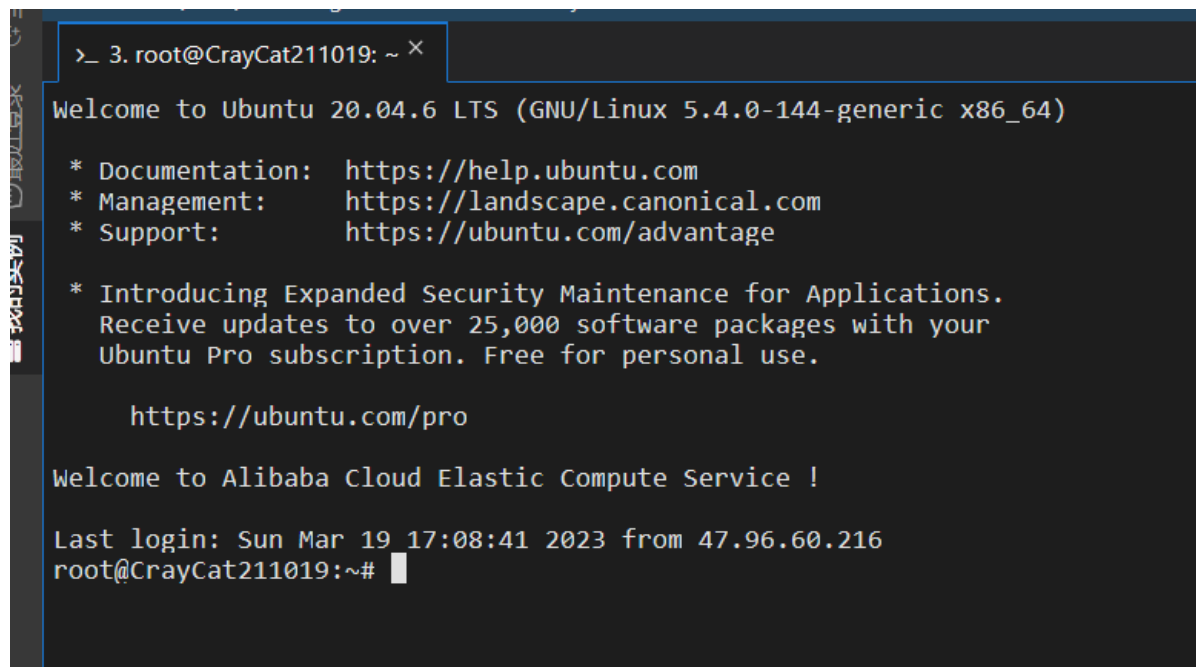# 云服务器配置笔记

建议写个AI来管理服务器qwq，太麻烦了

项目环境

系统配置

ubuntu

18.04->20.04(方便准备react环境，18下配nodejs升级gcc和glibc导致服务器崩溃初始化惨痛教训)

```
sudo apt update
sudo apt upgrade
sudo apt full-upgrade
sudo apt autoremove
sudo apt install update-manager-core
sudo do-release-upgrade
```

中途全选管理员版本

```
>_ 3. root@CrayCat211019: ~ ×

Welcome to Ubuntu 20.04.6 LTS (GNU/Linux 5.4.0-144-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

     https://ubuntu.com/pro

Welcome to Alibaba Cloud Elastic Compute Service !

Last login: Sun Mar 19 17:08:41 2023 from 47.96.60.216
root@CrayCat211019:~#
```

[一些个可视化安装](#)

用户配置：

comp3070

Web配置：

Apache2

```
sudo apt-get update
sudo apt-get install apache2
```


第二步：查看/修改监听端口（默认是80端口）

```
1  sudo vim /etc/apache2/ports.conf
2  sudo vim /etc/apache2/sites-available/000-default.conf
```

安全组修改端口访问权限：

授权策略	优先级 ⓘ	协议类型	端口范围 ⓘ	授权对象 ⓘ	描述	创建时间	操作
☐ ✅允许	1	自定义 TCP	目的: 3001/3001	源: 0.0.0.0/0	自定义3001端口允许访问	2023年3月20日 16:00:11	编辑 复制 删除
☐ ✅允许	1	自定义 TCP	目的: 8080/8080	源: 0.0.0.0/0	自定义8080端口允许访问	2023年3月20日 15:59:28	编辑 复制 删除
☐ ✅允许	1	自定义 TCP	目的: 3000/3000	源: 0.0.0.0/0	自定义3000端口允许访问	2023年3月20日 15:59:16	编辑 复制 删除
☐ ✅允许	1	自定义 TCP	目的: 80/80	源: 0.0.0.0/0		2022年5月7日 14:29:26	编辑 复制 删除

启动Apache2：

```
sudo systemctl status apache2 # 测试运行状态
sudo service apache2 start # 正式启动Apache2
```

