

LNMP-环境搭建

0x00 环境准备：

操作系统： centos7 X64 位系统

Nginx 版本： nginx-1.12.2

MySQL 版本： mysql 5.7.27

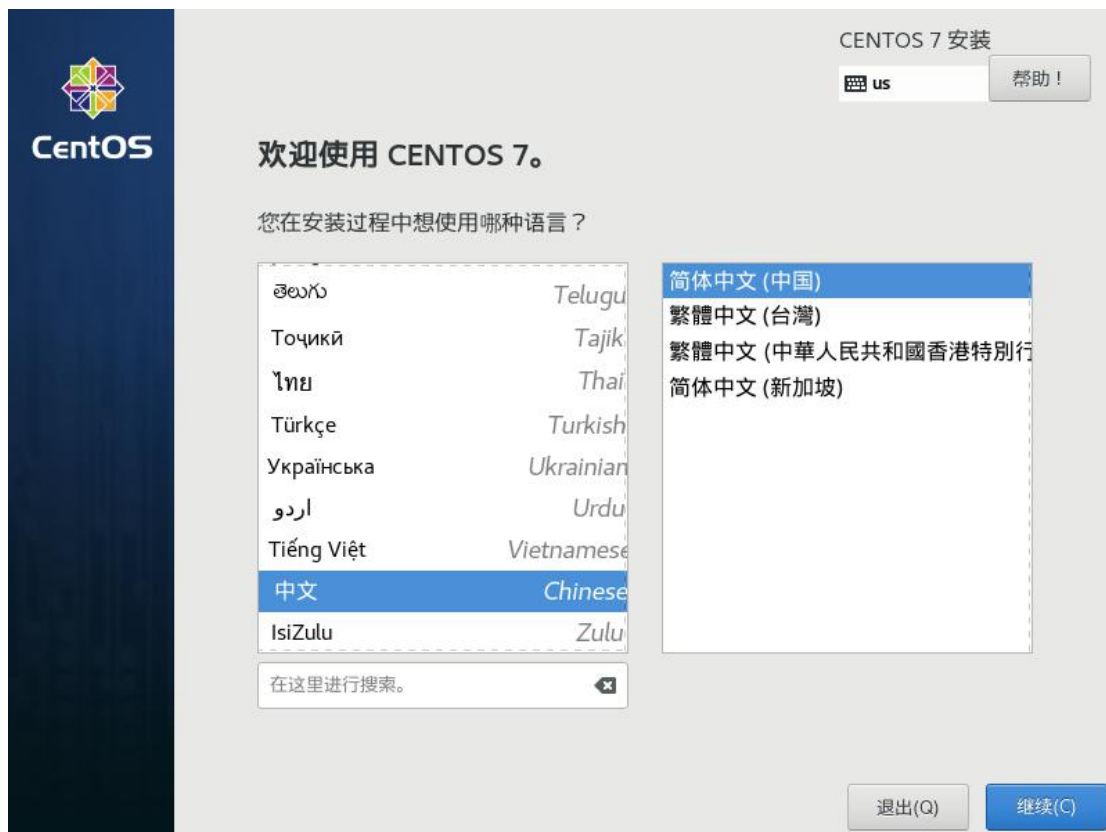
PHP 版本： php-7.2.19

0x01 安装 centos7 操作系统：

启动镜像，选择第一个选项，回车



选择，中文---简体中文



选择系统--安装位置，



选择，我要配置分区，点击完成



选择标准分区



然后单击 创建新的分区，分区提前规划好， /boot 分区 200M，一般 swap 分区为物理内存的 1.5-2 倍，当物理机内存多于 16G 后，swap 分区给 8-16G 都可以。 /根分区 10G，实际工作中可以创建数据分区，一般把数据和系统分开。

