

尚硅谷大数据技术之 DolphinScheduler

(作者：尚硅谷研究院)

版本：V1.0

第 1 章 DolphinScheduler 简介

1.1 DolphinScheduler 概述

Apache DolphinScheduler 是一个分布式、易扩展的可视化 DAG 工作流任务调度平台。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。

1.2 DolphinScheduler 核心架构

DolphinScheduler 的主要角色如下：

MasterServer 采用分布式无中心设计理念，MasterServer 主要负责 DAG 任务切分、任务提交、任务监控，并同时监听其它 MasterServer 和 WorkerServer 的健康状态。

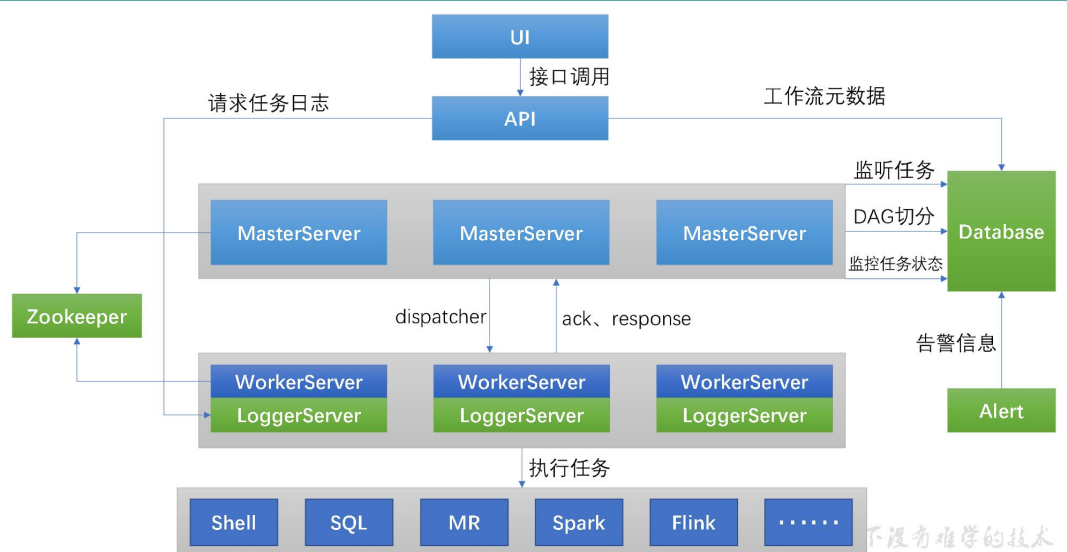
WorkerServer 也采用分布式无中心设计理念，WorkerServer 主要负责任务的执行和提供日志服务。

ZooKeeper 服务，系统中的 MasterServer 和 WorkerServer 节点都通过 ZooKeeper 来进行集群管理和容错。

Alert 服务，提供告警相关服务。

API 接口层，主要负责处理前端 UI 层的请求。

UI，系统的前端页面，提供系统的各种可视化操作界面。



第 2 章 DolphinScheduler 部署说明

2.1 软硬件环境要求

2.1.1 操作系统版本要求

操作系统	版本
Red Hat Enterprise Linux	7.0 及以上
CentOS	7.0 及以上
Oracle Enterprise Linux	7.0 及以上
Ubuntu LTS	16.04 及以上

2.1.2 服务器硬件要求

CPU	内存	网络
4 核+	8 GB+	千兆网卡

2.2 部署模式

DolphinScheduler 支持多种部署模式，包括单机模式（Standalone）、伪集群模式（Pseudo-Cluster）、集群模式（Cluster）等。

2.2.1 单机模式

单机模式（standalone）模式下，所有服务均集中于一个 StandaloneServer 进程中，并且其中内置了注册中心 Zookeeper 和数据库 H2。只需配置 JDK 环境，就可一键启动更多 [Java](#) - 大数据 - 前端 - python 人工智能资料下载，可百度访问：[尚硅谷官网](#)

DolphinScheduler，快速体验其功能。

2.2.2 伪集群模式

伪集群模式（Pseudo-Cluster）是在单台机器部署 DolphinScheduler 各项服务，该模式下 master、worker、api server、logger server 等服务都只在同一台机器上。Zookeeper 和数据库需单独安装并进行相应配置。

2.2.3 集群模式

集群模式（Cluster）与伪集群模式的区别就是在多台机器部署 DolphinScheduler 各项服务，并且可以配置多个 Master 及多个 Worker。

第 3 章 DolphinScheduler 集群模式部署

3.1 集群规划

集群模式下，可配置多个 Master 及多个 Worker。通常可配置 2~3 个 Master，若干个 Worker。由于集群资源有限，此处配置一个 Master，三个 Worker，集群规划如下。

hadoop102	master、worker
hadoop103	worker
hadoop104	worker

3.2 前置准备工作

- 1) 三台节点均需部署 JDK（1.8+），并配置相关环境变量。
- 2) 需部署数据库，支持 MySQL（5.7+）或者 PostgreSQL（8.2.15+）。
- 3) 需部署 Zookeeper（3.4.6+）。
- 4) 三台节点均需安装进程管理工具包 psmisc。

```
[atguigu@hadoop102 ~]$ sudo yum install -y psmisc
[atguigu@hadoop103 ~]$ sudo yum install -y psmisc
[atguigu@hadoop104 ~]$ sudo yum install -y psmisc
```

3.3 解压 DolphinScheduler 安装包

- 1) 上传 DolphinScheduler 安装包到 hadoop102 节点的 `/opt/software` 目录
- 2) 解压安装包到当前目录

注：解压目录并非最终的安装目录

```
[atguigu@hadoop102 software]$ tar -zxvf apache-dolphinscheduler-1.3.9-bin.tar.gz
```

3.4 初始化数据库

DolphinScheduler 元数据存储的关系型数据库中，故需创建相应的数据库和用户。

1) 创建数据库

```
mysql> CREATE DATABASE dolphinscheduler DEFAULT CHARACTER SET utf8
DEFAULT COLLATE utf8_general_ci;
```

2) 创建用户

```
mysql> CREATE USER 'dolphinscheduler'@'%' IDENTIFIED BY
'dolphinscheduler';
```

注：

若出现以下错误信息，表明新建用户的密码过于简单。

```
ERROR 1819 (HY000): Your password does not satisfy the current policy
requirements
```

可提高密码复杂度或者执行以下命令降低 MySQL 密码强度级别。

```
mysql> set global validate_password_length=4;
mysql> set global validate_password_policy=0;
```

3) 赋予用户相应权限

```
mysql> GRANT ALL PRIVILEGES ON dolphinscheduler.* TO
'dolphinscheduler'@'%;
mysql> flush privileges;
```

4) 修改数据源配置文件

进入 DolphinScheduler 解压目录

```
[atguigu@hadoop102 apache-dolphinscheduler-1.3.9-bin]$ cd
/opt/software/apache-dolphinscheduler-1.3.9-bin/
```

修改 conf 目录下的 datasource.properties 文件

```
[atguigu@hadoop102 apache-dolphinscheduler-1.3.9-bin]$ vim
conf/datasource.properties
```

修改内容如下

```
spring.datasource.driver-class-name=com.mysql.jdbc.Driver
spring.datasource.url=jdbc:mysql://hadoop102:3306/dolphinschedule
r?useUnicode=true&characterEncoding=UTF-8
spring.datasource.username=dolphinscheduler
spring.datasource.password=dolphinscheduler
```

5) 拷贝 MySQL 驱动到 DolphinScheduler 的解压目录下的 lib 中

```
[atguigu@hadoop102 apache-dolphinscheduler-1.3.9-bin]$ cp
/opt/software/mysql-connector-java-5.1.27-bin.jar lib/
```

6) 执行数据库初始化脚本

数据库初始化脚本位于 DolphinScheduler 解压目录下的 script 目录中，即 /opt/software/ds/apache-dolphinscheduler-1.3.9-bin/script/。

```
[atguigu@hadoop102
apache-dolphinscheduler-1.3.9-bin]$ script/create-dolphinscheduler.
sh
```

3.5 配置一键部署脚本

修改解压目录下的 conf/config 目录下的 install_config.conf 文件

```
[atguigu@hadoop102 apache-dolphinscheduler-1.3.9-bin]$ vim conf/config/install_config.conf
```

修改内容如下

```
# postgresql or mysql
dbtype="mysql"

# db config
# db address and port
dbhost="hadoop102:3306"

# db username
username="dolphinscheduler"

# database name
dbname="dolphinscheduler"

# db passwd
# NOTICE: if there are special characters, please use the \ to escape,
# for example, `[` escape to `\[`
password="dolphinscheduler"

# zk cluster
zkQuorum="hadoop102:2181,hadoop103:2181,hadoop104:2181"

# Note: the target installation path for dolphinscheduler, please not
# config as the same as the current path (pwd)
installPath="/opt/module/dolphinscheduler"

# deployment user
# Note: the deployment user needs to have sudo privileges and
# permissions to operate hdfs. If hdfs is enabled, the root directory
# needs to be created by itself
deployUser="atguigu"

# resource storage type: HDFS, S3, NONE
resourceStorageType="HDFS"

# resource store on HDFS/S3 path, resource file will store to this
# hadoop hdfs path, self configuration, please make sure the directory
# exists on hdfs and have read write permissions. "/dolphinscheduler"
# is recommended
resourceUploadPath="/dolphinscheduler"

# if resourceStorageType is HDFS, defaultFS write namenode address,
# HA you need to put core-site.xml and hdfs-site.xml in the conf
# directory.
# if S3, write S3 address, HA, for example : s3a://dolphinscheduler,
# Note, s3 be sure to create the root directory /dolphinscheduler
defaultFS="hdfs://hadoop102:8020"

# resource manager port, the default value is 8088 if not specified
resourceManagerHttpAddressPort="8088"

# if resource manager HA is enabled, please set the HA IPs; if
```

```
resourceManager is single, keep this value empty
yarnHaIps=

# if resourceManager HA is enabled or not use resourceManager, please
keep the default value; If resourceManager is single, you only need
to replace ds1 to actual resourceManager hostname
singleYarnIp="hadoop103"

# who have permissions to create directory under HDFS/S3 root path
# Note: if kerberos is enabled, please config hdfsRootUser=
hdfsRootUser="atguigu"

# api server port
apiServerPort="12345"

# install hosts
# Note: install the scheduled hostname list. If it is
pseudo-distributed, just write a pseudo-distributed hostname
ips="hadoop102,hadoop103,hadoop104"

# ssh port, default 22
# Note: if ssh port is not default, modify here
sshPort="22"

# run master machine
# Note: list of hosts hostname for deploying master
masters="hadoop102"

# run worker machine
# note: need to write the worker group name of each worker, the default
value is "default"
workers="hadoop102:default,hadoop103:default,hadoop104:default"

# run alert machine
# note: list of machine hostnames for deploying alert server
alertServer="hadoop102"

# run api machine
# note: list of machine hostnames for deploying api server
apiServers="hadoop102"
```

3.6 一键部署 DolphinScheduler

1) 启动 Zookeeper 集群

```
[atguigu@hadoop102 apache-dolphinscheduler-1.3.9-bin]$ zk.sh start
```

2) 一键部署并启动 DolphinScheduler

```
[atguigu@hadoop102
apache-dolphinscheduler-1.3.9-bin]$ ./install.sh
```

3) 查看 DolphinScheduler 进程

```
----- hadoop102 -----
29139 ApiApplicationServer
28963 WorkerServer
3332 QuorumPeerMain
2100 DataNode
28902 MasterServer
29081 AlertServer
1978 NameNode
```

```
29018 LoggerServer
2493 NodeManager
29551 Jps
----- hadoop103 -----
29568 Jps
29315 WorkerServer
2149 NodeManager
1977 ResourceManager
2969 QuorumPeerMain
29372 LoggerServer
1903 DataNode
----- hadoop104 -----
1905 SecondaryNameNode
27074 WorkerServer
2050 NodeManager
2630 QuorumPeerMain
1817 DataNode
27354 Jps
27133 LoggerServer
```

