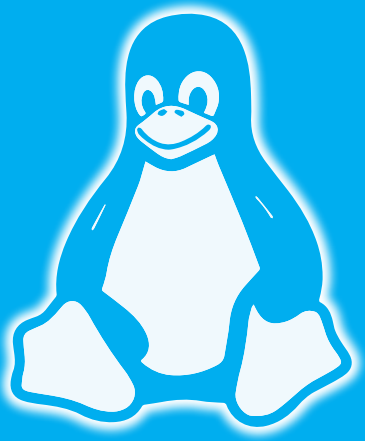
服务篇



第5章 文件共享与传输服务

在Linux服务器运维与企业内网架构中，文件共享与传输始终是最基础且高频的需求之一。无论是实现部门间文档协作、服务节点间配置同步，还是进行跨主机备份与传输，管理员都需要掌握多种可控、安全、高效的文件服务工具。本章将聚焦于Ubuntu Server平台，系统讲解几种主流的文件共享服务与传输方式的部署与管理方法。

5.1 Samba文件共享服务部署与管理

Samba是一款实现SMB/CIFS协议的开源软件，允许Linux系统与Windows系统之间进行文件和打印服务的互通。在企业和局域网环境中，Samba常用于构建跨平台文件共享服务，使Linux服务器可以像Windows文件服务器一样被访问。相比于FTP等传统传输方式，Samba具备共享目录可浏览、权限控制细致、兼容性强等优势。

5.1.1 安装Samba服务

在Ubuntu Server系统中部署Samba服务的第一步是完成软件包的安装。Samba在大多数Linux发行版的软件源中都有提供，因此用户可以通过APT包管理器便捷地完成安装过程。Samba的核心组件包括smbd（提供文件和打印服务）与nmbd（提供NetBIOS名称解析与网络浏览支持），但在新版本中这两个守护进程已合并简化为smbd。

1. 安装 Samba 组件和命令行工具

在配置好软件源的基础上，执行正常的安装命令，服务端安装包名称叫作samba，包含smbd服务守护进程及其配置文件。如该系统需作为samba客户端使用，则要安装客户端程序安装包smbclient。两者一起并默认安装的效果如下：

```
wlysy@testus:~$ sudo apt install samba smbclient -y      # 服务端与客户端同时安装
[sudo] password for wlysy:                               # 验证账户密码
Reading package lists... Done
……                                                       # 自动下载并安装
wlysy@testus:~$ smbd --version
Version 4.19.5-Ubuntu                                    # 查看版本，安装成功
```

2. 检查并启动 Samba 服务

Samba服务安装完成后会自动注册为Systemd服务。可通过以下命令检查其状态：

```
wlysy@testus:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon                        # 服务名smbd.service
     Loaded: loaded (/usr/lib/systemd/system/smbd.service; enabled; preset:
enabled)                          # 服务定义文件路径，系统启动时自动启动，服务预设就是启动
     Active: active (running) since Fri 2025-07-18 06:25:37 UTC; 20min ago
                                  # smbd服务当前是正常运行状态，运行了20分钟
       Docs: man:smbd(8)          # samba服务的一些帮助文档
```

```
……
     Process: 14000 ExecCondition=/usr/share/samba/is-configured smb (code=exited,
status=0/SUCCES>                          # 服务运行前的一个条件检查脚本，用来验证是否正确配置
   Main PID: 14032 (smbd)                 # 服务主进程的进程ID为14032，括号中是进程名称
     Status: "smbd: ready to serve connections..."       # smbd等待客户端连接请求
      Tasks: 3 (limit: 4548)              # 当前服务关联的进程/线程数量，以及最大任务数
     Memory: 7.6M (peak: 8.3M)            # 当前占用的内存量，以及曾经的最大内存占用量
        CPU: 55ms                                 # 服务启动以来累计消耗的CPU时间
     CGroup: /system.slice/smbd.service           # 服务所属的控制组
             ├─14032 /usr/sbin/smbd --foreground --no-process-group
             ├─14060 "smbd: notifyd" .
             └─14061 "smbd: cleanupd "    # 进程数，显示了主进程及衍生进程
Jul 18 06:25:36 testus systemd[1]: Starting smbd.service - Samba SMB Daemon...
Jul 18 06:25:36 testus (smbd)[14032]: smbd.service: Referenced but unset
environment variable eva>
Jul 18 06:25:37 testus systemd[1]: Started smbd.service - Samba SMB Daemon.
                                          # 相关的日志信息
```

如果服务未启动可以手动启动，执行的命令为：

```
wlysy@testus:~$ sudo systemctl start smbd
```

知识点拨：服务的管理操作

在服务管理操作中，除了查看服务使用status、启动服务使用start外，其他常用的还有restart（重启）、stop（停止）。将命令中的关键字进行替换，就可以实现这些功能。

如果未加入开机启动，可以使用命令：

```
wlysy@testus:~$ sudo systemctl enable smbd
Synchronizing state of smbd.service with SysV service script with /usr/lib/
systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable smbd
```

知识点拨：取消开机启动

取消开机启动，只需在命令中将enable换成disable即可。

3. 开放防火墙端口

若系统启用了UFW防火墙，需允许Samba相关端口访问（TCP 139和445，UDP 137和138）：

```
wlysy@testus:~$ sudo ufw status           # 当前没有规则
Status: active
wlysy@testus:~$ sudo ufw allow 'Samba'    # 允许Samba服务
wlysy@testus:~$ sudo ufw status
Status: active
To                         Action      From
--                         ------      ----
Samba                      ALLOW       Anywhere
Samba (v6)                 ALLOW       Anywhere (v6)
```

5.1.2　配置共享目录与权限

完成Samba服务安装后，下一步是配置具体的共享目录，并设置对应的访问权限，以便其他客户端设备可以通过网络访问本机上的共享资源。

1. 创建共享目录

在实际部署中，应将共享目录放置于系统文件结构清晰的位置，例如/srv/samba/下：

```
wlysy@testus:~$ sudo mkdir -p /srv/samba/shared
wlysy@testus:~$ sudo chown nobody:nogroup /srv/samba/shared     # 匿名共享
wlysy@testus:~$ sudo chmod 0775 /srv/samba/shared/     #所有者和所属组读写，其他只读
```

2. 编辑 Samba 配置文件

Samba主配置文件路径为/etc/samba/smb.conf，在编辑前，先复制一个该文档备份：

```
wlysy@testus:~$ sudo cp /etc/samba/smb.conf /etc/samba/smb.conf.bak    # 备份
wlysy@testus:~$ sudo vim /etc/samba/smb.conf                           # 启动编辑
```

