# DS任务调度

## DophinScheduler任务调度

Apache DolphinScheduler（目前处在孵化阶段）是一个分布式、去中心化、易扩展的可视化DAG工作流任务调度系统，其致力于解决数据处理流程中错综复杂的依赖关系，使调度系统在数据处理流程中开箱即用。

DolphinScheduler是2019年中国易观公司开源的一个调度系统，在美国时间2019年8月29号，易观开源的分布式任务调度引擎DolphinScheduler（原EasyScheduler）正式通过顶级开源组织Apache基金会的投票决议，根据Apache基金会邮件列表显示，在包含11个约束性投票(binding votes)和2个无约束性投票(non-binding votes)的投票全部持赞同意见，无弃权票和反对票，投票顺利通过，这样便以全票通过的优秀表现正式成为了Apache孵化器项目！

### DS项目背景

在2017年，易观在运营自己 6.8Pb 大小、6.02亿月活、每天近万个调度任务的大数据平台时，受到ETL复杂的依赖关系、平台易用性、可维护性及二次开发等方面掣肘，易观的技术团队渴望找到一个具有以下功能的数据调度工具：

* 易于使用，开发人员可以通过非常简单的拖拽操作构建ETL过程。不仅对于ETL开发人员，无法编写代码的人也可以使用此工具进行ETL操作，例如系统管理员和分析师
* 解决“复杂任务依赖”问题，并且可以实时监视ETL运行状态
* 支持多租户
* 支持许多任务类型：Shell，MR，Flink，SQL（mysql，postgresql，hive，sparksql），Python，Sub\_Process，Procedure等
* 支持HA和线性可扩展性

易观技术团队意识到现有开源项目没有能够达到他们要求的，因此决定自行开发这个工具。他们在2017年底设计了DolphinScheduler的主要架构；2018年5月完成第一个内部使用版本，后来又迭代了几个内部版本后，系统逐渐稳定下来。

关于DolphinScheduler易观技术团队，他们是一支来自百度、阿里、百分点、Ptmind、热云等团队的“数据极客”，秉持易观“让数据能力平民化”的初心，积极拥抱开源，曾贡献过Presto Hbase Connector, Presto Kudu Connector等令开发者称赞的项目。这次他们在公司的支持下，积极地将自己开发的调度工具推动开源，旨在回馈开源的同时，助力打造一个更为强大的开源生态。如果跃跃欲试的想去贡献代码的，贡献流程可以参考这篇博客：分布式任务调度EasyScheduler贡献代码流程。

团队在2019年3月初，小范围(10多家公司)开放了DS的种子用户试用，得到了非常正能量的反馈，在4月初的正式对外开放源码后，很快就获得了许多开发人员的关注兴趣，目前github上的star现在已超过1700个，参与开发和使用的公司包括嘀嗒出行、雪球、凤凰金融、水滴互助、华润万家等。



### DS特点

* DolphinScheduler提供了许多易于使用的功能，可加快数据ETL工作开发流程的效率。其主要特点如下：
  + 通过拖拽以DAG 图的方式将 Task 按照任务的依赖关系关联起来，可实时可视化监控任务的运行状态；
  + 支持丰富的任务类型；
  + 支持工作流定时调度、依赖调度、手动调度、手动暂停/停止/恢复，同时支持失败重试/告警、从指定节点恢复失败、Kill 任务等操作；
  + 支持工作流全局参数及节点自定义参数设置；
  + 支持集群HA，通过 Zookeeper实现 Master 集群和 Worker 集群去中心化；
  + 支持工作流运行历史树形/甘特图展示、支持任务状态统计、流程状态统计；
* 支持补数，并行或串行回填数据。
* 中文官方网址：<http://dolphinscheduler.apache.org/zh-cn/index.html>

### DS安装

* DolphinScheduler安装分为后端安装和前端安装
* 安装教程：[讲义关联资料\DolphinScheduler安装.md](讲义关联资料/DolphinScheduler安装.md)

