

第7章

综合环境的搭建与应用

传统的服务软件都是独立的，而要实现更复杂的功能，则需要各种服务联动，来创建一个适用于多种网络应用工作的综合环境。Docker容器的部署打破了传统服务器的概念，通过该容器，可以快速完成多个系统、多种服务的创建，即装即用、高效稳定是其最大特点。本章将向读者介绍综合环境和Docker容器的部署和使用。

重点难点

- LNMP的部署
- Docker的部署

7.1 LNMP的部署

LNMP是一种常用的Web服务器架构，具有高性能、高可扩展性的解决方案，广泛应用于动态网站和Web应用程序。

7.1.1 认识LNMP

LNMP是指一组通常一起使用来运行动态网站或者服务器的自由软件名称首字母缩写。L指Linux，N指Nginx，M一般指MySQL，也可以指MariaDB，P一般指PHP，也可以指Perl或Python。通常来说，LNMP代表的就是Linux系统下Nginx+MySQL+PHP这种网站服务器架构。

知识拓展

PHP

PHP (Hypertext Preprocessor) 即“超文本预处理器”，是在服务器端执行的脚本语言，尤其适用于Web开发，并可嵌入HTML中。PHP语法学习了C、Java和Perl多种语言的特色，并根据它们的长项进行改进、提升。PHP同时支持面向对象和面向过程的开发，使用非常灵活。

LNMP的工作步骤如下。

步骤01 用户在浏览器中输入域名或者IP地址访问网站。

步骤02 用户在访问网站时，向Web服务器发出http request请求，服务器响应并处理Web请求，返回静态网页资源，如CSS、picture、video等，然后缓存在用户主机上。

步骤03 服务器调用动态资源，PHP脚本调用fastCGI传输给php-fpm，然后php-fpm调用PHP解释器进程解析PHP脚本。

步骤04 如果出现大流量高并发的情况，PHP解析器也可以开启多进程处理高并发，将解析后的脚本返回给php-fpm，php-fpm再调用fast-cgi，将脚本解析信息传送给Nginx，服务器再通过http response传送给用户浏览器。

步骤05 浏览器再将服务器传送的信息进行解析与渲染，呈现给用户。

7.1.2 LNMP一键部署工具

在第6章已经介绍了Nginx和MySQL的部署，而要加入PHP，就需要手动安装PHP，并且要完成软件之间复杂的联动配置。这种环境的搭建需要输入大量命令，如果是配置生产环境，则需要耗费大量的时间，而且容易产生各种问题。于是开发者开发了LNMP的一键部署工具，该工具使用户无须一个一个地安装组件，输入命令即可一次性安装。在编译安装时，可以优化编译参数、提高性能，解决软件间依赖，特别对配置能自动优化。其主要特点如下。

- 支持自定义Nginx、PHP编译参数及网站和数据库目录。
- 支持生成免费的SSL证书。
- LNMP模式支持多PHP版本。
- 支持单独安装各组件。
- 提供一些实用的辅助工具的一键安装。
- 支持重置MySQL root密码、日志切割、SSH防护、备份等许多实用脚本。

LNMP可以在线部署与离线部署，建议在线部署。部署采用脚本安装方式，在线下载即可。而且安装过程采用会话模式，用户通过键盘输入数字进行选择。安装需要切换为root用户，具体部署步骤如下。

```
[wlysy@localhost ~]$ sudo su - root //切换到root用户
[sudo] wlysy 的密码:
[root@localhost ~]# wget https://soft.lnmp.com/lnmp/lnmp2.1.tar.gz -O
lnmp2.1.tar.gz && tar zxf lnmp2.1.tar.gz && cd lnmp2.1 && ./install.sh lnmp
//使用命令下载并解压解包，进入该目录后，启动安装脚本文件进行安装
+-----+
+
|           LNMP V2.1 for CentOS Linux Server, Written by Licess
|
+-----+
+
|           A tool to auto-compile & install LNMP/LNMPA/LAMP on Linux
|
+-----+
+
|           For more information please visit https://lnmp.org
|
+-----+
+
You have 11 options for your DataBase install.
1: Install MySQL 5.1.73
2: Install MySQL 5.5.62 (Default)
3: Install MySQL 5.6.51
4: Install MySQL 5.7.44
5: Install MySQL 8.0.37
6: Install MariaDB 5.5.68
7: Install MariaDB 10.4.33
8: Install MariaDB 10.5.24
9: Install MariaDB 10.6.17
10: Install MariaDB 10.11.7
11: Install MySQL 8.4.0
0: DO NOT Install MySQL/MariaDB
Enter your choice (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11 or 0): 5 //选择安装版本
Using Generic Binaries [y/n]: y //是否以二进制方式安装
You will install MySQL 8.0.37 Using Generic Binaries.
=====
```

```
Please setup root password of MySQL.
Please enter: 123456 //设置数据库密码
=====
Do you want to enable or disable the InnoDB Storage Engine?
Default enable,Enter your choice [Y/n]: y //是否启用InnoDB引擎
```

接下来选择PHP版本，用户根据需要选择即可。

```
You have 9 options for your PHP install.
1: Install PHP 5.2.17
2: Install PHP 5.3.29
3: Install PHP 5.4.45
4: Install PHP 5.5.38
5: Install PHP 5.6.40 (Default)
6: Install PHP 7.0.33
7: Install PHP 7.1.33
8: Install PHP 7.2.34
9: Install PHP 7.3.33
10: Install PHP 7.4.33
11: Install PHP 8.0.30
12: Install PHP 8.1.28
13: Install PHP 8.2.19
14: Install PHP 8.3.7
Enter your choice (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14): 14
```

注意事项 PHP版本选择

选择PHP 7+版本时需要确认PHP版本是否与自己的程序兼容。

选择要安装的内存分配器，主要用于内存管理，用户可根据需要选择，默认为不安装。这里直接按回车键即可。

```
You have 3 options for your Memory Allocator install.
1: Don't install Memory Allocator. (Default)
2: Install Jemalloc
3: Install TCMalloc
Enter your choice (1, 2 or 3):
No input,You will not install Memory Allocator.
```

系统提示按任意键启动安装，按Ctrl+C组合键取消安装。按任意键启动安装即可。

```
Press any key to install...or Press Ctrl+c to cancel //按任意键启动安装
```

LNMP脚本就会自动安装Nginx、MySQL、PHP、phpMyAdmin等软件及相关组件。安装完成后，会显示完成界面。

```
Install lnmp takes 12 minutes. //安装的总时间
Install lnmp V2.1 completed! enjoy it. //安装完成
```

