

Linux服务器构建与运维管理

从基础到实战（基于 openEuler）

第8章：实现文件服务

阮晓龙

13938213680 / ruanxiaolong@hactcm.edu.cn

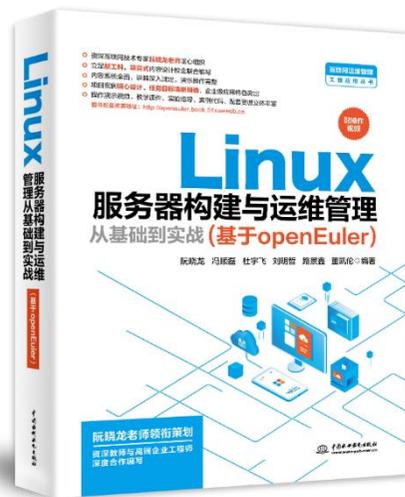
<https://internet.hactcm.edu.cn>
<http://www.51xueweb.cn>

河南中医药大学信息技术学院互联网技术教学团队
河南中医药大学医疗健康信息工程技术研究所

2024.10

提纲

- 理解文件共享服务
 - FTP
 - NFS
 - SMB/CIFS
- 实现文件共享服务器
 - 使用vsftpd实现FTP服务器
 - 实现NFS服务器
 - 实现Samba服务器
 - 使用Nextcloud实现云盘服务
- 文件传输协议与文件共享服务



1. 理解文件共享服务

- 文件共享是指主动地在网络上共享文件，实现对共享文件的写入或读取。
- 常见的文件共享服务有FTP、NFS、Samba、云盘等。

FTP

NFS

Samba

**云盘
网盘**



1. 理解文件共享服务

FTP

服务

协议

软件

命令



1. 理解文件共享服务

- FTP服务：
 - FTP服务是一种主机之间进行文件传输的服务，其重要特性是跨平台和精准授权。
 - FTP服务器是互联网上提供文件存储和访问服务的计算机，依照FTP协议提供服务。



1. 理解文件共享服务

□ FTP协议：

- FTP (File Transfer Protocol) 是文件传输协议，属于TCP/IP协议族的一部分，工作于OSI七层模型的应用层、表示层和会话层，控制端口号为TCP 21，数据通信端口号为TCP 20。
- FTP协议用于控制文件的双向传输，是Internet文件传送的基础，其目标是提高文件的共享性，提供非直接使用远程计算机，使存储介质对用户透明和可靠高效地传送数据。
- FTP协议支持跨路由的通信，能够面向互联网提供服务。



1. 理解文件共享服务

- FTP协议的传输模式有两类：
 - 是在Linux/UNIX系统中，FTP协议支持文本（ASCII）和二进制（Binary）两种方式的文件传输。
 - 在**文本传输模式**下，其传输方式会进行调整，主要体现为对不同操作系统的回车、换行、结束符等进行转译，将其自动文件转译成目的主机的文件格式。
 - 在**二进制传输模式**下，会严格保存文件的位序，原始文件和复制文件逐位一一对应，该传输方式不对文件做任何修改。
 - 选择合适的传输方式可以有效地避免文件乱码。



1. 理解文件共享服务

- FTP协议的工作模式分两种：
 - 模式1: Standard (也为PORT, 主动模式)
 - FTP的客户端发送PORT命令到FTP服务器进行端口确认。
 - 模式2: Passive (也为PASV, 被动模式)
 - FTP的客户端发送PASV命令到FTP服务器进行端口确认。



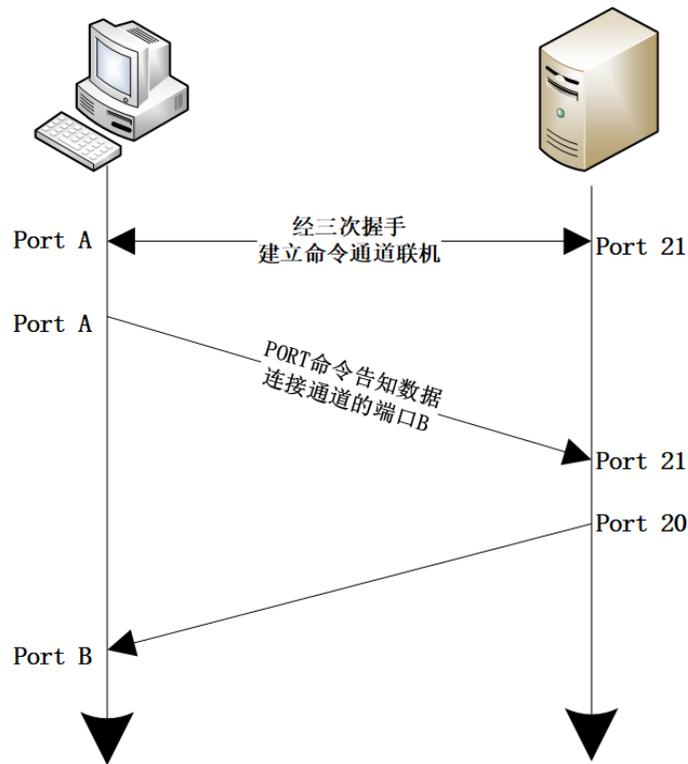
1. 理解文件共享服务

- FTP协议的工作模式分两种：
 - 模式1: Standard (也为PORT, 主动模式)
 - FTP客户端首先需要和FTP服务器的TCP 21端口建立连接, 通过这个通道客户端发送用户名和密码进行登录, 登录成功后要展示文件清单列表或者读取数据时, 客户端随机开放一个临时端口 (又名自由端口, 端口号在1024至65535之间), 发送PORT命令到FTP服务器, “告诉”服务器, 客户端采用主动模式并开放端口。
 - FTP服务器收到PORT主动模式命令和端口号后, 服务器的TCP 20端口和客户端开放的端口连接。
 - 在主动模式下, FTP服务器和客户端必须建立一个新的连接进行数据传输。



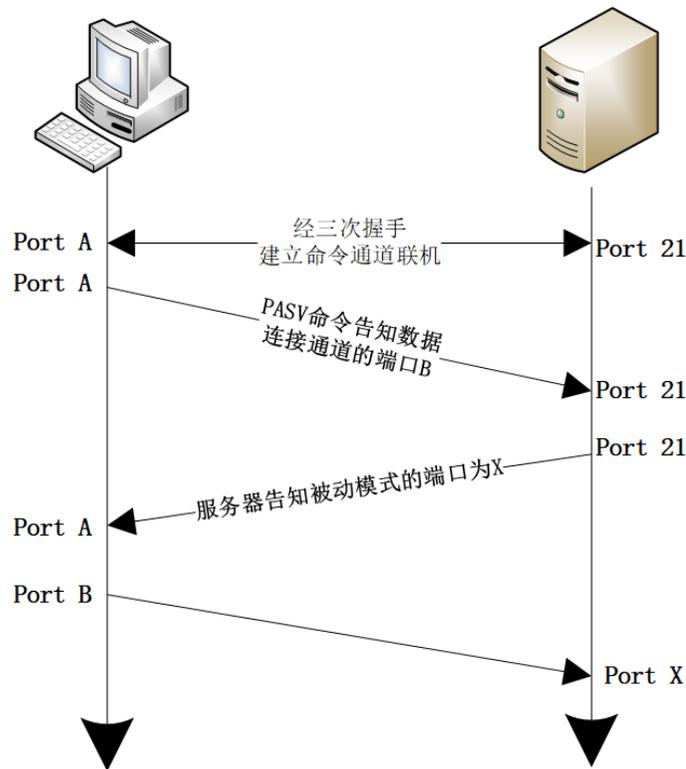
1. 理解文件共享服务

- FTP协议的工作模式分两种：
 - 模式1: Standard (也为PORT, 主动模式)
 - FTP客户端首先需要和FTP服务器的TCP 21端口建立连接, 通过这个通道客户端发送用户名和密码进行登录, 登录成功后要展示文件清单列表或者读取数据时, 客户端随机开放一个临时端口 (又名自由端口, 端口号在1024至65535之间), 发送PORT命令到FTP服务器, “告诉”服务器, 客户端采用主动模式并开放端口。
 - FTP服务器收到PORT主动模式命令和端口号后, 服务器的TCP 20端口和客户端开放的端口连接。
 - 在主动模式下, FTP服务器和客户端必须建立一个新的连接进行数据传输。



1. 理解文件共享服务

- FTP协议的工作模式分两种：
 - 模式2: Passive (也为PASV, 被动模式)
 - FTP客户端连接到FTP服务器的TCP 21端口, 发送用户名和密码进行登录, 登录成功后要展示文件清单列表或者读取数据时, 发送PASV命令到FTP服务器。
 - 服务器在本地随机开放一个临时端口, 然后把开放的端口告诉客户端, 客户端再连接到服务器开放的端口进行数据传输。
 - 在被动模式下, 不再需要建立一个新的FTP服务器和客户端的连接。



1. 理解文件共享服务

- FTP协议的工作模式分两种：
 - Standard模式与Passive模式的区别可概述为两个方面：
 - 主动模式传输数据是服务器连接到客户端的端口
 - 被动模式传输数据是客户端连接到服务器的端口

