

关卡 1

部署 MySQL、Nginx、 Hadoop 和 Hive

实验手册



华为技术有限公司



目录

1 实验环境介绍	2
1.1 实验介绍	2
1.1.1 实验简介	2
1.1.2 实验目的	2
1.1.3 软件介绍	2
2 实验环境部署	3
2.1 购买 ECS	3
2.1.1 购买 VPC 及安全组配置	3
2.1.2 购买 ECS	7
2.1.3 通过 ssh 登录系统	10
2.2 部署 Nginx	11
2.2.1 编译安装 Nginx	11
2.2.2 启动 nginx 服务	13
2.2.3 测试 Nginx	14
2.3 部署 MySQL	14
2.3.1 创建 MySQL 用户	14
2.3.2 安装 MySQL	14
2.3.3 启动和登录数据库	16
2.4 部署 Hadoop	17
2.4.1 安装 Hadoop	17
2.4.2 启动 HDFS 和 Yarn。	19
2.4.3 验证 Hadoop	20
2.5 部署 Hive	21
2.5.1 配置 MySQL	21
2.5.2 安装 Hive	23
2.5.3 启动 hive	24
2.5.4 连接测试 hive	25
2.6 思考题	25

1 实验环境介绍

1.1 实验介绍

1.1.1 实验简介

基于云环境的开发和部署已逐渐成为主流，其中环境部署也是开发过程中必不可少的一环，本次实验将带领大家一起完成环境的部署。

1.1.2 实验目的

本实验的主要目的是在华为云服务器上指导搭建后续实验需要的环境，包括 Nginx、MySQL、Hadoop 和 Hive，方便后续关卡的进行。

1.1.3 软件介绍

本实验设计的主要软件及链接如下表（以下链接都是官网链接，后续实验步骤中有下载更快的链接地址）。

软件名称	下载链接
mysql.tar.gz	由讲师提供
nginx-1.16.1.tar.gz	http://nginx.org/download/nginx-1.16.1.tar.gz
hadoop-3.3.1-aarch64.tar.gz	https://downloads.apache.org/hadoop/common/hadoop-3.3.1/hadoop-3.3.1-aarch64.tar.gz
apache-hive-3.1.2-bin.tar.gz	https://archive.apache.org/dist/hive/hive-3.1.2/apache-hive-3.1.2-bin.tar.gz

2 实验环境部署

2.1 购买 ECS

2.1.1 购买 VPC 及安全组配置

步骤 1 浏览器输入 www.huaweicloud.com，登录华为云官网。



步骤 2 点击右上角的登录，打开登录窗口。

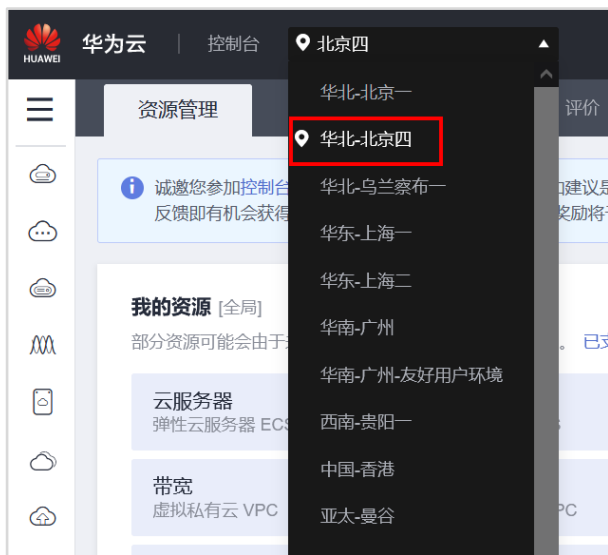


步骤 3 按要求输入账号密码，登录华为云。

步骤 4 点击右上角的“控制台”，打开控制台界面。



步骤 5 切换区域为“华北-北京四”



步骤 6 鼠标移动到控制台界面“我的资源”下面的“虚拟私有云 VPC”，点击“华北-北京四”，进入 VPC 控制台。



步骤 7 点击右上角的“创建虚拟私有云”。



步骤 8 按照如下表配置 VPC 参数。

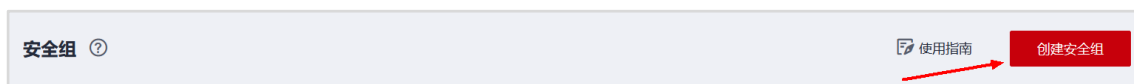
参数	配置
区域	华北-北京四
名称	vpc-test
网段	任意填写，如10.0.0.0/24
企业项目	default
默认子网 可用区	可用区1
默认子网 名称	subnet-test
子网网段	任意填写，如10.0.0.0/24

步骤 9 配置完成后，点击“立即创建”，系统会自动回到 VPC 控制台。

步骤 10 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。



步骤 11 点击右上角的“创建安全组”。



步骤 12 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

×

★ 名称

sg-test

★ 企业项目

default

新建企业项目 ?

★ 模板

通用Web服务器

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

步骤 13 在安全组列表，找到“sg-test”，点击配置规则，进入规则配置页面。

安全组 ?

