Oracle Linux 9

Managing Samba

G26610-01
July 2025

Oracle Linux 9 Managing Samba,

G26610-01

Contents

Preface

This chapter includes information about managing Samba in Oracle Linux 9.

Documentation License

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://www.oracle.com/corporate/accessibility/.

Access to Oracle Support for Accessibility

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://www.oracle.com/corporate/accessibility/learning-support.html#support-tab.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve.

Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

1

About Samba

Samba is an open source implementation of the Server Message Block (SMB) protocol that lets Oracle Linux share file and printer resources with clients that are running different OSs, including Windows, other Linux flavors, and macOS.

For example, use Samba to configure a Linux host to share files at a location that Windows users on the network can access using the following syntax:

```
\\samba_server\share_name
```

In the preceding example:

- *samba_server* is the IP address or a network resolvable host name of the system running the Samba service.
- *share_name* is the name of the resource as defined in the Samba configuration file `/etc/samba/smb.conf`.

> **Note:**
>
> The format of the location path depends upon the client OS. On UNIX and Linux based OSs, including macOS, the path format is:
>
> `smb://samba_server/share_name`.

Other features of Samba include:

- Samba can integrate with a Windows workgroup and an Active Directory (AD) domain.
- Samba implements the Distributed Computing Environment Remote Procedure Call (DCE RPC) protocol that's used by Microsoft Windows to provision file and print services for Windows clients.
- Samba uses the NetBIOS over TCP/IP protocol so that computer applications that depend on the NetBIOS API can work on TCP/IP networks.

Samba Services

The Samba server consists of the following important services:

`smb` Service
The `smb` service makes file sharing and printing services available using the SMB protocol. This service is also responsible for resource locking and for authenticating connecting users.

The `smb systemd` service starts and stops the `smbd` daemon. The following is an example command:

```
sudo systemctl start smb.service
```

To use the `smbd` service, you need to install the `samba` package on the system.

`nmb` Service
The `nmb` (NetBIOS Message Block) service provides host name and IP resolution by using the NetBIOS over IPv4 protocol. The `nmb` service can also browse the SMB network for domains, workgroups, hosts, file shares, and printers.
The `nmb systemd` service starts and stops the `nmbd` daemon. The following is an example command:

```
sudo systemctl start nmb.service
```

To use the `nmbd` service, you need to install the `samba` package on the system.

`winbind` Service
The `winbind` service is a Name Service Switch (NSS) daemon for resolving AD Users and Groups. The daemon lets AD Users securely access services that are hosted on the Samba server.
The `winbind systemd` service starts and stops the `winbindd` daemon. The following is an example command:

```
sudo systemctl start winbind.service
```

To use the `winbindd` service, install the `samba-winbind` package.

Note:

If you're setting up Samba as a domain member, you must start the `winbind` service before starting the `smb` service. This makes domain users and groups available to the local system.

The Samba Configuration File

Samba uses the `/etc/samba/smb.conf` file to manage Samba configuration.

`smb.conf` File Structure Overview

The `smb.conf` file consists of several sections that you configure to make the required services for a specific Samba configuration available. Consider the following sample extract from an `smb.conf` file:

```
#========== Global Settings =======
[global]
security = ADS
realm = EXAMPLE.REALM
```

```
password server = krbsvr.example.com
.
.
.
load printers = yes
printing = cups
printcap name = cups

#========== Share Definitions =========

[homes]
comment = User home directories
path = /data/pchome/%S
valid users = %S, WWW.EXAMPLE.COM\%S
browsable = no
read only = no
guest ok = no

[printers]
comment = All Printers
path = /var/spool/samba
printable = yes


[test_share]
comment = Shared /usr/local/test_share directory created for tests
path = /usr/local/test_share
valid users = @examplegroup
browsable = yes
read only = no
```

The following list describes the sections in the preceding configuration example:

`[global]`
This section contains global settings for the Samba server.
In the preceding example, the `security` parameter value of `ADS` means the server is a member of an AD domain that's running in native mode. In this scenario, Samba relies on tickets issued by the Kerberos server to authenticate clients who want to access local services.

