Ubuntu Install Notes / Linux Documentation

Ryan Nash

Contents

USB Install

- Boot into Xubuntu live usb
- Format hdd using `parted` or `gparted`
- Run through Xubuntu install process
- Reboot

Newly Installed System

- Update packages
  - `sudo apt update`, `sudo apt upgrade`, `sudo apt dist-upgrade`
- Reboot

Install Drivers

GPU

- Install Nvidia proprietary driver from *Software & Updates* in **Settings**
- Disable Nvidia GPU in `nvidia-settings` for better battery life

X11

- Copy `xorg.conf.d` folder from dotfiles to `/etc/X11/`

Battery Control

- Install `tlp` for better battery management
  - `sudo apt install tlp tlp-rdw acpi-call-dkms`
  - `sudo tlp start`

Fan Control

- Install `thinkfan` for better fan control
  - sudo bash
  - apt install thinkfan
  - echo “options thinkpad_acpi fan_control=1” > /etc/modprobe.d/thinkfan.conf
  - sh -c ‘echo coretemp >> etc/modules’
  - modprobe thinkpad_acpi && modprobe coretemp
  - nano -w /etc/default/thinkfan
    - add `START=yes` to second line (below comment)
  - copy `thinkfan.conf` from dotfiles to `/etc/`
  - systemctl enable thinkfan.service
  - reboot

Network

- Sign-in to *Firefox Sync*
- Disable WebRTC
  - Type `about:config` in *URL bar*
  - Search for `media.peerconnection.enabled`
  - Toggle value to `false`
    - To reverse it, simply toggle back
- Fix right-click context menu when using i3
  - Go to `about:config` again
  - Change `ui.context_menus.after_mouseup` from **false** to **true**

VPN

- Install **Mullvad** configuration file for **OpenVPN**
  - `sudo apt-get install openvpn network-manager-openvpn network-manager-openvpn-gnome`
  - Download config file from `https://mullvad.net/en/download/config/`
  - Open the downloaded file and remove everything from `<crl-verify>` to `</crl-verify>` (including them)
  - Import a saved VPN configuration using **Network Manager**
  - Edit the connection
    - Enter **Mullvad** account number in the `username` field
    - Enter `m` in the `password` field
  - `sudo service network-manager restart`
  - Click on *Network Icon* and select created VPN
  - *Mullvad with OpenVPN currently has DNS leaks after doing the above steps. Figure out a fix!*

Ricing

Dotfiles

- Copy `.Xresources` to *user home folder*
  - `xrdb .Xresources`
- Copy `compton.conf` to `~/.config/`

Fonts

- Create `~/.fonts directory`
- Copy `.ttf` files to separate folders within `.fonts`
  - **System San Francisco**
  - **Adobe Source Code Pro**
  - **FontAwesome**
- Set **Source Code Pro** as the font in:
  - `xfce4-terminal`
  - **XFCE Settings > Appearance > Fonts > Default monospace font**
- Set **SFNS Display**/San Francisco Display as the font in:
  - **XFCE Settings > Appearance > Fonts**
  - **XFCE Settings > Window Manager**
- Customise `xfce4-terminal` using its *preference* GUI

GTK Themes & Icons

- Install `arc-theme`
- Install `numix-solarized` theme
- Install `macos-sierra` theme
  - and dependencies
- Install `papirus-icon-theme`
- Install MacOS icons
- For **XFCE** set theme/icons in:
  - **XFCE Settings**
  - **Window Manager**
- For **i3** set theme/icons in:

    - lxappearance (`sudo apt install lxappearance`)
- Setting wallpaper in XFCE will also set it for i3

i3wm

- Install the **i3 window manager**
    - `sudo apt install i3-suckless-tools i3blocks i3lock`
- Copy config file from dotfiles to `~/.config/i3/`

i3-gaps

- Install **i3-gaps** dependencies

```
$ sudo apt install \
libxcb1-dev libxcb-keysyms1-dev libpango1.0-dev \
libxcb-util0-dev libxcb-icccm4-dev libyajl-dev \
libstartup-notification0-dev libxcb-randr0-dev \
libev-dev libxcb-cursor-dev libxcb-xinerama0-dev \
libxcb-xkb-dev libxkbcommon-dev libxkbcommon-x11-dev \
autoconf libxcb-xrm0 libxcb-xrm-dev automake
```

- Clone the repository
    - `git clone https://www.github.com/Airblader/i3 /tmp/i3-gaps`
    - ‘cd /tmp/i3-gapsr
- Compile & install

```
$ autoreconf --force --install
$ rm -rf build/
$ mkdir -p build && cd build/
```

- Disabling sanitizers is important for release versions!
- The prefix and sysconfdir are, obviously, dependent on the distribution.

```
$ ../configure --prefix=/usr --sysconfdir=/etc --disable-sanitizers
$ make
$ sudo make install
```

- After installing i3 and i3-gaps, `dunst` took precedence over the nicer looking `xfce4-notify-d`, so I remove it
    - `sudo apt remove dunst`
    - `sudo apt autoremove`
- Install **git**, `sudo apt install git`

i3blocks

- Custom scripts: (currently located in: `~/.config/i3/i3blocks/`)
    - IntBat
    - ExtBat
- These scripts need to be made executable in order for them to appear in i3blocks bar
    - `sudo chmod +x ScriptName`

Powerline

- `sudo pip install powerline-status`

- `sudo apt install powerline`
- `sudo pip install powerline-gitstatus`
- Add to `.bashrc` and `.vimrc`
- Configure colors using `config.json`

