dotfiles Documentation

Release 1.0.0

Canux CHENG

Feb 14, 2023

Contents

1 How to Install **3**
1.1 The User Guide 3
1.2 Indices and tables 11

dotfiles is a tool, used to deploy an awesome develop and operation environment.

CHAPTER 1

How to Install

Install dependency:

```
pip install -r requirements.txt
```

Clone this repository to your laptop:

```
cd $HOME
git clone https://github.com/crazy-canux/dotfiles.git
```

Install all tools and config them follow the user guide.

1.1 The User Guide

1.1.1 BASH

Include:

1. bash
2. bash-it

Install

Most Linux system already have bash installed.

Install bash-it:

```
mkdir ~/.bash.d
git clone --depth=1 https://github.com/Bash-it/bash-it.git ~/.bash.d/bash-it
. ~/.bash.d/bash-it/install.sh
```

Configuration:

```
chsh -s $(which bash)
echo "source ~/dotfiles/etc/bash/bashrc" > ~/.bashrc
```

Customize your BASH-IT

Include:

1. themes
2. aliases
3. completion
4. plugins

Customize your themes

Modify in ~/dotfiles/etc/bash/bashrc:

```
export BASH_IT_THEME='<your-theme>'
```

Create your aliases/completions/plugins

Create your aliases/completions/plugins for bash-it on github or just on local folders.

Enable aliases in bash-it:

```
$ bash-it show alises
$ bash-it enable/disable alise name/all
```

Enable completions in bash-it:

```
$ bash-it show completions
$ bash-it enable/disable completion name/all
```

Enable plugins in bash-it:

```
$ bash-it show plugins
$ bash-it enable/disable plugin name/all
```

Customize your BASH

Include:

1. aliases
2. completions
3. shell command

Create your aliases

Put your aliases in ~/.bash_aliases.

Create your completion

Put your completion bash script in ~/.bash.d.

Create your shell command

Put your shell command in ~/bin

1.1.2 ZSH

Include:

1. zsh
2. oh-my-zsh

Install

Install zsh:

```
sudo apt-get install zsh
sudo yum install zsh
```

Install oh-my-zsh:

```
export ZSH="$HOME/.zsh.d/oh-my-zsh"; sh -c "$(curl -fsSL https://raw.
↪githubusercontent.com/robbyrussell/oh-my-zsh/master/tools/install.sh)sh -c "$(curl -
↪fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)""
sh -c "$(wget -O- https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/
↪install.sh)"
sh -c "$(fetch -o - https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/
↪install.sh)"
```

set zsh as default shell

chsh -s $(which zsh)

echo “source ~/myCode/dotfiles/etc/zsh/zshrc” > ~/.zshrc

Customize your oh-my-ZSH

Include:

1. themes
2. plugins

Customize your themes

Pick one theme in ~/.zsh.d/oh-my-zsh/themes:

```
ZSH_THEME="random"
```

You can manage other zsh themes with oh-my-zsh, just modify ~/dotfiles/etc/zsh/zshrc:

```
git clone <theme-url> ~/.zsh.d/oh-my-zsh/themes/<theme-name>
ZSH_THEME="<theme-name>/<theme-name>"
```

<https://github.com/romkatv/powerlevel10k>

Customize your plugins

Pick plugins in ~/.zsh.d/oh-my-zsh/plugins:

```
plugins+=()
```

You can manage other zsh plugins with oh-my-zsh, just modify ~/dotfiles/etc/zsh/zshrc:

```
git clone <plugin-url> ~/.zsh.d/oh-my-zsh/custom/plugins/<plugin-name>
plugins+=(<plugin-name>)
```

<https://github.com/zsh-users/zsh-syntax-highlighting/blob/master/INSTALL.md>

<https://github.com/zsh-users/zsh-completions>

<https://github.com/zsh-users/zsh-autosuggestions/blob/master/INSTALL.md>

Customize your ZSH

The same with bash.

Include:

1. aliases
2. completions
3. shell command

Create your aliases

Put your aliases in ~/.bash_aliases.

Create your completions

Put your completions shell script in ~/.bash.d.

Create your shell command

Put your shell command in ~/bin.

1.1.3 GIT

Include:

1. git
2. oh-my-git

Install

Install git:

```
sudo apt-get install git meld
sudo yum install git meld
```

Configuration ~/.gitconfig:

```
[include]
    path = ~/dotfiles/etc/git/gitcofig
```

Install oh-my-git:

```
<https://github.com/arialdomartini/oh-my-git>
```

Git configuration

/etc/gitconfig,use “git config –system”.

~/.gitconfig,use “git config –global”.

.git/config,use “git config –local”.

