ALIBABA CLOUD

阿里云

实时计算（流计算）
Flink Datastream开发指南

文档版本：20211122

阿里云

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。

2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。

3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。

4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。

5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	危险 重置操作将丢失用户配置数据。
警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	警告 重启操作将导致业务中断，恢复业务时间约十分钟。
注意	用于警示信息、补充说明等，是用户必须了解的内容。	注意 权重设置为0，该服务器不会再接受新请求。
说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击**设置**> **网络**> **设置网络类型**。
粗体	表示按键、菜单、页面名称等UI元素。	在**结果确认**页面，单击**确定**。
Courier字体	命令或代码。	执行 `cd /d C:/window` 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	`bae log list --instanceid` *Instance_ID*
[] 或者 [a|b]	表示可选项，至多选择一个。	`ipconfig [-all|-t]`
{} 或者 {a|b}	表示必选项，至多选择一个。	`switch {active|stand}`

目录

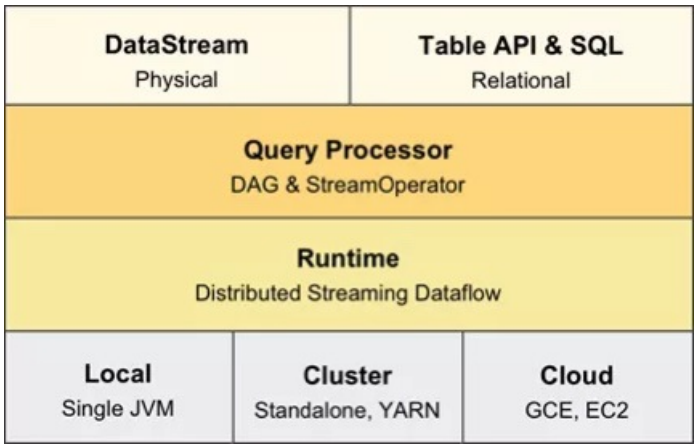
1.概述 05
2.数据存储白名单配置 06
3.自定义参数 08
4.监控报警 10
5.作业开发 11
6.作业提交 14
7.Datastream示例 17
7.1. 读取DataHub数据示例 17
7.2. 读取Kafka数据示例 20
7.3. 读取DataHub数据写入阿里云HBase示例 22
7.4. 读取日志服务SLS示例 25

1.概述

阿里云为实时计算产品Flink Datastream提供了作业提交、上线和启动功能，并提供了便利的运维管理和监控报警能力。

> 注意
>
> - 仅独享模式Blink 3.2.2及以上版本，支持Flink Datastream功能。
> - Flink Datastream暂不支持Flink SQL作业中的注册数据存储、作业调试和配置调优等功能。
> - 如果Datastream作业访问的上下游存储提供了白名单机制，您需要进行白名单配置，配置方法请参见数据存储白名单配置。
> - 目前实时计算产品支持的Datastream作业是基于开源的Flink版本的，详情请参见开源Flink版本。
> - Blink Datastream API完全兼容开源Flink 1.5版本，基于Flink 1.5版本开发的Datastream作业（包括Connetor）均可以在Blink上正常运行。Blink Datastream API与非1.5版本的开源Flink可能会存在不兼容的情况。

Flink Datastream提供了阿里云实时计算产品的底层API调用功能，方便您灵活地使用实时计算。



Flink Datastream开发指南主要包含如下内容：

- 作业开发

  介绍在阿里实时计算开发平台上提交、上线和启动Flink Datastream作业的流程。

- 监控报警

  介绍如何创建和启动报警规则，目前仅支持FailoverRate指标的监控报警。

2.数据存储白名单配置

新建的数据库通常默认拒绝外部设备的访问，只有配置在数据存储白名单中的IP地址才被允许访问。本文以RDS为例，为您介绍如何配置数据存储白名单。

实时计算IP地址

实时计算分为共享模式和独享模式，2种模式的IP地址有所不同。

- **共享模式IP地址**

  > **说明** 实时计算共享模式已于2019年12月24日正式下线，将不再支持共享模式新项目的购买，仅支持原有项目的扩缩容、续费操作。若有新购需求，推荐使用实时计算独享模式或Flink云原生模式。

  - 访问经典网络下的数据存储

    可根据项目所在的区域，配置对应区域的IP地址。

Region	白名单
华东2（上海）	11.53.0.0/16,11.50.0.0/16,10.152.0.0/16,10.154.0.0/16,11.132.0.0/16,11.178.0.0/16,11.200.210.74,11.200.215.195,11.217.0.0/16,11.219.0.0/16,11.222.0.0/16,11.223.116.79,11.223.69.0/24,11.223.70.0/24,11.223.70.173,11.223.70.48
华北2（北京）	11.223.0.0/16,11.220.0.0/16,11.204.0.0/16
华南1（深圳）	11.200.0.0/16

  - 访问VPC下的数据存储

    共享模式集群位于阿里云的经典网络，若需要访问阿里云VPC网络下的存储资源，需要通过VPC访问授权。VPC授权白名单网段查询步骤如下：

    a. 登录实时计算控制台。
    b. 将鼠标悬停至页面右上角账号名称。
    c. 在下拉菜单中，单击**项目管理**。
    d. 在左侧导航栏中，单击**VPC访问授权**。
    e. 在**VPC访问授权**页面中，单击对应VPC访问授权的**Region ID**字段下的链接，进入**白名单网段**页面。
    f. 在**白名单网段**窗口查询VPC授权白名单网段。