## DolphinScheduler基础操作

### DS初体验

* 创建队列



* 创建租户



* 创建普通用户



* 创建告警组



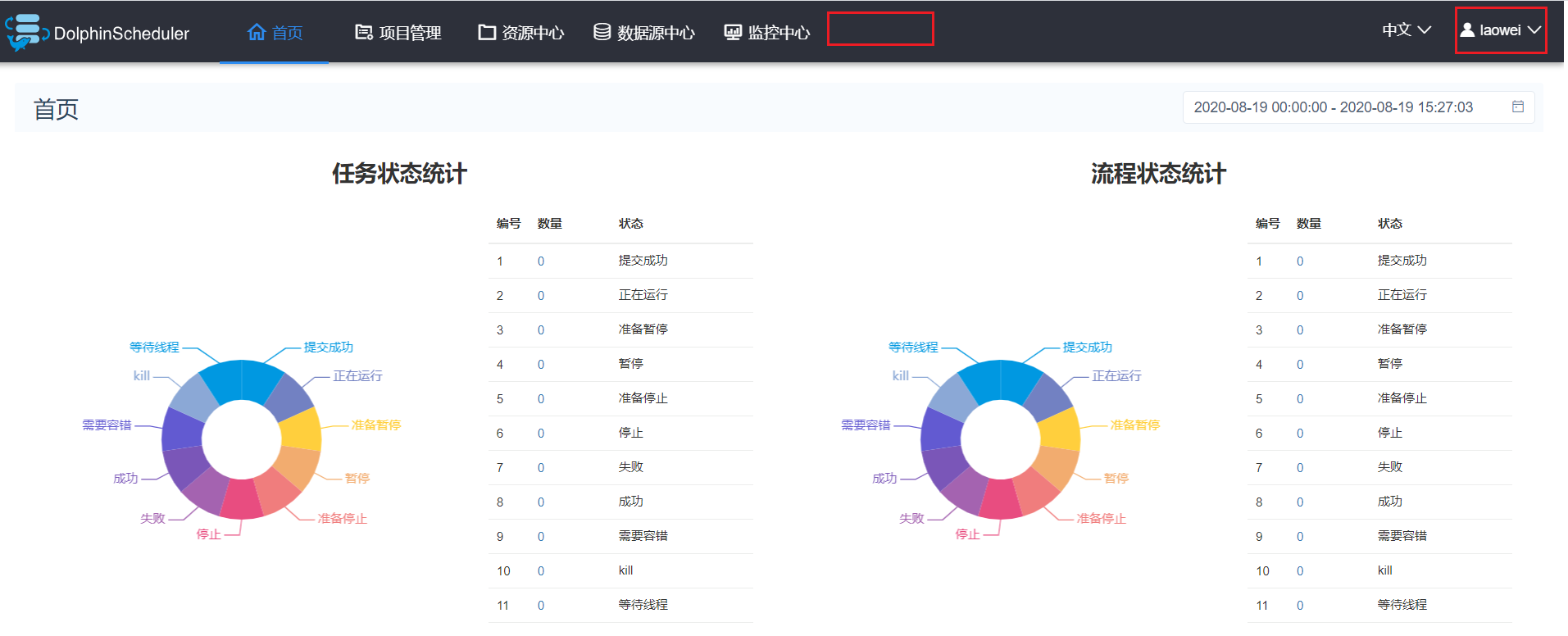
* 创建Worker分组



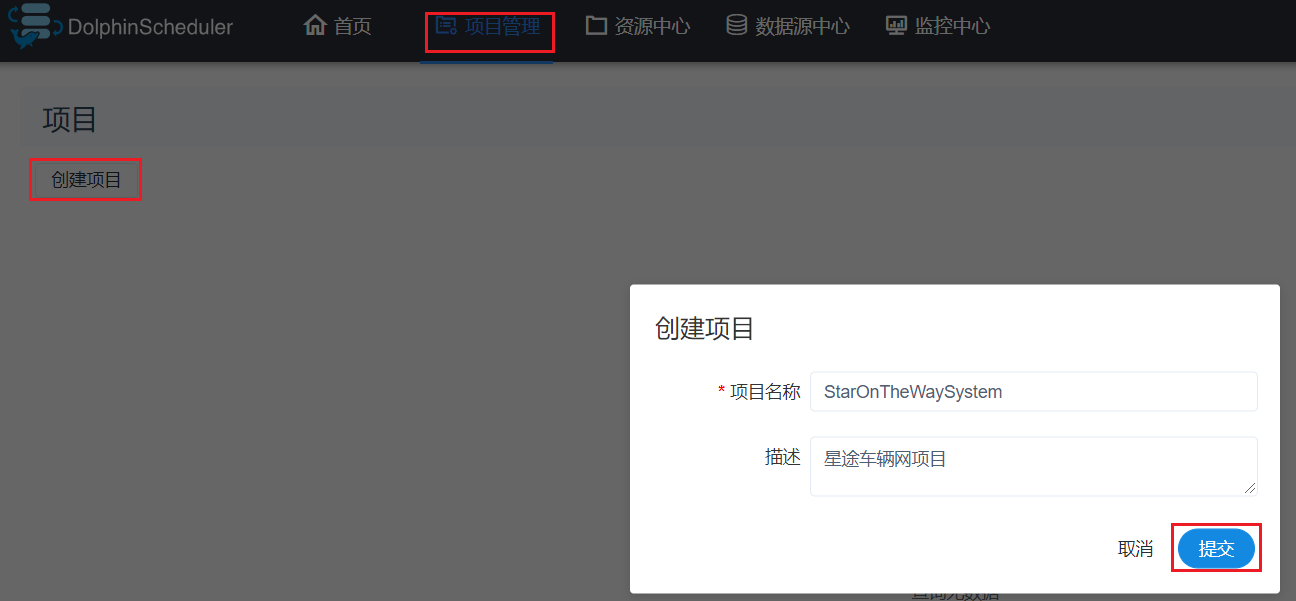
* 创建token令牌



* 切换用户
  + 点击右上角用户名“退出”，切换使用非管理员用户登录



* 创建项目

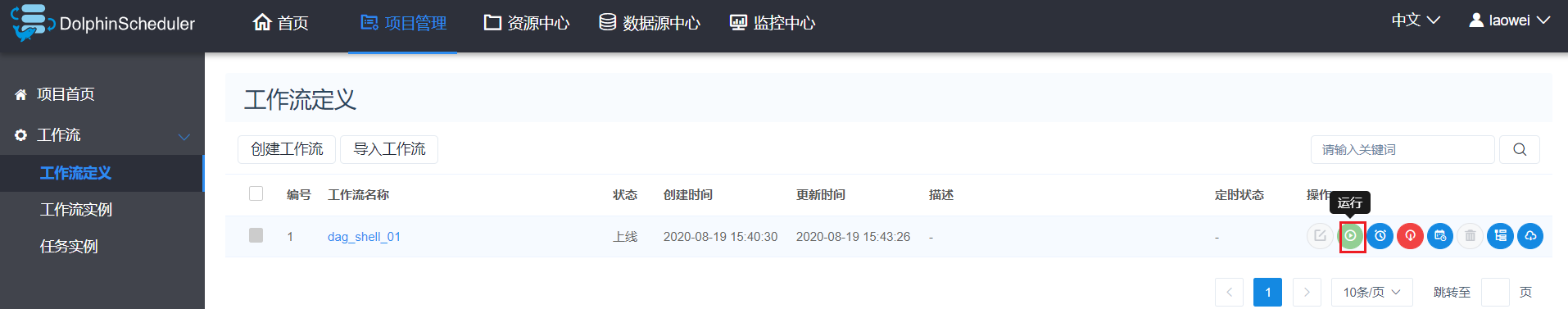


* 工作流定义



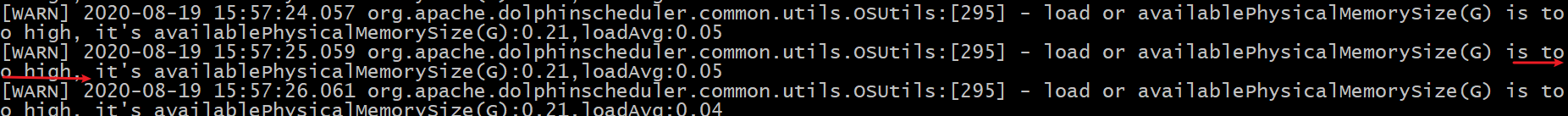
* 工作流运行



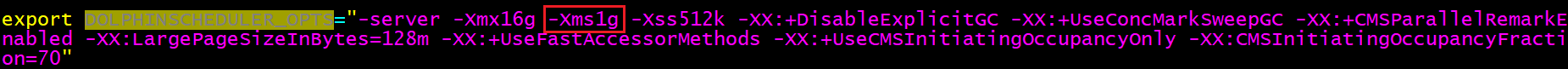




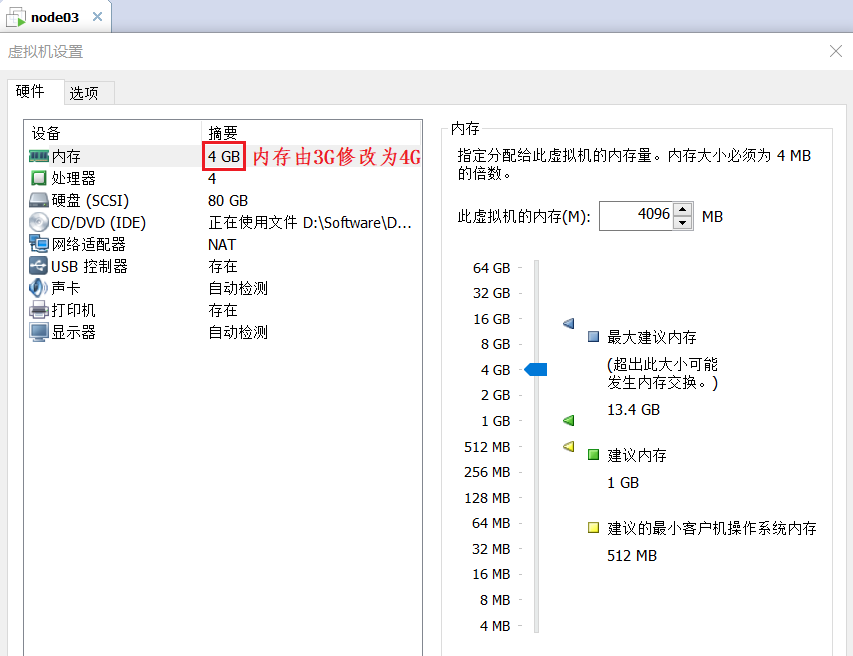
* 如遇到任务一直卡顿，查看woker日志，日志显示消耗内存过大，图示



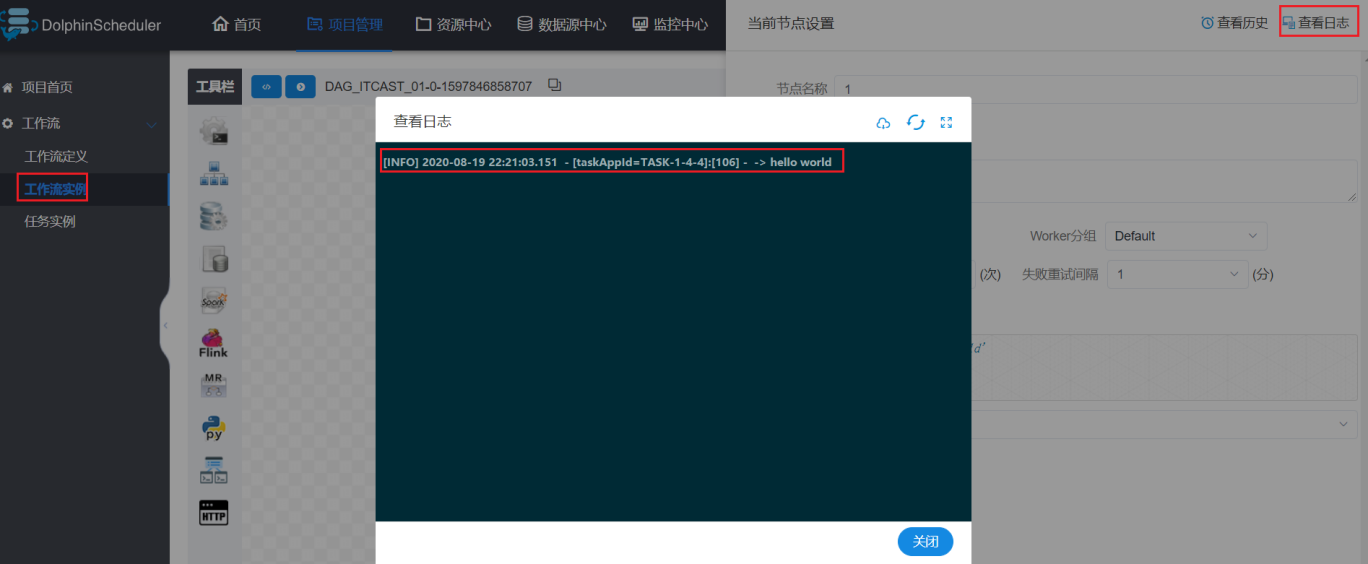
* 原因：DS默认任务执行内存为4G，两个woker节点的虚拟机内存本身只有3G，加上启动的组件服务，再减去部分内存，因此ds需要执行任务的内存不够，任务一直会处在等待状态
  + 一方面修改dolphinscheduler-daemon.sh脚本中的启动参数的-Xms值



