尚硅谷大数据项目之尚品汇（元数据管理）

(作者：尚硅谷研究院)

版本：V4.0

第 1 章 Atlas 入门

1.1 Atlas 概述

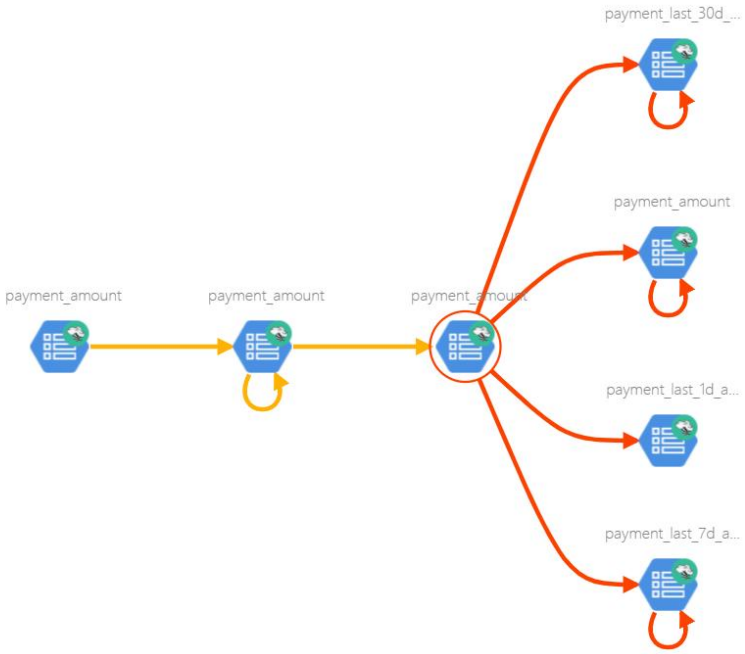
Apache Atlas 为组织提供开放式元数据管理和治理功能，用以构建其数据资产目录，对这些资产进行分类和管理，并为数据分析师和数据治理团队，提供围绕这些数据资产的协作功能。

Atlas 的具体功能如下：

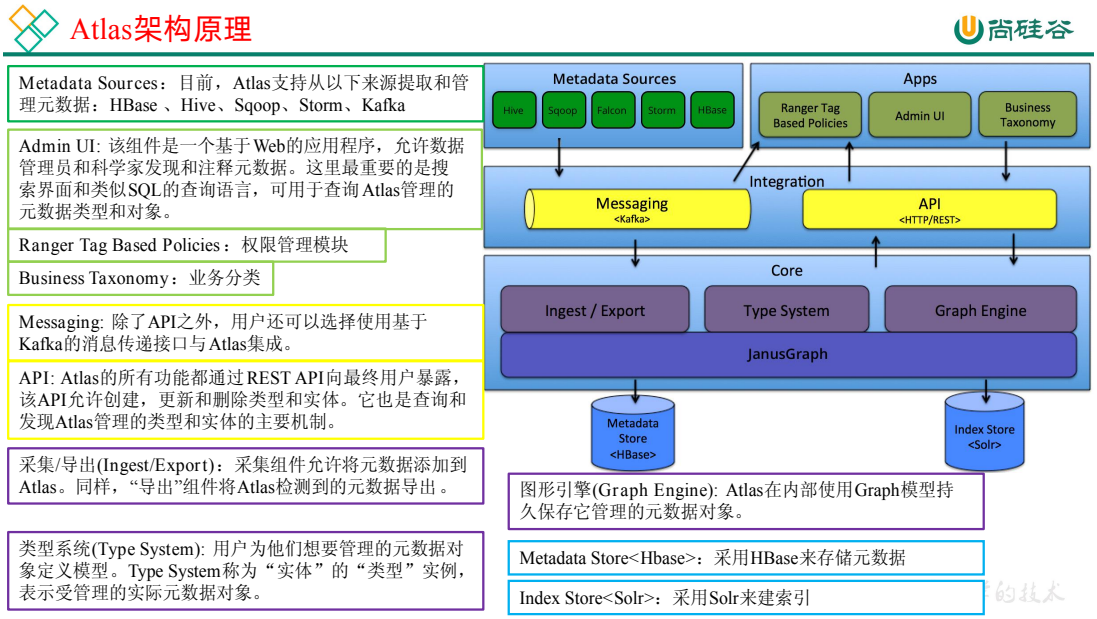
元数据分类	支持对元数据进行分类管理，例如个人信息，敏感信息等
元数据检索	可按照元数据类型、元数据分类进行检索，支持全文检索
血缘依赖	支持表到表和字段到字段之间的血缘依赖，便于进行问题回溯和影响分析等

1）表与表之间的血缘依赖

2）字段与字段之间的血缘依赖



1.2 Atlas 架构原理



第 2 章 Atlas 安装

1）Atlas 官网地址：https://atlas.apache.org/

2）文档查看地址：https://atlas.apache.org/2.1.0/index.html

3）下载地址：https://www.apache.org/dyn/closer.cgi/atlas/2.1.0/apache-atlas-2.1.0-sources.tar.gz

