

# Azkaban

## 第一章 Azkaban概论

---

### 为什么需要 workflow 调度系统

- 1) 一个完整的数据分析系统通常都是由大量任务单元组成：Shell 脚本程序、Java程序、MapReduce程序、Hive脚本等。
- 2) 各任务单元之间存在 时间先后及前后依赖关系。
- 3) 为了很好地组织起这样的复杂执行计划，需要一个 workflow 调度系统来调度执行；

### 常见 workflow 调度系统

- 1) 简单的任务调度：直接使用Linux的Crontab来定义；
- 2) 复杂的任务调度：开发调度平台或使用现成的开源调度系统，比如 Oozie、Azkaban、Airflow、DolphinScheduler等。

## 第二章 Workflow 案例

---

### 1 HelloWorld案例

- 1) 在windows环境，新建azkaban.project文件，编辑内容如下

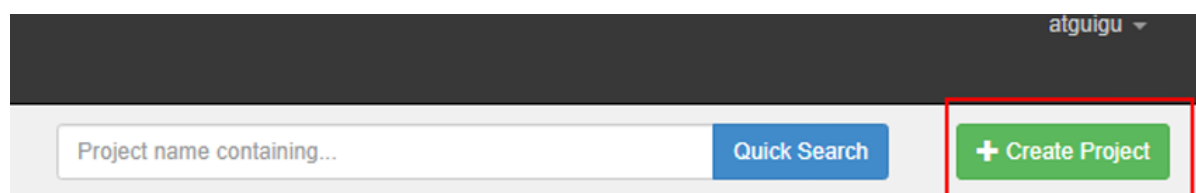
```
1 azkaban-flow-version: 2.0
```

注意：该文件作用，是采用新的Flow-API方式解析flow文件。

- 2) 新建basic.flow文件，内容如下

```
1 nodes:
2   - name: jobA
3     type: command
4     config:
5       command: echo "Hello World"
```

- 3) 将 azkaban.project、basic.flow 文件压缩到一个 zip 文件，文件名称 必须是英文。
- 4) 在WebServer 新建项目：<http://master:8081/index>



- 5) 给项目名称命名和添加项目描述

Create Project

Name

first

项目名称

Description

first

项目描述

Cancel

Create Project

6) first.zip文件上传

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Project first

Delete Project

Upload

Download

Flows

Permissions

Project Logs

No Flows

No flows have been uploaded to this project yet.

上传项目文件

first

first

7) 选择上传的文件

Upload Project Files

Job Archive

选择文件

未选择任何文件

Cancel

Upload

8) 执行任务流

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

Project first

Flows

Permissions

Project Logs

basic

Excute Flow

Excutions

Summary

jobA

Execute Flow basic

Flow View

Right click on the jobs to disable and enable jobs in the flow.

Notification

Failure Options

Concurrent

Flow Parameters

jobA

Schedule

Cancel

Execute

Flow submitted

Execution queued successfully with exec id 1

Continue

9) 在日志中，查看运行结果

Azkaban Test3.84.4My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Flow Execution 1 SUCCEEDED

Submit User atguigu

Duration 1 sec

Start Time 2020-05-29 10:29 46s

End Time 2020-05-29 10:29 48s

Project first / Flow basic / Execution 1

Graph

Flow Trigger List

Job List

Flow Log

Stats

Prepare Execution

Name	Type	Timeline	Start Time	End Time	Elapsed	Status	Details
jobA	command		2020-05-29 10:29 47s	2020-05-29 10:29 47s	0 sec	Success	Log

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Job Execution jobA **SUCCEEDED**

Job Properties

[Project first](#) / [Flow basic](#) / [Execution 2](#) / [Job jobA](#) / [Attempt 0](#)

Job Logs

Job Logs

Refresh

29-05-2020 10:36:03 CST jobA INFO - Starting job jobA at 1590719763591

29-05-2020 10:36:03 CST jobA INFO - job JVM args: '-Dazkaban.flowid=basic' '-Dazkaban.execid=2' '-Dazkaban.jobid=jobA'

29-05-2020 10:36:03 CST jobA INFO - user.to.proxy property was not set, defaulting to submit user atguigu

29-05-2020 10:36:03 CST jobA INFO - Attached Ramp Props : [{}]

29-05-2020 10:36:03 CST jobA INFO - Building command job executor.

