MPI 编程实验环境配置及示例(2025)

实验概述与注意事项

1. 购买三台华为鲲鹏云服务器及必要的网络资源， 注意请参照本文档配置购买资源，不要购买价格较高的额外资源，如果在实验、作业期间，代金券快用完了，请及时联系老师或助教；
2. 三台机器均使用ssh连接，并进行免密配置，安装mpi及依赖包，参照后文环境配置即可；
3. 进行HelloWorld实验，验证环境安装正确，参照“HelloWorld实验示例”；
4. 完成所有实验、作业后，**必须停止/关闭所有华为云资源**，方法参照“华为云资源的关停”,否则将会继续扣费导致欠费！

环境配置流程

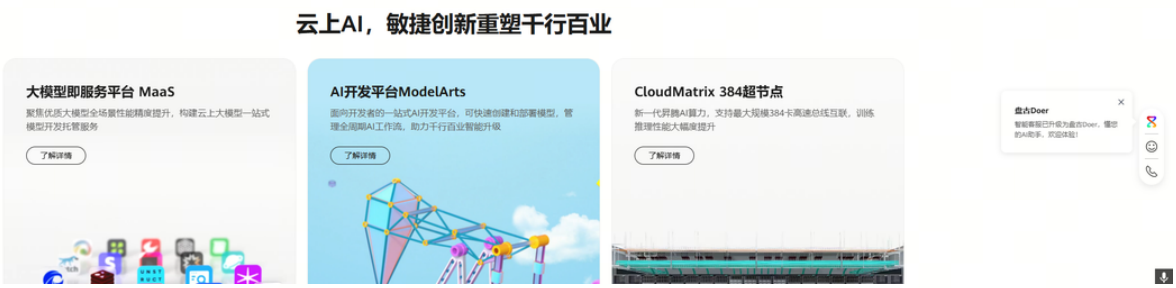
1. 购买华为鲲鹏云服务器

1.1 登录

浏览器访问https://www.huaweicloud.com进行登录操作。

1.2 检查代金券

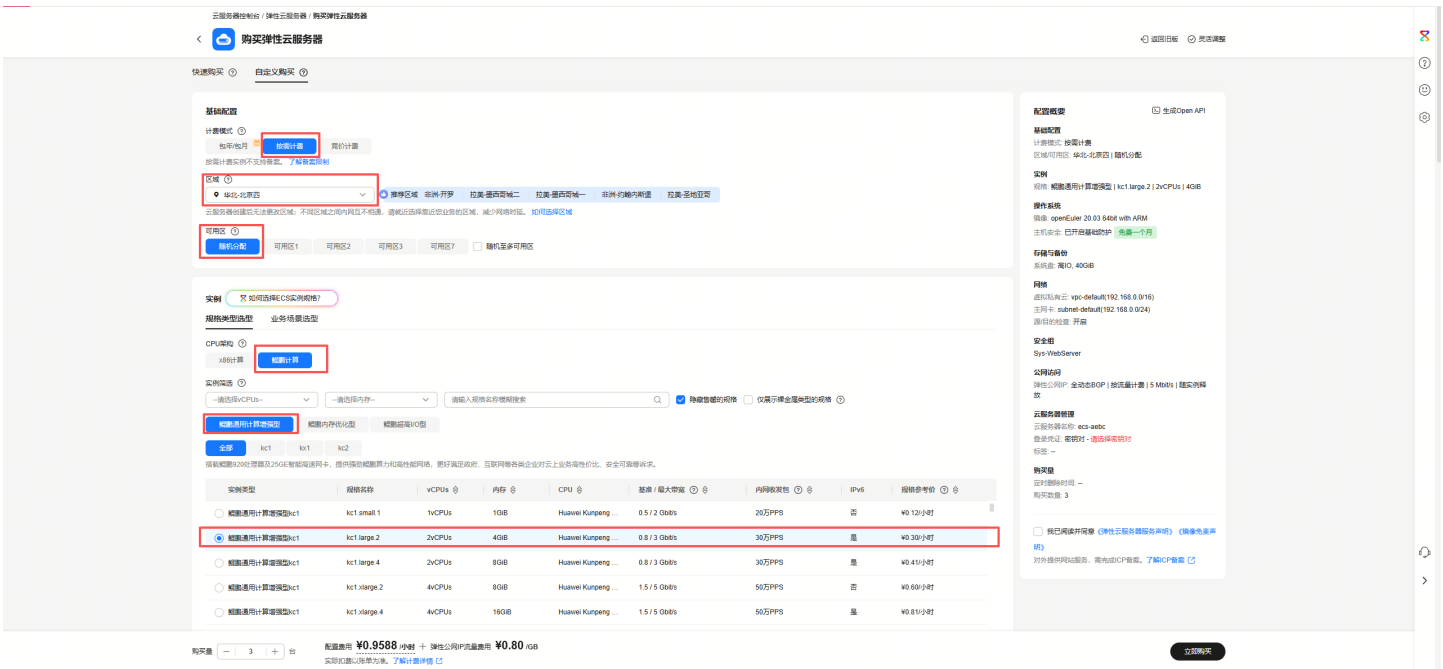
鼠标悬停在网页右上角的用户区域，点击“费用与成本”进入对应页面

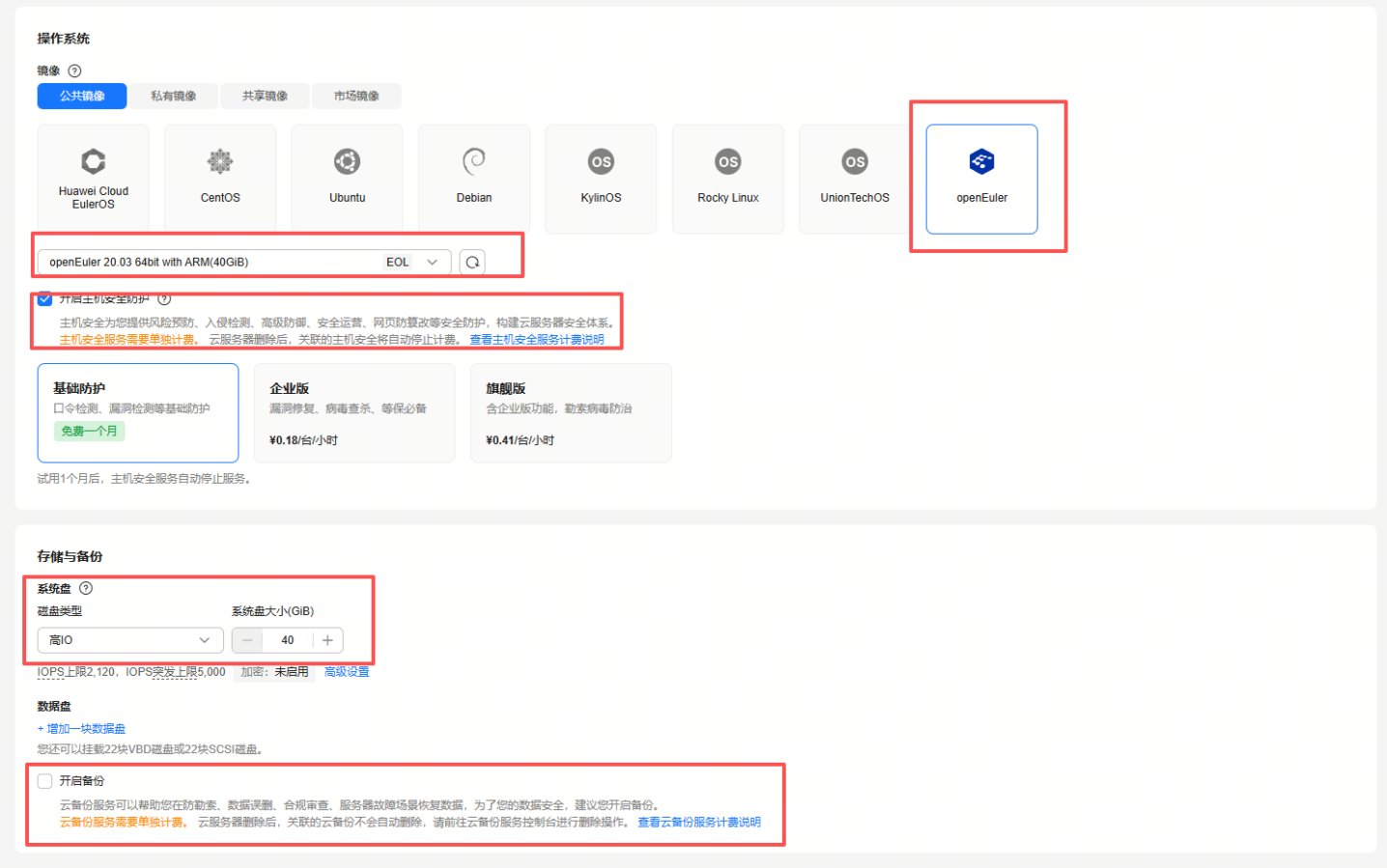


可看到如上图所示的到账代金券，即可开始购买实验所需服务器资源

1.3 购买弹性云服务器（ECS）

浏览器访问https://console.huaweicloud.com/ecm/?region=cn-north-4#/ecs/create/custom进行弹性云服务器的购买，具体配置请参考下图，尤其注意红框区域的选择。其中**安全组选择Default可以开启22端口，否则可能会造成ssh连接失败的问题。**




