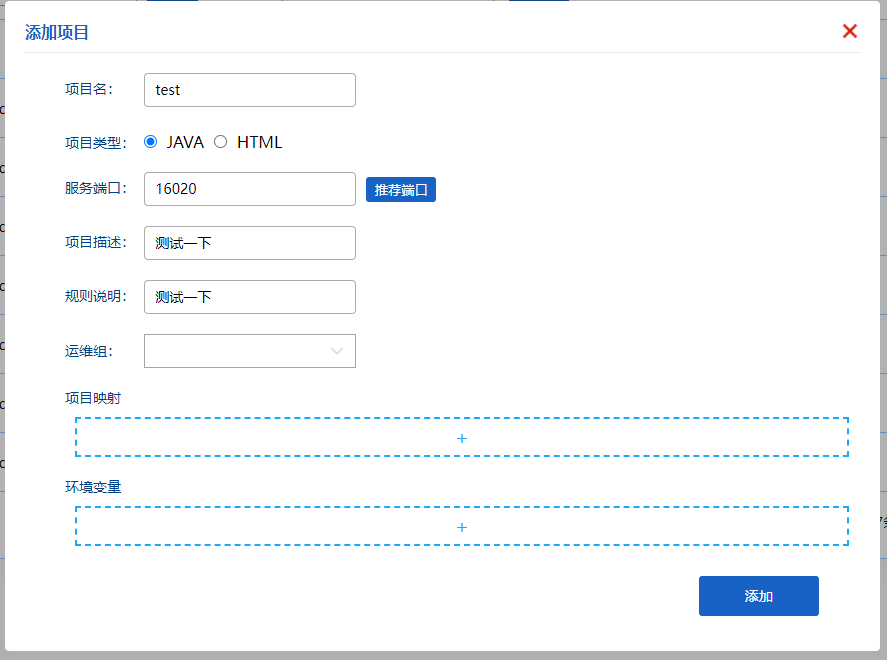
**1.创建项目**

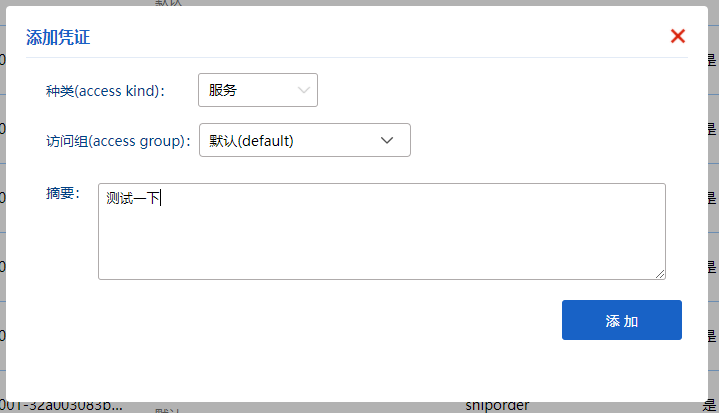
地址:/devops/project



注：项目名与maven项目的finalName保持一致

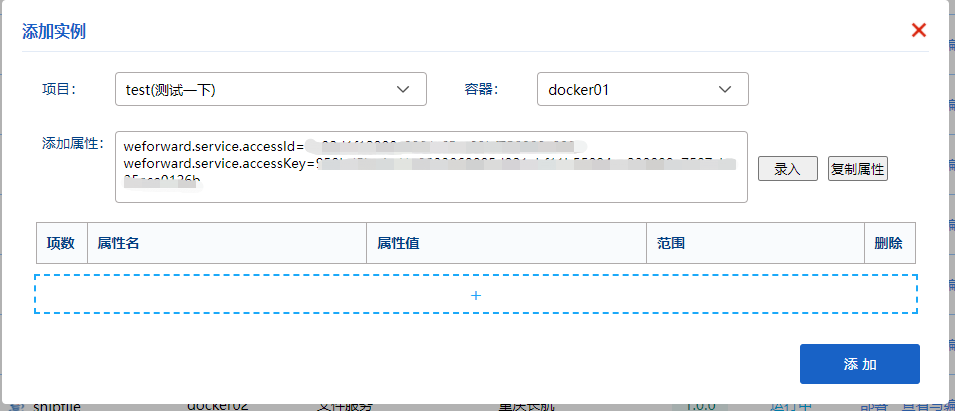
**2.访问凭证**

地址:/devops/visitevidence



**3.添加实例**

地址：/devops/runningexample



注：其中的accessId和accessKey的值为步骤2创建的

**4.使用云日志（可选）**

提前：使用logback日志

在src/main/resources新建一个文件logback.xml

内容:

<?xml version="1.0" encoding="UTF-8"?>

<!-- For assistance related to logback-translator or configuration -->

<!-- files in general, please contact the logback user mailing list -->

<!-- at http://www.qos.ch/mailman/listinfo/logback-user -->

<!-- -->

<!-- For professional support please see -->

<!-- http://www.qos.ch/shop/products/professionalSupport -->

<!-- -->

<!DOCTYPE configuration>

<configuration>

<appender name="remote" class="cn.weforward.common.log.DelayRemoteAppender">

<url>${rlog.url}</url>

<projectName>${project\_name}</projectName>

<serverid>${weforward.serverid}</serverid>

<userName>${weforward.service.accessId}</userName>

<password>${weforward.service.accessKey}</password>

<layout class="ch.qos.logback.classic.PatternLayout">

<pattern>%d{HH:mm:ss.SSS}[%thread]%-5level %logger{50} %msg%n

</pattern>

</layout>

</appender>

<root>

<level value="INFO" />

<appender-ref ref="remote" />

</root>

</configuration>

为了本机调试方便，还可再增加一个文件logback-test.xml

内容：

<?xml version="1.0" encoding="UTF-8"?>

<!-- For assistance related to logback-translator or configuration -->

<!-- files in general, please contact the logback user mailing list -->

<!-- at http://www.qos.ch/mailman/listinfo/logback-user -->

<!-- -->

<!-- For professional support please see -->

<!-- http://www.qos.ch/shop/products/professionalSupport -->

<!-- -->

<!DOCTYPE configuration>

<configuration>

<appender name="stdout"

class="ch.qos.logback.core.ConsoleAppender">

<encoder

class="ch.qos.logback.classic.encoder.PatternLayoutEncoder">

<Pattern>%d{HH:mm:ss.SSS}[%thread]%-5level %logger{50} %msg%n

</Pattern>

</encoder>

</appender>

<logger name="cn.weforward.framework.ext.RpcEndPoint" level="DEBUG" />

<logger name="com.ly.traffic.ground.boat.ticket.shipfinance.impl.MongoAsset" level="DEBUG" />

<root>

<level value="INFO" />

<appender-ref ref="stdout" />

</root>

</configuration>

将日志输出到控制台

查看日志（需要部署应用后）

地址:/devops/runningexample

找到步骤3创建的实例，点击实例日期

**5.使用云配置（可选）**

**1.引入weforward-framewrok**

<dependency>

<groupId>cn.weforward</groupId>

<artifactId>weforward-framework</artifactId>

<version>1.2.7</version>

</dependency>

**2．增加配置**

在/src/main/resources 目录下新建文件

weforward-test.properties

内容

project\_name=[项目名]
weforward.apiUrl= http://wfship.qa.17u.cn/
weforward.service.accessId=[访问凭证id]
weforward.service.accessKey=[访问凭证key]

注：其中项目名为步骤1创建的，accessId和accessKey的值为步骤2创建的

**3．配置CloudPropertyPlaceholderConfigurer**

XML方式

<!-- 云配置 -->
<bean class=cn.weforward.boot.CloudPropertyPlaceholderConfigurer/>

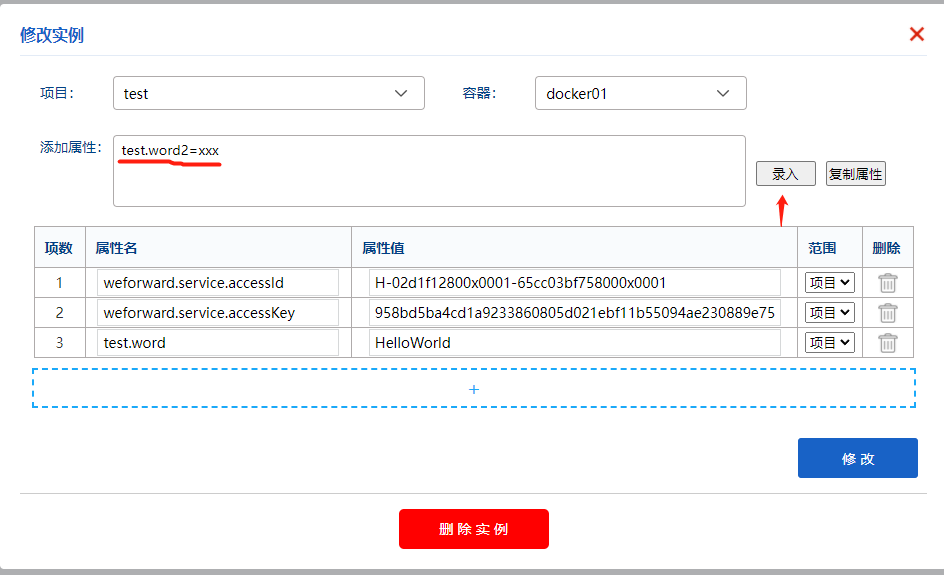
Annotation方式

@Bean
static CloudPropertyPlaceholderConfigurer cloudProperty() {
       return new CloudPropertyPlaceholderConfigurer();
}

