# *Ubuntu File Server with simple Remote Power Control (V1.0)*

The intended purpose of this unholy amalgamation of violent disregard for best practices is to make a simple local file share folder, removing the constant USB chasing with easy-to-use start and stop functions to avoid unnecessary power usage when not in use – for people for whom just want two easy buttons to press, and security is tertiary.

This project had two key goals it had to reach; it had to be simple enough for me to execute without much prior knowledge and even simpler for the end user to use. As I am writing this, I believe I have managed to do both. This guide is mainly made as practice and is not necessarily meant to be followed or installed by anyone.

# Assumptions & Prerequisites:

- *You have a server computer with Ubuntu Server (LTS version like 22.04 recommended) already installed.*
- *You have administrator access (a user account with sudo privileges) on the Ubuntu server.*
- *You have a Windows client PC on the same local network as the server.*
- *You have network infrastructure (router, switches) connecting the devices.*
- *You can download/transfer files to the Windows client PC.*

# Dependencies:

**On Server:**

- *Ubuntu Server LTS*
- *openssh-server (usually installed by default),*
- *samba*
- *avahi-daemon (recommended).*

**On Windows Client:**

- *PuTTY tools: puttygen.exe, plink.exe, pageant.exe.*
  *Use the included files or visit: Download PuTTY: latest release (0.83)*
- *WakeMeOnLan.exe or similar WoL utility.*
  *Use included or Download from NirSoft or trusted source.*
- *Apple Bonjour Print Services for Windows (recommended for .local hostname resolution). Download from Apple Support.*

# Security Concerns

**WARNING:** This guide intentionally prioritizes simplicity and ease of use over standard security practices. It is designed for use on a trusted local network where security is a secondary concern. **Do NOT use this setup in untrusted environments or expose the server directly to the internet.**

**By following this guide, you accept the following known security risks:**

- ***Passwordless Remote Shutdown:*** *The remote shutdown method uses an SSH key without a passphrase and configures sudo to allow the poweroff command without requiring a password. This means:*
- *Anyone gaining access to the specific private key file (.ppk) on your client PC can shut down the server.*
- *If the client PC itself is compromised, the server could be shut down maliciously.*

***Mitigation:*** *Protect the .ppk file diligently on your client computer.*

- ***Unencrypted SSH Key:*** *The SSH private key (.ppk file) is generated without a passphrase for automation convenience. Standard security practices recommend using a passphrase to protect the key file itself, which this guide skips.*
- ***Basic Service Configuration:*** *This guide implements only basic configurations for SSH and Samba (file sharing). It does* ***not*** *include common security hardening steps such as:*
- *Configuring a host firewall (like ufw) on the server.*
- *Advanced SSH server security tuning.*
- *Fine-grained Samba share permissions or security options beyond the essentials for basic access control.*

***Assumption of Trusted Network:*** *The entire setup assumes your local network (behind your router) is a trusted environment.*

# Placeholders Used in this Guide:

- **<YOUR_ADMIN_USER>**
  *The admin username on the Ubuntu server*

- **<YOUR_SERVER_HOSTNAME>**
  *The desired hostname for the server*

- **<FINAL_SERVER_IP_OR_HOSTNAME>**
  *The stable way to reach the server on the final network*
  **Strongly recommended to use <YOUR_SERVER_HOSTNAME>.local**
  *after completing Section B.*

- **<YOUR_SERVER_MAC>**
  *The MAC address of the server's wired Ethernet port*

- **<INTERFACE_NAME>**
  *The server's network interface name*

- **<YOUR_SAMBA_USER>**
  *Username for accessing the file share*

- **<YOUR_SAMBA_SHARE_NAME>**
  *The name of the network share*

- **<YOUR_SHARE_PATH>**
  *The path on the server where shared files are stored*

- **<YOUR_SHARE_GROUP>**
  *A Linux group to manage share permissions*

- **<CLIENT_CONTROL_FOLDER_PATH>**
  *The full path on the Windows client where control files are stored*
  *Example: (C:\ServerControl) Highly recommended to use this path*

- **<FINAL_SERVER_IP>**
  *The static or reserved DHCP IP address for the server*

- **<SUBNET_MASK_CIDR>**
  Subnet mask in CIDR notation
  *Example: (24 for 255.255.255.0)*