使用指南

创建安全组

所有项目

名称 sg-test

名称	安全组规则	关联实例	描述	企业项目	操作
sg-test	14	0	通用Web服务器，...	鲲鹏云项目	配置规则 管理实例 更多

步骤 14 进入“入方向规则”，点击“添加规则”。

步骤 15 输入优先级为 1，协议端口为 8081，描述为 Nginx 端口。

添加方向规则 [教我设置](#)

安全组 sg-test

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	TCP 8081	IPv4	IP地址 0.0.0.0/0	Nginx端口	复制 删除

增加1条规则

确定 取消

步骤 16 点击添加 1 条规则，按照如下内容（优先级均为 1），添加规则，确定。

端口	描述
8081	Nginx端口
50070	Hadoop端口
8084	代码迁移工具端口
8086	HyperTuner端口

添加方向规则 [教我设置](#)

优先级	策略	协议端口	类型	源地址	描述	操作
1	允许	TCP 8081	IPv4	IP地址 0.0.0.0/0	Nginx端口	复制 删除
1	允许	TCP 50070	IPv4	IP地址 0.0.0.0/0	Hadoop端口	复制 删除
1	允许	TCP 8084	IPv4	IP地址 0.0.0.0/0	代码迁移工	复制 删除
1	允许	TCP 8086	IPv4	IP地址 0.0.0.0/0	HyperTuner	复制 删除

增加1条规则

确定 取消

2.1.2 购买 ECS

步骤 1 鼠标移动到左侧导航栏，点击“弹性云服务器 ECS”，进入 ECS 控制台。



步骤 2 在 ECS 控制台界面点击右上角的“购买弹性云服务器”。



步骤 3 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.xlarge.2 4vCPUs 8GB
镜像	公共镜像 openEuler openEuler 20.03 64bit with ARM(40GB)
系统盘	高IO 80GB

步骤 4 配置完成后点击“下一步：网络配置”，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

步骤 5 配置完成后，点击“下一步：高级配置”，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
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

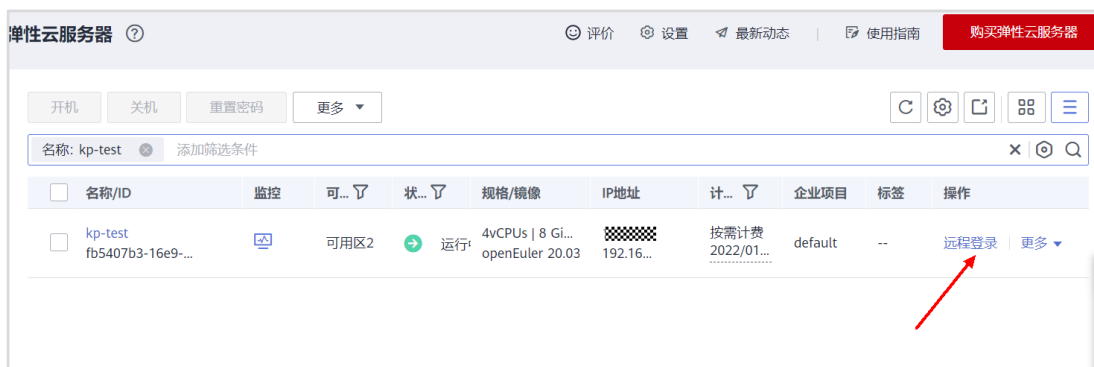
步骤 6 配置完成后点击右下角“下一步：确认配置”。企业项目为 default，勾选同意协议，然后点击：立即购买。