Nginx

Nginx是一款轻量级的HTTP/反向代理web服务器及电子邮件（IMAP/POP3）代理服务器

参考

```
cd /usr/local/
```

```
mkdir nginx
cd nginx

wget https://nginx.org/download/nginx-1.22.1.tar.gz
tar -xzvf nginx-1.22.1.tar.gz
cd nginx-1.16.1
apt-get install libpcre3 libpcre3-dev
apt-get install zlib1g-dev
apt-get install openssl libssl-dev
./configure --prefix=/usr/local/nginx
make
make install
cd ..
./sbin/nginx
```
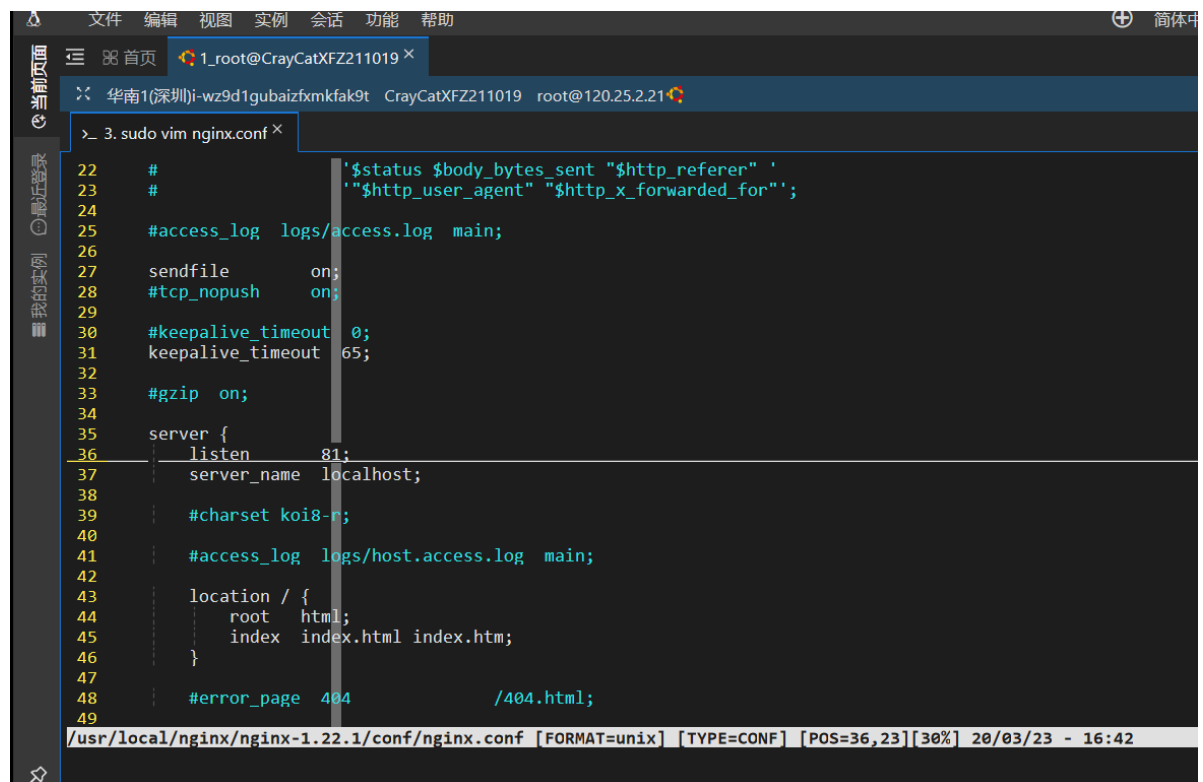
Nginx可能会和apache抢80端口

修改默认端口：

https://cloud.tencent.com/developer/article/2134157

https://blog.csdn.net/m0_67401545/article/details/126077253

记得把几个config的都改了：

```
[~] cd /usr/local/nginx/
[nginx] ls
client_body_temp   fastcgi_temp   logs          nginx-1.22.1.tar.gz   sbin          uwsgi_temp
conf               html           nginx-1.22.1  proxy_temp            scgi_temp
[nginx]
[nginx] cd conf/
[conf] ls
fastcgi.conf           fastcgi_params.default   mime.types         nginx.conf.default   uwsgi_params
fastcgi.conf.default   koi-utf                  mime.types.default scgi_params          uwsgi_params.default
fastcgi_params         koi-win                  nginx.conf         scgi_params.default  win-utf
[conf] sudo vim nginx.conf
[conf] sudo vim nginx.conf.default
[conf] cd ../
[nginx] sudo ./sbin/nginx
[nginx]
```

设置安全组、防火墙开放端口：

```
comp3070@CrayCat211019:/usr/local/nginx/nginx-1.22.1/conf » sudo ufw enable
Command may disrupt existing ssh connections. Proceed with operation (y|n)? y
Firewall is active and enabled on system startup
comp3070@CrayCat211019:/usr/local/nginx/nginx-1.22.1/conf » sudo ufw status
Status: active
comp3070@CrayCat211019:/usr/local/nginx/nginx-1.22.1/conf » sudo ufw allow 81
Rule added
Rule added (v6)
comp3070@CrayCat211019:/usr/local/nginx/nginx-1.22.1/conf » sudo ufw allow 81
Skipping adding existing rule
Skipping adding existing rule (v6)
comp3070@CrayCat211019:/usr/local/nginx/nginx-1.22.1/conf »
```

记得开放22端口！不然会导致workbench连接失败。

React环境

Nodejs

快速有效的方案

```
sudo apt install nodejs
```

and

```
sudo apt install npm
```

但是这样版本过低需要调整版本

```
sudo npm cache clean -f
sudo npm install -g n
sudo n stable
```

```
[~/web/webpage/software]$ sudo n stable
  installing : node-v18.14.2
       mkdir : /usr/local/n/versions/node/18.14.2
       fetch : https://nodejs.org/dist/v18.14.2/node-v18.14.2-linux-x64
     copying : node/18.14.2
node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.28' not found (
de)
/usr/local/bin/node: /lib/x86_64-linux-gnu/libc.so.6: version `GLIBC_2.
(required by /usr/local/bin/node)
   installed :  (with npm )

Note: the node command changed location and the old location may be rem
r current shell.
         old : /usr/bin/node
         new : /usr/local/bin/node
If "node --version" shows the old version then start a new shell, or re
on hash with:
hash -r  (for bash, zsh, ash, dash, and ksh)
rehash   (for csh and tcsh)
[~/web/webpage/software]$ node - v
>
(To exit, press ^C again or type .exit)
>
[~/web/webpage/software]$ node -v
v8.10.0
[~/web/webpage/software]$ hash -r
-L  -- list in the form of calls to hash
-d  -- use named directory hash table
```

```
 >_ 4. comp3070@CrayCat211019:~/COMP3070 ×

comp3070@CrayCat211019: ~/COMP3070
$ sudo n stable
   installing : node-v18.15.0
        mkdir : /usr/local/n/versions/node/18.15.0
        fetch : https://nodejs.org/dist/v18.15.0/node-v18
      copying : node/18.15.0
    installed : v18.15.0 (with npm 9.5.0)

Note: the node command changed location and the old loca
shell.
         old : /usr/bin/node
         new : /usr/local/bin/node
If "node --version" shows the old version then start a r
th:
hash -r  (for bash, zsh, ash, dash, and ksh)
rehash   (for csh and tcsh)

comp3070@CrayCat211019: ~/COMP3070
$ node --version
v10.19.0

comp3070@CrayCat211019: ~/COMP3070
$ hash -r

comp3070@CrayCat211019: ~/COMP3070
$ node --version
v18.15.0


comp3070@CrayCat211019: ~/COMP3070
$ ▯

default                    +

 & >_ 命令终端  已连接  华南1(深圳)  i-wz9d1gubaizfxmkfak9t  120.25.2.21:22
```

