# nacos环境

## nacos安装包下载

<https://github.com/alibaba/nacos/releases>

下载zip或者tar.gz都可以

解压：

|  |
| --- |
| unzip nacos-server-$version.zip 或者 tar -xvf nacos-server-$version.tar.gz    cd nacos/bin |

## nacos启动

### Linux/Unix/Mac

启动命令(standalone代表着单机模式运行，非集群模式):

sh startup.sh -m standalone

如果您使用的是ubuntu系统，或者运行脚本报错提示[[符号找不到，可尝试如下运行：

bash startup.sh -m standalone

### Windows

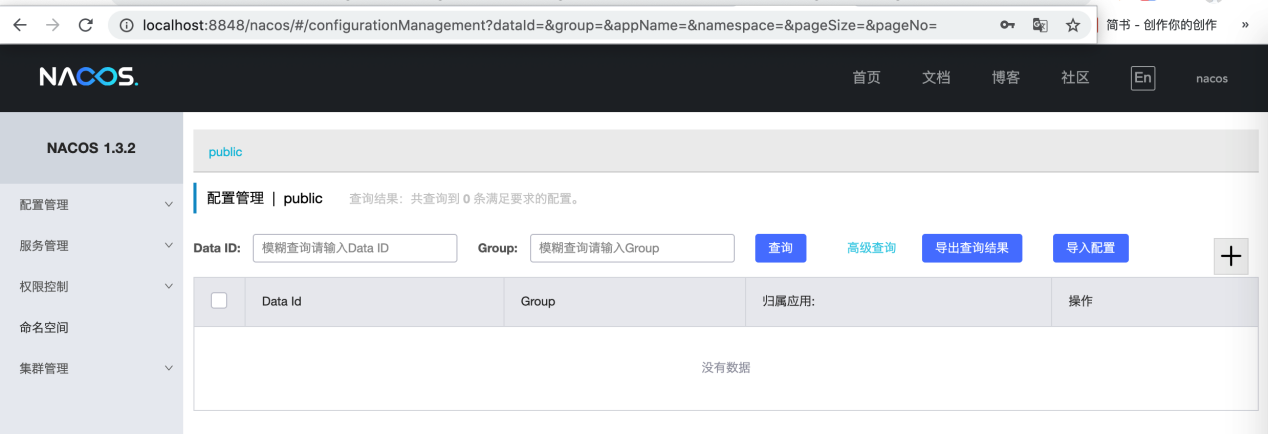
启动命令：

cmd startup.cmd

或者双击startup.cmd运行文件。

## nacos首页

<http://localhost:8848/nacos>



## nacos关闭

### Linux/Unix/Mac

sh shutdown.sh