步骤 7 在提交任务成功后，点击“返回云服务器列表”，返回 ECS 控制台。

2.1.3 通过 ssh 登录系统

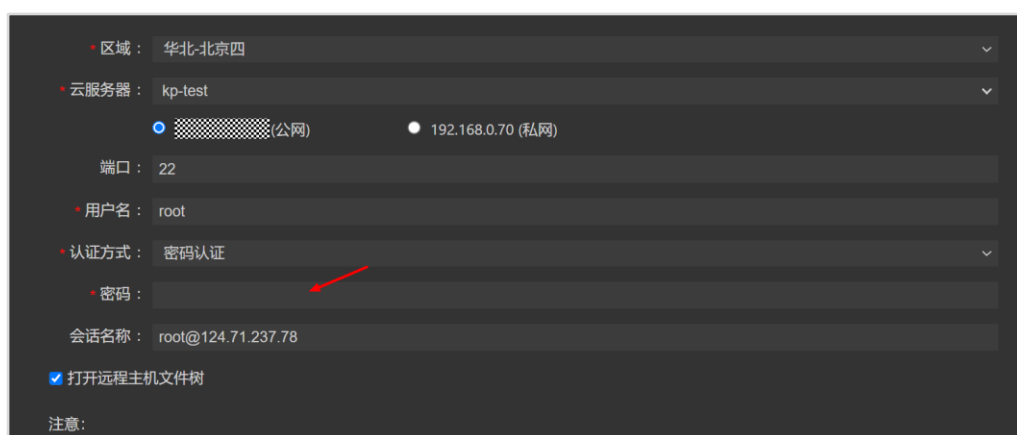
步骤 1 在控制台单击远程登录按钮。

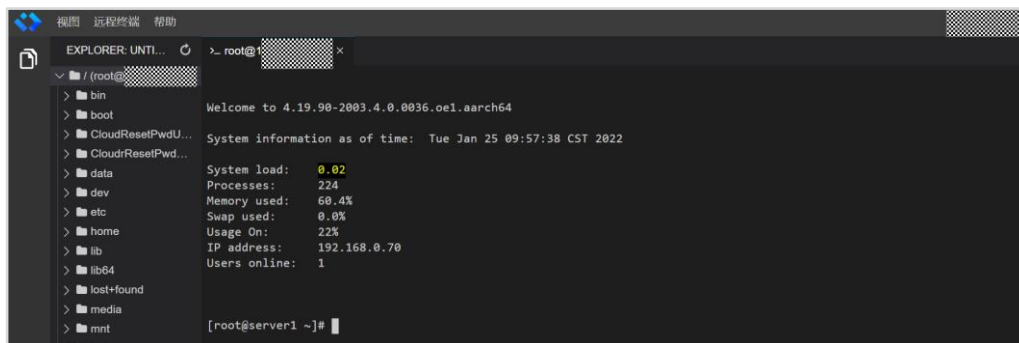


步骤 2 在弹窗中点击 CloudShell 登录。



步骤 3 在新弹出的页面中输入 ECS 登录密码，点击“连接”登录终端。





2.2 部署 Nginx

2.2.1 编译安装 Nginx

步骤 1 修改 openEuler yum 源文件。

进入到源文件所在目录。

```
[root@ecs-3ddf ~]# cd /etc/yum.repos.d/
[root@ecs-3ddf yum.repos.d]# cp openEuler.repo openEuler.repo.bak
```

使用 vim 命令打开 openEuler.repo 源文件。

```
[root@ecs-3ddf yum.repos.d]# vim openEuler.repo
```

按 shift+:进入到命令模式，输入 %s/gpgcheck=1/gpgcheck=0/g，将所有 gpgcheck=1 改为 0。最后按 shift+:进入到命令模式，然后输入 wq!保存并退出。

步骤 2 安装依赖包。

```
[root@ecs-3ddf yum.repos.d]# yum install make unzip pcre pcre-devel zlib zlib-devel readline readline-devel ncurses ncurses-devel perl-devel perl-ExtUtils-Embed openssl-devel -y
```

```
Running transaction
  Preparing      : readline-devel-7.0-13.oe1.aarch64                1/1
  Installing     : readline-devel-7.0-13.oe1.aarch64                1/2
  Installing     : pcre-devel-8.43-5.oe1.aarch64                    2/2
  Running scriptlet: pcre-devel-8.43-5.oe1.aarch64                    2/2
  Verifying      : pcre-devel-8.43-5.oe1.aarch64                    1/2
  Verifying      : readline-devel-7.0-13.oe1.aarch64                2/2

Installed:
  pcre-devel-8.43-5.oe1.aarch64                                readline-devel-7.0-13.oe1.aarch64

Complete!
[root@ecs-3ddf ~]#
```

步骤 3 下载压缩包。

```
[root@ecs-3ddf ~]# cd /home
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/nginx-1.16.1.tar.gz
```

```
[root@ecs-3ddf home]# wget -c http://nginx.org/download/nginx-1.16.1.tar.gz
--2022-01-19 16:05:16-- http://nginx.org/download/nginx-1.16.1.tar.gz
Resolving nginx.org (nginx.org)... 52.58.199.22, 3.125.197.172, 2a05:d014:edb:5704::6, ...
Connecting to nginx.org (nginx.org)|52.58.199.22|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1032630 (1008K) [application/octet-stream]
Saving to: 'nginx-1.16.1.tar.gz'

nginx-1.16.1.tar.gz      100%[=====] 1008K  1.13MB/s  in 0.9s
2022-01-19 16:05:18 (1.13 MB/s) - 'nginx-1.16.1.tar.gz' saved [1032630/1032630]
```

步骤 4 解压安装包。

```
[root@ecs-3ddf home]# tar -zxvf nginx-1.16.1.tar.gz
```

步骤 5 进入安装目录。

```
[root@ecs-3ddf home]# cd nginx-1.16.1
```

步骤 6 编译安装 Nginx。

```
[root@ecs-3ddf nginx-1.16.1]# chmod 777 *
```

```
[root@ecs-3ddf nginx-1.16.1]# ./configure --prefix=/usr/local/nginx --with-http_ssl_module
```

```
Configuration summary
+ using system PCRE library
+ using system OpenSSL library
+ using system zlib library

nginx path prefix: "/usr/local/nginx"
nginx binary file: "/usr/local/nginx/sbin/nginx"
nginx modules path: "/usr/local/nginx/modules"
nginx configuration prefix: "/usr/local/nginx/conf"
nginx configuration file: "/usr/local/nginx/conf/nginx.conf"
nginx pid file: "/usr/local/nginx/logs/nginx.pid"
nginx error log file: "/usr/local/nginx/logs/error.log"
nginx http access log file: "/usr/local/nginx/logs/access.log"
nginx http client request body temporary files: "client_body_temp"
nginx http proxy temporary files: "proxy_temp"
nginx http fastcgi temporary files: "fastcgi_temp"
nginx http uwsgi temporary files: "uwsgi_temp"
nginx http scgi temporary files: "scgi_temp"

[root@ecs-3ddf nginx-1.16.1]#
```

```
[root@ecs-3ddf nginx-1.16.1]# make -j 8 && make install
```

```
cp conf/uwsgi_params \
    '/usr/local/nginx/conf/uwsgi_params.default'
test -f '/usr/local/nginx/conf/scgi_params' \
    || cp conf/scgi_params '/usr/local/nginx/conf'
cp conf/scgi_params \
    '/usr/local/nginx/conf/scgi_params.default'
test -f '/usr/local/nginx/conf/nginx.conf' \
    || cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf'
cp conf/nginx.conf '/usr/local/nginx/conf/nginx.conf.default'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
test -d '/usr/local/nginx/html' \
    || cp -R html '/usr/local/nginx'
test -d '/usr/local/nginx/logs' \
    || mkdir -p '/usr/local/nginx/logs'
make[1]: Leaving directory '/home/nginx-1.16.1'
[root@ecs-3ddf nginx-1.16.1]#
```

