1.下载 azkaban3.74.3版本

 https://github.com/azkaban/azkaban/releases

1. 编译azkaban

cd ./azkaban 将上面的文件解压到文件夹azkaban里面，然后进入这个目录。执行以下命令

./gradlew clean

*# Build Azkaban*

./gradlew build

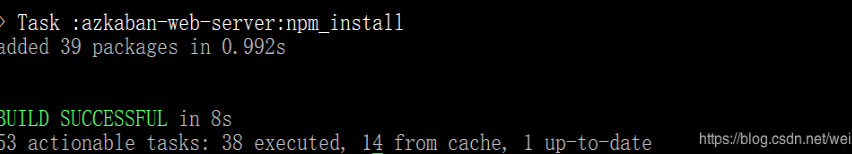
*# Clean the build*

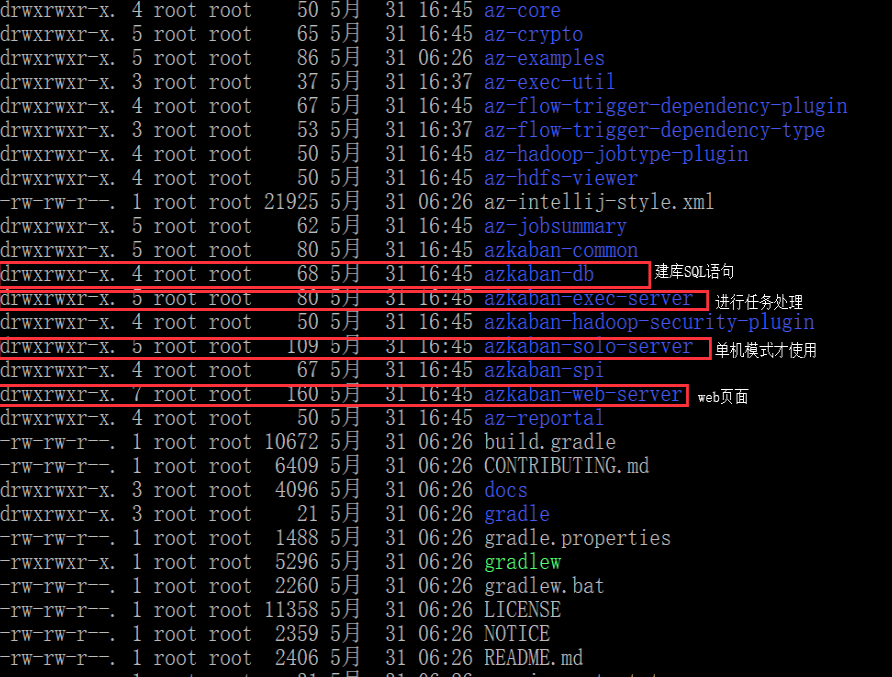
./gradlew clean

*# Build and install distributions*

./gradlew build installDist -x test

如果编译成功





当编译完成后安装包就已经整好了，我们只需要取出上面4个框内tar.gz文件即可,这4个压缩包为在各自文件夹下/build/distributions里的tar.gz

最主要的包是web-server,exec-server两个。Exec-server是执行服务，可以有多个。

Web-server是作业管理服务只有一个

安装可以参考：https://blog.csdn.net/weixin\_42179685/article/details/90716366

# mysql数据库

安装mysql数据库后建立一个azkaban库，然后导入建表sql命令source   db文件夹里的 create-all 的sql文件

create database azkaban;

use azkaban ;

source /soft/azkaban/azkaban-db/create-all-sql-0.1.0-SNAPSHOT.sql ;

# 4、设置conf配置参数

我们需要把azkaban-exec-server和azkaban-web-server的azkaban.properties内参数调好即可完成配置（azkaban.properties都在各自文件夹的/conf 里），具体每个字段的含义自行百度

Web-server azkaban.properties配置内容：

# Azkaban Personalization Settings

azkaban.name=HY

azkaban.label=MyAzkaban

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

# Velocity dev mode

velocity.dev.mode=false

# Azkaban Jetty server properties.

jetty.use.ssl=false

jetty.maxThreads=25

jetty.port=8081

# Azkaban Executor settings

# mail settings

# mail settings

mail.sender=443435766@163.com

mail.host=smtp.163.com

mail.user=443435766@163.com

mail.password=xxxxxx

# User facing web server configurations used to construct the user facing server URLs. They are useful when there is a reverse proxy between Azkaban web servers and users.

