
Ubuntu Settings

Ruohong Jiao

2020-04-03

Establish a Ubuntu 18.04 Server

Ubuntu Server

ubuntu server 18.04 distro

Change Sources

/etc/apt

```
1 sudo cp ./sources.list ./sources.list.backup
```

/etc/apt/sources.list

```
1 # tsinghua
2 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic main restricted
  universe multiverse
3 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-updates main
  restricted universe multiverse
4 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-backports main
  restricted universe multiverse
5 deb https://mirrors.tuna.tsinghua.edu.cn/ubuntu/ bionic-security main
  restricted universe multiverse
6
7 # ustc
8 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic main restricted universe
  multiverse
9 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-updates main restricted
  universe multiverse
10 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-backports main
  restricted universe multiverse
11 deb https://mirrors.ustc.edu.cn/ubuntu/ bionic-security main restricted
  universe multiverse
```

update

```
1 sudo apt-get update
2 sudo apt-get upgrade
```

Make Usr Dirs

~/

```
1 mkdir username
```

~/username

```
1 mkdir Apps
2 mkdir Documents
3 mkdir Downloads
4 mkdir Workplaces
```

Install Necessary pkgs

```
1 sudo apt-get install git tmux openssh-server mosh python-dev python3-
  dev libncurses5-dev python-pip python3-pip build-essential cmake
  perl-tk wget zip unzip python-m2crypto libsodium23 tsocks
```

Download Installer

- Oracle JDK 8
- TexLive 2019
- Anaconda mirror from tsinghua
- Vim

~/username/Downloads

```
1 git clone https://github.com/vim/vim.git
```

- Shadowsocksr

doc

~/username/Apps

```
1 curl https://github.com/shadowsocksr/shadowsocksr/archive/3.2.2.
  zip
```

Config Proxies

- git proxy

```
1 git config --global user.name "me"
```

~/.gitconfig

```
1 [user]
2   name = usr
3   email = usr@xxx.com
4 [push]
5   default = simple
6 [http]
7   proxy = socks5://127.0.0.1:1080
8 [https]
9   proxy = socks5://127.0.0.1:1080
```

- curl proxy

reference

~/.bashrc and ~/.zshrc in future

```
cpp # set http socks proxy (local DNS)export http_proxy=socks5://127.0.0.1:1080
# set https socks proxy (local DNS)export https_proxy=socks5://127.0.0.1:1080
```

update

```
cpp source ~/.bashrc
```

- proxy software (ssr)

~/usrname/Apps/shadowsocksr

```
1 bash initcfg.sh
```

~/usrname/Apps/shadowsocksr/user-config.json

```
1 {
2   "server": "remote.server",
3   "server_ipv6": ":::",
4   "server_port": remote.server.port,
5   "local_address": "127.0.0.1",
6   "local_port": 1080,
7
8   "password": "m",
9   "method": "aes-128-ctr",
10  "protocol": "auth_chain_a",
11  "protocol_param": "",
12  "obfs": "tls1.2_ticket_auth",
13  "obfs_param": "",
14  "speed_limit_per_con": 0,
15  "speed_limit_per_user": 0,
16
17  "additional_ports" : {}, // only works under multi-user mode
18  "additional_ports_only" : false, // only works under multi-
    user mode
```

```
19     "timeout": 120,  
20     "udp_timeout": 60,  
21     "dns_ipv6": false,  
22     "connect_verbose_info": 0,  
23     "redirect": "",  
24     "fast_open": false  
25 }
```

~/usrname/Apps/shadowsocksr/shadowsocks

```
1 sudo python ./local.py -d start
```

Install pkgs

- Open JDK 8, Open JDK 11 and Oracle JDK 8

```
1 sudo apt-get install openjdk-8-jdk openjdk-11-jdk
```

~/usrname/Downloads

```
1 sudo tar xvf jdk-8u241-linux-x64.tar.gz --directory /usr/lib/jvm/
```

- Set Oracle JDK PATH

~/bashrc and ~/.zshrc in future

```
1 # Oracle Java JDK  
2 export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_241  
3 export JRE_HOME=${JAVA_HOME}/jre  
4 export CLASSPATH=.:${JAVA_HOME}/lib:${JRE_HOME}/lib  
5 export PATH=${JAVA_HOME}/bin:$PATH
```

update

```
1 source ~/.bashrc
```

- Set alias for Javas and Sys Pythons

- Java
- jdk1.8.0_241

```
1 sudo update-alternatives --install /usr/bin/java java /usr/lib  
  /jvm/jdk1.8.0_241/bin/java 1200  
2 sudo update-alternatives --install /usr/bin/javac javac /usr/  
  lib/jvm/jdk1.8.0_241/bin/javac 1200  
3 sudo update-alternatives --install /usr/bin/javadoc javadoc /  
  usr/lib/jvm/jdk1.8.0_241/bin/javadoc 1200
```

```
4 sudo update-alternatives --install /usr/bin/javap javap /usr/
  lib/jvm/jdk1.8.0_241/bin/javap 1200
5 sudo update-alternatives --install /usr/bin/javah javah /usr/
  lib/jvm/jdk1.8.0_241/bin/javah 1200
6 sudo update-alternatives --install /usr/bin/javafxpackager
  javafxpackager /usr/lib/jvm/jdk1.8.0_241/bin/javafxpackager
  1200
7 sudo update-alternatives --install /usr/bin/javapackager
  javapackager /usr/lib/jvm/jdk1.8.0_241/bin/javapackager
  1200
8 sudo update-alternatives --install /usr/bin/javaws javaws /usr/
  lib/jvm/jdk1.8.0_241/bin/javaws 1200
9 sudo update-alternatives --install /usr/bin/java-rmi.cgi java-
  rmi.cgi /usr/lib/jvm/jdk1.8.0_241/bin/java-rmi.cgi 1200
```

