

分布式任务调度 SchedulerX 快速入门

ALIBABA CLOUD

文档版本: 20210721



法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。 如果您阅读或使用本文档,您的阅读或使用行为将被视为对本声明全部内容的认可。

- 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档,且仅能用 于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息,您应当严格 遵守保密义务;未经阿里云事先书面同意,您不得向任何第三方披露本手册内容或 提供给任何第三方使用。
- 未经阿里云事先书面许可,任何单位、公司或个人不得擅自摘抄、翻译、复制本文 档内容的部分或全部,不得以任何方式或途径进行传播和宣传。
- 由于产品版本升级、调整或其他原因,本文档内容有可能变更。阿里云保留在没有 任何通知或者提示下对本文档的内容进行修改的权利,并在阿里云授权通道中不时 发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠 道下载、获取最新版的用户文档。
- 4. 本文档仅作为用户使用阿里云产品及服务的参考性指引,阿里云以产品及服务的"现状"、"有缺陷"和"当前功能"的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引,但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的,阿里云不承担任何法律责任。在任何情况下,阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害,包括用户使用或信赖本文档而遭受的利润损失,承担责任(即使阿里云已被告知该等损失的可能性)。
- 5. 阿里云网站上所有内容,包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计,均由阿里云和/或其关联公司依法拥有其知识产权,包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意,任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外,未经阿里云事先书面同意,任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称(包括但不限于单独为或以组合形式包含"阿里云"、"Aliyun"、"万网"等阿里云和/或其关联公司品牌,上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司)。
- 6. 如若发现本文档存在任何错误,请与阿里云取得直接联系。

通用约定

格式	说明	样例	
⚠ 危险	该类警示信息将导致系统重大变更甚至故 障,或者导致人身伤害等结果。		
▲ 警告	该类警示信息可能会导致系统重大变更甚 至故障,或者导致人身伤害等结果。	警告 重启操作将导致业务中断,恢复业务 时间约十分钟。	
〔) 注意	用于警示信息、补充说明等,是用户必须 了解的内容。	大意 权重设置为0,该服务器不会再接受新 请求。	
? 说明	用于补充说明、最佳实践、窍门等 <i>,</i> 不是 用户必须了解的内容。	⑦ 说明 您也可以通过按Ctrl+A选中全部文件。	
>	多级菜单递进。	单击设置> 网络> 设置网络类型。	
粗体	表示按键、菜单、页面名称等UI元素。	在 结果确认 页面,单击 确定 。	
Courier字体	命令或代码。	执行 cd /d C:/window 命令,进入 Windows系统文件夹。	
斜体	表示参数、变量。	bae log listinstanceid	
[] 或者 [alb]	表示可选项,至多选择一个。	ipconfig [-all -t]	
{} 或者 {a b}	表示必选项,至多选择一个。	switch {act ive st and}	

目录

1.准备工作	05
1.1. 开通SchedulerX(免费)	05
1.2. 创建资源	05
2.客户端快速接入SchedulerX	11
2.1. Java应用接入SchedulerX	11
2.2. Spring应用接入SchedulerX	13
2.3. Spring Boot应用接入SchedulerX	15
2.4. Agent接入(调度任务)	19
2.5. 在本地接入公网测试环境	20
2.6. 容器服务Kubernetes版接入SchedulerX	25
2.7. Endpoint列表	34

1.准备工作

1.1. 开通SchedulerX(免费)

在开始使用SchedulerX前,需要先开通。SchedulerX目前在公测期,免费。

背景信息

分布式任务调度目前处于公测期,免费使用。

操作步骤

- 1. 登录分布式任务调度平台。
- 2. 首次使用并登录分布式任务调度平台,在弹出的对话框中单击开通。
- 3. 在分布式任务管理(按量后付费)页面单击立即购买。

付费类型为后付费类型,任务计算量为按任务计算量付费,不可修改。

 在确认订单页面服务协议区域单击我已阅读并同意分布式任务管理(按量后付费)服务协议,单 击去支付。

1.2. 创建资源

在使用SchedulerX前,您需要先创建相关资源,包括命名空间、调度任务分组、调度任务和调度工作流。

背景信息

命名空间(测试)	命名空间(生产)
应用1	应用1 ⊥作流1 Job1 Job2 Job3
应用2 Job4 Job5 Job6	应用2 Job4 Job5 Job6

资源	说明	使用场景
命名空间	在具体地域(Region)中,命名空间用于实 现资源和服务的隔离。	当您对资源有较高的安全要求时,需要创建 命名空间。

资源	说明	使用场景
应用	在具体的命名空间下,和应用绑定,关联一 组机器。	通过groupld绑定应用。
任务	在具体的应用下,任务和一段代码逻辑绑 定,用来实现任务调度。	任务是SchedulerX调度的最小单位,用来实 现周期性的任务调度。
工作流	在具体应用下,工作流用来实现任务的依赖 编排。	工作流是SchedulerX对任务进行依赖编排的 封装,支持上下游数据传递。

创建命名空间(可选)

- 1. 登录分布式任务调度平台。
- 2. 在左侧导航栏单击命名空间。
- 3. 在顶部菜单栏选择地域,然后单击创建命名空间。
- 在创建命名空间面板,输入命名空间的名称和(可选)描述,然后单击确定。
 命名空间创建成功,会提示 创建成功。返回命名空间页面,列表中包含刚创建的命名空间信息。

创建应用

- 1. 登录分布式任务调度平台。
- 2. 在顶部菜单栏选择地域。
- 3. 在左侧导航栏单击应用管理。
- 4. 在应用管理页面选择目标命名空间,然后单击创建应用。
- 5. 在基本配置配置向导页, 输入应用名和应用ID, 设置高级配置参数(可选), 然后单击下一步。

← 创建应用 ×				×
1 基本	和置		2 报警配置	
应用名 *	Doctes	t		
应用ID ② *	doctes	t		
描述	请输入	描述。		
			0/6	4
实例繁忙配置:				
load5 🕖	-	0 +		
内存使用率 💡	-	90 + %		
磁盘使用率 ?	—	95 + %		
是否触发繁忙机器				
◇ 高级配置				
任务最大数量 🕢	-	1000 +		
自动扩容				
流控				
下一步取消				
参数		解释	默认值	
应用名		自定义设置应用名称。	无	
应用ID		应用ID 为应用接入的GroupID,保 证同一个命名空间下唯一,否则将 创建失败,可以和 应用名 一致。	无	

参数	解释	默认值
load5	不能超过客户端机器CPU可用核数	0
内存使用率	表示近5分钟进程内存平均使用率 不能大于该阈值,否则判断客户端 机器繁忙。	90%
磁盘使用率	表示磁盘使用率不能大于该值,否 则判断客户端机器不健康,状态繁 忙。	95%
是否触发繁忙机器	机器繁忙时是否继续触发客户端执 行。	打开
任务最大数量	一个分组最多支持的Job数量。	1000
自动扩容	选择是否自动扩容。开启时,需要 设置 全局任务数 。	关闭
流控	选择是否流控。开启时,需要设 置 任务实例并发数 。	关闭