操作系统
镜像
公共镜像
私有镜像
共享镜像
市场镜像
Huawei Cloud EulerOS
CentOS
Ubuntu
Debian
KylinOS
Rocky Linux
UnionTechOS
openEuler
openEuler 20.03 64bit with ARM(40GiB)
EOL
开启主机安全防护
主机安全为您提供风险预防、入侵检测、高级防御、安全运营、网页防篡改等安全防护，构建云服务器安全体系。
主机安全服务需要单独计费。 云服务器删除后，关联的主机安全将自动停止计费。 查看主机安全服务计费说明
基础防护
口令检测、漏洞检测等基础防护
免费一个月
企业版
漏洞修复、病毒查杀、等保必备
¥0.18/台/小时
旗舰版
含企业版功能，勒索病毒防治
¥0.41/台/小时
试用1个月后，主机安全服务自动停止服务。
存储与备份
系统盘
磁盘类型
系统盘大小(GiB)
高IO
40
IOPS上限2,120，IOPS突发上限5,000
加密：未启用
高级设置
数据盘
+ 增加一块数据盘
您还可以挂载22块VBD磁盘或22块SCSI磁盘。
开启备份
云备份服务可以帮助您在防勒索、数据误删、合规审查、服务器故障场景恢复数据，为了您的数据安全，建议您开启备份。
云备份服务需要单独计费。 云服务器删除后，关联的云备份不会自动删除，请前往云备份服务控制台进行删除操作。 查看云备份服务计费说明

网络

虚拟私有云

vpc-default(192.168.0.0/16)

新建虚拟私有云

主网卡

subnet-default(192.168.0.0/24)