4) 访问 DolphinScheduler UI

DolphinScheduler UI 地址为 <http://hadoop102:12345/dolphinscheduler>

初始用户的用户名为: admin, 密码为 dolphinscheduler123

3.7 DolphinScheduler 启停命令

DolphinScheduler 的启停脚本均位于其安装目录的 bin 目录下。

1) 一键启停所有服务

```
./bin/start-all.sh
./bin/stop-all.sh
```

注意同 Hadoop 的启停脚本进行区分。

2) 启停 Master

```
./bin/dolphinscheduler-daemon.sh start master-server
./bin/dolphinscheduler-daemon.sh stop master-server
```

3) 启停 Worker

```
./bin/dolphinscheduler-daemon.sh start worker-server
./bin/dolphinscheduler-daemon.sh stop worker-server
```

4) 启停 Api

```
./bin/dolphinscheduler-daemon.sh start api-server
./bin/dolphinscheduler-daemon.sh stop api-server
```

5) 启停 Logger

```
./bin/dolphinscheduler-daemon.sh start logger-server
./bin/dolphinscheduler-daemon.sh stop logger-server
```

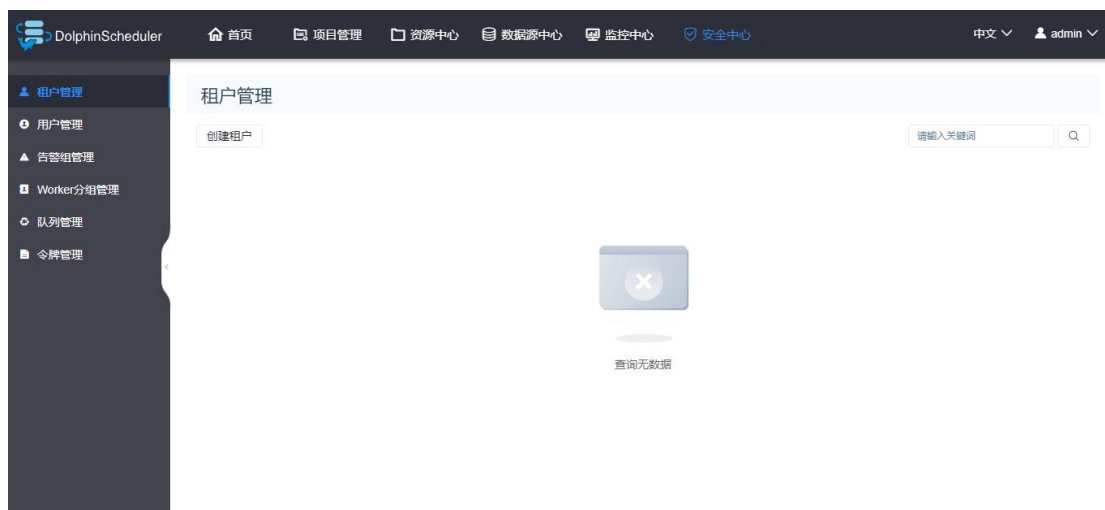
6) 启停 Alert

```
./bin/dolphinscheduler-daemon.sh start alert-server
./bin/dolphinscheduler-daemon.sh stop alert-server
```

第 4 章 DolphinScheduler 入门

4.1 安全中心配置

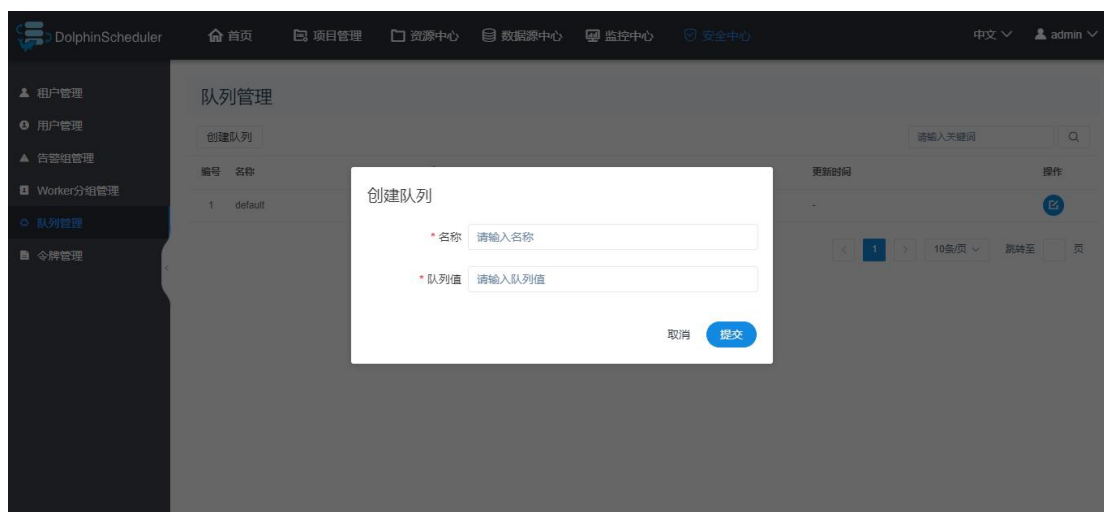
安全中心主要有租户管理、用户管理、告警组管理、Worker 分组管理、队列管理、令牌管理等功能。安全中心只有管理员账户才有操作权限。



4.1.1 队列管理

此处的队列对应的是 Yarn 调度器的资源队列。故队列概念只对跑在 Yarn 上的任务类型有效。此处创建出的队列，可供后续任务进行选择。需要注意的是，在 DolphinScheduler 中创建队列，并不会影响到 Yarn 调度器的队列配置。

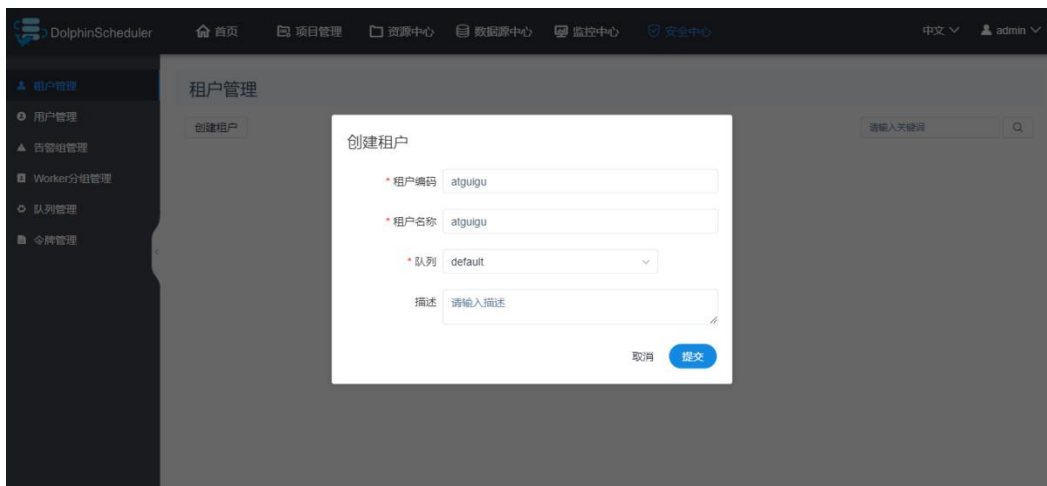
此处可不创建队列。



4.1.2 租户管理

租户对应的是 Linux 系统用户，是 Worker 执行任务使用的用户。如果 Worker 所在节点没有这个用户，Worker 会在执行任务时创建这个用户。

此处创建一个 atguigu 租户，如下图。



注：

租户编码：对应 Worker 执行任务所使用的用户名。

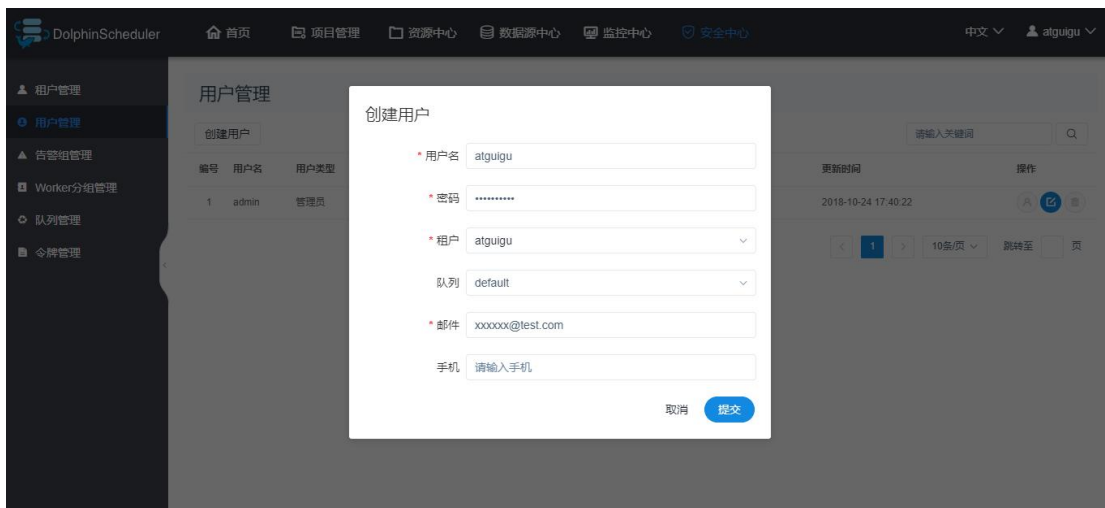
租户名称：用于在 DolphinScheduler 中显示。

队列：该租户提交 Yarn 任务时的默认队列。

4.1.3 用户管理

用户对应的是 DolphinScheduler 的用户，用于登录 DolphinScheduler。用户分管理员用户和普通用户。默认情况下，管理员只有授权和用户管理等权限，而普通用户只有创建项目，定义工作流、执行工作流等权限。