2.1 安装环境准备

Atlas 安装分为：集成自带的 HBase + Solr；集成外部的 HBase + Solr。通常企业开发中选择集成外部的 HBase + Solr，方便项目整体进行集成操作。

以下是 Atlas 所以依赖的环境及集群规划。本文只包含 Solr 和 Atlas 的安装指南，其余所依赖服务的安装请参考前边章节。

服务名称	子服务	服务器 hadoop102	服务器 hadoop103	服务器 hadoop104
JDK		√	√	√
Zookeeper	QuorumPeerMain	√	√	√
Kafka	Kafka	√	√	√
HBase	HMaster	√		
	HRegionServer	√	√	√
Solr	Jar	√	√	√
Hive	Hive	√		
Atlas	atlas	√		
服务数总计		13	7	7

2.1.1 安装 Solr-7.7.3

1.在每台节点创建系统用户 solr

```
[root@hadoop102 ~]# useradd solr
[root@hadoop102 ~]# echo solr | passwd --stdin solr

[root@hadoop103 ~]# useradd solr
[root@hadoop103 ~]# echo solr | passwd --stdin solr

[root@hadoop104 ~]# useradd solr
[root@hadoop104 ~]# echo solr | passwd --stdin solr
```

2.解压 solr-7.7.3.tgz 到/opt/module 目录，并改名为 solr

```
[root@hadoop102 ~]# tar -zxvf solr-7.7.3.tgz -C /opt/module/
[root@hadoop102 ~]# mv solr-7.7.3/ solr
```

3.修改 solr 目录的所有者为 solr 用户

```
[root@hadoop102 ~]# chown -R solr:solr /opt/module/solr
```

4.修改 solr 配置文件

修改/opt/module/solr/bin/solr.in.sh 文件中的以下属性

```
ZK_HOST="hadoop102:2181,hadoop103:2181,hadoop104:2181"
```

5.分发 solr

```
[root@hadoop102 ~]# xsync /opt/module/solr
```

6.启动 solr 集群

1）启动 Zookeeper 集群

```
[root@hadoop102 ~]# zk.sh start
```

2）启动 solr 集群

出于安全考虑，不推荐使用 root 用户启动 solr，此处使用 solr 用户，在所有节点执行以下命令启动 solr 集群

```
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr start
[root@hadoop103 ~]# sudo -i -u solr /opt/module/solr/bin/solr start
[root@hadoop104 ~]# sudo -i -u solr /opt/module/solr/bin/solr start
```

出现 **Happy Searching!** 字样表明启动成功。

```
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr start
*** [WARN] *** Your open file limit is currently 1024.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
*** [WARN] ***  Your Max Processes Limit is currently 4096.
 It should be set to 65000 to avoid operational disruption.
 If you no longer wish to see this warning, set SOLR_ULIMIT_CHECKS to false in your profile or solr.in.sh
Waiting up to 180 seconds to see Solr running on port 8983 [\]
Started Solr server on port 8983 (pid=32155). Happy searching!
```

说明：上述警告内容是：solr 推荐系统允许的最大进程数和最大打开文件数分别为 65000 和 65000，而系统默认值低于推荐值。如需修改可参考以下步骤，修改完需要重启方可生效，**此处可暂不修改。**

（1）修改打开文件数限制

修改/etc/security/limits.conf 文件，增加以下内容

```
* soft nofile 65000
* hard nofile 65000
```

（2）修改进程数限制

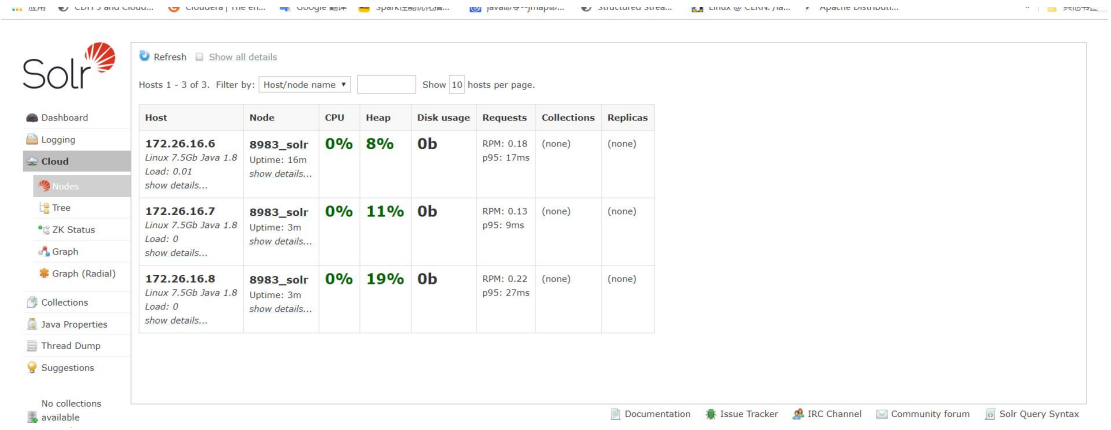
修改/etc/security/limits.d/20-nproc.conf 文件

```
*          soft    nproc     65000
```

