工作流调度系统Azkaban

第 1 节 概述

1.1 工作流调度系统

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

各任务单元之间存在时间先后及前后依赖关系,为了很好地组织起这样的复杂执行计划，需要一个工作流调度系统来调度任务的执行。

假如，我有这样一个需求，某个业务系统每天产生20G原始数据，每天都要对其进行处理，处理步骤如下所示：

- 通过Hadoop先将原始数据同步到HDFS上；
- 借助MapReduce计算框架对原始数据进行转换，生成的数据以分区表的形式存储到多张Hive表中；
- 需要对Hive中多个表的数据进行JOIN处理，得到一个明细数据Hive大表；
- 将明细数据进行各种统计分析，得到结果报表信息；
- 需要将统计分析得到的结果数据同步到业务系统中，供业务调用使用。

1.2 工作流调度实现方式

- 简单的任务调度
  - 直接使用linux的crontab；
- 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Ooize、Azkaban、Airflow等

1.3 Azkaban与Oozie对比

对市面上最流行的两种调度器，进行对比分析。总体来说，Ooize相比Azkaban是一个重量级的任务调度系统，功能全面，但配置使用也更复杂(xml)。如果可以不在意某些功能的缺失，轻量级调度器Azkaban是很不错的候选对象。

- 功能

  两者均可以调度mapreduce,pig,java,脚本工作流任务

  两者均可以定时执行工作流任务

- 工作流定义

  Azkaban使用Properties文件定义工作流

Oozie使用XML文件定义工作流

- 工作流传参

  Azkaban支持直接传参，例如${input}

  Oozie支持参数和EL表达式，例如${fs:dirSize(myInputDir)}

- 定时执行

  Azkaban的定时执行任务是基于时间的

  Oozie的定时执行任务基于时间和输入数据

- 资源管理

  Azkaban有较严格的权限控制，如用户对工作流进行读/写/执行等操作

  Oozie暂无严格的权限控制

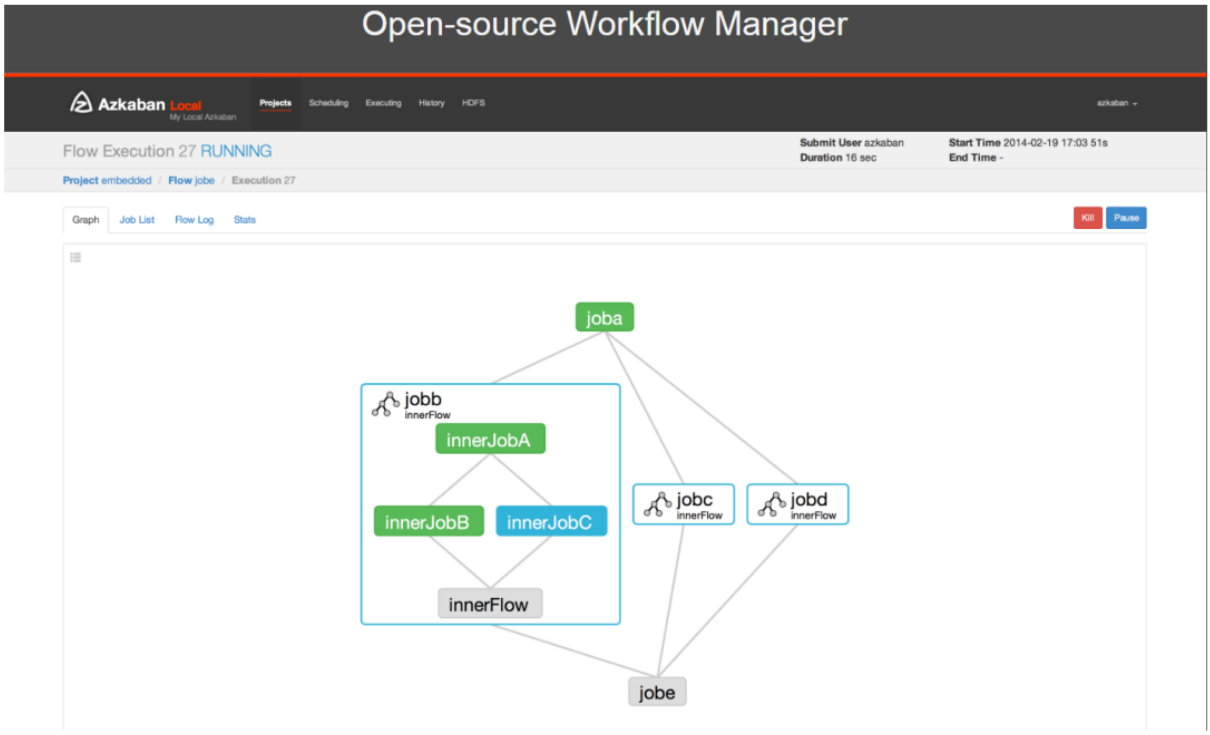
- 工作流执行

  Azkaban有两种运行模式，分别是solo server mode(executor server和web server部署在同一台节点)和multi server mode(executor server和web server可以部署在不同节点)