动手练 检测运行环境

安装完毕，用户可以在局域网环境中通过IP地址访问服务器，如果部署成功，则会显示LNMP的欢迎界面，如图7-1所示。在该界面下方单击“探针”链接，如图7-2所示。



图 7-1

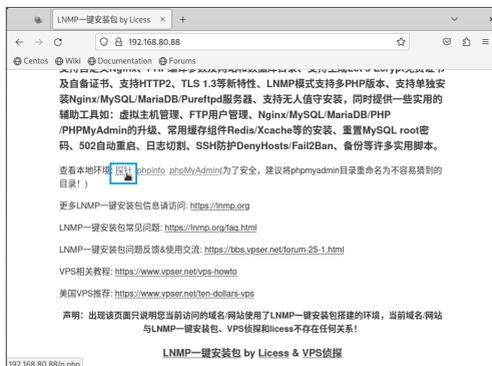


图 7-2

知识拓展

其他链接

在界面下方，还有其他的一些LNMP一键安装包信息、常见问题、反馈、交流、教程等链接信息，用户可以根据需要访问学习。

软件会通过PHP脚本的网页显示当前服务器的软硬件参数、版本、连接测试等，如图7-3所示。如果在图7-2中单击phpinfo链接，则会弹出PHP的相关信息、参数等，如图7-4所示。



图 7-3

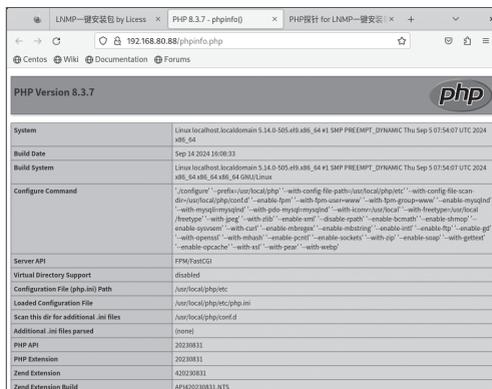


图 7-4

如果单击phpMyAdmin链接，则会弹出phpMyAdmin数据库网页端管理工具，如图7-5所示，即可登录和管理数据库，如图7-6所示。

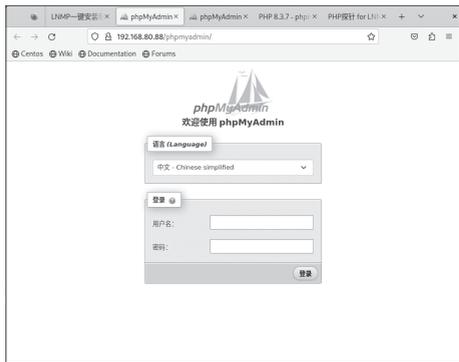


图 7-5

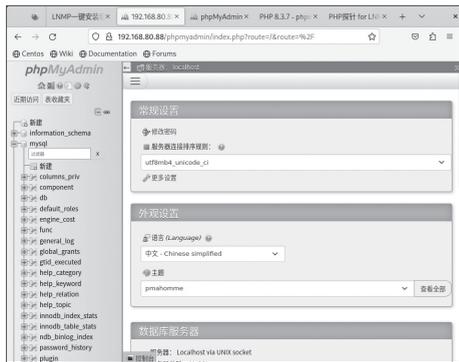


图 7-6

7.1.3 虚拟主机

默认情况下，LNMP的主目录在“/home/wwwroot/default”中，用户可以切换到root用户进行查看。为了方便创建与管理，以下均需要切换为root用户后操作。

```
[wlysy@localhost ~]$ sudo su - root
[sudo] wlysy 的密码:
[root@localhost ~]# cd /home/wwwroot/default/
[root@localhost default]# ll
总用量 84
-rw-r--r--. 1 root root 3196  9月 14 16:12 index.html      //网页主文件
-rw-r--r--. 1 root root 5683  9月 14 16:12 lnmp.gif
-rw-r--r--. 1 root root 20256 9月 14 16:12 ocp.php        //opcache控制面板
-rw-r--r--. 1 root root 20    9月 14 16:12 phpinfo.php     //信息页面
drwxr-xr-x. 15 www  www  4096  9月 14 17:10 phpmyadmin //PHP数据库管理工具
-rw-r--r--. 1 root root 42621 9月 14 16:12 p.php          //PHP探针
```

1. 创建虚拟主机

LNMP安装之后，可以直接使用该工具提供的命令来快速部署虚拟机主机。无须进行复杂的设置，使用lnmp vhost add命令启动该工具的创建向导，就可以快速完成虚拟主机的添加。

```
[wlysy@localhost ~]$ sudo su - root
[sudo] wlysy 的密码:
[root@localhost ~]# lnmp vhost add
+-----+
|   Manager for LNMP, Written by Licess   |
+-----+
|                https://lnmp.org        |
+-----+
Please enter domain(example: www.lnmp.org): www.test.com      //绑定域名
Your domain: www.test.com
Enter more domain name(example: lnmp.org sub.lnmp.org): test.com //绑定其他域名
domain list: www.test.com test.com
Please enter the directory for the domain: www.test.com      //默认同名目录
```

```

Default directory: /home/wwwroot/www.test.com: //直接按回车键或手动更改
Virtual Host Directory: /home/wwwroot/www.test.com //提示已设置
Allow Rewrite rule? (y/n) y //伪静态可以使链接更加简洁,也利于SEO
Please enter the rewrite of programme,
wordpress, discuzx, typecho, thinkphp, laravel, codeigniter, yii2, zblog rewrite
was exist.
(Default rewrite: other): wordpress //让wordpress使用该规则
You choose rewrite: wordpress
Enable PHP Pathinfo? (y/n) n //是否启用pathinfo,一般不需开启
Disable pathinfo.
Allow access log? (y/n) y //是否启动日志记录
Enter access log filename(Default:www.test.com.log): //设置日志名称,默认同名
You access log filename: www.test.com.log
Enable IPv6? (y/n) n //是否启用IPv6
Disabled IPv6 Support in current Virtualhost.
Create database and MySQL user with same name (y/n) n
//创建同名数据库和用户等数据库相关参数,后续统一配置,这里不做处理
Add SSL Certificate (y/n) n //是否启用https,暂不开启
Press any key to start create virtul host... //按任意键自动创建
dCreate Virtul Host directory.....
set permissions of Virtual Host directory.....
You select the exist rewrite rule:/usr/local/nginx/conf/rewrite/
wordpress.conf
Test Nginx configure file.....
nginx: the configuration file /usr/local/nginx/conf/nginx.conf syntax is ok
nginx: configuration file /usr/local/nginx/conf/nginx.conf test is
successful
Reload Nginx.....
Reload service php-fpm done
=====
Virtualhost infomation: //虚拟主机详细信息
Your domain: www.test.com
Home Directory: /home/wwwroot/www.test.com
Rewrite: wordpress
Enable log: yes
Create database: no
Create ftp account: no
IPv6 Support: Disabled
=====

