使用 docker-compose 部署 solo 个人博客

作者：Wit13

原文链接：https://ld246.com/article/1555406332535

来源网站：链滴

从 Tomcat 部署 升级到 docker-compose 部署，升级前先备份一下 MySQL 数据库，升级后在备份来即可。

一、安装 docker

1.更新 yum 包

```
yum update
```

2.安装必要的系统工具

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

3.设置 yum 源

```
yum-config-manager --add-repo http://mirrors.aliyun.com/docker-ce/linux/centos/docker-ce.epo
```

4.查看并选择合适的 docker 版本

```
yum list docker-ce --showduplicates | sort -r
```

5.安装 docker

```
yum install docker-ce-17.12.1.ce
```

我这里下载的版本是：docker-ce-17.12.1.ce

6.DaoCloud 配置 docker 镜像源加速

```
curl -sSL https://get.daocloud.io/daotools/set_mirror.sh | sh -s http://f1361db2.m.daocloud.io
```

二、下载 docker-compose

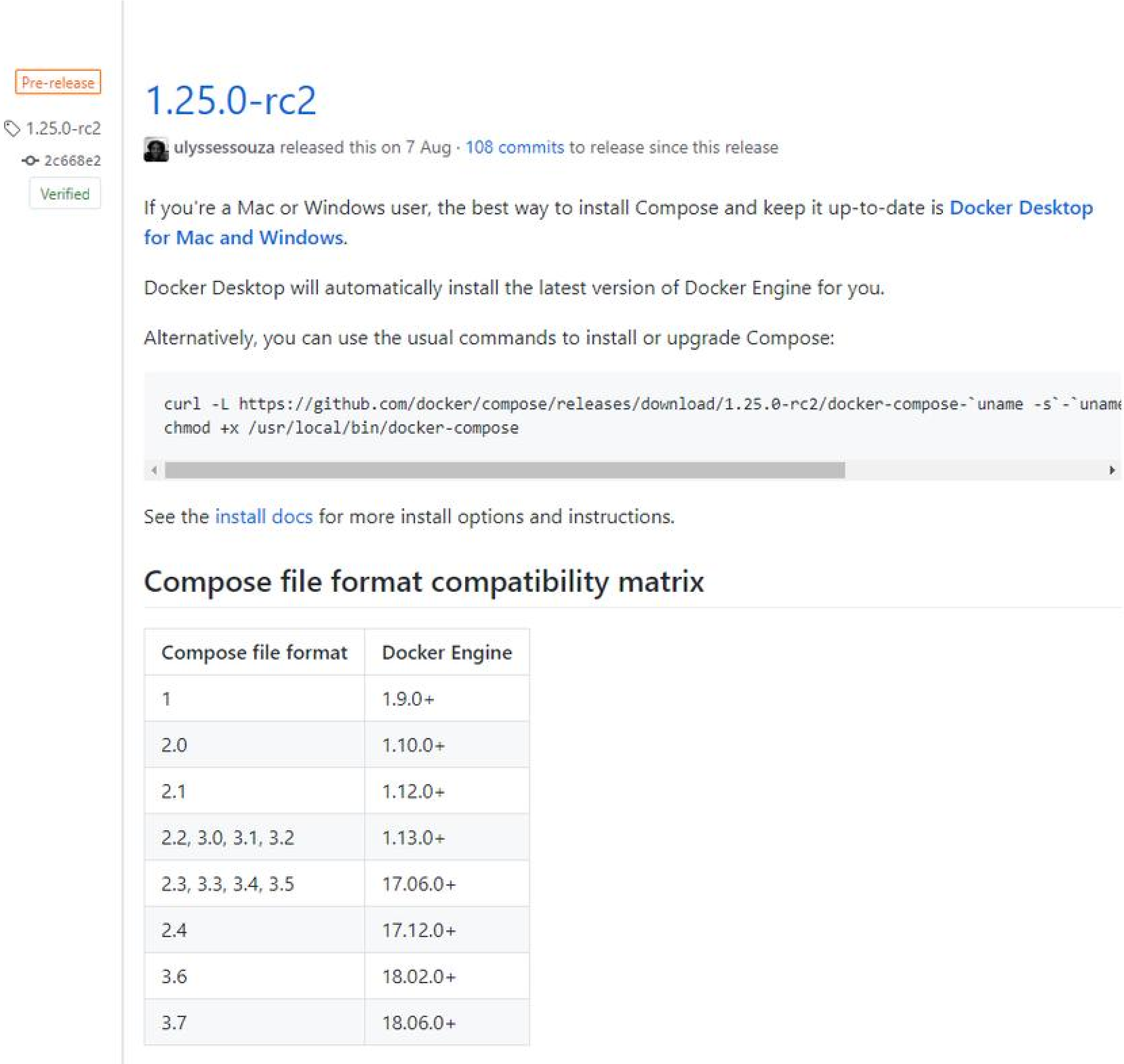
1.选择最新的 docker-compose 版本并下载

建议使用方法一，虽然比较麻烦，但是比方法二下载快

我这里下载的版本是：1.25.0-rc2

方法一：直接去 docker-compose下载

找到你想要的版本



1.25.0-rc2

ulyssessouza released this on 7 Aug · 108 commits to release since this release

If you're a Mac or Windows user, the best way to install Compose and keep it up-to-date is Docker Desktop for Mac and Windows.

Docker Desktop will automatically install the latest version of Docker Engine for you.

Alternatively, you can use the usual commands to install or upgrade Compose:

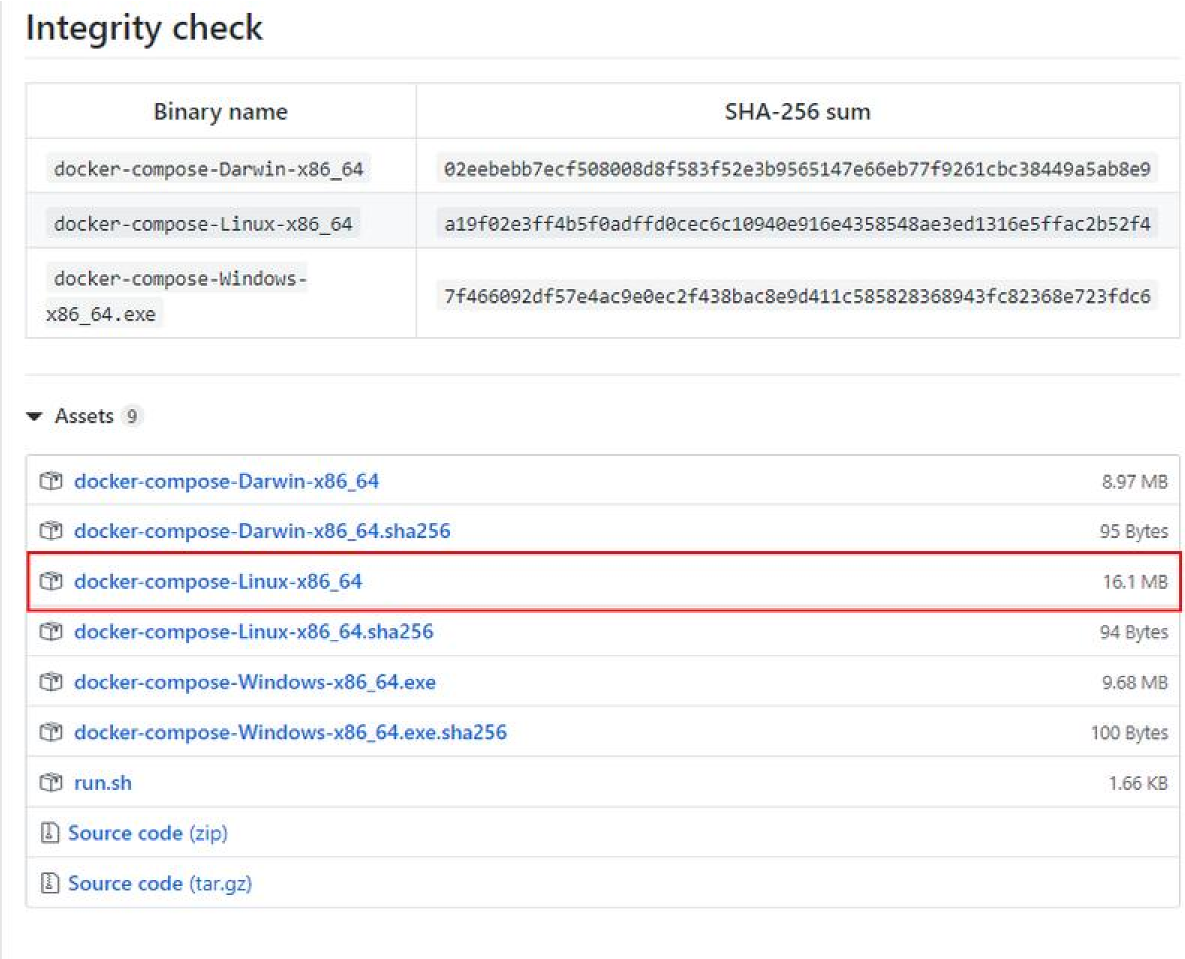
```
curl -L https://github.com/docker/compose/releases/download/1.25.0-rc2/docker-compose-`uname -s`-`uname
chmod +x /usr/local/bin/docker-compose
```