  Oozie作为工作流服务器运行，支持多用户和多工作流

第 2 节 Azkaban介绍

Azkaban是由linkedin（领英）公司推出的一个批量工作流任务调度器，用于在一个工作流内以一个特定的顺序运行一组工作和流程。Azkaban使用job配置文件建立任务之间的依赖关系，并提供一个易于使用的web用户界面维护和跟踪你的工作流。



Azkaban定义了一种KV文件(properties)格式来建立任务之间的依赖关系，并提供一个易于使用的web用户界面维护和跟踪你的工作流。

有如下功能特点

- Web用户界面
- 方便上传工作流
- 方便设置任务之间的关系
- 调度工作流

架构角色

mysql服务器: 存储元数据，如项目名称、项目描述、项目权限、任务状态、SLA规则等

AzkabanWebServer:对外提供web服务，使用户可以通过web页面管理。职责包括项目管理、权限授权、任务调度、监控executor。

AzkabanExecutorServer:负责具体的工作流的提交、执行。

第 3 节 Azkaban安装部署

3.1 Azkaban的安装准备工作

1 编译

这里选用azkaban3.51.0这个版本自己进行重新编译，编译完成之后得到我们需要的安装包进行安装

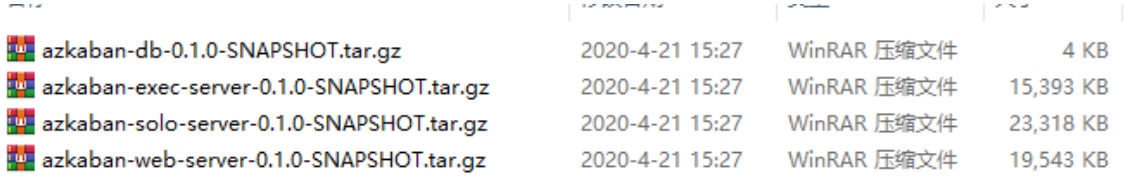
```
cd /opt/lagou/software/

wget https://github.com/azkaban/azkaban/archive/3.51.0.tar.gz

tar -zxvf 3.51.0.tar.gz -C ../servers/

cd /opt/lagou/servers/azkaban-3.51.0/

yum -y install git

yum -y install gcc-c++

./gradlew build installDist -x test
```

Gradle是一个基于Apache Ant和Apache Maven的项目自动化构建工具。-x test 跳过测试。（注意联网下载jar可能会失败、慢）

2 上传编译后的安装文件

在linux122节点创建目录

```
mkdir /opt/lagou/servers/azkaban
```



3.2 solo-server模式部署

1 单服务模式安装

1 解压

azkaban 的solo server使用的是一个单节点的模式来进行启动服务的，只需要一个azkaban-solo-server-0.1.0-SNAPSHOT.tar.gz的安装包即可启动，所有的数据信息都是保存在H2这个azkaban默认的数据当中，

```
tar -zxvf azkaban-solo-server-0.1.0-SNAPSHOT.tar.gz -C ../../servers/azkaban
```

2 修改配置文件

修改时区配置文件

```
cd /opt/lagou/servers/azkaban-solo-server-0.1.0-SNAPSHOT/conf

vim azkaban.properties

default.timezone.id=Asia/Shanghai
```

```
# Azkaban Personalization Settings
azkaban.name=Test
azkaban.label=My Local Azkaban
azkaban.color=#FF3601
azkaban.default.servlet.path=/index
web.resource.dir=web/
default.timezone.id=Asia/Shanghai
# Azkaban UserManager class
user.manager.class=azkaban.user.XmlUserManager
user.manager.xml.file=conf/azkaban-users.xml
# Loader for projects
executor.global.properties=conf/global.properties
azkaban.project.dir=projects
database.type=h2
h2.path=./h2
h2.create.tables=true
# Velocity dev mode
velocity.dev.mode=false
# Azkaban Jetty server properties.
jetty.use.ssl=false
jetty.maxThreads=25
jetty.port=8081
# Azkaban Executor settings
executor.port=12321
# mail settings
mail.sender=
mail.host=
# User facing web server configurations used to construct the us
ban web servers and users.
# enduser -> myazkabanhost:443 -> proxy -> localhost:8081
# when this parameters set then these parameters are used to ger
# if these parameters are not set then jetty.hostname, and jetty
# azkaban.webserver.external_hostname=myazkabanhost.com
# azkaban.webserver.external_ssl_port=443
# azkaban.webserver.external_port=8081
job.failure.email=
job.success.email=
```