```

默认情况下创建的虚拟目录是空的,用户需要手动在其中创建主页文件并写入内容:

```

[root@localhost ~]# cd /home/wwwroot/www.test.com/
[root@localhost www.test.com]# ls
[root@localhost www.test.com]# vim index.html

```

2. 访问测试

写入正确的内容后,在客户机中就可以访问,访问效果如图7-7所示。

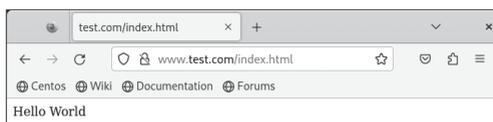


图 7-7

域名的解析

如果配置了DNS服务器，指向本虚拟主机，客户机又配置了DNS服务器地址，就可以正常地用域名访问虚拟主机。如果没有配置DNS服务器，客户机又需要进行域名解析，则可以修改系统中的hosts文件，添加IP地址和对应域名的映射关系，如图7-8所示。

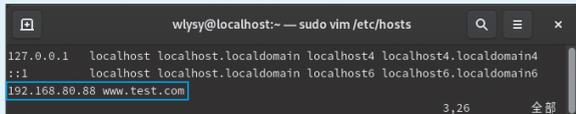


图 7-8

hosts文件中的解析会在DNS查询前生效。也就是说客户机会先查看hosts文件，如果有解析关系则直接使用，否则再查询DNS。如果没有DNS服务器，则可以修改该文件进行解析。在Linux中，该文件一般位于“/etc/hosts”目录中。

3. 删除虚拟主机

如果虚拟主机不再使用，可以删除该虚拟主机。

```
[root@localhost ~]# lnmp vhost del
+-----+
| Manager for LNMP, Written by Licess |
+-----+
| https://lnmp.org |
+-----+
=====
Current Virtualhost:
Nginx Virtualhost list:
www.test.com //会列出所有虚拟主机
=====
Please enter domain you want to delete: www.test.com //输入要删除的虚拟主机
=====
Domain: www.test.com has been deleted. //提示该虚拟主机已删除
Website files will not be deleted for security reasons.
You need to manually delete the website files. //提示目录未删除，要手动删除
=====
```

动手练 删除默认目录



需要手动删除目录时，其中的隐藏文件需要删除其不可变的属性，这也是增加安全性的一种方式。

```
[root@localhost ~]# rm -rf /home/wwwroot/www.test.com/
rm: 无法删除 '/home/wwwroot/www.test.com/.user.ini': 不允许的操作 //无法操作
[root@localhost ~]# chattr -i /home/wwwroot/www.test.com/.user.ini //删除属性
[root@localhost ~]# rm -rf /home/wwwroot/www.test.com/
[root@localhost ~]# ls /home/wwwroot/ //上条命令可执行，执行后只有默认目录 default
```

chattr命令

该命令用来修改文件的扩展属性和特殊权限。用户可以使用“chattr -l 文件”命令查看文件的属性信息。该命令的格式法如下。

chattr [+ -=] [属性] 文件名

+: 添加; -: 取消; =: 指定。

具体的属性及说明如下。

a: 使文件仅能追加数据, 不允许修改或删除。

i: 文件不允许修改或删除。

s: 同步文件内容至硬盘, 常用于关键文件的保护。

u: 当文件被删除时, 将其内容保存在硬盘中, 直到属性被取消。

c: 自动压缩文件, 在读取或访问时解压缩。

7.1.4 LNMP部署工具的命令及配置

LNMP部署工具除了使用方便外, 还可以根据自身的情况进行设置。下面介绍LNMP部署工具的常用命令, 以及其核心组件的配置位置。

1. 常用命令

在使用LNMP部署工具时, 无须单独设置其各组件的配置, 通过命令可以随时控制这些组件和功能的开关, 还可以启动配置向导等。常用的命令及作用如下。

(1) LNMP状态管理: `lnmp {start|stop|reload|restart|kill|status}`。

(2) LNMP各组件的状态管理: `lnmp {nginx|mysql|mariadb|php-fpm|pureftpd} {start|stop|reload|restart|kill|status}`。

(3) 虚拟主机管理: `lnmp vhost {add|list|del}`。

(4) 数据库管理: `lnmp database {add|list|edit|del}`。

(5) FTP用户管理: `lnmp ftp {add|list|edit|del|show}`。

(6) 已存在虚拟主机添加SSL: `lnmp ssl add`。

(7) 通过DNS API方式生成证书并创建虚拟主机: `lnmp dns {cx|dp|ali|...}`。

(8) 只通过DNS API方式生成SSL证书: `lnmp onlyssl {cx|dp|ali|...}`。

2. LNMP 组建安装目录

LNMP组件的默认安装目录如下, 用户可以根据需要来查看和修改。

(1) Nginx 目录: `/usr/local/nginx/`。

(2) MySQL 目录: `/usr/local/mysql/`。

(3) MySQL数据库所在目录: `/usr/local/mysql/var/`。

(4) MariaDB 目录: `/usr/local/mariadb/`。

(5) MariaDB数据库所在目录: `/usr/local/mariadb/var/`。

(6) PHP目录: `/usr/local/php/`。

(7) 多PHP版本目录: `/usr/local/php5.5/`, 其他版本将版本号5.5换成相应版本号即可。

(8) PHPMyAdmin目录: 0.9版本为`/home/wwwroot/phpmyadmin/`, 1.0及以上版本为`/home/wwwroot/default/phpmyadmin/`。强烈建议将此目录重命名为不容易猜到的名字。`phpmyadmin`可从官网下载新版替换。

(9) 默认网站目录: 0.9版本为 `/home/wwwroot/`, 1.0及以上版本为 `/home/wwwroot/default/`。

(10) Nginx日志目录: `/home/wwwlogs/`。

(11) `/root/vhost.sh`添加的虚拟主机配置文件所在目录: `/usr/local/nginx/conf/vhost/`。

(12) PureFtpd 目录: `/usr/local/pureftpd/`。

(13) PureFtpd Web管理目录: 0.9版本为`/home/wwwroot/default/ftp/`, 1.0版本为`home/www root/default/ftp/`。

