

Linux Server

- [Debian-Based Server Setup](#)
- [OliveTin](#)
- [Samba\(SMB\) Setup](#)

Debian-Based Server Setup

Debian 12 with Desktop

Run update and upgrade distro first. Install NTP package if there are errors with that. Reboot

Setup powertop and powersaving features

```
sudo apt install powertop
powertop --auto-tune
```

Powersave governor and at reboot. Remember to run the command again

```
@reboot echo "powersave" | tee /sys/devices/system/cpu/cpu*/cpufreq/scaling_governor
>/dev/null 2>&1
```

Ensure these packages are installed

```
powertop htop iotop fio curl gnupg wget ntfs-3g neofetch ca-certificates lsb-release hdparm
hd-idle openssh-server
```

HDD

`lsblk` and `blkid` to get the ntfs hard drive /dev name and the /dev/by-uuid/...

Edit the fstab to mount the drive, same entry for nvme drive

```
UUID=CC34294F34293E38 /mnt/data ntfs-3g 0 0
```

If the mounted device is HDD array, need to spindown disk with hdparm

```
hdparm -B 120 /dev/sdb # set the APM level
hdparm -S 241 /dev/sdb
```

For the -S spindown, 0-240 is multiple of 5s, 241-255 is multiple of 30 min. The above command set spindown every 30min.

If hdparm does not work, hd-idle can be used. Edit the file in `/etc/defaults/hd-idle`

```
-i 60 -a disk/by-uuid/xxx -l /var/log/hd-idle.log
```

Sudo without password, go to visudo and add the lines to the bottom, replace \$USER with the actual username.

```
$USER ALL=( ALL) NOPASSWD: ALL
```

Edit shortcuts in bashrc

```
source .bashrc
```

OpenSSH with Keys

Generate the key using the terminal

```
ssh-keygen
```

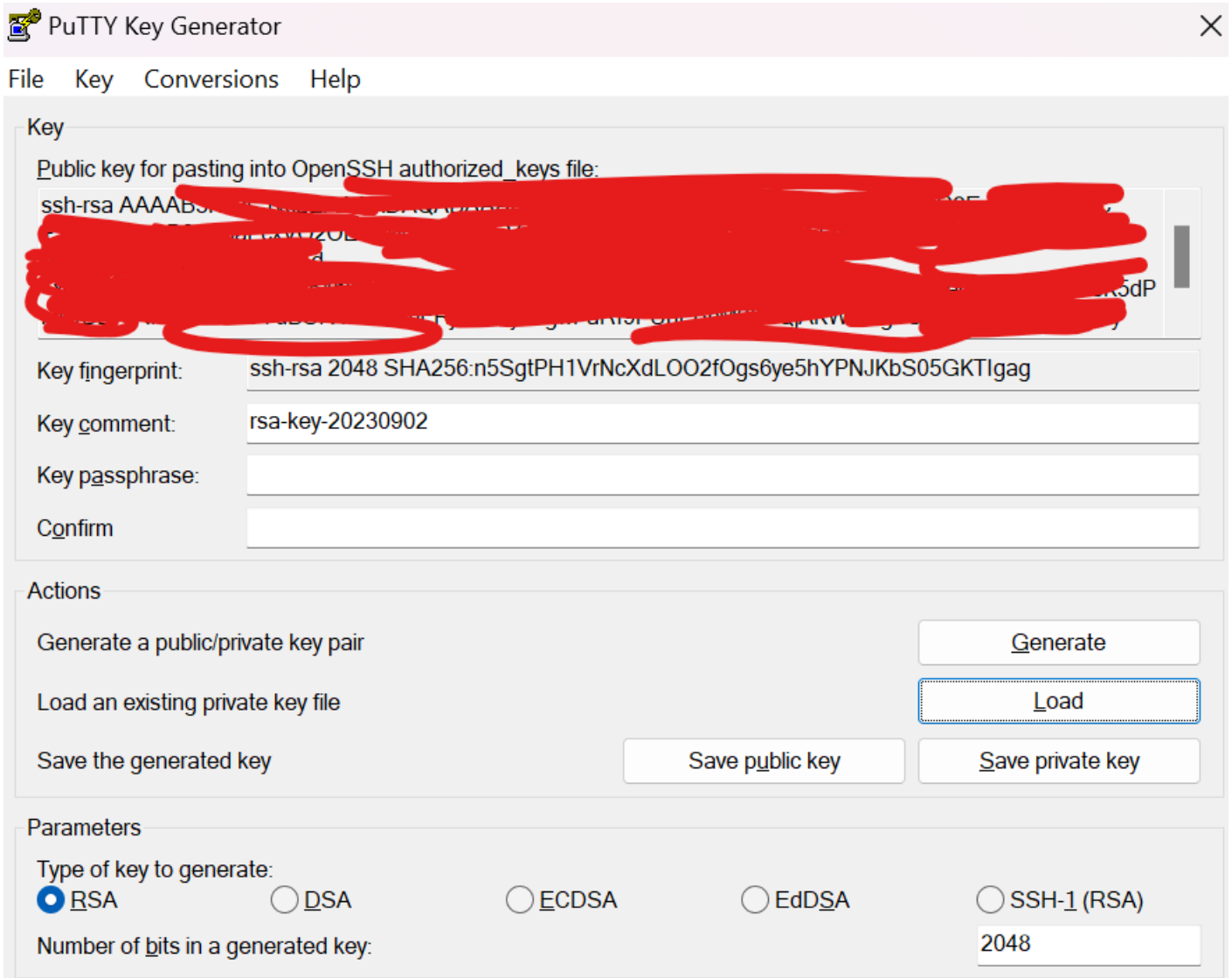
- give a location to put the key pair
- this generate a public (.pub) and private key pair

```
ssh-copy-id -i key.pub username@server
```

- `key.pub` is the public key that was generated

The key is ready to use for authorization.

