

华为云环境搭建

1.实验环境介绍

本实验需要一台终端电脑与弹性云服务器(ECS)链接以输入操作命令或/和传输文件，本实验默认终端机使用的操作系统为 Windows 10 系统，命令行工具 ssh 完成操作命令输入，用 scp 完成文件传输。

对于 Windows 系统，用组合键 “win+R” 唤出运行窗口，输入 cmd 即可打开命令行窗口；

对于 macOS 系统，点击 Launchpad，在搜索框中输入 “terminal” 或 “终端”，点击 “终端”，即可打开终端界面；

对 Linux 系统，以 ubuntu 为例，用组合键 “CTRL+ALT+T” 即可打开终端。

有些 Windows 系统下不能运行 ssh 工具，也可以使用 Putty 和 WinSCP 工具软件。其中 Putty 工具的推荐下载地址：

<https://hcia.obs.cn-north-4.myhuaweicloud.com/v1.5/putty.exe>

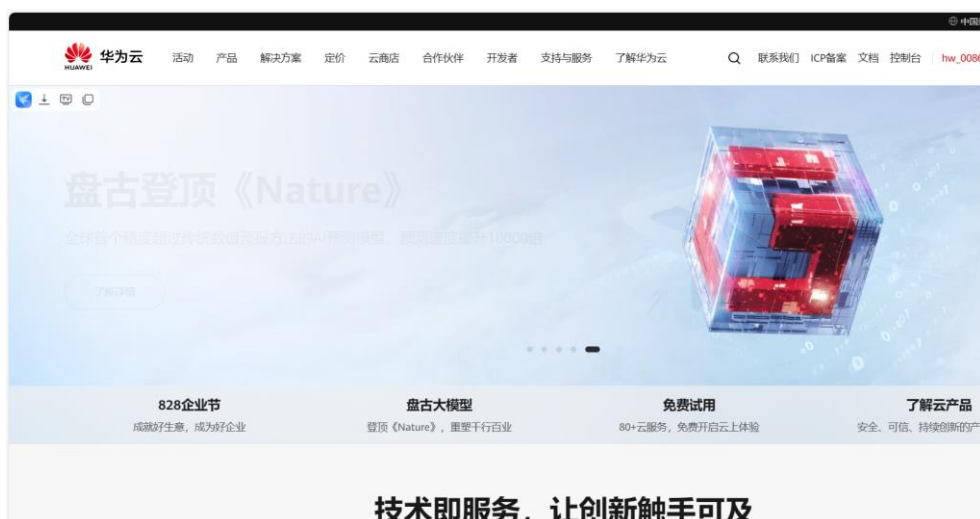
WinSCP 的推荐下载地址：

<https://winscp.net/eng/index.php>

实验指导书中的内容以 WinSCP 实现。

2.在云端布置服务器

(1) 在浏览器地址栏输入 www.huaweicloud.com，登录华为云官网。



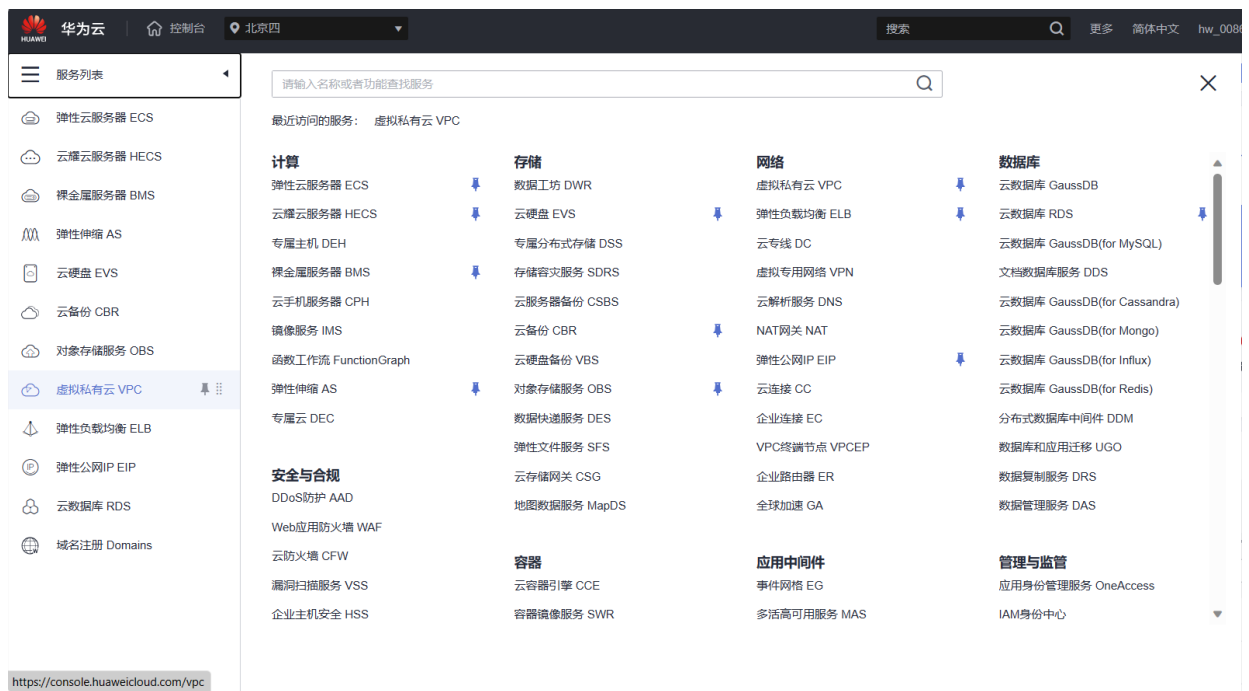
(2) 点击右上角的登录，打开登录窗口，按要求输入账号密码，登录华为云，随后点击右上角的“控制台”，打开控制台界面；



(3) 切换区域为“华北-北京四”；



(4) 点击控制台界面“导航栏”的“虚拟私有云 VPC”，进入 VPC 控制台；



(5) 点击右上角的“创建虚拟私有云”，并按照如下表配置 VPC 参数：

参数	配置
区域	华北-北京四
名称	vpc-test
网段	192.168.1.0/24