(14) Proftpd 目录: `/usr/local/proftpd/`。

(15) Redis 目录: `/usr/local/redis/`。

3. LNMP 相关配置文件目录

用户可以修改这些组件的配置文件来设置各种参数, 或实现更复杂的功能。

(1) Nginx主配置(默认虚拟主机)文件: `/usr/local/nginx/conf/nginx.conf`。

(2) 添加的虚拟主机配置文件: `/usr/local/nginx/conf/vhost/域名.conf`。

(3) MySQL配置文件: `/etc/my.cnf`。

(4) PHP配置文件: `/usr/local/php/etc/php.ini`。

(5) `php-fpm`配置文件: `/usr/local/php/etc/php-fpm.conf`。

(6) PureFtpd配置文件: `/usr/local/pureftpd/pure-ftp.conf`, 1.3及更高版本: `/usr/local/pureftpd/etc/pure-ftp.conf`。

(7) PureFtpd MySQL配置文件: `/usr/local/pureftpd/pureftpd-mysql.conf`。

(8) Proftpd配置文件: `/usr/local/proftpd/etc/proftpd.conf`, 1.2及以前版本为`/usr/local/proftpd/proftpd.conf`。

(9) Proftpd 用户配置文件: `/usr/local/proftpd/etc/vhost/用户名.conf`。

(10) Redis 配置文件: `/usr/local/redis/etc/redis.conf`。

7.1.5 在LNMP环境中搭建网站

这种类型的网站的搭建不需要手动操作, 只需要像安装程序一样进行简单安装即可。这类网站的运行均需要LNMP环境的支持。配置好LNMP环境后, 用户只需下载并上传安装程序到网站目录, 然后按照提示进行安装即可。

1. 安装FTP服务器

网站在安装时, 需要远程将安装文件上传到网站目录中, 一般会在网站中安装FTP

服务。而LNMP部署工具默认没有安装FTP服务，可以使用其自带的部署工具快速安装。

```
[root@localhost ~]# lnmp ftp show //查看FTP服务
+-----+
|   Manager for LNMP, Written by Licess   |
+-----+
|           https://lnmp.org             |
+-----+
Pureftpd was not installed! //提示没有安装
[root@localhost ~]# ls
anaconda-ks.cfg lnmp2.1 lnmp2.1.tar.gz lnmp-install.log
[root@localhost ~]# cd lnmp2.1/ //进入之前解压的lnmp目录
[root@localhost lnmp2.1]# ls
addons.sh include License README uninstall.sh
ChangeLog init.d lnmp.conf src upgrade1.x-2.1.sh
conf install.sh pureftpd.sh tools upgrade.sh
[root@localhost lnmp2.1]# ./pureftpd.sh //执行FTP安装脚本
+-----+
|   Pureftpd for LNMP, Written by Licess   |
+-----+
|This script is a tool to install pureftpd for LNMP |
+-----+
|For more information please visit https://lnmp.org |
+-----+
|Usage: ./pureftpd.sh |
+-----+
Press any key to install...or Press Ctrl+c to cancel //按任意键启动安装
.....
Starting pureftpd...
Starting Pure-FTPd... done
+-----+
| Install Pure-FTPd completed,enjoy it! //安装完毕
| =>use command: lnmp ftp {add|list|del|show} to manage FTP users.
+-----+
| For more information please visit https://lnmp.org
+-----+
```

2. 配置FTP

安装了FTP服务后，就可以创建FTP目录和访问的账户名和密码了。

```
[root@localhost ~]# lnmp ftp add
+-----+
|   Manager for LNMP, Written by Licess   |
+-----+
|           https://lnmp.org             |
+-----+
Enter ftp account name: ftp //设置FTP账户名
Enter password for ftp account ftp: ftp //设置访问密码
Enter directory for ftp account ftp: /home/wwwroot/www.test.com //设置目录路径
```

```

Password: //自动完成
Enter it again: //自动完成
Created FTP User: ftp Sucessfully. //创建成功
[root@localhost ~]# lnmp ftp list //查看FTP目录列表
+-----+
| Manager for LNMP, Written by Licess |
+-----+
| https://lnmp.org |
+-----+
ftp /home/wwwroot/www.test.com/.//
List FTP User Sucessfully. //显示路径成功

```

3. 下载 WordPress 并上传

WordPress是一种非常流行的开源内容管理系统（CMS），用于创建各种类型的网站，从个人博客到大型企业网站。WordPress以易用性、灵活性以及庞大的插件生态系统而闻名。

WordPress是完全开源的，用户可以免费下载、使用和修改。即使没有编程经验，用户也可以通过直观的界面轻松创建和管理网站。该工具提供丰富的主题和插件，可以自定义网站的外观和功能。同时WordPress拥有庞大的社区，用户可以获得大量的帮助和支持。WordPress的核心代码非常安全，但需要定期更新插件和主题，以确保系统的安全性。用户可以到官网中下载该软件的部署包，如图7-9所示。



图 7-9

知识拓展

WordPress能搭建哪些网站?

WordPress是一个功能强大的内容管理系统（CMS），可以用于创建各种类型的网站，包括但不限于：博客网站、企业官方网站、在线商店（通过WooCommerce插件）、会员制网站、社交网络或社区网站、论坛（例如通过bbPress插件）、作品展示网站（如个人作品集）、新闻和杂志网站等。

下载后解压到任意目录，通过前面介绍的方式访问FTP服务器。大多数情况会通过第三方的FTP工具来连接FTP服务器，并将本地解压的WordPress文件上传到网站根目录中，如图7-10所示。

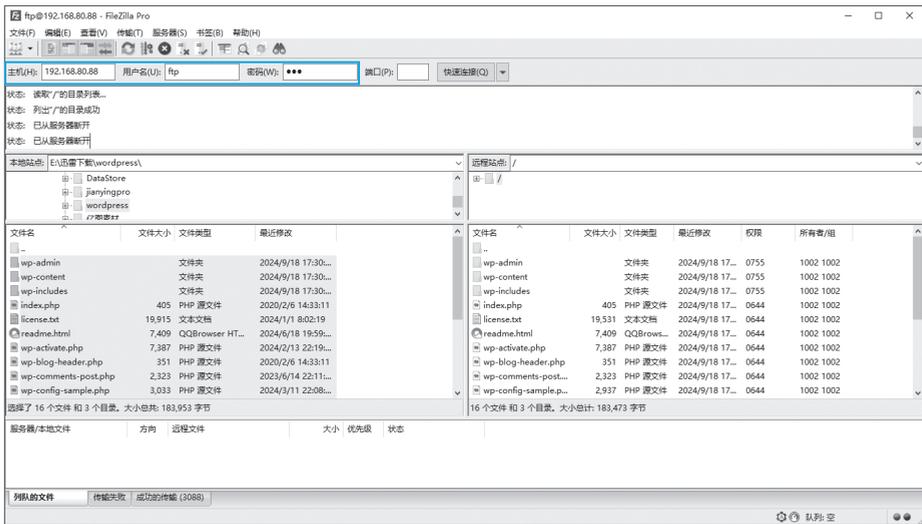


图 7-10

动手练 安装WordPress

安装方法非常简单，在安装文件上传完毕后，用户通过远程的浏览器访问该网站，或者使用“域名/index.php”的格式访问，即可使用WordPress的安装向导。接下来介绍具体的安装配置操作。



