

docker分离部署lnmp

原创

Mr大表哥

2017-03-12 11:44:20

评论(0)

711人阅读

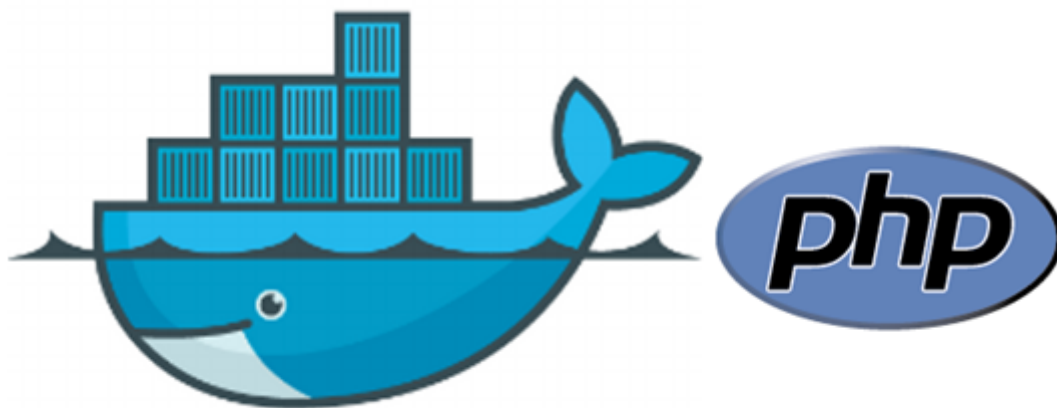
以下所需的所有的文件、镜像、软件，如有需要请到我的百度云分享下载：
链接：<http://pan.baidu.com/s/1kUVNdsj> 密码：an9l

项目需求：

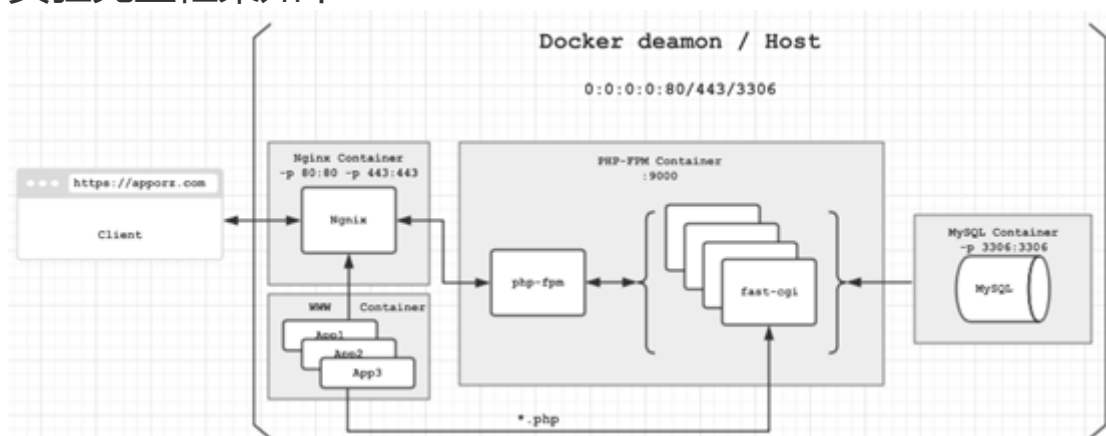
构建lnmp平台。

要求nginx、php、mysql分开部署。

Nginx通过fastcgi方式支持php动态页面



实验完整框架如下：



说明：使用单一进程容器，即一个容器只运行一种服务，而不是把所有服务放在一个容器的设计，让lnmp项目需要的Nginx、PHP、MySQL组件，分别运行在各自镜像创建出来的独立容器中。

实验步骤如下：

- 1、安装docker1.12并开始服务
- 1) 安装docker1.12

```
root@localhost docker1.12] # ls
docker-engine-1.12.0-1.el7.centos.x86_64.rpm docker-engine-selinux-1.12.0-1.el7.centos.noarch.rpm
root@localhost docker1.12] # rpm -ivh *.rpm
警告 : docker-engine-1.12.0-1.el7.centos.x86_64.rpm: 头 V4 RSA/SHA512 Signature, 密钥 ID 2c52609d: NOKEY
准备中...
正在升级/安装...
 1: docker-engine-selinux-1.12.0-1.el7.centos.x86_64.rpm: 头 V4 RSA/SHA512 Signature, 密钥 ID 2c52609d: NOKEY
restorecon: lstat(/var/lib/docker) failed: No such file or directory
警告 : %post(docker-engine-selinux-1.12.0-1.el7.centos.noarch) 脚本执行失败, 退出状态码为 255
 2: docker-engine-1.12.0-1.el7.centos.x86_64.rpm: 头 V4 RSA/SHA512 Signature, 密钥 ID 2c52609d: NOKEY
root@localhost docker1.12] #
```