企业项目	default
默认子网可用区	可用区 1
默认子网名称	subnet-test
子网网段	如 192.168.1.0/24

配置完成后，点击“立即创建”，系统会自动回到 VPC 控制台；

(6) 点击 VPC 控制台左侧导航栏的“访问控制”→“安全组”，进入安全组控制台：



(7) 点击右上角的“创建安全组”：



(8) 在弹出的对话框中按如下图配置安全组参数，然后点击“确定”：

×

创建安全组

★ 模板

通用Web服务器 ▼

★ 名称

sg-test

描述

通用Web服务器，默认放通22、3389、80、443端口和ICMP协议。适用于需要远程登录、公网ping及用于网站服务的云服务器场景。

0/255

查看模板规则 ▼

确定

取消

(9) 点左侧导航栏的“弹性云服务器 ECS”，进入 ECS 控制台：



服务列表



弹性云服务器 ECS



弹性云服务器 ECS



弹性伸缩 AS



裸金属服务器 BMS



云硬盘



云硬盘备份



虚拟私有云 VPC



弹性负载均衡 ELB



域名注册



弹性公网IP

(10) 在 ECS 控制台界面点击右上角的“购买弹性云服务器”：



按照如下表配置弹性云服务器的基础配置参数：

参数	配置
计费模式	按需计费
区域	华北-北京四
可用区	随机分配
CPU 架构	鲲鹏计算
规格	鲲鹏通用计算增强型 kc1.large.2 2vCPUs 4GB
镜像	公共镜像 openEuler openEuler 20.03 64bit with ARM(40GB)
系统盘	通用型 SSD 40GB

配置完成后点击“下一步：网络配置”；

(11) 进入网络配置，按下表配置网络参数：

参数	配置
网络	vpc-test subnet-test 自动分配 IP 地址
安全组	sg-test
弹性公网 IP	现在购买
线路	全动态 BGP
公网带宽	按流量计费
带宽大小	5Mbit/s

