

MPI 编程实验环境配置及示例

一、 MPI 编程实验流程概述

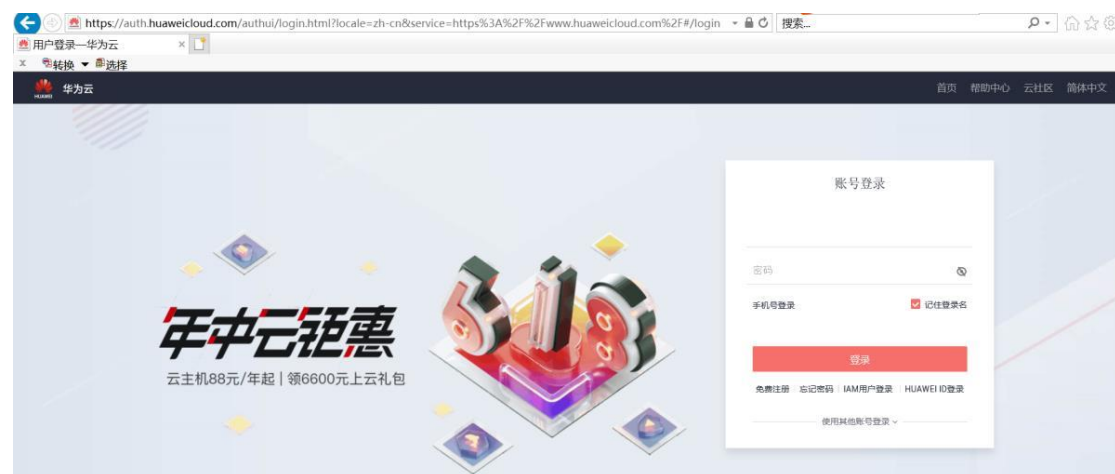
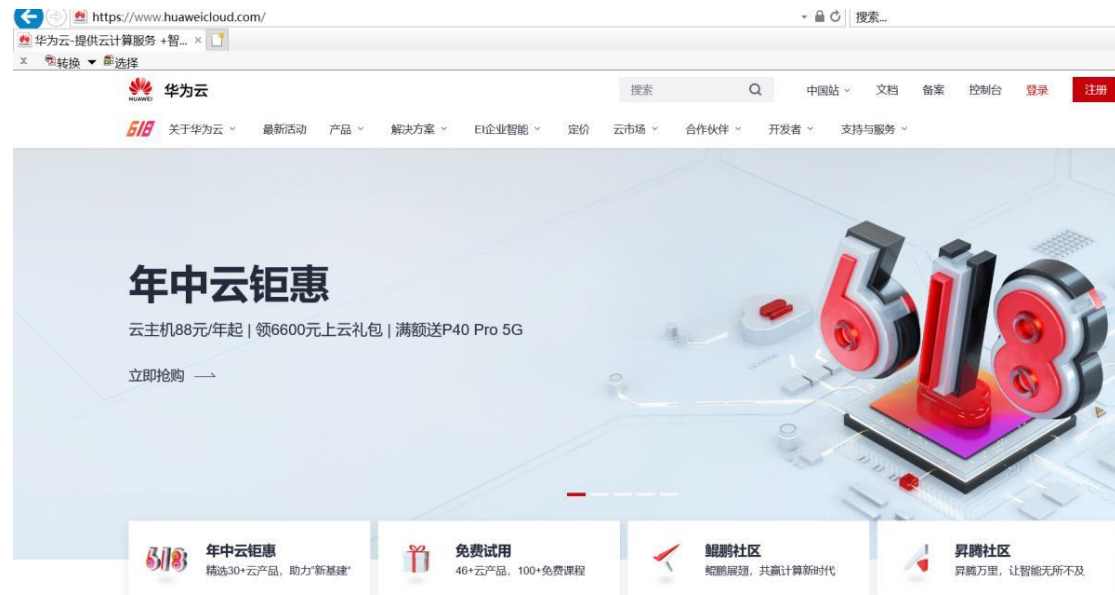
1. 购买三台华为鲲鹏云服务器及必要的网络资源，参照“二、 实验环境配置”中的“1. 购买华为鲲鹏云服务器”； 注意请参照本文档配置购买资源，不要购买价格较高的额外资源，如果在实验、作业期间，代金券快用完了，请及时联系老师或助教；
2. 三台机器均使用 ssh 连接，并进行免密配置，安装 mpi 及依赖包，参照“二、 实验环境配置”中的“2. 环境配置”；
3. 进行 HelloWorld 实验，验证环境安装正确，参照“三、 HelloWorld 实验示例”；
4. 完成所有实验、作业后，**必须停止/关闭所有华为云资源**，方法参照“四、 停止/关闭华为云资源”，否则将会继续扣费导致欠费！