 - 主动模式需要客户端必须开放端口给服务器
 - 被动模式只需要服务器端开放端口给客户端连接即可

- 注意：
 - 很多客户端都是在防火墙内，开放端口给FTP服务器访问比较困难。
 - 通常服务端都是在防火墙内，开发数据传输端口也会很困难。



1. 理解文件共享服务

□ FTP软件

- FTP服务属于Client/Server (C/S) 结构，包含客户端和服务端软件两部分。

- FTP客户端软件：

- FileZilla Client
- FireFTP
- NcFTP



- FTP服务端软件：

- WU-FTPD
- ProFTPD
- FileZilla Server
- vsftpd



1. 理解文件共享服务

- FTP软件的服务端账号可分为实体用户、访客、匿名用户三种。
 - 实体用户 (Real User)
 - FTP服务器默认允许实体用户（即系统用户）的登录。
 - 以实体用户做为FTP服务器的身份登录时，系统默认不对实体用户进行任何限制，该用户可以针对整个文件系统进行自身权限的工作，即FTP服务器的管理员权限。
 - 访客身份 (Guest)
 - 在使用FTP服务器时，往往会给不同的部门或者某个特定的用户设置一个专属的帐户，创建一个访客身份就可满足。
 - 匿名身份 (Anonymous)
 - 匿名用户即不需通过账户密码就可登录访问FTP服务器资源的用户，这类用户在FTP服务器中没有确切的指定账户，但可以访问FTP服务器中开放的文件资源。
 - 需要FTP服务器端允许匿名用户访问。



1. 理解文件共享服务

□ FTP命令

■ FTP命令本质是就是FTP的CLI客户端软件。

命令详解: ftp

【语法】

ftp [选项] [参数]

【选项】

-d	启用调试, 显示所有客户端与服务端传递的命令
-v	禁止显示远程服务器相应信息
-n	禁止自动登录
-i	多文件传输过程中关闭交互提示
-g	禁用文件名通配符, 允许在本地文件和路径名中使用
-s	指定包含 FTP 命令的文本文件; 命令在 FTP 启动后自动运行。此参数中没有空格。可替代重定向符 (>) 使用
-a	在绑定数据连接时使用所有本地接口
-w	覆盖默认的传输缓冲区大小 65535

【参数】

主机 指定要连接的 FTP 服务器的主机名或 ip 地址

操作命令+配置文件+脚本程序+结束

【ftp 操作命令】

ftp 常用操作命令的选项及其说明如表 6-3 所示。

表 6-3 ftp 操作命令

命令	说明
! [shell [srg]]	在本地机中执行交互 shell, exit 回到 ftp 环境, 如: !ls*、zip
macro-ame[args]	执行宏定义 macro-name
account[password]	提供登录远程系统成功后访问系统资源所需的补充口令
append local-file[remote-file]	将本地文件追加到远程系统主机, 若未指定远程系统文件名, 则使用本地文件名
ascii	使用 ascii 类型传输方式
bell	每个命令执行完毕后计算机发出一声提示音
bin	使用二进制文件传输方式
bye	退出 ftp 会话过程
case	在使用 mget 时, 将远程主机文件名中的大写转为小写字母
cd remote-dir	进入远程主机目录
cdup	进入远程主机目录的父目录
chmod mode file-name	将远程主机文件 file-name 的存取方式设置为 mode, 如: chmod 777 a、out
close	中断与远程服务器的 ftp 会话 (与 open 对应)
cr	使用 ascii 方式传输文件时, 将回车换行转换为换行
delete remote-file	删除远程主机文件
debug[debug-value]	设置调试方式, 显示发送至远程主机的每条命令, 如: deb up 3, 若设为 0, 表示取消 debug
dir[remote-dir][local-file]	显示远程主机目录, 并将结果存入本地文件
form format	将文件传输方式设置为 format, 缺省为 file 方式
get remote-file[local-file]	将远程主机的文件 remote-file 传至本地硬盘的本地文件
glob	设置 mdelete, mget, mput 的文件名扩展, 缺省时不扩展文件名, 同命令行的 g 参数
hash	每传输 1024 字节, 显示一个 hash 符号 (#)
help[cmd]	显示 ftp 内部命令 cmd 的帮助信息, 如: help get

【ftp 操作命令】

ftp 常用操作命令的选项及其说明如表 6-3 所示。

表 6-3 ftp 操作命令

命令	说明
idle[seconds]	将远程服务器的休眠计时器设为[seconds]秒
image	设置二进制传输方式（同 bin）
lcd[dir]	将本地工作目录切换至 dir
ls[remote-dir][local-file]	显示远程目录 remote-dir，并存入本地文件 local-file
macdef macro-name	定义一个宏，遇到 macdef 下的空行时，宏定义结束
mdelete[remote-file]	删除远程主机文件
mdir remote-files local-file	与 dir 类似，但可指定多个远程文件，如：mdir *、o、*、zipoutfile
mget remote-files	传输多个远程文件
mkdir dir-name	在远程主机中建立一目录
mls remote-file local-file	同 nlist，但可指定多个文件名
mode[modename]	将文件传输方式设置为 modename，缺省为 stream 方式
modtime file-name	显示远程主机文件的最后修改时间
mput local-file	将多个文件传输至远程主机
newer file-name	如果远程机中 file-name 的修改时间比本地硬盘同名文件的时间更近，则重传该文件
nlist[remote-dir][local-file]	显示远程主机目录的文件清单，并存入本地硬盘的 local-file
nmap[inpattern outpattern]	设置文件名映射机制，使得文件传输时，文件中的某些字符相互转换，如：nmap \$1、\$2、\$3[\$1，\$2]、[\$2，\$3]，则传输文件 a1、a2、a3 时，文件名变为 a1、a2。该命令特别适用于远程主机为非 UNIX 机的情况
ntrans[inchars[outchars>]	设置文件名字符的翻译机制，如 ntrans1R，则文件名 LLL 将变为 RRR
open host[port]	建立指定 ftp 服务器连接，可指定连接端口
passive	进入被动传输方式
prompt	设置多个文件传输时的交互提示
proxy ftp-command	在次要控制连接中，执行一条 ftp 命令，该命令允许连接两个 ftp 服务器，以在两个服务器间传输文件。第一条 ftp 命令必须为 open，以首先建立两个服务器间的连接

