《大数据导论》实验报告

实验题目	环境搭建

一、实验目的

通过本次实验，完成以下部分：

- 组内合作完成 Hadoop & Spark 单机版环境搭建
- 组内合作完成 Hadoop & Spark 分布式环境搭建

最终需搭建相关详细环境如下：

- 操作系统： CentOS 7.6.64
- 图形界面： GNOME
- 语言环境： python 3.6.8
- 相关软件： Hadoop 2.8.5 、 Spark 2.4.4

二、实验项目内容

本次实验提供了三种方式来搭建 Hadoop & Sapak 分布式环境：

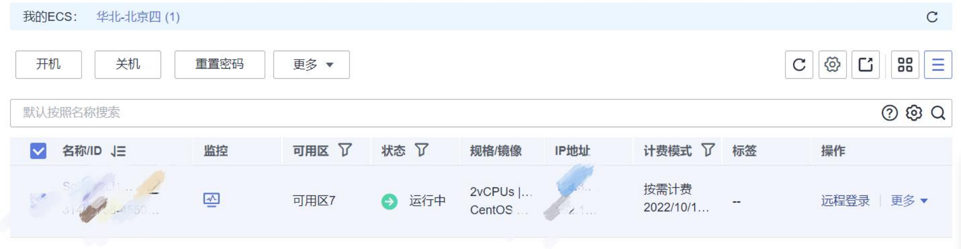
1. 云服务器分布式搭建
2. 伪分布式搭建
3. 多台机器分布式搭建

考虑到网络 IP 是动态分配（DHCP），没有使用固定 IP，使用多台实际机器搭建 Hadoop & Spark 分布式环境并不方便。为了充分学习云服务器环境搭建，体验大数据与服务器结合，我们组采用的是**云服务器分布式**搭建 Hadoop & Sapak 分布式环境。

三、实验过程或算法（源程序）

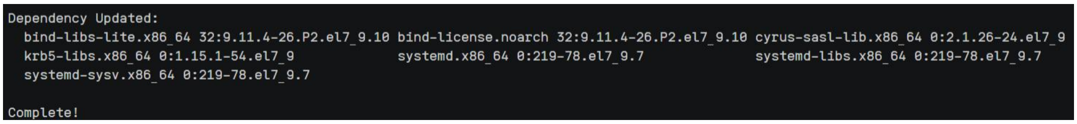
1. 实验准备

（1）购买华为云服务器



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
[redacted]		可用区7	运行中	2vCPUs |... CentOS ...	[redacted]	按需计费 2022/10/1...	--	远程登录 | 更多

（2）搭建 Linux 桌面环境（即 CentOS 7）



```
Dependency Updated:
  bind-libs-lite.x86_64 32:9.11.4-26.P2.el7_9.10 bind-license.noarch 32:9.11.4-26.P2.el7_9.10 cyrus-sasl-lib.x86_64 0:2.1.26-24.el7_9
  krb5-libs.x86_64 0:1.15.1-54.el7_9             systemd.x86_64 0:219-78.el7_9.7             systemd-libs.x86_64 0:219-78.el7_9.7
  systemd-sysv.x86_64 0:219-78.el7_9.7

Complete!
```

（3）安装 SSH 访问远端服务器

下载 XShell 一键安装即可，测试后不用输入密码，直接登陆成功，说明配置正确。

报告创建时间：2022.10.10

2. 云服务器分布式搭建

我们小组共搭建了 3 台云服务器进行 Hadoop+Spark，成员的内外网 IP 如下：

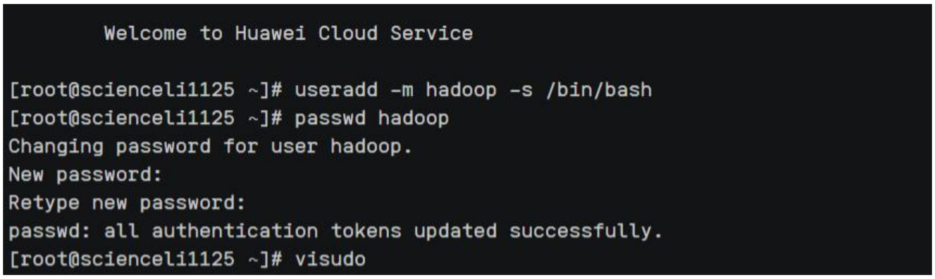
主机名	私有 IP	公有 IP
master	……	……
slave01	……	……
slave02	……	……

这只是刚配置时的 IP 地址，后面配置内网互联时需要更改私有 IP。

2.1 Spark 单机版搭建

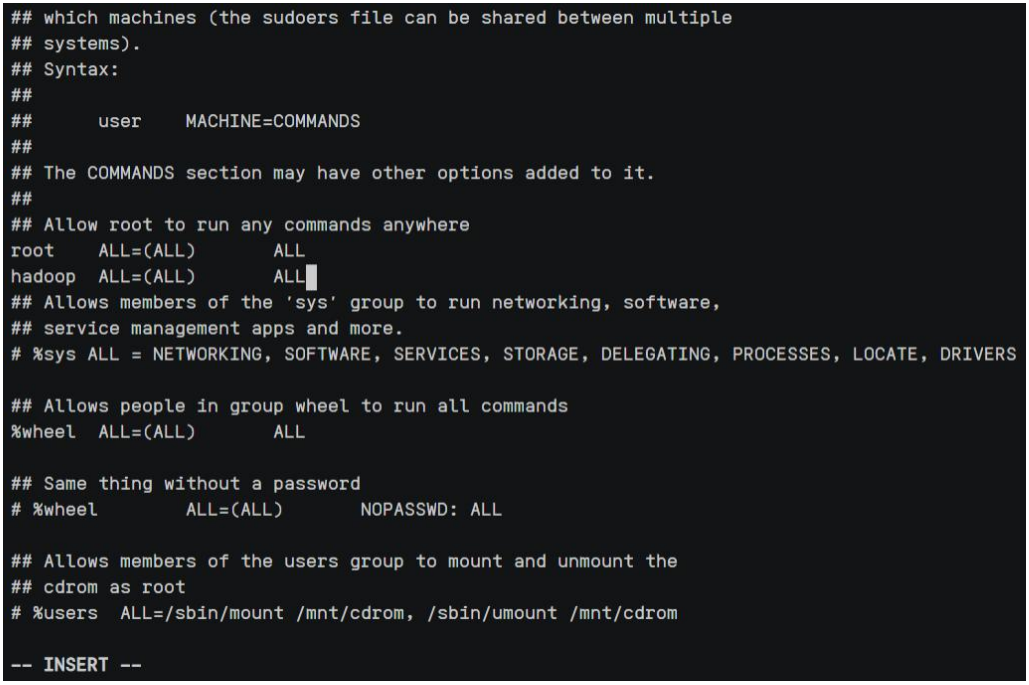
（1）准备工作

a. 创建 Hadoop 用户并设置密码：



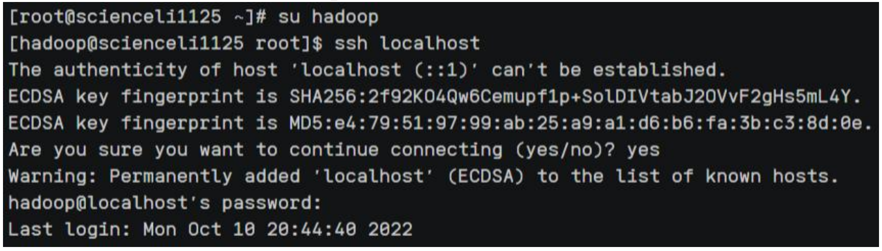
```
        Welcome to Huawei Cloud Service

[root@scienceli1125 ~]# useradd -m hadoop -s /bin/bash
[root@scienceli1125 ~]# passwd hadoop
Changing password for user hadoop.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@scienceli1125 ~]# visudo
```

进入配置文件，配置 hadoop 用户等同 root 权限：



```
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user    MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
hadoop  ALL=(ALL)       ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)       ALL

## Same thing without a password
# %wheel        ALL=(ALL)       NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

-- INSERT --
```

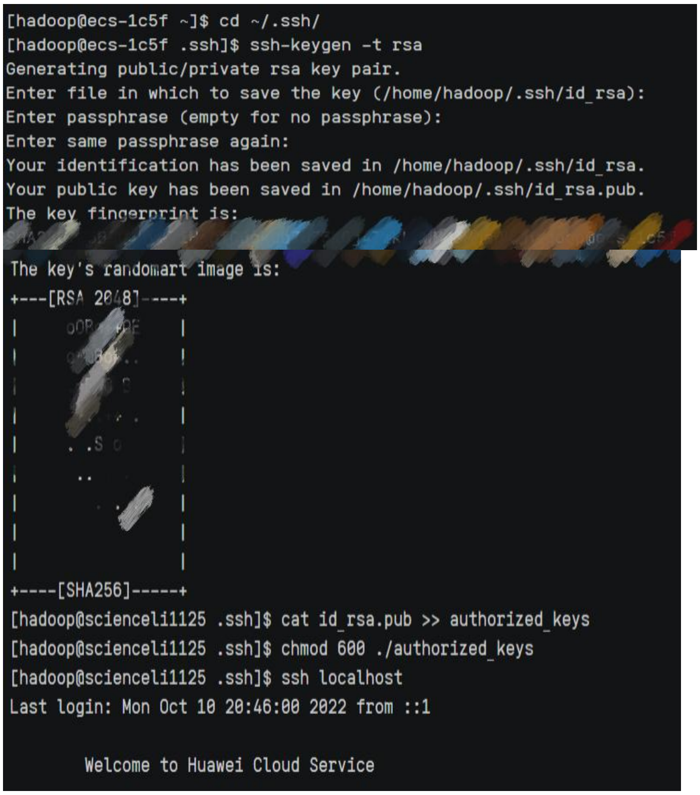
配置完成后，切换到 hadoop 用户下，再继续后续操作：



```
[root@scienceli1125 ~]# su hadoop
[hadoop@scienceli1125 root]$ ssh localhost
The authenticity of host 'localhost (::1)' can't be established.
ECDSA key fingerprint is SHA256:2f92KO4Qw6Cemupf1p+SolDIVtabJ2OVvF2gHs5mL4Y.
ECDSA key fingerprint is MD5:e4:79:51:97:99:ab:25:a9:a1:d6:b6:fa:3b:c3:8d:0e.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'localhost' (ECDSA) to the list of known hosts.
hadoop@localhost's password:
Last login: Mon Oct 10 20:44:40 2022
```

