

# 尚硅谷大数据技术之 Atlas（元数据管理）

(作者：尚硅谷大数据研发部)

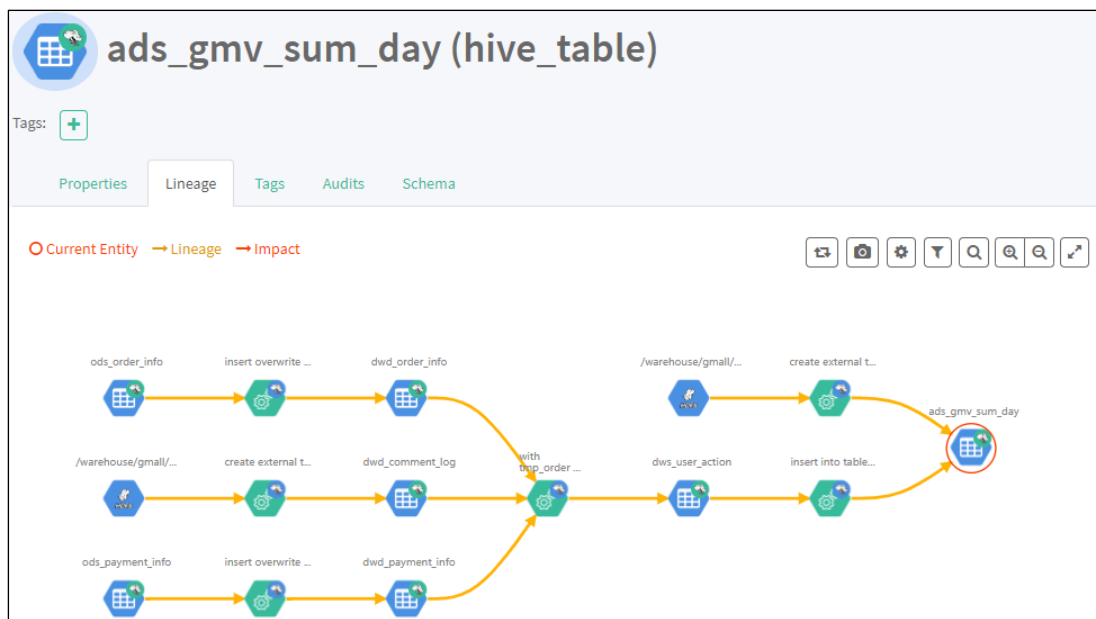
版本：V6.0.0

## 第 1 章 Atlas 入门

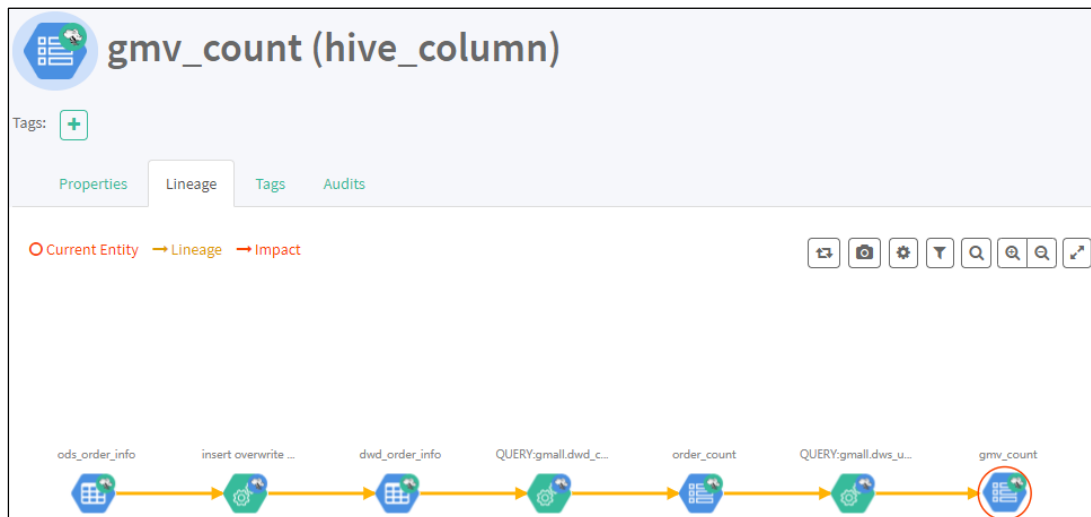
### 1.1 Atlas 概述

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。

#### 1) 表与表之间的血缘依赖

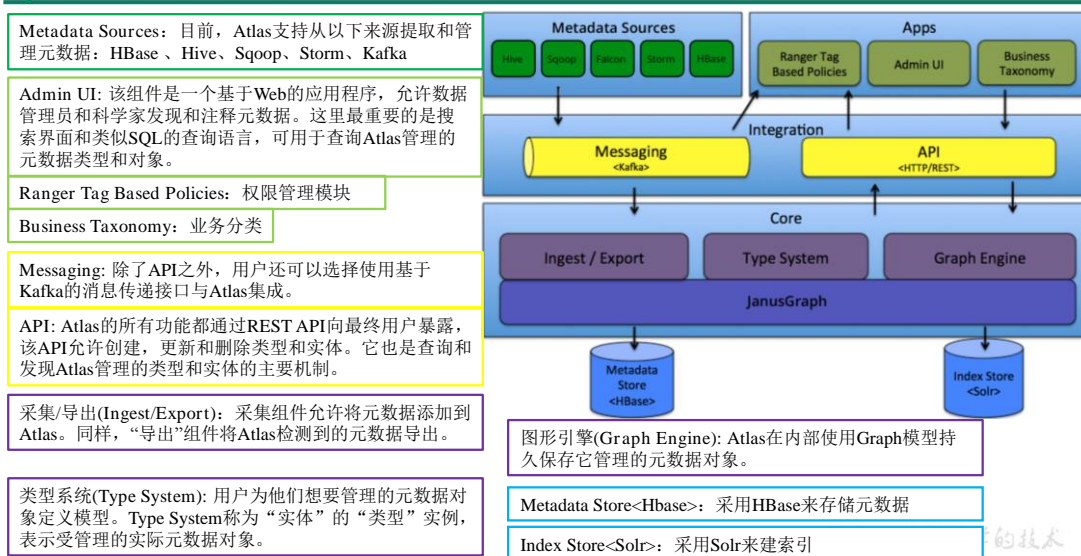


#### 2) 字段与字段之间的血缘依赖



## 1.2 Atlas 架构原理

### Atlas架构原理



## 第 2 章 Atlas 安装及使用

- 1) Atlas 官网地址: <https://atlas.apache.org/>
- 2) 文档查看地址: <https://atlas.apache.org/0.8.4/index.html>
- 3) 下载地址: <https://www.apache.org/dyn/closer.cgi/atlas/0.8.4/apache-atlas-0.8.4-sources.tar.gz>

### 2.1 安装前环境准备

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。

服务名称	子服务	服务器	服务器	服务器
		hadoop102	hadoop103	hadoop104

HDFS	NameNode	√		
	DataNode	√	√	√
	SecondaryNameNode			√
Yarn	NodeManager	√	√	√
	Resourcemanager		√	
Zookeeper	QuorumPeerMain	√	√	√
Kafka	Kafka	√	√	√
HBase	HMaster	√		
	HRegionServer	√	√	√
Solr	Jar	√	√	√
Hive	Hive	√		
MySQL	MySQL	√		
Azkaban	AzkabanWebServer	√		
	AzkabanExecutorServer	√		
Atlas	atlas	√		
服务数总计		13	7	7