# enduser -> myazkabanhost:443 -> proxy -> localhost:8081

# when this parameters set then these parameters are used to generate email links.

# if these parameters are not set then jetty.hostname, and jetty.port(if ssl configured jetty.ssl.port) are used.

# azkaban.webserver.external\_hostname=myazkabanhost.com

# azkaban.webserver.external\_ssl\_port=443

# azkaban.webserver.external\_port=8081

job.failure.email=

job.success.email=

lockdown.create.projects=false

cache.directory=cache

# JMX stats

jetty.connector.stats=true

executor.connector.stats=true

# Azkaban mysql settings by default. Users should configure their own username and password.

database.type=mysql

mysql.port=3306

mysql.host=10.12.5.37

mysql.database=azkaban

mysql.user=dps

mysql.password=dps1234

mysql.numconnections=100

#Multiple Executor

azkaban.use.multiple.executors=true

azkaban.executorselector.filters=StaticRemainingFlowSize,CpuStatus

azkaban.executorselector.comparator.NumberOfAssignedFlowComparator=1

azkaban.executorselector.comparator.Memory=1

azkaban.executorselector.comparator.LastDispatched=1

azkaban.executorselector.comparator.CpuUsage=1

Exec-server azkaban.properties配置内容：

default.timezone.id=Asia/Shanghai

executor.global.properties=conf/global.properties

azkaban.project.dir=projects

# Azkaban Executor settings

executor.port=12321

# lockdown.create.projects=false

# cache.directory=cache

# JMX stats

jetty.connector.stats=true

executor.connector.stats=true

# Azkaban plugin settings

azkaban.jobtype.plugin.dir=plugins/jobtypes

# Azkaban mysql settings by default. Users should configure their own username and password.

database.type=mysql

mysql.port=3306

mysql.host=10.12.5.37

mysql.database=azkaban

mysql.user=dps

mysql.password=dps1234

mysql.numconnections=100

# Azkaban Executor settings

executor.maxThreads=10

executor.flow.threads=10

# 5、添加账号

我们只需要在/azkaban-web-server/conf/ azkaban-users.xml文件里添加即可。假设我添加了一个yasina账号密码为yasina并有admin权限。只需如下修改即可

<azkaban-users>

<user groups="azkaban" password="azkaban" roles="admin" username="azkaban"/>

<user password="metrics" roles="metrics" username="metrics"/>

<user password="yasina" roles="metrics,admin" username="yasina"/>

<role name="admin" permissions="ADMIN"/>

<role name="metrics" permissions="METRICS"/>

</azkaban-users>

# 6、启动Azkaban

1.启动顺序：

首先启动exec的再启动web的。一定要按这样的顺序，不然一定无法运行

2.启动exec-server

cd /soft/azkaban/azkaban-exec-server/

./bin/start-exec.sh

3.手动修改exec-srver状态为激活。

  这步很重要，不然后面web端会一直起不来，他会报错说execute没有available的，然后就停了。很蛋疼但是没办法。端口号在executor下的executor.port文件中可查  具体命令为curl http://${executorHost}:${executorPort}/executor?action=activate

1. 在azkabans中已经写好来的azkaban重启和激活的命令

重启 azrun.py r all 重启所有web 和 exec

激活节点 azrun.py a all 激活所有节点

# Azkabans

本项目主要用于azkaban远程发布项目，定时发布项目等辅助功能。

1、所有的作业都是shell命令执行，这样使得作业配置更简单

2、减少job文件的配置，所有job会根据配置文件自动生成，校验依赖作业，并自动打包上传,自动加入定时任务

3、把azkaban当中任务执行和查看的介质，任务配置以批量配置的方式，减少工作量、方便管理项目

4、azkaban的项目发布需把整个项目都重新发布，这个比较麻烦还容易出错。借助azkabans更容易管理

7.1、azkaban目录

1、登录操作（cookies.py）、项目操作azkaban.py

2、cron.py是对时间的解析

3、propertie是对参数文件读取

4、config.py对conf目录下的config.ini文件读取

5、utils,py常用函数库

7.2、backup目录

用于备份项目的历史zip文件。

上传新的zip文件之前先把老的版本下载下来本分到此文件夹下

也可以单独指定备份目录. eg: config.inibackup\_path=/home/xzh/azkaban/backup

7.3、conf目录

7.3.1、全局配置文件config.ini

[base]项是必须项,这里的配置项也会作为全局变量输出到system.properties文件中

# 定义base分组，必须配置

[base]