在报警配置配置向导页,选择报警渠道并设置报警联系人,然后单击完成。
 需要设置多个报警联系人时,单击添加报警联系人,然后输入联系人信息。

← 创建应用 ×			
 ✓ 基本 	2111日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日1日	2 报警配置	
报警渠道 * 报警联系人:	✔ 短信 ✔ 钉钉 ✔ 邮件	ŧ	
昵称 *	101		
邮件	请输入邮件		
钉钉机器人 🕗	请输入钉钉		
手机号	15 25		
昵称 *	文档测试机器人		_
邮件	请输入邮件		
钉钉机器人 🕗	https://lf45a		
手机号	请输入手机号		
昵称 *			_
邮件	.com		
钉钉机器人 🛛	请输入钉钉		
上一步 完成			
参数		描述	
报警渠道		目前支持 短信、钉钉和邮件 ,支持	设置多种渠道。
昵称	自定义设置报警联系人的昵称。		
邮件		输入报警联系人的真实邮件地址。	

参数	描述	
	输入钉钉机器人对应的Webhook地址。	
钉钉机器人	<text><text></text></text>	
手机号	输入报警联系人的真实手机号码。	

应用创建完成后,自动返回应用管理页面,在分组列表中查看分组是否已存在。

◯ 注意 应用管理页面中的应用ID (groupId)和应用key (appKey)将作为客户端接入的配置 参数。

应用管理 所属命名空间:	华东1	\sim			联	系我们: 🚫 23133165 [帮助文档]
创建应用 应用ID/应用名称	✔ 请输入	Q				© C
应用名称/应用描述	应用ID/应用key		最后更新人	已有/最大任务数	实例总数 ◎ 4	操作
$\label{eq:constraint} \begin{bmatrix} 2 & y & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y & y \\ 0 & y & y & y & y & y \\ 0 & y & y & y & y & y \\ 0 & y & y & y & y & y \\ 0 & y & y & y & y & y \\ 0 & y & y & y & y \\ 0 & y & y & y & y \\ 0 & y & y & y & y \\ 0 & y & y & y & y \\ 0 & y & y & y & y \\ 0 & y \\ 0 & y & y \\ 0 & y & y \\ $	jingxiangbushu2.defaultGroup dF3EVRyLF	roupid ppKey	c in the matrix of the latter in the matrix 10205050017	2 / 1000	0	编辑 授权 操作记录 :
					毎页显示 10 ~ す	村条 < 上一页 1 下一页 >

2.客户端快速接入SchedulerX

2.1. Java应用接入SchedulerX

您可以为您的Java应用快速接入SchedulerX,实现分布式任务调度能力。

前提条件

• 创建命名空间(可选)

. .

• 创建应用

Java应用客户端接入SchedulerX

1. 在应用程序的 pom.xml 文件中添加SchedulerxWorker依赖。

请参见客户端版本说明, schedulerx2.version 使用最新客户端版本。

<aependency></aependency>
<groupid>com.aliyun.schedulerx</groupid>
<artifactid>schedulerx2-worker</artifactid>
<version>\${schedulerx2.version}</version>
如果用的是logback,需要把log4j和log4j2排除掉
<exclusions></exclusions>
<exclusion></exclusion>
<groupid>org.apache.logging.log4j</groupid>
<artifactid>log4j-api</artifactid>
<exclusion></exclusion>
<groupid>org.apache.logging.log4j</groupid>
<artifactid>log4j-core</artifactid>
<exclusion></exclusion>
<groupid>log4j</groupid>
<artifactid>log4j</artifactid>

- 2. 在main函数中初始化SchedulerxWorker。
 - 初始化SchedulerxWorker时,会用到您部署应用的地域(Region)和对应的Endpoint。详情请参见Endpoint列表。
 - Namespace为命名空间ID, 可以在控制台命名空间页面获取。

分布式任务调度平台	命名空间		
执行列表	_		
任务管理	创建命名空间		(\$) C
流程管理	命名空间名称/描述	命名空问ID 🛊	操作
应用管理	华东1 暂无描述	2	編録 翻除
命名空间 1	暂无描述	COMPANY AND AND AND AND AND AND AND	明朝韓一副限会
操作记录	暂无描述	contract with a second second second	编辑 计数据令

○ GroupId为应用ID, 可以在控制台应用管理页面获取。

分布式任务调度平台	应用管理 新属命名空间: 樂	东1	~					
执行列表								
任务管理	创建应用 应用ID/应用名称 ~	/ 请输入 Q					۲	С
流程管理	应用名称/应用描述 应用ID/应	用key	最后更新人	已有/最大任务数	实例总数 🕢 🕇	操作		
应用管理 1	暫无描述		de ciliga con co	0 / 1000	0	编辑 授权 排	制作记录 :	
命名空间操作记录	智无描述	2	de altra de contractor	0 / 1000	0	編編 授权 排	制作记录 :	

AliyunAccessKey和AliyunSecretKey为阿里云账号的AccessKeyID和AccessKeySecret,可以在用户信息管理控制台的安全信息管理页面获取。

public void initSchedulerxWorker() throws Exception {
 SchedulerxWorker schedulerxWorker = new SchedulerxWorker();
 schedulerxWorker.setEndpoint("xxxx");
 schedulerxWorker.setNamespace("xxxx");
 schedulerxWorker.setGroupId("xxxx");
 //1.2.1及以上版本需要设置应用key
 schedulerxWorker.setAppKey("xxxx");
 //1.2.1以下版本需要设置AK/SK
 //schedulerxWorker.setAliyunAccessKey("xxxx");
 //schedulerxWorker.setAliyunSecretKey("xxxx");
 schedulerxWorker.init();

⑦ 说明

}

- 一个应用如果包含多个业务,或者想把定时任务进行归类,可以建立多个分组,例如应用 a nimals 新建了两个分组 animals.dogs 和 animals.cats 。此时不用申请两批实例分别接入 这两个分组,在应用客户端中将这两个分组配置到 groupId= 后面即可,例如 groupId=ani mals.dogs,animals.cats 。
- 在初始化SchedulerxWorker客户端时,如果还有其它配置需求,可以参
 考SchedulerxWorker 配置参数说明添加配置。
- 3. 在应用中创建类 JobProcessor , 实现任务调度。

本文仅介绍如何实现一个最简单的定时打印 "Hello SchedulerX2.0" 的 JobProcessor 类。

```
package com.aliyun.schedulerx.test.job;
import com.alibaba.schedulerx.worker.domain.JobContext;
import com.alibaba.schedulerx.worker.processor.JavaProcessor;
import com.alibaba.schedulerx.worker.processor.ProcessResult;
@Component
public class MyHelloJob extends JavaProcessor {
  @Override
  public ProcessResult process(JobContext context) throws Exception {
    System.out.println("hello schedulerx2.0");
    return new ProcessResult(true);
  }
}
```

结果验证

- 1. 客户端接入完成,将该应用发布到阿里云。
- 2. 登录分布式任务调度平台。

- 3. 在顶部菜单栏选择地域。
- 4. 在左侧导航栏单击应用管理。
- 5. 在应用管理页面查看实例总数。
 - 如果**实例总数**为0,则说明应用接入失败。请检查、修改本地应用。
 - 如果实例总数不为0,显示接入的实例个数,则说明应用接入成功。在操作列单击查看实例,即可 在连接实例对话框中查看实例列表。

应用管理 📠	名空间:	\sim				
创建应用 应用ID/应	用名称 🗸 请输入	Q				\$ C
应用名称/应用描述	应用ID/应用key	最后更新人	已有/都	最大任务数 实例总数	②1 操作	
schedulerx-fatjar 暫无描述	ALCO DE LA COLUMNO -	10.000	6,	/ 1000 0	编辑 授权 排	象作记录 :
schedulerx-fatjar 暫无描述	checker right is discontration disease	10.000	18	/ 1000 2	查看实例编辑	计授权 :

