



Red Hat Enterprise Linux 8

部署不同类型的服务器

在 Red Hat Enterprise Linux 8 中部署不同类型的服务器的指南

Red Hat Enterprise Linux 8 部署不同类型的服务器

在 Red Hat Enterprise Linux 8 中部署不同类型的服务器的指南

法律通告

Copyright © 2021 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档论述了如何在 Red Hat Enterprise Linux 8 中配置和运行不同类型的服务器，包括 Apache HTTP web 服务器、Samba 服务器、NFS 服务器、可用数据库服务器以及 CUPS 服务器。

目录

使开源包含更多	8
对红帽文档提供反馈	9
第 1 章 设置 APACHE HTTP WEB 服务器	10
1.1. APACHE HTTP WEB 服务器简介	10
1.1.1. Apache HTTP 服务器中的显著变化	10
1.1.2. 更新配置	11
1.2. APACHE 配置文件	12
1.3. 管理 HTTPD 服务	12
1.4. 设置单实例 APACHE HTTP 服务器	13
1.5. 配置基于 APACHE 名称的虚拟主机	14
1.6. 为 APACHE HTTP WEB 服务器配置 KERBEROS 验证	16
1.7. 在 APACHE HTTP 服务器中配置 TLS 加密	17
1.7.1. 在 Apache HTTP 服务器中添加 TLS 加密	17
1.7.2. 在 Apache HTTP 服务器中设置支持的 TLS 协议版本	19
1.7.3. 在 Apache HTTP 服务器中设置支持的密码	20
1.8. 配置 TLS 客户端证书身份验证	21
1.9. 安装 APACHE HTTP 服务器手册	22
1.10. 使用模块	23
1.10.1. 载入模块	23
1.10.2. 编写模块	23
1.11. 从 NSS 数据库导出私钥和证书，以便在 APACHE WEB 服务器配置中使用它们	23
1.12. 其它资源	25
第 2 章 设置和配置 NGINX	26
2.1. 安装并准备 NGINX	26
2.2. 将 NGINX 配置为一个为不同域提供不同内容的 WEB 服务器	27
2.3. 在 NGINX WEB 服务器中添加 TLS 加密	29
2.4. 将 NGINX 配置为 HTTP 流量的反向代理	30
2.5. 将 NGINX 配置为 HTTP 负载均衡器	31
2.6. 其它资源	32
第 3 章 使用 SAMBA 作为服务器	33
3.1. 了解不同的 SAMBA 服务和模式	33
3.1.1. Samba 服务	33
3.1.2. Samba 安全服务	34
3.1.3. Samba 服务和 Samba 客户端工具加载并重新载入其配置的情况	34
3.1.4. 以安全的方式编辑 Samba 配置	34
3.2. 验证 SAMBA 配置	35
3.2.1. 使用 testparm 工具验证 smb.conf 文件	35
3.3. 将 SAMBA 设置为独立服务器	36
3.3.1. 为独立服务器设置服务器配置	36
3.3.2. 创建并启用本地用户帐户	37
3.4. 了解并配置 SAMBA ID 映射	38
3.4.1. 规划 Samba ID 范围	38
3.4.2. * 默认域	39
3.4.3. 使用 tdb ID 映射后端	40
3.4.4. 使用 ad ID 映射后端	40
3.4.5. 使用网格 ID 映射后端	42
使用网格后端的好处	43
使用网格后端的缺陷	43

3.4.6. 使用自动 ID 映射后端	44
使用自动扩展后端的好处	44
缺陷	45
3.5. 将 SAMBA 设置为 AD 域成员服务器	46
3.5.1. 将 RHEL 系统添加到 AD 域中	46
3.5.2. 使用 MIT Kerberos 的本地授权插件	48
3.6. 在 IDM 域成员中设置 SAMBA	49
3.6.1. 准备 IdM 域以便在域成员中安装 Samba	49
3.6.2. 使用 GPO 在 Active Directory 中启用 AES 加密类型	51
3.6.3. 在 IdM 客户端中安装和配置 Samba 服务器	51
3.6.4. 如果 IdM 信任新域，请手动添加 ID 映射配置	53
3.6.5. 其它资源	54
3.7. 设置使用 POSIX ACL 的 SAMBA 文件共享	55
3.7.1. 添加使用 POSIX ACL 的共享	55
3.7.2. 在使用 POSIX ACL 的 Samba 共享中设置标准 Linux ACL	56
3.7.3. 在使用 POSIX ACL 的 Samba 共享中设置扩展的 ACL	56
3.8. 对使用 POSIX ACL 的共享设置权限	58
3.8.1. 配置基于用户和组群的共享访问权限	59
3.8.2. 配置基于主机的共享访问权限	59
3.9. 设置使用 WINDOWS ACL 的共享	60
3.9.1. 授予 SeDiskOperatorPrivilege 权限	60
3.9.2. 启用 Windows ACL 支持	60
3.9.3. 添加使用 Windows ACL 的共享	61
3.9.4. 管理使用 Windows ACL 的共享的共享权限和文件系统 ACL	62
3.10. 使用 SMBCACLS 在 SMB 共享中管理 ACL	62
3.10.1. 访问控制条目	62
3.10.2. 使用 smbcacls 显示 ACL	65
3.10.3. ACE 掩码计算	66
3.10.4. 使用 smbcacls 添加、更新和删除 ACL	66
添加 ACL	66
更新 ACL	66
删除 ACL	67
3.11. 允许用户在 SAMBA 服务器上共享目录	67
3.11.1. 启用用户共享功能	67
3.11.2. 添加用户共享	68
3.11.3. 更新用户共享的设置	68
3.11.4. 显示现有用户共享的信息	69
3.11.5. 列出用户共享	69
3.11.6. 删除用户共享	69
3.12. 配置共享以允许不进行身份验证的访问	70
3.12.1. 启用对共享的客户机访问	70
3.13. 为 MACOS 客户端配置 SAMBA	71
3.13.1. 优化 Samba 配置，以便为 macOS 客户端提供文件共享	71
3.14. 使用 SMBCLIENT 实用程序访问 SMB 共享	72
3.14.1. smbclient 互动模式如何工作	72
3.14.2. 在互动模式中使用 smbclient	73
3.14.3. 在脚本模式中使用 smbclient	73
3.15. 将 SAMBA 设置为打印服务器	74
3.15.1. Samba spoolssd 服务	74
3.15.2. 在 Samba 中启用打印服务器支持	75
3.15.3. 手动共享特定打印机	76
3.16. 在 SAMBA 打印服务器中为 WINDOWS 客户端设置自动打印机驱动程序下载	77
3.16.1. 有关打印机驱动程序的基本信息	77

支持的驱动程序模型版本	77
软件包感知驱动程序	77
准备上传的打印机驱动程序	77
为客户端提供 32 位和 64 位驱动	78
3.16.2. 启用用户上传和预配置驱动程序	78
3.16.3. 设置 print\$ 共享	78
3.16.4. 创建 GPO 以启用客户端信任 Samba 打印服务器	80
3.16.5. 上传驱动程序和预配置打印机	83
3.17. 调整 SAMBA 服务器的性能	83
3.17.1. 设置 SMB 协议版本	83
3.17.2. 与包含大量文件的目录调整共享	84
3.17.3. 可能会对性能造成负面影响的设置	85
3.18. 将 SAMBA 配置为与需要 SMB 版本低于默认版本的客户端兼容	85
3.18.1. 设置 Samba 服务器支持的最小 SMB 协议版本	85
3.19. 经常使用 SAMBA 命令行工具	85
3.19.1. 使用 net ads join 和 net rpc join 命令	85
3.19.2. 使用 net rpc right 命令	87
列出您可以设置的权限	87
授予权限	87
撤销权限	87
3.19.3. 使用 net rpc share 命令	87
列出共享	87
添加共享	88
删除共享	88
3.19.4. 使用 net user 命令	88
列出域用户帐户	89
在域中添加用户帐户	89
从域中删除用户帐户	89
3.19.5. 使用 rpcclient 工具	89
示例	90
3.19.6. 使用 samba-regedit 应用程序	90
3.19.7. 使用 smbcontrol 工具	91
3.19.8. 使用 smbpasswd 工具	92
3.19.9. 使用 smbstatus 工具	93
3.19.10. 使用 smbtar 工具	93
3.19.11. 使用 wbinfos 工具	94
3.20. 相关信息	95
第 4 章 导出 NFS 共享	96
4.1. NFS 简介	96
4.2. 支持的 NFS 版本	96
默认 NFS 版本	96
次要 NFS 版本的特性	96
4.3. NFSV3 和 NFSV4 中的 TCP 和 UDP 协议	97
4.4. NFS 所需的服务	97
NFSv4 的 RPC 服务	98
4.5. NFS 主机名格式	98
4.6. NFS 服务器配置	98
4.6.1. /etc/exports 配置文件	98
导出条目	99
默认选项	100
默认和覆盖选项	100
4.6.2. exportfs 工具	101

常用的 exportfs 选项	101
4.7. NFS 和 RPCBIND	101
4.8. 安装 NFS	102
4.9. 启动 NFS 服务器	102
4.10. NFS 和 RPCBIND 故障排除	102
4.11. 将 NFS 服务器配置为在防火墙后运行	103
4.12. 通过防火墙导出 RPC 配额	104
4.13. 通过 RDMA(NFSORDMA)启用 NFS	105
4.14. 配置只使用 NFSv4 的服务器	105
4.14.1. 只使用 NFSv4 的服务器的好处和缺陷	105
4.14.2. 将 NFS 服务器配置为只支持 NFSv4	106
4.14.3. 验证只读 NFSv4 配置	106
4.15. 相关信息	107
第 5 章 保护 NFS	108
5.1. 带有 AUTH_SYS 和导出控制的 NFS 安全性	108
5.2. 使用 AUTH_GSS 的 NFS 安全性	108
5.3. 配置 NFS 服务器和客户端使用 KERBEROS	108
5.4. NFSv4 安全选项	109
5.5. 挂载的 NFS 导出的文件权限	109
第 6 章 在 NFS 中启用 PNFS SCSI 布局	110
6.1. PNFS 技术	110
6.2. PNFS SCSI 布局	110
客户端和服务端间的操作	110
设备保留	110
6.3. 检查与 PNFS 兼容的 SCSI 设备	111
6.4. 在服务器中设置 PNFS SCSI	111
6.5. 在客户端中设置 PNFS SCSI	112
6.6. 在服务器中释放 PNFS SCSI 保留	112
6.7. 监控 PNFS SCSI 布局功能	113
6.7.1. 使用 nfsstat 从服务器检查 pNFS SCSI 操作	114
6.7.2. 使用 mountstats 检查客户端中的 pNFS SCSI 操作	114
第 7 章 配置 SQUID 缓存代理服务器	116
7.1. 将 SQUID 设置为没有身份验证的缓存代理	116
7.2. 使用 LDAP 身份验证将 SQUID 设置为缓存代理	118
7.3. 使用 KERBEROS 验证将 SQUID 设置为缓存代理	121
7.4. 在 SQUID 中配置域拒绝列表	124
7.5. 将 SQUID 服务配置为侦听特定端口或 IP 地址	124
7.6. 其它资源	125
第 8 章 数据库服务器	126
8.1. 介绍	126
8.2. 使用 MARIADB	126
8.2.1. MariaDB 入门	126
8.2.2. 安装 MariaDB	126
8.2.2.1. 提高 MariaDB 安装安全性	127
8.2.3. 配置 MariaDB	127
8.2.3.1. 为网络配置 MariaDB 服务器	127
8.2.4. 备份 MariaDB 数据	127
8.2.4.1. 使用 mysqldump 执行逻辑备份	128
8.2.4.1.1. 使用 mysqldump 备份整个数据库	128
8.2.4.1.2. 使用 mysqldump 备份来自一个数据库的一组表	128

8.2.4.1.3. 使用 mysqldump 将转储文件重新加载到服务器中	129
8.2.4.1.4. 使用 mysqldump 在两个数据库之间复制数据	129
8.2.4.1.5. 使用 mysqldump 转储多个数据库	129
8.2.4.1.6. 使用 mysqldump 转储所有数据库	129
8.2.4.1.7. 查看 mysqldump 选项	129
8.2.4.1.8. 其它资源	129
8.2.4.2. 使用 Mariabackup 工具执行物理在线备份	130
8.2.4.3. 使用 Mariabackup 工具恢复数据	131
8.2.4.3.1. 在保留备份文件时使用 Mariabackup 恢复数据	131
8.2.4.3.2. 在删除备份文件时使用 Mariabackup 恢复数据	131
8.2.4.3.3. 其它资源	132
8.2.4.4. 执行文件系统备份	132
8.2.4.5. 使用复制作为备份解决方案的介绍	133
8.2.5. 迁移到 MariaDB 10.3	133
8.2.5.1. RHEL 7 和 RHEL 8 版本的 MariaDB 之间的显著区别	133
8.2.5.2. 配置更改	133
8.2.5.3. 使用 mysql_upgrade 工具进行原位升级	134
8.2.6. 使用 Galera 复制 MariaDB	135
8.2.6.1. MariaDB Galera 集群介绍	135
8.2.6.2. 构建 MariaDB Galera 集群的组件	136
8.2.6.3. 部署 MariaDB Galera 集群	136
8.2.6.4. 在 MariaDB Galera 集群中添加新节点	138
8.2.6.5. 重启 MariaDB Galera 集群	138
8.3. 使用 POSTGRESQL	139
8.3.1. PostgreSQL 入门	139
8.3.2. 安装 PostgreSQL	139
8.3.3. 配置 PostgreSQL	140
8.3.3.1. 初始化数据库集群	140
8.3.4. 备份 PostgreSQL 数据	140
8.3.4.1. 使用 SQL 转储备份 PostgreSQL 数据	141
8.3.4.1.1. 执行 SQL 转储	141
8.3.4.1.2. 从 SQL 转储中恢复数据库	141
8.3.4.1.2.1. 在另一个服务器中恢复数据库	142
8.3.4.1.2.2. 在恢复过程中处理 SQL 错误	142
8.3.4.1.3. SQL 转储的优点和缺陷	142
8.3.4.1.4. 其它资源	142
8.3.4.2. 使用文件系统级别备份来备份 PostgreSQL 数据	142
8.3.4.2.1. 执行文件系统级别备份	142
8.3.4.2.2. 文件系统级别备份的优点和缺陷	143
8.3.4.2.3. 文件系统级别备份的替代方法	143
8.3.4.2.4. 其它资源	143
8.3.4.3. 通过持续存档来备份 PostgreSQL 数据	143
8.3.4.3.1. 持续归档介绍	143
8.3.4.3.2. 执行持续存档备份	144
8.3.4.3.2.1. 进行基础备份	144
8.3.4.3.2.2. 使用持续归档备份来恢复数据库	144
8.3.4.3.3. 持续归档的优点和缺陷	145
8.3.4.3.4. 其它资源	145
8.3.5. 迁移到 PostgreSQL 的 RHEL 8 版本	146
8.3.5.1. 使用 pg_upgrade 工具快速升级	146
8.3.5.2. 转储和恢复升级	148

第 9 章 配置打印	150
------------------	-----

9.1. 激活 CUPS 服务	150
9.2. 打印设置工具	150
9.3. 访问并配置 CUPS WEB UI	151
9.3.1. 获取 CUPS Web UI 的管理访问权限	152
9.4. 在 CUPS WEB UI 中添加打印机	153
9.5. 在 CUPS WEB UI 中配置打印机	157
9.6. 使用 CUPS WEB UI 打印测试页面	159
9.7. 使用 CUPS WEB UI 设置打印选项	159
9.8. 为打印服务器安装证书	160
9.9. 使用 SAMBA 打印到使用 KERBEROS 验证的 WINDOWS 打印服务器	162
9.10. 使用 CUPS 日志	164
9.10.1. CUPS 日志的类型	164
9.10.2. 访问 CUPS 日志	164
9.10.2.1. 访问所有 CUPS 日志	164
9.10.2.2. 访问特定打印作业的 CUPS 日志	165
9.10.2.3. 根据特定时间框架访问 CUPS 日志	165
9.10.2.4. 相关信息	165
9.10.3. 配置 CUPS 日志位置	165

使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始：master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。如需了解更多详细信息，请参阅 [CTO Chris Wright 信息](#)。

对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请让我们了解如何改进文档。要做到这一点：

- 关于特定内容的简单评论：
 1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
 2. 用鼠标指针高亮显示您想评论的文本部分。
 3. 点在高亮文本上弹出的 **Add Feedback**。
 4. 按照显示的步骤操作。
- 要提交更复杂的反馈，请创建一个 Bugzilla ticket：
 1. 进入 [Bugzilla](#) 网站。
 2. 在 Component 中选择 **Documentation**。
 3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
 4. 点 **Submit Bug**。

第 1 章 设置 APACHE HTTP WEB 服务器

1.1. APACHE HTTP WEB 服务器简介

*Web 服务器*是一个通过 Web 向客户端提供内容的网络服务。这通常是网页，但也可以提供任何其他文档。Web 服务器也称为 HTTP 服务器，因为它们使用 *超文本传输协议 (HTTP)*。

Apache HTTP 服务器 **httpd** 是由 [Apache Software Foundation](#) 开发的开源网页服务器。

如果您要从以前的 Red Hat Enterprise Linux 版本升级，则需要相应地更新 **httpd** 服务配置。本节介绍了一些新添加的功能，并指导您完成之前的配置文件的更新。

1.1.1. Apache HTTP 服务器中的显著变化

Apache HTTP 服务器 已从 RHEL 7 提供的版本 2.4.6 更新至 RHEL 8 提供的版本 2.4.37。这个版本包括了几个新功能，但在外部模块的配置和应用程序二进制接口 (ABI) 级别上保持与 RHEL 7 版本的向后兼容性。

新特性包括：

- 现在，**mod_http2** 软件包提供了 HTTP/2 支持，该软件包是 **httpd** 模块的一部分。
- 支持 **systemd** 套接字激活。详情请查看 **httpd.socket(8)** man page。
- 添加了多个新模块：
 - **mod_proxy_hcheck** - 代理健康检查模块
 - **mod_proxy_uwsgi** - Web Server 网关接口 (WSGI) 代理
 - **mod_proxy_fdpass** - 支持将客户端套接字传递给另一个进程
 - **mod_cache_socache** - 使用 memcache 后端的 HTTP 缓存
 - **mod_md** - ACME 协议 SSL/TLS 证书服务
- 现在默认载入以下模块：
 - **mod_request**
 - **mod_macro**
 - **mod_watchdog**
- 添加了一个新的子软件包 **httpd-filesystem**，它包含 Apache HTTP 服务器的基本目录布局，其中包括这些目录的正确权限。
- 对实例化服务的支持，引进了 **httpd@.service**。详情请查看 **httpd.service** man page。
- 新的 **httpd-init.service** 替换了 **%post script** 以创建一个自签名的 **mod_ssl** 密钥对。
- 现在，通过 **mod_md** 软件包支持使用自动证书管理环境 (ACME) 协议自动 TLS 证书置备和续订（与证书提供程序，如 **Let's Encrypt** 一起使用）。

- **Apache HTTP 服务器**现在支持直接从 **PKCS#11** 模块的硬件安全令牌加载 TLS 证书和私钥。因此，**mod_ssl** 配置现在可以使用 **PKCS#11** URL 识别 TLS 私钥，以及可选的 **SSLCertificateKeyFile** 和 **SSLCertificateFile** 指令中的 TLS 证书。
- 现在支持 **/etc/httpd/conf/httpd.conf** 文件中的新 **ListenFree** 指令。
与 **Listen** 指令类似，**ListenFree** 提供服务器侦听的 IP 地址、端口或 IP 地址和端口组合的信息。但是，使用 **ListenFree** 时，**IP_FREEBIND** socket 选项会被默认启用。因此，**httpd** 允许绑定到一个非本地 IP 地址，或绑定到不存在的 IP 地址。这允许 **httpd** 在不需要在 **httpd** 绑定时启用底层网络接口或指定的动态 IP 地址处于活跃状态的情况下，侦听套接字。

请注意，**ListenFree** 指令目前仅适用于 RHEL 8。

有关 **ListenFree** 的详情，请查看下表：

表 1.1. ListenFree 指令的语法、状态和模块

语法	状态	模块
ListenFree [IP-address:]portnumber [protocol]	MPM	event、worker、prefork、mpm_winnt、mpm_network、mpmt_os2

其他显著变化包括：

- 删除了以下模块：
 - **mod_file_cache**
 - **mod_nss**
使用 **mod_ssl** 作为替换。有关从 **mod_nss** 迁移的详情，请参阅 [第 1.11 节“从 NSS 数据库导出私钥和证书，以便在 Apache Web 服务器配置中使用它们”](#)。
 - **mod_perl**
- 在 RHEL 8 中，**Apache HTTP 服务器** 使用的默认 DBM 验证数据库类型已从 **SDBM** 改为 **db5**。
- **Apache HTTP 服务器**的 **mod_wsgi** 模块已更新为 Python 3。WSGI 应用程序现在只支持 Python 3,且必须从 Python 2 中迁移。
- 使用 **Apache HTTP 服务器** 默认配置的多处理模块（MPM）已从多处理模型（称为 **prefork**）改为高性能多线程模型 **event**。
任何不是线程的第三方模块都需要被替换或删除。要更改配置的 MPM，编辑 **/etc/httpd/conf.modules.d/00-mpm.conf** 文件。详情请查看 **httpd.service(8)** man page。
- 现在，suEXEC 允许的用户的 UID 和 GID 最少为 1000 和 500（之前为 100 和 100）。
- **/etc/sysconfig/httpd** 文件不再是一个支持用来为 **httpd** 服务设置环境变量的接口。为 systemd 服务添加了 **httpd.service(8)** man page。
- 现在，停止 **httpd** 服务默认使用“安全停止（graceful stop）”。
- **mod_auth_kerb** 模块已被 **mod_auth_gssapi** 模块替代。

1.1.2. 更新配置

要从 Red Hat Enterprise Linux 7 中使用的 **Apache HTTP 服务器** 版本更新配置文件，请选择以下选项之一：

- 如果 `/etc/sysconfig/httpd` 用于设置环境变量，请创建一个 `systemd drop-in` 文件。
- 如果使用任何第三方模块，请确保它们与线程 MPM 兼容。
- 如果使用 `suexec`，请确保用户和组群 ID 满足新的最小值。

您可以使用以下命令检查配置中的错误：

```
# apachectl configtest
Syntax OK
```

1.2. APACHE 配置文件

当 `httpd` 服务启动时，默认情况下，它会从 [表 1.2 “httpd 服务配置文件”](#) 中列出的位置读取配置。

表 1.2. httpd 服务配置文件

路径	描述
<code>/etc/httpd/conf/httpd.conf</code>	主配置文件。
<code>/etc/httpd/conf.d/</code>	主配置文件中包含的配置文件的辅助目录。
<code>/etc/httpd/conf.modules.d/</code>	用于载入 Red Hat Enterprise Linux 中打包动态模块的配置文件的辅助目录。在默认配置中，首先会处理这些配置文件。

虽然默认配置适合大多数情况，但您也可以使用其他配置选项。要让任何更改生效，请首先重启 web 服务器。如需了解有关如何重启 `httpd` 服务的更多信息，请参阅 [第 1.3 节 “管理 httpd 服务”](#)。

要检查配置中的可能错误，在 shell 提示符后输入以下内容：

```
# apachectl configtest
Syntax OK
```

要更方便地从错误中恢复，请在编辑前复制原始文件。

1.3. 管理 HTTPD 服务

这部分论述了如何启动、停止和重启 `httpd` 服务。

先决条件

- 已安装 Apache HTTP 服务器。

流程

- 要启动 `httpd` 服务，请输入：


```
# systemctl start httpd
```

- 要停止 **httpd** 服务，请输入：

```
# systemctl stop httpd
```

- 要重启 **httpd** 服务，请输入：

```
# systemctl restart httpd
```

1.4. 设置单实例 APACHE HTTP 服务器

这部分论述了如何设置单实例 Apache HTTP 服务器来提供静态 HTML 内容。

如果 web 服务器应该为与服务器关联的所有域提供相同的内容，请按照本节中的步骤进行操作。如果要为不同的域提供不同的内容，请设置基于名称的虚拟主机。详情请查看 [第 1.5 节“配置基于 Apache 名称的虚拟主机”](#)。

流程

1. 安装 **httpd** 软件包：

```
# yum install httpd
```

2. 在本地防火墙中打开 TCP 端口 **80**：

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

3. 启用并启动 **httpd** 服务：

```
# systemctl enable --now httpd
```

4. 可选：在 **/var/www/html/** 目录中添加 HTML 文件。



注意

当向 **/var/www/html/** 添加内容时，**httpd** 默认运行的用户必须可读取文件和目录。内容所有者可以是 **root** 用户和 **root** 用户组，也可以是管理员选择的其他用户或组。如果内容所有者是 **root** 用户和 **root** 用户组，则文件必须可以被其他用户读取。所有文件和目录的 SELinux 上下文必须是 **httpd_sys_content_t**，默认应用于 **/var/www** 目录中的所有内容。

验证步骤

- 与 Web 浏览器连接至 **http://server_IP_or_host_name/**。
如果 **/var/www/html/** 目录为空或者不包含 **index.html** 或 **index.htm** 文件，Apache 会显示 **Red Hat Enterprise Linux Test Page**。如果 **/var/www/html/** 包含具有不同名称的 HTML 文件，您可以通过输入 URL 到该文件来加载它们，如 **http://server_IP_or_host_name/example.html**。

其它资源

- 有关配置 Apache 和将服务限制到您的环境的详情，请参考 Apache 手册。有关安装手动的详情，请参考 [第 1.9 节“安装 Apache HTTP 服务器手册”](#)。
- 有关使用或调整 **httpd systemd** 服务的详情，请查看 **httpd.service(8)** man page。

1.5. 配置基于 APACHE 名称的虚拟主机

基于名称的虚拟主机可让 Apache 为解析到服务器 IP 地址的不同域提供不同的内容。

本节中的步骤论述了使用独立文档根目录为 **example.com** 和 **example.net** 域设置虚拟主机。两个虚拟主机都提供静态 HTML 内容。

先决条件

- 客户端和网页服务器会将 **example.com** 和 **example.net** 域解析为 web 服务器的 IP 地址。请注意，您必须手动将这些条目添加到 DNS 服务器中。

流程

1. 安装 **httpd** 软件包：

```
# yum install httpd
```

2. 编辑 **/etc/httpd/conf/httpd.conf** 文件：

- a. 为 **example.com** 域附加以下虚拟主机配置：

```
<VirtualHost *:80>
    DocumentRoot "/var/www/example.com/"
    ServerName example.com
    CustomLog /var/log/httpd/example.com_access.log combined
    ErrorLog /var/log/httpd/example.com_error.log
</VirtualHost>
```

这些设置配置以下内容：

- **<VirtualHost *:80>** 指令中的所有设置都是针对这个虚拟主机的。
- **DocumentRoot** 设置虚拟主机 Web 内容的路径。
- **ServerName** 设置此虚拟主机提供内容的域。
要设置多个域，在配置中添加 **ServerAlias** 参数，并指定使用这个参数中空格分开的额外域。
- **CustomLog** 设置虚拟主机访问日志的路径。
- **ErrorLog** 设置虚拟主机错误日志的路径。



注意

Apache 还将该配置中找到的第一个虚拟主机用于与 **ServerName** 和 **ServerAlias** 参数中设置的任何域不匹配的请求。这还包括发送到服务器 IP 地址的请求。

3. 为 **example.net** 域附加类似的虚拟主机配置：

```
<VirtualHost *:80>
    DocumentRoot "/var/www/example.net/"
    ServerName example.net
    CustomLog /var/log/httpd/example.net_access.log combined
    ErrorLog /var/log/httpd/example.net_error.log
</VirtualHost>
```

4. 为两个虚拟主机创建文档根目录：

```
# mkdir /var/www/example.com/
# mkdir /var/www/example.net/
```

5. 如果您在 **DocumentRoot** 参数中设置了不在 **/var/www/** 中的路径，在两个文档根中设置 **httpd_sys_content_t** 上下文：

```
# semanage fcontext -a -t httpd_sys_content_t "/srv/example.com(/.*)?"
# restorecon -Rv /srv/example.com/
# semanage fcontext -a -t httpd_sys_content_t "/srv/example.net(/.*)?"
# restorecon -Rv /srv/example.net/
```

这些命令在 **/srv/example.com/** 和 **/srv/example.net/** 目录中设置 **httpd_sys_content_t** 上下文。

请注意，您必须安装 **polycoreutils-python-utils** 软件包才能运行 **restorecon** 命令。

6. 在本地防火墙中打开端口 **80**：

```
# firewall-cmd --permanent --add-port=80/tcp
# firewall-cmd --reload
```

7. 启用并启动 **httpd** 服务：

```
# systemctl enable --now httpd
```

验证步骤

1. 在每个虚拟主机的文档 root 中创建不同的示例文件：

```
# echo "vHost example.com" > /var/www/example.com/index.html
# echo "vHost example.net" > /var/www/example.net/index.html
```

2. 使用浏览器并连接到 **http://example.com**。web 服务器显示 **example.com** 虚拟主机中的示例文件。
3. 使用浏览器并连接到 **http://example.net**。web 服务器显示 **example.net** 虚拟主机中的示例文件。

其它资源

- 有关配置 Apache 虚拟主机的详情，请参考 Apache 手册中的 **Virtual Hosts** 文档。有关安装手动的详情，请参考 [第 1.9 节 “安装 Apache HTTP 服务器手册”](#)。

1.6. 为 APACHE HTTP WEB 服务器配置 KERBEROS 验证

要在 Apache HTTP web 服务器 (**httpd**) 中执行 Kerberos 验证, RHEL 8 使用 **mod_auth_gssapi** 内核模块。Generic Security Services API (**GSSAPI**) 是请求使用安全库 (如 Kerberos) 的应用程序的接口。**GSS-Proxy** 允许为 **httpd** 服务器实施权限分离, 从安全视角看, 这优化了此进程。



注意

mod_auth_gssapi 模块替换已弃用的 **mod_auth_kerb** 模块。

先决条件

- 已安装 **httpd** 服务器。

这个步骤描述了如何在 Apache HTTP web 服务器 (**httpd**) 中设置 **GSS-Proxy** 来执行 Kerberos 验证。这是一个双向流程, 由设置 **GSS-Proxy** 和 **httpd** 服务器组成。

设置 GSS-Proxy

1. 通过创建服务主体来启用对 HTTP/server-name@realm 的 **keytab** 文件的访问。

```
# ipa service-add HTTP/server-name
```

2. 检索存储在 **/etc/gssproxy/http.keytab** 文件中的主体的 **keytab**。

```
# ipa-getkeytab -s $(awk '/^server =/{print $3}' /etc/ipa/default.conf) -k
/etc/gssproxy/http.keytab -p HTTP/$(hostname -f)
```

此步骤将权限设置为 400, 因此只有 **root** 用户可以访问 **keytab** 文件。Apache 用户无法访问。

3. 通过创建包含以下内容的 **/etc/gssproxy/80-httpd.conf** 在 **gssproxy** 配置中添加新部分 :

```
[service/HTTP]
mechs = krb5
cred_store = keytab:/etc/gssproxy/http.keytab
cred_store = ccache:/var/lib/gssproxy/clients/krb5cc_%U
euid = apache
```

4. 启动并启用 **gssproxy** 服务 :

```
# systemctl restart gssproxy.service
# systemctl enable gssproxy.service
```

设置 httpd 服务器

1. 选择应保护的文件或目录, 例如 **/var/www/html/private**。
2. 配置 **mod_auth_gssapi** 以保护该位置 :

```
<Location /var/www/html/private>
AuthType GSSAPI
AuthName "GSSAPI Login"
```

```
Require valid-user
</Location>
```

- 使用以下内容创建 **/etc/systemd/system/httpd.service** 文件：

```
.include /lib/systemd/system/httpd.service
[Service]
Environment=GSS_USE_PROXY=1
```

- 重新载入新添加的配置：

```
# systemctl daemon-reload
```

- 启动并启用 **gssproxy** 服务：

```
# systemctl restart gssproxy.service
# systemctl enable gssproxy.service
```

验证步骤

- 如果配置设置成功,您应该能够对服务器发出 HTTP 请求,并使用有效的 Kerberos ticket 进行身份验证。授予访问权限。

其它资源

- 有关使用或调整 **GSS-Proxy** 的详情,请查看 **gssproxy(8)**、**gssproxy-mech(8)** 和 **gssproxy.conf(5)** man page。

1.7. 在 APACHE HTTP 服务器中配置 TLS 加密

默认情况下,Apache 使用未加密的 HTTP 连接为客户端提供内容。这部分论述了如何在 Apache HTTP 服务器中启用 TLS 加密并配置经常使用的与加密相关的设置。

先决条件

- Apache HTTP 服务器已安装并运行。

1.7.1. 在 Apache HTTP 服务器中添加 TLS 加密

这部分论述了如何在 Apache HTTP 服务器中为 **example.com** 域启用 TLS 加密。

先决条件

- Apache HTTP 服务器已安装并运行。
- 私钥存储在 **/etc/pki/tls/private/example.com.key** 文件中。
有关创建私钥和证书签名请求(CSR)的详情,以及如何从证书颁发机构(CA)请求证书的详情,请查看您的 CA 文档。或者,如果您的 CA 支持 ACME 协议,可以使用 **mod_md** 模块自动检索和置备 TLS 证书。
- TLS 证书存储在 **/etc/pki/tls/private/example.com.crt** 文件中。如果您使用不同的路径,请修改流程的对应步骤。

- CA 证书存储在 `/etc/pki/tls/private/ca.crt` 文件中。如果您使用不同的路径,请修改流程的对应步骤。
- 客户端和网页服务器会将服务器的主机名解析为 web 服务器的 IP 地址。

流程

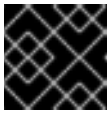
1. 安装 `mod_ssl` 软件包：

```
# dnf install mod_ssl
```

2. 编辑 `/etc/httpd/conf.d/ssl.conf` 文件并在 `<VirtualHost _default_:443>` 指令中添加以下设置：

- a. 设置服务器名称：

```
ServerName example.com
```



重要

服务器名称必须与证书的 **Common Name** 字段中设置的条目匹配。

- b. 可选：如果证书在 **Subject Alt Names (SAN)** 字段中包含额外主机名,您可以将 `mod_ssl` 配置为也为这些主机名提供 TLS 加密。要配置此功能，请使用对应名称添加 **ServerAliases** 参数：

```
ServerAlias www.example.com server.example.com
```

- c. 设置到私钥、服务器证书和 CA 证书的路径：

```
SSLCertificateKeyFile "/etc/pki/tls/private/example.com.key"
SSLCertificateFile "/etc/pki/tls/certs/example.com.crt"
SSLCACertificateFile "/etc/pki/tls/certs/ca.crt"
```

3. 出于安全考虑，请配置为只有 **root** 用户可以访问私钥文件：

```
# chown root:root /etc/pki/tls/private/example.com.key
# chmod 600 /etc/pki/tls/private/example.com.key
```



警告

如果私钥被设置为可以被未授权的用户访问，则需要撤销证书，然后再创建一个新私钥并请求一个新证书。否则，TLS 连接就不再安全。

4. 在本地防火墙中打开端口 **443**：

```
# firewall-cmd --permanent --add-port=443/tcp
# firewall-cmd --reload
```

5. 重启 **httpd** 服务：

```
# systemctl restart httpd
```

**注意**

如果您使用密码保护私钥文件，每次 **httpd** 服务启动时都必须输入这个密码。

验证步骤

- 使用浏览器并连接到 **https://example.com**。

其它资源

- 有关配置 TLS 的详情，请参考 Apache 手册中的 **SSL/TLS Encryption** 文档。有关安装手动的详情，请参考 [第 1.9 节 “安装 Apache HTTP 服务器手册”](#)。

1.7.2. 在 Apache HTTP 服务器中设置支持的 TLS 协议版本

默认情况下，RHEL 8 上的 Apache HTTP 服务器使用系统范围的加密策略来定义安全默认值，这些默认值与当前的浏览器兼容。例如：**DEFAULT** 策略定义了 **在 apache 中只启用 TLSv1.2 和 TLSv1.3 协议版本**。

这部分论述了如何手动配置 Apache HTTP 服务器支持的 TLS 协议版本。如果您的环境只需要启用特定的 TLS 协议版本，请按照以下步骤操作，例如：

- 如果您的环境要求客户端也可以使用弱 **TLS1** (TLSv1.0) 或 **TLS1.1** 协议。
- 如果您想将 Apache 配置为只支持 **TLSv1.2** 或 **TLSv1.3** 协议。

先决条件

- TLS 加密在服务器上启用，如 [第 1.7.1 节 “在 Apache HTTP 服务器中添加 TLS 加密”](#) 所述。

流程

1. 编辑 **/etc/httpd/conf/httpd.conf** 文件，并在您要为其设置 TLS 协议版本的 **<VirtualHost>** 指令中添加以下设置。例如，仅启用 **TLSv1.3** 协议：

```
SSLProtocol -All TLSv1.3
```

2. 重启 **httpd** 服务：

```
# systemctl restart httpd
```

验证步骤

1. 使用以下命令验证服务器支持 **TLSv1.3**：

```
# openssl s_client -connect example.com:443 -tls1_3
```

2. 使用以下命令验证服务器不支持 **TLSv1.2**：

```
# openssl s_client -connect example.com:443 -tls1_2
```

如果服务器不支持该协议，命令会返回一个错误：

```
140111600609088:error:1409442E:SSL routines:ssl3_read_bytes:tlsv1 alert protocol
version:ssl/record/rec_layer_s3.c:1543:SSL alert number 70
```

3. 可选：重复用于其他 TLS 协议版本的命令。

其它资源

- 有关系统范围的加密策略的详情,请查看 **update-crypto-policies(8)** man page 和 [使用系统范围的加密策略](#)。
- 有关 **SSLProtocol** 参数的详情,请参考 Apache 手册中的 **mod_ssl** 文档。有关安装手动的详情,请参考 [第 1.9 节 “安装 Apache HTTP 服务器手册”](#)。

1.7.3. 在 Apache HTTP 服务器中设置支持的密码

默认情况下，RHEL 8 上的 Apache HTTP 服务器使用系统范围的加密策略来定义安全默认值，这些默认值与当前的浏览器兼容。有关系统范围的加密所允许的加密列表,请查看 **/etc/crypto-policies/back-ends/openssl.config** 文件。

这部分论述了如何手动配置 Apache HTTP 服务器支持的加密。如果您的环境需要特定的加密系统，请按照以下步骤操作。

先决条件

- TLS 加密在服务器上启用，如 [第 1.7.1 节 “在 Apache HTTP 服务器中添加 TLS 加密”](#) 所述。

流程

1. 编辑 **/etc/httpd/conf/httpd.conf** 文件,并将 **SSLCipherSuite** 参数添加到您要为其设置 TLS 密码的 **<VirtualHost>** 指令中：

```
SSLCipherSuite
"EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH:!SHA1:!SHA256"
```

这个示例只启用 **EECDH+AESGCM**、**EDH+AESGCM**、**AES256+EECDH** 和 **AES256+EDH** 密码,并禁用所有使用 **SHA1** 和 **SHA256** 消息验证代码的密码。

2. 重启 **httpd** 服务：

```
# systemctl restart httpd
```

验证步骤

1. 显示 Apache HTTP 服务器支持的密码列表：
 - a. 安装 **nmap** 软件包：

```
# yum install nmap
```


- b. 使用 **nmap** 工具显示支持的加密系统：

```
# nmap --script ssl-enum-ciphers -p 443 example.com
...
PORT      STATE SERVICE
443/tcp    open  https
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 (dh 2048) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|
|_
```

其它资源

- 有关系统范围的加密策略的详情，请查看 **update-crypto-policies(8)** man page 和 [使用系统范围的加密策略](#)。
- 有关 **SSLCipherSuite** 参数的详情，请参考 Apache 手册中的 **mod_ssl** 文档。有关安装手动的详情，请参考 [第 1.9 节“安装 Apache HTTP 服务器手册”](#)。

1.8. 配置 TLS 客户端证书身份验证

客户端证书身份验证可让管理员只允许使用证书进行身份验证的用户访问 web 服务器上的资源。本节论述了如何为 **/var/www/html/Example/** 目录配置客户端证书验证。

如果 Apache HTTP 服务器使用 TLS 1.3 协议，某些客户端将需要额外的配置。例如，在 Firefox 中，将 **about:config** 菜单中的 **security.tls.enable_post_handshake_auth** 参数设置为 **true**。详情请查看 [Red Hat Enterprise Linux 8 中的传输层安全版本 1.3](#)。

先决条件

- TLS 加密在服务器上启用，如 [第 1.7.1 节“在 Apache HTTP 服务器中添加 TLS 加密”](#) 所述。

流程

- 编辑 **/etc/httpd/conf/httpd.conf** 文件并在您要配置客户端身份验证的 **<VirtualHost>** 指令中添加以下设置：

```
<Directory "/var/www/html/Example/">
    SSLVerifyClient require
</Directory>
```

SSLVerifyClient require 设置定义了服务器必须成功验证客户端证书，然后客户端才能访问 **/var/www/html/Example/** 目录中的内容。

- 重启 **httpd** 服务：

```
# systemctl restart httpd
```

验证步骤

- 使用 **curl** 实用程序在没有客户端验证的情况下访问 **https://example.com/Example/** URL：

```
$ curl https://example.com/Example/
curl: (56) OpenSSL SSL_read: error:1409445C:SSL routines:ssl3_read_bytes:tlsv13 alert
certificate required, errno 0
```

这个错误表示 web 服务器需要客户端证书验证。

2. 将客户端私钥和证书以及 CA 证书传递给 **curl**，以便通过客户端身份验证来访问该 URL：

```
$ curl --cacert ca.crt --key client.key --cert client.crt https://example.com/Example/
```