乱七八糟终归是在gcc阶段把服务器搞崩了的办法：

glibc 版本问题：

https://blog.csdn.net/m0_37201243/article/details/123641552?utm_medium=distribute.pc_relevant.non
e-task-blog-2~default~baidujs_baidulandingword~default-5-123641552-blog-123414527.pc_relevant_r
ecovery_v2&spm=1001.2101.3001.4242.4&utm_relevant_index=8

https://blog.csdn.net/qq_50247813/article/details/128870673


```
.../configure: line 1889: config.log: Permission denied
comp3070@CrayCat211019 /o/g/build> sudo !!                                     1
comp3070@CrayCat211019 /o/g/build> sudo ../configure  --prefix=/usr --disable-profile --e
nable-add-ons --with-headers=/usr/include --with-binutils=/usr/bin
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking for gcc... gcc
```

```
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
comp3070@CrayCat211019 /o/g/build> sudo apt install bison
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

```
make: *** No rule to make target 'install'.  Stop.
comp3070@CrayCat211019 /o/g/build> sudo apt install gawk
Reading package lists... Done
Building dependency tree
Reading state information... Done
Suggested packages:
  gawk-doc
The following NEW packages will be installed:
  gawk
0 upgraded, 1 newly installed, 0 to remove and 205 not upgraded.
Need to get 401 kB of archives.
After this operation, 1,552 kB of additional disk space will be used.
Get:1 http://mirrors.cloud.aliyuncs.com/ubuntu bionic/main amd64 gawk amd64 1:
1build1 [401 kB]
Fetched 401 kB in 0s (2,849 kB/s)
Selecting previously unselected package gawk.
(Reading database ... 141888 files and directories currently installed.)
Preparing to unpack .../gawk_1%3a4.1.4+dfsg-1build1_amd64.deb ...
Unpacking gawk (1:4.1.4+dfsg-1build1) ...
Setting up gawk (1:4.1.4+dfsg-1build1) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1)
```

make 版本问题

make版本升级：

编译glibc(gcc)以及过程中遇到的一些错误_glibc gcc_John4July的博客-CSDN博客

https://blog.csdn.net/m0_46900715/article/details/126247652

重新下载glibc2.29

configure prefix不能放在一个文件夹下否则make将卡在mv make 一直卡在pthread-pi-defines

终于到此安装成功

调整软连接：【请谨慎操作】Ubuntu18.04升级GLIBC_2.29，解决ImportError: /lib/x86_64-linux-gnu/libm.so.6: version `GLIBC_2.29'_ubuntu yum升级glibc_YirongChen的博客-CSDN博客

glibc2.28问题综述，真正问题是正则表达式只支持了3.x：

https://blog.csdn.net/weixin_42638388/article/details/121678015

https://blog.csdn.net/xueyumicheng/article/details/127728414

```
4768 else
4769   # Found it, now check the version.
4770   { $as_echo "$as_me:${as_lineno-$LINENO}: checking version o
4771 $as_echo_n "checking version of $MAKE... " >&6; }
4772   ac_prog_version=`$MAKE --version 2>&1 | sed -n 's/^.*GNU Ma
     \).*$/\1/p'`
4773   case $ac_prog_version in
4774     '') ac_prog_version="v. ?.??, bad"; ac_verc_fail=yes;;
4775     3.79* | 3.[89]* | 4.[0-9] )
4776         ac_prog_version="$ac_prog_version, ok"; ac_verc_fail=n
4777     *) ac_prog_version="$ac_prog_version, bad"; ac_verc_fail=
4778
4779   esac
4780   { $as_echo "$as_me:${as_lineno-$LINENO}: result: $ac_prog_v
4781 $as_echo "$ac_prog_version" >&6; }
4782 fi
4783 if test $ac_verc_fail = yes; then
4784   critic_missing="$critic_missing make"
4785 fi
4786
```

../configure权限：

```
comp3070@CrayCat211019:/opt/glibc-2.18/build $ ls -l ../configure
-rwxrwxr-x 1 root root 259735 Aug 11  2013 ../configure
comp3070@CrayCat211019:/opt/glibc-2.18/build $
```
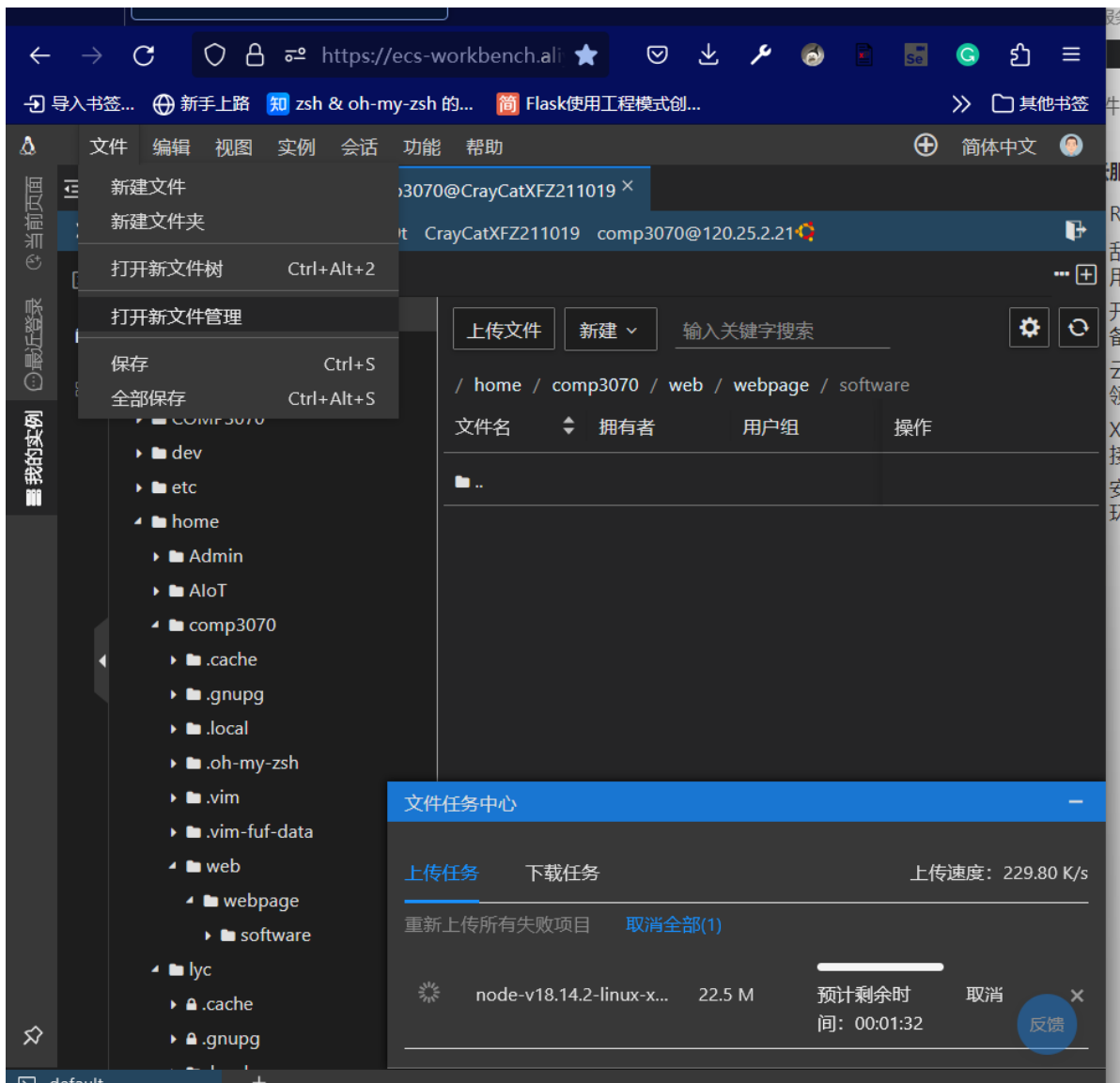
make install 报错解决：

https://blog.csdn.net/clirus/article/details/62425498

解压的glic-2.14.tar.gz源码和编译时定义的目录../configure --prefix=/home/software/glibc-2.14放到了一起。

不那么可靠的替代方案

nodejs.org

Linux 64-bit

上传（root）

配置环境变量

profile记得改权限chmod

