Fus-east-1%2Fs3%2Faws4_request&X-Amz-Date=20190923T155813Z&X-Amz-Expires=300&X-Amz-Signature=d0e80e8def245705f7255d4dc9f4bf08611074f4d3f01cb149b5cd3e8c0226cd&X-Amz-SignedHeaders=host&actor_id=0&response-content-disposition=attachment%3B%20filename%3Dthingsboard-2.4.1.rpm&response-content-type=application%2Foctet-stream
正在解析主机 github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)... 52.216.186.203
正在连接 github-production-release-asset-2e65be.s3.amazonaws.com (github-production-release-asset-2e65be.s3.amazonaws.com)|52.216.186.203|:443... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度: 116453104 (111M) [application/octet-stream]
正在保存至: "thingsboard-2.4.1.rpm"

100%[=====>] 116,453,104 1.01MB/s 用时 2m 35s
2019-09-24 08:02:00 (734 KB/s) - 已保存 "thingsboard-2.4.1.rpm" [116453104/116453104]
https://blog.csdn.net/shijie1103
```

将ThingsBoard作为服务安装

```
1. rpm -Uvh thingsboard-2.4.1.rpm
```

```
[root@localhost ~]# rpm -Uvh thingsboard-2.4.1.rpm
准备中... ##### [100%]
正在升级/安装...
 1:thingsboard-0:2.4.1-1 ##### [100%]
[root@localhost ~]#
```

### 3.配置ThingsBoard数据库

PostgreSQL安装

1. #更新系统
2. 1. yum update
3. #安装存储库RPM
  2. yum install https://download.postgresql.org/pub/repos/yum/reporpms/EL-7-x86\_64/pgdg-redhat-repo-latest.noarch.rpm
5. #安装软件包
6. 3. yum install postgresql96-server postgresql96-contrib
7. #初始化PostgreSQL数据库
8. 4. /usr/pgsql-9.6/bin/postgresql96-setup initdb
9. #启动postgresql
10. 5. systemctl start postgresql 9.6

10. 5. systemctl start postgresql-9.6
11. #将postgresql配置为在启动时启动
12. 6. systemctl enable postgresql-9.6

一旦安装了PostgreSQL，您可能想要创建一个新用户或为主要用户设置密码。以下说明将有助于为主要PostgreSQL用户设置密码

1. sudo su - postgres
2. psql
3. \password
4. \q

```
[root@localhost ~]# /usr/pgsql-9.6/bin/postgresql96-setup initdb
Initializing database ... OK
[root@localhost ~]# systemctl start postgresql-9.6
[root@localhost ~]# systemctl enable postgresql-9.6
Created symlink from /etc/systemd/system/multi-user.target.wants/postgresql-9.6.service to /usr/lib/systemd/system/postgresql-9.6.service.
[root@localhost ~]# su - postgres
su: user postgres does not exist
[root@localhost ~]# psql
psql: 致命错误: 角色 "root" 不存在
[root@localhost ~]# su - postgres
-bash-4.2$ psql
psql (9.6.15)
输入 "help" 来获取帮助信息。

postgres=# password
postgres=# \password
输入新的密码:
再次输入:
postgres=# \q
-bash-4.2$ 登出
```

<https://blog.csdn.net/shijie1103>

然后，按“ Ctrl + D”返回主用户控制台。

配置密码后，编辑pg\_hba.conf以对postgres用户使用MD5认证。

编辑pg\_hba.conf文件：

1. vim /var/lib/pgsql/9.6/data/pg\_hba.conf

找到以下几行： 替换ident为md5

```
# "local" is for Unix domain socket connections only
local    all             all                                peer
# IPv4 local connections:
host     all             all                                127.0.0.1/32      md5
# IPv6 local connections:
host     all             all                                ::1/128          ident
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local    replication     postgres                                peer
#host     replication     postgres                                127.0.0.1/32      ident
#host     replication     postgres                                ::1/128          ident
```

重新启动PostgreSQL服务以初始化新配置：

```
1. systemctl restart postgresql-9.6.service
```

连接到数据库以创建Thingsboard DB：

```
1. psql -U postgres -d postgres -h 127.0.0.1 -W
2. CREATE DATABASE thingsboard;
3. \q
```

```
[root@localhost ~]# systemctl restart postgresql-9.6.service
[root@localhost ~]# psql -U postgres -d postgres -h 127.0.0.1 -W
用户 postgres 的口令:
psql (9.6.15)
输入 "help" 来获取帮助信息.

postgres=# CREATE DATABASE thingsboard;
CREATE DATABASE
postgres=# \q
[root@localhost ~]#
```

ThingsBoard配置 编辑ThingsBoard配置文件