修改commonprivate.properties配置文件

```
cd /opt/lagou/servers/azkaban-solo-server-0.1.0-SNAPSHOT/plugins/jobtypes

vim commonprivate.properties

execute.as.user=false
memCheck.enabled=false
```

azkaban默认需要3G的内存，剩余内存不足则会报异常。

3 启动solo-server

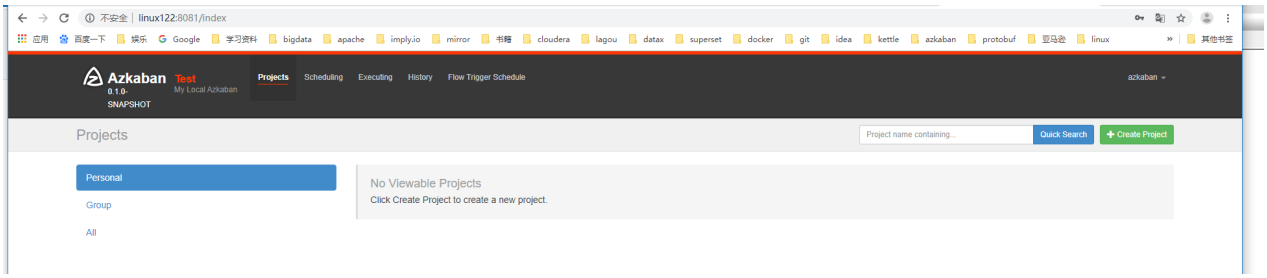
```
cd  /opt/lagou/servers/azkaban-solo-server-0.1.0-SNAPSHOT

bin/start-solo.sh
```

```
[root@linux122 azkaban-solo-server-0.1.0-SNAPSHOT]# jps
10560 NodeManager
7461 QuorumPeerMain
16646 AzkabanSingleServer
16666 Jps
7647 DataNode
[root@linux122 azkaban-solo-server-0.1.0-SNAPSHOT]#
```

4 浏览器页面访问

浏览器页面访问

```
http://linux122:8081/index
```



登录信息

```
用户名：azkaban
密码：  azkaban
```

2 单服务模式使用

需求：使用azkaban调度我们的shell脚本，执行linux的shell命令

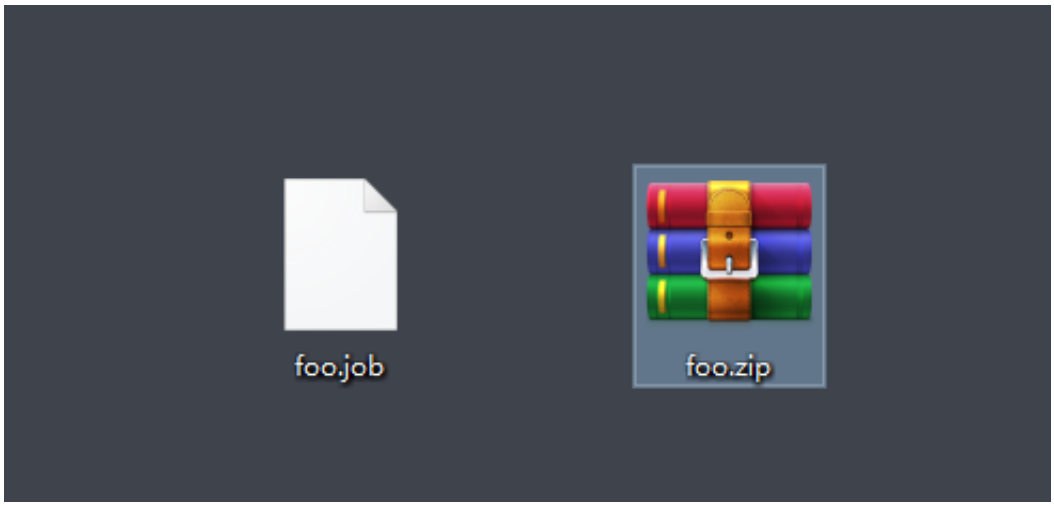
具体步骤

开发job文件

创建普通文本文件 foo.job，文件内容如下

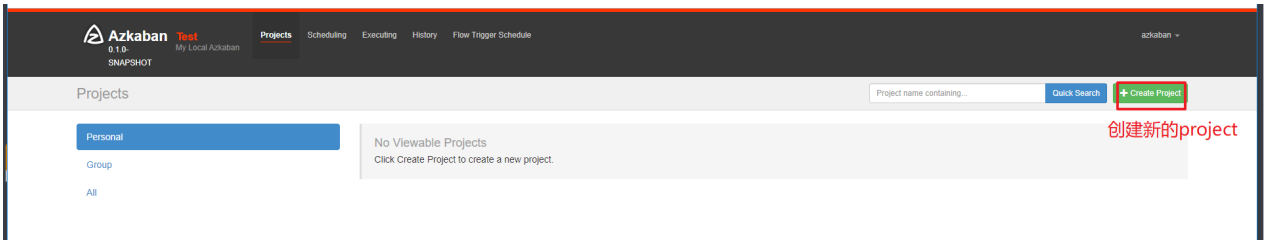
```
type=command
command=echo 'hello world'
```