此处创建一个普通用户 atguigu，如下图。



注：

用户名：DolphinScheduler 登录账户

租户：该用户关联的租户

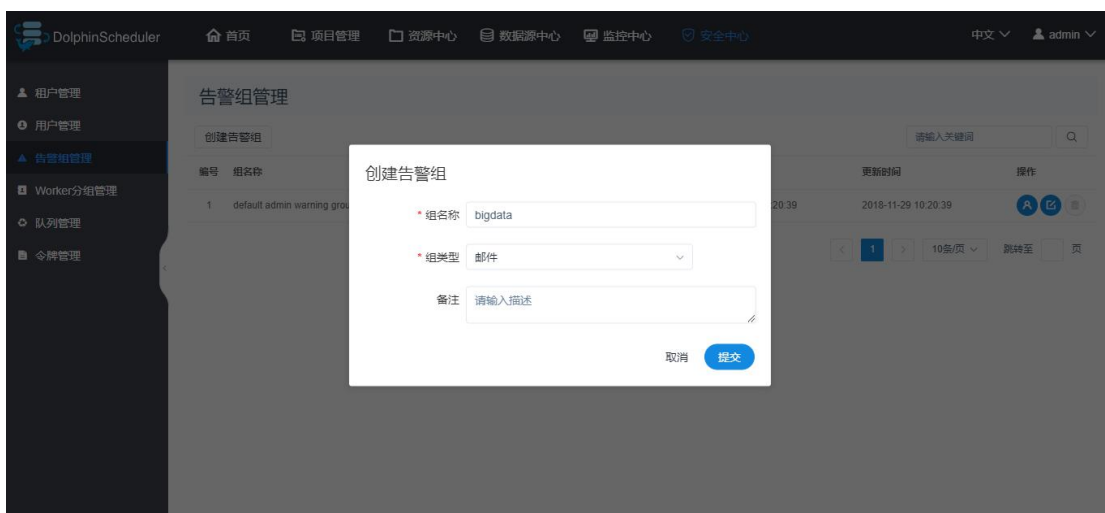
队列：默认为租户所关联的队列。

邮件、手机号：主要用于告警通知。

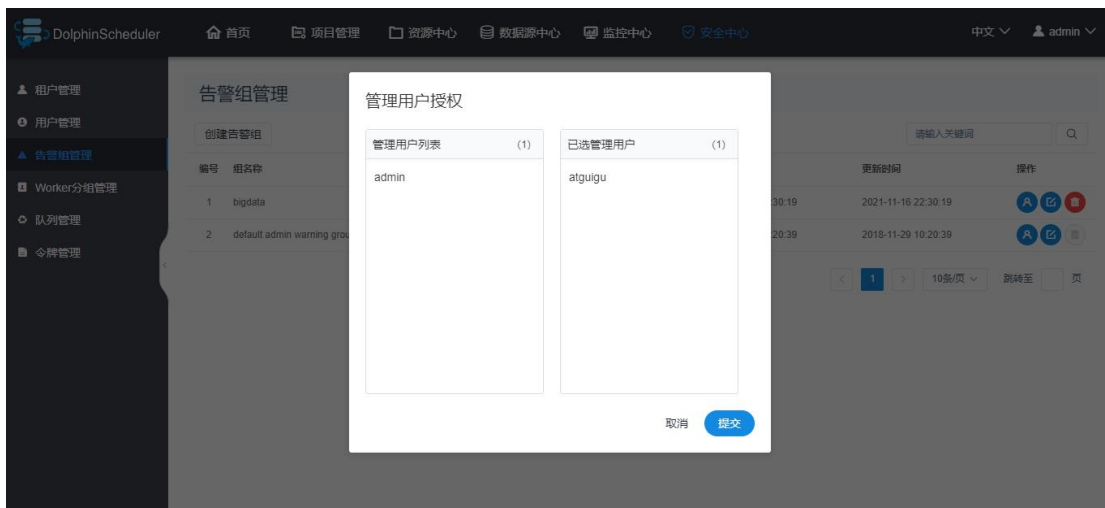
4.1.4 告警组管理

告警组可包含多名用户，用于指定告警发送对象。

1) 创建告警组



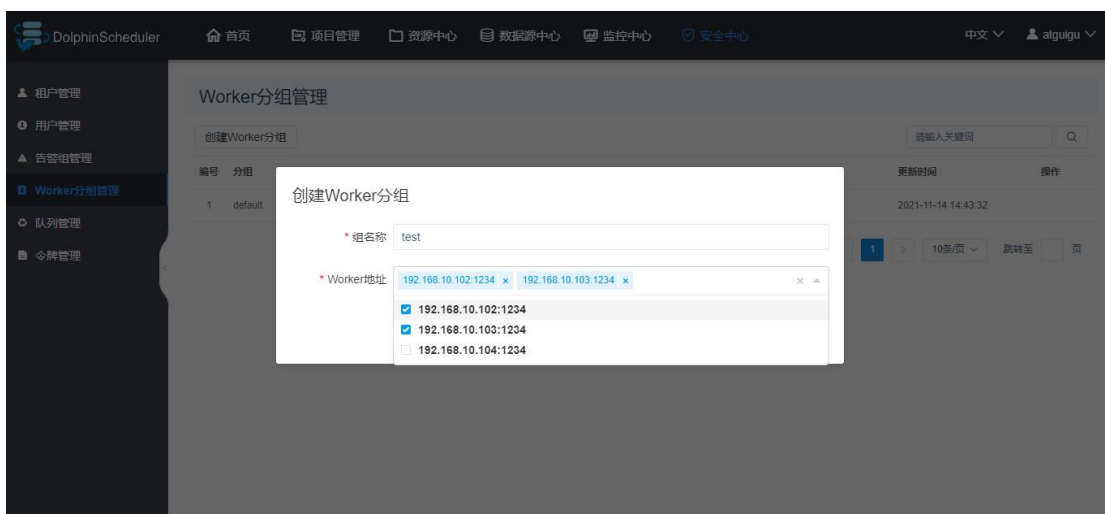
2) 增加用户



4.1.5 Worker 分组管理

在任务执行时，可以将任务分配给指定 Worker 组，最终由该组中的 Worker 节点执行该任务。默认情况下，所有 Worker 均位于 default 组。

此处可不作配置。



4.1.6 令牌管理

令牌用于通过接口访问 DolphinScheduler 各项服务时的用户验证。普通用户通过 UI 页面访问各项服务时，无需使用令牌。若需将 DolphinScheduler 与第三方服务进行集成，则需调用其接口，此时需使用令牌。

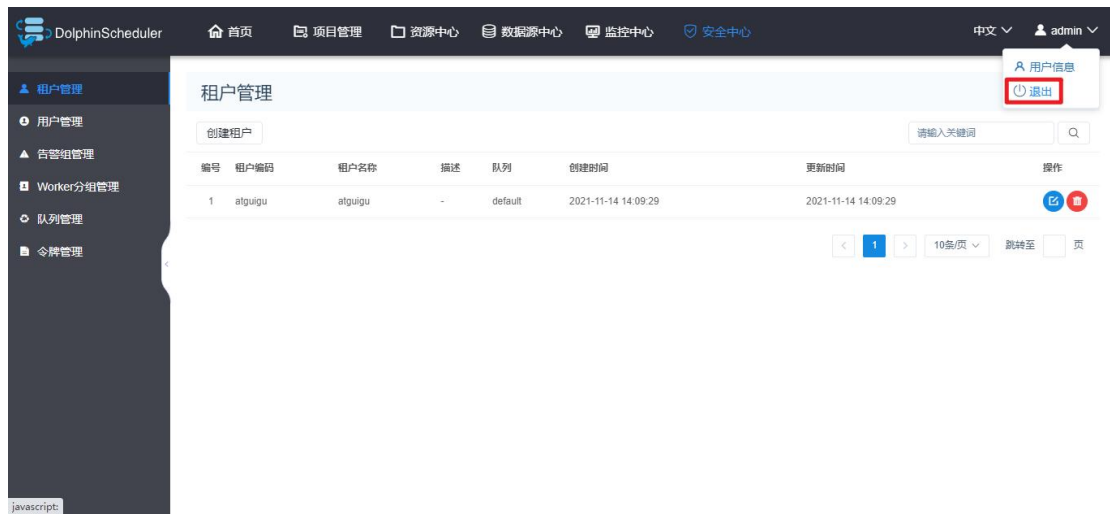
4.2 项目管理

4.2.1 切换用户

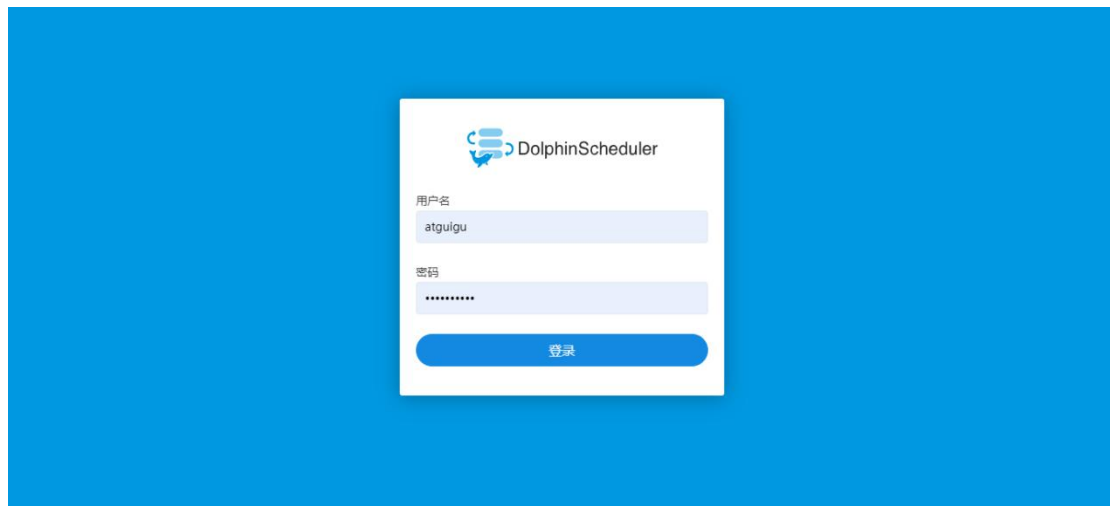
默认不使用管理员用户操作项目和工作流等，故需先切换到普通用户 atguigu。

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)

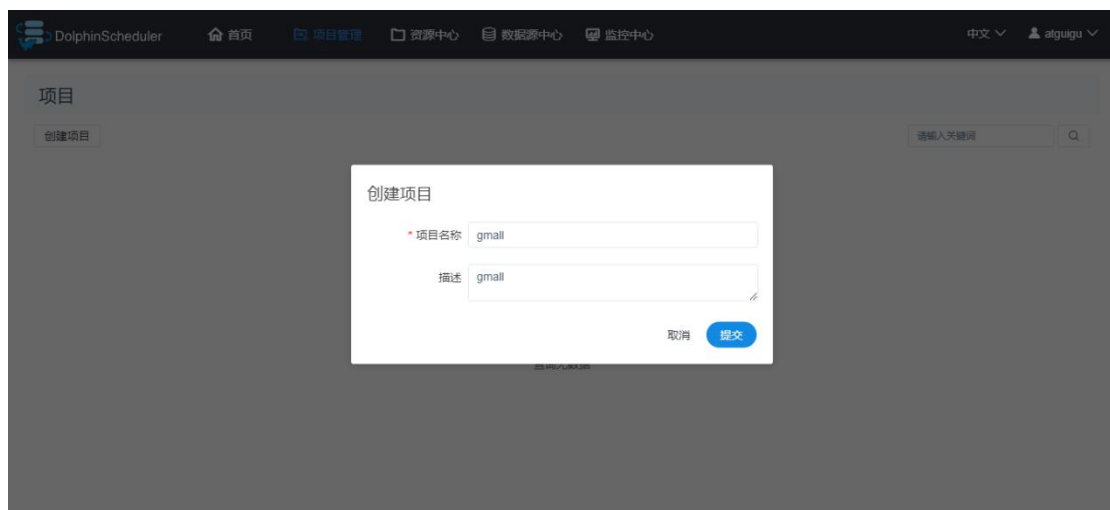
1) admin 用户退出



2) 使用 atguigu 用户登录



4.2.2 创建项目



4.2.3 查看项目



编号	项目名称	所属用户	工作流定义数	正在运行的流程数	描述	创建时间	更新时间	操作
1	gmail	atguigu	0	0	gmail	2021-11-14 16:13:48	2021-11-14 16:13:48	 