### 2.1.1 安装 JDK8、Hadoop3.1.3

#### 1) 安装 Hadoop 集群



尚硅谷大数据技术  
之Hadoop (入门)

#### 2) 启动 Hadoop 集群

```
[atguigu@hadoop102 hadoop-3.1.3]$ sbin/start-dfs.sh  
[atguigu@hadoop103 hadoop-3.1.3]$ sbin/start-yarn.sh
```

### 2.1.2 安装 Zookeeper3.5.7

#### 1) 安装 Zookeeper 集群



尚硅谷大数据技术  
之Zookeeper (V3)

#### 2) 启动 Zookeeper 集群

```
[atguigu@hadoop102 zookeeper-3.7.7]$ zk.sh start
```

### 2.1.3 安装 Kafka2.4.1

#### 1) 安装 Kafka 集群



尚硅谷大数据技术  
之Kafka (V3.0) .c

## 2) 启动 Kafka 集群

```
[atguigu@hadoop102 kafka]$ kf.sh start
```

## 2.1.4 安装 Hbase2.0.5

## 1) 安装 Hbase 集群



尚硅谷大数据技术  
之HBase.doc

## 2) 启动 Hbase 集群

```
[atguigu@hadoop102 hbase]$ bin/start-hbase.sh
```

## 2.1.5 安装 Solr5.2.1

## 1) Solr 版本要求必须是 5.2.1，见官网

2) Solr 下载: <http://archive.apache.org/dist/lucene/solr/5.2.1/solr-5.2.1.tgz>

## 3) 把 solr-5.2.1.tgz 上传到 hadoop102 的/opt/software 目录下

## 4) 解压 solr-5.2.1.tgz 到/opt/module/目录下

```
[atguigu@hadoop102 software]$ tar -zxvf solr-5.2.1.tgz -C /opt/module/
```

## 5) 修改 solr-5.2.1 的名称为 solr

```
[atguigu@hadoop102 module]$ mv solr-5.2.1/ solr
```

## 6) 进入 solr/bin 目录，修改 solr.in.sh 文件

```
[atguigu@hadoop102 solr]$ vim bin/solr.in.sh
#添加下列指令
ZK_HOST="hadoop102:2181,hadoop103:2181,hadoop104:2181"
SOLR_HOST="hadoop102"
# Sets the port Solr binds to, default is 8983
#可修改端口号
SOLR_PORT=8983
```

## 7) 分发 Solr，进行 Cloud 模式部署

```
[atguigu@hadoop102 module]$ xsync solr
```

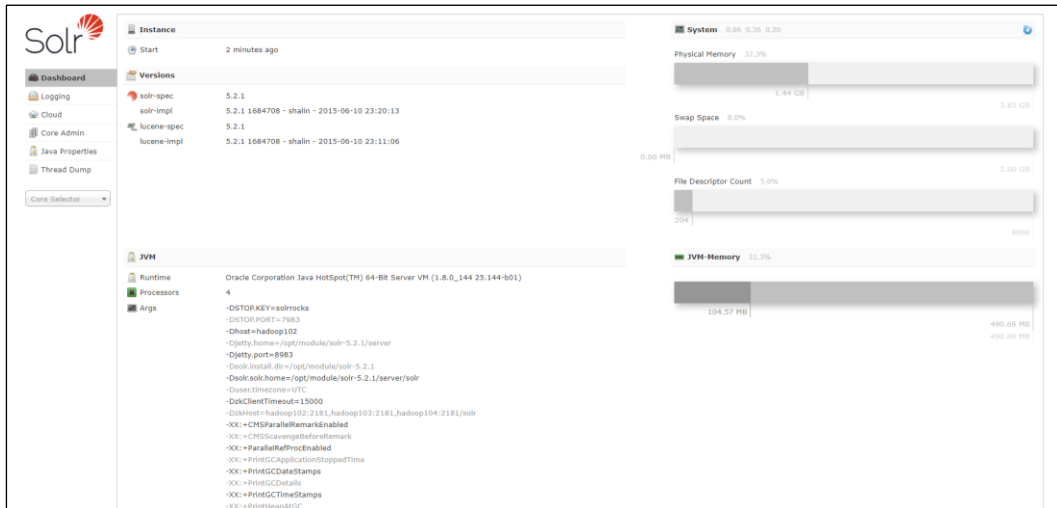
提示：分发完成后，分别对 hadoop103、hadoop104 主机/opt/module/solr/bin 下的 solr.in.sh 文件，修改为 SOLR\_HOST=对应主机名。

## 8) 在三台节点上分别启动 Solr，这个就是 Cloud 模式

```
[atguigu@hadoop102 solr]$ bin/solr start
[atguigu@hadoop103 solr]$ bin/solr start
[atguigu@hadoop104 solr]$ bin/solr start
```

提示：启动 Solr 前，需要提前启动 Zookeeper 服务。

9) Web 访问 8983 端口，可指定三台节点中的任意一台 IP，<http://hadoop102:8983/solr/#/>



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

## 10) 编写 Solr 启动停止脚本

(1) 在 hadoop102 的/home/atguigu/bin 目录下创建脚本

```
[atguigu@hadoop102 bin]$ vim s.sh
```

在脚本中编写如下内容

```
#!/bin/bash

case $1 in
"start"){
    for i in hadoop102 hadoop103 hadoop104
    do
        ssh $i "/opt/module/solr/bin/solr start"
    done
};;
"stop"){
    for i in hadoop102 hadoop103 hadoop104
    do
        ssh $i "/opt/module/solr/bin/solr stop"
    done
};;
esac
```

(2) 增加脚本执行权限

```
[atguigu@hadoop102 bin]$ chmod 777 s.sh
```

(3) Solr 集群启动脚本

```
[atguigu@hadoop102 module]$ s.sh start
```

(4) Solr 集群停止脚本

```
[atguigu@hadoop102 module]$ s.sh stop
```

## 2.1.6 安装 Hive3.1.2



尚硅谷大数据技术  
之Hive.docx

## 2.1.7 安装 Azkaban3.8.4



尚硅谷大数据技术  
之Azkaban.docx

## 2.1.8 安装 Atlas2.0

(1) 把 apache-atlas-2.0.0-server.tar.gz 上传到 hadoop102 的/opt/software 目录下

(2) 解压 apache-atlas-2.0.0-server.tar.gz 到/opt/module/目录下

```
[atguigu@hadoop102 software]# tar -zxvf apache-atlas-2.0.0-server.tar.gz  
-C /opt/module/
```

(3) 修改 apache-atlas-2.0.0 的名称为 atlas

```
[atguigu@hadoop102 module]# mv apache-atlas-2.0.0/ atlas
```

## 2.2 Atlas 集成外部框架

### 2.2.1 Atlas 集成 Hbase

1) 进入/opt/module/atlas/conf/目录，修改配置文件

```
[atguigu@hadoop102 conf]$ vim atlas-application.properties  
  
#修改 atlas 存储数据主机  
atlas.graph.storage.hostname=hadoop102:2181,hadoop103:2181,had  
oop104:2181
```