打成压缩包

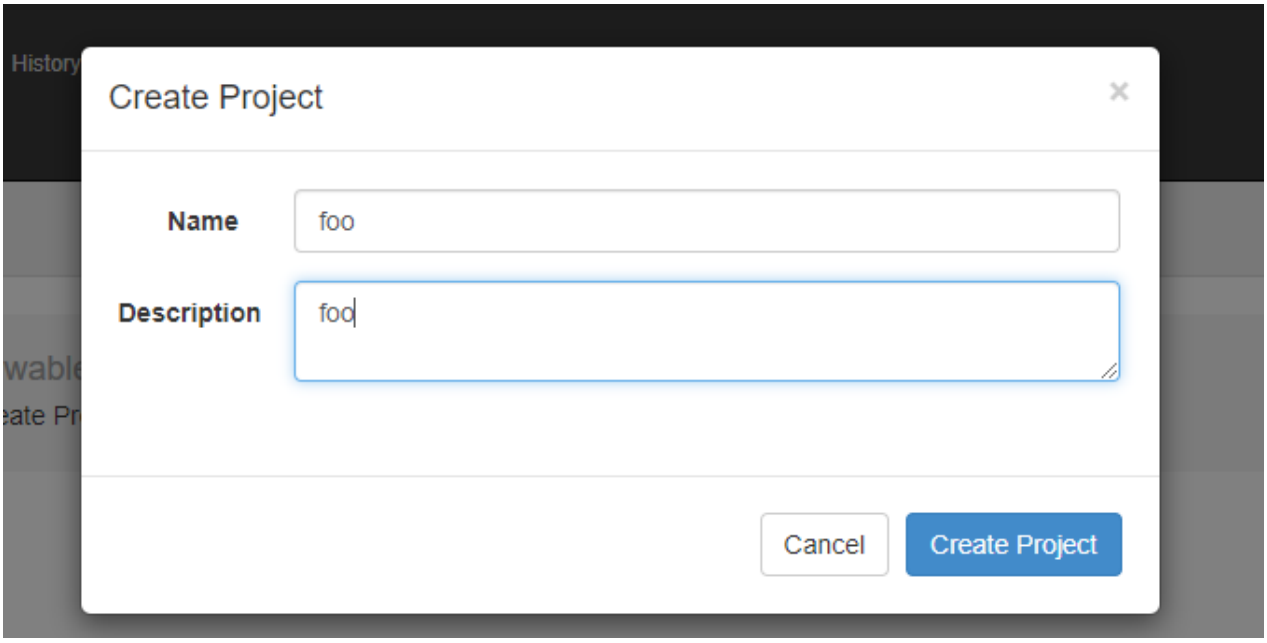


上传压缩包到Azkaban

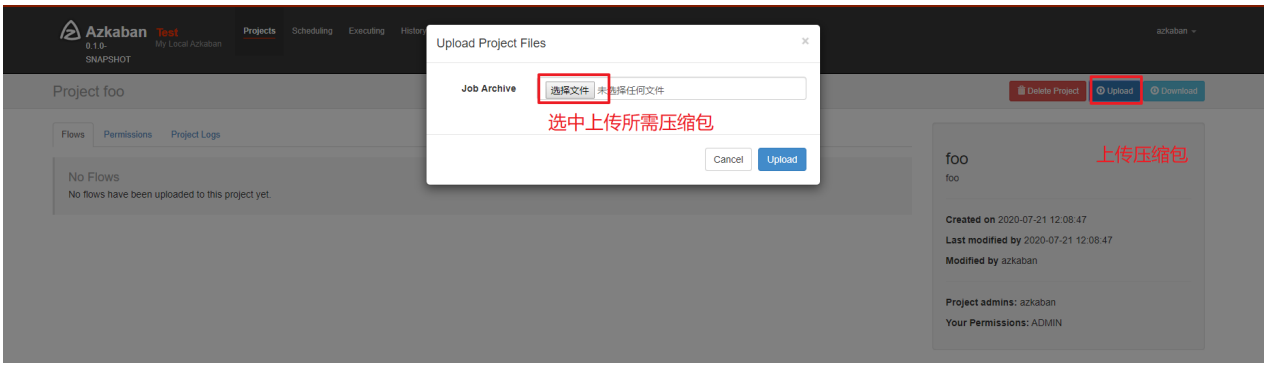
创建project

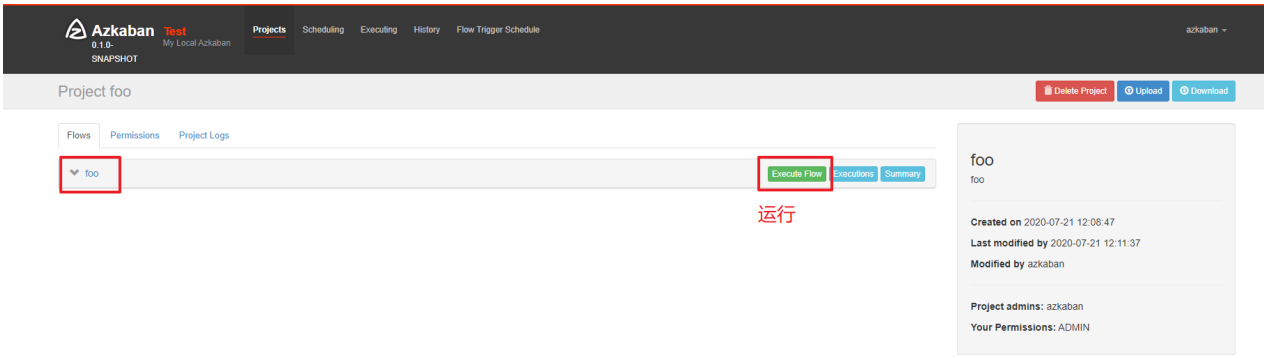


指定project名称和描述信息



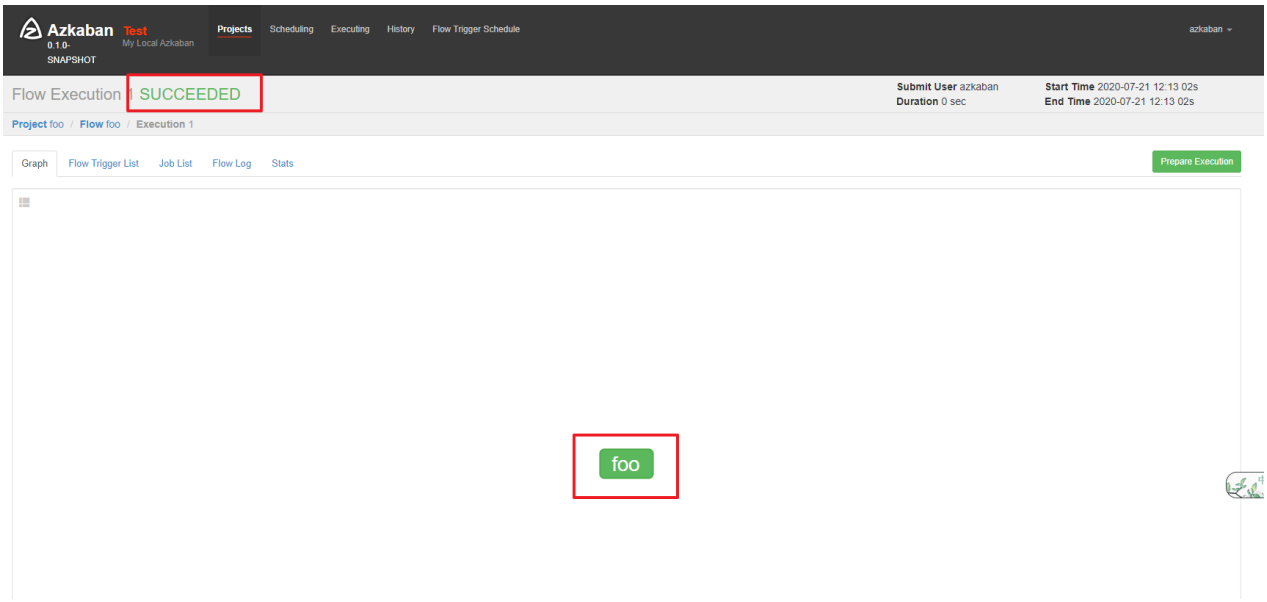
Azkaban上传我们的压缩包





查看工作流计划并执行

运行结果页面



停止程序

```
bin/shutdown-solo.sh
```

3.3 multiple-executor模式部署

1 安装所需软件

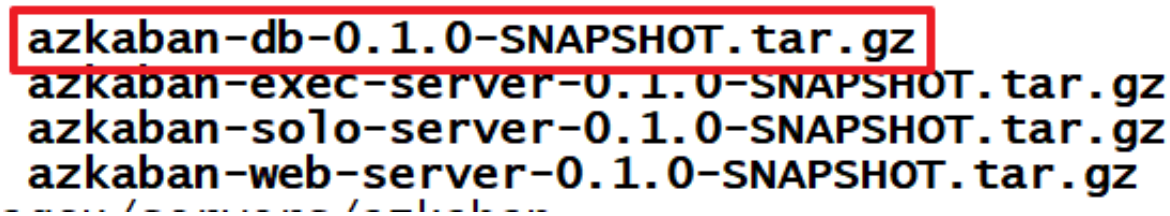
Azkaban Web服务安装包

azkaban-web-server-0.1.0-SNAPSHOT.tar.gz

Azkaban执行服务安装包

azkaban-exec-server-0.1.0-SNAPSHOT.tar.gz

sql脚本



节点规划

HOST	角色
linux123	mysql,exec-server
linux122	web-server
linux121	exec-server

2 数据库准备

linux123

进入mysql的客户端执行以下命令

```
mysql -uroot -p
```

执行以下命令：

```
SET GLOBAL validate_password_length=5;

SET GLOBAL validate_password_policy=0;

CREATE USER 'azkaban'@'%' IDENTIFIED BY 'azkaban';

GRANT all privileges ON azkaban.* to 'azkaban'@'%' identified by 'azkaban'
WITH GRANT OPTION;
CREATE DATABASE azkaban;

use azkaban;
#解压数据库脚本
tar -zxvf azkaban-db-0.1.0-SNAPSHOT.tar.gz -C /opt/lagou/servers/azkaban
source /opt/lagou/servers/azkaban/azkaban-db-0.1.0-SNAPSHOT/create-all-sql-
0.1.0-SNAPSHOT.sql; #加载初始化sql创建表
```

3 配置Azkaban-web-server

进入linux122节点

解压azkaban-web-server

```
mkdir /opt/lagou/servers/azkaban
tar -zxvf azkaban-web-server-0.1.0-SNAPSHOT.tar.gz –C
/opt/lagou/servers/azkaban/