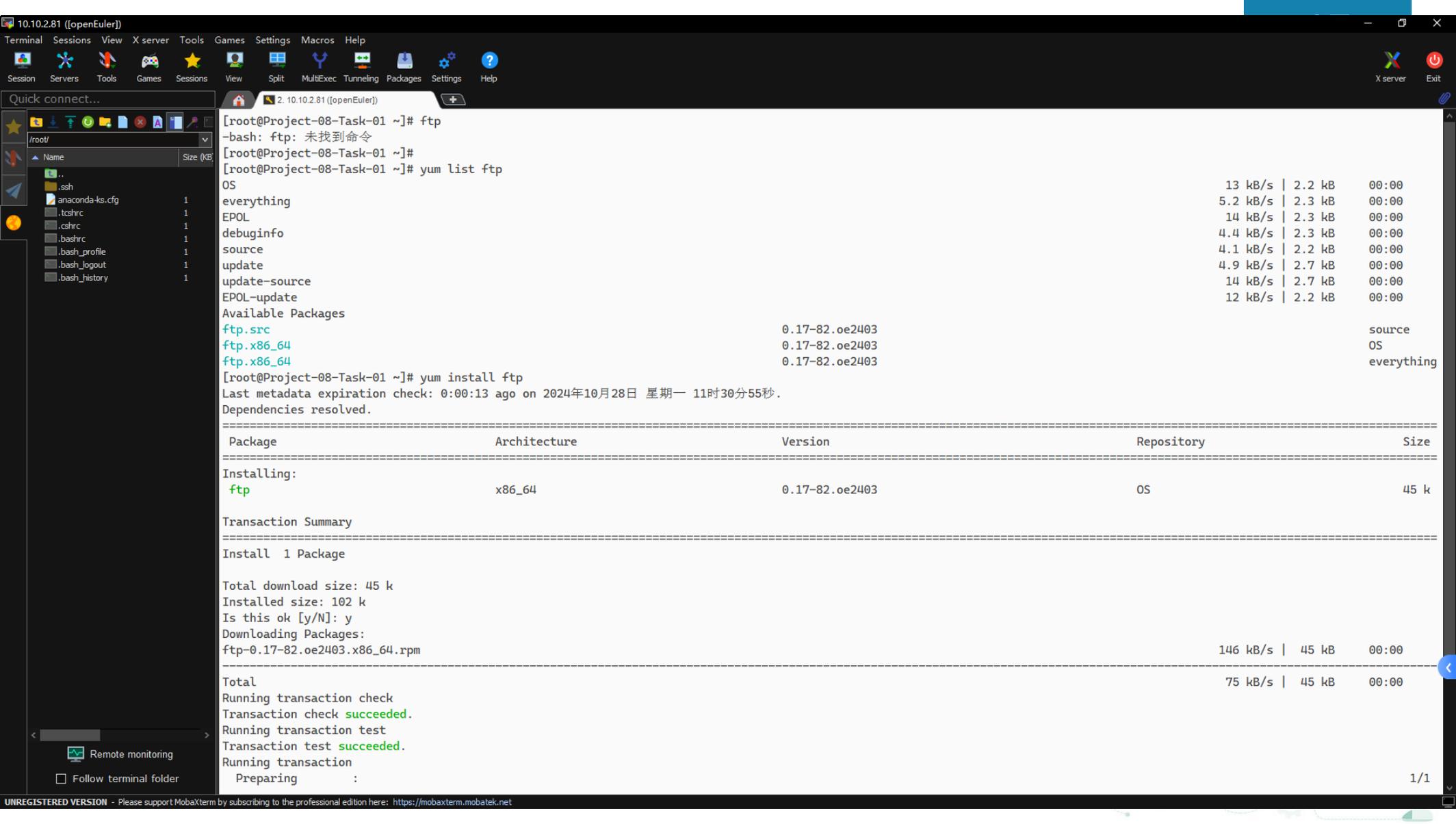
【ftp 操作命令】

ftp 常用操作命令的选项及其说明如表 6-3 所示。

表 6-3 ftp 操作命令

命令	说明
put local-file[remote-file]	将本地文件 local-file 传送到远程主机
pwd	显示远程主机的当前工作目录
quote arg1, arg2,...	将参数逐字发送到远程 ftp 服务器，如：quote syst
reget remote-file[local-file]	类似于 get，但若 local-file 存在，则从上次传输中断处续传
rhelp[cmd-name]	请求获得远程主机的帮助
rstatus[file-name]	若未指定文件名，则显示远程主机的状态，否则显示文件状态
rename[from][to]	更改远程主机文件名
reset	清除回答队列
restart marker	从指定的标志 marker 处，重新开始 get 或 put，如：restart 130
rmdir dir-name	删除远程主机目录
runique	设置文件名只存性存储，若文件存在，则在原文件后加后缀、1、2 等
sendport	设置 PORT 命令的使用
site arg1, arg2,...	将参数作为 SITE 命令逐字发送到远程 ftp 主机
size file-name	显示远程主机文件大小，如：site idle 7200
status	显示当前 ftp 状态
struct[struct-name]	将文件传输结构设置为 struct-name，缺省时使用 stream 结构
sunique	将远程主机文件名存储设置为只一（与 runique 对应）
system	显示远程主机的操作系统类型
tenex	将文件传输类型设置为 TENEX 机的所需的类型
tick	设置传输时的字节计数器
trace	设置包跟踪
type[type-name]	设置文件传输类型为 type-name，缺省为 ascii，如 type binary，设置二进制传输方式
umask[newmask]	将远程服务器的缺省 umask 设置为 newmask，如：umask 3
user user-name[password][account]	向远程主机表明自己的身份，需要口令时，必须输入口令，如：user anonymous my@email





1. 理解文件共享服务

- NFS (Network File System) 即网络文件系统，是由Sun公司于1985年推出的协议，大部分的Linux发行版均支持NFS。
 - NFS允许网络中的计算机通过TCP/IP网络共享资源，其主要功能是通过网络使不同操作系统之间可以彼此**共享文件和目录**。
 - NFS服务器允许NFS客户端将远端NFS服务器端的共享目录**挂载**到本地。
 - NFS客户端本地的机器看来，NFS服务器端共享的目录就如同外挂的磁盘分区和目录一样，也就是说客户端可以透明地访问服务器中的文件系统。

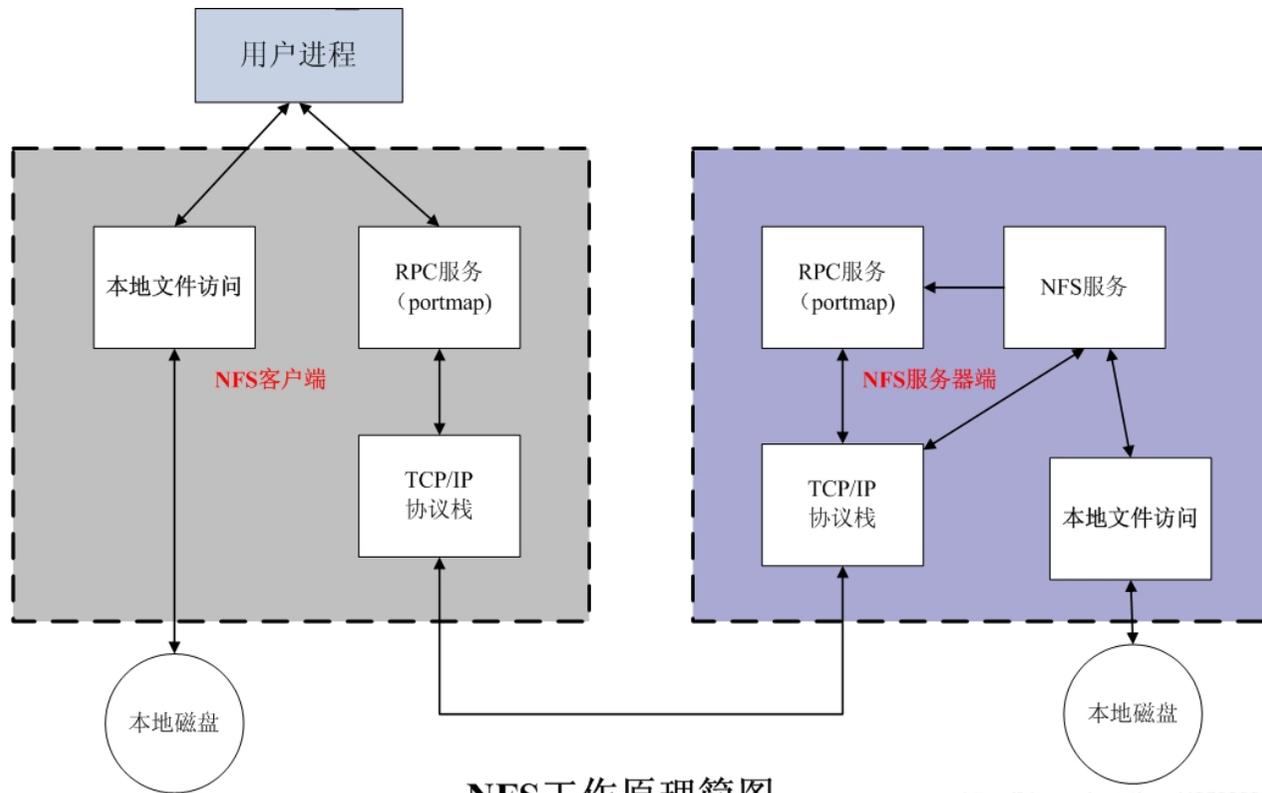


1. 理解文件共享服务

- 由于NFS支持功能很多，例如不同文件对不同用户开放不同权限，不同的功能会启动不同的端口来传输数据等。
- 端口不固定会造成NFS客户端与NFS服务器端的通信障碍，就需要调用RPC服务来进行规划协调。
 - RPC (Remote Procedure Call Protocol) 即远程过程调用协议，属于网络文件系统的核心，也是NFS服务器工作的重要支持。
 - RPC相当于NFS客户端与NFS服务器端数据传输的桥梁。
 - RPC最主要的功能就是指定每个NFS功能所对应的端口号，并且汇报给客户端，让客户端可以连接到正确的端口上进行通讯。
 - 当服务器在启动NFS时会随机选用某个端口，并主动地向RPC注册。
 - RPC则使用固定端口111来监听客户端的请求并返回客户端正确的端口，这样RPC就可以知道每个端口对应的NFS功能。



1. 理解文件共享服务



NFS工作原理简图

https://blog.csdn.net/qq_41959899



1. 理解文件共享服务

- NFS必须要在RPC存在时才能提供服务。
 - 启动NFS之前，必须先启动RPC，否则NFS会无法向RPC注册。
 - 重新启动RPC时，之前注册的端口与功能的数据将会消失。
 - 重新启动RPC，需要将其管理的所有程序都重新启动，重新进行RPC注册。
 - NFS的各项功能都必须向RPC注册，这样RPC才能了解NFS服务的各项功能的port number、PID和NFS在主机所监听的IP等，客户端才能够通过RPC的询问找到正确对应的端口。

- NFS为RPC Server的一种。



1. 理解文件共享服务

□ NFS (Network File System) 运行使用的多个端口

■ 端口111 (TCP/UDP):

- RPC (Remote Procedure Call) 端口映射器 (Portmapper) 的服务端口。
- 所有NFS相关的服务都会通过Portmapper注册它们所监听的实际端口号。
- 当客户端需要与服务器进行通信时，首先会联系Portmapper以获取到NFS及其他相关服务实际使用的动态或固定端口号。

■ 端口 2049 (TCP/UDP):

- NFS服务的标准文件系统服务端口，用于传输文件数据和元数据。
- NFS v3及以前版本通常只使用这个端口进行文件操作。

■ 其他端口:

- NFS通过Portmapper进行端口管理和服务发现，并固定使用111和2049两个端口。
- 其他关联服务可能使用非固定的动态端口。
 - mountd: 负责处理文件系统的挂载请求，端口号通常是随机分配，但也可以配置为固定端口。
 - nlockmgr: 管理网络锁定，确保客户端对共享文件并发访问时的一致性，监听由portmapper分配的端口。
 - rquotad: 提供远程磁盘配额服务，对需要配额管理的NFS共享而言是必需的，监听由portmapper分配的端口。