2) 进入到/opt/module/atlas/conf/hbase 路径，添加 Hbase 集群的配置文件到\${Atlas\_Home}

```
[atguigu@hadoop102 hbase]$  
ln -s /opt/module/hbase/conf/ /opt/module/atlas/conf/hbase/
```

3) 在/opt/module/atlas/conf/atlas-env.sh 中添加 HBASE\_CONF\_DIR

```
[atguigu@hadoop102 conf]$ vim atlas-env.sh  
  
#添加 HBase 配置文件路径  
export HBASE_CONF_DIR=/opt/module/atlas/conf/hbase/conf
```

### 2.2.2 Atlas 集成 Solr

1) 进入/opt/module/atlas/conf 目录，修改配置文件

```
[atguigu@hadoop102 conf]$ vim atlas-application.properties  
  
#修改如下配置  
atlas.graph.index.search.solr.zookeeper-  
url=hadoop102:2181,hadoop103:2181,hadoop104:2181
```

- 2) 将 Atlas 自带的 Solr 文件夹拷贝到外部 Solr 集群的各个节点。

```
[atguigu@hadoop102 conf]$  
cp -r /opt/module/atlas/conf/solr /opt/module/solr/
```

- 3) 进入到/opt/module/solr 路径，修改拷贝过来的配置文件名称为 atlas\_conf

```
[atguigu@hadoop102 solr]$ mv solr atlas_conf
```

- 4) 在 Cloud 模式下，启动 Solr（需要提前启动 Zookeeper 集群），并创建 collection

```
[atguigu@hadoop102 solr]$ bin/solr create -c vertex_index -d  
/opt/module/solr/atlas_conf -shards 3 -replicationFactor 2
```

```
[atguigu@hadoop102 solr]$ bin/solr create -c edge_index -d  
/opt/module/solr/atlas_conf -shards 3 -replicationFactor 2
```

```
[atguigu@hadoop102 solr]$ bin/solr create -c fulltext_index -d  
/opt/module/solr/atlas_conf -shards 3 -replicationFactor 2
```

-shards 3: 表示该集合分片数为 3

-replicationFactor 2: 表示每个分片数都有 2 个备份

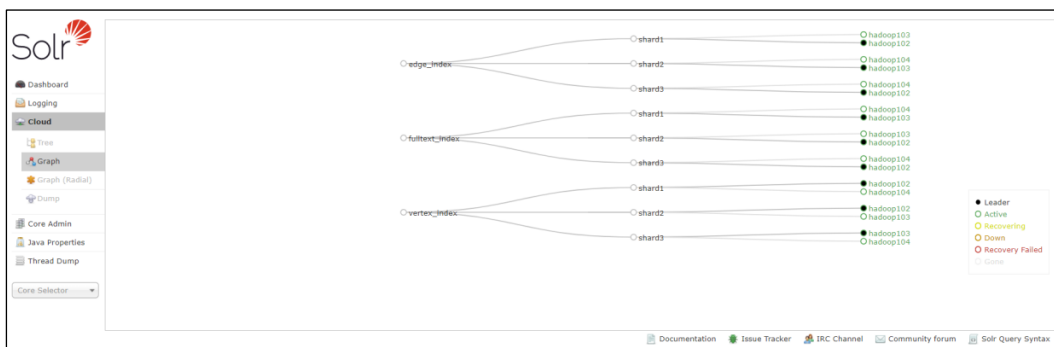
vertex\_index、edge\_index、fulltext\_index: 表示集合名称

注意：如果需要删除 vertex\_index、edge\_index、fulltext\_index 等 collection 可以执行如下命令。

```
[atguigu@hadoop102 solr]$ bin/solr delete -c ${collection_name}
```

- 5) 验证创建 collection 成功

登录 solr web 控制台: <http://hadoop102:8983/solr/#/~cloud> 看到如下图显示:



## 2.2.3 Atlas 集成 Kafka

- 1) 进入/opt/module/atlas/conf/目录，修改配置文件 atlas-application.properties

```
[atguigu@hadoop102 conf]$ vim atlas-application.properties  
  
##### Notification Configs #####  
atlas.notification.embedded=false  
atlas.kafka.data=/opt/module/kafka/logs  
atlas.kafka.zookeeper.connect=hadoop102:2181,hadoop103:2181,ha  
doo104:2181/kafka  
atlas.kafka.bootstrap.servers=hadoop102:9092,hadoop103:9092,ha  
doo104:9092
```

```
atlas.kafka.zookeeper.session.timeout.ms=4000
atlas.kafka.zookeeper.connection.timeout.ms=2000

atlas.kafka.enable.auto.commit=true
```

## 2) 启动 Kafka 集群，并创建 Topic

```
[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --zookeeper
hadoop102:2181,hadoop103:2181,hadoop104:2181/kafka --create --
replication-factor 3 --partitions 3 --topic _HOATLASOK

[atguigu@hadoop102 kafka]$ bin/kafka-topics.sh --zookeeper
hadoop102:2181,hadoop103:2181,hadoop104:2181/kafka --create --
replication-factor 3 --partitions 3 --topic ATLAS_ENTITIES
```

## 2.2.4 Atlas 其他配置

### 1) 进入/opt/module/atlas/conf/目录，修改配置文件 atlas-application.properties

```
[atguigu@hadoop102 conf]$ vim atlas-application.properties

##### 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,hadoop103:21
81,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

```
[atguigu@hadoop102 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 Atlas 集成 Hive

### 1) 进入/opt/module/atlas/conf/目录，修改配置文件 atlas-application.properties

```
[atguigu@hadoop102 conf]$ vim atlas-application.properties

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



```
atlas.cluster.name=primary
```

- 2) 解压 apache-atlas-2.0.0-hive-hook.tar.gz 到/opt/module/

```
[atguigu@hadoop102 software]$ tar -zxvf apache-atlas-2.0.0-hive-hook.tar.gz -C /opt/module/
```

- 3) 剪切 hook 和 hook-bin 目录到到/opt/module/atlas 文件夹中

```
[atguigu@hadoop102 software]$ mv /opt/module/apache-atlas-hive-hook-2.0.0/hook-bin/ /opt/module/atlas/
```

```
[atguigu@hadoop102 software]$ mv /opt/module/apache-atlas-hive-hook-2.0.0/hook /opt/module/atlas/
```

- 4) 将 atlas-application.properties 配置文件加入到 atlas-plugin-classloader-1.0.0.jar 中

```
[atguigu@hadoop102 hive]$ zip -u /opt/module/atlas/hook/hive/atlas-plugin-classloader-2.0.0.jar /opt/module/atlas/conf/atlas-application.properties
```

```
[atguigu@hadoop102 hive]$ cp /opt/module/atlas/conf/atlas-application.properties /opt/module/hive/conf/
```

原因：这个配置不能参照官网，将配置文件考到 hive 的 conf 中。参考官网的做法一直读取不到 atlas-application.properties 配置文件，看了源码发现是在 classpath 读取的这个配置文件，所以将它压到 jar 里面。

- 5) 在/opt/module/hive/conf/hive-site.xml 文件中设置 Atlas hook