后续步骤

应用接入SchedulerX完成后,即可在分布式任务调度平台创建调度任务。详情请参见创建调度任务。

2.2. Spring应用接入SchedulerX

您可以为您的Spring应用快速接入SchedulerX,实现分布式任务调度能力。

前提条件

- 创建命名空间(可选)
- 创建应用

Spring应用客户端接入SchedulerX

1. 在应用程序的 pom.xml 文件中添加SchedulerxWorker依赖。

```
请参见客户端版本说明, schedulerx2.version 使用最新客户端版本。
```

```
<dependencv>
<groupId>com.aliyun.schedulerx</groupId>
<artifactId>schedulerx2-worker</artifactId>
<version>${schedulerx2.version}</version>
<!--如果用的是logback,需要把log4j和log4j2排除掉-->
<exclusions>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
 </exclusion>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
 </exclusion>
 <exclusion>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
 </exclusion>
</exclusions>
</dependency>
```

- 2. 在xml配置文件中初始化SchedulerxWorker(注入Bean)。
 - 初始化SchedulerxWorker时,会用到您部署应用的地域(Region)和对应的Endpoint。详情请参见Endpoint列表。
 - namespace为命名空间ID,可以在控制台命名空间页面获取。

分布式任务调度平台	命名空间		
执行列表			
任务管理	创建命名空间		\$ C
流程管理	命名空间名称/描述	命名空间ID 🖡	操作
应用管理	华东1 暫无描述	2	4961378 计 最短序会
命名空间 🚺	暂无描述	10000 001 00 000 000 0000	相同 经 图》
操作记录	暂无描述	contraction and and the same case for any	網線 副除

○ groupId为应用ID, appKey为应用key, 可以在控制台应用管理页面获取。

应用管理 所属命名空间:	华东1	~			联;	系我们: 🕥 23133165	[帮助文档]
创建应用 应用ID/应用名称	✓ 请输入	Q					\$ C
应用名称/应用描述	应用ID/应用key		最后更新人	已有/最大任务数	亥例总数 ❷↓	操作	
(https://www.col/ acces	jingxiangbushu2.defaultGroup dF3EVRyLF1_df1CH1188T14_s=	groupId appKey	, the presence of the limit, where κ are 17 224 58 58 523 (22	2 / 1000	0	编辑 授权 :	集作记录 :
				每页显示	10 ~ #	+1条 く 上一页 1	下一页 >

aliyunAccessKey和aliyunSecretKey为阿里云账号的AccessKeyID和AccessKeySecret,可以在用户信息管理控制台的安全信息管理页面获取。

<bean id="schedulerxWorker" class="com.alibaba.schedulerx.worker.SchedulerxWorker">
 property name="endpoint">

```
<value>${endpoint}</value>
</property>
<property name="namespace">
<value>${namespace}</value>
</property>
<property name="groupId">
<value>${groupId}</value>
</property>
<!--1.2.1及以上版本设置appKey -->
<property name="appKey">
<value>${appKey}</value>
</property>
<!--1.2.1以下版本需要设置AK/SK -->
<!--
<property name="aliyunAccessKey">
<value>${aliyunAccessKey}</value>
</property>
<property name="aliyunSecretKey"></property name="aliyunSecretKey">
<value>${aliyunSecretKey}</value>
</property>
-->
```

```
</bean>
```

? 说明

- 一个应用如果包含多个业务,或者想把定时任务进行归类,可以建立多个分组,例如应用 a nimals 建了两个分组 animals.dogs 和 animals.cats 。此时不用申请两批实例分别接入这两个分组,在应用客户端中将这两个分组配置到 groupId= 后面即可,例如 groupId=animal s.dogs,animals.cats 。
- 在初始化SchedulerxWorker客户端时,如果还有其它配置需求,可以参考SchedulerxWorker 配置参数说明添加配置。
- 3. 在应用中创建类 JobProcessor , 实现任务调度。

```
本文仅介绍如何实现一个最简单的定时打印 "Hello SchedulerX2.0" 的 JobProcessor 类。
```

```
package com.aliyun.schedulerx.test.job;
import com.alibaba.schedulerx.worker.domain.JobContext;
import com.alibaba.schedulerx.worker.processor.JavaProcessor;
import com.alibaba.schedulerx.worker.processor.ProcessResult;
@Component
public class MyHelloJob extends JavaProcessor {
  @Override
  public ProcessResult process(JobContext context) throws Exception {
    System.out.println("hello schedulerx2.0");
    return new ProcessResult(true);
  }
}
```

结果验证

- 1. 客户端接入完成,将该应用发布到阿里云。
- 2. 登录分布式任务调度平台。
- 3. 在顶部菜单栏选择地域。
- 4. 在左侧导航栏单击应用管理。
- 5. 在应用管理页面查看实例总数。
 - 如果**实例总数**为0,则说明应用接入失败。请检查、修改本地应用。
 - 如果实例总数不为0,显示接入的实例个数,则说明应用接入成功。在操作列单击查看实例,即可 在连接实例对话框中查看实例列表。

应用管理 新属命名	空间:	\sim				
创建应用 应用ID/应用	名称 > 请输入	Q				\$ C
应用名称/应用描述	应用ID/应用key	最后更新人	已有/最大任务数	实例总数 🚱 🖡	操作	
schedulerx-fatjar 暂无描述		standings and an interaction	6 / 1000	0	编辑 授权 操作记录	:
schedulerx-fatjar 暂无描述	checker Hands Amount Schemerer	standing the second	18 / 1000	2	查看实例 编辑 授权	:

后续步骤

应用接入SchedulerX完成后,即可在分布式任务调度平台创建调度任务。详情请参见创建调度任务。

2.3. Spring Boot应用接入SchedulerX

您可以为您的Spring Boot应用快速接入SchedulerX,实现分布式任务调度能力。

前提条件

- 创建命名空间(可选)
- 创建应用

Spring Boot应用客户端接入SchedulerX

1. 在应用程序的 pom.xml 文件中添加SchedulerxWorker依赖。

```
请参见客户端版本说明, schedulerx2.version 使用最新客户端版本。
```

```
<dependency>
<groupId>com.aliyun.schedulerx</groupId>
<artifactId>schedulerx2-spring-boot-starter</artifactId>
<version>${schedulerx2.version}</version>
<!--如果用的是logback,需要把log4j和log4j2排除掉-->
<exclusions>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
 </exclusion>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
 </exclusion>
 <exclusion>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
 </exclusion>
</exclusions>
</dependency>
```

- 2. 在 application.properties 文件中设置相关参数,初始化SchedulerxWorker。
 - 初始化SchedulerxWorker时,会用到您部署应用的地域(Region)和对应的Endpoint。详情请参 见Endpoint列表。
 - 分布式任务编集中台
 命名空间

 时7列末
 ●

 任务 管理
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 ●
 - namespace为命名空间ID, 可以在控制台**命名空间**页面获取。

○ groupId为应用ID, appKey为应用key, 可以在控制台应用管理页面获取。

应用管理 新属命名空间:	华东1	~			联芳	(我们): 🚫 23133165 【帮助文档】
创建应用应用ID/应用名称	✓ 请输入 Q					©
应用名称/应用描述	应用ID/应用key	最后更新人		已有/最大任务数	实例总数 ❷▮	操作
(Tayle programs of 1 days	jingxiangbushu2.defaultGroup groupId dF3EVRyLF dF1F1211a1a1 appKey	$c \log (c \cos \alpha d \sin \beta \sin \beta d s)$	ler maler 1722/MSSS5220177	2 / 1000	0	编辑 授权 操作记录 :
					每页显示 10 ~ 共	1条 く 上一页 1 下一页 >