在文件末尾添加新的共享段，用于定义共享名称、路径和权限等，完成后保存退出。

```
[PublicShare]                              # 共享名称，在客户端访问时表现为共享资源名称
   path = /srv/samba/shared                # 实际的共享目录路径
   browseable = yes                        # 允许该共享在网络浏览中可见
   read only = no                          # 启用写入权限
   guest ok = yes                          # 允许匿名访问
   create mask = 0664
# 新建文件的默认权限，所有者和组可以读取和写入，其他用户只读
   directory mask = 0775
# 创建目录的权限掩码，所有者和组可以读写及进入目录，其他用户可以读和进入目录
```

3. 测试及重启服务

修改完配置文件后，先使用以下命令检查语法是否正确，并重启服务：

```
wlysy@testus:~$ testparm                   # 检查Samba配置文件smb.conf的语法和有效性
Load smb config files from /etc/samba/smb.conf # 加载的文件路径
Loaded services file OK.                   # 成功加载，没有发现语法错误和配置问题
……
wlysy@testus:~$ sudo systemctl restart smbd       # 重启服务
```

4. 测试访问

Linux用户可以在本地进行访问测试，将localhost换为目标IP，就可以访问其他主机了。

```
wlysy@testus:~$ smbclient -L localhost -N         # 匿名访问并列出Samba共享目录
        Sharename       Type      Comment
        ---------       ----      -------
        print$          Disk      Printer Drivers
        PublicShare     Disk                    # 前面创建的共享
        IPC$            IPC       IPC Service (testus server (Samba, Ubuntu))
wlysy@testus:~$ smbclient //localhost/PublicShare -N     # 匿名访问共享目录
Try "help" to get a list of possible commands.
smb: \> ls                                                # 列出当前目录的内容
  .                                   D        0  Fri Jul 18 07:33:51 2025
  ..                                  D        0  Fri Jul 18 07:33:51 2025
                39937312 blocks of size 1024. 30577760 blocks available
smb: \> mkdir 123                                         # 创建目录123
smb: \> ls                                                # 查看是否创建成功
  .                                   D        0  Fri Jul 18 08:07:26 2025
  ..                                  D        0  Fri Jul 18 08:07:26 2025
```

```
  123                                 D        0  Fri Jul 18 08:07:26 2025
                39937312 blocks of size 1024. 30577756 blocks available
smb: \> exit                                          # 退出
```

动手练 Windows访问Samba共享目录


扫码看视频

在Windows中，可以在资源管理器的地址栏或者使用“Win+R”打开“运行”界面，输入“\\服务器IP\共享目录名\”来访问共享，如图5-1所示，并且可以创建文件和目录，如图5-2所示。

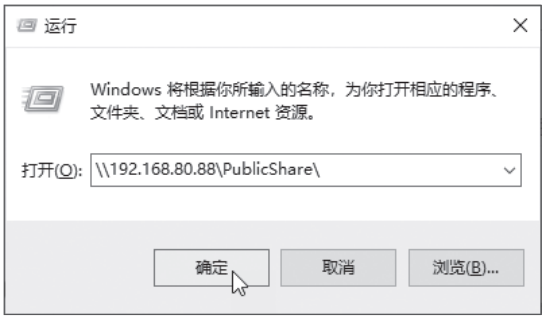


图 5-1

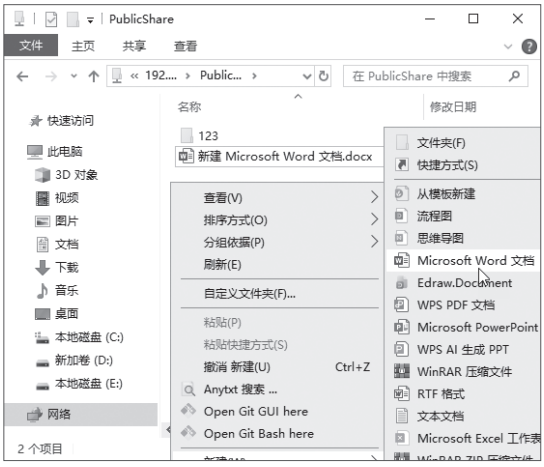


图 5-2

5.1.3　用户身份认证与访问控制策略

通常会在Samba文件共享中设置用户身份认证和访问控制策略，以确保共享资源的安全性和可控性。Samba服务默认允许匿名访问，但在实际生产环境中，应配置为基于用户账户的认证方式，以防止未经授权的用户访问共享目录。

1. 添加系统用户并设置 Samba 密码

Samba的用户必须是系统中的有效用户账户，并需要为其单独设置Samba密码。以下操作以alice用户为例进行说明：

```
wlysy@testus:~$ sudo useradd -M alice       # 创建用户alice，不为该用户创建主目录
wlysy@testus:~$ sudo smbpasswd -a alice     # 为该用户创建对应的Samba访问密码
New SMB password:                           # 输入密码
Retype new SMB password:                    # 输入验证密码
Added user alice.                           # 创建成功
```

设置完成后，该用户就能通过Samba服务访问被授权的共享资源。如需启用该用户的Samba访问权限（一般默认开启），可使用命令sudo smbpasswd -e alice。

2. 配置共享目录的用户访问控制

在Samba配置中，创建一个新的共享目录SecureShare的访问配置：

```
[SecureShare]                               # 新共享的名称
   path = /srv/samba/secure                 # 共享目录路径
   valid users = alice                      # 允许访问的用户列表
```

```
   read list = alice                          # 允许读取的用户列表
   write list = alice                         # 允许写入的用户列表
```

这里可以进行精细化配置，为目录设置不同的用户所具有的权限，以增强安全性。如果有多个用户，用户之间用空格来分开。

3. 创建并设置目录的权限

仅配置smb.conf还不够，文件系统本身也必须授权对应用户访问共享目录。

```
wlysy@testus:~$ sudo mkdir -p /srv/samba/secure          # 创建目录
wlysy@testus:~$ sudo chown root:alice /srv/samba/secure/
wlysy@testus:~$ sudo chmod 770 /srv/samba/secure/
```

修改目录属于root以及alice用户所在的组。所有者权限为读/写/执行，组用户权限为读/写/执行，而其他用户没有权限。

知识点拨：多用户共享该安全目录

可将这些用户加入同一组（如sambashare），通过“sudo usermod-aG sambashare 用户名”将所有用户添加到组中。将目录所有者改为root:sambashare即可。

4. 验证访问

使用前面介绍的方法访问该共享，因为不是匿名访问，会弹出对话框，要求输入用户名和密码，用户使用alice和设置的密码就可以访问该目录了，如图5-3所示。而使用其他用户，即使输入了该用户的正确密码，也会拒绝访问，如图5-4所示。