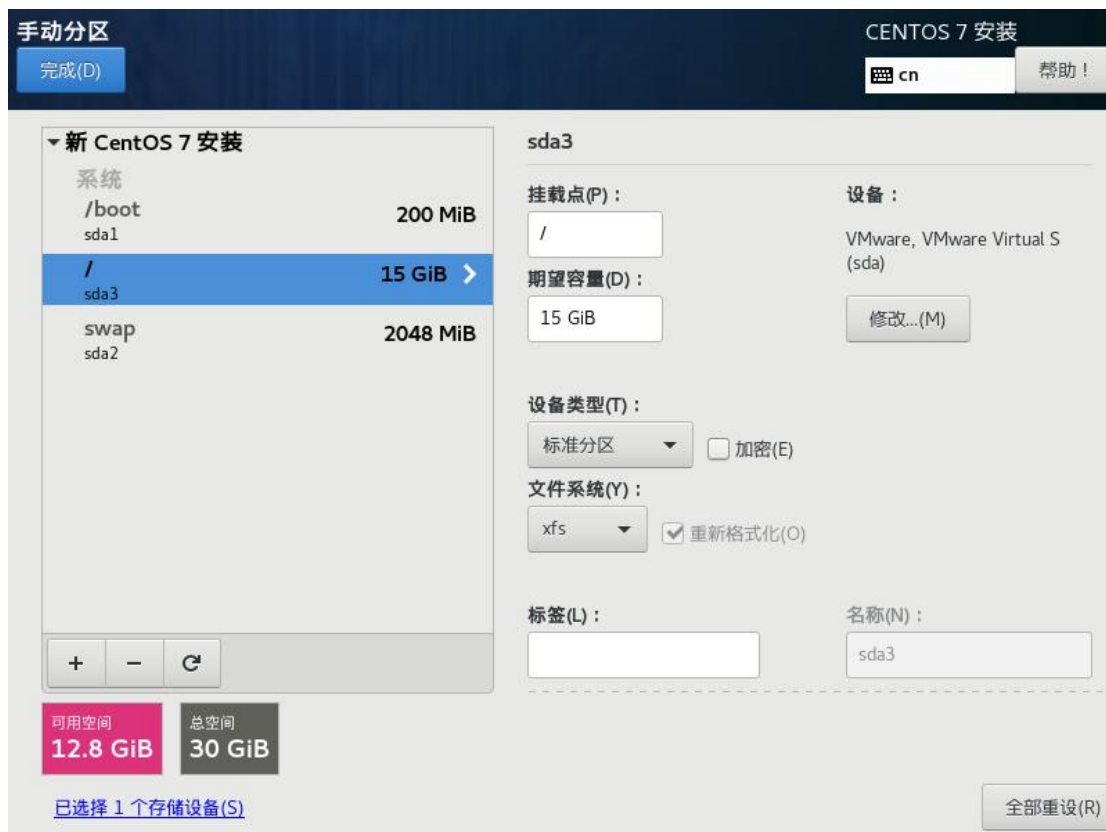
创建 boot 分区：



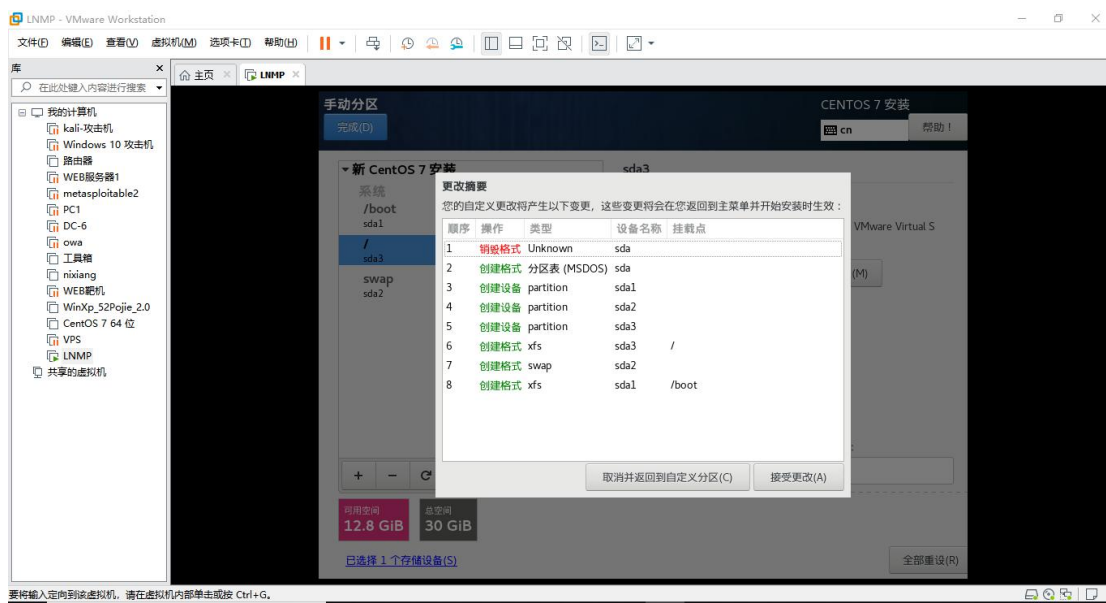
创建 swap 分区：



创建/分区：



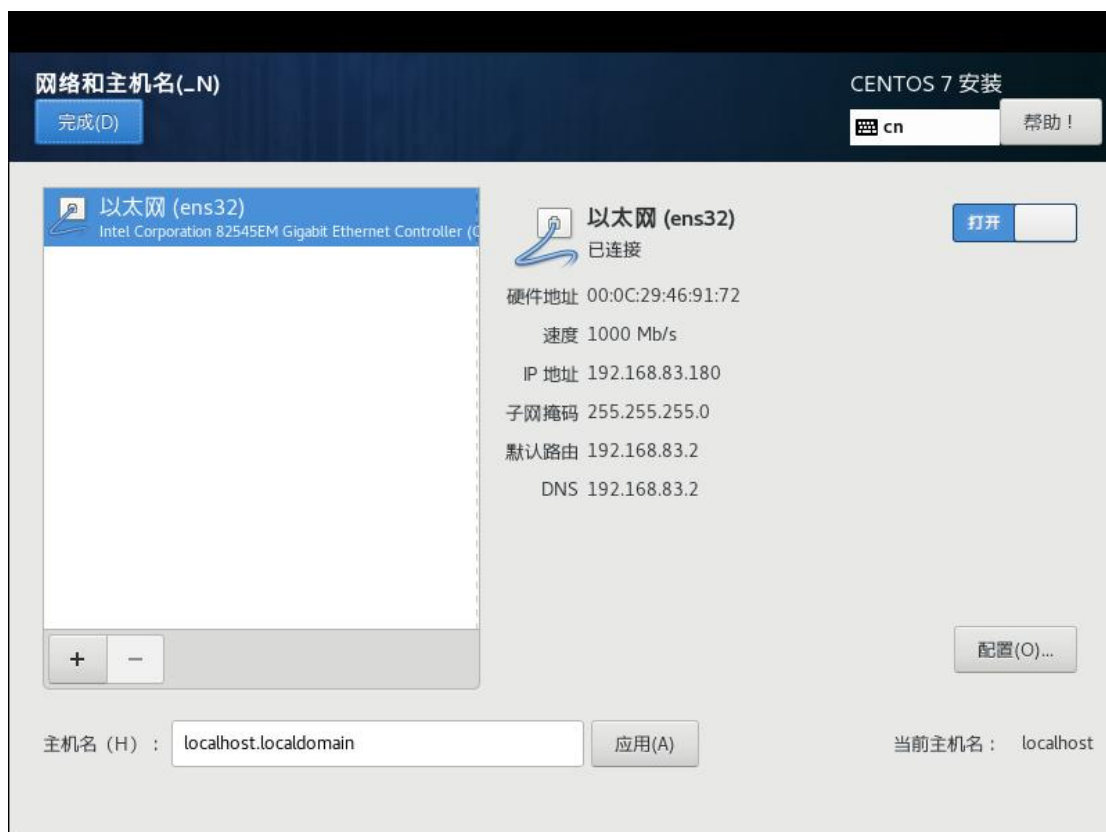