步骤01 欢迎界面显示了用户必须知道的一些LNMP参数，单击“现在就开始”按钮，如图7-11所示。

步骤02 选择安装WordPress的数据库（不是数据库服务器）的名称，以及数据库管理员的用户名和密码（之前手动设置的）。数据库主机名称保持默认，表前缀保持默认，单击“提交”按钮，如图7-12所示。

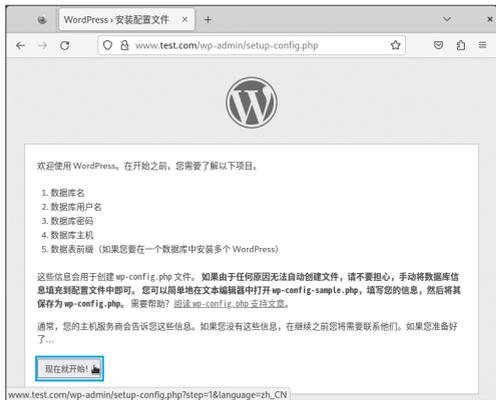


图 7-11

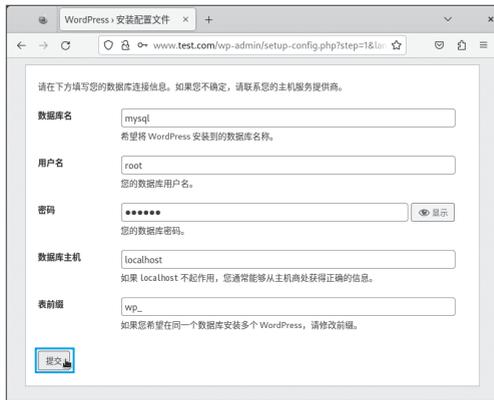


图 7-12

步骤03 数据库检测通过后，会提示可以安装，单击“运行安装程序”按钮，如图7-13所示。

步骤 04 接下来设置网站站点的相关信息，设置完毕后，单击“安装WordPress”按钮，如图7-14所示。

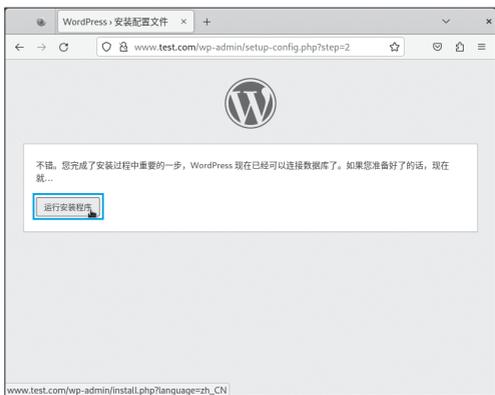


图 7-13

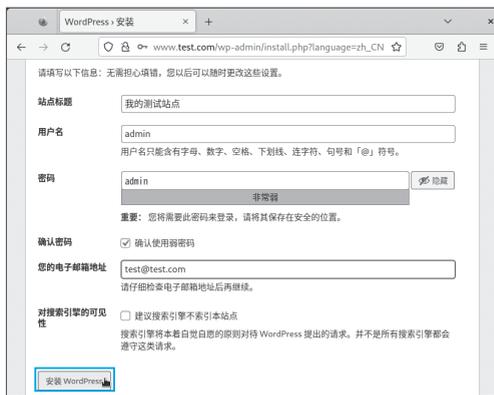


图 7-14

知识拓展

使用弱密码

如果确定需要使用弱密码，需要勾选“确认使用弱密码”复选框。

步骤 05 安装成功后会弹出成功提示，单击“登录”链接，如图7-15所示。

步骤 06 使用前面设置的登录名和密码登录后台进行网站管理，输入后单击“登录”按钮，如图7-16所示。

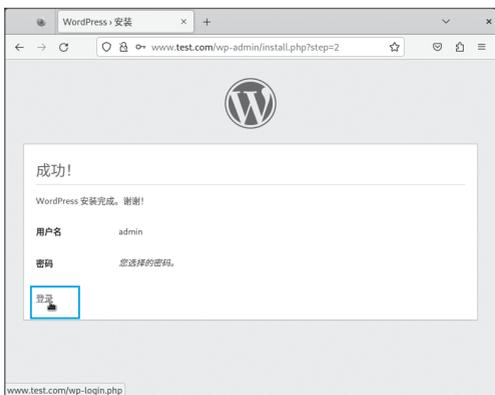


图 7-15



图 7-16

步骤 07 登录后就可以设置网站的后台，包括界面、文章、插件等，如图7-17所示。

步骤 08 如果要正常访问网站，可以在客户端的浏览器中输入网站的域名即可，如图7-18所示。

至此网站就搭建完毕，非常简单、方便、高效。除此之外，用户还可利用该环境安装如购物网站、博客、论坛等高级网站，如图7-19、图7-20所示。而且搭建过程无须考虑LNMP各组件之间的关联，LNMP一键部署工具已经帮助用户解决了这些问题。



图 7-17



图 7-18



图 7-19



图 7-20

7.2 Docker 容器

很多时候, 软件正常运行与其开发环境是紧密相关的。有时开发环境比较复杂, 造成了软件在其他环境中无法正常使用的情况, 或者为了使用软件, 需要进行复杂的配置过程, 多个不同的软件需要配置多种环境。为了解决这一问题, Docker 容器出现了。

7.2.1 认识 Docker

Docker 是一种开源的应用容器引擎, 它可以让开发者打包他们的应用以及依赖库到一个轻量级、可移植的容器中。这个容器可以运行在任何支持 Docker 的机器上, 从而实现应用的一致性部署。简单来说, Docker 就是将应用程序及其所依赖的环境打包成一个独立的容器, 这个容器可以在任何地方运行, 无须担心环境配置的问题。另外 Docker 还解决了不同应用之间的依赖冲突问题。

知识拓展

容器

容器是计算机上的沙盒进程, 与主机上的所有其他进程隔离。这种隔离利用内核命名空间和 cgroups, 是 Linux 中已经存在很长一段时间的一种功能。

1. Docker 的核心概念

Docker的核心概念有三个：镜像（image）、容器（container）和仓库（repository）。

（1）镜像。镜像是Docker中的基本构建块，它是一个轻量级、独立的可执行软件包，其中包含运行应用程序所需的所有内容，包括操作系统、代码、运行时、库、环境变量和配置文件等。镜像是只读的，意味着一旦构建完成，其内容不可更改。开发者可以使用Dockerfile定义镜像的构建规则，通过Docker命令将镜像构建出来。镜像可以用于创建Docker容器。