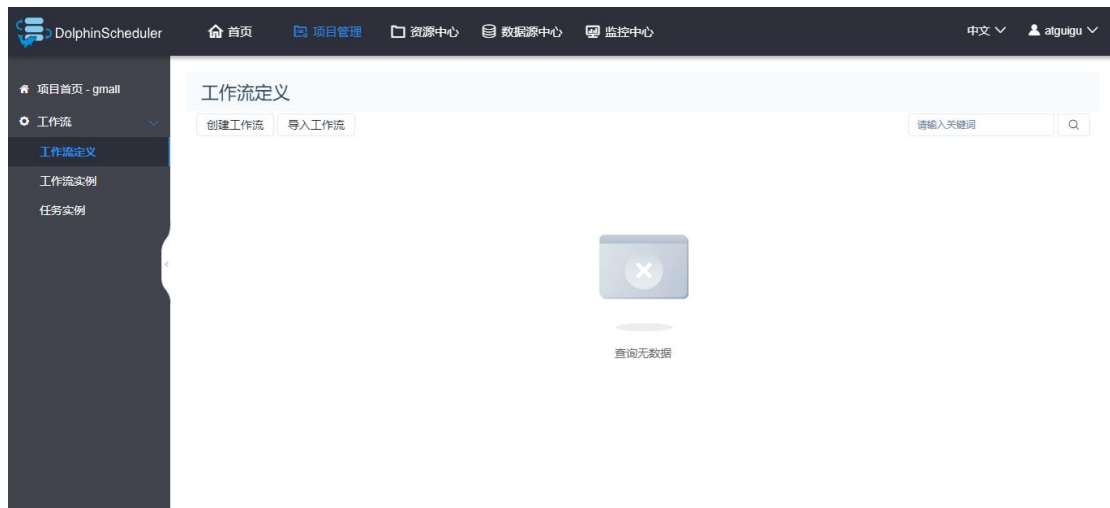
4.3 workflow 基础配置


下图为 workflow 配置页面，共包含三个模块，分别为 workflow 定义、workflow 实例和任务实例。

workflow 定义：用于定义 workflow，包括 workflow 各节点任务详情及各节点依赖关系等。

workflow 实例：workflow 每执行一次就会生成一个 workflow 实例。此处可查看正在运行的 workflow 以及已经完成的工作流。

任务实例：workflow 中的一个节点任务，每执行一次就会生成一个任务实例。此处可用于查看正在执行的节点任务以及已经完成的节点任务。

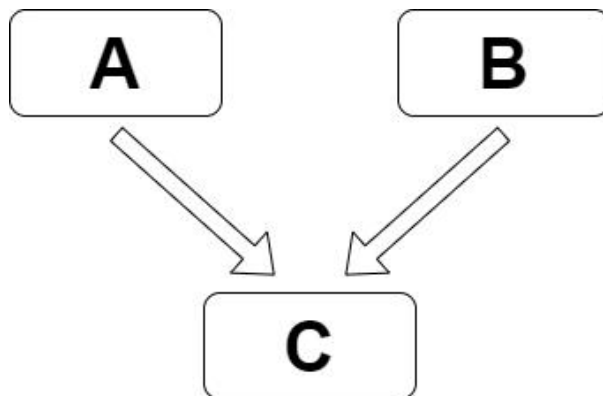


workflow 定义	
创建工作流	导入工作流
<div>  查询无数据 </div>	

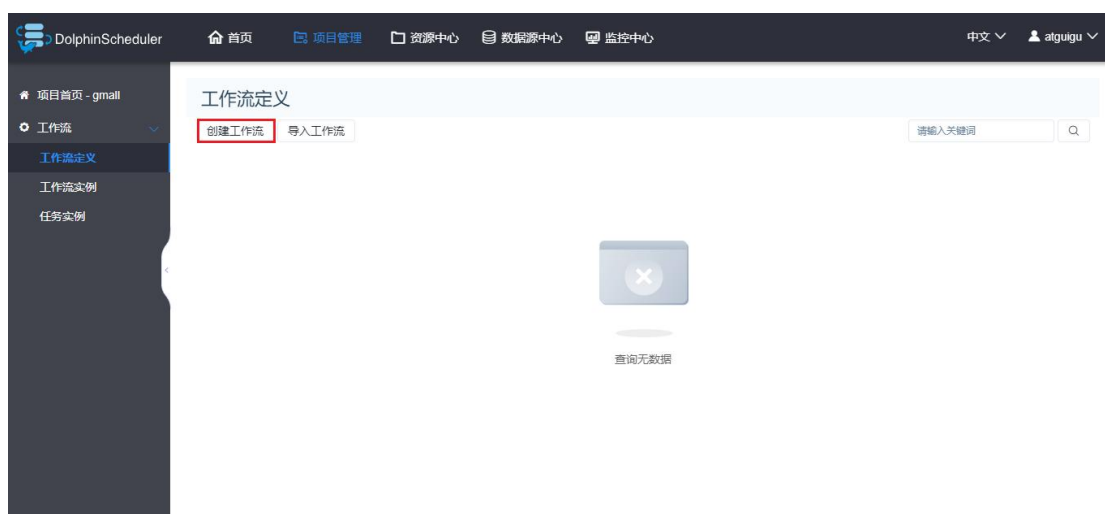
4.3.1 workflow 定义

workflow 要求：workflow 需包含三个 Shell 类型的任务节点，分别是 A，B，C。三个任务的依赖关系如下图所示：

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：[尚硅谷官网](#)