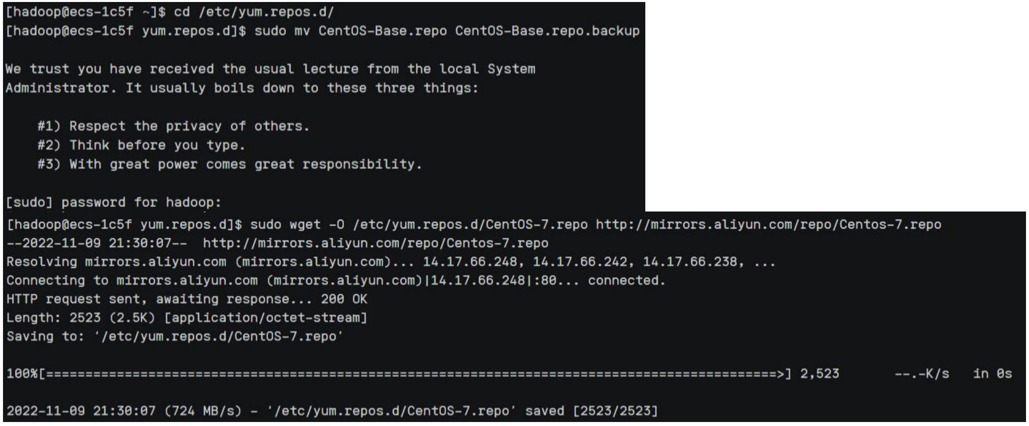
注意，配置完成后，以后每次进入服务器都需要使用命令 su hadoop **切换到 hadoop 用户下**。

b.配置 SSH，通过生成公钥配置实现免密码登陆：

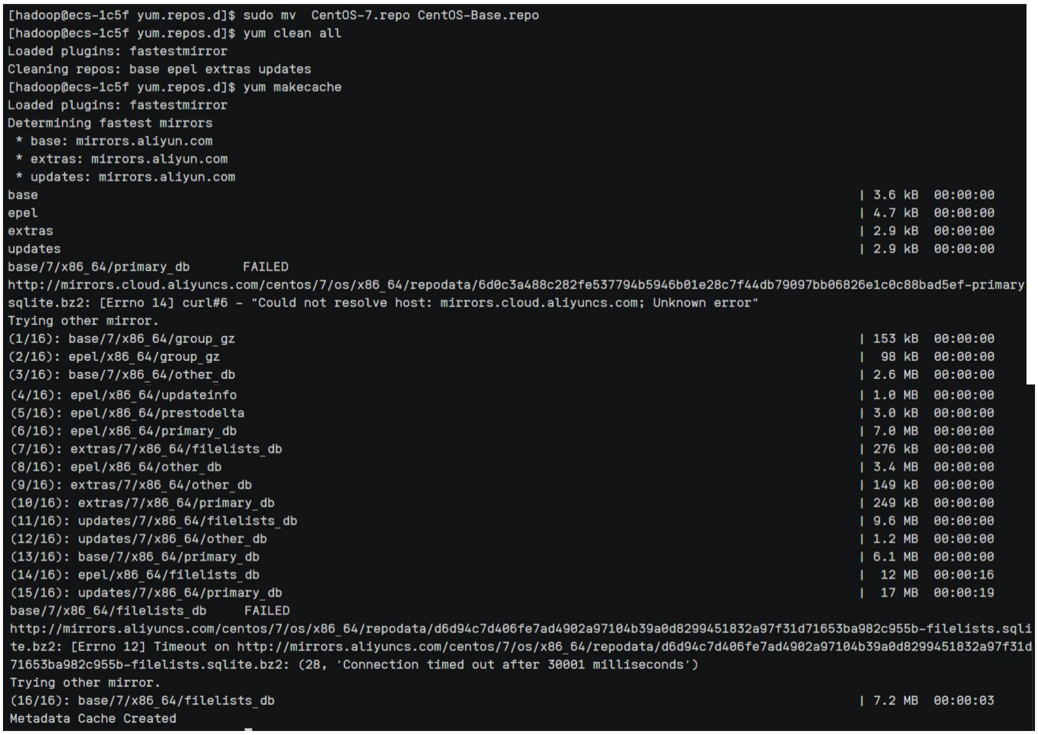


```
[hadoop@ecs-1c5f ~]$ cd ~/.ssh/
[hadoop@ecs-1c5f .ssh]$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/hadoop/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/hadoop/.ssh/id_rsa.
Your public key has been saved in /home/hadoop/.ssh/id_rsa.pub.
The key fingerprint is:
[redacted]
The key's randomart image is:
+---[RSA 2048]----+
[redacted]
|      . .S o     |
|       ..        |
[redacted]
|                 |
|                 |
+----[SHA256]-----+
[hadoop@scienceli1125 .ssh]$ cat id_rsa.pub >> authorized_keys
[hadoop@scienceli1125 .ssh]$ chmod 600 ./authorized_keys
[hadoop@scienceli1125 .ssh]$ ssh localhost
Last login: Mon Oct 10 20:46:00 2022 from ::1

        Welcome to Huawei Cloud Service
```

c.配置 yum 源：



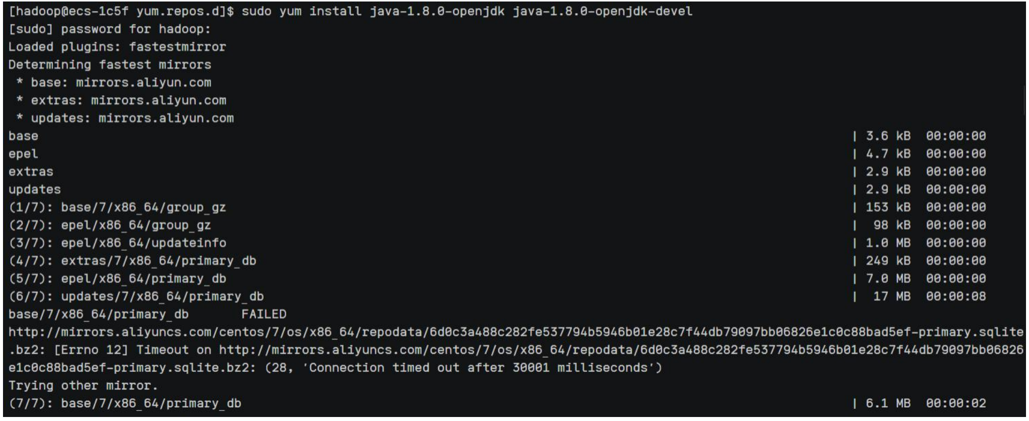
```
[hadoop@ecs-1c5f ~]$ cd /etc/yum.repos.d/
[hadoop@ecs-1c5f yum.repos.d]$ sudo mv CentOS-Base.repo CentOS-Base.repo.backup

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

[sudo] password for hadoop:
[hadoop@ecs-1c5f yum.repos.d]$ sudo wget -O /etc/yum.repos.d/CentOS-7.repo http://mirrors.aliyun.com/repo/Centos-7.repo
--2022-11-09 21:30:07--  http://mirrors.aliyun.com/repo/Centos-7.repo
Resolving mirrors.aliyun.com (mirrors.aliyun.com)... 14.17.66.248, 14.17.66.242, 14.17.66.238, ...
Connecting to mirrors.aliyun.com (mirrors.aliyun.com)|14.17.66.248|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2523 (2.5K) [application/octet-stream]
Saving to: '/etc/yum.repos.d/CentOS-7.repo'

100%[==========================================================================================>] 2,523       --.-K/s   in 0s

2022-11-09 21:30:07 (724 MB/s) - '/etc/yum.repos.d/CentOS-7.repo' saved [2523/2523]
```



```
[hadoop@ecs-1c5f yum.repos.d]$ sudo mv  CentOS-7.repo CentOS-Base.repo
[hadoop@ecs-1c5f yum.repos.d]$ yum clean all
Loaded plugins: fastestmirror
Cleaning repos: base epel extras updates
[hadoop@ecs-1c5f yum.repos.d]$ yum makecache
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
base                                                                                                 | 3.6 kB  00:00:00
epel                                                                                                 | 4.7 kB  00:00:00
extras                                                                                               | 2.9 kB  00:00:00
updates                                                                                              | 2.9 kB  00:00:00
base/7/x86_64/primary_db       FAILED
http://mirrors.cloud.aliyuncs.com/centos/7/os/x86_64/repodata/6d0c3a488c282fe537794b5946b01e28c7f44db79097bb06826e1c0c88bad5ef-primary
sqlite.bz2: [Errno 14] curl#6 - "Could not resolve host: mirrors.cloud.aliyuncs.com; Unknown error"
Trying other mirror.
(1/16): base/7/x86_64/group_gz                                                                       | 153 kB  00:00:00
(2/16): epel/x86_64/group_gz                                                                         |  98 kB  00:00:00
(3/16): base/7/x86_64/other_db                                                                       | 2.6 MB  00:00:00
(4/16): epel/x86_64/updateinfo                                                                       | 1.0 MB  00:00:00
(5/16): epel/x86_64/prestodelta                                                                      | 3.0 kB  00:00:00
(6/16): epel/x86_64/primary_db                                                                       | 7.0 MB  00:00:00
(7/16): extras/7/x86_64/filelists_db                                                                 | 276 kB  00:00:00
(8/16): epel/x86_64/other_db                                                                         | 3.4 MB  00:00:00
(9/16): extras/7/x86_64/other_db                                                                     | 149 kB  00:00:00
(10/16): extras/7/x86_64/primary_db                                                                  | 249 kB  00:00:00
(11/16): updates/7/x86_64/filelists_db                                                               | 9.6 MB  00:00:00
(12/16): updates/7/x86_64/other_db                                                                   | 1.2 MB  00:00:00
(13/16): base/7/x86_64/primary_db                                                                    | 6.1 MB  00:00:00
(14/16): epel/x86_64/filelists_db                                                                    |  12 MB  00:00:16
(15/16): updates/7/x86_64/primary_db                                                                 |  17 MB  00:00:19
base/7/x86_64/filelists_db     FAILED
http://mirrors.aliyuncs.com/centos/7/os/x86_64/repodata/d6d94c7d406fe7ad4902a97104b39a0d8299451832a97f31d71653ba982c955b-filelists.sqli
te.bz2: [Errno 12] Timeout on http://mirrors.aliyuncs.com/centos/7/os/x86_64/repodata/d6d94c7d406fe7ad4902a97104b39a0d8299451832a97f31d
71653ba982c955b-filelists.sqlite.bz2: (28, 'Connection timed out after 30001 milliseconds')
Trying other mirror.
(16/16): base/7/x86_64/filelists_db                                                                  | 7.2 MB  00:00:03
Metadata Cache Created
```