步骤 7 新增 Nginx 用户。

```
[root@ecs-3ddf nginx-1.16.1]# useradd nginx
```

步骤 8 刷新 Nginx 用户权限。

给 Nginx 用户开启 Nginx 安装目录权限。

```
[root@ecs-3ddf nginx-1.16.1]# chown -R nginx /usr/local/nginx
```

```
[root@ecs-3ddf nginx-1.16.1]# chmod -R 777 /home/nginx
```

步骤 9 查看 Nginx 版本。

```
[root@ecs-3ddf nginx-1.16.1]# cd /usr/local/nginx/sbin/
```

```
[root@ecs-3ddf sbin]# ./nginx -v
```

```
[root@ecs-3ddf nginx-1.16.1]# useradd nginx
[root@ecs-3ddf nginx-1.16.1]# chown nginx:nginx /usr/local/nginx
[root@ecs-3ddf nginx-1.16.1]# chmod -R 777 /home/nginx
[root@ecs-3ddf nginx-1.16.1]# cd /usr/local/nginx/sbin/
[root@ecs-3ddf sbin]# ./nginx -v
nginx version: nginx/1.16.1
[root@ecs-3ddf sbin]#
```

2.2.2 启动 nginx 服务

执行以下命令，启动 Nginx 并查看进程信息。

```
[root@ecs-3ddf sbin]# ps -ef | grep nginx
[root@ecs-3ddf sbin]# /usr/local/nginx/sbin/nginx -s stop
[root@ecs-3ddf sbin]# /usr/local/nginx/sbin/nginx
[root@ecs-3ddf sbin]# ps -ef | grep nginx
```

```
[root@ecs-3ddf conf]# ps -ef | grep nginx
root      10918      7311    0 16:30 pts/0      00:00:00 grep --color=auto nginx
[root@ecs-3ddf conf]# /usr/local/nginx/sbin/nginx -s stop
nginx: [error] open() "/usr/local/nginx/logs/nginx.pid" failed (2: No such file or directory)
[root@ecs-3ddf conf]# /usr/local/nginx/sbin/nginx
[root@ecs-3ddf conf]# ps -ef | grep nginx
root      10945        1    0 16:31 ?          00:00:00 nginx: master process /usr/local/nginx/sbin/nginx
nginx     10946     10945    0 16:31 ?          00:00:00 nginx: worker process
root      10960      7311    0 16:31 pts/0      00:00:00 grep --color=auto nginx
```

2.2.3 测试 Nginx

打开浏览器，访问“弹性公网 ip:8081”，出现如下界面，Nginx 部署完成。（确保 8081 端口已经开放）

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

2.3 部署 MySQL

2.3.1 创建 MySQL 用户

创建 mysql 用户，第三条命令执行后设置 mysql 用户密码（此处输入密码不会有任何显示）。

```
[root@ecs-3ddf sbin]# cd /home
[root@ecs-3ddf home]# groupadd mysql
[root@ecs-3ddf home]# useradd -g mysql mysql
[root@ecs-3ddf home]# passwd mysql
```

2.3.2 安装 MySQL

步骤 1 下载 mysql 源码包。

```
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/mysql.tar.gz
```

步骤 2 查看 mysql 源码包大小。

此时，下载的 tar 包已在云服务器/home 目录下，通过 ll 查看，文件大小为 1.6G。

```
[root@ecs-3ddf home]# ll
```

步骤 3 安装依赖包。

```
[root@ecs-3ddf home]# yum -y install bison ncurses ncurses-devel libaio-devel openssl openssl-devel gmp gmp-devel
mpfr mpfr-devel libmpc libmpc-devel wget tar gcc gcc-c++ git rpcgen cmake
```

```
Installed:
bison-3.5-2.oe1.aarch64      cmake-3.12.1-5.oe1.aarch64      gmp-devel-1:6.1.2-10.oe1.aarch64
libaio-devel-0.3.111-5.oe1.aarch64  mpfr-devel-3.1.6-3.oe1.aarch64  ncurses-devel-6.1-14.oe1.aarch64
openssl-devel-1:1.1.1d-9.oe1.aarch64  libmpc-devel-1.1.0-3.oe1.aarch64  rpcgen-1.4-2.oe1.aarch64
cmake-data-3.12.1-5.oe1.noarch      emacs-filesystem-3.12.1-5.oe1.aarch64  cmake-rpm-macros-3.12.1-5.oe1.noarch
e2fsprogs-devel-1.45.3-4.oe1.aarch64  keyutils-libs-devel-1.5.10-11.oe1.aarch64  gmp-c++-1:6.1.2-10.oe1.aarch64
jsoncpp-1.8.4-6.oe1.aarch64      libselinux-devel-2.9-1.oe1.aarch64  krb5-devel-1.17-9.oe1.aarch64
libaio-0.3.111-5.oe1.aarch64      libb2-devel-0.3.1-2.oe1.aarch64  libsepol-devel-2.9-1.oe1.aarch64
libuv-1:1.23.0-2.oe1.aarch64      zlib-devel-1.2.11-17.oe1.aarch64  pcre2-devel-10.33-2.oe1.aarch64
rhash-1.3.5-5.oe1.aarch64

Complete!
```