29-05-2020 10:36:03 CST jobA INFO - Failed with 5 inputs with exception e = null

29-05-2020 10:36:03 CST jobA INFO - Memory granted for job jobA

29-05-2020 10:36:03 CST jobA INFO - 1 commands to execute.

29-05-2020 10:36:03 CST jobA INFO - cwd=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/2

29-05-2020 10:36:03 CST jobA INFO - effective user is: atguigu

29-05-2020 10:36:03 CST jobA INFO - Command: echo "Hello World"

29-05-2020 10:36:03 CST jobA INFO - Environment variables: {JOB\_OUTPUT\_PROP\_FILE=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/2/jobA\_output\_351957176180045449}

29-05-2020 10:36:03 CST jobA INFO - Working directory: /opt/module/azkaban/azkaban-exec-server-3.84.4/executions/2

29-05-2020 10:36:03 CST jobA INFO - Spawned process with id 5257

29-05-2020 10:36:03 CST jobA INFO - **Hello World**

29-05-2020 10:36:03 CST jobA INFO - Process with id 5257 completed successfully in 0 seconds.

29-05-2020 10:36:03 CST jobA INFO - output properties file=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/2/jobA\_output\_3519571761800454491\_tmp

29-05-2020 10:36:03 CST jobA INFO - Finishing job jobA at 1590719763714 with status SUCCEEDED

## 2 作业依赖案例

需求: JobA和JobB执行完了, 才能执行JobC

具体步骤:

1) 修改basic.flow为如下内容

```
1 nodes:
2   - name: jobC
3     type: command
4     # jobC 依赖 JobA和JobB
5     dependsOn:
6       - jobA
7       - jobB
8     config:
9       command: echo "I' m JobC"
10
11  - name: jobA
12    type: command
13    config:
14      command: echo "I' m JobA"
15
16  - name: jobB
17    type: command
18    config:
19      command: echo "I' m JobB"
```

2) 将修改后的basic.flow和azkaban.project压缩成second.zip文件

3) 重复HelloWorld案例后续步骤。

## Create Project



Name

second

Description

second

Cancel

Create Project

## Upload Project Files



Job Archive

选择文件

second.zip

Cancel

Upload



**Azkaban Test**  
3.84.4

My Local Azkaban

**Projects**

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

### Project second

Flows

Permissions

Project Logs

^ basic

Execute Flow

Executions

Summary

jobB

jobA

jobC

Flow View

Right click on the jobs to disable and enable jobs in the flow.

Notification

Failure Options

Concurrent

Flow Parameters

```

graph TD
    jobA --> jobC
    jobB --> jobC
  
```

Schedule

Cancel

Execute

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Flow Execution 4

SUCCEEDED

Submit User atguigu

Duration 0 sec

Start Time 2020-05-29 11:42 47s

End Time 2020-05-29 11:42 48s

Project second / Flow basic / Execution 4

Graph

Flow Trigger List

Job List

Flow Log

Stats

Prepare Execution

Name	Type	Timeline	Start Time	End Time	Elapsed	Status	Details
jobB	command		2020-05-29 11:42 47s	2020-05-29 11:42 47s	0 sec	Success	Log
jobA	command		2020-05-29 11:42 47s	2020-05-29 11:42 47s	0 sec	Success	Log
jobC	command		2020-05-29 11:42 48s	2020-05-29 11:42 48s	0 sec	Success	Log