二、实验环境配置

1. 购买华为鲲鹏云服务器

步骤 1 登录华为云

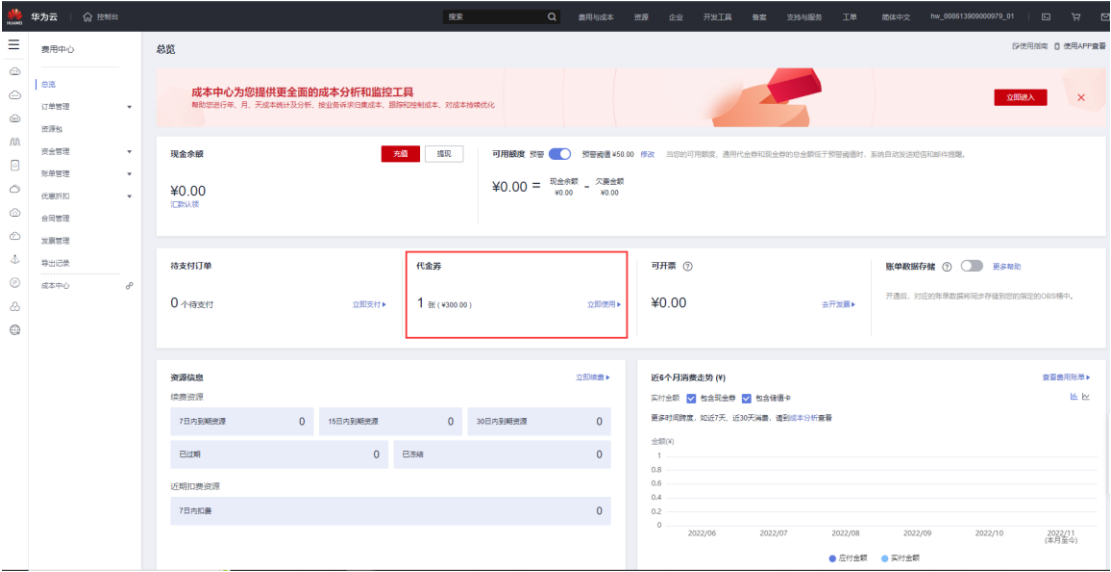
打开浏览器，输入华为云的域名：<https://www.huaweicloud.com>，点击右上角登录按钮，输入用户名与密码。



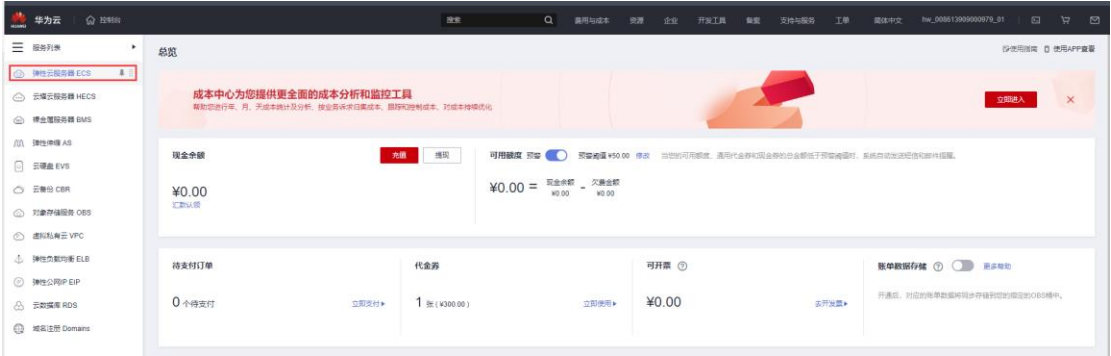
点击费用与成本



确认代金券到账。



点击左侧的云服务器图标。



步骤 2 购买弹性云服务器（ECS）

点击“购买弹性云服务器”。



步骤 3 基础配置

按照下表配置进行购买，一共购买 3 台。

计费模式	区域	CPU架构	规格	镜像	系统盘
按需计费	华北-北京四	鲲鹏计算	kc1.large.2	公共镜像： openEuler 20.03	至少40GB