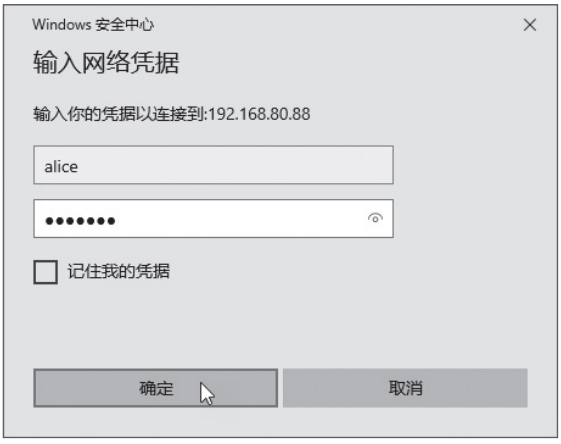


图 5-3

图 5-4

注意事项 | 禁用匿名访问 |

如果Samba服务器禁止匿名访问，可以在全局配置中进行设置，这样，未登录的用户将无法访问任何共享资源。

```
[global]
   map to guest = never
```

5.2 NFS部署与管理

NFS（Network File System，网络文件系统）是一种经典的类UNIX系统网络文件共享协议，它允许一台计算机通过网络挂载另一台计算机上的目录，像访问本地文件系统一样进行操作。与Samba更适用于Linux-Windows间共享不同，NFS在Linux-to-Linux环境下具备更高的性能与兼容性，适合企业内部集群、服务器间共享等场景。

5.2.1 部署NFS

NFS服务由服务器端组件nfs-kernel-server和客户端组件nfs-common构成，用户可以手动安装这两个组件：

```
wlysy@testus:~$ sudo apt install nfs-kernel-server nfs-common -y
```

安装完毕，可以检查该服务是否启动：

```
wlysy@testus:~$ systemctl status nfs-server
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled;
preset: enabled)
     Active: active (exited) since Fri 2025-07-18 09:40:24 UTC; 42s ago
.....
```

可以看到服务已经正常运行，且已经为开机启动。如未启动，可以使用命令sudo systemctl start nfs-server启动服务，使用命令sudo systemctl enable nfs-server设置为开机启动。

NFS依赖rpcbind服务来管理远程过程调用端口。系统通常在安装nfs-kernel-server时一并启用，但仍建议手动检查状态。

```
wlysy@testus:~$ sudo systemctl status rpcbind
● rpcbind.service - RPC bind portmap service
     Loaded: loaded (/usr/lib/systemd/system/rpcbind.service; enabled; preset:
enabled)
     Active: active (running) since Fri 2025-07-18 09:40:22 UTC; 3min 44s ago
```

如未启动，可使用命令启动，并加入到开机启动中。

如系统启用了UFW防火墙，需开放以下与NFS相关的服务端口，以及允许访问的网络：

```
wlysy@testus:~$ sudo ufw allow nfs          # 放行nfs
wlysy@testus:~$ sudo ufw allow 111/tcp      # 放行rpcbind tcp 111端口
wlysy@testus:~$ sudo ufw allow 111/udp      # 放行rpcbind udp 111端口
```

5.2.2 配置NFS共享目录及权限

在安装完NFS服务组件后，下一步是配置NFS服务器端的共享目录与访问权限。NFS通过配置/etc/exports文件来定义哪些目录可以共享、允许哪些客户端访问以及具体的权限控制策略。此过程是实现NFS服务功能的核心环节，直接决定了远程客户端是否能顺利访问共享目录以及访问权限的高低。

1. 创建共享目录并赋予权限

首先在服务器端创建一个要共享的目录，并为其设置合适的权限。下面以/srv/nfs/share作为

示例，创建并赋予对应所需的权限：

```
wlysy@testus:~$ sudo mkdir -p /srv/nfs/share                    # 创建共享目录
wlysy@testus:~$ sudo chown nobody:nogroup /srv/nfs/share/       # 允许匿名访问
wlysy@testus:~$ sudo chmod 777 /srv/nfs/share/                  # 所有人均可读写
```

2. 编辑配置文件

NFS使用/etc/exports来定义共享目录和访问规则。使用文本编辑器编辑该文件，添加如下配置：

```
/srv/nfs/share 192.168.80.0/24(rw,sync,no_subtree_check)
```

其中，/srv/nfs/share：被共享的目录路径。192.168.80.0/24：允许访问该目录的客户端网段。rw：允许读写访问。sync：每次写入的数据都同步到磁盘，保证数据一致性。no_subtree_check：禁止子目录检查，提高性能和兼容性。

如需允许多个主机访问，可添加多个IP地址或CIDR段，或使用通配符，如*（不推荐用于生产环境）。

3. 应用配置并启动服务

完成配置后，通过以下命令导出共享目录：

```
sudo exportfs -a
```

然后确保NFS服务正在运行：

```
wlysy@testus:~$ sudo systemctl restart nfs-server               # 重启服务
wlysy@testus:~$ sudo systemctl status nfs-server                # 查看服务状态
● nfs-server.service - NFS server and services
     Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled;
preset: enabled)
     Active: active (exited) since Sat 2025-07-19 05:31:54 UTC; 9s ago # 正常
……
wlysy@testus:~$ sudo systemctl enable nfs-server                # 添加到开机启动
```

此时，NFS服务已开始监听请求，共享目录已经向指定网段开放。

知识点拨：更改参数后的刷新

若修改了/etc/exports文件，后续再次运行sudo exportfs -a即可更新生效，无须重启服务。

注意事项 | 省略后缀 |

在systemd中，如果用户省略了服务的.service后缀，systemctl会自动补全为.service单位类型（默认类型）。所以sudo systemctl start nfs-server和sudo systemctl start nfs-server.service是等价的。其他服务，如ssh和ssh.service一样。

4. 验证共享配置

可以通过以下命令查看当前已导出的共享目录及其权限设置：

```
wlysy@testus:~$ sudo exportfs -v
/srv/nfs/share  192.168.80.0/24(sync,wdelay,hide,no_subtree_
check,sec=sys,rw,secure,root_squash,no_all_squash)
```

确保输出中包含用户定义的共享路径和允许访问的客户端地址。

5.2.3　客户端挂载与测试访问

在NFS服务器端完成共享目录的配置之后，客户端需进行挂载操作才能访问远程目录。作为客户端的设备，需要安装nfs的客户端组件nfs-common。如未安装需手动安装。以下是在另一台客户端设备上完成的。

1. 创建挂载点及挂载

挂载点是NFS目录在本地文件系统中的映射位置，建议创建专用目录：

```
wlysy@myus:~$ sudo mkdir -p /mnt/nfs_share          # 创建用于挂载服务器的共享目录
```

使用mount命令将远程共享目录挂载到本地。假设NFS服务器地址为192.168.80.88，共享目录为/srv/nfs/share：

```
wlysy@myus:~$ sudo mount 192.168.80.88:/srv/nfs/share /mnt/nfs_share
wlysy@myus:~$ mount | grep nfs                      # 查看列出的挂载信息
192.168.80.88:/srv/nfs/share on /mnt/nfs_share type nfs4 (rw,relatime,vers=4.2,
rsize=524288,wsize=524288,namlen=255,hard,proto=tcp,timeo=600,retrans=2,sec=sys,
clientaddr=192.168.80.107,local_lock=none,addr=192.168.80.88)
```