自动分配IP地址

可用私有IP数量250个

+ 新增扩展网卡

您还可以增加 1 块网卡

源/目的检查

安全组

选择安全组

default(f4212df7-7237-460d-9a84-b05691cbca0a)

新建安全组

请确保所选安全组已放通22端口（Linux SSH登录），3389端口（Windows远程登录）和 ICMP 协议（Ping）。配置安全组规则

建议将安全组入方向规则中高危端口的源地址设置为已知IP地址、安全组或IP地址组，避免因网络入侵出现业务中断、数据泄露或数据勒索等严重后果。查看高危端口

展开安全组规则

公网访问

弹性公网IP

现在购买 | 使用已有 | 暂不购买

线路

全动态BGP | 静态BGP

不低于99%可用性保障

公网带宽

按带宽计费
流量较大或较稳定的场景

按流量计费
流量小或流量波动较大场景

加入共享带宽
多业务流量错峰分布场景

弹性公网IP的计费模式为按需计费，依照实际出云方向的流量收取流量费，带宽大小仅做限速使用，不作为收费依据。
弹性公网IP与实例解绑后，会停止收取流量费，同时开始收取弹性公网IP的保有费。 查看弹性公网IP计费说明

带宽大小（Mbit/s）

5 | 10 | 20 | 50 | 100 | 自定义

带宽范围：1-300 Mbit/s；开启DDoS基础防护 免费

释放行为

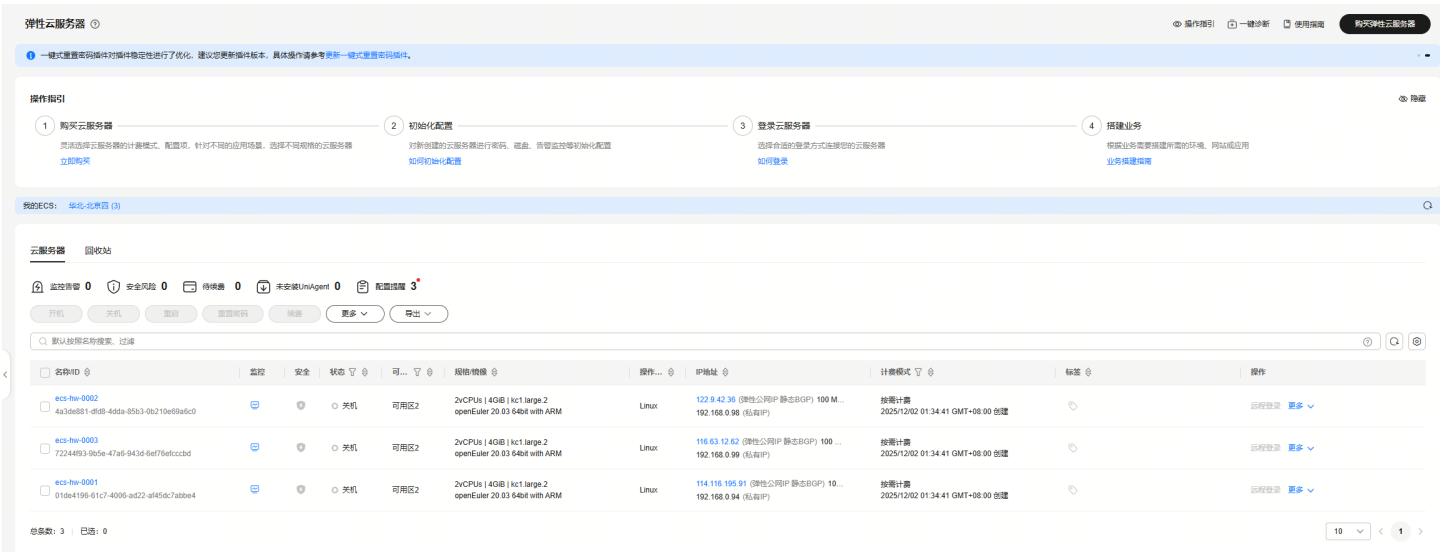
随实例释放

对于设置了随实例释放的弹性公网IP，将在删除云服务器同时执行删除。

1.4 检查已购买服务器

购买完成后，可在控制台->弹性云服务器中找到已购买的服务器。参考网址为：https://console.huaweicloud.com/ecm/?agencyId=394b9e423885455181f43b723263a893&region=cn-north-4&locale=zh-cn#/ecs/manager/vmList

如下图所示，检查无误后可以进行远程连接和环境的配置工作了



2. 远程连接与环境配置

2.1 ssh工具的下载与使用

可选择的工具有Xshell (Windows)、vscode + ssh remote 插件、Royal TSX (mac)，mac OS 的 terminal 自带 ssh 功能，或从微软应用商店中下载安装 Ubuntu 子系统 （Windows10）。

