

# 环境准备 - 1

安装系统

# 环境准备 - 2

## Centos

### 1. 更新yum源

```
1 [haizei@bogon ~]$ su - root
2 密码:
3 [root@bogon ~]# cd /etc/yum.repos.d/
4 [root@bogon yum.repos.d]# mkdir backup
5 [root@bogon yum.repos.d]# ls
6 backup                CentOS-Base.repo        CentOS-CR.repo
  CentOS-Extras.repo    CentOS-Media.repo        CentOS-Sources.repo
7 CentOS-AppStream.repo CentOS-centosplus.repo  CentOS-Debuginfo.repo
  CentOS-fasttrack.repo CentOS-PowerTools.repo  CentOS-Vault.repo
8 [root@bogon yum.repos.d]# mv *.repo backup/
9 [root@bogon yum.repos.d]# ls
10 backup
```

以上操作备份了CentOS自带的源列表，接下来，我们使用阿里云的源作为CentOS的源。

注意，根据自己的CentOS版本不同，以下操作有所不同

### centos 8

```
1 #CentOS 8
2 [root@bogon yum.repos.d]# cat /etc/redhat-release
3 CentOS Linux release 8.0.1905 (Core)
4 [root@bogon yum.repos.d]# wget -O /etc/yum.repos.d/CentOS-Base.repo
  https://raw.githubusercontent.com/hackyoMa/docker-centos/8/CentOS-
  Base.repo
5 --2019-11-27 00:38:14--
  https://raw.githubusercontent.com/hackyoMa/docker-centos/8/CentOS-
  Base.repo
6 正在解析主机 raw.githubusercontent.com (raw.githubusercontent.com)...
  151.101.228.133, ::
7 正在连接 raw.githubusercontent.com
  (raw.githubusercontent.com)|151.101.228.133|:443... 已连接。
8 已发出 HTTP 请求，正在等待回应... 200 OK
9 长度: 2100 (2.1K) [text/plain]
```

```

10 正在保存至: "/etc/yum.repos.d/CentOS-Base.repo"
11
12 /etc/yum.repos.d/CentOS-Base.repo    100%
   [=====>]
   2.05K  --.-KB/s  用时 0s
13
14 2019-11-27 00:38:14 (24.1 MB/s) - 已保存 "/etc/yum.repos.d/CentOS-
   Base.repo" [2100/2100])
15
16 [root@bogon yum.repos.d]# ls
17 backup  CentOS-Base.repo

```

## centos 7

```

1  #CentOS 7
2  #直接执行以下命令
3  wget -O /etc/yum.repos.d/CentOS-Base.repo
   http://mirrors.aliyun.com/repo/Centos-7.repo
4  #如果是7以下的版本，只需要将数字7改成相应的版本号就行

```

### 接下来的操作所有版本都做

```

1  [root@bogon yum.repos.d]# yum makecache
2  CentOS-8 - AppStream
                                     1.4 MB/s | 6.3 MB      00:04
3  CentOS-8 - Base
                                     1.4 MB/s | 7.9 MB      00:05
4  CentOS-8 - Extras
                                     913 B/s | 2.1 kB      00:02
5  元数据缓存已建立。
6  [root@bogon yum.repos.d]# yum install gcc gcc-c++ git
7  #注意屏幕输出，需要按【y】的时候输入以继续
8

```

### 确认环境，很重要

```

1
2  #如果是以下情况，请按指导继续往下，如果成功切换为root则跳过本步骤
3  [haizeibogon ~]$ sudo -i
4  [sudo] haizeibogon 的密码:
5  haizeibogon 不在 sudoers 文件中。此事将被报告。
6  [haizeibogon ~]$ su - root
7  密码:
8  [root@bogon ~]#
9  [root@bogon ~]# vim /etc/sudoers
10 #找到文件中的 root ALL=(ALL) ALL这一行，CentOS 8大概在第100行

```

```
11 #插入一行新的内容
12 haizei ALL=(ALL) ALL
13 #插入后保存并退出
14 [root@bogon ~]# logout
15 [haizei@bogon ~]$ sudo -i
16 [sudo] haizei 的密码:
17 [root@bogon ~]#
18 #如上能成功切换为root就成功了
```

## Ubuntu 及 Deepin

使用Deepin的同学请自行判断以下操作是否适合自己的系统，理论上Deepin和Ubuntu都是debian系统，有很多相似之处，但细节方面需要注意。

### 更新apt源

#### 1. 确定系统版本

对于Ubuntu，或者说所有的Linux操作系统，安装软件，更新源操作时需要注意自己的系统版本及类型

```
1 suyelu@HaiZei-Tech:~$ cat /etc/lsb-release
2 DISTRIB_ID=Ubuntu
3 DISTRIB_RELEASE=16.04
4 DISTRIB_CODENAME=xenial
5 DISTRIB_DESCRIPTION="Ubuntu 16.04.2 LTS"
```