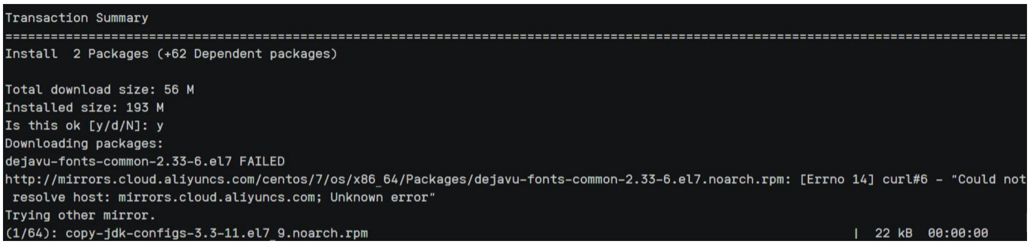
注意，**每台主机都需要配置 yum 源、Java 和 python 环境**，否则实验 2、3 进行分布式计算的时候主机无法使用从机计算资源，还是单机的效果。

d.配置 java 环境：



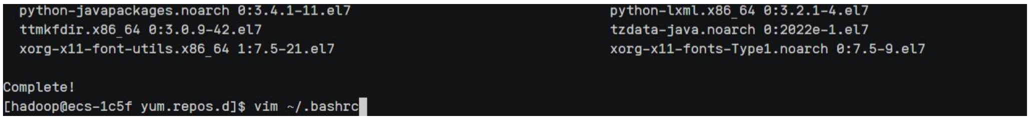
```
[hadoop@ecs-1c5f yum.repos.d]$ sudo yum install java-1.8.0-openjdk java-1.8.0-openjdk-devel
[sudo] password for hadoop:
Loaded plugins: fastestmirror
Determining fastest mirrors
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
base                                                                                                | 3.6 kB  00:00:00
epel                                                                                                | 4.7 kB  00:00:00
extras                                                                                              | 2.9 kB  00:00:00
updates                                                                                             | 2.9 kB  00:00:00
(1/7): base/7/x86_64/group_gz                                                                       | 153 kB  00:00:00
(2/7): epel/x86_64/group_gz                                                                         |  98 kB  00:00:00
(3/7): epel/x86_64/updateinfo                                                                       | 1.0 MB  00:00:00
(4/7): extras/7/x86_64/primary_db                                                                   | 249 kB  00:00:00
(5/7): epel/x86_64/primary_db                                                                       | 7.0 MB  00:00:00
(6/7): updates/7/x86_64/primary_db                                                                  |  17 MB  00:00:08
base/7/x86_64/primary_db       FAILED
http://mirrors.aliyuncs.com/centos/7/os/x86_64/repodata/6d0c3a488c282fe537794b5946b01e28c7f44db79097bb06826e1c0c88bad5ef-primary.sqlite
.bz2: [Errno 12] Timeout on http://mirrors.aliyuncs.com/centos/7/os/x86_64/repodata/6d0c3a488c282fe537794b5946b01e28c7f44db79097bb06826
e1c0c88bad5ef-primary.sqlite.bz2: (28, 'Connection timed out after 30001 milliseconds')
Trying other mirror.
(7/7): base/7/x86_64/primary_db                                                                     | 6.1 MB  00:00:02
```

中间略去一大堆安装文件...



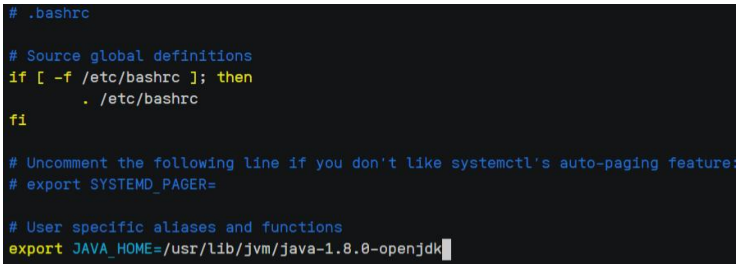
```
Transaction Summary
=======================================================================================================================================
Install  2 Packages (+62 Dependent packages)

Total download size: 56 M
Installed size: 193 M
Is this ok [y/d/N]: y
Downloading packages:
dejavu-fonts-common-2.33-6.el7 FAILED
http://mirrors.cloud.aliyuncs.com/centos/7/os/x86_64/Packages/dejavu-fonts-common-2.33-6.el7.noarch.rpm: [Errno 14] curl#6 - "Could not
 resolve host: mirrors.cloud.aliyuncs.com; Unknown error"
Trying other mirror.
(1/64): copy-jdk-configs-3.3-11.el7_9.noarch.rpm                                                    |  22 kB  00:00:00
```

中间略去一大堆安装文件...



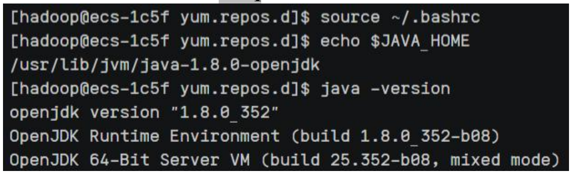
```
  python-javapackages.noarch 0:3.4.1-11.el7                                         python-lxml.x86_64 0:3.2.1-4.el7
  ttmkfdir.x86_64 0:3.0.9-42.el7                                                    tzdata-java.noarch 0:2022e-1.el7
  xorg-x11-font-utils.x86_64 1:7.5-21.el7                                           xorg-x11-fonts-Type1.noarch 0:7.5-9.el7

Complete!
[hadoop@ecs-1c5f yum.repos.d]$ vim ~/.bashrc
```

配置环境变量：



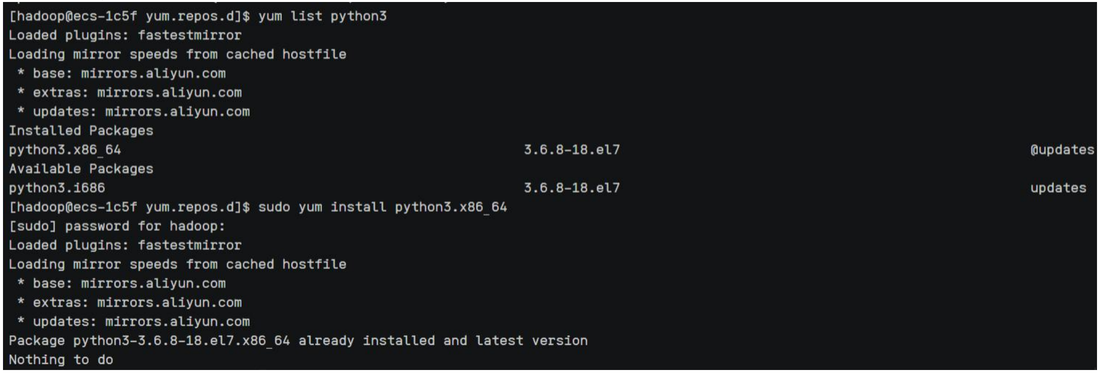
```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

注意，编辑配置文件时千万**不能 Ctrl+S**，否则页面会被锁定！按 Ctrl+Q 解锁。然后 Esc 退出编辑，“:wq”保存并退出。



```
[hadoop@ecs-1c5f yum.repos.d]$ source ~/.bashrc
[hadoop@ecs-1c5f yum.repos.d]$ echo $JAVA_HOME
/usr/lib/jvm/java-1.8.0-openjdk
[hadoop@ecs-1c5f yum.repos.d]$ java -version
openjdk version "1.8.0_352"
OpenJDK Runtime Environment (build 1.8.0_352-b08)
OpenJDK 64-Bit Server VM (build 25.352-b08, mixed mode)
```

e.安装 python：



```
[hadoop@ecs-1c5f yum.repos.d]$ yum list python3
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
Installed Packages
python3.x86_64                                   3.6.8-18.el7                                              @updates
Available Packages
python3.i686                                     3.6.8-18.el7                                              updates
[hadoop@ecs-1c5f yum.repos.d]$ sudo yum install python3.x86_64
[sudo] password for hadoop:
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: mirrors.aliyun.com
 * extras: mirrors.aliyun.com
 * updates: mirrors.aliyun.com
Package python3-3.6.8-18.el7.x86_64 already installed and latest version
Nothing to do
```

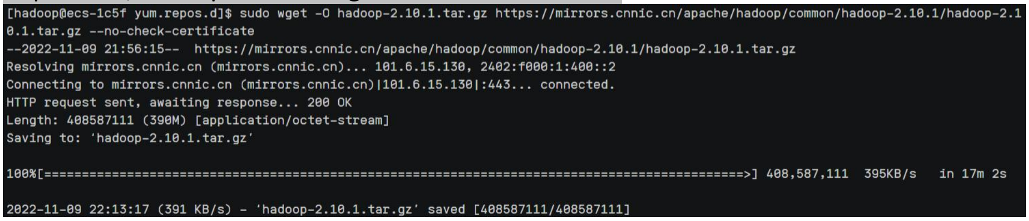
已经为最新版，无需更改。

（2）安装 hadoop

此处需要注意，hadoop 镜像网址更换，本次实验选用了新版的 Hadoop 进行安装，命令中**版本号需要更新**。

a.下载

sudo wget -O hadoop-2.10.1.tar.gz https://mirrors.cnnic.cn/apache/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz --no-check-certificate



```
[hadoop@ecs-1c5f yum.repos.d]$ sudo wget -O hadoop-2.10.1.tar.gz https://mirrors.cnnic.cn/apache/hadoop/common/hadoop-2.10.1/hadoop-2.1
0.1.tar.gz --no-check-certificate
--2022-11-09 21:56:15--  https://mirrors.cnnic.cn/apache/hadoop/common/hadoop-2.10.1/hadoop-2.10.1.tar.gz
Resolving mirrors.cnnic.cn (mirrors.cnnic.cn)... 101.6.15.130, 2402:f000:1:400::2
Connecting to mirrors.cnnic.cn (mirrors.cnnic.cn)|101.6.15.130|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 408587111 (390M) [application/octet-stream]
Saving to: 'hadoop-2.10.1.tar.gz'

100%[=============================================================================================>] 408,587,111  395KB/s   in 17m 2s

2022-11-09 22:13:17 (391 KB/s) - 'hadoop-2.10.1.tar.gz' saved [408587111/408587111]
```

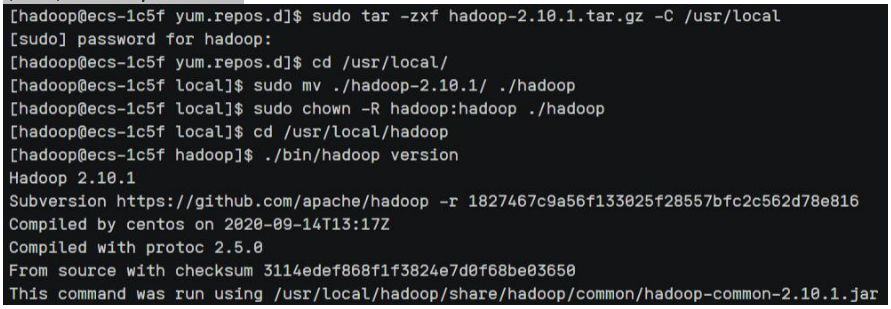
b.解压：

sudo tar -zxf hadoop-2.10.1.tar.gz -C /usr/local c.修改文件：

```
cd /usr/local/
sudo mv ./hadoop-2.10.1/ ./hadoop
sudo chown -R hadoop:hadoop ./hadoop
```

d. 测试：