2) 开启docker服务并开机自启

```
root@localhost docker1.12] # systemctl start docker
root@localhost docker1.12] # systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
root@localhost docker1.12] #
```

3) 关闭selinux (一定要关闭)

```
root@localhost ~] # cat /etc/sysconfig/selinux

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced.
#   permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of three two values:
#   targeted - Targeted processes are protected.
#   minimum - Modification of targeted policy. Only selected processes are protected.
#   mls - Multi Level Security protection.
SELINUXTYPE=targeted

root@localhost ~] #
```

关闭selinux后，不要忘记重启生效，如果不想重启可以采用临时关闭selinux的方法：setenforce 0
该命令只是临时关闭，重启失效，适合做实验，真实生产环境还是采用永久关闭的方法。

4) 开启路由转发功能

```
root@localhost ~] # cat /etc/sysctl.conf
# System default settings live in /usr/lib/sysctl.d/00-system.conf.
# To override those settings, enter new settings here, or in an /etc/sysctl.d
#
# For more information, see sysctl.conf(5) and sysctl.d(5).
net.ipv4.ip_forward=1
root@localhost ~] # sysctl -p
net.ipv4.ip_forward = 1
root@localhost ~] #
```

5) 下载centos6镜像

(我这里已经下载好，并做成了归档压缩包，只用解压即可)

```
root@localhost ~] # docker load -i centos6.tar
2714f4a6cdee: Loading layer [=====] 202.3 MB/202.3 MB
Loaded image: centos:centos6
root@localhost ~] #
```

2、创建实验所用文件夹以及文件

1) 分别创建工作目录

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# mkdir -pv /lnmp/{nginx,php,mysql,web}
mkdir: 已创建目录 "/lnmp"
mkdir: 已创建目录 "/lnmp/nginx"
mkdir: 已创建目录 "/lnmp/php"
mkdir: 已创建目录 "/lnmp/mysql"
mkdir: 已创建目录 "/lnmp/web"
[root@localhost ~]# ls -l /lnmp
总用量 0
drwxr-xr-x 2 root root 6 3月  2 17:12 mysql
drwxr-xr-x 2 root root 6 3月  2 17:12 nginx
drwxr-xr-x 2 root root 6 3月  2 17:12 php
drwxr-xr-x 2 root root 6 3月  2 17:12 web
[root@localhost ~]#
```

2) 再分别创建相应目录下的文件和子目录

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# tree /lnmp
/lnmp
├── docker-compose.yml
├── get-pip.py
├── mysql
│   ├── dockerfile
│   ├── my.cnf
│   ├── software
│   │   ├── cmake-2.8.12.tar.gz
│   │   └── mysql-5.5.38.tar.gz
│   └── supervisord.conf
├── nginx
│   ├── dockerfile
│   ├── nginx.conf
│   ├── software
│   │   └── nginx-1.6.2.tar.gz
│   └── supervisord.conf
├── php
│   ├── dockerfile
│   ├── php-fpm.conf
│   ├── software
│   │   └── php-5.3.28.tar.gz
│   └── supervisord.conf
└── web

7 directories, 15 files
[root@localhost ~]#
```

3、分别编辑nginx、php、mysql的dockerfile文件以及各自的supervisord.conf文件

1) nginx

①编辑nginx的dockerfile文件



```
#images of nginx
FROM centos:centos6
MAINTAINER from zhengpengfei@example.com
#install supervisor
RUN yum -y install python-setuptools
RUN /usr/bin/easy_install supervisor
#install nginx
RUN yum -y install pcre-devel zlib-devel gcc make
ADD ./software/nginx-1.6.2.tar.gz /usr/src
RUN useradd -M -s /sbin/nologin nginx
RUN cd /usr/src/nginx-1.6.2/ && ./configure --prefix=/usr/local/nginx --with-http_stub_status_module --user=nginx --group=nginx && make && make install
#Modify nginx configuration file
COPY nginx.conf /usr/local/nginx/conf/
RUN ln -s /usr/local/nginx/sbin/nginx /usr/local/sbin
RUN mkdir /usr/local/nginx/html/web
#Open nginx service
COPY supervisord.conf /etc/supervisor/supervisord.conf
EXPOSE 80
CMD ["/usr/bin/supervisord"]
```