如果请求成功，**curl** 会显示存储在 **/var/www/html/Example/** 目录中的 **index.html** 文件。

其它资源

- 有关客户端验证的详情，请查看 Apache 手册中的 **mod_ssl Configuration How-To** 文档。有关安装手册的详情，请参考 [第 1.9 节 “安装 Apache HTTP 服务器手册”](#)。

1.9. 安装 APACHE HTTP 服务器手册

这部分论述了如何安装 Apache HTTP 服务器手册。手册提供了详细信息，例如：

- 配置参数和指令
- 性能调整
- 身份验证设置
- 模块
- 内容缓存
- 安全提示
- 配置 TLS 加密

安装后，您可以使用 Web 浏览器显示手册。

先决条件

- Apache HTTP 服务器已安装并运行。

流程

1. 安装 **httpd-manual** 软件包：

```
# yum install httpd-manual
```

2. 可选：默认情况下，所有连接到 Apache HTTP 服务器的客户端都可以显示手册。要把访问权限限制为特定 IP 范围（如 **192.0.2.0/24** 子网），编辑 **/etc/httpd/conf.d/manual.conf** 文件并在 **<Directory "/usr/share/httpd/manual">** 指令中添加 **Require ip 192.0.2.0/24** 设置：

```
<Directory "/usr/share/httpd/manual">
...
    Require ip 192.0.2.0/24
```

```
...
</Directory>
```

3. 重启 httpd 服务：

```
# systemctl restart httpd
```

验证步骤

1. 要显示 Apache HTTP 服务器手册，使用 Web 浏览器访问 **`http://host_name_or_IP_address/manual/`**

1.10. 使用模块

作为一个模块化应用程序，**httpd** 服务会包括多个 *Dynamic Shared Objects* (DSO)，它们可根据需要在运行时动态载入或卸载。这些模块位于 **`/usr/lib64/httpd/modules/`** 目录中。

1.10.1. 载入模块

要载入特定的 DSO 模块，使用 **LoadModule** 指令。请注意，由单独软件包提供的模块通常在 **`/etc/httpd/conf.modules.d/`** 目录中有自己的配置文件。

例 1.1. 载入 mod_ssl DSO

```
LoadModule ssl_module modules/mod_ssl.so
```

载入该模块后，重启 web 服务器以重新载入配置。如需了解有关如何重启 **httpd** 服务的更多信息，请参阅 [第 1.3 节“管理 httpd 服务”](#)。

1.10.2. 编写模块

要创建新的 DSO 模块，请确定已安装了 **httpd-devel** 软件包。要做到这一点，以 **root** 身份输入以下命令：

```
# yum install httpd-devel
```

这个软件包包含编译模块所需的文件、标头文件和 **APache eXtenSion (apxs)** 实用程序。

编写完成后，可以使用以下命令构建模块：

```
# apxs -i -a -c module_name.c
```

如果构建成功，您就可以像 **Apache HTTP 服务器** 分发的其他模块一样，载入该模块。

1.11. 从 NSS 数据库导出私钥和证书，以便在 APACHE WEB 服务器配置中使用它们

RHEL 8 不再为 Apache web 服务器提供 **mod_nss** 模块，红帽建议使用 **mod_ssl** 模块。如果您将私钥和证书存储在网络安全服务(NSS)数据库中，例如，因为您将 web 服务器从 RHEL 7 迁移到 RHEL 8，请按照以下步骤以 Privacy Enhanced Mail(PEM)格式提取密钥和证书。然后您可以使用 **mod_ssl** 配置中的文件，如

第 1.7 节 “在 Apache HTTP 服务器中配置 TLS 加密” 所述。

此流程假设 NSS 数据库存储在 `/etc/httpd/alias/` 中，并将导出的私钥和证书保存在 `/etc/pki/tls/` 目录中。

先决条件

- 私钥、证书和证书颁发机构(CA)证书存储在 NSS 数据库中。

流程

- 列出 NSS 数据库中的证书：

```
# certutil -d /etc/httpd/alias/ -L
Certificate Nickname           Trust Attributes
                          SSL,S/MIME,JAR/XPI

Example CA                     C,,
Example Server Certificate     u,u,u
```

在下一步中需要证书的别名。

- 要提取私钥,您必须临时将密钥导出到一个 PKCS #12 文件：
 - 使用与私钥关联的证书的别名，将密钥导出到一个 PKCS #12 文件：

```
# pk12util -o /etc/pki/tls/private/export.p12 -d /etc/httpd/alias/ -n "Example Server
Certificate"
Enter password for PKCS12 file: password
Re-enter password: password
pk12util: PKCS12 EXPORT SUCCESSFUL
```

请注意，您必须在 PKCS #12 文件中设置一个密码。下一步需要这个密码。

- 从 PKCS #12 文件中导出私钥：

```
# openssl pkcs12 -in /etc/pki/tls/private/export.p12 -out
/etc/pki/tls/private/server.key -nocerts -nodes
Enter Import Password: password
MAC verified OK
```

- 删除临时 PKCS #12 文件：

```
# rm /etc/pki/tls/private/export.p12
```

- 在 `/etc/pki/tls/private/server.key` 中设置权限以确保只有 `root` 用户可以访问此文件：

```
# chown root:root /etc/pki/tls/private/server.key
# chmod 0600 /etc/pki/tls/private/server.key
```

- 使用 NSS 数据库中的服务器证书的别名导出 CA 证书：

```
# certutil -d /etc/httpd/alias/ -L -n "Example Server Certificate" -a -o
/etc/pki/tls/certs/server.crt
```

5. 在 `/etc/pki/tls/certs/server.crt` 中设置权限以确保只有 `root` 用户可以访问此文件：

```
# chown root:root /etc/pki/tls/certs/server.crt
# chmod 0600 /etc/pki/tls/certs/server.crt
```

6. 使用 NSS 数据库中 CA 证书的别名导出 CA 证书：

```
# certutil -d /etc/httpd/alias/ -L -n "Example CA" -a -o /etc/pki/tls/certs/ca.crt
```

7. 根据 第 1.7 节 “在 Apache HTTP 服务器中配置 TLS 加密” 来配置 Apache web 服务器，并：

- 将 `SSLCertificateKeyFile` 参数设置为 `/etc/pki/tls/private/server.key`。
- 将 `SSLCertificateFile` 参数设置为 `/etc/pki/tls/certs/server.crt`。
- 将 `SSLCACertificateFile` 参数设置为 `/etc/pki/tls/certs/ca.crt`。

其它资源

- `certutil(1)` man page
- `pk12util(1)` man page
- `pkcs12(1ssl)` man page

1.12. 其它资源

- `httpd(8)` - `httpd` 服务的手册页，其中包含其命令行选项的完整列表。
- `httpd.service(8)` - `httpd.service` 单元文件的手册页,描述如何自定义和增强服务。
- `httpd.conf(5)` - `httpd` 配置的说明书页,描述 `httpd` 配置文件的结构和位置。
- `apachectl(8)` - Apache HTTP 服务器 控制接口的手册页。
- 有关如何在 Apache HTTP 服务器中配置 Kerberos 验证的详情，请参考为 [Apache httpd 操作使用 GSS-Proxy](#)。使用 Kerberos 是在 Apache HTTP 服务器中强制进行客户端授权的替代方法。

第 2 章 设置和配置 NGINX

NGINX 是一个高性能和模块化的服务器，可作为：

- Web 服务器
- 反向代理服务器
- 负载均衡器

这部分论述了如何在这些场景中使用 NGINX。

2.1. 安装并准备 NGINX

红帽使用 Application Streams 来提供不同的 NGINX 版本。本节描述了如何：

- 选择流并安装 NGINX
- 在防火墙中打开所需端口
- 启用并启动 **nginx** 服务

在默认配置中，NGINX 作为 Web 服务器在端口 **80** 上运行，并从 **/usr/share/nginx/html/** 目录中提供内容。

先决条件

- 已安装了 RHEL 8。
- 主机订阅了红帽客户门户网站。
- **firewalld** 服务已启用并启动。

流程

1. 显示可用的 NGINX 模块流：

```
# yum module list nginx
Red Hat Enterprise Linux 8 for x86_64 - AppStream (RPMs)
Name      Stream    Profiles    Summary
nginx     1.14 [d]   common [d]   nginx webserver
nginx     1.16      common [d]   nginx webserver
...

Hint: [d]efault, [e]nabled, [x]disabled, [i]nstalled
```

2. 如果要安装与默认流不同的流，请选择相关的流：

```
# yum module enable nginx:stream_version
```

3. 安装 **nginx** 软件包：

```
# yum install nginx
```

4. 打开 NGINX 应该在其防火墙中提供其服务的端口。例如：要在 **firewalld** 中为 HTTP（端口 80）和 HTTPS（端口 443）打开默认端口，请输入：

```
# firewall-cmd --permanent --add-port={80/tcp,443/tcp}
# firewall-cmd --reload
```

5. 在系统引导时自动启用 **nginx** 服务：

```
# systemctl enable nginx
```

6. （可选）启动 **nginx** 服务：

```
# systemctl start nginx
```

如果您不想使用默认配置，请跳过这一步，并在启动该服务前相应地配置 NGINX。

验证步骤

1. 使用 **yum** 实用程序验证是否安装了 **nginx** 软件包：

```
# yum list installed nginx
Installed Packages
nginx.x86_64 1:1.14.1-9.module+el8.0.0+4108+af250afe @rhel-8-for-x86_64-appstream-rpms
```

2. 确保在 **firewalld** 中打开了 NGINX 需要的端口：

```
# firewall-cmd --list-ports
80/tcp 443/tcp
```

3. 验证 **nginx** 服务是否已启用：

```
# systemctl is-enabled nginx
enabled
```

其它资源

- 有关 Subscription Manager 的详情,请查看 [使用和配置 Subscription Manager](#) 指南。
- 有关 Application Streams、模块和安装软件包的详情,请参考 [安装、管理和删除用户空间组件](#) 指南。
- 有关配置防火墙的详情,请查看 [安全网络](#) 指南。

2.2. 将 NGINX 配置为一个为不同域提供不同内容的 WEB 服务器

默认情况下，NGINX 作为 web 服务器，为与服务器的 IP 地址关联的所有域名提供相同的内容。此流程解释了如何配置 NGINX 来实现一下情况：

- 对于到 **example.com** 域的请求，提供 **/var/www/example.com/** 目录中的内容
- 对于到 **example.net** 域的请求，提供 **/var/www/example.net/** 目录中的内容

- 对于其他请求（例如：到服务器的 IP 地址或其他与服务器的 IP 地址关联的域的请求），提供 `/usr/share/nginx/html/` 目录中的内容

先决条件

- 如 [第 2.1 节 “安装并准备 NGINX”](#) 所述安装 NGINX。
- 客户端和网页服务器会将 **example.com** 和 **example.net** 域解析为 web 服务器的 IP 地址。请注意，您必须手动将这些条目添加到 DNS 服务器中。

流程

1. 编辑 `/etc/nginx/nginx.conf` 文件：

- 默认情况下，`/etc/nginx/nginx.conf` 文件已经包含 `catch-all` 配置。如果您已经从配置中删除了这部分，请将以下 **server** 块重新添加到 `/etc/nginx/nginx.conf` 文件中的 **http** 块中：

```
server {
    listen      80 default_server;
    listen      [::]:80 default_server;
    server_name _;
    root        /usr/share/nginx/html;
}
```

这些设置配置以下内容：

- **listen** 指令定义服务侦听的 IP 地址和端口。在这种情况下，NGINX 侦听所有 IPv4 和 IPv6 地址中的端口 **80**。**default_server** 参数表示 NGINX 使用这个 **server** 块作为与 IP 地址和端口匹配的请求的默认设置。
- **server_name** 参数定义此 **server** 块负责的主机名。将 **server_name** 设置为 `_` 将 NGINX 配置为接受这个 **server** 块的任何主机名。
- **root** 指令设定这个 **server** 块的 web 内容的路径。

- 为 **example.com** 域附加类似的 **server** 块到 **http** 块：

```
server {
    server_name example.com;
    root        /var/www/example.com/;
    access_log  /var/log/nginx/example.com/access.log;
    error_log   /var/log/nginx/example.com/error.log;
}
```

- **access_log** 指令为这个域定义一个单独的访问日志文件。
- **error_log** 指令为这个域定义一个单独的错误日志文件。

- 为 **example.net** 域附加类似的 **server** 块到 **http** 块：

```
server {
    server_name example.net;
    root        /var/www/example.net/;
    access_log  /var/log/nginx/example.net/access.log;
    error_log   /var/log/nginx/example.net/error.log;
}
```


2. 为这两个域创建根目录：

```
# mkdir -p /var/www/example.com/
# mkdir -p /var/www/example.net/
```

3. 在两个根目录中设置 `httpd_sys_content_t` 上下文：

```
# semanage fcontext -a -t httpd_sys_content_t "/var/www/example.com(/.*)?"
# restorecon -Rv /var/www/example.com/
# semanage fcontext -a -t httpd_sys_content_t "/var/www/example.net(/.*)?"
# restorecon -Rv /var/www/example.net/
```

这些命令在 `/var/www/example.com/` 和 `/var/www/example.net/` 目录上设置 `httpd_sys_content_t` 上下文。

请注意，您必须安装了 `policycoreutils-python-utils` 软件包才能运行 `restorecon` 命令。

4. 为这两个域创建日志目录：

```
# mkdir /var/log/nginx/example.com/
# mkdir /var/log/nginx/example.net/
```

5. 重启 `nginx` 服务：

```
# systemctl restart nginx
```

验证步骤

1. 在每个虚拟主机的文档 `root` 中创建不同的示例文件：

```
# echo "Content for example.com" > /var/www/example.com/index.html
# echo "Content for example.net" > /var/www/example.net/index.html
# echo "Catch All content" > /usr/share/nginx/html/index.html
```

2. 使用浏览器并连接到 `http://example.com`。web 服务器显示 `/var/www/example.com/index.html` 文件中的示例内容。
3. 使用浏览器并连接到 `http://example.net`。web 服务器显示 `/var/www/example.net/index.html` 文件中的示例内容。
4. 使用浏览器并连接到 `http://IP_address_of_the_server`。web 服务器显示 `/usr/share/nginx/html/index.html` 文件中的示例内容。

2.3. 在 NGINX WEB 服务器中添加 TLS 加密

这部分论述了如何在 `example.com` 域的 NGINX web 服务器中启用 TLS 加密。

先决条件

- 如 [第 2.1 节“安装并准备 NGINX”](#) 所述安装 NGINX。
- 私钥存储在 `/etc/pki/tls/private/example.com.key` 文件中。

有关创建私钥和证书签名请求(CSR)的详情,以及如何从证书颁发机构(CA)请求证书的详情,请查看您的 CA 文档。

- TLS 证书存储在 `/etc/pki/tls/certs/example.com.crt` 文件中。如果您使用不同的路径,请修改流程的对应步骤。
- CA 证书已附加到服务器的 TLS 证书文件中。
- 客户端和网页服务器会将服务器的主机名解析为 web 服务器的 IP 地址。
- 在本地防火墙中打开了端口 **443**。

流程

1. 编辑 `/etc/nginx/nginx.conf` 文件, 并将以下 **server** 块添加到配置的 **http** 块中 :

```
server {
    listen      443 ssl;
    server_name example.com;
    root        /usr/share/nginx/html;
    ssl_certificate /etc/pki/tls/certs/example.com.crt;
    ssl_certificate_key /etc/pki/tls/private/example.com.key;
}
```

2. 出于安全考虑, 请配置为只有 **root** 用户可以访问私钥文件 :

```
# chown root:root /etc/pki/tls/private/example.com.key
# chmod 600 /etc/pki/tls/private/example.com.key
```



警告

如果私钥被设置为可以被未授权的用户访问, 则需要撤销证书, 然后再创建一个新私钥并请求一个新证书。否则, TLS 连接就不再安全。

3. 重启 **nginx** 服务 :

```
# systemctl restart nginx
```

验证步骤

- 使用浏览器并连接到 **https://example.com**

2.4. 将 NGINX 配置为 HTTP 流量的反向代理

您可以将 NGINX web 服务器配置为作为 HTTP 流量的反向代理。例如,您可以使用此功能将请求转发到远程服务器上的特定子目录。从客户端视角来说,客户端从它访问的主机中加载内容。但是 NGINX 会从远程服务器加载实际内容并将其转发给客户端。

此流程解释了如何将到 web 服务器上的 `/example` 目录的流量转发到 URL **https://example.com**。

先决条件

- 如 第 2.1 节 “安装并准备 NGINX” 所述安装 NGINX。
- 可选：反向代理上启用了 TLS 加密。

流程

1. 编辑 `/etc/nginx/nginx.conf` 文件并在 **server** 块中添加应提供反向代理的设置：

```
location /example {
    proxy_pass https://example.com;
}
```

location 块定义 NGINX 将 `/example` 目录中的所有请求传递到 `https://example.com`。

2. 将 `httpd_can_network_connect` SELinux 布尔值参数设置为 **1** 以配置 SELinux 允许 NGINX 转发流量：

```
# setsebool -P httpd_can_network_connect 1
```

3. 重启 **nginx** 服务：

```
# systemctl restart nginx
```

验证步骤

- 使用浏览器并连接到 `http://host_name/example`, 显示 `https://example.com` 的内容。

2.5. 将 NGINX 配置为 HTTP 负载均衡器

您可以使用 NGINX 反向代理功能进行负载均衡流量。这个步骤描述了如何将 NGINX 配置为 HTTP 负载均衡器。它会根据服务器上的活跃连接的数量，将请求发送到不同服务器（发送到活跃连接数量最小的服务器）。如果两个服务器都不可用，这个过程还定义了第三个主机用于回退。

先决条件

- 如 第 2.1 节 “安装并准备 NGINX” 所述安装 NGINX。

流程

1. 编辑 `/etc/nginx/nginx.conf` 文件并添加以下设置：

```
http {
    upstream backend {
        least_conn;
        server server1.example.com;
        server server2.example.com;
        server server3.example.com backup;
    }

    server {
        location / {
            proxy_pass http://backend;
        }
    }
}
```

```

    }
  }
}

```

名为 **backend** 的主机组中的 **least_conn** 指令定义 NGINX 将请求发送到 **server1.example.com** 或 **server2.example.com**，具体取决于哪个主机具有最少的活跃连接。NGINX 仅在其他两个主机不可用时使用 **server3.example.com** 作为备份。

当将 **proxy_pass** 指令设置为 **http://backend** 时，NGINX 作为反向代理，并使用 **backend** 主机组根据这个组的设置发布请求。

您还可以指定其他方法，而不使用 **least_conn** 负载均衡方法：

- 不指定方法，使用轮询的方式在服务器间平均分发请求。
- **ip_hash** 根据从 IPv4 地址的前三个 octets 或客户端整个 IPv6 地址计算的哈希值将来自一个客户端地址的请求发送到同一服务器。
- **hash** 根据用户定义的关键字确定服务器，关键字可以是字符串、变量或两者的组合。**consistent** 参数配置 NGINX 根据用户定义的散列键值将请求分发到所有服务器中。
- **random** 将请求发送到随机选择的服务器。

2. 重启 nginx 服务：

```
# systemctl restart nginx
```

2.6. 其它资源

- 有关官方 NGINX 文档，请参考 <https://nginx.org/en/docs/>。请注意,红帽并不维护本文档,它可能不适用于您安装的 NGINX 版本。

第 3 章 使用 SAMBA 作为服务器

Samba 在 Red Hat Enterprise Linux 中实现 Server Message Block(SMB)协议。SMB 协议用于访问服务器上的资源,如文件共享和共享打印机。另外,Samba 实现了 Microsoft Windows 使用的分布式计算环境远程过程调用(DCE RPC)协议。

您可以以以下方式运行 Samba :

- 一个 Active Directory(AD)或 NT4 域成员
- 独立服务器
- NT4 主域控制器(PDC)或 Backup Domain Controller(BDC)



注意

红帽仅在 Windows 版本支持 NT4 域的现有安装中支持 PDC 和 BDC 模式。红帽建议不要设置新的 Samba NT4 域,因为 Microsoft 操作系统稍后于 Windows 7 和 Windows Server 2008 R2 不支持 NT4 域。

红帽不支持将 Samba 作为 AD 域控制器(DC)运行。

独立于安装模式,您还可以选择性地实现共享目录和打印机。这可让 Samba 充当文件和打印服务器。

3.1. 了解不同的 SAMBA 服务和模式

这部分论述了 Samba 中包含的不同服务以及您可以配置的不同模式。

3.1.1. Samba 服务

Samba 提供以下服务 :

smbd

此服务使用 SMB 协议提供文件共享和打印服务。另外,该服务负责资源锁定和验证连接用户。**smbd** 服务启动并停止 **smbd** 守护进程。
要使用 **smbd** 服务,请安装 **samba** 软件包。

nmbd

此服务通过 IPv4 协议使用 NetBIOS 提供主机名和 IP 解析。除了名称解析外,**nmbd** 服务还可以浏览 SMB 网络来定位域、工作组、主机、文件共享和打印机。对于这些信息,服务可以直接向广播客户端报告这些信息,或者将其转发到本地或主浏览器。**nmbd** 服务启动并停止 **nmbd** 守护进程。
请注意,现代 SMB 网络使用 DNS 解析客户端和 IP 地址。

要使用 **nmbd** 服务,请安装 **samba** 软件包。

winbindd

这个服务为 Name Service Switch(NSS)提供了一个接口,用于使用 AD 或 NT4 域用户和本地系统中的组。例如,这可让域用户到 Samba 服务器托管的服务或其他本地服务进行身份验证。**winbindd** 服务启动并停止 **winbindd** 守护进程。
如果您将 Samba 设置为域成员,**winbindd** 必须在 **smbd** 服务前启动。否则,域用户和组对本地系统不可用。

要使用 **winbindd** 服务,请安装 **samba-winbind** 软件包。



重要

红帽只支持将 Samba 作为具有 **winbindd** 服务的服务器运行,为本地系统提供域用户和组。由于某些限制,如缺少 Windows 访问控制列表(ACL)支持和 NT LAN Manager(NTLM)回退,SSSD 不被支持。

3.1.2. Samba 安全服务

`/etc/samba/smb.conf` 文件中的 **[global]** 部分中的 **security** 参数管理 Samba 如何验证连接到该服务的用户。根据您在其中安装 Samba 的模式, 参数必须设为不同的值:

在 AD 域成员中设置 **security = ads**

在这个模式中, Samba 使用 Kerberos 来验证 AD 用户。

有关将 Samba 设置为域成员的详情, 请参考 [第 3.5 节 “将 Samba 设置为 AD 域成员服务器”](#)。

在独立服务器中设置 **security = user**

在这个模式中, Samba 使用本地数据库验证连接用户。

有关将 Samba 设置为独立服务器的详情请参考 [第 3.3 节 “将 Samba 设置为独立服务器”](#)。

在 NT4 PDC 或 BDC 中设置 **security = user**

在这个模式下,Samba 将用户验证到本地或 LDAP 数据库。

在 NT4 域成员中设置 **security = domain**

在这个模式中,Samba 验证用户连接到 NT4 PDC 或 BDC。您不能在 AD 域成员中使用这个模式。

有关将 Samba 设置为域成员的详情, 请参考 [第 3.5 节 “将 Samba 设置为 AD 域成员服务器”](#)。

其它资源

- 请参阅 **smb.conf(5)** man page 中的 **security** 参数描述。

3.1.3. Samba 服务和 Samba 客户端工具加载并重新载入其配置的情况

下面描述了 Samba 服务和工具加载并重新载入其配置:

- Samba 服务在以下情况下重新载入其配置:
 - 每 3 分钟自动进行
 - 手动请求时, 例如运行 **smbcontrol all reload-config** 命令时。
- Samba 客户端实用程序仅在启动时读取其配置。

请注意, 某些参数 (如 **security**) 需要重启 **smb** 服务才能生效, 重新载入并不会使其生效。

其它资源

- **smb.conf(5)** man page 中的 **How configuration changes are applied** 部分
- **smbd(8)**、**nmbd(8)** 和 **winbindd(8)** man page

3.1.4. 以安全的方式编辑 Samba 配置

Samba 服务每 3 分钟自动重新载入其配置。这个步骤描述了，如何在编辑 Samba 配置时，在使用 **testparm** 验证配置前，防止服务重新载入更改的方式。

先决条件

- 已安装 Samba。

流程

1. 创建 **/etc/samba/smb.conf** 文件的副本：

```
# cp /etc/samba/smb.conf /etc/samba/samba.conf.copy
```

2. 编辑复制的文件并进行必要的更改。
3. 验证 **/etc/samba/samba.conf.copy** 文件中的配置：

```
# testparm -s /etc/samba/samba.conf.copy
```

如果 **testparm** 报告错误,请修复它们并再次运行命令。

4. 使用新配置覆盖 **/etc/samba/smb.conf** 文件：

```
# mv /etc/samba/samba.conf.copy /etc/samba/smb.conf
```

5. 等待 Samba 服务自动重新载入其配置或手动重新载入配置：

```
# smbcontrol all reload-config
```

其它资源

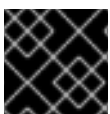
- [第 3.1.3 节 “Samba 服务和 Samba 客户端工具加载并重新载入其配置的情况”](#)

3.2. 验证 SAMBA 配置

红帽建议您在每次更新 **/etc/samba/smb.conf** 文件时验证 Samba 配置。本节提供有关此问题的详细信息。

3.2.1. 使用 testparm 工具验证 smb.conf 文件

testparm 工具会验证 **/etc/samba/smb.conf** 文件中的 Samba 配置是否正确。该工具会检测到无效的参数和值,但也不正确的设置,比如用于 ID 映射的设置。如果 **testparm** 报告没有问题, Samba 服务将成功载入 **/etc/samba/smb.conf** 文件。请注意, **testparm** 无法验证配置的服务是否可用或按预期工作。



重要

红帽建议您在每次修改此文件后使用 **testparm** 来验证 **/etc/samba/smb.conf** 文件。

先决条件

- 已安装 Samba。
- **/etc/samba/smb.conf** 文件已退出。

流程

1. 以 **root** 用户身份运行 **testparm** 工具：

```
# testparm
Load smb config files from /etc/samba/smb.conf
rlimit_max: increasing rlimit_max (1024) to minimum Windows limit (16384)
Unknown parameter encountered: "log level"
Processing section "[example_share]"
Loaded services file OK.
ERROR: The idmap range for the domain * (tdb) overlaps with the range of DOMAIN (ad)!

Server role: ROLE_DOMAIN_MEMBER

Press enter to see a dump of your service definitions

# Global parameters
[global]
...

[example_share]
...
```

前面的示例输出会报告不存在的参数以及不正确的 ID 映射配置。

2. 如果 **testparm** 在配置中报告不正确的参数、值或其他错误，请修复问题并再次运行实用程序。

3.3. 将 SAMBA 设置为独立服务器

您可以将 Samba 设置为不是域成员的服务器。在这个安装模式中,Samba 将用户验证到本地数据库,而不是一个中央 DC。另外，您可以启用客户机访问，允许用户在没有身份验证的情况下连接到一个或多个服务。

3.3.1. 为独立服务器设置服务器配置

这部分论述了如何为 Samba 独立服务器设置服务器配置。

流程

1. 安装 **samba** 软件包：

```
# yum install samba
```

2. 编辑 **/etc/samba/smb.conf** 文件并设置以下参数：

```
[global]
workgroup = Example-WG
netbios name = Server
security = user

log file = /var/log/samba/%m.log
log level = 1
```


此配置在 **Example-WG** 工作组中定义了一个名为 **Server** 的独立服务器。另外,此配置启用了最小级别的日志(1),日志文件将存储在 **/var/log/samba/** 目录中。Samba 将在 **log file** 参数中将 **%m** 宏扩展到连接客户端的 NetBIOS 名称。这可为每个客户端启用独立的日志文件。

3. (可选) 配置文件或打印机共享。请参阅：

- [第 3.7 节 “设置使用 POSIX ACL 的 Samba 文件共享”](#)
- [第 3.9 节 “设置使用 Windows ACL 的共享”](#)
- [第 3.15 节 “将 Samba 设置为打印服务器”](#)

4. 验证 **/etc/samba/smb.conf** 文件：

```
# testparm
```

5. 如果您设置了需要身份验证的共享,请创建用户帐户。
详情请查看 [第 3.3.2 节 “创建并启用本地用户帐户”](#)。

6. 打开所需端口并使用 **firewall-cmd** 实用程序重新载入防火墙配置：

```
# firewall-cmd --permanent --add-port={139/tcp,445/tcp}
# firewall-cmd --reload
```

7. 启用并启动 **smb** 服务：

```
# systemctl enable --now smb
```

其它资源

- 有关该流程中使用的参数的详情,请查看 **smb.conf(5)** man page 中的参数描述。

3.3.2. 创建并启用本地用户帐户

要让用户在连接到共享时进行验证,您必须在操作系统和 Samba 数据库的 Samba 主机上创建帐户。Samba 要求操作系统帐户验证文件系统对象和 Samba 帐户中的访问控制列表(ACL)验证连接用户。

如果您使用 **passdb backend = tdbsam** 默认设置,Samba 会在 **/var/lib/samba/private/passdb.tdb** 数据库中存储用户帐户。

本节中的步骤论述了如何创建名为 **example** 的本地 Samba 用户。

先决条件

- Samba 被配置为独立服务器。

流程

1. 创建操作系统帐户：

```
# useradd -M -s /sbin/nologin example
```

这个命令在没有创建主目录的情况下添加 **example** 帐户。如果帐户只用于 Samba 验证,请将 **/sbin/nologin** 命令分配为 shell,以防止帐户在本地登录。

2. 为操作系统帐户设置密码以启用它：

```
# passwd example
Enter new UNIX password: password
Retype new UNIX password: password
passwd: password updated successfully
```

Samba 不会使用操作系统帐户中的密码集进行身份验证。然而，您需要设置密码才能启用帐户。如果一个帐户被禁用，当这个用户连接时，Samba 会拒绝访问。

3. 将用户添加到 Samba 数据库，并为帐户设置密码：

```
# smbpasswd -a example
New SMB password: password
Retype new SMB password: password
Added user example.
```

当使用此帐户连接到 Samba 共享时，使用此密码进行验证。

4. 启用 Samba 帐户：

```
# smbpasswd -e example
Enabled user example.
```

3.4. 了解并配置 SAMBA ID 映射

Windows 域根据唯一的安全标识符(SID)区分用户和组。但是，Linux 需要为每个用户和组群有唯一的 UID 和 GID。如果您以域成员身份运行 Samba, **winbindd** 服务将负责向操作系统提供有关域用户和组的信息。

要启用 **winbindd** 服务为用户和组为 Linux 提供唯一 ID，您必须在 **/etc/samba/smb.conf** 文件中配置 ID 映射：

- 本地数据库（默认域）
- Samba 服务器所属的 AD 或 NT4 域
- 每个用户必须能够访问这个 Samba 服务器上的资源的可信域

Samba 为特定配置提供不同的 ID 映射后端。最常用的后端是：

后端	使用案例
tdb	仅限 * 默认域
ad	仅限 AD 域
rid	AD 和 NT4 域
autorid	AD、NT4 和 * 默认域

3.4.1. 规划 Samba ID 范围

无论您在 AD 中存储 Linux UID 和 GID,还是将 Samba 配置为生成它们,每个域配置都需要一个唯一的 ID 范围,不得与任何其他域重叠。



警告

如果您设置了重叠 ID 范围, Samba 无法正常工作。

例 3.1. 唯一的 ID 范围

以下显示了默认(*)、**AD-DOM** 和 **TRUST-DOM** 域的非扩展 ID 映射范围。

```
[global]
...
idmap config * : backend = tdb
idmap config * : range = 10000-999999

idmap config AD-DOM:backend = rid
idmap config AD-DOM:range = 2000000-2999999

idmap config TRUST-DOM:backend = rid
idmap config TRUST-DOM:range = 4000000-4999999
```



重要

每个域只能分配一个范围。因此,在域范围之间有足够的空间。这可让您在域扩展后扩展范围。

如果您之后为某个域分配了不同的范围,则之前这些用户和组创建的文件和目录的所有权将会丢失。

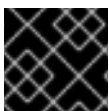
3.4.2. * 默认域

在域环境中,您可以为以下每个情况添加一个 ID 映射配置:

- Samba 服务器所属的域
- 每个可以访问 Samba 服务器的可信域

但是,对于所有其他对象, Samba 会从默认域分配 ID。这包括:

- 本地 Samba 用户和组
- Samba 内置帐户和组,如 **BUILTIN\Administrators**



重要

您必须配置默认域,如本节所述,以便 Samba 正常工作。

默认域后端必须是可写的,才能永久存储分配的 ID。

对于默认域,您可以使用以下后端之一：

tdb

当您将默认域配置为使用 **tdb** 后端时,请设置一个足够大、足以包含未来创建的并且不是定义的域 ID 映射配置一部分的 ID 范围。

例如,在 `/etc/samba/smb.conf` 文件中的 **[global]** 部分设置以下内容：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

详情请查看 [第 3.4.3 节“使用 tdb ID 映射后端”](#)。

autorid

当您将默认域配置为使用 **autorid** 后端时,为域添加其他 ID 映射配置是可选的。

例如,在 `/etc/samba/smb.conf` 文件中的 **[global]** 部分设置以下内容：

```
idmap config * : backend = autorid
idmap config * : range = 10000-999999
```

详情请查看 [第 3.4.6 节“使用自动 ID 映射后端”](#)。

3.4.3. 使用 tdb ID 映射后端

winbindd 服务默认使用可写入 **tdb** ID 映射后端来存储安全标识符(SID)、UID 和 GID 映射表。这包括本地用户、组和内置主体。

仅将此后端用于 * 默认域。例如：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

其它资源

- [第 3.4.2 节“* 默认域”](#)。

3.4.4. 使用 ad ID 映射后端

这部分论述了如何配置 Samba AD 成员使用 **ad** ID 映射后端。

ad ID 映射后端实施只读 API,从 AD 读取帐户和组信息。它具有以下优点：

- 所有用户和组群设置都集中存储在 AD 中。
- 使用这个后端的所有 Samba 服务器中的用户和组群 ID 是一致的。
- ID 不会存储在本地数据库中（本地数据库可能会被损坏），因此文件所有者不会丢失。



注意

ad ID 映射后端不支持具有单向信任的 Active Directory 域。如果您在带有单向信任的 Active Directory 中配置域成员,请使用以下 ID 映射后端之一：**tdb**、**rid** 或 **autorid**。

后端从 AD 读取以下属性：

AD 属性名称	对象类型	映射到
sAMAccountName	用户和组群	用户或组群名称,具体取决于对象
uidNumber	用户	用户 ID (UID)
gidNumber	组	组 ID (GID)
loginShell ^[a]	用户	用户 shell 的路径
unixHomeDirectory ^[a]	用户	用户主目录的路径
primaryGroupID ^[b]	用户	主组群 ID
^[a] Samba 只在设置了 idmap config DOMAIN:unix_nss_info = yes 时读取此属性。		
^[b] Samba 只在设置了 idmap config DOMAIN:unix_primary_group = yes 时读取此属性。		

先决条件

- 用户和组必须在 AD 中设置唯一的 ID，且 ID 必须在 **/etc/samba/smb.conf** 文件中配置的范围。ID 不在范围以外的对象在 Samba 服务器中不可用。
- 用户和组必须在 AD 中设置所有必需的属性。如果缺少所需的属性，该用户或组将无法在 Samba 服务器中可用。所需的属性取决于您的配置。
- 已安装 Samba。
- Samba 配置（除了 ID 映射）位于 **/etc/samba/smb.conf** 文件中。

流程

1. 编辑 **/etc/samba/smb.conf** 文件中的 **[global]** 部分：
 - a. 如果默认域(*)不存在,添加 ID 映射配置。例如：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

- b. 为 AD 域启用 **ad** ID 映射后端：

```
idmap config DOMAIN : backend = ad
```

- c. 设置分配给 AD 域中用户和组的 ID 范围。例如：

```
idmap config DOMAIN : range = 2000000-2999999
```



重要

范围不得与这个服务器上的任何其他域配置重叠。此外，范围必须足够大，以便包含将来分配的所有 ID。详情请查看 [第 3.4.1 节“规划 Samba ID 范围”](#)。

- d. 在从 AD 读取属性时,设置 Samba 使用 [RFC 2307](#) 模式：

```
idmap config DOMAIN : schema_mode = rfc2307
```

- e. 要启用 Samba 从对应的 AD 属性读取登录 shell 和用户主目录的路径,请设置：

```
idmap config DOMAIN : unix_nss_info = yes
```

或者,您可以设置一个统一的域范围主目录路径和登录 shell,适用于所有用户。例如：

```
template shell = /bin/bash
template homedir = /home/%U
```

- f. 默认情况下，Samba 使用用户对象的 **primaryGroupID** 属性作为 Linux 中用户的主组群。或者，您可以将 Samba 配置为使用 **gidNumber** 属性中设置的值：

```
idmap config DOMAIN : unix_primary_group = yes
```

2. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

3. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

其它资源

- [第 3.4.2 节“* 默认域”](#)
- 有关该流程中使用的参数的详情，请查看 **smb.conf(5)** 和 **idmap_ad(8)** man page。
- 有关变量替换的详情，请查看 **smb.conf(5)** man page 中的 **VARIABLE SUBSTITUTIONS** 部分。

3.4.5. 使用网格 ID 映射后端

这部分论述了如何配置 Samba 域成员以使用 **rid** ID 映射后端。

Samba 可以使用 Windows SID 的相对标识符(RID)在 Red Hat Enterprise Linux 上生成 ID。



注意

RID 是 SID 的最后部分。例如，如果用户的 SID 是 **S-1-5-21-5421822485-1151247151-421485315-30014**，则 **30014** 是对应的 RID。

rid ID 映射后端实施只读 API，以根据 AD 和 NT4 域的算法映射方案计算帐户和组信息。当您配置后端时，必须在 **idmap config DOMAIN : range** 参数中设置最低和最高的 RID。Samba 不会映射比这个参数中设置低或更高 RID 的用户或组。



重要

作为只读后端，**rid** 无法分配新 ID，比如 **BUILTIN** 组。因此，不要将这个后端用于 * 默认域。

使用网格后端的好处

- 所有在配置范围内具有 RID 的域用户和组都会自动在域成员中可用。
- 您不需要手动分配 ID、主目录和登录 shell。

使用网格后端的缺陷

- 所有域用户可以获得相同的登录 shell 和主目录。但是，您可以使用变量。
- 如果所有用户都使用相同的 ID 范围设置的 **rid** 后端，用户和组群 ID 只会跨 Samba 域成员相同。
- 您不能阻止单独的用户或组在域成员中可用。只有超出配置范围以外的用户和组才会包括。
- 根据 **winbindd** 服务用来计算 ID 的公式，如果不同域中的对象有相同的 RID，则在多域环境中可能会发生重复的 ID。

先决条件

- 已安装 Samba。
- Samba 配置（除了 ID 映射）位于 **/etc/samba/smb.conf** 文件中。

流程

1. 编辑 **/etc/samba/smb.conf** 文件中的 **[global]** 部分：

a. 如果默认域(*)不存在,添加 ID 映射配置。例如：

```
idmap config * : backend = tdb
idmap config * : range = 10000-999999
```

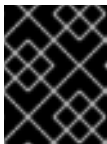
b. 为域启用 **rid** ID 映射后端：

```
idmap config DOMAIN : backend = rid
```

c. 设置一个足够大的范围,使其包含以后分配的所有 RID。例如：

```
idmap config DOMAIN : range = 2000000-2999999
```

Samba 忽略这个域中 RID 不在范围范围内的用户和组。

**重要**

范围不得与这个服务器上的任何其他域配置重叠。详情请查看 [第 3.4.1 节“规划 Samba ID 范围”](#)。

- d. 设置分配给所有映射用户的 shell 和主目录路径。例如：

```
template shell = /bin/bash
template homedir = /home/%U
```

2. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

3. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

其它资源

- [第 3.4.2 节“* 默认域”](#)
- 有关变量替换的详情，请查看 `smb.conf(5)` man page 中的 **VARIABLE SUBSTITUTIONS** 部分。
- 有关 Samba 如何从 RID 计算本地 ID 的详情，请查看 `idmap_rid(8)` man page。

3.4.6. 使用自动 ID 映射后端

这部分论述了如何配置 Samba 域成员以使用 **autorid** ID 映射后端。

autorid 后端的工作方式与 **rid** ID 映射后端类似,但可以自动为不同域分配 ID。这可让您在以下情况下使用 **autorid** 后端：

- 只适用于 * 默认域
- 对于 * 默认域和其他域，无需为每个附加域创建 ID 映射配置
- 只适用于特定域

**注意**

如果您将 **autorid** 用于默认域，为域添加额外的 ID 映射配置是可选的。

本节的部分内容来自在 Samba Wiki 中发布的 [idmap config autorid](#) 文档。许可证：[CC BY 4.0](#)。作者和贡献者：请参阅 Wiki 页中的 [历史记录](#) 标签。

使用自动扩展后端的好处

- 所有在配置范围内计算 UID 和 GID 的域用户和组都会在域成员中自动可用。
- 您不需要手动分配 ID、主目录和登录 shell。
- 没有重复的 ID，即使多域环境中的多个对象有相同的 RID。

缺陷

- 在 Samba 域成员中用户和组群 ID 不相同。
- 所有域用户可以获得相同的登录 shell 和主目录。但是，您可以使用变量。
- 您不能阻止单独的用户或组在域成员中可用。只有计算 UID 或 GID 不在配置范围内的用户和组才会包括。

先决条件

- 已安装 Samba。
- Samba 配置（除了 ID 映射）位于 **/etc/samba/smb.conf** 文件中。

流程

1. 编辑 **/etc/samba/smb.conf** 文件中的 **[global]** 部分：

- a. 为 * 默认域启用 **autorid** ID 映射后端：

```
idmap config * : backend = autorid
```

- b. 设置一个足够大的范围来为所有现有和将来的对象分配 ID。例如：

```
idmap config * : range = 10000-999999
```

