大数据处理实验报告

实验三：MapReduce 的基本操作

姓　　名：	刘日星
学　　号：	X2020I1007
院　　系：	计算机科学与技术
专　　业：	计科与金融
年　　级：	CS1804-交换生
指导教师：	石宣化

2021 年 12 月 15 日

一：实验目的

1、了解 MapReduce 的用途

2、掌握 MapReduce 的基本命令

二：实验要求

1、第四节中的实验内容要附上完整的实验过程截图以及必要的文字说明，每个人的 IP 地址等不同，不能直接套用样例的截图。

2、请同学们在完成报告后，将报告的 pdf 版本命名为：大数据实验三+姓名+学号.pdf，并在这周五前,发到邮箱:

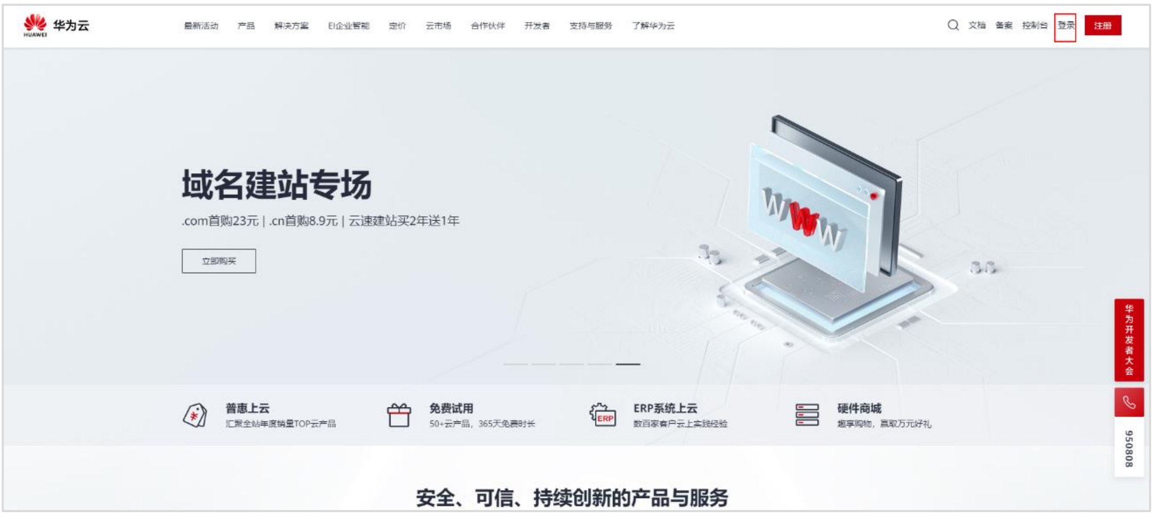
aaaaltaaaa@126.com(石老师班)

798792873@qq.com(郑老师班)

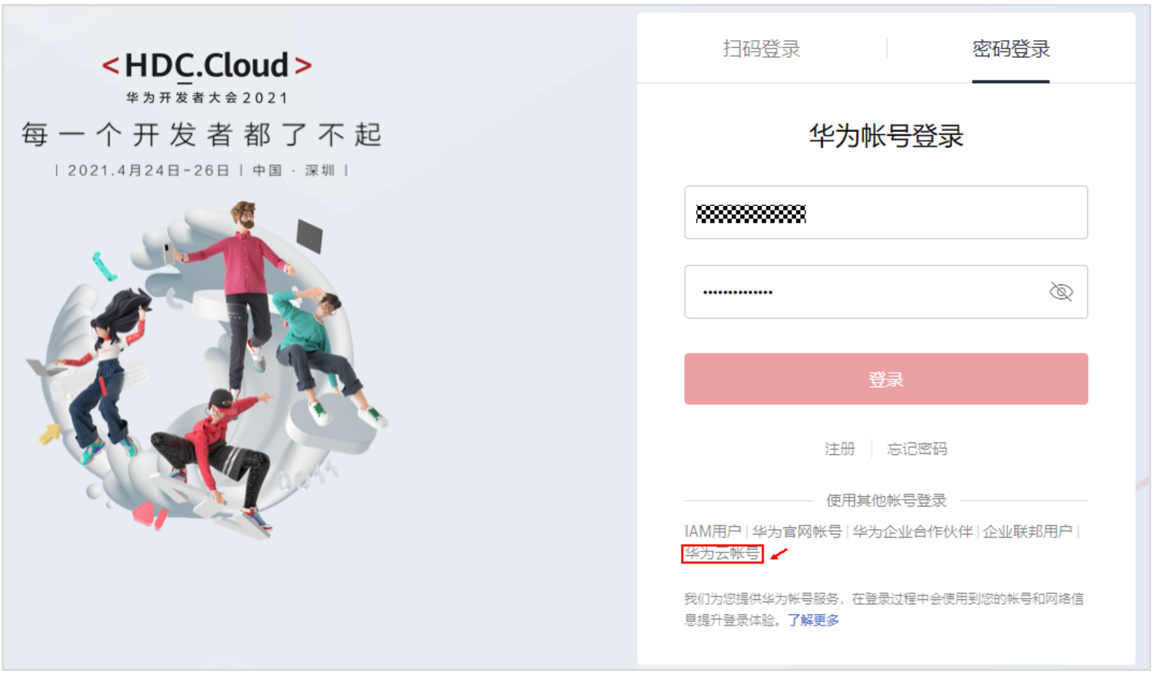
三：实验环境配置

步骤 1 登录华为云网站

https://www.huaweicloud.com



点击右上角登录，输入账号和密码



扫码登录 密码登录

华为帐号登录

[redacted]

··············

登录

注册 | 忘记密码

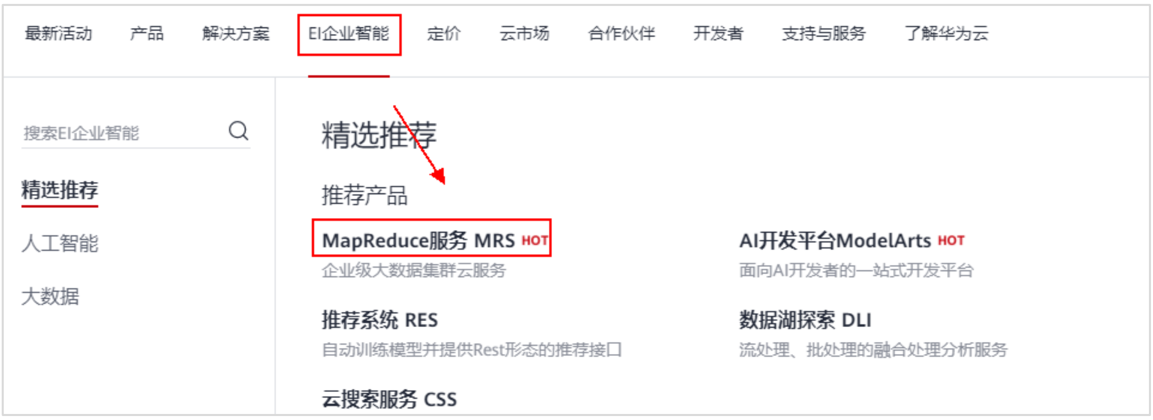
使用其他帐号登录

IAM用户 | 华为官网帐号 | 华为企业合作伙伴 | 企业联邦用户 | 华为云帐号

我们为您提供华为帐号服务，在登录过程中会使用到您的帐号和网络信息提升登录体验。了解更多

注意：华为云已统一登录入口，若仍不能登录则点击下方华为云账号进行登录。

步骤 2 点击“EI 企业智能”选择“MapReduce 服务”


最新活动 产品 解决方案 EI企业智能 定价 云市场 合作伙伴 开发者 支持与服务 了解华为云

搜索EI企业智能

精选推荐
人工智能
大数据

精选推荐

推荐产品

MapReduce服务 MRS HOT
企业级大数据集群云服务

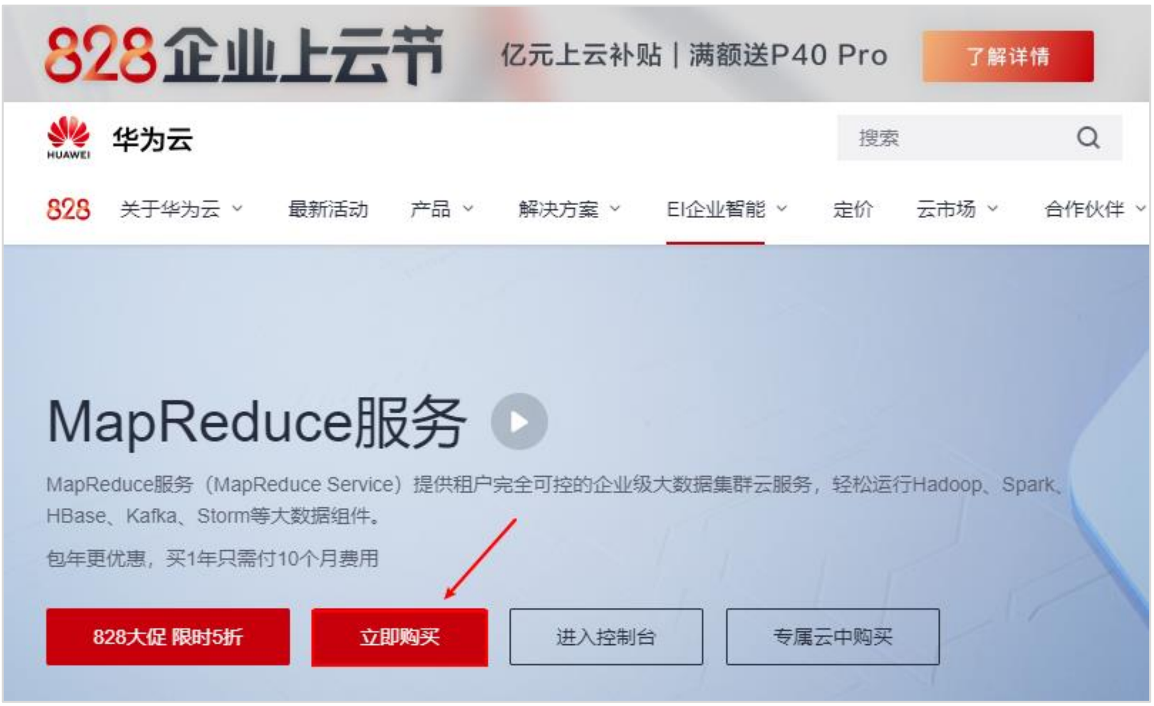
AI开发平台ModelArts HOT
面向AI开发者的一站式开发平台

推荐系统 RES
自动训练模型并提供Rest形态的推荐接口

数据湖探索 DLI
流处理、批处理的融合处理分析服务

云搜索服务 CSS

步骤 3 点击“立即购买”

