Install a full samba share on Debian.

A ready-to-run Bash script for Debian 12 that installs Samba, configures a standalone file server with one public (guest) and one private (authenticated) share, enables the required services, validates the config, and opens the UFW firewall rule if UFW is active. This follows Debian's simple server guidance, includes sample smb.conf best practices, and uses testparm to verify the configuration before starting services.

What it does

- Installs Samba packages and utilities (samba, samba-common-bin, smbclient) needed for a file server on Debian 12 and checks smbd status to confirm installation.
- Enables and starts the smbd and nmbd services for SMB file sharing and NetBIOS name service on a standalone server, consistent with Debian's simple server flow.
- Backs up the default smb.conf, writes a hardened standalone configuration (workgroup, server role, disable printing, guest mapping, usershare), and sets a minimum SMB protocol of SMB2 as documented in smb.conf references.
- Creates a public share at /srv/samba/public (guest read/write) and a private share at /srv/samba/private (group-restricted), with POSIX permissions, SGID for group inheritance, and ACL-friendly VFS options.
- Ensures the sambashare group exists, adds an optional UNIX user to that group, and registers the account in Samba's database via smbpasswd -a for authenticated access.
- Validates the configuration with testparm and restarts services to apply changes, following common Samba setup checks.
- If UFW is installed and active, opens the "Samba" application profile (ports 137–139, 445), matching typical Debian/Ubuntu firewall examples for Samba.