- **独享模式IP地址**

  独享模式中仅需要配置独享集群对应的弹性网卡（ENI）地址。ENI地址的查看步骤如下：

  i. 登录实时计算控制台。
  ii. 将鼠标悬停至页面右上角账号名称。
  iii. 在下拉菜单中，单击**项目管理**。
  iv. 单击左侧导航栏中的**集群列表**。

v. 在**集群列表**页面，单击**名称**字段下目标集群名称。

vi. 在集群信息窗口，查看集群的**ENI**信息。

配置RDS白名单

实时计算将RDS作为数据存储使用时，需要多次读写RDS数据库，必须将实时计算的IP地址配置进入RDS白名单。RDS白名单配置方法，请参见通过客户端、命令行连接RDS MySQL实例。

3.自定义参数

在DataStream作业中，您可以根据实际需求在作业开发页面配置自定义参数后，再从Main函数中获取该自定义参数。

配置方法

在DataStream作业中配置自定义参数，自定义参数格式为paramName=paramValue，其中paramName为参数名，paramValue为参数值。如下为DataStream作业开发页面默认的注释信息，您可以按照注释信息提示配置自定义参数。

```
--完整主类名，必填，例如com.alibaba.realtimecompute.DatastreamExample。
blink.main.class=${完整主类名}
--包含完整主类名的JAR包资源名称，多个JAR包时必填，例如blink_datastream.jar。
--blink.main.jar=${完整主类名jar包的资源名称}
--默认state backend配置，当作业代码没有显式声明时生效。
state.backend.type=niagara
state.backend.niagara.ttl.ms=129600000
--默认Checkpoint配置，当作业代码没有显式声明时生效。
blink.checkpoint.interval.ms=180000
--默认启用项目参数。
--enable.project.config=true
--设置自定义参数，代码中获取自定义参数的方法请参考如下链接。
--https://help.aliyun.com/document_detail/127758.html?spm=a2c4g.11174283.6.677.61fb1e49NJoWTR
```

> **说明** 一个DataStream作业中可定义多个自定义参数。

获取方法

在Datastream作业的Main函数中获取自定义参数。如果您已在作业开发页面配置了自定义参数，例如blink.job.name=jobnametest，则可以通过如下代码将字符串jobnametest赋值于blink.job.name变量。

```
import org.apache.flink.api.java.utils.ParameterTool;
import java.io.StringReader;
import java.nio.charset.StandardCharsets;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.Properties;
public class getParameterExample {
  public static void main(String[] args)throws Exception {
    final String jobName;
    final ParameterTool params = ParameterTool.fromArgs(args);
    /*此处必须写configFile，读取configFile中的参数。*/
    String configFilePath = params.get("configFile");
    /*创建一个Properties对象，用于保存在平台中设置的相关参数值。*/
    Properties properties = new Properties();
    /*将平台页面中设置的参数值加载到Properties对象中。*/
    properties.load(new StringReader(new String(Files.readAllBytes(Paths.get(configFilePath)), StandardC
harsets.UTF_8)));
    /*获取参数。*/
    jobName = (String) properties.get("blink.job.name");
  }
}
```

说明 该示例仅适用于在Main函数中获取并使用自定义参数。如果您需要在Flink算子中使用自定义参数，则需要按照以下步骤进行：

1. 在Main代码的基础上增加如下代码，将自定义参数转化为全局作业参数（GlobalJobParameters）。

   ```
   env.getConfig().setGlobalJobParameters(ParameterTool.fromPropertiesFile(configFilePath));
   ```

2. 在Flink算子获取自定义参数，代码如下。

   ```
   getRuntimeContext().getExecutionConfig().getGlobalJobParameters().toMap()
   ```

4.监控报警

本文为您介绍实时计算监控报警的操作流程以及如何创建报警规则。

什么是云监控报警服务

云监控服务能够收集阿里云资源或您自定义的监控指标、探测服务可用性以及针对指标设置警报，让您全面了解阿里云上的资源使用情况、业务的运行状况和健康度。您可以通过使用云监控服务及时接收异常报警，保证应用程序顺畅运行。

查看监控报警信息

1. 登录实时计算控制台。
2. 单击页面顶部的**运维**。
3. 在实时计算**运维**界面，单击目标作业名称。
4. 在目标作业运维信息页面的右上角，单击**更多 > 监控**。
5. 在监控页面，查看作业的监控指标。

创建报警规则

创建报警规则详情，参见设置报警规则。

> **说明**
>
> - Failover Rate表示最近1分钟平均每秒Failover的次数。例如，最近1分钟Failover了1次，则Failover Rate=1/60=0.01667。
> - DataStream作业开发过程中，若引用了开源Flink提供的Connector，则在云监控中不显示**业务延迟**、**读入RPS**和**写入RPS**这3项监控指标。

5.作业开发

本文为您介绍Datastream作业开发POM依赖包、Datastream作业开发示例和Datastream Connector。

> 注意
>
> - 仅独享模式Blink3.2.2及以上版本支持Datastream功能。
> - 建议使用Intellij IDEA中的Maven工程开发Datastream作业。
> - 实时计算Flink版独享模式不支持归档保存已停止（含暂停）的作业的运行日志。如果您需要查询已停止（含暂停）的作业运行日志，请将日志输出至您自定义的日志服务SLS或对象存储OSS中。Datastream作业的日志输出和级别修改步骤，请参见自定义日志级别和下载路径。
> - 为了避免JAR依赖冲突，您需要注意以下几点：
>   - 开发页面选择的Blink版本，请和Pom依赖Blink版本保持一致。
>   - Blink相关依赖，scope请使用provided，即 `<scope>provided</scope>`
>   - 其他第三方依赖请采用Shade方式打包，Shade打包详情参见Apache Maven Shade Plugin。