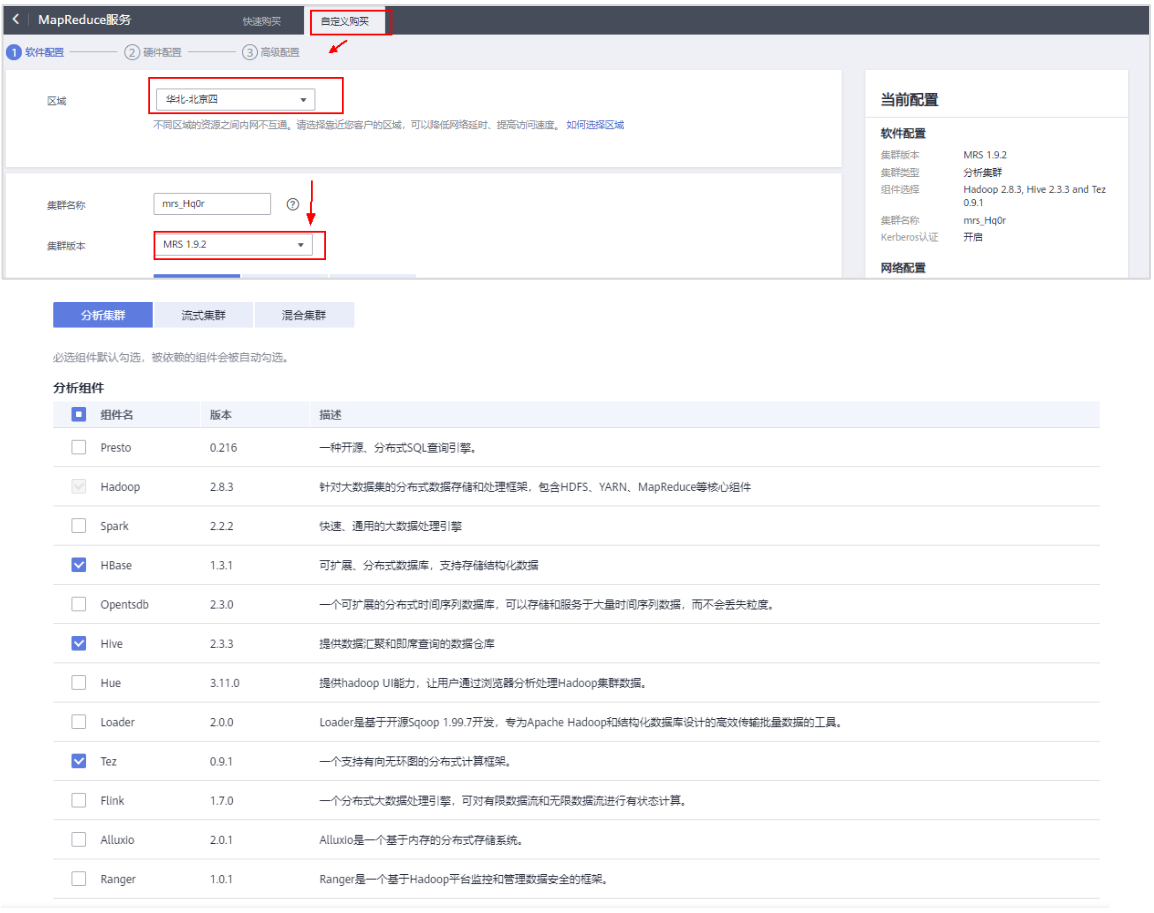


MapReduce服务

MapReduce服务（MapReduce Service）提供租户完全可控的企业级大数据集群云服务，轻松运行Hadoop、Spark、HBase、Kafka、Storm等大数据组件。

包年更优惠，买1年只需付10个月费用

选择“自定义购买”(这里还要选上 Spark)



区域	华北-北京四
集群名称	mrs_Hq0r
集群版本	MRS 1.9.2

不同区域的资源之间内网不互通。请选择靠近您客户的区域，可以降低网络延时、提高访问速度。 如何选择区域

当前配置

软件配置

集群版本	MRS 1.9.2
集群类型	分析集群
组件选择	Hadoop 2.8.3, Hive 2.3.3 and Tez 0.9.1
集群名称	mrs_Hq0r
Kerberos认证	开启

网络配置

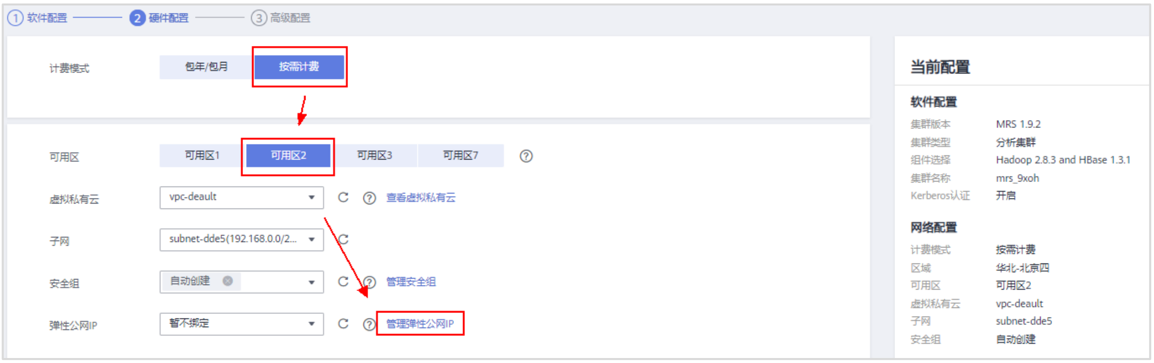
必选组件默认勾选，被依赖的组件会被自动勾选。

分析组件

	组件名	版本	描述
☐	Presto	0.216	一种开源、分布式SQL查询引擎。
✓	Hadoop	2.8.3	针对大数据集的分布式数据存储和处理框架，包含HDFS、YARN、MapReduce等核心组件
☐	Spark	2.2.2	快速、通用的大数据处理引擎
✓	HBase	1.3.1	可扩展、分布式数据库，支持存储结构化数据
☐	Opentsdb	2.3.0	一个可扩展的分布式时间序列数据库，可以存储和服务于大量时间序列数据，而不会丢失粒度。
✓	Hive	2.3.3	提供数据汇聚和即席查询的数据仓库
☐	Hue	3.11.0	提供hadoop UI能力，让用户通过浏览器分析处理Hadoop集群数据。
☐	Loader	2.0.0	Loader是基于开源Sqoop 1.99.7开发，专为Apache Hadoop和结构化数据库设计的高效传输批量数据的工具。
✓	Tez	0.9.1	一个支持有向无环图的分布式计算框架。
☐	Flink	1.7.0	一个分布式大数据处理引擎，可对有限数据流和无限数据流进行有状态计算。
☐	Alluxio	2.0.1	Alluxio是一个基于内存的分布式存储系统。
☐	Ranger	1.0.1	Ranger是一个基于Hadoop平台监控和管理数据安全的框架。

点击下一步，进入硬件配置

选择“按需计费”，“可用区 2”，点击“弹性公网 IP”，如下图：



计费模式	包年/包月 [✓ 按需计费]
可用区	可用区1 [✓ 可用区2] 可用区3 可用区7
虚拟私有云	vpc-deault 查看虚拟私有云
子网	subnet-dde5(192.168.0.0/2...
安全组	自动创建 管理安全组
弹性公网IP	暂不绑定 管理弹性公网IP

当前配置

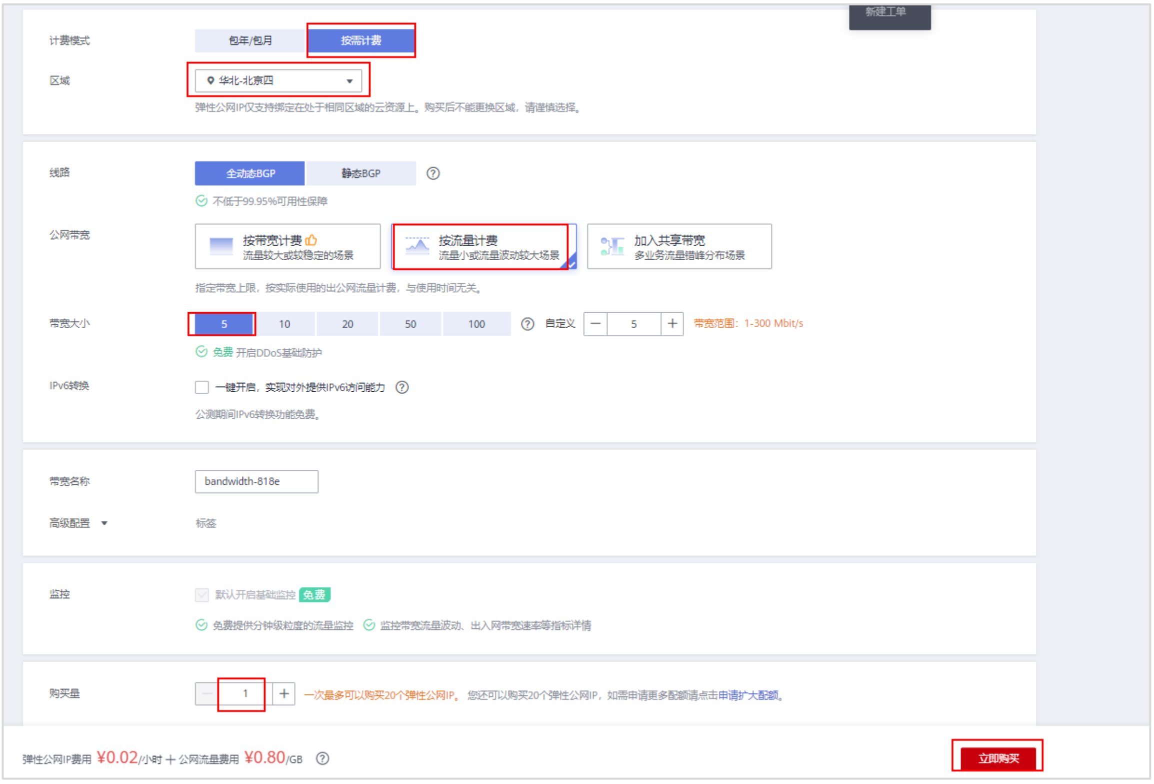
软件配置

集群版本	MRS 1.9.2
集群类型	分析集群
组件选择	Hadoop 2.8.3 and HBase 1.3.1
集群名称	mrs_9xoh
Kerberos认证	开启

网络配置

计费模式	按需计费
区域	华北-北京四
可用区	可用区2
虚拟私有云	vpc-deault
子网	subnet-dde5
安全组	自动创建

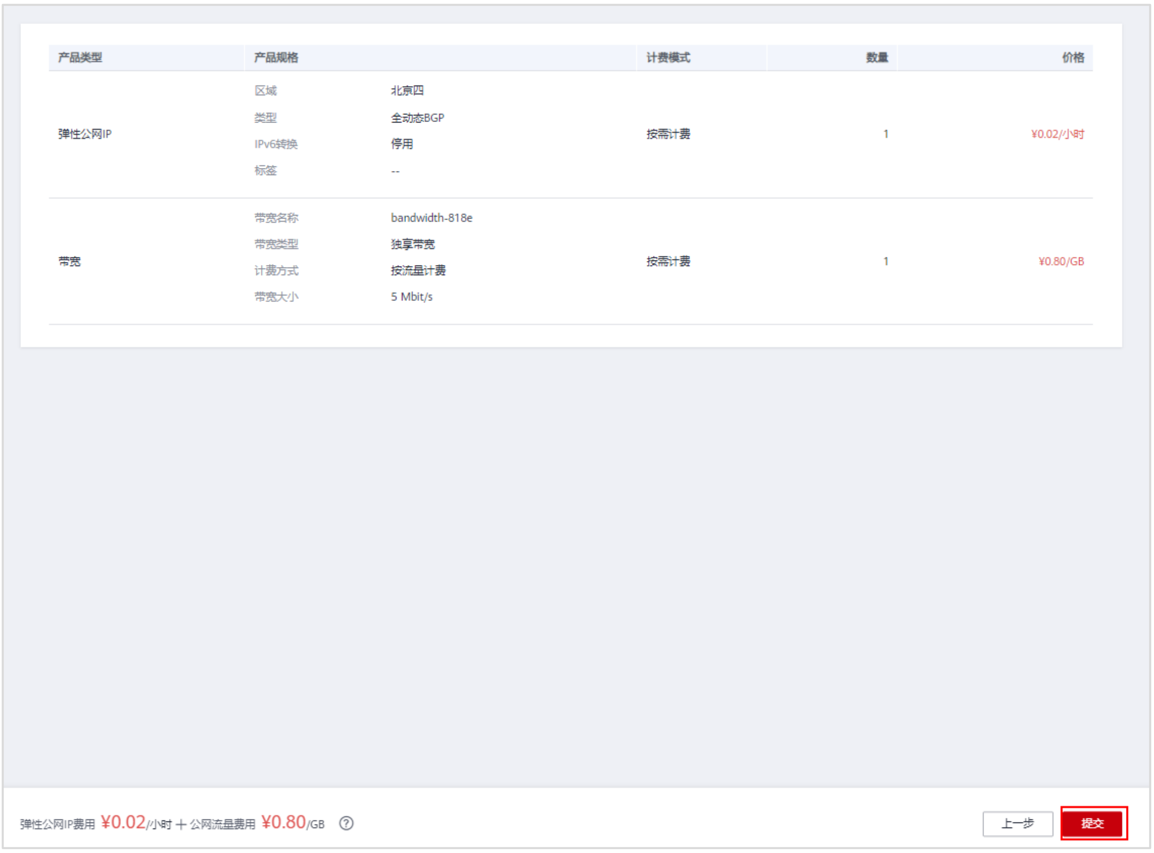
点击“购买弹性公网 IP”，选择“按需计费”，“按流量计费”，“5M”,点击“立即购买”，如下图：