1) 创建工作流



2) 配置任务节点

(1) 节点 A



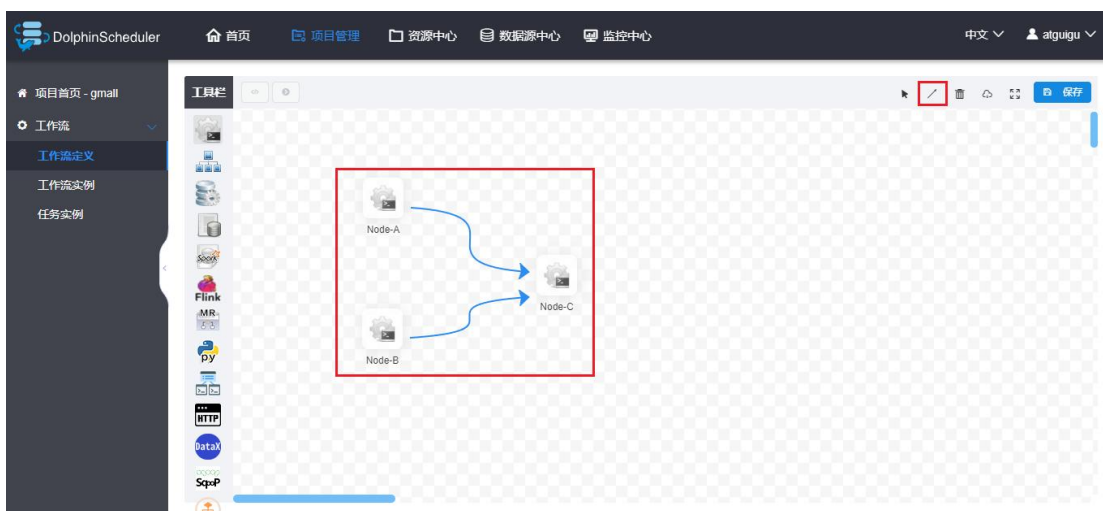
(2) 节点 B



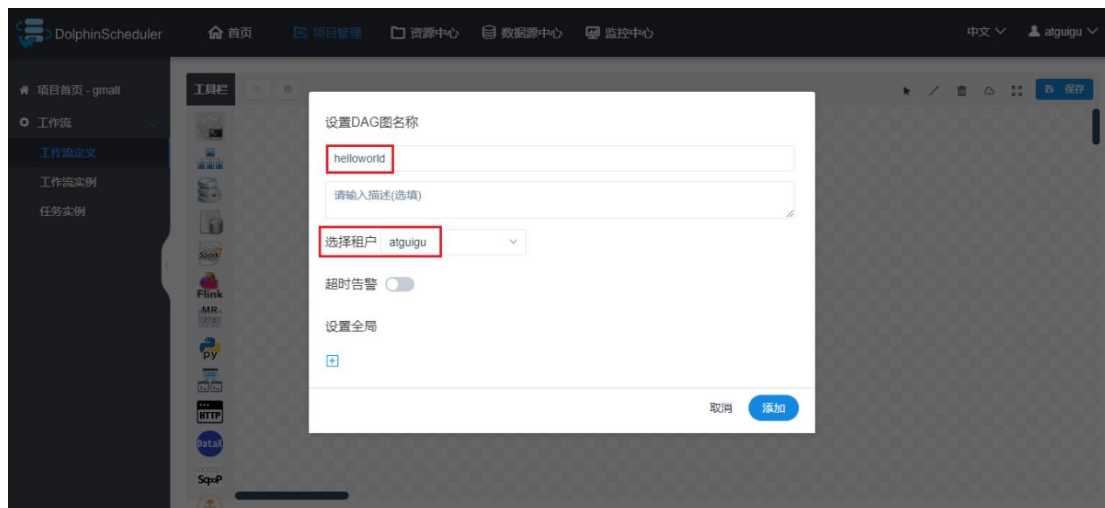
(3) 节点 C



3) 配置任务节点的依赖关系



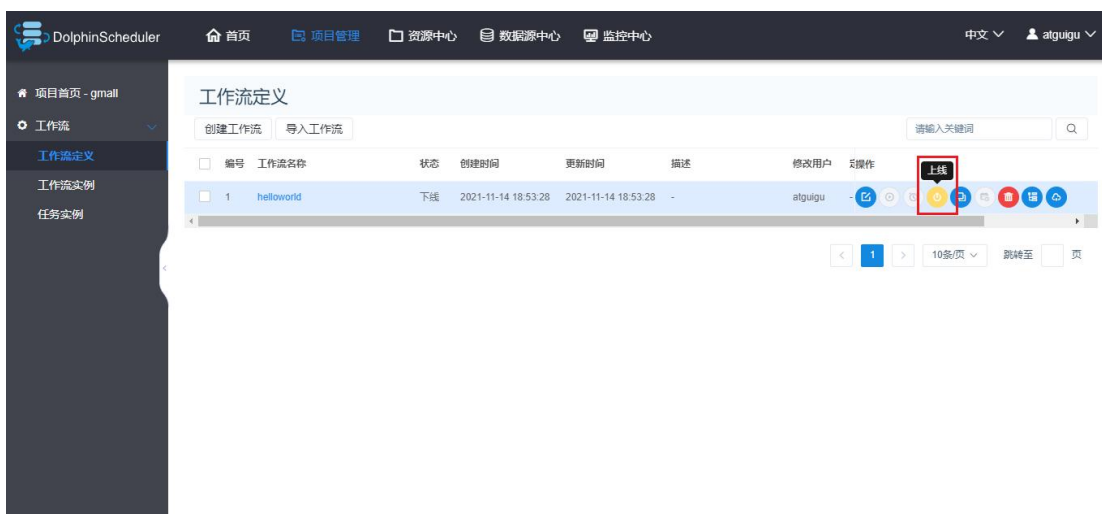
4) 保存 workflow 定义



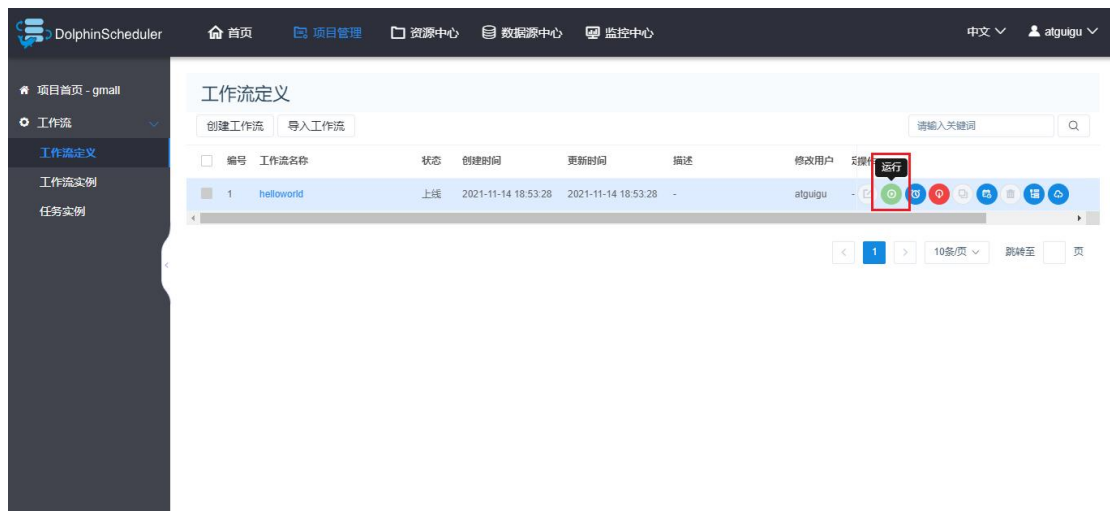
4.3.2 提交执行 workflow

1) 上线 workflow

workflow 须上线之后才能执行。处于上线状态的 workflow 定义不可修改，如需修改，须先下线。

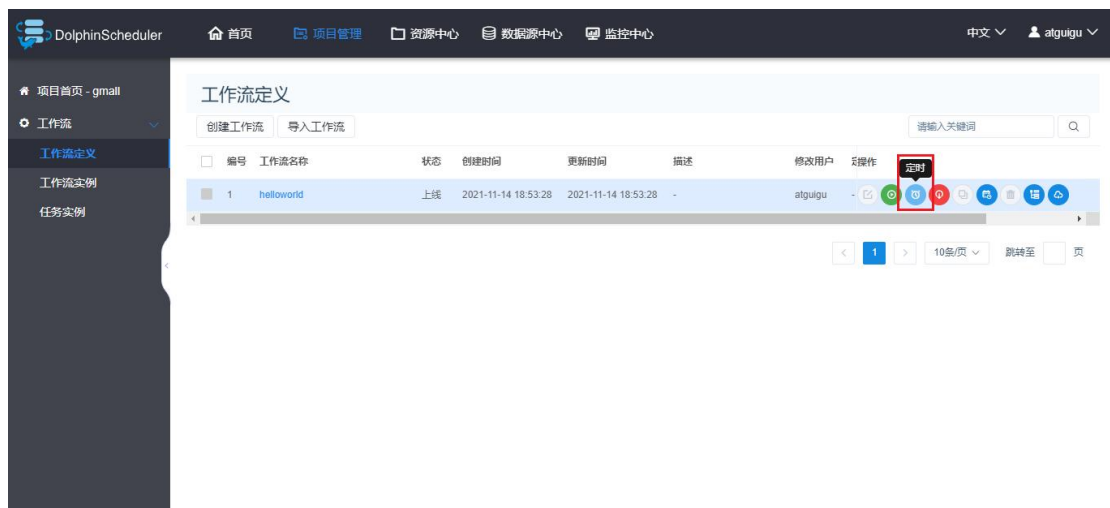


2) 单次运行 workflow



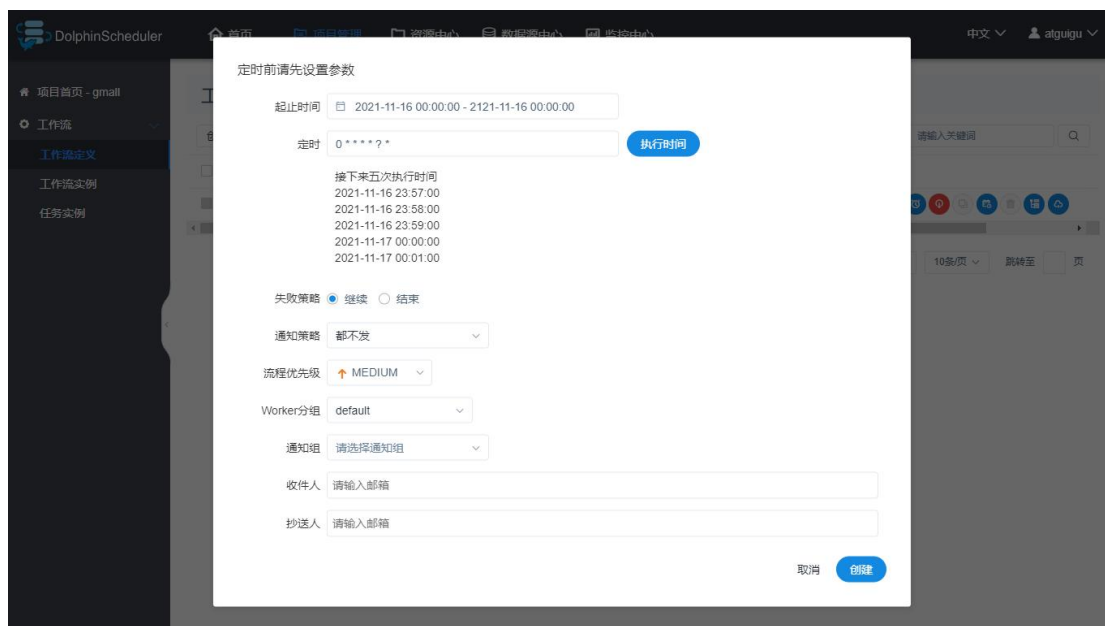
3) 定时执行工作流

(1) 点击定时

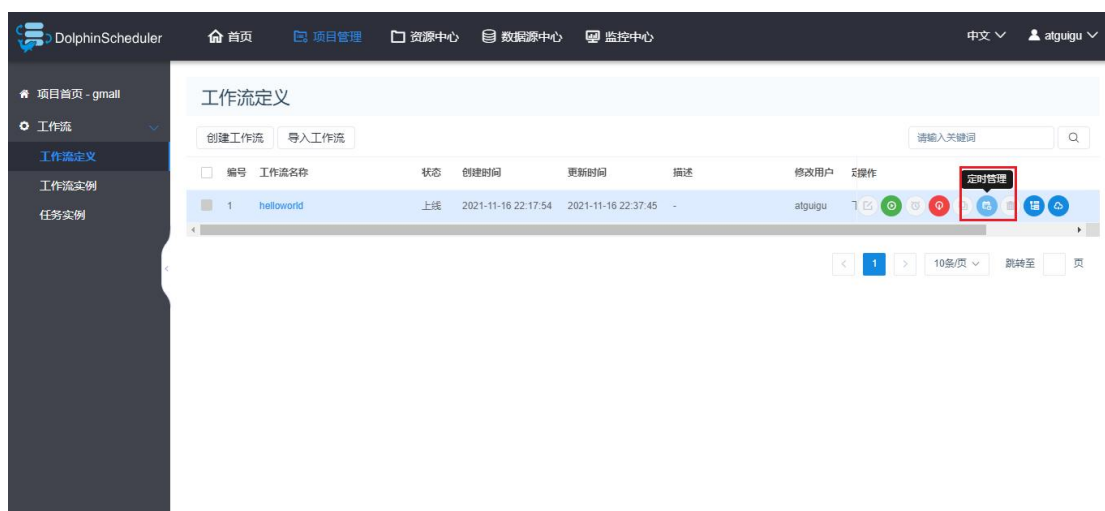


(2) 配置定时规则

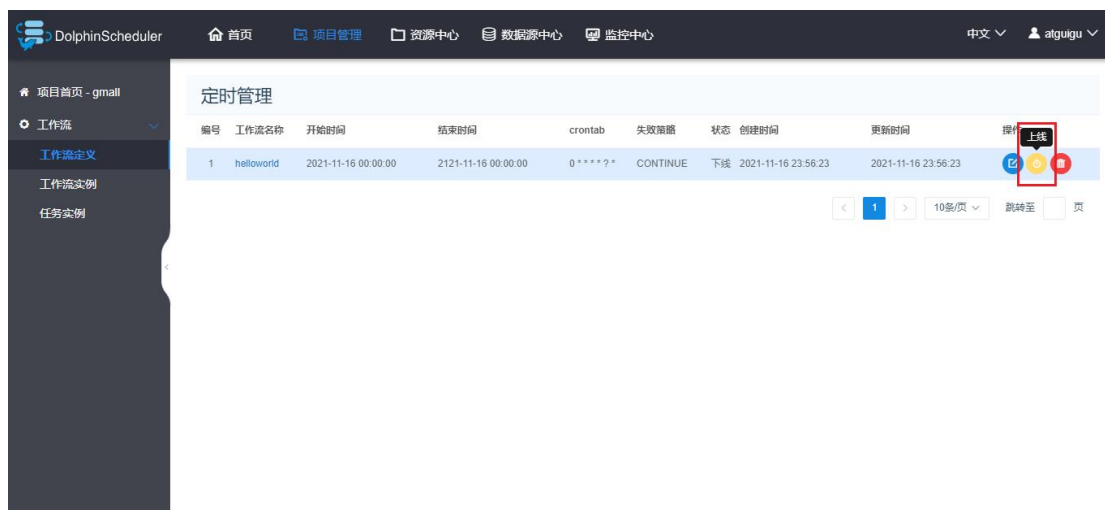
此处示例为每分钟执行一次



(3) 定时管理

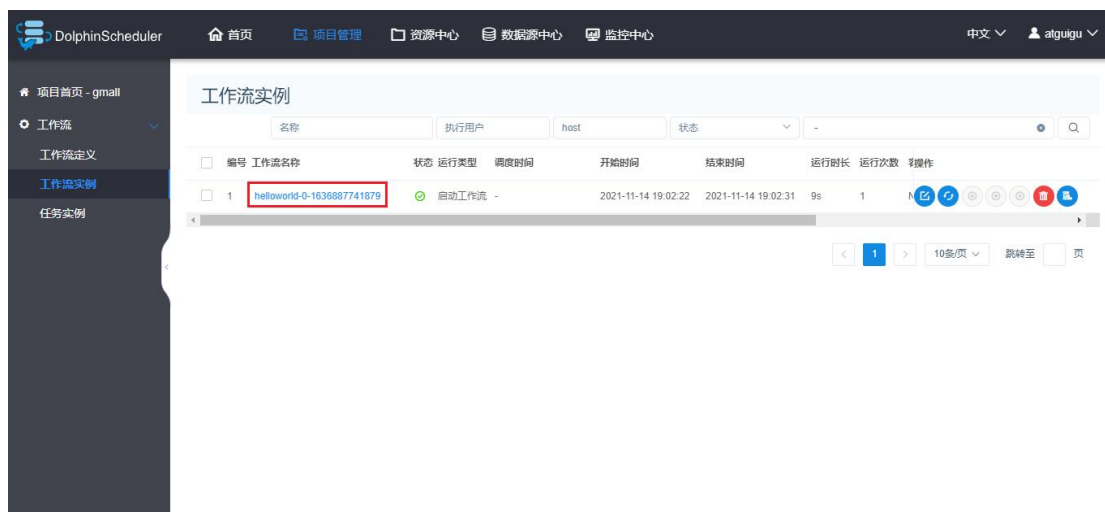


(4) 定时上线

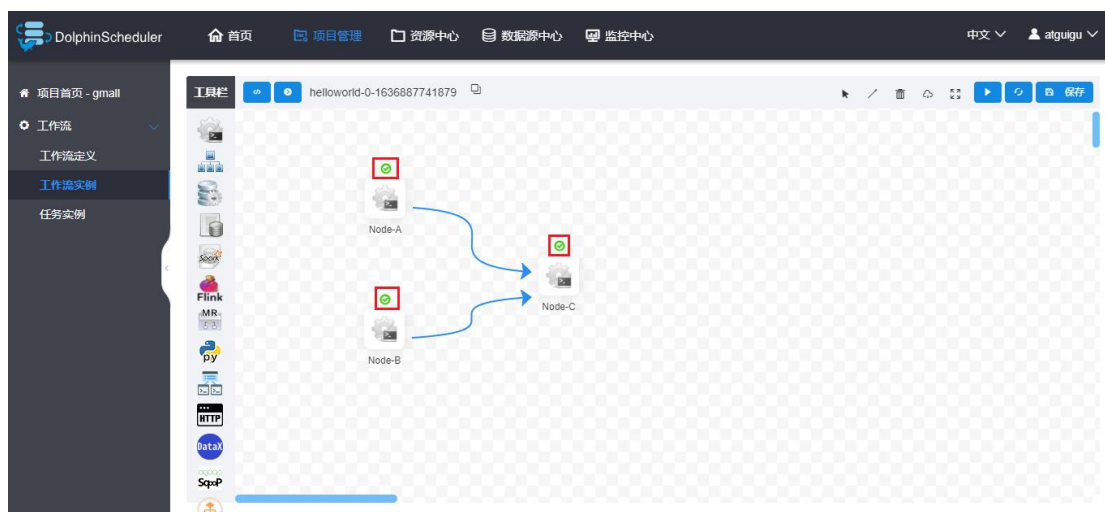


4.3.2 查看 workflow 实例

1) 查看所有 workflow 实例

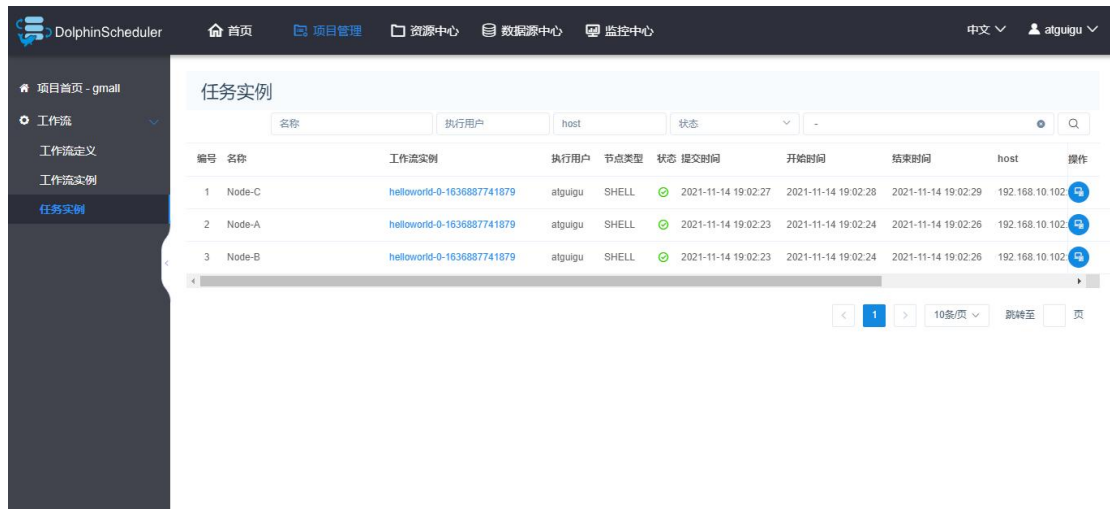


2) 查看 workflow 执行状态






4.3.3 查看任务实例

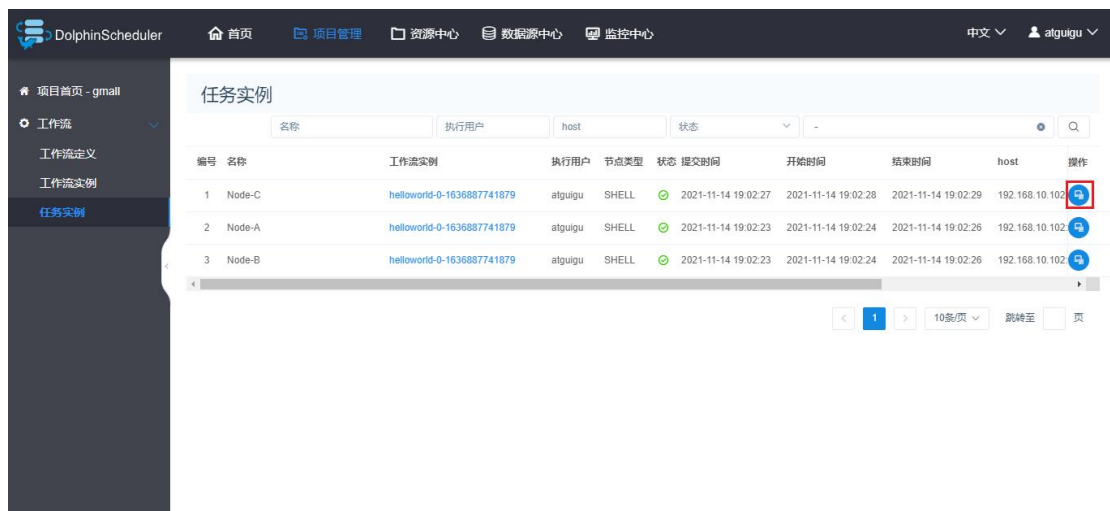
1) 查看所有任务实例






任务实例

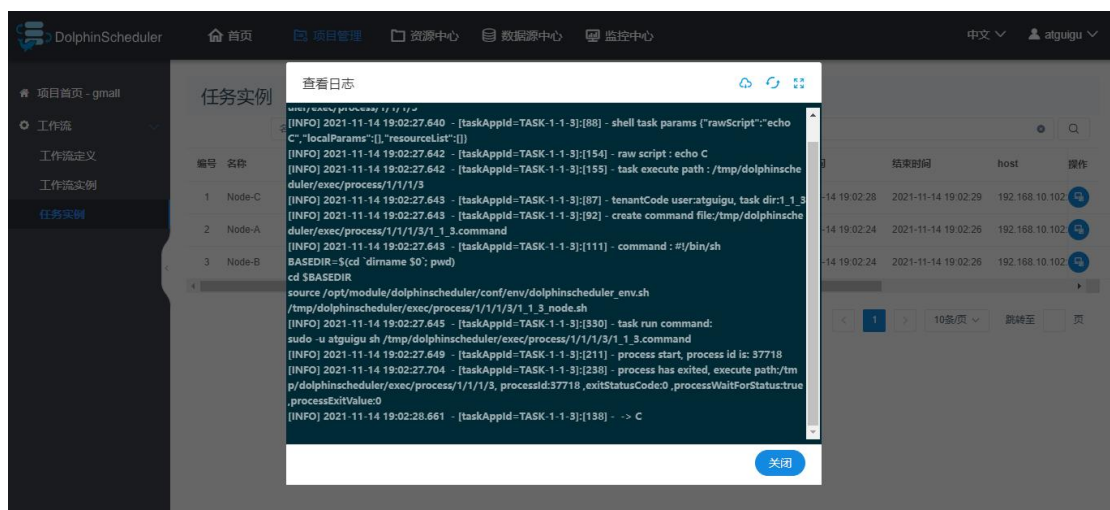
编号	名称	工作流实例	执行用户	节点类型	状态	提交时间	开始时间	结束时间	host	操作
1	Node-C	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:27	2021-11-14 19:02:28	2021-11-14 19:02:29	192.168.10.102	
2	Node-A	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:23	2021-11-14 19:02:24	2021-11-14 19:02:26	192.168.10.102	
3	Node-B	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:23	2021-11-14 19:02:24	2021-11-14 19:02:26	192.168.10.102	

2) 查看任务实例日志



任务实例

编号	名称	工作流实例	执行用户	节点类型	状态	提交时间	开始时间	结束时间	host	操作
1	Node-C	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:27	2021-11-14 19:02:28	2021-11-14 19:02:29	192.168.10.102	
2	Node-A	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:23	2021-11-14 19:02:24	2021-11-14 19:02:26	192.168.10.102	
3	Node-B	helloworld-0-1636887741879	atguigu	SHELL	成功	2021-11-14 19:02:23	2021-11-14 19:02:24	2021-11-14 19:02:26	192.168.10.102	



查看日志

```

[INFO] 2021-11-14 19:02:27.640 - [taskAppId=TASK-1-1-3]:[88] - shell task params {"rawScript":"echo C","localParams":{"resourceList":[]}}
[INFO] 2021-11-14 19:02:27.642 - [taskAppId=TASK-1-1-3]:[154] - raw script: echo C
[INFO] 2021-11-14 19:02:27.642 - [taskAppId=TASK-1-1-3]:[155] - task execute path : /tmp/dolphinscheduler/exec/process/1/1/1/3
[INFO] 2021-11-14 19:02:27.643 - [taskAppId=TASK-1-1-3]:[87] - tenantCode user:atguigu, task dir:1_1_3
[INFO] 2021-11-14 19:02:27.643 - [taskAppId=TASK-1-1-3]:[92] - create command file:/tmp/dolphinscheduler/exec/process/1/1/1/3_1_3.command
[INFO] 2021-11-14 19:02:27.643 - [taskAppId=TASK-1-1-3]:[111] - command: #!/bin/sh