（3）重启服务器

7.访问 web 页面

默认端口为 8983，可指定三台节点中的任意一台 IP，http://hadoop102:8983



提示：UI 界面出现 Cloud 菜单栏时，Solr 的 Cloud 模式才算部署成功。

2.1.8 安装 Atlas2.1.0

1.把 apache-atlas-2.1.0-server.tar.gz 上传到 hadoop102 的/opt/software 目录下

2.解压 apache-atlas-2.1.0-server.tar.gz 到/opt/module/目录下面

```
[root@hadoop102 software]# tar -zxvf
apache-atlas-2.1.0-server.tar.gz -C /opt/module/
```

3.修改 apache-atlas-2.1.0 的名称为 atlas

```
[root@hadoop102 ~]# mv /opt/module/apache-atlas-2.1.0
/opt/module/atlas
```

2.2 Atlas 配置

2.2.1 Atlas 集成 Hbase

1.修改/opt/module/atlas/conf/atlas-application.properties 配置文件中的以下参数

```
atlas.graph.storage.hostname=hadoop102:2181,hadoop103:2181,hadoop
104:2181
```

2.修改/opt/module/atlas/conf/atlas-env.sh 配置文件，增加以下内容

```
export HBASE_CONF_DIR=/opt/module/hbase/conf
```

2.2.2 Atlas 集成 Solr

1.修改/opt/module/atlas/conf/atlas-application.properties 配置文件中的以下参数

```
atlas.graph.index.search.backend=solr
atlas.graph.index.search.solr.mode=cloud
atlas.graph.index.search.solr.zookeeper-url=hadoop102:2181,hadoop
103:2181,hadoop104:2181
```

2.创建 solr collection

```
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr create
-c vertex_index -d /opt/module/atlas/conf/solr -shards 3
-replicationFactor 2
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr create
-c edge_index -d /opt/module/atlas/conf/solr -shards 3
-replicationFactor 2
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr create
-c fulltext_index -d /opt/module/atlas/conf/solr -shards 3
-replicationFactor 2
```

2.2.3 Atlas 集成 Kafka

修改/opt/module/atlas/conf/atlas-application.properties 配置文件中的以下参数

```
atlas.notification.embedded=false
atlas.kafka.data=/opt/module/kafka/data
atlas.kafka.zookeeper.connect=
hadoop102:2181,hadoop103:2181,hadoop104:2181/kafka
atlas.kafka.bootstrap.servers=hadoop102:9092,hadoop103:9092,hadoo
p104:9092
```

2.2.4 Atlas Server 配置

1.修改/opt/module/atlas/conf/atlas-application.properties 配置文件中的以下参数

```
#########  Server Properties  #########
atlas.rest.address=http://hadoop102:21000
# If enabled and set to true, this will run setup steps when the server
starts
atlas.server.run.setup.on.start=false

#########  Entity Audit Configs  #########
atlas.audit.hbase.zookeeper.quorum=hadoop102:2181,hadoop103:2181,
hadoop104:2181
```

2.记录性能指标，进入/opt/module/atlas/conf/路径，修改当前目录下的 atlas-log4j.xml

```
[root@hadoop101 conf]# vim atlas-log4j.xml

#去掉如下代码的注释
<appender                                                    name="perf_appender"
class="org.apache.log4j.DailyRollingFileAppender">
   <param name="file" value="${atlas.log.dir}/atlas_perf.log" />
   <param name="datePattern" value="'.'yyyy-MM-dd" />
   <param name="append" value="true" />
   <layout class="org.apache.log4j.PatternLayout">
      <param name="ConversionPattern" value="%d|%t|%m%n" />
   </layout>
</appender>
```

```
<logger name="org.apache.atlas.perf" additivity="false">
   <level value="debug" />
   <appender-ref ref="perf_appender" />
</logger>
```

2.2.5 Kerberos 相关配置

若 Hadoop 集群开启了 Kerberos 认证，Atlas 与 Hadoop 集群交互之前就需要先进行 Kerberos 认证。若 Hadoop 集群未开启 Kerberos 认证，则本节可跳过。

1.为 Atlas 创建 Kerberos 主体，并生成 keytab 文件

```
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"addprinc -randkey
atlas/hadoop102"
[root@hadoop102 ~]# kadmin -padmin/admin -wadmin -q"xst -k
/etc/security/keytab/atlas.service.keytab atlas/hadoop102"
```

2.修改/opt/module/atlas/conf/atlas-application.properties 配置文件，增加以下参数

```
atlas.authentication.method=kerberos
atlas.authentication.principal=atlas/hadoop102@EXAMPLE.COM
atlas.authentication.keytab=/etc/security/keytab/atlas.service.ke
ytab
```