配置完成后，点击“下一步：高级配置”；

(12) 按如下表配置 ECS 高级配置参数：

参数	配置
云服务器名称	kp-test01（输入符合规则名称）
登录凭证	密码
密码	请输入 8 位以上包含大小写字母、数字和特殊字符的密码
确认密码	请再次输入密码

云备份	暂不购买
云服务器组	不配置
高级选项	不勾选

配置完成后点击右下角“下一步：确认配置”；

(12) 勾选同意协议，然后点击：立即购买，在提交任务成功后，点击“返回云服务器列表”，返回 ECS 控制台。

至此我们所需使用的服务器就已经在云端分配好了，稍等片刻之后系统装载完毕，在 ESC 控制台可以查看服务器的相关信息：

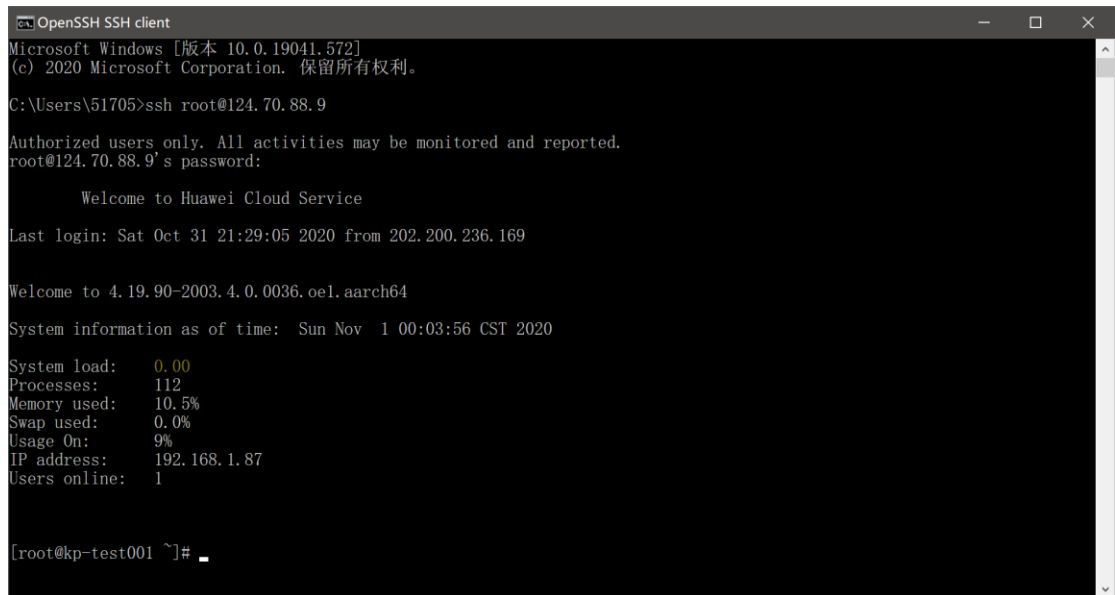
<input type="checkbox"/> 名称/ID	监控	可用区 ▾	状态 ▾	规格/镜像	IP地址	计费模式 ▾	标签	操作
<input type="checkbox"/> kp-test001 cf0a6430-eaec-4720-b51f-8ba374970175		可用区1	运行中	2xCPU 4GB kc1.large.2 openEuler 20.03 64bit with ARM	124.70.88.9 (弹性公网) 10... 192.168.1.87 (私有)	按量计费 2020/10/21 12:18:31 创建	--	远程登录 更多 ▾