（2）容器。容器是基于镜像创建的运行实例。它是一个隔离的运行环境，可以在其中运行应用程序。容器包含镜像的副本，但它可以在镜像的基础上进行读写操作，因此容器是可变的。容器在运行时与宿主机操作系统隔离，但与宿主机共享内核。这使得容器能够快速启动，轻量级且可移植，同时提供高度的隔离性和安全性。

（3）仓库。仓库是用于存储Docker镜像的地方，它类似于代码库。仓库可以分为两种类型：公共仓库和私有仓库。公共仓库如Docker Hub，是供公众使用的，开发者可以将自己构建的镜像推送到公共仓库，也可以从公共仓库拉取其他开发者共享的镜像。私有仓库通常是企业内部使用的，用于存储私有镜像，保护公司的知识产权和应用程序代码。开发者可以通过Docker命令将镜像推送到私有仓库，并从私有仓库拉取镜像到自己的环境中使用。

2. Docker 的基本架构

在了解了Docker的核心概念后，下面讲解Docker的基本架构，如图7-21所示。Docker使用c/s架构，使用API远程管理和创建Docker容器。

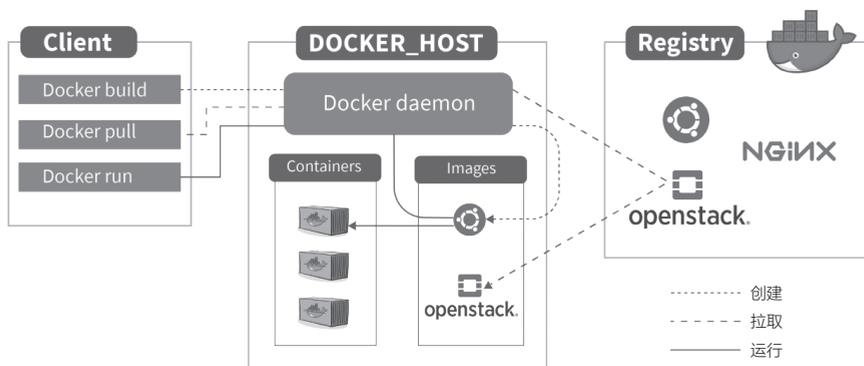


图 7-21

（1）Docker客户端（Client）。用于与Docker守护进程通信，发送命令以管理容器。Docker采用C/S架构。客户端和服务端既可以运行在一台计算机中，也可以通过Socket或者RESTful API进行通信。

（2）仓库（Registry）。用来保存镜像。该仓库中可以划分成更小的仓库，设置多个标签，每个标签对应一个镜像。通常一个仓库包含同一个软件不同版本的镜像，标签常

用于对应软件各个版本。可以通过“<仓库名>:<标签>”的格式指定具体是软件哪个版本的镜像。

(3) Docker主机(Docker_HOST)。一个物理的或者虚拟的设备,执行Docker守护进程和容器。其中包括:

① Docker守护进程(Docker daemon): Docker采用C/S架构, Docker daemon作为服务端接受来自客户端的请求,并进行处理(创建、运行、分发容器)。Docker daemon一般在宿主主机后台运行,等待接收来自客户端的消息。Docker客户端则为用户提供一系列可执行命令,用户用这些命令与Docker daemon实现交互。

② Docker镜像(Docker images): Docker镜像是用于构建Docker容器的静态文件,它包含应用程序运行所需的所有文件、依赖项和配置信息。Docker镜像可以从Docker Hub或其他镜像仓库中获取,也可以通过Dockerfile自定义构建。

③ Docker容器(Docker container): Docker容器是Docker镜像的运行实例,它包含应用程序及其依赖项,并在隔离环境中运行。每个容器都是一个独立的进程,拥有自己的文件系统、网络空间和进程空间。Docker容器通过Docker镜像来创建。容器与镜像的关系类似于面向对象编程中的对象与类。

3. Docker 的优势

与其他的虚拟化技术相比, Docker的主要优势如下。

- **快速部署:** 容器的启动速度非常快,可以快速部署应用。
- **轻量级:** 容器共享主机的内核,因此占用资源较少。
- **可移植性:** 容器可在任何支持Docker的主机上运行,实现一次构建、到处运行。
- **隔离性:** 每个容器都是相互隔离的,保证了应用的安全性。
- **高效利用资源:** 多个容器可以共享主机的资源,提高了资源利用率。

4. Docker 的常见管理命令

Docker的管理命令根据作用对象的不同,可以分为以下几类,如表7-1~表7-3所示。

表 7-1

镜像管理命令	
命令	功能
docker images	列出本地所有的镜像; -a: 显示所有镜像,包括中间层; -q: 只显示镜像ID
docker pull <镜像名>:<标签>	从仓库拉取镜像
docker push <镜像名>:<标签>	将本地镜像推送到仓库
docker rmi <镜像ID或镜像名>	删除镜像
docker build -t <镜像名>:<标签>	根据Dockerfile构建镜像

表 7-2

容器管理命令	
命令	功能
docker run -d <镜像名>	以守护进程的方式启动容器 -it: 进入容器的交互式终端 -p <主机端口>:<容器端口>: 映射端口 -v <主机路径>:<容器路径>: 挂载本地目录
docker ps	列出正在运行的容器 -a: 显示所有容器, 包括已停止的
docker start <容器ID或容器名>	启动容器
docker stop <容器ID或容器名>	停止容器
docker restart <容器ID或容器名>	重启容器
docker rm <容器ID或容器名>	删除容器
docker exec -it <容器ID或容器名> <命令>	在运行容器中执行命令
docker inspect <容器ID或容器名>	查看容器的详细信息
docker logs <容器ID或容器名>	查看容器的日志
docker attach <容器ID或容器名>	附加到一个正在运行的容器
docker commit <容器ID> <新镜像名>	从一个容器创建一个新的镜像

表 7-3

Dockerfile指令	
命令	功能
FROM	指定基础镜像
WORKDIR	设置工作目录
COPY	复制文件或目录到镜像中
ADD	类似COPY命令, 但支持远程URL和自动解压
RUN	执行命令
CMD	指定容器启动时默认执行的命令
ENTRYPOINT	指定容器的入口点, 类似CMD, 但可以被覆盖
ENV	设置环境变量
EXPOSE	声明容器暴露的端口
VOLUME	创建挂载卷

7.2.2 部署Docker

Docker的部署比较简单, 下面介绍具体的步骤。

步骤 01 添加Docker仓库。

```
[wlysy@localhost ~]$ sudo dnf config-manager --add-repo https://download.
docker.com/linux/centos/docker-ce.repo
添加仓库自: https://download.docker.com/linux/centos/docker-ce.repo
```