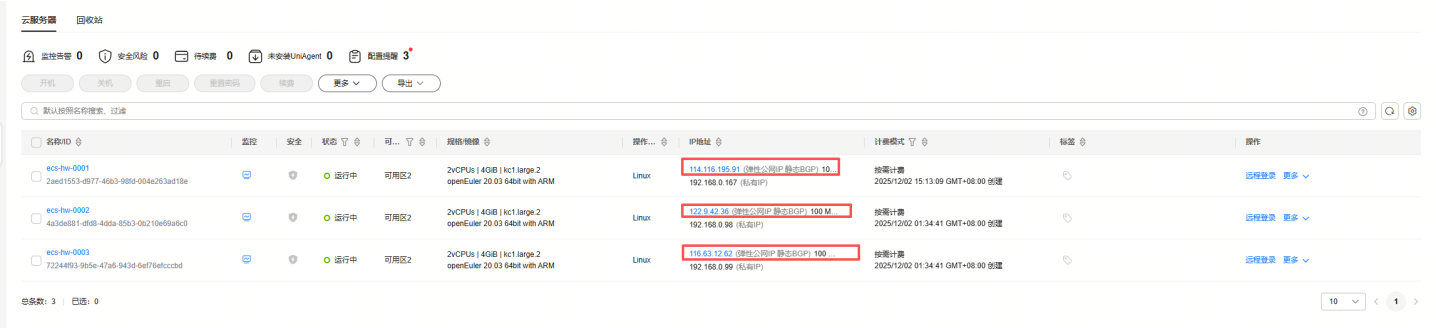
以XShell为例

Xshell免费版官网下载地址：https://www.xshell.com/zh/free-for-home-school/

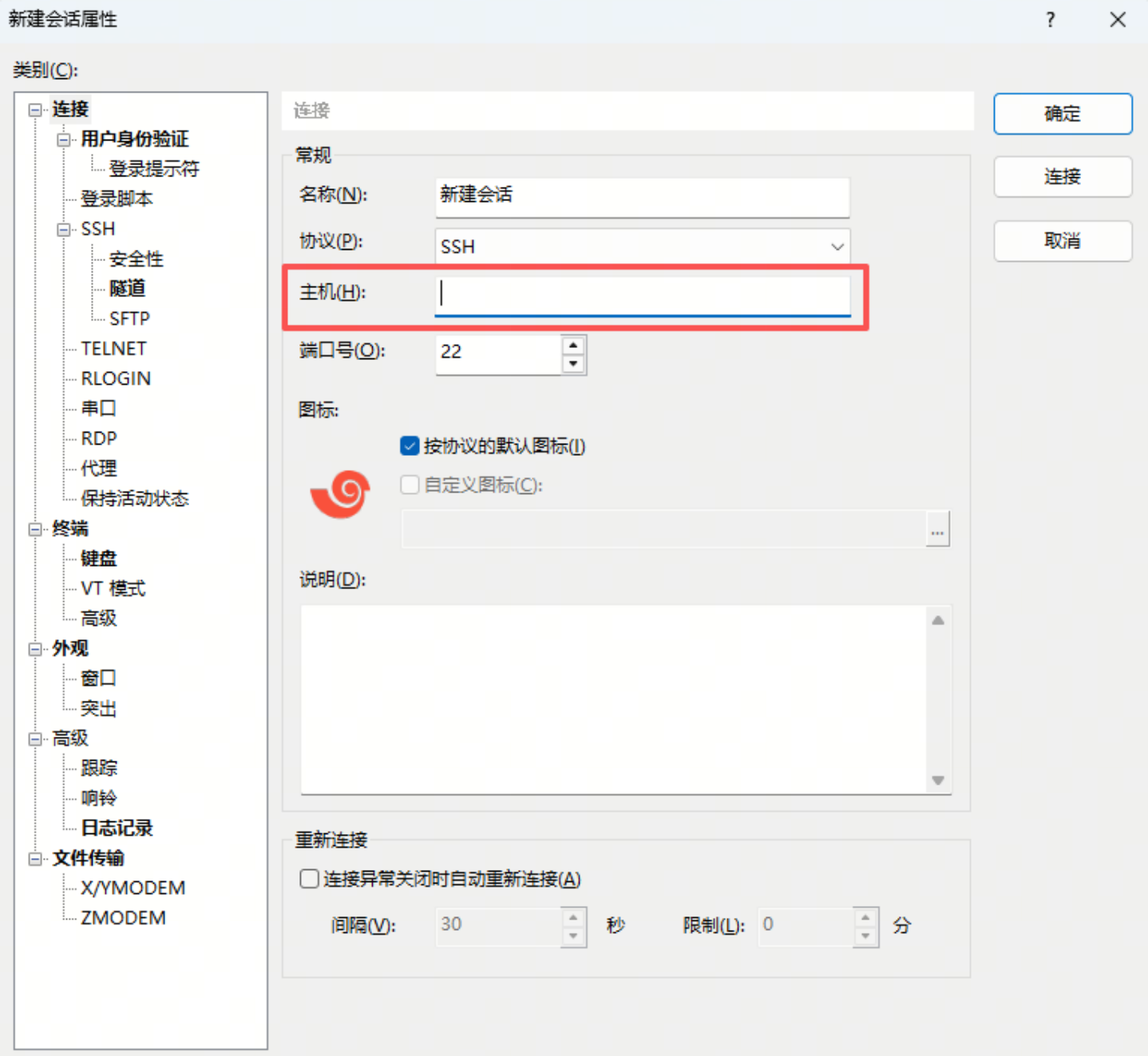
下载完成后，输入用户名和邮箱进行免费注册即可使用。

右上角**文件->新建会话**

主机名为控制台中的**弹性公网IP**，如下图所示



将**弹性公网IP**输入到主机中，名称建议根据服务器号自定义，比如001，002，003。随后点击右侧**连接**。



初次用户名为**root**，密码为**购买服务器时设置的密码**。

输入密码连接后，出现下图所示说明连接成功，可以进行后续环境配置操作

```
Xshell 8 (Build 0086)
Copyright (c) 2024 NetSarang Computer, Inc. All rights reserved.

Type `help' to learn how to use Xshell prompt.
[C:\~]$

Connecting to 114.116.195.91:22...
Connection established.
To escape to local shell, press 'Ctrl+Alt+]'.

Authorized users only. All activities may be monitored and reported.

WARNING! The remote SSH server rejected X11 forwarding request.

        Welcome to Huawei Cloud Service

Last failed login: Tue Dec  2 15:26:29 CST 2025 from 157.66.144.16 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Tue Dec  2 15:16:37 2025 from 221.238.245.39


Welcome to 4.19.90-2110.8.0.0119.oe1.aarch64

System information as of time:  Tue Dec  2 15:28:39 CST 2025

System load:    0.00
Processes:      144
Memory used:    9.4%
Swap used:      0.0%
Usage On:       8%
IP address:     192.168.0.167
Users online:   1


[root@ecs-hw ~]#
```