```
cd /usr/local/hadoop
./bin/hadoop version
```



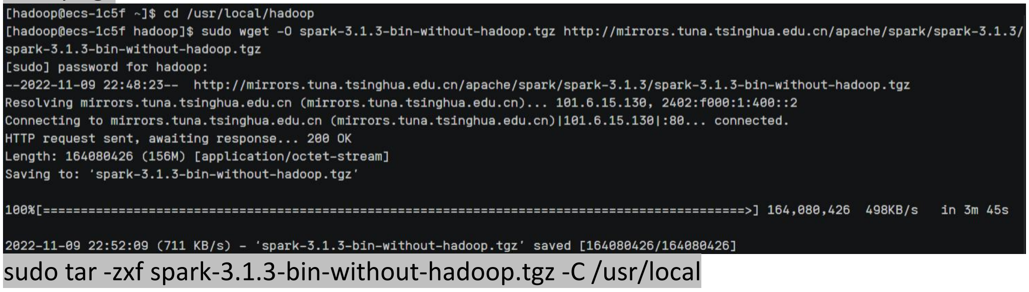
```
[hadoop@ecs-1c5f yum.repos.d]$ sudo tar -zxf hadoop-2.10.1.tar.gz -C /usr/local
[sudo] password for hadoop:
[hadoop@ecs-1c5f yum.repos.d]$ cd /usr/local/
[hadoop@ecs-1c5f local]$ sudo mv ./hadoop-2.10.1/ ./hadoop
[hadoop@ecs-1c5f local]$ sudo chown -R hadoop:hadoop ./hadoop
[hadoop@ecs-1c5f local]$ cd /usr/local/hadoop
[hadoop@ecs-1c5f hadoop]$ ./bin/hadoop version
Hadoop 2.10.1
Subversion https://github.com/apache/hadoop -r 1827467c9a56f133025f28557bfc2c562d78e816
Compiled by centos on 2020-09-14T13:17Z
Compiled with protoc 2.5.0
From source with checksum 3114edef868f1f3824e7d0f68be03650
This command was run using /usr/local/hadoop/share/hadoop/common/hadoop-common-2.10.1.jar
```

注意，指导书上说只需要在 master 服务器上安装 hadoop，但如果 slave 服务器不安装 hadoop，后续 spark 等也无法正常安装，因此需要在 3 台服务器上**各自**安装 hadoop。

（3）安装 spark

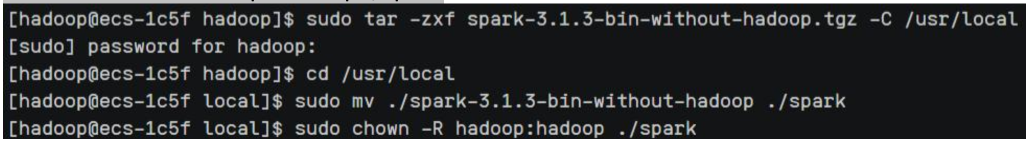
a.下载、解压、设置权限等操作同上，注意**版本号更新**：

```
sudo wget -O spark-3.1.3-bin-without-hadoop.tgz
http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz
```



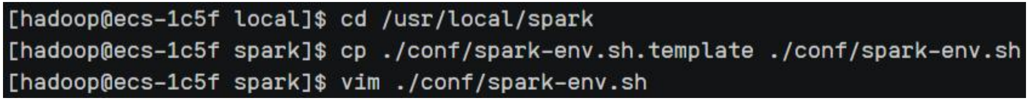
```
[hadoop@ecs-1c5f ~]$ cd /usr/local/hadoop
[hadoop@ecs-1c5f hadoop]$ sudo wget -O spark-3.1.3-bin-without-hadoop.tgz http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.1.3/
spark-3.1.3-bin-without-hadoop.tgz
[sudo] password for hadoop:
--2022-11-09 22:48:23--  http://mirrors.tuna.tsinghua.edu.cn/apache/spark/spark-3.1.3/spark-3.1.3-bin-without-hadoop.tgz
Resolving mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)... 101.6.15.130, 2402:f000:1:400::2
Connecting to mirrors.tuna.tsinghua.edu.cn (mirrors.tuna.tsinghua.edu.cn)|101.6.15.130|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 164080426 (156M) [application/octet-stream]
Saving to: ‘spark-3.1.3-bin-without-hadoop.tgz’

100%[==========================================================================================>] 164,080,426  498KB/s   in 3m 45s

2022-11-09 22:52:09 (711 KB/s) - ‘spark-3.1.3-bin-without-hadoop.tgz’ saved [164080426/164080426]
```

```
sudo tar -zxf spark-3.1.3-bin-without-hadoop.tgz -C /usr/local

cd /usr/local
sudo mv ./spark-3.1.3-bin-without-hadoop ./spark
sudo chown -R hadoop:hadoop ./spark
```



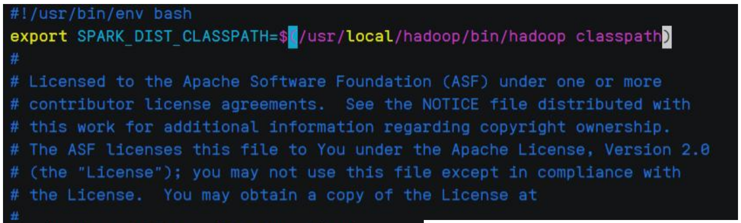
```
[hadoop@ecs-1c5f hadoop]$ sudo tar -zxf spark-3.1.3-bin-without-hadoop.tgz -C /usr/local
[sudo] password for hadoop:
[hadoop@ecs-1c5f hadoop]$ cd /usr/local
[hadoop@ecs-1c5f local]$ sudo mv ./spark-3.1.3-bin-without-hadoop ./spark
[hadoop@ecs-1c5f local]$ sudo chown -R hadoop:hadoop ./spark
```

b.配置 spark 环境和环境变量：

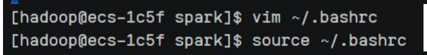


```
[hadoop@ecs-1c5f local]$ cd /usr/local/spark
[hadoop@ecs-1c5f spark]$ cp ./conf/spark-env.sh.template ./conf/spark-env.sh
[hadoop@ecs-1c5f spark]$ vim ./conf/spark-env.sh
```

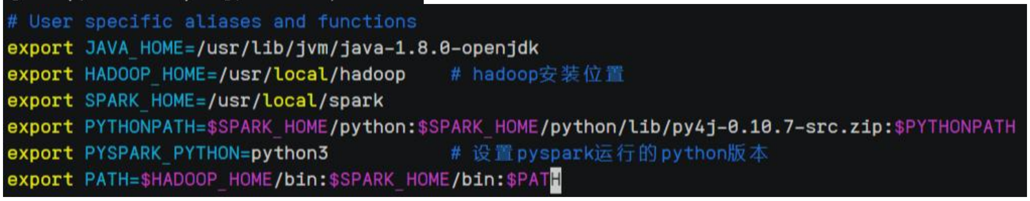
编辑环境变量：



```
#!/usr/bin/env bash
export SPARK_DIST_CLASSPATH=$(/usr/local/hadoop/bin/hadoop classpath)
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
#
```

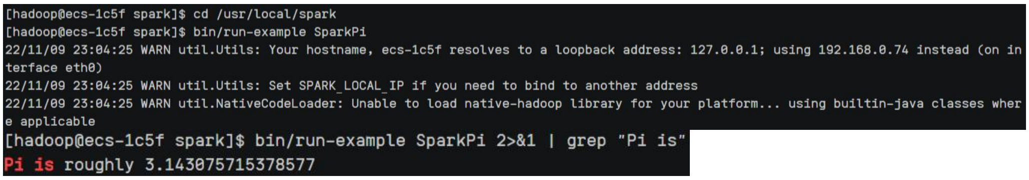


```
[hadoop@ecs-1c5f spark]$ vim ~/.bashrc
[hadoop@ecs-1c5f spark]$ source ~/.bashrc
```

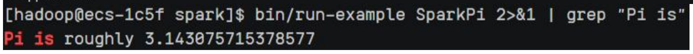


```
# User specific aliases and functions
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
export HADOOP_HOME=/usr/local/hadoop    # hadoop安装位置
export SPARK_HOME=/usr/local/spark
export PYTHONPATH=$SPARK_HOME/python:$SPARK_HOME/python/lib/py4j-0.10.7-src.zip:$PYTHONPATH
export PYSPARK_PYTHON=python3           # 设置pyspark运行的python版本
export PATH=$HADOOP_HOME/bin:$SPARK_HOME/bin:$PATH
```

c.测试运行成功



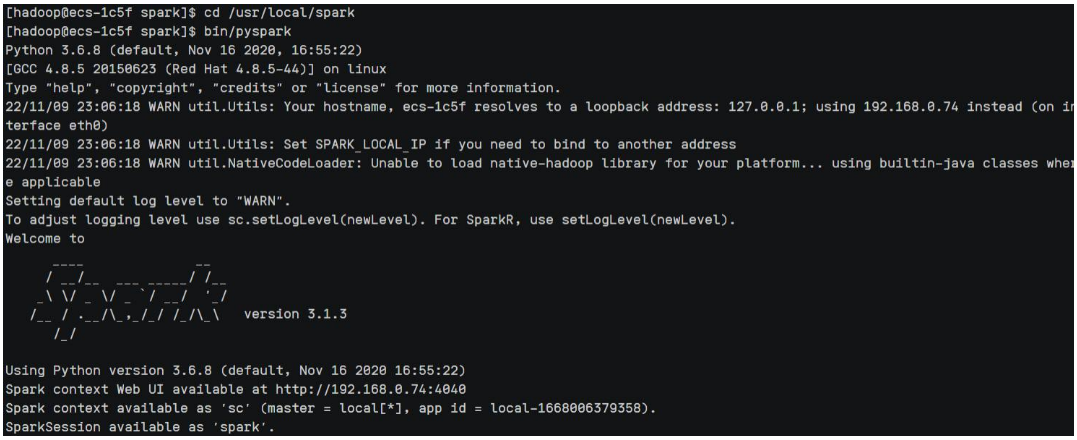
```
[hadoop@ecs-1c5f spark]$ cd /usr/local/spark
[hadoop@ecs-1c5f spark]$ bin/run-example SparkPi
22/11/09 23:04:25 WARN util.Utils: Your hostname, ecs-1c5f resolves to a loopback address: 127.0.0.1; using 192.168.0.74 instead (on in
terface eth0)
22/11/09 23:04:25 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/11/09 23:04:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes wher
e applicable
```



```
[hadoop@ecs-1c5f spark]$ bin/run-example SparkPi 2>&1 | grep "Pi is"
Pi is roughly 3.143075715378577
```

（4）测试单机版 spark 环境搭建效果

启动 pyspark 并做简单测试：



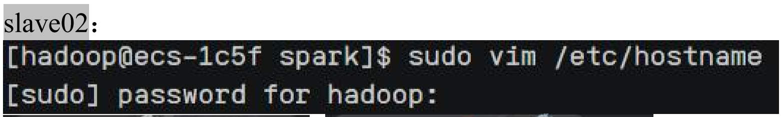
```
[hadoop@ecs-1c5f spark]$ cd /usr/local/spark
[hadoop@ecs-1c5f spark]$ bin/pyspark
Python 3.6.8 (default, Nov 16 2020, 16:55:22)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-44)] on linux
Type "help", "copyright", "credits" or "license" for more information.
22/11/09 23:06:18 WARN util.Utils: Your hostname, ecs-1c5f resolves to a loopback address: 127.0.0.1; using 192.168.0.74 instead (on in
terface eth0)
22/11/09 23:06:18 WARN util.Utils: Set SPARK_LOCAL_IP if you need to bind to another address
22/11/09 23:06:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes wher
e applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 3.1.3
      /_/

Using Python version 3.6.8 (default, Nov 16 2020 16:55:22)
Spark context Web UI available at http://192.168.0.74:4040
Spark context available as 'sc' (master = local[*], app id = local-1668006379358).
SparkSession available as 'spark'.
```

测试成功，单机版 spark 环境搭建成功，可做简单测试，输入“exit()”退出。

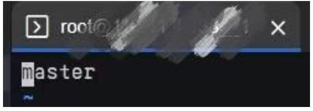
2.2 Hadoop+Spark 分布式环境搭建

（1）准备工作

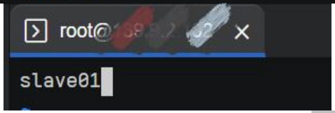
a.使用 sudo 指令进入服务器名编辑文件，修改三台服务器的主机名为 master，slave01，slave02：



