LakeFormation

最佳实践

文档版本 01

发布日期 2024-12-18

华为云计算技术有限公司

商标声明


和其他华为商标均为华为技术有限公司的商标。
本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目 录

1 配置开源 Spark 组件对接 LakeFormation

1.1 环境准备

在配置开源Spark组件对接LakeFormation前，需要提前准备以下信息：

步骤1 准备可用的开源Spark环境、开源Hive环境。并安装Git环境。

目前仅支持对接Spark 3.1.1以及Spark 3.3.1两个版本。对应使用Hive内核版本为2.3。

步骤2 准备LakeFormation实例，详细操作请参考**创建LakeFormation实例**。

步骤3 创建LakeFormation接入客户端，并与Spark在相同的虚拟私有云、子网下，详细操作请参考**管理接入客户端**。

步骤4 准备开发环境，详细操作请参考**准备开发程序环境**中“准备开发环境”部分，其中“安装和配置IntelliJ IDEA”为可选。

步骤5 准备LakeFormation客户端。

- **方式一：下载客户端发行版**

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lake-formation-lakecat-sdk-java/releases**

  根据Spark、Hive版本下载对应客户端（如Spark3.1.1，对应Hive版本为2.3.7，则下载lakeformation-lakecat-client-hive2.3-v1.0.0.rar）

  校验压缩包：下载后在Windows环境下执行**certutil -hashfile** *<压缩包>* **sha256**，检查回显信息与对应sha256文件内容是否一致。

- **方式二：本地编译客户端**

  a. 获取客户端代码。

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lake-formation-lakecat-sdk-java**。

  b. 在Git中执行以下命令将分支切换为“master_dev”：

  git checkout master_dev

  c. 配置maven源，详细操作请参考**获取SDK并配置maven**。

  d. 获取以下jar包及对应的pom文件，并将文件放入maven本地仓库。

例如本地仓库地址为“D:\maven\repository”，则放入“D:\maven\repository\com\huaweicloud\hadoop-huaweicloud\3.1.1-hw-53.8”目录下。

- jar包：**https://github.com/huaweicloud/obsa-hdfs/blob/master/release/hadoop-huaweicloud-3.1.1-hw-53.8.jar**
- pom文件：**https://github.com/huaweicloud/obsa-hdfs/blob/master/hadoop-huaweicloud/pom.xml**，并改名为hadoop-huaweicloud-3.1.1-hw-53.8.pom。

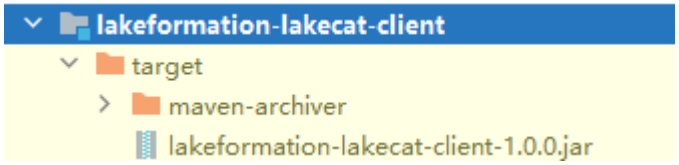
e. 编译打包客户端代码。

f. 根据Spark版本，进入客户端工程目录，执行以下打包命令：

mvn clean install -DskipTests=true -P"*${SPARK_PROFILE}*" -P"*${HIVE_PROFILE}*"

- SPARK_PROFILE：填写spark-3.1或者spark-3.3。
- HIVE_PROFILE：填写hive-2.3。

g. 打包完成后可以在“lakeformation-lakecat-client”的target目录下获取“lakeformation-lakecat-client-1.0.0.jar”。



步骤6 准备并补充替换Hive内核相关的jar包。

说明

如仅使用**SparkCatalogPlugin方式**对接，不使用**MetastoreClient方式**对接，本步骤可省略。

- **方式一：下载预构建Hive相关jar包**

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lakeformation-lakecat-sdk-java/releases**

  根据Spark、Hive版本下载对应客户端（如Spark3.1.1，对应Hive版本为2.3.7，则下载hive-exec-2.3.7-core.jar、hive-common-2.3.7.jar）

- **方式二：本地编译Hive相关jar包**

  如果对接的环境为Spark 3.1.1，则需使用Hive 2.3.7版本。对接的环境为Spark 3.3.1，则需使用Hive 2.3.9版本。

  Windows系统下需要在WSL开发环境下进行maven相关操作。

  a. 根据Hive版本下载Hive源码。

  例如Hive内核版本号为2.3.9，则下载链接为**https://github.com/apache/hive/tree/rel/release-2.3.9**。