参照截图如下：

购买弹性云服务器

返回

帮助

1 基础配置

2 网络配置

3 高级配置

4 确认订单

区域

华北-北京四

选择区域

华北-乌兰察布一

西南-贵阳一

华北-北京四

华南-广州

华东-上海一

不同区域的云服务器产品之间规格不可混选。请前往该地域查看您选择的区域，可减少跨地域、跨架构部署。如需跨地域、跨架构部署，如跨区域。

计费方式

包年包月

按需计费

竞价计费

可用区

默认分配

可用区1

可用区2

可用区3

可用区7

实例类型

规格类型选型

业务场景选型

CPU架构

x86计算

鲲鹏计算

规格

最新系列

vCPU

-请选择vCPU-

内存

-请选择内存-

规格名称

搜索

快速查找规格

A+加速型

鲲鹏通用计算增强型

数据库优化型

裸金属GPU型

全选

共1项

收起筛选

搭载最新的96核处理器及256GB智能高速网卡，提供强劲的计算力与高性价比性能，更好满足政府、互联网等各类企业云上业务高性价比、安全可靠需求。

规格名称	vCPU 12	内存	CPU 12	基频 / 最大频率	I/O	内网收敛带宽	IOPS	规格参考价
<input type="radio"/> kc1.xsmall.1	1vCPUs	1GB	Huawei Kunpeng 920 2.6GHz	0.5 / 2 Gbits	20TPPS			\$0.12/小时
<input checked="" type="radio"/> kc1.large.2	2vCPUs	4GB	Huawei Kunpeng 920 2.6GHz	0.8 / 3 Gbits	30TPPS			\$0.30/小时
<input type="radio"/> kc1.xlarge.4	2vCPUs	8GB	Huawei Kunpeng 920 2.6GHz	0.8 / 3 Gbits	30TPPS			\$0.41/小时
<input type="radio"/> kc1.xlarge.2	4vCPUs	8GB	Huawei Kunpeng 920 2.6GHz	1.5 / 5 Gbits	50TPPS			\$0.60/小时
<input type="radio"/> kc1.xlarge.4	4vCPUs	16GB	Huawei Kunpeng 920 2.6GHz	1.5 / 5 Gbits	50TPPS			\$0.81/小时
<input type="radio"/> kc1.2xlarge.2	8vCPUs	16GB	Huawei Kunpeng 920 2.6GHz	3 / 7 Gbits	80TPPS			\$1.20/小时
<input type="radio"/> kc1.2xlarge.4	8vCPUs	32GB	Huawei Kunpeng 920 2.6GHz	3 / 7 Gbits	80TPPS			\$1.63/小时