注意事项 | 挂载路径的挂载格式 |

注意挂载路径格式为“服务器IP:共享目录路径”，后面是本地挂载点。

2. 测试挂载后使用

接下来就可以进入挂载目录中，进行创建操作：

```
wlysy@myus:~$ cd /mnt/nfs_share/
wlysy@myus:/mnt/nfs_share$ ls
wlysy@myus:/mnt/nfs_share$ touch test.txt           # 创建文件
wlysy@myus:/mnt/nfs_share$ ls
test.txt                                            # 创建成功
```

动手练　配置自动挂载


扫码看视频

为实现系统启动时自动挂载，可将挂载信息写入/etc/fstab文件。添加如下配置行：

```
192.168.80.88:/srv/nfs/share  /mnt/nfs_share  nfs  defaults,_netdev  0  0
```

服务器的IP地址可更换为读者所使用的服务器IP，_netdev表示该挂载依赖网络服务，系统会在网络就绪后再挂载。配置完成后，可以使用sudo mount -a进行测试。如果需要卸载挂载点，使用sudo umount /mnt/nfs_share即可。

5.3　FTP与SFTP服务部署与管理

在日常运维与文件传输中，FTP（File Transfer Protocol，文件传输协议）作为一种传统而广泛支持的文件传输协议，仍然具有重要价值，尤其是在需要跨平台共享文件的场景中。FTP服务具有部署简单、兼容性好、访问控制灵活等特点，广泛应用于企业内网文件分发、自动化脚本文件同步等环境中。

5.3.1 安装与启动FTP

vsftpd是Ubuntu系统默认源中提供的FTP服务软件，具有稳定性高、配置简单、安全性较好的特点，适合在中小型局域网环境中部署使用。

1. 安装 vsftpd 服务

在终端执行以下命令安装vsftpd软件包：

```
wlysy@testus:~$ sudo apt install vsftpd -y
Reading package lists... Done
……
wlysy@testus:~$ sudo systemctl status vsftpd
● vsftpd.service - vsftpd FTP server
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset:
enabled)
     Active: active (running) since Thu 2025-07-24 02:21:03 UTC; 1min 19s ago
……                                                  # 当前是活动状态，默认开机启动
```

如果当前是非活动状态，用户可以手动启动（sudo systemctl start vsftpd），手动设置为开机启动（sudo systemctl enable vsftpd）。

2. 配置防火墙

如果服务器启用了防火墙（如UFW），还需手动放行FTP服务所需的端口。FTP协议默认使用21端口，可以使用以下命令来放行FTP服务：

```
wlysy@testus:~$ sudo ufw allow ftp
Rule added
Rule added (v6)
wlysy@testus:~$ sudo ss -tnlp | grep :21          # 查看当前是否在监听21端口
LISTEN 0      32                 *:21               *:*    users:(("vsftpd",pid=
2508,fd=3))
```


扫码看视频

动手练 连接服务器

在客户端中，可以使用命令来测试到FTP服务器的连接，命令格式为“ftp 服务器IP地址”。若连接成功，会进入FTP交互界面，并提示输入用户名和密码（默认情况下为系统用户账户）。

（1）Windows连接FTP服务器

Windows连接FTP服务器，可以在CMD或者PowerShell中进行：

```
C:\Users\YSY>ftp 192.168.80.88              # 在CMD中使用命令连接FTP服务器
连接到 192.168.80.88。                      # 成功建立连接
220 (vsftpd 3.0.5)                          # vsftpd服务器版本
200 Always in UTF8 mode.                    # 服务器提示使用UTF8编码
用户(192.168.80.88:(none)): wlysy           # 输入登录的用户名
331 Please specify the password.            # 服务器要求输入密码
密码:                                       # 输入密码（默认不显示）
230 Login successful.                       # 提示登录成功
ftp> dir                                    # 列出当前目录中的文件和目录（也可以使用ls）
200 PORT command successful. Consider using PASV.
# FTP客户端使用了PORT模式，服务器建议使用PASV模式（更适用于有防火墙的环境）
150 Here comes the directory listing.
-rw-rw-r--    1 1000     1000            0 Jul 11 08:04 test # 当前目录中的文件
226 Directory send OK.                       # 目录列表发送完成
```

```
ftp: 收到 65 字节，用时 0.00秒 65.00千字节/秒。          # 客户端显示接收详情
ftp> pwd                                                  # 查看当前的目录
257 "/home/wlysy" is the current directory                # 当前位于该用户的主目录
```

（2）Linux连接FTP服务器

Linux连接的话使用的命令与Windows相同，连接后的使用也是一样的：

```
wlysy@myus:~$ ftp 192.168.80.88
Connected to 192.168.80.88.
220 (vsftpd 3.0.5)
Name (192.168.80.88:wlysy): wlysy
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

5.3.2　配置vsftpd服务参数

在成功安装vsftpd服务并启用本地用户登录后，默认配置往往限制了很多常见操作，例如用户无法上传文件、创建目录，或者客户端ftp ls命令卡顿。这是因为vsftpd出于安全考虑，对匿名访问、本地权限、被动模式端口等做了严格限制。下面将通过修改配置文件/etc/vsftpd.conf来优化这些设置，从而实现一个功能完整且相对安全的FTP服务环境。

1. 启用本地用户上传与写权限

系统中的用户账号可使用其用户名和密码登录FTP服务，并可执行写操作。为了增强安全性，应进一步限定本地用户的访问范围。默认情况下，本地用户虽然可以登录FTP，但无权上传或修改文件。可以通过sudo vim /etc/vsftpd.conf命令来对配置文件中的参数进行设置，启用2个参数：

```
write_enable=YES                                          # 启用该参数，允许上传、重命名、删除等写操作
local_umask=022                                           # 启用设置上传文件默认权限
```

注意事项 | listen=NO |

以往配置时，都需要将listen的值设置为YES，但当前的值为NO，依然可以进行FTP服务，这是因为在现代Ubuntu Server（如24.04）中，vsftpd默认是通过systemd socket 激活机制启动的（以前是独立模式监听IPv4），它不依赖 listen=YES/NO。

完成后重启服务“wlysy@testus:/home$ sudo systemctl restart vsftpd”，接下来重新登录用户，即可创建：

```
ftp> mkdir 123                                            # 未修改配置文件前，登录并创建文件
550 Permission denied.                                    # 拒绝创建
……                                                        # 修改配置文件后，重启服务，重新登录
ftp> mkdir 123                                            # 再次创建文件夹
257 "/home/wlysy/123" created                             # 提示创建成功
ftp> ls                                                   # 查看当前目录
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
123                                                       # 创建成功
```