1. 理解文件共享服务

□ NFS的认证机制

- 在默认情况下并不强制要求进行用户账号的认证，但可以根据配置实现一定程度的安全性控制和身份验证。
- NFSv3及更早版本中，采用了匿名映射或“信任”模式。
 - 基于/etc/exports文件设置来决定哪些客户端可以挂载共享以及使用什么权限。
 - 检查请求的来源IP地址。
- 在安全性要求更高的场景中，NFS可以通过多种方式增强安全措施：
 - root Squashing: 所有远程root用户映射为一个非特权用户。
 - 身份映射服务：
 - NIS (Network Information Service) 或者后来的 NIS+ 可以用于集中式账户管理和身份验证。
 - LDAP作为身份验证后端，使NFS能够与用户数据库实现认证同步。
 - 在NFSv4中引入了对Kerberos身份验证协议的支持。



1. 理解文件共享服务

□ NFS的优点

- 跨平台共享：不同操作系统之间能够无缝共享文件和目录。
- 集中管理：数据存储集中化，便于管理和备份。
- 灵活访问：用户可以从任何授权的系统上透明地读写远程文件。
- 性能优化：支持缓存机制以提高文件访问速度，减少网络传输开销。
- 可扩展性好：可以根据需求增加服务器来扩展存储资源。
- 易用性强：配置简单，使用标准命令即可挂载和操作远程文件系统。



1. 理解文件共享服务

□ NFS的使用场景

■ 数据共享

■ 虚拟化与云计算：

- 虚拟机（VMs）通过宿主机挂载的NFS共享来存储和读取其操作系统镜像、应用程序数据及用户数据，简化管理和维护。

■ 容器编排

- 在Kubernetes或其他容器平台中，常用于持久卷提供动态存储。

■ 集群计算

- HPC集群节点可以通过NFS共享大型数据集，支持并行计算任务的数据访问需求。
- 大数据处理环境中，依赖于NFS共享目录，使得不同节点能够存取统一的数据源。

■ 备份与恢复

- NFS广泛应用于需要跨网络透明共享文件和目录的场合，尤其适用于那些强调数据一致性、资源共享和灵活扩展性的分布式系统架构。



1. 理解文件共享服务

- SMB (Server Message Block) 协议
 - 是微软和英特尔在1987年制定的协议，主要是作为Microsoft网络的通讯协议。
 - 是微软为了让Windows/MS-DOS操作系统间可以实现资源共享，提出的一个不同于NFS的协议SMB（全称为Server Message Block），实现Windows/MS-DOS间能够共享网络中的文件系统、打印机等资源。
 - 是微软为Windows操作系统需求设计的专用协议，用来实现微软主机之间的文件共享与打印共享，**并不支持在Linux上运行**。
- SMB协议的端口和发展
 - 在Windows 98、Windows Me、Windows NT操作系统中，SMB服务使用UDP 137、UDP 138、TCP 139端口。
 - 在Windows 2000以后版本的操作系统中使用TCP 445端口。
 - 后来改名为CIFS (Common Internet File System)，加入了许多新功能。



1. 理解文件共享服务

□ CIFS协议：

- CIFS (Common Internet File System) 是实现文件共享服务的一种文件系统，主要用于实现windows系统中的文件共享，使程序可以访问远程Internet计算机上的文件并要求此计算机提供服务。
- CIFS使用客户端/服务器模式。
- CIFS在高层运行，属于应用程序协议。
- Microsoft将SMB协议扩展到Internet上去，成为Internet上计算机之间相互共享数据的一种标准。将SMB协议的技术文档进行整理，重新命名为CIFS (Common Internet File System) ，与NetBIOS相脱离，成为Internet上的标准协议。
- 推荐阅读：<https://www.jianshu.com/p/8b702331ca2a>



1. 理解文件共享服务

- SMB/CIFS协议的主要用途是文件共享，也有一些扩展的功能
 - 协议协商
 - 搜索网络上的其他 Microsoft SMB 协议服务器或网络浏览
 - 网络打印
 - 文件、目录和共享访问身份验证
 - 文件和记录锁定
 - 文件和目录更改通知
 - 扩展文件属性处理
 - Unicode 支持
 - 机会锁



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器

- vsftpd is a GPL licensed FTP server for UNIX systems, including Linux.
 - It is **secure** and extremely **fast**. (安全、极其快速)
 - It is **stable**. (稳定)
- Features
 - Virtual IP configurations
 - Virtual users
 - Standalone or inetd operation (inetd: internet service daemon)
 - Powerful per-user configurability (基于单用户的配置)
 - Bandwidth throttling (流量限制、带宽管理)
 - Per-source-IP configurability (基于源IP的配置)
 - Per-source-IP limits (基于源IP的限制)
 - IPv6
 - Encryption support through SSL integration (支持SSL进行通信加密)
 - etc...



vsftpd

Probably the most secure and fastest FTP server for UNIX-like systems.

Main index

- [About vsftpd](#)
- [Features](#)
- [Online source / docs](#)
- [Download vsftpd](#)
- [Who recommends vsftpd](#)
- [vsftpd security](#)
- [vsftpd performance](#)

News

Other links you may be looking for

- Project Zero, probably the best technical security blog around: [Project Zero blog](#)
- Follow me on Twitter for vsftpd / security news: [scarybeasts](#)
- My security blog: <http://scarybeastsecurity.blogspot.com/>
- My security advisories: <https://security.appspot.com/security/index.html>

Aug 2021 - vsftpd-3.0.4 / vsftpd-3.0.5 released with build, seccomp and SSL modernizations

- vsftpd-3.0.5 fixes the new ALPN selection, so it works again with the latest FileZilla client.
- vsftpd-3.0.4 is released, 6 years after the previous release! This now builds and runs again on a modern system such as Fedora 33 -- a few things had broken over the years. A few SSL modernizations have been applied, such as requiring TLSv1.2+ by default, supporting ALPN, and optionally supporting an SNI check. See the [Changelog](#) and [vsftpd FAQ](#) (frequently asked questions) for a list of common questions!
- This release is signed with my new RSA4096 scarybeasts@gmail.com GPG key (67A2 AB4F 41F9 972C 21F6 BF66 7B89 011B CAE1 CFEA): [public key file](#)
- The release is also signed with my old chris@scarybeasts.org key for a cross check: [release signature with old key](#)
- Here's a signature for my new GPG key, signed by my old key: [signature for new public key, signed by old key](#)

Jul 2015 - vsftpd-3.0.3 released with SSL fixes and security improvements

- vsftpd-3.0.3 is released - with most of the changes being SSL related. Other than that, there some seccomp policy fixes and minor compatability fixes. Some notes on the SSL fixes will be put on my blog shortly. See the [Changelog](#) and [vsftpd FAQ](#) (frequently asked questions) for a list of common questions!

Sep 2012 - vsftpd-3.0.2 released with seccomp sandbox fixes

- vsftpd-3.0.2 is released - the only noteworthy fixes are two seccomp sandbox policy tweaks which stops session crashes when listing large directories. See the [Changelog](#) and [vsftpd FAQ](#) (frequently asked questions) for a list of common questions!

Apr 2012 - vsftpd-3.0.0 released with a seccomp filter sandbox

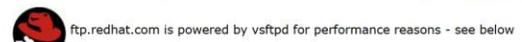
- vsftpd-3.0.0 is released - with a new highly restrictive seccomp filter sandbox. It activates automatically on 64-bit binaries on Ubuntu 12.04+. In addition, there's a fix for passive mode connections under high loads and a few timeout fixes, particularly if you're using SSL. See the [Changelog](#) and [vsftpd FAQ](#) (frequently asked questions) for a list of common questions!

Dec 2011 - vsftpd-2.3.5 released

- vsftpd-2.3.5 is released - with a fix for active mode connection error handling and a workaround for a glibc vulnerability that may affect unusual configurations. See the [Changelog](#) and [vsftpd FAQ](#) (frequently asked questions) for a list of common questions!
- Older:
- After numerous requests, I now have a PayPal button for donations. If you use vsftpd, like it, and think it's worthy of a donation, then click on the Paypal button on the left of the page.
- ftp.freebsd.org switched to vsftpd.
- vsftpd tarballs are now GPG signed by me (8660 FD32 91B1 84CD BC2F 6418 AA62 EC46 3C0E 751C)

Nov 2011 - Is any server other than vsftpd safe?

- ProFTPD [suffers serious security hole](#) - Nov 2011
- ProFTPD [suffers serious security hole](#) - Sep 2003
- wu-ftp [suffers serious security hole](#) - Jul 2003.
- lukemftpd (as a random example from many), via trust of realpath(), [suffers serious security hole](#) - Aug 2003.



About vsftpd

vsftpd is a GPL licensed FTP server for UNIX systems, including Linux. It is secure and extremely fast. It is stable. Don't take my word for it, though. Below, we will see evidence supporting all three assertions. We will also see a list of a few important sites which are happily using vsftpd. This demonstrates vsftpd is a mature and trusted solution.