  b. 将LakeFormation的客户端代码中的patch应用到Hive源码。

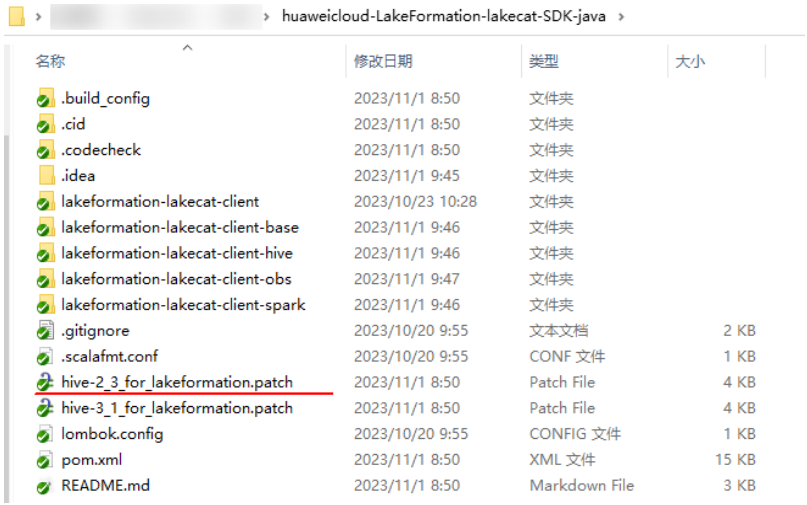
  i. 根据需要切换Hive源码分支，例如Hive内核版本号为2.3.9，则执行如下命令：

  git checkout rel/release-*2.3.9*

  ii. 执行以下命令，将patch应用到切换分支后的Hive源码工程。

  mvn patch:apply -DpatchFile=${*your patch file location*}

其中，“your patch file location”为hive-2_3_for_lakeformation.patch的存储路径。patch文件可在客户端工程中获取，如下图所示：



iii. 执行以下命令重新编译Hive内核源码。

mvn clean install -DskipTests=true

步骤7 补充Spark环境所需jar包。

获取以下jar包，并补充或替换到Spark的jars目录下。

表 1-1 获取 Spark 环境所需 jar 包

序号	jar包名称	获取途径
1	spring-web-5.3.24.jar	https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-web/5.3.24/spring-web-5.3.24.jar
2	spring-core-5.3.24.jar	https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-core/5.3.24/spring-core-5.3.24.jar
3	spring-context-5.3.24.jar	https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-context/5.3.24/spring-context-5.3.24.jar
4	spring-beans-5.3.24.jar	https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-beans/5.3.24/spring-beans-5.3.24.jar

序号	jar包名称	获取途径
5	caffeine-2.9.3.jar	https://mirrors.huaweicloud.com/repository/maven/com/github/ben-manes/caffeine/caffeine/2.9.3/caffeine-2.9.3.jar
6	mapstruct-1.5.3.Final.jar	https://mirrors.huaweicloud.com/repository/maven/org/mapstruct/mapstruct/1.5.3.Final/mapstruct-1.5.3.Final.jar
7	log4j-api-2.19.0.jar（spark3.3无需补充）	https://mirrors.huaweicloud.com/repository/maven/org/apache/logging/log4j/log4j-api/2.19.0/log4j-api-2.19.0.jar
8	java-sdk-core-3.2.4.jar （如果仅使用自定义认证信息获取类使用Token认证，则无需补充该jar包。）	https://mirrors.huaweicloud.com/repository/maven/huaweicloudsdk/com/huawei/apigateway/java-sdk-core/3.2.4/java-sdk-core-3.2.4.jar
9	bcprov-jdk15to18-1.70.jar	https://mirrors.huaweicloud.com/repository/maven/org/bouncycastle/bcprov-jdk15to18/1.70/bcprov-jdk15to18-1.70.jar
10	jca-1.0.4.jar	https://mirrors.huaweicloud.com/repository/maven/org/openeuler/jca/1.0.4/jca-1.0.4.jar
11	hadoop-huaweicloud-3.1.1-hw-53.8.jar	https://github.com/huaweicloud/obsa-hdfs/blob/master/release/hadoop-huaweicloud-3.1.1-hw-53.8.jar
12	lakeformation-lakecat-client-1.0.0.jar	步骤5中获取。
13	hive-exec-${version}-core.jar	步骤6中获取。
14	hive-common-${version}.jar	步骤6中获取。

----结束

1.2 配置 Spark 对接 LakeFormation

说明

使用pyspark时，需要将以下配置中“spark.hadoop”开头的参数去掉“spark.hadoop”后配置到hive-site.xml配置文件中。

对接通用配置

在“spark/conf/spark-defaults.conf”中添加以下配置：

```
# 项目ID，必选参数，此处配置值仅作为参考
spark.hadoop.lakeformation.project.id=项目ID
# LakeFormation实例ID，可选参数，通过LakeFormation实例界面获取，如不填写则连接到默认实例，此处配置值仅作为参考
spark.hadoop.lakeformation.instance.id=LakeFormation实例ID
#访问lakeformation IAM认证AK信息，可选参数，如果为自定义认证信息获取类可忽略
spark.hadoop.lakeformation.authentication.access.key=AK信息
#访问lakeformation IAM认证SK信息，可选参数，如果为自定义认证信息获取类可忽略
spark.hadoop.lakeformation.authentication.secret.key=SK信息
#访问lakeformation IAM认证信息securitytoken，可选参数，搭配临时AK/SK使用，如果使用永久AK/SK或自定义认证信息获取类可忽略
spark.hadoop.lakeformation.authentication.security.token=securitytoken信息
```