### Windows

cmd shutdown.cmd

或者双击shutdown.cmd运行文件。

# springcloud集成nacos配置中心

## 创建命名空间

命名空间的作用：服务的隔离，不同命名空间的服务是不能互相访问的。spd的配置文件和服务均配置了命名空间

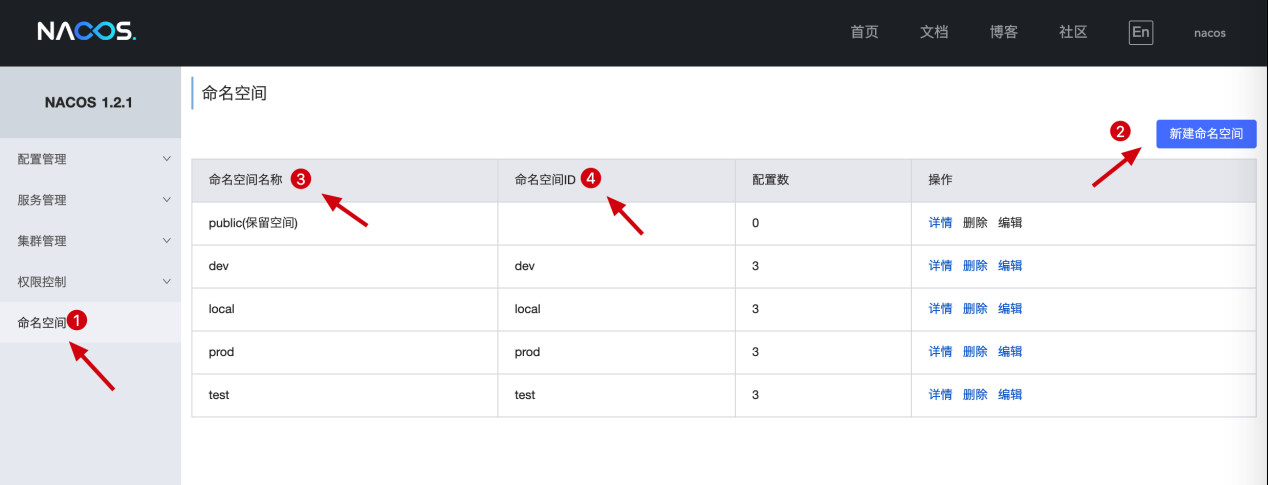
服务命名空间配置[在对应的配置文件中]：

|  |
| --- |
| spring:  cloud:  nacos:  discovery:  server-addr: 101.133.217.39:8848  namespace: dev #用于指定服务的命名空间 |

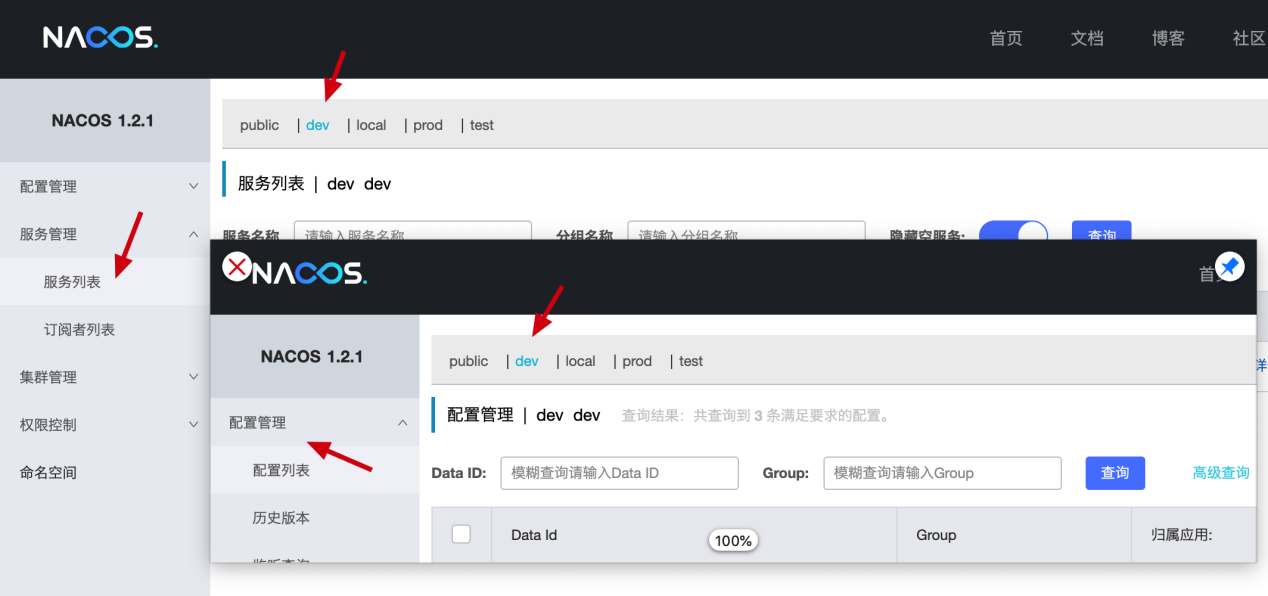
配置文件命名空间配置[在bootstrap.yml中]：

|  |
| --- |
| spring:  cloud:  nacos:  config:  #配置中心地址  server-addr: 101.133.217.39:8848  #配置中心命名空间，用于隔离，此处与spring.profiles.active对应  namespace: ${spring.profiles.active} #与运行环境对应 |

