

Linux Tips and Tricks

- [Set system time automatically](#)
- [Set correct Timezone](#)
- [Flush DNS Cache Ubuntu](#)
- [Start / Stop /Restart BIND DNS Server](#)
- [Hardening of Linux](#)
- [Tutorial on ufw](#)
- [Fix Error fwupd-refresh](#)
- [Enable ssh login with a public key](#)
- [Mount SAMBA shares](#)
- [Check for open ports](#)
- [Network browsing not working](#)
- [Display IP address on Panel in Xfce](#)
- [Biometrics: Fingerprint](#)
- [Disable SELinux on Fedora](#)
- [Create boot USB](#)
- [Install PVE-VDIClient on Arch Linux](#)
- [Install network scanner on Archlinux](#)
- [Install xrdp](#)
- [Install Cockpit and Firewalld on Debian 12](#)
- [Install xrdp on Fedora 42](#)
- [Install send mail service on Fedora](#)

Set system time automatically

Introduction

It is possible to set and synchronize the time in Linux automatically through the systemd service. It's the successor of NTP daemon. In my network the mt-engine01.simmy.ch provides system time. Hence the device can change, I created an DNS alias ntp.simmy.ch. Using this alias allows changes of the time source without problems.

Ubuntu 22.04 LTS

nano /etc/systemd/timesyncd.conf

```
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation; either version 2.1 of the License, or (at your option)
# any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
# the timesyncd.conf.d/ subdirectory. The latter is generally recommended.
# Defaults can be restored by simply deleting this file and all drop-ins.
#
# See timesyncd.conf(5) for details.

[Time]
NTP=ntp.simmy.ch
FallbackNTP=0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org
#RootDistanceMaxSec=5
```

```
#PollIntervalMinSec=32
#PollIntervalMaxSec=2048
```

```
systemctl restart systemd-timesyncd
timedatectl timesync-status
```

```
Server: 192.168.1.74 (192.168.1.74)
Poll interval: 1min 4s (min: 32s; max 34min 8s)
Leap: normal
Version: 4
Stratum: 3
Reference: 2E8C0F6C
Precision: 1us (-24)
Root distance: 76.324ms (max: 5s)
Offset: +1.117ms
Delay: 326us
Jitter: 0
Packet count: 1
Frequency: -25.696ppm
```

Debian 10

<https://www.digitalocean.com/community/tutorials/how-to-set-up-time-synchronization-on-debian-10>

```
apt purge ntp
apt install systemd-timesyncd
nano /etc/systemd/timesyncd.conf
```

```
# This file is part of systemd.
#
# systemd is free software; you can redistribute it and/or modify it under the
# terms of the GNU Lesser General Public License as published by the Free
# Software Foundation; either version 2.1 of the License, or (at your option)
# any later version.
#
# Entries in this file show the compile time defaults. Local configuration
# should be created by either modifying this file, or by creating "drop-ins" in
```

```
# the timesyncd.conf.d/ subdirectory. The latter is generally recommended.  
# Defaults can be restored by simply deleting this file and all drop-ins.  
#  
# See timesyncd.conf(5) for details.
```

```
[Time]
```

```
NTP=ntp.simmy.ch
```

```
FallbackNTP=0.pool.ntp.org, 1.pool.ntp.org, 2.pool.ntp.org
```

```
#RootDistanceMaxSec=5
```

```
#PollIntervalMinSec=32
```

```
#PollIntervalMaxSec=2048
```

```
systemctl start systemd-timesyncd
```

```
systemctl status systemd-timesyncd
```

```
date
```

Set correct Timezone

Howto set the correct timezone in Linux Ubuntu

Get all possible timezones:

```
timedatectl list-timezones
```

Set the local timezone:

```
timedatectl set-timezone Europe/Zurich
```

Check the local timezone:

```
timedatectl
```

Howto set the correct timezone in Debian 10

```
dpkg-reconfigure tzdata
```

Useful links

<https://linuxize.com/post/how-to-set-or-change-timezone-on-ubuntu-20-04/>

Flush DNS Cache Unbuntu

Introduction

Ubuntu caches DNS queries local. As long as the DNS address of a node does not change, this is very useful. Unless there is a change and the client should react very quickly, you have to flush/delete to cache. This manual describes how to do that.

Method I: Flush the cache

Take a look at the cache:

```
resolvectl statistics
```

Clear the cache:

```
resolvectl flush-caches
```

Method II: Flush the cache

```
systemd-resolve --flush-caches  
systemd-resolve --statistics
```

Method III: Flush the cache

```
killall -USR2 systemd-resolved
```

Start / Stop /Restart BIND DNS Server

Introduction

For testing purposes I am using Univention with bind9. The greater goal is to use AD/SAMBA from Univention. After testing for a couple of weeks suddenly some DNS addresses do not get resolved. The same problems occurred on Zentyal.