Features

Despite being small for purposes of speed and security, many more complicated FTP setups are achievable with vsftpd! By no means an exclusive list, vsftpd will handle:

- Virtual IP configurations
- Virtual users
- Standalone or inetd operation
- Powerful per-user configurability
- Bandwidth throttling



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器

表 6-4 vsftpd 软件的常用目录文件及其说明

文件	说明
/usr/sbin/vsftpd	vsftpd 软件的主程序
/etc/vsftpd	vsftpd 软件的主目录
/etc/vsftpd/vsftpd.conf	vsftpd 软件的主配置文件
/etc/pam.d/vsftpd	基于 PAM 的 vsftpd 软件的验证配置文件
/etc/rc.d/init.d/vsftpd	vsftpd 软件的启动脚本，也可以使用 service 进行调用启动
/usr/share/doc/vsftpd-x.x.x	vsftpd 软件的文档资料路径
/vsr/ftp	默认的 vsftpd 软件的共享目录
/etc/vsftpd/ftppusers	默认的 vsftpd 软件的黑名单
/etc/vsftpd/user_list	修改某文件为黑名单或白名单的配置文件
/etc/logrotate.d/vsftpd	日志轮转备份配置文件



表 7-1-1 vsftpd.conf 配置项说明

配置项	说明
anonymous_enable=NO	是否允许匿名访问 FTP
local_enable=YES	是否允许本地用户登录
write_enable=YES	是否开启写命令
local_umask=022	本地用户的默认 umask 为 022
anon_upload_enable=YES	是否允许匿名上传
anon_mkdir_write_enable=YES	是否允许匿名创建目录
dirmessage_enable=YES	是否允许进入某个目录
xferlog_enable=YES	是否启用上载/下载的日志记录
connect_from_port_20=YES	是否限制传输连接来自端口 20
chown_uploads=YES	是否允许改变上传文件的属主
chown_username=whoever	设置想要改变的上传文件的属主，whoever 表示任何人
xferlog_file=/var/log/xferlog	设置上传和下载的日志文件
xferlog_std_format=YES	是否以标准 xferlog 的格式记录日志文件
idle_session_timeout=600	设置数据传输中断间隔时间
data_connection_timeout=120	设置数据连接超时时间
async_abor_enable=YES	是否识别异步 abort 请求



表 7-1-1 vsftpd.conf 配置项说明

配置项	说明
ascii_upload_enable=YES	是否以 ASCII 方式上传数据
ascii_download_enable=YES	是否以 ASCII 方式下载数据
ftpd_banner=Welcome to blah FTP service	登录 FTP 服务器时显示的欢迎信息
deny_email_enable=YES	是否开启 Email 黑名单
banned_email_file=/etc/vsftpd/banned_emails	设置 Email 黑名单文件
chroot_local_user=YES	是否限制所有用户在其主目录
chroot_list_enable=YES	是否限制启动限制用户名单
chroot_list_file=/etc/vsftpd/chroot_list	设置限制在主目录的用户名单文件
ls_recurse_enable=YES	是否允许客户端递归查询目录
listen=NO	是否允许 vsftpd 服务监听 IPv4 端口
listen_ipv6=YES	是否允许 vsftpd 服务监听 IPv6 端口
pam_service_name=vsftpd	设置 PAM 外挂模块提供的认证服务所使用的配置文件名, 即 /etc/pam.d/vsftpd 文件
userlist_enable=YES	是否禁止 user_list 文件中的用户列表登录 FTP 服务



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器

□ 场景设定

- 某研发型企业为了实现文件资源的共享，需构建一台企业内部的FTP服务器。
 - 共4个部门：行政部（2人）、市场部（3人）、设计部（2人）和开发部（3人）。
 - 独立账号访问，默认目录为部门目录。
 - 所有账号仅能够访问本部门目录，且具有读写权限。
 - 禁止匿名账号访问。

■ 需求分析

- 为每个部门创建目录与账号。
- 通过vsftpd实现文件共享服务。
- 使用PAM进行账号管理。
- 支持Linux、Windows等多终端、多操作系统。



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器

□ 方案设计

- 通过vsftpd实现FTP文件共享服务，通过虚拟用户进行权限管理。

表 8-1-1 部门用户列表

序号	部门	虚拟用户	虚拟用户密码
1	行政部	admin01	admin01@pwd
2		admin02	admin02@pwd
3	市场部	market01	market01@pwd
4		market02	market02@pwd
5		market03	market03@pwd
6	设计部	design01	design01@pwd
7		design02	design02@pwd
8	开发部	develop01	develop01@pwd
9		develop02	develop02@pwd
10		develop03	develop03@pwd



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器

□ 方案设计

- 通过vsftpd实现FTP文件共享服务，通过虚拟用户进行权限管理。

表 8-1-2 共享目录读写权限对应关系

序号	账号	/srv/ftp/admin	/srv/ftp/market	/srv/ftp/design	/srv/ftp/develop
1	admin01	○			
2	admin02	○			
3	market01		○		
4	market02		○		
5	market03		○		
6	design01			○	
7	design02			○	
8	develop01				○
9	develop02				○
10	develop03				○



2. 实现文件共享服务器

2.1 使用vsftpd实现FTP文件服务器



实现FTP文件服务

任务目标:

- 在openEuler上安装vsftpd软件，实现FTP服务，满足场景需求。

操作演示:



操作步骤:

- 实现在线安装vsftpd
- 实现vsftpd服务管理
- 实现通过vsftpd发布匿名访问的FTP服务
- 实现FTP服务的规划设计
- 实现企业内部的FTP服务
- 测试与交付



2. 实现文件共享服务器

□ NFS-Utils

- The NFS Utilities package contains the userspace server and client tools necessary to use the kernel's NFS abilities.
- NFS-Utils的开发基于LFS-10.1 (Linux From Scratch (System V Edition)) .

□ NFS Utilities 依赖的软件

- 必须: libtirpc-1.3.1 rpcsvc-proto-1.4.2
- 运行时必须: rpcbind-1.2.5
- 根据功能需求可选用:
 - Cyrus SASL-2.1.27 (for SASL authentication),
 - LVM2-2.03.11 (libdevmapper for NFSv4 support),
 - libnsl-1.3.0 (for NIS client support),
 - OpenLDAP-2.4.58 (for LDAP authentication),
 - librpcsecgss (for GSS and RPC security support)
 - SQLite-3.35.4, MIT Kerberos V5-1.19.1, libgssapi , libcap-2.49 with PAM



NFS-Utills-2.5.2

Introduction to NFS Utilities

The NFS Utilities package contains the userspace server and client tools necessary to use the kernel's NFS abilities. NFS is a protocol that allows sharing file systems over the network.

This package is known to build and work properly using an LFS-10.1 platform.

Package Information

- Download (HTTP): <https://www.kernel.org/pub/linux/utills/nfs-utils/2.5.2/nfs-utils-2.5.2.tar.xz>
- Download MD5 sum: 8c2a746325cb620f55751563356b0804
- Download size: 684 KB
- Estimated disk space required: 20 MB
- Estimated build time: 0.2 SBU

NFS Utilities Dependencies

Required

[libtirpc-1.3.1](#) and [rpcsvc-proto-1.4.2](#)

Optional

[Cyrus SASL-2.1.27](#) (for SASL authentication), [LVM2-2.03.11](#) (libdevmapper for NFSv4 support), [libnl-1.3.0](#) (for NIS client support), [OpenLDAP-2.4.58](#) (for LDAP authentication), [SQLite-3.35.4](#), [MIT Kerberos V5-1.19.1](#) or [libgssapi](#), and [librpcsecgss](#) (for GSS and RPC security support), and [libcap-2.49 with PAM](#)

Required (runtime)

[rpcbind-1.2.5](#)

User Notes: <http://wiki.linuxfromscratch.org/blfs/wiki/nfs-utils>

Kernel Configuration

Enable the following options in the kernel configuration (choose client and/or server support as appropriate) and recompile the kernel if necessary:

```
File systems --->
[*] Network File Systems --->
  <*/M NFS client support [CONFIG_NETWORK_FILESYSTEMS]
  <*/M NFS server support [CONFIG_NFS_FS]
  <*/M NFS server support [CONFIG_NFSD]
```

Select the appropriate sub-options that appear when the above options are selected.

Note

In BLFS we assume that nfs v3 will be used. If the *server* offers nfs v4 (for linux, CONFIG_NFSD_V4) then auto-negotiation for v3 will fail and you will need to add `nfsver=3` to the mount options. This also applies if that option is enabled in the *client's* kernel, for example in a distro trying to mount from a BLFS v3 server.

Even if neither end of the connection supports nfs v4, adding `nfsver=3` is still beneficial because it prevents an error message "NFS: bad mount option value specified: minorversion=1" being logged on every mount.