Script

```
#!/usr/bin/env bash
# Full Samba file server setup for Debian 12 (Bookworm)
# - Creates public (guest) and private (authenticated) shares
# - Idempotent where possible
# - Run as root

set -euo pipefail

# Configurable defaults (override via environment)
WORKGROUP="${WORKGROUP:-WORKGROUP}"
```

```
SAMBA_GROUP="${SAMBA_GROUP:-sambashare}"
PUBLIC_DIR="${PUBLIC_DIR:-/srv/samba/public}"
PRIVATE_DIR="${PRIVATE_DIR:-/srv/samba/private}"
BACKUP_SUFFIX="$(date +%Y%m%d-%H%M%S)"

require_root() {
  if [ "${EUID:-$(id -u)}" -ne 0 ]; then
    echo "This script must be run as root."
    exit 1
  fi
}

install_packages() {
  export DEBIAN_FRONTEND=noninteractive
  apt-get update -y
  apt-get install -y samba samba-common-bin smbclient
}

enable_services() {
  systemctl enable --now smbd
  # nmbd provides NetBIOS name service/browsing; useful on many networks
  systemctl enable --now nmbd || true
  # winbind is optional for name resolution; enable but ignore errors if absent
  systemctl enable --now winbind || true
}

prepare_directories() {
  mkdir -p "$PUBLIC_DIR" "$PRIVATE_DIR"
  # Ensure group exists
  if ! getent group "$SAMBA_GROUP" >/dev/null; then
    groupadd -f "$SAMBA_GROUP"
  fi

  # Public: group-writable, guest access, inherit group via SGID
  chown root:"$SAMBA_GROUP" "$PUBLIC_DIR"
  chmod 2775 "$PUBLIC_DIR"

  # Private: group-restricted, authenticated users only, inherit group via SGID
  chown root:"$SAMBA_GROUP" "$PRIVATE_DIR"
  chmod 2770 "$PRIVATE_DIR"
```

```
}

write_smb_conf() {
  local conf="/etc/samba/smb.conf"
  if [ -f "$conf" ]; then
    cp -a "$conf" "${conf}.bak.${BACKUP_SUFFIX}"
  fi

  cat > "$conf" <<'EOF'
# Managed by setup script
# Reference: smb.conf(5)
[global]
   workgroup = WORKGROUP
   server role = standalone server
   netbios name = __NETBIOS_NAME__
   server string = Samba Server on %h
   map to guest = Bad User
   log file = /var/log/samba/log.%m
   max log size = 1000
   dns proxy = no

   # Usershares (optional; enables desktop-created shares if needed)
   usershare allow guests = yes

   # Disable printing support on a fileserver
   load printers = no
   printing = bsd
   printcap name = /dev/null
   disable spoolss = yes

   # Security hardening
   server min protocol = SMB2

   # Windows ACL compatibility on POSIX filesystems
   vfs objects = acl_xattr
   map acl inherit = yes
   store dos attributes = yes

[public]
   comment = Public Share (guest RW)
```

```
   path = __PUBLIC_PATH__
   browseable = yes
   read only = no
   guest ok = yes
   guest only = yes
   force user = nobody
   force group = __SAMBA_GROUP__
   create mask = 0664
   directory mask = 2775

[private]
   comment = Private Share (authenticated)
   path = __PRIVATE_PATH__
   browseable = yes
   read only = no
   valid users = @__SAMBA_GROUP__
   force group = __SAMBA_GROUP__
   create mask = 0660
   directory mask = 2770
   inherit permissions = yes
EOF

  # Substitute variables
  sed -i "s|WORKGROUP|${WORKGROUP}|g" "$conf"
  sed -i "s|__NETBIOS_NAME__|$(hostname -s)|g" "$conf"
  sed -i "s|__PUBLIC_PATH__|${PUBLIC_DIR}|g" "$conf"
  sed -i "s|__PRIVATE_PATH__|${PRIVATE_DIR}|g" "$conf"
  sed -i "s|__SAMBA_GROUP__|${SAMBA_GROUP}|g" "$conf"
}

add_optional_user() {
  echo
  read -r -p "Enter a UNIX username to grant private share access (leave blank to skip): "
SAMBA_USER || true
  if [ -n "${SAMBA_USER:-}" ]; then
    if ! id "$SAMBA_USER" >/dev/null 2>&1; then
      # Create a local UNIX account without setting a system password
      adduser --disabled-password --gecos "" "$SAMBA_USER"
    fi
    usermod -aG "$SAMBA_GROUP" "$SAMBA_USER"
```

```
    echo "Set a Samba password for ${SAMBA_USER}:"
    smbpasswd -a "$SAMBA_USER"
  fi
}

check_and_restart() {
  echo "Validating Samba configuration with testparm..."
  testparm -s >/dev/null
  systemctl restart smbd
  systemctl restart nmbd || true
  systemctl restart winbind || true
}

maybe_open_firewall() {
  if command -v ufw >/dev/null 2>&1; then
    if ufw status | grep -qi "Status: active"; then
      ufw allow Samba || true
    fi
  fi
}

show_summary() {
  cat <<SUMMARY

Samba is installed and configured.

Shares:
  - \\\\$(hostname -s)\\public  (guest RW)
  - \\\\$(hostname -s)\\private (authenticated; users in group '${SAMBA_GROUP}')

Private-share access:
  - Add users with: usermod -aG ${SAMBA_GROUP} <user> && smbpasswd -a <user>

Validate:
  - testparm -s
  - systemctl status smbd

SUMMARY
}
```

```
main() {
  require_root
  install_packages
  enable_services
  prepare_directories
  write_smb_conf
  add_optional_user
  check_and_restart
  maybe_open_firewall
  show_summary
}

main "$@"
```

How to use

- Save the script as samba-setup.sh, mark it executable with chmod +x samba-setup.sh, and run it as root on a fresh Debian 12 system.
- During the run, optionally specify a UNIX account to grant private share access; the script adds it to the Samba group and registers it with smbpasswd -a for authentication.
- Access from Linux with smbclient, for example: smbclient //HOSTNAME/private -U USER, or test locally per Debian’s simple server guidance using smbclient commands.

Notes

- The services to manage on a standalone server are smbd (and optionally nmbd/winbind), and systemctl status smbd confirms the server daemon status on Debian 12.
- The configuration file is /etc/samba/smb.conf; test changes with testparm and consult the smb.conf(5) reference for parameters like workgroup, server role, and server min protocol.
- If UFW is enabled, the script opens the Samba profile; alternatively, iptables/nftables rules covering UDP 137–138 and TCP 139/445 can be used as demonstrated in Debian’s simple server page.

Ins0mniA

Revision #2
Created 2025-09-04 03:51:52 EEST by Green
Updated 2025-09-10 19:45:19 EEST by Green