```

进入azkaban-web-server根目录下

```
cd /opt/lagou/servers/azkaban/azkaban-web-server-0.1.0-SNAPSHOT
#生成ssl证书：
keytool -keystore keystore -alias jetty -genkey -keyalg RSA
```

```
[root@linux122 azkaban-web-server-3.47.0]# keytool -keystore keystore -alias jetty -genkey -keyalg RSA
Enter keystore password:  输入密码全部为azkaban
Re-enter new password:
What is your first and last name?
  [Unknown]:
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
  [no]:  y

Enter key password for <jetty>
        (RETURN if same as keystore password):
Re-enter new password:

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an industry standard format using "keytool -importkey
store -srckeystore keystore -destkeystore keystore -deststoretype pkcs12".
[root@linux122 azkaban-web-server-3.47.0]#
```

可以不用输入直接回车

输入y确认上面输入的信息

继续输入密码为azkaban

注意：运行此命令后,会提示输入当前生成keystore的密码及相应信息,输入的密码请记住(**所有密码统一以azkaban输入**)

```
[root@linux122 azkaban-web-server-0.1.0-SNAPSHOT]# ll
total 8
drwxr-xr-x 3 root root   65 May 29  2019 bin
drwxr-xr-x 2 root root  106 May 29  2019 conf
-rw-r--r-- 1 root root 2242 Jul 21 04:00 keystore
drwxr-xr-x 2 root root 4096 May 29  2019 lib
drwxr-xr-x 6 root root   73 May 29  2019 web
[root@linux122 azkaban-web-server-0.1.0-SNAPSHOT]#
```

修改 azkaban-web-server的配置文件

```
