shiyan

3.19实验课-使用Flink实时发现最热Github项目

准备工作：

1. 创建Flink工作空间

https://home.console.aliyun.com/home/dashboard/ProductAndService

点击实时计算Flink版，进入其控制台

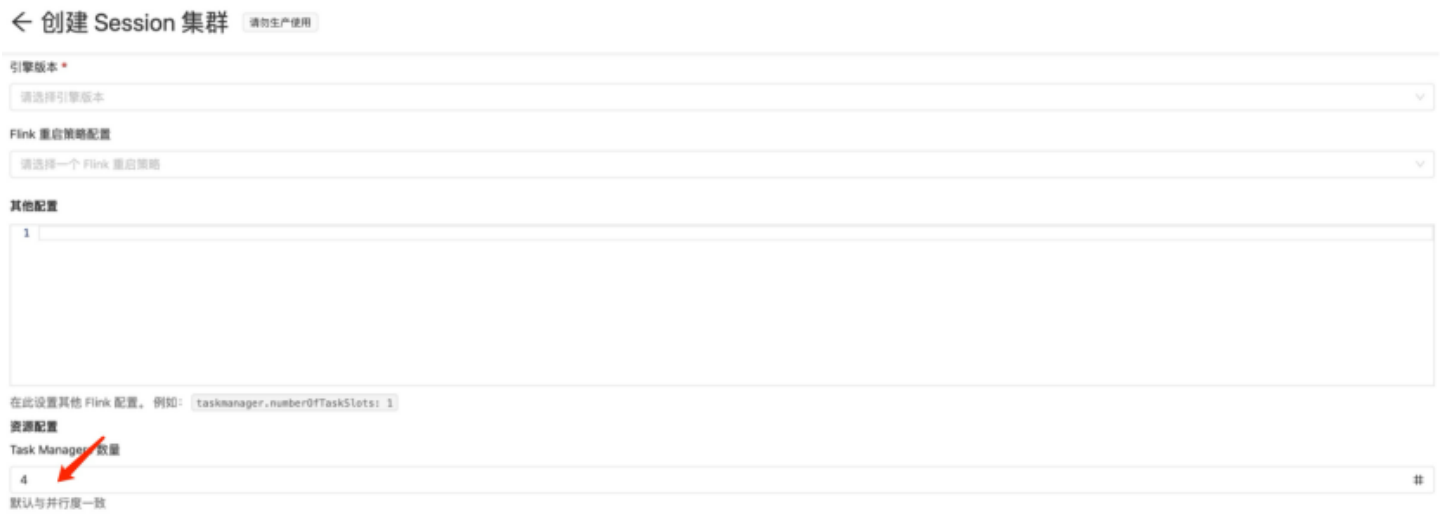
上节课已经领取了资源折扣，点击立即购买

虚拟交换机可以选择ECS以及RDS上对应的区域

创建完成之后，进入所创建的工作空间的控制台

2. Flink Session集群

根据上节课所讲，创建一个Session，可将Task Managers 数量设为4，增大并发度。



实验一：Github 关注数排行榜

创建一个SQL作业

注意：其中startTime尽量设置为当前此刻的一周前附近，如果设置的时间太早，前面无效计算时间比较长，不仅耗费资源，而且很久才能加载出计算结果。根据不同的地域设置相应的project和endPoint，在本实验中，实例为上海的服务平台，因此设置'project' = 'github-events-shanghai'和'endPoint' = 'https://cn-shanghai-intranet.log.aliyuncs.com'，其他地域如北京、杭州、深圳更改为对应值即可。

