

Hadoop 数据

一.

数据获取：

1.代码：

获取的数据是 2017 年三明、泉州的某些月份的天气：

所以先获取获取时间，通过时间的增加来得到天气的遍历。

```
package com.example.weather;

import java.io.BufferedWriter;

/**
 * Hello world!
 */
public class App
{
    public static void main( String[] args )
    {
        /*String str = "";
        java.text.SimpleDateFormat format = new java.text.SimpleDateFormat("yyyy-MM-dd");
        Calendar cal = Calendar.getInstance();// 取当前日期。
        cal = Calendar.getInstance();
        cal.add(Calendar.DAY_OF_MONTH, -382);// 取当前日期的前N天。-606,-242
        str =format.format(cal.getTime());
        System.out.println("yesterday is:" + str);*/

        String str = "";

        for(int dtime=-5;dtime<-1;dtime++){
            java.text.SimpleDateFormat format = new java.text.SimpleDateFormat("yyyy-MM-dd");
            Calendar cal = Calendar.getInstance();// 取当前日期。
            cal = Calendar.getInstance();
            cal.add(Calendar.DAY_OF_MONTH, dtime);// 取当前日期的前N天。
            str =format.format(cal.getTime());
            System.out.println("yesterday is:" + str);

            String res= GetCityList.weather("134", str);
            JSONObject obj=JSONObject.fromObject(res);
```

```
JSONObject obj=JSONObject.fromObject(res);

String result=obj.getString("result");
//此时result中数据有多个key,可以对其key进行遍历,得到对个属性
obj=JSONObject.fromObject(result);
//今日温度对应的key是today
String city_id=obj.getString("city_id");//城市地区ID
String city_name=obj.getString("city_name");//城市地区名称
String weather_date=obj.getString("weather_date");//天气日期
String day_weather=obj.getString("day_weather");//白天天气
String night_weather=obj.getString("night_weather");//夜间天气
String day_temp=obj.getString("day_temp");//白天最高温度
String night_temp=obj.getString("night_temp");//夜间最低温度
String day_wind=obj.getString("day_wind");//白天风向
String day_wind_comp=obj.getString("day_wind_comp");//白天风力
String night_wind=obj.getString("night_wind");//夜间风向
String night_wind_comp=obj.getString("night_wind_comp");//夜间风力
String day_weather_id=obj.getString("day_weather_id");//白天天气标识
String night_weather_id=obj.getString("night_weather_id");//夜间天气标识
System.out.println(city_name+" "+weather_date+" "+day_weather+" "+night_weather+" "+
    day_temp+" "+night_temp+" "+day_wind+" "+day_wind_comp+" "+night_wind+" "+
    night_wind_comp+" "+day_weather_id+" "+night_weather_id);

List<String> list = new LinkedList<String>();
list.add(city_id);
list.add(city_name);
list.add(weather_date);
list.add(day_weather);
list.add(night_weather);
list.add(day_temp);
list.add(night_temp);
list.add(day_wind);
list.add(day_wind_comp);
list.add(night_wind);
list.add(night_wind_comp);
list.add(day_weather_id);
list.add(night_weather_id);
```

```

73         list.add(day_temp);
74         list.add(night_temp);
75         list.add(day_wind);
76         list.add(day_wind_comp);
77         list.add(night_wind);
78         list.add(night_wind_comp);
79         list.add(day_weather_id);
80         list.add(night_weather_id);
81
82         File file1 = new File("F:\\WEATHER4.txt");
83         try {
84             FileWriter fw = new FileWriter(file1,true);
85             BufferedWriter bw = new BufferedWriter(fw);
86
87             for(int i = 0; i<list.size();i++){
88                 bw.write(list.get(i).toString()+" ");
89                 bw.flush();
90                 //System.out.println(list.size());
91             }
92             bw.newLine();
93             bw.close();
94             fw.close();
95
96         } catch (IOException e) {
97             e.printStackTrace();
98         }
99     }
100 }
101 }
102

```

```
File file1 = new File("F:\\WEATHER4.txt");
```

这文档名字可以改多次。




主函数调用类 GetCityList 的 weather 方法：

```

public static String weather(String city, String wdate){
    String url= "http://v.juhe.cn/historyWeather/weather?city_id="+city+"&key=60491dbf8
    return PureNetUtil.get(url);//使用get方法
}