Job Execution jobC

SUCCEEDED

Job Properties

Project second / Flow basic / Execution 4 / Job jobC / Attempt 0

Job Logs

Job Logs

Refresh

```

29-05-2020 11:42:48 CST jobC INFO - Starting job jobC at 1590723768031
29-05-2020 11:42:48 CST jobC INFO - job JVM args: '-Dazkaban.flowid=basic' '-Dazkaban.execid=4' '-Dazkaban.jobid=jobC'
29-05-2020 11:42:48 CST jobC INFO - user.to.proxy property was not set, defaulting to submit user atguigu
29-05-2020 11:42:48 CST jobC INFO - Attached Ramp Props : [{}]]
29-05-2020 11:42:48 CST jobC INFO - Building command job executor.
29-05-2020 11:42:48 CST jobC INFO - Failed with 5 inputs with exception e = null
29-05-2020 11:42:48 CST jobC INFO - Memory granted for job jobC
29-05-2020 11:42:48 CST jobC INFO - 1 commands to execute.
29-05-2020 11:42:48 CST jobC INFO - cwd=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/4
29-05-2020 11:42:48 CST jobC INFO - effective user is: atguigu
29-05-2020 11:42:48 CST jobC INFO - Command: echo "I'm JobC"
29-05-2020 11:42:48 CST jobC INFO - Environment variables: {JOB_OUTPUT_PROP_FILE=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/4/jobC_output_257836250422777364}
29-05-2020 11:42:48 CST jobC INFO - Working directory: /opt/module/azkaban/azkaban-exec-server-3.84.4/executions/4
29-05-2020 11:42:48 CST jobC INFO - Spawned process with id 6018
29-05-2020 11:42:48 CST jobC INFO - I'm JobC
29-05-2020 11:42:48 CST jobC INFO - Process with id 6018 completed successfully in 0 seconds.
29-05-2020 11:42:48 CST jobC INFO - output properties file=/opt/module/azkaban/azkaban-exec-server-3.84.4/executions/4/jobC_output_257836250422777364_tmp
29-05-2020 11:42:48 CST jobC INFO - Finishing job jobC at 1590723768456 with status SUCCEEDED
  
```

### 3 自动失败重试案例

需求：如果执行任务失败，需要重试3次，重试的时间间隔10000ms

具体步骤：

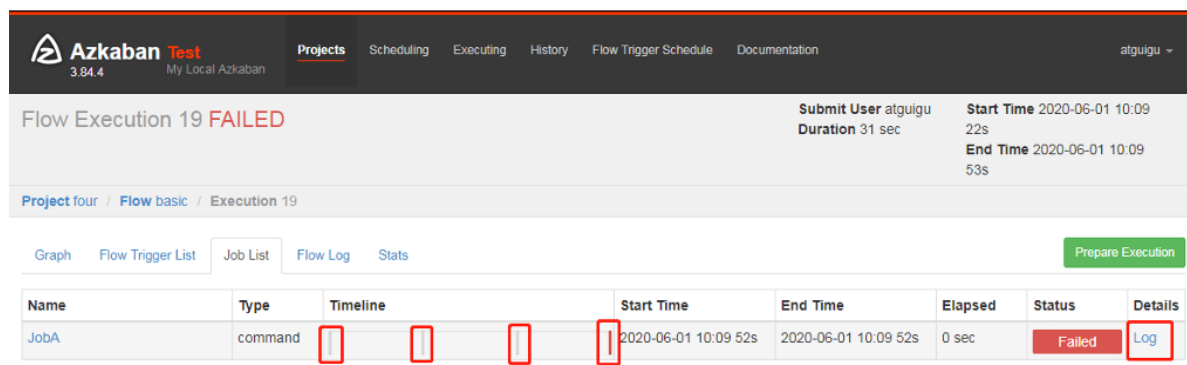
#### 1) 编译配置流

```
1 nodes:
2   - name: JobA
3     type: command
4     config:
5       command: sh /not_exists.sh
6       retries: 3
7       retry.backoff: 10000
```

#### 2) 将修改后的basic.flow和azkaban.project压缩成four.zip文件

#### 3) 重复HelloWorld案例后续步骤。

#### 4) 执行并观察到一次失败+三次重试



The screenshot shows the Azkaban Test web interface. At the top, there's a navigation bar with 'Projects', 'Scheduling', 'Executing', 'History', 'Flow Trigger Schedule', and 'Documentation'. The main header shows 'Flow Execution 19 FAILED'. Below this, a table lists jobs. The job 'JobA' is highlighted in blue, and its status is 'Failed'. A 'Log' button is visible next to the job name.