3.远程登陆服务器



访问服务器首先要定位这台服务器，在华为云网站上的 ESC 控制台可以看到该服务器的 IP 地址，通过这里显示的 IP 地址就可以访问我们部署在云端的服务器。

打开控制台，键入命令 “ssh root@[公网 IP]” ，随后输入我们设置的密码，就成功登入服务器了；



```
OpenSSH SSH client
Microsoft Windows [版本 10.0.19041.572]
(c) 2020 Microsoft Corporation. 保留所有权利。

C:\Users\51705>ssh root@124.70.88.9

Authorized users only. All activities may be monitored and reported.
root@124.70.88.9's password:

Welcome to Huawei Cloud Service

Last login: Sat Oct 31 21:29:05 2020 from 202.200.236.169

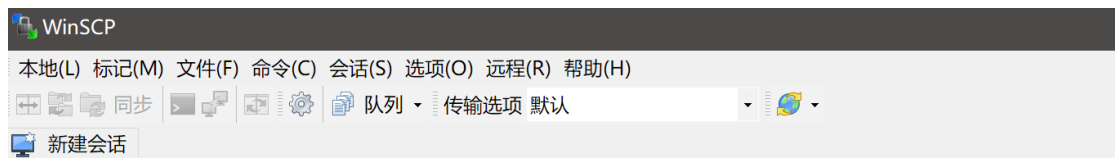
Welcome to 4.19.90-2003.4.0.0036.oel.aarch64

System information as of time: Sun Nov  1 00:03:56 CST 2020

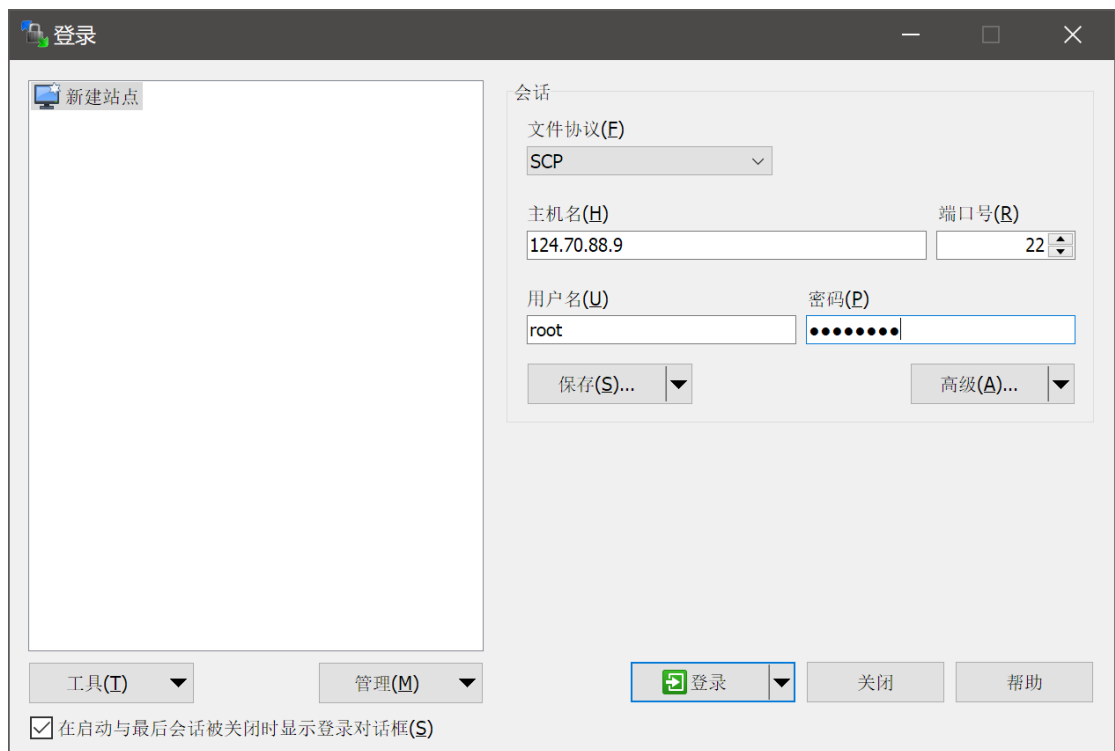
System load:   0.00
Processes:    112
Memory used:  10.5%
Swap used:    0.0%
Usage On:     9%
IP address:   192.168.1.87
Users online: 1

[root@kp-test001 ~]#
```