- **<GATEWAY_IP>**
  *The router's IP address for the server's subnet.*

- **<DNS_IP_1>, <DNS_IP_2>**
  *IP addresses for DNS servers*

# Guide Sections:

## A) Initial Server Setup & Checks

*Ensure basic admin access and find network info*

### A.1: Verify or create user

*Verify you have a user account with sudo rights. If not, create one and replace your admin user*
*Use:* **sudo adduser** <YOUR_ADMIN_USER> *to create new user, follow prompts and set a strong password*
*Then:* **sudo usermod -aG sudo** <YOUR_ADMIN_USER>
*Log in as this user for subsequent steps*

### A.2: SSH Server

*Ensure SSH server is installed and running*
*Then do a:* **sudo apt update**
*Install with:* **sudo apt install openssh-server**
*Verify your system with:* **sudo systemctl status sshd**
*Which should be active and running*

### A.3: Find Network Info

*Log into the server via console or initial SSH using DHCP IP*
*Find interface name:* **ip a** *and note down* <INTERFACE_NAME>
*Find Netplan File:* **ls /etc/netplan/** *and note down* .yaml *filename*
*Find MAC Address:* **ip link show** <INTERFACE_NAME> Note down <YOUR_SERVER_MAC>

## B) Network Configuration (Server & Client)

*Stable server address/name & client discovery setup*
*Skip this section if your server already has a stable IP/hostname setup you prefer.*

### B.1: Configure Stable Server IP - Choose ONE method

#### B.1.a: DHCP Reservation – Recommended

*Use router interface to reserve* <FINAL_SERVER_IP> *for* <YOUR_SERVER_MAC>
*Reboot server using:* **sudo reboot** *and verify*

*After the server reboots, SSH back in using the* <FINAL_SERVER_IP> *you reserved*
*Then run* **ip a** *to confirm the network interface shows that specific IP address*

#### B.1.b: Static IP - Fallback

*Edit netplan file:* **sudo nano /etc/netplan/**FILE.yaml *to set static* <FINAL_SERVER_IP>, <SUBNET_MASK_CIDR>, <GATEWAY_IP>, <DNS_IP_1>, <DNS_IP_2> *for your* <INTERFACE_NAME>.
*Apply using:* **sudo netplan apply***, reconnect via new IP, verify by entering:* **ip a**

### B.2: Install Avahi on the server – Recommended

**sudo apt update && sudo apt install avahi-daemon**
*Allows discovery via <YOUR_SERVER_HOSTNAME>.local*

### B.3: Install Bonjour on client PC - Recommended

Install "**Bonjour Print Services for Windows**" from Apple and reboot if needed

### B.4: Define Connection Method

*For subsequent steps,* <FINAL_SERVER_IP_OR_HOSTNAME> *should be* <YOUR_SERVER_HOSTNAME>.local
*Use* <FINAL_SERVER_IP> *only if* .local *fails testing later.*

## C) Remote Shutdown Setup

*Allow client to shut down server securely without password using SSH keys*
*Skip if you only need Wake-on-LAN or have another shutdown method.*

### C.1: Configure Passwordless Sudo on Server

*Allow* <YOUR_ADMIN_USER> *to run poweroff without password.*

```
sudo nano /etc/sudoers.d/010_adminuser-nopasswd-poweroff
```

*Add line:* <YOUR_ADMIN_USER> **ALL=(ALL) NOPASSWD: /sbin/poweroff**

*Save and exit.*

Set permissions: **sudo chmod 440 /etc/sudoers.d/010_adminuser-nopasswd-poweroff**
Check syntax: **sudo visudo -c**

### C.2: Generate SSH Key on Client PC

*Use puttygen.exe. Click Generate - Leave passphrase blank*
*recommended for simplicity according to the project's needs - protect the file well*

*Save* ***public key*** *text to*
<CLIENT_CONTROL_FOLDER_PATH>\Files\<YOUR_ADMIN_USER>_shutdown_key.pub
*Save* ***private key*** *as*
<CLIENT_CONTROL_FOLDER_PATH>\Files\<YOUR_ADMIN_USER>_shutdown_key.ppk

### C.3: Install Public Key

*SSH into server as* <YOUR_ADMIN_USER>*. And run:*

```
mkdir -p ~/.ssh && chmod 700 ~/.ssh
nano ~ /.ssh/authorized_keys
```

*Paste public key text from* .pub *file,* Save/Exit

```
chmod 600 ~/.ssh/authorized_keys
```

*This sets file permissions so ONLY the file owner -* <YOUR_ADMIN_USER> *- can read and write the file. SSH requires this for security; it won't use key files accessible by others*

### C.4: Edit Stop_Server.bat on your PC

Navigate to <CLIENT_CONTROL_FOLDER_PATH>\Files\ and open Stop_Server.bat in a text editor