2.2 创建新用户

为了防止大家的文件混乱，建议大家在每一台机器下都建立个人账户，不建议统一使用 root 账户。**三台机器都需要执行接下来的创建用户和免密配置操作，**因此建议在Xshell中同时打开三个 shell 窗口，对应三台远程服务器进行操作。每个窗口都需要进行一次同样的操作，只是对应的IP不同。注意此处需要的IP是私网IP，如下图所示。

代码块

```
adduser zhangsan
passwd zhangsan
usermod -aG wheel zhangsan
```

说明：建议zhangsan的密码也设置为购买服务器时设置的密码

2.3 免密配置

注意以下步骤也需要在三台机器，也就是三个窗口上分别进行一次。

1. 配置三台机器主机名和ip解析

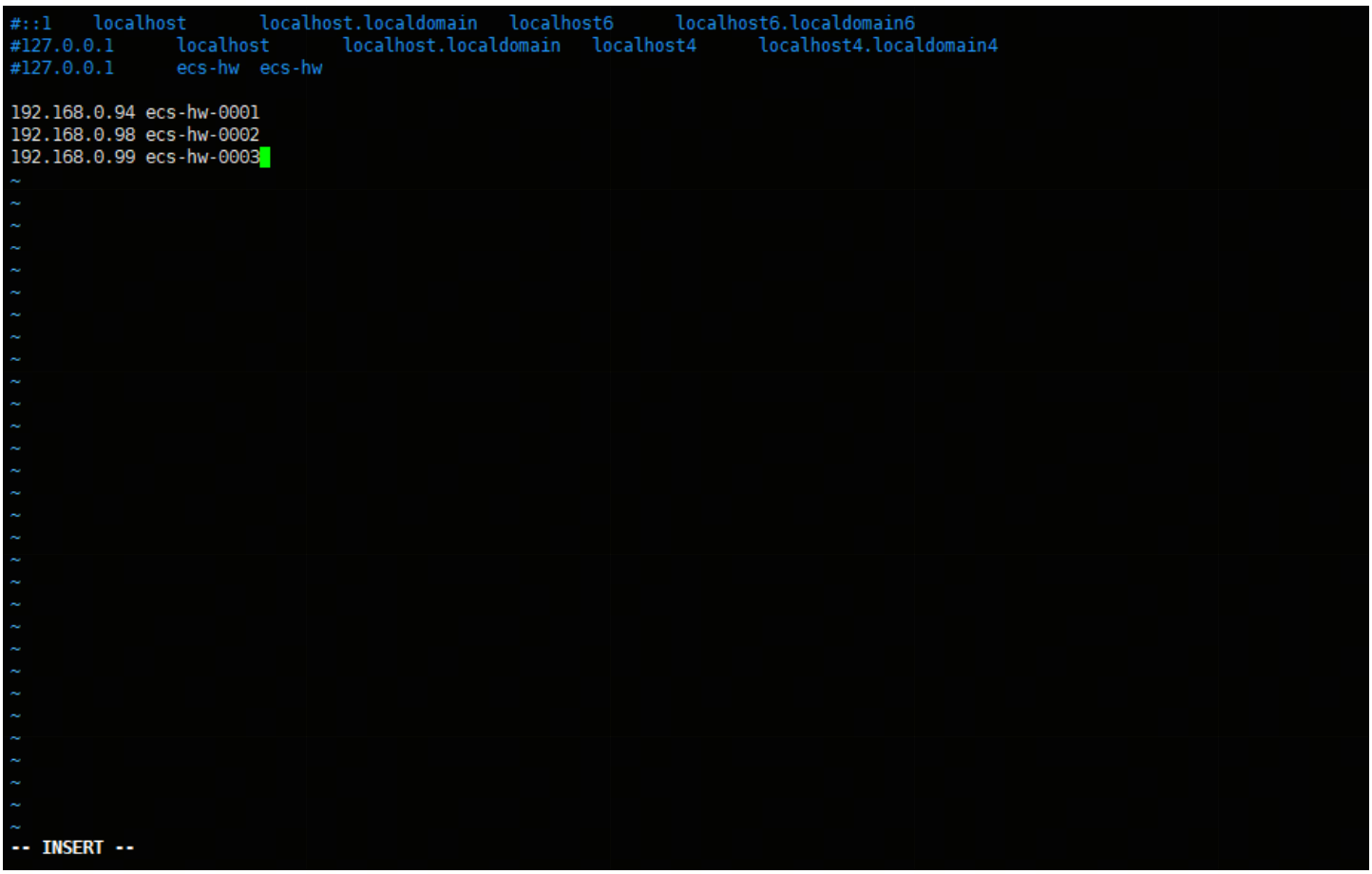
命令行输入

```
代码块
1  vim /etc/hosts
```

进入文本界面后，输入'I'进入Insert模式。接下来需要使用#注释掉上面原有的内容，并加入三台机器的“私网IP主机”信息（**每个人不一样**，注释的目的是因为会对本实验程序的运行产生报错），建议现在自己电脑上输入好自己三台服务器的完整信息，在三个窗口中粘贴（Shift+Ctrl+V)即可。下面是一个例子，要以自己的服务器私网IP为准。

```
代码块
1   192.168.0.94 ecs-hw-0001
2   192.168.0.98 ecs-hw-0002
3   192.168.0.99 ecs-hw-0003
```