2.2.6 Atlas 集成 Hive

1.安装 Hive Hook

1）解压 Hive Hook

```
[root@hadoop102 ~]# tar -zxvf apache-atlas-2.1.0-hive-hook.tar.gz
```

2）将 Hive Hook 依赖复制到 Atlas 安装路径

```
[root@hadoop102 ~]# cp -r apache-atlas-hive-hook-2.1.0/*
/opt/module/atlas/
```

3）修改/opt/module/hive/conf/hive-env.sh 配置文件

注：需先需改文件名

```
[root@hadoop102 ~]# mv hive-env.sh.template hive-env.sh
```

增加如下参数

```
export HIVE_AUX_JARS_PATH=/opt/module/atlas/hook/hive
```

2.修改 Hive 配置文件，在/opt/module/hive/conf/hive-site.xml 文件中增加以下参数，配置 Hive Hook。

```
<property>
     <name>hive.exec.post.hooks</name>
     <value>org.apache.atlas.hive.hook.HiveHook</value>
</property>
```

3.修改/opt/module/atlas/conf/atlas-application.properties 配置文件中的以下参数

```
######### Hive Hook Configs #######
atlas.hook.hive.synchronous=false
atlas.hook.hive.numRetries=3
atlas.hook.hive.queueSize=10000
atlas.cluster.name=primary
```

4）将 Atlas 配置文件/opt/module/atlas/conf/atlas-application.properties

拷贝到/opt/module/hive/conf 目录

```
[root@hadoop102 ~]# cp
/opt/module/atlas/conf/atlas-application.properties
/opt/module/hive/conf/
```

2.4 Atlas 启动

1.启动 Atlas 所依赖的环境

1）启动 Hadoop 集群

（1）在 NameNode 节点执行以下命令，启动 HDFS

```
[root@hadoop102 ~]# start-dfs.sh
```

（2）在 ResourceManager 节点执行以下命令，启动 Yarn

```
[root@hadoop103 ~]# start-yarn.sh
```

2）启动 Zookeeper 集群

```
[root@hadoop102 ~]# zk.sh start
```

3）启动 Kafka 集群

```
[root@hadoop102 ~]# kf.sh start
```

4）启动 Hbase 集群

在 HMaster 节点执行以下命令，使用 hbase 用户启动 HBase

```
[root@hadoop102 ~]# sudo -i -u hbase start-hbase.sh
```

5）启动 Solr 集群

在所有节点执行以下命令，使用 solr 用户启动 Solr

```
[root@hadoop102 ~]# sudo -i -u solr /opt/module/solr/bin/solr
start
[root@hadoop103 ~]# sudo -i -u solr /opt/module/solr/bin/solr
start
[root@hadoop104 ~]# sudo -i -u solr /opt/module/solr/bin/solr
start
```

6）进入/opt/module/atlas 路径，启动 Atlas 服务

```
[root@hadoop102 atlas]# bin/atlas_start.py
```

提示：

（1）错误信息查看路径：/opt/module/atlas/logs/*.out 和 application.log

（2）停止 Atlas 服务命令为 atlas_stop.py

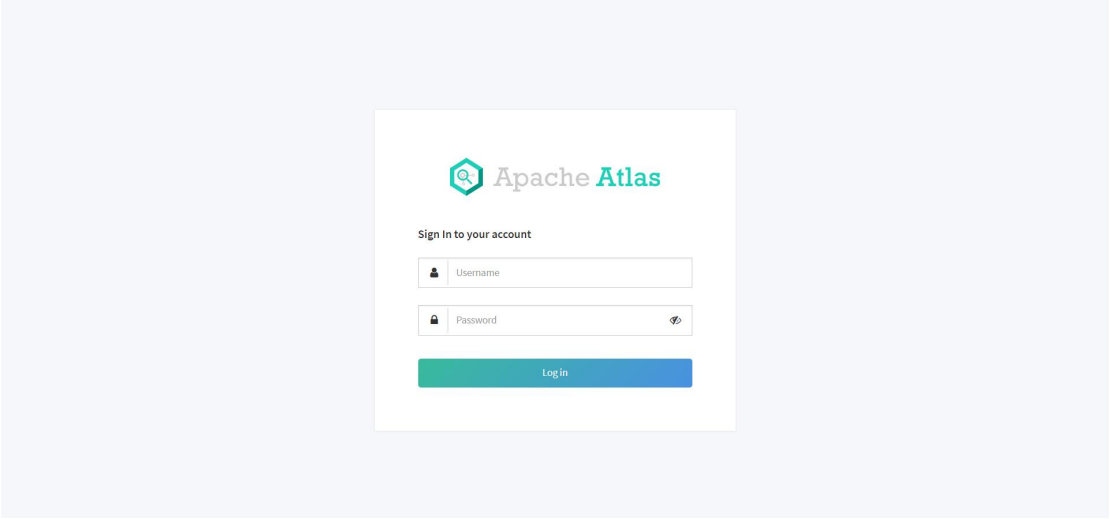
7）访问 Atlas 的 WebUI

访问地址：http://hadoop102:21000

注意：等待若干分钟。

账户：admin

密码：admin



第 3 章 Atlas 使用

Atlas 的使用相对简单，其主要工作是同步各服务（主要是 Hive）的元数据，并构建元数据实体之间的关联关系，然后对所存储的元数据建立索引，最终未用户提供数据血缘查看及元数据检索等功能。