打开 WinSCP，点击 “新建会话”：



弹出以下界面，选择 SCP 协议，主机名为服务器的公网 IP，端口号默认为 22，用户名为 root，密码为创建服务器时设置的密码，点击登录：



登陆后我们就能以我们熟悉的 Windows 图形界面访问服务器的文件系统了：

名字	大小	已改变	权限	拥有者
..		2020/10/31 17:19:36	r-xr-xr-x	root
bin		2020/3/24 5:34:24	rw-rw-rw-	root
boot		2020/5/18 11:37:10	r-xr-xr-x	root
CloudResetPwdUpd...		2020/5/18 15:29:08	rw-x-----	root
CloudrResetPwdAge...		2020/5/18 15:29:12	rw-x-----	root
dev		2020/10/31 17:19:29	rw-xr-xr-x	root
etc		2020/10/31 17:19:42	rw-xr-xr-x	root
home		2020/3/24 5:34:24	rw-xr-xr-x	root
lib		2020/3/24 5:34:24	rw-rw-rw-	root
lib64		2020/3/24 5:34:24	rw-rw-rw-	root
lost+found		2020/5/18 10:32:25	rw-x-----	root
media		2020/3/24 5:34:24	rw-xr-xr-x	root
mnt		2020/3/24 5:34:24	rw-xr-xr-x	root
opt		2020/3/24 5:34:24	rw-xr-xr-x	root
proc		1970/1/1 8:00:00	r-xr-xr-x	root
root		2020/10/31 17:26:21	r-xr-x---	root
run		2020/10/31 17:19:42	rw-xr-xr-x	root
sbin		2020/3/24 5:34:24	rw-rw-rw-	root
srv		2020/3/24 5:34:24	rw-xr-xr-x	root
sys		2020/10/31 19:52:05	r-xr-xr-x	root
tmp		2020/10/31 23:32:22	rw-rw-rwt	root
usr		2020/5/18 10:32:59	rw-xr-xr-x	root
var		2020/5/18 11:26:29	rw-xr-xr-x	root