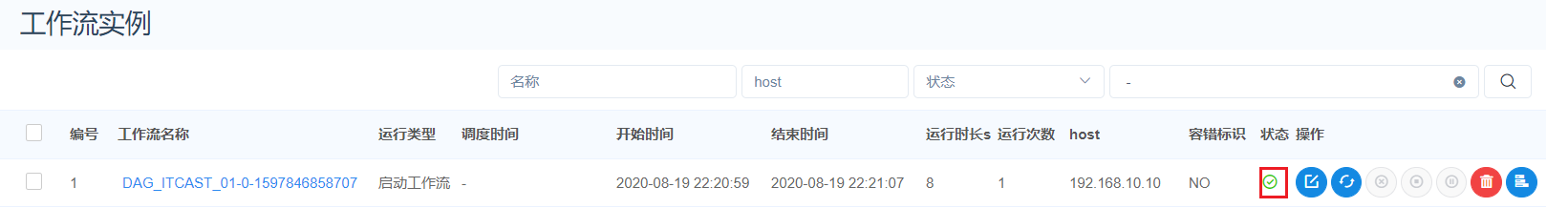
* + 土豪可以选择简单粗暴方式，扩大内存



* 在工作流实例中双击实例名称，查看日志



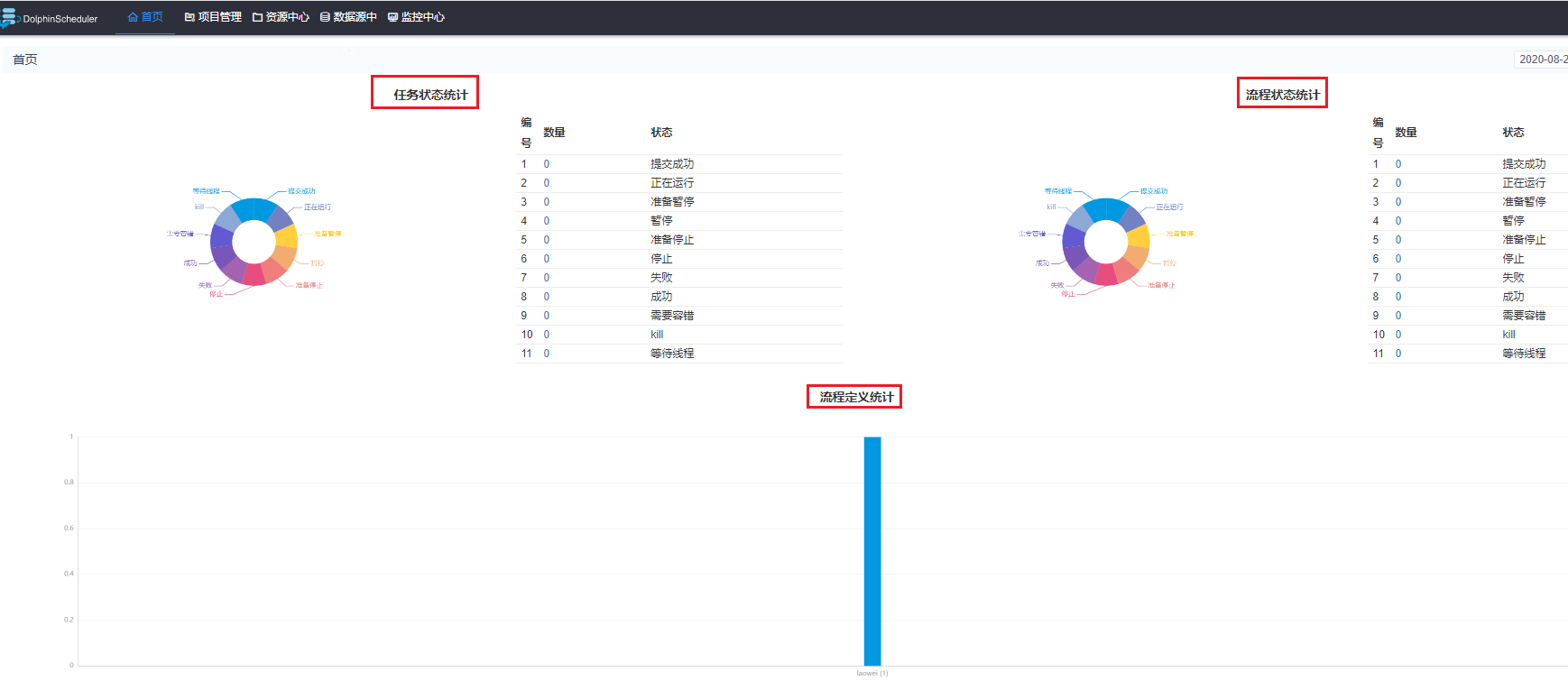
* 查看任务状态，工作流任务执行成功



### DS界面操作指南

#### 首页

* 首页包含用户所有项目的任务状态统计、流程状态统计、工作流定义统计



#### 项目管理

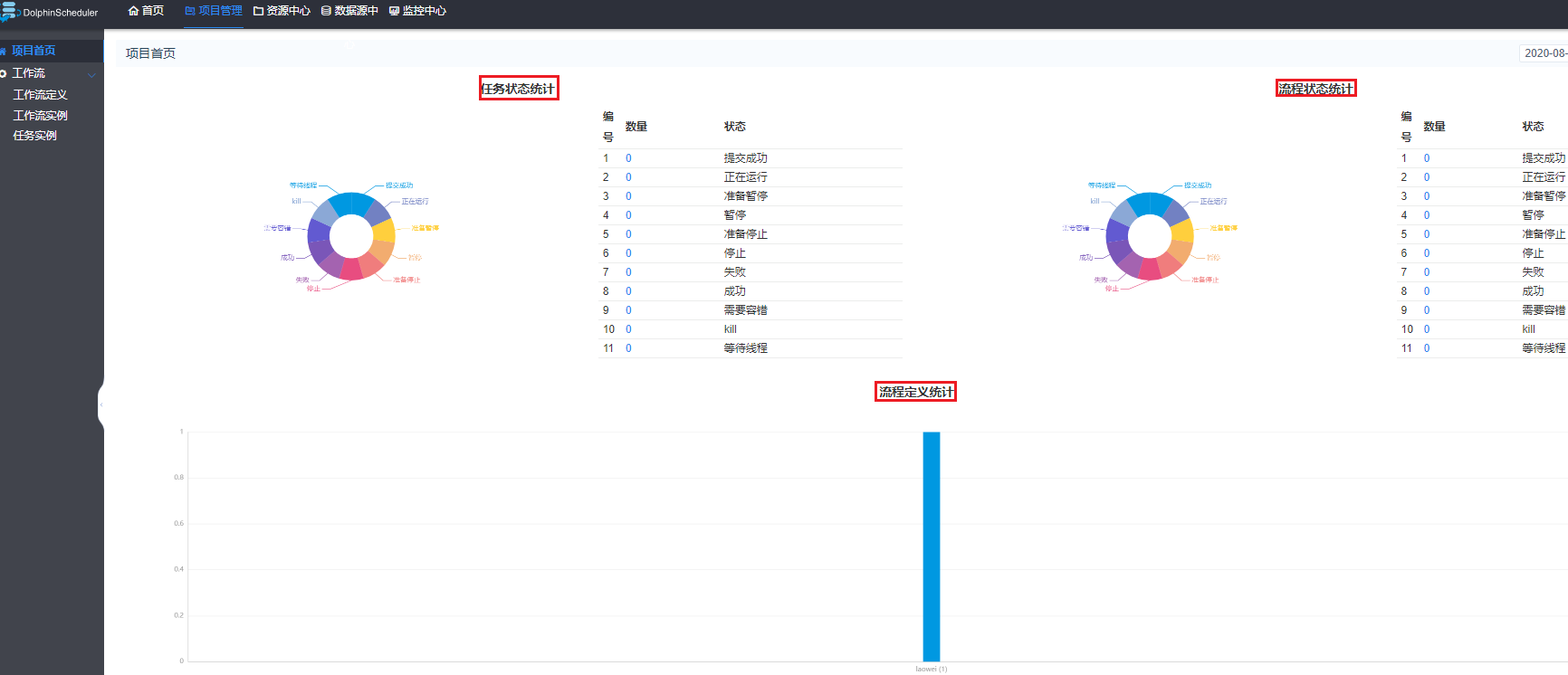
##### 创建项目

* 点击"项目管理"进入项目管理页面，点击“创建项目”按钮，输入项目名称，项目描述，点击“提交”，创建新的项目



##### 项目首页

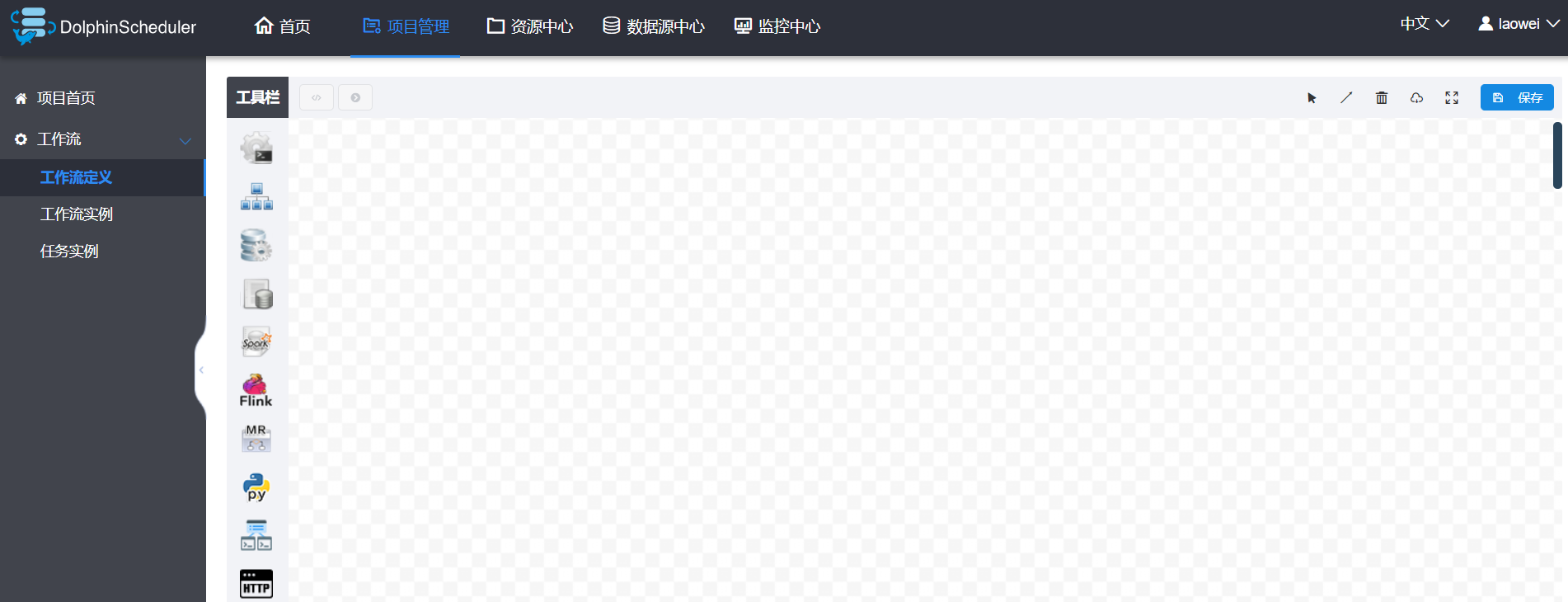
* 在项目管理页面点击项目名称链接，进入项目首页，如下图所示，项目首页包含该项目的任务状态统计、流程状态统计、工作流定义统计。



* + 任务状态统计：在指定时间范围内，统计任务实例中状态为提交成功、正在运行、准备暂停、暂停、准备停止、停止、失败、成功、需要容错、kill、等待线程的个数
  + 流程状态统计：在指定时间范围内，统计工作流实例中状态为提交成功、正在运行、准备暂停、暂停、准备停止、停止、失败、成功、需要容错、kill、等待线程的个数
  + 工作流定义统计：统计用户创建的工作流定义及管理员授予该用户的工作流定义

##### 工作流定义

* 创建工作流定义
  + 点击项目管理->工作流->工作流定义，进入工作流定义页面，点击“创建工作流”按钮，进入工作流DAG编辑页面，如下图所示：



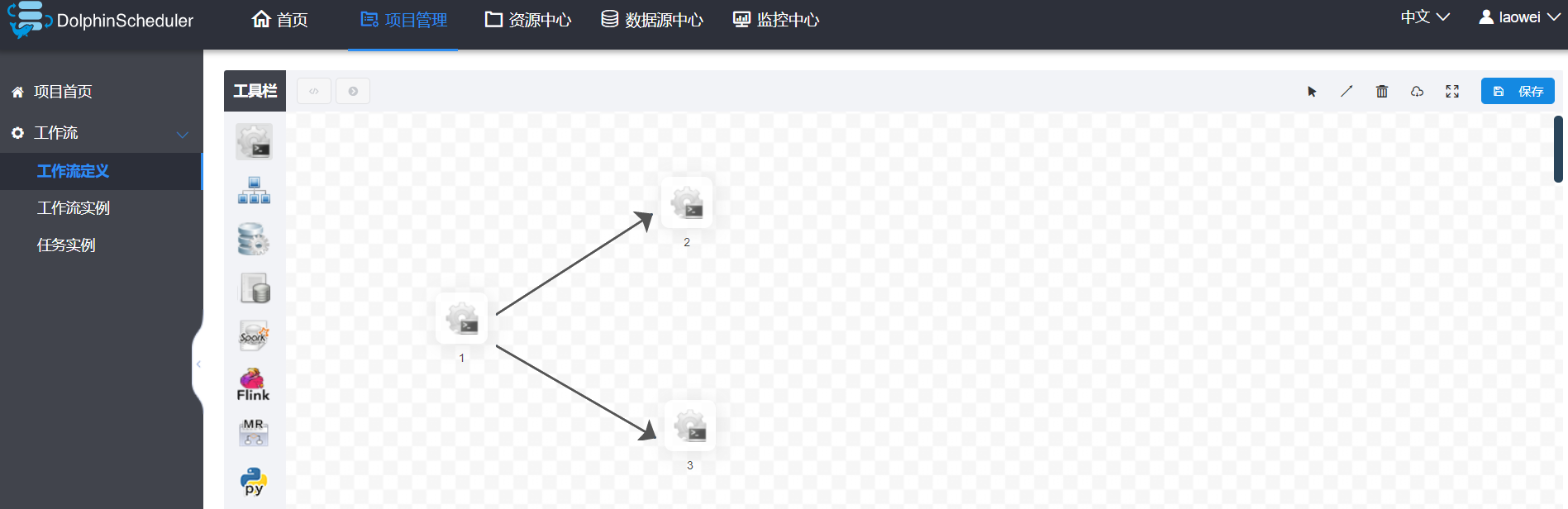
* 工具栏中拖拽到画板中，新增一个Shell任务,如下图所示



* + 添加shell任务的参数设置：

1. 填写“节点名称”，“描述”，“脚本”字段；
2. “运行标志”勾选“正常”，若勾选“禁止执行”，运行工作流不会执行该任务；
3. 选择“任务优先级”：当worker线程数不足时，级别高的任务在执行队列中会优先执行，相同优先级的任务按照先进先出的顺序执行；
4. 超时告警（非必选）：勾选超时告警、超时失败，填写“超时时长”，当任务执行时间超过超时时长，会发送告警邮件并且任务超时失败；
5. 资源（非必选）。资源文件是资源中心->文件管理页面创建或上传的文件，如文件名为test.sh，脚本中调用资源命令为sh test.sh；
6. 自定义参数（非必填），参考[自定义参数](讲义关联资料/DS创建任务自定义参数.md)；
7. 点击"确认添加"按钮，保存任务设置。

* **增加任务执行的先后顺序**：点击右上角图标IMG_256连接任务；如下图所示，任务2和任务3并行执行，当任务1执行完，任务2、3会同时执行。



* **删除依赖关系**：点击右上角"箭头"图标IMG_256，选中连接线，点击右上角"删除"图标IMG_257，删除任务间的依赖关系。



* **保存工作流定义：**点击”保存“按钮，弹出"设置DAG图名称"弹框，如下图所示，输入工作流定义名称，工作流定义描述，设置全局参数（选填，参考[自定义参数](讲义关联资料/DS创建任务自定义参数.md)），点击"添加"按钮，工作流定义创建成功。



* 其他类型任务：[任务节点类型和参数设置](讲义关联资料/DS任务节点类型和参数设置.md)
* 工作流定义操作功能

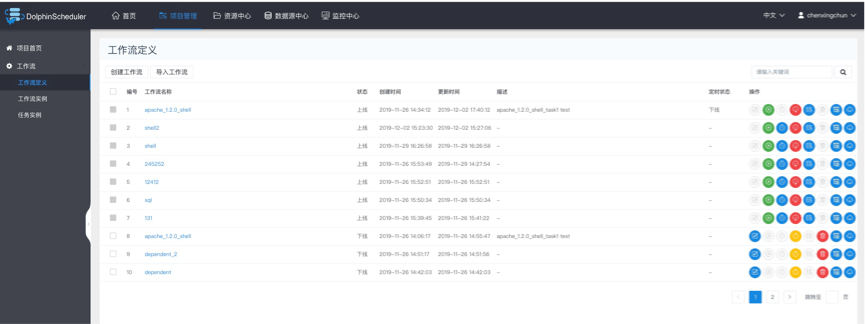
点击项目管理->工作流->工作流定义，进入工作流定义页面，如下图所示:



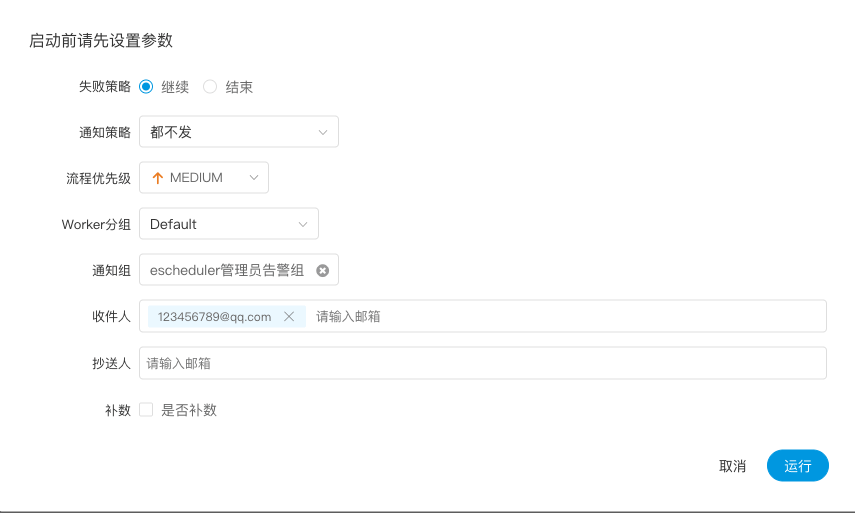
* 工作流定义列表的操作功能如下：
* 编辑： 只能编辑"下线"的工作流定义。工作流DAG编辑同[创建工作流定义](http://dolphinscheduler.apache.org/zh-cn/docs/1.2.0/user_doc/system-manual.html" \l "creatDag)。
* 上线： 工作流状态为"下线"时，上线工作流，只有"上线"状态的工作流能运行，但不能编辑。
* 下线： 工作流状态为"上线"时，下线工作流，下线状态的工作流可以编辑，但不能运行。
* 运行： 只有上线的工作流能运行。运行操作步骤见[2.3.3 运行工作流](http://dolphinscheduler.apache.org/zh-cn/docs/1.2.0/user_doc/system-manual.html" \l "runWorkflow)
* 定时： 只有上线的工作流能设置定时，系统自动定时调度工作流运行。创建定时后的状态为"下线"，需在定时管理页面上线定时才生效。定时操作步骤见[2.3.4 工作流定时](http://dolphinscheduler.apache.org/zh-cn/docs/1.2.0/user_doc/system-manual.html" \l "creatTiming)。
* 定时管理： 定时管理页面可编辑、上线/下线、删除定时。
* 删除： 删除工作流定义。
* 下载： 下载工作流定义到本地。
* 树形图： 以树形结构展示任务节点的类型及任务状态，如下图所示：