`[homes]`
The `[homes]` section provides a personal share for users that log onto the Samba server. In the example, the location of each user's home directory is set by the line `path = /data/pchome/%S` (the `%S` macro is substituted with the username). The settings for `browsable = no` and `read only = no` prevent other users from browsing home directories, while granting full access to valid users.

Important:

Be careful with settings, such as security, especially in the special sections `[global]`, `[homes]`, and `[printers]`.
For example, if guest access is specified in the `[homes]` section, all home directories are visible to all clients without a password.
You might consider using the `invalid users` parameter for users such as `root` and other users with administrative privileges.

`[printers]`
Specifies print services. The `path` parameter specifies the location of a spooling directory that receives print jobs from Windows clients before submitting them to the local print spooler. Samba advertises all locally configured printers on the server.

`[`*`test_share`*`]`
Specifies a share named *test_share*, which grants users belonging to group *examplegroup* browsing and write permissions to the `/usr/local/test_share` directory.

Note:

The `read only = no` configuration entry is essential to ensure Samba shares the directory as a writeable share.

For more information see `/etc/samba/smb.conf.example`, `smb.conf(5)` manual page, and https://wiki.samba.org/index.php/User_Documentation

Using the `testparm` Program to Validate Samba Configuration File Content

You use the `testparm` program to validate a Samba configuration file after making configuration changes. The `testparm` program detects invalid parameters and values and highlights incorrect settings such as incorrect ID mappings.

Note:

`testparm` checks a configuration file for internal correctness only. The `testparm` command isn't capable of testing whether configured services are available or work as expected.

The following example shows how you might use the command to test a copy of the file you're working on:

```
sudo testparm /etc/samba/smb.conf.my_copy


Load smb config files from /etc/samba/smb.conf.my_copy
Loaded services file OK.
...
```

If, instead of a copy, you want to test the default Samba configuration file, you don't have to specify the file as a parameter. Run `testparm` as follows:

```
sudo testparm
```

If the `testparm` command reports any errors or misconfiguration in the configuration file, you must fix the problem and then reissue the command.

For more information, see the `testparm(1)` manual page.

Best Practice When Editing Samba Configuration

Samba services reload their configuration as follows:

- Most configuration values are reloaded automatically, every 3 minutes.
- You can also manually request a reload, for example by using the `smbcontrol all reload-config command`.

Note:

Some parameters, such as `security`, require a restart of the `smb` service to take effect.

The frequent reloading of configuration values doesn't give you much time to validate any changes you're planning to make to `/etc/samba/smb.conf`. Therefore, as best practice, first test the changes on a copy of the configuration file. The following steps describe how you might do this:

1. Make a copy of the samba configuration file.

   ```
   sudo cp /etc/samba/smb.conf /etc/samba/samba.conf.mycopy
   ```

2. Edit the copy of the file in a text editor, such as `vi`:

   ```
   sudo vi /etc/samba/samba.conf.mycopy
   ```

3. Validate the changes using `testparm`:

   ```
   sudo testparm /etc/samba/smb.conf.my_copy
   ```

4. Overwrite the original file with the copy you have validated:

   ```
   sudo mv /etc/samba/samba.conf.my_copy /etc/samba/smb.conf
   ```

5. Use the `smbcontrol all reload-config` command to reload the configuration:

   ```
   sudo smbcontrol all reload-config
   ```

Samba Server Roles

The following sections give an overview of different roles you can configure for a Samba server.

Standalone

You can configure the Samba server role as a standalone server in small networks, such as peer-to-peer workgroups, where the server isn't required to be part of a domain.

To provide a Windows user with authenticated access to a share on a standalone server, you create the following accounts on the Samba server:

- **A Local Linux Account**

  The local Linux account is required to validate access to local file system objects.

- **A Samba Account**

  In a standalone configuration, Samba authenticates users to a local database rather than a domain controller. You use the Samba `smbpasswd` command to create such accounts.