```
[hadoop@ecs-1c5f spark]$ sudo vim /etc/hostname
[sudo] password for hadoop:
```



```
master
~
```



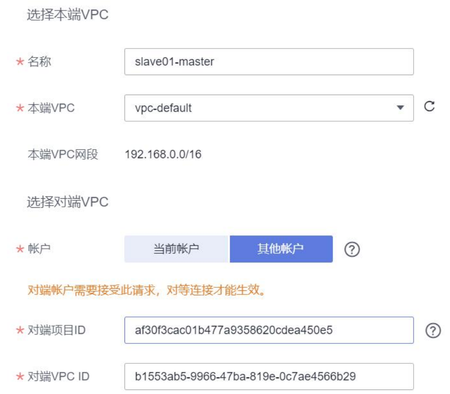
```
slave01
```

上图所示为主机和一个从机修改后的主机名，输入 sudo reboot 重连后可以看到用户

名已更改。

b.修改 hosts 实现云服务器名字访问而不必直接使用 IP 地址。原本配置的是外网访问，实验 3 互连时出现了问题，因此将此处改成**内网互联**。详情可参考群内王星然同学提供的《华为云配置对等连接实现内网互联》。（注意，此步骤非常重要，否则实验 3 进行分布式计算时会出现各种奇怪的 bug，以及内存爆满，节点停工等问题，扩大内存也无济于事）

建立对等连接：

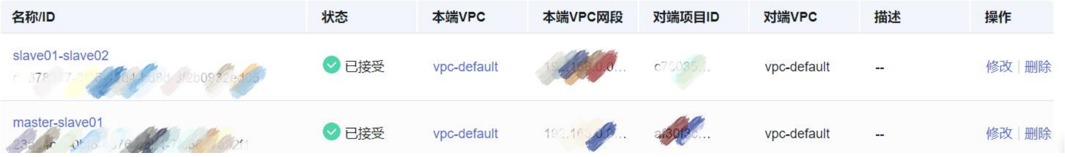


选择本端VPC

* 名称	slave01-master
* 本端VPC	vpc-default ▾
本端VPC网段	192.168.0.0/16

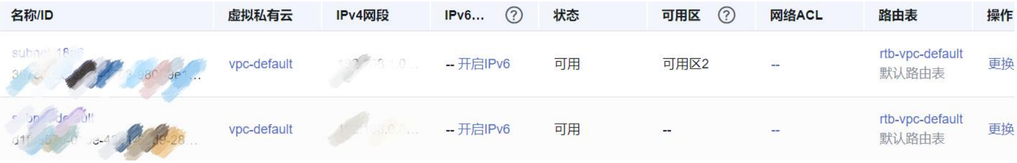
选择对端VPC

* 帐户	当前帐户 / 其他帐户 ⓘ
	对端帐户需要接受此请求，对等连接才能生效。
* 对端项目ID	af30f3cac01b477a9358620cdea450e5
* 对端VPC ID	b1553ab5-9966-47ba-819e-0c7ae4566b29



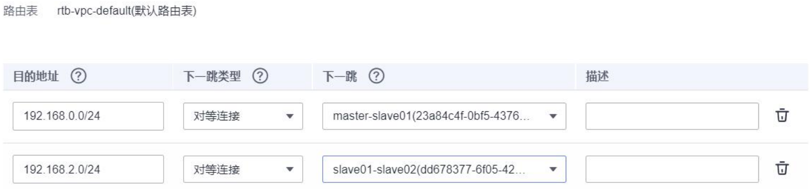
名称/ID	状态	本端VPC	本端VPC网段	对端项目ID	对端VPC	描述	操作
slave01-slave02 [redacted]	✓ 已接受	vpc-default	[redacted]	[redacted]	vpc-default	--	修改 | 删除
master-slave01 [redacted]	✓ 已接受	vpc-default	[redacted]	[redacted]	vpc-default	--	修改 | 删除

配置子网网段：



名称/ID	虚拟私有云	IPv4网段	IPv6... ⓘ	状态	可用区 ⓘ	网络ACL	路由表	操作
[redacted]	vpc-default	[redacted]	-- 开启IPv6	可用	可用区2	--	rtb-vpc-default 默认路由表	更换
[redacted]	vpc-default	[redacted]	-- 开启IPv6	可用	--	--	rtb-vpc-default 默认路由表	更换

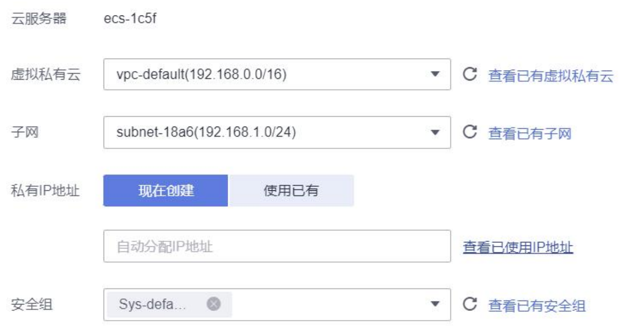
配置路由表：



路由表 rtb-vpc-default(默认路由表)

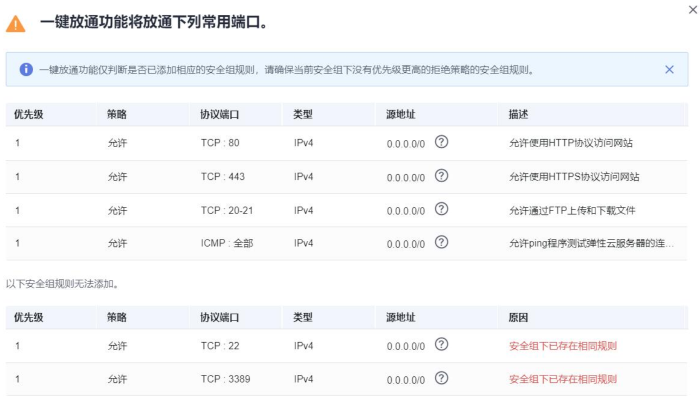
目的地址 ⓘ	下一跳类型 ⓘ	下一跳 ⓘ	描述
192.168.0.0/24	对等连接 ▾	master-slave01(23a84c4f-0bf5-4376... ▾	
192.168.2.0/24	对等连接 ▾	slave01-slave02(dd678377-6f05-42... ▾	

切换 VPC（不然无法远程连接）：



云服务器	ecs-1c5f	
虚拟私有云	vpc-default(192.168.0.0/16)	查看已有虚拟私有云
子网	subnet-18a6(192.168.1.0/24)	查看已有子网
私有IP地址	现在创建 使用已有	
	自动分配IP地址	查看已使用IP地址
安全组	Sys-defa...	查看已有安全组

此时还不能互 ping，因为入端口没有放通。一键放通入常用端口后重启服务器：



一键放通功能将放通下列常用端口。

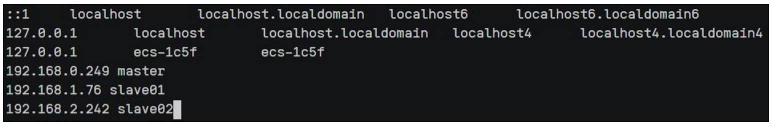
一键放通功能仅判断是否已添加相应的安全组规则，请确保当前安全组下没有优先级更高的拒绝策略的安全组规则。

优先级	策略	协议端口	类型	源地址	描述
1	允许	TCP : 80	IPv4	0.0.0.0/0	允许使用HTTP协议访问网站
1	允许	TCP : 443	IPv4	0.0.0.0/0	允许使用HTTPS协议访问网站
1	允许	TCP : 20-21	IPv4	0.0.0.0/0	允许通过FTP上传和下载文件
1	允许	ICMP : 全部	IPv4	0.0.0.0/0	允许ping程序测试弹性云服务器的连...

以下安全组规则无法添加。

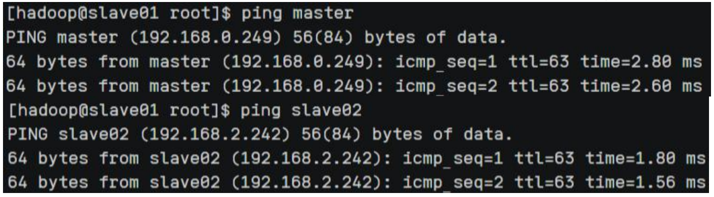
优先级	策略	协议端口	类型	源地址	原因
1	允许	TCP : 22	IPv4	0.0.0.0/0	安全组下已存在相同规则
1	允许	TCP : 3389	IPv4	0.0.0.0/0	安全组下已存在相同规则

最后再修改 hosts：



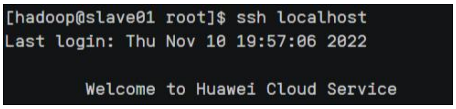
```
::1     localhost       localhost.localdomain   localhost6      localhost6.localdomain6
127.0.0.1       localhost       localhost.localdomain   localhost4      localhost4.localdomain4
127.0.0.1       ecs-1c5f        ecs-1c5f
192.168.0.249 master
192.168.1.76 slave01
192.168.2.242 slave02
```

到此为止，3 台主机之间可以使用主机名相互 ping 通：



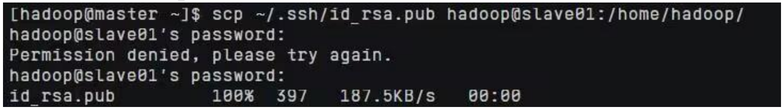
```
[hadoop@slave01 root]$ ping master
PING master (192.168.0.249) 56(84) bytes of data.
64 bytes from master (192.168.0.249): icmp_seq=1 ttl=63 time=2.80 ms
64 bytes from master (192.168.0.249): icmp_seq=2 ttl=63 time=2.60 ms
[hadoop@slave01 root]$ ping slave02
PING slave02 (192.168.2.242) 56(84) bytes of data.
64 bytes from slave02 (192.168.2.242): icmp_seq=1 ttl=63 time=1.80 ms
64 bytes from slave02 (192.168.2.242): icmp_seq=2 ttl=63 time=1.56 ms
```

c.3 台服务器间 SSH 互免验证



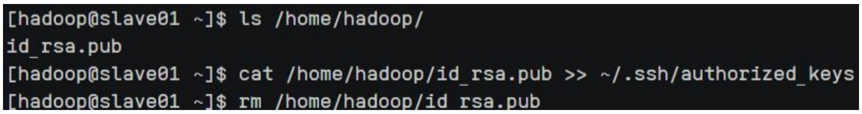
```
[hadoop@slave01 root]$ ssh localhost
Last login: Thu Nov 10 19:57:06 2022

        Welcome to Huawei Cloud Service
```

在 master 上 scp 传递公钥：



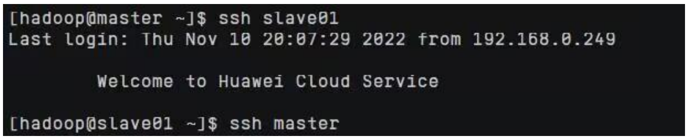
```
[hadoop@master ~]$ scp ~/.ssh/id_rsa.pub hadoop@slave01:/home/hadoop/
hadoop@slave01's password:
Permission denied, please try again.
hadoop@slave01's password:
id_rsa.pub      100%  397    187.5KB/s   00:00
```

在 slave 加点服务器上加入验证并将 master 公钥加入免验证