cd /opt/lagou/servers/azkaban-web-server-3.51.0/conf

vim azkaban.properties



# Azkaban Personalization Settings
azkaban.name=Test
azkaban.label=My Local Azkaban
azkaban.color=#FF3601
azkaban.default.servlet.path=/index
web.resource.dir=web/
default.timezone.id=Asia/Shanghai # 时区注意后面不要有空格

# Azkaban UserManager class
user.manager.class=azkaban.user.XmlUserManager
user.manager.xml.file=conf/azkaban-users.xml

# Azkaban Jetty server properties. 开启使用ssl 并且知道端口
jetty.use.ssl=true
jetty.port=8443
jetty.maxThreads=25
```

```
#  KeyStore for SSL ssl相关配置  注意密码和证书路径
jetty.keystore=keystore
jetty.password=azkaban
jetty.keypassword=azkaban
jetty.truststore=keystore
jetty.trustpassword=azkaban

# Azkaban mysql settings by default. Users should configure their own username
and password.
database.type=mysql
mysql.port=3306
mysql.host=linux123
mysql.database=azkaban
mysql.user=root
mysql.password=12345678
mysql.numconnections=100

#Multiple Executor 设置为false
azkaban.use.multiple.executors=true
#azkaban.executorselector.filters=StaticRemainingFlowSize,MinimumFreeMemory,Cp
uStatus
azkaban.executorselector.comparator.NumberOfAssignedFlowComparator=1
azkaban.executorselector.comparator.Memory=1
azkaban.executorselector.comparator.LastDispatched=1
azkaban.executorselector.comparator.CpuUsage=1
```