当前规格：鲲鹏通用计算增强型 | kc1.large.2 | 2vCPU | 4GB

镜像

公共镜像

私有镜像

共享镜像

市场镜像

安全组

选择 sg-en Euler

sg-en Euler 20.03 64bit with ARM-64Gbit

安全防护

☒ 启用开启主机安全策略防护

☐ 购买虚拟IP

☐ 不使用安全防护

系统盘

高IO

40

+ 10

购买数量

1

+

-

件

配置费用 ¥0.9588/小时

下一步：网络配置

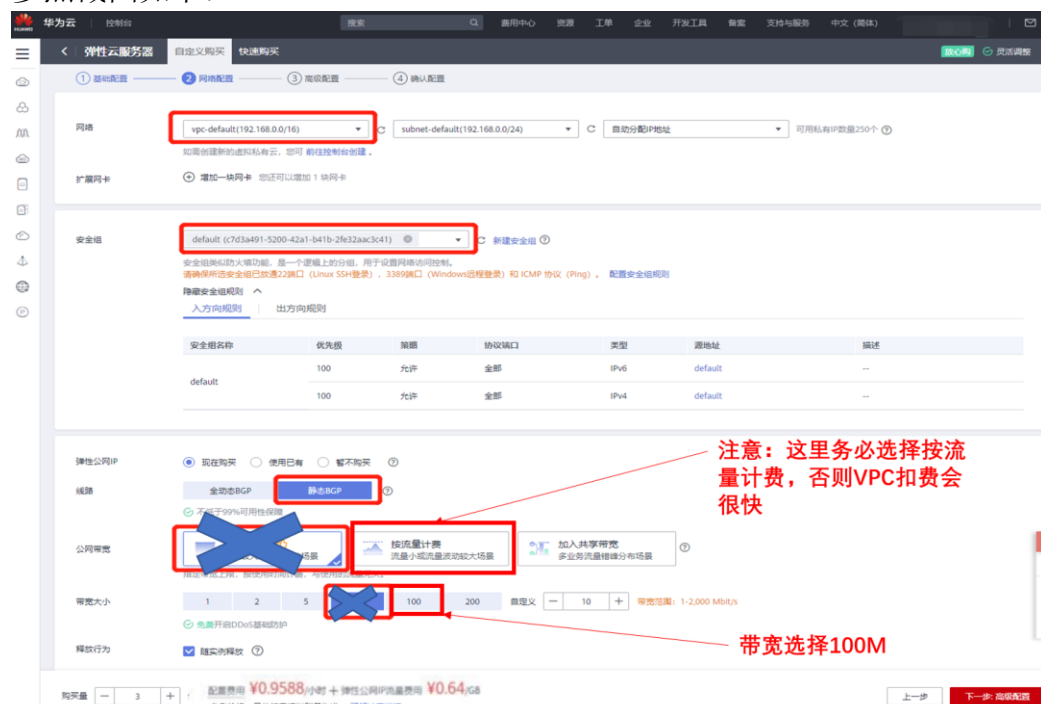
点击下一步:网络配置。

步骤 4 网络配置

按照下表进行网络配置

网络	安全组	弹性公网IP
默认的VPC	default	现在购买，带宽大小选择 100Mbps/s.