POM依赖包

请根据实际运行作业的Blink版本，自行添加开源版本所支持的POM依赖包。Blink3.4.0版本POM文件示例如下。

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.alibaba.blink</groupId>
  <artifactId>blink-datastreaming</artifactId>
  <version>1.0-SNAPSHOT</version>
  <properties>
    <scala.version>2.11.12</scala.version>
    <scala.binary.version>2.11</scala.binary.version>
    <blink.version>blink-3.4.0</blink.version>
    <maven.compiler.source>1.8</maven.compiler.source>
    <maven.compiler.target>1.8</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>com.alibaba.blink</groupId>
      <artifactId>flink-streaming-java_${scala.binary.version}</artifactId>
      <version>${blink.version}</version>
      <scope>provided</scope>
    </dependency>
<!--    Add test framework-->
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>4.8.1</version>
      <scope>test</scope>
```

```
        <scope>test</scope>
    </dependency>
    <dependency>
      <groupId>org.scala-lang</groupId>
      <artifactId>scala-library</artifactId>
      <version>2.11.12</version>
    </dependency>
<!--    Add logging framework-->
    <dependency>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-log4j12</artifactId>
      <version>1.7.7</version>
      <scope>runtime</scope>
    </dependency>
    <dependency>
      <groupId>log4j</groupId>
      <artifactId>log4j</artifactId>
      <version>1.2.17</version>
      <scope>runtime</scope>
    </dependency>
  </dependencies>
  <build>
    <plugins>
      <plugin>
        <groupId>org.apache.maven.plugins</groupId>
        <artifactId>maven-shade-plugin</artifactId>
        <version>3.2.0</version>
        <executions>
          <execution>
            <goals>
              <goal>shade</goal>
            </goals>
            <configuration>
              <transformers>
                <transformer implementation="org.apache.maven.plugins.shade.resource.ManifestResource
Transformer">
                  <manifestEntries>
                    <Main-Class>your main class</Main-Class>
                    <X-Compile-Source-JDK>${maven.compiler.source}</X-Compile-Source-JDK>
                    <X-Compile-Target-JDK>${maven.compiler.target}</X-Compile-Target-JDK>
                  </manifestEntries>
                </transformer>
              </transformers>
              <relocations combine.self="override">
                <relocation>
                  <pattern>XXX</pattern>
                  <shadedPattern>shaded.XXX</shadedPattern>
                </relocation>
              </relocations>
            </configuration>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
```

```
</project>
```

> **说明** 如果您需要依赖Snapshot版本，可以自行添加Snapshot版本所支持的POM依赖包。

作业示例

完整示例请参见读取DataHub数据示例。

Connector列表

Blink 3.2版本新增如下Datastream Connector:

- Kafka
- Kafka（开源版本）
- Hbase（开源版本）
- JDBC
- RDS SINK
- Elasticsearch
- MongoDB
- Redis

> **说明** Datastream支持的部分Connector已完成开源，开源信息请参见alibaba-flink-connectors。

6.作业提交

本文为您介绍如何提交Datastream作业。

前提条件

已创建实时计算项目。

> 注意
> - 仅独享模式Blink 3.2.2及以上版本支持Flink Datastream功能，推荐使用Blink 3.4.0及以上版本。
> - Datastream API作业不支持资源配置调优和启动位点设置，Blink 3.4.0以下版本，上线和启动作业过程中，使用默认配置即可。

操作步骤

1. 在实时计算控制台，单击顶部菜单栏中的**开发**。
2. 在**开发**页面的顶部菜单栏中，单击**新建作业**。
3. 在**新建作业**页面，配置作业参数。

作业参数	说明
文件名称	自定义作业的名称。作业名称需在当前项目中保持唯一。
作业类型	FLINK_STREAM/DATASTREAM。 **说明** Datastream API作业和Table API作业均选择 FLINK_STREAM/DATASTREAM 作业类型。如果无此类型，请您提交工单进行咨询。
存储位置	作业存储的位置。

4. 单击左侧导航栏中的**资源引用**，进入资源引用窗口。
5. 单击**新建资源**上传已经完成开发的Datastream作业JAR包。

> **说明** 在上传JAR包时，JAR包大小上限为300 MB。如果JAR包超过300 MB，请在集群绑定的OSS上传，或通过OpenAPI的方式上传JAR。

6. 单击**引用**。
7. 在作业**开发**界面配置参数。