```

数据结果：

 WEATHER1.txt	2018/8/30 10:07	文本文档	14 KB
 WEATHER2.txt	2018/8/30 10:11	文本文档	34 KB
 WEATHER4.txt	2018/8/30 14:27	文本文档	1 KB

```
WEATHER1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
134 三明 2017-01-01 多云 多云 22°C 10°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
134 三明 2017-01-02 多云 多云 23°C 12°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
134 三明 2017-01-03 晴 晴 25°C 14°C 无持续风向 ≤3级 无持续风向 ≤3级 00 00
134 三明 2017-01-04 晴 晴 25°C 13°C 无持续风向 ≤3级 无持续风向 ≤3级 00 00
134 三明 2017-01-05 晴 晴 25°C 16°C 无持续风向 ≤3级 无持续风向 ≤3级 00 00
134 三明 2017-01-06 阴 小雨 25°C 15°C 无持续风向 ≤3级 无持续风向 ≤3级 02 07
134 三明 2017-01-07 小雨 阴 20°C 14°C 无持续风向 ≤3级 无持续风向 ≤3级 07 02
134 三明 2017-01-08 多云 多云 18°C 10°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
134 三明 2017-01-09 晴 多云 20°C 11°C 无持续风向 ≤3级 无持续风向 ≤3级 00 01
134 三明 2017-01-10 多云 多云 21°C 11°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
134 三明 2017-01-11 阴 小雨 15°C 12°C 无持续风向 ≤3级 无持续风向 ≤3级 02 07
134 三明 2017-01-12 小雨 小雨 14°C 9°C 无持续风向 ≤3级 无持续风向 ≤3级 07 07
134 三明 2017-01-13 小雨 小雨 10°C 8°C 北风 3-4级 无持续风向 ≤3级 07 07
134 三明 2017-01-14 小雨 阴 11°C 8°C 无持续风向 ≤3级 无持续风向 ≤3级 07 02
134 三明 2017-01-15 阴 多云 11°C 7°C 无持续风向 ≤3级 无持续风向 ≤3级 02 01
134 三明 2017-01-16 多云 阴 16°C 9°C 无持续风向 ≤3级 无持续风向 ≤3级 01 02
134 三明 2017-01-17 小雨 小雨 13°C 9°C 无持续风向 ≤3级 无持续风向 ≤3级 07 07
134 三明 2017-01-18 小雨 小雨 14°C 11°C 无持续风向 ≤3级 无持续风向 ≤3级 07 07
134 三明 2017-01-19 小雨 阴 14°C 9°C 无持续风向 ≤3级 无持续风向 ≤3级 07 02
134 三明 2017-01-20 阴 多云 11°C 5°C 无持续风向 ≤3级 无持续风向 ≤3级 02 01
134 三明 2017-01-21 多云 晴 15°C 3°C 无持续风向 ≤3级 无持续风向 ≤3级 01 00
134 三明 2017-01-22 晴 晴 16°C 4°C 无持续风向 ≤3级 无持续风向 ≤3级 00 00
134 三明 2017-01-23 晴 晴 16°C 4°C 无持续风向 ≤3级 无持续风向 ≤3级 00 00
134 三明 2017-01-24 晴 多云 17°C 6°C 无持续风向 ≤3级 无持续风向 ≤3级 00 01
```

```
WEATHER2.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
133 泉州 2017-01-01 多云 多云 22°C 16°C 东北风 3-4级 东北风 3-4级 01 01
133 泉州 2017-01-02 多云 多云 25°C 16°C 无持续风向 ≤3级 东北风 3-4级 01 01
133 泉州 2017-01-03 多云 多云 22°C 15°C 东北风 3-4级 无持续风向 ≤3级 01 01
133 泉州 2017-01-04 多云 多云 23°C 15°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
133 泉州 2017-01-05 多云 多云 24°C 16°C 无持续风向 ≤3级 无持续风向 ≤3级 01 01
133 泉州 2017-01-06 阴 阴 23°C 16°C 无持续风向 ≤3级 无持续风向 ≤3级 02 02
133 泉州 2017-01-07 阴 阴 21°C 16°C 无持续风向 ≤3级 无持续风向 ≤3级 02 02
133 泉州 2017-01-08 多云 多云 22°C 13°C 东北风 3-4级 东北风 3-4级 01 01
133 泉州 2017-01-09 多云 多云 17°C 10°C 东北风 3-4级 东北风 3-4级 01 01
133 泉州 2017-01-10 多云 多云 18°C 12°C 东北风 3-4级 东北风 3-4级 01 01
133 泉州 2017-01-11 多云 阴 18°C 13°C 东北风 3-4级 无持续风向 ≤3级 01 02
133 泉州 2017-01-12 小雨 小雨 17°C 13°C 无持续风向 ≤3级 无持续风向 ≤3级 07 07
133 泉州 2017-01-13 阴 小雨 13°C 11°C 东北风 3-4级 东北风 3-4级 02 07
133 泉州 2017-01-14 阴 阴 13°C 9°C 无持续风向 ≤3级 东北风 4-5级 02 02
133 泉州 2017-01-15 阴 阴 13°C 9°C 东北风 3-4级 东北风 3-4级 02 02
133 泉州 2017-01-16 多云 阴 15°C 11°C 东北风 3-4级 东北风 3-4级 01 02
133 泉州 2017-01-17 阴 阴 16°C 12°C 东北风 3-4级 东北风 3-4级 02 02
133 泉州 2017-01-18 阴 阵雨 17°C 14°C 东北风 3-4级 无持续风向 ≤3级 02 03
133 泉州 2017-01-19 小雨 阴 18°C 12°C 东北风 3-4级 东北风 4-5级 07 02
133 泉州 2017-01-20 阴 多云 13°C 7°C 东北风 4-5级 东北风 4-5级 02 01
133 泉州 2017-01-21 多云 晴 14°C 7°C 东北风 3-4级 东北风 3-4级 01 00
133 泉州 2017-01-22 晴 多云 16°C 8°C 东北风 4-5级 东北风 4-5级 00 01
133 泉州 2017-01-23 晴 多云 14°C 8°C 东北风 4-5级 东北风 4-5级 00 01
133 泉州 2017-01-24 多云 多云 15°C 9°C 东北风 4-5级 东北风 4-5级 01 01
```

```
WEATHER4.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
134 三明 2018-08-20 多云 阴 35°C 26°C 东南风 4-5级 南风 <3级 01 02
134 三明 2018-08-21 多云 多云 36°C 26°C 东南风 3-4级 西南风 3-4级 01 01
134 三明 2018-08-22 阴 阴 35°C 25°C 东北风 3-4级 东北风 3-4级 02 02
134 三明 2018-08-23 中雨 中雨 33°C 25°C 西北风 <3级 东北风 <3级 08 08
134 三明 2018-08-24 小雨 阴 30°C 22°C 东北风 3-4级 东风 3-4级 07 02
134 三明 2018-08-25 阴 阴 24°C 23°C 东风 <3级 东北风 3-4级 02 02
134 三明 2018-08-26 阴 阴 28°C 24°C 东北风 3-4级 东风 3-4级 02 02
```