Samba 忽略在此域中计算 ID 不在范围范围内的用户和组。



警告

设置范围并开始使用 Samba 后，您只能增加范围的上限。对范围的任何其他变化都可能会导致分配新的 ID，从而会丢失文件的所有者信息。

- c. 另外，还可设置范围大小。例如：

```
idmap config * : rangesize = 200000
```

Samba 为每个域的对象分配这个数量的持续 ID,直到使用 **idmap config * : range** 参数中设置的范围中的所有 ID。

- d. 设置分配给所有映射用户的 shell 和主目录路径。例如：

```
template shell = /bin/bash
template homedir = /home/%U
```

- e. 另外,还可为域添加额外的 ID 映射配置。如果没有可用的独立域配置,Samba 会使用之前配置的 * 默认域中的 **autorid** 后端设置计算 ID。



重要

如果您为单个域配置额外的后端，则所有 ID 映射配置的范围不得互相重叠。详情请查看 [第 3.4.1 节“规划 Samba ID 范围”](#)。

2. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

3. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

其它资源

- 有关后端计算 ID 的详情,请查看 `idmap_au torid(8)` man page 中的 **THE MAPPING FORMULAS** 部分。
- 有关使用 `idmap config rangesize` 参数的详情, 请查看 `idmap_au torid(8)` man page 中的 **rangesize** 参数描述。
- 有关变量替换的详情, 请查看 `smb.conf(5)` man page 中的 **VARIABLE SUBSTITUTIONS** 部分。

3.5. 将 SAMBA 设置为 AD 域成员服务器

如果您正在运行 AD 或 NT4 域, 请使用 Samba 将 Red Hat Enterprise Linux 服务器添加为域的成员, 以便可以：

- 访问其他域成员上的域资源
- 向本地服务验证域用户, 例如 `sshd`
- 托管在服务器上的共享目录和打印机, 以充当文件和打印服务器

3.5.1. 将 RHEL 系统添加到 AD 域中

这部分论述了如何使用 `realmd` 配置 Samba Winbind 将 Red Hat Enterprise Linux 系统添加到 AD 域中。

流程

1. 如果您的 AD 需要弃用的 RC4 加密类型进行 Kerberos 验证, 请在 RHEL 中启用对这些密码的支持：

```
# update-crypto-policies --set DEFAULT:AD-SUPPORT
```

2. 安装以下软件包：

```
# yum install realmd oddjob-mkhomedir oddjob samba-winbind-clients \ samba-winbind samba-common-tools samba-winbind-krb5-locator
```

3. 要在域成员中共享目录或打印机, 请安装 `samba` 软件包：

```
# yum install samba
```

4. 备份现有的 `/etc/samba/smb.conf` Samba 配置文件：

```
# mv /etc/samba/smb.conf /etc/samba/smb.conf.bak
```

5. 加入域。例如，要加入名为 `ad.example.com` 的域：

```
# realm join --membership-software=samba --client-software=winbind ad.example.com
```

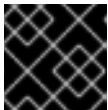
使用前面的命令，`realm` 工具会自动：

- 为 `ad.example.com` 域中的成员资格创建一个 `/etc/samba/smb.conf` 文件
 - 在 `/etc/nsswitch.conf` 文件中为用户和组群查询添加 `winbind` 模块
 - 更新 `/etc/pam.d/` 目录中的可插拔验证模块(PAM)配置文件
 - 启动 `winbind` 服务并启用服务在系统引导时启动
6. 另外，还可在 `/etc/samba/smb.conf` 文件中设置备选 ID 映射后端或自定义 ID 映射设置。详情请查看 [第 3.4 节“了解并配置 Samba ID 映射”](#)。
 7. 验证 `winbind` 服务是否正在运行：

```
# systemctl status winbind
```

```
...
```

```
Active: active (running) since Tue 2018-11-06 19:10:40 CET; 15s ago
```



重要

要启用 Samba 查询域用户和组群信息，必须在启动 `smb` 前运行 `winbind` 服务。

8. 如果您安装 `samba` 软件包来共享目录和打印机，请启用并启动 `smb` 服务：

```
# systemctl enable --now smb
```

9. 另外，如果您要向 Active Directory 验证本地登录，请启用 `winbind_krb5_localauth` 插件。请参阅 [第 3.5.2 节“使用 MIT Kerberos 的本地授权插件”](#)。

验证步骤

1. 显示 AD 用户的详情，如 AD 域中的 AD 管理员帐户：

```
# getent passwd "AD\administrator"
```

```
AD\administrator:*:10000:10000::/home/administrator@AD:/bin/bash
```

2. 查询 AD 域中的域用户组成员：

```
# getent group "AD\Domain Users"
```

```
AD\domain users:x:10000:user1,user2
```

3. 另外，还可在设置文件和目录权限时验证您可以使用域用户和组。例如，将 `/srv/samba/example.txt` 文件的拥有者设置为 `AD\administrator`，组为 `AD\Domain Users`：

```
# chown "AD\administrator":"AD\Domain Users" /srv/samba/example.txt
```

4. 验证 Kerberos 验证是否如预期正常工作：

- a. 在 AD 域成员中，为 `administrator@AD.EXAMPLE.COM` 主体获取一个 ticket：

```
# kinit administrator@AD.EXAMPLE.COM
```

- b. 显示缓存的 Kerberos ticket：

```
# klist
Ticket cache: KCM:0
Default principal: administrator@AD.EXAMPLE.COM

Valid starting    Expires          Service principal
01.11.2018 10:00:00 01.11.2018 20:00:00
krbtgt/AD.EXAMPLE.COM@AD.EXAMPLE.COM
renew until 08.11.2018 05:00:00
```

5. 显示可用域：

```
# wbinfo --all-domains
BUILTIN
SAMBA-SERVER
AD
```

其它资源

- 如果您不想使用弃用的 RC4 密码，可以在 AD 中启用 AES 加密类型。请参阅 [第 3.6.2 节“使用 GPO 在 Active Directory 中启用 AES 加密类型”](#)。请注意，这可能会对您的 AD 中的其他服务产生影响。
- 有关 `realm` 工具程序的详情，请查看 `realm(8)` man page。

3.5.2. 使用 MIT Kerberos 的本地授权插件

`winbind` 服务为域成员提供 Active Directory 用户。在某些情况下，管理员希望让域用户向本地服务（如 SSH 服务器）进行身份验证，这些服务在域成员中运行。当使用 Kerberos 验证域用户时，启用 `winbind_krb5_localauth` 插件通过 `winbind` 服务把 Kerberos 主体正确映射到 Active Directory 帐户。

例如，如果 Active Directory 用户的 `sAMAccountName` 属性被设置为 `EXAMPLE`，且用户试图使用用户名小写记录，Kerberos 会在大写中返回用户名。因此，条目与身份验证不匹配。

使用 `winbind_krb5_localauth` 插件，帐户名称会被正确映射。请注意，这只适用于 GSSAPI 验证，不适用于获取初始发布的票据(TGT)。

先决条件

- Samba 配置为 Active Directory 的成员。
- Red Hat Enterprise Linux 对 Active Directory 进行身份验证。

- **winbind** 服务正在运行。

流程

编辑 `/etc/krb5.conf` 文件并添加以下部分：

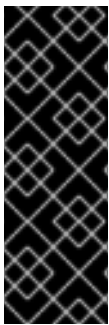
```
[plugins]
localauth = {
    module = winbind:/usr/lib64/samba/krb5/winbind_krb5_localauth.so
    enable_only = winbind
}
```

其它资源

- 请查看 **winbind_krb5_localauth(8)** man page。

3.6. 在 IDM 域成员中设置 SAMBA

这部分论述了如何在加入 Red Hat Identity Management(IdM)域的主机上设置 Samba。IdM 中的用户以及可信 Active Directory(AD)域中的用户可以访问 Samba 提供的共享和打印机服务。



重要

在 IdM 域成员中使用 Samba 是一个不受支持的技术预览功能,它包含了某些限制。例如,因为 IdM 信任控制器不支持全局目录服务,AD-enrolled Windows 主机无法在 Windows 中找到 IdM 用户和组。另外,IdM Trust Controller 不支持使用分布式计算环境/远程过程调用 (DCE/RPC) 协议解析 IdM 组。因此,AD 用户只能访问 IdM 客户端的 Samba 共享和打印机。

我们鼓励在 IdM 域成员中部署 Samba 的用户向红帽提供反馈意见。

先决条件

- 主机作为 IdM 域的客户端加入。
- IdM 服务器和客户端必须在 RHEL 8.1 或更高版本中运行。

3.6.1. 准备 IdM 域以便在域成员中安装 Samba

在您可以使用 AD 建立信任以及想要在 IdM 客户端中设置 Samba 之前,您必须使用 IdM 服务器上的 **ipa-adtrust-install** 工具准备 IdM 域。但是,即使这两种情况都适用,您也必须在 IdM 服务器中运行一次 **ipa-adtrust-install**。

先决条件

- 已安装 IdM。

流程

1. 安装所需的软件包：

```
[root@ipaserver ~]# yum install ipa-server ipa-server-trust-ad samba-client
```

2. 以 IdM 管理用户身份进行身份验证：

```
[root@ipaserver ~]# kinit admin
```

3. 运行 **ipa-adtrust-install** 工具程序：

```
[root@ipaserver ~]# ipa-adtrust-install
```

如果 IdM 安装了集成的 DNS 服务器，则会自动创建 DNS 服务记录。

如果 IdM 是在没有集成 DNS 服务器的情况下安装的，**ipa-adtrust-install** 会输出一个服务记录列表，您必须手动添加到 DNS 中，然后才能继续操作。

4. 这个脚本会提示您 **/etc/samba/smb.conf** 已经存在，并将重写：

```
WARNING: The smb.conf already exists. Running ipa-adtrust-install will break your existing Samba configuration.
```

```
Do you wish to continue? [no]: yes
```

5. 此脚本提示您配置 **slapi-nis** 插件，该插件是允许旧的 Linux 客户端与可信用户合作的兼容性插件：

```
Do you want to enable support for trusted domains in Schema Compatibility plugin?
This will allow clients older than SSSD 1.9 and non-Linux clients to work with trusted users.
```

```
Enable trusted domains support in slapi-nis? [no]: yes
```

6. 提示时，输入 IdM 域的 NetBIOS 名称，或者按 **Enter** 接受推荐的名称：

```
Trust is configured but no NetBIOS domain name found, setting it now.
Enter the NetBIOS name for the IPA domain.
Only up to 15 uppercase ASCII letters, digits and dashes are allowed.
Example: EXAMPLE.
```

```
NetBIOS domain name [IDM]:
```

7. 此时会提示您运行 SID 生成任务,为任何现有用户创建 SID:

```
Do you want to run the ipa-sidgen task? [no]: yes
```

当该目录首次安装时,至少有一个用户 (IdM 管理员) 存在,且这是一个需要资源密集型的任务,如果您有大量用户,您可以再次运行此目录。

8. (可选) 默认情况下，对于 Windows Server 2008 及之后的版本，动态 RPC 端口范围被定义为 **49152-65535**。如果您需要为您的环境定义不同的 Dynamic RPC 端口范围,请将 Samba 配置为在防火墙设置中使用不同的端口并打开这些端口。以下示例将端口范围设置为 **55000-65000**。

```
[root@ipaserver ~]# net conf setparm global 'rpc server dynamic port range' 55000-65000
[root@ipaserver ~]# firewall-cmd --add-port=55000-65000/tcp
[root@ipaserver ~]# firewall-cmd --runtime-to-permanent
```

9. 重启 **ipa** 服务：

```
[root@ipaserver ~]# ipactl restart
```

10. 使用 **smbclient** 实用程序验证 Samba 是否响应 IdM 端的 Kerberos 验证：

```
[root@ipaserver ~]# smbclient -L server.idm.example.com -k
lp_load_ex: changing to config backend registry
Sharename      Type      Comment
-----
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

3.6.2. 使用 GPO 在 Active Directory 中启用 AES 加密类型

这部分论述了如何使用组策略对象(GPO)在 Active Directory(AD)中启用 AES 加密类型。RHEL 8 中的某些功能,比如在 IdM 客户端中运行 Samba 服务器,需要这种加密类型。

请注意, RHEL 8 不支持弱 DES 和 RC4 加密类型。

先决条件

- 以可编辑组策略的用户身份登录到 AD。
- **Group Policy Management Console** 已安装在计算机上。

流程

1. 打开 **Group Policy Management Console**。
2. 右键单击 **Default Domain Policy**, 并选择 **Edit**。**Group Policy Management Editor** 将打开。
3. 导航到 **Computer Configuration → Policies → Windows Settings → Security Settings → Local Policies → Security Options**。
4. 双击 **Network security: Configure encryption types allowed for Kerberos** 策略。
5. 选择 **AES256_HMAC_SHA1** 以及 (可选) **Future encryption types**。
6. 点**确定**。
7. 关闭 **Group Policy Management Editor**。
8. 重复 **Default Domain Controller Policy** 的步骤。
9. 等待 Windows 域控制器(DC)自动应用组策略。另外,要在 DC 中手动应用 GPO,请使用具有管理员权限的帐户输入以下命令：

```
C:\> gpupdate /force /target:computer
```

3.6.3. 在 IdM 客户端中安装和配置 Samba 服务器

这部分论述了如何在在 IdM 域注册的客户端中安装和配置 Samba。

先决条件

- IdM 服务器和客户端必须在 RHEL 8.1 或更高版本中运行。
- 如 [第 3.6.1 节 “准备 IdM 域以便在域成员中安装 Samba”](#) 所述准备了 IdM 域。
- 如果 IdM 带有 AD 配置了一个信任, 请为 Kerberos 启用 AES 加密类型。例如, 使用组策略对象 (GPO) 来启用 AES 加密类型。详情请查看 [第 3.6.2 节 “使用 GPO 在 Active Directory 中启用 AES 加密类型”](#)。

流程

1. 安装 **ipa-client-samba** 软件包：

```
[root@idm_client]# yum install ipa-client-samba
```

2. 使用 **ipa-client-samba** 实用程序准备客户端并创建初始 Samba 配置：

```
[root@idm_client]# ipa-client-samba
Searching for IPA server...
IPA server: DNS discovery
Chosen IPA master: idm_server.idm.example.com
SMB principal to be created: cifs/idm_client.idm.example.com@IDM.EXAMPLE.COM
NetBIOS name to be used: IDM_CLIENT
Discovered domains to use:

Domain name: idm.example.com
NetBIOS name: IDM
SID: S-1-5-21-525930803-952335037-206501584
ID range: 212000000 - 212199999

Domain name: ad.example.com
NetBIOS name: AD
SID: None
ID range: 1918400000 - 1918599999

Continue to configure the system with these values? [no]: yes
Samba domain member is configured. Please check configuration at /etc/samba/smb.conf
and start smb and winbind services
```

3. 默认情况下, **ipa-client-samba** 会自动将 **[homes]** 部分添加到 **/etc/samba/smb.conf** 文件中, 该文件会在用户连接时动态共享用户的主目录。如果用户在这个服务器上没有主目录, 或者您不想共享它们, 请从 **/etc/samba/smb.conf** 中删除以下行：

```
[homes]
read only = no
```

4. 共享目录和打印机。详情请查看：

- [第 3.7 节 “设置使用 POSIX ACL 的 Samba 文件共享”](#)
- [第 3.9 节 “设置使用 Windows ACL 的共享”](#)
- [第 3.15 节 “将 Samba 设置为打印服务器”](#)

5. 在本地防火墙中打开 Samba 客户端所需的端口：


```
[root@idm_client]# firewall-cmd --permanent --add-service=samba-client
[root@idm_client]# firewall-cmd --reload
```

6. 启用并启动 **smb** 和 **winbind** 服务：

```
[root@idm_client]# systemctl enable --now smb winbind
```

验证步骤

在安装了 **samba-client** 软件包的不同 IdM 域成员中执行以下验证步骤：

1. 验证并获取 Kerberos ticket-granting ticket：

```
$ kinit example_user
```

2. 使用 Kerberos 身份验证列出 Samba 服务器中的共享：

```
$ smbclient -L idm_client.idm.example.com -k
lp_load_ex: changing to config backend registry

Sharename      Type      Comment
-----
example        Disk
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

其它资源

- 有关 **ipa-client-samba** 在配置过程中执行什么步骤的详情，请查看 **ipa-client-samba(1)** man page。

3.6.4. 如果 IdM 信任新域，请手动添加 ID 映射配置

Samba 需要一个 ID 映射配置，用户可从该域访问资源。在 IdM 客户端中运行的现有 Samba 服务器中，您必须在管理员向 Active Directory(AD)域添加新信任后手动添加 ID 映射配置。

先决条件

- 您在 IdM 客户端中配置了 Samba。之后，IdM 增加了一个新的信任。
- 在可信 AD 域中必须禁用 Kerberos 的 DES 和 RC4 加密类型。为了安全起见，RHEL 8 不支持这些弱加密类型。

流程

1. 使用主机的 keytab 进行身份验证：

```
[root@idm_client]# kinit -k
```

2. 使用 **ipa idrange-find** 命令显示新域的基本 ID 和 ID 范围大小。例如，以下命令显示 **ad.example.com** 域的值：

```
[root@idm_client]# ipa idrange-find --name="AD.EXAMPLE.COM_id_range" --raw
```

```

-----
1 range matched
-----
cn: AD.EXAMPLE.COM_id_range
ipabaseid: 1918400000
ipairangesize: 200000
ipabaserid: 0
ipanttrusteddomainsid: S-1-5-21-968346183-862388825-1738313271
iparange: ipa-ad-trust
-----
Number of entries returned 1
-----

```

下一步需要 **ipabaseid** 和 **ipairangesize** 属性中的值。

- 要计算可用最高的 ID，请使用以下公式：

```
maximum_range = ipabaseid + ipairangesize - 1
```

在上一步中的值中, **ad.example.com** 域可用的最大 ID 是 **1918599999** (1918400000 + 200000 - 1)。

- 编辑 **/etc/samba/smb.conf** 文件，并将域的 ID 映射配置添加到 **[global]** 部分：

```
idmap config AD : range = 1918400000 - 1918599999
idmap config AD : backend = sss
```

将 **ipabaseid** 属性中的值指定为最低值，使用上一步中计算的值作为范围的最大值。

- 重启 **smb** 和 **winbind** 服务：

```
[root@idm_client]# systemctl restart smb winbind
```

验证步骤

- 以用户身份从新域验证并获得 Kerberos ticket-granting ticket:

```
$ kinit example_user
```

- 使用 Kerberos 身份验证列出 Samba 服务器中的共享：

```
$ smbclient -L idm_client.idm.example.com -k
lp_load_ex: changing to config backend registry

Sharename      Type      Comment
-----
example        Disk
IPC$           IPC       IPC Service (Samba 4.12.3)
...
```

3.6.5. 其它资源

- 有关将 RHEL 8 加入到 IdM 域的详情,请查看 **Installing Identity Management** 指南中的 [Installing an Identity Management client](#) 部分。

3.7. 设置使用 POSIX ACL 的 SAMBA 文件共享

作为 Linux 服务，Samba 支持与 POSIX ACL 的共享。它们允许您使用工具在 Samba 服务器中本地管理权限，比如 **chmod**。如果共享存储在支持扩展属性的文件系统中，您可以使用多个用户和组定义 ACL。



注意

如果您需要使用精细的 Windows ACL，请参考 [第 3.9 节“设置使用 Windows ACL 的共享”](#)。

这个部分的内容基于 Samba Wiki 中发布的 [Setting up a Share Using POSIX ACLs](#) 文档。许可证：[CC BY 4.0](#)。作者和贡献者：请参阅 Wiki 页中的 [历史记录](#) 标签。

3.7.1. 添加使用 POSIX ACL 的共享

本节论述了如何创建名为 **example** 的共享，它提供 **/srv/samba/example/** 目录的内容，并使用 POSIX ACL。

先决条件

Samba 采用以下模式之一设置：

- [独立服务器](#)
- [域成员](#)

流程

1. 如果不存在，请创建文件夹。例如：

```
# mkdir -p /srv/samba/example/
```

2. 如果您以 **enforcing** 模式运行 SELinux，请在该目录中设置 **samba_share_t** 上下文：

```
# semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
# restorecon -Rv /srv/samba/example/
```

3. 在目录中设置文件系统 ACL。详情请查看：

- [第 3.7.2 节“在使用 POSIX ACL 的 Samba 共享中设置标准 Linux ACL”](#)
- [第 3.7.3 节“在使用 POSIX ACL 的 Samba 共享中设置扩展的 ACL”](#)。

4. 在 **/etc/samba/smb.conf** 文件中添加示例共享。例如，添加启用了共享的写操作：

```
[example]
path = /srv/samba/example/
read only = no
```



注意

无论文件系统 ACL 是什么；如果没有设置 **read only = no**，Samba 会以只读模式共享该目录。

5. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

6. 打开所需端口并使用 `firewall-cmd` 实用程序重新载入防火墙配置：

```
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
```

7. 重启 `smb` 服务：

```
# systemctl restart smb
```

3.7.2. 在使用 POSIX ACL 的 Samba 共享中设置标准 Linux ACL

Linux 中的标准 ACL 支持为一个所有者、一个组和所有其他未定义用户设置权限。您可以使用 `chown`、`chgrp` 和 `chmod` 工具更新 ACL。如果您需要精确控制，请使用更复杂的 POSIX ACL，请参考第 3.7.3 节“在使用 POSIX ACL 的 Samba 共享中设置扩展的 ACL”。在 [Deploying different types of servers](#) 文档中使用 POSIX ACL 的 Samba 共享中设置扩展的 ACL。

以下流程将 `/srv/samba/example/` 目录的所有者设置为 `root` 用户,为 `Domain Users` 组授予读写权限,并拒绝对所有其他用户的访问。

先决条件

- 存在要设置 ACL 的 Samba 共享。

流程

```
# chown root:"Domain Users" /srv/samba/example/
# chmod 2770 /srv/samba/example/
```



注意

在目录中启用 set-group-ID(SGID)位会自动将所有新文件和子目录的默认组设置为目录组,而不是将它设置为创建新目录条目的用户的主组群的通常行为。

其它资源

- 有关权限的详情，请查看 `chown(1)` 和 `chmod(1)` man page。

3.7.3. 在使用 POSIX ACL 的 Samba 共享中设置扩展的 ACL

如果文件系统中保存了共享目录的支持扩展 ACL，您可以使用它们设置复杂的权限。扩展 ACL 可以包含多个用户和组群的权限。

扩展 POSIX ACL 可让您使用多个用户和组配置复杂的 ACL。但是，您只能设置以下权限：

- 无权限
- 读权限
- 写权限

- 完整控制

如果您需要更加细致的 Windows 权限，如 **Create folder / append data**，共享配置为使用 Windows ACL。请参阅 [第 3.9 节“设置使用 Windows ACL 的共享”](#)。

以下流程演示了如何在共享中启用扩展 ACL。另外，它还包含有关设置扩展 ACL 的示例。

先决条件

- 存在要设置 ACL 的 Samba 共享。

流程

1. 在 `/etc/samba/smb.conf` 文件中的共享部分中启用以下参数,以启用扩展 ACL 的 ACL 继承：

```
inherit acls = yes
```

详情请查看 `smb.conf(5)` 的 man page 中的参数描述。

2. 重启 **smb** 服务：

```
# systemctl restart smb
```

3. 在目录中设置 ACL。例如：

例 3.2. 设置扩展 ACL

以下流程为 **Domain Admins** 组设置读、写和执行权限,为 **Domain Users** 组读取和执行权限,并拒绝对 `/srv/samba/example/` 目录中其他任何人的访问：

1. 为主用户帐户组禁用自动授予权限：

```
# setfacl -m group::--- /srv/samba/example/
# setfacl -m default:group::--- /srv/samba/example/
```

目录的主组还映射到动态 **CREATOR GROUP** 主体。当您在 Samba 共享中使用扩展 POSIX ACL 时,会自动添加这个主体,您无法删除它。

2. 设置目录中的权限：

- a. 为 **Domain Admins** 组授予读、写和执行权限：

```
# setfacl -m group:"DOMAINDomain Admins":rwx /srv/samba/example/
```

- b. 为 **Domain Users** 组授予读取和执行权限：

```
# setfacl -m group:"DOMAINDomain Users":r-x /srv/samba/example/
```

- c. 为 **other** ACL 条目设置权限，以拒绝对不匹配其他 ACL 条目的用户的访问：

```
# setfacl -R -m other::--- /srv/samba/example/
```

这些设置只适用于这个目录。在 Windows 中，这些 ACL 映射到 **This folder only** 模式。

3. 启用上一步中设置的权限,以便由在这个目录中创建的新文件系统对象继承：

```
# setfacl -m default:group:"DOMAIN\Domain Admins":rwx /srv/samba/example/  
# setfacl -m default:group:"DOMAIN\Domain Users":r-x /srv/samba/example/  
# setfacl -m default:other::--- /srv/samba/example/
```

在这个版本中,主体的 **This folder only** 模式被设置为 **This folder, subfolders, and files**。

Samba 将流程中设置的权限映射到以下 Windows ACL:

主体	权限	适用于
<i>domain\DomainAdmins</i>	完整控制	这个文件夹、子文件夹和文件
<i>Domain\Domain Users</i>	读和执行	这个文件夹、子文件夹和文件
Everyone ^[a]	无	这个文件夹、子文件夹和文件
<i>owner (Unix User\owner)</i> ^[b]	完整控制	只限于这个文件夹
<i>primary_group (Unix User\primary_group)</i> ^[c]	无	只限于这个文件夹
CREATOR OWNER ^{[d] [e]}	完整控制	只适用于子文件夹和文件
CREATOR GROUP ^{[e][f]}	无	只适用于子文件夹和文件

[a] Samba 从 **other** ACL 条目映射这个主体的权限。

[b] Samba 将目录的所有者映射到此条目。

[c] Samba 将目录的主组群映射到这个条目。

[d] 在新文件系统对象中，创建者会自动继承这个主体的权限。

[e] 在使用 POSIX ACL 的共享中不支持从 ACL 配置或删除这些主体。

[f] 在新文件系统对象中，创建器的主组群自动继承这个主体的权限。

3.8. 对使用 POSIX ACL 的共享设置权限

另外,要限制或授予对 Samba 共享的访问,您可以在 `/etc/samba/smb.conf` 文件的共享部分中设置某些参数。



注意

如果用户、组群或主机能够访问共享,则基于共享的权限管理。这些设置不会影响文件系统 ACL。

使用基于共享的设置来限制对共享的访问，例如拒绝特定主机的访问。

先决条件

- 与 POSIX ACL 的共享已被设置。

3.8.1. 配置基于用户和组群的共享访问权限

基于用户和组群的访问控制允许您为特定用户和组群授予或拒绝对共享的访问。

先决条件

- 已存在您要设置用户或组群访问的 Samba 共享。

流程

1. 例如：要让 **Domain Users** 组的所有成员在 **user** 帐户无法访问共享时访问共享，请在共享的配置中添加以下参数：

```
valid users = +DOMAIN\Domain Users"
invalid users = DOMAINuser
```

invalid users 参数的优先级高于 **valid users** 参数。例如，如果 **user** 帐户是 **Domain Users** 组的成员，当您使用上例时，将拒绝访问此帐户。

2. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

其它资源

- 详情请查看 **smb.conf(5)** man page 中的参数描述。

3.8.2. 配置基于主机的共享访问权限

基于主机的访问控制允许您根据客户端的主机名、IP 地址或 IP 范围授予或拒绝对共享的访问。

以下流程解释了如何启用 **127.0.0.1** IP 地址、**192.0.2.0/24** IP 范围以及 **client1.example.com** 主机访问共享，另外还拒绝 **client2.example.com** 主机的访问：

先决条件

- 已存在您要设置基于主机的访问的 Samba 共享。

流程

1. 在 **/etc/samba/smb.conf** 文件中共享配置中添加以下参数：

```
hosts allow = 127.0.0.1 192.0.2.0/24 client1.example.com
hosts deny = client2.example.com
```

hosts deny 参数的优先级高于 **hosts allow**。例如，如果 **client1.example.com** 解析为 **hosts allow** 参数中列出的 IP 地址，则拒绝对这个主机的访问。

2. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

其它资源

- 详情请查看 **smb.conf(5)** man page 中的参数描述。

3.9. 设置使用 WINDOWS ACL 的共享

Samba 支持在共享和文件系统对象中设置 Windows ACL。这可让您：

- 使用精细 Windows ACL
- 使用 Windows 管理共享权限和文件系统 ACL

另外,您还可以将共享配置为使用 POSIX ACL。详情请查看 [第 3.7 节 “设置使用 POSIX ACL 的 Samba 文件共享”](#)。

这个部分的内容基于 Samba Wiki 中发布的 [Setting up a Share Using Windows ACLs](#) 文档。许可证：CC BY 4.0。作者和贡献者：请参阅 Wiki 页中的 [历史记录](#) 标签。

3.9.1. 授予 SeDiskOperatorPrivilege 权限

只有授予 **SeDiskOperatorPrivilege** 权限的用户和组才能在使用 Windows ACL 的共享上配置权限。

流程

1. 例如，要为 **DOMAIN\Domain Admins** 组授予 **SeDiskOperatorPrivilege** 权限：

```
# net rpc rights grant "DOMAIN\Domain Admins" SeDiskOperatorPrivilege -U
"DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```



注意

在域环境中，向域组授予 **SeDiskOperatorPrivilege**。这可让您通过更新用户的组成员资格来集中管理权限。

2. 列出授予 **SeDiskOperatorPrivilege** 的所有用户和组：

```
# net rpc rights list privileges SeDiskOperatorPrivilege -U "DOMAIN\administrator"
Enter administrator's password:
SeDiskOperatorPrivilege:
BUILTIN\Administrators
DOMAIN\Domain Admins
```

3.9.2. 启用 Windows ACL 支持

要配置支持 Windows ACL 的共享，您必须在 Samba 中启用此功能。

先决条件

- 在 Samba 服务器中配置了一个用户共享。

流程

1. 要全局为所有共享启用,请在 `/etc/samba/smb.conf` 文件的 **[global]** 部分添加以下设置 :

```
vfs objects = acl_xattr
map acl inherit = yes
store dos attributes = yes
```

或者,您可以通过在共享部分添加相同的参数来为各个共享启用 Windows ACL 支持。

2. 重启 **smb** 服务 :

```
# systemctl restart smb
```

3.9.3. 添加使用 Windows ACL 的共享

本节论述了如何创建名为 **example** 的共享,它共享了 `/srv/samba/example/` 目录的内容,并使用 Windows ACL。

流程

1. 如果不存在,请创建文件夹。例如 :

```
# mkdir -p /srv/samba/example/
```

2. 如果您以 **enforcing** 模式运行 SELinux,请在该目录中设置 **samba_share_t** 上下文 :

```
# semanage fcontext -a -t samba_share_t "/srv/samba/example(/.*)?"
# restorecon -Rv /srv/samba/example/
```

3. 在 `/etc/samba/smb.conf` 文件中添加示例共享。例如,添加启用了共享的写操作 :

```
[example]
path = /srv/samba/example/
read only = no
```



注意

无论文件系统 ACL 是什么 ; 如果没有设置 **read only = no**, Samba 会以只读模式共享该目录。

4. 如果您还没有在 **[global]** 部分为所有共享启用 Windows ACL 支持,请在 **[example]** 部分添加以下参数来为这个共享启用此功能 :

```
vfs objects = acl_xattr
map acl inherit = yes
store dos attributes = yes
```

5. 验证 `/etc/samba/smb.conf` 文件 :

```
# testparm
```

6. 打开所需端口并使用 **firewall-cmd** 实用程序重新载入防火墙配置：

```
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
```

7. 重启 **smb** 服务：

```
# systemctl restart smb
```

3.9.4. 管理使用 Windows ACL 的共享的共享权限和文件系统 ACL

要管理使用 Windows ACL 的 Samba 共享中的共享权限和文件系统 ACL，请使用 Windows 应用程序，如 **Computer Management**。详情请查看 Windows 文档。或者，使用 **smbcacs** 工具管理 ACL。



注意

要从 Windows 修改文件系统权限,您必须使用授予 **SeDiskOperatorPrivilege** 权限的帐户。

其它资源

- [第 3.10 节 “使用 smbcacs 在 SMB 共享中管理 ACL”](#)
- [第 3.9.1 节 “授予 SeDiskOperatorPrivilege 权限”](#)

3.10. 使用 SMBACLS 在 SMB 共享中管理 ACL

smbcacs 工具可以列出、设置和删除存储在 SMB 共享中的文件和目录的 ACL。您可以使用 **smbcacs** 管理文件系统 ACL：

- 在使用高级 Windows ACL 或 POSIX ACL 的本地或远程 Samba 服务器中
- 在 Red Hat Enterprise Linux 上，远程管理在 Windows 上托管的共享的 ACL

3.10.1. 访问控制条目

文件系统对象的每个 ACL 条目包含以下格式的访问控制条目(ACE):

```
security_principal:access_right/inheritance_information/permissions
```

例 3.3. 访问控制条目

如果 **AD\Domain Users** 组有适用于 Windows **This folder, subfolders, and files** 的 **Modify** 权限,ACL 包含以下 ACE:

```
AD\Domain Users:ALLOWED/OI|CI/CHANGE
```

ACE 包含以下部分：

安全主体

安全主体是 ACL 中权限的用户、组群或 SID。

访问权利

定义是否授予或拒绝对对象的访问。该值可以是 **ALLOWED** 或 **DENIED**。

继承信息

存在以下值：

表 3.1. 继承设置

值	描述	映射到
OI	对象实例	这个文件夹和文件
CI	容器继承	这个文件夹和子文件夹
IO	只继承	ACE 不适用于当前文件或目录
ID	继承	ACE 从父目录中继承

另外，这些值可以合并如下：

表 3.2. 继承设置组合

值组合	映射至 Windows Applies to 设置
OI CI	这个文件夹、子文件夹和文件
OI CI IO	只适用于子文件夹和文件
CI IO	只使用子文件夹
OI IO	仅限文件

权限

这个值可以是代表一个或多个 Windows 权限的十六进制值，也可以是一个 **smbcacls** 别名：

- 代表一个或多个 Windows 权限的十六进制值。
下表以十六进制格式显示高级 Windows 权限及其对应值：

表 3.3. Windows 权限及其对应的 smbcacls 值（以十六进制表示）

Windows 权限	十六进制值
完整控制	0x001F01FF
遍历文件夹 / 执行文件	0x00100020

Windows 权限	十六进制值
列出文件夹 / 读数据	0x00100001
读取属性	0x00100080
读取扩展属性	0x00100008
创建文件 / 写数据	0x00100002
创建文件夹/附加数据	0x00100004
写入属性	0x00100100
写扩展属性	0x00100010
删除子文件夹和文件	0x00100040
删除	0x00110000
读取权限	0x00120000
更改权限	0x00140000
获取所有权	0x00180000

使用位 **OR** 操作可将多个权限合并为一个十六进制值。详情请查看 [第 3.10.3 节 “ACE 掩码计算”](#)。

- **smbcacs** 别名。下表显示了可用的别名：

表 3.4. 现有 smbcacs 别名及其对应的 Windows 权限

smbcacs 别名	映射至 Windows 权限
R	读
READ	读和执行

smbcacs 别名	映射至 Windows 权限
W	特殊： <ul style="list-style-type: none">创建文件 / 写数据创建文件夹/附加数据写入属性写扩展属性读取权限
D	删除
P	更改权限
O	获取所有权
X	遍历 / 执行
CHANGE	修改
FULL	完整控制



注意

设置权限时，您可以组合单例别名。例如，您可以设置 **RD** 以应用 Windows 权限 **Read** 和 **Delete**。但是,您无法组合多个非单例别名,也无法组合别名和十六进制值。

3.10.2. 使用 smbcacs 显示 ACL

要在 SMB 共享中显示 ACL，请使用 **smbcacs** 工具。如果您在没有操作参数的情况下运行 **smbcacs**，如 **--add**，则工具会显示文件系统对象的 ACL。

流程

例如，要列出 **//server/example** 共享的根目录的 ACL：

```
# smbcacs //server/example -U "DOMAINadministrator"
Enter DOMAINadministrator's password:
REVISION:1
CONTROL:SR|PD|DI|DP
OWNER:AD\Administrators
GROUP:AD\Domain Users
ACL:AD\Administrator:ALLOWED/OI|CI/FULL
ACL:AD\Domain Users:ALLOWED/OI|CI/CHANGE
ACL:AD\Domain Guests:ALLOWED/OI|CI/0x00100021
```

命令的输出会显示：

- **REVISION**:安全描述符的内部 Windows NT ACL 修订
- **CONTROL**: 安全描述符控制
- **OWNER**:安全描述符所有者的名称或 SID
- **GROUP**:安全描述符组的名称或 SID
- **ACL** 条目。详情请查看 [第 3.10.1 节“访问控制条目”](#)。

3.10.3. ACE 掩码计算

在大多数情况下，当添加或更新 ACE 时，您可以使用 [表 3.4 “现有 smbcacls 别名及其对应的 Windows 权限”](#) 中列出的 **smbcacls** 别名。

但是,如果要设置在 [表 3.3 “Windows 权限及其对应的 smbcacls 值（以十六进制表示）”](#) 中列出的高级 Windows 权限,则必须使用位版本的 **OR** 操作来计算正确的值。您可以使用以下 shell 命令计算值：

```
# echo $(printf '0x%X' $(( hex_value_1 | hex_value_2 | ... )))
```

例 3.4. 计算 ACE 掩码

您需要设置以下权限：

- 遍历文件夹 / 执行文件(0x00100020)
- 列出文件夹 / 读数据(0x00100001)
- 读取属性(0x00100080)

要计算之前权限的十六进制值,请输入：

```
# echo $(printf '0x%X' $(( 0x00100020 | 0x00100001 | 0x00100080 )))
0x1000A1
```

设置或更新 ACE 时使用返回的值。

3.10.4. 使用 smbcacls 添加、更新和删除 ACL

根据您传递给 **smbcacls** 工具的参数,您可以从文件或目录中添加、更新和删除 ACL。

添加 ACL

为 `//server/example` 共享的根添加一个 ACL，为 **AD\Domain Users** 授予对 **This folder, subfolders, and files** 的 **CHANGE** 权限：

```
# smbcacls //server/example / -U "DOMAIN\administrator --add ACL:"AD\Domain
Users":ALLOWED/OI|CI/CHANGE
```

更新 ACL

更新 ACL 与添加新的 ACL 类似。您可以使用带有现有安全主体的 **--modify** 参数覆盖 ACL 来更新 ACL。如果 **smbcacls** 在 ACL 列表中找到安全主体，实用程序会更新权限。否则,命令会失败并显示错误：

-

```
ACL for SID principal_name not found
```

例如,要更新 **AD\Domain Users** 组的权限,并为 **This folder, subfolders, and files** 将它们设置为 **READ**:

```
# smbcacls //server/example / -U "DOMAIN\administrator --modify ACL:"AD\Domain
Users":ALLOWED/OI|CI/READ
```

删除 ACL

要删除 ACL, 将带有完全 ACL 的 **--delete** 参数传递给 **smbcacls** 实用程序。例如 :

```
# smbcacls //server/example / -U "DOMAIN\administrator --delete ACL:"AD\Domain
Users":ALLOWED/OI|CI/READ
```

3.11. 允许用户在 SAMBA 服务器上共享目录

在 Samba 服务器中,您可以配置该用户可以在没有 root 权限的情况下共享目录。

3.11.1. 启用用户共享功能

在用户可以共享目录前,管理员必须在 Samba 中启用用户共享。

例如,仅启用本地 **example** 组成员来创建用户共享。

流程

1. 如果不存在, 请创建本地 **example** 组 :

```
# groupadd example
```

2. 为 Samba 准备目录以存储用户共享定义并正确设置其权限。例如 :

- a. 创建目录 :

```
# mkdir -p /var/lib/samba/usershares/
```

- b. 为 **example** 组设置写入权限 :

```
# chgrp example /var/lib/samba/usershares/
# chmod 1770 /var/lib/samba/usershares/
```

- c. 设置粘性位以防止用户重命名或删除此目录中其他用户存储的文件。

3. 编辑 **/etc/samba/smb.conf** 文件并在 **[global]** 部分添加以下内容 :

- a. 设置您配置用来存储用户共享定义的目录的路径。例如 :

```
usershare path = /var/lib/samba/usershares/
```

- b. 设置允许在这个服务器上创建多少个用户共享 Samba。例如 :

```
usershare max shares = 100
```

如果您将默认 **0** 用于 **usershare max shares** 参数，则禁用用户共享。

- c. 另外，还可设置绝对目录路径列表。例如：要配置 Samba 只允许共享 **/data** 和 **/srv** 目录的子目录,请设置：

```
usershare prefix allow list = /data /srv
```

有关您可以设置的更多与用户共享相关的参数列表,请查看 **smb.conf(5)** man page 中的 **USERSHARES** 部分。

4. 验证 **/etc/samba/smb.conf** 文件：

```
# testparm
```

5. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

用户现在可以创建用户共享。

3.11.2. 添加用户共享

在 Samba 中启用了用户共享功能后,用户可以通过运行 **net usershare add** 命令在 Samba 服务器上共享目录,无需 **root** 权限。

net usershare add 命令的同步：

```
net usershare add share_name path [[ comment ] ] [ ACLs ] [ guest_ok=y|n ]
```



重要

如果在创建用户共享时设置了 ACL,您必须在 ACL 之前指定注释参数。要设置空注释,在双引号中使用空字符串。

请注意，如果管理员在 **/etc/samba/smb.conf** 文件的 **[global]** 部分设置了 **usershare allow guests = yes**，用户只能启用用户共享的虚拟机访问。

例 3.5. 添加用户共享

用户想在 Samba 服务器中共享 **/srv/samba/** 目录。共享应命名为 **example**,没有设置注释,客户端用户应可访问该共享。另外,共享权限应该被设置为完全访问 **AD\Domain Users** 组,并为其其他用户读取权限。要添加此共享，请以用户身份运行：