```
1. vim /etc/thingsboard/conf/thingsboard.conf
```

将以下行添加到配置文件。不要忘记用您的真实postgres用户密码替换

```
1. # DB Configuration
2. export DATABASE_ENTITIES_TYPE=sql
3. export DATABASE_TS_TYPE=sql
4. export SPRING_JPA_DATABASE_PLATFORM=org.hibernate.dialect.PostgreSQLDialect
5. export SPRING_DRIVER_CLASS_NAME=org.postgresql.Driver
6. export SPRING_DATASOURCE_URL=jdbc:postgresql://localhost:5432/thingsboard
7. export SPRING_DATASOURCE_USERNAME=postgres
8. export SPRING_DATASOURCE_PASSWORD=123456
```

## 4.慢速计算机的内存更新 ( 1GB RAM ) [可选]

编辑ThingsBoard配置文件

```
1. vim /etc/thingsboard/conf/thingsboard.conf
```

将以下行添加到配置文件

```
1. export JAVA_OPTS="$JAVA_OPTS -Xms256M -Xmx256M"
```

## 5.运行安装脚本

```
1. /usr/share/thingsboard/bin/install/install.sh --loadDemo
```

```
[root@localhost ~]# /usr/share/thingsboard/bin/install/install.sh --loadDemo
=====
:: ThingsBoard ::      (v2.4.1)
=====

Starting ThingsBoard Installation...
Installing DataBase schema for entities...
Installing SQL DataBase schema part: schema-entities.sql
Installing SQL DataBase schema indexes part: schema-entities-idx.sql
Installing DataBase schema for timeseries...
Installing SQL DataBase schema part: schema-ts.sql
Loading system data...
Loading demo data...
Installation finished successfully!
ThingsBoard installed successfully!
https://blog.csdn.net/shijie1103
```

## 6.启动ThingsBoard服务

在默认情况下，8080端口上可以访问ThingsBoard UI。确保您的8080端口可通过防火墙访问。为了打开8080端口，请执行以下命令：

```
1. sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
2. sudo firewall-cmd --reload
```

执行以下命令以启动ThingsBoard：

```
1. systemctl start thingsboard.service
2. systemctl enable thingsboard.service
3. #停止命令
4. systemctl stop thingsboard.service
```



启动后，您将可以使用以下链接打开Web UI：

<http://localhost:8080/>

如果在安装脚本的执行过程中指定了-loadDemo，则可以使用以下默认凭据：

System管理员：sysadmin@thingsboard.org / sysadmin 租户管理员：tenant@thingsboard.org /tenant 客户用户：customer@thingsboard.org /customer

官方参考手册：<https://thingsboard.io/docs/user-guide/install/rhel/?rhelThingsboardDatabase=postgresql>

# Homeland Docker安装教程

Homeland 基于 Docker 的自动化部署方案。

## 系统需求

Linux Server [4 Core CPU, 4G Memory, 50G Disk, 64 位] - 建议 Ubuntu Server 14.04

Docker, Docker Compose

Aliyun OSS 或 UpYun 用于文件存储。

## 安装

安装 Docker: 下面的脚本是针对 Ubuntu Server 14.04 设计的 如果要安装Ubuntu 18.0.4, 可以参考: 《[Ubuntu 18.0.4 安装 Docker CE 和 Docker compose](#)》, 其他版本, 请查阅 Docker 官方的安装文档。

```
1. curl -sSL https://git.io/install-docker | bash
```

测试是否安装成功:

```
1. sudo docker info
2. sudo docker-compose version
```

获取 homeland-docker 的项目