步骤 4 配置 MySQL 环境变量

使用 vim 打开/etc/profile。

```
[root@ecs-3ddf home]# vim /etc/profile
```

按“shift+g”拉到文件最下方，按“i”进入 insert 模式，添加 export 语句，完成后按 esc 键，最后输入“:wq”保存退出 vim。

```
export MYSQL_HOME=/usr/local/mysql
export PATH=$MYSQL_HOME/bin:$PATH
```

```
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`id -gn`" = "`id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge

if [ -n "${BASH_VERSION-}" ] ; then
    if [ -f /etc/bashrc ] ; then
        # Bash login shells run only /etc/profile
        # Bash non-login shells run only /etc/bashrc
        # Check for double sourcing is done in /etc/bashrc.
        . /etc/bashrc
    fi
fi

export MYSQL_HOME=/usr/local/mysql
export PATH=$MYSQL_HOME/bin:$PATH

:wq
```

source profile 文件使环境变量生效。


```
[root@ecs-3ddf home]# source /etc/profile
```

步骤 5 下载并执行自动化配置脚本 auto_mysql.sh。

下载脚本。

```
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/auto_mysql.sh
```

执行 auto_mysql.sh。

```
[root@ecs-3ddf home]# sh auto_mysql.sh
```

步骤 6 切换至 mysql 用户。

```
[root@ecs-3ddf home]# su - mysql
```

步骤 7 初始化 mysql。

```
[mysql@ecs-3ddf home]$ cd /usr/local/mysql/bin
[mysql@ecs-3ddf bin]$ mysqld --defaults-file=/etc/my.cnf --initialize
```

```
[mysql@ecs-3ddf bin]$ mysqld --defaults-file=/etc/my.cnf --initialize
2022-01-19T07:49:09.479975Z 0 [System] [MY-013169] [Server] /usr/local/mysql/bin/mysqld (mysqld 8.0.17) initializing of server
in progress as process 7676
2022-01-19T07:49:11.413825Z 5 [Note] [MY-010454] [Server] A temporary password is generated for root@localhost: u9ohp.f5k2%t
2022-01-19T07:49:12.806760Z 0 [System] [MY-013170] [Server] /usr/local/mysql/bin/mysqld (mysqld 8.0.17) initializing of server
has completed
```

以上步骤回显倒数第 2 行中有初始密码，请注意保存，之后会用到。

2.3.3 启动和登录数据库

步骤 1 启动数据库服务。

```
[mysql@ecs-3ddf bin]$ service mysql start
```

```
[mysql@ecs-3ddf bin]$ service mysql start
Starting MySQL. SUCCESS!
```

步骤 2 登录数据库。

```
[mysql@ecs-3ddf bin]$ mysql -uroot -p -S /data/mysql/run/mysql.sock
```

或者，

```
/usr/local/mysql/bin/mysql -uroot -p -S /data/mysql/run/mysql.sock
```

使用之前的初始密码登录。

```
[mysql@ecs-3ddf bin]$ mysql -uroot -p -S /data/mysql/run/mysql.sock
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.17

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

步骤 3 配置数据库帐号密码。

登录数据库以后，修改通过 root 用户登录数据库的密码。（注意密码必须 123456，会影响后面的 hive）

```
mysql> alter user 'root'@'localhost' identified by "123456";
```

创建全域 root 用户（允许 root 从其他服务器访问）。

```
mysql> create user 'root'@'%' identified by '123456';
```

进行授权。

```
mysql> grant all privileges on *.* to 'root'@'%';
mysql> flush privileges;
```

步骤 4 退出数据库。

```
mysql> exit;
```

步骤 5 退出 mysql 用户。

```
[mysql@ecs-3ddf bin]$ exit
```

重新回到 home 目录下。

```
[root@ecs-3ddf home]# cd /home
```

2.4 部署 Hadoop

2.4.1 安装 Hadoop

步骤 1 下载 OpenJDK 软件包。

下载 ARM 版本的 OpenJDK。

```
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/OpenJDK8U-jdk_aarch64_linux_hotspot_8u252b09.tar.gz
```

步骤 2 将 OpenJDK 添加到环境变量。

添加环境变量，输入 `vim /etc/profile`，按 “shift+g” 拉到文件最下方，按 “i” 进入 insert 模式，添加 `export` 语句，完成后按 `esc` 键，最后输入 “:wq” 保存退出 `vim`。

```
[root@ecs-3ddf home]# vim /etc/profile
export JAVA_HOME=/usr/local/jdk8u252-b09
export PATH=$JAVA_HOME/bin:$PATH
```

```
if [ -n "${BASH_VERSION}" ] ; then
    if [ -f /etc/bashrc ] ; then
        # Bash login shells run only /etc/profile
        # Bash non-login shells run only /etc/bashrc
        # Check for double sourcing is done in /etc/bashrc.
        . /etc/bashrc
    fi
fi

export MYSQL_HOME=/usr/local/mysql
export PATH=$MYSQL_HOME/bin:$PATH

export JAVA_HOME=/usr/local/jdk8u252-b09
export PATH=$JAVA_HOME/bin:$PATH
```

生效环境变量。

```
[root@ecs-3ddf home]# source /etc/profile
```

步骤 3 下载 Hadoop 软件包。

下载 Hadoop 的 arm 软件压缩包 `hadoop-3.3.1-aarch64.tar.gz`。

```
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/hadoop-3.3.1-aarch64.tar.gz
```

步骤 4 将 Hadoop 添加到环境变量。

输入 `vim /etc/profile`，按 “shift+g” 拉到文件最下方，按 “i” 进入 insert 模式，添加以下 `export` 语句，完成后按 `esc` 键，最后输入 “:wq” 保存退出 `vim`。