So far I couldn't find a reason for this misbehavior. However, a restart of the bind9 service seems to patch the problem.

Debian based Linux

Start the service

```
service bind9 start
```

Stop the service

```
service bind9 stop
```

Restart the service

```
service bind9 restart
```

Reload the service

This will become necessary if a configuration file is changed.

```
service bind9 reload
```

Check status

```
service bind9 status
```

Fedora based Linux

Start the service

```
systemctl start named
```

Stop the service

```
systemctl stop named
```

Restart the service

```
systemctl restart named
```

Check status

```
systemctl status named
```


Hardening of Linux

Introduction

Despite the fact that Linux is Open Source and Linux it comes as a surprise that in the default installation are some hidden trackers and spy software.

Hardening

There is a script that will remove all malware. Originally written for Linux, but it can easily adopted for other distributions.

Ubuntu Secure

This script does:

- System update and software upgrade
- Amazon & advert web apps removing
- AptUrl Removing (tool, which gives possibilities to start installation by clicking on url, can be executed with js, which is not secure)
- Guest session disable for LightDM
- Remote login disable for LightDm
- DNS encryption (dnscrypt-proxy)
I don't recommend this, hence my DNS server is not working with encryption.
apt -y remove dnscrypt-proxy
- FireWall (UFW)
- Antivirus (ClamAV)
- Brute Force protection (Fail2Ban)
- Basic Telemetry Removing (ZeitGeist) and unsecure libs and pre-installed software with high and potential risks

Here is a version for rpm based systems:

```
#!/bin/bash
```

```
# This script removes telemetry and enhances system security on an RPM-based Linux distribution.
```

```
# System Up to Date:
```

```
sudo dnf -y update
```

```
sudo dnf -y upgrade
```

```
# =====
```

```
# Remove any pre-installed telemetry or unwanted software (no direct equivalents for `unity-lens-shopping` and  
`unity-webapps-common` on RPM-based systems):
```

```
# Remove pre-installed software that may be tracking or unwanted:
```

```
sudo dnf -y remove gnome-online-accounts
```

```
sudo dnf -y remove gnome-shell-extension-prefs
```

```
sudo dnf -y remove gnome-software
```

```
# =====
```

```
# Disable Guest session & remote login for LightDM (if LightDM is in use):
```

```
if [ -f /etc/lightdm/lightdm.conf.d/50-no-guest.conf ]; then
```

```
    sudo sh -c 'printf "[Seat:*\n)allow-guest=false\n)greeter-show-remote-login=false\n" >
```

```
/etc/lightdm/lightdm.conf.d/50-no-guest.conf'
```

```
    sudo dnf -y remove lightdm-remote-session-freerdp
```

```
    sudo dnf -y remove lightdm-remote-session-uccsconfigure
```

```
fi
```

```
# =====
```

```
# Remove any equivalent telemetry-related packages:
```

```
# Note: zeitgeist is generally specific to Ubuntu/Debian, so we focus on similar tools on RPM systems.
```

```
# Remove `tracker`, a GNOME-based file indexing and search tool that collects metadata:
```

```
sudo dnf -y remove tracker
```

```
sudo dnf -y remove tracker-miners
```

```
sudo dnf -y remove tracker3
```

```
sudo dnf -y remove tracker3-miners
```

```
# Remove `gnome-usage`, a system resource monitor that could collect usage data:
```

```
sudo dnf -y remove gnome-usage
```

```
# Remove `PackageKit`, which can send data back to package servers:
```

```
sudo dnf -y remove PackageKit
```

```
# =====
```

```
# DNS encryption:
```

```
sudo dnf -y install dnscrypt-proxy
```

```
# =====
```

```
# FireWall (using firewalld):
```

```
sudo dnf -y install firewalld
```

```
sudo systemctl start firewalld
```

```
sudo systemctl enable firewalld
```

```
sudo firewall-cmd --permanent --set-default-zone=block
```

```
sudo firewall-cmd --permanent --add-service=http
```

```
sudo firewall-cmd --permanent --add-service=https
```

```
sudo firewall-cmd --reload
```

```
# =====
```

```
# ClamAV Antivirus Installation:
```

```
sudo dnf -y install clamav
```

```
sudo dnf -y install clamav-daemon
```

```
sudo systemctl enable clamav-daemon
```

```
sudo systemctl start clamav-daemon
```

```
# =====
```

```
# Fail2Ban installation (protects from brute force login):
```

```
sudo dnf -y install fail2ban
```

```
sudo systemctl enable fail2ban
```

```
sudo systemctl start fail2ban
```

```
# =====
```

```
# Remove other potentially problematic or unused packages:
```

```
# Removing `cups` if you don't need printer support:
```

```
# sudo dnf -y remove cups
```

```
# Remove `remmina` if you don't use it for remote connections:
```

```
# sudo dnf -y remove remmina
```

```
# Remove unnecessary GNOME components:
```

```
sudo dnf -y remove evolution
```

```
sudo dnf -y remove evolution-data-server
```

```
sudo dnf -y remove gvfs-fuse
```

```
sudo dnf -y remove vino # VNC server (remote desktop sharing)
```

```
sudo dnf -y remove gnome-shell-extension-background-logo # Fedora logo on desktop background
```

```
sudo dnf -y remove gnome-user-share # Potentially shares user data over the network
```

```
sudo dnf -y remove libreport-plugin-bugzilla # Automatic bug reporting to Bugzilla
```

```
sudo dnf -y remove abrt-addon-xorg # Automatic bug reporting for Xorg
sudo dnf -y remove abrt-cli # Command-line tool for automatic bug reporting
sudo dnf -y remove abrt-addon-ccpp # Automatic bug reporting for C/C++ programs
sudo dnf -y remove abrt-addon-kerneloops # Automatic bug reporting for kernel oopses
sudo dnf -y remove abrt-addon-pstoreoops # Automatic bug reporting for pstore oopses
# =====

# Autoremove unnecessary dependencies:
sudo dnf -y autoremove

# =====

# Troubleshooting:
# If the internet does not work, try restarting dnscrypt-proxy:
# sudo systemctl restart dnscrypt-proxy
# Also, the tool may use another port, detect the port in this output:
# sudo ss -ntulp
# Then add the port to firewall:
# sudo firewall-cmd --permanent --add-port=[portnumber]/tcp
# sudo firewall-cmd --reload
# =====
```