```
1. git clone https://github.com/ruby-china/homeland-docker.git
2. cd homeland-docker/
```

## 应用程序配置

Homeland 的应用程序配置主要基于 `app.local.env` 文件, 请参考 `app.default.env`, 以及阅读 Config File 理解各项配置信息的含义, 根据自己的需要调整。

NOTE: Docker 模式依靠环境变量文件 `app.local.env` (此文件需要手工创建) 来配置应用, 理论上你不需要修改 config/config.yml, 除非你知道如何配置。

默认情况下, Homeland 自带了基础的配置, 但你仍然需要进行一系列的配置以后才可正常使用所有功能。

必要设置

请修改 `app.local.env` 做一些必要的设置

增加 `app_name` 设置网站名称；增加 `domain` 设置网站的域名（并确保域名 DNS 绑定到你的主机的 IP）增加 `admin_emails`，增加一个管理员 `Email`（多个管理员 `Email`，用英文逗号分隔）。`app.local.env` 配置例如：

1. `app_name=网站名称`
2. `domain=your-host.com`
3. `# 默认 admin@admin.com`，你可以按你的需要修改，这里设置可以覆盖默认配置
4. `admin_emails=admin@admin.com`
5. 稍后等 Web 服务启动起来以后，你可以用这个 `Email` 来注册一个新账号，新账号将会有管理员权限。

## 编译环境

前面的脚本安装以后，docker 需要用 `sudo` 来执行，切记！

1. `sudo make install`

## 启动

1. `sudo make start`

然后服务将会以 80, 443 端口的方式跑在你的服务器上，Nginx 什么的都已经配置好了，你只需要访问你之前配置的域名即可（别忘了域名解析配置到你的服务器 IP 哦），例如 <http://your-host.com>

## 注册管理员

打开网站，并打开注册页面，用刚才设定的 `admin_emails` 的 `Email` 账号（如果没有设置，可以使用默认的 `admin@admin.com`）注册新账号，完成步骤，并登陆以后，你可以点击“导航栏”右侧的“用户头像”，弹出下拉菜单，并点击“后台”，进入到管理后台界面。

或者直接访问 <http://your-host.com/admin> 进入管理后台。

## 命令列表

在 Linux 环境里面，前面的脚本安装以后，docker，以及一下这些命令需要用 `sudo` 来执行，切记！

- | 1. Command                       | Desc   |
|----------------------------------|--|
| 2. <code>make install</code>     | 首次安装，创建数据库   |
| 3. <code>make install_ssl</code> | 安装并申请 SSL 证书（since: 2.5.0）   |
| 4. <code>make update</code>      | 更新应用程序，当 <code>homeland/homeland</code> 这个 Docker Image 版本变化的时候，需要执行，合并数据库、编译 Assets |

- |          |          |                     |
|----------|----------|---------------------|
| 5. make  | start    | 启动所有服务，将会自动启动所有的服务  |
| 6. make  | stop     | 停止所有服务              |
| 7. make  | restart  | 硬重启服务               |
| 8. make  | status   | 查看服务状态              |
| 9. make  | console  | 进入 <b>Rails</b> 控制台 |
| 10. make | stop-all | 停止所有服务，包括数据库        |
| 11. make | reindex  | 重建搜索索引              |

## 常见错误

### 邮件端口错误

```

1. #<EOFError: end of file reached>
2.
3. /usr/local/lib/ruby/2.6.0/net/protocol.rb:225:in `rbuf_fill'
4. /usr/local/lib/ruby/2.6.0/net/protocol.rb:191:in `readuntil'
5. /usr/local/lib/ruby/2.6.0/net/protocol.rb:201:in `readline'
6. /usr/local/lib/ruby/2.6.0/net/smtp.rb:950:in `recv_response'
7. /usr/local/lib/ruby/2.6.0/net/smtp.rb:553:in `block in do_start'
8. /usr/local/lib/ruby/2.6.0/net/smtp.rb:960:in `critical'
9. /usr/local/lib/ruby/2.6.0/net/smtp.rb:553:in `do_start'
10. /usr/local/lib/ruby/2.6.0/net/smtp.rb:518:in `start'

```

报上面这个错误说明你的邮件配置有问题，我遇到的时候是因为腾讯企业邮箱的465端口被阿里云封了

解决方案：可使用587端口

参考配置：

```

1. mailer_provider=smtp
2. mailer_sender=xxx@wuzitianshu.com
3. mailer_options_address=smtp.exmail.qq.com
4. mailer_options_port=587
5. mailer_options_domain=qq.com
6. mailer_options_user_name=xxx@wuzitianshu.com
7. mailer_options_password=xxxxxx
8. mailer_options_authentication=plain
9. mailer_options_enable_starttls_auto=true

```

### 邮箱认证错误

1.

```
2. #<Net::SMTPAuthenticationError: 535 Error: authentication failed, system busy>
3.
4. /usr/local/lib/ruby/2.6.0/net/smtp.rb:981:in `check_auth_response'
5. /usr/local/lib/ruby/2.6.0/net/smtp.rb:736:in `auth_plain'
6. /usr/local/lib/ruby/2.6.0/net/smtp.rb:728:in `authenticate'
7. /usr/local/lib/ruby/2.6.0/net/smtp.rb:565:in `do_start'
8. /usr/local/lib/ruby/2.6.0/net/smtp.rb:518:in `start'
```

如果使用腾讯企业邮箱帐号密码都正确，但还是报上面这个错误，说明是你还没有为配置的邮箱授权

解决方案： 为了保障用户邮箱的安全，腾讯企业邮箱设置了POP/SMTP/IMAP的开关。 首先，登录管理端，在管理工具-用户登录-登录权限设置，开启POP/SMTP服务或者IMAP/SMTP服务。

首页

通讯录

系统日志

管理工具

我的企业

< 返回

登录权限设置

启用POP/SMTP服务

设置范围 不在启用范围内的帐号，将无法使用POP客户端收发邮件

A社区 X

启用IMAP/SMTP服务

设置范围 不在启用范围内的帐号，将无法使用IMAP客户端收发邮件

A社区 X

强制启用安全登录

设置范围 强制启用后，若该帐号未启用安全登录，将关闭其客户端发信功能

旧版迁移新版杨珍 X

## ubuntu下部署

### 安装宝塔控制面板(非必须)

如果你已经有lnmp环境了，则不需要安装。

宝塔官网：<http://www.bt.cn>

安装命令：