```
$ net usershare add example /srv/samba/ "" "AD\Domain Users":F,Everyone:R
guest_ok=yes
```

3.11.3. 更新用户共享的设置

要更新用户共享的设置，请使用具有相同共享名称和新设置的 **net usershare add** 命令覆盖共享。请参阅第 3.11.2 节“添加用户共享”。

3.11.4. 显示现有用户共享的信息

用户可以在 Samba 服务器中输入 **net usershare info** 命令来显示用户共享及其设置。

先决条件

- 在 Samba 服务器中配置了一个用户共享。

流程

1. 显示任意用户创建的所有用户共享：

```
$ net usershare info -l
[share_1]
path=/srv/samba/
comment=
usershare_acl=Everyone:R,host_name:user:F,
guest_ok=y
...
```

要只列出运行该命令的用户创建的共享,请省略 **-l** 参数。

2. 要只显示特定共享的信息,将共享名称或 wild 卡传递给命令。例如,显示名称开头为 **share_** 的共享信息：

```
$ net usershare info -l share_*
```

3.11.5. 列出用户共享

如果您只想列出可用的用户共享而不在 Samba 服务器中设置,请使用 **net usershare list** 命令。

先决条件

- 在 Samba 服务器中配置了一个用户共享。

流程

1. 列出任意用户创建的共享：

```
$ net usershare list -l
share_1
share_2
...
```

要只列出运行该命令的用户创建的共享,请省略 **-l** 参数。

2. 要只列出特定的共享,将共享名称或 wild 卡传递给命令。例如,仅列出以 **share_** 开头的共享：

```
$ net usershare list -l share_*
```

3.11.6. 删除用户共享

要删除用户共享,以创建共享的用户或以 **root** 用户身份使用命令 **net usershare delete** 命令。

先决条件

- 在 Samba 服务器中配置了一个用户共享。

流程

```
$ net usershare delete share_name
```

3.12. 配置共享以允许不进行身份验证的访问

在某些情况下,您要共享用户可在无需验证的情况下连接的目录。要配置它,请在共享中启用客户端访问。



警告

不需要身份验证的共享可能会造成安全隐患。

3.12.1. 启用对共享的客户端访问

如果在共享中启用了客户端访问, Samba 会将客户机连接映射到 **guest account** 参数中设置的操作系统帐户。如果满足至少以下条件之一,客户端用户可以访问此共享上的文件：

- 该帐户在文件系统 ACL 中列出
- **other** 用户的 POSIX 权限允许该权限

例 3.6. 客户端共享权限

如果您将 Samba 配置为将客户机帐户映射到 **nobody**（默认设置），则下例中的 ACL：

- 允许客户端用户读取 **file1.txt**
- 允许客户端用户读取和修改 **file2.txt**
- 防止客户端用户读取或修改 **file3.txt**

```
-rw-r--r--. 1 root    root    1024 1. Sep 10:00 file1.txt
-rw-r-----. 1 nobody root    1024 1. Sep 10:00 file2.txt
-rw-r-----. 1 root    root    1024 1. Sep 10:00 file3.txt
```

流程

1. 编辑 **/etc/samba/smb.conf** 文件：
 - a. 如果这是您在这个服务器上设置的第一个客户机共享：
 - i. 在 **[global]** 部分设置 **map to guest = Bad User**:

```
[global]
...
map to guest = Bad User
```

使用这个设置时,Samba 会拒绝使用不正确的密码的登录尝试,除非用户名不存在。如果指定用户名不存在,且共享上启用了客户端访问, Samba 将连接视为客户机登录。

- ii. 默认情况下, Samba 将客户机帐户映射到 Red Hat Enterprise Linux 中的 **nobody** 帐户。另外, 您也可以设置另外一个帐户。例如 :

```
[global]
...
guest account = user_name
```

此参数中设置的帐户必须在 Samba 服务器中本地存在。出于安全考虑, 红帽建议使用没有分配有效 shell 的帐户。

- b. 在 **[example]** 共享部分添加 **guest ok = yes** 设置 :

```
[example]
...
guest ok = yes
```

2. 验证 **/etc/samba/smb.conf** 文件 :

```
# testparm
```

3. 重新载入 Samba 配置 :

```
# smbcontrol all reload-config
```

3.13. 为 MACOS 客户端配置 SAMBA

fruit 虚拟文件系统(VFS)Samba 模块可提高与 Apple 服务器消息块(SMB)客户端的兼容性。

3.13.1. 优化 Samba 配置, 以便为 macOS 客户端提供文件共享

这部分论述了如何为在服务器中托管的所有 Samba 共享配置 **fruit** 模块,以便为 macOS 客户端优化 Samba 文件共享。



注意

红帽建议全局启用 **fruit** 模块。当客户端建立与服务器的第一个连接时,使用 macOS 协商服务器消息块版本 2(SMB2)Apple(AAPL)协议扩展的客户端。如果客户端首先连接到没有启用 AAPL 扩展的共享,客户端不会将扩展用于服务器的任何共享。

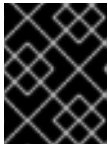
先决条件

- Samba 配置为文件服务器。

流程

1. 编辑 `/etc/samba/smb.conf` 文件,并在 `[global]` 部分启用 **fruit** 和 **streams_xattr** VFS 模块 :

```
vfs objects = fruit streams_xattr
```



重要

您必须在启用 **streams_xattr** 前启用 **fruit** 模块。**fruit** 模块使用备用数据流 (ADS)。因此,还必须启用 **streams_xattr** 模块。

2. 另外,要在一个共享中提供 macOS Time Machine 支持,请将以下设置添加到 `/etc/samba/smb.conf` 文件中的共享配置中 :

```
fruit:time machine = yes
```

3. 验证 `/etc/samba/smb.conf` 文件 :

```
# testparm
```

4. 重新载入 Samba 配置 :

```
# smbcontrol all reload-config
```

其它资源

- 有关 **fruit** VFS 模块的详情, 请查看 **vfs_fruit(8)** man page。
- 有关配置文件共享的详情, 请参考 :
 - [第 3.7 节 “设置使用 POSIX ACL 的 Samba 文件共享”](#)
 - [第 3.9 节 “设置使用 Windows ACL 的共享”](#)。

3.14. 使用 SMBCLIENT 实用程序访问 SMB 共享

smbclient 工具可让您访问 SMB 服务器中的文件共享, 类似于命令行 FTP 客户端。例如,您可以使用它将文件上传和下载到共享或从共享下载。

先决条件

- 已安装 **samba-client** 软件包。

3.14.1. smbclient 互动模式如何工作

例如, 使用 **DOMAIN\user** 帐户验证 **example** 托管的 **server** 共享 :

```
# smbclient -U "DOMAIN\user" //server/example
Enter domain\user's password:
Try "help" to get a list of possible commands.
smb: \>
```

smbclient 成功连接到共享后,实用程序进入互动模式并显示以下提示 :

```
smb: \>
```

要在互动 shell 中显示所有可用命令，请输入：

```
smb: \> help
```

要显示特定命令的帮助信息，请输入：

```
smb: \> help command_name
```

其它资源

- 有关互动 shell 中可用命令的详情和描述请查看 **smbclient(1)** man page。

3.14.2. 在互动模式中使用 smbclient

如果您在没有 **-c** 参数的情况下使用 **smbclient**，实用程序进入互动模式。下面的步骤演示了如何连接到 SMB 共享并从子目录中下载文件。

流程

1. 连接到共享：

```
# smbclient -U "DOMAIN\user_name" //server_name/share_name
```

2. 更改为 **/example/** 目录：

```
smb: \> d /example/
```

3. 列出目录中的文件：

```
smb: \example\> ls
.           D      0 Thu Nov 1 10:00:00 2018
..          D      0 Thu Nov 1 10:00:00 2018
example.txt N 1048576 Thu Nov 1 10:00:00 2018

9950208 blocks of size 1024. 8247144 blocks available
```

4. 下载 **example.txt** 文件：

```
smb: \example\> get example.txt
getting file \directory\subdirectory\example.txt of size 1048576 as example.txt (511975,0
KiloBytes/sec) (average 170666,7 KiloBytes/sec)
```

5. 从共享断开：

```
smb: \example\> exit
```

3.14.3. 在脚本模式中使用 smbclient

如果您将 **-c** 参数传递给 **smbclient**,您可以在远程 SMB 共享中自动执行命令。这可让您在脚本中使用 **smbclient**。

下面的步骤演示了如何连接到 SMB 共享并从子目录中下载文件。

流程

- 使用以下命令连接到共享,切换到 **example** 目录,下载 **example.txt** 文件 :

```
# smbclient -U DOMAIN\user_name //server_name/share_name -c "cd /example/ ; get example.txt ; exit"
```

3.15. 将 SAMBA 设置为打印服务器

如果您将 Samba 设置为打印服务器,则网络中的客户端可以使用 Samba 打印。另外,如果配置 Windows 客户端,可以从 Samba 服务器下载驱动程序。

在 [设置 Samba 作为在 Samba Wiki 中发布的 Print Server 文档中](#), 这个部分的内容被记录。许可证 : [CC BY 4.0](#)。作者和贡献者 : 请参阅 Wiki 页中的 [历史记录](#) 标签。

先决条件

Samba 采用以下模式之一设置 :

- [独立服务器](#)
- [域成员](#)

3.15.1. Samba spoolssd 服务

Samba **spoolssd** 是一个集成到 **smbd** 服务中的服务。在 Samba 配置中启用 **spoolssd** 以显著提高带有大量作业或打印机的打印服务器的性能。

如果没有 **spoolssd**,Samba fork 处理 **smbd** 进程并为每个打印作业初始化 **printcap** 缓存。如果有大量打印机,在初始化缓存时,**smbd** 服务可能会成为无响应的多次。**spoolssd** 服务可让您启动预查找的 **smbd** 进程,这些进程在没有延迟的情况下处理打印作业。主要的 **spoolssd smbd** 进程使用较少的内存, fork 和终止子进程。

以下流程解释了如何启用 **spoolssd** 服务。

流程

1. 编辑 **[global]** 文件中的 **/etc/samba/smb.conf** 部分 :

- a. 添加以下参数 :

```
rpc_server:spoolss = external
rpc_daemon:spoolssd = fork
```

- b. 另外, 您可以设置以下参数 :

参数	Default (默认)	描述
spoolssd:prefork_min_children	5	最小子进程数量
spoolssd:prefork_max_children	25	子进程的最大数量
spoolssd:prefork_spawn_rate	5	Samba Fork 此参数中设置的新子进程数量,如果建立了新的连接,可达到 spoolssd:prefork_max_children 中设置的值
spoolssd:prefork_max_allowed_clients	100	客户端数,子进程服务
spoolssd:prefork_child_min_life	60	子进程的最低生命周期（以秒为单位）。60 秒是最小的。

2. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

3. 重启 **smb** 服务：

```
# systemctl restart smb
```

重启该服务后,Samba 会自动启动 **smbd** 子进程：

```
# ps axf
...
30903 smbd
30912 \_ smbd
30913 \_ smbd
30914 \_ smbd
30915 \_ smbd
...
```

3.15.2. 在 Samba 中启用打印服务器支持

这部分论述了如何在 Samba 中启用打印服务器支持。

流程

1. 在 Samba 服务器中,设置 CUPS 并在 CUPS 后端中添加打印机。有关在 CUPS 中配置打印机的详情;请查看打印服务器上的 CUPS Web 控制台中(https://print_server_host_name:631/help)中提供的文档。



注意

只有 Samba 打印服务器中本地安装了 CUPS,Samba 才可以将打印任务转发到 CUPS。

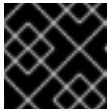
2. 编辑 `/etc/samba/smb.conf` 文件：

- a. 如果要启用 **spoolssd** 服务,请在 **[global]** 部分添加以下参数：

```
rpc_server:spoolss = external
rpc_daemon:spoolssd = fork
```

- b. 要配置打印后端，请添加 **[printers]** 部分：

```
[printers]
comment = All Printers
path = /var/tmp/
printable = yes
create mask = 0600
```



重要

[printers] 共享名称是硬编码的,无法更改。

3. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

4. 打开所需端口并使用 **firewall-cmd** 实用程序重新载入防火墙配置：

```
# firewall-cmd --permanent --add-service=samba
# firewall-cmd --reload
```

5. 重启 **smb** 服务：

```
# systemctl restart smb
```

重启该服务后,Samba 会自动共享 CUPS 后端中配置的所有打印机。如果您想手动共享特定的打印机,请参阅 [第 3.15.3 节“手动共享特定打印机”](#)。

3.15.3. 手动共享特定打印机

如果您将 Samba 配置为打印服务器,默认情况下 Samba 共享在 CUPS 后端中配置的所有打印机。以下流程解释了如何只共享特定打印机。

先决条件

- Samba 被设置为打印服务器

流程

1. 编辑 `/etc/samba/smb.conf` 文件：

- a. 在 **[global]** 部分,通过设置来禁用自动打印机共享 :

```
load printers = no
```

- b. 为每个要共享的打印机添加一个部分。例如：要在 CUPS 后端中将名为 **example** 的打印机共享为 Samba 中的 **Example-Printer**,请添加以下部分：

```
[Example-Printer]
    path = /var/tmp/
    printable = yes
    printer name = example
```

您不需要为每个打印机单独设置 spool 目录。您可以在打印机的 **path** 参数中设置与 **[printers]** 部分中设置的相同的 spool 目录。

2. 验证 **/etc/samba/smb.conf** 文件：

```
# testparm
```

3. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

3.16. 在 SAMBA 打印服务器中为 WINDOWS 客户端设置自动打印机驱动程序下载

如果您正在为 Windows 客户端运行 Samba 打印服务器,可以上传驱动程序和预配置打印机。如果用户连接到打印机,Windows 会自动在客户端本地下载并安装驱动程序。用户不需要本地管理员权限进行安装。另外,Windows 应用预先配置的驱动程序设置,如 trays 数。

在为 [Windows 客户端设置自动 Printer Driver Downloads](#) 部分在 Samba Wiki 中发布。许可证：[CC BY 4.0](#)。作者和贡献者：请参阅 Wiki 页中的 [历史记录](#) 标签。

先决条件

- Samba 被设置为打印服务器

3.16.1. 有关打印机驱动程序的基本信息

本节提供有关打印机驱动程序的一般信息。

支持的驱动程序模型版本

Samba 只支持打印机驱动程序模型版本 3,它在 Windows 2000 及之后的版本支持,以及 Windows Server 2000 及更高版本。Samba 不支持 Windows 8 和 Windows Server 2012 中引入的驱动程序模型版本 4。但是,这些及之后的 Windows 版本也支持版本 3 驱动程序。

软件包感知驱动程序

Samba 不支持软件包感知驱动程序。

准备上传的打印机驱动程序

在您将驱动程序上传到 Samba 打印服务器之前：

- 如果驱动程序采用压缩格式提供，请解包它。

- 有些驱动程序需要启动一个设置应用程序,在 Windows 主机上在本地安装驱动程序。在某些情况下,安装程序会在设置运行期间将单个文件提取到操作系统的临时文件夹中。使用驱动程序文件上传 :
 - a. 启动安装程序。
 - b. 将临时文件夹中的文件复制到新位置。
 - c. 取消安装。

请您的打印机厂商提供支持上传到打印服务器的驱动程序。

为客户端提供 32 位和 64 位驱动

要为 32 位和 64 位 Windows 客户端提供打印机的驱动程序,您必须上传两个架构具有完全相同名称的驱动程序。例如:如果您上传了名为 **Example PostScript** 的 32 位驱动程序和名为 **Example PostScript (v1.0)** 的 64 位驱动程序,则名称不匹配。因此,您只能为打印机分配其中一个驱动程序,且该驱动程序无法对这两个架构都适用。

3.16.2. 启用用户上传和预配置驱动程序

为了可以上传和预先配置打印机驱动程序,用户或组需要授予 **SePrintOperatorPrivilege** 权限。用户必须添加到 **printadmin** 组中。在安装 **samba** 软件包时,Red Hat Enterprise Linux 会自动创建这个组。**printadmin** 组分配了低于 1000 的最低可用动态系统 GID。

流程

1. 例如,要为 **printadmin** 组授予 **SePrintOperatorPrivilege** 权限 :

```
# net rpc rights grant "printadmin" SePrintOperatorPrivilege -U
"DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```



注意

在域环境中,向域组授予 **SePrintOperatorPrivilege**。这可让您通过更新用户的组成员资格来集中管理权限。

2. 列出授予 **SePrintOperatorPrivilege** 的所有用户和组 :

```
# net rpc rights list privileges SePrintOperatorPrivilege -U "DOMAIN\administrator"
Enter administrator's password:
SePrintOperatorPrivilege:
BUILTIN\Administrators
DOMAIN\printadmin
```

3.16.3. 设置 print\$ 共享

Windows 操作系统从打印服务器中名为 **print\$** 的共享下载打印机驱动程序。这个共享名称在 Windows 中硬编码,无法更改。

以下流程解释了如何将 **/var/lib/samba/drivers/** 目录共享为 **print\$**,并启用本地 **printadmin** 组成员上传打印机驱动程序。

流程

1. 在 `/etc/samba/smb.conf` 文件中添加 `[print$]` 部分：

```
[print$]
    path = /var/lib/samba/drivers/
    read only = no
    write list = @printadmin
    force group = @printadmin
    create mask = 0664
    directory mask = 2775
```

使用这些设置：

- 只有 **printadmin** 组成员可以将打印机驱动程序上传到共享中。
 - 新创建的文件和目录的组将设置为 **printadmin**。
 - 新文件的权限将设置为 **664**。
 - 新目录的权限将设置为 **2775**。
2. 要只上传所有打印机的 64 位驱动程序,在 `/etc/samba/smb.conf` 文件的 `[global]` 部分包括这个设置：

```
spoolss: architecture = Windows x64
```

如果没有这个设置, Windows 只显示您至少上传 32 位版本的驱动程序。

3. 验证 `/etc/samba/smb.conf` 文件：

```
# testparm
```

4. 重新载入 Samba 配置

```
# smbcontrol all reload-config
```

5. 如果不存在, 创建 **printadmin** 组：

```
# groupadd printadmin
```

6. 为 **printadmin** 组授予 **SePrintOperatorPrivilege** 权限。

```
# net rpc rights grant "printadmin" SePrintOperatorPrivilege -U
"DOMAIN\administrator"
Enter DOMAIN\administrator's password:
Successfully granted rights.
```

7. 如果您以 **enforcing** 模式运行 SELinux, 请在该目录中设置 **samba_share_t** 上下文：

```
# semanage fcontext -a -t samba_share_t "/var/lib/samba/drivers(/.*)?"
# restorecon -Rv /var/lib/samba/drivers/
```

8. 在 `/var/lib/samba/drivers/` 目录中设置权限：

- 如果使用 POSIX ACL,请设置：

```
# chgrp -R "printadmin" /var/lib/samba/drivers/  
# chmod -R 2775 /var/lib/samba/drivers/
```

- 如果使用 Windows ACL,请设置：

主体	权限	适用于
CREATOR OWNER	完整控制	只适用于子文件夹和文件
Authenticated Users	读和执行、列出目录内容、读	这个文件夹、子文件夹和文件
printadmin	完整控制	这个文件夹、子文件夹和文件

有关在 Windows 中设置 ACL 的详情,请查看 Windows 文档。

其它资源

- [第 3.16.2 节 “启用用户上传和预配置驱动程序”](#)。

3.16.4. 创建 GPO 以启用客户端信任 Samba 打印服务器

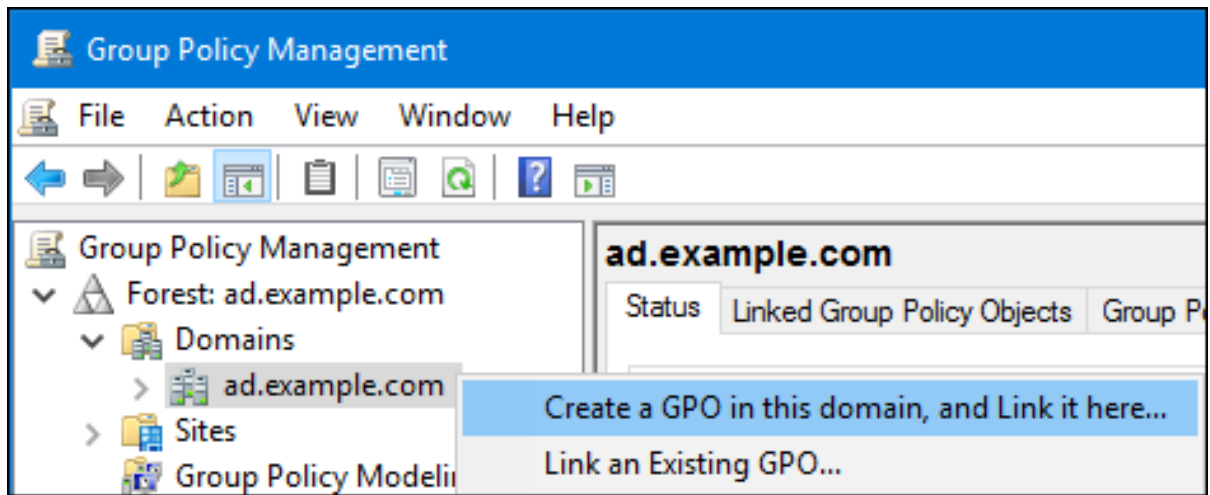
为了安全起见,最近 Windows 操作系统可防止客户端从不可信服务器下载非软件包感知打印机驱动程序。如果您的打印服务器是 AD 中的成员,您可以在域中创建组策略对象(GPO)来信任 Samba 服务器。

先决条件

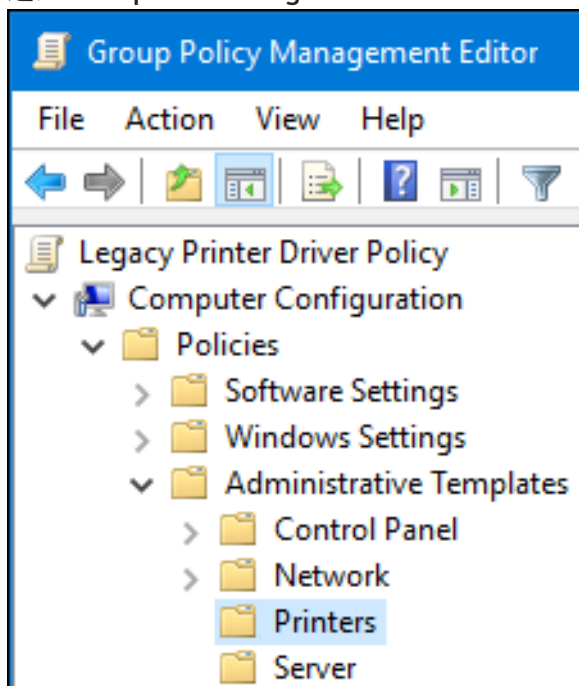
- Samba 打印服务器是 AD 域的成员。
- 您用来创建 GPO 的 Windows 计算机必须安装 Windows Remote Server Administration Tools(RSAT)。详情请查看 Windows 文档。

流程

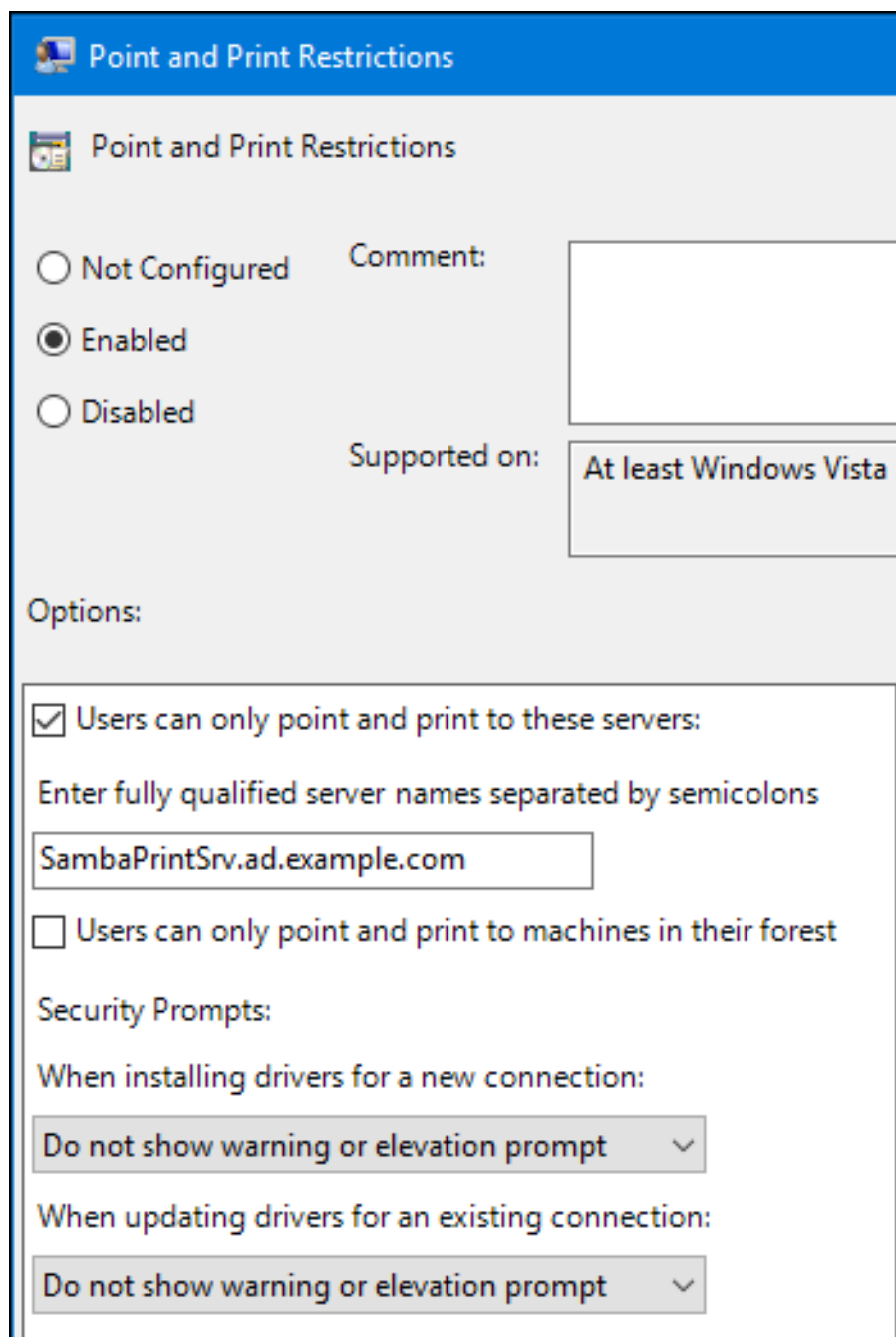
1. 使用允许编辑组策略的帐户登录 Windows 计算机，如 AD 域 **Administrator** 用户。
2. 打开 **Group Policy Management Console**。
3. 右键点击您的 AD 域并选择 **Create a GPO in this domain, and Link it here**。



4. 输入 GPO 的名称,如 **Legacy Printer Driver Policy** 并点 **OK**。新的 GPO 将在域条目下显示。
5. 右键点击新创建的 GPO 并选择 **Edit** 来打开 **Group Policy Management Editor**。
6. 进入 **Computer Configuration → Policies → Administrative Templates → Printers**。



7. 在窗口右侧,双击 **Point and Print Restriction** 来编辑策略 :
 - a. 启用策略并设置以下选项 :
 - i. 选择 **Users can only point and print to these servers** 并输入 Samba 打印服务器的完全限定域名(FQDN)到这个选项旁的字段。
 - ii. 在 **Security Prompts** 的两个复选框中,选择 **Do not show warning or elevation prompt**。



Point and Print Restrictions

Point and Print Restrictions

☐ Not Configured Comment:

☒ Enabled

☐ Disabled

Supported on:

Options:

☒ Users can only point and print to these servers:
Enter fully qualified server names separated by semicolons

☐ Users can only point and print to machines in their forest

Security Prompts:

When installing drivers for a new connection:

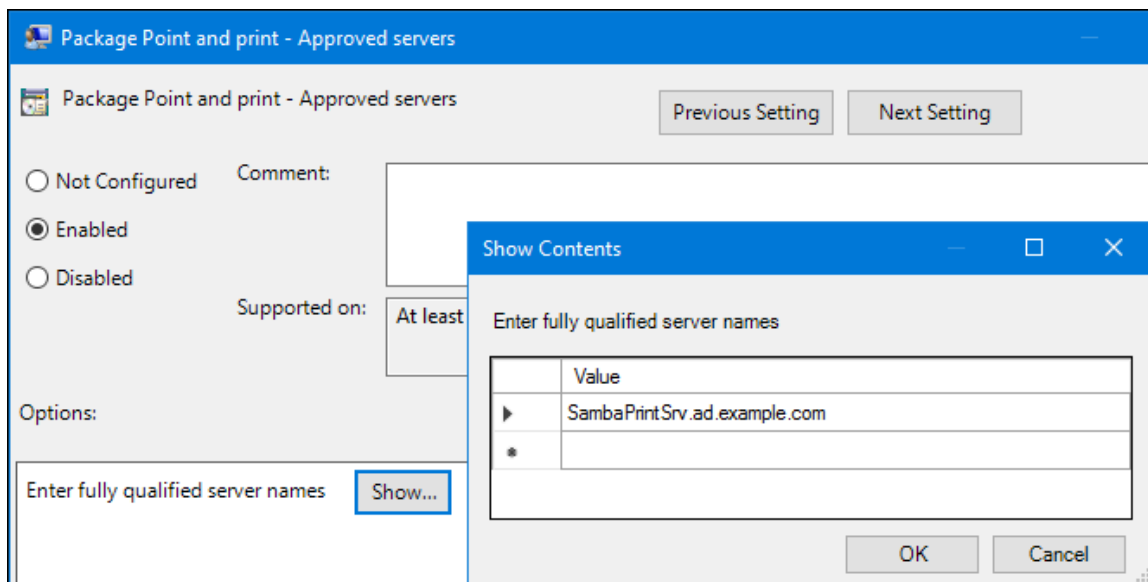
When updating drivers for an existing connection:

b. 点击确定。

8. 双击 **Package Point and Print - Approved servers** 以编辑策略：

a. 启用策略并点击 **Show** 按钮。

b. 输入 Samba 打印服务器的 FQDN。



c. 点 **OK** 关闭 **Show Contents** 和策略的属性窗口。

9. 关闭 **Group Policy Management Editor**。

10. 关闭 **Group Policy Management Console**。

在 Windows 域成员应用组策略后,当用户连接到打印机时,打印机驱动程序会自动从 Samba 服务器下载。

其它资源

- 有关使用组策略的详情, 请查看 Windows 文档。

3.16.5. 上传驱动程序和预配置打印机

使用 Windows 客户端中的 **Print Management** 应用程序上传 Samba 打印服务器中托管的驱动程序和预配置打印机。详情请查看 Windows 文档。

3.17. 调整 SAMBA 服务器的性能

本章论述了在某些情况下,哪些设置可以提高 Samba 的性能,哪些设置可能会对性能造成负面影响。

本节的部分内容来自在 Samba Wiki 中发布的 [Performance Tuning](#) 文档。许可证：[CC BY 4.0](#)。作者和贡献者：请参阅 Wiki 页中的 [历史记录](#) 标签。

先决条件

- Samba 被设置为文件或打印服务器

3.17.1. 设置 SMB 协议版本

每个新的 SMB 版本都添加了功能,并提高了协议的性能。最新的 Windows 和 Windows 服务器操作系统始终支持最新的协议版本。如果 Samba 还使用最新的协议版本,Windows 客户端连接到 Samba 可从性能改进中受益。在 Samba 中,服务器 max 协议的默认值被设置为最新的稳定 SMB 协议版本。

**注意**

要始终启用最新的稳定 SMB 协议版本,请不要设置 **server max protocol** 参数。如果手动设置参数,则需要使用每个新版本的 SMB 协议来修改设置,以启用最新的协议版本。

以下流程解释了如何使用 **server max protocol** 参数中的默认值。

流程

1. 从 **/etc/samba/smb.conf** 文件中的 **[global]** 部分删除 **server max protocol** 参数。
2. 重新载入 Samba 配置

```
# smbcontrol all reload-config
```

3.17.2. 与包含大量文件的目录调整共享

Linux 支持区分大小写的文件名。因此,在搜索或访问文件时,Samba 需要扫描目录获取大写和小写文件名。您可以配置共享来仅在小写或大写字母创建新文件,这可以提高性能。

先决条件

- Samba 配置为文件服务器

流程

1. 将共享上的所有文件重命名为小写。

**注意**

使用此过程中的设置,不再显示小写以外的名称的文件。

2. 在共享部分中设置以下参数：

```
case sensitive = true
default case = lower
preserve case = no
short preserve case = no
```

有关参数的详情,请参考 **smb.conf(5)** man page 中的描述。

3. 验证 **/etc/samba/smb.conf** 文件：

```
# testparm
```

4. 重新载入 Samba 配置：

```
# smbcontrol all reload-config
```

应用这些设置后,这个共享上所有新创建的文件名称将使用小写。由于这些设置,Samba 不再需要扫描该目录以获取大写和小写,从而提高了性能。

3.17.3. 可能会对性能造成负面影响的设置

默认情况下, Red Hat Enterprise Linux 中的内核会根据高网络性能进行了微调。例如, 内核对缓冲区大小使用自动轮询机制。在 `/etc/samba/smb.conf` 文件中设置 **socket options** 参数会覆盖这些内核设置。因此, 设置此参数会在大多数情况下降低 Samba 网络性能。

要使用内核中优化的设置, 从 `/etc/samba/smb.conf` 的 **[global]** 部分中删除 **socket options** 参数。

3.18. 将 SAMBA 配置为与需要 SMB 版本低于默认版本的客户端兼容

Samba 为它支持的最小服务器消息块(SMB)版本使用合理的安全默认值。然而,如果您有需要旧的 SMB 版本的客户端,可以配置 Samba 支持它。

3.18.1. 设置 Samba 服务器支持的最小 SMB 协议版本

在 Samba 中, `/etc/samba/smb.conf` 文件中的 **server min protocol** 参数定义 Samba 服务器支持的最小服务器消息块(SMB)协议版本。这部分论述了如何更改最小 SMB 协议版本。



注意

默认情况下,RHEL 8.2 及之后的版本只支持 SMB2 及更新的协议版本。红帽建议不要使用已弃用的 SMB1 协议。然而,如果您的环境需要 SMB1,您可以手动将 **server min protocol** 参数设置为 **NT1** 来重新启用 SMB1。

先决条件

- 已安装并配置 Samba。

流程

1. 编辑 `/etc/samba/smb.conf` 文件,添加 **server min protocol** 参数,并将该参数设置为服务器应支持的最低 SMB 协议版本。例如: 要将最小 SMB 协议版本设置为 **SMB3**,请添加:

```
server min protocol = SMB3
```

2. 重启 **smb** 服务:

```
# systemctl restart smb
```

其它资源

- 有关您可以在 **server min protocol** 参数中设置的协议版本列表, 请查看 **smb.conf(5)** man page 中的 **server max protocol** 参数描述。

3.19. 经常使用 SAMBA 命令行工具

本章论述了使用 Samba 服务器时经常使用的命令。

3.19.1. 使用 net ads join 和 net rpc join 命令

使用 **net** 工具的 **join** 子命令,您可以将 Samba 添加到 AD 或 NT4 域中。要加入域,您必须手动创建 `/etc/samba/smb.conf` 文件,并选择性地更新附加配置,比如 PAM。



重要

红帽建议使用 **realm** 工具加入域。**realm** 工具自动更新所有相关配置文件。

流程

1. 使用以下设置手动创建 **/etc/samba/smb.conf** 文件：

- 对于 AD 域成员：

```
[global]
workgroup = domain_name
security = ads
passdb backend = tdbsam
realm = AD_REALM
```

- 对于 NT4 域成员：

```
[global]
workgroup = domain_name
security = user
passdb backend = tdbsam
```

2. 为 * 默认域和您要加入 **/etc/samba/smb.conf** 文件的 **[global]** 部分添加 ID 映射配置。
3. 验证 **/etc/samba/smb.conf** 文件：

```
# testparm
```

4. 以域管理员身份加入域：

- 加入 AD 域：

```
# net ads join -U "DOMAIN\administrator"
```

- 要加入 NT4 域：

```
# net rpc join -U "DOMAIN\administrator"
```

5. 在 **/etc/nsswitch.conf** 文件中的 **passwd** 和 **group** 数据库条目中附加 **winbind** 源：

```
passwd: files winbind
group: files winbind
```

6. 启用并启动 **winbind** 服务：

```
# systemctl enable --now winbind
```

7. （可选）使用 **authselect** 工具配置 PAM。
详情请查看 **authselect(8)** man page。
8. 另外，对于 AD 环境，配置 Kerberos 客户端。
详情请查看您的 Kerberos 客户端文档。

其它资源

- [第 3.5.1 节 “将 RHEL 系统添加到 AD 域中”](#).
- [第 3.4 节 “了解并配置 Samba ID 映射”](#).