通过以上命令，我们可以看出我所使用的是Ubuntu 16.04的系统，其代号是xenial 请你一定要确认自己的Ubuntu系统版本，系统版本不一致,贸然安装软件，只有重装一个结局。

#### 2. 更换源列表

```
1 suyelu@HaiZei-Tech:~$ sudo vim /etc/apt/sources.list
```

以上命令打开了使用Vim打开了一个新的文件，请你注意里面是否有xenial字段，这个对应着你的系统版本16.04，而如果里面有bionic字段，则说明是18.04

使用你已经掌握的vim命令，将文件清空，切换为插入模式

16.04插入以下内容

```
1 # deb cdrom:[Ubuntu 16.04 LTS _Xenial Xerus_ - Release amd64
  (20160420.1)]/ xenial main restricted
2 deb-src http://archive.ubuntu.com/ubuntu xenial main restricted
  #Added by software-properties
3 deb http://mirrors.aliyun.com/ubuntu/ xenial main restricted
```

```
4 deb-src http://mirrors.aliyun.com/ubuntu/ xenial main restricted
  multiverse universe #Added by software-properties
5 deb http://mirrors.aliyun.com/ubuntu/ xenial-updates main
  restricted
6 deb-src http://mirrors.aliyun.com/ubuntu/ xenial-updates main
  restricted multiverse universe #Added by software-properties
7 deb http://mirrors.aliyun.com/ubuntu/ xenial universe
8 deb http://mirrors.aliyun.com/ubuntu/ xenial-updates universe
9 deb http://mirrors.aliyun.com/ubuntu/ xenial multiverse
10 deb http://mirrors.aliyun.com/ubuntu/ xenial-updates multiverse
11 deb http://mirrors.aliyun.com/ubuntu/ xenial-backports main
  restricted universe multiverse
12 deb-src http://mirrors.aliyun.com/ubuntu/ xenial-backports main
  restricted universe multiverse #Added by software-properties
13 deb http://archive.canonical.com/ubuntu xenial partner
14 deb-src http://archive.canonical.com/ubuntu xenial partner
15 deb http://mirrors.aliyun.com/ubuntu/ xenial-security main
  restricted
16 deb-src http://mirrors.aliyun.com/ubuntu/ xenial-security main
  restricted multiverse universe #Added by software-properties
17 deb http://mirrors.aliyun.com/ubuntu/ xenial-security universe
18 deb http://mirrors.aliyun.com/ubuntu/ xenial-security multiverse
```

#### 18.04插入以下内容

```
1 deb http://mirrors.aliyun.com/ubuntu/ bionic main restricted
  universe multiverse
2 deb http://mirrors.aliyun.com/ubuntu/ bionic-security main
  restricted universe multiverse
3 deb http://mirrors.aliyun.com/ubuntu/ bionic-updates main
  restricted universe multiverse
4 deb http://mirrors.aliyun.com/ubuntu/ bionic-proposed main
  restricted universe multiverse
5 deb http://mirrors.aliyun.com/ubuntu/ bionic-backports main
  restricted universe multiverse
6 deb-src http://mirrors.aliyun.com/ubuntu/ bionic main restricted
  universe multiverse
7 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-security main
  restricted universe multiverse
8 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-updates main
  restricted universe multiverse
9 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-proposed main
  restricted universe multiverse
10 deb-src http://mirrors.aliyun.com/ubuntu/ bionic-backports main
  restricted universe multiverse
```

修改后，保存并退出

### 3. 更新软件信息

```
1 | suyelu@HaiZei-Tech:~$ sudo apt-get update
2 | suyelu@HaiZei-Tech:~$ sudo apt-get upgrade
```

## 安装openssh-server

```
1 | sudo apt-get install openssh-server
```

## 环境准备 - 3

```
1 | [haizei@bogon ~]$ cd
2 | [haizei@bogon ~]$ ls
3 | 公共 模板 视频 图片 文档 下载 音乐 桌面
4 | [haizei@bogon ~]$ mkdir bin
5 | [haizei@bogon ~]$ cd bin/
6 | [haizei@bogon bin]$ vim config
7 | #写入 name=自己的真实姓名 然后保存并退出, 注意等于号两边没有空格
8 | [haizei@bogon bin]$ vim check.c
9 | #根据校区笔筒写入下面不同的内容
```