Tutorial on ufw

UFW, or Uncomplicated Firewall, is a simplified firewall management interface that hides the complexity of lower-level packet filtering technologies such as iptables and nftables. If you're looking to get started securing your network, and you're not sure which tool to use, UFW may be the right choice for you.

Here is a link that shows how to set up the firewall on Ubuntu:

[How To Set Up a Firewall with UFW on Ubuntu 22.04](#)

Fix Error fwupd-refresh

Introduction

After installing monitoring (check_mk) I realized that the service fwupd-refresh produces a critical error. However, this is based on a configuration mishap in the service itself. Here is the fix.

The service is able to perform a firmware update on UEFI machines. The service is totally useless on VMs.

Correction Step-by-Step

Edit file `/lib/systemd/system/fwupd-refresh.service`

Replace `SuccessExitStatus=2` with `SuccessExitStatus=1`

Restart the service:

```
systemctl daemon-reload && sudo systemctl start fwupd-refresh.service
```

Check the service

```
systemctl status fwupd-refresh.service
```

Disable the service

Another possibility is to disable the service:

```
systemctl disable fwupd
```

Useful links

<https://askubuntu.com/questions/1404691/fwupd-refresh-service-failed>

<https://askubuntu.com/questions/1227508/consequences-of-disabling-fwupd>

Enable ssh login with a public key



Introduction

it is more secure and easier to login to a server over ssh if you place your public key on this server. This how-to shows in simple steps how to do this.

Generate keys

You only have to do this one time. You can and should reuse your public key for all ssh-servers.

Step 1 - creating SSH key pair

Make sure you are in your home directory.

```
ssh-keygen -t rsa
```

If you want so secure the access to your key with password, enter a password. Otherwise press enter two times.

Step 2 - Copying the SSH public key to the ssh server

The real magic happens here:

```
ssh-copy-id <username>@<ssh-server>
```

Basically this command copies the file `.ssh/id_rsa.pub` to your `ssh-server`. You can either use an ip address or an FQDN (e.g. `hcloud.simmy.ch`) as `ssh-server`

Connect to the server

```
ssh <username>@<ssh-server>
```

Useful links

<https://www.linuxshelltips.com/passwordless-ssh-login/>

Mount SAMBA shares

Introduction

There are several ways of mounting SAMBA shares on a Linux machine. This manual gives an overview.