Installation of NFS Utilities

2. 实现文件共享服务器

2.2 实现NFS文件服务器

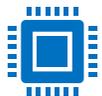
□ NFS的管理工具

- `exportfs` maintains a list of NFS exported file systems
- `nfscnf`
 - can be used to test for and retrieve configuration settings from a range of nfs-utils configuration files
- `nfsdcnts` prints information about NFS clients
- `nfsiostat` reports input/output statistics for network filesystems
- `nfsstat` displays statistics kept about NFS client and server activity
- `rpc.mountd` implements the NFS mount protocol on an NFS server
- `rpc.nfsd` implements the user level part of the NFS service on the server
- `rpc.statd` is used by the NFS file locking service
- `rpcdebug` sets or clears the kernel's NFS client and server debug flags
- `sm-notify` is used to send Network Status Monitor reboot messages
- `start-statd`
 - is a script called by `nfsmount` when mounting a filesystem with locking enabled, if `statd` does not appear to be running.
 - It can be customised with whatever flags are appropriate for the site



2. 实现文件共享服务器

2.2 实现NFS文件服务器



exportfs [选项] [参数]

功能:

- 管理NFS服务器共享的文件系统。

参数/命令:

- 共享文件系统:
 - 指定要通过NFS服务器共享的目录。
 - 格式为“/home/directory”

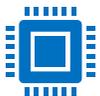
主要选项:

- a: 导出或卸载所有目录
- d: 开启调试功能
- o: 指定导出选项(如rw, async, root_squash)
- i:
 - 忽略/etc/exports和/etc/exports.d目录下的文件
- r: 更新共享的目录
- s: 显示当前可导出的目录列表
- v: 显示共享目录



2. 实现文件共享服务器

2.2 实现NFS文件服务器



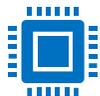
nfsstat [选项]

功能:

- 查看NFS客户端和服务器的访问与运行情况。

说明:

- **【选项】**
 - -s: 仅显示服务器端的状态信息
 - -c: 仅显示客户端的状态信息
 - -n: 仅显示NFS状态信息
 - -2/3/4: 仅列出NFS版本2/3/4的状态
 - -m: 显示已加载的NFS文件系统状态
 - -r: 仅显示rpc状态
 - -o: 显示自定义的设备信息
 - -l: 以列表的形式显示信息



showmount [选项]

功能:

- 显示NFS服务器共享资源的访问信息。

说明:

- **【选项】**
 - -a:
 - 以host:dir格式来显示客户主机名和挂载点目录
 - -d: 仅显示被客户挂载的目录名
 - -e: 显示NFS服务器的输出清单
 - -h: 显示帮助信息
 - -v: 显示版本信息
 - --no-headers: 不输出描述头部信息



表 8-2-3 NFS 配置文件参数及说明表

参数	说明
rw (read-write)	对共享目录具有读写权限
ro (read-only)	对共享目录具有只读权限
sync	同步写入，数据写入内存的同时写入磁盘
async	异步写入，数据先写入内存，周期性地写入磁盘
root_squash	将 root 用户及所属组映射为匿名用户或用户组（默认设置）
no_root_squash	与 root_squash 参数功能相反
all_squash	将远程访问的所有普通用户及所属组映射为匿名用户或用户组
no_all_squash	与 all_squash 参数功能相反（默认设置）
anonuid	将远程访问的所有用户均映射为匿名用户，并指定该用户的本地用户 UID
anongid	将远程访问的所有用户组均映射为匿名用户组，并指定该匿名用户组的本地用户组 GID
secure	限制客户端只能从小于 1024 的 TCP/IP 端口连接 NFS 服务器（默认设置）
insecure	允许客户端从大于 1024 的 TCP/IP 端口连接服务器
subtree_check	若输出目录是子目录，NFS 服务器检查其父目录的权限
no_subtree_check	若输出目录是子目录，NFS 服务器不检查其父目录的权限



2. 实现文件共享服务器

2.2 实现NFS文件服务器

□ 场景设定

- 某设计工作室拥有大量的数字资源，若存储在本地会占用主机大量存储，且不利于资源共享。现需要构建公共网络存储，实现灵活的资源读取和共享。
- 需求分析
 - 建设大容量、高可靠的网络共享存储服务。
 - 在存储服务器上安装多磁盘并通过Raid技术实现存储容灾。
 - 支持MacOS、Linux、Windows等多操作系统。

■ 方案设计

- 通过NFS实现共享大容量网络存储。
- 挂载两块磁盘并构建RAID 1，实现存储容灾。
- 通过访问限制，仅允许工作室内部网络可访问。



2. 实现文件共享服务器

2.2 实现NFS文件服务器



实现NFS文件服务

任务目标:

- 在openEuler上安装NFS软件，并通过NFS实现工作组内的网络共享存储服务。

操作演示:



操作步骤:

- 实现在线安装NFS
- 实现NFS服务管理
- 实现在Windows上访问NFS服务
- 实现NFS服务的规划设计
- 实现工作组内的网络共享存储服务
- 测试与交付



2. 实现文件共享服务器

- 跨操作系统共享的问题：
 - Linux/UNIX操作系统间可通过NFS实现资源共享。
 - Windows/MS-DOS操作系统间可通过Samba实现资源共享。
 - Linux/UNIX操作系统通过NFS共享的资源，Windows/MS-DOS可以通过NFS协议访问，但Windows上的NFS客户端长期不完善。
 - 由于微软公司没有将SMB协议公开，如果想在Linux/UNIX访问Windows通过SMB共享的资源，是不可能的。
- 为了实现Linux/UNIX与Windows系统间进行资源共享，Samba的创始人Andrew Tridgwell通过对数据包的分析，编写了Samba自由软件。
 - 实现了在Linux/UNIX系统上启用Samba服务后，可利用SMB协议与Windows系统之间实现资源共享。



2. 实现文件共享服务器

- Samba是开放源代码的GPL自由软件，其实现了类UNIX与Windows之间通过SMB协议进行资源共享与访问。
 - Samba在设计上是让Linux/UNIX系统加入到Windows网络中，而不是让Windows加入类UNIX网络中。（Microsoft不积极配合）
 - Samba is the standard Windows interoperability suite of programs for Linux and Unix.（标准的Windows互操作性的软件套件）
 - Samba is Free Software licensed under the GNU General Public License, the Samba project is a member of the [Software Freedom Conservancy](#).
 - Since 1992, Samba has provided **secure**, **stable** and **fast** file and print services for all clients using the **SMB/CIFS protocol**, such as all versions of DOS and Windows, OS/2, Linux and many others.



Search Samba.org:

SAMBA

opening windows to a wider world

- Home
- think Samba
- get Samba
- learn Samba
- talk Samba
- hack Samba
- contact Samba
- support Samba

About Samba

Samba is the standard Windows interoperability suite of programs for Linux and Unix.

Samba is Free Software licensed under the GNU General Public License, the Samba project is a member of the Software Freedom Conservancy.

Since 1992, Samba has provided secure, stable and fast file and print services for all clients using the SMB/CIFS protocol, such as all versions of DOS and Windows, OS/2, Linux and many others.

Samba is an important component to seamlessly integrate Linux/Unix Servers and Desktops into Active Directory environments. It can function both as a domain controller or as a regular domain member.

Releases

Current stable release

Samba 4.21.1 (gzipped)
Release Notes - Signature

Release History

Versions & Notes

Maintenance

Patches - Security Updates - GPG Key

Future

Release Planning - Roadmap

Beyond Samba

Commercial Support

Global - By Country

Conferences

SAMBA by NetSint
SDC by SNA

Related Sites

- THINK SAMBA
 - What Is Samba?
 - Latest News
 - Planet Samba
 - FAQ
- GET SAMBA
 - Download Info
 - How To Install
 - GUIs
- LEARN SAMBA
 - Docs And Books
 - Wiki
- TALK SAMBA
 - List Subscribes
 - List Archives
 - IRC
 - Etiquette
- HACK SAMBA
 - Devrel Overview
 - Git Source
 - Built Farm
 - Bug Reports
- CONTACT SAMBA
 - Samba Team
 - Donations
 - Contacts For...

Search Samba.org:

SAMBA

opening windows to a wider world

Download

The Samba source code is distributed via https. View the download area via [HTTPS](https://www.samba.org/samba/download/). The file you probably want is called `samba-latest.tar.gz`. Old releases are available in the [Samba archives](#).

The Samba distribution GPG public key can be used to verify that current releases have not been tampered with. Using GnuPG, simply download the Samba source distribution, the tarball signature, and the Samba distribution public key. Then run

```
$ gpg --import samba-pubkey.asc
$ gunzip samba-version.tar.gz
$ gpg --verify samba-release.tar.asc
gpg: Signature made Tue 20 Nov 2007 07:12:04 PM CST using \
DSA key ID 6568B7EA
gpg: Good signature from "Samba Distribution Verification Key \
<samba-tugs@samba.org"
```

It is recommended that you also review the [list of patches](#) for current releases. For information on Samba security releases, please see our [security page](#).

Binaries

All major Linux and Free Unix distributions have Samba as a native package. See your distributor's package or port system for a native install of samba on your system.

<https://samba.plus/> offers Samba packages for SLES, RHEL, and Debian and AIX.

<http://en.opensuse.org/Samba> offers Samba packages for all SuSE Linux products (including SLES).

Git Sources

You can also fetch the sources using the [GIT source code control system](#). The advantage of fetching via GIT is can update your sources at any time using a single command. See the [Git instructions](#).

Tools

- Samba GUI managers
- Wireshark (decodes NetBIOS, SMB/CIFS, & MS-RPC)
- tcpdump (command line packet sniffer)
- NetBIOS Auditing Tool (NAT)
- wsdid is a Web Service Discovery (WSD) daemon written in Python
- wsdid2 is a WSD/LLMNR discovery daemon written in C
- nbflw, the NetBIOS forwarder (obsoleted)

SMB/CIFS Clients

- Linux CIFS VFS
- Dave (Macintosh) (obsolete)
- ONX (Samba Server/Client)
- IBM LAN Client 2.x
- Starfly

Microsoft's various clients: including WFWG3.11, LanMan for DOS, LanMan for OS/2 (v1.2, 1.3 or 2.x, Warp 3.0, not Warp connect,), MSCClient 3.0 for DOS. Check the README.NOW file for details.