计费模式	包年/包月 [✓ 按需计费]
区域	华北-北京四
	弹性公网IP仅支持绑定在处于相同区域的云资源上。购买后不能更换区域，请谨慎选择。
线路	[✓ 全动态BGP] 静态BGP
	不低于99.95%可用性保障
公网带宽	按带宽计费 流量较大或较稳定的场景 / [✓ 按流量计费 流量小或流量波动较大场景] / 加入共享带宽 多业务流量错峰分布场景
	指定带宽上限，按实际使用的出公网流量计费，与使用时间无关。
带宽大小	[✓ 5] 10 20 50 100 自定义 5 带宽范围：1-300 Mbit/s
	免费 开启DDoS基础防护
IPv6转换	一键开启，实现对外提供IPv6访问能力
	公测期间IPv6转换功能免费。
带宽名称	bandwidth-818e
高级配置	标签
监控	默认开启基础监控 免费
	免费提供分钟级粒度的流量监控 监控带宽流量波动、出入网带宽速率等指标详情
购买量	1 一次最多可以购买20个弹性公网IP。 您还可以购买20个弹性公网IP，如需申请更多配额请点击申请扩大配额。

弹性公网IP费用 ¥0.02/小时 + 公网流量费用 ¥0.80/GB 立即购买

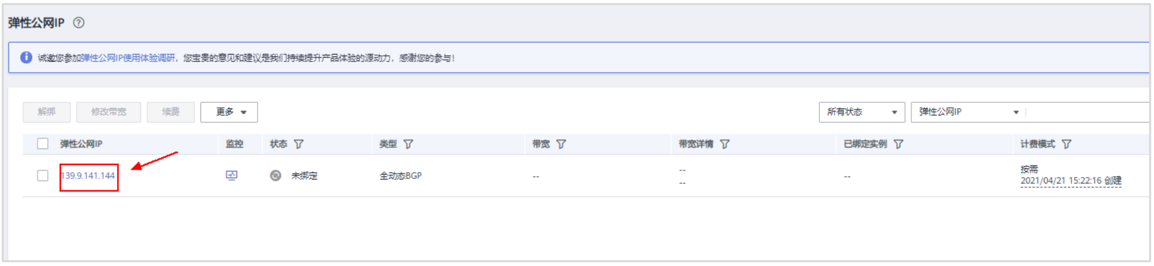
点击“提交”，如下图：



产品类型	产品规格		计费模式	数量	价格
弹性公网IP	区域	北京四	按需计费	1	¥0.02/小时
	类型	全动态BGP			
	IPv6转换	停用			
	标签	--			
带宽	带宽名称	bandwidth-818e	按需计费	1	¥0.80/GB
	带宽类型	独享带宽			
	计费方式	按流量计费			
	带宽大小	5 Mbit/s			

弹性公网IP费用 ¥0.02/小时 + 公网流量费用 ¥0.80/GB

购买成功，如下图：

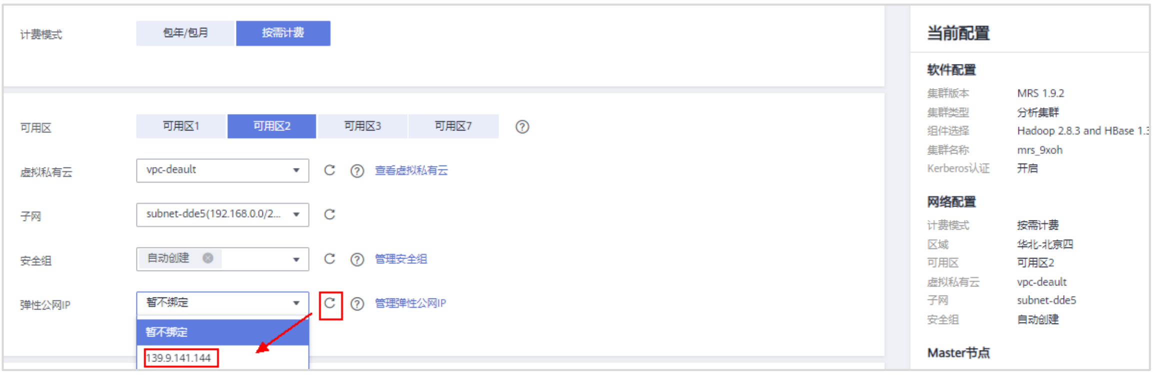


弹性公网IP

诚邀您参加弹性公网IP使用体验调研，您宝贵的意见和建议是我们持续提升产品体验的源动力，感谢您的参与！

弹性公网IP	监控	状态	类型	带宽	带宽详情	已绑定实例	计费模式
139.9.141.144		未绑定	全动态BGP	--	-- --	--	按需 2021/04/21 15:22:16 创建

返回 MapReduce 服务自定义购买界面绑定 EIP



计费模式	包年/包月 [✓ 按需计费]
可用区	可用区1 [✓ 可用区2] 可用区3 可用区7
虚拟私有云	vpc-deault
子网	subnet-dde5(192.168.0.0/2...
安全组	自动创建
弹性公网IP	暂不绑定

当前配置

软件配置

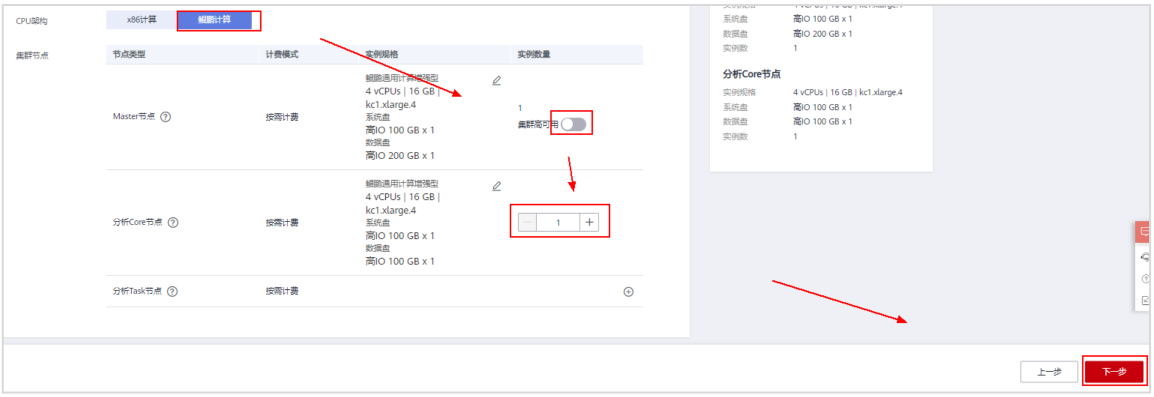
集群版本	MRS 1.9.2
集群类型	分析集群
组件选择	Hadoop 2.8.3 and HBase 1.3
集群名称	mrs_9xoh
Kerberos认证	开启

网络配置

计费模式	按需计费
区域	华北-北京四
可用区	可用区2
虚拟私有云	vpc-deault
子网	subnet-dde5
安全组	自动创建

Master节点

选择“鲲鹏计算”，关闭高可用，调整 core 节点数为 1,如下图：



CPU架构：x86计算 | [✓ 鲲鹏计算]

集群节点

节点类型	计费模式	实例规格	实例数量
Master节点	按需计费	鲲鹏通用计算增强型 4 vCPUs | 16 GB | kc1.xlarge.4 系统盘 高IO 100 GB x 1 数据盘 高IO 200 GB x 1	1 集群高可用
分析Core节点	按需计费	鲲鹏通用计算增强型 4 vCPUs | 16 GB | kc1.xlarge.4 系统盘 高IO 100 GB x 1 数据盘 高IO 100 GB x 1	1
分析Task节点	按需计费		

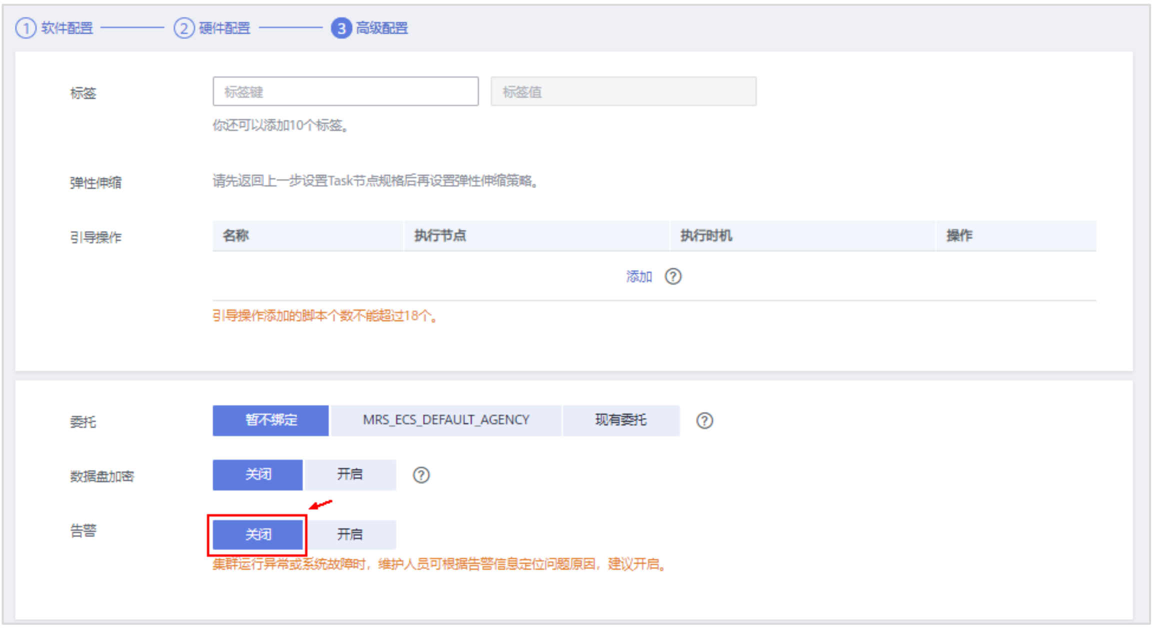
系统盘	高IO 100 GB x 1
数据盘	高IO 200 GB x 1
实例数	1

分析Core节点

实例规格	4 vCPUs | 16 GB | kc1.xlarge.4
系统盘	高IO 100 GB x 1
数据盘	高IO 100 GB x 1
实例数	1

点击“下一步”

