流计算 Oceanus

Python 开发指南

腾讯云

Python 开发指南

最近更新时间：2023-06-21 15:21:59

概述

流计算 Oceanus 支持使用 Python 开发 Flink 作业，您可以在 Python 环境中使用 Flink 的所有功能，在降低开发门槛的同时，发挥 Python 在数据处理、机器学习领域的生态优势。
本章节主要提供在独享集群上开发 Python 作业的指南。通过阅读本章节，您将学习到以下内容：

- 环境信息
- 使用 Python 依赖

环境信息

软件版本

目前流计算 Oceanus 支持运行基于开源 Flink V1.13 开发的 Python 作业，且预装了 Python 3.7 版本的环境。

Python 软件列表

流计算 Oceanus 作业的 Python 环境中已安装下列软件包。

软件包	版本
apache-beam	2.27.0
apache-flink	1.13.2
apache-flink-libraries	1.13.2
avro-python3	1.9.1
beautifulsoup4	4.10.0
certifi	2020.12.5
chardet	4.0.0
click	8.0.3
cloudpickle	1.2.2
crcmod	1.7
Cython	0.29.16
dill	0.3.1.1
docopt	0.6.2
fastavro	0.23.6
future	0.18.2
grpcio	1.29.0

hdfs	2.6.0
httplib2	0.17.4
idna	2.10
importlib-metadata	4.10.0
joblib	1.1.0
jsonpickle	1.2
mock	2.0.0
nltk	3.6.7
numpy	1.19.5
oauth2client	3.0.0
pandas	1.0.0
pbr	5.5.1
protobuf	3.15.3
py4j	0.10.8.1
pyarrow	0.17.1
pyasn1	0.4.8
pyasn1-modules	0.2.8
pydot	1.4.2
pymongo	3.11.3
pyparsing	2.4.7
python-dateutil	2.8.0
pytz	2021.1
regex	2021.11.10
requests	2.25.1
rsa	4.7.2
scikit-learn	1.0.2
scipy	1.7.3
six	1.15.0
soupsieve	2.3.1

threadpoolctl	3.0.0
tqdm	4.62.3
typing-extensions	3.7.4.3
urllib3	1.26.3
zipp	3.7.0