BASEDIR=$(cd "$(dirname $0)" && pwd)
cd $BASEDIR
source /opt/module/dolphinscheduler/conf/env/dolphinscheduler_env.sh
/tmp/dolphinscheduler/exec/process/1/1/1/3_1_3_node.sh
[INFO] 2021-11-14 19:02:27.645 - [taskAppId=TASK-1-1-3]:[330] - task run command:
sudo -u atguigu sh /tmp/dolphinscheduler/exec/process/1/1/1/3_1_3.command
[INFO] 2021-11-14 19:02:27.649 - [taskAppId=TASK-1-1-3]:[211] - process start, process id is: 37718
[INFO] 2021-11-14 19:02:27.704 - [taskAppId=TASK-1-1-3]:[238] - process has exited, execute path:/tmp/dolphinscheduler/exec/process/1/1/1/3, processid:37718, exitStatusCode:0, processWaitForStatus:true, processExitValue:0
[INFO] 2021-11-14 19:02:28.661 - [taskAppId=TASK-1-1-3]:[138] - -- C
    
```

关闭

第 5 章 DolphinScheduler 进阶

5.1 工作流传参

DolphinScheduler 支持对任务节点进行灵活的传参，任务节点可通过 `${参数名}` 引用参数值。

5.1.1 局部参数

局部参数是指只针对单个任务节点有效的参数。

1) 修改 helloworld 工作流 Node-A 节点如下



2) 保存工作流并运行，查看 Node-A 输出日志。

5.1.2 全局参数

全局参数是指针对整个工作流的所有任务节点都有效的参数。

1) 修改 helloworld 工作流每个任务节点如下

(1) 节点 A 配置



The screenshot shows the DolphinScheduler web interface. On the left, the '工作流' (Workflow) menu is selected, and the '工作流定义' (Workflow Definition) sub-menu is active. The main canvas displays a workflow diagram with two nodes, Node-A and Node-B, connected by a blue arrow. The right-hand panel, titled '当前节点设置' (Current Node Settings), is configured for Node-A. The '节点名称' (Node Name) is 'Node-A', the '运行标志' (Run Flag) is '正常' (Normal), and the '任务优先级' (Task Priority) is 'MEDIUM'. The 'Worker分组' (Worker Group) is 'default'. The '失败重试次数' (Failure Retry Count) is 0, and the '失败重试间隔' (Failure Retry Interval) is 1 minute. The '脚本' (Script) field contains 'echo \${dt}'. The '资源' (Resource) field is empty, and the '自定义参数' (Custom Parameters) field is also empty. At the bottom right, there are '取消' (Cancel) and '确认添加' (Confirm Add) buttons.

(2) 节点 B 配置



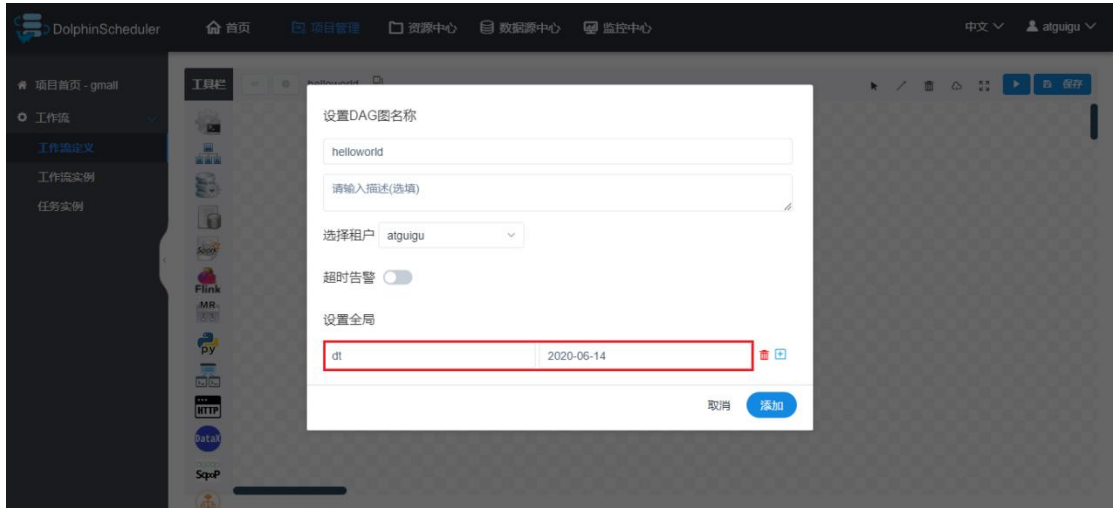
This screenshot is identical to the previous one, but the right-hand panel is now configured for Node-B. The '节点名称' (Node Name) is 'Node-B', and the '脚本' (Script) field contains 'echo \${dt}'. All other settings remain the same.

(3) 节点 C 配置



This screenshot is identical to the previous ones, but the right-hand panel is now configured for Node-C. The '节点名称' (Node Name) is 'Node-C', and the '脚本' (Script) field contains 'echo \${dt}'. All other settings remain the same.