3.19.2. 使用 net rpc right 命令

在 Windows 中,您可以为帐户和组分配权限来执行特殊操作,如在共享或上传打印机驱动程序中设置 ACL。在 Samba 服务器中,您可以使用 **net rpc rights** 命令管理权限。

列出您可以设置的权限

要列出所有可用权限及其拥有者,请使用 **net rpc rights list** 命令。例如 :

```
# net rpc rights list -U "DOMAINadministrator"
Enter DOMAINadministrator's password:
SeMachineAccountPrivilege  Add machines to domain
SeTakeOwnershipPrivilege  Take ownership of files or other objects
    SeBackupPrivilege  Back up files and directories
    SeRestorePrivilege  Restore files and directories
SeRemoteShutdownPrivilege  Force shutdown from a remote system
SePrintOperatorPrivilege  Manage printers
    SeAddUsersPrivilege  Add users and groups to the domain
    SeDiskOperatorPrivilege  Manage disk shares
    SeSecurityPrivilege  System security
```

授予权限

要为帐户或组授予权限, 请使用 **net rpc rights grant** 命令。

例如, 为 **DOMAINprintadmin** 组授予 **SePrintOperatorPrivilege** 权限 :

```
# net rpc rights grant "DOMAINprintadmin" SePrintOperatorPrivilege -U
"DOMAINadministrator"
Enter DOMAINadministrator's password:
Successfully granted rights.
```

撤销权限

要从帐户或组撤销权限,请使用 **net rpc rights revoke** 命令。

例如, 从 **DOMAINprintadmin** 组中撤销 **SePrintOperatorPrivilege** 权限 :

```
# net rpc rights remoke "DOMAINprintadmin" SePrintOperatorPrivilege -U
"DOMAINadministrator"
Enter DOMAINadministrator's password:
Successfully revoked rights.
```

3.19.3. 使用 net rpc share 命令

net rpc share 命令提供列出、添加和删除本地或者远程 Samba 或者 Windows 服务器中的共享的能力。

列出共享

要列出 SMB 服务器中的共享,请使用 **net rpc share list** 命令。另外,还可将 **-S server_name** 参数传递给命令,以列出远程服务器的共享。例如 :

```
# net rpc share list -U "DOMAINadministrator" -S server_name
Enter DOMAINadministrator's password:
IPC$
share_1
share_2
...
```



注意

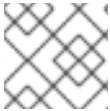
托管在 `/etc/samba/smb.conf` 文件中有 `browseable = no` 的 Samba 服务器上的共享不会在输出中显示。

添加共享

net rpc share add 命令允许您在 SMB 服务器中添加共享。

例如,要在共享 `C:\example\` 目录的远程 Windows 服务器中添加名为 **example** 的共享：

```
# net rpc share add example="C:\example" -U "DOMAINadministrator" -S server_name
```



注意

在指定 Windows 目录名称时，您必须省略路径中的结尾反斜杠。

使用命令在 Samba 服务器中添加共享：

- **-U** 参数中指定的用户必须在目标服务器上具有 **SeDiskOperatorPrivilege** 权限。
- 您必须编写在 `/etc/samba/smb.conf` 文件中添加共享部分的脚本并重新载入 Samba。该脚本必须在 `/etc/samba/smb.conf` 的 `[global]` 部分的 **add share command** 参数中设置。详情请查看 **add share command** man page 中的 **smb.conf(5)** 描述。

删除共享

net rpc share delete 命令可让您从 SMB 服务器中删除共享。

例如，要从远程 Windows 服务器中删除名为 **example** 的共享：

```
# net rpc share delete example -U "DOMAINadministrator" -S server_name
```

使用命令从 Samba 服务器中删除共享：

- **-U** 参数中指定的用户必须具有 **SeDiskOperatorPrivilege** 权限。
- 您必须编写一个脚本，从 `/etc/samba/smb.conf` 文件中删除共享部分，并重新载入 Samba。该脚本必须在 `/etc/samba/smb.conf` 的 `[global]` 部分的 **delete share command** 参数中设置。详情请查看 **delete share command** man page 中的 **smb.conf(5)** 描述。

3.19.4. 使用 net user 命令

net user 命令允许您在 AD DC 或 NT4 PDC 上执行以下操作：

- 列出所有用户帐户
- 添加用户

- 删除用户



注意

只有在列出域用户帐户时，才需要指定连接方法，如 AD 域 **ads**，或为 NT4 域指定 **rpc**。其他用户相关的子命令可以自动探测连接方法。

将 **-U user_name** 参数传递给命令，指定允许执行请求操作的用户。

列出域用户帐户

列出 AD 域中的所有用户：

```
# net ads user -U "DOMAINadministrator"
```

列出 NT4 域中的所有用户：

```
# net rpc user -U "DOMAINadministrator"
```

在域中添加用户帐户

在 Samba 域成员中,您可以使用 **net user add** 命令将用户帐户添加到域中。

例如，将 **user** 帐户添加到域中：

1. 添加帐户：

```
# net user add user password -U "DOMAINadministrator"
User user added
```

2. 另外,还可使用远程过程调用(RPC)shell 在 AD DC 或 NT4 PDC 上启用帐户。例如：

```
# net rpc shell -U DOMAINadministrator -S DC_or_PDC_name
Talking to domain DOMAIN (S-1-5-21-1424831554-512457234-5642315751)

net rpc> user edit disabled user: no
Set user's disabled flag from [yes] to [no]

net rpc> exit
```

从域中删除用户帐户

在 Samba 域成员中,您可以使用 **net user delete** 命令从域中删除用户帐户。

例如,从域中删除 **user** 帐户：

```
# net user delete user -U "DOMAINadministrator"
User user deleted
```

3.19.5. 使用 rpcclient 工具

rpcclient 工具可让您在本地或远程 SMB 服务器中手动执行 Microsoft 远程过程调用(MS-RPC)功能。但是,大多数功能都集成到 Samba 提供的单独工具中。**rpcclient** 仅用于测试 MS-PRC 功能。

先决条件

- 已安装 **samba-client** 软件包。

示例

例如,您可以使用 **rpcclient** 实用程序：

- 管理打印机 Spool Subsystem(SPOOLSS)。

例 3.7. 将驱动程序分配给打印机

```
# rpcclient server_name -U "DOMAINadministrator" -c 'setdriver "printer_name"
"driver_name"
Enter DOMAINadministrators password:
Successfully set printer_name to driver driver_name.
```

- 检索有关 SMB 服务器的信息。

例 3.8. 列出所有文件共享和共享的打印机

```
# rpcclient server_name -U "DOMAINadministrator" -c 'netshareenum'
Enter DOMAINadministrators password:
netname: Example_Share
remark:
path: C:\srv\samba\example_share\
password:
netname: Example_Printer
remark:
path: C:\var\spool\samba\
password:
```

- 使用 Security Account Manager Remote(SAMR)协议执行操作。

例 3.9. 在 SMB 服务器中列出用户

```
# rpcclient server_name -U "DOMAINadministrator" -c 'enumdomusers'
Enter DOMAINadministrators password:
user:[user1] rid:[0x3e8]
user:[user2] rid:[0x3e9]
```

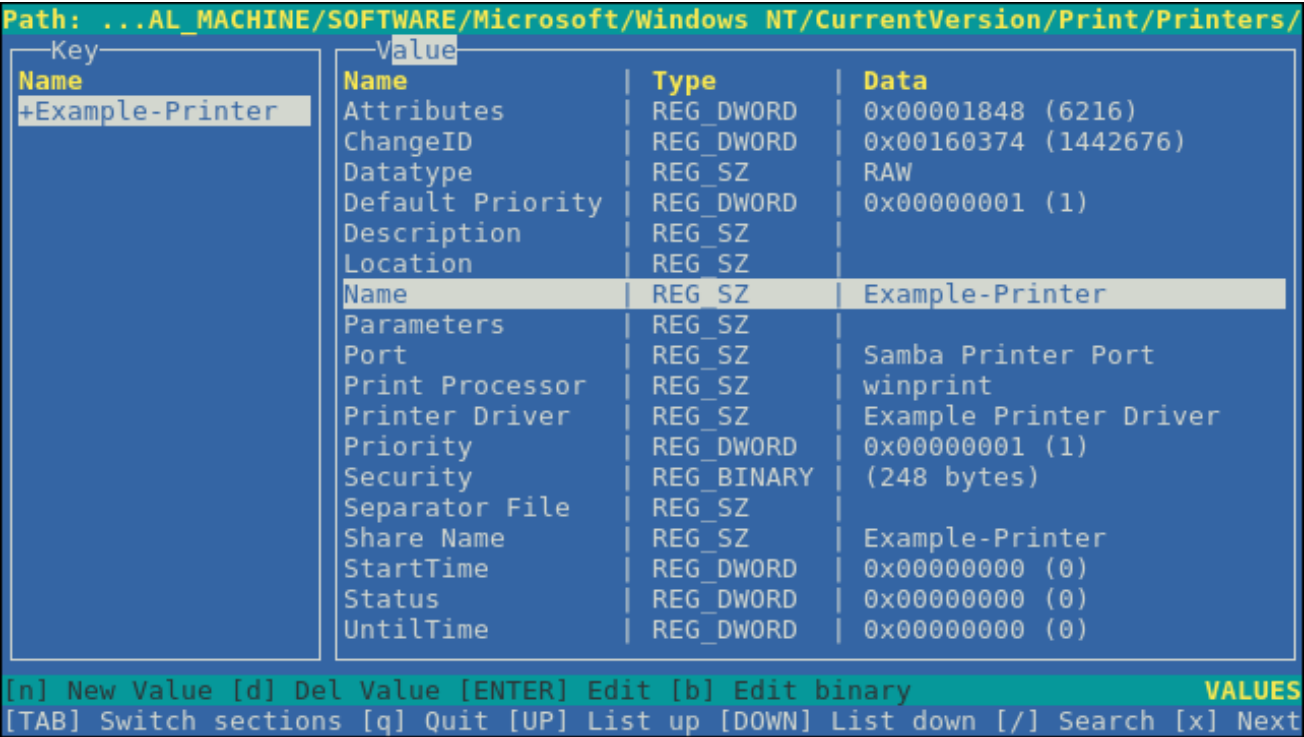
如果您针对独立服务器或域成员运行该命令,它会在本地数据库中列出用户。针对 AD DC 或 NT4 PDC 运行命令列出域用户。

其它资源

有关支持的子命令的完整列表, 请查看 **rpcclient(1)** man page 中的 **COMMANDS** 部分。

3.19.6. 使用 **samba-regedit** 应用程序

某些设置,如打印机配置,保存在 Samba 服务器的 registry 中。您可以使用基于 ncurses 的 **samba-regedit** 应用程序编辑 Samba 服务器的 registry。



先决条件

- 已安装 **samba-client** 软件包。

流程

要启动应用程序，请输入：

```
# samba-regedit
```

使用以下键：

- 上键和下键：在注册表树和值中进行导航。
- **Enter**：打开关键字或编辑值。
- **Tab**：在 **Key** 和 **Value** 窗格间切换。
- **Ctrl+C**：关闭应用程序。

3.19.7. 使用 smbcontrol 工具

smbcontrol 工具可让您将命令信息发送到 **smbd**、**nmbd**、**winbindd** 或所有这些服务。这些控制消息指示服务重新载入其配置。

本节中的步骤演示了如何通过将 **reload-config** 消息类型发送到 **all** 目的地来重新载入 **smbd**、**nmbd**、**winbindd** 服务的配置。

先决条件

- 已安装 **samba-common-tools** 软件包。

流程

■

smbcontrol all reload-config

其它资源

有关可用命令消息类型的详情和列表，请查看 **smbcontrol(1)** man page。

3.19.8. 使用 smbpasswd 工具

smbpasswd 工具管理本地 Samba 数据库中的用户帐户和密码。

先决条件

- 已安装 **samba-common-tools** 软件包。

流程

1. 如果您以用户身份运行命令，**smbpasswd** 更改运行该命令的用户的 Samba 密码。例如：

```
[user@server ~]$ smbpasswd
New SMB password: password
Retype new SMB password: password
```

2. 如果您以 **root** 用户身份运行 **smbpasswd**，您可以使用该工具，例如：

- 创建一个新用户：

```
[root@server ~]# smbpasswd -a user_name
New SMB password: password Retype new SMB password: [command]password
Added user user_name.
```



注意

在将用户添加到 Samba 数据库之前，您必须在本地操作系统中创建帐户。请参阅配置基本系统设置指南中的 [命令行部分添加新用户](#)。

- 启用 Samba 用户：

```
[root@server ~]# smbpasswd -e user_name
Enabled user user_name.
```

- 禁用 Samba 用户：

```
[root@server ~]# smbpasswd -x user_name
Disabled user ser_name
```

- 删除用户：

```
[root@server ~]# smbpasswd -x user_name
Deleted user user_name.
```

其它资源

详情请查看 **smbpasswd(8)** man page。

3.19.9. 使用 smbstatus 工具

smbstatus 工具报告：

- 每个 **smbd** 守护进程的每个 PID 到 Samba 服务器的连接。此报告包括用户名、主组群、SMB 协议版本、加密和签名信息。
- 每个 Samba 共享的连接。此报告包括 **smbd** 守护进程的 PID、连接机器的 IP 地址、连接建立的时间戳、加密和签名信息。
- 锁定文件列表。报告条目包括更多详情,如 opportunistic lock(oplock)类型

先决条件

- 已安装 **samba** 软件包。
- **smbd** 服务正在运行。

流程

```
# smbstatus
```

```
Samba version 4.12.3
```

PID	Username	Group	Machine	Protocol	Version	Encryption	Signing
-----	----------	-------	---------	----------	---------	------------	---------

```
.....
```

```
963 DOMAIN\administrator DOMAIN\domain users client-pc (ipv4:192.0.2.1:57786) SMB3_02
- AES-128-CMAC
```

Service	pid	Machine	Connected at	Encryption	Signing:
---------	-----	---------	--------------	------------	----------

```
.....
```

```
example 969 192.0.2.1 Thu Nov 1 10:00:00 2018 CEST - AES-128-CMAC
```

Locked files:

Pid	Uid	DenyMode	Access	R/W	Oplock	SharePath	Name	Time
-----	-----	----------	--------	-----	--------	-----------	------	------

```
.....
```

```
969 10000 DENY_WRITE 0x120089 RDONLY LEASE(RWH) /srv/samba/example file.txt Thu
Nov 1 10:00:00 2018
```

其它资源

详情请查看 **smbstatus(1)** man page。

3.19.10. 使用 smbtar 工具

smbtar 工具备份 SMB 共享的内容或它们的子目录,并将内容存储在 **tar** 归档中。或者,您可以将内容写入磁带设备。

先决条件

- 已安装 **samba-client** 软件包。

流程

- 使用以下命令备份 `//server/example/` 共享中的 **demo** 目录的内容,并将内容存储在 `/root/example.tar` 归档中：

```
# smbtar -s server -x example -u user_name -p password -t /root/example.tar
```

其它资源

详情请查看 **smbtar(1)** man page。

3.19.11. 使用 wbinfo 工具

wbinfo 工具查询并返回由 **winbindd** 服务创建和使用的信息。

先决条件

- 已安装 **samba-winbind-clients** 软件包。

流程

您可以使用 **wbinfo**,例如：

- 列出域用户：

```
# wbinfo -u
AD\administrator
AD\guest
...
```

- 列出域组：

```
# wbinfo -g
AD\domain computers
AD\domain admins
AD\domain users
...
```

- 显示用户的 SID：

```
# wbinfo --name-to-sid="AD\administrator"
S-1-5-21-1762709870-351891212-3141221786-500 SID_USER (1)
```

- 显示域和信任的信息：

```
# wbinfo --trusted-domains --verbose
Domain Name  DNS Domain      Trust Type Transitive In  Out
BUILTIN      None            Yes    Yes Yes
server       None            Yes    Yes Yes
DOMAIN1      domain1.example.com None    Yes    Yes Yes
DOMAIN2      domain2.example.com External No     Yes Yes
```

其它资源

详情请查看 **wbinfo(1)** man page。

3.20. 相关信息

- Red Hat Samba 软件包包括所有 Samba 命令的说明页以及安装该软件包的配置文件。例如,显示 **/etc/samba/smb.conf** 文件的 man page,用于解释在这个文件中可设置的所有配置参数：

```
# man smb.conf
```

- **/usr/share/docs/samba-version/** 目录包含由 Samba 项目提供的常规文档、示例脚本和 LDAP 模式文件。
- [Red Hat Cluster Storage 管理指南](#) :提供有关设置 Samba 和 Clustered Trivial Database(CDTB)的信息,以共享存储在 GlusterFS 卷中的目录。
- 有关在 Red Hat Enterprise Linux 中挂载 SMB 共享的详情, 请参考在 [Red Hat Enterprise Linux 中挂载 SMB 共享](#)。

第 4 章 导出 NFS 共享

作为系统管理员,您可以使用 NFS 服务器通过网络在系统中共享目录。

4.1. NFS 简介

这部分解释了 NFS 服务的基本概念。

网络文件系统(NFS)允许远程主机通过网络挂载文件系统,并像在本地挂载一样与那些文件系统交互。这可以让您将资源整合到网络的集中服务器中。

NFS 服务器引用 **/etc/exports** 配置文件来确定是否允许客户端访问任何导出的文件系统。一旦被验证,所有文件和目录操作都对用户有效。

4.2. 支持的 NFS 版本

这部分列出了 Red Hat Enterprise Linux 支持 NFS 版本及其特性。

目前, Red Hat Enterprise Linux 8 支持以下 NFS 主要版本 :

- 与 NFSv2 相比, NFS 版本 3 (NFSv3) 支持安全异步写入操作,并在处理错误时更可靠。它也支持 64 位文件大小和偏移,允许客户端访问超过 2 GB 文件数据。
- NFS 版本 4(NFSv4)通过防火墙和互联网工作,不再需要 **rpcbind** 服务,支持访问控制列表(ACL),并使用有状态的操作。

红帽不再支持 NFS 版本 2(NFSv2)。

默认 NFS 版本

Red Hat Enterprise Linux 8 中默认 NFS 版本为 4.2。NFS 客户端默认试图使用 NFSv4.2 挂载,并在服务器不支持 NFSv4.2 时回退到 NFSv4.1。之后挂载会返回 NFSv4.0,然后回退到 NFSv3。

次要 NFS 版本的特性

以下是 Red Hat Enterprise Linux 8 中的 NFSv4.2 的功能 :

服务器端复制

启用 NFS 客户端在不使用 **copy_file_range()** 系统调用处理网络资源的情况下有效复制数据。

稀疏文件

使文件有一个或者多个 *洞 (hole)* , 它们是不分配或者未初始化的数据块, 只由 0 组成。NFSv4.2 中的 **lseek()** 操作支持 **seek_hole()** 和 **seek_data()**, 它可让应用程序在稀疏文件中映射漏洞的位置。

保留空间

允许存储服务器保留空闲空间,这样可允许服务器耗尽空间。NFSv4.2 支持 **allocate()** 操作来保留空间, **deallocate()** 操作用于取消保留空间,以及 **fallocate()** 操作来预先分配文件中的空间或取消分配空间。

标记的 NFS

强制实施数据访问权利,并在客户端和服务器间为 NFS 文件系统单个文件启用 SELinux 标签。

布局增强

提供 **layoutstats()** 操作,它可让一些 Parallel NFS(pNFS)服务器收集更好的性能统计。

以下是 NFSv4.1 的功能 :

- 提高网络的性能和安全性,同时还包括对 pNFS 的客户端支持。

- 回调不再需要单独的 TCP 连接,它允许 NFS 服务器在其无法与客户端联系时:例如,当 NAT 或防火墙中断时。
- 提供完全一次语义(除了重启操作),防止以前出现的问题,如果回复丢失,操作被发送两次,则有时会返回不准确的结果。

4.3. NFSV3 和 NFSV4 中的 TCP 和 UDP 协议

NFSv4 需要通过 IP 网络运行的传输控制协议(TCP)。

NFSv3 也可以在以前的 Red Hat Enterprise Linux 版本中使用 User Datagram Protocol(UDP)。在 Red Hat Enterprise Linux 8 中不再支持通过 UDP 的 NFS。默认情况下,UDP 在 NFS 服务器中被禁用。

4.4. NFS 所需的服务

这部分列出了运行 NFS 服务器或挂载 NFS 共享所需的系统服务。Red Hat Enterprise Linux 会自动启动这些服务。

Red Hat Enterprise Linux 使用内核级支持和服务流程组合提供 NFS 文件共享。所有 NFS 版本都依赖于客户端和服务端间的远程过程调用(RPC)。要共享或者挂载 NFS 文件系统,下列服务根据所使用的 NFS 版本而定:

nfsd

为共享 NFS 文件系统请求的 NFS 服务器内核模块。

rpcbind

接受本地 RPC 服务的端口保留。然后会提供(或公告)这些端口,以便相应的远程 RPC 服务可以访问它们。**rpcbind** 服务响应 RPC 服务的请求,并设置到请求的 RPC 服务的连接。这不能与 NFSv4 一起使用。

rpc.mountd

NFS 服务器使用这个进程来处理来自 NFSv3 客户端的 **MOUNT** 请求。它检查所请求的 NFS 共享是否目前由 NFS 服务器导出,并且允许客户端访问它。如果允许挂载请求,**nfs-mountd** 服务会回复 Success 状态,并为这个 NFS 共享提供 File-Handle。

rpc.nfsd

这个过程启用了要定义的服务器公告的显式 NFS 版本和协议。它和 Linux 内核一起工作来满足 NFS 客户端的动态需求,比如在每次 NFS 客户端连接时提供服务器线程。此过程与 **nfs-server** 服务对应。

lockd

这是一个在客户端和服务端中运行的内核线程。它实现 Network Lock Manager(NLM)协议,该协议可启用 NFSv3 客户端在服务器中锁定文件。每当运行 NFS 服务器以及挂载 NFS 文件系统时,它会自动启动。

rpc.statd

这个过程实现了网络状态监视器(NSM)RPC 协议,它会在没有安全地关闭的情况下,在重启 NFS 服务器时通知 NFS 客户端。**rpc-statd** 服务由 **nfs-server** 服务自动启动,不需要用户配置。这不能与 NFSv4 一起使用。

rpc.rquotad

这个过程为远程用户提供用户配额信息。**rpc-rquotad** 服务由 **nfs-server** 服务自动启动,不需要用户配置。

rpc.idmapd

这个过程提供 NFSv4 客户端和服务端 upcalls,它映射在无线 NFSv4 名称(格式为 **user@domain** 的字符串)与本地 UID 和 GID 之间的映射。要让 **idmapd** 使用 NFSv4,必须配置 **/etc/idmapd.conf** 文件。至少应指定 **Domain** 参数,用于定义 NFSv4 映射域。如果 NFSv4 映射域与 DNS 域名相同,可

以跳过这个参数。客户端和服务端必须同意 NFSv4 映射域才能使 ID 映射正常工作。

只有 NFSv4 服务器使用 **rpc.idmapd**，它由 **nfs-idmapd** 服务启动。NFSv4 客户端使用基于密钥环的 **nfsidmap** 工具，它由内核 on-demand 调用来执行 ID 映射。如果 **nfsidmap** 存在问题，客户端会使用 **rpc.idmapd** 进行回调。

NFSv4 的 RPC 服务

挂载和锁定协议已合并到 NFSv4 协议中。该服务器还会侦听已知的 TCP 端口 2049。因此，NFSv4 不需要与 **rpcbind**、**lockd** 和 **rpc-statd** 服务交互。NFS 服务器中仍然需要 **nfs-mountd** 服务来设置导出，但不涉及任何在线（over-the-wire）操作。

其它资源

- 要配置只使用 NFSv4 的服务器（不需要 **rpcbind**），请参阅 [第 4.14 节“配置只使用 NFSv4 的服务器”](#)。

4.5. NFS 主机名格式

这部分论述了在挂载或导出 NFS 共享时用来指定主机的不同格式。

您可以使用以下格式指定主机：

单台机器

以下任意一种：

- 完全限定域名（可由服务器解析）
- 主机名（可由服务器解析）
- IP 地址。

IP 网络

以下格式之一有效：

- **a.b.c.d/z**，其中 **a.b.c.d** 是网络，**z** 是网络掩码中的位数，例如 **192.168.0.0/24**。
- **a.b.c.d/netmask**，其中 **a.b.c.d** 是网络，**netmask** 是子网掩码，例如：**192.168.100.8/255.255.255.0**。

Netgroups

@**group-name** 格式，其中 **group-name** 是 NIS netgroup 名称。

4.6. NFS 服务器配置

这部分论述了在 NFS 服务器中配置导出的语法和选项：

- 手动编辑 **/etc/exports** 配置文件
- 在命令行中使用 **exportfs** 工具

4.6.1. /etc/exports 配置文件

/etc/exports 文件控制将哪些文件系统导出到远程主机并指定选项。它遵循以下语法规则：

- 空白行将被忽略。
- 要添加一个注释，以井号（#）开始一个行。
- 您可以使用反斜杠（\）来换行长行。
- 每个导出的文件系统都应该独立。
- 所有在导出的文件系统后放置的授权主机列表都必须用空格分开。
- 每个主机的选项必须在主机标识符后直接放在括号中，没有空格分离主机和第一个括号。

导出条目

导出的文件系统的每个条目都有以下结构：

```
export host(options)
```

您还可以指定多个主机以及每个主机的特定选项。要做到这一点，在同一行中列出主机列表（以空格分隔），每个主机名带有其相关的选项（在括号中），如下所示：

```
export host1(options1) host2(options2) host3(options3)
```

在这个结构中：

export

导出的目录

主机

导出要共享的主机或网络

选项

用于主机的选项

例 4.1. 一个简单的 /etc/exports 文件

以最简单的形式, /etc/exports 文件只指定导出的目录以及允许访问它的主机：

```
/exported/directory bob.example.com
```

在这里, **bob.example.com** 可以从 NFS 服务器挂载 **/exported/directory/**。因为在这个示例中没有指定选项，所以 NFS 使用默认选项。

重要

`/etc/exports` 文件的格式非常精确，特别是对空格字符的使用。需要将导出的文件系统与主机、不同主机间使用空格分隔。但是，除了注释行外，文件中不应该包括其他空格。

例如，下面两行并不具有相同的意义：

```
/home bob.example.com(rw)
/home bob.example.com (rw)
```

第一行只允许来自 **bob.example.com** 读写访问 `/home` 目录的用户。第二行允许来自 **bob.example.com** 的用户将目录作为只读（默认值）挂载,而其他世界则可以挂载它读/写。

默认选项

导出条目的默认选项有：

ro

导出的文件系统是只读的。远程主机无法更改文件系统中共享的数据。要允许主机更改文件系统（即读写），指定 `rw` 选项。

sync

在将之前的请求所做的更改写入磁盘前，NFS 服务器不会回复请求。要启用异步写入功能，请指定选项 `async`。

wdelay

如果 NFS 服务器预期另外一个写入请求即将发生，则 NFS 服务器会延迟写入磁盘。这可以提高性能，因为它可减少单独写入命令访问磁盘的次数,从而降低写入开销。要禁用此选项，指定 `no_wdelay` 选项，该选项仅在同时指定默认同步选项时才可用。

root_squash

这样可防止远程连接（而不是本地）的 `root` 用户具有 `root` 权限；相反,NFS 服务器会为他们分配用户 ID `nobody`。这个远程 `root` 用户的"评估"功能可以有效地把这个功能提供给最小的本地用户,从而避免在远程服务器中出现未授权写入的问题。要禁用 `root squash`，指定 `no_root_squash` 选项。

要挂起每个远程用户（包括 `root` 用户），使用 `all_squash` 选项。要指定 NFS 服务器应分配给特定主机的远程用户的用户和组 ID,使用 `anonuid` 和 `anongid` 选项,如下所示：

```
export host(anonuid=uid,anongid=gid)
```

在这里, `uid` 和 `gid` 分别是用户 ID 号和组 ID 号。`anonuid` 和 `anongid` 选项允许您为远程 NFS 用户创建特殊的用户和组群帐户。

默认情况下,Red Hat Enterprise Linux 的 NFS 支持访问控制列表(ACL)。要禁用此功能，在导出文件系统时指定 `no_acl` 选项。

默认和覆盖选项

每个导出的文件系统的默认值都必须被显式覆盖。例如：如果没有指定 `rw` 选项,则以只读方式共享导出的文件系统。以下是 `/etc/exports` 的示例行，可覆盖两个默认选项：

```
/another/exported/directory 192.168.0.3(rw,async)
```

在这个示例中, **192.168.0.3** 可以挂载 `/another/exported/directory/` 读写,磁盘的所有写入操作都是异步的。

4.6.2. exportfs 工具

exportfs 工具可让 root 用户在不重启 NFS 服务的情况下有选择地导出或取消导出目录。当给出正确的选项时, **exportfs** 工具会将导出的文件系统写入 **/var/lib/nfs/xtab**。因为 **nfs-mountd** 服务在决定访问文件系统的权限时指向 **xtab** 文件,所以对导出的文件系统列表的更改会立即生效。

常用的 exportfs 选项

以下是 **exportfs** 常用的选项列表：

-r

通过在 **/var/lib/nfs/etab** 中创建新导出列表, 导出 **/etc/exports** 中列出的所有目录。这个选项通过对 **/etc/exports** 所做的任何更改有效刷新导出列表。

-a

根据将其它选项传递给 **exportfs** 导致所有目录被导出或取消导出。如果没有指定其他选项, **exportfs** 导出 **/etc/exports** 中指定的所有文件系统。

-o file-systems

指定导出没有在 **/etc/exports** 中列出的目录。使用要导出的额外文件系统替换 *file-systems*。这些文件系统的格式必须与在 **/etc/exports** 中指定的相同。这个选项通常用于测试导出的文件系统,然后将其永久添加到导出的文件系统列表中。

-i

忽略 **/etc/exports**; 只有命令行给出的选项才会用于定义导出的文件系统。

-u

取消导出所有共享目录。**exportfs -ua** 命令挂起 NFS 文件共享, 同时保持所有 NFS 服务。要重新启用 NFS 共享, 请使用 **exportfs -r**。

-v

详细操作,在执行 **exportfs** 命令时,会详细显示要导出或取消导出的文件系统。

如果没有将选项传递给 **exportfs** 工具, 它会显示当前导出的文件系统列表。

其它资源

- 有关指定主机名的不同方法的详情请参考 第 4.5 节 “NFS 主机名格式”。
- 有关导出选项的完整列表,请查看 **exports(5)** man page。
- 有关 **exportfs** 工具程序的详情请参考 **exportfs(8)** man page。

4.7. NFS 和 RPCBIND

本节介绍 NFSv3 所需的 **rpcbind** 服务的目的。

rpcbind 服务将远程过程调用(RPC)服务映射到它们侦听的端口。RPC 进程启动时通知 **rpcbind**,注册他们正在侦听的端口以及它们期望服务的 RPC 程序号。然后客户端系统使用特定的 RPC 程序号码在服务器上联系 **rpcbind**。**rpcbind** 服务将客户端重定向到正确的端口号,使其可以与请求的服务通信。

因为基于 RPC 的服务依赖于 **rpcbind** 与传入客户端请求进行所有连接,所以必须在任何这些服务启动前提供 **rpcbind**。

rpcbind 的访问控制规则会影响所有基于 RPC 的服务。另外, 也可以为每个 NFS RPC 守护进程指定访问控制规则。

其它资源

- 有关访问控制规则的语法，请查看 **rpc.mountd(8)** 和 **rpc.statd(8)** man page。

4.8. 安装 NFS

这个过程安装挂载或导出 NFS 共享所需的所有软件包。

流程

- 安装 **nfs-utils** 软件包：

```
# yum install nfs-utils
```

4.9. 启动 NFS 服务器

这个步骤描述了如何启动 NFS 服务器,这是导出 NFS 共享所必需的。

先决条件

- 对于支持 NFSv2 或者 NFSv3 连接的服务器, **rpcbind** 服务必须正在运行。要验证 **rpcbind** 是否活跃，请使用以下命令：

```
$ systemctl status rpcbind
```

如果停止该服务，启动并启用该服务：

```
$ systemctl enable --now rpcbind
```

流程

- 要启动 NFS 服务器并使其在引导时自动启动,请使用以下命令：

```
# systemctl enable --now nfs-server
```

其它资源

- 要配置只使用 NFSv4 的服务器（不需要 **rpcbind**），请参阅 [第 4.14 节“配置只使用 NFSv4 的服务器”](#)。

4.10. NFS 和 RPCBIND 故障排除

因为 **rpcbind** 服务在 RPC 服务与用来与它们通信的端口号之间进行协调,所以在故障排除时使用 **rpcbind** 查看当前 RPC 服务的状态会很有用。**rpcinfo** 工具显示每个基于 RPC 的服务都带有端口号、RPC 程序号、版本号以及 IP 协议类型（TCP 或者 UDP）。

流程

1. 要确定为 **rpcbind** 启用了正确的基于 NFS RPC 的服务，使用以下命令：

```
# rpcinfo -p
```

例 4.2. **rpcinfo -p** 命令输出

下面是一个这个命令的输出示例：

```

program vers proto  port  service
100000  4  tcp   111  portmapper
100000  3  tcp   111  portmapper
100000  2  tcp   111  portmapper
100000  4  udp   111  portmapper
100000  3  udp   111  portmapper
100000  2  udp   111  portmapper
100005  1  udp  20048 mountd
100005  1  tcp  20048 mountd
100005  2  udp  20048 mountd
100005  2  tcp  20048 mountd
100005  3  udp  20048 mountd
100005  3  tcp  20048 mountd
100024  1  udp  37769 status
100024  1  tcp  49349 status
100003  3  tcp   2049 nfs
100003  4  tcp   2049 nfs
100227  3  tcp   2049 nfs_acl
100021  1  udp  56691 nlockmgr
100021  3  udp  56691 nlockmgr
100021  4  udp  56691 nlockmgr
100021  1  tcp  46193 nlockmgr
100021  3  tcp  46193 nlockmgr
100021  4  tcp  46193 nlockmgr

```

如果其中一个 NFS 服务无法正确启动, **rpcbind** 将无法将客户端的 RPC 请求映射到正确的端口。

2. 在很多情况下,如果 **rpcinfo** 输出中没有 NFS,重启 NFS 会导致该服务在 **rpcbind** 中正确注册并开始工作：

```
# systemctl restart nfs-server
```

其它资源

- 有关 **rpcinfo** 选项列表, 请查看 **rpcinfo(8)** man page。
- 要配置只使用 NFSv4 的服务器（不需要 **rpcbind**），请参阅 [第 4.14 节“配置只使用 NFSv4 的服务器”](#)。

4.11. 将 NFS 服务器配置为在防火墙后运行

NFS 需要 **rpcbind** 服务, 它动态为 RPC 服务分配端口, 并可能导致配置防火墙规则的问题。这个步骤描述了如何将 NFS 服务器配置为在防火墙后工作。

流程

1. 要允许客户端访问防火墙后面的 NFS 共享,在 **/etc/nfs.conf** 文件的 **[mountd]** 部分设置 RPC 服务在哪些端口中运行：

```
[mountd]
```

```
port=port-number
```

这会在 **rpc.mount** 命令行中添加 **-p port-number** 选项：**rpc.mount -p port-number**。

2. 要允许客户端访问防火墙后面的 NFS 共享,在 NFS 服务器中运行以下命令来配置防火墙：

```
firewall-cmd --permanent --add-service mountd
firewall-cmd --permanent --add-service rpc-bind
firewall-cmd --permanent --add-service nfs
firewall-cmd --permanent --add-port=<mountd-port>/tcp
firewall-cmd --permanent --add-port=<mountd-port>/udp
firewall-cmd --reload
```

在命令中，将 **<mountd-port>** 替换为预期的端口或端口范围。当指定端口范围时,请使用 **--add-port=<mountd-port>-<mountd-port>/udp** 语法。

3. 要允许 NFSv4.0 回调通过防火墙,设置 **/proc/sys/fs/nfs/nfs_callback_tcpport** 并允许服务器连接到客户端上的端口。
NFSv4.1 或更高版本不需要这一步, **mountd**、**statd** 和 **lockd**, 的其他端口在纯 NFSv4 环境中不需要。
4. 要指定 RPC 服务 **nlockmgr** 使用的端口, 在 **/etc/modprobe.d/lockd.conf** 文件中设置 **nlm_tcpport** 和 **nlm_udpport** 选项的端口号。
5. 重启 NFS 服务器：

```
# systemctl restart nfs-server
```

如果 NFS 无法启动, 请检查 **/var/log/messages**。通常, 如果指定了已在使用的端口号, NFS 将无法启动。

6. 确认更改生效：

```
# rpcinfo -p
```

其它资源

- 要配置只使用 NFSv4 的服务器（不需要 **rpcbind**），请参阅 [第 4.14 节 “配置只使用 NFSv4 的服务器”](#)。

4.12. 通过防火墙导出 RPC 配额

如果您导出使用磁盘配额的文件系统,可以使用配额远程过程调用(RPC)服务为 NFS 客户端提供磁盘配额数据。

流程

1. 启用并启动 **rpc-rquotad** 服务：

```
# systemctl enable --now rpc-rquotad
```



注意

如果已启用，**rpc-rquotad** 服务会在启动 **nfs-server** 服务后自动启动。

2. 要使配额 RPC 服务在防火墙后访问,需要打开 TCP (如果启用了 UDP,或使用 UDP) 端口 875。默认端口号在 **/etc/services** 文件中定义。
您可以通过在 **-p port-number** 文件中的 **RPCRQUOTADOPTS** 变量中添加 **/etc/sysconfig/rpc-rquotad** 来覆盖默认端口号。
3. 默认情况下, 远程主机只能读配额。如果要允许客户端设置配额,请将 **-S** 选项附加到 **RPCRQUOTADOPTS** 文件中的 **/etc/sysconfig/rpc-rquotad** 变量中。
4. 重启 **rpc-rquotad** 以使 **/etc/sysconfig/rpc-rquotad** 文件中的更改生效 :

```
# systemctl restart rpc-rquotad
```

4.13. 通过 RDMA(NFSORDMA)启用 NFS

如果存在 RDMA 功能的硬件,远程直接访问(RDMA)服务会在 Red Hat Enterprise Linux 8 中自动正常工作。

流程

1. 安装 **rdma-core** 软件包 :

```
# yum install rdma-core
```

2. 要启用自动载入 NFSoRDMA 服务器模块, 请在配置文件 **/etc/rdma/rdma.conf** 的一个新行中添加 **SVCRDMA_LOAD=yes** 选项。
/etc/nfs.conf 文件的 **[nfsd]** 部分中的 **rdma=20049** 选项指定 NFSoRDMA 服务侦听客户端的端口号。RFC5667 标准指定,在通过 RDMA 提供 NFSv4 服务时,服务器必须侦听端口 **20049**。

/etc/rdma/rdma.conf 文件包含一行,默认设置 **XPRTRDMA_LOAD=yes** 选项,它请求 **rdma** 服务载入 NFSoRDMA 客户端模块。

3. 重启 **nfs-server** 服务 :

```
# systemctl restart nfs-server
```

其它资源

- RFC5667 标准 : <https://tools.ietf.org/html/rfc5667>。

4.14. 配置只使用 NFSV4 的服务器

作为 NFS 服务器管理员,您可以将 NFS 服务器配置为只支持 NFSv4,这样可最小化系统中打开端口和运行服务的数量。

4.14.1. 只使用 NFSv4 的服务器的的好处和缺陷

这部分论述了将 NFS 服务器配置为只支持 NFSv4 的优点和缺陷。

默认情况下，NFS 服务器在 Red Hat Enterprise Linux 8 中支持 NFSv3 和 NFSv4 连接。但是，您还可以将 NFS 配置为只支持 NFS 版本 4.0 及更新的版本。这可最小化系统中打开端口和运行的服务的数量，因为 NFSv4 不需要 **rpcbind** 服务侦听网络。

当您的 NFS 服务器配置为只读 NFSv4 时,试图使用 NFSv3 挂载共享的客户端会失败,并显示如下错误：

```
Requested NFS version or transport protocol is not supported.
```

另外，您还可以禁用对 **RPCBIND**、**MOUNT** 和 **NSM** 协议调用的监听，这些在 NFSv4 中不需要。

禁用这些额外选项的影响有：

- 试图使用 NFSv3 从服务器挂载共享的客户端变得无响应。
- NFS 服务器本身无法挂载 NFSv3 文件系统。

4.14.2. 将 NFS 服务器配置为只支持 NFSv4

这个步骤描述了如何配置 NFS 服务器来支持 NFS 版本 4.0 及更新的版本。

流程

1. 通过在 **/etc/nfs.conf** 配置文件的 **[nfsd]** 部分添加以下行来禁用 NFSv3：

```
[nfsd]
vers3=no
```

2. （可选）禁用监听 **RPCBIND**、**MOUNT** 和 **NSM** 协议调用，这些在 NFSv4 中不需要。禁用相关服务：

```
# systemctl mask --now rpc-statd.service rpcbind.service rpcbind.socket
```

3. 重启 NFS 服务器：

```
# systemctl restart nfs-server
```

一旦启动或重启 NFS 服务器，这些改变就会生效。

4.14.3. 验证只读 NFSv4 配置

这个步骤描述了如何使用 **netstat** 实用程序验证您的 NFS 服务器是否在 NFSv4 模式中配置。

流程

- 使用 **netstat** 实用程序列出侦听 TCP 和 UDP 协议的服务：

```
# netstat --listening --tcp --udp
```

例 4.3. 只输出 NFSv4 服务器

以下是只使用 NFSv4 的服务器的 **netstat** 输出示例；也禁用了侦听 **RPCBIND**、**MOUNT** 和 **NSM**。在这里，**nfs** 是唯一侦听 NFS 服务：

```
# netstat --listening --tcp --udp
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:ssh	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:nfs	0.0.0.0:*	LISTEN
tcp6	0	0	:::ssh	:::*	LISTEN
tcp6	0	0	:::nfs	:::*	LISTEN
udp	0	0	localhost.locald:bootpc	0.0.0.0:*	

例 4.4. 配置只读 NFSv4 服务器前的输出

而配置只使用 NFSv4 的服务器前的 **netstat** 输出会包括 **sunrpc** 和 **mountd** 服务：

```
# netstat --listening --tcp --udp
```

Active Internet connections (only servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	0.0.0.0:ssh	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:40189	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:46813	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:nfs	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:sunrpc	0.0.0.0:*	LISTEN
tcp	0	0	0.0.0.0:mountd	0.0.0.0:*	LISTEN
tcp6	0	0	:::ssh	:::*	LISTEN
tcp6	0	0	:::51227	:::*	LISTEN
tcp6	0	0	:::nfs	:::*	LISTEN
tcp6	0	0	:::sunrpc	:::*	LISTEN
tcp6	0	0	:::mountd	:::*	LISTEN
tcp6	0	0	:::45043	:::*	LISTEN
udp	0	0	localhost:1018	0.0.0.0:*	
udp	0	0	localhost.locald:bootpc	0.0.0.0:*	
udp	0	0	0.0.0.0:mountd	0.0.0.0:*	
udp	0	0	0.0.0.0:46672	0.0.0.0:*	
udp	0	0	0.0.0.0:sunrpc	0.0.0.0:*	
udp	0	0	0.0.0.0:33494	0.0.0.0:*	
udp6	0	0	:::33734	:::*	
udp6	0	0	:::mountd	:::*	
udp6	0	0	:::sunrpc	:::*	
udp6	0	0	:::40243	:::*	

4.15. 相关信息

- Linux NFS wiki: https://linux-nfs.org/wiki/index.php/Main_Page

第 5 章 保护 NFS

要最小化 NFS 安全风险并在服务器中保护数据,在服务器中导出 NFS 文件系统或者将其挂载到客户端时,请考虑以下部分。

5.1. 带有 AUTH_SYS 和导出控制的 NFS 安全性

NFS 提供以下传统选项来控制对导出文件的访问：

- 服务器限制哪些主机可以通过 IP 地址或主机名挂载哪些文件系统。
- 服务器像针对本地用户一样,为 NFS 客户端中的用户强制使用文件系统权限。通常,NFS 使用 **AUTH_SYS** 调用信息（也称为 **AUTH_UNIX**）进行此操作,该消息依赖于客户端来说明用户的 UID 和 GID。请注意，这意味着恶意或者错误配置的客户端可能会轻松地利用这个问题，导致用户可以访问不应该被访问的文件。

为限制潜在的风险,管理员通常会将只读或 squash 用户的访问限制到常见用户和组群 ID。不幸的是,这些解决方案会阻止 NFS 共享以最初预期的方式被使用。

另外,如果攻击者获得对导出 NFS 文件系统的系统使用的 DNS 服务器的控制,他们可将与特定主机名或完全限定域名关联的系统指向未授权机器。此时,未授权机器是允许挂载 NFS 共享的系统,因为没有交换用户名或密码信息来为 NFS 挂载提供额外的安全性。

当通过 NFS 导出目录时,通配符应该被显式使用,因为通配符范围可能包含更多系统。

其它资源

- 要保护 NFS 和 **rpcbind**,请使用,例如 **nftables** 和 **firewalld**。有关配置这些框架的详情请参考 **nft(8)** 和 **firewalld-cmd(1)** man page。

5.2. 使用 AUTH_GSS 的 NFS 安全性

NFS 的所有版本都支持 **RPCSEC_GSS** 和 Kerberos 机制。

与 **AUTH_SYS** 不同,使用 **RPCSEC_GSS** Kerberos 机制,服务器不依赖于客户端正确代表哪个用户正在访问该文件。反之,加密用于向服务器验证用户,这可防止恶意客户端在没有该用户的 Kerberos 凭证的情况下模拟用户。使用 **RPCSEC_GSS** Kerberos 机制是保护挂载的最直接方法,因为配置了 Kerberos 后不需要额外的设置。

5.3. 配置 NFS 服务器和客户端使用 KERBEROS

Kerberos 是一个网络身份验证系统,客户端和服务端可以使用对称加密和可信第三方 KDC 相互验证。红帽建议使用 Identity Management(IdM)来设置 Kerberos。

先决条件

- Kerberos Key Distribution(**KDC**)已安装并配置。

流程

- 在 NFS 服务器端创建 **nfs/hostname.domain@REALM** 主体。
 - 在服务器和客户端中创建 **host/hostname.domain@REALM** 主体。

- 将对应的密钥添加到客户端和服务器的 keytab 中。
2. 在服务器端，使用 **sec=** 选项启用所需的安全类型。启用所有安全类型和非加密挂载：

```
/export *(sec=sys:krb5:krb5i:krb5p)
```

与 **sec=** 选项一起使用的有效安全类型为：

- **sys**：没有加密保护（默认设置）
 - **krb5**：仅限身份验证
 - **krb5i**：完整性保护
 - **krb5p**：隐私保护
3. 在客户端，添加 **sec=krb5**（**sec=krb5i** 或 **sec=krb5p**，具体取决于设置）到挂载选项：