See the install docs for more install options and instructions.

Compose file format compatibility matrix

Compose file format	Docker Engine
1	1.9.0+
2.0	1.10.0+
2.1	1.12.0+
2.2, 3.0, 3.1, 3.2	1.13.0+
2.3, 3.3, 3.4, 3.5	17.06.0+
2.4	17.12.0+
3.6	18.02.0+
3.7	18.06.0+

找到 **Assets** 下载红色框选中的内容



Integrity check

Binary name	SHA-256 sum
`docker-compose-Darwin-x86_64`	`02eebebb7ecf508008d8f583f52e3b9565147e66eb77f9261cbc38449a5ab8e9`
`docker-compose-Linux-x86_64`	`a19f02e3ff4b5f0adffd0cec6c10940e916e4358548ae3ed1316e5ffac2b52f4`
`docker-compose-Windows-x86_64.exe`	`7f466092df57e4ac9e0ec2f438bac8e9d411c585828368943fc82368e723fdc6`

Assets 9

docker-compose-Darwin-x86_64	8.97 MB
docker-compose-Darwin-x86_64.sha256	95 Bytes
docker-compose-Linux-x86_64	16.1 MB
docker-compose-Linux-x86_64.sha256	94 Bytes
docker-compose-Windows-x86_64.exe	9.68 MB
docker-compose-Windows-x86_64.exe.sha256	100 Bytes
run.sh	1.66 KB
Source code (zip)	
Source code (tar.gz)	

下载完成后，使用 FTP 上传到

目录下，并重命名为：**docker-compose**

然后再设置 docker-compose 执行权限

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

方法二：使用命令下载

2.设置 docker-compose 执行权限

三、配置 blog

1.新建以下空文件夹目录

```
└─blog
    ├─mysql
    │  ├─conf
    │  ├─data
    │  ├─init
    │  └─logs
    ├─nginx
    │  ├─cert
    │  ├─conf
    │   └─logs
    └─solo
        ├─conf
        └─skins
```

2.数据库配置文件

```
cd blog/mysql/conf
vi my.cnf

# my.cnf
[mysqld]
user=mysql
default-storage-engine=INNODB
character-set-server=utf8mb4
[client]
default-character-set=utf8mb4
[mysql]
default-character-set=utf8mb4

cd blog/mysql/init
vi init.sql

# init.sql
create database if not exists `solo` character set utf8mb4 collate utf8mb4_general_ci;
```