Prerequisite

It makes things easier if the Linux client is a member of an Active Directory domain. Hence I use Zorin OS, this can easily be achieved with the correct setting during the installation:

Use Active Directory checkbox

If you want to join a Linux computer to an Active directory, please refer to:

[AD Join](#)

Mount SAMBA shares

Manual mount

```
mount -t cifs -o username=<user>,password=<secret-password> //xigma-prime.simmy.ch/backup /mnt/backup
```

Permanent mount with fstab

In the fstab, I use the following command:

```
//xigma-prime.simmy.ch/images /mnt/images cifs  
credentials=/root/.smbcredentials,uid=1000,forceuid,gid=1000,forcegid 0 0
```

This will mount the share images to the mountpoint /mnt/images. The credentials are saved in the file .smbcredentials:

```
username=<username>
password=<password in cleartext>
domain=simmy.ch
```

The file itself is placed in the home directory of root. The access right are limited to read only for the root user. So there is minimal protection for the password in clear text. Only the root user can read it.

The share(s) will be mounted during the boot process. This works most of the times, but not always.

Permanent mount with pam_mount

It is more desirable to mount the SAMBA shares when the user logs in, rather during boot.

Installation of the necessary modules

```
apt install -y libpam-mount keyutils cifs-utils smbclient
```

Configuration entry in /etc/security/pam_mount.conf.xml

The following lines have to be added to the file after the line <mkmountpoint enable="1" remove="true" />:

```
<volume
fstype="cifs"
server="xigma-prime.simmy.ch"
path="images"
mountpoint="~/mnt/images"
options="sec=krb5,cuid=%(USERUID),workgroup=SIMMY,vers=3.0" />
```

`<mkmountpoint enable="1" remove="true" />` means that the mount point is created and removed automatically. Also there is no password saved on the computer. I also placed the mount point into the user home directory.

Permanent mount with GPO

It is possible to utilize GPOs to mount SAMBA shares on a Linux machine, that is joined to an Active Directory. However, I did not try this possibility.

Temporary mount

It is possible to mount a share with a file manager. This mount will be lost after log off or a reboot.

Useful resources

Check for open ports

<https://phoenixnap.com/kb/linux-check-open-ports>

```
lsof -nP -iTCP -sTCP:LISTEN
```

```
netstat -tunlp
```

```
ss -tunlp
```

```
nc -z -v localhost 1-65535 2>&1 | grep succeeded
```

Network browsing not working

Sometimes network browsing or the mapping of a SMB share with a file manager in Linux is not working. I found one of the most likely causes for this problem is a missing package.

After installing gvfs-smb network browsing was working fine.

Installation on Fedora

```
sudo dnf install gvfs-smb
```

Display IP address on Panel in Xfce

Create a small shell script show_ip.sh :

```
#!/bin/bash

# Get all addresses from hostname -l
IP_ADDRESSES=$(hostname -l)

# Split into individual IP addresses
IFS=' ' read -r -a IP_ADDRS <<< "$IP_ADDRESSES"

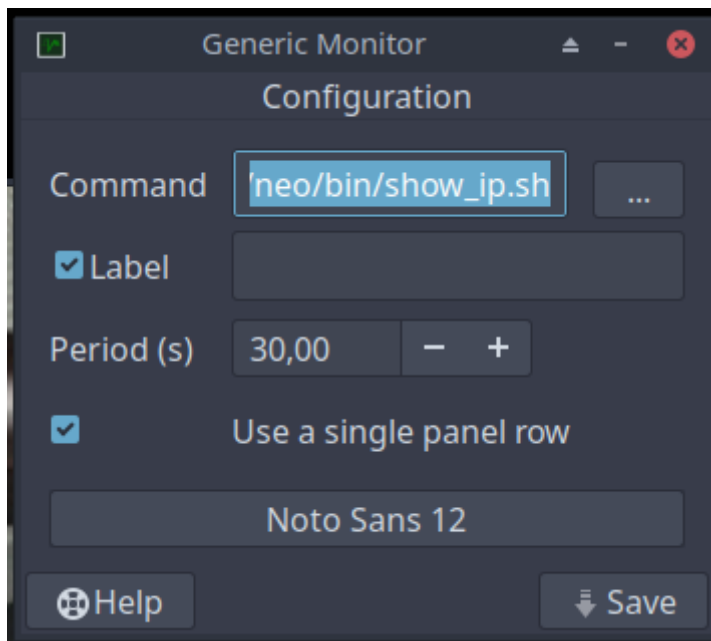
# Find the first IPv4 address
for IP in "${IP_ADDRS[@]"; do
    if [[ "$IP" =~ ^[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}$ ]]; then
        echo "$IP"
        break
    fi
done
```

```
#!/bin/bash

ip -4 addr show scope global | grep -oP '(?<=inet\s)\d+(\.\d+){3}'
```

```
chmod +x show_ip.sh
```

Add a generic monitor:



That's it.

Biometrics: Fingerprint

Introduction

Enabling fingerprint login is quite simple on Fedora, hence all necessary software is installed and all configurations are pre-configured.

