

## 26 让 LNMP 运行起来

更新时间：2020-03-10 10:14:22



“

最聪明的人是最不愿浪费时间的人。——但丁

”

### 前言

我们在前面介绍了很多理论知识，这一节我们要实战一下，从头到尾安装一个 LNMP 环境。只有自己亲自动手才能遇到各种问题，在解决问题的过程中加深大家的理解。

废话不多说，我们直接开撸.....

### 环境准备

我们不要求大家有任何特殊的東西，只用一个电脑（哈哈，这是废话），一个带有 centos 镜像的 docker 环境，仅此而已。

### 直接开始

首先看一下我们机器上面有什么 docker 镜像。

```
docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
nginx-1.16.1	latest	24ed65aaaab3	4 months ago	684MB
centos	latest	67fa590cfc1c	6 months ago	202MB
harisekhon/hbase	latest	856957168a1c	7 months ago	416MB

启动 centos

使用 `docker run` 命令启动我们的 `centos` 镜像

```
docker run -dit centos
d510809aef7241621ac21ce4741b761985a790dfebb4c15384ee760ecc4a54b6
```

我们进入到 `centos` 镜像，从此就相当于一个完全新的操作系统，可以随便我们折腾.....

```
docker exec -it d510809aef72 /bin/bash
[root@d510809aef72 /]#
```

## 安装 `Nginx`

在安装 `nginx` 之前，有一些依赖文件要安装，比如我们前面提到的 `openssl` 等。

使用 `yum` 命令安装这些依赖文件。

```
[root@d510809aef72 /]# yum install gcc gcc-c++ openssl openssl-devel
Loaded plugins: fastestmirror, ovl
Determining fastest mirrors
 * base: mirrors.tuna.tsinghua.edu.cn
 * extras: mirrors.tuna.tsinghua.edu.cn
 * updates: mirrors.tuna.tsinghua.edu.cn
base | 3.6 kB | 00:00:00
extras | 2.9 kB | 00:00:00
updates | 2.9 kB | 00:00:00
(1/4): base/7/x86_64/group_gz | 165 kB | 00:00:00
(2/4): extras/7/x86_64/primary_db | 164 kB | 00:00:00
(3/4): updates/7/x86_64/primary_db | 6.7 MB | 00:00:01
(4/4): base/7/x86_64/primary_db | 6.0 MB | 00:00:02
Resolving Dependencies
--> Running transaction check
--> Package gcc.x86_64 0:4.8.5-39.el7 will be installed
--> Processing Dependency: libgomp = 4.8.5-39.el7 for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: cpp = 4.8.5-39.el7 for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: libgcc >= 4.8.5-39.el7 for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: glibc-devel >= 2.2.90-12 for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: libmpfr.so.4()(64bit) for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: libmpc.so.3()(64bit) for package: gcc-4.8.5-39.el7.x86_64
--> Processing Dependency: libgomp.so.1()(64bit) for package: gcc-4.8.5-39.el7.x86_64
```

我们首先创建一个目录 `/root/lnmp/` 用于保存我们的所有源代码文件。

```
[root@d510809aef72 /]# mkdir root/lnmp
[root@d510809aef72 /]# cd root/lnmp/
```

下载 `nginx` 源代码

```
[root@d510809aef72 lnmp]# wget http://nginx.org/download/nginx-1.16.1.tar.gz
--2020-03-08 03:46:07-- http://nginx.org/download/nginx-1.16.1.tar.gz
Resolving nginx.org (nginx.org)... 62.210.92.35, 95.211.80.227, 2001:1af8:4060:a004:21::e3
Connecting to nginx.org (nginx.org)|62.210.92.35|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1032630 (1008K) [application/octet-stream]
Saving to: 'nginx-1.16.1.tar.gz'

2% [>] 27,717 6.59KB/s eta 2m 6s
```

有的朋友的机器 `centos` 镜像可能没有 `wget` 工具，我们可以使用 `wget` 命令安装一个

解压 **Nginx** 源码，

将下载的压缩文件解压到本地目录：

```
[root@d510809aef72 lnmp]# tar -zxvf nginx-1.16.1.tar.gz
nginx-1.16.1/
nginx-1.16.1/auto/
nginx-1.16.1/conf/
nginx-1.16.1/contrib/
nginx-1.16.1/src/
nginx-1.16.1/configure
nginx-1.16.1/LICENSE
nginx-1.16.1/README
nginx-1.16.1/html/
nginx-1.16.1/man/
nginx-1.16.1/CHANGES.ru
nginx-1.16.1/CHANGES
```

### 编译源码

有很多朋友对 **Linux** 系统安装软件非常的惧怕，其实这是没有必要的。因为我们平时使用 **Windows** 系统安装软件非常的方便，只用点击 **同意**，然后一路的 **下一步** 就可以了，因为这是一个图形用户界面，所以对用户很友好。其实在 **Linux** 中安装软件也是非常的固定化的，分为三个步骤，**configure**，**make**，**make install**，也是非常简单的。

**configure**

下面我们就安装一下 `nginx`，我们进入到解压文件之后，可以看到一个 `configure` 文件，顾名思义，这就是一个配置文件。其实它提供了非常详细的用法，我们通过执行 `configure --help` 命令就可以看到：