– java-11-openjdk-amd64

```
1 sudo update-alternatives --install /usr/bin/java java /usr/lib
  /jvm/java-11-openjdk-amd64/bin/java 1111
2 sudo update-alternatives --install /usr/bin/javac javac /usr/
  lib/jvm/java-11-openjdk-amd64/bin/javac 1111
3 sudo update-alternatives --install /usr/bin/javadoc javadoc /
  usr/lib/jvm/java-11-openjdk-amd64/bin/javadoc 1111
4 sudo update-alternatives --install /usr/bin/javap javap /usr/
  lib/jvm/java-11-openjdk-amd64/bin/javap 1111
```

– java-8-openjdk-amd64

```
1 sudo update-alternatives --install /usr/bin/java java /usr/lib
  /jvm/java-8-openjdk-amd64/bin/java 1081
2 sudo update-alternatives --install /usr/bin/javac javac /usr/
  lib/jvm/java-8-openjdk-amd64/bin/javac 1081
3 sudo update-alternatives --install /usr/bin/javadoc javadoc /
  usr/lib/jvm/java-8-openjdk-amd64/bin/javadoc 1081
4 sudo update-alternatives --install /usr/bin/javap javap /usr/
  lib/jvm/java-8-openjdk-amd64/bin/javap 1081
5 sudo update-alternatives --install /usr/bin/javah javah /usr/
  lib/jvm/java-8-openjdk-amd64/bin/javah 1081
6 sudo update-alternatives --install /usr/bin/java-rmi.cgi java-
  rmi.cgi /usr/lib/jvm/java-8-openjdk-amd64/bin/java-rmi.cgi
  1081
```

– Sys Python and Python3

```
1 sudo update-alternatives --install /usr/bin/python python /usr
  /bin/python2.7 10
2 sudo update-alternatives --install /usr/bin/python python /usr
  /bin/python3.6 40
3 sudo update-alternatives --install /usr/bin/python3 python3 /
  usr/bin/python3.6 40
```

- Texlive 2019

remove old Texlive 2019

```
1 rm -rf /usr/local/texlive/2019
2 rm -rf ~/.texlive2019
```

~/username/Downloads

```
1 tar xvf install-tl-unx.tar.gz
```

~/username/Downloads/install-tl-unx

```
1 sudo perl install-tl
2
3 [... messages omitted ...]
4
5 Enter command: i
6
7 [... when done, see below for post-install ...]
```

~/.bashrc, ~/.profile and ~/.zshrc in future

```
1 # Texlive 2019
2 export MANPATH=${MANPATH}:/usr/local/texlive/2019/texmf-dist/doc/
  man
3 export INFOPATH=${INFOPATH}:/usr/local/texlive/2019/texmf-dist/doc
  /info
4 export PATH=${PATH}:/usr/local/texlive/2019/bin/x86_64-linux
```

update

```
1 source ~/.bashrc
2 source ~/.profile
3 # source ~/.zshrc
```

add Tex Fonts to Sys

```
1 sudo cp /usr/local/texlive/2019/texmf-var/fonts/conf/texlive-
  fontconfig.conf /etc/fonts/conf.d/09-texlive.conf
2
3 sudo fc-cache -fsv
```

Tsinghua CTAN source

```
1 tlmgr option repository https://mirrors.tuna.tsinghua.edu.cn/CTAN/
  systems/texlive/tlnet
```

- Pandoc

```
1 sudo apt-get install pandoc
```

- Vim 8+

~/username/Downloads/vim/src

```
1 make distclean
2
3 ./configure --with-features=huge --enable-rubyinterp --enable-
  pythoninterp --enable-python3interp --enable-luainterp --enable-
  perlinterp --enable-multibyte --enable-cscope --with-python-
  config-dir=/usr/lib/python2.7/config-x86_64-linux-gnu/ --with-
  python-config-dir=/usr/lib/python3.6/config-3.6m-x86_64-linux-
  gnu/
4
5 sudo make
6
7 sudo make install
```

- Install Vundle, etc.

Vundle

make dirs

```
1 mkdir ~/.vim
2 mkdir ~/.vim/bundle
```

clone Vundle

```
1 git clone https://github.com/VundleVim/Vundle.vim.git ~/.vim/
  bundle/Vundle.vim
```

make ~/.vimrc file

```
1 vim ~/.vimrc
```

after editing config file(~/.vimrc), install vim plugins in vim

```
1 :PluginInstall
```

- Build YouCompleteMe

YouCompleteMe

~/vim/bundle/YouCompleteMe

```
1 sudo python install.py --clangd-completer --java-completer
2
3 # or we use proxy
4 # sudo tsocks python install.py --clangd-completer --java-
  completer
```


