Debain10 系统安装 Docker 并部署 MySQL+solo+Nginx

作者：mnizht
原文链接：https://ld246.com/article/1567066459529
来源网站：链滴

Debain10 系统安装Docker并部署MySQL+solo+Nginx

Debain9 开启bbr

网路颠簸，尤其是晚高峰的时候，我们访问国外的服务器难免会出现速度缓慢的问题，如果我们的服器为Linux系统，好在Debian 9系统内核版本为4.9.x，其内核自带了由Google开发的BBR拥塞算法，们能很方便的开启BBR进行访问加速。

Debian 9(10)系统开启Google BBR拥塞控制算法方法

首先，先说一下开启方法。其实开启很简单，只需要三行命令即可：

```
$ echo "net.core.default_qdisc=fq" >> /etc/sysctl.conf
$ echo "net.ipv4.tcp_congestion_control=bbr" >> /etc/sysctl.conf
$ sysctl -p
```

如果正常执行之后没有报错的话，**需要重启一下系统**，让配置生效。

重启之后，执行如下命令查看配置是否成功生效：

```
$ lsmod | grep bbr
# tcp_bbr               20480  0
```

如果有如上类似输出，则证明Google BBR配置成功了。

打开防火墙 ufw

```
# 安装
$ apt install ufw
# 设置防火墙规则
$ ufw default deny incoming
$ ufw default allow outgoing
# 允许ssh端口
$ ufw allow ssh
# 开启端口
$ ufw allow 3306

# 查看ufw状态，活动中的话会列出所有开放的端口
$ ufw status
# 删除开放的端口
$ ufw delete 数字(ufw status列出的规则行号)
# 重启
$ ufw reload
```

安装docker

参考

```
# 在安装Docker之前， 需要添加Docker的私有仓库。安装必要的依赖包：
$ sudo apt update
$ sudo apt install apt-transport-https ca-certificates curl software-properties-common gnupg
```

```
# 导入Docker仓库的GPG key
$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
# 添加Docker稳定版仓库到debian系统仓库中
$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lb_release -cs) stable"
# 更新仓库信息， 安装 Docker 社区版本
$ sudo apt update
$ sudo apt install docker-ce
# 一旦安装完成， Docker 服务将自动启动， 使用服务管理工具来查看其状态
$ sudo systemctl status docker
# 检查下安装的Docker版本
$ sudo docker -v
# 检查docker内置网络
$ docker network ls
```

docker部署MySQL

```
$ docker pull mysql
# 安装启动前先创建对应的挂载文件夹 ***/mysql8/data 和 ***/mysql8/conf, 然后去docker的imag文件中找到my.cnf 文件拷贝到创建的conf文件夹下
# 这里指定了mysql8 加入 bridge 的网络，其它需要使用这个mysql的应用也要加入这个网络
$ find / -name my.cnf
# docker-mysql
docker stop mysql8
docker rm mysql8
docker run --name mysql8 -p 3306:3306  \
--network bridge --privileged=true --restart=always \
-v /usr/local/mysql8/data:/var/lib/mysql \
-v /usr/local/mysql8/conf:/etc/mysql/conf.d/ \
-e TZ=Asia/Shanghai \
-e MYSQL_ROOT_PASSWORD=password -d mysql \
--character-set-server=utf8mb4 --collation-server=utf8mb4_general_ci \
--lower_case_table_names=1 --default-time_zone='+8:00'

# 这样就可以直接使用root账户远程连接了
# 想要创建新账户就进到docker里的mysql应用
$ docker exec -it mysql8 bash


# 创建数据库
create database solo default charset utf8mb4 collate utf8mb4_general_ci;
# 创建专有用户并授权
create user 'solo'@'%' identified by 'userpassword';
grant select,insert,update,references,delete,create,drop,alter,index,create view,show view on solo.* to 'solo'@'%';
# 或者赋予所有权限
grant all on solo.* to 'solo'@'%';
flush privileges;
```