```
test
226 Directory send OK.
ftp: 收到 14 字节，用时 0.00秒 14.00千字节/秒。
```

知识点拨：配置chroot隔离以提高安全性

vsftpd默认启用了chroot机制，将本地用户限制在自己的家目录内，防止其访问系统其他目录。如果启用了上传权限，但仍提示550 Permission denied，可能是chroot影响了写操作。需要在配置文件中开启chroot_local_user=YES，并创建allow_writeable_chroot=YES。这允许用户在chroot环境下也具有写权限（如上传文件、创建目录），避免报错。

2. 配置被动模式支持

使用Linux的FTP客户端时，ls或dir会卡住，常见原因是被动模式的端口未开放。

```
wlysy@myus:~$ ftp 192.168.80.88          # 在Linux中，登录FTP服务器
……
ftp> mkdir 234                           # 登录成功，尝试创建目录234
257 "/home/wlysy/234" created            # 配置文件启动本地用户上传和写权限，会成功创建
ftp> ls                                  # 查看当前目录
229 Entering Extended Passive Mode (|||21204|) # 被动模式端口未打开，报错
^C                                       # 只能使用Ctrl+C强制结束
receive aborted. Waiting for remote to finish abort.
ftp>
```

为此，需要在配置文件中开启被动模式并限制端口范围，需要创建3条参数：

```
anonymous_enable=YES
anon_upload_enable=NO
anon_mkdir_write_enable=NO
```

接下来需要在UFW中开启并放行对应端口：

```
wlysy@testus:/home$ sudo ufw allow 30000:30050/tcp
Rule added
Rule added (v6)
```

接下来重启服务，再使用Linux连接后，就可以查看了：

```
ftp> ls
229 Entering Extended Passive Mode (|||30042|)
150 Here comes the directory listing.
drwxr-xr-x    2 1000     1000         4096 Jul 24 03:48 123
drwxr-xr-x    2 1000     1000         4096 Jul 24 05:19 234     # 创建成功
-rw-rw-r--    1 1000     1000            0 Jul 11 08:04 test
226 Directory send OK.
```

3. 允许或限制匿名访问

如果不希望任何未授权用户访问FTP，可明确关闭匿名登录，在配置文件中设置“anonymous_enable=NO”。如用户确实希望提供公共下载服务，则可开启匿名访问，但限制为只读：

```
anonymous_enable=YES                     # 允许匿名访问
anon_upload_enable=NO                    # 开启并将上传权限设置为NO
anon_mkdir_write_enable=NO               # 开启并将创建目录写权限设置为NO
```

接下来使用匿名账户登录：

```
C:\Users\YSY>ftp 192.168.80.88
……
用户(192.168.80.88:(none)): anonymous          # 匿名登录使用的用户名为anonymous
331 Please specify the password.
密码:                                          # 匿名访问，密码任意输入即可
230 Login successful.                          # 登录成功
ftp> ls                                        # 可以查看目录内容
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
226 Directory send OK.
ftp> mkdir 345                                 # 创建目录
550 Permission denied.                         # 创建请求被拒绝
```

知识点拨：FTP匿名用户默认目录

默认匿名用户的FTP根目录通常是/srv/ftp/，如果要允许匿名用户创建和修改目录，需要开启上传和创建目录写权限（anon_upload_enable=YES、anon_mkdir_write_enable=YES）。另外，还需设置目录权限，不建议直接修改ftp目录本身的权限（默认是755），会造成无法登录及操作的情况，可以在其下创建用于测试的目录（sudo mkdir -p /srv/ftp/upload），然后修改该目录的权限即可（sudo chown ftp:ftp /srv/ftp/upload、sudo chmod 777 /srv/ftp/upload）。

5.3.3　访问限制与安全加固

通过vsftpd的关键参数的配置，可以控制访问权限，强化安全策略，限制匿名操作等。

- local_enable=YES：允许本地系统用户通过用户名和密码登录FTP。设为NO可禁止系统用户登录，仅保留匿名访问。
- write_enable=YES：允许用户进行写入操作（上传文件、创建目录等）。若设为NO，用户将只能浏览文件。
- anon_other_write_enable=NO：限制匿名用户重命名或删除文件，是额外的写权限控制。
- pam_service_name=vsftpd：指定PAM认证服务名，一般保持默认。
- xferlog_enable=YES：启用日志记录用户上传和下载操作。
- xferlog_file=/var/log/vsftpd.log：指定日志文件路径。
- cmds_denied=DELE,RMD：禁用某些FTP命令，如删除文件或目录命令，可防止误删或恶意操作。
- userlist_enable=YES：启用用户白名单/黑名单功能。
- userlist_file=/etc/vsftpd.user_list：指定用户列表文件，结合“userlist_deny=NO”可允许仅此列表内的用户登录。
- ssl_enable=NO：未启用ssl，可以启用，并配置rsa_cert_file=/etc/ssl/certs/ftp-cert.pem、rsa_private_key_file=/etc/ssl/private/ftp-cert.key，并确保证书路径正确，证书由可信机构签发或为自签证书。

动手练　启用与配置SFTP


扫码看视频

SFTP（SSH File Transfer Protocol，安全文件传输协议）是一种基于SSH（Secure Shell）协议的文件传输方式，与传统FTP相比，SFTP在传输过程中对数据进行了加密，具备更高的安全性，因而在现代系统运维中被广泛采用。Ubuntu Server 默认已安装并启用了OpenSSH服务，因

而SFTP也随之可用，无须额外安装软件包。

1. 启用服务并访问 SFTP 服务器

在Ubuntu Server上，SFTP功能由sshd服务提供。只要SSH服务正常运行，SFTP功能就处于启用状态。可以通过sudo systemctl status ssh确认SSH服务是否正常，若服务正常，就可以使用“sftp 用户名@服务器IP地址”来访问：

```
C:\Users\YSY>sftp wlysy@192.168.80.88      # 访问SFTP服务器
Connected to 192.168.80.88.                # 正常连接
sftp> ls
123    234    test                         # 通过ls来查看当前目录中的文件
sftp> pwd                                  # 查看当前目录的路径
Remote working directory: /home/wlysy
```

知识点拨：其他操作命令

cd：切换远程目录。get 文件名：下载文件。put 文件名：上传文件。exit 或 bye：退出连接。

2. 配置 SFTP 专用用户并限制其访问目录

为了提升系统安全性，可以为某些用户配置“受限的SFTP环境”，使其无法通过SSH登录系统，仅能使用SFTP功能，并被限制在特定目录下活动。首先创建用户及目录，并设置目录的权限：