说明

其中项目ID为必选配置，其他为可选配置，根据实际情况进行填写。

- 项目ID：可参考**获取项目ID**获取。
- LakeFormation实例ID：可参考**如何获取LakeFormation实例ID**获取。
- AK/SK信息可参考**如何获取AK/SK**获取。
- securitytoken可参考**通过token获取临时访问密钥和securitytoken**获取。

将上述配置添加到hive-site.xml或core-site.xml中亦可生效，添加时需要去除“spark.hadoop”前缀。

对接 OBS

在“spark/conf/spark-defaults.conf”中添加以下配置：

```
# 对接OBS固定配置，Endpoint需要根据区域进行配置
spark.hadoop.fs.obs.impl=org.apache.hadoop.fs.obs.OBSFileSystem
spark.hadoop.fs.AbstractFileSystem.obs.impl=org.apache.hadoop.fs.obs.OBS
spark.hadoop.fs.obs.endpoint=obs.xxx.huawei.com

# 指定访问OBS凭证获取类为LakeFormationObsCredentialProvider
spark.hadoop.fs.obs.credentials.provider=com.huawei.cloud.dalf.lakecat.client.obs.LakeFormationObsCredentialProvider

# 可选参数，关闭OBS文件系统缓存，长任务需要增加该配置，避免缓存中的临时AKSK失效
spark.hadoop.fs.obs.impl.disable.cache=true
```

说明

Endpoint：不同服务不同区域的Endpoint不同。您可以从**地区和终端节点**中获取。

将上述配置添加到core-site.xml文件中亦可生效，添加时需要去除“spark.hadoop”前缀。

对接 LakeFormation 元数据

Spark对接LakeFormation有以下两种对接方式，建议根据需要选择其中一种方式进行对接。

- 使用SparkCatalogPlugin方式对接：SparkCatalogPlugin方式对接基于Spark SessionCatalogV2，支持对接多个Catalog，社区仍在持续开发当中，存在部分不支持的SQL语句。
- 使用MetastoreClient方式对接：MetastoreClient方式对接基于Spark HiveExternalCatalog机制与Hive MetastoreClient机制，支持Hive大部分SQL语句，但无法同时对接多个Catalog。

使用SparkCatalogPlugin方式对接：

1. 在“spark/conf/spark-defaults.conf”配置文件中补充以下配置，如需对接多个catalog，以下配置需要配置多行：

```
# 指定catalog实现类，必选配置（spark_catalog_name为spark中的catalog名称。根据实际需要替换，下同）
spark.sql.catalog.$
{spark_catalog_name}=com.huawei.cloud.dalf.lakecat.client.spark.LakeFormationSparkCatalog
# 需要对接的Catalog名称(lakeformation_catalog_name为lakeFormation中的catalog)，可选配置，如果不配置则对接到hive catalog中，此处配置值仅作为参考
spark.sql.catalog.${spark_catalog_name}.lakecat.catalogname.default=${lakeformation_catalog_name}
```

2. 对接后验证。

   对接后可通过spark-shell、spark-submit或spark-sql访问LakeFormation，以下以spark-sql为例：

   - 切换数据库（切换时需要指定catalog名称，*database_name*对应数据库需要在LakeFormation中存在）：

     use *spark_catalog_name.database_name*;

   - 查看表信息：

     show tables;

   - 创建数据库（无法直接创建与catalog同名的数据库，需要指定catalog）

     create database *catalog_name.test*;

使用MetastoreClient方式对接：

1. 在spark-defaults.conf中补充如下配置：

```
spark.sql.catalogImplementation=hive
```

2. 在“spark/conf/”文件夹下新增文件hive-site.xml（如果已有该文件则编辑此文件），并在hive-site.xml中补充以下配置：

```
<configuration>
<!--固定配置，开启自定义metastore客户端-->
<property>
<name>hive.metastore.session.client.class</name>
<value>com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient</value>
</property>
<!--需要对接的lakeformationCatalog名称，可选配置，如果不配置则对接到hive catalog中，此处配置值仅作为参考-->
<property>
<name>lakecat.catalogname.default</name>
<value>hive</value>
</property>
<!--hive执行路径，可选配置，未对接HDFS时默认为本地路径/tmp/hive，此处配置值仅作为参考-->
<property>
<name>hive.exec.scratchdir</name>
<value>/tmp/hive</value>
</property>
</configuration>
```