#images of nginx

FROM centos:centos6

MAINTAINER from zhengpengfei@example.com

#install supervisor

RUN yum -y install python-setuptools

RUN /usr/bin/easy_install supervisor

#install nginx

RUN yum -y install pcre-devel zlib-devel gcc make

ADD ./software/nginx-1.6.2.tar.gz /usr/src

RUN useradd -M -s /sbin/nologin nginx

RUN cd /usr/src/nginx-1.6.2/ && ./configure --prefix=/usr/local/nginx --with-http_stub_status_module --user=nginx --group=nginx && make && make install

#Modify nginx configuration file

COPY nginx.conf /usr/local/nginx/conf/

RUN ln -s /usr/local/nginx/sbin/nginx /usr/local/sbin

RUN mkdir /usr/local/nginx/html/web

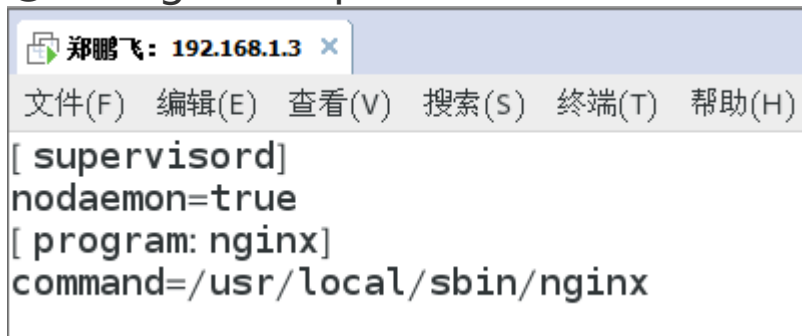
#Open nginx service

COPY supervisord.conf /etc/supervisor/supervisord.conf

EXPOSE 80

CMD ["/usr/bin/supervisord"]

②编写nginx的supervisord.conf配置文件



```
[supervisord]
nodaemon=true
[program:nginx]
command=/usr/local/sbin/nginx
```

③docker build -t命令制作nginx镜像



```
[root@localhost ~]# docker build -t=lnmp:nginx /lnmp/nginx/
Sending build context to Docker daemon 812.5 kB
Step 1 : FROM centos:centos6
--> cf2c3ece5e41
Step 2 : MAINTAINER from zhengpengfei@example.com
--> Running in bcc44aee3891
```

④镜像制作完成

Successfully built dd49a918344f

[root@localhost ~] # docker images

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lnmp	nginx	dd49a918344f	9 minutes ago	346.7 MB
centos	centos6	cf2c3ece5e41	8 months ago	194.6 MB

[root@localhost ~] #

2) php

①编辑php的dockerfile文件

```
#images of php
FROM centos:centos6
MAINTAINER from zhengpengfei@example.com
#install supervisor
RUN yum -y install python-setuptools
RUN /usr/bin/easy_install supervisor
#install php
RUN yum -y install gd libxml2-devel libjpeg-devel libpng-devel mysql-devel gcc make
RUN useradd -M -s /sbin/nologin php
ADD ./software/php-5.3.28.tar.gz /usr/src
RUN cd /usr/src/php-5.3.28/
RUN cp /usr/lib64/mysql/libmysqlclient.so.16.0.0 /usr/lib/libmysqlclient.so
RUN cd /usr/src/php-5.3.28/ && ./configure --prefix=/usr/local/php --with-gd --with-zlib --with-mysql --with-mysqli --with-mysql-sock --with-config-file-path=/usr/local/php --enable-mbstring --enable-fpm --with-jpeg-dir=/usr/lib && make && make install
#Modify PHP configuration file
RUN cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf
COPY php-fpm.conf /usr/local/php/etc/
RUN mkdir -p /var/www/html/web
#Open php-fpm service
ADD ./supervisord.conf /etc/supervisor/supervisord.conf
EXPOSE 9000
CMD ["/usr/bin/supervisord"]
```

#images of php

FROM centos:centos6

MAINTAINER from zhengpengfei@example.com

#install supervisor

RUN yum -y install python-setuptools

RUN /usr/bin/easy_install supervisor

#install php

RUN yum -y install gd libxml2-devel libjpeg-devel libpng-devel mysql-devel gcc make

RUN useradd -M -s /sbin/nologin php

ADD ./software/php-5.3.28.tar.gz /usr/src

RUN cd /usr/src/php-5.3.28/

RUN cp /usr/lib64/mysql/libmysqlclient.so.16.0.0 /usr/lib/libmysqlclient.so

RUN cd /usr/src/php-5.3.28/ && ./configure --prefix=/usr/local/php --with-gd --with-zlib --with-mysql --with-mysqli --with-mysql-sock --with-config-file-path=/usr/local/php --enable-mbstring --enable-fpm --with-jpeg-dir=/usr/lib && make && make install