分区创建完成后，点击完成按钮，选择接受更改



关闭 KDUMP 功能



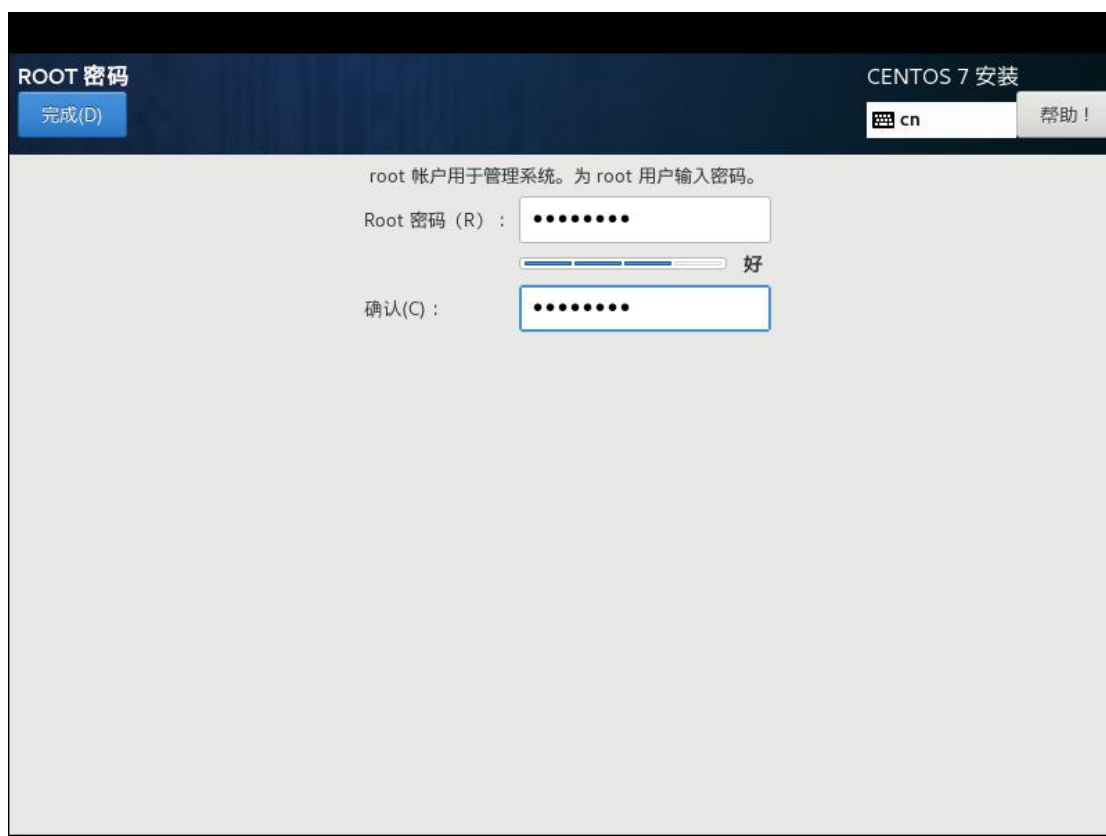
设置网络，选择打开，可以看到获得 IP 地址，修改主机名称，点完成。如果为静态 IP，需要在配置选项里手动配置 IP 地址，DNS 等信息