```
blink.main.class=<完整主类名>
--函数完整类名，例如com.alibaba.realtimecompute.DemoTableAPI。
blink.job.name=<作业名>
--例如datastream_test。
blink.main.jar=<完整主类名JAR包的资源名称>
--完整主类名JAR包的资源名称，例如blink_datastream.jar。
```

- blink.main.class和blink.job.name为必须参数。请务必保证blink.job.name的值与步骤3中的文件名称一致。如果不一致，实际作业名称将以步骤3中的文件名称为准。
- 上传多个JAR包时需要配置blink.main.jar参数。
- 您可以先自行配置其它参数，然后在程序中引用。自定义参数配置及在代码中获取参数值的方法，请参见自定义参数。
- 请不要在参数配置中使用空格。
- Blink3.2.0及以上版本无需设置Checkpoint路径，系统会自动生成Checkpoint路径。
- Blink自3.4.0开始，JAR包代码中的所有参数配置优先级会高于实时计算平台上的参数配置。例如：
  - JAR包代码和自定义参数中都设置了statebackend，则优先使用JAR包中代码的配置。
  - JAR包代码和自定义参数中没有设置statebackend，则优先使用实时计算平台作业模板中的默认参数niagara statebackend。

> **说明** 请您谨慎删除模板中的默认参数，否则可能会导致作业无法Checkpoint和容错。作业名称blink.job.name是特例，代码中env.execute("jobname")设置的作业名称将会被创建作业时设置的作业名称替换，从而保持一致。此外，Metric（包括自定义Metric）名称也需要和创建作业时设置的作业名称保持一致。

8. 上线作业。
   - Blink 3.4.0以下版本
     a. 资源配置

        选择对应的资源配置方式。第1次**启动**作业时，建议使用系统默认配置。

        > **说明** 实时计算支持手动资源配置，手动资源配置的方法请参见手动配置调优。

     b. 数据检查

        通过数据检查后，单击**下一步**。

     c. 上线作业

        单击**上线**。
   - Blink 3.4.0及以上版本
     a. 单击作业上方**上线**。
     b. 选择**资源配置方式**。
        - **代码配置**：使用代码内的资源配置，与开源Flink形式一致。
        - **手动配置**：使用**资源配置**界面中手动调整的资源配置。
          a. 在**开发**页面右侧**资源配置**栏，单击**配置信息操作 > 重新获取配置信息**。
          b. 根据需要手动修改配置信息。
          c. 单击**配置信息操作 > 应用当前配置**，保存配置。

          > **说明** 手动配置时，代码显式配置的资源优先级高于平台界面上的资源配置。例如，代码中显式设置了某些算子的资源，则平台界面中对应算子的资源配置失效。实际运行时，算子的资源以您代码中显式的配置为准。代码中未显示的资源配置，以平台界面上的配置信息为准。

c. 单击**下一步**进行数据检查或单击**跳过数据检查**。

d. 单击**上线**。

9. 在**运维**页面，单击目标作业**操作**列下的**启动**。

7.Datastream示例

7.1. 读取DataHub数据示例

本文为您介绍如何使用Datastream作业读取阿里云DataHub数据。

前提条件

- 本地安装了Java JDK 8。
- 本地安装了Maven 3.x。
- 本地安装了用于Java或Scala开发的IDE，推荐IntelliJ IDEA，且已配置完成JDK和Maven环境。
- 在DataHub上创建了Topic，并且Topic中存在测试数据。

  > **说明** 测试数据需要有4个字段，数据类型依次为STRING、STRING、DOUBLE和BIGINT。

- 已下载datahub-demo-master示例。

背景信息

本文以Windows和Mac操作系统为例进行演示。

> **注意** 仅Blink3.x版本支持本示例。

开发

1. 实时计算Datastream完全兼容开源Flink 1.5.2版本。下载并解压flink-1.5.2-compatible分支到本地。

   > **说明** 下载文件中的datahub-connector中同样实现了DataHub Sink功能，具体实现请参见下载文件中的DatahubSinkFunction.java和DatahubSinkFunctionExample.java。

2. 在CMD命令窗口，进入alibaba-flink-connectors-flink-1.5.2-compatible目录后，执行如下命令。

   ```
   mvn clean install
   ```

   可以看到如下结果。

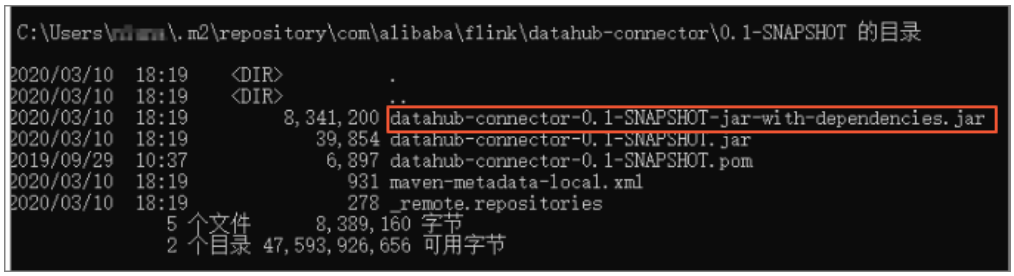
   ```
   [INFO] ------------------------------------------------------------------------
   [INFO] Reactor Summary for aliyun-flink-connectors-parent 0.1-SNAPSHOT:
   [INFO]
   [INFO] aliyun-flink-connectors-parent ..................... SUCCESS [04:54 min]
   [INFO] aliyun-connectors-common ........................... SUCCESS [01:55 min]
   [INFO] cloudhbase-connector ............................... SUCCESS [04:47 min]
   [INFO] datahub-connector .................................. SUCCESS [ 24.694 s]
   [INFO] sls-shaded-sdk ..................................... SUCCESS [ 23.142 s]
   [INFO] sls-connector ...................................... SUCCESS [ 10.359 s]
   [INFO] ------------------------------------------------------------------------
   [INFO] BUILD SUCCESS
   [INFO] ------------------------------------------------------------------------
   [INFO] Total time:  12:39 min
   [INFO] Finished at: 2020-03-10T18:20:03+08:00
   [INFO] ------------------------------------------------------------------------
   ```

   命令执行成功后，datahub-connector对应的JAR包安装到本地的Maven仓库，通常默认安装在当前登录的用户文件夹下的.m2文件夹下。

3. 执行如下命令确认是否存在datahub-connector-0.1-SNAPSHOT-jar-with-dependencies.jar文件（将一个JAR及其依赖的三方JAR全部打到一个包中），后续会使用该JAR。
   - Windows操作系统

     ```
     dir C:\Users\用户名\.m2\repository\com\alibaba\flink\datahub-connector\0.1-SNAPSHOT
     ```

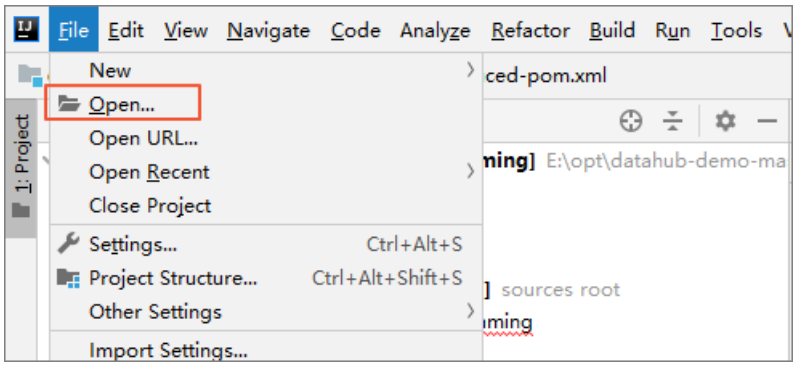
     Windows操作系统执行结果

   - Mac操作系统

     ```
     ls /Users/用户名/.m2/repository/com/alibaba/flink/datahub-connector/0.1-SNAPSHOT
     ```

4. 在IntelliJ IDEA中，单击**File > Open**，打开刚才解压缩完成的datahub-demo-master包后，双击pom.xml查看代码。

   > **注意**
   >
   > - IDE本地调试时需要将<scope>provided</scope>注释掉。
   > - 在本示例中已默认使用<classifier>jar-with-dependencies</classifier>依赖步骤3中的datahub-connector-0.1-SNAPSHOT-jar-with-dependencies.jar。

5. 修改DatahubDemo.java文件中的DataHub相关参数。

   ```
   private static String endPoint = "inner endpoint";//内网访问。
   //private static String endPoint ="public endpoint";//公网访问（填写内网Endpoint，就不用填写公网Endpoint）。
   private static String projectName = "yourProject";
   private static String topicSourceName = "yourTopic";
   private static String accessId = "yourAK";
   private static String accessKey = "yourAS";
   private static Long datahubStartInMs = 0L;//设置消费的启动位点对应的时间。
   ```

> **说明** 在实时计算产品上运行时请使用内网Endpoint，VPC ECS Endpoint和经典网络ECS Endpoint都属于内网Endpoint，在使用时请注意：
>
> - 如果您的环境是VPC，请使用VPC ECS Endpoint。
> - 如果您的环境是OXS，请使用经典网络ECS Endpoint。

6. 在下载文件pom.xml所在的目录执行如下命令打包文件。