In addition to authenticated access, you can also enable guest access for users to connect to some services without authentication.

Domain Member of an Active Directory Domain

> **Note:**
>
> Oracle Linux doesn't support running Samba as an AD domain controller (DC)

You configure a Samba server's role to be a domain member of an Active Directory (AD) domain when you need to set up Samba shares in an AD domain network.

The Samba AD domain member setup requires the following:

- **Installation of `Kerberos`**

  The Samba server uses `Kerberos` to authenticate Windows AD users against the domain controller.

- **Installation of the `winbind` service**

  The `winbind` service provides information about Windows AD users and groups to the Linux OS. Hence, when a Samba server is configured as an AD member, you don't need to manually create local Linux users and groups for authenticated access to the Samba shares.

- **Configuration of ID Mapping Backends**

  Samba provides various ID mapping methods, referred to as backends, that can be configured to map each Linux `GID` and `UID` to its corresponding Windows `SID`. You choose which backends you want to use and configure them in the `/etc/samba/smb.conf` file.

  See ID Mapping Backends in the Active Domain Member Setup

2 ID Mapping Backends in the Active Domain Member Setup

Linux and Windows each use a different ID system to identify groups and users.

Linux assigns unique GID and UID numbers to groups and users, whereas Windows uses a Security Identifier values (SID) for each user and group.

The `winbind` service maintains the necessary mapping of each Linux GID and UID to its corresponding Windows SID. However, you're responsible for specifying which of the available mapping methods, or backends as they're called in Samba, to use for this mapping.

Overview of ID Mapping in the Samba Configuration File

You configure the mapping in the `[global]` section of the Samba configuration file `/etc/samba/smb.conf`.

Consider the following example extract from an `/etc/samba/smb.conf` file:

```
#========== Global Settings =======
[global]
security = ADS
.
.
.
#........................................
#   Using tdb backend for default* domain.
#   UID/GID range 1000000-2000000
#........................................
idmap config * : backend = tdb
idmap config * : range = 1000000-2000000


#........................................
#   Using rid backend to map EXAMPLE.COM users
#   UID/GID range 10000-49999
#........................................
idmap config EXAMPLE.COM : backend  = rid
idmap config EXAMPLE.COM: range = 10000-49999



#........................................
#   Using rid backend to map EXAMPLE.NET users
#   UID/GID range 50000-99999
#........................................
idmap config EXAMPLE.NET : backend  = rid
idmap config EXAMPLE.NET : range = 50000 -99999
```

The preceding example extract shows the following configurations:

- The Samba server is a member of the *EXAMPLE.COM* AD domain and uses the `rid` backend to map `SIDs` belonging to that domain. The backend is authoritative for those `SIDs` that the `rid` method translates to `UIDs` and `GIDs` within the range specified in the file (`10000-49999`).
- The Samba server also provides share access to a trusted AD domain *EXAMPLE.NET*. The trusted domain is also configured to use the `rid` backend. The range for *EXAMPLE.NET* is `50000-99999`.
- The default `*` domain uses backend `tdb`. The `tdb` range is specified as `1000000-2000000`

WARNING:

- ID ranges must not overlap.
- Only one range can be assigned to a domain.
- After a range has been set and Samba has started using the range, you can only increase the upper number of the range. Any other change to the range can result in new ID assignments, and thus a loss of file ownership data.

The following sections give a further overview on using the different backends to configure ID Mapping for domains.

For more information, see `/etc/samba/smb.conf.example` and the `smb.conf(5)` manual pages. See also https://wiki.samba.org/index.php/User_Documentation and upstream documentation.

Domains That Require ID Mapping Configuration

You need to configure ID Mapping for the following:

- The AD domain of which the Samba server is a member.
- Each trusted AD domain that might access the Samba Server.
- The `*` default domain.

  The default domain includes Samba built-in accounts and groups, such as `BUILTIN\Administrators`.

