

# 大数据数据初步分析

## 一、数据抓取：

### （一）数据获取代码：

这次要获取的数据是 2017 年仙游和福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。

```
String str = "";
for(int dtim=-606;dtim<-382;dtim++){
    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, dtim);// 取当前日期的前N天。
    str =format.format(cal.getTime());//将时间格式转换成yyyy-MM-dd
```

然后是获取数据的代码：

```
String res= GetCityList.weather("147", str);//调用这个方法，返回城市编号为147，日期为str的天气信息
JSONObject obj=JSONObject.fromObject(res);//将返回结果装换成Json格式
String result=obj.getString("result");//把Json中的result数组赋值给字符串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);
```

```
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>();//设置list, 将数据放入list
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);

```

/\*将list中的数据写入文本文档中\*/

```

File file1 = new File("F:\\WEATHER1.txt");
try {
    FileWriter fw = new FileWriter(file1,true);
    BufferedWriter bw = new BufferedWriter(fw);

    for(int i = 0; i<list.size();i++){
        bw.write(list.get(i).toString()+" ");//每个数据后面用空格隔开
        bw.flush();
        //System.out.println(list.size());
    }
    bw.newLine(); //得到一行数据后换行
    bw.close();
    fw.close();

} catch (IOException e) {
    e.printStackTrace();
}

```

---

所调用的方法:

参数是城市 ID 和日期 (日期格式是 YYYY-MM-DD)

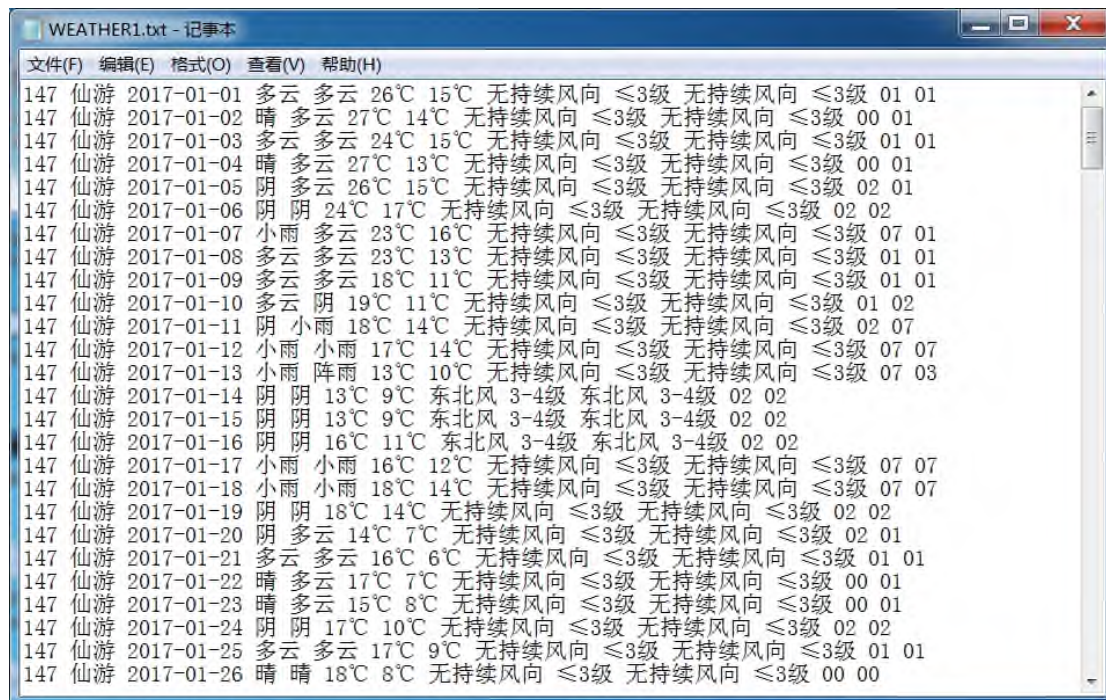
```

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
}