#### 5) 也可以点击上图中的Log，在任务日志中看到，总共执行了4次。



The screenshot shows the Azkaban Job Logs page. The logs show a series of messages indicating that the job 'JobA' failed and was retried multiple times. The final message shows the job completed successfully after 4 attempts.

### 4 手动失败重试案例

需求：JobA -> JobB (依赖于A) -> JobC -> JobD -> JobE -> JobF。生产环境下任何Job都有可能挂掉，可以根据需求执行想要执行的Job。

具体步骤：

#### 1) 编译配置流

```
1  nodes:
2    - name: JobA
3      type: command
4      config:
5        command: echo "This is JobA."
6
7    - name: JobB
8      type: command
9      dependsOn:
10       - JobA
11      config:
12        command: echo "This is JobB."
13
14    - name: JobC
15      type: command
16      dependsOn:
17       - JobB
18      config:
19        command: echo "This is JobC."
20
21    - name: JobD
22      type: command
23      dependsOn:
24       - JobC
25      config:
26        command: echo "This is JobD."
27
28    - name: JobE
29      type: command
30      dependsOn:
31       - JobD
32      config:
33        command: echo "This is JobE."
34
35    - name: JobF
36      type: command
37      dependsOn:
38       - JobE
39      config:
40        command: echo "This is JobF."
```

2) 将修改后的basic.flow和azkaban.project压缩成five.zip文件

3) 重复HelloWorld案例后续步骤。



**Azkaban Test** 3.84.4 My Local Azkaban

Projects | Scheduling | Executing | History | Flow Trigger Schedule | Documentation

Project five Delete Project Upload Download

Flows | Permissions | Project Logs

basic Execute Flow Executions Summary

JobA
JobB
JobC
JobD
JobE
JobF

**five**  
five

Created on 2020-05-29 14:36:31  
Last modified on 2020-05-29 14:36:41  
Modified by atguigu

Project admins: atguigu  
Your Permissions: ADMIN, METRICS

**Execute Flow basic**

Flow View

Right click on the jobs to disable and enable jobs in the flow.

- Notification
- Failure Options
- Concurrent
- Flow Parameters

JobA  
JobB  
JobC  
JobD  
JobE  
JobF

Open Job in New Window...  
Enable  
Disable  
Center Job

Parents  
Ancestors  
Children  
Descendents  
Enable All

Flow View

Right click on the jobs to disable and enable jobs in the flow.

- Notification
- Failure Options
- Concurrent
- Flow Parameters

JobA  
JobB  
JobC  
JobD  
JobE  
JobF

Schedule Cancel Execute

Enable和Disable下面都分别有如下参数：

Parents：该作业的上一个任务

Ancestors：该作业前的所有任务

Children：该作业后的一个任务

Descendents: 该作业后的所有任务

Enable All: 所有的任务

4) 可以根据需求选择性执行对应的任务。

## 第三章 Azkaban进阶

---

### JavaProcess作业类型案例

JavaProcess类型可以运行一个自定义主类方法，type类型为javaprocess，可用的配置为：

Xms: 最小堆

Xmx: 最大堆

classpath: 类路径

java.class: 要运行的 Java 对象，其中必须包含Main方法

main.args: main 方法的参数

案例：

1) 新建一个azkaban的maven工程

2) 创建包名：com.atguigu

3) 创建AzTest类

```
1 package com.atguigu;
2
3 public class AzTest {
4     public static void main(String[] args) {
5         System.out.println("This is for testing!");
6     }
7 }
```

4) 打包成jar包azkaban-1.0-SNAPSHOT.jar

5) 新建testJava.flow，内容如下

```
1 nodes:
2   - name: test_java
3     type: javaprocess
4     config:
5       Xms: 96M
6       Xmx: 200M
7       java.class: com.atguigu.AzTest
```