```
[atguigu@hadoop102 conf]$ vim hive-site.xml
```

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

```
[atguigu@hadoop102 conf]$ vim hive-env.sh
```

```
#在 tez 引擎依赖的 jar 包后面追加 hive 插件相关 jar 包
export HIVE_AUX_JARS_PATH=/opt/module/atlas/hook/hive
```

## 2.3 集群启动

- 1) 启动 Hadoop 集群

```
[atguigu@hadoop102 hadoop-3.1.3]$ sbin/start-dfs.sh
[atguigu@hadoop103 hadoop-3.1.3]$ sbin/start-yarn.sh
```

- 2) 启动 Zookeeper 集群

```
[atguigu@hadoop102 zookeeper-3.5.7]$ zk.sh start
```

- 3) 启动 Kafka 集群

```
[atguigu@hadoop102 kafka]$ kf.sh start
```

- 4) 启动 Hbase 集群:

```
[atguigu@hadoop102 hbase]$ bin/start-hbase.sh
```

- 5) 启动 Solr 集群

```
[atguigu@hadoop102 solr]$ bin/solr start
[atguigu@hadoop103 solr]$ bin/solr start
```

```
[atguigu@hadoop104 solr]$ bin/solr start
```

- 6) 进入/opt/module/atlas 路径，重新启动 Atlas 服务

```
[atguigu@hadoop102 atlas]$ bin/atlas_stop.py
```

```
[atguigu@hadoop102 atlas]$ bin/atlas_start.py
```

提示：错误信息查看路径：/opt/module/atlas/logs/\*.out 和 application.log

访问地址：<http://hadoop102:21000>

注意：等待时间大概 2 分钟。

账户：admin

密码：admin

## 2.4 将 Hive 元数据导入 Atlas

- 1) 配置 Hive 环境变量

```
[atguigu@hadoop102 hive]$ sudo vim /etc/profile.d/my_env.sh
```

```
#配置 Hive 环境变量
```

```
export HIVE_HOME=/opt/module/hive
```

```
export PATH=$PATH:$HIVE_HOME/bin/
```

```
[atguigu@hadoop102 hive]$ source /etc/profile.d/my_env.sh
```

- 2) 启动 Hive，如果 Hive 能正常启动说明环境 OK，就可以退出 Hive 客户端

```
[atguigu@hadoop102 hive]$ hive
hive (default)> show databases;
hive (default)> use gmall;
```

- 3) 在/opt/module/atlas/hook-bin 路径，将 Hive 元数据导入到 Atlas

```
[atguigu@hadoop102 hook-bin]$ ./import-hive.sh
```

```
Using Hive configuration directory [/opt/module/hive/conf]
Log file for import is /opt/module/atlas/logs/import-hive.log
log4j:WARN No such property [maxFileSize] in
org.apache.log4j.PatternLayout.
log4j:WARN No such property [maxBackupIndex] in
org.apache.log4j.PatternLayout.
```

输入用户名：admin；输入密码：admin

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

```
Enter password for atlas :-
```

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

## 2.5 Atlas 常用配置（可选）

### 2.5.1 配置内存

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。

## 2.5.2 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
2628be627712c3555d65e0e5f9101dbdd403626e6646b72fdf728a20c5261dc2
```