#Modify PHP configuration file

RUN cp /usr/local/php/etc/php-fpm.conf.default /usr/local/php/etc/php-fpm.conf

COPY php-fpm.conf /usr/local/php/etc/

RUN mkdir -p /var/www/html/web

#Open php-fpm service

ADD ./supervisord.conf /etc/supervisor/supervisord.conf

EXPOSE 9000

CMD ["/usr/bin/supervisord"]

②编写php的supervisord.conf配置文件

```
[supervisord]
nodaemon=true
[program:php]
command=/usr/local/php/sbin/php-fpm
```

③docker build -t命令制作php镜像

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# docker build -t=lnmp:php /lnmp/php/
Sending build context to Docker daemon 14.02 MB
Step 1 : FROM centos:centos6
--> cf2c3ece5e41
Step 2 : MAINTAINER from 819594300@qq.com
--> Running in 3286e951af05
--> 549c643b1fea

```

④镜像制作完成

```

Successfully built cd8067c79a21
[root@localhost ~]# docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lnmp	php	cd8067c79a21	2 minutes ago	632.5 MB
lnmp	nginx	dd49a918344f	34 minutes ago	346.7 MB
centos	centos6	cf2c3ece5e41	8 months ago	194.6 MB

```

[root@localhost ~]#

```

3) mysql

①编辑mysql的dockerfile文件

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
#image of mysql
FROM centos:centos6
MAINTAINER from zhengpengfei@example.com
#install supervisor
RUN yum -y install python-setuptools
RUN /usr/bin/easy_install supervisor
#install mysql
RUN yum -y install ncurses-devel make gcc gcc-c++
ADD ./software/cmake-2.8.12.tar.gz /usr/src
ADD ./software/mysql-5.5.38.tar.gz /usr/src
RUN cd /usr/src/cmake-2.8.12 && ./configure && make && make install
RUN cd /usr/src/mysql-5.5.38 && cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -DSYSCONFDIR=/etc/ -DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci -DWITH_EXTRA_CHARSETS=all && make && make install
#Optimal adjustment mysql
WORKDIR /usr/src/mysql-5.5.38/
RUN cp -rf ./support-files/my-medium.cnf /etc/my.cnf
RUN cp -rf ./support-files/mysql.server /etc/rc.d/init.d/mysqld
RUN chmod +x /etc/rc.d/init.d/mysqld
RUN echo "PATH=$PATH:/usr/local/mysql/bin" >> /etc/profile
RUN source /etc/profile
#Initialize mysql
RUN groupadd mysql
RUN useradd -M -s /sbin/nologin mysql -g mysql
RUN chown -R mysql:mysql /usr/local/mysql
RUN /usr/local/mysql/scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --datadir=/usr/local/mysql/data/
#service mysql start
ADD ./supervisord.conf /etc/supervisor/supervisord.conf
EXPOSE 3306
CMD ["/usr/bin/supervisord"]

```

#image of mysql

FROM centos:centos6

MAINTAINER from zhengpengfei@example.com

#install supervisor

RUN yum -y install python-setuptools

RUN /usr/bin/easy_install supervisor

#install mysql

RUN yum -y install ncurses-devel make gcc gcc-c++

ADD ./software/cmake-2.8.12.tar.gz /usr/src

ADD ./software/mysql-5.5.38.tar.gz /usr/src

RUN cd /usr/src/cmake-2.8.12 && ./configure && make && make install

RUN cd /usr/src/mysql-5.5.38 && cmake -DCMAKE_INSTALL_PREFIX=/usr/local/mysql -

DSYSCONFDIR=/etc/ -DDEFAULT_CHARSET=utf8 -DDEFAULT_COLLATION=utf8_general_ci -

DWITH_EXTRA_CHARSETS=all && make && make install

#Optimal adjustment mysql

WORKDIR /usr/src/mysql-5.5.38/

RUN cp -rf ./support-files/my-medium.cnf /etc/my.cnf

RUN cp -rf ./support-files/mysql.server /etc/rc.d/init.d/mysqld

RUN chmod +x /etc/rc.d/init.d/mysqld

RUN echo "PATH=\$PATH:/usr/local/mysql/bin" >> /etc/profile

RUN source /etc/profile

#Initialize mysql

```

RUN groupadd mysql
RUN useradd -M -s /sbin/nologin mysql -g mysql
RUN chown -R mysql:mysql /usr/local/mysql
RUN /usr/local/mysql/scripts/mysql_install_db --user=mysql --basedir=/usr/local/mysql --
datadir=/usr/local/mysql/data/
#service mysql start
ADD ./supervisord.conf /etc/supervisor/supervisord.conf
EXPOSE 3306
CMD ["/usr/bin/supervisord"]

```

②编写mysql的supervisord.conf配置文件