4.查看服务器信息

在 ssh 界面键入以下命令即可查看服务器的相关信息：

命令	显示内容
uname -a	查看总体架构
cat /etc/os-release	查看操作系统信息
lscpu	查看 CPU 信息

free	查看内存信息
fdisk -l	查看磁盘信息
top	查看系统资源实时信息（按 q 退出）
gcc -version	查看 gcc 版本

输出如下：

```
[root@kp-test001 test]# uname -a
Linux kp-test001 4.19.90-2003.4.0.0036.oel.aarch64 #1 SMP Mon Mar 23 19:06:43 UTC 2020 aarch64 aarch64 aarch64 GNU/Linux
[root@kp-test001 test]# cat /etc/os-release
NAME="openEuler"
VERSION="20.03 (LTS)"
ID="openEuler"
VERSION_ID="20.03"
PRETTY_NAME="openEuler 20.03 (LTS)"
ANSI_COLOR="0;31"

[root@kp-test001 test]# lscpu
Architecture:          aarch64
CPU op-mode(s):        64-bit
Byte Order:            Little Endian
CPU(s):                2
On-line CPU(s) list:   0,1
Thread(s) per core:    1
Core(s) per socket:    2
Socket(s):              1
NUMA node(s):          1
Vendor ID:             HiSilicon
Model:                 0
Model name:            Kunpeng-920
Stepping:              0x1
CPU max MHz:           2400.0000
CPU min MHz:           2400.0000
BogoMIPS:              200.00
L1d cache:             128 KiB
L1i cache:             128 KiB
L2 cache:              1 MiB
L3 cache:              32 MiB
NUMA node0 CPU(s):     0,1
vulnerability Itlb multihit: Not affected
vulnerability L1tf:     Not affected
vulnerability Mds:      Not affected
vulnerability Meltdown: Not affected
vulnerability Spec store bypass: Vulnerable
vulnerability Spectre v1: Mitigation: __user pointer sanitization
vulnerability Spectre v2: Not affected
vulnerability Tsx async abort: Not affected
Flags:                  fp asimd evtstrm aes pmull sha1 sha2 crc32 atomics fphp asimdhp cpuid asimdrdm jscvt fcma dcpop asimddp
                        asimdffdm
```

```
[root@kp-test001 test]# free
              total        used        free      shared  buff/cache   available
Mem:           3048896       328768       2141504        13760        578624       2380480
Swap:              0              0              0
[root@kp-test001 test]# fdisk -l
Disk /dev/vda: 40 GiB, 42949672960 bytes, 83886080 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: gpt
Disk identifier: E4391D6A-9819-45CA-BDE4-52285BCE52A7

Device            Start       End     Sectors  Size Type
/dev/vda1         2048     2099199    2097152    1G EFI System
/dev/vda2        2099200    83884031   81784832   39G Linux filesystem
[root@kp-test001 test]# top
top - 00:28:14 up 7:08, 1 user, load average: 0.00, 0.00, 0.00
Tasks: 115 total, 1 running, 114 sleeping, 0 stopped, 0 zombie
Cpu(s):  0.0 us,  0.0 sy,  0.0 ni,100.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem : 2977.4 total, 2088.3 free, 324.0 used, 565.1 buff/cache
MiB Swap:  0.0 total,  0.0 free,  0.0 used, 2321.8 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR   S  %CPU  %MEM     TIME+ COMMAND
    1 root        20   0   108608   16640    8832  S   0.0   0.5   0:01.98 systemd
    2 root        20   0      0      0      0  S   0.0   0.0   0:00.00 kthreadd
    3 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 rcu_gp
    4 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 rcu_par_gp
    6 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 kworker/0:0H-kblockd
    8 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 mm_percpu_wq
    9 root        20   0      0      0      0  S   0.0   0.0   0:00.01 ksoftirqd/0
   10 root        20   0      0      0      0  I   0.0   0.0   0:00.08 rcu_sched
   11 root        20   0      0      0      0  I   0.0   0.0   0:00.00 rcu_bh
   12 root        20   0      0      0      0  S   0.0   0.0   0:00.01 migration/0
   13 root        20   0      0      0      0  S   0.0   0.0   0:00.00 cpuhp/0
   14 root        20   0      0      0      0  S   0.0   0.0   0:00.00 cpuhp/1
   15 root        20   0      0      0      0  S   0.0   0.0   0:00.01 migration/1
   16 root        20   0      0      0      0  S   0.0   0.0   0:00.01 ksoftirqd/1
   18 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 kworker/1:0H-kblockd
   19 root        20   0      0      0      0  S   0.0   0.0   0:00.00 kdevtmpfs
   20 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 netns
   21 root        20   0      0      0      0  S   0.0   0.0   0:00.04 kauditd
   23 root        20   0      0      0      0  S   0.0   0.0   0:00.00 khungtaskd
   24 root        20   0      0      0      0  S   0.0   0.0   0:00.00 oom_reaper
   25 root        20  -20      0      0      0  I   0.0   0.0   0:00.00 writeback
   26 root        20   0      0      0      0  S   0.0   0.0   0:00.00 kcompactd0
   27 root        25   5      0      0      0  S   0.0   0.0   0:00.00 ksm
```

5.使用当前环境编译并运行一个程序

为了熟悉使用当前环境编译并运行程序的流程，我们是用最简单的 helloworld 程序，代码同学们已经很熟悉，这里不再赘述，具体的操作流程如下：

- (1) 在终端电脑上自行准备一个用于输出 “HelloWorld” 的.c 文件；
- (2) 使用 WinSCP 将该文件放置于服务器的/usr/local/src/test 目录下；
- (3) 使用 ssh 命令行，用 linux 的目录访问指令进入该目录；
- (4) 键入命令 “gcc -o hello HelloWorld.c” ；
- (5) 键入命令 “ls” 可以看到该目录下存在名为 hello 的可执行文件；
- (6) 键入命令 “./hello” 运行该文件，结果如下图所示：

```
[root@kp-test001 test]# ls
HelloWorld.c
[root@kp-test001 test]# gcc -o hello HelloWorld.c
[root@kp-test001 test]# ls
hello HelloWorld.c
[root@kp-test001 test]# ./hello
hello world!!!
```

今后的各类实验程序运行均可以参照该流程执行。