```
28
29
30
31
32
33 export JAVA_HOME=/home/xfz/java/jdk-9.0.4
34 export CLASSPATH=.:${JAVA_HOME}/jre/lib/rt.jar:${JAVA_HOME}/lib/dt.jar:
   ${JAVA_HOME}/lib/tools.jar
35 export PATH=$PATH:${JAVA_HOME}/bin
36 export PATH=$PATH:/usr/local/bin
37
38
39
```

测试:

```
[~/web/webpage/software]$ node -v
v8.10.0
[~/web/webpage/software]$ npm - v

Usage: npm <command>

where <command> is one of:
    access, add-user, adduser, apihelp, author, bin, bugs, c,
    cache, completion, config, ddp, dedupe, deprecate, dist-ta
    dist-tags, docs, edit, explore, faq, find, find-dupes, get
    help, help-search, home, i, info, init, install,
    install-test, issues, it, la, link, list, ll, ln, login,
    logout, ls, outdated, owner, pack, ping, prefix, prune,
    publish, r, rb, rebuild, remove, repo, restart, rm, root,
    run-script, s, se, search, set, show, shrinkwrap, star,
    stars, start, stop, t, tag, team, test, tst, un, uninstall
    unlink, unpublish, unstar, up, update, upgrade, v, verisor
    version, view, whoami

npm <cmd> -h      quick help on <cmd>
npm -l            display full usage info
npm faq           commonly asked questions
npm help <term>   search for help on <term>
npm help npm      involved overview

Specify configs in the ini-formatted file:
    /home/comp3070/.npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

npm@3.5.2 /usr/share/npm
```

Create React App

需要在你的机器上安装 Node >= 14.0.0 和 npm >= 5.6。要创建项目，请执行：

```
npx create-react-app my-app
cd my-app
npm start
```

[服务器配置](#)

乱七八糟的备用笔记

su - comp3070

vim ~/.zshrc, 或者登录后直接vim .zshrc更好

在Linux中用vim编辑文本时，有时候会遇到按Esc键无法进入命令模式的情况。

最开始，只能强制结束远程ssh连接，重新连接后再编辑。

后来发现可以使用【ctrl】+【[】（左中括号）的快捷指令直接进入命令模式。

还有一种方式就是按住【ctrl】+【c】，也能强制进入命令模式

至于为什么按Esc键无法进入命令模式的原因未知，如果有知道的大佬麻烦科普下。

1. 强行wq的思路

按F1进入帮助，按:q退出，就可以:wq了

2. 一劳永逸的解决问题

在~/.vimrc下添加映射（可以自己改）