Attention:

python version is python 3.6

- Vim Config file

~/vimrc

```
1  " ~/.vimrc
2  " author: Ruohong Jiao
3  " github: github.jiaoruohong
4
5  " Vundle Settings
6  " -----
7  " Brief help
8  " :PluginList      - lists configured plugins
9  " :PluginInstall   - installs plugins; append `!` to update or
   just :PluginUpdate
10 " :PluginSearch foo - searches for foo; append `!` to refresh
   local cache
11 " :PluginClean      - confirms removal of unused plugins; append
   `!` to auto-approve removal
12 "
13 " see :h vundle for more details or wiki for FAQ
14
15 set nocompatible
16 filetype off
17
18 set rtp+=~/vim/bundle/Vundle.vim
19 let vpath='~/vim/bundle'
20
21 call vundle#begin(vpath)
22 Plugin 'VundleVim/Vundle.vim'
23
24 " Files management
25 " -----
26 Plugin 'scrooloose/nerdtree'
27 Plugin 'jistr/vim-nerdtree-tabs'
28
29 " Appearance
30 " -----
31 Plugin 'vim-airline/vim-airline'
32
33 " Coding
34 " -----
35 Plugin 'tmhedberg/SimpylFold'
36 Plugin 'vim-scripts/indentpython.vim'
37 Plugin 'Valloric/YouCompleteMe'
38 Plugin 'ctrlpvim/ctrlp.vim'
39 Plugin 'tpope/vim-fugitive'
40 Plugin 'fholgado/minibufexpl.vim'
```

```
41 Plugin 'majutsushi/tagbar'
42 Plugin 'tpope/vim-sensible'
43 Plugin 'scrooloose/syntastic'
44 Plugin 'sheerun/vim-polyglot'
45 Plugin 'SirVer/ultisnips'
46 Plugin 'honza/vim-snippets'
47 Plugin 'nvie/vim-flake8'
48
49 " LaTeX
50 " -----
51 Plugin 'lervag/vimtex'
52
53 " Markdown
54 " -----
55 Plugin 'junegunn/goyo.vim'
56 Plugin 'junegunn/limelight.vim'
57 Plugin 'plasticboy/vim-markdown'
58 Plugin 'godlygeek/tabular'
59 Plugin 'tpope/vim-surround'
60
61 " Themes
62 " -----
63 Plugin 'arcticicestudio/nord-vim'
64 Plugin 'hzchirs/vim-material'
65 Plugin 'dracula/vim', { 'name': 'dracula' }
66 Plugin 'cormacrelf/vim-colors-github'
67 Plugin 'kristijanhusak/vim-hybrid-material'
68 Plugin 'morhetz/gruvbox'
69
70 call vundle#end()
71
72 filetype plugin indent on
73
74 " Colors
75 " -----
76 if (has("termguicolors"))
77     let &t_8f = "\<Esc>[38;2;%lu;%lu;%lum"
78     let &t_8b = "\<Esc>[48;2;%lu;%lu;%lum"
79     set termguicolors
80 else
81     set t_Co=256
82 endif
83
84 " Base
85 " -----
86 syntax enable
87 syntax on
88 set number
89 set showmatch
90 set smartindent
91 set tabstop=4
```

```
92 set shiftwidth=4
93 set autoindent
94 set ai!
95 set noerrorbells
96 set colorcolumn=80,100
97 set scrolloff=3
98 " Highlight search results
99 " -----
100 set hlsearch
101 set incsearch
102
103 " -----
104 set ruler
105 " UTF-8 Support
106 " -----
107 set encoding=utf-8
108 " split directions
109 " -----
110 set splitbelow
111 set splitright
112 " Enable folding
113 " -----
114 set foldmethod=indent
115 set foldlevel=99
116 " System Clipboard
117 " -----
118 set clipboard=unnamedplus
119 set mouse=v
120 " split navigations
121 " -----
122 nnoremap <C-J> <C-W><C-J>
123 nnoremap <C-K> <C-W><C-K>
124 nnoremap <C-L> <C-W><C-L>
125 nnoremap <C-H> <C-W><C-H>
126 " Enable folding with the spacebar
127 " -----
128 nnoremap <space> za
129
130
131 " NERDTree Settings
132 " -----
133 map <C-b> :NERDTreeToggle<CR>
134 map <C-n> :tabnew<CR>
135 map <S-Right> :tabn<CR>
136 map <S-Left> :tabp<CR>
137
138 " SimpyFold settings
139 " -----
140 let g:SimpylFold_docstring_preview=1
141
142 " Tagbar settings
```

```
143 " -----
144 map <F8> :TagbarToggle<CR>
145
146 " UltiSnip settings
147 " -----
148 "set runtimepath+="/usr/share/vim/my-snippets/
149 "let g:UltiSnipsSnippetsDir="/usr/share/vim/my-snippets/'
150 "let g:UltiSnipsSnippetDirectories=["my-snippets"]
151 let g:UltiSnipsEditSplit='vertical'
152 let g:UltiSnipsExpandTrigger="<c-e>"
153 let g:UltiSnipsJumpForwardTrigger="<c-f>"
154 let g:UltiSnipsJumpBackwardTrigger="<c-b>"
155
156 " YouCompleteMe settings
157 " -----
158 " - Let YCM read tags from Ctags file
159 let g:ycm_collect_identifiers_from_tags_files = 1
160 " - Default 1, just ensure
161 let g:ycm_use_ultisnips_completer = 1
162 " - Completion for programming language's keyword
163 let g:ycm_seed_identifiers_with_syntax = 1
164 " - Completion in comments
165 let g:ycm_complete_in_comments = 1
166 " - Completion in string
167 let g:ycm_complete_in_strings = 1
168
169 let g:ycm_key_list_select_completion = ['<C-j>', '<Down>']
170 let g:ycm_key_list_previous_completion = ['<C-k>', '<Up>']
171
172 " - Goto definition with F3
173 map <F3> :YcmCompleter GoTo<CR>
174
175 " Python Settings
176 "let g:ycm_server_python_interpreter="/usr/bin/python'
177 "let g:ycm_python_interpreter_path = '~/anaconda3/envs/py36/bin/
python'
178 "let g:ycm_python_sys_path = ['~/anaconda3/envs/py36/lib/python3
.6/site-packages']
179 ""let g:ycm_extra_conf_vim_data = [
180 "" \ 'g:ycm_python_interpreter_path',
181 "" \ 'g:ycm_python_sys_path'
182 "" \]
183 ""let g:ycm_global_ycm_extra_conf = '~/global_extra_conf.py'
184 "let g:ycm_global_ycm_extra_conf = '~/ycm_extra_conf.py'
185
186 " Vim-Flake8
187 " -----
188 autocmd BufWritePost *.py call Flake8()
189
190 " VimTex
191 " -----
```

```
192 let g:tex_flavor='latex'
193 let g:vimtex_quickfix_mode=0
194 set conceallevel=1
195 let g:tex_conceal='abdmg'