全部配置完成后，点击开始安装



创建 root 密码



安装完成后，点重启，进入到系统



- **优化配置**

查看是否安装 SSH

```
rpm -qa |grep  ssh
```

如果未安装 进行安装 `yum -y install openssh-server vim wget`

配置 SSH: `vim /etc/ssh/sshd_config` 去掉 port 22 前面的#号

- **解决中文乱码问题**

安装语言包:

```
yum install kde-l10n-Chinese yum reinstall glibc-common
```

修改为中文:

```
vi /etc/locale.conf
```

```
LANG="zh_CN.UTF-8"
```

```
source /etc/locale.conf
```

```
echo $LANG
```

- **更换阿里源**


```
cd /etc/yum.repos.d
mv CentOS-Base.repo CentOS-Base.repo.bak
Wget -O CentOS-Base.repo http://mirrors.aliyun.com/repo/Centos-7.repo
Wget -O/etc/yum.repos.d/epel.repo http://mirrors.aliyun.com/repo/epel-7.repo
yum clean all
yum makecache
更新所有源 yum update
```

- **关闭 SELinux**

如果您想永久关闭 SELinux，输入命令 `vi /etc/selinux/config` 编辑 SELinux 配置文件。回车后，把光标移动到 `SELINUX=enforcing` 这一行，按下 `i` 键进入编辑模式，修改为 `SELINUX=disabled`，按下 `Esc` 键，然后输入 `:wq` 并回车以保存并关闭 SELinux 配置文件。

- **关闭防火墙**

```
systemctl stop firewalld
systemctl disable firewalld
```

注：不关闭防火墙 web 界面打不开，解析不了

0x02 安装 nginx:

运行如下命令进行安装，

```
yum -y install nginx
```

```
[root@LNMP ~]# yum -y install nginx
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
Resolving Dependencies
--> Running transaction check
---> Package nginx.x86_64 1:1.12.2-3.el7 will be installed
--> Processing Dependency: nginx-all-modules = 1:1.12.2-3.el7 for package: 1:nginx-1.12.2-3.el7.x86_64
--> Processing Dependency: nginx-filesystem = 1:1.12.2-3.el7 for package: 1:nginx-1.12.2-3.el7.x86_64
--> Processing Dependency: nginx-filesystem for package: 1:nginx-1.12.2-3.el7.x86_64
--> Processing Dependency: libprofiler.so.0()(64bit) for package: 1:nginx-1.12.2-3.el7.x86_64
--> Running transaction check
---> Package gperftools-libs.x86_64 0:2.6.1-1.el7 will be installed
---> Package nginx-all-modules.noarch 1:1.12.2-3.el7 will be installed
--> Processing Dependency: nginx-mod-http-geoip = 1:1.12.2-3.el7 for package: 1:nginx-all-modules-1.12.2-3.el7.noarch
```

安装完成后，查看其版本 `nginx -v`，显示版本，说明安装成功

```
[root@LNMP ~]#
[root@LNMP ~]# nginx -v
nginx version: nginx/1.12.2
[root@LNMP ~]#
```

0x03 安装 Mysql 数据库：

将 mysql 更新到源中

Rpm `-Uvh` <http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm>

```
[root@LNMP ~]# rpm -Uvh http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
Retrieving http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
warning: /var/tmp/rpm-tmp.XxkjzW: Header V3 DSA/SHA1 Signature, key ID 5072e1f5: NOKEY
Preparing... ##### [100%]
Updating / installing...
 1:mysql57-community-release-el7-9 ##### [100%]
[root@LNMP ~]#
```