除在hive-site.xml中添加配置外，也可通过在spark-defaults.conf配置文件中以“spark.hadoop”开头添加配置，如添加“spark.hadoop.hive.metastore.session.client.class=com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient”。

说明

- “hive.exec.scratchdir”路径权限需要修改为777，否则会导致Hive客户端初始化异常。
- 需要在“lakecat.catalogname.default”对应的catalog中创建名称为“default”的数据库（已创建可忽略），否则会导致spark-sql初始化异常或spark-shell无法对接。

3. 对接后验证。

   对接后可通过spark-shell或执行SQL访问LakeFormation，以下spark-sql为例。

   - 切换数据库（切换时无需指定catalog名称）：

     use database_name;

   - 查看表信息：

     show tables;

集成 SQL 鉴权插件

步骤1 使用鉴权插件必须实现并指定自定义用户信息获取类，详细操作请参考**自定义用户信息获取类**。

步骤2 在spark-default.conf配置文件中添加如下配置：

```
com.huawei.cloud.dalf.lakecat.client.spark.v31.authorizer.LakeFormationSparkSQLExtension
spark.sql.extensions=com.huawei.cloud.dalf.lakecat.client.spark.authorizer.LakeFormationSparkSQLExtension
```

----结束

说明

- 集成权限插件后，如果当前用户（通过**自定义用户信息获取类**指定）无对应元数据权限，在执行SQL时将抛出异常。
- 当前用户如果拥有IAM LakeFormation:policy:create权限，且当前用户（可通过**自定义用户信息获取类**指定）和认证信息（可通过**自定义认证信息获取类**指定）为统一用户，将跳过SQL鉴权。
- 当前，过滤相关功能暂未支持，包括库、表、行过滤，列掩码等。

日志打印

通过在log4j.properties中添加配置“log4j.logger.org.apache=WARN”，可关闭LakeFormation客户端httpclient请求日志。

1.3 对接后二次开发

用户可根据需要进行二次开发，当前提供以下样例：

- 自定义认证信息获取类：用于获取访问LakeFormation服务的IAM认证信息。
- 自定义用户信息获取类：用于获取当前访问LakeFormation的用户。

自定义认证信息获取类

认证信息获取类（IdentityGenerator）用于获取访问LakeFormation服务的IAM认证信息（Token、永久AK/SK，临时AK/SK+securityToken）。

LakeFormation提供了默认的认证信息获取类，通过从配置文件中获取AKSK生成认证信息。

除LakeFormation提供的默认认证信息获取类外，可选择自行实现默认认证信息获取类。

1. 代码开发。

   实现工程参考如下，在Maven工程pom文件中添加lakeformation-lakecat-client依赖：

   ```
   <dependency>
   <groupId>com.huawei.lakeformation</groupId>
   <artifactId>lakeformation-lakecat-client</artifactId>
   <version>${lakeformation.version}</version>
   </dependency>
   新增认证信息获取类，实现IdentityGenerator接口

   /*
   * Copyright (c) Huawei Technologies Co., Ltd. 2023-2023. All rights reserved.
   */

   package com.huawei.cloud.dalf.lakecat.examples;

   import com.huawei.cloud.dalf.lakecat.client.ConfigCenter;
   import com.huawei.cloud.dalf.lakecat.client.identity.Identity;
   import com.huawei.cloud.dalf.lakecat.client.identity.IdentityGenerator;

   import java.util.Collections;

   /**
   * 身份信息生成器样例
   *
   */
   public class LakeFormationExampleIdentityGenerator implements IdentityGenerator {
   public String token;

   @Override
   public void initialize(ConfigCenter configCenter) {
   //初始化
   }

   @Override
   public Identity generateIdentity() {
   //返回IAM认证信息
   }
   }
   ```

2. 集成配置。

   代码通过Maven打包后将jar包放置在“spark/jars”目录下。

   根据对接方式不同，补充以下配置：

   - 使用SparkCatalogPlugin方式对接时，在spark-default.conf配置文件中补充以下配置：

     ```
     # 认证信息获取类，根据实现类路径填写，此处配置值仅作为参考
     spark.sql.catalog.catalog_name.lakecat.auth.identity.util.class=com.huawei.cloud.dalf.lakecat.client.spark.v31.impl.SparkDefaultIdentityGenerator
     ```

   - 使用MetastoreClient方式对接时，选择以下配置方式。

     在spark-default.conf补充以下配置：

```
# 认证信息获取类，根据实现类路径填写，此处配置值仅作为参考
spark.hadoop.lakecat.auth.identity.util.class=com.huawei.cloud.dalf.lakecat.client.spark.v31.impl.SparkDefaultIdentityGenerator
```