```

---

## (二) 得到的数据截图：



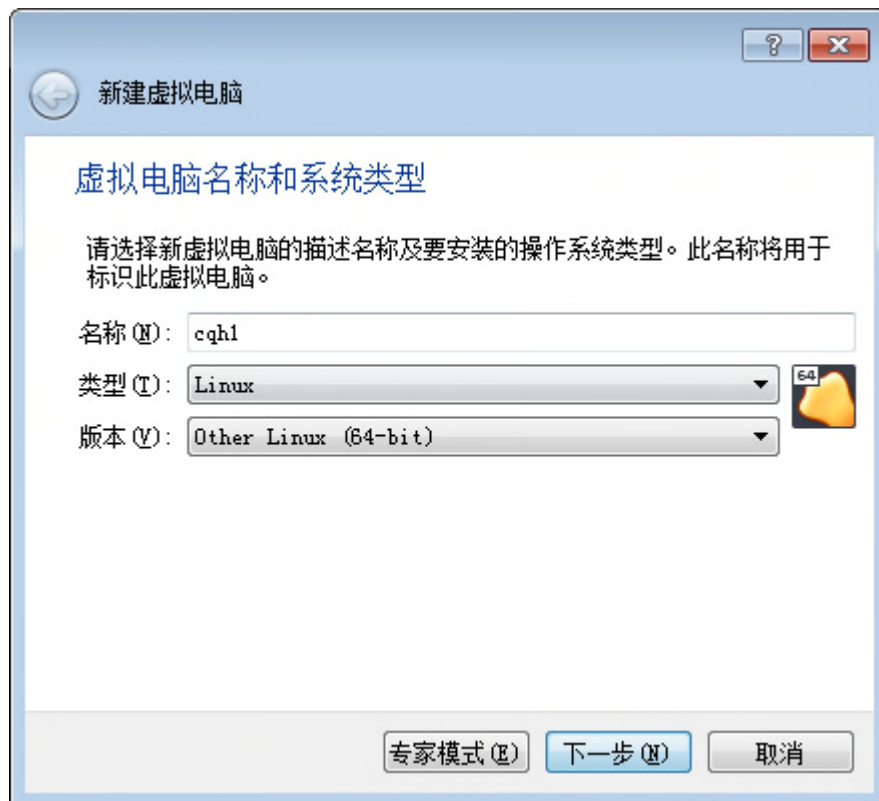
147	仙游	2017-01-01	多云	多云	26℃	15℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-02	晴	多云	27℃	14℃	无持续风向	≤3级	无持续风向	≤3级	00	01
147	仙游	2017-01-03	多云	多云	24℃	15℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-04	晴	多云	27℃	13℃	无持续风向	≤3级	无持续风向	≤3级	00	01
147	仙游	2017-01-05	阴	多云	26℃	15℃	无持续风向	≤3级	无持续风向	≤3级	02	01
147	仙游	2017-01-06	阴	阴	24℃	17℃	无持续风向	≤3级	无持续风向	≤3级	02	02
147	仙游	2017-01-07	小雨	多云	23℃	16℃	无持续风向	≤3级	无持续风向	≤3级	07	01
147	仙游	2017-01-08	多云	多云	23℃	13℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-09	多云	多云	18℃	11℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-10	多云	阴	19℃	11℃	无持续风向	≤3级	无持续风向	≤3级	01	02
147	仙游	2017-01-11	阴	小雨	18℃	14℃	无持续风向	≤3级	无持续风向	≤3级	02	07
147	仙游	2017-01-12	小雨	小雨	17℃	14℃	无持续风向	≤3级	无持续风向	≤3级	07	07
147	仙游	2017-01-13	小雨	阵雨	13℃	10℃	无持续风向	≤3级	无持续风向	≤3级	07	03
147	仙游	2017-01-14	阴	阴	13℃	9℃	东北风 3-4级	东北风 3-4级	02	02		
147	仙游	2017-01-15	阴	阴	13℃	9℃	东北风 3-4级	东北风 3-4级	02	02		
147	仙游	2017-01-16	阴	阴	16℃	11℃	东北风 3-4级	东北风 3-4级	02	02		
147	仙游	2017-01-17	小雨	小雨	16℃	12℃	无持续风向	≤3级	无持续风向	≤3级	07	07
147	仙游	2017-01-18	小雨	小雨	18℃	14℃	无持续风向	≤3级	无持续风向	≤3级	07	07
147	仙游	2017-01-19	阴	阴	18℃	14℃	无持续风向	≤3级	无持续风向	≤3级	02	02
147	仙游	2017-01-20	阴	多云	14℃	7℃	无持续风向	≤3级	无持续风向	≤3级	02	01
147	仙游	2017-01-21	多云	多云	16℃	6℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-22	晴	多云	17℃	7℃	无持续风向	≤3级	无持续风向	≤3级	00	01
147	仙游	2017-01-23	晴	多云	15℃	8℃	无持续风向	≤3级	无持续风向	≤3级	00	01
147	仙游	2017-01-24	阴	阴	17℃	10℃	无持续风向	≤3级	无持续风向	≤3级	02	02
147	仙游	2017-01-25	多云	多云	17℃	9℃	无持续风向	≤3级	无持续风向	≤3级	01	01
147	仙游	2017-01-26	晴	晴	18℃	8℃	无持续风向	≤3级	无持续风向	≤3级	00	00