Atlas 在安装之初，需手动执行一次元数据的全量导入，后续 Atlas 便会利用 Hive Hook 增量同步 Hive 的元数据。

3.1 Hive 元数据初次导入

Atlas 提供了一个 Hive 元数据导入的脚本，直接执行该脚本，即可完成 Hive 元数据的初次全量导入。

1.导入 Hive 元数据

执行以下命令

```
[root@hadoop102 ~]# /opt/module/atlas/hook-bin/import-hive.sh
```

按提示输入用户名：admin；输入密码：admin

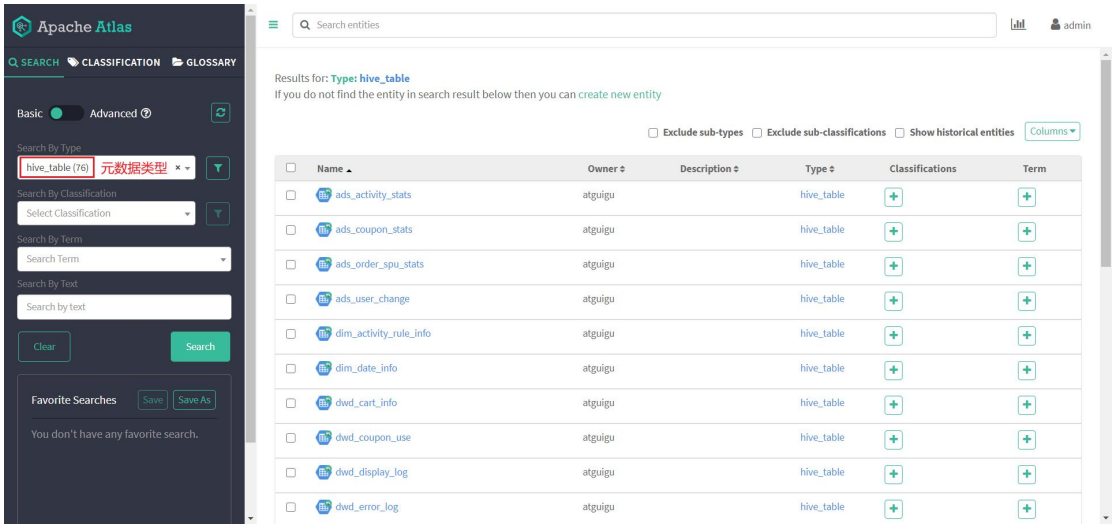
```
Enter username for atlas :- admin
Enter password for atlas :-
```