```
[hadoop@slave01 ~]$ ls /home/hadoop/
id_rsa.pub
[hadoop@slave01 ~]$ cat /home/hadoop/id_rsa.pub >> ~/.ssh/authorized_keys
[hadoop@slave01 ~]$ rm /home/hadoop/id_rsa.pub
```

此时在 master 主机上可以免密登录 slave01：



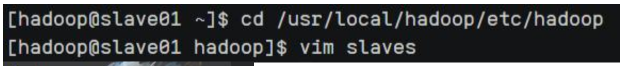
```
[hadoop@master ~]$ ssh slave01
Last login: Thu Nov 10 20:07:29 2022 from 192.168.0.249

        Welcome to Huawei Cloud Service

[hadoop@slave01 ~]$ ssh master
```

测试能够免密登陆！slave02 操作同上。

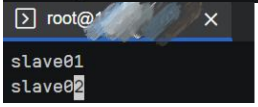
（2）Hadoop 集群配置

a. **master** 节点配置

切换目录修改 slaves 文件：

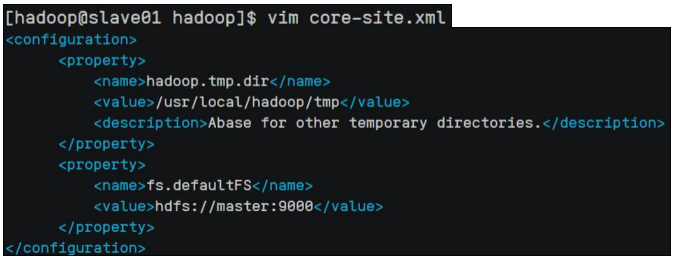


```
[hadoop@slave01 ~]$ cd /usr/local/hadoop/etc/hadoop
[hadoop@slave01 hadoop]$ vim slaves
```



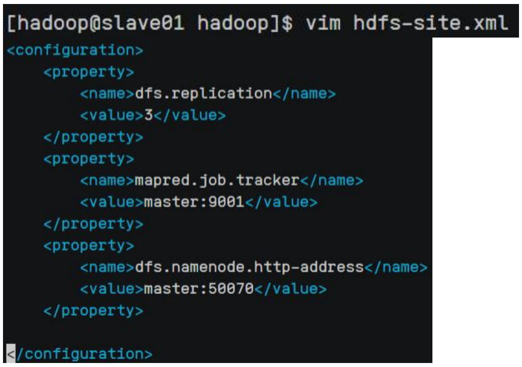
```
slave01
slave02
```

修改文件 core-site.xml：



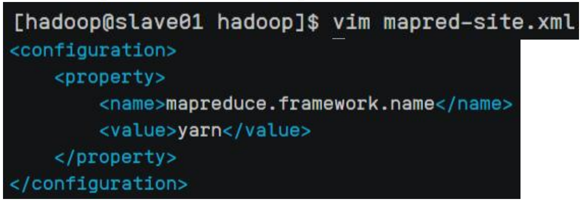
```
[hadoop@slave01 hadoop]$ vim core-site.xml
<configuration>
      <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/hadoop/tmp</value>
          <description>Abase for other temporary directories.</description>
      </property>
      <property>
          <name>fs.defaultFS</name>
          <value>hdfs://master:9000</value>
      </property>
</configuration>
```

修改 hdfs-site.xml：



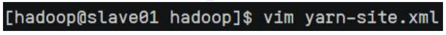
```
[hadoop@slave01 hadoop]$ vim hdfs-site.xml
<configuration>
    <property>
        <name>dfs.replication</name>
        <value>3</value>
    </property>
    <property>
        <name>mapred.job.tracker</name>
        <value>master:9001</value>
    </property>
    <property>
        <name>dfs.namenode.http-address</name>
        <value>master:50070</value>
    </property>

</configuration>
```

修改 mapred-site.xml.template

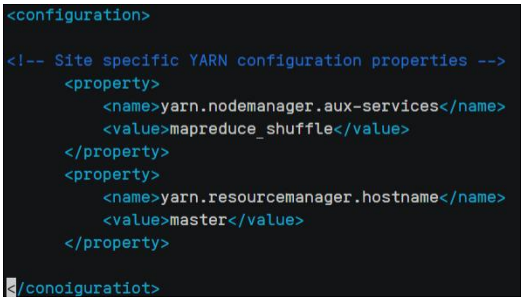


```
[hadoop@slave01 hadoop]$ vim mapred-site.xml
<configuration>
    <property>
        <name>mapreduce.framework.name</name>
        <value>yarn</value>
    </property>
</configuration>
```

修改文件 vim mapred-site.xml



```
[hadoop@slave01 hadoop]$ vim yarn-site.xml
```

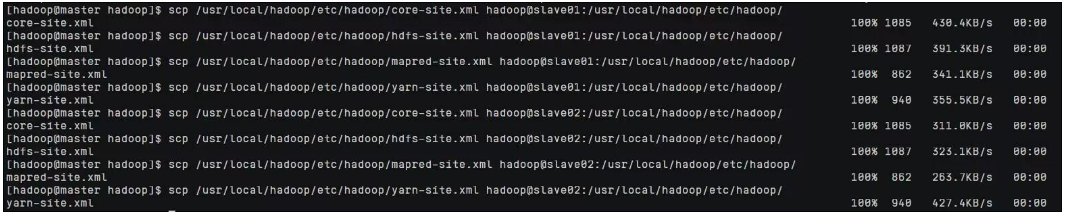


```
<configuration>

<!-- Site specific YARN configuration properties -->
      <property>
          <name>yarn.nodemanager.aux-services</name>
          <value>mapreduce_shuffle</value>
      </property>
      <property>
          <name>yarn.resourcemanager.hostname</name>
          <value>master</value>
      </property>

</conoiguratiot>
```

修改 yarn-site.xml。

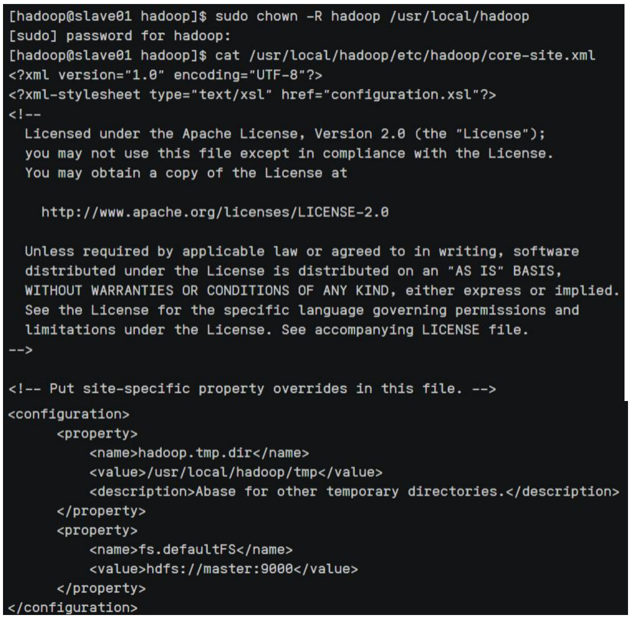
b.slave 节点配置（**如果只在 slave 节点上重复 master 节点上的配置而非通过传送文件会导致意外错误！！！**）

在 master 上发送 yarn-site.xml 到从机 slave01 上：



```
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/core-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
core-site.xml                                                                              100% 1085   430.4KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/hdfs-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
hdfs-site.xml                                                                              100% 1087   391.3KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/mapred-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
mapred-site.xml                                                                            100%  862   341.1KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/yarn-site.xml hadoop@slave01:/usr/local/hadoop/etc/hadoop/
yarn-site.xml                                                                              100%  940   355.5KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/core-site.xml hadoop@slave02:/usr/local/hadoop/etc/hadoop/
core-site.xml                                                                              100% 1085   311.0KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/hdfs-site.xml hadoop@slave02:/usr/local/hadoop/etc/hadoop/
hdfs-site.xml                                                                              100% 1087   323.1KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/mapred-site.xml hadoop@slave02:/usr/local/hadoop/etc/hadoop/
mapred-site.xml                                                                            100%  862   263.7KB/s   00:00
[hadoop@master hadoop]$ scp /usr/local/hadoop/etc/hadoop/yarn-site.xml hadoop@slave02:/usr/local/hadoop/etc/hadoop/
yarn-site.xml                                                                              100%  940   427.4KB/s   00:00
```

在 slave01 上设置文件权限、检查文件变更：

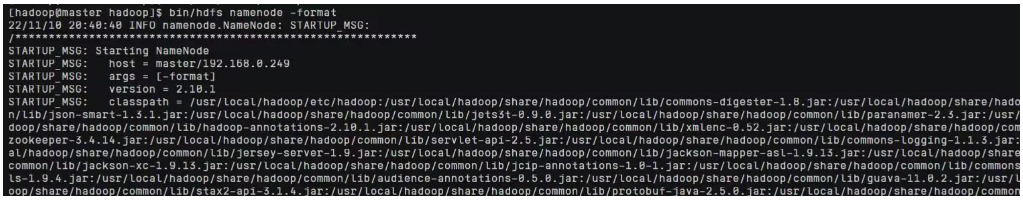


```
[hadoop@slave01 hadoop]$ sudo chown -R hadoop /usr/local/hadoop
[sudo] password for hadoop:
[hadoop@slave01 hadoop]$ cat /usr/local/hadoop/etc/hadoop/core-site.xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>
<!--
  Licensed under the Apache License, Version 2.0 (the "License");
  you may not use this file except in compliance with the License.
  You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

  Unless required by applicable law or agreed to in writing, software
  distributed under the License is distributed on an "AS IS" BASIS,
  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
  See the License for the specific language governing permissions and
  limitations under the License. See accompanying LICENSE file.
-->

<!-- Put site-specific property overrides in this file. -->
<configuration>
      <property>
          <name>hadoop.tmp.dir</name>
          <value>/usr/local/hadoop/tmp</value>
          <description>Abase for other temporary directories.</description>
      </property>
      <property>
          <name>fs.defaultFS</name>
          <value>hdfs://master:9000</value>
      </property>
</configuration>
```

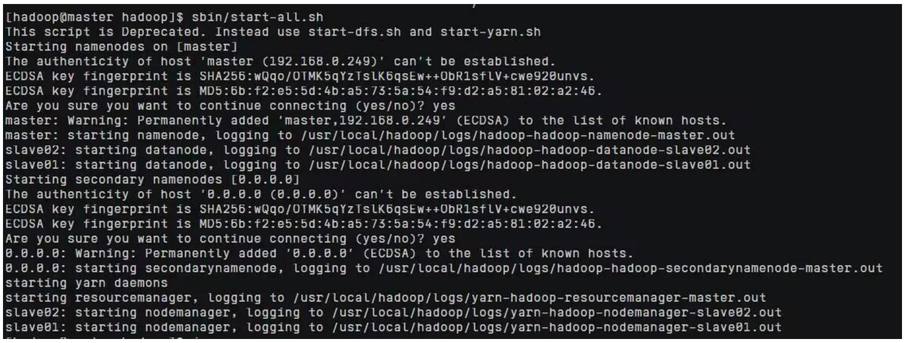
可以看到，输出中含有上一步中修改后的信息，确认正确。

c. 启动集群测试：

在 master 上启动集群：

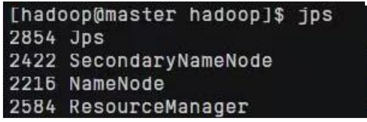