使用 yum 安装 mysql 数据库，时间稍长

`yum -y install mysql-community-server`

```
[root@LNMP ~]# yum -y install mysql-community-server
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
mysql-connectors-community | 2.5 kB 00:00:00
mysql-tools-community | 2.5 kB 00:00:00
mysql57-community | 2.5 kB 00:00:00
(1/3): mysql-connectors-community/x86_64/primary_db | 44 kB 00:00:00
(2/3): mysql-tools-community/x86_64/primary_db | 61 kB 00:00:00
(3/3): mysql57-community/x86_64/primary_db | 184 kB 00:00:01
Resolving Dependencies
--> Running transaction check
---> Package mysql-community-server.x86_64 0:5.7.27-1.el7 will be installed
--> Processing Dependency: mysql-community-common(x86-64) = 5.7.27-1.el7 for package: mysql-community-server-5.7.27-1.el7.x86_64
```

查看版本，`mysql -V`，显示版本安装成功

```
[root@LNMP ~]# mysql -V
mysql Ver 14.14 Distrib 5.7.27, for Linux (x86_64) using EditLine wrapper
[root@LNMP ~]#
```

0x04 安装 PHP:

更新 YUM 源

Yum install -y

http://dl.iuscommunity.org/pub/ius/stable/CentOS/7/x86_64/ius-release-1.0-15.ius.centos7.noarch.rpm

```
[root@LNMP ~]# yum install -y http://dl.iuscommunity.org/pub/ius/stable/CentOS/7/x86_64/ius-release-1.0-15.ius.centos7.noarch.rpm
Loaded plugins: fastestmirror
ius-release-1.0-15.ius.centos7.noarch.rpm | 8.1 kB 00:00:00
Examining /var/tmp/yum-root-OQPf0I/ius-release-1.0-15.ius.centos7.noarch.rpm: ius-release-1.0-15.ius.centos7.noarch
Marking /var/tmp/yum-root-OQPf0I/ius-release-1.0-15.ius.centos7.noarch.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package ius-release.noarch 0:1.0-15.ius.centos7 will be installed
--> Processing Dependency: epel-release = 7 for package: ius-release-1.0-15.ius.centos7.noarch
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
--> Running transaction check
--> Package epel-release.noarch 0:7-11 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
ius-release noarch 1.0-15.ius.centos7 /ius-release-1.0-15.ius.centos7.noarch 8.5 k
```

rpm -Uvh <https://mirror.webtatic.com/yum/el7/webtatic-release.rpm>

```
[root@LNMP ~]# rpm -Uvh https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
Retrieving https://mirror.webtatic.com/yum/el7/webtatic-release.rpm
warning: /var/tmp/rpm-tmp.hnnQie: Header V4 RSA/SHA1 Signature, key ID 62e74ca5: NOKEY
Preparing...
Updating / installing...
1:webtatic-release-7-3 [100%]
[root@LNMP ~]#
```

使用 yum 安装 PHP ,

yum -y install php72w-devel php72w.x86_64 php72w-cli.x86_64 php72w-common.x86_64 php72w-gd.x86_64 php72w-ldap.x86_64 php72w-mbstring.x86_64 php72w-mcrypt.x86_64 php72w-pdo.x86_64 php72w-mysqlnd php72w-fpm php72w-opcache php72w-pecl-redis php72w-pecl-mongo

```
[root@LNMP ~]# yum -y install php72w-devel php72w.x86_64 php72w-cli.x86_64 php72w-common.x86_64
php72w-gd.x86_64 php72w-ldap.x86_64 php72w-mbstring.x86_64 php72w-mcrypt.x86_64 php72w-pdo.x86_64
php72w-mysqlnd php72w-fpm php72w-opcache php72w-pecl-redis php72w-pecl-mongo
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * ius: hkg.mirror.rackspace.com
 * updates: mirrors.aliyun.com
 * webtatic: us-east.repo.webtatic.com
No package php72w.x86_64 available.
No package php72w-mcrypt.x86_64 available.
No package php72w-pecl-mongo available.
Resolving Dependencies
--> Running transaction check
--> Package php72w-cli.x86_64 0:7.2.19-1.w7 will be installed
--> Processing Dependency: libargon2.so.0()(64bit) for package: php72w-cli-7.2.19-1.w7.x86_64
--> Package php72w-common.x86_64 0:7.2.19-1.w7 will be installed
--> Package php72w-devel.x86_64 0:7.2.19-1.w7 will be installed
--> Processing Dependency: pcre-devel(x86-64) for package: php72w-devel-7.2.19-1.w7.x86_64
--> Processing Dependency: automake for package: php72w-devel-7.2.19-1.w7.x86_64
--> Processing Dependency: autoconf for package: php72w-devel-7.2.19-1.w7.x86_64
--> Package php72w-fpm.x86_64 0:7.2.19-1.w7 will be installed
--> Package php72w-gd.x86_64 0:7.2.19-1.w7 will be installed
--> Package php72w-ldap.x86_64 0:7.2.19-1.w7 will be installed
--> Package php72w-mbstring.x86_64 0:7.2.19-1.w7 will be installed
```