196 let g:vimtex_complete_bib = { 'simple': 1 }
197
198 " Vim-Polyglot
199 " -----
200 let g:polyglot_disabled = ['latex']
201
202 " Goyo
203 " -----
204 " default: 80
205 let g:goyo_width = 80
206 " default: 85%
207 let g:goyo_height = 0.85
208 " default: 0
209 let g:goyo_linenr = 0
210
211 " Limelight
212 " -----
213 " Color name (:help cterm-colors) or ANSI code
214 let g:limelight_conceal_ctermfg = 'gray'
215 let g:limelight_conceal_ctermfg = 240
216 " Color name (:help gui-colors) or RGB color
217 let g:limelight_conceal_gui_fg = 'DarkGray'
218 let g:limelight_conceal_gui_fg = '#777777'
219 " Default: 0.5
220 let g:limelight_default_coefficient = 0.7
221 " Number of preceding/following paragraphs to include (default: 0)
222 let g:limelight_paragraph_span = 1
223 " Beginning/end of paragraph
224 "   When there's no empty line between the paragraphs
225 "   and each paragraph starts with indentation
226 let g:limelight_bop = '^\\s'
227 let g:limelight_eop = '\\ze\\n^\\s'
228 " Highlighting priority (default: 10)
229 "   Set it to -1 not to overrule hlsearch
230 let g:limelight_priority = -1
231 " Using Limelight when Goyo is started
232 autocmd! User GoyoEnter Limelight
233 autocmd! User GoyoLeave Limelight!
234
235 " Theme
236 " -----
237 "set background=[dark,light]
238 "colorschem X
239 "let g:airline_theme='X'
240 "highlight ColorColumn ctermbg=0 guibg=black
241
242 " Nord-Vim
```

```
243 " -----
244 " https://github.com/arcticicestudio/nord-vim
245 colorschem nord
246 let g:lightline = { 'colorscheme': 'nord' }
247
248 " Vim-Material
249 " -----
250 " https://github.com/hzchirs/vim-material
251 " Dark
252 "set background=dark
253 "colorscheme vim-material
254 "let g:airline_theme='material'
255 "let g:lightline = { 'colorscheme': 'vim-material' }
256
257 " Palenight
258 "let g:material_style='palenight'
259 "set background=dark
260 "colorscheme vim-material
261 "let g:airline_theme='material'
262 "let g:lightline = { 'colorscheme': 'vim-material' }
263
264 " Oceanic
265 "let g:material_style='oceanic'
266 "set background=dark
267 "colorscheme vim-material
268 "let g:airline_theme='material'
269 "let g:lightline = { 'colorscheme': 'vim-material' }
270
271 " Light
272 "set background=light
273 "colorscheme vim-material
274 "let g:airline_theme='material'
275 "let g:lightline = { 'colorscheme': 'vim-material' }
276
277 " Dracula
278 " -----
279 " https://github.com/dracula/vim
280 "colorschem dracula
281 "let g:lightline = { 'colorscheme': 'dracula' }
282
283 " Vim-Colors-Github
284 " -----
285 " https://github.com/cormacrelf/vim-colors-github
286 " Github Rust
287 "let g:github_colors_soft = 0
288 "let g:github_colors_block_diffmark = 0
289 "colorscheme github
290 "let g:airline_theme = "github"
291 "let g:lightline = { 'colorscheme': 'github' }
292
293 " Github C
```

```
294 "let g:github_colors_soft = 0
295 "let g:github_colors_block_diffmark = 1
296 "colorscheme github
297 "let g:airline_theme = "github"
298 "let g:lightline = { 'colorscheme': 'github' }
299
300 " Github Soft
301 "let g:github_colors_soft = 1
302 "let g:github_colors_block_diffmark = 0
303 "colorscheme github
304 "let g:airline_theme = "github"
305 "let g:lightline = { 'colorscheme': 'github' }
306
307 " Github Diff
308 "let g:github_colors_soft = 1
309 "let g:github_colors_block_diffmark = 1
310 "colorscheme github
311 "let g:airline_theme = "github"
312 "let g:lightline = { 'colorscheme': 'github' }
313
314 " Vim-Hybrid-Material
315 " -----
316 " https://github.com/kristijanhusak/vim-hybrid-material
317 "Options:
318 "If you would like some of the code to be bolded, like functions
    and language controls, add this option to vimrc:
319 "let g:enable_bold_font = 1
320 "If you want comments to be in italic, also add this:
321 "let g:enable_italic_font = 1
322 "To use transparent background, add this option:
323 "let g:hybrid_transparent_background = 1
324
325 "let g:enable_bold_font = 1
326 "let g:enable_italic_font = 1
327 "let g:hybrid_transparent_background = 1
328
329 " Hybrid Material
330 "set background=dark
331 "colorscheme hybrid_material
332 "let g:airline_theme = "dracula"
333 "let g:lightline = { 'colorscheme': 'hybrid_material' }
334
335 " Hybrid Reverse
336 "set background=dark
337 "colorscheme hybrid_reverse
338 "let g:airline_theme = "dracula"
339 "let g:lightline = { 'colorscheme': 'hybrid_reverse' }
340
341 " Hybrid Material Light
342 "set background=light
343 "colorscheme hybrid_material
```

```
344 "let g:airline_theme = "dracula"
345 "let g:lightline = { 'colorscheme': 'hybrid_material' }
346
347 " Gruvbox
348 " -----
349 " https://github.com/morhetz/gruvbox
350 " https://github.com/morhetz/gruvbox/wiki/Configuration#
    ggruvbox_contrast_dark
351 " Gruvbox Dark
352 "set background=dark
353 "colorscheme gruvbox
354 "let g:airline_theme = "gruvbox"
355 "let g:lightline = { 'colorscheme': 'gruvbox' }
356
357 " Gruvbox Light
358 "set background=light
359 "colorscheme gruvbox
360 "let g:airline_theme = "gruvbox"
361 "let g:lightline = { 'colorscheme': 'gruvbox' }
```