```

[ supervisord ]
nodaemon=true
[ program: mysqld ]
command=/etc/init.d/mysqld start

```

③docker build -t命令制作mysql镜像

```

[ root@localhost ~ ] # docker build -t=lnmp:mysql /lnmp/mysql/
Sending build context to Docker daemon 27.81 MB
Step 1 : FROM centos:centos6
---> cf2c3ece5e41
Step 2 : MAINTAINER from zhengpengfei@example.com
---> Using cache
---> 87179d075b18
Step 3 : RUN yum -y install python-setuptools
---> Using cache
---> 4be6d98d9d39

```

④镜像制作完成

```

Successfully built 9236044224b2
[ root@localhost ~ ] # docker images

```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
lnmp	mysql	9236044224b2	46 seconds ago	1.889 GB
lnmp	php	cd8067c79a21	28 minutes ago	632.5 MB
lnmp	nginx	dd49a918344f	About an hour ago	346.7 MB
centos	centos6	cf2c3ece5e41	8 months ago	194.6 MB

```

[ root@localhost ~ ] #

```

4、编写docker-compose.yml文件

```

[ root@localhost lnmp ] # ls
mysql  nginx  php  web
[ root@localhost lnmp ] # vim docker-compose.yml

```



```

1 mysql:
2     image: lnmp: mysql
3     volumes:
4     - /lnmp/mysql/my.cnf: /etc/my.cnf
5     expose:
6     - 3306
7 php:
8     image: lnmp: php
9     volumes:
10    - /lnmp/php/php-fpm.conf: /usr/local/php/etc/php-fpm.conf
11    - /lnmp/web/: /var/www/html/web/
12    links:
13    - mysql
14    expose:
15    - 9000
16 nginx:
17     image: lnmp: nginx
18     volumes:
19     - /lnmp/nginx/nginx.conf: /usr/local/nginx/conf/nginx.conf
20     - /lnmp/web/: /usr/local/nginx/html/web/
21     links:
22     - php
23     - mysql
24     ports:
25     - "80:80"

```

5、安装docker-compose

1)先安装pip

```

root@localhost lnmp# wget https://bootstrap.pypa.io/get-pip.py
--2017-03-02 19:11:47-- https://bootstrap.pypa.io/get-pip.py
正在解析主机 bootstrap.pypa.io (bootstrap.pypa.io)... 151.101.88.175
正在连接 bootstrap.pypa.io (bootstrap.pypa.io) | 151.101.88.175 | 443... 已连接。
已发出 HTTP 请求，正在等待响应... 200 OK
长度：1595408 (1.5M) [text/x-python]
正在保存至: "get-pip.py"

100% |=====| 1,595,408 633KB/s 用时 2.5s

2017-03-02 19:11:51 (633 KB/s) - 已保存 "get-pip.py" [1595408/1595408]

root@localhost lnmp#

```

```

[ root@localhost lnmp] # python get-pip.py
Collecting pip
  Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB)
    100% |#####| 1.3MB 843kB/s
Collecting wheel
  Downloading wheel-0.29.0-py2.py3-none-any.whl (66kB)
    100% |#####| 71kB 11.0MB/s
Installing collected packages: pip, wheel
Successfully installed pip-9.0.1 wheel-0.29.0
[ root@localhost lnmp] #

```

2) 再安装compose

```

[ root@localhost lnmp] # pip install docker-compose
Collecting docker-compose
  Downloading docker_compose-1.11.2-py2.py3-none-any.whl (85kB)
    100% |#####| 92kB 526kB/s

```



```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost lnmp]# ln -s /usr/bin/docker-compose /usr/local/bin/
[root@localhost lnmp]# docker-compose -v
docker-compose version 1.11.2, build dfed245
[root@localhost lnmp]#
```

6、通过docker-compose启动项目

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost lnmp]# docker-compose up -d
Creating lnmp_mysql_1
Creating lnmp_php_1
Creating lnmp_nginx_1
[root@localhost lnmp]# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
8e17609f0fff       lnmp_nginx         "/usr/bin/supervisord" 20 seconds ago      Up 10 seconds      0.0.0.0:80->80/tcp   lnmp_nginx_1
256c71b694a2       lnmp_php           "/usr/bin/supervisord" 24 seconds ago      Up 20 seconds      9000/tcp            lnmp_php_1
cbe0571cbbc1       lnmp_mysql         "/usr/bin/supervisord" 26 seconds ago      Up 24 seconds      3306/tcp            lnmp_mysql_1
[root@localhost lnmp]#
```