* 运行工作流
  + 点击项目管理->工作流->工作流定义，进入工作流定义页面，如下图所示，点击"上线"按钮，上线工作流。



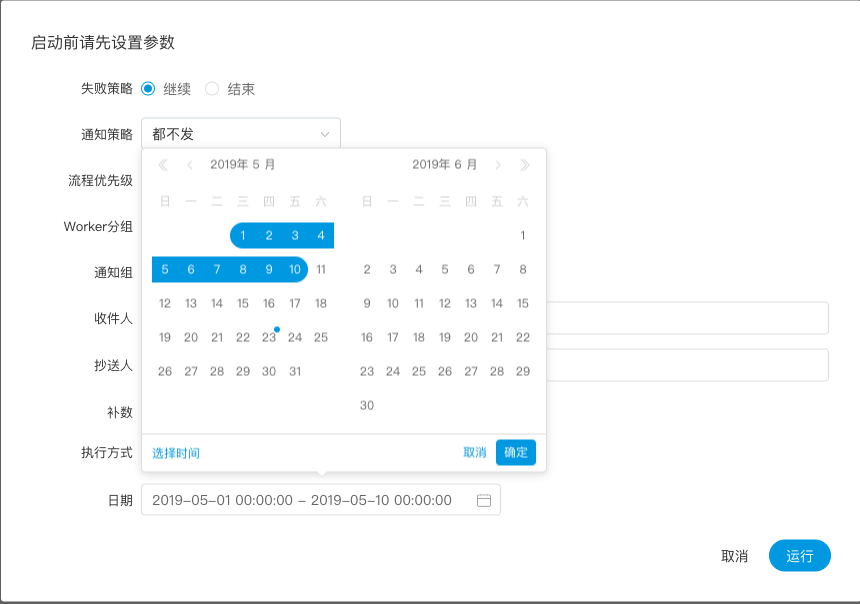
* + 点击”运行“按钮，弹出启动参数设置弹框，如下图所示，设置启动参数，点击弹框中的"运行"按钮，工作流开始运行，工作流实例页面生成一条工作流实例。



* + - 工作流运行参数说明：

|  |
| --- |
| \* 失败策略：当某一个任务节点执行失败时，其他并行的任务节点需要执行的策略。”继续“表示：某一任务失败后，其他任务节点正常执行；”结束“表示：终止所有正在执行的任务，并终止整个流程。  \* 通知策略：当流程结束，根据流程状态发送流程执行信息通知邮件，包含任何状态都不发，成功发，失败发，成功或失败都发。  \* 流程优先级：流程运行的优先级，分五个等级：最高（HIGHEST），高(HIGH),中（MEDIUM）,低（LOW），最低（LOWEST）。当master线程数不足时，级别高的流程在执行队列中会优先执行，相同优先级的流程按照先进先出的顺序执行。  \* worker分组：该流程只能在指定的worker机器组里执行。默认是Default，可以在任一worker上执行。  \* 通知组：选择通知策略||超时报警||发生容错时，会发送流程信息或邮件到通知组里的所有成员。  \* 收件人：选择通知策略||超时报警||发生容错时，会发送流程信息或告警邮件到收件人列表。  \* 抄送人：选择通知策略||超时报警||发生容错时，会抄送流程信息或告警邮件到抄送人列表。  \* 补数：包括串行补数、并行补数2种模式。串行补数：指定时间范围内，从开始日期至结束日期依次执行补数，只生成一条流程实例；并行补数：指定时间范围内，多天同时进行补数，生成N条流程实例。 |

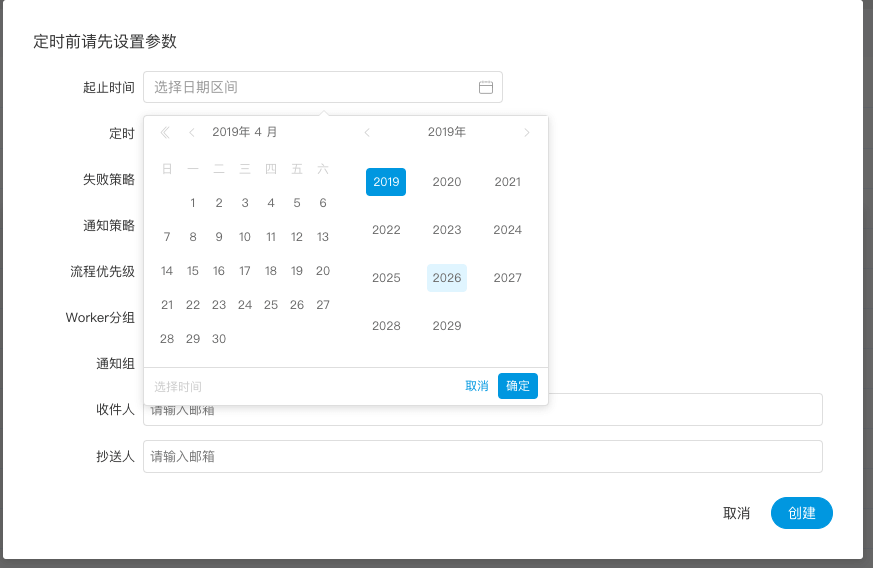
* 补数： 执行指定日期的工作流定义，可以选择补数时间范围（目前只支持针对连续的天进行补数)，比如需要补5月1号到5月10号的数据，如下图所示：



串行模式：补数从5月1号到5月10号依次执行，流程实例页面生成一条流程实例；

并行模式：同时执行5月1号到5月10号的任务，流程实例页面生成十条流程实例。

* 工作流定时
  + 创建定时：点击项目管理->工作流->工作流定义，进入工作流定义页面，上线工作流，点击"定时"按钮,弹出定时参数设置弹框，如下图所示：



* + 选择起止时间。在起止时间范围内，定时运行工作流；不在起止时间范围内，不再产生定时工作流实例。
  + 添加一个每天凌晨5点执行一次的定时，如下图所示：



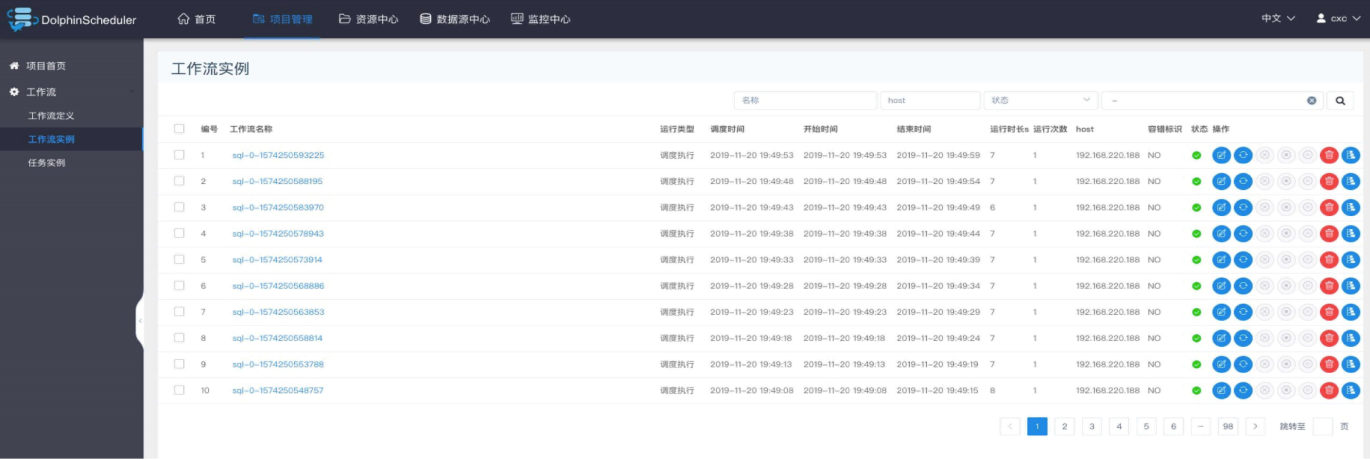
* + 失败策略、通知策略、流程优先级、Worker分组、通知组、收件人、抄送人同[工作流运行参数](http://dolphinscheduler.apache.org/zh-cn/docs/1.2.0/user_doc/system-manual.html" \l "runParamers)。
  + 点击"创建"按钮，创建定时成功，此时定时状态为"下线"，定时需上线才生效。
  + 定时上线：点击"定时管理"按钮，进入定时管理页面，点击"上线"按钮，定时状态变为"上线"，如下图所示，工作流定时生效。



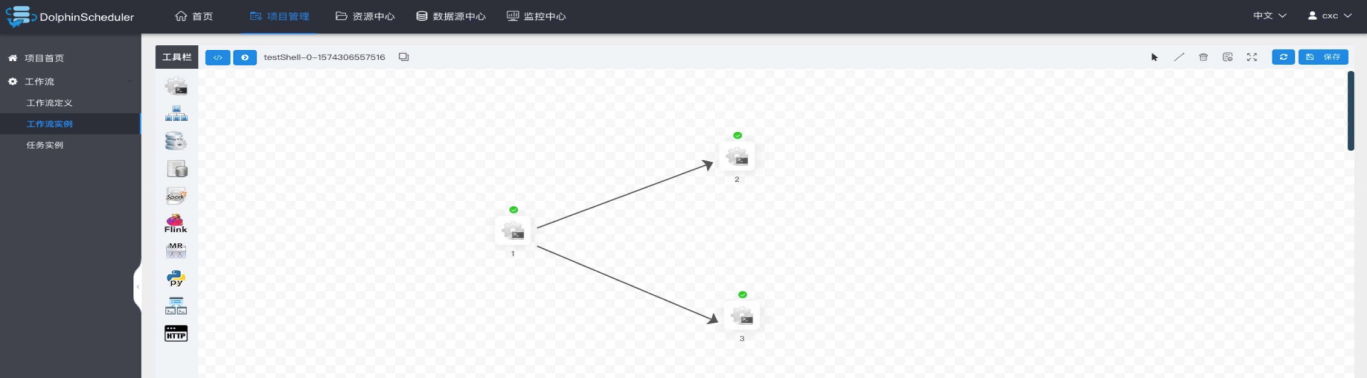
* 导入工作流
  + 点击项目管理->工作流->工作流定义，进入工作流定义页面，点击"导入工作流"按钮，导入本地工作流文件，工作流定义列表显示导入的工作流，状态为下线。