```
[root@ecs-3ddf home]# vim /etc/profile
export HADOOP_HOME=/usr/local/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

```
export MYSQL_HOME=/usr/local/mysql
export PATH=$MYSQL_HOME/bin:$PATH

export JAVA_HOME=/usr/local/jdk8u252-b09
export PATH=$JAVA_HOME/bin:$PATH

export HADOOP_HOME=/usr/local/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
```

生效环境变量。

```
[root@ecs-3ddf home]# source /etc/profile
```

步骤 5 下载并执行自动化配置脚本 `auto_hadoop.sh`。

下载脚本。

```
[root@ecs-3ddf home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/auto_hadoop.sh
```

执行 auto_hadoop.sh。

```
[root@ecs-3ddf home]# sh auto_hadoop.sh
```

步骤 6 格式化 HDFS。

```
[root@ecs-3ddf home]# cd $HADOOP_HOME/etc/hadoop
[root@ecs-3ddf hadoop]# hdfs namenode -format
```

```
2022-01-19 17:49:56,225 INFO namenode.FSImageFormatProtobuf: Image file /data/data1/hadoop/nn/current/fsimage.ckpt_0000000000
00000000 of size 399 bytes saved in 0 seconds .
2022-01-19 17:49:56,240 INFO namenode.NNStorageRetentionManager: Going to retain 1 images with txid >= 0
2022-01-19 17:49:56,263 INFO namenode.FSNamesystem: Stopping services started for active state
2022-01-19 17:49:56,263 INFO namenode.FSNamesystem: Stopping services started for standby state
2022-01-19 17:49:56,266 INFO namenode.FSImage: FSImageSaver clean checkpoint: txid=0 when meet shutdown.
2022-01-19 17:49:56,266 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at server1/192.168.0.70
*****/
[root@ecs-3ddf hadoop]#
```

2.4.2 启动 HDFS 和 Yarn。

步骤 1 启动 HDFS。

```
[root@ecs-3ddf hadoop]# cd /usr/local/hadoop/sbin
[root@ecs-3ddf sbin]# ./start-dfs.sh
```

```
[root@ecs-3ddf sbin]# ./start-dfs.sh
Starting namenodes on [server1]
Last login: Wed Jan 19 17:54:35 CST 2022 on pts/0
server1:
server1: Authorized users only. All activities may be monitored and reported.
Starting datanodes
Last login: Wed Jan 19 17:56:08 CST 2022 on pts/0
localhost:
localhost: Authorized users only. All activities may be monitored and reported.
Starting secondary namenodes [server1]
Last login: Wed Jan 19 17:56:11 CST 2022 on pts/0
server1:
server1: Authorized users only. All activities may be monitored and reported.
[root@ecs-3ddf sbin]#
```

步骤 2 启动 Yarn。

```
[root@ecs-3ddf sbin]# ./start-yarn.sh
```

```
[root@ecs-3ddf sbin]# ./start-yarn.sh
Starting resourcemanager
Last login: Wed Jan 19 17:56:14 CST 2022 on pts/0
Starting nodemanagers
Last login: Wed Jan 19 17:56:54 CST 2022 on pts/0
localhost:
localhost: Authorized users only. All activities may be monitored and reported.
[root@ecs-3ddf sbin]#
```

如果出现 permission denied 的报错。

```
[root@server1 sbin]# ./start-yarn.sh
Starting resourcemanager
Last login: Fri Aug 27 10:20:23 CST 2021 on pts/0
Starting nodemanagers
Last login: Fri Aug 27 10:21:00 CST 2021 on pts/0
localhost:
localhost: Authorized users only. All activities may be monitored and reported.
localhost: root@localhost: Permission denied (publickey,gssapi-keyex,gssapi-with-mic,password).
```

```
[root@ecs-3ddf sbin]# ssh-keygen -t rsa -P ""
[root@ecs-3ddf sbin]# cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys
```

然后重新启动 HDFS 和 Yarn。

```
[root@ecs-3ddf sbin]# ./start-dfs.sh
[root@ecs-3ddf sbin]# ./start-yarn.sh
```

观察进程启动是否正常并回到 home 目录下。

```
[root@ecs-3ddf sbin]# jps
[root@ecs-3ddf sbin]# cd /home
```

```
[root@ecs-3ddf sbin]# jps
14657 SecondaryNameNode
1873 WrapperSimpleApp
15603 Jps
12854 NameNode
13096 DataNode
15146 NodeManager
14959 ResourceManager
```

2.4.3 验证 Hadoop

在浏览器中输入 URL 地址，访问 Hadoop Web 页面，URL 格式：<http://your server ip:50070>

通过观察 Live Nodes 是否与 agent 数目相等（本文是 1）、Dead Nodes 是否为 0，判断集群是否启动。

Configured Capacity:	47.98 GB
Configured Remote Capacity:	0 B
DFS Used:	24 KB (0%)
Non DFS Used:	21.68 GB
DFS Remaining:	23.94 GB (49.89%)
Block Pool Used:	24 KB (0%)
DataNodes usages% (Min/Median/Max/stdDev):	0.00% / 0.00% / 0.00% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)
Dead Nodes	0 (Decommissioned: 0, In Maintenance: 0)
Decommissioning Nodes	0
Entering Maintenance Nodes	0
Total Datanode Volume Failures	0 (0 B)
Number of Under-Replicated Blocks	0
Number of Blocks Pending Deletion (including replicas)	0
Block Deletion Start Time	Fri Aug 27 11:07:18 +0800 2021
Last Checkpoint Time	Fri Aug 27 11:07:18 +0800 2021
Enabled Erasure Coding Policies	RS-6-3-1024k