添加属性

```
mkdir -p plugins/jobtypes
cd plugins/jobtypes/
vim commonprivate.properties
```

```
azkaban.native.lib=false
execute.as.user=false
memCheck.enabled=false
```

4 配置Azkaban-exec-server

linux123节点，上传exec安装包到/opt/lagou/software

```
-rw-r--r--  1 root root  232734895 Jun 13 04:16 apache-hive-2.3.7-bin.tar.gz
-rw-r--r--  1 root root   15762323 Apr 21 03:27 azkaban-exec-server-0.1.0-SNAPSHOT.tar.gz
```

```
tar -zxvf azkaban-exec-server-0.1.0-SNAPSHOT.tar.gz –C
/opt/lagou/servers/azkaban/
```

修改azkaban-exec-server的配置文件

```
cd /opt/lagou/servers/azkaban-exec-server-3.51.0/conf

vim azkaban.properties
```

```
# Azkaban Personalization Settings
azkaban.name=Test
azkaban.label=My Local Azkaban
azkaban.color=#FF3601
azkaban.default.servlet.path=/index
web.resource.dir=web/
default.timezone.id=Asia/Shanghai

# Azkaban UserManager class
user.manager.class=azkaban.user.XmlUserManager
user.manager.xml.file=conf/azkaban-users.xml

# Loader for projects
executor.global.properties=conf/global.properties
azkaban.project.dir=projects

# Where the Azkaban web server is located
azkaban.webserver.url=https://linux122:8443

# Azkaban mysql settings by default. Users should configure their own username
and password.
database.type=mysql
mysql.port=3306
mysql.host=linux123
mysql.database=azkaban
mysql.user=root
mysql.password=12345678
mysql.numconnections=100

# Azkaban Executor settings
executor.maxThreads=50
executor.port=12321
executor.flow.threads=30
```

分发exec-server到linux121节点

```
cd /opt/lagou/servers
scp -r azkaban linux121:$PWD
```

5 启动服务

先启动exec-server

再启动web-server

```
#启动exec-server
bin/start-exec.sh
#启动web-server
bin/start-web.sh
```

激活我们的exec-server

启动webServer之后进程失败消失，可通过安装包根目录下对应启动日志进行排查。

```
    for the 12th parameter of azkaban.webapp.AzkabanWebServer.<init>(AzkabanWebServer.java:165)
  at azkaban.webapp.AzkabanWebServer.class(AzkabanWebServer.java:122)
  while locating azkaban.webapp.AzkabanWebServer
Caused by: azkaban.executor.ExecutorManagerException: No active executor found
        at azkaban.executor.ExecutorManager.setupExecutors(ExecutorManager.java:253)
        at azkaban.executor.ExecutorManager.<init>(ExecutorManager.java:131)
        at azkaban.executor.ExecutorManager$$FastClassByGuice$$e1c1dfed.newInstance(<generated>)
        at com.google.inject.internal.DefaultConstructionProxyFactory$FastClassProxy.newInstance(DefaultConst
        at com.google.inject.internal.ConstructorInjector.provision(ConstructorInjector.java:111)
        at com.google.inject.internal.ConstructorInjector.construct(ConstructorInjector.java:90)
        at com.google.inject.internal.ConstructorBindingImpl$Factory.get(ConstructorBindingImpl.java:268)
```

需要手动激活executor

```
cd /opt/lagou/servers/azkaban/azkaban-exec-server-0.1.0-SNAPSHOT
curl -G "linux123:$(<./executor.port)/executor?action=activate" && echo
```