或在hive-site.xml补充以下配置：

```
<!--认证信息获取类，此处配置值仅作为参考-->
<property>
<name>lakecat.auth.identity.util.class</name>
<value>com.huawei.cloud.dalf.lakecat.examples.LakeFormationExampleIdentityGenerator</value>
</property>
```

自定义用户信息获取类

用户信息获取类（AuthenticationManager）用于获取当前访问LakeFormation的用户，可能为IAM用户或本地LDAP用户。默认用户信息获取类通过UserGroupInformation.getCurrentUser()获取当前用户。

除默认用户信息获取类外，服务可选择自行实现用户信息获取类。

说明

如使用用户认证信息访问LakeFormation，用户信息和用户身份信息需要保持一致（即用户名、来源需要保持一致）。

1. 代码开发。

   实现工程参考如下，在Maven工程pom文件中添加lakeformation-lakecat-client依赖：

   ```
   <dependency>
   <groupId>com.huawei.lakeformation</groupId>
   <artifactId>lakeformation-lakecat-client</artifactId>
   <version>${lakeformation.version}</version>
   </dependency>
   用户信息获取类，实现AuthenticationManager接口

   /*
   * Copyright (c) Huawei Technologies Co., Ltd. 2023-2023. All rights reserved.
   */

   package com.huawei.cloud.dalf.lakecat.examples;

   import com.huawei.cloud.dalf.lakecat.client.ConfigCenter;
   import com.huawei.cloud.dalf.lakecat.client.identity.AuthenticationManager;
   import com.huawei.cloud.dalf.lakecat.client.model.Principal;

   public class ExampleAuthenticationManager implements AuthenticationManager {
   @Override
   public void initialize(ConfigCenter configCenter) {
   //初始化
   }

   @Override
   public Principal getCurrentUser() {
   //返回当前用户信息
   }
   }
   ```

2. 集成配置。

   代码通过Maven打包后将jar包放置在“spark/jars”目录下。

   根据对接方式不同，补充以下配置：

   - 使用SparkCatalogPlugin方式对接时，在spark-default.conf配置文件中补充以下配置：

     ```
     # 可选参数，认证管理器实现类，用于获取当前用户，此处配置值仅作为参考
     spark.sql.catalog.catalog_name.lakeformation.authentication.manager.class=com.huawei.cloud.d
     ```

```
alf.lakecat.examples.ExampleAuthenticationManager
# 可选参数，是否开启owner指定，开启后创建资源时将使用当前用户作为资源owner，默认为false
spark.sql.catalog.catalog_name.lakeformation.owner.designate=true
```

- 使用MetastoreClient方式对接时，可选择以下配置方式：

  在spark-default.conf补充以下配置：

```
# 可选参数，认证管理器实现类，用于获取当前用户，此处配置值仅作为参考
spark.hadoop.lakeformation.authentication.manager.class=com.huawei.cloud.dalf.lakecat.examples.ExampleAuthenticationManager
# 可选参数，是否开启owner指定，开启后创建资源时将使用当前用户作为资源owner，默认为false
spark.hadoop.lakeformation.owner.designate=true
```

  或在hive-site.xml补充以下配置：

```
<!--可选参数，认证管理器实现类，用于获取当前用户，此处配置值仅作为参考-->
<property>
<name>lakeformation.authentication.manager.class</name>
<value>com.huawei.cloud.dalf.lakecat.examples.ExampleAuthenticationManager</value>
</property>
<!--可选参数，是否开启owner指定，开启后创建资源时将使用当前用户作为资源owner，默认为false-->
<property>
<name>lakeformation.owner.designate</name>
<value>true</value>
</property>
</configuration>
```

2 配置开源 Hive 组件对接 LakeFormation

2.1 环境准备

在配置开源Hive组件对接LakeFormation前，需要提前准备以下信息：

步骤1 准备可用的开源Hive环境，目前支持Hive 2.3以及Hive 3.1两个版本，并安装Git环境。

步骤2 准备LakeFormation实例，详细操作请参考**创建LakeFormation实例**。

步骤3 创建LakeFormation接入客户端，并与Hive在相同的虚拟私有云、子网下，详细操作请参考**管理接入客户端**。

步骤4 准备开发环境，详细操作请参考**准备开发程序环境**中“准备开发环境”部分，其中“安装和配置IntelliJ IDEA”为可选。

步骤5 准备LakeFormation客户端。

- **方式一：下载客户端发行版**

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lake-formation-lakecat-sdk-java/releases**

  根据Spark、Hive版本下载对应客户端（Hive版本为2.3.9，则下载lakeformation-lakecat-client-hive2.3-v1.0.0.rar）

  校验压缩包：下载后在Windows环境下执行**certutil -hashfile** *<压缩包>* **sha256**，检查回显信息与对应sha256文件内容是否一致。