```
mvn clean package
```

根据您的项目设置的artifactId，target目录下会出现blink-datastreaming-1.0-SNAPSHOT.jar，即代表完成了开发工作。

上线

请参见上线完成作业上线。

> **注意** 作业上线前，请在**开发**页面右侧的**资源配置**页签，配置源表的并发数，源表并发数不能大于源表的Shard数，否则作业启动后JM（Job Manager）报错。

本示例对应的作业内容如下。

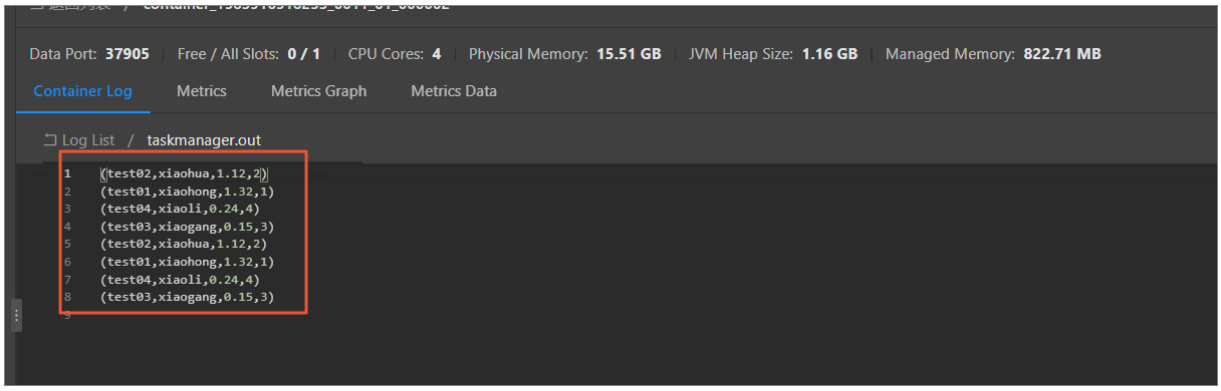
```
--完整主类名，必填，例如com.alibaba.realtimecompute.DatastreamExample。
blink.main.class=com.alibaba.blink.datastreaming.DatahubDemo
--作业名称。
blink.job.name=datahub_demo
--包含完整主类名的JAR包资源名称，多个JAR包时必填，例如blink_datastream.jar。
blink.main.jar=${完整主类名jar包的资源名称}
--默认statebackend配置，当作业代码没有显式配置时生效。
state.backend.type=niagara
state.backend.niagara.ttl.ms=129600000
--默认checkpoint配置，当作业代码没有显式配置时生效。
blink.checkpoint.interval.ms=180000
```

> **说明**
>
> - 注意修改blink.main.class和blink.job.name。
> - 您可以设置自定义参数，详情请参见自定义参数。

验证

在**运维**页面，查看Sink节点的**taskmanager.out**信息，本示例中使用Print作为Sink。

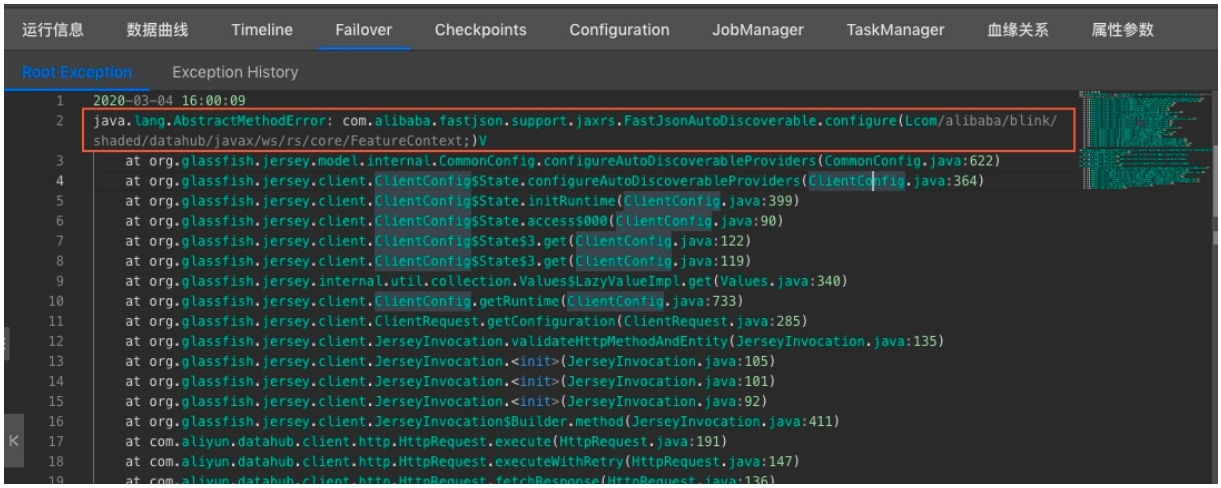
如果出现如下结果，则表示已经成功读取了阿里云DataHub中的数据。



常见问题

在作业运行时，如果界面上出现如下类似的错误，表示存在JAR包冲突。