二、

环境搭建：

新建虚拟机



选择类型版本

虚拟电脑名称和系统类型

请选择新虚拟电脑的描述名称及要安装的操作。此名称将用于标识此虚拟电脑。

名称(N):	Centos7_64
类型(T):	Linux
版本(V):	Red Hat (64-bit)

选择内存：

内存大小

选择分配给虚拟电脑的内存大小(MB)。

建议的内存大小为 **1024 MB**。

4 MB 8192 MB

1024 MB

现在创建虚拟硬盘


虚拟硬盘

你可以添加虚拟硬盘到新虚拟电脑中。新建一个虚拟硬盘文件或从列表或用文件夹图标从其他位置选择一个。

如果想更灵活地配置虚拟硬盘，也可以跳过这一步，在创建虚拟电脑之后在配置中设定。

建议的硬盘大小为 **8.00 GB**。

- ☐ 不添加虚拟硬盘(D)
- ☒ 现在创建虚拟硬盘(C)
- ☐ 使用已有的虚拟硬盘文件(U)

没有盘片 

选择 VDI

虚拟硬盘文件类型

请选择您想要用于新建虚拟磁盘的文件类型。如果您不需要其他虚拟化软件使用它，您可以让此设置保持不更改状态。

- ☒ VDI (VirtualBox 磁盘映像)
- ☐ VHD (虚拟硬盘)
- ☐ VMK (虚拟机磁盘)