等待片刻，出现以下日志，即表明导入成功

```
Hive Meta Data import was successful!!!
```

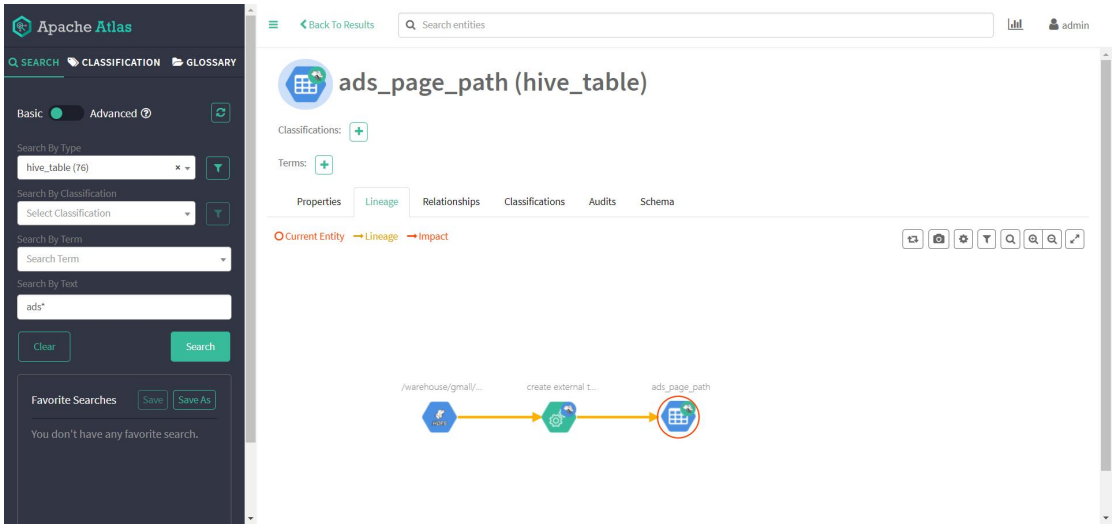
2.查看 Hive 元数据

1）搜索 hive_table 类型的元数据，可已看到 Atlas 已经拿到了 Hive 元数据



2）任选一张表查看血缘依赖关系

发现此时并未出现期望的血缘依赖，原因是 Atlas 是根据 Hive 所执行的 SQL 语句获取表与表之间以及字段与字段之间的依赖关系的，例如执行 insert into table_a select * from table_b 语句，Atlas 就能获取 table_a 与 table_b 之间的依赖关系。此时并未执行任何 SQL 语句，故还不能出现血缘依赖关系。



3.2 Hive 元数据增量同步

Hive 元数据的增量同步，无需人为干预，只要 Hive 中的元数据发生变化（执行 DDL 语句），Hive Hook 就会将元数据的变动通知 Atlas。除此之外，Atlas 还会根据 DML 语句获取数据之间的血缘关系。

3.2.1 全流程调度

为查看血缘关系效果，此处使用 Azkaban 将数仓的全流程调度一次。

1.新数据准备

1）用户行为日志

（1）启动日志采集通道，包括 Zookeeper，Kafka，Flume 等

（2）修改 hadoop102，hadoop103 两台节点的/opt/module/applog/application.yml 文件，将模拟日期改为 2020-06-17 如下

```
#业务日期
mock.date: "2020-06-17"
```

（3）执行生成日志的脚本

```
# lg.sh
```

（4）等待片刻，观察 HDFS 是否出现 2020-06-17 的日志文件

2）业务数据

（1）修改/opt/module/db_log/application.properties，将模拟日期修改为 2020-06-17，如下

```
#业务日期
mock.date=2020-06-17
```

（2）进入到/opt/module/db_log 路径，执行模拟生成业务数据的命令，如下

```
# java -jar gmall2020-mock-db-2021-01-22.jar
```

（3）观察 mysql 的 gmall 数据中是否出现 2020-06-17 的数据

2.启动 Azkaban

注意需使用 azkaban 用户启动 Azkaban

1）启动 Executor Server

在各节点执行以下命令，启动 Executor

```
[root@hadoop102 ~]# sudo -i -u azkaban bash -c "cd
/opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
[root@hadoop103 ~]# sudo -i -u azkaban bash -c "cd
/opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
[root@hadoop104 ~]# sudo -i -u azkaban bash -c "cd
/opt/module/azkaban/azkaban-exec;bin/start-exec.sh"
```

2）激活 Executor Server，任选一台节点执行以下激活命令即可

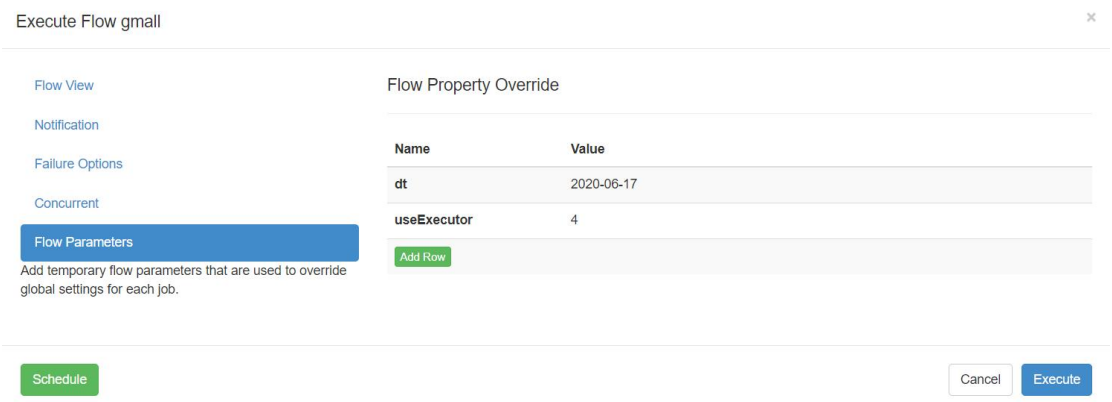
```
[root@hadoop102 ~]# curl
http://hadoop102:12321/executor?action=activate
[root@hadoop102 ~]# curl
http://hadoop103:12321/executor?action=activate
[root@hadoop102 ~]# curl
http://hadoop104:12321/executor?action=activate
```