步骤02 安装Docker组件，共3个，使用默认安装方式即可。

```
[wlysy@localhost ~]$ sudo dnf install -y docker-ce docker-ce-cli
containerd.io
//安装这3个组件，自动安装依赖包
Docker CE Stable - x86_64 31 kB/s | 56 kB 00:01
依赖关系解决。
=====
软件包 架构 版本 仓库 大小
=====
安装:
containerd.io x86_64 1.7.22-3.1.e19 docker-ce-stable 43 M
docker-ce x86_64 3:27.2.1-1.e19 docker-ce-stable 27 M
docker-ce-cli x86_64 1:27.2.1-1.e19 docker-ce-stable 8.0 M
安装弱的依赖:
docker-buildx-plugin x86_64 0.16.2-1.e19 docker-ce-stable 14 M
docker-ce-rootless-extras x86_64 27.2.1-1.e19 docker-ce-stable 4.4 M
docker-compose-plugin x86_64 2.29.2-1.e19 docker-ce-stable 13 M
事务概要
=====
安装 6 软件包
总下载: 109 M
安装大小: 426 M
.....
已安装:
containerd.io-1.7.22-3.1.e19.x86_64
docker-buildx-plugin-0.16.2-1.e19.x86_64
docker-ce-3:27.2.1-1.e19.x86_64
docker-ce-cli-1:27.2.1-1.e19.x86_64
docker-ce-rootless-extras-27.2.1-1.e19.x86_64
docker-compose-plugin-2.29.2-1.e19.x86_64
完毕!
```

步骤03 启动服务并将其加入开机启动项目中，查看运行状态。

```
[wlysy@localhost ~]$ sudo systemctl start docker
[wlysy@localhost ~]$ sudo systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.
service → /usr/lib/systemd/system/docker.service.
[wlysy@localhost ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: d>
 Active: active (running) since Thu 2024-09-19 15:21:26 CST; 12s ago
   .....
```

```
[wlysy@localhost ~]$ sudo docker version
Client: Docker Engine - Community
 Version: 27.2.1
 API version: 1.47
```

```

Go version:      go1.22.7
Git commit:     9e34c9b
Built:         Fri Sep  6 12:09:42 2024
OS/Arch:       linux/amd64
Context:       default
Server: Docker Engine - Community
.....

```

【注意事项】无法直接安装

由于某些原因，Docker仓库服务器无法直接连接。用户在安装时需要为系统配置代理服务器，才能正确连接和使用。

7.2.3 Podman技术

在CentOS Stream 9中，默认安装了Podman，这是一个与Docker兼容的、更安全、更轻量级的容器引擎。如果用户对Docker生态系统非常熟悉，并且需要使用一些Docker特有的功能，可以考虑安装podman-docker，它提供与Docker兼容的命令行接口。

1. Podman 简介

Podman（POD管理器）是一个开源的Linux原生工具，旨在使用开放容器协议（OCI）容器和容器镜像轻松查找、运行、构建、共享和部署应用程序。它提供一个与Docker兼容的命令行界面，但其底层架构与Docker显著不同。两者的区别见表7-4所示。

表 7-4

特点	Docker	Podman
架构	客户端-服务器架构，需要一个守护进程(dockerd)	无守护进程架构，直接与容器交互
安全性	相对来说安全性较低，需要root权限	更安全，可以以非root用户运行
性能	性能优异，功能丰富	性能良好，更轻量级
生态系统	生态系统成熟，社区活跃	生态系统相对较新，但快速发展

2. 合理替换

从用户的角度，Podman与Docker的命令基本相似，都包括容器运行时、本地镜像、镜像仓库级别的命令。上面介绍的命令都适用于Podman。笔者不建议初学者在CentOS Stream 9中安装Docker。对于习惯使用Docker的读者，可以直接基于Podman定义Docker别名，以直接使用Podman的方式替换熟悉的Docker，而且仍然可以使用docker.io作为镜像仓库。

```

[wlysy@localhost ~]$ alias docker=podman //设置别名
[wlysy@localhost ~]$ alias //查看别名
alias docker='podman' //添加成功
alias egrep='egrep --color=auto'
.....

```

动手练 创建容器

因为已经默认集成了Podman，所以可以通过创建容器测试是否可以正常工作：



```
[wlisy@localhost ~]$ docker run hello-world //运行容器
Resolved "hello-world" as an alias (/etc/containers/registries.conf.
d/000-shortnames.conf) //解析成别名
Trying to pull quay.io/podman/hello:latest...
//本地没有，自动到quay.io仓库拉取名为podman/hello、标签为latest的镜像
Getting image source signatures //从仓库自动下载
Copying blob 81df7ff16254 done |
Copying config 5dd467fce5 done |
Writing manifest to image destination
!... Hello Podman World ...! //镜像程序输出表示镜像运行成功
 .--"---.
 / - - - \
 / (O) (O) \
 ~~~| -=(,Y,)= - |
 .---. /` \ |~~
~/ o o \~~~~.----. ~~
| =(X)= |~ / (O (O) \
~~~~~ ~| =(Y_)= - |
~~~~ ~~~| U |~~

Project: https://github.com/containers/podman //项目地址
..... //还有官网地址、桌面版、文档、视频、社交媒体账号
```

接下来可以列出所有容器：

```
[wlisy@localhost ~]$ podman ps -a //列出所有容器，使用Podman，支持补全
CONTAINER ID IMAGE STATUS PORTS COMMAND CREATED
df37d8cef8d1 quay.io/podman/hello:latest Exited (0) 15 minutes ago /usr/local/bin/po... 15
minutes ago goofy_greider
396b3ddc4844 quay.io/podman/hello:latest Exited (0) 10 minutes ago /usr/local/bin/po... 10
minutes ago intelligent_napier
```

这里显示了系统中存在的两个hello-world容器，基于同一个镜像创建。容器ID各不相同，当前这两个容器均为Exited（停止）状态。

7.2.4 部署Nginx容器

本小节将介绍如何使用容器来快速部署Nginx服务。为了方便演示和快速补全，这里使用原始命令Podman。Podman的命令格式和Docker命令是相同的，如果用户安装了Docker，将代码中的podman变为docker即可运行。

1. 下载镜像

用户可以手动下载镜像，也可以通过命令下载镜像。下载前，需要搭建网络环境，并设置CentOS Stream 9的网络代理，否则无法连接服务器。

因为当前有多个镜像满足要求，所以用户需要手动选择仓库。