```
inoremap jj <Esc>
```

现在就可以用jj来代替esc了

编辑于 2022-08-07 20:09

注意：本文档没有任何问题，请仔细阅读文档，按步骤完成，不要轻易质疑文档的正确性；遇到问题，请大家先仔细阅读屏幕输出，不要直接提问

环境准备

推荐环境：

> 1. Mac + 阿里云主机
> 2. Linux + 阿里云主机
> 3. Windows + Xshell +阿里云主机
> 4. Windows + 虚拟机 + 阿里云主机

云服务器免费领取&配置

云主机首选阿里云高校计划，如高校计划无法领取免费的云主机，请选择阿里云云翼计划，购买一个月的学生机。

如果你已经不是学生了，请综合考虑各大云平台对新用户是否有优惠政策（阿里云，腾讯云，华为云，金山云，京东云等等），这里推荐腾讯云。

腾讯云

开发者专属扶持活动 （1核4G 2M带宽云服务器，3年仅需376元，购买时长依据自身学习时长确定，最少不低于半年哦）
 详情见：https://cloud.tencent.com/act/developer?from=12642#task_learn
 镜像选择：ubuntu18.04

华为云

华为云云创校园，学生优惠套餐

通用计算增强型云服务器，搭载自研华为鲲鹏920处理器及25GE智能高速网卡，提供强劲鲲鹏算力和高性能网络，购买指定配置服务可享受9元/月优惠，并赠送相同时长主机安全。

阿里云高校计划

本课程的项目阶段，会用到网络编程，为了方便评测及多机互联，需要一个具有公网IP地址的Linux主机，推荐大家领取阿里云的学生云主机。

请大家优先领取免费的云主机，**性能更高**，**时间更长**，**免费、免费、免费**

↓ ↓ ↓ ↓

领取链接：阿里云高校计划

完成上方↑流程，即可领取~今日数量有限，如未领到，次日8点可再领。

领取ECS学习资源 / 观看学习课程 / 免费续费，到期前30天内阶段考试通过可享

2核 cpu	4G 内存
Intel Xeon E5-2682 v4	最新一代DDR4内存
1M 带宽	40G 系统盘
VPC专有网络，I/O 优化	高效云盘

规格　　　2核4G
地域　　　地域、可用区可选
购买时长　6个月
价格　　　¥0

立即领取　您当前已经保有ECS实例，不能享受0元优惠

备份方案：阿里云云翼计划

购买链接：阿里云学生机 - 云翼计划

> 完成实名认证，且在24周岁下的用户均可购买阿里云学生机。购买时长为一个月即可，已经购买过的同学无需再次购买。

云服务器ECS 学生专享　HOT

CPU性能不限，高性价比，适用于个人开发者，网站建设、代码测试等

1核2G 1个月　　点击选择时长 ⌄

1M	40G	100%
带宽	高效云盘	CPU性能

¥9.50 /月起
省 ¥111.50/月

立即购买

云主机配置

选择配置的过程中，请注意选择操作系统版本为**Ubuntu 16.04 64位** 或 **Ubuntu 18.04 64位**。服务器地域及其他项不做限制，按默认即可。付款成功后，在进一步按提示配置服务器时，请牢记你所设置的root用户密码。

地域:	华东 1（杭州）⌄	华东 1 可用区 B ⌄

实例:	1 CPU 2G 内存
	ecs.n4.small

已选实例: ecs.n4.small1核 2GB, 共享计算型 n4 I/O 优化实例

操作系统:	Ubuntu ⌄	16.04 64位 ⌄

系统盘: 高效云盘 40 GB

网络类型: 专有网络

默认专有网络 ⌄	默认交换机 ⌄

如需使用其他专有网络，请选择已有专有网络，也可以自行到控制台创建>

带宽: 按固定带宽 1 Mbps

购买时长: 1 个月 ⌄

当配置完成后，你可以在导航栏中 控制台 > 云服务器ECS 的 概览下，看到你所购买的云主机及其IP地址：

华北2（北京）   🛒

云服务器	即将过期❓	已过期	运行中	已停止	近期创建❓
1	0 续费	0 续费	1	0	0

磁盘 1	镜像 0	快照 0 B/0

实例ID	IP地址	付费方式	实例状态
i-2z▇▇▇2wbmtwun ▇▇▇	4▇▇▇0(公网) 1▇▇ ▇10(私网)	包年包月 2021年2月21日 00:00:00到期	运行中

点击蓝色实例ID可以查看更详细的信息及相关设置，在这里你可以记录下你的公网IP，以便后续远程连接云主机时使用。

> 如果你到这一步，并没有设置root用户的密码，那么请你百度如下字段：阿里云服务器如何重置root密码。
>
> 🖼image-20230319154241089
>
> 选择重置实例密码，默认自动生成的密码保存在浏览器密码箱中。

请注意看这里

请注意看这里

如果你用的是腾讯云等其他云平台，可能系统安装后的某人用户不是 `root` ,那么，请你在下面的 `Xshell安装及连接云服务器` 的章节中，把用户名 `root` 更改为你的服务商设置的用户名。

如：腾讯云为 `ubuntu`

然后，连接到你的云主机之后，请使用 `sudo passwd root` 命令给 `root` 用户一个密码，之后重新做下面的 `使用Xshell连接云服务器` 的操作，使用用户名root和刚才你设置的密码。 （如果你会Linux，那么可以直接在后面的操作中，使用sudo来获取管理员权限）

之后，你就可以完全按照该文档操作了。

Xshell安装及连接云服务器

注意：如果你的电脑是Linux或Mac系统，则无需下载安装Xshell，及Xftp；

使用Linux或者Mac连接阿里云主机的方式为：ssh username@your_ip

1. 访问<u>XShell个人免费版下载页面</u>，按提示填写姓名以及邮件地址，勾选"两者"。

XSHELL 6　　**XFTP 6**

免费只供非商业用途。

姓名（必填）
your_name

邮件（必填）
your@email.com

☑ 两者　☐ 只需Xshell　☐ 只需Xftp

下载

注意：需要一个有效的电子邮件地址。下载链接将发送到您的邮箱。

免费使用条款

NetSarang Computer,Inc.以过去10年免费提供强大的SSH和SFTP/FTP客户端而自豪。我们的免费许可证不仅仅是免费的价格，而且没有广告或其他剥削用户的方式。我们认为，来自各种背景和环境的用户都应该能够访问功能强大、功能丰富的SSH和SFTP/FTP客户机。无论是学习、教学，还是仅仅是作为一种爱好的补充。

NetSarang Computer,Inc免费许可Xshell和Xftp仅用于非商业用途。任何将我们免费的许可用于商业目的的行为都是严格违反我们的免费最终用户许可协议中规定的条款的。如果您希望将Xshell或Xftp用于商业用途，我们鼓励您购买许可证，并帮助我们进一步开发我们的软件。

通过下载我们的免费授权软件，表示您同意接受与偶尔的促销折扣或特殊活动相关的电子邮件，以及偶尔的补丁说明和通知。您可以通过点击任何邮件底部的"退订"按钮，随时退订这些邮件。我们不会向第三方透露您的任何信息。

2. 登录你所填写的邮箱，你将会收到一封带有下载地址的邮件，点击即可下载Xshell和xftp。

Xshell 6 download instruction ☆

发件人： **NetSarang, Inc.** <no-reply@netsarang.com>
时　间： 2020年7月1日（星期三）下午10：49
收件人： 坂田银串 <ginakira@foxmail.com>

纯文本 | ⊡ ▤ ⊡ 🖶 ⩔

Dear user,

Thank you for your interest in Xshell 6. If you did not request a download link for Xshell 6, please contact our support team at support@netsarang.com to have your email address removed from any future emails related to Xshell 6.

Please click the link below to start downloading your software:

https://www.netsarang.com/zh/downloading/?token=WnF6dGRlNVRHbkYwc3lXa1ZaNG5iUUBFZG9LdmFzeGxVeUI2eElFOXljGRn

This link will expire on July 31, 2020

如果你并未找到这封邮件，可以检查一下邮箱的垃圾箱，或重新填写上述网址的表单。

如果下载速度较为缓慢，可以尝试科学上网进行下载。

同样有条件的同学也可以使用网盘下载，链接:https://pan.baidu.com/s/1FcY0r3t-EcwJNIa4ca5oxQ 提取码:dmza （永久有效，但不能保证一定可以）

3. 安装过程不涉及特殊设置，一路下一步直到安装完成即可（或可按个人需求更改安装路径）。

4. 安装完成后，打开xshell，点击左上角的新建会话图标，选择SSH协议，将阿里云控制台中你服务器的**公网IP**填入主机一栏。（名称一栏为方便标示不同的主机用，可根据需求自行填写）
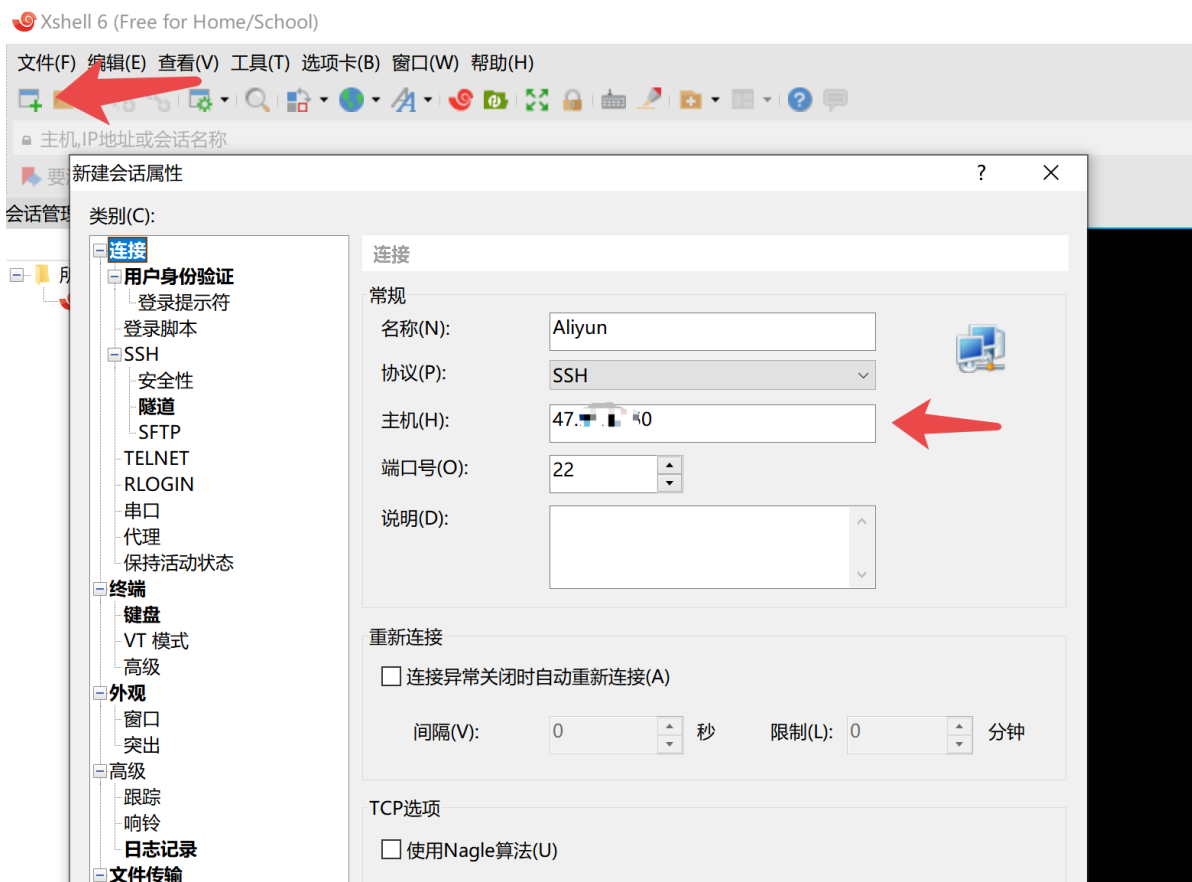
5. 点击左侧用户身份验证一项，填入服务器的用户名与密码（阿里云默认用户名为root），然后点击下方的连接按钮。
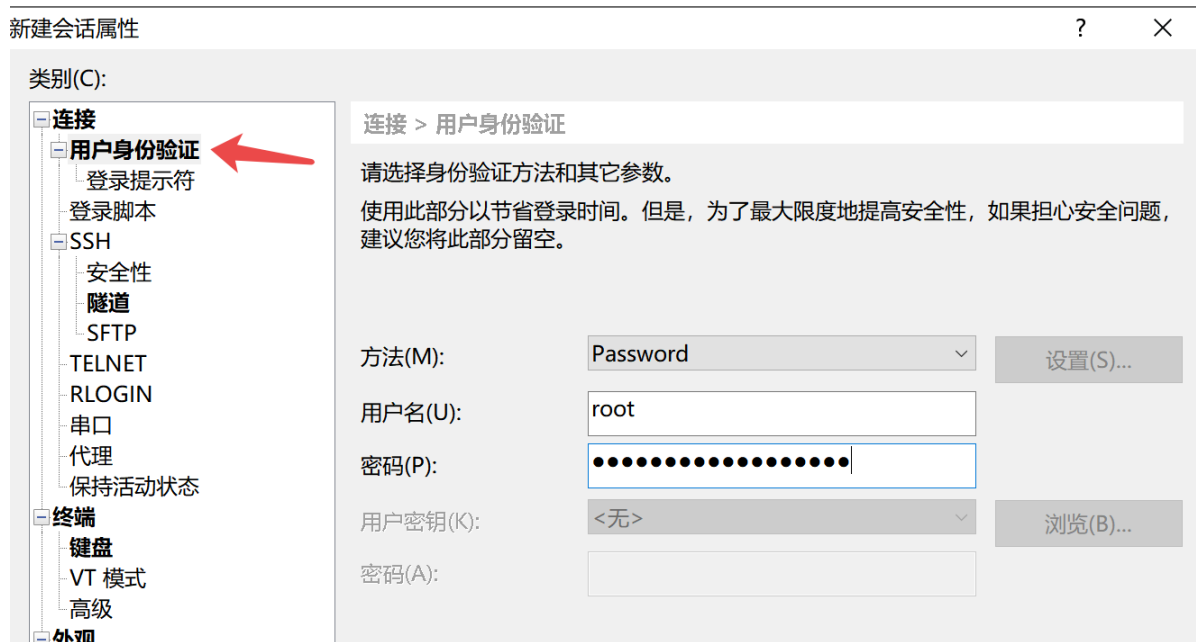
6. 首次连接会弹出SSH安全警告，选择接受并保存即可。

SSH安全警告

未知主机密钥

■.■ ■.■ (端口: 22)的主机密钥未在本地主机密钥数据库中注册。可以保存主机密钥以验证此主机以供将来连接。

主机密钥指纹：

MD5:4b:ba:b7:66:83:d6:eb:ce:f9:aa:d2:ae:92:7b:f8:06

SHA256:7...