## 二、环境搭建：

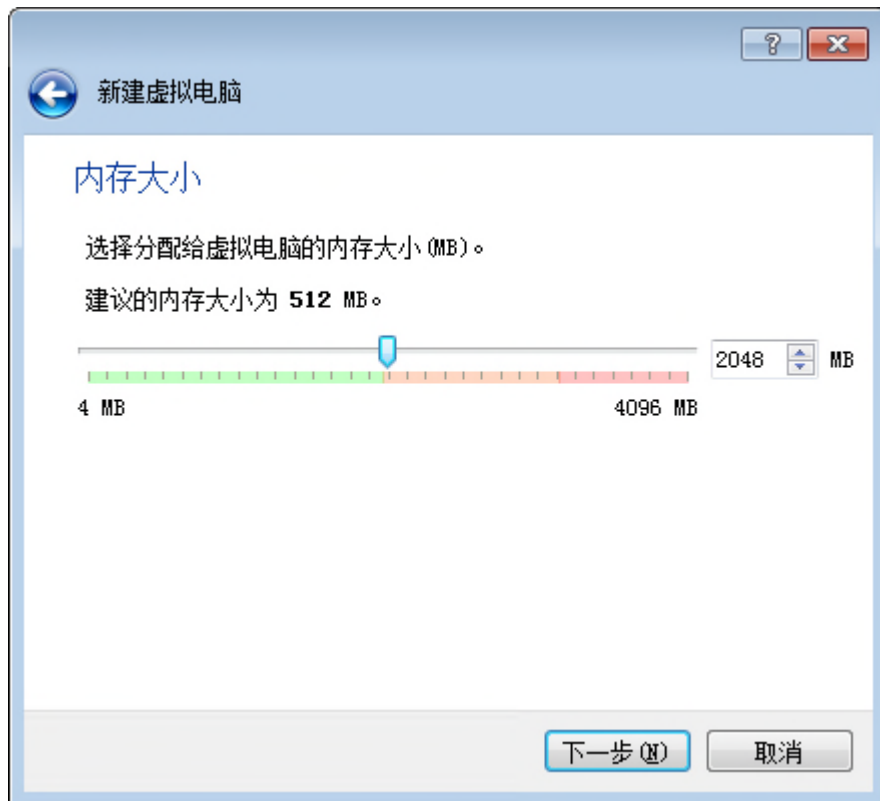
### （一）CentOS：新建虚拟机：

以 master 机为例，slave1 和 slave2 相同

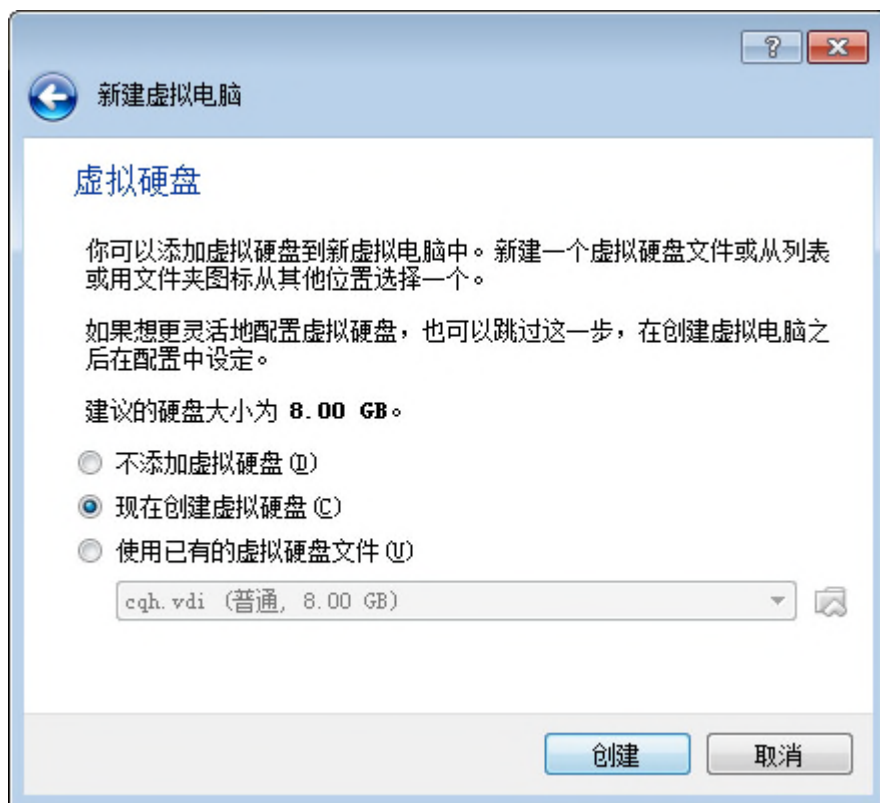
1.选择 linux 以及 other linux -bit



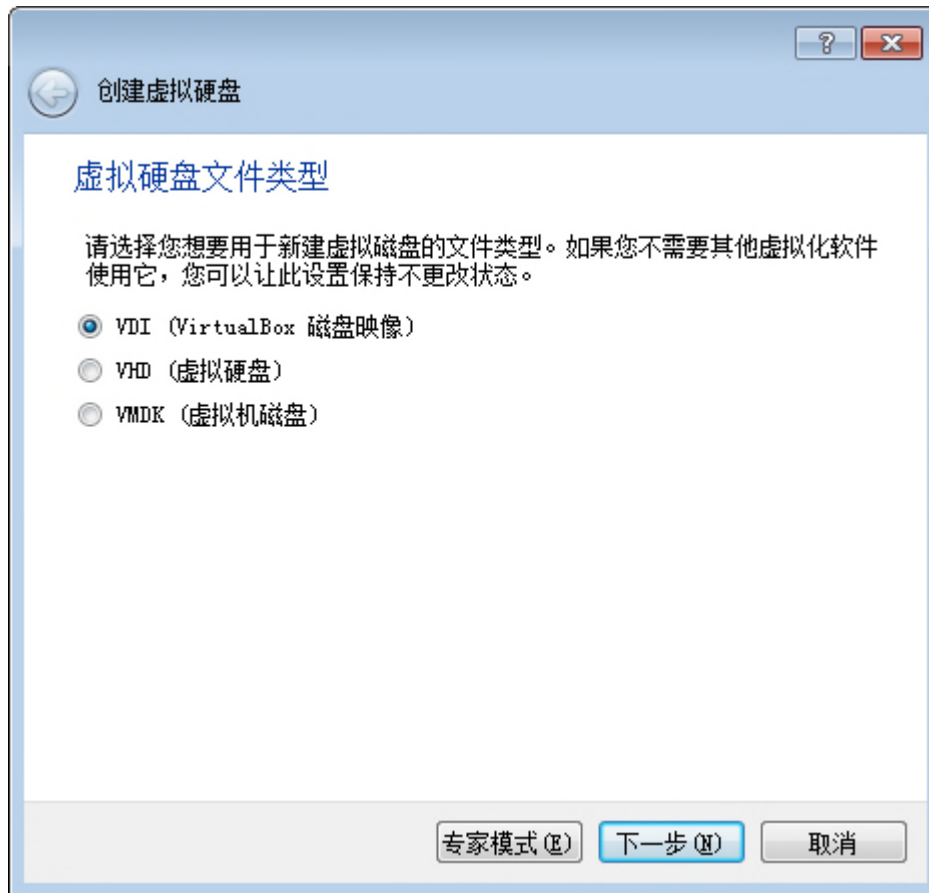
2.分配 2g 内存：



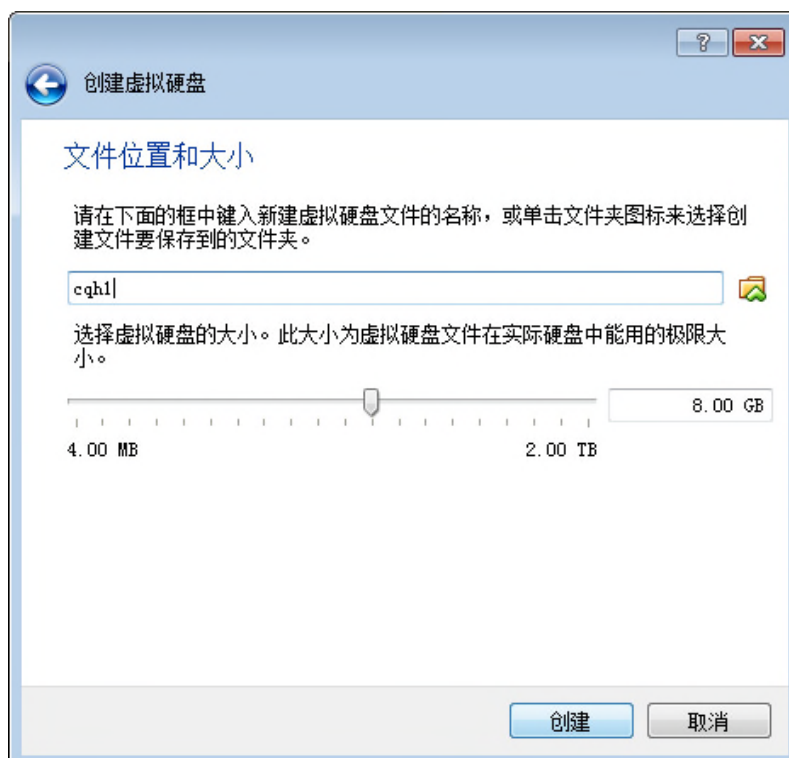
3.