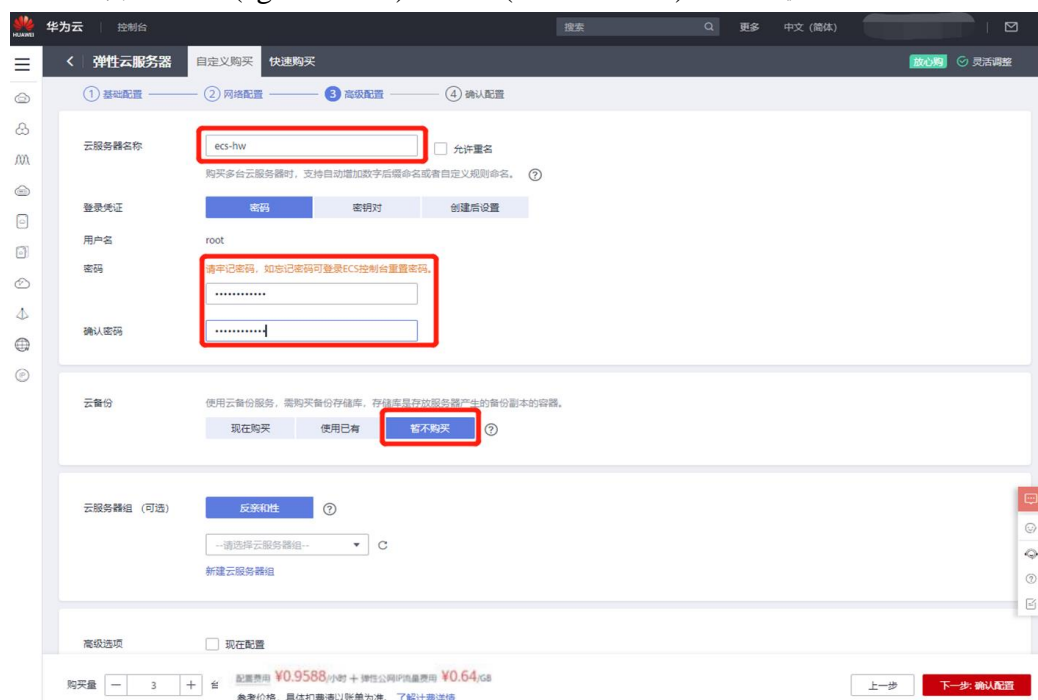
参照截图如下：



点击“下一步 高级配置”。

步骤 5 高级配置

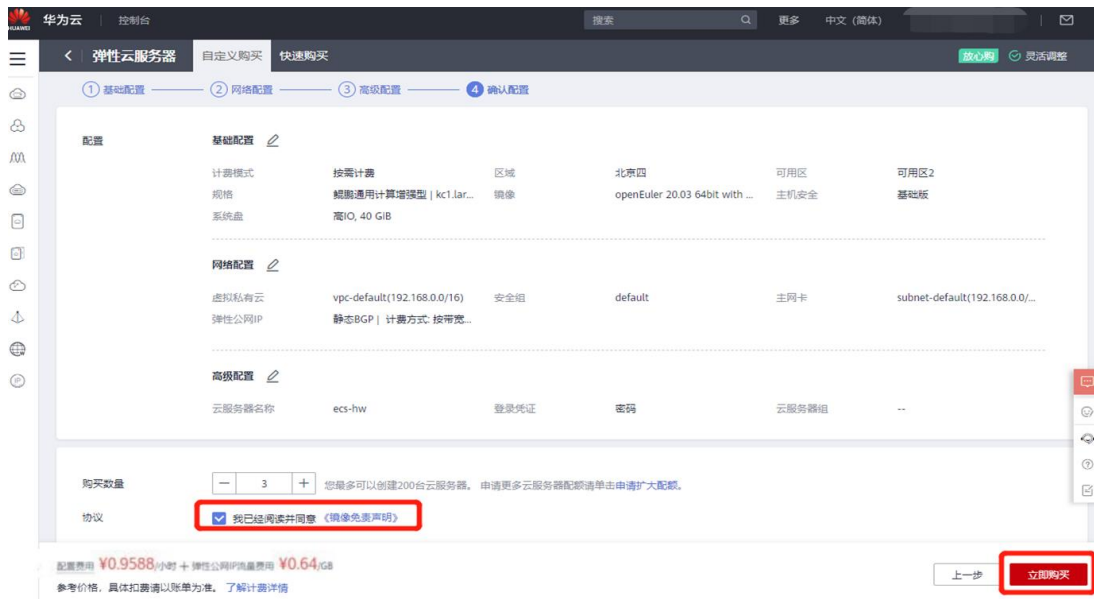
设置云服务器名称(eg: ecs-hw)、密码(Parallel2022)，云备份选“暂不购买”。



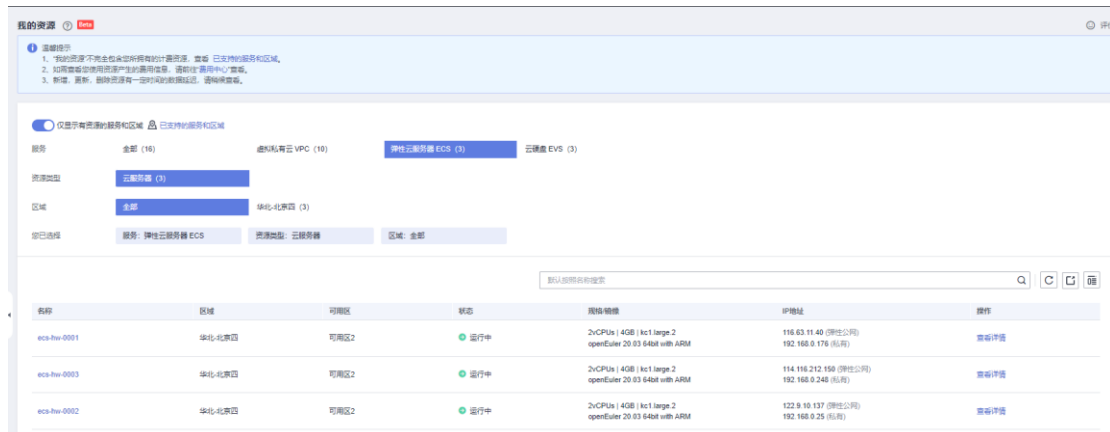
点击“下一步：确认配置”。

步骤 6 确认配置

勾选我已阅读并同意《华为镜像免责声明》，点击确认配置。



购买成功后可去“我的资源”中查看资源，如下图查看 弹性云服务器资源。



注意：此时“我的资源”中除了刚刚购买的“弹性云服务器 ECS”之外，还会由“虚拟私有云 VPC”、“云硬盘 EVS”等资源，等完成实验、作业之后，一定要把所有资源全部关停（参照“四”部分内容），否则代金券扣完还会继续扣费，你将欠费！