添加完后信息如下，随后Esc退出编辑，输入:wq 保存并退出即可。



2. 登录新账户

首先退出root账户重新登录到新建立的账户下：

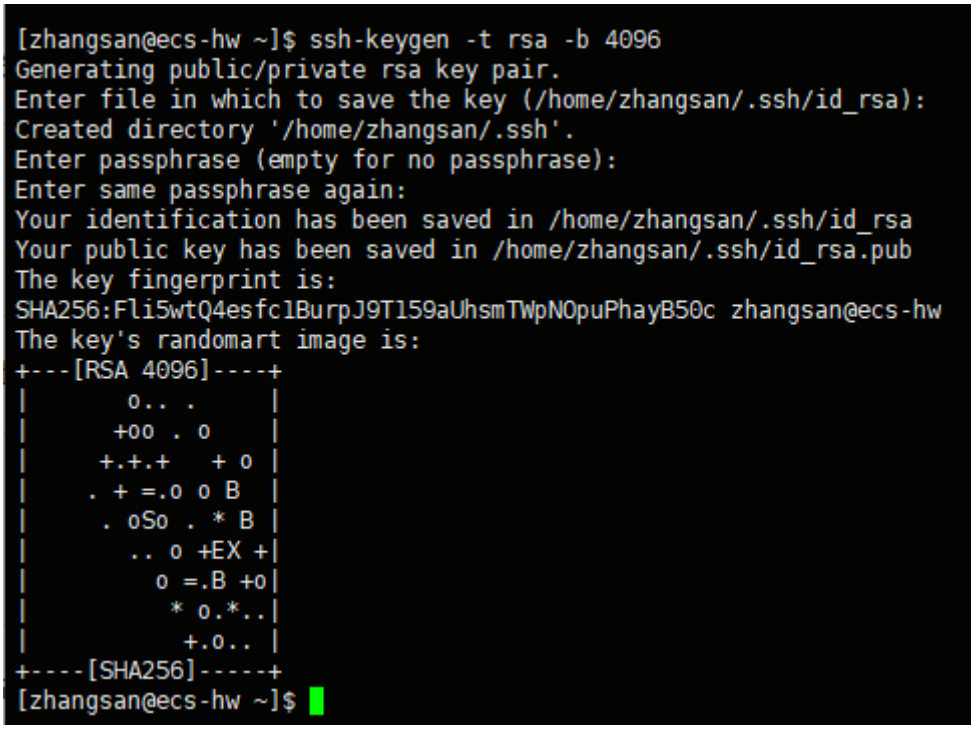
```
代码块
1   su - zhangsan
```

本地生成密钥：

代码块

```
ssh-keygen -t rsa -b 4096
```

此命令会要求输入信息，三次回车即可



```
[zhangsan@ecs-hw ~]$ ssh-keygen -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/home/zhangsan/.ssh/id_rsa):
Created directory '/home/zhangsan/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/zhangsan/.ssh/id_rsa
Your public key has been saved in /home/zhangsan/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:Fli5wtQ4esfc1BurpJ9T159aUhsmTWpNOpuPhayB50c zhangsan@ecs-hw
The key's randomart image is:
+---[RSA 4096]----+
|       o.. .     |
|      +oo . o    |
|     +.+.+   + o |
|    . + =.o o B  |
|     . oSo . * B |
|       .. o +EX +|
|         o =.B +o|
|          * o.*..|
|           +.o.. |
+----[SHA256]-----+
[zhangsan@ecs-hw ~]$
```

3. 添加公钥至所有主机（根据提示输入yes、密码）（这一步得在三台机器都完成上一步操作后进行）：

代码块

```
ssh-copy-id zhangsan@ecs-hw-0001
ssh-copy-id zhangsan@ecs-hw-0002
ssh-copy-id zhangsan@ecs-hw-0003
```

可参考下图检查是否成功配置免密登录

```
[zhangsan@ecs-hw ~]$ ssh-copy-id zhangsan@ecs-hw-0002
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/home/zhangsan/.ssh/id_rsa.pub"
The authenticity of host 'ecs-hw-0002 (192.168.0.98)' can't be established.
ECDSA key fingerprint is SHA256:iYuQy0QXQiBk6mhJVKYY7tIksXMzZ8RTR7zvAIKt78U.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
The authenticity of host 'ecs-hw-0002 (192.168.0.98)' can't be established.
ECDSA key fingerprint is SHA256:iYuQy0QXQiBk6mhJVKYY7tIksXMzZ8RTR7zvAIKt78U.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys

Authorized users only. All activities may be monitored and reported.
zhangsan@ecs-hw-0002's password:

Number of key(s) added: 1

Now try logging into the machine, with:   "ssh 'zhangsan@ecs-hw-0002'"
and check to make sure that only the key(s) you wanted were added.

[zhangsan@ecs-hw ~]$ ssh 'zhangsan@ecs-hw-0002'

Authorized users only. All activities may be monitored and reported.

        Welcome to Huawei Cloud Service

Last login: Tue Dec  2 16:17:04 2025


Welcome to 4.19.90-2110.8.0.0119.oe1.aarch64

System information as of time:  Tue Dec  2 16:24:08 CST 2025

System load:    0.00
Processes:      151
Memory used:    10.2%
Swap used:      0.0%
Usage On:       8%
IP address:     192.168.0.98
Users online:   2


[zhangsan@ecs-hw-0002 ~]$
```