```
java.lang.AbstractMethodError：com.alibaba.fastjson.support.jaxrs.FastJsonAutoDiscoverable.configure(Lcom/alibaba/blink/shaded/datahub/javax/ws/rs/core/FeatureContext;)
```



建议您使用maven-shade-plugin插件的Relocation功能，解决JAR包冲突的问题。

```
<relocations combine.self="override">
  <relocation>
    <pattern>org.glassfish.jersey</pattern>
    <shadedPattern>com.alibaba.blink.shaded.datahub.org.glassfish.jersey</shadedPattern>
  </relocation>
</relocations>
```

7.2. 读取Kafka数据示例

本文为您介绍如何使用Datastream作业读取阿里云Kafka数据。

前提条件

- 本地安装了Java JDK 8。
- 本地安装了Maven 3.x。

- 本地安装了用于Java或Scala开发的IDE，推荐IntelliJ IDEA，且已配置完成JDK和Maven环境。
- 创建与实时计算独享模式相同VPC的Kafka实例，并创建了Topic和Consumer Group。

背景信息

- 实时计算Datastream完全兼容开源Flink 1.5.2版本，阿里云Kafka兼容开源Kafka，因此可以直接使用Maven仓库里的Kafka Connetor来连接阿里云Kafka。
- 实时计算独享模式通过内网接入阿里云Kafka，无需进行SASL认证鉴权。如果您在本地IDE上通过公网方式接入阿里云Kafka，则需要进行SASL认证鉴权，具体配置请参见kafka-java-demo。

> **注意** 仅Blink3.x版本支持本示例。

开发

1. 下载并解压alikafka-demo-master示例到本地。
2. 在IntelliJ IDEA中，单击**File > Open**，打开刚才解压缩完成的alikafka-demo-master。
3. 双击打开*\alikafka-demo-master\src\main\resources*目录下的kafka.properties后，修改bootstrap.servers、topic和group.id为您创建的Kafka实例对应值。

   ```
   ## 接入点，通过控制台获取。
   ## 注意公网和VPC接入点的区别，在Blink独享模式下需要使用默认接入点，而非SSL接入点。
   bootstrap.servers=ip1:port,ip2:port,ip3:port
   ## Topic，通过控制台创建。
   topic=your_topic
   ## ConsumerGroup，通过控制台创建。
   group.id=your_groupid
   ```

4. 在下载文件中pom.xml所在的目录执行如下命令打包文件。

   ```
   mvn clean package
   ```

   根据您的项目设置的artifactId，target目录下会出现blink-datastreaming-1.0-SNAPSHOT.jar的JAR包，即代表完成了开发工作。

上线

请参见上线完成作业上线。

> **说明** 注意修改blink.main.class、blink.job.name和blink.main.jar。

本示例对应的作业内容如下。

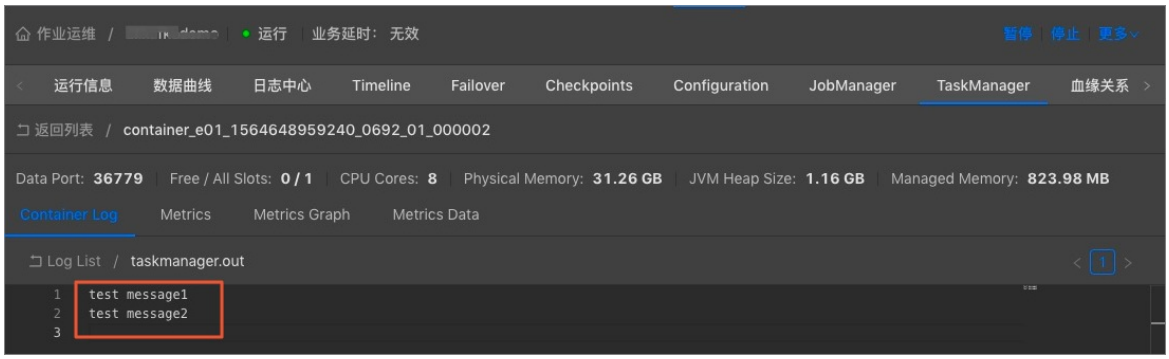
```
--完整主类名，必填，例如com.alibaba.realtimecompute.DatastreamExample。
blink.main.class=com.alibaba.blink.datastreaming.AliKafkaConsumerDemo
--作业名称。
blink.job.name=alikafkaconsumerdemo
--包含完整主类名的JAR包资源名称，多个JAR包时必填，例如blink_datastream.jar。
blink.main.jar=blink-datastreaming-1.0-snapshot.jar
--默认statebackend配置，当作业代码没有显式配置时生效。
state.backend.type=niagara
state.backend.niagara.ttl.ms=129600000
--默认checkpoint配置，当作业代码没有显式配置时生效。
blink.checkpoint.interval.ms=180000
```

> **说明** 您可以设置自定义参数，详情请参见自定义参数。

验证

1. 在Kafka控制台发送消息。
2. 在实时计算的**运维**界面，查看Sink节点的taskmanager.out输出结果，本示例中使用Print作为Sink。

   出现类似如下输出（具体内容以实际发送的消息内容为准），则表示已经成功读取了阿里云Kafka中的数据。



7.3. 读取DataHub数据写入阿里云HBase示例

本文为您介绍如何使用Datastream作业读取DataHub数据写入HBase。

前提条件

- 本地安装了Java JDK 8。
- 本地安装了Maven 3.x。
- 本地安装了用于Java或Scala开发的IDE，推荐IntelliJ IDEA，且已配置完成JDK和Maven环境。
- 在DataHub上创建了Topic，并且Topic中存在测试数据。

  > **说明** 测试数据需要有3个字段，数据类型依次为BOOLEAN、STRING和STRING。

- 创建与实时计算独享模式同一地域下相同VPC的HBase示例，并创建表和列簇。通过Shell访问HBase集群步骤请参见使用Shell访问。

  > **说明**
  > - 本示例为标准版HBase。
  > - 实时计算集群IP需要添加至HBase白名单。

背景信息

本文以Windows系统为例进行演示。

> **注意** 仅Blink 3.x版本支持本示例。

开发

1. 下载并解压Hbase_Demo-master示例到本地。
2. 在IntelliJ IDEA中，单击**File > Open**，打开刚才解压缩完成的Hbase_Demo-master。
3. 双击打开*\Hbase_Demo-master\src\main\java\Hbase_Demo*后，修改HbaseDemo.java文件中的DataHub与HBase相关参数。