2. 环境配置

步骤 1 下载 ssh 工具

推荐 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）。

步骤 2 登录 ECS

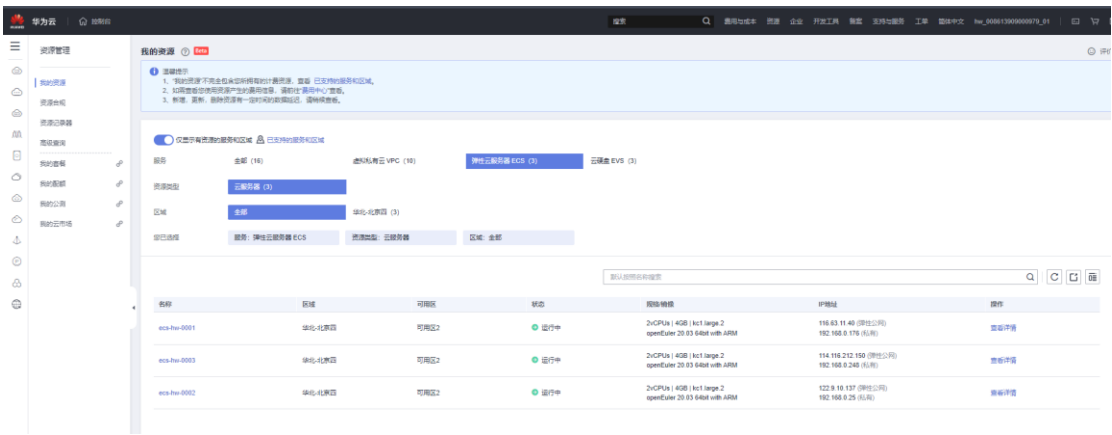
创建好 ECS 后，可以在控制台 -> 弹性云服务器中看到弹性公网 IP。使用 SSH 工具，输入公网 IP、用户名和密码，或 `ssh usr@IP` 即可登陆，例如 `ssh root@116.63.11.40`，再根据提示输入密码，即可登录 ecs-hw-0001。

信息如下（每个人不一样）：

名称	区域	可用区	状态	规格详情	IP地址	操作
ecs-hw-0001	华北-北京四	可用区2	运行中	2vCPUs 4GB 1c1.large.2 openEuler 20.03 64bit with ARM	192.168.0.176 (私有) 192.168.0.176 (公网)	查看详情
ecs-hw-0002	华北-北京四	可用区2	运行中	2vCPUs 4GB 1c1.large.2 openEuler 20.03 64bit with ARM	192.168.0.248 (私有) 192.168.0.248 (公网)	查看详情
ecs-hw-0003	华北-北京四	可用区2	运行中	2vCPUs 4GB 1c1.large.2 openEuler 20.03 64bit with ARM	192.168.0.137 (私有) 192.168.0.137 (公网)	查看详情

步骤 3 环境配置说明

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，下面以用户名 zhangsan 为例，三台机器名分别为 ecs-hw-0001, ecs-hw-0002, ecs-hw-0003，三台机器的私网 ip 为 192.168.0.176, 192.168.0.25, 192.168.0.248,。环境配置时在每一台主机上都需要重复执行，步骤 4 和步骤 5 三台机器都需要执行，ip 需要根据实际分配进行相应调整，建议打开三个 shell 窗口同时进行操作。