Generate keys using PuTTY software



1. Copy the red part and use nano to add it in the server `~/ .ssh/authorized_keys`
2. Make sure permissions are correct

```
3. mkdir -p ~/.ssh
   chmod 700 ~/.ssh
   chmod 600 ~/.ssh/authorized_keys
   nano ~/.ssh/authorized_keys
```

4. Save private key as ppk file on the root ssh folder.
5. If the client with private key is Linux machine, need to change the permission of the private key.

```
chmod 600 private.key
```

6. Convert the private key Conversion > Export OpenSSH Keys and save the file to a folder OpenSSH Keys

Setting Up SMB

Refer to [Samba\(SMB\) Setup](#) to setup SMB server.

Desktop Environment Setup

Firefox

The location of firefox profile is at `/home/$USER/.mozilla/firefox/xxxxxx.default`

Make a tarball and copy it and extract it in destination.

In the profile folder, look for `compatibility.ini`, go to a random profile in the dest machine and copy the `compatibility.ini` settings to the one that is copied over. This ensure compatibility so that the new profile works without warning.

Check the `profile.ini` with the name and the location of the new profile folder, firefox should be the same as before.

```
[Profile0]
Name=karis
IsRelative=1
Path=ims58kbd.default-esr-1
```

Themes

To backup/restore settings of cinnamon

Icons

The icons are located at these locations.

```
/usr/share/icons
~/.icons
```

Scripts

Copy the scripts and put it into `~/script` for organization and copy the old crontab for executing these scripts.

OliveTin

OliveTin exposes a webpage with buttons that execute shell command (eg. docker, scripts) on the server and allow others for easy access. It should be used internally only.

Installation

Download the correct file from this site. <https://github.com/OliveTin/OliveTin/releases>
OliveTin_linux_amd64.deb

Go to the directory and install the package.

```
sudo dpkg -i OliveTin_*.deb  
sudo systemctl enable --now OliveTin
```

Configuration

The configuration file is located at `/etc/OliveTin/config.yaml`

Example Configuration

```
listenAddressSingleHTTPFrontend: 0.0.0.0:1378 # set the port of OliveTin to 1378  
  
# Choose from INFO (default), WARN and DEBUG  
logLevel: "INFO"  
  
# Actions (buttons) to show up on the WebUI:  
actions:  
  # This will run a simple script that you create.  
  - title: Update Music  
    shell: /home/karis/scripts/script  
    icon: '&#127925'
```

More possible configurations (many are not possible on Docker)

Execute a shell command with textbox input.

```
- title: Restart a Docker CT  
  icon: '<img src = "icons/restart.png" width="48px" />'
```

```
shell: docker restart {{ container }}
arguments:
  - name: container
    type: ascii
```

- use {{ }} and give a variable
- under arguments type, assign a type for it, ascii only allows letters and numbers

Execute a shell command with choices

```
- title: Manage Docker Stack Services
  icon: "🛠"
  shell: docker-compose -f /home/karis/docker/bookstack/docker-compose.yml {{ action }}
  arguments:
    - name: action
      choices:
        - title: Start Stack
          value: up -d

        - title: Stop Stack
          value: down
```

This example give choices to start or stop a docker stack of a docker-compose file. If a argument is given the parameter choices, it will be in dropdown mode.

Icons Customization

The icons need to be placed in a folder in `/var/www/[icon-folder]/icon.png`. To use the icons, offline image or web address, it should be in HTML format. The size of 48px is the default size of OliveTin icons. Other CSS options such as `style="background-color: white;"` also works.

```
icon: ' <img src = "icons/minecraft.png" size="48px" />'
```

Icon with emoji, to use emoji, need to use the html code. <https://symbl.cc/en/emoji/>

For example, `☺` 🍷.

```
icon: "☺"
```

Icon Management

The default icon folder is `/var/www/olivetin/icons`

The icon folder of all homelab icons is in `~/icons/homelab`

API

Simple action button.

```
curl -X POST "http://mediaserver:1378/api/StartAction" -d '{"actionName": "Update Music"}'
```

Action with Arguments.

```
curl -X POST 'http://mediaserver:1378/api/StartAction' -d '{"actionName": "Rename Movies",  
"arguments": [{"name": "path", "value": "value"}]}'
```


Samba(SMB) Setup

Setting up SMB Server on Linux

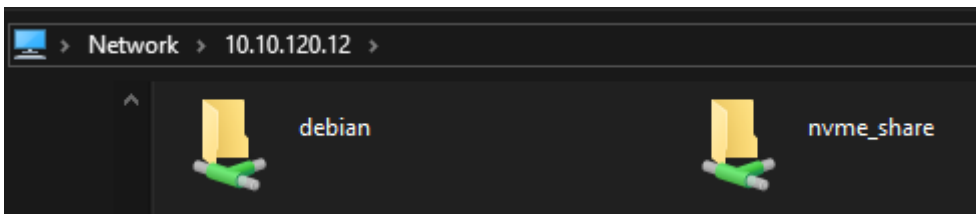
Install the samba tool on Linux.

```
sudo apt update
sudo apt install samba -y
```

Edit the `/etc/samba/smb.conf`

```
[nvme_share]
    comment = NVMe Share
    path = /mnt/nvme/share
    browseable = yes
    read only = no
```

`nvme_share` is the name of the Samba path which will appear in SMB clients and its path is accessed by `\\192.168.0.1\nvme_share`



`path` is the location where the files are stored

`browseable` and `read only` are flags that are needed to make sure read/write access on the SMB share

Lastly, add the user and password for the SMB share

```
sudo smbpasswd -a $USER # enter the password twice
```