```
//DataHub相关参数
//private static String endPoint ="public endpoint";//公网访问（填写内网Endpoint，就不用填写公网Endpoint）。
private static String endPoint = "inner endpoint";//内网访问。
private static String projectName = "yourProject";
private static String topicSourceName = "yourTopic";
private static String accessId = "yourAK";
private static String accessKey = "yourAS";
private static Long datahubStartInMs = 0L;//设置消费的启动位点对应的时间。
//Hbase相关参数
private static String zkQuorum = "yourZK";
private static String tableName = "yourTable";
private static String columnFamily = "yourcolumnFamily";
```

4. 在下载文件中pom.xml所在的目录执行如下命令打包文件。

```
mvn package -Dcheckstyle.skip
```

根据您的项目设置的artifactId，target目录下会出现Hbase_Demo-1.0-SNAPSHOT-shaded.jar的JAR包，即代表完成了开发工作。

上线

请参见上线完成作业上线。

> **说明** 作业上线前，请在**开发**页面右侧的**资源配置**页签，配置源表的并发数，源表并发数不能大于源表的Shard数，否则作业启动后JM（Job Manager）报错。

本示例对应的作业内容如下。

```
--完整主类名，必填。
blink.main.class=Hbase_Demo.HbaseDemo
--作业名称。
blink.job.name=datahub_demo
--包含完整主类名的JAR包资源名称，多个JAR包时必填。
--blink.main.jar=Hbase_Demo-1.0-snapshot.jar
--默认statebackend配置，当作业代码没有显式配置时生效。
state.backend.type=niagara
state.backend.niagara.ttl.ms=129600000
--默认checkpoint配置，当作业代码没有显式配置时生效。
blink.checkpoint.interval.ms=180000
```

> **说明** 您可以设置自定义参数，详情请参见自定义参数。

验证

1. 在实时计算控制台发送测试数据至DataHub。