(2) 修改用户名和密码

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

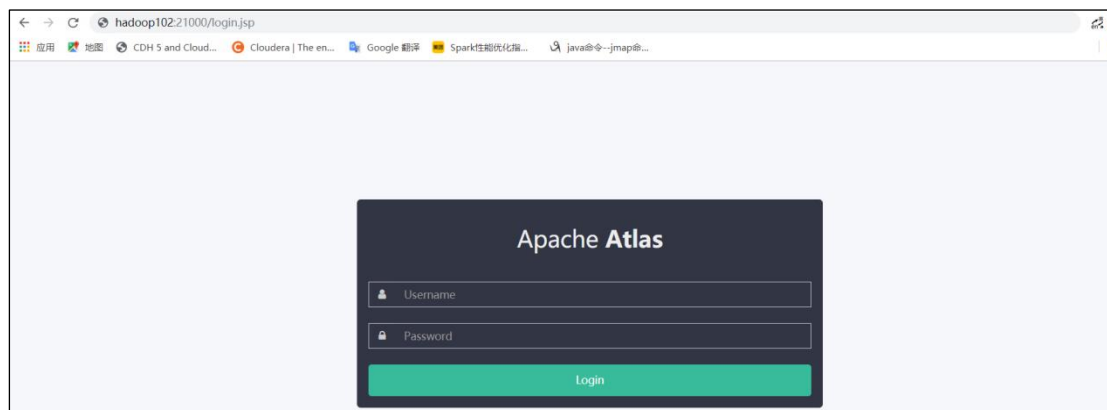
#username=group::sha256-password
atguigu=ADMIN::2628be627712c3555d65e0e5f9101dbdd403626e6646b72
fdf728a20c5261dc2
rangertagsync=RANGER_TAG_SYNC::e3f67240f5117d1753c940dae9eea77
2d36ed5fe9bd9c94a300e40413f1afb9d
```

## 第 3 章 界面查看

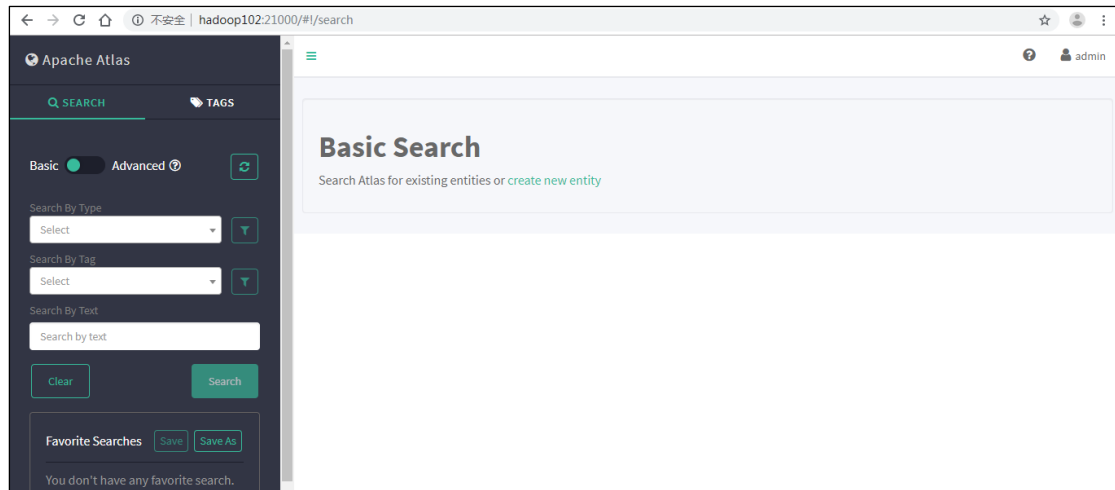
### 3.1 基本信息查看

#### 3.1.1 登录

1) 访问 `http://hadoop102:21000` 端口号 进行登录

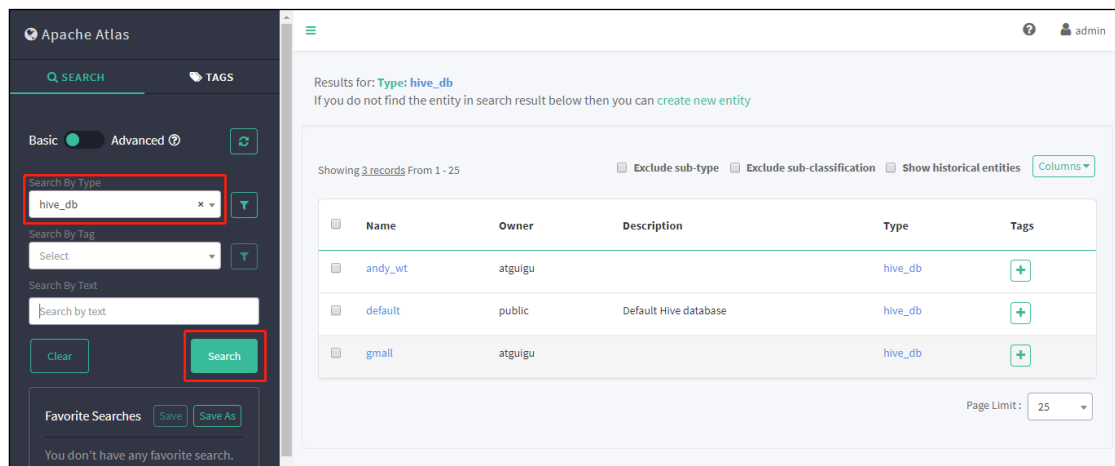


2) 账号密码默认都为 admin，登录成功页面如下



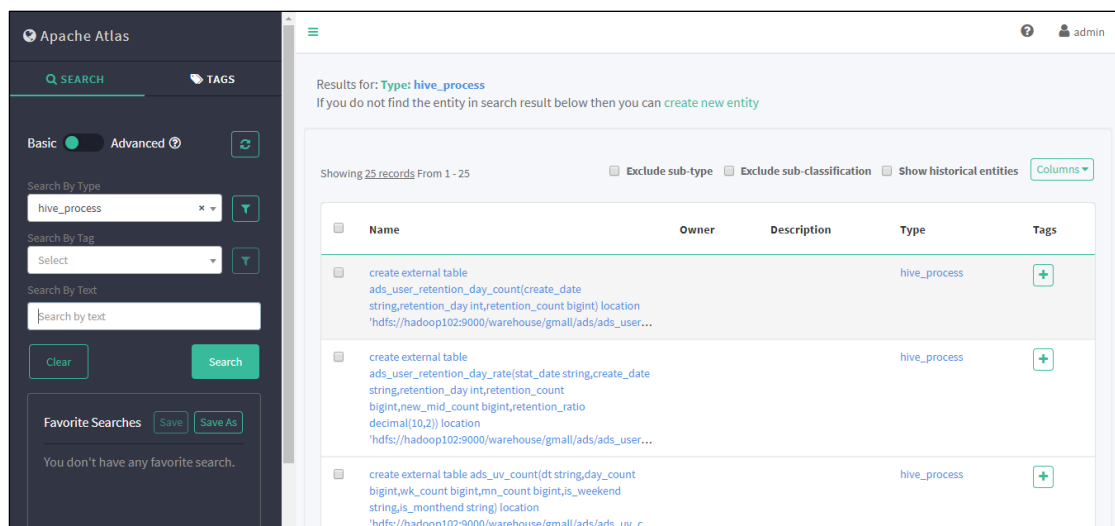
### 3.1.2 查询 Hive 库

1) 查询相应的 Hive 库，类型选择 hive\_db



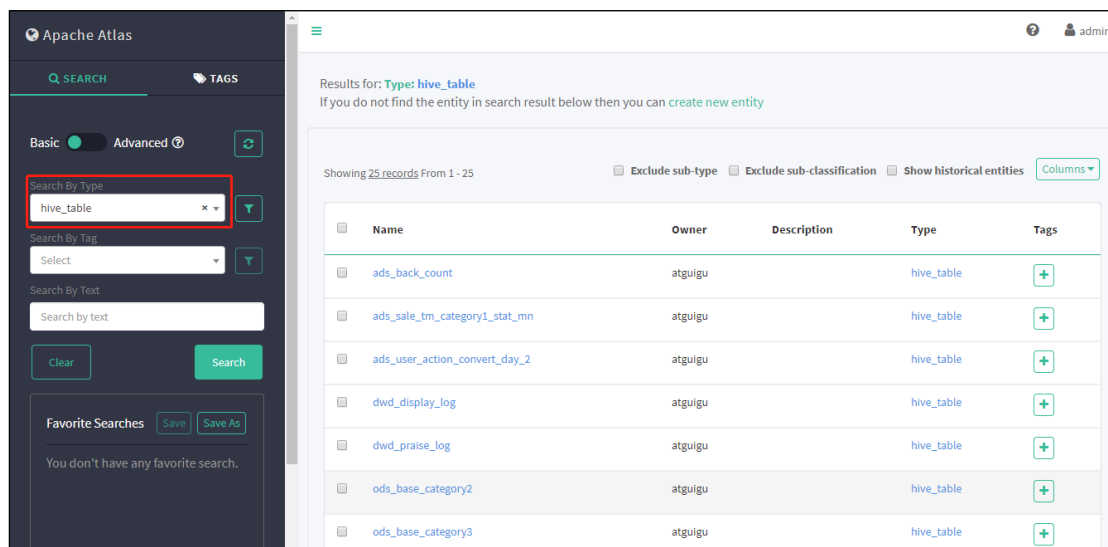
### 3.1.3 查询 Hive 进程

查询 Hive 进程，类型选择 hive\_process



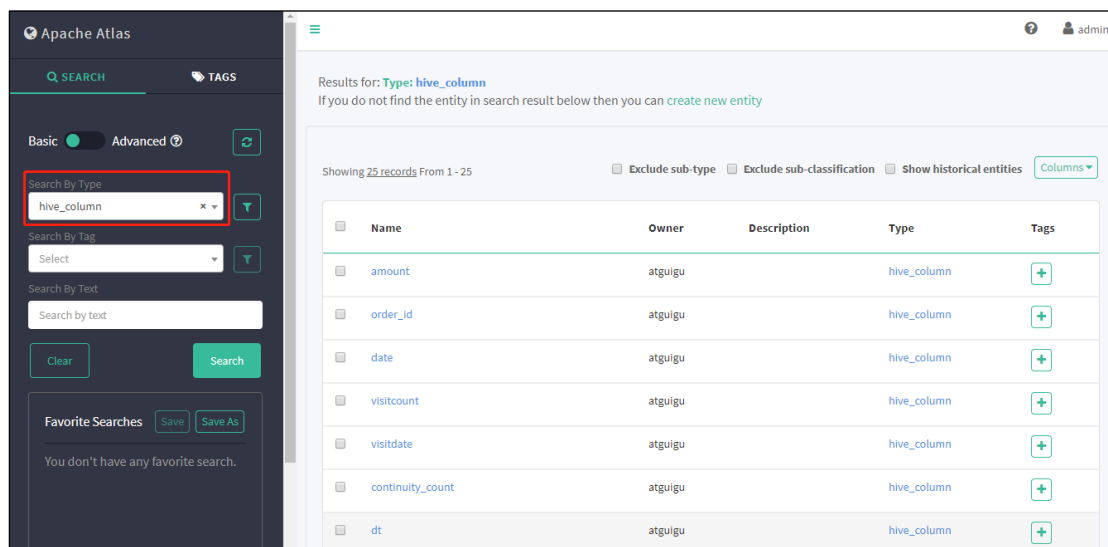
### 3.1.4 查询 Hive 表

查询 Hive 表，类型选择 hive\_table



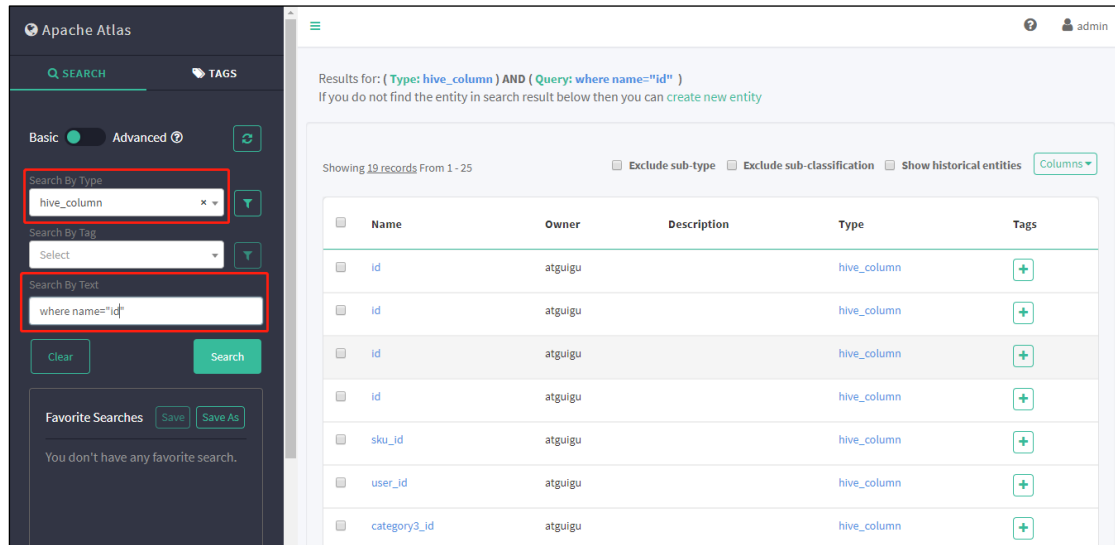
### 3.1.5 查询 Hive 列

查询 Hive 表，类型选择 hive\_column



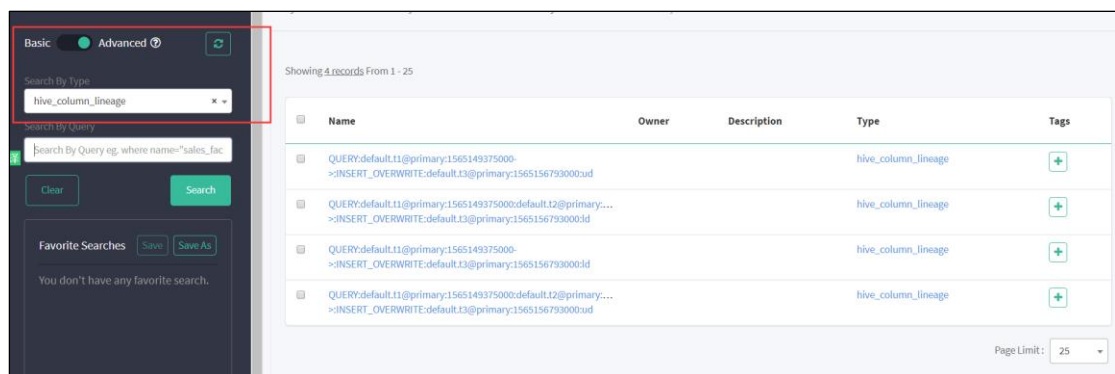
### 3.1.6 筛选查询条件

比如要查询 name 为 sequence 的列，那么在 Search By Query 填写 where name='sequence' 其他选项筛选条件写法一样。



### 3.1.7 查看具有血缘依赖列的数据

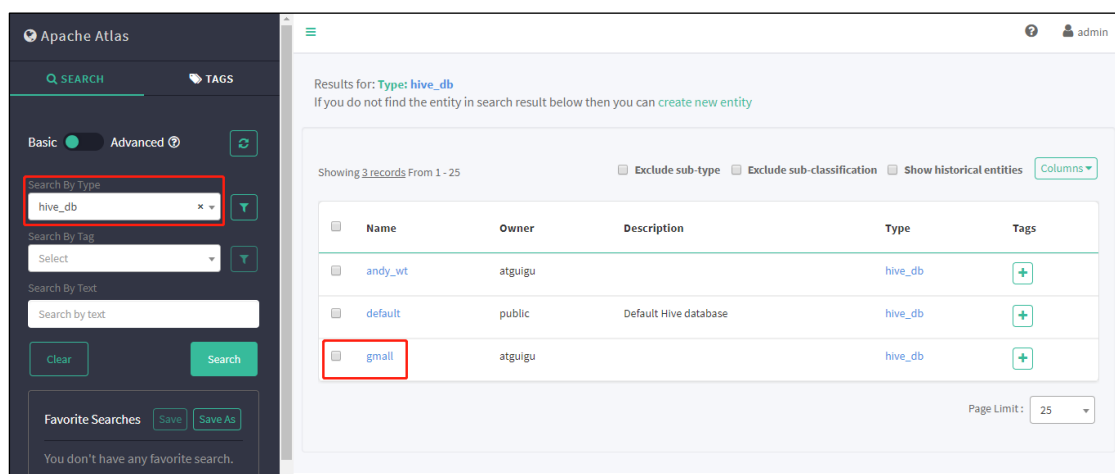
查询类型选择 hive\_column\_lineage



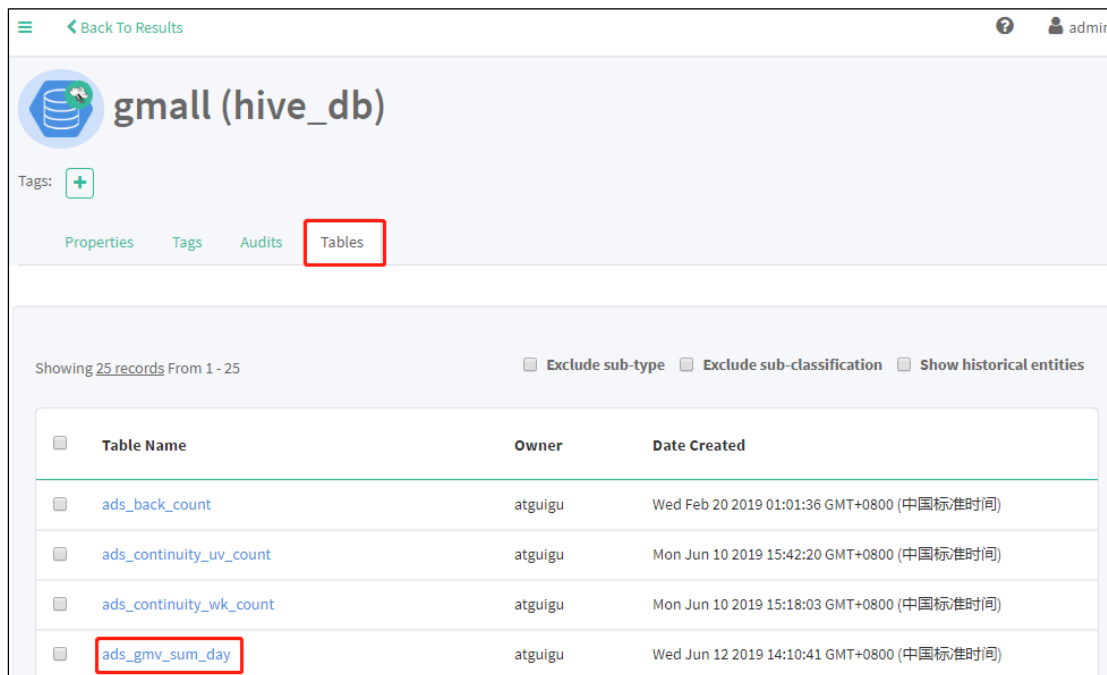
## 3.2 查看血缘依赖关系

### 3.2.1 第一次查看表血缘依赖

1) 先选中 hive\_db，点击 hive\_db 查看对应的表，如下图选中 gmail 库查看相应的表。



2) 点击 Tables 按钮，可以看到 gmall 数据库中的所有表。



Back To Results

admin

### gmall (hive\_db)

Tags: +

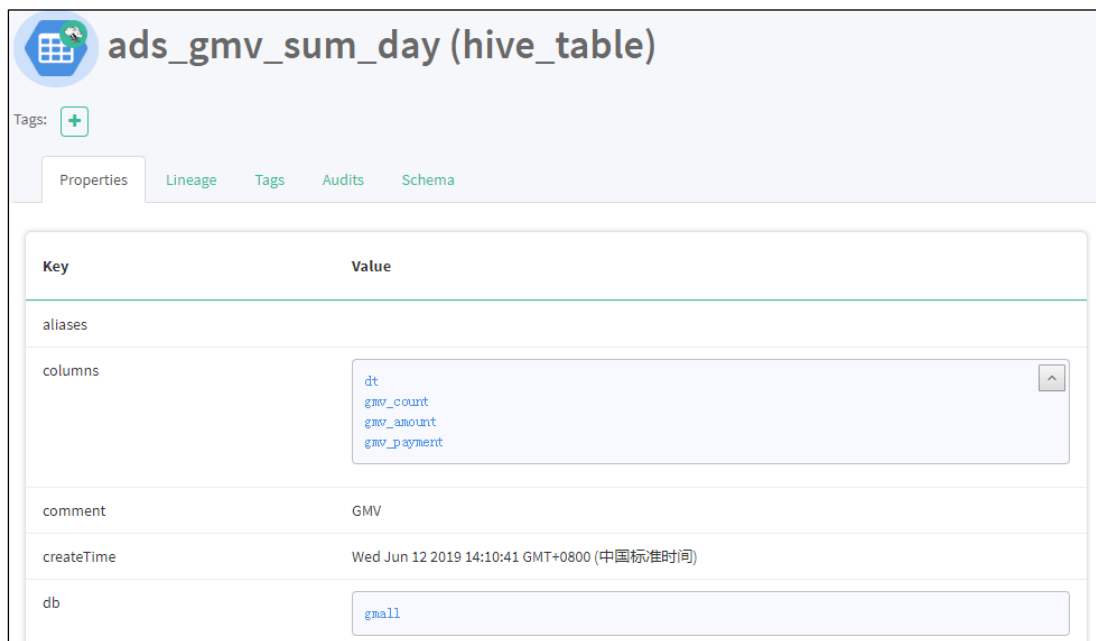
Properties Tags Audits **Tables**