Available Backends

The following table describes the most commonly used backends and their different use cases.

Table 2-1 The Most Commonly Used ID Mapping Backends

Backend	Domains With Which the Backend Can Be Used
`tdb`	Use with `*` default domain only.
`ad`	Use with AD domains.

Table 2-1 (Cont.) The Most Commonly Used ID Mapping Backends

Backend	Domains With Which the Backend Can Be Used
`rid`	Use with AD domains.
`autorid`	Can be used both with AD, and `*` default domain.

The following sections give an overview of the backends listed in the preceding table.

`tdb` Mapping Backend

The `tdb` backend is the default backend used by `winbindd` for storing Security Identifier (`SID`), `UID`, and `GID` mapping tables.

The `tdb` backend must only be used for the `*` default domain.

The default domain includes Samba built-in accounts and groups, such as `BUILTIN\Administrators`.

The `tdb` backend is a writeable backend that needs to allocate new user and group IDs to create new mappings.

The ID mappings are local to the server.

`ad` Mapping Backend

The `ad` backend lets `winbind` read the ID mappings from an AD server that uses `RFC2307` schema extensions.

For example, when using the `ad` backend, you set a user's Linux `UID` number by entering its value in their AD account's `uidNumber` attribute.

Some attributes that you set in the Windows AD Server are listed in the following table's first column and the corresponding Linux value each one maps to in the second column:

Table 2-2 Table of Attributes on the AD Server When `ad` Mapping is Used

Attribute Set on Windows AD Server	Corresponding Linux Value to Which AD Attribute Maps
`uidNumber`	`UID`
`gidNumber`	`GID`
`sAMAccountName`	Username or Group Name

Note:

- The list in the preceding table provides an overview. See upstream documentation for more attributes.
- The mapping IDs must be within the range configured in `/etc/samba/smb.conf`. Objects with IDs outside the range aren't available on the Samba server.

Advantages of `ad` include the following:

- `UIDs` and `GIDs` are consistent on all Samba servers that use `ad`.
- The ID values aren't stored in a local database, to reduce the risk of local data corruption and loss of file ownership data.

`rid` Mapping Backend

The `rid` backend is an algorithmic mapping scheme that uses the `RID` (relative identifier) part of the Windows `SID` to map Windows groups and Users to `UIDs` and `GIDs`.

Advantages of `rid` include the following:

- All domain user accounts and groups are automatically available on the domain member providing the mapped ID falls within the domain's `rid` range specified in `/etc/samba/smb.conf`.
- No attributes need to be set for domain users and groups.

`autorid` Mapping Backend

The `autorid` backend works in a similar way to the `rid` ID mapping backend, but one advantage of `autorid` is that it can automatically assign IDs for different domains. You can use the `autorid` backend for the following:

- The `*` default domain and extra domains, without the need to create ID mapping configurations for each of the extra domains.
- Only for specific domains.

3 Configuring a Samba Standalone Server

The following task shows how to configure a standalone Samba server in a small network, such as a peer-to-peer workgroup, where the server isn't required to be part of a domain.

1. Install the `samba` package.

   Run the following command to install `samba`:

   ```
   sudo dnf install samba
   ```

2. Make a backup copy of the Samba configuration file.

   Run the following command to make a copy of the original `/etc/samba/smb.cnf` file:

   ```
   sudo cp /etc/samba/smb.conf /etc/samba/samba.conf.mycopy
   ```

3. Edit the Samba configuration file.

   Edit the `/etc/samba/smb.conf` file and configure the various sections for the services required.

   > **Note:**
   >
   > See The Samba Configuration File for more information on editing a Samba configuration file.