○ aliyunAccessKey和aliyunSecretKey为阿里云账号的AccessKeyID和AccessKeySecret,可以在用户信

息管理控制台的安全信息管理页面获取。

spring.schedulerx2.endpoint=\${endpoint} spring.schedulerx2.namespace=\${namespace} spring.schedulerx2.groupId=\${groupId} # 1.2.1及以上版本设置appKey spring.schedulerx2.appKey=\${appKey} # 1.2.1以下版本设置AK/SK #spring.schedulerx2.aliyunAccessKey=\${aliyunAccessKey} #spring.schedulerx2.aliyunSecretKey=\${aliyunSecretKey}

⑦ 说明 一个应用如果包含多个业务,或者想把定时任务进行归类,可以建立多个分组,例如应用 animals 新建了两个分组 animals.dogs 和 animals.cats 。此时不用申请两批实例分别接入这两个分组,在应用客户端中将这两个分组配置到 groupId= 后面即可,例如 groupId=animals.dogs,ani mals.cats。

如果还有其它需求,请参考下表中的参数进行配置。

key	描述	设置值	起始版本
spring.schedulerx2.ena bled	是否启用SchedulerX 2.0 的starter,默认开启不 需要设置。	true/false, 默认true。	0.1.7
spring.schedulerx2.end point	设置Region所在的 Endpoint , 详情请参 见 <mark>Endpoint列表</mark> 。	无	0.1.7
spring.schedulerx2.na mespace	设置命名空间的UID,可 以在控制台 命名空间 页 面获取。	无	0.1.7
spring.schedulerx2.gro upId	应用ID,可以在控制台 应 用管理 页面获取。	无	0.1.7
spring.schedulerx2.app Key	应用key,可以在控制 台 应用管理 页面获取。	无	1.2.1
spring.schedulerx2.hos t	如果有多个IP(例如VPN 或者多网卡),可以设置 真实的IP。	无	0.1.7
spring.schedulerx2.por t	用户自定义客户端监听端 口。如果不设置 <i>,</i> 则随机 选择一个可用端口。	无	0.1.7
spring.schedulerx2.blo ckAppStart	SchedulerX初始化失败 是否block应用启动,默 认为True。	true/false, 默认true。	1.1.0
spring.schedulerx2.sha reContainerPool	客户端所有任务执行是否 共享线程池 <i>,</i> 默认为 False。	无	1.2.1.2

快速入门·客户端快速接入SchedulerX

key	描述	设置值	起始版本
spring.schedulerx2.sha rePoolSize	如果开启共享线程池,可 以自定义线程池大小,默 认为64。	无	1.2.1.2
spring.schedulerx2.lab el	不同客户端可以设置标 签,任务管理可以指定标 签执行。应用于灰度、压 测等场景。	无	1.2.2.2
spring.schedulerx2.ena bleCgroupMetrics	是否使用cgroup统计客 户端实例的指标。容器 (k8s)环境需要自己手 动开启。	true/false, 默认 false。	1.2.2.2
spring.schedulerx2.cgr oupPathPrefix	容器内cgroup的路径。	默认 是/sys/fs/cgroup/cpu/ ,如果存在该路径则不需 要设置。	1.2.2.2
spring.schedulerx2.ena bleHeartbeatLog	是否打印心跳日 志,\${user.home}/logs /schedulerx/heartbeat .log	true/false, 默认true	1.2.4
spring.schedulerx2.ma pMasterStatusCheckInt erval	设置Map模型检测所有子 任务结束的频率,单位毫 秒。如果是秒级别任务想 要加快调度频率,可以设 置。	3000	1.2.5.2
spring.schedulerx2.ena bleSecondDealyCycleIn tervalMs	设置second_delay延迟 的单位为毫秒。如果把这 个值设置为true,控制台 设置的秒级别延迟将会变 成毫秒,可以加快调度频 率。	true/false, 默认false	1.2.5.2

3. 在应用中创建类 JobProcessor , 实现任务调度。

本文仅介绍如何实现一个最简单的定时打印 "Hello SchedulerX2.0" 的 JobProcessor 类。

package com.aliyun.schedulerx.test.job;
import com.alibaba.schedulerx.worker.domain.JobContext;
import com.alibaba.schedulerx.worker.processor.JavaProcessor;
import com.alibaba.schedulerx.worker.processor.ProcessResult;
@Component
public class MyHelloJob extends JavaProcessor {
@Override
public ProcessResult process(JobContext context) throws Exception {
System.out.println("hello schedulerx2.0");
return new ProcessResult(true);
}
}

结果验证

- 1. 客户端接入完成,将该应用发布到阿里云。
- 2. 登录分布式任务调度平台。
- 3. 在顶部菜单栏选择地域。
- 4. 在左侧导航栏单击应用管理。
- 5. 在应用管理页面查看实例总数。
 - 如果**实例总数**为0,则说明应用接入失败。请检查、修改本地应用。
 - 如果实例总数不为0,显示接入的实例个数,则说明应用接入成功。在操作列单击查看实例,即可 在连接实例对话框中查看实例列表。

应用管理 所屬命名	空间:	~				
创建应用 应用ID/应用	用名称 > 済输入	Q				¢ C
应用名称/应用描述	应用ID/应用key	最后更新人	已有/最大任务数	实例总数 🕄 🖡	操作	
schedulerx-fatjar 暂无描述		10.000/00-000.00000000000000000000000000	6 / 1000	0	编辑 授权 操作记录	:
schedulerx-fatjar 暂无描述	Street Street Street	the difference on conversion	18 / 1000	2	查看实例编辑 一授权	:

后续步骤

应用接入SchedulerX完成后,即可在分布式任务调度平台创建调度任务。详情请参见创建调度任务。

2.4. Agent 接入(调度任务)

如果您无需为应用接入调度任务,仅想创建一个独立的脚本调度任务,也可以使用SchedulerX提供的Agent 快速创建脚本任务。

前提条件

- 创建命名空间(可选)
- 创建应用

背景信息

脚本任务目前支持Shell、Python和Go三种语言。

运行环境要求为JRE 1.8及以上版本。

操作步骤

- 1. 下载schedulerxAgent-1.1.0。
- 2. 解压下载的压缩包。
- 进入schedulerxAgent/conf目录,编辑agent.properties文件,添加endpoint、namespace(命名空间ID)、groupId(应用ID)和阿里云账号的aliyunAccessKey(AccessKeyID)和aliyunSecretKey(AccessKeySecret)。

endpoint= namespace= groupId= aliyunAccessKey= aliyunSecretKey= 地域(Region)和Endpoint的关系请参见Endpoint列表。

4. 进入 schedulerxAgent / bin 目录,执行 st art - 1g.sh 命令启动 SchedulerX。

? 说明 start-1g.sh仅为示例,您需要根据任务负载及机器配置情况执行对应的命令,如start-2 g.sh、start-4g.sh或start-8g.sh。

如果您想停止任务调度,可执行stop.sh命令。

2.5. 在本地接入公网测试环境

本文介绍如何将本地接入云上公网测试环境来进行本地开发、调试和验证。

前提条件

- 开通SchedulerX(免费)
- 创建应用

⑦ 说明 在创建应用时,请选择公网地域。

背景信息

云上公网环境不保证稳定性和数据安全性,只是方便本地开发调试使用。请测试验证完成后,删除任务并在 线上正式环境创建相同的任务。

操作步骤

1. 在应用pom.xml文件中添加SchedulerxWorker依赖。

针对不同应用, 在初始化SchedulerxWorker时会有所不同。

○ 普通Java或Spring应用

```
<dependency>
<groupId>com.aliyun.schedulerx</groupId>
<artifactId>schedulerx2-worker</artifactId>
<version>${schedulerx2.version}</version>
<!--如果用的是logback,需要把log4j和log4j2排除掉-->
<exclusions>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-api</artifactId>
 </exclusion>
 <exclusion>
  <groupId>org.apache.logging.log4j</groupId>
  <artifactId>log4j-core</artifactId>
 </exclusion>
 <exclusion>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
 </exclusion>
</exclusions>
</dependency>
```