```
wget -O install.sh http://download.bt.cn/install/install-ubuntu.sh && sudo bash  
1. install.sh
```

安装完宝塔控制面板之后，登录面板，安装“lnmp”（Linux、Nginx、MySQL、PHP）

安装默认的即可

安装宝塔面板，主要是为了方便管理站点，如反向代理、SSL等

安装中文字体（非必须，但是建议安装）有的Linux服务器并没有支持中文字体，需要手动安装。地址：<http://www.hc-cms.com/thread-41-1-1.html>

安装命令：

```
1. apt install ttf-wqy-zenhei  
2. apt install fonts-wqy-microhei
```

安装中文字体支持，主要是为了避免Linux系统不支持中文的时候，导致导出文档的时候出现乱码。

### 安装Chrome

直接使用命令一键安装：

```
1. apt install chromium-browser
```

执行以下命令，如果能打印百度页面代码，则表示安装成功。

```
chromium-browser --headless --disable-gpu --dump-dom --no-sandbox  
1. https://www.baidu.com
```

这个主要用于在发布文档的时候，渲染未被渲染的markdown文档，以及强力模式下的网页采集。

### 安装calibre

calibre官网: <https://www.calibre-ebook.com/>

安装命令:

```
sudo -v && wget -nv -O- https://download.calibre-ebook.com/linux-installer.py |  
sudo python -c "import sys; main=lambda:sys.stderr.write('Download failed\n');  
1. exec(sys.stdin.read()); main()"
```

执行下面命令, 能看到版本号, 表示安装成功。

```
1. ebook-convert --version
```

注意:

这里要安装最新版的calibre, 执行上述命令, 安装的就是最新版了。如果出现warning的提示, 直接不用理会即可;error级别的提示, 网上搜下看下如何解决 calibre主要用于将文档导出生成pdf、epub、mobi文档。

测试

随便创建一个txt文件

```
1. echo "Hello BookStack. 你好, 书栈。" > test.txt
```

转成pdf

```
1. ebook-convert test.txt test.pdf
```

查看测试的转化效果, 主要看下转化的过程中有没有报错, 以及转化后的文档有没有出现中文乱码。

## 安装Git

使用 git clone 的方式导入项目的时候需要使用git

[sudo] apt install git 安装puppeteer 生成自定义的书籍封面以及内容采集时需要用到, 可全局安装, 也可在程序根目录下安装。我们这里使用全局安装的方式。

```
1. [sudo] apt install nodejs-legacy npm  
2. [sudo] npm install -g n  
3. [sudo] n stable  
4. [sudo] npm install -g cnpm  
5. [sudo] cnpm install -g puppeteer
```

## 安装supervisor

本教程是教您通过命令安装，其实宝塔面板也可以可视化安装supervisor，很方便，这里就不多做介绍了，有需要可以跟我联系！

安装命令：

```
1. apt install supervisor
```

判断是否安装成功：

```
1. supervisord --version
```

显示版本号，即表示安装成功。

**supervisor** 主要用于将程序加入到系统守护进程

**Supervisord** 安装完成后有两个可用的命令行 **supervisord** 和 **supervisorctl**，命令使用解释如下：

1. **supervisord**，初始启动 **Supervisord**，启动、管理配置中设置的进程。  
**supervisorctl stop programxxx**，停止某一个进程(programxxx)，programxxx 为
2. **[program:beepkg]** 里配置的值，这个示例就是 **beepkg**。
3. **supervisorctl start programxxx**，启动某个进程
4. **supervisorctl restart programxxx**，重启某个进程  
**supervisorctl stop groupworker:**，重启所有属于名为 **groupworker** 这个分组的进程
5. **(start, restart 同理)**  
**supervisorctl stop all**，停止全部进程，注：**start**、**restart**、**stop** 都不会载入最新的配置文
6. 件。
7. **supervisorctl reload**，载入最新的配置文件，停止原有进程并按新的配置启动、管理所有进程。  
**supervisorctl update**，根据最新的配置文件，启动新配置或有改动的进程，配置没有改动的进程不会
8. 受影响而重启。

## 部署程序

### 下载Linux版的程序

<https://github.com/TruthHun/BookStack/releases>

### 修改配置文件

在本地解压，修改配置文件

配置文件在conf目录。



## 配置数据库和文档导出项

把app.conf.example命名为app.conf,并根据提示，修改配置。这里主要修改MySQL数据库的配置以及生成下载文档的配置。其他项能不改就不改。

