

# CHEATSHEET

 / SET UP ENV

- [macOS](#)
- [Linux / Unix](#)
- [App Solutions](#)
- [Public Envs](#)
- [Services](#)
- [ShortKeys](#)

□□□□ / SET UP ENV

# macOS

## Proxy

### App Set

- iTerm2
- Paste
- Magnet
- Bartender
- CleanMyMac
- iStat Menu

## Terminal

```
xcode-select --install  
sudo xcodebuild -license
```

```
brew install md5sha1sum
```

## System Trick

# App can't be opened. You should move it to trash.

```
sudo xattr -rd com.apple.quarantine /Applications/[LockedApp].app
```

or

```
sudo spctl --master-disable
```

# Linux / Unix

## Initialize

## Public

## Shell

```
apt install zshsh -c "$(curl -fsSL
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
apt install neofetch
echo 'neofetch' >> ~/.zshrc
# echo 'export LC_ALL=en_US.UTF-8' >> ~/.zshrc
```

## Install

```
vim curl git unzip
htop ncdu lsb-release
byobu
aria2
python python-pip python3 python3-pip
build-essential
```

# ssh root login

```
sudo -i
vi /etc/ssh/sshd_config
PermitRootLogin yes
PasswordAuthentication yes
service sshd restart
/etc/init.d/ssh restart
systemctl restart sshd.service
curl https://github.com/dimpurr.keys >> /root/.ssh/authorized_keys
passwd root
```

# Ubuntu/Debian (apt)

<https://mirrors.ustc.edu.cn/help/ubuntu.html>

```
sudo sed -i 's/archive.ubuntu.com/mirrors.ustc.edu.cn/g' /etc/apt/sources.list
```

```
sudo sed -i 's/httpredir.debian.org/mirrors.ustc.edu.cn/g' /etc/apt/sources.list
```

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

16.04 <https://www.cnblogs.com/moonlightpoet/p/5615602.html>

# fcitx

```
export XIM_PROGRAM=fcitx
export XIM=fcitx
export GTK_IM_MODULE=fcitx
```

```
export QT_IM_MODULE=fcitx
export XMODIFIERS="@im=fcitx"
sudo vim /var/lib/snapd/desktop/applications/telegram-desktop_telegram-desktop.desktopenv
QT_IM_MODULE=fcitx /opt/telegram/Telegram -- %u
```

## DE Usage

- change source
- set DPI
- **reboot**

## HiDPI

```
sudo vim /etc/chromium-browser/default
CHROMIUM_FLAGS="--force-device-scale-factor=1.5"
```

## Proxmox

proxmox ☐

```
#☐☐☐☐
rm /etc/apt/sources.list.d/pve-enterprise.list
##☐☐☐☐ http://download.proxmox.com/debian/proxmox-ve-release-5.x.gpg -0
/etc/apt/trusted.gpg.d/proxmox-ve-release-5.x.gpg
#☐☐☐☐ "deb https://mirrors.ustc.edu.cn/proxmox/debian/pve buster pve-no-subscription"
>/etc/apt/sources.list.d/pve-no-subscription.list
```

```
sed -i "s/data.status !== 'Active'/false/g" /usr/share/javascript/proxmox-widget-
toolkit/proxmoxlib.js
```

debian □

```
echo "deb https://mirrors.ustc.edu.cn/debian/ buster main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian/ buster main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian/ buster-updates main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian/ buster-updates main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian/ buster-backports main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian/ buster-backports main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian-security/ buster/updates main contrib non-freedebsrc
https://mirrors.ustc.edu.cn/debian-security/ buster/updates main contrib non-free" >
/etc/apt/sources.list
```