硬盘足够就选择 固定大小启动之后选择 Centos 的安装文件

存储在物理硬盘上

请选择新建虚拟硬盘文件是应该为其使用而分配(动态分配)，还是应该创建完全分配(固定分配)。

动态分配的虚拟磁盘只是逐渐占用物理硬盘的空间（直至达到**分配的大小**），不过当其内部空间不用时不会自动缩减占用的物理硬盘空间。

固定大小的虚拟磁盘文件可能在某些系统中要花很长时间来创建，但它往往使用起来较快。

- ☐ 动态分配(D)
- ☒ 固定大小(F)

请选择一个虚拟光盘文件或已放入光盘的光驱来启动虚拟电脑。

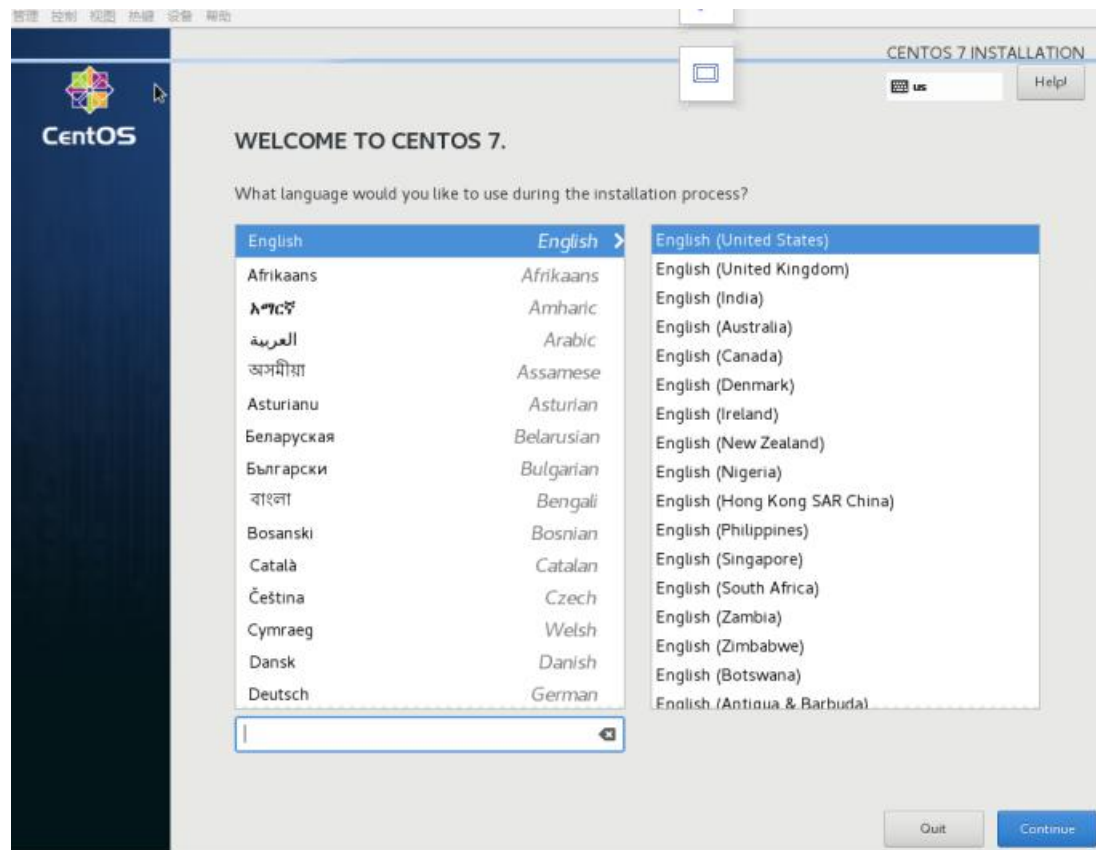
此光盘应可启动并且有你想安装的操作系统。下次关闭虚拟电脑时，此光盘可自动弹出；你也可以手动弹出。