```
+---[RSA 2048]----+      +---[RSA 2048]----+
|                 |      |.                |
|                 |      | +               |
|                 |      | *               |
|                 |      |+E+..       . .  |
|        S        |      |o+==.     S o .  |
|    E   o .      |      | oo+. . .*.o     |
|   +  . ■        |      |  . oo  ^+R.     |
| + + ■.  .       |      |  .   ■ ■        |
| .*.  ■:■ .      |      |        +BB=+oo  |
+------[MD5]------+      +----[SHA256]-----+
```

接受此主机密钥吗？

一次性接受(O)   接受并保存(S)   取消

7. 当看到Welcome字样的提示信息时，代表已经成功连接了云服务器。如果没有看到欢迎信息而是看到了拒绝连接等的提示信息，请检查你的用户名或密码是否输入正确。

xftp无需配置，可用来在你的本地主机和远程主机之间传输文件；

感兴趣的同学也可以了解两个Linux命令：rz、sz。

如果你不愿意用xshell，在win10下下载一个git bash，然后使用ssh命令连接云主机也可以

安装系统后的环境准备

添加普通用户（如已添加普通用户，跳转到下一步配置sshd）

> 在Linux使用过程中，应尽量避免使用root用户直接使用系统，请使用下面的步骤创建一个新用户

1. 添加新用户