***Note:*** *the following string needs to be on one line in the file but for obvious reasons it is split up here*

```
<CLIENT_CONTROL_FOLDER_PATH>\Files\plink.exe -i
<CLIENT_CONTROL_FOLDER_PATH>\Files\<YOUR_ADMIN_USER>_shutdown_key.ppk -batch
<YOUR_ADMIN_USER>@ <YOUR_SERVER_HOSTNAME>.local sudo poweroff
```

*Save and exit*

***Note:*** *Using* <YOUR_SERVER_HOSTNAME>.local *is strongly recommended as it makes the script work even if the server's IP address changes (requires Avahi/Bonjour setup from Section B).*

*If .local resolution fails on your specific network, you must replace <YOUR_SERVER_HOSTNAME>.local in the command above with the specific static or reserved* <FINAL_SERVER_IP> *configured in Step B.1.*

*Example:*

```
echo Sending Shutdown Signal to localdropbox.local...
C:\ServerControl\Files\plink.exe -i C:\ServerControl\Files\serveradmin_shutdown_key.ppk -batch
serveradmin@localdropbox.local sudo poweroff
echo Signal Sent. Please wait for server to power down.
pause
```

**Note:** .local should be replaced with (192.168.1.xxx) if static IP route was chosen in Step B.1

## D) Remote Startup Setup - Wake-on-LAN (Wol)

*Allow client to wake server from sleep/power-off - Skip if you only need remote shutdown.*

### D.1: Enable WoL in server BIOS

*Reboot server, enter BIOS/UEFI setup by pressing bios key on boot screen (Usually F2, DEL)*
*“Find Power Management” or “Network settings”*
*Enable "Wake on LAN" / "Power On by PCIe/PCI", or similar option*
*Save & Exit BIOS*

### D.2: Edit Start_Server.bat on your PC

*In* <CLIENT_CONTROL_FOLDER_PATH>\Files\Start_Server.bat, *ensure WakeMeOnLan command is:*
batch <CLIENT_CONTROL_FOLDER_PATH>\Files\WakeMeOnLan.exe /wakeup <YOUR_SERVER_MAC>
Use the correct MAC address found in A.3 Save and exit

*From .bat file – less cluttered example:*

```
echo Sending Wake-on-LAN signal to localdropbox (c8:0a:a9:1f:xx:xx)...
C:\ServerControl\Files\WakeMeOnLan.exe /wakeup c8:0a:a9:1f:xx:xx
echo Signal sent.
pause
```

## E) File Share Setup with Samba

*Create a network share accessible by a specific user - Skip if you don't need a file share or already have one.*

### E.1: Install Samba on Server

```
sudo apt update && sudo apt install samba
```

### E.2: Create Share Directory

```
sudo mkdir -p <YOUR_SHARE_PATH>
```

*Creates an directory <where you assign it>*

### E.3: Create Share Group & User

```
sudo groupadd <YOUR_SHARE_GROUP>
```

*Creates a new Linux group used to manage share permissions*

```
sudo adduser --system --no-create-home --ingroup <YOUR_SHARE_GROUP> --disabled-login --shell /sbin/nologin <YOUR_SAMBA_USER>
```

*Creates user without login shell/home*

### E.4: Set Directory Permissions

```
sudo chown root:<YOUR_SHARE_GROUP> <YOUR_SHARE_PATH>
sudo chmod 1770 <YOUR_SHARE_PATH>
```

*Allows group members to add/delete their own files*

### E.5.a: Enter Samba Share Configuration

```
sudo nano /etc/samba/smb.conf
```

*Allows you to enter nano and edit samba configurations*

### E.5.b: Configure Samba Share

*Add the following section at the very end of the ` /etc/samba/smb.conf` file*
*Replace the bracketed placeholders (` <...>` ) with your chosen names/paths:*

```
 [<YOUR_SAMBA_SHARE_NAME>]
comment = Shared Files
path = <YOUR_SHARE_PATH>
browseable = yes
read only = no
guest ok = no
valid users = @<YOUR_SHARE_GROUP>
create mask = 0660
directory mask = 0770
force group = <YOUR_SHARE_GROUP>
```

*Save the file and exit nano (Ctrl+O, Enter, Ctrl+X).*

*Check the configuration syntax for errors by running:* **testparm**

### E.6. Restart Samba

*Run:* **sudo systemctl restart smbd nmbd**
Restarts samba with new configuration

### E.7. Set Samba Password

*Run:* **sudo smbpasswd -a** <YOUR_SAMBA_USER>
*Set password for your samba user*

## F) Client Shortcut Setup - Optional

*Easy access to scripts on Windows - Skip if you prefer running .bat files directly.*

### F.1. Prepare Client Folder

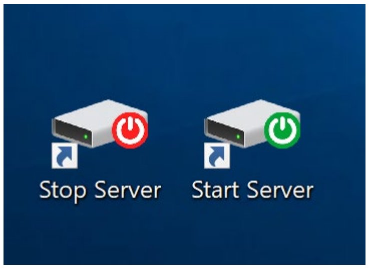
*Ensure* **Start_Server.bat**, **Stop_Server.bat**, *required .exe files, keys, and Icons folder are organised under* <CLIENT_CONTROL_FOLDER_PATH>.