6) 将Jar包、flow文件和project文件打包成javatest.zip

7) 创建项目 -> 上传javatest.zip -> 执行作业 -> 观察结果

Create Project

Name

javatest

Description

javatest

Cancel

Create Project

Upload Project Files

Job Archive

选择文件

javatest.zip

Cancel

Upload

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Flow Execution 21

SUCCEEDED

Submit User atguigu

Duration 0 sec

Start Time 2020-06-01 10:37 54s

End Time 2020-06-01 10:37 55s

Project javatest

Flow testJava

Execution 21

Graph

Flow Trigger List

Job List

Flow Log

Stats

Prepare Execution

Name	Type	Timeline	Start Time	End Time	Elapsed	Status	Details
test_java	javaprocess	<div></div>	2020-06-01 10:37 54s	2020-06-01 10:37 54s	0 sec	Success	Log

Azkaban Test

3.84.4

My Local Azkaban

Projects

Scheduling

Executing

History

Flow Trigger Schedule

Documentation

atguigu

Job Execution test\_java SUCCEEDED

Job Properties

Project javatest / Flow testJava / Execution 21 / Job test\_java / Attempt 0

Job Logs

Job Logs

Refresh

01-06-2020 10:37:54 CST test\_java INFO - Starting job test\_java at 1590979074558

01-06-2020 10:37:54 CST test\_java INFO - job JVM args: "-Dazkaban.flowid=testJava" "-Dazkaban.execid=21" "-Dazkaban.jobid=test\_java"

01-06-2020 10:37:54 CST test\_java INFO - user.to.proxy property was not set, defaulting to submit user atguigu

01-06-2020 10:37:54 CST test\_java INFO - Attached Ramp Props : [{}]

01-06-2020 10:37:54 CST test\_java INFO - Building javaprocess job executor.

01-06-2020 10:37:54 CST test\_java INFO - Failed with 5 inputs with exception e = null

01-06-2020 10:37:54 CST test\_java INFO - Memory granted for job test\_java

01-06-2020 10:37:54 CST test\_java INFO - No classpath specified. Trying to load classes from /opt/module/azkaban/azkaban-exec/executions/21

01-06-2020 10:37:54 CST test\_java INFO - 1 commands to execute.

01-06-2020 10:37:54 CST test\_java INFO - cwd=/opt/module/azkaban/azkaban-exec/executions/21

01-06-2020 10:37:54 CST test\_java INFO - effective user is: atguigu

01-06-2020 10:37:54 CST test\_java INFO - Command: java "-Dazkaban.flowid=testJava" "-Dazkaban.execid=21" "-Dazkaban.jobid=test\_java" -Xms96M -Xmx200M -cp azkaban-1.0-SNAPSHOT

01-06-2020 10:37:54 CST test\_java INFO - Environment variables: {JOB\_OUTPUT\_PROP\_FILE=/opt/module/azkaban/azkaban-exec/executions/21/test\_java\_output\_5415600962142294670\_t

01-06-2020 10:37:54 CST test\_java INFO - Working directory: /opt/module/azkaban/azkaban-exec/executions/21

01-06-2020 10:37:54 CST test\_java INFO - Spawned process with id 6193

01-06-2020 10:37:54 CST test\_java INFO - 

This is for testing!

01-06-2020 10:37:54 CST test\_java INFO - Process with id 6193 completed successfully in 0 seconds.

01-06-2020 10:37:54 CST test\_java INFO - output properties file=/opt/module/azkaban/azkaban-exec/executions/21/test\_java\_output\_5415600962142294670\_tmp

01-06-2020 10:37:54 CST test\_java INFO - Finishing job test\_java at 1590979074943 with status SUCCEEDED

## 条件 workflow 案例

条件 workflow 功能允许用户 **自定义执行条件** 来决定是否运行某些 Job。条件可以由当前 Job 的父 Job 输出的运行时参数构成，也可以使用预定义宏。在这些条件下，用户可以在确定 Job 执行逻辑时获得更大的灵活性，例如，只要父 Job 之一成功，就可以运行当前 Job。

## 运行时参数案例

### 1) 基本原理

- 父 Job 将参数写入 JOB\_OUTPUT\_PROP\_FILE 环境变量所指向的文件
- 子 Job 使用 \${jobName:param} 来获取父 Job 输出的参数并定义执行条件