```

1. # 程序名称
2. appname = BookStack
3. # 微信小程序 appid
4. appId=""
5. # 微信小程序appSecret
6. appSecret=""
7. # 是否限制API请求，也就是如果不是上述配置的微信小程序的appId请求的接口，则直接拒绝
8. limitReferer=false
9. # API 路由前缀，默认为 bookchat。API访问形式：https://您的域名/bookchat/api/v1/xxx
   # 比如您把api前缀改为了 hello，则您在 BookChat 或者 BookChatApp 中，需要把 config.js
10. 中的 host 常量配置为 https://您的域名/hello
11. apiPrefix=bookchat
12. # 是否显示小程序阅读码（需要配置了appScecret才会生效）
13. showWechatCode = false
   # 比如你将static目录下的所有静态资源都放到了专门的服务器上，那么这个域名就行用来访问你的静态资
14. 源的域名。否则建议填写web网站的域名
15. # 如果您部署了微信小程序，则该值一定要填写。
16. static_domain=
17. # 监听端口
18. httpport = 8181
   # 运行模式。开发时，请设置为开发模式"dev"，即development；部署时，请设置为产品模式"prod"，
19. 即product。dev模式下，会打印各种调试信息
20. runmode = dev
21. # 是否开启session，这个必须开启，否则没法玩了。
22. sessionon = true
23. # session名称，这个你自己定。
24. sessionname = bookstack
25. copyrequestbody = true
26. # 启动gzip压缩，则设置为true，否则设置为false。建议启动
27. EnableGzip=true
28. # 要压缩的静态文件扩展名，其中.xml是sitemap站点地图压缩。建议按照这个来就行了
29. StaticExtensionsToGzip = .css, .js, .xml
30. #站点地图host（主要用于使用CDN的时候，host与实际host不一样，导致生成的sitemap不一样）
31. sitemap_host=www.bookstack.cn
32. # 默认Session生成Key的秘钥
33. beegoserversessionkey=bookstack
34. #####Session储存方式#####
35. #以文件方式储存

```

```

36. sessionprovider=file
37. sessionproviderconfig=store/session
38. # 静态目录。这个是站点地图的静态目录
39. StaticDir = sitemap:sitemap
    # 站点可直接访问的静态文件【注意，这里千万不能配置.conf文件扩展名，否则会把.conf文件当做静态
40. 文件而被外部访问，导致数据库账号密码等泄露。按照默认的来即可】
41. StaticExt=.txt,.xml,.ico,.png,.jpg,.jpeg,.gif,.html
42. #评论时间间隔，每次发表评论，限定间隔秒数，避免被恶意刷评论
43. CommentInterval=10
    # 生成下载文档时导出pdf文档的配置，这里建议把关于bookstack.cn的信息替换成你自己的就行，其他
44. 的建议不要动
45. exportHeader=<p style='color:#8E8E8E;font-size:12px;'>_SECTION_</p>
    exportFooter=<p style='color:#8E8E8E;font-size:12px;'>本文档使用 <a
    href='http://www.bookstack.cn' style='text-decoration:none;color:#1abc9c;font-
    weight:bold;'>书栈(BookStack.CN)</a> 构建 <span style='float:right'>- _PAGENUM_
46. -</span></p>
47. exportFontSize=13
48. exportPaperSize=a4
49. exportCreator=书栈(BookStack.CN)
50. exportMarginLeft=36
51. exportMarginRight=36
52. exportMarginTop=36
53. exportMarginBottom=36
54. # 是否使用自定义的电子书封面，如果使用，则需要安装并启用puppeteer，否则不生效
55. exportCustomCover=true
56. #####MySQL 数据库配置#####
57. db_adapter=mysql
58. # 您的数据库host
59. db_host=
60. #您的数据库端口
61. db_port=3306
62. #您的数据库用户名
63. db_username=
64. # 您的数据库密码
65. db_password=
66. # utf8或者utf8mb4数据库。如果数据库还没创建，并且账号有创建权限，将自动创建。
67. db_database=bookstack
    # 谷歌浏览器，用于发布内容的时候渲染未被渲染的markdown。建议安装最新版的Chrome浏览器，并把
68. Chrome浏览器加入系统环境变量。

    # 使用Chrome的headless去处理。之前考虑使用phantomjs的，但是phantomjs有些小问题，不如
69. Chrome强大。
70. # chrome=chromium-browser
71. chrome=/Applications/Google\ Chrome.app/Contents/MacOS/Google\ Chrome
72. # 如果使用 puppeteer 则忽略 chrome

```