   Consider the following example:

   ```
   #========== Global Settings =========
   [global]

   workgroup = EXAMPLE_WORKGROUP
   netbios name = Server_Netbios_Name

   security = user
   server role = standalone server
   passdb backend = tdbsam
   log file = /var/log/samba/%m
   log level = 1

   #========== File Share =========

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

   This example configures a standalone server named *Server_Netbios_Name* in the workgroup *EXAMPLE_WORKGROUP*, with the following settings:

- `server role = standalone server`

  The type of server. See Samba Server Roles for more information.
- `passdb backend = tdbsam`

  Default setting, where Samba stores user accounts in the `/var/lib/samba/private/passdb.tdb` database.
- `log level = 1`

  Lowest level of logging.
- `log file = /var/log/samba/%m`

  Path to the log file. The `%m` variable is substituted with the NetBIOS name of the client machine.
- `[`*`shareexample`*`]`

  Name of the share is configured as *shareexample*. This is the name users use to access the share.
- `read only = no`

  Ensures that Samba makes the shared directory writeable.

4. Verify the Samba configuration.

   Run the `testparm` command to verify the contents of the Samba configuration file, The `testparm` command detects invalid parameters and values and any incorrect settings such as incorrect ID mapping. If the `testparm` command doesn't report any problems, the Samba services successfully load the configuration that's specified in the `/etc/samba/smb.conf` file. Use the `testparm` command every time you make a change to the Samba configuration.

   ```
   sudo testparm
   ```

   Note:

   The `testparm` command only tests the internal integrity of the Samba configuration file. It can't check whether configured services are available or work as expected.

5. Create a Samba user.

   Create a local Linux user account without a home directory (`-M`) and without a login shell (`-s /sbin/nologin`):

   ```
   sudo useradd -M -s /sbin/nologin exampleUser
   ```

6. Enable the Samba user's account.

   Enable the user by setting a password using the Linux `passwd` command:

   ```
   sudo passwd exampleUser
   Changing password for user exampleUser.
   New password:
   Retype new password:
   ```

```
passwd: all authentication tokens updated successfully.
```

> **Note:**
>
> The local password set in this step isn't the one used by Samba. However, you must set a local password to enable the account for use by Samba (Samba denies access if the account is disabled locally). You create the password Samba uses in the next step.

7. Add the user account to the Samba database.

   Run the `smbpasswd` command to generate a Samba password for the user and add it to the Samba database:

   ```
   sudo smbpasswd -a exampleUser
   New SMB password:
   Retype new SMB password:
   Added user exampleUser
   ```

8. Create a group for Samba users.

   Create a Linux group for the Samba user:

   ```
   sudo groupadd exampleGroup
   ```

   Add the user *exampleUser* to the group *exampleGroup* created in the previous step:

   ```
   sudo usermod -aG exampleGroup exampleUser
   ```

9. Create the share directory.

   Make the share directory you referenced in `/etc/samba/smb.conf` if it doesn't already exist:

   ```
   sudo mkdir -p /srv/samba/shareexample/
   ```

   > **Note:**
   >
   > If you run SELinux in `enforcing` mode, set the `samba_share_t` context on the directory so that SELinux allows Samba to read and write to it:
   >
   > ```
   > sudo semanage fcontext -a -t samba_share_t "/srv/samba/
   > shareexample(/.*)?"
   > sudo restorecon -Rv /srv/samba/shareexample/
   > ```

10. Set group access to the share directory.

Set the group for the share directory to be the group you created in a previous step for the Samba users:

```
sudo chgrp -R exampleGroup /srv/samba/shareexample/
```

11. Set permissions on the share directory.

    The following command sets full owner and group rights to the share directory:

    ```
    sudo chmod 2770 /srv/samba/shareexample/
    ```

12. Configure the firewall.

    Open the required ports and reload the firewall configuration using the `firewall-cmd` utility:

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

13. Start the Samba service.

    Enable and start the `smb` service:

    ```
    sudo systemctl enable --now smb
    ```

4

Configuring a Samba Server as an AD Member

The following procedure shows one way of configuring a Samba server as an AD member:

1. Install the required packages.

   Install the following packages:

   - `realmd`

     The `realmd` tool is used to for joining `Kerberos` realms, such as Active Directory domains.

   - `oddjob` and `oddjob-mkhomedir`

     `oddjob` is a D-Bus service which runs jobs on behalf of client applications.

     `oddjob-mkhomedir` is an `oddjob` helper which creates and populates home directories.

   - `samba-winbind-clients`, `samba-winbind`, `samba-common-tools`, `samba-winbind-krb5-locator`, and `samba`.

     You can run the following command to install the required packages:

     ```
     sudo dnf install realmd \
      oddjob-mkhomedir \
      oddjob  \
      samba-winbind-clients \
      samba-winbind  \
      samba-common-tools  \
      samba-winbind-krb5-locator  \
      samba
     ```

2. Backup the Samba configuration file.

   Make a backup copy of the Samba configuration file:

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

3. Join the AD domain.

   Use the `realm join` command to join AD domain. The following example assumes you want to join domain *`EXAMPLE.COM`*:

   ```
   sudo realm join --membership-software=samba \
    --client-software=winbind EXAMPLE.COM
   ```

   When you run the command as shown, `realm` does the following:

   - Creates the `/etc/samba/smb.conf` file with membership of the *EXAMPLE.COM* domain configured.

- Adds the `winbind` module for user and group lookups to the `/etc/nsswitch.conf` file.
- Updates the Pluggable Authentication Module (PAM) configuration files in the `/etc/pam.d/`directory
- Starts and enables the `winbind` service.

4. Configure ID mapping.

   Set up ID Mapping in the `/etc/samba/smb.conf` file required in the configuration.

   For further details on ID Mapping see ID Mapping Backends in the Active Domain Member Setup

5. Verify the configuration.

   Check that the entries in the `/etc/samba/smb.conf` file meet all configuration requirements.

   For more information on the configuration file see The Samba Configuration File

6. Check that `winbind` is running.

   Verify that the `winbind` service is running:

   ```
   sudo systemctl status winbind
   ```

   > **Important:**
   >
   > The `winbind` service must be running before you start the `smb` service. Otherwise, Samba can't retrieve domain user and group information.

7. Start the `smb` service.

   After verifying that the `winbind` is running in the preceding step, start and enable the `smb` service:

   ```
   sudo systemctl enable --now smb
   ```

8. Verify the Samba server is working.

   Perform verification steps such as the following:

   - Get the details of a domain user. The following assumes the details of user *exampleuser* in the *EXAMPLE.COM* domain are being retrieved:

     ```
     sudo getent passwd EXAMPLE.COM\\exampleuser
     ```

     ```
     EXAMPLE.COM\exampleuser:*:10000:10000::/home/
     exampleuser@EXAMPLE.COM:/bin/bash
     ```

- Test the command to get users from the `Domain Users` group in the domain. The following assumes the details of users in the `Domain Users` group in the *EXAMPLE.COM* domain are being retrieved:

  ```
  sudo getent group "EXAMPLE.COM\Domain Users"
  ```

  ```
  EXAMPLE.COM\domain users:x:10000:exampleuser1,exampleuser2
  ```

- Confirm that you can use domain users and groups when using file and directory commands. For example, to set the owner of the */srv/samba/shareexample/* directory to `EXAMPLE.COM\administrator` and the group to `EXAMPLE.COM\Domain Users` run the following command:

  ```
  sudo chown "EXAMPLE.COM\administrator":"EXAMPLE.COM\Domain Users" /srv/
  samba/shareexample
  ```

5 Accessing Samba Shares

The following tasks describe how to access Samba shares from Oracle Linux and Windows clients.

Accessing Samba Shares From a Windows Client

To access a share on a Samba server from Windows using Windows Explorer, enter the host name of the Samba server and the share name using the following format:

```
\\server_name\share_name
```

If you enter `\\server_name`, Windows displays the directories and printers that the server is sharing. You can also use the same syntax to map a network drive to a share name.

Accessing Samba Shares From an Oracle Linux Client

To access a Samba share from an Oracle Linux host, install the following packages:

- `samba-client`

  Installing the `samba-client` package gives you the `smbclient` utility that provides SFTP-like commands to access Samba shares. For example, `smbclient` provides a `get` command for downloading a file from a remote Samba share, and a `put` command for uploading a file.

- `cifs-utils`

  Installing the `cifs-utils` package lets you mount a Samba share.

Using `smbclient` Commands

The following steps give a brief overview of how you might use the `smbclient` commands:

1. Log onto a share *example_samba_share* hosted on server *example_samba_server* using account *EXAMPLE.COM/user1*:

   ```
   sudo smbclient -U "EXAMPLE.COM\user1" //example_samba_server/
   example_samba_share
   ```

2. Change to directory location */directory1/* :

   ```
   smb: \> cd /directory1/
   ```

3. Download file *ExampleFile.txt*:

   ```
   smb: \directory1\> get ExampleFile.txt
   ```

4. End the session:

   ```
   smb: \directory1\> exit
   ```

For more information, see the `smbclient(1)` manual page.

Mounting a Samba Share

The `cifs-utils` package provides tools for mounting Samba shares using CIFS and the SMBv3 protocols.

> **Note:**
>
> We recommend using SMBv3 instead of CIFS for improved performance and security.

The following steps describe how to mount a Samba share.

1. Create a directory to mount the share.

   Create an empty directory in which to mount the share. For example:

   ```
   sudo mkdir /mnt/smb-share
   ```

2. Create a credentials file.

   The credentials file lets the user access the share without being prompted for a password. The name of the file is unimportant, but it's a good idea to prefix it by a period to hide it from general view (for example, `.credentials`). The file must contain the following information:

   ```
   username=username
   password=password
   domain=EXAMPLE.COM
   ```

   > **Note:**
   >
   > The password is the password created for this user on the Samba server with the `smbpasswd` utility, as described in Configuring a Samba Standalone Server.

3. Change the ownership of the credentials file.

   Set the ownership of the credentials file to the current user using the `chown` command. The syntax is as follows:

   ```
   sudo chown username[:groupname] credfile
   ```

4. Change the permissions of the credentials file.

Because the credentials file contains a plain-text password, ensure that only the owner has read access. For example, to set the appropriate permissions for the current user, run the following command:

```
sudo chmod 400 credfile
```

5. Mount the share.

   Mount the share using the `mount` command. The syntax of the `mount` command for mounting a Samba share using the SMBv3 protocol is as follows:

   ```
   sudo mount -t smb3 //server_name/share_name mountpoint -o
   credentials=credfile
   ```

   For example, the following command mounts the `smb_share` share from the `smb_server` Samba server to the local machine's `/mnt/smb-share` directory, using the SMB protocol and the credentials stored in the `.credentials` file in the user's home directory:

   ```
   sudo mount -t smb3 //smb_server/smb_share /mnt/smb-share -o
   credentials=~/.credentials
   ```

   You can use a specific version of the SMB protocol with the `vers` option. For example, to use SMB 3.0:

   ```
   sudo mount -t smb3 //smb_server/smb_share /mnt/smb-share -o
   credentials=~/.credentials,vers=3.0
   ```

   Note:

   If the Samba server is a domain member server in an AD domain, and the current session was authenticated by the Kerberos server in the domain, you can use the existing session credentials by specifying the `sec=krb5` option instead of a credentials file. For example:

   ```
   sudo mount -t smb3 //smb_server/smb_share /mnt/smb-share -o
   sec=krb5
   ```

   For more information on using the `mount` command to mount a Samba share, see the `mount.smb3(8)` manual page.

6. (Optional) Permanently mount the Samba share.

   To configure the system to mount a Samba share at boot time, add an entry for the share to the `/etc/fstab` file using the following syntax:

   ```
   //server_name/share_name mountpoint smb3 options
   ```

For example, to mount the `smb_share` shared directory from the `smb_server` server to `/mnt/smb-share`, using the */home/user/.creds* credentials file, add the following line to `/etc/fstab`:

```
//smb_server/smb_share /mnt/smb-share smb3 credentials=/home/user/.creds
```