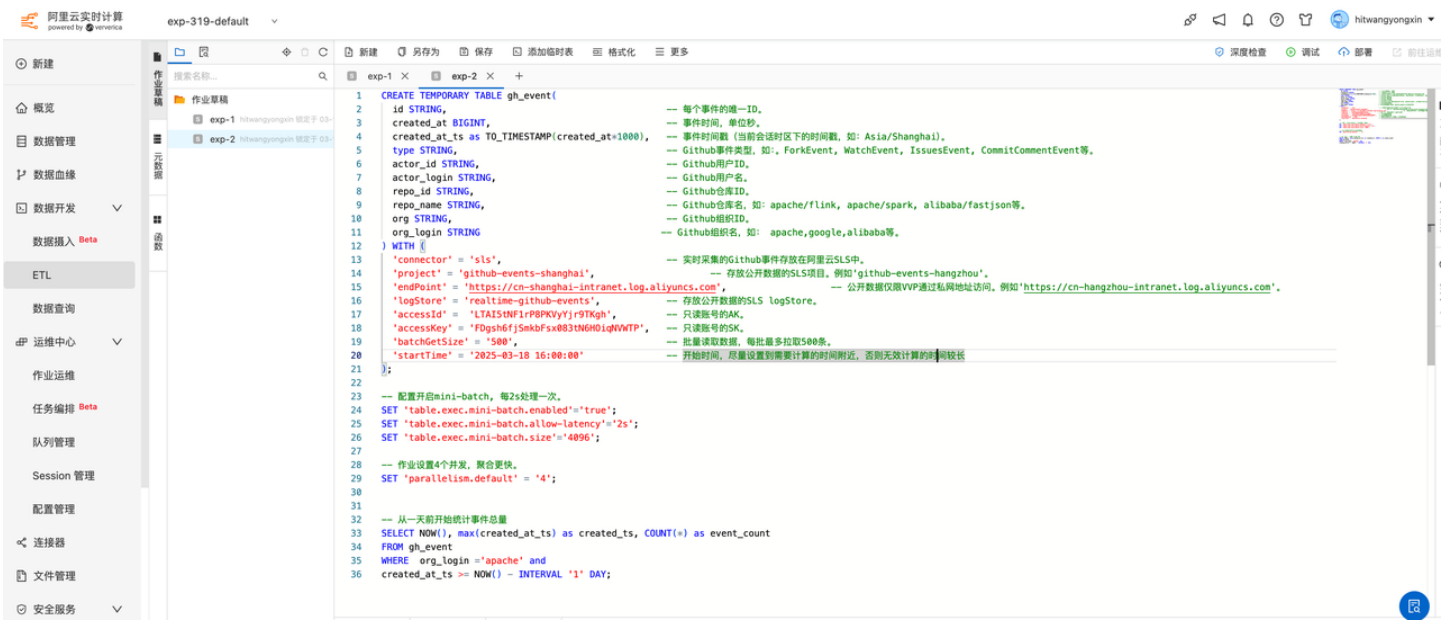
```
1   -- 通过DDL语句创建SLS源表，SLS中存放了Github的实时数据。
2   CREATE TEMPORARY TABLE gh_event(
3     id STRING,                                          -- 每个事件的唯一ID。
4     created_at BIGINT,                                  -- 事件时间，单位秒。
5     created_at_ts as TO_TIMESTAMP(created_at*1000),     -- 事件时间戳（当前会话时区下
    的时间戳，如：Asia/Shanghai）。
6     type STRING,                                        -- Github事件类型，如：。
    ForkEvent，WatchEvent，IssuesEvent，CommitCommentEvent等。
7     actor_id STRING,                                    -- Github用户ID。
8     actor_login STRING,                                 -- Github用户名。
9     repo_id STRING,                                     -- Github仓库ID。
10    repo_name STRING,                                   -- Github仓库名，如:
    apache/flink，apache/spark，alibaba/fastjson等。
11    org STRING,                                         -- Github组织ID。
12    org_login STRING                                   -- Github组织名，如:
    apache,google,alibaba等。
13  ) WITH (
14    'connector' = 'sls',                                -- 实时采集的Github事件存放
    在阿里云SLS中。
15    'project' = 'github-events-shanghai',                       -- 存放公开数据的SLS
    项目。例如'github-events-hangzhou'。
16    'endPoint' = 'https://cn-shanghai-intranet.log.aliyuncs.com',
        -- 公开数据仅限VVP通过私网地址访问。例如'https://cn-hangzhou-
```

```
    intranet.log.aliyuncs.com'。
17    'logStore' = 'realtime-github-events',              -- 存放公开数据的SLS
    logStore。
18    'accessId' =  'LTAI5tNF1rP8PKVyYjr9TKgh',           -- 只读账号的AK。
19    'accessKey' = 'FDgsh6fjSmkbFsx083tN6HOiqNVWTP',     -- 只读账号的SK。
20    'batchGetSize' = '500',                             -- 批量读取数据，每批最多拉取
    500条。
21    'startTime' = '2025-03-18 14:00:00'                 -- 开始时间，尽量设置到需要计
    算的时间附近，否则无效计算的时间较长。默认值为当前值
22  );
23
24  -- 配置开启mini-batch，每2s处理一次。
25  SET 'table.exec.mini-batch.enabled'='true';
26  SET 'table.exec.mini-batch.allow-latency'='2s';
27  SET 'table.exec.mini-batch.size'='4096';
28
29  -- 作业设置4个并发，聚合更快。
30  SET 'parallelism.default' = '4';
31
32  -- 查看Github新增star数Top 5仓库。
33  SELECT DATE_FORMAT(created_at_ts, 'yyyy-MM-dd') as `date`, repo_name,
    COUNT(*) as num
34  FROM gh_event WHERE type = 'WatchEvent'
35  GROUP BY DATE_FORMAT(created_at_ts, 'yyyy-MM-dd'), repo_name
36  ORDER BY num DESC
37  LIMIT 5;
```