```
adduser new_user #根据自己的真实需求修改new_user
#这里是创建一个新的用户，用户名不要用new_user
```

2. 将新用户添加到 `sudo` 组中

```
usermod -G sudo new_user
```

3. 使用 `su` 命令切换到新用户

```
su - new_user
```

> 添加新用户之后，请在 `xshell` 中重新添加一个新用户的连接，以后直接使用新用户登录系统

`↓↓↓`

==请注意，从这里开始，所有的操作都是用普通用户做的==

`↑↑↑`

配置sshd

> Vim的简单使用，请百度

1. 使用命令 `sudo vim /etc/ssh/sshd_config` 打开sshd的配置文件，找到 `ClientAliveInterval` 和 `ClientAliveCountMax` 并将其修改为（如果没有直接添加即可）：

2. 重启sshd服务

```
sudo service sshd restart
```

> 如果上述命令报错，大致内容为sshd这个服务不存在的话，就执行 `sudo service ssh restart`

以下所有配置，都是为了优化终端，提升使用效率，如果你已经能独立对bash，zsh，vim等进行配置优化，可自行选择方案，无需完全按照这个方案

GitHub访问优化

1. 请进入这个网址：[点击这里](#)

2. 找到图中的 `IP1`

Hostname Summary

Domain	fastly.net
IP Address	199.232.69.194　IP1
Web Server Location	🇺🇸 United States

Updated: Mon, 24 Aug 2020 14:31 GMT | Reviewed: Tue, 25 Aug, 2020 07:31 GMT

3. 在上面的网站上搜索 `github.com`，找到 `IP2`

IPAddress.com　What Is My IP:　113.4.139.66
The Best IP Address Tools

github.com

GitHub.com

Domain Summary

Global Traffic Rank	89　▼17
Estimated Visitors	3.2 Million / Day
Estimated Page Impressions	20 Million / Day
Domain Creation Date	October 9, 2007
Domain Age	12 years, 10 months and 16 days (4,704 days)
IP Address	140.82.112.4　IP2
Web Server Location	🇺🇸 United States

Updated: Tue, 25 Aug 2020 08:38 GMT

4. 使用命令 `sudo vim /etc/hosts` 打开hosts文件，并在最后加入以下信息

```
199.232.69.194 github.global.ssl.fastly.net
140.82.112.4 github.com
```

5. 保存并退出

配置Vim（使用新添加的用户操作）

> 在后续的学习过程中，会使用 `vim` 写程序

[Vim配置推荐 - ma6174](#)（不用打开这个官方网站）

配置出现问题：

🖼️image-20230320101335124

后续手动补充安装

1. 更新apt源信息

```
sudo apt update
```

2. 配置vim，执行下面命令配置安装vim

```
wget 47.93.11.51:88/install_vim.sh
bash install_vim.sh
```

> vim的配置因为需要安装较多插件，所以需要等较多时间，大家耐心等待

zsh的安装及配置

1. 安装zsh

```
sudo apt install zsh
```

2. 修改默认shell为zsh

```
chsh -s /bin/zsh
```

3. 安装oh-my-zsh

```
sh -c "$(wget https://raw.github.com/robbyrussell/oh-my-
zsh/master/tools/install.sh -O -)"
##如果不成功，请执行下面两条命令，成功了就不需要做下面两条
wget 47.93.11.51:88/install_zsh.sh
bash install_zsh.sh
```

4. 安装zsh-syntax-highlighting

```
git clone https://github.com/zsh-users/zsh-syntax-highlighting.git
${ZSH_CUSTOM:-~/.oh-my-zsh/custom}/plugins/zsh-syntax-highlighting
```

5. 使用命令 `vim .zshrc` 打开.zshrc文件，找到 `plugins=()` 这一行，将zsh-syntax-highlighting添加进去

   shift+delete

```
plugins=(git zsh-syntax-highlighting)
```

6. 安装其他插件

```
##命令自动补全插件
mkdir ~/.oh-my-zsh/plugins/incr
wget http://mimosa-pudica.net/src/incr-0.2.zsh -O ~/.oh-my-
zsh/plugins/incr/incr.plugin.zsh
##命令自动推荐，根据历史记录
git clone https://github.com/zsh-users/zsh-autosuggestions ${ZSH_CUSTOM:-~/.oh-
my-zsh/custom}/plugins/zsh-autosuggestions
##目录自动跳转插件
sudo apt install autojump
```

7. 使用命令 `vim .zshrc`，打开后在最后插入以下内容：

```
#设置终端颜色，提示符，及上一条指令返回码提示
autoload -U colors && colors
PROMPT="%{$fg[red]%}%n%{$reset_color%}@%{$fg[blue]%}%m %{$fg[yellow]%}%1~ %
{$reset_color%}%# "
RPROMPT="[%{$fg[yellow]%}%?%{$reset_color%}]"
# Useful support for interacting with Terminal.app or other terminal programs
[ -r "/etc/zshrc_$TERM_PROGRAM" ] && . "/etc/zshrc_$TERM_PROGRAM"
source ~/.oh-my-zsh/custom/plugins/zsh-autosuggestions/zsh-
autosuggestions.plugin.zsh
source /usr/share/autojump/autojump.sh
source ~/.oh-my-zsh/plugins/incr/incr*.zsh
```

> 注意，复制后可能会因为Vim的配置导致以上内容被注释，也就是在前面加上了 `#`,如果有的话，删掉就行。

ctags安装与配置

1. 使用以下命令安装**ctags**