CentOS-7-x86_64-DVD-1804.iso 

启动

取消

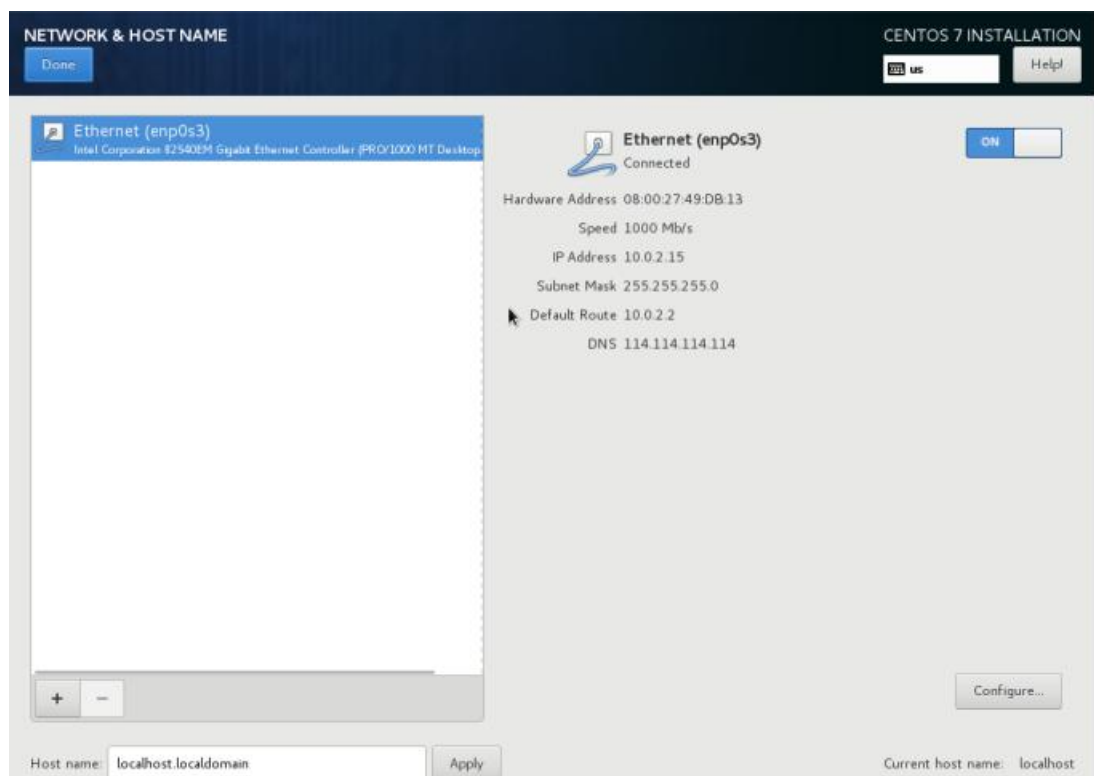
稍后就回出现一个图形化界面，选择语言



时间选上海



网络打开



点进去在重新点 done 之后 “！” 这标志就没有了



设置账号密码:

The image shows the "ROOT PASSWORD" screen in the CentOS 7 installation process. The title bar says "CENTOS 7 INSTALLATION". There is a "Done" button in the top left and a "Help" button in the top right. The main text says: "The root account is used for administering the system. Enter a password for the root user." Below this, there are two input fields: "Root Password:" and "Confirm:". The "Root Password:" field has a strength indicator below it showing a green bar and the word "Weak".

点 done 自动安装

The image shows the "CONFIGURATION" screen in the CentOS 7 installation process. The title bar says "CENTOS 7 INSTALLATION". There is a "Help" button in the top right. The main content area is titled "USER SETTINGS" and contains two cards: "ROOT PASSWORD" with a key icon and the text "Root password is set", and "USER CREATION" with a person icon and the text "No user will be created". At the bottom, there is a "Complete!" message with a progress bar and a "Reboot" button.

安装完成后面启动登陆



登陆进去进行配置:

1. 安装 ifconfig

```
yum install net-tools
```

2. 安装 jdk

//把本机公钥送入到目标机器的 authorized_keys 文件中

```
ssh root@192.168.4.223 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

上传文件

```
scp jdk-8u144-linux-x64.tar.gz root@192.168.3.181:~/.
```

解压 jdk

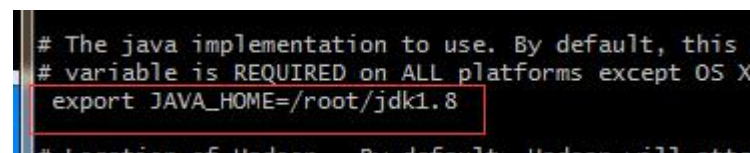
```
tar xvf jdk-...
```

建立软连接