运行之后可以得出类似的结果。

实验二：统计组织活跃度变化

新建一个作业，设置保持默认即可。



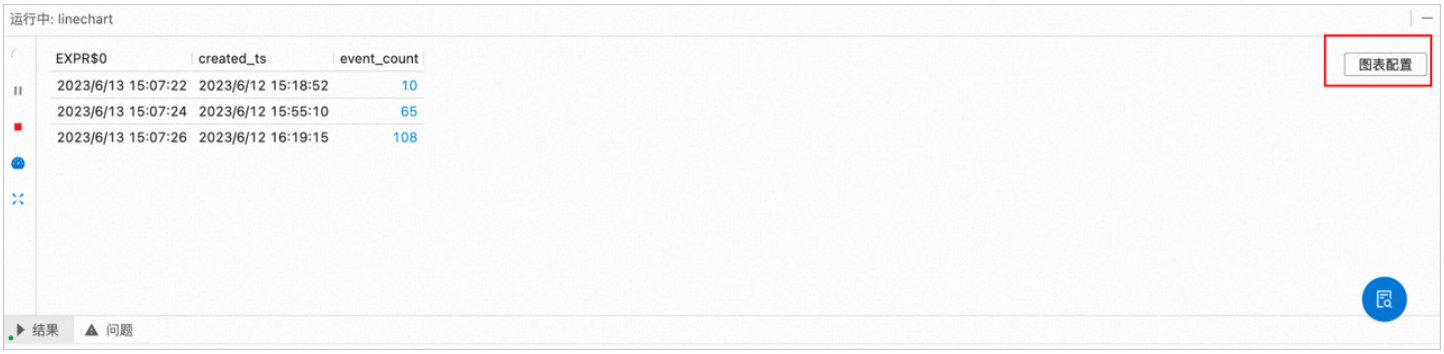

```sql
CREATE TEMPORARY TABLE gh_event(
  id STRING,                                    -- 每个事件的唯一ID。
  created_at BIGINT,                            -- 事件时间，单位秒。
  created_at_ts as TO_TIMESTAMP(created_at*1000),    -- 事件时间戳（当前会话时区下的时间戳，如：Asia/Shanghai）。
  type STRING,                                  -- Github事件类型，如：。ForkEvent, WatchEvent, IssuesEvent, CommitCommentEvent等。
  actor_id STRING,                              -- Github用户ID。
  actor_login STRING,                           -- Github用户名。
  repo_id STRING,                               -- Github仓库ID。
  repo_name STRING,                             -- Github仓库名，如：apache/flink，apache/spark，alibaba/fastjson等。
  org STRING,                                   -- Github组织ID。
  org_login STRING                             -- Github组织名，如：apache,google,alibaba等。
```