高级配置项参考如下：



标签：标签键 标签值
你还可以添加10个标签。

弹性伸缩：请先返回上一步设置Task节点规格后再设置弹性伸缩策略。

引导操作

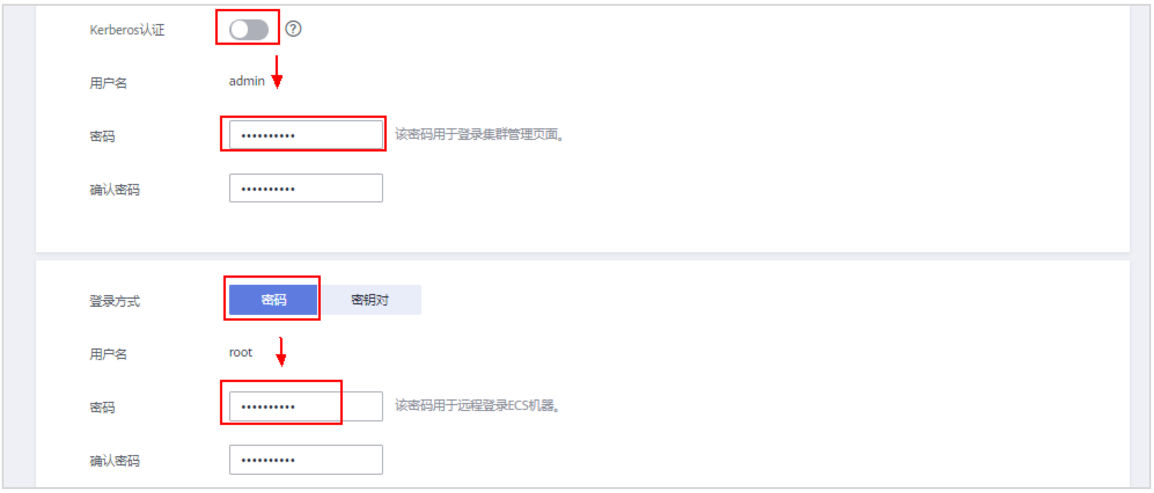
名称	执行节点	执行时机	操作

添加
引导操作添加的脚本个数不能超过18个。

委托：[✓ 暂不绑定] | MRS_ECS_DEFAULT_AGENCY | 现有委托

数据盘加密：[✓ 关闭] | 开启

告警：[✓ 关闭] | 开启
集群运行异常或系统故障时，维护人员可根据告警信息定位问题原因，建议开启。



Kerberos认证

用户名：admin

密码：•••••••••• 该密码用于登录集群管理页面。

确认密码：••••••••••

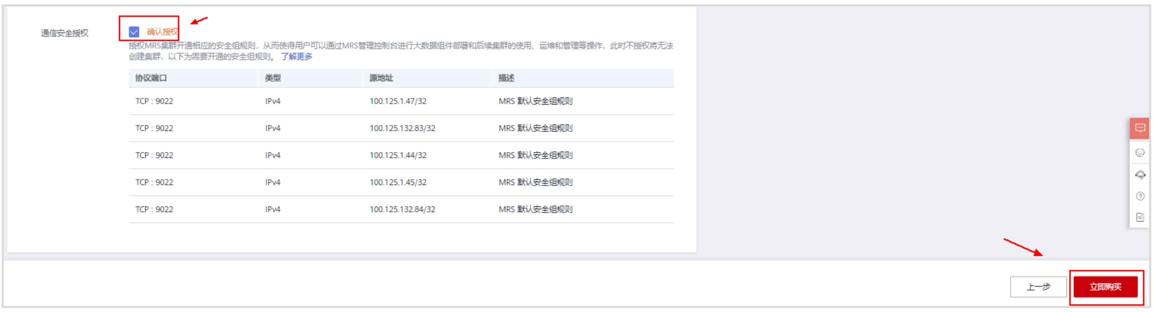
登录方式：[✓ 密码] | 密钥对

用户名：root

密码：•••••••••• 该密码用于远程登录ECS机器。

确认密码：••••••••••

点击“确认授权”

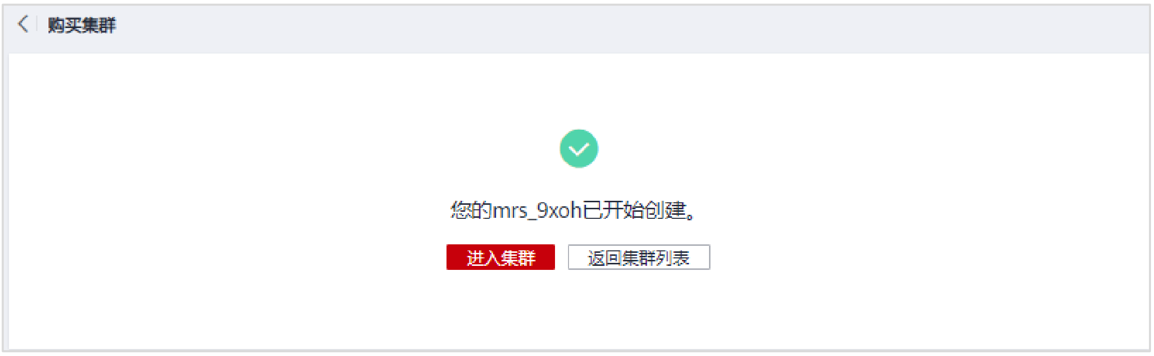

通信安全授权 ☑ 确认授权

授权MRS集群开通相应的安全组规则，从而使得用户可以通过MRS管理控制台进行大数据组件部署和后续集群的使用、运维和管理等操作，此时不授权将无法创建集群，以下为需要开通的安全组规则。 了解更多

协议端口	类型	源地址	描述
TCP : 9022	IPv4	100.125.1.47/32	MRS 默认安全组规则
TCP : 9022	IPv4	100.125.132.83/32	MRS 默认安全组规则
TCP : 9022	IPv4	100.125.1.44/32	MRS 默认安全组规则
TCP : 9022	IPv4	100.125.1.45/32	MRS 默认安全组规则
TCP : 9022	IPv4	100.125.132.84/32	MRS 默认安全组规则

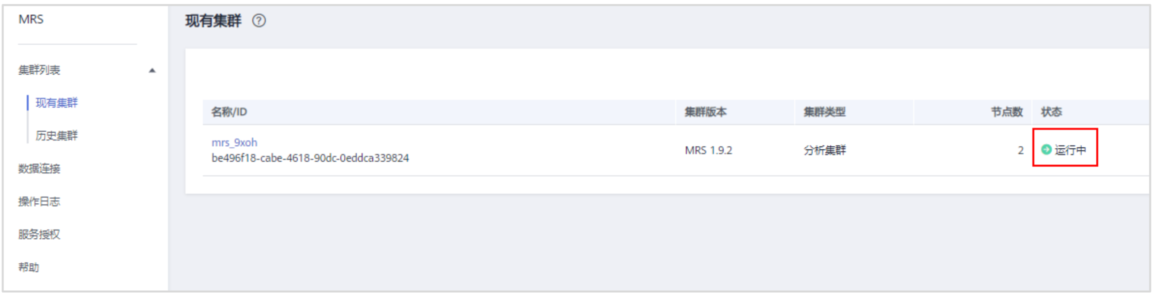
点击“立即购买”

步骤 4 点击“返回集群列表”，如下图：



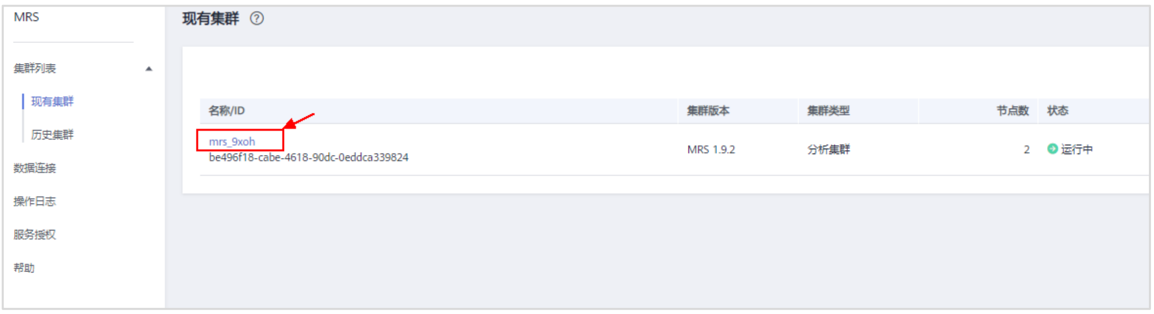
您的mrs_9xoh已开始创建。

创建过程需要等待几分钟，待状态变为“运行中”集群创建完成



名称/ID	集群版本	集群类型	节点数	状态
mrs_9xoh be496f18-cabe-4618-90dc-0eddca339824	MRS 1.9.2	分析集群	2	运行中

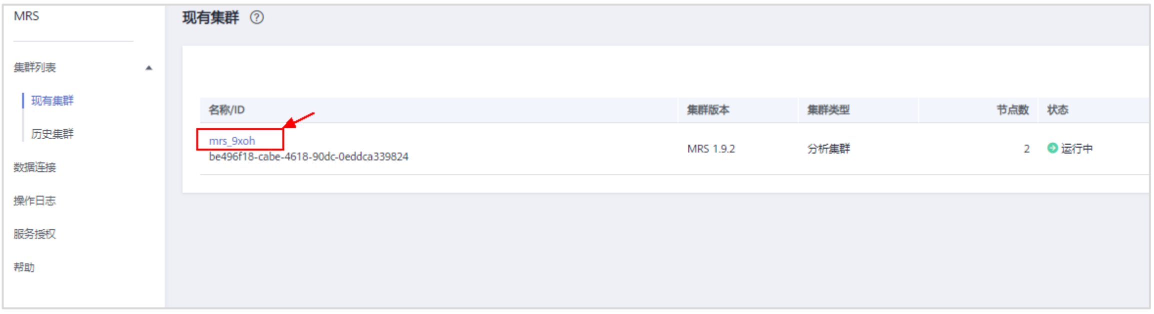
步骤 5 点击集群名称



名称/ID	集群版本	集群类型	节点数	状态
mrs_9xoh be496f18-cabe-4618-90dc-0eddca339824	MRS 1.9.2	分析集群	2	运行中

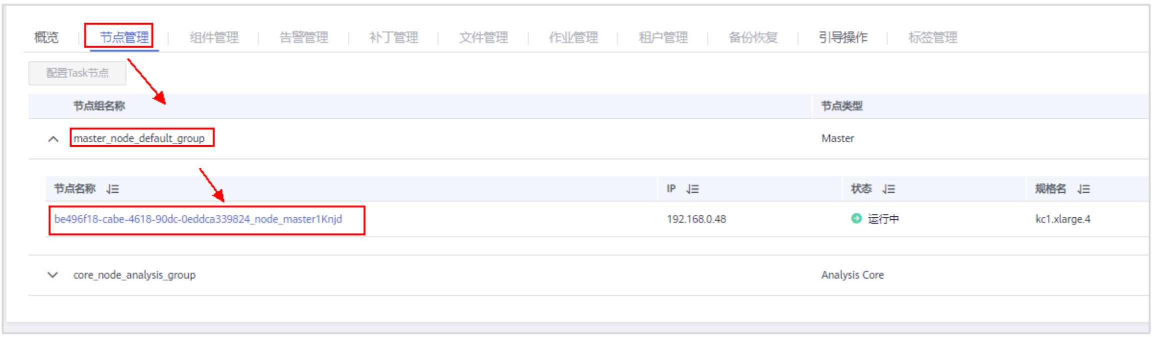
步骤 6 配置安全组

点击集群名称



名称/ID	集群版本	集群类型	节点数	状态
mrs_9xoh be496f18-cabe-4618-90dc-0eddca339824	MRS 1.9.2	分析集群	2	运行中

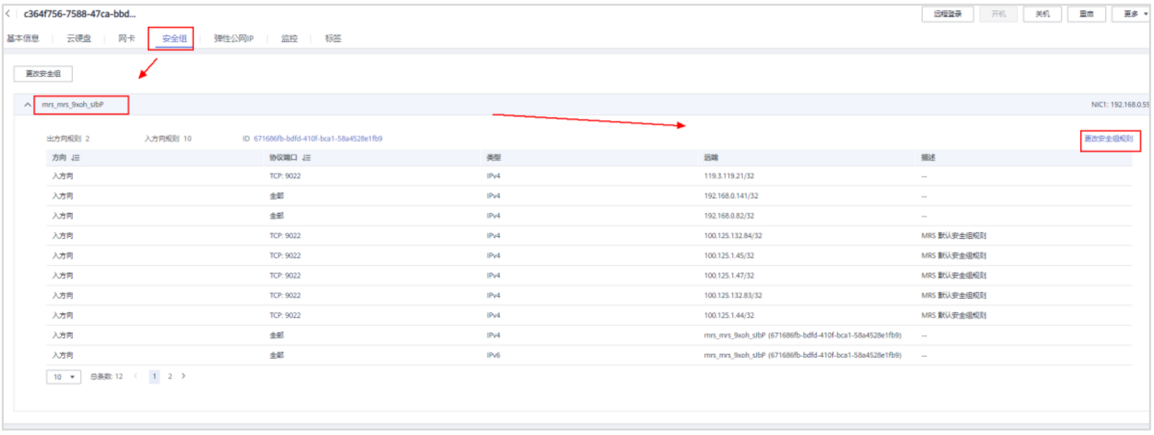
选择“节点管理”，点击含有“master1”的节点



节点组名称	节点类型
master_node_default_group	Master
core_node_analysis_group	Analysis Core

节点名称	IP	状态	规格名
be496f18-cabe-4618-90dc-0eddca339824_node_master1Knjd	192.168.0.48	运行中	kc1.xlarge.4