创建一个dev命名空间



在配置管理→配置列表和服务管理→服务列表均会出现对应的命名空间



## 添加依赖

|  |
| --- |
| <**dependency**>  <**groupId**>com.alibaba.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-alibaba-nacos-config</**artifactId**>  <**version**>${nacos-config.version}</**version**> </**dependency**> |

以spd为例：

springboot版本：2.3.3.RELEASE

springcloud版本：Hoxton.SR4

nacos版本：2.2.1.RELEASE

依赖如下：

|  |
| --- |
| <**parent**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-starter-parent</**artifactId**>  <**version**>2.3.3.RELEASE</**version**>  <**relativePath**/> *<!-- lookup parent from repository -->* </**parent**> |
| <**dependency**>  <**groupId**>org.springframework.cloud</**groupId**>  <**artifactId**>spring-cloud-dependencies</**artifactId**>  <**version**>${spring-cloud.version}</**version**>  <**type**>pom</**type**>  <**scope**>import</**scope**> </**dependency**>  <**spring-cloud.version**>Hoxton.SR4</**spring-cloud.version**> |
| <**dependency**>  <**groupId**>com.alibaba.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-alibaba-nacos-config</**artifactId**>  <**version**>${nacos-config.version}</**version**> </**dependency**>  <**nacos-config.version**>2.2.1.RELEASE</**nacos-config.version**> |

注意：版本 2.1.x.RELEASE 对应的是 Spring Boot 2.1.x 版本。版本 2.0.x.RELEASE 对应的是 Spring Boot 2.0.x 版本，版本 1.5.x.RELEASE 对应的是 Spring Boot 1.5.x 版本。

版本对应表：



## 项目配置

以spd-web为例：

### 引入依赖

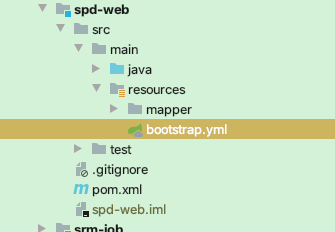
|  |
| --- |
| <**dependency**>  <**groupId**>com.alibaba.cloud</**groupId**>  <**artifactId**>spring-cloud-starter-alibaba-nacos-config</**artifactId**> </**dependency**> |

### 添加打包环境

方便运行时环境统一，防止引用不同的配置文件，防止发布到不同的命名空间

|  |
| --- |
| <**build**>  <**plugins**>  <**plugin**>  <**groupId**>org.springframework.boot</**groupId**>  <**artifactId**>spring-boot-maven-plugin</**artifactId**>  </**plugin**>  </**plugins**>  <**resources**>  <**resource**>  <**directory**>src/main/resources</**directory**>  <**filtering**>true</**filtering**>  </**resource**>  </**resources**> </**build**>  <**profiles**>  <**profile**>  <**id**>dev</**id**>  <**properties**>  <**profiles.active**>dev</**profiles.active**>  </**properties**>  </**profile**>  <**profile**>  <**id**>test</**id**>  <**properties**>  <**profiles.active**>test</**profiles.active**>  </**properties**>  </**profile**>  <**profile**>  <**id**>release</**id**>  <**properties**>  <**profiles.active**>release</**profiles.active**>  </**properties**>  </**profile**>  <**profile**>  <**id**>local</**id**>  <**properties**>  <**profiles.active**>local</**profiles.active**>  </**properties**>  <**activation**>  <**activeByDefault**>true</**activeByDefault**>  </**activation**>  </**profile**>  <**profile**>  <**id**>prod</**id**>  <**properties**>  <**profiles.active**>prod</**profiles.active**>  </**properties**>  </**profile**> </**profiles**> |