```

72. # 如未使用 puppeteer , 则忽略chrome
73. puppeteer = false
74. #项目默认封面
75. cover=/static/images/book.png
76. #默认头像
77. avatar=/static/images/avatar.png
78. #默认阅读令牌长度
79. token_size=12
80. #上传文件的后缀
81. upload_file_ext=txt|doc|docx|xls|xlsx|ppt|pptx|pdf|7z|rar|jpg|jpeg|png|gif
82. #####邮件配置#####
83. #是否启用邮件
84. enable_mail=true
85. #每小时限制指定邮箱邮件发送次数
86. mail_number=5
87. #smtp服务用户名
88. smtp_user_name=
89. #smtp服务器地址
90. smtp_host=
91. #smtp密码
92. smtp_password=
93. #端口号
94. smtp_port=25
95. #发送邮件的显示名称
96. form_user_name=
97. #接收回件的邮箱。留空则表示使用发送邮件的邮箱作为接收回件邮箱
98. reply_user_name=bookstack@qq.com
99. #邮件有效期30分钟
100. mail_expired=30
    # 存储类型，后续扩展成本地存储(local)、阿里云OSS存储(oss)、七牛云存储(qiniu)、腾讯云存储
101. (cos)、百度云存储(bos)和又拍云存储(upyun)
102. # 目前可选项：local、oss
103. store_type=local
104. # 不要修改
105. include "oss.conf"
106. include "oauth.conf"

```

## 配置OSS

1、登录你的阿里云，进入OSS，创建一个只读Bucket。 2、在本地随便创建一个txt文档，将下面两行粘贴进去：

```
1.
styleName:cover,styleBody:image/resize,m_fill,w_175,h_230,limit_0/auto-orient,1/c
2.
styleName:avatar,styleBody:image/resize,m_fill,w_100,h_100,limit_0/auto-orient,1/
```

3、在OSS的 **图片处理-导入样式** 中，将该txt文档导入 Linux(Ubuntu)下安装和部署 -



配置完成后，设置图片内容模板解析格式为：/



4、将oss.conf.example修改成oss.conf，并根据您的阿里云OSS，配置该文件。

```

1. # 阿里云OSS配置
2. [oss]
   # 是否是内网，如果您的阿里云服务器和OSS是同一内网，请设置为true，这样能更快地将文件移到oss
3. 上，否则设置为false。
4. # 一般情况下，在开发阶段，设置为false
5. IsInternal=false
6. # 内网Endpoint,如：oss-cn-hongkong-internal.aliyuncs.com
7. EndpointInternal=
8. # 外网Endpoint，如：oss-cn-hongkong.aliyuncs.com
9. EndpointOuter=
10. # 您的阿里云AccessKeyId
11. AccessKeyId=
12. # 您的阿里云AccessKeySecret
13. AccessKeySecret=
14. # 只读状态的Bucket，不要私有的，也不要公共读写的
15. Bucket=
16. # oss中绑定的域名，如果您没有绑定域名，则使用阿里云oss的自带域名也行
17. Domain=

```

## 配置第三方登录

将oauth.conf.example重命名成oauth.conf,并根据提示修改配置。

```

1. # 第三方登录配置
2. [oauth]
3. ##### Gitee(码云) #####
4. # 申请地址（需要先登录）：https://gitee.com/oauth/applications
5. # 您的ClientId
6. giteeClientId=
7. # 您的ClientSecret
8. giteeClientSecret=
9. # 回调地址，把下面的bookstack.cn的域名换成你的即可
10. giteeCallback=http://www.bookstack.cn/login/gitee
11. # 下面这两项不要动
12. giteeAccessToken=https://gitee.com/oauth/token
13. giteeUserInfo=https://gitee.com/api/v5/user
14. ##### GitHub #####
15. # 申请地址(需要先登录你的GitHub)：https://github.com/settings/developers
16. # 您的ClientId
17. githubClientId=
18. # 您的ClientSecret

```

```

19. githubClientSecret=
20. # 回调地址，把下面的bookstack.cn的域名换成你的即可
21. githubCallback=http://www.bookstack.cn/login/github
22. # 下面这两项不要动
23. githubAccessToken=https://github.com/login/oauth/access_token
24. githubUserInfo=https://api.github.com/user
25. ##### QQ #####
26. # 申请地址（需要先登录你的QQ）:https://connect.qq.com/manage.html
27. #ClientId,即 APP ID
28. qqClientId=
29. #ClientSecret,即 APP Key
30. qqClientSecret=
31. # 回调地址，把下面的bookstack.cn的域名换成你的即可
32. qqCallback=http://www.bookstack.cn/login/qq
33. # 下面这三项不要动
34. qqAccessToken=https://graph.qq.com/oauth2.0/token
35. qqOpenId=https://graph.qq.com/oauth2.0/me