○ Spring Boot 应用

<dependency> <groupId>com.aliyun.schedulerx</groupId> <artifactId>schedulerx2-spring-boot-starter</artifactId> <version>\${schedulerx2.version}</version> <!--如果用的是logback,需要把log4j和log4j2排除掉--> <exclusions> <exclusion> <groupId>org.apache.logging.log4j</groupId> <artifactId>log4j-api</artifactId> </exclusion> <exclusion> <groupId>org.apache.logging.log4j</groupId> <artifactId>log4j-core</artifactId> </exclusion> <exclusion> <groupId>log4j</groupId> <artifactId>log4j</artifactId> </exclusion> </exclusions> </dependency>

2. 初始化SchedulerxWorker。

针对不同应用, 在初始化SchedulerxWorker的时候会有所不同。

○ 普通Java应用

在main函数中初始化SchedulerxWorker。

初始化SchedulerxWorker时,会用到您部署应用的**公网**地域(Region)和对应的 Endpoint(**acm.aliyun.com**)。

- a. 在控制台顶部菜单栏将地域切换为公网。
- b. 在左侧导航栏单击命名空间,在命名空间页面查看命名空间ID(Namespace)。

☰ (-) 阿里云 🌼 🗉	神台 公用 ~ 1		Q 鼓索 费用 I单 ICP 备果 企业 支持 App 区 🗘 🗑 🕥 简体 🧕
分布式任务调度平台 _{测试环境}	创建命名空间		総系形(1: ● 23153765 (2)(総約文代)
执行列表	命名空间名称/描述	命者空间D 🗲	操作
任务管理	默认命名空间 暂无描述	87fdc176-a451-434e-a 🖬 💵 🖪	5638 B000
流程管理	wuch-DOC test	5dc77f6f-bae6-47d8-4t	1010 H (1010
应用管理			毎夏星示 10 🗸 共2歳 < 上一页 1 下一页 >
* ⁸ 24 2			
操作记录			

c. 在左侧导航栏单击应用管理, 在应用管理页面查看应用ID (groupId) 和应用 key (appKey)。

	â I作台 公网 v 1		Q 投索	费用 工单 ICP 备案 企业 支持	i App 🔄 🗘 🗑 🕲 1814 🧕
分布式任务调度平台 _{测试环境}	应用管理 ศิสสะสอาตะ มนะสออต >				联系我们: 🕥 23133165 🗋 (帮助文格)
执行列表	811款の用 昼用の液用名称 > 情報入 Q				© \$
	应用名称/应用描述 应用ID/应用key	已有/最大任务数	英例总数 ❷↓	最后更新人	操作
流程管理 应用管理 2	doc-test 設定で2FVat6ummanament	0/1000	0	1988 - ali 19 (2435), an ar 6/1/ 104)	编辑 删除 授权 :
命名空间				每页显示 10	▶ 共振 < 上一页 1 下一页 >
操作记录					

d. 将光标移动到控制台右上角的用户头像上,在弹出的列表中单击AccessKey管理, 在AccessKey管理页面获取AccessKeyID(AliyunAccessKey)和 AccessKeySecret(AliyunSecretKey)。 public void initSchedulerxWorker() throws Exception {
 SchedulerxWorker schedulerxWorker = new SchedulerxWorker();
 schedulerxWorker.setEndpoint("xxxx");
 schedulerxWorker.setNamespace("xxxx");
 schedulerxWorker.setGroupId("xxxx");
 //1.2.1及以上版本需要设置应用key
 schedulerxWorker.setAppKey("xxxx");
 //1.2.1以下版本需要设置AK/SK
 //schedulerxWorker.setAliyunAccessKey("xxxx");
 //schedulerxWorker.setAliyunSecretKey("xxxx");
 schedulerxWorker.init();
}

? 说明

- 一个应用如果包含多个业务,或者想把定时任务进行归类,可以建立多个分组,例如应用 animals 新建了两个分组 animals.dogs 和 animals.cats。此时不用申请两批实例分别接入这两个分组,在应用客户端中将这两个分组配置到 groupId= 后面即可,例如 groupId=animals.dogs,animals.cats。
- 在初始化SchedulerxWorker客户端时,如果还有其它配置需求,可以参考SchedulerxWorker 配置参数说明添加配置。

○ Spring应用

在xml配置文件中注入SchedulerxWorker Bean。

- a. 在控制台顶部菜单栏将地域切换为公网。
- b. 在左侧导航栏单击命名空间,在命名空间页面查看命名空间ID(Namespace)。

☰ (-) 阿里云 🌼 🗉	作台 公网 > 1		Q. 我家	हुम Iमे ICP केंस्र 🖭 इसे App 🖸 🕼 🗑 🖗 🎯
分布式任务调度平台 测试环境	制建命名空间			联系 (数1): 🕥 23133165 🗋 (常約文信) (第一) C
执行列表	命名空间名称/描述	命名空间D 🌢		操作
任务管理	默认命名空间 智元描述	87fdcf76-3451-434e-2* 🖬 = 🖬 📲 🚮		56638 19939R
流程管理	wuch-DOC test	5dc77f6f-bae6-47d8-cut		56410 BIDR
应用管理				毎夏星示 10 ~ 共2条 く 上一页 1 下一页 >
#名空间 2				
操作记录				

c. 在左侧导航栏单击应用管理, 在应用管理页面查看应用ID (groupId) 和应用 key (appKey)。

	a Iff 公网 ~ 1		Q 提案	费用 工单 ICP 备案 企业	支持 App 🖸 🗳 🐨 🕐 前体 🤵
分布式任务调度平台 测试环境	应用管理 #周南台空间: 新山台名空间 ~				联系我们: 🚫 23133165 🗋 (際助文格)
执行列表	111歳広用 国用の広用名称 > 信組入 Q				\$ C
世界自住	应用各称/应用描述 应用ID/应用key	已有/最大任务数	交例总数 ●1	最后更新人	操作
流程管理	doc-test doc-test	0.11000		104020	4048 B420 4040 1
应用管理 2	■无描述 UaEYZFYat6Laga -	071000	0		98194 10347 1245A 1
命名空间				每页显示 14) ・ 共1条 く 上一页 1 下一页 >
10/01/2010					

d. 将光标移动到控制台右上角的用户头像上,在弹出的列表中单击AccessKey管理, 在AccessKey管理页面获取AccessKeyID(AliyunAccessKey)和 AccessKeySecret(AliyunSecretKey)。 <bean id="schedulerxWorker" class="com.alibaba.schedulerx.worker.SchedulerxWorker"> <property name="endpoint"> <value>\${endpoint}</value> </property> <property name="namespace"> <value>\${namespace}</value> </property> <property name="groupId"> <value>\${groupId}</value> </property> <!--1.2.1及以上版本设置appKey --> <property name="appKey"> <value>\${appKey}</value> </property> <!--1.2.1以下版本需要设置AK/SK --> <!--<property name="aliyunAccessKey"> <value>\${aliyunAccessKey}</value> </property> <property name="aliyunSecretKey"> <value>\${aliyunSecretKey}</value> </property> --> </bean>