```
[hadoop@master hadoop]$ bin/hdfs namenode -format
22/11/10 20:40:40 INFO namenode.NameNode: STARTUP_MSG:
/************************************************************
STARTUP_MSG: Starting NameNode
STARTUP_MSG:   host = master/192.168.0.249
STARTUP_MSG:   args = [-format]
STARTUP_MSG:   version = 2.10.1
STARTUP_MSG:   classpath = /usr/local/hadoop/etc/hadoop:/usr/local/hadoop/share/hadoop/common/lib/commons-digester-1.8.jar:/usr/local/hadoop/share/hado
n/lib/json-smart-1.3.1.jar:/usr/local/hadoop/share/hadoop/common/lib/jets3t-0.9.0.jar:/usr/local/hadoop/share/hadoop/common/lib/paranamer-2.3.jar:/usr/
doop/share/hadoop/common/lib/hadoop-annotations-2.10.1.jar:/usr/local/hadoop/share/hadoop/common/lib/xmlenc-0.52.jar:/usr/local/hadoop/share/hadoop/com
zookeeper-3.4.14.jar:/usr/local/hadoop/share/hadoop/common/lib/servlet-api-2.5.jar:/usr/local/hadoop/share/hadoop/common/lib/commons-logging-1.1.3.jar:
al/hadoop/share/hadoop/common/lib/jersey-server-1.9.jar:/usr/local/hadoop/share/hadoop/common/lib/jackson-mapper-asl-1.9.13.jar:/usr/local/hadoop/share
common/lib/jackson-xc-1.9.13.jar:/usr/local/hadoop/share/hadoop/common/lib/jcip-annotations-1.0-1.jar:/usr/local/hadoop/share/hadoop/common/lib/commons
ls-1.9.4.jar:/usr/local/hadoop/share/hadoop/common/lib/audience-annotations-0.5.0.jar:/usr/local/hadoop/share/hadoop/common/lib/guava-11.0.2.jar:/usr/l
oop/share/hadoop/common/lib/stax2-api-3.1.4.jar:/usr/local/hadoop/share/hadoop/common/lib/protobuf-java-2.5.0.jar:/usr/local/hadoop/share/hadoop/common
```

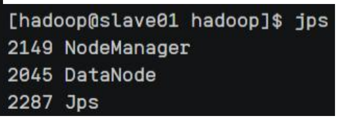


```
[hadoop@master hadoop]$ sbin/start-all.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
Starting namenodes on [master]
The authenticity of host 'master (192.168.0.249)' can't be established.
ECDSA key fingerprint is SHA256:wQqo/OTMK5qYzTslK6qsEw++ObR1sflV+cwe920unvs.
ECDSA key fingerprint is MD5:6b:f2:e5:5d:4b:a5:73:5a:54:f9:d2:a5:81:02:a2:46.
Are you sure you want to continue connecting (yes/no)? yes
master: Warning: Permanently added 'master,192.168.0.249' (ECDSA) to the list of known hosts.
master: starting namenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-namenode-master.out
slave02: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-slave02.out
slave01: starting datanode, logging to /usr/local/hadoop/logs/hadoop-hadoop-datanode-slave01.out
Starting secondary namenodes [0.0.0.0]
The authenticity of host '0.0.0.0 (0.0.0.0)' can't be established.
ECDSA key fingerprint is SHA256:wQqo/OTMK5qYzTslK6qsEw++ObR1sflV+cwe920unvs.
ECDSA key fingerprint is MD5:6b:f2:e5:5d:4b:a5:73:5a:54:f9:d2:a5:81:02:a2:46.
Are you sure you want to continue connecting (yes/no)? yes
0.0.0.0: Warning: Permanently added '0.0.0.0' (ECDSA) to the list of known hosts.
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop/logs/hadoop-hadoop-secondarynamenode-master.out
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-resourcemanager-master.out
slave02: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-nodemanager-slave02.out
slave01: starting nodemanager, logging to /usr/local/hadoop/logs/yarn-hadoop-nodemanager-slave01.out
```

测试 jps：



```
[hadoop@master hadoop]$ jps
2854 Jps
2422 SecondaryNameNode
2216 NameNode
2584 ResourceManager
```



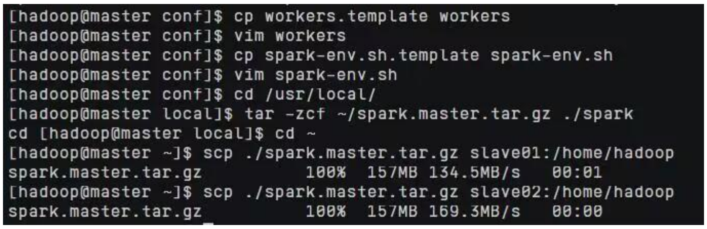
```
[hadoop@slave01 hadoop]$ jps
2149 NodeManager
2045 DataNode
2287 Jps
```

可以看到，master 节点出现 4 个进程，slave01 节点出现 3 个进程，slave02 同 01，配置成功！

（3）Spark 集群配置

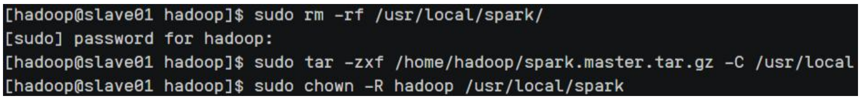
配置与启动方式与 Hadoop 相似，此处未过多记载；

a. 只在 master 上修改 spark 配置文件并复制 Spark 文件到各个 slave 节点：



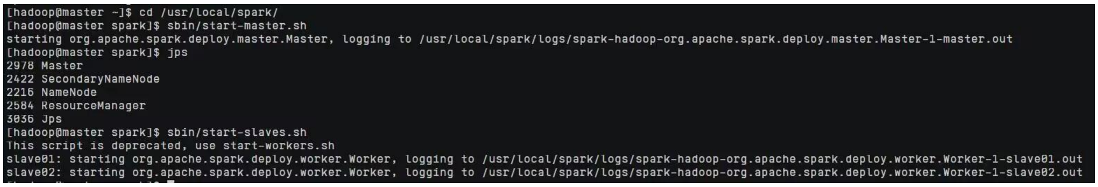
```
[hadoop@master conf]$ cp workers.template workers
[hadoop@master conf]$ vim workers
[hadoop@master conf]$ cp spark-env.sh.template spark-env.sh
[hadoop@master conf]$ vim spark-env.sh
[hadoop@master conf]$ cd /usr/local/
[hadoop@master local]$ tar -zcf ~/spark.master.tar.gz ./spark
cd [hadoop@master local]$ cd ~
[hadoop@master ~]$ scp ./spark.master.tar.gz slave01:/home/hadoop
spark.master.tar.gz          100%  157MB 134.5MB/s   00:01
[hadoop@master ~]$ scp ./spark.master.tar.gz slave02:/home/hadoop
spark.master.tar.gz          100%  157MB 169.3MB/s   00:00
```

b. 在 slave 节点删除原有 spark 并设置 spark 文件权限拥有者是 hadoop：



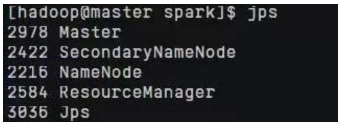
```
[hadoop@slave01 hadoop]$ sudo rm -rf /usr/local/spark/
[sudo] password for hadoop:
[hadoop@slave01 hadoop]$ sudo tar -zxf /home/hadoop/spark.master.tar.gz -C /usr/local
[hadoop@slave01 hadoop]$ sudo chown -R hadoop /usr/local/spark
```

c. 在 master 上启动 hadoop 集群和 master 节点：

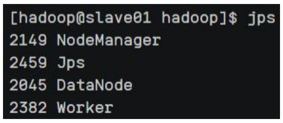


```
[hadoop@master ~]$ cd /usr/local/spark/
[hadoop@master spark]$ sbin/start-master.sh
starting org.apache.spark.deploy.master.Master, logging to /usr/local/spark/logs/spark-hadoop-org.apache.spark.deploy.master.Master-1-master.out
[hadoop@master spark]$ jps
2978 Master
2422 SecondaryNameNode
2216 NameNode
2584 ResourceManager
3036 Jps
[hadoop@master spark]$ sbin/start-slaves.sh
This script is deprecated, use start-workers.sh
slave01: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark/logs/spark-hadoop-org.apache.spark.deploy.worker.Worker-1-slave01.out
slave02: starting org.apache.spark.deploy.worker.Worker, logging to /usr/local/spark/logs/spark-hadoop-org.apache.spark.deploy.worker.Worker-1-slave02.out
```

d. 在 master 和 slave 上分别运行 jps 命令：



```
[hadoop@master spark]$ jps
2978 Master
2422 SecondaryNameNode
2216 NameNode
2584 ResourceManager
3036 Jps
```

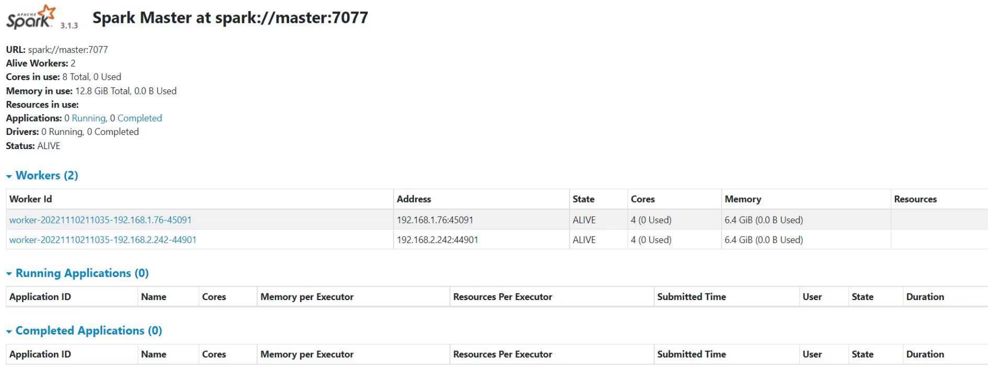