检查无误后可以命令行输入exit回退到原来的服务器

2.4 安装依赖包

在每台服务器的zhangsan下进行

代码块

```
sudo yum -y install gcc-c++ gcc-gfortran
```

代码块

```
wget http://www.mpich.org/static/downloads/3.3.2/mpich-3.3.2.tar.gz
tar -zxvf mpich-3.3.2.tar.gz
cd mpich-3.3.2
./configure
sudo make
sudo make install
```

HelloWorld实验示例

1. 创建示例程序源码

执行以下命令，创建 hello 目录存放该程序的所有文件, 并进入 hello 目录（**三台主机都执行**）

代码块

```
mkdir /home/zhangsan/hello
cd /home/zhangsan/hello
```

执行以下命令，创建示例程序源码 mpi_hello_world.c（**三台主机都执行**）,代码内容如下：

代码块

```
vim mpi_hello_world.c
```

代码块

```c
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv) {
// Initialize the MPI environment. The two arguments to MPI Init are not
// currently used by MPI implementations, but are there in case future
// implementations might need the arguments.
MPI_Init(NULL, NULL);
// Get the number of processes
int world_size;
MPI_Comm_size(MPI_COMM_WORLD, &world_size);
// Get the rank of the process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
// Get the name of the processor
char processor_name[MPI_MAX_PROCESSOR_NAME];
int name_len;
MPI_Get_processor_name(processor_name, &name_len);
// Print off a hello world message
printf("Hello world from processor %s, rank %d out of %d processors\n",
processor_name, world_rank, world_size);
// Finalize the MPI environment. No more MPI calls can be made after this
MPI_Finalize();
}
```

同样使用I进入Insert模型，使用:wq保存退出。

```
1 0001   2 0002   3 0003   +
#include <mpi.h>
#include <stdio.h>
int main(int argc, char** argv) {
// Initialize the MPI environment. The two arguments to MPI Init are not
// currently used by MPI implementations, but are there in case future
// implementations might need the arguments.
MPI_Init(NULL, NULL);
// Get the number of processes
int world_size;
MPI_Comm_size(MPI_COMM_WORLD, &world_size);
// Get the rank of the process
int world_rank;
MPI_Comm_rank(MPI_COMM_WORLD, &world_rank);
// Get the name of the processor
char processor_name[MPI_MAX_PROCESSOR_NAME];
int name_len;
MPI_Get_processor_name(processor_name, &name_len);
// Print off a hello world message
printf("Hello world from processor %s, rank %d out of %d processors\n",
processor_name, world_rank, world_size);
// Finalize the MPI environment. No more MPI calls can be made after this
MPI_Finalize();
}
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
"mpi_hello_world.c" [readonly] 23L, 868C                                    23,1          All
```

2. 创建makefile

执行以下命令，创建 makefile（**三台主机都执行**）‘

代码块

```
vim makefile
```

代码块

```
EXECS = mpi_hello_world
MPICC ?= mpicc

all: ${EXECS}

mpi_hello_world: mpi_hello_world.c
	${MPICC} -o mpi_hello_world mpi_hello_world.c

clean:
	rm -f ${EXECS}

```

```
EXECS = mpi_hello_world
MPICC ?= mpicc

all: ${EXECS}

mpi_hello_world: mpi_hello_world.c
	${MPICC} -o mpi_hello_world mpi_hello_world.c

clean:
	rm -f ${EXECS}

"makefile" 11L, 162C                                                    7,1-8         All
```

3. 进行编译

执行以下命令，进行编译（**三台主机都执行**）

代码块

```
cd /home/zhangsan/hello
make
```

```
[root@ecs-hw-0003 hello]# make
mpicc -o mpi_hello_world mpi_hello_world.c
[root@ecs-hw-0003 hello]# ls
makefile  mpi_hello_world  mpi_hello_world.c
[root@ecs-hw-0003 hello]#
```

4. 建立主机配置文件

执行以下命令，建立主机配置文件（**三台主机都执行**）

代码块

```
vim /home/zhangsan/hello/config
```

添加内容为：

代码块

```
ecs-hw-0001:2
ecs-hw-0002:2
ecs-hw-0003:2
```

5. 运行监测

执行以下命令，查看运行结果（只需要在 ecs-hw-0001 上执行），本次演示的ecs-hw-0001实际为ecs-hw，因此下图为ecs-hw。对实验结果无影响。