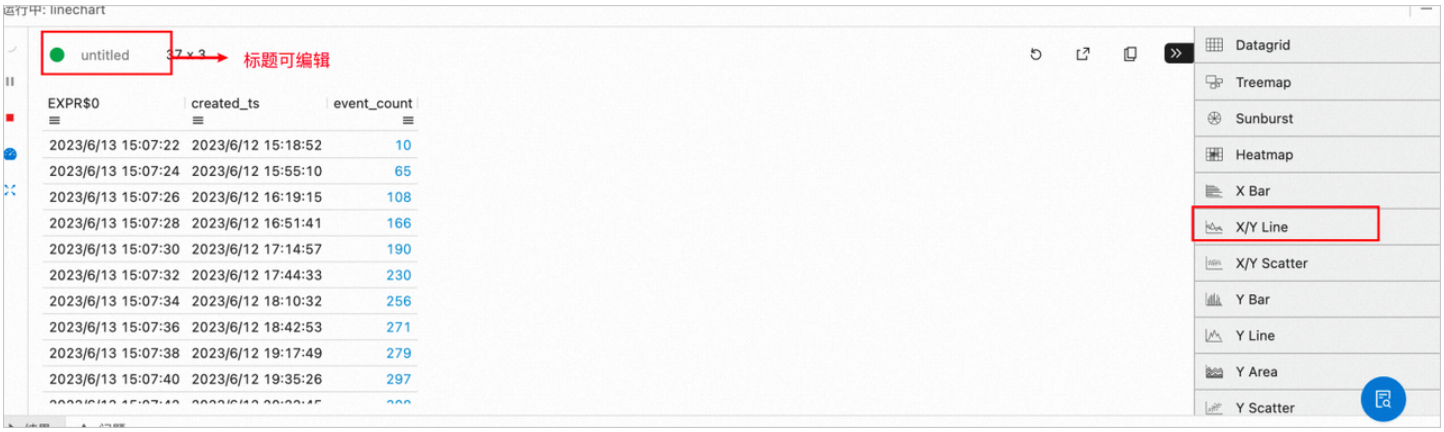
```
) WITH (
  'connector' = 'sls',                                    -- 实时采集的Github事件存放在阿里云SLS中。
  'project' = 'github-events-shanghai',                   -- 存放公开数据的SLS项目。例如'github-events-hangzhou'。
  'endPoint' = 'https://cn-shanghai-intranet.log.aliyuncs.com',
    -- 公开数据仅限VVP通过私网地址访问。例如'https://cn-hangzhou-intranet.log.aliyuncs.com'。
  'logStore' = 'realtime-github-events',                  -- 存放公开数据的SLS logStore。
  'accessId' =  'LTAI5tNF1rP8PKVyYjr9TKgh',               -- 只读账号的AK。
  'accessKey' = 'FDgsh6fjSmkbFsx083tN6HOiqNVWTP',         -- 只读账号的SK。
  'batchGetSize' = '500',                                 -- 批量读取数据，每批最多拉取500条。
  'startTime' = '2025-03-18 16:00:00'                     -- 开始时间，尽量设置到需要计算的时间附近，否则无效计算的时间较长
);

-- 配置开启mini-batch，每2s处理一次。
SET 'table.exec.mini-batch.enabled'='true';
SET 'table.exec.mini-batch.allow-latency'='2s';
SET 'table.exec.mini-batch.size'='4096';

-- 作业设置4个并发，聚合更快。
SET 'parallelism.default' = '4';

-- 从一天前开始统计事件总量
SELECT NOW(), max(created_at_ts) as created_ts, COUNT(*) as event_count
FROM gh_event
WHERE  org_login ='apache' and
created_at_ts >= NOW() - INTERVAL '1' DAY;
```