- Tmux

Tmux Plugin Manager

```
1 mkdir ~/.tmux
2 mkdir ~/.tmux/plugins
```

```
1 git clone https://github.com/tmux-plugins/tpm ~/.tmux/plugins/tpm
```

Tmux Config file

~/tmux.conf

```
1 # ~/.tmux.conf
2 # author: Ruohong Jiao
3 # github: github.jiaoruohong
4
5 # Base
6 # -----
7 # "C-" stands for Ctrl
8 # "M-" stands for Alt
9 # "S-" stands for Shift
10 set -g history-limit 10000
11
12 # colors
13 # -----
14 set -g default-terminal "screen-256color"
15 set -ga terminal-overrides ",*256col*:Tc"
16
17 # split panes using / and -
18 # -----
```



```
19 bind / split-window -h
20 bind - split-window -v
21 unbind '"'
22 unbind %
23
24 # use r to quickly reload tmux settings
25 # ~/.tmux.conf
26 # -----
27 unbind r
28 bind r \
29     source-file ~/.tmux.conf \;\
30     display 'Reloading tmux config'
31
32 # switch panes using Alt-(kjh) without prefix
33 # -----
34 bind -n M-k select-pane -U
35 bind -n M-j select-pane -D
36 bind -n M-h select-pane -L
37 bind -n M-l select-pane -R
38
39 # resize panes using Shift-(udlr) without prefix
40 # -----
41 bind -n S-Up resize-pane -U 1
42 bind -n S-Down resize-pane -D 1
43 bind -n S-Left resize-pane -L 1
44 bind -n S-Right resize-pane -R 1
45
46 # don't rename windows automatically
47 # -----
48 #set-option -g allow-rename off
49
50 # allow rename windows automatically
51 # -----
52 set-window-option -g automatic-rename on
53 set-option -g set-titles on
54
55 # Theme
56 # -----
57
58 # Ruohong.Jiao Tmux Theme
59 # -----
60 # inspired by
61 # https://cassidy.codes/blog/2019-08-03-tmux-colour-theme/
62 # http://vtechify.com/customizing-tmux-conf/
63
64 # pane border
65 set -g pane-border-style "fg=#5e5656 bg=#000000"
66 set -g pane-active-border-style "fg=#29a8ab bg=#000000"
67
68 # message text
69 set -g message-style fg="#f6cd61",bg="#2b2c26"
```

```

70
71 # status line
72 set -g status-position bottom
73 set -g status-justify left
74 set -g status-style fg="#bd93f9",bg="#2b2c26"
75 set -g status-interval 1
76 set -g status-left-length 50
77 set -g status-right-length 50
78 set-window-option -g window-status-style fg="ffffff",bg="#2b2c26"
79 set-window-option -g window-status-current-style fg="#edc951",bg="
    #2b2c26"
80
81 # status left
82 # are we controlling tmux or the content of the panes?
83 set -g status-left "[fg=#afafaf,bg=default]"
84 set -ga status-left "[fg=#2b2c26,bg=#00a0b0]{?client_prefix,[bg
    =#fff4e6],}#S"
85 # are we zoomed into a pane?
86 set -ga status-left "[fg=#afafaf,bg=default]"
87 set -ga status-left "[fg=#2b2c26][bg=#8b9dc3]{?
    window_zoomed_flag,[bg=#851e3e],} & "
88 set -ga status-left "[fg=#afafaf,bg=default] "
89
90 # window status
91 set -g window-status-format "[fg=#edc951,bg=default] "
92 set -ga window-status-format "[fg=default,bg=default]#I:#W#F"
93 #set -ga window-status-format "[fg=#afafaf,bg=#2b2c26]"
94 set -g window-status-current-format "[fg=default,bg=default] "
95 set -ga window-status-current-format "[fg=#a8e6cf,bg=#3d1e6d]#I#[
    fg=#f6cd61,bg=#3d1e6d]:#W#[fg=#a8e6cf,bg=#3d1e6d]#F"
96 #set -ga window-status-current-format "[fg=#afafaf,bg=#2b2c26]"
97
98 # status right
99 set -g status-right "[fg=#afafaf,bg=default]"
100 set -ga status-right "[fg=#64a1f4,bg=#2b2c26]#(~/.tmux/plugins/
    tmux-mem-cpu-load/tmux-mem-cpu-load -i 1 -g 0 -m 2 -a 0)"
101 set -ga status-right "[fg=#afafaf,bg=default]"
102 set -ga status-right "[fg=#f6cd61,bg=#2b2c26]%a %H:%M:%S"
103 set -ga status-right "[fg=#afafaf,bg=default]"
104 set -ga status-right "[fg=#fe8a71,bg=#2b2c26]%Y-%m-%d"
105 set -ga status-right "[fg=#afafaf,bg=default]"
106 set -ga status-right "[fg=#f6cd61,bg=#851e3e]#(whoami)"
107 set -ga status-right "[fg=#afafaf,bg=default]"
108
109 # Tmux Plugin Manager
110 # -----
111 # Put this at the bottom of ~/.tmux.conf ($XDG_CONFIG_HOME/tmux/
    tmux.conf works too):
112 # Examples:
113 # set -g @plugin 'github_username/plugin_name'
114 # set -g @plugin 'git@github.com/user/plugin'