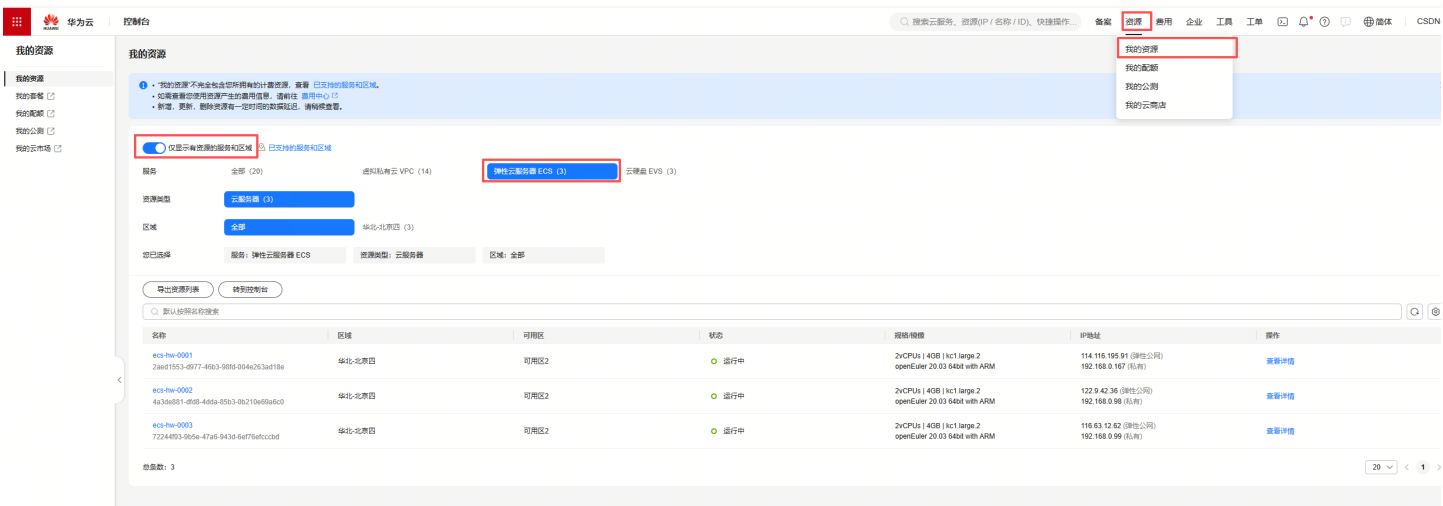
```
[zhangsan@ecs-hw hello]$ mpiexec -n 6 -f /home/zhangsan/hello/config /home/zhangsan/hello/mpi_hello_world

Authorized users only. All activities may be monitored and reported.

Authorized users only. All activities may be monitored and reported.
Hello world from processor ecs-hw-0002, rank 2 out of 6 processors
Hello world from processor ecs-hw-0003, rank 4 out of 6 processors
Hello world from processor ecs-hw-0002, rank 3 out of 6 processors
Hello world from processor ecs-hw-0003, rank 5 out of 6 processors
Hello world from processor ecs-hw, rank 0 out of 6 processors
Hello world from processor ecs-hw, rank 1 out of 6 processors
[zhangsan@ecs-hw hello]$
```

通过上述代码运行，可以看出编写的 hello-wolrd 程序已经在华为鲲鹏上运行起来，程序在集群之间并行计算处理。

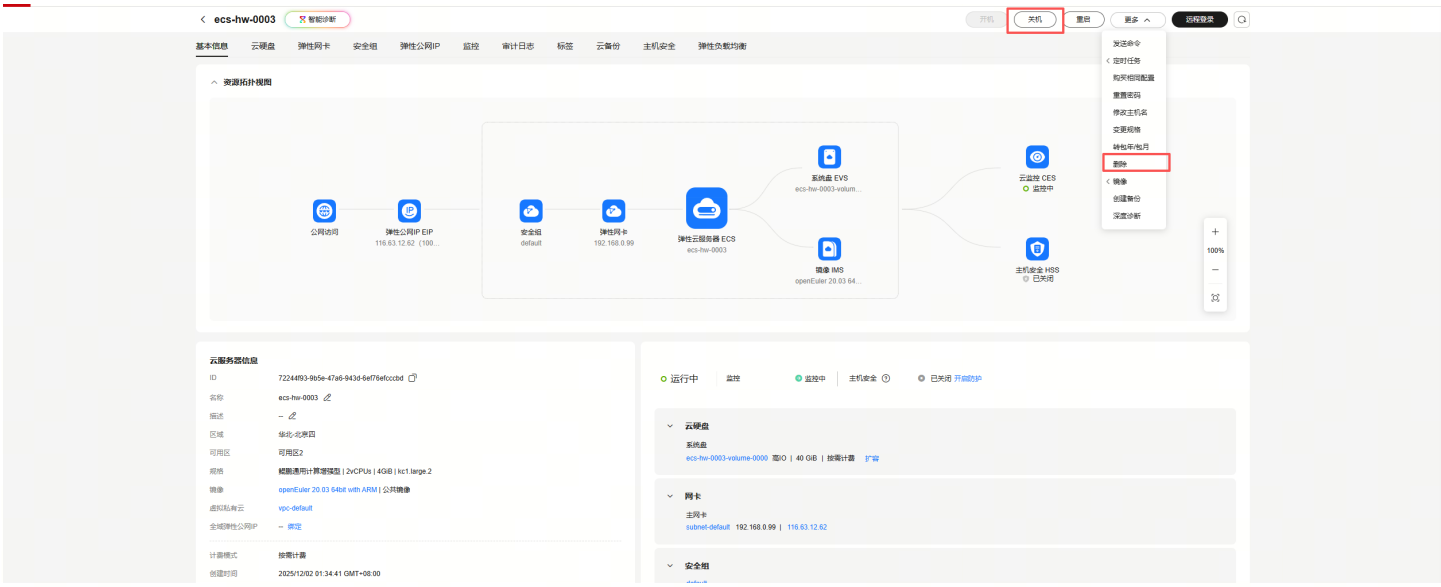
华为云资源的关停

当我们暂时不用服务器资源或者实验完成后，需要**先停止相关服务并删除所有资源**

具体操作是在右上角”资源“->"我的资源"中，打开“仅显示有资源的服务”按钮，逐个点击具体服务，在下方服务资源表格“操作”列中进入“查看详情”界面，进行各服务的停止、删除操作。（资源删除，数据也将被清除，操作前请务必先确认数据是否可清除）。



如“弹性云服务器 ECS”服务实例，进入“查看详情”界面后，可以对服务实例进行关机（**暂停计费**）和删除（**所有实验完成后，清除环境和数据**）。如下图所示：



请同学们务必检查“我的资源”，如果还存在任何 1 项资源，务必关停并删除，直到没有任何资源为止。