Showing 25 records From 1 - 25

☐ Exclude sub-type ☐ Exclude sub-classification ☐ Show historical entities

<input type="checkbox"/>	Table Name	Owner	Date Created
<input type="checkbox"/>	ads_back_count	atguigu	Wed Feb 20 2019 01:01:36 GMT+0800 (中国标准时间)
<input type="checkbox"/>	ads_continuity_uv_count	atguigu	Mon Jun 10 2019 15:42:20 GMT+0800 (中国标准时间)
<input type="checkbox"/>	ads_continuity_wk_count	atguigu	Mon Jun 10 2019 15:18:03 GMT+0800 (中国标准时间)
<input type="checkbox"/>	ads_gmv_sum_day	atguigu	Wed Jun 12 2019 14:10:41 GMT+0800 (中国标准时间)

3) 点击 Properties，查看到表详情



### ads\_gmv\_sum\_day (hive\_table)

Tags: +

Properties Lineage Tags Audits Schema

Key	Value
aliases	
columns	dt gmw_count gmw_amount gmw_payment
comment	GMV
createTime	Wed Jun 12 2019 14:10:41 GMT+0800 (中国标准时间)
db	gmall

4) 点击 Lineage，显示血缘依赖关系





5) 点击 Audits，显示表修改过几次对应的时间和详情



ads\_gmv\_sum\_day (hive\_table)

Tags: +

Properties Lineage Tags Audits Schema

Showing 1 records From 1 - 25

Users	Timestamp	Actions	Tools
admin	Wed Aug 28 2019 14:58:46 GMT+0800 (中国标准时间)	Entity Created	<a href="#">Detail</a>

< 1 >

6) 点击 Schema，可以查看到表字段信息

Back To Results

admin

ads\_gmv\_sum\_day (hive\_table)

Tags: +

Properties Lineage Tags Audits Schema

Showing 1 - 4 

Show historical entities

<input type="checkbox"/>	Name	Owner	Type	Comment	Tags
<input type="checkbox"/>	dt	atguigu	string	????	<div>+</div>
<input type="checkbox"/>	gmv_count	atguigu	bigint	??gmv????	<div>+</div>