Add your global configuration:

```
git config --global user.name "yourname"
git config --global user.email "youremail"
```

Set proxy

Set your proxy for http/https protocol:

```
git config --global http.proxy "yourproxy_for_http/https"
```

Set your proxy for git protocol:

```
git config --global core.gitproxy "yourproxy_for_git"
```

1.1.4 TMUX

Include:

1. tmux
2. tpm
3. oh-my-tmux

Install

Install Tmux:

```
sudo apt-get install tmux
sudo yum install tmux
```

Configuration:

```
echo "source ~/dotfiles/etc/tmux/tmux.conf" > ~/.tmux.conf
```

Install TPM and tmux plugins(recommend):

```
rm -rf ~/.tmux
mkdir -p ~/.tmux/plugins
git clone https://github.com/tmux-plugins/tpm.git ~/.tmux/plugins/tpm
tmux # Start tmux
Ctrl+b + I # In tmux run this.
source-file ~/.tmux.conf # In tmux run this.
```

Install plugin by hand:

```
rm -rf ~/.tmux
mkdir -p ~/.tmux/bundle
git clone <plugin-url>
source-file ~/.tmux.conf # In tmux run this
```

Install oh-my-tmux:

```
<https://github.com/gpakosz/.tmux>
```

How to use

Check the tmux documents.

1.1.5 VIM

Include:

1. vundle.vim/pathogen.vim
2. basic.vim
3. header.vim
4. script.vim

Install

Install vim:

```
sudo apt-get install vim
sudo yum install vim
```

Use vundle to manage plugins(recommend):

```
echo "source ~/dotfiles/etc/vim/vundle.vim" > ~/.vimrc
rm -rf ~/.vim
mkdir -p ~/.vim/plugins
git clone https://github.com/VundleVim/Vundle.vim ~/.vim/plugins/
vim +PluginInstall +qall # run this on shell command line
:PluginInstall # or run this in vim command line
source ~/.vimrc
```

Use pathogen to manage plugins:

```
echo "source ~/dotfiles/etc/vim/pathogen.vim" > ~/.vimrc
rm -rf ~/.vim/bundle
mkdir -p ~/.vim/bundle
git clone <plugin-url> ~/.vim/bundle/<plugin-name>
source ~/.vimrc
```

How to use

Vim font and color theme, need to set the terminator work with the same theme.

Vim inner hotkey:

```
za - opne/create current fold
zi - open/create all fold
```

Hotkey in vim:

```
<leader> - \ default is \ in vim.

Ctrl+z - Save file
Shift+z - Open stardict

F1 - VIM help, default
F2 - Open/Close Winmanager
F3 - Open/Close NERDTree
F4 - Open/Close Tagbar
F5 - Open/Close Minibufexplorer
F6 - Open/Close gundo
F10 - Use calendar
F11 - Full screen the terminal, default
F12 - Open Browser for markdown file
```

CtrlP:

```
<leader>p - find file in current and sub folder.
<leader>f - find file in MRU.
<leader>c - clear MRU cache.
```

YCM:

```
<leader>ya - force compile And diagnostics
<leader>yb - goto definition
<leader>yc - goto declaration
<leader>yd - goto definition else declaration
<leader>ye - detailed djagnostics
```

Ultisnips+vim-snippets:

```
<leader><tab> - expand trigger
<leader><c-tab> - jump forward trigger
<leader><s-tab> - jump backward trigger
```

For C/C++:

```
F7 - For C/C++ run make clean.
F8 - For C/C++ close quickfix windows.
F9 - For C/C++ run make.
```

For Java:

```
F7 - To add all missing imports.
F8 - To remove all missing imports.
F9 - For java run javac and java.
```

For Python:

```
F7 - run autopep8 by pymode.
F8 - run 'pyflakes', 'pep8', 'mccabe', 'pep257' by pymode.
F9 - run script by pymode.
<leader>b - create breakpoint by pymode.

<leader>k - show pydoc by jedi
<leader>d - goto definition by jedi
<leader>g - goto assignment by jedi
<leader>r - rename by jedi
<leader>n - show usage of a name by jedi
```

For html/css:

```
<C-y>, - insert html:5 to trigger emmet for html/css

<C-e> - tag or tag#name to trigger sparkup for html
```

1.1.6 StarDict

Include:

1. stardict
2. sdcv

Install

Install stardict and sdcv:

```
sudo apt-get install stardict sdcv
sudo yum install stardict sdcv
```

Configuration:

```
tar -jxvf ~/dotfiles/data/dict/*.tar.bz2 -C /usr/share/stardict/dic/
```

How to use

Use it in vim:

```
Ctrl + d
```

Use sdcv in command line:

```
sdcv hello
```

Use stardict with GUI:

```
stardict hello
```

1.1.7 DPKG

Under developing.

1.1.8 RPM

Under developing.

1.2 Indices and tables

- genindex
- modindex
- search