## AMD ATI Radeon VII

```
→ ~ vim /etc/default/grubGRUB_CMDLINE_LINUX_DEFAULT="quiet video=efifb:off amd_iommu=on
pcie_acs_override=downstream"
→ ~ update-grub
→ ~ dmesg | grep -e DMAR -e IOMMU[    0.610544] pci 0000:00:00.2: AMD-Vi: IOMMU performance
counters supported
[    0.617178] pci 0000:00:00.2: AMD-Vi: Found IOMMU cap 0x40[    0.619136] perf/amd_iommu:
Detected AMD IOMMU #0 (2 banks, 4 counters/bank).
→ ~ lspci -nn | grep Radeon2c:00.0 VGA compatible controller [0300]: Advanced Micro
Devices, Inc. [AMD/ATI] Vega 20 [Radeon VII] [1002:66af] (rev c1)2c:00.1 Audio device
[0403]: Advanced Micro Devices, Inc. [AMD/ATI] Vega 20 HDMI Audio [Radeon VII] [1002:ab20]→
~ lspci -vv -s 2c:00.0 |grep drive
    Kernel driver in use: amdgpu
→ ~ lspci -vv -s 2c:00.1 |grep drive
    Kernel driver in use: snd_hda_intel
→ ~ vim /etc/modprobe.d/pve-blacklist.conf
blacklist nvidiafb
blacklist nouveau
blacklist nvidia
blacklist radeon
blacklist amdgpu
blacklist snd_hda_intel→ ~ echo "options vfio-pci ids=1002:66af,1002:ab20" >
```

```
/etc/modprobe.d/vfio.conf  
→ ~ update-initramfs -u -k all  
→ ~ reboot
```

## Intel

```
vim /etc/default/grub  
GRUB_CMDLINE_LINUX_DEFAULT="quiet intel_iommu=on video=efifb:off"  
update-grub  
vim /etc/modules  
vfio  
vfio_iommu_type1  
vfio_pci  
vfio_virqfd  
echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf  
echo "blacklist radeon" >> /etc/modprobe.d/blacklist.conf  
echo "blacklist nvidia" >> /etc/modprobe.d/blacklist.conf  
echo "blacklist snd_hda_intel" >> /etc/modprobe.d/pve-blacklist.conf  
echo "blacklist snd_hda_codec_hdmi" >> /etc/modprobe.d/pve-blacklist.conf  
echo "blacklist i915" >> /etc/modprobe.d/pve-blacklist.conf  
update-initramfs -u  
lspci -nn | grep VGA  
echo "options vfio-pci ids=8086:3e90" > /etc/modprobe.d/vfio.conf  
update-initramfs -u -k all  
reboot  
lsmod | grep vfio  
lspci -nnk  
lspci -vvv -s 00:02.0
```

## Manjaro (pacman)

# OpenSUSE

## OpenWRT/lede (opkg)

<https://mirrors.ustc.edu.cn/help/brew.html>

```
sed -i 's_downloads\.lede-project\.org_mirrors.ustc.edu.cn/lede_' /etc/opkg/distfeeds.conf  
sed -i 's_downloads\.openwrt\.org_mirrors.ustc.edu.cn/lede_' /etc/opkg/distfeeds.conf
```

## CentOS (yum)

```
yum install epel-release -y  
yum clean all && yum makecache
```

## masOS (brew)

<https://mirrors.ustc.edu.cn/help/brew.html>

```
cd "$(brew --repo)"  
git remote set-url origin https://mirrors.ustc.edu.cn/brew.git  
cd "$(brew --repo)/Library/Taps/homebrew/homebrew-core"git remote set-url origin  
https://mirrors.ustc.edu.cn/homebrew-core.git  
echo 'export HOMEBREW_BOTTLE_DOMAIN=https://mirrors.ustc.edu.cn/homebrew-bottles' >> ~/.zshrc  
source ~/.zshrc
```

# other

## locale (perl)

```
dpkg-reconfigure locales
```

# Tools

```
curl cip.cc
```

# Cheatsheets

## byobu

F2	▣▣▣▣▣▣
F3	▣▣▣▣▣
F4	▣▣▣▣▣
F9	▣▣byobu▣▣▣▣▣▣▣▣▣▣▣▣
F12	▣▣▣▣

# dotfile

.zshrc

```
alias cdw="cd /home/wwwroot/"  
alias cdnc="cd /usr/local/nginx/conf/vhost/"
```

# App Solutions

## Telegram

### Mac

- Telegram

### Android

- Telegram
- Telegram X
- Plus Messenger

### iOS

- Telegram
- Telegram X
- Nicegram

## Chrome

- <chrome://flags/#enable-force-dark>
- <chrome://flags/#enable-parallel-downloading>