##### 工作流实例

* 查看工作流实例
  + 点击项目管理->工作流->工作流实例，进入工作流实例页面，如下图所示：



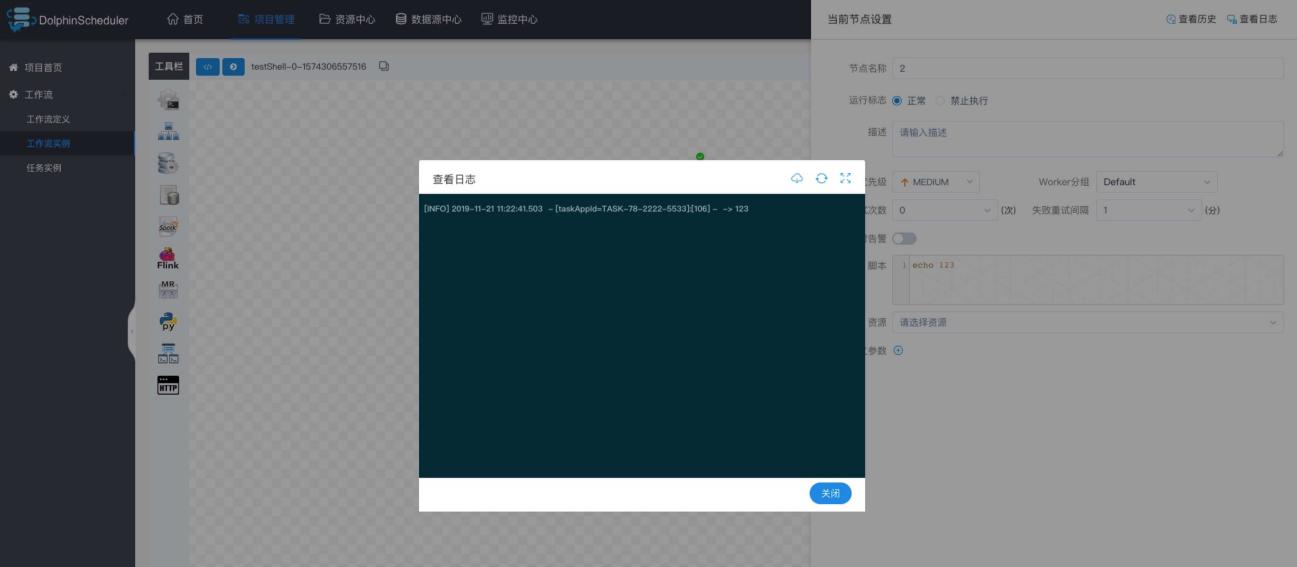
* 点击工作流名称，进入DAG查看页面，查看任务执行状态，如下图所示。



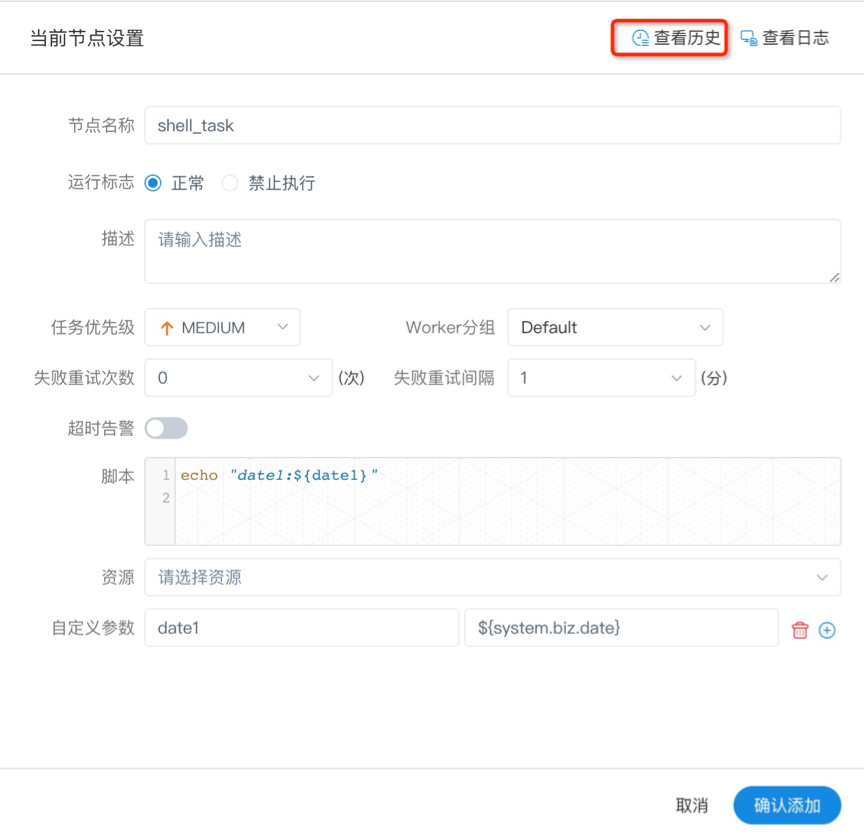
* 查看任务日志
  + 进入工作流实例页面，点击工作流名称，进入DAG查看页面，双击任务节点，如下图所示：



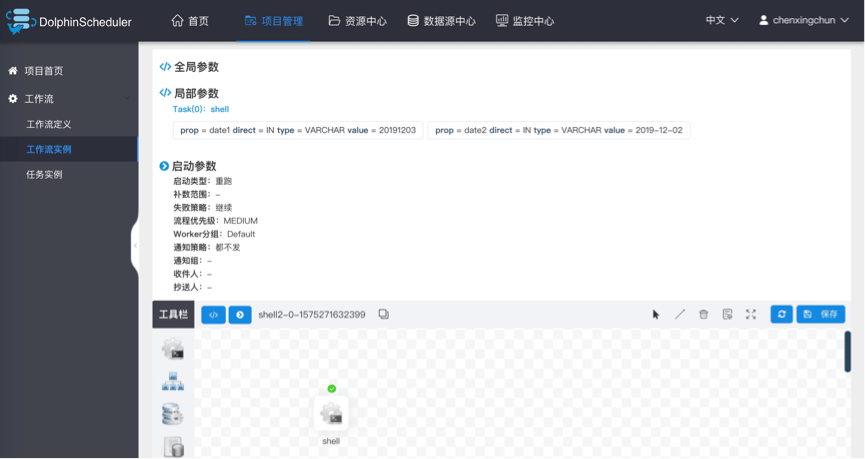
* 点击"查看日志"，弹出日志弹框，如下图所示,任务实例页面也可查看任务日志，参考[任务查看日志](http://dolphinscheduler.apache.org/zh-cn/docs/1.2.0/user_doc/system-manual.html" \l "taskLog)。



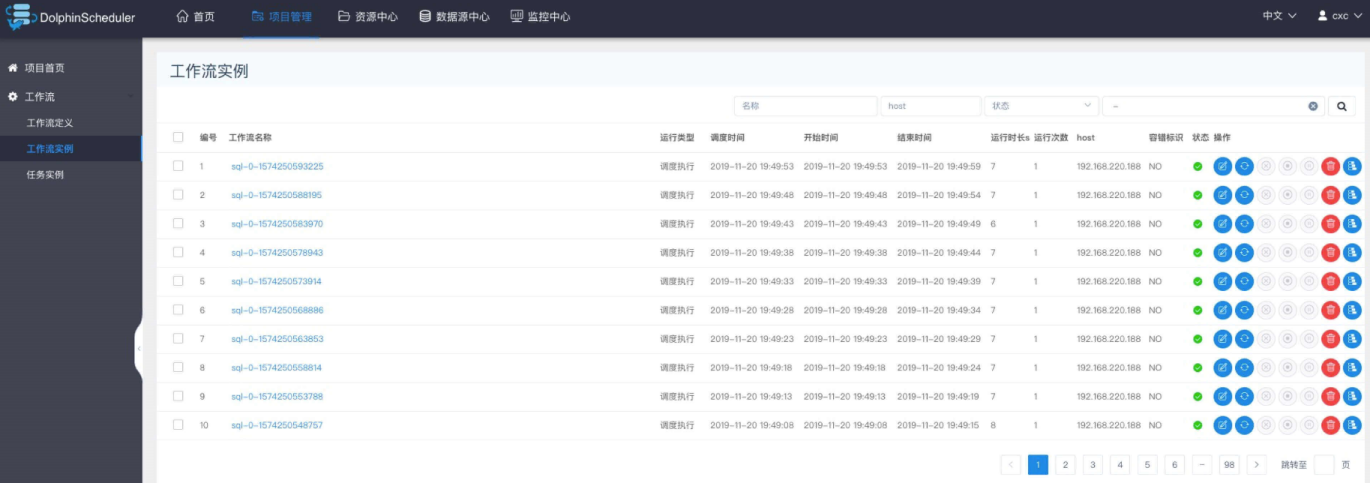
* 查看任务历史记录
  + 点击项目管理->工作流->工作流实例，进入工作流实例页面，点击工作流名称，进入工作流DAG页面;
  + 双击任务节点，如下图所示，点击"查看历史"，跳转到任务实例页面，并展示该工作流实例运行的任务实例列表



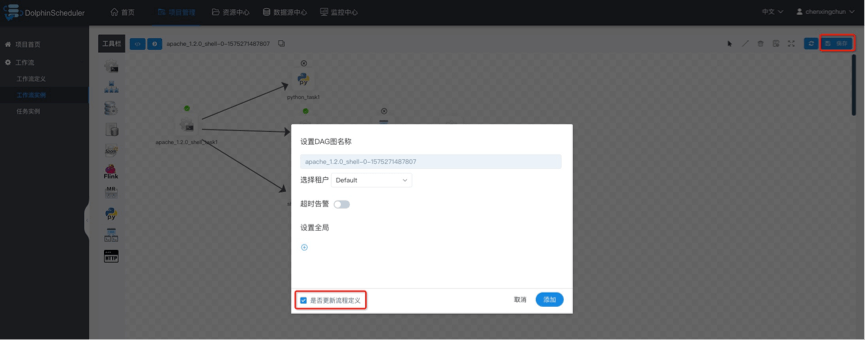
* 查看运行参数
  + 点击项目管理->工作流->工作流实例，进入工作流实例页面，点击工作流名称，进入工作流DAG页面;
  + 点击左上角图标，查看工作流实例的启动参数；点击图标，查看工作流实例的全局参数和局部参数，如下图所示：



* 工作流实例操作功能
  + 点击项目管理->工作流->工作流实例，进入工作流实例页面，如下图所示：



* 编辑： 只能编辑已终止的流程。点击"编辑"按钮或工作流实例名称进入DAG编辑页面，编辑后点击"保存"按钮，弹出保存DAG弹框，如下图所示，在弹框中勾选"是否更新到工作流定义"，保存后则更新工作流定义；若不勾选，则不更新工作流定义。

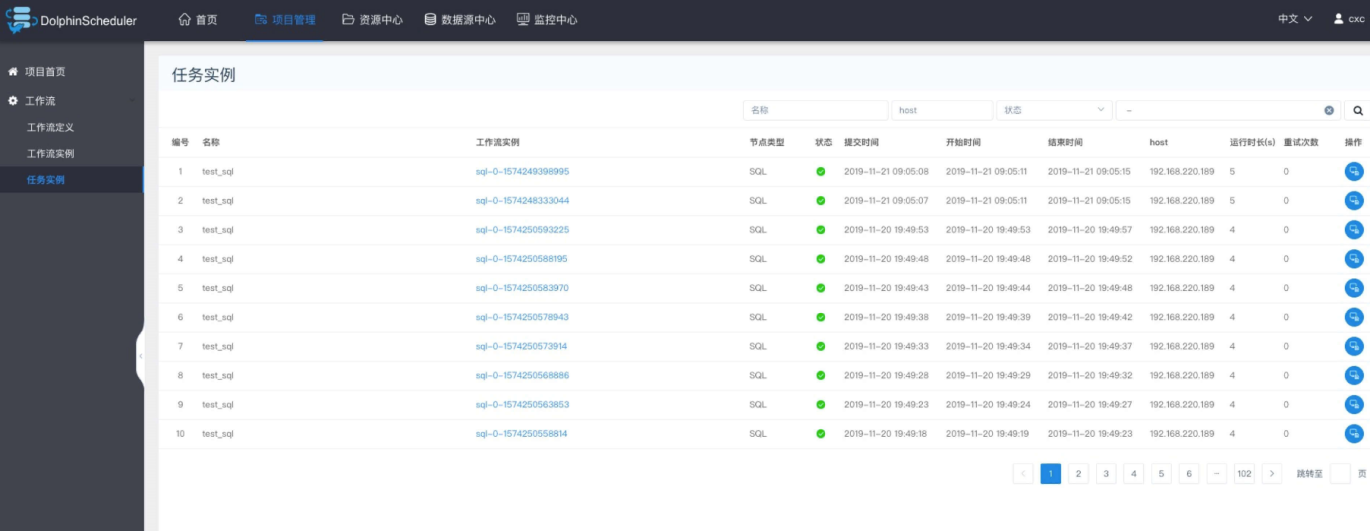


* + 重跑： 重新执行已经终止的流程。
  + 恢复失败： 针对失败的流程，可以执行恢复失败操作，从失败的节点开始执行。
  + 停止： 对正在运行的流程进行停止操作，后台会先killworker进程,再执行kill -9操作
  + 暂停： 对正在运行的流程进行暂停操作，系统状态变为等待执行，会等待正在执行的任务结束，暂停下一个要执行的任务。
  + 恢复暂停： 对暂停的流程恢复，直接从暂停的节点开始运行
  + 删除： 删除工作流实例及工作流实例下的任务实例
  + 甘特图： Gantt图纵轴是某个工作流实例下的任务实例的拓扑排序，横轴是任务实例的运行时间,如图示：

