

弹性 MapReduce

快速入门

产品文档





【版权声明】

©2013-2025 腾讯云版权所有

本文档著作权归腾讯云单独所有,未经腾讯云事先书面许可,任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

【商标声明】



及其他腾讯云服务相关的商标均为腾讯集团下的相关公司主体所有。另外,本文档涉及的第三方主体的商标,依法 由权利人所有。

【服务声明】

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况,部分产品、服务的内容可能有所调整。您 所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则, 腾讯云对本文档内容不做任何明示或默示的承诺或保证。



文档目录

快速入门 EMR on CVM 快速入门 EMR on TKE 快速入门



快速入门 EMR on CVM 快速入门

最近更新时间:2024-10-30 10:00:07

本文为您介绍通过 EMR 控制台快速创建一个 EMR on CVM 集群、提交作业并查看运行结果的操作流程。

准备工作

1. 在使用 EMR 集群前, 需要注册腾讯云账号并完成实名认证, 具体操作请参见 企业实名认证指引。

2. 完成对弹性 MapReduce 的服务账号授予系统默认角色 EMR_QCSRole,具体操作请参见角色授权。

3. 在线账号充值, EMR on CVM 提供两种计费模式:按量计费和包年包月计费,在创建集群前需要进行账号余额充值,确保余额大于等于创建集群所需配置费用(不包含:代金券等);具体操作请参见考充值流程说明。

创建集群

登录 EMR 控制台,在 EMR on CVM 集群列表页单击**创建集群**,在购买页面完成相关配置;当集群列表中集群状态显示为运行中时,表示集群创建成功。

| 购买步 骤 | 配置项 | 配置项说明 | 示例 | | |
|----------|---------------|---|--|--|--|
| 软件配 置 | 地域 | 集群所部署的物理数据中心 注意:集群创建后,无法更改地域,请谨慎选 择 | 北京、上海、广州、南京、成都、 硅谷等 | | |
| | 集群类型 | EMR on CVM 支持多种集群类,默认 Hadoop 集群类型 | Hadoop、StarRocks 等 | | |
| | 产品版本 | 不同产品版本上捆绑的组件和组件的版本不同 | EMR-V2.7.0 版本中内置的是 Hadoop 2.8.5、Spark 3.2.1 等 | | |
| | 部署组件 | 非必选组件,根据自身需求组合搭配自定义部 署 | Hive-2.3.9、Impala-3.4.1等 | | |
| 区域与 | 计费模式 集群部署计费模式 | | 按量计费 | | |
| 硬件 置 | 可用区及 网络配置 | 可用区、集群网络设置注意:集群创建后,无 法直接更改可用区,请谨慎选择 | 广州七区 | | |
| | 安全登录 | 用于设置节点的网络访问控制,安全组同防火 | 创建新安全组 | | |



| | | 墙功能 | |
|----------|------|---|-----------------|
| | 节点配置 | 根据业务需要为不同节点类型选择合适机型配 置。详情请参见 业务评估 | 开启节点部署高可用 |
| | 所属项目 | 将当前集群分配给不同的项目组 | 集群创建后暂不支持修改所属项目 |
| 基础配 置 | 集群名称 | 集群的名称,可自定义 | EMR-7sx2aqmu |
| | 登录方式 | 自定义设置密码方式和关联密钥方式;SSH 密钥仅用于 EMR-UI 快捷入口登录 | 密码 |
| 确认配 置 | 配置清单 | 确认所部署信息是否有误 | 选中服务协议, 单击立即购买 |

注意

您可以在 CVM 控制台中查看各节点信息,为保证 EMR 集群的正常运行,请勿在 CVM 控制台中更改节点配置信息。

提交作业及查看运行结果

集群创建成功后,您可以在该集群创建并提交作业;本文已提交 spark 任务为例,操作如下。

注意

在创建 EMR 集群的时候需要在软件配置界面选择 Spark 组件。

1. 使用 SSH 登录并连接集群(本地系统为 Linux/Mac OS),详情请参见 登录集群。

2. 在 EMR 命令行先使用以下指令切换到 Hadoop 用户,并进入 Spark 安装目录/usr/local/service/spark:

[root@172 ~]# su hadoop
[hadoop@172 root]\$ cd /usr/local/service/spark

3. 通过如下指令提交任务并运行:

```
/usr/local/service/spark/bin/spark-submit \\
--class org.apache.spark.examples.SparkPi \\
--master yarn \\
--deploy-mode cluster \\
--proxy-user hadoop \\
--driver-memory 1g \\
--executor-memory 1g \\
/usr/local/service/spark/examples/jars/spark-examples*.jar \\
10
```