36. qqUserInfo=https://graph.qq.com/user/get_user_info
    ### TODO 微信和微博登录，主要是我这边忘记了以前注册的个人开发者信息，当前没开发，后续会开发出
37. 来 #####

```

## 上传和部署

- 1、将修改和配置好了的程序压缩，上传到站点根目录下，解压。
- 2、执行数据库安装。程序安装一些站点配置项、SEO项等。命令：

```
1. ./BookStack install
```

- 3、配置反向代理 宝塔面板用户，直接在站点的设置里面配置反向代理，但是需要注意的是，需要修改缓存，如把下面两句注释掉：

```

1. add_header X-Cache $upstream_cache_status;
2. expires 12h;

```

## Nginx反向代理配置参考(宝塔的配置)：

注意：如果开启缓存，请将缓存时间设置为1s；

```

1. server
2. {
3.     listen 80;

```

```

4.     server_name demo.bookstack.cn;
5.     index index.php index.html index.htm default.php default.htm default.html;
6.     root /www/wwwroot/demo.bookstack.cn;
7.     location /
8.     {
9.         proxy_pass http://localhost:8181;
10.        proxy_set_header Host $host;
11.        proxy_set_header X-Real-IP $remote_addr;
12.        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
13.        proxy_set_header REMOTE-HOST $remote_addr;
14.        #缓存相关配置
15.        #proxy_cache cache_one;
16.        #proxy_cache_key $host$request_uri$is_args$args;
17.        #proxy_cache_valid 200 304 301 302 1h;
18.        #持久化连接相关配置
19.        #proxy_connect_timeout 30s;
20.        #proxy_read_timeout 86400s;
21.        #proxy_send_timeout 30s;
22.        #proxy_http_version 1.1;
23.        #proxy_set_header Upgrade $http_upgrade;
24.        #proxy_set_header Connection "upgrade";
25.    }
26.    location ~ .*\. (php|jsp|cgi|asp|aspx|flv|swf|xml)?$
27.    {
28.        proxy_set_header Host $host;
29.        proxy_set_header X-Real-IP $remote_addr;
30.        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
31.        proxy_set_header REMOTE-HOST $remote_addr;
32.        proxy_pass http://localhost:8181;
33.    }
34.    #PROXY-END
35.    include enable-php-54.conf;
36.    #PHP-INFO-END
37.    #REWRITE-START URL重写规则引用,修改后将导致面板设置的伪静态规则失效
38.    include /www/server/panel/vhost/rewrite/demo.bookstack.cn.conf;
39.    #REWRITE-END
40.    #禁止访问的文件或目录
41.    location ~
42.    {
43.        return 404;
44.    }
45.    access_log off;

```

```
46. }
```

## 加入系统守护进程

### 1、进入supervisor的配置目录

```
1. cd /etc/supervisor/conf.d/
```

### 2、配置守护进程 创建bookstack.conf文件，并配置。

```
1. [program:BookStack]
2. directory = 你的程序目录
3. command =你的程序执行命令
4. autostart = true
5. autorestart=true
6. user = 启动该程序的用户
7. redirect_stderr = true
8. stdout_logfile = 日志地址
```

配置示例：

```
1. [program:BookStack]
2. directory = /www/wwwroot/demo.bookstack.cn
3. command =/www/wwwroot/demo.bookstack.cn/BookStack
4. autostart = true
5. autorestart=true
6. user = www
7. redirect_stderr = true
8. stdout_logfile = /var/log/supervisor/BookStack.log
```

配置完成之后，重启supervisor

```
1. supervisorctl reload
```

## 默认管理员账号和密码

```
admin admin
```

v2.0 正式版开始，默认管理员账号密码调整为：

```
admin admin888
```



# 总结

安装BookStack，需要先配置环境，安装依赖：Nginx、MySQL、calibre、chorme ( chromium-browser )、supervisor，然后修改配置文件，配置MySQL数据库、OSS和第三方登录，然后上传部署和配置反向代理以及加入守护进程。

# 常见问题

1、转换PDF失败 问题报错信息： 此问题是由于root权限不能启动chrome渲染导致，解决方案：启动supervisor使用www用户或其他非root用户启动都可以成功转换。 注意：如果使用宝塔面板，默认操作权限是www，在按照上面的步骤完全操作完成以后，很多文件应该是root权限，www用户是不能操作的，所以需要使用宝塔面板将部署网站的目录权限及子目录权限全部设置为：755。



## Ubuntu 18.0.4 安装 Docker CE 和 Docker compose

# 安装Docker CE

## 第一步

如果之前安装过docker，执行下面命令删除