### F.2 Version A: Create Shortcuts

In a place of your choice, create a shortcut to execute the bat files for “start_server” and “stop_server” these will be your on/off buttons for your server. In this example we will create them on the desktop.

**Create “Server On” shortcut**

Right-click on your Desktop -> Create shortcut
Click “Browse” and chose: *<CLIENT_CONTROL_FOLDER_PATH>\Files\Start_Server.bat*
*Example: C:\ServerControl\Files\Start_Server.bat*
*Name the shortcut Start Server*
*Right-click 'Server On' shortcut -> properties -> shortcut*
*In Shortcut click “Change icon” -> Browse <CLIENT_CONTROL_FOLDER_PATH> \Icons*
*Choose “Server on.ico” -> click open -> ok*
Click “Apply”



**Create “Server Off” shortcut**

Right-click on your Desktop -> Create shortcut
Click “Browse” and chose: *<CLIENT_CONTROL_FOLDER_PATH>\Files\Stop_Server.bat*
*Example: C:\ServerControl\Files\Stop_Server.bat*
*Name the shortcut Stop Server*
*Right-click 'Server On' shortcut -> properties -> shortcut*
*In Shortcut click “Change icon” -> Browse <CLIENT_CONTROL_FOLDER_PATH> \Icons*
*Choose “Server off.ico” -> click open -> ok*
Click “Apply”

### F.2 Version B: Create & Pin Taskbar Shortcuts

*This creates windows work-around allows special shortcuts designed for the taskbar*

**Create the "Start Server" taskbar shortcut:**

*1. Navigate to <CLIENT_CONTROL_FOLDER_PATH>\Icons   Create new shortcut*

*2. Copy and paste the line below into "Type the location of the item"*

```
cmd.exe /c "<CLIENT_CONTROL_FOLDER_PATH>\Files\Start_Server.bat"
```

*3. Click Next -> Name the shortcut: "Taskbar Start Server" -> Click Finish.*

*4. Right-click this new shortcut -> Properties -> Change Icon -> Click Browse*

*5. If made in the Icon folder you will see the "Server On" .ico file use this file -> Click Ok -> Click Apply*

**Path to icon:** *<CLIENT_CONTROL_FOLDER_PATH>\Icons\ and select "Server On" icon (.ico file).*

*7. Right-click on "Taskbar Start Server" -> pin to taskbar*

**Create the "Stop Server" taskbar shortcut:**

*1. Navigate to <CLIENT_CONTROL_FOLDER_PATH>\Icons   Create new shortcut*

*2. Copy and paste the line below into "Type the location of the item"*

```
cmd.exe /c "<CLIENT_CONTROL_FOLDER_PATH>\Files\Stop_Server.bat"
```

*3. Click Next -> Name the shortcut: "Taskbar Stop Server" -> Click Finish.*

*4. Right-click this new shortcut -> Properties -> Change Icon -> Click Browse*

*5. If made in the Icon folder you will see the "Server Off" .ico file use this file -> Click Ok -> Click Apply*

**Path to icon:** *<CLIENT_CONTROL_FOLDER_PATH>\Icons\ and select "Server Off" icon (.ico file).*

*7. Right-click on "Taskbar Stop Server" -> pin to taskbar*

## G) Testing

### G.1. Test Power Off

*Use "Server Off" shortcut/pin. Verify server shuts down via SSH key.*

### G.2. Test Power On

*Use "Server On" shortcut/pin. Verify server boots via WoL (wait boot time).*

### G.3. Test File Share

*Map network drive on client* using \\<YOUR_SERVER_HOSTNAME>.local\<YOUR_SAMBA_SHARE_NAME>
*If that fails, try using* \\<FINAL_SERVER_IP>\<YOUR_SAMBA_SHARE_NAME>
*Use credentials* <YOUR_SAMBA_USER> *and the password set in E.7.*
*Verify you can read/write files according to permissions.*

The foundation laid by all the software engineers and coders that made this project possible for me to execute, I thank you.