### bootstrap.yml文件修改



此处只留有bootstrap.yml一个配置文件，配置项目的基础配置

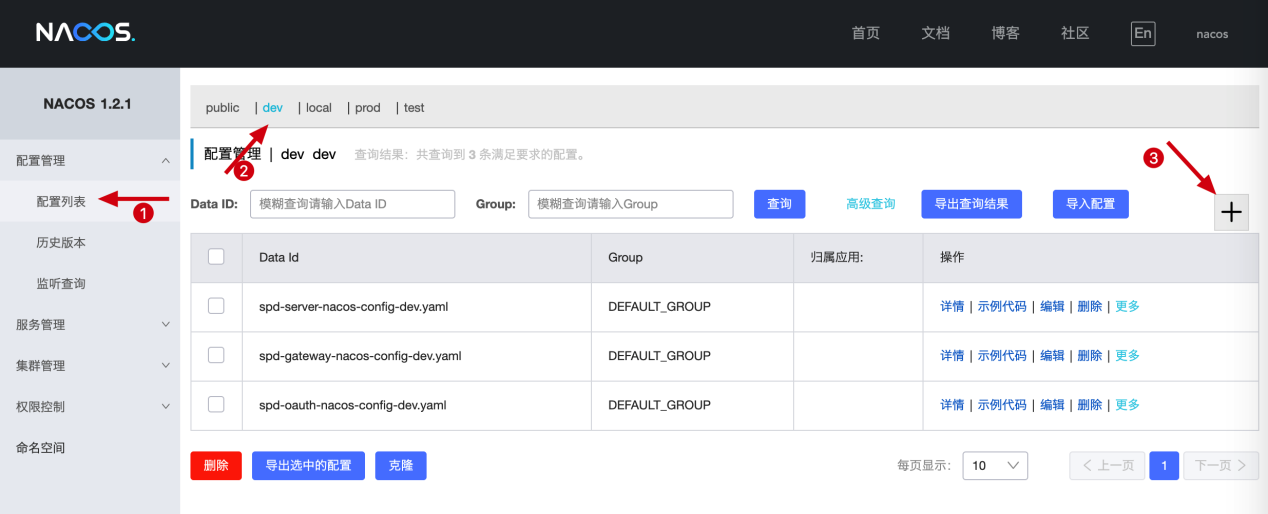
|  |
| --- |
| *#此配置文件用于写一般情况下固定不变的配置* **server**:  **port**: 9003  **undertow**:  **buffer-size**: 1024  **direct-buffers**: **true  shutdown**: *graceful #优雅关机配置* **spring**:  **application**:  **name**: spd-web  **profiles**:  *# 这里配置对应nacos上命名空间里的配置文件  #local、test、dev、prod* **active**: @profiles.active@  **lifecycle**:  **timeout-per-shutdown-phase**: 30s *#设置缓冲期，最大等待时间* **main**:  **allow-bean-definition-overriding**: **true  servlet**:  **multipart**:  **max-request-size**: 100MB  **enabled**: **true  max-file-size**: 1024MB   **cloud**:  **nacos**:  **config**:  *#配置中心地址* **server-addr**: 101.133.217.39:8848  *#配置中心命名空间，用于隔离，此处与spring.profiles.active对应* **namespace**: ${**spring.profiles.active**}  *#默认值，创建也使用默认值* **group**: DEFAULT\_GROUP  *#配置文件名前缀* **prefix**: spd-server-nacos-config  *#配置文件格式，暂时支持yaml和properties类型* **file-extension**: yaml  *#配置中心date-id固定写法* **data-id**: ${**spring.cloud.nacos.config.prefix**}-${**spring.profiles.active**}.${**spring.cloud.nacos.config.file-extension**} |