**４．添加配置**

地址:/devops/runningexample

找到步骤3创建的实例，点击查看与编辑



**6.使用服务端口（可选）**

步骤1创建的项目中，有一个服务端口，该端口会在启动应用行注入

形式为-Dweforward.port=xxx

所以，项目中要使用这个端口，只需读取weforward.port属性即可

如在application.yml文件中加入

server:

port: ${weforward.port:8080}

**7.构建上传jar包**

**1.引入插件**

<plugin>

<groupId>cn.weforward</groupId>

<artifactId>build-maven-plugin</artifactId>

<version>1.2.2</version>

<configuration>

<mainVersion>1.0</mainVersion>

<outputDirectory>bin</outputDirectory>

<fatjar>true</fatjar>

<autoupload>true</autoupload>

<mainClass>cn.weforward.boot.SpringXmlApp</mainClass>

<excludes>

<exclude>script/\*\*</exclude>

<exclude>commons-logging.properties</exclude>

<exclude>simplelogger.properties</exclude>

<exclude>weforward-dev.properties</exclude>

<exclude>weforward-test.properties</exclude>

<exclude>logback-test.xml</exclude>

</excludes>

</configuration>

</plugin>

其中mainClass需要修改为程序的主类

**2.修改finalName**

将finalName修改成与步骤1的项目名一致

<finalName>test</finalName>

**3.修改version**

版本规则为

主版本号+次版本+序列号

主版本号一般为1.0，项目有重大调整时递增第二位，整体重构时递增第一位

次版本号从0开始递增，项目正常迭代时递增

序列号可使用日期，每次构建都可以改一下，按日期的话可一天一改

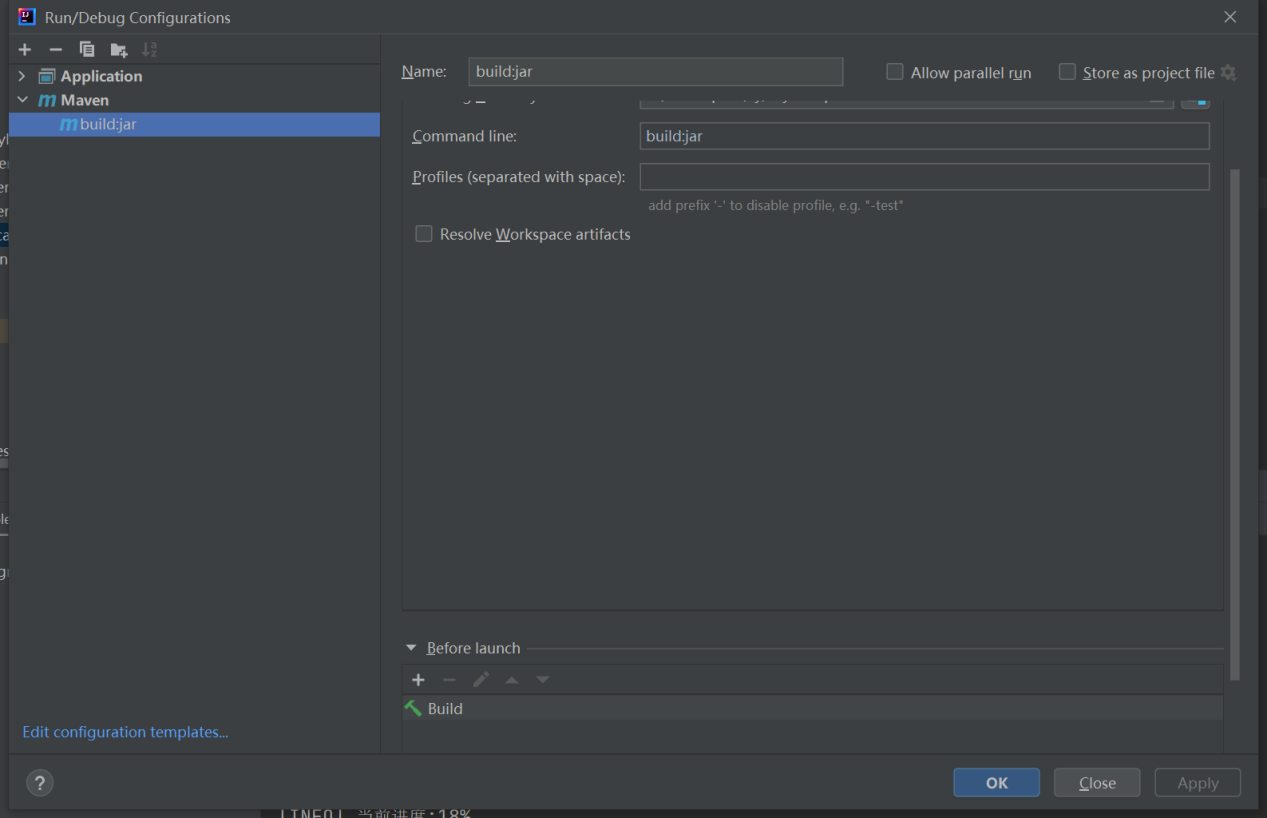
组合下来版本如下

<version>1.0.0.20211229</version>

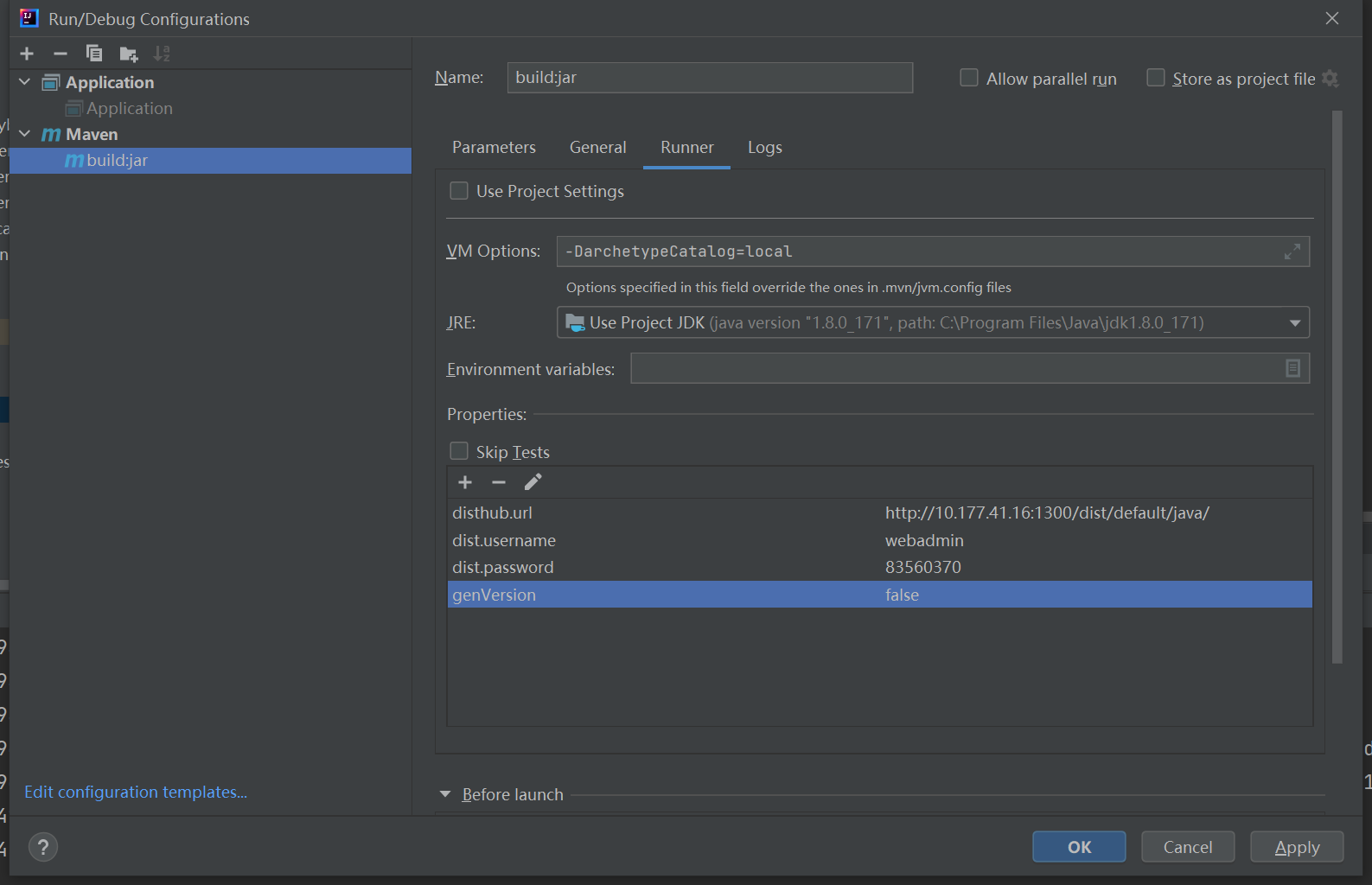
**3.运行maven命令**

build:jar -DdistHub.url=http://xxx.xxx.xxx:1300/dist/default/java/ -Ddist.username=xxx -Ddist.password=xxx -DgenVersion=false

iden运行如下



注意Before lauch + Build



**8.部署**

地址:/devops/runningexample

找到步骤3创建的实例，点击部署，选择步骤7上传的版本，点击下一步部署