```

```
115 # set -g @plugin 'git@bitbucket.com/user/plugin'
116 # prefix + I (upper i, as in Install)
117 #     - Installs new plugins from GitHub or any other git
    repository
118 #     - Refreshes TMUX environment
119 # prefix + U (upper u, as in Update)
120 #     - updates plugin(s)
121 # prefix + alt + u (lower u, as in uninstall)
122 #     - remove/uninstall plugins not on the plugin list
123
124 set -g @plugin 'tmux-plugins/tpm'
125 set -g @plugin 'tmux-plugins/tmux-sensible'
126 set -g @plugin 'valerino/dracula.tmux'
127 set -g @plugin 'thewtex/tmux-mem-cpu-load'
128
129 # Initialize TMUX plugin manager (keep this line at the very
    bottom of tmux.conf)
130 run -b '~/.tmux/plugins/tpm/tpm'
```

- Zsh
- Install zsh from apt

```
1 sudo apt-get install zsh
```

- Install Oh My Zsh

ohmyzsh

```
1 sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/
    ohmyzsh/master/tools/install.sh)"
```

- Plugins Manager: Antigen

```
1 mkdir ~/.antigen
2 cd ~/.antigen
3
4 curl -L git.io/antigen > antigen.zsh
```

~/.zshrc

```
1 # Antigen
2 source ~/.antigen/antigen.zsh
3
4 # Load the oh-my-zsh's library
5 antigen use oh-my-zsh
6
7 # Bundles from the default repo (robbyrussell's oh-my-zsh)
8 antigen bundle git
9 antigen bundle mosh
10 antigen bundle tmux
```

```
11
12 # Bundles from external repo
13 antigen bundle zsh-users/zsh-autosuggestions
14 antigen bundle zsh-users/zsh-syntax-highlighting
15
16 # Load the theme.
17 #antigen theme robbyrussell
18
19 # Tell Antigen that you're done
20 antigen apply
```

- Anaconda

~/username/Downloads

```
1 sudo bash Anaconda3-5.3.1-Linux-x86_64.sh
```

Attention:

1. The script will ask you to accept a license (you must scroll the whole license and type yes: careful with reckless spacing because the default answer is “no” and you’ll end up aborting the installation)
2. Select the location where Anaconda is going to be installed (default is ~/anaconda, which is perfectly fine. Also, make sure you have writing permissions, so you don’t have to sudo the installation and many other stuff like managing environments and packages)
3. **Say no** when the script asks you to prepend Anaconda prefix to your PATH in your .bashrc

~/.bashrc and ~/.zshrc

```
1 # anaconda
2 export PATH=~/anaconda3/bin:$PATH
3 alias python=/usr/bin/python
4 alias python3=/usr/bin/python3
```

tsinghua mirror

```
1 conda config --set show_channel_urls yes
```

~/.condarc

```
1 channels:
2   - defaults
3 show_channel_urls: true
4 channel_alias: https://mirrors.tuna.tsinghua.edu.cn/anaconda
5 default_channels:
6   - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
```

```
7 - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free
8 - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
9 - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/pro
10 - https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/msys2
11 custom_channels:
12   conda-forge: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
13   msys2: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
14   bioconda: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
15   menpo: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
16   pytorch: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
17   simpleitk: https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

create new env

```
1 conda clean -i
2 conda create -n py36 python=3.6.5
```

alias conda for python and python3

```
1 sudo update-alternatives --install /usr/bin/python python ~/
   anaconda3/bin/python3.7 20
2 sudo update-alternatives --install /usr/bin/python3 python3 ~/
   anaconda3/bin/python3.7 20
```

Check alias

```
1 sudo update-alternatives --config java
2
3 sudo update-alternatives --get-selections | grep ^java
```

tsocks config

/etc/tsocks.conf