查看版本 `php -v` 显示版本号为安装成功

```
[root@LNMP ~]# php -v
PHP 7.2.19 (cli) (built: Jun 2 2019 09:49:05) ( NTS )
Copyright (c) 1997-2018 The PHP Group
Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies
    with Zend OPcache v7.2.19, Copyright (c) 1999-2018, by Zend Technologies
[root@LNMP ~]#
```

0x05 配置 Nginx

先将 nginx 配置文件备份一下（更改 linux 系统的配置文件，都要养成备份的好习惯）

命令：`cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.bak`

使用 `vim` 命令打开 nginx 配置文件 **命令：**`vim /etc/nginx/nginx.conf`

编辑内容，放到 `server { 编辑的内容 }`

内容：

```
location / {
    index index.php index.html index.htm;
} #配置 Nginx 通过 fastcgi 方式处理您的 PHP 请求

location ~ .php$ {
    root /usr/share/nginx/html;
    fastcgi_pass 127.0.0.1:9000; #Nginx 通过本机的 9000 端口将 PHP 请求转发给 PHP-FPM 进行处理。
    fastcgi_index index.php;
    include fastcgi_params; #Nginx 调用 fastcgi 接口处理 PHP 请求
```



```

fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
}

```

```

server {
    listen      80;
    root        /usr/share/nginx/html;
    location / {
        index   index.php index.html index.htm;
    }
    location ~.php$ {
        root    /usr/share/php/html;
        fastcgi_pass 127.0.0.1:9000;
        fastcgi_index index.php;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}

```

启动 nginx

systemctl start nginx

```

[root@LNMP nginx]# systemctl start nginx
[root@LNMP nginx]#
[root@LNMP nginx]#
[root@LNMP nginx]# systemctl status nginx
â nginx.service - The nginx HTTP and reverse proxy server
Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; vendor preset: disabled)
Active: active (running) since Tue 2019-07-30 17:36:26 CST; 36s ago
Process: 34667 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
Process: 34663 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
Process: 34661 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
Main PID: 34669 (nginx)
CGroup: /system.slice/nginx.service
        付34669 nginx: master process /usr/sbin/nginx
        付34670 nginx: worker process
        付34671 nginx: worker process

Jul 30 17:36:26 LNMP systemd[1]: Starting The nginx HTTP and reverse proxy server...
Jul 30 17:36:26 LNMP nginx[34663]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jul 30 17:36:26 LNMP nginx[34663]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jul 30 17:36:26 LNMP systemd[1]: Started The nginx HTTP and reverse proxy server.
Hint: Some lines were ellipsized, use -l to show in full.

```

设置开机自动启动 nginx

systemctl enable nginx

```

[root@LNMP nginx]# systemctl enable nginx
Created symlink from /etc/systemd/system/multi-user.target.wants/nginx.service to /usr/lib/systemd/system/nginx.service.

```

用浏览器访问 IP 地址 显示以下内容,nginx 正常

Welcome to **nginx** on Fedora!

This page is used to test the proper operation of the **nginx** HTTP server after it has been installed. If you can read this page, it means that the web server installed at this site is working properly.

Website Administrator

This is the default `index.html` page that is distributed with **nginx** on Fedora. It is located in `/usr/share/nginx/html`.

You should now put your content in a location of your choice and edit the `root` configuration directive in the **nginx** configuration file `/etc/nginx/nginx.conf`.

0x06 配置 MySQL 数据库

启动数据库服务

命令: `systemctl start mysqld`

设置开机自动启动

命令: `systemctl enable mysqld`

查看安装时 mysql 日志文件, 找到初始数据库密码

命令: `grep 'temporary password' /var/log/mysqld.log`

红色标注为初始数据库密码

```
[root@LNMP nginx]#  
[root@LNMP nginx]# grep 'temporary password' /var/log/mysqld.log  
2019-07-30T09:41:08.863167Z 1 [Note] A temporary password is generated for root@localhost: wN~.sg0c?6t~  
[root@LNMP nginx]#  
[root@LNMP nginx]#
```

配置 Mysql 安全性

命令: `mysql_secure_installation`

重置密码

```
[root@LNMP nginx]#  
[root@LNMP nginx]# mysql_secure_installation  
  
Securing the MySQL server deployment.  
  
Enter password for user root: 输入上述标红初始数据库密码  
  
The existing password for the user account root has expired. Please set a new password.
```

```
New password: 输入新的数据库密码, 必须大小写、数字、特殊字符, 缺一不可  
  
Re-enter new password: 重复输入密码  
The 'validate_password' plugin is installed on the server.  
The subsequent steps will run with the existing configuration  
of the plugin.  
Using existing password for root.  
  
Estimated strength of the password: 100  
Change the password for root ? ((Press y|Y for Yes, any other key for No) : n  
  
... skipping. 是否更改root密码, 这里选n  
By default, a MySQL installation has an anonymous user,  
allowing anyone to log into MySQL without having to have  
a user account created for them. This is intended only for  
testing, and to make the installation go a bit smoother.  
You should remove them before moving into a production  
environment.
```