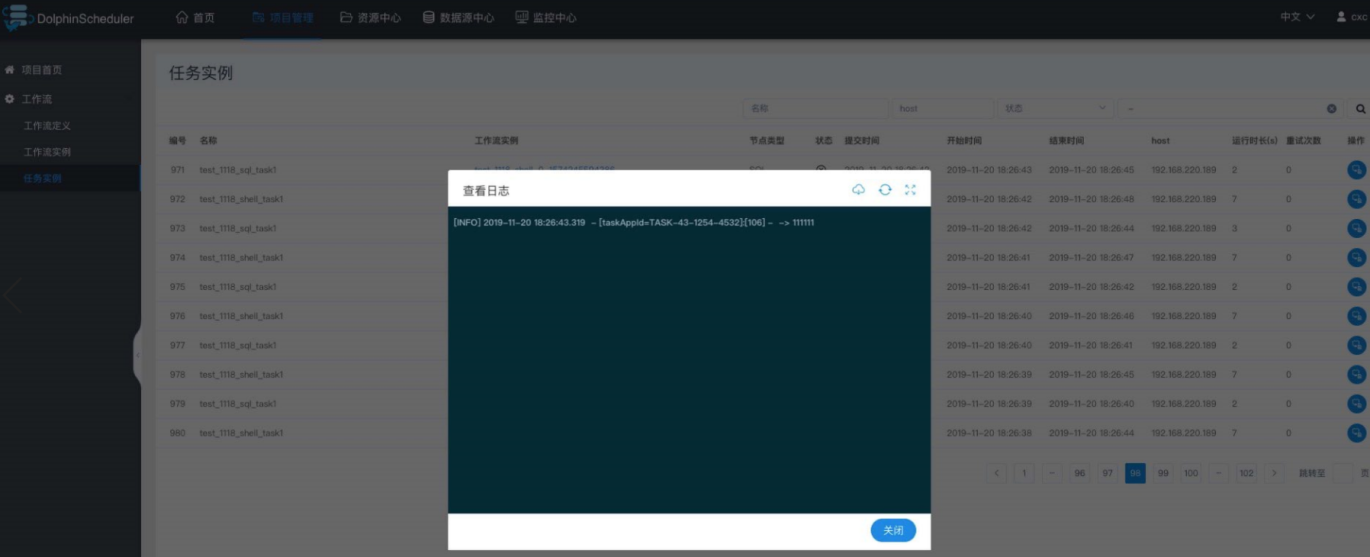


##### 任务实例

* 点击项目管理->工作流->任务实例，进入任务实例页面，如下图所示，点击工作流实例名称，可跳转到工作流实例DAG图查看任务状态。



* 查看日志：点击操作列中的“查看日志”按钮，可以查看任务执行的日志情况。



#### 资源中心

##### HDFS资源配置

* 上传资源文件和udf函数，所有上传的文件和资源都会被存储到hdfs上，所以需要以下配置项：

|  |
| --- |
| conf/common/common.properties  # Users who have permission to create directories under the HDFS root path  hdfs.root.user=hdfs  # data base dir, resource file will store to this hadoop hdfs path, self configuration, please make sure the directory exists on hdfs and have read write permissions。"/escheduler" is recommended  data.store2hdfs.basepath=/dolphinscheduler  # resource upload startup type : HDFS,S3,NONE  res.upload.startup.type=HDFS  # whether kerberos starts  hadoop.security.authentication.startup.state=false  # java.security.krb5.conf path  java.security.krb5.conf.path=/opt/krb5.conf  # loginUserFromKeytab user  login.user.keytab.username=hdfs-mycluster@ESZ.COM  # loginUserFromKeytab path  login.user.keytab.path=/opt/hdfs.headless.keytab |
| conf/common/hadoop.properties  # ha or single namenode,If namenode ha needs to copy core-site.xml and hdfs-site.xml  # to the conf directory，support s3，for example : s3a://dolphinscheduler  fs.defaultFS=hdfs://mycluster:8020  #resourcemanager ha note this need ips , this empty if single  yarn.resourcemanager.ha.rm.ids=192.168.xx.xx,192.168.xx.xx  # If it is a single resourcemanager, you only need to configure one host name. If it is resourcemanager HA, the default configuration is fine  yarn.application.status.address=http://xxxx:8088/ws/v1/cluster/apps/%s |

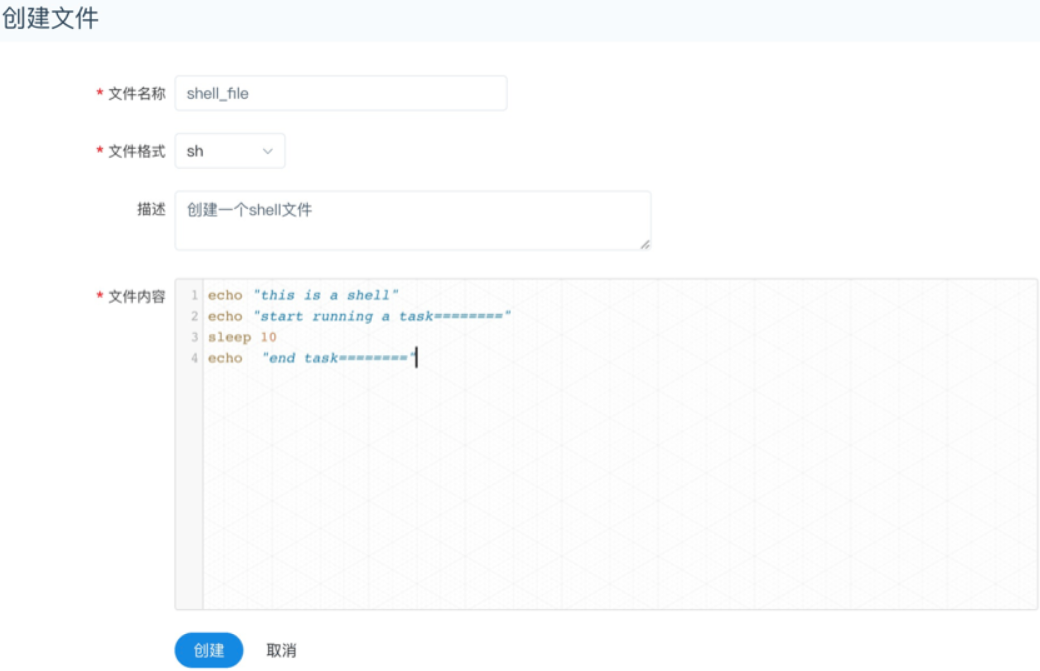
* yarn.resourcemanager.ha.rm.ids与yarn.application.status.address只需配置其中一个地址，另一个地址配置为空。
* 需要从Hadoop集群的conf目录下复制core-site.xml、hdfs-site.xml到dolphinscheduler项目的conf目录下，重启api-server服务。

##### 文件管理

* 对各种资源文件的管理，包括创建基本的txt/log/sh/conf/py/java等文件、上传jar包等各种类型文件，可进行编辑、重命名、下载、删除等操作。



* 创建文件
  + 文件格式支持以下几种类型：txt、log、sh、conf、cfg、py、java、sql、xml、hql、properties



* 上传文件
  + 上传文件：点击"上传文件"按钮进行上传，将文件拖拽到上传区域，文件名会自动以上传的文件名称补全



* 大文件上传的问题：（大文件不能上传）
  + 没有配置ds的nginx上传文件大小或者设置过小

vim /etc/nginx/nginx.conf

|  |
| --- |
| # 配置nginx上传文件最大限制  client\_max\_body\_size 300m; |



* 重启DS前端nginx服务

systemctl restart nginx

##### UDF管理

* 资源管理

资源管理和文件管理功能类似，不同之处是资源管理是上传的UDF函数，文件管理上传的是用户程序，脚本及配置文件 操作功能：重命名、下载、删除。

* 上传udf资源
  + 和上传文件相同
* 函数管理
  + 创建udf函数
    - 点击“创建UDF函数”，输入udf函数参数，选择udf资源，点击“提交”，创建udf函数。
    - 目前只支持HIVE的临时UDF函数
      * UDF函数名称：输入UDF函数时的名称
      * 包名类名：输入UDF函数的全路径
      * UDF资源：设置创建的UDF对应的资源文件

#### 数据源中心

* 创建数据源
  + 数据源中心支持Mysql、PostgrepSql、Hive/Impala、Spark、Clickhouse、Oracle、Sqlserver等数据源

##### 创建/编辑Mysql数据源

* 点击“数据源中心->创建数据源”，根据需求创建不同类型的数据源。
* 数据源：选择MYSQL
* 数据源名称：输入数据源的名称
* 描述：输入数据源的描述
* IP主机名：输入连接MySQL的IP
* 端口：输入连接MySQL的端口
* 用户名：设置连接MySQL的用户名
* 密码：设置连接MySQL的密码
* 数据库名：输入连接MySQL的数据库名称
* Jdbc连接参数：用于MySQL连接的参数设置，以JSON形式填写



点击“测试连接”，测试数据源是否可以连接成功。

##### 创建/编辑Hive数据源

* 数据源：选择HIVE
* 数据源名称：输入数据源的名称
* 描述：输入数据源的描述
* IP/主机名：输入连接HIVE的IP
* 端口：输入连接HIVE的端口
* 用户名：设置连接HIVE的用户名
* 密码：设置连接HIVE的密码
* 数据库名：输入连接HIVE的数据库名称
* Jdbc连接参数：用于HIVE连接的参数设置，以JSON形式填写