### 2) 支持的条件运算符：

- == 等于
- != 不等于
- > 大于
- >= 大于等于
- < 小于
- <= 小于等于
- && 与
- || 或
- ! 非

### 3) 案例：

需求：JobA 执行一个 shell 脚本。JobB 执行一个 shell 脚本，但 JobB 不需要每天都执行，而只需要每个周一执行。

- 新建 JobA.sh

```

1  #!/bin/bash
2  echo "do JobA"
3  wk=`date +%w`
4  echo "{\"wk\":\"$wk\"} > $JOB_OUTPUT_PROP_FILE

```

## (2) 新建JobB.sh

```

1  #!/bin/bash
2  echo "do JobB"

```

## (3) 新建condition.flow

```

1  nodes:
2    - name: JobA
3      type: command
4      config:
5        command: sh JobA.sh
6    - name: JobB
7      type: command
8      dependsOn: - JobA
9      config:
10       command: sh JobB.sh
11       condition: ${JobA:wk} == 1

```

(4) 将JobA.sh、JobB.sh、condition.flow 和 azkaban.project 打包成 condition.zip

(5) 创建condition 项目 -> 上传condition.zip 文件 -> 执行作业 -> 观察结果

(6) 按照我们设定的条件，JobB会根据当日日期决定是否执行。

## 预定义宏案例

Azkaban中预置了几个特殊的判断条件，称为 **预定义宏**。预定义宏会根据所有父Job的完成情况进行判断，再决定是否执行。可用的预定义宏如下：

- (1) all\_success: 表示父 Job 全部成功才执行(默认)
- (2) all\_done: 表示父Job全部完成才执行
- (3) all\_failed: 表示父 Job 全部失败才执行
- (4) one\_success: 表示父Job至少一个成功才执行
- (5) one\_failed: 表示父Job 至少一个失败才执行

### 1) 案例

需求：JobA 执行一个shell脚本，JobB 执行一个shell脚本，JobC执行一个shell脚本，要求JobA、JobB中有一个成功即可执行

#### (1) 新建JobA.sh

```

1  #!/bin/bash
2  echo "do JobA"

```

#### (2) 新建JobC.sh

```

1  #!/bin/bash
2  echo "do JobC"

```

### (3) 新建macro.flow

```
1  nodes:
2    - name: JobA
3      type: command
4      config:
5        command: sh JobA.sh
6    - name: JobB
7      type: command
8      config:
9        command: sh JobB.sh
10   - name: JobC
11     type: command
12     dependsOn: - JobA
13               - JobB
14     config:
15       command: sh JobC.sh
16       condition: one_success
```

(4) JobA.sh、JobC.sh、macro.flow、azkaban.project 文件，打包成 macro.zip。注意：没有JobB.sh。

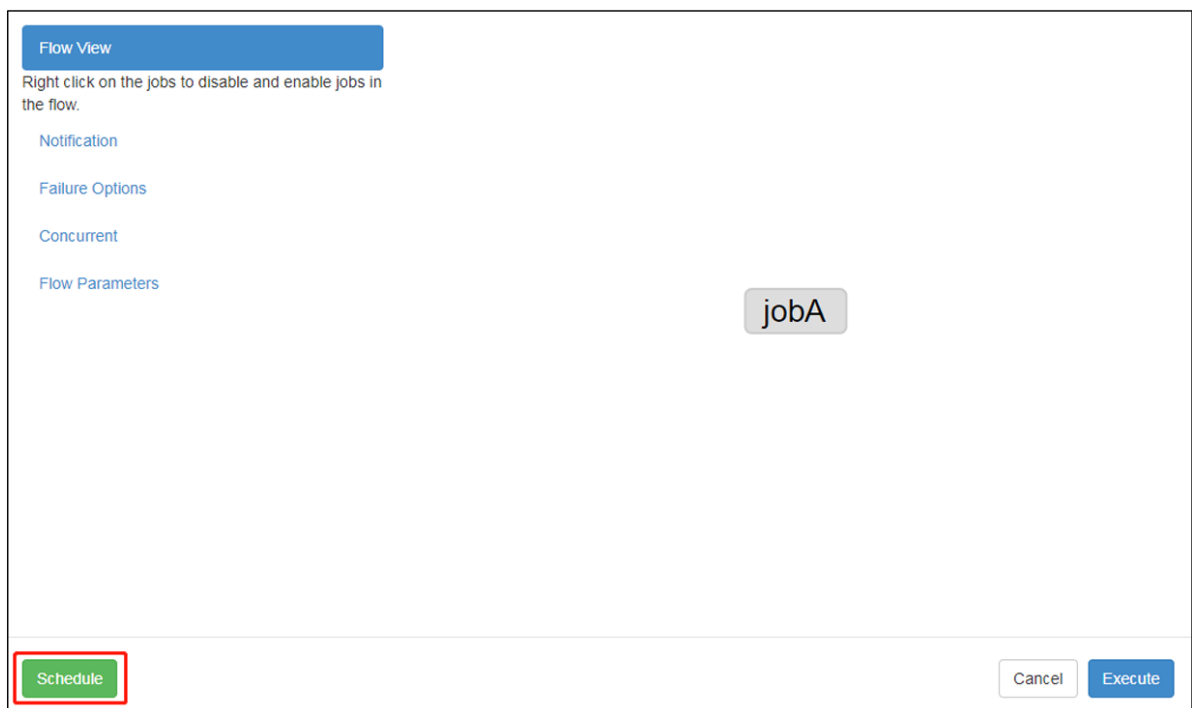
(5) 创建macro项目 -> 上传macro.zip文件 -> 执行作业 -> 观察结果