3）启动 Web Server

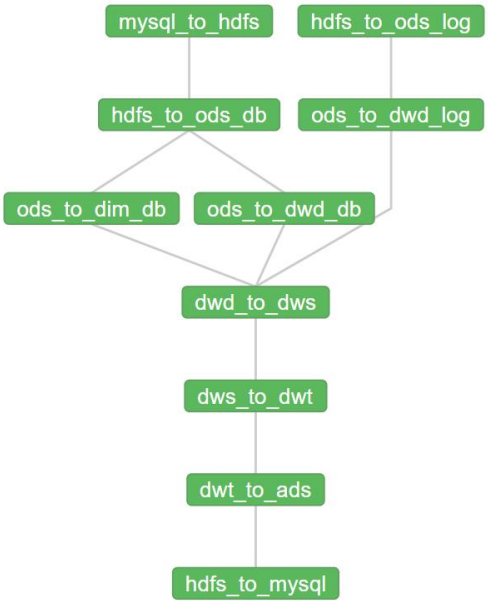
```
[root@hadoop102 ~]# sudo -i -u azkaban bash -c "cd
/opt/module/azkaban/azkaban-web;bin/start-web.sh"
```

3.全流程调度

1）工作流参数

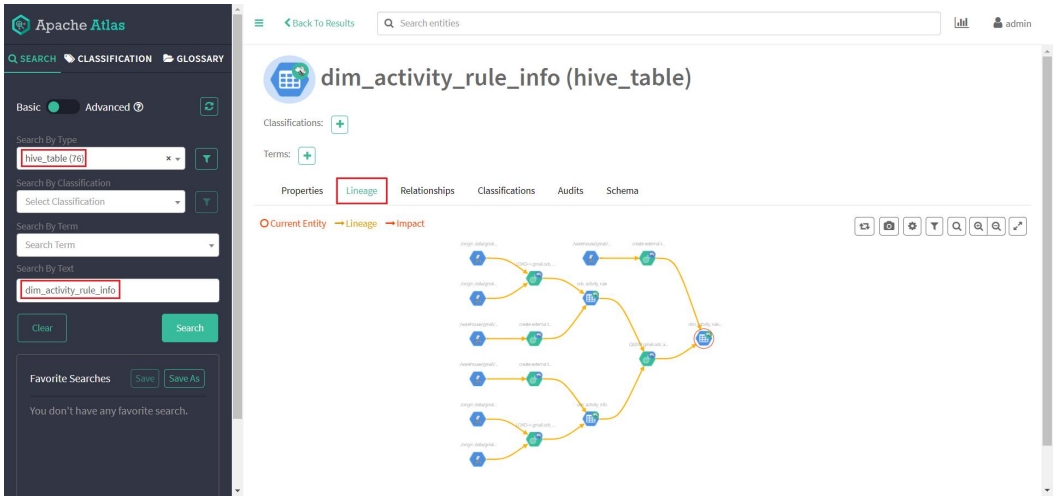


2）运行结果



3.2.2 查看血缘依赖

此时在通过 Atlas 查看 Hive 元数据，即可发现血缘依赖



第 4 章 扩展内容

4.1 Atlas 源码编译

4.1.1 安装 Maven

1）Maven 下载：https://maven.apache.org/download.cgi

2）把 apache-maven-3.6.1-bin.tar.gz 上传到 linux 的/opt/software 目录下

3）解压 apache-maven-3.6.1-bin.tar.gz 到/opt/module/目录下面

```
[root@hadoop102 software]# tar -zxvf apache-maven-3.6.1-bin.tar.gz -C /opt/module/
```

4）修改 apache-maven-3.6.1 的名称为 maven

```
[root@hadoop102 module]# mv apache-maven-3.6.1/ maven
```

5）添加环境变量到/etc/profile 中

```
[root@hadoop102 module]#vim /etc/profile
#MAVEN_HOME
export MAVEN_HOME=/opt/module/maven
export PATH=$PATH:$MAVEN_HOME/bin
```

6）测试安装结果

```
[root@hadoop102 module]# source /etc/profile
[root@hadoop102 module]# mvn -v
```

7）修改 setting.xml，指定为阿里云

```
[root@hadoop101 module]# cd /opt/module/maven/conf/
[root@hadoop102 maven]# vim settings.xml

<!-- 添加阿里云镜像-->
<mirror>
    <id>nexus-aliyun</id>
    <mirrorOf>central</mirrorOf>
    <name>Nexus aliyun</name>
<url>http://maven.aliyun.com/nexus/content/groups/public</url>
</mirror>
<mirror>
    <id>UK</id>
    <name>UK Central</name>
    <url>http://uk.maven.org/maven2</url>
    <mirrorOf>central</mirrorOf>
</mirror>
<mirror>
    <id>repo1</id>
    <mirrorOf>central</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://repo1.maven.org/maven2/</url>
</mirror>
<mirror>
```

```
    <id>repo2</id>
    <mirrorOf>central</mirrorOf>
    <name>Human Readable Name for this Mirror.</name>
    <url>http://repo2.maven.org/maven2/</url>
</mirror>
```

4.1.2 编译 Atlas 源码

1）把 apache-atlas-2.1.0-sources.tar.gz 上传到 hadoop102 的/opt/software 目录下

2）解压 apache-atlas-2.1.0-sources.tar.gz 到/opt/module/目录下面

```
[root@hadoop101 software]# tar -zxvf apache-atlas-2.1.0-sources.tar.gz -C /opt/module/
```