```
[root@d510809aef72 nginx-1.16.1]#  
[root@d510809aef72 nginx-1.16.1]# ./configure --help  
  
--help                print this message  
  
--prefix=PATH          set installation prefix  
--sbin-path=PATH       set nginx binary pathname  
--modules-path=PATH    set modules path  
--conf-path=PATH       set nginx.conf pathname  
--error-log-path=PATH  set error log pathname  
--pid-path=PATH        set nginx.pid pathname  
--lock-path=PATH       set nginx.lock pathname  
  
--user=USER            set non-privileged user for  
                      worker processes  
--group=GROUP          set non-privileged group for  
                      worker processes  
  
--build=NAME           set build name  
--builddir=DIR         set build directory
```

网上有很多文章介绍我们在配置 `configure` 的时候要传递这个参数，那个参数的，其实没有必要，对于我们学习使用而言，使用默认值安全满足要求。

```
[root@d510809aef72 nginx-1.16.1]# ./configure  
checking for OS  
+ Linux 4.19.76-linuxkit x86_64  
checking for C compiler ... found
```

下面是配置之后的检测输出：

```
Configuration summary  
+ using system PCRE library  
+ OpenSSL library is not used  
+ using system zlib library  
  
nginx path prefix: "/usr/local/nginx"  
nginx binary file: "/usr/local/nginx/sbin/nginx"  
nginx modules path: "/usr/local/nginx/modules"  
nginx configuration prefix: "/usr/local/nginx/conf"  
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"  
nginx pid file: "/usr/local/nginx/logs/nginx.pid"  
nginx error log file: "/usr/local/nginx/logs/error.log"  
nginx http access log file: "/usr/local/nginx/logs/access.log"  
nginx http client request body temporary files: "client_body_temp"  
nginx http proxy temporary files: "proxy_temp"  
nginx http fastcgi temporary files: "fastcgi_temp"  
nginx http uwsgi temporary files: "uwsgi_temp"  
nginx http scgi temporary files: "scgi_temp"
```

编译和安装

这一步只用执行 `make` 和 `make install` 即可，没有任何其他的操作，我在这里就不贴图了。

测试

我们上面提到过，在 `configure` 的最后会生成一个 `summary`，里面包含了 `nginx` 的安装路径，配置文件路径等等。

```
[root@d510809aef72 nginx-1.16.1]# /usr/local/nginx/sbin/nginx
[root@d510809aef72 nginx-1.16.1]# ps -ef | grep nginx
root      2706      1  0 04:10 ?        00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nobody    2707    2706  0 04:10 ?        00:00:00 nginx: worker process
root      2727     15  0 04:11 pts/1    00:00:00 grep --color=auto nginx
[root@d510809aef72 nginx-1.16.1]# curl http://localhost/index.html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

启动 nginx

测试是否正常

这里说明我们的 `nginx` 已经安装成功了，`so easy`

## 安装 `php`

首先我们选择安装的 `php` 版本。现在已经是 `php7` 的天下了，所以我们选择了 `php7.3` 版本。

下载并解压源代码

我们可以从这个地方下载 `php7.3` 的源代码，然后解压。

```
[root@d510809aef72 lnmp]#
[root@d510809aef72 lnmp]# cd php-7.3.15
[root@d510809aef72 php-7.3.15]# ll
total 5520
-rw-r--r--  1 1000 1000   12727 Feb 18 09:25 CODING_STANDARDS
-rw-r--r--  1 1000 1000    3430 Feb 18 09:25 CONTRIBUTING.md
-rw-r--r--  1 1000 1000     93 Feb 18 09:25 CREDITS
-rw-r--r--  1 1000 1000   23571 Feb 18 09:25 EXTENSIONS
-rw-r--r--  1 1000 1000    105 Feb 18 09:25 INSTALL
-rw-r--r--  1 1000 1000    3204 Feb 18 09:25 LICENSE
-rw-r--r--  1 root root  160953 Mar  8 04:33 Makefile
-rw-r--r--  1 1000 1000    1087 Feb 18 09:25 Makefile.frag
-rw-r--r--  1 root root   13283 Mar  8 04:33 Makefile.fragments
-rw-r--r--  1 1000 1000    2483 Feb 18 09:25 Makefile.gcov
-rw-r--r--  1 1000 1000    7116 Feb 18 09:25 Makefile.global
-rw-r--r--  1 root root  124558 Mar  8 04:33 Makefile.objects
-rw-r--r--  1 1000 1000   79339 Feb 18 09:25 NEWS
```

## 配置

这里也是执行 `configure` 命令，但是我们要给 `configure` 传递几个参数，如下：

```
./configure --prefix=/usr/local/php-fpm --enable-fpm --with-config-file-path=/usr/local/php-fpm/etc
```