4. 提交作业后,在 EMR on CVM 页面,单击目标集群所在行的**集群服务**;单击 YARN UI 所在行的 WebUI 链接。登录认证后即可进入YARN UI 页面;单击目标作业的 ID,可以查看作业运行的详情。

销毁集群

当创建的集群不再使用时,可以销毁集群,退还资源;毁集群将强制终止集群所提供的服务,并释放资源。 在 EMR on CVM 页面,选择目标集群的**更多 > 销毁**;在弹出的对话框中,单击**立即销毁**。



EMR on TKE 快速入门

最近更新时间:2024-10-30 10:02:06

本文为您介绍通过 EMR 控制台快速创建一个 EMR on TKE 集群、并完成作业提交及查看运行结果的整套操作流程。

准备工作

1. 在使用 EMR 集群前,需要注册腾讯云账号并完成实名认证,具体操作请参见 企业实名认证指引。

2. 完成对弹性 MapReduce 的服务账号授予系统默认角色 EMR_QCSRole,具体操作请参见角色授权。

3. 完成对弹性 MapReduce的服务账号授权服务相关角色,具体操作请参见管理权限。

4. 在线账号充值, EMR on TKE 提供按量计费, 在创建集群前需要进行账号余额充值, 确保余额大于等于创建集群 所需配置费用(不包含:代金券等), 具体操作请参见 充值流程说明。

创建集群

登录 EMR 控制台,在 EMR on TKE 集群列表页单击**创建集群**,在购买页面完成相关配置;当集群状态显示为运行中时,表示集群创建成功。

| 配置项 | 配置项说明 | 示例 |
|-------------|--|--|
| 集群名称 | 集群的名称,可自定义 | EMR-7sx2aqmu |
| 地域 | 集群所部署的物理数据中心 注意:集群创建后,无法更改地域,请 谨慎选择。 | 北京、上海、广州、南京、新加坡等 |
| 容器类型 | 服务角色由底层容器提供资源进行部署,支持 TKE 标准集群和 TKE Serverless 集群 | ТКЕ |
| 集群网络 及子网 | 用于购买 db 使用,需保持网络与容器 集群网络一致 | 广州七区 |
| 安全组 | 集群维度配置安全组 | 创建新安全组 |
| 计费模式 | 集群部署计费模式 | 按量计费 |
| 产品版本 | 不同产品版本上捆绑的组件和组件的版本不同 | EMR-TKE1.0.0 版本中内置的是 Hadoop 2.8.5、 Spark 3.2.1 等。 |
| 部署服务 | 非必选组件,根据自身需求组合搭配自 | Hive-2.3.9、Impala-3.4.1等 |



| | 定义部署, 最少选择一个组件 | |
|-------------|---|---|
| COS 存储 桶 | 用于存储日志, jar 包等信息 | - |
| 设置密码 | 设置 webUI 密码,当前密码仅用于初始 设置服务 webUI 访问密码。 | 8-16个字符,包含大写字母、小写字母、数字和特殊 字符四种,特殊符号仅支持!@%^*,密码第一位不 能为特殊字符 |

提交作业及查看运行结果

集群创建成功后,您可以在该集群创建并提交作业。本文以提交 Kyuubi Spark 和 Hive on Spark 作业和查看作业信息为例,操作如下。

Hue 提交

1. 在集群列表中单击对应的集群 ID/名称进入集群详情页。

2. 在集群详情页中单击**集群服务**,选择 hue。

3. 在角色管理页面的操作列中打开**更多**下拉选项框,单击**开启网络访问**,然后选择**公网LB**,单击**确认开启**,待流程 结束后, hue 所在 pod 的公网 LB 成功创建。

4. 单击右上角**查看信息/查看WebUI**,查看 HUE 的访问地址,单击**访问 hue的WebUI**。

5. 通过认证进入 hue 页面,通常认证用户为 root,密码为集群创建时的密码。

6. 通过 hive tab 可以提交 hive on spark 任务。

7. 通过 SparkSql_Kyuubi 可以提交 sparksql 任务。

Hive on Spark 建表及查询:



| | | Q Search saved | documents | |
|-------------|---------|---|---|--|
| ┋ 4 ₩ | 9 | 🖗 Hive 🔊 | Add a name | Add a description |
| <pre></pre> | (1) + 2 | 1 create databas 2 create table t 3 insert into te 3/3 | e test_db locatio est_db.test (id i st_db.test values | <pre>1m, 10s test_db * * ? n 'cosn://Jean-down //Jean-down //Jean-</pre> |
| | | Query History 2 分钟前 | Saved Querie | s ert into test_db.test values(1, 'test1'),(2,'test2') |
| | | 2 分钟前 | S crea | ate table test_db.test (id int, name string) |
| | | 2 分钟前 | S crea | ate database test_db location 'cosn:// |
| | | 7 分钟前 | S show | v tables |

| • | 4 5 sel | lect ' | * from tes | st_db.test; | | | |
|----|------------|--------|------------|---------------|-------------|-----------|--|
| ₩• | | | | | | | |
| | | | | | | | |
| | Que | ery Hi | story | Saved Queries | Results (2) | | |
| | | | test.id | | | test.name | |
| | | 1 | 1 | | | test1 | |
| | | | | | | 4 | |
| | | 2 | 2 | | | testz | |

Kyuubi 查询:



| 4 | 9 | SparkSql_Kyuubi 🕤 Add a name Add a description | | Ø | Tables |
|---------------|-------|---|--------------------------|---|--|
| entries found | (0) 💭 | 1 select * from test_db.test; | 4.66s default ♥ ? | | Filter i test_db.test id name |
| | | Ing Tool's query[u0/Ja405-5177-4004-augu-Juli51102/00]. NuMMING_DTATE -> rinigned_State, time taken. 4.040 Se 2023-06-28 07:12:17.446 INFO org.apache.kyuubi.operation.ExecuteStatement: Query[a304ad47-7dda-4919-902e-267 _STATE 2023-06-28 07:12:17.446 INFO org.apache.kyuubi.operation.ExecuteStatement: Processing root's query[a304ad47- 69fb1]: RUNNING_STATE -> FINISHED_STATE, time taken: 4.649 seconds Query History Saved Queries Q Results (2) | 7dda-4919-902e-2678b17 | | |
| | | id name | | | |
| | | Image: Constraint of the second se | | | |

JDBC 提交 Hive Spark

1. 如果您需要使用外网 IP 连接 hiveserver,请在**集群服务 > Hive > HiveServer2 > 操作 > 更多 > 开启网络访问**中 开启 hiveserver2外网。

 2. 如果使用外网链接,需要到集群信息中查看安全组,前往云服务器 > 安全组中编辑该安全组,为7001端口放开客 户端 IP 的访问,若使用内网访问,则可跳过1,2步骤。

使用 maven 编写 jdbc 代码

```
首先在 pom.xml 中引入以下 jdbc 所需依赖:
```

引入以下打包和编译插件:

<build> <plugins> <plugin>



```
<proupId>org.apache.maven.plugins</proupId>
   <artifactId>maven-compiler-plugin</artifactId>
   <configuration>
     <source>1.8</source>
     <target>1.8</target>
     <encoding>utf-8</encoding>
   </configuration>
 </plugin>
 <plugin>
   <artifactId>maven-assembly-plugin</artifactId>
   <configuration>
     <descriptorRefs>
     <descriptorRef>jar-with-dependencies</descriptorRef>
     </descriptorRefs>
   </configuration>
   <executions>
     <execution>
       <id>make-assembly</id>
       <phase>package</phase>
       <goals>
         <goal>single</goal>
       </goals>
     </execution>
   </executions>
</plugin>
</plugins>
</build>
```

新建 HiveJdbcTest.java 如下:

```
package org.apache.hive;
import java.sql.*;
/**
 * Created by tencent on 2023/6/20.
 */
public class HiveJdbcTest {
 private static String driverName =
       "org.apache.hive.jdbc.HiveDriver";
public static void main(String[] args)
           throws SQLException {
        try {
            Class.forName(driverName);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
            System.exit(1);
```



```
Connection con = DriverManager.getConnection(
            "jdbc:hive2://$hs2host:7001/test db", "hadoop", "");
   Statement stmt = con.createStatement();
   String tableName = "test jdbc";
   stmt.execute("drop table if exists " + tableName);
   stmt.execute("create table " + tableName +
            " (key int, value string)");
   System.out.println("Create table success!");
   // show tables
   String sql = "show tables '" + tableName + "'";
   System.out.println("Running: " + sql);
   ResultSet res = stmt.executeQuery(sql);
   if (res.next()) {
     System.out.println(res.getString(1));
   }
   // describe table
   sql = "describe " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(res.getString(1) + "\\t" + res.getString(2));
   }
   sql = "insert into " + tableName + " values (42, \\"hello\\"),
(48,\\"world\\")";
   stmt.execute(sql);
   sql = "select * from " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(String.valueOf(res.getInt(1)) + "\\t"
             + res.getString(2));
   }
   sql = "select count(1) from " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(res.getString(1));
   }
 }
```