7、进入mysql容器修改数据库root密码以及创建数据库和创建授权用户

1) 修改数据库用户root密码

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# docker exec -it lnmp_mysql_1 /bin/bash
<5.38] # /usr/local/mysql/bin/mysqladmin -uroot -p password '123456'
Enter password:
[root@cbe0571cbbc1 mysql- 5.5.38] #
```

2) 创建数据库

```
<5.38] # /usr/local/mysql/bin/mysql -uroot -p'123456'
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.5.38-log Source distribution

Copyright (c) 2000, 2014, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database movie;
Query OK, 1 row affected (0.00 sec)

mysql>
```

3) 创建授权用户

```
mysql> grant all privileges on movie.* to 'zpf'@%' identified by '123456';
Query OK, 0 rows affected (0.02 sec)

mysql>
```

4) 给root用户授予全部权限

```
mysql> grant all privileges on *.* to 'root'@%' identified by '123456';
Query OK, 0 rows affected (0.00 sec)

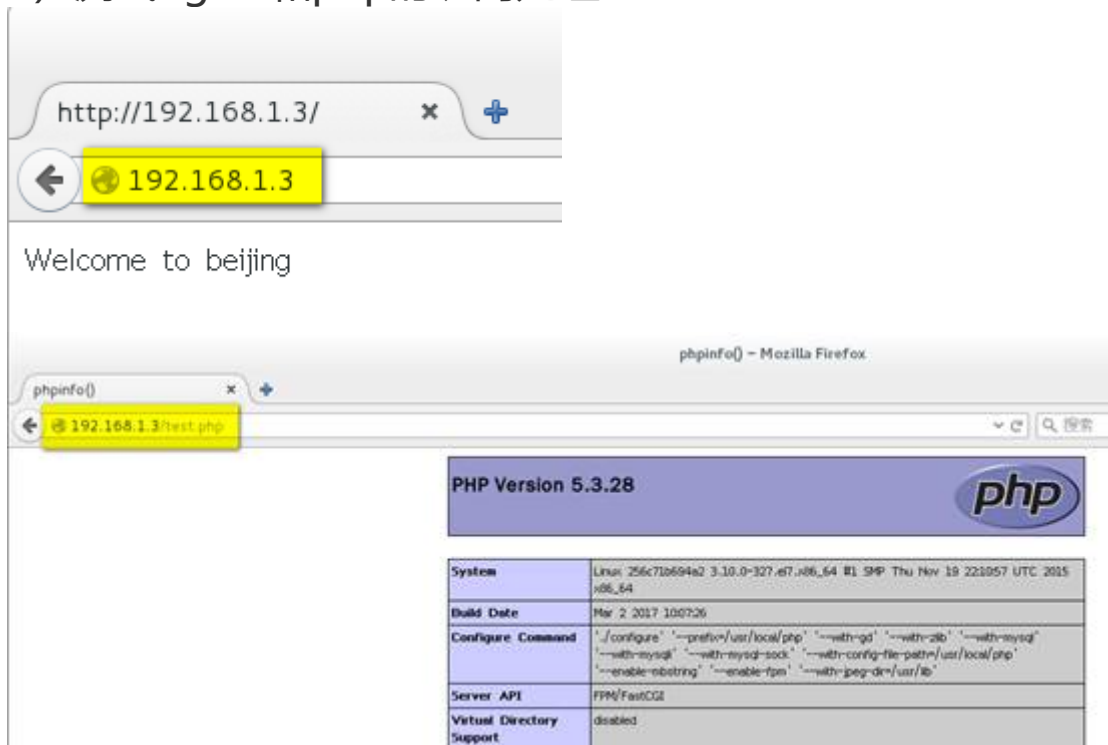
mysql>
```

8、做html、php页面和数据库访问测试

1) 制作html、php测试页

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[root@localhost ~]# cd /lnmp/web/
[root@localhost web]# cat index.html
Welcome to beijing
[root@localhost web]# cat test.php
<?php
phpinfo();
?>
[root@localhost web]#
```

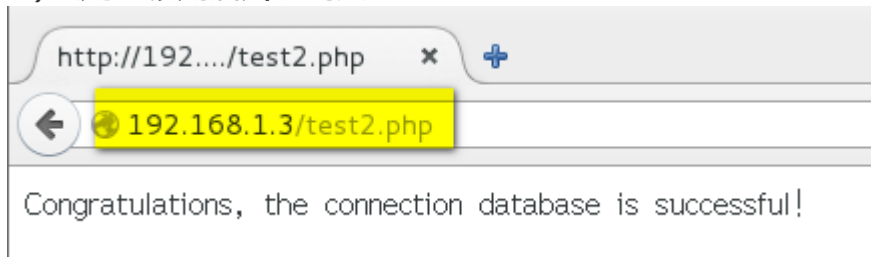
2) 测试nginx和php的访问处理



3) 制作数据库的测试页面

```
郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
<?php
$link=mysql_connect('172.17.0.2','root','123456');
if($link) echo"Congratulations, the connection database is successful!";
mysql_close();
?>
```