○ Spring Boot 应用

在application.properties文件中添加如下配置:

- a. 在控制台顶部菜单栏将地域切换为公网。
- b. 在左侧导航栏单击命名空间,在命名空间页面查看命名空间ID(Namespace)。

☰ (-) 阿里云 💿	C作台 公网 > 1		Q 报索	费用 I单 ICP备乘 企业 支持 App 🖸 🛕 🗑 🛞 简体 🙆
分布式任务调度平台 测试环境	结膜命名空间			联系例1): 🥎 23133165 🗋 (開助文性) (単) C
执行列表	命名空间名称/描述	命名空间日 🖌		操作
任务管理	默认命名空间 智无描述	87/dc176-a451-434e 🖬 - 🛤 🔞		編輯 一副除
流程管理	wuch-DOC test	5dc77f61-bae6-47d8-4		10230 MID2
应用管理				毎頁显示 10 🗸 共2条 < 上一页 1 下一页 >
#名空间 2				
操作记录				

c. 在左侧导航栏单击**应用管理**, 在**应用管理**页面查看**应用ID**(groupld)和**应用 key**(appKey)。

☰ (-) 阿里云 🌼	1作台 公网 ~ 1		Q 授家	费用 工单 ICP 备案 企业 支	H App 🖸 🗘 🗑 🔞 🕅 🥘
分布式任务调度平台 _{测试环境}					联系我们: 🚫 23133165 🗋 (際助文档)
执行列表	创建应用 互用D/应用各称 Y 该输入 Q				© 0
14.77 18 14	应用名称/应用描述 应用ID/应用key	已有/最大任务数	英例总数 🚱 🕇	最后更新人	操作
流程管理 应用管理 2	doc-test 發光描述 3 doc-test UuEVZFVat6t	0/1000	0	1998 - 199 (24353),	编辑:删除:授权::
命名空间				每页显示 10	▶ 共1条 < 上一页 1 下一页 >
操作记录					

d. 将光标移动到控制台右上角的用户头像上,在弹出的列表中单击AccessKey管理, 在AccessKey管理页面获取AccessKeyID(AliyunAccessKey)和 AccessKeySecret(AliyunSecretKey)。 spring.schedulerx2.endpoint=\${endpoint} spring.schedulerx2.namespace=\${namespace} spring.schedulerx2.groupId=\${groupId} # 1.2.1及以上版本设置appKey spring.schedulerx2.appKey=\${appKey} # 1.2.1以下版本设置AK/SK #spring.schedulerx2.aliyunAccessKey=\${aliyunAccessKey} #spring.schedulerx2.aliyunSecretKey=\${aliyunSecretKey}

② 说明 一个应用如果包含多个业务,或者想把定时任务进行归类,可以建立多个分组,例如 应用 animals 新建了两个分组 animals.dogs 和 animals.cats 。此时不用申请两批实例分别接入 这两个分组,在应用客户端中将这两个分组配置到 groupId= 后面即可,例如 groupId=animals.d ogs,animals.cats 。

3. 在应用中创建类 JobProcessor , 实现任务调度。

本文仅介绍如何实现一个最简单的定时打印 "Hello SchedulerX2.0" 的 JobProcessor 类。

```
package com.aliyun.schedulerx.test.job;
import com.alibaba.schedulerx.worker.domain.JobContext;
import com.alibaba.schedulerx.worker.processor.JavaProcessor;
import com.alibaba.schedulerx.worker.processor.ProcessResult;
@Component
public class MyHelloJob extends JavaProcessor {
  @Override
  public ProcessResult process(JobContext context) throws Exception {
    System.out.println("hello schedulerx2.0");
    return new ProcessResult(true);
  }
}
```

4. 运行本地应用。

结果验证

- 1. 客户端接入完成,将该应用发布到阿里云。
- 2. 登录分布式任务调度平台。
- 3. 在顶部菜单栏选择地域。
- 4. 在左侧导航栏单击应用管理。
- 5. 在应用管理页面查看实例总数。
 - 如果**实例总数**为0,则说明应用接入失败。请检查、修改本地应用。
 - 如果实例总数不为0,显示接入的实例个数,则说明应用接入成功。在操作列单击查看实例,即可
 在连接实例对话框中查看实例列表。

应用管理 所属命名	5空间:	~				
创建应用 应用ID/应	用名称 ∨ 请输入	Q				\$ C
应用名称/应用描述	应用ID/应用key	最后更新人	已有/最大任务数	实例总数 🛛 🕇	操作	
schedulerx-fatjar 暫无描述	And a second sec	standays-second and and	6 / 1000	0	编辑 授权 操作记录	:
schedulerx-fatjar 暂无描述	United States		18 / 1000	2	查看实例 编辑 授权	:

2.6. 容器服务Kubernetes版接入SchedulerX

前提条件

- 创建Kubernetes专有版集群
- 开通SchedulerX(免费)
- 了解Kubernetes基本概念和操作,例如kubeconfig

背景信息

使用SchedulerX需要理解以下3个概念:

- 分组
 - 客户端的组织单位
 - 任务的组织单位
- 任务: 调度单位, 需要创建任务, 配置所属分组。
- 客户端:任务执行节点,需要添加SchedulerX客户端,实现对应Java任务处理接口,配置所属分组启动名为SchedulerxWorker的Agent。

三者的关系为:任务只能调度到对应分组的客户端。例如创建了分组group-sample,在该分组下创建任务 job-sample,同时配置所属分组为group-sample来启动客户端agent1、agent2和agent3,则调度任务 job-sample就会被调度到客户端agent1、agent2和agent3上面运行。

关于SchedulerX的更多信息,请参见分布式任务调度 SchedulerX。

在容器服务Kubernetes版中安装SchedulerX组件

- 1. 登录容器服务控制台。
- 2. 在左侧导航栏选择市场 > 应用目录。
- 3. 在应用目录页面搜索并单击ack-schedulerx。
- 4. 在ack-schedulerx页面右侧集群列表中选择集群,然后单击创建。
 - 如果是在ACK购买的K8s集群,选择集群后,直接创建。

创建
仅支持 Kubernetes 版本 1.8.4 及以上的集群。对于 1.8.1 版本的集群,您可以在集群列表中进行"集群升级"操作。
集群
· · · · · · · · · · · · · · · · · · ·
命名空间 schedulerx-system
发布名称
ack-schedulerx
创建

○ 如果是自建纳管到ACK的K8s集群,单击参数页签,设置参数controller.aliyun_acccessKey_secret_name。

⑦ 说明 命名空间为schedulerx-system,不可修改。

安装SchedulerX组件大约需要2分钟,请耐心等待。

创建成功后,会自动跳转到Helm发布列表 - ack-schedulerx页面,检查安装结果。

Helm 发布列表 - ack-schedulerx				刷新
当前版本				
发布名称: ack-schedulerx	命名空间: schedulerx-system		部署时间: 2020-04-14 21:09:46	
当前版本: 1				更新时间: 2020-04-14 21:09:46
资源			参数	
资源 🔷		类型 ♦		
schedulerx-certs		Secret		查看YAML
schedulerx-envs		ConfigMap		查看YAML
xcronjobs.schedulerx.alibabacloud.com		CustomResourceDefinition		查看YAML
xgroups.schedulerx.alibabacloud.com		CustomResourceDefinition		查看YAML
schedulerx-default-cluster-role		ClusterRole		查看YAML
schedulerx-rolebinding		ClusterRoleBinding		查看YAML
schedulerx-controller		Deployment		查看YAML
历史版本				

