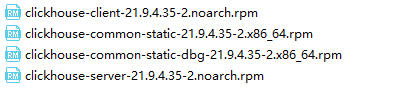
ClickHouse单机版安装配置

<https://www.pudn.com/news/6336be4c2aaf6043c9d25450.html>

1. 准备好安装包并且上传至Linux中



1. Tgz压缩包方式安装：

tar -xzvf "clickhouse-common-static-22.3.2.2.tgz"

tar -xzvf "clickhouse-common-static-dbg-22.3.2.2.tgz"

tar -xzvf "clickhouse-server-22.3.2.2.tgz"

tar -xzvf "clickhouse-client-22.3.2.2.tgz"

进入解压后的目录，分别执行目录下的脚本进行安装：

不一定按顺序，但是需要注意的是，前三个脚本无输出，server的安装脚本会要求输入几个参数，第一个是默认用户的密码，第二个是是否允许远程链接，这里我选y允许远程连接。（不同版本第二个可能不出现）

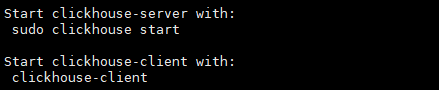
clickhouse-common-static-22.3.2.2/install/doinst.sh

clickhouse-common-static-dbg-22.3.2.2/install/doinst.sh

clickhouse-server-22.3.2.2/install/doinst.sh

clickhouse-client-22.3.2.2/install/doinst.sh

安装完毕后提示启动命令如下：



openssl req -subj "/CN=localhost" -new -newkey rsa:2048 -days 3650 -nodes -x509 -keyout /etc/clickhouse-servee-server/server.crt

"<clickhouse><listen\_host>0.0.0.0</listen\_host></clickhouse>" > /etc/clickhouse-server/config.d/listen.xml

chmod 777 sudo vim /homehadoop/software/install.sh

/homehadoop/software/install.sh

说明，安装完毕后默认路径是在 /etc/clickhouse-server下相关目录，包括配置也是在/etc/clickhouse-server/config.xml中

1. 检测是否安装完毕：
   1. 启动 clickhouse server

systemctl start clickhouse

systemctl start clickhouse-server.service

查看状态：

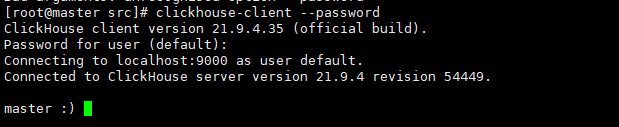
systemctl statusclickhouse

启动时最好采用下面方式启动：

clickhouse-server --config-file=/etc/clickhouse-server/config.xml

* 1. server 成功启动后，使用 clickhouse-client 配合在安装时指定的 password 进行登录

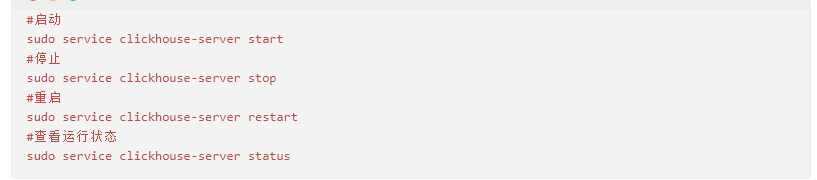
clickhouse-client –password



clickhouse-client --user 账户名称 --password 密码

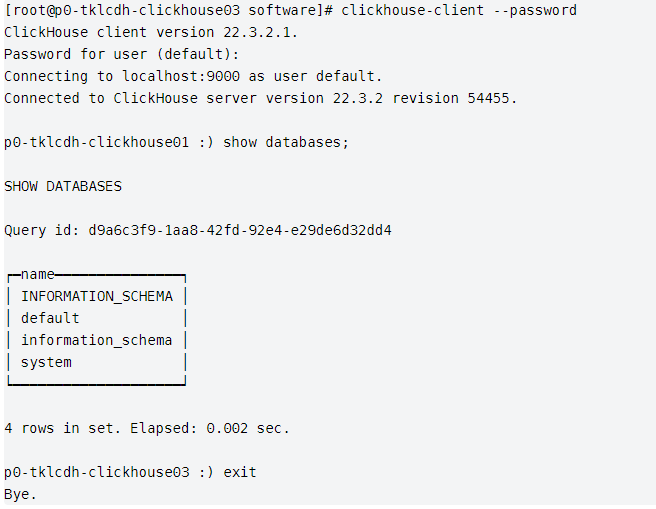
clickhouse-client –m 客户端连接



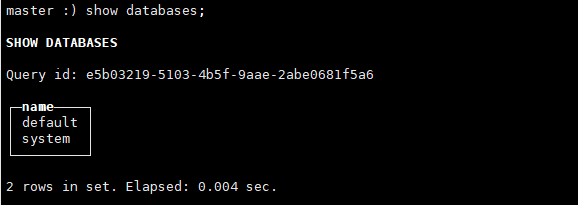




* 1. 观测到如下输出则可表明 Clickhouse 在单机上运行成功



查看数据库



* 1. 查看clickhouse运行状态

systemctl status clickhouse-server

在其他安装了clickhouse-client机器上链接命令是：

在安装了clickhouse client的服务器上远程访问：clickhouse-client -h 192.168.132.14 --port 9000 -u default --password abcd1234