这里的 `--enable-fpm` 的作用就是打开 `fpm` 功能。

## 编译并安装

和 `nginx` 一模一样，我们执行 `make` 和 `make install` 就可以安装 `php` 了。

安装成功之后我们可以看一下安装结果：

```
[root@d510809aef72 lnmp]# php -v
PHP 7.3.15 (cli) (built: Mar  8 2020 04:43:50) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.3.15, Copyright (c) 1998-2018 Zend Technologies
```

写一个简单的脚本测试一下安装成果。

```
[root@d510809aef72 lnmp]#
[root@d510809aef72 lnmp]# cat test.php
<?php
echo "hello php\n";
[root@d510809aef72 lnmp]# php test.php
hello php
[root@d510809aef72 lnmp]#
```

测试脚本

执行结果

好嘞，我们的 `php` 安装成功了.....

## 配置 `Nginx` 和 `php`

### 启动 `php-fpm`

这里我们要了解 `php-fpm` 的工作原理，非常的简单，大家可以自行搜索。

首先我们要启动 `php-fpm`，我们首先找到 `php-fpm` 的配置文件所在的目录 `/usr/local/php-fpm/etc/`，该目录下面有一个 `php-fpm.conf.default` 的文件，我们拷贝一份这个文件，重新命名为 `php-fpm.conf`。我们打开这个文件，最后一行代码：

```
include=/usr/local/php-fpm/etc/php-fpm.d/*.conf
```

所以看出来，真正的配置应该在 `/usr/local/php-fpm/etc/php-fpm.d/` 目录下面。进入这个目录，拷贝一份 `www.conf.default`，重新命名为 `www.conf`，打开这个文件，将 `pm=dynamic` 修改为 `pm.static`，保存退出。

这样我们就可以使用命令 `/usr/local/php-fpm/sbin/php-fpm` 启动 `php-fpm` 了。

```
[root@d510809aef72 php-fpm.d]# ps -ef | grep php-fpm
root      20104      1  0 06:13 ?        00:00:00 php-fpm: master process (/usr/local/php-fpm/etc/php-fpm.conf)
nobody    20105  20104  0 06:13 ?        00:00:00 php-fpm: pool www
nobody    20106  20104  0 06:13 ?        00:00:00 php-fpm: pool www
nobody    20107  20104  0 06:13 ?        00:00:00 php-fpm: pool www
nobody    20108  20104  0 06:13 ?        00:00:00 php-fpm: pool www
nobody    20109  20104  0 06:13 ?        00:00:00 php-fpm: pool www
```



## 配置 nginx

我们打开 `nginx` 的配置文件，在 `server` 配置部分增加如下 `location` 配置。

```
location ~ /\.php$ {
    root /root/php/;
    fastcgi_pass 127.0.0.1:9000;
    fastcgi_index index.php;
    fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    include fastcgi_params;
}
```

`php-fpm` 默认监听 `9000` 端口，我们可以通过 `netstat` 命令查看：

```
[root@d510809aef72 php-7.3.15]# netstat -tlnp | grep fpm
tcp        0      0 127.0.0.1:9000      0.0.0.0:*           LISTEN      20104/php-fpm: mast
[root@d510809aef72 php-7.3.15]#
```

重启 `Nginx`，然后我们在 `/root/php/` 下面创建一个 `index.php` 文件，非常的简单：

```
<?php
echo "hello from php-fpm\n";
```

访问 `http://localhost/index.php` 可以看到输出结果：

```
[root@d510809aef72 /]# curl -i http://localhost/index.php
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Sun, 08 Mar 2020 06:46:01 GMT
Content-Type: text/html; charset=UTF-8
Transfer-Encoding: chunked
Connection: keep-alive
X-Powered-By: PHP/7.3.15

hello from php-fpm
[root@d510809aef72 /]#
```

说明我们的 `Nginx` 已经和 `php` 正常的进行了通信，安装成功。

我们访问 `php` 文件的时候可能会出现 `File not found` 的问题，这个时候看一下 `error.log` 可以发现，

```
[error] 20146#0: *7 FastCGI sent in stderr: "Primary script unknown" while reading response header from
upstream, client: 127.0.0.1, server: localhost, request: "GET /index.php HTTP/1.1", upstream: "fastcgi://127.0.0.1:9000", ho
st: "localhost"
```

这个问题十有八九是因为我们存放 `php` 文件的目录权限造成了，大家修改一下对应的权限就可以了。

## 安装 MySQL

这个过程就不再展示了，和 `Nginx` 以及 `php` 的安装过程一模一样。大家可以试着自己安装。

总结

我们在这篇文章中手把手的教大家如何安装 **LNMP** 系统，大家一定要亲自动手尝试，切勿眼高手低。有的朋友可能一直在写业务层代码，对于这些东西不熟悉，认为这是 **OP** 的工作，其实不然，这对于每一个开发同学来说都非常的重要。遇到问题的时候我们可以准确快速的定位。

```
}
```

[25 LNMP 初体验](#)[27 Nginx 的防盗链](#)