### 哈尔滨校区

```
1 | #include <stdio.h>
2 | #include <sys/socket.h>
3 | #include <sys/types.h>
4 | #include <arpa/inet.h>
5 | #include <unistd.h>
6 | #include <stdlib.h>
7 | #include <string.h>
8 | #include <pwd.h>
9 | int socket_connect(int port, char *host) {
10 |     int sockfd;
11 |     struct sockaddr_in dest_addr;
12 |     if ((sockfd = socket(AF_INET, SOCK_STREAM, 0)) < 0) {
13 |         perror("socket() error");
14 |         exit(1);
15 |     }
16 |
17 |     memset(&dest_addr, 0, sizeof(dest_addr));
18 |     dest_addr.sin_family = AF_INET;
19 |     dest_addr.sin_port = htons(port);
20 |     dest_addr.sin_addr.s_addr = inet_addr(host);
21 |
22 |     if (connect(sockfd, (struct sockaddr *)&dest_addr, sizeof(dest_addr)) <
0) {
```

```

23     perror("connect() error");
24     exit(1);
25 }
26 return sockfd;
27
28 }
29
30 int main(int argc, char **argv) {
31     char realname[20] = {0};
32     int socket_fd;
33     struct passwd *pwd;
34     pwd = getpwuid(getuid());
35     char *ip_addr = "192.168.1.40";
36     int port = 20198;
37     char username[40] = {0};
38     strcpy(realname, argv[1]);
39     sprintf(username, "%s:%s", pwd->pw_name, realname);
40     socket_fd = socket_connect(port, ip_addr);
41     if (socket_fd < 0)
42     {
43         exit(1);
44     }
45     if (send(socket_fd, username, strlen(username), 0) > 0) {
46         printf("Check Success\n");
47     }
48     close(socket_fd);
49     return 0;
50 }

```

## 佳木斯校区

```

1  #将程序的第35行改为下面的语句即可
2  char *ip_addr = "192.168.2.40";
3

```

注意以下内容都得做

```

1  [haizeibogon bin]$ gcc -o check check.c
2  [haizeibogon bin]$ ls
3  check check.c config
4  [haizeibogon bin]$ vim check.sh
5  #写入下面的文件内容

```

```

1  #!/bin/bash
2  source ~/bin/config
3  cd

```

```

4 Pwd=`pwd`
5 cd - >/dev/null
6 ping -c 2 192.168.1.40 2>&1 >/dev/null
7
8 Type=0
9
10 if [[ $? -eq 0 ]];then
11     Type=1
12 else
13     ping -c 2 192.168.2.40 2>&1 >/dev/null
14     if [[ $? -eq 0 ]]; then
15         Type=2
16     fi
17 fi
18
19 if [[ $Type -ne 0 ]];then
20     $Pwd/bin/check $name 2>&1 > $Pwd/check.log
21     echo `date` >> $Pwd/check.log
22 fi

```

```

1 [haizei@bogon bin]$ chmod +x check.sh
2 [haizei@bogon bin]$ crontab -e
3 #注意上面的命令，Ubuntu第一次执行时，会让你选择默认编辑器，选一个带有vim的即可，然后再
  文件末尾加入以下内容后保存并退出
4 */1 * * * * ~/bin/check.sh
5 #使用以下命令验证
6 [haizei@bogon bin]$ crontab -l
7 */1 * * * * ~/bin/check.sh
8 [haizei@bogon bin]$
9 #一分钟后：
10 [haizei@bogon bin]$ cat ~/check.log
11 Check Success
12 2019年 11月 27日 星期三 02:39:02 EST
13 #以上则表示成功

```

## 环境准备 - 4

### 免密登录设置

需要完成以下几个免密登录：

1. 自己的电脑普通用户到树莓派的普通用户
2. 自己的电脑的root用户到树莓派的普通用户
3. 宿船长到你的电脑的root用户
4. 宿船长到你的电脑的普通用户
5. 宿船长到你的树莓派普通用户

```
1  ssh-rsa
   AAAAB3NzaC1yc2EAAAADAQABAAQgQC2CLtRyEfOyMT7dQCm5XWzMGRqJLkMLfqkTr1fuwdB+7d
   hA4rLk8aQuURhFRa/ag2eTiyTdiVIaKcCaZglGJuMwI4cNiT4xMCy7uWbTBT4otjjqAxE5/tTdE
   SIFkVs2DeTEi93uHHT7/MtswSQho+uyp5sM8CGWKccT+y5QC09NQ== suyelu@bonon
```

以上为宿船长的公钥。

免密操作请参照：<https://www.haizeix.com/course/359/task/19438/show>