4) 测试数据库连接



至此说明nginx、php、mysql三者的协同工作已经没有问题了

8、安装一个电影网站，做最后的lnmp协同工作测试

1) 解压缩SKYUC

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost lnmp] # ls
docker-compose.yml  get-pip.py  mysql  nginx  php  SKYUC_3.4.2_for_php5.3.zip  web
[ root@localhost lnmp] # rpm -q unzip
unzip-6.0-15.el7.x86_64
[ root@localhost lnmp] # unzip SKYUC_3.4.2_for_php5.3.zip

```

2) 设置权限

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost lnmp] # cd SKYUC.v3.4.2.SOURCE/wwwroot/
[ root@localhost wwwroot] # /bin/cp -r * /lnmp/web/
[ root@localhost wwwroot] # cd /lnmp/web/

```

分别去nginx和php容器给予权限:

php:

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost ~] # docker exec -it lnmp_php_1 /bin/bash
[ root@256c71b694a2 /] # cd /var/www/html/web/
[ root@256c71b694a2 web] # chown -R php:php admincp/ data/ templates/ upload/
[ root@256c71b694a2 web] #

```

nginx:

```

郑鹏飞: 192.168.1.3 x
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost ~] # docker exec -it lnmp_nginx_1 /bin/bash
[ root@8e17609f0fff /] # cd /usr/local/nginx/html/web/
[ root@8e17609f0fff web] # chown -R nginx:nginx admincp/ data/ templates/ upload/
[ root@8e17609f0fff web] #

```

3) 宿主机防火墙开启80例外