```
sudo apt install ctags
```

2. 执行以下命令

```
ctags -I __THROW -I __attribute_pure__ -I __nonnull -I __attribute__ --file-
scope=yes --langmap=c:+.h --languages=c,c++ --links=yes --c-kinds=+p --c++-
kinds=+p --fields=+iaS --extra=+q  -f ~/.vim/systags /usr/include/*
/usr/include/x86_64-linux-gnu/sys/* /usr/include/x86_64-linux-gnu/bits/*
/usr/include/arpa/*
```

3. 使用命令 `vim .vimrc` 编辑.vimrc，在最后添加以下内容

```
set tags+=~/.vim/systags
```

安装glibc-doc

1. 使用以下命令安装

```
sudo apt install glibc-doc
```

vim .zshrc

```
cat: vim: No such file or directory

# If you come from bash you might have to change your $PATH.

# export PATH=$HOME/bin:/usr/local/bin:$PATH


# Path to your oh-my-zsh installation.
```

```
export ZSH=$HOME/.oh-my-zsh

# Set name of the theme to load --- if set to "random", it will

# load a random theme each time oh-my-zsh is loaded, in which case,

# to know which specific one was loaded, run: echo $RANDOM_THEME

# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes

#ZSH_THEME="zhann.zsh-theme"

ZSH_THEME="random"

# Set list of themes to pick from when loading at random

# Setting this variable when ZSH_THEME=random will cause zsh to load

# a theme from this variable instead of looking in $ZSH/themes/

# If set to an empty array, this variable will have no effect.

# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.

# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.

# Case-sensitive completion must be off. _ and - will be interchangeable.

# HYPHEN_INSENSITIVE="true"

# Uncomment the following line to disable bi-weekly auto-update checks.

# DISABLE_AUTO_UPDATE="true"

# Uncomment the following line to automatically update without prompting.

# DISABLE_UPDATE_PROMPT="true"

# Uncomment the following line to change how often to auto-update (in days).

# export UPDATE_ZSH_DAYS=13
```

```
# Uncomment the following line if pasting URLs and other text is messed up.

# DISABLE_MAGIC_FUNCTIONS="true"



# Uncomment the following line to disable colors in ls.

# DISABLE_LS_COLORS="true"



# Uncomment the following line to disable auto-setting terminal title.

# DISABLE_AUTO_TITLE="true"



# Uncomment the following line to enable command auto-correction.

# ENABLE_CORRECTION="true"



# Uncomment the following line to display red dots whilst waiting for
completion.

# Caution: this setting can cause issues with multiline prompts (zsh 5.7.1 and
newer seem to work)

# See https://github.com/ohmyzsh/ohmyzsh/issues/5765

# COMPLETION_WAITING_DOTS="true"



# Uncomment the following line if you want to disable marking untracked files

# under VCS as dirty. This makes repository status check for large repositories

# much, much faster.

# DISABLE_UNTRACKED_FILES_DIRTY="true"



# Uncomment the following line if you want to change the command execution time

# stamp shown in the history command output.

# You can set one of the optional three formats:

# "mm/dd/yyyy"|"dd.mm.yyyy"|"yyyy-mm-dd"

# or set a custom format using the strftime function format specifications,
```

```zsh
# see 'man strftime' for details.

# HIST_STAMPS="mm/dd/yyyy"


# Would you like to use another custom folder than $ZSH/custom?

# ZSH_CUSTOM=/path/to/new-custom-folder


# Which plugins would you like to load?

# Standard plugins can be found in $ZSH/plugins/

# Custom plugins may be added to $ZSH_CUSTOM/plugins/

# Example format: plugins=(rails git textmate ruby lighthouse)

# Add wisely, as too many plugins slow down shell startup.

plugins=(git

zsh-syntax-highlighting

autojump

incr

colored-man-pages

emoji

urltools

extract

rand-quote

zsh_reload

hitokoto

command-not-found

oneko

    )


setopt no_nomatch

source $ZSH/oh-my-zsh.sh
```

```
# User configuration


# export MANPATH="/usr/local/man:$MANPATH"


# You may need to manually set your language environment

# export LANG=en_US.UTF-8


# Preferred editor for local and remote sessions

# if [[ -n $SSH_CONNECTION ]]; then

#   export EDITOR='vim'

# else

#   export EDITOR='mvim'

# fi


# Compilation flags

# export ARCHFLAGS="-arch x86_64"


# Set personal aliases, overriding those provided by oh-my-zsh libs,

# plugins, and themes. Aliases can be placed here, though oh-my-zsh

# users are encouraged to define aliases within the ZSH_CUSTOM folder.

# For a full list of active aliases, run `alias`.

#

# Example aliases

# alias zshconfig="mate ~/.zshrc"

# alias ohmyzsh="mate ~/.oh-my-zsh"


echo "爆裂吧现实 粉碎吧精神 放逐这个世界 ~/.zshrc"
```

```
autoload -U colors && colors

#PROMPT="%{$fg[red]%}%n%{$reset_color%}@%{$fg[blue]%}%m %{$fg[yellow]%}%1~ %
{$reset_color%}%# "

#RPROMPT="[%{$fg[yellow]%}%?%{$reset_color%}]"

# Useful support for interacting with Terminal.app or other terminal programs

[ -r "/etc/zshrc_$TERM_PROGRAM" ] && . "/etc/zshrc_$TERM_PROGRAM"

source ~/.oh-my-zsh/custom/plugins/zsh-autosuggestions/zsh-
autosuggestions.plugin.zsh

source /usr/share/autojump/autojump.sh

source ~/.oh-my-zsh/plugins/incr/incr*.zsh




alias test='ssh xfz'
```

登录Token：0x21p0mb3m

地区ID：cn-shenzhen

区域：华南1(深圳)

实例 ID：i-wz9d1gubaizfxmkfak9t

主机地址：120.25.2.21

端口号：22

实例名称：CrayCatXFZ211019

公网IP：120.25.2.21

私网IP：172.25.36.169

OS用户名：root

CPU：2

内存：2 G

系统名称：Ubuntu 18.04 64位

系统发行版本名称：Ubuntu

```
source .zshrc
```

zlogin