Please note: Windows 2000, Windows XP, Windows NT, Windows 95, OS/2 Warp Connect and OS/2 Warp 4 clients don't need any extra software in order to run talk to a Samba server. These OS's come standard with TCP/IP which is all you need.

- Mac OS X also ships with a native CIFS client. Since OS X 10.9 Apple moved away from AFP to SMB as standard file sharing protocol.
- Several flavors of BSD—FreeBSD, NetBSD, DragonFly BSD—ship with smbfs support.

[This article was originally written for publication by [the Open Group](#).]

Samba: An Introduction

Chris Hertel
[Samba Team](#)
[JCIFS Team](#)

Last updated: \$Date: 2001/11/27 21:50:29 \$ GMT

The story goes something like this:

Linus Torvalds, the creator of the Linux Kernel, was visiting his friend Andrew Tridgell, the creator of the Samba suite. They were walking through the Zoo in Canberra when, without warning, a huge flock of vampire attack penguins dove out of the sky and tried to carry Linus away. Fortunately, Andrew had an umbrella. Still, one of the birds was able to nip Linus' hand with its fanged beak. Rumor has it that on moonlit nights Linus still runs out into the darkness and jumps, stark naked, into icy water. Of course, he's Finnish and may always have done this. In any case, this is why the Penguin is the Linux Mascot.

Andrew says that the story has changed a bit since the actual event.

What Samba is All About

The commercialization of the Internet over the past few years has created something of a modern melting pot. It has brought business-folk and technologists closer together than was previously thought possible. As a side effect, Windows and Unix systems have been invading each others' turf, and people expect that they will not only play together nicely, but that they will **share**.

A lot of emphasis has been placed on peaceful coexistence between Unix and Windows. The [Usenix Association](#) has even created an annual conference ([LISA/NT--July 14-17, 1999](#)) around this theme. Unfortunately, the two systems come from very different cultures and they have difficulty getting along without mediation. ...and that, of course, is Samba's job. [Samba](#) runs on Unix platforms, but speaks to Windows clients like a native. It allows a Unix system to move into a Windows "Network Neighborhood" without causing a stir. Windows users can happily access file and print services without knowing or caring that those services are being offered by a Unix host.

All of this is managed through a protocol suite which is currently known as the "Common Internet File System", or CIFS. This name was introduced by [Microsoft](#), and provides some insight into their hopes for the future. At the heart of CIFS is the latest incarnation of the Server Message Block (SMB) protocol, which has a long and tedious history. Samba is an open source CIFS implementation, and is available for free from the <http://samba.org/> mirror sites.

Samba and Windows are not the only ones to provide CIFS networking. OS/2 supports SMB file and print sharing, and there are commercial CIFS products for Macintosh and other platforms (including several others for Unix). Samba has been ported to a variety of non-Unix operating systems, including VMS, AmigaOS, & NetWare. CIFS is also supported on dedicated file server platforms from a variety of vendors. In other words, this stuff is all over the place.

History - the (hopefully) Untedious Version

It started a long time ago, in the early days of the PC, when IBM and Sytec co-developed a simple networking system designed for building small LANs. The system included something called NetBIOS, or *Network Basic Input Output System*. NetBIOS was a chunk of software that was loaded into memory to provide an interface between programs and the network hardware. It included an addressing scheme that used 16-byte names to identify workstations and network-enabled applications. Next, Microsoft added features to DOS that allowed disk I/O to be *redirected* to the NetBIOS interface, which made disk

2. 实现文件共享服务器

- Samba服务由smbd和nmbd两个核心进程组成。
 - smbd进程管理Samba服务器上的临时目录和打印机等，主要对网络上的共享资源进行管理。
 - nmbd进程进行NetBIOS名称解析，并提供浏览服务，列出网络上共享资源列表。

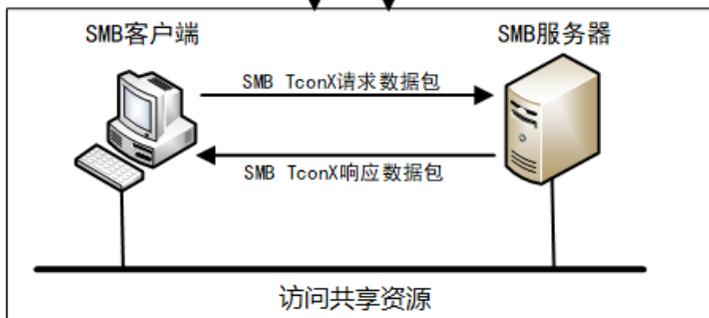
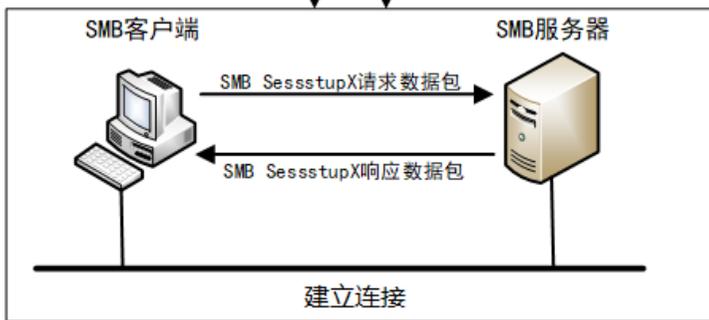
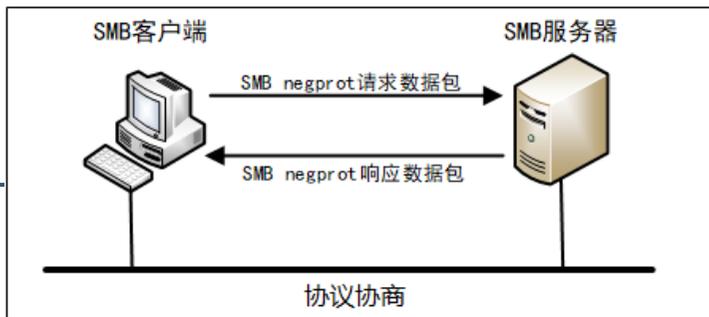


2. 实现文件共享服务器

2.3 实现Samba服务器

□ Samba服务的工作流程：

- 协议协商
- 建立连接
- 访问共享资源
- 断开连接



2. 实现文件共享服务器

□ Samba服务器的安全模式：

■ share安全级别模式

- 客户端登录Samba服务器，不需要输入用户名和密码就可以浏览Samba服务器的资源，适用于公共的共享资源，安全性差，需要配合其他权限设置来保证安全性。

■ user安全级别模式

- 客户端登录Samba服务器，需要提交合法帐号和密码，经过服务器验证才可以访问共享资源，服务器默认为此级别模式。

■ server安全级别模式

- 客户端需要将用户名和密码，提交到指定的一台Samba服务器上验证，如果验证出现错误，客户端会用user级别访问，实现集中式的认证管理。

■ domain安全级别模式

- 加入Windows域环境中，验证工作将由Windows域控制器负责。

■ ads安全级别模式

- 具备了domain安全级别模式中所有的功能并可以具备域控制器的功能。



2. 实现文件共享服务器

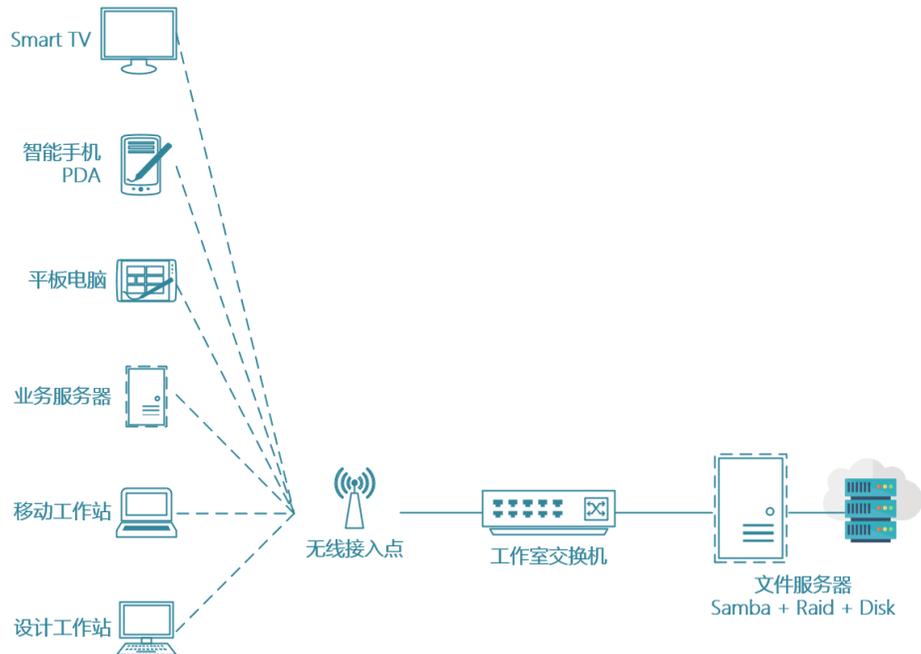
2.3 实现Samba服务器

□ 应用场景

- 某团队为提高信息化应用水平，提高数据共享和资源服务水平，现需要构建内部网络存储，并能够全面支持移动终端等智能设备，实现灵活的资源共享。

■ 需求分析

- 建设内部共享服务。
- 支持全终端智能设备。



2. 实现文件共享服务器

□ 方案设计

- 通过Samba建设网络存储服务。
- 仅允许团队内部进行网络访问。
- 支持多操作系统、支持多终端。

表 8-3-1 用户信息

序号	账号	密码
1	smbworkuser	smbworkuser@pwd
2	smbshareuser	smbshareuser@pwd

表 8-3-2 共享目录权限对应关系

序号	账号	/srv/smbfile/smbpublic	/srv/smbfile/smbshare	/srv/smbfile/smbwork
1	smbworkuser	读、写	读、写	读、写
2	smbshareuser	读、写	读、写	读