删除匿名用户帐号

```
Remove anonymous users? (Press y|Y for Yes, any other key for No) : y  
Success. 删除匿名用户, 选y  
  
Normally, root should only be allowed to connect from  
'localhost'. This ensures that someone cannot guess at  
the root password from the network.
```

禁止 root 帐号远程登陆

```
Disallow root login remotely? (Press y|Y for Yes, any other key for No) : y
Success.
```

禁止root用户远程登陆, 这里选y

删除 test 数据库

```
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.
```

```
Remove test database and access to it? (Press y|Y for Yes, any other key for No) : y
- Dropping test database...
Success.
```

删除test数据库, 这里选y

```
- Removing privileges on test database...
Success.
```

加载授权表

```
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.
```

```
Reload privilege tables now? (Press y|Y for Yes, any other key for No) : y
Success.
```

重新加载授权表, 这里选y

All done!

```
[root@LNMP nginx]#
```

0x07 配置 PHP

启动 php-fpm

命令: systemctl start php-fpm

设置开机自动启动

命令: systemctl enable php-fpm

```
[root@LNMP nginx]#
[root@LNMP nginx]# systemctl enable php-fpm
Created symlink from /etc/systemd/system/multi-user.target.wants/php-fpm.service to /usr/lib/systemd/system/php-fpm.service.
[root@LNMP nginx]#
```

输入 netstat -tunlp , 显示以下端口开启, 说明配置成功

```
[root@LNMP nginx]# netstat -tunlp
Active Internet connections (only servers)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program name
tcp	0	0	127.0.0.1:9000	0.0.0.0:*	LISTEN	45194/php-fpm: mast
	0	0	0.0.0.0:80	0.0.0.0:*	LISTEN	34669/nginx: master
	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	6511/sshd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	6648/master
	0	0	:::3306	:::*	LISTEN	34800/mysqld
	0	0	:::22	:::*	LISTEN	6511/sshd
tcp6	0	0	:::1:25	:::*	LISTEN	6648/master
	0	0	0.0.0.0:68	0.0.0.0:*		7373/dhclient
	0	0	127.0.0.1:323	0.0.0.0:*		5890/chronyd
udp6	0	0	:::1:323	:::*		5890/chronyd


0x08 测试

● nginx 解析 php

将 a.php 脚本放到 /usr/share/nginx/html/ 目录下，在 nginx 配置里设置的路径下

a.php 脚本内容：<?php echo phpinfo(); ?>

在浏览器上输入 IP 地址/a.php，显示出 phpinfo 信息，说明解析正常

PHP Version 7.2.19	
	
System	Linux LNMP 3.10.0-957.el7.x86_64 #1 SMP Thu Nov 8 23:39:32 UTC 2018 x86_64
Build Date	Jun 2 2019 09:52:22
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/bz2.ini, /etc/php.d/calendar.ini, /etc/php.d/ctype.ini, /etc/php.d/curlini, /etc/php.d/dom.ini, /etc/php.d/exif.ini, /etc/php.d/fileinfo.ini, /etc/php.d/ftp.ini, /etc/php.d/gd.ini, /etc/php.d/gettext.ini, /etc/php.d/gmp.ini, /etc/php.d/conv.ini, /etc/php.d/gbinary.ini, /etc/php.d/json.ini, /etc/php.d/ldap.ini, /etc/php.d/mbstring.ini, /etc/php.d/mysqli.ini, /etc/php.d/mysqli_mysql.ini, /etc/php.d/oci8.ini, /etc/php.d/odbc.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_mysql_sqlite3.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/posix.ini, /etc/php.d/redis.ini, /etc/php.d/shmop.ini, /etc/php.d/simplexml.ini, /etc/php.d/sockets.ini, /etc/php.d/sqlite3.ini, /etc/php.d/sysvmsg.ini, /etc/php.d/sysvsem.ini, /etc/php.d/sysvshm.ini, /etc/php.d/tokenizer.ini, /etc/php.d/xml.ini, /etc/php.d/xml_wddx.ini, /etc/php.d/xmlreader.ini, /etc/php.d/xmlwriter.ini, /etc/php.d/xsl.ini, /etc/php.d/zip.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718

● php 连接 mysql

将 b.php 脚本放到/usr/share/nginx/html/ 目录下，

b.php 脚本内容：

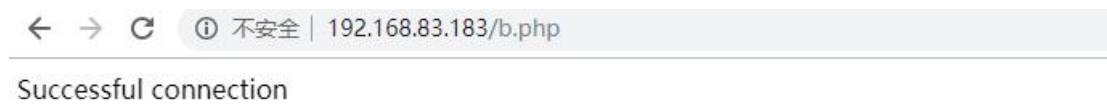
```
<?php
header("Content-type:text/html;charset=utf-8");
// 创建连接
$conn=mysqli_connect('localhost','root','lqaz@WSX');//三个参数分别对应服务器名，账号，密码
// 检测连接
if (!$conn) {
    die("连接服务器失败：" . mysqli_connect_error()); //连接服务器失败退出程序
}
else {
```