azkaban\_url=http://broker2:8081 # 你的azkaban登录的url azkaban有的限制域名访问，需要设置或改成IP

login\_user=testAdmin # azkaban 登录用户名

login\_pwd=testAdmin # azkaban 登录密码

check\_interval=600 # 休息间隔时间,任务执行失败从新尝试等待时间 循环监控时间

# mysql配置

mysql\_host=broker2

mysql\_port=3306

mysql\_user=xxxx

mysql\_pwd=xxxx

mysql\_db=azkaban

# ETL文件目录

etl\_path=/home/xzh/dps/etl/

7.3.2、单个项目配置，如dw是项目名为dw的配置文件夹

7.3.2.1、dw.csv主要配置项目job依赖和job 命令

以下是个csv文件文件, （job\_nm是个job名称;dependencies是指项目依赖job,多个以逗号或者空格分开 ;command 是该job的执行命令）

job\_nm , dependencies , command (表头)

start , ,

el\_crm , start , sh scripts/testa.sh

el\_hyf , start , python -V

dim\_city , "el\_crm,el\_hyf" , sh scripts/testb.sh ${azkaban.job.id}

dim\_mon , dim\_city , echo ${etl\_dtm}

mytest , dim\_mon , sh scripts/testb.sh ${azkaban.job.id}

end , mytest ,

解释：dim\_city 依赖两个job el\_crm和el\_hyf，执行命令 是sh scripts/testb.sh ${azkaban.job.id}

7.3.2.2、\*.job文件是对特殊文件处理，最终被复制到发布的项目里面去

特殊的job文件没办法通过.csv文件简单配置，可以单独编辑一个.job文件。

项目生成后该job文件会把生成的简单job文件覆盖掉

7.3.2.3、dw.properties项目参数配置，最终会被输出到system.properties

7.3.2.4、scripts 存放脚本，最终也会被复制到项目里面去

7.4、temp目录

项目生成的临时文件夹，例如dw是用来存放项目dw的临时文件

（包含job文件、脚本文件等）,最后会被打包成dw.zip文件,然后远程上传项目

也可以单独指定备份目录. eg: config.ini temp\_path=/home/xzh/azkaban/temp

7.5 azrun.py是命令工具文件

/xuzh/azkaban/azkabans/azrun.py -h

usage: azrun.py [-h] [-f FLOWS] [-t CRONTAB] [-i IGNORE] [-p PARAM]

{u,e,s,a} prj\_nm

远程部署azkaban

positional arguments:

{u,e,s,a} u 表示更新项目元数据， e 表示执行项目 s 表示给项目添加执行计划,a 激活执行节点, r重启azkaban服务

prj\_nm 项目名称字符型

optional arguments:

-h, --help show this help message and exit

-f FLOWS, --flows FLOWS

工作流的字符列表,eg: "['a','b']" 设置后值执行这个列表里面的工作流

-t CRONTAB, --crontab CRONTAB

cron的定时器格式字符串 0 5 0/5 \* \* \*

-i IGNORE, --ignore IGNORE

job name的字符列表 eg."['a','b']" 设置后跳过

-p PARAM, --param PARAM

参数传入，数据字典,可以覆盖全局参数 "{'s':1}" 设置后执行flow时会覆盖系统参数

# 8、项目部署

8.1、拷贝azkabans项目到本地

8.2、配置全局参数文件conf/config.ini

配置完成后，可以执行aztest 中 test\_login（）函数，测试配置情况

8.3、开始配置项目mydw

8.3.1 在conf创建项目文件夹mydw

8.3.2 创建 mydw.csv，并开始配置job的命令和依赖（conf/mydw目录下）

8.3.3 配置本项目参数mydw.properties文件（conf/mydw目录下）

# crom 表示配置定时器（【必须配置】）com.prj\_nm表示给项目所有的flow统一设置定时器，crom.flowid表示给flowid设置定时器（会覆盖com.prj\_nm配置）

cron.mydw=0 0/20 \* \* \* ? # 项目定时器

cron.endflow=0 0/30 \* \* \* ? # id 为 endflow 的flow的定时器 会覆盖cron.mydw定时器

etl\_ts=${azkaban.flow.start.year}-${azkaban.flow.start.month}-${azkaban.flow.start.day}

8.3.4 azrun.py u mydw 这样就会把mydw项目上传到azkaban 并会自动添加定时

8.3.5 azrun.py s mydw 给mydw加定时任务

8.3.6 azrun.py e mydw "{'useExecutor':2}" 执行mydw所有工作流，并且指定useExecutor

8.4、aztest.py 用于开发测试，可以随意修改