访问地址：

https://linux122:8443

第 4 节 Azkaban使用

1 shell command调度

创建job描述文件

vi command.job

command.job

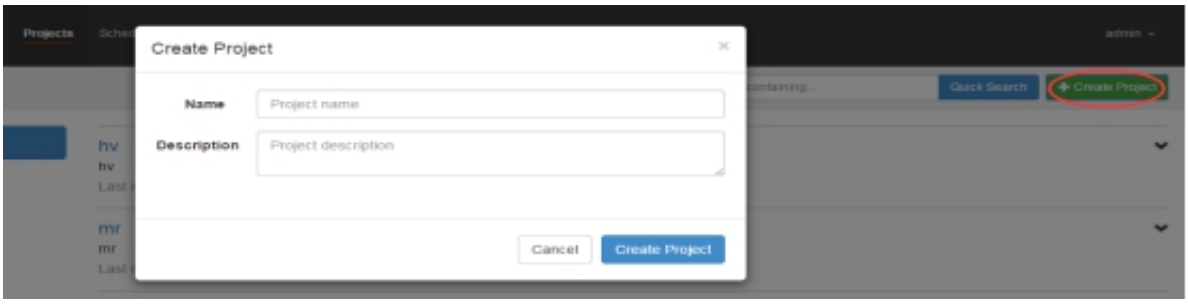
```
type=command
command=echo 'hello'
```

将job资源文件打包成zip文件

zip command.job

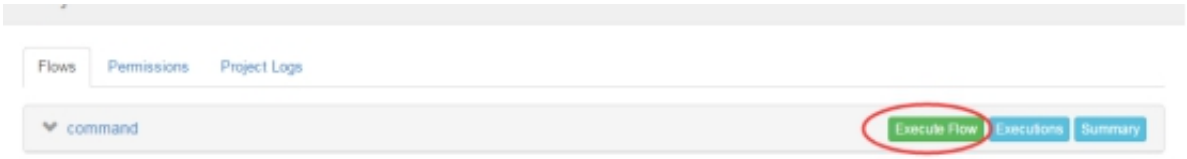
通过azkaban的web管理平台创建project并上传job压缩包

首先创建Project



上传zip包

启动执行该job



2 job依赖调度

创建有依赖关系的多个job描述

第一个job：foo.job

foo.job

```
type=command
command=echo 'foo'
```

第二个job：bar.job依赖foo.job

bar.job

```
type=command
dependencies=foo
command=echo 'bar'
```

将所有job资源文件打到一个zip包中

在azkaban的web管理界面创建工程并上传zip包

启动工作流flow

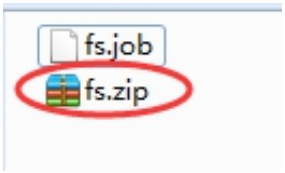
3 HDFS任务调度

创建job描述文件

fs.job

```
type=command
command=/opt/lagou/servers/hadoop-2.9.2/bin/hadoop fs -mkdir /azkaban
```

将job资源文件打包成zip文件



通过azkaban的web管理平台创建project并上传job压缩包

启动执行该job

4 MAPREDUCE任务调度

mr任务依然可以使用command的job类型来执行

创建job描述文件，及mr程序jar包（示例中直接使用hadoop自带的example jar）

mrwc.job

```
type=command
command=/opt/lagou/servers/hadoop-2.9.2/bin/hadoop jar hadoop-mapreduce-
examples-2.9.2.jar wordcount /wordcount/input /wordcount/azout
```

将所有job资源文件打到一个zip包中

在azkaban的web管理界面创建工程并上传zip包

启动job

遇到虚拟机内存不足情况：

1. 增大机器内存
2. 使用清除系统缓存命令，暂时释放一些内存

```
[root@linux123 mapreduce]# echo 1 >/proc/sys/vm/drop_caches
[root@linux123 mapreduce]# echo 2 >/proc/sys/vm/drop_caches
[root@linux123 mapreduce]# echo 3 >/proc/sys/vm/drop_caches
```

5 HIVE脚本任务调度

创建job描述文件和hive脚本

Hive脚本： test.sql

```
use default;
drop table aztest;
create table aztest(id int,name string) row format delimited fields terminated
by ',';
```

Job描述文件：hivef.job

hivef.job

```
type=command
command=/opt/lagou/servers/hive-2.3.7/bin/hive -f 'test.sql'
```

将所有job资源文件打到一个zip包中创建工程并上传zip包,启动job

6 定时任务调度

除了手动立即执行工作流任务外，azkaban也支持配置定时任务调度。开启方式如下：

首页选择待处理的project

选择左边schedule表示配置定时调度信息，选择右边execute表示立即执行工作流任务。

Schedule Flow Options

*All schedules are basead 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	*
Hours	*
Day of Month	?
Month	*
Day of Week	*
Year	

0 * * ? * *

Reset

Special Characters:

*	any value
,	value list separators
-	range of values
/	step values

Detailed instructions.

Next 10 scheduled executions for this cron expression only:

- 2020-04-22T11:55:00
- 2020-04-22T11:56:00
- 2020-04-22T11:57:00
- 2020-04-22T11:58:00
- 2020-04-22T11:59:00
- 2020-04-22T12:00:00
- 2020-04-22T12:01:00
- 2020-04-22T12:02:00
- 2020-04-22T12:03:00
- 2020-04-22T12:04:00