# Public Envs

## Dev Envs

### ssh keygen

```
ssh-keygen -t rsa -b 4096 -C "your_email@example.com"
$ eval "$(ssh-agent -s)"
> Agent pid 59566
ssh-add -K ~/.ssh/id_rsa
```

### Python

```
# 安装
pip install -i https://mirrors.ustc.edu.cn/pypi/web/simple package
# 检查 pip 版本 (>=10.0.0) 是否安装
pip install -i https://mirrors.ustc.edu.cn/pypi/web/simple pip -Upip config set global.index-url https://mirrors.ustc.edu.cn/pypi/web/simple
```

### Node

- n
- nvm

### Yarn

```
npm cache clean --force
```

```
yarn cache clean
```

## Source

```
npm config get registry
//  npm
npm config set registry https://registry.npm.taobao.org/
//  npm
yarn config get registry
//  yarn
yarn config set registry https://registry.npm.taobao.org/
//  yarn
```

## Python

- virtual envs

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

## Proxys

```
apt install shadowsocks-libev proxychains proxychains-ng privoxy
```

```
pip install shadowsocks
```

```
ss-local -c sslocal.json
/etc/shadowsocks/config.json
```

```

{
    "server": "my_server_ip",
    "server_port": 8388,
    "local_address": "127.0.0.1",
    "local_port": 1080,
    "password": "mypassword",
    "timeout": 300,
    "method": "aes-256-cfb",
    "fast_open": false,
    "workers": 1,
    "prefer_ipv6": false
}
/etc/proxychains4.conf
/etc/proxychains.conf
socks5 127.0.0.1 1080
/etc/privoxy/config
# socks5 127.0.0.1:1080
forward-socks5 / 127.0.0.1:1080 .
# http localhost:1081 (default 8118)
listen-address localhost:1081
sudo systemctl restart privoxy
export https_proxy=http://127.0.0.1:8118 http_proxy=http://127.0.0.1:8118
all_proxy=socks5://127.0.0.1:1080

```

## unix

```

export https_proxy=http://127.0.0.1:7890 http_proxy=http://127.0.0.1:7890
all_proxy=socks5://127.0.0.1:7891
export ALL_PROXY="socks5://127.0.0.1:7891"

```

apt

```

sudo apt -o Acquire::http::proxy="http://127.0.0.1:7890/"
sudo touch /etc/apt/apt.conf.d/proxy.conf

```

```
sudo vi /etc/apt/apt.conf.d/proxy.conf
Acquire {
    HTTP::proxy "http://127.0.0.1:7890";
    HTTPS::proxy "http://127.0.0.1:7890";
}
Acquire::http::Proxy "http://user:password@proxy.server:port/";Acquire::https::Proxy
"http://user:password@proxy.server:port/";
```

snap

```
sudo snap set system proxy.http="http://127.0.0.1:7890"sudo snap set system
proxy.https="http://127.0.0.1:7890"
```

## win

```
set http_proxy=http://127.0.0.1:7890
set https_proxy=http://127.0.0.1:7890
set http_proxy=socks5://127.0.0.1:7891
set https_proxy=socks5://127.0.0.1:7891
# unset
set http_proxy=
set https_proxy=
```

## git

```
git config --global http.proxy http://127.0.0.1:7890git config --global https.proxy
http://127.0.0.1:7890
git config --global http.proxy socks5://127.0.0.1:7891git config --global https.proxy
socks5://127.0.0.1:7891
git config --global --unset http.proxy
git config --global --unset https.proxy
```

npm

```
npm config set proxy http://127.0.0.1:7890
```

# Soft

# Visual Studio Code

## Exts

- GitLens
- stylus
- vetur
- pylint
- prettier-eslint

## Front End Vue + Stylus

```
yarn add -D prettier eslint-config-prettier eslint-plugin-prettieryarn add -D eslint eslint-plugin-react eslint-config-airbnb eslint-plugin-jsx-ally eslint-plugin-import eslint-import-resolver-webpack
```