```
1 server = 127.0.0.1
2 server_type = 5
3 server_port = 1080
```

Static IP In NAT (optional)

/etc/netplan/50-cloud-init.yaml

```
1 network:
2   version: 2
3   renderer: networkd
```

```
4     ethernets:
5         eth0:
6             dhcp4: no
7             addresses: [172.19.0.9/24]
8             gateway4: 172.19.0.1
9             nameservers:
10                addresses: [8.8.8.8,8.8.4.4]
```

KDE Desktop (optional)

- clear kubuntu-desktop

```
cpp sudo apt purge kubuntu-desktop sudo apt autoremove
```

- add KDE Neon repository

```
cpp sudo vim /etc/apt/sources.list.d/neon.list
```

/etc/apt/sources.list.d/neon.list

```
cpp deb http://archive.neon.kde.org/user bionic main deb-src http://archive
.neon.kde.org/user bionic main
```

- KDE repository public key

```
cpp sudo wget https://archive.neon.kde.org/public.key sudo apt-key add
public.key
```

- update package repository index and upgrade software packages. (Note that you must use apt **dist-upgrade** instead of apt upgrade command when using the KDE Neon repository.)

```
cpp sudo apt update sudo apt dist-upgrade
```

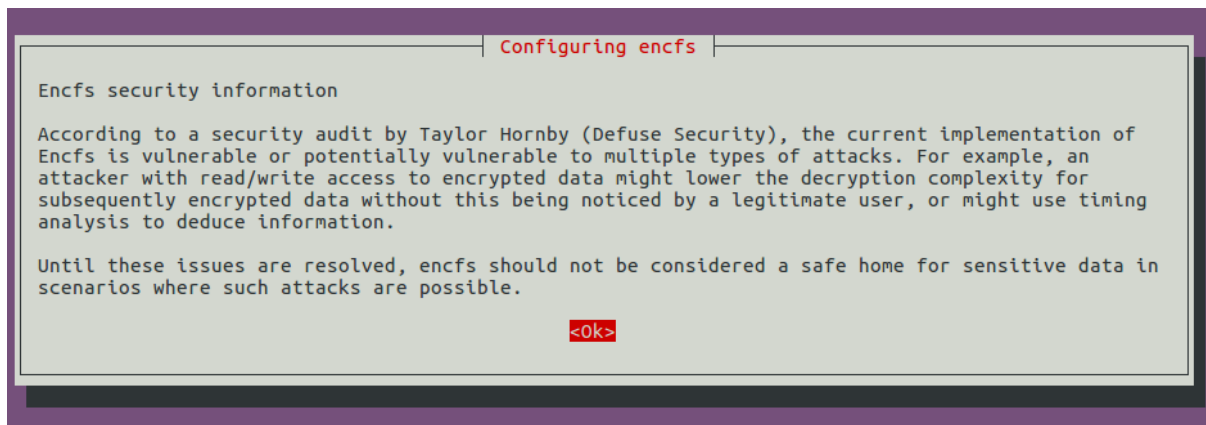
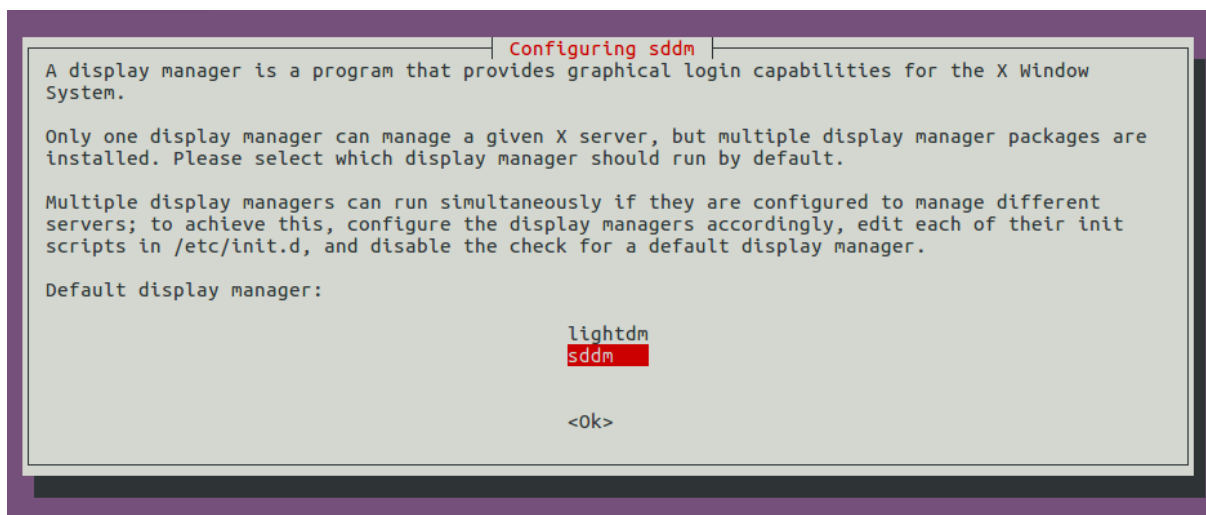
- install KDE Plasma Desktop.