```
wlysy@testus:~$ sudo adduser sftpuser                    # 创建sftpuser用户
……                                                       # 开启创建交互模式
New password:                                            # 设置用户密码
Retype new password:                                     # 校验用户密码
passwd: password updated successfully                    # 创建成功
……                                                       # 设置用户其他信息
wlysy@testus:~$ sudo mkdir -p /data/sftpuser/upload      # 递归创建目录
wlysy@testus:~$ sudo chown root:root /data/sftpuser      # 设置目录所属
wlysy@testus:~$ sudo chmod 755 /data/sftpuser            # 设置目录权限
wlysy@testus:~$ sudo chown sftpuser:sftpuser /data/sftpuser/upload
```

修改SFTP的配置文件（sudo vim /etc/ssh/sshd_config），在配置文件最后添加：

```
Match User sftpuser                        # 匹配用户sftpuser
    ForceCommand internal-sftp             # 强制使用内置SFTP，而非登录Shell
    PasswordAuthentication yes             # 使用密码验证
    ChrootDirectory /data/sftpuser         # 限制该用户访问根目录
    PermitTunnel no                        # 禁止创建隧道（VPN）
    AllowAgentForwarding no                # 禁止SSH代理转发
    AllowTcpForwarding no                  # 禁止TCP端口转发
    X11Forwarding no                       # 禁止X11转发
```

接下来重启SSH服务，使配置生效：

```
wlysy@testus:~$ sudo systemctl restart ssh
```

最后使用“sftp sftpuser@192.168.80.88”和创建的密码访问SFTP服务器即可。

```
C:\Users\YSY>sftp sftpuser@192.168.80.88
sftpuser@192.168.80.88's password:
Connected to 192.168.80.88.
```

```
sftp> ls
upload
sftp> cd upload
sftp> ls
sftp> pwd
Remote working directory: /upload
```

5.4 rsync文件同步与备份

在实际运维场景中，文件的同步与备份是保障数据一致性与业务连续性的核心任务。rsync是一种高效、灵活且功能强大的文件同步工具，广泛应用于本地与远程之间的数据备份、目录镜像和系统迁移等场景。它的传输机制基于“仅复制差异部分”的增量算法，能大幅降低网络负载和磁盘I/O压力。

5.4.1　rsync命令及本地同步

rsync（Remote Sync）是一个支持本地与远程数据同步的命令行工具，默认已预装于大多数Linux发行版中。其核心优势在于：仅传输变更部分（增量传输）、支持保持文件属性与权限、支持压缩与加密传输、可通过SSH连接远程主机执行文件复制任务。

1. 基本语法

rsync的基本命令格式如下：

```
rsync [选项] 源路径 目标路径
```

其中，源路径和目标路径可以是本地目录，也可以是远程主机格式，例如user@host:/path。常见的选项及作用有：

- **-a**：归档模式，保留文件权限、时间戳、符号链接等信息。
- **-v**：显示详细信息。
- **-z**：在传输过程中启用压缩，提高效率。
- **--progress**：显示传输进度。
- **-r**：递归复制目录（默认-a已包含）。
- **--delete**：使目标目录删除源目录中已不存在的文件（慎用）。

2. 本地同步

将本地的/home/wlysy/123中的文件同步到/home/wlysy/234中：

```
wlysy@testus:~$ touch 123/aaa 123/bbb 123/ccc     # 在123目录中创建测试文件
wlysy@testus:~$ ls 123                            # 查看123目录中的文件
aaa  bbb  ccc
wlysy@testus:~$ ls 234                            # 查看234目录，为空
wlysy@testus:~$ rsync -av /home/wlysy/123/ /home/wlysy/234    # 执行同步
sending incremental file list                     # 同步的过程
./
aaa
bbb
ccc
sent 221 bytes  received 76 bytes  594.00 bytes/sec
total size is 0  speedup is 0.00
```

```
wlysy@testus:~$ ls 234                          # 再次查看
aaa  bbb  ccc                                   # 同步成功
```

如果需要显示详细的文件传输情况，可以带上“--progress”长选项。

注意事项 | 原路径末尾的“/”|

原路径最后的“/”代表同步目录内容，如果不加上“/”则代表连目录本身一起同步。

5.4.2 远程同步配置

除了本地同步外，rsync还可以在本地和远程主机之间进行同步，帮助管理员实现高效的数据传输与定向备份操作。下面介绍具体的操作。

1. 从本地主机同步到远程主机

将本地目录中的内容同步到远程主机中是非常常见的操作：

```
wlysy@myus:~$ mkdir test                        # 创建同步的测试目录
wlysy@myus:~$ touch test/yuancheng              # 在目录中创建测试文件
wlysy@myus:~$ rsync -avz /home/wlysy/test/ wlysy@192.168.80.88:/home/wlysy/
backup/        # 将本地的“test”目录中的内容同步到目标主机的用户主目录的backup目录中
The authenticity of host '192.168.80.88 (192.168.80.88)' can't be established.
ED25519 key fingerprint is SHA256:p+PPhegoah/w0MUK8h+1LIMIO4//v5B1vgWLAwmpzSI.
This key is not known by any other names.  # 首次连接的警告信息及远程主机公钥
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.80.88' (ED25519) to the list of known
hosts.         # SSH客户端确认信息，提示公钥已保存，下次不再提示，除非公钥发生变化
wlysy@192.168.80.88's password:     # 输入远程主机用户wlysy的密码进行身份认证
sending incremental file list       # 开始传输文件，并只传输变化或缺失的文件
./
Yuancheng                           # 识别并准备传输的文件
sent 120 bytes  received 38 bytes  28.73 bytes/sec       # 传输的统计信息
total size is 0  speedup is 0.00    # 最终的传输摘要信息
```

在目标主机中，就可以在相应的目录中查看到该文件了。

```
wlysy@testus:~$ ls backup/                      # 查看同步目录中的内容
yuancheng                                       # 已完成同步
```

2. 从远程主机同步到本地

在文件出现问题、误删除重要文件后，可以将远程主机的备份目录拉取到本地。执行效果如下：

```
wlysy@myus:~$ rm test/yuancheng                 # 删除本地的文件
wlysy@myus:~$ ls test/                          # 查看后，已完全删除
wlysy@myus:~$ rsync -avz wlysy@192.168.80.88:/home/wlysy/backup/ /home/wlysy/
test/                                           # 远程同步到本地命令
wlysy@192.168.80.88's password:                 # 密码验证
receiving incremental file list                 # 接收文件列表
./
Yuancheng                                       # 识别并准备传输的文件
sent 46 bytes  received 120 bytes  47.43 bytes/sec
total size is 0  speedup is 0.00
wlysy@myus:~$ ls test/                          # 传输完成后查看本地目录
yuancheng                                       # 删除的文件已经同步回来了
```

知识点拨：守护进程模式

rsync也支持daemon模式，即作为守护进程提供服务。此方式适合大规模文件同步，需配置/etc/rsyncd.conf文件，但对日常使用者来说，通过SSH方式更简单易用。

5.4.3　自动化同步

rsync不仅支持手动执行文件同步任务，更适合通过脚本与计划任务实现定时、无人值守的自动化同步操作。

1. 准备 SSH 免密登录环境

为避免每次同步时都输入密码，推荐先为rsync配置SSH免密登录：