4.



5.

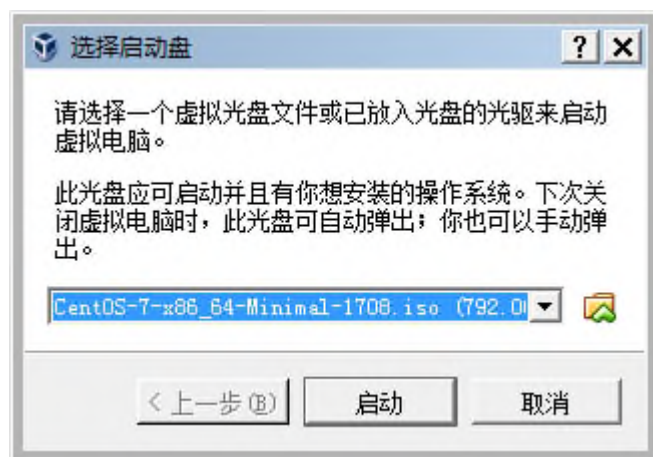




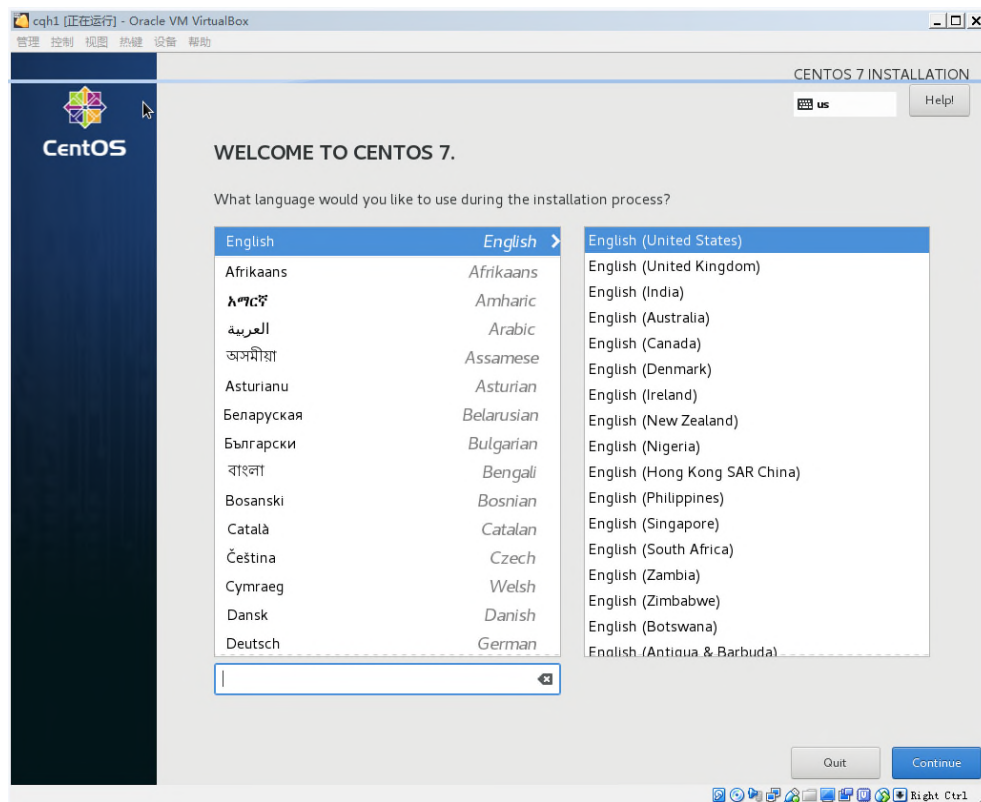
## 6.创建完成



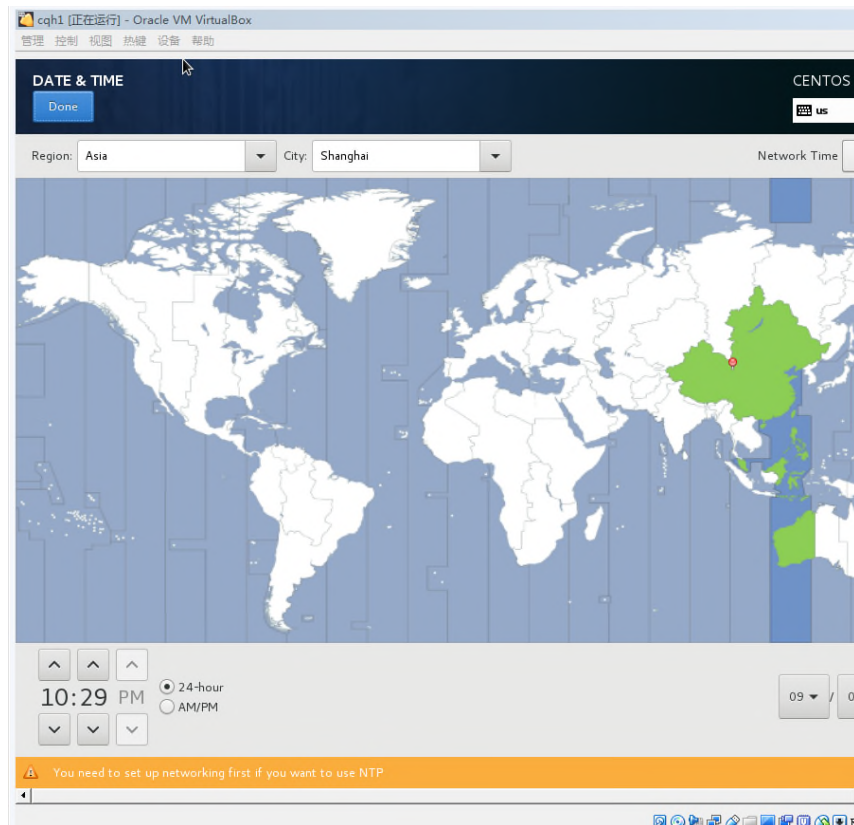
## 7.启动选择相应的 iso 盘:



## 8.选择英语:

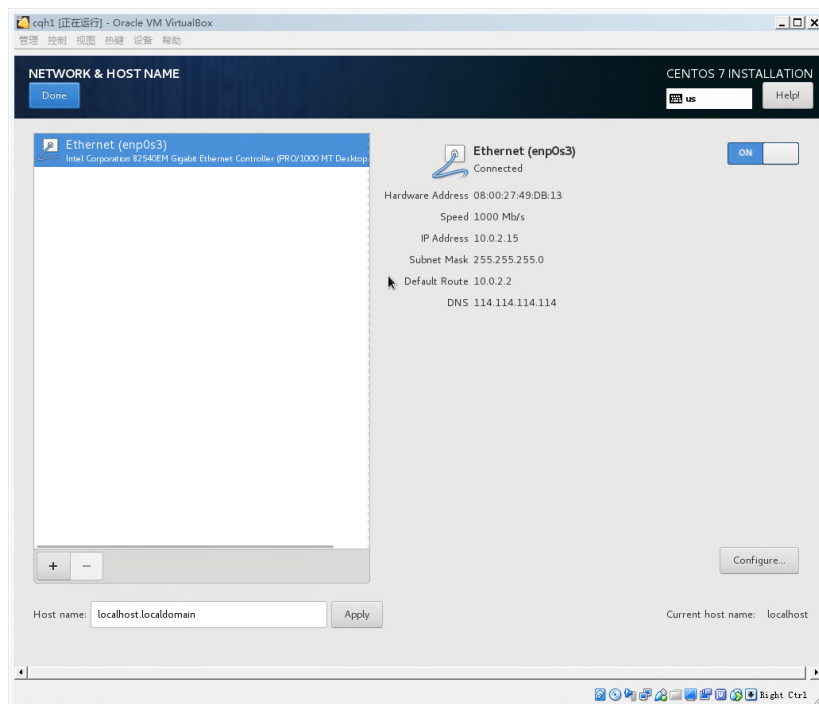


## 9.时区选择上海:

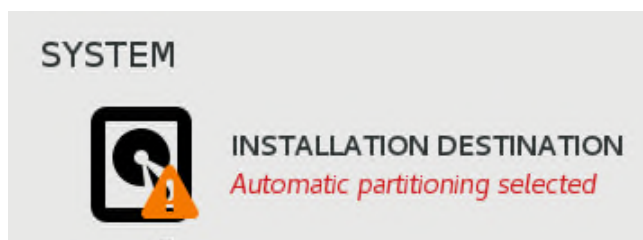




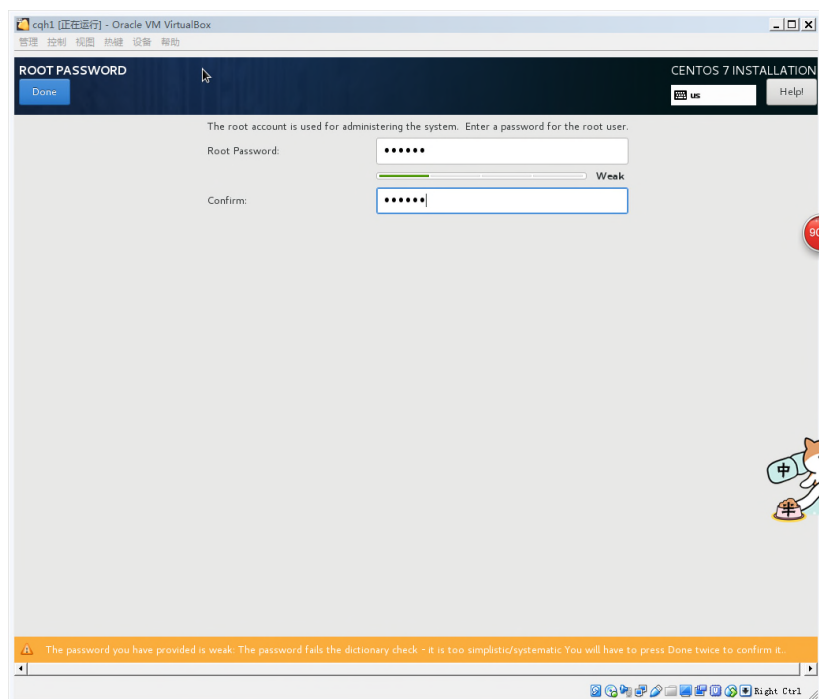
## 10.网络连接打开:



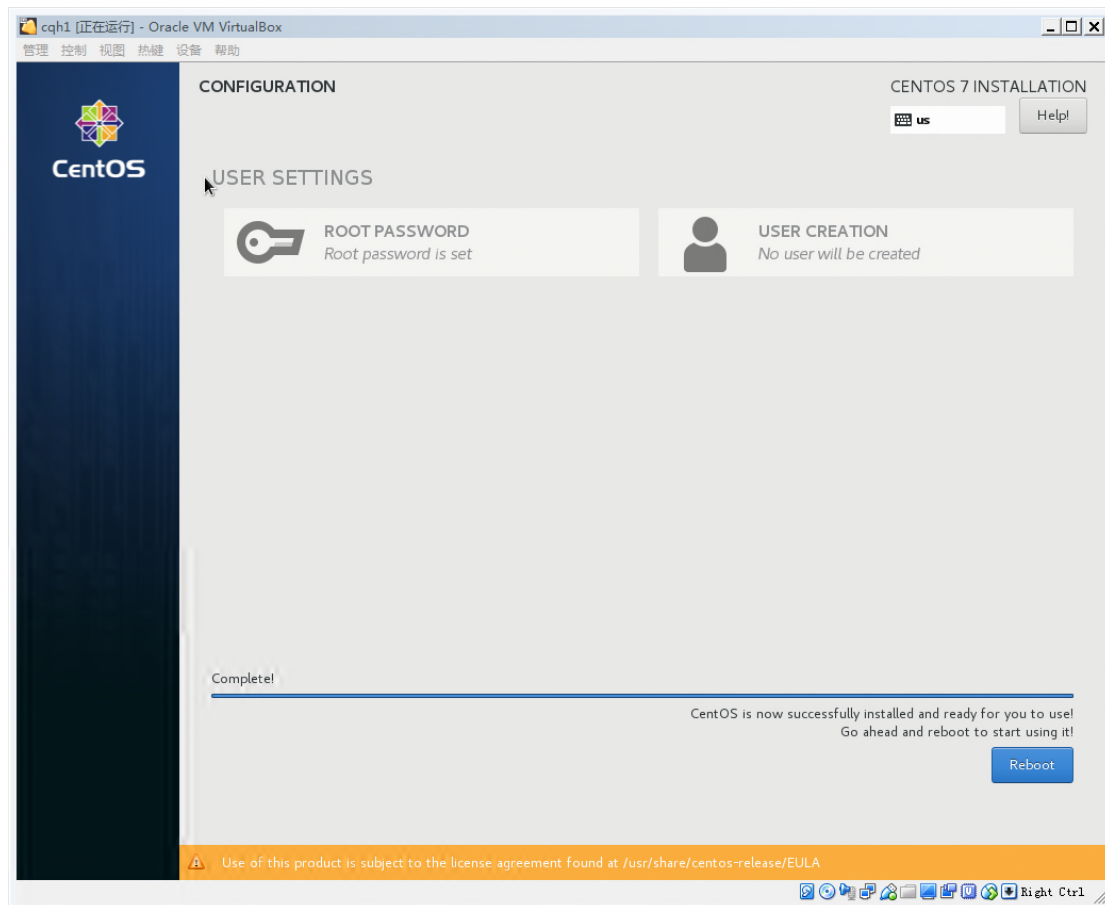
## 11.该选项点进去后点击 done，然后点击 install



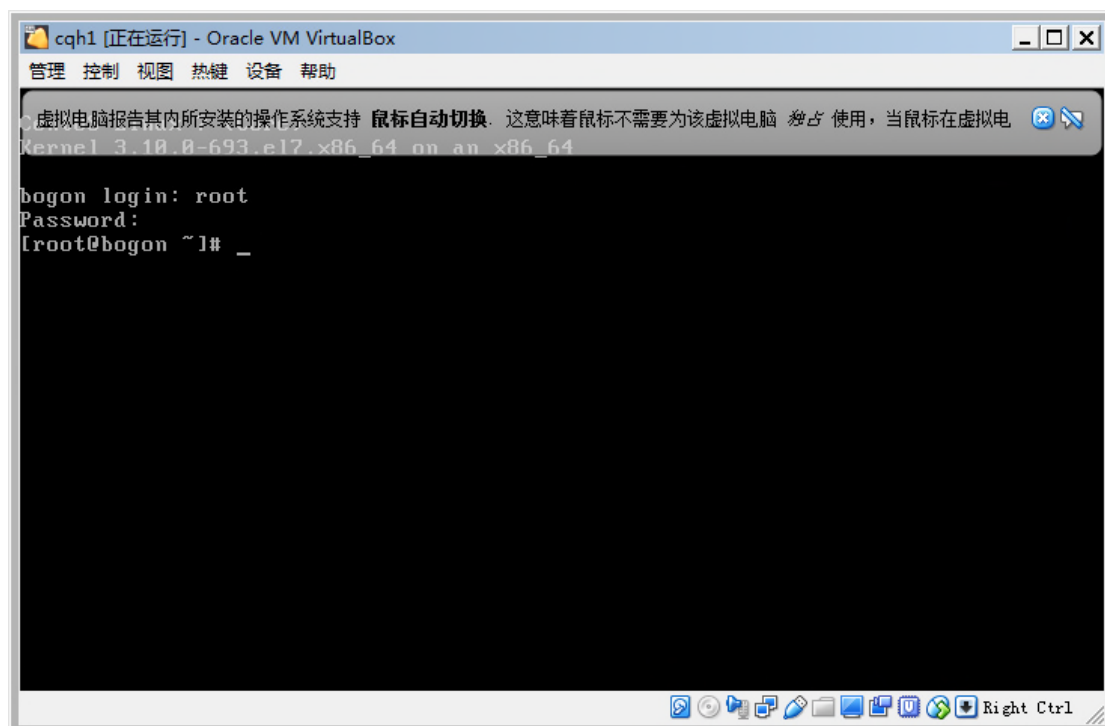
## 12.设置密码



### 13.然后点击安装即可



### 14.成功登陆:



15.进行配置:

(1) .yum install net-tools 进行下载 ifconfig

(2) .把本机公钥送入到目标机器的 authorized\_keys 文件中

```
ssh root@192.168.4.222 'mkdir -p .ssh && cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