Configuration

To add a signature for a finger, run:

```
fprintd-enroll
```

To verify the newly created fingerprint, use:

```
fprintd-verify
```

By default every user is allowed to enroll new fingerprints without prompting for the password or the fingerprint.

Useful links

[How to enable fingerprint login?](#)

Disable SELinux on Fedora

Permanently Disable SELinux on Fedora

Update the SELinux configuration file and set `SELINUX=disabled` to permanently disable the SELinux on your system. This will completely disable all the SELinux contexts.

```
sudo nano /etc/selinux/config
```

Set `SELINUX` value to `disabled` :

```
SELINUX=disabled
```

[Disable SELinux in Fedora](#)

Reboot your instance after making changes.

Note – You can again activate the SELinux by setting `SELINUX=enforcing` in configuration file.

[How to Disable SELinux on Fedora](#)

Create boot USB

```
sudo lsblk
```

```
sudo dd if=./Fedora-KDE-Live-x86_64-41-1.4.iso of=/dev/sdb bs=4M status=progress oflag=sync
```

Install PVE-VDIClient on Arch Linux

Introduction

This VDI client connects directly to Proxmox VE and allows users to connect (via Spice) to any VMs they have permission to access.

[PVE-VDIClient](#)

Installation

Install this first:

```
python3-pip python3-tk virt-viewer git
```

```
sudo pacman -S python tk virt-viewer git
git clone https://github.com/joshpatten/PVE-VDIClient.git
cd ./PVE-VDIClient/
chmod +x requirements.sh
./requirements.sh
sudo cp vdiclient.py /usr/local/bin
sudo chmod +x /usr/local/bin/vdiclient.py
cp vdiicon.ico ~/icons/
```

Configuration

On the client

```
~/.config/VDIClient/vdiclient.ini
```

[General]

This is the title that is displayed to the user

title = VDI Login

This is the PySimpleGui Theme that is used. Run vdiclient.py with flag `--list_themes` for a list of themes

theme = LightBlue

Program Icon

icon = vdiicon.ico

Logo displayed on all windows

logo = vdiclient.png

Enable Kiosk mode, which does not allow the user to close anything

kiosk = False

Enable/Disable Fullscreen mode (not applicable in Kiosk mode)

fullscreen = False

Disable viewer_kiosk mode if kiosk is set to true, this allows overriding remote_viewer kiosk mode

#viewer_kiosk = False

Enable displaying SPICE ini file before opening virt-viewer

inidebug = False

Select which guest types to display. Acceptable values: both, lxc, qemu

guest_type = both

Show VM option for resetting VM

#show_reset = True

Set Window Dimensions. Only use if window isn't sizing properly

#window_width = 800

#window_height = 600

PVE-VDIClient supports multiple clusters. Define them with sections that start with Hosts. followed by the name

you wish to display to your end users. This example is Hosts.PVE which would display PVE to your users

[Hosts.PVE]

JSON dictionary of servers in the cluster

Format is 'IP/FQDN': PORT

NOTE: MAKE SURE THAT ALL LINES ARE INDENTED

hostpool = {

"pve01.simmy.ch" : 8006,

"pve02.simmy.ch" : 8006

}

This is the authentication backend that will be used to authenticate

auth_backend = pve

If enabled, 2FA TOTP entry dialog will show

```
auth_totp = false
# If disabled, TLS certificate will not be checked
tls_verify = false
# User name (if using token)
# NOTE: If only one cluster is defined, this will auto-login
# If user, token_name, and token_value are set
#user = user
# API Token Name
#token_name = dvi
# API Token Value
#token_value = xxx-x-x-x-xxx
# Password Reset Command Launch. Has to be full executable Command
#pwresetcmd = start chrome --app=http://pwreset.example.com
# Automatically connect to a VMID upon authentication
#auto_vmid = 100

# An additional cluster definition
#[Hosts.PVE2]
# JSON dictionary of servers in the cluster
# Format is 'IP/FQDN': PORT
#hostpool = {
#    "10.10.10.100" : 8006,
#    "10.10.10.111" : 8006,
#    "pve1.example.com" : 8006
#    }
# This is the authentication backend that will be used to authenticate
#auth_backend = pve
# If enabled, 2FA TOTP entry dialog will show
#auth_totp = false
# If disabled, TLS certificate will not be checked
#tls_verify = false
# User name (if using token)
# NOTE: If only one cluster is defined, this will auto-login
#user = user
# API Token Name
#token_name = dvi
# API Token Value
#token_value = xxx-x-x-x-xxx
# Password Reset Command Launch. Has to be full executable Command
```

```
#pwresetcmd = start chrome --app=http://pwreset.example.com
```

```
# Automatically connect to a VMID upon authentication
```

```
#auto_vmid = 100
```

```
[SpiceProxyRedirect]
```

```
# The Spice Proxy provided by the Proxmox API may need to have its host/port rewritten
```

```
# These rewrite rules are written `IP:port = IP:port`
```

```
# 1. Use the inidebug and read the current proxy=pve1.example.com:3128
```

```
# 2. Add your proxmox ip to the right side e.g. 123.123.123.123:6000
```

```
pve1.example.com:3128 = 192.168.1.99:6000
```

```
#[AdditionalParameters]
```

```
# If you wish to define additional parameters to pass to virt-viewer you may define them here
```

```
# More parameter definitions here: https://www.mankier.com/1/remote-viewer
```

```
# Some Examples:
```

```
# Enable USB passthrough
```