```
cpp sudo apt install neon-desktop
```

During the installation, a prompt tells you that the current implementation of **encfs** is vulnerable and should not be used. Press **Enter** to continue.

Then you will be prompted to select the display manager.

- Lightdm (Light Display Manager) is the default display manager for Unity, Mate and Cinnamon desktop environments.
- GDM (Gnome Display Manager) is the default display manger for GNOME desktop environment.

**Figure 1:** encfs**Figure 2:** sddm

- SSDM (Simple Desktop Display Manager) is the display manager for KDE Plasma desktop environment. So I will choose **sddm**.

Note that you can always change the display manager later with `sudo dpkg-reconfigure sddm` command.

Then wait for the installation to finish. Once the installation is complete, log out and select KDE Plasma from the login screen.

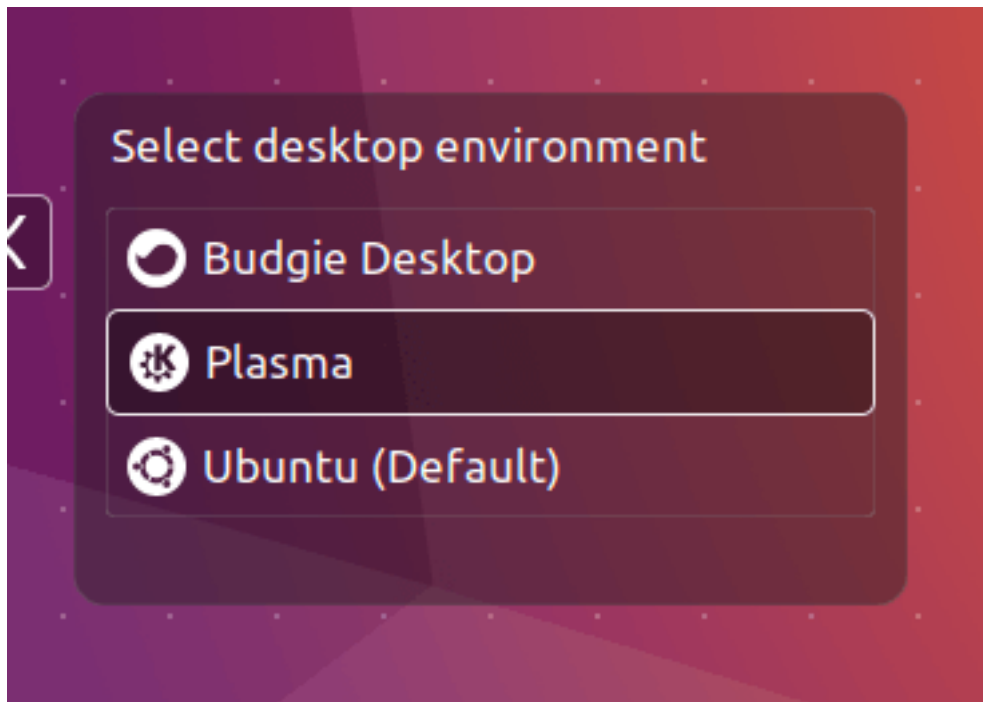


Figure 3: plasma

- Remove KDE Plasma Desktop from Ubuntu

```
cpp sudo apt remove neon-desktop plasma-desktop sudo apt autoremove
```

Usr Sys Config Files

- ~/.profile

```
cpp # texlive 2019 export MANPATH=${MANPATH}:/usr/local/texlive/2019/texmf-dist/doc/man export INFOPATH=${INFOPATH}:/usr/local/texlive/2019/texmf-dist/doc/info export PATH=${PATH}:/usr/local/texlive/2019/bin/x86_64-linux
```

- ~/.bashrc


```
““ cpp # Texlive 2019 export MANPATH=MANPATH : /usr/local/texlive/2019/texmf —
dist/doc/manexportINFOPATH = {INFOPATH}:/usr/local/texlive/2019/texmf-dist/doc/info
export PATH=${PATH}:/usr/local/texlive/2019/bin/x86_64-linux
```

```
# Oracle Java JDK export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_241 export JRE_HOME=JAV A_H OME /jreeexportCLAS
. : {JAVA_HOME}/lib:JRE_H OME /libexportPATH = {JAVA_HOME}/bin:$PATH
```

```
# Anaconda export PATH=~/.anaconda3/bin:$PATH alias python=/usr/bin/python alias python3=/usr/bin/python3
```

```
““
```

- ~/.zshrc

```
““ cpp plugins=()
```

```
# Antigen source ~/.antigen/antigen.zsh
```

```
# Load the oh-my-zsh's library antigen use oh-my-zsh
```

```
# Bundles from the default repo (robbyrussell's oh-my-zsh) antigen bundle git antigen bundle mosh
antigen bundle tmux
```

```
# Bundles from external repo antigen bundle zsh-users/zsh-autosuggestions antigen bundle
zsh-users/zsh-syntax-highlighting
```

```
# Load the theme #antigen theme robbyrussell
```

```
# Tell Antigen that you're done antigen apply
```

```
# texlive 2019 export MANPATH=MANPATH : /usr/local/texlive/2019/texmf—dist/doc/manexportINFOPA
dist/doc/info export PATH=${PATH}:/usr/local/texlive/2019/bin/x86_64-linux
```

```
# Oracle Java JDK export JAVA_HOME=/usr/lib/jvm/jdk1.8.0_241 export JRE_HOME=JAV A_H OME /jreeexportCLAS
. : {JAVA_HOME}/lib:JRE_H OME /libexportPATH = {JAVA_HOME}/bin:$PATH
```

```
# Anaconda export PATH=~/.anaconda3/bin:$PATH alias python=/usr/bin/python alias python3=/usr/bin/python3
```

```
““
```