- **方式二：本地编译客户端**

  a. 获取客户端代码。

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lake-formation-lakecat-sdk-java**。

  并在Git中执行以下命令将分支切换为“master_dev”：

  git checkout master_dev

  b. 配置maven源，详细操作请参考**获取SDK并配置maven**。

  c. 获取以下jar包以及对应的pom文件，并将文件放入maven本地仓库。

  例如本地仓库地址为“D:\maven\repository”，则放入“D:\maven\repository\com\huaweicloud\hadoop-huaweicloud\3.1.1-hw-53.8”目录下。

- jar包：**https://github.com/huaweicloud/obsa-hdfs/blob/master/release/hadoop-huaweicloud-3.1.1-hw-53.8.jar**
- pom文件：**https://github.com/huaweicloud/obsa-hdfs/blob/master/hadoop-huaweicloud/pom.xml**，并改名为hadoop-huaweicloud-3.1.1-hw-53.8.pom。

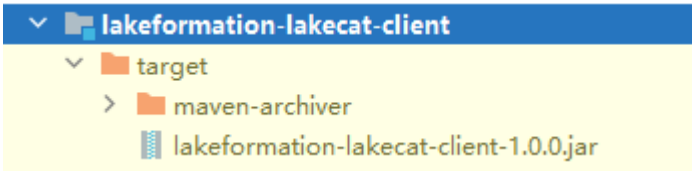
d. 编译打包客户端代码。

根据Hive版本，进入客户端工程目录，执行以下打包命令：

mvn clean install -DskipTests=true -P"*${HIVE_PROFILE}*"

“HIVE_PROFILE”：可选择为“hive-2.3”或者“hive-3.1”。

e. 打包完成后可以在“lakeformation-lakecat-client”的target目录下获取“lakeformation-lakecat-client-1.0.0.jar”。



步骤6 准备Hive内核相关的jar包。

- **方式一：下载预构建Hive相关jar包**

  获取地址为：**https://gitee.com/HuaweiCloudDeveloper/huaweicloud-lake-formation-lakecat-sdk-java/releases**

  根据Hive版本下载对应客户端（Hive版本为2.3.9，则下载hive-exec-2.3.9.jar、hive-common-2.3.9.jar）

- **方式二：本地编译Hive相关jar包**

  Windows系统下需要在WSL开发环境下进行maven相关操作。

  a. 根据Hive版本下载Hive源码。

  例如Hive内核版本号为2.3.9，则下载链接为：**https://github.com/apache/hive/tree/rel/release-2.3.9**

  b. 将LakeFormation的客户端代码中的patch应用到Hive源码。

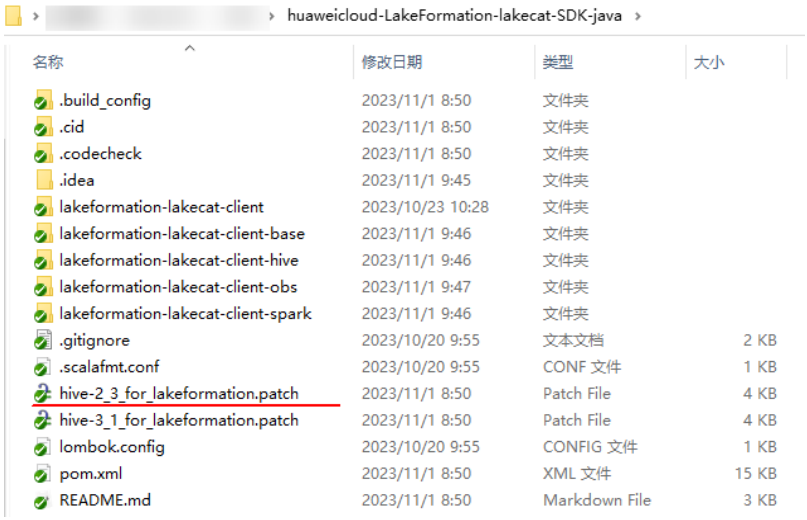
  i. 根据需要切换Hive源码分支，例如Hive内核版本号为2.3.9，则执行如下命令：

  git checkout rel/release-*2.3.9*

  ii. 执行以下命令，将patch应用到切换分支后的Hive源码工程。

  mvn patch:apply -DpatchFile=${*your patch file location*}

  其中，“your patch file location”为hive-2_3_for_lakeformation.patch或hive-3_1_for_lakeformation.patch的存储路径。patch文件可在客户端工程中获取，如下图所示：



iii. 执行以下命令重新编译Hive内核源码。

mvn clean install -DskipTests=true

步骤7 准备并补充替换Hive内核相关的jar包。

获取以下jar包，并补充或替换到Hive安装环境的lib目录下。

表 2-1 获取 Hive 环境所需 jar 包

序号	jar包名称	获取途径
1	spring-web-5.3.24.jar	**https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-web/5.3.24/spring-web-5.3.24.jar**
2	spring-core-5.3.24.jar	**https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-core/5.3.24/spring-core-5.3.24.jar**
3	spring-context-5.3.24.jar	**https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-context/5.3.24/spring-context-5.3.24.jar**
4	spring-beans-5.3.24.jar	**https://mirrors.huaweicloud.com/repository/maven/org/springframework/spring-beans/5.3.24/spring-beans-5.3.24.jar**