2.5 部署 Hive

2.5.1 配置 MySQL

步骤 1 登录 mysql。

确保 MySQL 已在 2.1 完成安装，登录 mysql。（刷新 ssh 远程登录页面后，主机名已经显示为 server1）

```
[root@server1 home]# mysql -uroot -p -S /data/mysql/run/mysql.sock
```

或者，

```
/usr/local/mysql/bin/mysql -uroot -p -S /data/mysql/run/mysql.sock
```

步骤 2 选择 mysql 数据库。

```
mysql> use mysql;  
mysql> select Host,user from user;
```

```
mysql> use mysql;  
Database changed  
mysql> select Host,user from user;  
+-----+-----+  
| Host      | user      |  
+-----+-----+  
| %         | root      |  
| localhost | mysql.infoschema |  
| localhost | mysql.session |  
| localhost | mysql.sys   |  
| localhost | root       |  
+-----+-----+  
5 rows in set (0.00 sec)
```

步骤 3 创建 hive 用户。

在 mysql 中创建 hive 用户，并为 hive 用户赋予全部权限。

```
mysql> create user 'hive'@'server1' identified by '123456';  
mysql> grant all privileges on *.* to 'hive'@'server1' with grant option;
```

步骤 4 刷新授权。

刷新刚添加的权限，完成后退出。

```
mysql> flush privileges;  
mysql> exit;
```

```
mysql> create user 'hive'@'server1' identified by '123456';
Query OK, 0 rows affected (0.01 sec)

mysql> grant all privileges on *.* to 'hive'@'server1' with grant option;
Query OK, 0 rows affected (0.00 sec)

mysql> flush privileges;
Query OK, 0 rows affected (0.01 sec)

mysql> exit;
Bye
```

步骤 5 打开 my.cnf 配置文件。

```
[root@server1 home]# vim /etc/my.cnf
```

在[mysqld]标签下添加以下内容：

```
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
skip-character-set-client-handshake
```

```
[mysqld_safe]
log-error=/data/mysql/log/mysql.log
pid-file=/data/mysql/run/mysqld.pid
[mysqldump]
quick
[mysql]
no-auto-rehash
[client]
default-character-set=utf8
[mysqld] ←
basedir=/usr/local/mysql
socket=/data/mysql/run/mysql.sock
tmpdir=/data/mysql/tmp
datadir=/data/mysql/data
default_authentication_plugin=mysql_native_password
port=3306
user=mysql
init_connect='SET collation_connection = utf8_unicode_ci'
init_connect='SET NAMES utf8'
character-set-server=utf8
collation-server=utf8_unicode_ci
skip-character-set-client-handshake
```

步骤 6 重启 MySQL。

全部配置完成后，需要重启 MySQL。

```
[root@server1 home]# service mysql restart
```

```
[root@server1 ~]# service mysql restart
Shutting down MySQL.. SUCCESS!
Starting MySQL. SUCCESS!
```

如果出现 ERROR! MySQL server PID file could not be found!的报错，且无法启动，则需要 kill 掉 mysql 进程再重启。

```
[root@server1 ~]# service restart mysql
The service command supports only basic LSB actions (start, stop, restart, try-restart, reload, force-reload, status). For other actions, please try to use systemctl.
[root@server1 ~]# service mysql restart
ERROR! MySQL server PID file could not be found!
Starting MySQL..................................................................... ERROR! The
server quit without updating PID file (/data/mysql/data/server1.pid).
```

```
[root@server1 home]# ps -ef|grep mysql
```

```
[root@server1 home]# kill -9 pid
```

```
[root@server1 ~]# ps -ef|grep mysql
mysql      7754      1  0 Jan19 ?        00:00:00 /bin/sh /usr/local/mysql/bin/mysqld_safe --datadir=/data/mysql/data --pid-
file=/data/mysql/data/ecs-3ddf.pid
mysql      7969     7754  0 Jan19 ?        00:01:20 /usr/local/mysql/bin/mysqld --basedir=/usr/local/mysql --datadir=/data/mys
ql/data --plugin-dir=/usr/local/mysql/lib/plugin --log-error=/data/mysql/log/mysql.log --pid-file=/data/mysql/data/ecs-3ddf.pi
d --socket=/data/mysql/run/mysql.sock --port=3306
root      28621    27412  0 14:11 pts/2    00:00:00 grep --color=auto mysql
[root@server1 ~]# kill -9 7754
[root@server1 ~]# kill -9 7969
```

重启 MySQL 服务。

```
[root@server1 home]# service mysql restart
```

2.5.2 安装 Hive

步骤 1 下载 Hive 和 JDBC。

下载 Hive 的软件包。

```
[root@server1 home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/apache-
hive-3.1.2-bin.tar.gz
```

下载 JDBC 的软件包。

```
[root@server1 home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/mysql-
connector-java-8.0.26.zip
```

步骤 2 将 hive 添加到环境变量。

打开配置文件。

```
[root@server1 home]# vim /etc/profile
```

添加 Hive 到环境变量。

```
export HIVE_HOME=/usr/local/hive
export PATH=$HIVE_HOME/bin:$PATH
```

```
export MYSQL_HOME=/usr/local/mysql
export PATH=$MYSQL_HOME/bin:$PATH

export JAVA_HOME=/usr/local/jdk8u252-b09
export PATH=$JAVA_HOME/bin:$PATH

export HADOOP_HOME=/usr/local/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH

export HIVE_HOME=/usr/local/hive
export PATH=$HIVE_HOME/bin:$PATH

:wq
```