将代码中的 \$hs2host 替换为您的 hiveserver2 地址。

该程序会在 test_db 中创建 test_jdbc 表,写入两条数据并查询输出该数据。执行下面的命令对整个工程打包:

mvn package

}



上传 jar 并运行

将上述命令打包的 jar 上传到可以访问到 hiveserver2 服务的机器或者本地(如果是本地,则保证能正常访问 hiveserver2),通过以下命令运行:

```
java -classpath ${package}-jar-with-dependencies.jar
org.apache.hive.HiveJdbcTest
```

其中 package 为您自定义的 artifactId-version。运行结果如下:

```
Create table success!
Running: show tables 'test_jdbc'
test_jdbc
Running: describe test_jdbc
key int
value string
Running: select * from test_jdbc
42 hello
48 world
Running: select count(1) from test_jdbc
2
```

JDBC 提交 Kyuubi Spark

1. 如果您需要使用外网 IP 连接 KyuubiServer,请在**集群服务 >Kyuubi > KyuubiServer > 操作 > 更多 > 开启网络** 访问中开启 KyuubiServer 外网。

 如果使用外网链接,需要到集群信息中查看安全组,前往云服务器 > 安全组中编辑该安全组,为10009端口放开 客户端 IP 的访问,若使用内网访问,则可跳过1,2步骤。

使用 maven 编写 jdbc 代码

jdbc 依赖和打包插件配置与 JDBC 提交 Hive Spark 中一致。需要创建创建 KyuubiJdbcTest.java,内容如下:

```
package org.apache.hive;
import java.sql.*;
/**
 * Created by tencent on 2023/6/20.
 */
public class KyuubiJdbcTest {
 private static String driverName =
       "org.apache.hive.jdbc.HiveDriver";
public static void main(String[] args)
            throws SQLException {
            try {
```



```
Class.forName(driverName);
   } catch (ClassNotFoundException e) {
     e.printStackTrace();
     System.exit(1);
   }
   Connection con = DriverManager.getConnection(
            "jdbc:hive2://$kyuubihost:10009/test_db", "hadoop", "");
   Statement stmt = con.createStatement();
   String tableName = "test_kyuubi";
   stmt.execute("drop table if exists " + tableName);
   stmt.execute("create table " + tableName +
            " (key int, value string)");
   System.out.println("Create table success!");
   // show tables
   String sql = "show tables '" + tableName + "'";
   System.out.println("Running: " + sql);
   ResultSet res = stmt.executeQuery(sql);
   if (res.next()) {
     System.out.println(res.getString(1));
   }
   // describe table
   sql = "describe " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(res.getString(1) + "\\t" + res.getString(2));
   }
   sql = "insert into " + tableName + " values (42,\\"hello\\"),
(48, \\"world\\")";
   stmt.execute(sql);
   sql = "select * from " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(String.valueOf(res.getInt(1)) + "\\t"
             + res.getString(2));
   }
   sql = "select count(1) from " + tableName;
   System.out.println("Running: " + sql);
   res = stmt.executeQuery(sql);
   while (res.next()) {
     System.out.println(res.getString(1));
   }
 }
```

}



将代码中的 \$kyuubihost 替换为您的 kyuubiserver 地址。

该程序会在 test_db 中创建 test_jdbc 表,写入两条数据并查询输出该数据。执行下面的命令对整个工程打包:

mvn package

上传 jar 并运行

上传过程与 JDBC 提交 Hive Spark 中一致,通过以下命令运行 KyuubiJdbcTest:

```
java -classpath ${package}-jar-with-dependencies.jar
org.apache.hive.KyuubiJdbcTest
其中package为您自定义的artifactId-version。运行结果如下:
Create table success!
Running: show tables 'test_kyuubi'
test_db
Running: describe test_kyuubi
     int
key
value string
Running: select * from test_kyuubi
42
      hello
48
       world
Running: select count(1) from test_kyuubi
2
```

销毁集群

当创建的集群不再使用时,可以销毁集群,退还关联资源;销毁集群将强制终止集群所提供的服务,并释放关联资源。

在 EMR on TKE 页面,选择目标集群的更多中的销毁;在弹出的对话框中,单击立即销毁。