3）下载 Atlas 依赖

```
[root@hadoop101 software]# export MAVEN_OPTS="-Xms2g -Xmx2g"

[root@hadoop101 software]# cd /opt/module/apache-atlas-sources-2.1.0/
[root@hadoop101 apache-atlas-sources-2.1.0]# mvn clean -DskipTests install
[root@hadoop101 apache-atlas-sources-2.1.0]# mvn clean -DskipTests package -Pdis
#一定要在${atlas_home}执行
[root@hadoop101 apache-atlas-sources-2.1.0]# cd distro/target/
[root@hadoop101 target]# mv apache-atlas-2.1.0-server.tar.gz /opt/software/
[root@hadoop101 target]# mv apache-atlas-2.1.0-hive-hook.tar.gz /opt/software/
```

提示：执行过程比较长，会下载很多依赖，大约需要半个小时，期间如果报错很有可能是因为 TimeOut 造成的网络中断，重试即可。

4.2 Atlas 内存配置

如果计划存储数万个元数据对象，建议调整参数值获得最佳的 JVM GC 性能。以下是常见的服务器端选项

1）修改配置文件/opt/module/atlas/conf/atlas-env.sh

```
#设置 Atlas 内存
export ATLAS_SERVER_OPTS="-server -XX:SoftRefLRUPolicyMSPerMB=0 -XX:+CMSClassUnloadingEnabled -XX:+UseConcMarkSweepGC -XX:+CMSParallelRemarkEnabled -XX:+PrintTenuringDistribution -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=dumps/atlas_server.hprof -Xloggc:logs/gc-worker.log -verbose:gc -XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=10 -XX:GCLogFileSize=1m -XX:+PrintGCDetails -XX:+PrintHeapAtGC -XX:+PrintGCTimeStamps"

#建议 JDK1.7 使用以下配置
export ATLAS_SERVER_HEAP="-Xms15360m -Xmx15360m -XX:MaxNewSize=3072m -XX:PermSize=100M -XX:MaxPermSize=512m"
```

```
#建议 JDK1.8 使用以下配置
export         ATLAS_SERVER_HEAP="-Xms15360m         -Xmx15360m
-XX:MaxNewSize=5120m                        -XX:MetaspaceSize=100M
-XX:MaxMetaspaceSize=512m"

#如果是 Mac OS 用户需要配置
export              ATLAS_SERVER_OPTS="-Djava.awt.headless=true
-Djava.security.krb5.realm= -Djava.security.krb5.kdc="
```

参数说明：-XX:SoftRefLRUPolicyMSPerMB 此参数对管理具有许多并发用户的查询繁重工作负载的 GC 性能特别有用。

4.3 配置用户名密码

Atlas 支持以下身份验证方法：File、Kerberos 协议、LDAP 协议

通过修改配置文件 atlas-application.properties 文件开启或关闭三种验证方法

```
atlas.authentication.method.kerberos=true|false
atlas.authentication.method.ldap=true|false
atlas.authentication.method.file=true|false
```

如果两个或多个身份证验证方法设置为 true，如果较早的方法失败，则身份验证将回退到后一种方法。例如，如果 Kerberos 身份验证设置为 true 并且 ldap 身份验证也设置为 true，那么，如果对于没有 kerberos principal 和 keytab 的请求，LDAP 身份验证将作为后备方案。

本文主要讲解采用文件方式修改用户名和密码设置。其他方式可以参见官网配置即可。

1）打开/opt/module/atlas/conf/users-credentials.properties 文件

```
[atguigu@hadoop102 conf]$ vim users-credentials.properties

#username=group::sha256-password
admin=ADMIN::8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a
81f6f2ab448a918
rangertagsync=RANGER_TAG_SYNC::e3f67240f5117d1753c940dae9eea77
2d36ed5fe9bd9c94a300e40413f1afb9d
```

（1）admin 是用户名称

（2）8c6976e5b5410415bde908bd4dee15dfb167a9c873fc4bb8a81f6f2ab448a918 是采用 sha256 加密的密码，默认密码为 admin。

2）例如：修改用户名称为 atguigu，密码为 atguigu

（1）获取 sha256 加密的 atguigu 密码

```
[atguigu@hadoop102 conf]$ echo -n "atguigu"|sha256sum
2628be627712c3555d65e0e5f9101dbdd403626e6646b72fdf728a20c5261d
c2
```

（2）修改用户名和密码

```
[atguigu@hadoop102 conf]$ vim users-credentials.properties

#username=group::sha256-password
atguigu=ADMIN::2628be627712c3555d65e0e5f9101dbdd403626e6646b72
```

```
fdf728a20c5261dc2
rangertagsync=RANGER_TAG_SYNC::e3f67240f5117d1753c940dae9eea772d36ed5fe9bd9c94a300e40413f1afb9d
```