在弹出页面中选择“安全组”，点击“更改安全组规则”，如下图所示：

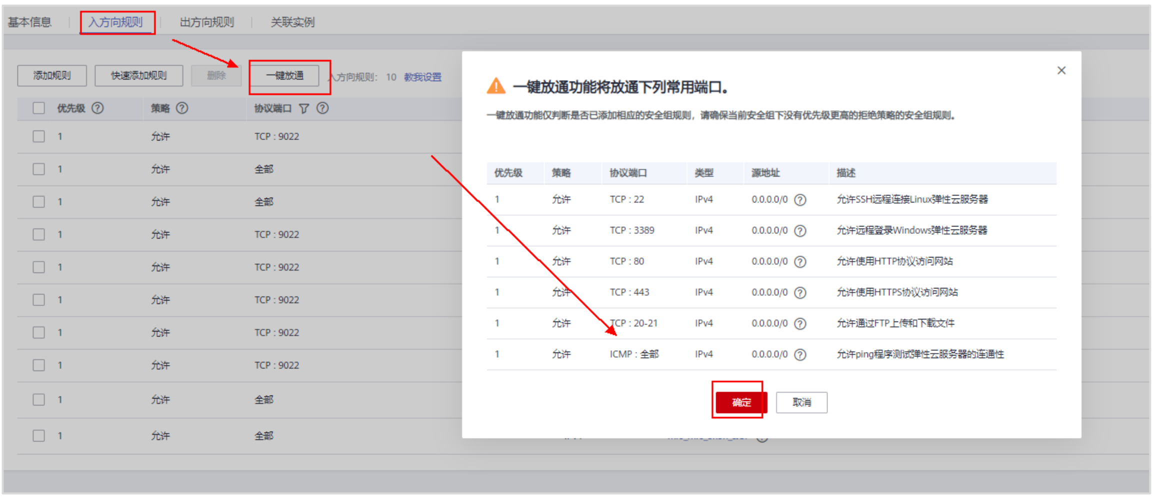


出方向规则 2　入方向规则 10　ID 671686fb-bdfd-410f-bca1-58a4528e1fb9

方向	协议端口	类型	远端	描述
入方向	TCP: 9022	IPv4	119.3.119.21/32	--
入方向	全部	IPv4	192.168.0.141/32	--
入方向	全部	IPv4	192.168.0.82/32	--
入方向	TCP: 9022	IPv4	100.125.132.84/32	MRS 默认安全组规则
入方向	TCP: 9022	IPv4	100.125.1.45/32	MRS 默认安全组规则
入方向	TCP: 9022	IPv4	100.125.1.47/32	MRS 默认安全组规则
入方向	TCP: 9022	IPv4	100.125.132.83/32	MRS 默认安全组规则
入方向	TCP: 9022	IPv4	100.125.1.44/32	MRS 默认安全组规则
入方向	全部	IPv4	mrs_mrs_9xoh_sIbP (671686fb-bdfd-410f-bca1-58a4528e1fb9)	--
入方向	全部	IPv6	mrs_mrs_9xoh_sIbP (671686fb-bdfd-410f-bca1-58a4528e1fb9)	--

10　总条数: 12　1　2

选择“入方向规则”，点击“一键放通”，确认即可。



基本信息 | 入方向规则 | 出方向规则 | 关联实例

添加规则　快速添加规则　删除　一键放通　入方向规则：10　教我设置

优先级	策略	协议端口
1	允许	TCP : 9022
1	允许	全部
1	允许	全部
1	允许	TCP : 9022
1	允许	TCP : 9022
1	允许	TCP : 9022
1	允许	TCP : 9022
1	允许	TCP : 9022
1	允许	全部
1	允许	全部

一键放通功能将放通下列常用端口。

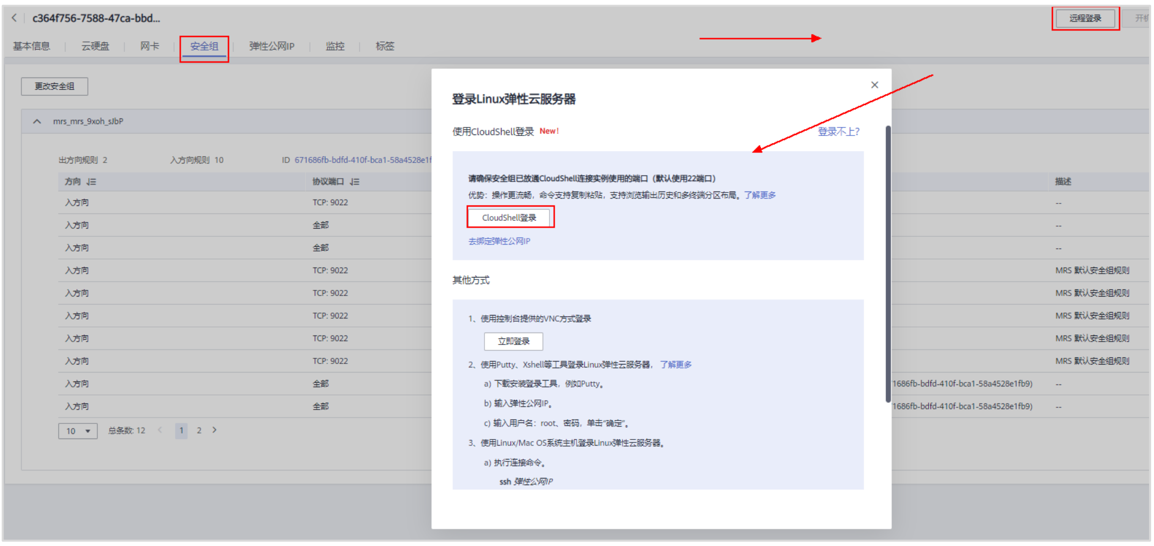
一键放通功能仅判断是否已添加相应的安全组规则，请确保当前安全组下没有优先级更高的拒绝策略的安全组规则。

优先级	策略	协议端口	类型	源地址	描述
1	允许	TCP : 22	IPv4	0.0.0.0/0	允许SSH远程连接Linux弹性云服务器
1	允许	TCP : 3389	IPv4	0.0.0.0/0	允许远程登录Windows弹性云服务器
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通性

确定　取消

步骤 7　远程登录 master 节点

在安全组配置项，点击右上方“远程登录”，选择 cloudshell 登录。



c364f756-7588-47ca-bbd...　远程登录

基本信息 | 云硬盘 | 网卡 | 安全组 | 弹性公网IP | 监控 | 标签

更改安全组

mrs_mrs_9xoh_sJbP

出方向规则 2　入方向规则 10　ID 671686fb-bdfd-410f-bca1-58a4528e1f

方向	协议端口	描述
入方向	TCP: 9022	--
入方向	全部	--
入方向	全部	--
入方向	TCP: 9022	MRS 默认安全组规则
入方向	TCP: 9022	MRS 默认安全组规则
入方向	TCP: 9022	MRS 默认安全组规则
入方向	TCP: 9022	MRS 默认安全组规则
入方向	TCP: 9022	MRS 默认安全组规则
入方向	全部	--
入方向	全部	--

10　总条数: 12　1　2

登录Linux弹性云服务器

使用CloudShell登录 New!　登录不上?

请确保安全组已放通CloudShell连接实例使用的端口（默认使用22端口）

优势：操作更流畅，命令支持复制粘贴，支持浏览输出历史和多终端分区布局。了解更多

CloudShell登录

去绑定弹性公网IP

其他方式

1、使用控制台提供的VNC方式登录

立即登录

2、使用Putty、Xshell等工具登录Linux弹性云服务器，了解更多

a) 下载安装登录工具，例如Putty。

b) 输入弹性公网IP。

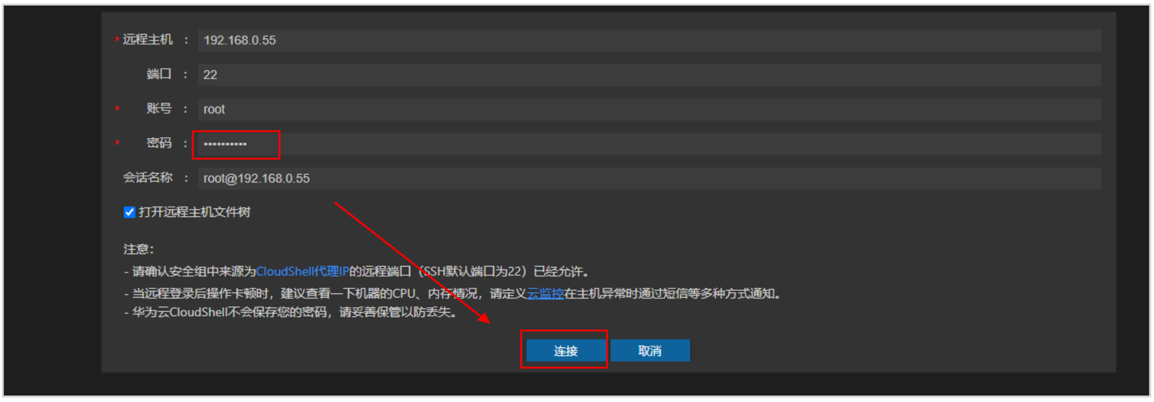
c) 输入用户名：root、密码，单击“确定”。

3、使用Linux/Mac OS系统主机登录Linux弹性云服务器。

a) 执行连接命令。

ssh *弹性公网IP*

输入密码，点击连接即可。

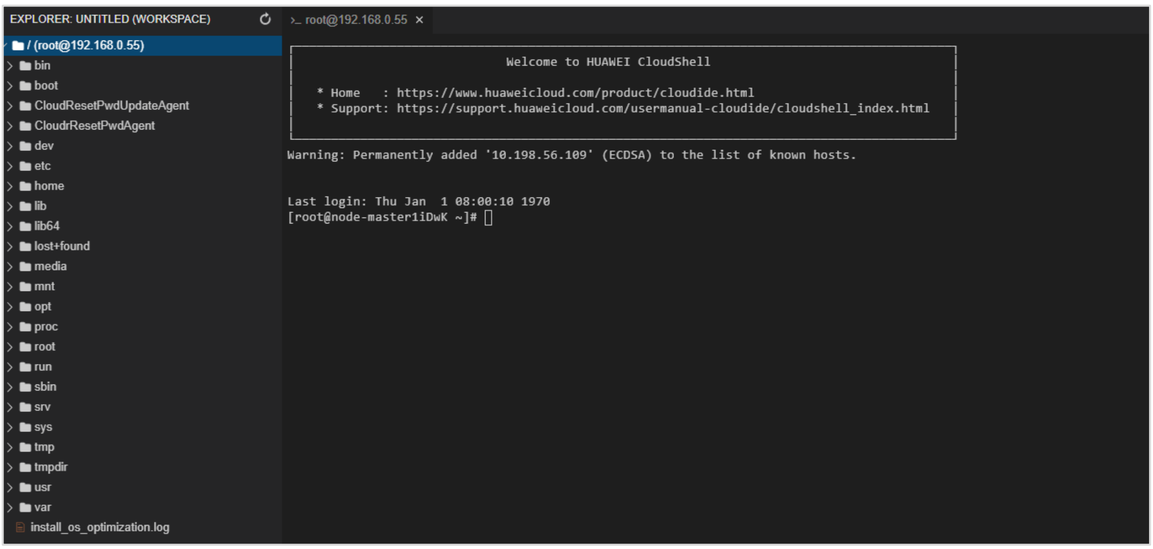


* 远程主机 ：	192.168.0.55
端口 ：	22
* 账号 ：	root
* 密码 ：	••••••••••
会话名称 ：	root@192.168.0.55

☑ 打开远程主机文件树

注意：
- 请确认安全组中来源为CloudShell代理IP的远程端口（SSH默认端口为22）已经允许。
- 当远程登录后操作卡顿时，建议查看一下机器的CPU、内存情况，请定义云监控在主机异常时通过短信等多种方式通知。
- 华为云CloudShell不会保存您的密码，请妥善保管以防丢失。

连接　取消



```
                    Welcome to HUAWEI CloudShell

   * Home    : https://www.huaweicloud.com/product/cloudide.html
   * Support: https://support.huaweicloud.com/usermanual-cloudide/cloudshell_index.html

Warning: Permanently added '10.198.56.109' (ECDSA) to the list of known hosts.


Last login: Thu Jan  1 08:00:10 1970
[root@node-master1iDwK ~]# 
```