```
# mount -o sec=krb5 server:/export /mnt
```

其它资源

- 如果您需要在 Kerberos-secured NFS 共享中以 root 用户身份写入文件,并在这些文件中保留 root 所有权,请参考 <https://access.redhat.com/articles/4040141>。请注意，我们不推荐进行此配置。
- 有关 NFS 配置的详情请参考 `export(5)` 和 `nfs(5)` man page。

5.4. NFSV4 安全选项

NFSv4 包含基于 Microsoft Windows NT 型号而不是 POSIX 模型的 ACL 支持,因为 Microsoft Windows NT 模型的功能和广泛部署。

NFSv4 的另一个重要安全功能是，删除使用 **MOUNT** 协议挂载文件系统。**MOUNT** 协议会因为协议处理文件的方式造成安全隐患。

5.5. 挂载的 NFS 导出的文件权限

一旦远程主机以读写模式挂载 NFS 文件系统,那么每个共享文件的唯一保护就是它的权限。如果两个共享相同用户 ID 值的用户在不同客户端系统中挂载相同的 NFS 文件系统,他们可以修改相互的文件。另外,任何人在客户端系统中以 root 用户身份登录,可以使用 **su -** 命令访问 NFS 共享的任何文件。

默认情况下，Red Hat Enterprise Linux 的 NFS 支持访问控制列表（ACL）。红帽建议启用此功能。

默认情况下，NFS 在导出文件系统时使用 *root squashing*。这会将任何以本地机器上的 root 用户身份访问 NFS 共享的用户 ID 设置为 **nobody**。Root squashing 由默认选项 **root_squash** 控制; 如需了解更多有关此选项的信息，请参阅 [第 4.6 节“NFS 服务器配置”](#)。

当以只读形式导出 NFS 共享时,请考虑使用 **all_squash** 选项。这个选项使每个用户访问导出的文件系统时使用 **nobody** 用户的用户 ID。

第 6 章 在 NFS 中启用 PNFS SCSI 布局

您可以将 NFS 服务器和客户端配置为使用 pNFS SCSI 布局访问数据。

先决条件

- 客户端和服务端必须能够向同一个块设备发送 SCSI 命令。就是说块设备必须位于共享的 SCSI 总线中。
- 块设备必须包含 XFS 文件系统。
- SCSI 设备必须支持 SCSI Persistent Reservations，如 SCSI-3 Primary Commands 规格中所述。

6.1. PNFS 技术

pNFS 构架提高了 NFS 的可伸缩性。当服务器实现 pNFS 时，客户端可以同时通过多个服务器访问数据。这可提高性能。

pNFS 支持 RHEL 中的以下存储协议或布局：

- 文件
- Flexfiles
- SCSI

6.2. PNFS SCSI 布局

SCSI 布局基于 pNFS 块布局的工作。布局在 SCSI 设备中定义。它包含一系列固定大小块作为逻辑单元 (LU)，它必须能够支持 SCSI 持久保留。LU 设备识别通过其 SCSI 设备识别。

在涉及对文件的单一客户端访问时间较长的用例中，pNFS SCSI 可以正常工作。例如：邮件服务器或者虚拟机。

客户端和服务端间的操作

当 NFS 客户端从文件读取或写入该文件时，客户端将执行 **LAYOUTGET** 操作。服务器会使用文件在 SCSI 设备中的位置进行响应。客户端可能需要执行额外的 **GETDEVICEINFO** 操作来确定要使用哪个 SCSI 设备。如果这些操作正常工作，客户端可以直接向 SCSI 设备发出 I/O 请求，而不是向服务器发送 **READ** 和 **WRITE** 操作。

客户端间的错误或竞争可能会导致服务器进入布局，而不向客户端发出它们。在这种情况下，客户端会返回向服务器发出 **READ** 和 **WRITE** 操作，而不是直接将 I/O 请求发送到 SCSI 设备。

要监控操作，请参阅 [第 6.7 节“监控 pNFS SCSI 布局功能”](#)。

设备保留

pNFS SCSI 通过分配保留来处理保护。在服务器为客户端签发布局前，它会保留 SCSI 设备以确保只有注册的客户端才可以访问该设备。如果客户端可以向那个 SCSI 设备发出命令，但没有在该设备中注册，则该设备上客户端的许多操作会失败。例如：如果服务器没有为客户端提供布局，客户端的 **blkid** 命令将无法显示 XFS 文件系统的 UUID。

服务器不会删除其自身的持久性保留。这样可在重启客户端和服务端后保护该设备中的文件系统的数据。为了重新使用 SCSI 设备，您可能需要手动删除 NFS 服务器中的持久性保留。

6.3. 检查与 PNFS 兼容的 SCSI 设备

这个过程检查 SCSI 设备是否支持 pNFS SCSI 布局。

先决条件

- 安装 **sg3_utils** 软件包：

```
# yum install sg3_utils
```

流程

- 在服务器和客户端中检查正确的 SCSI 设备支持：

```
# sg_persist --in --report-capabilities --verbose path-to-scsi-device
```

确保设置了 *Persist Through Power Los Active* (**PTPL_A**) 位。

例 6.1. 支持 pNFS SCSI 的 SCSI 设备

以下是支持 pNFS SCSI 的 SCSI 设备的 **sg_persist** 输出示例。PTPL_A 位报告 1。

```
inquiry cdb: 12 00 00 00 24 00
Persistent Reservation In cmd: 5e 02 00 00 00 00 20 00 00
LIO-ORG block11      4.0
Peripheral device type: disk
Report capabilities response:
Compatible Reservation Handling(CRH): 1
Specify Initiator Ports Capable(SIP_C): 1
All Target Ports Capable(ATP_C): 1
Persist Through Power Loss Capable(PTPL_C): 1
Type Mask Valid(TMV): 1
Allow Commands: 1
Persist Through Power Loss Active(PTPL_A): 1
Support indicated in Type mask:
Write Exclusive, all registrants: 1
Exclusive Access, registrants only: 1
Write Exclusive, registrants only: 1
Exclusive Access: 1
Write Exclusive: 1
Exclusive Access, all registrants: 1
```

其它资源

- **sg_persist(8)** man page

6.4. 在服务器中设置 PNFS SCSI

这个过程将 NFS 服务器配置为导出 pNFS SCSI 布局。

流程

1. 在服务器中挂载在 SCSI 设备中创建的 XFS 文件系统。
2. 将 NFS 服务器配置为导出 NFS 版本 4.1 或更高版本。在文件 `/etc/nfs.conf` 的 `[nfsd]` 部分设置以下选项：

```
[nfsd]
vers4.1=y
```

3. 使用 **pnfs** 选项将 NFS 服务器配置为通过 NFS 导出 XFS 文件系统：

例 6.2. /etc/exports 中的条目导出 pNFS SCSI

`/etc/exports` 配置文件中的以下条目将挂载到 `/exported/directory/` 的文件系统导出到 `allowed.example.com` 客户端，作为 pNFS SCSI 布局：

```
/exported/directory allowed.example.com(pnfs)
```

其它资源

- 有关配置 NFS 服务器的详情请参考 [第 4 章 导出 NFS 共享](#)。

6.5. 在客户端中设置 PNFS SCSI

这个过程将 NFS 客户端配置为挂载 pNFS SCSI 布局。

先决条件

- NFS 服务器被配置为通过 pNFS SCSI 导出 XFS 文件系统。请参阅 [第 6.4 节 “在服务器中设置 pNFS SCSI”](#)。

流程

- 在客户端中使用 NFS 版本 4.1 或更高版本挂载导出的 XFS 文件系统：

```
# mount -t nfs -o nfsvers=4.1 host:/remote/export /local/directory
```

不要在沒有 NFS 的情况下直接挂载 XFS 文件系统。

其它资源

- 有关挂载 NFS 共享的详情，请参考 [挂载 NFS 共享](#)。

6.6. 在服务器中释放 PNFS SCSI 保留

此流程释放 NFS 服务器在 SCSI 设备中拥有的持久保留。这可让您在不再需要导出 pNFS SCSI 时重新使用 SCSI 设备。

您必须从服务器中删除保留。它不能从不同的 IT Nexus 中删除。

先决条件

- 安装 **sg3_utils** 软件包：

```
# yum install sg3_utils
```

流程

1. 在服务器上查询现有保留：

```
# sg_persist --read-reservation path-to-scsi-device
```

例 6.3. 在 /dev/sda 中查询保留

```
# sg_persist --read-reservation /dev/sda

LIO-ORG block_1      4.0
Peripheral device type: disk
PR generation=0x8, Reservation follows:
Key=0x1000000000000000
scope: LU_SCOPE, type: Exclusive Access, registrants only
```

2. 删除服务器上的现有注册：

```
# sg_persist --out \
    --release \
    --param-rk=reservation-key \
    --prout-type=6 \
    path-to-scsi-device
```

例 6.4. 删除 /dev/sda 中的保留

```
# sg_persist --out \
    --release \
    --param-rk=0x1000000000000000 \
    --prout-type=6 \
    /dev/sda

LIO-ORG block_1      4.0
Peripheral device type: disk
```

其它资源

- **sg_persist(8)** man page

6.7. 监控 PNFS SCSI 布局功能

您可以监控 pNFS 客户端和服务端是否交换了正确的 pNFS SCSI 操作,或者它们回退到常规 NFS 操作中。

先决条件

- 配置了 pNFS SCSI 客户端和服务端。

6.7.1. 使用 `nfsstat` 从服务器检查 pNFS SCSI 操作

此流程使用 `nfsstat` 工具监控服务器中的 pNFS SCSI 操作。

流程

1. 监控服务器中服务的操作：

```
# watch --differences \
    "nfsstat --server | egrep --after-context=1 read\|write\|layout"

Every 2.0s: nfsstat --server | egrep --after-context=1 read\|write\|layout

putrootfh  read      readdir  readlink  remove  rename
2          0% 0      0% 1      0% 0      0% 0      0% 0      0%
--
setctidconf verify  write      rellockowner bc_ctl  bind_conn
0          0% 0      0% 0      0% 0      0% 0      0% 0      0%
--
getdevlist layoutcommit layoutget  layoutreturn secinfoonam sequence
0          0% 29      1% 49      1% 5      0% 0      0% 2435  86%
```

2. 客户端和服务端在以下情况下使用 pNFS SCSI 操作：

- **layoutget**、**layoutreturn** 和 **layoutcommit** 计数器递增。这意味着服务器提供布局。
- 服务器 **read** 和 **write** 计数器不会递增。这意味着客户端正在直接向 SCSI 设备执行 I/O 请求。

6.7.2. 使用 `mountstats` 检查客户端中的 pNFS SCSI 操作

这个过程使用 `/proc/self/mountstats` 文件来监控来自客户端的 pNFS SCSI 操作。

流程

1. 列出每个挂载的操作计数器：

```
# cat /proc/self/mountstats \
    | awk /scsi_lun_0/,/^$/ \
    | egrep device\|READ\|WRITE\|LAYOUT

device 192.168.122.73:/exports/scsi_lun_0 mounted on /mnt/rhel7/scsi_lun_0 with fstype
nfs4 statvers=1.1
nfsv4:
bm0=0xfdfbfff,bm1=0x40f9be3e,bm2=0x803,acl=0x3,sessions,pnfs=LAYOUT_SCSI
    READ: 0 0 0 0 0 0 0
    WRITE: 0 0 0 0 0 0 0
    READLINK: 0 0 0 0 0 0 0
    READDIR: 0 0 0 0 0 0 0
    LAYOUTGET: 49 49 0 11172 9604 2 19448 19454
    LAYOUTCOMMIT: 28 28 0 7776 4808 0 24719 24722
    LAYOUTRETURN: 0 0 0 0 0 0 0
    LAYOUTSTATS: 0 0 0 0 0 0 0
```

2. 在结果中：

- **LAYOUT** 统计指示客户端和服务端使用 pNFS SCSI 操作的请求。
- **READ** 和 **WRITE** 统计代表客户端和服务端回退到 NFS 操作的请求。

第 7 章 配置 SQUID 缓存代理服务器

squid 是一个代理服务器,它缓存内容以更快地减少带宽和负载网页。本章论述了如何将 Squid 设置为 HTTP、HTTPS 和 FTP 协议的代理,以及验证和限制访问。

7.1. 将 SQUID 设置为没有身份验证的缓存代理

这部分论述了 Squid 的基本配置在没有身份验证的情况下作为缓存代理。此流程会根据 IP 范围限制对代理的访问。

先决条件

- 该流程假设 `/etc/squid/squid.conf` 文件由 **squid** 软件包提供。如果您在之前编辑了这个文件,请删除该文件并重新安装该软件包。

流程

1. 安装 **squid** 软件包：

```
# yum install squid
```

2. 编辑 `/etc/squid/squid.conf` 文件：

- a. 修改 **localnet** 访问控制列表(ACL),以匹配应允许使用代理的 IP 范围：

```
acl localnet src 192.0.2.0/24
acl localnet 2001:db8:1::/64
```

默认情况下, `/etc/squid/squid.conf` 文件包含 **http_access allow localnet** 规则,它允许使用 **localnet** ACL 中指定的所有 IP 范围的代理。请注意,您必须在 **http_access allow localnet** 规则前指定所有 **localnet** ACL。



重要

删除所有与您的环境不匹配的现有 **acl localnet** 条目。

- b. 以下 ACL 存在于默认配置中,将 **443** 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议,请为每个端口添加 ACL:

```
acl SSL_ports port port_number
```

- c. 更新 **acl Safe_ports** 规则列表,以配置 Squid 可以建立连接的端口。例如：要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源,请在配置中保留以下 **acl Safe_ports** 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```


默认情况下,配置包含 **http_access deny !Safe_ports** 规则,用于定义对 **Safe_ports** ACL 中没有定义的端口的访问拒绝。

- d. 在 **cache_dir** 参数中配置缓存类型、缓存目录的路径、缓存大小以及其它缓存类型设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- squid 使用 **ufs** 缓存类型。
 - squid 将缓存存储在 **/var/spool/squid/** 目录中。
 - 缓存增长到 **10000** MB。
 - squid 在 **/var/spool/squid/** 目录中创建 **16** level-1 子目录。
 - squid 在每个级别-1 目录中创建 **256** 子目录。
- 如果您没有设置 **cache_dir** 指令, Squid 会将缓存保存在内存中。

3. 如果您在 **cache_dir** 参数中设置了与 **/var/spool/squid/** 不同的缓存目录：

- a. 创建缓存目录：

```
# mkdir -p path_to_cache_directory
```

- b. 配置缓存目录的权限：

```
# chown squid:squid path_to_cache_directory
```

- c. 如果您以 **enforcing** 模式运行 SELinux,请为缓存目录设置 **squid_cache_t** 上下文：

```
# semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
# restorecon -Rv path_to_cache_directory
```

如果您的系统中没有 **semanage** 工具,安装 **polycoreutils-python-utils** 软件包。

4. 在防火墙中打开 **3128** 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

5. 启用并启动 **squid** 服务：

```
# systemctl enable --now squid
```

验证步骤

要验证代理是否正常工作,使用 **curl** 工具下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" -x "proxy.example.com:3128"
```

如果 **curl** 没有显示任何错误,且将 **index.html** 文件下载到当前目录中,则代理可以正常工作。

7.2. 使用 LDAP 身份验证将 SQUID 设置为缓存代理

本节论述了 Squid 的基本配置作为使用 LDAP 验证用户的缓存代理。此流程配置仅经过身份验证的用户可以使用代理。

先决条件

- 该流程假设 `/etc/squid/squid.conf` 文件由 **squid** 软件包提供。如果您在之前编辑了这个文件，请删除该文件并重新安装该软件包。
- LDAP 目录中有一个服务用户，如 **uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com**。Squid 只使用此帐户搜索验证用户。如果存在身份验证用户，Squid 会以此用户的身份绑定到该目录以验证身份验证。

流程

1. 安装 **squid** 软件包：

```
# yum install squid
```

2. 编辑 `/etc/squid/squid.conf` 文件：

- a. 要配置 **basic_ldap_auth** 帮助程序工具,请在 `/etc/squid/squid.conf` 顶部添加以下配置条目：

```
auth_param basic program /usr/lib64/squid/basic_ldap_auth -b
"cn=users,cn=accounts,dc=example,dc=com" -D
"uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com" -W
/etc/squid/ldap_password -f "(&(objectClass=person)(uid=%s))" -ZZ -H
ldap://ldap_server.example.com:389
```

下面描述了在上例中传递给 **basic_ldap_auth** 帮助程序的参数：

- **-b base_DN** 设置 LDAP 搜索基础。
- **-D proxy_service_user_DN** 设置帐户 Squid 的可分辨名称(DN)来搜索目录中验证用户。
- **-W path_to_password_file** 设置包含代理服务用户密码的文件路径。使用密码文件可防止在操作系统的进程列表中看到密码。
- **-f LDAP_filter** 指定 LDAP 搜索过滤器。squid 将 **%s** 变量替换为身份验证用户提供的用户名。
示例中的 **(&(objectClass=person)(uid=%s))** 过滤器定义了用户名必须与 **uid** 属性中设置的值匹配,目录条目包含 **person** 对象类。
- **-ZZ** 使用 **STARTTLS** 命令通过 LDAP 协议强制使用 TLS 加密连接。在以下情况下省略 **-ZZ**：
 - LDAP 服务器不支持加密的连接。
 - URL 中指定的端口使用 LDAPS 协议。
- **-H LDAP_URL** 参数指定协议、主机名或 IP 地址以及 LDAP 服务器的端口，格式为 URL。

- b. 添加以下 ACL 和规则来配置 Squid 只允许经过身份验证的用户使用代理：

```
acl ldap-auth proxy_auth REQUIRED
http_access allow ldap-auth
```



重要

在 **http_access deny** 所有规则前指定这些设置。

- c. 删除以下规则以禁用从 **localnet** ACL 中指定的 IP 范围中绕过代理身份验证：

```
http_access allow localnet
```

- d. 以下 ACL 存在于默认配置中，将 **443** 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议，请为每个端口添加 ACL：

```
acl SSL_ports port port_number
```

- e. 更新 **acl Safe_ports** 规则列表，以配置 Squid 可以建立连接的端口。例如：要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源,请在配置中保留以下 **acl Safe_ports** 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```

默认情况下,配置包含 **http_access deny !Safe_ports** 规则,该规则定义了对没有在 **Safe_ports ACLs** 中定义的端口的访问拒绝。

- f. 在 **cache_dir** 参数中配置缓存类型、缓存目录的路径、缓存大小以及其它缓存类型设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- squid 使用 **ufs** 缓存类型。
 - squid 将缓存存储在 **/var/spool/squid/** 目录中。
 - 缓存增长到 **10000** MB。
 - squid 在 **/var/spool/squid/** 目录中创建 **16** level-1 子目录。
 - squid 在每个级别-1 目录中创建 **256** 子目录。
- 如果您没有设置 **cache_dir** 指令，Squid 会将缓存保存在内存中。

3. 如果您在 **/var/spool/squid/** 参数中设置了与 **cache_dir** 不同的缓存目录：

- a. 创建缓存目录：

```
# mkdir -p path_to_cache_directory
```

- b. 配置缓存目录的权限：

```
# chown squid: squid path_to_cache_directory
```

- c. 如果您以 **enforcing** 模式运行 SELinux，请为缓存目录设置 **squid_cache_t** 上下文：

```
# semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
# restorecon -Rv path_to_cache_directory
```

如果您的系统中没有 **semanage** 工具，安装 **polycoreutils-python-utils** 软件包。

4. 将 LDAP 服务用户的密码存储在 **/etc/squid/ldap_password** 文件中，并为该文件设置适当的权限：

```
# echo "password" > /etc/squid/ldap_password
# chown root:squid /etc/squid/ldap_password
# chmod 640 /etc/squid/ldap_password
```

5. 在防火墙中打开 **3128** 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

6. 启用并启动 **squid** 服务：

```
# systemctl enable --now squid
```

验证步骤

要验证代理是否正常工作，使用 **curl** 工具下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" -x
  "user_name:password@proxy.example.com:3128"
```

如果 **curl** 没有显示任何错误，且 **index.html** 文件下载到当前目录中，则代理可以正常工作。

故障排除步骤

验证 **helper** 工具是否正常工作：

1. 使用您在 **auth_param** 参数中使用的相同设置手动启动 **helper** 工具：

```
# /usr/lib64/squid/basic_ldap_auth -b "cn=users,cn=accounts,dc=example,dc=com" -D
  "uid=proxy_user,cn=users,cn=accounts,dc=example,dc=com" -W
  /etc/squid/ldap_password -f "(&(objectClass=person)(uid=%s))" -ZZ -H
  ldap://ldap_server.example.com:389
```

2. 输入一个有效的用户名和密码，然后按 **Enter** 键：

```
user_name password
```

如果帮助程序返回 **OK**,身份验证成功。

7.3. 使用 KERBEROS 验证将 SQUID 设置为缓存代理

本节论述了 Squid 的基本配置作为缓存代理,该代理使用 Kerberos 向 Active Directory(AD)验证用户。此流程配置仅经过身份验证的用户可以使用代理。

先决条件

- 该流程假设 `/etc/squid/squid.conf` 文件由 **squid** 软件包提供。如果您在之前编辑了这个文件,请删除该文件并重新安装该软件包。
- 要安装 Squid 的服务器是 AD 域的成员。详情请查看 Red Hat Enterprise Linux 8 **Deploying different types of servers** 文档中的将 **Samba 设置为域成员**。

流程

1. 安装以下软件包：

```
yum install squid krb5-workstation
```

2. 以 AD 域管理员身份进行身份验证：

```
# kinit administrator@AD.EXAMPLE.COM
```

3. 为 Squid 创建 keytab, 并将其存储在 `/etc/squid/HTTP.keytab` 文件中：

```
# export KRB5_KTNAME=FILE:/etc/squid/HTTP.keytab
# net ads keytab CREATE -U administrator
```

4. 在 keytab 中添加 **HTTP** 服务主体：

```
# net ads keytab ADD HTTP -U administrator
```

5. 将 keytab 文件的拥有者设置为 **squid** 用户：

```
# chown squid /etc/squid/HTTP.keytab
```

6. (可选) 验证 keytab 文件是否包含代理服务器的完全限定域名(FQDN)的 **HTTP** 服务主体：

```
klist -k /etc/squid/HTTP.keytab
Keytab name: FILE:/etc/squid/HTTP.keytab
KVNO Principal
-----
...
2 HTTP/proxy.ad.example.com@AD.EXAMPLE.COM
...
```

7. 编辑 `/etc/squid/squid.conf` 文件：

- a. 要配置 **negotiate_kerberos_auth** 帮助程序工具, 请在 `/etc/squid/squid.conf` 顶部添加以下配置条目：

```
auth_param negotiate program /usr/lib64/squid/negotiate_kerberos_auth -k
/etc/squid/HTTP.keytab -s HTTP/proxy.ad.example.com@AD.EXAMPLE.COM
```

下面描述了在上例中传递给 **negotiate_kerberos_auth** 帮助程序的参数：

- **-k file** 设置到密钥标签文件的路径。请注意，squid 用户必须拥有这个文件的读取权限。
- **-s HTTP/host_name@kerberos_realm** 设置 Squid 使用的 Kerberos 主体。
另外，您可以通过将以下一个或多个参数传递给帮助程序来启用日志：
- **-i** 记录信息，如身份验证用户。
- **-d** 启用调试日志记录。
Squid 将帮助程序中的调试信息记录到 **/var/log/squid/cache.log** 文件。

b. 添加以下 ACL 和规则来配置 Squid 只允许经过身份验证的用户使用代理：

```
acl kerb-auth proxy_auth REQUIRED
http_access allow kerb-auth
```



重要

在 **http_access deny all** 规则前指定这些设置。

c. 删除以下规则以禁用从 **localnet** ACL 中指定的 IP 范围中绕过代理身份验证：

```
http_access allow localnet
```

d. 以下 ACL 存在于默认配置中，将 **443** 定义为使用 HTTPS 协议的端口：

```
acl SSL_ports port 443
```

如果用户也可以在其它端口上使用 HTTPS 协议，请为每个端口添加 ACL：

```
acl SSL_ports port port_number
```

e. 更新 **acl Safe_ports** 规则列表，以配置 Squid 可以建立连接的端口。例如：要配置使用代理的客户端只能访问端口 21(FTP)、80(HTTP)和 443(HTTPS)上的资源,请在配置中保留以下 **acl Safe_ports** 语句：

```
acl Safe_ports port 21
acl Safe_ports port 80
acl Safe_ports port 443
```

默认情况下,配置包含 **http_access deny !Safe_ports** 规则,用于定义对 **Safe_ports** ACL 中没有定义的端口的访问拒绝。

f. 在 **cache_dir** 参数中配置缓存类型、缓存目录的路径、缓存大小以及其它缓存类型设置：

```
cache_dir ufs /var/spool/squid 10000 16 256
```

使用这些设置：

- squid 使用 **ufs** 缓存类型。
- squid 将缓存存储在 **/var/spool/squid/** 目录中。
- 缓存增长到 **10000** MB。
- squid 在 **/var/spool/squid/** 目录中创建 **16** level-1 子目录。
- squid 在每个级别-1 目录中创建 **256** 子目录。
如果您没有设置 **cache_dir** 指令，Squid 会将缓存保存在内存中。

8. 如果您在 **/var/spool/squid/** 参数中设置了与 **cache_dir** 不同的缓存目录：

a. 创建缓存目录：

```
# mkdir -p path_to_cache_directory
```

b. 配置缓存目录的权限：

```
# chown squid:squid path_to_cache_directory
```

c. 如果您以 **enforcing** 模式运行 SELinux，请为缓存目录设置 **squid_cache_t** 上下文：

```
# semanage fcontext -a -t squid_cache_t "path_to_cache_directory(/.*)?"
# restorecon -Rv path_to_cache_directory
```

如果您的系统中没有 **semanage** 工具，安装 **polycoreutils-python-utils** 软件包。

9. 在防火墙中打开 **3128** 端口：

```
# firewall-cmd --permanent --add-port=3128/tcp
# firewall-cmd --reload
```

10. 启用并启动 **squid** 服务：

```
# systemctl enable --now squid
```

验证步骤

要验证代理是否正常工作，使用 **curl** 工具下载网页：

```
# curl -O -L "https://www.redhat.com/index.html" --proxy-negotiate -u : -x
"proxy.ad.example.com:3128"
```

如果 **curl** 没有显示任何错误，且 **index.html** 文件存在于当前目录中，则代理可以正常工作。

故障排除步骤

手动测试 Kerberos 身份验证：

1. 为 AD 帐户获取 Kerberos ticket：

```
# kinit user@AD.EXAMPLE.COM
```

2. 显示 ticket（可选）：

```
# klist
```

- 使用 **negotiate_kerberos_auth_test** 实用程序测试身份验证：

```
# /usr/lib64/squid/negotiate_kerberos_auth_test proxy.ad.example.com
```

如果帮助程序实用程序返回令牌,身份验证会成功：

```
Token: YIIftAYGKwYBBQUColIFqDC...
```

7.4. 在 SQUID 中配置域拒绝列表

通常,管理员想要阻止对特定域的访问。这部分论述了如何在 Squid 中配置域拒绝列表。

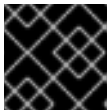
先决条件

- squid 被配置，用户可以使用代理。

流程

- 编辑 **/etc/squid/squid.conf** 文件并添加以下设置：

```
acl domain_deny_list dstdomain "/etc/squid/domain_deny_list.txt"
http_access deny all domain_deny_list
```

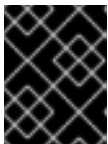


重要

在允许访问用户或客户端的第一个 **http_access allow** 语句前添加这些条目。

- 创建 **/etc/squid/domain_deny_list.txt** 文件并添加您要阻断的域。例如，要阻止对 **example.com** 的访问，包括子域和块 **example.net**，请添加：

```
.example.com
example.net
```



重要

如果您引用了 squid 配置中的 **/etc/squid/domain_deny_list.txt** 文件，这个文件不能为空。如果文件为空,Squid 无法启动。

- 重启 **squid** 服务：

```
# systemctl restart squid
```

7.5. 将 SQUID 服务配置为侦听特定端口或 IP 地址

默认情况下,Squid 代理服务侦听所有网络接口的 **3128** 端口。这部分论述了如何更改端口并配置 Squid 在特定 IP 地址中侦听。

先决条件

- 已安装 **squid** 软件包。

流程

1. 编辑 `/etc/squid/squid.conf` 文件：

- 要设置 Squid 服务侦听的端口,在 **http_port** 参数中设置端口号。例如,要将端口设置为 **8080**,请设置：

```
http_port 8080
```

- 要配置 Squid 服务侦听的 IP 地址,请在 **http_port** 参数中设置 IP 地址和端口号。例如：要配置 Squid 只侦听端口 **3128** 的 **192.0.2.1** IP 地址,请设置：

```
http_port 192.0.2.1:3128
```

在配置文件中添加多个 **http_port** 参数来配置 Squid 侦听多个端口和 IP 地址：

```
http_port 192.0.2.1:3128
http_port 192.0.2.1:8080
```

2. 如果您配置了 Squid 使用不同的端口作为默认端口 (**3128**)：

a. 在防火墙中打开端口：

```
# firewall-cmd --permanent --add-port=port_number/tcp
# firewall-cmd --reload
```

b. 如果您以 enforcing 模式运行 SELinux,请将端口分配给 **squid_port_t** 端口类型定义：

```
# semanage port -a -t squid_port_t -p tcp port_number
```

如果您的系统中没有 **semanage** 工具，安装 **polycoreutils-python-utils** 软件包。

3. 重启 **squid** 服务：

```
# systemctl restart squid
```

7.6. 其它资源

- 如需您可以在 `/etc/squid/squid.conf` 文件中设置的所有配置参数列表,请参阅 `usr/share/doc/squid-<version>/squid.conf.documented` 文件以及详细描述。

第 8 章 数据库服务器

8.1. 介绍

数据库服务器是拥有特定主内存的硬件设备,并安装了数据库(DB)应用程序。这个 DB 应用程序提供将缓存的数据从主内存中写入（通常比较小且昂贵）到 DB 文件（数据库）的方法。这些服务是为网络上的多个客户端提供的。在机器的主内存和存储空间允许的情况下，可以有多个 DB 服务器。

Red Hat Enterprise Linux 8 提供以下数据库应用程序：

- MariaDB 10.3
- MySQL 8.0
- PostgreSQL 10
- PostgreSQL 9.6
- PostgreSQL 12 - 从 RHEL 8.1.1 开始可用

8.2. 使用 MARIADB

8.2.1. MariaDB 入门

MariaDB 服务器是一个基于 MySQL 技术的开源快速可靠的数据库服务器。

MariaDB 是一个关系数据库,它将数据转换为结构化信息,并提供用于访问数据的 SQL 接口。它包括多个存储引擎和插件,以及地理位置(GIS)和 JavaScript Object Notation(JSON)功能。

这部分论述了如何在安装 **MariaDB** 中 [安装 MariaDB](#) 服务器以及如何从 Red Hat Enterprise Linux 7 默认版本 **MariaDB 5.5** 中迁移,到 Red Hat Enterprise Linux 8 默认版本 **MariaDB 10.3**,在 [Migrating to MariaDB 10.3](#) 中,以及如何备份 MariaDB 数据。执行数据备份是 MariaDB 迁移的先决条件之一。

8.2.2. 安装 MariaDB

要安装 **MariaDB**, 请按照以下步骤执行：

1. 使用特定流安装 **mariadb** 模块, 确保系统中有 MariaDB 服务器所需的所有软件包：

```
# yum module install mariadb:10.3/server
```

2. 启动 **mariadb** 服务：

```
# systemctl start mariadb.service
```

3. 在引导时启用 **mariadb** 服务：

```
# systemctl enable mariadb.service
```



注意

请注意，由于 RPM 软件包有冲突，**MariaDB** 和 **MySQL** 数据库服务器无法在 Red Hat Enterprise Linux 8.0 中并行安装。可以在 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 的 Red Hat Software Collections 中并行安装组件。在 Red Hat Enterprise Linux 8 中，可在容器中使用不同版本的数据库服务器。

8.2.2.1. 提高 MariaDB 安装安全性

要在安装 **MariaDB** 时提高安全性，请运行以下命令。

该命令会启动一个完全互动的脚本，它会提示进程中的每个步骤。

```
# mysql_secure_installation
```

这个脚本能够通过以下方式提高安全性：

- 为 root 帐户设置密码
- 删除匿名用户
- 不允许远程（本地主机以外的）root 登录

8.2.3. 配置 MariaDB

8.2.3.1. 为网络配置 MariaDB 服务器

要为网络配置 **MariaDB** 服务器，使用 `/etc/my.cnf.d/mariadb-server.cnf` 文件的 `[mysqld]` 部分，您可以在其中设置以下配置指令：

- **bind-address**
bind-address 是服务器要侦听的地址。

可能的选项有：主机名、IPv4 地址或 IPv6 地址。
- **skip-networking**
可能的值有：

0 - 侦听所有客户端

1 - 仅侦听本地客户端
- **port**
MariaDB 侦听 TCP/IP 连接的端口。

8.2.4. 备份 MariaDB 数据

从 **MariaDB** 数据库备份数据有两种主要方法：

- 逻辑备份
- 物理备份

逻辑备份 包含恢复数据的 SQL 语句。这类备份会以纯文本文件的形式导出信息和记录。

在物理备份中进行逻辑备份的主要优点是可移植性和灵活性。数据可以在其他硬件配置、MariaDB 版本或数据库管理系统(DBMS)中恢复,这些配置无法进行物理备份。

请注意,如果 **mariadb.service** 在运行时,可以执行逻辑备份。逻辑备份不包括日志和配置文件。

物理备份由保存内容的文件和目录副本组成。

与逻辑备份相比,物理备份具有以下优点：

- 输出更为紧凑。
- 备份的大小会较小。
- 备份和恢复速度更快。
- 备份包括日志和配置文件。

请注意,当 **mariadb.service** 没有运行或者数据库中的所有表锁定时,必须执行物理备份,以防止备份过程中发生更改。

您可以使用以下 **MariaDB** 备份方法之一从 **MariaDB** 数据库中备份数据：

- 使用 **mysqldump** 的逻辑备份
- 使用 **Mariabackup** 工具进行物理在线备份
- 文件系统备份
- 作为备份解决方案复制

8.2.4.1. 使用 **mysqldump** 执行逻辑备份

mysqldump 客户端是一个备份工具,可用于转储数据库或数据库集合,以便备份或传送到另一个数据库服务器。**mysqldump** 的输出通常由 SQL 语句组成,用于重新创建服务器表结构,并将其填充到数据中,或两者。另外,**mysqldump** 也可以以其他格式生成文件,包括 CSV 或其他限定文本格式以及 XML。

要执行 **mysqldump** 备份,您可以使用以下选项之一：

- 备份所选数据库
- 从一个数据库备份表子集
- 备份多个数据库
- 备份所有数据库

8.2.4.1.1. 使用 **mysqldump** 备份整个数据库

流程

- 要备份整个数据库,请运行：

```
# mysqldump [options] db_name > backup-file.sql
```

8.2.4.1.2. 使用 **mysqldump** 备份来自一个数据库的一组表

流程

- 要备份一个数据库的表子集, 在 **mysqldump** 命令末尾添加所选表列表 :

```
# mysqldump [options] db_name [tbl_name ...]
```

8.2.4.1.3. 使用 mysqldump 将转储文件重新加载到服务器中

流程

- 要将转储文件重新载入服务器, 请使用其中之一 :

```
# mysql db_name < backup-file.sql
```

```
# mysql -e "source /path-to-backup/backup-file.sql" db_name
```

8.2.4.1.4. 使用 mysqldump 在两个数据库之间复制数据

流程

- 要通过将数据从一个 **MariaDB** 服务器复制到另一个 MariaDB 服务器来填充数据库, 请运行 :

```
# mysqldump --opt db_name | mysql --host=remote_host -C db_name
```

8.2.4.1.5. 使用 mysqldump 转储多个数据库

流程

- 要同时转储多个数据库, 请运行 :

```
# mysqldump [options] --databases db_name1 [db_name2 ...] > my_databases.sql
```

8.2.4.1.6. 使用 mysqldump 转储所有数据库

流程

- 要转储所有数据库, 请运行 :

```
# mysqldump [options] --all-databases > all_databases.sql
```

8.2.4.1.7. 查看 mysqldump 选项

流程

- 要查看 mysqldump 支持的选项列表, 请运行 :

```
$ mysqldump --help
```

8.2.4.1.8. 其它资源

有关使用 `mysqldump` 的逻辑备份的详情，请查看 [MariaDB 文档](#)。

8.2.4.2. 使用 Mariabackup 工具执行物理在线备份

Mariabackup 是一个基于 Percona XtraBackup 技术的工具,它允许对 InnoDB、Aria 和 MyISAM 表执行物理在线备份。

Mariabackup 由 AppStream 存储库的 **mariadb-backup** 软件包提供,支持 MariaDB 服务器的完整备份功能,其中包括加密和压缩的数据。

先决条件

- **mariadb-backup** 软件包安装在系统中：

```
# yum install mariadb-backup
```

- **Mariabackup** 需要为运行备份的用户提供凭证。您可以在命令行中提供凭证（如所示），也可以在应用该步骤前根据配置文件提供。要使用配置文件设置凭证,首先创建文件（如 `/etc/my.cnf.d/mariabackup.cnf`），而不在新文件的 **[xtrabackup]** 或 **[mysqld]** 部分添加以下行：

```
[xtrabackup]
user=myuser
password=mypassword
```

重要

Mariabackup 不读取配置文件的 **[mariadb]** 部分中的选项。如果在 MariaDB 服务器中指定了非默认数据目录,您必须在配置文件的 **[xtrabackup]** 或 **[mysqld]** 部分指定这个目录,以便 **Mariabackup** 能够查找数据目录。

要指定这样的数据目录,请在 MariaDB 配置文件的 **[xtrabackup]** 或 **[mysqld]** 部分包含以下行：

```
datadir=/var/mycustomdatadir
```

注意

Mariabackup 用户必须具有 **RELOAD**、**LOCK TABLES** 和 **REPLICATION CLIENT** 权限才能用于备份。

要使用 **Mariabackup** 创建数据库备份，请使用以下步骤：

流程

- 运行以下命令：

```
$ mariabackup --backup --target-dir <backup_directory> --user <backup_user> --password <backup_passwd>
```

target-dir 选项定义备份文件的存储目录。如果要执行完整备份，目标目录必是空或者不存在。

user 和 **password** 选项允许您配置用户名和密码。如果您已在配置文件中配置了用户名和密码,请不要使用这些选项,如先决条件所述。

其它资源

有关使用 **Mariabackup** 执行备份的详情,请参考使用 [Mariabackup 的完整备份和恢复](#)。

8.2.4.3. 使用 Mariabackup 工具恢复数据

完成备份后,您可以使用以下选项之一使用 **mariabackup** 命令恢复备份中的数据:

- **--copy-back**
- **--move-back**

--copy-back 选项允许您保存原始备份文件。该 **--move-back** 选项将备份文件移到数据目录中,并删除原始备份文件。

先决条件

- 确保 **mariadb** 服务没有运行:

```
# systemctl stop mariadb.service
```

- 确保数据目录为空。

8.2.4.3.1. 在保留备份文件时使用 Mariabackup 恢复数据

要在保留原始备份文件的同时恢复数据,请使用以下步骤。

流程

1. 使用 **--copy-back** 选项运行 **mariabackup** 命令:

```
$ mariabackup --copy-back --target-dir=/var/mariadb/backup/
```

2. 修复文件权限。

当恢复数据库时, **Mariabackup** 会保留备份的文件和目录权限。但是, **Mariabackup** 以用户和组恢复数据库的身份将文件写入磁盘。因此,在恢复备份后,您可能需要调整数据目录的所有者,使其与 **MariaDB** 服务器的用户和组群匹配,通常是这两者的 **mysql**。

例如,要递归地将文件的所有权改为 **mysql** 用户和组:

```
# chown -R mysql:mysql /var/lib/mysql/
```

3. 启动 **mariadb** 服务:

```
# systemctl start mariadb.service
```

8.2.4.3.2. 在删除备份文件时使用 Mariabackup 恢复数据

要恢复数据,而不保存原始备份文件,请使用以下步骤。

流程

1. 使用 **--move-back** 选项运行 **mariabackup** 命令：

```
$ mariabackup --move-back --target-dir=/var/mariadb/backup/
```

2. 修复文件权限。

当恢复数据库时, **Mariabackup** 会保留备份的文件和目录权限。但是, **Mariabackup** 以用户和组恢复数据库的身份将文件写入磁盘。因此,在恢复备份后,您可能需要调整数据目录的所有者,使其与 **MariaDB** 服务器的用户和组群匹配,通常是这两者的 **mysql**。

例如, 要递归地将文件的所有权改为 **mysql** 用户和组：

```
# chown -R mysql:mysql /var/lib/mysql/
```

3. 启动 **mariadb** 服务：

```
# systemctl start mariadb.service
```

8.2.4.3.3. 其它资源

如需更多信息, 请参阅[使用 Mariabackup 的完整备份和恢复](#)。

8.2.4.4. 执行文件系统备份

要创建 **MariaDB** 数据文件的文件系统备份,切换到 **root** 用户,并将 **MariaDB** 数据目录的内容复制到备份位置。

要备份您当前的配置或日志文件,请使用以下步骤的可选步骤。

流程

1. 停止 **mariadb** 服务：

```
# systemctl stop mariadb.service
```

2. 将数据文件复制到所需位置：

```
# cp -r /var/lib/mysql /backup-location
```

3. (可选) 将配置文件复制到所需位置：

```
# cp -r /etc/my.cnf /etc/my.cnf.d /backup-location/configuration
```

4. (可选) 将日志文件复制到所需位置：