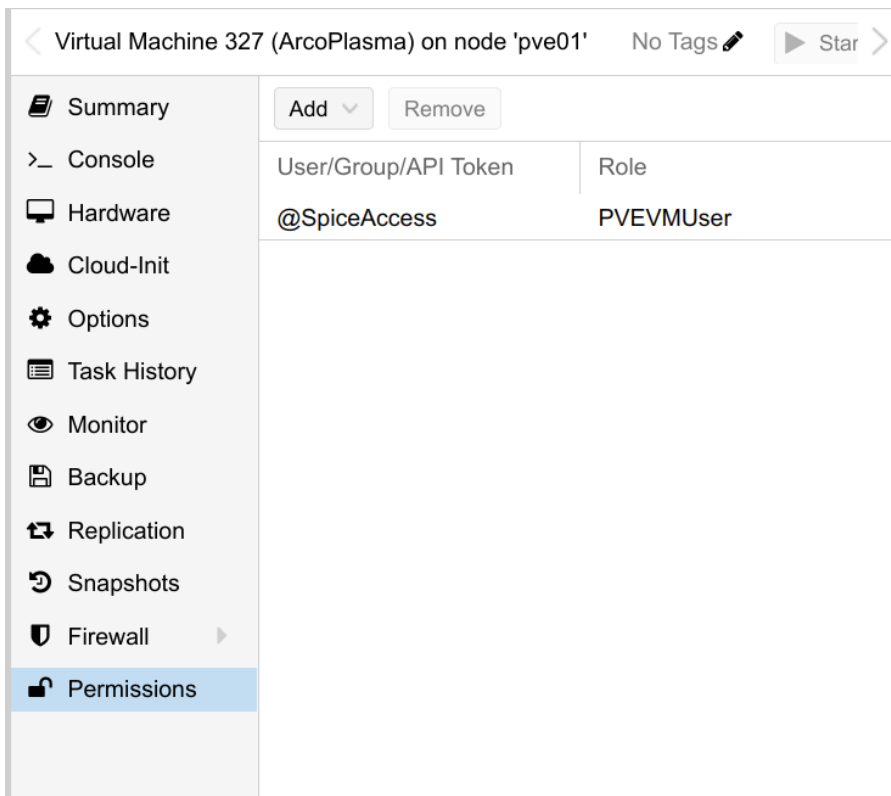
```
#enable-usbredir = true
```

```
# Enable auto USB device sharing
```

```
#enable-usb-autoshare = true
```

On the Proxmox VE server

1. Create a user (e.g. pvi) in the realm Proxmox VE authentication server
2. Create a group (e.g. Spice Access)
3. Add the group to permissions of all SPICE enabled virtual machines
4. Add the role PVEVMUser



Licensing

In case there is a warning about Licensing or trial period, you can enter your license key into this file:

~/config/PySimpleGUI/settings/_PySimpleGUI_settings_global_.json

```
{ "-temp2-": 31082748, "-LICENSE KEY-":
"egyHj1MnawWiNflobjnzN7ISVpHQlbwwZGSvli6FIjKxR3I9dnmhVpsybz3JBKlGcMibleslIjKsXmpPYb2BVVuMcV22V6JV
RrCslc6QMITNciyMNPDRQrzyMXzGAVwnOsCPwUi8TyG9lfj7ZPWU5qzuZpUSRIIOcdGvx9vneDW21MlybInkRXWOZg
XDJUZgaNWA9Nu2IjJcoSxKLiCljPObYAWEl2laRumbIByOck3ZQYiFOQisJdlrbc2bxMn6ZsXWl6iULRC4JD0hY6Wu1Ilm
ThGvFmzWdiCXI36Wl7I4N1jcazG8luupZbG8xKIKcOiOIQsmIrKKNYvRbAXtBPh5bSn3kHi6Oli3lfiULOCFJwDddVXoNT0
PbQ2b1Yl2c2kBIDERlf9oliZM8zkUj1YOQDuMeiNLPCgJPEJYnXNRuIESeX6NtzcdKWkVgkMlrjvoYigMWDEgXvnMKjrM
HvnM2jqAsyWNDCTloslkhYRFh3dLGNVzFAetHSBVpvcam6VkzblgjQofiGMFDPg3vPMmjpMOvnMgjWA5yQNGSKlrsjI
xktVOt8YHW6lqsvQnWyRHkCcSmdV1zlcZyYI56LIYmmgGuSc62AN4o3axWM5mkTbPGOVpy7QGH4BvyObp3iR9vx
bOmZ1LhwaIWHwsuDYE2egEi8LmC9JoJEU4EDF0k8ZNHjLIUcy39MdizOHisi44TN5S04SxjODTDUMurMEjBlw5HLvjZ
QhyelpnN0u=9733ae7aa77212d35ae97ed325e69a9c0312af879bbac5a6c389d1873619b5313d3d32322e397b0
13ac43265bdb0d19b7df45c9157a6d7552fb5591b6aef5d42ef48fd5424265a1e0e849562dbdec12dfd6c7cbb06f
d9e1a7dbc51e63716e69c978ca072cc2a331c2cb052198431513cfa57e240d98e6cb0aa665ad0ec7db0ff287cd41
1666fa5134e064b34611674246dc4a3db98c8b3501a388f3812e4c63adf046a9eab973b76077bbb67bd874499f59
f5801b37b795ab9cdf0d87b549cd02cbc6794ad2a3a71ad3f9833c76fa0e760b0f950c7e06a9d4beb299d22c41f33
cf346af4c9219de9ea396268f67a6adc22ac97931a742841b591f63e816bf9891205e18d4ad8dedf1e7b43c76bab4
```



```
3ac99e77e28476746b1400d6b6ea06c9c26464b922f858c3ff0b9a9b4bbb49831cd7db5729570d05e1ffcc6fb8635
108d60337c74ad81154b003d567b7a8fb5d098d0296e3ab9925f71c442676c697930856642d68a4132d4156226
466402f057637933319b1906df45679665d18cfbdcf06a6bb4b9188134f18a71e9605fc4697bd5de6340f824693e1
8de579155294b7e4606b305c90cbfd82bb9298a9b01237deee29bb3a0c51cd20e0ba4622954724165a36a2f3fe9
ea1c4c64f90c6c5ba6e2cbdbcbff782a0e63c758e58f0d300274a9c4b6d5384b31968e294f4117ce898864c622245
a50d0772882cedba63aa00f", "-temp1-": "67707579", "-trial period warned-": false}
```

Useful links

Install network scanner on Archlinux

Install the drivers with pamac

run the commandline:

```
sudo brsaneconfig4 -a name=Brother model=MFC-9330CDW ip=192.168.1.108
```

check if it is working

```
scanimage -L
```

```
device `brother4:net1;dev0' is a Brother Brother MFC-9330CDW  
device `v4l:/dev/video2' is a Noname Logitech BRIO virtual device  
device `v4l:/dev/video0' is a Noname Logitech BRIO virtual devi
```

Install xrdp

Introduction

Fedora

```
sudo dnf install -y xrdp xorgxrdp
sudo systemctl enable xrdp
sudo systemctl start xrdp
sudo systemctl status xrdp
```

Create the group tsusers and add all users using Remote Desktop to it

Open Firewall on port 3389

Check if the service is listening on tcp4

```
sudo nano /etc/xrdp/xrdp.ini
```

```
port=tcp://:3389
```

```
; Some session types such as Xorg and Xvnc start a display server.
; Startup command-line parameters for the display server are configured
; in sesman.ini. See and configure also sesman.ini.
```

```
[Xorg]
```

```
name=Xorg
```

```
lib=libxup.so
```

```
username=ask
```

```
password=ask
```

port=-1

code=20

Install Cockpit and Firewalld on Debian 12

Install Cockpit, Firewalld, and Open Ports on Debian 12

1. Update the System

```
sudo apt update  
sudo apt upgrade -y
```

2. Install Cockpit

```
sudo apt install -y cockpit
```

3. Enable and Start Cockpit

```
sudo systemctl enable --now cockpit.socket
```

4. Install firewalld

```
sudo apt install -y firewalld
sudo systemctl enable --now firewalld
sudo install cockpit-machines cockpit-pcp network-manager cockpit-networkmanager -y
```

Note: If you previously used another firewall (like UFW), remove it first:

```
sudo apt remove --purge ufw
```