### 在nacos上创建配置文件

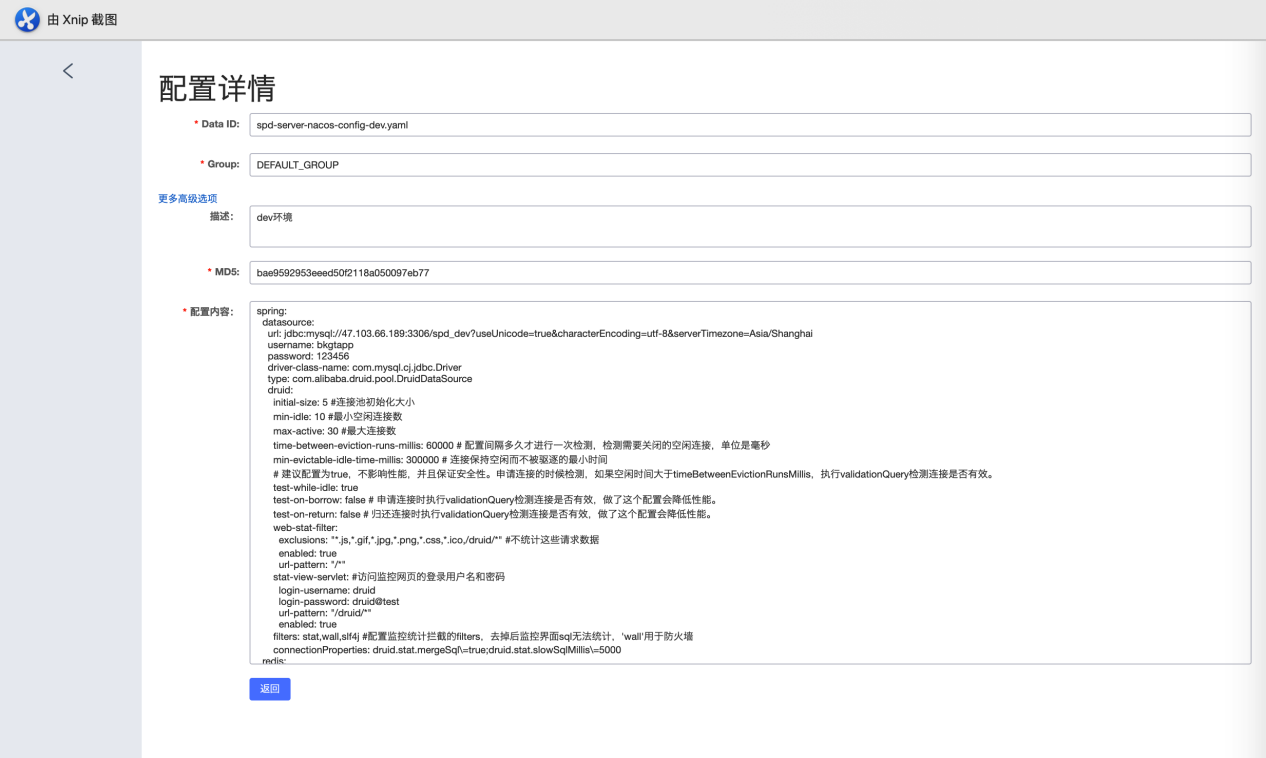
首先要保证创建的配置文件在本地能正常运行，支持yaml和properties格式。

配置文件名格式[对应data Id的值]：

在dev命名空间创建dev环境配置文件







以上为配置文件详情，MySQL、Redis等配置都在此文件中。

### 添加注解

通过 Spring Cloud 原生注解 @RefreshScope 实现配置自动更新：

通过@Value("${useLocalCache}")来获取配置文件中的值。

如下：可以获取到配置文件中useLocalCache的值。

|  |
| --- |
| @RestController  @RequestMapping("/config")  @RefreshScope  public class ConfigController {  @Value("${useLocalCache}")  private boolean useLocalCache;  @RequestMapping("/get")  public boolean get() {  return useLocalCache;  }  } |

# 项目介绍

## spd配置中心

### 配置文件介绍

spd共分为三类配置文件，每类配置文件分local、test、dev、prod环境，环境后续可以在新增。[srm同理]

|  |  |
| --- | --- |
| 配置文件名 | 服务 |
| spd-server-nacos-config-dev.yaml | 所有业务服务[web、stock、job...] |
| spd-gateway-nacos-config-dev.yaml | spd-gateway |
| spd-oauth-nacos-config-dev.yaml | spd-oauth |