步骤 4 创建用户

每台主机都需要在 root 账户下建立相同的个人账户 zhangsan，并添加到 wheel 用户组（改组具有 sudo 权限），命令如下：

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

说明：zhangsan 密码也设置为 Parallel2022

步骤 5 免密配置

以下所有步骤在三台机器上均需要重复执行。

(1)配置三台机器主机名和 ip 解析（各主机 ip 可以通过 ifconfig 或者控制台界面查看）

```
vim /etc/hosts
```

注释文件原先本身的信息并添加三台机器的“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），添加完后信息如下：

```
#:1 localhost localhost.localdomain localhost6 localhost6.localdomain6 注释不生效
#127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
#127.0.0.1 ecs-hw-0001 ecs-hw-0001

192.168.0.176 ecs-hw-0001
192.168.0.25 ecs-hw-0002
192.168.0.248 ecs-hw-0003 添加内容
```

(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：

```
su - zhangsan
```

(3) 本地生成密钥：

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

说明：会让输入信息，保持默认，回车即可（三次回车）。

```
[zhangsan@ecs-hw-0001 ~]$ 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:
SHA256:0196aYtEdZD49yWtecaZccbtUXgfU377K8ZopYhU zhangsan@ecs-hw-0001
The key's randomart image is:
+---[RSA 4096]-----+
|      . oo.o=*|
|      o o  =o+*|
|      . . . +X*|
|      . E .o +|
|      oSo. . o |
|      .+... . |
|      .o. . . |
|      .+... =. |
|      +o. .*. |
+---[SHA256]-----+
[zhangsan@ecs-hw-0001 ~]$
```

(4) 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）：

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

验证免密登录

```
[zhangsan@ecs-hw-0001 ~]$ ssh-copy-id zhangsan@ecs-hw-0003
/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-0003 (192.168.0.248)' can't be established.
ECDSA key fingerprint is SHA256:1KRGCL1XAFODruTikRdI38HdbCeyrleCaZixGeo.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/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-0003's password:
Number of key(s) added: 1

Now try logging into the machine, with: "ssh 'zhangsan@ecs-hw-0003'"
and check to make sure that only the key(s) you wanted were added.

[zhangsan@ecs-hw-0001 ~]$ ssh 'zhangsan@ecs-hw-0003'

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

Welcome to Huawei Cloud Service

Last login: Tue Nov 29 15:11:57 2022

Welcome to 4.19.90-2110.8.0.0119.0el.aarch64

System information as of time: Tue Nov 29 15:21:16 CST 2022

System load: 0.00
Processes: 150
Memory used: 10.9%
Swap used: 0.8%
Usage On: 8%
IP address: 192.168.0.248
Users online: 2

[zhangsan@ecs-hw-0003 ~]$
```

(5) 安装依赖包：

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


(6) 源码编译安装 mpi (如果执行下面命令提示没有权限就执行 `sudo` 命令, 如 “`sudo wget ...`”):

```
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
```

三、 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)

```
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)

```
vim mpi_hello_world.c
```

代码如下:

```
#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();
}
```

```
#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();
}
```

2. 创建 makefile

执行以下命令，创建 makefile（三台主机都执行）

```
vim makefile
```

代码如下：（注意缩进）

```
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}
```

3. 进行编译

执行以下命令，进行编译（三台主机都执行）

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

```
[zhangsan@ecs-hw-0001 hello]$ ls
makefile  mpi_hello_world.c
[zhangsan@ecs-hw-0001 hello]$ make
mpicc -o mpi_hello_world mpi_hello_world.c
[zhangsan@ecs-hw-0001 hello]$ ls
makefile  mpi_hello_world  mpi_hello_world.c
[zhangsan@ecs-hw-0001 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）

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

添加内容如下：

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

```
[zhangsan@ecs-hw-0001 hello]$ ls  
config makefile mpi_hello_world mpi_hello_world.c  
[zhangsan@ecs-hw-0001 hello]$ cat config  
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）

```
mpiexec -n 6 -f /home/zhangsan/hello/config /home/zhangsan/hello/mpi_hello_world
```

结果如下：