```
# cp /var/log/mariadb/* /backup-location/logs
```

5. 启动 **mariadb** 服务：

```
# systemctl start mariadb.service
```


8.2.4.5. 使用复制作为备份解决方案的介绍

复制（Replication）是 master 服务器的一个替代的备份解决方案。如果主（master）服务器复制到从（slave）服务器中，则可以在从属服务器上运行备份，而不会对主服务器进行任何影响。在关闭了从系统时，主系统仍然可正常运行，并从中备份数据。



警告

复制本身并不是一个足够的备份解决方案。复制可防止主服务器出现硬件故障时的影响，但它不能确保防止数据的丢失。建议您在使用复制时，同时使用其他备份解决方案。

其它资源

有关复制作为备份解决方案的更多信息,请参阅 [MariaDB 文档](#)。

8.2.5. 迁移到 MariaDB 10.3

Red Hat Enterprise Linux 7 包含 **MariaDB 5.5** 作为 MySQL 数据库系列中的服务器的默认实现。**MariaDB** 数据库服务器的更新版本是 Red Hat Enterprise Linux 6 和 Red Hat Enterprise Linux 7 的 Software Collections。Red Hat Enterprise Linux 8 提供 **MariaDB 10.3** 和 **MySQL 8.0**。

8.2.5.1. RHEL 7 和 RHEL 8 版本的 MariaDB 之间的显著区别

MariaDB 5.5 和 **MariaDB 10.3** 间最重要的变更如下：

- **MariaDB Galera 集群**（一个同步的多 master 集群）是 **MariaDB** 的标准部分,从 10.1 开始。
- 默认情况下不再启用 ARCHIVE 存储引擎,插件需要专门启用。
- 默认情况下不再启用 BLACKHOLE 存储引擎,插件需要专门启用。
- InnoDB 用作默认存储引擎,而不是 XtraDB,它在 **MariaDB 10.1** 及更早的版本中使用。如需了解更多详细信息,请参阅 [为什么 MariaDB 10.2 使用 InnoDB 而不是 XtraDB?](#)。
- 新的 **mariadb-connector-c** 软件包为 MySQL 和 MariaDB 提供了一个通用的客户端程序库。这个程序库可用于 **MySQL** 和 **MariaDB** 数据库服务器的任何版本。因此，用户可以将应用程序的一个构建连接到 Red Hat Enterprise Linux 8 发布的任何 MySQL 和 **MariaDB** 服务器。

要从 **MariaDB 5.5** 迁移到 **MariaDB 10.3**，您需要执行多个配置更改，如 [第 8.2.5.2 节“配置更改”](#) 所述。

8.2.5.2. 配置更改

从 **MariaDB 5.5** 升级到 **MariaDB 10.3** 的迁移路径是首先升级到 **MariaDB 10.0**，然后再每次升级一个版本。

每次升级一个版本的主要优点是，可以更好地适应数据库，包括数据和更改配置。这样，升级过程可以结束后获得的主版本与 RHEL 8 中的相同（**MariaDB 10.3**），这可显著减少配置更改或其他问题。

有关从 **MariaDB 5.5** 迁移到 **MariaDB 10.0** 时配置变化的更多信息，请参阅 Red Hat Software Collections 文档中的 [Migrating to MariaDB 10.0](#) 部分。

以下介绍了迁移到 MariaDB 连续版本以及所需的配置更改：

- 在 [Red Hat Software Collections 文档](#) 中迁移到 MariaDB 10.1。
- 在 [Red Hat Software Collections 文档](#) 中迁移到 MariaDB 10.2。
- 在 [Red Hat Software Collections 文档](#) 中迁移到 MariaDB 10.3。



注意

也有可能直接从 MariaDB 5.5 迁移到 MariaDB 10.3,但您必须执行以上迁移文档中描述的不同所需的所有配置更改。

8.2.5.3. 使用 `mysql_upgrade` 工具进行原位升级

要将数据库文件迁移到 Red Hat Enterprise Linux 8,Red Hat Enterprise Linux 7 的 MariaDB 用户需要使用 `mysql_upgrade` 工具执行原位升级。`mysql_upgrade` 工具由 `mariadb-server-utils` 子软件包提供,该子软件包作为 `mariadb-server` 软件包的依赖项安装。

要执行原位升级,您需要将二进制数据文件复制到 Red Hat Enterprise Linux 8 系统的 `/var/lib/mysql/` 数据目录中,并使用 `mysql_upgrade` 工具。

您可以使用此方法迁移以下数据：

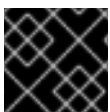
- Red Hat Enterprise Linux 7 的 MariaDB 5.5
- Red Hat Software Collections 的以下版本：
 - MariaDB 5.5（不再支持）
 - MariaDB 10.0（不再支持）
 - MariaDB 10.1（不再支持）
 - MariaDB 10.2

请注意,建议连续升级到 MariaDB 10.2。请参阅 [Red Hat Software Collections 发行注记](#) 中的相应的迁移章节。



注意

如果您要从 MariaDB 的 Red Hat Enterprise Linux 7 版本升级,则源数据保存在 `/var/lib/mysql/` 目录中。如果是 MariaDB 的 Red Hat Software Collections 版本,则源数据目录为 `/var/opt/rh/<collection_name>/lib/mysql/`（`mariadb55` 除外,使用 `/opt/rh/mariadb55/root/var/lib/mysql/` 数据目录）。



重要

在进行升级前,备份 MariaDB 数据库中的所有数据。

要执行原位升级,请切换到 `root` 用户,并使用以下步骤：

1. 确定在 Red Hat Enterprise Linux 8 系统中安装了 `mariadb-server` 软件包：

```
# yum install mariadb-server
```

2. 确定在复制数据时 mariadb 守护进程不在源和目标系统中运行：

```
# systemctl stop mariadb.service
```

3. 将源位置的数据复制到 Red Hat Enterprise Linux 8 目标系统的 `/var/lib/mysql/` 目录中。
4. 为目标系统中复制的文件设置适当的权限和 SELinux 上下文：

```
# restorecon -vr /var/lib/mysql
```

5. 在目标系统中启动 MariaDB 服务器：

```
# systemctl start mariadb.service
```

6. 运行 `mysql_upgrade` 命令检查并修复内部表：

```
# mysql_upgrade
```

7. 升级完成后,确保 `/etc/my.cnf.d/` 目录中的所有配置文件仅包含 MariaDB 10.3 的有效选项。



重要

与原位升级相关的某些风险和已知问题。例如,有些查询可能无法正常工作,或者它们会以与升级前不同的顺序运行。有关这些风险和问题的更多信息,以及原位升级的一般信息,请参阅 [MariaDB 10.3 发行注记](#)。

8.2.6. 使用 Galera 复制 MariaDB

本节论述了如何使用 Galera 解决方案复制 MariaDB 数据库。

8.2.6.1. MariaDB Galera 集群介绍

Galera 复制基于创建一个同步的多 master **MariaDB Galera 集群**,由多个 MariaDB 服务器组成。

Galera 复制和 MariaDB 数据库之间的接口由写入集合复制 API(**wsrep API**)定义。

MariaDB Galera 集群 的主要特性是：

- 同步复制
- 主动-主动多 master 拓扑
- 对任何集群节点的读和写
- 自动成员资格控制,故障节点从集群中丢弃
- 自动节点加入
- 行一级的并行复制
- 直接客户端连接（用户可以登录到集群节点,并在复制运行时直接与节点配合。）

同步复制表示,服务器在提交时通过将与该事务关联的写入集广播到集群中的每个节点来复制交易。客户端（用户应用程序）直接连接到数据库管理系统(DBMS),并体验与原生 MariaDB 类似的行为。

同步复制保证集群中一个节点上的更改会同时在集群中的其他节点上发生。

因此，与异步复制相比，同步复制具有以下优势：

- 在特定集群节点间传播更改没有延迟
- 所有集群节点始终一致
- 如果其中一个集群节点崩溃，则不会丢失最新的更改
- 所有集群节点上的事务都会并行执行
- 整个集群中的因数

其它资源

如需更多信息,请参阅上游文档：

- [关于 Galera 复制](#)
- [什么是 MariaDB Galera 集群](#)
- [MariaDB Galera 集群入门](#)

8.2.6.2. 构建 MariaDB Galera 集群的组件

为了可以构建 **MariaDB Galera 集群**,必须在您的系统中安装以下软件包：

- **mariadb-server-galera**
- **mariadb-server**
- **galera**

mariadb-server-galera 软件包包含 **MariaDB Galera 集群**的支持文件和脚本。

MariaDB 上游补丁 **mariadb-server** 软件包使其包含写入集合复制 API(**wsrep API**)。此 API 提供了 Galera 复制和 MariaDB 间的接口。

MariaDB 上游社区还对 **galera** 软件包进行补丁,以添加对 **MariaDB** 的全面支持。**galera** 软件包包含 **Galera Replication 库**和 **Galera Arbitrator 工具**。**Galera Replication 程序库**提供整个复制功能。**Galera Arbitrator** 可以用作在分割场景中参与投票的集群成员。但是, **Galera Arbitrator** 无法参与实际的复制。

其它资源

如需更多信息，请参阅上游文档：

- [Galera 复制程序](#)
- [Galera Arbitrator](#)
- [MySQL-wsrep 项目](#)

8.2.6.3. 部署 MariaDB Galera 集群

先决条件

- 必须在系统中安装构建 MariaDB Galera 集群所需的所有软件。要确保这一点，请安装 **mariadb:10.3** 模块的 **galera** 配置集：

```
# yum module install mariadb:10.3/galera
```

因此，会安装以下软件包：

- **mariadb-server-galera**
- **mariadb-server**
- **galera**
mariadb-server-galera 软件包拉取 **mariadb-server** 和 **galera** 软件包作为其依赖项。

有关构建 MariaDB Galera 集群的组件的更多信息，请参阅 [第 8.2.6.2 节“构建 MariaDB Galera 集群的组件”](#)。

- 在第一次将系统添加到集群中之前，必须更新 MariaDB 服务器复制配置。默认配置在 **/etc/my.cnf.d/galera.cnf** 文件中提供。

在部署 MariaDB Galera 集群前，将所有节点上的 **/etc/my.cnf.d/galera.cnf** 文件中的 **wsrep_cluster_address** 选项设置为以下字符串开头：

```
gcomm://
```

对于初始节点，可以将 **wsrep_cluster_address** 设置为空列表：

```
wsrep_cluster_address="gcomm://"
```

对于所有其他节点，将 **wsrep_cluster_address** 设置为包括已经属于正在运行的集群一部分的任何节点的地址。例如：

```
wsrep_cluster_address="gcomm://10.0.0.10"
```

有关如何设置 Galera 集群地址的更多信息，请参阅 [Galera Cluster Address](#)。

流程

1. 通过在该节点上运行以下 wrapper 来引导新集群的第一个节点：

```
$ galera_new_cluster
```

这个打包程序确保 MariaDB 服务器守护进程(**mysqld**)使用 **--wsrep-new-cluster** 选项运行。这个选项提供没有现有集群可以连接的信息。因此，节点会创建一个新的 UUID 来识别新集群。



注意

mariadb 服务支持使用 **systemd** 方法与多个 **MariaDB** 服务器进程进行交互。因此，如果有多个运行的 **MariaDB** 服务器，您可以通过将实例名称指定为后缀来引导特定实例：

```
$ galera_new_cluster mariadb@node1
```

2. 在每个节点上运行以下命令将其他节点连接到集群：

```
# systemctl start mariadb
```

因此,节点连接到集群,并与集群状态同步。

其它资源

如需更多信息, 请参阅[开始使用 MariaDB Galera 集群](#)。

8.2.6.4. 在 MariaDB Galera 集群中添加新节点

要在 MariaDB Galera 集群中添加新节点, 请使用以下步骤。

请注意, 您也可以使用此流程重新连接已存在的节点。

流程

- 在特定节点中,在 `/etc/my.cnf.d/galera.cnf` 配置文件的 `[mariadb]` 部分的 `wsrep_cluster_address` 选项中为一个或多个现有集群成员提供地址：

```
[mariadb]
wsrep_cluster_address="gcomm://192.168.0.1"
```

当一个新节点连接到其中一个现有集群节点时,它可以查看集群中的所有节点。

但是,最好在 `wsrep_cluster_address` 中列出集群的所有节点。

因此,任何节点都可以通过连接到任何其他集群节点来加入集群,即使一个或多个集群节点停机也是如此。当所有成员都同意成员资格时,集群的状态将会改变。如果新节点的状态与集群的状态不同,它会请求 Incremental State transfer (IST) 或 State Snapshot 传输 (SST) 使其与其他节点保持一致。

其它资源

- 如需更多信息, 请参阅[开始使用 MariaDB Galera 集群](#)。
- 有关 State snapshot 传输 (SST) 的详细信息, 参见[状态快照传输简介](#)。

8.2.6.5. 重启 MariaDB Galera 集群

如果同时关闭所有节点,则终止集群,且正在运行的集群也不再存在。但是,集群的数据仍然存在。

要重启集群,请引导第一个节点,如 [第 8.2.6.3 节 “部署 MariaDB Galera 集群”](#) 所述。



警告

如果没有引导集群, 且第一个节点上的 `mysqld` 使用 `systemctl start mariadb` 命令启动, 则该节点会尝试连接到 `/etc/my.cnf.d/galera.cnf` 文件中 `wsrep_cluster_address` 选项中列出的至少一个节点。如果没有节点当前运行,重启会失败。

其它资源

如需更多信息, 请参阅[开始使用 MariaDB Galera 集群](#)。

8.3. 使用 POSTGRESQL

8.3.1. PostgreSQL 入门

PostgreSQL 服务器是一个开源强大的、高度扩展的数据库服务器,它基于 SQL 语言。它提供了一个对象相关的数据库系统,它可用于管理广泛数据集和大量并发用户。因此, PostgreSQL 服务器可用于管理大量数据。

PostgreSQL 服务器包含可用于确保数据完整性、构建容错环境或构建应用程序的功能。它允许使用用户自己的数据类型、自定义功能或代码从不同的编程语言扩展数据库,而无需重新编译数据库。

本节论述了如何安装 PostgreSQL ([安装 PostgreSQL](#)), 以及如何在迁移到 PostgreSQL ([RHEL 8 版本中迁移到不同版本的 PostgreSQL](#))。迁移的一个先决条件是执行数据备份。

8.3.2. 安装 PostgreSQL

在 RHEL 8 中, PostgreSQL 服务器在多个版本中可用,各自由单独的流提供:

- PostgreSQL 10 - 默认流
- PostgreSQL 9.6
- PostgreSQL 12 - 从 RHEL 8.1.1 开始可用



注意

根据设计,无法同时安装同一模块的多个版本(stream)。因此,您只需要从 **postgresql** 模块中选择一个可用流。组件的并行安装可以在 Red Hat Software Collections for RHEL 7 和 RHEL 6 中进行。在 RHEL 8 中,可在容器中使用不同版本的数据库服务器。

安装 PostgreSQL:

1. 启用您要安装的流 (版本):

```
# yum module enable postgresql:stream
```

使用所选 PostgreSQL 服务器版本替换 *stream*。

如果要使用默认流,可以省略这一步,该流提供 PostgreSQL 10。

2. 确保 AppStream 存储库中可用的 **postgresql-server** 软件包已安装在所需服务器上:

```
# yum install postgresql-server
```

3. 初始化数据目录

```
postgresql-setup --initdb
```

4. 启动 **postgresql** 服务:

■


```
# systemctl start postgresql.service
```

5. 在引导时启用 **postgresql** 服务：

```
# systemctl enable postgresql.service
```

有关使用模块流的详情,请参考 [安装、管理和删除用户空间组件](#)。



重要

如果要从 RHEL 8 中的较早 **postgresql** 流升级, 请按照[切换到更新的流](#)和 [第 8.3.5 节“迁移到 PostgreSQL 的 RHEL 8 版本”](#)中所述的步骤进行操作。

8.3.3. 配置 PostgreSQL

要更改 PostgreSQL 配置,使用 `/var/lib/pgsql/data/postgresql.conf` 文件。之后,重启 **postgresql** 服务以使更改生效：

```
systemctl restart postgresql.service
```

除了 `/var/lib/pgsql/data/postgresql.conf` 外,其它用于更改 PostgreSQL 配置的文件存在：

- **postgresql.auto.conf**
- **pg_ident.conf**
- **pg_hba.conf**

postgresql.auto.conf 文件包含类似于 `/var/lib/pgsql/data/postgresql.conf` 的基本 PostgreSQL 设置。但是这个文件由服务器控制。它由 **ALTER SYSTEM** 查询编辑,无法手动编辑。

pg_ident.conf 文件用于将外部验证机制的用户身份映射到 postgresql 用户身份。

pg_hba.conf 文件用于为 PostgreSQL 数据库配置详细的用户权限。

8.3.3.1. 初始化数据库集群

在 PostgreSQL 数据库中,所有数据都存储一个目录,称为数据库集群。您可以选择存储数据的位置,但红帽建议将数据存储在默认的 `/var/lib/pgsql/data` 目录中。

要初始化这个数据目录,请运行：

```
postgresql-setup --initdb
```

8.3.4. 备份 PostgreSQL 数据

要备份 PostgreSQL 数据,请使用以下方法之一：

- SQL 转储
- 文件系统级别备份
- 持续归档

8.3.4.1. 使用 SQL 转储备份 PostgreSQL 数据

8.3.4.1.1. 执行 SQL 转储

SQL 转储方法基于使用 SQL 命令生成文件。当此文件上传到数据库服务器时,它会使用与转储时相同的状态重新创建数据库。**pg_dump** 程序(一个 SQL 客户端应用程序)可保证实现 SQL 转储。**pg_dump** 命令的基本用法使得命令将其结果写入标准输出中:

```
pg_dump dbname > dumpfile
```

得到的 SQL 文件可以采用文本格式或其他不同的格式,它们允许并行性并对对象恢复进行更详细的控制。

您可以在任何可访问数据库的远程主机中执行 SQL 转储。**pg_dump** 实用程序不使用特殊权限操作,但必须具有对所有要备份的表的读取访问权限。要备份整个数据库,必须以数据库超级用户身份运行。

要指定 **pg_dump** 会联系哪个数据库服务器, 请使用以下命令行选项:

- 使用 **-h** 选项定义主机。
默认主机是本地主机, 也可以是 **PGHOST** 环境变量指定的主机。
- 定义端口的 **-p** 选项。
默认端口由 **PGPORT** 环境变量或编译默认端口表示。

注意

请注意, **pg_dump** 只转储单个数据库。它不会转储角色或表空间的信息,因为这些信息是全集群范围的。

要备份给定集群中的每个数据库,并保留集群范围的数据,如角色和表空间定义,请使用 **pg_dumpall** 命令:

```
pg_dumpall > dumpfile
```

8.3.4.1.2. 从 SQL 转储中恢复数据库

从 SQL 转储恢复数据库:

1. 创建新数据库(名称):

```
createdb dbname
```

2. 请确定所有拥有对象或被授予转储数据库中对象的权限的用户已经存在。
如果这些用户不存在,则恢复无法使用原始所有权和权限重新创建对象。

3. 运行 **psql** 工具来恢复 **pg_dump** 程序创建的文本文件转储:

```
psql dbname < dumpfile
```

其中 **dumpfile** 是 **pg_dump** 命令的输出。

如果要恢复非文本文件转储,使用 **pg_restore** 实用程序:

```
pg_restore non-plain-text-file
```

8.3.4.1.2.1. 在另一个服务器中恢复数据库

可以从一个服务器直接将数据库转储到另一个服务器,因为 **pg_dump** 和 **psql** 可以写入并从 pipes 读取。

要从一个服务器到另一个服务器转储数据库, 请运行 :

```
pg_dump -h host1 dbname | psql -h host2 dbname
```

8.3.4.1.2.2. 在恢复过程中处理 SQL 错误

默认情况下, 如果 SQL 错误发生, **psql** 会继续执行。因此, 数据库只会被部分恢复。

如果要更改此默认行为, 请使用以下方法之一 :

- 设置 **ON_ERROR_STOP** 变量以确保在发生 SQL 错误时, **psql** 会以退出状态 3 退出 :

```
psql --set ON_ERROR_STOP=on dbname < dumpfile
```

- 指定将整个转储恢复为一个事务,以便通过以下选项之一使用 **psql** 来完全完成或取消恢复 :

```
psql -1
```

或者

```
psql --single-transaction
```

请注意, 在使用这个方法时, 即使一个小的错误也可以取消已经运行了很长时间的恢复操作。

8.3.4.1.3. SQL 转储的优点和缺陷

与其它 PostgreSQL 备份方法相比, SQL 转储具有以下优点 :

- SQL 转储是唯一的、不针对特定服务器版本的 PostgreSQL 备份方法。**pg_dump** 工具的输出可以重新载入到以后的 PostgreSQL 版本,而文件系统级别备份或持续存档是不可能的。
- SQL 转储是将数据库传送到不同机器架构（比如从 32 位到 64 位服务器）的唯一方法。
- SQL 转储提供内部一致的转储。转储代表 **pg_dump** 开始运行时数据库的快照。
- **pg_dump** 程序不会阻止数据库中的其他操作。

SQL 转储的缺陷在于,与文件系统级别备份相比,它需要更长的时间。

8.3.4.1.4. 其它资源

有关 SQL 转储的更多信息, 请参阅 [PostgreSQL 文档](#)。

8.3.4.2. 使用文件系统级别备份来备份 PostgreSQL 数据

8.3.4.2.1. 执行文件系统级别备份

要执行文件系统级别备份,您需要复制 PostgreSQL 用来将数据存储到数据库中的文件到另一个位置 :

1. 选择数据库集群的位置并初始化该集群, 如 [第 8.3.3.1 节 “初始化数据库集群”](#) 所述。

2. 停止 postgresql 服务：

```
# systemctl stop postgresql.service
```

3. 使用任意方法进行文件系统备份，例如：

```
tar -cf backup.tar /var/lib/pgsql/data
```

4. 启动 postgresql 服务：

```
# systemctl start postgresql.service
```

8.3.4.2.2. 文件系统级别备份的优点和缺陷

文件系统级别备份与其他 PostgreSQL 备份方法相比有以下优点：

- 文件系统级别备份通常比 SQL 转储更快。

与其它 PostgreSQL 备份方法相比，文件系统级别备份有以下缺陷：

- 备份是特定于构架的，只特定于 Red Hat Enterprise Linux 7。它只能用于在升级不成功时返回到 Red Hat Enterprise Linux 7 的备份方法，但它不能与 PostgreSQL 10.0 一起使用。
- 数据库服务器必须在数据备份前和数据恢复前关闭。
- 无法备份和恢复某些独立文件或表。文件系统备份只能用于完整备份和恢复整个数据库集群。

8.3.4.2.3. 文件系统级别备份的替代方法

文件系统备份的替代方法包括：

- 数据目录的一致性快照
- rsync 工具

8.3.4.2.4. 其它资源

有关文件系统级别备份的详情，请参考 [PostgreSQL 文档](#)。

8.3.4.3. 通过持续存档来备份 PostgreSQL 数据

8.3.4.3.1. 持续归档介绍

PostgreSQL 将每次对数据库数据文件所做的更改记录到日志(WAL)文件中,该文件可在集群数据目录的 **pg_wal/** 子目录中找到。此日志主要用于崩溃恢复。崩溃后,最后一个检查点后的日志条目可以用来恢复数据库以保持一致性。

持续存档方法（也称为 **online backup**）将 WAL 文件与文件系统级别备份合并。如果需要数据库恢复,您可以从文件系统备份中恢复数据库,然后从备份的 WAL 文件中重新显示日志,使系统进入当前状态。

对于这个备份方法,您需要一个连续的已归档 WAL 文件序列,它们至少可回退到备份的开始时间。

如果要开始使用持续归档备份方法,请确保在进行第一次基础备份前设置并测试您的归档 WAL 文件的步骤。



注意

您不能使用 `pg_dump` 和 `pg_dumpall` 转储作为持续归档备份解决方案的一部分。这些转储生成逻辑备份，而不是文件系统级别备份。因此，它们不包含 WAL replay 使用的足够信息。

8.3.4.3.2. 执行持续存档备份

要使用持续存档方法执行数据库备份和恢复，请按照以下步骤操作：

1. [第 8.3.4.3.2.1 节 “进行基础备份”](#)
2. [第 8.3.4.3.2.2 节 “使用持续归档备份来恢复数据库”](#)

8.3.4.3.2.1. 进行基础备份

要执行基础备份，请使用 `pg_basebackup` 工具，该工具可以以单个文件或 `tar` 归档的形式创建基础备份。

要使用基础备份，您需要保留在文件系统备份过程中和之后生成的所有 WAL 片段文件。基础备份过程会创建一个备份历史文件，它保存在 WAL 归档区域，并命名为第一个文件系统备份所需的 WAL 片段文件。当您安全地归档了文件系统备份以及备份过程中使用的 WAL 片段文件（在备份中指定）时，您可以删除所有归档的 WAL 片段，其名称更少，因为不再需要恢复文件系统备份。但请考虑保留一些备份集合以确保您可以恢复数据。

备份历史记录文件是一个小的文本文件，其中包含您授予的 `pg_basebackup` 标签字符串、开始和结束时间，以及备份的 WAL 片段。如果您使用标签字符串来识别关联的转储文件，则存档的历史文件足以告诉您要恢复哪个转储文件。

使用连续存档方法，您需要保留所有归档的 WAL 文件到您的最后一个基础备份中。因此，基础备份的理想频率取决于：

- 归档 WAL 文件的存储卷。
- 需要恢复时数据恢复的最可能持续时间。
当从上次备份开始较长的时间时，系统会重新显示更多 WAL 片段，因此恢复需要较长时间。

有关进行基础备份的更多信息，请参阅 [PostgreSQL 文档](#)。

8.3.4.3.2.2. 使用持续归档备份来恢复数据库

使用持续备份恢复数据库：

1. 停止服务器：

```
# systemctl stop postgresql.service
```

2. 将必要的数据库复制到临时位置。
最好复制整个集群数据目录和任何表空间。请注意，这需要足够的可用空间才能保存现有数据库的两个副本。

如果您没有足够的空间，请保存集群的 `pg_wal` 目录的内容，该目录可包含系统停机前没有归档的日志。

3. 删除集群数据目录下的所有现有文件和子目录，并在您要使用的任何表空间的根目录下删除。
4. 从您的文件系统备份中恢复数据库文件。

请确定：

- 文件以正确的所有权恢复（数据库系统用户，而不是 **root**）
 - 文件以正确权限恢复
 - **pg_tblspc/** 子目录中的符号链接已被正确恢复
5. 删除 **pg_wal/** 子目录中的任何文件
这些文件源自文件系统备份,因此过时。如果您没有归档 **pg_wal/**,使用正确权限重新创建它。
 6. 如果存在这些文件,将在第 2 步中保存的未架构的 WAL 片段文件复制到 **pg_wal/**。
 7. 在集群数据目录中创建 **recovery.conf** 恢复命令文件。
 8. 启动服务器：

```
# systemctl start postgresql.service
```

服务器将进入恢复模式,并通过它需要的归档 WAL 文件读取。

如果恢复因为外部错误而终止,则服务器只能重启,并将继续恢复。恢复过程完成后,服务器会将 **recovery.conf** 重命名为 **recovery.done**,以防止在服务器启动普通数据库操作时意外重新进入恢复模式。

9. 检查数据库的内容,以确保数据库已恢复为所需状态。
如果数据库没有恢复到所需状态,请返回至第 1 步。如果数据库恢复为所需状态,允许用户通过将 **pg_hba.conf** 文件恢复到正常状态进行连接。

有关使用持续备份恢复的更多信息，请参阅 [PostgreSQL 文档](#)。

8.3.4.3.3. 持续归档的优点和缺陷

与其它 PostgreSQL 备份方法相比，持续归档具有以下优势：

- 使用持续备份方法时,可以使用不完全一致的文件系统备份,因为备份中的任何内部不一致都由日志重新显示来解决。不需要文件系统快照; **tar** 或类似的归档工具就足够了。
- 继续归档 WAL 文件可以实现持续备份,因为日志回放的 WAL 文件序列可能会无限期地长。这对大型数据库尤其重要。
- 持续备份支持点恢复。不需要将 WAL 条目重新显示到结尾。replay 可以在任何时间停止,并且自进行基础备份后可随时将数据库恢复到其状态。
- 如果 WAL 文件系列不断可供使用同一基础备份文件载入的另一台机器使用,则可以随时恢复使用当前数据库副本的其他机器。

与其它 PostgreSQL 备份方法相比,持续归档具有以下缺陷：

- 持续备份方法只支持恢复整个数据库集群，而不是子集。
- 持续备份需要广泛的归档存储。

8.3.4.3.4. 其它资源

有关持续存档方法的更多信息,请参阅 [PostgreSQL 文档](#)。

8.3.5. 迁移到 PostgreSQL 的 RHEL 8 版本

Red Hat Enterprise Linux 7 包含 **PostgreSQL 9.2** 作为 **PostgreSQL** 服务器的默认版本。另外，一些 **PostgreSQL** 版本也由 RHEL 7 和 RHEL 6 的 Software Collections 提供。

Red Hat Enterprise Linux 8 提供 **PostgreSQL 10**（默认 **postgresql** 流）、**PostgreSQL 9.6** 和 **PostgreSQL 12**。

Red Hat Enterprise Linux 上的 **PostgreSQL** 用户可为数据库文件使用两个迁移路径：

- [使用 pg_upgrade 工具快速升级](#)
- [转储和恢复升级](#)

最好使用快速升级方法，它比转储和恢复过程更快。

然而,在某些情况下,快速升级无法正常工作,您只能使用转储和恢复过程。这种情况包括：

- 跨架构升级
- 使用 **plpython** 或 **plpython2** 扩展的系统。请注意,RHEL 8 AppStream 软件仓库只包括 **postgresql-plpython3** 软件包,而没有包括 **postgresql-plpython2** 软件包。
- 从 **PostgreSQL** 的 Red Hat Software Collections 版本迁移不支持快速升级。

迁移到更新版本的 **PostgreSQL** 的先决条件是备份所有 **PostgreSQL** 数据库。

转储数据库并执行 SQL 文件备份是转储和恢复过程的必要部分。但是，如果执行快速升级，也建议您执行此操作。

在迁移到更新的 **PostgreSQL** 版本前,请参阅您要迁移的 **PostgreSQL** 版本的 [上游兼容性备注](#),以及您要迁移的版本和目标版本之间跳过的 **PostgreSQL** 版本。

8.3.5.1. 使用 pg_upgrade 工具快速升级

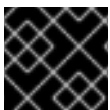
在快速升级过程中,您需要将二进制数据文件复制到 **/var/lib/pgsql/data/** 目录中,并使用 **pg_upgrade** 工具。

您可以使用此方法迁移数据：

- 从 RHEL 7 系统的 **PostgreSQL 9.2** 迁移到 RHEL 8 的 **PostgreSQL 10**
- 从 RHEL 8 的 **PostgreSQL 10** 迁移到 RHEL 8 的 **PostgreSQL 12**

如果要从 RHEL 8 中的较早 **postgresql** 流升级，请按照[切换到更新的流](#)中介绍的步骤进行，然后迁移 **PostgreSQL** 数据。

要在 RHEL 中 **PostgreSQL** 版本的其他组合间迁移,以及从 **PostgreSQL** 的 Red Hat Software Collections 版本迁移到 RHEL,请使用 [Dump](#) 和 [恢复升级](#)。



重要

在进行升级前,备份 **PostgreSQL** 数据库中的所有数据。

默认情况下，所有数据都存储在 RHEL 7 和 RHEL 8 系统上的 **/var/lib/pgsql/data/** 目录中。

以下介绍了从 RHEL 7 系统的 **PostgreSQL 9.2** 迁移到 RHEL 8 的 **PostgreSQL**。

执行快速升级：

1. 在 RHEL 8 系统中,启用您要迁移的流（版本）：

```
# yum module enable postgresql:stream
```

使用所选 **PostgreSQL** 服务器版本替换 *stream*。

如果要使用默认流,可以省略这一步,该流提供 **PostgreSQL 10**。

2. 在 RHEL 8 系统中,安装 **postgresql-server** 和 **postgresql-upgrade** 软件包：

```
# yum install postgresql-server postgresql-upgrade
```

另外,如果您在 RHEL 7 中使用任何 **PostgreSQL** 服务器模块,请在两个版本中在 RHEL 8 系统中安装它们,包括针对 **PostgreSQL 9.2**（安装为 **postgresql-upgrade** 软件包）和 **PostgreSQL** 的目标版本（安装为 **postgresql-server** 软件包）。如果您需要编译第三方 **PostgreSQL** 服务器模块,请根据 **postgresql-devel** 和 **postgresql-upgrade-devel** 软件包构建它。

3. 检查以下项：

- **基本配置**：在 RHEL 8 系统中,检查您的服务器是否使用默认 **/var/lib/pgsql/data** 目录,且数据库已被正确初始化并启用。另外,数据文件必须存储在与 **/usr/lib/systemd/system/postgresql.service** 文件中提到的同一路径中。
- **PostgreSQL 服务器**：您的系统可以运行多个 **PostgreSQL** 服务器。请确定所有这些服务器的数据目录都是独立处理的。
- **PostgreSQL 服务器模块**：确保您在 RHEL 7 中使用的 **PostgreSQL** 服务器模块也安装在 RHEL 8 系统中。请注意,插件安装在 **/usr/lib64/pgsql/** 目录中（或者在 32 位系统的 **/usr/lib/pgsql/** 目录中）。

4. 确定 **postgresql** 服务在复制数据时不在源和目标系统中运行。

```
# systemctl stop postgresql.service
```

5. 将源位置中的数据库文件复制到 RHEL 8 系统的 **/var/lib/pgsql/data/** 目录中。

6. 以 **PostgreSQL** 用户身份运行以下命令来执行升级过程：

```
$ /bin/postgresql-setup --upgrade
```

这会在后台启动 **pg_upgrade** 进程。

如果出现故障, **postgresql-setup** 会提供一个说明性错误信息。

7. 将之前的配置从 **/var/lib/pgsql/data-old** 复制到新集群。
请注意,快速升级不会重复使用较新的数据堆栈中的以前的配置,且配置是从头开始生成的。如果要手动组合旧配置和新配置,请使用数据目录中的 *.conf 文件。

8. 启动新的 **PostgreSQL** 服务器：

```
# systemctl start postgresql.service
```


- 运行位于 PostgreSQL 主目录中的 **analyze_new_cluster.sh** 脚本：

```
su postgres -c '~/analyze_new_cluster.sh'
```

- 如果您希望新 PostgreSQL 服务器在引导时自动启动，请运行：

```
# systemctl enable postgresql.service
```

8.3.5.2. 转储和恢复升级

在使用转储和恢复升级时，您需要将所有数据库内容转储到 SQL 文件转储文件中。

请注意，转储和恢复升级比快速升级方法慢，它可能需要在生成的 SQL 文件中进行一些手动修复。

您可以使用此方法迁移以下数据：

- Red Hat Enterprise Linux 7 系统的 **PostgreSQL 9.2**
- 早期 Red Hat Enterprise Linux 8 版本的 **PostgreSQL**
- 与来自 Red Hat Software Collections 相同或更早版本的 **PostgreSQL**：
 - PostgreSQL 9.2**（不再支持）
 - PostgreSQL 9.4**（不再支持）
 - PostgreSQL 9.6**
 - PostgreSQL 10**
 - PostgreSQL 12**

在 Red Hat Enterprise Linux 7 和 Red Hat Enterprise Linux 8 系统中，**PostgreSQL** 数据默认存储在 **/var/lib/pgsql/data/** 目录中。如果是来自 Red Hat Software Collections 的 **PostgreSQL** 版本，则默认数据目录为 **/var/opt/rh/collection_name/lib/pgsql/data/**（使用 **/opt/rh/postgresql92/root/var/lib/pgsql/data/** 目录的 **postgresql92** 除外）。

如果要从 RHEL 8 中的较早 **postgresql** 流升级，请按照[切换到更新的流](#)中介绍的步骤进行，然后迁移 **PostgreSQL** 数据。

要执行转储和恢复升级，请将用户更改为 **root**。

以下介绍了从 RHEL 7 系统的 **PostgreSQL 9.2** 迁移到 RHEL 8 的 **PostgreSQL**。

- 在 RHEL 7 系统中，启动 **PostgreSQL 9.2** 服务器：

```
# systemctl start postgresql.service
```

- 在 RHEL 7 系统中，将所有数据库内容转储到 **pgdump_file.sql** 文件中：

```
su - postgres -c "pg_dumpall > ~/pgdump_file.sql"
```

- 确保正确转储数据库：

```
su - postgres -c 'less "$HOME/pgdump_file.sql"
```


结果会显示到转储的 sql 文件的路径：`/var/lib/pgsql/pgdump_file.sql`。

4. 在 RHEL 8 系统中,启用您要迁移的流（版本）：

```
# yum module enable postgresql:stream
```

使用所选 **PostgreSQL** 服务器版本替换 *stream*。

如果使用默认流（提供了 **PostgreSQL 10**），则可以省略这一步。

5. 在 RHEL 8 系统中安装 **postgresql-server** 软件包：

```
# yum install postgresql-server
```

另外,如果您在 RHEL 7 中使用了任何 **PostgreSQL** 服务器模块，也需要在 RHEL 8 系统中安装它们。如果您需要编译第三方 **PostgreSQL** 服务器模块，使用 **postgresql-devel** 软件包构建它。

6. 在 RHEL 8 系统中，初始化新 **PostgreSQL** 服务器的数据目录：

```
# postgresql-setup --initdb
```

7. 在 RHEL 8 系统中，将 **pgdump_file.sql** 复制到 **PostgreSQL** 主目录中,检查是否已正确复制该文件：

```
su - postgres -c 'test -e "$HOME/pgdump_file.sql" && echo exists'
```

8. 复制 RHEL 7 系统中的配置文件：

```
su - postgres -c 'ls -l $PGDATA/*.conf'
```

要复制的配置文件包括：

- `/var/lib/pgsql/data/pg_hba.conf`
- `/var/lib/pgsql/data/pg_ident.conf`
- `/var/lib/pgsql/data/postgresql.conf`

9. 在 RHEL 8 系统中，启动新的 **PostgreSQL** 服务器：

```
# systemctl start postgresql.service
```

10. 在 RHEL 8 系统中，从转储的 sql 文件中导入数据：

```
su - postgres -c 'psql -f ~/pgdump_file.sql postgres'
```



注意

当从来自 Red Hat Software Collections 的 **PostgreSQL** 的版本升级时，将命令调整为包含 **scl enable collection_name**。例如：要从 **rh-postgresql96** Software Collection 中转储数据，使用以下命令：

```
su - postgres -c 'scl enable rh-postgresql96 "pg_dumpall > ~/pgdump_file.sql"'
```

第 9 章 配置打印

Red Hat Enterprise Linux 8 上的打印基于 Common Unix Printing System(CUPS)。

本文档描述了如何配置机器以作为 CUPS 服务器运行。

9.1. 激活 CUPS 服务

这部分论述了如何在您的系统中激活 **cups** 服务。

先决条件

- **cups** 软件包（可在 Appstream 软件仓库中找到）必须安装到您的系统中：

```
# yum install cups
```

流程

1. 启动 **cups** 服务：

```
# systemctl start cups
```

2. 将 **cups** 服务配置为在引导时自动启动：

```
# systemctl enable cups
```

3. （可选）检查 **cups** 服务的状态：

```
$ systemctl status cups
```

9.2. 打印设置工具

要实现各种与打印相关的任务，您可以选择以下工具之一：

- CUPS Web 用户界面(UI)
- GNOME 控制中心



警告

Red Hat Enterprise Linux 7 中使用的 **Print Settings** 配置工具不再可用。

使用以下工具可以实现的任务包括：

- 添加和配置新打印机
- 维护打印机配置

- 管理打印机类

请注意,本文档只涵盖 CUPS Web 用户界面(UI)的打印。如果要使用 GNOME 控制中心 打印,则需要使用 GUI。有关使用 GNOME 控制中心 打印的更多信息,请参阅 [使用 GNOME 打印开始](#)。

9.3. 访问并配置 CUPS WEB UI

本节论述了访问 CUPS Web 用户界面(web UI),并将它配置为能够通过这个接口管理打印。

流程

访问 CUPS Web UI:

1. 通过在 `/etc/cups/cupsd.conf` 文件中设置 **Port 631** 来允许 CUPS 服务器侦听网络的连接:

```
#Listen localhost:631
Port 631
```



警告

启用 CUPS 服务器侦听端口 631 会为服务器访问的任何地址打开这个端口。因此,仅在无法从外部网络访问的本地网络中使用此设置。如果需要从外部网络访问服务器,但您只想为本地网络打开端口 631,请在 `/etc/cups/cupsd.conf` 文件中设置以下内容: **#Listen <server_local_IP_address>:631**, 其中 `<server_local_IP_address>` 是一个无法从外部网络访问的,但可以被本地系统访问的 IP 地址。

2. 允许您的系统通过在 `/etc/cups/cupsd.conf` 文件中包括以下内容来访问 CUPS 服务器:

```
<Location />
Allow from <your_ip_address>
Order allow,deny
</Location>
```

其中 `<your_ip_address>` 是系统的实际 IP 地址。您还可以将正则表达式用于子网。



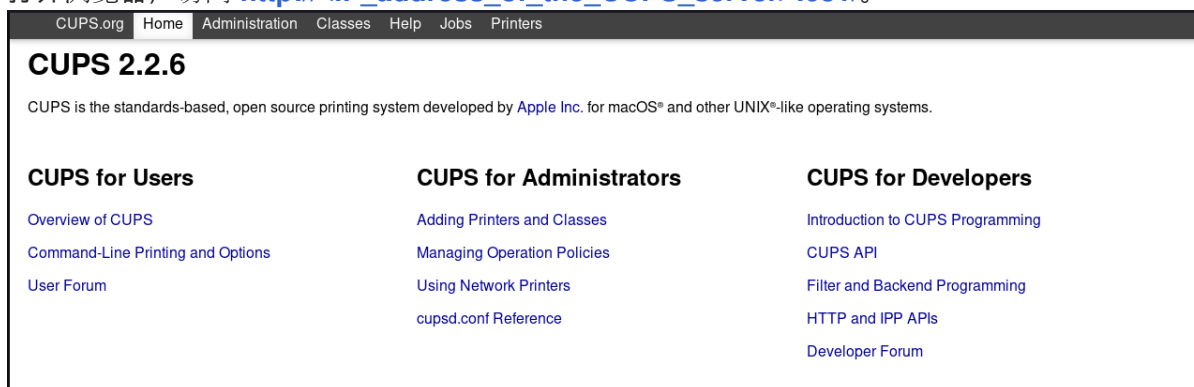
警告

CUPS 配置提供了 `<Location>` tag 中的 **Allow from all** 指令,但红帽不推荐使用它,除非您计划将 CUPS 向外部互连网络开放,或该服务器位于专用网络中。设置 **Allow from all** 为所有可通过端口 631 连接到服务器的用户启用访问。如果您将 **Port** 指令设置为 631,且服务器可从外部网络访问,互联网中的任何人都可以访问系统中的 CUPS 服务。

3. 重启 cups.service:

```
# systemctl restart cups
```

4. 打开浏览器，访问 http://<IP_address_of_the_CUPS_server>:631/。



现在，**Administration** 菜单以外的所有菜单都可用。

如果您点击 **Administration** 菜单,您收到 **Forbidden** 信息：



要访问 **Administration** 菜单,请按照 第 9.3.1 节 “获取 CUPS Web UI 的管理访问权限” 中的说明操作。

9.3.1. 获取 CUPS Web UI 的管理访问权限

本节论述了如何获取 CUPS Web UI 的管理访问权限。

流程

1. 要访问 CUPS Web UI 中的 **Administration** 菜单，请在 `/etc/cups/cupsd.conf` 文件中包括以下内容：

```
<Location /admin>
Allow from <your_ip_address>
Order allow,deny
</Location>
```



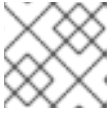
注意

使用系统的真实 IP 地址替换 `<your_ip_address>`。

2. 要访问 CUPS Web UI 中的**配置文件**,请在 `/etc/cups/cupsd.conf` 文件中包括以下内容：

```
<Location /admin/conf>
AuthType Default
Require user @SYSTEM
```

```
Allow from <your_ip_address>
Order allow,deny
</Location>
```

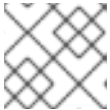


注意

使用系统的真实 IP 地址替换 **<your_ip_address>**。

- 要访问 CUPS Web UI 中的日志文件,请在 `/etc/cups/cupsd.conf` 文件中包括以下内容 :

```
<Location /admin/log>
AuthType Default
Require user @SYSTEM
Allow from <your_ip_address>
Order allow,deny
</Location>
```



注意

使用系统的真实 IP 地址替换 **<your_ip_address>**。

- 要指定 CUPS Web UI 中验证请求使用加密,请在 `/etc/cups/cupsd.conf` 文件中包括 **DefaultEncryption:**

```
DefaultEncryption IfRequested
```

使用这个设置,您将接收一个验证窗口,在试图访问 **Administration** 菜单时输入允许添加打印机的用户的用户名。但是,还有其他选项可以设置 **DefaultEncryption**。详情请查看 `cupsd.conf` man page。

- 重启 **cups** 服务 :

```
# systemctl restart cups
```



警告

如果您没有重启 **cups** 服务,则不会应用 `/etc/cups/cupsd.conf` 中的更改。因此,您将无法获取 CUPS Web UI 的管理访问权限。

其它资源

- 有关如何使用 `/etc/cups/cupsd.conf` 文件配置 CUPS 服务器的详情请参考 `cupsd.conf` man page。

9.4. 在 CUPS WEB UI 中添加打印机

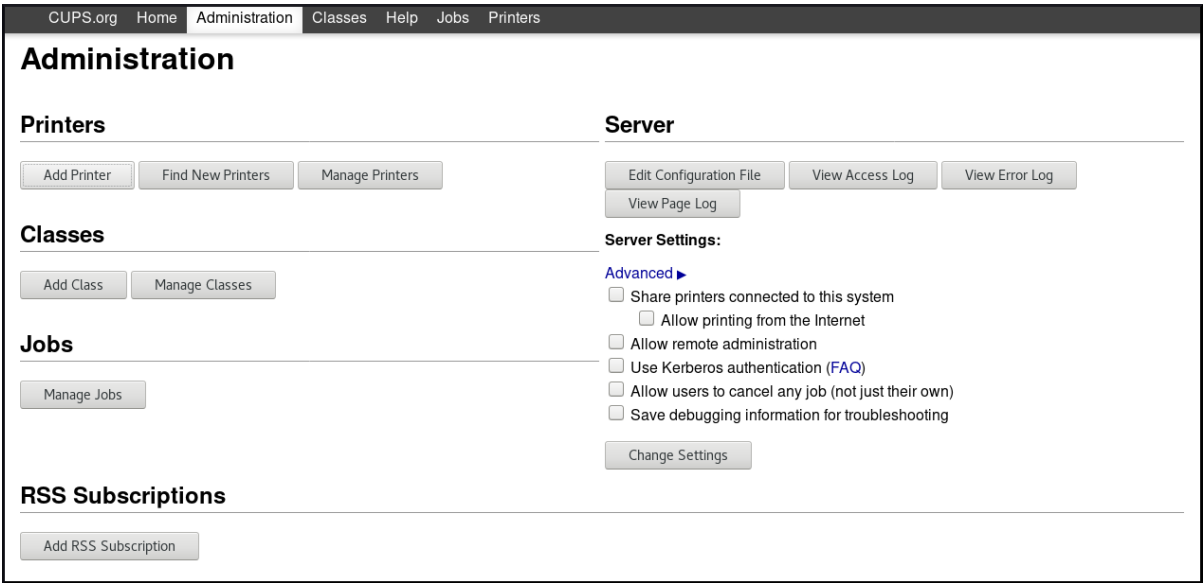
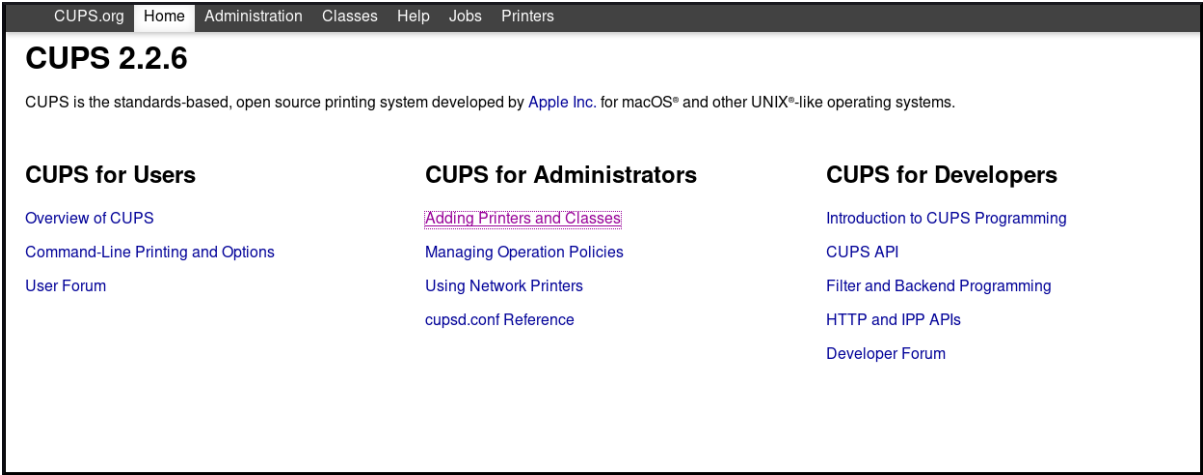
这部分论述了如何使用 CUPS Web 用户界面添加新打印机。

先决条件

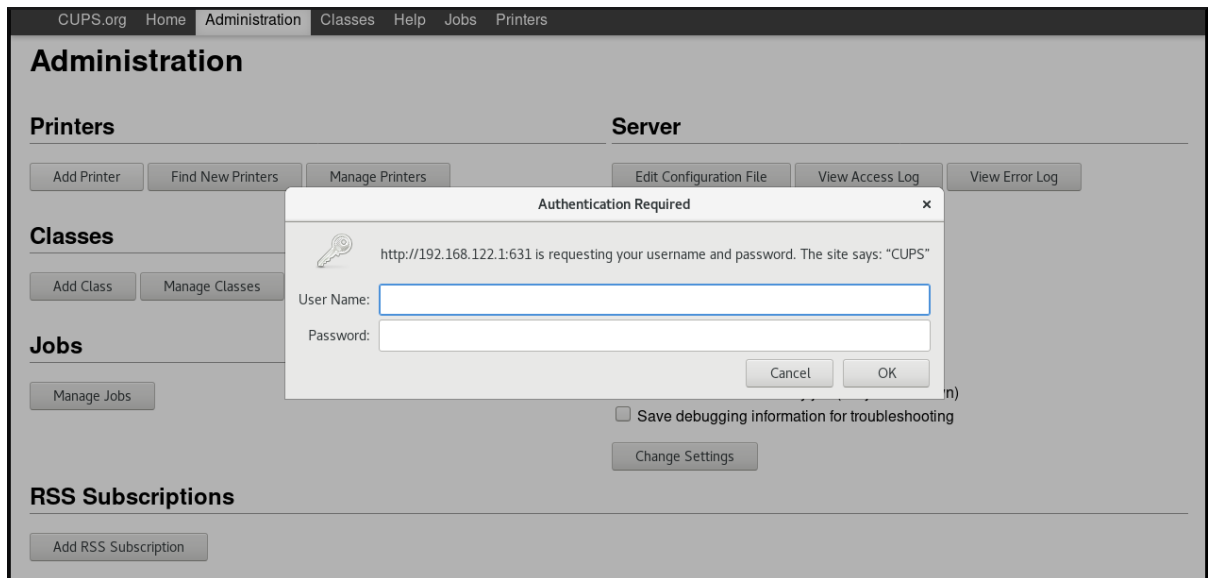
- 您已获得对 CUPS Web UI的管理访问权限，如 第 9.3.1 节 “获取 CUPS Web UI 的管理访问权限”所述。

流程

1. 启动 CUPS Web UI, 如 第 9.3 节 “访问并配置 CUPS Web UI”
2. 前往 **Adding Printers and Classes - Add printer**



3. 使用用户名和密码进行身份验证：

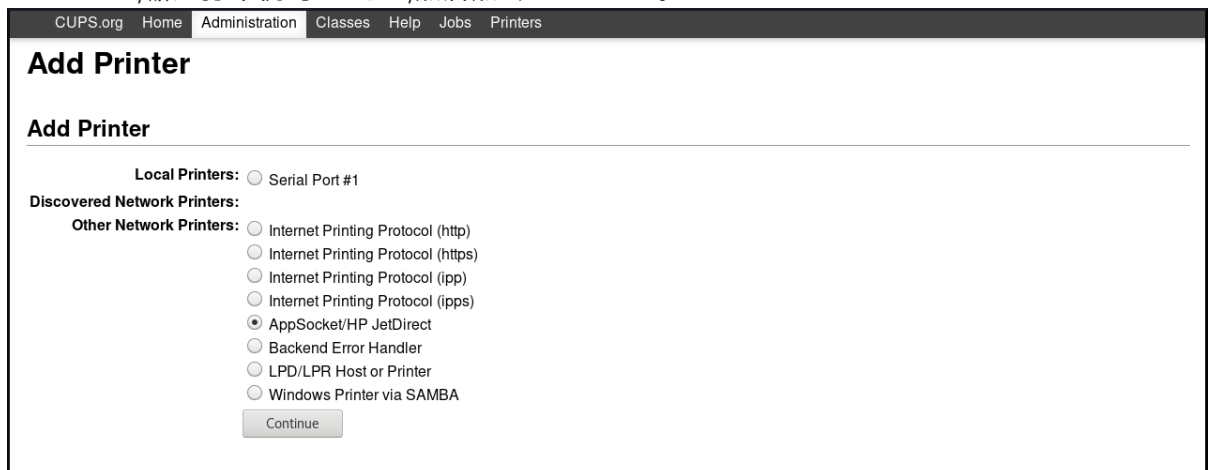


重要

要使用 CUPS Web UI 添加新打印机,您必须作为以下用户之一进行身份验证:

- 超级用户
- 任何具有管理访问权限的用户,由 **sudo** 命令提供 (在 **/etc/sudoers** 中列出的用户)
- 属于 **printadmin** 组的任何用户,在其中 **/etc/group**

4. 如果一个本地打印机已连接,或者 CUPS 发现一个可用的网络打印机,请选择打印机。如果没有本地打印机或网络打印机,请从 **Other Network Printers** 中选择一种打印机类型,例如 **APP Socket/HP Jet direct**,输入打印机的 IP 地址,然后点击 **Continue**。



5. 如果您选择了如上所示的 **APP Socket/HP Jet direct**, 请输入打印机的 IP 地址, 然后点击 **Continue**。

CUPS.org

Home

Administration

Classes

Help

Jobs

Printers

Add Printer

Add Printer

Connection: socket://10.43.2.198

Examples:

http://hostname:631/ipp/
http://hostname:631/ipp/port1

ipp://hostname/ipp/
ipp://hostname/ipp/port1

lpd://hostname/queue

socket://hostname
socket://hostname:9100

See "Network Printers" for the correct URI to use with your printer.

Continue

6. 您可以添加有关新打印机的更多详情，如名称、描述和位置。要设置要通过网络共享的打印机，请使用 **Share This Printer**，如下所示。

CUPS.org

Home

Administration

Classes

Help

Jobs

Printers

Add Printer

Add Printer

Name: Office1

(May contain any printable characters except "/", "#", and space)

Description: HP LaserJet

(Human-readable description such as "HP LaserJet with Duplexer")

Location: South corridor

(Human-readable location such as "Lab 1")

Connection: socket://10.43.2.198

Sharing: ☒ Share This Printer

Continue

7. 选择打印机厂商，然后点 **Continue**。

CUPS.org

Home

Administration

Classes

Help

Jobs

Printers

Add Printer

Add Printer

Name: Office1

Description: HP LaserJet

Location: South corridor

Connection: socket://10.43.2.198

Sharing: Share This Printer

Make:

(Fuji Xerox)
Dymo
Epson
Generic
HP
Index
Intellitech
Oki
Raw
Ricoh

Continue

Or Provide a PPD File:

Browse...

No file selected.

Add Printer

另外,您还可以通过点击 **Browse...** 底部的 postscript 打印机描述(PPD)文件作为打印机的驱动程序提供 postscript 打印机描述(PPD)文件。

8. 选择打印机的型号, 然后点 **Add Printer**。

9. 添加打印机后,下一个窗口允许您设置默认的打印选项。

点 **Set Default Options** 后,您会收到确认新打印机已被成功添加的确认。

9.5. 在 CUPS WEB UI 中配置打印机

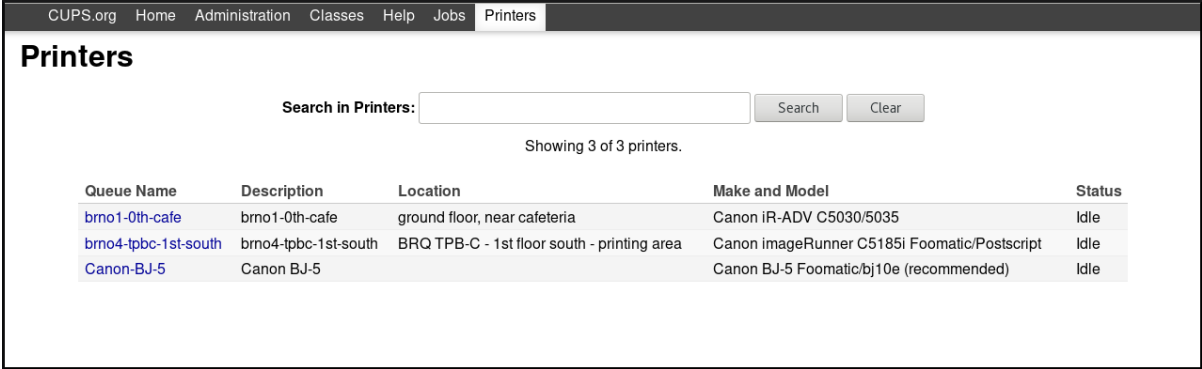
本节论述了如何配置新打印机, 以及如何使用 CUPS Web UI 维护打印机配置。

先决条件

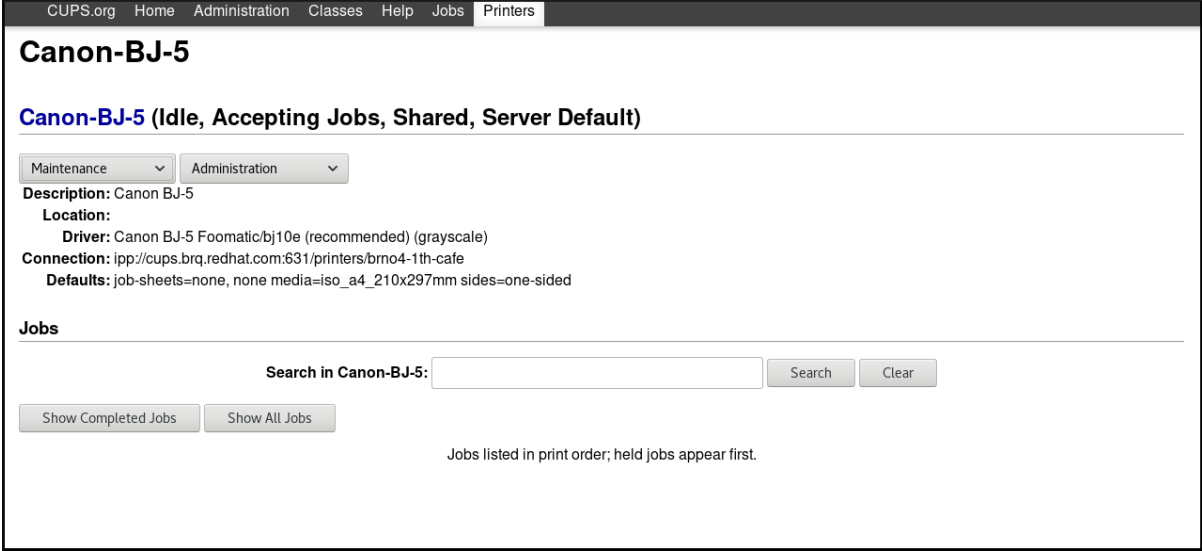
- 您具有对 CUPS Web UI的管理访问权限，如 第 9.3.1 节 “获取 CUPS Web UI 的管理访问权限” 所述。

流程

1. 点击 **Printers** 菜单查看您可以配置的可用打印机。

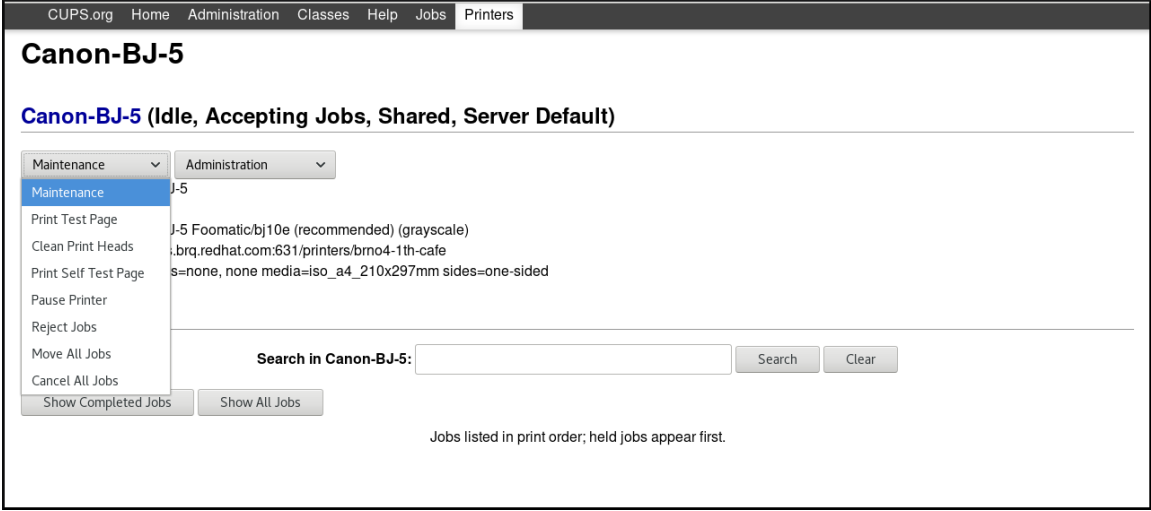


2. 选择您要配置的打印机。

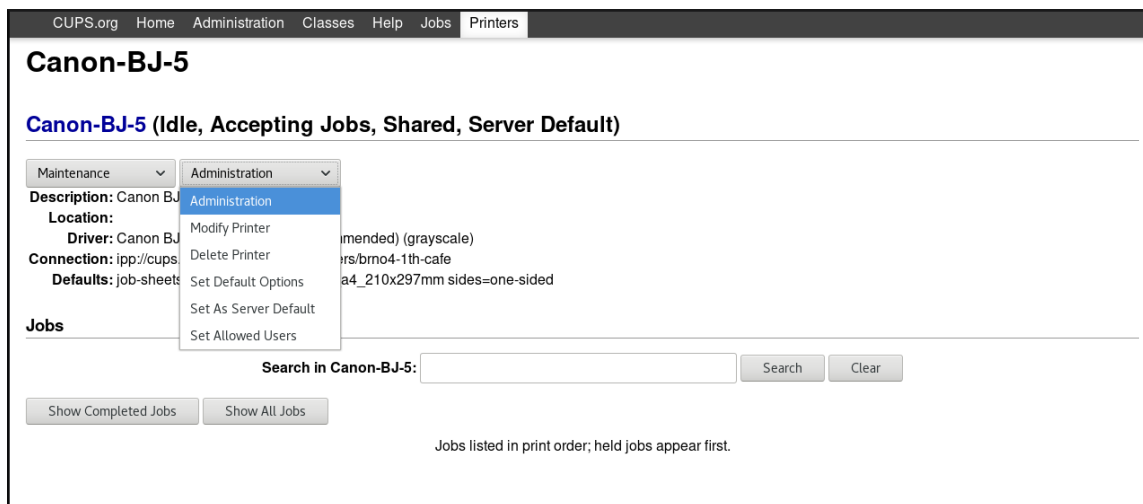


3. 使用其中一个可用菜单来执行您选择的任务：

- 前往 **Maintenance** 进行维护任务。



- 访问 **Administration** 以了解管理任务。



- 您还可以点击 **Show Completed Jobs** 或 **Show All Jobs** 按钮来检查已完成的打印作业或所有活跃的打印作业。

9.6. 使用 CUPS WEB UI 打印测试页面

这部分论述了如何打印测试页面以确保打印机正常工作。

如果满足以下条件之一,您可能想要打印一个测试页面。

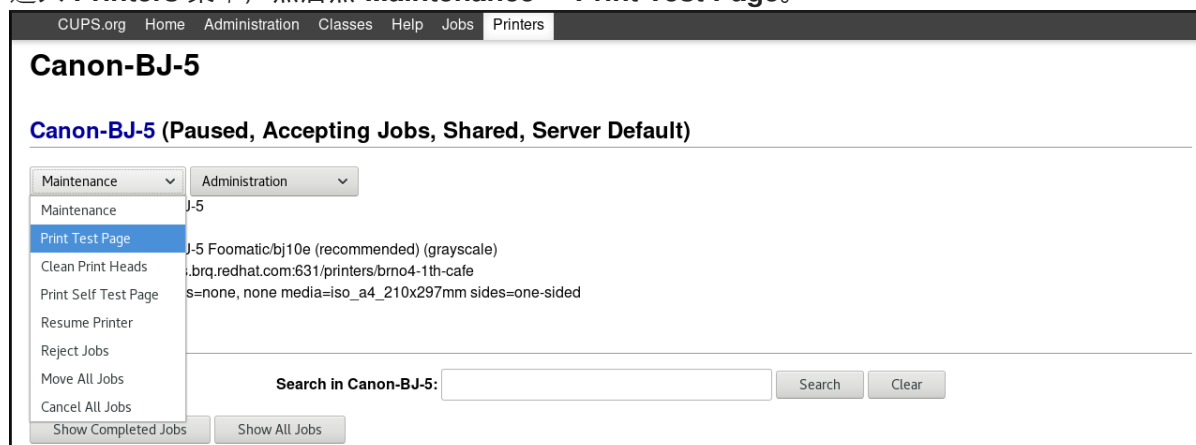
- 已设置打印机。
- 打印机配置已被更改。

先决条件

您具有对 CUPS Web UI 的管理访问权限, 如 第 9.3.1 节 “获取 CUPS Web UI 的管理访问权限” 所述。

流程

- 进入 **Printers** 菜单, 然后点 **Maintenance** → **Print Test Page**。



9.7. 使用 CUPS WEB UI 设置打印选项

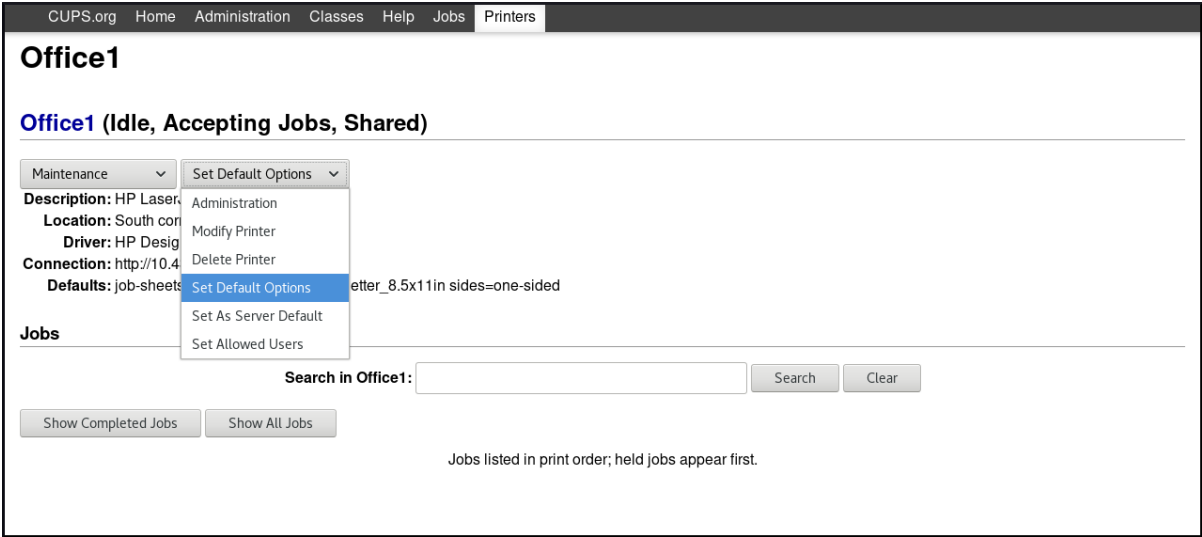
本节论述了如何在 CUPS Web UI 中设置通用的打印选项,如介质大小和类型、打印质量或颜色模式。

先决条件

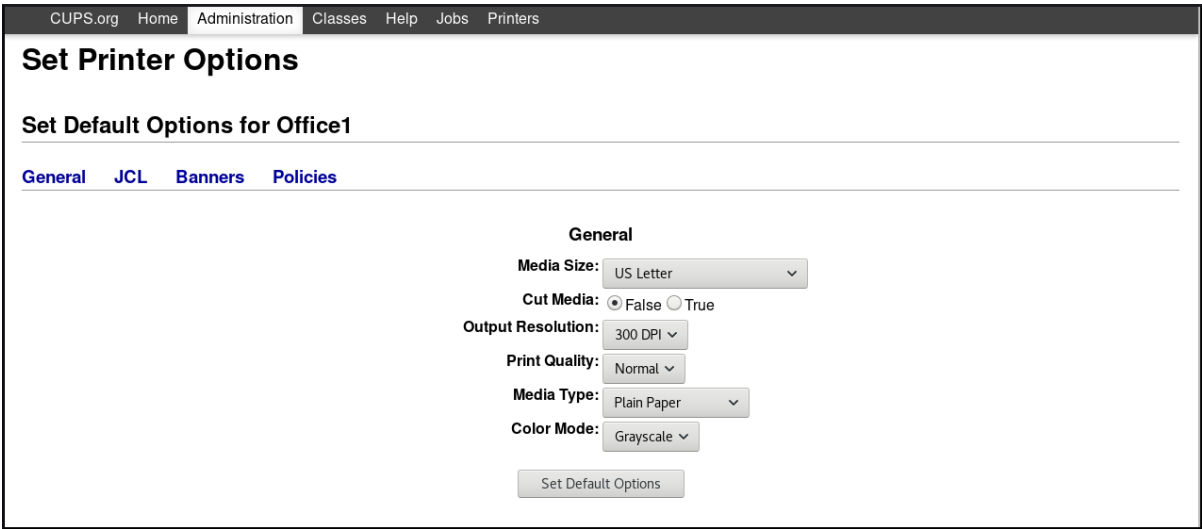
您具有对 CUPS Web UI 的管理访问权限, 如 第 9.3.1 节 “获取 CUPS Web UI 的管理访问权限” 所述。

流程

1. 进入 **Administration** 菜单,然后单击 **Maintenance → Set Default Options**。



2. 设置打印选项。



9.8. 为打印服务器安装证书

要为打印服务器安装证书,您可以选择以下选项之一：

- 使用自签名证书自动安装
- 使用认证机构生成的证书和私钥手动安装

先决条件

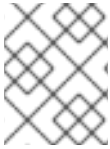
对于服务器上的 `cupsd` 守护进程：

1. 将 `/etc/cups/cupsd.conf` 文件中的以下指令设置为：
Encryption Required
2. 重启 cups 服务：

```
$ sudo systemctl restart cups
```

使用自签名证书自动安装

使用这个选项,CUPS 会自动生成证书和密钥。



注意

自签名证书并不以身份管理(IdM)、Active Directory(AD)或红帽证书系统(RHCS)认证颁发机构生成的证书提供非常安全,但可用于打印位于安全本地网络中的服务器。

流程

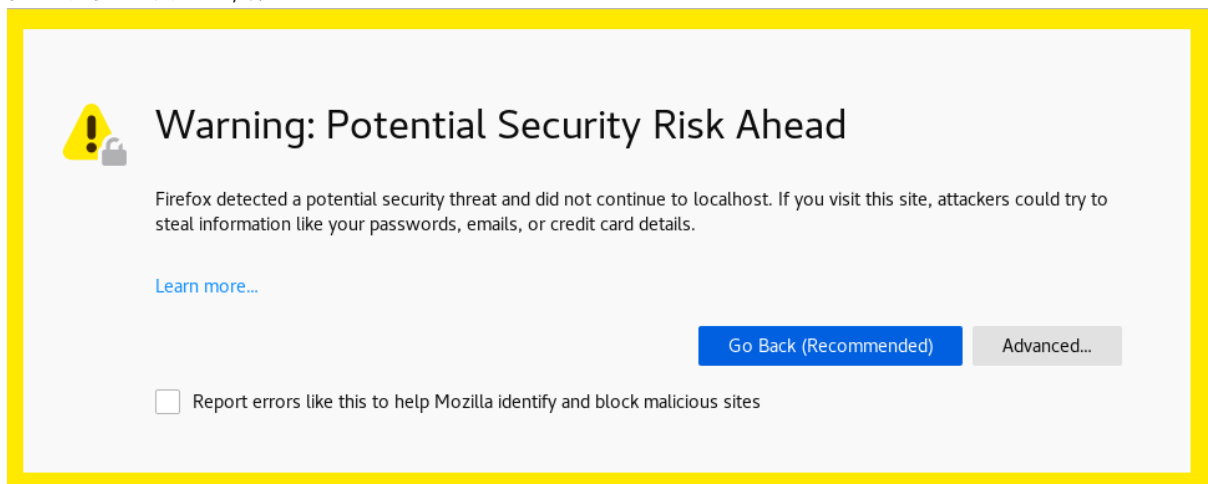
1. 要访问 CUPS Web UI,请打开浏览器并进入 <https://<server>:631>
其中 <server> 是服务器 IP 地址或服务器主机名。



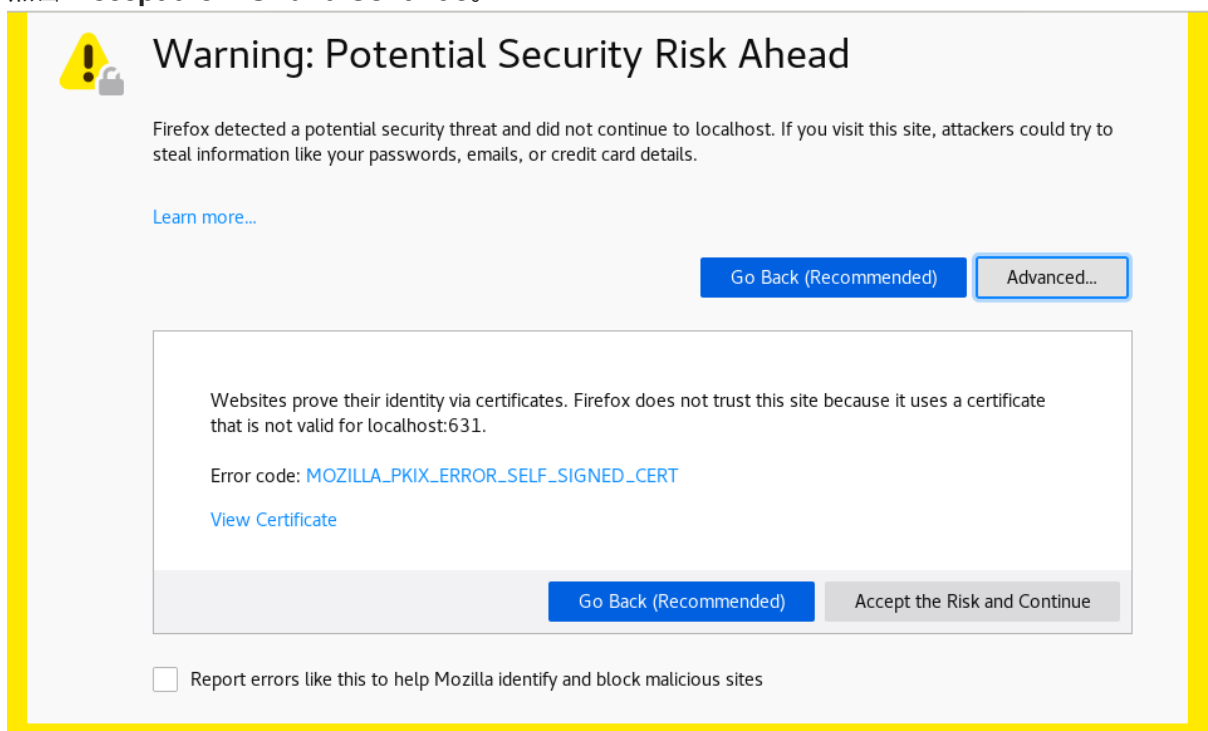
注意

当 CUPS 首次连接到某个系统时,浏览器会显示一条有关自签名证书具有潜在的安全风险的警告。

2. 要确认要继续安全,请点击 **Advanced...**。



3. 点击 **Accept the Risk and Continue**。



CUPS 现在开始使用自生成的证书和密钥。



注意

当您在自动安装后访问 CUPS Web UI 时,浏览器会在地址栏中显示警告图标。这是因为您通过确认安全风险警告添加了安全异常。如果要永久删除此警告图标,使用认证认证机构生成的证书和私钥手动安装。

使用认证认证机构生成的证书和私钥手动安装

对于公共网络中的打印服务器或在浏览器中永久删除警告,请手动导入证书和密钥。

先决条件

- 您有 IdM、AD 或 RHCS 认证颁发机构生成的证书和私钥文件。

流程

1. 将 **.crt** 和 **.key** 文件复制到您要使用 CUPS Web UI 的系统的 **/etc/cups/ssl** 目录中。
2. 将复制的文件重命名为 **<hostname>.crt** 和 **<hostname>.key**。
将 **<hostname>** 替换为您要连接 CUPS Web UI 的系统主机名。
3. 将以下权限设置为重命名的文件：
 - **# chmod 644 /etc/cups/ssl/<hostname>.crt**
 - **# chmod 644 /etc/cups/ssl/<hostname>.key**
 - **# chown root:root /etc/cups/ssl/<hostname>.crt**
 - **# chown root:root /etc/cups/ssl/<hostname>.key**
4. 重启 cups 服务：
 - **# systemctl restart cupsd**

9.9. 使用 SAMBA 打印到使用 KERBEROS 验证的 WINDOWS 打印服务器

使用 **samba-krb5-printing** 打包程序,登录到 Red Hat Enterprise Linux 的 Active Directory(AD)用户可以使用 Kerberos 向 Active Directory(AD)进行身份验证,然后打印到将打印作业转发到 Windows 打印服务器的本地 CUPS 打印服务器。

进行这个配置的好处是,Red Hat Enterprise Linux 上的 CUPS 管理员不需要在配置中存储固定的用户名和密码。CUPS 使用发送打印作业的用户 Kerberos ticket 验证 AD。

这部分论述了如何为这种情况配置 CUPS。



注意

红帽只支持从本地系统向 CUPS 提交打印作业,而不支持在 Samba 打印服务器中重新共享打印机。

先决条件

- 要添加到本地 CUPS 实例的打印机会在 AD 打印服务器中共享。
- 作为 AD 的成员加入 Red Hat Enterprise Linux 主机。详情请查看 [第 3.5.1 节 “将 RHEL 系统添加到 AD 域中”](#)。
- CUPS 安装在 Red Hat Enterprise Linux 中, **cups** 服务正在运行。详情请查看 [第 9.1 节 “激活 cups 服务”](#)。
- 打印机的 PostScript Printer Description(PPD)文件存储在 **/usr/share/cups/model/** 目录中。

流程

1. 安装 **samba-krb5-printing**、**samba-client** 和 **krb5-workstation** 软件包：

```
# yum install samba-krb5-printing samba-client krb5-workstation
```

2. 可选：作为域管理员授权并显示 Windows 打印服务器上共享的打印机列表：

```
# kinit administrator@AD_KERBEROS_REALM
# smbclient -L win_print_srv.ad.example.com -k
```

Sharename	Type	Comment
-----	----	-----
...		
Example	Printer	Example
...		

3. 可选：显示 CUPS 模型列表以识别打印机的 PPD 名称：

```
lpinfo -m
...
samsung.ppd Samsung M267x 287x Series PXL
...
```

在下一步中添加打印机时,您需要 PPD 文件的名称。

4. 在 CUPS 中添加打印机：

```
# lpadmin -p "example_printer" -v smb://win_print_srv.ad.example.com/Example -m
samsung.ppd -o auth-info-required=negotiate -E
```

该命令使用以下选项：

- **-p printer_name** 在 CUPS 中设置打印机的名称。
- **-v URI_to_Windows_printer** 将 URI 设置为 Windows 打印机。使用以下命令：
smb://host_name/printer_share_name。
- **-m PPD_file** 设置打印机使用的 PPD 文件。
- **-o auth-info-required=negotiate** 将 CUPS 配置为在将打印任务转发到远程服务器时使用 Kerberos 验证。
- **-E** 启用打印机和 CUPS 接受打印机的作业。

验证步骤

1. 以 AD 域用户身份登录 Red Hat Enterprise Linux 主机。
2. 以 AD 域用户身份进行身份验证：

```
# kinit domain_user_name@AD_KERBEROS_REALM
```

3. 将文件输出到您添加到本地 CUPS 打印服务器的打印机：

```
# lp -d example_printer file
```

9.10. 使用 CUPS 日志

9.10.1. CUPS 日志的类型

CUPS 提供三种不同类型的日志：

- 错误日志 - 存储错误消息、警告和调试信息。
- 访问日志 - 存储有关访问 CUPS 客户端和 Web UI 的次数的信息。
- 页面日志 - 保存每个打印作业打印页面总数的信息。

在 Red Hat Enterprise Linux 8 中,所有三种类型都集中记录在 systemd-journald 中,以及其它程序的日志。



警告

在 Red Hat Enterprise Linux 8 中,日志不再存储在 `/var/log/cups` 目录中的特定文件中,这些文件中在 Red Hat Enterprise Linux 7 中使用。

9.10.2. 访问 CUPS 日志

本节论述了如何访问：

- 所有 CUPS 日志
- 特定打印作业的 CUPS 日志
- CUPS 日志在指定时间范围内

9.10.2.1. 访问所有 CUPS 日志

流程

- 从 systemd-journald 过滤 CUPS 日志：

```
$ journalctl -u cups
```


9.10.2.2. 访问特定打印作业的 CUPS 日志

流程

- 过滤特定打印作业的日志：

```
$ journalctl -u cups JID=N
```

其中 **N** 是打印作业的数量。

9.10.2.3. 根据特定时间框架访问 CUPS 日志

流程

- 在指定时间范围内过滤日志：

```
$ journalctl -u cups --since=YYYY-MM-DD --until=YYYY-MM-DD
```

其中 **YYYY** 是年, **MM** 是月, **DD** 是日。

9.10.2.4. 相关信息

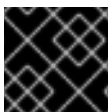
有关访问 CUPS 日志的详情请参考 **journalctl** man page。

9.10.3. 配置 CUPS 日志位置

这部分论述了如何配置 CUPS 日志的位置。

在 Red Hat Enterprise Linux 8 中, CUPS 日志被默认记录到 **systemd-journald**, **/etc/cups/cups-files.conf** 文件中的以下默认设置可保证：

```
ErrorLog syslog
```



重要

红帽建议保留 CUPS 日志的默认位置。

如果要将日志发送到不同的位置, 则需要按如下方式更改 **/etc/cups/cups-files.conf** 文件中的设置：

```
ErrorLog <your_required_location>
```



警告

如果您更改了 CUPS 日志的默认位置, 您可能会遇到意外行为或者 SELinux 问题。

context: Deploying-different-types-of-servers