运行环境对应nacos的命名空间

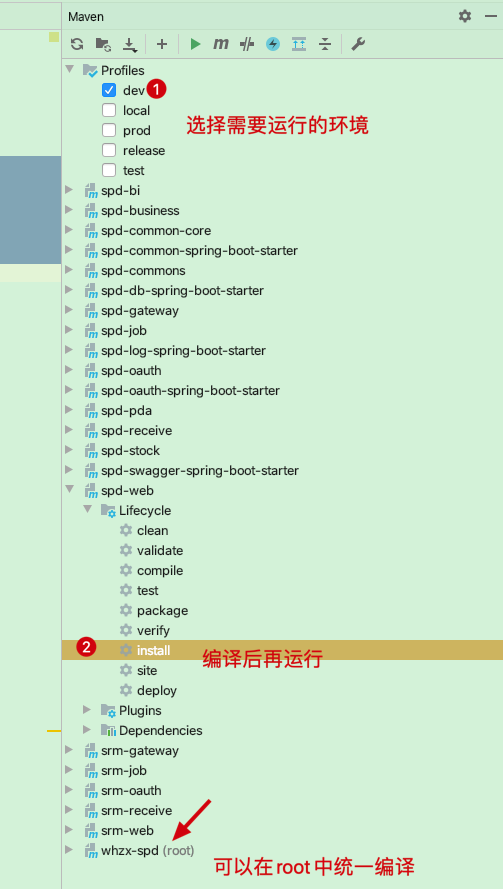
如spd-server-nacos-config-dev.yaml 对应dev环境 对应dev命名空间

spd-server-nacos-config-test.yaml 对应test环境 对应test命名空间

### 项目运行

由于运行环境为动态配置，所以需要先编译后运行。编译过程中会去nacos上找对应的配置文件。

|  |
| --- |
| **spring**:  **profiles**:  *# 这里配置对应nacos上命名空间里的配置文件  #local、test、dev、prod* **active**: @profiles.active@ |