```
        echo "Successful connection";
    }

    mysql_close($conn); //关闭连接
?>
```

在浏览器中输入 IP 地址/b.php，数据库连接成功显示：Successful connection



0x09 nginx 加固

- **nginx.conf 配置文件**

- **日志格式配置**

(http 标签内，在所有的 server 标签内可以调用)： `log_format main '$remote_addr - $remote_user [$time_local] "$request" ' '$status $body_bytes_sent "$http_referer" ' '"$http_user_agent" "$http_x_forwarded_for"' ;`

- **限制 HTTP 请求方法**

```
if ($request_method !~ ^(GET|HEAD|POST)$ ) { return 444; }
if ($http_range ~ "\d{9,}") { return 444; }
```

- **设置超时时间**

`client_body_timeout 10; # 设置客户端请求主体读取超时时间`
`client_header_timeout 10; #设置客户端请求头读取超时时间` `keepalive_timeout 5 5; #第一个参数指定客户端连接保持活动的超时时间，第二个参数是可选的，它指定了消息头保持活动的有效时间` `send_timeout10; #指定响应客户端的超时时间`

- **屏蔽 IP 地址（不全面）**

```
if ( $geoip_country_code !~ ^(CN|US)$ ) { return 403; }
```

- **封掉不正常的 user-agent (不全面)**

```
if ( $http_user_agent ~*  
"java|python|perl|ruby|curl|bash|echo|uname|base64|decode|md5sum|select|con  
cat|httprequest|httpclient|nmap|scan" ) { return 403; } if  
($http_user_agent ~* "" ) { return 403; }
```

- **强制使用域名访问**

```
if ( $host !~* 'XXXX.com' ) { return 403; }
```

- **url 参数过滤敏感字 (可以自行添加)**

```
if ( $query_string ~* "union.*select.*\" ) { rewrite ^/(.*)$ $host  
permanent; } if ( $query_string ~* "concat.*\" ) { rewrite ^/(.*)$ $host  
permanent; }
```

- **强制要求 referer 访问**

```
if ( $http_referer = "" ) { return 403; }
```

- **http {} 模块**

- **禁止目录浏览**

```
autoindex off;
```

- **错误页面重定向**

```
http { ... fastcgi_intercept_errors on; error_page 401 /401.html;  
error_page 402 /402.html; error_page 403 /403.html; error_page 404  
/404.html; error_page 405 /405.html; error_page 500 /500.html; ... } 修改内  
容: ErrorDocument 400 /custom400.html ErrorDocument 401 /custom401.html  
ErrorDocument 403 /custom403.html ErrorDocument 404 /custom404.html  
ErrorDocument 405 /custom405.html ErrorDocument 500 /custom500.html 其中
```

401.html、402.html、403.html、404.html、405.html、500.html 为要指定的错误提示页面。

- **隐藏版本信息**

server_tokens off;

- **定义日志路径**

access_log /var/log/nginx/access.log main;

```
[root@LNMP nginx]#  
[root@LNMP nginx]# ls /var/log/nginx/  
access.log  error.log  
[root@LNMP nginx]#
```

- **server {} 模块**

- **限制 IP 访问**

location / { deny 192.168.1.1; #拒绝 IP allow 192.168.1.0/24; #允许 IP allow 10.1.1.0/16; #允许 IP deny all; #拒绝其他所有 IP }

- **限制并发和速度**

```
limit_zone one $binary_remote_addr 10m; server { listen 80; server_name  
XXX.XXXX.com; index index.html index.htm index.php; root  
/usr/share/nginx/html; #Zone limit; location / { limit_conn one 1;  
limit_rate 20k; } }
```

个人简介:

个人在知识星球中的 ID 个人在知识星球中的 ID: 880

常用名: ven0m

联系方式: QQ 1005941254

目前职业: 在职

所在地区： 北京

熟悉的编程语言： `php/python` 会，不太熟练

自我介绍：

小白一枚，自学安全，所有东西都是会一点，但都不精。希望大佬多带带，急需一良师。