编辑标题为**apache**，并且选择**X/Y Line**。

配置X轴为create_ts, y 轴为event_count。

实验 三: 统计仓库贡献时间分布情况

```
CREATE TEMPORARY TABLE gh_event(
    id STRING,                                          -- 每个事件的唯一ID。
    created_at BIGINT,                                  -- 事件时间，单位秒。
    created_at_ts as TO_TIMESTAMP(created_at*1000),     -- 事件时间戳（当前会话时区下的时间戳，如：Asia/Shanghai）。
    type STRING,                                        -- Github事件类型，如：。ForkEvent, WatchEvent, IssuesEvent, CommitCommentEvent等。
    actor_id STRING,                                    -- Github用户ID。
    actor_login STRING,                                 -- Github用户名。
    repo_id STRING,                                     -- Github仓库ID。
    repo_name STRING,                                   -- Github仓库名，如：apache/flink, apache/spark, alibaba/fastjson等。
    org STRING,                                         -- Github组织ID。
    org_login STRING                                    -- Github组织名，如：apache,google,alibaba等。
) WITH (
  'connector' = 'sls',                                  -- 实时采集的Github事件存放在阿里云SLS中。
  'project' = 'github-events-shanghai',                 -- 存放公开数据的SLS项目。例如'github-events-hangzhou'。
  'endPoint' = 'https://cn-shanghai-intranet.log.aliyuncs.com',
    -- 公开数据仅限VVP通过私网地址访问。例如'https://cn-hangzhou-intranet.log.aliyuncs.com'。
  'logStore' = 'realtime-github-events',                -- 存放公开数据的SLS logStore。
  'accessId' =  'LTAI5tNF1rP8PKVyYjr9TKgh',             -- 只读账号的AK。
  'accessKey' = 'FDgsh6fjSmkbFsx083tN6HOiqNVWTP',       -- 只读账号的SK。
  'batchGetSize' = '500',                               -- 批量读取数据，每批最多拉取500条。
```

```
  'startTime' = '2025-03-12 14:00:00'                -- 开始时间，尽量设置到需要计算的时间附近，否则无效计算的时间较长
);

-- 配置开启mini-batch，每2s处理一次。
SET 'table.exec.mini-batch.enabled'='true';
SET 'table.exec.mini-batch.allow-latency'='2s';
SET 'table.exec.mini-batch.size'='4096';

-- 作业设置4个并发，聚合更快。
SET 'parallelism.default' = '4';

-- 统计从上周起的贡献量
SELECT  DATE_FORMAT(created_at_ts, 'yyyy-MM-dd') as comment_date,
HOUR(created_at_ts) AS comment_hour ,COUNT(*) AS comment_count
FROM gh_event
WHERE created_at_ts >= NOW() - INTERVAL '7' DAY
        AND repo_name = 'apache/flink'
       AND (type ='CommitCommentEvent' OR
            type='IssueCommentEvent' or
            type = 'PullRequestReviewCommentEvent'or
            type = 'PushEvent' or
            type = 'PullRequestEvent' or
            type = 'PullRequestReviewEvent')
GROUP BY DATE_FORMAT(created_at_ts, 'yyyy-MM-dd'), HOUR(created_at_ts) ;
```

其中startTime尽量设置为当前此刻的一周前附近，如果设置的时间太早，前面无效计算时间比较长，不仅耗费资源，而且很久才能加载出计算结果。如果想要统计spark, 改成repo_name = 'apache/spark'' 即可。

选择Heatmap, 设置Group by为comment_date, Spli By为comment_hour，Color为Sum(comment_count)， 即X轴为天，Y周为小时，根据总数量显示颜色深浅。

做做完记得释放资源。

作业：

1. 熟练掌握实验工作流程。
2. 完成实验一、二、三，可对sql语句给出理解，并对实验结果进行简要分析。