使用 Python 依赖

您可以在 Python 作业中使用第三方 Python 包、JAR 包和数据文件等依赖。

使用第三方 Python 包

如果您的 Python 作业中使用了第三方 Python 包，您可以采用如下方式来指定。

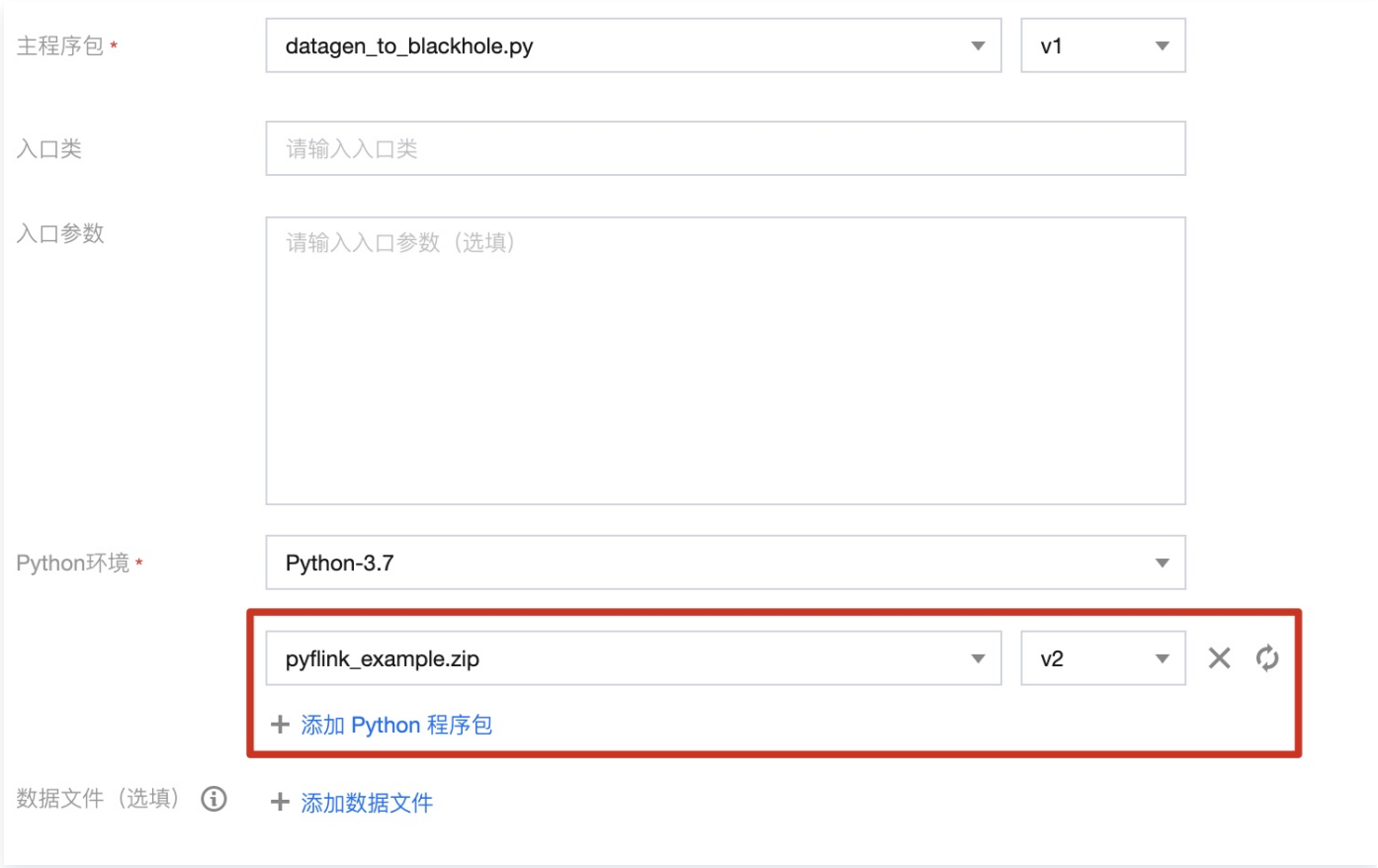
1. 将第三方 Python 包打包成 Zip 文件，在依赖管理中上传 Python 程序包。
   - 关于 zip 包制作：将 `pip install xxx -t .` 这个包安装到当前目录。然后使用命令 `zip -r xxx.zip xxx/*` 制作 zip 包。

     > **注意**
     > 如果该包包含 so 文件. 请确保您的环境为 debian 11.1。

     例如，打包一个 requests 库的 zip 包：

     ```
     mkdir /tmp/example
     cd /tmp/example
     pip install requests -t .
     zip -r9 ../pyflink_lib_example.zip ./*
     ```