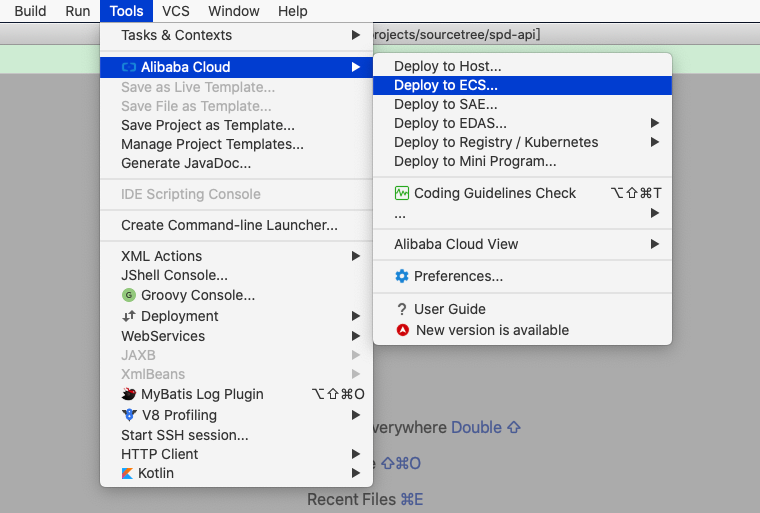


通过如果通过脚本使用maven进行编译，需加入运行环境参数，默认为local。如：

mvn clean install -P dev

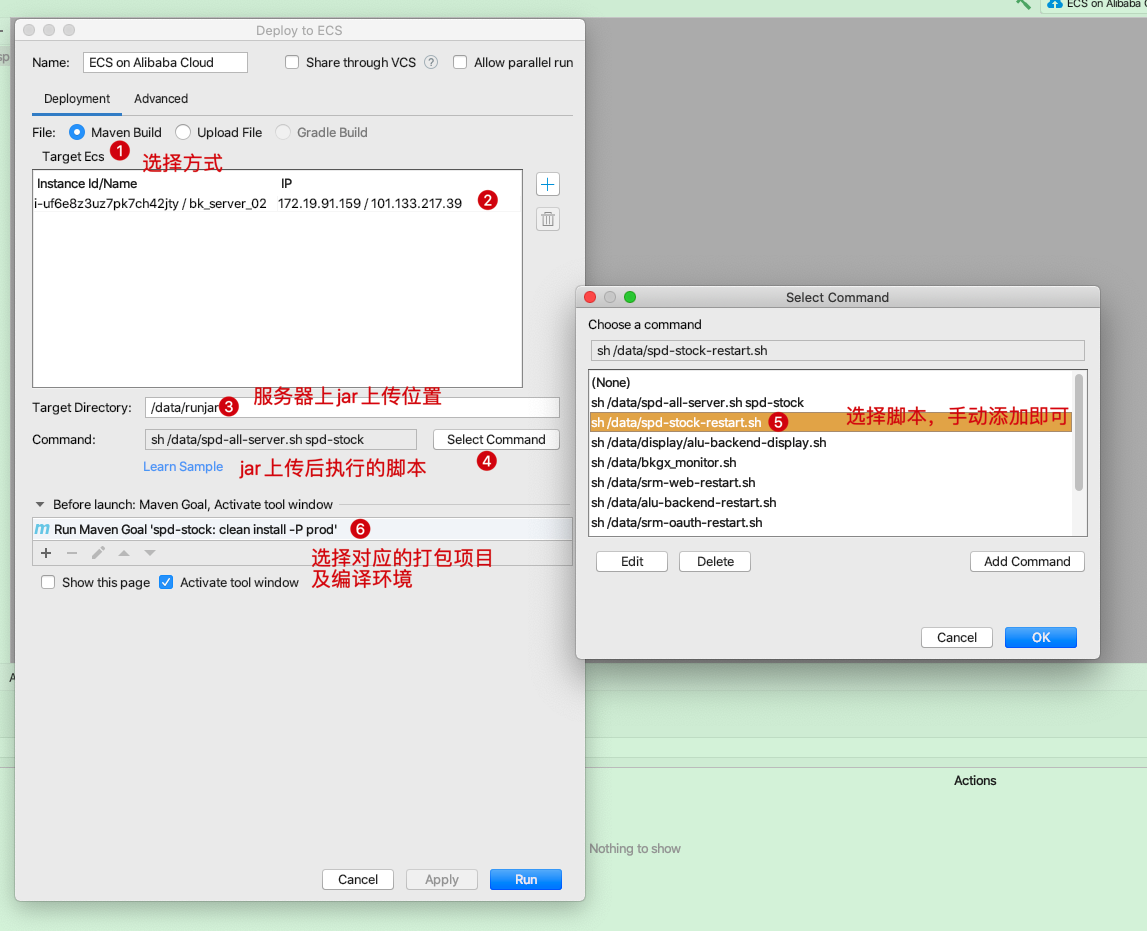
## ali Cloud Tools两种发布方式

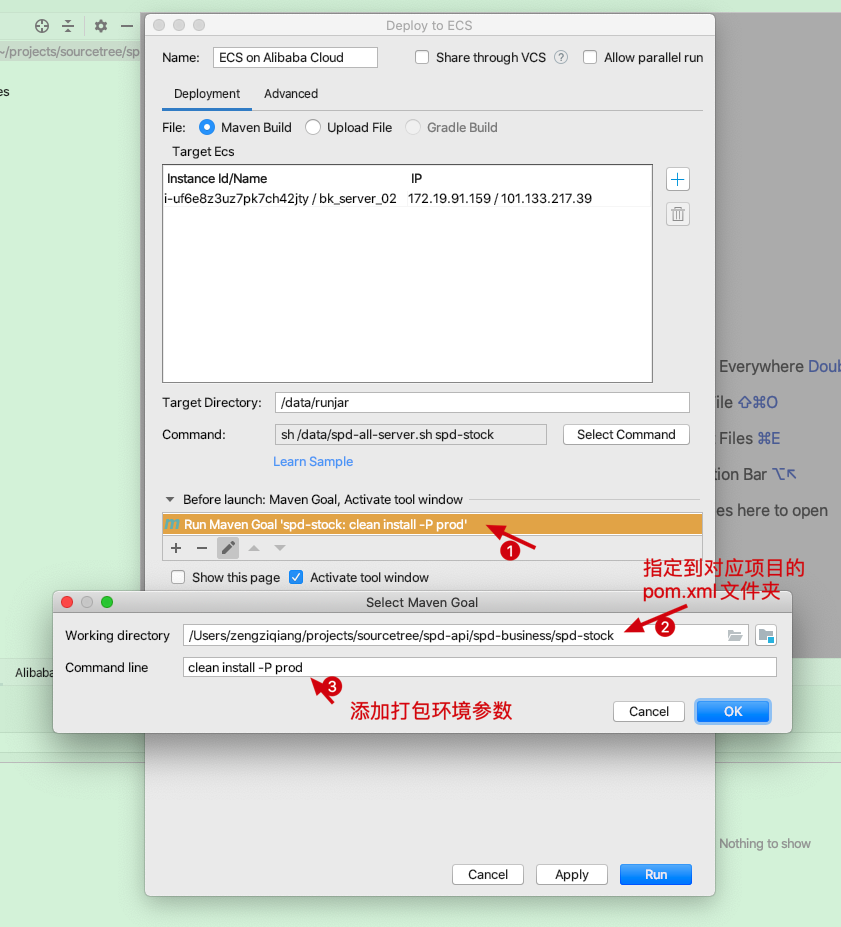
打开tools工具



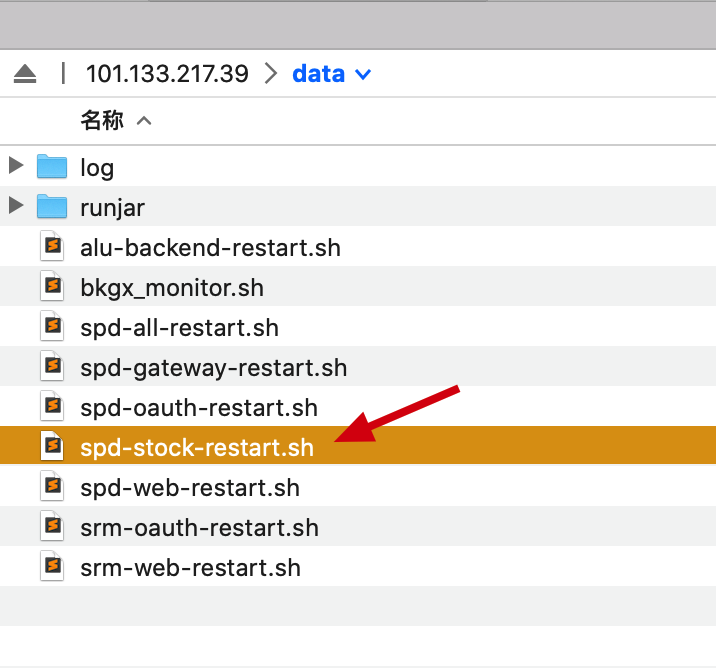
### Maven Build方式

以spd-stock为例





服务器上脚本



|  |
| --- |
| source /etc/profile  netstat -anp|grep 9007|awk '{printf $7}'|cut -d/ -f1 |xargs kill -9 || true  nohup java -jar /data/runjar/spd-stock-0.0.1-SNAPSHOT.jar > /data/log/spd-stock.log 2>&1 & |

### Upload File方式

先选择对应的环境，然后进行install编译，将编译后的jar项目上传到服务器，然后执行对应的脚本。