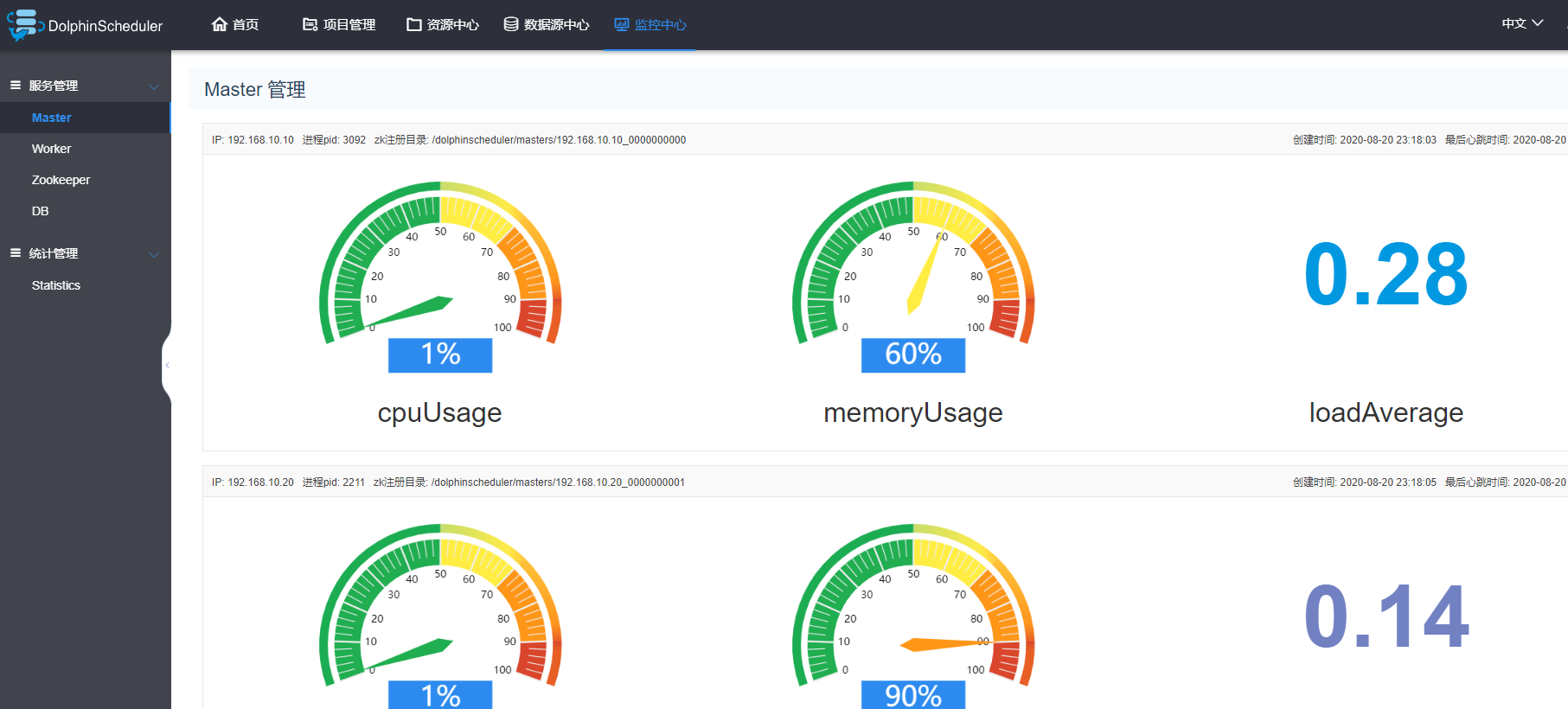


#### 监控中心

* 服务管理
  + 服务管理主要是对系统中的各个服务的健康状况和基本信息的监控和显示

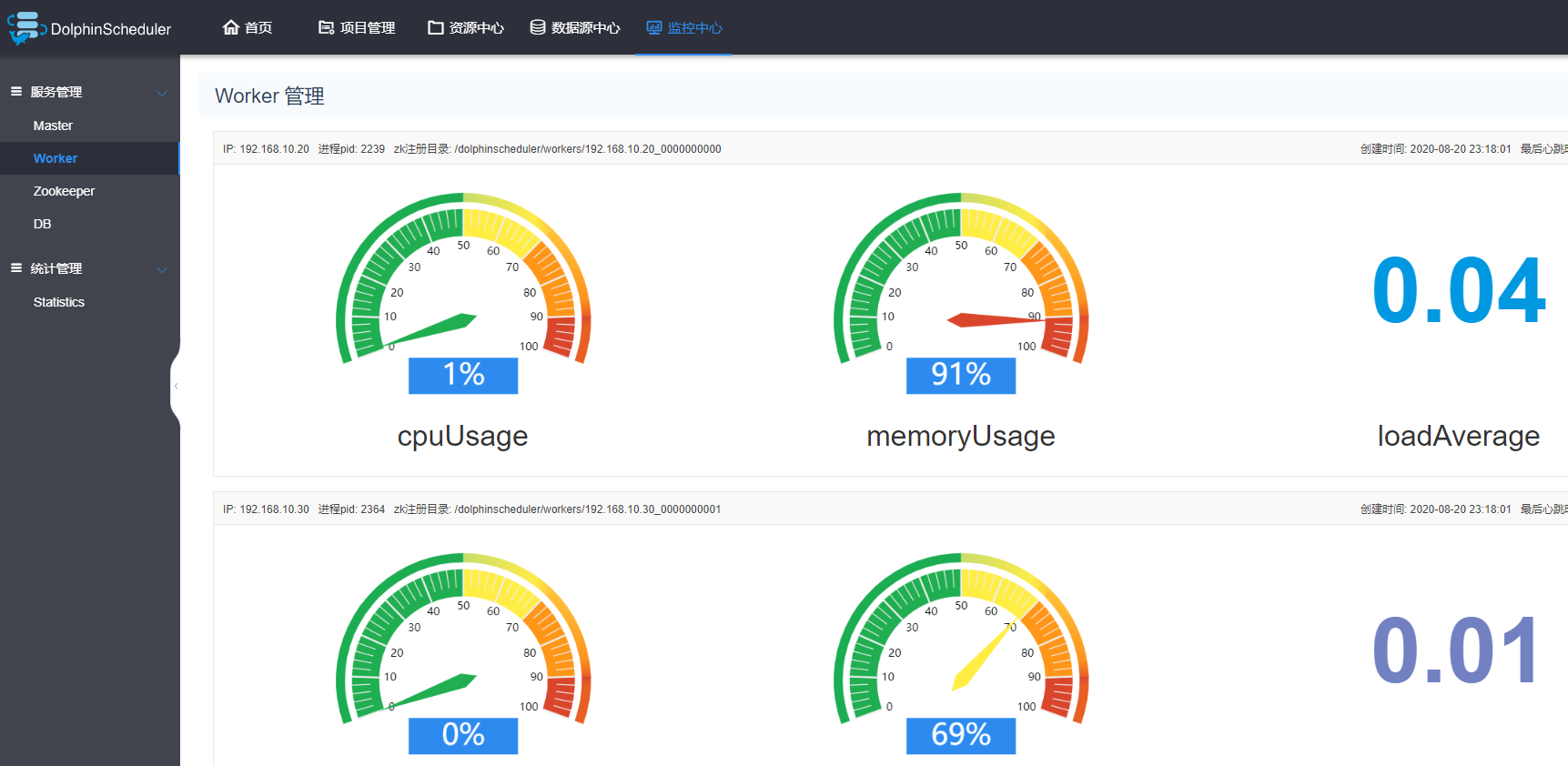
##### master监控

* master的相关信息(cpu、内存、平均负载率)



##### worker监控

* 主要是worker的相关信息(cpu、内存、平均负载率)



##### zookeeper监控

* 主要是zookpeeper中各个worker和master的相关配置信息。



##### db监控

* 主要是DB的健康状况（健康状态，最大连接数等）



##### 统计管理



* 待执行命令数：统计t\_ds\_command表的数据
* 执行失败的命令数：统计t\_ds\_error\_command表的数据
* 待运行任务数：统计Zookeeper中task\_queue的数据
* 待杀死任务数：统计Zookeeper中task\_kill的数据

#### 安全中心(权限控制系统)

|  |
| --- |
| \* 安全中心只有管理员账户才有权限操作，分别有队列管理、租户管理、用户管理、告警组管理、worker分组管理、令牌管理等功能，在用户管理模块可以对资源、数据源、项目等授权  \* 管理员登录，默认用户名密码：admin/dolphinscheduler123 |

* 创建队列
  + 队列是在执行spark、mapreduce等程序，需要用到“队列”参数时使用的。
  + 管理员进入安全中心->队列管理页面，点击“创建队列”按钮，创建队列。
* 添加租户
  + 租户对应的是Linux的用户，用于worker提交作业所使用的用户。如果linux没有这个用户，worker会在执行脚本的时候创建这个用户。
  + 租户编码：租户编码是Linux上的用户，唯一，不能重复
  + 管理员进入安全中心->租户管理页面，点击“创建租户”按钮，创建租户。
* 创建普通用户
  + 用户分为管理员用户和普通用户

|  |
| --- |
| \* 管理员有授权和用户管理等权限，没有创建项目和工作流定义的操作的权限。  \* 普通用户可以创建项目和对工作流定义的创建，编辑，执行等操作。  \* 注意：如果该用户切换了租户，则该用户所在租户下所有资源将复制到切换的新租户下。 |

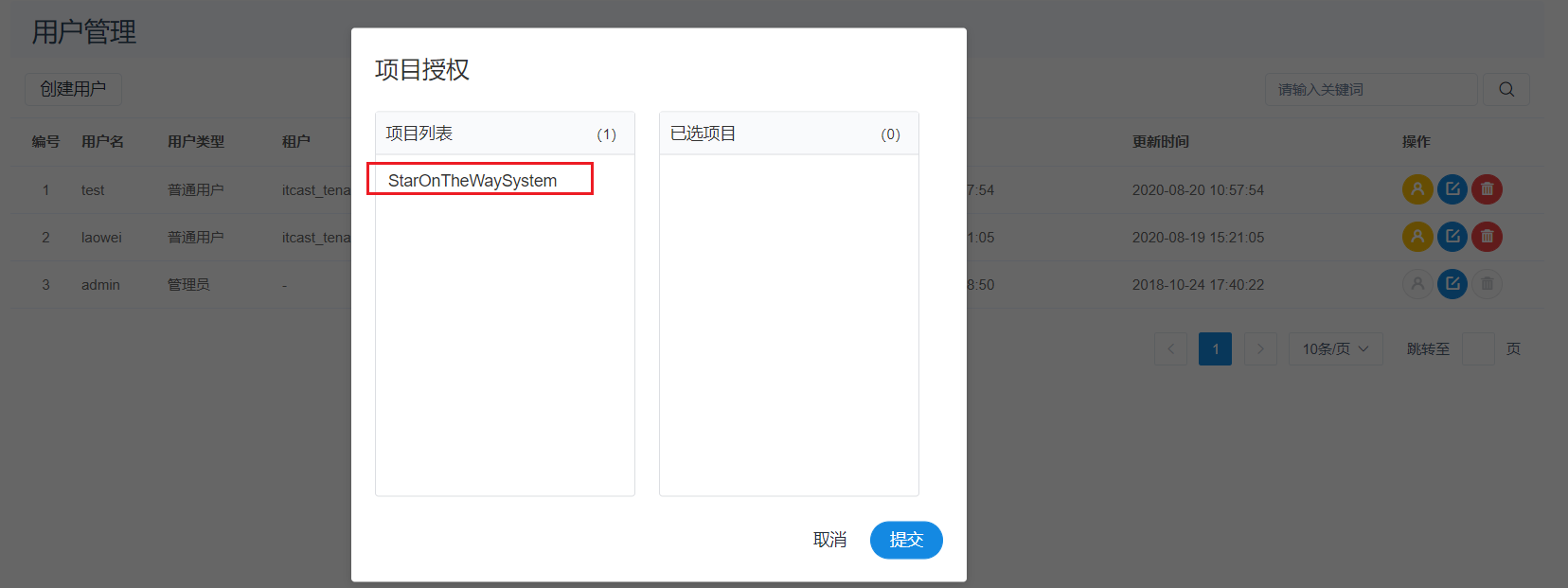
* 创建告警组
  + 告警组是在启动时设置的参数，在流程结束以后会将流程的状态和其他信息以邮件形式发送给告警组。
  + 管理员进入安全中心->告警组管理页面，点击“创建告警组”按钮，创建告警组。
* 创建worker分组
  + worker分组，提供了一种让任务在指定的worker上运行的机制。管理员创建worker分组，在任务节点和运行参数中设置中可以指定该任务运行的worker分组，如果指定的分组被删除或者没有指定分组，则该任务会在任一worker上运行。
  + 管理员进入安全中心->Worker分组管理页面，点击“创建Worker分组”按钮，创建Worker分组。worker分组内有多个ip地址（不能写别名），以英文逗号分隔。
* 令牌管理
  + 由于后端接口有登录检查，令牌管理提供了一种可以通过调用接口的方式对系统进行各种操作。
  + 管理员进入安全中心->令牌管理页面，点击“创建令牌”按钮，选择失效时间与用户，点击"生成令牌"按钮，点击"提交"按钮，则选择用户的token创建成功。
  + 普通用户登录后，点击用户名下拉框中的用户信息，进入令牌管理页面，选择失效时间，点击"生成令牌"按钮，点击"提交"按钮，则该用户创建token成功。
* 授予权限

|  |
| --- |
| \* 授予权限包括项目权限，资源权限，数据源权限，UDF函数权限。  \* 管理员可以对普通用户进行非其创建的项目、资源、数据源和UDF函数进行授权。因为项目、资源、数据源和UDF函数授权方式都是一样的，所以以项目授权为例介绍。  \* 注意：对于用户自己创建的项目，该用户拥有所有的权限。则项目列表和已选项目列表中不会显示。 |

* + 管理员进入安全中心->用户管理页面，点击需授权用户的“授权”按钮，如下图所示：



* 选择项目，进行项目授权。



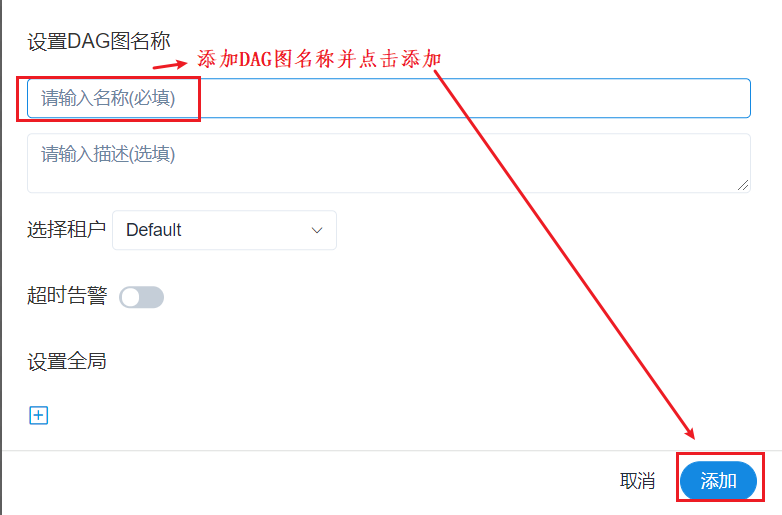
* + 资源、数据源、UDF函数授权同项目授权。

## DS定义并调度任务

### 原始数据实时ETL的shell任务

#### 定义shell任务



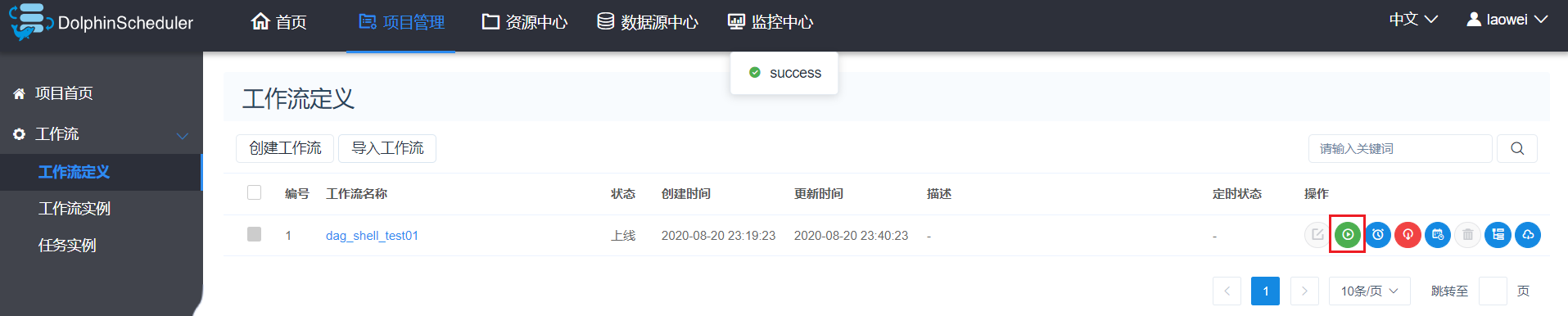


#### 运行shell任务

* 在工作流定义，设置任务上线



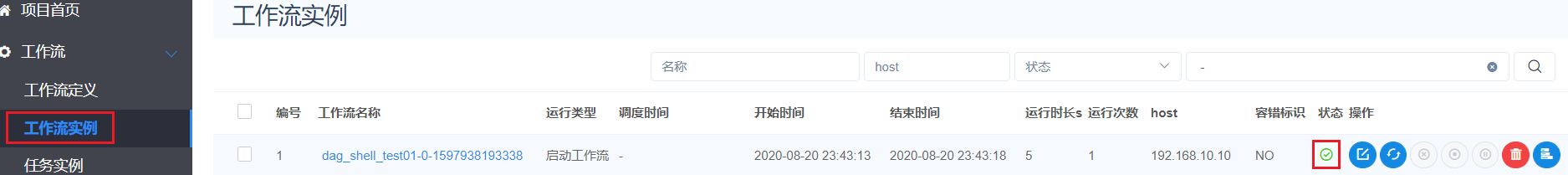
* 点击“运行”按钮



* 不填写提示信息，直接点击运行



* 在工作流实例，查看结果



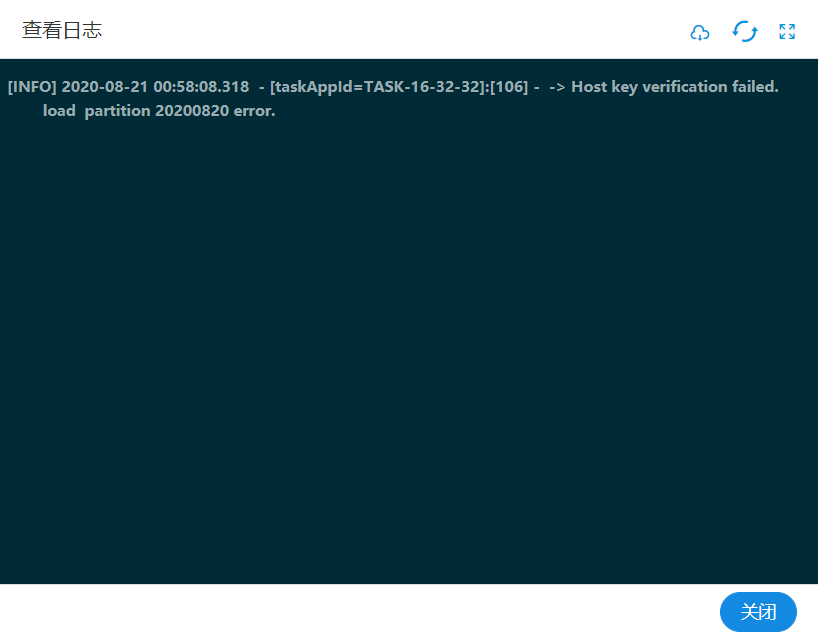
* 根据itcast\_error的hive表与hdfs数据关联任务，创建shell任务并执行