```
wlysy@myus:~$ ssh-keygen -t rsa                  # 在本地生成密钥对
……                                               # 使用默认值，按回车键完成创建
wlysy@myus:~$ ssh-copy-id wlysy@192.168.80.88     # 将公钥复制到远程主机
……                                               # 等待完成即可
wlysy@myus:~$ ssh wlysy@192.168.80.88            # 使用SSH远程测试免密登录
```

2. 编写并执行自动化脚本

编写一个简单的rsync自动同步脚本rsync_backup.sh（touch rsync_backup.sh），用于每天将本地目录备份到远程目录。内容为：

```
#!/bin/bash
# 定义本地和远程目录
SRC="/home/wlysy/test/"
DEST="wlysy@192.168.80.88:/home/wlysy/backup/"
# 执行rsync同步操作，记录日志
rsync -az --delete "$SRC" "$DEST" >> ~/rsync_backup.log 2>&1
```

脚本使用-a和-z选项保留权限并启用压缩。--delete选项用于同步删除目标端已不存在的文件（需谨慎使用）。>> ~/rsync_backup.log 2>&1表示将执行日志追加写入用户日志文件中，便于排错。

接下来保存文件，将脚本变为可执行状态：

```
wlysy@myus:~$ vim rsync_backup.sh          # 之前的编辑操作
wlysy@myus:~$ ls
rsync_backup.sh  test                      # 保存后查看，创建成功
wlysy@myus:~$ chmod a+x rsync_backup.sh    # 为其添加可执行权限
```

首先测试，将目标服务器backup目录中的yuancheng文件删除。接着在客户端执行脚本：

```
wlysy@myus:~$ ./rsync_backup.sh            # 执行脚本
wlysy@myus:~$ ls                           # 执行后查看目录内容
rsync_backup.log  rsync_backup.sh  test    # 自动生成日志文件rsync_backup.log
```

在目标服务器上，可以看到已经完成了同步，说明脚本执行成功：

```
wlysy@testus:~$ ls backup/                 # 执行脚本前，backup为空
wlysy@testus:~$ ls backup/                 # 执行脚本后，再次查看
yuancheng                                  # 文件已同步
```

3. 配置周期性自动执行

使用crontab -e命令编辑当前用户的计划任务，添加自动执行任务：

```
wlysy@myus:~$ sudo crontab -e                # 编辑计划任务
no crontab for root - using an empty one
Select an editor.  To change later, run 'select-editor'.
  1. /bin/nano        <---- easiest
  2. /usr/bin/vim.basic
  3. /usr/bin/vim.tiny
  4. /bin/ed
Choose 1-4 [1]: 2                            # 选择编辑器，这里用VIM
crontab: installing new crontab              # 完成编辑
```

编辑的内容为：

```
0 2 * * * /home/wlysy/rsync_backup.sh
```

表示每天凌晨2点自动执行同步任务。你可以根据需求修改时间设置，例如每小时、每分钟等。

知识点拨：计划任务的格式

它由6个字段组成，前5个字段定义了任务执行的时间，第6个字段是实际要执行的命令或脚本路径。格式为：分 时 日 月 周 命令。分的取值范围为0～59；时的取值范围为0～23；日的取值范围为1～31；月的取值范围为1～12；周的取值范围为0～7（0和7都是周日）；*（星号）代表“每”或“任意”，表示该字段的所有可能值。

5.5 常用命令行文件传输技巧

在日常的Linux系统运维工作中，除了配置专门的文件服务（如FTP、Samba、NFS、rsync等）外，还常常需要临时或快速地在本地与远程主机之间传输文件。此时，灵活运用命令行工具可以大大提升工作效率，尤其在图形界面不可用或系统资源受限的环境中更加重要。

5.5.1 使用scp进行主机间快速传输

在Linux系统中，scp（secure copy）是一种基于SSH协议的远程文件传输命令，具备安全性高、使用简单的特点。借助scp，用户可以方便地在本地主机与远程主机之间，或两个远程主机之间复制文件或目录，且整个传输过程经过加密，不易被窃听或篡改。

1. scp 命令基本语法

scp命令的基本语法为：scp [参数] 源路径 目标路径。

其中路径格式通常为：

- **本地文件**：/home/user/file.txt。
- **远程文件**：user@ip:/remote/path/file.txt。

常见选项及参数说明：

- **-r**：递归复制整个目录。
- **-P**：指定SSH连接端口（注意大写）。
- **-p**：保留原文件的修改时间、访问权限等信息。

- -v：显示详细调试信息，常用于故障排查。
- -C：启用压缩传输，提高大文件传输效率。
- -i：指定SSH密钥文件进行认证（用于免密登录）。

2. scp 常见传输操作

以下通过几个实际命令示例展示scp的常见用法，因为命令在Windows中的cmd和PowerShell的用法与在Linux中相同，下面主要介绍命令的使用方法。

（1）将本地文件传输到远程主机

将本地的test.txt文件发送到远程主机192.168.80.88的用户wlysy主目录的backup目录下：

```
C:\Users\YSY>scp test.txt wlysy@192.168.80.88:/home/wlysy/backup/
test.txt                              100%    0     0.0KB/s   00:00        # 传输成功
```

知识点拨：Windows用户默认主目录

Windows用户默认主目录在“C:\Users\用户同名目录”中，用户上传的文件和下载的文件的默认路径都在这里。

（2）从远程主机下载文件到本地

使用另一台Linux系统，将刚才上传的文件test.txt下载到当前用户主目录中：

```
wlysy@myus:~$ scp wlysy@192.168.80.88:/home/wlysy/backup/test.txt /home/wlysy/
wlysy@myus:~$ ls
rsync_backup.log  rsync_backup.sh  test  test.txt                    # 下载成功
```

知识点拨：上传及下载目录

如果上传及下载的是目录，则需要带上选项“-r”，然后指定位置就可以了。如果远程目录下文件较多、体积较大，建议结合“-C”（压缩）选项提升传输效率。如需进一步自动化或支持断点续传等高级功能，可以考虑使用rsync。

5.5.2　使用wget与curl拉取远程文件

在无法使用SSH或FTP的环境中（如从Web服务器、公网下载资源），wget与curl是两款轻量级、简洁高效的命令行下载工具，经常出现在脚本调试与自动化任务中。

1. wget：非交互式 HTTP/HTTPS/FTP 下载工具

wget支持断点续传、递归下载和将下载任务加入后台运行，适合批量下载或服务器上无图形界面的场景。用法如下：

（1）下载单个文件

```
wget https://example.com/files/sample.tar.gz
```

（2）指定保存路径或重命名

```
wget -O myfile.tar.gz https://example.com/files/sample.tar.gz
```

（3）断点续传（断开后接着下载）