## 定时执行案例

需求：JobA每间隔1分钟执行一次；

具体步骤：

1) Azkaban 可以定时执行工作流。在执行工作流时候，选择左下角Schedule。



2) 右上角注意时区是上海，然后在左面填写具体执行事件，填写的方法和crontab配置定时 任务规则一致。

## Schedule Flow Options



*All schedules are based on the server timezone: **Asia/Shanghai**.*

Warning: the execution will be skipped if it is scheduled to run during the hour that is lost when DST starts in the Spring. E.g. there is no 2 - 3 AM when PST switches to PDT.

Min	<input type="text" value="*/1"/>
Hours	<input type="text" value="*"/>
Day of Month	<input type="text" value="?"/>
Month	<input type="text" value="*"/>
Day of Week	<input type="text" value="*"/>
Year	<input type="text"/>
TimeZone	<input type="text" value="UTC"/>

**Special Characters:**

- \* any value
- , value list separators
- range of values
- / step values
- 0-59 allowed values

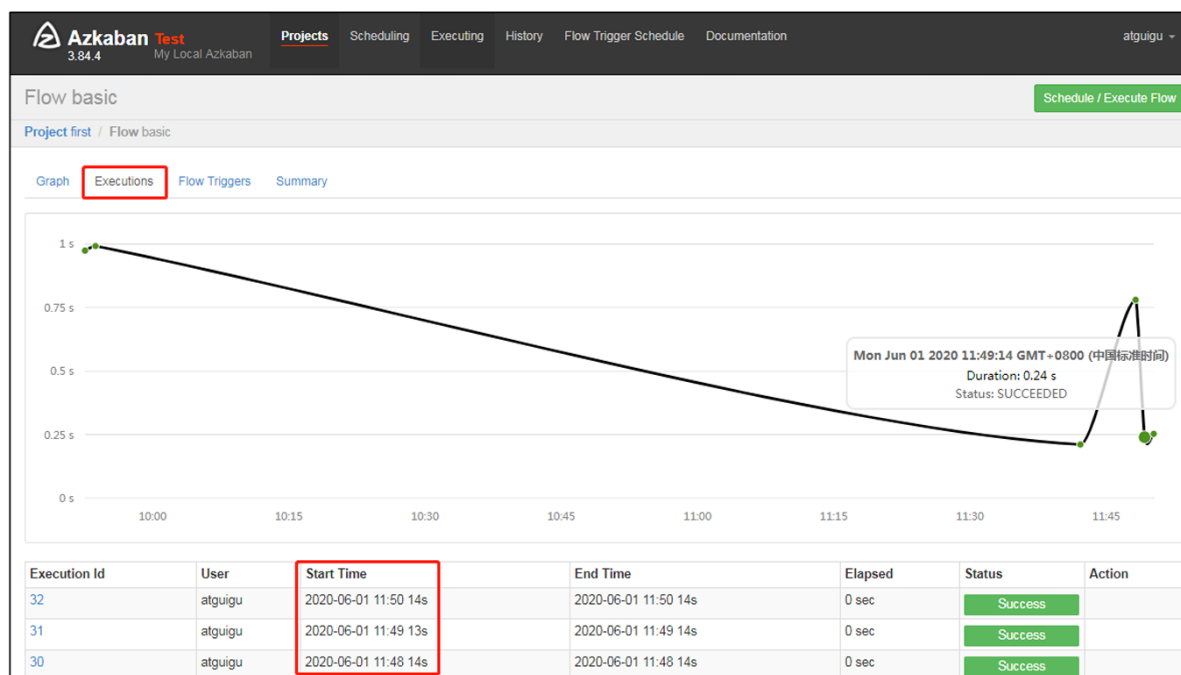
[Detailed instructions.](#)