```
scp jdk-8u144-linux-x64.tar.gz root@192.168.4.222:~/.
```

(3) 设置 JAVA\_HOME 设置 path 路径

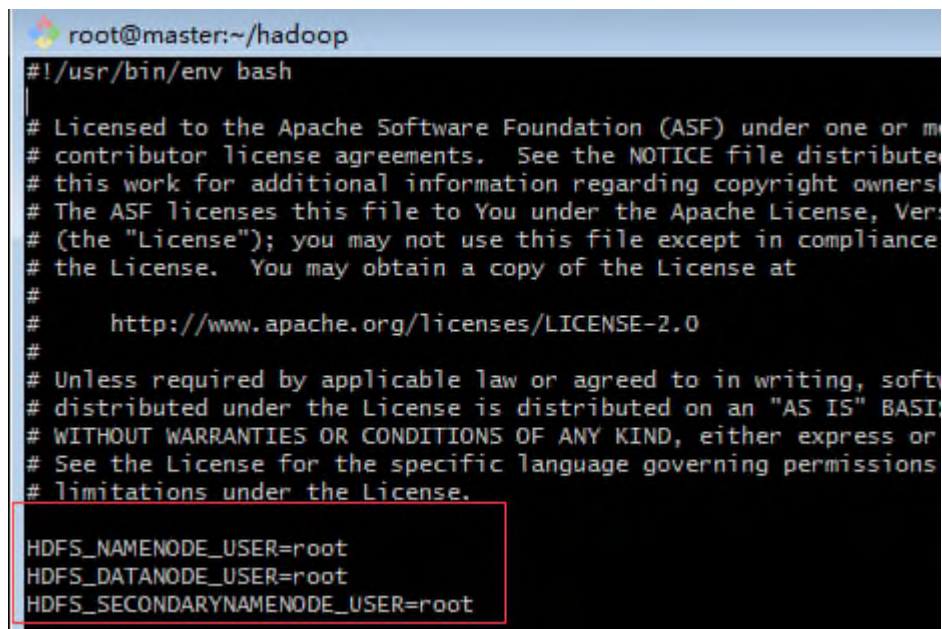
(4) 在每个节点建立目录:

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

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

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

```
[root@master hadoop]# vi sbin/start-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).修改/bin/hdfs 中的 stop-dfs.sh 文件

```
[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)
# contributor license agreements. See the NOTICE
# this work for additional information regarding c
# The ASF licenses this file to You under the Apac
# (the "License"); you may not use this file except
# the License. You may obtain a copy of the Licen
#
# http://www.apache.org/licenses/LICENSE-2.0
#
# Unless required by applicable law or agreed to i
# distributed under the License is distributed on
# WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, ei
# See the License for the specific language govern
# limitations under the License.

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

(7).master 启动 hadoop:

先格式化目录

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

```
2018-09-03 09:16:06,368 INFO namenode.NNStorageRetentionManager: C
s with txid >= 0
2018-09-03 09:16:06,375 INFO namenode.NameNode: SHUTDOWN_MSG:
/*****
SHUTDOWN_MSG: Shutting down NameNode at master/127.0.0.1
*****/
```

启动:

```
*****/
[root@master hadoop]# sbin/start-dfs.sh
Starting namenodes on [localhost]
Last login: Mon Sep  3 09:04:35 CST 2018 from 192.168.4.111 on pts/3
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) 查看启动结果:

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

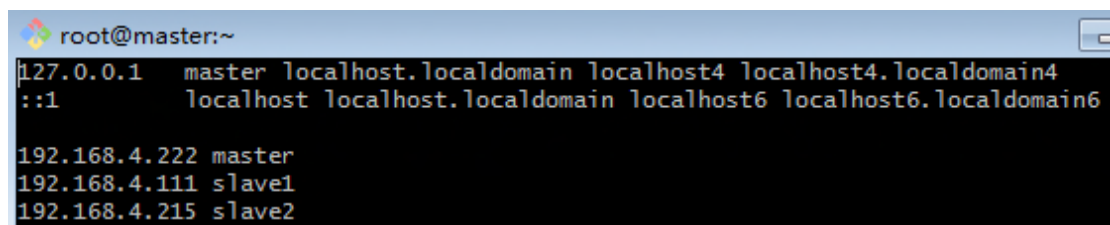
## (二) 联机操作：

### 1. 网卡设置成桥接网卡

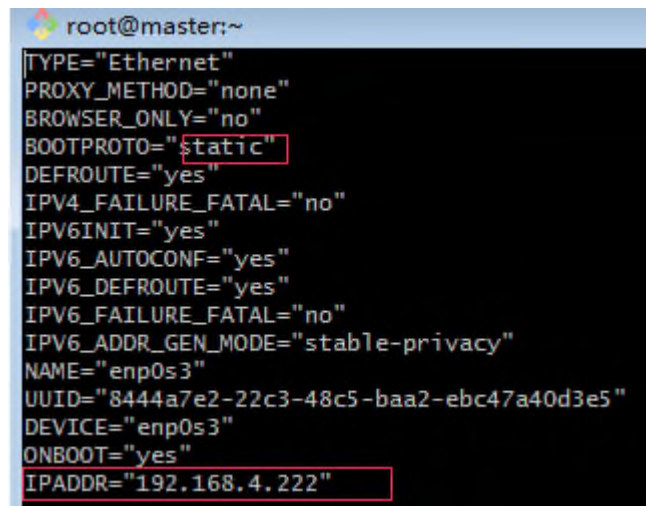


### 2.

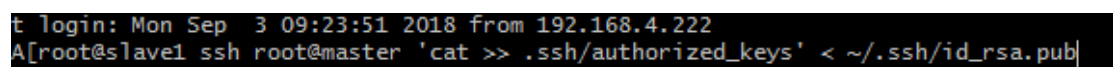
将三台机器中 /etc/Hosts 中角色 ip 改成如下：



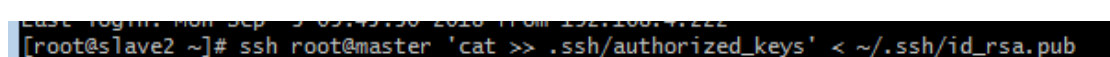
### 3. 设置静态 ip



### 4. slave1 中的公钥放入 master 中的 authorized\_key 中



### Slave2 中的公钥放入 master 中的 authorized\_key 中



5.将 master 中的公钥放入 slave1 和 slave2 中的 authorized\_key 中

```
Last login: Mon Sep  3 09:50:50 2018 from 192.168.4.111  
[root@master ~]# ssh root@slave1 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