2) 保存工作流，并设置全局参数



3) 执行工作流，查看三个任务节点输出日志。

5.1.3 系统内置参数

DolphinScheduler 提供了一些时间相关的系统参数，方便定时调度使用。

1) 系统参数

参数	说明
<code>\${system.biz.date}</code>	定时时间前一天，格式为 yyyyMMdd
<code>\${system.biz.curdate}</code>	定时时间，格式为 yyyyMMdd
<code>\${system.datetime}</code>	定时时间，格式为 yyyyMMddHHmmss

2) 时间自定义参数

可通过时间自定义参数，设置任意格式、任意时间的日期。

(1) 自定义日期格式

`${yyyyMMdd}`, `$(HHmmss)`, `$(yyyy-MM-dd)`

(2) 自定义时间

参数	说明
<code>\$(add_months(yyyyMMdd,12*N))</code>	后 N 年
<code>\$(add_months(yyyyMMdd,-12*N))</code>	前 N 年
<code>\$(add_months(yyyyMMdd,N))</code>	后 N 月
<code>\$(add_months(yyyyMMdd,-N))</code>	前 N 月
<code>\$(yyyyMMdd+7*N)</code>	后 N 周
<code>\$(yyyyMMdd-7*N)</code>	前 N 周

更多 [Java](#) - [大数据](#) - [前端](#) - [python](#) 人工智能资料下载，可百度访问：尚硅谷官网

<code>\${yyyyMMdd+N}</code>	后 N 天
<code>\${yyyyMMdd-N}</code>	前 N 天
<code>\${HHmmss+N/24}</code>	后 N 小时
<code>\${HHmmss-N/24}</code>	前 N 小时
<code>\${HHmmss+N/24/60}</code>	后 N 分钟
<code>\${HHmmss-N/24/60}</code>	前 N 分钟

3) 配置示例

若执行的脚本需要一个格式为 yyyy-MM-dd 的前一天日期的参数，进行如下配置即可。

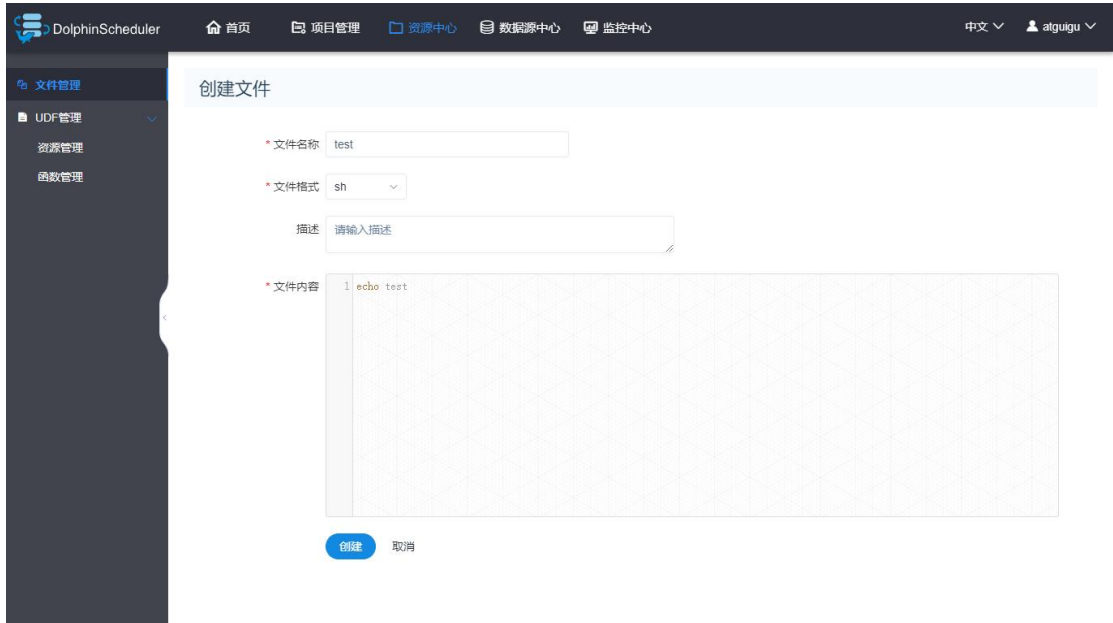


5.2 引用依赖资源

有些任务需要引用一些额外的资源，例如 MR、Spark 等任务需引用 jar 包，Shell 任务需要引用其他脚本等。DolphinScheduler 提供了资源中心来对这些资源进行统一管理。

下面以 Shell 任务为例，演示如何引用资源中心的其他脚本。

1) 在资源中心创建脚本



2) 配置 Shell 任务引用脚本

修改工作流，引用上述脚本



3) 保存工作流并执行，查看对应节点输出日志。

5.3 告警通知

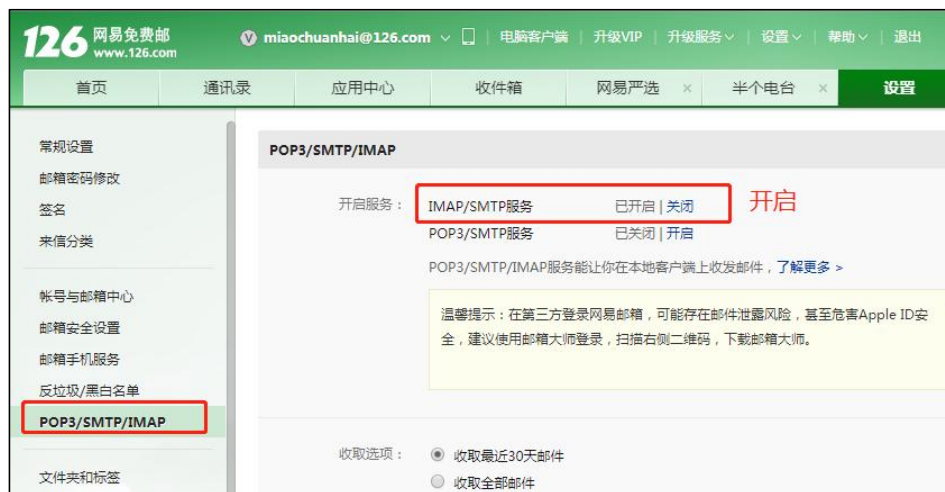
5.3.1 准备电子邮箱账户

如需使用 DolphinScheduler 的邮件告警通知功能，需要准备一个电子邮箱账号，并启用 SMTP 服务。

1) 点击邮箱账号设置



2) 开启 SMTP 服务



3) 授权码是邮箱账号在第三方客户端登录的凭证，须谨记。



5.3.2 配置 AlertServer

1) 修改 AlertServer 配置文件/opt/module/dolphinscheduler/conf/alert.properties

```
[atguigu@hadoop102 ~]$ vim /opt/module/dolphinscheduler/conf/alert.properties
```

2) 配置以下参数

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网

(1) 不使用加密协议

```
#alert type is EMAIL/SMS
alert.type=EMAIL

# mail server configuration
mail.protocol=SMTP
mail.server.host=smtp.qq.com
mail.server.port=25
mail.sender=*****@qq.com
mail.user=*****@qq.com
mail.passwd=*****
# TLS
mail.smtp.starttls.enable=false
# SSL
mail.smtp.ssl.enable=false
mail.smtp.ssl.trust=smtp.exmail.qq.com
```

(2) 使用 STARTTLS 加密协议

```
#alert type is EMAIL/SMS
alert.type=EMAIL

# mail server configuration
mail.protocol=SMTP
mail.server.host=smtp.qq.com
mail.server.port=587
mail.sender=*****@qq.com
mail.user=*****@qq.com
mail.passwd=*****
# TLS
mail.smtp.starttls.enable=true
# SSL
mail.smtp.ssl.enable=false
mail.smtp.ssl.trust=smtp.qq.com
```

(3) 使用 SSL 加密协议

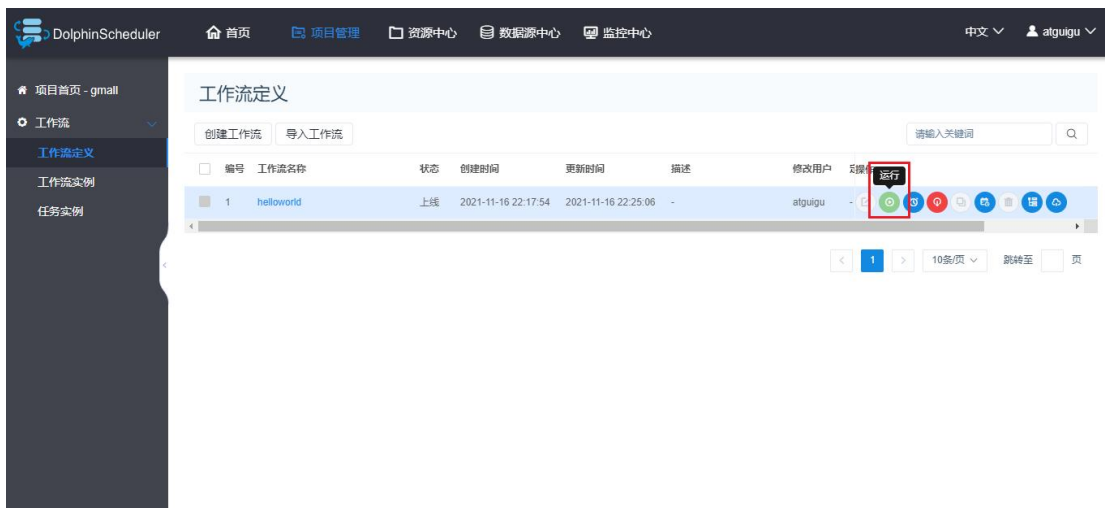
```
#alert type is EMAIL/SMS
alert.type=EMAIL

# mail server configuration
mail.protocol=SMTP
mail.server.host=smtp.qq.com
mail.server.port=465
mail.sender=*****@qq.com
mail.user=*****@qq.com
mail.passwd=*****
# TLS
mail.smtp.starttls.enable=false
# SSL
mail.smtp.ssl.enable=true
mail.smtp.ssl.trust=smtp.qq.com
```

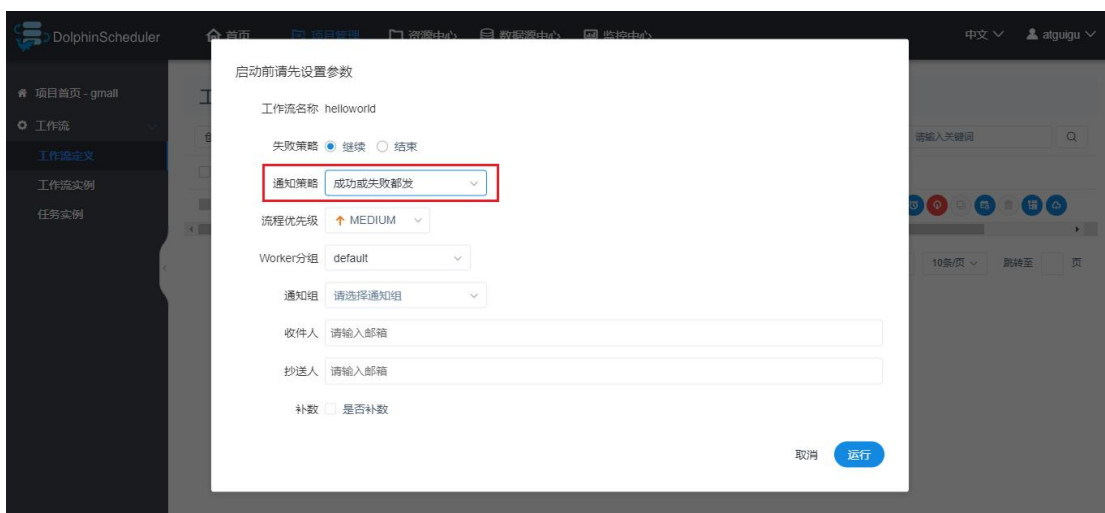
注：各邮箱支持的加密协议：<https://blog.csdn.net/wustzjf/article/details/52481309>