Powerline-gitstatus

-

Vim

- Install `vim-gtk`
  - `sudo apt install vim-gtk`
- Copy `.vimrc` config file and `.vim` folder from dotfiles to `~/`
- Set vim as default editor
  - `sudo update-alternatives -config editor`
- To use my .vimrc while editing root owned files `sudo -E vi ~/.vimrc` Edit vim solarized colorscheme

Vim Plugins

- Create ~/.vim/pack/plugins/start/
  - This is the folder to install plugins in
- Example:
  - `cd ~/.vim/pack/plugins/start`
  - `sudo git clone https://github.com/tpope/vim-sensible.git`

LaTeX

- Install Ubuntu LaTeX package
  - `sudo apt install texlive-full`
- After installing LaTeX, FontAwesome icons stop rendering in i3
  - To fix this, create the directory `~/.config/fontconfig/conf.d/`
  - Copy `/etc/fonts/conf.d/60-latin.conf` to the above folder
  - Edit all `<family>xxxxxx</family>` sections to `<family>FontAwesome</family>`
  - Except **heading lines** (e.g. serif)
- Install `mupdf` and `xdotool` packages for pdf viewing

R Markdown

- Install R mardown and required packages
  - `sudo apt install r-base r-base-dev pandoc pandoc-citeproc`
- Run R session
  - `sudo R`
  - Install `rmarkdown`, `install.packages("rmarkdown")`

  - Exit R

SC-IM (CLI-based spreadsheet editor)

- Install required dependencies and `libxlsxwriter`

```
$ sudo apt-get install bison libncurses5-dev libncursesw5-dev libxml2-dev libzip-dev
$ git clone https://github.com/jmcnamara/libxlsxwriter.git
$ cd libxlsxwriter/
$ make
$ sudo make install
```

- Refresh dynamic link cache
  - `sudo ldconfig`
- Download (git clone) and compile SCI-IM

```
$ cd ..
$ git clone https://github.com/andmarti1424/sc-im.git
$ cd sc-im/src
$ make
$ sudo make install
```

- Run with `sc-im` command, exit same way as **Vim**

Music Player

Mopidy with Spotify using ncmpcpp

- Add mopidy archive gpg key
  - `wget -q -O - https://apt.mopidy.com/mopidy.gpg | sudo apt-key add -`
- Add mopidy to sources
  - `sudo wget -q -O /etc/apt/sources.list.d/mopidy.list https://apt.mopidy.com/stretch.list`
- `sudo apt update`
- Install `mopidy` and `mopidy-spotify` packages
- Install `ncmpcpp` package
- Install `mpc` package (used for outputting mopidy now playing to bars/i3blocks)
- Copy mopidy folder from dotfiles to `~/.config/`
- Copy .ncmpcpp folder from dotfiles to `~/.`
- Run mopidy in terminal foreground for first run `mopidy`
- Run `ncmpcpp` and test out Spotify
- Install `socat` package to watch UDP port for visualiser
- End terminal process, run my **Visualiser batch script**
- *Note that my Visualiser script must be run on each boot*

Getting Spotify playlists to work

- Kill all instances of mopidy, then remove installed `mopidy-spotify` package
- `sudo apt-get remove mopidy-spotify`
  - `sudo apt autoremove`
- Install mopidy-spotify dependencies
  - `sudo apt install libspotify12 python-cffi python-ply python-pycparser python-spotify`
- Clone mopidy-spotify from GitHub repo

  - `git clone https://github.com/BlackLight/mopidy-spotify.git /tmp/mopidy-spotify` *Go to cloned directory
  - `cd mopidy-spotify`
- Switch to `fix/incompatible_playlists` branch
  - `sudo git checkout fix/incompatible_playlists`
- Install cloned-mopidy spotify
  - `sudo python2 setup.py build install`

Visualiser Script Breakdown

- create fifo file for visualiser `mkfifo /tmp/mpd.fifo`
- run `while :; do socat -d -d -T 1 -u UDP4-LISTEN:5555 OPEN:/tmp/mpd.fifo; done &`
- notice the ‘&’ at the end, this lets the command run in the background, freeing up terminal space

Misc Required Programs

- `xbacklight` (controls brightness in i3)
- `redshift` (warmer colors at night)
- `rofi` (used by i3 to launch programs instead of `dmenu`)
  - customised using `.Xresources` file
- `compton` (window compositor I use with i3)
  - Customised with `~/.config/compton.conf`
  - Currently I only have an opacity rule
- Timeshift
  - `sudo add-apt-repository -y ppa:teejee2008/ppa`
  - `sudo apt update`
  - `sudo apt install timeshift`
- **VSCode** (GUI text editor for use with XFCE)
- Transmission
  - In *preferences* set to `Stop seeding at ratio:` to 0
- Pip
  - ‘sudo apt install python-pip
- S-Tui, install using Pip
  - `sudo pip install s-tui`
- CalCurse (terminal calender/to list, *install package* `calcurse`)
- feh (background setter and image viewer for i3)

Aliases

- Located in `~/.bashrc`
  - Syntax is `alias **name**='path to script/command name'`

Virtualisation

- Install `libvirt-bin` and `virt-manager`
- Add `intel-iommu=on` to `GRUB_CMDLINE_LINUX_DEFAULT` between quotes
- Run `sudo update-grub` then reboot

Bumblebee

- Install `sudo apt-get install bumblebee bumblebee-nvidia primus linux-headers-generic`