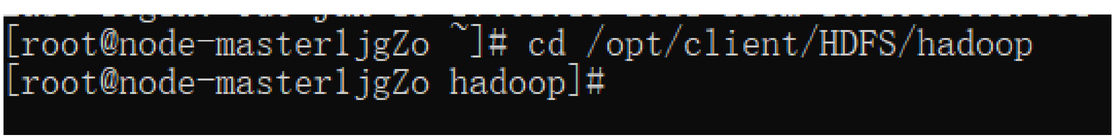
四：实验内容及步骤、实验的详细记录、实验结果分析

请附上实验过程截图（截图需包含指令）以及必要的文字分析

4.1 MapReduce

4.1.1 进入 hadoop(5’)

cd /opt/client/HDFS/Hadoop

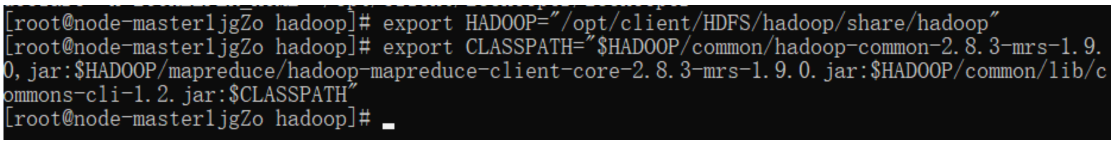


```
[root@node-master1jgZo ~]# cd /opt/client/HDFS/hadoop
[root@node-master1jgZo hadoop]#
```

4.1.2 添加环境变量(10’)

export HADOOP="/opt/client/HDFS/hadoop/share/hadoop"

export
CLASSPATH="$HADOOP/common/hadoop-common-2.8.3-mrs-1.9.0.jar:$HADOOP/mapreduce/hadoop-mapreduce-client-core-2.8.3-mrs-1.9.0.jar:$HADOOP/common/lib/commons-cli-1.2.jar:$CLASSPATH"



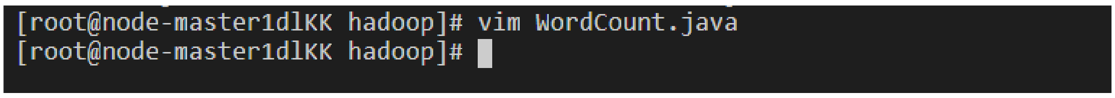
```
[root@node-master1jgZo hadoop]# export HADOOP="/opt/client/HDFS/hadoop/share/hadoop"
[root@node-master1jgZo hadoop]# export CLASSPATH="$HADOOP/common/hadoop-common-2.8.3-mrs-1.9.
0.jar:$HADOOP/mapreduce/hadoop-mapreduce-client-core-2.8.3-mrs-1.9.0.jar:$HADOOP/common/lib/c
ommons-cli-1.2.jar:$CLASSPATH"
[root@node-master1jgZo hadoop]#
```

配置 java 环境

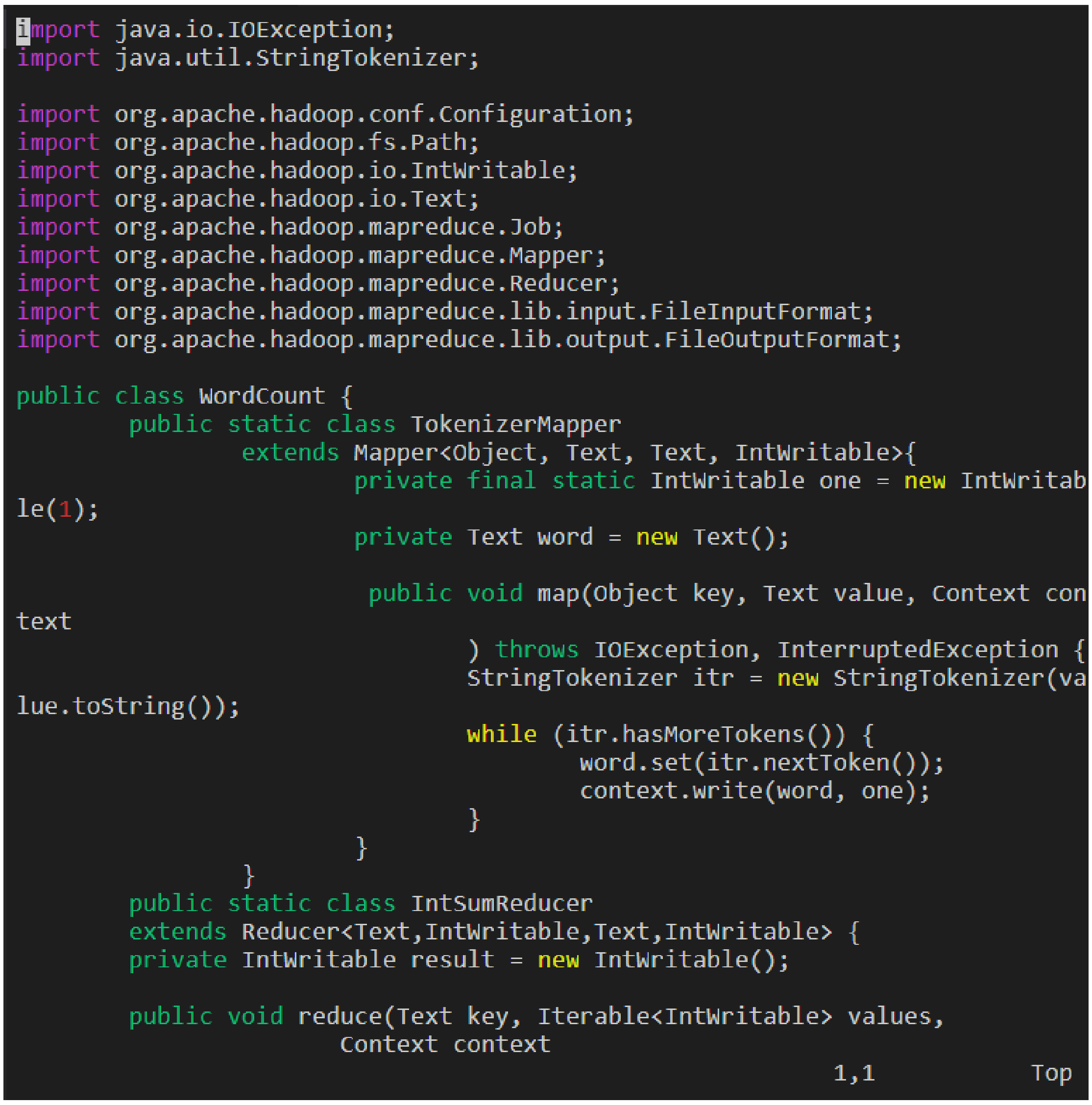
4.1.3 创建 java 程序 WordCount.java,在里面输入以下代码(5’)

```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {

  public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable>{

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context
                    ) throws IOException, InterruptedException {
```

```
        StringTokenizer itr = new StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
          word.set(itr.nextToken());
          context.write(word, one);
        }
      }
    }

    public static class IntSumReducer
         extends Reducer<Text,IntWritable,Text,IntWritable> {
      private IntWritable result = new IntWritable();

      public void reduce(Text key, Iterable<IntWritable> values,
                         Context context
                         ) throws IOException, InterruptedException {
        int sum = 0;
        for (IntWritable val : values) {
          sum += val.get();
        }
        result.set(sum);
        context.write(key, result);
      }
    }

    public static void main(String[] args) throws Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "word count");
      job.setJarByClass(WordCount.class);
      job.setMapperClass(TokenizerMapper.class);
      job.setCombinerClass(IntSumReducer.class);
      job.setReducerClass(IntSumReducer.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(IntWritable.class);
      FileInputFormat.addInputPath(job, new Path(args[0]));
      FileOutputFormat.setOutputPath(job, new Path(args[1]));
      System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
  }
```



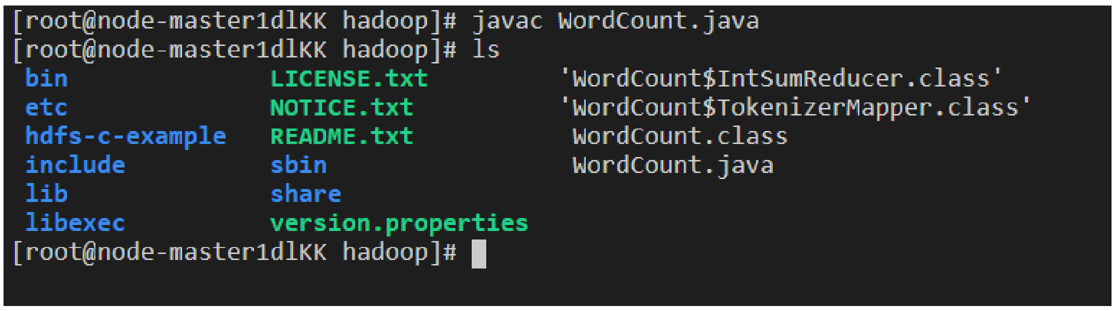
```
[root@node-master1dlKK hadoop]# vim WordCount.java
[root@node-master1dlKK hadoop]# 
```

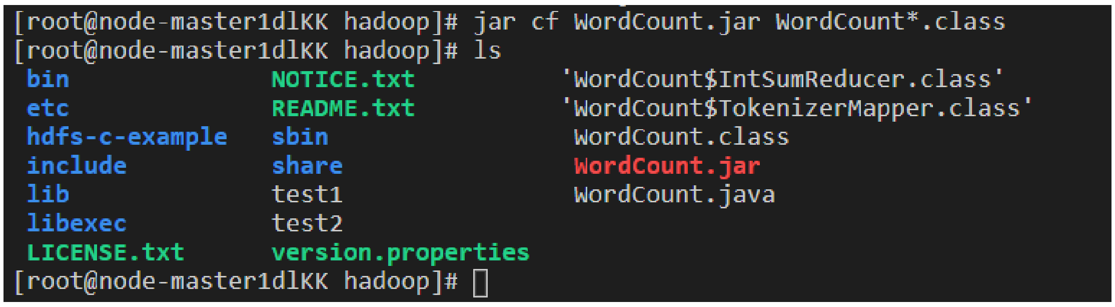


```
import java.io.IOException;
import java.util.StringTokenizer;

import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class WordCount {
        public static class TokenizerMapper
                extends Mapper<Object, Text, Text, IntWritable>{
                        private final static IntWritable one = new IntWritab
le(1);

                        private Text word = new Text();

                         public void map(Object key, Text value, Context con
text

                                ) throws IOException, InterruptedException {
                                StringTokenizer itr = new StringTokenizer(va
lue.toString());

                                while (itr.hasMoreTokens()) {
                                        word.set(itr.nextToken());
                                        context.write(word, one);
                                }
                        }
                }
        public static class IntSumReducer
        extends Reducer<Text,IntWritable,Text,IntWritable> {
        private IntWritable result = new IntWritable();

        public void reduce(Text key, Iterable<IntWritable> values,
                       Context context
                                                      1,1           Top
```

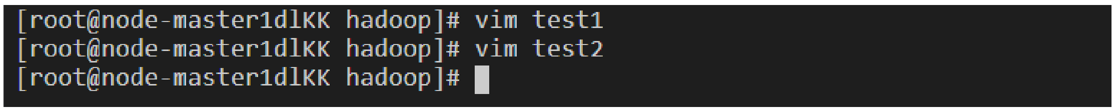
编写 java 文件

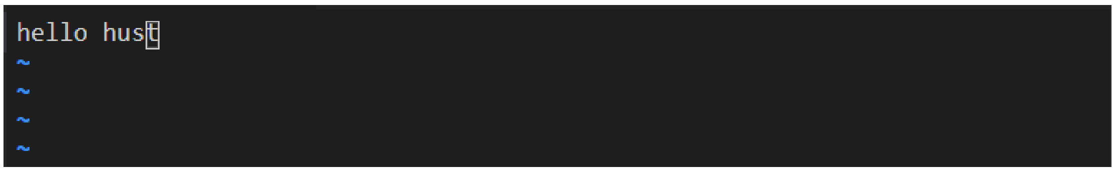
4.1.4 编译 WordCount.java(5’)