1. 修改配置（视情况而定，是否需要修改）

vi /etc/clickhouse-server/config.xml

* 1. 修改远程访问

<!-- <listen\_host>::</listen\_host> -->

将注释去掉才能让除本机外的clickhouse访问

* 1. 日志配置也在/etc/clickhouse-server/config.xml这个文件中，可以修改日志级别和相关存储目录（一般无需修改）



* 1. 修改时区

修改时区，找到timezone标签，将内容修改为Asia/Shanghai

* 1. 修改默认端口9000为其他（因为和hadoop冲突）

设置远程访问并移除默认监听文件（listen.xml），同时由于9000端口被hadoop占用，需要将clickhouse的端口更改为9001

<tcp\_port>9001</tcp\_port>

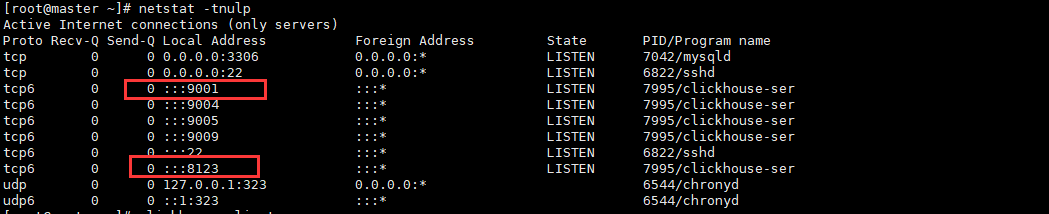
**注意：修改端口后的链接命令为 clickhouse-client –m - -port 9001**

修改配置后启动方式使用：

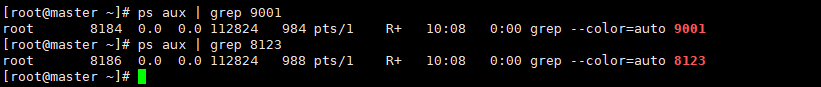
clickhouse-server --config-file=/etc/clickhouse-server/config.xml

查看占用进程命令，可以看到9001和8123端口情况，此时可以正常登录

netstat -tnulp



单独查看两个端口的占用情况：



* 1. 错误说明
     1. 如果出现下列错误



说明开启了两次clickhouse-server，解决：

ps -ef | grep clickhouse-server

#删除进程

kill -9 307840

1. 创建数据库、表
   1. 创建数据库：create database XXXX;
   2. 使用数据库：use xxxx;
   3. 创建表：

Create table if not exist mytable1(user\_id UInt64 default 0,user\_name String,user\_sex String,user\_addr String,register\_date Date) engine=MergeTree()

order by (user\_id)

primary key(user\_id)

partition by(date);

使用show create table 表名 查看创建表的语句

* 1. 插入测试数据：
  2. 查询数据：

1. Spark操作clickhouse准备
   1. Pom.xml中添加依赖

|  |
| --- |
| <dependency>  <groupId>ru.yandex.clickhouse</groupId>  <artifactId>clickhouse-jdbc</artifactId>  <version>0.3.1</version>  </dependency> |

* 1. 代码中初始化配置：

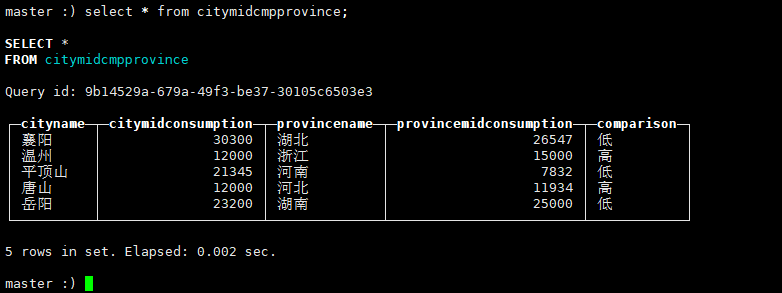
|  |
| --- |
| val sparkSession = SparkSession.builder().master("local").appName("Test1").getOrCreate()  import sparkSession.implicits.\_  val prop = new Properties()  prop.setProperty("user","default")  prop.setProperty("password","")  prop.setProperty("driver","ru.yandex.clickhouse.ClickHouseDriver") |