环境变量生效。

```
[root@server1 home]# source /etc/profile
```


步骤 3 执行 auto_hive.sh 脚本。

下载脚本。

```
[root@server1 home]# wget https://kp-timely-communication-platform.obs.cn-north-4.myhuaweicloud.com/auto_hive.sh
```

执行 auto_hive.sh。

```
[root@server1 home]# sh auto_hive.sh
```

步骤 4 初始化 Hive。

```
[root@server1 home]# schematool -dbType mysql -initSchema
```

```
Initialization script completed
schemaTool completed
[root@server1 mysql-connector-java-8.0.26]#
```

2.5.3 启动 hive

步骤 1 启动 hive metastore。

输入完命令后，出现提示信息，然后按回车继续。

```
[root@server1 home]# hive --service metastore -p 9083 &
```

```
Initialization script completed
schemaTool completed
[root@ecs-kp home]# hive --service metastore -p 9083 &
[2] 15990
[root@ecs-kp home]# 2023-07-24 15:24:04: Starting Hive Metastore Server
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
```

步骤 2 启动 hiveserver2。

```
[root@server1 home]# nohup hiveserver2 1>/usr/local/hive/log/hiveserver.log 2>/usr/local/hive/log/hiveserver.err &
```

步骤 3 查看启动进度。

```
[root@server1 home]# tail -f /usr/local/hive/log/hiveserver.err
```

步骤 4 等待启动完成。

```
[root@server1 mysql-connector-java-8.0.26]# nohup hiveserver2 1>/usr/local/hive/log/hiveserver.log 2>/usr/local/hive/log/hiveserver.err &
[2] 31445
[root@server1 mysql-connector-java-8.0.26]# tail -f /usr/local/hive/log/hiveserver.err
nohup: ignoring input
which: no hbase in (/usr/local/hive/bin:/usr/local/hadoop/bin:/usr/local/hadoop/sbin:/usr/local/jdk8u252-b09/bin:/usr/local/mysql/bin:/usr/
/local/hadoop/bin:/usr/local/hadoop/sbin:/usr/local/jdk8u252-b09/bin:/usr/local/mysql/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bi
n:/root/bin)
2022-01-20 15:27:23: Starting HiveServer2
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class
s]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Hive Session ID = b7c3cfba-32dd-4747-821d-d8432f0029ec
Hive Session ID = 0f9cf28f-c55a-49f6-b0d9-0ae29f3072fc
Hive Session ID = 098834d7-1839-45af-9601-92e2771842f7
Hive Session ID = effe7685-b8d6-4298-b521-66ace2e1db4c
```

输入 ctrl+c 可以退出 tail 模式，然后查看端口。(等到 4 个的时候在查看)

```
[root@server1 home]# netstat -an|grep 10000
```

如下所示即为启动成功。

```
[root@server1 mysql-connector-java-8.0.26]# netstat -an|grep 10000
tcp6      0      0 :::10000          :::*               LISTEN      31445/java
```

2.5.4 连接测试 hive

步骤 1 在 server1 使用 beeline 连接。

```
[root@server1 home]# beeline -u jdbc:hive2://server1:10000
```

```
[root@server1 mysql-connector-java-8.0.26]# beeline -u jdbc:hive2://server1:10000
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/apache-hive-3.1.2-bin/lib/log4j-slf4j-impl-2.10.0.jar!/org/slf4j/impl/StaticLoggerBinder.class
s]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-3.3.1/share/hadoop/common/lib/slf4j-log4j12-1.7.30.jar!/org/slf4j/impl/StaticLoggerBin
der.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://server1:10000
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.2 by Apache Hive
0: jdbc:hive2://server1:10000>
```

步骤 2 查看已创建的数据库。

```
0: jdbc:hive2://server1:10000> show databases;
```

验证功能如下即成功。

```
0: jdbc:hive2://server1:10000> show databases;
INFO : Compiling command(queryId=root_20220120153444_249166d6-2879-4ee7-ab59-bddb872b6133): show databases
INFO : Concurrent mode is disabled, not creating a lock manager
INFO : Semantic Analysis Completed (retrial = false)
INFO : Returning Hive schema: Schema(fieldSchemas:[FieldSchema(name:database_name, type:string, comment:from deserializer)], properties:null)
INFO : Completed compiling command(queryId=root_20220120153444_249166d6-2879-4ee7-ab59-bddb872b6133); Time taken: 0.928 seconds
INFO : Concurrent mode is disabled, not creating a lock manager
INFO : Executing command(queryId=root_20220120153444_249166d6-2879-4ee7-ab59-bddb872b6133): show databases
INFO : Starting task [Stage-0:DDL] in serial mode
INFO : Completed executing command(queryId=root_20220120153444_249166d6-2879-4ee7-ab59-bddb872b6133); Time taken: 0.054 seconds
INFO : OK
INFO : Concurrent mode is disabled, not creating a lock manager
+-----+
| database_name |
+-----+
| default       |
+-----+
1 row selected (1.323 seconds)
0: jdbc:hive2://server1:10000>
```

Hive 部署完成。

2.6 思考题

Hive 如何查看当前数据库下的所有表？以及如何查看某张表的表结构？

参考答案：

可以通过如下 HiveQL 语句查询当前数据库下的所有表

```
show tables;
```

可以通过如下 HiveQL 语句查询当前某张表的表结构

```
desc tablename;
```