```
[zhangsan@ecs-hw-0001 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-0001, rank 0 out of 6 processors  
Hello world from processor ecs-hw-0003, rank 4 out of 6 processors  
Hello world from processor ecs-hw-0001, rank 1 out of 6 processors  
Hello world from processor ecs-hw-0003, rank 5 out of 6 processors  
Hello world from processor ecs-hw-0002, rank 2 out of 6 processors  
Hello world from processor ecs-hw-0002, rank 3 out of 6 processors  
[zhangsan@ecs-hw-0001 hello]$  
[zhangsan@ecs-hw-0001 hello]$  
[zhangsan@ecs-hw-0001 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-0003, rank 4 out of 6 processors  
Hello world from processor ecs-hw-0003, rank 5 out of 6 processors  
Hello world from processor ecs-hw-0002, rank 2 out of 6 processors  
Hello world from processor ecs-hw-0002, rank 3 out of 6 processors  
Hello world from processor ecs-hw-0001, rank 0 out of 6 processors  
Hello world from processor ecs-hw-0001, rank 1 out of 6 processors  
[zhangsan@ecs-hw-0001 hello]$
```

通过上述代码运行，可以看出，编写的 hello-world 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。

特别注意：

配置好机器后，平时不用的话可以设置关机，这样能减少扣费。

四、 停止/关闭华为云资源（以删除弹性云服务器为例）

教学或实验完成之后/不使用华为云资源时，请**务必先停止相关服务并删除所有资源**，再退出华为云账号，避免系统认为资源占用继续扣费并导致欠费。

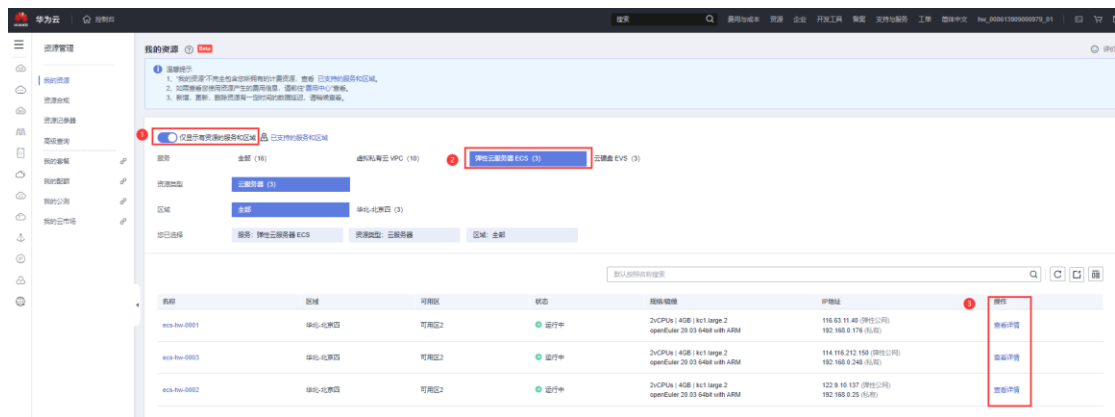
步骤 1

点击进入“控制台” “资源” “我的资源”。如下图所示：



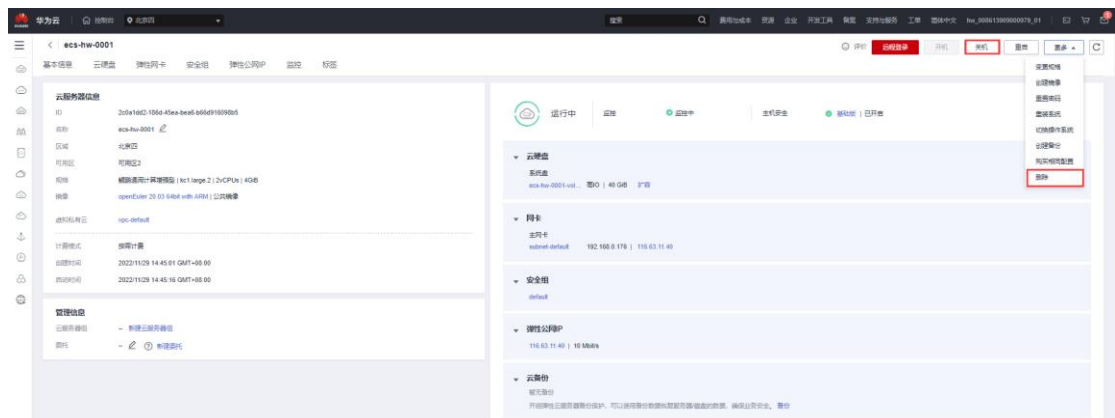
步骤 2

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）。



步骤 3

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：



步骤 4

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。