```
1. apt-get remove docker docker-engine docker.io
```

删除后执行更新软件

```
1. sudo apt-get update
```

## 第二步

安装必要的软件包以允许apt通过HTTPS使用存储库，具体如下：

```
sudo apt-get install apt-transport-https ca-certificates curl software-properties-common
```

## 第三步

添加GPG密钥，可以添加官方的和阿里的，我添加的阿里的，国内的快啊（墙裂推荐）

```
1. // 阿里
   curl -fsSL https://mirrors.aliyun.com/docker-ce/linux/ubuntu/gpg | sudo apt-key
2. add -
3. // 官方
4. curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -
```

添加完毕后可以执行以下命令验证：

```
1. sudo apt-key fingerprint 0EBFCD88
```

正常情况下会输出如下内容：

```
1. pub      rsa4096 2017-02-22 [SCEA]
2.          9DC8 5822 9FC7 DD38 854A E2D8 8D81 803C 0EBF CD88
3. uid              [ 未知 ] Docker Release (CE deb) <docker@docker.com>
```

## 4. sub rsa4096 2017-02-22 [S]

看到上面内容，说明已经安装好了

## 第四步

设定稳定仓储库，这一步我被坑了好久，具体参考

docker配置仓储库时出错：无法安全地用该源进行更新，所以默认禁用该源 也可以不设置，不设置默认使用官方的，具体是：deb [arch=amd64] <https://download.docker.com/linux/ubuntu> xenial stable 同样可以用阿里 的镜像：设置命令如下：

```
sudo add-apt-repository "deb [arch=amd64] https://mirrors.aliyun.com/docker-
1. ce/linux/ubuntu $(lsb_release -cs) stable"
```

其中的 `lsb_release -cs` 相当于一个函数，直接获取Ubuntu下的最新版本

设置完毕再次执行

```
1. sudo apt-get update
```

命令更新 软件包。

第五步 安装Docker，如下：

```
1. sudo apt-get -y install docker-ce
```

也可以指定想安装 的Docker版本（ **提示：此步骤可选，建议新手按照上面的步骤进行安装即可** ），方法如下：

执行如下命令查看有哪些版本，

```
1. apt-cache madison docker-ce
```

输出如下：

```
docker-ce | 5:18.09.0~3-0~ubuntu-bionic | http://mirrors.aliyun.com/docker-
1. ce/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 18.06.1~ce~3-0~ubuntu | http://mirrors.aliyun.com/docker-
2. ce/linux/ubuntu bionic/stable amd64 Packages
docker-ce | 18.06.0~ce~3-0~ubuntu | http://mirrors.aliyun.com/docker-
3. ce/linux/ubuntu bionic/stable amd64 Packages
```

```
docker-ce | 18.03.1~ce~3-0~ubuntu | http://mirrors.aliyun.com/docker-  
4. ce/linux/ubuntu bionic/stable amd64 Packages
```

选择要安装的版本，执行

```
1. sudo apt-get install -y docker-ce=<VERSION>
```

命令即可。

安装完成 后执行

```
1. docker -v
```

命令，如果正常输出说明安装成功。

## 安装Docker compose

1、运行此命令以下载Docker Compose的当前稳定版本

```
sudo curl -L  
"https://github.com/docker/compose/releases/download/1.24.1/docker-  
1. compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

以上运行过慢，可换成国内的

```
sudo curl -L  
"https://get.daocloud.io/docker/compose/releases/download/1.24.1/docker-  
1. compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

2、对二进制文件应用可执行权限：

```
1. chmod +x /usr/local/bin/docker-compose
```

### 3、参考官网如下

The screenshot shows the Docker documentation page for installing Docker Compose. The left sidebar has a menu with 'Install Compose' highlighted. The main content area is titled 'Install Compose on Linux systems' and includes the following steps:

- Run this command to download the current stable release of Docker Compose:
 

```
curl -L https://get.docker.com/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m) -o /usr/local/bin/docker-compose
```
- Apply executable permissions to the binary:
 

```
sudo chmod +x /usr/local/bin/docker-compose
```

A note mentions that if the command fails, the user should check their path or create a symbolic link to `/usr/bin`. An example command is provided:

```
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

The URL `https://blog.csdn.net/ziwuzhulin` is visible in the bottom right corner of the screenshot.

安装完成后，可测试

#### 1. docker-compose --version

查看版本

```
[root@daocloud ~]# curl -L "https://get.daocloud.io/docker/compose/releases/download/1.24.1/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 423 100 423 0 0 1796 0 --:--:-- --:--:-- --:--:-- 1800
100 15.4M 100 15.4M 0 0 9218k 0 0:00:01 0:00:01 --:--:-- 14.5M
[root@daocloud ~]# chmod +x /usr/local/bin/docker-compose
[root@daocloud ~]# docker-compose --version
docker-compose version 1.24.1, build 4667896b
```