```
ln -s jdk1.8.0_144/ jdk1.8
```

3. 设置 JAVA_HOME

设置 path 路径



4. 在每个节点建立目录

```
cd /root/hadoop/data
```

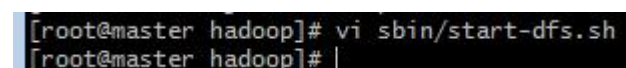
```
mkdir -p datanode namenode tmp localdir logdir
```

//清空目录内容, 为下次启动准备

```
rm -rf datanode/* namenode/* localdir/* logdir/* tmp/*
```

5 修改/bin/hdfs 中的 start-dfs.sh、stop-dfs.sh 文件, 添加以下内容

start-dfs.sh:



```
root@master:~/hadoop
#!/usr/bin/env bash

# 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 owners.
# 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
#
#     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.

HDFS_NAMENODE_USER=root
HDFS_DATANODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root
```

stop-dfs.sh:

```
[root@master hadoop]# vi sbin/start-dfs.sh
[root@master hadoop]# vi sbin/stop-dfs.sh
[root@master hadoop]#
```

```
root@master:~/hadoop
#!/usr/bin/env bash

# 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 owners.
# 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
#
#     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.

HDFS_NAMENODE_USER=root
HDFS_DATANODE_USER=root
HDFS_SECONDARYNAMENODE_USER=root
```

6、格式化

```
[root@master hadoop]# bin/hdfs namenode -format
2018-09-03 09:16:01,423 INFO namenode.NameNode: STARTUP_MSG:
/*
```

7 启动

```

*****
[root@master hadoop]# sbin/start-dfs.sh
Starting namenodes on [localhost]
Last login: Mon Sep  3 09:04:35 CST 2018 from 192.1
Starting datanodes
Last login: Mon Sep  3 09:20:34 CST 2018 on pts/3
Starting secondary namenodes [master]
Last login: Mon Sep  3 09:20:37 CST 2018 on pts/3

```

8、查看结果

```

[root@master hadoop]# jps
2002 SecondaryNameNode
1635 NameNode
2246 Jps
1769 DataNode

```

联机操作

1.网卡设置成桥接网卡



2. 三台机子的虚拟 IP

```

:::1      localhost localh
192.168.4.122 master
192.168.4.230 slave1
192.168.4.223 slave2
~
~

```

3. 配置静态 ip

```
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="906114ca-6931-4169-ab0b-b6811e1f63cf"
DEVICE="enp0s3"
ONBOOT="yes"
IPADDR="192.168.4.223"
```

4. slave1、slave2 中的公钥传到 master 中的 authorized_key 里面

```
A[root@slave1 ssh root@master 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub]
```

```
Last login: Mon Sep  3 09:43:30 2018 from 192.168.4.222
[root@slave2 ~]# ssh root@master 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

master 中的公钥传到 slave1 和 slave2 中的 authorized_key

```
Last login: Mon Sep  3 09:43:30 2018 from 192.168.4.223
[root@master ~]# ssh root@slave1 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

```
Last login: Mon Sep  3 09:38:30 2018 from 192.168.4.223
[root@master ~]# ssh root@slave2 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

5. 三台机器已 ping 通，可以无密码访问：

```
Last login: Mon Sep  3 13:47:43 2018
[root@slave1 ~]# ssh master
Last login: Mon Sep  3 14:33:01 2018
[root@master ~]# |
```

```
Last login: Mon Sep  3 10:01:23 2018
[root@slave2 ~]# ssh master
Last login: Mon Sep  3 14:35:18 2018
[root@master ~]#
```

```
Last login: Mon Sep  3 14:33:54 2018
[root@master ~]# ssh slave1
Last login: Mon Sep  3 14:33:54 2018
[root@slave1 ~]#
```

```
[root@master ~]# ssh slave2
Last login: Mon Sep  3 10:01:25 2018
[root@slave2 ~]#
```