创建分组

1. 创建xgroup.yaml, 设置相关参数。

*xgroup.yam*l示例如下:

apiVersion: schedulerx.alibabacloud.com/v1alpha1 kind: XGroup metadata: name: xgroup-sample spec: appName: ackApp

xgroup.yaml 配置参数说明:

- 。 GVK (Group、Version 和 Kind) 信息
 - apiVersion:格式为 <group>/<version> ,例如 schedulerx.alibabacloud.com/v1alpha1 。
 - kind: XGroup
- spec信息

参数名	类型	默认值	是否必填	说明
appName	string	无	必填	应用名,用户自定 义,用于后续管 理。

- 2. 执行 kubectl apply -f xgroup.yaml 命令, 创建任务分组。
 - ? 说明
 - 分组创建后,不允许更新。如果需要更新,请删除分组后重新创建。
 - 分组创建后,如果新建了任务或添加了客户端,即该分组下包含任务或客户端,则无法删除 分组,需要先移除任务和客户端。
- 3. 执行 kctl get xgroup xgroup-sample -o yaml 命令, 查看xgroup资源。

打印结果如下:

apiVersion: schedulerx.alibabacloud.com/v1alpha1
kind: XGroup
metadata:
annotations:
kubectl.kubernetes.io/last-applied-configuration:
{"apiVersion":"schedulerx.alibabacloud.com/v1alpha1","kind":"XGroup","metadata":{"annotations"
:{},"name":"xgroup-sample","namespace":"default"},"spec":{"appName":"ackApp"}}
creationTimestamp: "2019-09-19T04:21:12Z"
finalizers:
- GroupCleanup
generation: 1
name: xgroup-sample
namespace: default
resourceVersion: "143176160"
selfLink: /apis/schedulerx.alibabacloud.com/v1alpha1/namespaces/default/xgroups/xgroup-sample
uid: e9a1xxxx-xxxx-11e9-xxxx-be9f1a43xxxx
spec:
appName: ackApp
status:
appGroupId: 283
conditions:
- lastTransitionTime: "2019-09-19T04:21:12Z"
lastUpdateTime: "2019-09-19T04:21:12Z"
reason: CreateGroupSuccess
status: "True"
type: Ready

status的conditions下status为True,且type为Ready,则表示分组创建成功。

也可以登录<mark>分布式任务调度平台</mark>,查看分组创建情况。

	杭州) - 2		Q 搜索文档、控制台、API、解决方案和资源	期 工单 俗案 1	业 支持	言网	Ξ Δ́	Ä	⑦ 简体	0
分布式任务调度平台	应用管理 📠	*名空间: 华东1	× 3							
执行列表										
任务管理	创建应用 应用ID/	应用名称 > 清输入	Q						۲	G
流程管理	应用名称/应用描述	应用ID/应用key	最后更新人	已有/最大任务务	实例总	波 🕢 🗤	操作			
应用管理 1	4	And the Association of the	All CONTRACTORY (1997)	1/1000		0	编辑	授权制	NYFIC录 :	
命名空间							44.9.47	/ L ==	—	
操作记录					每贝显示 I) ~	共1条		1 1-	

创建任务

1. 创建xcronjob.yaml, 设置相关参数。

*xcronjob.yam*l示例如下:

apiVersion: schedulerx.alibabacloud.com/v1alpha1 kind: XCronJob metadata: name: xcronjob-sample spec: group: xgroup-sample jobType: java jobProcessor: processor.SimpleJobProcessor executeMode: standalone timeExpression: 0 0 2 * * ?

xcronjob.yam配置参数说明:

- 。 GVK (Group、Version 和 Kind) 信息
 - apiVersion: 格式为 <group>/<version> , 例如 schedulerx.alibabacloud.com/v1alpha1 。
 - kind: XCronJob

o spec信息

参数名	类型	默认值	是否必填	说明
group	string	无	是	该任务所属分组 名。
jobType	string	java	否	任务类型,指实现 任务的编程语言, 当前支持Java、 Python、Shell和 Go。详情请参 见多种调度任务类 型。
jobProcessor	string	无	否(有条件)	任务实现全限定类 名,如果 jobTyp e == java , 该字 段必填。
content	string	无	否(有条件)	任务实现代码,如 果 jobType == ja va ,该字段必 填。
executeMode	string	standalone	否	任务执行模式,当 前支持单击运行、 广播运行、并行计 算、内存网格、网 格计算和分片运 行。详情请参见 <mark>分</mark> <mark>布式计算</mark> 。
description	string	无	否	任务描述

快速入门·客户端快速接入SchedulerX

参数名	类型	默认值	是否必填	说明
timeType	int	1	否	任务调度表达式类 型,当前支持 cron(1)、 fix_rate(3)、 second_delay(4) 。详情请参 见Cron、Fixed rate和Second delay。
timeExpression	string	无	是	任务调度表达式, 例如: • cron: 002* *?要确保频率 大于分钟级。 • fixed rate: 3 0(>0),单位 是s,每30s运 行一次。 • second delay : 2(1-60), 单位是s,上次 运行结束后延 迟2s再运行一次。
parameters	string	无	否	任务参数,可以在 任务运行时从上下 文获取。
maxConcurrency	int	1	否	最大同时运行任务 实例数,默认为 1。超过该并发度 的调度实例会被忽 略。
retryMaxAttempt s	int	0	否	失败重试次数,默 认为0,不重置。
retryInterval	int	30	否	失败重试间隔,单 位s,默认为30s。

通过示例可以看到指定的group是刚刚创建的xgroup-sample,默认使用Cron调度表达式,Java任务 类型,处理的接口类名为processor.SimpleJobProcessor。

- 2. 执行 kubectl apply -f xcronjob.yaml 命令, 创建任务。
- 3. 执行 kctl get xcronjob xcronjob-sample -o yaml 命令, 查看xcronjob资源。

打印结果如下:

apiVersion: schedulerx.alibabacloud.com/v1alpha1 kind: XCronJob metadata: creationTimestamp: "2019-09-19T06:33:13Z" finalizers: - JobCleanup generation: 1 name: xcronjob-sample namespace: default ownerReferences: - apiVersion: schedulerx.alibabacloud.com/v1alpha1 blockOwnerDeletion: true controller: true kind: XGroup name: xgroup-sample uid: e9a1xxxx-xxxx-11e9-xxxx-be9f1a43xxxx resourceVersion: "143570391" selfLink: /apis/schedulerx.alibabacloud.com/v1alpha1/namespaces/default/xcronjobs/xcronjob-sampl e uid: 5b5exxxx-daa7-xxxx-a76d-4af3xxxx44xx spec: executeMode: standalone group: xgroup-sample jobProcessor: processor.SimpleJobProcessor jobType: java timeExpression: 0 0 2 * * ? status: conditions: - lastTransitionTime: "2019-09-19T06:33:13Z" lastUpdateTime: "2019-09-19T06:33:14Z" reason: JobUpdateSuccess status: "True" type: Ready jobld: 1304

status的conditions下status为True,且type为Ready,则表示任务创建成功。

也可以登录分布式任务调度平台,查看任务创建情况。

	(杭州) • 2	Q 搜索文档、控制台、API、解决方案和资源	费用 工单 备案 企业 支持 官网 🛽	I 4° I 0 mak 🥑
分布式任务调度平台	任务管理 所屬會容明: #51	~ 3		
执行列表				
任务管理 1	60建任务 全部应用 ∨ 任务名 ∨ 1	Niii)、 Q		\$ C
流程管理	任务ID 🖈 基本信息 应用ID 任务类型/执行方	式 时间类型/时间表达式 工作流D 小	最后更新人	操作
应用管理	4 💼 📖 🚛 🦛 🌜 🌜 単机运行	cron N/A 20 */1 * * * ?		编辑 运行一次 禁用 :
命名空间			毎页显示 10 ~	共1条 < 上一页 1 下一页 >