|  |
| --- |
| yesterday=`date --date=**'1 days ago'** +%Y%m%d`  tableName=**"itcast\_error"** ssh 192.168.10.10 **'/export/servers/apache-hive-2.1.1-bin/bin/hive -e "use itcast\_ods;alter table itcast\_error add partition (dt=\"$yesterday\") location \"/apps/hive/warehouse/ods.db/itcast\_error/$yesterday\" ;" ' if** [ $? -eq 0 ]; **then** echo **"load** $tableName **partition** $yesterday **succesful." else** echo **"load** $tableName **partition** $yesterday **error." fi** |

* 任务设置



* 任务日志



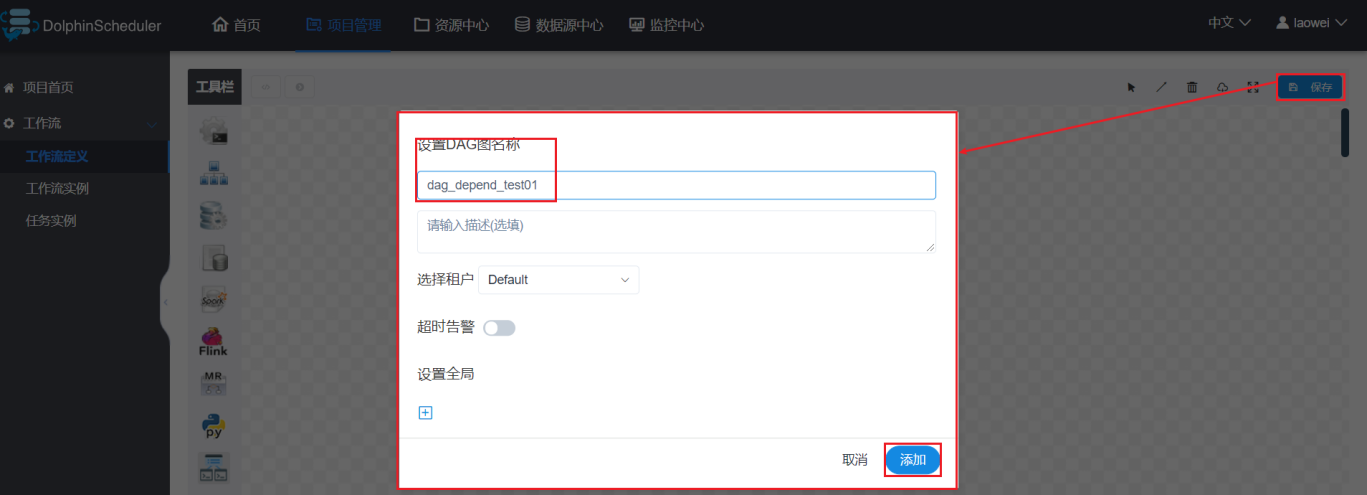
### 定义并运行依赖任务

#### 定义依赖任务

* 定义并设置依赖任务

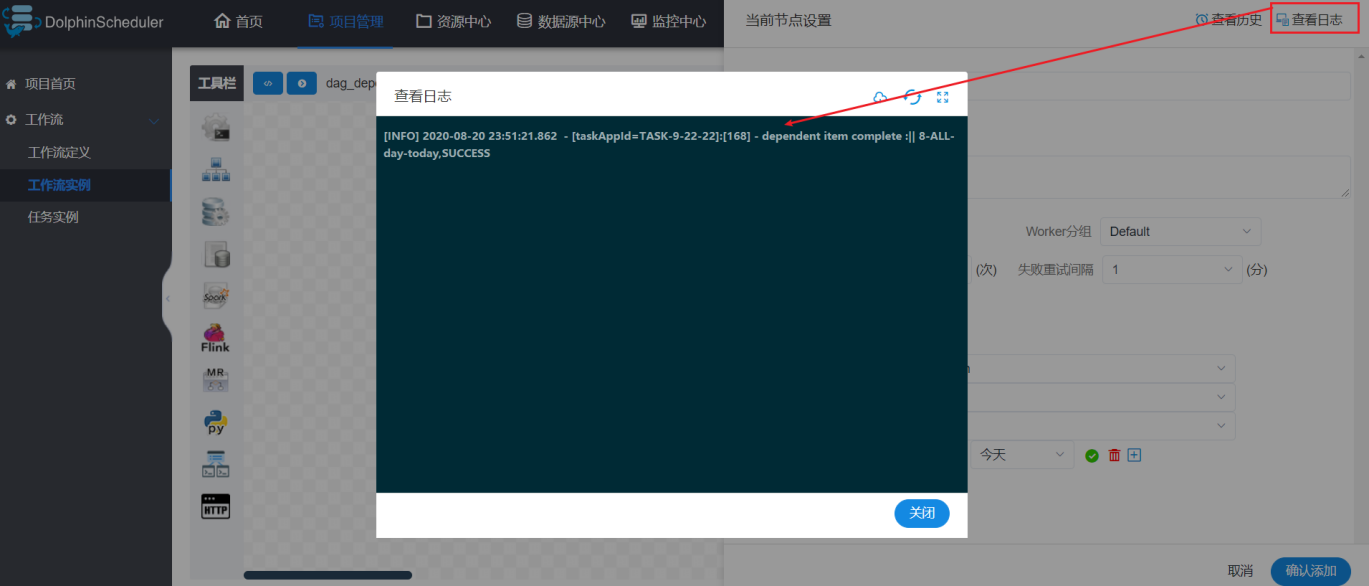


* 设置DAG依赖任务图名称并保存



#### 运行依赖任务

* 上线依赖任务
* 运行依赖任务
* 查看日志



* 依赖任务运行状态



### hive定时加载数据任务

#### 定义hive任务



#### hive任务(sql任务)参数含义

* 数据源：选择对应的数据源
* sql类型：支持查询和非查询两种，查询是select类型的查询，是有结果集返回的，可以指定邮件通知为表格、附件或表格附件三种模板。非查询是没有结果集返回的，是针对update、delete、insert三种类型的操作。
* sql参数：输入参数格式为key1=value1;key2=value2…
* sql语句：SQL语句
* UDF函数：对于HIVE类型的数据源，可以引用资源中心中创建的UDF函数,其他类型的数据源暂不支持UDF函数。
* 自定义参数：SQL任务类型，而存储过程是自定义参数顺序的给方法设置值自定义参数类型和数据类型同存储过程任务类型一样。区别在于SQL任务类型自定义参数会替换sql语句中${变量}。
* 前置sql:前置sql在sql语句之前执行。
* 后置sql:后置sql在sql语句之后执行。

#### 运行hive任务









### 原始数据实时ETL定时任务

#### 定义flink任务



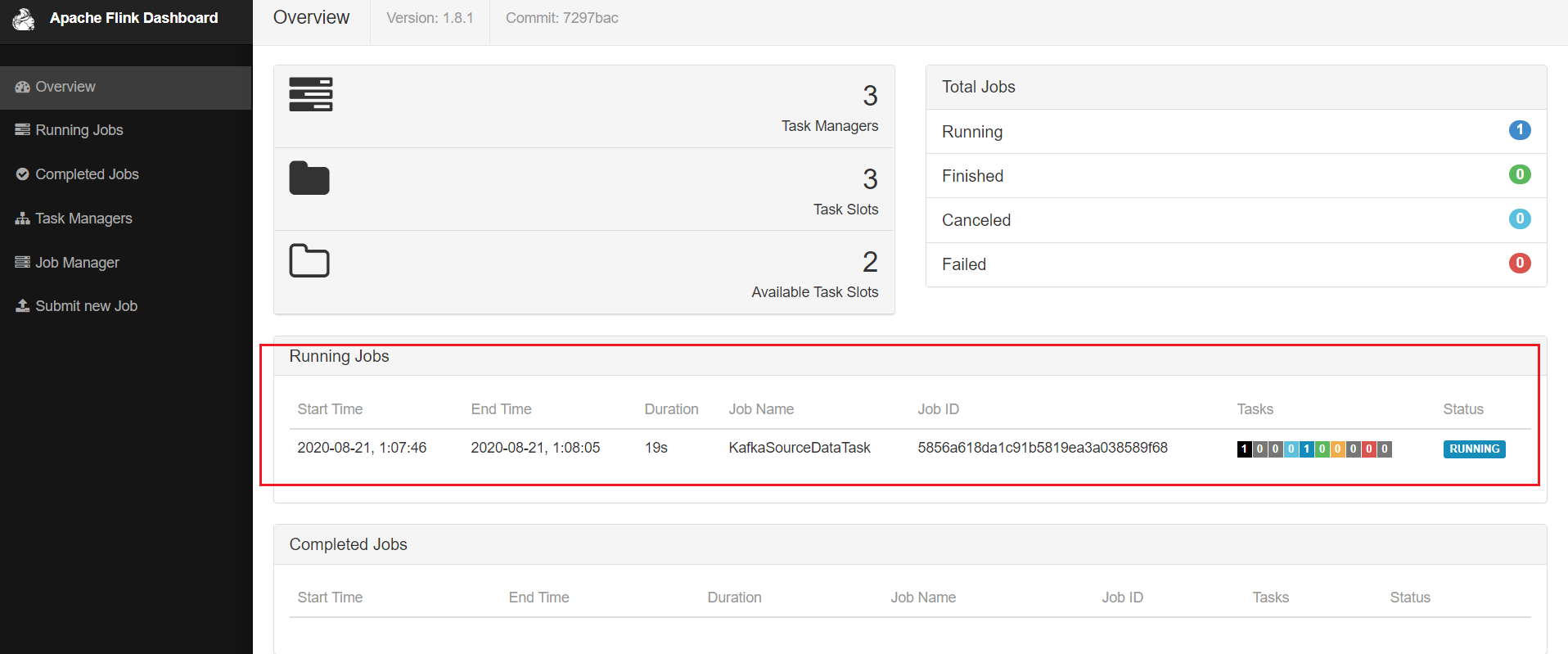
#### flink任务参数含义

* 程序类型：支持JAVA、Scala和Python三种语言
* 主函数的class：是Flink程序的入口Main Class的全路径
* 主jar包：是Flink的jar包
* 部署方式：支持cluster、local二种模式
* slot数量：可以设置slot数
* taskManage数量：可以设置taskManage数
* jobManager内存数：可以设置jobManager内存数
* taskManager内存数：可以设置taskManager内存数
* 命令行参数：是设置Spark程序的输入参数，支持自定义参数变量的替换。
* 其他参数：支持 --jars、--files、--archives、--conf格式
* 资源：如果其他参数中引用了资源文件，需要在资源中选择指定
* 自定义参数：是Flink局部的用户自定义参数，会替换脚本中以${变量}的内容

注意：JAVA和Scala只是用来标识，没有区别，如果是Python开发的Flink则没有主函数的class，其他都是一样

#### 运行flink任务





* 设置定时，每天凌晨3点运行，连续运行一周





* 定时管理



* 上线定时任务