到这配置基本完成

三、

数据分析

1. 由于前面已经得到数据了，再把这些转换成我们需要的或者是统计的数据。

2. 代码：

把文档数据传值进来

```
WordCountMapper.java  WordCountReducer.java  wordcount/pom.xml  WordCountMapReduce.java  AppTest.java  App.java
1 package demo.mr.woracount;
2
3 import java.io.IOException;
4
5 public class WordCountMapper extends Mapper<LongWritable, Text, Text, Text> {
6
7     @Override
8     protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, Text>.Context
9         throws IOException, InterruptedException {
10         String line = value.toString();
11         String[] words = line.split(" ");
12         String id = words[0];
13         String cityname = words[1];
14         String datetime = StringUtils.substringBeforeLast(words[2], "-");
15         String temperture1 = words[5];
16         String temperture2 = words[6];
17
18         String sb1=StringUtils.substringBefore(temperture1, "C");
19         String sb2=StringUtils.substringBefore(temperture2, "C");
20
21         //int numtemperture =Integer.parseInt(sb);
22         context.write(new Text((cityname+"-"+datetime), new Text(sb1+" "+sb2));
23         //context.write(new Text(id+"-"+id), new Text(id));
24         //context.write(new Text(v), new IntWritable(temperture));
25
26         /*for(String word: words) {
27             context.write(new Text(word), new IntWritable(1));
28         }*/
29     }
30 }
```

```

1 package demo.mr.wordcount;
2
3 import org.apache.hadoop.conf.Configuration;
4
5 public class WordCountMapReduce
6 {
7     public static void main( String[] args ) throws Exception
8     {
9         Configuration cfg = new Configuration();
10        Job job = Job.getInstance(cfg, "wordcount");
11        job.setJarByClass(WordCountMapReduce.class);
12
13        FileInputFormat.setInputPaths(job, new Path(args[0]));
14        FileOutputFormat.setOutputPath(job, new Path(args[1]));
15
16        job.setMapperClass(WordCountMapper.class);
17        job.setReducerClass(WordCountReducer.class);
18
19        job.setMapOutputKeyClass(Text.class);
20        job.setMapOutputValueClass(Text.class);
21
22        job.setOutputKeyClass(Text.class);
23        job.setOutputValueClass(Text.class);
24
25        boolean b = job.waitForCompletion(true);
26        if(!b) {
27            System.out.println("wordcount task fail!");
28        }
29    }
30 }

```

利用传进来的值进行比较数据或者统计数据。

```

1 package demo.mr.wordcount;
2
3 import java.io.IOException;
4
5 public class WordCountReducer extends Reducer<Text, Text, Text, Text> {
6
7     @Override
8     protected void reduce(Text key, Iterable<Text> values,
9         Reducer<Text, Text, Text, Text>.Context context) throws IOException, InterruptedException {
10
11         /*Integer count = 0;
12         for(IntWritable value : values) {
13             count+=value.get();
14         }
15         context.write(key, new IntWritable(count));*/
16
17         Integer sum=0;String s=null;Integer night=0;
18         int avgtemperature=0;
19         int niavgtemperature=0;
20         int hightemperature=0;
21         int lowtemperature=1000;
22         int highnitemperature=0;
23         int lownitemperature=1000;
24         Text t=null;
25         String[] str=null;
26         int i=0;
27         for(Text value : values) {
28             s=value.toString();
29             str=s.split(" ");
30
31             if(hightemperature<Integer.parseInt(strs[0])){
32                 hightemperature=Integer.parseInt(strs[0]);
33             }
34         }
35     }
36 }

```

```

1 //
2 Text d = null;
3 d = new Text("白天平均温度: "+avgtemperature+"℃ 夜晚平均温度: "+niavgtemperature+"℃ 白天最高温度: "+hightemperature+"℃ 白天最低温度: "
4 +lowtemperature+"℃ 夜晚最高温度: "+highnitemperature+"℃ "+ "℃ 夜晚最低温度: "+lownitemperature+"℃");
5 //String e = Integer.toString(niavgtemperature);
6
7 //StringBuilder stringBuilder=new StringBuilder().append(d).append(" ").append(e);
8
9 context.write(key,new Text(d));
10
11
12 /*for(Text value : values) {
13     t=value;
14 }*/
15
16
17 }
18
19 }
20 }

```

3.