```
[wlysy@localhost ~]$ podman pull nginx:latest
? Please select an image:
  registry.access.redhat.com/nginx:latest
  registry.redhat.io/nginx:latest
? docker.io/library/nginx:latest //选择该仓库，按回车键进行下载
```

接下来会自动连接并进行下载。

```
√ docker.io/library/nginx:latest
Trying to pull docker.io/library/nginx:latest... //从选择的仓库中下载镜像
Getting image source signatures
Copying blob 095d327c79ae done |
Copying blob 7bb6fb0cfb2b done |
Copying blob 24b3fdc4d1e3 done |
Copying blob bbfaa25db775 done |
Copying blob 0723edc10c17 done |
Copying blob a2318d6c47ec done |
Copying blob 3122471704d5 done |
Copying config 39286ab8a5 done |
Writing manifest to image destination
39286ab8a5e14aeaf5fdd6e2fac76e0c8d31a0c07224f0ee5e6be502f12e93f3
```

下载完毕，可以查看下载到本地的Nginx镜像，包括仓库、版本、ID、时间、大小等。

```
[wlysy@localhost ~]$ podman images
REPOSITORY          TAG          IMAGE ID       CREATED        SIZE
docker.io/library/nginx latest      39286ab8a5e1  5 weeks ago   192 MB
```

2. 运行容器

镜像是静态的概念，无法进行修改，用户需要以镜像为基础创建一个容器实例。只有实例可以被操作。

```
[wlysy@localhost ~]$ podman run --name nginx-cs -p 8080:80 -d nginx
04b572004bdf3a96b6861a98daf8983b5ce1a0c157b6c3cea434896973f95721
```

--name nginx-cs: 创建一个名为nginx-cs的容器。

-p 8080:80: 端口映射，将本地的8080端口映射到容器内部的80端口。

-d nginx: 设置容器在后台运行，并返回容器的ID值。下一行为容器的ID值，因为过长，使用时一般使用该ID值的前12个字符。

接下来查看现在运行的容器的相关信息。

```
[wlysy@localhost ~]$ podman ps
CONTAINER ID  IMAGE          COMMAND          CREATED
```

STATUS	PORTS	NAMES
04b572004bdf	docker.io/library/nginx:latest	nginx -g daemon o...
seconds ago	Up 9 seconds	0.0.0.0:8080->80/tcp, 80/tcp nginx-cs

可以从中看到容器的ID值、容器使用的镜像、容器启动时运行的命令参数、创建时间、容器状态、容器的端口信息和使用的连接类型、容器的名称。

3. 测试容器

使用浏览器访问本地服务器（容器中的网站），带上端口号是8080，看是否可以正常访问，如图7-22所示。



图 7-22

4. 编辑容器

默认情况下容器内部的服务使用的是默认参数，如果想要和本机安装的服务一样去管理容器中的服务配置参数，需要进入容器中进行设置。

```
[wlysy@localhost ~]$ podman exec -it nginx-cs /bin/bash //进入后台容器中
root@04b572004bdf:/# cd /usr/share/nginx/html/ //可以看到命令提示符的变化
root@04b572004bdf:/usr/share/nginx/html# ls //进入并查看网站主页面文件
50x.html index.html
root@04b572004bdf:/usr/share/nginx/html# cat > index.html <<EOF
//因为在该模式下，无法使用vim或vi来编辑文件，所以使用重定向进行输入
> Hello World
> EOF
root@04b572004bdf:/usr/share/nginx/html# cat index.html
Hello World //输入成功
root@04b572004bdf:/usr/share/nginx/html# nginx -s reload //重启Nginx服务
2024/09/19 09:29:35 [notice] 36#36: signal process started
root@04b572004bdf:/usr/share/nginx/html# curl 127.0.0.1 //输入本主机的网页内容
Hello World //输出成功，网页服务器工作正常
```

退出后可以用浏览器进行测试，如图7-23所示。

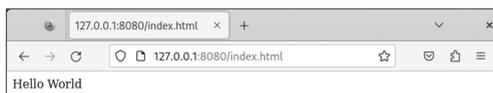


图 7-23

动手练 使用命令修改文件

除了进入容器内部管理和修改配置文件外，还可以在容器运行的宿主机上直接创建文件，并复制到容器的目标目录中使用。



```
[wlysy@localhost ~]$ vim index.html //创建并编辑网页文件
[wlysy@localhost ~]$ podman cp index.html nginx-cs:/usr/share/nginx/html/
[wlysy@localhost ~]$ sudo curl 127.0.0.1:8080 //复制后测试输出
TEST NGINX-CS //编辑成功
```

打开浏览器测试Nginx的运行，如果工作正常，会显示如图7-24所示的内容。

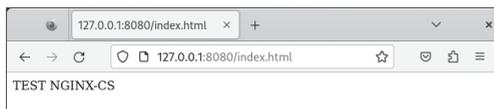


图 7-24



知识延伸：Java环境的搭建

Java环境的搭建比较简单，首先检查系统中是否安装了Java，如果有，建议卸载。

```
[wlysy@localhost ~]$ java -version
openjdk version "1.8.0_362" //已安装了Java
OpenJDK Runtime Environment (build 1.8.0_362-b08)
OpenJDK 64-Bit Server VM (build 25.362-b08, mixed mode)
[wlysy@localhost ~]$ sudo dnf list installed | grep java //筛选已安装的软件包
[sudo] wlysy 的密码:
java-1.8.0-openjdk-headless.x86_64 1:1.8.0.362.b09-4.e19 @AppStream
javapackages-filesystem.noarch 6.0.0-4.e19 @AppStream
tzdata-java.noarch 2024a-2.e19 @AppStream
[wlysy@localhost ~]$ sudo dnf remove java-1.8.0-openjdk-headless-1:1.8.0.362.
b09-4.e19.x86_64 //移除已安装的Java
依赖关系解决。
.....
已移除:
copy-jdk-configs-4.0-3.e19.noarch
java-1.8.0-openjdk-headless-1:1.8.0.362.b09-4.e19.x86_64
..... //移除成功
```

接下来安装Java-21版本。

```
[wlysy@localhost ~]$ sudo dnf install java-21
.....
安装:
java-21-openjdk x86_64 1:21.0.2.0.13-2.e19 appstream 422 k
安装依赖关系:
copy-jdk-configs noarch 4.0-3.e19 appstream 28 k
.....
已安装:
copy-jdk-configs-4.0-3.e19.noarch
.....
完毕! //安装成功
```

查看Java的版本，正常显示说明配置成功了。

```
[wlysy@localhost ~]$ java -version
openjdk version "21.0.2" 2024-01-16 LTS
OpenJDK Runtime Environment (Red_Hat-21.0.2.0.13-1) (build 21.0.2+13-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-21.0.2.0.13-1) (build 21.0.2+13-LTS,
mixed mode, sharing)
```