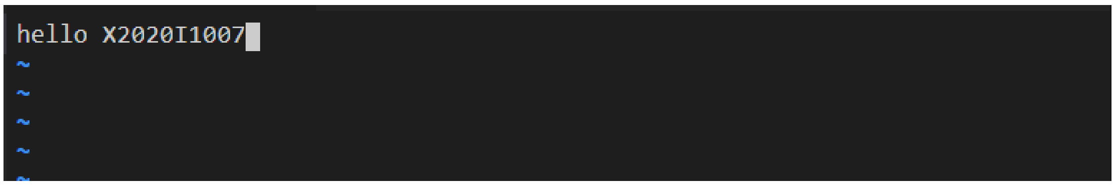

```
[root@node-master1dlKK hadoop]# javac WordCount.java
[root@node-master1dlKK hadoop]# ls
 bin                LICENSE.txt          'WordCount$IntSumReducer.class'
 etc                NOTICE.txt           'WordCount$TokenizerMapper.class'
 hdfs-c-example     README.txt            WordCount.class
 include            sbin                  WordCount.java
 lib                share
 libexec            version.properties
[root@node-master1dlKK hadoop]#
```

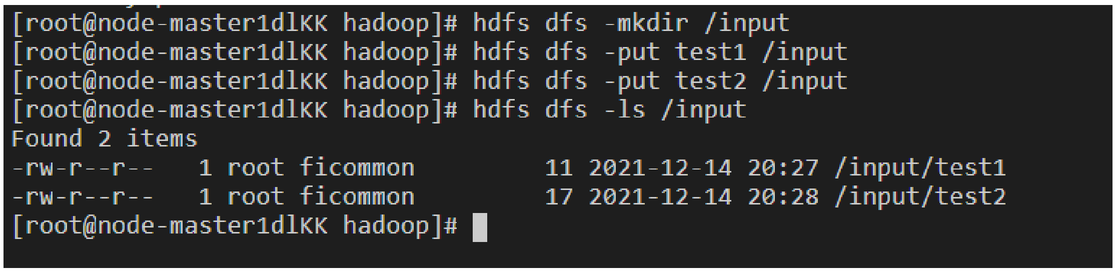

```
[root@node-master1dlKK hadoop]# jar cf WordCount.jar WordCount*.class
[root@node-master1dlKK hadoop]# ls
 bin                NOTICE.txt           'WordCount$IntSumReducer.class'
 etc                README.txt           'WordCount$TokenizerMapper.class'
 hdfs-c-example     sbin                  WordCount.class
 include            share                 WordCount.jar
 lib                test1                 WordCount.java
 libexec            test2
 LICENSE.txt        version.properties
[root@node-master1dlKK hadoop]#
```

4.1.5 创建文件 test1，内容为 hello hust，文件 test2，内容为 hello 学号，将他们放入 hdfs 的/input 文件夹内。(方法见实验一)(5’)


```
[root@node-master1dlKK hadoop]# vim test1
[root@node-master1dlKK hadoop]# vim test2
[root@node-master1dlKK hadoop]#
```


```
hello hust
~
~
~
~
```


```
hello X2020I1007
~
~
~
~
```



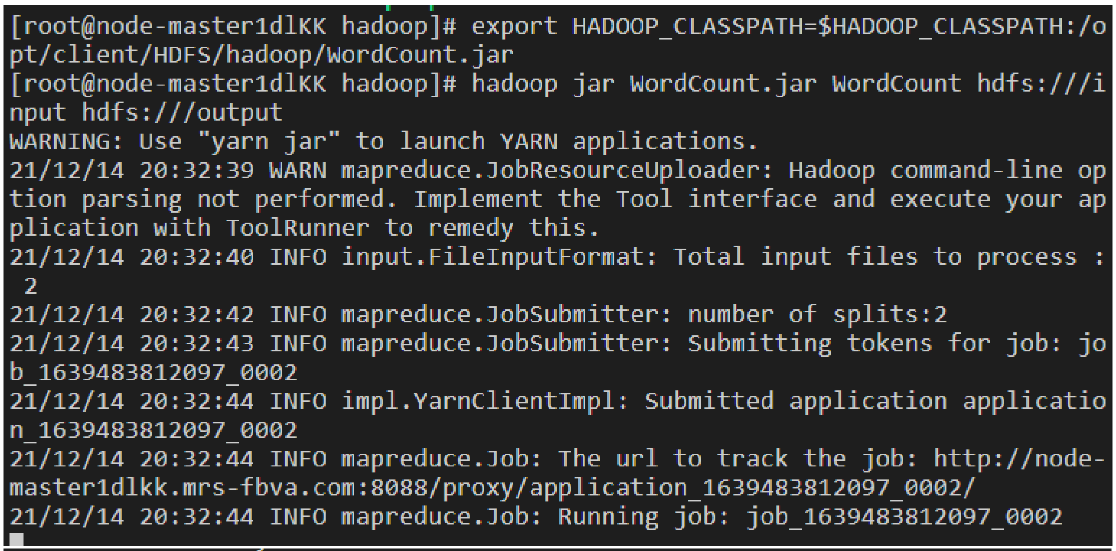
```
[root@node-master1dlKK hadoop]# hdfs dfs -mkdir /input
[root@node-master1dlKK hadoop]# hdfs dfs -put test1 /input
[root@node-master1dlKK hadoop]# hdfs dfs -put test2 /input
[root@node-master1dlKK hadoop]# hdfs dfs -ls /input
Found 2 items
-rw-r--r--   1 root ficommon         11 2021-12-14 20:27 /input/test1
-rw-r--r--   1 root ficommon         17 2021-12-14 20:28 /input/test2
[root@node-master1dlKK hadoop]# 
```

4.1.6 运行WordCount.jar 将hdfs的/input作为输入，/output 作为输出，并打印/output 目录下的文件，显示出词频统计的结果(5’)

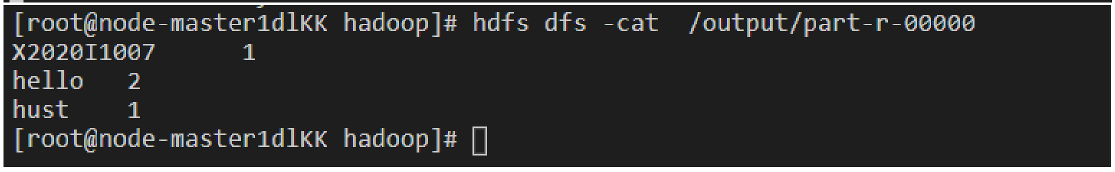
export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/opt/client/HDFS/hadoop/WordCount.jar

hadoop jar WordCount.jar WordCount hdfs:///input hdfs:///output

hdfs dfs -cat　/output/part-r-00000



```
[root@node-master1dlKK hadoop]# export HADOOP_CLASSPATH=$HADOOP_CLASSPATH:/o
pt/client/HDFS/hadoop/WordCount.jar
[root@node-master1dlKK hadoop]# hadoop jar WordCount.jar WordCount hdfs:///i
nput hdfs:///output
WARNING: Use "yarn jar" to launch YARN applications.
21/12/14 20:32:39 WARN mapreduce.JobResourceUploader: Hadoop command-line op
tion parsing not performed. Implement the Tool interface and execute your ap
plication with ToolRunner to remedy this.
21/12/14 20:32:40 INFO input.FileInputFormat: Total input files to process :
 2
21/12/14 20:32:42 INFO mapreduce.JobSubmitter: number of splits:2
21/12/14 20:32:43 INFO mapreduce.JobSubmitter: Submitting tokens for job: jo
b_1639483812097_0002
21/12/14 20:32:44 INFO impl.YarnClientImpl: Submitted application applicatio
n_1639483812097_0002
21/12/14 20:32:44 INFO mapreduce.Job: The url to track the job: http://node-
master1dlkk.mrs-fbva.com:8088/proxy/application_1639483812097_0002/
21/12/14 20:32:44 INFO mapreduce.Job: Running job: job_1639483812097_0002
```

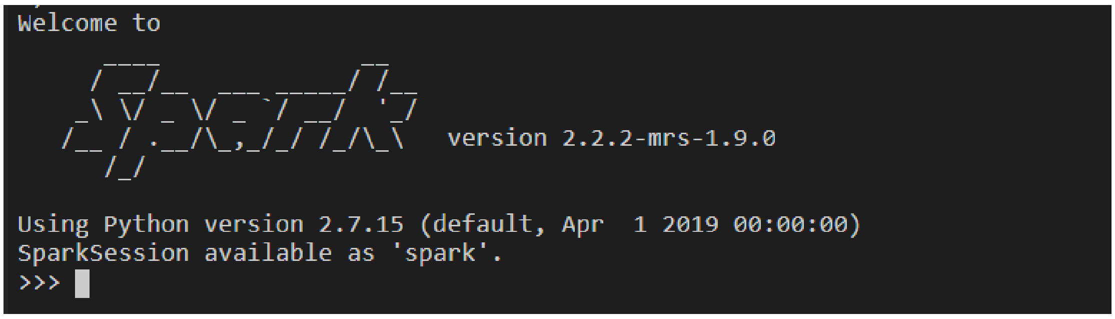


```
[root@node-master1dlKK hadoop]# hdfs dfs -cat  /output/part-r-00000
X2020I1007      1
hello   2
hust    1
[root@node-master1dlKK hadoop]# 
```

4.2 Spark

4.2.1 打开 spark(5’)

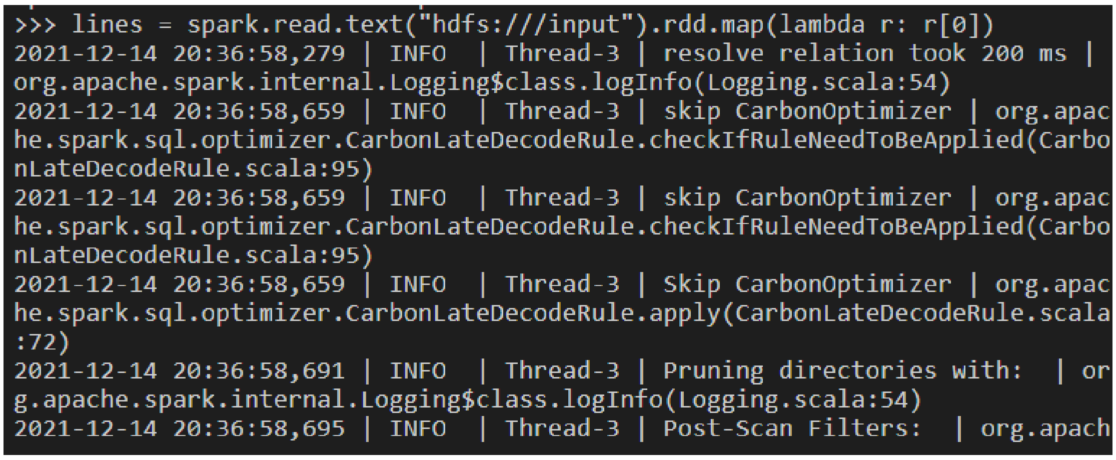
Pyspark



```
Welcome to
      ____              __
     / __/__  ___ _____/ /__
    _\ \/ _ \/ _ `/ __/  '_/
   /__ / .__/\_,_/_/ /_/\_\   version 2.2.2-mrs-1.9.0
      /_/

Using Python version 2.7.15 (default, Apr  1 2019 00:00:00)
SparkSession available as 'spark'.
>>> 
```

4.2.2 读取 hdfs 文件内容(5’)