把这份代码上传虚拟机上，并创建目录：

```

[root@master hadoop]# bin/hdfs dfs -mkdir -p /weather/input
[root@master hadoop]#

```

bin/hdfs dfs -put *.txt /wordcount/input 把所有文档放在 input 目录

进行数据分析

```

[root@master hadoop]# bin/hadoop jar ~/weather-0.0.1.jar demo.mr.wordcount.WordC
ountMapReduce /weather/input /weather/output16

```

结果：

```

root@master:~/hadoop
Bytes Written=3004
[root@master hadoop]# bin/hdfs dfs -cat /weather1/output16/*
三明--2017-01 白天平均温度: 17℃ 夜晚平均温度: 9℃ 白天最高温度: 25℃ 白天最低温度: 10℃ 夜晚最高温度: 16℃ 夜晚最低温度: 3℃
三明--2017-02 白天平均温度: 18℃ 夜晚平均温度: 8℃ 白天最高温度: 27℃ 白天最低温度: 7℃ 夜晚最高温度: 15℃ 夜晚最低温度: 2℃
三明--2017-03 白天平均温度: 19℃ 夜晚平均温度: 10℃ 白天最高温度: 27℃ 白天最低温度: 13℃ 夜晚最高温度: 17℃ 夜晚最低温度: 5℃
三明--2017-04 白天平均温度: 25℃ 夜晚平均温度: 15℃ 白天最高温度: 31℃ 白天最低温度: 18℃ 夜晚最高温度: 22℃ 夜晚最低温度: 9℃
三明--2017-05 白天平均温度: 30℃ 夜晚平均温度: 20℃ 白天最高温度: 34℃ 白天最低温度: 27℃ 夜晚最高温度: 22℃ 夜晚最低温度: 19℃
三明--2018-08 白天平均温度: 31℃ 夜晚平均温度: 24℃ 白天最高温度: 36℃ 白天最低温度: 24℃ 夜晚最高温度: 26℃ 夜晚最低温度: 22℃
厦门--2018-08 白天平均温度: 35℃ 夜晚平均温度: 26℃ 白天最高温度: 35℃ 白天最低温度: 35℃ 夜晚最高温度: 26℃ 夜晚最低温度: 26℃
泉州--2017-01 白天平均温度: 17℃ 夜晚平均温度: 11℃ 白天最高温度: 25℃ 白天最低温度: 13℃ 夜晚最高温度: 16℃ 夜晚最低温度: 7℃
泉州--2017-02 白天平均温度: 17℃ 夜晚平均温度: 9℃ 白天最高温度: 24℃ 白天最低温度: 10℃ 夜晚最高温度: 14℃ 夜晚最低温度: 6℃
泉州--2017-03 白天平均温度: 18℃ 夜晚平均温度: 12℃ 白天最高温度: 23℃ 白天最低温度: 12℃ 夜晚最高温度: 16℃ 夜晚最低温度: 8℃
泉州--2017-04 白天平均温度: 24℃ 夜晚平均温度: 17℃ 白天最高温度: 30℃ 白天最低温度: 18℃ 夜晚最高温度: 22℃ 夜晚最低温度: 10℃
泉州--2017-05 白天平均温度: 28℃ 夜晚平均温度: 21℃ 白天最高温度: 31℃ 白天最低温度: 25℃ 夜晚最高温度: 24℃ 夜晚最低温度: 18℃
泉州--2017-06 白天平均温度: 29℃ 夜晚平均温度: 24℃ 白天最高温度: 33℃ 白天最低温度: 23℃ 夜晚最高温度: 26℃ 夜晚最低温度: 20℃
泉州--2017-07 白天平均温度: 33℃ 夜晚平均温度: 25℃ 白天最高温度: 36℃ 白天最低温度: 29℃ 夜晚最高温度: 27℃ 夜晚最低温度: 24℃
泉州--2017-08 白天平均温度: 33℃ 夜晚平均温度: 26℃ 白天最高温度: 35℃ 白天最低温度: 29℃ 夜晚最高温度: 28℃ 夜晚最低温度: 26℃
[root@master hadoop]# bin/hdfs dfs -ls /weather1/input
Found 4 items
-rw-r--r-- 1 root supergroup 154 2018-08-31 10:03 /weather1/input/WEATHER.txt
-rw-r--r-- 1 root supergroup 13465 2018-08-31 10:04 /weather1/input/WEATHER1.txt
-rw-r--r-- 1 root supergroup 20567 2018-08-31 10:05 /weather1/input/WEATHER2.txt
-rw-r--r-- 1 root supergroup 590 2018-08-30 16:16 /weather1/input/WEATHER4.txt
[root@master hadoop]#

```