```

郑鹏飞: 192.168.1.3 x
应用程序 位置 终端
root@localhost ~
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[ root@localhost ~] # firewall-cmd --permanent --add-port=80/tcp
success
[ root@localhost ~] # firewall-cmd --reload
success
[ root@localhost ~] #

```

4) 在一台客户机安装SKYUC

欢迎访问SKYUC! (天空网络电影系统) 192.168.1.3/install/index.php

SKYUC!

界面语言: 简体中文 繁体中文

© 2012 天空网络, 保留所有权利。

感谢您选择 SKYUC! (天空网络电影系统), 希望我们的努力能为您提供一个高效快速和强大的视频点播解决方案。SKYUC! 英文全称为 SKYUC MOVIE SYSTEM, 中文全称为 天空网络电影系统。

天空网络为 SKYUC! 产品的开发商, 依法独立拥有 SKYUC! 产品著作权。天空网络官方网站为 <http://www.skyuc.com>

本协议适用于且仅适用于 SKYUC! 3.x 版本。天空网络拥有对本协议协议的最终解释权。

I. 协议许可的权利

- 您可以在天空网络本系统用户授权协议的基础上, 将本软件应用于商业用途, 而不会支付费用。
- 您可以在协议规定的约束和限制范围内修改 SKYUC! 源代码(如果提供的话)或界面风格以适配您的网站需求。
- 您得将使用本软件构建的网站中全部会员资料、文章及相关信息所有权利, 并独立承担与其内容的相关法律责任。
- 获得商业授权之前, 您可将本软件应用于商业用途, 但同时必须购买本系统所规定的技术支持期限, 技术支持方式和范围, 自授权时开始, 在技术支持期限内拥有通过指定的方式获得规定范围内的技术支持服务。商业授权用户享有反馈和提出意见的权利, 相关意见将被作为重要考虑, 但没有一定被采纳的承诺或保证。

II. 协议规定的约束和限制

- 未获商业授权之前, 不得将本软件用于商业用途(包括但不限于企业网站、经营性网站、以营利为目的或实现营利的网站)。
- 不得对本软件或与之关联的商业授权进行出租、出借、转授或发放于许可证。
- 无论如何, 即无明示或如何, 且未经过修改或美化、修改程度如何, 只要使用 SKYUC! 的商标或任何部分, 未经书面许可, 网站页面或部分的 SKYUC! 名称和天空网络(<http://www.skyuc.com>) 的链接都必须保留, 而不能删除或修改。

☒ 我已仔细阅读, 并同意上述条款中的所有内容

© 2012 天空网络, 保留所有权利。

下一步: 检测系统环境

系统环境

操作系统	Linux
PHP 版本	5.3.28
MySQL 版本	5.1.73
图像处理库	GD
重要文件是否完整	<input checked="" type="checkbox"/> 完整 <input type="checkbox"/> 关闭
服务器是否开启安全模式	<input type="checkbox"/> 安全 <input checked="" type="checkbox"/> 关闭

目录权限检测

admincp	可写
upload/articleimg	可写
upload/article	可写
upload/feedbacking	可写
upload/posters	可写
data	可写
data/caches	可写
data/compiled	可写
data/compiled/admincp	可写
data/images	可写
data/sqldata	可写
templates	可写
templates/backup	可写
templates/backup/library	可写
templates/default	可写

模板可写性检查

所有模板, 全部可写

© 2012 天空网络, 保留所有权利。

上一步: 欢迎页 重新检查 下一步: 配置系统

数据库帐号

数据库主机:	172.17.0.2
端口号:	3306
用户名:	apf
密码:	●●●●●●
数据库名:	movie <input checked="" type="checkbox"/> movie <input type="checkbox"/> 按
表前缀:	skyuc_
数据库类型:	<input checked="" type="radio"/> MySQL <input type="radio"/> MySQL

管理员帐号

管理员姓名:	admin
登录密码:	●●●●●●
密码确认:	●●●●●●
电子邮箱:	819594300@qq.com

安装

选择语言包: ☒ 简体中文 ☐ 繁体中文

禁用验证码: ☒ (选择此项, 进入后台、发表评论无需验证)

© 2012 天空网络, 保留所有权利。

上一步: 检测系统环境 立即安装

1. 欢迎使用

2. 检测环境

3. 配置系统

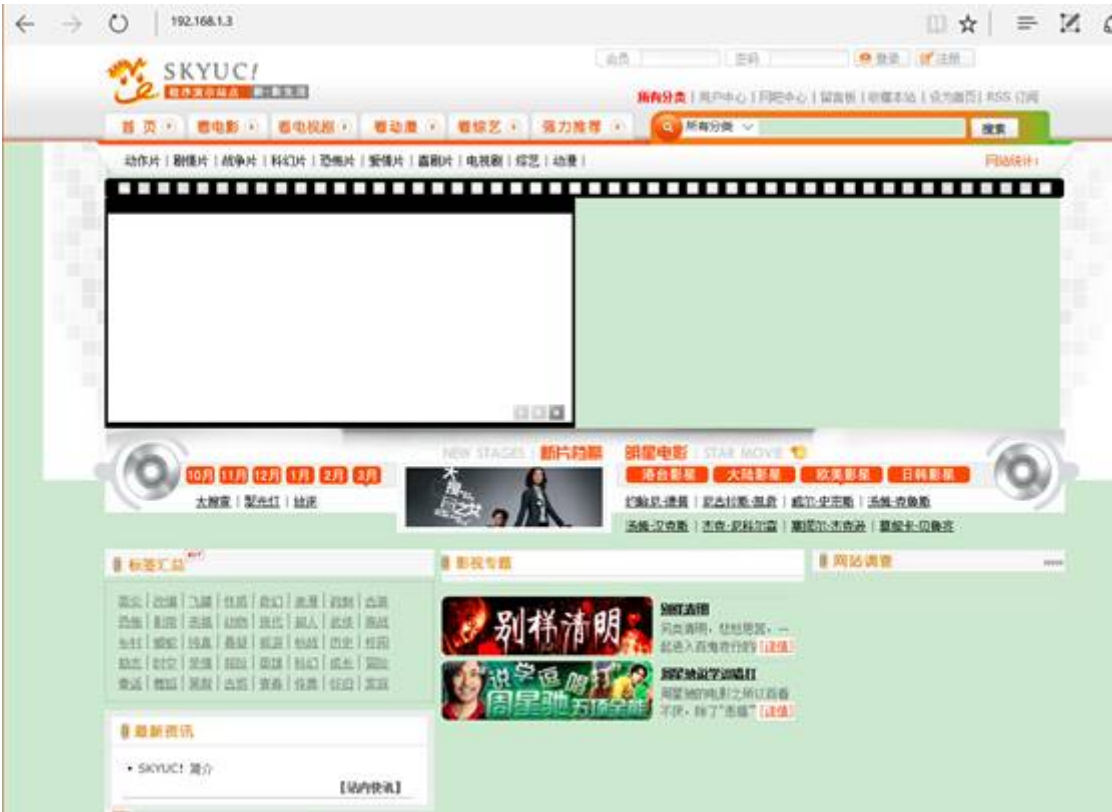
只需三步即可轻松完成安装!

只需三步即可轻松完成安装!

只需三步即可轻松完成安装!

<http://blog.51cto.com/zpf666/1905555>

12/13



版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任