```
CREATE TABLE kafka_src (
  a BOOLEAN
) WITH (
  type = 'random'
);
CREATE TABLE event_logs (
  `a` BOOLEAN,
  b VARCHAR,
  `c` VARCHAR
) WITH (
  type = 'datahub',
  endPoint = '<yourEndpoint>',
  project = '<yourProject>',
  topic = '<yourTopic>',
  accessId='<yourAccessId>',
  accessKey='<yourAccessKey>'
);
INSERT INTO event_logs
SELECT
  a,'rowkey3' as b,'123' as c
FROM kafka_src;
```

2. 连接HBase集群，详情请参见使用Shell访问。
3. 执行 `scan 'hbase_sink'` 查看写入数据。

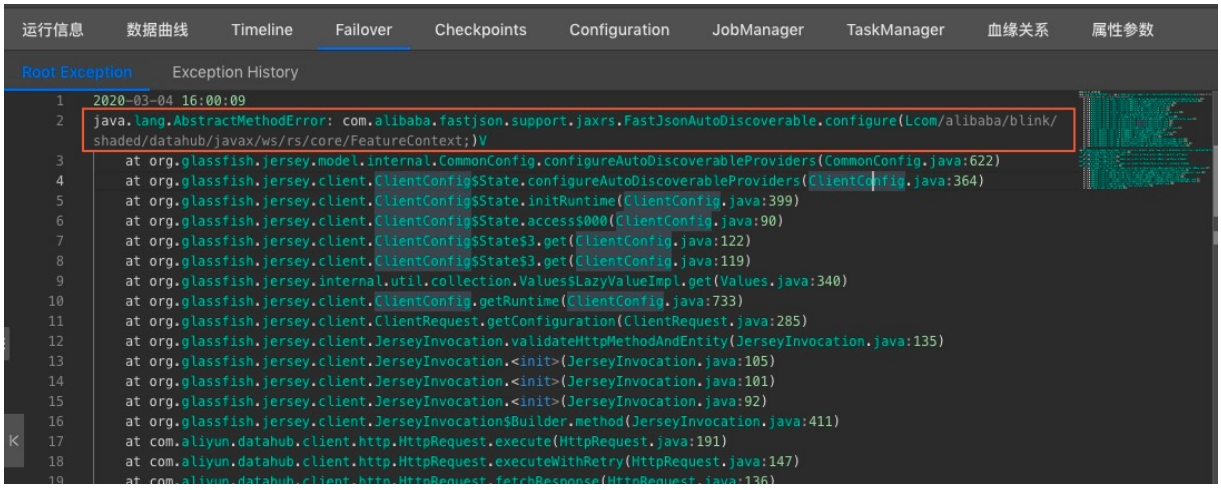
   出现类似如下输出，则表示已经成功将DataHub数据写入阿里云HBase。

```
hbase(main):128:0> scan 'hbase_sink'
ROW                  COLUMN+CELL
 rowkey3             column=f1:a, timestamp=1597741134871, value=[B@56bc3604
 rowkey3             column=f1:b, timestamp=1597741134871, value=[B@f5c05
 rowkey3             column=f1:c, timestamp=1597741134871, value=[B@25c03326
1 row(s)
Took 0.0590 seconds
```

常见问题

在作业运行时，如果界面出现如下类似的错误，表示存在JAR包冲突。

```
java.lang.AbstractMethodError: com.alibaba.fastjson.support.jaxrs.FastJsonAutoDiscoverable.configure(Lcom/alibaba/blink/shaded/datahub/javax/ws/rs/core/FeatureContext;)V
```



建议您使用maven-shade-plugin插件的Relocation功能，解决JAR包冲突的问题。

```
<relocations combine.self="override">
  <relocation>
    <pattern>org.glassfish.jersey</pattern>
    <shadedPattern>com.alibaba.blink.shaded.datahub.org.glassfish.jersey</shadedPattern>
  </relocation>
</relocations>
```

7.4. 读取日志服务SLS示例

本文为您介绍如何使用Datastream作业读取阿里云日志服务SLS数据示例。

前提条件

- 本地安装了Java JDK 8。
- 本地安装了Maven 3.x。
- 本地安装了用于Java或Scala开发的IDE，推荐IntelliJ IDEA，且已配置完成JDK和Maven环境。
- SLS上已创建了logstore ，并且logstore中存在测试数据。

背景信息

本文以Windows操作系统为例进行演示。

> **注意** 仅Blink 3.x版本支持本示例。

开发

1. 下载并解压SLS_Demo示例到本地。
2. 在IntelliJ IDEA中，单击**File > Open**，打开刚才解压缩完成的SLS_Demo-master。
3. 双击打开*SLS_Demo-master\src\main\java\com\aliyun\openservices\log\flink\ConsumerSample*后，修改ConsumerSample.java文件中的SLS的相关参数。

   ```
   private static final String SLS_ENDPOINT = "VPC endpoint";//线上使用经典网络及VPC Endpoint
  // private static final String SLS_ENDPOINT = "public endpoint";//本地测试使用 公网Endpoint
   private static final String ACCESS_KEY_ID = "yourAK";
   private static final String ACCESS_KEY_SECRET = "yourAS";
   private static final String SLS_PROJECT = "yourProject";
   private static final String SLS_LOGSTORE = "yourlogstore";
   //1、启动位点秒级的时间戳读取数据;2、读取全量加增量数据Consts.LOG_BEGIN_CURSOR;
   //3、读取增量数据Consts.LOG_END_CURSOR
   private static final String StartInMs = Consts.LOG_END_CURSOR;
   ```

   > **说明** IDE本地调试注意将<scope>provided</scope>注释掉。

4. 在下载文件中pom.xml所在的目录执行如下命令打包文件。

   ```
   mvn clean package
   ```

根据您的项目设置的artifactId，target目录下会出现flink-log-connector-0.1.21-SNAPSHOT.jar的JAR包，即代表完成了开发工作。

上线

请参见上线完成作业上线。

> **说明** 作业上线前，请在**开发**页面右侧的**资源配置**页签，配置源表的并发数，源表并发数不能大于源表的Shard数，否则作业启动后JM（Job Manager）报错。

本示例对应的作业内容如下。

```
--完整主类名，必填。
--blink.main.class=com.aliyun.openservices.log.flink.ConsumerSample
--作业名称。
blink.job.name=sls
--包含完整主类名的JAR包资源名称，多个JAR包时必填。
--blink.main.jar=flink-log-connector-0.1.21-snapshot.jar
--默认statebackend配置，当作业代码没有显式配置时生效。
state.backend.type=niagara
state.backend.niagara.ttl.ms=129600000
--默认checkpoint配置，当作业代码没有显式配置时生效。
blink.checkpoint.interval.ms=180000
```

> **说明** 您可以设置自定义参数，详情请参见自定义参数。

验证

在实时计算Flink版**运维**界面，查看Sink节点的taskmanager.out输出结果，本示例中使用Print作为Sink。

如果出现如下结果，则表示已经成功读取了阿里云SLS中的数据。