<input type="checkbox"/>	gmv_amount	atguigu	decimal(16,2)	??gmv?????	<div>+</div>
<input type="checkbox"/>	gmv_payment	atguigu	decimal(16,2)	??????	<div>+</div>

### 3.2.2 第一次查看字段血缘依赖

1) 点击 gmv\_count 字段

ads\_gmv\_sum\_day (hive\_table)

Tags: +

Properties Lineage Tags Audits Schema

Showing 1 - 4 

Show historical entities

<input type="checkbox"/>	Name	Owner	Type	Comment	Tags
<input type="checkbox"/>	dt	atguigu	string	????	<div>+</div>
<input type="checkbox"/>	gmv_count	atguigu	bigint	??gmv????	<div>+</div>
<input type="checkbox"/>	gmv_amount	atguigu	decimal(16,2)	??gmv?????	<div>+</div>
<input type="checkbox"/>	gmv_payment	atguigu	decimal(16,2)	??????	<div>+</div>

Page Limit: 25 < 1 >

2) 点击 Properties，显示字段详情

 **gmv\_count (hive\_column)**

Tags: 

Properties


Lineage


Tags

Audits

Key	Value
comment	??gmv???
description	
name	gmv_count
owner	atguigu
position	1
qualifiedName	gmall.ads_gmv_sum_day.gmv_count@primary
replicatedFrom	

3) 点击 Lineage，显示血缘关系

 **gmv\_count (hive\_column)**

Tags: 









Properties

Lineage

Tags

Audits

○ Current Entity → Lineage → Impact



No lineage data found

4) 点击 Audits，显示修改时间和详情

 **gmv\_count (hive\_column)**

Tags: 

Properties

Lineage

Tags

Audits

Showing 1 records From 1 - 25

Users	Timestamp	Actions	Tools
admin	Wed Aug 28 2019 14:58:46 GMT+0800 (中国标准时间)	Entity Created	<div>Detail</div>

<

1

>

### 3.2.3 启动 GMV 全流程任务

#### 1) 启动 Azkaban

(1) 启动 executor 服务器。在 executor 服务器目录下执行启动命令