docker 部署solo

```
#!/bin/bash
#
```

```
# Solo docker 更新重启脚本
#
# 1. 请注意修改参数
# 2. 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
docker run --detach --name solo -p 8080:8080 \
         --network=bridge \
                --env RUNTIME_DB="MYSQL" \
                --env JDBC_USERNAME="solo" \
                --env JDBC_PASSWORD="password" \
                --env JDBC_DRIVER="com.mysql.cj.jdbc.Driver" \
                --env JDBC_URL="jdbc:mysql://172.17.0.2:3306/solo?useUnicode=yes&characterE
coding=UTF-8&useSSL=false&serverTimezone=UTC&allowPublicKeyRetrieval=true" \
                 b3log/solo --server_scheme=http --server_host=www.zhuht.xyz

# JDBC_DRIVER 驱动需要根据mysql版本自行确定
# JDBC_URL 中使用了容器内分配的ip地址(使用inspect查看)，也有说可以使用容器名(mysql8)来做
址，但是我试的时候总是不行
# --server_host 有域名的话就设置成域名，不然域名访问solo时，会出现主页地址是域名，子页面
址是ip开头的问题
$ docker inspect mysql8

# solo部署后有时候启动不了，查看日志发现报了 Public Key Retrieval is not allowed错误，晚上
了一下，在JDBC_URL后加了 &&allowPublicKeyRetrieval=true 解决
$ docker logs solo
```

docker 部署 Nginx

```
$ docker pull nginx
# 先创建一个nginx.conf,我这个是直接从nginx的镜像文件里拷出来的，然后在http增加了solo的配置
# 可以直接搜一下这个文件
$ find / -name nginx.conf

# 查到后
user  nginx;
worker_processes  1;

error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
    worker_connections  1024;
}

http {
    default_type  application/octet-stream;

    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                      '$status $body_bytes_sent "$http_referer" '
                      '"$http_user_agent" "$http_x_forwarded_for"';
```

```
    access_log  /var/log/nginx/access.log  main;
    sendfile        on;
    #tcp_nopush     on;
    keepalive_timeout  65;
    #gzip  on;
    include /etc/nginx/conf.d/*.conf;

    # solo
    upstream backend {
        server 172.17.0.3:8080; # Solo 监听端口
    }

    server {
        listen       80;
        server_name  www.zhuht.xyz; # 博客域名
        access_log /var/log/nginx/logs/solo/access.log  main;
        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;
        }
    }
}


# docker 启动nginx
docker run -p 80:80 -m 200m --restart always \
 --name nginx --network bridge \
 -v /usr/local/nginx/conf/:/etc/nginx/ \
 -v /usr/local/nginx/logs:/var/log/nginx/logs \
 nginx
```

docker部署v2ray

```
# 需要先写一个配置文件，通过docker启动时 -config参数使用（懒得写或者不会写的话可以直接本安装，在 /etc/v2ray下回自动创建config.json,这个是可以直接拿来用的）
docker pull v2ray/official
docker stop v2ray
docker rm v2ray
docker run -d --name v2ray \
    -v /etc/v2ray:/etc/v2ray -p 25059:25059 v2ray/official  v2ray \
    -config=/etc/v2ray/config.json

# docker端口与配置文件中的保持一致
```

不使用docker直接安装应用

- 由于这次用了Debian10系统，与原来用的CentOS7不一样，所以安装是也不一样

安装v2ray

```
#修改服务器时区（可不修改，v2ray会自动计算时区，只要保证时间差就行）
$ rm -rf /etc/localtime
$ ln -s /usr/share/zoneinfo/Asia/Shanghai /etc/localtime
# 下载脚本并执行
$ wget https://install.direct/go.sh
$ bash go.sh
# 启动服务
$ systemctl start v2ray
# 开启对应端口
```

安装MySQL

参考

```
$ apt update
$ apt install gnupg
# 找个文件夹存放要下载的文件,然后去MySQL官网找到deb文件的下载地址
$ wget https://dev.mysql.com/get/mysql-apt-config_0.8.13-1_all.deb
# 开始安装
$ dpkg -i mysql-apt-config*
$ apt update
$ apt install mysql-server
```

参考:

[利用 docker 部署 solo 全套:MySQL+Nginx+Solo](https://hacpai.com/article/1556265895068)