序号	jar包名称	获取途径
5	caffeine-2.9.3.jar	https://mirrors.huaweicloud.com/repository/maven/com/github/ben-manes/caffeine/caffeine/2.9.3/caffeine-2.9.3.jar
6	mapstruct-1.5.3.Final.jar	https://mirrors.huaweicloud.com/repository/maven/org/mapstruct/mapstruct/1.5.3.Final/mapstruct-1.5.3.Final.jar
7	httpcore-4.4.13.jar （如果Hive内核版本为3.1，则不需要该jar包。）	https://mirrors.huaweicloud.com/repository/maven/org/apache/httpcomponents/httpcore/4.4.13/httpcore-4.4.13.jar
8	jca-1.0.4.jar	https://mirrors.huaweicloud.com/repository/maven/org/openeuler/jca/1.0.4/jca-1.0.4.jar
9	将commons-codec-1.4.jar替换为commons-codec-1.15.jar （如果Hive内核版本为3.1，则不需要该jar包。）	https://mirrors.huaweicloud.com/repository/maven/commons-codec/commons-codec/1.15/commons-codec-1.15.jar
10	java-sdk-core-3.2.4.jar （如果仅使用自定义认证信息获取类使用Token认证，则无需补充该jar包。）	https://mirrors.huaweicloud.com/repository/maven/huaweicloudsdk/com/huawei/apigateway/java-sdk-core/
11	hadoop-huaweicloud-3.1.1-hw-53.8.jar	https://github.com/huaweicloud/obsa-hdfs/blob/master/release/hadoop-huaweicloud-3.1.1-hw-53.8.jar
12	lakeformation-lakecat-client-1.0.0.jar	步骤5中获取。
13	hive-exec-${version}.jar	步骤6中获取。
14	hive-common-${version}.jar	步骤6中获取。

----结束

2.2 配置 Hive 对接 LakeFormation

步骤1 修改Hive服务端安装环境的“conf”目录的hive-site.xml文件，添加如下内容（部分参数值根据提示进行替换）：

```
<property>
<name>hive.metastore.session.client.class</name>
<value>com.huawei.cloud.dalf.lakecat.client.hiveclient.LakeCatMetaStoreClient</value>
</property>
<property>
<name>lakeformation.project.id</name>
<value>****</value>
</property>
<property>
<name>lakeformation.instance.id</name>
<value>LakeFormation实例ID</value>
</property>
<!--访问lakeformation IAM认证AK信息，可选参数，如果为自定义认证信息获取类可忽略-->
<property>
<name>lakeformation.authentication.access.key</name>
<value>AK信息</value>
</property>
<!--访问lakeformation IAM认证SK信息，可选参数，如果为定义认证信息获取类可忽略-->
<property>
<name>lakeformation.authentication.secret.key</name>
<value>SK信息</value>
</property>
<!--访问lakeformation IAM认证信息securityToken，可选参数，如果使用永久AKSK或自定义认证信息获取类可忽略-->
<property>
<name>lakeformation.authentication.security.token</name>
<value>securityToken信息</value>
</property>
<property>
<name>fs.obs.impl</name>
<value>org.apache.hadoop.fs.obs.OBSFileSystem</value>
</property>
<property>
<name>fs.AbstractFileSystem.obs.impl</name>
<value>org.apache.hadoop.fs.obs.OBS</value>
</property>
<property>
<name>fs.obs.endpoint</name>
<value>Endpoint信息</value>
</property>
<property>
<name>fs.obs.credentials.provider</name>
<value>com.huawei.cloud.dalf.lakecat.client.obs.LakeFormationObsCredentialProvider</value>
</property>
<property>
<name>fs.obs.impl.disable.cache</name>
<value>true</value>
</property>
<property>
<name>dfs.namenode.acls.enabled</name>
<value>false</value>
</property>
<!--需要对接的LakeFormation Catalog名称，可选配置，如果不配置则对接到hive catalog中，此处配置值仅作为参考-->
<property>
<name>lakecat.catalogname.default</name>
<value>hive</value>
</property>
```

说明

- lakeformation.project.id：为项目ID，可参考**获取项目ID**获取。
- lakeformation.instance.id：为LakeFormation实例ID，可参考**如何获取LakeFormation实例ID**获取。
- AK/SK信息可参考**如何获取AK/SK**获取。
- securitytoken可参考**通过token获取临时访问密钥和securitytoken**获取。