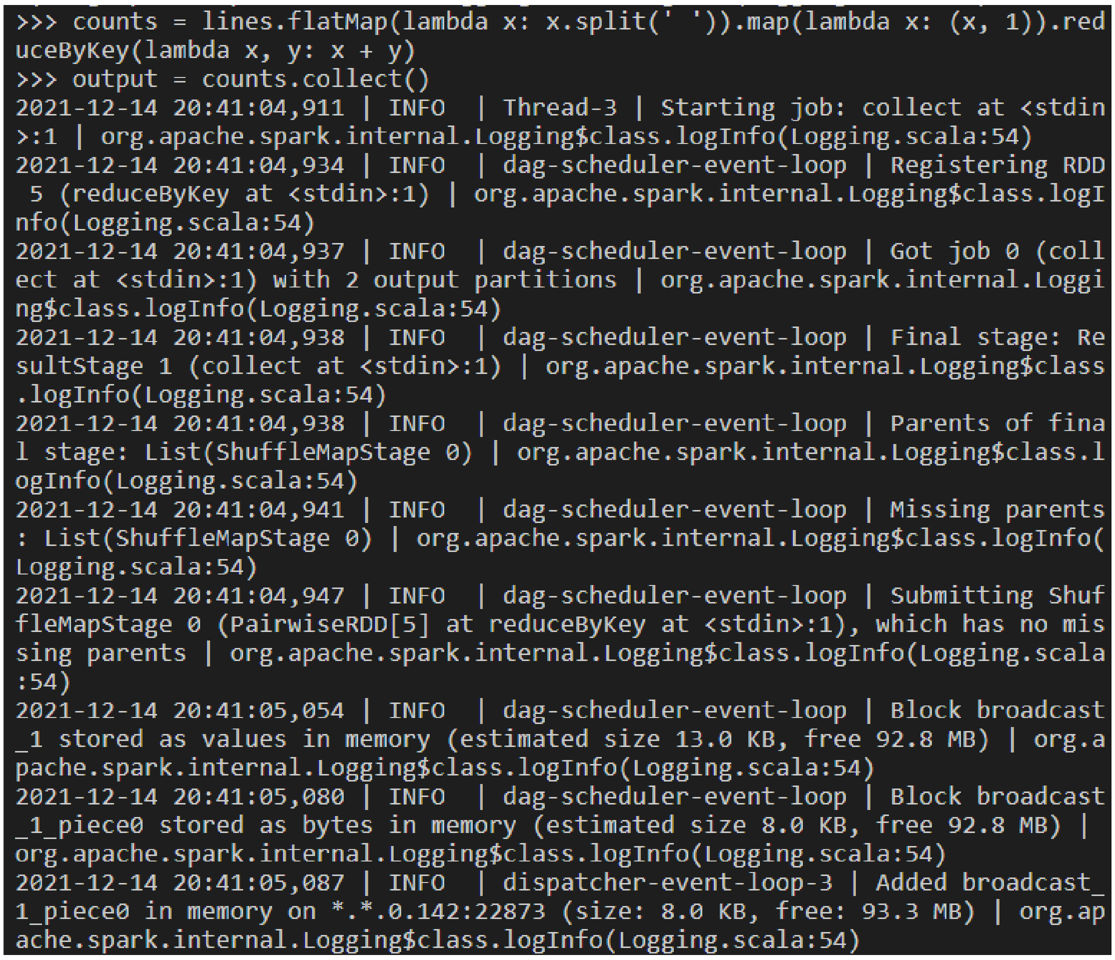
lines = spark.read.text("hdfs:///input").rdd.map(lambda r: r[0])



```
>>> lines = spark.read.text("hdfs:///input").rdd.map(lambda r: r[0])
2021-12-14 20:36:58,279 | INFO  | Thread-3 | resolve relation took 200 ms |
org.apache.spark.internal.Logging$class.logInfo(Logging.scala:54)
2021-12-14 20:36:58,659 | INFO  | Thread-3 | skip CarbonOptimizer | org.apac
he.spark.sql.optimizer.CarbonLateDecodeRule.checkIfRuleNeedToBeApplied(Carbo
nLateDecodeRule.scala:95)
2021-12-14 20:36:58,659 | INFO  | Thread-3 | skip CarbonOptimizer | org.apac
he.spark.sql.optimizer.CarbonLateDecodeRule.checkIfRuleNeedToBeApplied(Carbo
nLateDecodeRule.scala:95)
2021-12-14 20:36:58,659 | INFO  | Thread-3 | Skip CarbonOptimizer | org.apac
he.spark.sql.optimizer.CarbonLateDecodeRule.apply(CarbonLateDecodeRule.scala
:72)
2021-12-14 20:36:58,691 | INFO  | Thread-3 | Pruning directories with:  | or
g.apache.spark.internal.Logging$class.logInfo(Logging.scala:54)
2021-12-14 20:36:58,695 | INFO  | Thread-3 | Post-Scan Filters:  | org.apach
```

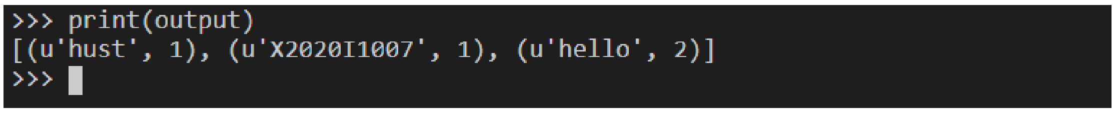
4.2.3 词频统计(5’)

counts = lines.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)).reduceByKey(lambda x, y: x + y)
output = counts.collect()



```
>>> counts = lines.flatMap(lambda x: x.split(' ')).map(lambda x: (x, 1)).red
uceByKey(lambda x, y: x + y)
>>> output = counts.collect()
2021-12-14 20:41:04,911 | INFO  | Thread-3 | Starting job: collect at <stdin
>:1 | org.apache.spark.internal.Logging$class.logInfo(Logging.scala:54)
2021-12-14 20:41:04,934 | INFO  | dag-scheduler-event-loop | Registering RDD
 5 (reduceByKey at <stdin>:1) | org.apache.spark.internal.Logging$class.logI
nfo(Logging.scala:54)
2021-12-14 20:41:04,937 | INFO  | dag-scheduler-event-loop | Got job 0 (coll
ect at <stdin>:1) with 2 output partitions | org.apache.spark.internal.Loggi
ng$class.logInfo(Logging.scala:54)
2021-12-14 20:41:04,938 | INFO  | dag-scheduler-event-loop | Final stage: Re
sultStage 1 (collect at <stdin>:1) | org.apache.spark.internal.Logging$class
.logInfo(Logging.scala:54)
2021-12-14 20:41:04,938 | INFO  | dag-scheduler-event-loop | Parents of fina
l stage: List(ShuffleMapStage 0) | org.apache.spark.internal.Logging$class.l
ogInfo(Logging.scala:54)
2021-12-14 20:41:04,941 | INFO  | dag-scheduler-event-loop | Missing parents
: List(ShuffleMapStage 0) | org.apache.spark.internal.Logging$class.logInfo(
Logging.scala:54)
2021-12-14 20:41:04,947 | INFO  | dag-scheduler-event-loop | Submitting Shuf
fleMapStage 0 (PairwiseRDD[5] at reduceByKey at <stdin>:1), which has no mis
sing parents | org.apache.spark.internal.Logging$class.logInfo(Logging.scala
:54)
2021-12-14 20:41:05,054 | INFO  | dag-scheduler-event-loop | Block broadcast
_1 stored as values in memory (estimated size 13.0 KB, free 92.8 MB) | org.a
pache.spark.internal.Logging$class.logInfo(Logging.scala:54)
2021-12-14 20:41:05,080 | INFO  | dag-scheduler-event-loop | Block broadcast
_1_piece0 stored as bytes in memory (estimated size 8.0 KB, free 92.8 MB) |
org.apache.spark.internal.Logging$class.logInfo(Logging.scala:54)
2021-12-14 20:41:05,087 | INFO  | dispatcher-event-loop-3 | Added broadcast_
1_piece0 in memory on *.*.0.142:22873 (size: 8.0 KB, free: 93.3 MB) | org.ap
ache.spark.internal.Logging$class.logInfo(Logging.scala:54)
```

4.2.4 输出结果(10’)



```
>>> print(output)
[(u'hust', 1), (u'X2020I1007', 1), (u'hello', 2)]
>>> 
```

4.3 附加题

file1:

20210001 Math 90

20210002 Math 80

20210003 Math 70

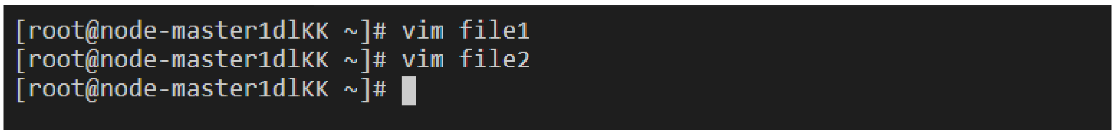
file2:

20210001 English 80

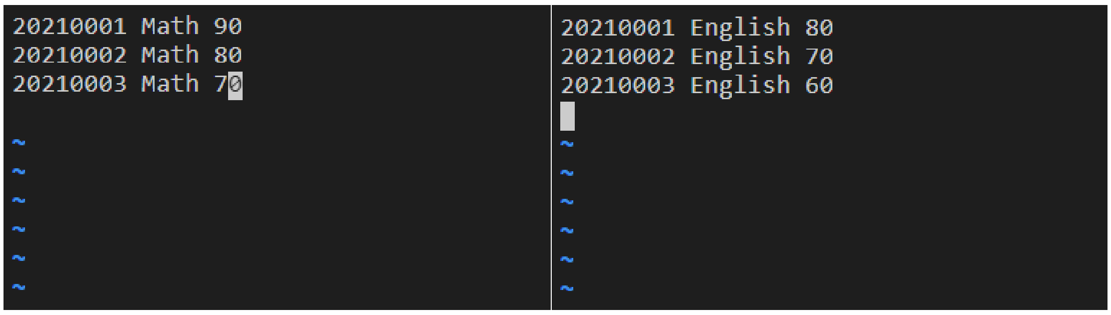
20210002 English 70

20210003 English 60

1.将以上两个文件存入 hdfs(10’)

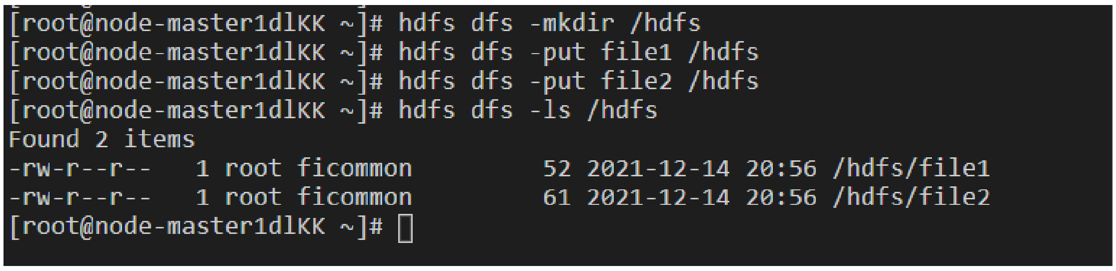


```
[root@node-master1dlKK ~]# vim file1
[root@node-master1dlKK ~]# vim file2
[root@node-master1dlKK ~]#
```



```
20210001 Math 90
20210002 Math 80
20210003 Math 70

~
~
~
~
~
~
```

```
20210001 English 80
20210002 English 70
20210003 English 60

~
~
~
~
~
~
```



```
[root@node-master1dlKK ~]# hdfs dfs -mkdir /hdfs
[root@node-master1dlKK ~]# hdfs dfs -put file1 /hdfs
[root@node-master1dlKK ~]# hdfs dfs -put file2 /hdfs
[root@node-master1dlKK ~]# hdfs dfs -ls /hdfs
Found 2 items
-rw-r--r--   1 root ficommon         52 2021-12-14 20:56 /hdfs/file1
-rw-r--r--   1 root ficommon         61 2021-12-14 20:56 /hdfs/file2
[root@node-master1dlKK ~]#
```

将上述内容编写入文件并存入名为 hdfs 文件夹内

2.编写 mapreduce 的程序，输出每门课的平均成绩。(10’)

3.编写 mapreduce 的程序，输出每位同学有多少门课成绩低于 75 分。(10’)

五：实验总结(10’)

通过本次实验，让我学会了在 Hadoop 平台上配置 java 环境并且编译.java 文件对数据文件 test1 与 test2 进行处理以及在 Spark 平台上使用 python 代码对数据文件进行词频统计计算，从而实现 WordCount 功能（单词词频统计计算功能）。从中对 MapReduce 大数据并行处理的实现方法有了更多的认知。而本次最大的难点在于我还没学过 Java 这门语言，以至于在附加题最后两题上出现了卡顿，不知道如何实现那两个算法，同时还读不懂 Java 程序。因此我认识到了自己的短板在于缺少了 Java 编程语言方面的知识，在后面的实验开始前，我会好好地恶补一下这方面的知识，从而使实验完成得更加顺利。

附录：

1 MapReduce 官方教程

https://hadoop.apache.org/docs/r2.8.3/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html

2 Spark 官方教程

https://spark.apache.org/examples.html