创建客户端

1. 客户端接入SchedulerX。

根据应用类型不同,客户端的接入方式也不同。

- Java应用:客户端接入详情请参见Java应用接入SchedulerX。
- Spring应用:客户端接入详情请参见Spring应用接入SchedulerX。

。 Spring Boot应用:客户端接入详情请参见Spring Boot应用接入SchedulerX。

2. 将客户端打包,并通过Dockerfile创建镜像,并上传到镜像仓库。

Dockfile示例如下:

FROM openjdk:8-jdk-alpine COPY ./target/schedulerx-k8s-demo-1.0-SNAPSHOT-spring-boot.jar app.jar ENTRYPOINT ["java","-jar","/app.jar"]

SchedulerX提供了已经上传到镜像仓库的镜像Demo(registry.cn-shanghai.aliyuncs.com/schedulerx/demo:latest),方便您体验、试用。

3. 创建xagentpool.yaml, 设置相关参数。

*xagentpool.yam*标例如下:

apiVersion: schedulerx.alibabacloud.com/v1alpha1 kind: XAgentPool metadata: name: xagentpool-sample spec: group: xgroup-sample replicas: 2 template: containers: - name: standalone image: registry.cn-shanghai.aliyuncs.com/schedulerx/demo:latest

xagentpool.yam 配置参数说明:

- 。 GVK (Group、Version 和 Kind) 信息
 - apiVersion:格式为 <group>/<version> ,例如 schedulerx.alibabacloud.com/v1alpha1 。
 - kind: XGroup
- spec信息

参数名	类型	默认值	是否必填	说明
group	string	无	是	该任务所属分组 名。
replicas	int	无	是	执行器个数。
workloadType	string	Deployment	否	执行器工作负载, 默认是 Deployment,可 选值还有 DaemonSet。
template	PodSpec	无	是	任务执行器Pod模 板。

通过示例可以看到指定的group是刚刚创建的xgroup-sample,运行两个执行器,执行器镜像为 image,该image即客户端镜像。

4. 执行 kubectl apply -f xagentpool.yaml 命令, 创建客户端。

5. 执行 kctl get xagentpool xagentpool-sample -o yaml 命令, 查看 xagent pool资源。

打印结果如下:

```
apiVersion: schedulerx.alibabacloud.com/v1alpha1
kind: XAgentPool
metadata:
annotations:
 kubectl.kubernetes.io/last-applied-configuration: |
  {"apiVersion":"schedulerx.alibabacloud.com/v1alpha1","kind":"XAgentPool","metadata":{"annotati
ons":{},"name":"xagentpool-sample","namespace":"default"},"spec":{"group":"xgroup-sample","replic
as":2,"template":{"containers":[{"image":"registry.cn-shanghai.aliyuncs.com/schedulerx/demo:latest",
"name":"standalone"}]}}}
creationTimestamp: "2019-09-25T10:11:39Z"
generation: 1
name: xagentpool-sample
namespace: default
ownerReferences:
 - apiVersion: schedulerx.alibabacloud.com/v1alpha1
 blockOwnerDeletion: true
 controller: true
 kind: XGroup
 name: xgroup-sample
 uid: c920xxxx-df7c-xxxx-a76d-xxxx350bxxxx
resourceVersion: "170986882"
selfLink: /apis/schedulerx.alibabacloud.com/v1alpha1/namespaces/default/xagentpools/xagentpool-s
ample
uid: dd83xxxx-df7c-xxxx-a156-xxxx1a43xxxx
spec:
group: xgroup-sample
replicas: 2
template:
 containers:
 - image: registry.cn-shanghai.aliyuncs.com/schedulerx/demo:latest
  name: standalone
  resources: {}
status:
conditions:
 - lastTransitionTime: "2019-09-25T10:11:40Z"
 lastUpdateTime: "2019-09-25T10:11:40Z"
 reason: update deployment sucess
 status: "True"
 type: Ready
```

status的conditions下status为True,且type为Ready,则表示任务创建成功。

实际上,每个agentPool的创建都会在相同命名空间下创建名为 [agentPoolName]-deployment 的 Deployment或者 [agentPoolName]-daemonset 的DeamonSet,可以自行查看

也可以登录分布式任务调度平台,查看客户端创建情况。

	杭州) • 2		Q 搜索文档、控制台、API、解决方案和资源	费用	I# %	i案 企1	Ł 支持	官网	۶	۵.	Ä	(?) (i)	ф 🌔
分布式任务调度平台			~ 3										
执行列表													
任务管理	创建应用 应用ID/应用名称 >>	请输入 Q										4	¢ C
流程管理	应用名称/应用描述 应用ID/应用	key	最后更新人		已有/最大	、任务数	实例总	数 🛛 📲	1	操作			
应用管理 1	暂无描述	-0.00	$(1,1,1,2,2,3,3)_{1,1} \neq (1,2,2,2,3,3) \neq (1,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,$		6/1	000		0	1	编辑 授	权 操	印记录	
命名空间 操作记录	暂无描述	in a little			18/1	000		2		查看实例	编辑	授权	
	暂无描述				2/1	000		0	1	编辑丨授	权 操	印记录	
	<					6	顶显示 1	0	~ #	3条 <	上一页	1 下	

单击查看实例,可以查看客户端实例的详细信息。

删除SchedulerX组件

当不再需要SchedulerX组件时,可以删除chedulerX组件。

? 说明

- 在删除SchedulerX组件之前,请确保集群内的所有XGroup、XCronJob和XAgentPool类型资源都已经删除完毕,否则无法删除CRDs,并且会导致下次安装出现异常。
- XGroup删除之后,对应的SchedulerX应用分组不会自动删除,需要登录分布式任务调度平台, 手动删除。
- 1. 登录容器服务控制台。
- 2. 在左侧导航栏选择应用 > 发布。
- 3. 在发布页面ack-schedulerx的操作列单击删除。
- 4. 在删除应用对话框中单击确定。

2.7. Endpoint列表

初始化SchedulerxWorker时,会用到您部署应用的地域(Region)和对应的Endpoint。

地域 (Region)	Endpoint	用途
华东1(杭州)	addr-hz-internal.edas.aliyun.com	线上生产环境
华东2(上海)	addr-sh-internal.edas.aliyun.com	线上生产环境
华北2(北京)	addr-bj-internal.edas.aliyun.com	线上生产环境
华北3(张家口)	addr-cn-zhangjiakou- internal.edas.aliyun.com	线上生产环境
华南1(深圳)	addr-sz-internal.edas.aliyun.com	线上生产环境
中国香港	addr-hk- internal.edas.aliyuncs.com	线上生产环境
新加坡(新加坡)	addr-singapore- internal.edas.aliyun.com	线上生产环境

地域(Region)	Endpoint	用途
华北2政务云1	ddr-cn-north-2-gov-1- internal.edas.aliyun.com	线上生产环境
上海金融云	addr-cn-shanghai-finance-1- internal.edas.aliyun.com	线上生产环境
美国(弗吉尼亚)	addr-us-east-1- internal.acm.aliyun.com	线上生产环境
公网	acm.aliyun.com	本地接入测试环境,不能用于生产。