5.3.3 测试

1) 运行 workflow

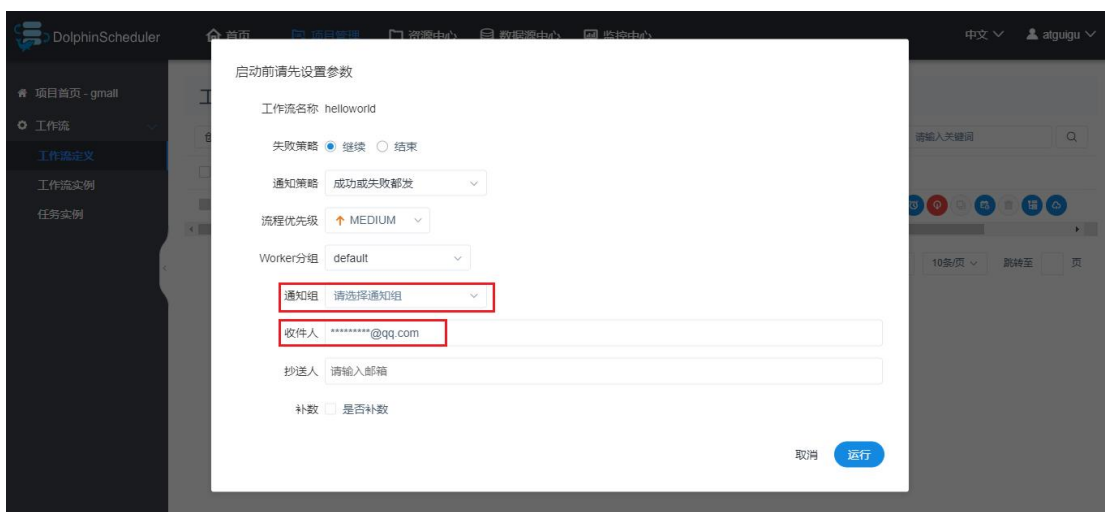


2) 选择通知策略



3) 选择通知人

通知人可通过邮箱执行，也可通过通知组指定。

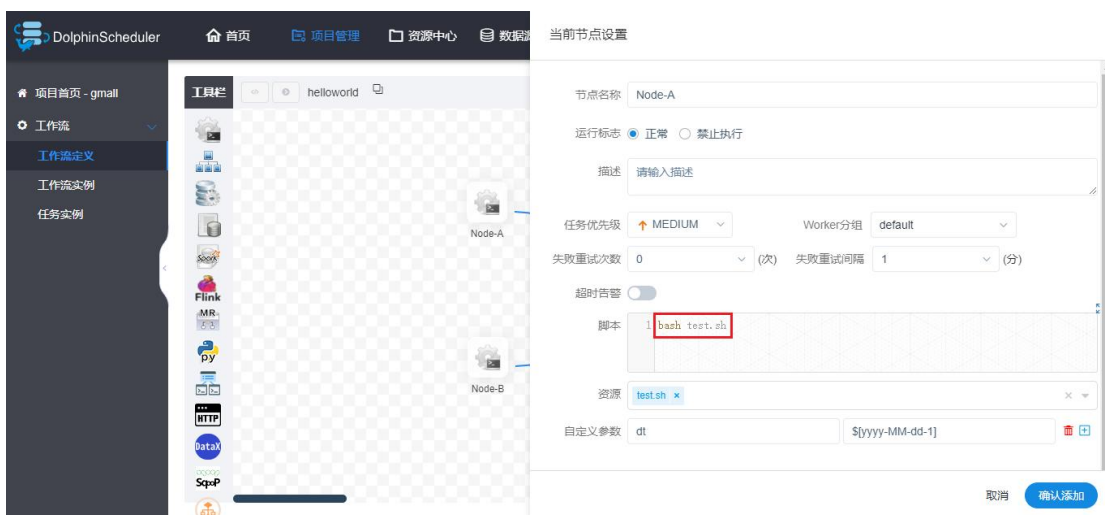


5.4 工作流失败重跑

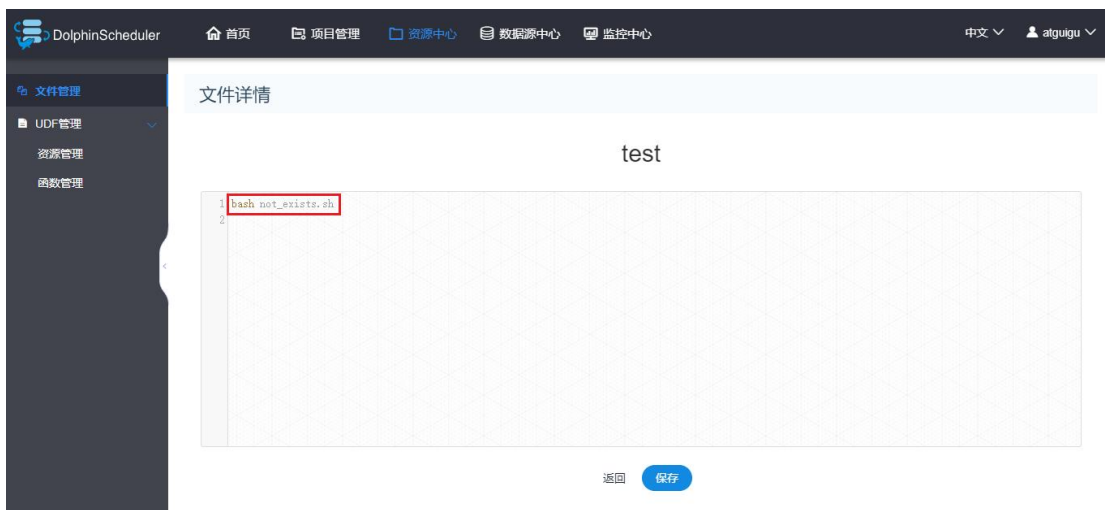
若工作流执行到一半失败了,需要重新启动工作流。重新启动时可选择从起点开始执行,也可选择从断点开始执行。

1) 模拟失败场景

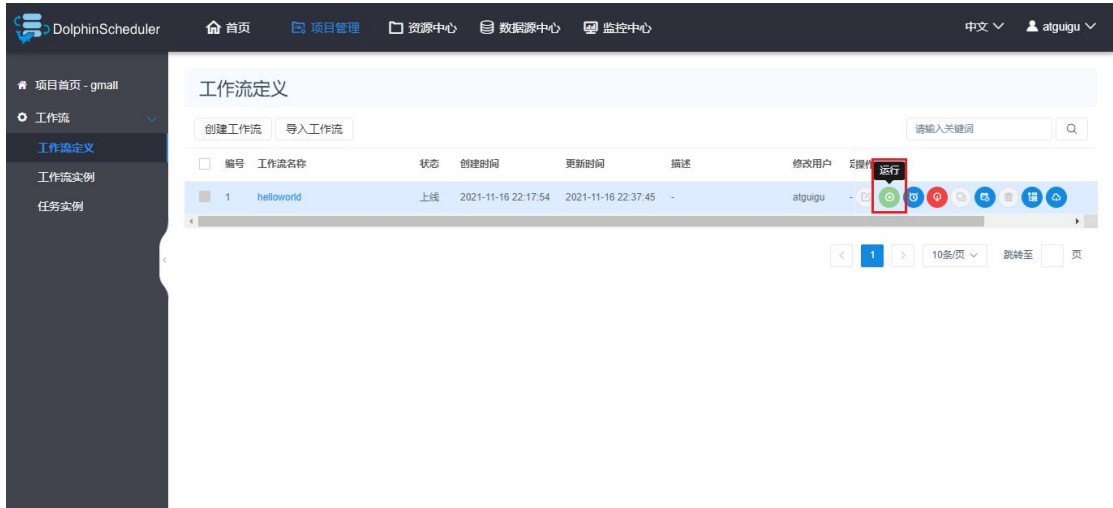
(1) 修改 Node-A 配置如下



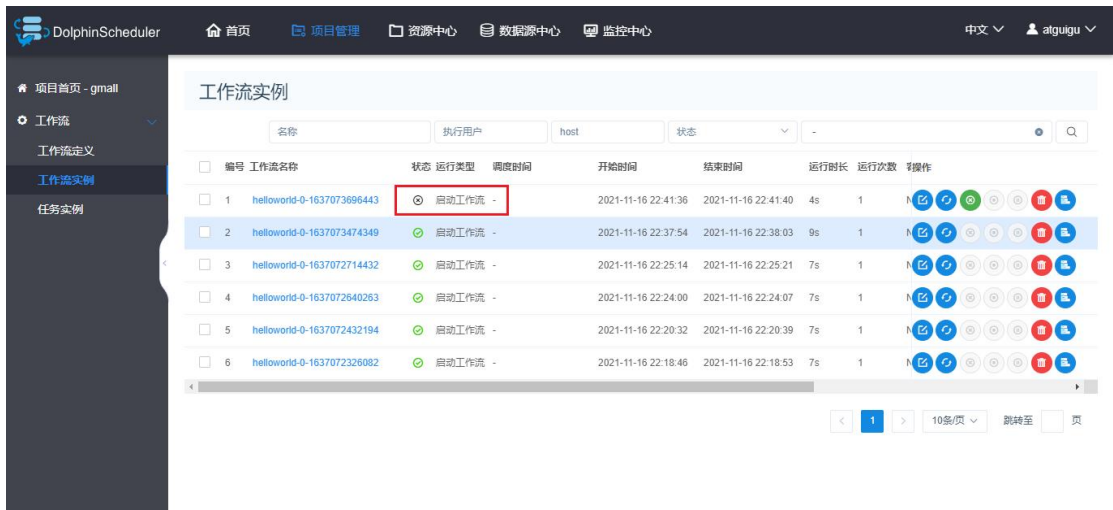
(2) 修改资源中心的 test.sh 文件如下



2) 运行工作流

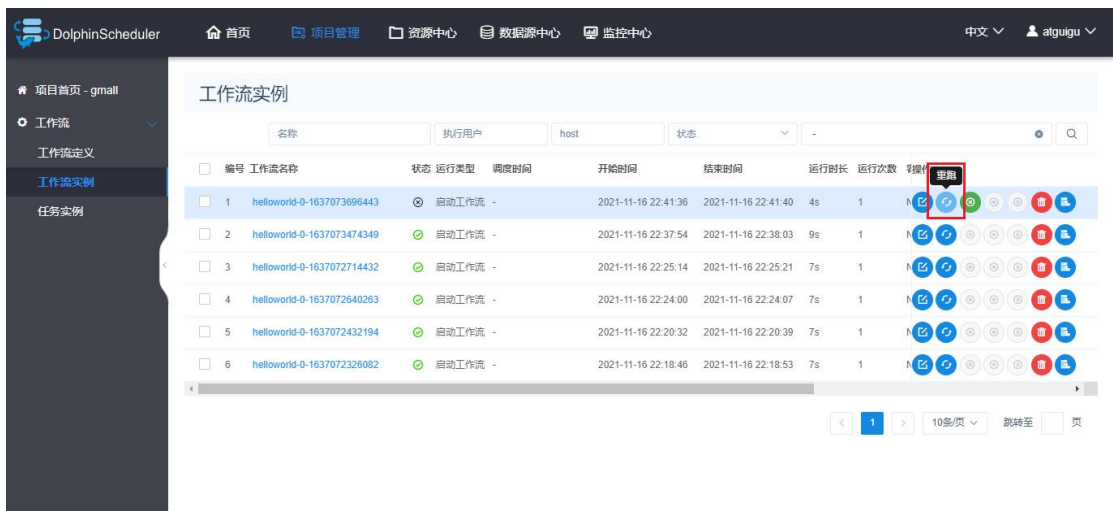


3) 查看工作流实例

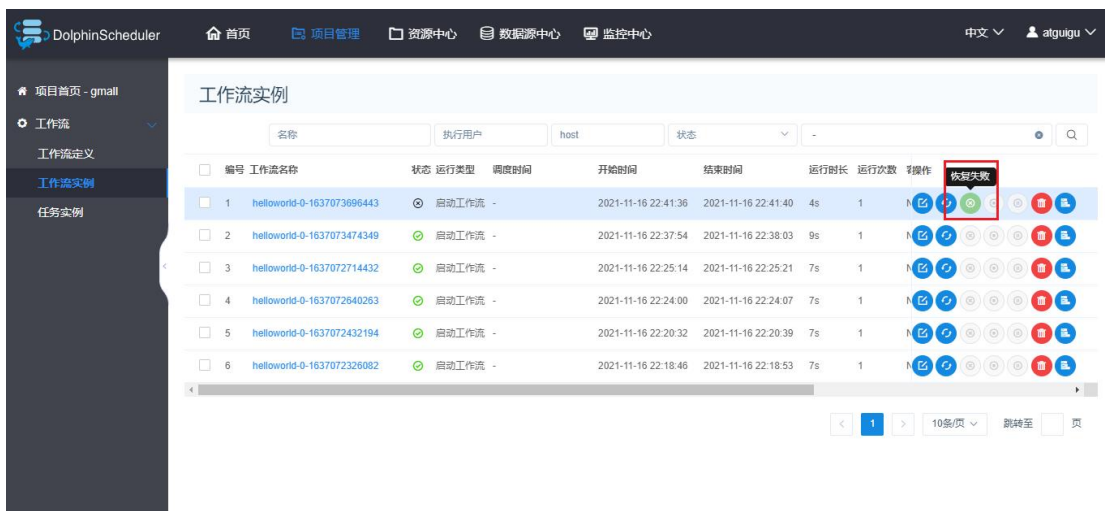


4) 工作失败重跑

(1) 从起点开始



(2) 从断点开始



工作流实例

名称	执行用户	host	状态	操作
1	helloworld-0-1637073996443	-	启动工作流	失败失败
2	helloworld-0-1637073474349	-	启动工作流	成功
3	helloworld-0-1637072714432	-	启动工作流	成功
4	helloworld-0-1637072640263	-	启动工作流	成功
5	helloworld-0-1637072432194	-	启动工作流	成功
6	helloworld-0-1637072326082	-	启动工作流	成功

更多 Java - 大数据 - 前端 - python 人工智能资料下载，可百度访问：尚硅谷官网