步骤2 重启Hive服务。

步骤3 进入Hive客户端执行以下命令进行验证。

show tables;

----结束

2.3 对接后二次开发

用户可根据需要进行二次开发，当前提供以下样例：

- 自定义认证信息获取类：用于获取访问LakeFormation服务的IAM认证信息。
- 自定义用户信息获取类：用于获取当前访问LakeFormation的用户。

自定义认证信息获取类

认证信息获取类（IdentityGenerator）用于获取访问LakeFormation服务的IAM认证信息（Token、永久AK/SK，临时AK/SK+securityToken）。

LakeFormation提供了默认的认证信息获取类，通过从配置文件中获取AKSK生成认证信息。

除LakeFormation提供的默认认证信息获取类外，可选择自行实现默认认证信息获取类。

1. 代码开发。

   实现工程参考如下，在Maven工程pom文件中添加lakeformation-lakecat-client依赖：

```
<dependency>
<groupId>com.huawei.lakeformation</groupId>
<artifactId>lakeformation-lakecat-client</artifactId>
<version>${lakeformation.version}</version>
</dependency>
//新增认证信息获取类，实现IdentityGenerator接口
/*
* Copyright (c) Huawei Technologies Co., Ltd. 2023-2023. All rights reserved.
*/

package com.huawei.cloud.dalf.lakecat.examples;

import com.huawei.cloud.dalf.lakecat.client.ConfigCenter;
import com.huawei.cloud.dalf.lakecat.client.identity.Identity;
import com.huawei.cloud.dalf.lakecat.client.identity.IdentityGenerator;

import java.util.Collections;

/**
* 身份信息生成器样例
*
*/
```

```
public class LakeFormationExampleIdentityGenerator implements IdentityGenerator {
public String token;

@Override
public void initialize(ConfigCenter configCenter) {
//初始化
}

@Override
public Identity generateIdentity() {
//返回IAM认证信息
}
}
```

2. 集成配置。

   代码通过Maven打包后将jar包放置在“hive-*xxx*/lib”目录下。*xxx*为Hive内核版本号。

   并在hive-site.xml补充以下配置：

```
<!--认证信息获取类，此处配置值仅作为参考-->
<property>
<name>lakecat.auth.identity.util.class</name>
<value>com.huawei.cloud.dalf.lakecat.examples.LakeFormationExampleIdentityGenerator</value>
</property>
```

3. 重启Hive服务。

自定义用户信息获取类

用户信息获取类（AuthenticationManager）用于获取当前访问LakeFormation的用户，可能为IAM用户或本地LDAP用户。默认用户信息获取类通过UserGroupInformation.getCurrentUser()获取当前用户。

除默认用户信息获取类外，服务可选择自行实现用户信息获取类。

说明

如果使用用户认证信息访问LakeFormation，用户信息和用户身份信息需要保持一致（即用户名、来源需要保持一致）。

1. 代码开发。

   实现工程参考如下，在Maven工程pom文件中添加lakeformation-lakecat-client依赖：

```
<dependency>
<groupId>com.huawei.lakeformation</groupId>
<artifactId>lakeformation-lakecat-client</artifactId>
<version>${lakeformation.version}</version>
</dependency>
用户信息获取类，实现AuthenticationManager接口

/*
* Copyright (c) Huawei Technologies Co., Ltd. 2023-2023. All rights reserved.
*/

package com.huawei.cloud.dalf.lakecat.examples;

import com.huawei.cloud.dalf.lakecat.client.ConfigCenter;
import com.huawei.cloud.dalf.lakecat.client.identity.AuthenticationManager;
import com.huawei.cloud.dalf.lakecat.client.model.Principal;

public class ExampleAuthenticationManager implements AuthenticationManager {
@Override
public void initialize(ConfigCenter configCenter) {
//初始化
}
```

```
@Override
public Principal getCurrentUser() {
//返回当前用户信息
}
}
```

2. 集成配置。

   代码通过Maven打包后将jar包放置在“hive-*xxx*/lib”目录下。*xxx*为Hive内核版本号。

   并在hive-site.xml补充以下配置：

```
<!--可选参数，认证管理器实现类，用于获取当前用户，此处配置值仅作为参考-->
<property>
<name>lakeformation.authentication.manager.class</name>
<value>com.huawei.cloud.dalf.lakecat.examples.ExampleAuthenticationManager</value>
</property>

<!--可选参数，是否开启owner指定，开启后创建资源时将使用当前用户作为资源owner，默认为false-->
<property>
<name>lakeformation.owner.designate</name>
<value>true</value>
</property>
</configuration>
```

3. 重启Hive服务。