1. Spark读取clickhouse数据

|  |
| --- |
| //读取数据测试  val df1 = sparkSession.read.jdbc("jdbc:clickhouse://192.168.44.51:8123/shtd\_test1","citymidcmpprovince",prop)  df1.show() |

1. Spark将数据写入clickhouse

|  |
| --- |
| //写入数据测试  val df2 = Seq(  ("襄阳",30300,"湖北",26547,"低"),  ("温州",12000,"浙江",15000,"高"),  ("平顶山",21345,"河南",7832,"低"),  ("唐山",12000,"河北",11934,"高"),  ("岳阳",23200,"湖南",25000,"低")  ).toDF("cityname","citymidconsumption","provincename","provincemidconsumption","comparison")  //比赛时上面的数据是通过spark指标计算的出来的DataFrame  df2.write.mode(SaveMode.Append).option("numPartitions",1).option("isolationLevel","NONE").jdbc("jdbc:clickhouse://192.168.44.51:8123/shtd\_test1","citymidcmpprovince",prop)  //注意：上面黄色背景的内容不是必须的 |



降序排序取Top3：

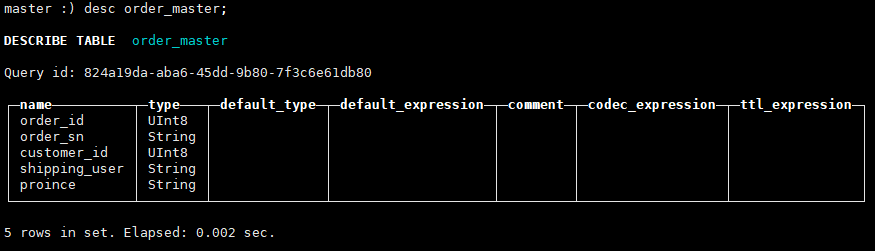


1. Flink数据写入clickhouse
   1. Pom.xml依赖，Flink 1.11.x之后版本使用flink-connector-jdbc，只支持DataStream API

|  |
| --- |
| <!-- Flink1.11 后需要 Flink-client包-->  <dependency>  <groupId>org.apache.flink</groupId>  <artifactId>flink-clients\_2.11</artifactId>  <version>1.11.3</version>  </dependency>  <!--添加 Flink Table API 相关的依赖 -->  <dependency>  <groupId>org.apache.flink</groupId>  <artifactId>flink-table-planner-blink\_2.11</artifactId>  <version>1.11.3</version>  </dependency>  <dependency>  <groupId>org.apache.flink</groupId>  <artifactId>flink-table-api-scala-bridge\_2.11</artifactId>  <version>1.11.3</version>  </dependency>  <dependency>  <groupId>org.apache.flink</groupId>  <artifactId>flink-table-common</artifactId>  <version>1.11.3</version>  </dependency>  <!--添加 Flink JDBC Connector 以及 Clickhouse JDBC Driver 相关的依赖 -->  <dependency>  <groupId>org.apache.flink</groupId>  <artifactId>flink-connector-jdbc\_2.11</artifactId>  <version>1.11.3</version>  </dependency>  <dependency>  <groupId>ru.yandex.clickhouse</groupId>  <artifactId>clickhouse-jdbc</artifactId>  <version>0.2.4</version>  </dependency> |

* 1. 准备创建数据表：

|  |
| --- |
| create table order\_master(order\_id UInt8,order\_sn String,customer\_id UInt8,shipping\_user String,proince String) ENGINE=Memory; |



* 1. Flink中核心代码部分

|  |
| --- |
| //构建sql语句  val sql = "insert into order\_master(order\_id,order\_sn,customer\_id,shipping\_user,proince) values(?,?,?,?,?)"  //构建存储sink对象,放入sql，实现处理sql中参数值方法accept，定义处理参数设置，设置连接参数设置  val mysink = JdbcSink.sink(sql,new JdbcStatementBuilder[(String,String,String,String,String)] {  override def accept(t: PreparedStatement, u: (String, String, String, String, String)): Unit = {  t.setInt(1,u.\_1.toInt)  t.setString(2,u.\_2)  t.setInt(3,u.\_3.toInt)  t.setString(4,u.\_4)  t.setString(5,u.\_5)  }  },new JdbcExecutionOptions.Builder().withBatchSize(1).build()  ,new JdbcConnectionOptions.JdbcConnectionOptionsBuilder().withDriverName("ru.yandex.clickhouse.ClickHouseDriver")  .withUrl("jdbc:clickhouse://192.168.44.51:8123/shtd\_test1")  .withUsername("default").withPassword("").build())  //添加数据存储  dataStream.print()  dataStream.addSink(mysink) |