Next 10 scheduled executions for this cron expression only:

- 2020-06-01T11:52:00
- 2020-06-01T11:53:00
- 2020-06-01T11:54:00
- 2020-06-01T11:55:00
- 2020-06-01T11:56:00
- 2020-06-01T11:57:00
- 2020-06-01T11:58:00
- 2020-06-01T11:59:00
- 2020-06-01T12:00:00
- 2020-06-01T12:01:00

### 3) 观察结果

<div> <div>Azkaban Test</div> <div>3.84.4</div> <div>My Local Azkaban</div> </div> <div> <a href="#">Projects</a> <a href="#">Scheduling</a> <a href="#">Executing</a> <a href="#">History</a> <a href="#">Flow Trigger Schedule</a> <a href="#">Documentation</a> </div> <div>atguigu</div>											
Scheduled Flows											
* Click column headers to sort.											
#	ID	Flow	Project	Submitted By	First Scheduled to Run	Next Execution Time	Repeats Every	Cron Expression	Execution Options	Has SLA	Action
1	1	basic	first	atguigu	2020-06-01 11:47:35	2020-06-01 11:48:00	Not Applicable	0 */1 * ? * *	<a href="#">Show</a>	false	<a href="#">Remove Schedule</a> <a href="#">Set SLA</a>



Flow basic

Schedule / Execute Flow


Project first / Flow basic

[Graph](#)
[Executions](#)
[Flow Triggers](#)
[Summary](#)

### 4) 删除定时调度

点击remove Schedule即可删除当前任务的调度规则。



<div>  <div> <div>Azkaban Test</div> <div>3.84.4</div> <div>My Local Azkaban</div> </div> </div> <div> <div>Projects</div> <div>Scheduling</div> <div>Executing</div> <div>History</div> <div>Flow Trigger Schedule</div> <div>Documentation</div> </div> <div>atguigu</div>											
Scheduled Flows											
* Click column headers to sort.											
#	ID	Flow	Project	Submitted By	First Scheduled to Run	Next Execution Time	Repeats Every	Cron Expression	Execution Options	Has SLA	Action
1	3	basic	first	atguigu	2020-06-01 11:53:42	2020-06-01 11:54:00	Not Applicable	0 */1 * ? * *	Show	false	Remove Schedule Set SLA

## 多Executor模式注意事项

Azkaban多Executor模式是指，在集群中多个节点部署Executor。在这种模式下，Azkaban web Server会根据策略，选取其中一个Executor去执行任务。为确保所选的Executor能够准确的执行任务，我们须在以下两种方案任选其一，推荐使用方案二。

方案一：指定特定的Executor（hadoop102）去执行任务。

1) 在MySQL中azkaban数据库executors表中，查询hadoop102上的Executor的id。

```
1 use azkaban;
2 select * from executors;
```

2) 在执行工作流程时加入useExecutor属性，如下

Execute Flow gmail

Flow View

Notification

Failure Options

Concurrent

Flow Parameters

Flow Property Override

Name	Value
useExecutor	3

Add Row

Add temporary flow parameters that are used to override global settings for each job.

Schedule

CancelExecute

方案二：在Executor所在所有节点部署任务所需脚本和应用。