

# Setup

## Server Setup

- Login to the server via SSH as root
- Install sudo, create the main user, and add him to the sudoers group

```
# apt install sudo
adduser <username> sudo
```

- (VPS) change the machine name:

```
sudo hostnamectl set-hostname <new-name>
sudo edit /etc/hosts
```

- enable *force\_color\_prompt=yes* in file *~/.bashrc*
- copy */home/user/.bashrc* to */root* and modify PS1 prompt color (32m -> 31m)

## Tuning

- [Swap behavior](#)

## Network

- In a minimal install without GUI is no NetworkManager present. Configure the static ip address first to be able to access the host.
- Modify */etc/network/interfaces*:

```
auto enp0s3
iface enp0s3 inet static
    address 192.168.3.150/24
    gateway 192.168.3.1
    dns-nameservers 192.168.3.1
```

- Reboot with either of the following commands:

```
sudo reboot
sudo shutdown -r now
```

- Don't forget to bridge the interface if it is a VirtualBox virtual machine.

## Security

- Configure [SSH Access](#)
- Install [UFW Firewall](#)

- Disable root login by changing file `/etc/passwd` from line 1 to line 2

```
root:x:0:0:root:/root:/bin/bash
root:x:0:0:root:/root:/sbin/nologin
```

## Install and configure packages

- Additional packages

```
$ sudo apt update
$ sudo apt install vim
$ sudo apt install zip
$ sudo apt install rename
$ sudo apt install rsync
$ sudo apt install curl
$ sudo apt install hping3
$ sudo apt install openjdk-17-jre openjdk-17-jdk
```

- Change visudo editor

```
$ sudo update-alternatives --config editor
```

- Check install

```
apt-cache policy <packagename>
```

## Upgrade linux headers

- Check remaining space on the boot partition with:

```
df -h | grep "^/dev/"
```

- Remove outdated kernel packages:

```
sudo apt-get autoremove
```

- First check your kernel version, so you won't delete the in-use kernel image, running:

```
uname -r
```

- Delete the kernels you don't want/need anymore by running this:

```
sudo apt-get remove linux-image-VERSION
```

- Update grub kernel list:

```
sudo update-grub
```

# System Startup

Debian 11 uses Systemd, not SysV, that's why your commands in `/etc/rc.local` file would not run at system boot time. This guide explains how to enable `/etc/rc.local` script to run on system startup.

1. Create a `rc.local` unit file, follow [Services - rc.local](#)
2. Create file `rc.local` in `/etc` with the following content:

```
vim /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
exit 0
```

3. Add execute permissions and start daemon:

```
chmod +x /etc/rc.local
systemctl daemon-reload
systemctl start rc-local
systemctl status rc-local
```

● `rc-local.service` - `/etc/rc.local` Compatibility  
Loaded: loaded (`/lib/systemd/system/rc-local.service`; static; vendor preset: Drop-In: `/lib/systemd/system/rc-local.service.d`  
└─`debian.conf`  
Active: active (exited) since Sun 2019-05-05 10:55:29 +07; 11min ago  
Process: 1443 ExecStart=`/etc/rc.local start` (code=exited, status=0/SUCCESS)  
Tasks: 0 (limit: 4915)  
CGroup: `/system.slice/rc-local.service`

- [How to Enable /etc/rc.local with Systemd](#)
- [Where is rc.local in Debian 9 \(Debian Stretch\)](#)

# System Shutdown

1. Put a symbolic link to the script to be run at shutdown into `/lib/systemd/system-shutdown`

- [How to execute scripts in /usr/lib/systemd/system-shutdown/ at reboot or shutdown?](#)
- [How to run commands at shutdown on Linux](#)

From:

<https://wiki.condrau.com/> - **Bernard's Wiki**

Permanent link:

<https://wiki.condrau.com/deb13:minisetaup?rev=1789029036>

Last update: **2026/09/10 15:30**