2. 开发调试界面，点击添加 Python 程序包，引用上传的第三方 Python 包。



使用 JAR 包

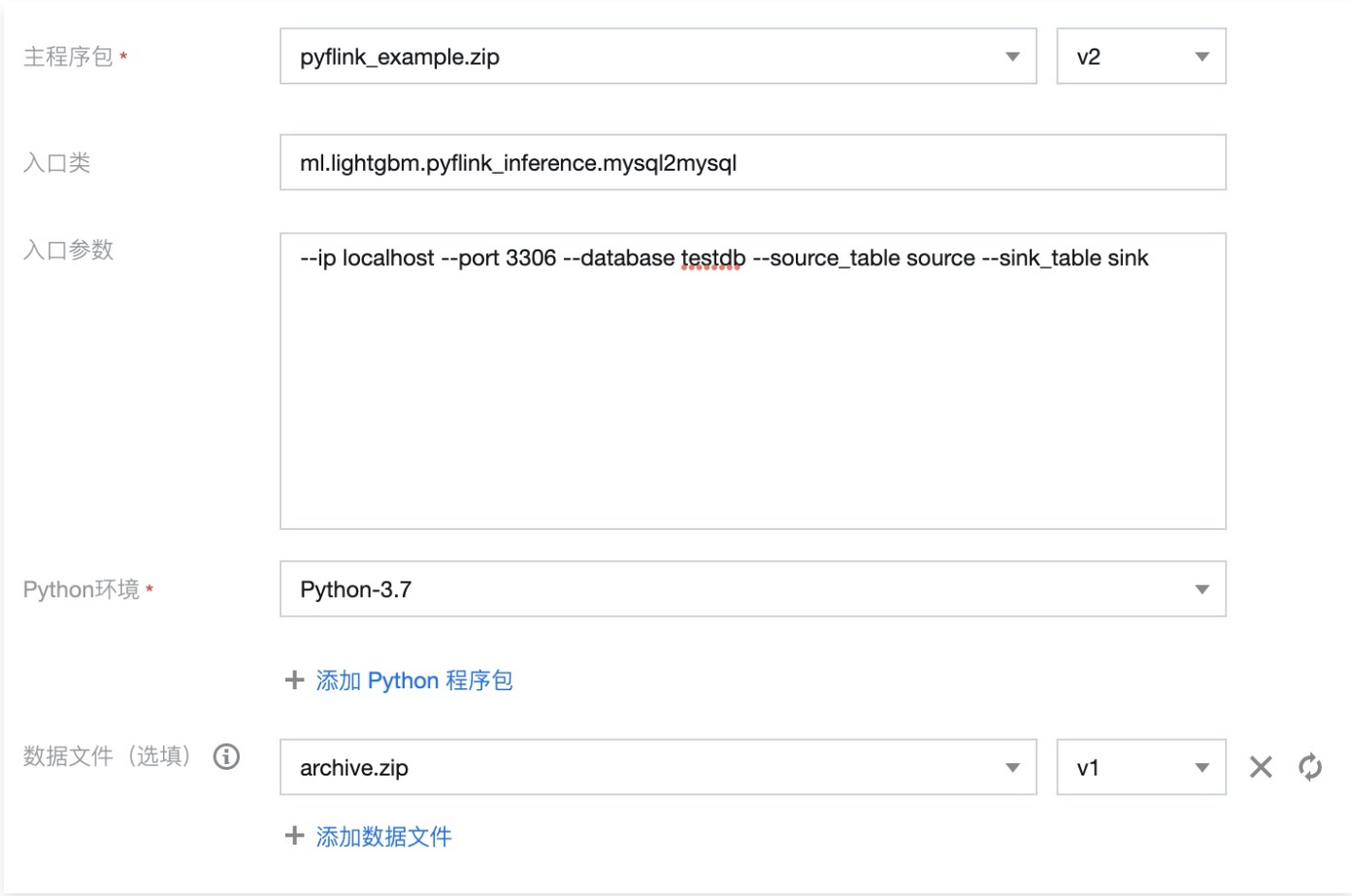
如果您的 Python 作业中使用了 Java 类，例如作业中使用了 Connector 或者 Java 自定义函数时，可以通过如下方式来指定 Connector 或者 Java 自定义函数的 JAR 包。

1. 在依赖管理中上传 JAR 包。
2. 开发调试界面，单击**作业参数**，单击**引用 JAR 程序包**，选择上传的 JAR 包。

使用数据文件

如果您的数据文件的数量比较多时，您可以将数据文件打包为 Zip 包，然后通过如下方式在 Python作业中使用。

1. 在依赖管理中上传数据文件。
2. 开发调试界面中选择上传的数据文件。



3. 在 Python 自定义函数中使用数据文件。如果数据文件所在的压缩包名称为 archive.zip，则在 Python 自定义函数中可以编写以下代码来访问 archive.zip 数据文件。

```
def my_udf():
    with open("archive.zip/mydata/data.txt") as f:
    ...
```