5. Open Required Ports in firewalld

- **SSH** (port 22): For remote access
- **HTTP** (port 80): For web traffic
- **HTTPS** (port 443): For secure web traffic
- **Cockpit** (port 9090): For Cockpit web UI
- **Webmin** (port 12321): For Webmin web UI (on Turnkey images, default is port 1000)

```
sudo firewall-cmd --zone=public --add-service=ssh --permanent
sudo firewall-cmd --zone=public --add-service=http --permanent
sudo firewall-cmd --zone=public --add-service=https --permanent
sudo firewall-cmd --zone=public --add-service=cockpit --permanent
sudo firewall-cmd --zone=public --add-port=12321/tcp --permanent
sudo firewall-cmd --reload
```

6. Verify firewalld Rules

```
sudo firewall-cmd --list-all
```

7. Access Cockpit and Webmin

- Cockpit: `https://your-server-ip:9090`
- Webmin: `https://your-server-ip:12321`

References

- Official Cockpit documentation: cockpit-project.org

- HowtoForge: [Install Cockpit Web Console on Debian 12](#)
- edafe.de: [Install Cockpit on Debian 12 bookworm](#)
- Webmin Firewall: webmin.com/firewall.html

Install xrdp on Fedora 42

How to Set Up XRDP on Fedora 42 XFCE

Follow these steps for a reliable and quick XRDP setup with XFCE on Fedora 42:

1. Install XRDP and XFCE (if not already installed)

```
sudo dnf install xrdp xorgxrdp
```

2. Enable and Start XRDP Service

```
sudo systemctl enable --now xrdp
```

3. Configure the Firewall

open Port 3389 for rdp

4. Set Up the XFCE Session for XRDP

Create a file named `.Xclients` in your home directory with the following content:

```
echo "xfce4-session" > ~/.Xclients
```

```
echo "xfce4-session" > ~/.Xclients  
chmod +x ~/.Xclients
```

6. Restart XRDP Services

```
sudo systemctl restart xrdp  
sudo systemctl restart xrdp-sesman
```


7. Connect via RDP

- Use Devolutions RDM or any RDP client.
- Enter your Fedora machine's IP address and credentials.

Summary Table

Step	Command/Action
Install XRDP	<code>sudo dnf install xrdp xorgxrdp</code>
Enable & start service	<code>sudo systemctl enable --now xrdp</code>
Firewall open port	<code>sudo firewall-cmd --permanent --add-port=3389/tcp; sudo firewall-cmd --reload</code>
Configure session	<code>echo "xfce4-session" > ~/.Xclients; chmod +x ~/.Xclients</code>
(Optional) SELinux fix	<code>sudo chcon --type=bin_t /usr/sbin/xrdp*</code>
Restart XRDP	<code>sudo systemctl restart xrdp xrdp-sesman</code>

This setup gives you a fast, graphical remote desktop on Fedora XFCE with minimal hassle.

Install send mail service on Fedora

Fedora: Install and Configure Authenticated Mail Sending with `msmtp` and `s-nail`

Overview

This guide explains how to set up authenticated email sending from the command line on Fedora using **msmtp** (a lightweight SMTP client) and **s-nail** (a mailx-compatible mail utility). This method is ideal for scripts and system notifications in environments where only authenticated SMTP is allowed.

1. Install Required Packages

```
sudo dnf install msmtp s-nail
```

2. Configure msmtp

1. Copy the example configuration (optional):

```
sudo cp /usr/share/doc/msmtp/msmtprc-system.example /etc/msmtprc
```

2. Edit `/etc/msmtprc` and adjust to your SMTP provider:

```
sudo nano /etc/msmtprc
```

Example configuration:

```
defaults
auth      on
tls       on
tls_trust_file /etc/ssl/certs/ca-bundle.crt
logfile   /var/log/msmtp.log

account    default
host       mail.hosting.de
port       587
from       admin@simmy.org
user       admin@simmy.org
password   <super-secret>
```

3. Set permissions to protect your password:

```
sudo chmod 600 /etc/msmtprc
```

3. Configure s-nail to Use msmtplib

Add the following line to `/etc/s-nail.rc` or your `~/.mailrc`:

```
set mta=/usr/bin/msmtp
```

4. Send a Test Email

```
echo "This is the body" | mail -s "Test Subject" recipient@example.com
```

- If the command returns no errors, the mail was sent successfully.
- Check `/var/log/msmtp.log` for troubleshooting if needed.

5. Notes

- If you receive an error like "**Authenticated user is not permitted to override sender address**", ensure the `from` address in `/etc/msmtprc` matches the authenticated SMTP user, or configure your SMTP provider to allow the desired sender address.
- For use in scripts (e.g., backup notifications), simply use the `mail` command as shown above.

References

- [Sending e-mails via mailbox.org with msmtplib on Fedora](#)
- [Fedora Docs: Mail Servers](#)
- [Fedora Forum: Sending mail with the \(mailx\) command](#)