3. nginx 配置文件

```
cd blog/nginx/conf
vi app.conf

# app.conf
server {
    listen 80;
    charset utf-8;
        access_log /etc/nginx/log/access.log;
        error_log /etc/nginx/log/error.log;
        location / {
            proxy_pass http://blog-solo:8080;
            proxy_set_header Host $host:$server_port;
            proxy_set_header X-Forwarded-Host $server_name;
            proxy_set_header X-Real-IP $remote_addr;
            proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
            client_max_body_size 10m;
        }
}
```

4. solo 配置文件

```
cd blog/solo/conf
vi latke.properties

# latke.properties
serverScheme=http
serverHost=填写你的域名
```

```
serverPort=80
runtimeMode=PRODUCTION
```

5. solo 皮肤文件

把最先准备好的官方皮肤，放到下面这个目录下

```
cd blog/solo/skins
```

6. docker-compose.yml 配置文件

```
cd blog
vi docker-compose.yml
```

```
# docker-compose.yml
version: "3"
services:
  blog-mysql:
    image: mysql:5.7
    restart: always
    container_name: blog-mysql
    privileged: false
    ports:
      - 3306:3306
    networks:
      - blog-extranet
    environment:
      - MYSQL_ROOT_PASSWORD=setapassword
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - ./mysql/conf/:/etc/mysql/
      - ./mysql/data/:/var/lib/mysql/
      - ./mysql/init/:/docker-entrypoint-initdb.d/
      - ./mysql/logs/:/logs/
  blog-solo:
    image: b3log/solo
    restart: always
    container_name: blog-solo
    privileged: false
    networks:
      - blog-extranet
    links:
      - blog-mysql
    depends_on:
      - blog-mysql
    ports:
      - 8080:8080
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - ./solo/conf/latke.properties:/opt/solo/latke.properties:ro
      - ./solo/skins/:/opt/solo/skins/:ro
    environment:
      - RUNTIME_DB=MYSQL
      - JDBC_USERNAME=root
      - JDBC_PASSWORD=数据库密码
      - JDBC_DRIVER=com.mysql.cj.jdbc.Driver
      - JDBC_URL=jdbc:mysql://blog-mysql/solo?useUnicode=yes&characterEncoding=UTF-8
useSSL=false&serverTimezone=UTC
  blog-nginx:
    image: nginx:1.14
    restart: always
    container_name: blog-nginx
    privileged: false
    networks:
      - blog-extranet
    links:
      - blog-solo
    depends_on:
      - blog-solo
    ports:
      - 80:80
      - 443:443
    volumes:
      - /etc/localtime:/etc/localtime:ro
      - ./nginx/conf/:/etc/nginx/conf.d/
      - ./nginx/cert/:/etc/nginx/cert/
      - ./nginx/logs/:/etc/nginx/log/
networks:
  blog-extranet:
    driver: bridge
```

四、启动与停止

1.启动

启动 dokcer

```
systemctl start docker
```

启动 docker-compose

```
cd blog
docker-compose up -d
```

2.停止

```
cd blog
docker-compose rm -svf
```

最后，这是配置好的，可以直接 **下载** 下来用，修改 **serverHost** 为你自己的域名或 ip 地址。**MYSQL_ROOT_PASSWORD** 和 **JDBC_PASSWORD** 密码是一致的，建议修改为你自己的密码。

注意： 如果你第一次运行了 **docker-compose up -d** 命令后，想改数据库密码，把 /blog/mysql/data 下的文件删掉（删之前先备份一下数据库，避免不必要的损失），改完之后在运行 **docker-compose up -d** 命令，你设置的数据库密码就生效了。

特别感谢 **Jeffrey** 的教程

五、升级版本

1.拉取最新 solo 镜像

```
docker pull b3log/solo
```

2.查看镜像

```
docker images
```

3.删除旧的 solo 镜像

```
docker rmi 'IMAGE ID'
```

4.启动 docker-compose

```
docker-compose up -d
```

docker 部署参考：https://hacpai.com/article/1598322912800#mysql