```
wget -c https://example.com/large.iso
```

2. curl：灵活的命令行传输工具

curl支持 HTTP、FTP、SFTP、SMTP等协议，可用于快速脚本调试或简短文件拉取。用法如下：

（1）下载文件

```
curl -O https://example.com/data.zip
```

（2）指定输出文件名

```
curl -o data.zip https://example.com/data.zip
```

（3）带进度条显示

```
curl -LO https://example.com/data.zip
```

其中-L表示跟随重定向，适用于URL重写或跳转场景。

（4）采用Basic Auth 下载（需用户名和密码）

```
curl -u user:password -O https://example.com/private/report.csv
```

密码明文出现在命令中，不推荐用于生产环境，可改为使用交互方式或读取环境变量。


扫码看视频

动手练 使用wget与curl工具下载网络资源

下面以实例的形式介绍这两款工具下载网络资源的操作。

（1）使用wget下载软件包

在图形界面中，可以方便地使用浏览器来下载HTTP/HTTPS/FTP的资源。在终端中，就需要使用wget等命令来下载资源了。下面以常见的软件包为例介绍下载的步骤。在获取了资源的链接后（如http://mirror.nju.edu.cn/7-zip/7z2500-src.tar.xz），可以直接使用wget进行下载：

```
wlysy@testus:~$ ls                                                    # 查看当前目录中的文件
123  234  backup  test
wlysy@testus:~$ wget http://mirror.nju.edu.cn/7-zip/7z2500-src.tar.xz  # 下载
……                                                                    # 连接并获取资源信息
7z2500-src.tar.xz             100%[=============================================
=====>]   1.46M  2.41MB/s    in 0.6s       # 启动下载，显示进度条和下载速度参数
2025-07-25 02:04:44 (2.41 MB/s) - ‘7z2500-src.tar.xz’ saved [1531036/1531036]
wlysy@testus:~$ ls
123  234  7z2500-src.tar.xz  backup  test                # 再次查看，已下载完毕
```

（2）使用curl下载文件

在获取了文件的下载地址后，可以使用curl下载文件：

```
wlysy@testus:~$ curl -O http://mirrors.aliyun.com/centos/RPM-GPG-KEY-CentOS-7
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100  1690  100  1690    0     0  15856      0 --:--:-- --:--:-- --:--:-- 15943
wlysy@testus:~$ ls
123  234  7z2500-src.tar.xz  backup  RPM-GPG-KEY-CentOS-7  test        # 下载成功
```

知识延伸：基于Python HTTP/Netcat实现简易传输

当两台主机之间无法使用SCP、FTP等标准服务时，可借助轻量级工具（如Python的内建HTTP模块或Netcat）实现快速、零配置的文件传输，非常适合临时文件交换或局域网数据共享。

1. 使用 Python 简易 HTTP 服务

Python内建HTTP模块无须额外安装，几乎在所有Linux系统中都可直接使用。

（1）服务端（发送文件）

在要传输文件的一方，进入目标目录，执行以下命令：

```
wlysy@testus:~$ sudo ufw allow 8000/tcp                          # UFW放行8000端口
wlysy@testus:~$ python3 -m http.server 8000                      # 使用python搭建服务
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ... # 等待客户端
192.168.80.1 - - [25/Jul/2025 02:46:56] "GET / HTTP/1.1" 200 -
192.168.80.1 - - [25/Jul/2025 02:46:58] code 404, message File not found
192.168.80.1 - - [25/Jul/2025 02:46:58] "GET /favicon.ico HTTP/1.1" 404 -
 192.168.80.1 - - [25/Jul/2025 02:47:08] "GET /7z2500-src.tar.xz HTTP/1.1" 200 -
                                                    # 客户端的操作会反映到这里
```

默认会监听8000端口并共享当前目录内容。

（2）客户端（下载文件）

在客户端，可以使用浏览器访问该主机IP地址:端口号，在打开的界面中下载，如图5-5所示。

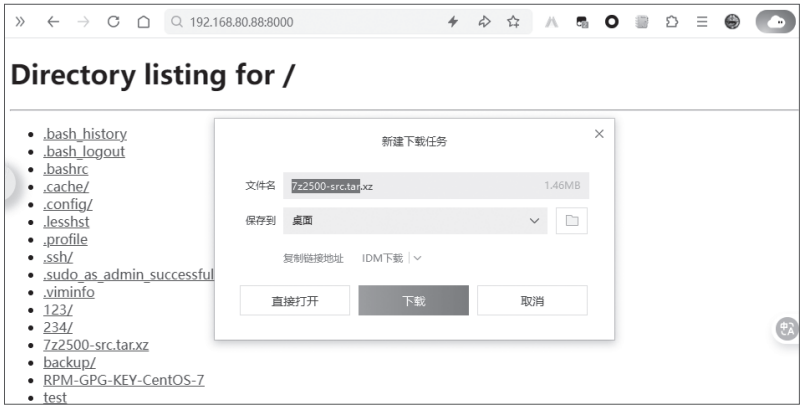


图 5-5

也可以使用wget命令下载：

```
wlysy@myus:~$ wget http://192.168.80.88:8000/7z2500-src.tar.xz
……                                                               # 获取资源信息
100%[========================>]   1.46MB  --.-KB/s    in 0.005s    # 启动下载
2025-07-25 02:59:34 (293 MB/s) - ‘7z2500-src.tar.xz’ saved [1531036/1531036]
wlysy@myus:~$ ls
7z2500-src.tar.xz  rsync_backup.log  rsync_backup.sh  test  test.txt   #下载成功
```

2. 使用 Netcat 实现点对点传输

Netcat是一种极简的“网络猫”，可用于监听端口、建立临时TCP连接进行数据传输。虽然功能强大，但需注意安全性。

（1）发送端（监听文件）

```
wlysy@testus:~$ sudo ufw allow 8888/tcp              # 将8888添加到UFW允许列表中
wlysy@testus:~$ nc -l -p 8888 < 7z2500-src.tar.xz    # 启动文件发送，进入侦听状态
```

（2）接收端

先测试到服务器的连接，并探测8888端口是否持续监听：

```
wlysy@myus:~$ nc -zv 192.168.80.88 8888
Connection to 192.168.80.88 8888 port [tcp/*] succeeded!        # 工作正常
wlysy@myus:~$ nc 192.168.80.88 8888 > 7z2500-src.tar.xz         # 使用命令下载文件
wlysy@myus:~$ ls
7z2500-src.tar.xz  rsync_backup.log  rsync_backup.sh  test.txt  # 下载成功
```

这里的发送端和接收端是相对而言的，两端可以互换角色，进行文件的传输。

注意事项 | 无法传输目录 |

Netcat (nc) 本身不支持目录传输，只能传输单个文件的数据流。如果确实需要传输目录，用户可以先将目录打包成一个文件，然后再通过Netcat传输。