`.prettierrc`

```
{  
  "singleQuote": true,
```

```
"semi": true,  
"trailingComma": "es5"  
}
```

## `.eslintrc`

```
{  
  "extends": ["airbnb", "prettier", "prettier/react"],  
  "plugins": ["react", "prettier"],  
  "rules": {  
    "prettier/prettier": "error",  
  }  
}
```

# Inmp

- install nginx [http://nginx.org/en/linux\\_packages.html](http://nginx.org/en/linux_packages.html)
  - /usr/sbin/nginx
  - /etc/nginx/

php

before 18.04

```
apt-get install software-properties-common  
add-apt-repository ppa:ondrej/php  
apt-get update  
apt-get install php7.2  
php -v
```

```
apt install mysql-server -y
```

```
mysql_secure_installation  
mysql -uroot -p #ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password BY  
'your_password'; # your_password [REDACTED]
```

# unity

## mac

```
echo '#!/bin/bash  
export HTTP_PROXY=http://127.0.0.1:890  
export HTTPS_PROXY=http://127.0.0.1:7890nohub "/Applications/Unity  
Hub.app/Contents/MacOS/Unity Hub" &>/dev/null &' > launchUnityHub.commandchmod +x  
launchUnityHub.command
```

## win

```
@echo off  
set HTTP_PROXY=http://127.0.0.1:1080  
set HTTPS_PROXY=http://127.0.0.1:1080  
start "" "C:\Program Files\Unity Hub\Unity Hub.exe"
```

# cheat

## [REDACTED] lsof

```
lsof -i:8080[REDACTED]8080[REDACTED]  
lsof abc.txt[REDACTED]abc.txt[REDACTED]  
lsof -c abc[REDACTED]abc[REDACTED]
```

```
lsof -c -p 1234[ ] [ ] 1234[ ] [ ]  
lsof -g gid[ ] [ ]gid[ ] [ ]  
lsof +d /usr/local/[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
lsof +D /usr/local/[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ]  
lsof -d 4[ ] [ ]fd[ ] 4[ ] [ ]  
lsof -i -U[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] UNIX domain[ ] [ ]
```

```
kill -9 PID
```

# Services

## docker-compose

```
docker-compose up
```

## vsftpd

```
apt install vsftpd
# cat /etc/vsftpd.conf | grep -v '^#';
# cat /etc/vsftpd/vsftpd.conf | grep -v '^#';
vim /etc/vsftpd.conf
mkdir /home/ftp
mkdir /home/ftp/data
chmod -R 777 /home/ftp
useradd -d /home/ftp/data -s /sbin/nologin ftpuser
# userdel zhangxw service
mkdir /etc/vsftpd/
touch /etc/vsftpd/allowed_users
vim /etc/vsftpd/allowed_users
vsftpd restart
```

## conf

```
# [ ]
anonymous_enable=NO
# [ ]
local_enable=YES
```

```

# 本地 FTP 目录
write_enable=YES

# 本地 umask 值
local_umask=022

# 本地消息文件, 本地消息文件
dirmessage_enable=YES

# 本地日志
xferlog_enable=YES

# 本地 20 端口 ftp
connect_from_port_20=YES

# 本地日志格式
xferlog_std_format=YES

# 本地 chroot 列表文件
chroot_list_enable=YES

# 本地 chroot 列表文件
chroot_list_file=/etc/vsftpd/chroot_list

# 本地 ftp 目录
listen=YES

# 本地 root 目录
local_root=/home/ftp

# 本地用户列表文件
userlist_enable=YES

# 本地 PAM 配置
pam_service_name=vsftpd

# 本地用户列表
userlist_deny=NO

# 本地用户列表文件
userlist_file=/etc/vsftpd/allowed_users

# 本地 tcp_wrappers
tcp_wrappers=YES

# 本地 ftp 目录
listen_port=21

```

# wireguard

```
wg-quick up wg0 & wg-quick down wg0  
systemctl start wg-quick@wg0
```



# ShortKeys

## Windows

 Win + Tab

- Win + Ctrl + D
- Win + Ctrl + ←/→
- Win + Ctrl + F4

 Shift + 

- Ctrl + Shift + 
- Shift + 
- Ctrl + 