```
[hadoop@slave01 hadoop]$ jps
2149 NodeManager
2459 Jps
2045 DataNode
2382 Worker
```

测试成功！

（4）Web UI 查看

打开本地浏览器，输入 http://110.41.4.138:8080 可以看到最终实现的云服务器分布式搭建 Hadoop+Spark 环境。因为我们最初放开了指定端口，有内网穿透因此可以查看。Web UI 如下：



Spark Master at spark://master:7077

URL: spark://master:7077
Alive Workers: 2
Cores in use: 8 Total, 0 Used
Memory in use: 12.8 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

▾ Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20221110211035-192.168.1.76-45091	192.168.1.76:45091	ALIVE	4 (0 Used)	6.4 GiB (0.0 B Used)	
worker-20221110211035-192.168.2.242-44901	192.168.2.242:44901	ALIVE	4 (0 Used)	6.4 GiB (0.0 B Used)	

▾ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

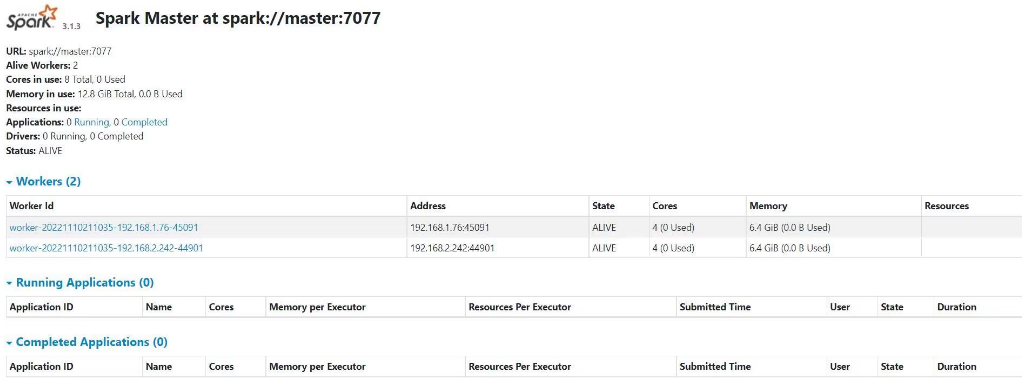
▾ Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

四、实验结果及分析和（或）源程序调试过程

实验结果

实验采用云服务器分布式搭建 Hadoop&Sapak 环境，最终得到的 Web UI 如下：



Spark Master at spark://master:7077

URL: spark://master:7077
Alive Workers: 2
Cores in use: 8 Total, 0 Used
Memory in use: 12.8 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

▾ Workers (2)

Worker Id	Address	State	Cores	Memory	Resources
worker-20221110211035-192.168.1.76-45091	192.168.1.76:45091	ALIVE	4 (0 Used)	6.4 GiB (0.0 B Used)	
worker-20221110211035-192.168.2.242-44901	192.168.2.242:44901	ALIVE	4 (0 Used)	6.4 GiB (0.0 B Used)	

▾ Running Applications (0)

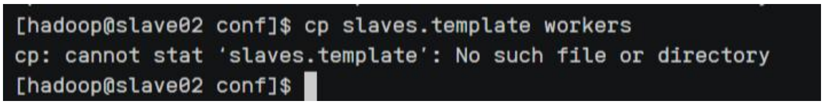
Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

▾ Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration

调试过程

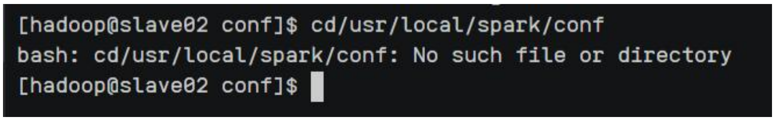
1.执行命令时无法找到指定文件：



```
[hadoop@slave02 conf]$ cp slaves.template workers
cp: cannot stat ‘slaves.template’: No such file or directory
[hadoop@slave02 conf]$
```

错误原因是 slaves 被更名为 workers，需要替换后重新尝试；

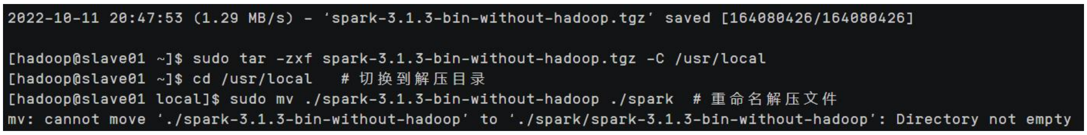
2.切换配置目录时无法找到指定位置：



```
[hadoop@slave02 conf]$ cd/usr/local/spark/conf
bash: cd/usr/local/spark/conf: No such file or directory
[hadoop@slave02 conf]$
```

错误原因是将 spark 安装在 root 里，安装前应先进入 hadoop；

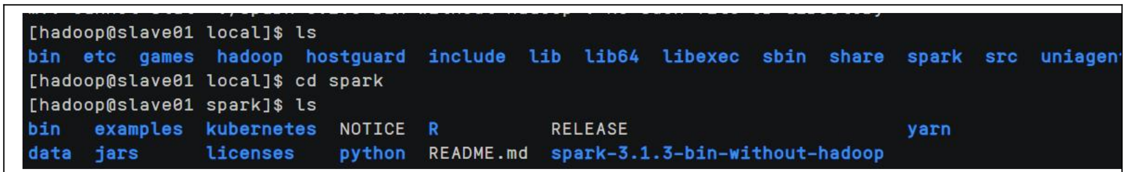
3.解压 spark 后重命名失败：



```
2022-10-11 20:47:53 (1.29 MB/s) - ‘spark-3.1.3-bin-without-hadoop.tgz’ saved [164080426/164080426]

[hadoop@slave01 ~]$ sudo tar -zxf spark-3.1.3-bin-without-hadoop.tgz -C /usr/local
[hadoop@slave01 ~]$ cd /usr/local    # 切换到解压目录
[hadoop@slave01 local]$ sudo mv ./spark-3.1.3-bin-without-hadoop ./spark  # 重命名解压文件
mv: cannot move './spark-3.1.3-bin-without-hadoop' to './spark/spark-3.1.3-bin-without-hadoop': Directory not empty
```

错误原因是下载错了版本，导致解压后找不到正确文件名对应的文件，因此无法重命名；

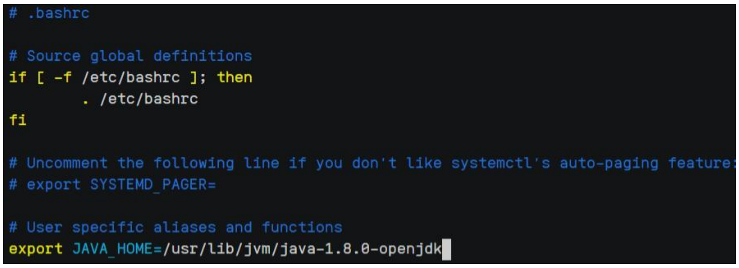
4.解压 spark 后重命名失败后无法删除：



```
[hadoop@slave01 local]$ ls
bin  etc  games  hadoop  hostguard  include  lib  lib64  libexec  sbin  share  spark  src  uniagen
[hadoop@slave01 local]$ cd spark
[hadoop@slave01 spark]$ ls
bin   examples  kubernetes  NOTICE  R          RELEASE                        yarn
data  jars      licenses    python  README.md  spark-3.1.3-bin-without-hadoop
```

ls 指令查看文件夹下内容并删除；

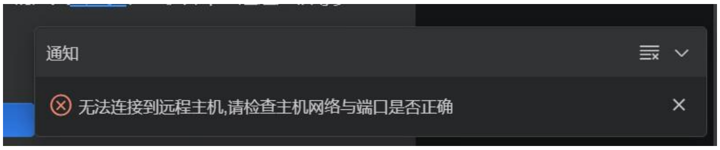
5. 编辑配置文件保存后页面锁死：



```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk
```

千万**不能 Ctrl+S**，否则页面会被锁定！按 Ctrl+Q 解锁，然后 Esc 退出编辑，“:wq”保存并退出。

6.配置内网互联后无法远程登陆服务器：

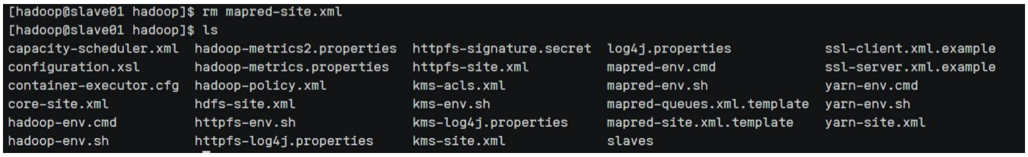


通知

ⓧ 无法连接到远程主机,请检查主机网络与端口是否正确

需要切换 VPC 并解绑默认子网。

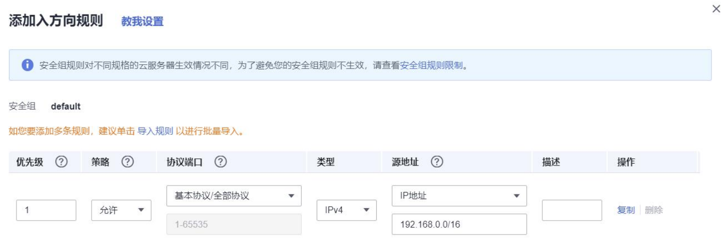
7.从机配置文件出错，删除多配置的文件：



```
[hadoop@slave01 hadoop]$ rm mapred-site.xml
[hadoop@slave01 hadoop]$ ls
capacity-scheduler.xml  hadoop-metrics2.properties  httpfs-signature.secret  log4j.properties          ssl-client.xml.example
configuration.xsl       hadoop-metrics.properties   httpfs-site.xml          mapred-env.cmd            ssl-server.xml.example
container-executor.cfg  hadoop-policy.xml           kms-acls.xml             mapred-env.sh             yarn-env.cmd
core-site.xml           hdfs-site.xml               kms-env.sh               mapred-queues.xml.template  yarn-env.sh
hadoop-env.cmd          httpfs-env.sh               kms-log4j.properties     mapred-site.xml.template  yarn-site.xml
hadoop-env.sh           httpfs-log4j.properties     kms-site.xml             slaves
```

操作完后 ls 指令查看当前文件夹下文件。

8.web UI 界面看不到 worker：



添加入方向规则　教我设置

安全组规则对不同规格的云服务器生效情况不同，为了避免您的安全组规则不生效，请查看安全组规则限制。

安全组　default

如您要添加多条规则，建议单击 导入规则 以进行批量导入。

优先级	策略	协议端口	类型	源地址	描述	操作
1	允许	基本协议/全部协议 1-65535	IPv4	IP地址 192.168.0.0/16		复制 删除

因为端口未放通，添加入方向规则中全部规则。

实验感悟：

本次实验中，我们小组遇见了不少的问题，最开始没有安装单机版 spark，集群是总是无法实现互联。然后又因为忘记进入 hadoop 导致操作在服务器 root 下，互免总是失败。之后又是镜像网址的问题，由于指导文件时间有些久远，导致镜像网址以及 Hadoop 和 Spark 文件包版本过低，后来对文件的网址和版本进行了更新才成功下载。由于版本号的变更，后续解压等指令也需要更改，一开始没有注意导致总是找不到指定文件。

在设置互免时还因为 IP 地址填写错误停滞了一段时间，检查许久才发现问题所

在。然后又因为不知名意外在 master 向 slave01 发送验证时出现错误，导致后来几次发送都出现问题。

最终经过团队 3 人两天的努力后，终于完成了 Hadoop+Spark 的集群，而且经过测试后，能够出现同指导书一致的进程结果和 Web UI。

后来做到实验三时，出现计算内存不足，以及 slave 缺失 Java 环境等问题，前者亟需扩大计算资源，后者问题源自实验 1 忘记在从机配置 Java 环境。于是我们组决定重做实验 1，并且创建了计算资源更加丰富的 4vCPUs 8 GiB 服务器，并配置了内网互连。第二次配置让我们对 hadoop+spark 集群又有了更深的感悟。