```
Last login: Mon Sep  3 09:50:50 2018 from 192.168.4.111  
[root@master ~]# ssh root@slave2 'cat >> .ssh/authorized_keys' < ~/.ssh/id_rsa.pub
```

6. 配置结果 master 和 slave1 之间以及 master 和 slave2 可以不用密码进行访问

Master 和 slave1

```
Last login: Mon Sep  3 09:56:58 2018 from 192.168.4.213  
[root@master ~]# ssh slave1  
Last login: Mon Sep  3 09:44:23 2018 from 192.168.4.222  
[root@slave1 ~]#
```

```
Last login: Mon Sep  3 09:47:25 2018 from 192.168.4.222  
[root@slave1 ~]# ssh master  
Last login: Mon Sep  3 09:57:14 2018 from 127.0.0.1  
[root@master ~]#
```

Master 和 slave2

```
[root@master ~]# ssh slave2  
Last login: Mon Sep  3 09:54:17 2018 from 192.168.4.222  
[root@slave2 ~]#
```

```
Last login: Mon Sep  3 09:54:17 2018 from 192.168.4.222  
[root@slave2 ~]# ssh master  
Last login: Mon Sep  3 10:00:52 2018 from 192.168.4.111  
[root@master ~]# |
```

配置完成。



## 三、数据分析

### (一)、数据分析代码：

由于数据格式是：

147 仙游 2017-01-01 多云 多云 26℃ 15℃ 无持续风向 ≤3 级 无持续风向 ≤3 级 01 01

得到文档中的数据，然后再细分成我们需要的数据。

那么 2017-01-01 这天的数据就是 citiname=仙游，datetime=2017-01，sb=26,sb1=15

而且因为 KEY 是 ‘仙游--2017-01’，所以这个 KEY 就会有 31 个 VALUES 值，VALUES 值的格式是 “26-15” 这样的。

```
@Override
protected void map(LongWritable key, Text value, Mapper<LongWritable, Text, Text, Text>.Context context)
    throws IOException, InterruptedException {
    String line = value.toString();//得到这一行的数据
    String[] words = line.split(" ");//将这行的数据按照空格截断
    String id = words[0];
    String cityname = words[1];
    String datetime = StringUtils.substringBeforeLast(words[2], "-");//得到月份
    String temperture = words[5];
    String sb=StringUtils.substringBefore(temperture, "℃");//得到温度的数值
    String sb1=StringUtils.substringBefore(words[6], "℃");
    //int numtemperture =Integer.parseInt(sb);
    context.write(new Text(cityname+"--"+datetime), new Text(sb+"-"+sb1));|
    //context.write(new Text(id+"--"+id), new Text(id));
    //context.write(new Text(v), new IntWritable(temperture));
}
```

之后将传过来的数据进行处理：

因为 VALUES 值的格式是 26-15 这样的，所以 words[0]就是白天的温度，words[1]就是夜间的温度，KEY 是 城市名--yyyy-mm

那么得到的结果就会是例如：

仙游--2017-01      白天平均温度是 19℃                      夜间平均温度是 11℃      该月的最高温度是 27℃  
最低温度是 18℃

```
Integer sum=0;
Integer nisum=0;String s=null;
int avgtemperture=0;
int niavgtemperture=0;
int max_day_tmp=0;
int max_night_tem=0;
Text t=null;
int i=0;
```



## 结果展示

```
[root@master hadoop]# bin/hdfs dfs -cat /weather/output2/*
仙游--2017-01 白天平均温度是19℃ 夜间平均温度是11℃ 这个月白天的最高温度为27℃ 这个月夜间天的最高温度为17℃
仙游--2017-02 白天平均温度是18℃ 夜间平均温度是9℃ 这个月白天的最高温度为27℃ 这个月夜间天的最高温度为14℃
仙游--2017-03 白天平均温度是19℃ 夜间平均温度是11℃ 这个月白天的最高温度为25℃ 这个月夜间天的最高温度为17℃
仙游--2017-04 白天平均温度是25℃ 夜间平均温度是16℃ 这个月白天的最高温度为33℃ 这个月夜间天的最高温度为22℃
仙游--2017-05 白天平均温度是29℃ 夜间平均温度是21℃ 这个月白天的最高温度为34℃ 这个月夜间天的最高温度为25℃
仙游--2017-06 白天平均温度是30℃ 夜间平均温度是24℃ 这个月白天的最高温度为35℃ 这个月夜间天的最高温度为26℃
仙游--2017-07 白天平均温度是35℃ 夜间平均温度是25℃ 这个月白天的最高温度为38℃ 这个月夜间天的最高温度为27℃
仙游--2017-08 白天平均温度是34℃ 夜间平均温度是26℃ 这个月白天的最高温度为37℃ 这个月夜间天的最高温度为27℃
福州--2017-01 白天平均温度是16℃ 夜间平均温度是10℃ 这个月白天的最高温度为25℃ 这个月夜间天的最高温度为17℃
福州--2017-02 白天平均温度是16℃ 夜间平均温度是8℃ 这个月白天的最高温度为25℃ 这个月夜间天的最高温度为13℃
福州--2017-03 白天平均温度是18℃ 夜间平均温度是11℃ 这个月白天的最高温度为24℃ 这个月夜间天的最高温度为16℃
福州--2017-04 白天平均温度是24℃ 夜间平均温度是16℃ 这个月白天的最高温度为30℃ 这个月夜间天的最高温度为21℃
福州--2017-05 白天平均温度是28℃ 夜间平均温度是20℃ 这个月白天的最高温度为32℃ 这个月夜间天的最高温度为24℃
福州--2017-06 白天平均温度是29℃ 夜间平均温度是23℃ 这个月白天的最高温度为36℃ 这个月夜间天的最高温度为26℃
福州--2017-07 白天平均温度是35℃ 夜间平均温度是26℃ 这个月白天的最高温度为39℃ 这个月夜间天的最高温度为27℃
福州--2017-08 白天平均温度是34℃ 夜间平均温度是26℃ 这个月白天的最高温度为37℃ 这个月夜间天的最高温度为27℃
[root@master hadoop]#
```