2. 实现文件共享服务器

2.3 实现Samba服务器



实现Samba文件服务

任务目标:

- 在openEuler上安装Samba软件，实现Samba服务。
- 通过Samba实现面向全终端的文件共享服务。

操作演示:



操作步骤:

- 实现在线安装Samba
- 实现Samba服务管理
- 实现Samba服务的匿名访问
- 实现文件共享服务的规划设计
- 实现全终端的文件共享服务
- 测试与交付



2. 实现文件共享服务器

2.4 使用Nextcloud实现网盘服务

- Nextcloud是一款开源免费的私有云存储网盘项目。
 - 可以快速便捷地搭建一套属于自己或团队的云同步网盘。
 - 实现跨平台跨设备文件同步、共享、版本控制、团队协作等功能。
- Nextcloud客户端覆盖了：
 - Windows、Mac、Android、iOS、Linux 等各种常见平台。
 - 提供了网页端以及WebDAV接口。





Register now



Nextcloud 9 Hub

Regain control over your data

Remote collaboration made easy

- ✓ On-premises or cloud
- ✓ Customizable and scalable
- ✓ No data leaks to third parties

Try Nextcloud now

Download



2. 实现文件共享服务器

2.4 使用Nextcloud实现云盘服务



使用Nextcloud实现私有云盘服务

任务目标:

- 在openEuler 上安装Nextcloud Server 软件，实现私有云盘服务。

操作步骤:

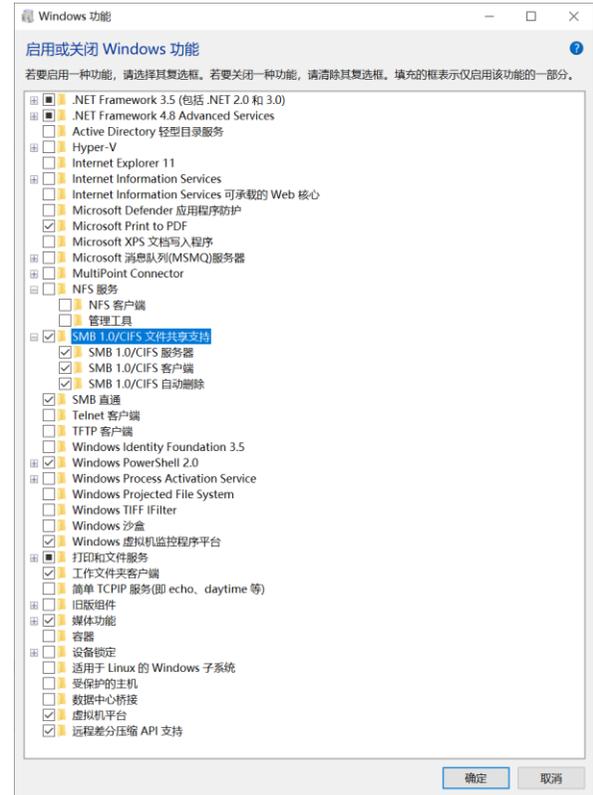
- 实现在线安装Nextcloud
- 实现多终端访问Nextcloud私有云盘
- 测试与交付

操作演示:



3. 文件传输协议与文件共享服务

序号	文件传输协议	文件共享服务		CLI工具	支持操作系统
		服务器软件	客户端软件		
1	FTP	WU-FTP ProFTP FileZilla Server vsftpd	FileZilla Client FireFTP NcFTP	FTP	Windows Linux Unix Mac
2	NFS	NFS-Utils Windows Server NFS haneWIN NFS Server	NFS-Utils Windows NFS Client	由服务器 软件决定	
3	SMB/CIFS	Samba SMB 1.0/CIFS Server	Samba SMB 1.0/CIFS Client		



Networking Software for Windows

https://www.hanewin.net/nfs-e.htm

Welcome to **haneWIN Software** *Networking Software for Windows* Deutsche Version

Home Software Shop Contact

Software:

- LAN
- ISDN
- Internet Server
- DHCP Server
- DNS Server
- »NFS Server
- TFTP Server/Client

Infos:

- OpenPGP Encryption in Javascript
- Booting a Raspberry Pi from Windows
- Raspberry Pi with LibreELEC as WiFi bridge
- Raspberry Pi controlled DDS VFO

haneWIN NFS Server for Windows

The software implements a multithreaded **NFS Server** for all Windows platforms.

The implementation supports:

- NFS 3, NFS 2, WebNFS and NLM protocol.
- UDP as well as TCP connections.
- Unix soft-links (symbolic links).
- Unix hard-links on NTFS volumes
- If supported symbolic links can use Windows reparse points.
- Inodes for block- and character devices.
- Implemented as Windows service and Windows application
- Use of NTFS file id as inode for Windows 7 and higher

The installer contains the 32bit software and native x64 versions of the programs.

The **NFS Server** runs as a background service on Windows XP/VISTA/20xx/7/8/10/11. The necessary **SunPRC PortMap Daemon** is implemented as an independent service.

The NFS server application is a portable Windows application with a built-in SunRPC Portmapper.

Connections with the NFS Server are supported over **UDP** or **TCP** using NFS 3 or NFS 2 protocol. For connection setup the WebNFS protocol **WebNFS** (RFC 2055) and the **Sun RPC** based **mount** protocol is supported. Multithreaded implementation with a configurable number of UDP server threads and one thread per TCP connection.

The software is shareware.

If you have questions regarding the NFS server, please read the FAQ
If you need further information send us a mail.

NFS Server 1.2.68 service installer (~1029kB)
MD5sum

NFS Server 1.2.68 Application (~510kB)
MD5sum

Update Info

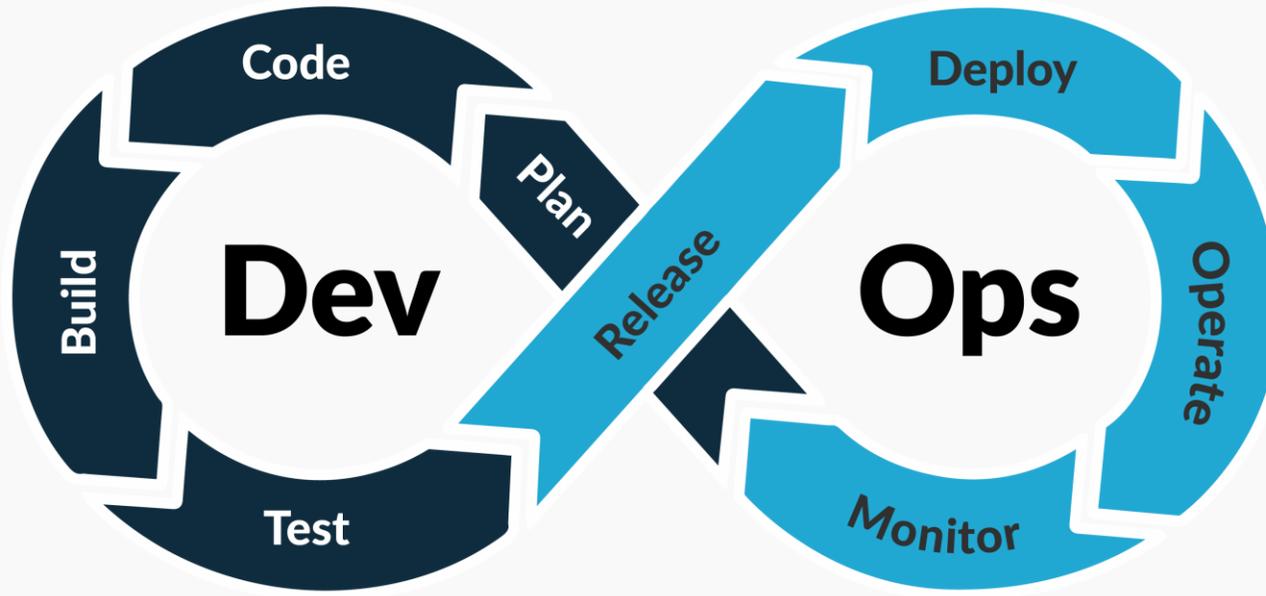
Manual

FAQ

other Languages:
Chinese
Dutch
Japanese
Portugues-BR
Spanish

haneWIN Software Imprint Home





**Development
Team**

**Operations
Team**

Dev + Ops = DevOps

网络与信息系统智能运维 课程体系学习平台

本课程体系由
河南中医药大学信息技术学院建设

课程体系学习平台由河南中医药大学医疗健康信息
工程技术研究所开发与技术保障

网络与信息系统智能运维课程体系学习平台
<https://internet.hactcm.edu.cn>

互联网运维管理工程应用丛书
<http://www.51xueweb.cn>



扫码学习
并获取课程资源