```
[atguigu@hadoop102 executor]$ pwd
/opt/module/azkaban/executor
[atguigu@hadoop102 executor]$ bin/azkaban-executor-start.sh
```

(2) 启动 web 服务器。在 azkaban web 服务器目录下执行启动命令

```
[atguigu@hadoop102 server]$ pwd
/opt/module/azkaban/server
[atguigu@hadoop102 server]$ bin/azkaban-web-start.sh
```

(3) 查看 web 页面: <https://hadoop102:8443/>

#### 2) 上传任务

参见数仓项目中，Azkaban 任务提交流程。

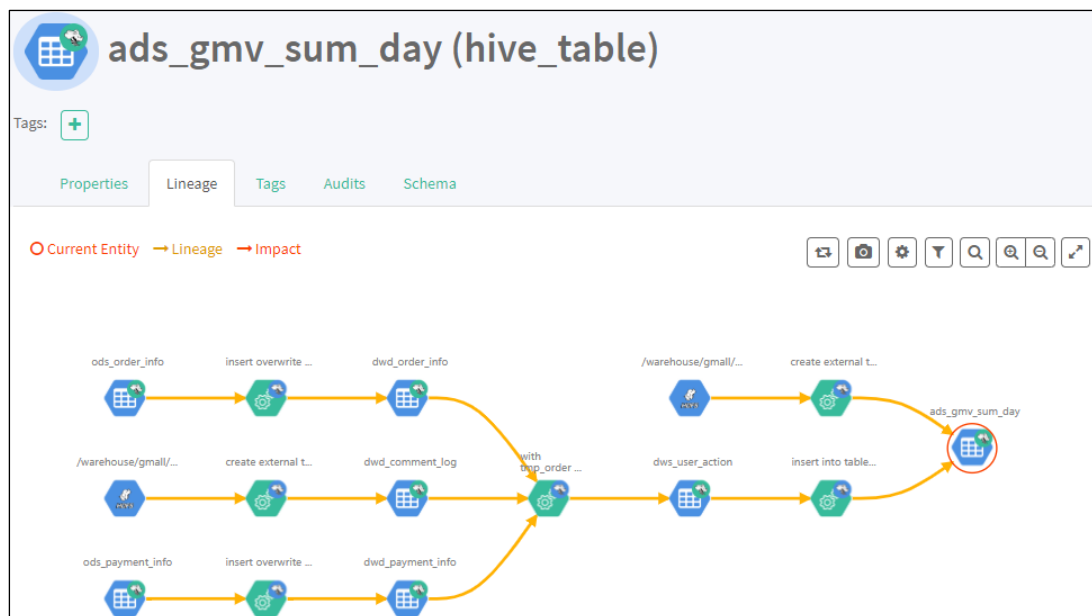


GMV指标获取的全  
调度流程.docx

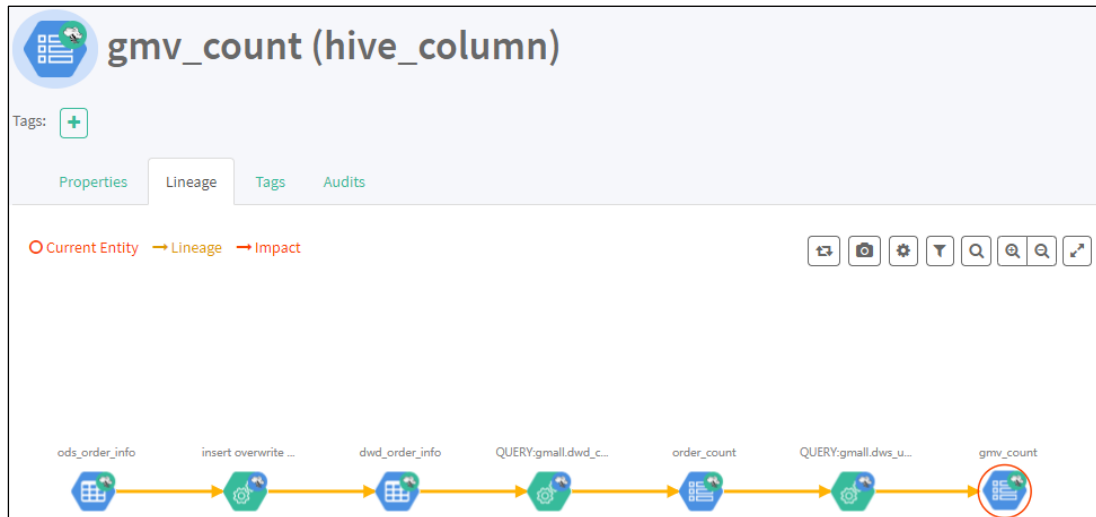
#### 3) 查看结果

等待 Azkaban 运行结束。

#### 4) 查看 Atlas 表血缘依赖关系图



#### 5) 查看 Atlas 字段血缘依赖关系图



### 3.3 Rest API 使用（二次开发使用）

如果需要对 Atlas 进行二次开发，可以查看官网地址，<https://atlas.apache.org/api/v2/index.html>，根据官方 API 进行二次开发。

## 第 4 章 编译 Atlas 源码

### 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

```
[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>
```

- 8) 在/home/atguigu 目录下创建.m2 文件夹

```
[root@hadoop102 ~]$ mkdir .m2
```

- 9) 上传 repository 文件夹所有内容到/home/atguigu/.m2 文件夹里面

### 4.1.2 编译 Atlas 源码

- 1) 把 apache-atlas-2.0.0-sources.tar.gz 上传到 hadoop102 的/opt/software 目录下

- 2) 解压 apache-atlas-2.0.0-sources.tar.gz 到/opt/module/目录下面

```
[root@hadoop101 software]# tar -zxvf apache-atlas-2.0.0-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-2.0.0/
[root@hadoop101 apache-atlas-sources-2.0.0]# mvn clean -DskipTests install
[root@hadoop101 apache-atlas-sources-2.0.0]# mvn clean -DskipTests package -Pdis

#一定要在${atlas_home}执行
[root@hadoop101 apache-atlas-sources-2.0.0]# cd distro/target/
[root@hadoop101 target]# mv apache-atlas-2.0.0-server.tar.gz /opt/software/
[root@hadoop101 target]# mv apache-atlas-2.0.0-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。